Межгосударственный Авиационный Комитет
Авиационный Регистр

КВАЛИФИКАЦИОННЫЕ ТРЕБОВАНИЯ

ЧАСТЬ 178В

ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ БОРТОВОЙ АППАРАТУРЫ И СИСТЕМ ПРИ СЕРТИФИКАЦИИ АВИАЦИОННОЙ ТЕХНИКИ

-2002-

© НИИАО 2002.

Documents
Document Sharing Hub
Full document details: КТ-178В: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

ЛИСТ УЧЕТА ИЗМЕНЕИЙ
к Квалификационным требованиям
«Требования к программному обеспечению бортовой аппаратуры и систем при сертификации авиационной техники» КТ-178В

№ п/п	Обозначение изменения	Дата вступления в силу	№ п/п	Обозначение изменения	Дата вступления в силу
1	**«1И1 – 8.2»**				

Примечание:
номер изменения состоит из цифр, указывающих:
- общий порядковый № изменения,
- буквы **«И»**,
- цифр, указывающих порядковый № изменения к данному пункту и
- № пункта, в который вносится изменение.

Например: **«3И2 – 5.2»** - это 3-е изменение к КТ-178В, являющееся вторым к пункту **5.2**.

© НИИАО 2002.

Documents
Document Sharing Hub
Full document details: КТ-178В: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

Предисловие

Основой настоящих Квалификационных требований КТ-178В «Требования к программному обеспечению бортовой аппаратуры и систем при сертификации авиационной техники» является документ **RTCA/DO-178B «Software considerations in airborne systems and equipment certification»** [1] с учетом его корректировок и изменений, приведенных в документе **«First annual report for clarification of ED-12B»** [2].

Документ КТ-178В развивает требования и методические подходы, приведенные в ранее выпущенном документе КТ-178А [3].

Термины и определения, использованные в КТ-178А, по возможности, сохранены.

Отказные состояния определены в соответствии с разделом А-0 Авиационных правил (части 23, 25 и 29) [4-6].

Заключение о соответствии ПО требованиям КТ-178В является обязательным при квалификации комплектующих изделий [7].

© НИИАО 2002.

Documents
Document Sharing Hub
Full document details: КТ-178В: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

ОГЛАВЛЕНИЕ

№ п/п	Наименование раздела	Стр.
1.	ВВЕДЕНИЕ	7
1.1.	Назначение документа	7
1.2.	Область применения документа	7
1.3.	Связь с другими документами	7
1.4.	Рекомендации относительно работы с документом	7
1.5.	Структура документа	9
1.6.	Отличия данного документа от DO-178B	10
2.	СИСТЕМНЫЕ АСПЕКТЫ, ОТНОСЯЩИЕСЯ К РАЗРАБОТКЕ ПО	11
2.1.	Информационные потоки между процессами жизненных циклов системы и программного обеспечения	11
2.1.1.	Информационный поток из процессов системы в процессы ПО	12
2.1.2.	Информационный поток из процессов ПО в процессы системы	12
2.2.	Отказное состояние и уровень ПО	12
2.2.1.	Классификация отказных состояний	13
2.2.2.	Определения уровней ПО	13
2.2.3.	Установление уровня ПО	14
2.3.	Учет особенностей архитектуры системы	15
2.3.1.	Обособление	15
2.3.2.	Многоверсионное разнородное ПО	16
2.3.3.	Обеспечение безопасности с помощью непрерывного контроля	16
2.4.	Учет влияния системных аспектов на ПО, модифицируемое пользователем, опционное ПО и готовое коммерческое ПО	17
2.5.	Учет влияния схемы построения системы на ПО, загружаемое на месте эксплуатации	18
2.6.	Учет требований к системе при верификации ПО	18
2.7.	Учет ПО при верификации системы	19
3.	ЖИЗНЕННЫЙ ЦИКЛ ПО	20
3.1.	Процессы жизненного цикла ПО	20
3.2.	Определение жизненного цикла ПО	20
3.3.	Критерии перехода между процессами	22
4.	ПРОЦЕСС ПЛАНИРОВАНИЯ СОЗДАНИЯ ПО	23
4.1.	Цели процесса планирования	23
4.2.	Мероприятия процесса планировании создания ПО	23
4.3.	Планы создания ПО	24
4.4.	Планирование среды жизненного цикла ПО	24
4.4.1.	Среда разработки ПО	25
4.4.2.	Языки программирования и компиляторы	26
4.4.3.	Среда испытаний ПО	26
4.5.	Стандарты на разработку ПО	26
4.6.	Рассмотрение и гарантия процесса планирования создания ПО	27
5.	ПРОЦЕССЫ РАЗРАБОТКИ ПО	28
5.1.	Процесс разработки требований к ПО	28
5.1.1.	Цели процесса разработки требований к ПО	28
5.1.2.	Мероприятия процесса разработки требований к ПО	29
5.2.	Процесс проектирования ПО	29
5.2.1.	Цели процесса проектирования ПО	29
5.2.2.	Мероприятия проектирования ПО	30
5.2.3.	Проектирование ПО, модифицируемого пользователем	30
5.3.	Процесс кодирования	31
5.3.1.	Цель процесса кодирования	31
5.3.2.	Мероприятия процесса кодирования	31
5.4.	Процесс интеграции	32
5.4.1.	Цель процесса интеграции	32

© НИИАО 2002.

Documents
Document Sharing Hub
Full document details: KT-178B: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

№ п/п	Наименование раздела	Стр.
5.4.2.	Мероприятия процесса интеграции	32
5.4.3.	Обстоятельства, которые следует учитывать в процессе интеграции	32
5.5.	Трассируемость	33
6.	ПРОЦЕСС ВЕРИФИКАЦИИ ПО	34
6.1.	Цели процесса верификации ПО	34
6.2.	Мероприятия процесса верификации ПО	35
6.3.	Рассмотрения и анализы ПО	35
6.3.1.	Рассмотрения и анализы требований высокого уровня	36
6.3.2.	Рассмотрения и анализы требований низкого уровня	36
6.3.3.	Рассмотрения и анализы архитектуры ПО	37
6.3.4.	Рассмотрения и анализы «Исходного кода»	37
6.3.5.	Рассмотрения и анализы результатов процесса интеграции	38
6.3.6.	Рассмотрения и анализы тестовых примеров, процедур и результатов испытаний	38
6.4.	Испытания ПО	39
6.4.1.	Среда испытаний	40
6.4.2.	Выбор тестовых примеров на основе требований	40
6.4.2.1.	Тестовые примеры в допустимом диапазоне	41
6.4.2.2.	Робастные тестовые примеры	41
6.4.3.	Методы испытаний на основе требований	41
6.4.4.	Анализ полноты испытаний	43
6.4.4.1.	Анализ тестовых покрытий на основе требований	43
6.4.4.2.	Анализ структурного покрытия	43
6.4.4.3.	Устранение недостатков по результатам анализа структурного покрытия	43
7.	ПРОЦЕСС УПРАВЛЕНИЯ КОНФИГУРАЦИЕЙ ПО	45
7.1.	Цели процесса управления конфигурацией ПО	45
7.2.	Мероприятия процесса управления конфигурацией ПО	45
7.2.1.	Идентификация конфигурации	45
7.2.2.	Базовые версии и трассируемость	46
7.2.3.	Сообщения о проблемах, отслеживание и корректирующие действия	47
7.2.4.	Управление изменениями	47
7.2.5.	Рассмотрение изменений	48
7.2.6.	Учет состояния конфигурации	48
7.2.7.	Архивирование, воспроизведение и выпуск	48
7.2.8.	Контроль загрузки ПО	49
7.2.9.	Контроль за состоянием среды жизненного цикла ПО	49
7.3.	Категории контроля данных	50
8.	ПРОЦЕСС ГАРАНТИИ КАЧЕСТВА ПО	51
8.1.	Цели процесса гарантии качества ПО	51
8.2.	Мероприятия процесса гарантии качества ПО	51
8.3.	Рассмотрение соответствия ПО требованиям	52
9.	ПРОЦЕСС ВЗАИМОДЕЙСТВИЯ ЗАЯВИТЕЛЯ И СЕРТИФИЦИРУЮЩЕГО ОРГАНА	53
9.1.	Методы определения соответствия и планирование	53
9.2.	Обоснование соответствия	53
9.3.	Минимальный состав документов жизненного цикла ПО, представляемых сертифицирующему органу	53
9.4.	Документы жизненного цикла ПО, относящиеся к типовой конструкции	54
10.	ОБЗОРНОЕ ОПИСАНИЕ ПРОЦЕССА СЕРТИФИКАЦИИ ВОЗДУШНОГО СУДНА И ДВИГАТЕЛЯ	55
10.1.	Сертификационный базис	55
10.2.	Учет ПО при сертификации воздушного судна и двигателя	55
10.3.	Определение соответствия	55
11.	ДОКУМЕНТЫ ЖИЗНЕННОГО ЦИКЛА ПО	56
11.1.	План сертификации ПО	57

© НИИАО 2002.

