ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВСТРОЕННЫХ СИСТЕМ

Общие требования к разработке и документированию

Издание официальное

ГОССТАНДАРТ РОССИИ
Москва

gost.e-ecolog.ru

Предисловие

1 РАЗРАБОТАН Государственным научно-исследовательским институтом авиационных систем с участием Научно-исследовательского института стандартизации и унификации

ВНЕСЕН Научно-исследовательским институтом стандартизации и унификации

2 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Постановлением Госстандарта России от 25 июня 2002 г. № 247-ст

3 Стандарт подготовлен в развитие ГОСТ Р ИСО/МЭК 12207—99 «Информационная технология. Процессы жизненного цикла программных средств» с целью учета специфики разработки и документирования программного обеспечения встроенных систем реального времени

4 ВВЕДЕН ВПЕРВЫЕ

5 ПЕРЕИЗДАНИЕ. Октябрь 2005 г.

© ИПК Издательство стандартов, 2002
© Стандартинформ, 2005

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1 Область применения 1
2 Нормативные ссылки 1
3 Определения и сокращения 1
4 Общие требования 5
4.1 Жизненный цикл ПО 5
4.2 Общие требования для разработки ПО 6
5 Системные аспекты, связанные с разработкой ПО 7
5.1 Поток информации между процессами жизненного цикла системы и ПО 7
5.2 Отказные ситуации и уровни ПО 7
5.3 Анализ системных требований 9
5.4 Проектирование системы 10
5.5 Стратегии архитектурного проектирования системы 11
5.6 Библиотека разработки ПО 11
5.7 Файлы разработки ПО 12
5.8 Непоставляемое ПО 12
5.9 Подготовка к использованию ПО 12
5.10 Подготовка к передаче ПО 13
5.11 Совместные технические и административные просмотры 14
5.12 Другие действия 14
6 Процесс планирования ПО 15
6.1 Цели процесса планирования ПО 15
6.2 Состав работ, выполняемых в процессе планирования ПО 15
6.3 Типы планов ПО 16
6.4 Планирование среды жизненного цикла ПО 16
6.5 Стандарты разработки ПО 17
6.6 Ответственность за выполнение планирования 18
6.7 Просмотр результатов процесса планирования ПО 18
7 Процессы разработки ПО 18
7.1 Процесс определения требований к ПО 19
7.2 Процесс проектирования ПО 19
7.3 Процесс кодирования ПО 21
7.4 Процесс интеграции 21
7.5 Трассируемость 22
8 Процесс верификации ПО 22
8.1 Цели верификации ПО 23
8.2 Состав работ, выполняемых в процессе верификации ПО 23
8.3 Просмотры и анализы ПО 23
8.4 Цели и методы тестирования ПО 25
8.5 Порядок выполнения тестирования ПО 28
9 Процесс управления конфигурацией ПО 30
9.1 Цели процесса управления конфигурацией ПО 30
9.2 Состав работ, выполняемых в процессе управления конфигурацией ПО 31
9.3 Категории контроля документов 35
9.4 Аудит конфигурации 35
9.5 Компоновка и поставка ПО 35
10 Процесс обеспечения качества ПО 35
10.1 Цели процесса обеспечения качества ПО 36
10.2 Состав работ, выполняемых в процессе обеспечения качества ПО 36
10.3 Просмотр согласованности ПО 36
10.4 Документирование обеспечения качества ПО 37
10.5 Независимость в обеспечении качества ПО 37

11 Процесс сертификационного сопровождения . . . 37
11.1 Средства согласования и планирования . . . 37
11.2 Обоснование согласованности . . . 37
11.3 Минимальный состав документов жизненного цикла ПО, передаваемых сертифицирующей организации . . . 38
11.4 Документы жизненного цикла ПО, относящиеся к типовому проекту . . . 38
12 Документы, создаваемые в процессах жизненного цикла ПО . . . 38
12.1 План сертификации в части ПО . . . 39
12.2 План разработки ПО . . . 40
12.3 План верификации ПО . . . 40
12.4 План квалификационного тестирования ПО . . . 41
12.5 План управления конфигурацией ПО . . . 41
12.6 План обеспечения качества ПО . . . 42
12.7 План установки ПО . . . 42
12.8 План передачи ПО . . . 42
12.9 Стандарты на разработку требований к ПО . . . 43
12.10 Стандарты на процесс проектирования ПО . . . 43
12.11 Стандарты кодирования ПО . . . 43
12.12 Спецификация системы/подсистемы . . . 44
12.13 Спецификация требований к ПО . . . 44
12.14 Спецификация требований к интерфейсу . . . 44
12.15 Описание проекта системы/подсистемы . . . 45
12.16 Описание проекта ПО . . . 45
12.17 Описание проекта интерфейса . . . 45
12.18 Описание проекта базы данных . . . 45
12.19 Исходный код ПО . . . 46
12.20 Исполняемый объектный код ПО . . . 46
12.21 Процедуры верификации ПО . . . 46
12.22 Описание квалификационного тестирования ПО . . . 46
12.23 Результаты верификации ПО . . . 46
12.24 Отчет о квалификационном тестировании ПО . . . 47
12.25 Указатель конфигурации среды жизненного цикла ПО . . . 47
12.26 Указатель конфигурации ПО . . . 47
12.27 Спецификация программного средства . . . 47
12.28 Сообщения о дефектах . . . 47
12.29 Протоколы управления конфигурацией ПО . . . 48
12.30 Протоколы обеспечения качества ПО . . . 48
12.31 Итоговый документ разработки ПО . . . 48
12.32 Описание эксплуатационной концепции . . . 49
12.33 Руководство по эксплуатации компьютера . . . 49
12.34 Руководство по программированию для компьютера . . . 49
12.35 Руководство поддержки программно-аппаратных средств . . . 50
12.36 Руководство оператора ПО . . . 50
12.37 Руководство по входной/выходной информации ПО . . . 50
12.38 Руководство пользователя ПО . . . 50
12.39 Описание версии ПО . . . 51
13 Дополнительные вопросы . . . 51
13.1 Использование ранее разработанного ПО . . . 51
13.2 Аттестация инструментальных средств . . . 53
ПРИЛОЖЕНИЕ А Цели и результаты процессов в зависимости от уровня ПО . . . 56

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВСТРОЕННЫХ СИСТЕМ

Общие требования к разработке и документированию

Embedded system software.
General requirements for development and documentation

Дата введения 2003—07—01

1 Область применения

Настоящий стандарт распространяется на процессы разработки и документирования программного обеспечения (ПО) встроенных систем реального времени. Стандарт распространяется на все действия, имеющие отношение к разработке программного обеспечения.

Настоящий стандарт применяют полностью ко всему поставляемому программному обеспечению, включая среду разработки, если контрактом не предусмотрено использование специальных стандартов для определенных заказчиком типов ПО. Стандарт неприменим для аппаратных элементов программно-аппаратного обеспечения.

2 Нормативные ссылки

В настоящем стандарте использована ссылка на следующий стандарт:

ГОСТ Р ИСО/МЭК 12207—99 Информационная технология. Процессы жизненного цикла программных средств

3 Определения и сокращения

В настоящем стандарте применяют термины с соответствующими определениями по ГОСТ Р ИСО/МЭК 12207, а также приведенные ниже:

3.1 **алгоритм:** Конечное множество четко определенных правил, которые задают последовательность действий для выполнения конкретной задачи.

3.2 **анализ полноты покрытия:** Определения степени, до которой работы процесса верификации ПО удовлетворяют поставленной цели.

3.3 **аномальное поведение:** Поведение, которое не соответствует заданным требованиям.

3.4 **аппаратные средства**: Материальная часть вычислительной системы, включающая в себя электрические и электронные элементы (например, приборы и схемы), электромеханические элементы (например, дисководы) и механические элементы (например, стойки).

3.5 **архитектура**: Организационная структура системы или ЭКПО, в которой идентифицированы компоненты, их интерфейсы и концепция взаимодействия между ними.

3.6 **аттестация инструментальных средств**: Процесс получения сертификационного доверия к программному инструментальному средству применительно к конкретной встроенной системе.

3.7 **база данных**: Совокупность взаимосвязанных данных, сохраненных в одном или более компьютерных файлах в виде, позволяющем обращаться к ним пользователям или компьютерным программам с помощью системы управления базой данных.

3.8 **библиотека разработки ПО**: Контролируемая совокупность документов, промежуточных и конечных программных продуктов, а также инструментальных средств и процедур, используемых для управления текущей разработкой и последующей поддержкой ПО.

3.9 **верификация:** Оценка результатов процесса с целью гарантии корректности и непротиворечивости в отношении входов и стандартов, существующих для данного процесса.

Издание официальное

★

3.10 **заплата:** Исправление, вносимое непосредственно в объектную программу, а не в текст на языке программирования.

3.11 **изменение ПО**: Модификация исходного кода, исполняемого объектного кода или сопутствующих документов относительно их базовой линии.

3.12 **имитатор:** Устройство, компьютерная программа или система, используемая при верификации ПО, которая принимает те же входные данные и производит те же выходные данные, что и объектная система.

3.13 **интегральный процесс**: Процесс разработки ПО, который остается активным на протяжении жизненного цикла ПО.

3.14 **интеграция аппаратуры и ПО:** Процесс объединения ПО с объектным компьютером.

3.15 **интеграция ПО:** Процесс объединения компонентов кода.

3.16 **интерфейс:** Взаимосвязь между двумя или более объектами (типа ЭКПО/ЭКПО, ЭКПО/ЭКА, ЭКПО/пользователь или между модулями ПО), которые совместно используют и обеспечивают данные или обмениваются ими.

3.17 **инструментальное средство:** Компьютерная программа, используемая как средство разработки, тестирования, анализа, производства или модификации других программ или документов на них.

3.18 **инструментальный компьютер:** Компьютер, на котором разрабатывают ПО.

3.19 **исходный код**: Код, написанный на исходном языке программирования, таком как язык ассемблера и/или язык высокого уровня, в машинно-читаемой форме, пригодной для ввода в ассемблер или компилятор.

3.20 **квалификационное тестирование:** Тестирование, выполняемое с целью убедить заказчика, что ПО соответствует заданным требованиям.

3.21 **класс эквивалентности:** Такое разбиение входной области программы, при котором тестирование для представительного значения класса эквивалентно тестированию для любого другого значения из этого класса.

3.22 **код**: Реализация конкретных данных или конкретной компьютерной программы в символьной форме, такой, например, как исходный код, объектный код или машинный код.

3.23 **коммерчески доступное ПО**: Коммерчески доступное программное средство, продаваемое производителем по официальным каталогам. Коммерчески доступное ПО не предназначено для переделки или усовершенствования. ПО, разработанное по специальным контрактам для специализированных приложений, не является коммерчески доступным ПО.

3.24 **компонент**: Замкнутая часть, комбинация частей или элемент, которые выполняют в системе отдельную функцию.

3.25 **контракт**: Соглашение о разработке ПО, установленное между заказчиком и разработчиком.

3.26 **критерии перехода**: Минимальные условия, определенные процессом планирования ПО, которые должны быть выполнены для входа в процесс.

3.27 **мертвый код**: Исполняемый объектный код (или данные), который в результате ошибки проектирования не может быть выполнен (код) или использован (данные) в функциональной конфигурации среды объектного компьютера и не может быть прослежен в системных или программных требованиях. Исключение составляют встроенные идентификаторы.

3.28 **многоверсионное неидентичное ПО**: Множество из двух или более программ, разработанных отдельно по одним и тем же функциональным требованиям. Ошибки одной версии обнаруживают путем сравнения выходных результатов разных программ.

3.29 **модифицированное покрытие условий/решений**: Такое выполнение программы при тестировании, при котором каждая точка входа и выхода программы должна быть вызвана хотя бы один раз; каждое условие в решении программы должно быть выполнено со всеми возможными результатами хотя бы один раз; все результаты каждого решения должны быть выполнены хотя бы один раз, и для каждого условия в решении должно быть показано его независимое влияние на результат решения. Независимость влияния условия на результат решения демонстрируют путем рассмотрения всех возможных комбинаций условий.

3.30 **непоставляемое программное средство**: Программное средство, которое в соответствии с контрактом не требуется поставлять заказчику или другому обозначенному получателю.

3.31 **объектный код**: Представление компьютерной программы на низком уровне, обычно не в форме, непосредственно пригодной для объектного компьютера, а в форме, включающей в себя, помимо информации о процессорных командах, информацию о размещении программы.

3.32 **объектный компьютер**: Компьютер, на котором эксплуатируют ПО.

3.33 **отказоустойчивость:** Свойство системы продолжать правильное выполнение функций при наличии ограниченного числа аппаратных или программных дефектов.

3.34 **отключенный код:** Исполняемый объектный код (или данные), который согласно проекту предназначен для выполнения (код) или использования (данные) только при определенных условиях.

3.35 **оценка безопасности системы**: Систематическая, всесторонняя оценка предлагаемой системы с целью показать, что она удовлетворяет требованиям, предъявленным к обеспечению безопасности.

3.36 **ошибка**: Неправильность в требованиях, проекте или коде.

3.37 **передача ПО**: Последовательность действий, определяющих ответственность за передачу разработанного ПО организацией, имеющей право на эти действия (обычно организацией, которая выполняет разработку ПО), в организацию, осуществляющую поддержку ПО.

3.38 **перепроектирование**: Процесс исследования и изменения существующей системы для преобразования ее в новую форму.

3.39 **поддержка ПО**: Набор действий, гарантирующий, что установленное для эксплуатационного использования ПО продолжает выполнять все функции в соответствии с предназначением системы. Поддержка ПО включает в себя сопровождение ПО, помощь пользователям и связанные с этим действия.

3.40 **покрытие операторов:** Такое выполнение программы при тестировании, при котором каждый оператор в программе должен быть выполнен хотя бы один раз.

3.41 **покрытие решений:** Такое выполнение программы при тестировании, при котором каждая точка входа и выхода программы должна быть вызвана хотя бы один раз; каждое условие в решении должно быть выполнено с каждым возможным результатом хотя бы один раз.

3.42 **покрытие условий/решений**: Такое выполнение программы при тестировании, при котором каждая точка входа и выхода программы должна быть вызвана хотя бы один раз; каждое условие в решении программы должно быть выполнено со всеми возможными результатами хотя бы один раз; все результаты каждого решения должны быть выполнены хотя бы один раз.

3.43 **поставляемое программное средство:** Программное средство, требуемое по контракту, которое будет поставлено заказчику или другому обозначенному получателю.

3.44 **построение:** Версия ПО, отвечающая определенному подмножеству требований, которые должны быть обеспечены в конечном ПО.

3.45 **прерывание:** Приостановка задачи, например выполнения компьютерной программы, вызванная событием, внешним для этой задачи. Прерывание позволяет обработать возникшее событие и вернуться к прерванной задаче.

3.46 **программная система**: Система, состоящая из ПО и, возможно, компьютерного оборудования для его выполнения.

3.47 **программное обеспечение (ПО):** Совокупность компьютерных программ и программных документов, необходимых для эксплуатации этих программ.

3.48 **программное средство**: ПО и связанные с ним документы, вновь созданные, модифицированные или сгруппированные для удовлетворения требованиям контракта.

3.49 **программное средство многократного использования**: Программное средство, разработанное для конкретного применения, но с возможностью другого применения, или разработанное специально для многократного использования в различных проектах или для многофункционального использования в одном проекте.

3.50 **производные требования**: Дополнительные требования, появившиеся в результате выполнения процессов разработки ПО, которые не являются непосредственно связанными с требованиями верхнего уровня.

3.51 **процедура тестирования:** Детальные инструкции для того, чтобы генерировать и выполнить множество тестовых наборов и оценить результаты их выполнения.

3.52 **разработка ПО:** Набор действий, результатом выполнения которых являются программные средства. Разработка ПО может включать в себя новую разработку, модификацию, многократное использование, перепроектирование или любое другое действие, требуемое для создания программных средств.

3.53 **решение**: Логическое выражение, состоящее из условий и, возможно, логических операций. Решение без логических операций — это условие. Если условие включено в решение более одного раза, то каждое его вхождение считают отдельным условием.

3.54 **связность по данным:** Зависимость программного компонента от данных, которые используются не только исключительно в этом компоненте.

3.55 **связность по управлению:** Степень влияния одного программного компонента на выполнение другого программного компонента.

3.56 **сертификационное доверие:** Принятие сертифицирующей организацией того факта, что процесс или средство удовлетворяет сертификационным требованиям.

3.57 **система:** Набор аппаратных и программных компонентов, созданный для выполнения определенной функции или множества функций.

3.58 **словарь данных**: Детальное описание данных, параметров, переменных и констант, используемых в системе.

3.59 **совместный просмотр**: Совещание с участием представителей и заказчика и разработчика, в процессе которого проверяют и обсуждают состояние проекта, программные средства и/или проблемы проекта.

3.60 **соискатель:** Человек или организация, претендующая на получение утверждения от сертифицирующей организации.

3.61 **соисполнитель разработки:** Организация, которая не является ни главным подрядчиком, ни субподрядчиком данной разработки, но которая принимает участие в разработке системы или проекта.

3.62 **среда разработки ПО:** Интегрированная система, включающая в себя аппаратные средства, ПО, программно-аппаратные средства, процедуры и документы, необходимые для разработки ПО.

3.63 **среда верификации/тестирования ПО**: Интегрированная система, включающая в себя аппаратные средства, ПО, программно-аппаратные средства, процедуры и документы, необходимые для выполнения верификации/тестирования ПО. Элементами данной среды могут являться имитаторы, статические анализаторы, генераторы тестовых данных, анализаторы путей и т.п., а также элементы, используемые в среде разработки ПО.

3.64 **средства достижения согласования:** Специальные методы, используемые соискателем для удовлетворения требованиям, заданным в сертификационном базисе.

3.65 **статический анализатор**: Программное инструментальное средство, которое позволяет получать определенные характеристики программы без ее выполнения.

3.66 **тестирование**: Процесс выполнения системы или компонента системы в целях проверки того, что она/он удовлетворяет заданным требованиям, и обнаружения ошибок.

3.67 **тестовый набор:** Множество тестовых входных данных, условий выполнения и результатов, разработанных для определенных целей, например для выполнения конкретного пути в программе или для верификации согласованности с заданными требованиями.

3.68 **трассируемость:** Доказательство связи между элементами, например между входной и выходной информацией процесса, между требованием и его реализацией.

3.69 **требование**: Характеристика того, чем система или ЭКПО должны обладать, чтобы быть приемлемыми для заказчика.

3.70 **требования верхнего уровня:** Требования к ПО, разработанные на основании анализа системных требований и требований, связанных с безопасностью системы.

3.71 **требования к ПО**: Описание того, что должно производить ПО, с заданием входных условий и ограничений. Требования к ПО включают в себя как требования верхнего уровня, так и требования нижнего уровня.

3.72 **требования нижнего уровня**: Требования к ПО, разработанные на основании требований верхнего уровня, производных требований и ограничений проекта, по которым исходный код может быть реализован непосредственно, без какой-либо дополнительной информации.

3.73 **управление конфигурацией:** Процесс идентификации и обеспечения целостности элементов конфигурации системы.

3.74 **условие:** Логическое выражение, не содержащее логических операций.

3.75 **устойчивость к ошибкам входных данных:** Свойство, благодаря которому ПО может продолжать корректно выполняться, несмотря на ошибки входных данных.

3.76 **файл разработки ПО:** Сохраняемая совокупность данных, необходимых для разработки конкретного ПО. Содержит обычно (либо непосредственно, либо путем ссылок) сведения, связанные с анализом требований, проектированием и реализацией; информацию о тестировании, проводимом разработчиком, а также план и информацию о состоянии разработки.

3.77 **элемент конфигурации аппаратуры (ЭКА):** Совокупность компонентов аппаратных средств, которая обеспечивает конечную функцию использования и предназначается заказчиком для независимого от других элементов управления конфигурацией.

3.78 **элемент конфигурации ПО (ЭКПО)**: Совокупность компонентов ПО, которая обеспечивает

конечную функцию использования и предназначается заказчиком для независимого от других элементов управления конфигурацией.

3.79 **эмулятор:** Устройство, компьютерная программа или система, которая принимает те же входные данные и производит те же выходные данные, что и данная система, и использующая тот же объектный код. Предназначен для выполнения на одном компьютере программ, написанных для другого компьютера.

4 Общие требования

4.1 Жизненный цикл ПО

4.1.1 Процессы жизненного цикла ПО

В настоящем стандарте рассмотрены следующие процессы жизненного цикла ПО:

Процесс планирования, который определяет и координирует действия процессов разработки и интегральных процессов для данного проекта (раздел 6).

Процессы разработки, в ходе выполнения которых создается программное средство. Этими процессами являются:

- процесс определения требований к ПО;
- процесс проектирования ПО;
- процесс кодирования ПО;
- процесс интеграции.

Процессы разработки описаны в разделе 7.

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

- процесс верификации ПО;
- процесс управления конфигурацией ПО;
- процесс обеспечения качества ПО;
- процесс сертификационного сопровождения.

Интегральные процессы выполняются одновременно с процессами разработки ПО (разделы 8—11).

4.1.2 Установление модели жизненного цикла ПО

В рамках конкретного проекта создания ПО должны быть установлены одна или несколько моделей жизненного цикла ПО, в соответствии с которыми выбирают необходимые работы для каждого процесса, определяют последовательность их выполнения, назначают ответственных за выполнение работ.

Для конкретного проекта последовательность процессов определяется сложностью проекта, функциональными возможностями разрабатываемой системы, объемом и сложностью ПО, стабильностью требований, использованием ранее полученных результатов, стратегией разработки и возможностями аппаратных средств. Обычная последовательность процессов разработки ПО — определение требований, проектирование, кодирование и интеграция.

Порядок представления в настоящем стандарте процессов и отдельных видов работ внутри процессов не предназначен для определения последовательности их выполнения в конкретном проекте. Многие работы могут быть выполнены одновременно; разные программные средства могут поступать (находиться) на разных этапах разработки. Если ПО разрабатывают для нескольких построений, некоторые работы могут быть выполнены для каждого построения, другие же — только для отдельного построения. Проект, включающий в себя одно построение, должен выполнять все требуемые для данного построения работы.

4.1.3 Критерии перехода между процессами

Критерии перехода используют для определения возможности первичного или повторного перехода к выполнению процессов. Каждый процесс жизненного цикла ПО выполняет некоторые виды работ над исходными данными с целью получения результирующих данных. Процесс может иметь обратную связь с другими, ранее выполненными процессами и, в свою очередь, получать обратную связь от тех процессов, которые будут выполнены позже. Под обратной связью понимают получение, распознавание и обработку информации, которая требует повторной активизации ранее выполненного процесса. Примером обратной связи может служить получение сообщения об ошибке. Критерии перехода зависят от запланированной последовательности выполнения процессов разработки ПО и интегральных процессов, а также от уровня ПО. Возможные примеры критериев перехода: выполнение верификационного просмотра выходных результатов; получение в качестве входных данных идентифицированного элемента конфигурации; выполнение анализа трассируемос-

ти для входных данных. Процесс может быть инициирован только после того, как получены все исходные данные для этого процесса и удовлетворен критерий перехода, установленный для этого процесса.

4.2 Общие требования для разработки ПО

4.2.1 Методы разработки ПО

Разработчик должен использовать для всех работ по созданию ПО систематизированные, зарегистрированные методы. План разработки ПО должен содержать описание этих методов или включать в себя ссылки на источники, в которых они описаны.

4.2.2 Стандарты ПО

Разработчик должен разработать и использовать стандарты для представления требований, проекта, кода, тестовых вариантов, тестовых процедур и результатов тестирования. План разработки ПО должен содержать описание этой информации или ссылки на источники, в которых они описаны.

4.2.3 Программные средства многократного использования

Разработчик должен рассмотреть и оценить возможность применения ранее разработанных программных средств многократного использования для выполнения требований контракта. Область исследования и критерии, используемые для оценки, должны быть описаны в Плане разработки ПО. Выбранные для применения программные средства должны отвечать требованиям контракта по правам собственности.

Разработчик должен рассмотреть возможность многократного использования программных средств, разработанных по контракту, оценить и идентифицировать для заказчика выгоды и издержки такого использования в случае его совместимости с задачами проекта.

Примечание — В контракт может быть включено требование на разработку программных средств, пригодных для многократного использования.

4.2.4 Отработка критических требований

Разработчик должен идентифицировать ЭКПО или части их, критические с точки зрения безопасности, сбой в которых может привести к отказной ситуации для системы (см. 5.2).

Разработчик должен идентифицировать ЭКПО или их части, критические с точки зрения защиты, сбой в которых может привести к нарушению защиты системы. Если имеется такое ПО, разработчик должен предусмотреть стратегию обеспечения защиты. Эта стратегия должна гарантировать, что требования, проект, реализация и эксплуатационные процедуры для идентифицированного ПО минимизируют или устраняют потенциальные нарушения защиты системы. Разработчик должен описать стратегию в Плане разработки ПО, реализовать стратегию и провести доказательство как в части требуемых программных средств, так и в части выполнения стратегии обеспечения защиты.

Разработчик должен идентифицировать ЭКПО или части их, критические с точки зрения секретности, сбой в которых может привести к нарушению секретности системы. Если имеется такое ПО, то разработчик должен представить стратегию обеспечения секретности. Стратегия должна гарантировать, что требования, проект, реализация и эксплуатационные процедуры для идентифицированного ПО минимизируют или устраняют потенциальные нарушения секретности системы. Разработчик должен описать стратегию в Плане разработки ПО, реализовать стратегию и провести доказательство как в части требуемых программных средств, так и в части выполнения стратегии обеспечения секретности.

В случаях, когда система возлагает на ПО реализацию каких-либо требований, которые в соответствии с контрактом или спецификациями системы считаются критическими, разработчик должен идентифицировать те ЭКПО или их части, сбой в которых может привести к нарушению этих критических требований; разработать стратегию для гарантирования того, что требования, проект, реализация и эксплуатационные процедуры для идентифицированного ПО минимизируют или устраняют потенциал для таких нарушений; описать стратегию в Плане разработки ПО; выполнить стратегию и провести доказательство как в части требуемых программных средств, так и в части выполнения стратегии.

4.2.5 Использование ресурсов аппаратных средств компьютера

Разработчик должен проанализировать требования контракта, относящиеся к использованию ресурсов аппаратных средств компьютера (например, максимально возможная производительность процессора, объем памяти, пропускная способность устройств ввода/вывода). Разработчик должен распределить аппаратные ресурсы компьютера между ЭКПО, контролировать использование этих

ресурсов при выполнении контракта и перераспределить их или идентифицировать потребность в дополнительных ресурсах по мере необходимости, чтобы удовлетворить требования контракта.

4.2.6 Доступ для проверки заказчиком

Разработчик должен обеспечить заказчику или его полномочному представителю доступ к средствам разработчика и субподрядчика, включая среды разработки и верификации ПО, для проверки программных средств и работ, требуемых в соответствии с контрактом.

5 Системные аспекты, связанные с разработкой ПО

Процесс обеспечения безопасности определяет информационный поток между процессами жизненного цикла системы управления и процессами жизненного цикла ПО. Вследствие взаимозависимости процесса обеспечения безопасности системы и процесса проектирования системы поток информации, описанный в следующих подразделах, является итерационным.

