



Институт системного программирования
Российской академии наук

В.В. Липаев

ЭКОНОМИКА ПРОИЗВОДСТВА ПРОГРАММНЫХ ПРОДУКТОВ

Издание второе

УДК 004.41(075.8)
ББК 32.973.26-018я73

Л61

Липаев В.В.

Экономика производства программных продуктов. Издание второе - М.: СИНТЕГ, 2011. - 358 с.

В монографии представлены основы экономики производства сложных программных продуктов высокого качества, которые базируются на традиционных принципах и методах экономики разработки сложных технических систем. Создание таких программных продуктов связанных с большими затратами, определяет необходимость их строгого планирования, формализации и стандартизации производственных процессов, а также экономического контроля и сопровождения, подобных применяемым при производстве других крупных промышленных изделий. Для конкретизации рассматриваемых задач и процессов изложены основы экономики индустриального производства сложных технических систем, а также особенности программной инженерии промышленного производства программных продуктов. Представлены факторы, определяющие экономику технологии создания компонентов и комплексов программ, основные современные методы прогнозирования их трудоемкости, длительности и числа необходимых специалистов для проектирования и производства программных продуктов высокого качества с учетом их характеристик и ограничений ресурсов. Значительное внимание уделено процессам организации, испытаний и сертификации создания программных продуктов. При изложении методов и процессов, которые являются основой для определения их экономических характеристик, активно используются международные стандарты в этой сфере.

Монография предназначена для руководителей, системных аналитиков и экономистов, ответственных за анализ, прогнозирование и контроль экономических характеристик при проектировании и производстве сложных программных продуктов.

ББК 32.973.26-018я73

ISBN 978-5-89638-116-7

© **В.В. Липаев**, автор, 2011

© **ООО «НПО СИНТЕГ»**, издательство, 2011

ОГЛАВЛЕНИЕ

Введение	6
Глава 1. Основы экономики индустриального производства сложных технических систем и продуктов	21
Общие принципы экономики и организации производства сложных технических систем и продуктов. Планирование и управление производством сложных технических систем и продуктов. Стандартизация производства сложных технических систем и продуктов. Бизнес-план, организация и анализ предприятий для производства сложных технических систем и продуктов.	
Глава 2. Задачи и основные особенности экономики производства сложных программных продуктов	47
Задачи анализа реальной экономики производства программных продуктов. Задачи организации экономически эффективного производства программных продуктов. Задачи организации коллективов специалистов для экономически эффективного производства программных продуктов.	
Глава 3. Экономические характеристики для оценивания производства программных продуктов	73
Статистические исследования экономики производства программных продуктов. Характеристики трудоемкости производства программных продуктов. Характеристики длительности разработки программных продуктов. Вспомогательные характеристики производства программных продуктов.	
Глава 4. Сложность и размер – основные факторы, определяющие экономические характеристики производства программных продуктов	99
Основные факторы, определяющие сложность производства программных продуктов. Единицы измерения сложности и размера программ для экономического анализа процессов их производства. Масштаб – размер комплексов программ по числу строк текста и объему занятой памяти. Масштаб – размер комплексов программ по числу функциональных точек.	
Глава 5. Характеристики качества программных продуктов, влияющие на экономику их производства	123
Влияние качества программных продуктов на экономические характеристики производства. Стандартизированные характеристики качества сложных программных продуктов. Гипотетические примеры определения требований к характеристикам качества двух типов программных продуктов.	

Глава 6. Экономика переноса и повторного использования компонентов при производстве программных продуктов	148
Цели и организация переноса и повторного использования программных компонентов. Факторы, определяющие экономическую эффективность производства программных продуктов на базе готовых компонентов. Оценивание экономической эффективности производства программных продуктов при повторном использовании компонентов. Оценивание экономической эффективности производства программных продуктов при повторном использовании информации баз данных. Применение методов и комплекса стандартов интерфейсов Открытых систем при производстве сложных программных продуктов.	
Глава 7. Производственные процессы, определяющие экономику сложных комплексов программ	175
Базовые производственные процессы сложных систем и программных комплексов. Производственные процессы обеспечения качества и верификации компонентов и комплексов программ. Производственные процессы тестирования сложных комплексов программ. Производственные процессы документирования сложных комплексов программ.	
Глава 8. Модели прогнозирования экономических характеристик производства программных продуктов	205
Методы и достоверность прогнозирования экономических характеристик производства программных продуктов. Экспертное прогнозирование экономических характеристик производства программных продуктов. Простейшие модели прогнозирования экономических характеристик производства программных продуктов. Модель прогнозирования экономических характеристик производства программных продуктов СОСОМО II.	
Глава 9. Влияние основных характеристик производства программных продуктов на экономические прогнозы при использовании модели СОСОМО II	236
Влияние масштабных факторов производства программных продуктов при прогнозировании экономических характеристик. Влияние свойств программных продуктов при прогнозировании экономических характеристик производства. Влияние свойств специалистов при прогнозировании экономических характеристик производства программных продуктов. Влияние технологической среды производства программных продуктов при прогнозировании экономических характеристик. Влияние аппаратной вычислительной среды при прогнозировании экономических характеристик производства программных продуктов. Примеры прогнозирования экономических характеристик производства программных продуктов.	

Глава 10. Экономика планирования производства компонентов и программных продуктов	273
Основы технологии производства компонентов и программных продуктов. Декомпозиция требований, структуры, процессов производства комплексов программ и сборка компонентов. Планирование и экономика производства модулей и компонентов для комплекса программ. Экономика и планирование производства программных продуктов. Применения графиков Ганта для экономического планирования производства программных продуктов.	
Глава 11. Экономика обеспечения и удостоверения качества при производстве сложных программных продуктов	305
Организация, экономика и управление качеством производства программных продуктов. Понятия и свойства дефектов и ошибок в сложных комплексах программ. Организация, экономика и процессы испытаний сложных комплексов программ. Стандартизация и сертификация для обеспечения качества производства сложных программных продуктов.	
Приложение. Перечень основных стандартов, регламентирующих проектирование и производство сложных программных продуктов	335
Литература	339

ВВЕДЕНИЕ

Во многих предприятиях и в вузах отношение к созданию программ для ЭВМ исторически базируется на подходе как к «искусству и художественному творчеству» отдельных специалистов. При этом считается, что *невозможно применять какие либо экономические характеристики* для определения стоимости и результатов такого творчества, и они оцениваются только с позиции качества выполняемых функций и «эстетики» их реализации. Такие программы создаются преимущественно для получения конкретных результатов исследований или для анализа относительно простых процессов обработки информации. Они не предназначены для массового тиражирования и распространения как программный продукт на рынке, их оценивают качественно и интуитивно, преимущественно как «художественные произведения». При этом, как правило, нет конкретного независимого заказчика-потребителя, определяющего требования к программам и их финансирование, программы не ограничиваются допустимой стоимостью, трудоемкостью и сроками их создания, требованиями обеспечения заданного качества и документирования. Их разработчики не знают и не применяют регламентирующих, нормативных документов, вследствие чего жизненный цикл таких изделий имеет *не предсказуемый характер* по качеству и стоимости основных результатов «творчества».

Для небольших относительно простых проектов программных комплексов во многих случаях достаточно достоверными могут быть *интуитивные оценки требуемых экономических ресурсов*, выполняемые опытными руководителями, реализовавшими несколько аналогичных проектов. Такие оценки размеров, сложности, трудоемкости и сроков создания конкретных программных проектов, как правило, *отличаются существенными недостатками и ошибками* вследствие того, что [7]:

- человек, в основном, оптимистичен, многим руководителям проект комплекса программ представляется меньше по размеру, более простым и дешевым, что ведет к первоначальным недооценкам

его сложности и к конфликтным ситуациям между заказчиками и разработчиками;

- человек обычно не полностью использует предыдущий опыт о сложности функций аналогичных проектов и особенно о большом размере вспомогательных компонентов комплексов программ, которые должны быть разработаны;

- отдельные специалисты, как правило, не знакомы со всем размером проекта и потребностями пользователей, что приводит к недооценке второстепенных функций и программных компонентов, а также накопленных знаний при оценивании размера и сложности проекта.

Следствием этого являются **большие ошибки при планировании экономических характеристик** – сроков, трудоемкости и стоимости создания программных продуктов. В большинстве случаев это приводит к значительному запаздыванию завершения разработок и превышению предполагавшихся затрат ресурсов. Вследствие пренебрежения тщательным экономическим обоснованием до 15% проектов сложных программных комплексов не доходит до завершения, а почти половина проектов не укладывается в выделенные ресурсы, бюджет и сроки, не обеспечивает требуемые характеристики качества. Типичны ситуации, когда отставание сроков внедрения больших промышленных систем управления и обработки информации полностью **зависит от неготовности для них программных продуктов**.

Массовое создание сложных и дорогих программных продуктов промышленными методами и большими коллективами специалистов (в основном, для военных систем) вызвало необходимость их достоверного **экономического прогнозирования и анализа**, четкой организации производства, планирования работ по затратам, этапам и срокам реализации. Для решения этих задач еще в **80-е годы** начала формироваться новая область знания и инженерная дисциплина – **экономика создания сложных программных продуктов** [3, 16]. Необходимо было освоить специалистам анализ и оценивание конкретных факторов, влияющих на экономические характеристики проектов программных продуктов вследствие реально существующих и потенциально возможных воздействий и ограничений ресурсов проектов комплексов программ. Это привело к появлению новой области экономической науки и практики – **экономики проектирования, произ-**

водства и жизненного цикла сложных программных продуктов как части экономики в промышленности и вычислительной технике в составе общей экономики некоторых предприятий. Ее основной задачей являлись анализ, прогнозирование, эффективное управление, распределение ресурсов и экономное использование необходимых быстро возрастающих капиталовложений в производство сложных комплексов программ высокого качества и различного назначения.

Развитие этой области экономики *связано с большими профессиональными и психологическими трудностями*, типичными для новых разделов современной науки и техники, появляющимися на стыке различных областей методов и знаний. В данном случае особенности состоят в том, что *менеджеры и разработчики* сложных комплексов программ, как правило, не знают даже основ экономики промышленного производства сложной технической продукции, а *экономисты* не представляют сущность и свойства объектов разработки – программных продуктов, а также особенностей технологических процессов их производства и применения. Широкий спектр технических характеристик таких объектов, количественных и качественных факторов, которые с различных сторон характеризуют содержание компонентов и комплексов программ, и невысокая достоверность оценки их значений определяют значительную трудность при попытке описать и измерить свойства и качество создаваемых или используемых сложных программных продуктов *для их экономической оценки и прогнозирования характеристик*.

Приступая к созданию сложных технических проектов, заказчики и исполнители, прежде всего, должны *понять целесообразна ли их разработка* и оценить возможную эффективность применения готового продукта, оправдаются ли затраты на его производство и использование. Поэтому такие технические проекты традиционно начинаются с *анализа и экономического обоснования* предстоящего производства и применения предполагаемого продукта или системы. Заказчику и возможным потребителям результатов проекта необходимо оценивать реальную потребность в его создании и конкурентоспособность, а потенциальному разработчику предполагаемого продукта проводить оценку реализуемости проекта при условиях и ресурсах, предлагаемых заказчиком. Однако часто разработчики не в состоянии привести заказчику или руководителю проекта достаточно

обоснованные доказательства нереальности выполнения выдвигаемых требований к продукту при предложенных ограниченных значениях бюджета и сроков. Руководители конкретных проектов обычно не в состоянии достаточно обоснованно определять, сколько времени и затрат труда потребуется на каждый этап проекта системы, и не могут оценить, насколько успешно выполняется план производства. Это, как правило, означает, что проект системы с самого начала **выходит из-под экономического контроля** и возможна катастрофа с реализацией и завершением проекта всей системы в требуемый срок с заданным бюджетом и качеством.

Возможности и широта применения экономики при производстве **сложных комплексов программ** существенно зависят от современных **методологий и стилей** организации работы коллективов специалистов – разработчиков. Эти методологии различаются сферами применения, методами достижения высокого качества комплексов программ, психологическими характеристиками участвующих специалистов и организацией их деятельности в реальном времени. Различие целей и стилей при создании комплексов программ привели к формированию стратегий, позволяющих получать **сложные программные продукты двух классов**, характеризующиеся высоким качеством при применении:

- **свободного (открытого) программного обеспечения** (СПО), ориентированного на участие большого числа практически независимых специалистов, различных по квалификации, дислокации и ответственности за результаты своей добровольной деятельности;

- **закрытых текстов и технологических документов** программ, создаваемых профессиональными коллективами высокой квалификации под централизованным управлением для конкретных заказчиков определенных систем управления и обработки информации, производство, распространение и модификация которых регламентируется и контролируется заказчиком и поставщиком.

Первый класс сложных программных комплексов обычно первоначально создается группами энтузиастов в университетах, корпорациях или сообществах разработчиков и пользователей программ и распространяется бесплатно. Функции таких крупных комплексов программ ориентированы на решение новых, оригинальных задач массового использования и конкурентоспособных на рынке программных продуктов, куда со временем они поступают. Их разработ-

ка, изменение и совершенствование не регламентируется в реальном времени и реализуется по инициативе пользователей или разработчиков без определенных контролируемых планов и сроков. По мере безвозмездного совершенствования функций и качества программного продукта различными заинтересованными, неуправляемыми пользователями – разработчиками, расширяется сфера его применения, повышается качество, надежность и безопасность использования, что приводит к их активному проникновению на рынок, в бизнес и системы образования. Вследствие этого слабо документированный, непрерывно, спонтанно изменяющийся программный продукт и его производство трудно оценивать и сопровождать экономически. Высокие тиражи распространения, применения и гибкая спонтанная модификация программных средств возможны благодаря доступности для многих пользователей открытых исходных текстов программ на стандартном языке программирования, а также унифицированной технологии процессов производства компонентов и сопровождения таких комплексов программ.

Второй класс программных продуктов обычно имеет конкретного заказчика, относительно узкую сферу применения, предназначен для конкретных систем или пользователей, жестко регламентирован технологией производства, модификаций и документирования, что сближает их создание с обычным промышленным производством сложных технических изделий. Оно управляется планами и ограниченными сроками поставки готовых испытанных продуктов заказчику и пользователям, которые допускается эпизодически изменять только с санкции заказчика. Современные комплексы программ для систем управления и обработки информации **в реальном времени** активно применяются в сложных критических и ответственных системах динамического управления объектами, в высокоточном технологическом производстве, в авиации, в управлении космическими аппаратами, атомными электростанциями и оборонной техникой. Такие изделия являются одними из наиболее сложных интеллектуальных систем высокого качества, создаваемых человеком, для которых доступна и необходима сертификация не только производственных процессов, но и их результатов – программных продуктов. Далее в монографии внимание сосредоточено на этом классе программ.

Важной особенностью таких **промышленных программных продуктов** является их **отчуждаемость от первичных производи-**

телей. В отличие от программ для решения многих научных и инженерных задач, которые не оформляются документацией, делающей их познаваемыми и доступными для использования различными специалистами, эти комплексы программ имеют все основные особенности промышленных изделий и представляют собой оформленный программный продукт. После завершения их производства и испытаний они отчуждаются от коллектива их создателей и могут применяться, эксплуатироваться, эпизодически модернизироваться и корректироваться зачастую не теми специалистами, которые первоначально разработали. Это обуславливает необходимость производить такие комплексы программ с соблюдением определенных стандартов, правил структурного построения, а также оформлять на них достаточно полную технологическую документацию, позволяющую применять, сопровождать и развивать программные продукты различными квалифицированными специалистами под управлением заказчиков и/или поставщиков.

Применение сложных программных комплексов как *производственной продукции* существенно повысило *актуальность экономического обоснования, необходимость прогнозирования и измерения* процессов производства и их характеристик качества. Основной целью производства многих программных продуктов является повышение эффективности промышленных систем обработки информации и/или управления объектами и административными системами, в которых применяются сложные комплексы программ. Таковыми системами могут быть средства автоматизированного управления в авиации, системами вооружения, электростанциями и энергосистемами, системами административного управления, автоматизации проектирования и обучения.

В ряде случаев программные продукты невозможно или очень трудно характеризовать *непосредственной экономической эффективностью применения* в зависимости от затрат ресурсов, и целесообразно из анализа исключать характеристики эффективности программных продуктов и сопутствующие ей функциональные критерии качества. В результате исследование экономики создания комплексов программ приходится проводить по величине *затрат ресурсов на производство программного продукта* в предположении, что обеспечена реализация заданных функциональных характеристик их применения с требуемым качеством.

В жизненном цикле сложной промышленной продукции при анализе экономических характеристик традиционно выделяются три крупных процесса [6, 28, 37]:

- **проектирование продукта**, когда определяются его функции, архитектура, требования к качеству, техническим и экономическим характеристикам, анализируется и подготавливается решение о целесообразности его создания и технологии производства;
- **производство продукта**, при котором реализуется подготовленный проект, изготавливается, испытывается и поставляется заказчику и/или пользователям для применения готовый продукт, соответствующий требованиям и документации;
- **эксплуатация и применение продукта** пользователями или в системе, когда реализуются цели создания продукта в соответствии с требуемыми его функциями и документацией.

В монографии выделены и рассматриваются в основном **первые два процесса**. Проектирование сложных программных продуктов наиболее творческий процесс в их жизненном цикле, в котором участвуют относительно небольшое число высококвалифицированных специалистов, при этом затраты не велики, а экономические характеристики наиболее трудно прогнозируемы. Производство спроектированного сложного программного продукта обычно реализуется регламентированными производственными процессами большим коллективом специалистов, требует существенно больших затрат, которые можно и следует оценивать традиционными экономическими характеристиками.

При планировании производства сложных программных продуктов часто имеется определенный **заказчик-потребитель, который способен задать** основные цели, требуемые характеристики качества и обеспечить ресурсы для реализации проекта. Однако иногда инициатором разработки комплекса программ является разработчик-поставщик, который самостоятельно принимает все решения о его производстве за счет собственных ресурсов и предполагает возместить затраты путем реализации программного продукта на рынке. Таким образом, **при экономическом анализе и обосновании проектов сложных комплексов программ возможны два сценария:**

- создание и весь жизненный цикл комплексов программ ориентируется на массовое тиражирование и распространение их на рынке среди заранее не известных пользователей в различных сферах

и внешней среде применения; при этом **отсутствует конкретный внешний потребитель – заказчик**, который определяет и диктует основные требования к программному продукту, выделяет ресурсы и финансирует проект;

- производство программного продукта предполагается с определенным, относительно небольшим тиражом, с известной областью и внешней средой применения **для конкретного потребителя – заказчика**, который определяет требования к функциям, характеристикам качества, финансирует и выделяет ресурсы.

Эти сценарии **принципиально различаются методами экономического анализа** и обоснования их экономических характеристик. **Первый сценарий** базируется на маркетинговых исследованиях рынка программных продуктов и на стремлении поставщика занять на рынке достаточно выгодное место. Для этого ему необходимо определить наличие на рынке всей гаммы близких по назначению и функциям продуктов, оценить их эффективность, стоимость и применимость, а также **возможную конкурентоспособность** предполагаемого к разработке программного продукта для потенциальных пользователей и их возможное число. Следует оценивать рентабельность затрат на создание нового продукта, выявлять факторы, функциональные, экономические и конструктивные характеристики качества, которые способны привлечь достаточное число покупателей и оправдать затраты на предстоящую разработку. Для этого разработчикам необходимо проводить оценки возможной конкурентоспособности предполагаемой продукции на рынке **по величине соотношения:**

- возможной **эффективности** (ценности, достоинств) последующего применения программного продукта и способности удовлетворить потребности многих пользователей при его приобретении и использовании;

- **к стоимости** (цене, затратам), которую готовы заплатить пользователи при приобретении и эксплуатации данного программного продукта.

При выборе продукции **покупатель** обычно стремится максимизировать это соотношение как за счет поиска продуктов с наибольшими эффективностью и качеством, так и за счет минимальной стоимости покупаемого продукта.

В этом сценарии при организации производства вся ответственность за цели, функции и показатели качества продукта ложится на его руководителей, и особую роль при экономических оценках должны играть специалисты по маркетинговому анализу продукта, предполагаемого для распространения на рынке. Они должны оценивать конъюнктуру и риск успешного продвижения создаваемого продукта на рынок, сроки и график выполнения этапов жизненного цикла, достаточность ресурсов для реализации проекта и **экономическую эффективность** длительного развития, модификаций и распространения версий программного продукта. Для этого им нужны, в частности, оценки собственных затрат, экономических характеристик производства и модификации продукта.

Второй сценарий предполагает наличие определенного заказчика – потребителя конкретного программного продукта, который должен соответствовать его **формализованным и утвержденным техническим и экономическим требованиям**. Он выбирает конкурентоспособного поставщика – производителя продукта, которого оценивает на возможность реализовать проект с необходимым качеством с учетом ограничения требуемых бюджета, сроков и других ресурсов. При этом результаты разработки не обязательно подлежат широкому тиражированию, могут не поступать на рынок, маркетинговые исследования для таких проектов являются второстепенными и предварительно могут не проводиться.

Однако для заказчика и разработчиков при заключении контракта необходимо достаточно достоверное **прогнозирование требований** к программному продукту и **экономическое обоснование необходимых ресурсов** по трудоемкости, стоимости, срокам реализации и другим показателям. Заказчик заинтересован в получении программного продукта высокого качества при минимальных затратах, а разработчик желает получить максимальную оплату за созданный продукт и достаточные ресурсы на его производство. Противоположность интересов поставщика и потребителя при оценке экономических характеристик, стоимости и других ресурсов проекта требует **поиска компромисса**, при котором производитель программного продукта не продешевит, а заказчик не переплатит за конкретные выполненные работы и весь проект. Поэтому оба партнера **заинтересованы в достоверном экономическом прогнозировании** и обосновании проектирования и производства программного продукта. Для

этого должен быть подготовлен согласованный между заказчиком и разработчиком договор, в котором определены **цели и задачи проекта, требуемые характеристики продукта, доступные экономические и другие ресурсы для его реализации** [28, 37]. Эти данные должны быть предварительно сбалансированы и обеспечивать реализацию целей проекта при выделенных ресурсах с минимальным допустимым риском. Однако масштабы целей и функций сложных программных проектов имеют устойчивую тенденцию изменяться и увеличиваться, первоначально выделенные ресурсы могут не обеспечивать их реализацию. Большую часть рисков и негативных последствий производства можно избежать, используя существующие методы оценивания и прогнозирования производственных затрат, а также управления проектами программных продуктов для их успешного завершения.

В ряде случаев эффективность систем новой техники и программных продуктов в процессе производства приходится прогнозировать **в условиях неопределенности** целей, различных факторов и характеристик заказываемых систем и продуктов. Обычно бывают недостаточно известны перспективы внедрения и применения объектов производства – новых программных продуктов. Трудно формализуемыми и оцениваемыми являются размеры (масштабы) и структура систем, взаимодействие основных подсистем, цели, функции, качество и критерии оценки эффективности их функционирования. Значительные неопределенности содержатся также в экономических характеристиках производственных технологий, а также инструментальных средств автоматизации проектирования и изготовления комплексов программ. Положение усугубляется трудностью поэтапного определения качества компонентов и его прогнозирования в процессе разработки, что непосредственно отражается на экономических характеристиках реализации комплекса программ в целом.

Методы и достоверность **экономического анализа производства** сложных программных продуктов можно разделить на **две части**, существенно различающиеся особенностями производственных процессов, экономическими характеристиками и влияющими на них факторами. **В первой части жизненного цикла** производятся: системный анализ, проектирование, разработка, тестирование и испытания первой (базовой) версии программного продукта. Номенклатура работ, их трудоемкость, длительность и другие характеристики на этих

этапах производства существенно зависят от *свойств создаваемого продукта*, требуемых показателей качества, внешней и технологической среды разработки. При этом первостепенное значение имеют размеры, архитектура, сложность функций, количество, состав и взаимосвязи компонентов комплекса программ, которые являются базой для оценивания экономических характеристик производства конкретного продукта. Изучение подобных зависимостей для различных прототипов программных продуктов позволяет достаточно достоверно прогнозировать состав и основные экономические характеристики производства, планы и графики работ для вновь создаваемых продуктов.

Вторая часть жизненного цикла, отражающая применение, сопровождение и модификацию версий программного продукта, связана не только с экономическими характеристиками продукта и среды производства, они зависят *от величин и интенсивности изменений* версий, сложности и стоимости каждой модификации программного продукта. Версии сложных программных продуктов обычно характеризуются длительной непрерывной эксплуатацией, суммарная продолжительность которой может значительно превышать длительность производства первой версии. После того как программный продукт создан и испытан в ряде случаев он становится недоступным для разработчиков и применяется неизменным до внедрения очередной версии после модернизации системы. Жизненный цикл таких комплексов программ может составлять много лет, в течение которых необходимо обеспечивать возможность их изменения. *В процессе сопровождения* программы могут подвергаться эпизодическим корректировкам, которые должны регистрироваться, накапливаться и передаваться пользователям экземпляров программного продукта. Необходимо обеспечивать пользователям адекватность технологической документации каждой версии эксплуатируемого продукта в любой момент времени.

Номенклатура работ на этапах сопровождения более или менее стабильная, а их сложность, трудоемкость и длительность могут сильно варьироваться и зависят от распространения и применения конкретного программного продукта. Успех продукта у пользователей и на рынке, а также *экономику процесса развития версий трудно предсказывать*, так как она связана не только с техническими и экономическими параметрами производства версий комплекса про-

грамм. Определяющими становятся потребительские характеристики и качество продукта, а его экономические особенности с позиции производства отходят на второй план. Вследствие этого в широких пределах могут изменяться и трудно прогнозировать экономические характеристики, трудоемкость и необходимое число специалистов, поддерживающих этапы производства систематических модификаций версий программного продукта. Это затрудняет обобщение экономических характеристик производства изменений для различных версий продуктов и прогнозирование на их основе аналогичных характеристик новой модификации. В результате планирование трудоемкости, длительности и числа специалистов для этих этапов трудно обобщать, и приходится производить итерационно на *базе накопления опыта и анализа прецедентов развития экономических характеристик конкретных типов и версий программных продуктов*.

В монографии изложены *основные особенности экономики проектирования и производства сложных программных продуктов*, однако *остаются многие задачи и проблемы*, которые необходимо исследовать и решать для их развития, практического освоения и массового применения.

Можно выделить *три группы наиболее актуальных задач* для их эффективного внедрения в современную практику программной инженерии: анализ современной экономики программных продуктов; создание и организация экономически эффективного их проектирования и производства; подготовка и обучение специалистов для экономически эффективного проектирования и производства программных продуктов.

Задачи анализа современной экономики проектирования и производства программных продуктов:

- исследование методов оценивания и характеристики экономической деятельности предприятий при проектировании и производстве сложных программных продуктов (прибыль, доход, рентабельность, производительность труда) в различных отраслях промышленности;
- определение, анализ и классификация экономических характеристик – трудоемкости, стоимости, длительности производства, и реальных факторов, влияющих на экономику проектирования и производства программных продуктов;

- измерение, сбор и обобщение реальных экономических характеристик процессов проектирования и производства сложных программных продуктов на предприятиях и в отраслях промышленности;
- анализ экономики, планирования и достоверности прогнозирования экономических характеристик реальных проектов программных продуктов;
- разработка методов оценивания и выбора, квалифицированных коллективов специалистов-подрядчиков, способных создавать сложные программные продукты и базы данных требуемого качества в разумные сроки с учетом ограничений на используемые ресурсы.

Задачи создания и организации экономически эффективного проектирования и производства программных продуктов:

- создание эффективной технологии и экономики деятельности предприятия, обеспечивающей рациональное сочетание целей, стратегий и использования доступных ресурсов, необходимых для достижения целей производства программного продукта высокого качества;
- создание эффективных методов экономического и технического формирования требований к функциям, характеристикам и качеству программного продукта, с учетом экономических и ресурсных ограничений проектирования и производства на предприятии;
- создание методов экономического обоснования и прогнозирования разработки комплексной системы качества, планов и Программы испытаний, методологии и инструментальных средств, обеспечивающих требуемое качество, надежность и безопасность функционирования программного продукта;
- обеспечение экономических характеристик производства, предсказуемого и управляемого уровня достигаемого качества программного продукта, непосредственно зависящего от инструментария производства, систем качества и эффективности технологий, используемых на этапах жизненного цикла;
- обеспечение и удостоверение требуемых функций и качества готового программного продукта с полным комплектом адекватной эксплуатационной и технологической документации в процессе тестирования и испытаний.

Задачи подготовки и обучения специалистов для экономически эффективного проектирования и производства программных продуктов:

- необходимо обучение и подготовка руководителей и специалистов по экономике производства сложных программных продуктов, освоение современных методов экономического анализа, оценивания и прогнозирования процессов и необходимых ресурсов при производстве комплексов программ;

- для обеспечения экономически эффективного производства программных продуктов, необходима подготовка и обучение «команд» системных аналитиков, архитекторов и менеджеров проектов, специалистов по комплексированию, верификации, тестированию и сертификации сложных программных продуктов, по испытаниям и обеспечению их качества;

- следует выделять и обучать специалистов, ответственных за анализ, оценивание и прогнозирование экономических характеристик производства, за соблюдение экономически эффективной промышленной технологии создания и совершенствования программных продуктов;

- руководителям и специалистам необходимо освоить современный комплекс экономически эффективных технологических задач, методов и стандартов промышленного создания и развития сложных, тиражируемых программных комплексов и баз данных требуемого высокого качества;

- руководители и специалисты для обеспечения высоких экономических характеристик при производстве сложных комплексов программ должны освоить типовые, стандартные решения и использование апробированных заготовок программных компонентов, не требующих при их применении высококвалифицированного творческого труда.

Перечисленные задачи более подробно отражены в главе 2 и конкретизируются в последующих главах монографии. Они могут служить основой для развития и совершенствования исследований, методов и практики в этой сфере. Для развития **«экономики производства программных продуктов»** требуются непрерывные обобщения и освоения накопленного опыта и опубликованных данных в этой новой области экономики. Для квалифицированного прогнозирования и отслеживания экономики на различных этапах производства сложных комплексов программ **необходимо обучение специалистов новой экономической профессии** по специальной программе в составе общей квалификации **«программная инженерия»**. Такие

специалисты – экономисты должны входить в состав проектных коллективов или предприятий, создающих сложные программные продукты.

Монография состоит из трех логически связанных частей. Факторам, влияющим на *экономику производства* программных продуктов, в монографии посвящены главы от 3-й до 7-й, а *анализ и прогнозирование экономических характеристик* производства сложных комплексов программ отражены в главах от 8-й до 11-й. Для конкретизации используемых экономических понятий в первых двух главах изложены *основы экономики индустриального производства* различных сложных технических систем, а также основные особенности промышленного производства программных продуктов.

При изложении производственных процессов, которые являются основой для определения экономических характеристик программных продуктов, ниже активно используются международные стандарты их жизненного цикла, многие из которых адаптированы на русском языке (Приложение).

В начале большинства глав книги приводятся рисунки с кратким содержанием рассматриваемых задач и процессов, которые ориентируют читателей при выборе и освоении материала. Их полезно использовать для подготовки иллюстраций к *курсу лекций при обучении экономике проектирования и производства* конкретных сложных программных продуктов высокого качества.

Глава 1

ОСНОВЫ ЭКОНОМИКИ ИНДУСТРИАЛЬНОГО ПРОИЗВОДСТВА СЛОЖНЫХ ТЕХНИЧЕСКИХ СИСТЕМ И ПРОДУКТОВ

Общие принципы экономики и организации производства сложных систем и продуктов

Для применения *экономики программной инженерии* руководителям и специалистам в области производства сложных комплексов программ полезно освоить традиционные понятия, методы организации и оценки характеристик *экономики промышленного производства сложных технических систем и продуктов*. Задачи повышения экономической эффективности производства состоят в сокращении сроков создания и освоения новой техники, снижения затрат на ее производство, в достижение высокой экономичности применения новых типов продуктов, что имеет важное экономическое значение в промышленности. Удлинение сроков внедрения новых изделий в производство снижает эффективность вложенных средств, замораживает их, уменьшает возможный прирост национального дохода. Замедление процессов освоения новой сложной техники приводит к тому, что она может устаревать еще до того, как начинается ее эксплуатация, и не давать ожидаемого экономического эффекта. Сосредоточивая значительные трудовые, материальные и финансовые ресурсы, сфера производства воздействует на экономическую эффективность работы предприятий за счет лучшего использования этих ресурсов, обеспечения высоких темпов создания новых видов техники при минимальных затратах живого и овеществленного труда [28,37].

Производственные процессы на предприятиях, в результате которых создается промышленная продукция, необходимо соответствующим образом организовывать, обеспечивая их эффективное функционирование в целях выпуска конкретных видов продукции высокого качества и в количествах, удовлетворяющих потребности заказчика и/или пользователей. *Государственное регулирование эко-*

номики производства включает регламентирование доступных ресурсов, размеров производства, финансов и реализуется путем проведения соответствующей экономической политики. **Экономическая политика** – система мер, совокупность действий по управлению экономикой, включающая определенный набор процессов и компонентов, основными из которых являются отраслевая политика, анти-монопольная политика, организация денежной системы, удовлетворение спроса на услуги и товары коллективного и индивидуального пользования. Основы экономики индустриального производства сложных технических систем и продуктов, ниже излагаются для сокращения в конспективной, декларативной форме без широких обоснований и разъяснений, которые можно найти в специальной литературе.

Производственное предприятие – коллектив людей различной квалификации, связанных определенными социально-экономическими отношениями и интересами для создания определенных продуктов и извлечения прибыли, которые служат основой для удовлетворения потребностей коллектива. Оно характеризуется экономическим, производственно-техническим, организационным и социальным единством [27, 37]. Это единство определяется комплексом средств производства, обладающих технологической взаимосвязью отдельных этапов производственных процессов, в результате которых, используемые на предприятии объекты превращаются в готовую продукцию.

Цель управления производством – экономически рациональное использование и предупреждение потерь ресурсов путем сбалансированного распределения их по частным работам на протяжении всего жизненного цикла объекта с заданным качеством. Управление производством – это особый вид деятельности, включающий постановку задач, подготовку решений, планирование, организацию и стимулирование специалистов, контроль хода работ и использования ресурсов при создании сложных систем и продуктов. Задачи целевого управления такими работами – сводить воедино усилия исполнителей – специалистов разной квалификации, подрядчиков и субподрядчиков, добиваясь, чтобы они **выступали как команда**, а не как разрозненная группа функциональных специалистов при создании компонентов систем. В результате должны обеспечиваться концептуальная целостность системы и высокое качество решения главных задач при

сбалансированном использовании ресурсов на все функциональные задачи – рис.1.1.

Экономика индустриального производства сложных технических систем и продуктов включает:

- общие принципы экономики и организации производства сложных технических систем и продуктов:
 - цели и основы планирования производства продуктов – предметов производства и предметов потребления;
 - организацию управления производством систем и продуктов;
 - обеспечение технологии проектирования и подготовки производства систем и продуктов;
 - принципы и функциональные компоненты организации производства систем и продуктов;
 - управление качеством, эффективную организацию производства продуктов, минимизацию рисков;
- планирование и управление производством сложных технических систем и продуктов:
 - системный анализ – методологическую базу целевого планирования и этапов производства;
 - обеспечение экономически эффективного плана и этапов функционирования и развития предприятия;
 - входные и выходные данные процессов проектирования и производства продуктов;
 - контроль функционирования системы менеджмента качества и мониторинга процессов производства;
- стандартизацию производства и качества сложных технических систем и продуктов:
 - применение стандартов системы менеджмента (административного управления) качеством;
 - создание и контроль применения системы менеджмента качества производства предприятия;
 - формирование общесистемных процедур для управления документами производства;
- бизнес-план, организацию и анализ предприятия для производства сложных технических систем и продуктов:
 - разработка общей концепции развития предприятия;
 - планирование, которое дает возможность контролировать деятельность предприятия;
 - экономический анализ деятельности промышленного предприятия.

Рис. 1.1

Любой **процесс производства** включает **три основных, тесно взаимодействующих между собой компонентов**: собственно производственный процесс, предметы – результаты производства и средства производства. С организационных позиций процесс труда представляет собой соединение экономических и производственных факторов в единую систему, называемую производственным процессом. Рациональная организация предполагает такой способ соединения всех производственных компонентов в единую взаимосвязанную систему, при которой будет использовано наименьшее количество трудовых ресурсов, предметов труда и средств производства.

Организационное единство предприятия определяется наличием целостного руководства и коллектива, что находит свое отражение в общей и организационной структуре предприятия. Экономическое единство определяется общностью экономических результатов работы – объемом созданной и реализованной продукции, уровнем рентабельности, массой прибыли, фондами предприятия. Извлечение прибыли обычно выступает в качестве основной цели деятельности предприятия. По назначению готовой продукции предприятия делятся на две большие группы: производящие **средства производства** и производящие **предметы потребления** или продукты непосредственного применения. Для классификации однородных предприятий имеет важное значение унификация – применение типовых стандартов, конструктивных и технологических решений, производственной структуры, типовой документации. Под **общей структурой предприятия** понимается комплекс производственных подразделений, организаций по управлению предприятием и обслуживанию работников, их количество, квалификация, взаимосвязи и соотношения между ними по размеру занятых площадей и оборудования, численности работников и пропускной способности при производстве.

Одной из основных задач при формировании производственной структуры является **обеспечение скоординированного функционирования** всех составляющих производственного процесса: подготовительных операций, основных производственных процессов, технического обслуживания. Необходимо обосновывать и применять экономически наиболее рациональные для конкретных производственно-технических условий, организационные формы и методы осуществления процессов производства. Важный элемент производственных процессов – организация труда квалифицированных специали-

стов, конкретно реализующая соединение рабочей силы со средствами производства. Методы организации труда в значительной мере определяются целями, формами и предметами производственного процесса.

Производственная структура предприятия обычно определяется формой организации производственного процесса, подразделений и групп специалистов по принципу: **технологическому** – формируется по признаку однородности технологического процесса изготовления различных компонентов или изделий; или **предметному** – объединяются рабочие места, участки по выпуску определенного вида и компонентов продукции [28]. Важно **обеспечивать экономически рациональное разделение труда**, и определение на этой основе профессионального и квалификационного состава специалистов, научная организация и оптимальное обслуживание рабочих мест, улучшение и оздоровление условий труда.

Организационная структура **управления предприятием** – упорядоченная совокупность служб, управляющих его деятельностью, взаимосвязями и подчинением. Всей деятельностью предприятия руководит управляющий, менеджер, который может быть как собственником имущества, так и наемным работником. Для обеспечения стратегического, текущего и оперативного руководства предприятием у руководителя должен иметься непосредственно подчиненный ему функциональный аппарат управления. Аппарат управления производством обычно включает следующие **основные службы**:

- оперативного руководства предприятием;
- управления персоналом (социальная служба);
- экономической и финансовой деятельности;
- маркетинга;
- внешних экономических и производственных связей;
- технического и экономического развития.

Структура аппарата управления зависит от многих факторов: типа производства, специализации, объема производства, конструктивной сложности изготавливаемой продукции. В системе управления можно выделить **два основных направления деятельности**. Первое направление – обеспечение **технологической оснащенности**, ориентированное на управление инвестициями и технологией. Его задачей является создание условий экономически эффективного использования технологий, оборудования, производственных ресур-

сов. Второе направление – **организация совместной деятельности персонала**, включая распределение заданий по конкретным исполнителям и срокам, контроль исполнения, устранение недочетов и сбоев в работе специалистов, подразделений и предприятия.

Цель технологии проектирования и подготовки производства – разработка проекта экономически эффективного технологического процесса изготовления продукта и реализация этого проекта в конкретных условиях предприятия. Технология производства – это методы, технические, инструментальные средства и система взаимосвязанных способов изготовления продукции или выполнения установленного вида работ. Технология должна включать весь перечень последовательных операций по превращению исходного замысла и требований потребителя в готовый продукт с указанием методов, типа и характеристик инструментов процессов, которыми специалисты пользуются на каждом этапе производства.

В определенных условиях может возникать необходимость проведения в жизнь целевых мероприятий по совершенствованию организации производства. В качестве таких условий могут быть: реконструкция и техническое перевооружение производства; освоение нового вида продукции; необходимость ликвидации узких мест; задачи повышения эффективности производства и качества продукции. Реализация мероприятий по улучшению организации в этом случае может быть осуществлена на основе **целевой комплексной Программы совершенствования организации производства**. Такая Программа представляет собой документ, в котором отражена планируемая совокупность работ, направленных на разработку и осуществление рациональных форм и методов организации производства в современных условиях. Она должна содержать комплекс экономических, организационных и методологических мероприятий, обеспечивающих ее реализацию в установленные сроки и в пределах выделенных ресурсов.

Организация производства – система мер, направленных на рационализацию сочетания в пространстве и времени вещественных компонентов и специалистов, занятых в процессе производства. Под организацией производственного процесса понимаются методы подбора и сочетания его компонентов в пространстве и времени с целью достижения эффективного конечного результата. Организация производственных процессов предполагает **сочетание их компонентов**

во времени, что обуславливает определенный порядок выполнения отдельных операций, рациональное совмещение времени выполнения различных видов работ, определение календарных и плановых нормативов труда (см. рис.1.1).

Принципы организации производственного процесса представляют собой исходные положения, на основе которых осуществляются построение, функционирование и развитие экономически эффективных производственных процессов. Дифференциация предполагает разделение производственного процесса на отдельные части (процессы, операции) и их закрепление за соответствующими подразделениями и специалистами предприятия. В практической деятельности по организации производства при использовании принципов дифференциации приоритет должен отдаваться тому методу, который обеспечит *наилучшие экономические и социальные характеристики производственного процесса*.

Функциональные компоненты **организации производства** должны решать задачи:

- организации комплексной подготовки производства и освоения новой продукции;
- организации производственных процессов по выпуску продукции;
- организации производственной инфраструктуры предприятия;
- организации работ по обеспечению качества продукции;
- организации материального обеспечения процессов производства;
- организации маркетинговых исследований, сбыта и реализации продукции.

Технологические задачи организации производства могут быть сгруппированы по компонентам производственного процесса [28]:

- организация участников производственного процесса, которые обеспечивают решение задач по подготовке и повышению квалификации кадров, внедрению научной организации труда специалистов, рационализации трудовых процессов, нормированию труда и организации материального и морального стимулирования работающих;
- организация использования средств производства, которая включает задачи по формированию оптимальной структуры оборудования, его обновлению, улучшению загрузки оборудования и исполь-

зованию его технических возможностей, повышению эффективности его технического обслуживания;

- организация движения компонентов производства, которая направлена на решение задач по обеспечению бесперебойного движения предметов – результатов труда на основе совершенствованию технологических процессов, маршрутов и планированных решений, разработке и внедрении обоснованных заделов;

- организация производственных потоков в производстве, которая решает задачи разработки и реализации информационной модели предприятия и его подразделений; отбора информации, необходимой для управления и соответствующих подсистем; передачи информации всем подразделениям предприятия.

Принцип *специализации* основан на ограничении разнообразия компонентов производственного процесса. Реализация этого принципа предполагает закрепление за каждым специалистом и/или подразделением ограниченной номенклатуры работ, операций или компонентов. Увеличение номенклатуры производства снижает уровень специализации. Высокая степень специализации подразделений и рабочих мест способствует росту производительности труда за счет выработки трудовых навыков специалистов, возможностей технического оснащения труда. Вместе с тем узкая специализация снижает требуемую квалификацию специалистов, увеличивает монотонность труда, ведет к утомляемости, ограничивает инициативу.

Рациональная организация производственных систем может быть построена на основе совершенных подсистем организации трудовых процессов, потребляющих на свое осуществление *наименьшее количество экономических ресурсов* на продукцию. Важнейшим признаком совершенной организации трудовых и производственных процессов в условиях рыночных экономических отношений может служить устойчивое платежеспособное положение предприятий, обеспечивающее не только производство продукции, но и развитие трудового и производственного потенциала.

Комплексная система обеспечения экономически эффективной организации производства должна включать следующие основные компоненты управления:

- разделение и кооперация труда, правильный выбор форм и видов, которые создают экономические основы для специализации и роста профессиональной квалификации специалистов;

- проектирование производственных процессов, которое должно строиться на принципах экономии рабочего времени и затрат труда специалистов;
- освоение трудовых процессов и норм, предусматривающее достижение равновесия фактических и проектных затрат времени на выполнение определенных работ;
- экономическая оценка затрат и результатов труда, заключающаяся в достижении максимальных результатов при заданных затратах, или минимальных затрат при заданных результатах;
- мотивация и стимулирование производительности специалистов, которые предусматривают удовлетворение личного интереса каждого работника в высокой оплате;
- разработка системы управления производственными процессами, предусматривающей комплексное воздействие различных факторов на конечные результаты производства.

Для регулирования экономики используются как *экономические*, так и *административные методы*. Существуют области, где применение административных методов эффективно и не противоречит рыночному механизму. С производством продукции, осуществлением любых видов производственной деятельности связан *производственный риск*. В экономике производства различают *глобальный* (государственный, отраслевой) и *локальный* (на уровне предприятия, проекта) риски. Отдельные решения, принятые на уровне предприятия, особенно мероприятия по ассортименту и объему производства, внедрению технологических процессов, выполнению социальных программ, могут входить в противоречие с общегосударственными интересами и сопровождаться глобальными рисками. Среди причин возникновения такого риска: уменьшение размеров производства, несоответствие качества продукции спросу, рост материальных затрат, уплата повышенных налогов. Коммерческий риск предприятия возникает в процессе реализации продуктов, произведенных или закупленных предприятием, снижения объема реализации вследствие негативного изменения конъюнктуры или управленческих ошибок, непредвиденного снижения объема закупок и роста издержек.

При совершенствовании организации процессов производства все компоненты необходимо применять последовательно и комплексно на каждом рабочем месте, на всех производственных участ-

ках и предприятиях различных форм собственности. В условиях перехода к рыночным отношениям иногда считали, что проблема **обоснования трудовых нормативов и норм** потеряла приоритетное и организующее значение, которое она имела в централизованной плановой экономике. Большой ущерб практике нанес неоправданный тезис о том, что эффективную рыночную экономику и прогрессивную организацию труда и производства можно создать без трудовых нормативов и норм. Между тем в мире нет ни одной страны с рыночной экономикой, в которой бы не применялись экономические нормы труда, причем достаточно жесткие, весьма точные и научно обоснованные [37].

Планирование и управление производством сложных технических систем и продуктов

Планирование деятельности предприятий является **экономической основой** отношений всех хозяйственных субъектов и экономических объектов с различными формами собственности. Посредством планирования обеспечивается необходимое равновесие между производством и потреблением продукции, величиной рыночного спроса на товары и объемом их предложения предприятием на макроэкономическом уровне. На уровне предприятия планирование призвано обеспечить выпуск высококачественной продукции в необходимых количествах и номенклатуре на основе эффективного и рационального использования ресурсов, а также координацию деятельности отдельных структурных подразделений. Критическим параметром планирования и управления производством обычно является **время**.

Методологической базой целевого планирования производства является системный анализ, который предполагает:

- обследование объектов и среды производства, для предварительной формализации целей, назначения и задач продукта;
- исследование и сопоставление альтернативных действий, которые должны приводить к достижению поставленных целей проектирования и производства;
- сравнение альтернатив по величине достигаемого эффекта проекта в зависимости от затрат на его достижение (желательно, по показателю «эффективность/стоимость»);

- учет и анализ влияния неопределенностей на эффект производства характеристик альтернатив, определяющих их выбор.

Целевое управление производством позволяет планировать, контролировать и анализировать информацию о состоянии и тенденциях изменения объекта разработки, его качества и затраченных ресурсов. При этом должны **сохраняться основные стратегические цели производства** системы или продукта и процессы их достижения. Для этого должны анализироваться требования заказчика и/или пользователей, включающие предлагаемые изменения, для обеспечения уверенности в том, что:

- требования заказчика к продукции четко определены;
- требования контракта или заказа, отличающиеся от тех, которые были высказаны ранее, рассмотрены;
- предприятие располагает возможностями для удовлетворения изменений требований заказчика к продукции.

Важной целью планирования производства является обеспечение **экономически эффективного функционирования и развития предприятия** путем решения следующих задач:

- предвидение вероятных рыночных тенденций и соответствующая им корректировка производственной программы предприятия;
- исследование требований потребителей и формирование программы, ориентированной на их запросы;
- обеспечение выпуска продукции более высокого качества;
- непрерывное повышение эффективности производства на основе дальнейшей специализации и кооперирования;
- применение наиболее экономичных технологий и оборудования;
- согласование действий с поставщиками, потребителями, посредниками предприятия и направленность этих действий на достижение взаимовыгодных результатов.

Планирование деятельности предприятия целесообразно осуществлять в соответствии со следующими принципами [6, 27]:

- **конкретность** – предприятие в целом и каждое его подразделение должно иметь четкие ориентиры для своей производственной деятельности, это позволяет контролировать ход выполнения планов и судить о том, в какой степени решены поставленные задачи;
- **ориентированность во времени** – каждый план имеет свои

строго определенные границы во времени, отсутствие временных границ не позволяет предприятию контролировать сроки выполнения своих планов;

- **гибкость** – предполагает разработку альтернативных планов, и гибкое реагирование на постоянно меняющиеся условия реальной экономической ситуации;

- **комплексность** – планы должны отражать все стороны деятельности предприятия: объем планируемых работ и сроки их выполнения, выделение необходимых ресурсов;

- **обязательность исполнения** – принятый к выполнению план предприятия (подразделения) становится обязательным для выполнения сотрудниками, при изменении обстоятельств, возможно уточнять или корректировать план.

План производства должен отражать **экономически рациональное сочетание целей, стратегий действий, конкретных процедур, доступных ресурсов и других компонентов**, необходимых для достижения поставленной **основной цели производства продукта с заданным качеством**. Планирование производства должно обеспечивать компромисс между требующимися характеристиками создаваемой системы или продукта и ограниченными ресурсами, необходимыми на их разработку и применение. По мере уточнения исходных данных об объекте разработки, внешней среде применения и ресурсах, в процессе системного анализа и проектирования возрастает достоверность планирования, которая обычно проходит **следующие этапы**:

- первичного прогнозирования возможных характеристик производства продукта на базе обобщения данных подобных прототипов ранее реализованных проектов и создание концепции проекта;

- подготовки предварительного рабочего плана выполнения этапов и частных работ с учетом допустимых затрат ресурсов на их реализацию в процессе жизненного цикла проекта;

- управления детализацией и реализацией плана проекта, с его оперативной корректировкой и перераспределением ресурсов в соответствии с особенностями развития частей жизненного цикла проекта;

- обобщения и накопления результатов планирования и управления конкретным производством продукта для использования этих

данных в качестве прототипов при разработке и совершенствовании планов последующих проектов.

Внутреннее взаимодействие между различными группами специалистов, вовлеченными в производство, должно управляться таким образом, чтобы обеспечить эффективную координацию и четкую ответственность исполнителей.

Входные данные для производства системы или продукта должны включать:

- функциональные и эксплуатационные требования к продукции для заказчика или рынка;
- применяемые нормативные и законодательные требования к производству продукции;
- применяемые требования по охране окружающей среды;
- требования, вытекающие из прежних аналогичных продуктов.

Эти входные данные должны быть проанализированы на адекватность и полноту; нечеткость или противоречивость требований должны быть устранены.

Выходные данные процесса проектирования и производства должны быть зарегистрированы в форме, дающей возможность проверки их по отношению к реализации входных требований:

- соответствовать входным требованиям для проектирования и производства;
- содержать или давать ссылку на критерии функций и качества, достаточных для испытаний, поставки продукции и применения;
- определять характеристики продукции, которые являются существенными с точки зрения безопасности и корректного использования.

Утверждение проекта и плана производства продукции должно проводиться с целью подтверждения того, что конечная продукция способна отвечать всем требованиям для конкретных условий использования заказчиком или пользователями. Когда это возможно, утверждение должно быть спланировано и выполнено до начала поставки или применения продукции. Там, где невозможно осуществить полное утверждение до начала поставки или применения, должно быть предпринято частичное утверждение выходных данных результатов производства в максимально возможном размере с точки зрения ограниченной практики.

Изменения производства или модификация продукции должны быть утверждены в планах уполномоченным персоналом заказчика и зарегистрированы до их внедрения, при этом следует определить влияние изменений на:

- взаимодействие между компонентами и этапами производства;
- взаимодействие между составными компонентами конечной продукции;
- имеющуюся продукцию и на функционирование, ранее поставленной продукции;
- необходимость проведения повторных испытаний или утверждения для всех или части выходных данных производства продукции.

Предприятие должно **спланировать экономически эффективные производственные и сервисные операции**, включая те, которые предпринимаются после первоначальной поставки, посредством:

- предоставления технических условий, определяющих характеристики продукции, которые должны быть достигнуты;
- предоставления четко понимаемых производственных требований или инструкций для тех видов деятельности, где они необходимы для достижения соответствия требованиям качества продукции;
- внедрения надлежащих действий по мониторингу или проверке состояния продукции;
- подходящих методов для выпуска, поставки и/или монтажа продукции.

Меры по утверждению производственных процессов должны быть определены и направлены на: аттестацию процессов до их использования; аттестацию оборудования и/или персонала. Предприятие должно определить, спланировать и внедрить процессы измерений, мониторинга, анализа и улучшения производства для обеспечения уверенности в том, что система менеджмента качества, процессы и продукция соответствуют установленным требованиям. Эффективность применяемых измерений должна периодически оцениваться.

Предприятие должно определить, спланировать и установить **процессы контроля функционирования системы менеджмента качества** (см. рис. 1.1). Удовлетворенность заказчика системой качества производства должна использоваться как способ оценки текущего состояния системы. Предприятие должно применять подходящие

методы измерения и мониторинга процессов качества производства, необходимые для удовлетворения требований заказчика и для демонстрации постоянной способности процессов удовлетворять поставленным требованиям к качеству продукции. Отчеты должны фиксировать и утверждать уполномоченные лица, ответственные за выпуск продукции.

Руководство предприятия должно обеспечить уверенность в том, что продукция, которая не соответствует требованиям, находится под контролем, обеспечивающим предотвращение ее непреднамеренного использования или поставки. Продукция, имеющая несоответствия, должна быть подвергнута коррекции или исправлению с целью обеспечения соответствия требованиям, или удалена (отбракована) как неприемлемая. Должны быть определены ответственность и полномочия по проведению анализа и принятию решений по несоответствиям продукции.

Для **совершенствования** производства должны быть установлены процедуры, направленные на определение эффективности системы менеджмента качества и выявление мест, где могут быть сделаны улучшения. Следует планировать и собирать данные, появляющиеся в результате действий по измерению и мониторингу, эффективности и адекватности системы менеджмента качества. Целесообразно **постоянно улучшать систему менеджмента качества производства**, установить общесистемные процедуры, которые определяют использование политики качества, результатов внутреннего аудита, анализа данных, корректирующих и предупреждающих действий со стороны руководства в целях поддержки постоянных улучшений производства.

На каждом этапе производства должен **проводиться поиск** эффективных экономических и технических решений реализации продукта. В результате процессы планирования и производства развиваются параллельно. Реализация проекта производства системы или продукта зависит от результатов выполнения частных работ и может требовать оперативной корректировки плана. При этом определяющими являются **организация, стимулирование и контроль развития проекта**. Для этого необходимо следить за ходом исполнения проекта на всем протяжении его жизненного цикла и сравнивать запланированные и фактические результаты работ. Контроль является органической функцией управления и должен иметь средства регули-

рования поведения отдельных специалистов и коллектива производства в целом. Одновременно следует обеспечивать наблюдение за состоянием системы или продукта и их характеристиками качества, что позволяет устанавливать частные компромиссы с используемыми ресурсами. **Объектами управления и контроля при этом являются:**

- функциональные и технические характеристики реализованных компонентов продукта, показатели качества процессов производства и результатов выполнения отдельных работ;
- затраты ресурсов на выполнение частных работ и реализацию компонентов продукта (трудоемкость, стоимость, время, материальные ресурсы);
- графики работ, степень их выполнения, наличие и причины отклонений реализации частных работ от планов, угроза нарушения сроков контракта.

Для интеграции усилий специалистов и эффективного использования ресурсов производства **должен выделяться руководитель – лидер, управляющий производством**. Все ресурсы и исходные данные, необходимые для эффективного выполнения проекта, руководитель получает от заказчика, функциональных подразделений и специалистов производства. Задача управляющего производством наряду с прямыми воздействиями на подчиненных и координацией их работ – стимулировать и контролировать активность прямых связей между исполнителями частных работ.

Регулярный обмен информацией между специалистами о процессах производства позволяет осуществлять:

- сбор исходных данных о состоянии, достигнутом качестве компонентов и использованных ресурсах;
- диспетчерское управление ресурсами и частными исполнителями работ;
- сравнение текущих результатов частных работ с техническими заданиями, спецификациями и планом производства;
- корректировку технических результатов работ, сроков и используемых ресурсов в соответствии с изменением требований в процессе развития проекта и производства.

Стандартизация производства и качества сложных технических систем и продуктов

Стандарты ISO 9000:2000 определяют процессный подход в административном управлении производством и системами качества предприятий, а также регламентируют способы выявления и реализации возможностей для их улучшения. По отношению ко всем процессам, необходимым для создания необходимой продукции, применяется управление процессами и проводится проверка «выходов». Измерение степени удовлетворенности заказчика и других заинтересованных сторон используется в качестве обратной связи для оценки и признания того, что требования заказчика были выполнены полностью (см. рис. 1.1).

Серия стандартов ISO 9000 разработана, чтобы помочь предприятиям всех типов и размеров внедрять и использовать эффективные *системы менеджмента (административного управления) качества*. Совместно они образуют комплект согласованных стандартов управления производством и системами качества:

- **ISO 9000** – представляет введение в системы управления качеством продукции и услуг и словарь качества;
- **ISO 9001** – устанавливает детальные требования для систем управления качеством, достаточные в случае необходимости продемонстрировать способность предприятия, обеспечить соответствие качества продукции требованиям заказчика;
- **ISO 9004** – содержит руководство, по внедрению и применению широко развитой системы управления качеством, чтобы достичь постоянного улучшения деловой деятельности и результатов предприятия.

В **стандарте ISO 9004** детализированы руководящие указания и рекомендации по применению системы менеджмента качества, которые изложены в том же порядке, как требования в **ISO 9001**. Оба стандарта ссылаются на **ISO 9000**, который объясняет используемую терминологию и определения. Структура основных требований и рекомендаций в этих стандартах сведена к четырем объединенным крупным процессам:

- **обязанности и ответственность администрации управления производством и качеством**: ориентация на потребителя; требования заказчика к качеству; политика обеспечения качества; це-

ли и планирование качества; система административного управления качеством; анализ системы качества со стороны руководства;

- **административное управление ресурсами**: отбор и подготовка персонала по квалификации и компетентности; инфраструктура, производственная среда и инструментальные средства для обеспечения производства и качества продукции;

- **процессы производства продукции и управления ее качеством**: планирование процессов производства; организация взаимодействия с заказчиком; проектирование и разработка; закупка; операции производства и обслуживания; контроль и измерение качества; обслуживание продукции после поставки;

- **измерения, анализ и совершенствование производства**: мониторинг корректировок; учет изменений требований заказчика; анализ и измерение характеристик продукта и процессов; процессы улучшения и корректирующие действия при производстве.

Стандарты серии **ISO 9000** рекомендуется применять в деятельности предприятия, начиная от идентификации требований заказчика, и охватывать все процессы системы управления качеством, вплоть до достижения соответствия требованиям. Применение сокращенного, адаптированного варианта требований не освобождает предприятие от ответственности предоставлять продукцию, которая удовлетворяет всем требованиям заказчика. Система качества производства должна быть внедрена, поддерживаться в рабочем состоянии и подвергаться улучшениям со стороны специалистов предприятия. Масштаб и глубина процедур должна определяться такими факторами как размер и тип предприятия, сложность и взаимосвязь процессов, применяемые методы, а также квалификация и степень подготовки персонала, участвующего в выполнении работ. Они **должны включать**:

- общесистемные процедуры, которые описывают деятельность, необходимую для внедрения и применения производства и системы качества;

- процедуры, описывающие последовательность и внутреннее содержание процессов, необходимых для обеспечения уверенности в соответствии продукции установленным требованиям;

- инструкции, описывающие операционную деятельность и управление процессами.

Предприятие должно установить и управлять процессами, необходимыми для обеспечения уверенности в том, что продукция соот-

ветствует требованиям заказчика. В качестве способа внедрения и демонстрации, установленных процессов, **предприятие должно создать систему менеджмента качества**, основываясь на требованиях международных стандартов. Система менеджмента должна быть внедрена, поддерживаться в рабочем состоянии и подвергаться улучшениям со стороны предприятия. Масштаб и глубина процедур должна определяться такими факторами как размер и тип предприятия, сложность и взаимосвязь процессов, применяемые методы, а также квалификация и степень подготовки персонала, участвующего в выполнении работ.

Высшее руководство предприятия должно обеспечивать и быть уверено, что требования заказчика полностью поняты и могут быть удовлетворены. Оно должно разработать **политику предприятия в области качества** и обеспечить уверенность в том, что она [37]:

- соответствует потребностям предприятия и заказчиков;
- включает обязательства по удовлетворению требований и постоянному улучшению;
- обеспечивает основу для разработки и анализа целей в области качества;
- распространена, понятна и внедрена в предприятии;
- анализируется с целью постоянного поддержания пригодности ее применения.

Предприятие должно создать **систему менеджмента качества** как средство реализации его политики в области качества и обеспечения уверенности в том, что продукция отвечает требованиям заказчика. Роли сотрудников и их взаимосвязи, а также ответственность и полномочия персонала должны быть установлены для того, чтобы способствовать эффективному управлению качеством, и должны быть доведены до соответствующих уровней организации. Высшее руководство должно уполномочить одного (или нескольких) лиц для обеспечения уверенности в том, что система менеджмента качества внедрена и поддерживается в рабочем состоянии в соответствии с требованиями международных стандартов.

Предприятие должно установить общесистемные процедуры для **управления документами**, необходимыми для функционирования системы менеджмента качества, обеспечивающей уверенность в том, что:

- документы проверены на адекватность до их применения;
- документы анализируются, при необходимости уточняются и снова утверждаются;
- соответствующие выпуски документов находятся в тех местах, где осуществляется деятельность, имеющая существенное значение для эффективности функционирования системы менеджмента качества;
- устаревшие документы изъяты из всех мест их рассылки и применения или предприняты другие методы управления, предотвращающие их непреднамеренное использование;
- любые устаревшие документы, оставленные для юридических целей или в целях сохранения знаний, должным образом идентифицированы.

Должен быть составлен специальный перечень или применяться другая эквивалентная процедура управления, идентифицирующая статус текущей версии документов в целях предотвращения использования недействительных и/или устаревших документов.

Управление ресурсами необходимо для создания и поддержания в рабочем состоянии системы менеджмента качества. Организация должна проводить назначение специалистов с целью обеспечения уверенности, что те, кто имеет обязанности, определенные системой менеджмента качества, являются компетентными для осуществления своей деятельности на основе соответствующего образования, подготовки, мастерства и опыта, создать и поддерживать в рабочем состоянии **общесистемные процедуры** по:

- определению потребностей в компетентном персонале и в подготовке специалистов;
- обеспечению подготовки специалистов в соответствии с выявленными потребностями;
- оценке эффективности процессов подготовки кадров;
- ведению соответствующих отчетов об образовании кадров, их подготовке, уровне мастерства и опыта.

Процессы, необходимые для **выпуска требуемой продукции**, их последовательность и взаимосвязи должны быть определены, спланированы и внедрены. При определении таких процессов организация должна учесть результаты планирования качества. Должна быть уверенность в том, что их результаты соответствуют требованиям заказчика.

Предприятие должно создать **процесс идентификации требований заказчика** – полноту требований заказчика к продукции, выявлять требования, не установленные заказчиком, но необходимые для целей сертификации. Организационная структура системы менеджмента качества должна базироваться на сложившейся структуре управления процессами производства, учитывать имеющиеся технологические связи, традиции и опыт коллектива. Руководителем системы качества должен быть руководитель производства предприятия. Он формирует политику в области качества, которая определяет стратегические цели, принципиальные направления деятельности и всю идеологию документов системы менеджмента качества. При руководителе системы качества целесообразно создавать совет по качеству – консультационный орган, основной задачей которого является анализ эффективности работы производства и системы качества предприятия.

Бизнес-план, организация и анализ предприятия для производства сложных технических систем и продуктов

Предприятие для производства продукции обычно формируется на **основе бизнес-плана**. Оно является субъектом предпринимательской деятельности, которое осуществляет самостоятельное производство продукта, для систематического извлечения прибыли от выполнения работ и продажи продукта, а также зарегистрировано в установленном законом порядке [6, 28]. Бизнес-план представляет собой документ, в котором описаны основные аспекты будущего предприятия и производства, проанализированы все риски, с которыми оно может столкнуться, и определены способы решения проблем. Главное в бизнес-плане – это обоснование экономически эффективного проекта производства технической системы или продукта. **Бизнес-планирование** должно выполнять следующие функции:

- разработку общей концепции развития предприятия;
- планирование, которое дает возможность оценивать и контролировать развитие основной производственной деятельности предприятия;
- привлечение финансовых средств (ссуды, кредиты) со стороны.

Структура бизнес-плана зависит от тех задач, для решения кото-

рых он предназначен. В теории и практике экономики производства нет жестко регламентированной структуры бизнес-плана: она может быть различной в зависимости от выполняемой функции и типа продукции. Составлять бизнес-план должен руководитель предприятия с привлечением сотрудников и независимых экспертов. Бизнес-план может **состоять из следующих разделов:**

- производственные возможности предприятия (резюме);
- виды продуктов, предполагаемых для производства;
- возможные рынки сбыта продуктов;
- состояние конкуренции на рынках сбыта конкретного продукта;
- план маркетинга продукта;
- план производства продукта;
- финансовый план производства продукта;
- организационный план производства продукта;
- правовое обеспечение деятельности предприятия;
- оценка риска и страхование производства и сбыта продукции.

В разделе возможности предприятия определяются в приоритетном порядке основные направления деятельности для стратегического планирования производства. По каждому направлению деятельности должны устанавливаться цели, к которым стремится предприятие, и стратегии их достижения, включающие перечень необходимых мероприятий. По каждой задаче стратегии должны определяться ответственные лица.

Бизнес-план может разрабатываться, например, по следующей форме [27, 28]:

Титульный раздел содержит наименование документа, название проекта, профиль предприятия (или ситуацию, для которой разрабатывается бизнес-план), фамилии руководителей, реквизиты предприятия, указание на время и место разработки бизнес-плана.

Резюме отражает содержание проекта:

- суть и экономическая эффективность проекта и продукта;
- общие сведения о предприятии;
- команда управления производством;
- план действий;
- источники финансирования;
- план и гарантии возврата инвестиций.

Описание предприятия характеризует: текущее состояние, ор-

ганизационную структуру, достижения предприятия, место на рынке, основных клиентов и партнеров.

Клиентура определяет, для каких заказчиков и пользователей предприятие выпускает свою продукцию. Устанавливаются критерии и потребительские сегменты предприятия и условия приобретения продукции предприятия.

Описание основной продукции и параметры товарной политики предприятия. При большом ассортименте продукция разбивается на товарные группы.

План маркетинга и выхода продукции на рынок, основные параметры в области ценовой политики и продвижения продукции:

- значимость (масштабность) проекта;
- требования к потребительским свойствам продукта;
- конкурентоспособность продукции;
- рынок сбыта;
- цена продукции;
- стратегия продвижения на рынок.

Производственный план: производственные возможности, технологическое обеспечение и возможные последствия производственной деятельности.

Организационный план: кадровое и правовое обеспечение проекта, наличие государственной поддержки и льгот, порядок реализации проекта:

- команда управления и ведущие специалисты;
- кадровое обеспечение;
- правовое обеспечение;
- партнеры по реализации проекта;
- государственная поддержка и льготы;
- организационная структура реализации проекта;
- график реализации проекта;
- характеристика активов.

Финансовый план и основные финансовые параметры деятельности предприятия:

- нормативы для финансово-экономических расчетов (платежи, налоги, инфляция, издержки);
- расходы на персонал;
- прямые расходы на производство;
- постоянные расходы на производство;

- калькуляция себестоимости продукции;
- смета затрат на производство продукта;
- отчет о возможных прибылях и убытках;
- прогнозный баланс;
- показатели эффективности реализации проекта (срок окупаемости, прибыльность, рентабельность).

Риски и гарантии связанные с реализацией проекта: риски предприятия и меры их предупреждения; гарантии инвесторам и партнерам.

Механизм производства на предприятии складывается в результате осознанного действия его руководства, основу которого должна составлять экономическая оценка принимаемых управленческих решений. В условиях рыночной экономической системы, когда целью предпринимательской деятельности является получение прибыли, результатом управленческих решений, их оценкой следует принять динамику прибыли предприятия. Это связано с необходимостью рассмотрения деятельности предприятия как единой экономической системы, чьи экономические показатели находятся в сложной взаимосвязи. Наиболее сильное влияние на результаты деятельности предприятия оказывает воздействие внешних управлений. Несмотря на детерминированность взаимосвязи экономических показателей работы предприятия, в ней присутствует случайная составляющая, связанная с риском. Наличие риска – важнейшая черта предпринимательства. Риск – это возможность ущерба, непредвиденных потерь предполагаемой прибыли, имущества предприятия, его денежных средств. Экономическая деятельность промышленного предприятия ниже структурирована группой задач анализа и оценки, которые целесообразно учитывать при представлении предприятия для определенного производства.

Задачами анализа эффективности средств труда являются:

- оценка динамики средств труда;
- оценка состояния средств труда;
- оценка использования средств труда;
- оценка эффективности средств труда;
- оценка влияния применения средств труда на динамику их эффективности;
- определение возможного прироста продукции при улучшении применения основных средств.

Общий объем разнообразных средств труда обычно представляется в стоимостном выражении. Для этого необходима оценка каждого объекта фондов, зависящая от времени, к которому она приурочена, и от состояния основных фондов. Важной задачей анализа является изучение изменений объема и состава основных фондов, которые происходят со временем.

Основными задачами анализа использования и эффективности труда являются:

- оценка динамики численности персонала предприятия;
- оценка использования рабочего времени;
- оценка динамики производительности труда;
- оценка влияния факторов производительности труда на его динамику;
- определение возможного прироста продукции под влиянием факторов повышения производительности труда.

Основными задачами анализа затрат на производство продукции являются:

- факторный анализ динамики всей товарной продукции;
- факторный анализ динамики сравнимой товарной продукции;
- факторный анализ основных слагаемых затрат;
- анализ себестоимости отдельных изделий.

Анализ затрат на производство продукции, ее себестоимости, является одним из важнейших в системе экономического анализа. От динамики себестоимости продукции во многом зависят такие показатели деятельности предприятия, как прибыль и рентабельность. В процессе анализа себестоимости продукции обычно исходят из действующей классификации затрат (их группировки):

- по элементам затрат;
- на прямые и косвенные затраты;
- на постоянные и переменные расходы;
- на явные и неявные издержки;
- затраты на оплату труда;
- отчисления на социальные нужды;
- амортизация основных производственных средств.

Промышленной продукцией считается прямой полезный результат производственной деятельности предприятия, выраженный в форме продуктов. **Продуктами предприятия** являются созданные на

нем изделия, детали и другие объекты, объем каждого вида которых, может быть выражен в натуральных единицах. По стадии готовности продукты промышленного предприятия могут представлять собой: готовые изделия, полуфабрикаты, незавершенное производство.

Учет продукции предприятия может производиться в натуральном, условно-натуральном и стоимостном выражении. Учет продукции в натуральном выражении сводится к характеристике ее в штуках, метрах, тоннах и т. д. Этот метод является основным методом учета. Дополнением и развитием учета продукции в натуральном выражении является ее учет в условных натуральных единицах. Он применяется для определения общего объема производства всех разновидностей одного и того же продукта. Для этого все разновидности продукта приводятся к одной из его разновидностей, единица которой называется условным натуральным измерителем продукции. Приведение осуществляется с помощью коэффициентов на основе важнейшей характеристики полезности продукции. Учет продукции в стоимостном выражении производится, прежде всего, для получения общих итогов произведенной и реализованной продукции для оценки их динамики.

Приведенные выше основные **понятия и процессы современной экономики индустриального производства сложных систем и продуктов** будут далее использоваться как база при формировании и изложении экономики индустриального производства сложных программных продуктов. Однако следует учитывать, что не все компоненты этой экономики детально разработаны вследствие принципиальной новизны и сложности производства программной продукции. В результате, естественно, имеется неравномерность в детализации и конкретизации экономики ряда производственных процессов.

Глава 2

ЗАДАЧИ И ОСНОВНЫЕ ОСОБЕННОСТИ ЭКОНОМИКИ ПРОИЗВОДСТВА СЛОЖНЫХ ПРОГРАММНЫХ ПРОДУКТОВ

Задачи анализа современной экономики проектирования и производства программных продуктов

Накоплен и опубликован значительный объем полезных методических и статистических данных об экономических характеристиках создания программных комплексов разных классов. Наиболее полные исследования, обобщения и экономические характеристики реализованных проектов отражены в двух книгах Б. Бозма [3, 38]. В предлагаемой монографии автором приведены также некоторые результаты собственных оригинальных исследований [16]. Анализ этих данных совместно с результатами других авторов позволил заложить *основы экономики* производства сложных программных продуктов, методик прогнозирования необходимых ресурсов для разработки комплексов программ, их достоверного экономического планирования и формирования производственных процессов. Новизна и отсутствие твердо установленной терминологии в данной области привело к необходимости интерпретации ниже некоторых использованных характеристик и терминов.

Сложные комплексы программ являются *компонентами систем*, реализующими обычно их основные, функциональные свойства и создающими предпосылки для последующего развития и изменений их жизненного цикла (ЖЦ). Производство программных продуктов, реализация их ЖЦ, методология управления и изменения программ зависит от многих факторов, от экономических ресурсов, квалификации специалистов, технических, организационных и договорных требований и сложности проекта. При экономическом анализе жизненного цикла сложных программных комплексов целесообразно выделять два крупных этапа: *анализ факторов, влияющих на экономику производства конкретных программных продуктов*, и *про-*

гнозирование их экономических характеристик. При проектировании необходимо формировать требования и архитектуру предполагаемого комплекса программ, оценивать его экономические характеристики и процессы производства программного продукта.

Множество текущих состояний и модификаций компонентов в жизненном цикле сложных комплексов программ, менеджерам необходимо упорядочивать, контролировать их развитие и реализацию участниками проекта. Организованное, контролируемое и методичное **отслеживание динамики проектирования и производства**, а также изменений в жизненном цикле программ и данных, их разработка при строгом учете и контроле ресурсов, является **базовой проблемой эффективного, поступательного развития** каждого крупного комплекса программ и системы **методами программной инженерии**.

Основным способом совершенствования производства программных продуктов и повышения квалификации специалистов в этой области многие менеджеры считают накопление и распространение опыта наиболее успешных разработчиков, который выражается в быстром продвижении проектов. Однако, повышая квалификацию **отдельных разработчиков**, невозможно решать современные проблемы производства сложных программных продуктов. Производительность индивидуального труда наиболее квалифицированных и продуктивных разработчиков может во много раз превышать производительность менее способных, в то время как численность последних в такое же число раз, превышает численность первых [7, 24]. Кроме того, опыт наиболее продуктивных специалистов **не может быть применен коллективами специалистов в промышленных масштабах**, поскольку **ориентирован на людей с исключительными способностями**. Для удовлетворения требований к производству сложных программных продуктов должны быть исключены надежды на расточительное использование таланта уникальных, высококвалифицированных разработчиков, в том числе для выполнения рутинных повторяющихся процессов, чтобы они могли больше времени тратить на постановку, формализацию и организацию производства сложных программ. В больших коллективах необходимо экономно использовать эти ограниченные, ценные ресурсы специалистов.

