МЕЖГОСУДАРСТВЕННЫЙ СОВЕТ ПО СТАНДАРТИЗАЦИИ, МЕТРОЛОГИИ И СЕРТИФИКАЦИИ
(МГС)
INTERSTATE COUNCIL FOR STANDARDIZATION, METROLOGY AND CERTIFICATION
(ISC)

МЕЖГОСУДАРСТВЕННЫЙ
СТАНДАРТ

**ГОСТ
IEC 62304—
2022**

ИЗДЕЛИЯ МЕДИЦИНСКИЕ. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Процессы жизненного цикла

(IEC 62304:2006 + Amd.1:2015, Medical device software — Software life cycle processes, IDT)

Издание официальное

**Москва
Российский институт стандартизации
2022**

Предисловие

Цели, основные принципы и общие правила проведения работ по межгосударственной стандартизации установлены ГОСТ 1.0 «Межгосударственная система стандартизации. Основные положения» и ГОСТ 1.2 «Межгосударственная система стандартизации. Стандарты межгосударственные, правила и рекомендации по межгосударственной стандартизации. Правила разработки, принятия, обновления и отмены»

Сведения о стандарте

1 ПОДГОТОВЛЕН рабочей группой, состоящей из представителей Общества с ограниченной ответственностью «МЕДИТЕСТ» (ООО «МЕДИТЕСТ», г. Москва), Общества с ограниченной ответственностью «Аурига» (ООО «Аурига», г. Москва) и Общества с ограниченной ответственностью «Компания «ЭЛТА» (ООО «Компания «ЭЛТА», г. Москва) на основе собственного перевода на русский язык англоязычной версии стандарта, указанного в пункте 5

2 ВНЕСЕН Федеральным агентством по техническому регулированию и метрологии

3 ПРИНЯТ Межгосударственным советом по стандартизации, метрологии и сертификации (протокол от 30 сентября 2022 г. № 154-П)

За принятие проголосовали:

Краткое наименование страны по МК (ИСО 3166) 004—97	Код страны по МК (ИСО 3166) 004—97	Сокращенное наименование национального органа по стандартизации
Беларусь	BY	Госстандарт Республики Беларусь
Киргизия	KG	Кыргызстандарт
Россия	RU	Росстандарт
Узбекистан	UZ	Узстандарт

4 Приказом Федерального агентства по техническому регулированию и метрологии от 26 октября 2022 г. № 1196-ст межгосударственный стандарт ГОСТ IEC 62304—2022 введен в действие в качестве национального стандарта Российской Федерации с 1 сентября 2023 г.

5 Настоящий стандарт идентичен международному стандарту IEC 62304:2006 «Программное обеспечение медицинского изделия. Процессы жизненного цикла программного обеспечения» («Medical device software — Software life cycle processes», IDT), включая изменение Amd.1:2015. Изменение Amd.1:2015 выделено двойной вертикальной линией, расположенной на полях напротив соответствующего текста.

Наименование настоящего стандарта изменено относительно наименования указанного международного стандарта для приведения в соответствие с ГОСТ 1.5 (подраздел 3.6).

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

6 Настоящий стандарт подготовлен на основе применения ГОСТ Р МЭК 62304—2013*

7 ВВЕДЕН ВПЕРВЫЕ

* Приказом Федерального агентства по техническому регулированию и метрологии от 26 октября 2022 г. № 1196-ст ГОСТ Р МЭК 62304—2013 отменен с 1 сентября 2023 г.

Информация о введении в действие (прекращении действия) настоящего стандарта и изменений к нему на территории указанных выше государств публикуется в указателях национальных стандартов, издаваемых в этих государствах, а также в сети Интернет на сайтах соответствующих национальных органов по стандартизации.

В случае пересмотра, изменения или отмены настоящего стандарта соответствующая информация будет опубликована на официальном интернет-сайте Межгосударственного совета по стандартизации, метрологии и сертификации в каталоге «Межгосударственные стандарты»

Содержание

* Символ «звездочка» (*), используемый как первый знак в наименовании или начале параграфа, указывает, что в приложении В имеется соответствующее руководство.

Введение

Программное обеспечение часто является неотъемлемой частью технологии МЕДИЦИНСКИХ ИЗДЕЛИЙ. Создание БЕЗОПАСНОГО и результативного МЕДИЦИНСКОГО ИЗДЕЛИЯ, содержащего программное обеспечение, требует знаний о его предназначении, а также доказательств того, что программное обеспечение надежно функционирует, не создавая недопустимых РИСКОВ.

Настоящий стандарт определяет основу ПРОЦЕССОВ жизненного цикла совместно с ДЕЯТЕЛЬНОСТЬЮ и ЗАДАЧАМИ, необходимыми для проектирования и технической поддержки (обслуживания) безопасного ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКИХ ИЗДЕЛИЙ. Настоящий стандарт определяет требования для каждого ПРОЦЕССА жизненного цикла. Каждый ПРОЦЕСС жизненного цикла состоит из совокупности видов ДЕЯТЕЛЬНОСТИ, причем большинство видов ДЕЯТЕЛЬНОСТИ, в свою очередь, состоят из набора ЗАДАЧ.

В качестве основной концепции полагается, что ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МЕДИЦИНСКИХ ИЗДЕЛИЙ проектируется и обслуживается с использованием систем менеджмента качества (см. 4.1) и систем МЕНЕДЖМЕНТА РИСКА (см. 4.2). ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА уже достаточно хорошо описан в международном стандарте ISO 14971:2007. Поэтому настоящий стандарт использует ссылки на этот стандарт. Некоторые незначительные дополнительные требования к МЕНЕДЖМЕНТУ РИСКА необходимы для программного обеспечения, особенно в области определения вклада факторов программного обеспечения, связанных с ОПАСНОСТЯМИ. Эти требования установлены в разделе 7 как ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА программного обеспечения.

Является ли программное обеспечение фактором, способствующим ОПАСНОЙ СИТУАЦИИ, определяется во время ДЕЯТЕЛЬНОСТИ по идентификации ОПАСНОСТИ в ПРОЦЕССЕ МЕНЕДЖМЕНТА РИСКА. ОПАСНЫЕ СИТУАЦИИ, которые могут быть косвенно вызваны программным обеспечением (например, предоставляя вводящую в заблуждение информацию, которая может вызвать неверную реакцию администрирования), рассматриваются, когда определяется, является ли программное обеспечение способствующим фактором. Решение подвергнуть программное обеспечение УПРАВЛЕНИЮ РИСКОМ принимается в течение ДЕЯТЕЛЬНОСТИ по УПРАВЛЕНИЮ РИСКОМ в ПРОЦЕССЕ МЕНЕДЖМЕНТА РИСКА. ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА программного обеспечения, требуемый настоящим стандартом, должен быть включен в ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА изделия согласно ISO 14971:2007.

ПРОЦЕСС разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ состоит из множества ДЕЙСТВИЙ. Эти ДЕЙСТВИЯ показаны на рисунке 1 и описаны в разделе 5. Поскольку множество инцидентов в этой области связано с обслуживанием или технической поддержкой СИСТЕМ МЕДИЦИНСКИХ ИЗДЕЛИЙ, включая неподходящие обновления программного обеспечения и его модернизации, ПРОЦЕСС технической поддержки (обслуживания) программного обеспечения считается столь же важным, как и ПРОЦЕСС разработки программного обеспечения. ПРОЦЕСС технической поддержки программного обеспечения очень похож на ПРОЦЕСС разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ. Это показано на рисунке 2 и описано в разделе 6.

Примечание — Национальным техническим комитетам следует обратить внимание на тот факт, что ИЗГОТОВИТЕЛЯМ оборудования и испытательным организациям может потребоваться переходный период после опубликования нового, измененного или пересмотренного документа IEC или ISO, в течение которого они могут производить продукцию в соответствии с новыми требованиями и оснащаться оборудованием для проведения новых или пересмотренных испытаний. Комитет рекомендует, чтобы содержание этой публикации было принято для обязательного применения на национальном уровне не ранее чем через три года с даты публикации.

Рисунок 1 — Краткий обзор ПРОЦЕССОВ и ДЕЯТЕЛЬНОСТИ по разработке программного обеспечения

Рисунок 2 — Краткий обзор ПРОЦЕССОВ и ДЕЯТЕЛЬНОСТИ по технической поддержке программного обеспечения

Настоящий стандарт идентифицирует два дополнительных ПРОЦЕССА, которые считаются важными для разработки безопасного ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКОГО ИЗДЕЛИЯ. Это ПРОЦЕСС менеджмента конфигурации программного обеспечения (раздел 8) и ПРОЦЕСС разрешения проблем программного обеспечения (раздел 9).

Изменение 1 добавляет в настоящий стандарт требования к УСТАРЕВШЕМУ/НАСЛЕДУЕМОМУ ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ, если разработка программного обеспечения была проведена до появления текущей версии, с целью оказания помощи изготовителям, которые должны продемонстрировать соответствие настоящему стандарту для соответствия Европейским директивам. Изменения в классификации безопасности программного обеспечения включают уточнение требований и обновление классификации безопасности программного обеспечения, а также подход, основанный на оценке рисков.

Настоящий стандарт не устанавливает организационную структуру ИЗГОТОВИТЕЛЯ или то, какое структурное подразделение организации должно осуществлять выполнение ПРОЦЕССА, ДЕЯТЕЛЬНОСТИ или ЗАДАЧИ. Требование состоит в том, что в целях соответствия настоящему стандарту ПРОЦЕСС, ДЕЯТЕЛЬНОСТЬ или ЗАДАЧА должны быть завершены.

Настоящий стандарт не устанавливает наименование, формат или точное содержание документации, которая будет создана. Требование состоит в том, чтобы ЗАДАЧИ документировались, а решение, как оформлять эту документацию, остается за пользователем этого стандарта.

Настоящий стандарт не предписывает конкретную модель жизненного цикла. Пользователи ответственны за выбор модели жизненного цикла для проекта программного обеспечения и за отображение ПРОЦЕССОВ, ДЕЯТЕЛЬНОСТИ и ЗАДАЧ настоящего стандарта применительно к этой модели.

Приложение А содержит разъяснения пунктов настоящего стандарта. Приложение В содержит рекомендации по положениям стандарта.

Для целей стандарта:

- «должен» означает, что соответствие требованиям или испытаниям обязательно для соответствия настоящему стандарту;
- «следует» означает, что соответствие требованиям или испытаниям настоящего стандарта рекомендовано, но не обязательно для соответствия требованиям настоящего стандарта;
- «может» используется для описания допустимого способа достижения соответствия требованию;
- «установить» означает определять, документировать и осуществлять выполнение;
- там, где в настоящем стандарте используется термин «если применимо» в сочетании с требуемым ПРОЦЕССОМ, ДЕЯТЕЛЬНОСТЬЮ, ЗАДАЧЕЙ или продукцией, ИЗГОТОВИТЕЛЬ должен использовать ПРОЦЕСС, ДЕЯТЕЛЬНОСТЬ, ЗАДАЧУ или продукцию, если не может документированно опровергнуть необходимость применения.

Введение к поправке 1

Первое издание стандарта IEC 62304 было опубликовано в 2006 году. Настоящая поправка предназначена для добавления требований к УСТАРЕВШЕМУ/НАСЛЕДУЕМОМУ ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ, где разработка программного обеспечения является предыдущей к существующей текущей версии, с целью оказания помощи изготовителям продемонстрировать соответствие настоящему стандарту для соответствия Европейским директивам. Изменения в классификации безопасности программного обеспечения, внесенные настоящей поправкой, включают уточнение требований и обновление классификации безопасности программного обеспечения с целью применения риск-ориентированного подхода. Параллельно продолжается работа по разработке второго издания IEC 62304.

МЕЖГОСУДАРСТВЕННЫЙ СТАНДАРТ

ИЗДЕЛИЯ МЕДИЦИНСКИЕ. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Процессы жизненного цикла

Medical devices. Software. Life cycle processes

Дата введения — 2023—09—01

1 Область применения

1.1 *Цель

Настоящий стандарт устанавливает требования к жизненному циклу ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКИХ ИЗДЕЛИЙ. Совокупность ПРОЦЕССОВ, ДЕЯТЕЛЬНОСТИ и ЗАДАЧ, описанных в настоящем стандарте, устанавливает общую основу для ПРОЦЕССОВ жизненного цикла ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКИХ ИЗДЕЛИЙ.

1.2 *Применимость

Настоящий стандарт применяется при разработке и технической поддержке ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКИХ ИЗДЕЛИЙ, когда программное обеспечение само по себе является МЕДИЦИНСКИМ ИЗДЕЛИЕМ или когда программное обеспечение является встроенной или неотъемлемой частью готового МЕДИЦИНСКОГО ИЗДЕЛИЯ.

Примечание 1 — Настоящий стандарт может применяться при разработке и обслуживании программного обеспечения, которое само по себе является медицинским изделием. Однако, прежде чем этот тип программного обеспечения может быть введен в эксплуатацию, необходимо осуществить дополнительную деятельность по разработке на системном уровне. Эта системная деятельность не входит в область применения настоящего стандарта, но ее описание приведено в IEC 82304-1 [22].

Настоящий стандарт описывает ПРОЦЕССЫ, предназначенные для применения к программному обеспечению, которое выполняется на процессоре или другим программным обеспечением (например, интерпретатором), выполняемым на процессоре.

Настоящий стандарт применяется независимо от устройства (устройств) постоянного хранения, используемого(ых) для хранения программного обеспечения (например: жесткий диск, оптический диск, постоянная или флеш-память).

Настоящий стандарт применяется независимо от способа доставки программного обеспечения (например: передача по сети или электронной почте, оптический диск, флеш-память или EEPROM). Сам способ доставки программного обеспечения не считается ПРОГРАММНЫМ ОБЕСПЕЧЕНИЕМ МЕДИЦИНСКИХ ИЗДЕЛИЙ.

Настоящий стандарт не затрагивает вопросы валидации и окончательного утверждения МЕДИЦИНСКОГО ИЗДЕЛИЯ, даже когда МЕДИЦИНСКОЕ ИЗДЕЛИЕ полностью состоит из программного обеспечения.

Примечание 2 — Если медицинское изделие включает встроенное программное обеспечение, предназначенное для работы на процессоре, то к программному обеспечению применяются требования настоящего стандарта, включая требования, касающиеся программного обеспечения неизвестного происхождения (см. 8.1.2).

Издание официальное

Примечание 3 — Валидацию и другую деятельность по разработке необходимо проводить на системном уровне, прежде чем программное обеспечение и медицинское изделие могут быть введены в эксплуатацию. Эта системная деятельность не входит в область применения настоящего стандарта, но ее описание приведено в соответствующих стандартах на продукцию (например, IEC 60601-1, IEC 82304-1 и т. д.).

1.3 Взаимосвязь с другими стандартами

При разработке МЕДИЦИНСКИХ ИЗДЕЛИЙ, в отношении жизненного цикла ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКИХ ИЗДЕЛИЙ, настоящий стандарт обычно используется совместно с другими применимыми стандартами. Приложение С показывает связь между настоящим стандартом и другими уместными стандартами.

1.4 Соответствие

Соответствие настоящему стандарту определяется как выполнение всех установленных в нем ПРОЦЕССОВ, ДЕЯТЕЛЬНОСТИ и ЗАДАЧ, в соответствии с классом безопасности программного обеспечения.

Примечание — Классы безопасности ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, назначенные каждому требованию, указываются в нормативном тексте, следующем за требованиями.

