ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ГОСТ Р МЭК 62138— 2010**

# АТОМНЫЕ ЭЛЕКТРОСТАНЦИИ

## Системы контроля и управления, важные для безопасности. Программное обеспечение компьютерных систем, выполняющих функции категорий B и C

IEC 62138:2004
Nuclear power plants —
Instrumentation and control systems important for safety —
Software aspects for computer-based
systems performing category B or C functions
(IDT)

Издание официальное

Москва
Стандартинформ
2011

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 на основе аутентичного перевода на русский язык стандарта, указанного в пункте 4, который выполнен Открытым акционерным обществом «Всероссийский научно-исследовательский институт атомных электростанций» (ОАО«ВНИИАЭС») и Автономной некоммерческой организацией «Измерительно-информационные технологии» (АНО «Изинтех»)

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 322 «Атомная техника»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 30 ноября 2010 г. № 740-ст

4 Настоящий стандарт идентичен международному стандарту МЭК 62138:2004 «Атомные электростанции. Системы контроля и управления, важные для безопасности. Программное обеспечение компьютерных систем, выполняющих функции категорий В и С» (IEC 62138:2004 «Nuclear power plants — Instrumentation and control systems important for safety — Software aspects for computer-based systems performing category B or C functions»).

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

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, 2011

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 . . . 2
4 Ключевые концепции и допущения . . . 5
4.1 Типы программного обеспечения . . . 5
4.2 Типы данных . . . 6
4.3 Жизненные циклы безопасности и программного обеспечения . . . 7
4.4 Принципы градации . . . 10
5 Требования к программному обеспечению СКУ, выполняющих функции категории С[1)] . . . 11
5.1 Общие требования . . . 11
5.2 Выбор ранее разработанного программного обеспечения . . . 14
5.3 Спецификация требований к программному обеспечению . . . 15
5.4 Проект программного обеспечения . . . 17
5.5 Реализация нового программного обеспечения . . . 17
5.6 Программные аспекты интеграции системы . . . 18
5.7 Программные аспекты валидации системы . . . 18
5.8 Инсталляция программного обеспечения на штатном месте . . . 19
5.9 Протоколы отклонений от нормы . . . 20
5.10 Модификация программного обеспечения . . . 20
6 Требования к программному обеспечению СКУ, выполняющих функции категории В . . . 20
6.1 Общие требования . . . 20
6.2 Выбор ранее разработанного программного обеспечения . . . 24
6.3 Спецификация требований к программному обеспечению . . . 28
6.4 Проект программного обеспечения . . . 30
6.5 Реализация нового программного обеспечения . . . 31
6.6 Программные аспекты интеграции системы . . . 33
6.7 Программные аспекты валидации системы . . . 33
6.8 Инсталляция программного обеспечения на штатном месте . . . 34
6.9 Протоколы отклонений от нормы . . . 35
6.10 Модификация программного обеспечения . . . 35
Приложение ДА (справочное) Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации . . . 37
Библиография . . . 38

# Введение

## Структура серии стандартов ПК 45А. Соотношения с другими документами МЭК, МАГАТЭ и ИСО

Документом высшего уровня серии стандартов ПК 45А является МЭК 61513. В этом стандарте рассматриваются требования к системам контроля и управления, важным для безопасности атомных станций (АС), и он лежит в основе серии стандартов ПК 45А.

В МЭК 61513 имеются непосредственные ссылки на другие стандарты ПК 45А по общим вопросам, связанным с категоризацией функций и классификацией систем, оценкой соответствия, разделением систем, защитой от отказов по общей причине, аспектами программного обеспечения компьютерных систем, аспектам технического обеспечения компьютерных систем и проектированию пунктов управления. Те стандарты, на которые имеются непосредственные ссылки, рекомендуется использовать совместно с МЭК 61513 в качестве согласованной подборки документов. Другие стандарты ПК 45А, на которые в МЭК 61513 нет непосредственных ссылок, являются стандартами, связанными с конкретным оборудованием, техническими методами или конкретной деятельностью. Обычно эти документы низкого уровня, в которых по общим вопросам имеются ссылки на документы более высокого уровня, могут использоваться самостоятельно.

Для МЭК 61513 принята форма представления, аналогичная форме представления базовой публикации по безопасности МЭК 61508, с его структурой общего жизненного цикла безопасности и структурой жизненного цикла системы, и в нем дана интерпретация общих требований МЭК 61508-1, МЭК 61508-2 и МЭК 61508-4 для применения в ядерной области. Согласованность с МЭК 61513 будет способствовать соответствию требованиям МЭК 61508, интерпретированным для атомной промышленности. В этой структуре МЭК 60880 и МЭК 62138 соответствуют МЭК 61508-3 применительно к ядерной области.

В МЭК 61513 есть ссылки на стандарты ИСО, а также на документ МАГАТЭ 50-С-QA (теперь замененный на МАГАТЭ 50-С/SG-Q) по вопросам, связанным с обеспечением качества.

В серии стандартов ПК 45А последовательно реализуются и детализируются принципы и базовые аспекты безопасности, предусмотренные Правилами МАГАТЭ по безопасности атомных электростанций, а также серией документов МАГАТЭ по безопасности, в частности требованиям NS-R-1 «Безопасность атомных электростанций: проектирование» и руководством по безопасности NS-G-1.3 «Системы контроля и управления, важные для безопасности атомных электростанций». Термины и определения, применяемые в серии стандартов ПК 45А, согласованы с терминами и определениями, применяемыми в МАГАТЭ.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

**АТОМНЫЕ ЭЛЕКТРОСТАНЦИИ**

# Системы контроля и управления, важные для безопасности. Программное обеспечение компьютерных систем, выполняющих функции категорий В и С

Nuclear power plants. Instrumentation and control systems important for safety. Sofware aspects for computer-based systems performing category B or C functions

**Дата введения — 2012—01—01**

## 1 Область применения

Настоящий стандарт содержит требования к программному обеспечению компьютерных систем, выполняющих функции безопасности категории В или С, определенные в МЭК 61226. Настоящий стандарт дополняет МЭК 60880 и МЭК 60880-2, которые содержат требования к программному обеспечению компьютерных систем, выполняющих функции безопасности категории А.

Настоящий стандарт также согласован с МЭК 61513 и дополняет его. Действия, относящиеся главным образом к системному уровню (например, интеграция, валидация и установка), в настоящем стандарте подробно не рассматриваются: неспецифичные для программного обеспечения требования, изложенные в МЭК 61513.

МЭК 61513 следующим образом устанавливает классы безопасности систем, важных для безопасности:

- системы контроля и управления (СКУ) класса безопасности 1 — в основном предназначены для выполнения функций безопасности категории А, но могут также выполнять функции безопасности категории В и/или С и функции, не классифицированные по безопасности;
- СКУ класса безопасности 2 — в основном предназначены для исполнения функций безопасности категории В, но могут также выполнять функции безопасности категории С и функции, не классифицированные по безопасности;
- СКУ класса безопасности 3 — в основном предназначены для выполнения функций безопасности категории С, но могут также выполнять функции, не классифицированные по безопасности.

Так как классифицированная конкретная СКУ может выполнять функции различных категорий безопасности и даже функции, не классифицированные по безопасности, требования настоящего стандарта относятся к классу безопасности СКУ.

Настоящий стандарт учитывает существующую практику по разработке программного обеспечения для СКУ, в частности:

- использование ранее разработанного программного обеспечения, оборудования и комплексов оборудования, которые не обязательно были разработаны по отраслевым стандартам ядерной промышленности;
- использование специализированных устройств типа «черный ящик» со встроенным программным обеспечением;
- использование проблемно-ориентированных языков программирования.

Настоящий стандарт не предназначен для использования в качестве технического руководства для программного обеспечения общего назначения. В нем представлены требования, которым должно соответствовать программное обеспечение СКУ классов безопасности 2 или 3 для достижения целей системного обеспечения ядерной безопасности.

**Издание официальное**

## 2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие международные стандарты:

МЭК 61226 Атомные станции. Системы контроля и управления, важные для безопасности. Классификация функций контроля и управления (IEC 61226, Nuclear power plants — Instrumentation and control systems important for safety — Classification of instrumentation and control functions)

МЭК 61513:2001 Атомные станции. Системы контроля и управления, важные для безопасности. Общие требования (IEC 61513, Nuclear power plants — Instrumentation and control for systems important to safety — General requirements for systems)

Для датированных ссылок применяют указанный вариант. Для недатированных ссылок применяют последнее издание документа (включая все изменения и поправки).

## 3 Термины и определения

В настоящем стандарте применяются следующие термины с соответствующими определениями:

3.1 **анимация** (animation): Процесс, посредством которого указанное в спецификации поведение демонстрируется с реальными значениями, полученными из задающих поведение выражений и некоторых входных величин.

[МЭК 60880]

3.2 **прикладная функция** (application function): Функция системы контроля и управления по выполнению задачи, связанной с контролируемым процессом, а не с функционированием самой системы.

[МЭК 61513]

3.3 **проблемно-ориентированный язык** (application oriented language): Компьютерный язык, специально разработанный для определенного типа применений и используемый лицами, являющимися специалистами в данном типе применений.

Примечание 1 — Группы оборудования обычно характеризуются проблемно-ориентированными языками, обеспечивающими удобное приспособление оборудования к специфичным требованиям.

Примечание 2 — Проблемно-ориентированные языки могут использоваться для обеспечения функциональных требований к системе контроля и управления и/или для установления или разработки прикладного программного обеспечения. Они могут базироваться на текстах, графике или на том и другом.

Примечание 3 — Например, языки функциональных блок-схем, языки, определенные МЭК 61131-3.

Примечание 4 — См. также термин «универсальный язык».

3.4 **прикладное программное обеспечение** (application software): Часть программного обеспечения СКУ, которая обеспечивает выполнение прикладных функций.

[МЭК 61513]

Примечание — См. также термины: «системное программное обеспечение», «операционная система».

3.5 **категория функции контроля и управления** (category of an I&C function): Одно из трех возможных обозначений (A,B,C) функций контроля и управления, устанавливаемое в результате рассмотрения влияния выполняемой функции на безопасность. Если функция не связана с безопасностью, то она не классифицируется.

[МЭК 61513]

Примечание — См. также термин «класс СКУ».

3.6 **класс СКУ** (class of an I&C system): Одно из трех возможных назначений (1, 2, 3) СКУ, важных для безопасности, устанавливаемое в результате рассмотрения требований, предъявляемых к выполнению функций контроля и управления, имеющих различное отношение к безопасности. Если система контроля и управления не выполняет функции, связанные с безопасностью, то она не классифицируется.

[МЭК 61513]

Примечание — См. также термин «категория функции контроля и управления».

3.7 **сложность** (complexity): Степень трудности понимания и верификации проекта, реализации или поведения системы или компонента.

[МЭК 61513]

3.8 **управление конфигурацией** (configuration management): Порядок применения технической и административной директив и контроля с целью определения и документирования функциональных и физических характеристик сложного устройства, управления их изменением, ведения записей и отчетов об изменении в работе и настройке, а также проверки соответствия конкретным требованиям.

[МЭК 61513]

3.9 **спецификация проекта** (design specification): Документ или комплект документов, который описывает устройство и работу изделия и используется в качестве основы для применения и интеграции изделия.

3.10 **документация по безопасности** (documentation for safety): Документ или комплект документов, который указывает, каким образом продукт может безопасно использоваться для решения прикладных задач, важных для безопасности.

3.11 **комплекс оборудования** (equipment family): Набор приборных и программных компонентов, которые могут работать совместно в одной или более определенных структурах (конфигурациях). Разработка специальной конфигурации для АС и соответствующего прикладного программного обеспечения может поддерживаться программными средствами; обеспечивает набор стандартных операций (библиотеку прикладных функций), которые могут быть объединены, образуя специальное прикладное программное обеспечение.

[МЭК 61513]

Примечание 1 — Комплекс оборудования может быть изделием определенного изготовителя или набором изделий, соединенных и настроенных поставщиком.

Примечание 2 — Термин «платформа оборудования» иногда используется как синоним термина «комплекс оборудования».

3.12 **погрешность** (error): Разность между рассчитанным, наблюдаемым или измеренным значением величины или параметра и истинным, установленным или теоретическим значением величины или параметра.

[МЭК 61513]

Примечание — См. также термины: «ошибка», «неисправность», «отказ».

3.13 **рабочая программа** (executable code): Программное обеспечение, которое включено в специализированную систему.

Примечание — Рабочая программа обычно включает в себя команды, которые должны выполняться техническим обеспечением специализированной системы, и сопутствующие данные.

3.14 **отказ** (failure): Отклонение реального функционирования от запланированного.

[МЭК 61513]

Примечание — См. также термины: «ошибка», «неисправность», «погрешность».

3.15 **дефект** (fault): Неисправность или ошибка в компоненте технического обеспечения, программного обеспечения или системы.

[МЭК 61513]

Примечание 1 — Дефекты могут быть подразделены на случайные и систематические. Случайные дефекты возникают в результате деградации технического обеспечения и вызывают отказы в непредвиденные моменты времени. Систематические неисправности возникают вследствие ошибок в проекте (например, ошибок в программном обеспечении) и при одинаковых условиях систематически ведут к одинаковым отказам.

Примечание 2 — Дефект (особенно дефект, связанный с проектированием) может оставаться незамеченным до тех пор, пока сохраняются условия, при которых он не отражается на выполнении функции, т.е. пока не произойдет отказ.

Примечание 3 — См. также термины: «ошибка», «погрешность», «отказ».

3.16 **функциональная валидация** (functional validation): Проверка правильности применения спецификаций прикладных функций относительно исходных требований к функциям и эксплуатационным характеристикам станции. Функциональная валидация дополняет валидацию системы и оценивается ее соответствие спецификации функций.

[МЭК 61513]

3.17 **универсальный язык** (general-purpose language): Компьютерный язык, разработанный для всех видов применения.

Примечание 1 — Программное обеспечение операционной системы групп оборудования обычно реализуется с использованием универсальных языков.

Примечание 2 — Например, Ада, Си, Паскаль.

Примечание 3 — См. также термин «проблемно-ориентированный язык».

3.18 **интеграция** (integration): Последовательная сборка компонентов и их проверка внутри завершенной системы.

3.19 **архитектура контроля и управления** (I&C architecture): Организованная структура важных для безопасности систем контроля и управления станции.

[МЭК 61513]

3.20 **ошибка** (mistake): Действие (или бездействие) человека, которое приводит к незапланированному результату.

[МЭК 60880]

Примечание — См. также термины: «дефект», «погрешность», «отказ».

3.21 **режим работы** (mode of behaviour): Функциональное состояние элемента программы, которое обеспечивает определенный рабочий режим.

Примечание — Пример: режим инициализации, нормальный режим, неполный режим в случае наличия дефекта в элементе или его окружении.

3.22 **операционное программное обеспечение системы** (operational system software): Часть программного обеспечения системы, рабочая программа которого выполняется на основном процессоре во время работы системы.

[МЭК 61513]

Примечание 1 — Например, операционная система, входные/выходные драйверы и драйверы коммуникаций, обработчик исключительных ситуаций, планировщик, сигнал прерывания управления, диагностика в реальном времени, управление резервированием и смягченной деградацией, библиотеки прикладного программного обеспечения.

Примечание 2 — См. также термины: «прикладное программное обеспечение», «системное программное обеспечение».

3.23 **параметр** (parameter): Элемент данных, управляющий поведением СКУ и/или ее программного обеспечения, который может быть изменен операторами во время работы станции.

3.24 **ранее разработанное программное обеспечение** (pre-developed software; PDS): Часть программного обеспечения, которая уже существует и доступна как коммерческий или запатентованный продукт.

[МЭК 61513]

Примечание 1 — Ранее разработанное программное обеспечение можно разделить на программное обеспечение общего назначения, которое не разрабатывалось для определенного технического обеспечения, и программное обеспечение, интегрированное в компоненты оборудования, которое применяется совместно с оборудованием.

Примечание 2 — В настоящем стандарте этот термин не охватывает инструментальные программы, даже если они были ранее разработаны.

3.25 **программа** (program): Написанный человеком документ, который преобразуется в рабочую программу автоматизированными инструментальными программами.

Примечание — К ним можно причислить традиционные программы, написанные с помощью универсальных языков. Сюда также входят программы, написанные с помощью проблемно-ориентированных языков.

