Институт системного программирования
Российской академии наук

В.В. Липаев

Надежность и функциональная безопасность комплексов программ реального времени

Москва - 2013

Липаев В.В.
Надежность и функциональная безопасность комплексов программ реального времени. *– М: 2013. 176 с.*

Рассмотрены основные факторы, определяющие надежность и безопасность функционирования программных комплексов реального времени, характеристики систем и среды, для которых должна обеспечиваться функциональная безопасность программных продуктов реального времени и необходимые ресурсы. Представлены требования к проектированию функциональной пригодности сложных программ, организация, планирование и процессы разработки требований к безопасности программных продуктов. Изложено содержание основных международных стандартов технологических процессов обеспечивающих функциональную безопасности в жизненном цикле комплексов программ. Значительное внимание уделено испытаниям надежности и функциональной безопасности сложных программных продуктов, Программе и методикам испытаний компонентов и комплексов программ, динамическим характеристикам программных продуктов. Рассмотрены технологические средства автоматизации испытаний функциональной безопасности программных продуктов, удостоверение требуемой функциональной безопасности и сертификации сложных программных продуктов.

Монография предназначена руководителям и разработчикам сложных систем и программных продуктов, к которым предъявляются высокие требования к надежности и функциональной безопасности применения. Книга ориентирована на исполнителей опытно-конструкторских работ, студентов и аспирантов, связанных с созданием систем и программных продуктов высокого качества и безопасности.

Оглавление

Введение

Безопасность и надежность – очень широкие понятия, относящееся к сферам экономики, техники, экологии, государственного управления, социальных процессов, средств массовой информации и к многим другим областям человеческой деятельности. Эти понятия отражают состояния, при которых отсутствует недопустимый риск (вероятность), связанный с причинением вреда техническим системам, жизни и здоровью граждан, имуществу физических и юридических лиц, государственному или муниципальному имуществу, окружающей среде. Возможности и широта применения ***сложных комплексов программ*** существенно зависит от ***методологий и стилей*** организации работы коллективов специалистов – разработчиков. Эти методологии различаются сферами применения, методами достижения высокого качества комплексов программ, психологическими и профессиональными 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:

- ***свободного (открытого) программного обеспечения*** (СПО), ориентированного на участие, практически независимых специалистов, различных по квалификации, дислокации и ответственности за результаты своей 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 определенным заказчиком и поставщиком.

Первый класс сложных программных комплексов обычно первоначально создается группами специалистов 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 разработка, изменение и совершенствование не регламентируется в реальном времени и реализуется по инициативе пользователей или разработчиков без определенных, контролируемых планов и сроков. Вследствие этого слабо документированный, непрерывно, спонтанно изменяющийся программный продукт и его производство, ***трудно или невозможно оценивать надежностью и безопасностью применения.***

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 про-***

мышленным 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 управления объектами, в высокоточном технологическом производстве, в авиации, в управлении космическими аппаратами, атомными электростанциями и оборонной техникой. Для ряда подобных систем, несмотря на бурное увеличение ресурсов современных компьютеров, ***сохраняются практические ограничения производительности и объемов памяти***, которые приходится учитывать заказчикам и разработчикам при создании и применении сложных программных продуктов этого класса. Такие изделия являются ***одними из наиболее сложных интеллектуальных систем высокой надежности и безопасности,*** создаваемых человеком. Далее ***в монографии внимание сосредоточено на этом классе сложных комплексов программ***.

Важной особенностью таких ***промышленных программных продуктов*** является их ***отчуждаемость от первичных производителей***. 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 обусловливает необходимость производить такие комплексы программ (КП) с соблюдением определенных стандартов, правил структурного построения, а также оформлять на них достаточно полную технологическую документацию, позволяющую применять, сопровождать и развивать программные продукты (ПП) различными квалифицированными специалистами под управлением заказчиков и/или поставщиков. При этом, под надежностью и функциональной безопасностью продукции, процессов производства и эксплуатации систем понимается их работоспособное состояние и функционирование в соответствии с требованиями заказчика и документацией, при которых ***отсутствуют опасные отказы и недопустимый ущерб***, связанный с причинением вреда жизни и здоровью граждан, государственному имуществу и окружающей среде, собственности физических и юридических лиц.

Понятия и характеристики функциональной безопасности систем близки к понятиям надежности, поэтому далее в тексте для сокращения, часто под термином «функциональная безопасность» будет подразумеваться и «надежность». Основное различие состоит в том, что в показателях надежности учитываются все реализации опасных отказов, а в характеристиках функциональной безопасности регистрируются и учитываются только те отказы, которые привели к столь большому или катастрофическому ущербу, который отразился на безопасности системы и информации для потребителей. Статистически таких отказов может быть в несколько раз меньше, чем учитываемых в значениях надежности. Однако методы, влияющие факторы и реальные значения показателей надежности ПП могут служить ориентирами при оценке функцио-

нальной безопасности критических систем. Поэтому способы оценки и испытаний функциональной безопасности, могут базироваться на методах, определения надежности функционирования комплексов программ и баз данных.

Ущерб от дефектов и ошибок программ и данных может иметь кумулятивный характер, и проявляться, в более или менее, систематических отказах, каждый из которых отражается на надежности, но не является катастрофой с большим ущербом, влияющим на безопасность системы. Накопление таких отказов со временем может приводить к последствиям, нарушающим функциональную безопасность систем и их применение. Таким образом, дополнительно сближаются понятия надежности и функциональной безопасности сложных систем и комплексов программ. При более или менее одинаковых источниках угроз и их проявлениях эти понятия можно разделить по величине последствий и ущерба при возникновении отказовых ситуаций.

В критических системах, в которых результаты обработки информации и управляющие воздействия непосредственно определяют работоспособность и качество применения сложных систем, не исключены и изредка происходят ***аварии и катастрофы вследствие недостаточной надежности или безопасности функционирования программных продуктов,*** иногда вследствие относительно простых ошибок и дефектов в программах. В наиболее тяжелых случаях ущерб измерялся ценой жизни и здоровья людей или большими материальными потерями. В то же время они имели вполне объяснимую природу и источники, влияние которых могло быть снижено, путем глубокого анализа, выявления дефектов и корректировок программ или информации баз данных. Имеющиеся достижения в области теории и практики управления сложной промышленной продукцией, как правило, не известны и не используются специалистами, создающими и применяющими системы на базе программных комплексов (ПК). Это во многих случаях определяет дефекты и отказовые ситуации при применении программных продуктов, конфликты между заказчиками и разработчиками из-за неопределенностей значений их безопасности. В результате проекты комплексов программ не соответствуют исходному, декларированному назначению и первоначальным спецификациям требований к основным функциям, характеристикам безопасности и надежности, не укладываются в согласованные графики и бюджет разработки. Формализация и оценивание реальной надежности и безопасности программ и систем обычно зависит от квалификации их разработчиков, заказчиков или пользователей.

В требованиях технических заданий и реализованных проектах сложных систем, ***систематически умалчиваются и/или недостаточно формализуются*** понятия и метрики надежности и безопасности программного продукта и выдаваемой информации, какими характеристиками они должны описываться, как их следует измерять и сравнивать с требованиями, отраженными в контракте, техническом задании или спецификациях. Кроме того, некоторые из характеристик надежности или безопасности часто вообще отсутствуют в требованиях заказчика и согласованных документах на систему и комплекс программ, что приводит к произвольному их учету или к пропуску при испытаниях. Этому способствует ограниченность ресурсов, необходимых для достижения и оценивания, требуемых и реализованных значений качества комплексов программ, а также недостаточная формализация и документирование всего процесса их выбора и анализа.

Разнообразие областей применения компьютеров становится все шире, и их корректная работа часто является определяющей для качественного управления объектами, успеха предприятий или безопасности человека. Поэтому тщательное специфицирование, оценивание и повышение надежности и безопасности систем, программного продукта и обработанной для потребителей информации – ***ключевой фактор обеспечения их адекватного и эффективного применения***. Это может быть достигнуто на основе выделения, определения и обеспечения необходимых характеристик качества с учетом целей использования и функциональных задач ПП и систем, с использованием стандартизированных и формализованных методов.

Непрерывно возрастающая сложность и вследствие этого уязвимость систем и ПП от случайных и предумышленных негативных воздействий, выдвинули ряд проблем, связанных с надежностью и безопасностью систем и программных продуктов, в разряд важнейших – ***стратегических***, определяющих принципиальную возможность и эффективность их применения в административных системах, в промышленности и в военной технике. При этом выделились области анализа и обеспечения: ***информационной безопасности***, связанные, в основном, с защитой от предумышленных, негативных воздействий на информационные ресурсы систем, и ***функциональной безопасности***, обусловленной отказовыми ситуациями и потерей работоспособности систем и ПП вследствие проявления в реальном времени непредумышленных, случайных дефектов и отказов программ, данных, аппаратуры и внешней среды. С позиции доминирующей категории обеспечения безопасности, автоматизированные системы, их программные средства и базы данных можно условно разделить на ***два крупных класса***:

- системы, в которых накапливаются, обрабатываются и хранятся большие объемы информации из внешней среды с активным участием пользователей, для которой должна обеспечиваться конфиденциальность, целостность и доступность данных потребителям, что отражается требованиями к *высокой информационной безопасности*;
- системы и объекты автоматизации, в аппаратуру которых ***встроены комплексы программ*** для управления и обработки информации в реальном времени, основные задачи которых состоят в обеспечении достоверной реализации эффективного и устойчивого управления объектами внешней среды при относительно малом участии пользователей в их решении и высоких требованиях к ***надежности и функциональной безопасности***.

Основными источниками отказовых ситуаций могут быть некорректные исходные требования, сбои и отказы в аппаратуре, дефекты или ошибки в программах и данных, проявляющиеся при их исполнении в соответствии с назначением. При таких воздействиях, внешняя, функциональная работоспособность систем может разрушаться не полностью, однако невозможно полноценное выполнение заданных функций и требований к качеству информации для потребителей. В рассматриваемых ниже системах, безопасность их функционирования определяется проявлениями *дестабилизирующих факторов, приносящих большой ущерб*:

- техническими отказами внешней аппаратуры и искажениями исходной информации от объектов внешней среды и от пользователей систем и обработанной информации;
- случайными сбоями и физическими разрушениями элементов и компонентов аппаратных средств вычислительных комплексов и средств телекоммуникации;
- дефектами и ошибками в комплексах программ обработки информации и в данных;
- пробелами и недостатками в средствах обнаружения опасных отказов и оперативного восстановления работоспособного состояния систем, программ и данных.

При анализе характеристик надежности и функциональной безопасности можно выделять ***два класса систем и их ПП***. ***Первый класс*** составляют системы, имеющие ***встроенные комплексы программ жесткого регламента реального времени***, автоматизировано управляющие внешними объектами или процессами. Время необходимой реакции на отказовые ситуации таких систем обычно исчисляется секундами или долями секунды, и процессы восстановления работоспособности, должны проводиться за это время, в достаточной степени автоматизировано (бортовые системы в авиации, на транспорте, в некоторых средствах вооружения, системы управления атомными электростанциями).

Системы второго класса, применяются для управления процессами и обработки деловой информации из внешней среды, в которых активно участвуют специалисты-операторы (банковские, административные, штабные военные системы). Допустимое время реакции на опасные отказы в этих системах может составлять десятки секунд и минуты, и операции по восстановлению работоспособности частично могут быть доверены специалистам-администраторам по обеспечению надежности и функциональной безопасности.

Проблемы неопределенностей концепции и характеристик функциональной безопасности конкретных систем, включающих программные продукты должны учитываться заказчиками, пользователями и разработчиками в течение всего их жизненного цикла. Чем сложнее системы и чем выше к ним требования безопасности, тем неопределеннее функции и характеристики их качества.

Неопределенности начинаются с требований заказчиков, которые при формулировке технического задания и спецификаций не полностью формализуют и принципиально не могут обеспечить достоверное содержание всего адекватного набора характеристик и значений безопасности, которые должны быть при завершении проекта и предъявлении конечного продукта заказчику. Эти требования итерационно формируются, детализируются и уточняются по согласованию между всеми участниками проекта вследствие естественной ограниченности первичных исходных данных, и изменения их под влиянием объективных и субъективных воздействий со стороны различных процессов жизненного цикла (ЖЦ).

Обычно заказчиком, ***не полностью, с необходимой детализацией определены и описаны все характеристики, особенности функционирования объектов внешней среды.*** Сложность, а поэтому и неопределенность их представления, как правило, адекватны сложности всей системы, функциональная безопасность которой должна обеспечиваться в течение ее ЖЦ. Квалификация и субъективные свойства потребителей и пользователей изменяются по мере освоения функциональных возможностей системы и ее работоспособности, что увеличивает неопределенность ее реальной надежности и безопасности. Различия свойств персонала, применяющего систему и ПП, дополнительно увеличивают неопределенность значений безопасности и трудности ее прогнозирования с учетом множества субъективных факторов различных специалистов, участвующих в их разработке и эксплуатации.

В процессе проектирования, разработки и всего жизненного цикла основных, функциональных задач, операционной среды, аппаратуры компьютеров, эти компоненты с течением времени развиваются и адаптируются, что отражается на необходимости адекватного изменения методов, задач и средств обеспечения их функциональной безопасности. Таким образом, ***проблемы обеспечения функциональной безопасности сложных систем должны решаться с учетом одновременного динамического развития всех компонентов внешней среды***, и факторов, непрерывно изменяющихся и воздействующих на результаты их решения. Однако такой сложный, непрерывный, многосвязный процесс трудно реализовать практически и его целесообразно решать поэтапно, возможно с необходимыми итерациями. При этом следует иметь в виду, что всегда могут проявиться отдаленные связи процессов, которые могут существенно повлиять на текущие работы по обеспечению качества системы и программных продуктов.

Существующие технологии способствуют повышению функциональной безопасности, снижению потенциального ущерба и рисков, однако практически всегда остается открытым вопрос, ***насколько применяемые методы оправдывают затраты*** на их реализацию и применение. Поэтому эффективность методов и средств обеспечения надежности и функциональной безопасности в общем случае, объективно и количественно отразить не удается. Для конкретных ПП и систем приходится устанавливать частные критерии эффективности

с учетом интуитивных оценок допустимых затрат на их реализацию. Испытания, эксплуатация и сертификация комплексов программ способствуют снижению неопределенности оценок эффективности и итерационному приближению к практически приемлемому уровню надежности и функциональной безопасности.

Роль ***негативных воздействий и их разрушительные последствия быстро возрастают*** в связи с ростом сложности разработки и применения современных систем на базе компьютеров и ответственности решаемых ими задач. Одновременно возрастает сложность внешней и операционной среды, в которой функционируют прикладные ПП и ответственность функций систем. Объективное повышение сложности функций, реализуемых программами в современных системах, непосредственно приводит к увеличению их объема и трудоемкости создания. Соответственно росту сложности программ ***возрастает относительное и абсолютное количество выявляемых и остающихся в них дефектов и ошибок***, что отражается на снижении потенциальной безопасности их функционирования. По мере увеличения сложности задач, решаемых программами, возрастают ошибки, которые могут угрожать катастрофами в системах, выполняющих критические функции управления крупными, дорогими и особенно важными объектами или процессами.

Для повышения надежности и функциональной безопасности рекомендуется упорядоченное, регламентированное проектирование архитектуры, разработки и сопровождения сложных систем и комплексов программ, на базе современных технологий и стандартов ISO 15288:2006 (для систем) и ISO 12207:2008 (для программных продуктов). Это позволяет предупреждать и устранять наиболее опасные системные, алгоритмические и программные дефекты и ошибки на ранних стадиях их жизненного цикла (рис. В1). Для обеспечения надежности и безопасности критических систем и программных продуктов необходимы эффективные методы и средства, предупреждающие и выявляющие дефекты, а также удостоверяющие безопасность использования программ и баз данных, оперативно защищающие их корректное функционирование при проявлении любых дефектов и отказовых ситуаций.

Технологические процессы, их действия и задачи рекомендуется располагать ***в виде упорядоченной последовательности***, подходящей для реализации конкретного проекта. Эти последовательности не предполагают и не устанавливают какой-либо зависимости от времени. Из-за невозможности достичь единого мнения или применить универсальную, развернутую во времени последовательность процессов, пользователь стандартов может самостоятельно выбрать и назначить процессы, виды деятельности и задачи, как наиболее подходящие и эффективные для определенного проекта.

Основные группы процессов жизненного цикла систем и комплексов программ

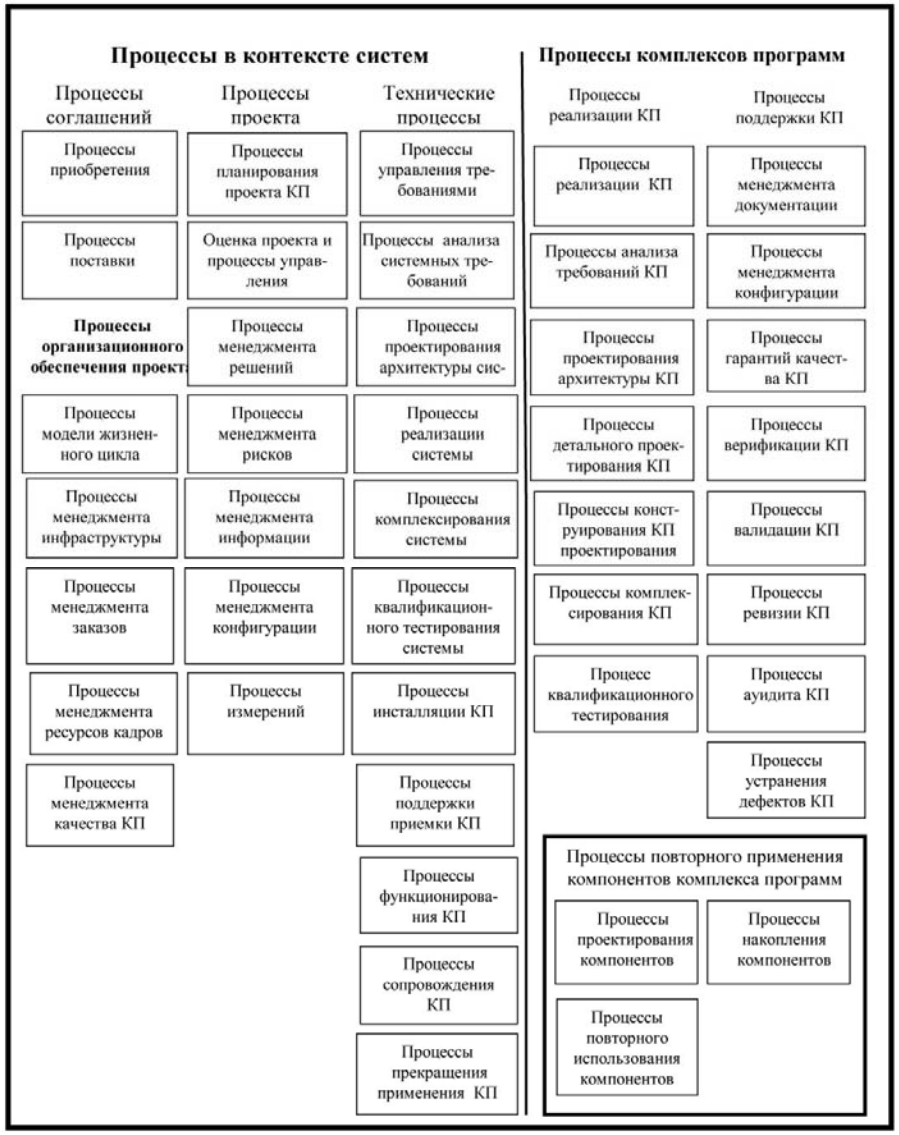

Рис. В1.

Процессы ***в контексте системы*** (**ISO 15288:2006**) делятся на три группы: процессы соглашений, проекта системы и технические процессы (всего 23 процесса). Раздел ***процессов комплексов программ*** включает две группы из 18 процессов: процессы реализации и процессы поддержки создания комплексов. По названиям они в значительной степени подобны процессам в стандарте **ISO 12207:2008**, но различаются по содержанию.

Работоспособность систем и программных продуктов может быть обеспечена при исходных данных, которые использовались при их разработке, отладке и испытаниях. ***Реальные исходные данные могут иметь значения, отличающиеся от заданных техническим заданием и от используемых при эксплуатации программ и баз данных.*** При таких исходных данных функционирование программ трудно предсказать заранее, и весьма вероятны различные аномалии, завершающиеся отказами, отражающимися на безопасности. Следует учитывать принципиальные трудности аналитического оценивания и прогнозирования значений надежности и функциональной безопасности программных комплексов, вследствие ***непредсказуемости положения, проявления и последствий дефектов и ошибок в программах и данных***. Это приводит к практической ***невозможности достоверных априорных аналитических расчетов*** функциональной безопасности комплексов программ при ее высоких значениях.

Проблема достижения требуемой надежности и функциональной безопасности систем, содержащих программы реального времени, решается путем использования ***современных регламентированных технологических процессов*** и инструментальных средств обеспечения их жизненного цикла (ЖЦ). Они должны быть поддержаны группой международных стандартов, определяющих состав и процессы выполнение требований к заданной функциональной безопасности систем и программ.

Для систематической, координированной борьбы с угрозами со стороны случайных дефектов и ошибок ***необходимы исследования факторов***, влияющих на надежность и безопасность, существующих и потенциально возможных в конкретных системах и комплексах программ. Следует ***формализовать их назначение, функции и основные характеристики***. На этой основе должны разрабатываться ***общие требования к безопасности и другим характеристикам качества ПП***, к обработанной информации для потребителей, адекватной назначению и функциям систем. ***Ограниченность ресурсов*** различных видов влияет на технико-экономические показатели, качество и функциональную безопасность всей системы и ПП. В результате сложность программ и баз данных, а также доступные ресурсы для их реализации становятся косвенными критериями или факторами, влияющими на выбор методов разработки, на достигаемое их качество.

Разработку систем и программных продуктов должны завершать ***комплексные испытания и удостоверение достигнутой функциональной безопасности и надежности*** систем с программными средствами, предусматривающие воз-

можность совершенствования их характеристик путем соответствующих корректировок программ. Повышение функциональной безопасности целесообразно путем анализа выявленных дефектов и ***оперативного восстановления вычислительного процесса, программ и данных (рестарта)*** после обнаружения аномалий и отказов функционирования программного продукта. Этому может способствовать накопление, мониторинг и хранение данных о выявленных дефектах, сбоях и отказах в процессе исполнения программ и обработки данных.

Основная цель книги – представить специалистам комплекс задач, методов и стандартов создания и развития систем, программных продуктов и баз данных с требуемой надежностью и функциональной безопасностью. Систематически изложены исходные данные, основные понятия и факторы, характеристики объектов и среды, для которых должна обеспечиваться высокая функциональная безопасность программных продуктов и систем, а также ресурсы необходимые для обеспечения безопасности. Внимание акцентировано на комплексе ***индустриальных методов и стандартов, которые непосредственно обеспечивают эффективный жизненный цикл сложных и безопасных систем и программных продуктов реального времени.***

Книга может быть полезна исполнителям опытно-конструкторских работ, студентам и аспирантам, связанным с созданием систем и программных продуктов высокого качества и безопасности.

Глава 1. Основные факторы, определяющие надежность и безопасность функционирования программных комплексов

1.1. Основные понятия и показатели надежности программных комплексов

Свойства надежности изделий изучаются ***теорией надежности***, которая является системой определенных идей, математических моделей и методов, направленных на решение проблем предсказания, оценки и оптимизации различных показателей надежности. Надежность технических систем определяется в основном двумя факторами: ***надежностью компонентов и дефектами в конструкции***, допущенными при проектировании или изготовлении. Относительно невысокая физическая надежность компонентов, их способность к разрушению, старению или снижению надежности в процессе эксплуатации привели к тому, что этот фактор оказался доминирующим для большинства комплексов аппаратуры. Этому способствовала также невысокая сложность многих технических систем, вследствие чего дефекты проектирования проявлялись относительно редко и вуалировались физическими отказами компонентов [1, 3].

Надежность сложных программных комплексов (ПК) определяется этими же факторами, однако доминирующими являются дефекты и ошибки проектирования, так как физическое хранение программ на магнитных носителях характеризуется очень высокой надежностью. Программа любой сложности и назначения при строго фиксированных исходных данных и абсолютно надежной аппаратуре исполняется по однозначно определенному маршруту и дает на выходе строго определенный результат. Однако случайное изменение исходных данных и накопленной при обработке информации, а также множество условных переходов в программе создают огромное число различных маршрутов исполнения каждого сложного ПК. Источниками ненадежности являются непроверенные сочетания исходных данных, при которых функционирующий ПК дает неверные результаты или отказы. В результате комплекс программ не соответствует требованиям функциональной пригодности и работоспособности.

При применении понятий надежности к программным продуктам следует учитывать ***особенности и отличия этих объектов от традиционных технических систем***, для которых первоначально разрабатывалась теория надежности:

- не для всех видов программ применимы понятия и методы теории надежности – их можно использовать только к программным комплек-

сам, функционирующим в реальном времени и непосредственно взаимодействующим с внешней средой;
- при разработке и оценке качества программных компонентов к ним не применимы понятия надежности функционирования, если при обработке информации они не используют значения реального времени и не взаимодействуют непосредственно с внешней средой;
- доминирующими факторами, определяющими надежность программ, являются дефекты и ошибки проектирования и разработки, и второстепенное значение имеет физическое разрушение программных компонентов при внешних воздействиях;
- относительно редкое разрушение программных компонентов и необходимость их физической замены, приводит к принципиальному изменению понятий сбоя и отказа программ и к разделению их по длительности восстановления относительно некоторого допустимого времени простоя для функционирования информационной системы;
- для повышения надежности комплексов программ особое значение имеют методы автоматического сокращения длительности восстановления и преобразования отказов в кратковременные сбои, путем введения в программные комплексы временной, программной и информационной избыточности;
- непредсказуемость места, времени и вероятности проявления дефектов и ошибок, а также их редкое обнаружение при реальной эксплуатации достаточно надежных программных продуктов, не позволяет эффективно использовать традиционные методы априорного расчета показателей надежности сложных систем, ориентированные на стабильные, измеряемые значения надежности составляющих компонентов;
- традиционные методы форсированных испытаний надежности систем путем физического воздействия на их компоненты не применимы для программных продуктов, и их следует заменять на методы форсированного воздействия информационных потоков внешней среды.

С учетом перечисленных особенностей применение основных понятий теории надежности сложных систем к жизненному циклу и оценке качества комплексов программ позволяет адаптировать и развивать эту теорию в особом направлении – ***надежности программных комплексов*** [1, 3, 12]. Предметом изучения теории надежности комплексов программ (Software Reliability) является работоспособность сложных программ обработки информации в реальном времени. ***К задачам теории и анализа надежности сложных программных продуктов можно отнести следующие:***

- формулирование основных понятий, используемых при исследовании и применении показателей надежности программных продуктов;
- выявление и исследование основных факторов, определяющих характеристики надежности сложных программных комплексов;

- выбор и обоснование критериев надежности для комплексов программ различного типа и назначения;
- исследование дефектов и ошибок, динамики их изменения при отладке и сопровождении, а также влияния на показатели надежности программных продуктов;
- исследование и разработка методов структурного построения сложных ПК, обеспечивающих их необходимую надежность;
- исследование методов и средств контроля и защиты от искажений программ, вычислительного процесса и данных путем использования различных видов избыточности и помехозащиты;
- разработка методов и средств определения и прогнозирования характеристик надежности в жизненном цикле комплексов программ с учетом их функционального назначения, сложности, структурного построения и технологии разработки.

Результаты решения этих задач являются основой для создания современных сложных программных продуктов с заданными показателями надежности. Использование и объединение результатов экспериментальных и теоретических исследований надежности ПП позволили заложить основы теории и методов в этой области. В жизненном цикле ПК значения показателей качества и надежности компонентов и комплексов программ в целом рекомендуется анализировать и прогнозировать с целью гарантированного обеспечения заданных показателей надежности. ***В реальных проектах работы по исследованию и обеспечению надежности программ целесообразно выделять в отдельную группу под единым руководством со специальным планом.***

В основе теории надежности лежат понятия о двух возможных состояниях объекта или системы: работоспособном и неработоспособном. ***Работоспособным*** называется такое состояние объекта, при котором он способен выполнять заданные функции с параметрами, установленными технической документации. В процессе функционирования возможен переход объекта из работоспособного состояния в не работоспособное и обратно. С этим переходами связаны события отказа и восстановления.

Определение степени работоспособности системы предполагает наличие в ней средств, способных установить соответствие ее характеристик требованиям технической документации. Для этого должны использоваться ***методы и средства контроля и диагностики функционирования системы***. Глубина и полнота проверок, степень автоматизации контрольных операций, длительность и порядок их выполнения – влияют на работоспособность системы и достоверность ее оценки. Методы и средства диагностического контроля предназначены для установления степени работоспособности системы, локализации отказов, определения их характеристик и причин, и скорейшего восстановления работоспособности, для накопления, обобщения и анализа данных, характеризующих работоспособность системы. Диагноз состояния системы принято

делить на ***тестовый и функциональный***. При тестовом диагнозе используются специально подготовленные исходные данные и эталонные результаты, которые позволяют проверять работоспособность определенных компонентов системы. Функциональный диагноз организуется на базе реальных исходных данных, поступающих в систему при ее использовании по прямому назначению. ***Основные задачи технической диагностики включают:***

- контроль исправности системы и полного соответствия ее состояния и функций технической документации;
- проверку работоспособности системы и возможности выполнения всех функций в заданном режиме работы в любой момент времени с характеристиками, заданными технической документацией;
- поиск, выявление и локализацию источников и результатов сбоев, отказов и неисправностей в системе.

Формализации показателей качества программных комплексов посвящена группа нормативных документов. В международном стандарте **ISO 9126:1-4:2002**, при отборе минимума стандартизируемых показателей выдвигались и учитывались следующие принципы: ясность и измеряемость значений, отсутствие перекрытия между используемыми показателями, соответствие установившимся понятиям и терминологии, возможность последующего уточнения и детализации. Выделены характеристики, которые позволяют оценивать ПП с позиции пользователя, разработчика и управляющего проектом.

Функциональная пригодность – это набор атрибутов, определяющий назначение, номенклатуру, основные необходимые и достаточные функции ПП, заданные техническим заданием заказчика или потенциального пользователя. В процессе проектирования ПК атрибуты функциональной пригодности конкретизируются в спецификации на компоненты. Эти атрибуты можно численно представить точностью вычислений, относительным числом поэтапно изменяемых функций, числом спецификаций требований заказчиков и т.д. Кроме них функциональную пригодность отражают множество различных специализированных критериев, которые тесно связаны с конкретными функциями программ. Их можно рассматривать как частные критерии или как факторы, влияющие на основные показатели. В наиболее общем виде функциональная пригодность проявляется в ***корректности и надежности*** ПП.

Понятие ***корректной (правильной) программы*** может рассматриваться статически вне ее исполнения во времени. Корректность программы не определена вне области изменения исходных данных, заданных требованиями спецификации, и не зависит от динамики функционирования программы в реальном времени. Степень некорректности программ определяется вероятностью попадания реальных исходных данных в область, которая задана требованиями спецификации и технического задания (ТЗ), однако не была проверена при испытаниях. Значения этого показателя зависят от функциональной корректности применяемых компонентов и могут рассматриваться в зависимости от методов их достижения и оценивания: детерминировано, стохастически и в реальном

времени. При анализе видов корректности и способов их измерения, естественно, они связываются с видами и методами процесса тестирования и испытания программ.

Надежная программа – прежде всего, должна обеспечивать достаточно низкую вероятность отказа в процессе функционирования в реальном времени. Быстрое реагирование на искажения программ, данных или вычислительного процесса и восстановление работоспособности за время меньшее, чем порог между сбоем и отказом, обеспечивают высокую надежность программ. При этом не корректная программ может функционировать абсолютно надежно. В реальных условиях по различным причинам исходные данные могут попадать в области значений, вызывающих сбои, не проверенные при испытаниях, а также не заданные требованиями спецификации и технического задания. Если в этих ситуациях происходит достаточно быстрое восстановление, такое что не фиксируется отказ, то такие события не влияют на основные показатели надежности – наработку на отказ и коэффициент готовности. Следовательно, ***надежность функционирования программ является понятием динамическим, проявляющимся во времени и существенно отличается от понятия корректности программ.***

При оценке качества программ по показателям надежности регистрируются только такие искажения в процессе динамического тестирования с исполнением программ, которые приводят к потере работоспособности ПП или их крупных компонентов. Первопричиной нарушения работоспособности программ при безотказности аппаратуры всегда является ***конфликт между реальными исходными данными, подлежащими обработке, и программой, осуществляющей эту обработку***. Работоспособность ПП можно гарантировать при конкретных исходных данных, которые использовались при отладке и испытаниях. Реальные исходные данные могут иметь значения, отличающиеся от заданных техническим заданием и от использованных при применении программ. При таких исходных данных функционирование программ трудно предсказать заранее, и весьма вероятны различные аномалии, завершающиеся отказами.

Непредсказуемость вида, места и времени проявления дефектов ПП в процессе эксплуатации приводит к необходимости создания специальных, дополнительных систем оперативной защиты от непредумышленных, случайных искажений вычислительного процесса, программ и данных. Системы оперативной защиты предназначены для выявления и блокирования распространения негативных последствий проявления дефектов и уменьшения их влияния на надежность функционирования ПП до устранения их первичных источников. Для этого в ПП должна вводиться временная, программная и информационная избыточность, осуществляющая оперативное обнаружение дефектов функционирования, их идентификацию и автоматическое восстановление (рестарт) нормального функционирования ПП**.** Надежность ПП должна повышаться за счет средств обеспечения помехоустойчивости, оперативного контроля и восстановления функционирования программ и баз данных. Эффективность

такой защиты зависит от используемых методов, их применения и выделяемых вычислительных ресурсов на их реализацию.

Основным ***принципом классификации сбоев и отказов*** в программах при отсутствии их физического разрушения является разделение ***по временному показателю длительности восстановления после любого искажения программ, данных или вычислительного процесса***, регистрируемого как нарушение работоспособности. При длительности восстановления, меньшей заданного порога, дефекты и аномалии при функционировании программ следует относить к сбоям, а при восстановлении, превышающем по длительности пороговое значение, происходящее искажение соответствует ***отказу***. Классификация программных сбоев и отказов по длительности восстановления приводит к необходимости анализа динамических характеристик абонентов, являющихся потребителями данных, обработанных исследуемым ПП, а также временных характеристик функционирования программ. Временная зона перерыва нормальной выдачи информации и потери работоспособности, которую следует рассматривать как зону сбоя, тем шире, чем более инертный объект находится под воздействием сообщений, подготовленных данным ПП. Пороговое время восстановления работоспособного состояния системы, при превышении которого следует фиксировать отказ, близко к периоду решения задач для подготовки информации соответствующему абоненту.

При нормальном темпе решения задач и выдаче их результатов потребителю, отклонения его характеристик от траектории, рассчитываемой ПП, находятся в допустимых пределах. Для любого потребителя информации существует допустимое время, отсутствия данных от ПП, при котором его характеристики, изменяясь по инерции, достигают предельного отклонения от значения, которое должно было бы быть рассчитано программами. Соответствующая этому отклонению временная зона перерыва выдачи информации потребителю позволяет установить ***границу допустимой длительности нарушения работоспособности, которая разделяет зоны сбоев и отказов***.

Чем более инерционным, является потребитель информации, тем больше может быть время отсутствия у него результатов функционирования и воздействий от ПП без катастрофических последствий нарушения работоспособности, соответствующего отказу. Это ***допустимое отклонение результатов после перерыва функционирования ПП зависит, в основном, от динамических характеристик источников и потребителей информации.*** Таким образом, установив в результате системного анализа динамических характеристик объектов системы величину порогового значения, можно определить интервал времени функционирования ПП при отсутствии выдачи данных потребителю, которые разделяют события сбоя и отказа без физического разрушения программ.

Надежность функционирования ПП наиболее широко характеризуется ***устойчивостью*** или способностью к безотказному функционированию и ***восстанавливаемостью*** работоспособного состояния после произошедших сбоев или отказов. В свою очередь устойчивость зависит от уровня не устраненных де-

фектов и ошибок и способности ПП реагировать на их проявления так, чтобы это не отражалось на показателях надежности. Последнее определяется эффективностью контроля данных, поступающих из внешней среды, и средств обнаружения аномалий функционирования ПП.

Восстанавливаемость – характеризуется полнотой и длительностью восстановления функционирования программ в процессе перезапуска - рестарта. Перезапуск должен обеспечивать возобновления нормального функционирования ПП, на что требуются ресурсы ЭВМ и время. Поэтому ***полнота и длительность восстановления функционирования после сбоев*** отражают качество и надежность ПП и возможность его использования по прямому назначению.

Показатели надежности ПП в значительной степени адекватны аналогичным характеристикам, принятым для других технических систем. Наиболее широко используется ***критерий длительности наработки на отказ***. Для определения этой величины измеряется время работоспособного состояния системы между последовательными отказами или началами нормального функционирования системы после них. Вероятностные характеристики этой величины в нескольких формах используются как ***разновидности критериев надежности***. Критерий надежности восстанавливаемых систем учитывает возможность многократных отказов и восстановлений. Для оценки надежности таких систем, которыми чаще всего являются сложные ПК, кроме вероятностных характеристик наработки на отказ, важную роль играют характеристики функционирования после отказа в процессе восстановления.

Основным показателем процесса восстановления является ***длительность восстановления*** и ее вероятностные характеристики. Этот критерий учитывает возможность многократных отказов и восстановлений. Обобщение характеристик отказов и восстановлений производится в критерии ***коэффициент готовности***. Этот показатель отражает вероятность иметь восстанавливаемую систему в работоспособном состоянии в произвольный момент времени. Значение коэффициента готовности соответствует доле времени полезной работы системы на достаточно большом интервале, содержащем отказы и восстановления.

Наработка на отказ учитывает ситуации потери работоспособности, когда длительность восстановления достаточно велика и превышает пороговое значение времени, разделяющее события сбоя и отказа. При этом большое значение имеют средства оперативного контроля и восстановления (рестарта). Качество проведенной отладки программ более полно отражает значение длительности между потерями работоспособности программ – ***наработка на отказовую ситуацию или устойчивость***, независимо от того, насколько быстро произошло восстановление. Средства оперативного контроля и восстановления не влияют на наработку на отказовую ситуацию, однако могут значительно улучшать показатели надежности программ. Поэтому при оценке необходимой отладки целесообразно измерять и контролировать наработку на отказовую ситуацию, а объем и длительность тестирования в ряде случаев устанавливать по наработке на отказ, с учетом эффективности средств рестарта.

В реальных системах достигаемая при отладке и испытаниях наработка на отказ ПП обычно должна быть соизмерима с наработкой на отказ аппаратуры, на которой исполняется программа. Для систем обработки информации и управления в реальном времени наработка на отказ программ измеряется десятками и сотнями часов, а для особо важных или широко тиражируемых систем может достигать десятков тысяч часов. При достаточно развитом программном оперативном контроле и восстановлении, наработка на отказовую ситуацию может быть на один-два порядка меньше, чем наработка на отказ. Реально очень трудно достичь наработку на отказовую ситуацию, на уровне сотен часов и поэтому для получения высокой надежности программ невозможно ограничиваться тестированием и отладкой без использования средств рестарта. Априори невозможно обеспечить абсолютное отсутствие дефектов проектирования в сложных ПП, вследствие чего ***надежность их функционирования имеет всегда конечное, ограниченное значение***.

Непредусмотренные при проектировании ситуации и ошибки функционирования программ и данных могут быть потенциальными источниками катастроф при применении таких ПП, ***влияющими на безопасность их функционирования***. Наиболее полно функциональная безопасность ПП характеризуется величиной ущерба, возможного при проявлении дестабилизирующих факторов и реализации конкретных угроз, а также средним временем между проявлениями непредумышленных угроз, нарушающих надежность и безопасность. Однако описать и измерить в достаточно общем виде возможный ущерб при нарушении безопасности для критических ПП разных классов практически невозможно. Поэтому реализации угроз целесообразно характеризовать интервалами времени между их проявлениями, нарушающими безопасность применения ПП, или наработкой на отказы, отражающиеся на безопасности. Это сближает понятия и характеристики функциональной безопасности с показателями надежности ПК. Различие состоит в том, что в показателях надежности учитываются все реализации отказов, а в характеристиках функциональной безопасности следует регистрировать только те случайные катастрофические отказы, которые отразились на безопасности. Статистически таких отказов может быть в несколько раз меньше, чем учитываемых в значениях надежности. Однако методы, влияющие факторы и реальные значения показателей надежности ПК могут служить ориентирами при оценке безопасности критических ПП.

Процесс отладки программ происходит во времени, и его динамические характеристики могут служить частными конструктивными критериями для оценки достигнутого качества программ [9.14]. Одним из таких показателей может служить ***интенсивность обнаружения ошибок***, или число ошибок, выявляемых в программах в процессе отладки за единицу времени при постоянных усилиях на его проведение. В этом критерии отражаются ошибки, приводящие к нарушению работоспособности программ, и дефекты, искажающие результаты, но не влияющие на надежность функционирования программ. В этом случае для интенсивности тестирования характерна положительная обратная

связь – ***чем больше выявляется ошибок в программе на некотором интервале времени, тем шире должно быть варьирование тестовых данных и больше тестов***. По мере устранения ошибок в программе частота их обнаружения снижается и специалисты, осуществляющие отладку, попадают в область низкого темпа обнаружения ошибок, при котором завершается отладка.

Применение основных понятий теории надежности для оценки сложных ПП позволяет получить ряд четких хорошо измеряемых интегральных показателей качества программ. Приведенные критерии используются в основном при испытании ПК и на завершающих фазах комплексной отладки. Их практически невозможно использовать для оценки качества программных модулей и групп программ, решающих частные функциональные задачи вне реального времени и без непосредственного взаимодействия с внешней средой.

1.2. Основные понятия и факторы, определяющие функциональную безопасность программных продуктов

Любые программные комплексы, прежде всего, должны иметь экономическую, техническую, научную или социальную ***эффективность применения***, которая в проектах должна отражать основную ***цель их жизненного цикла*** в системе. Эта ***системная эффективность*** может быть описана количественно или качественно, в виде набора полезных свойств и характеристик программ, их отличий от имеющихся у других комплексов программ, а также факторов и источников возможной эффективности. В результате должна быть формализована цель использования и набор требований заказчика и пользователей при создании или приобретении ПП, а также предполагаемое назначение и сфера применения [1, 5, 10].

Системная эффективность применения программных продуктов определяется степенью удовлетворения потребностей определенных лиц: заказчиков и/или пользователей. Их желательно измерять экономическими категориями: прибылью, стоимостью, трудоемкостью, предотвращенным ущербом, длительностью применения. В стандартах эта эффективность отражается основной, обобщенной характеристикой качества – ***функциональной пригодностью***. Данная характеристика связана с тем, ***какие*** функции и задачи решает комплекс программ для удовлетворения потребностей пользователей, в то время как другие, конструктивные характеристики (в том числе безопасность) главным образом связаны с тем, ***как и при каких условиях,*** заданные функции могут выполняться с требуемым качеством. Номенклатура и значения всех показателей качества непосредственно определяются требуемыми функциями программного продукта и, в той или иной степени, влияют на выполнение этих функций. Поэтому выбор и формализация функциональной пригодности, подробное и корректное описание ее свойств являются основными исходными данными для установления при проектировании требуемых значений всех остальных стандартизированных показателей качества.