Соответствие устанавливается посредством проверки всей документации, требуемой настоящим стандартом, включая ФАЙЛ МЕНЕДЖМЕНТА РИСКА, и оценки ПРОЦЕССОВ, ДЕЯТЕЛЬНОСТИ и ЗАДАЧ, требуемых согласно классу безопасности программного обеспечения.

Примечание 1 — Данные оценки могут быть сделаны путем внешнего или внутреннего аудита.

Примечание 2 — Несмотря на то что должны быть выполнены указанные ПРОЦЕССЫ, ДЕЯТЕЛЬНОСТЬ и ЗАДАЧИ, существует определенная гибкость в методах осуществления этих ПРОЦЕССОВ и выполнения ДЕЯТЕЛЬНОСТИ и ЗАДАЧ.

Примечание 3 — Если какие-либо требования, содержащие словосочетание «соответствующим образом», не были выполнены, то для проведения оценки необходимо предоставить документированные обоснования.

Примечание 4 — Термин «соответствие», используемый в стандарте ИСО/МЭК 12207, применяется в настоящем стандарте таким же образом.

Примечание 5 — Соответствие УСТАРЕВШЕГО/УНАСЛЕДОВАННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ см. в подразделе 4.4 настоящего стандарта.

2 Нормативные ссылки

В настоящем стандарте использована нормативная ссылка на следующий стандарт [для датированной ссылки применяют только указанное издание ссылочного стандарта, для недатированной — последнее издание (включая все изменения)]:

ISO 14971, Medical devices — Application of risk management to medical devices (Изделия медицинские. Применение менеджмента риска к медицинским изделиям)

3* Термины и определения

В настоящем стандарте применены следующие термины с соответствующими определениями.

3.1 **ДЕЯТЕЛЬНОСТЬ** (ACTIVITY): Совокупность из одной или более взаимосвязанных или взаимодействующих ЗАДАЧ.

3.2 **АНОМАЛИЯ** (ANOMALY): Любое условие или состояние, которое отклоняется от ожиданий, основанных на требованиях спецификаций, проектно-конструкторских документов, стандартов и т. д., либо от чьего-то восприятия или опыта. АНОМАЛИИ могут быть обнаружены во время проверки, тестов, анализа, компиляции, использования ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКОГО ИЗДЕЛИЯ или прилагаемой документации либо в других случаях.

Примечание — Основано на IEEE 1044:1993, определение 3.1.

3.3 **АРХИТЕКТУРА** (ARCHITECTURE): Организационная структура СИСТЕМЫ или компонента.
[IEEE 610.12:1990]

3.4 **ЗАПРОС НА ИЗМЕНЕНИЕ** (CHANGE REQUEST): Документированная спецификация изменения, которое будет сделано в ПРОГРАММНОМ ОБЕСПЕЧЕНИИ МЕДИЦИНСКОГО ИЗДЕЛИЯ.

3.5 **СОСТАВНАЯ ЧАСТЬ КОНФИГУРАЦИИ** (CONFIGURATION ITEM): Объект, который может быть однозначно определен в данной конкретной точке.

Примечание — Основано на ISO/IEC 12207:2008, 4.7.

3.6 **ПОСТАВЛЯЕМЫЙ РЕЗУЛЬТАТ** (DELIVERABLE): Требуемый итог или выход (включая документацию) ДЕЯТЕЛЬНОСТИ или ЗАДАЧИ.

3.7 **ОЦЕНИВАНИЕ** (EVALUATION): Систематическое определение степени соответствия объекта установленным критериям.
[ISO/IEC 12207:2008, 4.12]

3.8 **ВРЕД** (HARM): Нанесение физического повреждения или другого вреда здоровью людей либо вреда имуществу или окружающей среде.
[ISO 14971:2007 2.2]

3.9 **ОПАСНОСТЬ (**HAZARD): Потенциальный источник ВРЕДА
[ISO 14971:2007 2.3]

3.10 **ИЗГОТОВИТЕЛЬ** (MANUFACTURER): Физическое или юридическое лицо, ответственное за проектирование, изготовление, упаковывание и/или маркировку МЕДИЦИНСКИХ ИЗДЕЛИЙ; установку, сборку или монтаж СИСТЕМЫ; или адаптацию МЕДИЦИНСКОГО ИЗДЕЛИЯ перед выпуском его в обращение и/или вводом в эксплуатацию независимо от того, выполняет ли эти операции вышеупомянутое лицо или третья сторона от его имени.

Примечание 1 — К определению изготовителя могут применяться положения национального или регионального регулирования.

Примечание 2 — Определение маркировки см. в ISO 13485:2003, определение 3.6.

[ISO 14971:2007, 2.8]

3.11 **МЕДИЦИНСКОЕ ИЗДЕЛИЕ** (MEDICAL DEVICE): Любой инструмент, аппарат, прибор, устройство, оборудование, имплантат, *in vitro* реагент или калибратор, программное обеспечение, материал либо иные подобные или связанные с ними изделия, предназначенные ИЗГОТОВИТЕЛЕМ для применения к человеку по отдельности или в сочетании друг с другом в целях:

- диагностики, профилактики, мониторинга, лечения или облегчения заболеваний;
- диагностики, мониторинга, лечения, облегчения или компенсации последствий травмы;
- исследования, замещения или изменения анатомического строения или физиологических ПРОЦЕССОВ;
- поддержания или сохранения жизни;
- управления зачатием;
- дезинфекции МЕДИЦИНСКИХ ИЗДЕЛИЙ;
- получения информации для медицинских целей посредством исследования *in vitro* проб, взятых из тела человека, при условии, что их функциональное воздействие на человеческий организм не реализуется за счет фармакологических, иммунологических или метаболических средств, но может поддерживаться такими средствами.

Примечание 1 — Определение было разработано Целевой Группой Глобальной Гармонизации (GHTF). [ISO 13485:2003, определение 3.7].

Примечание 2 — Определения, используемые в регулировании разных стран, могут иметь некоторые различия.

Примечание 3 — В сочетании с IEC 60601-1:2005 и IEC 60601-1:2005/AMD1:2012 термин «медицинское изделие» имеет то же значение, что и МЕ ОБОРУДОВАНИЕ или МЕ СИСТЕМА (которые определены терминами IEC 60601-1).

3.12 **ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МЕДИЦИНСКОГО ИЗДЕЛИЯ** (MEDICAL DEVICE SOFTWARE): ПРОГРАММНАЯ СИСТЕМА, разработанная как составная часть разрабатываемого МЕДИЦИНСКОГО ИЗДЕЛИЯ или предназначенная для использования в качестве МЕДИЦИНСКОГО ИЗДЕЛИЯ.

Примечание — В это определение входит МЕДИЦИНСКОЕ ИЗДЕЛИЕ — программный продукт, который сам по себе является МЕДИЦИНСКИМ ИЗДЕЛИЕМ.

3.13 **ОТЧЕТ О ПРОБЛЕМАХ** (PROBLEM REPORT): Запись о фактическом или возможном поведении ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКОГО ИЗДЕЛИЯ, из которого пользователь или заинтересованное лицо могут узнать о том, что является опасным, не соответствующим предусмотренному назначению, или о том, что противоречит спецификации.

Примечание 1 — Настоящий стандарт не требует, чтобы каждый ОТЧЕТ О ПРОБЛЕМАХ приводил к изменениям в ПРОГРАММНОМ ОБЕСПЕЧЕНИИ МЕДИЦИНСКОГО ИЗДЕЛИЯ. ИЗГОТОВИТЕЛЬ может отклонить ОТЧЕТ О ПРОБЛЕМАХ для неверно понятого, ошибочного или несущественного события.

Примечание 2 — ОТЧЕТ О ПРОБЛЕМАХ может относиться к готовому ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ МЕДИЦИНСКОГО ИЗДЕЛИЯ или к ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ МЕДИЦИНСКОГО ИЗДЕЛИЯ, еще находящемуся в процессе разработки.

Примечание 3 — Настоящий стандарт требует от ИЗГОТОВИТЕЛЯ осуществлять некоторую дополнительную последовательность действий (см. раздел 6) по каждому ОТЧЕТУ О ПРОБЛЕМАХ, относящемуся к уже выпущенному продукту, с целью обеспечения идентификации и выполнения предписанных действий.

3.14 **ПРОЦЕСС** (PROCESS): Совокупность взаимосвязанных и взаимодействующих видов **ДЕЯТЕЛЬНОСТИ**, преобразующая входы в выходы.

[ISO 9000:2005, определение 3.4.1]

Примечание — Термин «ДЕЯТЕЛЬНОСТЬ» включает использование ресурсов.

3.15 **РЕГРЕССИОННОЕ ТЕСТИРОВАНИЕ** (REGRESSION TESTING): Испытание, которое необходимо для определения влияния изменений в компонентах СИСТЕМЫ на ее функциональность, надежность или эксплуатационные характеристики и на создание дополнительных дефектов.

[ISO/IEC 90003:2004, определение 3.11]

3.16 **РИСК** (RISK): Сочетание вероятности причинения ВРЕДА и тяжести этого ВРЕДА.

[ISO 14971:2007 2.16]

3.17 **АНАЛИЗ РИСКА** (RISK ANALYSIS): Систематическое использование доступной информации для идентификации ОПАСНОСТИ и определения РИСКА.

[ISO 14971:2007 2.17]

3.18 **УПРАВЛЕНИЕ РИСКОМ** (RISK CONTROL): ПРОЦЕСС принятия решений и выполнения мер по уменьшению РИСКОВ до установленных уровней или поддержанию их на установленных уровнях.

[ISO 14971:2007 2.19]

3.19 **МЕНЕДЖМЕНТ РИСКА** (RISK MANAGEMENT): Систематическое применение политики, процедур и практических методов менеджмента для решения ЗАДАЧ анализа, оценивания, управления и мониторинга РИСКА.

[ISO 14971:2007, 2.12, модифицировано — Фраза «и мониторинга» была удалена]

3.20 **ФАЙЛ МЕНЕДЖМЕНТА РИСКА** (RISK MANAGEMENT FILE): Совокупность записей и других документов, создаваемых в ПРОЦЕССЕ МЕНЕДЖМЕНТА РИСКА.

[ISO 14971:2007 2.23]

3.21 **БЕЗОПАСНОСТЬ** (SAFETY): Отсутствие недопустимого РИСКА.

[ISO 14971:2007 2.24]

3.22 **ЗАЩИЩЕННОСТЬ** (SECURITY): Защита информации и данных от чтения или изменения их посторонними лицами или системами таким образом, чтобы авторизованным лицам и системам доступ к ним не был запрещен.

Примечание — Основано на ISO/IEC 12207:2008, 4.39.

3.23 **СЕРЬЕЗНАЯ ТРАВМА** (SERIOUS INJURY): Повреждение или заболевание, которое:

- несет угрозу жизни;
- приводит к стойкому ухудшению функционирования организма или к постоянному ущербу (необратимому повреждению) структуры тела;
- требует медицинского или хирургического вмешательства с целью предотвращения стойкого ухудшения функционирования организма или постоянного ущерба (необратимого повреждения) структуры тела.

Примечание — Стойкое ухудшение означает необратимое ухудшение или утрату части структуры или функций организма, за исключением незначительного ухудшения или ущерба.

3.24 **МОДЕЛЬ ЖИЗНЕННОГО ЦИКЛА РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (**SOFTWARE DEVELOPMENT LIFE CYCLE MODEL): Концептуальная структура, охватывающая существование программного обеспечения от определения требований до выпуска программного обеспечения, которая:

- определяет ПРОЦЕССЫ, ДЕЯТЕЛЬНОСТЬ И ЗАДАЧИ, включенные в разработку ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКОГО ИЗДЕЛИЯ;
- описывает последовательность и взаимозависимость между ДЕЯТЕЛЬНОСТЬЮ и ЗАДАЧАМИ;
- идентифицирует этапы, на которых верифицируется полнота конкретных ПОСТАВЛЯЕМЫХ РЕЗУЛЬТАТОВ.

Примечание — Основано на ISO/IEC 12207:1995, определение 3.11.

3.25 **ПРОГРАММНАЯ СОСТАВНАЯ ЧАСТЬ** (SOFTWARE ITEM): Любая идентифицируемая (выделяемая) часть компьютерной программы, т. е. исходный код, объектный код, управляющий код, управляющие данные или набор этих элементов.

Примечание 1 — Разделение программы на составные части можно охарактеризовать тремя терминами. Верхний уровень — ПРОГРАММНАЯ СИСТЕМА. Самый нижний уровень, ниже которого подразделение на составные части не осуществляется, — ПРОГРАММНЫЙ БЛОК. Все уровни композиции, включая верхний и нижний уровни, можно назвать ПРОГРАММНЫМИ СОСТАВНЫМИ ЧАСТЯМИ. Тогда ПРОГРАММНАЯ СИСТЕМА состоит из одного или более ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ, и каждая ПРОГРАММНАЯ СОСТАВНАЯ ЧАСТЬ в свою очередь состоит из одного или более ПРОГРАММНЫХ БЛОКОВ или подразделенных ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ. Ответственность за обеспечение степени детализации ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ и ПРОГРАММНЫХ БЛОКОВ возлагается на ИЗГОТОВИТЕЛЯ.

Примечание 2 — Основано на ISO/IEC 90003:2004, 3.14 и ISO/IEC 12207:2008, 4.41.

3.26 е используется.

3.27 **ПРОГРАММНАЯ СИСТЕМА** (SOFTWARE SYSTEM): Совокупность ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ, предназначенных для выполнения конкретной функции или набора функций.

3.28 **ПРОГРАММНЫЙ БЛОК** (SOFTWARE UNIT): ПРОГРАММНАЯ СОСТАВНАЯ ЧАСТЬ, которая не может быть разделена на более мелкие части.

Примечание — Уровень детализации ПРОГРАММНЫХ БЛОКОВ определяется ИЗГОТОВИТЕЛЕМ (см. В.3).

3.29 **ПОНП Программное Обеспечение Неизвестного Происхождения (аббревиатура)** (SOUP software of unknown provenance (acronym)): ПРОГРАММНАЯ СОСТАВНАЯ ЧАСТЬ, которая уже разработана и общедоступна, но не была предназначена для включения в состав МЕДИЦИНСКОГО ИЗДЕЛИЯ (также известное как «готовое ПО») или ПРОГРАММНАЯ СОСТАВНАЯ ЧАСТЬ, разработанная ранее, для которой недоступны требуемые записи ПРОЦЕССОВ разработки.

Примечание — ПРОГРАММНАЯ СИСТЕМА МЕДИЦИНСКОГО ИЗДЕЛИЯ сама по себе не может считаться ПОНП.

3.30 **СИСТЕМА** (SYSTEM): Совокупная композиция, состоящая из одного или более ПРОЦЕССОВ, аппаратных средств, программного обеспечения, людей и средств, которая обеспечивает способность удовлетворить заявленную потребность или цель.

Примечание — Основано на ISO/IEC 12207:2008, 4.48.

3.31 **ЗАДАЧА** (TASK): Отдельная часть работы, которую необходимо выполнить.

3.32 **ПРОСЛЕЖИВАЕМОСТЬ** (TRACEABILITY): Степень, до которой может быть установлена взаимосвязь между двумя или более результатами (продуктами) ПРОЦЕССА разработки.

[IEEE 610.12:1990]

Примечание — Требования, архитектура, меры по управлению риском и т. д. являются примерами поставляемых результатов ПРОЦЕССА разработки.