3.26 **защищенность** (security): Способность компьютерной системы обеспечивать необходимую уверенность в том, что неуполномоченные лица и системы не смогут модифицировать программное обеспечение и его данные и не будут иметь доступ к функциям системы при том, что такая возможность будет обеспечена для уполномоченных лиц и систем.

[МЭК 61513]

3.27 **программное обеспечение** (software): Программы (т.е. наборы упорядоченных команд), данные, правила и любая связанная с этим документация, имеющая отношение к работе компьютеризированных СКУ.

[МЭК 60880]

3.28 **модификация программного обеспечения** (software modification): Изменение в уже согласованном документе (документах), которое(рые) ведет(дут) к изменению рабочей программы.

Примечание — Модификация программного обеспечения может происходить на начальной стадии разработки программного обеспечения (например, для устранения ошибок, найденных на более поздних стадиях разработки) либо уже после введения программного обеспечения в эксплуатацию.

3.29 **компонент программного обеспечения** (software component): Один из элементов проекта, составляющих часть программного обеспечения. Он может быть разделен на другие компоненты программного обеспечения.

[МЭК 61513]

3.30 **разработка программного обеспечения** (software development): Этап жизненного цикла программного обеспечения, который осуществляется для создания программного обеспечения СКУ или программного продукта.

Примечание — Разработка охватывает все действия — от создания спецификации требований к программному обеспечению до его валидации и установки на объекте.

3.31 **жизненный цикл безопасности программного обеспечения** (software safety lifecycle): Необходимая деятельность при разработке и эксплуатации программного обеспечения СКУ, важной для безопасности, осуществляемая в течение всего периода времени, начиная с разработки спецификации требований к программному обеспечению и заканчивая выведением программного обеспечения из эксплуатации.

[МЭК 61513]

3.32 **статический анализ** (static analysis): Процесс оценки системы или компонента, базирующийся на ее (его) форме, структуре, содержании или документации.

[МЭК 60880]

3.33 **системное программное обеспечение** (system software): Часть программного обеспечения СКУ, созданная для конкретного компьютера или семейства оборудования с целью облегчения разработки, эксплуатации и модификации этих объектов и связанных с ними программ.

[МЭК 61513]

Примечание 1 — Системное программное обеспечение комплекса оборудования обычно состоит из операционного программного обеспечения системы и поддерживающего программного обеспечения (инструментальных программ).

Примечание 2 — См. также термины: «прикладное программное обеспечение», «операционное программное обеспечение системы».

3.34 **валидация программного обеспечения** (software validation): Тестирование и оценка интегрированной системы на соответствие спецификациям функциональных, эксплуатационных характеристик и интерфейсов, содержащихся в требованиях к СКУ.

3.35 **верификация** (verification): Подтверждение экспертизой и предоставлением объективного свидетельства того, что результаты функционирования соответствуют целям и требованиям, определенным для такого функционирования.

[ИСО 12207]

## 4 Ключевые концепции и допущения

В настоящем разделе представлены некоторые ключевые концепции и допущения, которые касаются характера программного обеспечения и вопросов его разработки для СКУ классов безопасности 2 или 3 и на которых базируется нормативный документ.

### 4.1 Типы программного обеспечения

Различные виды работы, выполняемой программным обеспечением и компонентами программного обеспечения в типичной СКУ или в архитектуре контроля и управления показаны на рисунке 1. Компоненты программного обеспечения могут часто определяться как программное обеспечение системы или прикладное программное обеспечение. Программное обеспечение системы может также быть подразделено на операционное программное обеспечение системы, встроенное в СКУ, важную для безопасности, и поддерживающее программное обеспечение (или инструментальные программы), которое является автономным либо встроенным в системы поддержки, классифицируемые как не важные для безопасности. Программное обеспечение может также находиться в специализированных устройствах, таких как датчики и исполнительные механизмы, устройства связи и бесперебойного электропитания (БЭП).

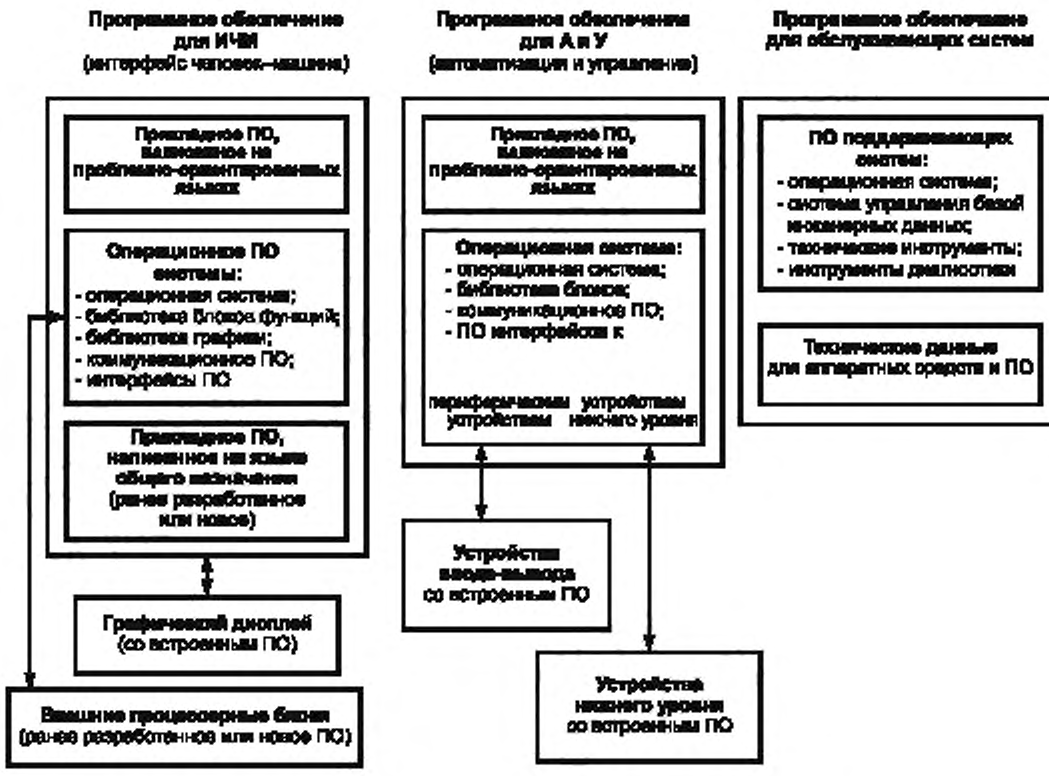



Рисунок 1 — Типичные части программного обеспечения в компьютерных СКУ

Программное обеспечение СКУ также подразделяют на ранее разработанное программное обеспечение (которое обычно обеспечивает функции, полезные для ряда СКУ) и новое (которое разработано для конкретных задач СКУ). Системное программное обеспечение обычно бывает ранее разработанным, а прикладное программное обеспечение — новым, но это правило не абсолютное. Требования настоящего стандарта, которым должно соответствовать новое программное обеспечение, могут также быть предъявлены и к ранее разработанному программному обеспечению. Настоящий стандарт также содержит альтернативные требования, которым должно, в частности, соответствовать ранее разработанное программное обеспечение или встроенное программное обеспечение специализированных устройств.

Многие современные комплексы оборудования обеспечены ориентированными на прикладные задачи обширными инструментальными средствами разработки, которые позволяют инженерам станции или системным инженерам устанавливать свои требования к программному обеспечению с использованием графических методов. Инструментальные средства могут автоматически транслировать графические программы в исполняемое прикладное программное обеспечение. При надлежащем качестве этих инструментальных средств данный подход может применяться для уменьшения риска появления ошибок.

### 4.2 Типы данных

Во многих проектах систем широко используются данные конфигурации. Данные конфигурации могут быть связаны с операционным программным обеспечением системы или с прикладным программным обеспечением. Данные конфигурации, связанные с прикладным программным обеспечением, состоят, главным образом, из технических данных станции, вытекающих из проекта станции, и обычно подготавливаются проектантами станции, которые не обязаны иметь навыков разработки программного обеспечения. Данные конфигурации могут быть разделены следующим образом:

- элементы данных, которые не предназначены для изменения операторами станции в режиме реального времени и должны соответствовать тем же требованиям, которые предъявляются к остальной части программного обеспечения;

- параметры, т.е. элементы данных, которые могут изменяться операторами при эксплуатации станции (например, пределы срабатывания аварийной сигнализации, данные калибровки аппаратуры) и к которым предъявляются специфичные требования.

### 4.3 Жизненные циклы безопасности и программного обеспечения

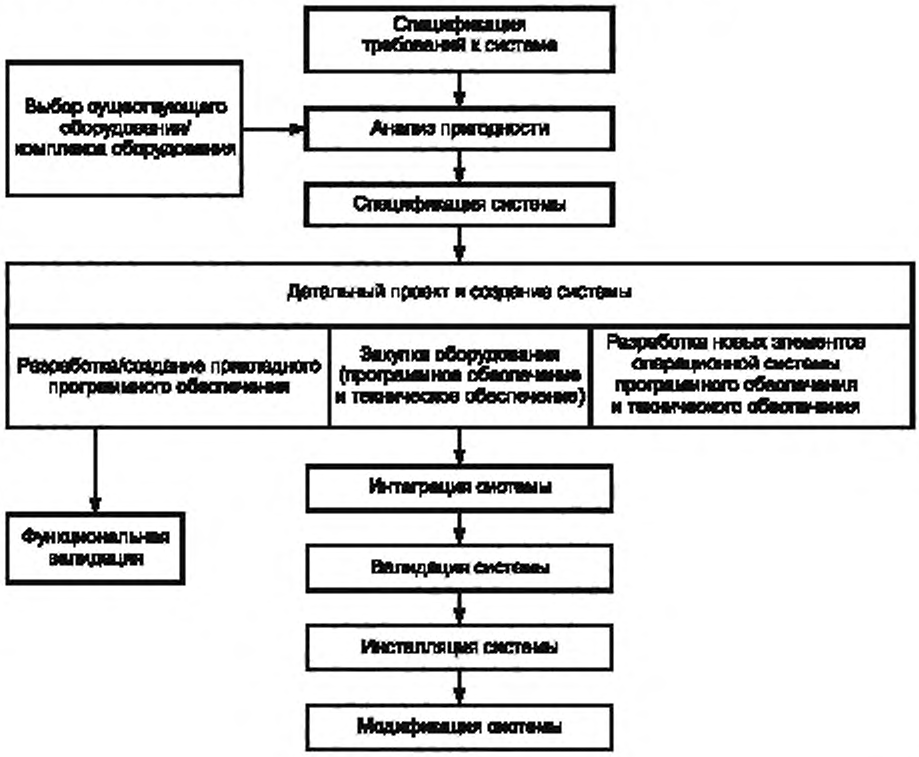



Рисунок 2 — Мероприятия жизненного цикла системы безопасности (в соответствии с МЭК 61513)

Программное обеспечение обычно вносит существенный вклад в функции, выполняемые СКУ. Оно может также поддерживать дополнительные функции, введенные в соответствии с проектом системы (например, инициализацию и контроль технического обеспечения, связь между подсистемами, синхронизацию подсистем). Таким образом, жизненный цикл безопасности программного обеспечения в большинстве случаев тесно связан с жизненным циклом безопасности системы. В частности, спецификация требований к программному обеспечению является частью или непосредственно вытекает из спецификации системы и проекта системы.

Хотя верификация новых компонентов программного обеспечения несомненно является частью жизненного цикла безопасности программного обеспечения, часто нет отдельной и четко установленной границы между интеграцией программного обеспечения и интеграцией системы. Поэтому в настоящем стандарте интеграция программного обеспечения рассматривается как часть интеграции системы. Валидация программного обеспечения также рассматривается как часть валидации системы.

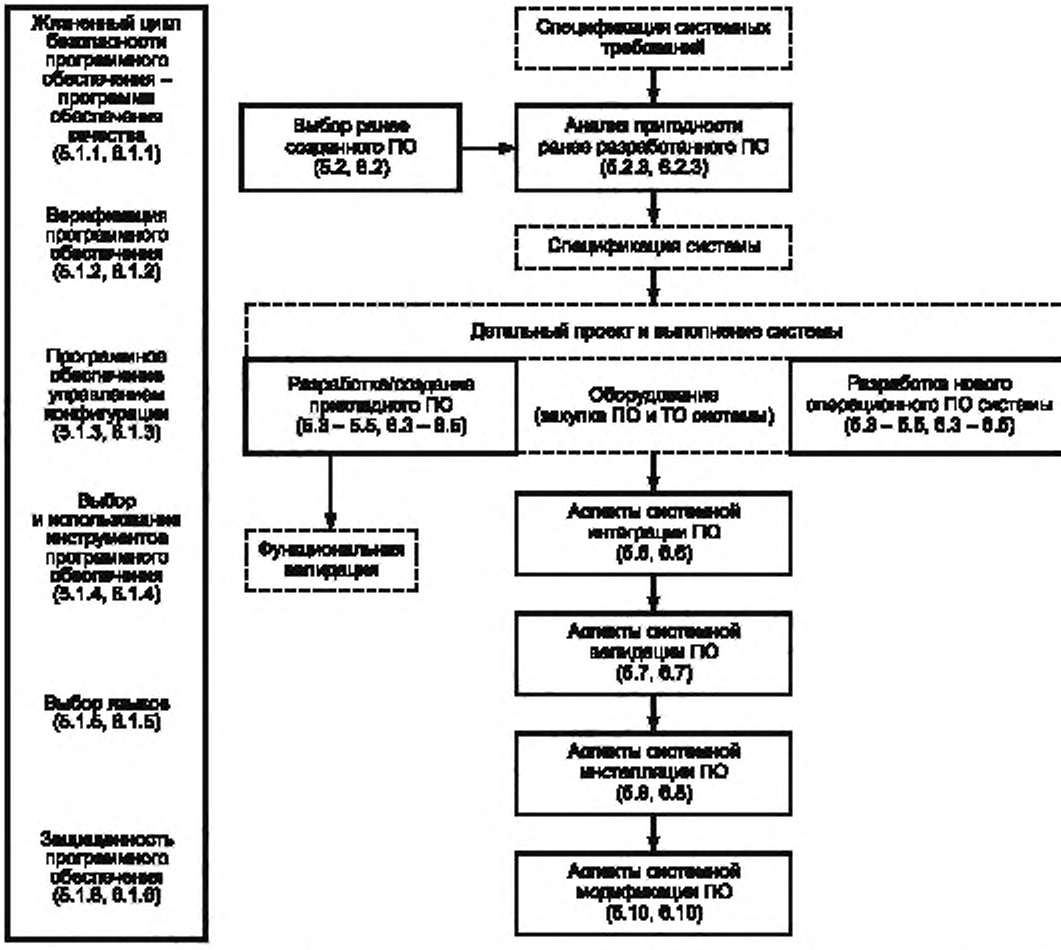



Рисунок 3 — Схема мероприятий, относящихся к программному обеспечению в жизненном цикле системы безопасности (блоки, выделенные пунктиром, относятся к мероприятиям, на которые требования настоящего стандарта не распространяются)

Соотношение между мероприятиями жизненного цикла безопасности системы и жизненного цикла безопасности программного обеспечения показано на рисунках 2 и 3.

Необходимо отметить, что, хотя МЭК 61513 устанавливает два различных пути создания нового программного обеспечения (прикладное ПО и операционное ПО системы, см. рисунки 2 и 3), в настоящем стандарте требования к созданию нового программного обеспечения объединены в четыре группы:

5.5.1 и 6.5.1 — предусматривают требования, которые являются применимыми независимо от того, какая техника создания используется;

5.5.2 и 6.5.2 — предусматривают требования, специфичные для конфигурации ранее разработанного программного обеспечения и устройств, содержащих программное обеспечение, в частности, для установления параметров и других данных конфигурации;

5.5.3 и 6.5.3 — предусматривают требования, специфичные для разработки и проверки программного обеспечения, написанного на проблемно-ориентированных языках;

5.5.4 и 6.5.4 — предусматривают требования, специфичные для реализации и верификации программного обеспечения, написанного на универсальных языках.