Стандартизированные в **ISO 9126:1-4:2002** и уточненные в **ISO 25010:2011** см. Приложение 1) характеристики качества ПП различаются по степени влияния на системную эффективность применения комплексов программ по прямому назначению. Высшие приоритеты естественно, должны присваиваться свойствам и атрибутам функциональной пригодности, необходимым для достижения основных ***стратегических целей*** использования ПП. Все остальные стандартизированные характеристики должны способствовать обеспечению ***тактических целей*** выбранных требований. Решение этих задач должно быть направлено на обеспечение высокой функциональной пригодности ПП ***путем сбалансированного улучшения остальных, характеристик качества (надежности и безопасности) в условиях ограниченных ресурсов на ЖЦ***. Для этого в процессе системного анализа при подготовке технического задания и требований спецификаций, значения различных факторов, характеристик качества и безопасности, должны выбираться с учетом их влияния на функциональную пригодность.

Улучшение каждой, вспомогательной – ***конструктивной характеристики качества***, требует некоторых затрат ресурсов (трудоемкости, финансов, времени), которые в той или иной степени должны отражаться на основной характеристике качества – на функциональной пригодности. В зависимости от назначения и функций комплекса программ почти каждая из конструктивных характеристик может стать доминирующей или даже почти полностью определяющей функциональную пригодность ПП. В процессе системного анализа и проектирования должны быть выявлены потенциальные предумышленные и случайные угрозы функционированию ПП и установлен ***необходимый уровень надежности и безопасности*** данного комплекса программ. В соответствие с этим уровнем, заказчиком и разработчиками должны выбираться и устанавливаться требуемые и необходимые наборы методов, свойств и средств обеспечения безопасности ПП с учетом ограниченных ресурсов на их реализацию. В результате сформированные требования должны обеспечивать равнопрочную защиту от различных, реальных угроз и реализацию необходимых мер контроля и подтверждения характеристик качества и функциональной пригодности комплекса программ.

Для обеспечения эффективности системы, ***комплекс программ, связанный с безопасностью целесообразно базировать на следующих общих принципах***:

- защита аппаратуры системы, функциональных программ и данных, должна быть комплексной и многоуровневой, ориентированной на все виды угроз, с учетом их опасностей для потребителя;
- стоимость (трудоемкость) создания и эксплуатации системы программной защиты должна быть меньше, чем размеры наиболее вероятного или возможного (в среднем), неприемлемого потребителям системы, ущерба – риска от любых потенциальных угроз;
- комплекс программ защиты должен иметь целевые, индивидуальные компоненты контрмер, предназначенные для обеспечения безопасно-

сти функционирования каждого отдельно взятого компонента и функциональной задачи системы с учетом их уязвимости и степени влияния на безопасность системы в целом;

- система программ защиты не должна приводить к ощутимым трудностям, помехам и снижению эффективности применения и решения основных, функциональных задач пользователями системы в целом.

Характеристики внешней среды, прикладные сферы применения комплексов программ, цели и задачи пользователей, уровень автоматизации их функций и многие другие факторы определяют методы и свойства средств обеспечения безопасности вычислительных систем. При последующем анализе внимание акцентируется на определении требований и применении сложных аппаратно-программных систем, и на их ***функциональной безопасности***. Соответственно ущерб при отказовых ситуациях определяется уязвимостью и нарушением корректного выполнения системой основного назначения и требуемых функций при ограниченных ресурсах на их реализацию.

Функции систем и их программных средств реального времени реализуются в определенной аппаратной, операционной и пользовательской внешней среде, характеристики, которых существенно влияют на функциональную пригодность программ. Для выполнения требуемых функций комплекса программ необходима ***адекватная исходная информация от объектов внешней среды***, содержание которой должно полностью обеспечивать реализацию функций, декларированных в требованиях к системе. Полнота формализации номенклатуры, структуры и качества входной информации для выполнения требуемых функций, является одной из важных составляющих при определении функциональной пригодности ПП в соответствующей внешней среде. Так как без дефектов и ошибок, принципиально невозможно создать и применять сложные комплексы программ, эти проблемы должны изучаться, и обобщаться в некоторую совокупность знаний. В ней следует оценивать возможные свойства, содержание и характеристики дефектов и ошибок в программах, их проявления и негативных последствий для потребителя, угрозы и ущерб для безопасности функционирования системы в целом.

Среда обеспечения функциональной безопасности программ включает политики и Программы организации безопасности предприятий и систем, опыт, специальные навыки и знания, определяющие контекст, предполагаемого применения системы. Среда включает также возможные угрозы безопасности, присутствие которых в этой среде установлено или предполагается. ***При формализации среды функциональной безопасности*** следует принимать во внимание:

- предназначение системы и комплекса программ, включая функции программного продукта и предполагаемую сферу его применения;
- активы – программы и данные, основных, функциональных задач системы, к которым применяются политики безопасности, а также ком-

поненты, которые косвенно подчинены требованиям безопасности реализации системы;

- физическую среду в той ее части, которая определяет все аспекты эксплуатационной среды системы, включая мероприятия, относящиеся к средствам физической защиты и к персоналу.

На основании разработанных политик безопасности, оценок угроз и рисков следует сформировать исходные данные, ***относящиеся к функциональной безопасности среды системы и основного комплекса программ***:

- предположения, которым должна удовлетворять среда для того, чтобы система или ПП считались безопасными;
- угрозы безопасности для активов, в которых были бы идентифицированы все угрозы среды, относящиеся к объекту функциональной безопасности;
- угрозы, которые раскрываются через понятия источника угроз, предполагаемого метода их реализации, предпосылки для отказов, и идентификация компонентов, которые являются объектами отказов;
- политика безопасности, в которой были бы достаточно точно идентифицированы и описаны цели, методы и правила реализации функциональной безопасности системы.

Результаты анализа среды безопасности могут использоваться для формулирования ***целей обеспечения функциональной безопасности системы***, которые направлены на противостояние установленным угрозам, а также проистекают из политики безопасности предприятия или проекта и сделанных предположений. Необходимо, чтобы общие цели безопасности системы были согласованы с определенными ранее целями применения и характеристиками функциональной пригодности системы или программ как продукта, а также со всеми известными сведениями о внешней среде системы.

Основные понятия в области функциональной безопасности систем и программных продуктов, характеризуются факторами, связанными с возникновением опасных состояний, приводящих к ***потере работоспособности***. Если эти события не обнаруживаются и не устраняются специально введенными в состав системы средствами обеспечения безопасности, то возникают опасные отказы и их последствия. Систему, в состав которой требуется вводить функции безопасности принято называть ***связанной с безопасностью***. Эти функции необходимы для достижения или поддержания безопасного состояния объекта управления или обработки информации – самостоятельно или совместно с другими, ***связанными с безопасностью системами***, а также с внешними средствами снижения риска для предотвращения или смягчения последствий опасного события.

Термины, используемые далее и связанные с функциональной безопасностью систем и ПП аналогичны тем, которые содержатся в стандартах **IEC 61508, ISO 15408** и др. (см. Приложение 1). Эти определения базируются на

понятии отказа как события, заключающегося в нарушении работоспособного состояния объекта. ***Определение отказа является базовым для оценки функциональной безопасности технических систем*** [14, 24, 25]***:***

- ***исправное состояние*** – при котором обеспечиваются ***все*** требования к системе, нормативно-технической и/или конструкторской и проектной документацией;
- ***неисправное состояние,*** – при котором не обеспечивается выполнение хотя бы одного требования нормативно - технической и/или конструкторской и проектной документации;
- ***работоспособное состояние,*** – при котором значения всех параметров, характеризующих способность выполнять заданные функции, соответствуют требованиям нормативно - технической и/или конструкторской и проектной документации;
- ***неработоспособное состояние,*** – при котором значение хотя бы одного параметра, характеризующего способность выполнять заданные функции, не соответствует требованиям нормативно - технической и/или конструкторской и проектной документации;
- ***защитное состояние,*** – при котором выполняются предусмотренные функции обеспечения безопасности в результате обнаружения неработоспособного состояния, хотя бы одного компонента системы и производится восстановление работоспособности отказавшего компонента;
- ***отказовая ситуация*** – скрытый отказ, не обнаруживаемый визуально или штатными методами и средствами контроля и диагностирования, но выявляемый средствами автоматизированного рестарта, а также при проведении технического обслуживания или специальными методами диагностики, который потенциально может превратиться в отказ;
- ***опасный отказ*** – событие, заключающееся в нарушении работоспособного или защитного состояния с большим ущербом;
- ***дефекты аппаратуры, программы или данных*** – негативные события, заключающиеся в непреднамеренном отклонении от требований спецификации или документации в процессах их жизненного цикла;
- ***ошибки аппаратуры, программ или данных*** – случайные, непредсказуемые искажения компонентов или комплекса программ, проявляющиеся в процессах их анализа или функционирования.

Для оценивания свойств функциональной безопасности систем и комплекса программ необходимо регистрировать, измерять, обобщать и упорядочивать реальные характеристики отказовых ситуаций и их последствия для безопасности. ***По методам и ресурсам***, необходимым для достижения различных значений безопасности в стандарте **IEC 61508** и в ряде других стандартов (**ГОСТ**

Р 51901), рекомендуется выделять четыре дискретных уровня, которые, называются ***уровнями полноты безопасности (УПБ)***:

- УПБ 4: самый высокий и, трудно достижимый уровень полноты безопасности, который требует использования высочайших приемов и технологий при формализации требований к ПП;
- УПБ 3: легче достижимый, чем уровень УПБ 4, но также требует высокой технологии разработки системы и ПП;
- УПБ 2: требует хорошего современного проектирования, разработки и практики применения ПП на уровне, не ниже требований стандарта ISO 9001;
- УПБ 1: самый низкий уровень, однако, также требующий использования современной технологии и опыта разработок, но в стандарте МЭК 61508 и в других документах он фигурирует как «не связанный с безопасностью».

С той же целью в стандарте **ГОСТ Р 51904** рекомендуются ***категории отказовых ситуаций систем и ПП***, в зависимости от серьезности их влияния на качество функционирования программ или данных объекта управления. Необходимый для безопасного функционирования уровень качества ПП рекомендуется определять исходя из ***опасности*** отказовых ситуаций и возможного при этом ***ущерба*** для программ, системы и потребителя. Стандартом установлена следующая классификация отказовых ситуаций:

- ***категория А – катастрофическая***: отказовая ситуация, которая препятствует полной работоспособности и функционированию системы или ПП в соответствии с требованиями, и способна нанести недопустимый по последствиям и величине ущерб системе или пользователям;
- ***категория В – опасная/критическая***: отказовая ситуация, которая приводит к значительному снижению работоспособности, возможностей применения и функционирования системы или к отсутствию способности персонала справиться с неблагоприятными эксплуатационными режимами, при которых возникают:
  - крайне тяжелые ситуации или перегрузки системы, которые могут вызвать неточное или неполное выполнение основных функциональных задач с большим ущербом;
  - недопустимо большое снижение характеристик функциональной пригодности, реализуемых функций и конструктивных характеристик качества системы или ПП;
  - неблагоприятные или потенциально фатальные воздействия системы или ПП для окружающей внешней среды;
- ***категория С – существенная***: отказовая ситуация, которая приводит к снижению возможностей применения и функциональной пригодно-

сти системы или к сокращению способности персонала справиться с неблагоприятными эксплуатационными режимами

- ***категория D – несущественная***: отказовая ситуация, незначительно уменьшающая безопасность функционирования и применения объекта, но отражающаяся на его надежности, или требующая действий персонала, которые осуществимы в пределах их возможностей для восстановления нормальной работоспособности;
- ***категория E – не влияющая***: отказовая ситуация, которая практически не воздействует на работоспособность, эксплуатационные характеристики и возможности объекта или не увеличивает рабочую нагрузку персонала.

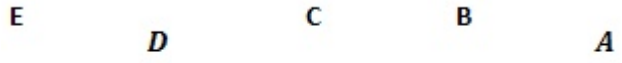

Рис.1.1.

Эти пять категорий отказовых ситуаций отражают различную степень их влияния на качество функционирования систем или ПП, которое целесообразно распределить между понятиями ***безопасность*** и ***надежность*** в зависимости от возможного ущерба – риска при отказах [5, 26, 28]. Первые две категории ситуаций, характеризуются большими значениями ущерба при отказах, и их следует рассматривать с позиции классов безопасности функционирования систем. Две последние категории практически не влияют на безопасность применения объектов, но отражаются на надежности их функционирования с относительно невысоким ущербом. Такие отказовые ситуации не целесообразно рассматривать с позиции безопасности. К наиболее сложному случаю относится третья – существенная категория отказовых ситуаций. Она требует детального и конкретного анализа функциональных характеристик системы, ПП и отказов, которые могут отражаться не только на надежности, но и в той или иной степени влиять на безопасность функционирования системы.

Можно предположить для простоты, что уровни функциональной безопасности и надежности, линейно зависят от категории отказов, и несколько отличаются для различных типов программных продуктов. Эти зависимости можно интерпретировать, как снижение работоспособности системы или ПП в результате проявления некоторого ущерба. Ущерб при отказах проявляется либо в снижении надежности при функционировании, либо, в худшем случае, как нарушение функциональной безопасности. Таким образом, категории отказовых ситуаций, величина ущерба – риска, трудоемкость и длительность восстановления нормального функционирования системы, могут рассматриваться как база для оценивания и категоризации ***пороговых уровней*** в требованиях к безопасности и/или надежности функционирования и применения систем и их программных продуктов.

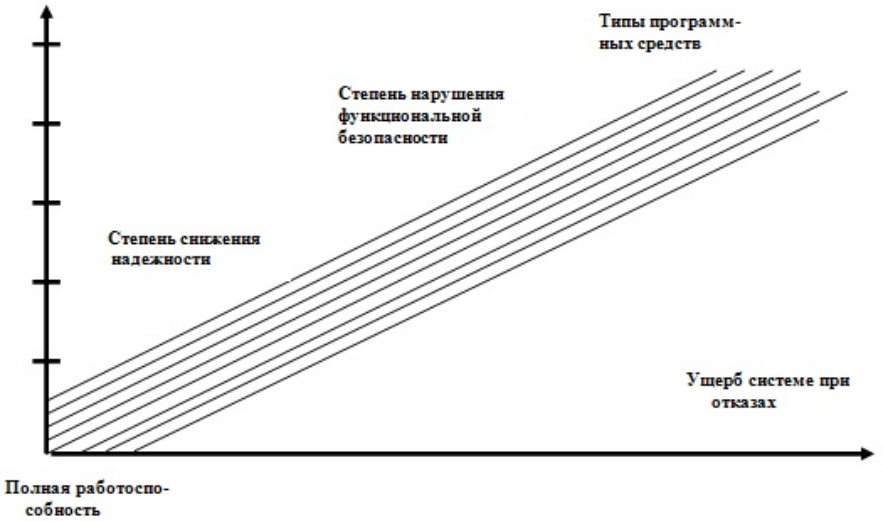

Рис.1.2

Наиболее полно ***степень функциональной безопасности систем характеризуется величиной предотвращенного или остаточного ущерба***, возможного при проявлении дестабилизирующих факторов и реализации конкретных угроз безопасности применения системы и программного продукта пользователями. С этой позиции затраты ресурсов разработчиками и заказчиками на обеспечение безопасности системы должны быть соизмеримыми с возможным средним ущербом у пользователей от нарушения безопасности. Однако описать и измерить в достаточно общем виде, возможный ущерб при нарушении безопасности для критических ПП разных классов весьма сложно. Перечисленные выше отказовые ситуации и уровни функциональной безопасности могут быть следствием различных, негативных явлений в системе, которые в основном оцениваются:

- величиной экономического, материального или социального ущерба системе вследствие возникновения отказовой ситуации;
- трудоемкостью устранения последствий отказовой ситуации и восстановления нормальной работоспособности системы в соответствии с требованиями после отказа;
- длительностью неработоспособного состояния системы вследствие каждого отказа до полного восстановления работоспособности;
- затратами ресурсов и времени, необходимыми после отказа для восстановления нормальной работоспособности системы в соответствии с требованиями.

Значения функциональной безопасности и надежности ***коррелированны с характеристикой корректность ПП***, однако, можно достигать высокую безопасность функционирования программ при относительно невысокой их корректности ***за счет автоматизации оперативного сокращения ущерба и времени восстановления при отказах***. Степень покрытия тестами структуры функциональных компонентов и комплекса программ в целом при разработке и испытаниях может служить ориентиром для прогнозирования их потенциальной безопасности. Распределение реальных длительностей и эффективности восстановления при ограниченных ресурсах для функционирования программ, может рассматриваться как дополнительная составляющая при оценивании функциональной безопасности.

В требованиях к каждой конкретной системе должны быть установлены заказчиком совместно с пользователем, ***пороговые значения***, разделяющие виды и величины ущерба, а также ***характеристики восстановления***, которые следует относить, либо к нарушению функциональной безопасности, либо только к снижению надежности. Эти пороговые значения зависят от назначения и базовых характеристик, отражающих функциональную пригодность конкретной системы или программного продукта. При малом (ниже порога) ущербе вследствие отказовой ситуации и быстром восстановлении такие события следует относить к ***снижению надежности***. Если последствия отказовых ситуаций – какой либо вид ущерба или характеристики восстановления превышают заданный критический порог, то такие события относятся к ***нарушению функциональной безопасности***. При этом в ряде ситуаций для нарушения безопасности может быть достаточным нарушение порогового значения одним из видов отказа. Если аномальное поведение компонента или системы может быть вызвано более чем одной отказовой ситуацией, уровень безопасности ПП следует определять по наиболее серьезной категории отказовой ситуации.

Внутренними источниками угроз безопасности функционирования системы и ПП являются:

- системные ошибки при постановке целей и задач проектирования функциональной пригодности ПП и системы, при формулировке требований к функциям и характеристикам средств обеспечения безопасности решения задач;
- дефекты и ошибки при определении функций, условий и параметров внешней среды, в которой предстоит применять защищаемые программные средства и систему;
- алгоритмические ошибки проектирования при непосредственной алгоритмизации функций обеспечения безопасности аппаратуры, программных средств и баз данных, при определении структуры и взаимодействия компонентов функциональных комплексов программ, а также при использовании информации баз данных;

- ошибки и дефекты программирования в текстах программ и описаниях данных, а также в исходной и результирующей документации на компоненты и комплекс программ;
- недостаточная эффективность используемых методов и средств оперативной защиты программ и данных, обеспечения безопасности функционирования и восстановления работоспособности системы в условиях случайных и предумышленных негативных воздействий от внешней среды.

Полное устранение перечисленных угроз безопасности функционирования критических систем принципиально невозможно. При создании сложных комплексов программ проблема состоит в выявлении факторов, от которых они зависят, в создании методов и средств уменьшения их влияния на безопасность. Необходимо ***оценивать уязвимость*** функциональных компонентов системы для различных, негативных воздействий и степень их влияния на основные характеристики качества и безопасности, а также на суммарный риск. В зависимости от этого следует ***распределять ресурсы*** для создания системы и ее компонентов, равнопрочных по безопасности функционирования при любых негативных внешних воздействиях. В результате должны быть сформулированы соответствующие методы и контрмеры, которые, в свою очередь, определяют необходимые функции и механизмы средств обеспечения работоспособности и безопасности.

Для обеспечения функциональной безопасности систем создаются ***соответствующие контрмеры*** – специализированные системы и средства, которые включают совокупность взаимосвязанных нормативных документов, организационно-технических мероприятий и соответствующих им методов и программных компонентов, предназначенных для предупреждения и/или ликвидации негативных последствий отказовых ситуаций, угроз безопасности, их выявления и локализации. Создание таких комплексов повышения безопасности предусматривает ***планирование и реализацию целенаправленной политики*** комплексного обеспечения функциональной безопасности системы, а также ***эффективное распределение ресурсов на контрмеры и средства обеспечения безопасности***. Разработчики и заказчики должны анализировать возможные угрозы, чтобы решать, какие из них действительно присущи внешней среде, системе и программному продукту.

Для прямых, количественных измерений функциональной безопасности необходимы ***инструментальные средства, встроенные в операционную систему или в соответствующие компоненты и функции ПП***. Эти средства должны в динамике реального функционирования, автоматически селектировать и регистрировать отказовые ситуации, дефекты и искажения вычислительного процесса программ и данных, выявляемые аппаратным, программно-алгоритмическим контролем или пользователями. Накопление и систематизация проявлений отказов при исполнении программ позволяет оценивать основные показатели безопасности, помогает определять причины сбоев и отка-

зов и подготавливать данные для улучшения функциональной безопасности ПП. Регулярная регистрация и обобщение таких данных способствует устранению ситуаций, негативно влияющих на функциональную пригодность и другие базовые характеристики программных продуктов.

Доступная величина и распределение ресурсов компьютеров на отдельные виды контрмер оказывают значительное влияние на достигаемую комплексную безопасность системы. Наиболее общим видом ресурсов, который приходится учитывать при проектировании, являются ***допустимые финансово-экономические затраты или трудоемкость разработки и функционирования комплекса средств, обеспечивающих безопасность системы***. Для размещения этих средств в компьютере реального времени, при проектировании должна быть ***предусмотрена программная и информационная избыточности*** в виде ресурсов внешней и внутренней памяти компьютера. Кроме того, для их функционирования необходима временная избыточность – дополнительная производительность компьютера. Эти ***виды вычислительных ресурсов при обеспечении функциональной безопасности*** используются также для:

- контроля и корректировки искажений информации, поступающей от внешней среды и источников данных;
- оперативного контроля и обнаружения дефектов исполнения программ и обработки данных при использовании ПП по основному назначению;
- размещения и обеспечения функционирования применяемых средств защиты от всех видов угроз безопасности системы;
- генерации тестовых наборов или хранения тестов для контроля работоспособности, сохранности и целостности ПП при функционировании системы;
- накопления, хранения и мониторинга данных о выявленных инцидентах, о дефектах, сбоях и отказах в процессе исполнения программ и обработки данных, влияющих на безопасность;
- реализации процедур анализа и мониторинга выявленных дефектов и оперативного восстановления вычислительного процесса, программ и данных (рестарта) после обнаружения дефектов и отказов функционирования системы.

Потребителя-заказчика, прежде всего, интересуют ***функции, безопасность и качество готового конечного продукта*** – системы и программного комплекса, и обычно не очень беспокоит, как они достигнуты. Требуемую функциональную безопасность при разработке системы и ПП, как и любой продукции, можно обеспечить ***двумя методами***:

- путем использования только заключительного, выходного контроля и ***испытаний*** готовых объектов и исключения из поставки или направлением на доработку программных продуктов, не соответствующих требуемым безопасности и качеству;
- посредством применения регламентированных технологий и систем обеспечения функциональной безопасности и качества в процессах проектирования, разработки и изготовления, предотвращающих дефекты и гарантирующих высокую безопасность и качество продукции во время ее создания и/или модификации.

Первый метод может приводить к значительным экономическим потерям за счет затрат на создание части не пригодного к использованию брака, что может быть очень дорого для сложных систем. Достижение необходимой безопасности за счет только выходного контроля и испытаний, при отсутствии адекватной технологии и системы обеспечения качества в процессе разработки, может приводить к длительному итерационному процессу массовых доработок и повторных испытаний продукции.

Второй метод обеспечивает высокое качество выполнения всего процесса проектирования, разработки и изготовления, и тем самым минимум экономических потерь от брака, что более рентабельно при создании сложных систем. При этом сокращается, но не исключается выходной контроль функциональной безопасности и качества продукции. Для создания современных прикладных высококачественных, безопасных систем необходимы оба метода, с акцентом на применение регламентированных технологий.

Установлено, что затраты на разработку резко возрастают, когда показатель качества приближается к пределу, достижимому при данной технологии и уровне автоматизации процесса разработки. Это привело к существенному ***изменению в последние годы методологии и культуры в области создания и совершенствования систем и ПП***. Непрерывный рост требований к функциональной безопасности и качеству ПП стимулировали создание и активное применение международных стандартов и регламентированных технологий, автоматизирующих процессы всего жизненного цикла, начиная с инициирования проекта (см. гл. 2).

Экспериментальное определение и прямое измерение ***величины реальной функциональной безопасности*** сложных систем и комплексов программ в большинстве случаев затруднены, необходимостью учитывать и оценивать редкие катастрофические отказовые ситуации. Для этого приходится проводить длительные эксперименты с имитацией критических и опасных ситуаций эксплуатации систем, что может быть связано с большим риском и весьма дорого. Кроме того, редкие и одиночные отказовые ситуации не позволяют оценить статистически достоверные значения длительностей наработки на отказы и величины вероятного ущерба при этом. Аналитические расчеты и экспертные оценки величины функциональной безопасности комплексов программ усложняет случайное, непредсказуемое размещение дефектов и ошибок [23,

28] в программных модулях и компонентах, которые могут полностью определять функциональную безопасность.

Поэтому высокая функциональная безопасность программных продуктов достигается и определяется преимущественно ***за счет технологий обеспечения качества*** на всех этапах жизненного цикла систем и ПП. Для этого в стандартах регламентирующих функциональную безопасность систем и КП значительное ***внимание уделяется технологическим процессам и инструментальным системам обеспечения безопасности*** и качества ЖЦ комплексов программ (см. гл. 3).

1.3. Характеристики систем и среды, для которых должна обеспечиваться функциональная безопасность программных продуктов реального времени

Эта характеристика может значительно модифицироваться в жизненном цикле ПП и соответственно изменяться конкретное содержание функций, которые подлежат применению и оцениванию. На последовательных этапах ЖЦ, функции промежуточных продуктов (спецификаций компонентов, модулей, текстов программ и т.п.) должны оцениваться на соответствие требованиям в отдельных документах. Это позволяет поэтапно формализовать применяемые субхарактеристики и атрибуты компонентов функциональной пригодности. Такими атрибутами могут быть: функциональная адекватность программ документам и декларированным требованиям, утвержденным заказчиком; степень покрытия тестами исходных требований; полнота и законченность реализации этих требований; точность выполнения требований детальных спецификаций на функциональные компоненты. Функциональная пригодность определяется качеством взаимосвязи и согласованности последовательных формулировок содержания и реализации основных фрагментов в цепочке стандартизированных требований технического задания на ПП: ***целей – назначения – функций – исходной информации – результатов для пользователей***, (рис.1.1) определяющих что:

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

Цель жизненного цикла или ***системная эффективность*** ПП может оцениваться, в основном, экспертно и является исходной для прослеживания всех последующих, производных свойств и атрибутов функциональной пригодности. Назначение ПП детализируются и формализуются в ***требованиях к функциям*** компонентов и всего комплекса программ, способного реализовать декларированные цели.

Функции программного продукта реализуются в определенной аппаратной, операционной и пользовательской внешней среде системы, характеристики которых существенно влияют на функциональную пригодность. Для выполнения требуемых функций комплекса программ 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 ПП в соответствующей внешней среде. Цель и функции программного продукта реализуются тогда, когда ***выходная информация*** достигает потребителей – объектов или операторов-пользователей, с требуемым содержанием и качеством, достаточным для обеспечения ее эффективного применения.

Содержательная часть этой информации определяется конкретными задачами системы, их основными технико-экономическими и/или социальными показателями функционирования и отражается метриками в использовании. ***Степень покрытия*** всей выходной информацией: целей, назначения и функций ПП для пользователей, следует рассматривать как ***основную меру качества функциональной пригодности***.

Исходные данные:
- задачи и требования к характеристикам системы;
- распределение функций между персоналом, аппаратурой и программным комплексом;
- комплексная модель системы и программ во внешней среде.

Цель программного продукта:

- удовлетворение экономических, социальных, технических или научных потребностей заказчика и пользователей в заданной среде применения.

Назначение программного продукта:

- реализация ЖЦ ПП и системы в соответствие с заданной целью, эффективностью, безопасностью и качеством, для применения пользователями.

Функции программного продукта:
- соответствие автоматизируемых функций и комплекса задач назначению системы и жизненному циклу ПП;
- общие технические требования к реализации функций.

Технико-экономическая эффективность программного продукта:
- требования к технико-экономическим характеристикам, к безопасности и качеству функционирования системы;
- ресурсы на создание, эксплуатацию и весь ЖЦ ПП.

Входная информация:
- соответствие состава входных данных функциям ЖЦ ПП;
- перечень, описание и регламент поступления информации.

Выходная информация:
- состав потребителей, назначение и содержание выходной информации;
- временной регламент и характеристики процесса реализации пользователем автоматизированных функций.

Рис.1.1

Прослеживание и оценивание адекватности и полноты состава выходной информации снизу вверх к назначению ПП должны завершать выбор базовых характеристик качества функциональной пригодности, независимо от сферы

применения системы. В процессе проектирования в составе функциональной пригодности могут быть выделены две группы базовых характеристик, определяющие ***функциональные и структурные требования и особенности ПП***. При формализации и выборе ***функциональных требований*** заказчиком следует, четко ***формулировать в документах контракта***:

- экономические, организационные, технические и/или социальные стратегические цели всего жизненного цикла системы, программного продукта и его компонентов;
- системную эффективность и, в том числе, требуемые технико-экономические показатели применения ПП в составе системы;
- назначение, внешнюю среду, условия эффективного и безопасного применения ПП;
- функциональные задачи основных компонентов и комплекса программ в целом, а также системную эффективность каждого компонента;
- необходимую и достаточную безопасность применения, характеристики качества и временной регламент решения каждой функциональной задачи;
- соответствие ПП и его компонентов выделенным стандартам и нормативным документам на проектирование и применение.

Функциональная пригодность комплекса программ реального времени зависит от ***структурных (архитектурных) характеристик***, которые должны отражаться в требованиях технического задания и/или спецификаций на компоненты и ПП в целом [9, 19, 31]:

- соответствие функций и структуры комплекса программ аппаратной и операционной среде и их ограниченным ресурсам;
- - соответствие правил структурного построения комплекса, функциональных компонентов и модулей, типовым требованиям к архитектуре ПП и его компонентов, а также к уровню покрытия ими заданных функций комплекса программ;
- должны быть заданы интенсивность и объемы потоков входной информации, временной регламент и динамические характеристики процесса реализации автоматизированных функций, допустимое время задержки (ожидания) выдачи результатов решения задач потребителям;
- должны быть предусмотрены и реализованы требования к контролю, хранению, обновлению и восстановлению программ и данных.

В ряде систем особое значение для функциональной пригодности имеет проектирование и ***организация информационного обеспечения и базы данных***. Состав, структура и способы организации данных, а также требования к обмену данными между компонентами комплекса, должны быть адекватны организации информационного обеспечения и функциям системы. При этом должны

быть определены: назначение базы данных и состав информации в ней; совместимость ПП с другими системами по интерфейсам, источникам и потребителям информации базы данных; организация сбора, передачи, контроля и корректировки информации базы данных.

В зависимости от назначения программного продукта функциональная безопасность и/или некоторая конструктивная характеристика может стать доминирующей или даже почти полностью определяющей функциональную пригодность ПП. Эти характеристики трудно свести к количественным мерам и зачастую их приходится оценивать по наличию свойств и ряда типовых процедур в ПП или по величине необходимых затрат ресурсов, достаточно заметно влияющих на функциональную пригодность.

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 к ПП при разработке должны быть прослежены сверху вниз до модулей, и использоваться как эталоны при установлении необходимой корректности соответствующих компонентов. Данное понятие включает обеспечение эталонных (ожидаемых) данных с необходимой степенью точности расчетных значений в соответствии с требованиями технического задания и спецификаций. В процессе проектирования и разработки модулей и групп программ применяются частные структурные критерии корректности, которые включают корректность структуры программ, обработки данных и межмодульных интерфейсов. Каждый из частных критериев может характеризоваться несколькими методами измерения качества и достигаемой степенью корректности программ: детерминировано, стохастически или в реальном времени.

Мерой выбранной корректности, может быть относительное число протестированных функций и маршрутов, которое может измеряться в процентах от общего числа исполняемых. Опыт показывает, что зачастую в готовом, сложном программном продукте оказываются протестированными только около 50-70% функций и маршрутов, и практически очень трудно эту величину довести до 90-95%. Косвенно эту величину при определенной автоматизации и квалификации специалистов отражает трудоемкость и длительность тестирования, что непосредственно влияет на функциональную пригодность [2, 7, 17].

При любом виде деятельности людям свойственно непредумышленно ошибаться, результаты чего проявляются в процессе создания или применения объектов или систем. В общем случае под ошибкой подразумевается дефект, погрешность или неумышленное искажение объекта или процесса. При этом предполагается, что известно его правильное, эталонное состояние, по отношению, к которому может быть определено наличие отклонения – ***дефекта или ошибки***. Для систематической, координированной борьбы с ними необходимы исследования факторов, влияющих на функциональную безопасность и качество программ со стороны различных, существующих и потенциально возможных дефектов в конкретных программах. Это позволит целенаправлен-

но разрабатывать комплексы методов и средств обеспечения безопасности при реально достижимом снижении уровня дефектов проектирования, разработки и производства.

Различия между ожидаемыми и полученными результатами функционирования программ могут быть следствием ошибок не только в созданных программах и данных, но и ***системных ошибок в первичных требованиях спецификаций***, явившихся исходной базой при создании ПП. Тем самым проявляется объективная реальность, заключающаяся в ***невозможности абсолютной корректности исходных спецификаций требований*** для проектирования. На практике в процессе разработки ПП исходные требования уточняются и детализируются по согласованию между заказчиком и разработчиком. Базой таких уточнений являются ***неформализованные представления и знания специалистов***, а также результаты промежуточных этапов жизненного цикла. Однако установить ошибочность исходных данных и спецификаций еще труднее, чем обнаружить ошибки в созданных программах, так как зачастую принципиально отсутствуют формализованные данные, которые можно использовать как эталонные, и их заменяют ***неформализованные представления заказчиков и разработчиков***.

Дефекты функционирования программных комплексов, не имеющие злоумышленных источников или последствий физических разрушений аппаратных компонентов, проявляются внешне как случайные, имеют разную природу и последствия. Полное устранение негативных воздействий и дефектов, отражающихся на безопасности и качестве функционирования сложных ПП, принципиально невозможно. Проблема состоит в выявлении факторов, от которых они зависят, в создании методов и средств уменьшения их влияния на функциональную пригодность ПП, а также в эффективном распределении ограниченных ресурсов для обеспечения необходимого качества функционирования комплекса программ, равнопрочного при всех реальных негативных воздействиях. Комплексное, скоординированное применение этих методов и средств, в процессе создания, развития и применения ПП позволяет исключать проявления ряда негативных факторов или значительно ослаблять их влияние. Тем самым уровень достигаемой ***функциональной безопасности и качества функционирования ПП может быть предсказуемым и управляемым***, непосредственно зависящим от ресурсов, выделяемых на его достижение, а главное, от системы качества и эффективности технологий, используемых на всех этапах жизненного цикла программ.

Для выбора при проектировании значений характеристик качества программных комплексов необходимо, прежде всего, установить диапазоны рациональные мер и шкал для каждой характеристики и ее атрибутов, которые целесообразно использовать в качестве ***первичных ограничений*** их значений для реальных ПП. Далее должны быть разработаны процессы определения и представления в спецификациях проекта, требований к свойствам и атрибутам каждой характеристики качества. Эти требования должны учитывать реальные ограничения ресурсов, доступных для их достижения в ЖЦ [20, 21, 31].

Решение этих задач должно быть направлено на обеспечение высокой функциональной пригодности ***путем сбалансированного улучшения безопасности и остальных характеристик качества в условиях ограниченных ресурсов***. Для этого в процессе системного анализа при подготовке технического задания и требований спецификаций, значения атрибутов и характеристик безопасности и качества, должны выбираться с учетом их влияния на функциональную пригодность. Излишне высокие требования к отдельным атрибутам качества, требующие для реализации больших дополнительных трудовых и вычислительных ресурсов, целесообразно снижать, если они слабо влияют на основные, функциональные характеристики ПП. Ориентирами могут служить диапазоны изменения атрибутов конструктивных характеристик качества ПП, границы количественных или качественных шкал которых сверху и снизу ***могут быть выбраны на основе следующих принципов***:

- предельные значения характеристик функциональной безопасности и качества, должны быть ограничены сверху, допустимыми или рациональными затратами ресурсов на их достижение, при разработке системы и ПП;
- наибольшие допустимые затраты ресурсов, например, труда и времени для реализации функциональной безопасности и конструктивных характеристик, должны обеспечивать функциональную пригодность жизненного цикла системы и ПП на достаточно высоком уровне;
- допустимые наихудшие значения безопасности и отдельных конструктивных характеристик качества, могут соответствовать значениям, при которых начинает снижаться функциональная пригодность при применении системы и ПП;
- ограниченные значения отдельных конструктивных характеристик качества, не должны негативно отражаться на возможных высоких значениях других приоритетных характеристик.

1.4. Ресурсы для обеспечения функциональной безопасности программных продуктов реального времени

Многие проекты по обеспечению безопасности систем и программных продуктов терпели и терпят неудачу из-за отсутствия у разработчиков и заказчиков при подготовке контракта четкого представления о реальных финансовых, трудовых, временных и иных ресурсах, необходимых для их реализации. Поэтому одной из основных задач при проектировании функциональной безопасности ПП является ***экономический анализ и определение необходимых ресурсов для создания и обеспечения всего ЖЦ комплекса программ*** в соответствии с требованиями контракта и технического задания [5, 9, 34].

При прогнозировании необходимых ресурсов для проекта, ЖЦ комплекса программ и обеспечение его безопасности можно разделить на две части, существенно различающиеся экономическими особенностями процессов, характеристиками и влияющими на них факторами. ***В первой части ЖЦ*** производятся системный анализ, проектирование, разработка, тестирование и испытания ***базовой версии ПП***. Номенклатура работ, их трудоемкость, длительность и другие экономические характеристики на этих этапах ЖЦ существенно зависят от требуемых характеристик системы и ее безопасности, технологии и инструментальной среды разработки.

Вторая часть ЖЦ, отражающая эксплуатацию, сопровождение, модификацию и перенос комплекса программ на иные платформы, в меньшей степени связана с функциональными характеристиками системы и среды разработки. Номенклатура работ на этих этапах более или менее определенная и стандартизирована (см. **ISO 12207**, **ISO 14764**, **ISO 15846**), но их трудоемкость и длительность могут сильно варьироваться, в зависимости от массовости и других внешних факторов распространения и применения версий ПК. Определяющими становятся ***потребительские характеристики ПП***, а их экономические особенности с позиции разработчиков и вложенные затраты на очередную версию отходят на второй план (см. первый сценарий). Поэтому планы на этих этапах имеют характер общих взаимосвязей содержания работ, которые требуют распределения во времени, индивидуально для каждого проекта. ***Для прогнозирования и планирования любых процессов или характеристик*** объектов (в частности программных продуктов) используются исходные данные ***двух типов***:

- функции и номенклатура характеристик прогнозируемого объекта или процесса, для которого необходимо спланировать жизненный цикл и его этапы;
- характеристики прототипов и пилотных проектов, в некоторой степени, подобных планируемому, о которых известны реализованные планы, необходимые ресурсы и экономические характеристики уже завершенных аналогичных систем или объектов.

Совместная, корректная обработка исходных данных этих двух типов позволяет при проектировании оценивать и получать новые, прогнозируемые характеристики процессов, планов и экономических показателей создания программных продуктов. Исходные данные ***первого типа*** отражают требуемые характеристики конкретного создаваемого объекта или системы, доступные методы и инструментальные средства автоматизации труда при их создании. Эти данные последовательно детализируются и уточняются в процессе проектирования и дальнейшего ЖЦ, что, в частности, позволяет уточнять отбор компонентов аналогичных объектов и их характеристик для исходных данных второго типа.

Второй тип исходных данных для обоснования и планирования разработки ПП составляют обобщенный опыт проектирования и экономические характе-

ристики реализованных прототипов нового комплекса программ. Для достоверного планирования необходимо накопление, обобщение и изучение конкретных данных о реализованных планах, затратах и использованных ресурсах завершенных разработок в различных аспектах.

Важнейшим ресурсом при создании любых программных продуктов являются люди – ***специалисты***, с их уровнем профессиональной квалификации, а также с многообразием знаний, опыта, стимулов и потребностей [20, 21, 31]. Быстрый рост сложности и повышение ответственности за качество и функциональную безопасность комплексов программ привели к ***появлению новых требований к специалистам,*** обеспечивающим все этапы их жизненного цикла. Эти специалисты должны владеть ***новой интеллектуальной профессией***, обеспечивающей высокое качество и безопасность программных продуктов, а также контроль, испытания и удостоверение реального достигнутого качества на каждом этапе разработки и совершенствования программ [5, 14].

Безопасность вычислительной системы обеспечивается ***двумя видами работ***: созданием функциональных программ высокого качества с минимальным количеством дефектов и ошибок, отражающихся на безопасности, и разработкой специального комплекса программ, для повышения безопасности функционирования основных программ. ***Руководство безопасностью сложного проекта ПК*** должны осуществлять лидеры – менеджеры (рис. 1.2):

- ***менеджер безопасности проекта*** – этот специалист, обеспечивающий коммуникацию между заказчиком и проектной командой специалистов, его задача – определить и обеспечить полное удовлетворение требований заказчика по безопасности системы и ПП;
- ***менеджер-архитектор комплекса программ повышения безопасности*** – управляет коммуникациями и взаимоотношениями в проектной команде, является координатором создания компонентов, разрабатывает базовые, функциональные спецификации и управляет ими, ведет график проекта и отчитывается за его состояние, инициирует принятие критичных для хода проекта решений.

В реализации любого крупного проекта можно выделить две категории специалистов: разрабатывающих компоненты и ПК в целом и обеспечивающие технологию, безопасность и качество программного продукта. При выборе заказчиком надежного поставщика-разработчика проекта необходима ***оценка тематической и технологической квалификации*** возможного коллектива специалистов, а также его способности реализовать проект с заданными требованиями, качеством и функциональной безопасностью.

Специалисты первой категории непосредственно создают компоненты и ПК в целом с заданными показателями качества и безопасности. В процессе разработки их функции заключаются в тщательном соблюдении принятой в фирме технологии и в формировании всех предписанных руководствами исходных и отчетных документов. Разделение труда специалистов этой категории в крупных проектных коллективах приводит к необходимости их дифференциации по квалификации и областям деятельности:

- ***спецификаторы*** – подготавливают описания функций соответствующих компонентов с уровнем детализации, достаточным для корректной разработки текстов программ программистами;
- ***разработчики программных компонентов – программисты*** создают компоненты, удовлетворяющие спецификациям, реализуют требуемые функции продукта;
- ***системные интеграторы*** создают на выходе требуемые крупные компоненты или весь комплекс программ;
- ***тестировщики*** – обеспечивают проверку функциональных спецификации, выполняют тестирование для каждой из фаз и компонентов проекта;

управляющие сопровождением и конфигурацией – обеспечивают взаимодействие компонентов и реализацию версий ПК;

-

Рис. 1.2

- ***документаторы*** – осуществляют подготовку и издание сводных технологических и эксплуатационных документов в соответствие с требованиями стандартов.

Специалисты второй категории – технологи, обслуживающие и сопровождающие технологический инструментарий, который применяется специалистами первой категории, обеспечивают применение системы качества проекта или предприятия, контролируют и инспектируют ее использование (см. рис.1.2). Специалисты, управляющие обеспечением качества и безопасности ПК, должны овладеть стандартами и методиками фирмы, поддерживающими регистрацию, контроль, документирование и воздействия на показатели качества на всех этапах ЖЦ комплекса программ. Они должны обеспечивать эксплуатацию системы качества проекта, выявление всех отклонений от заданных показателей качества и безопасности объектов и процессов, а также от предписанной технологии на промежуточных и заключительных этапах разработки. Для ***анализа затрат ресурсов в жизненном цикле ПК***, их целесообразно разделить на две части (рис. 1.3):

- 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;
- основные составляющие ***дополнительных затрат***, обеспечивающие требуемые конструктивные характеристики качества и функциональную безопасность в соответствии с целями и сферой его применения.