5.1 Поток информации между процессами жизненного цикла системы и ПО

5.1.1 Информационный поток от системных процессов к процессам ПО

В процессе оценки безопасности системы должны быть определены возможные отказные ситуации для системы и установлены их категории, определены требования, связанные с безопасностью, которые специфицируют желаемую отказоустойчивость и реакцию системы на отказные ситуации.

Требования, связанные с безопасностью, — это часть системных требований, которые являются входной информацией для процессов жизненного цикла ПО. Для гарантии правильной реализации требований, связанных с безопасностью, системные требования должны содержать (или ссылаются на):

- описание системы и определение аппаратуры;
- системные требования, относящиеся непосредственно к ПО, включая функциональные требования, требования по эффективности и требования, связанные с безопасностью;
- уровень(ни) ПО и информацию, подтверждающую их определение, отказные ситуации, их категории и функции, выполняемые ПО;
- стратегии обеспечения безопасности и ограничения проекта, включая методы проектирования, такие как использование разбиения, многоверсионного неидентичного ПО, избыточности или мониторинга безопасности.

Процессы жизненного цикла системы могут также определять требования к процессам жизненного цикла ПО, которые необходимы для поддержки верификации системы.

5.1.2 Информационный поток от процессов ПО к системным процессам

Процесс оценки безопасности системы должен определить влияние проектирования и реализации ПО на безопасность системы в целом, используя информацию, создаваемую процессами жизненного цикла ПО. Эта информация включает в себя идентификацию областей распространения отказов, требования к ПО, архитектуру ПО и источники ошибок, которые могут быть обнаружены или исключены посредством специальной организации архитектуры ПО или путем использования инструментальных средств, или другими методами, используемыми в процессе проектирования ПО. Для процесса оценки безопасности системы должна быть обеспечена трассируемость между системными требованиями и результатами проектирования ПО.

Изменения, внесенные при модификации ПО, могут воздействовать на безопасность системы и, следовательно, также должны быть идентифицированы для оценки безопасности системы.

5.2 Отказные ситуации и уровни ПО

Категорию отказной ситуации системы устанавливают, определяя серьезность проявления отказной ситуации на объекте управления. Ошибка в ПО может вызвать сбой, который приведет к отказной ситуации. Следовательно, необходимый для безопасного функционирования уровень ПО определяют исходя из отказных ситуаций системы.

5.2.1 Классификация отказных ситуаций

Категория А — катастрофическая: отказная ситуация, которая препятствует безопасному функционированию объекта управления.

Категория В — опасная/критическая: отказная ситуация, которая приводит к уменьшению возможностей объекта управления или способности персонала справиться с неблагоприятными эксплуатационными режимами, при которых возникают:

- большое снижение гарантийных резервов или функциональных возможностей;
- крайне тяжелое положение или перегрузки, которые могут вызвать неточное или неполное выполнение задачи;

- неблагоприятные или потенциально фатальные воздействия для окружающей среды.

Категория С — существенная: отказная ситуация, которая приводит к снижению возможностей объекта управления или способности персонала справиться с неблагоприятными эксплуатационными режимами, при продолжении которых могут возникать, например, большое снижение гарантийных резервов или функциональных возможностей, перегрузки или условия, вызывающие ухудшение работоспособности персонала.

Категория D — несущественная: отказная ситуация, незначительно уменьшающая безопасность объекта и требующая действий персонала, которые осуществимы в пределах их возможностей. Несущественная отказная ситуация может включать в себя, например, незначительное уменьшение гарантийных резервов или функциональных возможностей, незначительное увеличение рабочей нагрузки персонала или некоторое неудобство для персонала.

Категория Е — невлияющая: отказная ситуация, которая не воздействует на эксплуатационные возможности объекта управления или не увеличивает рабочую нагрузку персонала.

5.2.2 Определения уровня ПО

Уровень ПО определяется возможностью возникновения потенциальных отказных ситуаций, выявленных процессом оценки безопасности системы, в результате сбоев в ПО. Уровень ПО означает, что трудозатраты, необходимые для доказательства согласованности с требованиями сертификации, меняются в зависимости от категории отказной ситуации.

Уровень А: ПО, аномальное поведение которого, как показано процессом оценки безопасности системы, вызвало бы (или способствовало бы) возникновение(ю) отказа функционирования системы, приводящее к катастрофической отказной ситуации для объекта управления.

Уровень В: ПО, аномальное поведение которого, как показано процессом оценки безопасности системы, вызвало бы (или способствовало бы) возникновение(ю) отказа функционирования системы, приводящее к опасной/критической отказной ситуации для объекта управления.

Уровень С: ПО, аномальное поведение которого, как показано процессом оценки безопасности системы, вызвало бы (или способствовало бы) возникновение(ю) отказа функционирования системы, приводящее к существенной отказной ситуации для объекта управления.

Уровень D: ПО, аномальное поведение которого, как показано процессом оценки безопасности системы, вызвало бы (или способствовало бы) возникновение(ю) отказа функционирования системы, приводящее к несущественной отказной ситуации для объекта управления.

Уровень Е: ПО, аномальное поведение которого, как показано процессом оценки безопасности системы, вызвало бы (или способствовало бы) возникновение(ю) отказа функционирования системы, не влияющее на эксплуатационные возможности объекта и работоспособность персонала. Если для ПО был установлен сертифицирующей организацией уровень Е, то в дальнейшем для сертификации такого ПО никакие требования данного документа не являются обязательными.

5.2.3 Назначение уровня ПО

Первоначально процесс оценки безопасности системы присваивает уровень (ни) ПО, соответствующий(ие) компонентам ПО конкретной системы. При проведении данного назначения учитывают воздействие отказов как потери функции или неправильного функционирования.

Примечание — Если систему или ЭКПО разрабатывают для нескольких построений, то компоненту ПО данного построения может быть назначен более высокий уровень, чем требуется процессом оценки безопасности системы, если использование этого компонента в других построениях требует такого уровня.

Если аномальное поведение компонента ПО вызвано более чем одной отказной ситуацией, уровень ПО для данного компонента ПО определяет наиболее серьезная категория отказной ситуации. Существуют архитектурные стратегии (см. 5.5), использование которых в процессе проектирования системы может привести к изменению назначенного уровня ПО.

Системная функция может быть реализована одним или несколькими компонентами ПО. Параллельная реализация — это такая реализация, когда функция системы реализуется несколькими компонентами ПО. Тогда для возникновения отказной ситуации требуется аномальное поведение более чем одного компонента. При параллельной реализации по крайней мере один компонент ПО будет иметь уровень ПО, связанный с наиболее серьезной категорией отказной ситуации этой функции системы. Уровень ПО для других компонентов определяют, используя категорию отказной ситуации, связанную с потерей этой функции. Примеры таких реализаций описаны в 5.5.2 и 5.5.3.

Последовательная реализация — это такая реализация, когда несколько компонентов ПО используются для реализации функции системы так, что аномальное поведение любого из компо-

нентов может привести к отказной ситуации. При такой реализации все компоненты ПО будут иметь уровень ПО, связанный с наиболее серьезной категорией отказной ситуации этой функции системы.

Разработка ПО с заданным уровнем не подразумевает оценку интенсивности отказов для этого ПО. Таким образом, уровни ПО или оценки надежности ПО, основанные на уровнях ПО, не могут использоваться процессом оценки безопасности системы в отличие от использования интенсивности отказов аппаратуры.

5.3 Анализ системных требований

Разработчик должен принимать участие в анализе требований к системе. Если систему разрабатывают для нескольких различных построений, ее требования не могут быть полностью определены до завершения конечного построения. В этом случае разработчик должен идентифицировать подмножество требований системы, которые будут определены в каждом построении, и подмножество, которое будет реализовано в каждом из построений. Анализ требований к системе для данного построения следует интерпретировать так, чтобы определять требования к системе, идентифицированные для данного построения.

5.3.1 Анализ информации о потребностях пользователя

Разработчик должен принимать участие в анализе обеспечиваемой заказчиком информации, необходимой для достижения понимания потребностей пользователя. Эта информация может быть представлена в форме предложений, обзоров, сообщений о дефектах/изменениях, обратной связи к прототипам, интервью о потребностях пользователя или любой другой форме.

5.3.2 Эксплуатационная концепция

Разработчик должен принимать участие в определении и документировании эксплуатационной концепции для системы. Результат данной работы должен быть представлен в качестве документа «Описание эксплуатационной концепции» (12.32).

5.3.3 Требования к системе

Разработчик должен принимать участие в определении и документировании требований, которым должна удовлетворять система, и методов, которые необходимо использовать в целях гарантирования выполнения каждого требования. Результат данной работы должен быть представлен в качестве документа «Спецификация системы/подсистемы» (12.12). В зависимости от условий контракта требования относительно интерфейсов системы могут быть включены в Спецификацию системы/подсистемы или в Спецификацию требований к интерфейсу (12.14).

Если система состоит из подсистем, то предполагают, что работы, указанные в 5.3.3, будут выполнены итеративно с работами, указанными в 5.4 (проектирование системы) для определения требований к системе, проектирования системы и идентификации ее подсистем, определения требований к этим подсистемам, проектировании подсистем, идентификации их компонентов и т.д.

5.3.4 Модифицируемое пользователем ПО. ПО с возможностью выбора вариантов. Коммерчески доступное ПО.

Если системные требования предусматривают возможность модификации, осуществляемой пользователем, то пользователи могут изменять ПО в заданном диапазоне без рассмотрения, осуществляемого сертифицирующей организацией. В этом случае системные требования должны определить механизмы, которые устраняют влияние на безопасность системы осуществляемой пользователем модификации независимо от того, как она выполнена. ПО, обеспечивающее защиту, должно иметь уровень, такой как у функции, которую оно защищает от ошибок в модифицируемом компоненте. При проведении модификации пользователем последний должен нести ответственность за все аспекты модифицируемого им ПО, например управление конфигурацией, обеспечение качества и верификацию.

Системные требования к ПО с возможностью выбора вариантов должны определить средства для гарантии того, что не может быть сделан несанкционированный выбор.

Коммерчески доступное ПО, включаемое в состав системы, должно удовлетворять требованиям данного документа. Если существуют несоответствия в документах жизненного цикла коммерчески доступного ПО, то информация, представленная в них, должна быть расширена, чтобы удовлетворить требования данного документа.

5.3.5 Системные требования для ПО, загружаемого в полевых условиях

Загружаемое в полевых условиях прикладное ПО — программный продукт или таблицы данных, которые могут быть загружены без демонтажа системы или оборудования. Требования, определяемые безопасностью, связанные с функцией загрузки данных ПО, являются частью системных требований. Если нечеткое представление функции загрузки данных ПО может вызвать отказную ситуацию в системе, то требования, связанные с обеспечением безопасности для функции

загрузки ПО, должны быть определены в системных требованиях. Требования обеспечения безопасности системы для ПО, загружаемого в полевых условиях, следующие:

- обнаружение разрушенного или частично загруженного ПО;
- обнаружение загрузки несоответствующей конфигурации ПО;
- совместимость аппаратных и программных средств;
- совместимость компонентов ПО;
- совместимость типа объекта и ПО;
- отображение потери или искажения идентификации конфигурации ПО.

Требования к ПО, загружаемому в полевых условиях:

- если процессом оценки безопасности системы не определено специально, то механизму обнаружения разрушенного или частично загруженного ПО должна быть установлена такая же категория отказной ситуации или уровень ПО, как наиболее серьезной отказной ситуации или наиболее высокому уровню ПО, связанному с функцией, которая использует загрузку ПО;
- если для системы определен режим, заданный по умолчанию, в случае, когда загружено несоответствующее ПО или данные, то для каждого выделенного компонента системы должны быть указаны требования по обеспечению безопасности, определяющие действия в этом режиме;
- механизм загрузки ПО должен включать в себя программные и/или аппаратные средства для обнаружения некорректно загруженного ПО и обеспечивать защиту, определяемую отказной ситуацией;
- если ПО представляет собой часть встроенного бортового механизма отображения, являющегося средством гарантии того, что объект соответствует сертифицированной конфигурации, то такое ПО должно быть либо разработано как ПО с самым высоким уровнем из того, которое должно быть загружено, либо процесс оценки обеспечения безопасности системы должен подтверждать целостность конфигурации ПО.

5.3.6 Анализ системных требований при верификации ПО

Системные требования разрабатывают на основе эксплуатационных требований к системе и требований, связанных с обеспечением безопасности, которые являются выходным результатом процесса оценки безопасности системы.

В системных требованиях для прикладного ПО устанавливают следующие характеристики ПО:

- ПО должно выполнять специфицированные функции, как определено системными требованиями;
- ПО не должно проявлять аномального поведения, не определяемого процессом оценки безопасности системы. Должны быть сформулированы дополнительные системные требования для обработки возможного аномального поведения.

Системные требования должны быть переработаны затем в требования верхнего уровня к ПО, которые проверяются работами процесса верификации ПО.

5.3.7 Анализ ПО при верификации системы

Требования по выполнению верификации системы выходят за область применения настоящего стандарта. Однако процессы жизненного цикла ПО поддерживают процесс верификации системы и взаимодействуют с ним. Детали проектирования ПО, касающиеся функциональных возможностей системы, должны быть доступными при выполнении верификации системы. Верификация системы может обеспечивать значительное покрытие структуры кода. Для достижения критериев тестового покрытия, описанных в Плане верификации ПО, может быть использован анализ покрытия ПО тестами верификации системы.

5.4 Проектирование системы

Разработчик должен принимать участие в проектировании системы. Если систему разрабатывают для нескольких различных построений, то ее проект не может быть полностью определен до завершения всех построений. Разработчик должен идентифицировать части проекта системы, которые будут определены в каждом построении.

5.4.1 Проектные решения системного уровня

Разработчик должен принимать участие в определении и документировании проектных решений системного уровня (таких, как решения, относящиеся к проектированию режимов работы системы, и решения, влияющие на выбор и проектирование компонентов системы).

Результаты должны быть включены в раздел проектных решений системного уровня документа «Описание проекта системы/подсистемы» (12.15). В зависимости от условий контракта часть проекта, имеющая отношение к интерфейсам, может быть включена в Описание проекта системы/подсистемы или в Описание проекта интерфейса (12.17), а часть проекта, имеющая отношение

к базам данных, — в Описание проекта системы/подсистемы или в Описание проекта базы данных (12.18).

Примечание — Проектные решения являются прерогативой разработчика, если они формально не преобразованы в требования в процессе выполнения контракта. Разработчик ответствен за выполнение всех требований и демонстрацию этого выполнения посредством квалификационного тестирования (8.5.4). Реализация проектных решений, действующих как «внутренние требования» разработчика, должна быть подтверждена внутренним тестированием разработчика, выполнение которого нет необходимости демонстрировать заказчику.

5.4.2 Проектирование архитектуры системы

Разработчик должен участвовать в определении и документировании проекта архитектуры системы (идентификации компонентов системы, их интерфейсов и концепции их совместного выполнения) и прослеживании соответствия между компонентами системы и системными требованиями. Результат этих работ должен быть включен в документ «Описание проекта системы/подсистемы» (12.15). В зависимости от условий контракта часть проекта, имеющая отношение к интерфейсам, может быть включена в Описание проекта системы/подсистемы или в Описание проекта интерфейса (12.17).

5.5 Стратегии архитектурного проектирования системы

В процессе оценки безопасности системы устанавливают, как архитектурное проектирование системы предотвращает аномальное поведение ПО при появлении отказных ситуаций для системы. Уровень ПО назначают в соответствии с наиболее серьезной категорией возможных отказных ситуаций. Далее описаны некоторые архитектурные стратегии, которые позволяют ограничивать воздействие ошибок, обнаруживать ошибки и обеспечивать приемлемую реакцию системы для устранения их воздействия. Эти архитектурные стратегии не следует рассматривать как предпочтительные или обязательные.

5.5.1 Разбиение

Стратегию разбиения применяют для обеспечения изоляции между функционально независимыми компонентами ПО, чтобы предотвратить и/или изолировать дефекты и потенциально уменьшить трудозатраты процесса верификации ПО. Если с помощью разбиения обеспечивают защиту от ошибок, то уровень ПО для каждого полученного при разбиении компонента следует назначать в соответствии с наиболее серьезной категорией отказной ситуации, связанной с этим компонентом.

5.5.2 Многоверсионное неидентичное ПО

Многоверсионное неидентичное ПО является стратегией проектирования, которая предусматривает создание двух или более компонентов ПО для реализации одной и той же функции способами, исключающими источники общих ошибок в нескольких компонентах. Вместо термина многоверсионное неидентичное ПО могут быть использованы также термины многоверсионное ПО, неидентичное ПО, N-версионное ПО или разнесенная разработка ПО.

Конфигурация аппаратуры, которая обеспечивает выполнение многоверсионного неидентичного ПО, должна быть определена в системных требованиях. Степень неидентичности и, следовательно, степень защиты обычно не измеряют.

5.5.3 Мониторинг безопасности

Мониторинг безопасности применяют как средство защиты от конкретных отказных ситуаций с помощью прямого мониторинга функций, которые могут привести к отказной ситуации. Функции мониторинга могут быть реализованы аппаратными средствами, программными средствами или комбинацией аппаратных и программных средств.

Использование методов мониторинга безопасности может понизить уровень ПО, выполняющего функцию контроля, до уровня, связанного с потерей реализуемой данным ПО функции системы. Существуют три важных параметра мониторинга, которые должны быть определены, чтобы обеспечить снижение уровня:

- уровень ПО: ПО, которое осуществляет мониторинг безопасности, предписывается уровень, связанный с наиболее серьезной категорией отказной ситуации для контролируемой функции;
- покрытие отказов системы: оценка покрытия отказов системы с помощью мониторинга безопасности гарантирует, что проект монитора и его реализация таковы, что отказы, которые предполагается обнаружить, будут обнаружены при всех возможных условиях;
- независимость функции и монитора: монитор и защитный механизм не должны активизироваться теми же самыми отказными ситуациями, которые вызывают опасность.

5.6 Библиотека разработки ПО

Разработчик должен создать, контролировать и сопровождать Библиотеку разработки ПО, чтобы обеспечить упорядоченную разработку и последующую поддержку ПО. Библиотека разработ-

ки ПО может быть частью среды разработки ПО и среды верификации. Разработчик должен сопровождать Библиотеку разработки ПО на протяжении действия контракта.

5.7 Файлы разработки ПО

Разработчик должен создать, контролировать и сопровождать файлы разработки ПО для каждого модуля ПО или логически связанной группы модулей ПО, для каждого ЭКПО и, если применимо, для логических групп ЭКПО, для подсистем и для полной системы. Разработчик должен записывать информацию относительно разработки ПО в соответствующем файле разработки ПО и должен сопровождать файл разработки ПО на протяжении действия контракта.

5.8 Непоставляемое ПО

Разработчик может использовать непоставляемое ПО в разработке поставляемого ПО в том случае, если функционирование и поддержка поставляемого ПО после поставки заказчику не зависят от непоставляемого ПО или предусмотрены меры, гарантирующие, что заказчик имеет или может получить то же самое ПО. Разработчик должен гарантировать, что все непоставляемое ПО, используемое в проекте, выполняет предназначенные функции.

5.9 Подготовка к использованию ПО

Разработчик должен запланировать, какое конкретно ПО поставляется пользователю в рамках каждого построения, и объем устанавливаемого ПО (например, полная установка или установка только части, выбранной заказчиком). Подготовка к использованию ПО в каждом построении должна включать в себя все действия, необходимые для выполнения Плана установки для этого построения.

5.9.1 Подготовка исполняемого ПО

Разработчик должен подготовить к установке на каждом рабочем месте пользователя исполняемое ПО, включая все файлы пакетного режима, командные файлы, файлы данных или другие файлы ПО, необходимые для установки и эксплуатации ПО на объектном компьютере. Описания всех требуемых элементов должны быть включены в раздел, посвященный исполняемому ПО документа «Спецификация программного средства» (12.27).

5.9.2 Подготовка описания версии для пользователя

Разработчик должен идентифицировать и зарегистрировать каждую версию ПО, предназначенную для конкретного пользователя. Вся необходимая информация должна быть включена в документ «Описание версии ПО» (12.39).

5.9.3 Подготовка руководств пользователя

Разработчик должен подготовить следующие руководства пользователя:

- Руководство пользователя ПО. В данном руководстве разработчик должен идентифицировать и зарегистрировать информацию, необходимую для работы пользователям ПО (людям, которые и эксплуатируют ПО, и используют результаты его работы). Вся информация должна быть включена в документ «Руководство пользователя ПО» (12.38).

- Руководство по входной/выходной информации ПО. В данном руководстве разработчик должен идентифицировать и зарегистрировать информацию, необходимую тем, кто будет формировать входные данные и получать выходные данные при эксплуатации ПО в компьютерном центре или другой централизованной или сетевой установке ПО. Вся информация должна быть включена в документ «Руководство по входной/выходной информации ПО» (12.37).

- Руководство оператора ПО. В данном руководстве разработчик должен идентифицировать и зарегистрировать информацию, необходимую для эксплуатации ПО в компьютерном центре или другой централизованной или сетевой установке ПО. Вся информация должна быть включена в документ «Руководство оператора ПО» (12.36).

- Руководство по эксплуатации компьютера. В данном руководстве разработчик должен идентифицировать и зарегистрировать всю информацию, необходимую для эксплуатации компьютера, на котором будет выполнено ПО. Эта информация должна быть включена в документ «Руководство по эксплуатации компьютера» (12.33).

Примечание — Не все перечисленные руководства будут необходимы для каждой системы. Заказчик на основании информации, полученной от разработчика, должен определить, какие руководства являются необходимыми для данной системы, и требовать разработки только этих руководств. Все документы допускают замену на существующие коммерческие или другие руководства, которые содержат требуемую информацию. Руководства, перечисленные в 5.9.3, обычно разрабатываемые параллельно с разработкой ПО, должны быть готовы для использования при тестировании ЭКПО.

5.9.4 Установка на рабочих местах пользователя

Разработчик должен:

- установить исполняемое ПО и проверить функционирование всех его режимов, определенных в контракте, на рабочих местах пользователя;
- обеспечить обучение пользователей в соответствии с контрактом;
- обеспечить другую необходимую помощь пользователям в соответствии с контрактом.

5.10 Подготовка к передаче ПО

Разработчик должен идентифицировать ПО, передаваемое агентству поддержки, в составе каждого построения. Подготовка к передаче ПО каждого построения должна включать в себя действия, определенные Планами передачи данного построения.

5.10.1 Подготовка исполняемого ПО

Разработчик должен подготовить исполняемое ПО для передачи в организацию, осуществляющую поддержку, включая файлы пакетного режима, командные файлы, файлы данных или другие файлы ПО, необходимые для установки и эксплуатации ПО на объектном компьютере. Вся необходимая информация должна быть включена в раздел документа «Спецификация программного средства» (12.27).

5.10.2 Подготовка исходных файлов

Разработчик должен подготовить исходные файлы, которые должны быть переданы в организацию, осуществляющую поддержку: файлы пакетного режима, командные файлы, файлы данных и другие файлы ПО, необходимые для регенерации исполняемого ПО. Вся необходимая информация должна быть включена в раздел, описывающий исходные файлы, документа «Спецификация программного средства» (12.27).

5.10.3 Подготовка описания версии для организации, осуществляющей поддержку

Разработчик должен идентифицировать и зарегистрировать версию ПО для организации, осуществляющей поддержку. Вся необходимая информация должна быть включена в документ «Описание версии ПО» (12.39).

5.10.4 Подготовка проекта ЭКПО для построения ПО и связанной с ним информации

Разработчик должен модифицировать описание проекта каждого ЭКПО, чтобы оно соответствовало конкретному построению ПО, и должен определить и зарегистрировать следующее: методы, которые нужно использовать, чтобы проверить копии ПО; использование аппаратных ресурсов компьютера для ЭКПО; другую информацию, необходимую для поддержки ПО; прослеживание соответствия между исходными файлами ЭКПО и модулями ПО и между объемами используемых аппаратных ресурсов компьютера и требованиями ЭКПО относительно них. Все результаты должны быть включены в разделы, посвященные аттестации, поддержке ПО и прослеживанию соответствия, документа «Спецификация программного средства» (12.27).

5.10.5 Модификация описания проекта системы

Разработчик должен участвовать в модификации описания проекта системы в соответствии с конкретным построением системы. Все результаты должны быть включены в документ «Описание проекта системы/подсистемы» (12.15).

5.10.6 Подготовка руководств поддержки

Разработчик должен подготовить следующие руководства поддержки:

- Руководство по программированию для компьютера. Разработчик должен идентифицировать и зарегистрировать информацию, необходимую для программирования на компьютерах, на которых будет создаваться и выполняться ПО. Вся необходимая информация должна быть включена в документ «Руководство по программированию для компьютера» (12.34).
- Руководство поддержки программно-аппаратных средств. Разработчик должен идентифицировать и зарегистрировать информацию, необходимую для программирования и перепрограммирования программно-аппаратных устройств. Вся необходимая информация должна быть включена в документ «Руководство поддержки программно-аппаратных средств» (12.35).

Примечание — Перечисленные руководства не являются необходимыми для всех систем. Заказчик на основании данных, полученных от разработчика, должен определить, какие руководства являются необходимыми для данной системы, и требовать разработки только этих руководств. Все документы допускают замену на коммерческие или другие руководства, которые содержат требуемую информацию. Перечисленные руководства дополняют Описание проекта системы/подсистемы и Спецификации программного средства, которые служат как основные источники информации для поддержки ПО. Руководства пользователя, перечисленные в 5.9.3, также полезны для персонала, осуществляющего поддержку.

5.10.7 Передача организации, осуществляющей поддержку

Разработчик должен:

- установить и проверить поставляемое ПО в среде поддержки, обозначенной в контракте;
- продемонстрировать заказчику возможность регенерации (компиляции/редактирования связей/загрузки) и сопровождения поставляемого ПО с использованием коммерчески доступного, находящегося в собственности у заказчика или поставляемого по контракту ПО и аппаратных средств, указанных в контракте или одобренных заказчиком;
- обеспечить обучение персонала организации, осуществляющей поддержку, в соответствии с контрактом;
- обеспечить любую иную помощь организации, осуществляющей поддержку, в соответствии с контрактом.

5.11 Совместные технические и административные просмотры

5.11.1 Совместные технические просмотры

Разработчик 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 ПО. Время и место проведения совместных просмотров должны быть запланированы разработчиком и одобрены заказчиком. Назначение совместных технических просмотров:

- просмотр и оценка состояния разработки ПО;
- анализ и оценка предложенных технических решений;
- рассмотрение критических для выполнения контракта ситуаций, связанных с техническими, стоимостными и временными аспектами;
- достижение согласованных стратегий предотвращения критических ситуаций в рамках предоставленных полномочий;
- идентификация проблем, которые будут рассмотрены на совместных административных просмотрах;
- гарантия постоянной связи между заказчиком и техническим персоналом разработчика.

5.11.2 Совместные административные просмотры

Разработчик должен принимать участие в совместных с заказчиком административных просмотрах, проводимых в течение периода выполнения контракта. В этих просмотрах как со стороны разработчика, так и со стороны заказчика должны принимать участие лица, обладающие полномочиями для принятия решений о стоимостных и временных затратах. Назначение совместных административных просмотров:

- информирование администрации разработчика и заказчика относительно состояния проекта, о выбранных направлениях, о достигнутых технических соглашениях и общем состоянии разработки ПО;
- разрешение проблем, которые не могли быть решены во время совместных технических просмотров;
- достижение согласованных стратегий предотвращения критических ситуаций, которые не могли быть выработаны во время совместных технических просмотров;
- идентификация и решение проблем административного уровня и критических ситуаций, не рассмотренных во время совместных технических просмотров;
- получение заключения и одобрения заказчика, необходимого для своевременного выполнения проекта.