3.33 **ВЕРИФИКАЦИЯ** (VERIFICATION): Подтверждение выполнения установленных требований на основе представления объективных свидетельств.

Примечание 1 — Термин «верифицирован» используется для обозначения соответствующего статуса.

[ISO 9000:2005, определение 3.8.4]

Примечание 2 — При проектировании и разработке ВЕРИФИКАЦИЯ относится к ПРОЦЕССУ проверки результатов конкретной ДЕЯТЕЛЬНОСТИ, чтобы определить соответствие требованиям, установленным к этой ДЕЯТЕЛЬНОСТИ.

3.34 **ВЕРСИЯ** (VERSION): Идентифицируемый отдельный вариант СОСТАВНОЙ ЧАСТИ КОНФИГУРАЦИИ.

Примечание 1 — Изменение ВЕРСИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКОГО ИЗДЕЛИЯ, приводящее к появлению новой ВЕРСИИ, требует действий по управлению конфигурацией программного обеспечения.

Примечание 2 — Основано на ISO/IEC 12207:2008, 4.56.

3.35 **ОПАСНАЯ СИТУАЦИЯ** (HAZARDOUS SITUATION): Обстоятельство, при котором люди, имущество или окружающая среда подвергаются воздействию одной или нескольких ОПАСНОСТЕЙ.

[ИСТОЧНИК: ISO 14971:2007, 2.4]

3.36 **УСТАРЕВШЕЕ [НАСЛЕДУЕМОЕ] ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ** (LEGACY SOFTWARE): ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МЕДИЦИНСКОГО ИЗДЕЛИЯ, которое было легально выпущено в обращение и по-прежнему доступно на рынке, но для которого недостаточно объективных свидетельств того, что оно было разработано в соответствии с текущей версией настоящего стандарта.

3.37 **ВЫПУСК** (RELEASE): Конкретная ВЕРСИЯ СОСТАВНОЙ ЧАСТИ КОНФИГУРАЦИИ, доступная для определенной цели.

Примечание — Основано на ISO/IEC 12207:2008, определение 4.35.

3.38 **ОСТАТОЧНЫЙ РИСК** (RESIDUAL RISK): РИСК, остающийся после выполнения мер ПО УПРАВЛЕНИЮ РИСКОМ.

Примечание 1 — Адаптировано из ISO/IEC Guide 51:1999, определение 3.9.

Примечание 2 — ISO/IEC Guide 51:1999, определение 3.9 использует термин «защитные меры», а не «меры по УПРАВЛЕНИЮ РИСКОМ». Однако в контексте настоящего стандарта «защитные меры» являются лишь одним из вариантов управления РИСКОМ, как описано в 6.2 [ISO 14971:2007].

[ISO 14971:2007, 2.15]

3.39 **ОПРЕДЕЛЕНИЕ РИСКА** (RISK ESTIMATION): ПРОЦЕСС, применяемый для присвоения значений вероятности возникновения ВРЕДА и тяжести этого ВРЕДА.

[ISO 14971:2007, 2.20]

3.40 **ОЦЕНИВАНИЕ РИСКА** (RISK EVALUATION): ПРОЦЕСС сравнения РИСКА, который уже определен, с установленными критериями РИСКА для установления допустимости РИСКА.

[ISO 14971:2007, 2.21]

4* Общие требования

4.1* Система менеджмента качества

ИЗГОТОВИТЕЛЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКОГО ИЗДЕЛИЯ должен быть способен продемонстрировать его соответствие требованиям потребителя и применимым регулирующим требованиям.

Примечание 1 — Демонстрация этой способности может быть осуществлена при помощи СИСТЕМЫ менеджмента качества, которая соответствует следующим требованиям:

- ISO 13485 [8] или
- национальному стандарту на систему менеджмента качества или
- системе менеджмента качества, требуемой национальным регулированием.

Примечание 2 — Руководство по применению требований системы менеджмента качества к ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ можно найти в ISO/IEC 90003 [15].

4.2 * МЕНЕДЖМЕНТ РИСКА

ИЗГОТОВИТЕЛЬ должен применять ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА в соответствии с ISO 14971.

4.3 *Классификация программного обеспечения в отношении безопасности

а) ИЗГОТОВИТЕЛЬ должен присвоить каждой ПРОГРАММНОЙ СИСТЕМЕ класс безопасности программного обеспечения (A, B или C) согласно РИСКУ причинения ВРЕДА пациенту, пользователю или иным лицам, исходя из ОПАСНОЙ СИТУАЦИИ, в которую ПРОГРАММНАЯ СИСТЕМА может внести свой вклад в наихудшем сценарии, как показано на рисунке 3.

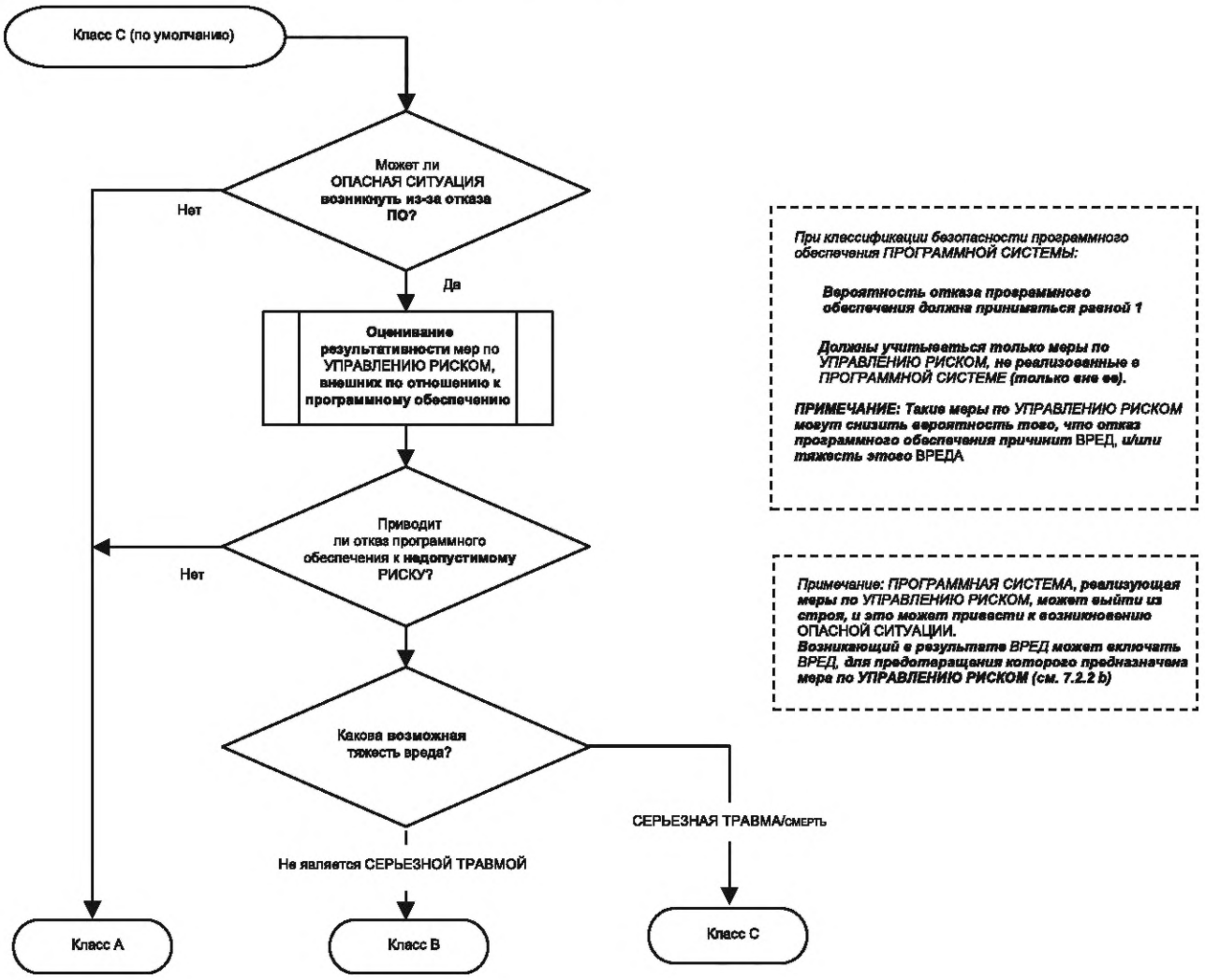


Рисунок 3 — Присвоение класса безопасности программного обеспечения

ПРОГРАММНАЯ СИСТЕМА относится к классу безопасности программного обеспечения A, если:

- ПРОГРАММНАЯ СИСТЕМА не может способствовать возникновению ОПАСНОЙ СИТУАЦИИ;
- ПРОГРАММНАЯ СИСТЕМА может способствовать возникновению ОПАСНОЙ СИТУАЦИИ, которая не приводит к недопустимому РИСКУ после рассмотрения мер по УПРАВЛЕНИЮ РИСКОМ, внешних по отношению к ПРОГРАММНОЙ СИСТЕМЕ.

- ПРОГРАММНАЯ СИСТЕМА относится к классу безопасности программного обеспечения B, если:
- ПРОГРАММНАЯ СИСТЕМА может способствовать возникновению ОПАСНОЙ СИТУАЦИИ, которая приводит к недопустимому РИСКУ после рассмотрения мер по УПРАВЛЕНИЮ РИСКОМ, внешних по отношению к ПРОГРАММНОЙ СИСТЕМЕ, и вытекающий из этого возможный ВРЕД не является СЕРЬЕЗНОЙ ТРАВМОЙ.

ПРОГРАММНАЯ СИСТЕМА относится к классу безопасности программного обеспечения С, если:

- ПРОГРАММНАЯ СИСТЕМА может способствовать возникновению ОПАСНОЙ СИТУАЦИИ, которая приводит к недопустимому РИСКУ, после рассмотрения мер по УПРАВЛЕНИЮ РИСКОМ, внешних по отношению к ПРОГРАММНОЙ СИСТЕМЕ, и в результате возможным ВРЕДОМ является смерть или СЕРЬЕЗНАЯ ТРАВМА.

Для ПРОГРАММНОЙ СИСТЕМЫ, первоначально классифицированной как класс безопасности программного обеспечения В или С, ИЗГОТОВИТЕЛЬ может реализовать дополнительные меры по УПРАВЛЕНИЮ РИСКОМ, внешние по отношению к ПРОГРАММНОЙ СИСТЕМЕ (включая пересмотр архитектуры системы, содержащей ПРОГРАММНУЮ СИСТЕМУ), и впоследствии присвоить ПРОГРАММНОЙ СИСТЕМЕ новую классификацию безопасности программного обеспечения.

Примечание 1 — Внешними мерами по УПРАВЛЕНИЮ РИСКОМ могут быть аппаратные средства, независимая ПРОГРАММНАЯ СИСТЕМА, медицинские процедуры или другие средства, позволяющие свести к минимуму способность программного обеспечения приводить к возникновению ОПАСНОЙ СИТУАЦИИ.

Примечание 2 — Определение допустимости риска см. в подразделе 3.2 «Ответственность руководства» ISO 14971:2007.

b) Не используется.

c) ИЗГОТОВИТЕЛЬ обязан документировать класс безопасности программного обеспечения, присвоенный каждой ПРОГРАММНОЙ СИСТЕМЕ, в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА.

d) Если ПРОГРАММНАЯ СИСТЕМА подразделяется на ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ и в дальнейшем ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ в свою очередь подразделяются на ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ, то такие ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ должны наследовать класс безопасности первоначальной ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ (или ПРОГРАММНОЙ СИСТЕМЫ), если только ИЗГОТОВИТЕЛЬ не обосновывает в документации присвоение другого класса безопасности программного обеспечения (классы безопасности программного обеспечения, присвоенные в соответствии с подразделом 4.3 a), заменяя «ПРОГРАММНУЮ СИСТЕМУ» на «ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ»). Это обоснование должно объяснять, почему ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ являются изолированными настолько, что могут классифицироваться отдельно.

e) ИЗГОТОВИТЕЛЬ должен документировать класс безопасности программного обеспечения каждой ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ, если этот класс отличается от класса ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ, из которой он был выделен при декомпозиции.

f) Для соответствия настоящему стандарту там, где настоящий стандарт применяется к группе ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ, ИЗГОТОВИТЕЛЬ должен использовать ПРОЦЕССЫ и ЗАДАЧИ, которые требуются для классификации ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ с наивысшей категорией из всей группы, если только ИЗГОТОВИТЕЛЬ в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА не приводит документированные обоснования для использования более низкой классификации.

g) Каждой ПРОГРАММНОЙ СИСТЕМЕ, если ей не присвоен класс безопасности программного обеспечения, по умолчанию должен присваиваться Класс С.

Примечание — В последующих разделах и подразделах классы безопасности программного обеспечения, к которым применяется конкретное требование, определяются в соответствии с требованием в форме [Класс...].

4.4* УСТАРЕВШЕЕ/НАСЛЕДУЕМОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

4.4.1 Общие сведения

В качестве альтернативы применению разделов 5—9 настоящего стандарта соответствие УСТАРЕВШЕГО/НАСЛЕДУЕМОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ может быть продемонстрировано, как указано в 4.4.2—4.4.5.

4.4.2 Деятельность по МЕНЕДЖМЕНТУ РИСКА

В соответствии с 4.2 настоящего стандарта ИЗГОТОВИТЕЛЬ должен:

a) оценить любую обратную связь, включая постпроизводственную информацию, об УСТАРЕВШЕМ ПРОГРАММНОМ ОБЕСПЕЧЕНИИ, касающуюся инцидентов и/или близких к ним ситуаций как внутри своей организации, так и/или от пользователей;

b) выполнить ДЕЯТЕЛЬНОСТЬ по УПРАВЛЕНИЮ РИСКОМ, связанную с продолжением использования УСТАРЕВШЕГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, с учетом следующих аспектов:

- интеграции УСТАРЕВШЕГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ в общую архитектуру МЕДИЦИНСКОГО ИЗДЕЛИЯ;

- постоянной пригодности мер по УПРАВЛЕНИЮ РИСКОМ, реализованных в рамках УСТАРЕВШЕГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ;

- идентификации ОПАСНЫХ СИТУАЦИЙ, связанных с продолжением использования УСТАРЕВШЕГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ;

- идентификации потенциальных причин, по которым УСТАРЕВШЕЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ может привести к возникновению ОПАСНОЙ СИТУАЦИИ;

- определения мер по УПРАВЛЕНИЮ РИСКОМ для каждой потенциальной причины, по которой УСТАРЕВШЕЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ способствует возникновению ОПАСНОЙ СИТУАЦИИ.

4.4.3 Анализ несоответствия ожидаемых и реализованных результатов

Основываясь на классе безопасности УСТАРЕВШЕГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (см. 4.3), ИЗГОТОВИТЕЛЬ должен провести анализ несоответствия ожидаемых и реализованных ПОСТАВЛЕННЫХ РЕЗУЛЬТАТОВ по сравнению с теми, которые требуются в соответствии с 5.2, 5.3, 5.7 и разделом 7.

a) ИЗГОТОВИТЕЛЬ должен оценить непрерывную пригодность полученных РЕЗУЛЬТАТОВ.

b) В случае идентификации несоответствий ожидаемых и реализованных результатов ИЗГОТОВИТЕЛЬ должен ОЦЕНИТЬ возможное снижение РИСКА в результате получения отсутствующих РЕЗУЛЬТАТОВ и связанной с ними ДЕЯТЕЛЬНОСТИ.

c) На основе этого оценивания ИЗГОТОВИТЕЛЬ должен определить РЕЗУЛЬТАТЫ, которые должны быть получены, а также связанную с ними ДЕЯТЕЛЬНОСТЬ, которая должна быть выполнена. Минимальным ПОСТАВЛЯЕМЫМ РЕЗУЛЬТАТОМ должны быть записи тестирования ПРОГРАММНЫХ СИСТЕМ (см. 5.7.5).