Реализация крупных программных проектов встречается со сложными проблемами, тем не менее, быстро возрастает как общий

объем производства программных продуктов, так и сложность отдельных проектов. Однако многие предприятия, занимающиеся производством программных продуктов, не уделяют должного внимания экономике и эффективному применению современных методов организации, автоматизации и обеспечения всего жизненного цикла программных средств. Эти проблемы и их компоненты в разной степени могут отражаться на экономических характеристиках, качестве и конкурентоспособности программных продуктов, на них целесообразно обращать внимание и по возможности решать заказчикам, руководителям и разработчикам сложных программных проектов.

Начиная разработку крупного проекта, руководители, прежде всего, должны решить *задачу целесообразно ли его создание* и оценить какова будет возможная эффективность применения готового продукта, оправдаются ли затраты на разработку и использование. Поэтому такие проекты традиционно должны начинаться с анализа и *экономического обоснования* предстоящего жизненного цикла и эффективности использования предполагаемого программного продукта – рис. 2.1. Экономическое обоснование проектов на начальном этапе их производства должно содержать оценки экономических рисков реализации поставленных целей, обеспечивать возможность планирования и выполнения жизненного цикла продукта, или указывать на недопустимо высокий риск его реализации и целесообразность прекращения проектирования и производства.

Задача экономического анализа проектов в программной инженерии заключается, прежде всего, в разработке, освоении и использовании методов для оценивания экономических характеристик производства сложных программных продуктов. Цель экономического обоснования проектов комплексов программ состоит в помощи их руководителям определять:

- целесообразно ли проводить или продолжать работы над конкретным проектом программного продукта для детализации требований, функций и экономических характеристик или следует его прекратить вследствие недостаточных ресурсов специалистов, времени или возможной трудоемкости производства;
- при наличии достаточных ресурсов следует ли провести маркетинговые исследования для определения рентабельности полного выполнения проекта и производства программного продукта для поставки заказчику или на рынок;

Задачи и основные особенности экономики проектирования и производства программных продуктов включают:

- задачи анализа современной экономики проектирования и производства сложных программных продуктов:
 - экономические характеристики предприятий при проектировании и производстве программных продуктов;
 - классификация экономических характеристик производства и факторов, влияющих на экономику программных продуктов;
 - сбор и обобщение экономических характеристик производства реальных программных продуктов;
 - анализ планирования и достоверность прогнозирования экономических характеристик программных продуктов;
 - оценивание и выбор квалифицированных специалистов - подрядчиков для производства программных продуктов;
- задачи создания и организации экономически эффективного проектирования и производства сложных программных продуктов:
 - создание и освоение методов и технологий экономического обоснования производства программных продуктов;
 - формирование требований, эффективных экономических и технических решений производства программных продуктов;
 - обеспечение экономических характеристик, предсказуемого и управляемого уровня качества программного продукта;
 - отслеживание изменений экономических характеристик комплексов программ с учетом затрат ресурсов;
 - обеспечение и удостоверение требуемых функций и качества готового программного продукта в процессе испытаний ;
- задачи подготовки и обучения специалистов для экономически эффективного проектирования и производства программных продуктов:
 - подготовку специалистов, ответственных за оценивание экономических характеристик процессов производства;
 - формирование «команды» квалифицированных специалистов для экономически эффективного производства программного продукта;
 - выделять и обучать специалистов, ответственных за анализ и прогнозирование экономических характеристик производства;
 - организацию структуры коллективов для экономически эффективного производства программных продуктов;
 - обучение и подготовку руководителей и специалистов по экономике производства сложных программных продуктов.

Рис. 2.1

- достаточно ли полно и корректно формализованы требования к проекту, на основе которых проводились оценки экономических характеристик, или их следует откорректировать и выполнить повторный анализ с уточненными исходными данными;
- есть ли возможность применить готовые повторно используемые компоненты, в каком относительном количестве от всего комплекса программ и рентабельно ли их применять в конкретном проекте или весь проект целесообразно разрабатывать как полностью новый.

За последние несколько лет ряд исследований и работ *по сбору и обобщению экономических данных о производстве программных продуктов* заложили основы для методов и моделей оценивания затрат, которые обладают удовлетворительной точностью. Современная экономическая модель оценки их производства считается хорошей, если с ее помощью можно оценить затраты на программные комплексы с точностью 20% , при условии использования модели, на которую она ориентирована [3, 31]. Имеющиеся модели не всегда столь точны, как хотелось бы, но могут весьма существенно помочь при экономическом анализе и обосновании решений создания сложных программных продуктов. Для сбора и обобщения экономических характеристик о производстве программных продуктов необходимо детально исследовать требуемые ресурсы для современных процессов создания и использования комплексов программ различных классов и назначения – встроенных, бортовых, коммерческих, административных, учебных, (см. рис. 2.1).

Скоординированное применение методов, стандартов и средств в процессах производства, развития и применения комплексов программ позволяет исключать многие виды дефектов или значительно ослаблять их влияние. Тем самым *экономические характеристики производства* и уровень достигаемого качества становится *предсказуемым и управляемым*, непосредственно зависящим от ресурсов, выделяемых на их достижение, а главное, от системы качества и эффективности технологий, используемых на всех этапах жизненного цикла. *Основными целями* упорядочивания, регламентирования планирования, этапов и процессов производства программных комплексов являются:

- снижение трудоемкости, длительности, стоимости и улучшение всех характеристик программных продуктов;

- повышение качества разрабатываемых и/или применяемых компонентов и программных комплексов в целом при их производстве, сопровождении и эксплуатации;
- обеспечение возможности расширять комплексы программ по набору прикладных функций, совершенствовать и масштабировать их в зависимости от изменения решаемых задач, внешней среды и потребностей заказчика;
- обеспечение переносимости прикладных программ и данных между разными аппаратными и операционными платформами и повторного использования программных компонентов в различных проектах.

Необходимы активные *исследования на разных уровнях детализации проектов*, начиная от экономики и планирования создания программных продуктов в масштабах предприятия и кончая экономикой выполнения частных операций отдельными специалистами при проектировании или производстве конкретных продуктов. Одна из важнейших задач состоит в том, чтобы увязать четкими экономическими категориями взаимодействие разных специалистов и предприятий в *типовой производственной цепочке*: заказчик – проектировщик – производитель – пользователь. Для этого объект – программный продукт и все процессы взаимодействия в такой цепочке должны быть связаны системой экономических и технических характеристик в той или иной степени использующих основные экономические показатели – реальные затраты ресурсов: финансов, труда и времени специалистов на конечный продукт. Это позволит создавать *теоретические и практические основы современной экономики производства сложных программных продуктов*.

Крупные программные продукты являются *одними из наиболее сложных объектов, создаваемых человеком*, и в процессе их производства – *творчество специалистов* как поиск новых методов, альтернативных решений и способов осуществления заданных требований, а также формирование и декомпозиция этих требований составляют значительную часть всех трудозатрат. Индустриализация производства комплексов программ позволяет автоматизировать многие не творческие, технические и рутинные операции и этапы, а также облегчать творческие процессы за счет селекции, обработки и отображения информации, необходимой для принятия творческих

решений. Следствием этого должно являться значительное сокращение доли затрат на творческий труд в непосредственных затратах на разработку комплексов программ.

В программной инженерии **неуклонно повышаются размеры и сложность** создаваемых продуктов, что вызывает возрастание затрат творческого труда на единицу размера новых программ. В перспективе, несмотря на автоматизацию и повышение инструментальной оснащенности технологии производства комплексов программ, доля творческого труда при создании полностью новых крупных программных продуктов возрастает. Даже при сокращении суммарных затрат на разработку программных компонентов за счет автоматизации нетворческого труда, все более определяющей для экономических характеристик создания программных продуктов становится доля затрат на творческий труд и **возрастают требования к творческим способностям при отборе и обучении специалистов**.

По мере повышения квалификации коллективов и автоматизации творческой части труда следует ожидать асимптотического приближения проектов к предельным значениям **относительных** экономических характеристик новых разработок. Эти значения определяются **интеллектуальными возможностями человека по интенсивности принятия творческих решений**. Им соответствуют наличие предельных значений производительности труда и длительности разработки сложных комплексов программ. Для их оценки необходимо изучение и экстраполяция прецедентов и экспериментальных данных реальных разработок программных продуктов с наилучшими экономическими характеристиками с учетом возрастания квалификации специалистов и уровня автоматизации производства. Вряд ли можно ожидать в ближайшие годы радикального повышения производительности труда при создании полностью новых, крупных программных продуктов. Еще более консервативна длительность таких разработок [18, 36].

Ограниченность ресурсов при создании крупных программных продуктов приводит к целесообразности решения задач планирования, упорядочения и применения экономически эффективных методов автоматизации обеспечения ЖЦ комплексов программ с целью достижения требуемого качества и достоверного его определения. Для каждого крупного проекта, выполняющего ответственные функции, необходимо **прогнозировать требующиеся ресурсы**, разрабаты-

вать и применять комплексную систему качества, специальные планы и Программу, методологию и инструментальные средства, обеспечивающие требуемые качество, надежность и безопасность функционирования программного продукта. Для этого следует применять современные методы и стандарты при подготовке промышленных технологий, методик производства и испытания конкретных продуктов, однозначно отражающих степень удовлетворения исходных требований заказчика и пользователей, а также для сравнения характеристик продуктов разных поставщиков и выявления среди них предпочтительных.

Задачи оценивания и выбора квалифицированных и надежных специалистов-подрядчиков, способных создавать сложные программные продукты и базы данных требуемого качества в разумные сроки с учетом ограничений на используемые ресурсы стоят остро для многих заказчиков и пользователей современных сложных вычислительных систем. Для их решения поставщикам комплексов программ, кроме программистов – кодировщиков, необходимо иметь системных аналитиков, архитекторов и топ-менеджеров проектов, а также специалистов по комплексированию, испытаниям и обеспечению качества современных сложных программных продуктов. Они должны знать передовые индустриальные методы, технологии и международные стандарты, поддерживающие и регламентирующие жизненный цикл комплексов программ, а также инструментальные системы обеспечения качества, верификации, тестирования и сертификации программных продуктов. Для этого требуется, прежде всего, **системотехническая квалификация специалистов** предприятий, берущихся за производство сложных программных продуктов высокого качества.

Большие ошибки оценивания экономических характеристик на начальных этапах разработки, можно значительно сокращать при относительно небольших усилиях, применяя, в частности, формализованные методики их экспертной оценки. Тем самым такие проекты с самого начала могут выполняться с учетом более достоверной оценки необходимых экономических и других факторов. Большую часть негативных последствий можно избежать, используя существующие, достаточно точные **методы оценивания и прогнозирования экономических характеристик, а также управление проектами** для их успешного завершения с учетом следующих факторов:

- исходные тексты программных компонентов различны по размерам и сложности, и по отдельности не определяют сложность и размер конечного программного продукта;
- разработка сложных продуктов требует творчества и сотрудничества разных специалистов, индивидуальное и групповое поведение которых, как правило, трудно предсказать;
- в области экономики жизненного цикла сложных комплексов программ накоплен относительно небольшой опыт анализа и количественных оценок, и его трудно увеличивать, не обобщая разрозненные прецеденты проектов.

*Задача состоит в освоении специалистами современных методов экономического анализа, оценивания и **прогнозирования необходимых ресурсов при производстве комплексов программ.*** Тем самым должен быть выделен очень важный, базовый раздел из всей экономики жизненного цикла программных комплексов. Такое выделение определяется тем, что без подобных базовых исследований вряд ли возможно последующее серьезное развитие экономики в этой отрасли. Внимание должно быть сосредоточено на концептуальной основе анализа и распределения труда в процессе разработки комплексов программ на ***факторах, определяющих реальные трудозатраты и другие экономические характеристики,*** а также на исследовании их в реализованных современных разработках.

Задачи создания и организации экономически эффективного проектирования и производства программных продуктов

Задачи рационального сочетания целей, стратегий действий, конкретных процедур и доступных ресурсов необходимо решать для достижения основной цели ***получения программного продукта с заданными функциональными характеристиками и качеством*** (см. рис. 2.1). Базой эффективного управления проектом программного комплекса является ***план,*** в котором задачи исполнителей частных работ должны быть согласованы с выделяемыми для них ресурсами, а также между собой, по результатам и срокам их достижения. Планирование программных проектов должно обеспечивать компромисс между требующимися характеристиками создаваемой системы и ограниченными ресурсами, необходимыми на ее разработку и применение.

По мере уточнения исходных требований к объекту разработки, внешней среде применения и ресурсам, в процессе системного анализа и проектирования возрастает достоверность планирования, которое должно проходить *этапы* [5, 15]:

- обследование объектов и среды проектирования для предварительной формализации целей, назначения и задач проекта;
- первичное прогнозирование возможных характеристик и требований к программному продукту на базе обобщения данных ранее реализованных подобных прототипов и создание концепции проекта;
- подготовка предварительного плана выполнения этапов и частных работ с учетом допустимых затрат ресурсов на их реализацию;
- управление детализацией и реализацией плана производства, его оперативной корректировкой и перераспределением ресурсов в соответствии с особенностями развития компонентов программного комплекса;
- обобщение и накопление результатов планирования и управления конкретным проектом для использования этих данных в качестве прототипов при производстве программных продуктов.

На каждом этапе должен проводиться поиск *эффективных технических и экономических решений реализации проекта*, исследование и сопоставление альтернативных действий, которые должны приводить к достижению поставленных целей производства программного продукта. В результате процессы планирования проекта и его выполнения обычно развиваются параллельно. Уже при первичном прогнозировании развития проекта должны оцениваться альтернативные характеристики объекта и среды разработки и выбираться наиболее подходящие для производства в соответствии с поставленными целями и имеющимися ресурсами. Сравнение альтернатив следует проводить по величине достигаемого эффекта проекта в зависимости от затрат на его достижение (желательно, по показателю «эффективность/стоимость»).

Программные комплексы все больше встраиваются в различные системы. Работа с такими проектами требует от программных специалистов широкого взгляда на *общие технические и экономические задачи проектирования сложных систем*. Менеджерам и аналитикам программных проектов необходимо участвовать в выработке требований для всей системы, а также понимать прикладную область применения продукта еще до начала обдумывания функций компо-

нентов и их интерфейсов, требованиям которых должен будет отвечать программный продукт.

Для сокращения затрат **необходимы новые технологии, методы создания и управления** сложными проектами программных систем. Это приводит к увеличению роли интеграции таких компонентов, соответствующих методов и инструментария программной инженерии. Однако вследствие их принципиальной новизны и сложности, они трудно воспринимаются традиционными программистами компонентов и множеством преподавателей отечественной высшей школы. Коренные отличия между методами и инструментарием индивидуального, «художественного» программирования небольших программ и технологией планомерного, регламентированного производства крупных программных продуктов приводят к тому, что последние медленно осваиваются и входят в практику слаженной работы больших коллективов специалистов. Эти обстоятельства **отражаются на существенном отставании от мирового уровня** по конкурентоспособности, количеству и качеству отечественных программных продуктов. Каждая производственная технология должна быть достаточно зрелой, чтобы быть развернутой в промышленных масштабах, их **интеграция включает** [7]:

- разработку архитектур – структур для производства продуктов на основе общих архитектурных стилей;
- разработку инструментов на базе выбранного языка для поддержки производства продуктов посредством адаптации, конфигурирования и сборки структурных компонентов;
- использование инструментов проектирования для вовлечения заказчиков и быстрой реакции на изменения требований за счет инкрементного построения программ с сохранением их работоспособности при внесении изменений;
- фиксация проектных решений в форме, непосредственно порождающей исполняемые программы.

Неопределенность применяемых понятий, требований и характеристик качества присуща крупным, наукоемким проектам комплексов программ, однако многочисленные спекуляции разработчиков на их значениях приучила заказчиков не доверять рекламируемым достоинствам производства программных продуктов [9, 31]. Во многих случаях контракты и предварительные планы на создание сложных программных комплексов и баз данных подготавливаются и

экономически оцениваются на основе неформализованных представлений заказчиков и разработчиков о требуемых функциях и характеристиках качества систем. Многочисленные провалы проектов выявили необходимость формализации методов взаимодействия и обеспечения взаимопонимания разработчиков с заказчиком или потенциальными пользователями создаваемого продукта с самого начала проекта с целью конкретизации его функций и уточнения требований к качеству. Ошибки, обусловленные неопределенностью или некорректностью технических заданий и спецификаций требований, могут и должны быть выявлены на ранних стадиях проектирования, что способствует его ускорению и повышению качества.

Возрастание *сложности и ответственности современных задач*, решаемых сложными системами, а также возможного ущерба от недостаточного качества комплексов программ, значительно повысило актуальность проблемы освоения методов стандартизированного описания требований, оценивания экономических характеристик и качества на различных этапах их жизненного цикла. Выявилась необходимость систематизации реальных характеристик качества программных продуктов, применения стандартов для выбора из них необходимой номенклатуры и требуемых решений для конкретных комплексов программ. Обещания разработчиков в контрактах с заказчиками создать высококачественные продукты в согласованные сроки во многих случаях не выполняются как вследствие различий в понимании требуемого качества, так и вследствие неумения оценить экономические ресурсы, необходимые для достижения заданного качества программ. Стратегической задачей в жизненном цикле современных систем стало *обеспечение и совершенствование качества производства сложных программных продуктов при реальных ограничениях на использование доступных экономических ресурсов производства*.

Широкое многообразие классов и видов программ, обусловленное различными функциями систем, предопределяет формальные трудности, связанные с методами и процедурами *доказательства соответствия поставляемых продуктов* условиям контрактов и требованиям потребителей. По мере расширения применения и увеличения сложности программных продуктов выделились области, в которых ошибки или недостаточное качество программ или данных могут нанести ущерб, значительно превышающий положительный

эффект от их использования. В этих критических случаях недопустимы аномалии и дефекты функционирования программных продуктов при любых искажениях исходных данных, сбоях и частичных отказах аппаратуры и других нештатных ситуациях.

Задачи формирования требований к характеристикам и качеству программного продукта включают анализ свойств, характеризующих его функционирование с учетом технологических и экономических ресурсных возможностей производства. При этом под качеством функционирования понимается совокупность свойств, обуславливающих пригодность продукта обеспечивать надежное, безопасное и своевременное представление требуемой информации потребителю для ее дальнейшего использования по назначению. В соответствии с принципиальными особенностями программного продукта при проектировании должны выбираться номенклатура и значения требований к характеристикам качества, необходимым для его эффективного применения пользователями, которые впоследствии отражаются в технической документации и в спецификации требований на конечный продукт. Каждый критерий качества может использоваться, если определена его метрика и может быть указан способ ее оценивания и сопоставления с требуемым значением. Для конкретных видов программных продуктов доминирующие критерии качества выделяются и определяются при проектировании систем их функциональным назначением и требованиями технического задания.

Качество в использовании – это основное качество системы, содержащей программный продукт, которое воспринимается пользователями [17, 41]. Оно измеряется в терминах результатов функционирования и применения программ. Цель такого оценивания – определение, имеет ли продукт требуемый экономический эффект в специфическом контексте использования. Качество в среде пользователей может отличаться от качества в среде разработчиков, поскольку некоторые функции могут быть невидимы пользователю или не использоваться им. Пользователь оценивает только те атрибуты качества программного продукта, которые видимы и полезны ему в процессе реального применения.

Качество изменяется в течение жизненного цикла комплекса программ, то есть его требуемое и реальное значение вначале ЖЦ почти всегда **отличается от фактически достигнутого при завершении производства** и качества поставляемой пользователям

версии продукта. На практике важно оценивать качество программ не только в завершённом виде, но и в процессах их проектирования, производства и сопровождения. Кроме того, оценки показателей качества могут быть субъективными и отражать различные точки зрения и потребности разных специалистов. Чтобы эффективно управлять качеством на каждом этапе ЖЦ, необходимо уметь определять и примирять эти различные представления требуемого качества и его изменения. Требуемые характеристики качества с различных позиций отражают их свойства и особенности, и, в свою очередь, зависят от ряда факторов и ограничений.

Обеспечение и удостоверение качества сложных программных продуктов должно базироваться на проверках и испытаниях:

- технологий обеспечения жизненного цикла программных комплексов, поддержанных регламентированными системами качества;
- готового программного продукта с полным комплектом адекватной эксплуатационной документации.

При системном анализе и проектировании программных комплексов необходимо определять и учитывать связи, влияние и взаимодействие следующих **основных факторов**, которые отражаются на их качестве:

- назначение, содержание и описание функциональных характеристик и атрибутов, определяющих специфические особенности целей, задач, свойств и сферы применения конкретного программного продукта – его функциональную пригодность;
- конструктивные характеристики качества, способствующие улучшению и совершенствованию назначения, функций и возможностей применения программного продукта;
- метрики, меры и шкалы, выбранных и пригодных для измерения и оценивания конкретных характеристик и атрибутов качества продукта с учетом требуемой достоверности;
- цели и особенности потребителей, оценивающих качество продукта;
- внешние и внутренние, негативные факторы, влияющие на достигаемое качество создания и применения программного продукта;
- доступные ресурсы, ограничивающие возможные величины реальных характеристик качества.

Задача удостоверения достигнутого качества функционирования сложных программных продуктов и методов обеспечения их жизненного цикла базируется на сертификации, аттестованными проблемно-ориентированными испытательными лабораториями. Применение сертифицированных систем качества на предприятиях-разработчиках должно гарантировать высокое, устойчивое качество процессов производства конечного программного продукта. Основой сертификации должны быть детальные и эффективные методики испытаний, специально разработанные тестовые задачи и генераторы тестов для их формирования, а также квалификация и авторитет испытателей. Для этого заказчики должны выбирать подрядчиков-исполнителей своих проектов, имеющих системы обеспечения качества программных средств и сертификаты, удостоверяющие реализацию и применение системы качества производства предприятием.

Глубокая взаимосвязь качества разработанных комплексов программ с качеством технологии их создания и с затратами на разработку становится особенно существенной при необходимости получения конечного продукта с предельно высокими значениями показателей качества. Непрерывный рост требований к качеству стимулировал создание и активное применение международных стандартов и регламентированных технологий, автоматизирующих процессы их жизненного цикла, начиная с инициирования проекта. Для решения этой задачи необходимо детально исследовать современные процессы производства и использования программ различных классов и назначения. Перечисленные задачи требуют для своего решения выполнения крупных, комплексных научно-исследовательских работ, многие из которых еще не поставлены и далеки от разрешения [36].

Задача анализа экономической эффективности выполнения проекта требует использования сведений, определяющих показатели выполнения применяемых процессов производства. Такие сведения и уроки следует собирать, пока особенности реализации проекта свежи в памяти его разработчиков. Их необходимо применять в процессах производства программных продуктов после завершения основных и промежуточных этапов выполнения проекта, когда они могут оказать влияние на текущее состояние производства и качество программного продукта. Это особенно важно для крупных, продолжительных проектов производства версий программных продуктов. Такой анализ

нужен не только для оценки процесса выполнения проекта и мелких промежуточных этапов жизненного цикла разработки программного продукта, но и для совершенствования процессов производства.

Совещания с целью анализа состояния и экономики производства должны быть организованы для конструктивного сбора информации, а не для всякого рода разбирательств. Именно здесь может пригодиться опыт посредничества и умение своевременно вносить изменения. Менеджер проекта должен в самом его начале уведомить коллектив о своем намерении отслеживать все производственные процессы и их экономические характеристики, а также дать им ряд указаний относительно тех сведений, которые будут собираться для анализа. Выполнять анализ весьма важно, поскольку именно здесь уроки могут быть обобщены и переданы по наследству для сохранения в архиве предприятия и дальнейшего совершенствования производственных процессов.

Каждое предприятие, занимающееся разработкой программных продуктов, должно создавать архив компонентов и характеристик производственных процессов, который позволяет находить образцы настраиваемых процессов, поддерживать журналы регистрации усовершенствований и **изучать уроки, извлеченные из предыдущих проектов**. На начальном этапе нового программного проекта следует ознакомиться с содержимым такой библиотеки и воспользоваться уроками прошлого в новом проекте, причем подобный процесс усовершенствований должен быть непрерывным. К главным элементам подобных отчетов относятся сводки по выбранным и реализованным характеристикам процессов и рекомендации по их изменению, которые вытекают из уроков, извлеченных из проекта. Отчеты должны быть краткими и удобными для чтения, ибо они приносят большую пользу, если содержат наиболее важные сведения о результатах совещаний с целью проведения анализа выполнения проекта.

Каждое предприятие в целом и каждый процесс должен быть рассмотрен с точки зрения ответственности юридической стороны договора с заказчиком. Предприятие может выполнять один или несколько производственных процессов. Сторона, выполняющая процесс, несет ответственность за весь процесс, даже если выполнение отдельных задач поручено другим специалистам. **Распределение ответственности** специалистов по архитектуре и процессам жизненного цикла комплекса программ облегчает контроль применения

стандартов для конкретного проекта, в который может быть вовлечено множество лиц.

Задачи подготовки и обучения специалистов для экономически эффективного проектирования и производства программных продуктов

В жизненном цикле сложных комплексов программ для обеспечения их высокого качества *целесообразно выделять специалистов, ответственных за анализ, оценивание и прогнозирование экономических характеристик производства*, за соблюдение промышленной технологии создания и совершенствование программных продуктов, за измерение и контроль затрат, качества комплексов программ в целом и их компонентов. Задача состоит в том, чтобы научить специалистов анализу и оцениванию конкретных экономических факторов, влияющих на характеристики функционирования программных продуктов со стороны реально существующих и потенциально возможных негативных воздействий и ограничений ресурсов проектов. Необходима подготовка и воспитание квалифицированных специалистов в области индустрии, экономики и производства сложных программных комплексов, их обучение методам и современной программистской культуре промышленного создания сложных высококачественных программных продуктов (см. рис. 2.1).

В монографии рассматриваются преимущественно средние и большие проекты комплексов программ, создаваемые крупными коллективами специалистов. В таких проектах на чистое творчество, искусство и научные исследования отдельных специалистов, преобладающие в небольших индивидуальных разработках, накладывается множество технических работ, характерных для индустриального проектирования и производства программных продуктов. Вследствие этого значительно нивелируются индивидуальные особенности и квалификация отдельных специалистов, и появляется возможность оценивать усредненную производительности труда и другие экономические характеристики крупных коллективов. Программный специалист – это *член команды*, поэтому должен обладать *навыками общения* и межличностных отношений, а также уметь планировать не только свою работу, но и координировать ее с работой других специалистов. Анализируемые ниже процессы

обычно ограничены временем от этапа оформления требований технического задания на разработку до этапа завершения испытаний опытного образца первой версии программного продукта, соответствующего требованиям заказчика.

Задачи выбора, организации структуры и состава коллектива специалистов привели к появлению **новых требований к обучению и к дифференциации специалистов по программной инженерии**, обеспечивающих основные этапы жизненного цикла комплексов программ [24, 42]. Этим специалистам недостаточно навыков процедурного программирования небольших модулей и компонентов, а необходимы глубокие знания системотехники, технологии и стандартов проектирования, методов обеспечения и контроля качества сложных комплексов программ в определенной области их применения. Эти специалисты должны владеть **новой интеллектуальной профессией**, обеспечивающей высокое качество программных продуктов, а также испытания и удостоверение реального достигнутого качества на каждом этапе разработки и совершенствования программ. Производство крупных программных продуктов различных классов, разделение труда специалистов по квалификации при разработке программ и данных, структура и организация коллективов, а также экономика таких разработок, стали важнейшей частью задачи выбора и обучения коллективов специалистов для обеспечения всего ЖЦ сложных программных продуктов.

В жизненном цикле сложных комплексов программ участвуют специалисты различной квалификации и степени ответственности за результаты своей деятельности:

- **заказчики** определяют и несут ответственность за требования к функциям и характеристикам качества программного продукта, за финансирование, доступные, адекватные ресурсы для обеспечения его жизненного цикла;

- **разработчики** должны гарантировать и отвечать за выполнение требований заказчиков программного продукта с учетом выделенных ресурсов;

- **пользователи** имеют право применять программный продукт и адаптировать его к особенностям использования и внешней среды только в пределах, определенных эксплуатационной документацией, созданной разработчиками.

Для *организации эффективной структуры коллектива разработчиков программных комплексов* следует научиться учитывать конкретные цели специалистов, участвующих в проекте, их *психологическую совместимость*, способность к дружной коллективной работе, опыт взаимодействия в составе определенного коллектива, а также другие объективные и субъективные свойства участников проекта. При этом большое значение может иметь личная мотивация и психологические особенности поведения разных специалистов при комплексной работе над сложным проектом. Эти характеристики могут быть обобщены в качественный показатель влияния сложности взаимодействия специалистов в коллективе, которому могут быть сопоставлены оценки возможного изменения экономической эффективности производства программного продукта. Наилучшим обычно считается непрерывное корректное взаимодействие организованных специалистов с большим опытом работы в конкретном коллективе при согласованности их целей, планов и методов работы. В остальных случаях в той или иной степени (даже в 3 – 5 раз) может возрастать трудоемкость производства продукта, что нельзя не учитывать при обосновании крупных проектов.

Уровень *квалификации заказчика* и определенность требований технического задания на производство продукта может сильно влиять на суммарные затраты и длительность создания комплекса программ. Первоначальное, техническое задание зачастую оказывается недостаточно квалифицированным и подвергается в дальнейшем многократным изменениям. Изменения технического задания заказчиком и объем переделок непосредственно отражаются на эффективной производительности труда специалистов производства. Особенно сильно низкая достоверность технического задания может влиять на попытки заказчика форсировать сроки разработки. Этому же может способствовать различие между заказчиком и разработчиком в квалификации, уровне понимания целей разработки и необходимых затрат на реализацию требований.

Даже при испытаниях заказчик зачастую обнаруживает, что решаются не совсем те задачи и не совсем так, как ему нужно, вследствие чего может быть необходима большая переработка готовых программ. Даже квалифицированные заказчики вынуждены иногда корректировать техническое задание на любых этапах разработки, что может влиять на снижение экономической эффективности

производства на 10 – 20%. Представители заказчика, участвующие в проекте, должны обучаться формализации автоматизируемых функций систем и технологических процессов, для которых предназначены соответствующие программные продукты, и иметь представление об экономически эффективных путях их реализации.

В каждом проекте программного продукта должен быть лидер – руководитель. *Лидер должен иметь талант и квалификацию, а также быть обучен решать следующие задачи:*

- руководить процессом выявления, конкретизации и формирования требований заказчика продукта;
- осуществлять проверку спецификаций требований к программному комплексу, чтобы удостовериться, что они соответствуют реальной концепции заказчика, представленной детальными функциями;
- вести переговоры с заказчиком, пользователями и разработчиками, определять и поддерживать равновесие между тем, чего хочет заказчик, и тем, что может создать коллектив разработчиков за бюджет, ресурсы и время, выделенные заказчиком для реализации продукта;
- рассматривать конфликтующие пожелания, поступающие от различных участников проекта и находить компромиссы, необходимые для определения набора функций и характеристик, представляющих наибольшую ценность для максимального числа участников и проекта в целом.

При производстве сложных комплексов программ большими коллективами значительно повышается роль *квалификации менеджеров – руководителей реализации* проекта, что непосредственно отражается на производительности труда всего коллектива. Из-за различной квалификации руководителей проектов, суммарные затраты на разработку могут изменяться в несколько раз, как в лучшую, так и в худшую сторону. Некоторые методы организации структуры коллектива и процессов производства, позволяют сокращать негативное влияние этого человеческого фактора.

Разработчики должны иметь в своем составе квалифицированных менеджеров, *проблемно-ориентированных системных архитекторов*, способных переводить не всегда полные и корректные функциональные требования заказчика в конкретные спецификации и технические требования к комплексам программ и к их компонентам.

Специалисты по проектированию сложных комплексов программ (системные архитекторы) должны иметь, прежде всего, хорошую подготовку по системному анализу алгоритмов и комплексов программ в определенной проблемной области, по методам оценки экономической и технической эффективности проектов, организации и планированию крупных разработок компонентов программ и баз данных. Им необходима высокая квалификация по архитектурному построению, комплексному тестированию и испытаниям компонентов и комплексов программ определенных классов, умение организовать коллектив для решения общей целевой задачи системы. Это позволит на ранних этапах исключать или сокращать дефекты, обусловленные различием представления целей и задач проектов, а также их характеристик качества.

Для создания высококачественных комплексов программ, прежде всего, необходимо *обучение специалистов*, процессам организации *взаимодействия представителей заказчика и разработчиков* проекта. Взгляды и требования заказчика, в основном, отражаются на функциональных и потребительских характеристиках программного продукта. Устремления разработчиков направлены на способы их реализации с требуемым качеством. Эти различия исходных интересов и точек зрения на проект приводят к тому, что некоторые неформализованные представления тех и других имеют зоны неоднозначности и взаимного непонимания, что может приводить к конфликтам. Организация четкого взаимодействия и сокращение этих зон требует определенных мероприятий и контактов по обмену знаниями в процессе проведения совещаний, взаимному повышению квалификации и обучению.

Следует *активно привлекать заказчиков* к совещаниям по управлению требованиями и масштабом проекта, чтобы обеспечить как качество, так и своевременность производства программного продукта. Необходимы указания и согласия заказчиков при принятии основных изменений и решений, и только они могут реально определить, как, развивая функции, получить полезный комплекс программ высокого качества, выполненный в срок и в пределах бюджета. Для определения проекта и целей заказчика может понадобиться вести с ним переговоры об объеме работ для команды. Во время переговоров с заказчиком о техническом задании и требованиях, разработчикам необходимо учитывать неизбежные издержки при производстве ком-

плекса (непредвиденные технологические и экономические риски, изменения требований, задержки при приобретении закупаемых компонентов, непредвиденный уход членов команды), которые могут приводить к существенному нарушению графика реализации продукта. Специалистам команды следует **научиться пониманию требований заказчиков**, поскольку именно они, добиваются успеха при применении программного продукта (см. рис. 2.1).

Для производства сложных программных продуктов наиболее часто применяются **две схемы организации коллективов, которые должны осваиваться при обучении специалистами:**

- формирование для выполнения каждого крупного проекта жесткой организационной структуры целостного коллектива с полным составом необходимых специалистов под единым, централизованным руководством лидера проекта;

- выделение руководителя (главного конструктора) и небольшой группы интеграторов, по заданиям которых выполняются частные работы узкими специалистами по компонентам, не входящими организационно в единый коллектив для реализации конкретного крупного проекта.

Первая схема предпочтительна, когда предприятие реализует небольшое число особенно крупных проектов-заказов и имеет возможность для каждого из них комплектовать полноценную, организационно замкнутую, «команду». Она полностью реализует проект и несет ответственность за его качество. Однако при этом возможны простои отдельных специалистов из-за несинхронного ожидания технических заданий или результатов на последовательных этапах разработки компонентов другими специалистами. **Вторая схема** может иметь преимущества для предприятия при большом числе относительно небольших проектов, близких по содержанию и функциональному назначению компонентов. В этом случае большинство специалистов одновременно участвуют в нескольких заказах по локальным заданиям лидеров и интеграторов различных проектов, и может использоваться более полно. Однако задачи интеграторов, их взаимодействие с поставщиками компонентов, руководство и контроль качества, при этом усложняются и требуют более высокой квалификации. В обеих схемах для реализации мероприятий по планированию и управлению жизненным циклом концептуально целостных, крупных продуктов и обеспечения их качества, необходимы организационные

действия системных архитекторов, направленные на подбор и обучение коллектива специалистов разных категорий и специализаций.

Принципиальной задачей улучшения экономических характеристик при производстве сложных комплексов программ является исключение творчества на тех этапах, где возможны **типовые, стандартные решения** и использование апробированных заготовок программных компонентов, не требующие при их применении высококвалифицированного творческого труда. Основой этого является применение унифицированной технологии, готовых, испытанных программных компонентов и стандартизированной архитектуры определенных классов комплексов программ. Использование готовых апробированных модулей может почти исключать творческий труд по их программированию, автономной отладке и документированию. На этих этапах творческие усилия необходимы для отбора и контроля готовых компонентов, а также для разработки новых, отсутствующих среди апробированных. Однако практически полностью сохраняется творческий труд при системном анализе, при комплексировании компонентов и их комплексной отладке, а также во время испытания программного продукта в целом на соответствие требованиям заказчика.

Крупные проекты, как правило, требуют координированной работы больших коллективов или ряда команд. Возрастание сложности производства снижает возможность человека решать задачи интуитивно по мере их возникновения. Чтобы добиться успеха в большом проекте, **необходима четкая координация действий членов «команды»**, которая должна работать по общей утвержденной методологии, чтобы решить проблему реализации требований и качества программного продукта экономически эффективно организованной командой. Одним из наиболее важных факторов является то, что члены команды имеют **различные талант, профессиональные навыки и квалификацию**. Поэтому при выборе специалистов важно учитывать возможную структуру и личный состав большой команды.

При выборе заказчиком надежного поставщика-разработчика продукта необходима **оценка тематической и технологической квалификации** возможного коллектива специалистов, а также его способности экономно реализовать проект с заданными требованиями и качеством. Тематическую квалификацию специалистов в области создания сложных комплексов программ определенного функцио-

нального назначения приблизительно можно характеризовать средней продолжительностью работы в данной проблемной области основной части команды, непосредственно участвующей в разработке алгоритмов, спецификаций требований, программ и баз данных. Технологическая квалификация коллектива характеризуется опытом и длительностью работы с регламентированными технологиями, инструментальными комплексами автоматизации программной инженерии, языками проектирования, программирования и тестирования комплексов программ.

По мере накопления эксплуатируемых программных продуктов и их компонентов все большее число специалистов переходит из области непосредственного программирования и тестирования новых программ в область системного проектирования, управления конфигурацией и создания новых версий программного продукта на базе повторно используемых компонентов (см. рис. 2.1). По некоторым оценкам непосредственным программированием **новых компонентов в мире** занято только около 15 – 20% специалистов, участвующих в создании программных продуктов. При организации **сопровождения и модернизации** крупных программных продуктов следует учитывать важные **психологические факторы**, усложняющие отбор, обучение и деятельность менеджеров и квалифицированных специалистов в этой области [4, 12, 31]:

- эта деятельность требует высокой квалификации, больших творческих и умственных затрат, связанных, прежде всего, с необходимостью одновременного, широкого охвата и анализа множества компонентов и их взаимосвязей, находящихся в различных состояниях завершенности модификаций или устранения дефектов;
- корректируемые компоненты зачастую разрабатывались в прошлом: в разное время, различными специалистами, в различном стиле, с неодинаковой и не полной документацией, что усложняет освоение их содержания при внесении изменений и устранении дефектов;
- сложная, творческая сторона работ при сопровождении вуализуется тем, что приходится овладевать и анализировать программы, разработанные ранее другими специалистами, которые зачастую может оказаться проще не корректировать, а разработать заново;
- комплексы программ, прошедшие широкие испытания и эксплуатацию у заказчиков гарантируют достигнутое качество резуль-

татов функционирования, и любые в них изменения имеют высокий риск внесения дополнительных ошибок и ухудшения этого качества, что усложняет коренные модификации;

- выполняемые работы требуют особой, скоординированной тщательности корректировок и четкого регламентированного взаимодействия ряда специалистов, различающихся квалификацией и уровнем ответственности;

- процессы и результаты сопровождения не отличаются наглядностью и внешним эффектом, проявлением их размера и сложности, вследствие чего не престижны среди рядовых программистов и недооцениваются руководителями крупных проектов.

Зачастую менеджеры и разработчики новой версии продукта не предусматривают этот процесс и требующиеся ресурсы, что значительно снижает эффективность последующего совершенствования и применения созданного программного продукта.

Перечисленные выше специализации и квалификации персонала, участвующего в крупных проектах программных продуктов, **требуют соответствующей их подготовки, отбора и обучения**, которые являются самостоятельной, важной **проблемой развития экономики программной инженерии**. Обучение представляет собой сложный процесс и требует организации и сопровождения обучаемого персонала. Должны быть разработаны и документированы планы, требования и цели обучения, а также разработаны учебники и руководства, включая материалы, используемые для обучения. Персонал, ответственный за выполнение конкретных задач, если это необходимо, должен быть **аттестован** на основе соответствующего образования, подготовки и/или опыта работы. Может также стать необходимым включение в подготовку, ознакомление со специфической (проблемно-ориентированной) областью, для которой будет производиться программный продукт, и повышение квалификации в этой области [12, 22, 30].

Специалисты по проектированию и производству программных продуктов должны быть обучены и хорошо знать системную технику вычислительных систем, поскольку в них программный продукт играет определяющую интеллектуальную роль. Технология и экономика программной инженерии часто являются критическим фактором при разработке сложных вычислительных систем. Высокие темпы роста основных доступных ресурсов аппаратных средств, и

сохраняющаяся потребность в увеличении их использования со стороны некоторых пользователей и сфер применения, приводят к необходимости адекватного совершенствования технологии и экономики производства программных продуктов.

К сожалению, большинство *вузов страны пока этому не учит*, и ограничивается обучением студентов элементам программирования только небольших *простейших программ*. Они *не готовят остро необходимых для современной программной индустрии*: системных аналитиков, *экономистов*, архитекторов и менеджеров комплексов программ, специалистов по комплексированию, испытаниям и обеспечению качества крупных комплексов программ реального времени. Выпускники вузов, как правило, не знают: современных промышленных методов, технологий и международных стандартов программной инженерии, поддерживающих и регламентирующих жизненный цикл комплексов программ, инструментальных систем обеспечения качества, верификацию, тестирование и сертификацию сложных программных продуктов. Это определяет низкую системотехническую и экономическую квалификацию специалистов и ряда отечественных предприятий, берущихся за создание сложных программных продуктов. В результате, многие проекты оказываются неконкурентоспособными, недостаточного качества и требуют длительной доработки для устранения системных и технических дефектов и ошибок.

Руководителям и специалистам необходимо освоить современный комплекс задач, методов и стандартов промышленного создания и развития сложных, тиражируемых программных средств и баз данных требуемого высокого качества. При обучении внимание следует акцентировать на комплексе индустриальных методов и международных стандартов программной инженерии, которые непосредственно обеспечивают *экономически эффективный жизненный цикл* сложных высококачественных программных продуктов и баз данных. Необходимо обучение специалистов современной *программистской культуре промышленного создания* высококачественных программных продуктов – умению формализовать требования и достигать конкретные значения характеристик функционирования и применения сложных комплексов программ, с *учетом технических и экономических ресурсов*, которые доступны для обеспечения и совершенствования качества.

Глава 3

ЭКОНОМИЧЕСКИЕ ХАРАКТЕРИСТИКИ ДЛЯ ОЦЕНИВАНИЯ ПРОИЗВОДСТВА ПРОГРАММНЫХ ПРОДУКТОВ

Статистические исследования экономики производства программных продуктов

При промышленном проектировании и производстве любой продукции применяются следующие *основные экономические характеристики* для оценивания производственных процессов и изделий:

- *стоимость* – финансовые затраты на создание одного экземпляра или комплекта полноценного продукта, полностью удовлетворяющего требования пользователей;
- затраты труда – *трудоемкость* коллектива специалистов, участвующих в создании готового продукта, соответствующего требованиям пользователей;
- *длительность* – время, затраченное специалистом или коллективом на проектирование и производство продукта, удовлетворяющего пользователей;
- *число специалистов* разной квалификации, участвующих в создании продукта соответствующего требованиям.

В ряде промышленных отраслей значительные затраты требуются на изготовление материализованных компонентов и изделий, которые отсутствуют при создании интеллектуальных, программных продуктов. Поэтому целесообразно рассмотреть *экономические особенности* перечисленных выше характеристик, которые в монографии применяются для анализа и оценивания сложных комплексов программ и баз данных.

Экономический анализ производства программного продукта *в денежном выражении* имеет ряд существенных особенностей, кото-

рые ограничили его применение при оценке и прогнозировании экономики таких проектов по следующим причинам:

- предприятия, создающие комплексы программ, имеют значительные различия в уровне заработной платы специалистов, что не всегда адекватно отражается на их производительности труда и финансовых затратах;
- каждое предприятие имеет накладные расходы и налоги, которые могут значительно различаться и никак не влияют на трудоемкость и длительность непосредственного производства программного продукта;
- весьма различны оснащенности предприятий технологиями и средствами вычислительной техники, а также затраты на их приобретение и эксплуатацию, которые многократно используются для разных проектов;
- из общих затрат на аппаратуру и эксплуатацию технологических ЭВМ и отладочных стендов сложно выделить долю, которую необходимо включать в стоимость производства конкретного программного продукта.

Тем не менее, при заключении контрактов на производство продуктов и для оценки интегральных экономических характеристик комплексов программ приходится применять величины затрат *в денежном выражении* [7, 31]. Для этого вырабатываются соглашения (калькуляции) между заказчиком и разработчиком по преодолению перечисленных трудностей при анализе стоимости производства программных продуктов.

Суммарные затраты интеллектуального труда специалистов на производство программного продукта – трудоемкость, является основным интегральным экономическим показателем каждого программного проекта. Эти затраты подлежат оценке и минимизации при условии обеспечения заданных функциональных характеристик программного продукта и его качества. Полный анализ и оптимизацию суммарных затрат на проект целесообразно проводить *на всем жизненном цикле комплекса программ*. При этом, в ряде случаев, желательно учитывать затраты на сопровождение и на эксплуатацию программного продукта. Эти виды затрат характеризуются значительной неопределенностью из-за сложности прогнозирования длительности жизненного цикла, требований качества, степени модификации программ и затрат на сопровождение.

На **совокупную трудоемкость** при создании программного продукта влияет ряда факторов, при определении которых на практике используются различные единицы. Трудоемкость характеризуется временем производительного труда определенного числа специалистов, необходимого для создания программного продукта, его компонентов или выполнения определенного этапа работ в жизненном цикле. Такой подход привел к активному использованию единиц трудоемкости: человеко-день, человеко-месяц, человеко-год (при этом человеко-год предполагается состоящим в среднем из 250 рабочих человеко-дней с учетом выходных и праздничных дней). Подобные единицы трудоемкости позволяют сопоставлять затраты в разных организациях и даже в разных странах на аналогичные по размеру комплексы программ и не зависят от особенностей валюты, налогов и подобных экономических факторов при оценке стоимости. Эти единицы трудоемкости достаточно прочно вошли в практику планирования и оценки процессов производства, вследствие чего являются **базовыми в данной монографии**.

Длительность производства программного продукта зависит от многих факторов и, прежде всего, от его сложности. Технологический процесс создания любых комплексов программ включает ряд базовых этапов, которые **обязательно приходится реализовать** независимо от затрат. Каждый этап требует некоторого времени, что приводит для конкретных комплексов программ к относительно небольшим (по сравнению с затратами труда) вариациям суммарной длительности производства. Если разработка ведется на достаточно высоком технологическом уровне, то цикл производства сложного программного продукта принципиально трудно сокращать без ущерба для его качества. Поэтому даже при увеличении затрат труда в несколько раз длительность производства имеет тенденцию уменьшаться только на проценты.

Число необходимых специалистов и особенности их квалификации неравномерно распределяется по этапам жизненного цикла комплекса программ. На начальные этапы системного проектирования и завершающие этапы испытаний программного продукта требуется относительно небольшое число, но наиболее высококвалифицированных специалистов. На средних этапах разработки и тестирования программных модулей и компонентов в проекте участвует наибольшее число специалистов относительно невысокой квалификации.

Однако для оценок и прогнозирования экономических характеристик широко применяется среднее число специалистов. Ему соответствует **средняя производительность труда** коллектива при производстве полностью нового сложного комплекса программ, которая может служить **ориентиром** для сравнения эффективности труда при создании различных программных продуктов. Эта характеристика, конечно, различается для различных классов, размеров и других параметров комплексов программ, однако диапазон этих различий не столь велик как изменения размера, требований к качеству и других параметров.

Для прогнозирования экономических характеристик новых комплексов программ при проектировании и производстве необходимы исходные данные:

- обобщенные характеристики использованных ресурсов и экономические характеристики завершенных разработок – прототипов проектируемого комплекса программ, а также оценки влияния на их характеристики различных факторов продуктов и среды производства;
- реализованные и обобщенные перечни выполненных работ, и реальные графики проведенного ранее производства программных продуктов различных классов;
- цели и содержание частных работ в процессе производства сложных комплексов программ и требования к их выполнению для обеспечения необходимого качества программных продуктов в целом;
- структура и содержание документов, являвшихся результатом выполнения частных производственных работ.

Экономические характеристики производства реальных завершенных проектов программных продуктов собираются, накапливаются и обрабатываются с начала 80-х годов в разных отечественных организациях и за рубежом [3, 16]. Они позволили прогнозировать основные характеристики процессов производства сложных программных продуктов. При этом компоненты операционных систем, драйверы, средства контроля и тестирования, а также повторно используемые компоненты обычно не учитывались при оценке размера вновь созданных комплексов программ и экономических характеристик их производства. Поэтому характеристики производства этого периода можно отнести к полностью оригинальным разработкам

комплексов программ. Учитывались *все категории специалистов*, участвующих в создании программного продукта и обеспечивающих процессы разработки, а также все виды работ, связанные непосредственно с проектированием и производством программного продукта на выделенном интервале времени. Теоретические работы и системный анализ до подготовки требований в значениях экономических характеристик обычно не учитывались.

Наиболее важен анализ и учет факторов на начальных этапах производства, когда прогнозируются первичные совокупные затраты на создание комплекса программ. На этих этапах неопределенность оценки характеристик и факторов проекта наибольшая, однако применение прогнозируемых характеристик позволяет избегать крупных ошибок при оценке затрат, которые делаются экспертами без детального анализа влияния ряда важных факторов.

Только на базе серьезных статистических исследований экономических характеристик жизненного цикла и практического маркетинга программных продуктов *возможны были обобщения и создание теоретических и практических основ экономики производства сложных программных продуктов*. Эти задачи требовали для своего решения выполнения крупных, комплексных научно-исследовательских работ. Специалистам необходимо было исследовать и сформулировать методы экономического анализа, оценивания и прогнозирования необходимых ресурсов для проектов комплексов программ. Тем самым следовало определить *базовый фундамент всей экономики производства программных продуктов*. Такое выделение определялось тем, что без подобных исследований вряд ли возможно было последующее серьезное развитие экономики в этой области. Внимание было сосредоточено на концептуальной основе распределения затрат труда в процессе производства программных продуктов, на *факторах, определяющих реальные трудозатраты и другие экономические показатели*, а также на исследовании таких характеристик в реализованных проектах – рис. 3.1.

Прежде всего, необходимо было *изучение реальных экономических характеристик* проектирования и производства современных сложных программных продуктов, реализованных *прототипов* для прогнозирования организации и планирования новых проектов. Оценки экономических характеристик проектирования и производст-

ва сложных комплексов программ далее базируются на **двух типах** экспериментальных статистических исходных данных (см. главу 8).

Экономические характеристики для оценивания производства программных продуктов включают:

- статистические исследования экономики производства программных продуктов:
 - свойства и особенности основных экономических характеристик производства программных продуктов;
 - организацию и методики статистических исследований экономических характеристик программных продуктов;
 - основные составляющие затрат на производство программного продукта;
 - анализ влияния факторов на экономические характеристики производства программных продуктов;
 - примеры экономических характеристик производства программных продуктов;
- характеристики трудоемкости производства программных продуктов:
 - определение целей при оценивании трудоемкости производства комплексов программ;
 - факторы, которые целесообразно учитывать при оценке трудоемкости производства программных продуктов;
 - учет предела производительности, доступной разработчику – одиночке, строк новых программ в год;
- характеристики длительности разработки программных продуктов:
 - особенности фиксирования времени начала и окончания производства программного продукта;
 - определение границ возможного сокращения длительностей производства программных продуктов;
 - определение границ нерационально больших длительностей производства комплексов программ;
- вспомогательные характеристики производства программных продуктов:
 - среднюю производительность труда коллектива при производстве сложного комплекса программ;
 - оценки требуемого среднего числа специалистов для производства конкретного программного продукта.

Рис. 3.1

Первый тип основных исходных данных получается анализом и обобщением реальной экономики, результатов трудоемкости и длительности процессов создания предшествующих завершенных сложных комплексов программ. На базе этих данных строятся математические модели для прогнозирования обобщенных экономических характеристик новых проектов с учетом их конкретных факторов. Эти данные путем декомпозиции позволяют определять необходимое число различных специалистов для создания компонентов и сложных комплексов программ, а также их среднюю производительность труда в единицу времени.

Второй тип исходных данных получается регистрированием, анализом, обобщением и усреднением *индивидуальной производительности* труда конкретных специалистов при создании компонентов и комплексов программ. Практически для оценок реальных экономических характеристик проектов часто используется стоимость (например, в долларах) разработки одной строки текста программы. Суммирование этих данных по совокупности модулей и компонентов позволяет обобщать и прогнозировать экономические характеристики сложных комплексов программ в зависимости от их размеров и характеристик.

В некоторых случаях *ориентиром затрат* (стоимости) на создание особенно сложных программных продуктов реального времени (миллионы строк текста) высокого качества, надежности и безопасности применения, может служить *соизмеримая* с ними стоимость аппаратуры, на которых они функционируют (например, бортовых вычислительных систем в авиации). При этом каждая строка текста программ в таких сертифицированных комплексах может превышать сотню долларов, а средняя производительность труда специалистов в слаженных высококвалифицированных коллективах, составлять только около одной строки программы в день.

Рассматривались преимущественно средние и крупные проекты, создаваемые большими коллективами специалистов. В таких проектах на чистое творчество и научные исследования отдельных специалистов, (преобладающие в небольших индивидуальных разработках), накладывается множество технических работ, характерных для индустриального проектирования и производства программных продуктов. Вследствие этого значительно *нивелировались индивидуальные особенности и квалификация отдельных специалистов*, и появля-

лась возможность оценивать усредненные характеристики производительности труда и другие экономические характеристики производства в больших коллективах. Анализируемые интервалы времени производственных процессов были ограничены: в основном от этапа оформления требований технического задания на разработку проекта и до этапа завершения испытаний программного продукта, соответствующего требованиям заказчика. Тем самым только попутно рассматривались процессы сопровождения, тиражирования, документирования, внедрения и применения программных продуктов. Процессы формирования рыночной стоимости программных продуктов и маркетинговые исследования в области экономики создания, распространения и продажи массовых промышленных программных средств в те годы обычно не учитывались.

Индустриальное, коллективное создание сложных программных продуктов необходимо было структурировать, планировать и поэтапно регламентировать с учетом заданных ограничений сроков и стоимости проектов. Однако обычно у специалистов *отсутствовали количественные оценки экономических характеристик* на основе статистических данных реально созданных программных продуктов. Задача состояла в том, чтобы детально исследовать реальные экономические показатели достаточного набора завершенных проектов комплексов программ и создать на этой статистической базе *методику прогнозирования* трудоемкости, длительности и числа необходимых специалистов по этапам работ и интегрально по программным продуктам разных классов и размеров.

Наиболее подробно *основные закономерности и влияние факторов на экономические характеристики* процессов производства сложных программных продуктов в 80-е годы исследовались *за рубежом*. В 1981 году на основе исследования процессов разработки *63 проектов* была опубликована модель прогнозирования экономических характеристик под названием КОМОСТ [3]. В последующем эта модель была развита, детализирована и опубликована как СОСОМО, а в *2000 году* под названием СОСОМО II [38] (см. главы 8 и 9). В этой модели на основе анализа более *160 реальных проектов* разработки программных продуктов различной сложности уточнены рейтинги влияния выделенных факторов на основные экономические характеристики производства. Однако методика сбора статистических данных и детальные характеристики анализировавшихся программных

продуктов не опубликованы. Обобщенные данные этих работ ниже используются и рекомендуются как базовые для прогнозирования затрат при создании сложных программных продуктов.

Методическим примером статистических исследований экономических характеристик программных продуктов являются данные, полученные в **НИР ПРОМЕТЕЙ** [22]. С этой целью были разработаны методические указания и анкета, разосланные в ряд предприятий для сбора сведений о каждой завершённой **промышленной разработке программного продукта** [16]. **В анкетах предлагалось регистрировать:**

- технические (не функциональные) характеристики и размер объекта разработки;
- трудовые и временные ресурсы, непосредственно затраченные на проектирование и производство программного продукта;
- программную, инструментальную оснащённость производства средствами автоматизации технологии;
- аппаратную оснащённость производства комплекса программ средствами вычислительной техники.

При этом фиксировались **основные составляющие затрат на производство программного продукта:**

- трудоемкость и длительность непосредственной разработки программного продукта;
- среднее число специалистов, участвовавших в проекте;
- затраты на программные, инструментальные системы автоматизации разработки программного продукта;
- затраты на технологическую аппаратуру ЭВМ, используемую в процессе разработки комплекса программ.

При расчете производительности труда разработчиков учитывался **труд только на непосредственное производство функциональных компонентов комплекса программ**, а при расчете стоимости разработки в целом и одной команды – все перечисленные составляющие затрат. Относительно небольшие затраты на системы автоматизации производства комплексов программ не учитывались, так как были обусловлены многократным применением одних и тех же инструментальных средств для создания различных проектов.

Представленные примеры экономических характеристик производства комплексов программ наиболее систематично были собраны, обработаны и проанализированы, от поступивших с 30 предпри-

ятий оборонной промышленности. Они отражали данные о **231 программном продуктах** трех классов общим объемом свыше 16 млн. команд. Из них выборки по классам комплексов программ (см. главу 4) составлялись для:

- класса СРВ **142 проекта**, объемом около 10,5 млн. команд;
- класса АС – **29 разработок** объемом около 1,5 млн. команд;
- класса ППП – **60 проектов** объемом около 4 млн. команд.

Таким образом, выборки данных о комплексах программ классов СРВ и ППП были достаточно представительными для достоверных статистических выводов по основным экономическим характеристикам их разработки. Характеристиками проектов являлись как абсолютные экономические показатели (размер, трудоемкость и длительность разработки), так и относительные (производительность труда разработчиков, стоимость разработки одной команды программного продукта). Относительные экономические характеристики позволили сопоставить различные разработки программных продуктов и коллективы разработчиков предприятий по эффективности их труда.

За меру производительности труда разработчиков принималось отношение размера поставляемого программного продукта в командах **нового исходного текста программ** (без учета заимствования из предшествовавших проектов) к трудозатратам на его разработку в человеко-днях. Учитывался только размер программного продукта, прошедшего испытания и переданного на эксплуатацию потребителю без компонентов и версий, разработанных, но не вошедших в окончательный программный продукт. В число специалистов – разработчиков включались основные категории специалистов: руководители разработки, аналитики, системотехники, программисты, тестировщики, а также технический персонал, занятый оформлением текущей и итоговой документации.

В результате обработки фактических статистических данных были получены **примеры зависимости экономических характеристик от основных факторов**, выявлено изменение их во времени, роль применения различных языков программирования, методов доступа к ЭВМ, определена реальная обеспеченность разработчиков дисплеями и машинным временем. По классу СРВ в **1986 году** производительность труда специалистов в среднем возросла до 2,6 команд

на человека в день по сравнению с 1,6 в *1980 году*. Из общего объема производства программных продуктов около 23% составляли средние комплексы программ (объемом от 30 тыс. до 100 тыс. команд). К *1986 году* крупные комплексы (объемом более 100 тыс. команд) составляли 32%. Данные по производительности труда программных продуктов реального времени в *80-е годы*, были сопоставлены с аналогичными данными в США [3]. При выборке *примеров* проектов с приблизительно одинаковыми характеристиками типов и размеров, производительность труда отечественных специалистов в то время оказалась *близка* к доступным аналогичным данным специалистов за рубежом.

Важнейшей экономической характеристикой производства программных продуктов является ее *продолжительность*. В *1985 году* средняя длительность разработки сложного комплекса программ была: в классе СРВ – 4 года, в классе АС – 2,7 года, в классе ППП – 3,2 года. По сравнению с *1980 годом* длительность сократилась в среднем на 30%. Факторами, снижающими длительность разработки, явились:

- повышение производительности труда разработчиков программных продуктов;
- применение готовых переносимых программных компонентов.

Выявлено, что наиболее эффективным направлением для улучшения экономических характеристик производства являлось применение готовых программных компонентов. Заимствование 50 % готовых программных компонентов (от общего размера) позволило сокращать длительность разработки, в среднем, на 25%, а при заимствовании в 80% – в два раза.

Приведенные выше данные следует рассматривать только как методологические *примеры исследований* экономических характеристик производства программных продуктов, когда они были актуальны и служили *ориентирами* при планировании реальных проектов. Более современные данные можно получить, используя модель СОСОМО II [38] (см. главы 8 и 9). Однако реальные экономические характеристики производства программных продуктов публикуются очень скупо и оперативно не обобщаются, по-видимому, они считаются коммерческой тайной предприятий, которым не выгодно раскрывать фактические недостатки и неэффективность своих производств.

Характеристики трудоемкости производства программных продуктов

Обеспечение функциональной пригодности является основной целью при использовании финансовых, трудовых, вычислительных и других ресурсов в жизненном цикле комплексов программ. Однако это не значит, что затраты на решение этой основной задачи всегда являются доминирующими по величине. Необходимость выполнения ряда требований к остальным, конструктивным характеристикам качества (например, надежности), часто приводит к тому, что использование ресурсов на их реализацию может превышать базовые затраты на обеспечение только функциональной пригодности. В то же время, затраты на выполнение этих требований, всегда направлены на повышение и совершенствование функциональной пригодности. Поэтому обычно трудно четко выделить и количественно оценить все виды затраты, используемые **только на реализацию функциональной пригодности**.

При первичной оценке **трудоемкости и стоимости программного продукта** предстоит выполнить два основных этапа:

- первый этап состоит в **оценивании размера – масштаба комплекса программ**;
- на втором этапе представление об этих размерах наряду с информацией о других факторах среды производства служит для **оценивания экономических характеристик, трудозатрат и стоимости производства**.

Уточнение размеров создаваемого комплекса программ должно предшествовать этапам проектирования и кодирования программ, выполняемым с целью реализации требований на практике. **При оценках трудоемкости производства программного продукта** целесообразно учитывать:

- цели оценивания трудоемкости должны быть согласованы с потребностями в объеме и точности информации, способствующей принятию решений на соответствующем этапе проекта;
- достоверность оценок трудоемкости должны быть сбалансированы для различных компонентов системы, и величина уровня неопределенности для каждого компонента должна быть примерно одинаковой, если в процессе принятия решения все компоненты имеют одинаковый вес;

- следует возвращаться к предшествующим целям оценивания и изменять их, когда это необходимо для ответственных бюджетных решений, принимаемых на ранних этапах и влияющих на последующие этапы экономического анализа проекта.

В любом программном комплексе можно выделить компоненты, в которых сосредоточены функциональные алгоритмы и программы, предназначенные для решения основных *целевых задач*. В некоторые из них органически входят компоненты для повышения качества решения функциональных задач или они построены с учетом требований высокого качества результатов основных функций. При анализе затрат ресурсов в жизненном цикле комплексов программ целесообразно разделить на две части:

- затраты на создание программных компонентов, обеспечивающих *базовые свойства функциональной пригодности* комплекса программ для его применения по прямому назначению пользователями, в соответствии с требованиями контракта и технического задания;

- составляющие *дополнительных затрат*, обеспечивающие требуемые конструктивные характеристики качества для улучшения функциональной пригодности продукта в соответствии с целями и сферой его применения.

Кроме того, следует учитывать совокупность затрат ресурсов на технологию, инструментарий автоматизации производства и систему качества, обеспечивающие жизненный цикл. Эти составляющие затрат зависят не только от характеристик проектируемого комплекса, но и от интенсивности применения технологических средств.

Основным ресурсом при производстве сложных комплексов программ являются допустимые *трудоzатраты* на производство программного продукта с требуемым качеством, а также необходимое и доступное *число специалистов* соответствующей квалификации. Потребность в этих ресурсах в наибольшей степени зависит от размера – масштаба и сложности разрабатываемого продукта (см. главу 4). Когда впервые рассматривается масштаб нового проекта, интуитивные и экспертные оценки его трудоемкости могут отличаться от конечного значения примерно в полтора – два раза в ту или другую сторону. Такая достоверность оценок обусловлена уровнем неопределенности на данном этапе знаний о конечном содержании и возможном размере программного продукта. Общая

тенденция состоит в том, что на начальных этапах оценки затрат чаще всего занижаются.

Эта неопределенность уменьшается, по мере детализации и углубления содержания и функций проекта, как только фиксируются конкретные принципы функционирования и концепция комплекса программ. На этом этапе оценка достоверности размера и трудоемкости уменьшается приблизительно до 40%. Это вполне объяснимо, поскольку еще не уточнены структура и многие детали продукта. Эти вопросы могут быть разрешены во время разработки структуры и спецификаций требований к комплексу программ и тогда можно оценить его размер с точностью до 15 – 20%. После завершения разработки и подтверждения проектных спецификаций при детальном проектировании комплекса программ может быть определена структура внутренних данных и функции программных компонентов. На этом этапе, ошибки оценки размера и трудоемкости проекта могут составить около 10%. **Неопределенности оценок трудоемкости** могут быть обусловлены: особенностями конкретных алгоритмов; управления их функционированием; обработкой ошибок; инициализацией и завершением сеансов решения задач. Уточнения размеров комплекса и компонентов могут быть решены к концу детального проектирования, однако при этом сохраняется неопределенность оценки размера комплекса программ и его трудоемкости порядка 5 – 10%, связанная с тем, насколько хорошо специалисты понимают спецификации требований к компонентам, в соответствии с которыми они должны создавать их программы.

На начальных этапах производства для прогнозирования его трудоемкости и суммарных затрат необходимо применять рациональные гипотезы об особенностях жизненного цикла конкретного комплекса программ. Даже приблизительный учет распределения вероятных затрат на этапах жизненного цикла позволяет более эффективно использовать экономические ресурсы при создании программного продукта. При этом условия, обеспечивающие минимум суммарных затрат на всем жизненном цикле, могут приводить к необходимости принципиального изменения предполагаемого процесса производства программ, при котором достигается минимум только на интервале производства первой версии продукта. В связи с этим последующий анализ суммарных затрат на разработку может прово-

даться в двух постановках задачи: **автономно** на интервале производства первой версии программного продукта без учета последующего его использования и развития, и **комплексно** с учетом влияния затрат на его эксплуатацию и сопровождение, что должно оговариваться в каждом случае.

В общем случае желательно достигать **сбалансированный набор целей оценивания трудоемкости** производства комплекса программ, который бы давал примерно одинаковую величину уровня неопределенности для всех исходных данных и компонентов комплекса. Вследствие этого изложение анализа и прогнозирование экономических характеристик ниже разделено на две крупные части. В главе 8 рассматриваются методы и модели для оценок трудоемкости, длительности и числа специалистов с учетом размера, повторного использования компонентов и ряда факторов. В главе 9 прогнозируются экономические характеристики, и оценивается влияние на них дополнительных факторов: особенностей и качества объектов и среды разработки, квалификации и организации специалистов, технологической, инструментальной и аппаратурной оснащенности проекта.

По мере выполнения проекта, экономические оценки комплекса программ необходимо пересматривать и изменять, когда это становится выгодным или необходимым. Можно начинать с определения оценок трудоемкости в соответствии с точностью спецификаций требований с учетом минимума факторов, но при дальнейшем анализе уточнять оценки с точностью до деталей функционирования и с учетом влияния совокупности важнейших факторов и характеристик проекта.

Специалисты при производстве программ **различаются квалификацией и степенью участия** в непосредственной разработке программного продукта. Встречаются особо талантливые программисты, способные создавать очень быстро программные компоненты высокого качества. Однако при любых способностях **есть предел**, доступный разработчику – одиночке, и он редко превышает 10 тыс. строк новых программ в год. Кроме того, даже в этом случае такому программисту необходима помощь при составлении тестов, оформлении документации, испытаниях и ряде других работ, обеспечивающих превращение программы в **программный продукт**.

При разработке сложных комплексов программ пишут и отлаживают программы модулей и компонентов только около половины

общего числа лиц, затраты на которых необходимо учитывать при экономическом анализе. Остальные специалисты осуществляют руководство проектом, проводят системный анализ, исследуют алгоритмы, оформляют документацию, выполняют вспомогательные технические работы. Их труд направлен либо полностью на создание определенного комплекса программ, либо распределяются между несколькими проектами. Если специально не оговаривается, то далее учитывается трудоемкость на программный продукт **всех категорий специалистов** в соответствии с их долей участия в создании конкретных программ определенного проекта.

В ряде исследований трудоемкости **размер базы данных** либо совсем не учитывается, либо включается в объем программного продукта. Этому способствует также развитие теории сложности в направлении анализа функциональной сложности комплексов программ при почти полном игнорировании сложности данных и их влияния на экономические характеристики. В статистической теории сложности программ [42] показано, что для программных модулей и относительно небольших групп программ может быть большая корреляция между числом имен переменных и числом операторов в программе. Однако для сложных комплексов эта корреляция невелика, что определяет целесообразность **разделять анализ таких комплексов программ на два типа**. Первый тип – осуществляет, преимущественно, логическую обработку относительно небольшого потока данных, а второй тип – программы административных и информационно-поисковых систем при наличии больших баз данных и относительно простой их логической обработке. Для отражения экономических характеристик производства этих типов программ может использоваться отношение числа имен переменных к числу строк или операторов текста программ. Для первого типа комплексов это отношение не превышает десяти процентов, а для второго оно может значительно превышать единицу и достигать десятков и сотен. Затраты при производстве комплексов второго типа оказываются зависящими от размера базы данных, который влияет на сложность взаимодействия программ с данными.

Хотя размеры баз данных для сложных комплексов программ по числу типов имен переменных изменяются в очень широких пределах от 10^3 до 10^8 , их непосредственное влияние **на трудоемкость разработки строки текста** в программе даже при числе пере-

менных, в десятки раз превышающем размер программы, может оцениваться на уровне 10% [3]. Степень этого влияния трудно формализовать, так как большую роль играет структура базы данных и ее функциональное назначение.

Характеристики длительности производства программных продуктов

Для учета затрат времени коллектива специалистов на конкретный комплекс программ особенно сложно фиксировать *начало разработки*. Дело в том, что системный анализ зачастую входит в научно-исследовательские работы, финансируемые, планируемые и учитываемые независимо от начала производства конкретного программного проекта. В ряде случаев первичная разработка концепции комплекса программ является обобщением опыта создания и эксплуатации ранее разработанных, унаследованных программ. Системный анализ, исследования и моделирование алгоритмов иногда в последующем реализуются в нескольких комплексах программ, и весьма трудно оценить долю экономических затрат исследовательских работ, которую следует отнести к каждому конкретному проекту. Перечисленные обстоятельства привели к тому, что затраты на системный анализ и предварительные исследования могут различаться на один – два порядка и обычно не включаются в совокупные экономические затраты на производство конкретного программного продукта, что соответствует практике выделения и отдельного учета финансирования на научно-исследовательские работы в промышленности. При последующем изложении началом разработки считается создание технического задания или спецификации требований, согласуемых с заказчиком и будущими пользователями продукта.

Окончанием разработки для прекращения учета затрат при оценке трудоемкости и длительности конкретного производства продукта далее принимается момент завершения межведомственных или государственных испытаний программного продукта и оформления акта соответствующей комиссией заказчика. Однако при анализе реальных проектов эта граница оказывается тоже размытой, как и начальная, хотя и в меньшей степени. Это обусловлено разнообразием видов испытаний, возможностью проведения испытаний по отдель-

ным функциональным задачам, различием методов формализации передачи продукта на эксплуатацию.

Ряд факторов и неопределенность достигаемого качества программ приводят к тому, что влияние затрат на длительность разработки имеет размытую характеристику. При изменении размеров сложных комплексов и трудоемкости в широком диапазоне длительность разработки может изменяться мало по сравнению с величиной затрат. Для каждого размера комплекса программ при заданном качестве существует **область невозможного сокращения длительности производства**, которую не удастся преодолеть при увеличении затрат труда – рис. 3.2.

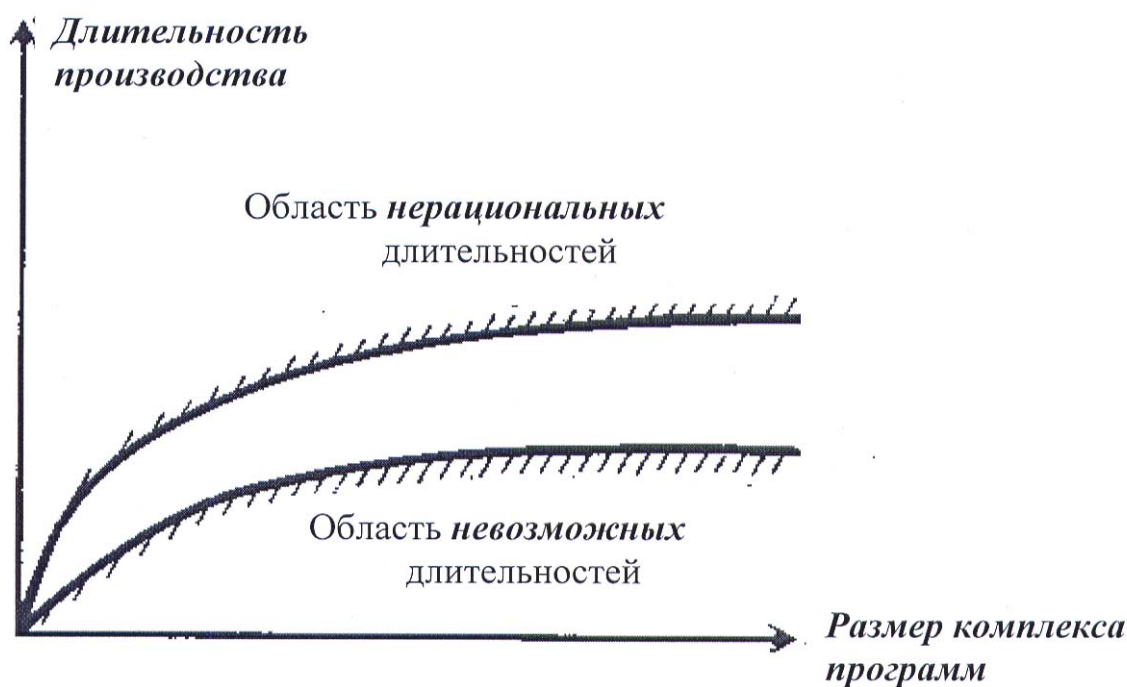


Рис.3.2

Для планирования разработки комплекса программ и регулярного управления этим процессом необходимы частные экономические характеристики в зависимости от различных факторов. Такие показатели могут формироваться: по этапам разработки, на единицу продукции, как относительные затраты на достижение заданной характеристики качества программ или как составляющие суммарных затрат в жизненном цикле комплекса программ.

Длительность производства программных продуктов является важнейшей экономической характеристикой, поскольку часто она определяет общие сроки производства систем, а значит, быстроту реализации идей и методов в различных областях автоматизации.

Диапазону размеров современных программных продуктов в три-четыре порядка (до 10 млн. строк) соответствуют приблизительно такие же диапазоны относительного изменения трудоемкостей и стоимостей их производства. Однако принципиально нерентабельно производство даже очень сложных программных продуктов более 3 – 5 лет. С другой стороны, комплексы программ даже в несколько тысяч строк по полному технологическому циклу с испытаниями и документацией как продукции, редко создаются за время, меньшее, чем полгода – год. Таким образом, вариация длительностей производства программных продуктов много меньше, чем вариация их трудоемкостей, и не превышает десятикратный диапазон. Практически длительность производства программных продуктов *ограничена сверху и снизу* и одним из основных факторов, определяющим эти границы, является размер комплексов программ.