Documents
Document Sharing Hub
Full document details: KT-178B: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

№ п/п	Наименование раздела	Стр.
11.2.	План разработки ПО	58
11.3.	План верификации ПО	58
11.4.	План управления конфигурацией ПО	59
11.5.	План гарантии качества ПО	60
11.6.	Стандарты на разработку требований ПО	61
11.7.	Стандарты на проектирование ПО	61
11.8.	Стандарты на кодирование	62
11.9.	Требования к ПО	62
11.10.	Описание проекта ПО	62
11.11.	Исходный код	63
11.12.	Исполняемый объектный код	63
11.13.	Примеры и процедуры верификации ПО	63
11.14.	Результаты верификации ПО	63
11.15.	Каталог среды жизненного цикла ПО	64
11.16.	Каталог комплектации ПО	64
11.17.	Сообщения о проблемах	65
11.18.	Протоколы УКПО	65
11.19.	Протоколы ГКПО	65
11.20.	Итоговое заключение о ПО	65
12.	ДОПОЛНИТЕЛЬНЫЕ УКАЗАНИЯ	67
12.1.	Использование ранее разработанного ПО	67
12.1.1.	Модификация ранее разработанного ПО	67
12.1.2.	Установка на другое воздушное судно	67
12.1.3.	Изменение условий применения или среды разработки	68
12.1.4.	Доработка базовой версии	69
12.1.5.	Указания по вопросам управления конфигурацией ПО	69
12.1.6.	Указания по вопросам гарантии качества ПО	69
12.2.	Квалификация инструмента	70
12.2.1.	Критерии квалификации инструментов для разработки ПО	71
12.2.2.	Критерии квалификации инструментов для верификации ПО	71
12.2.3.	Документы квалификации инструмента	72
12.2.3.1.	План квалификации инструмента	72
12.2.3.2.	Эксплуатационные требования к инструменту	72
12.2.4.	Одобрение квалификации инструмента	73
12.3.	Альтернативные методы	73
12.3.1.	Формальные методы	73
12.3.2.	Испытания по методу полного перебора	74
12.3.3.	Указания по верификации многоверсионного разнородного ПО	75
12.3.3.1.	Независимость многоверсионного разнородного ПО	75
12.3.3.2.	Верификация ПО, реализуемого на нескольких процессорах	76
12.3.3.3.	Верификация многоверсионного «Исходного кода»	76
12.3.3.4.	Квалификация инструмента для многоверсионного разнородного ПО	76
12.3.3.5.	Применение нескольких имитаторов при верификации	76
12.3.4.	Модели надёжности ПО	77
12.3.5.	Заархивированный период эксплуатации	77
	ПРИЛОЖЕНИЕ А	79
	ПРИЛОЖЕНИЕ В. ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ	90
	ПЕРЕЧЕНЬ ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	98
	БЛАНК ДЛЯ ИЗЛОЖЕНИЯ ПРЕДЛОЖЕНИЙ ПО СОВЕРШЕНСТВОВАНИЮ	99

© НИИАО 2002.

Documents
Document Sharing Hub
Full document details: КТ-178В: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

1. ВВЕДЕНИЕ

1.1. Назначение документа

Назначением данного документа является изложение инструктивных материалов относительно создания программного обеспечения (ПО) бортовых систем и оборудования, которые выполняют предписанные им функции с уровнем доверия к безопасности, удовлетворяющим требованиям к летной годности.

Эти инструктивные материалы имеют форму:

- Целей процессов жизненного цикла ПО.
- Описания мероприятий и конструктивных соображений для достижения этих целей.
- Описания доказательных материалов, подтверждающих, что цели достигнуты.

1.2. Область применения документа

В документе рассматриваются те аспекты сертификации, которые относятся к созданию ПО для бортовых систем и оборудования, используемых на воздушных судах или двигателях.

Чтобы облегчить понимание процесса сертификации, при обсуждении этих вопросов описывается жизненный цикл системы и его связь с жизненным циклом ПО.

Полное описание жизненного цикла ПО, включая процесс оценки безопасности системы и процесс удостоверения пригодности или процесса сертификации самолета и двигателя, не является назначением данного документа.

1.3. Связь с другими документами

Дополнительно к требованиям летной годности существуют национальные и международные стандарты на ПО.

В некоторых сообществах может потребоваться их выполнение.

Однако обращение к таким стандартам или предложение методов применения этих стандартов как альтернативы или дополнения к данному документу выходит за его рамки.

Везде, где в документе используется термин «стандарты», его следует понимать как стандарты, принимаемые разработчиком бортовой системы, бортового оборудования, двигателя или воздушного судна в конкретном проекте.

Такие стандарты могут быть разработаны на основании общих стандартов, созданных или приспособленных изготовителем для своих мероприятий.

1.4. Рекомендации относительно работы с документом

При использовании документа следует иметь в виду:

- Чтобы облегчить читателю понимание обсуждаемого вопроса, в текст документа включены пояснительные материалы.

Например, в **разделе 2** дается информация, необходимая для понимания взаимосвязи между жизненными циклами системы и ПО.

Раздел 3 представляет собой описание жизненного цикла ПО, а **раздел 10** - обзорное описание процесса сертификации воздушного судна или двигателя.

- Документ предназначен для использования и на уровне международного авиационного сообщества.

Чтобы способствовать этому, ссылки на конкретные национальные нормы и правила сведены к минимуму.

Применяются общеупотребляемые термины.

Например, используется термин «Сертифицирующий орган», который означает организацию или лицо, дающее удостоверение летной годности от имени страны, ответственной за сертификацию воздушного судна или двигателя.

Если другая страна или группа стран санкционирует такую сертификацию или участвует в ней, данный документ может быть использован при должном признании двусторонних соглашений или меморандумов о взаимопонимании между странами.

- В документе признается, что изложенные в нем инструктивные материалы не имеют обязательной юридической силы, но представляют единодушное мнение авиационного сообщества.

Кроме того, признается, что Заявитель может использовать методы, отличающиеся от приведенных в настоящем документе.

© НИИАО 2002.

AVA Documents
Document Sharing Hub

Full document details: KT-178B: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

- В документе устанавливаются цели для уровней ПО, определения которых даны в п. 2.2.2.

В **Приложении А** раскрывается содержание этих целей в зависимости от уровня ПО.

Если Заявитель выбирает данный документ для решения вопросов сертификации, его можно применять как сборник инструктивных материалов относительно удовлетворения этих целей.

- В **разделе 11** содержится перечень материалов, которые обычно составляются для того, чтобы помочь сертификации ПО.

В тексте документа наименования таких документов выделяются с помощью кавычек, например, «Исходный код».

- В **разделе 12** обсуждаются дополнительные указания, касающиеся использования ранее разработанного ПО, квалификации инструмента и применения методов, альтернативных описанным в **разделах 2 ... 11**.

Указания **раздела 12** могут применяться не во всяком сертификационном процессе.

- Таблицы целей и выходных документов процессов в зависимости от уровня ПО и толковый словарь терминов содержатся в **Приложениях**, являющихся нормативными частями данного документа.
- В тех случаях, когда для иллюстрации возможного применения инструктивных материалов используются примеры в графической или повествовательной форме, их не следует рассматривать в качестве предпочтительных методов.
- Перечень рассмотренных вопросов не следует считать исчерпывающим.
- В данном документе примечания используются для изложения пояснительных материалов, придания особого значения некоторому вопросу или для привлечения внимания к смежным вопросам, выходящим за рамки контекста.

Примечания не содержат инструктивных материалов.

© НИИАО 2002.

Documents
Document Sharing Hub
Full document details: КТ-178В: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

1.5. Структура документа

Структура данного документа в виде перечня разделов и их взаимосвязи приведена на Рис. 1-1.