Примечание — Анализ несоответствий ожидаемых и реализованных результатов должен обеспечивать включение в требования к программному обеспечению мер по УПРАВЛЕНИЮ РИСКОМ, реализованных в УСТАРЕВШЕМ ПРОГРАММНОМ ОБЕСПЕЧЕНИИ.

4.4.4 Деятельность по устранению несоответствий ожидаемых и реализованных результатов

a) ИЗГОТОВИТЕЛЬ должен разработать и выполнить план по созданию идентифицированных ПОСТАВЛЯЕМЫХ РЕЗУЛЬТАТОВ. Там, где это возможно, объективные свидетельства могут быть использованы для формирования требуемых ПОСТАВЛЯЕМЫХ РЕЗУЛЬТАТОВ без выполнения ДЕЯТЕЛЬНОСТИ, установленной в 5.2, 5.3, 5.7 и разделе 7.

Примечание — План устранения идентифицированных несоответствий ожидаемых и реализованных результатов может быть включен в план технической поддержки программного обеспечения (см. 6.1).

b) План должен предусматривать использование ПРОЦЕССА решения проблем для обработки проблем, обнаруженных в УСТАРЕВШЕМ ПРОГРАММНОМ ОБЕСПЕЧЕНИИ и ПОСТАВЛЯЕМЫХ РЕЗУЛЬТАТАХ, в соответствии с Разделом 9.

c) Изменения в УСТАРЕВШЕМ ПРОГРАММНОМ ОБЕСПЕЧЕНИИ должны быть выполнены в соответствии с разделом 6.

4.4.5 Обоснование использования УСТАРЕВШЕГО/НАСЛЕДУЕМОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

ИЗГОТОВИТЕЛЬ должен задокументировать ВЕРСИЮ УСТАРЕВШЕГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ вместе с обоснованием дальнейшего использования УСТАРЕВШЕГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ на основе результатов 4.4.

Примечание — Выполнение 4.4 позволяет в дальнейшем использовать УСТАРЕВШЕЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ в соответствии с IEC 62304.

5 ПРОЦЕСС разработки программного обеспечения

5.1* Планирование разработки программного обеспечения

5.1.1 План разработки программного обеспечения

ИЗГОТОВИТЕЛЬ должен создать план (или планы) разработки программного обеспечения с целью провести всю необходимую ДЕЯТЕЛЬНОСТЬ в отношении ПРОЦЕССА разработки программно-

го обеспечения, соответствующий области, значимости и классу безопасности разрабатываемой ПРОГРАММНОЙ СИСТЕМЫ. МОДЕЛЬ ЖИЗНЕННОГО ЦИКЛА РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ должна быть либо полностью определена, либо указана в плане (или в планах). План должен содержать:

a) ПРОЦЕССЫ, которые будут использоваться при разработке ПРОГРАММНОЙ СИСТЕМЫ (см. примечание 4);

b) ПОСТАВЛЯЕМЫЕ РЕЗУЛЬТАТЫ (включая документацию) ДЕЯТЕЛЬНОСТИ и ЗАДАЧ;

c) ПРОСЛЕЖИВАЕМОСТЬ между требованиями СИСТЕМЫ, требованиями программного обеспечения, испытанием ПРОГРАММНОЙ СИСТЕМЫ и мерами по УПРАВЛЕНИЮ РИСКОМ, включенными в программное обеспечение;

d) конфигурацию программного обеспечения и менеджмент изменений, включая СОСТАВНЫЕ ЧАСТИ КОНФИГУРАЦИИ ПОНП (программное обеспечение неизвестного происхождения), и программного обеспечения, используемого для поддержки разработки;

e) решение проблем с программным обеспечением для обработки проблем, обнаруженных в ПРОГРАММНОМ ОБЕСПЕЧЕНИИ МЕДИЦИНСКОГО ИЗДЕЛИЯ, ПОСТАВЛЯЕМЫХ РЕЗУЛЬТАТАХ и ДЕЯТЕЛЬНОСТИ на каждой стадии жизненного цикла. [Классы A, B, C]

Примечание 1 — МОДЕЛЬ ЖИЗНЕННОГО ЦИКЛА РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ может определять различные элементы (ПРОЦЕССЫ, ДЕЯТЕЛЬНОСТЬ, ЗАДАЧИ и ПОСТАВЛЯЕМЫЕ РЕЗУЛЬТАТЫ) для различных ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ в соответствии с классами безопасности программного обеспечения для каждой ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ ПРОГРАММНОЙ СИСТЕМЫ.

Примечание 2 — ДЕЯТЕЛЬНОСТЬ и ЗАДАЧИ могут перекрываться или взаимодействовать и могут выполняться итеративно или рекурсивно. Это не подразумевает того, что должна использоваться определенная модель жизненного цикла.

Примечание 3 — Другие ПРОЦЕССЫ описываются в настоящем стандарте отдельно от ПРОЦЕССА разработки. Это не подразумевает того, что они должны быть реализованы в виде отдельной ДЕЯТЕЛЬНОСТИ и ЗАДАЧ. ДЕЯТЕЛЬНОСТЬ и ЗАДАЧИ других ПРОЦЕССОВ могут быть включены в ПРОЦЕСС разработки.

Примечание 4 — План разработки программного обеспечения может ссылаться на существующие ПРОЦЕССЫ или определять новые.

Примечание 5 — План разработки программного обеспечения может быть включен в план разработки общей СИСТЕМЫ.

5.1.2 Поддержание плана разработки программного обеспечения в актуальном состоянии

ИЗГОТОВИТЕЛЬ должен обновлять план по мере того, как осуществляется разработка. [Классы A, B, C]

5.1.3 План разработки программного обеспечения относительно проектирования и разработки СИСТЕМЫ

a) ИЗГОТОВИТЕЛЬ должен указать требования СИСТЕМЫ в плане разработки программного обеспечения в качестве входных данных.

b) В план разработки программного обеспечения ИЗГОТОВИТЕЛЬ должен включать или ссылаться на процедуры по координации разработки программного обеспечения с разработкой системы, необходимой для выполнения требований 4.1 (например, системная интеграция, верификация и валидация). [Классы A, B, C]

Примечание — Может не существовать различий между требованиями ПРОГРАММНОЙ СИСТЕМЫ и требованиями СИСТЕМЫ, если ПРОГРАММНАЯ СИСТЕМА является отдельной СИСТЕМОЙ (например, если программное обеспечение само по себе является изделием).

5.1.4 Стандарты, методы и инструменты планирования разработки программного обеспечения

В план разработки программного обеспечения ИЗГОТОВИТЕЛЬ должен включать или ссылаться:

a) на стандарты;

b) методы;

c) инструменты,

связанные с разработкой ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ класса C. [Класс C]

5.1.5 Программная интеграция и планирование тестирования интеграции

ИЗГОТОВИТЕЛЬ должен включить в план разработки программного обеспечения план интеграции ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ (включая ПОНП) или сослаться на него, а также выполнить тестирование во время интеграции. [Классы B, C]

Примечание 1 — Допускается объединение тестирования интеграции и тестирования ПРОГРАММНОЙ СИСТЕМЫ в единые план и совокупность ДЕЯТЕЛЬНОСТИ.

Примечание 2 — См. 5.6.

5.1.6 Планирование ВЕРИФИКАЦИИ программного обеспечения

В план разработки программного обеспечения ИЗГОТОВИТЕЛЬ должен включить или сослаться на следующую информацию по ВЕРИФИКАЦИИ:

a) ПОСТАВЛЯЕМЫЕ РЕЗУЛЬТАТЫ, требующие ВЕРИФИКАЦИИ;

b) требуемые ВЕРИФИКАЦИОННЫЕ ЗАДАЧИ для каждой ДЕЯТЕЛЬНОСТИ в жизненном цикле;

c) контрольные точки, на которых ВЕРИФИЦИРУЮТСЯ ПОСТАВЛЯЕМЫЕ РЕЗУЛЬТАТЫ;

d) критерии приемки для ВЕРИФИКАЦИИ ПОСТАВЛЯЕМЫХ РЕЗУЛЬТАТОВ. [Классы A, B, C]

5.1.7 Планирование МЕНЕДЖМЕНТА РИСКА программного обеспечения

В план разработки программного обеспечения ИЗГОТОВИТЕЛЬ должен включать или ссылаться на план осуществления ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА программного обеспечения в отношении ДЕЯТЕЛЬНОСТИ и ЗАДАЧ, включая МЕНЕДЖМЕНТ РИСКА, применяющийся к ПОНП. [Классы A, B, C]

5.1.8 Документация по планированию

В план разработки программного обеспечения ИЗГОТОВИТЕЛЬ должен включать, или ссылаться на информацию о документации, которая будет создана во время жизненного цикла разработки программного обеспечения. Каждому идентифицированному документу или типу документа должна быть присвоена (или содержаться непосредственно) следующая информация:

a) титульный лист, наименование или обозначение;

b) цель;

c) процедуры и ответственность за разработку, анализ, одобрение и модификацию. [Классы A, B, C]

Примечание — Информация для рассмотрения менеджмента конфигурации документации приведена в разделе 8.

5.1.9 Планирование менеджмента конфигурации программного обеспечения

В план разработки программного обеспечения ИЗГОТОВИТЕЛЬ должен включать или ссылаться на информацию о менеджменте конфигурации программного обеспечения. Эта информация должна содержать или ссылаться:

a) на классы, типы, категории или списки элементов, подлежащих управлению;

b) ДЕЯТЕЛЬНОСТЬ и ЗАДАЧИ по менеджменту конфигурации программного обеспечения;

c) структуру (структуры), отвечающую(ие) за ДЕЯТЕЛЬНОСТЬ по менеджменту конфигурации программного обеспечения;

d) их взаимосвязь с другими структурами, такими как разработка или техническая поддержка программного обеспечения;

e) случаи, когда элементы должны находиться под управлением конфигурации;

f) случаи, когда следует использовать ПРОЦЕСС решения проблем. [Классы A, B, C]

Примечание — См. раздел 8.

5.1.10 Поддержка элементов, подлежащих управлению

Элементы, подлежащие управлению, должны включать инструменты, элементы или настройки, использующиеся для разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКОГО ИЗДЕЛИЯ, которые могут воздействовать на ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МЕДИЦИНСКОГО ИЗДЕЛИЯ. [Классы B, C]

Примечание 1 — Примеры подобных элементов включают компиляторные/ассемблерные версии, созданные файлы, командные файлы и специфичные настройки окружения.

Примечание 2 — См. раздел 8.

5.1.11 Управление СОСТАВНЫМИ ЧАСТЯМИ КОНФИГУРАЦИИ программного обеспечения до ВЕРИФИКАЦИИ

ИЗГОТОВИТЕЛЬ должен запланировать размещение СОСТАВНОЙ ЧАСТИ КОНФИГУРАЦИИ под управление менеджмента конфигурации прежде, чем они будут ВЕРИФИЦИРОВАНЫ. [Классы B, C]

5.1.12 Идентификация и предотвращение распространенных дефектов программного обеспечения

ИЗГОТОВИТЕЛЬ должен включить или указать в плане разработки программного обеспечения процедуру:

a) для определения категорий дефектов, которые могут быть введены на основе выбранной технологии программирования и имеют отношение к их ПРОГРАММНОЙ СИСТЕМЕ;

б) документирования свидетельств, демонстрирующих неспособность этих дефектов приводить к недопустимому РИСКУ.

Примечание — Примеры категорий дефектов или причин, способствующих возникновению ОПАСНЫХ СИТУАЦИЙ, см. в приложении B из IEC/TR 80002-1:2009.

[Классы B, C]

5.2* Анализ требований к программному обеспечению

5.2.1 Отделение и документирование требований к программному обеспечению на основе требований СИСТЕМЫ

Для каждой ПРОГРАММНОЙ СИСТЕМЫ МЕДИЦИНСКОГО ИЗДЕЛИЯ ИЗГОТОВИТЕЛЬ должен определить и документировать требования к ПРОГРАММНОЙ СИСТЕМЕ исходя из требований уровня СИСТЕМЫ. [Классы A, B, C]

Примечание — Может не существовать различий между требованиями ПРОГРАММНОЙ СИСТЕМЫ и требованиями СИСТЕМЫ, если ПРОГРАММНАЯ СИСТЕМА является отдельной СИСТЕМОЙ (например, если программное обеспечение само по себе является изделием).

5.2.2 Содержание требований к программному обеспечению

Как применимые и подходящие в отношении ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКИХ ИЗДЕЛИЙ, ИЗГОТОВИТЕЛЬ должен включать в требования к программному обеспечению:

a) требования к потенциальным возможностям и функциональности.

Примечание 1 — Примеры включают:

- характеристики, связанные с выполнением (например, цели/назначение программного обеспечения, требования по синхронизации);
- физические характеристики (например, язык машинного кода, платформу, операционную СИСТЕМУ);
- компьютерное окружение (например, аппаратные средства, размер памяти, процессор, часовой пояс, инфраструктуру сети);
- необходимость совместимости с модернизациями или многими ПОНП или другими версиями изделий;

b) входные и выходные данные ПРОГРАММНОЙ СИСТЕМЫ.

Примечание 2 — Например:

- характеристики данных (например, цифровые, буквенно-цифровые, формат);
- диапазоны;
- пределы;
- значения по умолчанию;

c) интерфейсы между ПРОГРАММНОЙ СИСТЕМОЙ и другими СИСТЕМАМИ;

d) программные средства управления сигналами тревоги, предупреждениями и оповещением оператора;

e) требования к ЗАЩИЩЕННОСТИ.

Примечание 3 — Например:

- связанные с компромиссом относительно конфиденциальной информации;
- идентификация;
- авторизация;
- системный журнал;
- непрерывность связи;
- система безопасности/защита от взлома;

f) требования пользовательского интерфейса, реализованные в программном обеспечении.

Примечание 4 — Примеры в этой области связаны:
- с поддержкой операций, выполняемых вручную;
- взаимодействием между человеком и оборудованием;
- ограничениями в отношении персонала;
- областями, где требуется пристальное человеческое внимание.

Примечание 5 — Информацию относительно требований к разработке удобства и простоты использования (эксплуатационной пригодности) можно найти в IEC 62366-1 [21] среди прочих стандартов (например, IEC 60601-1-6 [3]);

g) определение данных и требований к базе данных.

Примечание 6 — Например:
- форма;
- размерность;
- функция;

h) требования по инсталляции и приемке поставляемого ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКИХ ИЗДЕЛИЙ для разработки и технической поддержки сайта или сайтов;

i) требования, относящиеся к методам разработки и технической поддержки;

j) требования, связанные с аспектами информационных сетей.

Примечание 9 — Примеры включают аспекты, которые связаны:
- с сетевыми сигналами тревоги, предупреждения и сообщения оператора;
- сетевыми протоколами;
- обработкой недоступности сетевых услуг;

k) требования к поддержке пользователей;

l) регулирующие требования.

