ФЕДЕРАЛЬНОЕ АГЕНТСТВО
ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р МЭК 61508-3— 2007

ФУНКЦИОНАЛЬНАЯ БЕЗОПАСНОСТЬ СИСТЕМ ЭЛЕКТРИЧЕСКИХ, ЭЛЕКТРОННЫХ, ПРОГРАММИРУЕМЫХ ЭЛЕКТРОННЫХ, СВЯЗАННЫХ С БЕЗОПАСНОСТЬЮ

Часть 3

Требования к программному обеспечению

IEC 61508-3:1998
Functional safety of electrical/electronic/programmable electronic safety-related systems —
Part 3: Software requirements
(IDT)

Издание официальное

БЗ 3—2006/37

Москва
Стандартинформ
2008

gost.e-ecolog.ru

Предисловие

Цели и принципы стандартизации в Российской Федерации установлены Федеральным законом от 27 декабря 2002 г. № 184-ФЗ «О техническом регулировании», а правила применения национальных стандартов Российской Федерации — ГОСТ Р 1.0—2004 «Стандартизация в Российской Федерации. Основные положения»

Сведения о стандарте

1 ПОДГОТОВЛЕН Обществом с ограниченной ответственностью «Корпоративные электронные системы» и Техническим комитетом по стандартизации ТК 10 «Перспективные производственные технологии, менеджмент и оценка рисков» на основе собственного аутентичного перевода стандарта, указанного в пункте 4

2 ВНЕСЕН Управлением развития, информационного обеспечения и аккредитации Федерального агентства по техническому регулированию и метрологии

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 27 декабря 2007 г. № 582-ст

4 Настоящий стандарт идентичен международному стандарту МЭК 61508-3:1998 «Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 3. Требования к программному обеспечению» (IEC 61508-3:1998 «Functional safety of electrical/electronic/programmable electronic safety-related systems — Part 3: Software requirements»).

Наименование настоящего стандарта изменено относительно наименования указанного международного стандарта для приведения в соответствие с ГОСТ Р 1.5 (подраздел 3.5).

При применении настоящего стандарта рекомендуется использовать вместо ссылочных международных стандартов соответствующие им национальные стандарты, сведения о которых приведены в дополнительном приложении С

5 ВВЕДЕН ВПЕРВЫЕ

Информация об изменениях к настоящему стандарту публикуется в ежегодно издаваемом информационном указателе «Национальные стандарты», а текст изменений и поправок — в ежемесячно издаваемых информационных указателях «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ежемесячно издаваемом информационном указателе «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет

© Стандартинформ, 2008

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 2
3 Термины и определения 4
4 Соответствие настоящему стандарту 4
5 Документация 4
6 Система управления качеством программного обеспечения 4
7 Требования к жизненному циклу программного обеспечения, связанного с безопасностью 5
8 Оценка функциональной безопасности 28
Приложение А (обязательное) Руководство по выбору методов и средств 29
Приложение В (обязательное) Подробные таблицы 34
Приложение С (справочное) Сведения о соответствии национальных стандартов Российской Федерации ссылочным международным стандартам 37
Библиография 38

Введение

Системы, состоящие из электрических и/или электронных компонентов, в течение многих лет используются для выполнения функций безопасности в большинстве областей применения. Компьютерные системы [обычно называемые программируемыми электронными системами (PES)], применяемые во всех областях для выполнения задач, не связанных с безопасностью, во все более увеличивающихся объемах используются для решения задач обеспечения безопасности. Для эффективной и безопасной эксплуатации технологий, основанных на использовании компьютерных систем, чрезвычайно важно, чтобы лица, ответственные за принятие решений, имели в своем распоряжении руководства по вопросам безопасности, которые они могли бы использовать в своей работе.

Настоящий стандарт устанавливает общий подход к вопросам обеспечения безопасности для всего жизненного цикла систем, состоящих из электрических и/или электронных, и/или программируемых электронных компонентов [электрических/ электронных/программируемых электронных систем (E/E/PES)], которые используются для выполнения функций безопасности. Этот унифицированный подход был принят для того, чтобы разработать рациональную и последовательную техническую концепцию для всех электрических систем, связанных с безопасностью. Основной целью при этом является содействие разработке стандартов.

В большинстве случаев безопасность достигается за счет использования нескольких систем защиты, в которых применяются различные технологии (например, механические, гидравлические, пневматические, электрические, электронные, программируемые электронные). Любая стратегия безопасности должна, следовательно, учитывать не только все элементы, входящие в состав отдельных систем (например, датчики, управляющие устройства и исполнительные механизмы), но также и все подсистемы, связанные с безопасностью, входящие в состав комбинированной системы, связанной с безопасностью. Таким образом, хотя данный стандарт посвящен в основном электрическим/ электронным/ программируемым электронным (E/E/PE) системам, связанным с безопасностью, он может также предоставлять общую структуру, в рамках которой рассматриваются системы, связанные с безопасностью, основанные на других технологиях.

Признанным фактом является существование огромного разнообразия использования E/E/PES в различных областях применения, отличающихся различной степенью сложности, опасностями и возможными рисками. В каждом конкретном случае необходимые меры безопасности будут зависеть от многочисленных факторов, которые являются специфичными для этого применения. Настоящий стандарт, являясь базовым стандартом, позволит формулировать такие меры в будущих международных стандартах для областей применения.

Настоящий стандарт:

- рассматривает все соответствующие стадии жизненного цикла систем безопасности в целом, а также подсистем E/E/PES и программного обеспечения (например, начиная от исходной концепции, проектирование, разработку, эксплуатацию, техническое обслуживание и вывод из эксплуатации), в ходе которых подсистемы E/E/PES используются для выполнения задач обеспечения безопасности;
- был задуман с учетом быстрого развития технологий; его структура является достаточно устойчивой и полной для того, чтобы удовлетворять потребностям разработок, которые могут появиться в будущем;
- делает возможной разработку стандартов, предназначенных для прикладных отраслей и посвященных вопросам обеспечения безопасности на базе E/E/PES; разработка стандартов для прикладных отраслей в рамках общей структуры, вводимой настоящим стандартом, должна приводить к более высокому уровню согласованности (например, в отношении принципов, положенных в основу, терминологии и т.п.) как для отдельных прикладных отраслей, так и для их совокупности; это приносит преимущества как в плане безопасности, так и в плане экономики;
- предоставляет метод разработки спецификаций для требований безопасности, необходимых для достижения требуемой функциональной безопасности E/E/PE систем, связанных с безопасностью;
- использует уровни полноты безопасности для задания планируемого уровня полноты безопасности для функций, которые должны быть реализованы E/E/PE системами, связанными с безопасностью;
- использует для определения уровней полноты безопасности подход, основанный на оценке рисков;
- устанавливает количественные величины отказов E/E/PE систем, связанных с безопасностью, которые связаны с уровнями полноты безопасности;
- устанавливает нижний предел для планируемой величины отказов в режиме опасных отказов, который может быть задан для отдельной E/E/PE системы, связанной с безопасностью; для E/E/PE систем, связанных с безопасностью, работающих:

в режиме с низкой интенсивностью запросов нижний предел для выполнения планируемой функции по запросу устанавливается на средней вероятности отказов 10^{-5},

в режиме с высокой интенсивностью запросов, нижний предел устанавливается на вероятности опасных отказов 10^{-9} в час.

Примечание — Отдельная E/E/PE система, связанная с безопасностью, необязательно предполагает одноканальную архитектуру.

- применяет широкий набор принципов, методов и мер для достижения функциональной безопасности E/E/PE систем, связанных с безопасностью, но не использует концепцию безаварийности, которая может иметь важное значение, когда виды отказов хорошо определены, а уровень сложности является относительно невысоким. Концепция безаварийности признана неподходящей из-за широкого диапазона сложности E/E/PE систем, связанных с безопасностью, которые находятся в области применения настоящего стандарта.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФУНКЦИОНАЛЬНАЯ БЕЗОПАСНОСТЬ СИСТЕМ ЭЛЕКТРИЧЕСКИХ, ЭЛЕКТРОННЫХ, ПРОГРАММИРУЕМЫХ ЭЛЕКТРОННЫХ, СВЯЗАННЫХ С БЕЗОПАСНОСТЬЮ

Часть 3

Требования к программному обеспечению

Functional safety of electrical, electronic, programmable electronic safety-related systems.
Part 3. Software requirements

Дата введения — 2008—06—01

1 Область применения

1.1 Настоящий стандарт:

a) применяется совместно с МЭК 61508-1 и МЭК 61508-2;

b) применяется к любому программному обеспечению, являющемуся частью системы, связанной с безопасностью, либо используемому для разработки системы, связанной с безопасностью, в рамках области применения МЭК 61508-1 и МЭК 61508-2. Такое программное обеспечение называется программным обеспечением, связанным с безопасностью.

Программное обеспечение, связанное с безопасностью, включает в себя операционные системы, системное программное обеспечение, программы, используемые в коммуникационных сетях, интерфейсы пользователей и обслуживающего персонала, инструментальные средства поддержки, встроенные программно-аппаратные средства, а также прикладные программы.

Прикладные программы включают в себя программы высокого и низкого уровней, а также специальные программы на языках с ограниченной варьируемостью (МЭК 61508-4, пункт 3.2.7);

c) предусматривает определение функции безопасности и уровней целостности программного обеспечения.

Примечания

1 Если это уже было сделано как часть спецификации E/E/PE систем, связанных с безопасностью (МЭК 61508-2, пункт 7.2), то это не следует повторять в настоящем стандарте.

2 Определение функций безопасности и уровней полноты безопасности программного обеспечения представляет собой интерактивную процедуру; см. рисунки 2 и 6.

3 Структуру документации см. в МЭК 61508-1 (пункт 5 и приложение А). Структура документации может учитывать процедуры, используемые в компаниях, а также реальную практику, сложившуюся в отдельных прикладных отраслях.

d) устанавливает требования к стадиям жизненного цикла безопасности и действиям, которые должны предприниматься в процессе проектирования и разработки программного обеспечения, связанного с безопасностью (модель жизненного цикла безопасности программного обеспечения). Эти требования включают в себя применение мероприятий и методов, ранжированных по уровням полноты безопасности и предназначенных для того, чтобы избегать ошибок и отказов программного обеспечения и принимать необходимые меры при их возникновении;

Издание официальное

e) предоставляет требования к информации, относящейся к подтверждению безопасности программного обеспечения, которая должна передаваться в организацию, выполняющую интеграцию систем E/E/PES;

f) предоставляет требования к подготовке информации и процедур, касающихся программного обеспечения, которое необходимо пользователям для работы и поддержки E/E/ PE систем, связанных с безопасностью;

g) предоставляет требования, предъявляемые к организациям, выполняющим модификацию программного обеспечения, связанного с безопасностью;

h) предоставляет вместе с МЭК 61508-1 и МЭК 61508-2 требования к инструментальным средствам поддержки, таким как средства разработки и проектирования, средства тестирования и отладки, средства управления конфигурацией.

Примечание — На рисунках 4 и 6 показана взаимосвязь между МЭК 61508-2 и МЭК 61508-3.

1.2 МЭК 61508-1 — МЭК 61508-4 представляют собой основополагающие стандарты по безопасности, хотя этот статус не применяется в контексте E/E/PE систем, связанных с безопасностью, имеющих небольшую сложность (МЭК 61508-4, пункт 3.4.4). Как основополагающие стандарты по безопасности они предназначены для использования техническими комитетами при подготовке стандартов в соответствии с МЭК Руководство 104 и ИСО/МЭК Руководство 51. МЭК 61508-1 — МЭК 61508-4 предназначены, кроме того, для использования в качестве самостоятельных стандартов.

В круг обязанностей технического комитета входит использование, где это возможно, основополагающих стандартов по безопасности при подготовке собственных стандартов. В этом случае требования, методы проверки или условия проверки настоящего основополагающего стандарта по безопасности не будут применяться, если это не указано специально, или они будут включаться в стандарты, подготовленные этими техническими комитетами.

Примечание — В США и Канаде до тех пор, пока там не будет опубликована в качестве международного стандарта предлагаемая реализация МЭК 61508 для обрабатывающих отраслей (т.е. МЭК 61511), вместо МЭК 61508 в обрабатывающих отраслях допускается использовать национальный стандарт, базирующийся на МЭК 61508 (т.е. ANSI/ISA S 84.01—1996).

1.3 На рисунке 1 показана общая структура МЭК 61508-1 — МЭК 61508-7 и указана роль МЭК 61508-3 в достижении функциональной безопасности E/E/PE систем, связанных с безопасностью. В МЭК 61508-6 (приложение А) описано применение МЭК 61508-2 и МЭК 61508-3.

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие стандарты:

МЭК 61508-1:1998 Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 1. Общие требования

МЭК 61508-2:2000 Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 2. Требования к системам электрическим/электронным/программируемым электронным, связанным с безопасностью

МЭК 61508-4:1998 Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 4. Термины и определения

МЭК 61508-5:1998 Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 5. Примеры методов определения уровней полноты защиты

МЭК 61508-6:2000 Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 6. Руководство по применению МЭК 61508-2:2000 и МЭК 61508-3:1998

МЭК 61508-7:2000 Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 7. Анализ методов и средств

ИСО/МЭК Руководство 51:1999 Руководство по включению в стандарты аспектов, связанных с безопасностью

МЭК Руководство 104:1997 Подготовка публикаций по безопасности и использование основополагающих публикаций и групповых публикаций по безопасности

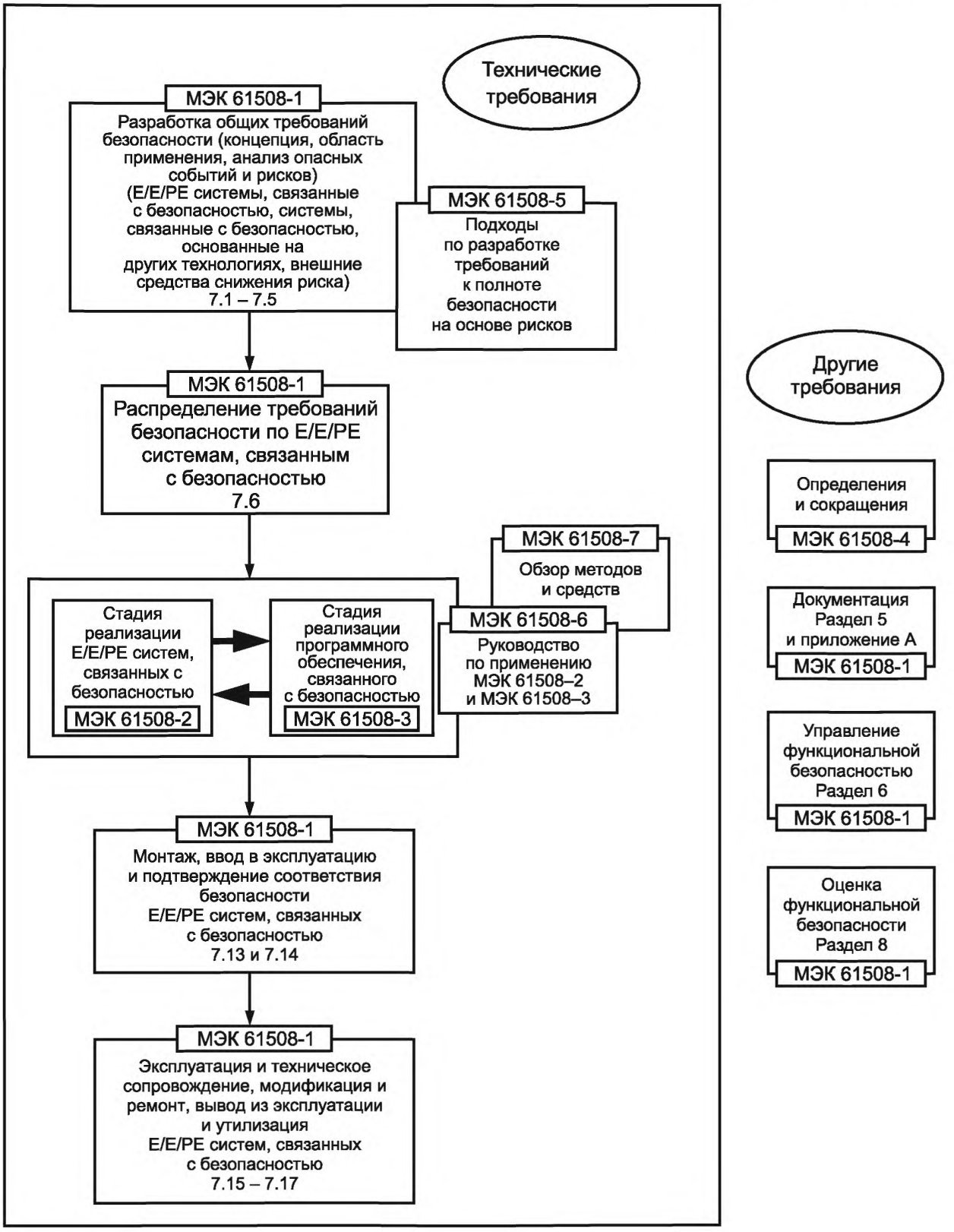


Рисунок 1 — Общая структура настоящего стандарта

3 Термины и определения

В настоящем стандарте использованы термины по МЭК 61508-4.