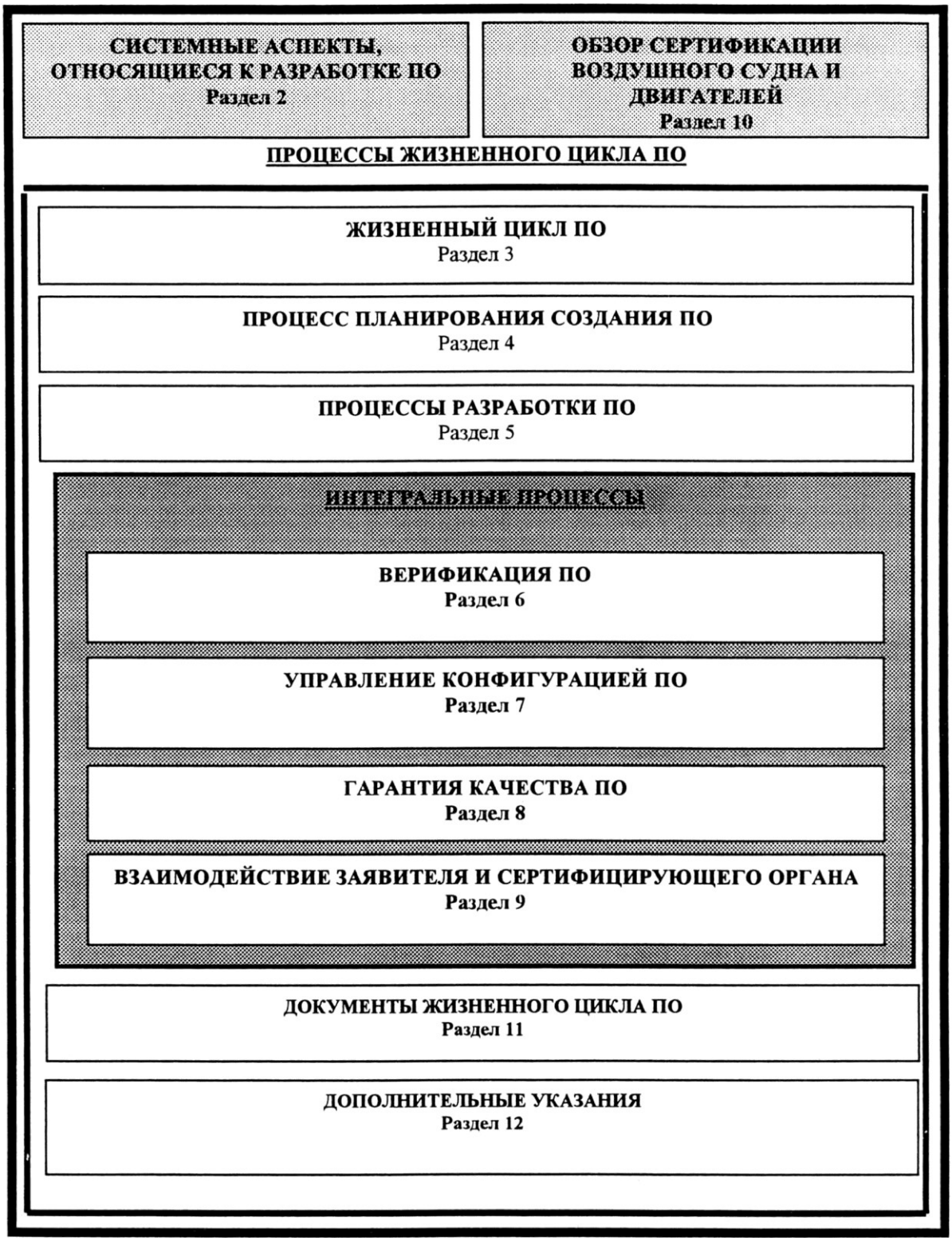


Рис. 1-1. Структура документа КТ-178В.

© НИИАО 2002.
Формат А4
Documents
Document Sharing Hub
Full document details: KT-178B: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

1.6. Отличия данного документа от DO-178B

В настоящем документе сохранены все разделы и подразделы документа **DO-178B «Software Considerations in Airborne Systems and Equipment Certification»** кроме **Дополнений**.

Раздел 1 дополнен настоящим подразделом.

В **Приложении В** приведен ряд терминов и определений, установленных в ГОСТ 27.002-89 [8].

© НИИАО 2002.

Documents
Document Sharing Hub
Full document details: КТ-178В: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

2. СИСТЕМНЫЕ АСПЕКТЫ, ОТНОСЯЩИЕСЯ К РАЗРАБОТКЕ ПО

В данном разделе рассматриваются аспекты процессов жизненного цикла системы, которые необходимы для понимания процессов жизненного цикла ПО.

К таким аспектам относятся:

- Обмен данными между процессами жизненного цикла системы и программного обеспечения (п. 2.1).
- Установление категории критичности, определение уровней ПО и назначение уровня ПО (п. 2.2).
- Учет особенностей архитектуры системы (п. 2.3).
- Учет влияния системных аспектов на ПО, модифицируемое пользователем, опционное ПО и готовое коммерческое ПО (п. 2.4).
- Учет влияния схемы построения системы на ПО, загружаемое на месте эксплуатации (п. 2.5).
- Учет требований к системе при верификации ПО (п. 2.6).
- Учет ПО при верификации системы (п. 2.7).

2.1. Информационные потоки между процессами жизненных циклов системы и программного обеспечения

На Рис. 2-1 представлены потоки информации, связанные с обеспечением безопасности, между процессами жизненного цикла системы и процессами жизненного цикла ПО.

Ввиду взаимосвязи процесса оценки безопасности системы и процесса разработки системы, рассматриваемые информационные потоки имеют итерационный характер.



Рис. 2-1. Потоки информации, связанные с безопасностью, между процессами жизненных циклов системы и ПО

© НИИАО 2002.

Documents
Document Sharing Hub
Full document details: КТ-178В: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

2.1.1. Информационный поток из процессов системы в процессы ПО

Процесс оценки безопасности системы устанавливает отказные состояния системы и определяет их категории.

В ходе этого процесса в результате анализа проекта системы устанавливаются требования, связанные с безопасностью её применения, которые определяют желаемую степень устойчивости и реакцию системы в отказных состояниях.

Такие требования устанавливаются для аппаратуры и ПО с целью исключения или ограничения влияния неисправностей и могут обеспечивать обнаружение неисправностей и устойчивость к последним.

В связи с тем, что решения принимаются в процессе проектирования аппаратуры и в процессах разработки ПО, то в процессе оценки безопасности системы проводится анализ результирующего проекта системы с целью подтверждения того, что этот проект удовлетворяет требованиям к безопасности.

Требования, связанные с обеспечением безопасности, являются частью требований к системе, причем последние являются исходными данными для процессов жизненного цикла ПО.

Чтобы гарантировать надлежащий учет требований к безопасности в течение этого цикла, в требованиях к системе обычно даётся следующая информация или ссылки на документы, в которых она содержится:

- Описание системы и характеристика аппаратного обеспечения.
- Сертификационные требования, включающие Квалификационные требования и другие нормативные документы.
- Требования к системе, относящиеся к требованиям к ПО, в том числе:
  - функциональные требования,
  - эксплуатационные требования,
  - требования к безопасности.
- Уровень(и) ПО и данные по их обоснованию, отказные состояния, их категории, а также соответствующие функции, относящиеся к ПО.
- Концепции обеспечения безопасности и проектные ограничения, включая методы проектирования, такие, как обособление, разнородность, резервирование и контроль.
- Если система является составной частью другой системы, указываются требования к безопасности системы более высокого уровня.

Для поддержки мероприятий верификации системы процессы жизненного цикла системы могут формулировать требования к процессам жизненного цикла ПО.

2.1.2. Информационный поток из процессов ПО в процессы системы

Процесс оценки безопасности системы учитывает влияние проекта и реализации ПО на безопасность системы на основе информации, получаемой из процессов жизненного цикла ПО.

Эта информация включает границы, распространения неисправностей, требования к ПО, описание архитектуры ПО, источники ошибок, которые могут быть обнаружены или устранены за счет выбора архитектуры ПО, посредством использования инструментов или с помощью других методов, применяемых при проектировании программного обеспечения.

Трассируемость связей между требованиями к системе и проектом ПО важна для процесса оценки безопасности системы.

Изменения, вносимые в ПО, могут повлиять на безопасность системы, и поэтому они должны быть определены для их оценки в процессе оценки безопасности системы.

2.2. Отказное состояние и уровень ПО

Ниже изложен материал относительно классификации отказных состояний, определения уровней ПО, связи между уровнями программного обеспечения и категориями отказных состояний, а также приведены рекомендации по назначению уровней ПО.

© НИИАО 2002.

Documents
Document Sharing Hub
Full document details: КТ-178В: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

Категория отказного состояния системы устанавливается посредством определения тяжести отказных состояний для воздушного судна и находящихся в нём людей.

Ошибка ПО может явиться причиной неисправности, способствующей появлению отказного состояния.

Таким образом, уровень ПО в целом, необходимый для безопасности, соотносится с отказными состояниями системы.

2.2.1. Классификация отказных состояний

Для получения исчерпывающих сведений о классификации отказных состояний следует обращаться к **Авиационным правилам, части 23, 25, 29, раздел А-0**.

Приведенный здесь перечень отказных состояний заимствован из указанных документов и включен в данный документ для облегчения его использования.

Классификация категорий отказных состояний следующая:

a. Катастрофическое - отказное состояние, для которого принимается, что при его возникновении предотвращение гибели людей оказывается практически невозможным.

b. Аварийное - отказное состояние, которое может привести к значительному ухудшению характеристик воздушного судна, и/или физическому утомлению или такой рабочей нагрузке экипажа, что уже нельзя полагаться на то, что он выполнит свои задачи точно и полностью.

c. Сложное - отказное состояние, которое может привести к заметному ухудшению характеристик воздушного судна, и/или выходу одного или нескольких параметров за эксплуатационные ограничения, но без достижения предельных ограничений, и/или уменьшению способности экипажа справиться с неблагоприятными условиями, как из-за увеличения рабочей нагрузки, так и из-за условий, понижающих эффективность действий экипажа.

d. Усложнение условий полёта - отказное состояние, которое может привести к незначительному ухудшению характеристик воздушного судна, и/или незначительному увеличению рабочей нагрузки на экипаж.

e. Без последствий - отказное состояние, которое не влияет на характеристики воздушного судна и не увеличивает рабочую нагрузку на экипаж.

2.2.2. Определения уровней ПО

Уровень ПО основан на вкладе программного обеспечения в возможные отказные состояния, определяемые в процессе оценки безопасности системы.