5.12 Другие действия

5.12.1 Контроль критических ситуаций

Разработчик должен осуществлять контроль за критическими для выполнения контракта ситуациями, которые могут возникнуть во время разработки ПО. Разработчик должен выявить, идентифицировать и проанализировать потенциальные технические, стоимостные или временные критические ситуации; разработать стратегии для предотвращения или устранения таких ситуаций; зарегистрировать возможные критические ситуации и стратегии их предотвращения в Плане разработки ПО и реализовать эти стратегии в соответствии с Планом.

5.12.2 Показатели управления разработкой ПО

Разработчик должен использовать показатели управления разработкой ПО для поддержки управления процессом разработки ПО и уведомления заказчика о состоянии разработки. Разработчик должен идентифицировать данные, необходимые для определения показателей, методы, которые нужно использовать для интерпретации и применения этих данных, и механизм регистрации.

Разработчик должен включить эту информацию в План разработки ПО и использовать показатели управления разработкой в соответствии с Планом.

5.12.3 Защита и секретность

Разработчик должен удовлетворять требования защиты и секретности, определенные в контракте.

5.12.4 Управление субподрядчиком

Если в проекте принимают участие субподрядчики, разработчик должен включить в контракт все договорные требования, необходимые для гарантии, что ПО будет разработано в соответствии с требованиями контракта.

5.12.5 Связь с агентством независимой верификации ПО

Разработчик должен поддерживать постоянную связь с агентством независимой верификации ПО, если это определено в контракте.

5.12.6 Координация действий с соисполнителями

Разработчик должен координировать действия соисполнителей, рабочих групп и групп связи в соответствии с контрактом.

5.12.7 Изменения в выполнении процессов проекта

Разработчик должен периодически оценивать процессы жизненного цикла ПО, используемые в данном проекте, для определения их пригодности и эффективности. Основываясь на этих оценках, разработчик должен идентифицировать любые необходимые и полезные изменения в выполнении процессов, идентифицировать эти изменения для заказчика в форме предлагаемых модификаций к Плану разработки ПО и в случае их одобрения должен реализовать эти изменения в проекте.

6 Процесс планирования ПО

6.1 Цели процесса планирования ПО

Назначение процесса планирования ПО состоит в том, чтобы определить методы создания такого ПО, которое позволит реализовать системные требования и обеспечить уровень качества, соответствующий требованиям настоящего стандарта. Таблица А.1 содержит резюме целей и результатов процесса планирования ПО в зависимости от уровня ПО.

Цели процесса планирования ПО:

а) определить конкретные виды работ процессов разработки и интегральных процессов жизненного цикла, которые позволят реализовать системные требования и создать ПО требуемого уровня (6.2);

б) определить модели жизненного цикла ПО, включающие в себя описание взаимосвязей между процессами, последовательность их выполнения, механизмы обратной связи и критерии перехода (4.1);

в) выбрать среду поддержки жизненного цикла, включающую в себя методы и инструментальные средства, которые нужно использовать для выполнения работ в каждом процессе жизненного цикла (6.4);

г) в случае необходимости рассмотреть дополнительные аспекты разработки, обсуждаемые в разделе 13;

д) определить стандарты разработки, позволяющие обеспечить требования по безопасности системы в части разрабатываемого ПО (6.5);

е) разработать документы процесса планирования ПО в соответствии с 6.3 и разделом 12;

ж) координировать разработку и изменение планов ПО (6.3).

6.2 Состав работ, выполняемых в процессе планирования ПО

В процессе планирования ПО должны быть выполнены следующие работы:

- разработка планов создания ПО и передача их исполнителям, осуществляющим процессы разработки и интегральные процессы (см. требования 11.1);
- определение и выбор стандартов разработки ПО, которые будут использованы для данного проекта;
- выбор методов и инструментальных средств, которые позволят в процессах разработки предотвратить внесение ошибок в ПО;
- обеспечение координации между планами разработки ПО и планами интегральных процессов для получения согласованных стратегий выполнения различных процессов жизненного цикла;
- определение процедуры пересмотра и уточнение планов по мере развития проекта;
- выбор методов и инструментальных средств, позволяющих предотвратить и обнаружить

ошибки и обеспечивающих безопасность системы в случае использования многоверсионного неидентичного ПО;

- если предполагается использование отключенного кода (7.4.3), то в процессе планирования должно быть описано, как отключенный код будет определен, верифицирован и обработан для обеспечения требований безопасности системы;

- если предполагается использовать модифицируемый пользователем код, то в стандартах и планах ПО в соответствии с требованиями 7.2.3 должны быть указаны используемые инструментальные средства и элементы данных;

- процесс планирования считают завершенным, если были выполнены контроль изменений и просмотры для всех планов ПО и стандартов разработки ПО (6.7).

До завершения процесса планирования могут быть инициированы другие процессы жизненного цикла ПО, если разработаны планы и стандарты для этих процессов.

6.3 Типы планов ПО

Цель создания планов ПО состоит в том, чтобы определить средства для удовлетворения требованиям настоящего стандарта, в том числе определить организационные подразделения, которые будут выполнять эти работы. В процессе планирования должны быть разработаны следующие типы планов ПО:

- План сертификации в части ПО (12.1) служит основным средством для согласования предложенных методов разработки с сертифицирующей организацией и определяет средства для выполнения требований данного документа;

- План разработки ПО (12.2) определяет используемые модели жизненного цикла ПО и среду разработки ПО;

- План верификации ПО (12.3) определяет средства, с помощью которых будут удовлетворены цели процесса верификации ПО;

- План квалификационного тестирования ПО (12.4) определяет порядок выполнения квалификационного тестирования ПО;

- План управления конфигурацией ПО (12.5) определяет средства, с помощью которых будут удовлетворены цели процесса управления конфигурацией ПО;

- План обеспечения качества ПО (12.6) определяет средства, с помощью которых будут удовлетворены цели процесса обеспечения качества ПО;

- План установки ПО (12.7) определяет действия по установке разработанного ПО на рабочих местах пользователей, включая подготовку и обучение персонала и адаптацию существующих систем;

- План передачи ПО (12.8) определяет аппаратное обеспечение и ПО, а также другие ресурсы, необходимые для поддержки жизненного цикла передаваемого ПО, и описывает планы разработчиков для поставки передаваемых элементов через организации, осуществляющие поддержку.

Планы ПО должны соответствовать требованиям настоящего стандарта, устанавливать процедуры, используемые для реализации требуемых изменений в ПО до его применения на сертифицируемом объекте. Такие изменения могут быть результатом обратной связи от других процессов и могут, в свою очередь, вызывать изменение планов ПО. Планы ПО должны определять критерии переходов между процессами жизненного цикла ПО путем указания:

- входных данных для процесса, включая обратную связь от других процессов;

- действий интегральных процессов, которые могут потребоваться для обработки этих входных данных;

- необходимых инструментальных средств, методов, стандартов и процедур.

6.4 Планирование среды жизненного цикла ПО

Цель планирования среды жизненного цикла ПО состоит в определении методов, инструментальных средств, процедур, языков программирования и аппаратных средств, которые будут использованы для выполнения процессов жизненного цикла ПО и подготовки документов жизненного цикла ПО (раздел 12). Основными элементами среды жизненного цикла ПО являются среда разработки ПО и среда верификации ПО.

6.4.1 Среда разработки ПО

Среда разработки — важный фактор создания ПО высокого качества. Разработчик должен установить, контролировать и сопровождать среду разработки ПО. Разработчик должен гарантировать, что каждый элемент среды корректно выполняет предназначенные функции. Основные принципы выбора методов и инструментальных средств среды разработки ПО следующие:

- в процессе планирования ПО должна быть выбрана такая среда программирования, которая минимизирует потенциальный риск применения конечного программного средства;

- использование аттестованных инструментальных средств или комбинаций инструментальных средств и частей среды разработки ПО должно обеспечивать уверенность в том, что ошибка, внесенная одной частью, будет обнаружена другой. Среда разработки ПО считается приемлемой, если такие части используются согласованно;

- при определении работ процесса верификации ПО или стандартов разработки ПО необходимо учитывать уровень ПО для того, чтобы минимизировать число потенциальных ошибок, связанных со средой программирования;

- если сертификационное доверие к использованию определенной комбинации инструментальных средств достаточно высокое, то применение этих инструментальных средств должно быть определено в соответствующем плане;

- если дополнительные возможности (опции) инструментальных средств разработки ПО выбраны для использования в проекте, то эффект их применения должен быть рассмотрен и определен в соответствующем плане.

6.4.2 Язык программирования и компилятор

В процессе планирования ПО должна быть оценена допустимость использования конкретного языка программирования и компилятора. Необходимо учитывать следующее:

- некоторые компиляторы имеют возможности оптимизировать эффективность объектного кода. Если тестовые варианты дают покрытие, требуемое в соответствии с уровнем ПО, правильность оптимизации не нуждается в проверке, в противном случае воздействие этих возможностей на структурное покрытие должно быть определено в соответствии с требованиями 8.4.4;

- для реализации определенных возможностей компиляторы для некоторых языков могут производить объектный код, который не является непосредственно трассируемым к исходному тексту, например инициализация, встроенное обнаружение ошибок или обработка исключительных ситуаций (8.4.4.2, перечисление б); процесс планирования ПО должен определить в соответствующем плане средства, обнаруживающие этот объектный код, и гарантировать его тестовое покрытие;

- если в течение жизненного цикла ПО вводится новая версия компилятора, редактора связей или загрузчика или изменены опции компилятора, результаты предыдущих тестов и анализ покрытия больше не могут быть рассмотрены как адекватные. Планирование верификации должно предусматривать средства повторной верификации в соответствии с требованиями раздела 8.

6.4.3 Среда верификации ПО

Цель планирования среды верификации ПО состоит в том, чтобы определить методы, инструментальные средства, процедуры и аппаратные средства, которые будут использованы, чтобы проверить выходные результаты процессов разработки. Разработчик должен установить, контролировать и сопровождать среду верификации ПО. Разработчик должен гарантировать, что каждый элемент среды корректно выполняет предназначенные функции. Верификационное тестирование может быть выполнено на объектном компьютере, эмуляторе объектного компьютера или с использованием моделирования на инструментальном компьютере (интерпретатора). Основные требования следующие:

- эмулятор или интерпретатор в некоторых случаях требуется аттестовать как описано в 13.2;

- должны быть рассмотрены различия между результатами, полученными на объектном компьютере и эмуляторе или интерпретаторе, и воздействие этих различий на способности обнаруживать ошибки и проверять функциональные возможности; обнаружение тех ошибок, которые остаются невыявленными, необходимо обеспечивать другими процессами верификации ПО, которые должны быть определены в Плане верификации ПО.

6.5 Стандарты разработки ПО

Целью стандартов разработки ПО является определение правил и ограничений для процессов разработки. К стандартам разработки ПО относятся стандарты на требования к ПО, проектирование и кодирование ПО. Процесс верификации ПО использует эти стандарты для оценки соответствия фактических выходных данных некоторого процесса ожидаемым результатам. Стандарты разработки ПО должны:

- удовлетворять требованиям раздела 11;

- обеспечивать единообразие разработки компонентов ПО данного программного продукта или необходимого набора средств;

- исключать использование конструкций или методов, результаты которых не могут быть верифицированы или несовместимы с требованиями безопасности.

6.6 Ответственность за выполнение планирования

Разработчик должен осуществлять планирование проекта и надзор за его выполнением в соответствии со следующими требованиями. Если систему или ЭКПО разрабатывают для нескольких различных построений, планирование для каждого построения должно предусматривать:

- полное планирование для контракта;
- детализированное планирование для текущего построения;
- планирование будущих построений, предусмотренных контрактом, с уровнем детализации, соответствующим доступной на данный момент информации.

Разработчик должен создать официальный документ, планирующий проведение работ в соответствии с требованиями настоящего стандарта и требованиями контракта, связанными с ПО. Планирование должно быть выполнено в соответствии с уровнем системы и завершено включением всей требуемой информации в документы планирования.

Примечания

1 Эта формулировка здесь и далее в настоящем стандарте предназначена для того, чтобы:

- подчеркнуть, что создание и регистрация информации о планировании и технологии разработки — существенная часть процесса разработки ПО — должны быть выполнены независимо от требований к поставляемому средству;
- использовать документ как контрольный список построений, которые покрываются действиями разработки или планирования;
- допустить для регистрации представления информацию, отличную от традиционных документов (например, автоматизированные инструментальные средства проектирования ПО).

2 Если контракт предусматривает передачу информации в соответствии с настоящим стандартом, в обязанности разработчика входит форматирование, сбор, маркировка, копирование и рассылка поставляемых документов, что требует дополнительного времени и усилий со стороны разработчика.

3 План разработки ПО может покрывать все виды работ, требуемых настоящим стандартом, и включать в себя описание планирования интегральных процессов, если контрактом не предусмотрены отдельные документы планирования для этих процессов.

Разработчик должен участвовать в разработке и регистрации планов проведения квалификационного тестирования системы, результаты должны быть включены в документ «План квалификационного тестирования ПО».

Разработчик должен создать и зарегистрировать план выполнения установки ПО и обучения пользователей, определенный в контракте, результаты должны быть включены в документ «План установки ПО».

Разработчик должен идентифицировать все ресурсы разработки ПО, которые будут необходимы для реализации концепции поддержки организации, осуществляющей поддержку. Разработчик должен создать и зарегистрировать планы, идентифицирующие эти ресурсы, и описать действия, необходимые при передаче поставляемых элементов агентству поддержки. Результаты данного планирования должны быть включены в документ «План передачи ПО».

После утверждения заказчиком любого из планов, указанных в данном подразделе, разработчик должен выполнить релевантные действия в соответствии с планами. Руководство разработчика и Служба обеспечения качества ПО должны осуществлять аудит процесса разработки ПО в интервалах, определенных в Плане разработки ПО, для гарантии того, что процесс будет выполнен в соответствии с контрактом и утвержденными планами. За исключением внутреннего планирования разработчика и информации, связанной с укомплектованием персонала, любая модификация планов должна быть одобрена заказчиком.

6.7 Просмотр результатов процесса планирования ПО

Просмотры результатов процесса планирования ПО проводят для гарантии того, что планы ПО и стандарты разработки ПО соответствуют требованиям настоящего стандарта и обеспечивают согласованное выполнение процессов жизненного цикла ПО.

7 Процессы разработки ПО

Процессы разработки ПО должны быть выполнены в соответствии с процессом планирования ПО (раздел 6) и Планом разработки ПО (12.2). Таблица А.2 содержит резюме целей и результатов процессов разработки ПО в зависимости от уровня ПО. Процессами разработки ПО являются следующие процессы:

- определение требований к ПО;
- проектирование ПО;

- кодирование ПО;
- интеграция.

7.1 Процесс определения требований к ПО

Разработчик должен определить и зарегистрировать требования к ПО, которые будут выполнены каждым ЭКПО, методы, которые нужно использовать, для гарантии того, что каждое требование было выполнено, и проследить соответствие между требованиями к ЭКПО и системными требованиями. Результат этих работ должен быть включен в документ «Спецификация требований к ПО». В зависимости от условий контракта требования относительно интерфейсов ЭКПО могут быть включены либо в Спецификацию требований к ПО, либо в Спецификацию требований к интерфейсу.

Примечание — Если ЭКПО разрабатывают для нескольких различных построений, требования к ЭКПО не могут быть полностью определены до завершения конечного построения. Планирование, выполняемое разработчиком, для каждого ЭКПО должно идентифицировать подмножество требований, которые будут определены в каждом построении, и подмножество, которое будет реализовано в каждом построении. Анализ требований к ПО для данного построения следует интерпретировать как определение требований к ЭКПО, идентифицированных для этого построения.

7.1.1 Цели процесса определения требований к ПО

Цели данного процесса состоят в том, чтобы:

а) разработать требования верхнего уровня;

б) оценить производные требования верхнего уровня с точки зрения безопасности системы.

7.1.2 Состав работ, выполняемых в процессе определения требований к ПО

Входными данными для процесса определения требований к ПО являются системные требования, описания аппаратного интерфейса и архитектуры системы (если они не включены в системные требования), определяемые процессами жизненного цикла системы, План разработки ПО и стандарты на разработку требований к ПО, определяемые процессом планирования. После того как удовлетворены указанные в Плане разработки ПО критерии перехода к данному процессу разработки, входные данные используют для разработки требований верхнего уровня к ПО. Требования верхнего уровня включают в себя функциональные требования, требования к эффективности, требования к интерфейсу и требования, связанные с безопасностью. Результатами данного процесса являются документы «Спецификация требований к ПО» (12.13) и «Спецификация требований к интерфейсу» (12.14). Процесс определения требований к ПО считают завершенным, когда достигнуты его цели и цели связанных с ним интегральных процессов. Процесс определения требований к ПО должен обеспечить следующее:

- анализ функциональных системных требований и требований к интерфейсам, которые предназначены для программной реализации, на отсутствие противоречий, несоответствий и неопределенностей;
- регистрацию для последующего уточнения или исправления и передачи в качестве входной информации обратной связи к исходным процессам тех входных данных процесса определения требований к ПО, которые оказались неадекватными или некорректными;
- спецификацию в документе требований верхнего уровня каждого системного требования, которое предназначено для программной реализации;
- определение всех требований верхнего уровня, соответствующих системным требованиям, которые связаны с предотвращением риска;
- верифицируемость, непротиворечивость и соответствие требований верхнего уровня стандартам на разработку требований к ПО;
- установление требований верхнего уровня в количественных показателях с погрешностями в тех случаях, когда это необходимо;
- требования верхнего уровня не должны описывать детали проектирования или верификации, исключая определения и обоснования ограничений проектирования;
- трассируемость каждого системного требования, которое предназначено для программной реализации, к одному или нескольким требованиям верхнего уровня для ПО;
- трассируемость каждого требования верхнего уровня, кроме производных требований к одному или нескольким системным требованиям;
- оценку производных требований верхнего уровня с точки зрения безопасности системы.

7.2 Процесс проектирования ПО

Требования верхнего уровня к ПО уточняют далее в процессе проектирования ПО одной или несколькими итерациями, чтобы разработать архитектуру ПО и требования нижнего уровня, которые могут быть непосредственно использованы для получения исходного кода.

7.2.1 Цели процесса проектирования ПО

Цели данного процесса состоят в том, чтобы:

а) разработать архитектуру ПО и требования нижнего уровня на основе требований верхнего уровня;

б) оценить с точки зрения безопасности системы производные требования нижнего уровня.

7.2.2 Состав работ, выполняемых в процессе проектирования ПО

Входными данными процесса проектирования ПО являются требования к ПО, План разработки ПО и стандарты на процесс проектирования ПО. После того как удовлетворены указанные в Плане разработки ПО критерии перехода к данному процессу разработки, эти входные данные используются в процессе проектирования для разработки архитектуры ПО и требований нижнего уровня. Требования нижнего уровня могут включать в себя одно или несколько требований более низких уровней. Основным выходным результатом процесса является документ «Описание проекта ПО» (12.16), который содержит описание архитектуры ПО и требования нижнего уровня. Если это предусмотрено условиями контракта, часть проекта, имеющая отношение к интерфейсам, может быть включена в документ «Описание проекта интерфейса» (12.17), а часть проекта, имеющая отношение к базам данных, может быть включена в документ «Описание проекта базы данных» (12.18). Процесс проектирования ПО считают завершенным, когда удовлетворены его цели и цели связанных с ним интегральных процессов. Процесс проектирования ПО должен обеспечивать следующее:

- архитектура ПО и требования нижнего уровня, разработанные в процессе проектирования ПО, должны соответствовать стандартам на процесс проектирования ПО и быть прослеживаемыми, верифицируемыми и непротиворечивыми;

- производные требования должны быть определены и проанализированы для гарантии того, что они не противоречат требованиям верхнего уровня;

- работы процесса проектирования ПО могут привести к появлению возможных отказов в ПО или, наоборот, предотвратить их. Использование метода разбиения или других архитектурных методов при проектировании ПО может изменить установленный уровень критичности ПО для некоторых компонентов ПО. В таких случаях должна быть определена информация о производных требованиях, позволяющая обеспечить процесс оценки безопасности системы;

- необходимо контролировать поток управления и поток данных, когда это связано с требованиями безопасности;

- реакция на отказные ситуации должна быть согласована с требованиями безопасности;

- при обнаружении в процессе проектирования ПО неадекватных или некорректных входных данных следует осуществлять обратную связь с процессами жизненного цикла системы, с процессами определения требований к ПО или с процессом планирования ПО для исследования проблемы или исправления входных данных.

Примечание — На современном уровне развития технологии разработки ПО не применяют количественную корреляцию между сложностью и достигаемой безопасностью. Тем не менее, в процессе проектирования ПО необходимо избегать сложных конструкций, поскольку с увеличением сложности ПО становятся более трудными верификация и доказательство того, что удовлетворены цели безопасности.

7.2.3 Проектирование модифицируемого пользователем ПО

Для разработки ПО, которое предназначено для модификации пользователями, необходимы дополнительные требования. Модифицируемый компонент — часть ПО, которая предназначена для изменения пользователем, а немодифицируемый компонент не допускает изменения пользователем. Модифицируемое пользователем ПО может широко варьироваться по сложности. ПО любого уровня может включать в себя модифицируемые компоненты. Требования для проектирования модифицируемого пользователем ПО следующие:

- немодифицируемый компонент должен быть защищен от модифицируемого компонента, чтобы предотвратить влияние на безопасность функционирования немодифицируемого компонента. Эта защита может быть обеспечена аппаратными, программными, инструментальными средствами, используемыми для реализации изменений, или их комбинацией;

- нужно показать, что специально предназначенные для реализации изменений средства являются единственно допустимыми для модификации.

7.2.4 Проектные решения уровня ЭКПО

Разработчик должен определить и зарегистрировать проектные решения уровня ЭКПО. Результаты должны быть включены в раздел проектных решений уровня ЭКПО документов проектирования ПО (12.16, 12.17, 12.18).

Разработчик должен определить и зарегистрировать проект архитектуры ПО (идентифицировать модули ПО, входящие в ЭКПО, их интерфейсы и концепцию их совместного выполнения) и проследить соответствие между модулями ПО и требованиями к ЭКПО. Результат этих работ должен быть включен в разделы документа «Описание проекта ПО», посвященные архитектуре ПО и прослеживанию соответствия. В зависимости от условий контракта часть проекта, имеющая отношение к интерфейсам, может быть включена либо в Описание проекта ПО, либо в Описание проекта интерфейса.

Разработчик должен разработать и зарегистрировать описание каждого модуля ПО. Результат этих работ должен быть включен в соответствующий раздел документа «Описание проекта ПО». В зависимости от условий контракта часть проекта, которая относится к интерфейсам, может быть включена либо в Описание проекта ПО, либо в Описание проекта интерфейса. В свою очередь проект программных модулей, которые являются базами данных или осуществляют доступ к базам данных или управление ими, может быть включен либо в Описание проекта ПО, либо в Описание проекта базы данных.

Примечание — Если ЭКПО разрабатывают для нескольких различных построений, проект не может быть полностью определен до завершения последнего построения. Проект ПО для данного построения должен обязательно удовлетворять требованиям к ЭКПО, которые должны быть реализованы для этого построения.

7.3 Процесс кодирования ПО

В процессе кодирования ПО на основании архитектуры ПО и требований нижнего уровня создают исходный код.

Разработчик должен разработать и зарегистрировать исходный код ПО, соответствующий каждому модулю ПО в проекте ЭКПО. Реализация ПО должна включать в себя, если это применимо, кодирование машинных команд и определение данных, создание базы данных, заполнение базы данных и других файлов данных значениями данных, а также другие работы, необходимые для реализации проекта. Если для кодирования поставляемого ПО предполагается использовать язык программирования, отличный от указанного в контракте, разработчик должен получить одобрение заказчика на использование этого языка.

7.3.1 Цели процесса кодирования ПО

Цели процесса кодирования ПО состоят в том, чтобы разработать исходный код, который должен быть прослеживаемым, верифицируемым, непротиворечивым и корректно реализующим требования нижнего уровня.

7.3.2 Состав работ, выполняемых в процессе кодирования ПО

Входными данными процесса кодирования ПО являются требования нижнего уровня, архитектура ПО, План разработки ПО и стандарты кодирования ПО. Когда указанные в плане критерии перехода удовлетворены, может быть осуществлен первичный или повторный переход к процессу кодирования ПО. Исходный код, полученный при выполнении этого процесса, базируется на архитектуре ПО и требованиях нижнего уровня. Результат этого процесса — исходный код (12.19) и объектный код. Процесс кодирования ПО является завершенным, когда реализованы все его цели и цели интегральных процессов, связанных с ним. Требования для этого процесса следующие:

- исходный код должен реализовывать требования нижнего уровня и соответствовать архитектуре ПО;
- исходный код должен соответствовать стандартам кодирования ПО;
- исходный код должен быть трассируемым к описанию проекта;
- для неадекватных или некорректных входных данных, обнаруженных при выполнении процесса кодирования ПО, необходимо обеспечить обратную связь с процессами определения требований к ПО, проектирования ПО или планирования ПО для исследования или исправления.

7.4 Процесс интеграции

Объектный компьютер, исходный код и объектный код, полученные в процессе кодирования ПО, используют при редактировании связей и загрузке с целью создать интегрированную систему.

7.4.1 Цели процесса интеграции

Цели процесса интеграции состоят в получении интегрированной системы.

7.4.2 Состав работ, выполняемых в процессе интеграции

После того как указанные в Плане разработки ПО критерии перехода будут удовлетворены, может быть осуществлен первичный или повторный переход к процессу интеграции. Входными данными процесса интеграции являются описание архитектуры ПО из процесса проектирования ПО, а также исходный и объектный код из процесса кодирования ПО.

Выходной результат процесса интеграции — исполняемый объектный код, описанный в 12.20, и информация о редактировании связей и загрузке. Процесс интеграции является завершенным, когда

удовлетворены его цели и цели интегральных процессов, связанных с ним. Требования для этого процесса:

- исполняемый объектный код должен быть генерирован на основе исходного кода и информации о редактировании связей и загрузке;
- ПО должно быть загружено в объектный компьютер для интеграции аппаратных средств и ПО;
- для неадекватных или некорректных входных данных, обнаруженных в процессе интеграции, необходимо обеспечить обратную связь с процессами определения требований к ПО, проектирования ПО, кодирования ПО или планирования ПО для исследования или исправления.