4 Соответствие настоящему стандарту

Требования, которые следует выполнять для соответствия настоящему стандарту, приведены в МЭК 61508-1 (раздел 4).

5 Документация

Задачи и требования, предъявляемые к документации, приведены в МЭК 61508-1 (раздел 5).

6 Система управления качеством программного обеспечения

6.1 Цели

Цели подробно рассмотрены в МЭК 61508-1 (пункт 6.1).

6.2 Требования

6.2.1 В дополнение к требованиям, описанным в МЭК 61508-1 (пункт 6.2), предъявляются следующие требования.

6.2.2 Планирование функциональной безопасности должно определять стратегию поставок, разработки, интеграции, верификации, приемки и модификации программного обеспечения в той мере, в какой этого требует уровень полноты безопасности E/E/PE системы, связанной с безопасностью.

Примечание — Философия настоящего подхода состоит в использовании планирования функциональной безопасности в качестве возможности для приспособления настоящего стандарта для учета различной степени полноты безопасности, необходимой для компонентов E/E/PE систем, связанных с безопасностью. При совместном использовании компонентов E/E/PE систем, связанных с безопасностью, имеющих разные уровни полноты безопасности, следует учитывать требования пункта 7.4.2.8.

6.2.3 Система управления конфигурацией программного обеспечения должна:

a) использовать административные и технические средства контроля на протяжении всего жизненного цикла программ для того, чтобы управлять изменениями в программах и таким образом гарантировать выполнение указанных в спецификациях требований к безопасности программных средств;

b) гарантировать выполнение всех операций, необходимых для того, чтобы продемонстрировать достижение заданной полноты безопасности программного обеспечения;

c) осуществлять аккуратную поддержку с использованием уникальной идентификации всех элементов конфигурации, которые необходимы для обеспечения целостности E/E/PE систем, связанных с безопасностью. Элементы конфигурации должны включать в себя, как минимум, следующее:

- анализ безопасности и требования к безопасности,
- спецификацию программного обеспечения и проектные документы,
- исходный текст программ,
- план и результаты тестирования,
- ранее разработанные программные компоненты и пакеты, которые должны быть включены в E/E/PE систему, связанную с безопасностью,
- все инструментальные средства и системы разработки, которые использовались при создании, тестировании или выполнении иных действий с программным обеспечением E/E/PE систем, связанных с безопасностью;

d) использовать процедуры контроля над внесением изменений для предотвращения несанкционированных модификаций; документировать запросы на выполнение модификаций; анализировать влияние предлагаемых модификаций и утверждать либо отвергать модификации; подробно документировать модификации и выдавать полномочия на выполнение всех утвержденных модификаций; устанавливать основные параметры конфигурации системы для этапов разработки программного обеспечения и документировать (частичное) тестирование интеграции системы, которое подтверждает выполнение задач этапа (см. 7.8); гарантировать объединение и встраивание всех подсистем программного обеспечения (включая переработку более ранних версий);

Примечание — Для осуществления руководства и применения административных и технических средств контроля необходимы принятие управленческих решений и наличие полномочий.

e) документировать перечисленную ниже информацию, для того чтобы обеспечить возможность последующего аудита: состояние конфигурации, текущее состояние системы, обоснование и утверждение всех модификаций, подробное описание всех модификаций;

f) строго документировать каждую версию программного обеспечения, связанного с безопасностью. Обеспечить хранение всех версий программного обеспечения и всей относящейся к ним документации для обеспечения возможности сопровождения и выполнения модификаций на протяжении всего периода использования разработанного программного продукта.

Примечание — Дополнительную информацию по управлению конфигурацией см. в ИСО/МЭК 12207.

7 Требования к жизненному циклу программного обеспечения, связанного с безопасностью

7.1 Общие положения

7.1.1 Цели

Целью требований, излагаемых в настоящем подразделе, является разделение процесса разработки программного обеспечения на этапы и процессы (см. таблицу 1 и рисунки 2 — 5).

7.1.2 Требования

7.1.2.1 Жизненный цикл систем безопасности при разработке программного обеспечения должен быть выбран и специфицирован при планировании безопасности в соответствии с МЭК 61508-1 (раздел 6).

Примечание — Модель жизненного цикла систем безопасности, удовлетворяющая требованиям МЭК 61508-1 (раздел 7), может быть переработана в соответствии с конкретными потребностями проекта или организации.

7.1.2.2 Процедуры оценки качества и безопасности должны быть интегрированы в процессы жизненного цикла систем безопасности.

7.1.2.3 Каждая фаза жизненного цикла безопасности программного обеспечения должна быть разделена на элементарные процессы. Для каждой стадии должны быть определены область применения, входные данные и выходные данные.

Примечания

1 Более подробная информация о фазах жизненного цикла приведена в ИСО/МЭК 12207.

2 Выходные данные стадий жизненного цикла систем безопасности рассматриваются в МЭК 61508-1 (раздел 5). При разработке некоторых E/E/PE систем, связанных с безопасностью, выходные данные некоторых стадий жизненного цикла систем безопасности могут представлять собой отдельные документы, тогда как выходные данные от других стадий могут объединяться в один документ. Существенным является требование, чтобы выходные данные стадии жизненного цикла системы безопасности удовлетворяли ее предназначению. В случае простых разработок некоторые стадии жизненного цикла систем безопасности также могут объединяться (см. 7.4.5).

7.1.2.4 Если жизненный цикл модулей безопасности удовлетворяет требованиям, приведенным на рисунке 3 и в таблице 1, допускается изменять глубину, число и рабочий размер стадий V-модели (см. рисунок 5) в соответствии с полнотой безопасности и сложностью проекта.

Примечание — Полный список стадий жизненного цикла, приведенный в таблице 1, относится к большим системам, разрабатываемым с самого начала. Для небольших систем может оказаться целесообразным, например, объединить стадии проектирования системы программного обеспечения и проектирования архитектуры.

7.1.2.5 При условии выполнения всех задач и требований настоящего раздела допускается организовывать программный проект иначе, чем предписывается настоящим стандартом (т.е. использовать иную модель жизненного цикла систем безопасности).

7.1.2.6 Для каждой стадии жизненного цикла следует использовать соответствующие методы и мероприятия. Рекомендации приведены в приложениях А и В (руководство по выбору методов и мероприятий). Выбор методов из приложений А и В не гарантирует сам по себе достижения необходимой полноты безопасности.

7.1.2.7 Результаты процессов жизненного цикла систем безопасности программного обеспечения должны быть документированы (см. раздел 5).

7.1.2.8 Если на какой-либо стадии жизненного цикла модулей безопасности программного обеспечения возникает необходимость внести изменение, относящееся к более ранней стадии жизненного цикла, необходимо повторно выполнять эту стадию и стадии, следующие за ней.



Рисунок 2 — Жизненный цикл безопасности E/E/PES систем (стадия реализации)



Рисунок 3 — Жизненный цикл безопасности программного обеспечения (стадия реализации)

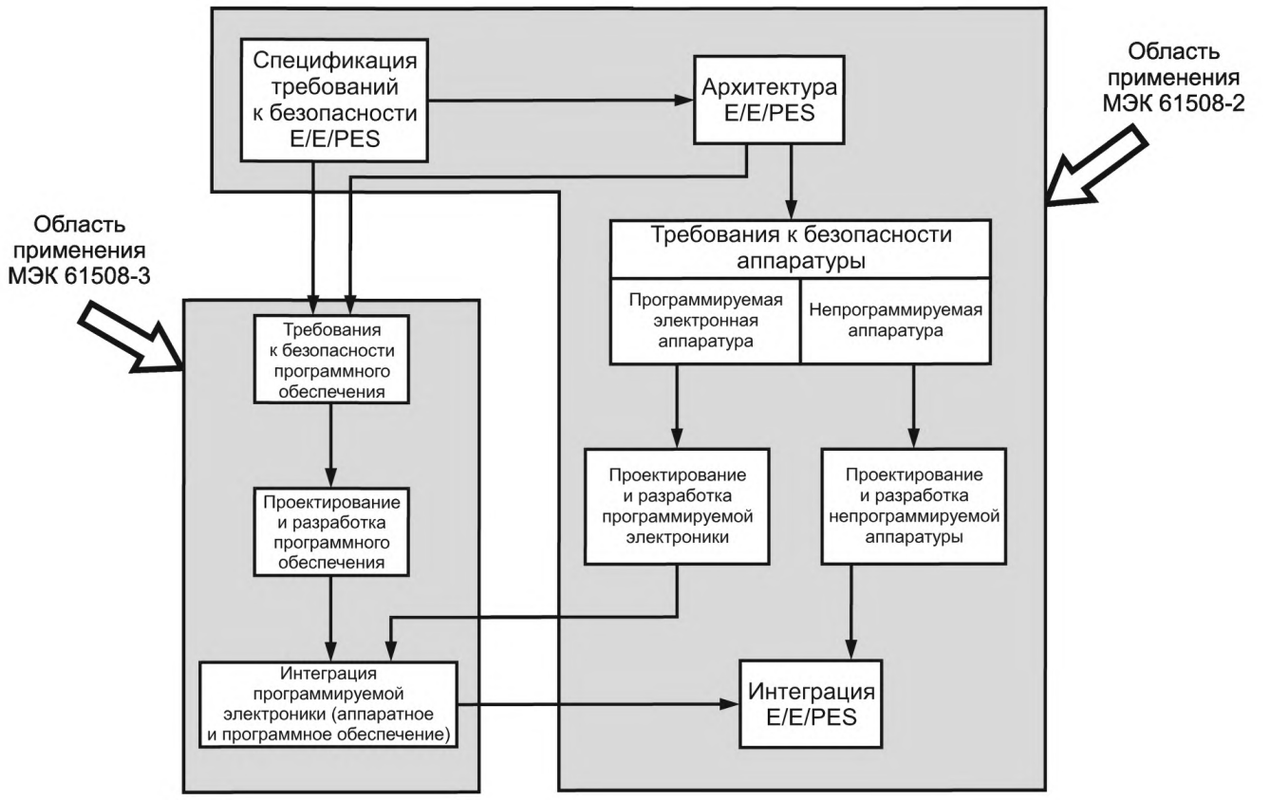


Рисунок 4 — Соотношение между областями применения МЭК 61508-2 и МЭК 61508-3

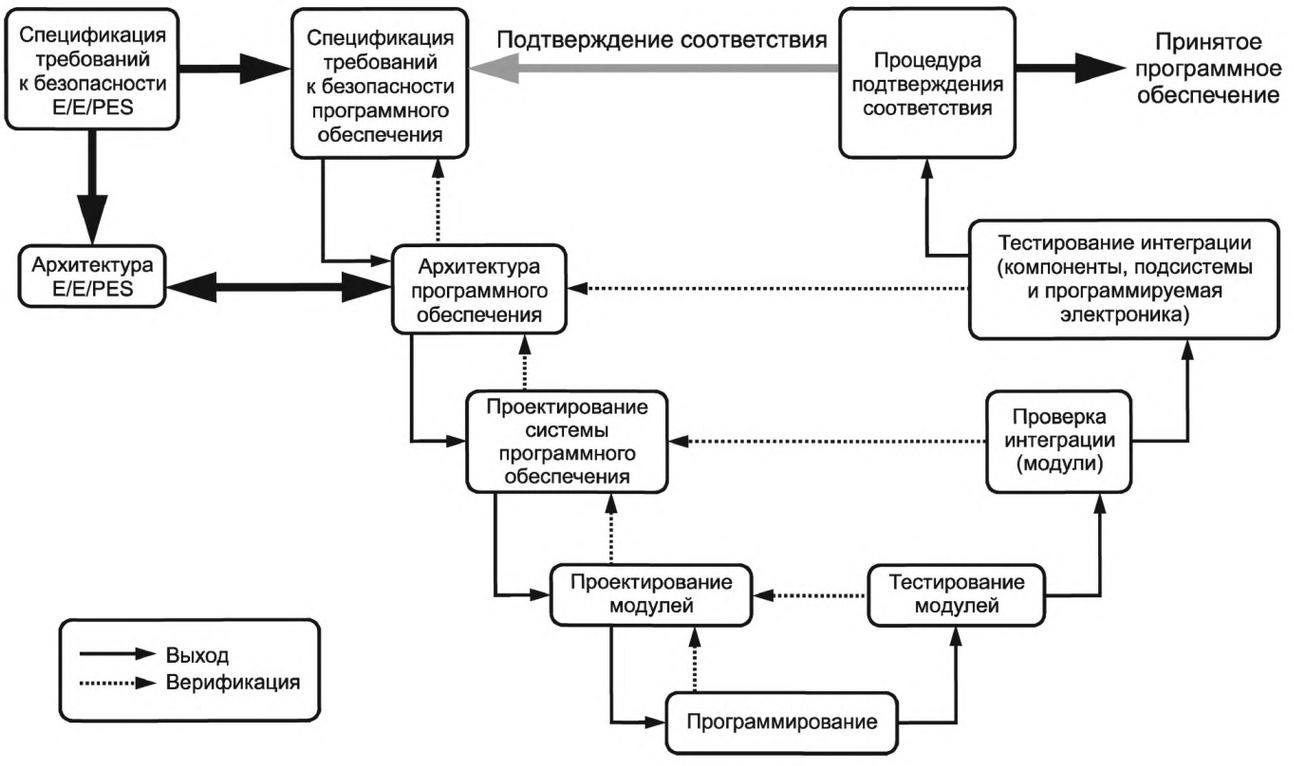


Рисунок 5 — Полнота безопасности программного обеспечения и жизненный цикл разработки (V-модель)

Таблица 1 — Обзор жизненного цикла модулей безопасности программного обеспечения

Стадия жизненного цикла безопасности (номер стадии соответствует номеру блока на рисунке 3)	Задача	Область применения	Номер пункта или раздела	Входные данные	Выходные данные
9.1 Спецификация требований к безопасности программного обеспечения	Указать требования к безопасности программного обеспечения в виде требований к функциям безопасности программного обеспечения и требований к полноте безопасности программного обеспечения. Указать необходимые для реализации требуемых функций безопасности требования к функциям безопасности программного обеспечения для каждой E/E/PE системы, связанной с безопасностью. Указать требования к полноте безопасности программного обеспечения для каждой E/E/PE системы, связанной с безопасностью, необходимые для достижения уровня полноты безопасности, указанного для каждой функции безопасности, назначенной этой E/E/PE системе, связанной с безопасностью	PES. Система программного обеспечения	7.2.2	Спецификация требований к безопасности E/E/PES (МЭК 61508-2)	Спецификация требований к безопасности программного обеспечения
9.2 Планирование подтверждения соответствия требованиям к безопасности программного обеспечения	Разработать план подтверждения соответствия требованиям к безопасности программного обеспечения	PES. Система программного обеспечения	7.3.2	Спецификация требований к безопасности программного обеспечения	План подтверждения соответствия требованиям к безопасности программного обеспечения
9.3 Проектирование и разработка программного обеспечения	Архитектура: разработать архитектуру программного обеспечения, которая удовлетворяет указанным требованиям к безопасности в отношении необходимого уровня полноты безопасности; рассмотреть и оценить требования, предъявляемые к программному обеспечению со стороны архитектуры аппаратных средств E/E/PE системы, связанной с безопасностью, включая значение взаимодействия между программным обеспечением и аппаратурой E/E/PE системы для безопасности управляемого оборудования	PES. Система программного обеспечения	7.4.3	Спецификация требований к безопасности программного обеспечения. Проект архитектуры аппаратных средств E/E/PES (МЭК 61508-2)	Описание проекта архитектуры программного обеспечения. Спецификация проверки интеграции архитектуры программного обеспечения. Спецификация проверки интеграции программного обеспечения и программируемых электронных устройств (как требует МЭК 61508-2)
	Инструментальные средства поддержки и языки программирования: выбрать подходящий набор инструментальных средств, включая языки программирования и компиляторы, для требуемого уровня полноты безопас-	PES. Система программного обеспечения.	7.4.3	Спецификация требований к безопасности программного обеспечения.	Средства разработки и стандарты кодирования.