Уровень ПО предполагает, что объем работ, выполнение которых необходимо для доказательства соответствия сертификационным требованиям, изменяется в зависимости от категории отказного состояния.

Определения уровней ПО следующие:

a. Уровень А: ПО, ненормальная работа которого согласно оценке, полученной в процессе анализа безопасности системы, может вызвать или способствовать отказу функции системы, приводящему к катастрофическим отказным состояниям для воздушного судна.

b. Уровень В: ПО, ненормальная работа которого согласно оценке, полученной в процессе анализа безопасности системы, может вызвать или способствовать отказу функции системы, приводящему к аварийным отказным состояниям для воздушного судна.

c. Уровень С: ПО, ненормальная работа которого согласно оценке, полученной в процессе анализа безопасности системы, может вызвать или способствовать отказу функции системы, приводящему к сложным отказным состояниям для воздушного судна.

d. Уровень D: ПО, ненормальная работа которого согласно оценке, полученной в процессе анализа безопасности системы, может вызвать или способствовать отказу функции системы, приводящему к отказным состояниям типа «усложнение условий полета» воздушного судна.

e. Уровень Е: ПО, ненормальная работа которого согласно оценке, полученной в процессе анализа безопасности системы, может вызвать или способствовать отказу функции системы без влияния на эксплуатационные возможности воздушного судна или загрузку пилота.

© НИИАО 2002.

AVA Documents Document Sharing Hub
Full document details: KT-178B: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

Получение от сертифицирующего органа подтверждения, что данное программное обеспечение относится к **Уровню Е**, означает, что к нему не применимы изложенные ниже положения данного документа.

2.2.3. Установление уровня ПО

Вначале по результатам мероприятий, выполняемых в процессе оценки безопасности системы, назначается уровень(ни) ПО, соответствующий(ие) компонентам программного обеспечения конкретной системы без учета особенностей её построения.

При этом учитывается воздействие отказа, проявляющееся как в невозможности выполнения функции, так и в нарушении её выполнения.

Примечание:

(1) Заявитель может пожелать рассмотреть запланированные функции, которые будут введены в дальнейшем, а также возможные изменения в требованиях к системе, относящиеся к ПО.

Это может привести к отказным состояниям более опасной категории и более высокому уровню ПО.

Поэтому может оказаться желательным разработать ПО более высокого уровня, чем тот, который был определён в процессе оценки безопасности системы для её первоначального применения, так как последующая разработка документов, устанавливающих применение с более высоким уровнем ПО, может оказаться затруднительной.

(2) Уровень ПО бортовых систем и оборудования, которые не влияют на лётную годность воздушного судна, но установка которых на борту является обязательной согласно эксплуатационным правилам (примером является аварийный самописец полётных данных), должен быть соразмерен выполняемой ими функции.

В некоторых случаях уровень ПО может быть установлен в стандартах, определяющих минимальные требования к характеристикам оборудования.

Если ненормальная работа некоторой компоненты ПО способствует возникновению более чем одного отказного состояния, то уровень этой компоненты определяется ее наиболее опасным отказным состоянием.

Существуют различные концепции архитектурного построения программного обеспечения (например, описанные в п. 2.3), которые могут привести к пересмотру уровня(ей) ПО в ходе эволюционного совершенствования проекта системы.

Для выполнения некоторой функции системы может быть выделена одна или более обособленных компонент ПО.

Параллельной реализацией является такая, при которой функция системы выполняется несколькими компонентами ПО таким образом, что отказное состояние возникает в случае ненормальной работы более чем одной компоненты.

При параллельной реализации по крайней мере одна из компонент ПО должна иметь уровень ПО, соответствующий наиболее тяжелой категории отказного состояния для функции системы.

Уровни ПО других компонент определяются категорией отказного состояния, соответствующего потере этой функции.

Примеры такой реализации описаны в п. 2.3.2 **«Многоверсионное разнородное ПО»** и п. 2.3.3 **«Обеспечение безопасности с помощью непрерывного контроля»**.

Последовательной реализацией является такая, при которой для выполнения функции системы используется несколько компонент ПО таким образом, что отказное состояние может возникнуть при ненормальной работе любой из этих компонент.

При такой реализации компоненты ПО будут иметь уровень ПО, соответствующий наиболее тяжелой категории отказного состояния функции системы.

© НИИАО 2002.

Documents
Document Sharing Hub
Full document details: КТ-178В: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

Разработка ПО, соответствующего некоторому уровню ПО, не подразумевает назначения интенсивности его отказов.

Таким образом, уровни ПО или показатели его надежности, основанные на уровнях ПО, не могут быть использованы в процессе оценки безопасности системы, как используются интенсивности отказов аппаратного обеспечения.

Концепции построения, которые не соответствуют инструктивным материалам данного параграфа, следует обосновывать в процессе оценки безопасности системы.

2.3. Учет особенностей архитектуры системы

Если в процессе оценки безопасности системы установлено, что благодаря её архитектуре предотвращается возникновение наиболее опасного отказного состояния из числа возникающих по вине ненормальной работы ПО, то его уровень определяется категорией наиболее опасного из остальных отказных состояний, возникновению которых может способствовать ненормальная работа ПО.

В процессе оценки безопасности системы рассматриваются архитектурные решения для определения, влияют ли они на уровень ПО и функционирование ПО.

Ниже приведен инструктивный материал относительно нескольких концепций архитектурного построения систем, использование которых может ограничить влияние ошибок или позволяет обнаружить ошибки и обеспечить приемлемую реакцию системы на них.

Эти методы архитектурного построения не следует рассматривать в качестве предпочитаемых или обязательных.

2.3.1. Обособление

Обособление - это метод, позволяющий отделить функционально независимые компоненты ПО с целью сдерживания и/или изоляции отказов и потенциального снижения объёма работ, связанных с верификацией ПО.

Если предусматривается защита с использованием метода обособления, то уровень ПО каждой обособленной компоненты определяется наиболее опасной категорией отказного состояния, связанного с данной компонентой.

При использовании метода обособления следует придерживаться следующих инструктивных указаний:

a. При проектировании защиты по методу обособления для оценки потенциальной возможности нарушения обособления следует учесть следующие характеристики системы:

(1) ресурсы аппаратного обеспечения: процессоры, память, устройства ввода/вывода, прерывания и таймеры,

(2) связь по управлению: уязвимость к доступу извне,

(3) связь по данным: общие или перекрывающиеся данные, включая стеки и регистры процессора,

(4) отказные режимы работы аппаратного обеспечения, связанные с действием механизмов защиты.

b. Процессы жизненного цикла ПО должны учитывать проектные решения, связанные с применением метода обособления, включая степень и область взаимодействия, которые разрешены для обособленных компонент, а также способ реализации такой защиты: с помощью аппаратных средств или с помощью сочетания аппаратных и программных средств.

c. Если защита по методу обособления связана с использованием ПО, то такому программному обеспечению следует назначать уровень ПО, соответствующий наивысшему из уровней обособленных компонент ПО.

© НИИАО 2002.

Documents
Document Sharing Hub
Full document details: KT-178B: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

2.3.2. Многоверсионное разнородное ПО

Многоверсионное разнородное ПО - это метод проектирования системы, связанный с созданием двух или более его компонент, обеспечивающих выполнение одной и той же функции таким образом, чтобы избежать источников общих ошибок для компонент.

Этот метод называют также многоверсионным программным обеспечением, разнородным программным обеспечением, **N**-версионным кодированием или вариантным программным обеспечением.

Процессы жизненного цикла ПО, выполненные или начатые до применения метода разнородности, остаются потенциальными источниками ошибок.

В требованиях к системе может оговариваться конфигурация аппаратных средств, обеспечивающая реализацию многоверсионного разнородного ПО.

Степень разнородности и, следовательно, степень защиты, обычно не поддается измерению.

Вероятность потери функции системы возрастает до величины, которая зависит от вероятности обнаружения с помощью непрерывного контроля разнородных версий ПО действительных ошибок или переходных процессов, выходящих за пороги срабатывания устройств сравнения.

Поэтому разнородные версии ПО обычно используются в качестве средства дополнительной защиты, применяемой после того, как цели процесса верификации для рассматриваемого уровня ПО, описанные в **разделе 6**, удовлетворены.

Если в процессе оценки безопасности системы можно в итоге показать приемлемость потенциальной утраты функции системы, то методы верификации многоверсионного разнородного ПО могут быть сведены к методам верификации одноверсионного ПО.

Верификация многоверсионного разнородного ПО рассмотрена в п. 12.3.3.

2.3.3. Обеспечение безопасности с помощью непрерывного контроля

Непрерывный контроль для обеспечения безопасности - это метод защиты от конкретных отказных состояний с помощью непосредственного непрерывного контроля правильности выполнения функции для отказов, которые могут способствовать возникновению отказного состояния.

Функции контроля могут быть реализованы с использованием аппаратных средств или сочетания аппаратных и программных средств.

С помощью применения методов непрерывного контроля, уровень ПО контролируемой функции может быть понижен до уровня, соответствующего утрате данной функции системы.