Блоки схемы, обозначенные на рисунке 3 как «разработка/создание прикладного программного обеспечения» и «разработка нового операционного программного обеспечения системы», представляют большую и существенную часть жизненного цикла безопасности программного обеспечения. Более под-

робно деятельность, осуществляемая между составлением спецификации требований к ПО и валидацией системы с четким представлением трех различных путей реализации проекта программного обеспечения (конфигурация ранее разработанного программного обеспечения и устройств, использование проблемно-ориентированных языков и использование языков общего назначения) показана на рисунке 4.

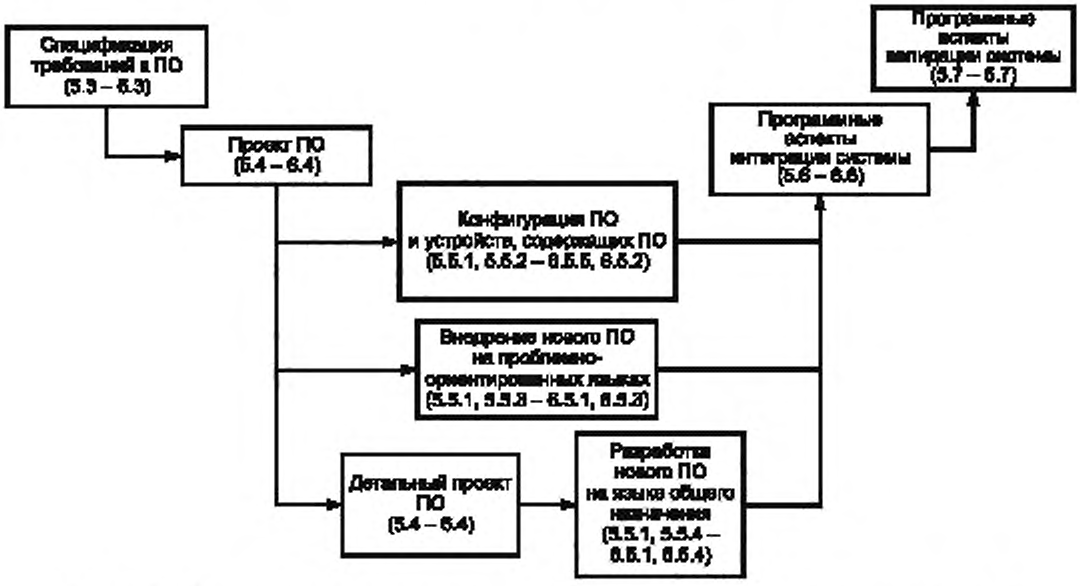



Рисунок 4 — Мероприятия жизненного цикла безопасности программного обеспечения по МЭК 62138

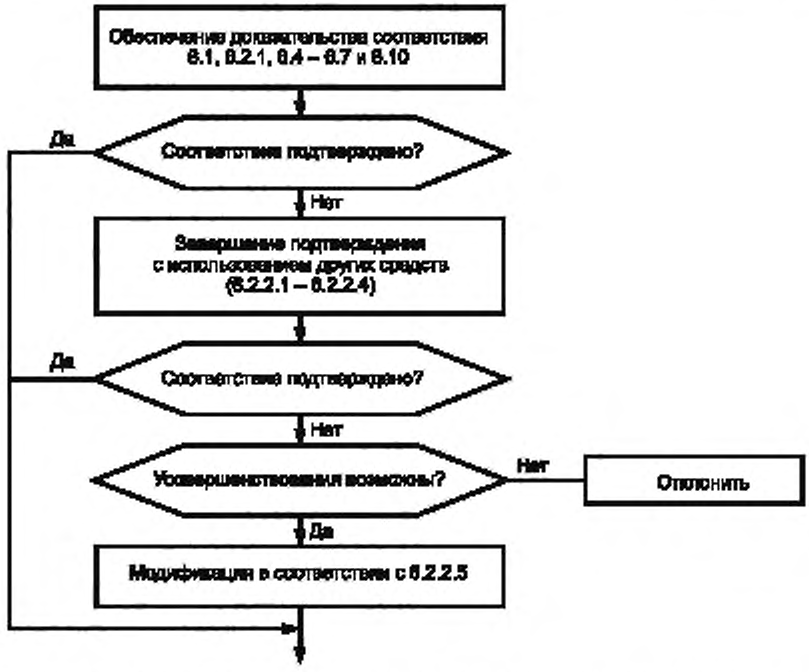



Рисунок 5 — Алгоритм проверки корректности ранее разработанного программного обеспечения для СКУ класса безопасности 2

Еще одним мероприятием особой важности жизненного цикла безопасности программного обеспечения является выбор ранее разработанного программного обеспечения, поскольку этот тип программного обеспечения обычно представляет существенную часть завершенного и интегрированного программного обеспечения. Более подробно алгоритм проверки корректности ранее разработанного программного обеспечения для СКУ класса безопасности 2 представлен на рисунке 5.

### 4.4 Принципы градации

Вследствие распределения функций в отношении безопасности по категориям А, В и С соответствующая градация была введена и для требований к программному обеспечению СКУ, относящемуся к классам безопасности 1, 2 и 3.

Применение требований настоящего стандарта для класса безопасности 3 обеспечивает базовый уровень доверительности, соответствующий программному обеспечению СКУ, важных для безопасности. При этом руководствуются следующими принципами:

- наличие программы обеспечения качества;
- особое внимание уделяют тому, чтобы программное обеспечение:

1) вносило необходимый вклад в функции, важные для безопасности, и не оказывало на них отрицательного влияния,

2) соответствовало положениям спецификации требований программного обеспечения, которые устанавливают ограничения, важные для безопасности;

- обеспечение как можно более раннего информирования операторов СКУ об ошибках и отказах программного обеспечения, которые могут повлиять на функции, важные для безопасности, с тем чтобы можно было предпринять необходимые действия;
- наличие документально оформленных спецификаций требований к программному обеспечению, спецификаций проекта, спецификаций интеграции, спецификаций валидации и спецификаций модификации.

Для класса безопасности 2 в дополнение к принципам, уже изложенным для класса 3, в настоящем стандарте добавляют принципы:

- базирующееся на тестированиях и проекте обоснования того, что требование к характеристике, связанной с безопасностью (например, время реакции), будет выполнено для всех заданных условий;
- использование документации по безопасности для ранее разработанного программного обеспечения и для ранее разработанных устройств с встроенным программным обеспечением; цель такой документации состоит в том, чтобы обеспечить всю необходимую информацию для безопасного использования программного обеспечения или устройства; в частности, требование по представлению основанного на проекте обоснования эксплуатационных характеристик, связанных с безопасностью, устанавливает минимальный уровень необходимой информации;
- использование ранее разработанного программного обеспечения и устройств с встроенным программным обеспечением в соответствии с правилами, основанными на соответствующей документации по безопасности;
- конфигурирование и использование ранее разработанных устройств по типу «черного ящика» согласно правилам, обеспечивающим уменьшение влияния известных или ожидаемых типов отказов;
- обоснование корректности и функциональной пригодности ранее разработанного программного обеспечения и ранее разработанных устройств с встроенным программным обеспечением; процесс такого обоснования корректности использования ПО и устройств по классу безопасности 2 показан на рисунке 5;
- обширная и документально оформленная верификация детального проекта и реализации нового программного обеспечения; верификация может включать в себя анализ документации, анализ с помощью инструментальных программ и тестирования;
- более жесткие требования по верификации, управлению конфигурацией ПО, выбору и использованию инструментальных программ и языков программирования, защищенности и сохранению работоспособности при отказе отдельных элементов;
- четкие требования к простоте, ясности, точности, проверяемости, тестируемости и модифицируемости.

Если одно и то же требование применимо к обоим классам безопасности, то степень необходимого обоснования соответствия может зависеть от класса безопасности. В частности, для класса 3 эта степень может быть меньшей применительно к функциям, не определенным в качестве важных для безопасности и не подвергающих риску функции, определенные в качестве важных для безопасности.

Требования для программного обеспечения СКУ класса безопасности 1 приведены в МЭК 60880.

Требования и рекомендации настоящего стандарта приведены в разделе 5 (для класса 3) и 6 (для класса 2). Эти разделы могут применяться независимо друг от друга. Они имеют одинаковую структуру, так что подразделы 5.х.у и 6.х.у устанавливают требования и рекомендации по одному типу. Требования раздела 6, не идентичные требованиям раздела 5 (дополнительные или модифицированные), выделены курсивом. Все требования и рекомендации двух подразделов структурированы и пронумерованы, и те из них, которые относятся к одной проблеме, имеют одинаковый номер. Все другие пункты являются справочными.

Целью настоящего стандарта является не установление конкретного комплекта документации, а скорее определение информации, которая должна быть оформлена документально. Конкретные иерархия и формат принятой документации могут изменяться при условии соблюдения принципов, изложенных в настоящем стандарте.

## 5 Требования к программному обеспечению СКУ, выполняющих функции категории С[1)]

### 5.1 Общие требования

#### 5.1.1 Жизненный цикл безопасности программного обеспечения. Обеспечение качества программного обеспечения

Пункт 6.2.1 МЭК 61513 предусматривает требования к обеспечению качества на уровне СКУ. Данный подраздел содержит следующие специфичные или особо важные дополнительные требования к программному обеспечению:

1 Разработка программного обеспечения должна проводиться в соответствии с жизненным циклом безопасности программного обеспечения. Положения этого жизненного цикла безопасности программного обеспечения должны быть определены в программе обеспечения качества.

Программа обеспечения качества может быть частью программы обеспечения качества системы или отдельной программой обеспечения качества программного обеспечения.

2 Если для программного обеспечения используется отдельная программа обеспечения качества, то она должна быть совместима с программой обеспечения качества системы. В этих двух программах необходимо учитывать применимые требования 6.2.1 МЭК 61513.

3 Этап разработки жизненного цикла безопасности программного обеспечения в программе обеспечения качества должен быть разделен на конкретные мероприятия. Эти мероприятия должны включать в себя действия, необходимые для достижения требуемого качества программного обеспечения, и обеспечивать объективное свидетельство того, что это качество достигнуто.

4 В спецификации мероприятий должны быть указаны:

- их цели;
- их отношения и взаимосвязь с другими мероприятиями;
- исходные данные и результаты;
- организация мероприятий и связанная с ними ответственность.

Должны быть также указаны необходимое содержание и свойства исходных данных и результатов.

5 Программа обеспечения качества должна содержать требование о том, чтобы выполнение каждого мероприятия возлагалось на компетентных лиц, обеспеченных соответствующими ресурсами.

6 Программа обеспечения качества должна содержать требование о том, чтобы изменения в уже утвержденной документации были определены, проанализированы и утверждены уполномоченными на это лицами.

7 Программа обеспечения качества должна содержать требование о том, чтобы методы, языки, инструменты, правила и используемые стандарты были определены, оформлены документально и изучены и усвоены соответствующими лицами.

8 Программа обеспечения качества должна содержать требование о том, чтобы в случае использования нескольких методов, языков, инструментов, правил и/или стандартов было ясно, какие из них должны использоваться для каждого мероприятия.

9 Программа обеспечения качества должна содержать требование о том, чтобы специфичные для проекта термины, выражения, сокращения и условные обозначения были четко определены.

10 Программа обеспечения качества должна содержать требование о том, чтобы возникающие проблемы были отслежены и решены.

---

[1] Нумерация пунктов в подразделениях соответствует нумерации раздела 6.

11 Программа обеспечения качества должна содержать требование регистрации результатов ее применения. В частности, должно содержаться требование регистрации результатов верификации и анализа вместе с областью их проведения, а также регистрации полученных заключений и согласованных решений. Любое отклонение от программы обеспечения качества должно быть обосновано и оформлено документально.

Дополнительные руководства по обеспечению качества программного обеспечения приведены в ИСО 9000-3.

**5.1.2 Верификация**

1 План верификации должен определять область верификации программного обеспечения и предусматривать необходимые мероприятия.

2 Верификация и анализ должны быть выполнены в соответствии с документально оформленными положениями. В частности, на этапах жизненного цикла безопасности программного обеспечения, указанных в плане верификации, должны быть верифицированы результаты мероприятий, обозначенных в плане верификации и подтверждающих, что:

- результаты находятся в рамках управления конфигурацией;
- для мероприятий имеются точно определенные исходные данные и результаты мероприятий согласуются с этими исходными данными;
- мероприятия реализуют указанные для них цели, а их результаты обладают требуемыми содержанием и свойствами и соответствуют каждому из согласованных решений;
- результаты являются ясными, точными и своевременными;
- результаты согласуются с каждым из применимых к ним правил;
- результаты согласуются с применимыми для них требованиями данного стандарта.

«Точно определенный» означает то, что текст понятен и не является двусмысленным. «Ясный» означает то, что лица, которые должны читать документ, могут полностью понять его без чрезмерных усилий, даже если они ранее не участвовали в проекте, при условии, что они обладают необходимыми знаниями. «Точный» означает, что двусмысленности отсутствуют.

Объем мероприятий по верификации и анализу может зависеть от масштаба и характера программного обеспечения, масштаба и характера результатов, подвергаемых верификации и анализу, а также от применяемых методов и инструментальных программ. Объем мероприятий может быть уменьшен для конкретных требований, которые не определены в качестве важных для безопасности (см. 5.3.3, перечисление 6) и не могут отрицательно влиять на функции, определенные в качестве важных для безопасности.

3 Верификация результатов мероприятий должна проводиться компетентными лицами, не принимавшими участия в этих мероприятиях. В их число должны входить представители структур, связанных с использованием этих результатов, а также, при необходимости, другие эксперты.

Это не означает, что лицо, являющееся автором одного документа, не может верифицировать другой документ.

4 Верификации должны подвергаться: спецификация требований к программному обеспечению, спецификация проекта программного обеспечения и план валидации программного обеспечения.

**5.1.3 Управление конфигурацией**

Подпункт 6.2.1.2 МЭК 61513 содержит требования по управлению конфигурацией на уровне СКУ. В нем предусмотрены следующие дополнительные требования, специфичные или особенно важные для программного обеспечения:

1 Управление конфигурацией программного обеспечения должно быть выполнено согласно положениям плана управления конфигурацией или программы обеспечения качества. Эти положения должны согласовываться с положениями управления конфигурацией на уровне системы.

2 Управление конфигурацией должно применяться к элементам, связанным с правильностью программного обеспечения. В плане управления конфигурацией должно быть определено, какие элементы программного обеспечения или какие типы элементов программного обеспечения должны находиться под управлением конфигурацией. В частности, в него должны входить:

- ключевые документы жизненного цикла безопасности (в особенности документы, требующие верификации);
- компоненты программного обеспечения, необходимые для построения рабочей программы, и сама рабочая программа;
- инструментальные программы, влияющие на правильность программного обеспечения и/или проекта системы.

3 В плане управления конфигурацией должны быть определены технические средства для идентификации элементов программного обеспечения и их версий, находящихся под управлением конфигурацией.

4 План управления конфигурацией должен обеспечивать однозначное определение версии программного обеспечения, используемой в данной версии системы или оборудования, и версии элементов, которые вместе составляют данную версию программного обеспечения.

**5.1.4 Выбор и использование инструментальных программ**

Инструментальные программы могут играть важную роль в предотвращении внесения дефектов в программное обеспечение или проект системы, а также в обнаружение уже существующего дефекта. В частности, инструментальные программы могут помочь в создании проекта архитектуры СКУ и нового прикладного программного обеспечения или автоматизировать это создание.

1 Инструментальные программы должны способствовать тем этапам разработки, которые обеспечивают правильность программного обеспечения и проекта системы.

Обычно предпочтительно уделить внимание не только качеству и использованию отдельных инструментальных программ, но и их совместимости с другими инструментальными программами, с тем чтобы собранные вместе инструментальные программы составляли взаимосвязанный набор. В целом, предпочтительней использовать известную инструментальную программу с обширным опытом эксплуатации вместо инструментальной программы без опыта эксплуатации, хотя каждый случай требует отдельного рассмотрения с учетом его преимуществ.

2 Комплексы оборудования, используемые для создания СКУ, рекомендуется связывать с инструментальными программами, снижающими риск внесения дефектов в новое прикладное программное обеспечение.

Эти инструментальные программы обычно включают в себя поддержку проблемно-ориентированных языков, позволяя операторам станции и системы устанавливать или верифицировать прикладные функции. Другими существенными задачами для таких инструментальных программ могут быть анимация, генерация кодов и помощь в определении функционального набора тестовых примеров.