Обеспечение функциональной пригодности является основной целью при использовании финансовых, трудовых, вычислительных и других ресурсов в жизненном цикле ПК (см. рис.1.3). Эти затраты зависят, в первом приближении, от сложности алгоритмов, объема комплекса программ и баз данных, которые определяют трудоемкость и длительность полного цикла их разработки.

Основные затраты идут на овеществленный, преимущественно, интеллектуальный труд специалистов различных категорий. Поэтому для их измерения наиболее универсальной единицей стали трудозатраты специалистов в человеко-днях или человеко-месяцах, которые обычно могут преобразовываться в стоимость процесса разработки [13, 16, 21]. 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 и безопасности, часто приводит к тому, что использование ресурсов на их реализацию может превышать базовые затраты на обеспечение функциональной пригодности.

Рис.1.3.

Абсолютная величина затрат на разработку ПП, также как и ее длительность, зависят от ряда факторов, которые могут изменять их в различных направлениях. При анализе ***затрат на обеспечение требуемых функций*** сложных ПП доминирующим фактором, влияющим на их величину, является сложность функциональной части комплекса программ и базы данных. Наиболее активно в качестве простейшего показателя сложности используется ***объем программ, выраженный числом*** операторов (команд) или строк текста на языке программирования (с учетом коэффициента, зависящего от класса ПП и специфики языка).

Объем программ без комментариев является одной из наиболее достоверно измеряемых характеристик сложности ПП и достаточно адекватен экономическим затратам на его разработку. Реальное изменение создаваемых в настоящее время ***новых сложных ПП*** объемом от 10^4 до 10^7 строк определяет диапазон трудоемкости разработки таких программ от человеко-года до тысяч человеко-лет. По большому числу проектов подтверждена высокая статистическая корреляция (0,85) между объемом новых программ в операторах (строках текста без учета комментариев) и трудоемкостью их разработки.

Ограниченные ресурсы времени реализации проектов ПП являются одним из ***сильных факторов***, влияющих на достижимое качество и безопасность комплексов программ. При реальном допустимом времени реализации проекта, оно ограничивает затраты на все промежуточные этапы работ, и тем самым на качество их выполнения. При современных технологиях ***полностью новые, сложные комплексы программ***, реализуемые в допустимое время 2 – 4 года, ограничены объемом 10^6 – 10^7 строк текста. Очевидна принципиальная нерентабельность разработки очень сложных ПП за 5 – 10 лет. С другой стороны, даже относительно небольшие программы высокого качества и безопасности в сотни тысяч строк по полному технологическому циклу с сертификационными испытаниями продукции редко создаются за время, меньшее, чем полгода – год. Таким образом, диапазон вариации длительностей разработок ПП много меньше, чем вариация их трудоемкости, а эти длительности ограничены сверху и снизу, и объем новых программ является одним из основных факторов, определяющим эти границы [16, 19, 27].

Границу длительностей снизу определяют естественный технологический процесс коллективной разработки и необходимость выполнения ряда взаимоувязанных работ на последовательных этапах, которые обеспечивают получение сложных ПК требуемого качества. Даже большие усилия по автоматизации технологии и организации разработки программ приводят к сокращению длительностей только на десятки процентов, в то время как трудоемкость может уменьшаться на порядок и больше.

Крупные затраты, достигающие 30% от полных затрат на разработку сложных ПК, приходятся на ***верификацию и тестирование программных компонентов***, что должно обеспечивать корректность, функциональную безопасность и надежность ПП в целом. Хотя эти характеристики различаются, затраты на их

реализацию полностью разделить невозможно. Поэтому оценивание и выделение ресурсов на решение этих задач целесообразно анализировать совместно.

Затраты на тестирование и испытания ПП в целом обычно могут быть достаточно четко выделены из остальных затрат, так как в этих процессах могут непосредственно участвовать заказчик и пользователи. Величина этих затрат, без учета ресурсов, необходимых для имитации внешней среды, может составлять около 10% от общих затрат на разработку. При этом практически невозможно разделять затраты на оценивание отдельных стандартизированных характеристик.

Затраты на функциональную безопасность и надежность ПП определяются требуемым уровнем защищенности и сложностью (объемом) программ для ее реализации. При наличии особенно высоких требований к безопасности критических ПП эти затраты могут даже в 2 – 3 раза превышать затраты на решение базовых, функциональных задач. Для типовых систем трудоемкость создания программных комплексов функциональной безопасности обычно составляет 20 – 40% от затрат на решение основных, функциональных задач при требованиях наработки на отказ в десятки тысяч часов и для минимального обеспечения автоматического рестарта в системах.

Возможные затраты трудовых и временных ресурсов ***на развитие и совершенствование качества и безопасности комплекса программ*** в процессе сопровождения зависят не только от внутренних свойств программ, но также от запросов и потребностей пользователей на новые функции и от готовности заказчика и разработчика удовлетворить эти потребности. Величина этих затрат определяется рыночной конъюнктурой для данного программного продукта и коммерческой целесообразностью его модификации и развития. [21, 44, 54].

Затраты на обеспечение и реализацию сопровождения программ определяются длительностью цикла жизни комплекса программ; его мобильностью, уровнем автоматизации технологии разработки и тиражом программного продукта. Для их оценивания, прежде всего, необходимо выделять основные виды затрат при сопровождении конкретного комплекса программ и наиболее существенные факторы, которые на них влияют. Сокращение затрат на сопровождение возможно за счет некоторого увеличения затрат при разработке ПП, так что при рациональном проектировании в сумме затраты могут быть уменьшены иногда весьма заметно. Затраты на сопровождение можно считать аддитивными и включающими ***составляющие*** (см.рис 1.3):

- затраты на обнаружение и устранение ошибок и дефектов в каждой версии ПП;
- затраты на доработку и совершенствование программ, формирование и испытание новых модернизированных версии ПП;
- затраты на тиражирование каждой новой версии и ее внедрение в эксплуатируемых и новых системах.

В современных проектах ПП большую или меньшую долю составляют ***готовые апробированные компоненты*** из других подобных разработок – ***прототипы и/или покупные пакеты прикладных программ***. Это позволяет значительно ускорять работы и сокращать затраты на создание сложных комплексов программ. Перед разработчиками проекта ПП зачастую возникает дилемма: разрабатывать ли весь комплекс программ (в том числе для обеспечения безопасности) полностью из новых компонентов или использовать, адаптировать и приспосабливать готовые компоненты, какие и в каком количестве. В результате при первичном экономическом анализе затрат на создание ПП с требуемыми характеристиками безопасности, целесообразно рассматривать ***два альтернативных варианта*** определения затрат на разработку:

- полностью нового программного продукта, для которого отсутствуют или недоступны подходящие готовые компоненты – прототипы и/или их заведомо нерентабельно использовать;
- программного продукты на базе комплексирования набора готовых программных компонентов и информации баз данных, для которого почти не требуется создания новых компонентов.

При сборке нового ПК из комплектующих компонентов, может потребоваться доработка некоторых из них, или создание специальных программ, для их взаимодействия. В пределе при построении нового ПК на 80 – 90% из готовых компонентов суммарные затраты могут сокращаться в 3 – 5 раз. В этом случае, кроме 10 – 20% затрат на создание новых программных компонентов, необходимы ресурсы на комплексирование нового ПК, его тестирования, испытания и документирование.

При создании сложных ПП, одной из больших составляющих затрат могут быть ***ресурсы на генерацию тестов***. В ряде случаев они соизмеримы с затратами на создание основных функций комплексов программ (в том числе для обеспечения безопасности), что определяется принципиальным соответствием ***сложности необходимых наборов тестов*** и тестового покрытия программ, и ***сложности функций***, реализуемых испытываемым ПП. Создание представительных совокупностей тестов возможно путем использования реальных объектов внешней среды или с помощью ***программных имитаторов, адекватных этим объектам по результатам функционирования и генерируемой информации***. При этом возникает проблема – какой метод и когда выгодней по затратам на генерацию тестов и по обеспечению необходимой степени покрытия тестами испытываемых ПП. Затраты ресурсов на натурные эксперименты для генерации тестов при проведении разработки, испытаний и определения качества пропорциональны реальному времени функционирования проверяемого ПП и затратам на применение привлекаемых средств реальной внешней среды. Они включают стоимость эксплуатации реального объекта, создающего тесты в единицу времени.

Имитаторы тестов необходимы не только для оценивания достигнутых характеристик безопасности и качества комплексов программ, но также для их комплексной отладки, испытаний и при создания версий [16, 23]. Поэтому затраты на программные имитаторы и их экономическую эффективность целесообразно рассматривать с учетом всего комплекса задач, которые они способны и должны решать в ЖЦ ПК. Затраты на программную имитацию тестовых данных определяются ресурсами, необходимыми на проектирование и эксплуатацию сложных комплексов программ для этих целей:

- затратами на разработку комплекса программ для имитации информации внешних объектов и среды их функционирования;
- затратами на эксплуатацию программ имитации за время проведения тестирования, испытаний и/или определения характеристик безопасности и качества тестируемого ПП;
- затратами на первичную установку и эксплуатацию моделирующего компьютера и вспомогательного оборудования, используемого в имитационном стенде.

Затраты на эксплуатацию программ имитации в основном определяются длительностью проведения тестирования, испытаний и/или измерения характеристик безопасности и качества ПП. Значения этого времени соответствуют реальному времени генерации тестовых данных и тестирования программ. Обычно имитаторы используются для тестирования нескольких ПП разного, но близкого целевого назначения. В результате удельные затраты на их создание и эксплуатацию быстро убывают при унификации имитаторов и расширении области их применения для тестирования и оценивания качества большого числа ПП, имеющих близкое функциональное назначение. Даже приближенные оценки этих затрат в большинстве случаев показывают ***высокую рентабельность программных имитаторов внешней среды***, особенно для квалификационного тестирования и оценивания характеристик безопасности сложных ПП реального времени.

Глава 2. Разработка требований к надежности и функциональной безопасности программных продуктов

2.1. Общие требования к проектированию и производству сложных программных продуктов

Команда специалистов при проектировании сложного программного продукта должна ***понять проблемы заказчика до начала его проектирования и производства***. Для этого следует использовать ***анализ, выявление и освоение его профессиональных проблем и интересов***. Программный менеджер и/или системный инженер должен быть знаком с основными способами проектирования сложных систем, знать, ***как перевести расплывчатые требования и пожелания заказчика системы в четкое техническое задание***, уметь разговаривать с заказчиком и потенциальными пользователями системы на языке предметной области. ***Понимание потребностей заказчика и пользователей*** необходимый организационный этап, так как разработчики редко получают совершенные спецификации требований к создаваемой системе, они должны сами «***добывать» у заказчика*** информацию, необходимую им для успешной работы. Термин ***выявление требований*** точно отражает данный процесс, в котором проектировщики должны играть активную роль. В соответствии с принятой политикой предприятия для ***определения требований к системе и продукту*** должны осуществляться ***следующие действия специалистов проектирования*** [4,11, 26]:

- идентификацию заинтересованных лиц или групп, имеющих законный интерес к системе и программному продукту в течение его жизненного цикла;
- определение ограничений системных решений, которые являются неизбежным следствием существующих соглашений, управленческих или технических решений специалистов;
- установление масштаба и требуемых ресурсов для реализации системы и программного продукта;
- определение представительного набора последовательных действий специалистов для идентификации всех требуемых функциональных возможностей, которые отвечают предполагаемым сценариям и внешней среде применения, функционирования и сопровождения системы и ПП;
- анализ полноты и корректности множества установленных требований к системе и ПП;

- специфицирование безопасности и других требований заказчика, имеющих отношение к критическим показателям применения системы;
- разрешение проблем и конфликтов, возникающих в связи с определением требований к системе;
- документирование требований заказчика в форме, приемлемой для управления требованиями в течение проектирования и производства системы и ПП.

Цель анализа требований состоит в преобразовании потребностей заказчика, выраженных в виде пользовательского представления о системе и ПП, в ***формализованные функциональные возможности***. В ходе этого процесса должно создаваться четкое представление о будущей системе, которая будет удовлетворять требованиям заказчика и не потребует специальных мероприятий в связи с ее практическим применением. В результате определяется комплекс оцениваемых требований, какими функциями и характеристиками должна обладать система и какими должны быть их значения, чтобы удовлетворить требованиям заказчика.

Определения и формирования требований заказчика состоит в формулировании требований к системе, выполнение которых должно обеспечить ***функциональные возможности***, необходимые пользователям системы и иным заинтересованным лицам, в заданной эксплуатационной среде. Должны быть определены цели создания и назначения компонентов системы, а также функции и область ее применения. Эти данные анализируются и преобразуются в общий набор требований заказчика, они описывают необходимое поведение системы в процессе взаимодействия с эксплуатационной средой, и совокупность образцовых показателей, проверка на соответствие которым является целью процесса аттестации и позволяет подтвердить, что система и ПП отвечает заявленным требованиям.

Проектирование общих требований к компонентам и комплексам программ включает:

общие требования к проектированию сложных программных продуктов;

- анализ, выявление и освоение профессиональных проблем заказчика;
- этапы проектирования производства сложных программных продуктов;
- особенности внешней среды системы;
- распределение системных требований между аппаратными и программными компонентами, интерфейсов с внешней средой;
- учет ограниченных ресурсов для реализации требований функциональной пригодности, времени и допустимой длительности проектирования;

общие требования и ограничения системы и комплекса программ реального времени;

- три пространства функций и характеристик программного продукта – заказчика, проектировщика, пользователя;
- функциональные требования к проектированию сложных заказных комплексов программ;
- выявление и прояснение не четких, неоднозначных требований к программному продукту;

формирование требований компонентов и модулей путем декомпозиции функций комплексов программ:

- унификацию архитектуры и интерфейсов модулей, компонентов и комплексов программ;
- нисходящее - восходящее проектирование модулей и программных компонентов;
- распределение и установление ответственности специалистов за качество и сроки создания модулей и компонентов.

повторное использование модулей и компонентов в комплексах программ:

- цели и требования создания и повторного применения программных компонентов и модулей;
- подготовку возможности многократного использования компонентов в различном операционном и внешнем окружении;
- оценивание затрат на применение готовых компонентов при проектировании комплексов программ;
- изменения трудоемкости и длительности создания комплексов программ при проектировании из готовых компонентов.

Рис. 2.1.

Поэтапное проектирование требований к производству способно остановить нерентабельное развитие системы и ПП и избежать крупных затрат заказчикам и разработчикам. В то же время, на базе рекомендуемых при проектировании методов, инструментальных средств и стандартов, может быть подготовлен и обеспечен, эффективный жизненный цикл производства версий высококачественных программных продуктов и их компонентов. Требования заказчика при проектировании могут выражаться ***в форме потребностей, пожеланий и ограничений***. Сценарии применения системы должны использоваться для анализа ее функционирования в заданной среде, с целью ***выявления требований***, которые формально могли быть не заданы заказчиком. Также следует анализировать социальное воздействие организации на пользователей, которые могут повлиять на использование системы или сдерживать процесс ее проектирования. Стандарты и правила должны использоваться для ***определения особенностей внешней среды системы.***

Анализ корректности сформированных требований к системе и программному продукту включает выявление и идентификацию противоречивых, пропущенных, неполных, неоднозначных, нелогичных или непроверяемых требований; расстановку приоритетов и разрешение проблем, возникающие в связи с определением требований. Сюда же относятся требования, которые не могут быть реализованы или которые нецелесообразно реализовывать.

Исходные данные и требования к проектированию программного продукта, включая установленные законодательные и регламентирующие нормативные требования, должны быть, оформлены документально, а их выбор проанализирован поставщиком на адекватность. Спецификацию требований должен представить ***потребитель*** – ***заказчик***. Однако по взаимному согласию ее может подготовить ***поставщик*** – ***разработчик***, в тесном сотрудничестве с потребителем для предупреждений разногласий путем, уточнения определений терминов, объяснения предпосылок и обоснования требований. Неполные, двусмысленные или противоречивые требования должны быть предметом урегулирования с заказчиком и заинтересованными лицами, ответственными за их предъявление.

Для конкретного комплекса программ доминирующие требования выделяются и определяются его ***функциональным назначением***. Программы для компьютеров как ***объекты проектирования***, производства, испытаний и оценки качества характеризуются в жизненном цикле следующими ***обобщенными характеристиками***:

- проблемно – ориентированной областью применения, техническим и социальным назначением программного комплекса;
- конкретным классом и назначением решаемых функциональных задач с достаточно определенной областью применения квалифицированными пользователями;

- масштабом и сложностью комплекса программ и базы данных, решающих единую целевую задачу системы;
- архитектурой комплекса программ и базы данных;
- необходимыми составом и требуемыми значениями характеристик качества и безопасности функционирования программ и величиной допустимого ущерба – риска из-за недостаточного их качества или ресурсов;
- составом потребителей характеристик комплекса программ, для которых важны соответствующие атрибуты качества;
- комплектом стандартов и их содержания, которые целесообразно использовать при выборе технологии и характеристик комплекса программ;
- реальными ограничениями всех видов ресурсов проекта;
- степенью связи решаемых задач с реальным масштабом времени или допустимой длительностью ожидания результатов решения задач;
- прогнозируемыми значениями длительности эксплуатации и перспективой создания множества версий программного продукта;
- предполагаемым тиражом производства и применения программного продукта;
- степенью необходимой документированности программного продукта.
- В соответствии с принципиальными особенностями конкретного комплекса программ при проектировании должны выбираться номенклатура и значения показателей качества, необходимых для его эффективного применения пользователями, которые отражаются в технической документации и в спецификациях требований на конечный программный продукт. При ***проектировании рекомендуется набор критериев*** качества требований к комплексам программ, который включает:
- корректность – отсутствуют дефекты и ошибки в формулировках требований к комплексу программ;
- недвусмысленность – каждое требование должно быть однозначно и не допускать различного понимания и толкования специалистами;
- полнота – состав и содержание требований должны быть достаточны для производства и применения корректного комплекса программ и компонентов;
- непротиворечивость – между разными требованиями к компонентам и комплексу программ отсутствуют конфликты и противоречия;
- модифицируемость – каждое требование допускает возможность его согласованного изменения и развития при производстве комплекса программ.

Системная эффективность применения программных продуктов в стандартах **ISO 9126:1-4, ISO 25000** определяется степенью ***удовлетворения потребностей заинтересованных лиц*** – заказчиков и/или пользователей, которые, в ряде случаев, желательно измерять экономическими категориями: прибылью, стоимостью, трудоемкостью, предотвращенным ущербом, длительностью применения и т.п. В стандартах эта эффективность отражается основной, обобщенной характеристикой качества – ***функциональной пригодностью*** 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 при применении программного продукта [4,14].

Улучшение каждой, ***нефункциональной – конструктивной характеристики качества***, требует некоторых затрат ресурсов, которые в той или иной степени должны отражаться на улучшении основной характеристике качества – на функциональной пригодности. Требования к конструктивным характеристикам имеют значение для проекта постольку, поскольку они обеспечивают требуемое качество реализации основного назначения и функций комплекса программ. Поэтому для каждого проекта необходимо ранжировать характеристики и их атрибуты (приоритеты) и выделять, прежде всего, те требования, которые могут в наибольшей степени ***улучшить функциональную пригодность*** для конкретных целей.

Ограниченные ресурсы для реализации требований функциональной пригодности, могут негативно отражаться на конструктивных характеристиках: на надежности, безопасности, пропускной способности, качестве взаимодействия с внешней средой и с пользователями, качестве документации и других эксплуатационных факторах (см. рис 2.1). Наиболее общим видом ресурсов, используемых в жизненном цикле комплексов программ, являются ***допустимые финансово-экономические, бюджетные затраты***. При анализе требований качества этот показатель может применяться или как вид ресурсных ограничений, или как оптимизируемый критерий. При этом необходимо также учитывать затраты на разработку, закупку и эксплуатацию системы обеспечения качества, технологии и комплекса средств автоматизации разработки программ и баз данных, которые могут составлять существенную часть совокупной стоимости программного продукта.

Обычно заказчики и разработчики первоначально устанавливают требования к каждой характеристике без учета относительных затрат на их достижение, а также без детального анализа их совместного влияния на полную функциональную пригодность у потребителей. Это может приводить к ***несбалансированным значениям требований*** к отдельным, взаимосвязанным характеристикам качества, на которые не рационально используются ограниченные ресурсы проекта, или к не адекватно низким их значениям.

Общие требования к сложным системам и комплексам программ реального времени для автоматизации управления и обработки информации о динамических объектах состоят в следующем:

- для обработки потоки данных из внешней среды могут быть независимыми, несинхронными, различными по интенсивности, содержанию, реальному времени формирования и поступления для обработки;
- информация в сообщениях от источников и объектов внешней среды должна содержать реальное время, к которому относятся сообщения, их координаты и характеристики состояния;
- реальное время решения различных функциональных задач должно эффективно упорядочиваться в соответствии с установленной дисциплиной диспетчеризации, определенной при производстве системы, и реальным временем приема информации из внешней среды;
- алгоритмы и программы функциональных задач системы могут различаться по длительности решения и важности для пользователей, должны включать и использовать текущее и расчетное реальное время результатов решения и исходной информации;
- для эффективного использования ограниченных ресурсов производительности и оперативной памяти компьютеров целесообразно применять приоритеты и прерывания исполнения программ в соответствии с выбранными дисциплинами решения различных функциональных задач;
- информация и сообщения для потребителей и внешней среды могут выдаваться асинхронно, в соответствии с установленной дисциплиной и содержать значения реального времени, которому соответствуют данные в сообщениях.

Требования к программному продукту при проектировании должны детализировать и конкретизировать описания функций до уровня, позволяющего разработчикам вести производство модулей, компонентов и всего комплекса, которые могут быть проверены на корректность их реализации. Функции и основные характеристики сложных комплексов программ условно можно отразить многомерными ***пространствами свойств и значений,*** контроля и обеспечения их реализации. Особенности, соответствие и покрытие этих многомерных пространств, их взаимодействия при различных задачах применения упрощенно можно представить как ***три пространства***, ***функции и характеристики*** которых, используются и взаимодействуют в следующих целях:

- исходные, ***утвержденные требования*** к функциям и характеристикам программного комплекса, согласованные с разработчиками в виде конкретных документов, в соответствии с которыми разработчики обязаны создать и обеспечить применение программного продукта пользователями или в составе системы;
- ***реализованные*** разработчиками функции и характеристики программного комплекса, которые обычно не могут полностью и абсолютно точно соответствовать исходным эталонным требованиям, достоверно

не известны заказчику, разработчикам и пользователям, и не полностью отражены документами;

- реальные функции и характеристики программного продукта, которые ***практически используются*** пользователями и/или системой в соответствии с эксплуатационной документацией, и могут не совпадать с исходными реализованными требованиями, вследствие превышения их значений или не полного их использования.

Требуется удостоверять, что требования являются, с одной стороны, необходимыми и достаточными для удовлетворения при применении компонента, а с другой – необходимыми и достаточными входными данными для других процессов и компонентов, в частности ***для проектирования функций и архитектуры комплекса программ***. Для этого при производстве в общем случае должны быть выделены ***руководители и коллективы специалистов,*** которые должны планировать, утверждать, выпускать, распространять и сопровождать ***комплекты достоверных документов на программный продукт***. Они должны стимулировать разработчиков компонентов, программных комплексов и их корректировок, осуществлять непрерывное, регламентированное документирование процессов и результатов своей деятельности, а также контролировать полноту и качество документов.

Техническое задание на проект и исходные требования к производству программного продукта целесообразно формировать при проектировании на основе общих положений программной инженерии, и должны содержать поэтапно уточняющиеся, детализирующиеся и дополняющиеся при детальном и рабочем проектировании разделы:

- общие технические требования, перечень стандартов и базовых нормативных документов для выполнения проекта;
- общие требования к программному комплексу; требования к функциям и основным характеристикам качества;
- требования к внешней среде применения комплекса программ;
- специальные требования к аппаратной и операционной платформам для реализации программного комплекса;
- требования к структуре, оформлению и содержанию эксплуатационной и технологической документации;
- этапы и график выполнения основных работ проекта;
- ожидаемые результаты проекта и форма их представления;
- порядок контроля исполнения проекта и приемки результатов работы.

Выходные проектные данные для производства программного комплекса должны быть документально оформлены и выражены в требованиях заказчика так, чтобы ***их можно было проверить***, и подтвердить соответствие входным проектным требованиям. Эти данные должны содержать ***критерии приемки***

программного продукта заказчиком или ссылки на них, а также идентифицировать те характеристики, которые являются ***критическими для его надежного и безопасного функционирования и применения.*** Для разработчиков особенно важно формализовать требования в документах и согласовать их с заказчиком при утверждении контракта и технического задания на проект. Требования к характеристикам, утвержденные после предварительного проектирования, могут быть закреплены в техническом задании ***как обязательные для детального проектирования и производства***.

В процессе проектирования, необходимо проверять ***качество и корректность требований они*** должны быть ***верифицируемыми***. Определение требований напрямую связано с процедурами проверки и утверждения аттестации (стандарт **ISO12207:2008)**. ***Определения качества требований*** позволяют выявить и прояснить ***нечеткие, неоднозначные требования***. Такое неоднозначное требование целесообразно заменить несколькими требованиями, каждое из которых будет иметь свой собственный критерий качества. Не каждое требование может иметь четкий критерий качества, который можно использовать для проверки того, удовлетворяет ли какое-либо решение этому требованию. Добавив разъяснения в критерий качества для каждого требования, можно сделать их осязаемыми и понятными. Это первый шаг по определению критериев для измерения качества решений.

Каждый этап разработки комплекса программ формирует, уточняет и реорганизует требования, чтобы сделать их как можно ближе к назначению нового программного продукта. Каждое требование должно иметь ***уникальный идентификатор,*** требование должно отражать ***отдельно распознаваемую, измеряемую сущность***. В проектах сложных комплексов программ нужно применять способ работы с большим числом требований и сложными связями между ними. Следует учитывать, что наряду с существованием некоторого числа требований, связанных с одним основным событием и/или сценарием использования, любое требование может быть связано с другими событиями и/или сценариями использования.

Системная эффективность – функциональная пригодность при проектировании комплекса программ может быть описана количественно или качественно, в виде набора полезных свойств и характеристик программного продукта, их отличий от имеющихся у других комплексов программ, а также источники возможной эффективности. Она определяет назначение, основные функции и требования заказчика, какие задачи должны ***обязательно решаться*** для удовлетворения пользователей, а дополнительные, конструктивные характеристики качества – как, и при каких условиях, заданные функции могут выполняться с требуемым качеством. В результате может быть формализована ***цель использования*** и набор главных характеристик, требований заказчика и пользователей при заказе или приобретении программного продукта, а также предполагаемая его сфера назначения и применения. Полнота и точность представления этих характеристик является исходной для прослеживания реализации

всех последующих, производных свойств и качества функциональной пригодности комплексов программ.

Если ***масштаб проекта и сопутствующие требования заказчика превышают реальные доступные ресурсы***, в любом случае придется ограничиваться в функциях и качестве комплекса программ. Поэтому следует определять, что ***обязательно должно*** быть сделано в первой или очередной версии системы и программного продукта при имеющихся ресурсах проекта. Для этого приходится вести переговоры. ***Привлечение заказчика*** к итерационному проектированию и управлению масштабом и функциями, повышает взаимные обязательства сторон, способствует росту взаимопонимания и доверия между заказчиком и разработчиками.

Декомпозиция требований, функций, процессов проектирования компонентов и комплексов программ весьма сильно влияет на возможность использования ***апробированного задела*** из предыдущих реализованных проектов. Результаты системного анализа, применение функциональных и информационных моделей пред-метной области, формализация спецификаций требований, функци-ональная ***декомпозиция программных компле-ксов*** и последо-вательная детализация проектов позволяют применять готовые технические решения в различных формах и сочетаниях. Для этого необходимо проектирование функционально законченных программных компонентов и модулей, потенциально готовых, к многократному применению в различной внешней и операционной среде, а также в различных сочетаниях их взаимодействия в комплексах программ. Программные компоненты и модули для производства сложных комплексов программ могут создаваться ***двумя методами***:

- в процессе системного проектирования конкретного комплекса программ и его ***последовательной декомпозиции*** на функциональные задачи и далее на небольшие ***уникальные*** программные компоненты и модули, которые могут иметь произвольную архитектуру и интерфейсы для определенного проекта;
- ***путем поиска, подбора и повторного использования*** готовых апробированных компонентов и модулей, созданных для предшествовавших проектов, с учетом возможности их эффективного использования в других комплексах программ за счет ***унифицированной архитектуры и интерфейсов***.

Декомпозиция требований и процессов проектирования 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, представления и оценивания характеристик***, которую руководитель сможет определить по размеру, сложности выполнения и необходимым ресурсам**.**

Декомпозиция работ обеспечивает руководителей базой для разбиения крупных проектировочных задач на достаточно обозримые, управляемые компоненты. После того, как последние определены, их можно использовать для распределения между специалистами с учетом их квалификации, 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 задач проектирования.

Унификация и структурирование процессов декомпозиции комплексов программ оправданы, если они обеспечивают экономию времени производства и/или применения продукта. Разработка унифицированной структуры компонентов особенно целесообразна для версий программных комплексов, когда затраты могут эффективно окупаться при проектировании множества последовательных вариантов – версий продуктов. Для обеспечения эффективного проек-тирования и сокращения затрат целесообразно-формулировать и соблюдать ряд принципов и правил ***структурного построения и повторного применения программных компонентов*** и модулей. Эти принципы и правила могут иметь некоторые особенности в различных проблемно-ориентированных областях. Однако их формализация и выполнение в конкретных проектах обеспечивают значительное ***снижение трудоемкости и длительности проекти-рования*** – программных продуктов и их версий.

Нижнему иерархическому уровню представления комплексов программ соответствуют программные и информационные модули (модули данных). Эти компоненты (например, 100 – 1000 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.

Нисходящее – восходящее проектирование модулей и программных компонентов занимает важное место при проектировании 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, комплексировании 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 заказчика.

2.2. Организация и планирование разработки требований к надежности и безопасности программных продуктов

Одна из трудностей планирования процессов для достижения высокой функциональной безопасности систем состоит обычно в ***отсутствии полной совокупности достоверных требований заказчика и значений критериев качества*** на начальных этапах проектирования и разработки, а также итерационный процесс их конкретизации в течение всего жизненного цикла ПП. Для ***выявления реальных потребностей пользователей*** необходим организационный и технический процесс, в котором разработчики и заказчики должны совместно играть активную роль. Чтобы помочь разработчикам решить эти проблемы, лучше понять потребности пользователей и других заинтересованных лиц, целесообразно использовать методы [21, 30, 31]:

- интервьюирование и анкетирование – создание структурированного интервью потребностей заказчика в функциональной безопасности системы;
- совещания, посвященные анализу и синтезу требований, – формулирование и определение целей безопасности системы и программного продукта, ознакомление с ними всех участников проекта;
- отбор идей и методов, позволяющих выявить и/или уточнить функции и характеристики системы и ее компонентов, провести их классификацию, определить атрибуты приоритетов (критический, важный, полезный), трудоемкости реализации;
- анализ прецедентов концепций требований к функциональной безопасности систем и комплексов программ;
- выявление или создание прототипа объекта или системы на основе первичных требований к функциональной безопасности.

В ***требованиях к функциональной безопасности*** должны быть полно зафиксированы потребности пользователей в таком виде, чтобы разработчик мог ***построить удовлетворяющее их ПП и его компоненты***. Кроме того, требования должны быть достаточно конкретными, чтобы можно было определить, когда они удовлетворены. Зачастую именно разработчики обеспечивают эту конкретизацию. Вся разработка должна вытекать из требований согласованных с заказчиком, а все спецификации ПК и компонентов должны найти отражение в процессах и продуктах разработки. Разработчики должны применить методы и процессы для того, чтобы ***понять решаемую проблему безопасности до начала разработки системы и ПК***. Для этого стандартами рекомендуется:

- достигнуть соглашения между заказчиком и разработчиком по определению проблемы и задач обеспечения безопасности системы и ПП;

- выделить основные причины – проблемы, являющиеся источниками отказов ПК и стоящие за основной проблемой, функции проекта обеспечения безопасности;
- выявить заинтересованных лиц и пользователей, чье коллективное мнение о необходимой безопасности в конечном итоге определяет успех или неудачу проекта;
- определить, где приблизительно находятся область и границы возможных решений;
- понять ограничения, которые будут наложены на проект, разработчиков и решения.

Степень функциональной безопасности ПП или систем характеризуется величиной предотвращенного или остаточного ущерба, возможного при проявлении дестабилизирующих факторов и реализации конкретных угроз безопасности применения системы и ПП пользователями. При разработке требований к функциональной безопасности систем и программных продуктов необходимо иметь возможность количественно или качественно описать и задать характеристики недопустимого ущерба вследствие отказов, отражающегося катастрофически на их работоспособности и функциональной пригодности. При выборе и формализации требований к функциональной безопасности программного комплекса или системы целесообразно использовать ***набор общих критериев качества, формулирования и описания требований*** [3, 13, 20, 26], которые должны быть:

- полными – набор требований должен содержать все важные для заказчика и пользователей свойства и характеристики функциональной безопасности ПК и системы, которые можно достигнуть, зафиксировать и оценить;
- корректными – каждое сформулированное требование должно отражать необходимое свойство или характеристику безопасности ПП или системы;
- поддающимися проверке – каждое отдельное требование должно быть верифицируемым, то есть существует конечный, эффективный процесс, с помощью которого можно определить, что функциональная безопасность ПП или системы удовлетворяет этому конкретному требованию;
- недвусмысленными – каждое требование можно однозначно интерпретировать для применения и оценивания одной из характеристик безопасности;
- непротиворечивыми – ни одно из требований к функциональной безопасности, не должно противоречить и не конфликтовать ни с каким из остальных требований;

- модифицируемыми – любое изменение требования можно выполнить полно и согласованно, не нарушая структуры и стиля всего набора требований к безопасности.

При разработке требований к функциональной безопасности ПП и систем, эти правила должны применяться в процессе системного анализа к конкретным свойствам и характеристикам, отражающим особенности различных видов отказов и ущерба при нарушении безопасности. В зависимости от назначения, функциональной пригодности и среды применения системы и ПП доминирующими могут быть различные ***виды и значения недопустимого ущерба***, определяющего функциональную безопасность. При разработке требований к безопасности необходимо выбирать номенклатуру возможных опасных отказов, а также виды и значения ущерба в наибольшей степени, отражающиеся на безопасности. Для каждого из них желательно иметь ориентировочные методы оценки, которые позволяют задавать и проверять реализацию требований к функциональной безопасности.

Первичным ориентиром при системном анализе ***недопустимого ущерба, нарушающего работоспособность системы,*** может быть величина стоимости или трудоемкости создания этой системы. Обычно при отказах происходит не полное разрушение системы и имеется возможность ее восстановления. Поэтому катастрофические отказы, целесообразно оценивать ***относительной долей потери работоспособности системы***. Даже очень небольшие потери работоспособности при отказах в особенно уязвимых компонентах, могут отражаться на полной потере безопасности сложных систем, что необходимо учитывать при анализе и определении требований к функциональной безопасности. Таким образом, качественный системный анализ недопустимого ущерба работоспособности при отказах, может использоваться для оценивания, как превышения требований к функциональной безопасности системы, так и при задании недостаточного ее уровня.

Требования к функциональной безопасности некоторых систем, возможно устанавливать с учетом ***трудоемкости устранения последствий отказа*** и затрат на восстановление нормальной работоспособности системы в соответствии с требованиями. Разрушения компонентов программных продуктов или данных во многих случаях ликвидируются относительно просто их корректировкой и перезаписью, если они задублированы. Автоматизация обнаружения отказов и восстановления работоспособности ПК и системы (путем рестарта), может значительно снизить трудоемкость устранения последствий отказов, но требует соответствующих ресурсов. Поэтому в требованиях к безопасности системы может быть полезно, учитывать методы и трудоемкость устранения отказов, и ресурсы необходимые для автоматизации рестарта.

В некоторых случаях требования к безопасности ПП и системы могут определяться с учетом ***длительности неработоспособного состояния*** вследствие каждого отказа. При этом материальный ущерб из-за отказов и трудоемкость

восстановления работоспособности могут быть второстепенными, а главный фактор обеспечения безопасности может заключаться в максимальном сокращении длительности неработоспособного состояния при отказах. Эта длительность в некоторых пределах может не влиять на функциональную безопасность системы, а при превышении ***пороговой длительности отказового состояния*** отражаться на безопасности. Значения пороговой длительности должны задаваться в требованиях и зависят от особенностей динамики функционирования и применения системы.

Кроме того, в некоторых случаях функциональную безопасность может быть полезно отражать ***суммарной длительностью работоспособного состояния системы*** между событиями опасных отказов, относительно общей длительности применения системы, с учетом затрат времени на обнаружение и ликвидацию отказов. Эта характеристика ***относительного полезного времени безопасного функционирования системы*** с исключением затрат времени на обнаружение и ликвидацию опасных отказов, которые отразились на нарушении безопасности с большим ущербом, в теории надежности квалифицируется как ***коэффициент готовности системы***. В этих случаях требования к методам и ресурсам обнаружения отказов и восстановления должны быть направлены на автоматизацию, прежде всего, временных характеристик обеспечения безопасности системы. В требованиях к безопасности должны учитываться затраты ресурсов и времени, необходимые для восстановления нормальной работоспособности системы в соответствии с требованиями безопасности.

В результате системного анализа первично сформулированные ***требования к характеристикам безопасности системы и ПП последовательно уточняются*** и корректируются в процессе взаимодействия заказчика и разработчика с учетом объективно изменяющихся характеристик развивающегося проекта. Для этого целесообразно планировать регулярное проведение разработчиком и заказчиком совместных анализов состояний проекта, или проведение таких анализов в случае значительных проектных событий, чтобы оценивать состояние и развитие выполняемых работ по разработке и модификации системы безопасности.

Высшее руководство проекта должно поставить четкую цель и всесторонне оказывать свою поддержку проектированию функциональной безопасности применения системы посредством распространения среди сотрудников предприятия принятой политики и технологии обеспечения безопасности. ***Письменный документ о политике обеспечения безопасности*** должен быть доступен всем сотрудникам, отвечающим за режим безопасности, а также всем подразделениям организации. ***Этот документ должен содержать***:

- определение функциональной безопасности, ее основные цели, область применения, а также ее значение как механизма, позволяющего эффективно эксплуатировать систему;

- изложение позиции и методов руководства по вопросам реализации целей, технологии и принципов безопасности;
- разъяснение конкретных вариантов политики безопасности, принципов, технологии, стандартов и требований к ее соблюдению, включая: выполнение правовых и договорных требований; требования к обучению персонала правилам безопасности; политику обеспечения бесперебойной работы системы;
- определение общих и конкретных обязанностей специалистов по обеспечению режима безопасности;
- разъяснение процесса уведомления специалистов о событиях, таящих угрозу безопасности.

Необходимо ***разработать процесс проверки***, определить обязанности лиц и задать даты проверок для соблюдения реализации требований документа о политике безопасности. Не исключено, что при эксплуатации ПП могут проявиться не обнаруженные до этого ошибки или уязвимости, а также может возникнуть необходимость пересмотра предположений относительно внешней среды функционирования системы. Тогда по результатам эксплуатации потребуется внесение разработчиком исправлений в ПК, либо переопределение требований безопасности или предположений относительно среды эксплуатации.

Для сложных систем может быть необходима ***стратегия управления модификациями требований к безопасности***. Для этого следует применять информационную иерархию; она начинается с потребностей пользователей, описанных с помощью функций системы, которые затем превращаются в более подробные требования к ПП и компонентам, выраженные посредством прецедентов или традиционных форм описания и стандартизированных характеристик. В требованиях к функциональной безопасности ПП и системы следует указывать, ***какие функции*** должны осуществляться, а ***не то, как*** они могут реализоваться. Они используются для задания функциональных и конструктивных требований, а также ограничений при проектировании. Руководитель проекта должен создать ***совет по контролю за изменениями требований к безопасности***, который призван помогать лидеру в принятии действительно сложных решений и гарантировать, что изменения и утверждения требований будут приниматься только после их обсуждения.

Процессы и технология проектирования и ЖЦ комплекса программ, обеспечения функциональной безопасности, как самостоятельной системы, принципиально не отличаются от проектирования любых других сложных программных комплексов (см. гл. 3, а также ISO 12207:2008). Для этого, прежде всего, необходимо проанализировать и конкретизировать в спецификации требований к ПП, политику, задачи, а также исходные данные и факторы, определяющие безопасность функционирования системы и комплекса программ [14, 26, 36]:

- цели управления безопасностью, критерии качества и их значения, характеризующие необходимый и достаточный уровень функциональной безопасности применения системы пользователями в целом, и каждого из ее основных компонентов в соответствии с условиями среды применения и требованиями спецификаций заказчика;
- перечень и характеристики возможных внешних и внутренних дестабилизирующих факторов и угроз, способных влиять на безопасность функционирования системы в целом, программных продуктов и баз данных;
- требования к методам и средствам контрмер для предотвращения или снижения влияния угроз безопасности, обусловленных негативными внешними воздействиями, а также возможными случайными дефектами программ и данных;
- перечень подлежащих решению задач безопасности, перекрывающих все потенциально возможные угрозы, и оценки эффективности решения отдельных задач, необходимых для обеспечения равнопрочной защиты системы с заданной эффективностью;
- методы управления и достижения заданных значений безопасности; процедуры, которые должны выполняться для каждого процесса и на протяжении всего жизненного цикла ПК; действия, связанные с отчетностью об ошибках, с трассировкой и системой корректирующих действий;
- оперативные методы и средства повышения безопасности функционирования комплекса программ, в течение всего его жизненного цикла путем введения в программы временной, программной и информационной избыточности для реализации системы защиты от актуальных видов угроз;
- организация разработчиков и технология создания ПП; утвержденные обязанности специалистов по обеспечению безопасности, их ответственность и полномочия на утверждение программных компонентов и документов;
- ресурсы, необходимые и доступные для разработки и размещения программной системы обеспечения безопасности (финансово-экономические, ограниченная квалификация специалистов и вычислительных ресурсов компьютера);
- базовые документы и стандарты, используемые для обеспечения безопасности на всех этапах разработки, нормативные документы и методики воспроизводимых измерений показателей достигнутой безопасности, а также состав и значения исходных и результирующих данных, обязательных для проведения испытаний;

- оценки влияния средств обеспечения безопасности на функциональные и временные характеристики системы при автоматизированной обработке основной информации, определяющей ее назначение;
- структура и содержание отчетных документов, удостоверяющих достижение определенной безопасности и качества компонентов и ПП в целом на последовательных этапах разработки, а также их соответствие стандартам и требованиям заказчика.

В процессе решения перечисленных задач должно быть проведено обоснование ***структуры и технологии функционирования программно-аппаратной системы обеспечения безопасности***, определена общая структура системы, состав ее ядра и компонентов. Следует определить комплекс организационного, программного и информационного обеспечения, обосновать технологические схемы функционирования системы безопасности во всех режимах обработки информации, оперативного и административного управления системой.