7.4.3 Дополнительные задачи интеграции

Далее рассмотрены задачи, связанные с отключенным кодом и заплатами в ПО. Управляющая система или оборудование могут быть предназначены для включения нескольких вариантов конфигураций, не все из которых предназначены для использования в каждом приложении. Это может привести к появлению отключенного кода, который может быть не выполнен, или данных, которые могут быть не использованы. Такой код отличается от мертвого кода, который определен в разделе 3 и объяснен в 8.4.4. Требования для отключенного кода и заплат следующие:

- должно быть приведено доказательство того, что отключенный код заблокирован для среды, где его использование не предусмотрено. Непреднамеренную активацию отключенного кода, возникающую в аварийных ситуациях системы, следует рассматривать также как непреднамеренную активацию обычного активизированного кода;
- должны быть использованы методы работы с отключенным кодом в соответствии с планами ПО;
- нельзя использовать заплаты в ПО, предназначенном для поставки в сертифицируемый объект, для выполнения изменения в требованиях или архитектуре, или изменений, оказавшихся необходимыми в результате работ процесса верификации ПО. Заплаты следует использовать в ограниченных случаях, например, чтобы устранить замеченные неточности среды программирования, типа обнаружения ошибки компилятора и др.;
- при использовании заплаты необходимо подтвердить, что процесс управления конфигурацией ПО может эффективно прослеживать эту заплату; провести регрессионный анализ для доказательства того, что включение заплаты удовлетворяет всем требованиям к ПО, разработанному с помощью обычных методов; обосновать использование заплаты в Итоговом документе разработки ПО.

7.5 Трассируемость

Требования трассируемости включают в себя обеспечение соответствия:

- между системными требованиями и требованиями к ПО, чтобы гарантировать полноту реализации системных требований и видимость производных требований;
- между требованиями нижнего уровня и требованиями верхнего уровня, чтобы гарантировать полноту реализации требований верхнего уровня и видимость производных требований и архитектурных решений, принятых при выполнении процесса проектирования ПО;
- между исходным кодом и требованиями нижнего уровня, чтобы верифицировать отсутствие неописанного исходного кода и полноту реализации требований нижнего уровня.

8 Процесс верификации ПО

Верификация ПО обеспечивает техническую оценку всех средств разработки ПО, в том числе и результатов верификации ПО. Верификацию ПО выполняют в соответствии с Планом верификации ПО (12.3) и Планом квалификационного тестирования ПО (12.4), которые разрабатывают в процессе планирования ПО.

Таблицы А.3 — А.7 содержат резюме целей и результатов верификации в зависимости от уровня ПО.

Примечание — Чем ниже уровень ПО, тем меньше внимания уделяют:

- верификации требований нижнего уровня;
- верификации архитектуры ПО;
- полноте покрытия тестами;
- контролю процедур верификации;
- независимости работ процесса верификации ПО;
- перекрытию работ процесса верификации ПО, т.е. выполнению различных верификационных работ, каждая из которых обнаруживает ошибки одного и того же класса;
- тестированию отказоустойчивости;
- верификационным работам, обнаруживающим ошибки, которые оказывают лишь косвенное влияние на результаты разработки, например отклонение от стандартов разработки ПО.

8.1 Цели верификации ПО

Назначение верификации ПО состоит в том, чтобы обнаружить и зарегистрировать ошибки, которые могли быть внесены в ПО во время его разработки (устранение ошибок является задачей разработки ПО). Основное назначение верификации ПО — проверить что:

- системные требования, предназначенные для программной реализации, были должным образом переработаны в требования верхнего уровня к ПО, которые удовлетворяют этим системным требованиям;
- требования верхнего уровня были переработаны в архитектуру ПО и требования нижнего уровня, которые удовлетворяют требованиям верхнего уровня; если разработано несколько уровней требований к ПО между требованиями верхнего уровня и требованиями нижнего уровня, то каждый последующий уровень требований разработан так, чтобы удовлетворять требованиям более высокого уровня;
- архитектура ПО и требования нижнего уровня должным образом преобразованы в исходный код, удовлетворяющий им;
- исполняемый объектный код удовлетворяет требованиям к ПО;
- инструментальные средства, используемые для выполнения указанных работ, являются технически корректными и полными для заданного уровня ПО.

8.2 Состав работ, выполняемых в процессе верификации ПО

Цели верификации ПО должны быть достигнуты посредством выполнения комбинации просмотров, анализов, разработки тестовых наборов и процедур и последующего выполнения этих тестовых процедур. Просмотры и анализы обеспечивают оценку точности, полноты и верифицируемости требований к ПО, архитектуры ПО и исходного кода. Разработка тестовых наборов должна обеспечивать дальнейшую оценку внутренней непротиворечивости и полноты требований. Выполнение тестовых процедур обеспечивает демонстрацию соответствия требованиям.

Входная информация для процесса верификации ПО включает в себя системные требования, требования к ПО, описание архитектуры, данные о трассируемости, исходный код, исполняемый объектный код, План верификации ПО, План квалификационного тестирования ПО.

Выходные результаты верификации ПО должны быть включены в документы «Процедуры верификации ПО» (12.21), «Результаты верификации ПО» (12.23), «Описание квалификационного тестирования ПО» (12.22), «Отчет о квалификационном тестировании ПО» (12.24).

8.3 Просмотры и анализы ПО

Просмотры и анализы ПО применяют к результатам процессов разработки и верификации ПО. Различия между просмотрами и анализами заключаются в том, что анализ дает воспроизводимое доказательство, а просмотр предоставляет качественную (экспертную) оценку. Просмотр может включать в себя инспекцию выходных результатов указанных процессов, использующую контрольные листы или другие подобные средства. Анализ может заключаться в детальном исследовании функциональности, эффективности и безопасности компонентов ПО, а также их связи с другими компонентами системы или с оборудованием.

8.3.1 Просмотры и анализы требований верхнего уровня

Цель этих просмотров и анализов — обнаружить и зарегистрировать ошибки, которые могли быть внесены в процессе разработки требований к ПО. Данные просмотры и анализы должны подтвердить, что требования верхнего уровня удовлетворяют следующим целям:

а) согласованность с системными требованиями: гарантировать, что функции системы, которые должно выполнять ПО, определены, что требования по функциональности, эффективности и требования, связанные с безопасностью системы, удовлетворены в требованиях верхнего уровня к ПО и что правильно определены производные требования и обоснована их необходимость;

б) точность и непротиворечивость: гарантировать, что каждое требование верхнего уровня является точным, однозначным и достаточно детализированным и что требования не противоречат друг другу;

в) совместимость с объектным компьютером: гарантировать, что не существует никаких противоречий между требованиями верхнего уровня и возможностями аппаратных/программных средств объектного вычислителя, особенно такими, как время реакции системы и аппаратура ввода/вывода;

г) верифицируемость: гарантировать, что каждое требование верхнего уровня может быть верифицировано;

д) соответствие стандартам: гарантировать, что процесс разработки требований к ПО полностью соответствует стандартам на разработку требований и обоснованы любые отклонения от данных стандартов;

е) трассируемость: гарантировать, что функциональные системные требования, требования по эффективности и требования к безопасности системы, предназначенные для программной реализации, были включены в требования верхнего уровня;

ж) алгоритмические аспекты: гарантировать точность и корректность поведения предложенных алгоритмов, особенно в областях отсутствия непрерывности.

8.3.2 Просмотры и анализы архитектуры ПО

Цель этих просмотров и анализов — обнаружить и зарегистрировать ошибки, которые могли быть внесены во время разработки архитектуры ПО. Данные просмотры и анализы должны подтвердить, что архитектура ПО соответствует следующим требованиям:

а) согласованность с требованиями верхнего уровня: гарантировать, что архитектура ПО не находится в противоречии с требованиями верхнего уровня, особенно те функции, которые гарантируют целостность системы, например схемы разбиения;

б) непротиворечивость: гарантировать, что существует корректная связь между компонентами архитектуры ПО, осуществляемая через потоки данных и поток управления;

в) совместимость с объектным компьютером: гарантировать, что не существует никаких противоречий между архитектурой ПО и программно-аппаратными возможностями объектного компьютера, особенно такими, как инициализация, асинхронные операции, синхронизация и прерывания;

г) верифицируемость: гарантировать, что архитектура ПО может быть верифицирована, например не существует неограниченных рекурсивных алгоритмов;

д) соответствие стандартам: гарантировать, что процесс проектирования ПО полностью соответствовал стандартам на процесс проектирования ПО и отклонения от этих стандартов обоснованы, особенно ограничения сложности и использования конструкций проекта, которые не согласуются с задачами безопасности системы;

е) целостность разбиения: гарантировать, что будут предотвращены или изолированы любые нарушения в декомпозиции.

8.3.3 Просмотры и анализы требований нижнего уровня

Цель этих просмотров и анализов — обнаружить и зарегистрировать ошибки, которые могли быть внесены в процессе проектирования ПО. Эти просмотры и анализы должны подтвердить, что требования нижнего уровня удовлетворяют следующим целям:

а) согласованность с требованиями верхнего уровня: гарантировать, что требования нижнего уровня к ПО удовлетворяют требованиям верхнего уровня к ПО, что формируемые требования правильно определены и обоснована их необходимость;

б) точность и непротиворечивость: гарантировать, что каждое требование нижнего уровня является точным, однозначным и достаточно детализированным и что требования нижнего уровня не противоречат друг другу;

в) совместимость с объектным компьютером: гарантировать, что не существует никаких противоречий между требованиями к ПО и возможностями аппаратных и программных средств объектного компьютера, особенно по использованию ресурсов (таких, как загрузка шины, время реакции системы и аппаратуры ввода/вывода);

г) верифицируемость: гарантировать, что каждое требование нижнего уровня может быть верифицировано;

д) соответствие стандартам: гарантировать, что процесс проектирования ПО полностью соответствует стандартам на процесс проектирования ПО и что отклонения от этих стандартов обоснованы;

е) трассируемость: гарантировать, что требования верхнего уровня и производные требования были реализованы в требованиях нижнего уровня;

ж) алгоритмические аспекты: гарантировать точность и корректность поведения предложенных алгоритмов, особенно в областях отсутствия непрерывности.

8.3.4 Просмотры и анализы исходного кода

Цель этих просмотров и анализов — выявление и регистрация ошибок, которые могли быть внесены в процессе кодирования ПО. Эти просмотры и анализы подтверждают, что выходные результаты кодирования являются точными, полными и могут быть верифицированы. Прежде всего проверяют корректность кода по отношению к требованиям к ПО и архитектуре ПО и соответствие стандартам на кодирование. Эти просмотры и анализы обычно ограничиваются исходным кодом. На данном этапе необходимо показать:

а) согласованность с требованиями нижнего уровня: гарантировать, что исходный код является

корректным, полным и соответствует требованиям нижнего уровня, а также то, что исходный код не содержит неописанных функций;

б) согласованность с архитектурой ПО: гарантировать, что исходный код соответствует потоку данных и потоку управления, которые определены архитектурой ПО;

в) верифицируемость: гарантировать, что исходный код не содержит операторов и структур, которые не могут быть верифицированы, и что код не подвергался изменениям в целях тестирования;

г) соответствие стандартам: гарантировать, что процесс разработки кода ПО полностью соответствует стандартам кодирования ПО и отклонения от этих стандартов обоснованы, особенно в случаях ограничения сложности и использования конструкций кода, предназначенных для удовлетворения целей безопасности системы (сложность в данном контексте — степень связности программных компонентов, уровень вложенности управляющих структур и сложность логических или числовых выражений); этот анализ также должен гарантировать, что отклонения от стандартов оправданы;

д) трассируемость: гарантировать, что требования нижнего уровня к ПО были воплощены в исходный код;

е) точность и непротиворечивость: гарантировать правильность и непротиворечивость исходного кода, включая реализацию стеков, переполнение и разрешающую способность для арифметики с фиксированной точкой, конкуренцию ресурсов, синхронизацию выполнения самых сложных случаев, обработку особых ситуаций, использование неинициализированных переменных или констант, неиспользуемые переменные или константы и нарушения целостности данных из-за конфликтов прерываний или задач.

8.3.5 Просмотры и анализы выходных результатов процесса интеграции

Цель этих просмотров и анализов — гарантировать, что результаты процесса интеграции ЭКПО являются полными и корректными. Это может быть выполнено путем детального исследования информации о редактировании связей, загрузке и картах памяти. Должны быть проконтролированы и исключены:

- неправильные аппаратные адреса;
- перекрытия памяти;
- отсутствующие компоненты ПО.

8.3.6 Просмотры и анализы тестовых вариантов, процедур и результатов

Цель этих просмотров и анализов — гарантировать, что тестирование кода было разработано и выполнено точно и полностью. Должны быть рассмотрены следующие вопросы:

а) тестовые варианты: верификация тестовых вариантов представлена в 8.4.4;

б) тестовые процедуры: проверить, что тестовые варианты правильно представлены в процедурах тестирования и ожидаемых результатах;

в) результаты тестирования: гарантировать, что результаты тестирования корректны и что расхождения между фактическими и ожидаемыми результатами объяснимы.

8.4 Цели и методы тестирования ПО

Тестирование ПО систем управления имеет две взаимодополняющие цели. Первая цель — показать, что ПО удовлетворяет требованиям к нему. Вторая цель — продемонстрировать с высокой степенью доверия, что были устранены ошибки, которые могли бы привести к возникновению отказных ситуаций, определенных процессом оценки безопасности системы. Выделяют три уровня тестирования:

- тестирование интеграции ЭКПО/ЭКА, верифицирующее корректность функционирования ПО в среде объектного вычислителя;
- тестирование интеграции ЭКПО, верифицирующее взаимосвязи между требованиями и компонентами ПО и реализацию требований и компонентов в рамках архитектуры;
- тестирование нижнего уровня (модульное тестирование), верифицирующее реализацию требований нижнего уровня.

Примечание — Если разработан тестовый вариант и выполнена соответствующая процедура для тестирования интеграции ЭКПО/ЭКА или тестирования интеграции ЭКПО и удовлетворены критерии покрытия, базирующиеся на требованиях и структуре, то нет необходимости дублировать этот тестовый вариант для тестирования нижнего уровня. Замена тестов верхнего уровня номинально эквивалентными тестами нижнего уровня может быть менее эффективной из-за меньшего объема тестированных функциональных требований.

Для удовлетворения целей тестирования ПО:

- тестовые варианты должны быть основаны, прежде всего, на требованиях к ПО;

- тестовые варианты должны быть разработаны так, чтобы верифицировать корректность функционирования и сформировать условия, которые выявляют потенциальные ошибки;
- анализ покрытия требований к ПО должен определить, какие требования к ПО не были тестированы;
- анализ структурного покрытия должен определить, какие структуры ПО не были выполнены при тестировании.

8.4.1 Среда тестирования

Для достижения целей тестирования ПО может потребоваться более одной среды тестирования. Идеальная среда тестирования включает в себя ПО, загруженное в объектный вычислитель и тестируемое в среде, которая имитирует среду объектного вычислителя с высокой точностью.

Возможно сертификационное доверие для эмулятора или имитатора объектного вычислителя на инструментальном компьютере. Однако рекомендации относительно среды тестирования сводятся к следующему: некоторые тесты должны быть выполнены только в интегрированной объектной вычислительной среде, так как некоторые ошибки могут быть обнаружены только в этой среде.

8.4.2 Выбор тестовых вариантов, основанных на требованиях

Тестированию, основанному на требованиях, уделяют особое внимание, потому что эту стратегию признают наиболее эффективной в обнаружении ошибок. Рекомендации для выбора тестовых вариантов, основанных на требованиях, заключаются в следующем:

- для того чтобы выполнить задачи тестирования ПО, необходимы две категории тестовых вариантов: тесты для проверки функционирования в области допустимых значений и тесты для проверки на устойчивость к ошибкам входных данных (вне данной области);
- обеспечить особые тестовые варианты, разработанные на основе требований к ПО с учетом потенциальных источников ошибок, присущих процессам разработки ПО.

Назначение тестовых вариантов для области допустимых значений — продемонстрировать способность ПО корректно функционировать в штатных условиях и для входных данных из области допустимых значений. Тестовые варианты данной категории включают в себя следующее:

- вещественные и целые входные переменные, которые выбирают с использованием допустимых классов эквивалентности и граничных значений;
- выполнение многократных итераций кода для функций, зависящих от времени, таких как фильтры и задержки, чтобы проверить характеристики этих функций в правильном контексте;
- для проверки перехода состояний разрабатывают тестовые варианты, реализующие переходы, возможные при нормальной работе;
- тестовые наборы, которые должны проверить использование переменных и выполнение булевых операторов для требований к ПО, выраженных логическими уравнениями.

Цель тестовых вариантов проверки устойчивости к ошибкам — показать способность ПО отрабатывать недопустимые входные данные и условия. Требования к тестовым вариантам устойчивости к ошибкам следующие. Должны быть:

- выбраны вещественные и целые переменные из недопустимых классов эквивалентности;
- проверена инициализация системы для недопустимых условий;
- определены режимы с возможными ошибками для поступающих данных, особенно для сложных цифровых последовательностей данных из внешней системы;
- разработаны тестовые наборы для циклов, когда счетчик цикла — вычисляемое значение, чтобы попытаться получить значения счетчика цикла, выходящие из диапазона допустимых значений, и таким образом показать устойчивость кода, связанного с циклом;
- разработаны тестовые наборы для проверки механизмов защиты от арифметического переполнения для функций, зависящих от времени, типа фильтров и задержек;
- разработаны тестовые наборы, чтобы проверить переходы в состояния, которые невозможны в соответствии с требованиями к ПО.

8.4.3 Методы тестирования, основанные на требованиях

Тестирование, основанное на требованиях, является основным методом для тестирования любого уровня: тестирования интеграции ЭКПО/ЭКА, тестирования интеграции ЭКПО и модульного тестирования. За исключением тестирования интеграции ЭКПО/ЭКА, эти методы не требуют специальной среды тестирования или специальной стратегии. Рекомендации для выполнения тестирования, основанного на требованиях, следующие:

а) тестирование, основанное на требованиях, интеграции ЭКПО/ЭКА: данный метод тестирования должен быть сконцентрирован на источниках ошибок, связанных с выполнением ПО в среде объектного вычислителя, и на функционировании на верхнем уровне. Цель тестирования

интеграции ЭКПО/ЭКА — гарантировать, что ПО в объектной среде функционирует в соответствии с требованиями верхнего уровня. Типичные ошибки, выявляемые этим методом тестирования, следующие:

1) неправильная обработка прерываний;
2) отказы, связанные с требованиями по времени выполнения;
3) некорректная реакция ПО на переходные процессы в аппаратуре или аппаратные отказы, например упорядочение начальных действий, сбой при загрузке входных данных и нестабильность питания;
4) проблемы конкуренции для шин данных и других ресурсов;
5) неспособность встроенных тестов обнаруживать некоторые виды отказов;
6) ошибки в интерфейсах аппаратных/программных средств;
7) некорректное поведение циклов обратной связи;
8) некорректная работа аппаратуры, управляющей памятью, или другой аппаратуры, работающей под управлением ПО;
9) переполнение стека;
10) неправильное функционирование механизмов, поддерживающих корректность и совместимость ПО, загружаемого в условиях эксплуатации;
11) нарушения разбиения ПО;

б) тестирование, основанное на требованиях, интеграции ЭКПО: данный метод тестирования должен быть сконцентрирован на взаимосвязях между требованиями к ПО и на реализации требований архитектурой ПО. Цель такого тестирования — гарантировать, что программные компоненты взаимодействуют друг с другом корректно и удовлетворяют требованиям к ПО и архитектуре ПО. Этот метод может быть реализован путем расширения области действия требований посредством последовательной интеграции компонентов кода с соответствующим расширением области действия тестовых вариантов. Типичные ошибки, обнаруживаемые этим методом:

1) некорректная инициализация переменных и констант;
2) ошибки передачи параметров;
3) затирание данных, особенно глобальных;
4) некорректная последовательность событий и действий;

в) модульное тестирование, основанное на требованиях: этот метод тестирования следует применять для демонстрации того, что каждый программный компонент выполняет требования нижнего уровня. Цель тестирования нижнего уровня, основанного на требованиях, — гарантировать, что программные компоненты удовлетворяют этим требованиям нижнего уровня. Типичные ошибки, выявляемые данным методом тестирования:

1) ошибка алгоритма в реализации требования к ПО;
2) некорректная работа цикла;
3) некорректные логические решения;
4) отказ при обработке правильно сформированных комбинаций входных условий;
5) некорректная реакция на отсутствующие или искаженные входные данные;
6) некорректная обработка исключительных ситуаций типа арифметических ошибок или выхода за границы массива;
7) некорректная последовательность вычислений;
8) неадекватные точность вычисления в алгоритме, точность представления данных или выполнение расчетов.

8.4.4 Анализ тестового покрытия

Анализ тестового покрытия — процесс, состоящий из двух шагов, включающий в себя анализ покрытия, основанного на требованиях, и анализ структурного покрытия. Первый шаг — анализ тестовых наборов относительно требований ПО, чтобы подтвердить, что выбранные наборы тестов удовлетворяют установленным критериям. Второй шаг — подтверждение того, что процедуры тестирования, основанные на требованиях, покрыли структуру кода. Анализ структурного покрытия может в некоторых случаях не удовлетворять заданному критерию покрытия. Например, предусмотрены дополнительные рекомендации для разрешения таких ситуаций, как мертвый код.

8.4.4.1 Анализ тестового покрытия, основанного на требованиях

Цель анализа — определить, насколько полно проверены требования к ПО во время выполнения тестирования. Анализ может выявить потребность в дополнительных тестовых наборах, основанных на требованиях. Данный анализ тестового покрытия должен показать, что:

- существуют тестовые варианты для каждого требования к ПО;

- тестовые варианты удовлетворяют критериям тестирования области определения и тестирования на устойчивость к ошибкам, как установлено в 8.4.2.

8.4.4.2 Анализ структурного покрытия.

Цель анализа — определить, существуют ли структуры кода, которые не были проверены тестовыми процедурами, основанными на требованиях. Тестовые варианты, основанные на требованиях, могут не полностью покрыть структуру кода, поэтому выполняют анализ структурного покрытия и проводят дополнительное тестирование, чтобы обеспечить полное структурное покрытие. Рекомендации для анализа структурного покрытия:

а) анализ должен подтвердить полноту структурного покрытия, соответствующую уровню ПО;

б) анализ структурного покрытия может быть выполнен для исходного кода только в том случае, когда уровень ПО не является уровнем А и компилятор генерирует объектный код, который является непосредственно трассируемым к операторам исходного кода. В противном случае должна быть выполнена дополнительная проверка объектного кода, чтобы установить корректность генерированных последовательностей кода. Объектный код, генерированный компилятором для контроля границ массива, — пример такого объектного кода, который не является непосредственно трассируемым к операторам исходного кода;

в) анализ должен подтвердить связность по данным и связность по управлению между компонентами кода.

Анализ структурного покрытия может обнаружить структуры кода, которые не были выполнены во время тестирования. В этом случае требуются дополнительные работы процесса верификации ПО. Наличие таких невыполненных структур кода может быть результатом:

- недостаточности тестовых вариантов или процедур, основанных на требованиях: следовательно, должны быть генерированы дополнительные тестовые варианты или изменены процедуры тестирования, чтобы обеспечить недостающее покрытие. Может потребоваться пересмотреть метод, используемый для выполнения анализа покрытия, основанного на требованиях;

- несоответствия в требованиях к ПО: требования к ПО должны быть модифицированы, разработаны дополнительные тестовые варианты и выполнены соответствующие процедуры тестирования;

- мертвого кода: код должен быть удален и должен быть проведен анализ, чтобы оценить эффект изменения и потребность в повторной верификации;

- отключенного кода: для отключенного кода, не предназначенного для того, чтобы быть выполненным в некоторой конфигурации при реальной эксплуатации, комбинация анализа и тестирования должна показать, что предотвращены, изолированы или устранены ситуации, при которых такой код мог бы быть случайно выполнен. Для отключенного кода, который может быть выполнен только в специальных конфигурациях среды объектного компьютера, должна быть установлена рабочая конфигурация, необходимая для нормального выполнения этого кода, и должны быть разработаны дополнительные тестовые варианты и процедуры тестирования для достижения требуемого критерия покрытия.

8.5 Порядок выполнения тестирования ПО

8.5.1 Модульное тестирование ПО

Подготовка к тестированию модулей. Разработчик должен определить тестовые варианты (в терминах входных данных, ожидаемых результатов и критериев оценки) и тестовые процедуры для тестирования каждого модуля ПО. Тестовые варианты должны покрывать все аспекты проекта для данного модуля. Разработчик должен зарегистрировать эту информацию в соответствующих файлах разработки ПО.

Выполнение модульного тестирования. Разработчик должен выполнить тестирование программного кода, соответствующего каждому модулю. Тестирование должно быть выполнено в соответствии с заранее определенными тестовыми вариантами и тестовыми процедурами.

Изменение и повторное тестирование. Разработчик должен выполнить изменения ПО, связанные с коррекцией дефектов, выявленных в процессе верификации, выполнить повторное тестирование в необходимом объеме и модифицировать файлы разработки ПО и другие программные средства, основываясь на результатах модульного тестирования.

Анализ и регистрация результатов модульного тестирования. Разработчик должен проанализировать результаты модульного тестирования и зарегистрировать результаты тестирования и анализа в соответствующих файлах разработки ПО.

8.5.2 Интеграционное тестирование

Интеграция и тестирование модулей означают объединение программного кода, соответствующего двум или более программным модулям, тестирование полученного в результате кода, чтобы гарантировать, что вместе они работают так, как предполагалось, и продолжение этого процесса до полной интеграции и тестирования кода каждого ЭКПО. Последняя стадия этого тестирования — внутреннее тестирование ЭКПО разработчиком.

Если ЭКПО разрабатывают для нескольких различных построений, то до завершения разработки всех построений не могут быть выполнены полностью интеграция и тестирование модулей данного ЭКПО. Интеграцию и тестирование модулей для каждого построения следует интерпретировать как интеграцию ПО, разработанного для текущего построения, с другим ПО, разработанным для данного и предыдущих построений, и тестирование результата интеграции.

Подготовка к интеграционному тестированию. Разработчик должен определить тестовые варианты (в терминах входных данных, ожидаемых результатов и критериев оценки) и тестовые процедуры для выполнения интеграционного тестирования. Тестовые варианты должны покрывать все аспекты проекта уровня ЭКПО и эскизного проекта ЭКПО. Разработчик должен зарегистрировать эту информацию в соответствующих файлах разработки ПО.

Выполнение интеграционного тестирования. Тестирование следует выполнять в соответствии с тестовыми вариантами и тестовыми процедурами интеграционного тестирования.

Изменение и повторное тестирование. Разработчик должен выполнить изменения ПО, связанные с коррекцией дефектов, выявленных в процессе верификации, выполнить повторное тестирование в необходимом объеме и модифицировать файлы разработки ПО и другие программные средства, основываясь на результатах интеграционного тестирования.

Анализ и регистрация результатов интеграционного тестирования. Разработчик должен проанализировать результаты интеграционного тестирования и зарегистрировать результаты тестирования и анализа в соответствующих файлах разработки ПО.

8.5.3 Интеграция и тестирование ЭКПО/ЭКА

Интеграция и тестирование ЭКПО/ЭКА означают объединение ЭКПО с взаимодействующими ЭКА и ЭКПО, тестирование полученного объединения с целью определить, работают ли они вместе, как предполагалось, и продолжение этого процесса до тех пор, пока интеграция и тестирование не будут выполнены для всех ЭКПО и ЭКА в системе. Последняя стадия этого тестирования — внутреннее тестирование системы разработчиком.