3 Комплексы оборудования, используемые при разработке СКУ, должны быть связаны с инструментальными программами, что может снизить риск появления дефектов в конфигурации их ранее разработанного программного обеспечения и проекта системы.

Такие инструменты могут, например, помочь проектировщикам системы в:

- организации системы в виде соответствующего набора связанных подсистем;
- распределении прикладных функций между подсистемами;
- конфигурировании подсистем, их коммуникаций и операционных систем;
- обеспечении необходимых ресурсов для всех режимов работы системы;
- учете существующих ограничений при проектировании и реализации, в особенности обеспечивающих корректность и устойчивость системы.

4 Программа обеспечения качества должна точно определять инструментальные программы, которые могут повлиять на корректность программного обеспечения и/или проекта системы.

5 Для таких инструментальных программ должны быть предусмотрены эксплуатационные документы, обеспечивающие их использование по назначению.

7 Рекомендуется, чтобы были представлены свидетельства качества тех инструментальных программ, которые могут внести дефекты в программное обеспечение или проект системы, а также свидетельства их способности приводить к правильным результатам.

Примерами таких инструментальных программ являются генераторы кодов и компиляторы.

Свидетельство качества инструментальных программ и их способности приводить к правильным результатам может базироваться на опыте эксплуатации, сертификации инструментальной программы, сертификации поставщиков инструментальных программ для соответствующей деятельности, гарантии применения соответствующих процессов разработки инструментальной программы и/или ее тестирований.

**5.1.5 Выбор языков**

1 Используемые для создания программного обеспечения языки (проблемно-ориентированные или универсальные) должны иметь точные и документально оформленные синтаксис и семантику.

2 По возможности, предпочтение должно отдаваться проблемно-ориентированным языкам.

4 Если для создания одной рабочей программы используется более одного языка, интерфейсы между языками должны быть оформлены документально.

Интерфейс между языками включает в себя схемы передачи аргумента и представление структур данных.

6 Используемые универсальные языки должны поддерживать статический ввод переменных.

Прямой и статический ввод переменных предпочтителен по сравнению с косвенным или динамическим вводом.

#### 5.1.6 Защищенность

Цель защищенности состоит в обеспечении необходимой уверенности в том, что неуполномоченные лица и системы не смогут модифицировать программное обеспечение и его данные и не будут иметь доступа к функциям системы, в то время как такая возможность будет предоставлена для уполномоченных лиц и систем. Пункты 5.4.2 и 6.2.2 МЭК 61513 предусматривают требования к защищенности на уровне архитектуры СКУ и индивидуальной системы. В настоящем подразделе предусмотрены следующие дополнительные требования, специфичные или особенно важные для программного обеспечения:

1 Должен быть проведен и документально оформлен анализ угроз защищенности и уязвимости в отношении аспектов программного обеспечения СКУ. В нем следует учитывать необходимые этапы жизненного цикла безопасности системы и программного обеспечения. В этом анализе следует определить требования, касающиеся защиты, доступности, конфиденциальности и целостности данных и функций.

В анализ могут быть включены:

- идентификация защищенности ключевых данных и функций;
- идентификация и подтверждение права доступа персонала;
- управление защищенностью доступа к ключевым данным и функциям;
- управление защищенностью ключевых данных и функций;
- оперативный контроль действий персонала, связанных с защищенностью.

2 Разработка программного обеспечения должна быть выполнена согласно положениям программы обеспечения защищенности или программы обеспечения качества. Эти требования должны быть основаны на результатах анализа угроз и уязвимостей и согласованы с требованиями 5.4.2 и 6.2.2 МЭК 61513.

3 В ответственных случаях программное обеспечение должно иметь такие конфигурацию и параметры, которые позволят избежать его лишней уязвимости.

4 В план следует включать положения по оценке эффективности осуществленных решений.

### 5.2 Выбор ранее разработанного программного обеспечения

Подпункт 6.1.2.1 МЭК 61513 предусматривает общие требования при выборе ранее разработанных компонентов (не обязательно компонентов программного обеспечения). Настоящий подраздел вводит следующие дополнительные требования, специфичные или особенно важные для программного обеспечения.

#### 5.2.1 Эксплуатационные документы

5.2.1.1 Цели

1 Ранее разработанное программное обеспечение должно сопровождаться документами, содержащими необходимую информацию для использования этого программного обеспечения в СКУ.

В настоящем стандарте сопроводительный документ (пакет документов), содержащий(х) необходимую информацию для функционирования СКУ, назван «эксплуатационные документы». Если ранее разработанное программное обеспечение является частью оборудования или комплекса оборудования, то такие документы могут быть частью эксплуатационных документов на оборудование или семейство оборудования.

5.2.1.2 Содержание

1 Эксплуатационные документы должны включать в себя описание:

- предусмотренных функций;
- интерфейсов с приложениями;
- ролевых имен, типов, форматов, диапазонов и пределов входных, выходных и исключающих сигналов, параметров и данных конфигурации (если присвоены);
- различных режимов работы и соответствующих условий перехода;
- любых ограничений, относящихся к использованию ранее разработанного программного обеспечения.

3 Там, где это необходимо, эксплуатационные документы должны также предоставлять информацию о характеристиках функций (например, в виде времени срабатывания).

Функции, интерфейсы и характеристики могут зависеть от режима работы, значений параметров, данных конфигурации и условий, предусмотренных для программного обеспечения.

5.2.1.3 Свойства

1 Эксплуатационные документы должны быть четкими, не допускающими двоякой интерпретации.

**5.2.2 Свидетельство корректности**

5.2.2.1 Общие требования

1 Должна быть подтверждена корректность ранее разработанного программного обеспечения в отношении к его эксплуатационным документам.

Обычно подтверждение является качественным, т.к. общепризнанных методов количественной оценки не существует. Для ранее разработанного программного обеспечения подтверждение может базироваться на таких различных типах свидетельств, как, например:

- свидетельство соответствия всем требованиям или части требований 5.1, 5.2.1, 5.4, 5.5, 5.6, 5.7 и 5.10 (если применимы);
- специфичные для проекта дополнительные тестирования;
- применимый и надежный опыт эксплуатации;
- сертификация, проводимая уполномоченными органами.

Доверие к ранее разработанному программному обеспечению достигается легче, если оно может использоваться только ограниченным числом способов и/или если проект СКУ и его программного обеспечения предусматривает четкий набор условий их использования.

5.2.2.2 Дополнительные тестирования

Для класса безопасности 3 нет необходимости в требованиях по данному разделу.

5.2.2.3 Эксплуатационный опыт

Для класса безопасности 3 нет необходимости в требованиях по данному разделу.

5.2.2.4 Сертификация

Для класса безопасности 3 нет необходимости в требованиях по данному разделу.

5.2.2.5 Модификация

Для класса безопасности 3 нет необходимости в требованиях по данному разделу.

**5.2.3 Функциональная пригодность**

Для класса безопасности 3 нет необходимости в требованиях по данному разделу.

**5.2.4 Выбор и использование специализированных устройств с встроенным программным обеспечением**

Устройства по типу «черный ящик» с встроенным программным обеспечением могут использоваться в СКУ при следующих условиях:

1 Программное обеспечение устройства должно быть интегрировано так, чтобы оно не могло быть изменено пользователем и использоваться отдельно от остальной части устройства.

2 Число функций устройства, его потенциал для конфигурации и объем его интерфейсов и взаимодействий с остальной частью СКУ должны быть ограничены так, чтобы тестирования могли охватить все функции.

3 Должно быть предоставлено свидетельство того, что данное устройство соответствует требованиям 5.2.1 и 5.2.2.

4 Должно быть представлено свидетельство того, что конфигурация и использование устройства в СКУ соответствуют требованиям 5.4, 5.5.1 и 5.5.2.

**5.3 Спецификация требований к программному обеспечению**

Настоящий подраздел завершает и уточняет требования подпункта 6.1.2.3 МЭК 61513.

**5.3.1 Цели**

1 Требования к программному обеспечению СКУ должны быть определены и оформлены документально.

В настоящем стандарте соответствующий документ или пакет документов назван «Спецификация требований к программному обеспечению». Его цель состоит в определении задач программного обеспечения без определения путей их решения. Однако ограничивающие факторы проекта и разработки, вероятно, придется определить, если это потребуется при рассмотрении проекта СКУ или архитектуры СКУ.

3 Спецификация требований к программному обеспечению должна быть такой, чтобы:

- она способствовала уверенности в корректности проекта СКУ;
- было показано соответствие СКУ требованиям МЭК 61513.

Требования, относящиеся к спецификации требований к программному обеспечению, содержатся главным образом в 6.1.1.2, 6.1.1.3, 6.1.1.4, 6.1.2.2, 6.1.2.4 и 6.1.3 МЭК 61513.

4 Спецификация требований к программному обеспечению должна быть основой проекта программного обеспечения, валидации программного обеспечения и возможных модификаций программного обеспечения.

**5.3.2 Исходная информация**

1 Ссылки на другие документы, имеющиеся в спецификации требований к программному обеспечению, должны быть четкими и однозначными.

**5.3.3 Содержание**

1 Спецификация требований к программному обеспечению должна определять:

- прикладные функции, которые должны быть снабжены программным обеспечением;
- различные режимы работы программного обеспечения и соответствующие условия переходов;
- интерфейсы и взаимодействие программного обеспечения с его окружающей средой (например, операторами, остальной частью системы СКУ, другими системами и оборудованием, с которыми оно взаимодействует или разделяет ресурсы), включая ролевые имена, типы, форматы, диапазоны и ограничения по входам и выходам;
- параметры программного обеспечения, которые (при необходимости) изменяются операторами во время эксплуатации, их ролевые имена, типы, форматы, диапазоны и ограничения, а также проверки, осуществляемые программным обеспечением в случае изменений этих параметров;
- требуемые рабочие характеристики (там, где это необходимо);
- указание на то, чего программное обеспечение не должно делать или избегать (там, где это необходимо);
- при необходимости, требования или допущения, устанавливаемые программным обеспечением к его окружению.

2 Спецификация требований к программному обеспечению должна также устанавливать условия (например, требуемую нагрузку), создаваемые для программного обеспечения окружающей средой, особенно для наихудшего случая.

Требования к функциям, интерфейсам и характеристикам могут зависеть от режима работы, значений параметров, данных конфигурации и условий, создаваемых для программного обеспечения.

3 Спецификация требований к программному обеспечению должна определять режимы его работы при обнаружении ошибок или отказов. Если для системы СКУ требуются периодические тестирования, то спецификация требований к программному обеспечению должна также определять требования к режиму выполнения таких тестирований.

4 Спецификация требований к программному обеспечению должна содержать требования к качеству программного обеспечения и указывать на необходимые ограничения проекта программного обеспечения и его работы с целью обеспечения его корректности и работоспособности.

Например, в спецификацию требований к программному обеспечению могут входить ограничения, направленные на:

- обеспечение уверенности в правильности программного обеспечения и проекта системы (например, пределы, устанавливаемые при использовании динамически размещенных ресурсов, таких как память, вычислительная мощность, полоса пропускания канала связи, ресурсы операционной системы);
- увеличение способности программного обеспечения и СКУ быть устойчивым к дефектам, обнаруживать ошибки и отказы и оповещать о них, работать в указанных режимах и восстанавливаться после отказов;
- обеспечение уверенности в том, что ошибки операторов и отказы других систем или оборудования, с которыми программное обеспечение взаимодействует или использует общие ресурсы, не будут приводить к недопустимым результатам.

5 Спецификация требований к программному обеспечению должна определять задачи программного обеспечения по своевременному информированию операторов об ошибках или отказах, касающихся функций СКУ, идентифицированных как «важные для безопасности». Информация, предоставляемая операторам, должна позволять им предпринимать соответствующее действие.

6 Спецификация требований к программному обеспечению должна определять функции и требования категории безопасности С.

**5.3.4 Свойства**

Для класса безопасности 3 нет необходимости в требованиях по данному разделу.

### 5.4 Проект программного обеспечения

#### 5.4.1 Цели

1 Проект программного обеспечения должен быть оформлен документально. В документации должен быть приведен обзор организации и функционирования программного обеспечения.

В настоящем стандарте соответствующий документ или пакет документов назван «Спецификация проекта программного обеспечения». Если используется ранее разработанное программное обеспечение, то в спецификации проекта программного обеспечения может быть дана ссылка на соответствующий эксплуатационный документ.

3 Спецификация проекта программного обеспечения должна обеспечивать свидетельство того, что положения спецификации требований к программному обеспечению, важному для безопасности, приняты во внимание и будут удовлетворяться при всех указанных условиях.

5 Спецификация проекта программного обеспечения должна обеспечивать (в случае необходимости) устранение неблагоприятных побочных эффектов, связанных с ошибками и отказами программного обеспечения, до его возвращения к нормальному режиму работы.

7 Спецификация проекта программного обеспечения должна служить основой для реализации и интеграции программного обеспечения, а также возможных его модификаций.

#### 5.4.2 Исходные данные

1 Исходные данные для процесса проектирования программного обеспечения должны включать в себя спецификацию требований к программному обеспечению и эксплуатационные документы на ранее разработанное программное обеспечение.

Исходные данные могут также включать в себя другие документы, такие, например, как специфичные проектные ограничения и/или правила и стандарты, применимые к исходным данным.

#### 5.4.3 Содержание

1 Спецификация проекта программного обеспечения должна включать в себя спецификацию:

- полной организации программного обеспечения;
- полного функционирования программного обеспечения в условиях и режимах работы, требуемых в соответствии со спецификацией требований к программному обеспечению.

2 Полная организация должна обеспечить информацию относительно:

- четкой идентификации и конфигурации ранее разработанного программного обеспечения;
- распределения ресурсов, компонентов и задач программного обеспечения по подсистемам;
- распределения (под) функций и характеристик программного обеспечения по определенным для него задачам;
- главных внутренних интерфейсов, в частности интерфейсов между задачами программного обеспечения.

Полное функционирование должно обеспечивать информацию относительно:

- взаимодействия, протоколов связи и информационных потоков;
- установления последовательностей и временных ограничений;
- использования ресурсов;
- синхронизации, особенно при использовании разделенных ресурсов.

#### 5.4.4 Свойства

Для класса 3 безопасности нет необходимости в требованиях по данному разделу.

### 5.5 Реализация нового программного обеспечения

#### 5.5.1 Общие требования

Требования настоящего пункта применимы ко всему новому программному обеспечению, т.е. к конфигурации ранее разработанного программного обеспечения и программам, написанным на проблемно-ориентированных языках или универсальных языках:

1 Использование ранее разработанного программного обеспечения должно быть верифицировано на соответствие эксплуатационным документам и ограничениям, установленным спецификацией проекта программного обеспечения.

2 Процедуры, используемые для трансляции новых программ в рабочую программу, должны быть оформлены документально и верифицированы.

**5.5.2 Конфигурация программного обеспечения и устройств, содержащих программное обеспечение**

Требование настоящего пункта является специфичным для конфигурации настраиваемого программного обеспечения. Такое программное обеспечение может быть ранее разработанным или новым. Требование также применимо к конфигурации настраиваемых устройств с встроенным программным обеспечением. Тем не менее, если данные конфигурации представляют собой выполняемую программным обеспечением или системой обработку (т.е. когда они фактически являются программой), то применяются требования 5.5.3.

1 Конфигурация настраиваемого программного обеспечения и устройств с встроенным программным обеспечением должна быть оформлена документально.

**5.5.3 Реализация с помощью проблемно-ориентированных языков**

Требования настоящего пункта специфичны для программ, написанных на проблемно-ориентированных языках. В общем случае для отражения всей или части спецификации требований к программному обеспечению или спецификации проекта программного обеспечения могут использоваться проблемно-ориентированные форматы (такие, как логические диаграммы или блок-схемы функций). При этом необходимы только ограниченный детальный проект и некоторые усилия по реализации преобразования спецификации в программы, которые могут быть автоматически транслированы в рабочую программу.

1 Те части спецификации требований к программному обеспечению и/или спецификации проекта программного обеспечения, которые используются для получения рабочей программы с помощью средств автоматизации, должны рассматриваться как программы, написанные на проблемно-ориентированных языках.

2 Программы, написанные на проблемно-ориентированных языках, которые связаны с функциями, важными для безопасности, должны быть верифицированы на функциональную корректность и согласованность.