При проектировании целесообразно ***разделять вычислительные ресурсы***, необходимые для непосредственного решения основных, ***функциональных задач системы*** (функциональной пригодности), и ресурсы, требующиеся ***для безопасного функционирования ПП***. Соотношение между этими видами ресурсов в реальных системах зависит от сложности и состава решаемых функциональных задач, степени их критичности и требований к безопасности всей системы. Их реализация в критических системах, может потребовать значительных ресурсов памяти и производительности вычислительных средств, которые в особых случаях могут превышать ресурсы, используемые на решение основных, функциональных задач в 2 – 4 раза. В более простых системах средства обеспечения безопасности обычно используют 10 –20% всех видов трудовых, аппаратных и вычислительных ресурсов.

Организационной основой управления безопасностью ПП на базе стандартов жизненного цикла (см. гл. 3) является ***план обеспечения заданных характеристик безопасности*** на всех этапах ЖЦ комплекса программ. Для этого до начала разработки в процессе формирования требований технического задания следует сформулировать основные положения методики обеспечения безопасности, разработки и поэтапных испытаний компонентов и определения требуемых значений характеристик, допустимых для продолжения работ на следующих этапах. Реальные ограничения ресурсов, используемых в процессе разработки, квалификация специалистов, изменения внешней среды и требований заказчика объективно приводят к отклонениям процессов и реализации плана, от предполагавшегося. Величина таких отклонений в значительной степени зависит от принятой технологии разработки, от уровня и характеристик инструментальных средств автоматизации создания программ. Для своевременного обнаружения отклонений от плана необходимо регулярно регистрировать результаты выполненных работ и их качество. Для реализации таких изменений целесообразно предусмотреть и согласовать с заказчиком специальный документ, регламентирующий правила корректировки плана обеспе-

чения безопасности и качества ПП, а также состав и содержание поддерживающей его документации.

Проекты, как правило, первоначально инициируются с ***объемом функциональных возможностей ПП, превышающим*** тот, который разработчик может реализовать, обеспечив приемлемую безопасность системы. Поэтому целесообразно определять, что ***обязательно должно*** быть сделано разработчиком в очередной версии ПК при имеющихся ресурсах проекта. Для этого ***следует вести переговоры с заказчиком***. Привлечение заказчика к решению проблемы управления масштабом и уровнем безопасности ПП повышает взаимные обязательства сторон, способствует росту взаимопонимания и доверия между заказчиком и разработчиками. Зачастую имея концепцию безопасности продукта и сократив масштаб проекта до разумного уровня, можно надеяться на успех в следующих фазах или версиях проекта. На основе выполненных оценок трудозатрат стандартами рекомендуется определять ***базовый уровень безопасности для*** каждой версии требований, достигнуть соглашения с заказчиком относительно масштаба ПК***,*** а также принять жесткие решения по реализуемому масштабу версии. ***Итеративная разработка*** и управление изменениями, используя базовый уровень безопасности, позволяет контролировать, что все предложенные функции реализованы.

Технологию и жизненный цикл ПП следует оформить документами, которые необходимо проанализировать на реализуемость и утвердить. В планах должны определяться организационные подразделения и специалисты, которые будут исполнять различные виды работ. Стандартами (см. гл. 3) рекомендуется в процессе планирования ЖЦ ПК ***подготовить и утвердить содержание*** следующих планов (рис. 2.2):

- разработки и реализации всего ЖЦ ПК, который должен определять используемую модель жизненного цикла комплекса программ и его компонентов, а также внешнюю и технологическую среду проектирования;
- верификации и тестирования, который определяет методы и средства, способные удовлетворить последовательные цели процесса устранения дефектов, контроля достигнутой безопасности и качества ПК и его компонентов;
- реализации процессов интеграции компонентов в версии комплекса программ безопасности;
- управления конфигурацией и сопровождения ПК, при помощи которого будут удовлетворяться цели процесса последовательного совершенствования, управления изменениями и корректировками программ;
- тиражирования, адаптации и внедрения версий ПК для конкретных пользователей, включая их подготовку и обучение;

- документирования процессов и результатов жизненного цикла ПК, создания и выпуска технологической и эксплуатационной документации;
- управления и обеспечения безопасности ПК, определяющих методы и средства, при помощи которых будет гарантировано требуемая безопасность при развитии комплекса программ.

Цель ***планирования технологической среды жизненного цикла ПК*** состоит в том, чтобы определить методы, инструментальные средства, процедуры, языки программирования и аппаратные средства, которые будут использоваться и совершенствоваться, для разработки, верификации, управления и подготовки документации программного средства. План должен включать стандарты, методы предотвращения ошибок и обеспечения безопасности, которые ограничивают возможность внесения ошибок, и такие методы тестирования, которые гарантируют их обнаружение. Цель методов обеспечения безопасности состоит в том, чтобы включить в проект такие средства, которые могут гарантировать, что ПК будет адекватно реагировать на ошибки входных данных, предотвращать выдачу ошибочных данных и контролировать возможность проявления ошибок.

В составе перечисленных планов или автономно может быть полезной ***разработка вспомогательных планов*** (см. рис. 2.2.): плана обучения и подготовки пользователей для квалифицированной и безопасной эксплуатации версий ПП; плана обслуживания пользователей в процессе безопасной эксплуатации ПП. Эти планы должны иметь в своем составе графики, идентифицирующие: этапы работ; входные, выходные данные и описания решаемых задач; необходимые ресурсы и сроки выполнения; взаимосвязи этапов и работ. В планах работ поставщиков и субподрядчиков следует четко определить границы ответственности за каждую часть программного комплекса и за способ обмена технической информацией между всеми сторонами проекта.

Планирование создания программ обеспечения функциональной безопасности целесообразно отделять от непосредственного управления процессами создания и совершенствования основных функций комплексов программ. Это позволяет сосредоточить внимание выделенных специалистов на совокупности мероприятий и средств, гарантирующих безопасность конечного продукта. Чтобы такие гарантии достигались при минимальных затратах, необходимы целенаправленное, координируемое планирование и управление для предотвращения ошибок проектирования, а также для выявления и устранения дефектов проекта на самых ранних этапах разработки.

Цель ***планирования технологической среды жизненного цикла ПК*** состоит в том, чтобы определить методы, инструментальные средства, процедуры, языки программирования и аппаратные средства, которые будут использоваться и совершенствоваться, для разработки, верификации, управления и подготовки документации программного средства.

План должен включать стандарты, методы предотвращения ошибок и обеспечения безопасности, которые ограничивают возможность внесения ошибок, и такие методы тестирования, которые гарантируют их обнаружение. Цель методов обеспечения безопасности состоит в том, чтобы включить в проект такие средства, которые могут гарантировать, что ПК будет адекватно реагировать на ошибки входных данных, предотвращать выдачу ошибочных данных и контролировать возможность проявления ошибок.

В составе перечисленных планов или автономно может быть полезной ***разработка вспомогательных планов*** (см. рис. 2.2): плана обучения и подготовки пользователей для квалифицированной и безопасной эксплуатации версий ПП; плана обслуживания пользователей в процессе безопасной эксплуатации ПП. Эти планы должны иметь в своем составе графики, идентифицирующие: этапы работ; входные, выходные данные и описания решаемых задач; необходимые ресурсы и сроки выполнения; взаимосвязи этапов и работ. В планах работ поставщиков и субподрядчиков следует четко определить границы ответственности за каждую часть программного комплекса и за способ обмена технической информацией между всеми сторонами проекта.

Рис. 2.2

Анализ состояния и результатов проекта следует официально планировать и регулярно проводить на соответствующих этапах ЖЦ ПК. Перед тем, как

предложить продукт потребителю на испытания и приемку, поставщик должен оценить его качество и безопасность согласно назначению и спецификациям требований.

Планирование создания программ обеспечения функциональной безопасности целесообразно отделять от непосредственного управления процессами создания и совершенствования основных функций комплексов программ. Это позволяет сосредоточить внимание выделенных специалистов на совокупности мероприятий и средств, гарантирующих безопасность конечного продукта. Чтобы такие гарантии достигались при минимальных затратах, необходимы целенаправленное, координируемое планирование и управление для предотвращения ошибок проектирования, а также для выявления и устранения дефектов проекта на самых ранних этапах разработки. Следует учитывать ***глубокую взаимосвязь плана обеспечения безопасности ПП с планом управления непосредственной разработкой программ,*** с процессами тестирования, испытаний и сертификации ПП. Эта связь выражается в аналогичном поэтапном представлении планов и в наличии в них значительной части близких по содержанию процессов и документов.

В общем случае ***в технологическом процессе обеспечения безопасности и качества ПП и системы*** следует учитывать:

- анализ контракта и спецификаций требований заказчика к ПК, выделение и ранжирование приоритетов характеристик и атрибутов безопасности и качества конечного продукта;
- декомпозицию требований к характеристикам безопасности и качества по контролируемым этапам и компонентам разработки и создание разделов по детальным требованиям в спецификациях на программные компоненты и ПК в целом;
- выбор или создание методов, технологии и инструментальных средств автоматизации разработки, обеспечивающих создание ПК и его компонентов с требуемыми характеристиками безопасности и качества;
- разработку методик контроля соблюдения стандартов, правил технологии проектирования и системы обеспечения качества жизненного цикла программных продуктов;
- создание методов, методик и средств объективного измерения свойств и/или значений атрибутов характеристик безопасности и качества программных компонентов на этапах их создания и всего ЖЦ ПК для испытаний заказчиком и эксплуатации пользователями;
- организацию, обучение и стимулирование коллектива специалистов на создание компонентов и ПК в целом, в максимальной степени удовлетворяющих требования заказчиков и пользователей к характеристикам функциональной безопасности и качества.

Утверждение и выпуск документации о достигнутой безопасности и качестве ПП должны контролироваться уполномоченным персоналом на пред-

мет их адекватности. Следует разработать и поддерживать в актуальном состоянии процедуры управления документами, которые идентифицируют текущий статус корректировки и пересмотра документов, с тем, чтобы предотвратить использование недействующих и/или устаревших документов; немедленное их изъятие из всех пунктов их рассылки или применения. Изменения документов должны утверждаться теми же службами, которые проводили первоначальный анализ и утверждали эту документацию.

2.3. Процессы разработки требований к функциональной безопасности и качеству программных продуктов

При разработке и реализации ***требований к функциональной безопасности и качеству ПП*** необходимо, в первую очередь, учитывать следующие основные факторы:

- функциональную пригодность (функциональность) конкретного проекта ПП, объекта или системы;
- возможные конструктивные характеристики качества и безопасности комплекса программ, необходимые для улучшения функциональной пригодности;
- доступные ресурсы для создания и обеспечения всего жизненного цикла ПП с требуемой функциональной безопасностью и качеством.

При первоначальном определении требований к функциональной пригодности и к безопасности, заданные заказчиком ограничения ресурсов не всегда могут учитывать ряд особенностей проекта, что может обусловить недопустимое снижение (или завышение) требований к некоторым характеристикам качества ПК. Кроме того, возможно, что некоторые конструктивные характеристики противоречивы или принципиально нереализуемы в данном проекте. В результате ***не сбалансированные требования*** к безопасности и качеству проявятся как ущерб в виде потерь в качестве или в перерасходовании ресурсов. Для устранения или снижения ущерба до допустимых пределов потребуется изменение требований к функциональной пригодности, к безопасности и/или к некоторым конструктивным характеристикам. Таким образом, целесообразно анализ и разработку требований к ПП проводить в два этапа: предварительно ***максимизируя*** функциональную пригодность и безопасность, а затем ***минимизируя*** ущерб путем снижения требуемого качества или используемых ресурсов.

Разработку и утверждение требований к безопасности и качеству, целесообразно проводить ***итерационно на этапах системного и детального проектирования ПК***. Полная и однократная формализация требований к характеристикам качества в начале жизненного цикла сложного ПП обычно невозможна, прежде всего, из-за разных представлений заказчика и разработчиков о деталях назначения, функций и возможностей реализации их при доступных

ресурсах. Чем крупнее и сложней проект ПК и соответственно выше его стоимость (трудоемкость), тем тщательнее следует разрабатывать требования к его характеристикам и распределять ресурсы на их реализацию. Поэтому при средней и относительно невысокой сложности ПК во многих случаях можно удовлетвориться подготовкой требований к безопасности с подробностью анализа, соответствующей системному проектированию. Для особо сложных проектов необходим более детальный анализ факторов при разработке требований и их оптимизация по критерию качество/затраты.

В зависимости от сложности проекта, окончательным результатом работ при системном или детальном проектировании, должны быть ***детализированные и утвержденные требования,*** к номенклатуре, свойствам и значениям функциональной пригодности, к атрибутам функциональной безопасности и качества ПП, которые, в обоих случаях, достаточны для его полноценного рабочего проектирования и последующей, эффективной эксплуатации. В реальных проектах при подготовке требований к ПК могут исключаться этапы системного проектирования, однако содержание и последовательность работ по анализу и детализации требований к характеристикам ПП целесообразно сохранять. Эти ***требования закрепляются в контракте и техническом задании***, по которым разработчик впоследствии ***должен отчитываться перед заказчиком*** при завершении проекта. Однако на последующих этапах жизненного цикла и при конфигурационном управлении, требования могут изменяться по согласованию между заказчиком и разработчиком, которые чаще всего приурочиваются к подготовке новой версии ПК. Для этого необходим мониторинг требований и технологии реализаций характеристик в течении всего ЖЦ.

Выбор требований к характеристикам качества при проектировании программных продуктов начинается с ***определения исходных данных***. Для корректного выбора и установления требований к безопасности и характеристикам качества, прежде всего, необходимо определить базовые особенности проекта (рис. 2.3):

- класс, назначение и основные функции создаваемого ПП;
- комплект стандартов и их содержание, которые целесообразно использовать при выборе характеристик и технологии обеспечения функциональной безопасности и качества ПК;
- состав возможных потребителей ПП, для которых важны функциональная безопасность и соответствующие атрибуты качества;
- реальные ограничения всех видов ресурсов проекта.

Этап разработки концепции проекта целесообразно начинать с формализации и обоснования набора исходных данных, отражающих общие особенности класса, потребителей и этапов жизненного цикла проекта ПП или системы, каждый из которых влияет на выбор определенных характеристик безопасности и качества комплекса программ. Из конструктивных характеристик и атрибутов качества, прежде всего, следует выделить, те на которые в наиболь-

шей степени воздействует класс, назначение и функции ПК. Для этого первоначально целесообразно использовать классификацию программных средств по стандарту **ISO 12182** и всю базовую номенклатуру характеристик качества, стандартизированных в **ISO 9126:1-4**. Их описания желательно предварительно упорядочить по приоритетам с учетом особенностей назначения и сферы применения конкретного ПП и системы.

Далее необходимо ***выделить и ранжировать по приоритетам потребителей***, которым необходимы определенные показатели безопасности и качества ПП с учетом их специализации и профессиональных интересов. Широкая номенклатура характеристик, представленная в стандарте **ISO 9126:1-4,** определяет разнообразные требования, из которых следует селектировать и выбирать те, которые необходимы с позиции потребителей этих данных:

- - ***пользователей***, для которых необходима функциональная пригодность, корректность, безопасность, надежность и другие показатели качества при оперативном использовании ПП по основному назначению;
- ***разработчиков***, для которых особенно важны: ясность и конкретность описаний требований к функциям и характеристикам ПК, его возможная архитектура и интерфейсы между компонентами и с внешней средой;
- ***специалистов сопровождающих и модифицирующих ПК***, которые отдают приоритет характеристикам, поддерживающим изменения и конфигурационное управление версиями комплекса программ и его компонентов.

После определения назначения и функций ПК подготовка исходных данных и концепции проекта должны завершаться ***выделением номенклатуры приоритетных характеристик безопасности и качества***, имеющих достаточное влияние на функциональную пригодность ПП для определенных потребителей.

На ***этапе системного проектирования*** (см. рис. 2.3), после фиксирования исходных данных и приоритетов характеристик для конкретного проекта и его потребителей, начинаются выбор требований к свойствам и значениям безопасности, а также установление и утверждение конкретных мер и шкал характеристик и атрибутов качества. Такой анализ должны проводить заказчик и некоторые потенциальные пользователи совместно со специалистами, обеспечивающими технологию ЖЦ комплекса программ и реализацию установленных требований к показателям безопасности и качества. Этими специалистами для каждого из выбранных атрибутов должна быть установлена и согласована мера и шкала оценок характеристик и их атрибутов для конкретного проекта и потребителя результатов анализа. Для показателей, представляемых качественными свойствами и признаками их наличия, желательно определить и зафиксировать в спецификациях описания допустимых условий, при которых следует считать, что данная характеристика может или должна быть реализована в конкретном проекте ПП.

Рис.2.3.

Требования к безопасности и значениям характеристик качества должны быть предварительно проверены разработчиками на их реализуемость с учетом доступных ресурсов конкретного проекта и при необходимости откорректированы по составу и значениям. В зависимости от ограниченности ресурсов проекта ПК распределение приоритетов должно становиться более строгим и могут снижаться приоритеты характеристик, для реализации которых ресурсов недостаточно. В результате формируется полный ***набор требуемых характеристик, свойств и атрибутов, их мер и значений безопасности и качества для определенных потребителей***.

Результаты системного анализа и выбора номенклатуры и мер характеристик функциональной безопасности и качества проектов средней или относительно невысокой сложности должны быть документированы в спецификациях требований, согласованы с их потребителями и утверждены заказчиком системного проекта для последующей реализации. Для разработчиков особенно важно формализовать требования к технологии обеспечения безопасности и согласовать их с заказчиком при утверждении контракта и технического задания на проект ПП. Требования к характеристикам безопасности и качества, утвержденные после системного проектирования, могут быть закреплены в техническом задании ***как обязательные для детального и рабочего проектирования***.

При детальном проектировании может быть целесообразно дополнительное уточнение совокупности выбранных характеристик и технологии обеспечения функциональной безопасности сложных ПК с учетом соотношения достигаемого качества и затрат ресурсов. Определение отдельных характеристик проекта ПК может быть в значительной степени объективным, даже при их экспертном анализе. Однако оценивание важности или приоритета ***совокупности характеристик*** или их атрибутов для программного компонента или проекта в целом, более субъективный процесс, зависящий от мнений экспертов и потребителей их значений. Для заказчиков и пользователей интегральное качество сосредоточивается на функциональной пригодности, безопасности и метриках в использовании.

Каждая характеристика качества и затраты ресурсов первоначально ***анализируются независимо***, что может использоваться в качестве исходных данных для их сопоставления с отдельными характеристиками аналогичных ПК. Обычно заказчики и разработчики первоначально устанавливают завышенные требования к каждой характеристике качества без учета относительных затрат на их достижение, а также без детального анализа их совместного влияния на полную функциональную пригодность у потребителей. Это может приводить к ***несбалансированным значениям требований*** к отдельным, взаимосвязанным характеристикам, на которые не рационально используются ограниченные ресурсы, или к не адекватно низким их значениям. В проектах сложных ПК это может угрожать значительным повышением стоимости и/или снижением конкурентоспособности создаваемого программного продукта из-за недостаточного уровня отдельных показателей качества.

Для обобщенного оценивания качества ПП необходим учет относительного влияния каждой характеристики на функциональную пригодность. При этом не всегда учитываются ресурсы для их реализации в конкретном ПП. Это часто приводит к выдвижению ряда нерациональных требований, которые значительно отличаются: либо по степени влияния на функциональную пригодность, либо по величине ресурсов, необходимых для их реализации. Для целенаправленного эффективного управления безопасностью и качеством сложного ПП при проектировании, целесообразно иметь механизм объединения разнородных характеристик в некоторый ***интегральный показатель***, отражающий их совокупное влияние на функциональную пригодность. Таким образом, при разработке требований выявляется проблема анализа системной эффективности ПП и обобщения его характеристик, а также ***оценивания совместного влияния безопасности и различных характеристик качества на функциональную пригодность ПП с учетом затрат*** на их реализацию.

Для конкретного проекта ПК состав и значения приоритетов следует поэтапно адаптировать и уточнять с учетом их назначения и функций. Наивысший приоритет следует интерпретировать как обязательное выполнение разработчиком соответствующего требования к указанному свойству или атрибуту качества. Низшее значение приоритета означает, что данный показатель может не учитываться в данном проекте. Промежуточные значения приоритетов должны отражать относительное влияние соответствующих атрибутов на функциональную пригодность и ее свойства с учетом доступных ресурсов на их реализацию.

Эти данные могут использоваться, прежде всего, ***как ориентиры*** для селекции и исключения из требований, атрибутов качества с особенно низкими обобщенными приоритетами, в наименьшей степени влияющих на функциональную пригодность и безопасность ПП и не оправдывающих больших затрат на реализацию. Анализ оставшихся атрибутов качества может проводиться для выделения завышенных требований, а также, возможно, для снижения их значений и приближения их влияния к средним значениям.

Для ***сравнения качества при выборе предпочтительного из конкурирующих проектов ПК,*** при одинаковом или подобном содержании функциональной пригодности может быть полезно использовать интегральную характеристику – отношение качество/затраты. Для такого сравнения следует рассчитать сумму уровней обобщенных приоритетов безопасности и качества с учетом затрат в зависимости от содержания проекта ПК. Эта сумма обобщенных приоритетов в некоторой степени отражает полезный эффект от реализованного качества ПП при соответствующих затратах и может рассматриваться как интегральный критерий ***– качество/затраты***, что близко к традиционному понятию маркетингового критерия качества продукции по показателю ***эффективность/стоимость***. Такие интегральные оценки требований вряд ли целесообразно помещать в контракты и технические задания вследствие их субъективности и относительности, однако они могут быть весьма полезными ***ориенти-***

рами при управлении технологией и поставках сложных ПП в условиях необходимости использования особенно больших ресурсов.

Дефекты в ЖЦ ПП могут быть обусловлены недостатками деятельности различных лиц, участвующих в создании или применении программного продукта. Основными их источниками, которые могут приводить к ущербу при его разработке и применении, являются:

- ***заказчики*** программного продукта, которые могут задать некорректные или нереализуемые разработчиками требования к нему, а также ограничивают выделенные и доступные для проекта ресурсы и технологию;
- ***разработчики*** проекта ПП и специалисты, обеспечивающие его весь ЖЦ, которые могут не выполнить согласованные требования заказчика к безопасности и качеству комплекса программ и/или превышают допустимое использование ресурсов, что может отражаться на образовании, проявлении и величине дефектов на определенных технологических этапах.

Косвенными источниками и причинами недостатков безопасности могут быть также ***пользователи***, некомпетентно применяющие программный продукт с отклонениями от требований документации по функциональной пригодности, обеспечению безопасности или с недопустимым использованием ресурсов при эксплуатации ПП.

Для обеспечения требуемой безопасности и качества ПП необходима организация процесса управления проектированием под руководством ***менеджера управления*** – координатора взаимодействия заказчика и разработчиков (см. рис. 1.3). Это лицо должно быть уполномочено заказчиком принимать решения, о допустимости применения системы и ПП с достигнутыми, конкретными уровнями качества и/или о необходимости их снижения, путем применения необходимых контрмер.

При начальном формировании требований к характеристикам комплекса программ практически невозможно достоверно предусмотреть сбалансированное выделение каждого вида ресурса для полной реализации безопасности и каждой требуемой характеристики качества. При этом некоторые характеристики в реальном проекте могут приобретать значения более высокие, чем действительно требуются, на что нерационально расходуются ресурсы, а другие – не удовлетворять требованиям контракта и технического задания. Для разрешения этого противоречия основное значение имеет деятельность ***независимого менеджера*** или координационного совета, которые способны проводить поэтапный анализ, контроль, оценивание и мониторинг отклонений от требуемых характеристик и используемых ресурсов, ***управление контрмерами и последовательное изменение их для сокращения и минимизации дефектов*** всей системы. При таких стратегиях ***наиболее стабильными должны быть требования к функциональной пригодности.*** Для этого в ***технологическом***

процессе обеспечения безопасности необходимо использовать ***методы, которые включают***:

- выделение и идентификацию атрибутов функциональной пригодности, требуемых для конкретного проекта системы и ПП;
- систематизацию, документирование и оценивание эффективности доступных методов, средств и ресурсов контрмер для обеспечения функциональной пригодности и выделенных характеристик безопасности и качества;
- оценивание вероятности каждого вида угроз безопасности и качеству, потенциальной величины и вероятности их возможного негативного воздействия на каждую характеристику функциональной пригодности системы и ПП;
- оценивание уязвимости каждой характеристики безопасности и качества и затрат ресурсов для восстановления требуемой функциональной пригодности системы и ПП;
- планирование и разработку решений по контрмерам для обеспечения допустимого уровня функциональной пригодности и других характеристик качества системы, в том числе возможно за счет изменения требований к системе и ПП и доступных ресурсов.

Полное оценивание интегрального качества и затрат, мониторинг и оптимизация функциональной безопасности проектов сложных систем и ПП достаточно трудоемкий процесс, который требует привлечения квалифицированных экспертов – менеджеров, хорошо понимающих цель и функциональную сущность анализируемой системы и ее программ. Такой анализ может быть рентабельным и положительно влиять на обеспечение функциональной безопасности для особо сложных систем и ПП и на сокращение больших затрат ресурсов при их реализации. Поэтому перед выполнением подобного анализа необходимо, в свою очередь, предварительно оценивать его возможную трудоемкость и экономическую рентабельность.

Высокий уровень технологии способствует сокращению дефектов и ошибок в программах и тем самым повышению функциональной безопасности систем. Однако при этом возрастают затраты на технологический инструментарий и его применение в жизненном цикле. Поэтому для обеспечения высокой функциональной безопасности ПП и систем необходимо уделять большое внимание выбору уровня технологии и его влиянию на потенциальные дефекты и ошибки в программах с учетом их сложности и других характеристик (см. гл. 3 и Приложение). Применение изложенных ниже методов и стандартов обеспечения функциональной безопасности ПП и систем всегда должно учитывать (может быть не явно) затраты ресурсов, которые требуются для их реализации и применения.

Глава 3. Стандартизация технологических процессов обеспечения функциональной безопасности в жизненном цикле программных комплексов

Введение к стандартам

Выше в главе 1 отмечалось, что высокое качество при создании объектов и систем может быть обеспечено измерением их реальных характеристик и отбраковкой не удовлетворяющих требованиям, или применением высококачественных технологий и процессов, поддерживающих их жизненный цикл. В этом случае не всегда известно, какое качество объектов достигнуто, но применены соответствующие, возможные и доступные методы и усилия, чтобы качество было максимально высоким. Прямое измерение высокой достигнутой функциональной безопасности объектов и систем сталкивается с большими трудностями, прежде всего вследствие необходимости фиксировать редкие, непредсказуемые отказовые ситуации с неожиданным ущербом. Поэтому ***основное содержание стандартов и нормативных документов в области функциональной безопасности в основном составляют методы, процессы и технологии обеспечения высокой безопасности ПП и систем в жизненном цикле*** и почти не уделяется внимание определению при этом значений достигаемой безопасности.

Встроенные программные продукты, в частности, реализующие функции обеспечения безопасности управления динамическими объектами и системами, используемыми в авиации, космосе, на транспорте, создаются с применением технологий и инструментальных средств, по существу, аналогичных применяемым для разработки других классов сложных ПП реального времени. Наиболее общими и широко применяемыми стандартами, регламентирующими процессы жизненного цикла крупных ПК высокого качества является ISO 12207:2008 и целая серия стандартов, детализирующих и конкретизирующих базовые процессы этого стандарта (см. Приложение 1). Однако в этих стандартах непосредственно безопасности и защите уделяется относительно мало внимания, и специфика измерения функциональной безопасности сложных ПК обычно не выделяется и не комментируется. В то же время эти стандарты целесообразно адаптировать и применять при создании ПП, реализующих функции безопасности, что в частности поддерживается ссылками на них в стандартах, посвященных непосредственно безопасности ПП, объектов и систем, реализуемых на базе компьютеров [33, 34].

Систематично и подробно процессы ЖЦ сложных ПК представлены в базовом стандарте **ISO 12207:2008,** и в свыше десяти сопутствующих ему стандартах,

детализирующих и углубляющих требования и процессы ЖЦ ПК (см. [24, 26] и Приложение 1). Этот комплекс стандартов целесообразно учитывать при создании и применении ПП и систем, обеспечивающих функциональную безопасность. Поэтому ниже приведены краткие аннотации этих стандартов (см. п. 3.4 и 3.5), которые дают представление об особенностях, рекомендуемых в них процессах жизненного цикла ПК и систем. Содержание большинства представленных в данной главе ***стандартов весьма близки по существу, однако значительно различается терминологией и структурой.***

3.1. Процессы обеспечения функциональной безопасности программных продуктов в стандарте IEC 61508:1-6: 1998-2000

Технология создания и всего жизненного цикла комплексов программ для ***обеспечения функциональной безопасности специализированных компьютеров***, встроенных в аппаратуру объектов и систем наиболее четко представлена в стандарте **IEC 61508:1-6: 1998 –** ***Функциональная безопасность электрических / электронных / программируемых электронных систем безопасности***. Третья часть стандарта **IEC 61508-3 –** Требования к программному обеспечению **–** устанавливает общий подход ко всем видам деятельности на протяжении цикла обеспечения безопасности для систем, содержащих программируемые электронные компоненты (ПЭС), которые используются для выполнения различных функций и, в частности, для обеспечения безопасности систем. Этот унифицированный подход рекомендован с целью выработки последовательной политики безопасности для всех систем, основанных на компьютерах. ***Любая стратегия функциональной безопасности*** должна учитывать не только все элементы в каждой конкретной системе (например, датчики, управляющие устройства и исполнительные механизмы), но, также, все системы и программные компоненты, обеспечивающие безопасность, и создавать суммарную комбинацию систем, обеспечивающих безопасность. Стандарт **IEC 61508-3** ***содержит следующие особенности*** [35]:

- рассматривает все этапы циклов обеспечения общей функциональной безопасности ПЭС и программных комплексов;
- учитывает быстро развивающиеся технологии; его структура достаточно прочная и всеобъемлющая для того, чтобы обслуживать будущие разработки;
- позволяет разработать область применения международных стандартов; обеспечивает высокий уровень согласованности (например, основополагающих принципов, терминологии и т.п.) как внутри каждой области применения ПЭС, так и между областями применения;
- обеспечивает метод разработки спецификаций требований по безопасности ПЭС, необходимых для достижения заданной функциональной безопасности;

- использует основанный на рисках подход для определения требований к уровням соответствия комплексу требований по функциональной безопасности;
- устанавливает количественные меры отказов для систем безопасности ПЭС, связанные с уровнями соответствия комплексу требований по функциональной безопасности;
- программные комплексы, обеспечивающие безопасность включают операционные системы, системное программное обеспечение, программное обеспечение в коммуникационных сетях, функции интерфейса человек-машина, инструментальные средства поддержки и программно-аппаратные средства, а также прикладные функциональные программы;
- устанавливает требования для этапов жизненного цикла обеспечения безопасности и деятельности, которая должна проводиться при проектировании и разработке ПК, обеспечивающего безопасность (модель жизненного цикла обеспечения безопасности);
- обеспечивает требования для информации, относящейся к аттестации ПП, которая должна передаваться организациям, производящим компоновку ПЭС в составе системы;
- обеспечивает требования по подготовке информации и процедур, относящихся к ПК, необходимых пользователю для эксплуатации и обслуживания систем безопасности;
- обеспечивает требования к технологическим средствам поддержки, таким как инструментальные средства проектирования и разработки, языковые трансляторы, инструментальные средства для тестирования, отыскания и устранения ошибок, средства управления конфигурацией.

Планирование функциональной безопасности ПК должно определять стратегию получения, разработки, компоновки, верификации, аттестации и модификации комплекса программ в той мере, которая необходима для требуемого уровня соответствия ПЭС комплексу требований по безопасности.

Требования к циклу обеспечения безопасности программных продуктов (рис. 3.1)

1. Общие положения. Цикл обеспечения безопасности при разработке ПС должен быть выбран и задан при планировании безопасности, приспособлен для практических нужд проекта или организации. В деятельность, связанную с циклом обеспечения безопасности, должны быть включены процедуры гарантирования качества и безопасности. Каждый этап цикла обеспечения безопасности ПК должен быть подразделён на элементарные операции в рамках сферы действия, а также входных и выходных данных, оговоренных для каждого этапа. Полный перечень этапов цикла обеспечения безопасности подходит для больших вновь разрабатываемых систем. В малых системах может

оказаться целесообразным объединять этапы проектирования ПК и архитектурного проектирования объекта или системы. Рекомендуется использовать основные положения, комплекс этапов и работ, представленный в стандарте **ISO 12207:2008**. Для каждого этапа цикла обеспечения безопасности должны использоваться соответствующие методы и меры. По результатам деятельности в рамках цикла обеспечения безопасности ПП должна быть составлена документация. Если на любом этапе обеспечения безопасности ПП потребуется изменение, относящееся к более раннему этапу, то более ранний и последующие этапы цикла обеспечения безопасности должны быть повторены.

2. Спецификация требований по безопасности программного продукта. Спецификация должна основываться на заданных требованиях по безопасности системы, общего и специального прикладного ПК. Спецификация требований по безопасности должна быть достаточно подробной для того, чтобы конструкция и ее выполнение достигали требуемого соответствия комплексу требований по безопасности, и можно было произвести оценку функциональной безопасности. В частности, разработчик ПП должен учитывать: функции безопасности; конфигурацию или архитектуру системы; производительность и время срабатывания; взаимодействие между оборудованием и оператором. Эти требования в большинстве случаев ***должны достигаться комбинацией основного функционального программного комплекса и специального ПК обеспечения безопасности***:

- обеспечивающим достижение и поддержание безопасного состояния объекта или системы;
- относящимся: к выявлению отказов; извещению о их наличии и управлению отказами в аппаратуре программируемой электроники;
- отказами датчиков и исполнительных механизмов, а также в самом программном продукте;
- относящимся, к периодической проверке функций обеспечения безопасности в оперативном режиме и в отключённом состоянии;
- производительности и времени срабатывания основных функций ПП.

3. Планирование аттестации программного комплекса. Должно быть произведено планирование с целью определения шагов, как процедурных, так и технических, которые должны быть использованы для демонстрации того, что программный продукт удовлетворяет требованиям по его безопасности. План аттестации безопасности ПП должен содержать следующее: подробные указания о том, когда должна производиться аттестация; кто должен производить аттестацию; определение ответственных режимов эксплуатации технических средств; техническую стратегию аттестации; требуемые условия окружающей среды, в которой должна производиться аттестация.

4. Проектирование и разработка программного продукта

4.1. Цели. Создание архитектуры программного комплекса, которая отвечает заданным требованиям по безопасности, рассмотрение и оценка требований, возлагаемых на ПП со стороны архитектуры аппаратуры системы безопасно-

сти, включая взаимодействие аппаратуры и программного комплекса для обеспечения безопасности объекта или системы.

Требования к циклу обеспечения безопасности программного продукта

1. Общие положения
2. Спецификация требований по безопасности программного продукта
3. Планирование аттестации программного продукта
4. Проектирование и разработка программного продукта:
   - 4.1. Цели
   - 4.2. Общие требования
   - 4.3. Требования к архитектуре программного продукта
   - 4.4 Требования к инструментальным средствам поддержки и языкам программирования
   - 4.5 Требования к рабочему проекту и разработке
   - 4.6 Требования к кодированию
   - 4.7 Требования к испытаниям модулей программного комплекса
   - 4.8 Требования к компоновочным испытаниям программного комплекса
5. Компоновка программируемой электроники
6. Процедуры эксплуатации и обслуживания программного продукта, обеспечение безопасности
7. Аттестация программного продукта
8. Модификация программного продукта
9. Верификация программного продукта

Оценка функциональной безопасности

Рис.3.1.

Выбор подходящего для требуемого уровня соответствия комплексу требований по безопасности, комплекта инструментальных средств (включая языки и компиляторы), который способствовал бы проведению верификации, аттестации, оценки и модификации на протяжении всего цикла обеспечения безопасности ПП. Проектирование и изготовление ПК, отвечающего заданным требованиям по безопасности в отношении требуемого уровня соответствия комплексу требований по безопасности системы, которое можно проанализировать и проверить и безопасно модифицировать.

4.2. Общие требования. В соответствии с требуемым уровнем комплекса требований по безопасности, выбранный метод проектирования должен обладать свойствами, которые способствуют: абстрагированию, модулированию и другим методам, влияющим на сложность разработки. Выбранный метод проектирования должен обладать свойствами, облегчающими модификацию программного продукта. Такие свойства включают модульность, сокрытие информации и формирование пакетов данных. Насколько это возможно практически, конструкция должна сводить к минимуму ту часть ПП, которая относится к обеспечению безопасности. Конструкция должна включать функции программного комплекса по проведению контрольных испытаний и всех диагностических процедур с тем, чтобы выполнять требования по соответствию комплекса требований по безопасности системы. Конструкция ПК должна включать соответствующий уровню комплекса требований по безопасности, самоконтроль потока управления и потока данных. При выявлении неисправностей должны предприниматься соответствующие контрмеры.

4.3. Требования к архитектуре программного комплекса. Архитектура ПП определяет основные компоненты и подсистемы комплекса программ, как они соединяются между собой, и как могут быть достигнуты требуемые свойства, в частности, соответствие комплексу требований по безопасности. Основные компоненты ПК включают: операционные системы, базы данных, внутренние подсистемы ввода/вывода, коммуникационные подсистемы, прикладные программы, инструментальные средства программирования и диагностики и т.п. В крайнем случае, с использованием языков широкого применения, архитектура системы должна создаваться поставщиком специально для этого применения (или этого класса применений). С точки зрения безопасности создание архитектуры ПК является этапом, на котором разрабатывается стратегия основной безопасности комплекса программ и системы. Конструкция архитектуры программного комплекса должна устанавливаться поставщиком и/или разработчиком, при этом должно быть создано подробное описание конструкции. Описание должно быть основано на подразделении на компоненты/подсистемы, для каждой из которых предоставлена следующая информация;

- являются ли они новыми, существующими или собственными;
- проходили ли они ранее верификацию, и если да, условия проведения их верификации;
- относится ли каждая подсистема/компонент к обеспечению безопасности или нет;
- уровень соответствия ПП комплексу требований по безопасности для каждой подсистемы/компонента;
- определять все взаимодействия ПК и аппаратуры и подробно оценивать их значение;

- использовать для представления архитектуры систему обозначений, которая однозначно определяет или ограничивается однозначно определяемыми свойствами.

4.4. Требования к инструментальным средствам поддержки и языкам программирования. Выбор инструментальных средств разработки зависит от характера деятельности по разработке и от архитектуры ПК. Для требуемого уровня безопасности должен быть выбран соответствующий комплект технологических средств, включая языки, компиляторы, инструментальные средства управления конфигурацией, и при необходимости, автоматические испытательные средства. Необходимо рассмотреть наличие подходящих инструментальных средств (не обязательно таких, какие применялись при первоначальной разработке системы) для обеспечения необходимых действий на протяжении всего срока службы системы безопасности.

4.5. Требования к рабочему проекту и разработке ПП. Характер рабочего проекта и процесса разработки может изменяться в зависимости от архитектуры ПК. В случае прикладного программирования пользователем с использованием языка ограниченного применения, например, многоступенчатой логики и функциональных блоков, рабочий проект может рассматриваться скорее как конфигурирование, а не программирование. Однако, считается хорошей практикой проектировать программы структурным методом, в том числе организация ПК в виде модульной структуры, которая подразделяется (насколько это возможно) на части, относящиеся к обеспечению безопасности, которые обеспечивают защиту от дефектов и ошибок при введении новых данных; использовании уже апробированных модулей, и выполнении конструкций, которые облегчают будущие модификации ПК. В описании архитектуры дальнейшие уточнения конструкции каждого компонента/подсистемы должны основываться на подразделении программ на модули. Должны быть оговорены конструкция каждого модуля и испытания, относящиеся к модулю ПК. Должны быть установлены соответствующие компоновочные испытания системы для доказательства того, что комплекс программ отвечает заданным требованиям по безопасности.

4.6. Требования к кодированию. Исходный код должен быть читаемым, понимаемым и поддающимся проверке; удовлетворять заданным требованиям для конструкции модулей ПК; удовлетворять заданным требованиям стандартов кодирования; удовлетворять всем требованиям, заданным при планировании безопасности.

4.7. Требования к испытаниям модулей программного продукта. Каждый модуль должен быть испытан, как это оговорено при проектировании конструкции. Эти испытания должны показать, что каждый модуль выполняет предназначенную ему функцию и не выполняет не предназначенных функций. По результатам испытаний модуля ПП должна быть составлена документация. Должна быть оговорена процедура корректировки в случае неудачного прохождения испытаний.

4.8. Требования к компоновочным испытаниям программного продукта. Комплекс программ должен быть протестирован в соответствии со спецификацией квалификационных испытаний. Эти испытания должны подтвердить, что все модули и компоненты/подсистемы ПП взаимодействуют правильно и выполняют предназначенные им функции и не выполняют не предназначенных функций. По результатам компоновочных испытаний должна быть составлена документация, содержащая результаты испытаний, а также заключение, достигнуты ли все цели и выполнены ли все критерии испытаний. Если испытания прошли неудачно, в документации должны быть указаны причины. В процессе компоновки ПК любая модификация или изменение должны быть проанализированы с точки зрения их влияния, при анализе должны быть выявлены все подвергшиеся воздействию модули и произведены необходимые действия по их корректировке, повторной верификации и изменению конструкции.

5. Компоновка программируемой электроники. Целью требований является встраивание комплекса программ в заданную аппаратуру, соединение ПП и аппаратуры в программируемую электронику, обеспечивающую безопасность, их совместимость и удовлетворение требования заданного уровня по безопасности. При компоновочных испытаниях аппаратуры и программного комплекса любая модификация или изменение объединённой системы должны анализироваться с точки зрения взаимодействия, при котором должны быть выявлены подвергающиеся воздействию модули ПК и проведена необходимая деятельность по повторной верификации. По компоновочным испытаниям должна быть составлена документация, содержащая результаты испытаний и заключение о том, достигнуты ли цели и критерии испытаний. Если испытания прошли неудачно, в документации должны быть указаны причины этого.

6. Процедуры эксплуатации и обслуживания программного продукта. Установление информации и относящихся к ПП процедур, необходимых для подтверждения того, что функциональная безопасность системы сохраняется в процессе эксплуатации и модификации. В данном стандарте программный продукт (в отличие от аппаратуры) невозможно оперативно обслуживать – он всегда только модифицируется.

7. Аттестация программного продукта. Целью требований является обеспечение соответствия объединённой системы, заданным требованиям по безопасности ПП при заданном уровне безопасности внешней среды и системы. Аттестация должна производиться, как это оговорено при планировании аттестации программного продукта (см. гл. 2). Для каждой функции безопасности в документации по аттестации должны быть указаны следующие результаты: использованный вариант плана аттестации; функция, проходившая аттестацию (путём анализа, экспертизы или экспериментальных испытаний) вместе со ссылками на план аттестации ПП; инструментальные средства и оборудование, и данные об их калибровке; результаты аттестации; расхождение между ожидаемыми и полученными результатами. Если обнаружены расхождения между ожидаемыми и полученными результатами, следует принять

решение о том, продолжать ли аттестацию или сделать заявку на изменения и вернуться к более ранней части разработки цикла обеспечения безопасности, для этого составляется документация, являющаяся частью результатов аттестации программного продукта. Основным методом аттестации ПП должны быть экспериментальные испытания; синтез динамических изображений и моделирование могут использоваться дополнительно; программный продукт должен быть проверен тестированием при имитации:

- входных сигналов, существующих при нормальной эксплуатации;
- ожидаемых происшествий и аномалий;
- нежелательных ситуаций, требующих вмешательства системы безопасности.