Если систему или ЭКПО разрабатывают для нескольких различных построений, интеграция и тестирование ЭКПО/ЭКА не могут быть выполнены полностью до завершения последнего построения. Интеграцию и тестирование для каждого построения следует интерпретировать как интеграцию текущего построения каждого ЭКПО с текущим построением других ЭКПО и ЭКА и тестирование результатов интеграции с целью показать, что системные требования, которые должны быть реализованы в данном построении, были удовлетворены.

Подготовка к интеграции и тестированию ЭКПО/ЭКА.

Разработчик должен разработать и зарегистрировать тестовые варианты (в терминах входных данных, ожидаемых результатов и критериев оценки) и тестовые процедуры для проведения интеграционного тестирования ЭКПО/ЭКА. Тестовые варианты должны покрывать все аспекты системного уровня проектирования. Разработчик должен зарегистрировать связанную с тестированием информацию в соответствующих файлах разработки ПО.

Выполнение интеграции и тестирования ЭКПО/ЭКА.

Разработчик должен выполнить интеграционное тестирование ЭКПО/ЭКА. Тестирование должно быть выполнено в соответствии с тестовыми вариантами и процедурами интеграционного тестирования ЭКПО/ЭКА.

Изменение и повторное тестирование.

Разработчик должен выполнить все необходимые изменения в ПО, принять участие в повторном тестировании в необходимом объеме и модифицировать файлы разработки ПО и другие программные средства, основываясь на результатах интеграционного тестирования ЭКПО/ЭКА.

Анализ и регистрация результатов интеграции и тестирования ЭКПО/ЭКА.

Разработчик должен участвовать в выполнении анализа результатов интеграции и тестирования ЭКПО/ЭКА. Результаты анализа и тестирования должны быть зарегистрированы в соответствующих файлах разработки ПО.

8.5.4 Квалификационное тестирование системы

Разработчик должен принимать участие в квалификационном тестировании системы.

Квалификационное тестирование системы выполняют для демонстрирования заказчику, что были удовлетворены системные требования. Квалификационное тестирование системы должно покрывать системные требования в Спецификации системы/подсистемы и в соответствующих Спецификациях требований к интерфейсу. Это тестирование противопоставляется внутреннему тестированию системы, выполненному разработчиком, как заключительная стадия интеграции и тестирования ЭКПО/ЭКА.

Если систему разрабатывают для нескольких различных построений, квалификационное тестирование системы в целом не может быть выполнено до завершения последнего построения. Квалификационное тестирование системы для каждого построения следует интерпретировать как планирование и выполнение тестирования для текущего построения системы с целью показать выполнение системных требований, которые должны быть реализованы в данной конфигурации.

Лицом, ответственным за выполнение требований 8.5.4, не должно быть лицо, принимавшее участие в выполнении проектирования или кодировании ПО системы. Это не исключает возможность оказания помощи в проведении квалификационного тестирования со стороны лиц, выполнявших проектирование или кодирование, например путем предоставления тестовых вариантов, основанных на знании внутренней реализации системы.

Квалификационное тестирование системы, выполняемое разработчиком, должно включать в себя тестирование в объектной или альтернативной среде, одобренной заказчиком.

Разработчик должен участвовать в разработке и регистрации процесса подготовки к тестированию, тестовых вариантов и тестовых процедур, которые нужно использовать для квалификационного тестирования системы, и прослеживании соответствия между тестовыми вариантами и требованиями к системе. Для систем ПО все полученные результаты должны быть включены в документ «Описание тестирования ПО». Разработчик должен предварительно уведомить заказчика о времени и месте проведения квалификационного тестирования системы.

Если квалификационное тестирование системы должно быть засвидетельствовано заказчиком, то до его проведения разработчик должен проверить тестовые варианты и тестовые процедуры, чтобы гарантировать, что они полны и точны и что ПО готово для проведения тестирования в присутствии заказчика. Разработчик должен зарегистрировать результаты этой работы в соответствующих файлах разработки ПО и должен модифицировать тестовые варианты и тестовые процедуры соответствующим образом.

Разработчик должен участвовать в квалификационном тестировании системы. Тестирование должно быть выполнено в соответствии с тестовыми вариантами и тестовыми процедурами системного тестирования.

Разработчик должен реализовать все необходимые изменения в ПО и участвовать в проведении повторного тестирования в требуемом объеме, заранее уведомляя заказчика о повторном тестировании. Разработчик обязан вносить необходимые изменения в файлы разработки ПО и другие программные средства в соответствии с результатами квалификационного тестирования системы.

Разработчик должен участвовать в анализе и регистрации результатов квалификационного тестирования системы. Все полученные результаты должны быть включены в соответствующий документ «Отчет о тестировании ПО».

9 Процесс управления конфигурацией ПО

Процесс управления конфигурацией ПО должен быть выполнен так, как определено процессом планирования ПО (раздел 6) и документом «План управления конфигурацией ПО» (12.5). Выходные результаты процесса управления конфигурацией фиксируют в Протоколах управления конфигурацией ПО (12.29) или в других документах жизненного цикла. Таблица А.8 представляет перечень целей и результатов процесса управления конфигурацией.

Разработчик должен осуществлять управление конфигурацией ПО в соответствии с нижеперечисленными требованиями.

Примечание — Если систему или ЭКПО разрабатывают для нескольких построений, то программные средства для каждого из построений могут иметь изменения или дополнения по отношению к программным средствам предыдущих построений. Управление конфигурацией ПО в каждом построении следует понимать как состояние программных средств и контроля в точке начала построения.

9.1 Цели процесса управления конфигурацией ПО

Процесс управления конфигурацией, выполняемый совместно с другими процессами жизненного цикла ПО, направлен на достижение основных целей, а именно на то, чтобы обеспечить:

- определяемую и управляемую конфигурацию ПО на протяжении жизненного цикла;

- целостность при тиражировании исполняемого объектного кода для производства ПО или, в случае необходимости, его повторной генерации для проведения исследований или модификации;
- управление входными и выходными данными процесса в течение жизненного цикла, что гарантирует непротиворечивость и повторяемость работ в процессах;
- контрольную точку для проверки, оценки состояния и контроля изменений посредством управления элементами конфигурации и определения базовой линии;
- контроль над тем, чтобы дефектам и ошибкам было уделено внимание, а изменения были зарегистрированы, утверждены и реализованы;
- оценку соответствия программного средства требованиям;
- надежное физическое архивирование, восстановление и сопровождение элементов конфигурации.

Цели процесса управления конфигурацией не зависят от уровня ПО. Однако выделяют две категории документов жизненного цикла ПО в зависимости от содержания работ управления конфигурацией (9.3).

9.2 Состав работ, выполняемых в процессе управления конфигурацией ПО

Процесс управления конфигурацией включает в себя работы, связанные с идентификацией конфигурации, контролем изменений, определением базовой линии разработки и архивированием программного средства, включая соответствующие документы жизненного цикла, аудитом конфигурации, компоновкой и поставкой программного средства. Процесс управления конфигурацией не прекращается после того, как программное средство принимается заказчиком, а продолжается в течение жизненного цикла системы.

9.2.1 Идентификация конфигурации

Цель работ по идентификации конфигурации заключается в однозначной маркировке каждого элемента конфигурации и последующих версий, чтобы установить базис для управления и ссылок на элементы конфигурации. Должны быть выполнены следующие работы:

- идентификация конфигурации для документов жизненного цикла ПО;
- идентификация конфигурации для каждого элемента конфигурации, для каждого отдельно управляемого компонента элемента конфигурации и для комбинаций элементов конфигурации, которые составляют программное средство;
- идентификация элементов конфигурации до начала реализации контроля изменений и трассируемости документов;
- идентификация элемента конфигурации прежде, чем он будет использован другими процессами жизненного цикла ПО или же будет использован для производства ПО или загрузки ПО;
- если идентификация программного средства не может быть определена физически путем осмотра (например, осмотр номера компонента платы), то исполняемый объектный код должен содержать идентификацию конфигурации, которая должна быть доступна для других компонентов системы. Это также применимо к ПО, загружаемому в полевых условиях (5.3.5).

Разработчик должен участвовать в выборе ЭКПО, выполняемом согласно проекту архитектуры системы, должен идентифицировать объекты, которые будут помещены под управление конфигурацией, и должен назначить уникальный для проекта идентификатор каждому ЭКПО и каждому дополнительному объекту, находящемуся под управлением конфигурацией. Эти объекты включают в себя программные средства, которые должны быть разработаны или использованы согласно контракту, и элементы среды разработки ПО. Схема идентификации должна быть составлена на том уровне, на котором объекты будут фактически контролироваться, например компьютерные файлы, электронные носители данных, документы, модули ПО, элементы конфигурации. Схема идентификации должна включать в себя статус версии/ревизии/выпуска официальной версии для каждого объекта.

9.2.2 Контроль конфигурации

Разработчик должен установить и выполнить процедуры контроля конфигурации в соответствии с уровнем контроля, установленным для каждого идентифицированного объекта (например, авторский контроль, контроль на уровне проекта, контроль заказчика); установить полномочия людей или групп для санкционирования и выполнения изменений на каждом уровне (например, программист/аналитик, руководитель группы программистов, руководитель проекта, заказчик); последовательность работ, которые необходимо выполнить для того, чтобы запросить разрешение на изменение; обработать запрос на изменение, проследить изменение, распределить изменения и сопровождать предыдущие версии. Изменения, которые воздействуют на объект, уже находящийся под контролем заказчика, должны быть предоставлены заказчику в соответствии с установленными контрактом формами и процедурами.

9.2.3 Базовые линии и трассируемость

Цель установления базовой линии — определить основу для последующих работ процессов жизненного цикла ПО, позволить осуществлять ссылки, управлять элементами конфигурации и контролировать трассируемость. В рамках данной работы требуется:

а) установить базовые линии для элементов конфигурации, на которые распространяется сертификационное доверие (промежуточные базовые линии могут быть установлены с целью обеспечить контроль работ процессов жизненного цикла ПО);

б) установить базовую линию для программного средства и определить ее в Указателе конфигурации ПО (12.26).

Примечание — Модифицируемое пользователем ПО не входит в базовую линию программного средства, за исключением связанных с ним компонентов защиты и граничных компонентов. Следовательно, в модифицируемое пользователем программное средство могут быть внесены изменения, которые не будут влиять на идентификацию конфигурации базового программного средства;

в) базовые линии следует хранить в контролируемых библиотеках ПО (физических, электронных или других), чтобы обеспечить их целостность. После того как базовая линия будет установлена, она должна быть защищена от внесения изменений;

г) после проведения работ, относящихся к контролю изменений, должна быть разработана базовая линия, производная от ранее установленной базовой линии;

д) базовая линия должна быть трассируема к той базовой линии, производной от которой она является, если при сертификации новой базовой линии используется сертификационное доверие к работам или документам процессов жизненного цикла, связанных с разработкой предшествующей базовой линии;

е) когда правомерно сертификационное доверие к работам или документам процессов жизненного цикла ПО, связанным с разработкой предшествующей версии элемента конфигурации, каждый элемент конфигурации должен быть трассируем к тому элементу конфигурации, производным от которого он является;

ж) базовая линия и элемент конфигурации должны быть трассируемы либо к выходным данным, которые они идентифицируют, либо к процессу, с которым они связаны.

9.2.4 Отчетность о дефектах, трассируемость и корректирующие действия

Цель отчетности о дефектах, трассируемости и корректирующих действий заключается в том, чтобы зарегистрировать несоответствие процесса требованиям планов и стандартов, отсутствие выходных данных процессов жизненного цикла ПО, аномальное поведение программных средств, а также гарантировать разрешение этих проблем.

Примечание — Дефекты, связанные с процессами жизненного цикла ПО, и дефекты, связанные с программными средствами, могут быть зафиксированы в отдельных системах отчетности о дефектах.

Требования к выполнению данных работ:

- должно быть подготовлено сообщение о дефекте, которое описывает несоответствие процесса планам, отсутствие выходных данных или аномальное поведение ПО, а также о предпринятых корректирующих действиях (как это установлено в 12.28);
- отчетность о дефектах должна предусматривать идентификацию затрагиваемых элементов конфигурации или определение затрагиваемых работ в процессах, отчетность о состоянии сообщений о дефектах, утверждение и закрытие сообщений о дефектах;
- сообщения о дефектах, для которых требуются корректирующие действия в отношении программного средства или выходных данных процессов жизненного цикла ПО, должны активизировать работы по контролю изменений.

Примечание — Отчетность о дефектах и работы по контролю изменений являются взаимосвязанными.

Разработчик должен составлять сообщения о дефектах/изменениях, чтобы описать каждый дефект, обнаруженный в программных средствах, находящихся под контролем конфигурации, и каждую проблему выполнения работ, необходимых по контракту или описанных в Плане разработки ПО. Сообщения о дефектах/изменениях должны описывать дефекты/изменения, необходимые действия, связанные с коррекцией, а также предполагаемые сроки их выполнения. Эти сообщения следует использовать как входные данные для системы корректирующих действий.

Разработчик должен реализовать систему корректирующих действий для обработки каждого дефекта, обнаруженного в программных средствах, находящихся под контролем конфигурации, и каждую проблему выполнения работ, необходимых по контракту или описанных в Плане разработки ПО. Система должна отвечать следующим требованиям:

- входная информация системы должна состоять из сообщений о дефектах/изменениях;

- система должна быть закрытым циклом, гарантирующим, что все обнаруженные дефекты немедленно регистрируются и вводятся в систему, необходимые действия инициируются, принятые решения осуществляются, состояние корректирующих действий отслеживается и сообщения о дефектах сопровождаются в течение срока действия контракта;

- каждый дефект должен быть классифицирован по категориям и приоритетам;

- должен быть выполнен анализ для выявления возможных тенденций в зарегистрированных дефектах;

- корректирующие действия должны быть оценены, чтобы определить, были ли дефекты устранены, неблагоприятные тенденции преодолены, а изменения правильно выполнены без внесения дополнительных дефектов.

9.2.5 Контроль изменений и трассируемость

Цель контроля изменений — обеспечить регистрацию, оценку, рассмотрение и утверждение изменений на протяжении жизненного цикла ПО. Требования к выполнению работ по контролю изменений:

а) контроль изменений должен обеспечить целостность элементов конфигурации и базовых линий и защиту их от некорректных изменений;

б) контроль изменений должен гарантировать, что каждое изменение элемента конфигурации учтено в изменении идентификации конфигурации;

в) изменения в базовых линиях и элементах конфигурации, находящихся под контролем, должны быть зарегистрированы, утверждены и прослежены. Отчетность о дефектах связана с контролем изменений, поскольку устранение дефекта, который представлен в сообщении, может привести к изменениям элементов конфигурации или базовых линий.

Примечание — Общепризнанно, что ранняя реализация контроля изменений помогает управлению и организации работ в процессах жизненного цикла ПО;

г) изменения ПО должны быть прослежены вплоть до места их источника, а выполнение процессов жизненного цикла ПО необходимо повторить с момента, начиная с которого изменения сказываются на выходных данных. Так, например, ошибка, обнаруженная в интеграции ПО/аппаратуры, которая является результатом некорректного проектирования, должна повлечь за собой исправление проекта, исправление кода и повторение работ соответствующих интеграционных процессов;

д) при проведении работ по внесению изменений должны быть модифицированы документы жизненного цикла ПО, на которые эти изменения влияют, а обновление документов следует сопровождать работами по контролю изменений.

Работы по контролю изменений следует сопровождать работами по просмотру изменений.

9.2.6 Просмотр изменений

Цель работ по просмотру изменений — обеспечить оценку дефектов и изменений, их утверждение или неутверждение, реализацию утвержденных изменений и обратную связь с процессами, на которые изменения воздействуют, путем выполнения отчетности о дефектах и использования методов контроля изменений. Методы контроля определяют в процессе планирования ПО. Работы по просмотру изменений:

- подтверждение того, что затронутые изменениями элементы конфигурации идентифицированы;

- оценка воздействия изменений на требования безопасности и обеспечение обратной связи с процессом оценки безопасности системы;

- анализ дефектов или изменений и решений о действиях, которые следует предпринять для их коррекции;

- обеспечение обратной связи от сообщений о дефектах, контроля изменений и корректирующих действий к процессам, в которых реализуются изменения.

9.2.7 Отчет о состоянии конфигурации

Цель отчетности о состоянии конфигурации состоит в обеспечении информации для управления конфигурацией процессов жизненного цикла ПО относительно идентификации конфигурации, базовых линий, сообщений о дефектах и контроля изменений. Работы, связанные с отчетностью о состоянии конфигурации:

- регистрация идентификации элементов конфигурации, идентификации базовых линий, состояния сообщений о дефектах, хронологии изменений и состояния выпускаемой линии;

- определение информации, подлежащей сопровождению, и способов регистрации и ведения отчетности о состоянии конфигурации этой информации.

Разработчик должен создавать периодические отчеты о состоянии конфигурации всех объектов, которые были помещены под управление конфигурацией. Эти отчеты должны поддерживаться в течение срока действия контракта. Они должны включать в себя информацию о текущем состоянии базовой линии, ревизии, официальной выпускаемой версии каждого объекта и информацию об истории изменений объекта с момента его помещения под контроль конфигурации, а также о состоянии сообщений о дефектах /изменениях, связанных с данным объектом.

9.2.8 Архивирование и получение документов. Выпуск версии

Цель работ по архивированию и получению документов — обеспечить получение связанных с программным средством документов жизненного цикла ПО, которые необходимы для копирования, повторной генерации, повторного тестирования и модификации программного средства. Целью работы по выпуску версии является гарантирование использования, особенно для производства, исключительно санкционированного ПО, которое предварительно должно быть архивировано и официальная версия которого должна быть получена только из архива. Требования к выполнению работ данного вида:

а) документы жизненного цикла ПО, связанные с программным средством, должны быть получены из утвержденного источника (например, от организации-разработчика);

б) следует установить процедуры, обеспечивающие целостность хранимых данных (независимо от носителей данных), которые должны:

1) гарантировать, что никакое несанкционированное изменение не может быть выполнено;
2) выбирать носители данных, которые минимизируют ошибки регенерации и износа;
3) проверять и/или обновлять архивные данные с частотой, соответствующей сроку службы носителя;
4) хранить копии в физически отдаленных архивах, что минимизирует риск потери данных в катастрофической ситуации;

в) процесс копирования должен быть верифицирован, чтобы гарантировать получение точных копий, и должны существовать процедуры, гарантирующие безошибочное копирование исполняемого объектного кода;

г) элементы конфигурации должны быть идентифицированы и должна быть выпущена их официальная версия до того, как осуществляется производство ПО. Должны быть установлены полномочия по выпуску версий элементов конфигурации, как минимум, должен быть выполнен выпуск версий компонентов программного средства, загружаемого в вычислительную систему или оборудование;

д) необходимо установить процедуры хранения, включения, удаления и т.д., чтобы удовлетворить требования пригодности к применению и обеспечить модификацию ПО.

9.2.9 Контроль загрузки ПО

Цель работ по контролю загрузки ПО заключается в обеспечении загрузки исполняемого объектного кода в систему с соответствующей защитой. Контроль загрузки ПО относится к процессу, посредством которого программные инструкции и данные передаются из главного запоминающего устройства в вычислительную систему и оборудование. Используемыми методами могут быть, например, установка заранее запрограммированных в заводских условиях запоминающих устройств или повторное программирование управляющей системы или оборудования на месте с использованием устройства загрузки в полевых условиях (выбор метода является предметом для утверждения сертифицирующей организацией). Какой бы метод ни использовали, контроль загрузки должен включать в себя:

- процедуры присваивания регистрационных номеров и идентификации носителей данных, которые определяют конфигурацию ПО, предназначенного для загрузки в управляющую систему;
- обеспечение информации, подтверждающей совместимость ПО с управляющей системой и аппаратурой независимо от того, поставляют ли ПО в качестве конечного элемента или его устанавливают в вычислительной системе.

9.2.10 Контроль среды жизненного цикла ПО

Цель контроля среды жизненного цикла ПО — гарантировать, что инструментальные средства, используемые для создания ПО, идентифицируются, контролируются и могут быть получены из соответствующих источников. Инструментальные средства среды жизненного цикла ПО определяются в процессе планирования ПО и идентифицируются в документе «Указатель конфигурации среды жизненного цикла ПО» (12.25). Требования к выполнению работ данного вида:

- установка идентификации конфигурации для исполняемого объектного кода (или его эквивалента), используемого для разработки, управления, компоновки, верификации и загрузки ПО;

- контроль соответствия процесса управления конфигурацией для управления аттестованными инструментальными средствами целям, относящимся к категории 1 или 2 контроля документов (9.3), как определено в 13.2.3, перечисление б);

- если требования 9.2.9 неприменимы, то процесс управления конфигурацией для управления исполняемым объектным кодом (или его эквивалентом) для инструментальных средств, используемых для компоновки и загрузки ПО (например, компиляторов, ассемблеров, редакторов связей), должен соответствовать целям, относящимся, как минимум, к категории 2 контроля документов.

9.3 Категории контроля документов

Документы жизненного цикла ПО могут быть отнесены к одной из двух категорий: категории контроля 1 (КК1) и категории контроля 2 (КК2). Эти категории касаются функций управления конфигурацией в части документов. Таблица 1 определяет перечень целей процесса управления конфигурацией, связанных с каждой категорией контроля. В таблице 1 указано, какие функции управления конфигурацией должны быть выполнены для документов жизненного цикла ПО, относящихся к данной категории. Таблицы приложения А определяют категорию контроля каждого документа жизненного цикла ПО для уровней ПО. Для категорий контроля документов требуется:

- целевые функции процесса управления конфигурацией для документов жизненного цикла ПО, отнесенных к категории КК1, применять согласно таблице 1;

- целевые функции процесса управления конфигурацией для документов жизненного цикла ПО, отнесенных к категории КК2, применять согласно таблице 1, как минимум.

Таблица 1 — Целевые функции процесса управления конфигурацией, связанные с документами категорий КК1 и КК2

Цель процесса управления конфигурацией	Ссылка	КК1	КК2
Идентификация конфигурации	9.2.1	*	*
Базовая линия	9.2.3 а), б), в), г), д)	*	
Трассируемость	9.2.3 е), ж)	*	*
Отчетность о дефектах	9.2.4	*	
Контроль изменений — целостность и идентификация	9.2.5 а), б)	*	*
Контроль изменений — трассируемость	9.2.5 в), г), д)	*	
Просмотр изменений	9.2.6	*	
Отчетность о состоянии конфигурации	9.2.7	*	
Получение документа из архива	9.2.8 а)	*	*
Защита от несанкционированных изменений	9.2.8 б 1)	*	*
Выбор носителей, обновление, копирование	9.2.8 б 2), б 3), б 4), в)	*	
Выпуск версии	9.2.8 г)	*	
Хранение данных	9.2.8 д)	*	*

Обозначения:
* — цель должна быть удовлетворена для документов данной категории;
пробел — удовлетворение цели на усмотрение разработчика.

9.4 Аудит конфигурации

Разработчик должен поддерживать проводимый заказчиком аудит конфигурации, как определено в контракте.

9.5 Компоновка и поставка ПО

Разработчик должен устанавливать и выполнять процедуры по компоновке, хранению, обработке и поставке программного средства. Разработчик должен сохранять оригинал поставляемого программного средства в течение срока действия контракта.

10 Процесс обеспечения качества ПО

Процесс обеспечения качества ПО должен быть выполнен в соответствии с процессом планирования ПО (раздел 6) и документом «План обеспечения качества ПО» (12.6). Выходные результаты процесса обеспечения качества представлены в Протоколах обеспечения качества ПО (12.30) или в других документах жизненного цикла ПО. Процесс обеспечения качества оценивает процессы жизненного цикла ПО, их выходные результаты и гарантирует, что цели этих процессов

удовлетворены, отклонения от установленных требований обнаружены, оценены, прослежены, разрешены и что программные средства и документы жизненного цикла ПО соответствуют сертификационным требованиям. Работы процесса обеспечения качества должны быть выполнены разработчиком.

Если систему или ЭКПО разрабатывают для нескольких различных построений, работы и программные средства для каждого построения следует оценивать для целей данного конкретного построения. Работы или программное средство, соответствующее этим целям, можно считать удовлетворительным даже в случае отсутствия информации для разработки в более поздних построениях. Планирование обеспечения качества ПО в данном случае должно быть включено в планирование разработки ПО (6.6).

10.1 Цели процесса обеспечения качества ПО

Цель процесса обеспечения качества — обеспечить уверенность в том, что:

а) процессы разработки ПО и интегральные процессы выполняют по утвержденным планам ПО и стандартам;

б) критерии перехода для процессов жизненного цикла ПО удовлетворены;

в) просмотр соответствия программного средства выполнен для каждого программного средства, разрабатываемого в соответствии с требованиями настоящего стандарта и условиями контракта.

Применимость целей обеспечения качества для конкретного уровня ПО определена в таблице А.9.

10.2 Состав работ, выполняемых в процессе обеспечения качества ПО

Для того чтобы цели процесса обеспечения качества были выполнены:

а) процесс обеспечения качества должен играть активную роль в работах процессов жизненного цикла ПО на всех этапах жизненного цикла, обладая при этом необходимыми полномочиями, ответственностью и независимостью, чтобы гарантировать удовлетворение целям процесса обеспечения качества;

б) процесс обеспечения качества должен гарантировать, что планы ПО и стандарты разработаны и проверены на непротиворечивость;

в) процесс обеспечения качества должен гарантировать, что процессы жизненного цикла ПО выполнены в соответствии с утвержденными планами ПО и стандартами;

г) процесс обеспечения качества должен включать в себя аудиты процессов разработки ПО и интегральных процессов в течение жизненного цикла ПО, позволяющие гарантировать, что:

1) разработаны планы ПО, определенные в 6.2;
2) обнаружены, зарегистрированы, прослежены и утверждены заказчиком отклонения в выполнении требований планов ПО и стандартов;
3) зарегистрированы принятые отклонения;
4) обеспечена среда разработки ПО в соответствии с определением в планах ПО;
5) отчетность о дефектах, трассируемость и корректирующие действия соответствуют Плану управления конфигурацией ПО;
6) ввод данных, требуемых для процессов жизненного цикла ПО, находится под контролем постоянно выполняемого процесса оценки безопасности системы;

д) процесс обеспечения качества должен гарантировать, что критерии переходов для процессов жизненного цикла ПО были удовлетворены в соответствии с утвержденными планами ПО;

е) процесс обеспечения качества должен гарантировать, что для документов жизненного цикла ПО осуществлен контроль в соответствии с категориями контроля, определенными в 9.3 и таблицах приложения А;

ж) должен быть проведен просмотр согласованности ПО до поставки программных средств, представленных для сертификации;

з) должны быть выполнены работы, связанные с отчетностью по работам процесса обеспечения качества (12.30), включающие в себя результаты аудита и доказательство завершения просмотра согласованности ПО для каждого программного средства, представленного для сертификации.