**5.5.4 Реализация с помощью универсальных языков**

Требование настоящего пункта является специфичным для программ, написанных на универсальных языках.

4 Программы, написанные на универсальных языках, должны соответствовать документально оформленным правилам, обеспечивающим четкость, возможность модификации и тестируемость.

Набор правил может быть специфичным для языка или набора программ.

**5.6 Программные аспекты интеграции системы**

Интеграция программного обеспечения рассматривается как часть интеграции системы. Настоящий подраздел дополняет 6.1.4 и 6.2.3 МЭК 61513, устанавливая следующие дополнительные требования, специфичные или особенно важные для программного обеспечения:

1 Интеграция программного обеспечения и/или его анализ должны показать, что интегрированные система и программное обеспечение:

- соответствуют тем положениям проекта, которые обеспечивают выполнение указаний спецификации требований к программному обеспечению, определенных в качестве важных для безопасности;
- соответствуют ограничениям, изложенным в спецификации требований к программному обеспечению, касающимся его корректности и работоспособности.

3 Интеграция программного обеспечения должна быть выполнена согласно положениям плана интеграции системы или плана интеграции программного обеспечения.

4 Должны быть составлены отчеты по выполнению плана, используемого для интеграции программного обеспечения, например, результаты тестирований. В случае необходимости проведения модификации программного обеспечения или системы должна иметься возможность повторения всех или части интеграционных тестирований для оценки степени возможных изменений в работе.

**5.7 Программные аспекты валидации системы**

Цель валидации программного обеспечения состоит в обеспечении соответствия интегрированного программного обеспечения функциональной спецификации, а также спецификаций характеристик и интерфейса, определяемых требованиями к СКУ. Таким образом, валидация программного обеспечения рассматривается как часть валидации системы. Настоящий подраздел дополняет 6.1.5 и 6.2.4 МЭК 61513, выдвигая следующие дополнительные требования, специфичные или особенно важные для программного обеспечения:

1 Валидация программного обеспечения должна показать, что в завершенной системе интегрированное программное обеспечение соответствует требованиям к функциональности, характеристикам и интерфейсу, определенным в качестве важных для безопасности. Сюда должно быть включено обоснование того, что:

- установленные функции программного обеспечения, важные для безопасности, правильно выполняются, когда их аргументы и исходные данные находятся в диапазонах, указанных в спецификации требований к программному обеспечению, в условиях применения, указанных в этой спецификации;

- важные для безопасности функции системы, в реализацию которых программное обеспечение вносит свой вклад, выполняются правильно в условиях применения, определенных в спецификации требований к системе;

- программное обеспечение обеспечивает защиту от ошибок операторов и отказов других систем и оборудования, как это требуется в спецификации требований к программному обеспечению;

- технические данные станции, используемые или интегрированные в СКУ для осуществления функций, важных для безопасности, правильны; в частности, валидация программного обеспечения должна показать, что эти данные правильно описывают и адресуются к системам и оборудованию станции, с которыми программное обеспечение взаимодействует или разделяет ресурсы.

Примечание — Если представлено соответствующее обоснование, то для некоторых аспектов тестирований при валидации допускается использовать техническое обеспечение на платформе, идентичной штатной платформе.

Условия использования функций, важных для безопасности, могут включать в себя параллельное выполнение функций, не важных для безопасности.

2 Валидация программного обеспечения должна быть выполнена в соответствии с положениями плана валидации системы. В противном случае она должна быть выполнена в соответствии с положениями плана валидации программного обеспечения.

3 План валидации программного обеспечения должен устанавливать необходимые действия по валидации, а также показывать, что все указания спецификации требований к программному обеспечению, определенные в качестве важных для безопасности и касающиеся функциональности характеристик и интерфейса, правильно учтены при выполнении этих действий. План должен также определить основные этапы валидации программного обеспечения (например, этап до размещения на штатное место, за которым следует этап на штатном месте) и соответствующие средства, методы и инструменты, которые при этом должны быть использованы.

4 Должны быть составлены отчеты по выполнению плана, используемого для валидации программного обеспечения. В случае необходимости проведения модификации программного обеспечения или системы должна быть возможность повторения всех или части валидационных тестирований для оценки степени возможных изменений в работе. Рекомендуется, чтобы результаты валидации программного обеспечения были проверены лицами, компетентными в данной области, но не участвовавшими непосредственно в процессе валидации.

5 В этих отчетах должны быть документально оформлены конфигурация программного обеспечения и конфигурация окружающей среды при проведении валидации (например, техническое обеспечение, инструментальные программы, если использовались).

6 Группа, которая составляет план валидации программного обеспечения, должна включать в себя по крайней мере одного человека, не участвовавшего в разработке проекта и его реализации.

### 5.8 Инсталляция программного обеспечения на штатном месте

Подраздел 6.1.6 МЭК 61513 содержит требования относительно установки СКУ системы на штатном месте. В настоящем подразделе приводятся следующие дополнительные требования, специфичные или особенно важные для программного обеспечения:

1 Процедура инсталляции программного обеспечения на штатном месте должна быть оформлена документально. Она должна обеспечивать гарантированную инсталляцию правильной и полной версии программного обеспечения.

2 В процедуру инсталляции программного обеспечения на штатном месте должны быть включены и детально описаны проверки и тестирования на штатном месте, которые должны проводиться до начала полномасштабной эксплуатации СКУ. В частности, должно быть верифицировано выполнение условий, необходимых для правильной работы программного обеспечения.

Например, эти условия могут касаться технического обеспечения, на котором установлено программное обеспечение, или других систем, с которыми программное обеспечение взаимодействует или разделяет ресурсы.

### 5.9 Протоколы отклонений от нормы

1 Если обнаружена неожиданная, очевидно неправильная, необъяснимая или ненормальная работа программы после ее принятия в эксплуатацию, то следует составить протокол отклонения от нормы.

2 В протоколе отклонения от нормы указывают подробности работы программы, конфигурацию программного обеспечения и технического обеспечения и управляющих действий во время анормальной работы. В протоколе следует указать его обозначение, его составителя, а также место и время составления.

После краткого обзора протокола может быть добавлено его обозначение, чтобы подчеркнуть действительность протокола.

3 Протоколы отклонения от нормы следует анализировать. Возникшие проблемы оформляют документально, отслеживают и решают.

### 5.10 Модификация программного обеспечения

Решение о проведении модификации программного обеспечения зависит от его влияния на СКУ. Поэтому такие решения подчинены требованиям 6.1.7 и 6.3.6 МЭК 61513. В настоящем подразделе представлены следующие дополнительные требования, специфичные или особенно важные для программного обеспечения.

1 Модификации программного обеспечения должны разрабатываться в соответствии с требованиями 5.1, 5.2, 5.3, 5.4 и 5.5. Модификации должны инсталлироваться в штатную аппаратуру в соответствии с требованиями 5.8.

2 Интеграцию и валидацию модификаций программного обеспечения следует проводить в соответствии с 5.6 и 5.7. Когда объем модификации не требует выполнения требований этих двух подпунктов в полном объеме, интеграция модифицированного программного обеспечения должна быть выполнена согласно регрессивному плану интеграции программного обеспечения, а валидация — согласно регрессивному плану валидации программного обеспечения. Полнота и подробность этих планов должны быть обоснованы с учетом объема любых модификаций, выполненных в спецификации требований к программному обеспечению и спецификации проекта программного обеспечения. Должны быть составлены отчеты по применению этих планов.

4 Модификации программного обеспечения должны быть оформлены документально в полном объеме. В частности, все связанные с программным обеспечением документы, на которые повлияли модификации, должны быть обновлены.

7 Должна быть проведена оценка влияния модификации программного обеспечения на остальные части СКУ и на другие системы, с которыми программное обеспечение взаимодействует или разделяет ресурсы. Все необходимые меры должны быть предприняты для обеспечения правильной работы СКУ.

8 Должны быть оценены эффекты влияния на программное обеспечение модификаций в остальной части системы или других системах, с которыми это программное обеспечение взаимодействует или делит ресурсы. Должны быть предприняты все необходимые меры для обеспечения правильной работы СКУ.

## 6 Требования к программному обеспечению СКУ, выполняющих функции категории В

В разделе 6 представлены дополнительные требования к программному обеспечению СКУ, выполняющих функции категории В (т.е. системы класса безопасности 2); для того, чтобы упростить применение настоящего стандарта, данный раздел повторяет соответствующие требования для класса безопасности 3; дополнительные или модифицированные требования в настоящем разделе выделены курсивом.

### 6.1 Общие требования

#### 6.1.1 Жизненный цикл безопасности программного обеспечения. Обеспечение качества программного обеспечения

Пункт 6.2.1 МЭК 61513 предусматривает требования к обеспечению качества на уровне СКУ. Данный подраздел содержит следующие специфичные или особенно важные дополнительные требования к программному обеспечению:

1 Разработка программного обеспечения должна проводиться в соответствии с жизненным циклом безопасности программного обеспечения. Положения этого жизненного цикла безопасности программного обеспечения должны быть определены в программе обеспечения качества.

Программа обеспечения качества может быть частью программы обеспечения качества системы или отдельной программой обеспечения качества программного обеспечения.

2 Если для программного обеспечения используется отдельная программа обеспечения качества, то она должна быть совместима с программой обеспечения качества системы. В этих двух программах необходимо учитывать применимые требования 6.2.1 МЭК 61513.

3 Этап разработки жизненного цикла безопасности программного обеспечения в программе обеспечения качества должен предусматривать выполнение конкретных мероприятий. Эти мероприятия должны включать в себя действия, необходимые для достижения требуемого качества программного обеспечения, и обеспечивать объективное свидетельство того, что это качество достигнуто.

4 В спецификации мероприятия должны быть указаны:

- цели;
- отношения и взаимодействия с другими мероприятиями;
- исходные данные и результаты;
- организация мероприятия и связанная с ним ответственность.

Должны быть также специфицированы содержание и свойства требуемых исходных данных и результатов.

5 Программа обеспечения качества должна содержать требование о том, чтобы выполнение каждого мероприятия возлагалось на компетентных людей, обеспеченных соответствующими ресурсами.

6 Программа обеспечения качества должна содержать требование о том, чтобы изменения в уже утвержденных документах были зафиксированы, проанализированы и утверждены уполномоченными для этого лицами.

7 Программа обеспечения качества должна содержать требование о том, чтобы методы, языки, инструменты, правила и используемые стандарты были зафиксированы, оформлены документально и изучены и усвоены соответствующими лицами.

8 Программа обеспечения качества должна содержать требование о том, чтобы в случае использования нескольких методов, языков, инструментов, правил и/или стандартов было ясно, какие из них должны использоваться для каждого мероприятия.

9 Программа обеспечения качества должна содержать требование о том, чтобы специфичные для проекта термины, выражения, сокращения и условные обозначения были четко определены.

10 Программа обеспечения качества должна содержать требование о том, чтобы возникающие проблемы были отслежены и решены.

11 Программа обеспечения качества должна содержать требование регистрации результатов ее применения. В частности, должно содержаться требование регистрации результатов верификации и анализа вместе с областью их проведения, а также о регистрации полученных заключений и согласованных решений. Любое отклонение от программы обеспечения качества должно быть обосновано и оформлено документально.

В ИСО 9000-3 приведены дополнительные руководящие указания по обеспечению качества программного обеспечения.

### 6.1.2 Верификация

1 План верификации должен определить область верификации программного обеспечения и предусмотреть необходимые мероприятия.

2 Верификация и анализ должны быть выполнены в соответствии с документально оформленными положениями. В частности, на этапах жизненного цикла безопасности программного обеспечения, указанных в плане верификации, должны быть верифицированы результаты мероприятий, обозначенных в плане верификации и подтверждающих, что:

- результаты находятся в рамках управления конфигурацией;
- для мероприятий имеются точно определенные исходные данные и результаты мероприятий согласуются с этими исходными данными;
- мероприятия реализуют указанные для них цели, а их результаты обладают требуемым содержанием и свойствами и соответствуют каждому из согласованных решений;
- результаты являются ясными, точными и своевременными;
- результаты согласуются с каждым из применимых к ним правил;
- результаты согласуются с применимыми для них требованиями настоящего стандарта.

«Точно определенный» означает, что текст признан понятным без каких-либо двусмысленностей. «Ясный» означает, что лица, которые должны читать документ, могут полностью понять его без чрезмерных усилий, даже если они ранее не участвовали в проекте, при условии, что они обладают необходимыми знаниями. «Точный» означает, что отсутствуют двусмысленности.

Перечень мероприятий по верификации и анализу может зависеть от масштаба и характера программного обеспечения, масштаба и характера результатов, подвергаемых верификации и анализу, а также от применяемых методов и инструментальных программ.

3 Верификация результатов мероприятий должна проводиться компетентными лицами, не принимавшими участия в этих мероприятиях. В их число должны входить представители структур, связанных с использованием этих результатов, а также (при необходимости) другие эксперты.

Это не означает, что лицо, являющееся автором одного документа, не может верифицировать другой документ.

4 Верификации должны подвергаться спецификация требований к программному обеспечению, спецификация проекта программного обеспечения и план валидации программного обеспечения.

*5 Правила проектирования и реализации должны быть верифицированы.*

### 6.1.3 Управление конфигурацией

Подпункт 6.2.1.2 МЭК 61513 содержит требования по управлению конфигурацией на уровне СКУ. В настоящем пункте предусмотрены следующие дополнительные требования, специфичные или особенно важные для программного обеспечения:

1 Управление конфигурацией для программного обеспечения должно быть выполнено согласно положениям плана управления конфигурацией или программы обеспечения качества. Эти положения должны быть согласованы с положениями управления конфигурацией на уровне системы.

2 Управление конфигурацией должно применяться к элементам, связанным с правильностью программного обеспечения. В плане управления конфигурацией должно быть определено, какие элементы программного обеспечения или какие типы элементов программного обеспечения должны находиться под управлением конфигурацией. В частности, в него должны входить:

- ключевые документы жизненного цикла безопасности (в особенности документы, требующие верификации);
- компоненты программного обеспечения, необходимые для построения рабочей программы, и сама рабочая программа;
- инструментальные программы, влияющие на правильность программного обеспечения и/или проекта системы.

3 В плане управления конфигурацией должны быть определены технические средства для аутентификации элементов программного обеспечения и их версий, находящихся под управлением конфигурацией.

4 План управления конфигурацией должен обеспечивать однозначное определение версии программного обеспечения, используемой в данной версии системы или оборудования, и версии элементов, которые вместе составляют данную версию программного обеспечения.

### 6.1.4 Выбор и использование инструментальных программ

Инструментальные программы могут играть важную роль в предотвращении внесения дефектов в программное обеспечение или проект системы, а также в обнаружении уже существующего дефекта. В частности, инструментальные программы могут помочь в создании проекта архитектуры СКУ и нового прикладного программного обеспечения или автоматизировать это создание.

1 Инструментальные программы должны способствовать тем этапам разработки, которые обеспечивают правильность программного обеспечения и проекта системы.

Обычно следует уделять внимание не только качеству отдельных инструментальных программ и методам их применения, но также их совместимости с другими инструментальными программами с тем, чтобы, собранные вместе, они составляли взаимосвязанный набор инструментальных программ. В целом, предпочтительней использовать известную инструментальную программу с обширным опытом эксплуатации вместо инструментальной программы без опыта эксплуатации, хотя каждый случай требует отдельного рассмотрения с учетом его преимуществ.

*2 Комплексы оборудования, используемые для создания СКУ, должны быть связаны с инструментальными программами, снижающими риск внесения дефектов в новое прикладное программное обеспечение.*

Эти инструментальные программы обычно включают в себя поддержку проблемно–ориентированных языков, позволяя операторам станции и системы устанавливать или верифицировать прикладные

функции. Другими существенными задачами для таких инструментальных программ могут быть анимация, генерация кодов и помощь в определении совокупности функциональных тестовых примеров.

3 Комплексы оборудования, используемые при разработке СКУ, должны быть связаны с инструментальными программами, что может снизить риск появления дефектов в конфигурации ранее разработанного программного обеспечения и в проекте системы.