Примечание 10 — Требования с a) до l) могут пересекаться.

[Классы A, B, C]

Примечание 11 — Все эти требования могут не иметься в наличии на момент начала разработки.

Примечание 12 — Среди прочих, ISO/IEC 25010 [12] приводит информацию о качественных характеристиках, которая может быть полезна при определении требований к программному обеспечению.

5.2.3 Включение мер УПРАВЛЕНИЯ РИСКОМ в требования к программному обеспечению

ИЗГОТОВИТЕЛЬ должен включать меры по УПРАВЛЕНИЮ РИСКОМ, реализованные в программном обеспечении в требования, соответствующие ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ МЕДИЦИНСКИХ ИЗДЕЛИЙ. [Классы B, C]

Примечание — Эти требования могут быть недоступны в начале процесса разработки и изменяться по мере того, как создается программное обеспечение и устанавливаются дальнейшие меры по УПРАВЛЕНИЮ РИСКОМ.

5.2.4 ПЕРЕОЦЕНИВАНИЕ АНАЛИЗА РИСКА МЕДИЦИНСКОГО ИЗДЕЛИЯ

ИЗГОТОВИТЕЛЬ должен осуществить ПЕРЕОЦЕНИВАНИЕ АНАЛИЗА РИСКА МЕДИЦИНСКОГО ИЗДЕЛИЯ, когда требования к программному обеспечению установлены, и, соответственно, обновить эти требования по результатам переоценки. [Классы A, B, C]

5.2.5 Обновление требований

ИЗГОТОВИТЕЛЬ должен удостовериться, что существующие требования, включая требования к СИСТЕМЕ, ПЕРЕОЦЕНЕНЫ и обновлены, в соответствии с результатами ДЕЯТЕЛЬНОСТИ по анализу требований к программному обеспечению. [Классы A, B, C]

5.2.6 Верификация требований к программному обеспечению

ИЗГОТОВИТЕЛЬ должен верифицировать и документировать, что требования к программному обеспечению:

a) включают требования к СИСТЕМЕ, в том числе относящиеся к УПРАВЛЕНИЮ РИСКОМ;

b) не противоречат друг другу;

c) выражены в терминах, которые избегают двусмысленности;

d) формулируются в терминах, которые позволяют установить критерии тестирования и осуществить тестирование;

e) могут быть идентифицированы уникальным образом;

f) являются прослеживаемыми в отношении требований к СИСТЕМЕ или к другому источнику.

[Классы A, B, C]

Примечание — Настоящий стандарт не требует использования формально установленного языка.

5.3 Проектирование АРХИТЕКТУРЫ программного обеспечения

5.3.1 Преобразование требований к программному обеспечению в АРХИТЕКТУРУ

ИЗГОТОВИТЕЛЬ должен преобразовать требования к ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ МЕДИЦИНСКОГО ИЗДЕЛИЯ в документированную АРХИТЕКТУРУ, которая описывает структуру программного обеспечения и идентифицирует ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ. [Классы B, C]

5.3.2 Разработка АРХИТЕКТУРЫ для интерфейсов ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ

ИЗГОТОВИТЕЛЬ должен разработать и документировать АРХИТЕКТУРУ для интерфейсов между ПРОГРАММНЫМИ СОСТАВНЫМИ ЧАСТЯМИ и компонентами, внешними по отношению к ПРОГРАММНЫМ СОСТАВНЫМ ЧАСТЯМ (как для программной, так и для аппаратной части), а также между ПРОГРАММНЫМИ СОСТАВНЫМИ ЧАСТЯМИ. [Классы B, C]

5.3.3 Определение требований к функциональным и эксплуатационным характеристикам элементов ПОНП

Если ПРОГРАММНАЯ СОСТАВНАЯ ЧАСТЬ идентифицирован как ПОНП, ИЗГОТОВИТЕЛЬ должен определить требования к функциональным и эксплуатационным характеристикам элементов ПОНП, которые необходимы для их использования согласно предусмотренному назначению. [Классы B, C]

5.3.4 Определение требований к аппаратным и программным средствам СИСТЕМЫ, требуемых элементами ПОНП

Если ПРОГРАММНАЯ СОСТАВНАЯ ЧАСТЬ определяется как ПОНП, ИЗГОТОВИТЕЛЬ должен определить аппаратные и программные средства СИСТЕМЫ, необходимые для поддержания правильной работы элемента ПОНП. [Классы B, C]

Примечание — Примеры включают тип и скорость процессора, тип и размер памяти, тип программного обеспечения СИСТЕМЫ, коммуникационные и дисплейные требования.

5.3.5 Идентификация обособленности, необходимой для УПРАВЛЕНИЯ РИСКОМ

ИЗГОТОВИТЕЛЬ должен идентифицировать любую обособленность ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ, которая необходима для УПРАВЛЕНИЯ РИСКОМ, и указать, как обеспечить результативность созданной обособленности. [Класс C]

Примечание — В качестве примера создания обособленности можно привести ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ, выполняемые на разных процессорах. Результативность обособленности может быть обеспечена за счет отсутствия общих ресурсов у разных процессоров. Другие способы создания обособленности могут применяться, когда результативность может быть обеспечена посредством разработки АРХИТЕКТУРЫ программного обеспечения (см. B. 4.3).

5.3.6 Верификация АРХИТЕКТУРЫ программного обеспечения

ИЗГОТОВИТЕЛЬ должен верифицировать и документировать, что:

a) АРХИТЕКТУРА программного обеспечения реализует требования к СИСТЕМЕ и к программному обеспечению, включая относящиеся к УПРАВЛЕНИЮ РИСКОМ;

b) АРХИТЕКТУРА программного обеспечения способна поддерживать взаимодействие между ПРОГРАММНЫМИ СОСТАВНЫМИ ЧАСТЯМИ, а также между ПРОГРАММНЫМИ СОСТАВНЫМИ ЧАСТЯМИ и аппаратными средствами;

c) АРХИТЕКТУРА МЕДИЦИНСКОГО ИЗДЕЛИЯ поддерживает правильную работу любых элементов ПОНП. [Классы B, C]

Примечание — Для выполнения требования a) может быть использован анализ ПРОСЛЕЖИВАЕМОСТИ АРХИТЕКТУРЫ к требованиям по программному обеспечению.

5.4* Разработка детального дизайна программного обеспечения

5.4.1 Дробление программного обеспечения на ПРОГРАММНЫЕ БЛОКИ

ИЗГОТОВИТЕЛЬ должен дробить программное обеспечение, пока оно не будет представлено в виде ПРОГРАММНЫХ БЛОКОВ. [Классы B, C]

Примечание — Некоторые ПРОГРАММНЫЕ СИСТЕМЫ далее не могут быть раздроблены.

5.4.2 Разработка детального дизайна для каждого ПРОГРАММНОГО БЛОКА

ИЗГОТОВИТЕЛЬ должен документировать дизайн с достаточной степенью детализации, с целью обеспечения правильной реализации каждого ПРОГРАММНОГО БЛОКА. [Класс C]

5.4.3 Разработка детального дизайна для интерфейсов

ИЗГОТОВИТЕЛЬ должен документировать дизайн любых интерфейсов между ПРОГРАММНЫМ БЛОКОМ и внешними компонентами (аппаратными или программными средствами), а также любых интерфейсов между ПРОГРАММНЫМИ БЛОКАМИ, который должен быть достаточно подробным для правильной реализации каждого ПРОГРАММНОГО БЛОКА и его интерфейсов. [Класс C]

5.4.4 ВЕРИФИКАЦИЯ детального дизайна

ИЗГОТОВИТЕЛЬ должен верифицировать и документировать, что детальный дизайн программного обеспечения:

a) реализует АРХИТЕКТУРУ программного обеспечения;

b) не вступает в противоречия с АРХИТЕКТУРОЙ программного обеспечения. [Класс C]

Примечание — Для выполнения требования a) может быть использован анализ ПРОСЛЕЖИВАЕМОСТИ АРХИТЕКТУРЫ к детальному дизайну программного обеспечения.

5.5* Имплементация ПРОГРАММНЫХ БЛОКОВ

5.5.1 Имплементация каждого ПРОГРАММНОГО БЛОКА

ИЗГОТОВИТЕЛЬ должен имплементировать каждый ПРОГРАММНЫЙ БЛОК. [Классы A, B, C]

5.5.2 Установление ПРОЦЕССА ВЕРИФИКАЦИИ ПРОГРАММНОГО БЛОКА

ИЗГОТОВИТЕЛЬ должен установить стратегии, методы и процедуры для верификации ПРОГРАММНЫХ БЛОКОВ. Там, где ВЕРИФИКАЦИЯ осуществляется посредством тестирования, правильность процедур проведения тестирования должна быть ОЦЕНЕНА на адекватность. [Классы B, C]

Примечание — Допускается объединение интеграционного тестирования и тестирование ПРОГРАММНОЙ СИСТЕМЫ в единый план и совокупность ДЕЯТЕЛЬНОСТИ.

5.5.3 Критерии приемки ПРОГРАММНЫХ БЛОКОВ

ИЗГОТОВИТЕЛЬ должен установить критерии приемки для ПРОГРАММНЫХ БЛОКОВ до их интеграции в более крупные ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ и удостовериться, что ПРОГРАММНЫЕ БЛОКИ соответствуют критериям приемки. [Классы B, C]

Примечание — Примеры критериев приемки:

- Имплементированы ли требования в программном коде, включая меры по УПРАВЛЕНИЮ РИСКОМ?
- Нет ли в программном коде противоречий с проектом (дизайном) интерфейса ПРОГРАММНЫХ БЛОКОВ?
- Соответствует ли программный код процедурам программирования или стандартам кодирования?

5.5.4 Дополнительные критерии приемки ПРОГРАММНЫХ БЛОКОВ

Если детальный дизайн разработан, ИЗГОТОВИТЕЛЬ должен включить в дизайн дополнительные критерии приемки, предназначенные:

- для правильной (соответствующей) последовательности событий;
- потока данных и управления;
- планируемого распределения ресурсов;
- работы с ошибками (определение ошибки, локализация и восстановление);
- инициализации переменных;
- самодиагностики;
- управления памятью и переполнений памяти;
- граничных условий.

[Класс C]

5.5.5 ВЕРИФИКАЦИЯ ПРОГРАММНЫХ БЛОКОВ

ИЗГОТОВИТЕЛЬ должен выполнять ВЕРИФИКАЦИЮ ПРОГРАММНЫХ БЛОКОВ и документировать результаты. [Классы B, C]

5.6 Интеграция программного обеспечения и тестирование интеграции

5.6.1 Интеграция ПРОГРАММНЫХ БЛОКОВ

ИЗГОТОВИТЕЛЬ должен интегрировать ПРОГРАММНЫЕ БЛОКИ согласно плану интеграции (см. 5.1.5). [Классы B, C]

5.6.2 Верификация интеграции программного обеспечения

ИЗГОТОВИТЕЛЬ должен верифицировать, что ПРОГРАММНЫЕ БЛОКИ были интегрированы в ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ и/или ПРОГРАММНУЮ СИСТЕМУ в соответствии с планом интеграции (см. 5.1.5), а также сохранить записи, свидетельствующие о проведении такой верификации. [Классы B, C]

Примечание — Данная верификация заключается только в проверке выполнения интеграции в соответствии с планом. Эта ВЕРИФИКАЦИЯ, скорее всего, осуществляется в форме какого-либо контрольного мероприятия.

5.6.3 Интеграционное тестирование программного обеспечения

ИЗГОТОВИТЕЛЬ должен тестировать интегрированные ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ в соответствии с планом интеграции (см. 5.1.5) и документировать полученные результаты. [Классы B, C]

5.6.4 Содержание тестирования интеграции программного обеспечения

При тестировании интеграции программного обеспечения ИЗГОТОВИТЕЛЬ должен установить, что интегрированная ПРОГРАММНАЯ СОСТАВНАЯ ЧАСТЬ функционирует в соответствии с предусмотренным назначением. [Классы B, C]

Примечание 1 — Примерами могут служить:

- требуемая функциональность программного обеспечения;
- выполнение мер по УПРАВЛЕНИЮ РИСКОМ;
- определенная синхронизация и другие режимы работы;
- определенное функционирование внутренних и внешних интерфейсов;
- тестирование в ненормальных условиях, включая обоснованно прогнозируемое неправильное применение.

Примечание 2 — Возможно объединять тестирование интеграции и тестирование ПРОГРАММНОЙ СИСТЕМЫ в единый план и совокупность ДЕЯТЕЛЬНОСТИ.

5.6.5 Оценивание процедур тестирования интеграции программного обеспечения

ИЗГОТОВИТЕЛЬ должен ОЦЕНИВАТЬ процедуры тестирования интеграции на адекватность. [Классы B, C]

5.6.6 Проведение регрессионного тестирования

По завершении интеграции программных составных частей, ИЗГОТОВИТЕЛЬ должен провести РЕГРЕССИОННОЕ ТЕСТИРОВАНИЕ, подходящее для демонстрации того, что в ранее интегрированном программном обеспечении не были обнаружены дефекты. [Классы B, C]

5.6.7 Содержание записей в отношении регрессионного тестирования

ИЗГОТОВИТЕЛЬ должен:

a) документировать результаты тестирования (соответствует, не соответствует и перечень АНОМАЛИЙ);

b) сохранить существенные записи с целью сделать возможным повторное тестирование;

c) указать лицо, которое проводило тестирование.

[Классы B, C]

Примечание — Требование b) может быть выполнено путем сохранения, например:

- спецификаций тестового примера, показывающих требуемые действия и ожидаемые результаты;
- записей об оборудовании;
- записей о тестовом окружении (включая программные инструменты), используемом при проведении тестирования.

5.6.8 Использование ПРОЦЕССА решения проблем с программным обеспечением

ИЗГОТОВИТЕЛЬ должен вводить АНОМАЛИИ, обнаруженные во время интеграции программного обеспечения и тестирования интеграции, в ПРОЦЕСС решения проблем с программным обеспечением. [Классы B, C]

Примечание — см. раздел 9.

5.7 Тестирование ПРОГРАММНОЙ СИСТЕМЫ

5.7.1 Установление тестирования в отношении требований к программному обеспечению

a) Для проведения тестирования ПРОГРАММНОЙ СИСТЕМЫ ИЗГОТОВИТЕЛЬ должен установить и выполнить перечень тестов, выраженных как входные данные, ожидаемые результаты, критерии приемки и процедуры, с целью учета и охвата тестированием всех требований к программному обеспечению. [Классы A, B, C]

Примечание 1 — Допускается объединять тестирование интеграции и тестирование ПРОГРАММНОЙ СИСТЕМЫ в единый план и совокупность ДЕЯТЕЛЬНОСТИ. Также допустимо тестировать программное обеспечение на более ранних стадиях.

Примечание 2 — Могут проводиться не только тестирования отдельных требований, но и тестирования комбинаций требований, особенно если между требованиями существуют зависимости.

b) ИЗГОТОВИТЕЛЬ должен ОЦЕНИТЬ адекватность стратегий проведения ВЕРИФИКАЦИИ и тестовых процедур.