10.3 Просмотр согласованности ПО

Цель просмотра согласованности ПО — гарантировать, что полностью подготовлены процессы и документы жизненного цикла ПО для сертифицируемого программного средства, а исполняемый объектный код находится под контролем конфигурации и может быть повторно генерирован. Этот просмотр должен определить, что:

- завершены запланированные работы процессов жизненного цикла ПО для получения сертификационного доверия, включая генерацию документов жизненного цикла ПО, и отчеты об их завершении сохранены;

- документы жизненного цикла ПО, которые разработаны в соответствии со специфицированными системными требованиями и с требованиями, определяемыми безопасностью, или требованиями к ПО, удовлетворяют этим требованиям;
- документы жизненного цикла ПО сформированы в соответствии с планами ПО и стандартами разработки (12.9 — 12.11) и проконтролированы в соответствии с Планом управления конфигурацией ПО;
- сообщения о дефектах подготовлены в соответствии с Планом управления конфигурацией ПО, их статус был оценен и зарегистрирован;
- отклонения реализации от требований к ПО зарегистрированы и утверждены;
- исполняемый объектный код может быть восстановлен из исходного текста, зарегистрированного в архиве;
- утвержденный код ПО может быть успешно загружен;
- сообщения о дефектах, оставшиеся от предыдущего просмотра согласованности ПО, заново переоцениваются, чтобы определить их состояние;
- если сертификационное доверие распространяется на использование ранее разработанного ПО, то текущая базовая линия программного средства должна быть трассируемой к предыдущей базовой линии и изменения в базовой линии должны быть утверждены.

10.4 Документирование обеспечения качества ПО

Разработчик должен регистрировать отчеты о каждой выполненной работе по обеспечению качества ПО. Эти отчеты следует сохранять в течение срока действия контракта. Проблемы, обнаруженные в программных средствах, при управлении конфигурацией должны быть обработаны, как описано в 9.2.4.

10.5 Независимость в обеспечении качества ПО

Лица, ответственные за обеспечение качества ПО, не должны участвовать в разработке программного средства или быть ответственными за программное средство или работы по его созданию. Однако это не запрещает таким лицам принимать участие в оценках программных средств или выполненных работ. Лица, ответственные за обеспечение качества ПО в соответствии с контрактом, должны иметь возможности, обязательства, полномочия и организационную независимость для объективной оценки качества ПО, а также для инициирования и верификации действий, связанных с коррекцией.

11 Процесс сертификационного сопровождения

Цель процесса сертификационного сопровождения — установить взаимодействие и взаимопонимание между соискателем и сертифицирующей организацией для поддержки процесса сертификации. Процесс сертификационного сопровождения выполняют так, как определено процессом планирования ПО (раздел 6) и Планом сертификации в части ПО (12.1). Таблица А.10 содержит резюме целей и результатов данного процесса.

11.1 Средства согласования и планирования

Соискатель предлагает средства согласования, которые показывают, что система будет удовлетворять сертификационному базису. План сертификации в части ПО (12.1) определяет аспекты ПО прикладной системы или оборудования применительно к предлагаемым средствам согласования. Этот план устанавливает также уровни ПО, которые определяются процессом оценки безопасности системы. Соискатель должен:

- передать План сертификации в части ПО и другие требуемые документы для просмотра сертифицирующей организации в тот момент времени, когда влияние изменений минимально, т.е. когда они могут быть проведены в рамках проектных ограничений;
- разрешить разногласия, идентифицируемые сертифицирующей организацией, касающиеся аспектов сертификации;
- получить согласие сертифицирующей организации на реализацию Плана сертификации в части ПО.

11.2 Обоснование согласованности

Соискатель должен обеспечить доказательства того, что процессы жизненного цикла ПО удовлетворяют требованиям планов ПО. Эти доказательства могут включать в себя просмотры и обсуждения работ, составляющих процессы жизненного цикла ПО, и, в случае необходимости, документов жизненного цикла ПО, проводимые с соискателем или его поставщиками. Соискатель должен:

- разрешить спорные вопросы, поднятые сертифицирующей организацией и являющиеся результатом выполненных ею просмотров;
- передать Итоговый документ разработки ПО (12.31) и Указатель конфигурации ПО (12.26) сертифицирующей организации;
- передать или сделать доступными другие документы и доказательства соответствия, требуемые сертифицирующей организацией.

11.3 Минимальный состав документов жизненного цикла ПО, передаваемых сертифицирующей организации

Минимальный состав документов жизненного цикла ПО, передаваемых сертифицирующей организации на утверждение:
- План сертификации в части ПО;
- Указатель конфигурации ПО;
- Итоговый документ разработки ПО.

11.4 Документы жизненного цикла ПО, относящиеся к типовому проекту

Если ничто другое не согласовано с сертифицирующей организацией, то правила получения и утверждения документов жизненного цикла ПО, связанных с типовым проектом, распространяются на следующие документы:
- Спецификация требований к ПО;
- Описание проекта ПО;
- Исходный код ПО;
- Исполняемый объектный код ПО;
- Указатель конфигурации ПО;
- Итоговый документ разработки ПО.

12 Документы, создаваемые в процессах жизненного цикла ПО

Документы создают в течение всего жизненного цикла ПО, чтобы планировать требуемые действия, управлять ими, объяснять, определять, регистрировать выполнение требуемых действий или обеспечивать доказательство процессов. Эти документы позволяют реализовать процессы жизненного цикла ПО, сертификацию системы и постсертификационную модификацию программного средства. Заказчик осуществляет выбор необходимого и экономически обоснованного состава и содержания документов для конкретной разработки. Заказчик разрешает любые конфликты между требованиями сертифицирующей организации и требованиями контракта. В настоящем стандарте не ставилась задача описать все документы, которые могут быть необходимы для разработки конкретного программного средства, и предложить конкретные методы объединения и организации информации. Дополнительно к документам, определяемым в указанных подразделах, могут быть подготовлены другие документы для поддержки процесса сертификации и удовлетворения требованиям контракта. В настоящем разделе обсуждаются характеристики, форма, методы контроля конфигурации и содержание документов жизненного цикла ПО. Характеристиками документов жизненного цикла ПО являются:
- однозначность: информация является однозначной, если она написана в терминах, которые допускают только единственную интерпретацию, уточненную, если необходимо, соответствующими определениями;
- полнота: информация является полной, если она включает в себя необходимые, релевантные требования и/или описательные материалы, определяет ответную реакцию для всего диапазона допустимых входных данных, используемые рисунки и таблицы сопровождаются необходимыми обозначениями, термины и единицы измерений определены;
- верифицируемость: информация является верифицируемой, если она может быть проверена на корректность человеком или инструментальным средством;
- согласованность: информация является согласованной, если не существует противоречий внутри нее;
- модифицируемость: информация является модифицируемой, если она структурирована и имеет такой стиль, что изменения могут быть выполнены в необходимом объеме, согласованно и корректно без нарушения структуры;
- трассируемость: информация является трассируемой, если для каждого ее компонента может быть определен первоисточник.

Дополнительные требования:
- форма: форма должна обеспечивать эффективный поиск и просмотр документов жизненного

цикла ПО в процессе обслуживания систем. Состав документов и их конкретная форма должны быть определены в Плане сертификации в части ПО.

Примечания

1 Документы жизненного цикла ПО могут иметь различные формы. Например, они могут быть подготовлены как компьютерный файл, хранящийся на магнитных носителях, или как отображение на удаленном терминале. Документация может быть оформлена в виде отдельных документов, может объединять несколько документов или быть разделена на несколько документов.

2 План сертификации в части ПО и Итоговый документ разработки ПО могут быть потребованы сертифицирующей организацией как отдельно напечатанные документы.

Документы жизненного цикла ПО могут быть отнесены к одной из двух категорий контроля в соответствии с применяемыми методами управления конфигурацией: категории контроля 1 (КК1) и категории контроля 2 (КК2) (9.3). Введение различных категорий контроля позволяет снизить стоимость разработки в случаях, когда менее строгий контроль может быть применен без снижения безопасности. Минимальная категория контроля для каждого документа и ее изменения в зависимости от уровня ПО определены в приложении А. Хотя назначение и содержание этих документов могут быть разными, к ним, как минимум, следует применять метод контроля КК2.

12.1 План сертификации в части ПО

План сертификации в части ПО, в первую очередь, предназначен для использования сертифицирующей организацией с целью определить, что предлагаемый соискателем жизненный цикл ПО соответствует требованиям для разработки ПО указанного уровня. Этот план должен иметь следующие разделы:

а) Обзор системы. Этот раздел представляет обзор системы, включающий в себя описание ее функций и их распределения между аппаратным и программным обеспечением, архитектуру, используемые процессоры, аппаратно-программные интерфейсы и особенности обеспечения безопасности.

б) Обзор ПО. Этот раздел кратко описывает функции ПО, уделяя особое внимание предлагаемым концепциям обеспечения безопасности и разбиения структуры, например совместное использование ресурсов, резервирование, многоверсионное программирование, обеспечение отказоустойчивости, стратегия синхронизации и планирования выполнения ПО.

в) Вопросы сертификации. Этот раздел описывает сертификационный базис, включая средства доказательства соответствия разработки ПО требованиям сертификации ПО; раздел также устанавливает предлагаемые уровни ПО (уровни критичности) и суммирует пояснения, обеспечивающие процесс оценки безопасности системы, включая потенциальный вклад ПО в создание отказных ситуаций.

г) Жизненный цикл ПО. Этот раздел определяет используемую модель жизненного цикла ПО, которая должна быть выполнена и которую контролируют процессы жизненного цикла ПО, детализируемая информация для последних определена в соответствующих планах ПО. Данный раздел поясняет то, каким образом должны быть удовлетворены цели каждого процесса жизненного цикла, и точно определяет организации, участвующие в разработке, организационную ответственность, а также ответственность за процессы жизненного цикла системы и процесс сертификационного взаимодействия.

д) Документы жизненного цикла ПО. Этот раздел точно специфицирует документы жизненного цикла ПО, которые должны быть разработаны и должны контролироваться процессами жизненного цикла ПО. Данный раздел также описывает отношения между этими документами или другими документами, определяющими систему, документами жизненного цикла, представляемыми на рассмотрение сертифицирующей организации, форму документов и способ, посредством которого документы жизненного цикла становятся доступными для сертифицирующей организации.

е) План-график. Этот раздел описывает средства соискателя, которые должны обеспечивать прозрачность работ процессов жизненного цикла ПО для сертифицирующей организации (в целях планирования просмотров).

ж) Дополнительные вопросы. Этот раздел описывает специфические особенности, которые могут влиять на процесс сертификации, например:

1) альтернативные методы согласования;
2) аттестация инструментальных средств;
3) использование ранее разработанного ПО;
4) использование ПО, разработанного в необязательном порядке;

5) использование модифицируемого пользователем ПО;
6) использование коммерчески доступного ПО;
7) использование ПО, загружаемого в полевых условиях;
8) использование многоверсионного неидентичного ПО.

12.2 План разработки ПО

План разработки ПО содержит описание целей, стандартов и модели жизненного цикла ПО, которые должны быть использованы в процессах разработки ПО. Этот план может быть включен в План сертификации в части ПО. План разработки ПО должен включать в себя следующие разделы:

а) Стандарты: идентификация стандартов на разработку требований к ПО, стандартов на процесс проектирования ПО, стандартов кодирования ПО для данного проекта, а также ссылки на стандарты для ранее разработанного ПО, включая коммерчески доступное ПО, если эти стандарты различаются.

б) Жизненный цикл ПО: описание процессов жизненного цикла ПО, которые должны быть использованы для формирования конкретного жизненного цикла данного проекта, включая критерии перехода между процессами ПО. Это описание отличается от резюме в Плане сертификации в части ПО тем, что оно содержит подробности, необходимые для гарантии соответствующей реализации процессов жизненного цикла ПО.

в) Среда разработки ПО: обоснование выбора используемой среды разработки ПО в аппаратной и программной частях, включая:

1) выбор методов и средств разработки требований;
2) выбор методов и средств проектирования ПО;
3) выбор языков программирования, средств кодирования, компиляторов, редакторов связей и загрузчиков;
4) аппаратную поддержку для инструментальных средств.

12.3 План верификации ПО

План верификации ПО включает в себя описание процедур верификации, удовлетворяющих целям процесса верификации. Эти процедуры могут варьироваться в зависимости от уровня ПО, как определено в таблицах приложения А. Данный план должен включать в себя следующие разделы:

а) Организация: организационная ответственность внутри процесса верификации ПО и интерфейсы с другими процессами жизненного цикла ПО.

б) Независимость: описание методов для обеспечения независимости верификации, когда это требуется.

в) Методы верификации: описание методов верификации, которые будут использованы на каждом этапе процесса верификации ПО:

1) методы просмотра, включающие в себя контрольные листы и другие средства поддержки;
2) методы анализа, включающие в себя методы анализа трассируемости и оценки полноты покрытия;
3) методы тестирования, включающие в себя рекомендации для выбора тестовых вариантов, используемых тестовых процедур, генерации тестовых данных.

г) Среда верификации: описание оборудования для тестирования, инструментальных средств тестирования и анализа, а также руководств по применению этих средств и аппаратного тестового оборудования.

д) Критерии перехода: критерии перехода к процессу верификации ПО, определяемому в этом плане.

е) Проверка разбиения: если используют разбиение на части, то описывают метод верификации целостности.

ж) Допустимость использования компилятора: описание соглашений относительно корректности применения компилятора, редактора связей или загрузчика (6.4.2).

з) Руководство по повторной верификации: описание методов идентификации модифицируемых областей ПО и измененных частей исполняемого объектного кода. Повторная верификация должна гарантировать, что ранее зарегистрированные ошибки или классы ошибок были устранены.

и) Ранее разработанное ПО: если для базовой линии ранее разработанного ПО требования к процессу верификации не согласуются с требованиями данного документа, приводят описание методов верификации, удовлетворяющих этим требованиям.

к) Многоверсионное ПО: при использовании многоверсионного ПО необходимо описание работ процесса верификации для него.

12.4 План квалификационного тестирования ПО

План квалификационного тестирования ПО содержит информацию для проведения квалификационного тестирования (испытаний) систем и подсистем ПО, описание тестовой среды, которая будет использована при тестировании, идентифицирует выполняемые тесты и указывает план-график выполнения тестирования.

Для каждой предполагаемой тестовой установки должны быть указаны:

- идентификация, перечень и используемые версии ПО, для которых будет выполнено тестирование на данной установке, их назначение;
- идентификация, перечень и используемые виды аппаратных средств, интерфейсного оборудования, устройств связи, дополнительных внешних устройств, генераторов тестовых сообщений, устройств синхронизации тестов и т.п.;
- права собственности и лицензирование;
- организации, принимающие участие в квалификационном тестировании, их роли и ответственность.

Кроме того, в данном документе должны быть представлены план-график тестирования и матрица трассирования тестов к требованиям к ПО.

Допускается включение перечисленной в настоящем подразделе информации в документ «План верификации ПО» (см. 12.3), если заказчик не требует разработки отдельного документа, описывающего план квалификационного тестирования.

12.5 План управления конфигурацией ПО

План управления конфигурацией ПО устанавливает методы, используемые для достижения целей процесса управления конфигурацией ПО на протяжении жизненного цикла ПО. Разделы плана следующие:

- Среда: описание среды управления конфигурацией, которая будет использована, включая процедуры, инструментальные средства, методы, стандарты, организационную ответственность и интерфейсы.
- Состав работ: описание работ процесса управления конфигурацией в жизненном цикле ПО, которые обеспечат реализацию целей данного процесса.
- Идентификация конфигурации: элементы конфигурации, которые должны быть идентифицированы; срок, когда они будут идентифицированы; методы идентификации документов жизненного цикла ПО (например, регистрационные номера) и связь идентификации ПО и системы.
- Базовая линия и трассируемость: средства установки базовой линии, как базовая линия будет установлена, когда эта базовая линия будет установлена, средства управления библиотекой ПО и трассируемость элементов конфигурации и базовой линии.
- Отчетность о дефектах: содержание и идентификация сообщений о дефектах для программного средства и процессов жизненного цикла, в каких случаях они должны быть оформлены, процедуры закрытия сообщений о дефектах и взаимодействие отчетности о дефектах с контролем изменений.
- Контроль изменений: элементы конфигурации и базовая линия, которые следует контролировать, в каких случаях они должны быть проконтролированы, работы по контролю дефектов/изменений, предсертификационный и постсертификационный контроль, средства, обеспечивающие целостность элементов конфигурации и базовой линии.
- Просмотр изменений: метод установления обратной связи с процессами жизненного цикла ПО; методы оценки и определения приоритетности в устранении дефектов, утверждение изменений, реализация решений об изменениях и связь этих методов с отчетностью о дефектах и работами по контролю за изменениями.
- Отчет о состоянии конфигурации: информация, которая должна быть зарегистрирована, чтобы можно было осуществлять отчетность о состоянии управления конфигурацией, определение места хранения информации, как она будет воспроизведена для отчетности и когда она будет доступна.
- Архивирование, получение из архива и выпуск официальной версии: контроль целостности, способы внесения информации в архив и получения из архива, метод и полномочия для выпуска версии.
- Контроль загрузки ПО: описание защиты и регистрации контроля загрузки ПО.
- Контроль среды жизненного цикла ПО: контроль инструментальных средств, используемых для разработки, комплексирования, верификации и загрузки ПО. Кроме того, в раздел должен быть включен контроль аттестованных инструментальных средств.

- Контроль документов жизненного цикла ПО: средства контроля документов, требуемые для категорий контроля 1 и 2.
- Критерии перехода: критерии перехода для начала процесса управления конфигурацией.
- Документы управления конфигурацией: определение документов жизненного цикла ПО, генерируемых в процессе управления конфигурацией, включая отчеты управления конфигурацией, указатель конфигурации ПО и указатель среды жизненного цикла ПО.
- Контроль поставщика: использование требований процесса управления конфигурацией для контроля поставщика.

12.6 План обеспечения качества ПО

План обеспечения качества ПО устанавливает методы, которые должны быть использованы для того, чтобы достичь цели процесса обеспечения качества ПО. Содержание плана следующее:
- Среда: описание среды обеспечения качества, включая область действия, организационную ответственность и интерфейсы, стандарты, процедуры, инструментальные средства и методы.
- Полномочия: утверждение полномочий службы обеспечения качества, ответственности и независимости, включая полномочия на утверждение (одобрение) программных средств.
- Состав работ: работы обеспечения качества, которые должны быть выполнены для каждого процесса жизненного цикла ПО и на протяжении всего жизненного цикла ПО, включая:
  1) методы обеспечения качества, например просмотры, аудиты, отчетность, инспекции и мониторинг процессов жизненного цикла ПО;
  2) работы, связанные с отчетностью о дефектах, трассируемостью и системой корректирующих действий;
  3) описание работ во время просмотров согласованности ПО.
- Критерии перехода: критерии перехода для начала процесса обеспечения качества.
- Синхронизация: синхронизация работ процесса обеспечения качества относительно работ других процессов жизненного цикла ПО.
- Отчеты обеспечения качества: определение отчетов, которые будут произведены процессом обеспечения качества.
- Контроль поставщика: описание средств, гарантирующих, что действия поставщиков и результаты их работы соответствуют Плану обеспечения качества ПО.

12.7 План установки ПО

План установки ПО содержит описание работ для установки ПО на пользовательских местах, включая подготовку, обучение пользователей и адаптацию существующих систем.

Данный план необходим, когда разработчик должен выполнить установку ПО на пользовательских местах и когда процесс установки ПО настолько сложен, что без оформленного в виде документа плана обойтись невозможно.

План установки ПО включает в себя:
- перечень пользовательских мест, на которых должно быть установлено ПО;
- запланированные сроки установки ПО;
- методы установки ПО;
- организационные сведения: номер телефона, факс, официальное наименование организации, осуществляющей установку, и т.д.;
- технические средства поддержки: перечень всех типов, характеристик и источников средств, необходимых для установки ПО (магнитные ленты, диски, бумага для принтера и т.д.);
- организация процесса обучения персонала: классные комнаты, расписание теоретических и практических занятий и т.д.

12.8 План передачи ПО

План передачи ПО определяет аппаратное и программное обеспечение, а также другие ресурсы, необходимые для поддержки жизненного цикла передаваемого ПО, и описывает планы разработчиков для поставки передаваемых элементов через организации, осуществляющие поддержку.

Данный план разрабатывают в том случае, если используют концепцию передачи ПО отдельной организации, осуществляющей поддержку.

План должен содержать краткий обзор системы и документов, относящихся к передаваемому ПО, общий обзор разработки системы и сопровождения, идентифицировать спонсоров, заказчиков, пользователей, разработчиков и организаций, осуществляющих поддержку, запланированные рабочие места и перечень передаваемых документов.

План содержит детальное описание ресурсов, необходимых для поддержки передаваемого ПО, требования к квалификации и составу персонала. Такие ресурсы должны включать в себя элементы,

необходимые для копирования, контроля и распространения ПО и соответствующей документации, а также чтобы специфицировать, разрабатывать, документировать, тестировать, оценивать, контролировать, копировать и распространять ПО.

План содержит перечень рекомендуемых мероприятий, в том числе консультации и лекции, которые должен проводить разработчик в целях поддержки передаваемого ПО и соответствующей среды поддержки.

План включает в себя описание процесса подготовки персонала, который будет осуществлять поддержку передаваемого ПО: тематика, дата, продолжительность и место проведения занятий по подготовке как теоретических, так и практических, в том числе знакомство с системным ПО, объектными компьютерами, программной поддержкой и базовой системой.

В плане должны быть указаны предполагаемые области изменений передаваемого ПО.

План содержит порядок передачи, включающий в себя все работы, необходимые при передаче ПО со стороны организаций, осуществляющих поддержку, с детальной проработкой координационных встреч.

12.9 Стандарты на разработку требований к ПО

Цель стандартов на разработку требований к ПО состоит в том, чтобы определить методы, правила и инструментальные средства, которые должны быть использованы при разработке требований верхнего уровня. Эти стандарты должны включать в себя:

- методы, которые должны быть применены для разработки требований к ПО;
- системы обозначений, которые применяют для описания требований, такие как диаграммы потока данных и формальные языки спецификаций;
- ограничения на использование инструментальных средств разработки требований;
- метод, который должен быть применен для получения производных требований.

12.10 Стандарты на процесс проектирования ПО

Цель стандартов на процесс проектирования ПО состоит в определении методов, правил и инструментальных средств, которые следует применять для разработки архитектуры ПО и требований нижнего уровня. Эти стандарты должны содержать:

- методы описания проекта, которые будут использованы;
- соглашения по наименованию;
- ограничения, налагаемые на применяемые методы проектирования, например распределение ресурсов, использование прерываний и структур, управляемых событиями, использование динамических задач, повторный вход, использование глобальных данных, механизм обработки исключительных ситуаций и обоснования для их использования;
- ограничения на использования инструментальных средств проектирования;
- ограничения на проектирование (например, запрещение использования рекурсий, динамических объектов, альтернативных имен, сокращенных выражений);
- ограничения по сложности (например, максимальный уровень вложенности вызовов и условных структур, использование безусловных переходов, число входных/выходных точек элементов кода программы).

12.11 Стандарты кодирования ПО

Целью стандартов кодирования ПО является определение языков программирования, методов, правил и инструментальных средств, которые будут использованы для кодирования ПО. Стандарты кодирования должны включать в себя:

- используемые языки программирования и/или какое-либо их заданное подмножество; должна быть указана ссылка на документы, которые однозначно определяют синтаксис, режим контроля, характер данных и побочные эффекты языка программирования; стандарты могут требовать ограничений на использование некоторых возможностей языка;
- стандарты представления исходного текста (например, ограничение на длину строки, структурное расположение текста, использование пустых строк) и стандарты документирования исходного кода (например, имя автора, история изменений, входные и выходные данные, а также наиболее значимые глобальные данные);
- соглашения по наименованию для компонентов, подпрограмм, переменных, констант;
- условия и ограничения, налагаемые на установленные соглашения кодирования, такие как информационная связность между компонентами ПО, сложность логических или числовых выражений, а также обоснования для их использования;
- ограничения на использование инструментальных средств кодирования.

12.12 Спецификация системы/подсистемы

Спецификация системы/подсистемы определяет требования для системы или подсистемы и методы, которые должны быть использованы для гарантии того, что каждое требование выполнено. Требования, относящиеся к внешним интерфейсам системы или подсистемам, должны быть представлены либо в данной спецификации, либо в спецификации требований к интерфейсу, на которую должны быть ссылки в спецификации системы/подсистемы.

Каждое требование соответствует конкретным обоснованным характеристикам системы, имеет уникальный для проекта идентификатор, чтобы можно было провести тестирование и проследить его выполнение с помощью объективного теста. Для каждого требования выбирают квалификационный(е) метод(ы), требования для подсистемы должны быть прослеживаемы к требованиям к системе. Степень детализации выбирают, исходя из следующих правил: указывают те характеристики системы, которые внесены в условия приемки системы; предпочтение отдают тем характеристикам, которые требует обеспечить заказчик.

Должны быть описаны требования:

- к режимам работы;
- к производительности системы;
- к внешнему интерфейсу системы;
- к внутреннему интерфейсу системы;
- к внутренним данным системы;
- по адаптации;
- по безопасности;
- по обеспечению защиты и секретности;
- к системному окружению (среде);
- к ресурсам вычислителя (к аппаратуре, коэффициенту использования ресурсов аппаратуры, ПО вычислителя, организации сети компьютеров, если она необходима);
- по ограничениям проекта;
- по обучению персонала.

Должны быть также определены:

- относительная важность и критичность требований;
- средства аттестации, включающие в себя демонстрацию, тестирование, анализ, инспекцию и требуемые специальные методы для конкретной системы.

Все указанные требования должны быть трассируемыми.

12.13 Спецификация требований к ПО

Спецификация требований к ПО — документ, который определяет требования верхнего уровня, включая производные требования. Этот документ должен содержать:

- описание распределения системных требований по компонентам ПО с учетом требований, которые обеспечивают безопасность, и потенциальным отказным ситуациям;
- функциональные и эксплуатационные требования для каждого режима работы;
- критерии производительности, например точность представления;
- временные требования и ограничения;
- ограничения объема памяти;
- интерфейсы аппаратуры и ПО, например протоколы, форматы, частоту ввода и вывода данных;
- требования к обнаружению отказов и мониторингу безопасности;
- требования по разбиению ПО, по взаимодействию выделенных компонентов ПО друг с другом и к уровням ПО для каждой части.

12.14 Спецификация требований к интерфейсу

Спецификация требований к интерфейсу определяет требования к интерфейсам между системными компонентами: системами, подсистемами, элементами конфигурации ПО и аппаратуры. Данный документ включает в себя:

- идентификацию и диаграммы функционирования интерфейсов;
- требования по типам интерфейсов;
- характеристики передаваемых элементов данных (идентификация, типы данных, размер и формат, единицы измерений, точность, источники и приемники);
- характеристики методов коммуникации;
- характеристики протоколов обмена;
- приоритеты и критичность требований;

- методы аттестации, которые должны быть использованы для демонстрации выполнения требований;
- доказательство трассируемости требований.

12.15 Описание проекта системы/подсистемы

Документ «Описание проекта системы/подсистемы» описывает проект системы/подсистемы как целого, а также проект архитектуры системы/подсистемы, может быть дополнен описанием проекта интерфейса и описанием проекта базы данных. Данный документ включает в себя:

- обоснование выбора проектных решений уровня системы, выбора компонентов системы, описание поведения системы с точки зрения пользователя;
- проект архитектуры системы, содержащий идентификацию компонентов системы, их назначение, статус/тип разработки, аппаратные ресурсы;
- концепцию совместного функционирования компонентов, описание их динамических связей;
- описание интерфейсов между компонентами;
- анализ трассируемости проекта системы к системным требованиям.