Относительный *«консерватизм»* значений длительностей по сравнению с трудоемкостью, определяется объективной необходимостью создавать программные продукты в рациональные сроки [3, 18]. Любые программные продукты должны поступать на эксплуатацию до того, как в них пропадет необходимость. Их цели, концептуальная основа и алгоритмы *не должны устареть* за время проектирования и производства. Отсюда появляется *верхний предел* допустимых длительностей производства. Этот верхний предел не может иметь единственное значение для любых классов и размеров программных продуктов. Однако недопустима его вариация в том же диапазоне, что размера и трудоемкости программного продукта. Поэтому на практике по мере возрастания размеров комплексов программ, увеличиваются коллективы специалистов-разработчиков, что обеспечивает основной прирост необходимой трудоемкости. Чем крупнее создаваемый комплекс программ, тем большие усилия обычно прилагаются для автоматизации и совершенствования технологии производства. Это также способствует замедлению роста длительностей разработки, однако по мере увеличения сложности программ, длительность их производства все же заметно возрастает.

Стремление ограничивать длительность производства программных продуктов приводит к объективному формированию верхнего предела, за которым распространяется зона *«нерациональных»* длительностей, зависящих от размера и трудоемкости комплекса программ (см. рис. 3.2). Даже для довольно сложных программных

продуктов, имеющих размер *свыше* 500 тыс. строк, вряд ли допустима длительность разработки более 3 лет. Большие длительности, иногда имеющиеся на практике, обусловлены в основном низкой квалификацией разработчиков и заказчиков, недостаточной автоматизацией технологии, малым коллективом специалистов и рядом других, преимущественно организационных и технологических причин. Аналогичные ситуации чаще встречаются при относительно небольших проектах (50 – 100 тыс. строк), когда у руководителей и коллектива мал опыт их проведения, следствием чего является *избыточный «оптимизм»* в начале разработки, а также пренебрежение технологией и организацией работ.

Границу снизу определяют естественный технологический процесс коллективного производства и необходимость выполнения ряда скоординированных работ на последовательных этапах, которые обеспечивают получение продукта требуемого качества. Подготовка текстов программ, их тестирование, комплексирование, документирование и испытания могут проводиться только последовательно, и каждый этап требует определенного времени. Попытки форсировать отдельные этапы работ, как правило, приводят к увеличению продолжительности других этапов. Подключение дополнительных специалистов увеличивает затраты на разработку и только в небольших пределах дает сокращение сроков. В некоторых случаях увеличение числа специалистов *может давать обратный эффект* – длительность производства увеличивается вместе с увеличением трудоемкости, вследствие роста затрат на взаимодействие специалистов.

Под воздействием перечисленных факторов формируется объективный минимум длительностей – граница снизу области *«невозможного»*, зависящая, в первую очередь, от размеров разрабатываемых комплексов. Нижняя граница длительностей еще более *«консервативна»*, чем верхняя. Изменение этой границы возможно в небольших пределах только за счет совершенствования технологии, повышения программной и аппаратурной оснащенности разработки, а также роста коллективной квалификации разработчиков и заказчиков.

Практическая граница *«нерациональных»* длительностей имеет значения, приблизительно вдвое большие, чем значения границы *«невозможных»* длительностей, при том же размере программного продукта. Это означает, что даже большие усилия по автоматизации и

организации производства комплексов программ приводят к сокращению длительностей только в 2 – 3 раза, в то время, как трудоемкость уменьшается значительно больше. По результатам реальных проектов может быть оценена средняя или наиболее вероятная, длительность производства программных продуктов определенного класса и размера при заданных условиях. Конкретное распределение длительностей, можно оценивать на основе исходных данных, имеющих в базе данных экономических характеристик завершенных разработок предприятия, и от метода их усреднения.

Для конкретного планирования длительностей производства программных продуктов определенных классов необходимо для каждого предприятия исследовать и обобщать экономические характеристики реальных разработок, однородных по технологии и другим условиям. Такие обобщения при конкретных условиях производства позволяют получить опорные значения длительностей для некоторых размеров программ. Эти абсолютные значения могут быть использованы для расчета коэффициентов регрессии с целью прогнозирования длительностей производства на базе выявленных закономерностей и реальных опорных значений для конкретных условий проектов (см. главу 9).

Вспомогательные характеристики производства программных продуктов

Экономические характеристики производства на единицу размера программной продукции можно оценивать с использованием унифицированной единицы измерения – оператора (команды) на строку текста программы (ЛОС). Усредненные затраты труда всех категорий специалистов, участвующих в производстве программного продукта определенного размера, позволяют оценить *среднюю производительность труда при его проектировании и производстве*. В качестве единицы измерения этого показателя ниже используются число операторов ассемблера в месяц на человека или число строк текста программы в месяц на человека для комплексов программ на языках высокого уровня. При этом следует подчеркнуть интегральный характер всех величин, используемых при расчете этой характеристики. Этот показатель особенно важен при сопоставлении экономических характеристик производства продуктов различных классов и размеров, а также для оценки эффективности технологий и средств

автоматизации разработки программ. Средние **трудозатраты на создание каждой команды** (оператора) в комплексе программ соответствуют обратной величине производительности труда при разработке, измеренной в операторах или числе строк программы на один человеко-месяц.

Часто при оценке производительности труда разработчиков программ учитываются трудозатраты только непосредственных участников процесса проектирования и производства без трудозатрат на эксплуатацию технологических ЭВМ и отчислений на амортизацию вычислительной техники. Эти затраты могут быть весьма значительными, однако их не всегда удобно выражать трудоемкостью на строку текста или команду. Поэтому в качестве экономической характеристики на единицу продукции, с учетом всех видов затрат иногда применяется **стоимость одного оператора или строки текста программы** в завершенном производстве и испытанном программном продукте, представленная в денежном выражении (руб. или долл. на команду или строку текста программы).

Экономические характеристики этапов производства программного продукта целесообразно оценивать аддитивными экономическими показателями. Такими характеристиками могут служить суммарные трудозатраты на выполнение этапа работ при планировании и создании комплекса программ определенного размера и класса или поэтапные трудозатраты на одну интегральную команду – строку текста программы. Эти характеристики позволяют выявлять наиболее трудоемкие этапы и помогают рациональнее распределять затраты по этапам работ. В поэтапных затратах целесообразно выделять совокупные затраты на средства автоматизации разработки, что позволяет выявлять эффективные компоненты технологии и инструментальных средств с учетом стоимости их приобретения и эксплуатации.

Во многих случаях важны не столько экономические затраты на создание программного продукта сколько **длительность разработки**. Локальное ускорение отдельных этапов разработки (особенно начальных) может приводить к значительному увеличению длительности других этапов и к общему возрастанию длительности производства. Поэтому совершенствование технологии и средств автоматизации производства сопряжено с перераспределением затрат и длительностей этапов работ с целью сокращения общей длительности разработ-

ки проекта, в некоторых случаях даже за счет увеличения суммарных затрат.

Этапы жизненного цикла комплексов программ существенно различаются между собой степенью определенности и прогнозируемости их характеристик. Соответственно изменяются возможности подготовки и **достоверность планов** проведения работ. В процессе разработки сложных программных комплексов уточняются и детализируются их спецификации требований, функции, структура и характеристики компонентов. Поэтому на начальных этапах, особенно принципиально новых продуктов, трудно спланировать точно все последующие этапы. В результате планирование проводится итерационно в соответствии с последовательным повышением достоверности и точности сведений о характеристиках продукта и среды производства.

Производство сложных программных продуктов, особенно на начальных и завершающих этапах, характеризуется **высокой долей творческого труда**. Поэтому трудоемкость и длительность отдельных операций и частных работ существенно зависят от индивидуальных особенностей и квалификации их исполнителей и характеристик конкретного проекта. Вследствие этого отсутствует достоверная пооперационная статистика разработки комплексов программ, и пока **невозможно создать типовые нормативы** на большинство частных операций при их создании. Отсюда принципиальной особенностью планирования разработки комплексов программ должно являться активное участие руководителей и заказчиков проекта в составлении планов на базе исследованных характеристик прототипов завершенных разработок и их личного опыта. Такие планы должны иметь разумные ограничения в детализации работ на уровне, обеспечивающем необходимую управляемость всего процесса производства. В процессах и системах автоматизации планирования и управления разработкой комплексов программ **целесообразно учитывать следующие их особенности**:

- использование прототипов экономических характеристик, перечней и графиков частных работ как основных исходных данных для прогнозирования и планирования производства новых программных продуктов;
- последовательную, иерархическую детализацию и поэтапное уточнение планов и прогнозов в соответствии с повышением полноты

и достоверности исходных данных о продукте и среде разработки, получаемых в процессе производства;

- целесообразность и возможность выбора первичного прототипа перечня работ, достаточно адекватного исходным данным проектируемого комплекса, и возможность его уточнения проектировщиком;
- регистрацию, обобщение и хранение реализованных рабочих планов и значений экономических характеристик для их использования в качестве прототипов при планировании аналогичных проектов.

В процессе производства, после использования прототипов для прогнозирования и планирования очередного проекта, происходит его реализация, данные которой могут быть использованы в качестве расширения прототипов для последующих проектов. Тем самым может быть создана *последовательно уточняющаяся база данных*, позволяющая повышать достоверность прогнозирования и планирования производства программных продуктов определенного класса.

Оценка требуемого среднего числа специалистов для производства конкретного программного продукта предварительно может быть рассчитана путем деления оценки величины трудоемкости на длительность производства. Однако рациональное число специалистов, участвующих в производстве распределяется не равномерно по этапам работ. Поэтому целесообразно определять число и квалификацию необходимых специалистов с учетом этапов производства комплексов программ.

При разработке программных модулей и компонентов отдельными специалистами или небольшими группами, производительность труда при написании одних и тех же текстов автономных программ, может различаться в десяток раз в зависимости от их таланта и трудоспособности и достигать тысячи строк за человеко-месяц. Однако достаточно полное тестирование, документирование, комплексирование и интегрирование в крупные программные продукты, приводят к снижению интегральной производительности до величин в несколько сотен строк текста за человеко-месяц. Для *примера* крупных проектов класса СРВ в 80-е годы [3] приводятся величины 100 – 150 строк на человеко-месяц, в отечественных проектах в те же годы [16] эта величина приближалась к 100. При диапазоне изменения размеров программ СРВ на четыре порядка средняя производительность труда обычно изменяется только в два раза, что в ряде случаев существенно

облегчает упрощенные оценки и прогнозирование экономических характеристик. Совершенствование технологии, квалификации специалистов и инструментальных средств автоматизации разработки позволили в те годы повысить среднюю производительность труда при создании полностью новых оригинальных программных продуктов СРВ в несколько раз, по экспертным оценкам до величин 300 – 500 строк на человеко-месяц.

При производстве комплексов программ необходимо учитывать, что экономические, временные, вычислительные и другие ресурсы *на весь ЖЦ комплекса*, всегда ограничены и используемые затраты для улучшения каждой характеристики должны учитывать эти ограничения. Для рационального распределения этих ресурсов необходимо знать *как отражается изменение затрат на улучшении каждой характеристики качества программного продукта*. Эта корреляция затрат ресурсов и значений каждой характеристики качества зависит от назначения, а также от ряда свойств и других особенностей комплекса программ, что усложняет учет влияния таких связей. Тем не менее, выявлены основные тенденции такого взаимодействия, которые могут служить *ориентирами* при выборе и установлении требований к определенным характеристикам качества программных продуктов.

Характеристики ошибок при разработке комплексов программ значительно влияют на затраты при их создании. Поэтому целесообразны оценки относительных и абсолютных трудозатрат на выявление и устранение дефектов и ошибок [17, 34]. Общие тенденции состоят в быстром росте затрат на выявление каждой ошибки по последовательным этапам производства комплексов программ. Поэтому *экономически целесообразно* в максимальной степени выявлять ошибки на *начальных этапах ЖЦ* комплексов программ. Это, естественно, приводит к увеличению затрат и длительности этих этапов, однако способствует минимизации суммарных затрат и длительности производства программного продукта в целом.

Не выявленные ошибки непосредственно связаны с достигаемой корректностью, безопасностью и надежностью функционирования программных продуктов. Изучение характеристик ошибок при разработке реальных комплексов программ позволило создать ряд математических моделей, обеспечивающих возможность прогнозирования длительности разработки и затрат, необходимых для достижения оп-

ределенной безопасности и надежности программного продукта [17, 42, 50]. Анализ и обобщение характеристик выявленных и устраненных ошибок в процессе разработки позволяет контролировать и прогнозировать качество комплексов программ при аналогичных разработках.

Стремление заказчиков резко ускорить разработку, снизить затраты или нерационально увеличить нормативы для специалистов, всегда сопряжено со снижением качества. Имеющиеся попытки введения заказчиками и менеджерами нормативов на экономические характеристики производства для разработчиков либо не способствуют выполнению их управляющей и регламентирующей роли (если они занижены), либо приводят к снижению качества программ (если они завышены). Поэтому приводимые далее экономические характеристики следует использовать с учетом всех условий, для которых они получены, и **нецелесообразно применять в качестве нормативов** при конкретных проектах. Они могут служить **только ориентирами** для оценки и обоснования экономических характеристик проектирования и производства аналогичных программных продуктов.

Глава 4

СЛОЖНОСТЬ И РАЗМЕР – ОСНОВНЫЕ ФАКТОРЫ, ОПРЕДЕЛЯЮЩИЕ ЭКОНОМИЧЕСКИЕ ХАРАКТЕРИСТИКИ ПРОИЗВОДСТВА ПРОГРАММНЫХ ПРОДУКТОВ

Основные факторы, определяющие сложность производства программных продуктов

Экономические характеристики производства программных продуктов зависят, прежде всего, от сложности алгоритмов, размера комплекса программ и базы данных, которые определяют затраты труда и длительность полного цикла их производства. Основные затраты при проектировании и производстве программных комплексов приходятся на овеществленный, преимущественно, *интеллектуальный труд* специалистов различных категорий. Труднее всего прогнозировать *экономические характеристики производства программного продукта в начале проекта*, когда еще не сформировались достаточно четкие представления о функциях, свойствах, алгоритмах и *размере* программного комплекса, подлежащего разработке. На базе этих оценок желательно делать общий вывод, стоит ли заниматься данным проектом в дальнейшем, и на каких условиях следует заключить контракт на его выполнение. Когда разработка программного продукта близится к завершению, с целью уточнения экономических характеристик целесообразно учитывать некоторые дополнительные аспекты и спецификации. Однако общую экономическую смету, время работы над проектом и необходимые трудозатраты необходимо оценивать как можно раньше. При этом целесообразно поэтапно рассматривать ряд факторов, влияющих на экономические характеристики производства программного продукта – рис. 4.1.

Сложность и размер – основные факторы, определяющие экономические характеристики производства программных продуктов, включают:

- основные факторы, определяющие сложность производства программных продуктов:
 - обобщенные характеристики размеров комплексов программ в соответствии со стандартом ISO 12182;
 - особенности экономических характеристик производства основных классов программных продуктов;
 - концептуальные требования при формировании архитектуры, компонентов и их связей в комплексах программ;
- единицы измерения сложности и размера комплексов программ для экономического анализа их производства:
 - сложность проектирования и производства, а также сложность функционирования и применения продукта;
 - тексты программ, которые анализируются человеком и тексты которые размещаются для исполнения в ЭВМ;
 - обобщенные характеристики сложности компонентов и комплексов программ;
 - временная, информационная и программная сложность комплексов программ;
- масштаб – размер комплексов программ по числу строк текста и объему занятой памяти ЭВМ:
 - размер программы – сложность, основной ориентир трудоемкости производства программного продукта;
 - память, занятая программами в ЭВМ – программная сложность, ориентир размера комплекса программ;
 - преимущества и недостатки размера комплекса программ в строках текста программного кода (LOC);
- измерение размера – масштаба комплексов программ по числу функциональных точек:
 - стандартизация методов функциональных точек для измерения размера – сложности комплексов программ;
 - число и типы элементов функциональных точек и технология определения размера комплексов программ.

Рис. 4.1

К основной группе факторов, *отражающихся на экономических характеристиках производства программного продукта*, относятся:

- масштаб – размер комплекса программ, выраженный числом строк текста, функциональных точек или количеством программных модулей и компонентов в комплексе;
- количество обрабатываемых переменных или объем памяти, используемой для размещения базы данных;
- трудоемкость разработки комплекса программ;
- длительность разработки комплекса программ;
- число специалистов, участвующих в создании программного продукта.

Некоторые из перечисленных факторов коррелированы между собой, причем, определяющими, обычно являются размер комплекса программ и объем базы данных. Остальные характеристики можно рассматривать как производные, вторичные, однако они могут представлять самостоятельный интерес при анализе и прогнозировании экономических характеристик производства программного продукта.

Анализ предварительных экономических характеристик проекта должен начинаться с прогнозирования возможного размера комплекса программ, с учетом того, что его *достоверность может быть недостаточной, и обусловлена следующими факторами*:

- проблема проектирования может быть недостаточно хорошо понята разработчиками и/или заказчиками из-за того, что некоторые производственные факторы были упущены или искажены из-за предвзятого к ним отношения;
- недостаток либо полное отсутствие прототипов не позволили создать базу данных для экономических оценок и их прогнозирования;
- проектирующее предприятие не располагает стандартами и методиками, с помощью которых можно выполнять производство и экономический прогноз размера проекта (либо в случае наличия стандартов их никто не придерживается);
- менеджеры проектов полагают, что необходимо фиксировать требования к сложности и размеру программного продукта в начале проекта, заказчики же зачастую считают, что не стоит тратить время на разработку спецификации требований, оценки размеров и экономических характеристик продукта;

- могут потребоваться дополнительные компоненты в комплексе программ (для интерфейсов системы, аппаратного обеспечения и внешней среды), что отражается на размерах программного продукта;

- имеет место недостаточно четкое представление об ограничениях ресурсов на уровне системы и возможностях совершенствования создаваемого программного продукта.

Для конкретизации основной области дальнейшего анализа экономики производства целесообразно выделить и классифицировать **обобщенные характеристики комплексов программ** в соответствии со стандартом **ISO 12182** – (см. Приложение). В стандарте выделены три группы видов характеристик: внутренние виды; виды среды и виды данных. Для каждого вида представлен перечень классов, из которых рекомендуется выбирать подходящие характеристики для отражения особенностей конкретной системы или описания достаточно широкой **сферы анализа и применения комплекса программ**. Из общего числа видов, **далее целесообразно учитывать следующие**:

- базовые функции комплексов программ – системы управления объектами или процессами в реальном времени;

- организационные, административные, проектировочные и/или обучающие системы;

- прикладная область системы – характеристики оборудования и аппаратуры управления процессами и объектами; информационные, административные, и обучающие системы;

- режимы эксплуатации – обработка данных в пакетном режиме или в реальном времени;

- масштаб, размер комплекса программ – средний или большой;

- представление данных – предметное, формализованные описания объектов или процессов;

- критичность программного продукта – средняя или высокая, должна быть предотвращена возможность больших экономических потерь, повреждения дорогой собственности или угроза человеческой жизни;

- класс пользователей – технические процессы, внешние системы, средства и объекты, обучающиеся или квалифицированные специалисты;

- стабильность программного продукта – маловероятное или дискретное внесение изменений в процессе регламентированного сопровождения;
- характер использования программного продукта – заказной для конкретного применения в определенной системе или для массового распространения на рынке среди предприятий и пользователей;
- требуемые динамические характеристики: время отклика – быстрое (секунды или минуты); производительность – большая или средняя;
- требования безопасности и надежности – произвольные, высокие или критические;
- требования к использованию вычислительных ресурсов – не критические или высокие, почти полное использование ресурсов по основному функциональному назначению.

Имеющаяся *статистика экономических характеристик производства программных продуктов различных классов* позволила выявить основные факторы, от которых они зависят (см. рис. 4.1). Изучены тенденции их изменения от важнейших параметров. Для учета классов комплексов программ с позиции экономики их производства проведено ранжирование экспериментальных данных и выделены *три достаточно различающихся базовых класса* при одном и том же размере, характеризующиеся [3, 16, 38]:

- максимальной трудоемкостью производства – *встроенные* программные продукты сложных систем реального времени (СРВ);
- средней трудоемкостью разработки – *полунезависимые* программные средства административных, обучающих, информационных и поисковых систем (АС);
- минимальной трудоемкостью создания – *распространенные* комплексы программ, практически независимые от реального времени, аппаратуры систем и внешней среды, научные и исследовательские пакеты прикладных программ (ППП).

Все остальные классы комплексов программ могут быть упорядочены между выделенными классами, и для них изменения трудоемкости обычно оцениваются относительно максимальной, для программных продуктов реального времени. Экспериментальные оценки трудоемкости имеют значительную дисперсию, которая обусловлена рядом трудно учитываемых факторов. Малые комплексы программ при производстве небольшими коллективами специалистов

характеризуются высокими коэффициентами вариации значений трудоемкости, вследствие *особой роли индивидуальной квалификации* участвующих специалистов. Трудоемкость разработки сложных программных продуктов размером свыше 100 тыс. строк отражается более стабильными значениями, что обусловлено *усреднением творческих возможностей* специалистов и условий их труда в больших коллективах, а также возрастанием роли руководящего и вспомогательного персонала. В этой области размеров статистические данные лучше аппроксимируются эмпирическими зависимостями с учетом основных факторов.

Встроенный класс программных продуктов реального времени (СРВ) отличается от других жесткими ограничениями на характеристики и ресурсы ЭВМ, реализующих программы, и правила использования интерфейсов. Такие продукты обычно практически полностью используют ресурсы вычислительных машин по памяти и производительности, снабжаются подробной документацией, эксплуатируются и поэтапно модернизируются как промышленная продукция многие годы и даже десятилетия. Эти комплексы программ непосредственно определяют степень автоматизации производства в промышленности, качество управления военными и техническими объектами и системами. Такие программные продукты должны функционировать при тесной взаимосвязи аппаратуры, программ, и вычислительных процессов, а также *тщательно оцениваться и контролироваться экономикой производства*. Примерами являются системы резервирования авиабилетов, управления динамическими объектами и воздушным движением. Как правило, производственные затраты на изменение частей и функций комплекса программ настолько велики, что иногда гораздо выгоднее оставлять характеристики комплекса неизменными. Следствием этого является стабильность программ этого класса. Комплексы программ требуют больших затрат труда при динамическом тестировании для установления их соответствия спецификациям требований (более высокие затраты на верификацию и аттестацию) и на обеспечение контроля правильности изменений (более высокие затраты на испытания и управление конфигурацией). Все эти факторы приводят к повышению затрат времени и труда, а также к увеличению влияния на нее масштаба больших проектов.

Встроенный программный комплекс, как правило, планируется при недостаточной исходной информации об условиях и требованиях

для его производства. После завершения предварительного проектирования, рациональной стратегией создания встроенного программного продукта является привлечение большой группы специалистов для параллельного выполнения детального проектирования, кодирования и автономного тестирования полного набора компонентов. Для встроенных продуктов указанная стратегия приводит к большим затратам труда, чем в проектах других классов, реализуемых в такие же общие сроки.

Полунезависимый класс производства программных продуктов (АС) для административных, обучающих и информационных систем, по экономическим характеристикам занимает промежуточное положение, между распространенным и встроенным классом. Эти комплексы программ не полностью используют ресурсы вычислительных систем, могут тиражироваться и передаваться пользователям как законченные **программные продукты**. Их эксплуатация обычно превышает длительность разработки, однако в ходе эксплуатации они непрерывно развиваются. В производстве, модификации и эксплуатации таких продуктов участвуют большие коллективы специалистов, труд которых должен организовываться, планироваться и анализироваться на уровне отдельных команд. Такой может быть система обработки сообщений с весьма строгим соблюдением интерфейсов, например, налагаемых терминальными, организационными или административными требованиями системы и пользователей.

Распространенный класс (ППП) характеризуется тем, что комплексы программ для решения научных или инженерных задач разрабатываются относительно небольшими группами специалистов в хорошо знакомых условиях своего предприятия. Длительность разработки таких программ обычно больше, чем длительность использования, и они практически недоступны для применения никому, кроме авторов. Процесс и технологию создания таких программ трудно регламентировать какими-либо стандартами и нормативами, так как, по существу, программы имеют экспериментальный и научно-исследовательский характер, не предполагают тиражирования и промышленного внедрения. Так же, как любые другие виды научной работы, разработку таких программ **вряд ли целесообразно оценивать экономически** по размеру производства, по производительности труда или средним срокам создания комплексов программ. Большинство специалистов, связанных с проектом обычно обладают большим

опытом работы с аналогичными по функциям комплексами программ в условиях своего предприятия и хорошо понимают, как и каким образом разрабатываемая система будет способствовать достижению целей.

При *анализе экономики производства* сложных комплексов программ первого и второго классов, при формировании их архитектуры, выборе компонентов и их связей **целесообразно применять и учитывать ряд современных концептуальных требований:**

- архитектура программных продуктов должна соответствовать текущим и перспективным целям и стратегическим, функциональным задачам, создаваемой системы;
- в структуре и компонентах следует предусматривать обеспечение максимально возможной сохранности инвестиций в аппаратные и программные продукты, а также в базы данных для длительного развития, сопровождения и модернизации системы и программного продукта;
- для обеспечения перспективы развития систем следует предусматривать возможность интеграции гетерогенных вычислительных компонентов и мобильность комплексов программ на различные аппаратные и операционные платформы (например, на основе концепции и стандартов Открытых систем);
- архитектура комплекса программ должна быть достаточно гибкой и допускать относительно простое, без коренных, структурных изменений, развитие и наращивание функций и ресурсов системы в соответствии с расширением сфер и задач ее применения;
- необходимо обеспечивать эффективное использование ресурсов системы и минимизировать интегральные затраты на обработку данных в типовых режимах ее функционирования с учетом текущих эксплуатационных затрат и капитальных вложений в создание программного продукта;
- следует обеспечить комфортный, максимально упрощенный доступ конечных пользователей к управлению и результатам функционирования системы на основе современных графических средств и наглядных пользовательских интерфейсов.

Описанные особенности классов комплексов программ определили концентрацию в монографии на *наиболее сложной* для анализа *экономике производства* программных продуктах *первого* класса, а также на некоторых комплексах *второго* класса. Программы третьего

класса практически не рассматриваются вследствие их непромышленной разработки, невозможности оценивания и прогнозирования экономики производства и несоответствия требованиям программных продуктов, отчуждаемых от разработчиков.

Единицы измерения сложности и размера программ для экономического анализа их производства

Выбор и применение единиц измерения сложности и размера компонентов и комплексов программ одна из задач при анализе и прогнозировании характеристик *экономики производства* и затрат на их создание. Этому способствует неоднозначность учитываемых этапов разработки комплексов программ и категорий специалистов, трудозатраты которых учитываются и заметно влияют на экономические характеристики. При одних и тех же процессах измерения продуктов производства и затрат на их создание методики определения количественных значений основных экономических характеристик пока не стандартизированы, что вносит дополнительную дисперсию в их значения, приводимые в различных проектах и исследованиях. В результате могут делаться различные и даже принципиально неверные выводы, например, об очень высокой эффективности некоторых частных технологических методов или инструментальных систем автоматизации этапов производства. Для уменьшения этих неопределенностей и сокращения возможных методических ошибок, в экономике производства необходимо определить основные понятия и единицы измерения: сложности, размера или масштаба комплексов программ, трудозатрат на их разработку, производительности труда специалистов и некоторых других экономических характеристик.

Весьма расплывчатое понятие *сложность комплексов программ* значительно конкретизируется и становится измеряемым, когда устанавливается связь этого понятия *с конкретными ресурсами, необходимыми для решения соответствующей задачи*. Если хотя бы один из ресурсов, необходимых для решения, велик или оказывается на пределе, доступном для использования, то такую задачу вряд ли посчитают простой. Сложность задачи может быть велика по одному используемому ограниченному ресурсу и невелика из-за малого использования остальных ресурсов. Например, программа, простая по количеству модулей и длине, может оказаться

сложной по объему вычислений или по числу и структуре обрабатываемых данных, и наоборот.

При разработке комплексов программ основными экономическими лимитирующими ресурсами обычно являются: допустимые трудозатраты специалистов; ограничения на сроки разработки; параметры ЭВМ, технология и инструментарий проектирования и производства. Для этапа эксплуатации комплексов программ основными определяющими ресурсами может становиться память и производительность реализующей ЭВМ, а в качестве ограничений могут выступать время счета типовых задач, доступный объем памяти базы данных. Показатели сложности программного продукта можно рассматривать с позиций статической и динамической, и делить их на *две группы*:

- *сложность проектирования и производства* программного продукта: статическая сложность, когда реализуются его назначение, функции и характеристики, а также вносятся и устраняются основные дефекты и ошибки;

- *сложность функционирования и применения* программного продукта для получения результатов: динамическая сложность, когда реализуется его функциональное назначение и качество, а также могут проявляться дефекты и ошибки.

Размер или масштаб комплексов программ в настоящее время приводится в проектах в различных единицах, что может изменять их численные значения для одних и тех же комплексов в несколько раз [4, 12, 41]. В качестве таких единиц чаще всего используются численные значения: строк текста программы на языке программирования, предложений на ассемблере, число символов в тексте программы, байт или команд в объектном коде после трансляции. Каждая из этих единиц измерения имеет некоторые преимущества при определенных целях исследования и применения. Однако при сравнительном экономическом анализе различных проектов применение в каждом из них отличающихся единиц для характеристики объекта производства приводит к *дополнительному разбросу численных значений размеров и к несопоставимости измеренных экономических затрат*. Влияние единиц измерения размера программ на экономику производства можно значительно уменьшить, если учитывать их принципиальные особенности и разделить на *две группы единиц измерения*:

- группу, характеризующую размер исходных текстов программ, которые *разрабатываются и анализируются человеком*, отражающую сложность и трудоемкость создания комплекса программ и его компонентов;

- группу, отражающую размер *программ и данных, размещаемых в реализующей (объектной) ЭВМ*, и характеризующую объем памяти и производительность ЭВМ, необходимые для функционирования и исполнения программного продукта при его применении.

Эти две группы единиц отражают размер программ, с разных позиций, и должны использоваться в зависимости от целей анализа. Хотя между ними есть корреляция, но в общем случае размеры программ, измеренные различными группами единиц, зачастую оказываются практически несопоставимыми по экономическим характеристикам из-за наличия между ними промежуточного звена – преобразователя (транслятора) с различными, не полностью определенными характеристиками. Это обусловлено особенностями трансляторов, которые преобразуют исходные тексты программ, разработанные человеком – программистом, в разделы памяти реализующей ЭВМ, заполненные командами, константами или выделенные под данные, а также особенностями языков программирования, структуры и содержания машинных команд.

Сложность создания и экономические характеристики комплексов программ целесообразно анализировать на базе трех наиболее специфических компонентов:

- *сложность совокупности программных модулей и компонентов* характеризуется конструктивной структурой компонентов программы и может быть оценена с позиции сложности внутренней структуры и преобразования переменных в каждом модуле, а также интегрально по некоторым внешним статистическим характеристикам размеров модулей и компонентов;

- *сложность структуры программного комплекса* или крупных функциональных компонентов и связей между модулями по задачам управления и по обмену информацией, определяется глубиной взаимодействия модулей и регулярностью структуры межмодульных связей;

- *сложность структуры и совокупности данных* определяется количеством и структурой переменных в базе данных, регуляр-

ностью их размещения в массивах, а также сложностью доступа к этим данным.

Динамическая группа показателей сложности характеризует программные продукты на этапе эксплуатации законченных функционирующих систем. Эти показатели объединяются понятиями **вычислительная сложность применения комплексов программ и сложность подготовки и анализа результатов**. Вычислительная сложность непосредственно связана с ресурсами вычислительной системы, необходимыми для нормального функционирования программного продукта и получения совокупности законченных результатов, может быть представлена следующими **тремя компонентами**:

- **временная сложность** отражает необходимую длительность исполнения программного продукта или время обработки на ЭВМ определенной, типовой совокупности исходных данных для получения требуемых результатов;

- **информационную сложность** можно представить как объем базы данных, обрабатываемых комплексом программ, или как емкость памяти, используемой для накопления и хранения информации при исполнении и применении программного продукта;

- **программная сложность** характеризуется размером программы или объемом памяти ЭВМ, необходимой для размещения и функционирования программного продукта.

В системах реального времени временная сложность непосредственно отражается **временем отклика** (ожидания выходных результатов) на некоторые исходные данные. Пропускная способность комплекса программ при его исполнении на конкретной ЭВМ характеризует динамику обработки исходных данных, и сложность исполнения программного продукта в реальном масштабе времени.

Масштаб – размер комплексов программ по числу строк текста и объему занятой памяти ЭВМ

Трудоемкость проектирования и производства программных продуктов непосредственно зависит от **размера – масштаба комплексов программ, выраженного числом**: операторов на ассемблере, строк на языке программирования высокого уровня или функцио-

нальных точек. Реальное изменение размера создаваемых в настоящее время сложных программных продуктов от 10^4 до 10^8 строк (LOC) определяет диапазон трудоемкости разработки таких программ от человеко-года до десятков тысяч человеко-лет. **Размер исходных текстов** программ, прежде всего, отражается на трудоемкости и длительности их разработки, что позволяет оценивать относительные характеристики производительности труда специалистов – разработчиков. По большому числу проектов подтверждена **высокая корреляция**, между размером текста комплекса программ и трудоемкостью (с коэффициентом) его производства [3, 18, 36]. Однако исходные тексты могут содержать конструкции, которые не исполняются при рабочем функционировании программ: комментарии разных уровней, указания, заголовки и дополнительные описания, требующие определенного труда для подготовки. Для сближения понятий и значений размера программ по обеим (разрабатываемым и исполняемым) группам характеристик целесообразно эти неисполняемые части программы предварительно не учитывать.

Основной труд специалистов вкладывается в разработку текста программ, поэтому желательно, чтобы выбранная единица измерения размера была бы в наибольшей степени **адекватна (пропорциональна) трудоемкости разработки**. При этом должна учитываться не только трудоемкость непосредственной подготовки текста и разработки компонентов программы, но отражаться трудоемкость их интеграции и комплексного создания всего сложного продукта. Кроме того, единица измерения размера должна быть наглядной и просто измеряемой. Расчет показателей производительности труда с учетом полного размера комплекса, в том числе ранее отработанных и испытанных программ, в этом случае неправомерен, так как не учитывается труд, затраченный ранее на готовые компоненты, используемые при сборке. Возможный выход для уменьшения неадекватности численных оценок производительности труда состоит в **учете размера и затрат на комплексирование только вновь разработанных программ**. В этом случае затраты на комплексирование уже ранее отлаженных программ учитываются только в полной трудоемкости производства программного продукта.

При анализе трудоемкости разработки программ, кроме затрат на создание текста программных компонентов на языке программирования, необходимо учитывать затраты: на разработку тестов для их

тестирования, на комментарии к программам, на интерфейсы различных видов, на документацию. В результате **сложность создания программ (непосредственный труд)** значительно превышает, затраты труда с учетом только размера текста. **Масштаб – размер** текста программ может использоваться как **исходный ориентир** для оценки сложности производства компонентов программного продукта. Пересчет величины размера компонента к величине сложности его производства, позволяет более корректно сопоставлять экономические характеристики производства в различных проектах, коллективах и у отдельных специалистов. В пределах одного целостного сложного комплекса программ сравнение специалистов по величине масштаба – размера разработанных компонентов может быть удобным и достаточным для первичной оценки производительности труда отдельных специалистов.

Память для программ, размещаемых в ЭВМ для исполнения – программная сложность, также может служить **ориентиром** при оценке размера комплекса программ. Он влияет на характеристики и стоимость используемых машин, которая зависит от объема памяти и производительности ЭВМ, что особенно важно учитывать в системах реального времени (например, бортовых). Число команд, занятых программой в реализующей ЭВМ, соответствует числу исполняемых операторов в исходном тексте программы. Современные затраты на реализацию программных продуктов в различных ЭВМ изменяются приблизительно в таком же широком диапазоне, как и трудоемкость их разработки (на четыре порядка). Исследование и применение различных единиц измерения, используемых при оценке размеров комплексов программ, обычно зависит от конкретного проекта и потребностей предприятия. За рубежом чаще всего размер комплекса программ определяется в терминах строк текста программного кода на языке программирования (Lines of code – LOC) или функциональных точек [3, 12, 15] (см. рис. 4.1).

Единицы измерения размера программ, используемые специалистами, при создании и анализе текстов программ, а также для экономического анализа процессов их производства должны учитывать следующие особенности программ:

- **новый код текстов программ**, разработанный полностью для нового проекта, который не включает фрагменты и процедуры ранее написанного и испытанного кода на языке программирования;

- **модифицированный код программных компонентов**, разработанный для предыдущих проектов, который пригоден для использования в новом проекте, после адаптации и внесения умеренного объема изменений;

- **повторно используемый код компонента**, разработанный для предыдущих проектов, который полностью пригоден для нового программного продукта без внесения изменений.

Код, повторно используемый в полном объеме, должен иметь полную идентичную документацию, идентичные тестовые процедуры и текст на языке программирования, а также копию с целью поддержки библиотеки менеджмента производства. Если изменяется лишь одна строка исполняемого кода, то подвергается изменению тестовый код и документация. Первый этап при **экономической оценке размера программных комплексов**, которые могут повторно использовать код, заключается в отделении нового кода от модифицируемых компонентов. Это необходимо по той причине, что модифицируемые компоненты практически никогда не могут быть использованы непосредственно, для выполнения их интеграции в комплекс и требуются некоторые изменения, в результате чего возрастает размер компонентов и трудозатрат, требуемых для реализации подобных изменений. Если компонент в целом не был изменен, он может **повторно использоваться полностью**. Если компонент был изменен (пусть в одном комментарии или выполняемом операторе), он является **модифицируемым**. Целесообразно отделять модификации, предназначенные для корректировки дефектов, от направленных на изменение и расширение функций.

После завершения оценки размера разработанного программного комплекса, повторно используемый компонент целесообразно **преобразовать в эквивалентный новый код**. В ходе процесса преобразования ставится цель минимизировать трудозатраты при внедрении повторно используемых или модифицируемых компонентов по сравнению с применением полностью новых программ. Обычно определяется множитель преобразования с целью экономического отражения количества трудозатрат, сэкономленных при повторном использовании модулей или компонентов. При внедрении повторно используемых компонентов может экономиться до 70% трудозатрат по сравнению с внедрением новых компонентов. При использовании

модифицируемых программ экономия может составлять около 40% (см. главу 6).

Оценку размера комплекса и компонентов программ и производственных трудозатрат приходится выполнять неоднократно во время жизненного цикла, причем после каждого оценивания повышается уровень доверия к полученным результатам. **Точность экономического прогнозирования** значительно повышается поэтапно: после формулирования начальных требований заказчика, проведения анализа требований, после завершения разработки детального проекта. Квалифицированный менеджер проекта обязан оценивать (повторно) размеры комплекса программ, используя результаты оценивания в качестве выходных **экономических критериев для каждого этапа**. К ним **относятся числа**:

- строк текста программы на используемом языке программирования высокого уровня;
- строки кода в терминах Lines of code – LOC;
- функциональных точек.

Число строк программы на языках высокого уровня (ЯВУ) в значительной степени зависит от конкретных особенностей используемого языка. Языки программирования высокого уровня ориентированы на определенные классы задач. Благодаря этому проблемно-ориентированные языки позволяют получать компактные тексты соответствующих классов программ. Однако применение их для других классов программных продуктов приводит к расширению текста и не всегда рационально. ЯВУ значительно различаются степенью обобщенного описания содержания алгоритмов, вследствие их проблемной ориентации, применения разных структур операторов, операндов и процедур, что может приводить к различию (в несколько раз) числа строк текста, по сути, одних и тех же программ [3].

Число строк текста на ЯВУ характеризует сложность программы для ее разработчика, однако оно не отражает функциональную (потенциальную) сложность, которая значительно выше за счет применения процедур и комментариев. Эти факторы приводят к тому, что размеры одинаковых программ на различных ЯВУ, трудно сводить к унифицированным единицам. Для такой унификации желательно выделять **базовый, широко применяемый язык программирования** (например, Си) и определять коэффициенты пересчета числа

строк при разработке комплексов программ на других ЯВУ с учетом классов программ.

Для обобщения *экономических характеристик производства программных продуктов* необходимо обеспечивать [16]:

- расчет числа строк на ЯВУ и определение среднестатистических коэффициентов для их приведения (без учета расширения вследствие трансляции) к числу операторов той же ЭВМ, для которой разработаны программы на ЯВУ (коэффициента сжатия текстов на ЯВУ относительно текста, например, на Си);
- определение коэффициентов расширения трансляторов с ЯВУ при получении программ в объектном коде реализующей ЭВМ, как отношения числа команд в объектном коде реализующей ЭВМ, эквивалентного числу операторов на Си, к числу строк на ЯВУ в тексте программ (полное расширение транслированного текста в объектном коде реализующей ЭВМ относительно текста на ЯВУ).

Разнообразие ЯВУ и архитектуры реализующих ЭВМ не позволяет заранее определить перечисленные коэффициенты для всех условий анализа экономических характеристик. Поэтому целесообразны варианты таких оценок, которые могут служить *ориентирами при анализе конкретных проектов*, а также методик, для определения коэффициентов пересчета единиц измерения размеров программ.

Преимущества использования LOC в качестве единиц измерения размера программ [4, 12, 36]:

- позволяют выполнять сопоставление методов измерения размеров программ и производительности в различных группах разработчиков одного проекта комплекса программ;
- непосредственно связаны с размером конечного программного продукта;
- единицы LOC могут применяться на промежуточных этапах до завершения проекта;
- оценка размеров компонентов производится с точки зрения разработчика – физического размера созданного программного продукта, количества написанных и отлаженных строк кода всех компонентов;
- совершенствование программ и размер их расширения может быть сопоставлен с реальным размером компонентов в ходе доработок.

Недостатки, связанные с применением LOC при оценке размера программ:

- единицы измерения LOC затруднительны в применении при прогнозировании размера комплекса программ на ранних стадиях его жизненного цикла;
- исходные инструкции могут различаться в зависимости от типов языков программирования, методов проектирования, стиля и способностей программистов;
- разработка программного продукта может быть связана с большими затратами, которые прямо не зависят от размеров программного кода, на спецификации требований и пользовательские документы, которые обычно не входят в прямые затраты на кодирование программ;
- при подсчете числа LOC следует различать автоматически и вручную созданный код – эта задача является более сложной, чем простой подсчет, который может быть выполнен на основе листинга, сгенерированного компилятором, либо с помощью утилиты, выполняющей подсчет строк программного кода;
- генераторы кода, не имеющие эффективную оптимизацию, создают чрезмерный размер программного кода, в результате чего искажаются показатели LOC.

Масштаб - размер комплексов программ по числу функциональных точек

Использование функциональных точек в качестве единиц измерения основывается на том, что размер программного компонента (или модуля) можно оценивать в терминах **количества и сложности функций**, реализованных в данном программном коде, а не только посредством количества строк кода [12, 31, 36]. При использовании этого метода измеряются и применяются категории пользовательских функций. При этом способ их определения является более методичным и адекватным трудоемкости, чем в случае подсчета строк LOC. Выше представленный подход учитывал лишь размер компонентов и комплекса программ – этот метод учитывает размер и их функциональные свойства (см. рис. 4.1).

Метод функциональных точек применяется для оценки сложности функциональности программ в количественных величинах. Це-

лью является оценка единого числа, которое бы полностью характеризовало сложность программного компонента (модуля) и адекватно соотносилось с показателями затрат труда, соответствующими конкретным программистам. Фактически, функциональные точки характеризуют сложность программного компонента и могут использоваться для прогнозирования времени, необходимого на его разработку, а также нужного количества специалистов для производства программного продукта. Метод функциональных точек разработан эмпирически, используется *в экономике производства программных комплексов* и позволяет решать *следующие задачи*:

- оценивать категории сложности компонентов и пользовательских функций;
- разрешать проблему, связанную с не адекватностью применения единиц измерения LOC на ранних стадиях жизненного цикла комплекса программ;
- определять и учитывать количество и сложность входных и выходных данных, запросов к базе данных, файлов либо структур данных, а также внешних интерфейсов, связанных с программными компонентами.

Метод функциональных точек для измерения размера программных комплексов отражен в международных стандартах (см. рис. 4.1) **ISO 20926**, **ISO 20926** и **ISO 14143:1-5**. В последнем стандарте вначале (первая часть) представлена концепция методов измерения функциональных точек программных компонентов и перечень обязательных требований к стандартизированному методу. Назначение этой части облегчить возможность сравнения функционального размера разных комплексов программ. В части 2 изложена схема оценивания выбранного метода измерения функционального размера на соответствие концепции и обязательным требованиям стандарта, для обеспечения объективности, беспристрастность и повторяемость измерений размера компонентов и комплекса программ. Схема включает требования к составу и квалификации экспертных групп, к входным документам для оценивания, к процедурам выполнения оценивания и к оформлению результатов.

При выборе метода оценивания размера, должны учитываться класс и функциональная область применения программного продукта. Для этого конкретный метод функциональных точек должен быть верифицирован путем сопоставления результатов оценивания этим

методом и другим, ранее применявшимся, методом. В четвертой части приводятся эталонная модель и способы оценки эффективности методов измерения функционального размера, для различных классов программных продуктов в различных областях применения.

Пользователи метода должны иметь возможность проверить его соответствие функциональной области; оценщики методов – располагать эталонными объектами, по отношению к которым метод может быть применен и сопоставлен. Менеджеры проектов могут использовать эталонные данные для контрольного тестирования качества программного продукта и производительности при разработке проекта. Стандарт содержит примеры эталонных требований для двух крупных классов комплексов программ – административных информационных систем и систем реального времени. Цель пятой части стандарта – определение функциональных доменов (компонентов текста программ) для упрощения их использования при измерении функционального размера и для классификации требований потребителя при применении этого метода измерения. Оценщики и эксперты по методам измерения функционального размера должны быть обеспечены руководствами для определения функциональных областей, в которых можно эффективно использовать эти методы.

В этом методе рекомендуются **пять базовых элементов** для определения сложности каждого программного компонента, и величина функциональности конкретного компонента является взвешенной суммой этих пяти элементов. Веса элементов могут быть получены эмпирически и обоснованы наблюдениями за их применением в конкретных проектах. При измерении учитываются только видимые, или необходимые пользователю функциональности компонентов [7, 36]:

- **внешние входы**, которые по-разному влияют на выполняемую функцию, считаются отдельными, например, если функция заключается в сложении двух чисел, то она имеет один вход, а не два;
- **внешние выходы** считаются только для существенно различных алгоритмов и нетривиальной функциональности, если выходные данные посылаются на существенно разные устройства (например, принтер и монитор), то они считаются отдельными выходами;
- **внешние запросы** – каждый независимый внешний запрос считается за один;
- **внутренние логические файлы** – каждая уникальная логиче-

ская группа пользовательских данных, которая создается или поддерживается функцией, считается за одну, комбинации таких групп не считаются, каждая группа данных должна учитываться один раз;

- **внешние логические файлы** – каждая уникальная логическая группа пользовательских данных, размещенная во внешних по отношению к приложению файлах, считается за одну.

Количество вводов и выводов определяет число элементов, направляемых пользователем в систему и возвращаемых системой пользователю, соответственно. При подсчете количества выводов группа элементов, например экран или файл, принимается за один элемент, т.е. входящие в группу элементы отдельно не считаются. Число запросов – это количество выполненных пользователем интерактивных запросов, требующих определенной реакции программного компонента. Файлы представляют собой любую группу взаимосвязанных элементов информации, поддерживаемую компонентом от имени пользователя, либо предназначенную для организации компонента. Наконец, оценка интерфейсов – это число внешних интерфейсов с другими компонентами (например, количество файлов для считывания или записи или количество файлов, переданных в другие компоненты или полученных из них). Таким образом, если компонент считывает файл, созданный другим компонентом, тогда данный файл считается два раза: один раз – как входная информация, а второй – как интерфейс. Если файл поддерживается двумя разными компонентами, то для каждого он рассматривается как интерфейс. Конечной целью функциональной оценки является измерение сложности программного компонента только на основании функций, которые данный компонент должен предоставлять пользователям.

Процесс применения метода функциональных точек включает:

- ввод перечня категорий элементов: вывод и ввод, процессы и функции, структуры данных и интерфейсы;
- определение сложности каждой категории и функции – простая, средняя, сложная;
- представление требования для оценки сложности каждой из категорий элементов;
- подсчитывание величины функции в каждой категории элементов;

- каждая функция умножается на соответствующий ей параметр сложности, а затем производится суммирование с целью получения общего количества функциональных точек в компоненте;
- преобразование величины функциональных точек в единицы измерения ЛОС и их суммирование для оценки компонента.

Множитель преобразования основывается на статистических показателях для определенных классов комплексов программ и языка программирования, представляя среднее количество строк кода, применяемых в среднем для реализации простой функции. Потребность в этом параметре обосновывается тем, что ряд автоматизированных инструментов, предназначенных для оценки трудозатрат, а также для составления графика работ, нуждаются в использовании ЛОС программных компонентов в качестве входных данных.

Метод функциональных точек обеспечивает предварительную оценку размера потенциальных компонентов программ либо программных комплексов. При этом осуществляется анализ сложности будущих функциональных свойств с пользовательской точки зрения. Языки программирования различны по своим характеристикам, однако существует некое среднее количество выполняемых операторов, требуемых для реализации одной функциональной точки. **Преобразование функциональных точек в строки ЛОС** может требоваться с целью:

- измерения и сравнения производительности при создании программных компонентов, либо комплексов, которые были разработаны на различных языках программирования;
- использования стандартизированных единиц измерения для осуществления ввода данных в инструментальные средства оценки размера крупного комплекса программ;
- преобразования размера комплекса программ либо компонента, написанных на любом языке программирования, в эквивалентный размер в случае взаимодействия с компонентами, написанными на другом языке программирования.

Некоторые **преимущества, проявляющиеся при использовании функциональных точек** в качестве единиц измерения сложности комплексов программ, составляют [36]:

- метод не зависит от используемого языка программирования и технологии, а также исключает некоторые корректировки, выполняемые на заключительном этапе экономической оценки проекта;

- для пользователей эти единицы измерения могут быть более привычными и выгодными, при этом облегчается понимание руководителями степени влияния на экономические характеристики изменений функциональных требований в процессе разработки компонентов;
- может измеряться и сравниваться уровень производительности при создании комплексов программ, написанных на различных языках программирования или различными коллективами специалистов;
- функциональные точки могут учитываться на ранних этапах производства – результаты подсчета функциональных точек могут сравниваться в конце этапов формирования требований, проведения анализа концепции, производства программного продукта, а также его внедрения, если количество функциональных точек увеличивается при каждом подсчете, это означает, что проект лучше определен, либо размер проекта реально увеличился;
- функциональные точки могут использоваться в системах, в которых реализован графический пользовательский интерфейс, в клиент-серверных системах, а также при объектно-ориентированной разработке.

Так же, как и в случае с моделями определения размеров и оценивания по величине ЛОС, может выполняться *адаптация и калибровка функциональных моделей* для повышения их точности и адекватности в конкретных проектах. Могут учитываться способы применения приоритетов и относительных весов, а также факторы внешней среды, которые в целом являются изменяемыми.

Недостатки анализа размера компонентов и особенно крупных комплексов программ методом функциональных точек состоят в следующем:

- используются субъективные оценки параметров размера и сложности функциональных точек, вследствие чего возможны конфликты между заказчиком и разработчиками при оценке стоимости и производительности при реализации проекта;
- получаемые результаты размеров и сложности зависят от технологии, используемой для их оценки и реализации;
- многие модели трудозатрат зависят от показателя ЛОС, поэтому приходится преобразовывать количество функциональных точек в значения ЛОС;

- требуются дополнительные исследования и сравнения оценок на базе LOC, отличных от оценок, основанных на функциональных точках;
- метод эффективно применим после создания спецификаций компонентов, а также архитектуры компонентов и комплекса программ.

Имея систему измерений функциональности компонентов, можно рассчитывать и сравнивать месячную **индивидуальную производительность** определенного специалиста или целой рабочей группы путем деления функциональности созданного программного компонента на количество месяцев, потраченное на его разработку. Это позволяет экономически более адекватно взаимодействовать производителям с заказчиком при расчетах стоимости компонентов. Можно также измерять коэффициент ошибок на функциональную точку путем деления числа выявленных ошибок на функциональность соответствующих компонентов для **оценки качества труда определенных специалистов** полностью создающих компонент. Эти относящиеся к функциональным точкам числа можно фиксировать и использовать при будущем планировании или при сравнении показателей производства компонентов одного проекта с показателями других.

Однако при коллективном производстве сложных программных продуктов индивидуальные характеристики специалистов нивелируются, многие из них не участвуют в непосредственном программировании (аналитики-спецификаторы, тестировщики, документаторы), в результате чего расчет функциональности и ее **применение в экономике производства сложных программных продуктов теряет смысл**. В таких проектах целесообразнее пользоваться для учета размера компонентов и комплексов программ традиционной величиной числа строк LOC. Поэтому далее в монографии все экономические оценки трудоемкости, длительности и производительности базируются на величине LOC.

Глава 5

ХАРАКТЕРИСТИКИ КАЧЕСТВА ПРОГРАММНЫХ ПРОДУКТОВ, ВЛИЯЮЩИЕ НА ЭКОНОМИКУ ИХ ПРОИЗВОДСТВА

Влияние качества программных продуктов на экономические характеристики производства

На экономические характеристики производства программных продуктов значительно влияют *основные требования к их характеристикам качества*. Подробный состав требований к характеристикам сложных программных продуктов, регламентирован в международных стандартах **ISO 9126** и **ISO 25000**. В них рассмотрено около двадцати основных характеристик и ряд их атрибутов качества, однако, ниже при анализе выделены доминирующие и их количество несколько сокращено. Стандартизированные в этих документах характеристики качества целесообразно использовать при экономическом обосновании производства сложных программных продуктов, к которым предъявляются *высокие требования к качеству* функционирования и применения.

Высшие приоритеты в конкретных проектах, естественно, должны присваиваться свойствам и атрибутам качества, необходимым для достижения *стратегических целей*, отражающих назначение и функции применения программного продукта. Для их реализации следует выделять достаточные ресурсы. Все остальные стандартизированные характеристики программного продукта, должны способствовать обеспечению *тактических целей* реализации выбранных конструктивных требований качества при доступных ресурсах. Поэтому им могут быть присвоены более низкие приоритеты и на выполнение соответствующих требований, могут выделяться меньшие ресурсы, что, в частности, может отражаться на не полной реализации некоторых формализованных заказчиком требований качества

вследствие ограниченности ресурсов. Стандартами рекомендуется, чтобы было обеспечено измерение каждой характеристики качества при испытаниях с точностью и определенностью, достаточной для сравнений с требованиями технических заданий и спецификаций, и чтобы измерения были объективны и воспроизводимы.

Системная эффективность целевого применения программных продуктов определяется степенью и качеством удовлетворения потребностей определенных лиц – заказчиков и/или пользователей. В стандартах эта эффективность отражается основной, обобщенной характеристикой качества – **функциональной пригодностью программного продукта**. В связи с тем, что ее абсолютную величину обычно трудно измерить непосредственно и количественно, то по ряду показателей необходима и возможна качественная оценка свойств и достоинств основных функций продукта при применении.

Выбор требований к характеристикам качества программного продукта, естественно, должен начинаться с определения и формирования требований к его функциональной пригодности. Это наиболее ответственная, стратегическая задача начальных этапов проектирования и всего последующего производства. Решение задачи должно быть направлено на обеспечение требуемой высокой функциональной пригодности продукта **путем сбалансированного установления остальных характеристик качества в условиях ограниченных ресурсов**. Излишне высокие требования к отдельным атрибутам качества, требующие для реализации больших экономических, трудовых и вычислительных ресурсов, целесообразно снижать, если они слабо влияют на основные, функциональные характеристики. Ориентирами могут служить диапазоны изменения характеристик, границы количественных или качественных шкал которых сверху и снизу могут быть выбраны **на основе следующих принципов**:

- предельные значения характеристик качества должны быть ограничены сверху допустимыми или рациональными затратами ресурсов на их достижение при производстве и совершенствовании программного продукта;
- наибольшие допустимые затраты ресурсов, например труда и времени для реализации характеристик качества, должны обеспечивать функциональную пригодность применения программного продукта на достаточно высоком уровне;

- допустимые наихудшие значения отдельных характеристик качества, могут соответствовать значениям, при которых заметно начинает снижаться функциональная пригодность при применении программного продукта;
- ограниченные значения отдельных характеристик качества, не должны негативно отражаться на возможных высоких значениях других основных приоритетных характеристик.

Функциональная пригодность – это набор и описания атрибутов, определяющих *назначение, основные, необходимые и достаточные функции программного продукта*, удовлетворяющие потребителей, заданные техническим заданием и спецификациями требований заказчика или потенциального пользователя – рис. 5.1. Данная характеристика определяется тем, *какие* функции и задачи должен решать продукт для удовлетворения потребностей пользователей, в то время как другие, конструктивные характеристики главным образом связаны с тем, *как и при каких условиях*, заданные функции могут выполняться с требуемым качеством. Атрибуты функциональной пригодности можно характеризовать в основном свойствами, категориями и качественным описанием функций проблемно-ориентированной сферы применения, для которых зачастую трудно определить численные меры и шкалы [17, 42].

В процессе производства комплекса программ *атрибуты функциональной пригодности* должны конкретизироваться в спецификациях требований к комплексу программ в целом и к его компонентам *с учетом экономических ограничений и доступных ресурсов*. Атрибутами этой характеристики качества могут быть функциональная полнота решения заданного комплекса задач, степень покрытия функциональных требований спецификациями и их стабильность при совершенствовании версий продукта, количество, достоверность и полнота реализуемых требований заказчиков. На последовательных этапах жизненного цикла, функции промежуточных продуктов (спецификаций компонентов, модулей, текстов программ) должны конкретизироваться и оцениваться в отдельных, частных документах на соответствие исходным требованиям к комплексу программ. Такими атрибутами могут быть: функциональная адекватность программного продукта документам и декларированным требованиям, утвержденным заказчиком; степень покрытия тестами исходных требований к продукту и компонентам; полнота и законченность

реализации этих требований; точность выполнения требований детальных спецификаций на функциональные компоненты.

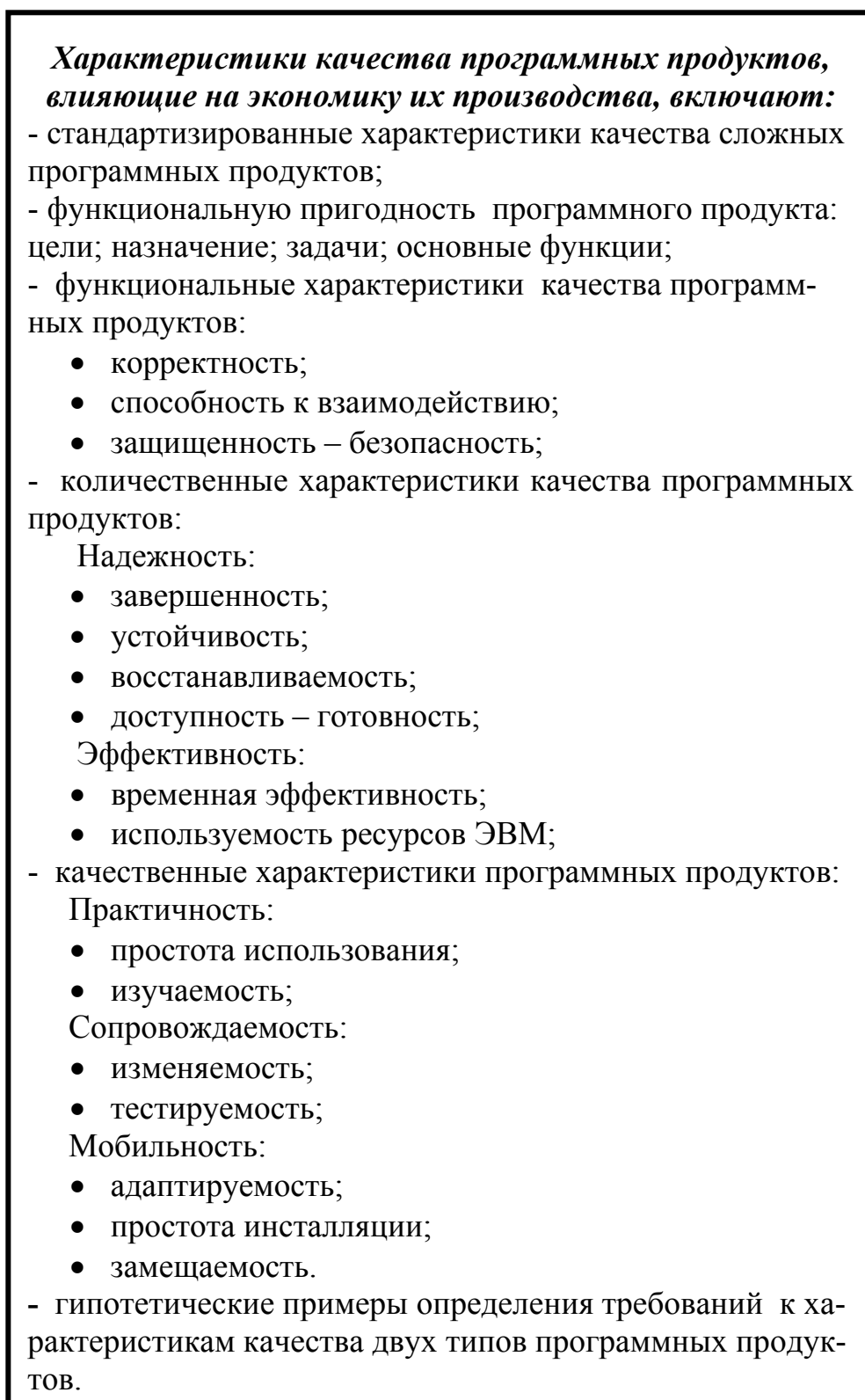


Рис. 5.1

Функциональная пригодность определяется качеством взаимосвязей и трассируемости последовательных формулировок содержания и реализации основных фрагментов стандартизированных требований технического задания на программный продукт, определяющих что:

- описание целей программного продукта корректно переработано в подробное описание его назначения и внешней среды применения;
- назначение продукта полностью и корректно детализировано в требованиях к его функциям и компонентам;
- реализация требований к функциям программного продукта обеспечена достоверным и адекватным составом и содержанием исходной информации от объектов внешней среды и пользователей;
- реализация функций программного продукта способна определять всю требуемую и достаточно корректную информацию для пользователей и объектов внешней среды.

Функции программного продукта реализуются в определенной аппаратной, операционной и пользовательской внешней среде системы, характеристики которых существенно влияют на функциональную пригодность. Для выполнения требуемых функций необходима **адекватная исходная информация от объектов внешней среды**, содержание которой должно полностью обеспечивать реализацию декларированных функций. Полнота формализации номенклатуры, структуры и качества **входной информации** для выполнения требуемых функций, является одной из важных составляющих при определении функциональной пригодности программного продукта в соответствующей внешней среде.

Цель и функции программного продукта реализуются тогда, когда **выходная информация** достигает потребителей – объектов системы или операторов-пользователей, с требуемым содержанием и качеством, достаточным для обеспечения ее эффективного применения. Степень покрытия всей выходной информацией требований: целей, назначения и функций продукта для пользователей, следует рассматривать как основную **меру качества функциональной пригодности**. Прослеживание и оценивание адекватности и полноты состава выходной информации снизу вверх к назначению программного продукта должны завершать выбор базовых характеристик качества

функциональной пригодности, независимо от сферы применения системы.

В ряде систем особое значение для функциональной пригодности программного продукта имеет **организация информационного обеспечения и базы данных**. При этом должны быть определены:

- назначение базы данных и состав информации в ней;
- совместимость программного продукта с другими системами по источникам и потребителям информации базы данных;
- организация сбора, передачи, контроля и корректировки информации базы данных;
- интенсивность и объемы потоков информации для базы данных.

Улучшение каждой, вспомогательной – **конструктивной характеристики качества** программного продукта, требует некоторых экономических ресурсов (трудоемкости, финансов, времени), которые **отражаются** на основной характеристике – на функциональной пригодности. Эти конструктивные характеристики имеют значение для программного продукта постольку, поскольку они обеспечивают требуемое качество реализации основного его назначения и функций. При выборе конкретных конструктивных характеристик качества, следует учитывать затраты ресурсов на их достижение и на результирующее повышение функциональной пригодности, желательно, в **сопоставимых экономических единицах**, в тех же мерах и масштабах. Такое, даже приблизительное, качественное сравнение эффекта и затрат позволяет избегать при производстве многих нерентабельных завышений требований к отдельным конструктивным характеристикам качества, которые не достаточно адекватно отражаются на улучшении функций продукта.

Ориентирами степени влияния ряда основных факторов на качество функциональной пригодности могут служить рейтинги модели СОСОМО II. В этой модели на экономику функциональной пригодности учитывается влияние **новизны проекта F1** (диапазон рейтингов 6,2 – 0) и **согласованности требований F2** (диапазон рейтингов 5,07 – 0) (см. таблицы 8.6 и 8.7), а также **сложность функций комплекса M3** (диапазон рейтингов 0,73 – 1,74) и **размер базы данных M2** (диапазон рейтингов 0,90 – 1,28) (см. таблицы 9.2 и 9.3). Эти факторы при прогнозировании экономических характеристик – трудоемкости и длительности производства программных продуктов могут

существенно влиять на их значения и **в несколько раз изменять** затраты на производство сложных программных продуктов. Значения приведенных рейтингов может быть полезно использовать автономно, независимо от базовых выражений модели СОСОМО II при первичных оценках и формировании функциональных требований к конкретному программному продукту.

Стандартизированные характеристики качества сложных программных продуктов

В зависимости от назначения и размера программного продукта (см. главу 4) почти каждая из его **характеристик качества может стать доминирующей** или даже почти полностью определяющей функциональную пригодность программного продукта (см. ISO 9126). В наибольшей степени функциональная пригодность во многих случаях, например, в системах реального времени, зависит от **корректности и надежности программного продукта**. Значительные трудности при создании ориентиров для выбора мер и шкал требований к характеристикам качества, проявляются при анализе корректности, способности к взаимодействию и к безопасности программного продукта (см. рис. 5.1).

Правильность – корректность это способность программного продукта обеспечивать правильные или приемлемые результаты в соответствии с требованиями заказчика и пользователей. Эталонами для выбора требований к корректности при проектировании могут быть верифицированные и взаимоувязанные требования к функциям комплекса, компонентов и модулей программ, а также правила их структурного построения, организация взаимодействия и интерфейсов. Эти требования при разработке должны быть прослежены сверху вниз до модулей, и использоваться как эталоны при установлении необходимой корректности соответствующих компонентов.

Требования к характеристике **корректность** могут представляться в виде описания двух основных свойств, которым должны соответствовать все программные компоненты и комплекс в целом. Первое требование состоит в выполнении определенной степени (%) прослеживаемости сверху вниз реализации требований технического задания и спецификации на программный продукт при последовательной детализации и верификации описаний программных компо-

нентов вплоть до текстов и объектного кода программ. Второе требование заключается в выборе степени и стратегии покрытия тестами структуры и функций программных компонентов, совокупности маршрутов исполнения модулей и всего комплекса программ для процесса верификации и тестирования, достаточного для функционирования программного продукта с требуемым качеством и точностью результатов, при реальных ограничениях экономических ресурсов на тестирование.

Способность к взаимодействию – состоит в свойстве комплекса программ и его компонентов взаимодействовать с множеством определенных компонентов внутренней и внешней среды. При выборе и установлении при производстве способности программных и информационных компонентов к взаимодействию, ее можно оценивать объемом технологических изменений в комплексе программ, которые необходимо выполнять при дополнении или исключении некоторой функции или компонента, когда отсутствуют изменения операционной, аппаратной или пользовательской среды. С этим показателем связана корректность и унифицированность межмодульных интерфейсов, которые определяются двумя видами связей: по управлению и по информации. При этом важно учитывать возможность повторного использования апробированных компонентов и переноса их на различные платформы. Ряд общих понятий, методов и функций, которые могут рассматриваться как достаточно полная база и набор свойств и интерфейсов компонентов, обеспечивающих высокую способность к взаимодействию, в частности обобщены в *концепции, методах и стандартах Открытых систем* [49].

Защищенность – безопасность тесно связана с особенностями функциональной пригодности каждого программного продукта реального времени [19, 33, 50]. Разработка и формирование требований к свойствам безопасности должны осуществляться на основе потребностей эффективной реализации назначения и функций продукта при различных, реальных угрозах. В процессе проектирования должны быть выявлены потенциальные предумышленные и случайные угрозы функционированию, и установлен необходимый уровень безопасности данного программного продукта.

Эффективная система защиты информации и программных продуктов подразумевает наличие совокупности организационных и технических мероприятий, направленных на предупреждение различных

угроз безопасности, их выявление, локализацию и ликвидацию. Создание такой системы защиты предусматривает **планирование и реализацию целенаправленной политики комплексного обеспечения безопасности**. Потенциальная возможность и реальные проявления информационных диверсий, а также непредумышленных, негативных воздействий заставляет в программных продуктах высокого качества разрабатывать методы и средства для обеспечения безопасности применения пользователями всего комплекса функциональных программ в течение его жизненного цикла. Их реализация в критических системах, может требовать значительных ресурсов памяти и производительности вычислительных средств. Из-за их ограниченности и других факторов априори невозможно обеспечить отсутствие проявления дефектов производства и абсолютную защиту крупных программных продуктов, вследствие чего безопасность их функционирования в системах имеет **всегда конечное, ограниченное значение**.

Наиболее полно **степень безопасности системы характеризуется величиной предотвращенного ущерба – риска**, возможного при проявлении дестабилизирующих факторов и реализации конкретных угроз безопасности применения продукта пользователями, а также средним временем между возможными проявлениями угроз, нарушающих безопасность [20, 39, 43]. С этой позиции **затраты экономических ресурсов** разработчиками и заказчиками на обеспечение безопасности должны быть соизмеримыми с возможным ущербом у пользователей от ее нарушения. Поэтому реализации угроз, в ряде случаев, целесообразно характеризовать интервалами времени между их проявлениями, нарушающими безопасность применения системы, или наработкой на отказы, отражающиеся на безопасности. Это сближает понятия и характеристики степени функциональной безопасности с показателями надежности систем. Различие состоит в том, что в показателях надежности учитываются все реализации отказов, а в характеристиках безопасности следует регистрировать только те катастрофические отказы, которые отразились на проявлении значительных рисков и нарушении безопасности.

При проектировании комплексов программ целесообразно разделять вычислительные ресурсы, необходимые для непосредственного решения основных, функциональных задач, и ресурсы, требующиеся для защиты и безопасного функционирования. Соотношение между этими видами ресурсов в реальных крупных продуктах зави-

сит от сложности и состава решаемых функциональных задач, степени их критичности и требований к безопасности всей системы. Реальные ограничения ресурсов, используемых в процессах производства, ограниченная квалификация специалистов, изменения внешней среды и требований заказчика объективно приводят к отклонениям процессов и реализации плана от предполагавшегося. В процессе формирования технического задания следует формулировать **основные положения методологии и план последовательного повышения безопасности** путем наращивания комплекса средств защиты, поэтапных испытаний компонентов и определения характеристик, допустимых для продолжения работ на следующих этапах.

Количественные характеристики качества сложных программных продуктов ниже сокращенно изложены в соответствии со стандартом **ISO 9126** [17, 45, 50]. Конструктивные характеристики могут быть разделены на две группы: количественные и качественные, которые различаются возможностями конкретизацией мер оценивания. Две группы стандартизированных характеристик качества программных продуктов – Надежность и Эффективность в наибольшей степени доступны количественным измерениям (см. рис. 5.1).

Надежность: свойства программного продукта обеспечивать достаточно низкую вероятность потери работоспособности – отказа в процессе его функционирования в реальном времени [17, 34, 50]. Основные атрибуты надежности могут быть объективно измерены и сопоставлены с требованиями. Требования к значениям атрибутов характеристики завершенность – допустимой наработки на отказ при тестировании, устанавливаются при отсутствии автоматического рестарта и при наличии администратора, контролирующего работоспособность программного продукта. Применением программно-аппаратных механизмов автоматического рестарта эта наработка при проявлении отказов может быть значительно повышена, так как при некоторых отказах возможно их автоматическое обнаружение и оперативное восстановление работоспособности, вследствие чего значения устойчивости и наработки на отказ возрастают. Допустимая длительность прерывания оперативной работы пользователей системы для полного восстановления нормального ее функционирования обычно может составлять несколько секунд или минут. Надежность программных продуктов наиболее полно характеризуется устойчивостью или способностью к безотказному функционированию, а также вос-

становливаемостью работоспособного состояния после произошедших сбоев или отказов.

Завершенность: свойство программного продукта не попадать в состояния отказов вследствие ошибок и дефектов в программах и данных. Завершенность можно характеризовать наработкой (длительностью) на отказ (при отсутствии автоматического восстановления – рестарта), измеряемой обычно часами времени.

Устойчивость к дефектам и ошибкам: свойство программного продукта автоматически поддерживать заданный уровень качества функционирования при проявлениях дефектов и ошибок или нарушениях установленного интерфейса. Эффективное, оперативное устранение проявления дефектов, ошибок и некорректного взаимодействия с операционной и внешней средой определяют характеристику – устойчивость программного продукта.

Восстанавливаемость: свойство программного продукта в случае отказа возобновлять требуемый уровень качества функционирования, а также поврежденные программы и данные. Основными показателями качества процесса восстановления являются его длительность (минуты, секунды) и вероятностные характеристики.

Доступность или готовность: свойство программного продукта быть в состоянии выполнять требуемую функцию в данный момент времени при заданных условиях использования. Для определения этой величины измеряется относительное время работоспособного состояния комплекса программ между последовательными отказами. Обобщение характеристик отказов и восстановления производится в критерии коэффициент готовности.

Нижняя граница допустимой надежности обычно отражается значениям, при которых резко уменьшается функциональная пригодность, и использование данного типа программного продукта становится неудобным, опасным или нерентабельным. Примером таких наихудших, предельных величин для многих классов программных продуктов могут быть наработка на отказ менее десяти часов, коэффициент готовности ниже 0,9 и время восстановления более десяти минут. С другой стороны, наилучшие значения этих атрибутов практически ограничены теми ресурсами, которые могут быть выделены для их достижения при производстве и эксплуатации.

В модели СОСОМО II экономику повышения **требуемой надежности** функционирования комплекса программ отражает харак-

теристика **M1** (рейтинги в диапазоне 0,82 – 1,26) (см. таблицы 9.2 и 9.3). Этот фактор участвует в выражениях для расчета экономических характеристик – трудоемкости и длительности производства программных продуктов, и может заметно их увеличивать. Для обеспечения высокой надежности затраты могут повышаться в полтора раза. Рейтинги для учета дополнительных затрат на повышении безопасности функционирования в этой модели не представлены, однако они могут быть соизмеримыми с совокупными затратами на обеспечение решения основных функциональных задач программного продукта в критических системах и составлять 10 – 20% от бюджета в ординарных проектах.

Эффективность: в стандарте **ISO 9126** отражена двумя характеристиками – временной эффективностью и используемостью ресурсов ЭВМ, которые рекомендуется описывать, в основном, количественными атрибутами, характеризующими динамику функционирования программного продукта. В этой стандартизированной характеристике отражается только **частная конструктивная эффективность** использования ресурсов ЭВМ, которую не следует смешивать с **системной эффективностью** функциональной пригодности программного продукта при применении в конкретной системе.