Продолжение таблицы 1

Стадия жизненного цикла безопасности (номер стадии соответствует номеру блока на рисунке 3)	Задача	Область применения	Номер пункта или раздела	Входные данные	Выходные данные
9.3 Проектирование и разработка программного обеспечения	ности на весь период поддержки безопасности программного обеспечения с использованием верификации, подтверждения соответствия, оценки и модификации	Инструментальные средства поддержки. Языки программирования	7.4.3	Описание проекта архитектуры программного обеспечения	Выбор инструментов разработки
	Детальное проектирование и разработка (проект программной системы): спроектировать и реализовать программное обеспечение, которое удовлетворяет указанным требованиям к безопасности программного обеспечения в отношении необходимого уровня полноты безопасности; программное обеспечение должно быть пригодным к анализу и верификации и поддерживать возможность безопасной модификации	Проектирование архитектуры основных компонентов и подсистем программного обеспечения	7.4.5	Проектное описание архитектуры программного обеспечения. Инструментальные средства поддержки и стандарты кодирования	Спецификация проекта программного обеспечения. Спецификация тестирования интеграции системы программного обеспечения
	Детальное проектирование и разработка (проект отдельных программных модулей): спроектировать и реализовать программное обеспечение, которое удовлетворяет указанным требованиям к безопасности программного обеспечения в отношении необходимого уровня полноты безопасности; программное обеспечение должно быть пригодным к анализу и верификации и поддерживать возможность безопасной модификации	Проект системы программного обеспечения	7.4.5	Спецификация проекта программной системы. Инструментальные средства поддержки и стандарты кодирования	Спецификация проекта программного модуля. Спецификация тестирования программного модуля
	Детальная реализация исходного текста: спроектировать и реализовать программное обеспечение, которое удовлетворяет указанным требованиям к безопасности программного обеспечения в отношении необходимого уровня полноты безопасности; программное обеспечение должно быть пригодным к анализу и верификации и поддерживать возможность безопасной модификации	Отдельные программные модули	7.4.6	Спецификация проекта программного модуля. Инструментальные средства поддержки и стандарты кодирования	Листинг исходного текста. Обзорный отчет по исходному тексту
	Тестирование программных модулей: верифицировать выполнение требований к безопасности программного обеспечения в отношении требуемых функций и уровней полноты безопасности — показать, что каждый программный модуль выполняет предназначенные для него функции и не выполняет непредусмотренных действий	Программные модули	7.4.7	Спецификация тестирования программного модуля. Листинг исходного текста. Обзорный отчет по исходному тексту	Результаты тестирования программного модуля. Верифицированные и проверенные программные модули

Продолжение таблицы 1

Стадия жизненного цикла безопасности (номер стадии соответствует номеру блока на рисунке 3)	Задача	Область применения	Номер пункта или раздела	Входные данные	Выходные данные
9.3 Проектирование и разработка программного обеспечения	Проверка интеграции программного обеспечения: верифицировать выполнение требований к безопасности программного обеспечения в отношении требуемых функций и уровней полноты безопасности — показать, что все программные модули, компоненты и подсистемы корректно выполняют предназначенные для них функции и не выполняют непредусмотренных действий	Архитектура программного обеспечения. Система программного обеспечения	7.4.7	Спецификация тестирования интеграции программной системы	Результаты тестирования интеграции программной системы. Верифицированная и проверенная программная система
9.4 Интеграция программируемых электронных устройств (аппаратура и программное обеспечение)	Интегрировать программное обеспечение с выбранной программируемой электронной аппаратурой. Объединить программное обеспечение и аппаратные средства в связанных с безопасностью программируемых электронных устройствах для того, чтобы удостовериться в их совместимости и выполнении требований к необходимому уровню полноты безопасности	Аппаратное обеспечение программируемой электроники. Интегрированное программное обеспечение	7.5.2	Спецификация тестирования интеграции архитектуры программного обеспечения. Спецификация тестирования интеграции программируемой электроники (МЭК 61508-2). Интегрированная программируемая электроника	Результаты тестирования интеграции архитектуры программного обеспечения. Результаты тестирования интеграции программируемой электроники. Верифицированная и проверенная интегрированная программируемая электроника
9.5 Процедуры, относящиеся к эксплуатации и сопровождению программного обеспечения	Предоставить информацию и процедуры, относящиеся к программному обеспечению и необходимые для того, чтобы гарантировать соблюдение функциональной безопасности E/E/PE систем, связанных с безопасностью, во время эксплуатации и сопровождения	Аппаратное обеспечение программируемой электроники. Интегрированное программное обеспечение	7.6.2	По необходимости вся информация, описанная выше	Процедуры эксплуатации и сопровождения программного обеспечения
9.6 Подтверждение соответствия безопасности программного обеспечения	Обеспечить гарантию, что интегрированная система соответствует указанным требованиям к безопасности программного обеспечения для заданного уровня полноты безопасности	Аппаратное обеспечение программируемой электроники. Интегрированное программное обеспечение	7.7.2	План подтверждения соответствия безопасности программного обеспечения	Результаты подтверждения соответствия безопасности программного обеспечения. Принятое программное обеспечение

Окончание таблицы 1

Стадия жизненного цикла безопасности (номер стадии соответствует номеру блока на рисунке 3)	Задача	Область применения	Номер пункта или раздела	Входные данные	Выходные данные
Модификация программного обеспечения	Внести поправки, улучшения или модификации в принятое программное обеспечение, гарантируя, что требуемый уровень полноты безопасности будет сохранен	Аппаратное обеспечение программируемой электроники. Интегрированное программное обеспечение	7.8.2	Процедуры модификации программного обеспечения. Результаты модификации программного обеспечения	Результаты анализа влияния модификации программного обеспечения. Журнал модификации программного обеспечения
Верификация программного обеспечения	В той степени, в которой этого требует уровень полноты безопасности, протестировать и оценить выходные данные для заданной стадии жизненного цикла программного обеспечения, связанного с безопасностью, для того чтобы гарантировать правильность и совместимость по отношению к выходным данным и стандартам, для этой стадии	Зависит от стадии	7.9.2	План верификации (зависит от стадии)	Отчет по верификации (зависит от стадии)
Оценка функций программного обеспечения	Изучить и представить на обсуждение функциональную безопасность, достигнутую E/E/PE системами, связанными с безопасностью	Стадии 9.1, 9.2, 9.3, 9.4, 9.5 и 9.6, указанные на рисунке 3	8	План оценки функциональной безопасности программного обеспечения	Отчет по оценке функциональной безопасности программного обеспечения

7.2 Спецификация требований к безопасности программного обеспечения

Примечания

1 См. также таблицы А.1 (приложение А) и В.7 (приложение В).

2 Данная стадия представлена на рисунке 3 (см. 9.1).

7.2.1 Цели

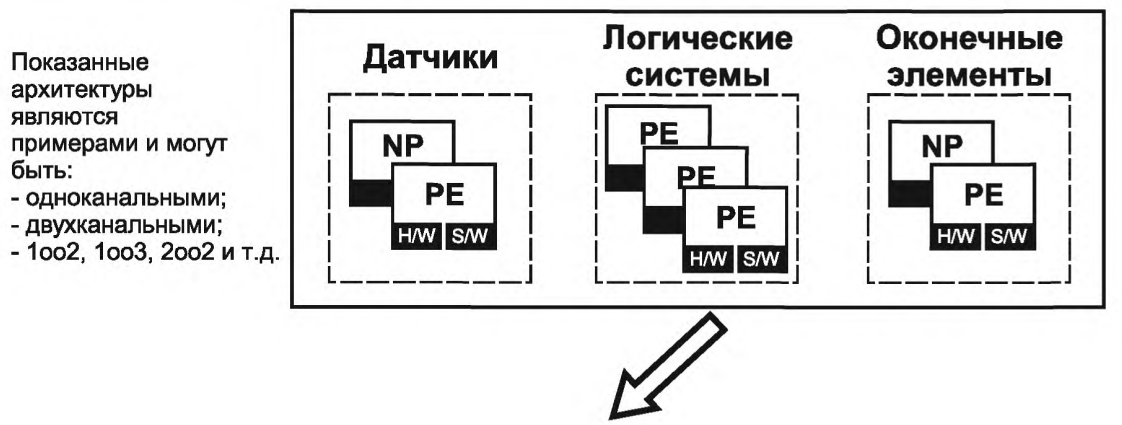
7.2.1.1 Первой целью настоящего подраздела является определение требований к безопасности программного обеспечения как требований к функциям безопасности программного обеспечения и требований к полноте безопасности программного обеспечения.

7.2.1.2 Второй целью настоящего подраздела является определение требований к функциям безопасности каждой E/E/PE системы, которые нужны для реализации этих функций безопасности.

7.2.1.3 Третьей целью настоящего подраздела является определение требований к полноте безопасности для каждой связанной с безопасностью E/E/PE системы, необходимых для достижения уровня полноты безопасности, назначенного каждой функции безопасности, предназначенной для этой E/E/PE системы, связанной с безопасностью.

7.2.2 Требования

Примечание — В большинстве случаев эти требования выполняются комбинацией базового встраиваемого программного обеспечения и программных модулей, которые разработаны специально для конкретного приложения. Именно комбинация этих двух видов программного обеспечения позволяет достигать характеристик, описанных в подразделах, приводимых ниже. Точная граница между базовым и прикладным программным обеспечением зависит от выбранной архитектуры программной системы (см. 7.4.3 и рисунок 6).



Архитектура программируемой электроники		
Архитектура аппаратных средств PE	Архитектура программного обеспечения PE	
Базовые и прикладные аппаратные средства PE	Встроенное программное обеспечение PE	Прикладное программное обеспечение PE
Примеры: - диагностические тесты, - избыточные процессоры, - сдвоенные платы ввода/вывода.	Примеры: - коммуникационные драйверы, - обработка отказов, - управляющие программы.	Примеры: - функции ввода/вывода, - производные функции (например, проверка датчиков, если она не предоставляется как утилита или встроенная программа).

PE — программируемая электроника; NP — непрограммируемые устройства; H/W — аппаратные средства; S/W — программное обеспечение; MooN — M из N (например 1oo2 представляет собой 1 из 2).

Рисунок 6 — Взаимосвязь между архитектурой аппаратного и программного обеспечения программируемой электроники

7.2.2.1 Если требования к безопасности программного обеспечения уже были определены в требованиях к E/E/PE системам, связанным с безопасностью (МЭК 61508-2, пункт 7.2), повторять их не требуется.

7.2.2.2 Спецификация требований к безопасности программного обеспечения должна быть выработана на основе требований к безопасности E/E/PE систем, связанных с безопасностью (МЭК 61508-2), и требований к планированию безопасности (см. раздел 6). Эта информация должна быть доступна для разработчика программного обеспечения.

Примечание — Это требование означает, что должно быть тесное взаимодействие между разработчиком E/E/PES и разработчиком программного обеспечения (МЭК 61508-2 и МЭК 61508-3). По мере того, как требования к безопасности и архитектура программного обеспечения (см. 7.4.3) становятся более определенными, может проявляться влияние на архитектуру аппаратных средств E/E/PES, и по этой причине становится важным тесное взаимодействие между разработчиками аппаратных средств и программного обеспечения (см. рисунок 4).

7.2.2.3 Спецификация требований к безопасности программного обеспечения должна быть достаточно подробной для того, чтобы обеспечить стадии проектирования и внедрения информацией для обеспечения требуемой полноты безопасности и позволить выполнить оценку функциональной безопасности.

Примечание — Уровень детальности спецификации может изменяться в зависимости от сложности приложения.

7.2.2.4 Разработчик программного обеспечения должен просмотреть информацию, содержащуюся в 7.2.2.2, для того чтобы гарантировать, что требования определены адекватным образом. В частности, разработчик программного обеспечения должен учесть следующее:

a) функции безопасности;

b) конфигурацию или архитектуру системы;

c) требования к полноте безопасности аппаратных средств (программируемой электроники, датчиков и устройств привода);

d) требования к полноте безопасности программного обеспечения;

e) производительность и время отклика;

f) интерфейсы оборудования и оператора.

7.2.2.5 Разработчик программного обеспечения должен установить процедуры для устранения разногласий при назначении уровня полноты безопасности программного обеспечения.

7.2.2.6 В той степени, в которой этого требует уровень полноты безопасности, требования к безопасности программного обеспечения должны быть выражены и структурированы так, чтобы они:

a) были ясными, точными, недвусмысленными, пригодными для верификации, тестирования, поддержки и выполнения и соразмерными с уровнем полноты безопасности;

b) были пригодными для того, чтобы можно было определить их источник в спецификации требований к безопасности E/E/PE систем;

c) не содержали информации и описаний, которые являются двусмысленными и/или могут быть не поняты теми, кто использует документ на той или иной стадии жизненного цикла систем безопасности.

7.2.2.7 В требованиях к безопасности программного обеспечения должны быть подробно описаны все соответствующие режимы работы EUC, если только они не были уже адекватно определены в требованиях к безопасности E/E/PE систем, связанных с безопасностью.

Примечание — В большинстве случаев это требование будет достигаться объединением общего встроенного и специального прикладного программного обеспечения. Именно от этого объединения требуется обеспечение характеристик, удовлетворяющих требованиям к безопасности. Точное разделение между общим и специальным прикладным программным обеспечением зависит от выбранной архитектуры программного обеспечения (см. 7.4.3 и рисунок 4).