Всё, оборудование, используемое для аттестации, должно быть квалифицировано в соответствии со спецификацией, согласующейся со стандартом или общепринятой процедурой. К результатам аттестации ПП предъявляются следующие требования: испытания должны показать, что все заданные требования по безопасности программного средства выполняются правильно, и что ПП не выполняет не предназначенных ему функций. По каждому испытанию и его результатам должна быть составлена документация для проведения последующего анализа и независимой оценки соответствия с требуемым уровнем безопасности.

8. Модификация программного комплекса. Целью является произведение корректировок, улучшение или приспособление аттестуемого ПК для гарантии поддержания требуемого уровня соответствия комплексу требований по безопасности. Модификация может производиться только при наличии санкционированной заявки на изменения в соответствии с процедурами, установленными при планировании совершенствования безопасности. Все модификации, оказывающие влияние на функциональную безопасность системы, должны вызывать возврат к соответствующему этапу обеспечения безопасности ПП. Все последующие этапы должны быть повторены в соответствии с процедурами, установленными для конкретных этапов в соответствии с требованиями стандарта. Планирование безопасности при модификации ПК, относящегося к обеспечению безопасности, должно включать следующую информацию: определение персонала и спецификацию его требуемой компетентности; детальную спецификацию модификации; планирование верификации; объём повторной аттестации и испытаний модификации в пределах, требуемых уровнем соответствия комплексу требований по безопасности. Все подробности модификации должны быть изложены в документации. Они могут использоваться на этапах компоновки программируемой электроники, общей установки и ввода в эксплуатацию.

9. Верификация программного комплекса. Целью является, испытание и оценка выходных данные этапа ЖЦ обеспечения безопасности ПП для подтверждения правильности и согласованности с выходными данными и стандартами, которые использовались в качестве входных данных на этом этапе.

Основные аспекты верификации являются общими для нескольких этапов цикла обеспечения безопасности. Верификация ПК должна планироваться одновременно с разработкой для каждого этапа обеспечения безопасности и эта информация должна быть внесена в документацию. При планировании верификации ПК должны быть указаны критерии, методы и инструментальные средства, которые должны использоваться во время верификации. При верификации должны выполняться следующие действия:

- верификация требований по безопасности программного комплекса;
- верификация архитектуры программного комплекса, конструкции системы и встроенного программного комплекса;
- верификация конструкции модулей ПК, объектного кода и данных;
- испытания модулей программного комплекса;
- компоновочные испытания программного комплекса;
- компоновочные испытания комплекса программируемой электроники;
- тестирование реализации требований по безопасности программного продукта и системы.

Оценка функциональной безопасности программного продукта. Выбор методов оценки не гарантирует сам по себе, что будет достигнуто соответствие комплексу требований по безопасности. Производящие оценку специалисты должны учитывать:

- обоснованность и согласованность выбранных методов, языков и инструментальных средств со всем циклом разработки;
- используют ли разработчики методы, которые они полностью понимают;
- хорошо ли методы, языки и инструментальные средства подходят для исключения дефектов и ошибок, возникающих при разработке.

Для каждого метода в таблицах стандарта приводятся рекомендации по уровням соответствия комплексу требований по безопасности. Подходящие методы должны быть выбраны в соответствии с уровнем соответствия комплексу требований по безопасности. Оценка методов связана с понятием ***эффективность***. При всех прочих равных факторах методы, обозначенные HR, являются более эффективными с точки зрения предотвращения систематических отказовых ситуаций при разработке программного комплекса или (в случае архитектуры), более эффективными с точки зрения контроля за остаточными дефектами в ПК, выявляемыми в процессе исполнения. Для конкретных областей применения соответствующая комбинация методов или мер должна быть указана при планировании безопасности.

Функциональные шаги применения IEC 61508-3 начинаются с получения требований к системе безопасности и соответствующей части планов безопасности от заказчика, где должны быть:

1) Оговорены требуемые функции безопасности и относящиеся к ним уровни соответствия комплексу требований по безопасности объекта или системы:

адресованы функции безопасности, предназначенным для них компонентам системы безопасности; адресованы функции программам в каждом компоненте системы безопасности.

2) Определена архитектура программного комплекса для всех функций безопасности, адресованных ПП.
3) Совместно с поставщиком/разработчиком рассмотрена архитектура и безопасное переменное рассмотрение функций программного продукта и аппаратуры.
4) Начато планирование верификации и аттестации ПП.
5) Спроектировано, разработан, проверен и протестирован ПК в соответствии с планированием безопасности; уровнем соответствия ПП комплексу требований по безопасности объекта или системы; циклом обеспечения безопасности ПП.
6) Завершена деятельность по верификации ПП, скомпонован проверенный программный комплекс в заданную аппаратуру, параллельно разработаны относящиеся к ПП описания процедур для пользователей и обслуживающего персонала, которые они должны выполнять при эксплуатации системы.
7) Совместно с разработчиком аппаратуры аттестовано программный продукт в скомпонованных объектах или системах безопасности.
8) Переданы результаты аттестации ПП системным инженерам для дальнейшей компоновки в общую систему.
9) Если во время эксплуатации потребуется модификация ПК, повторены некоторые этапы.

Для выполнения этих шагов должны выбираться методы и меры по безопасности ПП, соответствующие требуемому комплексу требований по безопасности системы. Для облегчения этого выбора в стандарте разработаны таблицы, квалифицирующие различные методы и меры в зависимости от четырёх уровней соответствия комплексу требований по безопасности. Перекрёстные ссылки в таблицах являются рассмотрением каждого метода или меры в связи со ссылками на дальнейшие источники информации. Примеры применения таблиц соответствия комплексу требований по безопасности даны в приложении, а в стандарт **IEC 61508-7** включён вероятностный подход к определению соответствия комплексу требований по безопасности для разработанного программного продукта.

3.2. Особенности процессов обеспечения функциональной безопасности программных продуктов в стандарте ISO 15408:1999

Современным стандартом ***– Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий, – состоящим из трех частей*** является **ISO 15408 :1-3:1999** [22]. Первая часть определяет концепцию всего стандарта, а вторая самая большая часть формализует

методы и требования к ***информационной безопасности***. Его третья часть полностью посвящена процессам обеспечения ***доверия – (качества) компонентов*** информационных систем (ИС), реализующих функции их безопасности. По существу рассматривается регламентирование технологии и процессов обеспечения жизненного цикла программных средств, создаваемых для обеспечения безопасности функционирования и применения систем. При этом акцент документа сосредоточен на ***информационной безопасности*** сложных программных продуктов ИС. Однако основные положения этой части стандарта практически полностью применимы к технологии и процессам создания ПП и обеспечению ***функциональной безопасности***, в том числе для встроенных систем реального времени. Поэтому ниже в данном разделе многие положения этой части стандарта трактуются с позиции обеспечения функциональной безопасности, а ***термин – доверие*** применяется как понятие ***качество или уверенность выполнения требования безопасности***.

Основная концепция **ISO 15408-3** – обеспечение доверия, основанное на оценке качества (активном исследовании реализации функций) продукта или системы [22]. Нарушения безопасности ПП возникают вследствие преднамеренного использования или случайной ***активизации уязвимостей*** при применении систем и ПП по назначению. Рекомендуется ряд шагов для предотвращения уязвимостей, возникающих в продуктах и системах. ***Уязвимости могут возникать вследствие недостатков***:

- требований – продукт или система могут обладать требуемыми от них функциями и свойствами, но все же содержать уязвимости, которые делают их непригодными или неэффективными в части безопасности применения;
- проектирования – продукт или система не отвечают спецификации, и/или уязвимости, являются следствием некачественных стандартов проектирования или неправильных проектных решений;
- эксплуатации – продукт или система разработаны в полном соответствии с корректными спецификациями, но уязвимости возникают как результат неадекватного управления пользователями при эксплуатации.

Оценка и утверждение ***целей функциональной безопасности*** требуется для демонстрации заказчику или пользователю, что установленные цели проекта адекватны проблеме его безопасности. Существуют цели и функции безопасности для объекта или ПП и цели безопасности для среды. Рекомендуется сопоставлять эти цели безопасности с идентифицированными угрозами, которым они противостоят, и/или с политикой и предположениями, которым они должны соответствовать. Не все цели безопасности могут быть реализованы соответствующим объектом, так как некоторые могут зависеть от требований безопасности системы, выполняемых ее средой. В этом случае требования безопасности, относящиеся к внешней среде, необходимо ясно изложить и оценить в контексте требований к системе.

Использование требований стандарта означает, что требования могут быть четко идентифицированы, что они автономны, и применение каждого требования возможно и даст значимый результат при оценке качества, основанный на анализе соответствия объекта этому конкретному требованию. Отношение между функциями безопасности системы и функциональными требованиями может быть отношением типа «многие ко многим». Тем не менее, каждая функция безопасности должна способствовать удовлетворению, по меньшей мере, одного требования безопасности. Функции безопасности, которые не соответствуют этому положению, обычно необязательны.

Доверие – основа для уверенности в том, что продукт или система отвечают целям и требованиям функциональной безопасности. Активное исследование доверия – это оценка процесса функционирования системы для определения его свойств безопасности. Методы оценки могут, в частности, включать в себя:

- анализ и проверку выполнения процессов и процедур;
- проверку, что процессы и процедуры действительно применяются;
- анализ соответствия реализации каждого положения проекта требованиям;
- верификацию доказательств правильности реализации функций;
- анализ руководств применения;
- анализ разработанных функциональных тестов и полученных результатов;
- независимое функциональное тестирование ПП и системы;
- анализ уязвимостей, включающий предположения о дефектах и ошибках.

В стандарте применяется иерархия детализации требований к безопасности и качеству систем и ПК с использованием специфических терминов: ***класс*** – наиболее общая характеристика качества, которая структурируется ***семейством*** – субхарактеристиками. Каждое семейство может иметь несколько ***компонентов*** – атрибутов, состоящих из ***элементов*** качества и объяснение их содержания и разграничения.

Семейство доверия (качества) в стандарте может содержать один или несколько компонентов доверия. Этот подраздел семейства доверия включает описание имеющихся компонентов. Подраздел идентификации компонента содержит описательную информацию, необходимую для категорирования, регистрации и ссылок на компонент. Каждый элемент представляет собой требование для выполнения. Формулировки этих требований к ПП должны быть четкими, краткими и однозначными. Поэтому каждое требование рекомендуется излагать как отдельный элемент (рис. 3.2).

Управление конфигурацией программных комплексов
Автоматизация управления конфигурацией
Возможности управления конфигурацией
Область управления конфигурацией

Поставка и эксплуатация программных продуктов
Поставка программного продукта
Установка, генерация и запуск ПП

Разработка программного продукта
Функциональная спецификация ПП
Проект верхнего уровня ПП
Представление реализации ПП
Внутренняя структура функциональной безопасности объекта
Проект нижнего уровня ПК

Руководства по безопасности программного продукта
Руководство администратора
Руководство пользователя

Поддержка жизненного цикла безопасности программного продукта
Безопасность разработки ПП
Устранение дефектов
Определение жизненного цикла ПП
Инструментальные средства и методы

Тестирование программного продукта
Покрытие тестами ПП
Глубина тестирования ПП
Функциональное тестирование ПП
Независимое тестирование ПП

Оценка уязвимостей программного продукта
Неправильное применение ПП
Стойкость функций безопасности объекта
Анализ уязвимостей объекта

Рис.3.2.

Класс управление конфигурацией (УК) обеспечивает сохранение целостности объектов, устанавливая и контролируя определенный порядок процессов корректировки, модификации и предоставления связанной с ними информации. УК предотвращает несанкционированную модификацию, добавление или уничтожение составляющих объектов, обеспечивая тем самым качество и документацию компонентов, которые подготовлены к распространению. Управления конфигурацией устанавливает уровень автоматизации, используемый для изменения элементов конфигурации. Возможности управления определяют характеристики системы УК. Область управления указывает на те элементы объектов, для которых необходим контроль со стороны системы управления конфигурацией.

Класс поставка и эксплуатация определяет требования к мерам, процедурам и стандартам, применяемым для безопасной поставки, установки и эксплуатации ПП, обеспечивая, чтобы безопасность объектов не нарушалась во время его распространения, внедрения и эксплуатации. Поставка распространяется на процедуры, используемые для поддержки безопасности во время передачи объекта пользователю при первоначальной поставке и последующих модификациях. Она включает в себя специальные процедуры, необходимые для демонстрации подлинности поставленного объекта. Такие процедуры и меры – основа обеспечения безопасности во время и после передачи ПП для применения. Установка, генерация и запуск предусматривает, чтобы копия объекта была конфигурирована и активизирована администратором так, чтобы показать те же самые свойства безопасности, что и у оригинала. Эти процедуры обеспечивают уверенность в том, что администратор будет осведомлен о параметрах конфигурации объекта и о том, как они способны повлиять на функциональную безопасность.

Класс разработка определяет требования для пошагового уточнения функциональной безопасности, начиная с краткой спецификации объекта в задании на безопасность (ЗБ) и вплоть до фактической реализации (см. рис. 3.2). Каждое из получаемых представлений содержит информацию, помогающую оценщику решить, были ли выполнены функциональные требования к безопасности. Функциональная спецификация описывает функции безопасности, и необходимо, чтобы она была полным и точным отображением требований безопасности объекта. Функциональная спецификация также детализирует его внешний интерфейс. Предполагается, что пользователи и заказчики объекта взаимодействуют с функциональной безопасностью через этот интерфейс.

Проект верхнего уровня – проектная спецификация самого высокого уровня, которая уточняет функциональную спецификацию безопасности системы в основных составляющих частях. Она идентифицирует базовую структуру функциональной безопасности, а также основные элементы аппаратных, программных и программно-аппаратных средств. Представление реализации – наименее абстрактное отражение функциональной безопасности. Оно фиксирует ее детализированное внутреннее содержание на уровне исходного текста,

аппаратных схем и т.д. Требования к внутренней структуре определяют необходимое структурирование функций безопасности.

Проект нижнего уровня – детализированная проектная спецификация, уточняющая проект верхнего уровня до необходимой детализации, которая может быть использована как основа для программирования и/или проектирования аппаратуры и программных компонентов. Соответствие представлений – демонстрация отображения между всеми смежными парами имеющихся представлений функций безопасности, от краткой спецификации объекта до наименее абстрактного из имеющихся представлений.

Модели политики безопасности – структурные представления методов, используемые для обеспечения повышенного доверия, что функциональная спецификация соответствует принятой политике безопасности и, в конечном счете, функциональным требованиям безопасности системы. Это достигается посредством определения соответствия между функциональной спецификацией, моделью политики безопасности и моделируемыми методами обеспечения безопасности ПП.

Класс руководства по безопасности определяет требования, направленные на обеспечение понятности, достаточности и законченности эксплуатационной документации, представляемой разработчиком. Эта документация, которая содержит две категории информации (для пользователей и администраторов), является важным фактором безопасной эксплуатации объекта и ПП.

Требования к руководству администратора должны обеспечивать отражения ограничений среды, которые будут поняты администраторами и операторами. Руководство администратора – основной документ, имеющийся в распоряжении разработчика, для предоставления администраторам объекта, детальной и точной информации о том, как осуществлять администрирование безопасным способом и эффективно использовать доступные процедуры обеспечения безопасности.

Требования к руководству пользователя должны обеспечивать возможность эксплуатировать объект безопасным способом. Руководство – основной документ, имеющийся в распоряжении разработчика, для предоставления пользователям необходимой общей и специфической информации о том, как правильно использовать функции безопасности. В руководстве необходимо осветить два аспекта. Во-первых, требуется объяснить, что делают доступные пользователю процедуры безопасности, и как они будут использоваться, чтобы пользователи имели возможность последовательно и действенно защищать свою систему. Во-вторых, требуется разъяснить роль пользователя в поддержании безопасности системы и ПП.

Класс поддержка жизненного цикла определяет требования для реализации всех этапов разработки четко определенной модели, включая политики и процедуры устранения недостатков и дефектов, правильное использование инструментальных средств и методов, а также меры безопасности для защиты среды разработки.

Безопасность разработки охватывает физические, процедурные, относящиеся к персоналу и другие меры безопасности, используемые применительно к среде разработки. Она также содержит требования к физической безопасности местоположения разработки и к контролю за отбором и наймом персонала разработчиков. Устранение дефектов обеспечивает, чтобы недостатки, обнаруженные потребителями, отслеживались и исправлялись, пока объект сопровождается разработчиком. Несмотря на то, что при оценке объекта не может быть принято решение о потенциальном соответствии требованиям устранения недостатков, можно оценить политики и процедуры, которые разработчик предусмотрел для выявления и устранения дефектов и распространения исправлений потребителям.

Определение жизненного цикла ПК устанавливает, что технология разработки, используемая разработчиком для создания объекта, включает в себя положения и действия, указанные в требованиях к процессу разработки и поддержке эксплуатации. Уверенность в соответствии объекта требованиям больше, когда анализ безопасности и подготовка свидетельств осуществляются на регулярной основе, как неотъемлемая часть процесса разработки и поддержки эксплуатации всей системы и ПП. Инструментальные средства и методы связаны с необходимостью определения средств разработки, используемых для анализа и создания объекта и ПП. Сюда включены требования, относящиеся к инструментальным средствам разработки и опциям этих инструментальных средств, зависящим от их реализации.

Класс тестирование устанавливает требования, которые должны демонстрировать, что реализованные функции удовлетворяют функциональным требованиям безопасности системы. Покрытие определяет полноту функциональных тестов, выполненных разработчиком для анализа качества системы и ПК. Оно связано со степенью тестирования функций безопасности. Глубина тестирования характеризуется уровнем детализации, на котором разработчик проверяет программы. Тестирование функций безопасности основано на последовательно увеличивающейся глубине информации, получаемой из анализа представлений безопасности.

Функциональное тестирование ПК должно устанавливать, что функции безопасности действительно демонстрируют свойства, необходимые для удовлетворения требований спецификации. Функциональное тестирование обеспечивает доверие, что функции удовлетворяют, по меньшей мере, требованиям выбранных функциональных компонентов. Эти процедуры сосредоточены на функциональном тестировании, выполняемом разработчиком. Независимое тестирование определяет степень выполнения функционального тестирования третьей стороной, кроме разработчика и заказчика. Эти процедуры повышают ценность тестирования добавлением тестов, которые расширяют тесты разработчика.

Класс оценка уязвимости ПП определяет требования, направленные на идентификацию уязвимостей, которые могут проявиться и быть активизированы. Особое внимание должно быть уделено уязвимостям, которые вносятся при

проектировании, эксплуатации, неправильном применении или неверной конфигурации объекта. Анализ скрытых каналов направлен на выявление и анализ непредусмотренных коммуникационных каналов, которые могут применяться при нарушениях предписанных функций безопасности. Анализ неправильного применения позволяет выяснить, способен ли администратор или пользователь, используя руководства, определить, что система или ПП конфигурированы или эксплуатируются небезопасным способом.

Анализ стойкости направлен на определение функций безопасности, которые реализованы с помощью вероятностного или перестановочного механизма. Даже если такие функции нельзя обойти, отключить или исказить, не исключено, что их все же можно преодолеть прямой атакой. Может быть заявлен уровень или специальная метрика стойкости для каждой из этих функций. Анализ стойкости функций выполняют для принятия решения, отвечают ли такие функции сделанным заявлениям. Анализ уязвимостей заключается в идентификации недостатков, которые могли быть внесены на различных этапах разработки. Эти потенциальные уязвимости оцениваются посредством тестирования проникновения, позволяющим сделать заключение, могут ли они в действительности быть использованы для нарушения безопасности системы и ПП.

Для систематичного применения приведенных на рис. 3.2 классов и семейств требований в стандарте ведено понятие ***профиль защиты (ПЗ)*** – независимая от реализации совокупность требований безопасности для некоторой категории объектов, отвечающая специфическим требованиям проекта и потребителя. Цель разработки и оценки ПЗ – показать, что он является полным, непротиворечивым, технически правильным и поэтому пригоден для изложения конкретных требований к одному или нескольким типам объектов. Оцененный ПЗ пригоден в качестве основы для разработки задания по безопасности. Для принятия решения о достаточности требований безопасности в составе ПЗ важно, чтобы решаемая задача безопасности ясно понималась всеми участниками оценки.

Задание по безопасности (ЗБ) – совокупность требований и спецификаций, предназначенная для использования в качестве основы для оценки конкретного объекта. Цель оценки ЗБ – показать, что оно является полным, непротиворечивым, технически правильным и поэтому пригодно для использования в качестве основы при оценке уровня безопасности соответствующей системы. Если, после тщательного рассмотрения, окажется, что ни один из компонентов требований настоящего стандарта не применим непосредственно ко всем или к части требований безопасности системы, разработчик ЗБ может сформулировать другие требования, которые не ссылаются на этот стандарт. Использование таких требований должно быть строго обосновано.

Не все перечисленные выше классы и семейства процессов обеспечения функциональной безопасности систем и ПП целесообразно применять в каждом проекте. В зависимости от сложности и критичности требования к безопасности функционирования системы и доступных ресурсов для ее реализации,

стандартом рекомендуется выбирать набор классов и семейств процессов, достаточных для обеспечения необходимого качества комплекса функциональной безопасности проекта – ***оценочный уровень доверия***. Оценочные уровни доверия (ОУД) образуют возрастающую шкалу достигаемого качества функциональной безопасности, которая позволяет соотнести получаемый уровень качества с трудоемкостью его реализации и возможностью достижения этой степени доверия.

Не все классы и семейства настоящего стандарта включены в каждый из перечисленных ниже оценочных уровней доверия. Эти семейства рекомендуется использовать для повышения уровней доверия в тех ПЗ и ЗБ, для которых они действительно полезны и необходимы. Определены семь иерархически упорядоченных оценочных уровней доверия для ранжирования при выборе доверия к безопасности объектов оценки. Каждый последующий ОУД представляет более высокое доверие (гарантированное качество), чем любой из предыдущих. Связь уровней доверия с классами и семействами технологических процессов обеспечения ЖЦ безопасности систем и программных средств в стандарте иллюстрированы семью таблицами. В каждой из них выделены и отмечены обязательные (белые) и рекомендуемые (остальные) классы и семейства процессов, обеспечивающие определенный уровень доверия при создании и применении методов и средств соответствующей безопасности. Эти таблицы помогают выбирать технологии в соответствии с требованиями к безопасности системы и ПП с учетом доступных ресурсов.

Оценочный уровень доверия 1 (ОУД1) – предусматривает функциональное тестирование и применим, когда требуется некоторая уверенность в правильном функционировании системы, а угрозы безопасности (и надежности) не рассматривают как серьезные. Он полезен там, где требуется, чтобы было уделено должное внимание безопасности, путем независимого тестирования на соответствие спецификации и экспертизы представленной документации. Предполагается, что оценка может успешно проводиться без помощи разработчика и с минимальными затратами, посредством анализа экспертами заданных функций безопасности с использованием функциональной спецификации, спецификации интерфейсов и руководств.

Оценочный уровень доверия 2 (ОУД2) – включает структурное тестирование, содержит требование сотрудничества с разработчиком для получения информации о проекте и результатах тестирования. Он применим в тех случаях, когда разработчикам или пользователям требуется, подтверждаемый уровень доверия (от невысокого до умеренного). Такая ситуация может возникать при обеспечении безопасности разработанных ранее (наследуемых) систем или при ограниченной доступности к ним разработчика. ОУД2 обеспечивает доверие посредством анализа применяемых функций безопасности с использованием функциональной спецификации, спецификации интерфейсов, руководств и проекта верхнего уровня. Этот уровень требует тестирования и анализа уязвимостей разработчиком, основанного на более детализированных спецификациях.

Оценочный уровень доверия 3 (ОУД3 – предусматривает методическое тестирование и проверку, которая позволяет разработчику достичь доверия, путем проектирования безопасности без значительного изменения существующей технологии качественной разработки всей системы. ОУД3 применим в тех случаях, когда разработчикам или пользователям требуется независимо подтверждаемый умеренный уровень доверия на основе исследования системы и процесса ее разработки без существенных затрат на изменение технологии. Этот уровень представляет значимое увеличение доверия, требуя более полного покрытия тестированием функций и процедур безопасности.

Оценочный уровень доверия 4 (ОУД4) – предусматривает методическое проектирование, тестирование и углубленную проверку, что позволяет разработчику достичь максимального качества, основанного на регламентированной технологии разработки, которая не требует глубоких специальных знаний, навыков и других ресурсов. ОУД4 – самый высокий уровень, на который, экономически целесообразно ориентироваться при оценке уже существующих продуктов. Анализ поддержан независимым тестированием, свидетельством разработчика об испытаниях, основанных на функциональной спецификации и проекте верхнего уровня, подтверждением результатов тестирования разработчиком и независимым анализом уязвимостей. Уровень также обеспечивает доверие посредством использования контроля среды разработки и дополнительного управления конфигурацией системы и ПП.

Оценочный уровень доверия 5 (ОУД5) **–** позволяет разработчику достичь максимального качества путем систематического проектирования функциональной безопасности, основанного на строгой технологии разработки, поддержанной умеренным применением узко специализированных методов, не влекущих излишних затрат на методы проектирования безопасности. Доверие достигается применением формальной модели политики безопасности и полуформального представления функциональной спецификации и проекта верхнего уровня системы, а также полуформальной демонстрации соответствия между ними. Кроме этого, требуется модульное проектирование системы. Анализ поддержан независимым свидетельством разработчика об испытаниях, основанных на функциональной спецификации, проектах верхнего и нижнего уровня, независимым подтверждением результатов тестирования разработчиком и независимым анализом уязвимостей. ОУД5 также обеспечивает качество посредством использования контроля среды разработки и управления конфигурацией системы, требуя соблюдать структурированную архитектуру системы.

Оценочный уровень доверия 6 (ОУД6) – позволяет разработчикам достичь высокой безопасности путем полуформальной верификации всего проекта и тестирования, применением специальных методов проектирования безопасности в строго контролируемой среде разработки с целью получения высокой безопасности системы и защиты активов от значительных рисков, где ценность защищаемых активов оправдывает дополнительные затраты. ОУД6 также обеспечивает повышение доверия посредством использования структуриро-

ванного процесса разработки, контроля среды разработки и управления конфигурацией системы, включая полную автоматизацию**,** и свидетельства безопасности процедур поставки. Этот уровень представляет значительное увеличение доверия по сравнению с предыдущим, требует всестороннего анализа, структурированное представление реализации, более стройную структуру системы, всесторонний независимый анализ уязвимостей, систематическую идентификацию скрытых каналов, улучшенное управление конфигурацией и глубокий контроль среды разработки.

Оценочный уровень доверия 7 (ОУД7 – применим при разработке безопасных систем для использования в ситуациях чрезвычайно высокого риска и/или там, где высокая ценность активов или систем оправдывает максимальные затраты на их безопасность. Практическое применение уровня ограничено системами, которые строго ориентированы на реализацию полных функциональных возможностей безопасности, для которых возможен и целесообразен подробный формальный анализ. Уровень обеспечивает качество посредством использования всех представленных выше классов и семейств, а также процессов предшествующих уровней, структурированного процесса разработки, средств контроля среды разработки и всестороннего управления конфигурацией системы, включая полную автоматизацию, и свидетельства безопасных процедур поставки (см. рис. 3.2). Этот уровень представляет значительное увеличение доверия, требует всестороннего анализа, использующего формальные представления и формальное соответствие, а также всестороннее независимое тестирование.

3.3. Особенности методологии обеспечения безопасности программных продуктов в стандарте ISO 13335: 1-5: 1998

Широкий комплекс методологических задач, которые необходимо решать при проектировании безопасности систем отражает стандарт **– ISO 13335: 1-5: 1996-1998 –** Руководство по управлению безопасностью. В его пяти частях **–** 1. Концепция и модели обеспечения безопасности информационных технологий; 2. Планирование и управление безопасностью (информационных) технологий; 3. Техника управления безопасностью ИТ; 4. Селекция (выбор) средств обеспечения безопасности; 5. Безопасность внешних связей **–** внимание сосредоточено на основных принципах и методах проектирования, равнопрочных систем обеспечения безопасности (информационных) систем от угроз различных видов. Это руководство достаточно полно систематизирует основные методы и процессы подготовки проекта защиты для последующей разработки обеспечения безопасности функционирования систем. Изложение базируется на понятии риска от угроз любых негативных воздействий на ИС. В первой части стандарта представлены функции средств защиты и необходимые действия по их реализации, модели уязвимости и взаимодействие средств защиты.

При проектировании систем рекомендуется учитывать: необходимые функции защиты; угрозы; уязвимости; негативные воздействия; защитные меры; ресурсы (аппаратные, информационные, программные, специалистов) и их ограниченность. В остальных частях стандарта предложена и развивается концепция и модель управления и планирования построения системы обеспечения безопасности.

В стандарте выделены функциональные компоненты и средства обеспечения безопасности, а также их взаимодействие. ***Процессы управления безопасностью должны включать***:

- управление изменениями и конфигурацией функций и компонентов системы безопасности;
- анализ и управление рисками;
- прослеживаемость функций и результатов комплексов средств обеспечения безопасности;
- регистрацию, обработку и мониторинг инцидентов.

Приводятся общие требования к оценке результатов обеспечения безопасности, а также возможные варианты организации специалистов для комплексного обеспечения безопасности. Систематизирована политика и техника планирования, выбора, построения и использования средств обеспечения безопасности для ограничения допустимых рисков при различных схемах взаимодействия и средствах защиты. Рекомендуются различные подходы и стратегии при создании систем защиты и поддержке их последующего развития. Содержание частей стандарта детализирует общие концепции и достаточно точно определяется их названиями. Изложенную в стандарте модель планирования обеспечения безопасности, целесообразно конкретизировать и использовать как ***фрагмент проекта функциональной безо-пасности ПП***.

3.4.Особенности процессов жизненного цикла программных комплексов в стандарте ISO 12207:2008

Основу ***стандартизации в программной инженерии*** многие годы составлял базовый стандарт **ISO/IEC 12207:1995 –** Процессы жизненного цикла программных средств. Современный вариант стандарта **– ISO/IEC 12207:2008** разработан ***для повышения культуры, безопасности и качества производства организациями проектирующими, разрабатывающими, приобретающими системы и сложные программные продукты.*** Процессы в стандарте составляют ***полную совокупность*** функций при ***проектировании и производстве заказных программных продуктов***. ***Модель жизненного цикла*** представляется в виде последовательности этапов – процессов, которые могут перекрываться и (или) повторятся циклически в соответствии с областью применения, размером, сложностью, потребностью в изменениях. Стандарт требует, чтобы в каждом проекте определялась подходя-

щая модель жизненного цикла. Предпочтительно чтобы это была та модель, которая уже определялась для применения в ранее выполненных проектах. Разделы ***процессов комплексов программ*** включает две группы из 18 процессов: реализации и поддержки разработки комплексов – рис. В1. По названиям они в значительной степени подобны процессам в стандарте **ISO 12207:1995**, однако различаются полнотой и конкретностью содержания.

Задачи стандарта выражаются в форме требования, рекомендации или допустимых действий, предназначенных для поддержки достижения выходов процесса. ***Атрибуты*** характеризуют специфику каждого процесса, предварительно определенную цель процесса и его результаты, которые достигаются посредством выполнения определенных действий. Каждый процесс стандарта должен удовлетворять описанным критериям.

Стандарт **ISO/IEC 12207:2008** устанавливает процессы ***жизненного цикла для заказных программных комплексов***, включая процессы и действия, применяемые во время сбора и конфигурации системы. ***Основная цель –*** представление общей структуры стандарта с такой позиции, чтобы покупатели, поставщики, разработчики, специалисты по обслуживанию, операторы, менеджеры и технический персонал, связанный с разработкой программного продукта, ***использовали общий язык.*** и результаты десятилетнего опыта разработки и применения стандартов **ISO/IEC 12207** и **ISO/IEC 15288** (см. рис. В1 в Введении).

Структура стандарта имеет гибкий, модульный формат, что позволяет адаптироваться к потребностям детализации, которая необходима отдельному пользователю. Процессы делятся на три базовых типа:

- ***основные;***
- ***вспомогательные;***
- ***организационные.***

Группа основных процессов жизненного цикла включает в себя базовые процессы, участвующие в создании и применении программного продукта. Выделяются пять основных групп: заказ; поставка; разработка; эксплуатация; сопровождение.

Процессы заказа определяют работы заказчика, то есть организации, которая приобретает систему, программный продукт или программную услугу. Каждый этап подразумевает, что предыдущий завершен.

Подготовка:

- Собрана информация, объясняющая необходимость разработки или модифицирования продукта;
- Составлен и утвержден список системных требований;
- Определены глобальные требования к программному продукту;
- Рассмотрены варианты приобретения готового программного продукта, покупки и модернизации, создания "с нуля";

- Проанализированы технические требования;
- Подготовлен, документально оформлен и выполнен план заказа, который содержит:
  - требования к системе;
  - планируемую загрузку системы;
  - тип реализуемого договора;
  - обязанности организаций, участвующих в договоре;
  - обеспечение методов реализации договора;
  - анализ возможных рискованных ситуаций, а также методы управления такими ситуациями.

Определены и документально оформлены принятые правила и условия (критерии) реализации договора.

Подготовка заявки на подряд: Документально оформлены требования к заказу, состав которых зависит от вариантов его реализации. Соответствующая документация по заказу должна содержать:

- требования к системе;
- описание области применения системы;
- указания для участников торгов;
- список программных продуктов;
- сроки и условия реализации заказа;
- правила контроля над субподрядчиками;
- технические ограничения (например, по условиям эксплуатации).

- Определены, какие из процессов, работ и задач, описанных в настоящем стандарте, применимы к условиям проекта, и соответствующим образом адаптированы.
- Определены контрольные пункты договора, при выполнении которых анализируется и проверяется деятельность поставщика.
- Требования к заказу представлены организации, выбранной для выполнения работ в процессе заказа.

Подготовка договора:

- Определена процедура выбора поставщика, включающая критерии оценки поступающих предложений по реализации заказа и их соответствие установленным требованиям.
- Выбран поставщик, исходя из оценки предложений, поступивших от потенциальных поставщиков, их возможностей и других рассматриваемых факторов.
- В зависимости от того, была ли проведена адаптация настоящего стандарта к условиям проекта, включение в текст договора (или указание на источник) адаптированный настоящий стандарт.

- Подписан обновленный контракт с учетом корректировок, внесенных при обсуждении заказчика и поставщика.

Приемка и закрытие договора:

- Подготовлены контрольные примеры, контрольные данные, процедуры тестирования и условий проведения испытаний. Заказчик должен определить степень участия поставщика при проведении приемки.
- Заказчиком проверена готовность поставщика к проведению приемки и проведению приемочных испытаний поставляемого программного продукта.
- Заказчиком принят от поставщика продукт (при выполнении всех условий приемки).
- После приемки заказчик принимает на себя ответственность за управление конфигурацией поставленного программного продукта.

Разработка программного продукта

Данный процесс описывает все фазы разработки программного продукта (создание, тестирование и приведение к конечному результату, готовому к сдаче заказчику). Выбор метода разработки зависит от конкретной ситуации. Самый частый метод разработки V-модель:

- Подготовка программного комплекса.
- Анализ требований технического задания.
- Проектирование архитектуры программного комплекса.
- Детальное проектирование программного комплекса.
- Конструирование программного комплекса.
- Комплексирование программного комплекса.
- Тестирование.

Подготовка программного комплекса:

- Выбор модели жизненного цикла программного комплекса, соответствующей области реализации, величине и сложности проекта (если это не указано заказчиком в договоре)
- Оформление выходных результатов в соответствии с процессом документирования.
- Оформление возникающих проблем и устранение несоответствий, обнаруженных в программных продуктах и задачах, если таковые проявляются при разработке
- Выбор и адаптация стандартов, методов, инструментария, языков программирования (если они не установлены в договоре), которые будут использоваться для выполнения работ в процессе разработки и во вспомогательных процессах.

- Разработка плана проведения процессов разработки, планы должны охватывать конкретные стандарты, методы, инструментарий, действия и обязанности, связанные с разработкой и квалификацией всех требований, включая безопасность и защиту.

***Анализ требований технического задания*:**

- Анализ области применения разрабатываемой системы с точки зрения определения требований к ней.
- Оформление требований к программному продукту, которые должны описывать:
  - функциональные и технические требования, включая производительность, физические характеристики и окружающие условия среды, под которые должен быть создан программный объект архитектуры (далее - программный объект);
  - требования к внешним интерфейсам программного объекта архитектуры;
  - квалификационные требования;
  - требования безопасности, включая требования, относящиеся к методам эксплуатации и сопровождения, воздействию окружающей среды и безопасности персонала;
  - требования защиты, включая требования, относящиеся к допустимой точности информации;
  - эргономические требования, включая требования, относящиеся к ручным операциям, взаимодействию "человек-машина", персоналу и областям, требующим концентрации внимания человека, связанным с чувствительностью объекта к ошибкам человека и квалификацией персонала;
  - требования к определению данных и базе данных;
  - требования по вводу в действие и приемке поставляемого программного продукта на объекте(ах) эксплуатации и сопровождения;
  - требования к документации пользователя;
  - требования к эксплуатации объекта пользователем;
  - требования к обслуживанию пользователя.
  - Оценка технического задания (ТЗ) с учетом следующих критериев (при этом результаты оценок должны быть документально оформлены):
  - учет потребностей заказчика;
  - соответствие потребностям заказчика;
  - тестируемость;
  - выполнимость проектирования системной архитектуры;
  - возможность эксплуатации и сопровождения.

Проектирование архитектуры программного комплекса:

- Определение общей архитектуры системы (архитектура верхнего уровня). В архитектуре должны быть указаны объекты технических и программных средств и ручных операций. Должно быть обеспечено распределение всех требований к системе между объектами архитектуры.
- Оценка системной архитектуры и требований к объектам архитектуры с учетом следующих критериев:
  - учет требований к системе;
  - соответствие требованиям к системе;
  - соответствие используемых стандартов и методов проектирования;
  - возможность программных объектов архитектуры выполнять установленные для них требования;
  - возможности эксплуатации и сопровождения.

Детальное проектирование программного комплекса:

- Трансформирование требований к программному объекту в архитектуру, которая описывает общую структуру объекта и определяет компоненты программного объекта.
- Разработка и оформление общего (эскизного) проекта внешних интерфейсов программного объекта и интерфейсов между компонентами объекта.
- Разработка и оформление общего проекта базы данных.
- Разработка и оформление предварительной версии документации пользователя.
- Разработка и оформление предварительных общих требований к тестированию программного объекта и графику сборки программного продукта.
- Оценка архитектуры программного объекта и эскизные проекты интерфейсов и базы данных по следующим критериям:
  - учет требований к программному объекту;
  - внешняя согласованность с требованиями к программному объекту;
  - внутренняя согласованность между компонентами программного объекта;
  - соответствие методов проектирования и используемых стандартов;
  - возможность технического проектирования;
  - возможность эксплуатации и сопровождения.

Конструирование программного комплекса:

- Разработка технического проекта для каждого компонента программного объекта.
- Разработка технического проекта внешних интерфейсов программного объекта, интерфейсов между компонентами программного объекта и между программными модулями.
- Разработка технического проекта базы данных.
- Определение требований к испытаниям и программе испытаний программных модулей.
- Оценка технического проекта тестирования по следующим критериям:
  - учет требований к программному объекту;
  - внешнее соответствие спроектированной архитектуре;
  - внутренняя согласованность между компонентами программного объекта и программными модулями;
  - соответствие методов проектирования и используемых стандартов;
  - возможность тестирования;
  - возможность эксплуатации и сопровождения.

Комплексирование программного продукта

- Разработка и документальное оформление следующих продуктов:
  - каждого программного модуля и базы данных;
  - процедуры испытаний (тестирования) и данные для тестирования каждого программного модуля и базы данных.
- Разработка плана сборки для объединения программных модулей и компонентов в программный комплекс. План должен включать требования к испытаниям (тестированию), процедуры тестирования, контрольные данные, обязанности исполнителя и программу испытаний.
- Сбор программных модулей и компонентов.
- Сбор программных объектов в единую систему вместе с объектами технической конфигурации, ручными операциями и, при необходимости, с другими системами.

Тестирование

- Тестирование в соответствии квалификационным требованиям к программному продукту.
- Оценка проекта, запрограммированного программного объекта, тестирование по следующим критериям:
  - тестовое покрытие требований к программному объекту;
  - соответствие ожидаемым результатам;
  - возможность сборки и тестирования продукта и системы (при их проведении);

- возможность эксплуатации и сопровождения.

- Тестирование системы и оценка по следующим критериям:
  - тестовое покрытие требований к программному продукту и системе;
  - соответствие ожидаемым результатам;
  - возможность эксплуатации и сопровождения.
- Проведение аудиторской проверки и доработка.

Эксплуатация

Процесс эксплуатации состоит из работ и задач оператора. Процесс охватывает эксплуатацию программного продукта и поддержку пользователей в процессе эксплуатации. Так как эксплуатация программного продукта входит в эксплуатацию системы, работы и задачи данного процесса связаны с системой.

Сопровождение

Процесс сопровождения состоит из работ и задач, выполняемых персоналом сопровождения. Данный процесс реализуется при изменениях (модификациях) программного продукта и соответствующей документации, вызванных возникшими проблемами (дефектами) или потребностями в модернизации или настройке. Целью процесса является изменение существующего программного продукта при сохранении его целостности. Данный процесс охватывает вопросы переносимости и снятия программного продукта с эксплуатации. Процесс заканчивается снятием программного продукта с эксплуатации.

Вспомогательные процессы жизненного цикла комплексов программ:

- процесс документирования:

подготовка процесса;
  - проектирование и разработка;
  - выпуск;
  - сопровождение.
- процесс управления конфигурацией:
  - подготовка процесса;
  - определение конфигурации;
  - контроль конфигурации;
  - учет состояний конфигурации;
  - оценка конфигурации;
  - управление выпуском и поставка.
- процесс обеспечения качества:
  - подготовка процесса;
  - обеспечение продукта;

  - обеспечение процесса;
  - обеспечение системы качества.
- процесс верификации:
  - подготовка процесса;
  - верификация.
- процесс аттестации:
  - подготовка процесса;
  - аттестация.
- процесс совместного анализа:
  - подготовка процесса;
  - анализы управления проектом;
  - технические анализы.
- процесс аудита:
  - подготовка процесса;
  - аудиторская проверка.
- процесс решения проблем (устранения дефектов):
  - подготовка процесса;
  - решение проблемы.

Организационные процессы жизненного цикла комплексов программ

- процесс управления:
  - определение области управления;
  - планирование;
  - выполнение и контроль;
  - проверка и оценка;
- процесс создания инфраструктуры:
  - подготовка процесса;
  - создание инфраструктуры;
  - сопровождение инфраструктуры.
- процесс усовершенствования:
  - создание процесса;
  - оценка процесса;
  - усовершенствование процесса.
- процесс обучения:
  - подготовка процесса;
  - разработка учебных материалов;
  - реализация плана обучения.

Ответственность за работы и задачи вспомогательного и организационного процессов несет организация, выполняющая эти процессы. Организация гарантирует реальность существования и функциональные особенности конкретного процесса. Данная организация организует и выполняет управление процессами на проектном уровне в соответствии с процессом управления; определяет инфраструктуру для данного процесса в соответствии с процессом создания инфраструктуры; адаптирует процессы к условиям проекта в соответствии с процессом адаптации и управляет процессами на организационном уровне в соответствии с процессами усовершенствования и обучения. В качестве методов обеспечения качества могут быть использованы: совместные анализы, аудиторские проверки, верификация и аттестация.