Основные требования к атрибутам характеристики эффективность использования вычислительных ресурсов программным продуктом, сосредоточены на наиболее критичных показателях производительности и длительности решения функциональных задач. Используемость ресурсов памяти и производительности вычислительных средств могут устанавливаться исходя, с одной стороны, из экономической целесообразности применения наиболее дешевой, с минимальными ресурсами ЭВМ, загрузка которой будет в среднем не ниже 0,5. С другой стороны высокая загрузка (выше 0,9) может приводить к задержке или даже потере заданий при случайном, кратковременном повышении их интенсивностей, что может негативно отражаться на применимости и рисках программного продукта.

Временная эффективность: свойства программного продукта, характеризующие требуемые времена отклика и обработки заданий, а также производительность решения задач с учетом количества используемых вычислительных ресурсов в установленных условиях (см. стандарт **ISO 14756**). Временная эффективность программного продукта определяется длительностью выполнения заданных функ-

ций и ожидания результатов в средних и/или наихудших случаях, с учетом приоритетов задач. Пропускная способность комплекса программ на конкретной ЭВМ отражается числом сообщений или заданий на решение определенных задач, обрабатываемых в единицу времени, зависящую от характеристик внешней среды.

Используемость ресурсов: степень загрузки доступных вычислительных ресурсов в течение заданного времени при выполнении функций программным продуктом в установленных условиях. Ресурсная экономичность отражается занятостью ресурсов центрального процессора, оперативной, внешней и виртуальной памяти, каналов ввода-вывода, терминалов и каналов сетей связи. Ресурсная экономия влияет не только на *стоимость решения функциональных задач*, но, зачастую, особенно для критических встраиваемых ЭВМ реального времени, определяет принципиальную возможность полноценного функционирования конкретного программного продукта в условиях реально ограниченных вычислительных ресурсов.

В модели СОСОМО II экономику *требуемой* временной эффективности и ограничения используемых ресурсов ЭВМ отражают характеристики двух факторов: фактор М6 – *ограничения времени исполнения* комплекса программ (рейтинги в диапазоне 1,00 – 1,63) и рейтинги М7 – *ограничения по оперативной памяти* использования комплексом программ (рейтинги в диапазоне 1,00 – 1,46) (см. таблицы 9.8 и 9.9). Эти факторы участвуют в выражениях для расчета экономических характеристик – трудоемкости и длительности производства программных продуктов и могут их изменять на 50 – 60%.

Три группы качественных (конструктивных) характеристик качества программного продукта – Практичность, Сопровождаемость и Мобильность трудно измерять количественно, и они доступны в основном *качественным оценкам их свойств* (см. рис. 5.1). В некоторых проектах для характеристик Сопровождаемости и Мобильности при проектировании могут доминировать *экономические меры* трудоемкости (человеко-часы) и длительности (часы) для процедур, обеспечивающих реализацию атрибутов этих характеристик в комплексе программ.

Практичность – применимость: свойства программного продукта, отражающие сложность его понимания, изучения и использования для квалифицированных пользователей при применении в заданных условиях. Требования к практичности и ее характеристикам –

понятности и простоте использования, зависят от назначения и функций программного продукта и могут формализоваться заказчиками набором свойств, необходимых для обеспечения удобной и комфортной эксплуатации продукта.

Понятность: свойства программного продукта, обеспечивающие пользователю понимание, является ли программа пригодной для его целей, и как ее можно использовать для конкретных задач и условий применения. Понятность зависит от качества документации и субъективных впечатлений от функций и характеристик продукта. Она должна обеспечиваться корректностью и полнотой описания исходной и результирующей информации, а также всех деталей функций программного продукта для пользователей.

Простота использования: возможность пользователю удобно и комфортно эксплуатировать и управлять программным продуктом. Эта характеристика учитывает физические и психологические особенности пользователей и отражает уровень контролируемости и комфортности условий эксплуатации программного продукта, возможность предотвращения ошибок пользователей.

Изучаемость: свойства программного продукта, обеспечивающие удобное освоение его применения достаточно квалифицированными пользователями. Она может определяться трудоемкостью и длительностью подготовки пользователя к полноценной эксплуатации продукта. Атрибуты изучаемости зависят от возможности предварительного обучения и от совершенствования знаний в процессе эксплуатации, от возможностей оперативной помощи и подсказки при использовании программного продукта, а так же от полноты, доступности и удобства использования руководств и инструкций по эксплуатации. Изучаемость можно отражать трудоемкостью и продолжительностью изучения пользователями соответствующей квалификации, методов и инструкций применения программного продукта для полноценной эксплуатации.

В модели СОСОМО II экономику **требуемой практичности** отражают архитектура и свойства комплекса программ, а также качество и **соответствие документации M5** (рейтинги в диапазоне 0,81 – 1,23) (см. таблицы 9.2 и 9.3). Этот фактор участвует в выражениях для прогнозирования экономических характеристик – трудоемкости и длительности производства программных продуктов и может их изменять приблизительно также как рейтинги надежности. Затраты на

достижение высокой практичности документации сложных программных продуктов могут достигать 20 – 30% общего бюджета проекта (см. главу 7).

Сопровождаемость: приспособленность программного продукта к модификации и изменению конфигурации. Модификации могут включать исправления, усовершенствования или адаптацию комплекса программ к изменениям во внешней среде применения, а также в требованиях и функциональных спецификациях заказчика [2, 17, 35]. Трудоемкость модификаций определяется внутренними метриками качества комплекса программ, которые отражаются на внешнем качестве и качестве в использовании, а также на сложности управления конфигурациями версий программного продукта (см. стандарты **ISO 14764** и **ISO 15846**).

Требования к сопровождаемости количественно можно установить для характеристик изменяемости и тестируемости **экономическими категориями** допустимой трудоемкости и длительности реализации этих задач при некоторых средних условиях, обусловленных необходимостью устранения дефектов и усовершенствованиями функций программного продукта. Для подготовки и выполнения каждого изменения (без учета затрат времени на обнаружение и локализацию дефекта) нужно устанавливать допустимую среднюю продолжительность и суммарную трудоемкость работ специалистов при их реализации.

Мобильность: подготовленность программного продукта к переносу из одной аппаратно-операционной среды в другую. Установление требований к мобильности может быть сведено к формализации трудоемкости и длительности процессов: адаптации к новым характеристикам пользователей и внешней среды, инсталляции версий программного продукта в среде пользователей и замены крупных компонентов по требованиям заказчиков или конкретных пользователей.

Гипотетические примеры определения требований к характеристикам качества двух типов программных продуктов

Разнообразие функций и характеристик качества программных средств делает **невозможной унификацию всей совокупности требований к характеристикам качества**, к их составу, значениям и

экономическим затратам для всех типов программных продуктов. Поэтому целесообразно ограничиться *гипотетическими примерами* формирования требований к характеристикам и значениям качества для *вариантов двух типов систем*. Эти типы комплексов программ (КП) выделены по принципу: наиболее высоких требований к качеству и требуемых ресурсов для их реализации – 1-й тип (СРВ); и организационного управления – 2-й тип (АС) административных систем, который характеризуется массовым применением и более низкими требованиями к показателям качества. Приводимые для них ниже значения требуемых характеристик качества являются экспертными оценками на основе опыта создания сложных программных продуктов, и их следует рассматривать *только как методические примеры и ориентиры*. Читателям целесообразно критически оценивать приводимые в таблицах данные и корректировать их с учетом особенностей собственных проектов.

В качестве примеров ниже за основу приняты сложные комплексы программ, приблизительно одинакового объема (~ 500 тыс. строк), с определенными функциями, которые можно предварительно характеризовать следующим образом (см. главу 4):

- 1-й тип – программные продукты управления динамическими объектами в жестком регламенте реального времени, с критическими наиболее высокими требованиями к качеству функционирования программ при ограниченных вычислительных ресурсах, которые применяются, например, для обработки информации в бортовых авиационных и космических системах;

- 2-й тип – программные продукты сложных административных систем, с десятками действующих операторов-пользователей, работающих в реальном времени, например, по схеме клиент-сервер, к которым предъявляются достаточно высокие требования к разнообразию и качеству решения функциональных задач при относительно больших вычислительных ресурсах, например, банковские, налоговые, таможенные и другие административные системы.

Стандартизированные в **ISO 9126** характеристики и атрибуты качества комплексов программ различаются по степени влияния на *функциональную пригодность*, системную эффективность их применения по прямому назначению, а также *по экономическим характеристикам при реализации*. Вследствие реальной ограниченности ресурсов, которые доступны в жизненном цикле, их необходимо вы-

делять с учетом влияния на функциональную пригодность, экономическую эффективность и интегральные риски. Для этого целесообразно в процессе системного анализа оценивать и учитывать уровень приоритета требований к каждому атрибуту качества, который отражает его влияние на функциональную пригодность конкретного типа комплекса программ.

Имеются значительные различия распределения приоритетов характеристик качества в зависимости от типов комплексов программ. Для 1-го типа высокими приоритетами характеризуются надежность и ресурсная эффективность, которые для остальных типов могут не иметь столь важного значения. Защищенность наиболее важна для продуктов 2-го типа вследствие возможности различных предумышленных угроз от объектов внешней среды, которые слабее влияют на программы других типов. Для 2-го типа могут быть достаточно приоритетными сопровождаемость и мобильность. Требования к характеристикам практичности отражаются средними и низкими приоритетами, которые несколько различаются в зависимости от типов продуктов. Подобную таблицу желательно подготавливать заказчиком и согласовывать с разработчиками, как дополнение к техническому заданию и использовать при выборе и реализации предельных значений требований к характеристикам качества при проектировании с учетом получающихся рисков.

Выбор требований к характеристикам качества, естественно, должен начинаться с определения необходимых свойств и значений группы показателей, отражающих функциональные возможности комплекса программ. Формирование *требований к функциональной пригодности* наиболее ответственная, стратегическая задача начальных этапов проектирования и всего последующего развития жизненного цикла. В представленном примере эта *задача не иллюстрируется*, вследствие разнообразия возможных вариантов назначений и функций программных продуктов.

Требования к характеристике *корректность* (стандарт ISO 9126) могут представляться в виде двух основных свойств, которым должны соответствовать программные продукты (таблица 5.1). Первое требование состоит в выполнении прослеживаемости реализации исходных требований технического задания и спецификации при верификации программных компонентов. Второе требование заключается в полноте покрытия тестами, достаточными для функционирования

ния программного продукта с необходимым качеством результатов, при ограничениях ресурсов.

Таблица 5.1

Пример требований к характеристикам функциональных возможностей для двух типов программных продуктов

Характеристика качества	Содержание требований	Требования по типам КП	
		СРВ	АС
Корректность	<i>Требуемая прослеживаемость со-ответствия текстов программ требованиям к функциям программных компонентов и КП в целом (%)</i>	95	80
	<i>Требуемая степень покрытия тестами структуры программных компонентов и КП в целом (%)</i>	95	80
	<i>Соответствие КП утвержденным документам на интерфейсы: с аппаратной, операционной и внешней средой системы (%)</i>	90	90
Способность к взаимодействию	<i>Соответствие КП утвержденным документам на интерфейсы с пользователями (%)</i>	80	90
	<i>Соответствие стандартизированным категориям защищенности КП (%)</i>	60	95
Защищенность - безопасность	<i>Использование выбранных и согласованных методов и средств обеспечения безопасности КП (%)</i>	80	95

Качество и полнота реализации этих требований должны обеспечивать функциональную пригодность, тем выше, чем больше возможен риск от дефектов функционирования программ. В соответствии с этим наиболее высокие требования к корректности должны предъявляться к продуктам 1-го типа и могут быть несколько ниже для других типов.

Требования к характеристике *способность к взаимодействию* могут быть достаточно полно формализованы и утверждены в процессе системного проектирования, с некоторыми уточнениями на последующих этапах. Их основой являются нормативные документы на интерфейсы Открытых систем или выбранные для конкретного проекта стандарты де-факто. При выборе элементов программных ком-

понентов, обеспечивающих способность к взаимодействию в конкретном проекте, следует учитывать величину вычислительных ресурсов, необходимых для их реализации. Чем более полно используются международные стандарты Открытых систем, тем шире способность к взаимодействию с различными повторно используемыми компонентами, однако требуются дополнительные затраты вычислительных ресурсов, что может быть не всегда целесообразно для комплексов программ 1-го типа. При формализации интерфейсов важно также учитывать перспективы продолжительного сопровождения и модификации множества версий продукта, возможность повторного использования их компонентов и переноса на различные платформы, что наиболее характерно для комплексов программ 2-го типа.

Выбор и формирование требований к характеристике *защищенность* должны осуществляться на основе потребностей эффективной реализации назначения и функций программного продукта. В процессе системного анализа и проектирования должны быть выявлены потенциальные угрозы функционированию и установлены требования к необходимому уровню защиты данного комплекса программ. В соответствие с допустимым уровнем риска заказчиком выбирается и фиксируется стандартизированная категория защищенности программного продукта и необходимый набор методов и средств защиты с учетом ограниченных ресурсов на их реализацию. Наибольшие предумышленные угрозы безопасности обычно возможны в программах 2-го типа, для которых должен быть наивысший уровень защиты. Требования к защите продуктов 1-го типа могут быть несколько ниже вследствие меньшей вероятности проявления предумышленных угроз от объектов внешней среды. В результате сформированные требования должны обеспечивать равнопрочную защиту от реальных угроз применению соответствующих типов программных продуктов и реализацию необходимых мер контроля и подтверждения целостности и характеристик качества функциональной пригодности.

Пример требований к основным *количественным характеристикам качества двух типов комплексов программ* представлен в таблице 5.2. *Надежность* является характеристикой качества, требования к которой наиболее сильно различаются для рассматриваемых типов программных продуктов.

**Пример требований к количественным характеристикам
для двух типов программных продуктов**

Характеристики качества	Мера	Требуемые значения по типам КП	
		СРВ	АС
Надежность			
Завершенность:			
- наработка на отказ при отсутствии рестарта.	Часы	100	10
Устойчивость:			
- наработка на отказ при наличии автоматического рестарта;	Часы	1000	50
- относительные ресурсы на обеспечение надежности и рестарта.	%	30	10
Восстанавливаемость:			
- длительность восстановления.	Минуты	0,1	5
Доступность-готовность:			
- относительное время работоспособного функционирования.	Вероятность	0,999	0,99
Эффективность			
Временная эффективность:			
- время отклика – получения результатов на типовое задание;	Секунды	2	10
- пропускная способность – число типовых заданий, исполняемых в единицу времени.	Число в минуту	100	10
Используемость ресурсов:			
- относительная величина использования ресурсов ЭВМ при нормальном функционировании программного средства.	Вероятность	> 0,9	> 0,8

При отсутствии автоматического рестарта в комплексе программ, и при наличии администратора, контролирующего работоспособность *программного продукта 2-го типа*, можно считать допустимым требованием к наработке на отказ порядка 10 часов. За счет программно-аппаратных механизмов автоматического рестарта эта наработка при проявлении отказов, может быть повышена приблизительно в 5 раз, т.е. можно предположить, что при 80% отказов, воз-

можно, их автоматическое обнаружение и оперативное восстановление, вследствие чего наработка на отказ (устойчивость) возрастет до 50 часов. По опыту на это может потребоваться около 10% вычислительных ресурсов системы. В примере предполагается, что для оперативной работы пользователей административной системой допустимая длительность прерывания работы для полного восстановления ее функционирования может составлять не более 5 минут. В результате при таких значениях атрибутов надежности, коэффициент готовности – вероятность застать программный продукт в работоспособном состоянии, может составить без рестарта достаточно высокую величину порядка 0,99.

Для *продуктов 1-го типа* требования к надежности часто могут быть значительно выше – наработка на отказ ~ 1000 часов и коэффициент готовности ~ 0,999, для чего может потребоваться до 30% вычислительных ресурсов реализующей ЭВМ. Основные атрибуты надежности при испытаниях могут быть объективно измерены количественно и сопоставлены с требованиями. При этом следует учитывать, что в примерах принято весьма малое время восстановления (секунды), обусловленное только мелкими программными дефектами без учета физического разрушения компонентов. Автоматизированное восстановление должно поддерживаться дублированием вычислительных средств и аппаратурным обеспечением надежности.

Основные требования к атрибутам характеристики *эффективность использования вычислительных ресурсов программным продуктом* сосредоточены на наиболее критичных показателях производительности и длительности решения функциональных задач. Для оперативной работы пользователей важно имеет малое время отклика из ЭВМ после получения типового задания и решения требуемой функциональной задачи. Это время обычно желательно иметь для комплексов программ 2-го типа в пределах нескольких (для примера принято пяти) секунд, хотя длительность полной реализации задания может быть значительно больше. Требуемая пропускная способность решения функциональных задач зависит от их содержания и числа действующих источников информации. В примере (2-й тип) предполагается, что десять операторов, могут вводить в минуту по два задания каждый, которые должны исполняться в отведенное время, что приводит к требованию пропускной способности данного продукта на выбранной вычислительной среде – 20 заданий в минуту.

Требования к используемости ресурсов памяти и производительности вычислительных средств могут устанавливаться исходя, с одной стороны, из экономической целесообразности применения наиболее дешевой, с минимальными ресурсами ЭВМ, загрузка которой будет в среднем $\sim 0,5$. С другой стороны высокая загрузка ($> 0,9$) может приводить к нежелательной задержке или даже потере заданий при случайном, кратковременном повышении интенсивностей потоков данных, что может негативно отразиться на функциональной пригодности.

Примеры требований к *качественным характеристикам программных продуктов* представлены в таблице 5.3. Требования к *практичности* и ее субхарактеристикам – понятности и простоты использования, зависят от назначения и функций программного продукта. Количественно, простоту использования можно в некоторой степени характеризовать требованиями ограничения средней длительности ввода типовых заданий и *времени отклика* на них, которое должно быть в несколько раз меньше. Эта характеристика коррелирована с временной эффективностью, однако, здесь рассматривается только с позиции удобства оперативного пользователя.

Требования к продолжительности изучения программного продукта 2-го типа, достаточной для эффективной эксплуатации сложной административной системы квалифицированным специалистом, в данном примере могут составить около недели или порядка 50 часов. Для коллектива из четырех человек-эксплуатационников это потребует трудоемкости около 200 человеко-часов. Для обеспечения полноценного изучения процессов применения программного продукта этими специалистами может потребоваться эксплуатационная документация объемом около 1000 страниц, а также желательны адекватные по содержанию и сложности электронные учебники. Для продуктов 1-го типа, вследствие более ограниченного взаимодействия с внешними пользователями, длительность изучения и объем эксплуатационной документации может быть значительно меньше (см. таблицу 5.3).

Требования к компонентам *сопровождаемости* количественно можно установить для характеристик изменяемости и тестируемости. Требуемые значения зависят от четкости концепции и архитектуры комплекса программ, от унифицированности внутренних, внешних и с пользователями интерфейсов, от качества технологической доку-

ментации, а также от возможных потребностей в модификации и инструментальной оснащенности ЖЦ комплексов программ данного типа.

Таблица 5.3

**Пример требований к качественным характеристикам
для двух типов программных продуктов**

Характеристики качества	Мера	Требуемые значения по типам КП	
		СРВ	АС
Практичность			
<i>Простота использования:</i>			
- среднее время ввода заданий;	Секунды	2	10
- среднее время отклика на задание.	Секунды	1	5
<i>Изучаемость:</i>			
- трудоемкость изучения применения комплекса программ;	Чел.-часы	50	200
- продолжительность изучения;	Часы	20	50
- объем эксплуатационной документации;	Страницы	100	1000
Сопровождаемость			
<i>Изменяемость:</i>			
- трудоемкость подготовки изменений;	Чел.-часы	20	10
- длительность подготовки изменений.	Часы	10	5
<i>Тестируемость:</i>			
- трудоемкость тестирования изменений;	Чел.-часы	40	20
- длительность тестирования изменений.	Часы	10	5
Мобильность			
<i>Адаптируемость:</i>			
- трудоемкость адаптации;	Чел.-часы	10	50
- длительность адаптации.	Часы	5	10
<i>Простота установки:</i>			
- трудоемкость инсталляции;	Чел.-часы	4	10
- длительность инсталляции.	Часы	2	5
<i>Замещаемость:</i>			
- трудоемкость замены компонентов;	Чел.-часы	20	50
- длительность замены компонентов.	Часы	10	10

Обобщенно это отражается на длительности и трудоемкости подготовки и реализации типовых изменений, обусловленных необходимостью устранения дефектов и усовершенствованиями функций

программного продукта. В рассматриваемом примере комплекса программ 2-го типа для подготовки и выполнения каждого изменения можно принять его среднюю продолжительность ~ 5 часов и суммарную трудоемкость двух специалистов ~ 10 человеко-часов (без учета затрат времени на обнаружение и локализацию дефекта). Требования к продолжительности тестирования таких изменений, могут составить так же ~ 5 часов, но трудоемкость может увеличиться до 20 человеко-часов, так как требуемый коллектив тестировщиков, может возрасти до трех-четырех специалистов. Изменяемость и тестируемость комплексов программ 1-го типа могут требовать несколько большего времени и труда, так как эти работы зачастую приходится выполнять в условиях ограниченных ресурсов по памяти и производительности объектной ЭВМ.

Выбор и установление требований к *мобильности* комплекса программ в данном примере сведены к трудоемкости и длительности процессов: адаптации к характеристикам пользователей и внешней среды, инсталляции версий в среде пользователей и замены крупных компонентов версий по требованиям заказчиков или конкретных пользователей. Наиболее простым и легко формализуемым из перечисленных процессов является инсталляция готовой версии комплекса программ 2-го типа с комплектом документации без дополнительных изменений на платформе пользователя, которая может требовать до 5 часов работы двух специалистов (10 человеко-часов). Более сложный процесс включает адаптацию программного продукта по формализованным инструкциям к специфической аппаратной и внешней среде конкретного пользователя, которая может потребовать приблизительно вдвое большего времени и в несколько раз (в примере = 5) большего числа специалистов. Еще более сложный и трудоемкий процесс замены крупных программных компонентов и перенос их на иную аппаратную и операционную платформу. Для этого процесса в примере требуется не менее 20 часов и коллектив около 5 человек (100 человеко-часов).

Рассмотренные *гипотетические примеры* требований к характеристикам качества программных продуктов разных типов, *могут служить методическими ориентирами* при анализе экономических факторов и реализации процессов установления требований к ним в технических заданиях и спецификациях. Обсуждение и согласование между заказчиком и разработчиками рациональных значений всего

ансамбля характеристик и их атрибутов качества позволяет избегать как экономически нецелесообразного завышения требований, так и снижения требований к отдельным характеристикам, которые могут негативно отразиться на функциональной пригодности. При этом важно учитывать ограниченность реальных экономических ресурсов при выборе свойств, мер и шкал характеристик в жизненном цикле программных комплексов и необходимость компромиссов между ними, вследствие многочисленных связей и взаимовлияний. Подобный анализ может отражаться на существенном снижении рисков, стоимости, трудоемкости и длительности производства программных продуктов и на *повышении экономической эффективности всего их жизненного цикла.*

Глава 6

ЭКОНОМИКА ПЕРЕНОСА И ПОВТОРНОГО ИСПОЛЬЗОВАНИЯ КОМПОНЕНТОВ ПРИ ПРОИЗВОДСТВЕ ПРОГРАММНЫХ ПРОДУКТОВ

Цели и организация переноса и повторного использования программных компонентов

На *экономические характеристики* проектирования и производства программных продуктов значительно влияет возможность использования апробированного задела из предыдущих реализованных проектов. Результаты системного анализа, применение функциональных и информационных моделей предметной области, формализация спецификаций требований, функциональная декомпозиция программных комплексов и последовательная детализация проектов позволяют *применять готовые производственные решения* в различных формах и сочетаниях. В программной инженерии во многих случаях имеется возможность повторного использования компонентов программ и данных на разных уровнях представления, относящихся к разным этапам их жизненного цикла, потенциально готовых к многократному применению в различной внешней и операционной среде, а также в различных сочетаниях их взаимодействия. В ряде проблемно-ориентированных областей, зачастую, имеется широкий спектр готовых компонентов, которые могут быть использованы в различных сочетаниях для решения типовых функциональных, задач. Таким образом, сложились *благоприятные условия для эффективного применения современных технологий производства программных продуктов путем переноса и повторного использования компонентов* (ПИК) на различных аппаратных и операционных платформах [12, 13, 41].

Во многих случаях компоненты, комплексы программ и информация баз данных создавались и разрабатываются до сих пор *без ори-*

ентации на последующее повторное использование и/или перенос на иную аппаратную и операционную платформы. Это обусловлено отсутствием у разработчиков стимулов доводить программы и их документацию до уровня завершаемых поставляемых продуктов, а также практическим отсутствием опыта и знаний о современных методах создания мобильных (переносимых) программных продуктов и компонентов высокого качества. Огромный парк *унаследованных* программных продуктов и их компонентов разрабатывался под конкретные проекты и платформы систем, и является потенциально мобильным только в пределах, расширяющихся по параметрам и ресурсам моделей ЭВМ определенной архитектуры.

На обеспечение мобильности и повторного использования компонентов, комплексов программ и данных направлена значительная часть методов и средств *современной программной инженерии*. Четкое разделение результатов работ, выполняемых на каждой стадии жизненного цикла комплекса программ, и определенные условия переходов между их этапами позволяют выделить следующие *уровни повторного использования компонентов*:

- модели предметной области и спецификаций требований, реализуемые разными способами на этапе проектирования комплекса программ;
- проектные спецификации требований к комплексу и/или компонентам на этапе их проектирования;
- исходные тексты программ на языках программирования, применявшиеся при разработке повторно используемых программных компонентов и комплексов;
- тесты проверки функционирования, повторно используемых программ, стандартизированных интерфейсов, комплексных тестов на соответствие требованиям к программному продукту.

Освоение методов и средств решения этих проблем, позволяет качественно изменять экономические характеристики процессов производства сложных комплексов программ и *значительно повышать производительность труда* специалистов при их разработке. Это активизировало в последние годы интерес специалистов к проблеме мобильности программ и данных во всех отраслях применения вычислительной техники. Создание новых программных продуктов на базе повторно используемых компонентов (ПИК) ниже рассматривается с двух позиций: *путем выбора и сборки готовых* ПИК на

определенной платформе и **путем переноса** их с других аппаратных и операционных платформ. Последнее стало особенно актуальным для современных административных систем государственного и регионального управления, управления отраслями и предприятиями промышленности, а также банковскими системами и в социальной сфере.

Требования заказчиков и пользователей по совершенствованию экономических характеристик и снижению затрат на информатизацию объектов и процессов отразились на формировании следующих основных **целей создания и повторного применения программных компонентов и модулей** – рис. 6.1:

- обеспечение сохранения инвестиций, вложенных в реализованные и апробированные программные комплексы и компоненты, в процессе развития, модификации и появления новых требований к ним, а также при совершенствовании архитектуры, возрастании ресурсов и функций аппаратных и операционных платформ;
- снижение трудоемкости, стоимости и длительности непосредственного проектирования и производства сложных программных продуктов и их компонентов;
- обеспечение возможности эффективного по экономическим показателям и качеству переноса апробированных программных компонентов и комплексов, с минимальными изменениями на различные операционные и аппаратные платформы;
- экономная реализация совместной работы и расширения функций программных продуктов во взаимодействии с другими комплексами программ при решении крупных целевых задач на различных локальных и распределенных платформах;
- обеспечение пользователям взаимодействия с программными продуктами в унифицированном стиле, облегчающем им переход к использованию новых систем или с расширенными функциями.

Высокие темпы роста основных ресурсов аппаратных средств ЭВМ, и сохраняющаяся потребность увеличения их использования со стороны заказчиков и пользователей, приводят к необходимости адекватного совершенствования технологий создания программных продуктов и баз данных. Увеличение затрат на производство повторно используемых компонентов должно компенсироваться сокращением затрат при производстве целостных комплексов программ и баз данных на их основе, по сравнению с полной разработкой аналогичных программных продуктов.

Экономика переноса и повторного использования компонентов при производстве программных продуктов включает:

- цели и организацию переноса на иные платформы и повторного использования программных компонентов и комплексов:
 - эффективное использование готовых технических и производственных решений для различных продуктов;
 - оценки экономической рентабельности повторно используемых программных компонентов;
 - стандартизацию и унификацию структуры и взаимодействия компонентов и комплексов программ;
 - повышение экономических характеристик процессов проектирования и производства программных продуктов;
- факторы, определяющие экономическую эффективность производства продуктов из готовых программных компонентов:
 - размер – масштаб программного продукта и кратность повторного использования компонентов;
 - методики производства и повторного применения готовых унифицированных компонентов и комплексов программ;
 - структурирование комплексов программ и технологию сборки готовых повторно используемых компонентов;
- оценивание экономической эффективности производства при повторном использовании программных компонентов:
 - выделение и оценивание изменения затрат производства за счет использования готовых программных компонентов;
 - оценивание трудоемкости и длительности производства продукта с учетом применения готовых компонентов;
- оценивание экономической эффективности производства программного продукта при повторном использовании информации баз данных;
- применение методов и комплекса стандартов интерфейсов Открытых систем при производстве программных продуктов.

Рис. 6.1

В каждом конкретном случае **необходима оценка экономической рентабельности применения мобильности** к компонентам с учетом вариантов, характеризующих комплекс программ и данных и их среду:

- комплексов программ и компонентов, а также операционной среды в целом, решающих все функциональные задачи определенной

сложной системы и полностью сохраняющих свою структуру на новой аппаратной платформе;

- автономных, крупных комплексов и массивов информации баз данных, решающих функциональные задачи во взаимодействии с существующими на новой аппаратной платформе операционными средствами;

- отдельных функциональных компонентов программ и массивов данных для расширения и совершенствования функций, ранее реализованных задач на той же аппаратной и операционной платформах.

Для достижения перечисленных целей и обеспечения мобильности требуются различные ресурсы при их реализации. Потребность в конкретных ресурсах и рентабельность их использования зависит от ряда параметров, которые образуют широкий спектр ситуаций для анализа и применения свойства мобильности комплексов программ и данных. Такими *ресурсами являются*:

- трудовые затраты специалистов и время на создание требований к унифицированным интерфейсам компонентов в программах, обеспечивающим их эффективную мобильность на определенные типы платформ, например, в соответствии с концепцией и стандартами Открытых систем;

- дополнительные ресурсы памяти и производительности вычислительных средств, необходимые для реализации и функционирования компонентов программ, обеспечивающих их мобильность, например, для реализации стандартизированных интерфейсов с внешней и внутренней средой;

- трудовые затраты специалистов и время на создание, приобретение и эксплуатацию инструментальных средств, автоматизирующих разработку и сопровождение мобильных компонентов и комплексов программ.

В сложных проектах повторное использование компонентов должно быть *систематизированным и планируемым*. Перед началом этапа проектирования разработчики должны выполнять поиск компонентов или комплексов, подходящих для повторного использования. Требования к системе могут изменяться с учетом свойств имеющихся компонентов, выбранных для повторного использования. Такой подход предполагает определенные компромиссы в реализации требований. Для обеспечения экономически эффективного производ-

ства и сокращения затрат, необходимо и соблюдать ряд принципов и правил структурного построения и повторного применения компонентов (см. рис. 6.1). Эти принципы и правила могут иметь особенности в различных проблемно-ориентированных областях. Однако их формализация и выполнение в конкретных проектах обеспечивают значительное **снижение трудоемкости и длительности производства** программных продуктов и их версий. Потеря гибкости архитектуры комплексов программ, некоторое возрастание ресурсов, необходимых для их реализации, обычно полностью компенсируются **повышением экономических характеристик** процессов проектирования и производства продуктов. Для этого рекомендуется соблюдать типовые требования к их архитектуре, содержащие:

- стандартизированную структуру (архитектуру) комплексов программ определенного класса;
- унифицированные правила структурного построения программных компонентов;
- стандартизированную структуру баз данных, обрабатываемых комплексами программ;
- унифицированные правила структурного построения информационных модулей, заполняющих базу данных;
- унифицированные правила структурного построения и организации межмодульного интерфейса программных компонентов;
- унифицированные правила внешнего интерфейса и взаимодействия компонентов программного продукта с внешней средой, с операционной системой и другими типовыми средствами организации вычислительного процесса и контроля в прикладных системах.

При переносе программного продукта на иную платформу его свойства всегда несколько изменяются, что следует учитывать при анализе целесообразности переноса, а также могут быть необходимы его тестирование, отладка, испытания и сертификация в новой среде. Следует также учитывать, что любой **перенос готовых компонентов и комплексов связан с затратами**, которые чаще всего требуются для:

- системного анализа рентабельности переноса на иную или ту же платформу и оценки экономических показателей этого процесса;
- реализации самого процесса переноса и интеграции компонентов и/или программного продукта с операционной и внешней средой на новой аппаратной платформе или в существующей среде;

- квалификационного тестирования, испытаний и комплексной проверки функционирования версии программного продукта в новом окружении или на новой платформе;
- сертификации перенесенных на новую платформу программных продуктов и функционирующих в иной операционной и внешней среде;
- корректировки или дополнения эксплуатационной и технологической документации на перенесенный программный продукт.

Однако *следует избегать излишнего оптимизма при оценке затрат на перенос*, так как при создании мобильных продуктов трудно предусмотреть все возможные особенности различных платформ и внешней среды, для которых декларируется мобильность конкретных программных средств. Эти особенности и возможное расширение окружающих прикладных программ и данных могут преподносить *неприятные сюрпризы нестыковки*, для ликвидации которых потребуются дополнительные затраты.

Факторы, определяющие экономическую эффективность производства программных продуктов на базе готовых компонентов

Проектирование систем *с повторно используемыми компонентами* в последнее время стало *особенно рентабельным для сложных программных продуктов*, содержащих сотни или тысячи модулей, и с большими объемами обрабатываемой информации. Выбор, выделение компонентов и комплексов для подготовки их повторного использования и/или переноса на другие аппаратные и операционные платформы зависит, прежде всего, *от их размеров и от кратности* возможного применения компонентов. При разработке комплексов небольшого масштаба (порядка нескольких тысяч строк исходного текста) поиск и подбор готовых компонентов для их применения в новом комплексе чаще всего оказываются не рентабельными. Таким образом, существует некоторый диапазон малых размеров комплексов программ, для которых не следует искать готовые компоненты и *нецелесообразно применять* ранее созданные программы и массивы данных.

Кратность применения компонентов значительно влияет на экономическую эффективность создания из них комплексов про-

грамм. При этом важно тщательно формулировать требования, осуществлять унификацию интерфейсов и оформление документации для тех компонентов, которые в перспективе будут использоваться многократно различными специалистами, в различных вариантах, на той же или на различных платформах. При анализе экономических характеристик производства конкретных комплексов трудно предвидеть кратность повторного использования определенных компонентов в будущих проектах. Поэтому обычно затраты на производство переносимых компонентов рассматриваются отдельно или включаются в экономические характеристики первой версии программного продукта, в которой они применены.

Для обеспечения переноса текстов программных компонентов необходимо при их первичной разработке **подготовить потенциальную возможность их последующего многократного использования** в различном операционном и внешнем окружении. Для этого должна быть унифицирована технология разработки модулей и групп программ, структура межмодульных связей и технология комплексирования, стандартизирована система идентификации и спецификации программ и данных, а также дисциплина испытаний и документирования компонентов и комплексов программ. Повторно используемые компоненты должны быть подготовлены для **отчуждения** от их первичных разработчиков и от комплексов программ, в составе которых они первично отлаживались, испытывались и применялись. Таким образом, возникает необходимость проведения ряда **общесистемных и организационных работ**, а также анализа некоторых дополнительных затрат на обеспечение возможности эффективного переноса программ и данных в различную внешнюю среду.

При этом проявляются следующие **преимущества повторного использования программных компонентов**:

- ускорение разработки – повторное использование компонентов ускоряет создание программных продуктов, так как сокращается время на их отбор и тестирование;
- повышение надежности – компоненты, повторно используемые в других системах, оказываются значительно надежнее новых компонентов, они протестированы и проверены в разных условиях работы, ошибки, допущенные при их проектировании и реализации, в основном обнаружены и устранены при первом их применении;

- уменьшение проектных рисков – для готовых ПИК можно более точно прогнозировать затраты, связанные с их повторным использованием, такой прогноз – важный фактор организации производства комплексов программ;

- более эффективное использование специалистов – часть специалистов, выполняющих одинаковую работу в разных проектах, может заниматься разработкой компонентов для их дальнейшего повторного использования, эффективно применяя накопленные ранее знания и опыт;

- интерфейсы компонентов, внешней среды и пользователей, можно реализовать в виде набора стандартных компонентов, что повышает надежность систем, так как, работая со знакомым интерфейсом, пользователи совершают меньше ошибок.

Процессы переноса ПИК и данных на иные платформы, выбор методов обеспечения мобильности комплексов программ, и характеристики используемых ресурсов для их реализации, прежде всего, зависят от параметров компонентов, предполагаемых для переноса. **Экономические ресурсы** требуются в той или иной степени на **двух фазах реализации требований** переноса и повторного использования комплексов и компонентов программ:

- при создании потенциально переносимых компонентов и комплексов программ, когда требования эффективной мобильности предусматриваются и реализуются при их проектировании и производстве, и определяются возможные платформы и области повторного применения таких программ и данных;

- при непосредственной реализации переноса компонентов и/или программных комплексов, в различной степени подготовленных для переноса на иные платформы и/или для повторного использования на той же платформе.

Для этого следует создавать методологию и технологию, а также применять стандарты, поддерживающие требования и разработку повторно используемых и/или переносимых (мобильных) программ и данных (см. стандарт **ISO 14764**). Основные особенности требований к компонентам и комплексам программ для повторного использования в системах **определяют две группы задач**. Первая задача – **структурирование** программ и данных на стадии проектирования систем, предполагающее последовательную декомпозицию заданных функций системы, что позволяет выделять программные компоненты

и модули, которые могут быть применены повторно как готовые, и описание их взаимодействия с другими компонентами. Вторая задача – **сборка или интеграция** готовых ПИК, а также комплексное, квалификационное тестирование и испытания программного продукта в целом. Для успешного, экономически эффективного, проектирования и производства крупных комплексов программ на базе ПИК должны выполняться **следующие основные условия:**

- возможность поиска необходимых программных компонентов – должен быть каталог документированных апробированных компонентов, предназначенных для повторного использования, который обеспечивал бы быстрый поиск нужных компонентов;

- необходимо удостовериться, что поведение ПИК предсказуемо и надежно, все компоненты, представленные в каталоге, должны быть сертифицированы, чтобы подтвердить соответствие определенным стандартам качества;

- на каждый ПИК должна быть соответствующая документация, чтобы можно было получить нужную информацию о компоненте и адаптировать его к новому применению, и информация о том, где и как используется данный компонент.

Формирование повторно используемых компонентов при производстве комплексов программ может осуществляться следующими методами:

- путем выделения в процессе производства комплекса программ некоторой доли повторяющихся компонентов, их унификация по содержанию, структуре и интерфейсам, при сохранении многих компонентов проекта с однократным применением;

- созданием стандарта и руководства, обязательного для оформления и производства основной массы программных компонентов (80 – 90%) с возможностью их применения как повторяющихся в определенном семействе комплексов программ, однако допускающем однократное применение уникальных компонентов с произвольным оформлением;

- проектирование и производство комплексов программ с унифицированной архитектурой, только из пригодных для повторного применения компонентов, стандартизированных по структуре и интерфейсам.

Для повторного использования текстов и тестов компонентов и комплексов программ целесообразно предусматривать:

- возможность встраивания готового ПИК в создаваемую новую систему при условии, что поставщик гарантирует его функционирование на выбранной платформе;
- перенос компонентов и/или комплексов программ и данных с платформ, в среде которых они были ранее реализованы, на выбранную для системы новую операционную и/или аппаратную платформу;
- обеспечение доступа к информационным ресурсам других распределенных систем и сетей.

Важная особенность программных компонентов, пригодных для повторного использования, это возможность *отделения спецификации требований и функций от реализации структуры текстов программы компонента*. Спецификация ПИК должна содержать функции компонента, и как он поведет себя, когда его возможности используются другими компонентами. Обладая этой спецификацией можно сосредоточиться на общем решении проблемы, не углубляясь в то, как реализованы функции и текст программы компонента. В общем случае могут существовать более одной реализации спецификации компонента – каждая со своими отличиями в определенных условиях, таких как платформа, производительность или стоимость. Изменения в реализации или подстановка одной реализации вместо другой не должны нарушать работу комплекса программ, которые реализованы с зависимостью только от спецификаций компонента.

Любые программные продукты функционируют в среде операционной системы (ОС) ЭВМ и типовых сервисных средств, расширяющих их эксплуатационные возможности. Эти системы в большинстве случаев универсальные, и применяются при создании различных комплексов. Поэтому далее многократное применение ОС и их дополнительных средств не учитывается. Эффективность повторного использования программных компонентов анализируется в пределах версий программных продуктов, создаваемых для решения конкретных функциональных задач в соответствии с техническим заданием.

Унификация всегда требует некоторых ресурсов, которые в данном случае, выражается в дополнительной трудоемкости создания ПИК, а также в увеличении необходимого объема памяти и производительности ЭВМ для их реализации. Развитие довольно широкого спектра архитектур ЭВМ, естественно привело к повторному использованию компонентов не только на однотипных платформах, но и к

производству комплексов программ и баз данных, переносимых на различные аппаратные и операционные платформы. В связи с этим выше выделялись *две технологические проблемы* [13, 26]:

- *создание* программных компонентов и баз данных, которые рентабельно повторно применять в различных проектах и/или переносить на различные операционные и аппаратные платформы;
- *применение* повторного использования и/или переноса компонентов программ и баз данных для создания из них новых программных систем на различных платформах.

При *анализе экономических характеристик и рентабельности* производства сложных комплексов программ на базе ПИК и/или путем их переноса с другой аппаратной и операционной платформы *в первую очередь необходимо учитывать*:

- степень подобия архитектуры и соотношения основных параметров ресурсов аппаратных платформ, между которыми предполагается перенос и /или применение готовых компонентов;
- уровень унификации интерфейсов приложений с операционными платформами, между которыми предполагается перенос готовых программ и данных;
- наличие или отсутствие при разработке исходных программ и информации баз данных, ориентации на будущий перенос и/или повторное использование на других платформах.

Различия операционных систем, принципов организации вычислительного процесса и контроля функционирования, а также драйверов ввода-вывода могут практически полностью исключить эффективный автоматизированный перенос соответствующей части *унаследованных готовых компонентов* между разнотипными по архитектуре ЭВМ. Важнейшим фактором, влияющим на рентабельность переноса, является *степень унификации интерфейсов* программных продуктов с операционной и внешней средой и с пользователями. Применение одних и тех же стандартизированных языков программирования и аттестованных компиляторов позволяет значительно сокращать затраты при переносе.

Различие архитектуры и ресурсов ЭВМ, между которыми предполагается перенос программ и данных, является важнейшим фактором, определяющим экономическую эффективность повторного использования компонентов и программных продуктов. Особое значение для переносимости имеет специфика реализации взаимодействия

с внешней средой и процедур ввода-вывода в ядре операционных систем, сопоставляемых ЭВМ. Программы, реализующие эти функции должны отражать специфику конкретной системы ввода-вывода и особенности источников и потребителей информации. Такая ориентация программ ввода-вывода может делать их практически не переносимыми и требовать выделения в автономные, заново разрабатываемые, автоматизировано непереносимые процедуры.

Применяемые при производстве языки программирования отражаются на эффективности переноса программ и данных тем сильнее, чем больше различается архитектура ЭВМ и чем ниже уровень языка. Повышение уровня языка программирования сопряжено с возрастанием универсальности текстов программ и уменьшением их машинной зависимости. Одновременно уменьшается эффективность объектного кода и расширяется исполняемый текст программ, получаемых в результате трансляции. Таким образом, за снижение трудоемкости производства программных продуктов путем переноса на различных языках, зачастую приходится расплачиваться увеличением памяти для их размещения и времени при исполнении на новой ЭВМ.

Особенности повторного использования крупных «коммерческих» программных продуктов могут увеличивать потенциальный выигрыш, получаемый от повторного использования. Некоторые типы «коммерческих» систем используются повторно на протяжении многих лет, например, базы данных. При этом уменьшаются риски, так как крупные «коммерческие» системы уже применяются и разработчики могут увидеть, удовлетворяют ли они предъявляемым к ним требованиям нового проекта. Для этого необходимо изучить интерфейсы комплекса программ и использовать их для организации взаимодействия с другими системными компонентами, также необходимо разработать архитектуру, которая поддерживала бы крупные «коммерческие» системы при совместной работе. Однако такие программные продукты представляют собой сложные системы и часто продаются как отдельные автономные продукты, которые вносит дополнительные проблемы при их применении. При интеграции таких крупных систем могут возникать **следующие проблемы**:

- недостаточный контроль над функциональностью и производительностью крупных продуктов;
- хотя считается, что их интерфейсы известны, не исключена

вероятность наличия скрытых операций, которые будут «пересекаться» с системными операциями новой платформы и системы;

- проблемы, связанные с организацией взаимодействия крупных комплексов, – иногда сложно подобрать продукты для совместной работы, поскольку каждый продукт разрабатывался на основе различных предположений по поводу его сопряжения и использования;

- отсутствие контроля за текущей модификацией таких крупных продуктов, – их производители принимают решения по изменению своих комплексов под давлением рынка, новые версии могут обладать дополнительной функциональностью, не поддерживаемой предыдущими версиями;

- уровень поддержки, оказываемой производителями крупных продуктов, варьируется в широких пределах, – поскольку эти системы распространяются свободно, поддержка производителей особенно важна в тех случаях, когда у разработчиков возникают проблемы, например, при тестировании, связанные с получением доступа к исходному коду и к технологической документации системы;

- несмотря на то, что производитель берет на себя обязательства по поддержке своих программных комплексов, изменение ситуации на рынке и экономических условий могут привести к тому, что ему станет трудно продолжать выполнение взятых обязательств по их сопровождению.

Оценивание экономической эффективности производства программных продуктов при повторном использовании компонентов

Разработчиков обычно, прежде всего, интересует задача проводить ли проектирование и производство *нового* сложного программного продукта по технологии полного жизненного цикла, или попытаться найти прототипы и воспользоваться готовыми апробированными решениями и компонентами. При этом критерием экономической эффективности использования готовых компонентов в первом приближении может быть относительное *число компонентов или доля от полной трудоемкости и длительности производства программного продукта*, которые сокращаются за счет повторного использования готовых апробированных компонентов. Для этого из

оценки полного производства нового продукта, должна быть выделена и исключена часть, которая обусловлена затратами на первичное создание ПИК и применение их готовыми в этом продукте [17]. Затраты на проектирование комплекса программ при этом практически не изменяются, сокращаются, в основном, на этапы программирования и автономного тестирования компонентов, а также частично на последующие этапы производства – комплексирование, документирование и испытания. На этих этапах сокращению затрат может способствовать также накопленный опыт создания предшествующих подобных комплексов программ. Однако на начальных и конечных этапах производства программного продукта трудно оценить степень влияние этого опыта и использования готовых компонентов.

Оценки изменения полной трудоемкости производства программного продукта за счет повторного использования компонентов зависят от доли готовых ПИК из полного состава программных компонентов в комплексе. Использование готовых программных компонентов позволяет сокращать или исключать из полных затрат на производство продукта затраты на программирование и тестирование компонентов, которые зависят от относительного их числа и объема (доли ПИК – D) в составе общего числа компонентов в комплексе программ. Кроме того, большое число ПИК может отражаться на сокращении затрат на этапах предварительного и детального проектирования, и также положительно влиять на уменьшение затрат на сборку, испытания и документирование программного продукта.

Для оценки экономической эффективности производства комплекса программ с применением ПИК необходимы характеристики трудоемкости в зависимости от этапов жизненного цикла и влияния их доли на общую трудоемкость. Эти данные частично получены экспериментально (таблица 6.1), а влияние доли ПИК – g_k (D) на другие k -е этапы производства приходится оценивать, используя **правдоподобные гипотезы** [3].

Предположим, что каждый k -й этап (из r этапов жизненного цикла) проектирования и производства при полном цикле создания продукта с необходимым набором ПИК, характеризуется относительной долей – q_k от полной трудоемкости C . На каждом этапе выделим долю труда, которую можно отнести к созданию и применению готовых используемых компонентов $g_k < 1$, а остальную часть будем считать относящейся к системному анализу, сборке и комплексиро-

ванию компонентов. В первом приближении можно предположить, что в зависимости от этапа производства величина g_k зависит от доли ПИК – *Д линейно* в некоторых пределах от 1 до g_k .

Таблица 6.1

**Распределение затрат по этапам производства
сложных комплексов программ реального времени**

<i>Этапы производства</i>	<i>Трудоем- кость, %</i>	<i>Длитель- ность, %</i>	<i>Численность специали- стов, % от средней</i>
1. Предварительное проектирование	8	20	40
2. Детальное проектирование	14	20	70
3. Программирование компонентов	22	16	140
4. Автономное тестирование компонентов	24	16	150
5. Интеграция и комплексная отладка	24	20	120
6. Испытания и документирование	8	8	100

Опыт производства предшествующих программных продуктов может достаточно сильно отражаться на относительной трудоемкости различных этапов разработки. Для определения относительного изменения трудоемкости при создании продукта с помощью ПИК необходимо учесть относительную роль, каждого этапа q_k , и долю снижения трудоемкости g_k (\mathcal{D}) на соответствующем этапе за счет ПИК:

$$C_i = \sum_{k=1}^r q_k g_k (\mathcal{D})$$

Предельный выигрыш в сокращении трудоемкости производства комплекса программ, собираемого из полного набора готовых ПИК, когда отсутствует необходимость разработки дополнительных программных компонентов ($D = 100\%$), упрощенно можно оценить на основе распределения затрат по этапам, разработки (см. таблицу 6.1) следующим образом [17]. Предположим, что трудоемкость сокращается *только за счет исключения этапов программирования и автономного тестирования программных компонентов* в процессе производства. Трудоемкость этих этапов для программных продуктов СРВ составляет около 46% (см. таблицу 6.1) от полной трудоемкости при отсутствии ПИК. Таким образом, суммарная трудоемкость в данном варианте уменьшается для этого класса продуктов при ($D = 100\%$) почти в 2 раза (рис. 6.2).

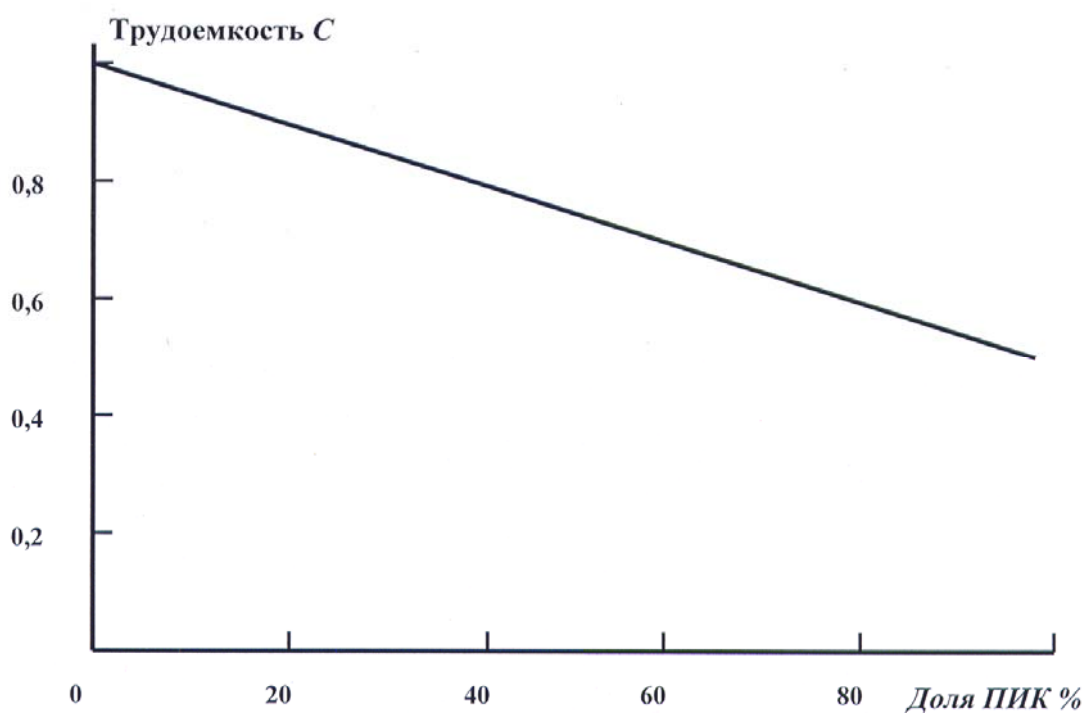


Рис.6.2

В процессе совершенствования производства комплекса с ПИК возможно некоторое дополнительное снижение затрат при детальном проектировании и комплексной отладке. Применение ПИК возможно и при глубокой функциональной преемственности последовательно разрабатываемых версий программных продуктов. В этих случаях доля затрат на системный анализ, детальное проектирование, комплексирование, отладку и испытания может быть значительно меньше приведенных оценок в таблице 6.1. Вследствие этого экономическая эффективность применения ПИК может соответственно, повысится.

На практике при создании нового программного продукта *не всегда имеется полный набор готовых и пригодных для применения программных компонентов*. Тогда при сборке продукта может потребоваться доработка отдельных компонентов, их сопряжение в новых сочетаниях и/или создание новых ПИК для решения дополнительных задач. Поэтому целесообразно оценить трудоемкость с учетом частичных затрат на новые компоненты. Относительное снижение трудоемкости производства в первом приближении пропорционально доле применения готовых ПИК. В пределе при производстве версии продукта полностью из многократно применяемых готовых компонентов, как показано выше, трудоемкость может сократиться в 2 раза. В промежуточных случаях, когда готовые компоненты используются частично, оценку изменения трудоемкости можно провести по степени сокращения затрат на программирование и автономное тестирование всех новых необходимых компонентов. В результате могут быть получены практически линейные зависимости эффективности изменения трудоемкости комплекса программ от доли ПИК для СРВ (см. рис. 6.2).

Изменение длительности производства комплексов программ за счет повторного использования компонентов относительно меньше, чем трудоемкости или средней производительности труда [3, 16]. Необходимость последовательного выполнения определенной совокупности производственных этапов и операций в заданной технологической системе остается более или менее постоянной при различных воздействиях на процесс производства (см. главу 3). Исключением является применение ПИК, при котором значительно сокращаются этапы программирования и автономного тестирования модулей и групп программ, а также, в той или иной степени, длительность других этапов. В *рассматриваемом упрощенном варианте* при наличии полного набора ПИК можно исключить программирование и отладку компонентов. В результате длительность разработки программных комплексов СРВ уменьшается приблизительно на 32% (в 1,5 раза) (см. таблицу 6.1 и рис. 6.3).

Если необходимо вновь создать некоторую часть программных компонентов, полная длительность разработки может мало измениться по сравнению с разработкой без ПИК. Это объясняется тем, что полная длительность программирования и автономной отладки компонентов относительно слабо зависит от того, какое количество

новых компонентов предстоит создавать. Поэтому зависимость продолжительности производства, от доли ПИК оказывается нелинейной, и заметное сокращение длительности разработки проявляется только при создании версии программного продукта практически полностью из готовых компонентов (см. рис. 6.3).

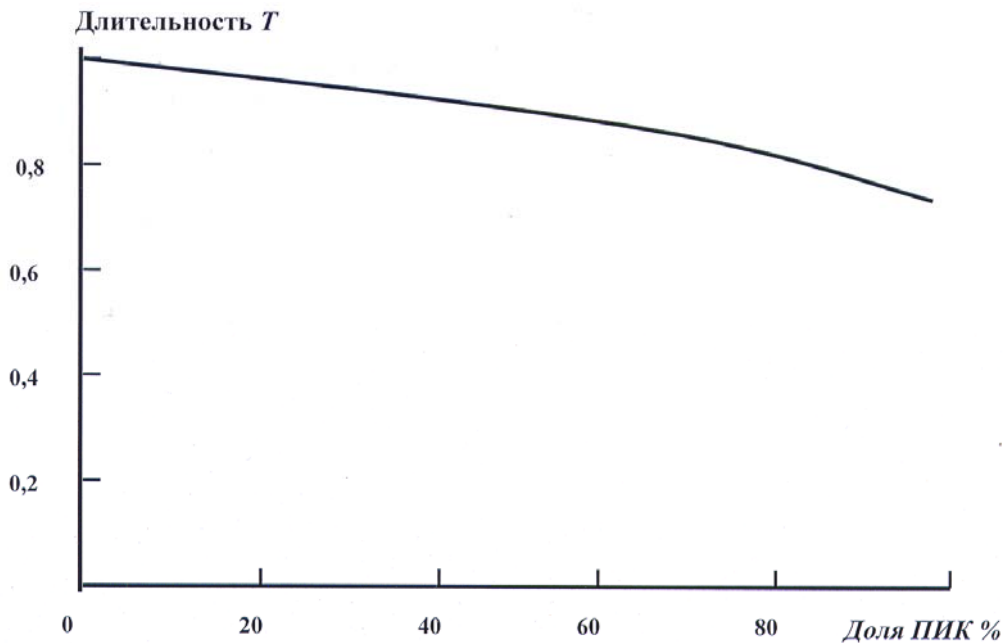


Рис.6.3

Относительное изменение длительностей производства, заметно меньше при повторном использовании всех компонентов, чем изменение трудоемкости. Это обусловлено сравнительно небольшой долей затрат времени на разработку ПИК. На этапах программирования и автономной отладки, рост необходимой трудоемкости обеспечивается в основном путем увеличения численности специалистов. В то же время, длительность начальных этапов, интеграции, комплексной отладки и испытаний нельзя практически ускорить, привлекая дополнительных специалистов. Эти этапы значительно сокращаются, только если использовать системный задел и прототипы, а также отработанные методы и средства при комплексной отладке. Тем не менее, эти обстоятельства слабее отражаются на изменении длительности, чем на изменении трудоемкости.

Наиболее широко ПИК применяются в административных, финансовых, социальных системах, для которых разработано множество крупных пакетов функциональных продуктов и могут собираться проблемно-ориентированные комплексы программ для конкрет-

ных заказчиков. В ряде случаев оказывается необходимой разработка и/или замена только небольшой доли новых ПИК, в основном для обеспечения интерфейсов с пользователями. В таких системах значительную роль могут играть первичные капиталовложения в производство технологии и аппаратного обеспечения средствами вычислительной техники создание и применения крупных ПИК. Эти виды затрат обычно сосредоточиваются в производстве первой версии программного продукта и в дальнейшем могут не учитываться. В некоторых случаях их можно распределять как амортизационные расходы по всем версиям продукта более или менее равномерно.

Производству комплексов программ с ПИК присущ **ряд недостатков и проблем**, которые препятствуют большому ожидаемому сокращению расходов на производство программных продуктов:

- недоступность исходного кода ПИК может приводить к увеличению расходов на сопровождение системы, так как повторно используемые системные элементы могут со временем оказаться не совместимыми с изменениями, производимыми в системе;
- некоторые разработчики предпочитают вместо ПИК, переписывать программные компоненты, так как полагают, что смогут при этом их усовершенствовать, и создание программ «с нуля» им кажется перспективнее повторного использования программ написанных другими специалистами;
- заполнение библиотеки компонентов и ее сопровождение может стоить дорого, так как еще недостаточно разработаны и применяются методы классификации, каталогизации и стандартизации информации о ПИК.

Оценивание экономической эффективности производства программных продуктов при повторном использовании информации баз данных

В ряде случаев особое значение имеет не столько использование готовых программных компонентов, сколько **перенос апробированных баз данных**. Информация о процессах, происходящих во внешней среде, может иметь большой объем и трудоемкость первичного накопления и актуализации, что определяет необходимость ее тщательного хранения. Возможны ситуации, когда подобные данные являются уникальными и невозстановливаемыми. Однако первичные

аппаратная или операционная платформы, в которых они первоначально накапливались и обрабатывались, может требовать замены. Формирование и заполнение информацией баз данных достаточно сложный и трудоемкий процесс, экономические характеристики которого сильно зависят от структуры, состава, объема, связности и других особенностей исходной информации. Для осуществления переноса данных большого объема необходимы методы и средства автоматизации, которые обеспечивают достаточную эффективность этого процесса [14, 36].

Простейший вариант переноса информации базы данных на иную аппаратную платформу реализуется, когда на обеих платформах имеются апробированные СУБД одного типа и версий. Однако чаще предстоящий *перенос информации базы данных (ИБД) не предусматривался при ее первичном формировании и продолжительном наполнении данными, и возникает после длительной эксплуатации* (унаследованные системы). Причиной обычно являются неудовлетворительные показатели качества функционирования базы данных, потребность в дополнительных ресурсах памяти и производительности ЭВМ, большое время реакции на запросы данных. Одновременно должно быть обеспечено сохранение или повышение качества функционирования базы данных на новой платформе. В общем случае при переносе информации баз данных можно выделить две принципиальные проблемы, которые различаются затратами, сложностью и методами решения.

Первая проблема сводится к обеспечению полного достоверного переноса информации базы данных (ИБД), накопленной в ЭВМ-1, на ЭВМ-2 с иной архитектурой. Форматы, кодировка, длина слова и другие особенности записи данных зависят от архитектуры ЭВМ-1 и типа СУБД и могут требоваться преобразования исходных данных для эффективного их использования после переноса в ЭВМ-2 с иной СУБД. *Экономические затраты и сложность переноса информации* базы данных в этом случае зависят прежде всего от ее характеристик, которые отражают степень форматной, лингвистической и физической совместимости содержания переносимой ИБД между рассматриваемыми платформами.

Вторая проблема заключается в обеспечении переноса функций и процедур конкретной системы управления базой данных. Для этого необходимо иметь полные и достоверные сведения о структуре

записей ИБД, их связях, взаимодействии и логике использования в исходной ЭВМ-1. Если одновременно с информацией БД переносятся полностью программные средства СУБД, то это может обеспечивать сохранение конкретных функций управления данными. Однако может потребоваться перенос ИБД на иную аппаратную платформу с другим типом СУБД. В соответствии с этим, СУБД может иметь особенности компонентов и операторов, осуществляющих упорядочивание, чтение и запись данных, а также их основную обработку. Перечисленные характеристики могут оказаться совокупностью факторов, существенно снижающих экономическую эффективность переноса информации базы данных.

Для средних и крупных проектов программных продуктов анализ экономической эффективности использования готовых компонентов целесообразно *оценивать полной трудоемкостью и длительностью переноса и их сопоставлением с новой полной разработкой программного продукта и ИБД* при использовании алгоритмического и системного задела. Кроме того, создаваемые программные продукты и базы данных следует оценивать по эффективности использования памяти и производительности новой ЭВМ-2. Такую оценку необходимо проводить с учетом возможного тиража программного продукта и перспективы длительной его эксплуатации. Ориентация на снижение затрат при использовании и переносе готовых компонент, комплексов программ и ИБД зачастую отражается значительными потерями в эффективности их эксплуатации вследствие увеличения размера программ и снижения производительности систем, особенно когда программные продукты длительно используются в реальном времени на многих экземплярах ЭВМ. С другой стороны, может оказаться необходимым учитывать перенос не единственной версии продукта, а развития множества версий некоторого проблемно-ориентированного класса или совокупности версий программных продуктов на целый спектр перспективных ЭВМ различной архитектуры. Тогда при *экономическом анализе* следует оценивать полный размер программ и ИБД, которые надо создать для ЭВМ-2 и последующих операционных и аппаратных платформ.

Таким образом, для первичного вывода о целесообразности использования готовых компонентов, переноса СУБД и информации БД *следует решать задачу сравнения достигаемого эффекта и затрат* для методов переноса и повторной разработки программных компонен-

тов и информации базы данных в конкретных условиях с учетом различных факторов системы. Эти задачи значительно упрощаются при одновременном сокращении затрат, за счет применения идеологии и концепции Открытых компьютерных систем, поддержанных комплексом международных стандартов и современных версий ОС, как стандартов де-факто [49].

Даже при предварительной разработке, потенциально мобильных сложных программных продуктов, баз данных и ПИК, затраты на их перенос могут быть весьма большими. Практически *всегда необходимо время и трудоемкость* на:

- первичный системный анализ целесообразности переноса ИБД на иную платформу;
- поиск, адаптацию и использование готовых для переноса компонентов и СУБД;
- оценку затрат с учетом стоимости приобретения и адаптации переносимых комплексов программ и информации баз данных;
- интегрирование ИБД в новой аппаратной и операционной среде;
- тестирование и испытания компонентов в комплексе с унаследованными программами и ИБД новой аппаратной и операционной среды.

Некоторые, казалось бы, второстепенные детали и факторы несовместимости с новой платформой переносимых программ и информации базы данных могут заметно отражаться на трудоемкости и длительности переноса. Особенно тщательно их следует анализировать в унаследованных системах, в которых относительно мелкие детали взаимодействия новых и старых программ и данных могут существенно *влиять на экономические характеристики переноса*. Среди этих показателей наиболее важным для реализации проекта и зависящим от большинства его особенностей и факторов является трудоемкость, непосредственно определяющая стоимость переноса программ и данных. Значения длительности переноса ИБД и числа участвующих специалистов взаимосвязаны и в некоторых пределах могут размениваться. Поэтому оценки этих показателей можно варьировать, и при недостаточном числе специалистов естественно возрастает длительность проектирования ИБД на новой платформе, хотя трудоемкость остается практически неизменной.

Применение методов и комплекса стандартов интерфейсов Открытых систем при производстве сложных программных продуктов

Основными целями создания и применения концепции, методов и стандартов Открытых систем являлось повышение общей *экономической эффективности производства* и функционирования комплексов программ, а также логической и технической совместимости их компонентов, для обеспечения повторного использования программ и данных. Для достижения этих целей развиваются и применяются различные проблемно-ориентированные технологии, и комплексы средств автоматизации производства комплексов программ, *базирующиеся на повторном использовании апробированных программных компонентов, комплексов и данных*, их эффективном переносе на различные аппаратные и операционные платформы и согласованном взаимодействии в распределенных системах. Концепция Открытых систем – отражена в совокупности международных стандартов, которые специфицируют интерфейсы, услуги и поддерживающие форматы для достижения взаимодействия и переносимости программных продуктов, данных и персонала, достаточные для того, чтобы обеспечивать [36, 49]:

- возможность переноса (*мобильность*) компонентов, комплексов программ и данных, разработанных должным образом, с минимальными изменениями и затратами на широкий диапазон аппаратных и операционных платформ;
- совместную работу (*интероперабельность*) с другими прикладными программными продуктами и системами на локальных и удаленных платформах;
- взаимодействие с пользователями в стиле, облегчающем последним переход от системы к системе (*мобильность пользователей*).

Открытые системы облегчают доступность для пользователей широкого круга программных продуктов и их компонентов, позволяют предприятиям *экономить средства* на документации и переподготовке персонала, а кроме того, делают их независимыми от конкретных поставщиков программных и аппаратных продуктов. Построение открытых систем из унифицированных по взаимодействию продуктов стимулирует конкуренцию среди поставщиков как по

соотношению цена/производительность, так и по функциональным возможностям. Профессионалы в области Открытых систем акцентируют усилия на поиске и создании гибкой, способной к наращиванию программной и информационной среды, что **базируется на трех направлениях стандартизации**:

- стандартизация аппаратных и операционных платформ;
- стандартизация методов и технологии обеспечения проектирования и производства программных комплексов;
- стандартизация интерфейсов программных компонентов между собой, с операционной и внешней средой.

Развивающаяся инфраструктура Открытых систем базируется на интеграции компонентов для удовлетворения требований стандартов на интерфейсы компонентов и комплексов программ, а также с операционной и внешней средой для последующего широкого применения. Идеология Открытых систем базируется на строгом соблюдении совокупности протоколов и стандартов де-юре и де-факто. По этой идеологии программные и аппаратные компоненты, должны отвечать **двум важнейшим требованиям**: переносимости и возможности согласованной, совместной работы с другими, возможно удаленными, компонентами. Это позволяет обеспечивать совместимость различных программных компонентов, комплексов, систем и средств передачи данных.

Целью является **максимально возможное повторное использование разработанных и апробированных программных и информационных компонентов** при расширении масштаба комплексов программ, изменении вычислительных аппаратных платформ, их операционных систем и процессов взаимодействия. Стандарты **открытых вычислительных систем** должны обеспечивать возможность относительно простого и экономически эффективного переноса апробированных программных продуктов и информации баз данных на различные типы аппаратных платформ за счет стандартизации процессов и интерфейсов взаимодействия программ с операционными системами ЭВМ. **Основной задачей** является транспортировка и перенос функций и процедур обработки информации, а также содержания баз данных между различными платформами, осуществляющими их обработку. Подобные обмены функциями, процедурами и данными имеют неоперативный характер и могут осуществляться при проектировании вне реального масштаба времени. Проблемы состоят

преимущественно в обеспечении сохранности апробированного функционального ядра текстов программ и информации баз данных при их переносе на иные аппаратные и операционные платформы для **снижения трудоемкости и длительности производства сложных программных продуктов.**

Технология Открытых систем для повторного использования компонентов и переноса комплексов программ на иные платформы наиболее сильно **отражается на сокращении затрат ресурсов** при создании крупных административных систем обработки информации (класс АС). При этом возможно снижение трудоемкости даже в несколько раз, существенное сокращение длительности и числа необходимых специалистов относительно значений при полной новой разработке без применения современной технологии переноса и ПИК. Экономическая эффективность переноса программных продуктов систем реального времени (СРВ) ниже, чем в предыдущем случае, вследствие высокой доли затрат на перепрограммирование компонентов и интерфейсов, а также на постановку и освоение технологии производства и испытаний в реальном времени на новой платформе.

Рядом зарубежных организаций и промышленных фирм под руководством **IEEE**, начиная с 1990 года ведется активная разработка последовательных версий стандартов интерфейсов Открытых систем **POSIX** (Portable operating system interfaces). В результате подготовлен комплексный проект фундаментального международного стандарта из четырех крупных частей **ISO 9945:1-4:2003 (IEEE 1003.1 – 2003)**. Цель документа – **стандартизация обеспечения переносимости программ на уровне исходных текстов.** В нем определены основные интерфейсы операционных систем и окружения, интерфейсы командного интерпретатора, а также программы общих утилит. **Три отдельных крупных тома** включают: базовые определения; системные интерфейсы; команды управления и сервисные программы (утилиты). Кроме того, имеется большой **четвертый том** общего обоснования выбранных решений системы **POSIX**. Важными свойствами разработанных программных интерфейсов являются целостность, модульность их построения и параметризуемость.

Стандарты открытых систем – POSIX регламентируют совокупность базовых, системных сервисов для **обеспечения унифицированных интерфейсов прикладных программ**, специфицированных для языка **Си**, командного языка и совокупности служебных про-

грамм. **Основная цель – сделать программы переносимыми** на уровне различных исходных языков. У каждого интерфейса компонентов программ существует вызывающая и вызываемая сторона, стандарты **POSIX** ориентированы преимущественно на формализацию вызывающей стороны. Мобильность приложений должна обеспечиваться благодаря применению большого числа стандартизированных системных интерфейсных сервисов и возможности динамического выяснения характеристик целевой платформы и подстройки под них интерфейсов компонентов и комплексов программ. При проектировании и производстве сложных программных продуктов целесообразно анализировать и выбирать подходящие группы рекомендаций открытых стандартов. При этом одновременно **повышаются экономические характеристики программных продуктов и их компонентов**, эффективность переноса на различные платформы, а главное качество технологии и результатов производства.

Таким образом, для обеспечения эффективного управления производством сложных комплексов программ необходимо **стандартизировать и соблюдать ряд требований архитектурного построения и интерфейсов комплексов и компонентов программ**. Эти требования могут иметь особенности для проектов в различных проблемно-ориентированных областях. Однако их стандартизация обеспечивает значительный эффект в снижении трудоемкости и длительности последующей разработки программных продуктов и их версий. Частичная потеря гибкости архитектуры, некоторое возрастание ресурсов, необходимых для их реализации, обычно полностью компенсируются повышением управляемости и экономических характеристик процессов производства, а также качества программных продуктов.

Глава 7

ПРОИЗВОДСТВЕННЫЕ ПРОЦЕССЫ, ОПРЕДЕЛЯЮЩИЕ ЭКОНОМИКУ СЛОЖНЫХ КОМПЛЕКСОВ ПРОГРАММ

Базовые производственные процессы сложных систем и комплексов программ

В предыдущих главах 3 – 6-й основное внимание сосредоточено на характеристиках процессов и объектов, определяющих *экономику проектирования* программных продуктов. Эта и последующая главы акцентированы на процессах *производства* и *экономических характеристиках* программных продуктов, удовлетворяющих требованиям заказчика. Эти процессы построены на основе общих принципов и стандартов организации производства сложных систем и продуктов, изложенных в главе 1 и в [6, 28, 37]. Они составляют 60 – 70% общей трудоемкости и длительности создания крупных программных продуктов. В последние годы имеется тенденция роста относительных затрат на проектирование, планирование, управление и испытания, а также на повторное использование программных компонентов. Вследствие этого может сокращаться до 40% *доля производственных затрат* на верификацию, тестирование, испытания и документирование сложных комплексов программ. Особенности этих процессов, определяют качество производства программных продуктов, и отражены в представленных ниже современных международных стандартах, в их содержании и рекомендациях, *обеспечивающих экономическую эффективность производства*.

Процессы жизненного цикла сложных компьютерных систем регламентируются стандартом ISO 15288 (см. Приложение). Стандартизируемые системы включают аппаратные компоненты, программные средства и их пользователей. Этот стандарт целесообразно использовать как головной, в *иерархии стандартов*, поддерживающих процессы и экономику жизненного цикла сложных программных комплексов. Стандарт содержит основы общих производ-

ственных процессов, составляющих жизненный цикл сложных систем и охватывают концепции и идеи, определяющие экономное производство систем, вплоть до момента их снятия с эксплуатации. Он включает процессы заказа и поставки системы, рекомендации для **экономической оценки** и совершенствования процессов их жизненного цикла. Процессы в данном стандарте образуют полное множество, из которого специалисты предприятия могут конструировать **базовые производственные процессы жизненного цикла конкретных систем** – рис. 7.1.

Процессы предприятия определяют его способность приобретать и поставлять продукцию в результате производства, их поддержки и контроля. Они должны обеспечивать ресурсы и инфраструктуру, необходимые для осуществления проектов и гарантировать достижение целей и исполнение обязательств по договорам с заказчиками. В стандарте в процессы предприятия **включено управление**:

- средой предприятия;
- инвестициями предприятия;
- процессами жизненного цикла продукции;
- экономическими и производственными ресурсами;
- качеством систем.

Процессы проекта рекомендуются для установления и выполнения планов, оценки фактических достижений в соответствии с планами и для контроля над выполнением проекта с требуемыми техническими и экономическими характеристиками. В стандарте термин **проект** используется в контексте описания процессов, связанных с планированием, оценкой и контролем производства. Планирование проекта состоит в составлении **экономически эффективного и выполнимого плана**, который определяет проектные задачи, результаты производства и отчетность, устанавливает графики контроля выполнения задач проекта, включая критерии достижения требуемого качества и ресурсы, необходимые для выполнения требований и задач.

Технологические процессы производства предназначены для определения требований к системе, для преобразования требований в полезную продукцию, для создания условий по выпуску продукции, отвечающей **техническим и экономическим требованиям заказчика**, по использованию продукции и ее функциональных возможно-

стей в соответствии с их назначением, а также для ликвидации продукции после ее снятия с эксплуатации.