Такие инструменты могут, например, помочь проектировщикам системы в:

- организации системы в виде подходящего набора связанных между собой подсистем;
- распределении прикладных функций между подсистемами;
- конфигурировании подсистем, их коммуникаций и операционных систем;
- обеспечении необходимых ресурсов для всех режимов работы системы;
- учете существующих ограничений при проектировании и реализации, в особенности обеспечивающих корректность и устойчивость системы.

4 Программа обеспечения качества должна точно определять инструментальные программы, которые могут повлиять на корректность программного обеспечения и/или проекта системы.

5 Для таких инструментальных программ должны быть предусмотрены эксплуатационные документы, обеспечивающие их использование по назначению.

*6 В программе обеспечения качества должны различаться инструментальные программы, которые могут внести ошибки в программное обеспечение или проект системы, и инструментальные программы, которые могут привести лишь к пропуску уже существующей ошибки.*

*Генераторы команд и компиляторы — примеры инструментальных программ первой категории, тогда как статические анализаторы кодов и генераторы тестовых команд — примеры инструментальных программ второй категории.*

*7 Инструментальные программы, которые могут внести ошибки в программное обеспечение или в проект системы, должны быть выделены и должны использоваться согласно документально оформленным процедурам и правилам, обеспечивающим снижение риска такого внесения. Должно быть обеспечено свидетельство их качества и способности производить правильные результаты. Их использование должно отслеживаться так, чтобы инструментальные программы, применяемые для генерации заданных элементов или информации, могли быть идентифицированы.*

*Свидетельство качества инструментальных программ и их способности давать правильные результаты может базироваться на опыте эксплуатации, сертификации инструментальной программы, сертификации ее поставщиков для соответствующей деятельности, гарантии применения соответствующих процессов создания инструментальной программы и/или ее тестирований. Строгость свидетельства может зависеть от условий использования инструментальной программы, глубины проверки выходных данных, вероятности ошибок, которые должны быть выявлены в инструментальных программах, и серьезности последствий необнаруженных ошибочных результатов. Наоборот, убедительное свидетельство (например, аттестация инструментальной программы согласно МЭК 60880) может использоваться в некоторых случаях в качестве замены верификации выходных данных.*

*8 Инструментальные программы, которые могут привести к невыявленным дефектам, уже существующим в программном обеспечении или проекте системы, следует отметить и использовать так, чтобы снизить этот риск. Их использование должно отслеживаться.*

*9 Если инструментальную программу или ее версию, потенциально способную внести дефект в программное обеспечение или проект системы, заменяют другой инструментальной программой или ее версией, то должны быть приняты разумные меры предосторожности для устранения влияния на правильность программного обеспечения и проекта системы.*

*Например, в дополнение к оценке качества и способности новой инструментальной программы производить правильные результаты должна быть оценена ее совместимость с предыдущей инструментальной программой.*

### 6.1.5 Выбор языков

1 Используемые для создания программного обеспечения языки (проблемно-ориентированные или универсальные) должны иметь точные и документально оформленные синтаксис и семантику.

2 По возможности, предпочтение должно отдаваться проблемно-ориентированным языкам.

3 *Машино-ориентированные универсальные языки низкого уровня (например, языки ассемблера) могут использоваться для отдельных программ, но это использование следует обосновать.*

4 Если для создания одной рабочей программы используется более одного языка, интерфейсы между языками должны быть оформлены документально.

Интерфейс между языками включает в себя схемы передачи аргумента и представление структур данных.

5 *Рекомендуется, чтобы используемые универсальные языки имели опции, облегчающие статический анализ программ с помощью инструментальных программ.*

6 В частности, используемые универсальные языки должны поддерживать статический ввод переменных. *Прямой и статический ввод переменных предпочтителен по отношению к косвенному или динамическому вводу.*

7 *Используемые языки и соответствующие им библиотеки рабочих программ должны обеспечивать предсказуемое поведение программы во время ее работы.*

*Например, прерывание рабочего режима программы для освобождения памяти в произвольные моменты времени обычно неприемлемо.*

### 6.1.6 Защищенность

Цель защищенности состоит в обеспечении необходимой уверенности в том, что неуполномоченные лица и системы не смогут модифицировать программное обеспечение и его данные и не будут иметь доступа к функциям системы, в то время как такая возможность будет предоставлена для уполномоченных лиц и систем. В пунктах 5.4.2 и 6.2.2 МЭК 61513 предусмотрены требования к защищенности на уровне архитектуры СКУ и индивидуальной системы. Настоящий подраздел предусматривает следующие дополнительные требования, специфичные или особенно важные для программного обеспечения:

1 Должен быть проведен и документально оформлен анализ угроз защищенности и уязвимости в отношении аспектов программного обеспечения СКУ. *Анализ должен учитывать необходимые этапы жизненного цикла безопасности системы и программного обеспечения. Этот анализ также должен определить требования, касающиеся защиты, доступности, конфиденциальности и целостности данных и функций.*

Сюда могут быть включены:

- идентификация защищенности ключевых данных и функций;
- идентификация и подтверждение права доступа персонала;
- управление защищенностью доступа к ключевым данным и функциям;
- управление защищенностью ключевых данных и функций;
- оперативный контроль действий персонала, связанных с защищенностью.

2 Разработка программного обеспечения должна быть выполнена согласно положениям программы обеспечения защищенности или программы обеспечения качества. Эти требования должны быть основаны на результатах анализа угроз и уязвимостей. Требования должны быть согласованы с требованиями 5.4.2 и 6.2.2 МЭК 61513.

3 В ответственных случаях программное обеспечение должно иметь такие конфигурацию и параметры, которые позволят избежать лишней его уязвимости.

4 *План должен включать в себя положения по оценке эффективности осуществленных решений.*

## 6.2 Выбор ранее разработанного программного обеспечения

Подпункт 6.1.2.1 МЭК 61513 устанавливает общие требования при выборе ранее разработанных компонентов (не обязательно компонентов программного обеспечения). Данный подраздел вводит дополнительные требования, специфичные или особенно важные для программного обеспечения.

### 6.2.1 Документация по безопасности

6.2.1.1 Цели

*1 Ранее разработанное программное обеспечение должно сопровождаться документацией, содержащей необходимую информацию для безопасного использования этого программного обеспечения в СКУ и обеспечивающей свидетельство соответствия требованиям 6.2.3 (функциональная пригодность) и 6.4 (проект программного обеспечения).*

*В настоящем стандарте соответствующий документ или пакет документов назван «Документация по безопасности». Если ранее разработанное программное обеспечение являлось частью оборудования или комплекса оборудования, то данная документация может быть частью документации по безопасности для оборудования или семейства оборудования.*

*Документация по безопасности может быть общей или специфичной для проекта. Она может включать в себя не только руководство пользователя, предоставленное поставщиком ранее разработанного программного обеспечения. Например, в ее состав может быть включена информация, полученная по результатам дополнительных тестирований, измерений и/или анализа, а также опыта эксплуатации.*

6.2.1.2 Содержание

1 Документация по безопасности должна включать в себя описание:

- предусмотренных функций;
- интерфейсов с приложениями;
- ролевых имен, типов, форматов, диапазонов и пределов входных, выходных и исключающих сигналов, параметров и данных конфигурации, если они присвоены;
- различных режимов работы и соответствующих условий перехода;
- любых ограничений, относящихся к использованию ранее разработанного программного обеспечения.

*2 В соответствующих случаях эти ограничения направлены на:*

*- обеспечение уверенности в правильности интегрированного программного обеспечения и проекта системы (например, границы использования динамически размещаемых ресурсов, таких как память, производительность, полоса пропускания коммуникаций, ресурсы операционной системы);*

*- повышение способности интегрированного программного обеспечения и СКУ регистрировать отказы, сигнализировать об их наличии и сохранять к ним устойчивость, переключаться на указанные режимы работы и восстанавливаться после отказов;*

*- обеспечение уверенности в том, что ошибки операторов и отказы других систем или оборудования, с которыми объединенное программное обеспечение взаимодействует или разделяет ресурсы, приведут к заданным режимам работы;*

*- гарантию того, что новое окружение ранее разработанного программного обеспечения обеспечит все необходимые ресурсы во всех условиях использования в СКУ.*

3 Там, где необходимо, документация по безопасности должна также предоставлять информацию о характеристиках функций (например, в виде времени срабатывания).

Функции, интерфейсы и характеристики могут зависеть от режима работы, значений параметров, данных конфигурации и условий, предусмотренных для программного обеспечения.

*4 Документация по безопасности должна также содержать информацию относительно:*

*- выполнения самонаблюдения, способности системы сохранять работоспособность при дефектах и отказах;*

*- требований ранее разработанного программного обеспечения к его окружению (например, к техническому обеспечению или другим компонентам программного обеспечения);*

*- взаимодействия интерфейсов ранее разработанного программного обеспечения с техническим обеспечением, количеством информации, необходимым для полного определения безопасности функциональной работы системы.*

*5 Документация по безопасности операционной системы ранее разработанного комплекса оборудования должна обеспечить поступление информации, позволяющей производить правильное прогнозирование относительно ключевых моментов безопасности существенных элементов работы системы, например, максимального времени реакции соответствующих приложений или максимального использования ими ресурсов.*

*Такая информация может быть представлена в виде данных, формул и/или моделей, позволяющих вычислить время реакции соответствующих приложений и использование ими ресурсов для наихудшего случая. Если программное обеспечение предлагает слишком широкий диапазон функций, интерфейсов и возможностей для конфигурации, может быть затруднено обеспечение необходимой уверенности в правильности информации без знания функционирования программного обеспечения.*

6.2.1.3 Свойства

1 Эксплуатационные документы должны быть четкими, не допускающими двоякой интерпретации.

### 6.2.2 Свидетельство корректности

6.2.2.1 Общие требования

1 Должна быть подтверждена корректность ранее разработанного программного обеспечения по отношению к его документации по безопасности.

Обычно подтверждение является качественным, общепризнанных методов количественной оценки не существует. *Подходы, которые могут использоваться для проверки корректности ранее разработанного программного обеспечения, представлены на рисунке 5, если:*

*- ранее разработанное программное обеспечение было создано в соответствии с требованиями 6.1, 6.2.1, 6.4, 6.5, 6.6, 6.7 и 6.10, то обоснование правильности не требуется;*

*- соответствие этим требованиям не может быть должным образом подтверждено, то требования 6.2.2.2, 6.2.2.3 и 6.2.2.4 обеспечиваются дополнительными средствами, которые могут быть использованы для завершения обоснования.*

*2 При использовании дополнительных средств для обоснования корректности следует определить и обосновать критерии приемки на ранних стадиях жизненного цикла безопасности программного обеспечения. Эти критерии следует обосновать с учетом тех требований настоящего стандарта, соответствие которым не было должным образом установлено.*

Доверие к ранее разработанному программному обеспечению достигается легче, если оно может использоваться только ограниченным числом способов и/или если проект СКУ и его программного обеспечения предусматривает четкий набор условий их использования.

6.2.2.2 Дополнительные тестирования

*1 Тестирования ранее разработанного программного обеспечения, выполненные при создании СКУ, должны быть оформлены документально. Эти тестирования должны подтвердить, что в условиях применения в СКУ ранее разработанное программное обеспечение и его работа соответствуют документации по безопасности.*

*Условия использования могут касаться таких аспектов, как конфигурация ранее разработанного программного обеспечения (особенно установка параметров и данных конфигурации), использование функций и интерфейсов, аппаратных средств, процессора и загрузки по требованию.*

*2 Правила, используемые при разработке дополнительных тестирований, должны быть оформлены документально и обоснованы.*

*3 В документации по дополнительным тестированиям должны быть сделаны следующие записи:*

*- использованная версия и (если возможно) конфигурация ранее разработанного программного обеспечения;*

*- описание проведенных тестирований и (если возможно) использованное техническое обеспечение с тем, чтобы была возможность повторить эти тестирования в идентичных условиях;*

*- принятые гипотезы и доказательство их достоверности;*

*- полученные результаты и доказательство их правильности;*

*- заключения и согласованные решения.*

6.2.2.3 Эксплуатационный опыт

*Эксплуатационный опыт может использоваться как дополнительное доказательство корректности ранее разработанного программного обеспечения, при следующих условиях:*

*1 Принимаемый во внимание эксплуатационный опыт должен соответствовать точно идентифицированным версиям ранее разработанного программного обеспечения и, если это программное обеспечение привязано к определенному оборудованию, то этот опыт должен быть накоплен на этом оборудовании.*

*2 Если весь эксплуатационный опыт или его часть соответствует другим версиям ранее разработанного программного обеспечения и/или оборудования, различия с версиями, которые используются в СКУ, должны быть оценены и уместность данного эксплуатационного опыта должна быть обоснована.*

*3 Должно быть представлено документально оформленное обоснование того, что принимаемый во внимание эксплуатационный опыт соответствует условиям использования СКУ или еще более жестким условиям.*

*4 Объем учтенного эксплуатационного опыта должен быть оформлен документально.*

*5 Методы, использованные для накопления принимаемого во внимание эксплуатационного опыта, должны быть оформлены документально. В частности, должен быть оформлен документально тот факт, что все отказы, вызванные ранее разработанным программным обеспечением в течение учтенного эксплуатационного опыта, были правильно обнаружены и зафиксированы.*

*6 Должно быть показано, что эти отказы были правильно проанализированы и что соответствующие ошибки программного обеспечения исправлены.*

*Эксплуатационный опыт в системах более низкого класса безопасности или в системах, не классифицированных по безопасности, может быть учтен при условии соответствия требованиям данного перечисления.*

6.2.2.4 Сертификация

*Ранее разработанное программное обеспечение, уже действующее в системах, важных для безопасности (хотя не обязательно в СКУ атомных электростанций), возможно, было аттестовано на соответствие некоторым стандартам по безопасности. Свидетельство, обеспеченное такой сертификацией, может быть принято во внимание при следующих условиях:*

*1 Четкая идентификация ранее разработанного сертифицированного программного обеспечения должна быть оформлена документально. Если оно было аттестовано как часть большего изделия (например, как часть оборудования или комплекса оборудования), четкая идентификация этого изделия должна также быть оформлена документально.*

*2 Должны быть оценены факторы, на основе которых осуществляется сертификация, в том числе:*

- *условия сертификации (например, условия использования и допущения);*
- *методы и инструменты, используемые для сертификации;*
- *полученные результаты (например, свойства и/или аттестованные измерения).*

*3 Должна быть обоснована значимость этих условий и результатов для корректности и СКУ.*

*4 Должна быть обоснована эффективность методов и инструментов, используемых для сертификации.*

*5 Должен быть определен сертификационный орган, и он должен быть компетентным в отношении аттестуемых свойств и/или измерений.*

*6 Версия ранее разработанного сертифицированного программного обеспечения должна быть такой же, что используется в СКУ.*

6.2.2.5 Модификация

*Если проведена хорошо определенная и ограниченная модификация ранее разработанного программного обеспечения, для которого уже существует обоснование корректности либо необходимо устранить дефекты, то для обновления или завершения обоснования могут использоваться следующие требования в качестве замены требований 6.2.2.1—6.2.2.4. Изменение в конфигурации ранее разработанного программного обеспечения не является модификацией при условии, что его новая конфигурация остается в пределах, охватываемых обоснованием корректности.*

*1 Модификация ранее разработанного программного обеспечения должна быть оформлена документально. Документация должна содержать:*

- *четкую идентификацию измененного программного обеспечения;*
- *обстоятельства модификации, если программное обеспечение является частью большего продукта (например, оборудования или комплекса оборудования);*
- *цели, спецификацию и ограничения модификации;*
- *изменения, произведенные в документации по безопасности.*

*Следует также установить изменения, внесенные в проект ранее разработанного программного обеспечения.*

*Контекст модификации может, например, отражать:*

- *четкую идентификацию измененного большего продукта;*
- *цели, спецификацию и ограничения модификации изделия;*
- *модификации, которые должны быть внесены в остальную часть продукта или которые могут иметь воздействие на ранее разработанное программное обеспечение;*
- *верификацию и валидацию, выполняемые на уровне продукта.*

*2 Документально оформленное свидетельство (например, основанное на визуальном анализе, анализах и/или тестированиях с применением инструментальных программ), касающееся измененного программного обеспечения, а возможно, и большего продукта, должно подтверждать, что:*

- *цели модификации достигнуты;*
- *дефекты не были внесены;*
- *модифицированное программное обеспечение соответствует его обновленной документации по безопасности.*

*3 Достаточность этого свидетельства должна быть обоснована, по возможности, с учетом сделанных модификаций и условий использования в СКУ.*

### 6.2.3 Функциональная пригодность

*Цель настоящего пункта состоит в обеспечении соответствия ранее разработанного программного обеспечения потребностям СКУ и того, что оно не является слишком сложным для этих потребностей.*

*1 В случае, если используется документация по безопасности ранее разработанного программного обеспечения, она должна быть сопоставлена со спецификацией системы и проектом системы. Несоответствия должны быть устранены.*

*2 Следует идентифицировать те функции ранее разработанного программного обеспечения, которые не требуются в спецификации требований к системе. Необходимо представить обоснование безопасности.*

**6.2.4 Выбор и использование специализированных устройств с встроенным программным обеспечением**

Устройства типа «черный ящик» с встроенным программным обеспечением могут использоваться в СКУ при следующих условиях:

1 Программное обеспечение устройства должно быть интегрировано так, чтобы оно не могло быть изменено пользователем и не могло использоваться отдельно от остальной части устройства.