Стандарт ISO 12207:2008 дополнительно поддерживается группой международных стандартов, детализирующих отдельные этапы и процессы жизненного цикла, которые целесообразно применять для обеспечения надежности, функциональной безопасности и высокого качества сложных программных продуктов.

3.5. Особенности создания программных продуктов в системах безопасности атомных электростанций по стандарту МЭК 60880:2002

Стандарт **МЭК 60880** состоит из двух частей под общим названием – Программное обеспечение компьютеров в системах безопасности атомных электростанций. ***Первая часть утверждена в 1986 году***. ***Вторая часть, утвержденная в 2000 году***, является дополнением и развитием первой части по трем специальным направлениям, которые отражены в ее подзаголовке – Программные аспекты защиты от отказов по общим причинам; использование программных инструментов; применение ранее разработанного программного обеспечения. Основная особенность этой части стандарта состоит в концентрации рекомендаций на методах и процедурах, обеспечивающих и гарантирующих высокое качество и безопасность функционирования программ на всех этапах их жизненного цикла, предотвращающих ошибки и отказы системы. Почти половина объема в обеих частях стандарта уделена Приложениям, в которых конкретизируются и детализируются общие рекомендации.

В ***первой части*** рассмотрены общие вопросы и некоторые фазы жизненного цикла ПК, задачи обеспечения качества, верификация, испытания и документирование комплексов программ. Рекомендуется тщательно разрабатывать требования к ПП и подробно отражать в них: функции, конфигурацию системы, взаимодействие с внешней средой, ограничения характеристик аппаратуры и комплекса программ, необходимость непрерывного контроля программных и аппаратных средств, методы и процессы периодических и заключительных испытаний системы. Выделен раздел рекомендаций процессов разработки ПК

– проектирования и программирования (кодирования) программ. Обращается внимание на необходимость самоконтроля логики и данных программ, на декомпозицию и модульное построение ПК, на стройность и простоту структуры программ для сокращения дефектов и ошибок. Для этого же предлагается подробная верификация программ по фазам ЖЦ ПК, планирование тестирования, проверка критических ситуаций, специфицирование испытаний и их документирование. В разделе интеграции аппаратных и программных средств рассматривается планирование и фазы этого процесса, контроль конфигурации и верификации системы, исправление дефектов и ошибок. Выделены аттестация качества ПК и системы, и представление отчетов о достигнутом качестве и функциональной безопасности. Рекомендуется формализовать процедуры сопровождения и модификации для обеспечения их корректности. Для обеспечения функциональной безопасности, обращается внимание на необходимость применения тренажеров и обучения операторов, а также периодических испытаний ПП и системы. В шести Приложениях детализируются процессы и рекомендации по описанию требований к ПП и их реализации, по планированию и обеспечению характеристик качества и функциональной безопасности, по выбору языка программирования, транслятора, редактора связей, а также по тестированию ПП. В подробных таблицах иллюстрируются процессы разработки ПП и их документирования. Содержание этой части стандарта не отличается методичностью изложения рекомендаций и процессов жизненного цикла комплексов программ и ее вряд ли целесообразно использовать при наличии представленных выше, более совершенных стандартов в области функциональной безопасности.

Основное содержание ***второй части*** стандарта базируется на требованиях МАГАТЭ по глубоко эшелонированной защите ПП и системы от отказов функционирования при отсутствии предумышленных негативных воздействий. Отмечается специфика программных дефектов и ошибок, состоящая в единичности и непредсказуемости положения и последствий, что требует их практически полного исключения и отличает от дефектов в аппаратуре, надежность и безопасность которой может рассчитываться аналитически. Рекомендуется создавать систему защитных барьеров и самоконтроля исполнения программ для обеспечения гарантированной работоспособности и безопасности систем управления на базе ПП. Предлагается совокупность методов предотвращения ошибок путем: создания разнообразия условий функционирования, N-версионного программирования, дублирования спецификаций компонентов при одинаковых функциях, доказательства формальной корректности программ. Специальный раздел посвящен развитию рекомендаций предыдущей части стандарта по выбору программного инструментария, обеспечивающего минимизацию дефектов и ошибок в программах по всем этапам ЖЦ ПП и системы. В отдельный раздел выделены процессы повторного использования ранее разработанных и апробированных на подобных системах компонентов комплексов программ. Рекомендуется проводить детальную оценку функций, качества, результатов опытной эксплуатации и совместимость интерфейсов

таких компонентов с вновь разработанной частью ПК, а также возможности их последующей модификации и сопровождения. В четырех Приложениях детализируются некоторые положения этой части стандарта.

Стандарт **МЭК 60880** практически не содержит принципиальных или существенных положений, которые не отражены в совокупности современных международных и национальных стандартов. Исключением является раздел защиты от отказов по общим причинам во второй части стандарта, в котором имеется ряд весьма полезных рекомендаций по обеспечению функциональной безопасности. Приведенные выше в данной главе стандарты полнее и более подробно регламентируют на современном уровне обеспечение функциональной безопасности в ЖЦ сложных комплексов программ и позволяют создавать программный продукт высокого качества и безопасности. Поэтому при создании ПП для атомных электростанций целесообразно применять в основном приведенные выше стандарты.

3.6 Особенности процессов разработки и документирования встроенных программных продуктов в стандарте ГОСТ Р 51904:2002

Стандарт **ГОСТ Р 51904:2002** – Программное обеспечение встроенных систем. Общие требования к разработке и документированию, – создан в развитие **ISO 12207:2008** с целью, учета специфики жизненного цикла программных средств встроенных систем реального времени высокого качества, преимущественно для авиационных, космических и транспортных систем. В стандарте значительное внимание уделяется обеспечению качества и функциональной безопасности ПП, чем структура и рекомендации этого стандарта наиболее близки к **IEC 61508**. В стандарте представлены: общие требования (п. 4), системные аспекты, связанные с разработкой ПП (п.5) и шесть крупных разделов (п.п. 6 – 11), ***подробно описывающие и рекомендующие основные процессы ЖЦ встроенных ПК***. Последовательно, детально рассмотрены требования и методы реализации процессов жизненного цикла сложных ПК. Выделена группа ***процессов планирования***, которые определяют и координируют для проекта действия процессов разработки и интегральных процессов.

Процессы разработки, в ходе выполнения которых, создается программный продукт, отражены в разделах:

- определение требований к ПП: состав работ, выполняемых при определении требований к ПП; установление модели жизненного цикла; критерии переходов между процессами; общие требования для разработки; стандарты; ПК многократного использования; требования к системе; отработка критических требований;
- планирование: состав работ, выполняемых в процессе планирования ПП; планирование среды жизненного цикла; язык программирования

и компилятор; стандарты разработки; состав работ, выполняемых в процессе кодирования программ;
- проектирование и разработка ПК: состав работ, выполняемых в процессе проектирования; потоки информации между процессами жизненного цикла системы и ПП; отказовые ситуации и назначение уровня ПК; анализ системных требований; анализ информации о потребностях пользователя; проектирование архитектуры системы; мониторинг функциональной безопасности ПП; подготовка руководств пользователя; интеграция компонентов.

Интегральные процессы, которые обеспечивают корректную реализацию и качество выполнения процессов разработки и их выходных данных детально представлены в разделах:

- верификация ПП: состав работ, выполняемых в процессе верификации; просмотры и анализы требований: верхнего уровня; архитектуры ПК; требований нижнего уровня; исходного кода; тестовых вариантов, процедур и результатов; выбор тестовых вариантов, основанных на требованиях; анализ тестового покрытия; интеграционное тестирование; квалификационное тестирование системы;
- управление конфигурацией ПК: состав работ, выполняемых в процессе управления конфигурацией; отчетность о дефектах, трассируемость и корректирующие действия; архивирование и получение документов; выпуск версий; контроль среды жизненного цикла; аудит конфигурации;
- обеспечение качества ПП: состав работ, выполняемых в процессе обеспечения качества; просмотр согласованности; документирование обеспечения качества; независимость в обеспечении качества ПП;
- сертификационное сопровождение: средства согласования и планирования; обоснование согласованности; минимальный состав документов жизненного цикла ПП, передаваемых сертифицирующей организации; документы жизненного цикла , относящиеся к типовому проекту; критерии и документация по аттестации для инструментальных средств разработки и верификации ПП.

Интегральные процессы должны выполняться частично одновременно с процессами разработки ПК. В рамках конкретного проекта должны быть установлены одна или несколько моделей жизненного цикла ПК, в соответствие, с которыми выбираются необходимые процедуры для каждого процесса, определяется последовательность их выполнения, назначаются ответственные за выполнение работ. Для конкретного проекта последовательность процессов определяется сложностью проекта в целом, функциональными возможностями разрабатываемой системы, объемом и сложностью ПП, стабильностью требований, использованием ранее полученных результатов, стратегией разработки и возможностями аппаратных средств.

Для всех работ по созданию ПП ***должны использоваться систематизированные, зарегистрированные методы***. План разработки ПП должен содержать описание этих методов или включать в себя ссылки на источники и стандарты, в которых они описаны. Следует разработать и использовать руководства для представления требований: проекта, кода, тестовых вариантов, тестовых процедур и результатов тестирования. В документе представлены подробные рекомендации, на которых акцентируется дополнительное внимание, и детализируются положения стандарта **ISO 12207:2008**. ***Основные требования и рекомендации стандарта сводятся к следующим***.

Разработчик должен принимать участие в определении и документировании проектных решений системного уровня. Эти решения являются прерогативой разработчика, если они формально не преобразованы в требования при выполнении контракта. Он ответствен за выполнение всех требований и демонстрацию этого выполнения посредством квалификационного тестирования. Реализация проектных решений, действующих как внутренние требования, должна быть подтверждена внутренним тестированием разработчика, выполнение которого нет необходимости демонстрировать заказчику.

Рекомендуется участие разработчика в определении и документировании проекта архитектуры ПК (идентификации компонентов, их интерфейсов и концепции их совместного выполнения) и в прослеживании соответствия между компонентами и системными требованиями. В ***процессе оценки безопасности*** должно устанавливаться, как архитектурное проектирование ПК предотвращает аномальное поведение при появлении отказовых ситуаций. Должны применяться архитектурные стратегии, которые позволяют ограничивать воздействие дефектов, обнаруживать ошибки и обеспечивать приемлемую реакцию ПП для устранения их воздействия. Библиотека разработки может быть частью среды разработки и среды верификации. Следует ***сопровождать Библиотеку разработки ПП на протяжении действия контракта.***

Разработчик должен подготовить исполняемое ПП для передачи в организацию, осуществляющую сопровождение, а также файлы, необходимые для установки и эксплуатации ПП на объектной ЭВМ. Он должен принимать участие в совместных с заказчиком технических просмотрах, проводимых в течение всего периода выполнения контракта. В этих просмотрах, как со стороны разработчика, так и со стороны заказчика должны принимать участие лица с достаточными техническими знаниями о разрабатываемом ПП.

Следует ***осуществлять контроль за критическими для выполнения контракта ситуациями***, которые могут возникнуть во время разработки ПК. Необходимо выявлять, идентифицировать и анализировать ***потенциальные технические, стоимостные или временные критические ситуации и риски***; разработать стратегии для предотвращения или устранения таких ситуаций; регистрировать возможные риски и стратегии их предотвращения и реализовать эти стратегии в соответствии с Планом. В течение всего жизненного цикла ПП должны создаваться документы, чтобы планировать требуемые действия, управлять, объяснять, регистрировать выполнение требуемых действий. Эти документы должны отражать реализацию процессов жизненного цикла, сертификацию системы и последующую модификацию ПП. Заказчик должен осуществлять выбор необходимого и экономически обоснованного состава и содержания документов для конкретной разработки из представленных в стандарте 39-и типов и структур. Их форма должна обеспечивать эффективный поиск и просмотр документов жизненного цикла ПП в процессе обслуживания системы. ***Состав документов и их конкретная форма должны быть определены в Плане документирования***. Заказчик на основании информации, полученной от разработчика, должен определить, какие руководства являются необходимыми для данной системы, и требовать разработки только этих руководств.

Глава 4. Испытания надежности и функциональной безопасности программных продуктов

4.1. Испытания надежности функционирования программных продуктов

В составе требований к системам и программным продуктам, функционирующим в реальном времени, особое значение имеют ***динамические функции и характеристики***. Для обеспечения их высокого качества недостаточны отдельные сценарии и процедуры тестов, а необходимо создавать ***потоки динамических тестов в реальном времени***, адекватные соответствующим данным при функционировании внешней среды систем и/или пользователей – рис. 4.1. Эти потоки тестов должны обеспечивать динамическую проверку функционирования комплексов программ на соответствие требованиям, выявление дефектов и ошибок в реализации их функций и характеристик в реальном времени. Основная особенность такого тестирования состоит в создании динамической среды функционирования программного продукта, максимально приближенной к реальной, при его практическом применении.

Задача состоит в определении соответствия требованиям, функций и характеристик программного продукта при различной интенсивности потоков тестов, адекватных нормальным условиям применения программного продукта, а также критическим по составу и интенсивности, для выявления предельных условий его работоспособности. Такие ***условия испытаний отражаются на интегральных характеристиках***, на оценивании надежности и/или безопасности, а также рисков применения программного продукта. Для комплексов программ реального времени ***особое значение могут иметь*** причины и методы уменьшения ***рисков надежности и производительности***, вследствие дефектов и ошибок, а также при формировании и реализации требований к этим характеристикам программных продуктов. Эти взаимосвязанные характеристики качества программного продукта зависят от одних и тех же свойств воздействий из внешней среды, требуют совместного анализа и методов для выявления и устранения дефектов. Локализация и устранение динамических дефектов обычно осуществляется вне реального времени, путем применения ***детерминированных сценариев и тестовых процедур***, а иногда за счет изменения требований заказчика.

Испытания надежности функционирования сложных программных продуктов:

- экспериментальные методы оценивания надежности программных продуктов в штатном режиме;

- форсированные испытания для оценивания надежности программных продуктов:

- повышение надежности комплексов программ, путем оперативного контроля и рестарта.

Испытания функциональной безопасности сложных программных продуктов:

- оценивание эффективности тестирования функциональной безопасности программных продуктов;

- учет ограниченности ресурсов для обеспечения функциональной безопасности программных продуктов;

- удостоверение достигнутой функциональной безопасности систем с программными продуктами.

Испытания производительности и динамического использования ресурсов компьютерами сложных программных продуктов:

- нарушения временного баланс между длительностью решения задач и производительностью компьютера в реальном времени;

- оценивания предельной пропускной способности системы с данным программным продуктом реального времени;

- определение качества и функциональной безопасности динамического функционирования программного продукта при перегрузке.

Испытания эксплуатационной документации на соответствие требованиям к программным продуктам

- тестирование в документах требований заказчика к функциям и характеристикам программного продукта;

- тестирование эксплуатационной документации на практичность.

Рис. 4. 1

Оценивание надежности программных комплексов включает измерение количественных характеристик: завершенности, устойчивости к дефектам, восстанавливаемости и доступности-готовности. При этом предполагается, что в

контракте, техническом задании или спецификации требований зафиксированы, и утверждены заказчиком определенные значения этих характеристик и их приоритеты. Измерения проводятся при функционировании готового программного продукта для сопоставления с заданными требованиями и для оценивания рисков соответствия спецификациями требований [8, 12, 26, 33]. Испытания для оценки надежности комплекса программ должны проводиться в тестовом окружении, которое максимально приближено к реальным условиям применения системы. Входные параметры тестов следует задавать на основе вероятностного распределения соответствующих характеристик или их наборов при эксплуатации программного продукта, исходя из частоты возможных сценариев работы пользователей или системы.

Надежность можно оценивать в процессе разработки по прогнозируемой плотности обнаружения скрытых дефектов и ошибок, а также по плотности выявляемых и устраняемых ошибок выходных результатов при тестировании динамического функционирования комплекса программ. Степень покрытия тестами структуры функциональных компонентов и комплекса программ в целом может служить ***ориентиром для прогнозирования их потенциальной надежности***. Распределение реальных длительностей и эффективности восстановления при ограниченных ресурсах для функционирования программ, может рассматриваться как дополнительная составляющая при оценивании надежности.

Для прямых, количественных измерений надежности необходимы ***инструментальные средства, встроенные в операционную систему или в соответствующие компоненты комплекса программ***. Эти средства должны, в динамике реального функционирования программного продукта, автоматически селектировать и регистрировать аномальные ситуации, дефекты и искажения вычислительного процесса, программ и данных, выявляемые аппаратным, программно-алгоритмическим контролем или пользователями. Накопление и систематизация проявлений дефектов при исполнении программ позволяет оценивать основные показатели надежности, помогает определять причины сбоев и отказов и подготавливать данные для повышения надежности программных продуктов. Регулярная регистрация и обобщение таких данных способствует устранению ситуаций, негативно влияющих на функциональную пригодность и другие важные динамические характеристики программного продукта.

Прямые экспериментальные методы оценивания интегральных характеристик надежности программных продуктов, в ряде случаев весьма трудно реализовать при нормальных штатных условиях функционирования крупных комплексов программ, из-за больших значений времени наработки на отказ (сотни и тысячи часов), которые необходимо достигать при разработке и фиксировать при испытаниях. Сложность выявления и регистрации редких отказов, а также высокая стоимость экспериментов при длительном многосуточном функционировании комплексов программ приводят к тому, что на испытаниях получаются ***малые выборки зарегистрированных отказов и низка достоверность оценки надежности***. Кроме того, при таких экспериментах трудно гарантировать полную представительность выборки исходных данных, так как проверки определяются конкретными условиями применения данного программного продукта на испытаниях.

Для диагностики и устранения случайных редких отказов должна быть организована ***служба их регистрации*** с максимально полным фиксированием характеристик ситуаций, при которых проявилась каждая. Сбои в аппаратуре носят более или менее случайный характер и полное повторение отказовой ситуации мало вероятно. Ошибки и дефекты программ содержатся в ***определенном месте*** и должны регулярно проявляться при полном повторении внешних ситуаций. На основе таких признаков и, по возможности, детального описания ситуаций возникновения отказа, могут строиться предположения о его причине.

Эти гипотезы должны использоваться, прежде всего, для дополнительного, интенсивного тестирования всей системы. Если в аппаратуре не обнаруживается причина отказа, то следует провести углубленное тестирование функционального компонента программного комплекса, в котором по предположению может содержаться дефект, вызвавший отказ. Для повышения надежности при высокой наработке на отказ необходима тщательная, систематическая работа специалистов, накапливающих, регистрирующих и анализирующих все отказовые ситуации при функционировании комплекса программ. Эти специалисты должны также регистрировать все проведенные корректировки для прогнозирования причин появления возможных, дополнительных источников отказов, вызванных дефектами корректировок.

Для выявления ***тенденции изменения показателей надежности***, их зарегистрированные значения необходимо связывать во времени с моментами корректировки программ. Анализируя корреляцию между значениями надежности и процессом изменения программ, можно выявлять ***некоторые корректировки, которые содержат ошибки и снижают надежность***. Получающиеся при этом показатели позволяют прогнозировать число ошибок, подлежащих исправлению для достижения требуемых значений надежности в зависимости от длительности испытаний. В результате может быть оценена наработка до следующего выявления ошибки или отказа.

При заключительных приемо-сдаточных и сертификационных испытаниях ***для достоверного определения надежности*** организуются многочасовые и многосуточные прогоны динамического функционирования комплекса программ ***в реальной и/или имитированной внешней среде*** в условиях широкого варьирования исходных данных с акцентом на стрессовые ситуации, стимулирующие проявления ***угроз надежности***. Такие прогоны позволяют измерять достигнутые характеристики надежности и определять степень их соответствия требованиям технического задания, а также закреплять их в технических условиях и документации на программный продукт.

Если интенсивное тестирование программ в течение достаточно длительного времени не приводит к обнаружению дефектов или ошибок, то у специалистов, ведущих испытания, создается ощущение бесполезности дальнейшего тестирования программного продукта, и он передается на эксплуатацию. Систематическое исследование характеристик сложных ПК позволило оценить ***темп обнаружения дефектов, при котором крупные программные продукты передаются на регулярную эксплуатацию. Экспериментально оценено, что*** предельное выявление 0,002 – 0,005 дефектов в день на человека, – специалиста по испытаниям или всех пользователей в совокупности, соответствует обнаружению только около одной ошибки или дефекта каждые два – три месяца использования ПП. Интенсивность обнаружения ошибок ниже 0,001 ошибок в день на человека, т.е. меньше одной ошибки в год на трех-четырех специалистов, непосредственно выполняющих динамическое квалификационное тестирование и/или эксплуатацию комплекса программ, по-видимому, может служить ***эталоном высокой надежности*** обработки информации. Если динамическое функционирование программного продукта происходит непрерывно, то эти показатели соответствуют высокой наработке на обнаружение дефектов или отказов ***порядка 5 – 10 тысяч часов и коэффициенту готовности выше 0,99***. При использовании этого критерия обычно учитывается календарное время испытаний, включающее длительность непосредственного тестирования, как для обнаружения, так и для локализации дефектов, а также длительность корректировки программ и других вспомогательных работ для восстановления нормального функционирования программного продукта.

Форсированные (стрессовые) испытания для оценивания надежности (а также функциональной безопасности) программных продуктов значительно отличаются от традиционных методов испытаний аппаратуры. Основными факторами, влияющими на надежность ПП, являются исходные данные и их взаимодействие с дефектами и ошибками программ или сбоями в компьютерах. Поэтому форсирование испытаний надежности осуществляется повышением интенсивности искажений исходных данных и расширением варьирования их значений, а также специальным увеличением интенсивности потоков информации и загрузки программами компьютеров выше нормальной.

Планирование форсированных динамических испытаний должно предусматривать последующий ***пересчет полученных значений надежности*** на условия нормального функционирования. Для этого необходимо оценивать надежность испытываемых программ в зависимости от интенсивности искажений данных или от характеристик перегрузки компьютера, а также применять способы корректного пересчета получаемых показателей на нормальные условия эксплуатации. При форсированных испытаниях целесообразно выделять следующие ***режимы тестирования***:

- полное искажение, предельные и критические значения ключевых параметров тестов каждого типа внешней информации и воздействий пользователей;

- предельные и критические сочетания значений различных взаимодействующих параметров тестов при эксплуатации программного продукта;
- предельно большие и малые интенсивности суммарного потока и каждого типа внешней информации;
- умышленные нарушения пользователями определенных положений инструкций и рекомендаций эксплуатационной документации на программный продукт.

Как вид форсированных испытаний можно рассматривать тестирование и контроль результатов функционирования одних и тех же ПП при увеличении числа испытываемых экземпляров и нормальных исходных данных – ***Бета-тестирование***. На этапе тестирования в соответствии с эксплуатационной документацией, пользователями некоторого предварительного тиража программного продукта, происходит естественное расширение вариантов исходных данных, если они взаимно независимы. Это увеличивает наборы тестов и тем самым дает возможность оценивать наработки на отказ в сотни и тысячи часов. Они позволяют выявлять и устранять значительное число дефектов за относительно ***небольшое календарное время*** и тем самым доводить надежность до требуемого уровня. Однако следует учитывать, что при этом пропорционально возрастает суммарная трудоемкость таких испытаний.

Особым видом форсированных испытаний является целенаправленное тестирование эффективности средств оперативного контроля и восстановления программ, данных и вычислительного процесса ***для оценивания восстанавливаемости***. Для этого имитируются запланированные условия функционирования программ, при которых в наибольшей степени стимулируется срабатывание средств программного рестарта и оперативного, автоматического восстановления работоспособности.

Следует особо отметить трудности достижения и регистрации надежности программ, характеризующейся наработкой на отказ >>100 ч. При такой надежности резко возрастают сложность обнаружения возникающих отказов и диагностирования их причин. Наработка на отказ в тысячи часов в ряде случаев достигалась только при эксплуатации и сопровождении сложных программных продуктов в течение нескольких лет. При требовании особо высокой надежности функционирования, суммарные затраты ресурсов на ее достижение и оценивание могут увеличиваться на порядок. Однако требующееся увеличение затрат для получения такой высокой надежности программного продукта в процессе разработки трудно обеспечить практически. Поэтому для ее достижения активно применяются различные методы программной защиты от сбоев и отказов программ (методы оперативного рестарта). Они позволяют замедлить рост затрат ресурсов на разработку при повышении требований к их надежности.

В любых ситуациях функционирования сложных комплексов программ, прежде всего, должны исключаться катастрофические последствия и длительные

отказы или в максимальной степени смягчаться их негативное влияние на результаты, выдаваемые пользователю. Неизбежность дефектов и ошибок в сложных комплексах программ, искажений исходных данных и аппаратурных сбоев приводит к необходимости регулярного контроля процесса исполнения комплекса программ, и сохранности данных. Предвидеть заранее все ситуации исполнения программ, и протестировать при них крупные программные продукты, оказывается невозможным из-за их огромного количества. Поэтому применяются методы, которые направлены на ***оперативное обнаружение последствий*** дефектов и аномального функционирования программ, а также ***на автоматическое восстановление (рестарт) нормального вычислительного процесса*** и искаженных текстов программ и данных. [7, 8, 33].

Для снижения влияния негативных возмущений различных типов и происхождения на результаты, а также для защиты вычислительного процесса и информации программно-алгоритмическими методами в комплексы программ реального времени должна иметься ***избыточность ресурсов***. Избыточность вычислительных ресурсов необходима, прежде всего, для селекции оперативных искажений процесса функционирования ПП и для выработки решений по снижению последствий этих аномалий. Основная цель использования избыточности состоит в ограничении или исключении возможности аварийных последствий от динамических возмущений, соответствующих отказу системы. Любые дефекты при исполнении программ необходимо блокировать и по возможным последствиям сводить до уровня сбоя путем оперативного автоматического восстановления. При этом не обязательно сразу устанавливать причины искажения, главная задача сводится к ***максимально быстрому восстановлению*** нормального функционирования и ограничению его негативных последствий.

Введение средств контроля функционирования и помехозащиты в программы, позволяет ***скомпенсировать влияние на надежность неполной корректности программных продуктов***, а также снизить негативные воздействия внешних возмущений различных типов. Однако только средствами контроля и обеспечения программной помехозащиты невозможно достигнуть высокой надежности динамического функционирования ПП. Требуемое сокращение вероятности отказа может достигаться путем ***повышения затрат ресурсов на тестирование*** и увеличения его длительности, что допускает соответствующее снижение затрат на средства обнаружения искажений и восстановление. Подобные результаты можно получать за счет высокой эффективности ***средств оперативного контроля и восстановления***, что позволяет быстрее иметь менее корректные комплексы программ с той же надежностью функционирования. Таким образом, может быть сформулирована ***оптимизационная задача***: распределение ограниченных ресурсов на создание корректных программ, а также на их помехозащиту, обеспечивающие совместно ***заданные характеристики надежности их функционирования при минимальных интегральных затратах.***

Наряду с оперативной реакцией на искажения функционирования программного продукта, должны вестись ***мониторинг и накопление информации о их проявлениях*** с тем, чтобы использовать эти данные для локализации первичных источников дефектов и ошибок и исправления соответствующих программ или компонентов аппаратуры. Подготовку, статистическую обработку и накопление данных проявлений искажений целесообразно проводить автоматически с выдачей периодически или по запросу сводных данных на индикацию для подготовки специалистами решений о корректировке программ или восстановлении аппаратуры.

Контроль работоспособности комплекса программ, исправление дефектов и восстановление при отказовых ситуациях сокращают ресурсы времени функционирования компьютера, доступные для выполнения основных функций, и в общем случае негативно ***отражаются на производительности системы***. Однако если некоторые ситуации контроля и восстановления проводятся достаточно быстро так, что их последствия не фиксируются как отказ и не отражаются на снижении работоспособности, то такие затраты времени полезны для решения функциональных задач и должны быть отнесены к улучшению качества программного продукта. Таким образом, при анализе ***эффективности применения рестарта возникают две задачи***:

- оценивание снижения производительности (временной эффективности) компьютера вследствие: затрат его ресурсов на контроль и оперативное восстановление вычислительного процесса, программ и данных, а также увеличения времени не занятого решением функциональных задач;
- оценивание сокращения числа отказовых ситуаций, отражающихся на потере работоспособности и снижении надежности, за счет уменьшения времени восстановления, когда оно не достигает порогового значения, достаточного для регистрации отказа и уменьшения надежности.

Оценивание рациональных затрат ресурсов на надежность состоит в определении оптимальной длительности разработки и ***тестирования комплекса программ для устранения дефектов***, при которой затраты ресурсов на ее выполнение и затраты на оперативный контроль, рестарт и восстановление в процессе эксплуатации в сумме – минимальные. Эти суммарные затраты могут обеспечивать одну и ту же интенсивность пропуска не обнаруженных дефектов, приводящих к отказу при различной длительности отладки. Задаваясь либо допустимой ***длительностью тестирования***, либо требуемой ***наработкой*** на отказовую ситуацию, либо ***суммарными затратами***, можно оценивать остальные характеристики, влияющие на надежность программного продукта, а также значения этих параметров.

4.2. Испытания динамических характеристик программных продуктов на соответствие требованиям функциональной безопасности

Функциональная безопасность программных продуктов и систем зависит от отказовых ситуаций, негативно отражающихся на работоспособности и реализации их основных функций, причинами которых могут быть дефекты и аномалии в аппаратуре, комплексах программ, данных или вычислительных процессах (рис. 4.2). При этом катастрофически, критически или существенно искажается процесс функционирования программных продуктов и/или систем, что наносит значительный ущерб при их применении. Основными источниками отказовых ситуаций могут быть некорректные исходные требования, сбои и отказы в аппаратуре, дефекты или ошибки в программах и данных функциональных задач, проявляющиеся при их динамическом исполнении в соответствии с назначением.

Методы подготовки тестов для испытаний крупных комплексов программ:
- классификация методов подготовки тестов;
- оценки методов автоматизации разработки тестов.

Требования к генерации динамических тестов внешней среды в реальном времени:
- задачи автоматизированного формирования динамических тестов внешней среды в реальном времени;
- преимущества программной имитации динамических тестов в реальном времени.

Компоненты генераторов динамических тестов внешней среды в реальном времени:
- исходные данные для имитации тестов внешней среды;
- набор средств для формирования тестов внешней среды.

Средства обработка результатов динамического тестирования комплексов программ в реальном времени:
- средства оперативной обработки результатов динамического тестирования в реальном времени;
- средства определения характеристик комплекса программ и системы в реальном времени.

Оценки эффективности динамической генерации тестов внешней среды в реальном времени:
- оценки затрат на программную имитацию тестов;
- оценки экономической эффективности программной имитации тестов реального времени.

Рис. 4. 2.

При таких воздействиях, внешняя, функциональная работоспособность систем может разрушаться не полностью, однако ***невозможно полноценное выполнение заданных функций и требований к программному продукту***. В реальных сложных системах, связанных с безопасностью, возможны катастрофические последствия и отказы функционирования с большим ущербом, даже при отсутствии воздействия лиц, заинтересованных в нарушениях работоспособности систем и ПП [17, 26, 31].

Понятия, методы испытаний и характеристики функциональной безопасности программных продуктов и систем близки к аналогичным для на-

дежности. Поэтому способы оценки и испытаний функциональной безопасности могут базироваться на методах тестирования, определения и обеспечения надежности функционирования комплексов программ. При более или менее одинаковых источниках угроз и их проявлениях эти понятия можно разделить ***по величине негативных последствий*** и ущерба при возникновении отказовых ситуаций. Чем сложнее системы и чем выше к ним требования безопасности, тем неопределеннее функции и характеристики тестирования требований для обеспечения их безопасности. ***Неопределенности начинаются с требований заказчиков,*** которые при формулировке технического задания и спецификаций не полностью формализуют и принципиально не могут обеспечить достоверное содержание всего адекватного набора характеристик и значений требований безопасности, которые должны быть при завершении проекта и предъявлении конечного программного продукта заказчику. Эти требования итерационно формируются, детализируются и уточняются по согласованию между всеми участниками проекта вследствие естественной ограниченности первичных исходных данных, и изменения их под влиянием объективных и субъективных воздействий со стороны различных процессов на последовательных этапах ЖЦ комплекса программ.

Всегда не полностью, с необходимой детализацией определены и описаны все характеристики, особенности функционирования и безопасности объектов внешней среды. Эти характеристики в той или иной степени обычно находятся под воздействием управляемой системы. Квалификация и субъективные свойства потребителей и пользователей изменяются по мере освоения функциональных возможностей системы и ее работоспособности, что увеличивает неопределенность ее реальной безопасности. Различия свойств персонала, применяющего систему, дополнительно увеличивают неопределенность значений безопасности и трудности ее прогнозирования при тестировании с учетом множества ***субъективных факторов различных специалистов, участвующих в эксплуатации***.

При анализе характеристик функциональной безопасности целесообразно выделять и учитывать особенности ***двух классов систем и их программных продуктов***. ***Первый класс*** составляют системы, имеющие встроенные комплексы программ ***жесткого регламента реального времени***, автоматизировано управляющие внешними объектами или процессами. Время необходимой реакции на отказовые ситуации таких систем обычно исчисляется секундами или долями секунды, и процессы восстановления работоспособности должны проходить за это время, в достаточной степени автоматизировано (бортовые системы в авиации, на транспорте, в некоторых средствах вооружения, системы управления атомными электростанциями).

Системы второго класса, применяются для управления процессами и обработки деловой информации из внешней среды, в которых активно участвуют специалисты-операторы (банковские, административные, штабные военные системы). Допустимое время реакции на опасные отказы в этих системах может составлять десятки секунд и минуты, и операции по восстановлению работоспособности частично могут быть доверены специалистам-администраторам по обеспечению функциональной безопасности.

Тестирование и обеспечение функциональной безопасности сложных систем должны решаться с учетом одновременного динамического развития всех компонентов внешней среды, и факторов, непрерывно изменяющихся и воздействующих на результаты их решения. Однако такой сложный, непрерывный, многосвязный процесс трудно реализовать практически и его целесообразно решать поэтапно, возможно с необходимыми итерациями и упрощениями. Эти факторы влияют на ***неопределенность критериев, методов и оценивания значений эффективности функциональной безопасности*** конкретных программных продуктов и систем. Существующие технологии тестирования способствуют повышению функциональной безопасности, снижению потенциального ущерба и рисков, однако практически всегда остается открытым вопрос, насколько применяемые методы оправдывают затраты на реализацию требований заказчика. Испытания, эксплуатация и сертификация способствуют снижению неопределенности оценок эффективности и итерационному приближению к практически приемлемому уровню функциональной безопасности программных продуктов.

В связи с ростом сложности проектов и применения современных систем на базе компьютеров, быстро возрастает ответственность решаемых ими задач. Одновременно возрастает сложность внешней и операционной среды, в которой функционируют комплексы программ и ***ответственность функций систем, связанных с безопасностью***. Объективное повышение сложности функций, реализуемых программными продуктами в современных системах, непосредственно приводит к увеличению их объема и трудоемкости создания. Соответственно росту сложности программ возрастает относительное и абсолютное количество выявляемых и остающихся в них дефектов и ошибок, что ***отражается на снижении потенциальной безопасности их функционирования***.

Работоспособность ПП может быть обеспечена при исходных данных, которые использовались при их разработке, отладке и испытаниях. Реальные исходные данные могут иметь значения, отличающиеся от заданных техническим заданием и от используемых при эксплуатации программ и баз данных. При таких исходных данных функционирование программного продукта трудно предсказать заранее, и весьма вероятны различные аномалии, завершающиеся отказами, отражающимися на безопасности. Это приводит к практической ***невозможности достоверных априорных аналитических оценок функциональной безопасности*** комплексов программ при ее высоких значениях.

Достижение требуемой функциональной безопасности систем, содержащих программные продукты реального времени, решаются путем использования ***современных регламентированных технологических процессов динамического тестирования***, подобных применяемым при обеспечении надежности. Они должны быть поддержаны группой международных стандартов, оп-

ределяющих состав и процессы выполнение требований к заданной функциональной безопасности систем и комплексов программ. Для систематической, координированной борьбы с угрозами безопасности программных продуктов необходимы исследования факторов, влияющих на функциональную безопасность со стороны случайных дефектов и ошибок, существующих и потенциально ***возможных в конкретных системах и комплексах программ***. Это позволит целенаправленно разрабатывать и применять методы и средства обеспечения функциональной безопасности критических программных продуктов различного назначения при реально достижимом снижении уровня дефектов и разработки. Проблема в значительной степени решается посредством применения современных методов, инструментальных средств и стандартов, поддерживающих системный анализ, технологию проектирования, производства, тестирования и сопровождения систем, и их программных продуктов.

Требования к функциям систем и программных продуктов, а также к безопасности их функционирования должны ***соответствовать доступным ресурсам для их реализации*** с учетом допустимого ущерба – рисков вследствие отказов при неполном выполнении требований. ***Ограниченности ресурсов*** различных видов для тестирования и обеспечения функциональной безопасности значительно влияют на технико-экономические показатели, качество и функциональную безопасность всей системы и ПП. В результате сложность комплексов программ, а также доступные ресурсы для их реализации становятся косвенными критериями или факторами, влияющими на выбор методов разработки, на достигаемую безопасность программных продуктов.

Разработку систем должны завершать ***комплексные испытания и удостоверение достигнутой функциональной безопасности и надежности*** систем с программным продуктом, предусматривающие возможность совершенствования их характеристик путем соответствующих корректировок программ. Повышение функциональной безопасности целесообразно также путем анализа выявленных дефектов и ***оперативного восстановления вычислительного процесса, программ и данных*** (рестарта) после обнаружения аномалий и отказов функционирования программного продукта. Этому может способствовать накопление, мониторинг и хранение данных о выявленных дефектах, сбоях и отказах в процессе исполнения программ и обработки данных. Испытания и измерения реальной функциональной безопасности систем и комплексов программ связаны с необходимостью определения возможного материального или иного ущерба – риска при отказовых ситуациях. Однако аналитически описать и измерить в ***возможный обобщенный ущерб*** (человеческий, материальный, социальный, внешней среде или иной) при нарушении безопасности для критических систем разных классов весьма трудно. Поэтому, на практике в ряде случаев, ущерб от реализации угроз целесообразно характеризовать интервалами времени между их проявлениями, нарушающими безопасность применения программного продукта, или наработкой на опасные отказы, отражающиеся на безопасности с большим ущербом.

До начала испытаний функциональной безопасности комплексов программ подлежат проверке и ***паспортизации средства***, обеспечивающие получение эталонных данных, средства формирования тестов от внешних объектов, средства фиксирования и обработки результатов испытаний. При этом важную роль играет оценка и обеспечение методической достоверности результатов испытаний. ***Методическая достоверность испытаний функциональной безопасности программных комплексов определяется следующими факторами:***

- полнотой Программы испытаний и корректностью методик тестирования, по охвату возможных сценариев функционирования компонентов и комплекса программ, и исходных требований для программного продукта;
- достоверностью и точностью эталонных значений требований функциональной безопасности, с которыми должны сравниваться результаты испытываемого комплекса программ, которые служат опорными при расчете параметров, зафиксированных в техническом задании и спецификациях требований;
- адекватностью и точностью моделей, используемых для формирования сценариев тестов от внешней среды и их реакции на управляющие воздействия программного комплекса;
- точностью и корректностью регистрации и обработки результатов испытаний, а также сравнения полученных данных с требованиями и эталонами технического задания и спецификаций.

Определение задач испытаний функциональной безопасности предусматривает анализ документов, в которых сформулированы требования к программному продукту, и выделение необходимых работ, чтобы затем проверять функции и характеристики, отражающие эти требования. В дополнение к обычному анализу требований функциональной безопасности необходимо учесть задачи, которые не имеют прямого отношения к документам с требованиями, но ***влияют на процессы тестирования***:

- составление плана и Программы проведения испытаний функциональной безопасности;
- разработка тестовых сценариев и динамических моделей внешней среды для генерации тестов;
- отладка тестовых сценариев и генераторов динамических тестов;
- верификация, проверка, выявление и исправление дефектов генераторов динамических тестов;
- определение качества тестов и степени реализации функций и характеристик безопасности программного продукта, процесса их сборки и использования;
- пересмотр и корректировка эксплуатационной документации.

После подготовки такого перечня работ следует его ***ревизовать*** вместе с группой тестирования, и провести через полный жизненный цикл, определяя

дополнительные компоненты и задачи, которые должны выполняться для корректных испытаний функциональной безопасности. Такой перечень является динамичным документом, претерпевающим изменения на протяжении всего жизненного цикла и испытаний по мере роста понимания того, что должно быть сделано для получения программного продукта высокого качества, ***соответствующего требованиям и эталонам заказчика***. Декомпозиция работ часто может выглядеть как иерархический упорядоченный список видов деятельности, в которых каждой задаче присваивается идентификатор. Дескриптор позволяет отслеживать задачу на протяжении всего жизненного цикла разработки и предоставляет возможность связывать измерения трудозатрат или расходов ресурсов с различными задачами функциональной безопасности.

Все что, декларировано заказчиком, должно быть отражено в требованиях, в Программе и плане проведения испытаний с тем, чтобы это протестировать до поставки продукта заказчику. Основным источником, позволяющим фиксировать, что было согласовано с заказчиком, служат ***документы определения требований и эталонов***. Перечень требований должен сопровождаться технико-экономическим обоснованием и контрольной таблицей для вычисления трудозатрат, необходимых для проведения испытаний безопасности программного комплекса. Каждая позиция этого перечня должна соответствовать конкретным пунктам документов описания требований, в то же время каждой такой позиции должен соответствовать один или большее число тестов, определенных в ***Программе и спецификациях методик испытаний функциональной безопасности***.

Внутренние испытания компонентов и программного комплекса (***испытания менеджера проекта***), которые зачастую совмещаются с завершением комплексной отладки, должны оформляться документально и могут являться основанием при предъявлении программного продукта заказчику на испытания для завершающего оценивания функций и функциональной безопасности программного продукта (см. **ISO 12207:2008, ISO 16326**). Руководитель разработки должен ***оценить уровень реализации проекта***, состояние комплекса программ, результаты тестирования и документацию для пользователя, учитывая:

- полноту охвата испытаниями всех требований функциональной безопасности и эталонов к функциональным компонентам и к комплексу в целом;
- согласованность с требуемыми заказчиком результатами применения программного продукта;
- возможность интеграции и тестирования безопасности комплекса программ в составе аппаратуры системы;
- возможность функционирования и сопровождения версии программного продукта в соответствии с требованиями контракта.

Этапы тестирования компонентов и комплекса в целом обычно реализуются с позиции последовательного увеличения функциональной сложности тестов и

взаимодействия с объектами внешней среды. Этапы и процессы тестирования ***с целью формального удостоверения для заказчика достигнутых характеристик и реализации требований*** к комплексу программ и его компонентам в составе системы регламентированы в стандартах **ISO 12207:2008**. В них выделены ***основные, функциональные*** этапы реализации испытаний функциональной безопасности на соответствие требованиям заказчика.

Результаты этих действий должны быть включены в отчет для заказчика о результатах проведенных разработчиком ***предварительных испытаниях комплекса программ***. Группа управления проектом должна установить, что выполненное комплексное испытание достаточно, и дальнейшее продолжение комплексного тестирования, по всей вероятности, не выявит новых дефектов, связанных с интеграцией программных компонентов. Совещание разработчиков и заказчика, принимает решение о том, что критерии завершения испытаний вне системы выполнены.

Интеграция и испытание комплекса программ в составе системы должно содержать их объединение, тестирование полученного в результате комплекса, с целью определения работают ли они корректно вместе, как требуется по контракту. При интеграции ***функциональных компонентов,*** комплексов программ ***реального времени*** целесообразно выделять этапы:

- комплексирование и испытания функциональных групп программ без взаимодействия с другими компонентами и возможно без подключения к операционной системе реального времени;
- интеграция и испытания функциональных групп программ с учетом взаимодействия с некоторыми другими компонентами и с базой данных;
- интеграция и испытания программных компонентов в реальном времени во взаимодействии с другими функциональными компонентами и с компонентами операционной системы и базы данных.