Производственные процессы, определяющие экономику сложных комплексов программ, включают:

- базовые производственные процессы сложных систем и комплексов программ:
 - типовые производственные процессы предприятия;
 - производственные процессы проектов сложных систем;
 - основные стандартизированные группы базовых производственных процессов комплексов программ;
 - вспомогательные процессы и работы, поддерживающие жизненный цикл комплексов программ;
 - организационные процессы и управление жизненным циклом комплексов программ;
 - аттестацию производственных процессов жизненного цикла комплексов программ;
- производственные процессы обеспечения качества и верификации компонентов и комплексов программ:
 - комплексные системы обеспечения и контроля качества производства программных продуктов;
 - верификацию требований и тестов компонентов и комплексов программ;
- производственные процессы тестирования сложных комплексов программ:
 - выбор типов и применение эталонов для тестирования компонентов и комплексов программ;
 - классификацию категорий специалистов как источников ошибок в компонентах и комплексах программ;
 - связь стратегий последовательностей программирования компонентов, их тестирования и тестовых процедур;
 - удостоверение качества результатов тестирования компонентов и комплексов программ;
- производственные процессы документирования сложных комплексов программ:
 - цели и задачи документирования производства и применения программных продуктов;
 - задачи и особенности технологической и эксплуатационной документации программных продуктов;
 - организацию производства, изменения и хранения документации на программные продукты.

Рис. 7.1

Процессы должны позволять выявлять заинтересованных лиц, которые связаны с системой на протяжении жизненного цикла, а также их потребности и пожелания.

Технологические процессы составляют: определение и анализ требований заказчика, проектирование архитектуры, реализацию и комплексирование системы, а также верификацию, тестирование, аттестацию, документирование системы. Эти процессы должны обеспечивать условия для того, чтобы продукция были **нужной и полезной, экономически выгодной**, пригодной для обслуживания, производства и применения, а также обладала другими конкретными качествами, необходимыми для того, чтобы удовлетворять требования заказчика, пользователей и поставщика. **Технологические процессы** в стандарте регламентируют:

- определение требований заказчика;
- анализ и согласование требований заказчика и разработчиков;
- проектирование архитектуры системы;
- реализацию проекта, производство компонентов и системы;
- комплексирование (интеграцию) компонентов системы;
- верификацию и тестирование компонентов и системы;
- валидацию (аттестацию) системы;
- испытания и контроль функционирования системы;
- передачу системы заказчику и пользователям;
- сопровождение системы;
- прекращение эксплуатации и ликвидацию системы.

В приложениях к стандарту обращается внимание на необходимость **адаптации рекомендуемых производственных процессов** к конкретным характеристикам предприятия и проекта системы, изложены базовые концепции стандарта **ISO 15288**, его соответствие и взаимосвязь со стандартом жизненного цикла программных средств **ISO 12207**.

Стандарт **ISO 12207** – наиболее полно (на уровне международных стандартов) отражает **технология проектирования, производства и обеспечения качества сложных программных продуктов** (см. рис. 7.1). Жизненный цикл комплексов программ представлен набором этапов, частных работ и операций в последовательности их выполнения и взаимосвязи, регламентирующих **экономически эффективное** проектирование и производство на всех этапах от подго-

товки технического задания до завершения испытаний ряда версий и окончания эксплуатации программного продукта. В ЖЦ включены описания исходной информации, способы выполнения операций и работ, устанавливаются требования к результатам и правилам их контроля, а также к содержанию технологических и эксплуатационных документов. Определяется организационная структура коллективов, распределение и планирование работ, а также контроль за реализацией ЖЦ. Компоненты стандарта **ISO 12207** и архитектура его разделов непосредственно использованы при детализации структуры и содержания стандартов: управления программными проектами – **ISO 16326**; сопровождения и модификации комплексов программ – **ISO 14764**; управления их конфигурацией – **ISO 15846**; документирования программных средств – **ISO 15910** и **ISO 18019** (см. Приложение).

Стандарт **ISO 12207** может использоваться как непосредственный директивный, руководящий или как рекомендательный документ, а также как организационная база при создании *экономически эффективных* процессов и средств автоматизации соответствующих технологических этапов производства. Для реализации положений стандарта должны быть разработаны или выбраны инструментальные средства, совместно образующие взаимосвязанный комплекс технологической поддержки и автоматизации производства и не противоречащие предварительно скомпонованному набору процессов. Имеющиеся в стандарте пробелы следует заполнять спецификациями или нормативными документами, регламентирующими применение выбранных или созданных средств автоматизации производства и документирования комплексов программ.

Этот стандарт определяет *набор проектных и производственных процессов*, ориентированных на большие и сложные программные продукты. Стандарт может быть адаптирован и применен к программному проекту любого типа, размера и сложности. Он может быть использован для программных комплексов, являющихся самостоятельными объектами или частями системы более высокого уровня. Производственные процессы, работы и задачи в стандарте описаны в естественной последовательности этапов жизненного цикла комплексов программ. Описанная последовательность предназначена для того, чтобы в конкретном проекте выбрать, упорядочить и применить подходящие для него задачи, процессы, работы, виды деятельности.

Архитектура ЖЦ комплексов программ в стандарте основана на **трех группах базовых производственных процессов**, в которые входят (рис. 7.2):

- основные производственные процессы жизненного цикла, базовые этапы и работы (раздел 5);
- вспомогательные процессы и работы, поддерживающие жизненный цикл комплексов программ (раздел 6);
- организационные процессы и управление жизненным циклом комплексов программ (раздел 7).

Стандарт ISO 12207:1995 –

Процессы жизненного цикла программных средств

**5. Основные процессы
жизненного цикла**

- 5.1. Приобретение – заказ
- 5.2. Поставка
- 5.3. Разработка – производство
- 5.4. Эксплуатация
- 5.5. Сопровождение

**7. Организационные процессы
жизненного цикла**

- 7.1. Управление проектом
- 7.2. Создание инфраструктуры
- 7.3. Усовершенствование
- 7.4. Обучение

**6. Вспомогательные процессы
жизненного цикла**

- 6.1. Документирование
- 6.2. Конфигурационное управление
- 6.3. Обеспечение качества
- 6.4. Верификация
- 6.5. Контроль соответствия – аттестация
- 6.6. Совместный анализ
- 6.7. Аудит
- 6.8. Решение проблем – устранение дефектов

Рис. 7.2

Эти базовые разделы стандарта состоят из ряда подразделов, в которых подробно раскрывается содержание основные работы, и комментируются особенности их выполнения. Рекомендации к каждому подразделу состоят в среднем из 3 – 6 пунктов – производственных работ (действий). Общее число работ и комментариев к ним в стандарте свыше 220.

В *разделе 5* изложены основы жизненного цикла и рекомендации по подготовке, производству, эксплуатации и сопровождению программных комплексов. Процессы приобретения и/или подготовки к созданию комплекса программ должны начинаться с инициализации проекта, анализа концепции, анализа рынка продуктов, выработки требований и состава поддерживающих документов, создания предварительного плана проекта. Следует оценивать отклики предприятий на предложение по созданию проекта, заключить контракт, спланировать жизненный цикл, организовать поддержку разработки отчетами, а также условия сдачи заказчику и завершения проекта.

Основные работы по созданию сложного комплекса программ рекомендуется начинать с определения состава базовых документов, выбора средств конфигурационного управления и обеспечения качества, а также выбора методов и средств технологического обеспечения производства всей системы. Следует проанализировать и формализовать системные требования и критерии качества системы: функциональные, пользовательские, безопасности, интерфейсов с внешней средой, сопровождаемости. На этой базе должна проектироваться архитектура всей системы, выделяться и анализироваться требования к программному продукту. Рекомендуется при определении и формировании характеристик качества программного продукта руководствоваться стандартами **ISO 9126** и **ISO 25000** и представленной в них номенклатурой характеристик (см. главу 5). Кодирование и тестирование каждого компонента должно быть оформлено совокупностью документов, удостоверяющих соответствие компонента первичной спецификации требований, содержащих тесты и результаты тестирования.

Рекомендуется разрабатывать план работ, включающий комплексирование компонентов, тестирование по всем разделам требований заказчика и показателям качества, а также документирование плана, результатов интеграции, использованных тестов, критериев оценки и полученных результатов. Далее комплекс программ следует подвергать квалификационному (аттестационному) тестированию по всем разделам требований контракта, при широком варьировании тестов, изменениях значений критериев, а также тестировать полноту и адекватность технологической и эксплуатационной документации реальному программному продукту. Проверенный таким образом программный продукт интегрируется в вычислительные средства

системы, средства визуализации и телекоммуникации. После объединения всех средств, система должна подвергаться квалификационному тестированию и испытаниям на соответствие совокупности требований, а также производиться оформление и проверка полного комплекта документации.

Вспомогательные технологические работы, поддерживающие жизненный цикл комплекса программ, и рекомендации по их выполнению изложены в **разделе 6** стандарта. Процессы документирования должны охватывать планирование, учитывать рекомендации по стандартизации, проектированию и разработке, а также по производству, конфигурационному управлению и сопровождению комплекта документации на комплексы программ. Конфигурационное управление предлагается включать в общий план управления проектом с процедурами конфигурационной идентификации, контроля, учета, отчетности и развития конфигурации. Для обеспечения гарантий качества продукта следует использовать планирование, методологию, процедуры и стандарты обеспечения качества в соответствии с контрактом и с учетом доступных ресурсов. Верификация компонентов и комплекса программ должна включать ее организацию, планирование и технологическое обеспечение. Представлена структура раздела контракта на верификацию, содержание процесса, состав требований, проектирование процесса тестирования, обобщение и документирование результатов. Валидация (аттестация) – удостоверение правильности должна гарантировать полное соответствие программного продукта спецификациям, требованиям и документации, а также возможность его надежного функционирования и безопасного применения пользователями. Рекомендуется ее выполнять путем тестирования и испытаний во всех возможных ситуациях исходных данных и проводить независимыми специалистами.

Управление проектом должно быть сосредоточено, в основном, в подготовке и обеспечении планирования и управления ресурсами, персоналом, аппаратурой, программными средствами и инструментарием. Процессы ревизии – аудита служат для установления соответствия реальных работ и отчетов требованиям, планам и контракту. В процессе решения технологических задач должны выявляться и регистрировать проблемы и дефекты применения программных продуктов и их функционирования.

Каждые дефект или ошибка должны быть определены, идентифицированы, описаны, проанализированы и устранены в процессе производства и сопровождения в соответствии с контрактом.

Организации жизненного цикла комплексов программ посвящен *раздел 7* стандарта. Он включает основные работы по управлению проектом, производством и инструментальными средствами. Процессы формирования инфраструктуры проекта должны состоять из выбора и установления аппаратных и программных средств, технологии, стандартов и обслуживания, используемых для проектирования и производства программных продуктов. Процессы совершенствования жизненного цикла состоят в установлении, оценивании, измерении, контроле и корректировке процессов жизненного цикла конкретных комплексов программ. Совершенствование жизненного цикла комплексов программ должно учитывать требования пользователей и развитие определенной технологии производства. Процессы обучения определяются требованиями к проекту и специалистам, должны учитывать необходимые ресурсы, управление и технические средства, облегчающие обучение эксплуатации программного продукта.

Изменения (улучшения, редакции) стандарта ISO 12207 изложены в дополнении **ISO 12207:AMENDMENT1**. В нем содержатся корректировки фрагментов этого стандарта, и набор Приложений модернизаций, новых производственных процессов и рекомендаций для базового стандарта. Все Приложения базируются на обширной таблице **E1**, в которой представлены рекомендуемые документы – источники процессов жизненного цикла комплексов программ, их наименование и тип, детализирующиеся в последующих Приложениях **F; G; H** этого документа. Новые, а также расширенные комментарии и рекомендации *к каждому производственному процессу* сложных комплексов программ, структурированы и изложены в составе четких подразделов: *цель* и *результаты*, которые конкретизируют и существенно улучшают возможность и удобство применения пользователями *рекомендаций экономически эффективных производственных процессов* базового стандарта жизненного цикла. В Приложении **F** подробно прокомментированы *цели и результаты*, существующих в стандарте **ISO 12207** процессов: базовых (5 раздел), вспомогательных (6 раздел) и организационных (7 раздел). Особое внимание уделено *организационным процессам* и их новым фрагментам. В При-

ложении **G** детально изложены *новые*, рекомендуемые процессы, отсутствовавшие в предшествовавших версиях стандартов:

- организация и *управление экономически эффективной деятельностью специалистов* в процессах жизненного цикла комплексов программ;
- использование процессов измерения целей и результатов деятельности специалистов в ЖЦ комплексов программ;
- оценки необходимых человеческих ресурсов коллективов разработчиков сложных комплексов программ;
- анализ и *управление качеством* деятельности и процессов в ЖЦ;
- управление качеством *при повторном использовании* программных компонентов;
- цели и действия при *организации технологического обеспечения* проектов сложных комплексов программ.

Стандарт ISO 12207:AMENDMENT 2 – Изменения (улучшения, редакция) стандарта **ISO 12207** – содержит *редакционные коррективы* и дополнения описаний некоторых процессов в **AMENDMENT 1**.

Оценка (аттестация) процессов жизненного цикла программных средств – стандарт **ISO 15504:1-5** – содержит рекомендации по *аттестации качества производственных процессов комплексов программ* с целью:

- выяснения состояния и качества его собственных производственных процессов для их усовершенствования;
- определения пригодности процессов производства для выполнения определенного требования или класса требований заказчика к качеству программных продуктов;
- определения качества и пригодности производственных процессов конкретного предприятия для определенного договора или класса договоров на ЖЦ комплексов программ.

Аттестация реализации ЖЦ направлена на обеспечение *адекватности управления производственными процессами комплексов программ* и должна учитывать внешнюю среду, в которой выполняются аттестуемые процессы. Чтобы предприятие могло улучшить качество своей продукции, оно должно иметь проверенный, последовательный и надежный метод для аттестации состояния своих процессов, а также иметь средства использования ее результатов как часть

Программы усовершенствования производственных процессов. Использование аттестации производственных процессов внутри предприятия должно способствовать выработке культуры постоянного совершенствования и повышения характеристик качества программных продуктов, а также соответствующих механизмов поддержания этой культуры и оптимизации использования ресурсов. Это должно приводить к появлению зрелых предприятий, максимально восприимчивых к возрастающим требованиям потребителей и рынка, обеспечивающих **высокие экономические характеристики**, минимальную стоимость полного жизненного цикла своей продукции и, как результат, максимально удовлетворяющих требования заказчика и пользователей к качеству продуктов [29].

Покупателям и заказчикам программных продуктов выгодно использование аттестацию процессов ЖЦ при **определении производственной зрелости поставщика**, что:

- уменьшит неопределенность при выборе поставщиков программных комплексов за счет того, что риски, связанные с реальной зрелостью подрядчика, выявляются еще до заключения договора;
- позволит заранее предусмотреть необходимые меры на случай возникновения рискованного события;
- предоставит количественные критерии выбора при сопоставлении потребностей бизнеса, **требований и оценочной стоимости проекта** с реальной зрелостью конкурирующих поставщиков;
- приведет к общему пониманию необходимости использования результатов аттестации для усовершенствования процессов, оценки **экономической и технологической зрелости** поставщика при прогнозировании характеристик производства комплексов программ.

Это помогает определить, **эффективны ли экономически производственные процессы** для достижения заданных целей проекта программного продукта, а также выявить существенные причины недостаточного качества продукции, **риски превышения бюджет или сроков**. Все вместе позволяет расставить приоритеты при совершенствовании производственных процессов.

Стандарт **ISO 15271 – Руководство по применению ISO 12207** – поддерживает его **практическое использование**. Оно содержит подробные рекомендации по внедрению, применению в проектах комплексов программ, а также при организации производственных

работ и реализации требований стандарта. Изложены рекомендации по формализации методов внедрения, по подготовке и утверждению плана работ, по адаптации требований стандарта к конкретным особенностям характеристик ЖЦ и внешней среды конкретного комплекса программ, по его сопровождению. Перечислены и прокомментированы особенности и факторы, которые следует учитывать при применении стандарта в различных проектах, связанные: с используемой моделью жизненного цикла системы и комплекса программ; со стратегией и процедурами функционирования заинтересованных организаций заказчика и поставщика; с характеристиками системы и типом комплекса программ; с особенностями его сопровождения.

Производственные процессы обеспечения качества и верификации компонентов и комплексов программ

На *экономические характеристики производства* большое влияние оказывают требования заказчиков по обеспечению качества, надежности и безопасности применения программных продуктов. Эти требования могут приводить к *увеличению затрат* даже в несколько раз. В последнее время вследствие роста сфер применения и ответственности функций, выполняемых комплексами программ, резко возросла необходимость *гарантирования высокого качества программных продуктов*, регламентирования и корректного формирования требований к характеристикам качества и их достоверного определения (см. рис. 7.1). К сожалению, во многих случаях реальные контракты и предварительные планы на создание сложных программных продуктов для критических систем подготавливаются и оцениваются неквалифицированно, на основе неформализованных представлений заказчиков и разработчиков о требуемых функциях, характеристиках качества систем и *экономических ресурсах* для их реализации.

Значительные системные и технологические ошибки при определении требуемых показателей качества, оценке трудоемкости, стоимости и длительности производства являются достаточно массовыми и типичными. Многие созданные *системы не способны выполнять полностью требуемые функциональные задачи с гарантированным качеством* и их приходится долго и иногда безуспешно дорабатывать для достижения необходимого качества и надежности функ-

ционирования, затрачивая дополнительно большие средства и время. В результате часто проекты не соответствуют исходному, декларированному назначению и первоначальным спецификациям требований к характеристикам качества, не укладываются в согласованные графики и бюджет разработки.

В технических заданиях и реализованных программных продуктах систематически *умалчиваются и/или недостаточно формализуются понятия и метрики требуемого качества продукта*, какими характеристиками они описываются, как их следует измерять и сравнивать с требованиями, отраженными в контракте, техническом задании или спецификациях. Кроме того, некоторые из характеристик часто вообще отсутствуют в требованиях и согласованных документах на продукт, что приводит к произвольному их учету или к пропуску при испытаниях [5, 13, 15, 40]. Этому способствует *ограниченность экономических ресурсов*, особенно времени, необходимых для достижения и оценивания, требуемых значений характеристик качества, а также недостаточная формализация и документирование всего процесса выбора, контроля и анализа качества.

Качество продукта или процесса зависит от того, для какой *цели*, для какого *потребителя* и для каких *условий* делается его оценка. Один и тот же объект может иметь несколько различных оценок качества, произведенных для различных целей и разных условий определения. При измерениях и оценках, качество должно рассматриваться как иерархическая совокупность свойств, расположенных на различных уровнях. Каждое из свойств на одном уровне зависит от ряда других свойств, лежащих на более низких уровнях. Число уровней свойств по мере углубления знаний о конкретной продукции или процессе может возрастать. Изучение взаимосвязи между свойствами, входящими в состав обобщенного качества должно теоретически обосновать правомочность его предварительной декомпозиции для целей последующего объединения оценок отдельных свойств в комплексные оценки качества. Для получения комплексной оценки может использоваться экспертное определение весомости (приоритета) каждого свойства и в первую очередь должно учитываться влияние этого свойства на *экономику и эффективность использования данного вида продукции*.

Радикальное повышение качества производства программных продуктов и обеспечение их конкурентоспособности на мировом

рынке возможно только на базе внедрения современных **стандартизированных технологий и систем качества**, поддерживающих и контролирующих весь их жизненный цикл. Тщательное специфицирование и оценивание характеристик качества программных продуктов – ключевой фактор обеспечения их адекватного применения. Это может быть достигнуто на основе выделения и определения подходящих характеристик с учетом целей использования и функциональных задач продукта. Применительно к программным комплексам **система обеспечения качества производства** – это совокупность методов и средств организации управляющих и производственных подразделений предприятия, участвующих в проектировании, производстве и сопровождении комплексов программ с целью придания им свойств, обеспечивающих удовлетворение потребностей заказчиков и потребителей при минимальном или допустимом расходовании ресурсов. Различия фактических и требуемых характеристик качества объектов или процессов должно квалифицироваться как дефекты или ошибки и являются первичными стимулами для принятия и реализации решений по изменению определяемых значений качества. Для этого **необходимы экономические ресурсы**, а также **воля** руководителей, организация исполнителей, методы и технология для управления качеством и корректировками программ.

В жизненном цикле сложных комплексов программ для обеспечения их высокого качества целесообразно выделять специалистов, ответственных за **соблюдение промышленной технологии** их производства и совершенствования, за измерение и контроль качества комплексов в целом и их компонентов. Для систематической, координированной борьбы с угрозами качеству необходимо научить специалистов анализу и оцениванию конкретных факторов, влияющих на характеристики качества функционирования программных продуктов со стороны реально существующих и потенциально возможных дефектов в программах. Для обеспечения качества в конкретных проектах должны быть организованы и стимулированы разработка, освоение и применение современных методов, автоматизированных технологий и инструментальных средств производства, обеспечивающих **предупреждение или исключение большинства видов дефектов и ошибок** при производстве и модификации программных комплексов и их компонентов.

Все этапы производства и сопровождения необходимо поддерживать *методами и средствами* для измерения качества и определения реальных характеристик программ на любых этапах их жизненного цикла, для выявления и устранения дефектов. Наличие достаточно полных эталонов на основе совокупности требований спецификаций заказчика и поэтапная их декомпозиция – необходимая база для измерения реального качества программ. Ограниченность ресурсов при производстве приводит к целесообразности тщательного *планирования, упорядочения и применения экономических и эффективных методов автоматизации жизненного цикла* с целью достижения требуемого качества и достоверного его определения. Для каждого сложного продукта, выполняющего ответственные функции, рекомендуется разрабатывать и применять комплексную систему качества, специальные планы и Программу, методологию и инструментальные средства, обеспечивающие требуемое качество, надежность и безопасность функционирования комплексов программ. Последовательная детализация рекомендаций базовых стандартов должна доводиться до формирования должностных инструкций специалистам, образуя в совокупности иерархический комплекс документов – *систему качества производства программных продуктов на предприятии*.

Для обеспечения качества программного продукта необходима верификация и трассирование требований к функциям и компонентам комплекса, а также обеспечение баланса требований к ним. *Цели верификации* состоят в том, чтобы обнаружить и зарегистрировать ошибки, которые могли быть внесены в комплекс программ и компоненты во время его производства, и проверить что [1, 2, 15, 26]:

- системные требования к продукту, предназначенные для программной реализации, были должным образом переработаны в требования верхнего уровня к комплексу программ, которые удовлетворяют этим системным требованиям;
- требования верхнего уровня переработаны в архитектуру комплекса программ и требования к компонентам нижнего уровня, которые удовлетворяют требованиям верхнего уровня;
- архитектура комплекса программ и требования к компонентам нижнего уровня должным образом преобразованы в исходный код, удовлетворяющий им;

- исполняемый объектный код удовлетворяет требованиям к компонентам и к программному продукту;
- инструментальные средства, используемые для выполнения этих работ, являются технически корректными и полными для заданного уровня компонентов комплекса программ.

Результаты верификации комплекса программ должны быть достигнуты посредством выполнения комбинации из просмотров, анализов, разработки тестовых сценариев и процедур и последующего выполнения этих тестовых процедур. Просмотры и анализы должны обеспечивать оценку точности, полноты и верифицируемости требований, архитектуры комплекса программ, компонентов и исходного кода. Разработка сценариев верификации должна обеспечивать оценку внутренней непротиворечивости и полноты состава требований к компонентам, а также демонстрацию соответствия продукта требованиям.

Для обеспечения высокого качества программного продукта параллельно с **верификацией требований** и разработкой корректировок, следует разрабатывать и верифицировать **спецификации и сценарии тестов**, отражающие методы и конкретные процедуры проверки реализации этих требований. Тестовые спецификации могут использоваться для проверки согласованности, внутренней непротиворечивости и полноты реализации требований без исполнения программ. Для каждого требования к функциям комплекса программ, его архитектуре, функциональным компонентам должна быть разработана спецификация требований к тестам, обеспечивающая проверку корректности, адекватности и возможности в последующем реализовать тестирование компонента на соответствие этому требованию. Такая взаимная проверка функций компонентов, отраженных в требованиях и в спецификациях тестов, обеспечивает повышение их качества, сокращение дефектов, ошибок, неоднозначностей и противоречий.

Независимая разработка спецификаций тестов на основе спецификаций требований, создает базу для обнаружения, какие требования не тестировались или принципиально не могут быть проверены тестированием. Таким образом, **верификация спецификаций требований тестов** к функциям и характеристикам программных компонентов и комплекса могут использоваться с **двумя целями**:

- для проверки программ и интерфейсов взаимодействия программных компонентов разных уровней в комплексе программ;
- для создания требований к скоординированному комплексу тестов для проверки совокупности компонентов, обеспечивающих взаимную проверку реализации спецификаций требований комплексом программ.

Производственные процессы тестирования сложных комплексов программ

Тестирование комплексов программ может иметь *две цели*, которые одинаково важны для качества программного продукта. Первая цель – тестирование для идентификации дефектов подразумевает успешность процедуры тестирования, если *дефект найден и устранен*. Это отличается от подхода в тестировании (вторая цель), когда тесты исполняются для демонстрации того, что программный комплекс удовлетворяет предъявляемым требованиям и, соответственно, тест считается успешным, если *не найдено дефектов*. Тестирование программ может использоваться для демонстрации наличия дефектов, но никогда не гарантирует их отсутствие. Основная причина этого в том, что *полное, всеобъемлющее тестирование недостижимо* для реального сложного программного продукта [1, 11, 44]. Когда пытаются оценивать методы тестирования, надо четко определять, что подразумевается под их эффективностью, желательно, в количественных величинах.

При управлении производством необходимо выработать рекомендации по практическому применению методов тестирования и системы качества. Обязанностью администраторов программного проекта является *управление связями между производственными процессами и тестированием*:

- полный план тестирования должен быть определен в начале жизненного цикла программного комплекса;
- должны быть установлены правила регистрации, хранения, модернизации и сопровождения комплекса программ, тестовых данных и среды тестирования;
- необходимо учитывать, что следствием любого нарушения графика работ, предшествующих тестированию без соответствующей корректировки условий и даты поставки, может быть недостаточно

полное тестирование продукта;

- необходим интегрированный план сборки системы и компонентов программного комплекса в соответствии со стратегией выпуска версий системы и/или программного продукта;

- должно быть представлено подтверждение функциональных возможностей и качества программного продукта, показывающее их соответствие текущим требованиям заказчика;

- для обеспечения экономически эффективного процесса управления производством следует учитывать ресурсы необходимые для генерации тестов, используемые в процессе тестирования, и их результатов.

Основные особенности *тестирования сложных программных комплексов на соответствие требованиям заказчика* состоят в следующем:

- в результате тестирования программные продукты должны соответствовать целостному комплексу требований к функциям, характеристикам, архитектуре и качеству, утвержденному заказчиком и согласованному с разработчиком;

- комплексы применяемых тестов должны адекватно отражать и покрывать набор требований заказчика к программному продукту;

- разработка требований и тестирование комплексов программ должно осуществляться в условиях регулярного планирования и управления, а также ограниченных ресурсов, выделяемых заказчиком;

- для тестирования комплексов программ должны применяться сценарии и процедуры детерминированных тестов и генераторы динамических, стохастических тестов на основе аттестованных, сложных моделей внешней среды и систем реального времени;

- процессы и продукты формирования требований и реализации тестирования должны быть охвачены глубоким, детальным документированием, архивированием и конфигурационным управлением исходными, промежуточными и отчетными документами.

Программные продукты все чаще *встраиваются* в различные сложные системы реального времени. Работа над такими продуктами требует от программистов и тестировщиков, освоения *общих проблем проектирования, производства и применения систем* определенного назначения и сферы использования. Программному специалисту необходимо участвовать в разработке требований для всей сис-

темы, а также освоить прикладную область применения создаваемого комплекса программ до начала тестирования функций, их характеристик и тестов, требованиям которых должен отвечать программный продукт [2, 10, 26]. В жизненном цикле любых крупных систем и комплексов программ, определяющими их назначение, создание, свойства и качество являются **четыре ключевых объекта**:

- **требования** заказчика и/или пользователей к системе, которые определяют **цели создания и назначение программного продукта, его функции, качество и область применения**, необходимые пользователям системы и иным заинтересованным лицам, в заданной эксплуатационной среде;

- **тесты** – **эталон и вторая адекватная форма описания содержания, функционирования и качества программного продукта** для проверки полноты реализации требований, а также для верификации соответствия исходным требованиям к системе, комплексу и компонентам программ;

- **эксплуатационная документация**, которая обеспечивает применение программного продукта пользователями в соответствии с требованиями к его функциям и характеристикам, и является **третьем адекватным эталоном** корректности их реализации;

- **программный продукт и система, результат реализации требований**: функциональные характеристики системы, ее поведение и предусмотренные свойства, реализованные в системе и комплексе программ для обеспечения корректного применения; критические параметры, характеристики качества и эффективности; **экономические характеристик производства программного продукта** и эксплуатационные риски.

В **требованиях к программному продукту (первый эталон)** должно быть зафиксировано соглашение между заказчиком и выполняющими проект специалистами, отражающее потребности заказчика и пользователей в таком виде, чтобы разработчик мог построить удовлетворяющий их комплекс программ и его компоненты [9, 13, 15]. Требования являются **исходным эталоном** при разработке любой системы или программы, должны быть достаточно конкретными, чтобы можно было при тестировании и испытаниях установить, когда они удовлетворены. Вся разработка проекта должна вытекать из требований, а все спецификации программного комплекса и компонентов найти отражение в процессах, функциях и качестве продуктов

производства. Эти данные должны устанавливать состав, содержание и значения результатов исполнения программ, которые следует получать системе или пользователям при определенных условиях и исходных данных.

При разработке требований к программным продуктам необходимо учитывать функции, основные свойства и общие требования к проектам систем. Формирование назначения, функций и технического задания на проект системы должно включать основные требования к комплексу программ и особенности требований к нему заинтересованных лиц. Общие требования к качеству функционирования программных продуктов реального времени обязательно включают: требования к надежности и функциональной безопасности продуктов; требования к производительности и эффективности использования ресурсов ЭВМ программными продуктами в реальном времени, а также к допустимым рискам при их применении. Архитектурные требования к комплексам программ должны предусматривать возможность управления изменениями требований, организацию изменений и сопровождение требований, обеспечение повторного использования программных компонентов и комплексов.

В требованиях должно устанавливаться *ожидаемое поведение системы и программного продукта*, выраженное, желательно, количественными характеристиками, а также границы их допустимых значений. Требования могут зависеть от абстрактных представлений о подходящих характеристиках системы и включать методы и виды моделирования функционирования и среды, для достаточно полного описания заданных требований. Требуется удостоверить, что требования являются, с одной стороны, *необходимыми и достаточными* для удовлетворения заказчика, а с другой – необходимыми и достаточными входными данными для других процессов, в частности для проектирования функций, качества и архитектуры комплекса программ.

При формировании требований следует учитывать, что *источниками ошибок* в комплексах программ являются специалисты – конкретные люди с их индивидуальными особенностями, квалификацией, талантом и опытом (таблица 7.1).

Таблица 7.1

Специалисты – источники и типы первичных дефектов и ошибок

Специалисты – источники дефектов и ошибок	Типы первичных дефектов и ошибок программных комплексов и документации
Заказчик проекта программного продукта	Дефекты организации проекта и исходных требований заказчика на программный продукт
Менеджер проекта программного продукта	Дефекты, обусловленные реальной сложностью программного продукта
Менеджер-архитектор комплекса программ	Ошибки организации проектирования и архитектуры комплекса программ
Проблемно-ориентированные аналитики и системные архитекторы	Системные и алгоритмические дефекты и ошибки комплекса программ
Спецификаторы компонентов комплекса программ	Алгоритмические ошибки компонентов и документов комплекса программ
Системные интеграторы комплекса программ	Системные ошибки и дефекты реализации версий программного продукта и документации
Тестировщики требований и реализации компонентов и комплекса программ	Пропущенные системные, программные и алгоритмические ошибки компонентов, комплекса программ и документации
Управляющие сопровождением и конфигурацией версий программного продукта	Ошибки проектирования и реализации версий программного продукта
Документаторы программного продукта	Дефекты и ошибки технологических и эксплуатационных документов программного продукта

При этом можно выделить некоторые предсказуемые дефекты требований, модификаций, расширения и совершенствования комплекса, и необходимые изменения, обусловленные выявлением случайных, непредсказуемых дефектов и ошибок [13, 40]. Вследствие этого, плотность потоков и размеры необходимых **корректировок в требованиях** к комплексу и компонентам программ могут различаться в десяток раз. Однако, в сложных комплексах программ статистика и распределение типов ошибок и выполняемых изменений для коллективов разных специалистов нивелируются и проявляются достаточно общие закономерности, которые могут использоваться как ориентиры при их выявлении. Каждому типу необходимых корректировок соответствует более или менее определенная категория специалистов, являющихся источником дефектов данного типа. Такую корреляцию целесообразно учитывать как **общую качественную тенденцию** при анализе и поиске их причин. Этому могут помогать оценки типовых дефектов и корректировок, путем их накопления и обобщения по опыту разработки определенных классов комплексов программ в конкретных предприятиях.

Одной из основных причин ошибок в комплексах программ являются **организационные дефекты** при определении и использовании требований к качеству программного продукта, которые отличаются от остальных типов и условно могут быть выделены как самостоятельные. Ошибки и дефекты данного типа появляются из-за недостаточного понимания коллективом специалистов целей и функций комплекса программ, а также вследствие отсутствия четкой его организации и поэтапного контроля требований качества продуктов. Это порождается **пренебрежением руководителей к организации всего технологического процесса** формализации требований сложных программных продуктов и приводит к серьезной недооценке их дефектов, а также трудоемкости и сложности их выявления и устранения. Для сокращения этого типа массовых ошибок, активную роль должны играть лидеры – менеджеры и аналитики-системотехники, способные вести контроль и конфигурационное управление требованиями, качеством, изменениями и развитием версий и компонентов комплексов программ.

Реализация этих целей верификации и тестирования может производиться разными методами и специалистами – программистами, интеграторами и тестировщиками, что позволяет использовать ре-

зультаты их деятельности для сравнения одних и тех же описаний программ, представленных на языках программирования и на языках тестов. Особенности описаний и реализации программ, а также **мышления их разработчиков (программистов)** – на основе требований, функций, характеристик структуры и исполнения программ, существенно **отличаются от представлений и методов описаний тех же функций комплекса программ тестировщиками** – создателями сценариев и эталонов требуемых результатов тестирования. Они акцентируют свою деятельность на конкретных процедурах проверки функционирования, возможных результатах и взаимодействии компонентов комплекса программ. Это позволяет выявлять вторичные дефекты, и повышать качество путем **сопоставления двух методов** и результатов описания **одних и тех же** функций и характеристик программ, за счет того, что мала вероятность одинаковых ошибок в сценариях и реализациях тестов, и в описаниях требований к функциям и характеристикам программ.

Для обеспечения эффективности затрат ресурсов, тестирование должно быть интегрировано как можно раньше с основными процессами проектирования, разработки и сопровождения в жизненном цикле программных комплексов. Оно должно проводиться, начиная от анализа общих требований, заданных в техническом задании и/или в спецификации на всю систему, до детальных испытаний программных продуктов и их взаимодействия с внешней средой. По существу, требования к комплексу программ и тесты для проверки их адекватности и полноты, **отражают один и тот же объект**, но в разной форме. Поэтому сложность представительного описания комплекса тестов **соизмерима** со сложностью описания полной совокупности требований к функциям, характеристикам и качеству соответствующего комплекса программ. Вследствие этого трудоемкость и другие ресурсы для разработки **адекватных тестов должны соответствовать ресурсам и трудоемкости** при создании и реализации корректных требований, и определяющих их полностью тексты программ. Однако на практике обычно это не учитывается, и выделяемые на тестирование ресурсы оказываются значительно меньше и недостаточными для полноценного тестирования требований к крупным комплексам программ, что определяет их не полное соответствие требованиям и недостаточное качество.

Создание плана тестирования – итеративный процесс, требующий обратной связи с различными участниками проекта и согласованности с определенными в нем процессами, стратегиями тестирования и сроками выполнения работ. С этим планом должен быть коррелирован и предшествовать план программирования и подготовки к тестированию модулей и компонентов сложных функциональных задач. Тест-менеджер должен **утвердить стратегию программирования компонентов, их тестирования** и тестовые процедуры, которые должны быть подробно описаны в плане тестирования, и определять какие компоненты и модули, сценарии и тесты когда будут выполняться. Кроме того, предполагается, что руководитель проекта согласен с тем, что план тестирования компонентов и связанные с ним тестовые сценарии достаточно проверяют покрытие тестами требований, эталонов или сценариев использования комплекса программ и системы. Подробное изучение системных требований или сценариев применения системы вместе с тщательным определением параметров плана тестирования и требований к тестам необходимы для эффективного тестирования программного комплекса.

Тестирование всегда предполагает компромисс между ограниченными экономическими ресурсами и заданными сроками, с одной стороны, и практически неограниченными требованиями к качеству результатов тестирования и программного продукта, с другой. Оценка стоимости и затрат, а также другие измерения процессов, связанных с оценкой ресурсов, необходимых для тестирования, как и оценка **экономической эффективности тестирования** на разных этапах и уровнях, основывается на точке зрения и практиках менеджеров проектов и используется для оценки и улучшения процесса тестирования. Разные техники, концепции и модели тестирования требуют разных затрат по времени и необходимым ресурсам (до 20 – 30% общего бюджета проекта). Учет соответствия между стоимостью или затратами, необходимыми для того или иного метода, и эффективностью тестирования является обязательной частью эффективного управления проектами сложных программных комплексов.

Характеристики качества встроенных программных продуктов реального времени зависят не только от их внутренних свойств, но и **от свойств внешней среды – сценариев и генераторов динамических тестов**, в которой они испытываются и применяются. Для сокращения неопределенностей и прямых ошибок при оценивании ка-

чества комплексов программ необходимо до начала разработки определять основные параметры внешней среды, при которых должен функционировать комплекс программ с требуемыми характеристиками. Для этого заказчик и разработчики совместно должны структурировать, описать и **согласовать модель внешней среды** и ее параметры в среднем, типовом режиме применения программного продукта, а также в наиболее вероятных и критических режимах генерации тестов, в которых должны обеспечиваться требуемые характеристики качества функционирования. Увеличение необходимости проверять широкий набор требований и характеристик сложных программных комплексов приводит к необходимости **динамического генерирования тестов** посредством моделей внешней среды, охватывающих по возможности весь спектр данных, воздействующих на тестируемый объект. Подготовку тестов для испытаний крупных комплексов программ целесообразно осуществлять с применением автоматизированной генерации динамических тестов внешней среды в реальном времени, а также средств обработки результатов динамического тестирования.

Производственные процессы документирования сложных комплексов программ

Документация является органической, составной **частью экономики производства (10 – 15% затрат) программного продукта** и требуются значительные ресурсы для ее создания и применения. Тексты и объектный код комплексов программ **могут стать программным продуктом** только в совокупности с документами, полностью соответствующими их содержанию и достаточными для его освоения, применения и изменения. Для этого документы должны быть корректными, строго адекватными текстам программ и содержанию баз данных – систематически, структурировано и понятно изложены, для возможности их успешного освоения и использования достаточно квалифицированными специалистами различных рангов и назначения. Качество и полнота отображения в документах процессов и продуктов в жизненном цикле программных комплексов, должны полностью определять достоверность информации для взаимодействия заказчиков, пользователей и разработчиков, а тем самым, корректность функций, **качество программных продуктов** и соответствующих

систем. Посредством документов (электронных или бумажных) специалисты взаимодействуют с программными продуктами и данными в реализующих их вычислительных машинах, а также между собой.

Существует большая разница между тем, чтобы просто написать и запрограммировать некоторую функцию для индивидуального использования ее разработчиком, и тем, чтобы изготовить ее как качественный компонент или сложный **программный продукт, отчуждаемый от разработчиков и поставляемый заказчику и/или пользователям**. Создание программного продукта требует значительных интеллектуальных и организационных усилий, а его документация – это сложный компонент, подверженный изменениям, которые могут вноситься многими специалистами. Управление документацией должно непрерывно поддерживать ее полноту, корректность и согласованность с содержанием программного продукта, что требует значительных затрат. Необходимо обеспечивать возможность достоверного, формально точного общения всех участников производства между собой, с создаваемым продуктом и с документами для гарантии поступательного развития и совершенствования комплекса программ.

Адекватность документации требованиям, состоянию текстов и объектных кодов программ должна инспектироваться и удостоверяться (подписываться) ответственными руководителями и заказчиками проекта. **Ошибки и дефекты документов не менее опасны для применения и изменения программного продукта**, чем ошибки в структуре, интерфейсах, файлах текстов программ и в содержании данных. Поэтому к разработке, полноте, корректности и качеству документации необходимо столь же тщательное отношение, как к производству и изменениям текстов программ и данных [5, 21].

Экономика реализации корректных документов в значительной степени определяет достигаемое качество сложных программных продуктов, трудоемкость и длительность их создания. Для этого должна формироваться и использоваться регламентированная стратегия, стандарты, распределение ресурсов и планы создания, изменения и применения документов на программы и данные сложных систем. В общем случае должны быть **выделены руководители и коллектив специалистов**, которые способны планировать, описывать, утверждать, выпускать и сопровождать комплекты адекватных документов. Они должны стимулировать разработчиков осуществлять непрерывное, полноценное документирование процессов и результатов

своей деятельности, а также контролировать полноту и качество исходных, результирующих и отчетных документов. Официальная, описанная и утвержденная стратегия документирования должна устанавливать дисциплину, необходимую для эффективного создания высококачественных документов на продукты и процессы в жизненном цикле комплексов программ.

Совокупные экономические затраты на документирование сложных программных продуктов **могут достигать 15%** от общего бюджета проекта, и необходимого числа (многие десятки) специалистов в жизненном цикле комплекса программ. В более простых случаях, организация работ может быть упрощена, затраты на документирование снижаются приблизительно **до 10%**. Обычно целесообразно выделять специалистов, непосредственно ответственных за создание, адекватность и контроль полноценного комплекта документов на программный продукт. Состав и общий объем документов широко варьируется в зависимости от класса и характеристик объекта разработки, а также в зависимости от используемой технологии. Наиболее сложному случаю разработки критических программных продуктов реального времени высокого качества соответствует самая широкая номенклатура документов. Такой перечень документов может быть использован как базовый для формирования на его основе состава и шаблонов документов в остальных более простых проектах [21].

Технологическая документация должна отражать производственные процессы жизненного цикла комплексов программ и данных, требования к ним, включать подробные технические описания, подготавливаемые для специалистов, ведущих проектирование, разработку и сопровождение комплексов программ, обеспечивающие возможность их отчуждения, детального освоения, развития и корректировки программ и данных. Стандарты и нормативные документы должны регламентировать структуру, состав производственных этапов, работ и подготовки документов. Они должны: формализовать выполнение и документирование конкретных работ при проектировании, разработке и сопровождении комплексов программ; обеспечивать адаптацию документов к характеристикам среды разработки, внешней и операционной системы; регламентировать процессы обеспечения качества комплексов и его компонентов, методы и средства их достижения, реальные **значения достигнутых показателей ка-**

чества. Для контроля возможных изменений целесообразно предусматривать и согласовывать с заказчиком специальный документ, регламентирующий правила применения и корректировки их номенклатуры, а также состава и содержания документации поддерживающей производство комплексов программ.

Эксплуатационная документация должна обеспечивать отчуждаемость программного продукта от их первичных поставщиков – разработчиков, возможность освоения и эффективного применения комплексов программ достаточно квалифицированными специалистами – пользователями для решения конкретных функциональных задач систем. Эксплуатационные документы должны исключать возможность некорректного использования программных продуктов за пределами условий эксплуатации, при которых документами **гарантируются требуемые показатели качества** их функционирования. Основная их задача состоит в фиксировании, полноценном использовании и обобщении результатов функционирования программных продуктов и системы.

Базой эффективного управления документированием должен быть **План**, в котором задачи исполнителей частных работ согласованы с выделяемыми для них **экономическими ресурсами**, а также между собой по результатам и срокам их достижения. Реализация плана зависит от результатов выполнения производственных работ и документов, и может требовать оперативной корректировки плана. **Контроль адекватности и качества документов является органической функцией управления проектом** и должен иметь средства регулирования поведения отдельных специалистов и коллектива разработчиков в целом. В плане управления документированием и обеспечением качества внимание специалистов должно акцентироваться на анализе достигнутых результатов разработки, методах и средствах достижения заданных заказчиком характеристик комплекса программ. При планировании и разработке комплекс документации должен проверяться и аттестовываться на выполнимость и полноту, в условиях ограниченных ресурсов, а также на корректность, адекватность и непротиворечивость отдельных документов.

Различия в организационных службах предприятий, процедурах, методах и стратегиях создания программных продуктов, масштабах и сложности проектов, требованиях систем и методах их разработки влияют на **способы разработки, применения и сопровождения до-**

кументов. В ряде предприятий могут использоваться собственные нормативные шаблоны документов на производственные процессы и продукты, адаптированные к конкретным условиям разработки и характеристикам создаваемых продуктов. В них выделяются состав и шаблоны документов, наиболее важные для определенного проекта и качества создаваемого продукта, а также для конкретных заказчиков и пользователей.

Сложность, количество и полнота содержания комплекса документов в первую очередь зависят от **масштаба – размера проекта программного продукта**, что целесообразно оценивать в начале его проектирования. Для решения этой задачи необходимо детально исследовать требуемые ресурсы современных процессов создания, документирования и использования программ различных классов и назначения – встроенных, коммерческих, административных, учебных. Состав и шаблоны документов целесообразно базировать на серии международных стандартов (см. Приложение). Менеджеры проекта программного продукта, ответственны за **выбор экономически эффективной модели комплекса документации** для конкретного проекта и за адаптацию производственных процессов и шаблонов документов применительно к выбранной модели документооборота проекта программного средства.

Особое внимание следует уделять совершенствованию и детализации **документов, обеспечивающих высокое качество создаваемых программных продуктов**, а также возможности их эффективного итерационного развития длительное время в многочисленных версиях. Соответственно должны изменяться документы, отражающие состояние производственных процессов и компонентов программных проектов. Для этого организация процессов документирования должна обеспечивать возможность гибкого и точного изменения документов – сопровождения и конфигурационного управления версиями и редакциями каждого документа.

Для хранения, тиражирования и распространения документов, сложных комплексов программ высокого качества, целесообразно выделять группу специалистов, ответственных за **контроль, обеспечение и гарантированное сохранение документации**. Для критических, важных систем, документация на программы и данные должна храниться и дублироваться на различных типах носителей и эпизодически выводиться на бумажные носители. При определении схемы

обеспечения сохранности документации разного содержания, следует учитывать ее важность, трудоемкость хранения и возможность аварийного восстановления. Кроме того, должна быть организована служба контроля, ответственная за соблюдение требований стандартов, нормативных и руководящих документов при подготовке документации всеми специалистами, участвующими в сложном проекте. Для обеспечения достоверных данных об объектах и процессах управления документами программных продуктов, необходима автоматизированная **база данных – информационная система обеспечения и хранения документов проектов.**

Глава 8

МОДЕЛИ ПРОГНОЗИРОВАНИЯ ЭКОНОМИЧЕСКИХ ХАРАКТЕРИСТИК ПРОИЗВОДСТВА ПРОГРАММНЫХ ПРОДУКТОВ

Методы и достоверность прогнозирования экономических характеристик производства программных продуктов

Приступая к разработке комплекса программ, как в любой производственной деятельности, необходимо провести реалистичную оценку исходных данных: возможного *масштаба проекта* – сложности поставленных целей, функций, ресурсов проекта и выделенного времени. Задача определения и управления масштабом состоит в задании базовых требований, которые включают разбитое на компоненты ограниченное число функций, компонентов и требований, намеченных для реализации в конкретном программном продукте. *Базовый уровень масштаба комплекса программ должен обеспечивать* приемлемый для заказчика *минимум функций* и требований к программному продукту и разумную вероятность успеха проекта с точки зрения производственных возможностей коллектива разработчиков.

При оценивании масштаба следует определить приоритеты функций для установления состава работ, согласованного между заказчиком и разработчиком, которые *обязательно* должны быть выполнены. Сокращение масштаба проекта до размеров, адекватных выделенному времени и ресурсам, может привести к конфликтам заказчиков и разработчиков. Для уменьшения вероятности таких конфликтов целесообразно активно привлекать заказчиков к управлению их требованиями и масштабом программного продукта, чтобы обеспечить как качество, так и своевременность его производства. При этом следует учитывать, что:

- именно *заказчики несут финансовую ответственность* за выполнение обязательств перед внешними клиентами и пользователями;

- комплекс программ, его важнейшие функции и удовлетворяемые требования принадлежат заказчику, а *не коллективу разработчиков или поставщику*.

Прогнозирование масштаба – размера, оценка и составление графика производства сложным образом переплетаются в процессе планирования экономики проекта. Довольно сложно создать реальный график (учитывающий затраты, обязанности исполнителей, их зависимости, перекрытие действий, а также дату поставки продукта) без информации об объеме трудозатрат, требуемых для выполнения каждой задачи (например, определения загрузки и ожидаемых результатов сотрудников за месяц). Достаточно трудно оценить объем трудозатрат, необходимых для выполнения задач и функций, без достоверной информации относительно их *размера*. Таким образом, измерение размера (сложности) предшествует оценке экономических характеристик, а эта оценка, в свою очередь, предшествует составлению графика производства. Недостаточно достоверные оценки функций, размера, экономических характеристик и плана работ могут вызывать негативные проблемы взаимодействия разработчиков с заказчиком и увеличивать степень риска проекта комплекса программ.

Какая бы технология не использовалась, оценка размера будущего продукта является весьма важной задачей, поскольку она является основой *реальных экономических характеристик производства*. Нереальные ожидания участников проекта, основанные на неточных оценках, представляют собой одну из частых причин провала проектов. Зачастую причина кроется не в недостаточной производительности команды специалистов проекта, а в неточном прогнозировании сложности и размера продукта. Точное прогнозирование экономических характеристик обеспечивает улучшенный контроль над проектом и является жизненно важным в деле проектного менеджмента.

Привлечение заказчика помогает менее болезненно решать проблемы управления масштабом проекта и реализуемыми функциями с учетом ограничений ресурсов. Любые дополнительные функции за пределами базовых требований, которые реализует производитель, будут восприняты заказчиком как превзошедшие ожидания. В зависимости от этапа производства сложного комплекса программ и достоверности исходных данных о характеристиках и особенностях проекта

целесообразно выбирать и применять методики и сценарии прогнозирования экономических характеристик.

При прогнозировании экономических характеристик производства программного продукта менеджерам проекта целесообразно учитывать:

- цели оценивания экономических характеристик должны быть согласованы с потребностями в информации для принятия решений на соответствующем этапе проектирования и производства комплекса программ;

- достоверности оценок должны быть сбалансированы для различных компонентов комплекса программ, величина уровня неопределенности для каждого учитываемого фактора и компонента должна быть примерно одинаковой, если в процессе принятия решения все они имеют одинаковый вес;

- следует возвращаться к предшествующим оценкам экономических характеристик и изменять их, когда это необходимо для ответственных бюджетных решений, принимаемых на последовательных этапах и влияющих на реализацию проекта.

С самого начала работы над проектом важно вести постоянный учет данных о его *действительной трудоемкости, стоимости и затратах*, и сравнивать эти данные *с прогнозами* экономических характеристик проекта. Несовершенство исходных данных (оценки сложности функций, размера, влияния факторов) определяет для руководителя проекта необходимость пересматривать их оценки, учитывая новую информацию, чтобы обеспечить более реальную основу для дальнейшего управления проектом. Вследствие несовершенства методов оценивания будущих функций комплекса программ, следует оперативно сравнивать оценки экономических характеристик с действительными значениями, получающимися в процессе производства, и использовать эти результаты для улучшения методов и результатов прогнозирования, идентифицировать эти изменения и выполнять реалистичное обновление прогнозов.

Каждая оценка экономических характеристик должна сопровождаться, *указанием степени ее неопределенности*. По мере разработки проекта их необходимо пересматривать и изменять, когда это становится полезным. Экономические решения, принимаемые на ранних этапах должны влиять преимущественно только на сле-

дующий этап прогнозов. Для этого в данной главе последовательно рассмотрены и рекомендуются **три базовых метода** (рис. 8.1):

- первичный экономический прогноз при **подготовке концепции** и технического задания на новый комплекс программ на основе экспертных данных о его размере, средней производительности труда специалистов или стоимости разработки одной строки текста программ – прототипов;
- прогнозирование основных экономических характеристик **при предварительном проектировании** на базе расчетных значений трудоемкости и длительности разработки комплекса программ по данным упрощенных моделей КОМОС или ПРОМЕТЕЙ [3,16, 46] с учетом влияния минимума дополнительных факторов;
- определение возможных экономических характеристик проекта с учетом **доступных оценок множества факторов** для календарного планирования производства сложного программного продукта с использованием систем прогнозирования ПРОМЕТЕЙ, СОСОМО или СОСОМО II [38].

Неопределенность прогнозов уменьшается, по мере детализации и углубления содержания функций проекта, как только фиксируются конкретные принципы функционирования и концепция комплекса программ. На этапе **подготовки концепции** оценка достоверности размера продукта по прототипам может составлять приблизительно 40% (см. рис. 8.1). Это вполне объяснимо, поскольку еще не уточнены структура и многие детали проекта. Эти задачи могут быть разрешены во время разработки структуры и спецификаций требований к комплексу программ, и тогда можно оценить будущий его размер с точностью до 15 – 20%.

После завершения разработки и подтверждения проектных спецификаций при **предварительном проектировании** комплекса программ может быть определена структура внутренних данных и функции программных компонентов. На этом этапе, достоверность оценки размера и трудоемкости проекта может составить около 10%, что приводит к целесообразности при прогнозировании применять более точные модели и учитывать несколько факторов производства. Неопределенности оценок характеристик проекта, могут быть обусловлены: особенностями конкретных алгоритмов, управлением их функционированием, обработкой ошибок, инициализацией и завершением сеансов работы.

Модели прогнозирования экономических характеристик производства программных продуктов включают:

- на этапе разработки целей и концепции проекта:
 - экспертные оценки экономических характеристик по прототипам – достоверность оценивания размера продукта 40 – 50%;
- на этапе предварительного проектирования:
 - оценивание экономических характеристик в базовой модели КОМОСТ – учитывается один фактор и достоверность оценивания размера продукта – 20 – 30%;
 - оценивание экономических характеристик в детализированной модели КОМОСТ – учитывается три фактора и достоверность оценивания размера продукта – 10 – 20%;
- на этапе детального проектирования:
 - оценивание экономических характеристик в упрощенной, предварительной модели СОСОМО или ПРОМЕТЕЙ – учитывается семь факторов и достоверность оценивания размера продукта – 5 – 10%;
 - оценивание экономических характеристик в детальной модели СОСОМО II – учитывается двадцать два фактора и достоверность оценивания размера продукта 3 – 5%.

Рис. 8.1

Эти уточнения размеров комплекса и функций компонентов программ могут быть решены к концу **детального проектирования**, однако при этом сохраняется неопределенность оценки размера комплекса программ и его трудоемкости порядка 5 – 10%, связанная с тем, насколько хорошо программисты понимают спецификации и алгоритмы, в соответствии с которыми они должны кодировать программу. Поэтому целесообразно последовательно уточнять исходные данные и проводить прогнозирование экономических характеристик для различных компонентов программного продукта и этапов проектирования.

В общем случае желательно достигать **сбалансированного** выбора целей прогнозирования экономических характеристик комплекса программ, который бы давал примерно одинаковую величину влияния уровня неопределенности для всех исходных данных, учитываемых факторов и компонентов продукта. Вследствие этого

изложение анализа и обоснования прогнозирования экономических характеристик ниже разделено на три крупные части. В данном разделе рассматриваются первичные, базовые оценки трудоемкости, длительности и числа специалистов в зависимости от размера комплекса программ, повторного использования компонентов и небольшого числа факторов. Далее уточняются экономические характеристики, и оценивается влияние на них ряда дополнительных факторов: особенностей и качества объектов и среды разработки, квалификации и организации специалистов, технологической, инструментальной и аппаратурной оснащенности проекта с использованием модели СОСОМО II (см. рис. 8.1). По мере проектирования и производства комплекса программ, оценки экономических характеристик необходимо пересматривать и изменять, когда это становится выгодным или необходимым.

Экспертное прогнозирование экономических характеристик производства программных продуктов

В простейшем методе может быть реализован прогноз экономических характеристик производства программного продукта с учетом **экспертной оценки минимального числа факторов**. Эта методика может применяться, когда определены цели, масштаб и общие функции комплекса программ в концепции и первичных требованиях с достоверностью около 30 – 40%. Основная **цель такого прогноза экономических характеристик** – подготовка обоснованного решения о допустимости развития проекта в область системного анализа, разработки требований и предварительного проектирования.

Если оказывается, что прогнозируемые экономические показатели и требуемые ресурсы не могут быть обеспечены для продолжения проектирования и производства, то возможны **кардинальные решения**: либо изменение некоторых требований к характеристикам программного продукта и выделяемых ресурсов, либо прекращение проектирования данного комплекса программ. Учитывая полноту и достоверность доступных характеристик и требований к проекту должны быть определены **цели и возможная достоверность экономического прогнозирования затрат при продолжении проектирования и производства продукта**.

При первичном экономическом обосновании сложных комплексов программ наибольшее значение имеют **ключевые факторы**:

- сложность и размер – масштаб, подлежащих разработке полностью новых программных компонентов и всего комплекса (количество строк текста с указанием языка программирования);
- размер и относительная доля готовых программных компонентов, которые могут быть заимствованы из предшествовавших продуктов и повторно использованы в новом проекте комплекса программ (%);
- относительные затраты ресурсов на создание программного продукта: труда специалистов (человеко-месяцы), времени или бюджета на единицу размера (на строку текста программ) проектируемого комплекса программ;
- возможная исходная стоимость разработки одной строки текста в комплексе программ.

Эти факторы могут быть прогнозируемы квалифицированными экспертами на основе имеющегося у них опыта реализации предшествовавших подобных проектов, а также с использованием опубликованных данных (оптимистических и/или пессимистических оценок). При наличии необходимых данных важно оценить их достоверность и возможную точность (20 – 30%). Наименее точный из перечисленных факторов может полностью определять достоверность прогнозов экономических характеристик, поэтому желательно, чтобы значения точности экспертного оценивания перечисленных факторов были сбалансированы. При наличии перечисленных исходных данных и положительной оценке целесообразности экспертного прогноза экономики проекта может реализовываться **методика, состоящая из следующих основных шагов**:

- экспертная оценка размера – масштаба, числа строк (функциональных точек) предполагаемого текста разрабатываемых программ с учетом размера повторно используемых компонентов и характеристик возможного языка программирования;
- экспертная оценка возможной средней производительности труда в команде специалистов при разработке комплекса программ и/или стоимости разработки одной строки текста программ продукта;
- расчет возможной полной трудоемкости и длительности проектирования и производства программного продукта, а также среднего числа специалистов, необходимых для его реализации;
- обобщение основных экономических характеристик и полной стоимости программного продукта, анализ результатов и экономиче-

ское обоснование рентабельности продолжения проектирования и производства комплекса программ.

Достоверность прогнозов экономических характеристик зависит, прежде всего, от **точности экспертной оценки исходных данных**: размера – масштаба комплекса программ, и от достоверности экспертной оценки производительности труда специалистов или стоимости разработки одной строки текста программ. Экспертные прогнозы предполагают использование опыта прошлых разработок и их отличия от новых методов, предусмотренных в конкретных проектах, а также учет индивидуальных возможностей специалистов коллектива разработчиков и/или другие уникальные особенности проекта. Экспертные оценки зависят от компетенции и объективности экспертов, их оптимистичности, пессимистичности, знания существенных особенностей предполагаемого продукта. Оценки одного эксперта трудно проверять и контролировать на достоверность. Из-за индивидуальных особенностей экспертов предпочтительным является прогнозирование экономических характеристик несколькими экспертами и получение средних оценок на совещании группы экспертов для формирования единой оценки.

Экспертная оценка размера – сложности программного комплекса основная задача в этом методе (см. главу 4). Точность оценок должна быть сбалансирована с достоверностью экспертного определения или выбора характеристик других факторов, используемых при прогнозировании (удельной трудоемкости или стоимости строки текста программы). Эти методы оценивания основываются на опыте предшествующих разработок, которые не всегда могут обладать особенностями структурного программирования, использовать автоматизированные инструментальные средства, языков спецификаций или распределенную обработку данных прототипов.

Основными исходными данными для прогнозов являются **концепция проекта программного продукта и состав требований** к иерархическому набору их функций и характеристик, которые могут быть детализированы на предполагаемые функциональные компоненты (с учетом приоритетов) комплекса программ. В дальнейшем требования могут детализоваться, формируя упрощенный или более точный уровень абстракции архитектуры комплекса и взаимодействия компонентов. Наиболее глубокий уровень детализации (до модулей), как правило, редко формируется ко времени первоначальной

экспертной оценки размера программного продукта. Один из путей оценки размера комплекса программ, находящегося на этапе создания концепции проекта, заключается в сравнении его функциональных задач и свойств с уже существующими аналогами. Конечно, данный метод не является точным, поскольку при производстве компонентов прототипов могли использоваться языки программирования, относящиеся к разным областям применения приложений. Также могли применяться разнообразные алгоритмы, имеющие различный уровень сложности, совместно с не проверенными функциональными свойствами, а также с различной степенью адекватности моделирования внешней среды и оценки влияния общесистемных работ (интеграции компонентов, испытаний и документирования программного продукта).

Экспертный прогноз удельных затрат на строку текста программного продукта обычно относится к полному циклу производства сложных комплексов программ, начиная от создания концепции и требований до завершения испытаний и передачи программного продукта заказчику или пользователям. В составе участников проекта учитываются все категории специалистов, обеспечивающие реализацию программного продукта. Несмотря на появление новых методов, и инструментальных средств для разработки сложных комплексов программ, средняя производительность при их создании за последние годы оставалась почти неизменной и составляет около 3000 строк кода на одного разработчика продукта в год (порядка 250 строк на человеко-месяц) [3, 16]. Уменьшение времени, затрачиваемого на цикл производства крупного продукта, не может быть достигнуто за счет значительного повышения производительности труда только отдельных специалистов – программистов. Она практически не зависит от усовершенствований языков программирования, организационных усилий со стороны менеджеров, от наличия или отсутствия некоторых отдельных видов инструментария и автоматизации производства компонентов и модулей. Однако *значительно влияет доля повторно используемых компонентов* (см. главу 6). При достаточно высоком уровне технологии (см. главу 9, рис. 9.1, уровни 4 – 5 – СММІ) большое значение имеет возросший размер и сложность состава функциональных задач комплексов программ, а также значительное *повышение требуемого качества и ответственности* за создаваемые программные продукты.

Так как предприятия и разработчики сложных комплексов программ **не заинтересованы раскрывать реальную экономику** выполненных проектов и производства программных продуктов, (они эти данные склонны относить к коммерческой тайне), опубликованные экономические характеристики программных проектов носят отрывочный, не упорядоченный и не всегда достоверный характер. Кроме того, обычно не представляются детальные сведения об особенностях и требованиях к качеству объекта разработки, применявшейся технологии и инструментальных средствах, характеристиках коллектива специалистов и других факторах. Поэтому такие данные трудно обобщать и использовать для прогнозирования новых проектов.

Весьма общие **примеры** опубликованы в виде диапазонов производительности труда [3, 7, 31]: для относительно простых программных продуктов – 8 LOC на человеко-день, и для сложных продуктов – 4 LOC на человеко-день. Там же приведены диапазоны производительности труда при производстве комплексов программ на ассемблере – 60 – 500 LOC на человеко-месяц, и 50 – 300 LOC на человеко-месяц для языков высокого уровня. Более полные оценки производительности при разработке комплексов программ различного размера и классов на основе обобщения статистических данных множества проектов представлены в **примерах** применения базовой модели КОМОСТ [3]:

- для встроенных комплексов программ (СРВ) размером 30 тысяч строк для прогнозов рекомендуется использовать производительность около 140 строк на человеко-месяц, а для крупных программных продуктов размером 500 тысяч строк предлагается значение производительности около 80 строк на человеко-месяц;
- для программ административных систем (АС) размером 30 тысяч строк оценка производительности составляет около 220 строк на человеко-месяц, а для комплексов размером 500 тысяч строк – 160 строк на человеко-месяц.

Эти данные находятся в середине представленных выше диапазонов и их можно использовать при экспертной оценке полной трудоемкости производства соответствующих полностью **новых** программных продуктов. При применении готовых повторно используемых компонентов обобщенная производительность труда возрастает и зависит от доли таких компонентов в комплексе программ. Для таких ситуаций можно использовать **примеры** коэффициентов снижения

трудоемкости и длительности разработки программных продуктов за счет применения ПИК, изложенные в главе 6 и на соответствующих графиках (см. рис. 6.2 – 6.3).

Эти данные можно использовать для прогнозирования **полной стоимости программного продукта**. Однако при этом необходимы удельные данные стоимости труда одного человеко-месяца специалистов в **конкретном предприятии** с учетом всех накладных расходов, которые могут различаться в несколько раз. Такие сведения обычно являются коммерческой тайной, и при использовании данного метода для определенного проекта, их следует запрашивать у экономических служб конкретного предприятия. Опубликованы **ориентир**ы около 100\$ и более стоимость разработки одной строки текста программ реального времени высокого качества (1000\$ бортовые комплексы программ системы Шатл), а для административных систем – около 20 – 50\$.

Экспертное прогнозирование длительности производства сложных программных продуктов может базироваться на использовании рассчитанной ранее трудоемкости его разработки, от которой не линейно зависит длительность. Например, крупные новые проекты комплексов программ реального времени размером около 500 тысяч строк (без ПИК) требовали для реализации около 3,5 лет, а небольшие (30 тысяч строк) – около одного года [3]. Полная длительность производства программного продукта может быть структурирована на затраты времени по шести технологическим этапам с использованием экспериментальной таблицы 6.1.

Экспертная оценка необходимого числа специалистов может быть рассчитана путем деления полной трудоемкости производства программного продукта на длительность его реализации. Для **примера** крупного продукта реального времени, размером 500 тысяч строк, необходимое число специалистов достигало 160 человек, а для относительно небольшого проекта (30 тысяч строк) – в десять раз меньше (16 человек) [3]. Аналогично можно получить оценки необходимого числа специалистов на выделенных крупных этапах производства комплексов программ, что может быть полезно для **первичного формирования коллектива** и оценки возможности реализации им конкретного проекта.

Обобщенные экспертные оценки экономических характеристик проектов целесообразно представлять в виде таблиц (желательно) с

указанием достоверности оценок результатов расчетов следующих характеристик:

- полной трудоемкости производства программного продукта – C (человеко-месяцы с указанием учитываемых факторов);
- полной длительности производства программного продукта – T (месяцы);
- необходимого среднего числа специалистов – N (человек);
- средней производительности труда P специалистов (число строк на человеко-месяц).

На основе анализа результатов оценивания рассчитанных характеристик целесообразно выполнить *экономическое обоснование перспективы производства программного продукта* и определить (см. главу 2):

- следует ли продолжать работы над конкретным проектом программного продукта;
- полезно ли провести маркетинговые исследования для определения рентабельности полного выполнения проекта;
- корректно ли формализованы концепция и требования к программному продукту, и возможно ли их совершенствование;
- есть ли возможность применить готовые повторно используемые компоненты или весь проект целесообразно разрабатывать как полностью новый.

Простейшие модели прогнозирования экономических характеристик производства программных продуктов

На *этапе предварительного проектирования комплекса программ*, после утверждения требований и концепции проекта, повышается достоверность данных о функциях, спецификациях, компонентах и структуре разрабатываемого комплекса программ. Это позволяет, прежде всего, более точно оценить размер – масштаб комплекса и возможность использования готовых программных компонентов из других аналогичных проектов. Если достоверность оценки размера достигает 15 – 20%, то при определении экономических характеристик, целесообразно выделять и учитывать факторы, влияние которых на трудоемкость и стоимость достаточно велико, и составляет также около 20%. Таких факторов может быть около 3 – 5, и их число зависит от конкретных характеристик программного продукта

и среды его производства. На этом этапе, состав и номенклатура учитываемых факторов должны выбираться путем включения в анализ тех, которые могут достаточно сильно влиять на экономику конкретного проекта. Таким образом, методика и сценарии оценки экономических характеристик должны **калиброваться по шагам** в соответствии с уменьшением неопределенности влияния факторов и исходных данных о размере проекта (см. рис. 8.1).

Так как **величина и достоверность определения размера комплекса программ ключевой фактор** последующего экономического анализа, то целесообразно применять несколько доступных методов для его оценивания. Прежде всего, необходимо использовать, ревизовать и углубить данные экспертной оценки возможного размера продукта, полученные при использовании предыдущей методики, с учетом повышения достоверности характеристик проекта по результатам предварительного проектирования. Конкретизация функций, структуры комплекса программ и состава компонентов проекта позволяет более достоверно определить размеры компонентов комплекса программ и, суммируя их, оценить размер всего комплекса. Кроме того, на этом этапе повышается возможность выбора и использования аналогов данного проекта, так как становятся более определенными задачи, функции и компоненты разрабатываемого комплекса, для которого желательно найти аналоги с известными апробированными характеристиками. Особенно сильно на экономические характеристики влияет использование готовых компонентов из предшествующих разработок. При анализе аналогов могут быть выделены компоненты, пригодные для повторного применения в новом проекте. Это позволяет оценить возможную долю использования готовых компонентов и тем самым определить эквивалентный размер комплекса программ, подлежащий непосредственной разработке.

В моделях для первичной оценки экономических характеристик полного цикла производства, размер комплекса программ и остальные факторы рекомендуется учитывать поправочными коэффициентами при уточнении интегральных показателей. Для комплексов программ первого и второго классов, характеристики которых приведены в главе 4, исследовалась зависимость трудоемкости производства программного продукта C от его размеров – $П$. По мере увеличения размера комплекса программ возрастают не только абсолютные, но и относительные затраты на разработку каждой строки текста

в программе. Вследствие этого затраты труда при разработке одной строки текста в программах разного размера могут различаться в несколько раз. Обычно, трудоемкость разработки **программного модуля** пропорциональна квадрату размера текста программы. Модульно-иерархическое построение крупных комплексов программ позволяет замедлить квадратичный рост сложности, и соответствующей ей трудоемкости производства, при увеличении размера программ.

При производстве комплексов программ большого размера может возрасти сложность разработки по сравнению с программами малого размера, так как в них усложняются взаимосвязи компонентов по информации и управлению, а также становятся более трудоемкими процессы планирования, тестирования и управления компонентами в ходе производства. Суммарную трудоемкость разработки сложного комплекса программ можно представить **двумя сомножителями**. Первый сомножитель является **доминирующим**, он прямо пропорционален размеру программы и отражает практически линейное возрастание трудоемкости создания любых комплексов программ при увеличении их размера.

При увеличении размера комплекса программ наблюдается более быстрый рост сложности их разработки, чем описываемый простой линейной зависимостью. Экспериментально подтверждено, что по мере увеличения размера комплекса возрастает относительная трудоемкость разработки каждой строки в программе. Это обстоятельство можно учесть поправочным **вторым сомножителем**, отражающим изменение трудоемкости на разработку каждой строки в программе при увеличении ее размера. Для аппроксимации зависимости трудоемкости от размера комплекса программ удобно использовать **степенную функцию** вида:

$$C = A \times P^E . \quad (8.1)$$

Гипотеза о возрастании трудоемкости разработки с ростом размера комплекса быстрее, чем по линейному закону, справедлива, если показатель степени в полученном уравнении регрессии $E > 1$. В ряде работ при анализе статистики трудоемкости реальных проектов определены коэффициенты A и E , показывающие характер зависимости трудоемкости от размера программного продукта. В таблице 8.1 представлены значения коэффициентов регрессии для моделей КОМОСТ для двух основных классов проектов программных продуктов.

Таблица 8.1.

**Коэффициенты моделей (8.1) для оценки трудоемкости
производства программных продуктов**

Коэффициент <i>A</i>	Коэффициент <i>E</i>	Модель и тип программных продуктов
2,4	1,05	Базовая – КОМОСТ
2,8 3,0	1,20 1,12	Детализированная модель КОМОСТ: - встроенный - СРВ; - полунезависимый – АС.

Выражение (8.1) с использованием этих коэффициентов и значений Π размера комплекса программ (в тысячах строк ассемблера) можно рекомендовать для прогнозирования трудоемкости его полного проектирования и производства (в человеко-месяцах) [3, 16, 46].

При производстве крупных программных продуктов большие затраты приходится на создание технологии, средств автоматизации и унификации производства, на выбор специалистов и предприятия – поставщика компонентов. Небольшие комплексы программ часто разрабатываются неопытными коллективами, которые к тому же пренебрегают автоматизацией технологии и применением современных методов структурного проектирования комплексов программ. Эти обстоятельства приводят к тому, что может возрастать трудоемкость создания относительно небольших комплексов, а рост относительных суммарных затрат на производство крупных комплексов замедляется, что отражается на величине показателя степени E , значения которого в некоторых анализируемых выборках иногда были получены меньше единицы. Однако значительно чаще трудоемкость возрастала при увеличении Π с коэффициентом $E > 1$. На практике изредка пользуются упрощенной *линейной зависимостью трудозатрат* от размера программного продукта ($E = 1$). Такое упрощение при недостаточно корректных исходных данных о размере проекта комплекса

программ и отсутствии сведений о факторах влияющих на производство комплекса программ можно считать допустимым.

На рис. 8.2 по уравнению (8.1) построены в логарифмическом масштабе зависимости трудозатрат C от размера Π для комплексов программ двух классов.

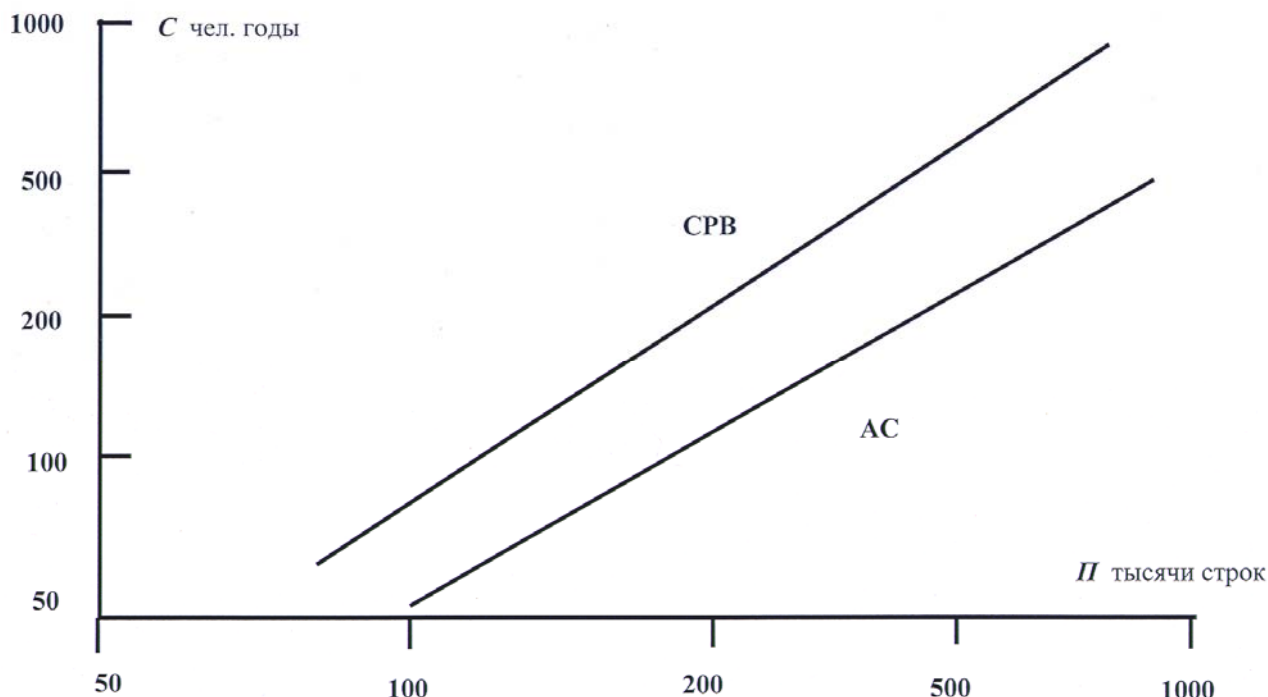


Рис. 8.2

Первый (встроенные – CPB) и второй (AC) классы программного продукта, отчетливо различаются по трудоемкости разработки. Более высокой точности прогноза трудоемкости производства, при учете только одной переменной – размера комплекса, по-видимому, невозможно получить, так как этот процесс производства зависит от большого числа факторов, которые желательно учитывать при прогнозировании трудоемкости. Наибольшие трудозатраты, обычно необходимы для производства крупных программных продуктов реального времени, так как этот класс программ используется в наиболее ответственных автоматизированных системах и требует высокого качества.