Чтобы такое понижение стало допустимым, следует установить, что средство контроля обладает следующими тремя отличительными особенностями:

a. Уровень ПО. Программному обеспечению, осуществляющему контроль, назначен уровень ПО, соответствующий наиболее опасной категории отказного состояния из числа тех, которые связаны с выполнением контролируемой функции.

b. Полнота контроля неисправностей системы. Оценка полноты контроля неисправностей системы контролирующим устройством гарантирует, что схема этого устройства и её реализация таковы, что неисправности, подлежащие обнаружению, будут обнаружены при всех необходимых условиях.

c. Независимость средств контроля и средств, обеспечивающих выполнение функции. Устройство контроля и защищаемая схема не теряют работоспособность в результате возникновения одного и того же отказа, который вызывает опасную ситуацию.

© НИИАО 2002.

Documents
Document Sharing Hub
Full document details: KT-178B: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

2.4. Учет влияния системных аспектов на ПО, модифицируемое пользователем, опционное ПО и готовое коммерческое ПО

Влияние модификаций, вносимых пользователями, устанавливается в процессе оценки безопасности системы и используется для разработки требований к ПО, а затем и при разработке мероприятий процесса верификации.

Проектирование ПО, модифицируемого пользователем, рассматривается далее в п. 5.2.3.

Изменение, которое влияет на не модифицируемое программное обеспечение, его защиту или на границы модифицируемого программного обеспечения, является модификацией ПО и рассматривается в п. 12.1.1.

В данном документе модифицируемой компонентой считается такая часть ПО, для которой предусмотрена возможность изменения пользователем, а не модифицируемой - компонента, для которой такая возможность не предусмотрена.

Некоторые бортовые системы могут выполнять опционные функции, выбор которых осуществляется не с помощью программирующих контактов на штепсельных разъёмах аппаратуры, а с помощью специального ПО.

Для выбора конкретной конфигурации ПО в целевом вычислителе используются специальные функции программного обеспечения.

Инструктивный материал по вопросу отключенного кода изложен в п. 5.4.3.

Ниже приведены инструкции, относящиеся к ПО, модифицируемому пользователем, опционному ПО, и к готовому коммерческому ПО.

a. ПО, модифицируемое пользователем. Если в требованиях к системе предусмотрена возможность модификации ПО пользователем, то он может в пределах установленных ограничений вносить изменения в программы без их повторного рассмотрения сертифицирующим органом.

b. В требованиях к системе следует четко оговорить меры, предотвращающие возможность влияния внесенных пользователем изменений на безопасность, вне зависимости от того, правильно или нет реализуются такие изменения.

Программам, которые обеспечивают защиту от ошибок при внесении изменений пользователем, следует назначать тот же уровень ПО, что и программам в модифицируемой компоненте, которые они защищают.

c. Если в требованиях к системе не предусмотрена её модификация пользователем, то он не должен вносить изменения в ПО без демонстрации их соответствия данному документу.

d. На период времени, в течение которого пользователь вносит изменения, ему следует взять на себя ответственность за всё, что связано с модифицируемым ПО, например, ответственность за верификацию, управление конфигурацией и гарантию качества программного обеспечения.

e. Опционное ПО. В тех случаях, когда в состав ПО включены дополнительные функции, выбираемые программным способом, следует предусмотреть меры предосторожности, гарантирующие невозможность случайного выбора конфигурации ПО, не утверждённой для данного целевого вычислителя и данных условий его установки.

f. Готовое коммерческое ПО. Если такое программное обеспечение используется в бортовых системах, то оно должно удовлетворять требованиям данного документа.

g. Если документов, составленных в течение жизненного цикла готового коммерческого ПО, недостаточно, то их состав должен быть дополнен таким образом, чтобы удовлетворялись требования к безопасности, изложенные в данном документе.

В подобных случаях могут быть полезны указания, изложенные в п. 12.1.4 **Доработка базовой версии** и в п. 12.3.5 **Заархивированный период эксплуатации**.

© НИИАО 2002.

Documents
Document Sharing Hub
Full document details: КТ-178В: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

2.5. Учет влияния схемы построения системы на ПО, загружаемое на месте эксплуатации

К программному обеспечению, загружаемому на месте эксплуатации, относятся такие программы или таблицы данных, которые могут быть загружены без демонтажа системы или оборудования с места установки.

Требования, связанные с обеспечением безопасности при загрузке такого ПО, являются частью требований к системе.

Если случайное включение загрузки данных, относящихся к такому ПО, может привести к возникновению в системе отказного состояния, то требования к безопасности загрузки оговариваются в требованиях к системе.

При использовании ПО, загружаемого на месте эксплуатации, следует учитывать следующие факторы, связанные с безопасностью применения системы:

- наличие средств обнаружения неправильной или неполной загрузки,
- наличие средств обнаружения последствий загрузки несанкционированного ПО,
- аппаратно-программная совместимость,
- программно-программная совместимость,
- совместимость ПО с воздушным судном,
- возможность непреднамеренного включения загрузки на месте эксплуатации,
- возможность потери или неправильной индикации идентификатора конфигурации ПО.

Ниже приведены инструкции, относящиеся к ПО, загружаемому на месте эксплуатации:

a. Если в ходе работ, выполняемых для оценки безопасности системы, не обоснована возможность отступления от изложенного ниже правила, то программу, обнаруживающую неполную или неправильную загрузку ПО, следует относить к источникам отказных состояний наиболее опасной категории или назначать ей наиболее высокий уровень ПО из тех, которые связаны с функцией, для выполнения которой загружается данное программное обеспечение.

b. Если в системе предусмотрен режим загрузки несанкционированных программ или данных «по умолчанию», то для каждой обособленной компоненты системы следует сформулировать требования к безопасности использования этого режима, учитывающие возможное отказное состояние.

c. При реализации функции загрузки следует предусмотреть меры, обеспечивающие обнаружение неправильных комбинаций «программное обеспечение/аппаратное обеспечение/воздушное судно» и защиту, соответствующую отказному состоянию этой функции.

При этом следует учитывать системы и процедуры, используемые для поддержки загрузки.

d. Если ПО является частью бортового средства отображения, гарантирующего что воздушное судно соответствует сертифицированному типу, тогда либо данное ПО должно быть разработано по уровню ПО, соответствующему самому высокому уровню загружаемого ПО, либо в процессе оценки безопасности системы должна быть подтверждена целостность проверки идентификатора конфигурации программного обеспечения.

2.6. Учет требований к системе при верификации ПО

Требования к системе разрабатываются на основе соответствующих эксплуатационных требований и требований, связанных с безопасностью.

Последние являются результатом процесса оценки безопасности системы.

При этом учитывается следующее:

a. В требованиях к системе указываются две обязательные характеристики ПО:

(1) ПО должно выполнять предписанные ему функции так, как это указано в требованиях к системе.

(2) ПО не должно приводить к ненормальной работе, определенной в процессе оценки безопасности системы.

© НИИАО 2002.

AVA Documents Document Sharing Hub
Full document details: КТ-178В: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

Для исключения ненормальной работы формулируются дополнительные требования к системе.

b. Эти требования к системе затем трансформируются в требования к ПО высокого уровня, которые проверяются при выполнении мероприятий процесса верификации ПО.

2.7. Учет ПО при верификации системы

Инструктивный материал по верификации (проверкам и испытаниям) системы выходит за рамки данного документа.

Однако, процессы жизненного цикла ПО поддерживают процесс верификации системы и взаимодействуют с ним.

Для поддержки верификации системы необходимы сведения о построении ПО, связанные с функционированием системы.

Верификация системы может обеспечивать значительное покрытие структуры кода.

Анализ покрытия испытаний при верификации системы может быть использован и для достижения целей покрытия различных видов испытаний, рассматриваемых при верификации ПО.

© НИИАО 2002.

Documents
Document Sharing Hub
Full document details: КТ-178В: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

3. ЖИЗНЕННЫЙ ЦИКЛ ПО

В данном разделе рассматриваются процессы, образующие жизненный цикл ПО, даётся определение этого понятия, описываются критерии перехода от одного процесса к другому.

Инструктивный материал, изложенный в данном документе, не предписывает предпочтительный цикл, а описывает отдельные процессы, которые входят в большинство циклов и взаимосвязи между ними.

Разделение процессов не означает, что оно отражает структуру организации(ий), выполняющей эти работы.

Для каждого программного продукта планируется его жизненный цикл(ы), состоящий из описанных ниже процессов.

3.1. Процессы жизненного цикла ПО

Процессами жизненного цикла ПО являются:

- Процесс планирования создания ПО, который определяет и координирует мероприятия процессов разработки программного обеспечения и интегральных процессов в рамках проекта.

Процесс планирования описывается в **разделе 4**.

- Процессы разработки ПО, в результате выполнения которых получается программный продукт.

Эти процессы включают:

- процесс разработки требований к ПО,
- процесс проектирования ПО,
- процесс кодирования ПО,
- процесс интеграции.

Эти процессы описаны в **разделе 5**.

- Интегральные процессы, которые гарантируют корректность, управляемость и доверие к процессам жизненного цикла ПО и их результатам.