После интегрирования всех откорректированных функциональных компонентов начинаются испытания функциональной безопасности в составе комплекса программ в целом на соответствие требованиям. Для них наиболее характерны следующие ***этапы интеграции и испытания версии программного продукта в реальном времени***:

- по данным моделирующего стенда или генераторов динамических тестов, имитирующих отдельные объекты и функционирование внешней среды;
- с имитаторами отдельных объектов внешней среды и с реальными воздействиями от операторов-пользователей;
- в полностью адекватной реальной или имитированной внешней среде и с реальными воздействиями от операторов-пользователей на соответствие требованиям.

При испытаниях системы в целом заказчику демонстрируется, что ***удовлетворены все требования*** технического задания, – функции и характеристики

качества соответствуют условиям контракта. Оно должно покрывать все требования в спецификациях системы и подсистем, а также требования к интерфейсу с внешней средой. Испытания функциональной безопасности должны включать тестирование на объектной вычислительной системе или на альтернативной модели системы, одобренной представителем заказчика. В процессе ***системного испытания*** проверяется интеграция отдельных частей, в совокупности составляющих систему в целом. Испытание на системном уровне обычно проводится специальной группой специалистов заказчика, которая должна провести анализ с целью определения выполнения требований и компонентов функциональности, которые ***вызывают наибольшее количество проблем***. ***Тест-менеджер несет ответственность*** за проведение испытаний согласно Программе и плану-графику, а также за распределение и перераспределение работ между тестировщиками для разрешения проблем, возникающих в ходе испытаний функциональной безопасности.

Испытания функциональной безопасности должны быть выполнены в соответствии с ***утвержденными заказчиком*** планом, Программой, тестовыми вариантами, сценариями и процедурами. Сводку дефектов, обнаруженных во время испытаний, целесообразно включать в отчетный доклад. В таком случае всем сторонам, заинтересованным в успехе разработки, не потребуется обращаться в базу данных дефектов за сведениями о ходе испытаний и о качестве программного продукта. Окончательная версия отчета по результатам анализа дефектов, в котором показаны все обнаруженные дефекты и их окончательные корректировки, а также редакция отчета о не устраненных дефектах и ошибках являются необходимыми дополнениями отчетного доклада о результатах испытаний.

Приемо-сдаточные испытания функциональной безопасности, как правило, применяют подмножество тестов, использованных на системном уровне. Наиболее полным и разносторонним испытаниям функциональной безопасности должна подвергаться первая версия программного продукта. При ***испытаниях очередных версий программного продукта*** возможны значительные сокращения объемов испытаний апробированных повторно используемых компонентов. Однако комплексные и завершающие испытания каждой новой версии программного продукта, как правило, должны проводиться в достаточном объеме, гарантирующем проверку выполнения ***всех требований технического задания***. Для выявления дефектов в процессе эксплуатации серийных образцов, в каждом из них должен быть предусмотрен некоторый минимум средств для оперативной проверки функционирования и обнаружения искажений результатов. Эти средства должны позволять фиксировать условия неправильной работы программ и характер проявления дефектов при применении программного продукта. Последующее исправление ошибок должно проводиться специалистами, осуществляющими сопровождение программного продукта.

Любые ***испытания ограничены допустимым количеством и объемом тестирования***, а также длительностью работы комиссии испытателей функциональной безопасности, поэтому ***не могут гарантировать абсолютную про-***

верку программного продукта, на соответствие требованиям к функциям и характеристикам. Для повышения достоверности определения и улучшения оценивания характеристик, после внутренних испытаний, некоторые экземпляры комплексов программ целесообразно передавать пользователям на ***опытную эксплуатацию в типовых условиях***. Это позволяет более глубоко оценить эксплуатационные характеристики созданного комплекса и оперативно устранять некоторые дефекты и ошибки. Опытную эксплуатацию целесообразно проводить разработчиками с участием испытателей-заказчиков и некоторых пользователей, назначаемых заказчиком.

Для заказчика и пользователей доминирующее значение могут иметь номенклатура и особенности реализации некоторых функций комплекса программ, которые, как правило, требуют наибольших затрат на тестирование и определяют основной эффект от применения программного продукта, а также потенциальный уровень спроса на рынке. Если затраты на разработку комплекса программ можно оценивать и прогнозировать с некоторой достоверностью, то эффективность применения и особенно будущий спрос на конкретную версию программного продукта со стороны пользователей априори оценить трудно. Такие оценки могут проводиться на основе специальных маркетинговых исследований и опыта эксплуатации аналогичных комплексов программ или достаточно близких их прототипов.

Представленные выше процессы испытаний компонентов и сложных комплексов программ ориентированы на ***наличие конкретного заказчика*** программного продукта и ограниченное число пользователей, контролируемых заказчиком. Несколько иначе организуются испытания коммерческих пакетов прикладных программ, создаваемых по инициативе предприятия или коллектива разработчиков для продажи широкому кругу пользователей при отсутствии конкретного заказчика. Для таких комплексов программ принято проводить ***квалификационные испытания на соответствие требованиям***, формализованным руководителем проекта в два последовательных этапа – ***Альфа и Бета тестирование*** (см. п.4.1).

4.3. Испытания производительности и динамического использования программными продуктами ресурсов компьютера

Цель испытания производительности – продемонстрировать заказчику, что система функционирует в соответствии с требованиями, содержащимися в спецификациях на производительность, и приемлемо время отклика при обработке заданного количества транзакций. (см. рис. 4.2). При этом предполагается, что в контракте, техническом задании и спецификации требований зафиксированы и утверждены требуемые значения этих характеристик. Оценивание динамических характеристик может проводиться при функционировании гото-

вого программного продукта или расчетными методами, при разработке, для оценки ***степени соответствия этим требованиям***.

При испытаниях производительности должны применяться промышленные нагрузки, что позволяет предсказать функциональную безопасность программного продукта и системы при реальной эксплуатации. Средства, обеспечивающее испытания отказов производительности, должны позволять оценивать влияние перегрузок. ***Оценивание перегрузок*** – это процесс испытаний работоспособности вычислительных машин при выполнении динамических сценариев большого потока данных с целью выяснения того, когда и где программный продукт выйдет из строя, работая под высокой нагрузкой. При тестировании перегрузок система подвергается предельным и максимальным нагрузкам для определения, выйдет ли она из строя и где это произойдет, а также для идентификации того, что выйдет из строя. Эти ***допустимые пределы должны быть определены в системных требованиях*** к программному продукту, где также должна определяться реакция функциональной безопасности системы на перегрузки.

Адекватное проведение динамического ***испытания программного продукта на перегрузки***, при использовании ручных методов подготовки тестов – дорого, трудоемко, неточно и занимает много времени. В распределенных системах требуется большое количество пользователей для осуществления внешней среды, обеспечивающей процесс динамического испытания. Выделение значительных ресурсов для тестирования требует больших затрат, и трудно организовать совместную работу необходимого числа пользователей и компьютеров. Увеличение количества средств испытания не позволяет создать тестирование перегрузок вручную, имитируя координированное взаимодействие многих пользователей с системой с ограниченным числом клиентских рабочих станций. Необходимы средства автоматизированного тестирования, которые включают имитаторы нагрузки и позволяют тестировщикам динамически имитировать сотни и тысячи виртуальных пользователей или объектов, одновременно работающих с функциональным программным продуктом.

Для ***измерения характеристик временной эффективности программного продукта*** необходимы инструментальные средства, встроенные в операционную систему или в соответствующий комплекс программ. Эти средства должны в динамике реального функционирования программного продукта регистрировать:

- загрузку вычислительной системы функционирующими программами;
- значения интенсивности потоков данных от конкретных внешних абонентов;
- длительность исполнения заданий при реализации конкретных функций;
- характеристики функционирования устройств ввода/вывода;
- время ожидания результатов (отклика) на функциональные задания пользователей или системы;

- заполнение памяти обмена с внешними абонентами в различных режимах применения программного продукта.

Значения этих характеристик зависят не только от свойств и функций комплекса программ, но также от особенностей архитектуры и операционной системы компьютера. Регулярная регистрация и обобщение таких данных позволяет выявлять ситуации, ***негативно влияющие на функциональную пригодность, надежность и другие важные характеристики программного продукта***. Потребность в ресурсах памяти и производительности компьютера в процессе динамического решения задач может значительно изменяться в зависимости от их свойств, а также от потока, состава и объема исходных данных. Степень использования памяти и производительности компьютера в некоторых пределах не влияет на качество решения функциональных задач комплексом программ. При излишне высокой интенсивности поступления исходных данных может ***нарушаться временной баланс*** между длительностью решения требуемой совокупности задач программным продуктом в реальном времени, и производительностью компьютера при решении этих задач. Также возможно нарушение баланса между имеющейся в компьютере памятью и памятью, необходимой для хранения всей поступившей и обрабатываемой информации. Для выявления подобных ситуаций и определения характеристик программного продукта в условиях недостаточности ресурсов компьютера проводятся испытания при высокой, но допустимой, в соответствие с требованиями, интенсивности поступления исходных данных.

Наиболее сложным является ***оценивание эффективности использования ресурсов производительности компьютера в реальном времени*** [17, 26]. При этом должна быть определена зависимость качества решения задач от интенсивности поступающей информации различных типов. Основная задача испытаний состоит в определении отказов – вероятностей, с которыми будет нарушаться соответствие между потребностями в производительности для решения всей требуемой совокупности задач и реальными возможностями компьютера и других компонентов системы. Если эта вероятность невелика, и можно считать допустимым эпизодическое снижение качества за счет получающихся задержек и пропусков в обработке сообщений или заданий, то делается вывод о соответствии производительности компьютера функциям данного программного продукта.

Для оценивания характеристик использования производительности при тестировании крупных программных продуктов ***должны быть измерены***:

- реальные значения интенсивностей поступающих исходных данных и заданий на вызов функциональных программ, а также распределения вероятностей этих интенсивностей для различных источников и типов заданий;
- длительности автономного решения отдельно каждой функциональной задачи, обрабатывающей исходные данные или включаемой внешними заданиями, а также периодически;

- загрузка компьютера в нормальном режиме поступления сообщений и заданий, а также вероятность перегрузки заданиями различных типов и возможные распределения длительностей перегрузки в реальных условиях;
- влияние пропуска в обработке заданий или сообщений каждого типа и снижения темпа решения определенных задач на функциональную пригодность и другие важные характеристики программного продукта.

Перечисленные задачи могут быть ***решены экспериментально*** в процессе тестирования завершенного разработкой программного продукта, однако при этом производительность компьютера может оказаться недостаточной для решения заданной совокупности задач в реальном времени, что негативно отразится на качестве использования системы. Кроме того, не всегда условия испытаний или опытной эксплуатации системы соответствуют режимам массового ее применения. Поэтому при оценивании требуется принимать специальные меры для создания реальных, а также контролируемых, наиболее тяжелых по загрузке условий функционирования комплекса программ и внешней среды. Такие критические ситуации могут быть в значительной степени предотвращены в процессе разработки комплекса программ путем расчета длительностей исполнения компонентов по тексту программ, и объединения этих характеристик в соответствии со структурой всего комплекса программ.

Для корректного ***оценивания предельной пропускной способности*** системы с данным программным продуктом необходимо измерять следующие характеристики реализации функциональных программ в процессе разработки:

- экстремальные значения длительностей их исполнения и маршруты, на которых эти значения достигаются;
- среднее значение длительности исполнения каждой функциональной группы программ на всем возможном множестве маршрутов и его дисперсию;
- распределение вероятностей и значений длительности исполнения функциональных групп программ.

Достоверность оценивания пропускной способности системы, с конкретным программным продуктом, зависит от корректности моделирования потоков внешних сообщений, а также от используемых распределений длительности исполнения программ. Для оценивания ресурсной эффективности, при подготовке технического задания и спецификаций требований ***следует согласовывать с заказчиком модель и характеристики внешней среды***, в которой будет применяться программный продукт, а также динамику приема и передачи данных.

Для определения использования комплексами программ временных ресурсов компьютеров полезно применять рекомендации стандарта **ISO 14756**. Стандарт ориентирован на динамическое оценивание: программных продуктов, операционных систем и вычислительных комплексов, включающих аппаратные и программные средства. Описание метода измерения производительно-

сти начинается с имитации пользователей и потоков данных из внешней среды: их случайных характеристик и процессов; функционирования терминалов; установления параметров рабочих нагрузок пользователей и вычислительных средств. Стандарт рекомендуется для использования: испытателями; разработчиками и покупателями программных продуктов; а также системными интеграторами сложных вычислительных систем реального времени.

По результатам испытаний могут быть решены ***задачи динамического оценивания ресурсной эффективности программного продукта***, что позволяет анализировать факторы, определяющие необходимую пропускную способность компьютеров, и разрабатывать меры для приведения ее в соответствие с потребностями. Если предварительно в процессе проектирования производительность системы не оценивалась или определялась слишком грубо, то, ***велик риск***, что доработки будут большими или может понадобиться заменить компьютер на более быстродействующий. Это обусловлено, как правило, «оптимизмом» разработчиков, что приводит к ***занижению интуитивных оценок длительностей решения функциональных задач*** и возможных предельных интенсивностей потоков внешней информации. Длительная регистрация и накопление значений ресурсной эффективности, способствуют выявлению ситуаций, при которых проявляются некоторые дефекты функциональной пригодности.

4.4. Испытания эксплуатации и модификации программной документации

Документация должна точно отражать назначение, функции, характеристики и требования, для безопасного использования и модификации программного продукта ***квалифицированными специалистами-пользователями***. Для этого эксплуатационные документы необходимо тестировать на полное соответствие всей ***совокупности требований на программный продукт***, согласованных между разработчиками и заказчиком. В результате эти документы можно использовать как отдельный, независимый при разработке ***третий эталон и вид тестов, для испытаний*** реализации требований к функциям и характеристикам программного продукта.

Практическое апробирование документов пользователями при применении комплекса программ, является практическим методом тестирования корректности реализации требований к программному продукту, завершающим полный цикл испытаний и контроля его качества (Рис.4.3). Разработчики документов должны обеспечивать комфортное и корректное применение комплекса программ пользователями, на основе ясного и непротиворечивого изложения в документах концепции, технологических процедур и операций для его штатного функционирования и получения ***требуемых результатов***. Организация документирования должна определять стратегию, стандарты, процедуры, распределение ресурсов и планы создания, изменения и применения документов на

комплекс программ. Для этого в общем случае должны быть выделены ***специалисты***, которые обязаны планировать, утверждать, выпускать, распространять и сопровождать комплекты апробированных и утвержденных эксплуатационных документов. Они должны стимулировать разработчиков программных комплексов, осуществлять непрерывное, регламентированное документирование процессов и результатов своей деятельности, а также контролировать полноту и качество утвержденных эксплуатационных и отчетных документов. Состав и содержание комплекта документов конкретного программного продукта, целесообразно адаптировать разработчиками к его особенностям и свойствам ***на основе использования стандартов и типовых структур*** – ***шаблонов*** для двух классов продуктов, которые в наибольшей степени различаются особенностями эксплуатации [17, 26].

Первый класс составляют программные продукты автоматизированного управления динамическими объектами и процессами в реальном масштабе времени. В процессе их применения допускается минимальное вмешательство пользователями в процедуры управления применением, и необходим, соответственно, небольшой объем эксплуатационных документов, выделяемых из стандартизированного комплекта. Для продуктов ***второго класса*** возможно применение пользователями широкого набора процедур управления, которые должны быть регламентированы достаточно полным набором и подробным содержанием документов. Пользователей таких программных продуктов можно разделить на две крупных группы, каждая из которых должна быть обеспечена ***комплектной эксплуатационной документацией***:

- администраторы, подготавливающие программный продукт к эксплуатации и обеспечивающие их функционирование и использование по прямому назначению;
- операторы – пользователи, реализующие применение программ-ных продуктов в системе, их функционирование, обработку и анализ результатов.

Документация администрирования при эксплуатации системы должна обеспечивать поддержку первичной инсталляции, безопасного функционирования и восстановления программ и данных после сбоев и отказов. Администратор системы и программного продукта должен быть информирован о всех изменениях функционирования устройств системы и внешней среды, могущих привести к сбою или возникновению аварийной ситуации, и предпринимать соответствующие профилактические действия.

Для этого требуется полная информация о требованиях к программному продукту, к компонентам системы и внешней среды, которые имеют свои особенности в управлении с помощью специальных программ, поддерживающих администрирование и управление системой. Каждый из документов административного управления ***должен не противоречить международным стандартам*** на коммуникацию, интерфейсы с пользователями и базами данных, на защиту и обеспечение безопасности информации.

Документация для оперативных пользователей программных продуктов в системе, их функционирования, обработки и анализа результатов должна обеспечивать взаимодействие пользователей с различными аппаратно-программными реализациями терминалов. Для этого необходима унификация концепции, архитектуры, функций и методов визуализации пользовательского интерфейса. Типовые ***формы-шаблоны документов*** и процедуры работы с ними, рассматриваемые ***как объекты стандартизации***, относятся в основном к функциональному, оперативному уровню взаимодействия пользователей с системами и программными продуктами.

Стандарты **ISO 15910:1999** и **ISO 18019:2004** являются наиболее полными нормативными документами, регламентирующими процессы ***создания эксплуатационной документации для пользователей крупных программных продуктов***.

Рис. 4.3.

В них изложены детальные структуры планов документирования, ориентированных на разработчиков документов, соответствующих требованиям на программный продукт, и процедуры контроля реализации плана (см. рис. 4.3).

Эксплуатационная документация должна проходить ***испытания на достоверность***, которые должны быть спланированы и реализованы на базе требований к функциям и характеристикам, а также к применению эксплуатационных процедур реального программного продукта. В стандартах описана координация документирования у субподрядчиков, а также конфигурационное управление изменениями и сопровождение эксплуатационных документов. Изложены требования и правила оформления ***твердой копии документов*** и правила структурирования и представления схем компонентов, окружения, иллюстраций и основного текста документов.

Стандарты представляют разработчикам метод определения и применения процесса документирования при создании конкретного программного продукта. Основной работой по стандартам является создание ***комплексного плана разработки документации***, реализация которого обеспечивает достоверное документирование программного продукта. Стандарты определяют реализацию процесса документирования, описанного в **ISO 12207:2008**, и могут быть ***адаптированы к условиям и требованиям конкретного проекта*** (см. рис. 4.3).

Описание требований эксплуатационной концепции для системы управления и программного продукта, должно содержать действия пользователя, необходимые для работы с системой, ее связи с существующими системами и процедурами. Его используют при создании соглашения между поставщиком, разработчиком и пользователями. Данный документ фиксирует текущее состояние системы и программного продукта, назначение, требования, возможности и ограничения в зависимости от режима или конкретного состояния эксплуатации и включает ***описания***:

- конкретной эксплуатационной среды и ее характеристики;
- основных компонентов и функций программного продукта, системы и связей между ними;
- внешних интерфейсов программного продукта и системы;
- возможностей, функций и характеристик программного продукта и системы;
- состава эксплуатационного персонала, его организационной структуры, уровня технической подготовки, обязанностей, взаимодействия;
- форм регистрации обнаруженных дефектов и ошибок;
- соглашений о внесении изменений, возникающих в процессе сопровождения версии программного продукта;
- концепцию поставки новой или модифицированной версии программного продукта, эксплуатационный сценарий;
- информацию о взаимодействии пользователей, поставщика, разработчика и предприятия, осуществляющих поддержку программного продукта, во время эксплуатационного периода.

В обязанности документаторов входит обеспечение ***контактов заказчика с разработчиками*** программного продукта, гарантирующее, понимание сути и требований к выпускаемой продукции и соответствующих ей аудиторий. Обя-

занностью разработчика является обеспечение полноты, корректности и актуальности всех документов, предъявляемых разработчиком на момент их поставки. Документатор должен предпринять соответствующие шаги по сохранению материалов, представленных заказчиком, обеспечить ***защиту информации о требованиях заказчика***. В ряде случаев требуется сохранить конфиденциальность и секретность предоставленных материалов. В договоре должны быть установлены уровни (грифы) конфиденциальности или секретности материалов, представляемых заказчиком документатору. План документирования должен включать ***определение аудитории пользователей документации***, уровня образования, квалификации, способностей, подготовки, опыта пользователей и другие характеристики, связанные с содержанием, структурой и использованием документации.

План документирования должен быть официально согласован и утвержден, что подтверждает полный учет и ***испытания в документах всех требований заказчика к программному продукту***. Должны быть определены процессы тестирования и проверки, выполняемые при разработке документации. План документирования должен быть подготовлен и утвержден до начала разработки документации, чтобы гарантировать согласование всеми сторонами содержания поставленных задач и используемых в проекте методов. После утверждения плана он должен быть доведен до всех заинтересованных сторон, включая разработчиков документации, а также заказчика и субподрядчиков.

Проверка результатов выполнения плана должна проводиться заказчиком с привлечением документаторов. Целью проверки является гарантирование полноты и достоверности, представленных материалов и удовлетворения ими возможности выполнения ***требований заказчика к функциям и характеристикам программного продукта*** в соответствии с условиями договора и плана документирования. Проверка должна проводиться персоналом заказчика, обладающим соответствующей квалификацией и полномочиями на изменения в документах и их утверждению. Непосредственно перед представлением любой публикации на утверждение, в ней должны быть обновлены все распечатки экранов для гарантирования ее актуальности.

При ***контроле изменений и сопровождение документации*** должны быть предусмотрены:

- функциональные изменения данной версии – изменения требований и функций программного продукта, внесенные при разработке и тестировании документации и отраженные в опубликованной документации;
- функциональные изменения для последующей версии – изменения требований к программному продукту, внесенные при разработке документации и не отраженные в опубликованной документации, но подлежащие учету в последующей версии;

- изменения документа после публикации – изменения в опубликованной документации, обусловленные изменениями комплекса программ или обнаружением дефектов и ошибок в данной документации.

Документатор должен обеспечить разработку документации так, чтобы допустить возможность внесения в нее корректировок всех типов. Для этого необходимо, чтобы разработчики документации получали учтенные копии ***изменений требований к программному продукту***, подтверждающие внесение соответствующих изменений в данную версию после конкретной даты ее пересмотра.

Испытания эксплуатационной документации на практичность следует рассматривать как дополнение к проводимым проверкам и анализам документации. При необходимости в плане документирования следует определить внешнюю среду тестирования, которая должна полностью соответствовать ***эксплуатационной среде конечного пользователя и требованиям к программному продукту***. При проведении тестирования документации на практичность, необходимо проверить соответствие и актуальность документов конкретному программному продукту. При составлении графика испытаний эксплуатационных документов необходимо учитывать тестирование отдельных компонентов и выполняемых ими функций. При проведении тестирования после завершения разработки следует использовать выпускаемую версию программного продукта.

При тестировании может быть использовано для оценки практичности определения характеристик качества в стандартах **ISO 9126:1-4** и **ISO 12119**. ***Практичность – применимость***: свойства программного продукта и документации, отражающие сложность его понимания, изучения и использования ***для квалифицированных пользователей***. Требования к практичности и ее характеристикам – понятности и простоте использования, зависят от назначения и функций продукта и могут формализоваться заказчиками набором свойств, необходимых для обеспечения удобной и комфортной эксплуатации. Для обеспечения полноценного изучения процессов применения программного продукта специалистами, необходима эксплуатационная документация, ***объем*** которой существенно зависит от назначения и функций, и может быть задан на основе анализа прецедентов подобных успешных проектов. Для некоторых проектов, подлежащих широкому тиражированию, могут быть желательны адекватные по содержанию электронные учебники, требования к объему и функциям которых целесообразно оценивать также по прецедентам. Следует учитывать, что малый объем эксплуатационной документации может снизить качество и полноту использования функций сложного продукта, а очень большой объем – также может ухудшить эксплуатацию из-за трудности выделения из множества второстепенных деталей и освоения, наиболее существенных свойств и особенностей его применения.

Простота использования: возможность пользователю удобно и комфортно эксплуатировать и управлять программным продуктом. Аспекты изменяемости, адаптируемости и легкости инсталляции могут быть предпосылками для простоты использования и выбора конкретного продукта. Она соответствует управляемости, устойчивости к ошибкам пользователей и согласованности с их ожиданиями и навыками. Эта характеристика должна учитывать физические и психологические особенности пользователей и отражать уровень комфортности условий эксплуатации ПП, возможность предотвращения ошибок пользователей.

Оценки ***практичности*** зависят не только от собственных характеристик продуктов, но также от организации и адекватности документирования процессов их эксплуатации. При этом предполагается, что в контракте, техническом задании или спецификации зафиксированы и утверждены требования к основным параметрам и качеству методов и средств документирования и поддержки использования продукта. Эти требования ***могут влиять на функциональную пригодность и успех внедрения комплекса программ у пользователей***, а также значительно различаться в зависимости от функционального назначения и сферы применения программного продукта. Затраты на разработку комплекта ***эксплуатационной документации*** для сложных программных продуктов, подлежащих длительному сопровождению, обычно определяются затратами на объем текста документов ***ориентировочно*** 5 – 10 страниц на тысячу строк программы.

В ***технологической документации***, обеспечивающей конфигурационное управление и многолетнее сопровождение версий программного продукта, необходимы требования, тесты, тексты программ и описания алгоритмов. Это приводит к увеличению объема документации на порядок, т. е. может составлять около 100 страниц совокупности документов на тысячу строк программы. Статистический анализ объема документации для программных продуктов различных классов дает широкий разброс числа страниц на единицу объема программ [12, 16, 26]. Однако выявились некоторые средние характеристики, которые близки к указанным оценкам. Подтверждена наиболее высокая документированность тиражиру-емых программ реального времени, особенно в тех случаях, когда необходима высокая безопасность и надежность функционирования продукта (до 200 страниц на тысячу строк).

При оценках можно предполагать средний объем ***технологической документации*** ~ 50 – 100 страниц на тысячу строк. При этом затраты на документацию остаются практически пропорциональными размеру комплекса программ. Эта часть документации не подлежит массовому тиражированию и поставке каждому пользователю, что позволяет несколько снижать удельные затраты на ее подготовку. Однако необходимость ее тщательной отработки, и высокого соответствия текущей версии программного продукта, приводит к большим затратам, как при первичном изготовлении технологических

документов, так и при их модификации в процессе последующего сопровождения.

Оценивание качества и соответствия требованиям документов программного продукта при приемо-сдаточных испытаниях проводится комиссией заказчика, в соответствии с Программой и методиками испытаний ПП. В них участвуют руководитель (главный менеджер) разработки и некоторые ведущие разработчики. ***Комиссия при испытаниях должна руководствоваться следующими документами*:**

- 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 испытаний документов по всем требованиям контракта, технического задания и спецификаций;
- методиками испытаний и набором тестов, охватывающими каждый раздел требований технического задания, спецификаций и Программы испытаний;
- комплектом адекватной эксплуатационной и технологической документации на программный продукт.

Результаты испытаний фиксируются в протоколах, которые обычно содержат следующие разделы:

- условия и сценарии проведения тестирования и характеристики исходных данных при динамическом тестировании;
- обобщенные результаты испытаний с оценкой их на соответствие требованиям технического задания и другим руководящим документам, а также технической и эксплуатационной документации;
- описание обнаруженных дефектов и ошибок и рекомендуемых улучшений в испытываемом комплексе программ;
- выводы о результатах испытаний и о соответствии созданного программного комплекса или его функционального компонента определенному разделу требований технического задания и исходных спецификаций.

Протоколы по всей программе испытаний ***обобщаются в акте,*** в результате чего делается ***заключение о соответствии системы требованиям*** заказчика и о завершении работы с положительным или отрицательным итогом. При выполнении всех требований технического задания заказчик обязан принять программный продукт, и проект считается завершенным.

Завершение испытаний документов программного продукта в составе системы должно удостоверить и зафиксировать следующие ***процессы и результаты***:

- подготовлена полная версия программного комплекса с контролируемыми изменениями после испытаний;
- проведены совещания менеджеров и специалистов, посвященные управлению незавершенными работами по устранению дефектов и оценки времени для исправления дефектов;
- в предшествующие несколько недель не произошло ни одного сбоя, остановки, неожиданного прекращения процесса или другой аномалии функционирования комплекса программ на объектном компьютере в системе;
- анализ динамического функционирования показал, что комплекс программ достиг приемлемого уровня качества, стабильности, надежности и безопасности;
- менеджеры и группа управления системой установила, что программный продукт, как это определено в ходе заключительного цикла испытаний, удовлетворяет требованиям заказчика и пользователей;
- проведено совещание менеджеров производства с заказчиком и установлено, что ***критерии завершения испытаний программного продукта выполнены***.

Завершаются испытания предъявлением заказчику на утверждение ***комплекта документов***, содержащих ***результаты комплексных испытаний*** версии программного продукта:

- откорректированные тексты программ и данных на языке программирования, а также полные спецификации требований на программные компоненты и комплекс в целом после полного завершения тестирования и испытаний;
- отчет о подтверждении заданного качества, полные характеристики достигнутого качества функционирования, а также степени покрытия тестами спецификации требований к программному продукту;
- технические условия на версию программного продукта, базу данных управления конфигурацией и эксплуатационную 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;
- отчет о технико-экономических показателях завершенного проекта версии программного продукта, выполнении планов и использованных ресурсах;

- ***акт о завершении испытаний и готовности к поставке*** и/или предъявлению для сертификационных испытаний версии программного продукта.

После завершения испытаний новой версии программного продукта, обычно осуществляется процесс ее внедрение для применения пользователями. Это производится, как правило, в два этапа; силами разработчиков версии в целях обкатки, проверки и выявления ошибок в изменениях на этапе опытной эксплуатации, и 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.

При сопоставлении результатов оценивания функций и характеристик качества с требованиями технического задания и спецификаций, разработчик или поставщик ***обязан удовлетворять требования заказчика только в пределах согласованных параметров*** контракта и модели внешней среды и системы. Оценивание функционирования комплекса программ за этими пределам должно дополнительно согласовываться испытателями с заказчиком и производителями. При этом не выполнение требований может квалифицироваться как их расширение за пределы, ограниченные контрактом, и не учитываться при оценивании заказчиком характеристик качества программного продукта, или как дополнительные работы, подлежащие соответствующему финансированию со стороны заказчика, для доработки комплекса программ с целью удовлетворения этих требований.

Глава 5. Удостоверение надежности и безопасности применения программного продукта реального времени

5.1.. Средства автоматизации для испытаний надежности и функциональной безопасности программных продуктов реального времени

Ручная пошаговая подготовка сценариев и содержания тестов не может обеспечить достаточно представительный их набор, для испытаний сложных комплексов программ реального времени на соответствие требованиям к функциям и характеристикам качества. Для обеспечения высокого качества таких комплексов часто ориентируются на ***Альфа- и Бета-тестирование коллективами потенциальных пользователей или на тестирование при опытной эксплуатации***. Однако эти методы приводят к длительным процессам устранения ошибок и дефектов, сохраняется неопределенность достигнутого качества и возможных рисков. Кроме того, такое тестирование может быть недопустимо для программных продуктов критических систем, где проявление случайных ошибок при эксплуатации может нанести большой ущерб.

Как правило, разработчиками моделей автоматизированной генерации тестов недооценивается сложность их разработки и пренебрегается необходимость детализации к ним требований. ***Требования и реализация совокупности тестов должны полностью отражать возможность проверки выполнения требований к комплексу программ***, и соответственно должны быть принципиально аналогичными по сложности и трудоемкости, разработке программного продукта. Эти ***два представления комплексов программ*** отличаются только формой описания их содержания: функциональным (процессным) или событийным (сценариями и результатами исполнения). Генерация тестов особенно сложна (также как и требований к программному продукту) для крупных комплексов программ реального времени. Выбор типов моделей генерации тестов зависит от глубины знаний об алгоритмах функционирования программ, характеристиках их качества и обобщенных параметрах результатов работы системы в целом. Кроме того, существенным для выбора типов моделей для генерации тестов может быть длительность расчета имитированных данных и обеспечение возможности проводить полную обработку результатов испытаний.

При наличии реальных планов и разумных предположений использование автоматизированных инструментальных средств и автоматизированных тестовых сценариев представляет собой ***способ снижения временных затрат на испытания программного продукта***. Эффективная Программа испытаний имеет свой собственный ***жизненный цикл разработки***, в который входят планирование стратегий и целей, определение требований к тестам, анализ, проек-

тирование и кодирование. По аналогии с процессом, которому следуют при разработке комплекса программ, требования к тестам необходимо определить прежде, чем тесты будут спроектированы. Требования к тестам должны быть ясно сформулированы и документированы, чтобы все участники проекта понимали, на чем основаны работы по испытаниям.

Обычно на автоматизацию задачи уходит намного больше времени, чем на ее выполнение, поэтому для каждой задачи, которая может быть автоматизирована, целесообразно проводить тщательный ***анализ потенциального выигрыша от автоматизации***. Эффективная автоматизация испытаний требует специальной подготовки персонала, разработки, отладки и верификации, как и любой другой проект разработки программного продукта. Беспланова и плохо выполненная автоматизация означает не только напрасный расход ресурсов, она даже может привести к нарушению графика выполняемого проекта, если время будет тратиться на отладку средств их автоматизации, а не на испытания.

Автоматизация испытаний обычно выражается в способности системы тестов формироваться и выполняться без участия или при частичном участии человека [2, 7, 8, 17]. Процессы могут заключаться в автоматизированной работе с системой, предназначенной для испытаний, в загрузке данных для нее и в сравнении полученного результата с ожидаемым. Автоматизация испытаний должна позволять испытателю спроектировать и разработать полный комплект тестовых сценариев, а затем с небольшими затратами или вовсе без них ***повторять сценарии испытаний***.

Затраты, необходимые для автоматизации генерации тестов, могут быть не равны затратам на ручное написание тестов. Для создания автоматизированных тестов, удобных для развития и сопровождения ПК, должна быть выстроена инфраструктура, позволяющая испытателю определять действия, данные и ожидаемые результаты в удобном формате. Помимо затрат на создание испытательной инфраструктуры, часто серьезные вложения приходится делать в сами инструменты, которые могут стоить сотни тысяч долларов. Наконец, автоматизация тестирования предполагает внесение изменений в процесс испытаний и освоенные навыки, необходимые группе испытателей, причем все это должны быть управляемо.

Если все-таки решено вложить средства в автоматизацию испытаний, следует проанализировать последствия этого решения в рамках выбранной стратегии. Поскольку для задач, решаемых в ручном режиме, и для задач, решаемых в автоматическом режиме, необходимо приобретать одни и те же аппаратные средства, соединять их между собой, подключать к компьютерным сетям и вводить их в эксплуатацию, может, имеет смысл сделать ***автоматизированную динамическую тестовую среду постоянно действующей конфигурацией***. Если предполагается исполнять автоматизированные регрессионные испытания или для целей технического обслуживания системы, может иметь смысл построить специализированный ***испытательный стенд***, который будет находиться на рабочей площадке испытателей весь период, пока поддерживается и

развивается программный продукт. Однако такое решение влечет за собой существенное увеличение расходов на установку соответствующих аппаратных средств и на испытательный стенд.

Группа испытателей может при необходимости создавать специальные инструменты – ***генераторы динамических тестов***, которые на основе некоторого набора правил автоматически генерируют тесты. Эти правила можно получить из спецификаций требований к программному продукту, из документации базы данных проекта; испытатели могут разработать их вручную, в соответствие с требованиями к испытаниям. При необходимости генераторы могут динамически создавать тестовые данные, например, для проведения нагрузочного тестирования. Некоторые генераторы тестовых процедур, могут обладать высокой степенью интеграции в процессе анализа и проектирования программных продуктов и позволять разработчикам тестировать функции в соответствии со спецификациями требований системной архитектуры.

В результате испытаний программных продуктов необходимо достоверно устанавливать ***степень их соответствия утвержденным требованиям*** к функциям, надежности и безопасности. Для этого они должны проходить тщательные динамические испытания в условиях их последующего применения. В реальных системах создание таких условий может быть очень дорого и опасно при проявлении не устраненных при тестировании дефектов и ошибок, что недопустимо. Альтернативой является создание и применение ***математических моделей на компьютерах***, ***динамически имитирующих реальную внешнюю среду*** для генерации тестов и функционирования испытываемых комплексов программ. Таким образом, формируется возможность выполнять испытания, выявлять и устранять ошибки, а также ***достигать высокого уровня соответствия программного продукта заданным требованиям***.

Для обеспечения функциональной безопасности и высокого качества систем и комплексов программ необходимы соответствующие ***проблемно-ориентированные интегрированные системы имитации внешней среды и автоматизации обработки результатов испытаний***, способные достаточно полно заменить или сократить испытания программ с реальными объектами внешней среды. При этом высокая стоимость испытаний с реальными объектами почти всегда оправдывают значительные затраты на такие интегрированные системы, если предстоят испытания критических ПП с высокими требованиями к безопасности функционирования программ, с длительным жизненным циклом и множеством развивающихся версий. ***Инструментальные средства автоматизации процессов испытаний программ***, должны обеспечивать [7, 17, 26]:

- определение тестов – реализацию процесса тестирования разработчиком: генерацию тестовых данных; ввод тестовых наборов; ввод ожидаемых, эталонных результатов;
- управление тестами и участком программы, для которого средство испытаний может автоматически выполнять тестовые наборы;

- анализ и обработку тестовых результатов – возможность средства испытаний автоматически анализировать тестовые результаты: сравнение ожидаемых и реальных результатов; сравнение файлов; статистическую обработку результатов;
- анализ покрытия тестами исходного кода программ для обнаружения: операторов, которые не были выполнены; процедур, которые не были вызваны; переменных, к которым не были обращения;
- анализ и испытания производительности программы, когда она исполняется: загрузку центрального процессора; загрузку памяти; обращения к специфицированным элементам данных и/или к сегментам кода; временные характеристики функционирования испытываемой программы;
- моделирование внешней среды – поддержку процесса испытаний с помощью модели имитации данных из внешних для ПП, компонентов системы.

В отличие от натурного эксперимента моделирование внешней среды и тестов на компьютере имеет большие возможности контроля**,** как исходных данных, так и всех промежуточных и выходных результатов функционирования испытываемого объекта. В реальных системах ряд компонентов иногда оказывается недоступным для контроля их состояния, так как, либо невозможно поместить измерители контролируемых сигналов в реальные подсистемы, подлежащие испытаниям, либо это сопряжено с изменением характеристик самого анализируемого объекта. Преимуществом моделирования внешней среды на компьютере является также повторяемость результатов функционирования и возможность относительно дешевого исследования большого количества вариантов и сценариев испытаний. В отличие от этого натурные эксперименты зачастую невозможно остановить на некоторой промежуточной фазе или повторить с абсолютно теми же исходными данными. ***Программная имитация внешней среды на компьютере позволяет***:

- проводить длительное генерирование имитируемых данных, для определения характеристик функциональной безопасности и качества ПП, в широком диапазоне изменения условий и параметров, что зачастую сложно или невозможно при использовании реальных объектов;
- расширять диапазоны характеристик имитируемых объектов за пределы, реально существующих или доступных источников данных, а также генерировать потоки информации, отражающие перспек-тивные характеристики создаваемых систем и компонентов внешней среды;
- создавать тестовые данные, соответствующие критическим или опасным ситуациям функционирования объектов внешней среды, которые невозможно или рискованно реализовать при натурных экспериментах;

- обеспечивать высокую повторяемость имитируемых данных при заданных условиях их генерации и возможность прекращения или приостановки имитации на любых фазах моделирования внешней среды.

Методы ***динамического тестирования с исполнением контролируемого комплекса программ***, в большей или меньшей степени, ориентированы на обнаружение ошибок определенных типов, преимущественно в структуре комплекса программ и реализуемых маршрутах обработки информации. Методы тестирования потоков данных ориентируются на выявление ошибок в вычислительной части программ и в процессах преобразования различной информации. Такая ориентация позволяет упорядочивать последовательность применения методов с целью устранения, прежде всего, ошибок в наибольшей степени отражающихся на корректности исполнения программ, а также сосредоточиваться на методах, позволяющих решать частные задачи достижения необходимого их качества и соответствия требованиям при минимальных затратах.

Характеристики динамического функционирования программных продуктов зависят не только от их внутренних свойств, но и ***от свойств внешней среды***, в которой они применяются (см.стандарт **ISO 12119**). Для сокращения неопределенностей и прямых ошибок при оценивании качества программ необходимо до начала испытаний определить основные параметры внешней среды и потоки информации, при которых должен функционировать комплекс программ с требуемыми характеристиками при оценивании его качества и эксплуатации. Для этого заказчик и разработчик совместно должны структурировать, описать и согласовать ***модель внешней среды*** и ее параметры в среднем, типовом режиме применения, а также в наиболее вероятных и критических режимах, в которых должны обеспечиваться требуемые характеристики качества динамического функционирования программного продукта.

При создании генераторов динамических тестов внешней среды применяется два принципиально различающихся подхода, которые условно можно назвать ***интегральным и дифференциальным***. При ***интегральном*** или эмпирико-статистическом подходе основой является формальное описание входной и выходной информации имитируемого динамического объекта, а также функциональной связи между данными на его входе и выходе. При этом структура объекта и процессы, реализующиеся при реальном функционировании его компонентов, не имеют значения и не моделируются. Исходные данные и характеристики для построения таких генераторов тестов получаются в натурных экспериментах или при исследовании более детальных – дифференциальных моделей.

Дифференциальные или имитационные модели генераторов динамических тестов базируются на описаниях внутренних процессов функционирования компонентов объекта моделирования, его структуры и взаимодействия составляющих. Результаты функционирования таких моделей определяются адекват-

ностью знаний о компонентах и их характеристиках, а также об их взаимосвязях. Для этого необходимы достаточно подробные сведения о всех процессах функционирования компонентов объектов внешней среды, которые в свою очередь могут потребовать еще более глубокого моделирования их составляющих.

В реальных системах ряд компонентов иногда оказывается недоступным для контроля их состояния, так как, либо невозможно поместить измерители контролируемых сигналов в реальные подсистемы, подлежащие тестированию, либо это сопряжено с изменением характеристик самого анализируемого объекта. Преимуществом моделирования внешней среды на компьютерах является также повторяемость результатов функционирования и возможность исследования большого количества вариантов и сценариев тестирования. В отличие от этого натурные эксперименты зачастую невозможно остановить на некоторой промежуточной фазе или повторить с абсолютно теми же исходными данными.

Одними из наиболее сложных и дорогих имитаторов внешней среды, применяемых для испытаний крупных программных продуктов, являются модели: полета космических аппаратов; диспетчерских пунктов управления воздушным движением; объектов систем противовоздушной обороны; сложных административных систем и другие объекты [2, 7, 17]. Применяемые для этого ***моделирующие испытательные стенды (МИС)*** проблемно – ориентированы и размеры программ, моделирующих в них динамическую внешнюю среду, могут даже значительно превышать размеры соответствующих испытываемых программных продуктов. Для их реализации должны выделяться достаточно мощные универсальные ***моделирующие компьютеры***. Кроме того, для автоматизации разработки программных продуктов могут использоваться отдельные ***технологические компьютеры***, что в совокупности образует инструментальную базу для обеспечения всего ЖЦ сложных комплексов программ реального времени на ***объектных реализующих компьютерах***.