Для учета влияния на трудоемкость C основных факторов можно использовать коэффициенты (рейтинги) изменения трудоемкости (КИТ) производства – $M(i)$, учитывающими влияние i -ой составляющей совокупных затрат. В них входят факторы процесса непосредственной разработки, факторы программной и аппаратурной оснащен-

ности, а также квалификация специалистов. При этом затраты на разработку можно представить в зависимости от размера комплекса программ Π , корректируемого произведением коэффициентов изменения трудоемкости (КИТ – $M(i)$) :

$$C = A \times \prod_i^E \times \Pi M(i) . \quad (8.2)$$

Возможные группы и значения коэффициентов $M(i)$ подробно исследованы и отражены в монографиях [3, 16]. Для детальной модели КОМОСТ [3] рекомендовалось четыре группы характеристик: размер объекта производства; квалификация специалистов; технологическая среда; вычислительная среда. Всего было проанализировано около 20-и факторов, для которых подробно рассмотрены изменения соответствующих коэффициентов. Впоследствии при формировании предварительной модели СОСОМО были сохранены эти группы факторов, а их количества сокращено до семи. Одновременно были изменены исходные выражения для прогнозирования трудоемкости и введены масштабные факторы. В предварительной модели СОСОМО масштабные факторы не учитываются.

После того как выбрана архитектура, этапы разработки предварительного проекта и кодирования возможен ввод уточненного размера комплекса программ в предварительной модели трудоемкости (8.2). В *упрощенной предварительной модели СОСОМО* при достаточной достоверности определения масштаба комплекса программ рекомендуется в выражении (8.2) учитывать семь факторов $M(i)$, представленных в таблице 8.2. Для каждого из них в таблице 8.3 приведены значения рейтингов (коэффициентов изменения трудоемкости – КИТ), которые целесообразно использовать при выборе определенного содержания факторов конкретного проекта. Учет этих факторов при прогнозировании экономических характеристик может быть достаточным для проектов комплексов программ на этапе предварительного проектирования. Ориентирами содержания характеристик этих семи факторов могут служить подробные описания 17-ти факторов модели СОСОМО II в главе 9.

Таблица 8.2.

Состав и содержание факторов упрощенной предварительной модели (8.2) СОСОМО

Символ	Содержание групп факторов
	<i>Требования к объекту разработки</i>
<i>M1</i>	Сложность и надежность программного продукта
<i>M2</i>	Требование повторного использования компонентов
	<i>Характеристики коллектива специалистов</i>
<i>M4</i>	Квалификация специалистов и стабильность коллектива
<i>M5</i>	Опыт работы по тематике и с инструментарием
	<i>Технологическая среда производства</i>
<i>M6</i>	Уровень инструментальной поддержки и необходимость распределения производства
<i>M7</i>	Ограничение длительности производства
	<i>Аппаратурно-вычислительная среда производства</i>
<i>M3</i>	Ограничения аппаратной платформы производства и применения продукта

Таблица 8.3.

Коэффициенты изменения трудоемкости производства программных продуктов в упрощенной предварительной модели (8.2) СОСОМО

Факторы	Рейтинги оценки					
	Очень низкий	Низкий	Номинальный	Высокий	Очень высокий	Сверх высокий
Сложность и надежность <i>M1</i>	0,60	0,83	1,00	1,33	1,91	2,72
Требования повторного использования компонентов <i>M2</i>		0,95	1,00	1,07	1,15	1,24
Квалификация специалистов <i>M4</i>	1,62	1,26	1,00	0,83	0,63	0,50
Опыт работы <i>M5</i>	1,33	1,12	1,00	0,87	0,74	0,62
Инструментальная поддержка <i>M6</i>	1,30	1,10	1,00	0,87	0,73	0,62
Ограничение длительности разработки <i>M7</i>	1,43	1,14	1,00	1,00	1,00	
Аппаратурно-вычислительная среда <i>M3</i>		0,87	1,00	1,29	1,81	2,61

Содержание и комментарии к фактору $M1$ – сложность и надежность программного продукта, достаточно подробно изложены в главах 4 и 5, которые целесообразно использовать при определении рейтингов прогнозирования экономических характеристик конкретного проекта.

Повторному использованию программных компонентов – фактор $M2$, посвящена глава 6, в которой содержатся исходные данные для интерпретации этого фактора. Характеристики коллектива специалистов для использования их при оценке рейтингов факторов $M4$ и $M5$, подробно анализируются в главе 9. Там же содержатся разделы, посвященные рейтингам факторов $M6$ и $M7$ технологической среды производства, и фактору $M3$, ограничений ресурсов аппаратной платформы производства и применения программного продукта.

Подобная модель прогнозирования экономических характеристик была создана в проекте ПРОМЕТЕЙ [22]. В модели учитывались **тринадцать факторов**, скомпонованных в аналогичные четыре группы, для комплексов программ реального времени, разработанных на ассемблере. При использовании подобных моделей с множеством факторов, следует учитывать, что для определения рейтингов при прогнозировании экономических характеристик сложных комплексов программ, требуется опыт и интуиция квалифицированных системных аналитиков и экономистов в области программной инженерии.

Длительность разработки программных продуктов является важнейшей экономической характеристикой, поскольку часто она определяет общие сроки разработки систем, а значит, быстроту реализации идей в различных областях автоматизации (см. главу 3). На основе значений трудоемкости C , размера комплекса Π и выбранных значений $M(i)$ могут быть рассчитаны **длительность T** (таблица 8.4) и требуемое среднее **числа специалистов**:

$$T = G \times C^H. \quad (8.3)$$

При определении коэффициентов за начало разработки комплекса программ принят момент создания технического задания, а за окончание – завершение испытаний программного продукта в целом. Диапазону размеров современных программных продуктов в три-четыре порядка (до 10 млн. строк) соответствуют приблизительно такой же диапазон изменения трудоемкости и стоимости производства программных продуктов.

Таблица 8.4.

**Коэффициенты моделей для оценки длительностей разработки
(8.3) программных продуктов**

Коэффициент <i>G</i>	Коэффициент <i>H</i>	Модель и тип программных продуктов
2,5	0,38	Базовая – КОМОСТ
2,5 2,5	0,32 0,35	Детализированная модель КОМОСТ: - встроенный - СРВ; - полунезависимый АС.

Однако вариация длительностей производства программных продуктов значительно меньше, чем вариация их трудоемкости, и не превышает десятикратный диапазон от не-скольких месяцев до 4 – 5 лет. Длительности разработок *T* ограничены сверху и снизу, и одним из основных факторов, определяющих эти границы, является масштаб комплекса программ – *П* (см. главу 4).

Зависимости длительности производства *T* от размера программ *П* значительно различаются для классов комплексов программ. Чтобы сократить ошибки, связанные с неопределенностью измерения размера программ, исследована *зависимость длительности производства от ее трудоемкости*. При этом учитывалась только трудоемкость непосредственной разработки программ *C* без затрат на средства автоматизации разработки. Обработка наборов данных позволила получить коэффициенты уравнения регрессии представленные в таблице 8.4, а также *уравнение регрессии* модели КОМОСТ в зависимости от трудоемкости производства *C* [3].

Длительности производства по этой модели для комплексов программ СРВ и АС оказались очень близкими и не разделены на рис. 8.3. Установлено, что длительность производства комплексов программ меньше подвергается изменениям при автоматизации разработки или другими методами, чем трудоемкость или производительность труда специалистов.

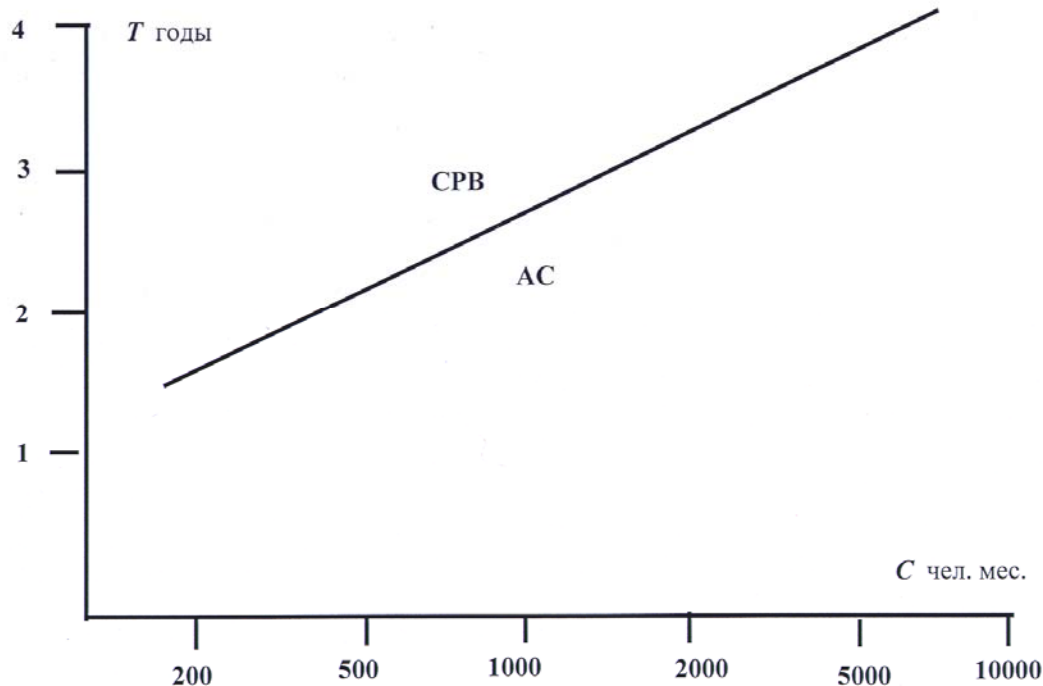


Рис. 8.3

Необходимость выполнения при производстве определенной совокупности этапов и операций в заданной технологической последовательности остается более или менее постоянной при различных воздействиях на процессы разработки. Исключением является применение повторно используемых компонентов (ПИК), при котором значительно сокращаются этапы программирования и автономного тестирования модулей и компонентов программ, а также в той или иной степени длительность других этапов. Поэтому зависимость T , от доли ПИК оказывается нелинейной, и заметное сокращение длительности разработки проявляется только при создании базовой версии программного продукта практически полностью из готовых компонентов.

Оценка требуемого среднего числа специалистов для конкретного комплекса программ предварительно может быть рассчитана путем деления величины трудоемкости производства (8.2) на длительность его производства (8.3): $N = C/T$. **Средняя производительность труда** коллектива специалистов при разработке нового сложного комплекса программ $P = \Pi/C$, конечно, различается для различных классов, размеров и других параметров комплексов программ,

однако диапазон этих различий не столь велик как изменения размера, требований к качеству и других факторов.

Модель прогнозирования экономических характеристик производства программных продуктов СОСОМО II

Основными факторами при прогнозировании экономических характеристик производства программных продуктов выше рассматривались, в основном, сложность, размер – масштаб комплекса программ (см. главу 4) и доля повторного использования готовых программных компонентов (см. главу 6). Это может быть достаточным при первичной оценке затрат, когда размер программ оценивается экспертно и приближенно по общему набору функций программного продукта. Для *более точного экономического обоснования проектов программных продуктов при детальном проектировании* целесообразно учитывать влияние требований качества и ряда производственных факторов. Разработка полного содержания функций и структуры программных компонентов, их взаимодействия и интерфейсов, архитектуры всего комплекса программ и базы данных, обычно позволяет до 10% повысить точность определения размера комплекса, с учетом особенностей языка программирования. Поэтому при прогнозировании трудоемкости разработки, целесообразно выбирать и учитывать влияние ряда дополнительных факторов, которые не оценивались ранее, вследствие их относительно малого влияния. Это позволяет повысить достоверность прогнозирования экономических характеристик до уровня около 5 – 10%.

Накопленный опыт и обобщение проведенных исследований позволили детализировать в таблице 8.5 влияние *четырех основных групп факторов* (тех же, что в таблице 8.2), определяющих экономические характеристики при проектировании и производстве программных продуктов – рис. 8.4:

- факторы, отражающие *требования к характеристикам* создаваемого комплекса программ как объекта проектирования и производства;
- факторы, определяющие *организацию коллектива* для производства программного продукта и его обеспечение квалифицированными специалистами;

- факторы, характеризующие *технологическую среду* и оснащенность инструментальными средствами автоматизации процесса производства программного продукта;
- факторы, отражающие оснащенность процесса производства программного продукта *аппаратурными вычислительными средствами*, на которых реализуется программный продукт и базируются инструментальные системы автоматизации разработки.

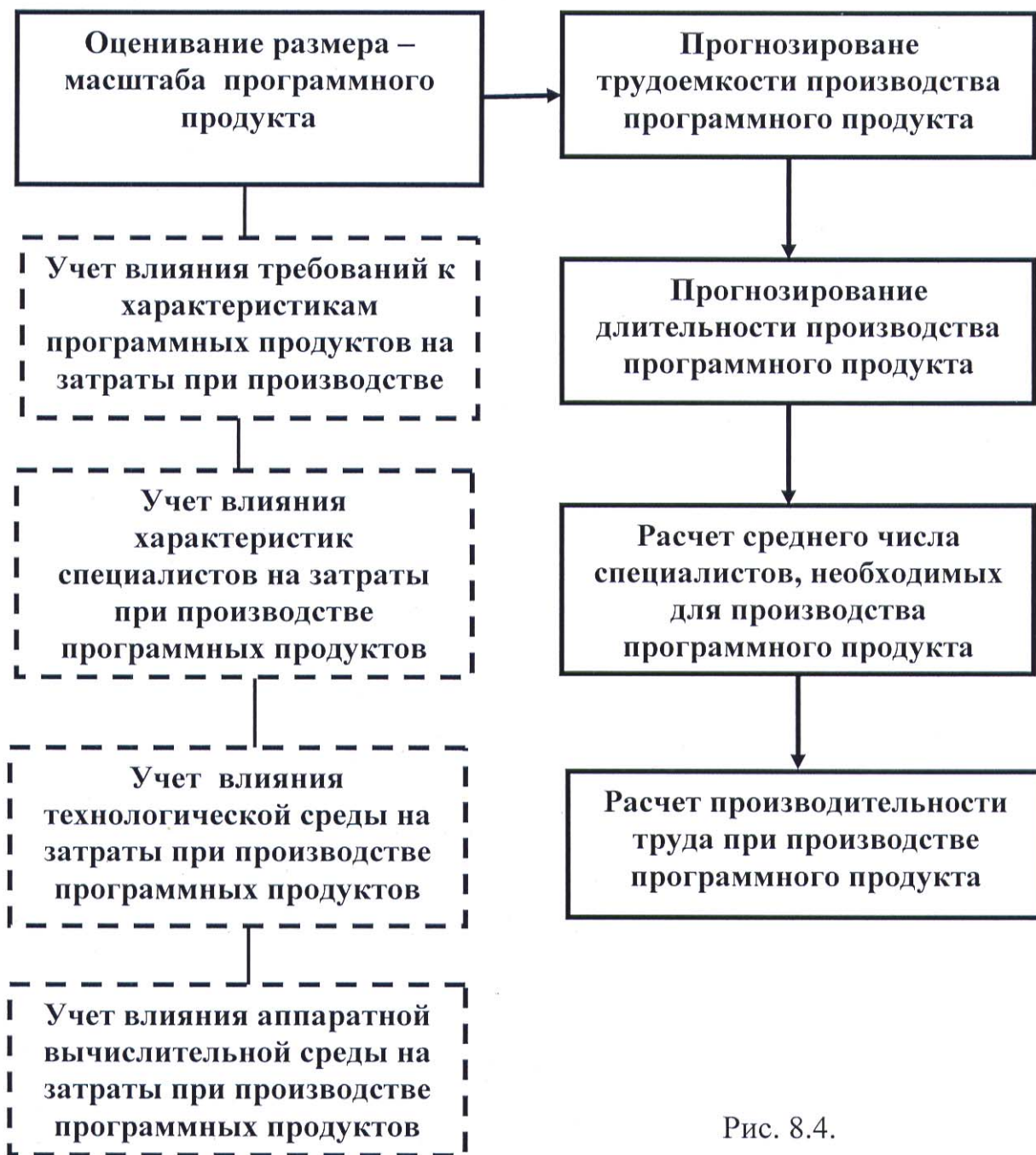


Рис. 8.4.

Таблица 8.5.

Состав факторов детальной модели СОСОМО II

Сим-вол	Содержание фактора
Масштабные факторы	
<i>F1</i>	Новизна проекта
<i>F2</i>	Согласованность с требованиями и интерфейсами
<i>F3</i>	Управление рисками и архитектурой проекта
<i>F4</i>	Слаженность работы коллектива
<i>F5</i>	Технологическая зрелость обеспечения разработки
Определяется сумма рейтингов $F(j)$ для расчетов трудоемкости и длительности в выражениях (8.4) и (8.5)	
Факторы, влияющие на затраты производства	
<i>Требования и характеристики объекта производства</i>	
<i>M1</i>	Надежность функционирования
<i>M2</i>	Размер базы данных
<i>M3</i>	Сложность функций и структуры
<i>M4</i>	Требование повторного использования компонентов
<i>M5</i>	Полнота и соответствие документации проекта
<i>Характеристики коллектива специалистов</i>	
<i>M9</i>	Квалификация аналитиков
<i>M10</i>	Квалификация программистов
<i>M11</i>	Стабильность коллектива
<i>M12</i>	Опыт работы по тематике проекта
<i>M13</i>	Опыт работы в инструментальной среде
<i>M14</i>	Опыт работы с языками программирования
<i>Технологическая среда производства</i>	
<i>M15</i>	Уровень инструментальной поддержки проекта
<i>M16</i>	Необходимость распределенной разработки проекта
<i>M17</i>	Ограничения длительности производства продукта
<i>Аппаратурно-вычислительная среда производства</i>	
<i>M6</i>	Ограниченность времени исполнения программ
<i>M7</i>	Ограниченность доступной оперативной памяти
<i>M8</i>	Изменчивость виртуальной среды разработки проекта
Определяется произведение рейтингов $M(i)$ для расчетов трудоемкости и длительности в выражениях (8.4) и (8.5)	

Для *прогнозирования трудоемкости производства программных продуктов* (человеко-месяцы) в модели СОСОМО II [38] предложены выражения, уточняющие и детализирующие зависимость (8.2), представленную выше:

$$C = A \times E \times \prod_{i=1}^n M(i), \quad (8.4)$$

где: $A = 2,94; \quad E = B + 0,01 \times \sum_{j=1}^5 F(j); \quad B = 0,91.$

В *детальной модели СОСОМО II* на трудоемкость производства программного продукта учитывается влияние 22 факторов – таблица 8.5.

Из них пять – *масштабные факторы*, характеризуются суммой $F(j)$ в значении степени влияния размера комплекса программ E . В уравнении оценки трудозатрат в детальной модели СОСОМО II варьируются и применяются факторы масштабирования $F(j)$ с целью обобщения воздействия основных параметров при прогнозировании производства программных продуктов (таблицы 8.6 и 8.7). Произведение множителей $M(i)$ – 17 факторов, непосредственно влияет на изменение трудоемкости производства в выражении (8.4). Совокупность 22 факторов этой модели представляется достаточно полной при оценивании экономических характеристик производства комплекса программ.

При использовании представленных выражений (8.4) для прогнозирования экономических характеристик конкретных проектов следует выбирать набор факторов (*калибровать модель*), имеющих значения коэффициентов изменения трудоемкости (КИТ) $F(j)$ в соответствии с характеристикам *конкретного проекта*. Для выбора значений $M(i)$ в соответствии с *характеристиками конкретного продукта и среды разработки* следует использовать таблицы рейтингов трудоемкости для выделенных четырех групп, представленные в главе 9 в *таблицах от 9.1 до 9.9*.

Из 17 факторов детальной модели СОСОМО II в этих таблицах исключен только фактор $M16$ вследствие неопределенности способов его оценки. Номинальными (средними) в выражении (8.4) прини-

маются значения $M(i) = 1,00$, при которых соответствующий фактор не влияет на трудоемкость производства программного продукта.

Таблица 8.6.

Масштабные факторы требований к объектам разработки

Факторы	Характеристики факторов					
	Очень низкий	Низкий	Номинальный	Высокий	Очень высокий	Сверх высокий
Новизна проекта $F1$	Полностью новый	Во многом новый	Частично новый	В основном известный	Значительно известный	Полностью известный
Согласованность с требованиями $F2$	Строгая согласованность	Допускаются компромиссы	Значительная	Относительная	Незначительная	При необходимости
Коллективность $F4$	Некоторое взаимодействие	Сложное взаимодействие	Зачастую коллективная работа	Высокая степень взаимодействия	Непрерывное взаимодействие	Активное взаимодействие
Зрелость производства $F5$	СММІ Уровень 1	СММІ Уровень 1	СММІ Уровень 2	СММІ Уровень 3	СММІ Уровень 4	СММІ Уровень 5

Таблица 8.7.

Рейтинги масштабных факторов объектов разработки

Факторы	Рейтинги факторов					
	Очень низкий	Низкий	Номинальный	Высокий	Очень высокий	Сверх высокий
Новизна проекта $F1$	6,20	4,96	3,72	2,48	1,24	0,00
Согласованность требований $F2$	5,07	4,05	3,04	2,03	1,01	0,00
Коллективность $F4$	5,48	4,38	3,29	2,19	1,10	0,00
Зрелость производства $F5$	7,8	6,24	4,68	3,12	1,56	0,00

В модели СОСОМО II поддерживаются вероятностные диапазоны оценок рейтингов, представляющие одно стандартное отклонение из средних оценок. Реально для различных конкретных проектов с учетом их особенностей используется только около половины факторов приведенных в таблице 8.5.

Для прогнозирования длительности производства программных продуктов в модели СОСОМО II более детально рассмотрены исходные данные на основе анализа трудоемкости его производства C . Также как выше (см. выражение (8.3)) **зависимость длительности производства от ее трудоемкости**, практически одинаковая для двух классов комплексов программ (СРВ и АС). При этом рекомендуется учитывать те же наборы факторов, как в выражении (8.4), которые применяются при прогнозировании трудоемкости непосредственного производства программного продукта C :

$$T = G \times C^H. \quad (8.5)$$

где: $G = 3,67$;

$$H = D + 0,02 \times 0,01 \times \sum_{j=1}^5 F(j) = D + 0,02 \times (E - B); \quad D = 0,28.$$

Максимальные величины каждого из КИТ производства программных продуктов экспериментально оценены в [3, 16, 38] при предположении, что остальные параметры зафиксированы. В действительности многие факторы взаимно коррелированы. Так, например, высокой сложности комплекса программ обычно сопутствует требование высокой безопасности и надежности функционирования, а также длительная эксплуатация. Ряд факторов отражается одновременно на нескольких КИТ. Воздействие в процессе производства на такие факторы и субъективное стремление специалистов на сокращение определенных видов затрат, в некоторых случаях, может оказываться нерентабельным с позиции снижения полных затрат на производство программного продукта. Например, стремление уменьшить затраты в период производства без учета последующего использования продукта, его компонентов и всего жизненного цикла может оказаться мало полезным, а, в некоторых случаях, привести к значительному увеличению совокупных затрат.

При создании многих сложных продуктов затраты исчисляются сотнями человеко-лет, что определяет особую актуальность возможно точного учета некоторых факторов, способствующих снижению

прогнозируемых затрат при таких масштабах производства и применения. Поэтому зачастую необходим *системный анализ* и тщательное распределение ресурсов на этапы производства конкретных комплексов программ путем выбора КИТ (калибровки модели) с учетом всего их жизненного цикла.

Большую роль в повышении экономической эффективности производства программных продуктов играет *повторное использование готовых программных компонентов* из других проектов (см. главу 6). При этом необходимо учитывать дополнительные затраты для обеспечения возможности повторного использования компонентов вместо разработки компонентов заново каждый раз, и сокращение интегральных затрат на производство продукта при их применении. В выражениях (8.4) и (8.5) СОСОМО II предусмотрен учет затрат на подготовку компонентов для возможности их многократного применения в составе требований к характеристикам программных продуктов (см. фактор *M4* в таблицах 9.2 и 9.3).

Для оценки и учета сокращения трудоемкости и длительности производства за счет повторного использования готовых компонентов в модели СОСОМО II [38] рекомендуются соответствующие дополнительные выражения и процедуры. При прогнозировании экономических характеристик предлагается в качестве дополнительных данных учитывать долю (%) затрат на программный продукт, которая может быть сокращена за счет применения модифицируемых и/или полностью готовых ПИК. Кроме того, возможно некоторое уменьшение трудоемкости при комплексировании компонентов, документировании и испытаниях программного продукта. Упрощенный учет этого фактора можно проводить путем использования графиков подобных 6.2 и 6.3 в главе 6 для определения относительного уменьшения трудоемкости и длительности создания комплекса программ в зависимости доли повторно используемых компонентов.

В модели СОСОМО II рекомендуется разработка специальным образом *калиброванной версии коэффициентов* в формулах (8.4) и (8.5), которая должна отражать применяемые технологические процессы, особенности и возможности производства программного продукта. При калибровке модели СОСОМО II предлагаются следующие последовательные процедуры для конкретного проекта:

- устанавливаются значения и *сумма рейтингов* масштабных факторов $F(j)$ при которой производится первичное оценивание эко-

номических характеристик;

- выбирается набор, значения и оценивается **произведение рейтингов** факторов $M(i)$, оказывающих наибольшее влияние на прогнозируемую трудоемкость производства программного продукта, которые сравниваются с предыдущей оценкой;

- для каждого набора рейтингов выбранных факторов производится расчет и анализ трудоемкости (8.4) и длительности (8.5) для конкретного проекта комплекса программ.

Рекомендуются три способа [31], для выбора и калибровки коэффициентов изменения трудоемкости модели СОСОМО II при варьировании конкретных условий, которые основаны:

- на различных данных предшествовавших подобных проектов;
- на повторной калибровке уравнений трудозатрат, показателей масштабных факторов и коэффициентов при получении нежелательных значениях трудоемкости и/или длительности проекта и выявление негативных факторов;

- на проверке интегральных и уточнении значений конкретных факторов и допустимых ограничений экономических характеристик производства программного продукта.

Повторная калибровка коэффициентов требует наличия базы данных, которая содержит несколько аналогичных реализованных проектов. Для более точной повторной калибровки характеристик производства, база данных должна включать около пяти проектов.

Преимущества модели СОСОМО II [31, 38] состоят в следующем:

- возможен учет достаточно полной номенклатуры факторов, влияющих на экономические характеристики производства сложных программных продуктов;

- предложены наборы для выбора возможных рейтингов для всей номенклатуры используемых факторов;

- фактические данные подбираются в соответствии с реальными проектами комплексов программ и факторами корректировки, которые могут соответствовать конкретному проекту и предприятию;

- прогнозы производственных процессов являются варьируемыми и повторяемыми;

- метод позволяет добавлять уникальные факторы для корректировки экономических характеристик, связанные со специфическим проектом и предприятием;

- метод является достаточно универсальным и может поддерживать различные размеры, режимы и уровни качества продуктов;
- модель хорошо подходит для проектов, между которыми нет существенных отличий относительно размера, сложности или функций процессов производства;
- возможна высокая степень достоверности калибровки с опорой на предыдущий опыт коллектива специалистов;
- результаты прогнозирования сопровождаются обязательной документацией;
- модель относительно проста в освоении и применении.

Недостатки модели СОСОМО II:

- все результаты зависят от размера программного продукта – точность оценки размера, оказывает *определяющее влияние* на точность прогноза трудозатрат, длительности разработки, численности специалистов и производительности труда разработчиков;
- не достаточно обоснованы и не поддержаны достижимой достоверностью исходных данных некоторые усложнения выражений (8.4) и (8.5) и значения коэффициентов в них;
- игнорируется учет сочетаний комплекса требований к характеристикам качества программного продукта;
- недостаточно учтено влияние требований функциональной безопасности программного продукта;
- не учитывается зависимость между интегральными затратами и количеством времени, затрачиваемым на каждом этапе проекта;
- игнорируется изменяемость требований к программному продукту в процессе производства;
- не достаточно учитывается внешняя среда производства и применения программного продукта;
- игнорируются многие особенности, связанные с аппаратным обеспечением проекта.

Кроме того, модель СОСОМО II изначально представляет *трудозатраты по сумме всех этапов производства* (от этапа планирования концепции до этапа поставки программного продукта). Проблемы сопровождения, переноса и повторного использования компонентов трудно описывать в рамках одной и той же модели. Эти действия могут быть также оценены с помощью дополнений к базовой модели. При использовании модели СОСОМО II предполагается наличие базового уровня трудозатрат, используемого при управлении конфигурацией и обеспечении качества. В этом случае используется

примерно 5% общего бюджета (основываясь на типичной коммерческой практике, принятой при использовании модели СОСОМО II). В модели не учитываются накладные расходы; расходы на командировки и другие побочные расходы при оценках стоимости.

При прогнозировании экономических характеристик производства программных продуктов *для систем реального времени* необходимо учитывать затраты на создание моделей внешней среды, которые могут по величине быть сопоставимыми с затратами на основной программный продукт. Системная интеграция и инструментальная поддержка динамического тестирования и испытаний программного продукта реального времени на соответствие требованиям составляют отдельную специфическую задачу [23], которая не входила в состав целей СОСОМО II и далее не учитывается.

Приведенные недостатки и трудности применения модели СОСОМО II для прогнозирования экономических характеристик производства сложных программных продуктов в значительной степени могут быть скомпенсированы при приобретении квалификации и *накоплении опыта специалистами* экономических оценок в конкретной прикладной области. Рекомендуемые в [38] формулы и таблицы рейтингов целесообразно использовать *как базовые для развития инструментария прогнозирования*, и по мере накопления опыта на конкретном предприятии, уточнять коэффициенты в выражениях (8.4) и (8.5), а также создавать уточненную базу исходных рейтингов и выполненных прогнозов конкретных продуктов.

Глава 9

ВЛИЯНИЕ ОСНОВНЫХ ХАРАКТЕРИСТИК ПРОИЗВОДСТВА ПРОГРАММНЫХ ПРОДУКТОВ НА ЭКОНОМИЧЕСКИЕ ПРОГНОЗЫ ПРИ ИСПОЛЬЗОВАНИИ МОДЕЛИ СОСОМО II

Влияние масштабных факторов производства программных продуктов при прогнозировании экономических характеристик

В предыдущей главе описаны наиболее значительные интегральные факторы модели СОСОМО II, влияющие на трудоемкость программных продуктов. Из них **5 масштабных факторов прогнозирования $F(j)$** могут существенно изменять трудоемкость и стоимость производства. Остальные 17 факторов $M(i)$ объединены в **четыре группы** модели СОСОМО II (см. рис. 8.4 и таблицу 8.5). Каждый из них может влиять на экономические характеристики производства сложных программных продуктов на уровне 5 – 10%.

Разделы данной главы посвящены комментированию факторов из этих групп при этом предполагается, что для основных факторов имеются их описания, представленные в главах 2 – 7. Рекомендуется выбирать и учитывать те факторы, коэффициенты влияния которых на трудоемкость в конкретном проекте, имеют достаточно большую величину, коррелированную с точностью определения размера комплекса программ или превышают ее. В **группу – $F(j)$** следует включать, совокупность факторов из таблиц 8.6 – 8.7, которые способны значительно изменять трудоемкость:

- новизну проекта комплекса программ;
- необходимую степень согласованности проекта с требованиями технического задания;
- наличие управления рисками и архитектурой проекта;
- уровень обобщенной квалификации, слаженности и организованности коллектива разработки проекта;

- уровень обеспечения и оснащения технологии производства по оценке зрелости модели СММІ.

Новизну проекта F1 рекомендуется учитывать при прогнозировании по величине формализации целей и функций программного продукта, наличия у специалистов опыта аналогичных разработок, а также необходимости создания новой архитектуры, компонентов, алгоритмов обработки данных. Влияние этого фактора в выражении (8.4) может превышать коэффициент 6, и достаточно подробно комментирован в таблице 8.6, что позволяет корректно выбирать значения рейтингов в таблице 8.7, адекватные конкретному проекту. При этом в крайних значениях рейтингов предполагается, что полностью отсутствовало проектирование и производство подобного программного продукта или такие продукты были освоены для серийного производства.

Согласованность (жесткость) F2 проекта программного продукта определена как степень обеспечения соответствия заданным требованиям и спецификациям заказчика к компонентам, к внутренним и внешним интерфейсам, ко всему комплексу программ в процессе его проектирования и производства. В первом предельном случае считается, что все требования должны быть выполнены обязательно полностью, а в другом крайнем варианте рейтингов эти требования могут рассматриваться только как необязательные пожелания. Для учета этого фактора предложены рейтинги соответствующих коэффициентов (до 5) изменения трудоемкости разработки (см. таблицу 8.7).

Наличие управления рисками F3 и архитектурой проекта трудно формализовать содержательно и адекватными значениями рейтингов. Поэтому далее в монографии этот фактор не рассматривается.

В модели СОСОМО II значительное внимание уделено **влиянию организации и взаимодействия коллектива квалифицированных разработчиков F4** на трудоемкость создания сложных программных продуктов. Этот фактор подробно иллюстрируется его составляющими в таблице 9.1. В составе организационных характеристик коллектива рекомендуется учитывать согласованность целей специалистов, участвующих в проекте, их психологическую совместимость и способность к дружной коллективной работе, наличие опыта

работы в данном коллективе и другие объективные и субъективные свойства участников проекта.

Таблица 9.1.

Характеристики и влияние коллективизма разработчиков программных продуктов на трудоемкость их производства

Коллективизм	Значения характеристик				
	Очень низкий	Низкий	Номинальный	Высокий	Очень высокий
Согласованность целей коллектива	Минимальная	Незначительная	Относительная	Значительная	Полная
Способность членов коллектива адаптироваться к целям других	Малая	Незначительная	Относительная	Значительная	Полная
Опыт работы в составе данного коллектива	Нет	Малый	Незначительный	Значительный	Большой
Степень доверия и взаимодействия в коллективе	Нет	В малой степени	В некоторой степени	Значительная	Большая
Обобщенная коллективность работ	Некоторое взаимодействие в коллективе	Сложное взаимодействие	Зачастую коллективная работа	Высокая степень взаимодействия	Непрерывное взаимодействие
Обобщенный коэффициент влияния коллективности F_4	5,48	4,38	3,29	2,19	1,10

При этом большое значение может иметь личная мотивация и психологические особенности поведения специалистов при комплексной работе над проектированием и производством сложного программного продукта.

Эти характеристики обобщены в показатель *влияния коллективности – сложности взаимодействия специалистов F_4* в кол-

лективе, которому сопоставлены рейтинги изменения трудоемкости производства продукта (строка в таблицах 8.7 и 9.1). Наилучшим считается непрерывное корректное взаимодействие организованной «команды» специалистов с большим опытом работы в данном коллективе при полной согласованности их целей, планов и методов работы. В остальных случаях может отсутствовать глубокое взаимодействие отдельных специалистов, вследствие чего возрастать (даже в несколько раз) трудоемкость производства продукта.

В модели СОСОМО II для *оценивания экономических характеристик при проектировании и производстве F5* рекомендуется учитывать уровень *методологии* производства и обеспечения качества сложных программных комплексов на основе *системы и модели оценки зрелости СММІ*, применяемых технологических процессов [18, 47]. Она основана на формализации и использовании пяти уровней зрелости технологий поддержки всего ЖЦ комплекса программ, которые также определяют потенциально возможные экономические характеристики и качество создаваемых на предприятии программных продуктов (рис. 9.1).

Чем выше уровень зрелости, тем выше статус предприятия среди поставщиков, доверие к его продукции, ее конкурентоспособность, а также возможное качество продуктов. Тем самым при прогнозировании экономических характеристик и выборе требований качества продукта можно в соответствующей степени доверять поставщику и предприятию разработчика, что оно сможет реализовать согласованные требования заказчика. Эти *уровни зрелости характеризуются*: степенью формализации; адекватностью измерения и документирования процессов производства и программных продуктов; шириной применения стандартов и инструментальных средств автоматизации работ; наличием и полнотой реализации функций системой обеспечения качества технологических процессов и их результатов.

Методология обеспечения качества программных продуктов поддержана рядом методических документов, а также формализована международными стандартами. Качество процессов производства зависит от технологической среды, в которой они выполняются. *Зрелость процессов* – это степень их управляемости, возможности поэтапной количественной оценки затрат и качества, контролируемости и эффективности результатов.

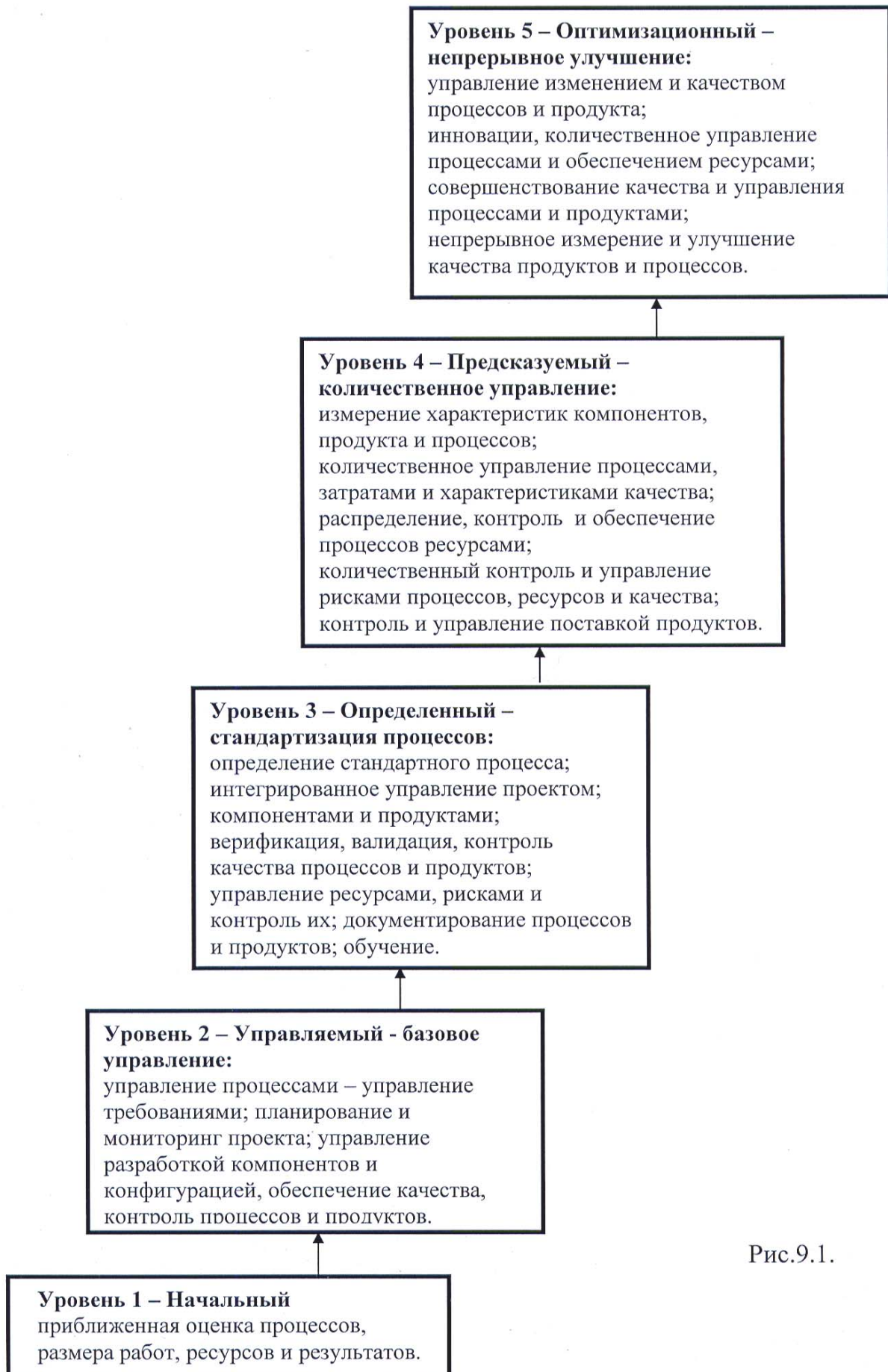


Рис.9.1.

Модель зрелости предприятия представляет собой методический материал, определяющий правила создания и применения системы управления проектированием и производством, повышения культуры и качества комплексов программ.

Реальное использование рекомендуемых процессов производства предполагает их документирование, поэтапный контроль экономических характеристик и качества. На предприятиях, достигших высокого уровня зрелости, процессы производства должны принимать статус стандарта, фиксироваться в организационных структурах и определять производственную тактику и стратегию корпоративной культуры производства и системы обеспечения качества программных продуктов (см. главу 7).

При экономическом обосновании программных проектов, для достижения устойчивых результатов в процессе развития технологии и организации управления жизненным циклом комплексов программ рекомендуется использовать эволюционный путь, который позволяет совершенствоваться, постепенно снижать затраты и повышать качество процессов производства и продуктов. Методология СММІ рекомендует большой комплекс процессов, который предприятие должно выполнять для приобретения, поставки, производства, использования и сопровождения крупных комплексов программ, и виды деятельности, характеризующие степень технологической зрелости этих процессов.

В модели СОСОМО II приводятся количественные рекомендации **коэффициентов влияния уровней зрелости СММІ на трудоемкость** реализации сложных программных продуктов – таблицы 8.6 – 8.7. Рейтингам технологической зрелости производства в детальной модели СОСОМО II сопоставлены уровни СММІ, для каждого из которых приводятся коэффициенты изменения трудоемкости разработки (почти до 8). Применение совершенных технологий и инструментальных средств высшего уровня зрелости может снизить трудоемкость производства сложных программных продуктов более чем в 5 раз по сравнению с наиболее широко применяемыми технологиями второго – третьего уровня зрелости СММІ. Независимую оценку уровня СММІ на предприятии и выбранной технологии производства, можно использовать как **ориентир при прогнозировании и экономическом обосновании производства** предполагаемого продукта.

Факторы четырех групп коэффициентов $M(i)$ рекомендуется выбирать из совокупности перечисленных в таблице 8.5, которые в

конкретном проекте могут повлиять на изменение трудоемкости производства на 10 – 20%, соизмеримое с точностью оценок размера комплекса программ. Анализ, выбор и оценивание коэффициентов влияния $M(i)$, оправдан, когда совместное влияние совокупности этих дополнительных факторов может заметно изменить оценки трудоемкости. Таких факторов практически может быть выделено до 10 из 17-и, влияние каждого целесообразно рассматривать и учитывать при оценках, если он способен изменить трудоемкости разработки конкретного проекта на 5 – 10%. Ниже представлены четыре раздела в соответствии с группами факторов $M(i)$ на рис. 8.4 и в таблице 8.5.

Влияние свойств программных продуктов при прогнозировании экономических характеристик производства

Свойства и характеристики сложных программных продуктов в основном отражены в предыдущих главах 2 – 7. Поэтому в этой главе можно было обойтись небольшими комментариями из описания СОСОМО II [38]. Среди *характеристик объекта разработки*, кроме его размера и доли, повторно используемых компонентов, наибольшее влияние на экономические характеристики могут оказывать требуемая *надежность M1* и *сложность M3* функций комплекса программ. Содержание и возможные значения этих факторов описаны в таблицах 9.2 – 9.3 и подробно рассмотрены выше в главах 4 и 5. Особое влияние на экономические характеристики некоторых критических программных продуктов могут оказывать высокие требования надежности и документированности программного продукта. Они могут увеличивать трудоемкость их производства более чем на 20%, а в некоторых критических системах даже в два раза.

Влияние затрат на обеспечение *требований повторного использования* программных компонентов *M4* проанализированы в главе 6 и достаточно ясно комментируются в таблицах. Фактор *размер базы данных M2* наиболее неопределенный и трудно измеряемый. Поэтому он приводится в таблицах, так как представлен в описании СОСОМО II [38] без дополнительных комментариев. Он отражается соотношением объема памяти, занятой информацией базы данных (БД), к величине памяти используемой комплексом программ(КП).

Соответствие программного продукта документации M5 анализируется в главе 7, поэтому описания в таблице 9.2 и рейтингов

влияния на трудоемкость производства в таблице 9.3 может быть вполне достаточным для учета значений этого фактора.

Таблица 9.2.

Требваемые характеристики программных продуктов

Характеристика	Уровень требований					
	Очень низкий	Низкий	Номинальный	Высокий	Очень высокий	Сверх высокий
Требуемая надежность <i>M1</i>	Малые неудобства	Небольшие возможные потери	Средний возможный ущерб	Высокий финансовый ущерб	Риск для жизни	
Сложность функций комплекса <i>M3</i>	Очень низкая	Низкая	Обычная	Высокая	Очень высокая	Особенно высокая
Размер базы данных <i>M2</i>		БД-байт/ SLOC КП < 10	10 < БД/КП < 100	100 < БД/КП < 1000	100 < БД/КП < 1000	БД/КП > 1000
Требуемое повторное использование <i>M4</i>	Нет	Низкое	В пределах продукта	В пределах семейства продуктов	В пределах фирмы	Для любого продукта
Соответствие документации <i>M5</i>	Фрагменты документов	Документы этапов и компонентов	Эксплуатационные документы	Отдельные технологические документы	Полные технологические и эксплуатационные документы	

Таблица 9.3.

Рейтинги характеристик программных продуктов

Характеристика	Уровень оценки					
	Очень низкий	Низкий	Номинальный	Высокий	Очень высокий	Сверх высокий
Требования надежности <i>M1</i>	0,82	0,92	1,00	1,10	1,26	
Сложность комплекса <i>M3</i>	0,73	0,87	1,00	1,17	1,34	1,74
Размер базы данных <i>M2</i>		0,90	1,00	1,14	1,28	
Повторное использование <i>M4</i>		0,95	1,00	1,07	1,15	1,24
Соответствие документации <i>M5</i>	0,81	0,91	1,00	1,11	1,23	

**Влияние свойств специалистов при прогнозировании
экономических характеристик производства
программных продуктов**

Важнейшим фактором при экономическом прогнозировании, отражающим производство программных продуктов являются люди – *специалисты*, с их уровнем профессиональной квалификации, а также с многообразием знаний, опыта, стимулов и потребностей. Быстрый рост сложности и повышение ответственности за качество привели к появлению *новых требований к специалистам*, обеспечивающим все этапы их жизненного цикла комплексов программ (см. главу 2).

Структура коллектива разработчиков обычно в значительной степени отражает структуру разрабатываемого комплекса программ. Особенно это заметно при создании продуктов размером порядка $10^5 - 10^7$ строк текста. При этом распределение функциональных программ локализуется по соответствующим группам специалистов

с целью минимизации связей и взаимодействия, как между группами разработчиков, так и между создаваемыми ими компонентами комплексов программ. Процесс производства сложных программных продуктов весьма трудоемок, а области применения специалистов различны. Поэтому трудно ожидать, что некий способ организации производства и команды специалистов будет во всех случаях предпочтительнее, чем альтернативные практические варианты. Тем не менее, определенные общие элементы присутствуют во многих успешных командах [24].

В модели СОСОМО II изложены экспертные оценки для учета влияния квалификации и характеристик различных специалистов на экономические характеристики производства программных продуктов (таблицы 9.4 и 9.5) [38].

Таблица 9.4.

Характеристики разработчиков программного продукта

Характеристика	Значение характеристики				
	Очень низкая	Низкая	Номинальная	Высокая	Очень высокая
Квалификация аналитиков <i>M9</i>	15%	35%	55%	75%	90%
Квалификация программистов <i>M10</i>	15%	35%	55%	75%	90%
Тематический опыт <i>M12</i>	2 мес.	6 мес.	1 год	3 года	6 лет
Инструментальный опыт <i>M13</i>	2 мес.	6 мес.	1 год	3 года	6 лет
Опыт работы с языками <i>M14</i>	2 мес.	6 мес.	1 год	3 года	6 лет
Стабильность коллектива <i>M11</i>	48%	24%	12%	6%	3%

Таблица 9.5.

Рейтинги характеристик разработчиков программного продукта

Характеристика	Рейтинги характеристик				
	Очень низкий	Низкий	Номинальный	Высокий	Очень высокий
Квалификация аналитиков <i>M9</i>	1,42	1,19	1,00	0,85	0,71
Квалификация программистов <i>M10</i>	1,34	1,15	1,00	0,88	0,76
Тематический опыт <i>M12</i>	1,22	1,10	1,00	0,88	0,81
Инструментальный опыт <i>M13</i>	1,19	1,09	1,00	0,91	0,85
Опыт работы с языками <i>M14</i>	1,20	1,09	1,00	0,91	0,84
Стабильность коллектива <i>M11</i>	1,29	1,12	1,00	0,90	0,81

Аналогичные оценки особенностей специалистов и их влияния на экономические характеристики, базируются на отечественном опыте создания крупных программных продуктов [24]. Эти характеристики сводятся в основном к длительности и опыту работы в определенной области, и к экспертной оценке (%) квалификации специалистов.

Выбор и использование их значений при прогнозировании экономических характеристик требует высокой квалификации экспертов – руководителей и детального знания особенностей среды производства реального продукта. Для обеспечения возможности их использования при прогнозировании трудоемкости и длительности разработки сложных комплексов программ в модели СОСОМО II представлены соответствующие относительные рейтинги категорий специалистов – таблица 9.5. При применении этих рейтингов следует учитывать, что некоторые из них взаимосвязаны, и целесообразно анализировать их

корреляцию, возможность объединения или исключения при оценке экономических характеристик реальных проектов.

Среди множества *характеристик коллектива разработчиков*, наибольшее влияние на трудоемкость могут оказывать тематическая и технологическая квалификация специалистов, которые в таблицах 9.4 – 9.5 представлены совокупностью из шести факторов. Совместно эти факторы могут изменять трудоемкость производства программных продуктов на 30 – 40%. Кроме того, в модели СОСОМО II выделен и обобщен на основе пяти характеристик коэффициент – комплексная коллективность участников проекта (см. таблицу 9.1), влияние которого может достигать пятикратного изменения трудоемкости.

Разработчики должны иметь в своем составе квалифицированных, *проблемно-ориентированных аналитиков и системных архитекторов – М9*, способных переводить функциональные требования и общие пожелания заказчика в конкретные спецификации и технические требования к комплексу программ и его компонентам (см. таблицы 9.4 и 9.5). Уровень квалификации аналитиков в СОСОМО II предлагается оценивать в процентах от высшей квалификации, что может снизить трудоемкость производства почти на 30% от номинальной, которой соответствует средний рейтинг 1,00 (также как для всех факторов $M(i)$).

Трудоемкость творческой части затрат на производство программных продуктов в *основном определяет человеческий фактор* — квалификация и организация коллектива специалистов. Следствием этого является большой разброс трудоемкости, производительности труда и длительности создания аналогичных программных продуктов разными коллективами. Коллективы с наилучшими экономическими характеристиками могут служить *ориентирами* достижимых в ближайшие годы значений экономических характеристик для соответствующих классов программных продуктов. Однако неуклонно повышается размер и сложность комплексов программ, что вызывает возрастание затрат творческого труда на единицу размера продукта.

Затраты и труд при реализации крупного продукта в *экономике производства*, традиционно принято распределять по *двум категориям специалистов*: разрабатывающим компоненты и комплекс в целом, и обеспечивающим производственную технологию и качество

программного продукта. Организационное разделение специалистов, осуществляющих *производство программного продукта* (первая категория), и *специалистов – технологов*, контролирующих и управляющих его качеством в процессе производства (вторая категория), должно обеспечивать эффективное достижение заданных характеристик, а также независимый, достоверный контроль экономики и качества результатов производства.

Специалисты первой категории непосредственно создают компоненты и комплекс программ в целом с заданными показателями качества (см. таблицу 9.4 – *M10, M13, M14*). В процессе производства их функции заключаются в тщательном соблюдении принятой в предприятии технологии и в формировании всех предписанных руководствами исходных и отчетных документов. При этом предполагается, что выбранная технология способна обеспечить необходимые значения конструктивных показателей качества компонентов и комплекса. Достижение заданных функциональных характеристик гарантируется *тематической квалификацией M12* соответствующих специалистов и регулярным контролем этих характеристик в процессе производства. Система стандартизированного документирования частных работ должна обеспечить объективное отражение качества компонентов и процессов их создания на всех этапах ЖЦ комплекса программ [8, 21, 36].

Разделение труда специалистов этой категории в крупных проектных коллективах приводит к необходимости их дифференциации *по квалификации и областям деятельности*:

- спецификаторы и аналитики *M9* подготавливают требования и описания функций соответствующих компонентов с уровнем детализации, достаточным для корректной разработки программистами текстов программных компонентов и их интерфейсов;
- разработчики программных компонентов – программисты *M10* создают компоненты, удовлетворяющие спецификациям требований, реализуют возможности компонентов продукта, отслеживают и исправляют ошибки, при производстве сложных программных компонентов, что требует детального знания высокоуровневых языков программирования, визуального программирования, сетевых технологий и проектирования баз данных – *M13, M14* в таблице 9.4;
- системные интеграторы сложных проблемно-ориентированных комплексов программ работают над продуктом в значительной

степени отличными от программистов методами, на разных языках производства, используют различные средства автоматизации производства и имеют на выходе различные результаты: крупные функциональные компоненты и комплексы программ;

- тестировщики обеспечивают проверку функциональных спецификаций требований, пользовательских интерфейсов, разрабатывают стратегию, планы и выполняют тестирование на соответствие требованиям для выявления и устранения дефектов и ошибок каждого компонента и всего комплекса программ [2, 26];

- управляющие сопровождением и конфигурацией отвечают за снижение затрат на модификацию и сопровождение программного продукта, обеспечение максимальной эффективности взаимодействия компонентов и производство версий программного продукта, принимают участие в обсуждениях интерфейсов и архитектуры программного продукта [22];

- документаторы объектов и процессов жизненного цикла комплекса программ обеспечивают подготовку и издание сводных технологических и эксплуатационных документов на программный продукт в соответствии с требованиями стандартов [21].

Однако в модели СОСОМО II в составе «команды» специалистов для производства сложного программного продукта *не учитываются такие важнейшие квалификации* как:

- тестировщики компонентов и комплексов программ;
- системные интеграторы компонентов сложных комплексов программ;
- управляющие сопровождением и конфигурацией комплексов программ.

При оценке трудоемкости производства комплекса программ влияние этих факторов можно учитывать, увеличивая, например, характеристики (изменяя рейтинги): программистов (с тестировщиками), аналитиков (с интеграторами), инструментальный опыт (с сопровождением), а также стабильность коллектива.

Успех и качество при разработке сложных программных продуктов все больше зависит от слаженности работы и профессионализма коллектива специалистов на всех этапах и уровнях производства продуктов – *от стабильности коллектива М11* (таблицы 9.4 и 9.5). Особенно важна не индивидуальная характеристика каждого специалиста, а, прежде всего, *интегральный показатель квалифи-*

кации «команды», реализующей некоторую, достаточно крупную функциональную задачу или весь проект комплекса программ [3, 24].

Тематическая квалификация и опыт – M12 специалистов в конкретной прикладной области, для которой, разрабатывается программный продукт, приближенно может оцениваться продолжительностью работы по данной тематике. При низкой тематической квалификации допускаются наиболее грубые системные ошибки, требующие больших затрат для доработки программ. Имеются примеры, когда из-за таких ошибок, допущенных на этапе системного анализа, приходилось в процессе производства изменять до 70 – 90% программ.

Целесообразность использования в качестве показателя квалификации значений **длительности работы в определенной прикладной области** или специализации, подтверждается достаточно высокой ее корреляцией с коэффициентом изменения трудоемкости. При этом квалификация системных аналитиков и непосредственных разработчиков программ в конкретной прикладной области особенно важна не столько как индивидуальная характеристика каждого специалиста, а прежде всего как интегральный показатель «команды», реализующей достаточно крупную функциональную задачу. Приводимые в разных работах оценки показывают, что при изменении опыта работы в данной области от 1 до 10 лет производительность труда может повышаться в 1,5 – 2 раза, что отражается величиной рейтингов **M12**.

Технологическая квалификация – M13 программистов в использовании инструментальной системы автоматизации производства программных компонентов отражает опыт применения методов, средств и всего технологического процесса при создании данного типа программных модулей и компонентов. Этот опыт можно характеризовать длительностью работы с конкретной инструментальной системой автоматизации или ее версиями, базирующимися на единых технологических концепциях, опыте и длительности работы с регламентированными технологиями, инструментальными комплексами автоматизации разработки, **языками проектирования, программирования и тестирования программ**. Особое значение имеет коллективный опыт организации и выполнения сложных проектов на базе современных автоматизированных технологий и инструментальных средств программной инженерии. Опыт применения конкретного

комплекса автоматизации, языков проектирования и программирования может являться существенным фактором при выборе технологии для создания новых компонентов и обеспечении качества программных продуктов (см. рейтинги в таблице 9.5).

Квалификация специалистов существенно зависит от *стабильности состава и психологического климата* в коллективе, их способности к сотрудничеству и дружной совместной работе над единым проектом (см. *F4* в таблице 9.1 и *M11* в таблицах 9.4 и 9.5). В данном факторе годы работы с конкретной технологической системой отражают не только опыт работы с инструментами, но и слаженность коллектива при проведении больших комплексных работ. Нарушение технологии, задержка при разработке отдельных модулей или групп программ может приводить к большим дополнительным затратам и значительной задержке производства продукта в целом. Это вызывает простои групп специалистов, соответствующее увеличение совокупных затрат и снижение производительности труда. Однако чаще всего коллективы специалистов имеют уже некоторый технологический опыт, и дальнейшее повышение их квалификации может дать относительно немного 10 – 20% производительности. Небольшое влияние этого фактора обусловлено также нивелирующим влиянием трудозатрат вспомогательного и руководящего персонала, которые практически не зависят от технологической квалификации и инструментального опыта.

Программистская квалификация специалистов – *M10*, *M14* и опыт работы с языками программирования, из приведенных факторов квалификации относительно слабо отражается на производительности труда при сложном программном проекте. В данном факторе учитывается освоенность не только языка непосредственного программирования, но и всех компонентов средств, используемых при создании программ (спецификаций, диалога, тестирования компонентов и их комплексирования). После двух-трех лет работы проявляются индивидуальные особенности конкретных специалистов, их творческие способности, тщательность в работе, рациональность использования средств автоматизации. При производстве сложных комплексов после первых лет работы возрастание программистской квалификации может повысить общую производительность труда на 5 – 10% .

Совокупность перечисленных видов квалификации можно представить как **обобщенный человеческий фактор «команды» при экономическом прогнозировании** в процессе производства сложных программных продуктов. На эту величину влияет еще множество трудно учитываемых условий. Сложность психологических экспериментов с регистрацией всех основных условий производства приводит к большому разбросу результатов. В некоторых случаях этот фактор изменял совокупные затраты даже в 2 – 5 раз. Вследствие этого некоторые специалисты в основном на **опыте разработки относительно простых небольших программ** утверждают, что при производстве комплексов программ только человеческий фактор программистов определяет длительность и трудоемкость разработки. Отсюда делается вывод о принципиальной невозможности планирования и прогнозирования процессов и экономики производства программных продуктов. В действительности при крупных разработках **человеческий фактор в коллективах значительно усредняется**. Тем не менее, достаточно часто по этой причине возможен разброс производительности труда в 2 – 3 раза при производстве крупных практически аналогичных продуктов [24, 36].

Специалисты второй категории – технологи, создающие, обслуживающие и сопровождающие технологический инструментарий, который применяется специалистами первой категории, обеспечивают применение системы качества предприятия, контролируют и инспектируют процессы производства для минимизации затрат. Основные задачи второй категории специалистов должны быть сосредоточены **на контроле экономических характеристик** процессов и результатов выполнения работ и на принятии организационных и технологических мер для достижения необходимого качества, обеспечивающего выполнение всех требований технического задания на программный продукт с учетом производственной экономики.

Технологи должны выбирать, приобретать и осваивать **экономически наиболее эффективный инструментарий** для проектов, реализуемых конкретным предприятием с учетом особенностей создаваемых комплексов программ требуемого качества и рентабельности технологических средств. Они должны разрабатывать регламентированный технологический процесс и систему качества, поддерживающие весь ЖЦ комплексов программ и обучать разработчиков ква-

лифицированному применению соответствующих инструментальных средств и технологий.

Специалисты, управляющие обеспечением качества программных продуктов, должны владеть стандартами и методиками предприятия, поддерживающими регистрацию, контроль, документирование и воздействия на показатели качества и экономические характеристики на всех этапах производства комплекса программ. Они должны обеспечивать эксплуатацию системы качества проекта, выявление всех отклонений от заданных показателей качества продуктов и процессов, а также от предписанной технологии. В результате должны приниматься меры либо по устранению отклонений, либо по корректировке требований, если устранение отклонений требует больших ресурсов.

Для планирования и управления жизненным циклом концептуально целостных, крупных программных продуктов и обеспечения их качества в пределах допустимых затрат, необходимы организационные действия менеджеров, направленные на подбор и обучение коллектива специалистов разных категорий и специализаций. Перечисленные выше специализации и квалификации персонала, участвующего в крупных проектах, **требуют соответствующей их подготовки, отбора и обучения**, которые являются самостоятельной, важной **проблемой в экономике производства** программных продуктов. Обучение представляет собой сложный процесс, требует организации и сопровождения обучаемого персонала. Должны быть разработаны и документированы планы, требования и цели обучения, а также разработаны материалы, используемые для обучения [24, 30]. Может быть необходимым включение в подготовку, ознакомление со специфической (проблемно-ориентированной) областью, в которой будет работать данный программный продукт, и повышение квалификации в этой области.

По мере повышения квалификации коллектива и автоматизации творческой части труда следует ожидать асимптотического приближения к **потенциальным значениям интегральных экономических характеристик** производства новых программных продуктов. Эти предельные значения определяются возможностями человека по интенсивности принятия творческих решений. При **экономическом прогнозировании проектов** сложных комплексов программ рассчитать такие характеристики вряд ли возможно, и реальным путем их

оценки являются изучение и экстраполяция экспериментальных данных реальных разработок программных продуктов с учетом возрастания квалификации специалистов и уровня автоматизации. Эти факторы способствуют эволюционному, относительно медленному, приближению к предельным экономическим характеристикам производства новых крупных продуктов. Предельные значения производительности труда, по-видимому, в два-три раза выше, чем современные значения для наилучших коллективов при среднем уровне автоматизации разработки и квалификации специалистов. Вряд ли можно ожидать в ближайшие годы повышения производительности труда на порядок **для полностью новых, сложных программных продуктов**. Еще более консервативна длительность разработки. Принципиальным путем улучшения экономических характеристик при производстве сложных программных продуктов является **сокращение творчества** на тех этапах, где возможны типовые стандартные решения и заготовки, не требующие при их многократном применении высококвалифицированного творческого труда.

Представленные в данном разделе факторы и экономические оценки являются усредненными и относятся к большим коллективам специалистов при производстве сложных программных продуктов. При этом целесообразно учитывать распределение затрат по этапам производства и необходимую численность занятых в коллективе специалистов – % от средней (см. таблицу 6.1 для **новых** сложных комплексов программ реального времени). На этапах предварительного и детального проектирования в коллективе может быть занято около половины специалистов (соответствующих категорий от среднего числа). В программировании и тестировании компонентов число участвующих специалистов превышает средние значения приблизительно в полтора раза, а затем при интеграции компонентов, испытаниях и документировании комплексов программ соответствует среднему уровню. Эти оценки учитывают все категории специалистов, включая руководителей и вспомогательный персонал, которые не прикасаются к программам.

Влияние технологической среды производства программных продуктов при прогнозировании экономических характеристик

Для приближенной оценки влияния на трудоемкость **инструментальных характеристик производства** в модели СОСОМО II

выделен показатель и соответствующий набор рейтингов (таблицы 9.6. и 9.7).

Таблица 9.6.

Характеристики процессов производства программных продуктов

Характеристики	Содержание характеристик				
	Очень низкое	Низкое	Номинальное	Высокое	Очень высокое
Инструментальная поддержка <i>M15</i>	Отдельные простые инструменты	Простые интегрированные инструменты	Базовые интегрированные инструменты	Комплексные интегрированные системы	
Ограничения сроков разработки <i>M17</i>	75%	85%	100%	130%	160%

Таблица 9.7.

Рейтинги процессов производства программных продуктов

Характеристики	Рейтинги характеристик				
	Очень низкое	Низкое	Номинальное	Высокое	Очень высокое
Инструментальная поддержка <i>M15</i>	1,17	1,09	1,00	0,90	0,78
Ограничения сроков разработки <i>M17</i>	1,43	1,14	1,00	1,00	1,00

Инструментальные системы – M15, поддерживающие производство, могут быть описаны качественными характеристиками и

рейтингами, изменяющими трудоемкость в пределах приблизительно 20% от средней – номинальной. Уровень технологии и комплекса инструментальных средств особенно сильно влияет на **экономику крупных программных продуктов**. Поэтому затраты на их реализацию и применение, целесообразно учитывать конкретно с использованием функций и характеристик проекта. Рейтинги инструментальной поддержки коррелированы с уровнями зрелости технологий производства программ СММІ – *F5* и комментированы выше.

Затраты на технологию и программные средства автоматизации производства обычно являются весьма весомыми при использовании высокоэффективных автоматизированных технологий и инструментов. При **экономическом обосновании проекта следует учитывать**, что размер и сложность создаваемого продукта значительно влияет на выбор инструментальных средств и уровня автоматизации технологии, а также на долю этих затрат в общих затратах на производство. Встречаются ситуации, при которых затраты на технологию достигают 50% общих затрат на производство. Такие затраты могут быть **оправданы повышением производительности труда, сокращением сроков разработки** и последующим снижением затрат на множество версий программного продукта. Однако чаще эта группа затрат при создании первой версии сложного программного продукта находится в пределах 20 – 30% от суммарных затрат. В первом приближении степень автоматизации производства комплексов программ отражается размером программных средств, используемых в технологических системах. Этот показатель соответствует сложности систем автоматизации производства программ и пропорционален затратам на их приобретение или создание, и эксплуатацию.

Стремление уменьшить технологические затраты в период производства без учета последующего использования программного продукта, его компонентов и всего жизненного цикла, может оказаться мало полезным, а в некоторых случаях привести к значительному увеличению совокупных затрат. При применении сложных программных продуктов эти затраты могут исчисляться сотнями человеко-лет, что определяет особую актуальность их снижения. Поэтому необходим системный анализ распределения и использования технологических ресурсов на разработку комплексов программ **с учетом всего их жизненного цикла**, включая сопровождение и возможный перенос на другие платформы.

В ряде случаев технология и средства автоматизации создаются для производства многих программных продуктов. При этом затраты на их создание входят в стоимость каждой версии продукта только некоторой долей. Величина этой доли зависит от широты применения данного средства, т.е. от общего числа экземпляров технологической системы, использованных за цикл жизни технологии или за некоторый нормативный срок. При производстве каждого продукта эта доля увеличивается на величину, обусловленную затратами на внедрение и освоение технологии и на величину, связанную с ее эксплуатацией. Таким образом, **затраты на технологическое обеспечение производства программного продукта** можно представить следующими составляющими:

- долей суммарных затрат на приобретение или создание технологии и системы автоматизации производства программ;
- однократными затратами на внедрение и освоение технологии и средств автоматизации, а также на подготовку версии технологической системы, адаптированной к конкретным условиям применения для разработки данного программного продукта;
- затратами на эксплуатацию средств автоматизации в течение всего календарного времени производства или ЖЦ данного программного продукта;
- на создание технологии и инструментальных средств для испытаний и оценивания достигнутых значений характеристик качества программного продукта.

Стремление использовать апробированные технологии и инструментальные системы приводит к тому, что в этих случаях первую составляющую иногда можно игнорировать. Некоторые средства автоматизации используются в готовой версии и не адаптируются к конкретным условиям применения, тогда доминирующей становится последняя составляющая. Минимизации этих затрат способствует массовое применение типовых технологий с высоким уровнем автоматизации.

Уровень автоматизации, качество технологии и инструментальных средств, используемых для поддержки всего жизненного цикла комплекса программ, обычно сильно коррелирован с достигаемым качеством производства комплекса, а также с качеством средств автоматизации для оценивания этого качества. Оценивание качества технологической базы производства позволяет **прогнозировать воз-**

можное качество программного продукта и ориентировать заказчика и пользователей при выборе разработчика и поставщика для определенного проекта с требуемыми характеристиками качества. Очевидно, что низкий уровень технологии и средств производства программ, не может обеспечить их высокое качество и достоверное его оценивание. Поэтому определение уровня **зрелости** технологической поддержки процессов жизненного цикла, организационного и инструментального обеспечения качества, непосредственно связано с выбором и оцениванием реальных или возможных характеристик качества конкретного программного продукта [36, 40].

При производстве комплексов программ для систем реального времени **большие затраты** могут потребоваться для обеспечения тестирования, отладки и испытаний продукта, которые отдельно **не учитываются моделью СОСОМО II**. Приведенные в главе 6 (таблица 6.1) распределения трудоемкости и длительности по этапам работ получены экспериментально, когда относительно мало внимания уделялось организации проектирования, автоматизации начальных этапов разработки, применению спецификации требований и использованию отладки в реальном времени. Поэтому основные усилия сосредотачивались на процессах программирования и автономной отладки компонентов. На этих этапах выявлялась основная масса ошибок, хотя при использовании программного продукта и сопровождении также обнаруживалось некоторое их количество. Впоследствии центр тяжести процессов производства сложных комплексов программ, сдвинулся к начальным этапам, и внимание было акцентировано на **предотвращение ошибок**, прежде всего путем тщательного системного проектирования из готовых программных компонентов, а также на комплексной отладке и испытаниях в реальном времени. Вследствие этого распределения во времени анализируемых экономических характеристик деформировались, их размеры сдвинулись в сторону начальных и конечных этапов.

Этапы комплексной отладки, испытаний и модификации комплексов программ имеют много общего, в основе которого лежит широкое применение их тестирования для обнаружения ошибок и удостоверения функциональной корректности программного продукта. Разработка детерминированных тестов при отладке модулей и некоторых групп программ в большинстве случаев производится вручную. Однако их доля может составлять заметную часть общих затрат

на тестирование компонентов. Достаточно автономными и локализуемыми обычно являются затраты на стохастические тесты и динамические тесты в реальном времени, используемые при комплексной отладке и испытаниях продукта. Для этого необходимы методы и средства автоматизации, обеспечивающие генерацию динамических тестов для проверки функциональных групп программ и программного продукта в целом в реальном времени.

Значительная доля от совокупных затрат на производство программного продукта реального времени может приходиться на создание и эксплуатацию средств *обеспечения динамического тестирования и испытаний*, которая может быть соизмеримой и даже превышать затраты на непосредственную разработку программного продукта. Сложность программного продукта зачастую *адекватна сложности внешней среды и/или системы*, являющейся источником обрабатываемой информации и потребителем обработанных данных, получаемых при функционировании программного продукта. Только для очень небольших программ можно вручную разработать достаточно полные наборы тестов для отладки, отражающие реальные условия их эксплуатации. Для средних и больших комплексов программ, необходим принципиально иной подход, заключающийся в создании и применении автоматизированных, динамических моделей внешней среды, в качестве источников тестовых данных в реальном времени, при тестировании программного продукта на соответствие требованиям заказчика. Затраты на них необходимо *учитывать при экономическом прогнозировании и обосновании таких проектов*.

Имитировать динамические тесты, особенно сложно для комплексов программ, функционирующих в реальном времени, когда данные, отражающие внешнюю среду, должны поступать синхронно также в реальном времени. Анализ эффективности подобной имитации динамической внешней среды при производстве целесообразно разделить на две части: оценка факторов, определяющих эффективность программных средств динамической имитации тестов, и оценка *экономического выигрыша* при моделировании внешней среды на ЭВМ по сравнению с натурными экспериментами. Программная *имитация динамической внешней среды на ЭВМ позволяет*:

- расширять диапазоны характеристик имитируемых объектов за пределы реально существующих или доступных источников дан-

ных и генерировать динамические потоки информации, отражающие перспективные характеристики создаваемых систем;

- создавать тестовые данные, соответствующие критическим или опасным ситуациям функционирования объектов внешней среды, которые невозможно или рискованно реализовать при натуральных экспериментах;

- обеспечивать высокую повторяемость имитируемых данных при заданных условиях их генерации и возможность прекращения или приостановки реального времени имитации на любых фазах моделирования внешней среды;

- проводить длительное непрерывное генерирование имитируемых данных для определения безопасности и надежности функционирования программного продукта в широком диапазоне условий, что зачастую невозможно при использовании реальных объектов внешней среды [1, 17, 31].

В этой же паре таблиц 9.6 и 9.7 представлено влияние на трудоемкость директивного **ограничения заказчиком сроков производства продукта – M17** относительно типовых – номинальных. При ограничении сроков до уровня 75% от номинальных, сокращению трудоемкости может сопутствовать значительное снижение качества и увеличение рисков реализации проекта. В приведенных данных неявно предполагается, что руководство проекта заранее знает о требуемом уменьшении или возможном увеличении сроков производства и в состоянии вести планирование и управление проектом наиболее выгодным способом, с точки зрения минимизации трудоемкости или стоимости. Для уменьшения сроков разработки есть ряд путей, с помощью которых руководство может добиваться **некоторого ускорения разработки за счет увеличения трудоемкости и стоимости продукта**:

- обеспечить детальное структурирование комплекса программ на модули и спецификации интерфейсов для обеспечения максимального параллелизма работы специалистов – программистов и тестировщиков;

- приобрести технологические, инструментальные средства для более быстрого кодирования, контроля, тестирования компонентов, и обучить разработчиков их использованию;

- обеспечить дополнительную подготовку программистов и тестировщиков к работе в тематической области функций программного продукта;
- привлечь дополнительный вспомогательный персонал;
- отложить на время некоторое не существенное документирование программного продукта.

Тем не менее, есть предел сокращению сроков производства от номинальных с помощью увеличения числа специалистов и приобретения дополнительного оборудования. При максимально возможном сокращении сроков разработки до 75% от оптимальных, затраты возрастают на 30 – 40% (см. таблицу 9.7).

В таблицах 9.6 и 9.7 пропущен фактор и рейтинги – *M16* модели СОСОМО II – *необходимость распределенной разработки программного продукта*. Это обусловлено неопределенностью его описания в [38] и относительно небольшим влиянием этого фактора на экономические характеристики производства программного продукта

Влияние аппаратной вычислительной среды при прогнозировании экономических характеристик производства программных продуктов

В некоторых комплексах программ реального времени (СРВ) на увеличение трудоемкости разработки (до 50%) может оказать *ограничение вычислительных ресурсов* оперативной памяти – *M7* и производительности объектной ЭВМ – *M6*, на которой должен функционировать программный продукт (таблицы 9.8 и 9.9).