Данный документ содержит обоснование выбора конкретной системы/подсистемы с учетом требований интерфейса, заданных характеристик входов и выходов, физической модели системы, выбранных соотношений/алгоритмов/правил и обработки недопустимых входных данных или условий.

12.16 Описание проекта ПО

Документ «Описание проекта ПО» содержит описание архитектуры и требований нижнего уровня к ПО, которые должны удовлетворять требованиям верхнего уровня к ПО. Этот документ должен включать в себя:

- детализированное описание того, как ПО удовлетворяет специфицированным требованиям верхнего уровня к ПО, включая алгоритмы, структуры данных, и описание распределения по процессорам и задачам требований к ПО;
- описание архитектуры ПО, которая определяет структуру ПО, предназначенного для реализации заданных требований;
- описание входных/выходных данных (например, словарь данных) для внутренних и внешних интерфейсов архитектуры ПО;
- описание потока данных и потока управления;
- ограничения на использование ресурсов, стратегию для управления каждым ресурсом, границы рабочего диапазона и методы измерения этих границ, например времени выполнения и памяти;
- процедуры планирования и механизмы межпроцессорной и межзадачной связей, включая жесткую временную последовательность, приоритетное планирование, механизмы рандеву в языке Ада и прерывания;
- методы, используемые в проекте, и детали их реализации, например загрузку данных ПО, модифицируемое пользователем ПО или многоверсионное неидентичное ПО;
- методы разбиения ПО и средства обеспечения целостности при разбиении;
- описания компонентов ПО с указаниями о том, являются ли они вновь создаваемыми или ранее разработанными и, если они ранее разработаны, делаются ли ссылки на базовую линию;
- производные требования, полученные в процессе проектирования ПО;
- для отключенного кода описание средств, гарантирующих невозможность его непреднамеренного выполнения;
- обоснование тех решений проекта, которые относятся к требованиям, связанным с безопасностью системы.

12.17 Описание проекта интерфейса

Описание проекта интерфейса содержит описание характеристик интерфейсов одной или более систем, подсистем, элементов конфигурации аппаратуры и ПО и других системных компонентов. Данный документ включает в себя:

- идентификацию и диаграммы всех интерфейсных связей;
- приоритеты и типы интерфейсов;
- характеристики передаваемых данных, методов коммуникации, протоколов.

12.18 Описание проекта базы данных

Описание проекта базы данных включает в себя описание базы данных, рассматриваемой как набор связанных данных, хранящихся в одном или более компьютерных файлах, доступ к которым пользователем осуществляется через систему управления базами данных.

В данном документе должны быть описаны проектные решения, связанные с базой данных, поведение базы данных с точки зрения пользователя, способы доступа к базе данных, интерфейсы базы данных с другими системами, элементами конфигурации ПО и аппаратуры. Определяется реакция базы данных на входные запросы, включая действия, время реакции и другие эксплуатационные характеристики, выбранные соотношения/алгоритмы/правила/обработка недопустимых входных данных.

Детальный проект базы данных содержит характеристики элементов данных, проект программных модулей, осуществляющих доступ к базе данных или работу с ней, алгоритмы работы с базой, возможные ограничения, язык программирования, интерфейсы между программными модулями, характеристики интерфейсов, характеристики методов коммуникации, характеристики протоколов.

В данном документе должна быть показана трассируемость проектных требований к системным требованиям и требованиям к ПО.

12.19 Исходный код ПО

Этот документ содержит код ПО, написанный на исходном(ых) языке(ах) программирования, и команды компилятора, генерирующие объектный код из исходного текста, а также информацию для редактирования связей и загрузки. Документ должен содержать идентификацию ПО, включая идентификатор и дату создания версии.

12.20 Исполняемый объектный код ПО

Исполняемый объектный код представляет собой код, который является непосредственно пригодным для использования центральным процессором объектного компьютера, и является, следовательно, загружаемым в аппаратные средства или систему ПО.

12.21 Процедуры верификации ПО

Процедуры верификации ПО детально описывают выполнение работ процесса верификации ПО. Данный документ должен включать в себя следующие описания:

- процедуры выполнения просмотра и анализа: детализация информации Плана верификации ПО в части области действия, глубины методов просмотров или анализа;
- тестовые варианты: назначение каждого тестового варианта, набор входных данных, условия, ожидаемые результаты, требуемые критерии покрытия и критерии прохода/невыполнения тестов;
- процедуры тестирования: пошаговые инструкции того, как каждый тестовый вариант должен быть инициирован и выполнен, как должны быть оценены результаты тестирования и какая среда тестирования должна быть использована.

12.22 Описание квалификационного тестирования ПО

Данный документ описывает организацию квалификационного тестирования (испытаний), тестовые варианты и процедуры тестирования, которые используют для выполнения квалификационного тестирования системы или подсистемы ПО.

Организация тестирования: каждый тест должен иметь уникальный для данного проекта идентификатор; должны быть представлены инструкции для проведения тестирования, описание аппаратуры и ПО тестирования, а также инструкции для выполнения повторного тестирования.

Описание тестов: каждый тест должен иметь уникальный для проекта идентификатор и ссылку на соответствующий пункт в разделе организации тестирования, последнее относится и к тестовым вариантам. Кроме того, должны быть приведены ссылки на проверяемые требования, указаны условия выполнения (конфигурация аппаратуры и ПО и др.), входные данные, ожидаемые результаты, критерии оценки результатов, процедура тестирования для каждого тестового варианта, допущения и ограничения.

Допускается включение перечисленной в данном подразделе информации в документ «Процедуры верификации ПО» (см. 12.21), если заказчик не требует разработки отдельного документа, описывающего квалификационное тестирование.

12.23 Результаты верификации ПО

Документ «Результаты верификации ПО» описывает выходные результаты процесса верификации ПО. Результаты верификации ПО должны:

- указать результат выполнения (прошел/не прошел) для каждого просмотра, анализа и выполненного теста и заключительный результат верификации;
- идентифицировать элемент конфигурации и/или версию ПО, которые прошли просмотр, анализ или тестирование;
- содержать результаты анализа покрытия и анализа трассируемости для тестов, просмотров и анализов, выполненных в процессе верификации.

12.24 Отчет о квалификационном тестировании ПО

Отчет о тестировании ПО — отчет о квалификационном тестировании (испытаниях), выполненном для системы или подсистемы ПО. Данный документ должен включать в себя:

- общую оценку результатов тестирования, идентификацию всех несоответствий и ограничений;
- описание возможных различий тестовой и эксплуатационной сред;
- описание рекомендуемых улучшений в тестируемом ПО;
- детальные результаты тестирования;
- описание обнаруженных дефектов.

Допускается включение перечисленной в данном разделе информации в документ «Результаты верификации ПО» (12.23), если заказчик не требует разработки отдельного документа, описывающего результаты квалификационного тестирования.

12.25 Указатель конфигурации среды жизненного цикла ПО

Документ «Указатель конфигурации среды жизненного цикла ПО» идентифицирует конфигурацию среды жизненного цикла ПО, определяет аппаратную и программную среду разработки для регенерации ПО, повторной верификации или модификации ПО. Этот указатель должен идентифицировать:

- аппаратную среду и системное ПО, используемые для разработки ПО на протяжении всего жизненного цикла;
- инструментальные средства разработки ПО, такие как компиляторы, редакторы связей и загрузчики, средства обеспечения целостности данных (такие, как средства вычисления контрольных сумм или циклического избыточного кода);
- среду тестирования, используемую для верификации программного средства, например инструментальные средства верификации ПО;
- аттестованные инструментальные средства и соответствующие документы об аттестации этих средств.

Примечание — Данный документ может быть включен в Указатель конфигурации ПО как его часть.

12.26 Указатель конфигурации ПО

Указатель конфигурации ПО определяет конфигурацию программного средства. Указатель конфигурации ПО должен идентифицировать:

- программное средство;
- исполняемый объектный код;
- каждый компонент исходного кода;
- ранее разработанное ПО, если оно используется в данном программном средстве;
- документы жизненного цикла ПО;
- носители данных для архива и выпуска версии;
- инструкции для компоновки исполняемого объектного кода, включая, например, инструкции и информацию для компилирования и редактирования связей; процедуры, используемые для восстановления ПО при регенерации, тестировании или модификации;
- ссылку на Указатель конфигурации среды жизненного цикла ПО (12.25), если он оформлен как отдельный документ;
- способ контроля целостности данных для исполняемого объектного кода, если он используется.

Примечание — Данный документ может быть создан для одной версии программного средства или может включать в себя информацию о последующих или альтернативных версиях программного средства.

12.27 Спецификация программного средства

Спецификация программного средства содержит описание или ссылки на описания исполняемого ПО, исходных файлов и информацию о программной реализации, включая информацию проекта построения, компиляции, построения и процедуры модификации для ЭКПО.

Спецификация программного средства должна содержать описание:

- требований, включающих в себя обеспечение передачи ПО и обоснование требований, которым должна соответствовать достоверная копия ЭКПО;
- методов, используемых для демонстрации того, что данное ПО является достоверной копией ЭКПО.

12.28 Сообщения о дефектах

Сообщения о дефектах являются средством для идентификации и регистрации аномального поведения программного средства, несогласованности процессов с планами ПО и стандартами

разработки ПО и недостатков документации жизненного цикла ПО. Сообщения о дефектах должны включать в себя:

- идентификацию элемента конфигурации и/или этапа жизненного цикла ПО, где был обнаружен дефект;
- идентификацию элемента конфигурации, который необходимо модифицировать, или описание процесса, который должен быть изменен;
- описание дефекта, достаточное для его понимания и устранения;
- описание корректирующих действий, предназначенных для устранения зарегистрированного дефекта.

12.29 Протоколы управления конфигурацией ПО

Результаты работ процесса управления конфигурацией ПО должны быть зарегистрированы в протоколах управления конфигурацией ПО. Они включают в себя, например, все идентификации конфигурации, протоколы об установлении базовой линии и регистрации в библиотеке, отчеты об истории изменений, протоколы о передаче в архив и протоколы о выпуске версии. Приведенные выше примеры не содержат всех конкретных типов информации, которую необходимо представлять в указанных документах.

Примечание — Поскольку процесс управления конфигурацией ПО интегральный по своей природе, результаты его часто включают как составную часть в другие документы жизненного цикла ПО.

12.30 Протоколы обеспечения качества ПО

Результаты работ процесса обеспечения качества ПО должны быть зарегистрированы в протоколах обеспечения качества. Они могут включать в себя протоколы просмотров и аудитов, протоколы совещаний, регистрацию отклонений от санкционированных процессов или протоколы проверки соответствия ПО.

12.31 Итоговый документ разработки ПО

Итоговый документ разработки ПО — основной документ по демонстрации соответствия Плану сертификации в части ПО. Этот документ должен содержать следующие разделы:

- Краткий обзор системы. Данный раздел содержит краткий обзор системы, включая описание ее функций и их распределение на программную и аппаратную реализацию, архитектуру, используемые процессоры, интерфейсы аппаратных средств/ПО, требования по обеспечению безопасности. Этот раздел также описывает все отличия от краткого обзора системы в Плане сертификации в части ПО.
- Краткий обзор ПО. Этот раздел кратко описывает функции ПО с акцентированием на обеспечении безопасности и используемой концепции разбиения и объясняет отличия от краткого обзора ПО в Плане сертификации в части ПО.
- Вопросы сертификации. Этот раздел вновь рассматривает вопросы сертификации, определенные в Плане сертификации в части ПО, и объясняет все существующие от указанного плана отличия.
- Характеристики ПО. В этом разделе указаны размер исполняемого объектного кода, ограничения по времени и памяти, ограничения ресурсов и способы измерения каждой характеристики.
- Жизненный цикл ПО. Этот раздел описывает фактически используемую модель жизненного цикла ПО и объясняет ее отличия от предложенной в Плане сертификации в части ПО.
- Документы жизненного цикла ПО. В этом разделе даны ссылки на документы жизненного цикла ПО, являющиеся выходными результатами процессов разработки ПО и интегральных процессов. Здесь описаны связь между представляемыми документами и другими документами, определяющими систему, а также способы передачи документов жизненного цикла ПО сертифицирующей организации. В этом разделе также рассмотрены любые отклонения в описании документов от Плана сертификации в части ПО.
- Идентификация ПО. Этот раздел идентифицирует конфигурацию ПО посредством указания регистрационного номера и версии.
- Хронология изменения. В случае необходимости этот раздел включает в себя резюме изменений ПО с указанием изменений, вызванных отказами, влияющими на безопасность, и идентификацией изменений, выполненных после предыдущей сертификации.
- Текущее состояние ПО. Этот раздел содержит резюме сообщений о дефектах, не устраненных ко времени сертификации, включая заявления о функциональных ограничениях.
- Утверждение о соответствии. Этот раздел включает в себя утверждение о соответствии требованиям настоящего стандарта и резюме методов, позволяющих показать выполнение критери-

ев, определенных в планах ПО. Этот раздел также указывает дополнительные соглашения и отклонения от требований планов, стандартов разработки и настоящего стандарта.

12.32 Описание эксплуатационной концепции

Описание эксплуатационной концепции для системы управления содержит описание действий пользователя, необходимых для работы с предлагаемой системой, ее связи с существующими системами и процедурами. Данное описание используют для получения соглашения между поставщиком, разработчиком, организацией, осуществляющей поддержку, и пользователями.

Данный документ фиксирует текущее состояние системы, ее назначение, возможности и ограничения в зависимости от режима или конкретного состояния эксплуатации (например, стандартный режим, сопровождение, обучение, снижение функций, аварийные ситуации) и включает в себя описание:

- конкретной эксплуатационной среды и ее характеристики;
- основных компонентов системы и связей между ними;
- внешних интерфейсов системы;
- возможностей/функций системы;
- таблиц и дополнительных графических представлений входов, выходов, потоков данных, а также руководств, позволяющих разобраться в текущем состоянии системы с точки зрения пользователя;
- состава персонала, его организационной структуры, технической подготовки, обязанностей, взаимодействия;
- критериев ремонта/замены;
- уровней и циклов технического обслуживания;
- форм регистрации обнаруженных дефектов.

Данный документ также содержит:

- соглашения о внесении изменений, возникающих в процессе сопровождения (их классификация и порядок внесения, включая поставку необходимого оборудования и обучение персонала);
- концепцию поставки новой или модифицированной версии, эксплуатационный сценарий;
- информацию о взаимодействии пользователей, поставщика, разработчика и организации, осуществляющей поддержку, во время эксплуатационного периода.

12.33 Руководство по эксплуатации компьютера

Руководство по эксплуатации компьютера обеспечивает информацию, необходимую для эксплуатации компьютера, на котором будет выполняться разработанное ПО, и его периферийного оборудования. Данное руководство, главным образом, касается эксплуатации самого компьютера, а не конкретного ПО, которое на нем будет выполняться. Содержание этого руководства:

- эксплуатационные процедуры для компьютерной системы (включение и отключение, сбой питания, инициализация, ввод/вывод, мониторинг и др.);
- процедуры обработки ошибок;
- диагностические процедуры, включая идентификацию аппаратных, программных и программно-аппаратных средств, необходимых для выполнения этих процедур; пошаговые инструкции для выполнения процедур; диагностические сообщения и требуемые действия;
- диагностические инструментальные средства.

12.34 Руководство по программированию для компьютера

Руководство по программированию для компьютера содержит информацию, необходимую пользователю для создания программ для данного компьютера. Указанное руководство посвящено собственно описанию компьютера, а не функционального ПО, которое будет выполняться на компьютере. Руководство должно включать в себя:

- описание среды программирования — конфигурацию и перечень компонентов компьютерной системы, рабочие характеристики, возможности и ограничения, включая машинный цикл, длину слова, объем памяти и ее характеристики, перечень команд, прерываний, режимы работы (пакетный, интерактивный, привилегированный, непривилегированный), рабочие регистры, характеристики ввода/вывода, специальные возможности, описание носителей данных (ленты, диски и другие периферийные устройства);
- информацию о возможностях программирования — представление данных (байт, слово, целые, с плавающей точкой, двойная точность); формат команд и методы адресации; специальные регистры; команды передачи управления (условный и безусловный переходы и др.), процедуры и подпрограммы; обработка прерываний; синхронизация и таймеры, возможности защиты памяти; детальное описание каждой команды (их использование, синтаксис, время выполнения и др.);

программирование управления вводом/выводом; примеры, демонстрирующие возможности программирования.

12.35 Руководство поддержки программно-аппаратных средств

Руководство поддержки программно-аппаратных средств содержит информацию, необходимую для программирования и перепрограммирования системных устройств программно-аппаратных средств. Это касается запоминающих и других устройств. Данное руководство содержит описание:

- самих устройств — аппаратуры и ПО;
- процедур для стирания устройств;
- процедур для загрузки ПО в устройства;
- процедур для контроля процесса загрузки;
- процедур маркировки загруженных устройств.

12.36 Руководство оператора ПО

Руководство оператора ПО содержит описание запуска системы управления либо непосредственно с центрального компьютера, либо другим централизованным способом, либо через сеть.

Данное руководство содержит описание аппаратных и программных средств, требуемых для работы системы:

- технические характеристики используемых устройств;
- структура ПО, обзор назначения/функционирования каждого компонента ПО;
- перечень входных команд, команд доступа к ПО и реакция на их выполнение;
- аварийные сообщения и другие выходные данные, формируемые ПО;
- типовые времена выполнения;
- последовательность действий для запуска;
- перечень требуемых библиотек поддержки, интерфейсов;
- форма и средства регистрации ошибок, возникающих в процессе эксплуатации ПО;
- перечень процедур, выполняемых оператором при установке ПО для конкретного выбранного окружения, конкретной конфигурации ПО.

12.37 Руководство по входной/выходной информации ПО

Руководство по входной/выходной информации ПО объясняет пользователю как представить, ввести входную информацию и как интерпретировать выходную информацию, в каком режиме (пакетном или интерактивном) работает система ПО, запускаемая непосредственно с центрального компьютера или другим централизованным способом, или через сеть.

Данное руководство содержит краткое описание прикладного ПО, перечень файлов, включая базу данных и файлы со справочной информацией для пользователя, описание аппаратуры, ПО и прочих ресурсов, необходимых для доступа к данному прикладному ПО и использования этого ПО в полном объеме, включая:

- режимы работы;
- описание процедур, позволяющих получить помощь при возникновении ошибочных ситуаций при работе с прикладным ПО;
- терминалы, принтеры и другие входные/выходные устройства;
- необходимые процедуры, утилиты, в том числе процедуры для установки ПО;
- форматы представления входной/выходной информации, их тип, объем;
- точность представления, скорость передачи, ожидаемое время реакции;
- способ задания конца информации и другие требуемые соглашения;
- ограничения и наиболее типичные ошибки задания информации;
- описание используемой системы управления базой данных.

12.38 Руководство пользователя ПО

Руководство пользователя ПО описывает порядок действий пользователя ПО для установки и использования ЭКПО, группы ЭКПО или системы/подсистемы ПО.

Руководство разрабатывают для ПО, которое прогоняется самим пользователем, и для используемого им интерфейса, требуемого для введения входных данных и интерпретации получаемых выходных результатов. При наличии в системе встроенного ПО не требуется отдельного руководства для пользователя, перечень требуемых процедур для работы с таким ПО можно включить в данный документ. Содержание руководства:

- краткое описание характеристик ПО;
- перечень файлов, включая файлы базы данных, необходимых для работы ПО;
- порядок действий для продолжения или возобновления работы в случаях возникновения непредвиденных ситуаций;

- описание программной среды;
- организация ПО и функционирование ПО с точки зрения пользователя;
- описание процедур, позволяющих фиксировать ошибки;
- детальные, пошаговые действия пользователя при включении системы и дальнейшей работе с ней;
- ссылки на другие переданные руководства;
- перечень и пояснение выводимых системой сообщений.

12.39 Описание версии ПО

Документ «Описание версии ПО» является полным описанием версии ПО, которая предназначена для передачи пользователю. Данный документ должен содержать следующую информацию:

- полную идентификацию системы и ПО, к которым применяют данный документ, включая регистрационные номера управления конфигурацией и номера версий;
- краткий обзор назначения системы и ПО, историю разработки, эксплуатации и сопровождения системы, идентификацию заказчика, пользователя, разработчика, организации, осуществляющей поддержку, текущих и планируемых мест установки системы;
- полную идентификацию физических носителей, содержащих ПО и связанные с ними документы;
- полную идентификацию всех компьютерных файлов, содержащих ПО;
- перечень всех изменений, внесенных после выпуска предыдущей версии в ПО;
- перечень документов ПО, связанных с данной версией;
- инструкцию по установке ПО;
- перечень возможных проблем и известных ошибок.

13 Дополнительные вопросы

В этом разделе представлены дополнительные аспекты сертификации ПО в случаях использования ранее разработанного ПО и аттестации инструментальных средств ПО.

13.1 Использование ранее разработанного ПО

В последующих подразделах рассмотрены вопросы, связанные с использованием ранее разработанного ПО, включая оценку модификаций, изменение объекта среды приложения или среды разработки, обновление базовой линии разработки и вопросы управления конфигурацией и обеспечения качества. Намерение использовать ранее разработанное ПО должно быть отражено в Плане сертификации в части ПО.

13.1.1 Модификация ранее разработанного ПО

Модификация ранее разработанного в соответствии с требованиями настоящего стандарта ПО может быть следствием изменения требований, обнаружения ошибок и/или расширения функциональных возможностей ПО. Анализ предполагаемой модификации заключается в следующем:

- пересмотр результатов оценки безопасности системы с учетом предполагаемой модификации;
- выполнение требования 13.1.4, если уровень ПО пересмотрен;
- анализ воздействия изменений требований к ПО и воздействия изменений архитектуры ПО;
- анализ последствий воздействия изменений требований к ПО на другие требования;
- учет связей между несколькими компонентами ПО, что может привести к повторной верификации дополнительного объема информации, включающей в себя также неизмененные участки кода;
- определение области, на которую воздействует изменение, что может быть выполнено с помощью анализа потока данных, анализа потока управления, временного анализа и анализа трассируемости;
- повторная верификация областей, на которые воздействует изменение, должна быть выполнена с учетом требований раздела 8.

13.1.2 Изменение системы или объекта управления

Система или объект, содержащий ПО, которое было ранее сертифицировано в соответствии с определенными уровнем ПО и сертификационным базисом, могут быть использованы на другом объекте. При использовании ранее разработанного ПО на другом объекте необходимо:

- оценить безопасность системы или нового объекта и определить уровень ПО и сертификационный базис. Никаких дополнительных действий не будет требоваться, если они те же самые для новой установки, что и для предыдущей;
- если требуются функциональные модификации для новой установки, то необходимо учесть требования 13.1.1;

- если действия предыдущей разработки не обеспечивали выходных результатов, требуемых для подтверждения безопасности новой установки, то необходимо учесть требования 13.1.4.

13.1.3 Изменения среды применения или среды разработки

Использование и модификация ранее разработанного ПО могут включать в себя новую среду разработки, новый объектный процессор или другие аппаратные средства, или интеграцию с ПО, которое отлично от используемого для первоначального применения.

Новая среда разработки может увеличивать или уменьшать объем некоторых работ в процессах жизненного цикла ПО. Новая среда применения может требовать проведения дополнительных работ процесса жизненного цикла ПО, которые предназначены для модификации. Требования для изменения среды применения или среды разработки следующие:

- если новая среда разработки использует инструментальные средства разработки ПО, то может быть необходимо применить требования 13.2;

- строгая оценка изменения среды применения должна включать в себя рассмотрение сложности и возможностей языка программирования. Например, строгость оценки для родовых функций Ада должна быть большей, если родовые параметры отличны в новом приложении. Для объектно-ориентированного языка строгость должна быть большей, если объекты отличны в новом приложении;

- когда используют другой компилятор или другой набор опций компилятора, что приводит к различиям в объектном коде, результаты предыдущих работ процесса верификации ПО, использующие объектный код, не могут быть рассмотрены как правильные и не должны быть использованы для нового применения. В этом случае предыдущие тестовые результаты больше не могут быть допустимы для критериев структурного покрытия в новом приложении. Точно так же соглашения относительно оптимизации компилятора не могут быть допустимы;

- когда применяют другой процессор, результаты предыдущих работ процесса верификации интерфейса аппаратных средств/ПО не могут быть использованы для нового применения; должны быть выполнены все ранее выполняемые тесты интеграции аппаратных средств/ПО; должна быть повторена проверка совместимости аппаратных средств/ПО; могут потребоваться дополнительные тесты интеграции аппаратных средств/ПО и просмотры;

- должна быть выполнена верификация интерфейсов ПО в тех случаях, когда ранее разработанное ПО использовали с другим программным интерфейсом.

13.1.4 Обновление базовой линии разработки

В данном пункте рассмотрены требования для ПО, документы жизненного цикла которого из предыдущего применения признаны неадекватными или не удовлетворяющими требованиям настоящего стандарта из-за целей безопасности, связанных с новым применением. Эти требования предназначены для использования при сертификации:

- коммерчески доступного ПО;

- прикладного ПО, разработанного в соответствии со стандартами, отличными от настоящего стандарта;

- прикладного ПО, разработанного до принятия настоящего стандарта;

- ПО, ранее разработанного в соответствии с настоящим стандартом, но как ПО более низкого уровня.

При обновлении базовой линии разработки необходимо руководствоваться следующим:

- для удовлетворения требованиям настоящего стандарта могут быть использованы те документы жизненного цикла ПО предыдущей разработки, которые соответствуют требованиям нового применения;

- сертификация в части ПО должна регламентироваться отказными ситуациями и уровнями ПО, определенными процессом оценки безопасности системы. Сравнение с отказными ситуациями предыдущего применения позволит определить области, которые могут потребовать изменения;

- документы жизненного цикла ПО из предыдущей разработки должны быть оценены заново, чтобы гарантировать, что цели процесса верификации ПО требуемого уровня удовлетворены для нового применения;

- для восстановления документации жизненного цикла ПО, которая отсутствует или является неадекватной при удовлетворении целям данного документа, может быть применен метод обратной разработки. Кроме непосредственных работ по созданию программного средства, могут потребоваться дополнительные работы для удовлетворения целям процесса верификации ПО;

- необходимо определить стратегию, обеспечивающую соответствие Плана сертификации в части ПО с настоящим стандартом.

13.1.5 Управление конфигурацией ПО

Если используют ранее разработанное ПО, то процесс управления конфигурацией ПО для нового применения должен включать в себя дополнительно к рекомендациям раздела 9 следующее:

- трассируемость от программного средства и его документов для предыдущего применения к программному средству и документам для нового применения;
- контроль изменений, который позволяет регистрировать дефекты, выяснять причины их появления и прослеживать изменения к программным компонентам, используемым в более чем одном приложении.

13.1.6 Обеспечения качества ПО

Если используют ранее разработанное ПО, то процесс обеспечения качества для нового применения, в дополнение к рекомендациям раздела 10, должен включать в себя обеспечение того, что:

- компоненты ПО удовлетворяют или превышают критерии соответствующего уровня ПО для нового применения;
- изменения в процессах жизненного цикла ПО отражены в планах ПО.

13.2 Аттестация инструментальных средств

Аттестация инструментальных средств необходима, когда процессы, представленные в настоящем стандарте, могут быть исключены, сокращены или автоматизированы посредством использования инструментальных средств, без верификации их выходных данных, как это установлено в разделе 8. Использование инструментальных средств для автоматизации работ в процессах жизненного цикла ПО помогает обеспечить надежность системы, поскольку эти средства способствуют удовлетворению требованиям стандартов разработки ПО и осуществляют автоматический контроль.