Имитация конкретных динамических тестов с реальными характеристиками, адекватными объектам внешней среды, является основной частью типовых моделирующих стендов. В соответствии с полной номенклатурой и реальными характеристиками таких объектов создаются их интегральные или дифференциальные модели. Выбор типов моделей зависит от глубины знаний об алгоритмах функционирования внешних объектов, характеристиках их компонентов и обобщенных параметрах работы объекта в целом. Кроме того, существенным для выбора типов моделей является длительность расчета имитированных данных и обеспечение возможности проводить полную имитацию динамической внешней среды на моделирующем ***компьютере в реальном времени*** с учетом затрат времени на обработку результатов.

Трудность адекватного моделирования некоторых динамических объектов внешней среды, особенно если при их функционировании активно участвует оператор-пользователь, не позволяет сосредоточить и полностью автоматизировать для крупных программных продуктов всю имитацию тестовых данных на ***моделирующих компьютерах***. Поэтому при реализации интегрированных

проблемно-ориентированных ***МИС*** для испытаний качества функционирования сложных программных продуктов приходится использовать аналоги реальных объектов внешней среды для формирования части данных. Разумное ***сочетание части реальных объектов внешней среды и имитаторов на компьютерах*** обеспечивает создание высокоэффективных МИС с комплексными моделями совокупностей объектов внешней среды, необходимых для динамических испытаний качества программных продуктов и систем в реальном времени. Такие стенды позволяют автоматическую генерацию тестов с помощью имитаторов на компьютерах и аналогов реальной аппаратуры дополнять реальными данными от операторов пользователей, контролирующих и корректирующих функционирование систем обработки информации.

В схеме типового МИС можно выделить ***ряд базовых компонентов***. Для каждого эксперимента при динамических испытаниях комплексов программ реального времени следует подготавливать план сценариев тестирования и ***обобщенные исходные данные.*** В ***моделирующем компьютере*** план и обобщенные исходные данные преобразуются в конкретные значения параметров для задания динамического функционирования каждого имитатора или реального объекта внешней среды. Эти данные вводятся и преобразуются на моделирующем компьютере вне реального масштаба времени и подготавливают старт сеанса функционирования стенда и испытываемых программ в реальном времени. После этого начинают генерироваться тестовые данные.

Данные с рабочих мест операторов-пользователей должны отражать реальные характеристики динамических воздействий на тестируемый программный продукт с учетом особенностей и квалификации человека, которому предстоит использовать испытываемые программы в реальной системе обработки информации. На эту часть МИС кроме первичных исходных данных от моделирующего компьютера могут вводиться данные обработки ряда тестов испытываемой системой. В результате через человека и его характеристики замыкается контур обратной связи ручного и автоматизированного управления динамическими объектами внешней среды. Такое же замыкание контура автоматизированного управления возможно в аналогах и аппаратных имитаторах реальных объектов.

Данные натурных экспериментов с объектами внешней среды могут подготавливаться заранее вне сеансов динамических испытаний программного продукта, например, при отладке аппаратной части системы обработки информации. Эти данные отражают характеристики и динамику функционирования объектов, которые трудно или опасно подключать для непосредственного взаимодействия с недостаточно проверенными программами. Кроме того, такие данные могут использоваться для аттестации адекватности имитаторов некоторых объектов внешней среды. Они могут быть полезны в тех случаях, когда создание определенных условий динамического функционирования объектов внешней среды очень дорого или опасно и может быть выполнено только в исключительных случаях. Однако данные натурных экспериментов не всегда удается адекватно описать условиями и обобщенными характеристика-

ми их поведения. Наличие ряда случайных неконтролируемых факторов усложняет картину исходных данных и может затруднять сопоставление результатов натурных экспериментов с полученными, при программной имитации тестов.

При динамическом тестировании в ряде случаев необходимо иметь ***эталонные характеристики данных***, поступающих на испытываемый программный продукт или систему. При работе с реальными объектами зачастую приходится создавать специальные измерительные комплексы, которые определяют, регистрируют и подготавливают к обработке все необходимые характеристики в процессе реального функционирования этих объектов (например, координаты движения самолетов при испытаниях систем УВД). Такие измерения проводятся при автономном функционировании внешних объектов или при их взаимодействии с комплексом программ в реальном времени. Имитация эталонных характеристик объектов внешней среды может служить для определения надежности и безопасности функционирования программного продукта в идеальных условиях – при отсутствии искажений исходных данных, ошибок в измерениях их параметров, сбоев и предумышленных отказов аппаратуры. Проверка при таких исходных данных позволяет оценить характеристики дефектов и ошибок результатов, обусловленные недостаточным качеством комплекса программ.

Синхронизация и обобщение тестовых данных предназначены для упорядочения динамических тестов от источников различных типов, в соответствии с реальным временем их поступления на комплекс программ, и для распределения между моделирующим и объектным компьютером. В результате формируются потоки тестовых данных каждого реального объекта внешней среды, которые вводятся в объектный компьютер в соответствии с логикой функционирования системы обработки информации через соответствующие устройства сопряжения с моделирующим компьютером.

Повторяемость сеансов испытаний при автоматической имитации динамических тестов обеспечивается фиксированием всех исходных данных и применением программного формирования псевдослучайных чисел. При надежной работе аналогов реальных объектов и моделирующего компьютера, в принципе, можно добиться почти ***абсолютной повторяемости весьма длительных экспериментов и сценариев испытаний***. Некоторая не идентичность результатов при повторных экспериментах может быть обусловлена сбоями и частичными отказами аппаратуры. Труднее обеспечивать повторяемость сценариев динамических испытаний, в которых активно участвует оператор-пользователь. В этом случае необходимо регистрировать и сохранять действия оператора в зависимости от времени, а затем повторять их в соответствии с записанным сценарием. При необходимости временная диаграмма может соблюдаться с точностью около 0,5-1 с, однако ошибки в действиях оператора и вводимых им параметрах могут отличаться в каждом сценарии тестирования. Вследствие этого повторяемость тестов реализуется только статистически.

Приведенные выше рекомендации по функциям и применению МИС ориентированы на создание крупных программных продуктов, их динамическое тестирование и испытания, в основном, до передачи в регулярную эксплуатацию. После приемки заказчиком или приобретения пользователями, в процессе функционирования и применения программного продукта должно обеспечиваться их регулярное тестирование и оценка текущего качества. Для этого в ***составе программного продукта необходимы средства, обеспечивающие:***

- генерацию динамических тестовых сценариев или хранения тестов для контроля работоспособности, сохранности и целостности программного продукта при функционировании и применении;
- оперативный контроль и обнаружение дефектов исполнения программ и обработки данных при использовании программного продукта по прямому назначению;
- реализацию процедур предварительного анализа выявленных дефектов и оперативное восстановление вычислительного процесса, программ и данных (рестарт) после обнаружения аномалий динамического функционирования программного продукта;
- мониторинг, накопление и хранение данных о выявленных дефектах, сбоях и отказах в процессе исполнения комплекса программ и обработки данных.

Средства генерации динамических тестов и имитации внешней среды в составе поставляемого программного продукта предназначены для оперативной подготовки исходных данных при проверке различных режимов функционирования в процессе применения программного продукта и при диагностике проявившихся дефектов. Минимальный состав средств генерации тестов должен передаваться пользователям для контроля использования рабочих версий в реальном времени и входить в комплект поставки каждой пользовательской версии программного продукта. Для размещения таких средств мониторинга и контроля качества функционирования крупных комплексов программ необходимы ресурсы памяти, а также дополнительная производительность компьютера. Более глубокие испытания функционирования версий и локализации ошибок следует проводить на базе комплекса средств имитации внешней среды высшего уровня (МИС) на моделирующем компьютере, которые используются специалистами для испытаний и/или сертификации системы. Часть этих средств имитации может применяться как средства нижнего уровня (пользовательские) на объектном компьютере для диагностики и обеспечения полного повторения ситуаций, при которых пользователями могут быть обнаружены динамические дефекты функционирования ПП.

Важной функцией испытательных стендов является их использование в качестве ***тренажеров для операторов-пользователей***. Так как качество функционирования комплексов программ может существенно зависеть от характеристик конкретного человека, участвующего в эксплуатации и обработке информации, то необходимо измерять эти характеристики. Необходимо также

иметь возможность их улучшать до уровня, обеспечивающего выполнение ***заданных требований к программному продукту***. Поэтому в процесс испытаний органически должен входить процесс динамической тренировки и измерения характеристик реального функционирования и реакции операторов-пользователей, а также использование МИС для обучения и регулярной подготовки операторов-пользователей в процессе тиражирования и эксплуатации программного продукта.

При использовании программных моделей на компьютере ***достоверность генерации динамических тестов*** определяется следующими факторами:

- адекватностью имитатора моделируемому объекту внешней среды или источнику информации;
- инструментальной точностью средств, реализующих имитатор внешней среды;
- статистической точностью процесса имитации и объемом динамических тестовых данных, учитываемых при статистическом обобщении результатов испытаний;
- точностью дискретизации имитаторами реальных непрерывных процессов в моделируемых объектах внешней среды.

Важнейшее значение для определения качества программных продуктов имеет ***адекватность имитаторов динамических тестов***, которая зависит от степени учета второстепенных факторов, характеризующих функционирование реальных объектов или источников информации, при создании их моделей. Точность моделей на компьютере, прежде всего, определяется алгоритмами, на которых они базируются, и полнотой учета в них всех особенностей моделируемых объектов. Кроме того, на адекватность имитации влияет уровень дефектов и ошибок в программах имитации. Каждый, не учитываемый в имитаторе элемент или фактор моделируемой системы, необходимо оценивать путем сопоставления частных имитируемых данных с результатами аналитических исследований или с данными, полученными на реальных системах, и определять его возможное влияние на полную требуемую точность модели и генерируемых динамических тестов с учетом других составляющих, отражающихся на достоверности имитации. Перечисленные факторы, влияющие на достоверность генерации тестов в МИС, взаимозависимы, и повышение достоверности имитации за счет одного из факторов при ограниченных ресурсах приводит, как правило, к снижению достоверности вследствие влияния остальных. Поэтому важной задачей при создании имитационных моделей является достижение наибольшей суммарной достоверности имитации и определения значений характеристик качества функционирования программного продукта, при сбалансированном влиянии каждого из факторов. Поэтому при создании сложных генераторов тестов необходимо достигать, по возможности, равное влияние отмеченных факторов на суммарную достоверность оценки качества программного продукта при испытаниях.

Регистрация и обработка характеристик динамических тестовых данных должна обеспечивать их контроль на соответствие заданным требованиям к программному продукту, обобщенным характеристикам каждого объекта внешней среды и исходным данным сеанса испытаний. Так проводится процесс испытаний по конечным результатам функционирования комплекса программ, выдаваемым внешним абонентам и для определения интегральных характеристик качества и их соответствия требованиям к программному продукту. Однако для диагностики и локализации отказов, дефектов и ошибок, а также для оценки некоторых частных характеристик качества могут быть необходимы промежуточные данные исполнения программ. Для регистрации промежуточных данных следует иметь возможность разорвать процесс естественного исполнения комплекса программ на любом заданном компоненте, операторе или при обращении на запись или чтение к конкретных данных ***объектного компьютера***.

Селекция результатов динамических испытаний может основываться на стратегии контроля функционирования программного продукта ***снизу вверх***, т.е. от анализа исполнения отдельных компонентов программы и далее до стохастических результатов функционирования всего комплекса в динамике реального времени. При этом регистрируется избыточное количество данных, из которых затем отбирается минимум, необходимый для анализа. Может использоваться стратегия ***сверху вниз***, т.е. упорядоченное, иерархическое выделение в первую очередь обобщенных результатов функционирования комплекса программ с последующим уточнением регистрируемых и анализируемых результатов вплоть до детального контроля исполнения отмеченных программных компонентов. В этом случае регистрируются только те данные, которые необходимы для анализа в конкретном сеансе динамического испытания. При обеих стратегиях необходимо иметь возможность управлять объемом и видом выделяемой и регистрируемой информации тестирования в зависимости от целей испытаний. Данные, получаемые и выделяемые в процессе испытаний качества ПК, целесообразно делить на следующие ***группы***:

- данные, характеризующие исходную тестовую информацию и выходные результаты динамического тестирования;
- маршруты исполнения программных компонентов и их операторов при некоторых фиксированных тестовых данных;
- аномальные события, сбои, отказы и данные, характеризующиеся отклонением результатов динамического тестирования от требований за допустимые пределы и ограничения;
- характеристики динамического использования различных ресурсов объектного компьютера.

Регистрация промежуточных данных обычно соответствует некоторым достаточно завершенным этапам испытаний функционирования программного продукта. Вызовы регистрирующих программ должны подчиняться определенной системе контроля динамического функционирования программ при исходной

гипотезе, что ***некоторые ошибки и дефекты в программах и данных могут проявиться на любой стадии тестирования.*** Однако количество вызовов регистрирующих программ и контроль промежуточных результатов, требующих нарушения целостности исполнения функциональных программ, следует ограничивать, учитывая допустимые расходы ресурсов времени на их реализацию. Так как основная задача регистрации при тестировании в реальном времени состоит в обнаружении и локализации ошибок и причин отказов с точностью до функциональной группы программ или компонента, то более точное определение места дефекта следует переносить на тестирование по детерминированным сценариям вне реального времени.

Так как испытания современных крупных систем обработки информации и/или управления позволяют получать такое большое количество контрольных данных, что достаточно полный их анализ представляет трудную методическую и техническую задачу, ***обработку динамических результатов целесообразно осуществлять иерархически и дифференцировано***. При избытке контролируемых величин снижается общее быстродействие имитаторов и комплекса программ в результате затрат времени на контроль и регистрацию. При переходе к массовым экспериментам испытаний функций и реализации их качества, приходится значительно сокращать количество анализируемых параметров и по возможности представлять их в обобщенном виде. В каждом конкретном случае необходимо стремиться к компромиссу между полнотой регистрации промежуточных данных тестирования и удобством ***анализа обобщенных результатов на соответствие требованиям***.

Обработка результатов испытаний программных продуктов реального времени может быть разделена на две достаточно автономные части: оперативную и обобщающую. ***Оперативная обработка результатов динамического тестирования*** должна производиться по упрощенным алгоритмам с большой пропускной способностью, обеспечивающим сохранение реального масштаба времени для всего испытываемого комплекса программ. Основная часть оперативной обработки результатов связана с замыканием контура обратной связи ***для имитации динамики функционирования управляемых объектов внешней среды***. Оперативно следует производить селекцию некоторых результатов тестирования и их предварительную обработку для значительного сокращения объема сохраняемых результатов.

В оперативную обработку целесообразно включать расчет части интегральных данных динамического тестирования, ***позволяющих контролировать текущий процесс обработки информации испытываемым комплексом***. Желательно выделять, регистрировать и отображать критические значения параметров или ситуации, угрожающие надежности и безопасности функционирования. Объем таких оперативно отображаемых данных должен быть максимально сокращенным и в то же время достаточным для анализа критических ситуаций, отражающихся на качестве динамического функционирования программного продукта. Эти данные должны позволять специалистам, ведущим испытания, фиксировать условия, при которых проявляются дефекты в функционировании программ, с учетом того, что автоматическая регистрация всегда имеет пробелы в составе фиксируемых параметров.

Обобщающая обработка накопленных результатов испытаний может производиться вне реального времени после завершения одного или серии испытаний. Основная задача при этом состоит в расчете различных интегральных характеристик качества функционирования программного продукта и ***соответствия их заданным требованиям***. При натурных экспериментах с внешними объектами для получения эталонных данных в реальном времени могут использоваться специальные измерительные комплексы.

Зарегистрированные и обработанные результаты испытаний должны использоваться для установления соответствия полученных характеристик качества заданным требованиям. При выявлении их отклонения от требований технического задания заказчика, спецификаций требований или декларируемых в документации, должны разрабатываться корректировки программ для устранения несоответствия. Для этого все этапы тестирования и испытаний программных продуктов должны быть поддержаны системой конфигурационного управления версиями программных компонентов и базой данных документирования тестов, результатов испытаний и выполненных корректировок программ (см. стандарты **ISO 10007** и **ISO 15846**). Средства накопления сообщений об отказах, ошибках, предложениях на изменения, выполненных корректировках и оцененных характеристиках качества версий являются основой для конфигурационного управления развитием и совершенствованием комплекса программ.

Опыт ***динамических испытаний надежности и функциональной безопасности крупных программных продуктов реального времени*** показал, что качество решения функциональных задач на отдельных этапах управления может быть объективно оценено лишь в комплексе со всей цепью управления и имитации необходимой информации: внешней обстановки, характеристик входной информации с учетом ошибок, взаимодействующих и обеспечивающих систем. Это возможно за счет создания имитационно-моделирующих стендов и обеспечения взаимодействия с опытными образцами компонентов систем и их программными продуктами. Такой подход оказался наиболее целесообразным, способствующим повышению эффективности научных и опытно-конструкторских разработок для крупных комплексов программ реального времени.

Затраты на имитацию внешней среды и на генерацию динамических тестов для испытаний программных продуктов реального времени могут быть одной из существенных составляющих при их создании. В ряде случаев, они соизмеримы с затратами на создание основных функций комплексов программ, что определяется принципиальным соответствием ***сложности необходимых наборов тестов*** и тестового покрытия программ, и ***сложности функций***, реализуемых испытываемым комплексом программ. Создание представительных совокупностей динамических тестов возможно путем использования реальных объектов внешней среды или с помощью программных имитаторов, адекват-

ных этим объектам по результатам функционирования и генерируемой информации. При этом возникает проблема – ***какой метод и когда выгодней*** по затратам на генерацию тестов и по обеспечению необходимой степени покрытия тестами испытываемых комплексов программ.

Имитаторы тестов могут быть необходимы не только для оценивания достигнутых характеристик качества комплексов программ, но также для их комплексной отладки, квалификационного тестирования, испытаний и при создании версий. Поэтому затраты на программные имитаторы и их экономическую эффективность целесообразно рассматривать в проекте с учетом всего комплекса задач, которые они способны и должны решать в жизненном цикле программного комплекса. Анализ эффективности программной имитации внешней среды при разработке и определении качества целесообразно разделять ***на две части***: оценка факторов, определяющих эффективность средств имитации тестов, и оценка экономического выигрыша при моделировании внешней среды на компьютерах по сравнению с натурными экспериментами в реальных системах.

Факторы, определяющие эффективность программной имитации внешней среды на компьютере при разработке крупных комплексов программ, могут оцениваться в основном по их воздействию на качество создаваемых систем. Некоторые значения тестов не только трудно создать при натурных экспериментах, но они являются маловероятными в реальных условиях. Однако такие, даже маловероятные ситуации и значения тестов могут быть ***критическими и/или особо важными*** для безопасности при функционировании всей системы, для которой разрабатывается программный продукт. Выбор и имитация подобных ситуаций позволяют отрабатывать и оценивать качество в критических маловероятных ситуациях, которые невозможно или опасно создавать на реальных объектах, но без их выполнения некоторые продукты недопустимо эксплуатировать в критических системах управления и обработки информации реального времени.

Экономическую эффективность программной имитации внешней среды на компьютере по сравнению с натурными экспериментами целесообразно оценивать при одинаковых объемах динамических тестовых данных для испытаний и определения качества. Показателем экономической эффективности имитации может служить ***соотношение затрат*** на проведение натурных экспериментов и затрат на программную имитацию той же совокупности тестовых и эталонных данных. Затраты ресурсов на натурные эксперименты для генерации тестов при проведении разработки, испытаний и определения качества пропорциональны реальному времени функционирования проверяемого программного продукта и затратам на применение привлекаемых средств реальной внешней среды. Они включают стоимость эксплуатации реального объекта, создающего динамические тесты в единицу времени (например, затраты на функционирование административной системы, прокатного стана или системы управления воздушным движением и всех управляемых ею объектов). Таким образом, затраты на натурные эксперименты для оценивания характеристик

комплексов программ определяются использованием всей реальной внешней среды, в которой предстоит в дальнейшем функционировать программам, а также затратами на средства измерения характеристик этой среды и проверяемого программного продукта в процессе разработки, испытаний и определения качества.

Затраты на программную имитацию динамических тестовых данных определяются ресурсами необходимыми на проектирование и эксплуатацию сложных комплексов программ для этих целей. Имитационные стенды практически всегда являются уникальными. В ряде случаев эти комплексы программ могут иметь объем порядка $10^5 – 10^6$ строк текста и должны создаваться с применением современных технологических систем. Затраты на эксплуатацию программ имитации в основном определяются длительностью проведения динамического тестирования, испытаний и/или измерения надежности и безопасности.

Даже приближенные оценки при системном анализе соотношения этих затрат в большинстве случаев показывают ***высокую рентабельность программных имитаторов внешней среды***, особенно для оценивания характеристик качества крупных программных продуктов реального времени. Например, при тестировании системы для управления воздушным движением, применение имитационных стендов, по крайней мере, на порядок снижает затраты по сравнению с натурными экспериментами и использованием реальных объектов (самолетов), а для управления космическими аппаратами или атомными электростанциями это соотношение может быть значительно больше (~ 10 – 100). При создании и определении качества административных систем с полной загрузкой, имитация способна заменить сложную организацию функционирования по определенной программе большого коллектива операторов банка, налоговой инспекции или таможенного органа.

Примером сложного испытательного стенда и моделей внешней среды для динамического тестирования и испытаний ***на соответствие требованиям*** к функциям и характеристикам комплексов программ может рассматриваться система ***управления полетами воздушных судов и диспетчерских систем в центрах управления воздушным движением***. Для комплексной отладки, тестирования, испытаний и сертификации программных продуктов управления воздушным движением (УВД) проводится имитация в реальном времени всей информации, поступающей из внешней среды. Источниками информации для центров УВД являются радиолокационные станции, летный состав на борту воздушных судов, диспетчеры управления воздушным движением и исходные планы полетов. Вследствие этого необходимо динамически имитировать ряд разнородных объектов с учетом интенсивных случайных воздействий от них, а также при наличии управления со стороны диспетчеров центра УВД и летного состава на борту воздушных судов. Некоторые редкие проявления ошибок в программных продуктах могут компенсироваться диспетчерами, контролирующими функционирование центра УВД. Имитировать реакцию и действия диспетчеров автоматически на компьютерах очень трудно, так как они в значи-

тельной степени зависят от квалификации и конкретных психологических особенностей поведения диспетчеров при различных ситуациях воздушной обстановки.

5.2. Сертификация функциональной безопасности сложных программных продуктов реального времени

Потребителей – заказчиков программных продуктов управляющих систем интересует, прежде всего, ***качество и безопасность готового продукта*** и обычно не очень беспокоит, как оно достигнуто. Однако это качество должно быть ответственно ***удостоверено и гарантировано*** компетентными, независимыми организациями. Гарантирование качества продукции возможно посредством сертификационных испытаний ***процессов производства*** комплексов программ и/или испытаний их результатов – ***готовых программных продуктов*** [3, 18, 34]. Рассматриваемые далее заказные программные продукты для систем управления и обработки информации в реальном времени активно применяются в сложных критических и ответственных системах динамического управления объектами. Проектирование и производство таких программных продуктов, обычно требуется заказчиками базировать на международных стандартах, охватывающих весь их жизненный цикл.

Сертификация – это процедуры подтверждения соответствия продукции требованиям и стандартам, установленным заказчиком независимым от изготовителя и потребителя. Сертификационные испытания должны, ***технически и юридически удостоверять*** в письменной форме, что состояние продукции, процессов его производства и системы менеджмента качества, способны обеспечить требуемое качество и стабильность характеристик изготовляемой продукции любыми ***двумя методами*** [18].

Первый метод должен обеспечивать ***высокое качество выполнения всего технологического процесса*** проектирования и производства, и тем самым минимум экономических потерь от брака, что особенно важно при создании сложных дорогих систем. Результаты испытаний качества ***процессов проектирования и производства*** трудно измерять количественными критериями, и обычно характеризуются рядом требований к качественному выполнению наборов стандартизированных производственных процессов. Они оцениваются свойствами различных процессов, непосредственно отражающимися на характеристиках качества программного продукта, однако при этом нет гарантии адекватного и однозначного требуемого качества конечного продукта. При производстве этот метод может приводить к неконтролируемому, неизвестному качеству компонентов и комплексов программ в целом, и к значительным экономическим потерям за счет затрат на создание не пригодного к использованию продукта (брака), что может быть дорого для сложных систем.

Второй метод сертификации, акцентирован на ***анализе, контроле и удостоверении качества готового программного продукта***, которое удостоверяется при его испытаниях. Отсутствие или недостатки системы обеспечения качества

в технологическом процессе разработки, могут приводить к длительному итерационному процессу доработок и повторных испытаний. При сертификационных испытаниях готового программного продукта могут использоваться его стандартизированные количественные и качественные критерии качества и характеристики, которые непосредственно отражают функции и свойства продукции, интересующие заказчика и потребителей, их можно измерить и достоверно установить реальные значения.

Соответственно можно выделить ***два вида сертификационных испытаний***: ***технологий*** обеспечения жизненного цикла программных комплексов, поддержанных регламентированными системами качества; и испытаний готового программного ***продукта*** с полным комплектом эксплуатационной документации. Взаимосвязь качества разработанных комплексов программ с качеством технологии их создания и с затратами на производство становится особенно существенной при необходимости получения ***критического заказного программного продукта реального времени*** с особенно высокими значениями характеристик качества при ограниченных ресурсах [18]. Этот вид комплексной сертификации должен обеспечивать контроль реализации требований алгоритмической и функциональной корректности программного продукта, что особенно важно в программных комплексах для ***обеспечения функциональной безопасности применения сложных управляющих систем***.

Сертификация производства продукции различных видов регламентирована стандартами: **ГОСТ Р ИСО 9001:2001; ГОСТ Р 40.003: 2005** и **ГОСТ Р ИСО 19011: 2003,** а также ***комплексом международных стандартов*** создания и жизненного цикла программных продуктов и их компонентов (см. Приложение 1). Они акцентированы на ***Системе менеджмента качества производства.*** При этом ***сертификация производства*** определена как ***процедура подтверждения соответствия***, посредством которой независимая от изготовителя (продавца, исполнителя) и потребителя (покупателя) организация удостоверяет в письменной форме, что состояние производства (системы менеджмента качества производства) способно обеспечить требуемое качество и стабильность характеристик изготовляемой конкретной продукции.

Испытания производства и программных продуктов должны осуществлять эксперты (аудиторы) по сертификации производств, зарегистрированные в Регистре системы сертификации персонала – ***сертифицированные специалисты***. Область сертификации определяет заказчик по согласованию с председателем комиссии органа по сертификации конкретной продукции. На практике акцент, распределение ресурсов и усилий на два вида сертификации зависят от особенностей характеристик комплекса программ, квалификации коллектива специалистов – разработчиков, требований заказчика – потребителей, и наличия у сертификационной лаборатории соответствующей тематической квалификации. Для этого организация и процессы сертификации должны специализироваться на определенные классы программных продуктов, предусматривать соответствующие технологические работы и документы, обеспечивающие создание продукта требуемого качества.

Для ***сертификации технологических процессов*** предприятие должно установить ***перечень процессов и документов***, которые необходимы для управления производством программных продуктов, а также для обеспечения уверенности в соответствии продукции требованиям заказчика и стандартов.

Достижение высоких значений качества комплексов программ существенно зависит от качества – ***зрелости*** технологии и инструментальных средств, используемых разработчиками при проектировании и производстве программного продукта. Оценивание уровня зрелости технологической базы жизненного цикла (ЖЦ) позволяет прогнозировать возможное качество продукта и ориентировать заказчика и пользователей при выборе разработчика и поставщика проекта с требуемыми характеристиками. Поэтому определение уровня ***зрелости*** технологической поддержки процессов жизненного цикла, организационного и инструментального обеспечения, непосредственно ***связано с оцениванием реальных или возможных характеристик качества производства*** конкретного программного продукта.

Практически все требования к производству программных продуктов должны ***соответствовать*** регламентированным и детализированным требованиям в стандартах **ISO 9001:2000, ISO 12207:2008** и базовых компонентах ***стандартов жизненного цикла сложных*** комплексов программ (см. главу 3). На их основе формируются процедуры сертификации производства программных продуктов. Для сертификации производства и системы менеджмента качества предприятия, необходима ***четкая организация документирования производства***. Входные документы для производства должны включать все требования, существенные для проектирования и производства программного продукта. Выходные данные процесса проектирования и/или производства должны быть зарегистрированы в документах в форме, дающей возможность проверки их по отношению к входным требованиям. Документы, содержащие выходные данные, должны быть утверждены до их применения при сертификации.

Документ, содержащий ***результаты сертификационных испытаний*** производства и системы менеджмента качества программных продуктов, должен включать Программу и методики сертификационных испытаний – аудита производства предприятия, протоколы и отчет аудиторов о результатах испытаний качества производства программного продукта. В документе должны быть представлены результаты аудита, выводы и рекомендации комиссии, оформленные в виде акта удостоверяющего качество производства. Завершение сертификации, выдача и регистрация сертификата по результатам аудита – испытаний производства программного продукта, должен определять достаточность качества программного продукта для поставки потребителям и обеспечения его жизненного цикла, а также для регистрации лицензии на применение знаков соответствия.

Сертификация проектирования и производства программных продуктов высокого качества включает следующие основные процессы (рис. 5.1)***:***

Требования к составу документов для организации сертифицируемого производства включают:

комплект должностных инструкций, определяющих ответственность, и порядок взаимодействия специалистов, для производства программного продукта;

- положение о подразделениях и должностные инструкции, обязанности и полномочия специалистов, реализующих производство программных продуктов;
- набор характеристик комплекса программ для сертификационных испытаний;

состав документации процессов и результатов сертификации – аудита технологии и системы качества производства программных продуктов:

- задание клиента на проведение сертификационных испытаний производства программного продукта;
- план сертификации производства программного продукта;

документы сертификационной лаборатории:

- требования к документации результатов внутренних испытаний производства программного продукта;
- состав технологических средства автоматизации и Программа сертификационных испытаний производства;
- методики испытаний по каждому разделу требований процессов производства программного продукта;
- отчеты выполнения и результаты испытаний производства;
- отчет клиенту – заявителю о проверках организации сертификационных испытаний – аудита и системы качества производства программного продукта.

Рис. 5.1.

При ***сертификации программных продуктов*** сертификаторы должны иметь четкое ***представление о потребностях заказчика***. Необходимо изучить системные требования, сценарии использования системы и/или описание назначения системы для того, чтобы лучше понять цель разработки программного продукта, выбор методов и средств его тестирования, подготовить и согласовать с заказчиком процессы:

2.1. Определение конкретной среды, процессов производства и основных характеристик программного продукта.

Подготовка к сертификации производства и системы качества программных комплексов и предприятия.

2.3. Подготовка и документирование организации процессов сертификации производства и системы менеджмента качества предприятия.

2.4. Организация сертификационных испытаний и системы менеджмента управления качеством производства программных продуктов.

2.5. Анализ результатов и завершение сертификационных испытаний производства программного продукта.

Требования к тестам и документы должны содержать подробный перечень того, что и как должно быть испытано. ***Стратегия сертификационных испытаний*** – это документ, отражающий совокупность выбранных методов, требований и решений, вытекающих из целей и задач проекта и его тестирования, общие правила и принципы, способствующие достижению целей разработки программного продукта высокого качества. ***Сертификация готовых заказных программных продуктов высокого качества включает следующие основные процессы:***

3.1. Формирование требований к характеристикам качества для сертификации программного продукта.

3.2. Оценка процессов и ограничений сертификационных испытаний программных продуктов на соответствие требованиям.

3.3. Организация и планирование сертификационных испытаний программных продуктов и компонентов на соответствие требованиям.

3.4. Стратегии испытаний качества программных продуктов.

3.5. Подготовка тестов для сертификационных испытаний программных продуктов.

3.6. Предварительные испытания и опытная эксплуатация разработчиками качества программного продукта.

3.7. Завершение сертификационных испытаний и удостоверение качества программных продуктов (см.рис. 5.2).

Результаты сертификационных испытаний сложного программного продукта включают:

исходные документы заявителя для выполнения сертификации заказного программного продукта:

- техническое задание – требования к функциям, характеристикам, качеству программного продукта, системы и внешней среды;
- договор заказчика с производителем на качество программного продукта;
- полные тексты программ, содержание базы данных и технологической документации программного продукта;
- комплект эксплуатационных документов, поставляемых заказчику и пользователям для применения программного продукта;
- план, Программу и методики испытаний, применения и оценки качества программного продукта;
- руководство по генерации и инсталляции пользовательских версий и загрузке базы данных в соответствии с условиями и характеристиками внешней среды;

договор заявителя с сертифицирующей организацией на проведение испытаний версии программного продукта:

- отчет сертификаторов о реализации Программы и методик проведения сертификационных испытаний программного продукта в соответствии с Договором на сертификацию с заявителем;
- результаты аттестации имитаторов внешней среды и генераторов динамических тестов для сертификационных испытаний программного продукта;
- результаты выполнения планов и методик сертификационных испытаний, протоколы испытаний предъявляемым требованиям, утвержденным сертификаторами и согласованным с заявителями;

Акт результатов сертификационных испытаний программного продукта, реальные характеристики программного продукта, выводы о их соответствии требованиям к характеристикам заказчика программного продукта;

- **сертификат** заказного программного продукта, лицензия на применение знаков соответствия;
- **удостоверение** для поставки и применения пользователями сертифицированного программного продукта.

Рис. 5.2.

Завершаются предварительные испытания разработчиков предъявлением заказчику на утверждение комплекта документов, содержащих результаты, необходимые для сертификационных испытаний программного продукта. При сертификационных испытаниях программного продукта целесообразно выбо-

рочно или полностью использовать результаты предварительных испытаний с учетом их полноты и достоверности**.** Утверждение комплекта документов для конкретного программного продукта дает право на присвоение ему ***сертификата и знака качества***. Отчетный доклад о результатах испытаний должен содержать перечень ***всех не устраненных дефектов***, с соглашением и планом того, будут ли они исправлены в более поздних версиях или же их исправление откладывается на неопределенное время.

Снятие с эксплуатации и развития версий сертифицированного программного продукта должны быть подготовлены анализом обосновывающее это решение. При снятии программного продукта с сопровождения следует определить необходимые для этого действия, а затем разработать и документально оформить этапы работ, обеспечивающие их выполнение. Должны быть предусмотрены возможности ***доступа к полным архивным документам*** снятого с сопровождения программного продукта.

Современные заказные программные продукты для управления и обработки информации активно применяются в сложных критических и ответственных системах динамического управления объектами различного назначения. Ущерб от дефектов программных продуктов таких систем может определяться огромной их стоимостью и жизней пользователей. На реализацию требуемого качества требуются ресурсы, которые могут быть соизмеримыми с первичными затратами на весь процесс проектирования и производства системы и программного продукта, а также на подготовку и воспитание профессиональных квалифицированных коллективов специалистов в области индустрии сложных программных продуктов высокого качества. Для достижения и удостоверения требуемого высокого качества необходимо создание и ***внедрение методов и средств автоматизации сертификации производственных процессов*** и их результатов – ***программных продуктов*** сложных систем.

Литература

1. Барлоу Р., Прошан Ф. Статистическая теория надежности и испытания на безотказность. - М.: Наука, 1984.
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. Вигерс К.И. Разработка требований к программному обеспечению. Пер. с англ. – М.: Русская редакция. 2004.
5. Галатенко В.А. Основы информационной безопасности. Курс лекций. Интернет - Университет Информационных Технологий. М.: ИНТУИТ. 2003.
6. Гецци К., Джазайери М., Мандриоли Д. Основы инженерии программного обеспечения. Пер. с англ. – СПб.: БХВ-Петербург. 2005.
7. Дастин Э., Рэшка Д., Пол Д. Автоматизированное тестирование программного обеспечения. Внедрение, управление и эксплуатация. Пер. с англ. – М. ЛОРИ. 2003.
8. Канер С., Фолк Д., Нгуен Е. Тестирование программного обеспечения. Пер. с англ. – М: ДиаСофт. 2001.
9. Кантор М. Управление программными проектами. Практическое руководство по разработке успешного программного обеспечения. Пер. с англ. – М.: Вильямс. 2002.
10. Концепция защиты средств вычислительной техники и автоматизированных систем от несанкционированного доступа к информации. Руководящий документ. Гостехкомиссия России. М.: Военное издательство. 1999.
11. Леффингуэлл Д., Уидриг Д. Принципы работы с требованиями к программному обеспечению. Унифицированный подход. Пер. с англ. – М.: Вильямс. 2002.
12. Липаев В.В. Надежность программных средств. М.: СИНТЕГ. 1998.
13. Липаев В.В. Методы обеспечение качества крупномасштабных программных средств. – М.: РФФИ. СИНТЕГ. 2003.
14. Липаев В.В. Функциональная безопасность программных средств. – М.: СИНТЕГ. 2004.
15. Липаев В.В. Анализ и сокращение рисков проектов сложных программных средств. – М.: СИНТЕГ. 2004.
16. Липаев В.В. Программная инженерия. Методологические основы. Учебник. – М.: ТЕИС. 2006.
17. Липаев В.В. Тестирование компонентов и комплексов программ. Учебник. – М.: СИНТЕГ. 2010.
18. Липаев В.В. Сертификация программных средств, Учебник. – М.: СИНТЕГ. 2010.
19. Соммервилл И. Инженерия программного обеспечения. 6-е издание. Пер. с англ. – М.: Вильямс. 2002.
20. Стрелков Ю. К. Инженерная и профессиональная психология. – М.: Академия. 2001.
21. Торингтон Д., Холл Л., Темлер С. Управление человеческими ресурсами. Учебник. Пер. с англ. – М.: Дело и сервис. 2004.
22. 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.

23. Тэллес М., Хсих Ю. Наука отладки. Пер. с англ. – М.: Кудиц-образ. 2003.
24. Устинов Г.Н. Основы информационной безопасности систем и сетей передачи данных. М.: СИНТЕГ. 2000.
25. Уткин Л. В., Шубинский И. Б. Нетрадиционные методы оценки надежности информационных систем /Под ред. проф. И. Б. Шубинского – С.-Петербург, «Любавич», 2000.
26. Фатрелл Р. Т., Шафер Д. Ф., Шафер Л. И. Управление программными проектами: достижение оптимального качества при минимальных затратах. Пер. с англ. – М.: Вильямс. 2003.
27. Boehm B.W. et al. Software cost estimation with COCOMO II. Prentice Hall PTR. New Jersey. 2000.
28. Boehm B.W. Software risk management. IEEE Computer Society Press. Washington. 1989.
29. Charett R. Software engineering risk analysis and management. N.Y.: McGraw – Hill. 1989.
30. Davis A. Software requirements: Objects, functions and states. – Englewood Cliffs. NY. Prentice-Hall. 1993.
31. Jones C. Applied software measurement, assuring productivity and quality. McGraw-Hill. NY. 1996.
32. Higuera R., Haimes Y. Software risk management. Pittsburg. Software engineering institute, Cornegie Mellon University. – 1996.
33. Kit E. Software Testing in the Real World - Improving the Process. Addison-Wesley.1996.
34. Karolak D. W. Software engineering risk management. IEEE Computer Society Press. Washington. 1996.
35. Smith D, Simpson K. Functional Safety (A Straightforward Guide IEC 61508 and Related Standards) – Oxford: Planta Tree, 2001.
36. Shooman M.L. Software Engineering: Reliability, Development and Management. N.Y. McGraw-Hill. 1983.

Приложение

Перечень основных стандартов в области обеспечения надежности и функциональной безопасности сложных комплексов программ

1. **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.
2. **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.
3. **ISO 13335 - 1-5. 1996-1998.** ИТ. ТО. Руководство по управлению безопасностью. Ч. 1. Концепция и модели обеспечения безопасности 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.
4. **ISO 10181: 1-7. BOC. 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.
5. **IEC 60880: Ч. 1.1986**. Программное обеспечение компьютеров в системах безопасности атомных электростанций. **Ч.2. 2000**. Программные аспекты защиты от отказов по общим причинам, использование программных инструментов и ранее разработанного программного обеспечения.
6. **ГОСТ Р 51904 – 2002**. Программное обеспечение встроенных систем. Общие требования к разработке и документированию.
7. **ISO 12207: 2008.** ИТ. Процессы жизненного цикла программных средств.
8. **ISO 15271:1998.** (ГОСТ Р – 2002). ИТ. Руководство по применению ISO 12207.
9. **ISO 16326:1999.** (ГОСТ Р – 2002). ИТ. Руководство по применению ISO 12207 при административном управлении проектами.
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 10005: 1995** - Административное управление качеством. Руководящие указания по программам качества.
14. **ISO 10006: 1997** - Руководство по качеству при управлении проектом.
15. **ISO 10007: 1995** - Административное управление качеством. Руководящие указания при управлении конфигурацией.
16. **ISO 12182:1998.** (ГОСТ Р– 2002). ИТ. Классификация программных средств.

17. **ISO 9126:1991.** (ГОСТ – 1993). ИТ. Оценка программного продукта. Характеристики качества и руководство по их применению.
18. **ISO 9126-1-4:2011.** ИТ. Качество программных средств: Ч.1. Модель качества. Ч.2. Внешние метрики. Ч. 3. Внутренние метрики. Ч. 4. Метрики качества в использовании.
19. **ISO 25000:2005.** Программная инженерия. Качество и развитие программных продуктов: Термины и определения; базовая модель; основное руководство; требования к спецификациям; планирование и управление; измерение и развитие.
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 14756: 1999.** ИТ. Измерение и оценивание производительности программных средств компьютерных вычислительных систем.
22. **ISO 12119:1994.** (ГОСТ Р – 2000 г). ИТ. Требования к качеству и тестирование.
23. **ISO 15846:1998.** ТО. Процессы жизненного цикла программных средств. Конфигурационное управление программными средствами.
24. **ISO 14764: 1999.** (ГОСТ Р – 2002). ИТ. Сопровождение программных средств.
25. **ISO 15910:1999.** (ГОСТ Р – 2002) ИТ. Пользовательская документация программных средств.
26. **ISO 6592:2000.** ОИ. Руководство по документации для вычислительных систем.
27. **ISO 9294:1990.** (ГОСТ–1993 г). ТО. ИТ. Руководство по управлению документированием программного обеспечения.
28. **ГОСТ Р 51901-2002.** Управление надежностью. Анализ риска технологических систем.
29. **DO-178 B -1995.** Соглашение по сертификации бортовых систем и оборудования в части программного обеспечения.

Автор – **Владимир Васильевич Липаев** – доктор технических наук, профессор, главный научный сотрудник Института системного программирования РАН, Заслуженный деятель науки и техники РСФСР, Лауреат премии Совета министров СССР, Лауреат премии Правительства РФ в области образования. Награжден орденами СССР: Красной Звезды и Трудового Красного Знамени.

С 1954 по 1988 год работал в Московском НИИ приборной автоматики. До 1988 года – главный конструктор Министерства радиопромышленности СССР по автоматизации проектирования программных средств, по технологии создания крупномасштабных программных продуктов для оборонных систем реального времени.

Многие годы занимался исследованиями и руководил разработкой программных комплексов для обработки радиолокационной информации систем противовоздушной обороны, в сотне экземпляров размещенных на территории Страны. Под его руководством разработан ряд больших инструментальных системы программной инженерии реального времени высокого качества для автоматизации технологических процессов жизненного цикла сложных комплексов программ, широко использовавшихся в оборонной промышленности.

Автор около 50 монографий и учебников в области методов, технологий, инструментальных средств, тестирования и испытаний, стандартизации и сертификации проектирования и производства сложных программных продуктов. В последние годы опубликовал цикл учебников и монографий для вузов по программной инженерии, по методам и процессам промышленного производства крупных программных продуктов, реального времени.