При таких ограничениях ресурсов резко усложняется проектирование и производство продукта и приходится адаптировать размеры алгоритмов и программ с целью не нарушить ограничения вычислительных ресурсов. Это привело к необходимости разрабатывать *методы эффективного использования программами аппаратных ресурсов ЭВМ*.

На практике наибольшие технические трудности обычно вызывают *ограничения производительности M6* реализующей ЭВМ, для преодоления которых приходится не столько экономить программные ресурсы и сокращать размеры программного продукта, сколько искать пути увеличения производительности ЭВМ и эффективности их использования.

Таблица 9.8.

**Ограничения вычислительной среды реализации
программных продуктов**

Характеристика	Содержание характеристики				
	Низкое	Номинальное	Высокое	Очень высокое	Сверх высокое
Ограничения времени исполнения <i>M6</i>		< 50%	70%	85%	95%
Ограничения по оперативной памяти <i>M7</i>		< 50%	70%	85%	95%
Изменчивость среды производства <i>M8</i>	Каждые 12 месяцев	Каждые 6 – 0,5 месяца	Каждые 2 – 0,25 Месяца	Каждые 0,5 – 0,1 месяца	

Таблица 9.9.

**Рейтинги ограничений вычислительной среды реализации
программных продуктов**

Характеристика	Рейтинги характеристики				
	Низкое	Номинальное	Высокое	Очень высокое	Сверх высокое
Ограничения времени исполнения <i>M6</i>		1,00	1,11	1,29	1,63
Ограничения по оперативной памяти <i>M7</i>		1,00	1,05	1,17	1,46
Изменчивость среды производства <i>M8</i>	0,87	1,00	1,15	1,30	

При этом существенным ограничивающим фактором являются реальное время, в течение которого ЭВМ может быть предоставлена для решения данной задачи, или то время, в пределах которого целесообразно получить результаты (время отклика) для их практического использования. Таким образом, уже на стадии формулирования требований к программному продукту в некоторых проектах (например, бортовые системы) следует сопоставлять доступную производительность ЭВМ с необходимыми затратами времени для решения требуемого набора задач или с допустимой длительностью ожидания результатов (временем отклика).

Поэтому весьма актуальной для систем реального времени является *проблема эффективного распределения ограниченной производительности ЭВМ* для решения комплекса функциональных задач и поиск методов рациональной организации вычислительного процесса при применении программного продукта.

Быстрый рост количества решаемых задач, их сложности и требуемой производительности вычислительных средств стимулировал поиск различных путей удовлетворения потребностей программных продуктов в ресурсах для решения таких задач. Мировая практика развития вычислительной техники показала, что потребность в вычислительных ресурсах для решения некоторых задач растет быстрее, чем доступные реальные характеристики ЭВМ. Технические трудности улучшения характеристик процессоров и памяти непосредственно за счет увеличения быстродействия элементной базы вычислительных систем привели к созданию многомашинных и многопроцессорных комплексов.

В таких ЭВМ проблема эффективного распределения и использования ресурсов еще больше усложняется, так как необходимо оперативно распределять производительность (в принципе различных) вычислительных процессоров для решения всей заданной совокупности задач. Так как задачи могут быть взаимосвязаны последовательностью решения или обрабатываемой информацией, возникала необходимость эффективного использования памяти и производительности вычислительных систем *при параллельном решении совокупности взаимодействующих задач*.

В последние годы стали возможными ситуации, когда *в современных ЭВМ реального времени* может быть создан такой избыток производительности и памяти, что нет необходимости в детальном

выборе и применении методов рационального и экономного использования вычислительных ресурсов при экономическом прогнозировании и обосновании таких проектов. Однако применение более дешевых и простых ЭВМ может позволить те же задачи решать при меньших аппаратных затратах, особенно это актуально для мобильных ЭВМ реального времени (например, бортовых систем). При ограничениях ресурсов вследствие требований минимизации весов и габаритов специализированных, реализующих ЭВМ в авиационных, ракетных и космических системах, их *экономное использование остается актуальным* и следует учитывать при проектировании и производстве соответствующих программных продуктов.

Влияние *ограничения объема памяти реализующей ЭВМ – М7* в модели СОСОМО II, определяется долей от фактического размера доступной оперативной памяти, которая может использоваться для размещения программного продукта и данных при решении возлагаемых на ЭВМ совокупности задач с приемлемой для практики точностью и надежностью. Ограничения этого фактора реально меньше влияют на экономические характеристики производства программных продуктов, так как технически легче преодаливаются при конструировании ЭВМ.

Изменчивость среды разработки – М8 (технологических средств) может быть вызвана изменением и расширением функций программного продукта, модернизациями устранения ошибок, частота которых вызывает смену версий программного продукта и соответствующие дополнительные затраты на инструментальную среду производства. Номинальной в модели СОСОМО II принята стабильность вычислительной среды, когда изменения происходят в среднем каждые 3 месяца. При более частых изменениях среды (каждую неделю) трудоемкость может возрасти на 30% (см. таблицу 9.9).

При производстве крупных комплексов программ и ограниченных ресурсах реализующих ЭВМ реального времени, кроме этих машин могут быть необходимы дополнительные технологические и моделирующие внешнюю среду ЭВМ. В полном жизненном цикле комплексов программ для систем реального времени в этом случае следует прогнозировать *затраты на ЭВМ для реализации* программного продукта, автоматизации его разработки – на *технологической ЭВМ* и для имитации внешней среды – на *моделирующей ЭВМ*. Их применение изменяется по этапам и значительно зависит

от требований к качеству продукта, от его сложности, предполагаемого тиража и ряда других факторов. Соотношение между этими затратами зависит от уровня автоматизации производства комплекса программ. При низких уровнях автоматизации все функции разработки решаются на реализующей ЭВМ. По мере возрастания автоматизации повышается доля использования технологических ЭВМ, а затем и моделирующих ЭВМ, имитирующих внешнюю среду. При создании особо сложных программных продуктов реального времени с применением высоко автоматизированных технологий (уровни СММІ – 4 – 5) суммарные затраты на применение реализующей, технологической и моделирующей ЭВМ могут становиться соизмеримыми.

Примеры прогнозирования экономических характеристик производства программных продуктов

Ниже представлены и анализируются *гипотетические варианты прогнозов экономических характеристик программных продуктов*, рассчитанных с использованием программного пакета ПЛАПС-2 – на базе формул модели СОСОМО II (выражения (8.4) и (8.5) с коэффициентами из таблиц от 8.5 – 8.7 и 9.1 – 9.9). Пакет ПЛАПС-2 – рис. 9.2 (разработчик Н. Липкин) предназначен для прогнозирования экономических характеристик проектирования и производства проектов сложных программных продуктов.

На панели слева отображается определенный пользователем размер комплекса программ. При использовании пакета возможно указание различных факторов и характеристик проекта, а результаты отражаются сверху на окне «*Экономические характеристики*» (см. рис. 9.2). В них содержатся значения прогнозов трудоемкости и длительности реализации проекта (первая и вторая строка), а также производительности труда (четвертая строка) и число специалистов, необходимых для производства прогнозируемого комплекса программ (третья строка). При проведении оценок возможен учет повторного использования готовых компонентов для сборки нового программного продукта. Параметры модели логически сгруппированы и расположены на различных вкладках в соответствии со структурой таблицы 8.5. Программный пакет представляет собой стандартное Windows приложение с комплектом эксплуатационной документации.

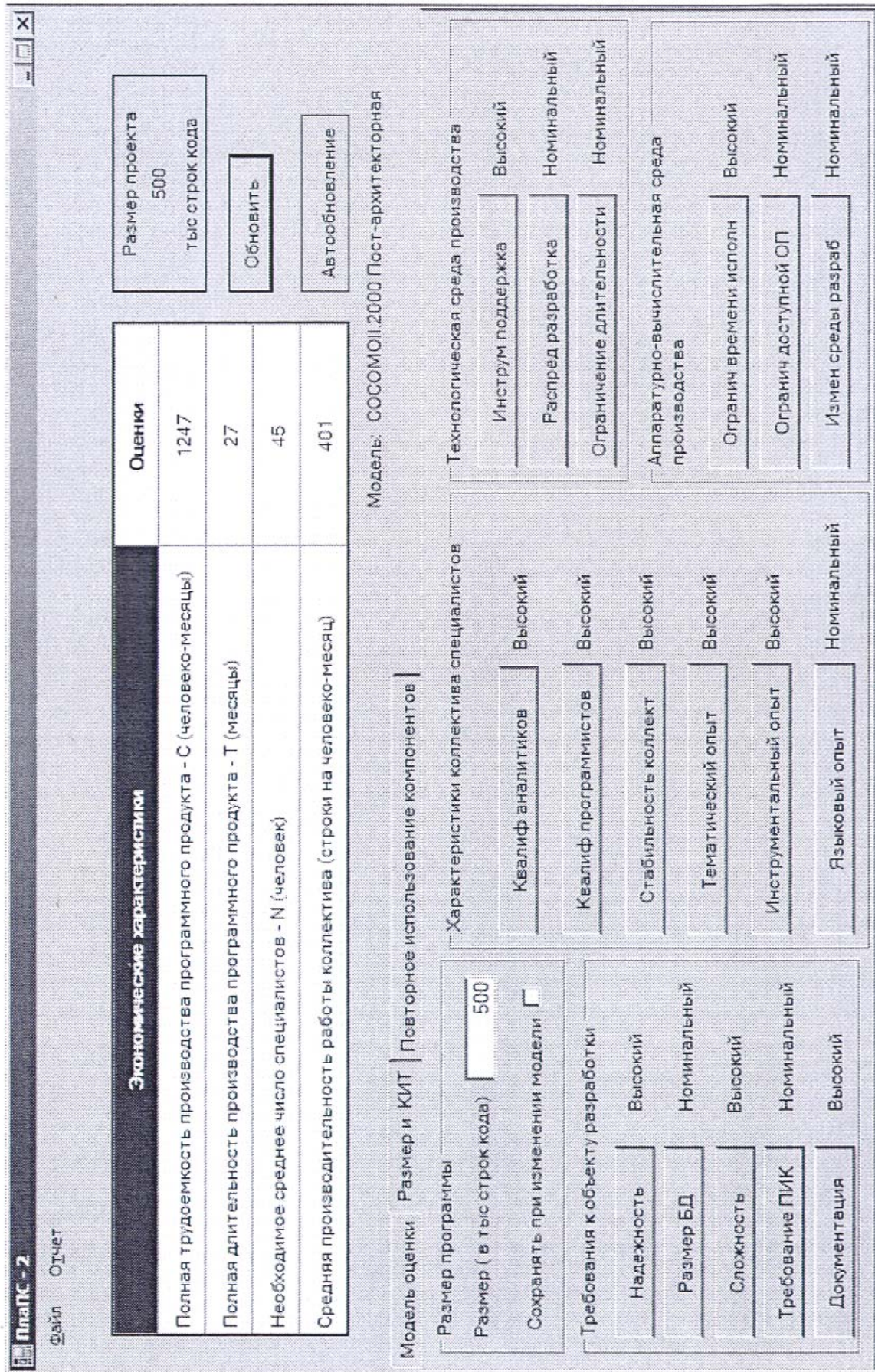


Рис. 9.2

Выбор параметров модели оценки и установка ее факторов производятся с помощью обычных элементов управления: кнопки, текстовые поля, поля со списком и элементы меню. Результаты оценки всех экономических характеристик проекта могут отображаться в структурированной таблице. При выборе элементов меню «Файл» и «Отчет» доступны дополнительные функции: сохранение, открытие файла и создание документа MS Word, содержащего полную информацию о текущем прогнозе экономических характеристик проекта.

Представленные ниже **примеры вариантов** могут использоваться как иллюстрации и контрольные ориентиры при предварительных прогнозах необходимых затрат на крупные программные продукты. Модель СОСОМО II позволяет определять основные прогнозируемые экономические характеристики, а также уточнять некоторые дополнительные данные для **экономического обоснования производства конкретного программного продукта**. Анализ тенденций изменения совокупных затрат и влияния выделенных факторов позволяет избежать грубых ошибок, связанных с нерациональным планированием распределения ресурсов проекта при детальном проектировании и последующем производстве комплекса программ. При этом обычно значения ряда факторов являются фиксированными в силу объективных условий производства на конкретном предприятии. В отдельных случаях для выбора наилучшей стратегии реализации проекта может быть полезным варьирование нескольких вариантов характеристик факторов перед детальным проектированием конкретного комплекса программ. В крайнем случае, на основе выполненных оценок на этапах предварительного или детального проектирования может быть сделано достаточно достоверное **обоснование прекращения разработки** предполагавшегося программного продукта.

Ниже анализируются два **класса программных продуктов** размером 500 KLOC: СРВ – варианты 1 – 3 и АС варианты 4 – 5, которые представлены в таблице 9.10.

Модель СОСОМО II и перечень факторов, представленных в таблице 8.5 – четыре масштабных фактора (кроме управления рисками) и еще семнадцать факторов из этой таблицы, использованы ниже при расчетах экономических характеристик **вариантов двух классов продуктов**.

Таблица 9.10

**Примеры прогнозирования экономических характеристик
производства программных продуктов**

Прогнозы	Варианты				
	1	2	3	4	5
<i>S</i> чел. месяцы	7601	1247	450	2025	409
<i>T</i> месяцы	46	27	21	32	20
<i>N</i> человек	164	45	22	64	20
<i>P</i> строк на месяц	66	401	433	96	477

При прогнозировании экономических характеристик программных *продуктов реального времени* (СРВ) учитывались факторы доступные для оценки на уровне детальных проектов *крупных комплексов программ* размером 500 KLOC: средней новизны и полной согласованности с требованиями. Три представленных варианта (1 – 3), последовательно повышаются по уровню квалификации и слаженности коллектива специалистов и их оснащенности технологией и инструментальными средствами. Кроме того, в варианте 3 учитывается повторное использование программных компонентов.

Вариант 1 отличается тем, что должен быть создан новый программный продукт высокой сложности и надежности, однако коллектив имеет невысокую квалификацию и слаженность, относительно слабо оснащен технологией и инструментарием (2 уровень СММІ). Заказчик не предъявляет специальных требований по ограничению сроков разработки. При надежном функционировании программного продукта в реальном времени, должны эффективно использоваться вычислительные ресурсы реализующей ЭВМ на уровне 90% по оперативной памяти и производительности. В результате экономические характеристики близки к наихудшим по трудоемкости и производительности труда, требуется наибольшее число специалистов (таблица 9.10).

Вариант 2 – предполагается, что должен создаваться почти такой же, как в предыдущем варианте, новый программный продукт высокой сложности и надежности в соответствии требованиями заказчика, компоненты которого допускают повторное использование в аналогичных проектах и их версиях. Разработка продукта предстоит стабильному коллективу специалистов высокой квалификации и слаженности, имеющему значительный опыт создания подобных проектов. Процесс производства поддержан современным инструментарием и технологией на уровне 4 СММІ. Остальные факторы имеют, в основном, номинальные значения. Этот вариант может считаться наилучшим по трудоемкости (меньше в три раза) и производительности труда (больше в шесть раз) и требует более, чем в четыре раза меньшего числа специалистов, по сравнению с предыдущим вариантом.

Вариант 3 в основном подобен по характеристикам факторов варианту 2, но базируется на применении 80% готовых программных компонентов, что отразилось на дополнительном улучшении экономических характеристик. В наибольшей степени это отразилось на снижении трудоемкости и числа специалистов, и относительно мало сократилась длительность производства и производительность труда.

Представленные прогнозы могут служить *ориентирами* при анализе реальных условий производства программных продуктов. Оценки экономических характеристик могут отличаться для СРВ в несколько раз. Уменьшение затрат труда в последние годы объективно обусловлено усовершенствованием технологий и инструментария при производстве программных продуктов. Производительность труда для комплексов программ СРВ при высоком качестве технологии повысилась более чем в два раза до величины свыше четырехсот строк на человеко-месяц, а длительность разработки программных продуктов почти не изменилась (см. таблицу 9.10). Это можно объяснить повышением требований к качеству комплексов программ и необходимостью жесткого соблюдения технологических процессов и документирования.

Таким образом, при экспертных оценках трудоемкости разработки современных новых программных продуктов реального времени квалифицированными командами можно пользоваться *средней производительностью труда около 400 строк на человеко-месяц*. Однако низкий уровень технологии и инструментария, а также недос-

таточная квалификации специалистов, может приводить к повышению затрат более чем в пять раз и к соответствующему снижению производительности труда (ср. варианты 1 и 2). Более подробный учет дополнительных факторов по результатам детального проектирования, хотя и более трудоемок, особенно при подготовке исходных данных, однако позволяет точнее прогнозировать особенности и экономические характеристики производства сложных комплексов программ.

Прогнозирование экономических характеристик для проекта с составом факторов адекватных варианту 2, но при размере комплекса программ *около* 100 KLOC (снижение в 5 раз) показало, что трудоемкость, по сравнению с вариантом 2, сокращается почти в 7 раз, длительность – в 1,6 раза, а число необходимых специалистов почти в четыре раза. При этом производительность труда возрастает почти в 1,4 раза. Таким образом, пятикратное уменьшение размера программного продукта в наименьшей степени (в среднем в полтора раза) влияет на длительность производства и производительность труда, и значительно больше отражается на общей трудоемкости и необходимом числе специалистов.

Прогнозирование экономических характеристик производства комплексов программ *административных систем* (АС) размером 500 KLOC представлено в таблице 9.10 для двух вариантов.

Вариант 4. Административная система применяет большую базу данных. Должен быть создан в соответствии с требованиями заказчика почти новый программный продукт средней сложности и надежности, компоненты которого допускают повторное использование в аналогичных проектах и их версиях. Разработка продукта предстоит коллективу специалистов средней квалификации и слаженности, имеющему небольшой опыт создания подобных проектов. Процесс производства поддержан современным инструментарием и технологией на уровне 3 СММІ. Заказчик не предъявляет специальных требований по ограничению сроков разработки. Для функционирования программного продукта не должны особенно эффективно использоваться вычислительные ресурсы реализующей ЭВМ (см. таблицу 9.10).

Вариант 5 отличается от предыдущего более высоким уровнем технологии (4 СММІ) и квалификации специалистов. Это отразилось на значительном снижении трудоемкости и необходимого числа спе-

циалистов, а также на повышении производительности труда, при относительно небольшом сокращении длительности. В целом, экономические характеристики производства комплексов программ для **административных систем** обычно несколько лучше, чем для СРВ, в основном вследствие меньшей сложности, относительно слабых требований к надежности и качеству продуктов, а также к средней квалификации специалистов.

После предварительных оценок экономических характеристик, в процессе производства целесообразно их сравнивать с реальными доступными ресурсами коллектива специалистов и заданными сроками, и при необходимости производить пересчеты с использованием уточненных и откорректированных значений основных факторов. При **обобщении рассчитанных экономических характеристик** на практике, некоторые значения, могут не удовлетворить специалистов ведущих прогнозирование. Поэтому в методиках прогнозирования должна допускаться **возможность пересчета** получаемых прогнозных значений на основе уточненных значений некоторых факторов.

Например, на базе учтенных при прогнозе факторов, производительность труда может оказаться отличающейся от реальной, характерной для данного предприятия, определявшейся по реализованным предшествующим разработкам. Для того чтобы учесть это различие, должна быть предусмотрена возможность ввода реальной производительности специалистов данного предприятия и пересчета всех остальных экономических характеристик. При этом, индивидуальная производительность труда отдельных специалистов коллектива, может существенно отличаться от «усредненной», заложенной в модель по статистике предыдущих разработок различного размера. Кроме того, на практике могут существовать ограничения реальной численности коллектива, выделяемого для данного проекта комплекса программ, или допустимой (директивной) длительности производства. Для учета таких ограничений должна быть предусмотрена возможность пересчета экономических характеристик при дополнительно откорректированных значениях некоторых факторов.

Достоверность рассчитанных значений основных экономических характеристик целесообразно проверять путем сопоставления с показателями аналогов, созданных на том же предприятии, наиболее близких к прогнозируемому продукту. Полная стои-

мость и длительность разработки обычно подлежат согласованию с заказчиком. В процессе согласования уточняются сценарии проектирования и производства, и возможно изменение не только влияния некоторых факторов, но и требований технического задания на объект и характеристики продукта. Это особенно необходимо, если превышаются допустимые значения стоимости или длительности разработки. Для удостоверения корректности выполненного прогнозирования и исключения грубых ошибок может быть полезным сопоставить полученные результаты с оценками по формулам простейшей модели (8.2) и (8.3) с набором коэффициентов и рейтингов из таблиц 8.2 – 8.4.

Согласованные с заказчиком основные экономические характеристики позволяют фиксировать базовый сценарий и план производства продукта, проводить предварительные расчеты распределений трудоемкости и длительности по этапам производства и по специалистам. По этим данным может проводиться детальное календарное планирование и контролирование работ всего проекта. Для расчета распределения трудоемкости, длительности и числа специалистов по этапам работ, желательно использовать аналогичные характеристики проектов подобного прототипа. Полученные данные могут служить основой последующего календарного планирования проектирования и производства программного продукта. Представленные выше выражения, таблицы факторов и рейтингов целесообразно использовать **как ориентиры и методическую базу для прогнозирования экономических характеристик** проектирования и производства сложных программных продуктов, которые необходимо уточнять и развивать на основе анализа реальных характеристик и опыта выполненных проектов программных продуктов.

Глава 10

ЭКОНОМИКА ПЛАНИРОВАНИЯ ПРОИЗВОДСТВА КОМПОНЕНТОВ И ПРОГРАММНЫХ ПРОДУКТОВ

Основы технологии производства компонентов и программных продуктов

В предыдущих главах экономика проектирования и производства программных продуктов рассматривалась в целом, как сложных изделий, без выделения компонентов и этапов их создания. Ниже внимание сосредоточено преимущественно на экономике технологических процессов, планировании и управлении производством программных компонентов и комплексов, обеспечении и удостоверении их качества. Эта деятельность регламентируется комплексом методов и стандартов, которые являются *компонентами программной инженерии и технологического обеспечения производства программных продуктов*. Их применение предполагает высокую дисциплину, проектировочную и производственную культуру коллективов «команд» специалистов, использование ими методик, стандартов, типовых нормативных документов и средств автоматизации разработки. Они определяют организацию и проведение работ по выполнению технологических операций, направленных на получение готового программного продукта с заданными функциями и качеством в имеющихся организационно-технических условиях – рис. 10.1.

Методической основой технологии производства компонентов и комплексов программ, регламентирующей деятельность специалистов, является *типовой технологический процесс* создания промышленной продукции. Он отражается набором этапов и операций в последовательности их выполнения и взаимосвязи, обеспечивающих упорядоченное ведение работ на всех стадиях от инициирования проекта и подготовки технических требований до завершения испытаний или применения версии программного продукта.

Экономика планирования производства компонентов и программных продуктов включает:

- основы технологии производства компонентов и программных продуктов:
 - определение методологической основы производства программ и типового технологического процесса;
 - подготовка экономически эффективной технологии производства программного продукта;
 - освоение методов описания требований и оценивания характеристик качества производственных процессов;
- декомпозиция требований, структуры, процессов производства комплексов программ и сборка компонентов:
 - цели и способы декомпозиция требований, структуры и процессов производства комплексов программ;
 - потоки взаимодействий между модулями и компонентами разных уровней в комплексах программ;
 - восходящая и нисходящая сборка компонентов и модулей комплексов программ;
- планирование и экономика производства модулей и компонентов комплексов программ:
 - ввод ограничений размеров модулей и компонентов;
 - выбор стратегии программирования и тестирования модулей и компонентов снизу вверх или сверху вниз;
 - установление ответственности специалистов за качество конкретных модулей и компонентов;
- экономика и планирование производства программных продуктов:
 - планирование проектирования и производства программных продуктов с учетом ограничений ресурсов;
 - состав и содержание входных и выходных данных планов производства программных продуктов;
 - подготовка детализирующих и вспомогательных планов по каждому виду и этапу производства;
- применение графиков Ганта для экономического планирования производства программных продуктов:
 - распределение ресурсов на производство программных комплексов по этапам, компонентам и видам работ;
 - календарное планирование использования специалистов при производстве комплексов программ;
 - управление последствиями неопределенностей учета ресурсов в графике с помощью временных резервов;
 - планирование и мониторинг графиков производства посредством использования инструментальных технологических программных продуктов.

Рис. 10.1

Индустриализация технологий создания комплексов программ базируется на стандартизации процессов производства, их структурного построения и интерфейсов с операционной и внешней средой.

Для этого с самого начала разработки должны определяться **задачи, состав и этапы работ**, необходимые для достижения конечной цели, а также требуемые для их выполнения материальные и людские **ресурсы**. Технические и управленческие проверки экономических характеристик, анализ качества результатов промежуточных работ и компонентов, а также корректности их взаимосвязей должны обеспечивать руководителям и всем разработчикам уверенность достижения требуемого конечного результатов производства – **получения продукта требуемого качества**. Для этого необходима **подготовка экономически эффективной технологии** производства программного продукта:

- приобретение или разработка и освоение технологии, структуры, свойств и экономики технологических процессов, состава отчетных документах о компонентах и процессах производства продукта;
- предварительный выбор и освоение среды проектирования, методов, инструментальных средств автоматизации и стандартов производства;
- формирование базы данных производства, концептуальное, логическое и физическое распределение информационных и программных компонентов и документов проекта;
- определение организационной структуры и числа специалистов, а также потребностей в субподрядных организациях для реализации проектов;
- разработка предварительного плана проектирования и производства с учетом возможного технического, экономического и планового рисков;
- подготовка комплекта руководств для детального проектирования, программирования и тестирования компонентов и комплекса программ, согласование словарей понятий и идентификаторов конфигурации продукта.

Характеристики комплексов программ существенно зависят от **качества технологии, стандартов и инструментальных средств**, используемых разработчиками для обеспечения всего жизненного цикла. Оценивание достоинств (**зрелости** – см. главу 9) технологической базы позволяет прогнозировать возможное качество программных продуктов и ориентировать заказчика и пользователей при выборе разработчика и поставщика для определенного проекта с требуе-

мыми характеристиками. Поэтому определение зрелости технологической поддержки производственных процессов, организационного и инструментального обеспечения производства, непосредственно связано с прогнозированием реальных или возможных экономических характеристик и качества конкретного комплекса программ.

В современных автоматизированных технологиях создания и совершенствования сложных комплексов программ, с позиции обеспечения их качества можно выделить *базовые методы и средства*, позволяющие:

- создавать программные модули и функциональные компоненты высокого, гарантированного качества;
- предотвращать дефекты проектирования за счет систем обеспечения качества, эффективных технологий и инструментальных средств автоматизации производства компонентов и комплексов программ;
- предотвращать, обнаруживать и устранять различные дефекты и ошибки проектирования, разработки и сопровождения программ путем верификации и систематического тестирования на всех этапах жизненного цикла комплекса программ;
- удостоверять достигнутые значения качества функционирования компонентов и программного продукта в процессе их испытаний и сертификации перед передачей заказчику и в регулярную эксплуатацию пользователям.

В процессе эксплуатации программного продукта у каждого пользователя могут появляться некоторые претензии к функционированию, которые квалифицируются им как ошибки или дефекты базовой или собственной, адаптированной версии программного продукта. От пользователей или заказчиков могут поступать также предложения по дополнительному внесению изменений в базовую версию для улучшения эксплуатационных характеристик и расширения функциональных возможностей. Аналогичные предложения могут поступать от разработчиков комплекса программ. Для решения таких задач разработаны и активно применяются стандартизированные методы, методики и средства автоматизации регламентированного *сопровождения и управления конфигурацией*. Они позволяют представить отдельным специалистам и руководителям, состояние проекта и его компонентов в любой момент времени и не допускать хаоса при коллективной модификации программ и данных. Дисциплина со-

провождения и конфигурационного управления, а также ее соблюдение, в значительной степени, **определяют экономические характеристики** сложного программного продукта, его качество, длительность применения и конкурентоспособность [7, 22, 36].

Возрастание сложности и ответственности современных задач, решаемых крупными системами, а также возможного ущерба от недостаточного качества программного продукта, значительно повысило актуальность освоения методов **стандартизированного описания требований и оценивания характеристик качества** производственных процессов, компонентов и продуктов на различных этапах жизненного цикла. Выявилась необходимость систематизации реальных характеристик качества, применения стандартов для выбора из них необходимой номенклатуры и требуемых значений для конкретных производств комплексов программ. Обещания разработчиков в контрактах с заказчиками создать высококачественные программные продукты в согласованные сроки **во многих случаях не выполняются**, как вследствие различий в понимании требуемого качества, так и вследствие неумения оценить ресурсы, необходимые для достижения заданного заказчиком качества программного продукта. **Стратегической проблемой** в жизненном цикле современных систем стало планирование производства и обеспечение требуемого качества крупных программных продуктов при реальных ограничениях на использование ресурсов [12, 15, 17, 31].

Цель планирования и определения технологической среды жизненного цикла комплекса программ состоит в том, чтобы установить производственные методы, инструментальные средства, процедуры, языки программирования и аппаратные средства, которые будут использоваться для производства, верификации, управления и подготовки документации программного продукта. План должен включать стандарты, методы предотвращения ошибок и обеспечения отказоустойчивости, которые ограничивают возможность случайного внесения ошибок, и такие методы тестирования, которые гарантируют их обнаружение. Цель методов обеспечения отказоустойчивости состоит в том, чтобы включить в проект такие средства тестирования, обеспечения качества и безопасности применения программного продукта, которые могут гарантировать, что он будет адекватно реагировать на ошибки входных данных, предотвращать выдачу ошибочных данных и контролировать возможность проявления ошибок.

Декомпозиция требований, структуры, процессов производства комплексов программ и сборка компонентов

Декомпозиция требований и процессов производства при создании конкретных компонентов и программных продуктов основывается на разбиении общей цели проекта на несколько промежуточных целей и этапов, каждый из которых также можно разделить. Этот процесс можно повторять до тех пор, пока каждая цель не станет достаточно четкой для ее полного **функционального, экономического представления и оценивания характеристик**, которую руководитель сможет оценить по размеру, сложности выполнения и необходимым экономическим ресурсам [7].

Целью декомпозиции является идентификация всех задач и компонентов производства, которые могут выполняться при реализации проекта. Для крупных комплексов программ первоначально экономические характеристики удастся оценивать в соответствии с требованиями только для основных этапов жизненного цикла и для нескольких сложных функциональных задач.

Декомпозиция функций сложного комплекса программ и выявление особенностей при создании, программировании и тестировании его конкретных компонентов основывается на разбиении общей цели и функций проекта, на несколько промежуточных целей и этапов, каждый из которых также можно разделить. Этот процесс можно повторять до тех пор, пока каждая цель и функция не станет достаточной для ее полного **представления, оценивания и разработки в виде программного модуля или компонента**. Декомпозиция работ обеспечивает руководителей базой для разбиения крупных производственных задач на достаточно обозримые, управляемые части. После того, как последние определены, их можно использовать в качестве **элементов для экономической оценки**, распределения между специалистами с учетом их квалификации, именем ответственного, с соответствующей запланированной продолжительностью выполнения, датой начала и окончания работы. Декомпозицию работ также можно использовать как исходную информацию в процессе календарного планирования проекта. При этом производится упорядочивание технологических процессов для обеспечения своевременного и скоординированного решения всего комплекса производственных задач.

Основной задачей декомпозиции комплекса программ является идентификация всех компонентов, видов и объектов производства, которые могут выполнять **отдельные специалисты или небольшие группы** при реализации проекта. Для сложных комплексов программ экономические характеристики обычно трудно прогнозировать для отдельных компонентов и операций, вследствие разнообразия сложности и размеров компонентов, индивидуальных характеристик создающих их специалистов, и множества других производственных факторов.

Процессы производства сложных комплексов программ основаны на исходных принципах **модульности**. Процессы являются модульными в том смысле, что их исходные компоненты: **невелики, обозримы и взаимосвязаны с другими компонентами**. Сложность и число интерфейсов между процессами и компонентами желательно сводить к минимуму. В принципе каждый процесс предназначен для реализации уникальной функции компонента (модуля) в жизненном цикле комплекса программ, может привлекать и использовать другие процессы и компоненты для выполнения более сложной специализированной функции.

Унификация и структурирование процессов декомпозиции комплексов программ оправданы, если они обеспечивают экономию времени производства и/или применения продукта. **Архитектура процессов и компонентов комплекса программ** – это структура, организующая процессы взаимодействия компонентов, используемых при построении сложных функций, образующих версии программных комплексов. Каждая часть архитектуры обычно включает множество компонентов, даже в относительно простых комплексах. Разработка унифицированной структуры компонентов особенно целесообразна для версий программных комплексов, когда затраты могут эффективно окупаться при производстве множества последовательных вариантов – версий продуктов.

Таким образом, для обеспечения эффективного производства и сокращения затрат целесообразно формулировать и соблюдать ряд принципов и правил структурного построения и повторного применения программных компонентов и модулей (см. рис. 10.1). Эти принципы и правила могут иметь некоторые особенности в различных проблемно-ориентированных областях. Однако их формализация и выполнение в конкретных проектах обеспечивают значитель-

ное **снижение трудоемкости и длительности производства** программных продуктов и их версий. Потеря гибкости архитектуры комплексов программ, некоторое возрастание ресурсов, необходимых для их реализации, обычно полностью компенсируются **улучшением экономических характеристик** процессов проектирования и производства программных продуктов.

Многоуровневое, иерархическое построение сложных программных комплексов позволяет ограничивать и локализовать на каждом уровне сложности, соответствующие ему компоненты. **Нижнему иерархическому уровню** представления программ соответствуют программные и информационные модули (модули данных). Эти компоненты (например, 10 – 100 модулей) объединяются в группы на **среднем уровне компонентов программ** определенного функционального назначения с автономной целевой задачей. Несколько групп функциональных программ образует целостный функциональный **комплекс программ высокого уровня** определенной системы. В сложных случаях возможно создание программного продукта из нескольких взаимодействующих функциональных комплексов программ высокого уровня. Всем иерархическим системам (в частности, программным комплексам) присущ **ряд свойств и требований**, важнейшими из которых являются:

- вертикальная соподчиненность, заключающаяся в последовательном упорядоченном расположении взаимодействующих компонентов, составляющих программный комплекс;
- право вмешательства и приоритетного воздействия компонентов верхнего уровня **сверху вниз** на компоненты нижних уровней;
- взаимозависимость действий **снизу вверх** компонентов верхних уровней от реакций на воздействия и от функционирования компонентов нижних уровней, информация о которых передается верхним уровням.

В результате в иерархических структурах комплексов программ образуются **два потока взаимодействий** между компонентами разных уровней: **сверху вниз** – координирующие и управляющие воздействия верхних уровней на компоненты нижних уровней, и **снизу вверх** – информация о состоянии и реализации предписанных сверху функций компонентами нижних уровней. Координируемые компоненты обычно имеют некоторую автономность поведения и подготовки локальных функциональных решений. Степень автономности

компонентов и интенсивность координирующих воздействий устанавливаются в результате *компромисса* при выделении *экономически выгодного* количества и размера компонентов различных иерархических уровней. Взаимодействие компонентов в пределах уровня целесообразно максимально ограничивать, что позволяет упростить общее координирование компонентов и проводить его преимущественно по вертикали.

Нисходящая – восходящая сборка модулей и программных компонентов занимает важное место при создании сложных программных продуктов. В зависимости от роли в общем технологическом процессе создания комплексов программ различаются два метода и уровня сборки: *нисходящий* и *восходящий*. При нисходящем методе компоненты высокого и последующих уровней разрабатываются и интегрируются *сверху вниз* на основе задачи удовлетворения требований к программному продукту при проектировании, комплексирования компонентов и реализации комплекса программ. При восходящей интеграции для разработки и сборки программного продукта используются преимущественно готовые модули и компоненты нижних уровней, которые последовательно объединяются и формируются *снизу вверх* в программный продукт, соответствующий требованиям.

Нисходящий технологический процесс основан на сочетании пошаговой детализаций технического задания и требований на последовательных этапах проектирования с ее завершением на таком уровне, когда проект может быть описан требованиями к компонентам и модулям. На основании требований технического задания и описания проблемной области создается параметризованное функциональное описание разрабатываемого комплекса программ. При проектировании осуществляется последовательная разработка спецификаций требований к создаваемому комплексу, к его составным компонентам и модулям – рис. 10.2.

При построении комплексов из повторно используемых компонентов (ПИК) может не оказаться элементов, полностью подходящих по функциональным и программным свойствам. В этом случае недостающие модули разрабатываются как ПИК с тем, чтобы конструктивно-технологические требования ко всем составным частям комплекса соответствовали типовой структуре, выработанной для всей проблемной области.

Нисходящая технология сборки компонентов

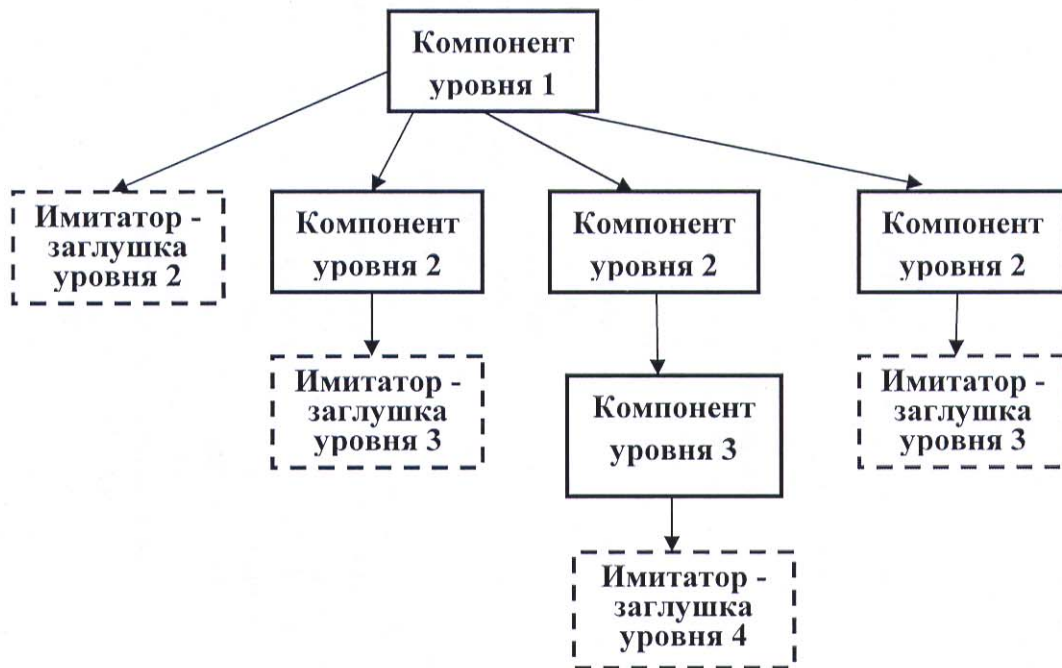


Рис. 10.2

Восходящая технология сборки компонентов

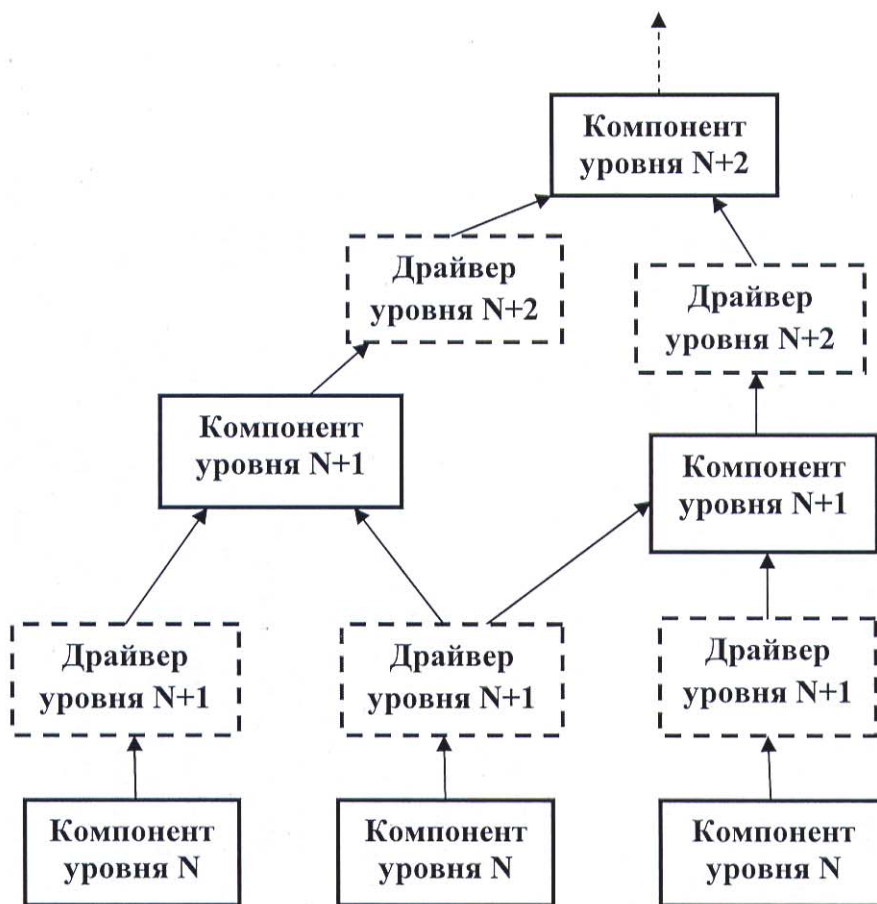


Рис. 10.3

При наличии ограничений осуществляется контроль создаваемых компонентов по функциям и конструктивным показателям (емкость памяти, интерфейсы, время функционирования). По завершении этого этапа в базе данных инструментального комплекса должны содержаться описания (спецификации) всех программных и информационных модулей и компонентов. Эти компоненты должны быть согласованы в функциональном и структурном плане и по конструктивно-технологическим параметрам. В результате может быть подготовлена завершающая сборка комплекса программ.

Нисходящее тестирование – является неотъемлемой частью процесса разработки сложных программных комплексов, при котором сначала разрабатываются и тестируются компоненты верхнего уровня, а затем компоненты, находящиеся на нижних уровнях иерархии. При наличии не полного состава готовых ПИК комплекс можно представить в виде одного абстрактного компонента с упрощенными, **временными функциональными «заглушками»**, которые последовательно, по мере разработки, заменяются на реальные модули и компоненты. Заглушки-имитаторы компонентов должны иметь такой же интерфейс, что и готовые ПИК, но может быть с ограниченной функциональностью.

Восходящий технологический процесс преимущественно применяют в тех случаях, когда разработчики хорошо знают функции, структуру и имеют почти **весь состав готовых модулей и компонентов**. Существует **прототип** создаваемого комплекса, соглашения, принятые, при разработке прототипа, распространяются и на новый программный комплекс. Предполагается функциональное подобие нового комплекса и прототипа, разработка осуществляется итерационным путем. Вначале оценивают состав и качество имеющихся готовых ПИК, из них планируют собрать предварительную версию, выбирают направления и объекты для последовательных доработок. Доработки, как правило, касаются изменения и совершенствования функциональных возможностей, интерфейсов разных видов при смене оборудования или операционной системы, изменений эргономических требований или модификации дисциплины обмена с абонентами сети или вычислительной системы – рис. 10.3.

При восходящем технологическом процессе по завершении испытаний программного продукта полезно осуществлять отбор компонентов, которые могут рассматриваться как повторно используемые, а также ввод их в базу данных испытанных гарантированных ПИК. В качестве новых могут рассматриваться компоненты, полученные в процессе доработки ранее гарантированных ПИК. В этом случае они отличаются от предыдущих версий либо расширенными функциями и областью определения параметров, либо учетом какого-либо нового вида интерфейса или устройства. Замены ПИК, в ранее отлаженных комплексах программ, на новые, доработанные, следует решать с осторожностью, особенно в комплексах, сильно связанных по информации. В результате включение нового ПИК вместо старого может приводить к необходимости повторной комплексной отладки и испытаний всего программного комплекса.

На практике при производстве сложных программных продуктов чаще всего используется *композиция восходящих и нисходящих методов*. Разные сроки разработки для разных компонентов предполагают, что группа, проводящая интеграцию, должна работать с какими-либо версиями готовых компонентов. Поэтому во время процесса сборки приходится разрабатывать как заглушки, так и тестовые драйверы.

Завершающим этапом при нисходящем технологическом процессе является *отбор и наполнение базы данных предприятия новыми высококачественными апробированными компонентами и модулями*. Этот этап важен не только как поставляющий новые ПИК, но и как методическая основа для обучения специалистов современным методам разработки и интеграции комплексов программ. Такая стратегия рассчитана на несколько лет, на которую *должно ориентироваться* предприятие, осуществляющее разработку программных продуктов в некоторой проблемной области. Поэтому требуются значительные начальные затраты на создание нормативно-технической базы, наращивание аппаратной и программной оснащенности, обучение специалистов, с тем чтобы длительно обеспечить существенный рост производительности труда и значительное сокращение сроков производства программных продуктов.

Планирование и экономика производства модулей и компонентов для комплексов программ

Производство модулей и компонентов составляет около половины затрат на полное проектирование и производство сложного программного комплекса. Составление графика их производства, выполняемое менеджером проекта, – необходима для определения интегральных *экономических характеристик производства программных комплексов*. Руководитель должен оценивать длительность этапов создания модулей, компонентов и всего проекта, определять виды и размер ресурсов, необходимых для реализации отдельных этапов и типов работ, и представлять их в виде согласованной последовательности. Если данный проект подобен ранее реализованному, то график производства компонентов нового продукта можно взять за основу. Если проект является инновационным, первоначальные оценки длительности и требуемых ресурсов для модулей и компонентов почти наверняка будут *слишком оптимистичными*, даже если менеджер попытается предусмотреть все возможные неожиданности. С этой точки зрения производство программных комплексов не отличаются от больших инновационных технических проектов, в которых при планировании компонентов также неожиданно возникают проблемы и трудности.

Процесс планирования должен начинаться с определения *проектных ограничений* (временных ограничений, предельных возможностей и числа специалистов, бюджетных ограничений). Эти ограничения должны определяться параллельно с оцениванием проектных параметров, таких как размер и структура проекта, а также при распределении функций среди исполнителей. Затем определяются этапы разработки и то, какие результаты и документы (компоненты, подсистемы или версии программного комплекса) должны быть получены по окончании этих этапов. Далее начинается циклическая часть планирования. Сначала разрабатывается план работ по выполнению проекта или дается разрешение на продолжение использования ранее созданного графика. По мере поступления новой информации о ходе выполнения проекта, возможен пересмотр первоначальных оценок плана и параметров всего проекта. Это, в свою очередь, может привести к изменению графика работ. Главное достоинство любого хорошего плана и прогноза в том, что он сбывается.

Конечно, возможно какое-то отклонение, ошибка на какой-то процент по времени и бюджету, но, обычно половина оценок будет завышенной, а половина заниженной.

Цель состоит в том, чтобы подготовить *реалистичную оценку затрат*, а затем обсудить соответствующие изменения экономики в рамках проекта и компонентов в общем графике производства комплекса программ. Строгое следование правилам декомпозиции компонентов и работ один из способов обеспечить полноту включения всей проектной группы, особенно опытных специалистов, в процесс оценки затрат. Для получения реалистичной оценки необходимо, чтобы она была основана на разумной продолжительности решения отдельных задач.

При организации структуры сложного комплекса программ, создаваемого большим коллективом специалистов, естественно возникает проблема оценки и упорядочивания модулей и компонентов *целесообразных размеров* на разных иерархических уровнях. Рациональные размеры программных модулей могут быть ограничены удобством и обозримостью разработки текстов программ и их тестирования отдельными специалистами. Хотя у некоторых *программистов «виртуозов»* есть тенденция писать монолитные модули размером во многие сотни строк, однако при этом возникают трудности тестирования и обеспечения высокой корректности таких программ. Экспериментально установлено, что во многих программах управления и обработки информации реального времени – линейные участки программ между предикатами – узлами с ветвлением *в среднем* составляют около 5 – 10 строк. Поэтому число маршрутов исполнения программ и соответствующее число тестов, необходимых для их проверки возрастает не пропорционально числу строк в программе, а значительно быстрее (почти квадратично, см. главу 4). Уже при ста строках в программе (10 – 20 предикатов – узлов ветвления) для ее тестирования может потребоваться более 100 тестов. Таким образом, при разработке модулей целесообразно учитывать рациональное *ограничение их размеров на уровне трехсот строк текста*, что соответствует приблизительно тридцати альтернативам в таких программах. При этом для полного покрытия таких модулей тестами необходимо задавать до 1000 условий в тестах, что практически достаточно трудно или невозможно реализовать в ограниченное время. В среднем полное тестирование программ с 30-ю вершинами ветвления

производятся тестами с суммарной сложностью около 300 – 500 узлов – предикатов [1, 26].

Программные компоненты высокой сложности целесообразно делить на более простые и легче тестируемые модули, что во многих случаях делается программистами интуитивно. Анализ числа тестов для большой реальной выборки программных модулей в комплексе программ, создаваемым коллективом специалистов показал, что основная часть модулей содержала около 200 строк (до 20 – 30 узлов ветвления). Это обычно устанавливалось средними специалистами вследствие психологической сложности разработки и тестирования более крупных модулей. Поэтому в ряде предприятий при разработке был **рекомендован рациональный размер программ модулей** в пределах 100 – 300 строк текста, для полного тестирования которых достаточно использовать 10 – 50 тестов с суммарным числом условий ветвления около 100. При превышении рекомендуемых размеров модулей их трудно протестировать и **целесообразно программистам делить на более мелкие компоненты**, доступные для практически полного покрытия тестами, размеры которых регламентировать в инструкциях проекта.

Каждый модуль должен рассматриваться с точки зрения **ответственности за качество определенных личностей – программиста и тестировщика**. Коллектив или личность, выполняющая разработку, несет персональную ответственность за весь данный процесс и компонент, даже если выполнение отдельных задач поручено другому предприятию. Принцип фиксирования ответственности лиц в архитектуре, компонентах и процессах производства программного комплекса облегчает применение стандартов и оценку характеристик частей конкретного проекта, в который может быть вовлечено множество специалистов.

Группа специалистов должна начинать подготовку процесса разработки с получения от менеджера проекта или создания первичного шаблона **плана производства модулей и компонентов**, а затем уточнять план. Для планирования процессов программирования и тестирования компонентов целесообразно формировать график поставки компонентов и модулей для сборки функциональных групп программ. При этом каждый компонент может быть вызывающим другие компоненты, и одновременно вызываемым из других компонентов и зависящим от результатов их функционирования. Эти связи

определяют иерархическую схему взаимодействия между компонентами и модулями комплекса программ по управлению и по информации. Компоненты в сложном комплексе программ обычно наиболее сильно взаимосвязаны в пределах решения крупных функциональных задач. Поэтому целесообразно создавать **два уровня графиков** планирования программирования и тестирования компонентов сложного комплекса программ: на уровне взаимодействия крупных функциональных задач и на уровне взаимодействия компонентов и модулей в составе функциональных задач комплекса программ.

При **разработке снизу вверх** (восходящая сборка) вызываемые компоненты предшествуют вызывающим, которые являются источниками информации для вызывающих, и могут использоваться как генераторы тестов для вызывающих компонентов. На графиках планов следует учитывать эти связи и **распределять во времени поставку компонентов для сборки** и комплексирования с учетом последовательности времени их разработки и иерархии связей. На графике планирования должны отражаться эти связи между моментами завершения разработки и тестирования вызываемого компонента и началом тестирования вызывающего компонента в составе фрагмента комплекса программ. Однако реальные процессы и последовательности программирования и автономного тестирования компонентов не всегда позволяют соблюдать рациональную логику последовательной разработки компонентов снизу вверх с учетом времени подготовки и взаимодействия в комплексе программ. Тогда для предварительного тестирования групп вызывающих компонентов сверху вниз, приходится разрабатывать временные имитаторы-заглушки для тестов, подменяющих вызываемые компоненты (см. рис. 10.2).

Планирование производства компонентов должно быть сконцентрировано на определении и **документировании требований к качеству результатов**, на создании документации для производства, на планировании, необходимом для поддержки среды, и на разработке графика. План должен иметь реальные ограничения по числу и квалификации специалистов, по человеко-часам и по **временному графику реализации** каждого компонента. В плане также следует отражать требуемое качество, предварительные условия и допустимые риски результатов производства компонентов. Сюда включаются все события, действия или обстоятельства, которые могут помешать созданию компонента в запланированный срок. При разработке плана

должно быть учтено выделение функций, разработка которых имеет наибольшее значение для успеха проекта, и функций, разработка которых связана с наибольшим риском. Определение наивысшего риска дает возможность группе тестирования **сосредоточить усилия на функциях и компонентах высокой значимости** для пользователей и достоверности результатов производства комплекса программ.

Группа планирования должна составлять график разработки компонентов, чтобы определить время и ресурсы, на эти работы. **При подготовке графика:**

- должны назначаться специалисты, ответственные за выполнение каждой из работ по программированию и тестированию определенных модулей, компонентов комплекса программ;
- следует учитывать последовательность разработки текстов программ и тестов, их взаимозависимость и связь с процессами и созданием компонентов и функций комплекса программ;
- тестовые процедуры могут быть объединены в группы в соответствии с функциями и связями компонентов и программного комплекса.

План должен **определять, кто из специалистов отвечает** за выполнение конкретного вида работ и компонентов. Также необходимо учитывать **последовательность** выполнения процедур тестирования и их **взаимозависимость**, поскольку при тестировании определенная функция зачастую не может быть выполнена, пока предыдущая функция не сформирует нужные данные. Необходимо планировать работы по адаптации внешней среды, тестированию подготовки обязательных отчетов в графике разработки и выполнения тестовых процедур. График тестирования должен учитывать проверку функций и компонентов, наиболее значимых для программного продукта, и функций повышенного риска. Такие тесты должны выполняться в первую очередь, и график обязан предусматривать достаточно времени для проверки этих функций и, в случае необходимости, для регрессионного тестирования.

Менеджеры должны осуществлять текущий контроль за выполнением эффективной последовательности тестирования модулей, компонентов и комплекса программ, подготавливая как **внутренние отчеты** о развитии каждого процесса, так и **внешние обобщенные отчеты** для руководителя проекта и/или заказчика в соответствии с условиями договора. Все обнаруженные дефекты и результаты их

устранения должны быть документально оформлены, а также в установленные сроки подтверждена полная реализация процессов и выполнение утвержденных планов тестирования. После создания всех запланированных программных компонентов и комплекса, менеджер должен определить степень их **соответствия критериям** качества, установленным в договоре.

Должны быть документированы **квалификация, права и навыки специалистов**, необходимых для корректировки требований и выполнения изменений при производстве определенных компонентов – таблица 10.1. Менеджер должен оценивать разницу между требуемой квалификацией и реальной подготовкой персонала, чтобы определить необходимые направления и объем дополнительного обучения. Планируемое обучение следует отразить в графике, чтобы соответствующие работы по производству конкретного компонента не предшествовали обучению. Также необходимо определять и документировать в плане роли и **ответственность специалистов** за качество определенного компонента с учетом особенности проекта.

Экономика и планирование производства программных продуктов

Управление проектированием и производством программных продуктов регламентировано стандартом **ISO 16326**, где приводятся подробные рекомендации по планированию на основе требований п. 7.1 стандарта **ISO 12207** (см. главу 7). В стандартах детально изложены рекомендации по **планированию и процедуры выполнения процессов управления** на различных этапах производства комплекса программ. Выделенные для проектирования менеджеры должны отвечать за планирование и управление проектом, работами и задачами реализации планов производственных процессов, таких как заказ, поставка, разработка, эксплуатация, сопровождение или вспомогательные процессы. Подготовка и определение области планирования должны начинаться с установления требований к реализуемому процессу и продукту. Менеджер должен **определить экономические возможности реализации** планируемых процессов, проверяя наличие и достаточность ресурсов, выделенных для выполнения и управления процессами (специалистов, технологии и условий), а также ре-

альность сроков завершения производства программного продукта (см. рис. 10.1).

План должен иметь в своем составе **детальные графики, идентифицирующие**: этапы работ; входные, выходные данные и описания решаемых задач; необходимые ресурсы и сроки выполнения; взаимосвязи и зависимости этапов и работ. Должно быть установлено организационно-техническое взаимодействие между различными группами специалистов, которые вносят свой вклад в процессы и обеспечение качества производства, а необходимая информация должна документироваться и регулярно анализироваться. В планах работ поставщиков и субподрядчиков следует четко определить границы ответственности за каждую часть программного продукта и за способ обмена технической информацией и документами между всеми сторонами.

Входные проектные данные и требования к планам производства комплекса программ, включая установленные законодательные и нормативные требования, должны быть оформлены документально, а их выбор проанализирован заказчиком на адекватность. Неполные, двусмысленные или противоречивые требования должны быть предметом урегулирования с лицами, ответственными за их предъявление. Спецификацию требований должен представить потребитель-заказчик. Однако по взаимному согласию ее может подготовить разработчик, в тесном сотрудничестве с потребителями для предупреждений разногласий путем, например, уточнения определенных терминов, объяснения предпосылок и обоснования требований. Спецификация требований к планам может быть представлена и согласована в составе спецификации всей системы. При заключении контракта спецификация требований может быть определена не полностью, она может быть доработана в ходе реализации планов проекта. Если программный продукт нуждается во взаимодействии с другими программными или аппаратными продуктами, то в спецификации требований должны быть оговорены непосредственно или при помощи ссылок на планы и интерфейсы между разрабатываемыми и другими применяемыми продуктами.

Выходные проектные данные при реализации планов должны быть документально оформлены и выражены так, чтобы их можно было проверить и подтвердить соответствие входным проектным требованиям. Выходные проектные данные должны содержать **кри-**

терии приемки продукта заказчиком или ссылки на них, а также идентифицировать те характеристики проекта, которые являются критическими для безопасного и надежного функционирования и применения программного продукта. К составу **выходных проектных данных могут относиться**:

- описание результатов испытаний и применения программного продукта;
- комплект оформленного программного продукта и эксплуатационной документации для пользователей;
- комплект технологической документации для обеспечения возможности модификации и сопровождения версий программного продукта.

Планы могут **регулярно пересматриваться** в процессе реализации проекта. Для внесения изменений в план требуется специальная организация и регламентирующий документ, позволяющие отслеживать эти изменения и отображающие выполнение очередного этапа плана производства программного продукта. Без этой информации нельзя судить о степени готовности создаваемого продукта, невозможно оценить **произведенные затраты** или изменять график работ. При планировании процесса производства программного продукта определяются контрольные отметки, регистрирующие окончание определенного этапа работ. Для каждой контрольной отметки создается **отчет**, который предоставляется руководству производством. Эти отчеты должны подводить итоги окончания отдельного, логически завершенного этапа плана проекта. Обычно при завершении основных больших этапов, заказчику предоставляются результаты и **контрольные проектные компоненты**. Это может быть документация, прототип программного продукта, законченные функциональные подсистемы или крупные компоненты.

При планировании менеджер должен подготовить планы для выполнения производственных процессов, которые должны содержать оценивания **экономики соответствующих работ**, а также обозначения создаваемых программных компонентов и продуктов. В **планах должны быть**:

- установлены графики своевременного решения конкретных задач;
- оценки трудозатрат, времени и ресурсов на их решение;
- распределение задач по исполнителям и их обязанности;

- используемые в процессах производства критерии и результаты управления качеством;
- обеспечение условий и инфраструктура выполнения плановых процессов.

Менеджер должен осуществлять текущий контроль за выполнением планов, подготавливая как внутренние отчеты о развитии каждого процесса, так и внешние обобщенные отчеты для заказчика в соответствии с условиями договора. Все обнаруженные дефекты и результаты их устранения должны быть документально оформлены. После создания всех запланированных программных компонентов и продуктов, менеджер должен определить степень их *соответствия критериям*, установленным в договоре или организационной процедуре. Для этого необходимо [7, 12, 36]:

- обеспечить координацию между планами разработки компонентов и функций для согласованного выполнения завершенных процессов жизненного цикла комплекса программ;
- спланировать действия по разработке компонентов программного комплекса и интегральные процессы жизненного цикла, которые позволят реализовать заданные требования и создать программный продукт требуемого качества;
- выбрать внешнюю среду поддержки и контроля жизненного цикла комплекса программ, включающую методы и инструментальные средства, которые целесообразно использовать для выполнения каждого процесса и обеспечивающие предотвращение дефектов и ошибок;
- выбрать и применять фрагменты стандартов, позволяющие обеспечивать требуемое качество, надежность и безопасность производства и применения программного продукта и системы;
- разработать документацию для планирования и последующей реализации возможных модификаций версий программного продукта.

Следует установить и поддерживать в рабочем состоянии документированные процедуры, гарантирующие производство компонентов и программного продукта в соответствии с заданными требованиями и согласно плану. Такие планы должны описывать эти виды деятельности или содержать ссылки на стандарты и определять ответственность за их осуществление. В планах необходимо определять, каким образом следует управлять проектом, анализировать выполнение работ, а также установить *вид и частоту отчетов* для ру-

ководства проектом, потребителя и других заинтересованных сторон, принимая во внимание все конкретные требования заказчика. В планах должны определяться специалисты и производственные подразделения, которые будут исполнять соответствующие виды работ.

Стандартами **ISO 16326** и **ISO 90003** рекомендуется в процессе планирования производства подготовить и утвердить содержание следующих **основных детализированных планов**:

- производства компонентов и комплекса программ, который должен определять используемую модель жизненного цикла комплекса и его компонентов;
- верификации и тестирования компонентов и комплекса программ, которые определяют методы и средства, способные удовлетворить последовательные цели процессов устранения дефектов и контроля качества программного продукта;
- обеспечения критериев качества компонентов и программного продукта, определяющего методы и средства, при помощи которых будет гарантировано их требуемое качество;
- реализации процессов интеграции компонентов в версии программного продукта;
- обеспечения возможности повторного использования и переноса компонентов и комплекса программ на иные аппаратные и операционные платформы;
- сопровождения и управления конфигурацией версиями программного продукта, которые должны устанавливать методы и средства, при помощи которых будут удовлетворяться цели процесса управления изменениями и корректировками комплекса программ;
- тиражирования, адаптации и внедрения версий программного продукта для конкретных пользователей;
- документирования результатов производства, выпуска технологической и эксплуатационной документации.

Кроме того, в составе перечисленных планов или автономно может быть полезной разработка **вспомогательных планов**:

- обучения и подготовки пользователей для квалифицированной эксплуатации очередных версий программного продукта;
- обслуживания пользователей в процессе эксплуатации программного продукта.

Каждый представленный план должен четко **отражать экономику и ресурсы**, необходимые для его реализации, разделение работ

на этапы и временной график выполнения этих этапов. В таком плане могут присутствовать ссылки на планы других видов. При этом в планах **рекомендуется**:

- определить производственные процессы, которые необходимо выполнить и возможность достижения целей плана в рамках существующих ресурсов и экономических ограничений;
- оценить варианты достижения целей плана и определить, на основе анализа рисков и возможностей, какая целесообразна стратегия производства;
- выявить конкретных лиц и группы специалистов, создающих требуемые результаты при производстве продукта, определить им конкретные зоны ответственности, и обеспечить, чтобы обязанности были поняты и приняты, профинансированы и экономически достижимы;
- обеспечить инструментарий для того, чтобы планы проекта были формально разработаны, реализованы, поддержаны и доступны лицам, вовлеченным в производство;
- регулярно оценивать процессы выполнения планов, принимать меры, для корректировки отклонений от каждого плана и предотвращения повторения проблем, выявленных в процессе производства.

Особого внимания требует **сотрудничество разработчиков с заказчиком**, своевременное предоставление ему нужной информации и решение оперативных вопросов для обеспечения качества программного продукта. Целесообразно планировать и регулярно проводить разработчиком и заказчиком **совместные анализы реализации планов проекта**, либо проводить такие анализы в случае значительных проектных событий, чтобы охватить:

- состояние и развитие выполняемых работ по производству и модификации программного комплекса и компонентов;
- соответствие результатов производства, согласованной спецификации требований заказчика;
- результаты проверок текущего состояния проекта и степени готовности к приемочным испытаниям заказчика;
- состояние подготовки и обучения конечных пользователей разрабатываемого программного продукта и системы.

Применение графиков Ганта для экономического планирования производства программных продуктов

В процессе составления графиков производства сложных программных продуктов весь комплекс работ разбивается на отдельные этапы и оценивается время и затраты ресурсов, требующееся для выполнения каждого этапа. Обычно многие этапы выполняются параллельно, график работ должен это предусматривать и распределять производственные ресурсы между ними оптимальным образом. Нехватка ресурсов для выполнения какого-либо критического этапа – частая причина задержки выполнения всего проекта. Если есть этапы, имеющие большую длительность, их целесообразно разбивать на этапы меньшей длительности. При расчете длительности этапов производства менеджер должен учитывать, что выполнение любого этапа обычно не обходится без больших или малых проблем и задержек. Разработчики могут допускать ошибки или задерживать свою работу, техника может выйти из строя, аппаратные или программные средства поддержки процесса производства могут поступить с опозданием. Если проект инновационный и технически сложный, это становится дополнительным фактором появления непредвиденных проблем и увеличения длительности реализации проекта по сравнению с первоначальными оценками [26, 35].

Кроме временных затрат, менеджер должен рассчитывать *другие экономические и производственные ресурсы*, необходимые для успешного выполнения каждого этапа. Особый вид ресурсов – это «*команда специалистов*», привлеченная к выполнению проекта. Существует эмпирическое правило: оценивать временные затраты так, как будто ничего непредвиденного не может случиться, а затем значительно (на 30 – 50%) увеличивать эти оценки для учета возможных проблем. Трудно прогнозируемые проблемы существенно зависят от типа и параметров проекта, а также от квалификации и опыта членов команды разработчиков. Чаще всего для представления графиков работ используются *диаграммы Ганта* – рис. 10.4.

Графики Ганта – *это методика планирования проектов*, которую можно использовать для достижения нескольких целей, включая календарное планирование производства, финансовое планирование, планирование использования специалистов и других ресурсов.

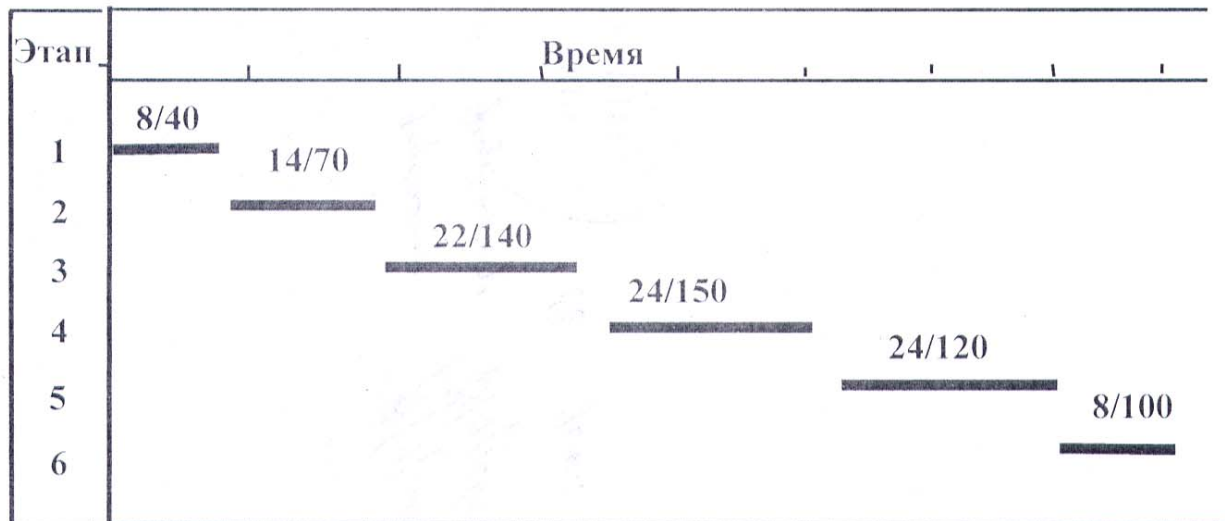


Рис. 10.4

График представляет собой гистограмму, где каждая горизонтальная линейка обозначает отдельный вид или компонент производственной деятельности. Линейки изображаются относительно временной шкалы планирования проекта. Длина каждой линейки пропорциональна продолжительности времени, запланированного на выполнение определенной работы или этапа. График работ *по всему проекту* обычно представляется в виде набора диаграмм, показывающих разбиение производственных работ на основные этапы, зависимости между работами, распределение специалистов и затрат по этапам и работам (см. *пример*, на рис. 10.4 – шесть этапов из таблицы 6.1, в числителе трудоемкость работ; в знаменателе – число специалистов). Для его детализации может использоваться обобщенный перечень этапов и работ при проектировании и производстве сложных программных продуктов реального времени. Визуализация плана в виде *графика Ганта* может быть произведена по укрупненным этапам жизненного цикла и множеству частных работ составляющих эти этапы (например, около 50 работ, из раздела 5 стандарта **ISO 12207**).

В составе и характеристиках работ и этапов можно отражать трудоемкость, длительность и число необходимых специалистов, а также их взаимодействие с предшествующими и последующими работами. *Для каждого этапа*, в свою очередь, может быть составлен детализирующий его частный график Ганта, и оценены его *экономические характеристики*. Необходимые временные, трудовые и другие ресурсы для выполнения и отражения работ на этих графиках

ках, зависят от требований к детализации характеристик конкретного программного продукта и условий его производства.

Если итерационные методы предусматривают и допускают постепенное **уточнение и расширение требований заказчика**, целесообразно назначать некоторую дату, после которой в проект не должны вноситься новые требования. Рекомендуется включать в график резервные периоды времени. Это делается из-за того, что невозможно учесть при планировании влияние всех внешних факторов. К тому же люди часто растягивают решение производственной задачи на все отведенное время. Если при этом в графике не предусмотрено буферного времени, то все временные рамки наползают друг на друга и начинают перекрываться. Одним из вариантов предупреждения возможных неожиданностей является отведение под этапы заведомо большего количества времени. Другой вариант заключается в создании **буферных интервалов в производстве**, во время которых не планируется решение каких-либо специальных задач.

Изменения, порожденные адаптацией такого рода, часто довольно существенны и дают результат, **мало похожий на первичную структуру плана производства**. Структурирование должно использовать детальную информацию о проекте: правила конфигурирования компонентов комплекса, наборы инструкций разработчикам, программистам и тестировщикам, рекомендации на формирование документации и программные интерфейсы компонентов. Кроме того, необходимо выделять ключевых специалистов, для выполнения важнейших шагов процесса интеграции компонентов, отслеживания дефектов или функциональных сборок, фиксации изменений, контроля стиля кодирования и взаимного просмотра кода, а также многие другие детали, специфичные для взаимодействия специалистов и компонентов любого сложного проекта.

График становится более детальным и достоверным по мере продвижения проекта и проведения ревизий. Как только утверждаются требования и архитектура комплекса программ, могут определяться частные производственные задачи и компоненты. В это же время могут более детально рассчитываться трудозатраты на каждый компонент и их взаимодействие, а также кто, когда и над чем будет работать. **Пример фрагмента** такого графика представлен на рис. 10.5.

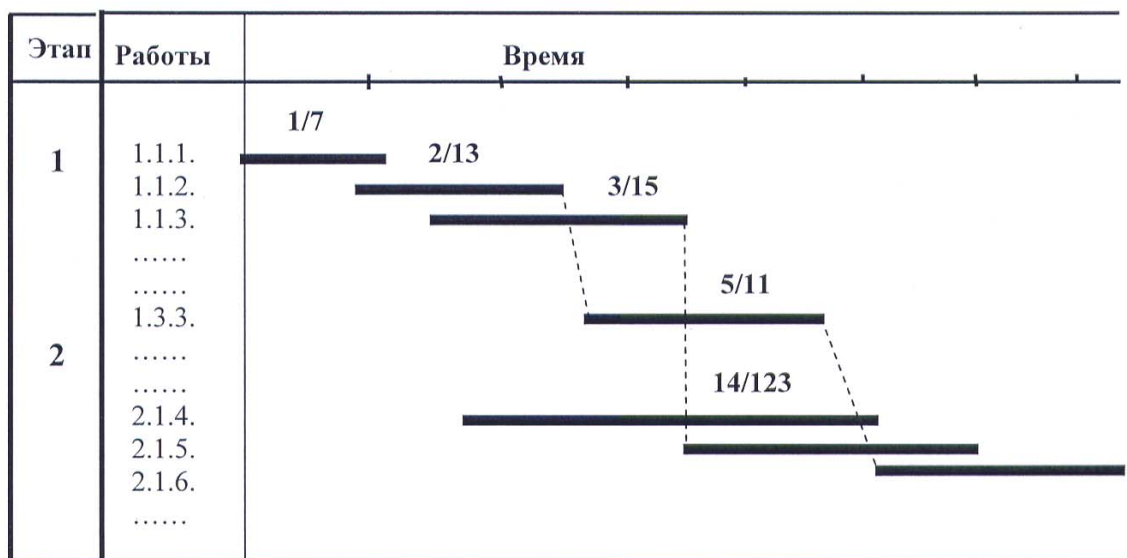


Рис. 10.5

Слева перечислены производственные действия, идентификаторы компонентов (например, компонентов 71 и 92) и их модулей (7.1.1 ... 9.2.6 и т.д.), а справа – указаны горизонтальные полоски, длина которых соответствует длительности выполнения каждого этапа производства компонента в соответствии с временной шкалой. Трудоемкость процессов и число необходимых специалистов может указываться числами над линейкой для каждого модуля или компонента (рис. 10.5 – в числителе – число специалистов, в знаменателе трудоемкость производства). Пунктирные линии отражают связи и зависимости начала некоторых работ только после завершения предшествующей работы. Например, начало работы 9.2.5 возможно только после завершения процесса 7.1.3. Графики Ганта помогают выполнять **календарное планирование производства**. Однако во время выполнения календарного планирования и его реализации могут быть определены новые виды деятельности, не предусмотренные во время первоначального укрупненного планирования. Тогда руководитель проекта должен вернуться назад и пересмотреть структуру декомпозиции работ и календарные планы для того, чтобы найти место и время вновь появившимся дополнительным видам деятельности.

При выполнении больших проектов, содержащих множество действий и отображаемой информации, диаграммы Ганта могут быть очень громоздкими. Они позволяют увидеть **«общую картину» процессов производства**, однако ориентироваться в подобных

графиках сложно. Поэтому полезно структурировать крупные графики на несколько более мелких по функциональным задачам или компонентам проекта, и выделять обобщающий сводный график производства всего программного комплекса без детализации процессов для компонентов (см. рис. 10.4).

График Ганта можно использовать для **распределения ресурсов и планирования кадрового обеспечения**. Если необходимо спланировать работу и результаты конкретных специалистов по срокам, то можно воспользоваться графиком Ганта, на котором каждая полоса будет отражать работы одного из них. На таком графике **специалисты являются ресурсами**, и на нем можно отражать их нагрузку во время производства. Можно пометить соответствующие части полос для обозначения количества времени, которое каждый специалист предположительно должен тратить на выполнение той или иной работы. Несмотря на то, что графики Ганта наглядно показывают каждый вид деятельности, они зачастую не отражают зависимостей между решаемыми задачами и компонентами.