2 Число функций устройства, его потенциал для конфигурации и объем его интерфейсов и взаимодействий с остальной частью СКУ должны быть ограничены так, чтобы тестирования могли охватить все функции.

3 Должно быть представлено свидетельство того, что данное устройство соответствует требованиям 6.2.1, 6.2.2 и 6.2.3.

4 Должно быть представлено свидетельство того, что конфигурация и использование устройства в СКУ соответствуют требованиям 6.4, 6.5.1 и 6.5.2.

5 Проект СКУ и/или ее программного обеспечения должен обеспечивать использование устройства в четко определенных условиях.

**6.3 Спецификация требований к программному обеспечению**

Настоящий подраздел завершает и уточняет требования подпункта 6.1.2.3 МЭК 61513.

**6.3.1 Цели**

1 Требования к программному обеспечению СКУ должны быть определены и оформлены документально.

В настоящем стандарте соответствующий документ или пакет документов назван «Спецификация требований к программному обеспечению». Его цель состоит в определении задач программного обеспечения без определения путей их решения. Однако ограничивающие факторы проекта и разработки, вероятно, придется определить, если это потребуется при рассмотрении проекта СКУ или архитектуры СКУ.

*2 В спецификации требований к программному обеспечению следует избегать излишней сложности проекта программного обеспечения и обеспечивать стабильные условия использования программного обеспечения.*

3 Спецификация требований к программному обеспечению должна быть такой, чтобы:

- она способствовала уверенности в корректности проекта СКУ;
- было показано соответствие СКУ требованиям МЭК 61513.

Требования МЭК 61513, относящиеся к спецификации требований к программному обеспечению, содержатся главным образом в 6.1.1.2, 6.1.1.3, 6.1.1.4, 6.1.2.2, 6.1.2.4 и 6.1.3.

4 Спецификация требований к программному обеспечению должна быть основой проекта программного обеспечения, валидации программного обеспечения и возможных модификаций программного обеспечения.

**6.3.2 Исходная информация**

*1 Исходная информация для спецификации требований к программному обеспечению должна включать в себя спецификацию системы и документацию проекта системы.*

*Исходная информация может также включать в себя другие документы, например, такие как специфичные ограничения проекта и/или правила и стандарты, применимые к исходной информации.*

*2 Структура спецификации требований к программному обеспечению должна способствовать проведению сертификации обеспечивать соответствие и полноту по отношению к исходным документам.*

*Спецификация требований к программному обеспечению может ссылаться на части исходных документов во избежание ненужных дублирований и минимизирования риска несогласованности. Она может также ссылаться на другие ранее существовавшие документы, такие, например, как документация по ранее разработанному программному обеспечению.*

*3 Ссылки на другие документы, имеющиеся в спецификации требований к программному обеспечению, должны быть четкими и однозначными.*

*4 В спецификации требований к программному обеспечению следует избегать ненужных дополнений к ее исходным данным.*

*В принципе, желательно, чтобы программное обеспечение не имело большего числа возможностей, чем требуется, чтобы свести к минимуму ее сложность. Тем не менее поскольку существующая в промышленности практика основана на использовании ранее разработанных компонентов, может быть обосновано также включение невостребованных возможностей.*

### 6.3.3 Содержание

1 Спецификация требований к программному обеспечению должна определять:

- прикладные функции, которые должны быть обеспечены программным обеспечением;
- различные режимы работы программного обеспечения и соответствующие условия переходов;
- интерфейсы и взаимодействие программного обеспечения с его окружением (например, с операторами, остальной частью системы СКУ, другими системами и оборудованием, с которыми оно взаимодействует или разделяет ресурсы), включая ролевые имена, типы, форматы, диапазоны и ограничения по входам и выходам;
- параметры программного обеспечения, которые, при необходимости, изменяются операторами во время эксплуатации, их ролевые имена, типы, форматы, диапазоны и ограничения и проверки, осуществляемые программным обеспечением в случае изменений этих параметров;
- требуемые рабочие характеристики (при необходимости);
- указание на то, чего программное обеспечение не должно делать или чего должно избегать (при необходимости);
- при необходимости требования или допущения, устанавливаемые программным обеспечением к его окружению.

2 В спецификации требований к программному обеспечению следует также устанавливать параметры (например, загрузка по запросу), подверженные воздействию окружения программного обеспечения, особенно для наихудших условий.

Требования к функциям, интерфейсам и характеристикам могут зависеть от режима работы, значений параметров, данных конфигурации и условий, создаваемых для программного обеспечения.

3 В спецификации требований к программному обеспечению следует определять режимы его работы при обнаружении ошибок или отказов. Если для СКУ требуются периодические тестирования, то спецификация требований к программному обеспечению должна также определить требования к режиму выполнения таких тестирований.

4 Спецификация требований к программному обеспечению должна содержать требования к качеству программного обеспечения и указывать на необходимые ограничения проекта программного обеспечения и его работы с целью обеспечения его корректности и работоспособности.

Например, в спецификацию требований к программному обеспечению могут входить ограничения:

- направленные на обеспечение уверенности в правильности программного обеспечения и проекта системы (например, пределы, устанавливаемые при использовании динамически размещенных ресурсов, таких как память, вычислительная мощность, полоса пропускания канала связи, ресурсы операционной системы);
- направленные на увеличение устойчивости СКУ к дефектам, способности программного обеспечения обнаруживать ошибки и отказы и оповещать о них, работать в указанных режимах и восстанавливаться после отказов;
- направленные на обеспечение уверенности в том, что ошибки операторов и отказы других систем или оборудования, с которыми программное обеспечение взаимодействует или использует общие ресурсы, не будут приводить к недопустимым результатам.

5 Спецификация требований к программному обеспечению должна определить задачи программного обеспечения по своевременному информированию операторов об ошибках или отказах, касающихся функций СКУ, определенных в качестве важных для безопасности. Информация, предоставляемая операторам, должна позволить им предпринимать любое необходимое действие.

6 Спецификация требований к программному обеспечению должна определять функции и требования, связанные с категорией безопасности С.

### 6.3.4 Свойства

*1 Языки, правила и стандарты, используемые для разработки спецификации требований к программному обеспечению, должны способствовать ее ясности и четкости, и их следует выбирать с учетом использования этих объектов в соответствии с исходной информацией, а также при проектировании и реализации нового программного обеспечения.*

*Так как конкретный формат спецификации не всегда обеспечивает ясное, четкое и проверяемое выражение всего того, что требует определения, в одной и той же спецификации требований к программному обеспечению могут быть использованы различные и дополняющие друг друга форматы; например, для определения прикладных функций могут использоваться форматы, отличные от используемых для других функций.*

*2 Требования спецификации должны быть выражены таким способом, чтобы их соблюдение могло быть объективно оценено.*

### 6.4 Проект программного обеспечения

#### 6.4.1 Цели

1 Проект программного обеспечения должен быть оформлен документально. В документации следует приводить обзор организации и функционирования программного обеспечения.

В настоящем стандарте соответствующий документ или пакет документов назван «Спецификация проекта программного обеспечения». Если используется ранее разработанное программное обеспечение, то в спецификации проекта программного обеспечения может быть дана ссылка на соответствующий эксплуатационный документ.

*2 Спецификация проекта программного обеспечения должна способствовать достижению уверенности в качестве проекта программного обеспечения и его корректности относительно спецификации требований к программному обеспечению.*

3 Спецификация проекта программного обеспечения должна обеспечить свидетельство о том, что положения спецификации требований к программному обеспечению, важному для безопасности, приняты во внимание и будут соблюдаться при всех указанных в ней условиях.

*4 В спецификации проекта программного обеспечения следует в документальной форме представлять меры, гарантирующие раннее обнаружение любой ошибки или отказ программного обеспечения и ее (его) нераспространение за установленные пределы. В спецификации проекта программного обеспечения следует также документально представлять действия, предпринимаемые при обнаружении ошибки или отказа.*

5 Спецификация проекта программного обеспечения должна обеспечивать (в случае необходимости) устранение неблагоприятных побочных эффектов, связанных с ошибками и отказами программного обеспечения до его возвращения к нормальному режиму работы.

*6 Если это не приводит к* чрезмерной сложности, *проект программного обеспечения СКУ должен облегчать:*

- *анализ и тестирование программного обеспечения и его компонентов;*
- *локализацию дефектов;*
- *идентификацию результатов модификации.*

7 Спецификация проекта программного обеспечения должна служить основой для реализации и интеграции программного обеспечения, а также для возможных его модификаций.

#### 6.4.2 Исходные данные

1 Исходные данные для процесса проектирования программного обеспечения должны включать в себя Спецификацию требований к программному обеспечению и эксплуатационные документы ранее разработанного программного обеспечения.

Исходные данные могут также включить в себя другие документы, такие, например, как специфичные проектные ограничения и/или применимые правила и стандарты.

#### 6.4.3 Содержание

1 Спецификация проекта программного обеспечения должна включать в себя спецификацию по:

- полной организации программного обеспечения;
- полному функционированию программного обеспечения при условиях и режимах работы в соответствии со спецификацией требований к программному обеспечению.

2 Полная организация программного обеспечения должна обеспечить информацию:

- о четкой идентификации и конфигурации ранее разработанного программного обеспечения;
- о распределении ресурсов, компонентов и задач программного обеспечения по подсистемам;
- о распределении (под)функций и характеристик программного обеспечения по определенным для него задачам;
- об основных внутренних взаимосвязях, в частности, взаимосвязях между задачами программного обеспечения.

Полное функционирование должно обеспечивать информацию:

- о взаимодействиях, протоколах связи и информационных потоках;
- об установлении последовательностей и временных ограничений;
- об использовании ресурсов;
- о синхронизации, особенно при использовании разделенных ресурсов.

*3 В спецификации проекта программного обеспечения должен быть документально отражен порядок соблюдения важных для безопасности требований к программному обеспечению при всех заданных условиях. При использовании ранее разработанного программного обеспечения подтверждение важных для безопасности свойств программного обеспечения должно основываться на про-*

*гнозной информации, содержащейся в соответствующей документации по безопасности (см. требование 5, подпункт 6.2.1.2).*

*4 В спецификации проекта программного обеспечения и в документации по проектированию системы должны быть установлены и обоснованы меры, предпринимаемые для уменьшения влияния известных и ожидаемых режимов отказов любого ранее разработанного программного обеспечения или устройства со встроенным программным обеспечением, для которых использованы дополнительные меры по подтверждению корректности (см. 6.2.2.1).*

*5 В спецификации проекта программного обеспечения должны быть установлены правила реализации программного обеспечения. В частности, должны быть установлены правила конфигурирования и использования ранее разработанного программного обеспечения с тем, чтобы обеспечить использование этого программного обеспечения контролируемым образом, согласующимся с соответствующей документацией по безопасности.*

*6 Спецификация проекта программного обеспечения должна включать в себя детальный проект любого нового программного обеспечения, реализованного на универсальном языке. В спецификации компонента такого программного обеспечения следует указывать:*

*- функции, обеспечиваемые компонентами с их взаимосвязями, ролями, типами, форматами, диапазонами и ограничениями для входов и выходов сигналов об аномальном состоянии, данных конфигурации;*

*- рабочие характеристики, например, время отклика, точность (если это необходимо);*

*- требования компонентов к их окружению, например, потребность в динамически распределяемой памяти, ресурсам операционной системы и т.п. (если это необходимо);*

*- любая другая информация, о которой пользователи компонента должны быть осведомлены;*

*- любые важные ограничения по реализации.*

#### 6.4.4 Свойства

*1 В спецификации проекта программного обеспечения должен быть четко и ясно представлен проект программного обеспечения. Формат и синтаксис, используемые в спецификации проекта, должны способствовать этой четкости и ясности.*

*Основной подход может базироваться на принципе «сверху — вниз», но некоторые документы могут также содержать информацию, которая обращает внимание на особо важные аспекты (например, устойчивость к отказам), учитываемые во всем программном обеспечении или СКУ.*

### 6.5 Реализация нового программного обеспечения

#### 6.5.1 Общие требования

Требования настоящего пункта применимы ко всему новому программному обеспечению, т.е. к конфигурации ранее разработанного программного обеспечения и программам, написанным на проблемно-ориентированных или универсальных языках.

1 Использование ранее разработанного программного обеспечения должно быть верифицировано на соответствие документации по безопасности и ограничениям, установленным спецификацией проекта программного обеспечения.

2 Процедуры, используемые для трансляции новых программ в рабочую программу, должны быть оформлены документально и верифицированы.

*3 Обновление рабочей программы после внесения изменений в программы должно быть выполнено, по возможности, автоматизированными средствами.*

#### 6.5.2 Конфигурация программного обеспечения и устройств, содержащих программное обеспечение

Требование настоящего пункта является специфичным для конфигурации настраиваемого программного обеспечения. Такое программное обеспечение может быть ранее разработанным или новым. Данное требование также применимо к конфигурации настраиваемых устройств со встроенным программным обеспечением. Тем не менее, если данные конфигурации представляют собой выполняемую программным обеспечением или системой обработку (т.е. когда они фактически являются программой), то применяют требования 6.5.3.

1 Конфигурация настраиваемого программного обеспечения и устройств с встроенным программным обеспечением должна быть оформлена документально.

#### 6.5.3 Реализация с помощью проблемно-ориентированных языков

Требования настоящего пункта являются специфичными для программ, написанных на проблемно-ориентированных языках. В общем случае для отражения всей или части спецификации требований к программному обеспечению или спецификации проекта программного обеспечения могут использо-

ваться проблемно-ориентированные форматы (такие как логические диаграммы или блок-схемы функций). Затем необходимы только ограниченный детальный проект и некоторые усилия по реализации для преобразования спецификации в программы, которые могут быть автоматически транслированы в рабочую программу.

1 Части спецификации требований к программному обеспечению и/или спецификации проекта программного обеспечения, которые используются для получения рабочей программы с помощью средств автоматизации, должны рассматриваться как программы, написанные на проблемно-ориентированных языках.

*2 Программы, написанные на проблемно-ориентированных языках, должны быть верифицированы на предмет функциональной корректности и согласованности. Верификация должна подтверждать, что:*

*- все особенности проекта полностью поняты (т.е. не будет неожиданного режима работы при всех указанных условиях);*

*- указанный режим работы согласуется с целями, установленными исходными данными к спецификации требований к программному обеспечению.*

*Для улучшения понимания спецификаций и верификации их функциональной корректности и согласованности могут быть применены анимация, тестирования, проверка, сквозной контроль, формальный анализ и доказательство.*

*3 Для тестирования этих программ должны быть определены и документально оформлены критерии достаточности, по возможности, с учетом результатов других средств верификации. Если эти критерии не выполняются, то должно быть приведено обоснование.*

*Такие критерии могут основываться на функциональных и/или структурных показателях.*

*4 Написанные на проблемно-ориентированных языках программы должны соответствовать документально оформленным правилам, разработанным для улучшения их ясности, модифицируемости и проверяемости. Несоответствия должны быть обоснованы.*

*Набор правил может быть специфичным для языка или пакета программ. Невысокая сложность, ясность и стандартные расположение и представление, модульность, наличие необходимых комментариев, отсутствие небезопасных особенностей языка и его инструментов — вот примеры свойств, которые в общем случае облегчают понимание, проверку, тестирование и последующую модификацию программ.*

### 6.5.4 Реализация с помощью универсальных языков

Требование настоящего пункта является специфичным для программ, написанных на универсальных языках.