5.7.2 Применение ПРОЦЕССА решения проблем с программным обеспечением

ИЗГОТОВИТЕЛЬ должен ввести АНОМАЛИИ, обнаруженные во время испытаний ПРОГРАММНОЙ СИСТЕМЫ, в ПРОЦЕСС решения проблем с программным обеспечением. [Классы A, B, C]

5.7.3 Повторное тестирование после внесения изменений

При внесении изменений в ходе тестирования ПРОГРАММНОЙ СИСТЕМЫ ИЗГОТОВИТЕЛЬ должен:

a) повторить тестирование, выполнить модифицированные тесты или дополнительные тесты, если применимо, с целью проверки результативности вносимых изменений для исправления проблем;

b) провести соответствующее тестирование, необходимое для демонстрации отсутствия возникновения непреднамеренных побочных эффектов;

c) выполнить соответствующую ДЕЯТЕЛЬНОСТЬ по МЕНЕДЖМЕНТУ РИСКА, как установлено в 7.4.

[Классы A, B, C]

5.7.4 Оценивание тестирования ПРОГРАММНОЙ СИСТЕМЫ

ИЗГОТОВИТЕЛЬ должен ОЦЕНИТЬ целесообразность стратегий ВЕРИФИКАЦИИ и процедур тестирования.

ИЗГОТОВИТЕЛЬ должен проверить, что:

a) все требования к программному обеспечению были протестированы или иным образом ВЕРИФИЦИРОВАНЫ;

b) ведутся записи по ПРОСЛЕЖИВАЕМОСТИ между требованиями к программному обеспечению и тестами или другой ВЕРИФИКАЦИЕЙ;

c) результаты тестирования соответствуют требуемым критериям приемки.

[Классы A, B, C]

5.7.5 Содержание отчета по тестированию ПРОГРАММНОЙ СИСТЕМЫ

Для обеспечения повторяемости тестов ИЗГОТОВИТЕЛЬ должен документировать:

a) ссылки на конкретные процедуры тестирования с указанием требуемых действий и ожидаемых результатов;

b) результаты тестирования (соответствует, не соответствует и список АНОМАЛИЙ);

c) версию тестируемого программного обеспечения;

d) соответствующие конфигурации тестируемого аппаратного и программного обеспечения;

e) соответствующие средства тестирования;

f) дату выполненного тестирования;

g) идентификацию лица, ответственного за проведение тестирования и запись его результатов.

[Классы A, B, C]

5.8* Выпуск программного обеспечения на системном уровне

5.8.1 Обеспечение завершенности ВЕРИФИКАЦИИ программного обеспечения

ИЗГОТОВИТЕЛЬ до выпуска ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ в обращение должен обеспечить, чтобы ДЕЯТЕЛЬНОСТЬ по ВЕРИФИКАЦИИ всего программного обеспечения была полностью завершена, а результаты были ОЦЕНЕНЫ. [Классы A, B, C]

5.8.2 Документирование известных остаточных АНОМАЛИЙ

ИЗГОТОВИТЕЛЬ должен задокументировать все известные остаточные АНОМАЛИИ. [Классы A, B, C]

5.8.3 ОЦЕНИВАНИЕ известных остаточных АНОМАЛИЙ

ИЗГОТОВИТЕЛЬ должен обеспечить, чтобы все известные остаточные АНОМАЛИИ были ОЦЕНЕНЫ, с целью обеспечения отсутствия их способности содействовать возникновению недопустимых РИСКОВ. [Классы B, C]

5.8.4 Документирование выпущенных ВЕРСИЙ

ИЗГОТОВИТЕЛЬ должен документировать ВЕРСИЮ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКОГО ИЗДЕЛИЯ, которая будет выпускаться. [Классы A, B, C]

5.8.5 Документирование создания выпущенного программного обеспечения

ИЗГОТОВИТЕЛЬ должен документировать процедуру и окружение (среду), используемые для создания выпущенного программного обеспечения. [Классы B, C]

5.8.6 Обеспечение полного завершения деятельности и задач

ИЗГОТОВИТЕЛЬ должен обеспечить выполнение всей ДЕЯТЕЛЬНОСТИ и всех ЗАДАЧ, входящих в состав плана разработки (или плана технической поддержки) программного обеспечения, наряду со связанной с ними документацией. [Классы B, C]

Примечание — См. 5.1.3.b).

5.8.7 Архивирование программного обеспечения

Изготовитель должен хранить в архиве:

a) ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МЕДИЦИНСКОГО ИЗДЕЛИЯ и СОСТАВНОЙ ЧАСТИ КОНФИГУРАЦИИ;

b) документацию

в течение как минимум срока службы ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКОГО ИЗДЕЛИЯ, установленного ИЗГОТОВИТЕЛЕМ, или в течение срока, установленного соответствующими регулирующими требованиями.

[Классы A, B, C]

5.8.8 Обеспечение надежной поставки выпущенного программного обеспечения

ИЗГОТОВИТЕЛЬ должен установить процедуры, обеспечивающие, чтобы выпущенное ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МЕДИЦИНСКОГО ИЗДЕЛИЯ было поставлено пользователю (к месту его применения) без искажения или несанкционированного изменения. Эти процедуры должны распространяться на производство и обращение со средствами, содержащими ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МЕДИЦИНСКОГО ИЗДЕЛИЯ, и включать, если применимо:

- создание копии;
- средства маркировки;
- упаковку;
- защиту;
- хранение;
- поставку.

[Классы A, B, C]

6 Техническая поддержка программного обеспечения

6.1* Установление плана технической поддержки программного обеспечения

ИЗГОТОВИТЕЛЬ должен установить план (или планы) технической поддержки программного обеспечения для выполнения ДЕЯТЕЛЬНОСТИ и ЗАДАЧ ПРОЦЕССА технической поддержки. Этот план должен содержать:

a) процедуры:

- для получения (установления),

- документирования,
- оценивания,
- исправления,
- отслеживания

по обратной связи, возникающей (устанавливаемой) после выпуска ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКИХ ИЗДЕЛИЙ;

b) критерии для определения того, что обратная связь является проблемой;

c) использование ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА программного обеспечения;

d) использование ПРОЦЕССА решения проблем программного обеспечения для анализа и принятия решений по проблемам, возникающим после выпуска ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКИХ ИЗДЕЛИЙ;

e) использование ПРОЦЕССА менеджмента конфигурации программного обеспечения (раздел 8) для управления модификациями существующей ПРОГРАММНОЙ СИСТЕМЫ;

f) процедуры по ОЦЕНИВАНИЮ и проведению:

- обновления,
- исправления ошибок,
- исправлений, вносимых в коды («заплатки», «патчи»),
- признания программного обеспечения устаревшим, обращения с ПОНП.

[Классы A, B, C]

6.2* Анализ модификации и проблем

6.2.1 Документирование и оценивание обратной связи

6.2.1.1 Мониторинг обратной связи

ИЗГОТОВИТЕЛЬ должен осуществлять мониторинг обратной связи ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКОГО ИЗДЕЛИЯ, выпущенного для использования по назначению. [Классы A, B, C]

6.2.1.2 Документирование и ОЦЕНИВАНИЕ обратной связи

Обратная связь должна быть документирована и ОЦЕНЕНА с целью определения существования проблемы в выпущенном ПРОГРАММНОМ ОБЕСПЕЧЕНИИ МЕДИЦИНСКОГО ИЗДЕЛИЯ. Любая такая проблема должна быть зарегистрирована в ОТЧЕТЕ О ПРОБЛЕМАХ (см. раздел 9). ОТЧЕТ О ПРОБЛЕМАХ должен содержать фактические или возможные неблагоприятные события и отклонения от спецификации. [Классы A, B, C]

6.2.1.3 ОЦЕНИВАНИЕ влияния ОТЧЕТОВ О ПРОБЛЕМАХ на БЕЗОПАСНОСТЬ

Каждый ОТЧЕТ О ПРОБЛЕМАХ должен быть ОЦЕНЕН с целью определения его влияния на БЕЗОПАСНОСТЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКОГО ИЗДЕЛИЯ, выпущенного для использования по назначению (см. 9.2), и требуется ли изменение этого программного обеспечения для решения проблемы. [Классы A, B, C]

6.2.2 Использование ПРОЦЕССА решения проблем программного обеспечения

ИЗГОТОВИТЕЛЬ должен использовать ПРОЦЕСС решения проблем программного обеспечения (см. раздел 9) в отношении ОТЧЕТОВ О ПРОБЛЕМАХ. [Классы A, B, C]

Примечание — Проблема может указывать на то, что ПРОГРАММНАЯ СИСТЕМА или ПРОГРАММНАЯ СОСТАВНАЯ ЧАСТЬ не были правильно отнесены к классу безопасности программного обеспечения. Процесс решения проблемы может состоять в изменении класса безопасности программного обеспечения. После завершения ПРОЦЕССА любое изменение класса безопасности ПРОГРАММНОЙ СИСТЕМЫ или ее ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ должно быть известно и документировано.

6.2.3 Анализ ЗАПРОСОВ НА ИЗМЕНЕНИЕ

В дополнение к анализу, требуемому разделом 9, ИЗГОТОВИТЕЛЬ должен анализировать каждый ЗАПРОС НА ИЗМЕНЕНИЕ с целью определения его влияния на организацию, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МЕДИЦИНСКОГО ИЗДЕЛИЯ, выпущенного для использования по назначению, и СИСТЕМЫ, с которыми оно взаимодействует. [Классы A, B, C]

6.2.4 Одобрение ЗАПРОСА НА ИЗМЕНЕНИЕ

ИЗГОТОВИТЕЛЬ должен ОЦЕНИТЬ и одобрить ЗАПРОСЫ НА ИЗМЕНЕНИЯ, которые модифицируют выпущенное ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МЕДИЦИНСКОГО ИЗДЕЛИЯ. [Классы A, B, C]

6.2.5 Информирование пользователей и регулирующих органов

ИЗГОТОВИТЕЛЬ должен идентифицировать одобренные ЗАПРОСЫ НА ИЗМЕНЕНИЯ, которые влияют на выпущенное ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МЕДИЦИНСКОГО ИЗДЕЛИЯ.

Если требуется региональным регулированием, ИЗГОТОВИТЕЛЬ должен информировать пользователей и регулирующие органы:

a) о любых проблемах в отношении выпущенного ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКОГО ИЗДЕЛИЯ и последствиях длительного использования неизмененного продукта;

b) о характере любых доступных изменений в выпущенном ПРОГРАММНОМ ОБЕСПЕЧЕНИИ МЕДИЦИНСКОГО ИЗДЕЛИЯ и о том, как получить и установить эти изменения.

[Классы A, B, C]

6.3* Осуществление модификации

6.3.1 Использование установленного ПРОЦЕССА осуществления модификации

ИЗГОТОВИТЕЛЬ должен идентифицировать и выполнять любую ДЕЯТЕЛЬНОСТЬ, указанную в разделе 5, которую необходимо повторить в результате модификации. [Классы A, B, C]

Примечание — Требования в отношении МЕНЕДЖМЕНТА РИСКА для изменений программного обеспечения см. в 7.4.

6.3.2 Повторный выпуск модифицированной ПРОГРАММНОЙ СИСТЕМЫ

ИЗГОТОВИТЕЛЬ должен выпускать модификации согласно 5.8. [Классы A, B, C]

Примечание — Модификации могут быть реализованы как часть полной повторно выпущенной ПРОГРАММНОЙ СИСТЕМЫ или как набор модификаций, включающий измененные ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ, а также инструменты, необходимые для установки изменений как модификации существующей ПРОГРАММНОЙ СИСТЕМЫ.

7 ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА программного обеспечения

7.1* Анализ программного обеспечения, способствующего опасным ситуациям

7.1.1 Идентификация ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ, которые могут способствовать возникновению опасных ситуаций

ИЗГОТОВИТЕЛЬ должен идентифицировать ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ, которые могут способствовать возникновению опасных ситуаций, идентифицированных при осуществлении ДЕЯТЕЛЬНОСТИ по АНАЛИЗУ РИСКА МЕДИЦИНСКОГО ИЗДЕЛИЯ, которая должна проводиться согласно ISO 14971 (см. 4.2). [Классы B, C]

Примечание — Опасные ситуации могут являться прямым следствием отказа ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ или возникнуть в результате отказа мер по УПРАВЛЕНИЮ РИСКАМИ, которые включены в программное обеспечение.

7.1.2 Идентификация потенциальных причин, способствующих возникновению опасных ситуаций

ИЗГОТОВИТЕЛЬ должен идентифицировать потенциальные причины, по которым указанные в предыдущем пункте ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ могут способствовать возникновению опасных ситуаций.

ИЗГОТОВИТЕЛЬ должен рассмотреть потенциальные причины, включая, если применимо:

- неправильную или неполную спецификацию функциональности;
- дефекты программного обеспечения, идентифицированные в определенных функциях ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ;
- отказы или неожидаемые результаты, исходящие от ПОНП;
- отказы аппаратных средств или другие дефекты ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, которые могут привести к непредсказуемым операциям программного обеспечения;
- обоснованно прогнозируемое неправильное применение.

[Классы B, C]

7.1.3 ОЦЕНИВАНИЕ опубликованных списков АНОМАЛИЙ ПОНП

Если отказ или исходящие от ПОНП неожидаемые результаты являются потенциальной причиной того, что ПРОГРАММНАЯ СОСТАВНАЯ ЧАСТЬ может способствовать возникновению опасных ситуаций, то ИЗГОТОВИТЕЛЬ должен ОЦЕНИВАТЬ как минимум любой список АНОМАЛИЙ, опубликованный поставщиком элементов ПОНП, используемых в МЕДИЦИНСКОМ ИЗДЕЛИИ, чтобы определить,

приводит ли любая из известных АНОМАЛИЙ к последовательности событий, которые могут привести к опасной ситуации. [Классы B, C]

7.1.4 Документирование потенциальных причин

ИЗГОТОВИТЕЛЬ должен документировать в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА потенциальные причины, по которым ПРОГРАММНАЯ СОСТАВНАЯ ЧАСТЬ может способствовать возникновению опасной ситуации (см. ISO 14971). [Классы B, C]

7.2 Меры по УПРАВЛЕНИЮ РИСКОМ

7.2.1 Определение мер по УПРАВЛЕНИЮ РИСКОМ

В отношении каждого случая, зарегистрированного в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА, при котором ПРОГРАММНАЯ СОСТАВНАЯ ЧАСТЬ может способствовать возникновению ОПАСНОЙ СИТУАЦИИ, ИЗГОТОВИТЕЛЬ должен определить и документировать меры по УПРАВЛЕНИЮ РИСКОМ в соответствии с ISO 14971. [Классы B, C]

Примечание — Меры по УПРАВЛЕНИЮ РИСКОМ могут быть реализованы в аппаратных средствах, программном обеспечении, рабочей среде или в инструкциях пользователя.

7.2.2 Меры по УПРАВЛЕНИЮ РИСКОМ, реализованные в программном обеспечении

Если мера по УПРАВЛЕНИЮ РИСКОМ реализуются как часть функций ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ, то ИЗГОТОВИТЕЛЬ должен:

a) включить меру по УПРАВЛЕНИЮ РИСКОМ в требования к программному обеспечению;

b) назначить каждой ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ, которая способствует реализации меры по УПРАВЛЕНИЮ РИСКОМ, класс безопасности программного обеспечения, основанный на РИСКЕ, которым управляет данная мера по УПРАВЛЕНИЮ РИСКОМ (см. 4.3 a);

c) разработать ПРОГРАММНУЮ СОСТАВНУЮ ЧАСТЬ в соответствии с разделом 5.

[Классы B, C]

Примечание — Это требование обеспечивает дополнительное уточнение требований по УПРАВЛЕНИЮ РИСКОМ в ISO 14971.