Интегральные процессы включают:

- процесс верификации ПО,
- процесс управления конфигурацией ПО,
- процесс гарантии качества ПО,
- процесс взаимодействия с сертифицирующим органом..

Важно понимать, что интегральные процессы выполняются одновременно с процессом разработки ПО в течение всего жизненного цикла ПО.

Описание интегральных процессов даётся в **разделах 6...9**.

3.2. Определение жизненного цикла ПО

Проект, как правило, определяет один или более жизненных циклов ПО посредством выбора мероприятий для каждого из процессов, установления их последовательности и назначения ответственных за эти мероприятия.

В конкретном проекте последовательность этих процессов определяется его характерными особенностями: составом выполняемых функций и сложностью системы, объёмом и сложностью ПО, стабильностью требований, использованием результатов предыдущих разработок, концепциями, положенными в основу разработки, наличием аппаратного обеспечения.

Обычная последовательность процессов в цикле следующая: требования, проект, кодирование и интеграция.

© НИИАО 2002.

Documents
Document Sharing Hub
Full document details: KT-178B: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

Рис. 3-1 иллюстрирует последовательность процессов при разработке нескольких компонент одного программного продукта, имеющих различные жизненные циклы.

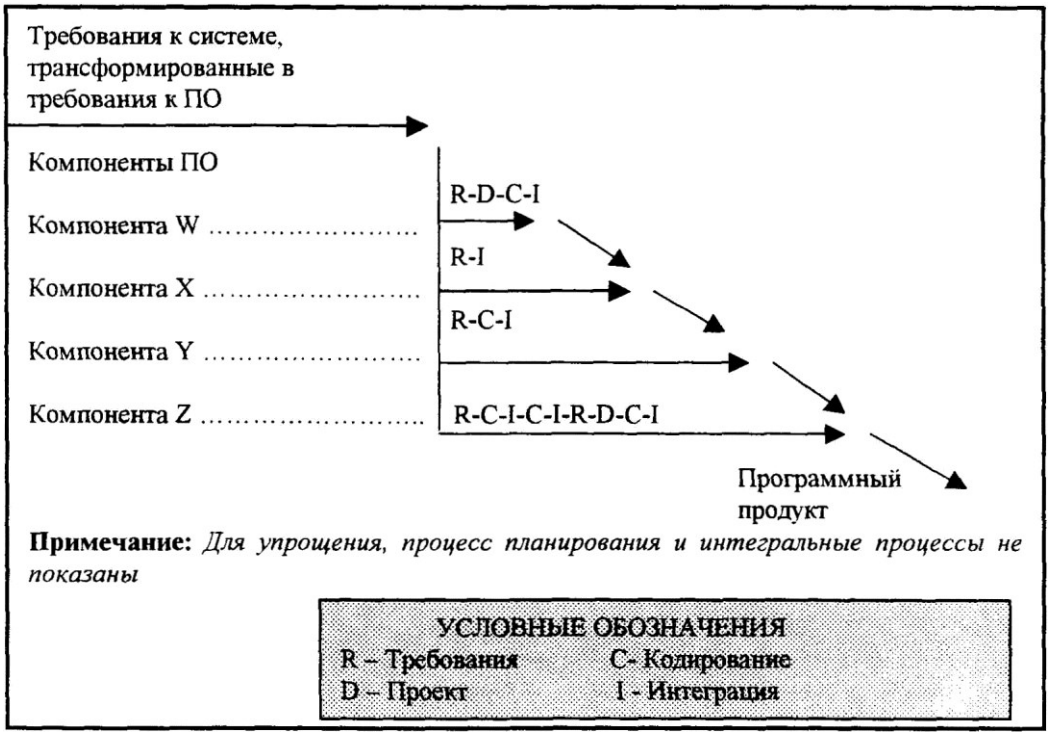


Рис. 3-1. Пример проекта ПО, в котором используются четыре различные последовательности разработки.

Компонента «W» реализует набор требований к системе посредством разработки требования к ПО.

На основе этих требований составляется проект ПО, который далее воплощается в Исходный код и интегрируется с аппаратурой.

Компонента «X» иллюстрирует применение ранее разработанного ПО, используемого на сертифицированном воздушном судне или двигателе.

Компонента «Y» иллюстрирует применение ПО простой обособленной функции, для которой Исходный код можно написать прямо на основании требований к программному обеспечению.

Компонента «Z» иллюстрирует метод разработки, основанный на построении прототипа.

Обычно, прототип создается для лучшего понимания требований к ПО и для уменьшения объёма доводочных работ и технического риска.

Исходные требования используются как базис прототипа.

Его работа оценивается в ожидаемых условиях эксплуатации разрабатываемой системы.

Результаты оценки используются для уточнения требований.

Процессы жизненного цикла ПО могут иметь итерационный характер, то есть, могут повторяться.

Время выполнения и степень уточнения при совершении итерационных шагов изменяются в зависимости от степени наращивания функций в процессе разработки, сложности, степени уточнения требований, наличия аппаратуры, наличия обратной связи в предшествующие процессы и от других особенностей проекта.

Различные этапы жизненного цикла ПО, выбранного для реализации, связаны между собой последовательным процессом интеграции и мероприятиями процесса верификации.

© НИИАО 2002.
Documents
Document Sharing Hub
Формат А4
Full document details: КТ-178В: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

3.3. Критерии перехода между процессами

Эти критерии используются для определения, можно ли начинать или повторно начинать некоторый процесс.

Каждый процесс жизненного цикла ПО определяет мероприятия, связанные с обработкой входных данных для получения выходных данных.

Процесс может порождать обратные связи в другие процессы, а также использовать обратные связи из других процессов.

Формирование обратной связи включает решение вопросов, каким образом информация распознается, контролируется и используется для принятия решения принимающим процессом.

Примером обратной связи являются сообщения о проблемах.

Критерии перехода зависят от запланированной последовательности процессов разработки ПО и интегральных процессов, на них также может влиять уровень ПО.

Примерами критериев перехода могут быть следующие:

- рассмотрения процесса верификации завершены,
- входным объектом является идентифицированная единица конфигурации,
- анализ трассируемости для входного объекта завершен.

Если удовлетворены критерии перехода для данного вида процесса, то нет необходимости в том, чтобы к моменту его начала были получены все входные данные.

Инструкции следующие:

a. Если для некоторого процесса входные данные поступают по частям, то при поступлении каждого очередного их набора следует проверить, остались ли в силе предыдущие выходные данные процессов разработки и верификации

???

© НИИАО 2002.

Documents
Document Sharing Hub
Full document details: КТ-178В: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

4. ПРОЦЕСС ПЛАНИРОВАНИЯ СОЗДАНИЯ ПО

В данном разделе описываются цели и мероприятия процесса планирования создания ПО.

Результатом процесса планирования являются планы и стандарты, которые непосредственно определяют процессы разработки ПО и интегральные процессы.

Таблица А-1 Приложения А представляет собой сводку целей и результатов процесса планирования для разных уровней ПО.

4.1. Цели процесса планирования

Назначением процесса планирования является определение методов и средств, используемых для создания ПО, удовлетворяющего требованиям к системе, а также позволяющих получить уровень доверия к программному обеспечению, удовлетворяющий требованиям лётной годности.

Цели процесса планирования следующие:

a. С учетом требований к системе и уровня(ей) ПО следует определить мероприятия процессов разработки и интегральных процессов жизненного цикла программного обеспечения (п. 4.2).

b. Следует определить жизненный(ые) цикл(ы) ПО с указанием взаимосвязей между процессами, последовательности их выполнения, схемы обратных связей и критериев перехода (**Раздел 3**).

c. Следует выбрать среду жизненного цикла ПО, включая методы и инструменты, которые будут использованы в мероприятиях каждого процесса жизненного цикла (п. 4.4).

d. При необходимости следует учесть дополнительные инструктивные материалы, подобные приведенным в **Разделе 12**.

e. Следует установить стандарты на разработку ПО, согласованные с требованиями к безопасности системы (п. 4.5).

f. Планы создания ПО должны соответствовать требованиям, изложенным в п. 4.3 и **Разделе 11**.

g. Следует обеспечить координацию разработки и пересмотра планов ПО (п. 4.3).

4.2. Мероприятия процесса планировании создания ПО

Для создания ПО, соответствующего требованиям данного документа, определяющим фактором является эффективное планирование.

Рекомендации по процессу планирования следующие:

a. Планы ПО следует разрабатывать так, чтобы обеспечить своевременное руководство для персонала, участвующего в процессе разработки и интегральных процессах.

Инструктивный материал приведен в п. 9.1.

b. Следует сформулировать или выбрать стандарты разработки ПО для использования в рассматриваемом проекте.

c. Следует выбрать методы и инструменты, обеспечивающие предотвращение ошибок в процессе разработки ПО.

d. Процесс планирования должен обеспечить координацию процессов разработки и интегральных процессов таким образом, чтобы обеспечить последовательность стратегий, изложенных в планах.

e. При планировании следует предусмотреть процедуру пересмотра планов по мере осуществления проекта.

f. Если в системе используется многоверсионное разнородное ПО, то в процессе планирования следует выбрать такие методы и инструменты предотвращения или обнаружения ошибок, которые необходимы для удовлетворения требований к безопасности системы.

g. Для завершения процесса планирования планы и стандарты на разработку ПО должны находиться под управлением изменениями, а их рассмотрения должны быть завершены (п. 4.6).