7.2.2.8 Спецификация требований к безопасности программного обеспечения должна определять и документировать все относящиеся к безопасности и иные необходимые ограничения, связанные с взаимодействием между аппаратными средствами и программным обеспечением.

7.2.2.9 В той степени, в которой этого требует описание проекта архитектуры аппаратных средств E/E/PE систем, спецификация требований к безопасности программного обеспечения должна учитывать следующее:

a) самоконтроль программного обеспечения (например, МЭК 61508-7, пункты C.2.5 и C.3.10 приложения C);

b) мониторинг программируемой электронной аппаратуры, датчиков и устройств привода;

c) периодическое тестирование функций безопасности во время выполнения программы;

d) разрешение тестирования функций безопасности во время работы EUC.

7.2.2.10 Если E/E/PE система, связанная с безопасностью, должна выполнять функции, не относящиеся к безопасности, эти функции должны быть четко указаны в спецификации требований к безопасности программного обеспечения.

7.2.2.11 Спецификация требований к безопасности программного обеспечения должна выражать необходимые характеристики безопасности продукта, а не проекта. С учетом 7.2.2.1 — 7.2.2.10, в зависимости от конкретных обстоятельств, должны быть определены следующие положения:

a) требования к функциям безопасности программного обеспечения:
- функции, которые позволяют EUC достигать или поддерживать безопасное состояние,
- функции, связанные с обнаружением, оповещением и обработкой ошибок аппаратных средств программируемой электроники,
- функции, связанные с обнаружением, оповещением и обработкой ошибок датчиков и устройств привода,
- функции, связанные с обнаружением, оповещением и обработкой ошибок в самом программном обеспечении (самоконтроль программного обеспечения),
- функции, связанные с периодическим тестированием функций в режиме реального времени,
- функции, связанные с периодическим тестированием функций в автономном режиме,
- функции, обеспечивающие безопасную модификацию PES,
- интерфейсы функций, не связанных с безопасностью,
- производительность и время отклика,
- интерфейсы между программным обеспечением и PES.

Примечание — Интерфейсы должны включать средства онлайнового и автономного программирования;

b) требования к полноте безопасности программного обеспечения:
- уровни полноты безопасности для каждой функции перечисления a).

Примечание — Назначение полноты безопасности компонентам программного обеспечения описано в МЭК 61508-5, приложение А.

7.3 Планирование подтверждения соответствия безопасности программного обеспечения

Примечание — Эта стадия представлена на рисунке 3 (см. 9.2).

7.3.1 Цели

Целью требований настоящего подраздела является разработка плана подтверждения соответствия безопасности программного обеспечения.

7.3.2 Требования

7.3.2.1 В ходе планирования должны быть определены процедурные и технические шаги, которые нужно использовать для того, чтобы продемонстрировать, что программное обеспечение удовлетворяет требованиям безопасности (см. 7.2).

7.3.2.2 План подтверждения соответствия безопасности программного обеспечения должен содержать следующие положения:

a) описание этапов подтверждения соответствия;

b) перечень лиц, осуществляющих подтверждение соответствия;

c) идентификацию соответствующих режимов работы EUC, включая:
- подготовку к использованию, а также установку и настройку,
- работу в режиме запуска и обучения, в автоматическом, ручном, полуавтоматическом и стационарном режимах,
- переустановку, выключение, сопровождение,
- предполагаемые ненормальные режимы;

d) идентификацию программного обеспечения, связанного с безопасностью, для которого должна быть проведена процедура подтверждения соответствия, для каждого режима работы EUC до момента его ввода в эксплуатацию;

e) техническую стратегию для подтверждения соответствия (например, аналитические методы, статистическое тестирование и т.п.) (см. 7.3.2.3);

f) средства (методы) и процедуры в соответствии с перечислением e), которые должны быть использованы для подтверждения соответствия каждой функции требованиям к функциям безопасности программного обеспечения (см. 7.2) и требованиям к полноте безопасности программного обеспечения (см. 7.2);

g) конкретные ссылки на требования к безопасности программного обеспечения (см. 7.2);

h) условия, в которых должны происходить процедуры подтверждения соответствия (например, при тестировании может потребоваться использование калиброванных инструментов и оборудования);

i) критерии прохождения/непрохождения подтверждения соответствия (см. 7.3.2.5);

j) политику и процедуры, используемые для оценки результатов подтверждения соответствия, в частности, при оценке отказов.

Примечание — Эти требования основаны на общих требованиях МЭК 61508-1, пункт 7.8.

7.3.2.3 Техническая стратегия для подтверждения соответствия программного обеспечения, связанного с безопасностью (см. таблицу А.7, приложение А), должна содержать следующую информацию:

a) выбор ручных или автоматических методов, или и тех и других;

b) выбор статических или динамических методов, или и тех и других;

c) выбор аналитических или статистических методов, или и тех и других.

7.3.2.4 В рамках процедуры подтверждения соответствия программного обеспечения, связанного с безопасностью, если этого требует уровень полноты безопасности (МЭК 61508-1, пункт 8.2.12), область применения и содержание планирования подтверждения соответствия безопасности программного обеспечения должны быть изучены экспертом или третьей стороной, представляющей эксперта. Эта процедура должна также включать заявление о присутствии эксперта при испытаниях.

7.3.2.5 Критерии прохождения/непрохождения при завершении подтверждения соответствия программного обеспечения должны включать:

a) необходимые входные сигналы, включая их последовательность и значения;

b) предполагаемые выходные сигналы, включая их последовательность и значения и

c) другие критерии приемки, например использование памяти, хронометраж, допустимые интервалы для значений.

7.4 Проектирование и разработка программного обеспечения

Примечание — Эта фаза представлена на рисунке 3 (см. 9.3).

7.4.1 Цели

7.4.1.1 Первой целью требований настоящего подраздела является создание такой архитектуры программного обеспечения, которая удовлетворяла бы требованиям к безопасности программного обеспечения (см. 7.2) в отношении необходимого уровня полноты безопасности.

7.4.1.2 Второй целью требований настоящего подраздела является анализ и оценка требований, предъявляемых к программному обеспечению со стороны аппаратных средств E/E/PE систем, связанных с безопасностью, включая значение взаимодействия между аппаратными средствами и программным обеспечением E/E/PE систем для безопасности управляемого оборудования.

7.4.1.3 Третьей целью требований настоящего подраздела является выбор подходящего набора инструментальных средств, включая языки программирования и компиляторы, который соответствовал бы заданному уровню полноты безопасности на протяжении всего жизненного цикла систем безопасности и способствовал бы выполнению процессов верификации, подтверждения соответствия, оценки и модификации.

7.4.1.4 Четвертой целью требований настоящего подраздела является проектирование и реализация программного обеспечения, которое удовлетворяло бы специфицированным требованиям к безопасности программного обеспечения (см. 7.2) в отношении необходимого уровня полноты безопасности. Это программное обеспечение должно быть пригодным для анализа и верификации и обладать способностью к безопасной модификации.

7.4.1.5 Пятой целью требований настоящего подраздела является проверка выполнения требований к безопасности программного обеспечения (в отношении необходимых функций и уровня полноты безопасности программного обеспечения).

7.4.2 Общие требования

7.4.2.1 В зависимости от характера процесса разработки программного обеспечения ответственность за соответствие 7.4 может лежать только на поставщике, только на пользователе или на обеих сторонах. Распределение ответственности должно быть определено во время планирования безопасности (см. раздел 6).

7.4.2.2 В соответствии с требуемым уровнем полноты безопасности выбранный метод проектирования должен обладать характеристиками, которые облегчают:

a) абстракцию, разделение на модули и другие характеристики, контролирующие уровень сложности;

b) выражение:

- выполняемых функций,

- обмена данными между компонентами,
- информации, относящейся к последовательности и времени выполнения,
- ограничений на время выполнения,
- параллельного выполнения,
- структур данных и их свойств,
- проектных предположений и их зависимостей;

c) понимание разработчиками и другими лицами, которые должны иметь дело с проектом;

d) верификацию и оценку соответствия.

Примечание — См. также таблицы А.1 — А.10 (приложение А) и таблицы В.1 — В.10 (приложение В).

7.4.2.3 Тестируемость и способность к безопасной модификации должны быть предусмотрены на этапе проектирования для того, чтобы облегчить реализацию этих характеристик в окончательной версии системы, связанной с безопасностью.

Примечание — Примеры включают в себя эксплуатационные режимы в машиностроении и на обрабатывающих предприятиях.

7.4.2.4 Выбранный метод проектирования должен обладать характеристиками, которые облегчают модификацию программного обеспечения. К числу таких характеристик относят модульность, скрытие информации и инкапсуляцию.

7.4.2.5 Представление проекта должно основываться на нотации, которая является однозначно определенной или ограничена до однозначно определенных свойств.

7.4.2.6 Проект должен минимизировать настолько, насколько это возможно, ту часть программного обеспечения, которая относится к безопасности.

7.4.2.7 Если программное обеспечение должно реализовать функции как относящиеся, так и не относящиеся к безопасности, оно в целом должно рассматриваться как относящееся к безопасности, если только в проекте не продемонстрирована достаточная независимость между этими функциями.

7.4.2.8 Если программное обеспечение должно реализовать функции безопасности, имеющие различный уровень полноты безопасности, то следует считать, что все программное обеспечение имеет наивысший среди этих уровней, если только в проекте не будет продемонстрирована достаточная независимость функций, имеющих различный уровень полноты безопасности. Обоснование независимости должно быть документировано.

Примечание — Уровень полноты безопасности программного обеспечения должен быть не ниже чем уровень полноты функции безопасности, к которой оно относится. Однако уровень полноты безопасности компонента программного обеспечения может быть ниже, чем уровень полноты функции безопасности, к которой он относится, если этот компонент используется в сочетании с другими аппаратными компонентами, такими, что уровень полноты безопасности сочетания компонентов, по меньшей мере, равен уровню полноты безопасности функции безопасности.

7.4.2.9 В той степени, насколько это возможно, проект должен включать функции, выполняющие проверки и все диагностические тесты для того, чтобы обеспечить соблюдение требований к полноте безопасности E/E/PE системы, связанной с безопасностью (как установлено в МЭК 61508-2).

7.4.2.10 Проект программного обеспечения должен включать соразмерно требуемому уровню полноты безопасности средства самоконтроля потоков управления и потоков данных. При обнаружении ошибки должны быть выполнены соответствующие действия [см. таблицы А.2 и А.4 (приложение А)].

7.4.2.11 Если стандартное или ранее разработанное программное обеспечение должны использоваться как часть проекта [см. таблицы А.3 и А.4 (приложение А)], они должны быть четко идентифицированы. Способность программного обеспечения удовлетворять требованиям спецификации по отношению к безопасности программных систем (см. 7.2) должна быть обоснована. Эта способность должна основываться на данных по удовлетворительной работе в схожем приложении или быть предметом тех же самых процедур верификации и подтверждения соответствия, которые подразумеваются для любого вновь разрабатываемого программного обеспечения. Следует оценить ограничения, связанные с условиями, в которых работало программное обеспечение (например, зависимость от операционной системы и компилятора).

Примечание — Такое обоснование может быть разработано при планировании безопасности (см. раздел 6).

7.4.2.12 Настоящий подраздел (7.4) должен в той мере, насколько это возможно, применяться к данным, включая любой язык генерации данных.

7.4.3 Требования к архитектуре программного обеспечения

Примечания

1 См. также таблицы А.2 (приложение А) и В.7 (приложение В).

2 Архитектура программного обеспечения определяет основные компоненты и подсистемы программного обеспечения, их взаимосвязь, способ реализации необходимых характеристик и, в частности, полноты безопасности. Примеры основных компонентов программного обеспечения включают операционные системы, базы данных, подсистемы ввода и вывода производственных данных, коммуникационные подсистемы, прикладные программы, инструментальные средства программирования и диагностики и т.п.

3 В некоторых отраслях промышленности архитектура программного обеспечения может называться описанием функций или спецификацией функций проекта (хотя эти документы могут также включать вопросы, относящиеся к аппаратным средствам).

4 Для пользовательского прикладного программирования в языках с ограниченной изменчивостью, в частности в языках, используемых в ПЛК (МЭК 61508-6, приложение Е), архитектура определяется поставщиком как стандартная характеристика ПЛК. Однако в рамках настоящего стандарта к поставщику может быть предъявлено требование, гарантировать пользователю соответствие поставляемого продукта требованиям 7.4. Пользователь приспосабливает ПЛК, используя стандартные возможности программирования, например многозвенные логические схемы. Требования 7.4.3 — 7.4.8 сохраняют свою силу. Требование определения и документирования архитектуры может рассматриваться как информация, которую пользователь может использовать при выборе ПЛК (или эквивалентного ему устройства) для приложения.

5 В другом крайнем случае, в некоторых встроенных приложениях, использующих язык с полной изменчивостью, например в машине, управляемой микропроцессором, архитектура должна создаваться поставщиком специально для приложения (или класса приложений). Пользователь обычно не имеет инструментария для программирования. В этих условиях ответственность за обеспечение соответствия 7.4 ложится на поставщика.

6 Имеются системы, попадающие в промежуток между типами, упомянутыми в примечаниях 4 и 5, в таких случаях ответственность за соответствие разделяется между пользователем и поставщиком.

7 С точки зрения безопасности стадия разработки архитектуры программного обеспечения соответствует периоду, когда разрабатывается базовая стратегия безопасности для программного обеспечения.

7.4.3.1 В зависимости от характера разработки программного обеспечения ответственность за соответствие 7.4.3 может лежать только на поставщике, только на разработчике или на обеих сторонах (см. примечания, приведенные выше). Распределение ответственности должно быть документировано во время планирования безопасности (см. раздел 6).

7.4.3.2 Предлагаемый проект архитектуры программного обеспечения должен быть создан поставщиком программного обеспечения и/или разработчиком, описание архитектуры должно быть подробным. Описание должно:

a) содержать выбор и обоснование интегрированного набора методов и мероприятий, которые будут необходимы в течение жизненного цикла модулей безопасности для того, чтобы удовлетворить требования к безопасности программного обеспечения на заданном уровне полноты безопасности (см. 7.2). Эти методы и мероприятия включают стратегию проектирования программного обеспечения для обеспечения устойчивости к отказам (совместимую с аппаратными средствами) и для избежания отказов, в том числе (при необходимости) избыточность и разнообразие;

b) основываться на разделении на компоненты/подсистемы, для каждой из которых должна предоставляться следующая информация:

- являются ли они вновь разработанными, уже существующими или находящимися в частной собственности,
- проводилась ли верификация и если проводилась, то при каких условиях,
- связан ли каждый из этих компонентов/подсистем с безопасностью или нет,
- уровень полноты безопасности для компонента/подсистемы;

c) определять все взаимодействия между программными средствами и аппаратным обеспечением, а также оценивать и детализировать их значение;

d) использовать для представления архитектуры нотацию, которая является однозначно определенной или ограничена до подмножества однозначно определенных характеристик;

e) содержать набор проектных характеристик, которые должны использоваться для поддержания полноты безопасности всех данных. В число таких данных допускается включать входные и выходные производственные данные, коммуникационные данные, данные интерфейса оператора, данные сопровождения и данные, хранящиеся во внутренних базах данных;

f) определять тесты интеграции архитектуры программного обеспечения для того, чтобы обеспечить выполнение спецификации требований к безопасности программного обеспечения на заданном уровне полноты безопасности (см. 7.2).

7.4.3.3 Любые изменения, которые может потребоваться внести в специфицированные требования к E/E/PE системе, связанной с безопасностью, после использования мероприятий 7.4.3.2 должны быть согласованы с разработчиком E/E/PE систем и документированы.