7.3 Верификация мер по УПРАВЛЕНИЮ РИСКОМ

7.3.1 Проведение верификации мер по УПРАВЛЕНИЮ РИСКОМ

Выполнение каждой меры по УПРАВЛЕНИЮ РИСКОМ, документированной в 7.2, должно быть верифицировано, а сама ВЕРИФИКАЦИЯ должна быть документирована. ИЗГОТОВИТЕЛЬ должен проанализировать меры по УПРАВЛЕНИЮ РИСКАМИ и определить их способность привести к возникновению новой ОПАСНОЙ СИТУАЦИИ. [Классы B, C]

7.3.2 Документирование любых новых последовательностей событий

Не применяется.

7.3.3 Документирование ПРОСЛЕЖИВАЕМОСТИ

ИЗГОТОВИТЕЛЬ должен соответствующим образом документировать ПРОСЛЕЖИВАЕМОСТЬ в отношении ОПАСНОСТЕЙ программного обеспечения:

- от опасной ситуации до ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ;
- от ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ до конкретной причины в программном обеспечении;
- от причины в программном обеспечении до меры по УПРАВЛЕНИЮ РИСКОМ;
- от меры по УПРАВЛЕНИЮ РИСКОМ до ВЕРИФИКАЦИИ меры по УПРАВЛЕНИЮ РИСКОМ.

[Классы B, C]

Примечание — См. ISO 14971:2007, отчет по МЕНЕДЖМЕНТУ РИСКА.

7.4 МЕНЕДЖМЕНТ РИСКА в отношении изменений программного обеспечения

7.4.1 Анализ изменений ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКОГО ИЗДЕЛИЯ в отношении БЕЗОПАСНОСТИ

ИЗГОТОВИТЕЛЬ обязан анализировать изменения в ПРОГРАММНОМ ОБЕСПЕЧЕНИИ МЕДИЦИНСКОГО ИЗДЕЛИЯ (включая ПОНП) с целью определения:

- существования не выявленных ранее причин, способствующих возникновению опасной ситуации;
- требуются ли дополнительные программные меры по УПРАВЛЕНИЮ РИСКОМ.

[Классы A, B, C]

7.4.2 Анализ влияния изменений программного обеспечения на выполненные меры по УПРАВЛЕНИЮ РИСКОМ

ИЗГОТОВИТЕЛЬ должен анализировать изменения программного обеспечения, включая изменения ПОНП, с целью определения возможности конфликта модифицированного программного обеспечения и выполненных мер по УПРАВЛЕНИЮ РИСКОМ. [Классы B, C]

7.4.3 Осуществление ДЕЯТЕЛЬНОСТИ по МЕНЕДЖМЕНТУ РИСКА, основанной на результатах анализа

ИЗГОТОВИТЕЛЬ должен осуществить уместную ДЕЯТЕЛЬНОСТЬ по МЕНЕДЖМЕНТУ РИСКА, которая определена в 7.1, 7.2 и 7.3, основанную на результатах проведенного анализа. [Классы B, C]

8* ПРОЦЕСС менеджмента конфигурации программного обеспечения

8.1* Идентификация конфигурации

8.1.1 Установление средств идентификации СОСТАВНОЙ ЧАСТИ КОНФИГУРАЦИИ

ИЗГОТОВИТЕЛЬ должен установить схему уникальной идентификации подлежащих управлению СОСТАВНЫХ ЧАСТЕЙ КОНФИГУРАЦИИ и их ВЕРСИЙ в соответствии с планированием разработки и конфигурации, установленной в 5.1. [Классы A, B, C]

8.1.2 Идентификация ПОНП

Для каждой СОСТАВНОЙ ЧАСТИ КОНФИГУРАЦИИ ПОНП, который будет использоваться, включая библиотеки стандартов, ИЗГОТОВИТЕЛЬ должен документировать:

a) наименование;

b) ИЗГОТОВИТЕЛЯ;

c) уникальный указатель (обозначение) ПОНП.

[Классы A, B, C]

Примечание — Уникальным указателем ПОНП может быть, например, ВЕРСИЯ, дата выпуска, номер патча или обозначение модернизации.

8.1.3 Идентификация документации конфигурации СИСТЕМЫ

ИЗГОТОВИТЕЛЬ должен документировать набор СОСТАВНЫХ ЧАСТЕЙ КОНФИГУРАЦИИ и их ВЕРСИЙ, входящих в состав конфигурации ПРОГРАММНОЙ СИСТЕМЫ.

[Классы A, B, C]

8.2* Управление изменениями

8.2.1 Одобрение ЗАПРОСОВ НА ИЗМЕНЕНИЯ

ИЗГОТОВИТЕЛЬ может изменять СОСТАВНЫЕ ЧАСТИ КОНФИГУРАЦИИ, идентифицированные как подлежащие управлению согласно 8.1, только после того, как будет одобрен ЗАПРОС НА ИЗМЕНЕНИЯ. [Классы A, B, C]

Примечание 1 — Решение одобрить ЗАПРОС НА ИЗМЕНЕНИЯ может быть частью ПРОЦЕССА управления изменениями или частью другого ПРОЦЕССА. Этот подпункт требует только того, чтобы одобрение изменения предшествовало его выполнению.

Примечание 2 — В отношении ЗАПРОСОВ НА ИЗМЕНЕНИЯ на разных стадиях жизненного цикла могут использоваться различные ПРОЦЕССЫ одобрения, как это установлено в планах, см. 5.1.1 d) и 6.1 e).

8.2.2 Осуществление изменений

ИЗГОТОВИТЕЛЬ должен осуществить изменение так, как это определено в ЗАПРОСЕ НА ИЗМЕНЕНИЯ. ИЗГОТОВИТЕЛЬ должен идентифицировать и выполнить любую ДЕЯТЕЛЬНОСТЬ, которую нужно повторить из-за произведенных изменений, включая изменение класса безопасности ПРОГРАММНЫХ СИСТЕМ и ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ. [Классы A, B, C]

Примечание — Данный подпункт устанавливает, как изменение должно быть реализовано для обеспечения надлежащего управления изменениями. Это не означает, что внедрение (реализация) является неотъемлемой частью ПРОЦЕССА управления изменениями. При внедрении должны использоваться запланированные ПРОЦЕССЫ, см. 5.1.1 e) и 6.1 e).

8.2.3 ВЕРИФИКАЦИЯ изменений

ИЗГОТОВИТЕЛЬ должен проверять изменения, включая повторение любой ВЕРИФИКАЦИИ, которая стала недействительной после внесения изменений, а также уделить внимание 5.7.2 и 9.7. [Классы A, B, C]

Примечание — Данный подпункт требует только ВЕРИФИКАЦИИ изменений. Он не подразумевает того, что ВЕРИФИКАЦИЯ — неотъемлемая часть ПРОЦЕССА управления изменениями. ВЕРИФИКАЦИЯ должна использовать запланированные ПРОЦЕССЫ, см. 5.1.1 д) и 6.1 е).

8.2.4 Обеспечение средствами для ПРОСЛЕЖИВАЕМОСТИ изменений

ИЗГОТОВИТЕЛЬ должен поддерживать записи о взаимосвязях и зависимостях между:

a) ЗАПРОСАМИ НА ИЗМЕНЕНИЕ,

b) соответствующими ОТЧЕТАМИ О ПРОБЛЕМАХ,

c) одобрениями ЗАПРОСА НА ИЗМЕНЕНИЕ.

[Классы A, B, C]

8.3* Учет статуса конфигурации

ИЗГОТОВИТЕЛЬ должен сохранять восстанавливаемые записи об истории управляемых СОСТАВНЫХ ЧАСТЕЙ КОНФИГУРАЦИИ, включая конфигурацию СИСТЕМЫ. [Классы A, B, C]

9 ПРОЦЕСС решения проблем программного обеспечения

9.1 Подготовка ОТЧЕТОВ О ПРОБЛЕМАХ

ИЗГОТОВИТЕЛЬ должен подготовить ОТЧЕТ О ПРОБЛЕМАХ в отношении каждой проблемы, обнаруженной в ПРОГРАММНОМ ОБЕСПЕЧЕНИИ МЕДИЦИНСКОГО ИЗДЕЛИЯ. ОТЧЕТЫ О ПРОБЛЕМАХ должны включать заключение о критичности (например, влияние на функциональные характеристики, БЕЗОПАСНОСТЬ или ЗАЩИЩЕННОСТЬ), а также другую информацию, которая может помочь в решении проблемы (например, затронутые устройства, затронутое вспомогательное оборудование). [Классы A, B, C]

Примечание — Проблемы могут быть обнаружены до или после выпуска, внутри организации ИЗГОТОВИТЕЛЯ или вне ее.

9.2 Исследование проблемы

ИЗГОТОВИТЕЛЬ должен:

- исследовать проблему и, если возможно, определить причины;
- ОЦЕНИТЬ влияние проблемы на БЕЗОПАСНОСТЬ, используя ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА (раздел 7);
- документировать результаты исследования и оценки;
- создать ЗАПРОС (ЗАПРОСЫ) НА ИЗМЕНЕНИЕ в отношении действий, необходимых для исправления проблемы, или документировать объяснение того, почему никакие действия не предприняты. [Классы A, B, C]

Примечание — Проблема не обязательно должна быть исправлена ИЗГОТОВИТЕЛЕМ, чтобы соответствовать ПРОЦЕССУ решения проблем программного обеспечения, при условии, что проблема не является важной для обеспечения БЕЗОПАСНОСТИ.

9.3 Консультирование заинтересованных сторон

Если применимо, ИЗГОТОВИТЕЛЬ должен консультировать заинтересованные стороны относительно существующей проблемы. [Классы A, B, C]

Примечание — Проблемы могут быть обнаружены до или после выпуска, внутри организации ИЗГОТОВИТЕЛЯ или вне ее. ИЗГОТОВИТЕЛЬ сам определяет заинтересованные стороны в зависимости от ситуации.

9.4 Использование процесса управления изменениями

ИЗГОТОВИТЕЛЬ должен одобрить и осуществить все ЗАПРОСЫ НА ИЗМЕНЕНИЯ, соблюдая требования ПРОЦЕССА управления изменениями (см. пункт 8.2). [Классы A, B, C]

9.5 Поддержание записей

ИЗГОТОВИТЕЛЬ должен поддерживать записи в отношении ОТЧЕТОВ О ПРОБЛЕМАХ и принятых решениях, включая их ВЕРИФИКАЦИЮ.

Если применимо, ИЗГОТОВИТЕЛЬ должен обновлять ФАЙЛ МЕНЕДЖМЕНТА РИСКА. [Классы A, B, C]

9.6 Анализ проблем на предмет выявления тенденций

ИЗГОТОВИТЕЛЬ должен проводить анализ с целью определения тенденции в ОТЧЕТАХ О ПРОБЛЕМАХ. [Классы A, B, C]

9.7 ВЕРИФИКАЦИЯ решения проблем программного обеспечения

- ИЗГОТОВИТЕЛЬ должен верифицировать решения с целью определения:
- была ли проблема решена и был ли завершен ОТЧЕТ О ПРОБЛЕМЕ;
- были ли преодолены неблагоприятные тенденции;
- был ли ЗАПРОС НА ИЗМЕНЕНИЯ реализован в соответствующем ПРОГРАММНОМ ОБЕСПЕЧЕНИИ МЕДИЦИНСКИХ ИЗДЕЛИЙ и ДЕЯТЕЛЬНОСТИ;
- появились ли дополнительные проблемы.

[Классы A, B, C]

9.8 Содержание документации по тестированию

При проведении тестирования, при повторном тестировании или РЕГРЕССИОННОМ ТЕСТИРОВАНИИ ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ и СИСТЕМ, следующих за изменением, ИЗГОТОВИТЕЛЬ должен включить в документацию по испытаниям:

a) результаты тестирования;
b) обнаруженные АНОМАЛИИ;
c) ВЕРСИЮ тестируемого программного обеспечения;
d) соответствующие аппаратные и тестовые конфигурации программного обеспечения;
e) соответствующие инструменты тестирования;
f) дату проведения тестирования;
g) идентификацию лица, проводившего тестирование.

[Классы A, B, C]

Приложение А
(справочное)

Обоснование требований настоящего стандарта

В данном приложении приведено обоснование положений настоящего стандарта.

А.1 Обоснование

Основным требованием настоящего стандарта является выполнение совокупности ПРОЦЕССОВ, которые надлежит применять при разработке и технической поддержке ПРОГРАММНОГО ООБЕСПЕЧЕНИЯ МЕДИЦИНСКОГО ИЗДЕЛИЯ, а также выбор этих ПРОЦЕССОВ, исходя из РИСКА для пациентов и третьих лиц. Это следует из твердого убеждения в том, что для установления безопасности функционирования программного обеспечения одного лишь тестирования недостаточно.

ПРОЦЕССЫ, требуемые настоящим стандартом, можно разделить на две категории:

- ПРОЦЕССЫ для определения РИСКОВ, возникающих от функционирования каждой ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ в программном обеспечении;
- ПРОЦЕССЫ для снижения вероятности отказа программного обеспечения для каждой ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ, выбранные на основе этих определенных РИСКОВ.

Настоящий стандарт требует, чтобы первая категория процессов выполнялась для любого ПО МЕДИЦИНСКИХ ИЗДЕЛИЙ, а вторая категория — только для выбранных ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ.

Следовательно, для соответствия настоящему стандарту должен быть реализован документированный АНАЛИЗ РИСКОВ, идентифицирующий предсказуемые последовательности событий, связанные с наличием программного обеспечения, которые могут привести к ОПАСНОЙ СИТУАЦИИ (см. ISO 14971). ОПАСНЫЕ СИТУАЦИИ, которые могут быть косвенно вызваны программным обеспечением (например, путем предоставления вводящей в заблуждение информации, которая может привести к назначению неверного лечения), должны быть включены в этот АНАЛИЗ РИСКОВ.

Вся ДЕЯТЕЛЬНОСТЬ требующаяся в рамках первой категории ПРОЦЕССОВ, обозначена в нормативном тексте как «[Классы А, В, С]», указывая, что она требуется вне зависимости от класса безопасности программного обеспечения, к которому она относится.

ДЕЯТЕЛЬНОСТЬ требуется для классов А, В и С по следующим причинам:

- ДЕЯТЕЛЬНОСТЬ создает план, имеющий отношение к МЕНЕДЖМЕНТУ РИСКА или классификации программного обеспечения по безопасности;
- ДЕЯТЕЛЬНОСТЬ производит результат, который является входными данными для МЕНЕДЖМЕНТА РИСКА или классификации безопасности программного обеспечения;
- ДЕЯТЕЛЬНОСТЬ является частью МЕНЕДЖМЕНТА РИСКА или классификации безопасности программного обеспечения;
- ДЕЯТЕЛЬНОСТЬ устанавливает систему управления, документации или ведения записей, которая поддерживает МЕНЕДЖМЕНТ РИСКА или классификацию безопасности программного обеспечения;
- ДЕЯТЕЛЬНОСТЬ обычно имеет место, когда классификация связанного с ней ПО неизвестна;
- ДЕЯТЕЛЬНОСТЬ может вызвать изменение, приводящее к изменению класса безопасности связанного с ней программного обеспечения. Это включает обнаружение и анализ проблем связанных с безопасностью после выпуска.