© НИИАО 2002.

AVA Documents Document Sharing Hub
Full document details: КТ-178B: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

h. Если предусматривается наличие отключенного кода (п. 2.4), то процесс планирования должен описывать, как отключенный код (выбираемые опции, лётные испытания) будет определяться, верифицироваться и как с ним нужно обращаться для того, чтобы удовлетворять требованиям к безопасности системы.

i. Если предусматривается код, модифицируемый пользователем, то в планах и в стандартах следует указать процесс, инструменты, среду и данные, доказывающие выполнение указаний п. 5.2.3.

Некоторые процессы жизненного цикла ПО могут начинаться до завершения процесса планирования, если планы и процедуры для конкретного мероприятия процесса уже имеются.

4.3. Планы создания ПО

Назначением этих планов является определение средств, обеспечивающих достижение целей данного документа.

Планы определяют также организации (группы исполнителей), которые будут выполнять указанные в планах мероприятия.

Ниже приведен перечень планов:

• **«План сертификации ПО» (п. 11.1)** является основным документом, в котором предлагаемые методы разработки программного обеспечения представляются сертифицирующему органу на утверждение, и документом, в котором указываются методы определения соответствия требованиям данного документа.

• **«План разработки ПО» (п. 11.2)** определяет жизненный цикл(ы) и среду разработки ПО.

• **«План верификации ПО» (п. 11.3)** определяет средства, с помощью которых будут достигаться цели процесса верификации.

• **«План управления конфигурацией ПО» (п. 11.4)** определяет средства, с помощью которых будут достигаться цели процесса управления конфигурацией программного обеспечения.

• **«План гарантии качества ПО» (п. 11.5)** определяет средства, с помощью которых будут достигаться цели процесса гарантии качества программного обеспечения.

Инструкции, касающиеся планов создания ПО, следующие:

a. Планы следует составлять в соответствии с данным документом.

b. В планах следует определить критерии перехода между процессами цикла создания ПО, указав при этом:

(1) Входные данные процесса, включая обратные связи от других процессов.

(2) Мероприятия интегральных процессов, которые могут потребоваться для реакции на эти входные данные.

(3) Доступность инструментов, методов, планов и процедур.

c. В планах создания ПО следует установить процедуры внесения изменений в программное обеспечение до его применения на сертифицированном воздушном судне или двигателе.

Такие изменения могут быть результатами обратных связей от других процессов и они могут приводить к необходимости внесения изменений в планы создания ПО.

4.4. Планирование среды жизненного цикла ПО

Целью планирования среды жизненного цикла ПО является определение методов, инструментов, процедур, языков программирования и аппаратных средств, используемых для разработки, контроля, выпуска документации жизненного цикла ПО (**Раздел 11**) и программного продукта.

Примером положительного влияния среды на бортовое ПО является введение стандартов, средств обнаружения ошибок, а также методов предотвращения ошибок и обеспечение отказобезопасности.

Среда, в которой осуществляется жизненный цикл ПО, является потенциальным источником ошибок, которые могут привести к отказным состояниям.

© НИИАО 2002.

Documents
Document Sharing Hub
Full document details: КТ-178В: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

На состав средств, образующих среду, могут оказывать влияние требования к безопасности, установленные в результате оценки безопасности системы, например, требование применения разнородных резервированных компонент.

Целью методов предотвращения ошибок является исключение в процессах разработки ПО тех из них, которые могут способствовать возникновению отказных состояний.

Основным принципом при этом является выбор методов разработки требований и методов проектирования, инструментов и языков программирования, ограничивающих возможность внесения ошибок, а также выбор методов верификации, гарантирующих обнаружение внесенных ошибок.

Целью методов обеспечения устойчивости к неисправностям является внесение в проект ПО и в Исходный код таких особенностей обеспечения безопасности, которые гарантируют, что программное обеспечение будет правильно реагировать на ошибки входных данных и не допустит ошибок на выходе и в управлении.

Необходимость применения методов предотвращения ошибок или методов обеспечения работоспособности при возникновении неисправностей устанавливается требованиями к системе и процессом анализа её безопасности.

Изложенные выше соображения могут оказать влияние на:

- Методы и систему обозначений, используемые в процессе разработки требований к ПО и в процессе проектирования программного обеспечения.
- Используемые языки и методы в процессе кодирования.
- Инструменты среды разработки ПО.
- Инструменты, используемые при верификации и управлении конфигурацией.
- Необходимость квалификации инструментов (п. 12.3).

4.4.1. Среда разработки ПО

Среда, в которой осуществляется разработка ПО, является важным фактором производства ПО высокого качества.

Среда же различными путями может оказать и неблагоприятное влияние на бортовое ПО.

Например, компилятор может внести ошибки, исказив откомпилированную программу, или редактор связей может не обнаружить ошибку, допущенную при распределении памяти.

Инструкции относительно выбора методов и инструментов, используемых для разработки ПО, следующие:

a. В процессе планирования следует выбрать среду, минимизирующую риск её неблагоприятного влияния на готовое бортовое ПО.

b. Выбор квалифицированных инструментов или комбинации инструментов и элементов среды следует осуществлять таким образом, чтобы достигался необходимый уровень уверенности, что ошибка, внесенная в одном элементе среды, будет обнаружена в другом элементе.

Приемлемая среда формируется тогда, когда оба элемента используются согласованно.

c. Чтобы свести к минимуму ошибки, вносимые средой, следует предусмотреть мероприятия процесса верификации ПО или стандарты на разработку программного обеспечения, учитывающие уровень программного обеспечения.

d. Если Заявитель планирует использовать комбинации инструментов, то в соответствующем плане следует оговорить последовательность их работы.

e. Если в проекте предусматривается использование опционных особенностей инструментов разработки ПО, то влияние этих опций должно быть проверено и отражено в соответствующем плане.

Примечание: Это особенно важно в тех случаях, когда инструменты непосредственно генерируют часть программного продукта.

В этом отношении компиляторы, по видимому, являются наиболее важными из инструментов, подлежащих рассмотрению.

© НИИАО 2002.

Documents
Document Sharing Hub
Full document details: KT-178B: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

4.4.2. Языки программирования и компиляторы

Компилятор считается пригодным для создания программного продукта после успешного завершения верификации данного продукта.

Чтобы это было справедливым, мероприятия процесса верификации должны рассматривать характерные особенности языков программирования и компиляторов.

Процесс планирования рассматривает эти особенности при выборе языка программирования и планировании верификации.

Инструкции по этому вопросу следующие:

a. Некоторые компиляторы имеют особенности, связанные с оптимизацией характеристик объектного кода.

Если контрольные примеры дают покрытие, соответствующее уровню ПО, нет необходимости проверять правильность оптимизации.

В противном случае следует оценить влияние таких особенностей на анализ структурного покрытия в соответствии с указаниями, изложенными в п. 6.4.4.2.

b. Компиляторы ряда языков, реализуя присущие им характерные особенности, могут создавать объектный код, не трассируемый непосредственно на Исходный код, например, инициализация, встроенные средства обнаружения ошибок, обработчики исключительных ситуаций (п. 6.4.4.2,b).

В процессе планирования ПО следует предусмотреть средства обнаружения такого объектного кода и гарантии верификационного покрытия и определить эти средства в соответствующем плане.

c. Если в течение жизненного цикла ПО вводятся новые версии компилятора, редактора связей или загрузчика или изменяются опции компилятора, то прежние испытания и анализ покрытия могут стать недействительными.

Поэтому при планировании верификации следует предусматривать средства повторной верификации в соответствии с инструктивным материалом, изложенным в **Разделе 6** и п. 12.1.3.

4.4.3. Среда испытаний ПО

Целью планирования среды, в которой осуществляются испытания ПО, является определение методов, инструментов, процедур и аппаратных средств, используемых для испытаний результатов процесса интеграции.

Испытания могут проводиться с использованием целевого вычислителя, эмулятора или имитатора.

Инструкции следующие:

a. Может потребоваться квалификация эмулятора или имитатора в соответствии с инструктивным материалом, изложенным в п. 12.2.

b. Следует оценить различия между целевым вычислителем и эмулятором или имитатором и оценить влияние этих различий на способность указанных средств обнаруживать ошибки и выполнять проверку выполнения функций.

Обнаружение таких ошибок должно обеспечиваться другими мероприятиями процесса верификации и это должно быть указано в **«Плане верификации ПО»**.

4.5. Стандарты на разработку ПО

Целью стандартов на разработку ПО является установление правил и ограничений для процессов разработки ПО.

В состав этих стандартов входят **«Стандарты на разработку требований к ПО»**, **«Стандарты проектирования ПО»** и **«Стандарты на кодирование»**.

В процессе верификации ПО указанные стандарты используются в качестве основы для оценки соответствия действительных выходных результатов некоторого процесса желаемым результатам.

Инструкции, относящиеся к стандартам на разработку ПО, следующие:

© НИИАО 2002.