Примечание — Итерационное взаимодействие между архитектурой аппаратных средств и программного обеспечения является неизбежным (см. рисунок 5), поэтому существует необходимость в обсуждении с разработчиком аппаратуры таких вопросов, как спецификация тестирования интеграции программируемой электронной аппаратуры и программного обеспечения (см. 7.5).

7.4.4 Требования к инструментальным средствам поддержки и языкам программирования

Примечания

1 См. также таблицу А.3 (приложение А).

2 Выбор инструментов для разработки зависит от характера процессов разработки программного обеспечения и его архитектуры (см. 7.4.3).

Если пользовательское прикладное программирование выполняется на языке с ограниченной варьируемостью при низких уровнях полноты безопасности, то круг необходимых инструментов и языков программирования может быть ограничен стандартными языками ПЛК, редакторами и загрузчиками. Ответственность за соответствие 7.4.4 будет лежать, следовательно, главным образом на поставщике.

При более высоких уровнях полноты безопасности могут потребоваться ограниченные подмножества языка ПЛК, а также средства верификации и отладки, такие как анализаторы кода и имитаторы. В этих условиях ответственность лежит и на поставщике, и на пользователе.

Инструментарий для встраиваемых приложений, использующий языки с полной варьируемостью, должен быть более разнообразным даже в случае низких уровней полноты безопасности. Ответственность за соответствие 7.4.4 будет лежать, главным образом, на разработчиках программного обеспечения. В их число входит поставщик ПЛК, который может использовать языки с полной варьируемостью в языке с низким уровнем варьируемости для обеспечения пользовательского прикладного программирования.

7.4.4.1 В зависимости от характера программного обеспечения ответственность за соответствие 7.4.4 может лежать только на поставщике, только на пользователе или на обеих сторонах (см. 7.4.4, примечание 2). Разделение ответственности должно быть документировано во время планирования безопасности (см. раздел 6).

7.4.4.2 Должен быть выбран набор интегрированных инструментальных средств, соответствующий требуемому уровню полноты безопасности и включающий языки программирования, компиляторы, средства управления конфигурацией и, при необходимости, автоматизированные средства тестирования. Следует учитывать способность инструментальных средств (необязательно тех, которые использовались при первоначальной разработке системы) выполнять необходимые задачи на протяжении всего жизненного цикла E/E/PE систем, связанных с безопасностью.

7.4.4.3 В той степени, в которой этого требует уровень полноты безопасности, выбранные языки программирования должны:

a) иметь транслятор/компилятор, который либо обладает сертификатом, подтверждающим соответствие национальному или международному стандарту, либо должен быть оценен для проверки его пригодности;

b) быть полностью и однозначно определенными либо ограниченными до подмножества однозначно определяемых элементов;

c) соответствовать характеристикам приложения;

d) обладать свойствами, облегчающими обнаружение ошибок программирования;

e) поддерживать характеристики, соответствующие методу проектирования.

7.4.4.4 Если требования 7.4.4.3 не могут быть выполнены, то при описании проекта архитектуры программного обеспечения (см. 7.4.3) следует документировать обоснование использования альтернативного языка программирования. В обосновании должны быть подробно рассмотрены пригодность языка программирования, а также дополнительные мероприятия, относящиеся к известным недостаткам языка.

7.4.4.5 Стандарты составления программ должны быть:

a) рассмотрены экспертом на предмет определения пригодности и

b) использованы при разработке всего программного обеспечения, связанного с безопасностью.

7.4.4.6 Стандарты составления программ должны определять правильные методы программирования, запрещать использование небезопасных возможностей языка (например, неопределенных особенностей языка, неструктурированных конструкций и т.п.) и определять процедуры для

документирования исходного текста. Документация, относящаяся к исходному тексту, должна содержать, по меньшей мере, следующую информацию:

a) юридическое лицо (например, компания, авторы и т.п.);

b) описание;

c) входные и выходные данные;

d) историю изменения конфигурации.

7.4.5 Требования к детальному проектированию и разработке

Примечания

1 См. также таблицы А.4 (приложение А), В.1, В.7 и В.9 (приложение В).

2 Под детальным проектированием здесь понимается разделение основных компонентов архитектуры на систему программных модулей, проектирование отдельных программных модулей и их программирование. В небольших приложениях проектирование программных систем и архитектуры могут быть объединены.

3 Характер детального проектирования и разработки может изменяться в зависимости от характера процессов разработки программ и архитектуры программного обеспечения (см. 7.4.3). Когда прикладное программирование выполняет пользователь, использующий языки с ограниченной варьируемостью, например языки многозвенных логических схем и языки функциональных блоков, детальное проектирование может рассматриваться скорее как конфигурирование, чем как программирование. Тем не менее, хороший стиль программирования состоит в структурировании программного обеспечения, включая организацию модульной структуры, которая выделяет (настолько, насколько это возможно) блоки, связанные с безопасностью; в использовании проверок на попадание в интервал допустимых значений и других возможностей защиты от ошибок при вводе исходных данных; в использовании ранее верифицированных программных модулей; в применении конструкций, которые облегчают выполнение будущих модификаций программного обеспечения.

7.4.5.1 В зависимости от характера программного обеспечения ответственность за соответствие 7.4.5 может лежать только на поставщике, только на пользователе или на обеих сторонах (см. примечание 3). Разделение ответственности должно быть документировано во время планирования безопасности (см. раздел 6).

7.4.5.2 До начала детального проектирования должна быть следующая информация:

- спецификация требований к безопасности программного обеспечения (см. 7.2);
- описание проекта архитектуры (см. 7.4.3);
- план проверки безопасности (см. 7.3).

7.4.5.3 Программное обеспечение следует разрабатывать таким образом, чтобы достигалась модульность, тестируемость и способность к безопасной модификации.

7.4.5.4 Дальнейшее уточнение проекта для каждого главного компонента/подсистемы в описании проекта архитектуры программного обеспечения (см. 7.4.3) должно основываться на разделении на программные модули (т.е. на спецификации конструкции программной системы). Необходимо определить конструкцию каждого программного модуля и проверки, которые должны использоваться для этих модулей.

Примечание — Для стандартных или ранее разработанных компонентов программных модулей не требуется проекта или спецификации тестирования, если может быть показано, что они удовлетворяют требованиям 7.4.2.11.

7.4.5.5 Должны быть определены соответствующие проверки интеграции программных систем, показывающие, что программные системы удовлетворяют требованиям к безопасности программного обеспечения для заданного уровня полноты безопасности (см. 7.2).

7.4.6 Требования к реализации исходных текстов программ

Примечание — См. также таблицы А.4 (приложение А), В.1, В.7 и В.9 (приложение В).

7.4.6.1 Исходные тексты программ должны:

a) быть читаемыми, понятными и пригодными к проверке;

b) удовлетворять специфицированным требованиям к конструкции программного модуля (см. 7.4.5);

c) удовлетворять специфицированным требованиям к стандартам составления программ (см. 7.4.4);

d) удовлетворять всем требованиям, определенным при планировании безопасности (см. раздел 6).

7.4.6.2 Каждый модуль программного обеспечения должен быть просмотрен.

Примечание — Просмотр кода относится к процессам верификации (см. 7.9).

7.4.7 Требования к тестированию программных модулей

Примечания

1 См. также таблицы А.5 (приложения А), В.2, В.3 и В.6 (приложения В).

2 Процесс проверки того, что программный модуль корректно выполняет все требования, содержащиеся в спецификации тестирования, относится к процессам верификации (см. 7.9). Сочетание просмотра исходных текстов и тестирования программных модулей дает гарантию того, что программный модуль удовлетворяет требованиям своей спецификации, т.е. верифицирует модуль.

7.4.7.1 Каждый программный модуль должен быть протестирован в соответствии со спецификацией, разработанной при проектировании программного обеспечения (см. 7.4.5).

7.4.7.2 Эти проверки должны продемонстрировать, что каждый программный модуль выполняет функции, для которых он предназначен, и не выполняет функции, которые не были для него предусмотрены.

Примечания

1 Сказанное выше не означает тестирования всех комбинаций входных данных и всех комбинаций выходных данных. Достаточным может быть тестирование всех классов эквивалентности (МЭК 61508-7, пункт С.5.7 приложения С) или структурное тестирование (МЭК 61508-7, пункт С.5.8 приложения С). Анализ граничных значений (МЭК 61508-7, пункт С.5.4 приложения С), анализ управляющей логики (МЭК 61508-7, пункт С.5.9 приложения С) или анализ скрытых путей выполнения программы (МЭК 61508-7, пункт С.5.11 приложения С) могут уменьшить количество проверок до приемлемого уровня. Программы, пригодные для анализа (МЭК 61508-7, пункт С.2.7 приложения С), могут позволить достичь более быстрого выполнения требований.

2 Если при разработке используются формальные методы (МЭК 61508-7, пункт С.2.4 приложения С), формальные доказательства (МЭК 61508-7, пункт С.5.13 приложения С) или операторы проверки условий (МЭК 61508-7, пункт С.3.3 приложения С), область применения подобных проверок может быть уменьшена.

3 Допускается использовать также статистические данные (МЭК 61508-7, приложение D).

7.4.7.3 Результаты тестирования программных модулей должны быть документированы.

7.4.7.4 Должны быть определены процедуры для коррекции при непрохождении теста.

7.4.8 Требования к тестированию интеграции программного обеспечения

Примечания

1 См. также таблицы А.5 (приложение А), В.2, В.3 и В.6 (приложение В).

2 Проверка того, что интеграция программного обеспечения является корректной, относится к процессам верификации (см. 7.9).

7.4.8.1 Проверки интеграции программного обеспечения должны разрабатываться на этапе проектирования и разработки.

7.4.8.2 Проверки интеграции программного обеспечения должны определять следующее:

a) разделение программного обеспечения на контролируемые интегрируемые подмножества;

b) контрольные примеры и контрольные данные;

c) типы проверок, которые должны быть выполнены;

d) условия тестирования, используемые инструменты, конфигурацию и программы;

e) условия, при которых проверка считается выполненной, и

f) процедуры, которые необходимо выполнить, если проверка дала отрицательный результат.

7.4.8.3 Программное обеспечение должно быть проверено в соответствии с заранее определенными тестами интеграции программ. Эти тесты должны продемонстрировать, что все программные модули и программные компоненты/подсистемы корректно взаимодействуют для выполнения функций, для которых они предназначены, и не выполняют непредусмотренных функций.

Примечания

1 Сказанное выше не означает тестирования всех комбинаций входных данных и всех комбинаций выходных данных. Достаточным может быть тестирование всех классов эквивалентности (МЭК 61508-7, пункт С.5.7 приложения С) или структурное тестирование (МЭК 61508-7, пункт С.5.8 приложения С). Анализ граничных значений (МЭК 61508-7, пункт С.5.4 приложения С), анализ управляющей логики (МЭК 61508-7, пункт С.5.9 приложения С) или анализ скрытых путей выполнения программы (МЭК 61508-7, пункт С.5.11 приложения С) могут уменьшить количество проверок до приемлемого уровня. Если выполняемая разработка ведет к созданию программ, пригодных для анализа (МЭК 61508-7, пункт С.2.7 приложения С), то можно достичь более быстрого выполнения требований.

2 Если при разработке используются формальные методы (МЭК 61508-7, пункт С.2.4 приложения С), формальные доказательства (МЭК 61508-7, пункт С.5.13 приложения С) или операторы проверки условий (МЭК 61508-7, пункт С.3.3 приложения С), область применения подобных проверок может быть уменьшена.

3 Допускается использовать также статистические данные (МЭК 61508-7, приложение D).

7.4.8.4 Результаты проверки интеграции программного обеспечения должны быть документированы; в документации должны быть сформулированы результаты проверки и должно быть указано, были ли выполнены цели и критерии проверки. Если тестирование окончилось неудачно, должны быть описаны причины этого.

7.4.8.5 При интеграции программного обеспечения все модификации или изменения должны быть объектом анализа влияния, который должен определить, какие программные модули затрагиваются изменениями, и установить необходимость повторной верификации и проектирования.

7.5 Интеграция программируемой электроники (аппаратные средства и программное обеспечение)

Примечания

1 См. также таблицы А.6 (приложение А), В.3 и В.6 (приложение В).

2 Эта стадия представлена на рисунке 3 (см. 9.4).

7.5.1 Цели

7.5.1.1 Первой целью требований настоящего подраздела является интеграция программного обеспечения с используемой программируемой электронной аппаратурой.

7.5.1.2 Второй целью требований настоящего подраздела является объединение программного обеспечения и аппаратных средств в программируемый электронный комплекс, связанный с безопасностью, проверка их совместимости и выполнения требований назначенного уровня полноты безопасности.

Примечания

1 Проверка корректности интеграции программного обеспечения с аппаратными средствами программируемой электроники относится к процессам верификации (см. 7.9).

2 В зависимости от характера приложения эти проверки могут быть объединены с проверками, описываемыми в 7.4.8.

7.5.2 Требования

7.5.2.1 Проверки интеграции должны быть определены на этапе проектирования и разработки, их целью является проверка совместимости программного обеспечения и аппаратных средств в программируемом электронном устройстве, связанном с безопасностью.

Примечание — При разработке проверок интеграции может потребоваться тесная кооперация с разработчиком E/E/PES систем.

7.5.2.2 Тесты интеграции для программируемой электроники (аппаратные средства и программное обеспечение) должны определять следующее:

a) разбиение системы на уровни интеграции;

b) тестовые примеры и тестовые данные;

c) типы выполняемых проверок;

d) условия тестирования, используемые инструменты, конфигурацию и программы;

e) условия, при которых проверка считается выполненной.

7.5.2.3 Тесты интеграции программируемой электроники (аппаратные средства и программное обеспечение) должны различать операции, которые выполняются разработчиком на его оборудовании, и операции, требующие доступа к пользовательскому оборудованию.

7.5.2.4 Тесты интеграции программируемой электроники (аппаратные средства и программное обеспечение) должны различать следующие процессы:

a) включение программного обеспечения в целевое программируемое электронное оборудование;

b) интеграцию E/E/PE систем, т.е. добавление интерфейсов, таких как датчики и устройства привода;

c) полную интеграцию EUC и E/E/PE систем, связанных с безопасностью.

Примечание — Перечисления b) и c) охватываются МЭК 61508-1 и МЭК 61508-2, они включены в контекст перечисления a) для полноты.

7.5.2.5 Программное обеспечение должно быть интегрировано с программируемой электронной аппаратурой, связанной с безопасностью, в соответствии со специфицированными тестами интеграции для программируемой электроники (аппаратные средства и программное обеспечение).

7.5.2.6 При тестировании интеграции программируемой электроники, связанной с безопасностью (аппаратных средств и программного обеспечения), все модификации или изменения должны быть

объектом анализа влияния, который должен определить, какие программные модули затрагиваются изменениями, и установить необходимость повторной верификации.

7.5.2.7 Тестовые примеры и результаты их выполнения должны быть документированы для последующего анализа.

7.5.2.8 Результаты проверки интеграции программируемой электроники (аппаратных средств и программного обеспечения) должны быть документированы, в документации должны быть сформулированы результаты проверки, а также указано, были ли выполнены цели и критерии проверки. Если тестирование окончилось неудачно, должны быть описаны причины этого. Все модификации или изменения, являющиеся результатом тестирования, должны быть объектом анализа влияния, который должен определить, какие программные модули затрагиваются изменениями, и установить необходимость повторной верификации и проектирования.