Независимо от того, как **выявляются отклонения от плана**, руководитель – менеджер должен принимать решение об их устранении. Там, где производство ограничивается доступным человеческим трудом, обычно коррекция производится добавлением сотрудников или увеличением времени сверхурочной работы соответствующих специалистов. Однако в программировании недостаток чаще всего ощущается не в рабочей силе как таковой, а в **интеллектуальных ресурсах конкретных специалистов**, поэтому с отклонениями от календарного плана здесь бороться сложнее. Формальное увеличение числа программистов или тестировщиков не поможет «уложиться» в поставленные сроки, настолько же важно учитывать, что помочь в этом могут интеллектуальные способности только некоторых особых специалистов. Таким образом, приемлемым вариантом может стать временный перевод подходящих специалистов на работу над проблемной частью проекта, либо наем специалистов – консультантов для устранения недостатков ресурсов.

Другой способ уложиться в заданные сроки – подробное, критичное **изучение требований к программному продукту и исключение из них не самых актуальных**. Иногда, требования оказываются чересчур высокими, так как делаются попытки придать программному продукту дополнительные «украшения», практически не влияющие на

его функции. Избавление от ненужных требований состоит в «**чистке**» **эталонов – требований**. Важнейшим принципом планирования производства комплексов программ является **упорядочивание приоритетов** в порядке возрастания важности требований для заказчика и/или пользователей. Зачастую можно согласовать с заказчиком и принять решение об исключении некоторых требований для того, чтобы уложиться в сроки. Следовательно, необходимо придерживаться принципа последовательного приращения функций и качества, не только на этапе анализа требований, но и во время планирования производства (в том числе календарного), реализации компонентов и комплекса программ.

Иногда после рассмотрения всех вариантов выхода из цейтнота, подходящее решение не найдено. Тогда следует **признать неадекватности первоначальных планов и графиков**, и провести их пересмотр, исходя из нового понимания сложности поставленных задач, профессиональных способностей имеющихся специалистов и доступности ресурсов. Важно учитывать риски, связанные с прогнозированием графика, и быть готовыми к возможному пересмотру сроков. В менее тяжелых случаях принятие задержки результатов производства продукта может быть единственно правильным решением. Составляя любой рабочий график, следует всегда учитывать **принцип неопределенности**. Спланированный рабочий график – это предположение о будущих действиях разработчиков – программистов, тестировщиков и других специалистов, при оптимальных условиях. Возможными последствиями неопределенностей можно управлять с помощью предусмотренных в графике **временных резервов**, которые являются аналогами экономических и финансовых резервов, применяемых при построении бизнес-планов. Производство программного комплекса обычно должно быть окончено к определенной дате (что интересует руководителей проекта и/или заказчика в первую очередь) **с возможностью некоторой допустимой задержки**, поэтому при объявлении предполагаемой даты завершения создания продукта, полезно оценивать и указывать вероятность успеха.

Для того чтобы **эффективно управлять неопределенностью**, целесообразно использовать в графике начальные оценки длительности производственных этапов. Некоторую неопределенность можно внести в буферы завершающего действия, что при непосредственной работе над проектом позволит управлять неопределенностью

сразу для всех производственных действий. Вероятность своевременного завершения каждого этапа может составлять 50%, поэтому некоторые из них будут завершены позже установленного срока. Все разницы в сроках завершения этапов должны быть вычтены из буфера непредвиденных обстоятельств последнего этапа или всего производства в целом. Однако некоторые этапы могут быть завершены досрочно, поэтому размер суммарных сроков в буфере может оставаться практически неизменным.

Для планирования и мониторинга планов производства, создано несколько *универсальных технологических программных продуктов*. Они обеспечивают представление планов в различной графической форме – *графиков Ганта, сетевых и других графиков*. Большинство из них не имеют проблемного ориентирования, и требуют перед применением подготовки и наполнения в них: состава, содержания и характеристик этапов, компонентов или работ, значений времени и трудоемкости, необходимых ресурсов и состава конкретных специалистов для реализации графиков.

В *качестве примера*, для представления, изменения и управления *сложными экономическими планами производства* в виде графиков Ганта может рассматриваться достаточно простой и удобный программный продукт **Microsoft Project**. Основные функции пакета отражены *четырьмя группами процедур*: задачи проекта; ресурсы проекта; отслеживание реализации; выполнение изменений проекта и отчеты. Эти группы позволяют создавать, манипулировать планами, конкретными работами и специалистами с учетом их взаимосвязей, доступных ресурсов, локальных и общих целей планирования. Для крупных проектов возможно выделение определенных этапов, компонентов и групп работ для построения более детальных графиков Ганта. Исходные данные о характеристиках компонентов проекта, необходимые для составления и описания плана, должны быть представлены в группе *Задачи компонентов и комплекса программ*: определение проекта; определение рабочего времени задач и компонентов; организация этапов и последовательности решения задач; установка крайних сроков и ограничений; определение допустимых рисков проекта при размещении задач.

Ресурсы проекта: выбор состава и квалификации специалистов; определение рабочих часов использования ресурсов для конкретных

задач; назначение специалистов и оборудования на решение задач; публикация сводных данных о доступных ресурсах производства.

Отслеживание и изменение графика производства: создание и сохранение первичного базового плана; организационная и техническая подготовка для отслеживания хода работ; контроль и изменение директивных сроков выполнения компонентов проекта; внесение изменений в компоненты, ресурсы и риски проекта; публикация данных о выполненных компонентах экономических планов, частей и всего графика проекта.

Отчеты о результатах проекта и его графиках: отображение текущего состояния проекта; анализ и контроль критических задач экономики производства; контроль, сокращение или устранение рисков проекта; контроль выделения и использования времени и ресурсов; контроль и анализ экономических характеристик проекта; публикация сводных данных о реализации и изменениях графиков производства.

Графики Ганта позволяют динамически **планировать экономику производства программных комплексов**. Такое итерационное планирование целесообразно проводить, последовательно детализируя содержание производственных процессов с учетом размера, функций и характеристик компонентов конкретного комплекса программ в виде **совокупности графиков**:

- программирования и тестирования модулей и компонентов каждой выделенной функциональной задачи;
- каждой крупной функциональной задачи и взаимодействия ее модулей и компонентов в комплексе программ;
- этапов, производственных процессов и ресурсов всего жизненного цикла комплекса программ.

Суммарные затраты ресурсов, уточняемые при их оценках по этапам, компонентам и по перечням работ, **не должны выходить за пределы**, ограниченные договором. Если эти условия не выполняются, то **итерационно следует**: пересматривать и изменять или ресурсы, или графики и распределение ресурсов по этапам и компонентам производства, или сокращать состав и содержание работ на производственных этапах компонентов. Во всех случаях следует контролировать реализацию утвержденных требований к функциям, характеристикам и экономике программного продукта.

На основе значений длительности программирования и тестирования модулей и компонентов и взаимосвязи между ними может строиться *сетевой график* декомпозиции и последовательности их реализации (см. рис. 10.2). На этом графике должно быть видно, какие компоненты могут программироваться и тестироваться параллельно, а какие взаимозависимы и должны выполняться последовательно друг за другом. Тестирование ряда определенных компонентов не может начаться, пока не выполнены все компоненты на всех путях, ведущих от начала тестирования компонентов комплекса к данному компоненту.

Минимальное время выполнения всего тестирования комплекса можно рассчитать, просуммировав в сетевой диаграмме длительности тестирования *на самом длинном маршруте* от начала тестирования группы компонентов до его окончания (это так называемый *критический путь*). Таким образом, общая продолжительность производства и тестирования компонентов проекта зависит от компонентов, находящихся на *критическом пути*. Любая задержка в завершении производства любого компонента на критическом пути приведет к задержке всего проекта. Задержка в завершении производства и тестирования компонентов, не входящих в критический путь, не влияет на продолжительность создания всего комплекса программ до тех пор, пока суммарная длительность тестирования этих компонентов (с учетом задержек на каком-нибудь пути) не превысит продолжительности работ на критическом маршруте. Важной частью работы менеджера проекта является *оценка рисков*, которые могут повлиять на график работ или на качество создаваемого *программного комплекса*, и разработка мероприятий по предотвращению или сокращению рисков [20, 31].

В процессах предварительного и/или детального проектирования рекомендуется переходить к регистрации и анализу *фактических экономических затрат* на производство комплексов программ и управлять реализацией проектов с учетом ограниченности использованных и доступных ресурсов и требуемого качества продукта. Мониторинг и варьирование экономических параметров проекта может положительно влиять на сокращение общих затрат ресурсов на конкретные этапы и работы и обеспечивать накопление значений экономических характеристик реализованных проектов и их компонентов для использования их в качестве прототипов.

Глава 11

ЭКОНОМИКА ОБЕСПЕЧЕНИЯ И УДОСТОВЕРЕНИЯ КАЧЕСТВА ПРИ ПРОИЗВОДСТВЕ СЛОЖНЫХ ПРОГРАММНЫХ ПРОДУКТОВ

Организация, экономика и управление качеством производства программных продуктов

Основу управления качеством сложных программных систем определяют *две группы нормативных документов*: стандарты менеджмента качества процессов жизненного цикла систем – **СММІ: 2003**, и стандарты менеджмента (административного управления) системой качества – **ISO 9000:2000** [8, 17, 29, 47]. Эти стандарты имеют много общего и трудно выделить их преимущества, поэтому при реальном производстве сложных программных продуктов целесообразно уделять приоритет одной из групп стандартов в зависимости от особенностей конкретного проекта и предшествовавшего опыта специалистов предприятия. Однако не все требования и процессы стандартов **ISO 9000** целесообразны для непосредственного применения при производстве программных продуктов, что учтено в их расширении – руководстве **ISO 90003**. Кроме того, при описании ряда процессов управления проектами для их уточнения и конкретизации в нем делаются ссылки на основные стандарты, регламентирующие жизненный цикл комплексов программ: **ISO 12207**, их характеристики качества **ISO 9126**, **ISO 25000** и другие, представленные в Приложении. Ряд работ, особенно на наиболее творческих этапах создания программного комплекса, не регламентируется стандартами. Поэтому иногда целесообразно дополнительно определять такие работы нормативными документами и спецификациями разработчиков системы или ведомственными нормативными документами. Для оценивания и учета влияния отдельных факторов производства на качество и экономические характеристики программных продуктов целесооб-

разно использовать **как ориентиры** таблицы рейтингов факторов 8.6 – 8.7 и 9.1 – 9.9.

Достижение высоких значений качества программных продуктов существенно зависит от **зрелости технологии производства** и **инструментальных средств**, используемых для обеспечения их жизненного цикла (см. главу 9 и рис. 9.1). Оценивание зрелости технологической базы позволяет прогнозировать возможное качество программного продукта и ориентировать заказчика и пользователей при выборе проектировщика и/или поставщика для определенного проекта с требуемыми характеристиками. Поэтому определение уровня зрелости технологической поддержки процессов жизненного цикла, организационного и инструментального обеспечения производства, непосредственно связано с достижением реальных или возможных характеристик качества конкретного комплекса программ – рис. 11.1.

В стандарте **ISO 12207** (см. главу 7) определен **общий процесс управления качеством программных проектов**, который может быть применен любой стороной, управляющей производством комплексов программ. Некоторые проектные или производственные процессы, работы и задачи могут выполняться одновременно, при этом они могут быть взаимосвязаны или координированы в организационных структурах по классификации работ в зависимости от этапов жизненного цикла программного комплекса. Менеджер, руководитель отвечает за управление производством продукта, и задачами соответствующих процессов, такими как заказ, поставка, разработка, эксплуатация, сопровождение или вспомогательные процессы.

Руководитель – управляющий программным проектом **должен гарантировать поставку продуктов** в соответствии с условиями договора, включать в производство только те задания и функции стандартов, которые обеспечивают успешную реализацию требований заказчика и создание соответствующего продукта. Процесс управления производством должен начинаться **с установления требований** к каждому реализуемому процессу, компоненту и их качеству. После установления требований руководитель должен определить возможности их реализации, проверяя наличие, соответствие и применимость ресурсов, выделенных для выполнения и управления процессом производства (специалистов, технологии и условий среды), а также реальность установленных сроков производства продукта.

- Экономика обеспечения и удостоверения качества при производстве сложных программных продуктов включает:**
- организацию, экономику и управление качеством производства программных продуктов:
 - определение зрелости технологической поддержки процессов жизненного цикла комплекса программ;
 - определение требований к характеристикам качества производства конкретного программного продукта;
 - обеспечение требуемого качества и экономически эффективного производства программного продукта;
 - изменение и поэтапное управление характеристиками качества производства программного продукта;
 - понятия и свойства дефектов и ошибок в сложных комплексах программ:
 - отсутствие полностью определенной программы-эталона, которой должен соответствовать программный продукт;
 - свойства первичных и вторичных ошибок и дефектов программ;
 - влияние анализа дефектов и ошибок на формирование профессиональной квалификации специалистов;
 - организацию, экономику и управление процессами испытаний программных продуктов:
 - план испытаний программного продукта в составе системы на соответствие требованиям;
 - Программу и методики испытаний сложных программных продуктов;
 - критерии завершения приемо-сдаточных испытаний программных продуктов на соответствие требованиям;
 - оценивание экономики, достигнутого качества тестирования и возможности применения комплекса программ по назначению;
 - утверждение комплекта документов, содержащих результаты и акт испытаний версии программного продукта;
 - стандартизацию и сертификацию для обеспечения и удостоверения качества производства программных продуктов:
 - выбор и адаптацию профилей стандартов для обеспечения проектирования и производства программного продукта;
 - сертификацию конечных объектов производства – программных продуктов;
 - сертификацию технологий и систем качества, обеспечивающих создание и совершенствование программных продуктов.

Рис. 11.1

При необходимости и по согласованию со всеми заинтересованными сторонами, требования к программному продукту могут быть изменены для удовлетворения критериям успешного завершения производства.

Подготовка и **определение области управления** заключается в установлении выполнимости процессов производства при наличии, достаточности и соответствии квалификации специалистов, среды программной инженерии и технологий, **необходимых для выполнения проекта и управления им**, а также для определения обоснованных сроков его реализации (см. главу 10). Работа по подготовке и определению области управления (характеристик, экономики, качества проектируемого продукта и методов их оценивания) может быть затруднена, если новый проект не имеет аналогов и отсутствует соответствующий опыт в данном предприятии. Для уникального, не имеющего аналогов проекта должны быть предприняты специальные мероприятия и анализы, обеспечивающие определение области управления качеством и соответствующий надзор за реализацией проекта. Мероприятия должны сопровождаться анализами, оцениваниями опыта аналогичных проектов и экспертными заключениями.

Должны быть предусмотрены специальные мероприятия для управления изменениями требований при реализации жизненного цикла программного комплекса. Все изменения, вносимые в процессы управления и требования, должны быть оценены по их влиянию на экономические характеристики, стоимость, графики работ, риски и эффективность реализации заданного качества продукта. Следует предусматривать специальные мероприятия по реализации, определению и документированию возможных изменений **характеристик качества продукта**. Когда программный продукт встраивается в систему более высокого уровня, некоторые функции распределяются между программными и аппаратными средствами или между данным программным продуктом и другими, внешне взаимосвязанными с ним. При этом важно установить **критерии успешного завершения производства**. Следует учитывать, что основная цель производства и качества продукта может быть не достигнута при выборе варианта его выполнения в минимальные сроки или с наименьшими затратами, например для систем, критичных по безопасности, когда требуется наиболее глубокое тестирование. Должны быть детально определены требования по тиражированию, распространению, вводу в действие

(инсталляции) и тестированию программного комплекса, установлена и отслежена взаимосвязь (трассировка) между требованиями к системе и программному продукту, требованиями к программному продукту и к полноте тестирования.

Руководитель должен реализовать план, **выполняя управление процессами производства**, чтобы удовлетворить поставленным целям и требованиям к качеству программного продукта. Он должен осуществлять текущий надзор за выполнением процессов специалистами, подготавливая как внутренние отчеты о развитии производства, так и внешние отчеты для заказчика в соответствии с условиями договора, исследовать, анализировать и решать проблемы, обнаруженные при выполнении процессов производства. Решение проблем может приводить к изменениям планов. Его обязанностью является обеспечение того, чтобы влияние любых **изменений на процессы производства, были выявлены, управляемы и контролируемы**. Все обнаруженные проблемы и их решения должны быть документально оформлены.

В установленные сроки руководитель должен **отчитываться о реализации, качестве процессов и результатов производства**, подтверждая выполнение утвержденных планов и преодолевая возникающие затруднения. При реализации контрольной части данной работы менеджер программного проекта должен отвечать за надзор, обеспечивающий обнаружение и анализ любых отклонений от запланированного выполнения конкретного контролируемого процесса производства, и внесение соответствующих корректировок в ход процессов. Инструментарием надзора и контроля за выполнением процессов являются измерения характеристик качества программного продукта, контроль вносимых изменений, а также оценки и аудиторские проверки производственных процессов и продуктов. **Измерения экономических характеристик производства программного продукта** должны быть использованы для поэтапной проверки соответствия между ожидавшимися затратами, функциями и характеристиками программного продукта и их реализацией.

Руководитель программного проекта должен **отвечать за соблюдение коммуникационных требований в коллективе специалистов**, включая своевременность представления отчетов заказчику и заинтересованным лицам, распространение планов проверок и выдачу заданий, а при необходимости, отвечать за нарушения сроков, от-

четности и документирования. Для обеспечения взаимодействий между специалистами полезно иметь централизованную, обновляемую базу данных регистрации конфигурации процессов и компонентов реализации проекта. Оно должно обеспечивать экономически эффективное выполнение программных проектов в установленные сроки согласно требованиям договора и возможно совместно с заказчиком определять промежуточные цели проекта. При **организации управления качеством производства** программного продукта рекомендуется [7, 31, 35]:

- использовать макетирование для разработки части функций программного продукта с целью продемонстрировать заказчику реализуемость его функциональных возможностей в целом;
- проводить работы по проектированию критических систем при наличии у специалистов достаточных тематических знаний в области соответствующей программной инженерии;
- при реализации проекта совместно с заказчиком проводить анализы приоритетов требований к программному продукту, обеспечивающие соответствие целям проекта или их корректировку (по стоимости, срокам и эффективности);
- численность специалистов и количество рабочих групп (команд) должны быть увязаны с объемами финансирования и графиком производства продукта и не должны превышать установленные пределы;
- поэтапно оценивать качество результатов производства программного продукта, освоить методы прогнозирования экономической эффективности производства, позволяющие своевременно обнаруживать нарушения ограничений стоимости и/или графика работ;
- необходимы постоянные контакты с заказчиком во избежание неожиданных изменений в функциональных требованиях, стоимости, графике работ и экономической эффективности проекта.

Руководитель должен **обеспечить поэтапную оценку качества производства программных продуктов, затрат и реализации планов** на соответствие установленным требованиям, участвовать в решающих анализах, для отслеживания процессов и работ по программному проекту. Необходимыми элементами успешной реализации производства являются периодические анализы и оценки состояния выполнения и завершения заданий, соответствие прогнозируемых и реальных функций программного продукта. Ход работ должен быть

определен по фактическим измерениям размера и качества компонентов продукта, объема выполненных работ, их стоимости, реализации графика работ, наличия готовых или разработки новых компонентов для программного комплекса.

Оценивание хода выполнения заданий рекомендуется проводить при участии специалистов, знающих требования к программному продукту, используемые технологии, создаваемые компоненты продукта, технические требования к ним, соответствующие процессы и инфраструктуру производства. Проверка выполнения графика работ, особенно в части создания компонентов программных продуктов, должна быть структурирована с целью получать реальную оценку состояния результатов производства. Необходимо документально оформлять наиболее существенные и оперативные проблемы, рассмотренные в ходе проверок и оценок, и решения, принятые по их результатам. Наиболее существенные и оперативные вопросы должны быть решены, а обнаруженные при этом проблемы введены в систему деятельности по **корректировке производства и продукта**.

После создания всего программного продукта и выполнения всех производственных работ и задач, руководитель должен **определить степень их соответствия требованиям и критериям качества, установленным в договоре**. Он должен проконтролировать результаты и полноту документации созданных программных продуктов и выполненных работ, например, права собственности, гарантии, интеллектуальные права, патенты и авторские права. Все представленные окончательные результаты и соответствующая документация должны быть сохранены в архиве в соответствии с условиями договора. Управление должно обеспечивать **оценку завершенности проекта и гарантировать**, что он удовлетворяет критериям качества и требованиям заказчика. Подобные проверки могут выполняться по мере успешного завершения производства компонентов программного продукта, реализации определенных процессов или функциональных задач. Результаты и отчетные материалы по программным компонентам, продуктам или задачам должны быть проверены на полноту и, если они признаны полными, – сохранены в архиве в соответствии с договорными требованиями проекта и/или инструкциями, принятыми в предприятии.

На этапе *завершения производства программного продукта* следует использовать полученные ранее результаты тестирования с соответствующими данными и отчетными материалами:

- результаты приемочных испытаний, тестирования, верификации, аттестации и измерений качества компонентов и программного продукта;
- отчеты о достижении требуемых характеристик качества программного продукта и результаты опытного эксплуатационного тестирования;
- отчеты о дефектах и ошибках, устраненных в программном продукте;
- результаты аудиторских проверок процессов и продуктов производства;
- результаты приемки и регистрации окончания запланированных производственных работ;
- замечания заказчика и результаты взаимодействия с ним.

По окончании производственных работ заказчиком должны быть выданы *предложения по сопровождению и развитию версии программного продукта* и другие документы, по их архивированию, размещению носителей и длительности хранения полной информации о производстве и качестве программного продукта.

Понятия и свойства дефектов и ошибок в сложных комплексах программ

При любых видах деятельности людям любой квалификации свойственно *допускать дефекты и ошибки*, которые отражаются на качестве проектирования и производства комплексов программ. Это всегда необходимо учитывать заказчикам, руководителям и специалистам при анализе экономики создания таких изделий как сложные программные продукты. *Понятие ошибки в программе* – подразумевает неправильность, погрешность или неумышленное искажение компонента, процесса или их характеристик качества, что может быть *причиной ущерба – риска* при функционировании и применении программного продукта. При этом предполагается, что *известно или можно предположить* – правильное, эталонное состояние объекта или процесса, по отношению к которому может быть выявлено наличие искажения – дефекта или ошибки.

Исходными эталонами для любого программного продукта являются спецификации требований заказчика или потенциального пользователя, предъявляемые к программам. Подобные документы устанавливают состав, содержание и значения результатов, которые должен получать пользователь или внешняя система при определенных условиях и исходных данных. Любое отклонение результатов функционирования программы от предъявляемых к ней требований и сформированных по ним эталонов – тестов, следует квалифицировать как **дефект – ошибку в программе**, наносящий некоторый, возможно экономический, ущерб. Различия между требуемыми и полученными результатами функционирования программ могут быть следствием ошибок не только в созданных программах, но и в первичных требованиях спецификаций и в документах, явившихся базой при создании эталонов – тестов. Тем самым проявляется объективная реальность, заключающаяся в **невозможности абсолютной корректности** и полноты исходных требований спецификаций и эталонов для сложных программных продуктов. Чтобы делать программы **высокого качества**, желательно учитывать психологию и индивидуальные особенности, квалификацию и опыт людей – специалистов, и понимать, почему они делают определенные ошибки. Только тогда квалифицированными специалистами можно применять превентивные меры, создавать методы и средства, которые помогают предотвращать или облегчать обнаружение и исправление ошибок (см. таблицу 7.1) [1, 11, 34]. Чтобы понять, что привело к ошибке, желательно знать: какими были действия специалиста при выполнении ошибочных процессов. В определении понятия ошибки **решающее значение имеет время действий**:

- **прошлое** – ошибочное действие уже совершено, оно всегда в прошлом, с ошибкой всегда имеют дело как со свершившимся фактом;
- **будущее** – профилактические меры направлены в будущее, любое воспитание и обучение направлено на то, чтобы последующие действия позволяли их выявлять и не повторять предыдущих ошибок.

Оценки качества программных продуктов при наличии дефектов и ошибок могут проводиться с двух позиций: с **позиции положительной** эффективности и непосредственной адекватности их характеристик назначению, целям создания, требованиям заказчика, а также с **негативной позиции** возможного при этом ущерба – риска от использования программного продукта или системы с ошибками [20,

43, 45]. В процессе жизненного цикла комплекса программ его требования подвергаются декомпозиции на спецификации программных и информационных компонентов и модулей. Эти спецификации рассматриваются как частные эталоны для составных частей комплекса, однако они редко бывают абсолютно полными и корректными. В процессе декомпозиции и верификации исходных требований, возможно *появление ошибок* в спецификациях на группы программ и на отдельные компоненты. Это способствует расширению спектра возможных дефектов и вызывает необходимость создания *гаммы методов и средств тестирования* для выявления некорректностей в спецификациях и компонентах разных уровней детализации комплексов программ.

При проектировании и производстве систем стараются *оценить вероятность ошибочных действий* специалистов. Ошибки возникают в наиболее сложных пунктах трудового процесса, там, где велика его нагрузка и сложность. Все эти возможности определяются количеством обрабатываемой информации, временными характеристиками обработки, состоянием человека. Изучение труда профессионалов показывает, что он идет большей частью автоматически, а сознательное планирование и исполнение, профессионал ведет только в наиболее сложных участках. Именно там вероятнее всего ошибки и ошибочные действия. Чем сложнее задача, тем с большей вероятностью специалист оказывается на грани, где переход от правильного действия к ошибочному, становится почти неуловимым, пока действие не закончилось.

Ошибка – это факт, случай из практики проектирования и производства компонентов и комплекса программ. Концепция ошибки строится на основе представлений о нормальном функционировании системы, которое определяется позитивно. Для расследования причин ошибок необходимы соответствующие (в том числе и психологические) методики. Представление о *системе психологических процессов* может быть основой для методики анализа типов ошибок в системе [1, 24, 34]:

- *ошибки восприятия* – не успел обнаружить, не сумел различить, не узнал;
- *памяти* – забыл, не успел запомнить, не сумел удержать в памяти, сохранить, восстановить, воспроизвести;
- *мышления* – не понял, не предусмотрел, не разобрался, не

проанализировал, не обобщил, не сопоставил;

- **внимания** – не сумел сосредоточиться, собраться, переключиться, удержаться, устал.

Расследования причины ошибки, имеют дело с человеком, совершившим действие. Для ее локализации реконструируют ситуацию исполнения действий, совершенных в прошлом. От него стремятся получить сведения о совершенных действиях, а он, вольно или невольно, скрывает то, что происходило тогда, когда он совершал ошибочные действия. Особенно велико **влияние психологического анализа ошибок на формирование профессионального опыта** специалистов и совершенствование их действий. Путь к профессиональному, экономически эффективному мастерству специалистов, лежит через выявление и преодоление ошибок в системах.

После достижения определенного профессионального уровня интенсивность развития квалификации специалиста сокращается – он достигает своего потолка – **уровня некомпетентности** [24, 34]. Если сложность задач выше этого потолка – специалист не может действовать так же успешно, как при решении более обычных, простых задач. Высший профессионализм предполагает высокую требовательность специалиста к себе, критическое отношение к совершенным ошибкам, усиленный контроль за действиями после допущенной ошибки.

Опыт – это не только высокое качество действий, но и пристальный контроль за их исполнением. Людям свойственно не признаваться в ошибках, либо эти признания бывают поверхностными. Дело в том, что понятие **ошибка** имеет особый психологический характер, **оно чуждо субъекту**. Но поскольку само понятие ошибки является неоднозначным; поскольку велика роль случайности, стихии; ошибка предполагает сопоставление результата и процесса, который привел к ошибочному результату; поскольку ошибочный результат был достигнут вопреки стремлениям субъекта, которые были направлены на положительный результат – **субъект сопротивляется оцениванию его ошибки**.

Важной особенностью дефектов и ошибок в комплексах программ обычно является **отсутствие полностью определенной программы – эталона**, которой должны соответствовать текст и результаты функционирования программного продукта. Поэтому установить наличие и локализовать дефекты непосредственным сравнением с программой без ошибок в большинстве случаев невозможно. При

тестировании специалистами обычно сначала обнаруживаются **вторичные** ошибки и/или **риски**, т.е. последствия и результаты проявления некоторых внутренних дефектов или некорректностей программ. Эти внутренние **дефекты** следует квалифицировать как **первичные** ошибки или причины обнаруженных аномалий результатов. Последующая локализация и корректировка таких первичных ошибок должна приводить к устранению ошибок, первоначально обнаруживаемых в результатах функционирования программ.

Потери эффективности и качества программ за счет их неполной корректности в первом приближении можно считать прямо пропорциональными (с коэффициентом) проявлениям вторичных ошибок в выходных результатах. Типичным является случай, когда одинаковые по величине и виду вторичные ошибки в различных результирующих данных существенно различаются по своему воздействию на общую экономическую эффективность применения программного продукта. Таким образом, оценка последствий, отражающихся на вторичных ошибках и функционировании программ, может, в принципе, производиться **по значениям экономического ущерба – риска вследствие не устраненных их причин – первичных ошибок в программе**. Вторичные ошибки являются определяющими для эффективности функционирования программ, так как не каждая первичная ошибка вносит заметный вклад в выходные результаты. Вследствие этого ряд первичных ошибок может оставаться не обнаруженным и, по существу, не влияет на функциональные характеристики программного продукта.

Наибольшее число первичных ошибок вносится на этапах системного анализа, проектирования, производства или модификаций программ. При этом на долю системного анализа приходится наиболее сложные для обнаружения и устранения дефекты. Интенсивность проявления и обнаружения вторичных ошибок наиболее велика на этапе активного тестирования и автономной отладки программных компонентов. Затем в процессе производства она снижается приблизительно экспоненциально. На практике исходные требования поэтапно уточняются, модифицируются, расширяются и детализируются по согласованию между заказчиком и разработчиком. Базой таких уточнений являются **неформализованные представления, знания и квалификация специалистов – заказчиков, пользователей и разработчиков**, а также результаты промежуточных этапов производства.

Однако установить не корректность таких эталонов еще труднее, чем обнаружить дефекты в готовых программах, так как принципиально отсутствуют формализованные данные, которые можно использовать как эталоны.

В сложных комплексах программ статистика и распределение типов ошибок и выполняемых изменений для коллективов разных специалистов нивелируются и проявляются достаточно общие закономерности, которые могут использоваться **как ориентиры** при их выявлении. Каждому типу необходимых корректировок соответствует более или менее определенная категория специалистов, являющихся источником дефектов данного типа (см. таблицу 7.1). Такую корреляцию целесообразно учитывать как **общую качественную тенденцию** при анализе и поиске их причин. Этому могут помогать оценки типовых дефектов и корректировок, путем их накопления и обобщения по опыту разработки определенных классов комплексов программ в конкретных предприятиях.

Изучение свойств и освоение характеристик типовых ошибок и дефектов непосредственно связано с достигаемой корректностью, безопасностью и надежностью функционирования программных продуктов и **помогает повышать профессиональную квалификацию специалистов, чтобы уметь:**

- оценивать реальное состояние проекта и планировать необходимую трудоемкость и длительность для его завершения и устранения актуальных ошибок;
- выбирать методы и средства автоматизации тестирования компонентов и комплекса программ, адекватные текущему состоянию разработки и сопровождения, наиболее экономически эффективные для устранения определенных видов обнаруженных дефектов и ошибок;
- рассчитывать необходимую экономическую эффективность контрмер и дополнительных средств оперативной защиты от потенциальных дефектов и не выявленных ошибок;
- оценивать требующиеся ресурсы, с учетом экономических затрат на реализацию контрмер при модификации программ для устранения дефектов и ошибок.

Статистика ошибок и дефектов в комплексах программ и их характеристики в конкретных типах проектов, могут служить **ориентирами для квалифицированных руководителей и специалистов**

при распределении ограниченных ресурсов в жизненном цикле комплексов программ и предохранять от *излишнего оптимизма* при оценке достигнутого качества программных продуктов. В сложных комплексах программ статистика и распределение типов выполняемых изменений для коллективов разных специалистов нивелируются и проявляются достаточно общие закономерности дефектов и ошибок, которые целесообразно использовать как ориентиры при их систематизации, выявлении и устранении. Этому могут помогать оценки типовых дефектов, модификаций и корректировок, путем их накопления и обобщения по опыту создания определенных классов и уровня сложности программных продуктов в конкретных предприятиях.

Организация, экономика и процессы испытаний сложных комплексов программ

Любые *испытания экономически ограничены допустимым количеством и объемом тестирования*, а также длительностью работы комиссии испытателей, поэтому *не могут гарантировать абсолютную проверку* соответствия требованиям функций и характеристик качества программного продукта. Экономические ресурсы, определяющие прямые затраты на выполнение испытаний сложных программных продуктов обычно находятся в пределах 10 – 15% бюджета проекта. Однако подготовка тестов и динамических моделей имитации внешней среды реального времени могут требовать больших экономических затрат *соизмеримых со стоимостью разработки испытываемого комплекса программ*. Поэтому при анализе экономики испытаний комплексов программ реального времени, особое внимание следует уделять средствам их инструментальной и тестовой поддержке на основе имеющихся прецедентов.

Для повышения достоверности определения и улучшения оценивания характеристик, после внутренних или квалификационных испытаний, некоторые типы комплексов программ целесообразно передавать пользователям на *опытную эксплуатацию в типовых условиях* (см. рис. 11.1). Это позволяет более глубоко оценить эксплуатационные характеристики созданного комплекса и устранить некоторые дефекты и ошибки. Опытную эксплуатацию целесообразно проводить разработчиками с участием испытателей-заказчиков и некоторых пользователей, назначаемых заказчиком. Результаты и характе-

ристики качества, полученные при опытной эксплуатации, могут учитываться для сокращения заказчиком приемочных испытаний.

Определение задач испытаний должно предусматривать анализ документов, в которых сформулированы требования к программному продукту, и выделение необходимых работ, чтобы затем проверять функции и характеристики, отражающиеся этими требованиями [10]. По списку и содержанию требований может быть определена матрица, составляющая задачи испытаний, которым присваиваются определенные приоритеты. В качестве руководящих принципов для присвоения приоритетов функциям должны использоваться документы, содержащие формулировки требований. Новым функциям и характеристикам имеет смысл посвящать повышенное внимание, по сравнению со слегка модифицированными компонентами. Кроме того, необходимо уделять больше времени на проверку проектных решений и структур, а также на отладку новых компонентов. Необходимо учитывать затраты на задачи, которые не имеют прямого отношения к документам с требованиями, но в то же время **определяют и экономически обеспечивают процессы испытаний**:

- инспекции и критические просмотры для проверки и пересмотра архитектуры программного комплекса и компонентов;
- составление плана и Программы проведения испытаний;
- разработка тестовых сценариев и динамических моделей внешней среды для генерации тестов;
- отладка тестовых сценариев и генераторов тестов;
- верификация, проверка, выявление и исправление дефектов генераторов динамических тестов;
- определение качества тестов и степени отражения функций и характеристик программного продукта, процесса их сбора и использования;
- пересмотр и корректировка эксплуатационной документации.

Испытания программного продукта и системы в целом выполняется, чтобы продемонстрировать представителям заказчика, что **удовлетворены все требования** технического задания, а функции и характеристики качества соответствуют условиям контракта (см. рис. 11.1). Они должны покрывать все требования в спецификациях системы и подсистем, а также требования к интерфейсу с внешней средой. Испытания должны включать тестирование на объектной вычислительной системе или на альтернативной модели системы, одобрен-

ной представителем заказчика. В процессе **системного тестирования** проверяется интеграция отдельных частей, в совокупности составляющих систему в целом. Испытания на системном уровне обычно проводится специальной группой тестирования, которая должна провести анализ с целью определения требований и компонентов функциональности, которые могут **вызывать наибольшее количество проблем**. Тест-менеджер несет ответственность за применение тестов согласно Программе и плану-графику, а также за распределение и перераспределение работ между тестировщиками для разрешения проблем, возникающих в ходе испытаний – рис. 11.2.

Программа испытаний сложных программных продуктов включает:

- утвержденные заказчиком требования технического задания, и эталоны на компоненты, программный продукт и систему;
- комплект стандартов и нормативных документов проекта программного продукта;
- комплект адекватной эксплуатационной и технологической документации на программный продукт;
- план испытаний программного продукта на соответствие требованиям и эталонам;
- методики и матрицы трассирования тестов испытаний на соответствие требованиям, плану и Программе испытаний;
- протоколы результатов испытаний программного продукта по разделам требований, эталонов, Программы и плана испытаний;
- протокол завершения опытной эксплуатации программного продукта и перспективы его развития;
- анализ результатов испытаний для усовершенствования методов и процессов тестирования комплексов программ;
- протокол готовности программного продукта к поставке пользователям;
- акт о завершения успешных испытаний и выполнении требований на программный продукт и систему.

Рис. 11.2

Руководитель производства программного продукта должен участвовать в разработке и регистрации процессов подготовки к тестированию, тестовых вариантов, сценариев и тестовых процедур, которые нужно использовать для полного испытания программного продукта и системы, и в прослеживании полноты **соответствия**

между тестовыми вариантами и требованиями к функциям и характеристикам системы. Каждое проверяемое требование должно соответствовать конкретным, обоснованным характеристикам системы, иметь уникальный для проекта идентификатор, чтобы можно было провести тестирование и проследить его выполнение с помощью объективного теста. Для каждого требования должны выбираться квалификационные методы для функциональных подсистем и компонентов программного продукта, которые необходимо прослеживать в требованиях к системе. Степень детализации требований следует выбирать, учитывая в первую очередь те функции и характеристики качества программного продукта, которые внесены в условия приемки системы, и отдавать *приоритет* тем из них, которые заказчик требует обеспечить *обязательно*.

Приемо-сдаточные испытания, подразумевают участие в них заказчика и пользователей. Приемо-сдаточные испытания, как правило, представляют собой применение подмножества совокупности тестов, использованных на системном уровне. Обнаруженные в ходе приемо-сдаточных испытаний дефекты документируются в отчетах о проблемах, и определяется их приоритеты. Проблемы, устранение которых невозможно осуществить в отведенные на приемо-сдаточные испытания ресурсы и сроки, следует передавать техническому Совету заказчика и разработчиков для дальнейшего рассмотрения и оценки.

По итогам проведения приемо-сдаточных испытаний должен готовиться отчет, который содержит краткое описание произведенных работ, оценку экономики и полученных результатов тестирования. Наиболее полным и разносторонним испытаниям должна подвергаться первая версия программного продукта. При *испытаниях очередных версий программного продукта возможно* значительные сокращения объемов тестирования апробированных повторно используемых компонентов. Однако комплексные и завершающие испытания каждой новой версии программного продукта, как правило, должны проводиться в полном объеме, гарантирующем проверку выполнения *всех требований модифицированного технического задания*. Для выявления дефектов в процессе эксплуатации серийных образцов в каждом из них должен быть предусмотрен некоторый минимум средств тестирования для проверки и регистрации качества функционирования программного продукта и обнаружения искаже-

ний результатов. Эти средства должны позволять фиксировать условия неправильного функционирования программ и характер проявления дефектов при применении программного продукта.

Оценивание *качества и соответствия требованиям* программного продукта при приемо-сдаточных испытаниях должно проводиться комиссией заказчика, в которой участвует руководитель (главный менеджер) разработки и некоторые ведущие разработчики, или аттестованной сертификационной лабораторией [9, 26, 31]. *Комиссия при испытаниях должна руководствоваться следующими основными документами* (см. рис. 11.2):

- утвержденными заказчиком и согласованными с разработчиком контрактом, техническим заданием и спецификациями требований на программный продукт и систему;
- действующими государственными и ведомственными стандартами на жизненный цикл и испытания крупных комплексов программ, на технологическую и эксплуатационную документацию, а также стандартами де-факто, согласованными с заказчиком для использования – профилем стандартов и нормативных документов;
- Программой и планом испытаний по всем требованиям контракта, технического задания и спецификаций;
- методиками испытаний и матрицей тестов, охватывающими каждый раздел требований технического задания, спецификаций и Программы испытаний;
- комплектом адекватной эксплуатационной и технологической документации на программный продукт.

План испытаний комплекса программ должен описывать порядок тестирования функциональных компонентов и подсистем, тестовую внешнюю среду, которая будет использоваться при тестировании, идентифицировать выполняемые тесты и определять план-график тестовых действий (см. главу 10). Завершаются испытания предъявлением заказчику на утверждение комплекта документов, содержащих *результаты комплексных испытаний* версии программного продукта:

- откорректированные тексты программ и данных на языке программирования и в объектном коде, а также полные спецификации требований на программные компоненты и комплекс в целом после завершения тестирования и испытаний;

- тесты, сценарии и генераторы динамических тестовых данных, использованные для испытаний программных компонентов и версии продукта в целом;
- результаты и протоколы тестирования, функциональные и конструктивные характеристики программного продукта в реальной внешней среде;
- отчет о подтверждении полных характеристик достигнутого качества функционирования, а также о степени покрытия тестами спецификации требований к программному продукту;
- комплект эксплуатационной документации, описание программного продукта и руководство пользователя в соответствии с условиями контракта;
- план, методики и средства автоматизации обучения заказчика и пользователей применению испытанной версии программного продукта;
- технические условия на версию программного продукта, базу данных управления конфигурацией и полную технологическую документацию для тиражирования и серийного производства;
- руководство по генерации и инсталляции пользовательских версий и загрузке базы данных в соответствии с условиями и характеристиками внешней среды;
- протокол готовности программного продукта к поставке пользователям;
- *отчет об экономических показателях* завершеного проекта версии программного продукта, выполненных планах и использованных ресурсах.

Критерии завершения испытаний могут содержать решения:

- экономические ресурсы и время, отведенное на испытания, истекло, хотя неизвестно достигнутое качество программного продукта;
- завершены все запланированные циклы тестирования, выполнен план и Программа проведения испытаний;
- профиль устраненных ошибок и дефектов соответствует критерию выхода из испытаний и достигнуто возможное количество не устраненных ошибок на тысячу строк программного кода.

В задачу комиссии испытаний *входит представление технических и экономических результатов* и выработка рекомендаций относительно готовности продукта к поставкам. Эта оценка должна

содержать ссылку на отчетные документы по результатам испытаний. С целью оценки готовности версии программного продукта к поставке целесообразно проводить специальные **совещания разработчиков и заказчика**, фиксировать имена участников, принятое и утвержденное ими решение, касающееся возможности поставки программного продукта.

Перечисленные документы по всей Программе испытаний **обобщаются в акте** о завершении испытаний и готовности к поставке и/или к предъявлению для сертификационных испытаний версии программного продукта. В результате должно делаться **заключение о степени соответствия программного продукта требованиям заказчика** и о завершении работы с положительным или отрицательным итогом. При выполнении всех требований технического задания заказчик обязан принять программный продукт и документацию, рассчитаться за финансирование с разработчиками и проект считается завершенным.

Стандартизация и сертификация для обеспечения качества производства сложных программных продуктов

Основные концепции программной инженерии сконцентрировались, детализировались и формализовались в **целостном комплексе (около 50-и) систематизированных международных стандартов**, охватывающих и регламентирующих практически все производственные процессы жизненного цикла сложных программных комплексов (см. Приложение). Эти стандарты сосредоточили опыт, а также эффективные технологии разработки и сопровождения крупных программных продуктов высокого качества в различных отраслях промышленности и в оборонной технике. Несколько десятков стандартов этого комплекса допускают отбор унифицированных процессов производства в зависимости от характеристик и особенностей конкретного проекта, которые целесообразно использовать при **формировании и оценке эффективной экономики производства программных продуктов**.

Практическое применение стандартов, **сосредоточивших мировой опыт** создания различных типов сложных комплексов программ, способствует значительному снижению стоимости, трудоемкости, длительности и улучшению других экономических показателей про-

ектов программных продуктов, повышению производительности труда специалистов и качества создаваемых продуктов. Жизненный цикл сложных программных комплексов отражается в стандартах набором производственных процессов, этапов, частных работ и операций в последовательности их выполнения и взаимосвязи, регламентирующих проектирование и производство, сопровождение и эксплуатацию, от анализа и подготовки требований до завершения испытаний ряда версий программного продукта и прекращения их использования. Стандарты включают описания исходной информации, способов и методов выполнения операций и работ, устанавливают требования к результатам и правилам их контроля, а также определяют содержание технологических и эксплуатационных документов. Они определяют организационную структуру коллективов специалистов, распределение и планирование работ, а также контроль за ходом производства. Эти стандарты, в частности, определяют модификацию и способы повторного применения программных компонентов и комплексов, их расширяемость и переносимость на различные аппаратные и операционные платформы, что непосредственно отражается на **экономической эффективности технологий и процессов** создания различных программных продуктов и систем.

Для регламентирования процессов жизненного цикла стандарты должны адаптироваться и конкретизироваться применительно к определенным классам и функциям проектов, процессов и компонентов программных комплексов [9, 29, 36]. Комплексное, скоординированное применение стандартов в процессе создания, развития и применения комплексов программ позволяет исключать многие виды ошибок и дефектов или значительно ослаблять их влияние. Тем самым уровень достигаемого качества продуктов **становится предсказуемым и управляемым**, непосредственно зависящим от **экономических ресурсов**, выделяемых на его достижение, а главное, от системы качества и эффективности технологии, используемых на всех этапах жизненного цикла программного комплекса.

Для **экономически эффективной деятельности специалистов**, на базе международных стандартов рекомендуется выделять для определенного проекта комплект регламентирующих документов производства программного продукта, каждый из которых имеет конкретных пользователей. В них **должно быть отражено**:

- обязательным положениям стандартов в процессе их жизненного содержания и описание выбранных положений и разделов стандартов и нормативных документов с позиции их конкретных пользователей;
- параметры адаптации разделов стандартов и содержание дополнительных нормативных документов;
- методики и сценарии корректного применения всех обязательных и рекомендуемых положений стандартов, и оценки затрат на их реализацию;
- требования к содержанию отчетов о результатах контроля и тестирования компонентов и комплекса программ на соответствие цикла.

Стандарты регламентируют определенную дисциплину коллективов специалистов при создании и применении сложных программных продуктов требуемого качества при ограниченных ресурсах. Это должно обеспечивать **экономическую эффективность и конкурентоспособность** программных продуктов. Для регламентирования жизненного цикла конкретных сложных систем и комплексов программ, целесообразно выбирать, адаптировать и применять **группы стандартов – профили**, которые определяют процессы и продукты при их производстве. Профиль стандартов – это совокупность нескольких (или подмножество одного) базовых стандартов (и других нормативных документов) с четко определенными и гармонизированными подмножествами обязательных и факультативных возможностей и требований, предназначенная для реализации заданных функций. В профиле выделяются и устанавливаются допустимые возможности и значения параметров каждого базового стандарта и/или нормативного документа, входящего в профиль. Профиль не может противоречить использованным в нем базовым стандартам и нормативным документам. Эти ограничения базовых документов профиля и их гармонизация, проведенная разработчиками профиля, должны обеспечивать высокое качество, совместимость и корректное взаимодействие компонентов системы, соответствующих профилю, в области его применения. Для проверки корректного применения стандартов и достигнутого качества производства программных продуктов применяется их сертификация.

Сертификация программного продукта служит для независимого удостоверения реального качества в испытанной, **утвер-**

жденной разработчиком и заказчиком версии программного продукта [8, 12, 25]. По заявке разработчика, заказчика или пользователей она может подвергаться обязательной или добровольной сертификации. Для этого комплект, состоящий из копии физических носителей программ, технологической и эксплуатационной документации, а также применявшиеся разработчиками требования к функциям и характеристикам комплекса программ, тесты, Программа и методики испытаний, передаются сертификационной лаборатории. При сертификационных испытаниях допускается расширение набора и параметров тестов, однако, в пределах, ограниченных требованиями и технической документацией на конкретную версию программного продукта. Успешно проведенные испытания и полученный *сертификат качества* подтверждает соответствие программного продукта утвержденным требованиям, документации, а также надежность и безопасность применения очередной версии до тех пор, пока в нее *не будут внесены какие-либо изменения*. Сертификат может быть аннулирован при любых, даже внешне незначительных, корректировках версии программного продукта.

Основной целью сертификации программных продуктов, является контроль и удостоверение качества продукции, гарантирование ее высоких потребительских свойств. Задача состоит в повышении эффективности затрат в сфере создания и применения ответственного программного продукта, а также улучшение объективности оценок его функций, характеристик и конкурентоспособности. Формальной целью сертификации является подготовка и принятие решения о целесообразности выдачи производителю *сертификата соответствия* с учетом следующих факторов:

- полноты, точности и достоверности исходного технического задания и спецификаций требований, представленных в документации на программный продукт, а также на технологию поддержки его жизненного цикла;
- достоверности и точности измерения и обобщения результатов сертификационных испытаний, получения адекватных показателей качества конечных продуктов и соответствия требованиям заказчика, а также технологических процессов их создания;
- методологии и качества интерпретации данных об объекте испытаний и/или технологии с учетом достоверности оценок, квалификации и объективности испытателей, заказчиков и пользователей.

В международных стандартах **сертификация соответствия** определена как действие **независимой стороны**, доказывающее, что обеспечивается необходимая уверенность в том, что должным образом идентифицированная продукция или процесс соответствует конкретным стандартам и/или другим нормативным документам. В понятие нормативные документы включены документы, содержащие правила, общие принципы или характеристики, касающиеся различных видов деятельности или их результатов, стандарты, технические требования, инструкции и регламенты по применению конкретной продукции или технологии. Специалисты – сертифициаторы имеют право на расширение условий испытаний в пределах требований на программный продукт и нормативной документации, при которых должно обеспечиваться заданное качество и безопасность результатов применения продукта. Разработчики или поставщики комплексов программ и их компонентов, заказчики, потребители или пользователи могут выступать инициатором – **заявителем на сертификационные испытания**.

Результатом положительных испытаний является **сертификат соответствия** – документ, изданный по правилам определенной Системы сертификации, удостоверяющий, что обеспечивается необходимая уверенность в том, что должным образом идентифицированная продукция или процесс соответствует конкретным стандартам и/или другим нормативным документам. Срок действия сертификата обычно ограничен либо по времени (например, 3 года), либо до проведения модификации продукта или процесса. Сертификат вступает в действие с момента его регистрации в государственном реестре. Для удостоверения качества программных продуктов и их компонентов, заявитель может потребовать сертифицировать технологические процессы, обеспечивающие их жизненный цикл. Поэтому часто рассматриваются совместно задачи сертификации **конечных объектов – программных продуктов**, а также **технологий и систем качества**, обеспечивающих их создание и совершенствование – рис. 11.3. Сертификационные испытания являются наиболее формализованным и регламентированным этапом тестирования, как **объектов** – программных продуктов, так и **процессов** их создания, поддерживаемым значительным числом стандартов и документов.

Обеспечение и удостоверение высокого качества сложных программных продуктов – сертификация технологий и конечного программного продукта

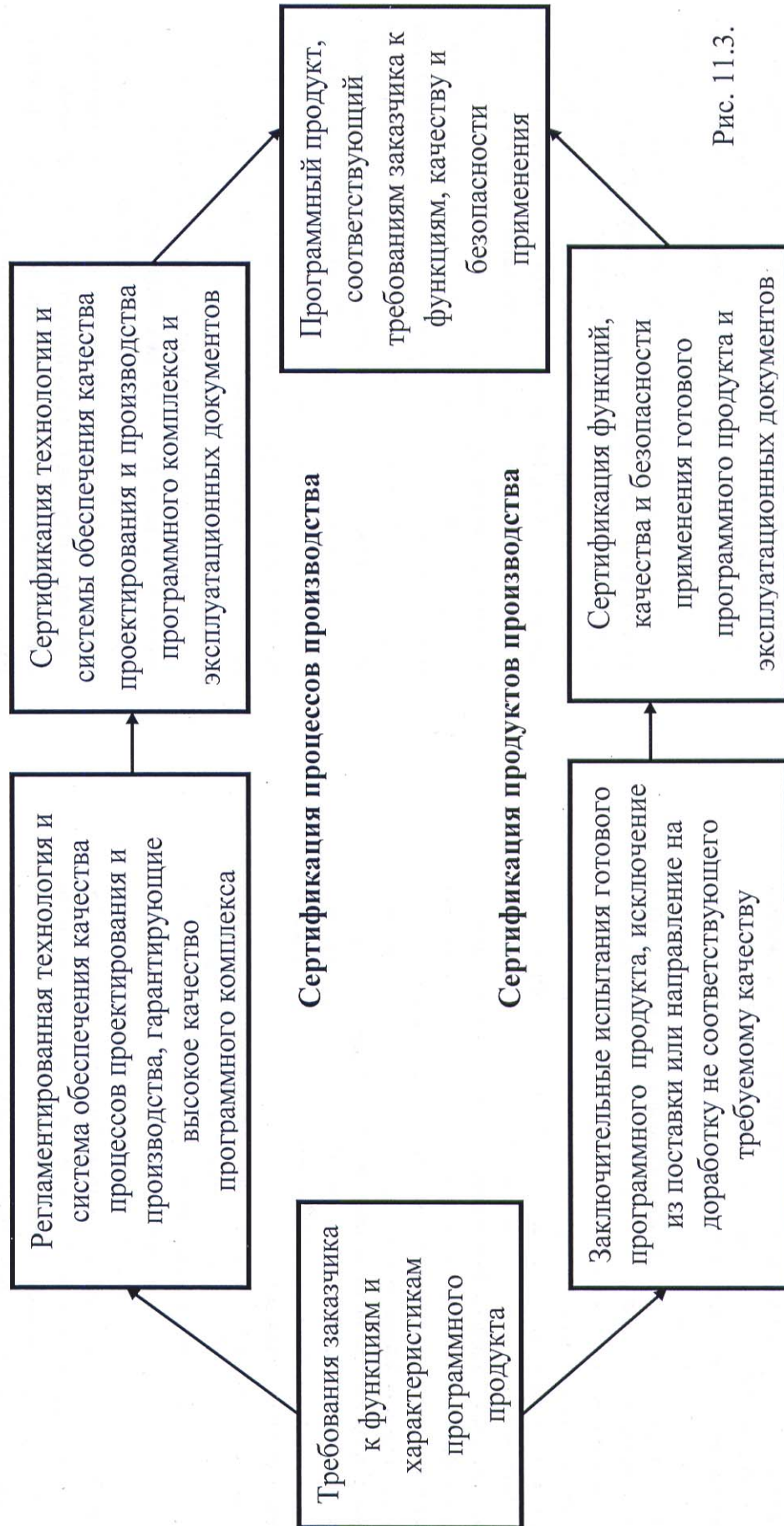


Рис. 11.3.

В исходных нормативных документах и требованиях по сертификации должны быть сосредоточены **все функциональные и эксплуатационные характеристики**, обеспечивающие заказчику и пользователям возможность корректного применения сертифицированного продукта и/или технологического процесса во всем многообразии его функций и характеристик качества. Для особенно важной продукции, например, программных продуктов по государственным заказам для оборонной техники, результаты положительной сертификации технологии производства и системы качества, могут использоваться заказчиком как основание для выдачи лицензии на поставку соответствующей продукции по госзаказу.

Испытания для сертификации проводятся в проблемно-ориентированных, технически компетентных испытательных лабораториях, **аккредитованных на право проведения тех испытаний**, которые предусмотрены в их нормативных документах. Такие проверки могут проводиться по графику или вследствие важных изменений технологии и системы качества предприятия, процессов жизненного цикла и качества продукции, а также после проведения корректирующих действий требований заказчика к версии программного продукта. При сертификации обычно руководствуются следующими основными **исходными документами**:

- действующими международными, государственными и ведомственными стандартами на проектирование, производство и испытания программных продуктов, на их жизненный цикл, системы обеспечения и характеристики качества, а также на эксплуатационную и технологическую документацию;
- утвержденным заказчиком и согласованным с разработчиком техническим заданием и/или спецификацией требований на функции и характеристики программного продукта, утвержденным комплектом технологической и эксплуатационной документации, а также на технологию и систему обеспечения их качества;
- Программой и планом сертификационных испытаний по всем требованиям технического задания и спецификации требований, положениям эксплуатационной документации;
- методиками испытаний по каждому разделу требований технического задания и документации.

Ресурсы для сертификации программных продуктов, технологий и систем качества предприятия, должны выделяться заказчи-

ком или поставщиком в зависимости от характеристик испытываемого продукта или процесса. Определяющими ресурсами сертификации обычно являются: возможная трудоемкость и длительность испытаний, совокупная численность и структура коллектива специалистов – сертифицированных, а также их квалификация и подготовленность к коллективной проверке конкретного типа программных продуктов, его компонентов или технологии и системы качества предприятия. На основании испытаний оцениваются полученные результаты и обосновываются выводы о соответствии – несоответствии продукции или технологических процессов требованиям документов.

Заключение по результатам сертификационных испытаний разрабатывается экспертами – сертифицированными и содержит обобщенные сведения о результатах испытаний и обоснование целесообразности выдачи сертификата. Результаты анализа состояния технологии или продукции **оформляются актом**, в котором даются оценки по всем позициям Программы и плана испытаний и содержатся выводы, включающие общую оценку состояния производства, необходимость корректирующих мероприятий. За сертифицированными программными продуктами в процессе их эксплуатации в течение всего срока действия сертификата соответствия может осуществляться **инспекционный контроль**. Контроль проводится в форме периодических и/или внеплановых проверок результатов эксплуатации программного продукта, а также соблюдения на предприятии требований к качеству сертифицированной технологии при производстве новой или версии продукции. **Гарантии высокого качества** критических программных продуктов, достигаемых за счет стандартизации и сертификации их производства, могут требовать значительных экономических ресурсов, которые могут достигать 20% от общего бюджета сложного проекта.

Заявитель должен подготовить и предъявить испытательной лаборатории согласованный между заказчиком и разработчиком и утвержденный комплект документов для проверки их достоверности, достаточности состава и качества изготовления в соответствии с нормативными документами. **Ориентировочный комплект основных документов при сертификации состоит из трех групп:**

- **базовые нормативные документы** на программные продукты и технологии в соответствии с номенклатурой и содержанием **серии стандартов ISO 9000:2000** или **СММІ:2003** и **на жизненный цикл**

комплексов программ, а также подготовленные разработчиками на их основе Программа, Руководства и инструкции, предъявляемые испытателям (экспертам) системы сертификации;

- **исходные документы**, характеризующие: конкретное предприятие или проект, требования к функциям и характеристикам программного продукта, комплекс тестов и эксплуатационных документов подготавливаемые руководством и специалистами предприятия или проекта для сертификации его качества;

- **отчетные документы** испытателей – сертифицированных, отражающие результаты проверки (сертификации) технологии предприятия и/или программного продукта, представляемые органу сертификации, заявителю и руководству проверяемого предприятия.

После завершения сертификационных испытаний новой версии программного продукта, обычно **осуществляется процесс ее внедрение для применения**. Это производится, как правило, **в два этапа**: силами разработчиков версии в целях обкатки, проверки и выявления ошибок в изменениях на этапе опытной эксплуатации; и/или посредством использования специализированных коллективов сопровождения, для тиражирования и распространения версий программного продукта. Основные обязанности специалистов сводятся к передаче физических носителей с текстами программ и комплектом эксплуатационной документации, а также к проведению консультаций для выделенной группы специалистов заказчика и пользователей. Разработчики и испытатели в этом случае получают возможность непосредственно **контролировать работу пользователей** с системой и документацией, что обеспечивает высокую оперативность отработки замечаний и рекламаций, формирование квалифицированных предложений для изменений, оценку эффективности применения версии программного продукта. Кроме того, должны разрабатываться учебно-методический план, подготавливаться учебные пособия, необходимые для **обучения пользователей**, а также проводиться обучение выделенной группы специалистов, ответственных за последующее обучение коллективов пользователей.

В процессе жизненного цикла большое значение имеет **история эксплуатации программного продукта**, развития его версий и соответствующая документация. Еще на стадии проектирования первой версии могут возникать **идеи совершенствования комплекса программ**, которые в то время невозможно реализовать из-за высокой

стоимости, ограниченных сроков проектирования или по иным причинам. Идеи изменения могут быть направлены на коренное улучшение функциональных возможностей программ или некоторые «косметические» улучшения реализуемых функций. Идеи небольших корректировок программ целесообразно накапливать отдельно от предложений по существенному совершенствованию программного продукта. Таким образом, должен создаваться документ – **исходные данные для изменения требований** и планирования доработок в процессе сопровождения, содержащий разделы:

- выявленные дефекты и ошибки в программном продукте;
- предложения по совершенствованию функций и улучшению качества эксплуатируемых версий программного продукта;
- идеи и предполагаемая экономическая эффективность коренной модернизации, расширения функций и улучшения характеристик программного продукта.

После внесения изменений в требования специалисты разрабатывают и испытывают конкретные корректировки программного продукта. Они должны провести проверки каждого внесенного изменения совместно с заказчиком, утвердившим изменение в целях подтверждения целостности и работоспособности измененного программного продукта. Следует получить подтверждения того, что внесенные **изменения удовлетворяют требованиям заказчика**, установленным в договоре, посредством вспомогательного процесса испытаний качества, проведения аудита функциональной и физической конфигурации. Результатами данной работы являются: **новая версия** программного продукта, включающая в себя принятые изменения требований; отклоненные изменения; отчет о приемке версии; отчеты о проверках и аудитах; отчет об испытаниях откорректированного программного продукта.

Снятие сертифицированной версии программного продукта с эксплуатации и развитие версий должны быть подготовлены анализом, обосновывающим такое решение. При анализе следует определить и экономически обосновать: возможность и полезность сохранения устаревшей версии комплекса программ, а также необходимость создания и применения новой версии программного продукта. При снятии программного продукта с сопровождения следует определить необходимые для этого действия, а затем разработать и документально оформить этапы работ, обеспечивающие их эффективное выпол-

нение. Должны быть предусмотрены возможности **доступа к полным архивным документам** снятого с сопровождения сертифицированного программного продукта.

Перечень основных стандартов, регламентирующих проектирование и производство сложных программных продуктов

1. **CMMI – Capability Maturity Model Integration for Product and Process Development** – Интегрированная модель оценивания зрелости продуктов и процессов разработки программных средств.
2. **ISO 15288:2002.** Системная инженерия. Процессы жизненного цикла систем.
3. **ISO 19760:2003.** – Системная инженерия. Руководство по применению стандарта ISO 15288.
4. **ISO 12207:1995.** (ГОСТ Р – 1999). ИТ. Процессы жизненного цикла программных средств.
5. **ISO 12207:1995.** – ИТ. Процессы жизненного цикла программных средств. **Изменения 1 и 2:2002-2004.**
6. **ISO 15271:1998.** (ГОСТ Р – 2002). ИТ. Руководство по применению ISO 12207.
7. **ISO 16326:1999.** (ГОСТ Р – 2002). ИТ. Руководство по применению ISO 12207 при административном управлении проектами.
8. **ISO 15504:1-5:2003-2006.** ИТ. Процесс аттестации. Ч.1. Концепция и словарь. Ч.2. Выполнение аттестации. Ч. 3. Руководство по производству аттестации. Ч.4. Руководство пользователей для процессов усовершенствования и определения зрелости процессов. Ч. 5. Образец модели процессов аттестации.
9. **ГОСТ Р 51904 – 2002.** Программное обеспечение встроенных систем. Общие требования к разработке и документированию.
10. **ISO 9000:2000.** (ГОСТ Р – 2001). Система менеджмента (административного управления) качества. Основы и словарь.
11. **ISO 9001:2000.** (ГОСТ Р – 2001). Система менеджмента (административного управления) качества. Требования.
12. **ISO 9004:2000.** (ГОСТ Р – 2001). Система менеджмента (административного управления) качества. Руководство по улучшению деятельности.
13. **ISO 90003:2004** – Руководство по организации применения стандарта **ISO 9001:2000** для программных средств.

14. **ISO 10005: 1995.** (ГОСТ) Административное управление качеством. Руководящие указания по программам качества.

15. **ISO 10006: 1997.** (ГОСТ) Руководство по качеству при управлении проектом.

16. **ISO 10007: 1995.** (ГОСТ) Административное управление качеством. Руководящие указания при управлении конфигурацией.

17. **ISO 10011-1-3: 1990.** (ГОСТ) Руководящие положения по проверке систем качества. Ч.1. Проверка. Ч.2. Квалификационные критерии для инспекторов-аудиторов систем качества. Ч.3. Управление программами проверок.

18. **ISO 12182:1998.** (ГОСТ Р– 2002). ИТ. Классификация программных средств.

19. **ISO 9126:1991.** (ГОСТ – 1993). ИТ. Оценка программного продукта. Характеристики качества и руководство по их применению.

20. **ISO 14598-1-6:1998-2000.** Оценивание программного продукта. Ч.1. Общий обзор. Ч. 2. Планирование и управление. Ч. 3. Процессы для разработчиков. Ч.4. Процессы для покупателей. Ч.5. Процессы для оценщиков. Ч. 6. Документирование и оценивание модулей.

21. **ISO 9126-1-4: 2002.** ИТ. ТО. Качество программных средств: Ч.1. Модель качества. Ч.2. Внешние метрики. Ч. 3. Внутренние метрики. Ч. 4. Метрики качества в использовании.

22. **ISO 25000:2005** ТО. – Руководство для применения новой серии стандартов по качеству программных средств на базе обобщения стандартов ISO 9126:1-4: 2002 и ISO 14598:1-6:1998-2000.

23. **ISO 15939: 2002** – Процесс измерения программных средств.

24. **IEC 61508:1-6: 1998-2000.** Функциональная безопасность электрических / электронных и программируемых электронных систем. Часть 3. Требования к программному обеспечению. Часть 6. Руководство по применению стандартов IEC 61508-2 и IEC 61508-3.

25. **ISO 15408 -1-3. 1999.** (ГОСТ Р – 2002). Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Ч.1. Введение и общая модель. Ч.2. Защита функциональных требований. Ч. 3. Защита требований к качеству.

26. **ISO 13335 - 1-5. 1996-1998.** ИТ. ТО. Руководство по управлению безопасностью. Ч.1. Концепция и модели обеспечения безо-

пасности информационных технологий. Ч.2. Планирование и управление безопасностью информационных технологий. Ч.3. Техника управления безопасностью ИТ. Ч.4. Селекция (выбор) средств обеспечения безопасности. Ч.5. Безопасность внешних связей.

27. **ISO 10181: 1-7. ВОО. 1996-1998.** Структура работ по безопасности в открытых системах. Ч.1. Обзор. Ч. 2. Структура работ по аутентификации. Ч.3. Структура работ по управлению доступом. Ч.4. Структура работ по безотказности. Ч.5. Структура работ по конфиденциальности. Ч.6. Структура работ по обеспечению целостности. Ч.7. Структура работ по проведению аудита на безопасность.

28. **ISO 14252:1996 (IEEE 1003.0).** Руководство по функциональной среде открытых систем POSIX.

29. **ISO 9945:1-4:2003** – ИТ. Интерфейсы переносимых операционных систем. Ч.1. Базовые определения. Ч.2. Системные интерфейсы. Ч.3. Команды управления и сервисные программы. Ч 4. Обоснование.

30. **ISO 13210:1994.** ИТ. Методы тестирования для измерения соответствия стандартам POSIX.

31. **ISO 14756: 1999.** ИТ. Измерение и оценивание производительности программных средств компьютерных вычислительных систем.

32. **ISO 12119:1994.** (ГОСТ Р – 2000 г). ИТ. Требования к качеству и тестирование.

33. **ANSI/IEEE 829 - 1983.** Документация при тестировании программ.

34. **ANSI/IEEE 1008 - 1986.** Тестирование программных модулей и компонентов ПС.

35. **ANSI/IEEE 1012 - 1986.** Планирование верификации и подтверждения достоверности качества (валидации) программных средств.

36. **ISO 14764: 1999.** (ГОСТ Р – 2002). ИТ. Сопровождение программных средств.

37. **ISO 15846:1998.** ТО. Процессы жизненного цикла программных средств. Конфигурационное управление программными средствами.

38. **ISO 16085: 2004** – Характеристики процессов управление рисками при разработке, применении и сопровождении программных средств.

39. **ISO 6592:2000** ОИ – Руководство по документации для вычислительных систем.

40. **ISO 9294:1990** (ГОСТ–1993 г). ТО. ИТ. Руководство по управлению документированием программного обеспечения.

41. **ISO 9127:1990** (ГОСТ–1993 г). ТО. ИТ. Руководство по управлению документированием программного обеспечения.

42. **ISO 15910:1999** (ГОСТ Р – 2002) ИТ. Пользовательская документация программных средств.

43. **ISO 18019:2004** ИТ. Руководство по разработке пользовательской документации на прикладные программные средства для офисов, бизнеса и профессиональных применений.

44. **РД 50-34.698-90**. Методические указания. Информационная технология. Автоматизированные системы. Требования к содержанию документов.

45. **ГОСТ Р 51901-2002**. Управление надежностью. Анализ риска технологических систем.

46. **DO-178 В -1995**. Соглашение по сертификации бортовых систем и оборудования в части программного обеспечения.

47. **ISO 14143: 1-5: 1998 – 2004**. ИТ. Измерение программных средств. Измерение функционального размера. Ч.1. 1998. Определение концепции. Ч.2. 2002. Оценивание соответствия методов измерения размера программных средств стандарту ISO 14143:1:1998. Ч.3. 2003. Верификация методов измерения функционального размера. Ч.4.2002. Эталонная модель. Ч.5. 2004. Определение функциональных доменов для использования при измерении функционального размера.

48. **ISO 20926:2003**. Руководство по практическому методу измерения функционального размера программных средств.

49. **ISO 20968:2002**. Руководство по расчетам на основе анализа функциональных точек – Марк II.

Литература

1. Бейзер Б. Тестирование черного ящика. Технология функционального тестирования программного обеспечения и систем. Пер. с англ. – СПб: ПИТЕР. 2004.
2. Блэк Р. Ключевые процессы тестирования. Пер. с англ. – М: ЛОРИ. 2006.
3. Боэм Б.У. Инженерное проектирование программного обеспечения. Пер. с англ./Под ред. А.А. Красилова. – М.: Радио и связь. 1985. (Barry W. Boehm. Software Engineering Economics. Prentice-Hall. 1981).
4. Брауде Э. Д. Технология разработки программного обеспечения. Пер. с англ. – М.: ПИТЕР. 2004.
5. Вигерс К.И. Разработка требований к программному обеспечению. Пер. с англ. – М.: Русская редакция. 2004.
6. Волков О.И., Склярченко В.К. Экономика предприятия. – М.: ИНФРА-М. 2007.
7. Гецци К., Джазайери М., Мандриоли Д. Основы инженерии программного обеспечения. Пер. с англ. – СПб.: БХВ-Петербург. 2005.
8. Гринфилд Д., Шорт К. Фабрика разработки программ. Пер. с англ. – М: Диалектика. 2007.
9. Дастин Э., Рэшка Д., Пол Д. Автоматизированное тестирование программного обеспечения. Внедрение, управление и эксплуатация. Пер. с англ. – М. ЛОРИ. 2003.
10. Калбертсон Р., Браун К., Кобб Г. Быстрое тестирование. Пер. с англ. – М.: Вильямс. 2002.
11. Канер С., Фолк Д., Нгуен Е. Тестирование программного обеспечения. Пер. с англ. – М: ДиаСофт. 2001.
12. Кантор М. Управление программными проектами. Практическое руководство по разработке успешного программного обеспечения. Пер. с англ. – М.: Вильямс. 2002.
13. Коберн А. Современные методы описания требований к системам. Пер. с англ. – М.: Лори. 2002.
14. Кузнецов С.Д. Базы данных. Модели и языки. Учебник. –

М. БИНОМ. 2008.

15. Леффингуэлл Д., Уидриг Д. Принципы работы с требованиями к программному обеспечению. Унифицированный подход. Пер. с англ. – М.: Вильямс. 2002.

16. Липаев В.В., Потапов А.И. Оценка затрат на разработку программных средств. – М.: Финансы и статистика. 1988.

17. Липаев В.В. Методы обеспечения качества крупномасштабных программных средств. – М.: РФФИ. СИНТЕГ. 2003.

18. Липаев В.В. Техничко-экономическое обоснование проектов сложных программных средств. – М.: СИНТЕГ. 2004.

19. Липаев В.В. Функциональная безопасность программных средств. – М.: СИНТЕГ. 2004.

20. Липаев В.В. Анализ и сокращение рисков проектов сложных программных средств. – М.: СИНТЕГ. 2004.

21. Липаев В.В. Документирование сложных программных средств. – М.: СИНТЕГ. 2005.

22. Липаев В.В. Программная инженерия. Методологические основы. Учебник. – М.: ТЕИС. 2006.

23. Липаев В.В. Тестирование крупных комплексов программ на соответствие требованиям. – М.: ГЛОБУС. 2008.

24. Липаев В.В. Человеческие факторы в программной инженерии: рекомендации и требования к профессиональной квалификации специалистов. Учебник. – М.: СИНТЕГ. 2009.

25. Липаев В.В. Сертификация программных средств, Учебник. – М.: СИНТЕГ. 2010.

26. Липаев В.В. Тестирование компонентов и комплексов программ. Учебник. – М.: СИНТЕГ. 2010.

27. Мазер Г.Я., Мхитарян Н.А. Экономика машиностроения. – М.: Изд. МГОУ. 2008.

28. Организация производства и управления предприятием. Учебник. Под ред. О.Г. Тууровца. – М.: ИНФРА-М. 2006.

29. Оценка и аттестация зрелости процессов создания и сопровождения программных средств и информационных систем (ISO/IEC TR 15504 – СММ). – М.: Книга и бизнес. 2001.

30. Рекомендации по преподаванию программной инженерии и информатики в университетах. Пер. с англ. – М.: ИНТУИТ. 2007.

31. Соммервилл И. Инженерия программного обеспечения. 6-е издание. Пер. с англ. – М.: Вильямс. 2002.

32. Тельнов Ю.Ф. Интеллектуальные информационные системы в экономике. Учебное пособие. – М.: СИНТЕГ. 2002.
33. Трубачев А.П., Долинин М.Ю., Кобзарь М.Т. и др. Оценка безопасности информационных технологий. Общие критерии. Пер. с англ. Под ред. В.А. Галатенко. – М.: СИП РИА, 2001.
34. Тэллес М., Хсих Ю. Наука отладки. Пер. с англ. – М.: Кудиц-образ. 2003.
35. Уайт Б.А. Управление конфигурацией программных средств. Практическое руководство по Rational ClearCase. Пер. с англ. – М. ДМК Пресс. 2002.
36. Фатрелл Р. Т., Шафер Д. Ф., Шафер Л. И. Управление программными проектами: достижение оптимального качества при минимальных затратах. Пер. с англ. – М.: Вильямс. 2003.
37. Экономика промышленного предприятия. Учебник. Под ред. Е.Л. Кантора. – М.: MapT. 2007.
38. Boehm V.W. et al. Software cost estimation with COCOMO II. Prentice Hall PTR. New Jersey. 2000.
39. Charet R. Software engineering risk analysis and management. N.Y.: McGraw – Hill. 1989.
40. Davis A. Software requirements: Objects, functions and states. – Englewood Cliffs. NY. Prentice-Hall. 1993.
41. Grady R. Practical software metrics for project management and process improvement. – Englewood Cliffs. NY. Prentice-Hall. 1992.
42. Jones C. Applied software measurement, assuring productivity and quality. McGraw-Hill. NY. 1996.
43. Higuera R., Haimes Y. Software risk management. Pittsburg. Software engineering institute, Cornegie Mellon University. – 1996.
44. Howden W.E. Functional program testing and analysis. N.Y.: McGraw - Hill, 1987.
45. Karolak D. W. Software engineering risk management. IEEE Computer Society Press. Washington. 1996.
46. Londeix B. Cost estimation for software development. Cornwall: Addison-Wesley. 1987.
47. Mutafelija B., Stromberg H. Systematic process improvement using ISO 9001:2000 and CMMI. SEI. 2003.
48. Perry W.E. Effective Methods for Software Testing. NY. Wiley. 2000.

49. Quarterman J.S., Wilhelm S. Unix, Posix and open systems: The open standards puzzle. N.Y., Addison - Wesley. 1993.

50. Shooman M.L. Software Engineering: Reliability, Development and Management. N.Y. McGraw-Hill. 1983.