Documents
Document Sharing Hub
Full document details: КТ-178В: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

a. Стандарты на разработку ПО должны соответствовать инструктивному материалу, изложенному в **Разделе 11**.

b. Стандарты на разработку ПО должны предоставлять возможность единообразного проектирования и реализации заданного программного продукта или комплекта продуктов.

c. Стандарты на разработку ПО должны запрещать использование конструкций или методов, приводящих к результатам, не поддающимся проверке или не совместимым с требованиями безопасности.

Примечание: В стандартах разработки следует учитывать предыдущий опыт.

В них для контроля сложности могут быть включены ограничения и правила, относящиеся к методам разработки, проектирования и кодирования.

Для улучшения робастности (**сохранение внешних характеристик** системы управления при изменении ее **внутренних характеристик**) могут быть учтены проверенные на практике приёмы «надежного» кодирования.

4.6. Рассмотрение и гарантия процесса планирования создания ПО

Рассмотрения и гарантии процесса планирования проводятся и выполняются для того, чтобы гарантировать, что планы и стандарты на разработку ПО соответствуют инструкциям данного документа и существуют средства, которые обеспечивают их выполнение.

Инструкции следующие:

a. Выбранные методы обеспечат достижение целей данного документа.

b. Процессы жизненного цикла ПО могут применяться согласованно.

c. Каждый процесс дает возможность убедиться в том, что его результаты могут быть трассированы на его мероприятия и входные данные, показывая степень независимости мероприятия, среду и используемые методы.

d. Результаты процесса планирования создания ПО согласованы и соответствуют инструкциям, изложенным в **Разделе 11**.

© НИИАО 2002.

Documents
Document Sharing Hub
Full document details: КТ-178В: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

5. ПРОЦЕССЫ РАЗРАБОТКИ ПО

В данном разделе рассматриваются цели и мероприятия процессов разработки ПО.

Эти процессы выполняются так, как это определено процессом планирования создания программного обеспечения (**Раздел 4**) и **«Планом разработки ПО»** (п. 11.2).

Таблица А-2 Приложения А представляет собой итоговую сводку целей и выходных данных процессов разработки ПО для программного обеспечения разных уровней.

Процессы разработки ПО следующие:

- Процесс разработки требований к ПО
- Процесс проектирования ПО
- Процесс кодирования
- Процесс интеграции

В процессах разработки ПО формулируются требования одного или более уровней.

Требования высокого уровня получаются непосредственно из анализа требований к системе и архитектуры её построения.

Обычно на дальнейших этапах проектирования они детализируются и формулируются в виде требований более низкого уровня или ряда последовательных более низких уровней.

Однако, если Исходный код генерируется непосредственно по требованиям высокого уровня, то принимается, что они же являются и требованиями низкого уровня, и к ним применим инструктивный материал, относящийся к последним.

Разработка архитектуры ПО связана с принятием решений относительно структуры.

В процессе проектирования ПО определяется его архитектура и формулируются требования низкого уровня.

Это – такие требования, по которым непосредственно, без дополнительной информации, можно составить исходный код.

Каждый процесс разработки ПО может порождать производные требования.

Это такие требования, которые не трассируются непосредственно на требования высокого уровня.

Примером производного требования является требование разработать программу обработки прерываний, зависящую от особенностей выбранного целевого вычислителя.

Производные требования могут быть включены в требования как высокого, так и низкого уровней.

Влияние производных требований на требования к безопасности определяется в процессе оценки безопасности системы.

5.1. Процесс разработки требований к ПО

Для составления требований высокого уровня процесс разработки требований к ПО использует выходные данные процесса жизненного цикла системы.

Эти требования высокого уровня включают функциональные и технические требования, требования к взаимодействию и требования к безопасности.

5.1.1. Цели процесса разработки требований к ПО

Целями этого процесса являются:

a. Разработка требований высокого уровня,

b. Выдача производных требований высокого уровня для процесса оценки безопасности системы.

© НИИАО 2002.

Documents
Document Sharing Hub
Full document details: КТ-178В: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

5.1.2. Мероприятия процесса разработки требований к ПО

Входными данными для процесса разработки требований к ПО являются: требования к системе, описание взаимодействия с аппаратными средствами и описание архитектуры системы (если оно не включено в требования к системе) из процесса жизненного цикла системы, **«План разработки ПО»** и **«Стандарты на разработку требований к ПО»** из процесса планирования создания ПО.

После выполнения критериев перехода эти входные данные используются для разработки требований к ПО высокого уровня.

Основным результатом этого процесса является документ «Требования к ПО» (п. 11.9).

Процесс разработки требований к ПО считается завершенным, когда его цели достигнуты и достигнуты цели интегральных процессов.

Инструкции для этого процесса следующие:

a. Следует проанализировать функциональные требования и требования к взаимодействию для исключения неоднозначности, противоречий и неопределенности условий.

b. Если входные данные для процесса разработки требований к ПО признаны недостаточными или неверными, то их следует вернуть на входы соответствующих процессов для внесения ясности или поправок.

c. Каждое требование к системе, отнесенное к ПО, следует включить в требования высокого уровня.

d. Следует определить те требования высокого уровня, связанные с требованиями к системе и отнесенные к ПО, которые введены для предотвращения опасности применения системы.

e. Требования высокого уровня должны удовлетворять «Стандартам на разработку требований к ПО» и быть проверяемыми и согласованными.

f. Требования высокого уровня следует устанавливать в количественной форме с указанием допусков там, где это применимо.

g. В требованиях высокого уровня не следует описывать детали разработки и верификации, за исключением описания установленных и обоснованных проектных ограничений.

h. Каждое требование к системе, отнесенное к ПО, должно быть трассируемо на одно или более требований высокого уровня.

i. Каждое требование высокого уровня ПО должно быть трассируемо на одно или более требований к системе, за исключением производных требований.

j. Производные требования высокого уровня должны быть переданы в процесс оценки безопасности системы.

5.2. Процесс проектирования ПО

Требования к ПО высокого уровня уточняются в течение одной или более итераций процесса проектирования ПО для определения его архитектуры и требований низкого уровня, по которым можно составить исходный код.

5.2.1. Цели процесса проектирования ПО

Целями этого процесса являются:

a. Архитектура ПО и требования низкого уровня разрабатываются на основе требований высокого уровня;

b. Производные требования низкого уровня направляются в процесс оценки безопасности системы.

© НИИАО 2002.

Documents
Document Sharing Hub
Full document details: KT-178B: Requirements for Software in Airborne Equipment and Systems for Aviation Certification

5.2.2. Мероприятия проектирования ПО

Входными данными для процесса проектирования ПО являются:
«Требования к ПО»,
«План разработки ПО» и
«Стандарты на проектирование ПО».

Требования высокого уровня используются для разработки архитектуры ПО и требований низкого уровня после того, как удовлетворяются указанные в плане критерии перехода.

Это может быть связано с формулированием требований одного или нескольких более низких уровней.

Основным результатом этого процесса является документ «Описание проекта ПО» (п. 11.10), в котором дается описание архитектуры программного обеспечения и формулируются требования низкого уровня.

Процесс проектирования ПО считается завершенным, когда его цели достигнуты и достигнуты цели интегральных процессов.

Инструкции для этого процесса следующие:

a. Требования низкого уровня и архитектура ПО, разработанные в процессе проектирования, должны удовлетворять «Стандартам на проектирование ПО», быть трассируемыми, проверяемыми и согласованными.

b. Чтобы гарантировать отсутствие нарушений требований высокого уровня, следует установить и проанализировать производные требования.

c. Мероприятия процесса проектирования ПО могут вносить определенные типы отказов или, наоборот, предотвращать другие.

Использование обособления или других архитектурных решений может изменить уровни ПО отдельных его компонент.

В таких случаях следует определить дополнительные производные требования и обеспечить их анализ в процессе оценки безопасности системы.

d. При формулировании требований к безопасности следует предусмотреть контроль потоков управления и данных, например, предусмотреть сторожевой таймер, проверку на непротиворечивость и перекрестное сравнение каналов.

e. Реакция на отказные состояния должна удовлетворять требованиям к безопасности.

f. Недостаточные или неправильные входные данные, выявленные в процессе проектирования, следует возвратить в процесс жизненного цикла системы, процесс разработки требований к ПО или в процесс планирования создания ПО для внесения ясности или поправок.

Примечание: Современный уровень технологии программирования не позволяет установить количественную связь между сложностью ПО и удовлетворением требований к безопасности.

И, хотя нет возможности дать объективные инструкции по этому вопросу, при проектировании программного обеспечения следует избегать его усложнения, поскольку в этом случае труднее верифицировать проект ПО и показать, что он удовлетворяет требованиям к безопасности.

5.2.3. Проектирование ПО, модифицируемого пользователем

Ниже изложен инструктивный материал, относящийся к ПО, предназначенному для модификации пользователем.

Модифицируемой компонентой является такая компонента ПО, которая предназначена для внесения в неё изменений пользователем, немодифицируемой – та, которая не предназначена для этой цели.

© НИИАО 2002.

Documents
Document Sharing Hub
Full document details: KT-178B: Requirements for Software in Airborne Equipment and Systems for Aviation Certification