7.6 Работа программного обеспечения и процедуры модификации

Примечания

1 См. также таблицу А.8 (приложение А).

2 Эта стадия представлена на рисунке 3 (см. 9.5).

7.6.1 Цели

Целью требований настоящего подраздела является представление информации и процедур, касающихся программного обеспечения, необходимых для того, чтобы убедиться в том, что функциональная безопасность E/E/PE систем, связанных с безопасностью, сохраняется при работе и модификациях.

7.6.2 Требования

Требования приведены в МЭК 61508-2, пункт 7.6 и в 7.8 настоящего стандарта.

Примечание — В настоящем стандарте программное обеспечение (в отличие от аппаратных средств) не может поддерживаться, оно всегда модифицируется.

7.7 Подтверждение соответствия безопасности программного обеспечения

Примечания

1 См. также таблицы А.7 (приложение А), В.3 и В.5 (приложение В).

2 Данная стадия представлена на рисунке 3 (см. 9.6).

7.7.1 Цели

Цель требований настоящего подраздела — гарантировать соответствие интегрированной системы специфицированным требованиям к безопасности программного обеспечения (см. 7.2) на заданном уровне полноты безопасности.

7.7.2 Требования

7.7.2.1 Если соответствие требованиям к безопасности программного обеспечения уже было установлено для E/E/PE системы, связанной с безопасностью (МЭК 61508-2, пункт 7.7), проводить повторное подтверждение соответствия не требуется.

7.7.2.2 Операции подтверждения соответствия должны выполняться в соответствии со спецификациями, разработанными при планировании подтверждения соответствия безопасности программного обеспечения (см. 7.3).

7.7.2.3 Результаты подтверждения соответствия безопасности программного обеспечения должны быть документированы.

7.7.2.4 При проведении подтверждения соответствия безопасности программного обеспечения для каждой функции безопасности должны быть документированы следующие результаты:

a) хронологический перечень операций подтверждения соответствия;

b) используемая версия плана подтверждения соответствия безопасности программного обеспечения (см. 7.3);

c) подтверждаемые функции безопасности (с использованием тестирования или анализа), со ссылками на план подтверждения соответствия безопасности программного обеспечения (см. 7.3);

d) использованные инструменты и оборудование, а также данные калибровки;

e) результаты операций подтверждения соответствия;

f) расхождения между ожидаемыми и фактическими результатами.

7.7.2.5 При наличии расхождений между ожидаемыми и фактическими результатами проводится анализ и принимается решение, продолжать ли проверку или подготовить запрос на изменение и вернуться к более ранней стадии жизненного цикла разработки. Это решение должно быть документировано как часть результатов подтверждения соответствия безопасности программного обеспечения.

Примечание — Требования 7.7.2.2 — 7.7.2.5 основываются на общих требованиях МЭК 61508-1 (пункт 7.14).

7.7.2.6 Подтверждение соответствия программного обеспечения, связанного с безопасностью, должно удовлетворять следующим требованиям:

a) основным методом подтверждения соответствия для программного обеспечения должно быть тестирование; анимацию и моделирование допускается использовать как дополнительные методы;

b) прогон программного обеспечения должен выполняться путем имитации:

- входных сигналов в нормальном режиме работы,
- предполагаемых случаев,
- нежелательных условий, требующих вмешательства системы;

c) поставщик и/или разработчик должны предоставить документированные результаты подтверждения соответствия безопасности программного обеспечения и всю имеющую отношение к этой операции документацию в распоряжение разработчика системы для того, чтобы дать ему возможность выполнить требования МЭК 61508-1 и МЭК 61508-2.

7.7.2.7 К качеству программных средств предъявляют следующие требования:

a) все программные средства, используемые при подтверждении соответствия, должны быть квалифицированы в соответствии со спецификациями, разработанными на основе международного стандарта (если таковой имеется) или национального стандарта (если таковой имеется), либо в соответствии с общепринятой процедурой;

b) оборудование, используемое при подтверждении соответствия, должно быть соответствующим образом квалифицировано, должна быть продемонстрирована пригодность для подтверждения соответствия всех используемых программных и аппаратных средств.

Примечание — В настоящем стандарте квалификация представляет собой операцию, которая демонстрирует выполнение конкретной спецификации, в отличие от базовых процедур проверки соответствия, которые могут использоваться по отношению к любой спецификации.

7.7.2.8 К результатам подтверждения соответствия программного обеспечения предъявляются следующие требования:

a) проверки должны показать, что все требования, предъявляемые к безопасности программного обеспечения (см. 7.2), выполняются правильно и что программная система не выполняет непредусмотренных функций;

b) тестовые примеры и их результаты должны быть документированы для последующего анализа и независимой экспертизы в соответствии с требованиями уровня полноты безопасности (МЭК 61508-1, пункт 8.2.12);

c) документированные результаты подтверждения соответствия безопасности программного обеспечения должны содержать утверждение о том, что программа прошла подтверждение соответствия, либо причины, по которым она не прошла его.

7.8 Модификация программного обеспечения

Примечания

1 См. также таблицу А.8 (приложение А).

2 Эта стадия представлена на рисунке 3 (см. 9.5).

7.8.1 Цели

Целью требований настоящего подраздела является внесение корректировок, улучшений или изменений в принятое программное обеспечение, гарантирующих сохранение уровня полноты безопасности программного обеспечения.

Примечание — В настоящем стандарте программное обеспечение (в отличие от аппаратного обеспечения) не может поддерживаться, оно всегда модифицируется.

7.8.2 Требования

7.8.2.1 Перед выполнением какой-либо модификации программного обеспечения должны быть подготовлены процедуры модификации (МЭК 61508-1, пункт 7.16).

Примечания

1 Требования 7.8.2.1 — 7.8.2.9 относятся, в первую очередь, к изменениям, выполняемым на этапе работы программного обеспечения. Они могут также применяться во время интеграции программируемой электроники, а также во время общей установки и ввода в эксплуатацию (МЭК 61508-1, пункт 7.13).

2 Пример модели процедуры модификации приведен в МЭК 61508-1 (рисунок 9).

7.8.2.2 Процесс модификации может начинаться только после появления запроса на санкционированную модификацию программного обеспечения в рамках процедур, определенных на этапе планирования безопасности (см. раздел 6), в котором приведена подробная информация:

a) об опасностях, на которые могут повлиять изменения;

b) о предлагаемых изменениях;

c) о причинах изменений.

Примечание — Причины появления запроса на модификацию могут быть, например, связаны с:

- тем, что функциональная безопасность оказалась ниже той, которая была определена в спецификациях;
- систематическими отказами;
- появлением нового или изменением действующего законодательства, относящегося к безопасности;
- модификацией EUC или способа его использования;
- модификацией общих требований к безопасности;
- анализом характеристик работы и обслуживания, который показывает, что эти характеристики имеют значения ниже запланированных;
- текущим аудитом функциональной безопасности.

7.8.2.3 Должен быть выполнен анализ влияния предлагаемых модификаций программного обеспечения на функциональную безопасность E/E/PE систем, связанных с безопасностью:

a) определить, необходим или нет анализ рисков;

b) определить, какие фазы жизненного цикла модулей безопасности следует повторить.

7.8.2.4 Результаты анализа влияния, полученные в 7.8.2.3, должны быть документированы.

7.8.2.5 Все модификации, оказывающие влияние на функциональную безопасность E/E/PE систем, связанных с безопасностью, должны приводить к возврату на соответствующую стадию жизненного цикла модулей безопасности. Все последующие стадии должны выполняться в соответствии с процедурами, определенными для отдельных стадий в соответствии с требованиями настоящего стандарта. При планировании безопасности (см. раздел 6) должны быть подробно описаны все последующие процессы.

Примечание — Может потребоваться выполнение полного анализа рисков и опасностей, в результате которого может появиться потребность в иных уровнях полноты безопасности, чем те, которые определены для систем, связанных с безопасностью, и внешних средств уменьшения риска.

7.8.2.6 Планирование безопасности для модификации программного обеспечения, связанного с безопасностью, должно включать в себя следующую информацию:

a) идентификацию персонала и определение требований к его квалификации;

b) подробную спецификацию модификации;

c) планирование верификации;

d) область применения операций повторного подтверждения соответствия и тестирования модификации в той степени, в которой этого требует уровень полноты безопасности.

7.8.2.7 Модификация должна быть выполнена в соответствии с разработанным планом.

7.8.2.8 Все модификации должны быть подробно документированы, включая:

a) запрос на модификацию/корректировку;

b) результаты анализа влияния, которое окажут предлагаемые модификации программного обеспечения на функциональную безопасность, и принятые решения с их обоснованием;

c) сведения об изменениях конфигурации программного обеспечения;

d) отклонения от нормальной работы и нормальных условий работы;

e) все документы, которые затрагиваются процессами модификации.

7.8.2.9 Информация (например, хронологическая) о деталях всех выполненных модификаций должна быть документирована. Документация должна включать в себя данные и результаты повторной верификации и повторного подтверждения соответствия.

Примечание — Требования 7.8.2.1 — 7.8.2.9 относятся, в первую очередь, к изменениям, выполняемым на этапе работы программного обеспечения. Они могут также применяться во время интеграции программируемой электроники, а также во время общей установки и ввода в эксплуатацию (МЭК 61508-1, пункт 7.13).

7.8.2.10 Оценка необходимых модификаций или корректировок должна зависеть от результатов анализа влияния модификаций и уровня полноты безопасности программного обеспечения.

7.9 Верификация программного обеспечения

Примечание — См. также 8.2, таблицы А.9 (приложение А) и В.8 (приложение В).

7.9.1 Цели

Цель требований настоящего подраздела — подтвердить в соответствии с требуемым уровнем полноты безопасности, что результаты, полученные на заданной стадии жизненного цикла модулей безопасности, являются корректными и соответствуют требованиям и стандартам, использовавшимся в качестве исходной информации для соответствующей стадии.

Примечания

1 Настоящий подраздел учитывает базовые аспекты верификации, которые являются общими для нескольких стадий жизненного цикла модулей безопасности. Настоящий подраздел не предъявляет дополнительных требований к элементам проверки верификации 7.4 (проверка программных модулей), 7.4.8 (интеграция программного обеспечения) и 7.5 (интеграция программируемой электроники), которые сами по себе представляют процессы верификации. Данный подраздел не требует также дополнительной верификации для процессов подтверждения соответствия программного обеспечения (см. 7.7), которое в настоящем стандарте определяется как демонстрация соответствия спецификации требований к безопасности (конечная верификация). Проверка того, является ли корректной сама спецификация, выполняется специалистами по предметным областям.

2 В зависимости от архитектуры программного обеспечения ответственность за проведение верификации программного обеспечения может быть разделена между всеми организациями, вовлеченными в разработку и модификацию программного обеспечения.

7.9.2 Требования

7.9.2.1 Верификация программного обеспечения для каждой стадии жизненного цикла модулей безопасности должна планироваться (см. 7.4) одновременно с разработкой; вся информация, относящаяся к этому вопросу, должна документироваться.

7.9.2.2 Планирование верификации программного обеспечения должно касаться критериев, методов и инструментария, используемого при верификации, в ходе его должны быть рассмотрены:

a) оценка требований полноты безопасности;

b) выбор и документирование стратегии, процессов и методов верификации;

c) выбор и использование инструментов верификации (тестовая программа, специальные программные средства для тестирования, имитаторы ввода/вывода и т.п.);

d) оценка результатов верификации;

e) исправления, которые должны быть сделаны.

7.9.2.3 Верификация программного обеспечения должна быть выполнена в соответствии с планом.

Примечание — Выбор методов и средств, предназначенных для верификации, а также степень независимости процессов верификации определяются рядом факторов и могут быть определены в стандартах для прикладных отраслей. К числу таких факторов относятся, например:

- размер проекта;
- степень сложности;
- степень новизны проекта;
- степень новизны технологии.

7.9.2.4 Должны быть документированы свидетельства того, что верифицируемая стадия завершена удовлетворительно во всех отношениях.

7.9.2.5 Документация, составляемая после каждой верификации, должна включать в себя:

a) перечень пунктов, подлежащих верификации;

b) идентификацию информации, по отношению к которой выполняется верификация;

c) перечень несоответствий.

Примечание — Примерами несоответствий являются программные модули, структуры данных и алгоритмы, которые плохо адаптированы к задаче.

7.9.2.6 Вся существенная информация, относящаяся к стадии N жизненного цикла модулей безопасности, которая необходима для правильного выполнения следующей стадии N + 1, должна быть доступна и верифицирована. К выходной информации стадии N относятся:

a) информация об адекватности спецификации описания проекта либо исходного текста программ, разработанных в ходе стадии N:

- функциональности,
- полноте безопасности, характеристикам и другим требованиям планирования безопасности (см. раздел 6),
- требованию понятности для коллектива разработчиков,
- безопасной модификации, допускающей дальнейшее развитие;

b) информация об адекватности планирования подтверждения соответствия и проверок, определенных для стадии N, определению и описанию проекта стадии N;

c) результаты несоответствия между:

- проверками, определенными для стадии N, и проверками, определенными для предыдущей стадии N–1,
- выходными данными стадии N.

7.9.2.7 С учетом 7.1.2.1 должны быть выполнены следующие операции верификации:

a) верификация требований к безопасности программного обеспечения (см. 7.9.2.8);

b) верификация архитектуры программного обеспечения (см. 7.9.2.9);

c) верификация проекта системы программного обеспечения (см. 7.9.2.10);

d) верификация проектов программных модулей (см. 7.9.2.11);

e) верификация исходных текстов программ (см. 7.9.2.12);

f) верификация данных (см. 7.9.2.13);

g) тестирование программных модулей (см. 7.4.7);

h) тестирование интеграции программного обеспечения (см. 7.4.8);

i) тестирование интеграции программируемой электроники (см. 7.5);

j) тестирование требований к безопасности программного обеспечения (подтверждение соответствия программного обеспечения) (см. 7.7).

7.9.2.8 Верификация требований к безопасности программного обеспечения

Когда определены требования к безопасности программного обеспечения (см. 7.2) и перед следующей стадией начато проектирование и разработка программного обеспечения, верификация должна проверять:

a) соответствуют ли требования к безопасности программного обеспечения (см. 7.2) требованиям к безопасности E/E/PES систем (МЭК 61508-2) в отношении функциональности, безопасности, полноты, характеристик и других требований к планированию безопасности;

b) соответствует ли планирование подтверждения соответствия программ для обеспечения безопасности (см. 7.3) требованиям к безопасности программного обеспечения (см. 7.2);

c) наличие несоответствия между:

- специфицированными требованиями к безопасности программного обеспечения (см. 7.2) и специфицированными требованиями к безопасности E/E/PE систем (МЭК 61508-2),
- специфицированными требованиями к безопасности программного обеспечения (см. 7.2) и планированием подтверждения соответствия безопасности программного обеспечения (см. 7.3).

7.9.2.9 Верификация архитектуры программного обеспечения

После того, как установлена спроектированная архитектура программного обеспечения, верификация должна проверить:

a) удовлетворяет ли описание проекта архитектуры программного обеспечения (см. 7.4.3) специфицированным требованиям к безопасности программного обеспечения (см. 7.2);

b) адекватны ли специфицированные проверки интеграции архитектуры программного обеспечения (см. 7.4.3) описанию проекта архитектуры программного обеспечения (см. 7.4.3);

c) адекватность атрибутов каждого основного компонента/подсистемы по отношению к:

- реализуемости требуемых характеристик безопасности,
- возможности проверки при последующей верификации,
- пониманию персоналом, выполняющим разработку и верификацию,
- безопасной модификации, позволяющей выполнять дальнейшее развитие программы;

d) наличие несовместимости между:

- описанием проекта архитектуры программного обеспечения (см. 7.4.3) и специфицированными требованиями к безопасности программного обеспечения (см. 7.2),
- описанием проекта архитектуры программного обеспечения (см. 7.4.3) и специфицированными тестами интеграции архитектуры программного обеспечения (см. 7.4.3),
- специфицированными тестами интеграции архитектуры программного обеспечения (см. 7.4.3) и планированием подтверждения соответствия безопасности программного обеспечения (см. 7.3).