*1 Документально оформленная верификация должна обеспечивать обоснование того, что программы, написанные на универсальных языках, соответствуют их спецификации, как это определено спецификацией проекта программного обеспечения.*

*Верификация может состоять из комбинации визуального анализа, анализа с помощью инструментальных программ и/или тестирований.*

*Проверки программы, сквозной контроль, контрольные таблицы и другие подобные методы часто являются мощными методами визуального анализа, которыми можно пользоваться для выявления дефектов программного обеспечения.*

*Анализы с помощью инструментальных программ могут проводиться для определения программ, которые с наибольшей вероятностью могут содержать дефекты, и/или для формального доказательства того, что программа имеет (или не имеет) заданные свойства. Например, анализы могут обеспечить уверенность в том, что при данных условиях (например, при условии нахождения исходных данных в пределах заданных диапазонов) программы или определенные части программы не содержат дефектов определенного вида (например, неинициализированных переменных, арифметического переполнения или исчезновения значащих разрядов).*

*Тестирования могут быть выполнены на основном компьютере или с помощью средств поддержки программных разработок.*

*2 В документации по верификации должны фиксироваться:*

- *идентичность и версия проверяемых программ;*
- *информация, необходимая для повторения верификации в аналогичных условиях;*
- *принятые гипотезы и обоснование их справедливости;*
- *полученные результаты и обоснование их правильности;*
- *выводы и согласованные решения;*
- *обоснование соответствия критериям достаточности.*

*3 Должны быть установлены и документально оформлены критерии приемки при тестированиях этих программ, по возможности, с учетом результатов других средств верификации. Несоответствие этим критериям должно быть обосновано.*

*Критерии приемки могут базироваться на функциональных и/или структурных показателях.*

4 Программы, написанные на универсальных языках, должны соответствовать оформленным документально правилам, обеспечивающим четкость, возможность модификации и тестируемость. *Эти правила следует выражать так, чтобы их можно было верифицировать, и они должны быть нацелены на раннее обнаружение и ограничение ошибок в программном обеспечении.*

*Набор правил может быть специфичным для языка или пакета программ. Невысокая сложность, структурное программирование, модульность, инкапсуляция, скрытая информация (для того, чтобы пользователи программного продукта имели дело лишь с предоставляемым им сервисом, но не с внутренней работой продукта), наличие соответствующих комментариев, отсутствие опасных особенностей языка и его инструментов являются примерами свойств, которые могут облегчить понимание, верификацию, тестирование и последующую модификацию программного обеспечения.*

*5 Если могут использоваться инструменты статического анализа сложности кода, то правила должны устанавливать допустимые метрические пределы.*

*6 Программы, написанные на универсальных языках, должны быть верифицированы на соответствие применяемым правилам и стандартам. Несоответствия должны быть обоснованы и должны быть предприняты, документально оформлены и, при необходимости, обоснованы соответствующие контрмеры.*

*Контрмеры в случае несоответствия могут представлять собой, например, более полную верификацию.*

### 6.6 Программные аспекты интеграции системы

Интеграция программного обеспечения рассматривается как часть интеграции системы. Настоящий подраздел дополняет пункты 6.1.4 и 6.2.3 МЭК 61513, устанавливая дополнительные требования, специфичные или особенно важные для программного обеспечения.

1 Интеграция программного обеспечения и/или его анализ должны показать, что интегрированные система и программное обеспечение:

- соответствуют тем положениям проекта, которые обеспечивают выполнение указаний спецификации требований к программному обеспечению, определенных в качестве важных для безопасности;
- соответствуют ограничениям, изложенным в спецификации требований к программному обеспечению, касающимся корректности и работоспособности.

*2 Если валидационные тестирования программного обеспечения полагаются недостаточными для проверки программы, то необходимая уверенность в правильности работы должна быть обеспечена дополнительными интеграционными тестированиями программного обеспечения либо другими средствами.*

3 Интеграция программного обеспечения должна быть выполнена согласно положениям плана интеграции системы или плана интеграции программного обеспечения.

4 Должны быть составлены отчеты по результатам выполнения плана, используемого для интеграции программного обеспечения, например, результатам тестирований. В случае необходимости проведения модификации программного обеспечения или системы должна быть возможность повторения всех или части интеграционных тестирований для оценки степени возможных изменений в работе программы.

### 6.7 Программные аспекты валидации системы

Цель валидации программного обеспечения состоит в обеспечении соответствия интегрированного программного обеспечения функциональной спецификации, а также спецификациям характеристик и интерфейса, определяемым требованиями к СКУ. Таким образом, валидация программного обеспечения рассматривается как часть валидации системы. Настоящий подраздел дополняет пункты 6.1.5 и 6.2.4 МЭК 61513, выдвигая следующие дополнительные требования, специфичные или особенно важные для программного обеспечения:

*1 Валидация программного обеспечения должна показать, что интегрированное в завершенную систему программное обеспечение соответствует каждому из положений спецификации требований к программному обеспечению, касающихся функциональности, характеристик и интерфейса, и по своему назначению способствует соответствию спецификации требований к системе. При этом должно быть включено обоснование того, что:*

*- установленные функции программного обеспечения правильно выполняются, если их аргументы и исходные данные находятся в диапазонах, указанных в спецификации требований к программному обеспечению, а условия применения соответствуют указанным в этой спецификации;*

*- функции системы, реализация которых осуществляется с помощью программного обеспечения, выполняются правильно в условиях применения, определенных в спецификации требований к системе;*

*- программное обеспечение обеспечивает защиту от ошибок операторов и отказов других систем и оборудования в соответствии со спецификацией требований к программному обеспечению;*

*- программное обеспечение вносит свой вклад в обеспечение необходимой защиты системы от ошибок операторов и отказов других систем и оборудования в соответствии со спецификацией требований к системе;*

*- технические данные станции, используемые СКУ или интегрированные в нее, являются правильными; в частности, валидация программного обеспечения должна показать, что эти данные правильно описывают системы и оборудование станции, с которыми программное обеспечение взаимодействует или разделяет ресурсы и правильно обращается к этим системам и оборудованию.*

Примечание — Если представлено соответствующее обоснование, то для некоторых аспектов тестирований при валидации допускается использовать техническое обеспечение на платформе, идентичной штатной.

2 Валидация программного обеспечения должна быть выполнена в соответствии с положениями плана валидации системы. В противном случае валидация должна быть выполнена в соответствии с положениями плана валидации программного обеспечения.

*3 План валидации программного обеспечения должен устанавливать необходимые действия по валидации, а также показать, что все указания спецификации требований к программному обеспечению, касающиеся функциональности, характеристик и интерфейса, правильно учтены при выполнении этих действий. В плане также должны быть установлены основные этапы валидации программного обеспечения (например, этап до размещения на штатное место, за которым следует этап функционирования на штатном месте) и соответствующие средства, методы и инструменты, которые при этом должны быть использованы.*

4 Должны быть составлены отчеты по результатам выполнения плана, используемого при валидации программного обеспечения. В случае необходимости проведения модификаций программного обеспечения или системы должна быть возможность повторения всех или части валидационных тестирований для оценки степени возможных изменений в работе. *Результаты валидации программного обеспечения должны быть понятны лицам, компетентным в данной области, но не участвовавшим непосредственно в процессе валидации.*

5 В отчетах должны быть документально оформлены конфигурация программного обеспечения и конфигурация окружающей среды при проведении валидации, например, техническое обеспечение, инструментальные программы (если использовались).

6 Группа, которая составляет план валидации программного обеспечения, должна включать в себя, по крайней мере, одного человека, не участвовавшего в разработке проекта и его реализации.

### 6.8 Инсталляция программного обеспечения на штатном месте

Пункт 6.1.6 МЭК 61513 содержит требования к установке СКУ на штатном месте. В настоящем подразделе приводятся дополнительные требования, специфичные или особо важные для программного обеспечения.

1 Процедура инсталляции программного обеспечения на штатном месте должна быть оформлена документально. Процедура должна обеспечивать гарантированную инсталляцию правильной и полной версии программного обеспечения.

2 В процедуру инсталляции программного обеспечения на штатном месте должны быть включены и детально описаны проверки и тестирования на штатном месте, которые должны выполняться до начала полномасштабной эксплуатации СКУ. В частности, должно быть верифицировано выполнение условий, необходимых для правильной работы программного обеспечения.

Например, эти условия могут касаться средств технического обеспечения, на которых установлено программное обеспечение, или других систем, с которыми программное обеспечение взаимодействует или разделяет ресурсы.

### 6.9 Протоколы отклонений от нормы

1 Если обнаружена неожиданная, очевидно неправильная, необъяснимая или ненормальная работа программы после ее принятия в эксплуатацию, то должен быть составлен протокол отклонения от нормы работы программы.

2 В протоколе отклонения от нормы работы программы следует указать подробности работы программы, конфигурацию программного обеспечения и технического обеспечения и управляющих действий во время анормальной работы. В протоколе также следует указать его обозначение, составителя, а также место и время составления.

После краткого обзора протокола может быть добавлено его обозначение с тем, чтобы подчеркнуть действительность протокола.

3 Протоколы отклонения от нормы подвергают анализу. Возникшие проблемы оформляют документально, отслеживают и решают.

### 6.10 Модификация программного обеспечения

Решение о проведении модификаций программного обеспечения зависит от его влияния на СКУ, поэтому такие решения подчинены требованиям 6.1.7 и 6.3.6 МЭК 61513. В настоящем подразделе представлены следующие дополнительные требования, специфичные или особенно важные для программного обеспечения:

1 Модификации программного обеспечения должны разрабатываться в соответствии с требованиями 6.1—6.5. Они должны инсталлироваться в штатную аппаратуру в соответствии с требованиями 6.8.

2 Интеграцию и валидацию модификаций программного обеспечения следует проводить в соответствии с 6.6 и 6.7. Если объем модификации не требует соблюдения требований этих двух подразделов в полном объеме, интеграция модифицированного программного обеспечения должна проводиться согласно регрессивному плану интеграции программного обеспечения, а валидация — регрессивному плану валидации программного обеспечения. Полнота и подробность этих планов должны быть обоснованы с учетом объема каждой из модификаций, осуществленных в спецификации требований к программному обеспечению и спецификации проекта программного обеспечения. Должны быть составлены отчеты по результатам реализации этих планов.

*3 Если используется регрессивный подход, то регрессивный план интеграции программного обеспечения и регрессивный план валидации программного обеспечения должны обеспечить достаточную уверенность в том, что модифицированное программное обеспечение полностью соответствует новой спецификации требований к программному обеспечению, а также, что:*

*- цели модификации достигнуты;*

*- ошибки не внесены;*

*- модифицированное и/или вводимое ранее разработанное программное обеспечение работает в соответствии с документацией по безопасности и так, как того требует измененная Спецификация проекта программного обеспечения;*

*- другие модифицированные и/или новые компоненты программного обеспечения соответствуют их спецификации.*

4 Модификации программного обеспечения должны быть оформлены документально в полном объеме. В частности, все связанные с программным обеспечением документы, на которые повлияли модификации, должны быть обновлены.

*5 В документации по модификации программного обеспечения следует указывать:*

*- цели модификации программного обеспечения, включая любые цели на уровне системы;*

*- любые изменения, произведенные в его спецификации;*

*- любые ограничения, которые должны соблюдаться при выполнении модификации;*

*- компоненты программного обеспечения, которые затрагиваются или вновь создаются в процессе модификации;*

*- идентификацию версий этих компонентов до и после модификации;*

*- ссылки на модифицированный проект и/или документы по реализации*

*В контексте модификации на уровне системы могут, например, быть указаны:*

*- идентификация версии системы (или оборудования) до и после модификации;*

*- цели, спецификация и ограничения, связанные с модификацией системы;*

*- модификации в остальной части СКУ и/или других систем, взаимодействующих с программным обеспечением, которые необходимо осуществить или которые могут повлиять на программное обеспечение;*

*- влияние модификации на СКУ и/или другие системы, взаимодействующие с программным обеспечением, на работу системы и данные, существующие на штатном месте;*

*- ограничения, применяемые к инсталляции модификации на штатном месте.*

*6 Степень подробности документации по модификации программного обеспечения должна быть такой, чтобы:*

*- она обеспечивала уверенность в правильности измененного программного обеспечения и СКУ;*

*- могло быть подтверждено соответствие СКУ соответствующим требованиям МЭК 61513.*

*Соответствующие требования МЭК 61513 находятся главным образом в 6.1.1.2, 6.1.1.3, 6.1.1.4, 6.1.2.2, 6.1.2.4 и 6.1.3.*

7 Должна быть проведена оценка влияния модификации программного обеспечения на остальные части СКУ и другие системы, с которыми программное обеспечение взаимодействует или разделяет ресурсы. Все необходимые меры должны быть предприняты для обеспечения правильной работы СКУ.

8 Должны быть оценены влияния на программное обеспечение модификаций в остальной части СКУ или других системах, с которыми программное обеспечение взаимодействует или делит ресурсы.

Должны быть предприняты все необходимые меры для обеспечения правильной работы СКУ.

**Приложение ДА**
**(справочное)**

## Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации

Таблица ДА.1

| Обозначение ссылочного международного стандарта | Степень соответствия | Обозначение и наименование соответствующего национального стандарта |
|---|---|---|
| МЭК 61226 | IDT | ГОСТ Р МЭК 61226—2011 «Атомные станции. Системы контроля и управления, важные для безопасности. Классификация функций контроля и управления» |
| МЭК 61513 | IDT | ГОСТ Р МЭК 61513—2011 «Атомные станции. Системы контроля и управления, важные для безопасности. Общие требования к системам» |
| Примечание — В настоящей таблице использовано следующее условное обозначения степени соответствия стандартов:<br>IDT — идентичные стандарты. | | |

**Библиография**


| | |
|---|---|
| [1] IEC 60880 | Software for computers in the safety systems of nuclear power stations |
| [2] IEC 60880-2:2000 | Software for computers important to safety for nuclear power plant — Part 2: Software aspects of defense against common cause failures, use of software tools and of pre-developed software |
| [3] IEC 61508-3 | Functional safety of electrical/electronical/programmable electronic safety-related systems — Part 3: Software requirements |
| [4] IEC 61508-4 | Functional safety of electrical/electronical/programmable electronic safety-related systems — Part 4: Definitions and abbreviations |
| [5] IEC 61511-1 | Functional safety: safety-instrumented systems for the process industry sector — Part1: Framework, definitions, system, and software requirements |
| [6] ISO/IEC 12207 | Information Technology — Software life cycle processes |
| [7] ISO 9001 | Quality management systems |
| ISO 9000-3 | Quality management and quality assurance standards — Part 3: Guidelines for the application of ISO 9001:1994 to the development, supply, installation and maintenance of computer software |
| [8] IAEA Safety Standards Series, N NS-R-1 | Safety of Nuclear Power Plants: Design Requirements |
| [9] IAEA Safety Standards Series, N NS-G-1.1 | Software for Computer-Based Systems Important to Safety in Nuclear Power Plants: Safety Guide |
| [10] IAEA Safety Standards Series, N NS-G-1.3 | Software for Instrumentation and Control Systems Important to Safety in Nuclear Power Plants: Safety Guide |
| [11] IAEA Safety Series, N 50-/SG-Q | Quality Assurance for Safety in Nuclear Power Plants and Other Nuclear Installations — Code and Safety Guides Q1-Q14 |

УДК 004.384:004.4.4: 621.311.25: 621.039: 006.354 ОКС 27.120.20

Ключевые слова: атомная электростанция; архитектура контроля и управления; системы контроля и управления, важные для безопасности; жизненный цикл безопасности; функции контроля и управления; функции контроля и управления категории В; функции контроля и управления категории С; разработка программного обеспечения; инсталляция; функциональная валидация

Редактор *В.Н. Копысов*
Технический редактор *В.Н. Прусакова*
Корректор *Е.Д. Дульнева*
Компьютерная верстка *А.Н. Золотаревой*

Сдано в набор 24.10.2011. Подписано в печать 11.11.2011. Формат 60 × 84 1/8. Гарнитура Ариал.
Усл. печ. л. 4,65. Уч.-изд. л. 4,10. Тираж 84 экз. Зак. 1085.

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

gost.e-ecolog.ru