Цель процесса аттестации — гарантировать, что инструментальное средство обеспечивает доверие, по крайней мере, эквивалентное доверию к тем процессам, которые будут исключены, сокращены или автоматизированы. Если возможна демонстрация разбиения инструментальных средств по функциям, то должны быть аттестованы только те средства, которые будут использованы в целях устранения, сокращения или автоматизации работ в процессах жизненного цикла, или те средства, выходные результаты которых не были аттестованы.

Могут быть аттестованы только детерминированные инструментальные средства. Это такие средства, которые выдают те же самые результаты для тех же самых входных данных при работе в той же самой среде. Процесс аттестации инструментальных средств может быть применен либо к одному средству, либо к группе средств.

Инструментальные средства могут быть классифицированы одним из двух типов:

- Инструментальные средства разработки ПО: инструментальные средства, выходные данные которых являются частью прикладного ПО и которые, таким образом, могут внести ошибки в разрабатываемое программное средство. Например, инструментальное средство, генерирующее исходный текст непосредственно из требований нижнего уровня, должно быть аттестовано, если генерируемый исходный текст не верифицируется, как установлено в разделе 8.
- Инструментальные средства верификации ПО: инструментальные средства, которые не могут внести ошибки, но могут пропустить последние при их выявлении. Например, статический анализатор, который автоматизирует работы процесса верификации ПО, должен быть аттестован, если функции, которые он выполняет, не верифицируются другим способом; средства контроля типов данных, средства анализа и средства тестирования являются другими примерами подобных инструментальных средств.

Требования к аттестации инструментальных средств:

- инструментальные средства должны быть аттестованы в соответствии с их типом, определенным выше;
- комбинированные инструментальные средства разработки и верификации должны быть аттестованы в соответствии с требованиями 13.2.1, даже если может быть продемонстрировано разделение этих двух функций;
- все виды работ процессов управления конфигурацией и обеспечения качества обязательно выполняются для прикладного ПО, а также и для аттестуемых инструментальных средств.

Цели процесса верификации для инструментальных средств разработки ПО представлены в 13.2.1, перечисление г).

Инструментальные средства могут быть аттестованы только для использования конкретными системами, для которых использование данных средств включено в План сертификации в части ПО. Использование этих средств другими системами может потребовать дополнительной аттестации.

13.2.1 Критерии аттестации для инструментальных средств разработки ПО

Критерии аттестации для инструментальных средств разработки ПО следующие:

а) если инструментальное средство разработки ПО должно быть аттестовано, то процессы разработки ПО для этого средства должны удовлетворять тем же самым целям, что и процессы разработки прикладного ПО, при разработке которого предполагается использовать это инструментальное средство;

б) уровень ПО, назначенный инструментальным средствам, должен быть таким же, как и для прикладного ПО, которое с их помощью разрабатывают, за исключением тех случаев, когда соискатель может обосновать для сертифицирующей организации возможность снижения уровня ПО для этих средств.

Примечание — Возможность снижения уровня ПО для инструментальных средств разработки зависит от значимости работ процессов верификации ПО, которые будут исключены, сокращены или автоматизированы в результате применения данного средства, в сравнении с полным набором всех работ по верификации. Значимость является функцией:

- вида работ процесса верификации ПО, которые могут быть устранены, сокращены или автоматизированы. Например, работы верификации по проверке согласования исходного текста со стандартами по структурированному представлению текста являются менее важными, чем работы верификации, направленные на определение соответствия исполняемого объектного кода требованиям верхнего уровня;
- вероятности того, что другие верификационные работы выявят те же самые ошибки;

в) соискатель должен продемонстрировать, что средства соответствуют эксплуатационным требованиям к инструментальным средствам (13.2.3); эта демонстрация может включать в себя период испытаний, во время которого выполняют верификацию выходных результатов инструментального средства и анализируют, регистрируют и корректируют ошибки, связанные с работой средства;

г) инструментальные средства разработки ПО должны быть верифицированы, чтобы проконтролировать корректность, согласованность и полноту эксплуатационных требований к этим средствам и продемонстрировать соответствие этим требованиям. Цели процесса верификации инструментальных средств отличаются от аналогичных целей для прикладного ПО, так как требованиями верхнего уровня для инструментальных средств являются их эксплуатационные требования вместо системных требований, как в случае прикладного ПО. Цели верификации инструментальных средств разработки ПО могут быть достигнуты с помощью:

1) просмотра документа «Эксплуатационные требования к инструментальному средству», как описано в 8.3.1, перечисления а), б);
2) демонстрации того, что инструментальное средство удовлетворяет Эксплуатационным требованиям к инструментальному средству для нормального рабочего режима;
3) демонстрации того, что средство соответствует Эксплуатационным требованиям к инструментальному средству для внештатных рабочих условий, включая внешние помехи и выборочные отказы, возникающие в инструментальном средстве и среде его функционирования;
4) анализа покрытия, основанного на требованиях, и дополнительного тестирования для завершения покрытия требований;
5) анализа структурного покрытия, соответствующего уровню ПО для инструментального средства;
6) тестирования на устойчивость к ошибкам в соответствии с уровнем ПО инструментального средства, как описано в 8.4.2;
7) анализа потенциальных ошибок, возникающих из-за использования средства, чтобы подтвердить достоверность Плана аттестации инструментального средства.

13.2.2 Критерии аттестации для инструментальных средств верификации ПО

Критерии аттестации для инструментальных средств верификации ПО удовлетворяются демонстрацией того, что указанные средства соответствуют Эксплуатационным требованиям к инструментальному средству для нормальных условий эксплуатации.

13.2.3 Документы по аттестации инструментальных средств

Требования к документам по аттестации инструментальных средств:

а) в случае аттестации инструментальных средств План сертификации в части ПО для прикладного ПО должен определить те инструментальные средства, которые должны быть аттестованы, и иметь ссылки на документы по аттестации этих средств;

б) документы по аттестации инструментальных средств должны иметь 1-ю категорию контроля для средств разработки ПО и 2-ю категорию контроля для средств верификации ПО;

в) документы по аттестации инструментальных средств разработки ПО должны соответствовать документам раздела 11, иметь те же характеристики и содержать ту же информацию, что и для прикладного ПО, с учетом того, что:

1) План аттестации инструментального средства решает те же задачи, что и План сертификации в части ПО для прикладного ПО;
2) Эксплуатационные требования к инструментальному средству соответствуют Спецификации требований к ПО для прикладного ПО;
3) Итоговый документ разработки инструментального средства для инструментальных средств содержит ту же информацию, что и Итоговый документ разработки ПО для прикладного ПО.

Для инструментальных средств разработки ПО, которые должны быть аттестованы, План аттестации инструментального средства описывает процесс аттестации средства. Указанный план должен включать в себя:

- идентификацию конфигурации для инструментального средства;
- описание полученного в результате применения инструментального средства сертификационного доверия, т.е. тех работ процесса верификации, которые будут исключены, сокращены или автоматизированы;
- указание уровня ПО, присваиваемого для инструментального средства;
- описание архитектуры инструментального средства;
- работы, которые должны быть выполнены для аттестации инструментального средства;
- документы по аттестации инструментального средства.

Документ «Эксплуатационные требования к инструментальному средству» описывает инструментальное средство на функциональном уровне. Этот документ должен включать в себя:

- описание функций инструментального средства и его технических возможностей. Для средств разработки ПО документ должен включать в себя описание работ процессов разработки, выполняемых с помощью данного средства;
- информацию для пользователя, такую как руководство для установки и руководство пользователя;
- описание операционной среды, необходимой для работы инструментального средства;
- ожидаемую ответную реакцию средств разработки ПО в случаях внештатных условий работы.

13.2.4 Согласие сертифицирующей организации на использование инструментального средства

Выдача согласия сертифицирующей организации на использование инструментального средства включает в себя два этапа:

- для средств разработки ПО — согласие с Планом аттестации инструментального средства; для средств верификации ПО — согласие с Планом сертификации в части ПО для прикладного ПО;
- для средств разработки ПО — согласие с Итоговым документом разработки инструментального средства; для средств верификации ПО — согласие с Итоговым документом разработки ПО для прикладного ПО.

ПРИЛОЖЕНИЕ А
(рекомендуемое)

Цели и результаты процессов в зависимости от уровня ПО

В настоящем приложении приведено описание требований сертификации для целей и результатов процессов жизненного цикла ПО в зависимости от уровня ПО. В таблицах А.1 — А.10 даны ссылки на ранее описанные в настоящем стандарте цели и результаты.

Таблица А.1 — Процесс планирования ПО

Цель		Применимость к уровням ПО				Результат		Категория контроля по уровням ПО			
Описание	Ссылка	A	B	C	D	Описание	Ссылка	A	B	C	D
Определить виды работ процессов разработки ПО и интегральных процессов	6.1а), 6.3	0	0	0	0	План сертификации в части ПО	12.1	1	1	1	1
						План разработки ПО	12.2	1	1	2	2
						План верификации ПО	12.3	1	1	2	2
Определить критерии перехода, взаимосвязи и последовательность выполнения процессов	6.1б), 6.3	0	0	0		План квалификационного тестирования ПО	12.4	1	1	2	2
						План управления конфигурацией ПО	12.5	1	1	2	2
Определить среду жизненного цикла ПО	6.1 в)	0	0	0		План обеспечения качества ПО	12.6	1	1	2	2
						План установки ПО	12.7	1	1	2	2
Рассмотреть дополнительные вопросы	6.1 г)	0	0	0	0	План передачи ПО	12.8	1	1	2	2
Определить стандарты на разработку ПО	6.1 д)	0	0	0		Стандарты на разработку требований к ПО	12.9	1	1	2	
						Стандарты на процесс проектирования ПО	12.10	1	1	2	
						Стандарты кодирования ПО	12.11	1	1	2	
Согласование планов ПО с настоящим стандартом	6.1 е), 6.7	0	0	0		Протоколы обеспечения качества ПО	12.30	2	2	2	
						Результаты верификации ПО	12.23	2	2	2	
Координация планов создания ПО	6.1 ж), 6.7	0	0	0		Протоколы обеспечения качества ПО	12.30	2	2	2	
						Результаты верификации ПО	12.23	2	2	2	

Обозначения:
0 — цель должна быть удовлетворена;
пробел — удовлетворение цели на усмотрение заказчика;
1 — документ должен удовлетворять целям категории контроля 1 (КК1);
2 — документ должен удовлетворять целям категории контроля 2 (КК2).

Таблица А.2 — Процессы разработки ПО

Цель		Применимость к уровням ПО				Результат		Категория контроля по уровням ПО			
Описание	Ссылка	A	B	C	D	Описание	Ссылка	A	B	C	D
Разработать требования верхнего уровня	7.1.1 а)	0	0	0	0	Спецификация системы/подсистемы	12.12	1	1	1	1
						Спецификация требований к ПО	12.13	1	1	1	1
						Спецификация требований к интерфейсу	12.14	1	1	1	1
Определить производные требования верхнего уровня	7.1.1 б)	0	0	0	0	Спецификация требований к ПО	12.13	1	1	1	1
						Спецификация требований к интерфейсу	12.14	1	1	1	1
Разработать архитектуру ПО	7.2.1 а)	0	0	0	0	Описание проекта системы/подсистемы	12.15	1	1	2	2
						Описание проекта ПО	12.16	1	1	2	2
						Описание проекта интерфейса	12.17	1	1	2	2
						Описание проекта базы данных	12.18	1	1	2	2
Разработать требования нижнего уровня	7.2.1 а)	0	0	0	0	Описание проекта ПО	12.16	1	1	2	2
Определить производные требования нижнего уровня	7.2.1 б)	0	0	0	0	Описание проекта ПО	12.16	1	1	2	2
Разработать исходный код	7.3.1	0	0	0	0	Исходный код ПО	12.19	1	1	1	1
Получить исполняемый объектный код и выполнить интеграцию ПО/аппаратуры	7.4.1	0	0	0	0	Исполняемый объектный код ПО	12.20	1	1	1	1
Подготовить руководства пользователя и руководства поддержки	5.9.3, 5.10.6					Спецификация программного средства	12.27	2	2	2	2
						Описание эксплуатационной концепции	12.32	2	2	2	2
						Руководство по эксплуатации компьютера	12.33	2	2	2	2
						Руководство по программированию для компьютера	12.34	2	2	2	2
						Руководство поддержки программно-аппаратных средств	12.35	2	2	2	2
						Руководство оператора ПО	12.36	2	2	2	2
						Руководство по входной / выходной информации ПО	12.37	2	2	2	2
						Руководство пользователя ПО	12.38	2	2	2	2
						Описание версии ПО	12.39	2	2	2	2

Обозначения:

0 — цель должна быть удовлетворена;

пробел — удовлетворение цели на усмотрение заказчика;

1 — документ должен удовлетворять целям категории контроля 1 (КК1);

2 — документ должен удовлетворять целям категории контроля 2 (КК2).

Таблица А.3 — Верификация результатов процесса разработки требований к ПО

Цель		Применимость к уровням ПО				Результат		Категория контроля по уровням ПО			
Описание	Ссылка	A	B	C	D	Описание	Ссылка	A	B	C	D
Требования верхнего уровня к ПО согласуются с требованиями к системе	8.3.1 а)	*	*	0	0	Результаты верификации ПО	12.23	2	2	2	2
Требования верхнего уровня точны и непротиворечивы	8.3.1 б)	*	*	0	0	Результаты верификации ПО	12.23	2	2	2	2
Требования верхнего уровня совместимы с объектным компьютером	8.3.1 в)	0	0			Результаты верификации ПО	12.23	2	2		
Требования верхнего уровня верифицируемы	8.3.1 г)	0	0	0		Результаты верификации ПО	12.23	2	2	2	
Требования верхнего уровня соответствуют стандартам на разработку требований к ПО	8.3.1 д)	0	0	0	0	Результаты верификации ПО	12.23	2	2	2	2
Требования верхнего уровня трассируемы к системным требованиям	8.3.1 е)	0	0	0	0	Результаты верификации ПО	12.23	2	2	2	2
Алгоритмы точны и корректны	8.3.1 ж)	*	*	0		Результаты верификации ПО	12.23	2	2	2	

Обозначения:
* — цель должна быть удовлетворена с обеспечением независимости;
0 — цель должна быть удовлетворена;
пробел — удовлетворение цели на усмотрение заказчика;
2 — документ должен удовлетворять целям категории контроля 2 (КК2).

Таблица А.4 — Верификация результатов процесса проектирования ПО

Цель		Применимость к уровням ПО				Результат		Категория контроля по уровням ПО			
Описание	Ссылка	A	B	C	D	Описание	Ссылка	A	B	C	D
Требования нижнего уровня к ПО согласуются с требованиями верхнего уровня	8.3.3 а)	*	*	0		Результаты верификации ПО	12.23	2	2	2	
Требования нижнего уровня точны и непротиворечивы	8.3.3 б)	*	*	0		Результаты верификации ПО	12.23	2	2	2	
Требования нижнего уровня совместимы с объектным компьютером	8.3.3 в)	0	0			Результаты верификации ПО	12.23	2	2		
Требования нижнего уровня верифицируемы	8.3.3 г)	0	0			Результаты верификации ПО	12.23	2	2		
Требования нижнего уровня соответствуют стандартам	8.3.3 д)	0	0	0		Результаты верификации ПО	12.23	2	2	2	
Требования нижнего уровня трассируемы к требованиям верхнего уровня	8.3.3 е)	0	0	0		Результаты верификации ПО	12.23	2	2	2	
Алгоритмы точны и корректны	8.3.3 ж)	*	*	0		Результаты верификации ПО	12.23	2	2	2	
Архитектура ПО согласуется с требованиями верхнего уровня	8.3.2 а)	*	0	0		Результаты верификации ПО	12.23	2	2	2	
Архитектура ПО непротиворечива	8.3.2 б)	*	0	0		Результаты верификации ПО	12.23	2	2	2	
Архитектура ПО совместима с объектным компьютером	8.3.2 в)	0	0			Результаты верификации ПО	12.23	2	2		

Окончание таблицы А.4

Цель		Применимость к уровням ПО				Результат		Категория контроля по уровням ПО			
Описание	Ссылка	A	B	C	D	Описание	Ссылка	A	B	C	D
Архитектура ПО верифицируема	8.3.2 г)	0	0			Результаты верификации ПО	12.23	2	2		
Архитектура ПО соответствует стандартам на процесс проектирования ПО	8.3.2 д)	0	0	0		Результаты верификации ПО	12.23	2	2	2	
Подтверждается целостность разбиения ПО	8.3.2 е)	*	0	0	0	Результаты верификации ПО	12.23	2	2	2	2

Обозначения:
* — цель должна быть удовлетворена с обеспечением независимости;
0 — цель должна быть удовлетворена;
пробел — удовлетворение цели на усмотрение заказчика;
2 — документ должен удовлетворять целям категории контроля 2 (КК2).

Таблица А.5 — Верификация результатов процесса кодирования и интеграции ПО

Цель		Применимость к уровням ПО				Результат		Категория контроля по уровням ПО			
Описание	Ссылка	A	B	C	D	Описание	Ссылка	A	B	C	D
Исходный код согласуется с требованиями нижнего уровня	8.3.4 а)	*	*	0		Результаты верификации ПО	12.23	2	2	2	
Исходный код согласуется с архитектурой ПО	8.3.4 б)	*	0	0		Результаты верификации ПО	12.23	2	2	2	
Исходный код верифицируем	8.3.4 в)	0	0			Результаты верификации ПО	12.23	2	2		
Исходный код соответствует стандартам	8.3.4 г)	0	0	0		Результаты верификации ПО	12.23	2	2	2	
Исходный код трассируем к требованиям нижнего уровня	8.3.4 д)	0	0	0		Результаты верификации ПО	12.23	2	2	2	
Исходный код точен и непротиворечив	8.3.4 е)	*	0	0		Результаты верификации ПО	12.23	2	2	2	
Результаты процесса интеграции ПО полны и корректны	8.3.5	0	0	0		Результаты верификации ПО	12.23	2	2	2	

Обозначения:
* — цель должна быть удовлетворена с обеспечением независимости;
0 — цель должна быть удовлетворена;
пробел — удовлетворение цели на усмотрение заказчика;
2 — документ должен удовлетворять целям категории контроля 2 (КК2).

Таблица А.6 — Тестирование результатов процесса интеграции ПО

Цель		Применимость к уровням ПО				Результат		Категория контроля по уровням ПО			
Описание	Ссылка	A	B	C	D	Описание	Ссылка	A	B	C	D
Исполняемый объектный код согласуется с требованиями верхнего уровня	8.4.2, 8.4.3 а),б), 8.5.4	0	0	0	0	Процедуры верификации ПО	12.21	1	1	2	2
						Описание квалификационного тестирования ПО	12.22	1	1	2	2
						Результаты верификации ПО	12.23	2	2	2	2
						Отчет о квалификационном тестировании ПО	12.24	2	2	2	2
Исполняемый объектный код устойчив относительно входов, определенных требованиями верхнего уровня	8.4.2, 8.4.3 а), б), 8.5.4	0	0	0	0	Процедуры верификации ПО	12.21	1	1	2	2
						Описание квалификационного тестирования ПО	12.22	1	1	2	2
						Результаты верификации ПО	12.23	2	2	2	2
						Отчет о квалификационном тестировании ПО	12.24	2	2	2	2
Исполняемый объектный код согласуется с требованиями нижнего уровня	8.4.2, 8.4.3 в)	*	*	0		Процедуры верификации ПО	12.21	1	1	2	
						Результаты верификации ПО	12.23	2	2	2	
Исполняемый объектный код устойчив относительно входов, определенных требованиями нижнего уровня	8.4.2, 8.4.3 в)	*	0	0	0	Процедуры верификации ПО	12.21	1	1	2	2
						Результаты верификации ПО	12.23	2	2	2	2
Исполняемый код совместим с объектным компьютером	8.4.3 а)	0	0	0	0	Процедуры верификации ПО	12.21	1	1	2	2
						Описание квалификационного тестирования ПО	12.22	1	1	2	2
						Результаты верификации ПО	12.23	2	2	2	2
						Отчет о квалификационном тестировании ПО	12.24	2	2	2	2

Обозначения:
* — цель должна быть удовлетворена с обеспечением независимости;
0 — цель должна быть удовлетворена;
пробел — удовлетворение цели на усмотрение заказчика;
1 — документ должен удовлетворять целям категории контроля 1 (КК1);
2 — документ должен удовлетворять целям категории контроля 2 (КК2).

Таблица А.7 — Верификация результатов процесса верификации ПО

Цель		Применимость к уровням ПО				Результат		Категория контроля по уровням ПО			
Описание	Ссылка	A	B	C	D	Описание	Ссылка	A	B	C	D
Тестовые процедуры корректны	8.3.6 б), 8.5.4	*	0	0		Процедуры верификации ПО	12.21	2	2	2	
						Описание квалификационного тестирования ПО	12.22	2	2	2	
Результаты тестов корректны и все расхождения объяснены	8.3.6 в), 8.5.4	*	0	0		Результаты верификации ПО	12.23	2	2	2	
						Отчет о квалификационном тестировании ПО	12.24	2	2	2	
Тестовое покрытие требований верхнего уровня достигнуто	8.4.4.1, 8.5.4	*	0	0	0	Результаты верификации ПО	12.23	2	2	2	2
						Отчет о квалификационном тестировании ПО	12.24	2	2	2	2

Окончание таблицы А.7

Цель		Применимость к уровням ПО				Результат		Категория контроля по уровням ПО			
Описание	Ссылка	A	B	C	D	Описание	Ссылка	A	B	C	D
Тестовое покрытие требований нижнего уровня достигнуто	8.4.4.1	*	0	0		Результаты верификации ПО	12.23	2	2	2	
Тестовое покрытие структуры ПО (модифицированное покрытие условий/решений) достигнуто	8.4.4.2	*				Результаты верификации ПО	12.23	2			
Тестовое покрытие структуры ПО (покрытие решений) достигнуто	8.4.4.2 а), 8.4.4.2 б)	*	*			Результаты верификации ПО	12.23	2	2		
Тестовое покрытие структуры ПО (покрытие операторов) достигнуто	8.4.4.2 а), 8.4.4.2 б)	*	*	0		Результаты верификации ПО	12.23	2	2	2	
Тестовое покрытие структуры ПО (связи по управлению и связи по данным) достигнуто	8.4.4.2 в)	*	*	0		Результаты верификации ПО	12.23	2	2	2	

Обозначения:
* — цель должна быть удовлетворена с обеспечением независимости;
0 — цель должна быть удовлетворена;
пробел — удовлетворение цели на усмотрение заказчика;
2 — документ должен удовлетворять целям категории контроля 2 (КК2).

Таблица А.8 — Процесс управления конфигурацией ПО

Цель		Применимость к уровням ПО				Результат		Категория контроля по уровням ПО			
Описание	Ссылка	A	B	C	D	Описание	Ссылка	A	B	C	D
Элементы конфигурации идентифицированы	9.2.1	0	0	0	0	Протоколы управления конфигурацией ПО	12.29	2	2	2	2
Установлены базовая линия и трассируемость	9.2.3	0	0	0	0	Указатель конфигурации ПО	12.26	1	1	1	1
						Протоколы управления конфигурацией ПО	12.29	2	2	2	2
Установлены отчетность о дефектах, просмотры изменений, регистрация состояния конфигурацией	9.2.4, 9.2.5 9.2.6, 9.2.7	0	0	0	0	Сообщения о дефектах	12.28	2	2	2	2
						Протоколы управления конфигурацией ПО	12.29	2	2	2	2
Установлены архивирование, получение из архива и выпуск версии	9.2.8	0	0	0	0	Протоколы управления конфигурацией ПО	12.29	2	2	2	2
Установлено управление загрузкой ПО	9.2.9	0	0	0	0	Протоколы управления конфигурацией ПО	12.29	2	2	2	2
Установлен контроль среды жизненного цикла ПО	9.2.10	0	0	0	0	Указатель конфигурации среды жизненного цикла ПО	12.25	1	1	1	2
						Протоколы управления конфигурацией ПО	12.29	2	2	2	2

Обозначения:
0 — цель должна быть удовлетворена;
1 — документ должен удовлетворять целям категории контроля 1 (КК1);
2 — документ должен удовлетворять целям категории контроля 2 (КК2).

Таблица А.9 — Процесс обеспечения качества ПО

Цель		Применимость к уровням ПО				Результат		Категория контроля по уровням ПО			
Описание	Ссылка	A	B	C	D	Описание	Ссылка	A	B	C	D
Обеспечена уверенность в том, что процессы разработки ПО и интегральные процессы соответствуют утвержденным планам и стандартам ПО	10.1 а)	*	*	*	*	Протоколы обеспечения качества ПО	12.30	2	2	2	2
Обеспечена уверенность, что удовлетворены критерии переходов между процессами жизненного цикла ПО	10.1 б)	*	*			Протоколы обеспечения качества ПО	12.30	2	2		
Выполнен просмотр соответствия ПО	10.1 в), 10.3	*	*	*	*	Протоколы обеспечения качества ПО	12.30	2	2	2	2

Обозначения:
* — цель должна быть удовлетворена с обеспечением независимости;
пробел — удовлетворение цели на усмотрение заказчика;
2 — документ должен удовлетворять целям категории контроля 2 (КК2).

Таблица А.10 — Процесс сертификационного сопровождения ПО

Цель		Применимость к уровням ПО				Результат		Категория контроля по уровням ПО			
Описание	Ссылка	A	B	C	D	Описание	Ссылка	A	B	C	D
Установлено взаимодействие и взаимопонимание между соискателем и сертифицирующей организацией	11	0	0	0	0	План сертификации в части ПО	12.1	1	1	1	1
Предложены средства достижения согласия и достигнута согласованность с Планом сертификации в части ПО	11.1	0	0	0	0	План сертификации в части ПО	12.1	1	1	1	1
Представлены доказательства согласованности	11.2	0	0	0	0	Итоговый документ разработки ПО	12.31	1	1	1	1
						Указатель конфигурации ПО	12.26	1	1	1	1

Обозначения:
0 — цель должна быть удовлетворена;
1 — документ должен удовлетворять целям категории контроля 1 (КК1).

УДК 681.3.06:006.354 ОКС 35.080 П85 ОКСТУ 4090

Ключевые слова: встроенные системы, программное обеспечение, жизненный цикл, разработка, документирование, верификация, управление конфигурацией, обеспечение качества

Редактор *Л.В. Афанасенко*
Технический редактор *В.Н. Прусакова*
Корректор *Н.Л. Рыбалко*
Компьютерная верстка *Л.А. Круговой*

Подписано в печать 26.10.2005. Формат 60×84 1/8. Бумага офсетная. Гарнитура Таймс. Печать офсетная.
Усл. печ. л. 7,44. Уч.-изд. л. 7,90. Тираж 50 экз. Зак. 819. С 2057.

ФГУП «Стандартинформ», 123995 Москва, Гранатный пер., 4.
gost.e-ecolog.ru
Набрано во ФГУП «Стандартинформ» на ПЭВМ
Отпечатано в филиале ФГУП «Стандартинформ» — тип. «Московский печатник», 105062 Москва, Лялин пер., 6.

gost.e-ecolog.ru