7.9.2.10 Верификация проекта системы программного обеспечения

После завершения спецификации системы программного обеспечения верификация должна проверить:

a) удовлетворяет ли специфицированный проект системы программного обеспечения (см. 7.4.5) проекту архитектуры программного обеспечения (см. 7.4.3);

b) удовлетворяют ли специфицированные тесты интеграции системы программного обеспечения (см. 7.4.5) проекту системы программного обеспечения (см. 7.4.5);

c) адекватность атрибутов каждого основного компонента проекта системы программного обеспечения (см. 7.4.5) по отношению к:

- реализуемости требуемых характеристик безопасности,
- возможности проверки при последующей верификации,
- пониманию персоналом, выполняющим разработку и верификацию,
- безопасной модификации, позволяющей выполнять дальнейшее развитие программы.

Примечание — Проверки интеграции системы программного обеспечения могут быть определены как часть проверок интеграции архитектуры программного обеспечении.

d) наличие несоответствий между:

- специфицированным проектом системы программного обеспечения (см. 7.4.5) и описанием проекта архитектуры программного обеспечения (см. 7.4.3),
- описанием проекта системы программного обеспечения (см. 7.4.5) и специфицированными тестами интеграции системы программного обеспечения (см. 7.4.5),
- тестами интеграции системы программного обеспечения (см. 7.4.5) и специфицированными тестами интеграции архитектуры (см. 7.4.3).

7.9.2.11 Верификация проекта модулей программного обеспечения

После того, как определен проект каждого программного модуля, верификация должна проверить:

a) удовлетворяет ли специфицированный проект программного модуля (см. 7.4.5) проекту системного программного обеспечения (см. 7.4.5);

b) адекватны ли специфицированные проверки каждого программного модуля (см. 7.4.5) проекту программного модуля (см. 7.4.5);

c) адекватность атрибутов каждого программного модуля по отношению к:

- реализуемости требуемых характеристик безопасности (см. 7.2),
- возможности проверки при последующей верификации,
- пониманию персоналом, выполняющим разработку и верификацию,
- безопасной модификации, позволяющей выполнять дальнейшее развитие программы;

d) наличие несоответствий между:

- специфицированным проектом программного модуля (см. 7.4.5) и специфицированным проектом системы программного обеспечения (см. 7.4.5),
- специфицированным проектом каждого программного модуля (см. 7.4.5) и специфицированными проверками программных модулей (см. 7.4.5),
- специфицированными проверками программных модулей (см. 7.4.5) и специфицированными проверками интеграции системы программного обеспечения (см. 7.4.5).

7.9.2.12 Верификация исходного текста

Исходный текст должен быть верифицирован статическими методами для того, чтобы гарантировать соответствие специфицированным проектам программных модулей (см. 7.4.5), необходимым стандартам кодирования (см. 7.4.4) и требованиям планирования безопасности (см. 7.3).

Примечание — На ранних стадиях жизненного цикла безопасности программного обеспечения верификация является статической (например, изучение, просмотр, формальная проверка и т.п.). Верификация исходного текста включает такие методы, как просмотр и прогон программного обеспечения. Сочетание результатов верификации исходных текстов и проверок программного обеспечения гарантирует, что каждый программный модуль будет удовлетворять своей спецификации.

7.9.2.13 Верификация данных

a) Структуры данных, специфицированные во время проектирования, должны быть проверены на:

- полноту;
- согласованность;
- защиту от изменения или повреждения;
- соответствие функциональным требованиям системы, управляемой данными.

b) Прикладные данные должны быть проверены на:

- соответствие структурам данных;
- полноту;
- совместимость с базовым программным обеспечением (например, последовательность исполнения, совместимость на этапе исполнения и др.) и
- правильность значений данных.

Примечание — Примером прикладных данных являются программы для станков с числовым программным управлением. Системное программное обеспечение (обычно представляющее собой набор подпрограмм) действует как интерпретатор по отношению к прикладным данным. В других контекстах такие прикладные данные могут рассматриваться как прикладные программы.

c) Все параметры, которые могут быть изменены, должны быть проверены на защиту от:
- неверных и неопределенных начальных значений;
- ошибочных, несовместимых или необоснованных значений;
- несанкционированных изменений;
- повреждения данных.

d) Все промышленные интерфейсы и соответствующее программное обеспечение (т.е. датчики и устройства привода, а также автономные интерфейсы: см. 7.2.2.11) должны быть проверены на:
- выявление предполагаемых отказов интерфейса;
- устойчивость по отношению к предполагаемым отказам интерфейса.

e) Все коммуникационные интерфейсы и соответствующее программное обеспечение должны быть проверены на наличие адекватного уровня:
- обнаружения ошибок;
- защиты от повреждения;
- подтверждения данных.

8 Оценка функциональной безопасности

8.1 Цели и требования раздела 8 МЭК 61508-1 относятся к оценке программного обеспечения, связанного с безопасностью.

8.2 Если иное не оговорено в международных стандартах для прикладной отрасли, минимальный уровень безопасности для выполняющих оценку функциональной безопасности должен быть по МЭК 61508-1, пункт 8.2.12.

8.3 Оценка функциональной безопасности может использовать результаты процессов, приведенных в таблице А.10 (приложение А).

Примечание — Выбор методов, приведенных в приложениях А и В, не гарантирует, что будет достигнута необходимая полнота безопасности (см. 7.1.2.6). Лицо, выполняющее оценку, должно также рассмотреть:
- совместимость и взаимное дополнение выбранных методов, языков и инструментов для всего цикла разработки;
- полностью ли понимают разработчики методы, языки и инструменты, которые они используют;
- насколько хорошо адаптированы методы, языки и инструменты к конкретным проблемам, с которыми приходится сталкиваться при разработке.

Приложение А
(обязательное)

Руководство по выбору методов и средств

Некоторые из подразделов настоящего стандарта имеют ассоциированные с ними таблицы, например подраздел 7.2 связан с таблицей А.1. Более подробные таблицы, содержащиеся в приложении В, раскрывают содержание некоторых элементов таблиц приложения А, например таблица В.2 раскрывает содержание динамического анализа и тестирования из таблицы А.5.

Обзор методов и средств, упоминаемых в приложениях А и В, приведен в МЭК 61508-7. Для каждого из них даны рекомендации по уровню полноты безопасности, изменяющемуся от 1 до 4. Эти рекомендации обозначаются следующим образом:

HR: настоятельно рекомендуется использовать этот метод или средство для данного уровня полноты безопасности. Если метод или средство не используются, то на этапе планирования безопасности этому должно быть дано подробное объяснение, которое должно быть согласовано с экспертом.

R: метод или средство рекомендуется использовать для данного уровня полноты безопасности, но степень обязательности рекомендации ниже, чем в случае рекомендации HR.

---: для данного метода или средства не даются рекомендации ни за, ни против.

NR: данный метод или средство определенно не рекомендуется для этого уровня полноты безопасности. Если данный метод или средство используются, то на стадии планирования безопасности этому должно быть дано подробное обоснование, которое следует согласовать с экспертом.

Методы и средства следует выбирать в соответствии с уровнем полноты безопасности. Альтернативные или эквивалентные методы и средства обозначаются буквой, следующей за номером. Следует выполнять только один из альтернативных или эквивалентных методов/средств.

Ранжирование методов и средств связано с концепцией эффективности, используемой в МЭК 61508-2. При прочих равных условиях методы, имеющие ранг HR, будут более эффективны в предотвращении внесения систематических ошибок при разработке программного обеспечения либо (при разработке архитектуры программ) будут более эффективны при выявлении ошибок, оставшихся необнаруженными на этапе выполнения, по сравнению с методами, имеющими ранг R.

При большом числе факторов, влияющих на полноту безопасности программного обеспечения, невозможно дать алгоритм, определяющий такую комбинацию методов и средств, которая была бы корректной для любого заданного приложения. Тем не менее, руководство по использованию этих таблиц, иллюстрированное двумя рабочими примерами, приведено в МЭК 61508-6.

В случае конкретного приложения соответствующая комбинация методов или средств должна быть сформулирована при планировании безопасности, при этом методы и средства должны использоваться, если примечания к таблице не налагают иных требований.

Предварительное руководство по интерпретации таблиц для прикладного программирования приведено в МЭК 61508-6.

Примечание — Ссылки указывают на подробные описания методов/средств, приведенные в МЭК 61508-7.

Таблица А.1 — Спецификация требований к безопасности программного обеспечения (см. 7.2)

Метод/средство[1)]	Ссылка	SIL1	SIL2	SIL3	SIL4
1 Компьютерные средства разработки спецификаций	B.2.4	R	R	HR	HR
2a Полуформальные методы	Таблица B.7	R	R	HR	HR
2b Формальные методы, использующие, например, CCS, CSP, HOL, OBJ, LOTOS, временную логику, VDM и Z	C.2.4	---	R	R	HR

[1)] Методы/средства следует выбирать в соответствии с уровнем полноты безопасности. Альтернативные или эквивалентные методы/средства обозначаются буквами, следующими за числом. Следует выполнять только один из альтернативных или эквивалентных методов/мероприятий.

Примечания

1 Спецификация требований к безопасности программного обеспечения всегда будет требовать описания задачи на естественном языке и использования необходимой системы математических обозначений, отражающих содержание приложения.

2 Таблица отражает дополнительные требования для ясного и точного определения требований к безопасности программного обеспечения.

Таблица А.2 — Проектирование и разработка программного обеспечения: проектирование архитектуры программ (см. 7.4.3)

Метод/средство[1)]	Ссылка	SIL1	SIL2	SIL3	SIL4
1 Обнаружение и диагностика ошибок	C.3.1	---	R	HR	HR
2 Коды с обнаружением и исправлением ошибок	C.3.2	R	R	R	HR
3a Программирование с проверкой на ошибки	C.3.3	R	R	R	HR
3b Методы «подушки безопасности»	C.3.4	---	R	R	R
3c Многовариантное программирование	C.3.5	R	R	R	HR
3d Блоки восстановления	C.3.6	R	R	R	R
3e Восстановление предыдущего состояния	C.3.7	R	R	R	R
3f Переход к последнему достигнутому состоянию	C.3.8	R	R	R	R
3g Повторный запуск механизмов восстановления после ошибок	C.3.9	R	R	R	HR
3h Запоминание достигнутых состояний	C.3.10	---	R	R	HR
4 Постепенное отключение блоков	C.3.11	R	R	HR	HR
5 Искусственный интеллект — исправление ошибок	C.3.12	---	NR	NR	NR
6 Динамическая реконфигурация	C.3.13	---	NR	NR	NR
7a Структурные методы, включая, например, JSD, MASCOT, SADT и Yourdon	C.2.1	HR	HR	HR	HR
7b Полуформальные методы	Таблица B.7	R	R	HR	HR
7c Формальные методы, включая, например, CCS, CSP, HOL, LOTOS, OBJ, временную логику, VDM и Z	C.2.4	---	R	R	HR
8 Компьютерные средства разработки спецификаций	B.2.4	R	R	HR	HR

[1)] Методы/средства следует выбирать в соответствии с уровнем полноты безопасности. Альтернативные или эквивалентные методы/средства обозначаются буквами, следующими за числом. Следует выполнять только один из альтернативных или эквивалентных методов/средств.

Примечание — Приведенные в данной таблице средства, касающиеся устойчивости к ошибкам (контроль ошибок), должны рассматриваться совместно с требованиями, описанными в МЭК 61508-2, к архитектуре и контролю ошибок для аппаратных средств программируемых электронных устройств.

Таблица А.3 — Проектирование и разработка программного обеспечения: инструментальные средства поддержки и языки программирования (см. 7.4.4)

Метод/средство[1)]	Ссылка	SIL1	SIL2	SIL3	SIL4
1 Выбор соответствующего языка программирования	C.4.6	HR	HR	HR	HR
2 Использование языков программирования со строгой типизацией	C.4.1	HR	HR	HR	HR
3 Подмножество языка	C.4.2	---	---	HR	HR
4a Сертифицированные средства	C.4.3	R	HR	HR	HR
4b Инструментальные средства, заслуживающие доверия на основании опыта использования	C.4.4	HR	HR	HR	HR
5a Сертифицированный компилятор	C.4.3	R	HR	HR	HR
5b Трансляторы, заслуживающие доверия на основании опыта использования	C.4.4	HR	HR	HR	HR
6 Библиотека проверенных/сертифицированных модулей и компонентов	C.4.5	R	HR	HR	HR

[1)] Методы/средства следует выбирать в соответствии с уровнем полноты безопасности. Альтернативные или эквивалентные методы/средства обозначают буквами, следующими за числом. Следует выполнять только один из альтернативных или эквивалентных методов/средств.

Таблица А.4 — Проектирование и разработка программного обеспечения: детальное проектирование (см. 7.4.5 и 7.4.6)

Метод/средство[1)]	Ссылка	SIL1	SIL2	SIL3	SIL4
1a Методы, использующие структурирование, включая, например, JSD, MASCOT, SADT и Yourdon	C.2.1	HR	HR	HR	HR
1b Полуформальные методы	Таблица B.7	R	HR	HR	HR
1c Формальные методы, включая, например, CCS, CSP, HOL, LOTOS, OBJ, временную логику, VDM и Z	C.2.4	---	R	R	HR
2 Компьютерные средства проектирования	B.3.5	R	R	HR	HR
3 Защитное программирование	C.2.5	---	R	HR	HR
4 Модульный подход	Таблица B.9	HR	HR	HR	HR
5 Стандарты для проектирования и кодирования	Таблица B.1	R	HR	HR	HR
6 Структурное программирование	C.2.7	HR	HR	HR	HR
7 Использование проверенных/верифицированных программных модулей и компонентов (по возможности)	C.2.10, C.4.5	R	HR	HR	HR

[1)] Методы/средства следует выбирать в соответствии с уровнем полноты безопасности. Альтернативные или эквивалентные методы/средства обозначают буквами, следующими за числом. Следует выполнять только один из альтернативных или эквивалентных методов/средств.

Таблица А.5 — Проектирование и разработка программного обеспечения: тестирование программных модулей и интеграция (см. 7.4.7 и 7.4.8)

Метод/средство[1)]	Ссылка	SIL1	SIL2	SIL3	SIL4
1 Вероятностное тестирование	C.5.1	---	R	R	HR
2 Динамический анализ и тестирование	B.6.5, таблица B.2	R	HR	HR	HR
3 Запись и анализ данных	C.5.2	HR	HR	HR	HR
4 Функциональное тестирование и тестирование методом черного ящика	B.5.1, B.5.2, таблица B.3	HR	HR	HR	HR
5 Тестирование характеристик	C.5.20, таблица B.6	R	R	HR	HR
6 Тестирование интерфейса	C.5.3	R	R	HR	HR

[1)] Методы/средства следует выбирать в соответствии с уровнем полноты безопасности.

Примечание — Тестирование программных модулей и интеграции относится к процессам верификации (см. таблицу А.9, приложение А).

Таблица А.6 — Интеграция программируемых электронных устройств (программное обеспечение и аппаратные средства) (см. 7.5)

Метод/средство[1)]	Ссылка	SIL1	SIL2	SIL3	SIL4
1 Функциональное тестирование и тестирование методом черного ящика	B.5.1, B.5.2, таблица B.3	HR	HR	HR	HR
2 Тестирование характеристик	C.5.20, таблица B.6	R	R	HR	HR

[1)] Методы/средства следует выбирать в соответствии с уровнем полноты безопасности.

Примечание — Интеграция программируемых электронных устройств относится к процессам верификации (см. таблицу А.9, приложение А).

Таблица А.7 — Проверка безопасности программного обеспечения (см. 7.7)

Метод/средство[1)]	Ссылка	SIL1	SIL2	SIL3	SIL4
1 Вероятностное тестирование	С.5.1	---	R	R	HR
2 Имитация/моделирование	Таблица В.5	R	R	HR	HR
3 Функциональное тестирование и тестирование методом черного ящика	В.5.1, В.5.2, таблица В.3	HR	HR	HR	HR
[1)] Методы/средства следует выбирать в соответствии с уровнем полноты безопасности.					

Таблица А.8 — Модификация (см. 7.8)

Метод/средство[1)]	Ссылка	SIL1	SIL2	SIL3	SIL4
1 Анализ влияния	С.5.23	HR	HR	HR	HR
2 Повторная верификация измененных программных модулей	С.5.23	HR	HR	HR	HR
3 Повторная верификация программных модулей, на которые оказывают влияние изменения в других модулях	С.5.23	R	HR	HR	HR
4 Повторная верификация системы в целом	С.5.23	---	R	HR	HR
5 Управление конфигурацией программного обеспечения	С.5.24	HR	HR	HR	HR
6 Запись и анализ данных	С.5.2	HR	HR	HR	HR
[1)] Методы/средства следует выбирать в соответствии с уровнем полноты безопасности.					

Таблица А.9 — Верификация программного обеспечения (см. 7.9)

Метод/средство[1)]	Ссылка	SIL1	SIL2	SIL3	SIL4
1 Формальная проверка	С.5.13	---	R	R	HR
2 Вероятностное тестирование	С.5.1	---	R	R	HR
3 Статический анализ	В.6.4, таблица В.8	R	HR	HR	HR
4 Динамический анализ и тестирование	В.6.5, таблица В.2	R	HR	HR	HR
5 Метрики сложности программного обеспечения	С.5.14	R	R	R	R
Тестирование и интеграция программных модулей	См. таблицу А.5				
Проверка интеграции программируемых электронных устройств	См. таблицу А.6				
Тестирование программной системы (подтверждение соответствия)	См. таблицу А.7				

[1)] Методы/средства следует выбирать в соответствии с уровнем полноты безопасности.

Примечания

1 Для удобства все процессы, связанные с верификацией, были объединены в настоящей таблице. Это, однако, не предъявляет дополнительных требований к элементам верификации, связанным с динамическим тестированием в таблицах А.5 и А.6, которые относятся к процессам верификации. Настоящая таблица также не требует проведения верификационного тестирования в дополнение к подтверждению соответствия программного обеспечения (см. таблицу А.7, приложение А), которая в настоящем стандарте представляет демонстрацию соответствия спецификации требований к безопасности (конечную верификацию).

2 Верификация охватывает МЭК 61508-1 — МЭК 61508-3. Следовательно, первая верификация системы, связанной с безопасностью, относится к системным спецификациям более ранних уровней.

3 На ранних стадиях жизненного цикла системы безопасности программного обеспечения верификация является статической, она может включать в себя, например, изучение, просмотр, формальную проверку. Когда программа готова, становится возможным проведение динамического тестирования. Для верификации требуется объединение информации обоих типов. Например, верификация программного модуля статическими средствами включает в себя такие методы, как просмотр программ, прогон, статический анализ, формальная проверка. Верификация программ динамическими средствами включает функциональное тестирование, тестирование методом белого ящика, статистическое тестирование. Использование проверок обоих типов позволяет утверждать, что каждый программный модуль удовлетворяет соответствующей спецификации.

Таблица А.10 — Оценка функциональной безопасности (см. раздел 8)

Метод/средство[1)]	Ссылка	SIL1	SIL2	SIL3	SIL4
1 Таблица контрольных проверок	B.2.5	R	R	R	R
2 Таблицы решений и таблицы истинности	C.6.1	R	R	R	R
3 Метрики сложности программного обеспечения	C.5.14	R	R	R	R
4 Анализ отказов	Таблица B.4	R	R	HR	HR
5 Анализ общих отказов многовариантного программного обеспечения (если оно действительно используется)	C.6.3	---	R	HR	HR
6 Блок диаграммы надежности	C.6.5	R	R	R	R
[1)] Соответствующие методы/средства следует выбирать в соответствии с уровнем полноты безопасности.					

Приложение В
(обязательное)

Подробные таблицы

Примечание — Ссылки указывают на подробные описания методов/средств, приведенные в МЭК 61508-7.

Таблица В.1 — Стандарты для проектирования и кодирования (указанные в таблице А.4, приложение А)

Метод/средство[1)]	Ссылка	SIL1	SIL2	SIL3	SIL4
1 Использование стандартов кодирования	C.2.6.2	HR	HR	HR	HR
2 Не использовать динамические объекты	C.2.6.3	R	HR	HR	HR
3a Не использовать динамические переменные	C.2.6.3	---	R	HR	HR
3b Проверка создания динамических переменных при выполнении программы	C.2.6.4	---	R	HR	HR
4 Ограниченное использование прерываний	C.2.6.5	R	R	HR	HR
5 Ограниченное использование указателей	C.2.6.6	---	R	HR	HR
6 Ограниченное использование рекурсии	C.2.6.7	---	R	HR	HR
7 Не использовать безусловные переходы в программах, написанных на языках высокого уровня	C.2.6.2	R	HR	HR	HR

[1)] Методы/средства следует выбирать в соответствии с уровнем полноты безопасности. Альтернативные или эквивалентные методы/мероприятия обозначаются буквами, следующими за числом. Следует выполнять только один из альтернативных или эквивалентных методов/мероприятий.

Примечание — Не требуется применять методы 2 и 3a, если используют компилятор, который гарантирует выделение достаточного количества памяти для всех динамических переменных и объектов до начала выполнения программы либо вставляет проверки корректного выделения памяти в процессе выполнения.

Таблица В.2 — Динамический анализ и проверка (упоминаемые в таблицах А.5 и А.9, приложение А)

Метод/средство[1)]	Ссылка	SIL1	SIL2	SIL3	SIL4
1 Выполнение контрольного примера, начиная с анализа граничных значений	C.5.4	R	HR	HR	HR
2 Выполнение контрольного примера, начиная с обнаружения ошибки	C.5.5	R	R	R	R
3 Выполнение контрольного примера, начиная с внесения ошибки	C.5.6	---	R	R	R
4 Моделирование характеристик	C.5.20	R	R	R	HR
5 Разделение входных данных на классы эквивалентности	C.5.7	R	R	R	HR
6 Структурное тестирование	C.5.8	R	R	HR	HR

[1)] Методы/средства следует выбирать в соответствии с уровнем полноты безопасности.

Примечание — Анализ с использованием тестовых примеров проводят на уровне подсистем, он основывается на спецификациях и/или спецификациях и текстах программ.

Таблица В.3 — Функциональное тестирование и проверка методом черного ящика (упоминаемые в таблицах А.5, А.6 и А.7, приложение А)

Метод/средство[1)]	Ссылка	SIL1	SIL2	SIL3	SIL4
1 Выполнение контрольного примера, начиная с причинно-следственных диаграмм	8.6.6.2	---	---	R	R
2 Макетирование/анимация	С.5.17	---	---	R	R
3 Анализ граничных значений	С.5.4	R	HR	HR	HR
4 Разделение входных данных на классы эквивалентности	С.5.7	R	HR	HR	HR
5 Моделирование процесса	С.5.18	R	R	R	R

[1)] Методы/средства следует выбирать в соответствии с уровнем полноты безопасности.

Примечания

1 Анализ с использованием контрольных примеров выполняется на уровне систем программного обеспечения и основывается только на спецификациях.

2 Полнота моделирования будет зависеть от уровня полноты безопасности, сложности и применения.

Таблица В.4 — Анализ отказов (упоминается в таблице А.10, приложение А)

Метод/средство[1)]	Ссылка	SIL1	SIL2	SIL3	SIL4
1а Причинно-следственные диаграммы	В.6.6.2	R	R	R	R
1в Анализ методом дерева событий	В.6.6.3	R	R	R	R
2 Анализ методом дерева отказов	В.6.6.5	R	R	HR	HR
3 Анализ режимов, последствий и критичности отказов	В.6.6.4	R	R	HR	HR
4 Моделирование методом Монте-Карло	С.6.6	R	R	R	R

[1)] Методы/средства следует выбирать в соответствии с уровнем полноты безопасности. Альтернативные или эквивалентные методы/средства обозначают буквами, следующими за числом. Следует выполнять только один из альтернативных или эквивалентных методов/средств.

Примечание — Предварительно должен быть выполнен анализ рисков для отнесения программного обеспечения к соответствующему уровню полноты безопасности.

Таблица В.5 — Моделирование (упоминается в таблице А.7, приложение А)

Метод/средство[1)]	Ссылка	SIL1	SIL2	SIL3	SIL4
1 Схемы потока данных	С.2.2	R	R	R	R
2 Конечные автоматы	В.2.3.2	---	R	HR	HR
3 Формальные методы	С.2.4	---	R	R	HR
4 Моделирование характеристик	С.5.20	R	HR	HR	HR
5 Метод сетей Петри	В.2.3.3	---	R	HR	HR
6 Макетирование/анимация	С.5.17	R	R	R	R
7 Структурные схемы	С.2.3	R	R	R	HR

[1)] Методы/средства следует выбирать в соответствии с уровнем полноты безопасности.

Примечание — Если какой-то конкретный метод не перечислен в таблице, не следует считать, что он был исключен из рассмотрения. Этот метод должен соответствовать настоящему стандарту.

Таблица В.6 — Тестирование характеристик (упоминается в таблицах А.5 и А.6, приложение А)

Метод/средство[1)]	Ссылка	SIL1	SIL2	SIL3	SIL4
1 Проверка на критические и напряженные нагрузки	С.5.21	R	R	HR	HR
2 Ограничения на время реакции и память	С.5.22	HR	HR	HR	HR
3 Требования к характеристикам	С.5.19	HR	HR	HR	HR
[1)] Методы/средства следует выбирать в соответствии с уровнем полноты безопасности.					

Таблица В.7 — Полуформальные методы (упоминаются в таблицах А.1, А.2 и А.4, приложение А)

Метод/средство[1)]	Ссылка	SIL1	SIL2	SIL3	SIL4
1 Логические/функциональные блок-схемы	См. примечание ниже	R	R	HR	HR
2 Диаграммы последовательности	См. примечание ниже	R	R	HR	HR
3 Диаграммы потоков данных	С.2.2	R	R	R	R
4 Конечные автоматы/диаграммы переходов	8.2.3.2	R	R	HR	HR
5 Метод сетей Петри	В.2.3.3	R	R	HR	HR
6 Таблицы решений, таблицы истинности	С.6.1	R	R	HR	HR
[1)] Методы/средства следует выбирать в соответствии с уровнем полноты безопасности. Примечание — Логические и функциональные блок-схемы и диаграммы последовательности описаны в МЭК 61131-3.					

Таблица В.8 — Статический анализ (упоминается в таблице А.9, приложение А)

Метод/средство[1)]	Ссылка	SIL1	SIL2	SIL3	SIL4
1 Анализ граничных значений	С.5.4	R	R	HR	HR
2 Таблица контрольных проверок	В.2.5	R	R	R	R
3 Анализ управляющей логики	С.5.9	R	HR	HR	HR
4 Анализ потоков данных	С.5.10	R	HR	HR	HR
5 Поиск ошибок	С.5.5	R	R	R	R
6 Проверка исходных текстов	С.5.15	---	R	R	HR
7 Анализ скрытых путей исполнения	С.5.11	---	---	R	R
8 Символьный редактор	С.5.12	R	R	HR	HR
9 Прогоны/просмотры проекта	С.5.16	HR	HR	HR	HR
[1)] Методы/средства следует выбирать в соответствии с уровнем полноты безопасности.					

Таблица В.9 — Модульный подход (упоминается в таблице А.4, приложение А)

Метод/средство[1)]	Ссылка	SIL1	SIL2	SIL3	SIL4
1 Предельный размер программного модуля	С.2.9	HR	HR	HR	HR
2 Скрытие информации/инкапсуляция	С.2.8	R	HR	HR	HR
3 Предельное число параметров	С.2.9	R	R	R	R
4 Одна точка входа и одна точка выхода в каждой подпрограмме и функции	С.2.9	HR	HR	HR	HR
5 Полностью определенный интерфейс	С.2.9	HR	HR	HR	HR
[1)] Использование одного метода является, по-видимому, недостаточным. Следует рассматривать все подходящие методы. Примечание — Информацию по всем этим методам, за исключением скрытия информации/инкапсуляции, см. в МЭК 61508-7, пункт С.2.9, приложение С.					

Приложение С
(справочное)

Сведения о соответствии национальных стандартов Российской Федерации ссылочным международным стандартам

Таблица С.1

Обозначение ссылочного международного стандарта	Обозначение и наименование соответствующего национального стандарта
МЭК 61508-1:1998	ГОСТ Р МЭК 61508-1—2007 Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 1. Общие требования
МЭК 61508-2:2000	*
МЭК 61508-4:1998	ГОСТ Р МЭК 61508-4—2007 Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 4. Термины и определения
МЭК 61508-5:1998	ГОСТ Р МЭК 61508-5—2007 Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 5. Рекомендации по применению методов определения уровней полноты безопасности
МЭК 61508-6:2000	*
МЭК 61508-7:2000	*
ИСО/МЭК Руководство 51:1999	ГОС Р 51898—2002 Аспекты безопасности. Правила включения в стандарты
МЭК Руководство 104:1997	*
* Соответствующий национальный стандарт отсутствует. До его утверждения рекомендуется использовать перевод на русский язык данного международного стандарта. Перевод данного международного стандарта находится в Федеральном информационном фонде технических регламентов и стандартов.	

Библиография

МЭК 61151:1992	Приборы атомной промышленности. Усилители и предварительные усилители, используемые с детекторами ионизирующего излучения. Процедуры проверки
ИСО/МЭК 12207:1995	Информационные технологии. Процессы в жизненном цикле программного обеспечения
ANSI/ISA S84:1996	Применение систем, оснащенных средствами обеспечения безопасности в обрабатывающих отраслях

УДК 62-783:614.8:331.454:006.354 ОКС 13.110 Т51

Ключевые слова: безопасность функциональная, жизненный цикл систем, электрические компоненты, электронные компоненты, программируемые электронные компоненты и системы, системы, связанные с безопасностью, планирование функциональной безопасности, программное обеспечение, уровень полноты безопасности

Редактор *Р.Г. Говердовская*
Технический редактор *Н.С. Гришанова*
Корректор *В.Е. Нестерова*
Компьютерная верстка *Л.А. Круговой*

Сдано в набор 17.04.2008. Подписано в печать 04.06.2008. Формат 60 × 84 1/8. Бумага офсетная. Гарнитура Ариал. Печать офсетная. Усл. печ. л. 4,65. Уч.-изд. л. 4,50. Тираж 258 экз. Зак. 641.

ФГУП «СТАНДАРТИНФОРМ», 123995 Москва, Гранатный пер., 4.

gost.e-ecolog.ru 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