Другие ПРОЦЕССЫ, требующиеся только для ПРОГРАММНЫХ СИСТЕМ или для ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ, классифицируются как классы безопасности В или С. ДЕЯТЕЛЬНОСТЬ, требующаяся как часть этих ПРОЦЕССОВ, указана в нормативном тексте как «[Классы В, С]» или «[Класс С]», указывая, что она требуется в зависимости от класса безопасности программного обеспечения, к которому она относится.

ДЕЯТЕЛЬНОСТЬ требуется выборочно для программного обеспечения классов В и С по следующим причинам:

- ДЕЯТЕЛЬНОСТЬ повышает надежность программного обеспечения, требуя большей детальности или большей точности в дизайне, тестировании или другой ВЕРИФИКАЦИИ;
- ДЕЯТЕЛЬНОСТЬ является управленческой, поддерживающей другую деятельность, требуемую для классов В и С;
- ДЕЯТЕЛЬНОСТЬ поддерживает коррекцию связанных с БЕЗОПАСНОСТЬЮ проблем;
- ДЕЯТЕЛЬНОСТЬ обеспечивает ведение записей по проекту (дизайну), имплементации, ВЕРИФИКАЦИИ и выпуску связанного с БЕЗОПАСНОСТЬЮ программного обеспечения;

ДЕЯТЕЛЬНОСТЬ требуется выборочно для класса С по следующим причинам:

- ДЕЯТЕЛЬНОСТЬ еще больше повышает надежность СИСТЕМЫ, требуя более тщательного, или более точного, или более внимательного отношения к отдельным вопросам проекта (дизайна), тестирования или другой ВЕРИФИКАЦИИ.

Следует отметить, что все ПРОЦЕССЫ и ДЕЯТЕЛЬНОСТЬ, указанные в настоящем стандарте, являются значимыми для обеспечения разработки и технической поддержки высококачественного программного обеспече-

ния. Отсутствие многих из этих ПРОЦЕССОВ и ДЕЯТЕЛЬНОСТИ в качестве требований для ПО класса А не подразумевает, что эти ПРОЦЕССЫ и ДЕЯТЕЛЬНОСТЬ не являются важными или не рекомендуются. Их отсутствие оправдано тем, что БЕЗОПАСНОСТЬ и результативность программного обеспечения, которое не может быть причиной ОПАСНОСТИ, можно обеспечить посредством совокупной ДЕЯТЕЛЬНОСТИ по валидации в рамках проектирования МЕДИЦИНСКОГО ИЗДЕЛИЯ (что выходит за рамки области применения настоящего стандарта), а также посредством простых средств управления жизненным циклом программного обеспечения.

А.2 Краткое изложение требований по классификации

Таблица А.1 показывает, какие классы безопасности программного обеспечения назначены каждому требованию. Это информационная таблица и предоставлена она только для удобства. Нормативный раздел указывает классы безопасности программного обеспечения для каждого требования.

Таблица А.1 — Краткое изложение требований в зависимости от классификации безопасности программного обеспечения

Пункты и подпункты		Класс А	Класс В	Класс С
Раздел 4	Все требования	X	X	X
5.1	5.1.1, 5.1.2, 5.1.3, 5.1.6, 5.1.7, 5.1.8 5.1.9	X	X	X
	5.1.5, 5.1.10, 5.1.11, 5.1.12		X	X
	5.1.4			X
5.2	5.2.1, 5.2.2, 5.2.4, 5.2.5, 5.2.6	X	X	X
	5.2.3		X	X
5.3	5.3.1, 5.3.2, 5.3.3, 5.3.4, 5.3.6		X	X
	5.3.5			X
5.4	5.4.1		X	X
	5.4.2, 5.4.3, 5.4.4			X
Раздел 4	Все требования	X	X	X
5.5	5.5.1	X	X	X
	5.5.2, 5.5.3, 5.5.5		X	X
	5.5.4			X
5.6	Все требования		X	X
5.7	Все требования	X	X	X
5.8	5.8.1, 5.8.2, 5.8.4, 5.8.7, 5.8.8	X	X	X
	5.8.3, 5.8.5, 5.8.6,		X	X
6	Все требования	X	X	X
7.1	Все требования		X	X
7.2	Все требования		X	X
7.3	Все требования		X	X
7.4	7.4.1	X	X	X
	7.4.2, 7.4.3		X	X
Раздел 8	Все требования	X	X	X
Раздел 9	Все требования	X	X	X

Приложение B
(справочное)

Руководство по положениям настоящего стандарта

B.1 Область применения

B.1.1 Цель

Цель настоящего стандарта состоит в том, чтобы обеспечить ПРОЦЕСС разработки, который позволит последовательно создавать высококачественное и безопасное ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МЕДИЦИНСКИХ ИЗДЕЛИЙ. Для достижения этой цели настоящий стандарт устанавливает минимальную ДЕЯТЕЛЬНОСТЬ и ЗАДАЧИ, которые необходимо выполнить для уверенности в том, что разработанное таким образом программное обеспечение (далее — ПО) позволяет создавать высоконадежное и безопасное ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МЕДИЦИНСКОГО ИЗДЕЛИЯ.

Данное приложение содержит рекомендации по применению, которые не дополняют и не изменяют требования настоящего стандарта. Данное приложение может использоваться для лучшего понимания требований настоящего стандарта.

Следует отметить, что в настоящем стандарте ДЕЯТЕЛЬНОСТЬ выполняется в рамках ПРОЦЕССОВ, а ЗАДАЧИ определяются при осуществлении ДЕЯТЕЛЬНОСТИ. Например, ДЕЯТЕЛЬНОСТЬЮ, выполняемой в рамках ПРОЦЕССА разработки ПО, являются планирование разработки ПО, анализ требований к ПО, проектирование АРХИТЕКТУРЫ ПО, разработка детального дизайна ПО, имплементация ПРОГРАММНЫХ БЛОКОВ и их ВЕРИФИКАЦИЯ, интеграция и тестирование интеграции ПО, тестирование ПРОГРАММНОЙ СИСТЕМЫ и выпуск ПО. ЗАДАЧИ являются конкретными требованиями при осуществлении ДЕЯТЕЛЬНОСТИ.

Настоящий стандарт не требует использования определенной МОДЕЛИ ЖИЗНЕННОГО ЦИКЛА РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ. Тем не менее соответствие настоящему стандарту подразумевает наличие зависимости между ПРОЦЕССАМИ, поскольку входные данные одного ПРОЦЕССА являются выходами другого ПРОЦЕССА. Например, классификация безопасности ПО для ПРОГРАММНОЙ СИСТЕМЫ должна быть завершена после того, как ПРОЦЕСС АНАЛИЗА РИСКОВ установит, какой ВРЕД может быть причинен в результате отказа ПРОГРАММНОЙ СИСТЕМЫ.

Из-за указанных логических зависимостей между процессами в настоящем стандарте целесообразней описывать процессы в последовательности, подразумевающей «водопадную» или «сквозную» модель жизненного цикла. Однако можно использовать и другие модели. Некоторые стратегии разработки (модели) определены в стандарте ISO/IEC 12207 [9] и включают (см. также таблицу B. 1):

- водопад. «Сквозная» стратегия, также называемая «водопад», состоит в выполнении ПРОЦЕССА разработки за один раз. Нужно установить потребности потребителя, определить требования, разработать проект (дизайн) СИСТЕМЫ, имплементировать СИСТЕМУ, протестировать, исправить и осуществить поставку;
- пошаговую. Она устанавливает потребности потребителя и определяет требования СИСТЕМЫ. Далее разработка осуществляется в виде последовательной сборки. Первая сборка реализует часть запланированных возможностей, следующая добавляет еще часть возможностей, и так далее, до тех пор, пока СИСТЕМА не будет завершена;
- эволюционную. Она так же развивает СИСТЕМУ в сборке, но в отличие от пошаговой стратегии признавая, что нужды потребителя до конца не изучены и все требования не могут быть определены заранее. В этой стратегии нужды заказчика и системные требования определяются заранее, а затем актуализируются при каждой последующей сборке.

Таблица B.1 — Разработка стратегий (моделей), как это определено в ISO/IEC 12207

Стратегия разработки	С самого начала определяет все требования?	Многократные циклы разработки?	Поставляет временное программное обеспечение?
Водопад (сквозная)	Да	Нет	Нет
Пошаговая (предварительно запланированное улучшение продукции)	Да	Да	Возможно
Эволюционная	Нет	Да	Да

Какой бы жизненный цикл ни был выбран, необходимо поддерживать логические зависимости между выходными данными ПРОЦЕССОВ, такими как спецификации, проектные документы и ПО. Модель жизненного цикла «водопад» достигает этого, откладывая старт ПРОЦЕССА до тех пор, пока входные данные для этого процесса не будут определены и одобрены.

Другие жизненные циклы, особенно эволюционные, позволяют ПРОЦЕССУ вырабатывать выходные данные до того, как будут доступны все входные данные для этого процесса. Например, новая ПРОГРАММНАЯ СОСТАВНАЯ ЧАСТЬ может быть определена, классифицирована, имплементирована и ВЕРИФИЦИРОВАНА до того, как будет закончена целая программная АРХИТЕКТУРА. Такие жизненные циклы несут в себе РИСК того, что изменение или разработка выходных данных одного процесса сделает недействительными выходные данные другого ПРОЦЕССА. Как бы там ни было, все жизненные циклы используют комплексную систему управления конфигурацией, чтобы убедиться, что выходные данные всех ПРОЦЕССОВ доводятся до согласованного состояния и поддерживаются все необходимые зависимости.

Следующие принципы важны вне зависимости от того, какой жизненный цикл разработки ПО используется:

- все выходные данные ПРОЦЕССА должны поддерживаться в согласованном состоянии; каждый раз, когда выходные данные ПРОЦЕССА создаются или меняются, все выходные данные всех связанных с ними ПРОЦЕССОВ должны обновляться, чтобы поддерживать их согласованность друг с другом и поддерживать все зависимости, явные или подразумевающиеся, требуемые настоящим стандартом;
- все выходные данные ПРОЦЕССА должны быть доступны в случае необходимости в качестве входных данных для дальнейшей работы над ПО;
- перед тем, как любое ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МЕДИЦИНСКИХ ИЗДЕЛИЙ будет выпущено, все выходные данные ПРОЦЕССА должны быть приведены в соответствие друг с другом и должны соблюдаться все зависимости между ПРОЦЕССАМИ, явные или подразумевающиеся, требуемые настоящим стандартом.

B.1.2 Область применения

Настоящий стандарт применяется для разработки и технической поддержки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКИХ ИЗДЕЛИЙ, а также для разработки и технической поддержки МЕДИЦИНСКИХ ИЗДЕЛИЙ, которые содержат ПОНП.

Использование настоящего стандарта требует от ИЗГОТОВИТЕЛЯ выполнения МЕНЕДЖМЕНТА РИСКА МЕДИЦИНСКИХ ИЗДЕЛИЙ в соответствии с ИСО 14971. Следовательно, когда АРХИТЕКТУРА СИСТЕМЫ МЕДИЦИНСКОГО ИЗДЕЛИЯ включает приобретенный компонент (это может быть закупленный компонент или компонент неизвестного происхождения), такой как принтер/плоттер, который содержит ПОНП, этот приобретенный компонент становится ответственностью ИЗГОТОВИТЕЛЯ и должен быть включен в МЕНЕДЖМЕНТ РИСКА МЕДИЦИНСКИХ ИЗДЕЛИЙ. Считается, что посредством надлежащего выполнения МЕНЕДЖМЕНТА РИСКА МЕДИЦИНСКИХ ИЗДЕЛИЙ ИЗГОТОВИТЕЛЬ поймет этот компонент и признает, что он содержит ПОНП. ИЗГОТОВИТЕЛЬ, применяющий настоящий стандарт, должен ввести ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА ПО как часть полного ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА МЕДИЦИНСКОГО ИЗДЕЛИЯ.

Техническая поддержка выпущенного ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКИХ ИЗДЕЛИЙ относится к постпроизводственному опыту работы с ПРОГРАММНЫМ ОБЕСПЕЧЕНИЕМ МЕДИЦИНСКОГО ИЗДЕЛИЯ. Техническая поддержка ПО состоит из сочетания всех технических и административных средств, включая действия по наблюдению для обработки отчетов о проблемах, чтобы сохранить или восстановить элемент в состоянии, в котором он может выполнять требуемую функцию, а также запросы на модификацию, связанные с выпущенным ПРОГРАММНЫМ ОБЕСПЕЧЕНИЕМ МЕДИЦИНСКОГО ИЗДЕЛИЯ. Например, это включает исправление проблемы, регламентированную отчетность, повторную валидацию и предупреждающие действия. См. ISO/IEC 14764 [10].

B.2 Нормативные ссылки

ISO/IEC 90003 [11] предоставляет руководство для применения систем менеджмента качества к разработке ПО. Использование этого руководства не требуется настоящим стандартом, но рекомендуется.

B.3 Термины и определения

Там, где это возможно, терминам даны определения из международных стандартов.

Для описания декомпозиции ПРОГРАММНОЙ СИСТЕМЫ (высший уровень) настоящий стандарт использует три термина. ПРОГРАММНАЯ СИСТЕМА, которая впоследствии становится программным обеспечением МЕДИЦИНСКОГО ИЗДЕЛИЯ, может быть подсистемой МЕДИЦИНСКОГО ИЗДЕЛИЯ (см. IEC 60601-1-4 [2]) или сама по себе являться МЕДИЦИНСКИМ ИЗДЕЛИЕМ, которое затем становится программным МЕДИЦИНСКИМ ИЗДЕЛИЕМ. Самым нижним уровнем, ниже которого дальнейшая декомпозиция для целей тестирования или менеджмента конфигурации ПО не проводится, является ПРОГРАММНЫЙ БЛОК. Все уровни композиции, включая верхний и нижний уровни, могут быть названы ПРОГРАММНЫМИ СОСТАВНЫМИ ЧАСТЯМИ. Таким образом, ПРОГРАММНАЯ СИСТЕМА состоит из одного или нескольких ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ, а каждая ПРОГРАММНАЯ СОСТАВНАЯ ЧАСТЬ — из ПРОГРАММНЫХ БЛОКОВ или разделяемых ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ. Ответственность за определение и степень детализации ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ и ПРОГРАММНЫХ БЛОКОВ возлагается на ИЗГОТОВИТЕЛЯ. Отсутствие четкого определения этих терминов позволяет применять их ко многим разным методам разработки и типам ПО, используемым в МЕДИЦИНСКИХ ИЗДЕЛИЯХ.

B.4 Общие требования

Не существует метода, чтобы обеспечить 100 %-ную БЕЗОПАСНОСТЬ для любого вида ПО.

Есть три главных принципа, которые способствуют обеспечению БЕЗОПАСНОСТИ ПО МЕДИЦИНСКИХ ИЗДЕЛИЙ:

- МЕНЕДЖМЕНТ РИСКА;
- менеджмент качества;
- разработка ПО.

Для разработки и технической поддержки безопасного ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКОГО ИЗДЕЛИЯ необходимо установить МЕНЕДЖМЕНТ РИСКА как неотъемлемую часть системы менеджмента качества, как общий каркас для приложения соответствующих методов и техник разработки ПО. Комбинация этих трех принципов позволяет ИЗГОТОВИТЕЛЮ МЕДИЦИНСКИХ ИЗДЕЛИЙ следовать последовательно повторяемому ПРОЦЕССУ принятия решений, способствующему БЕЗОПАСНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКИХ ИЗДЕЛИЙ.

B.4.1 Система менеджмента качества

Управляемая и результативная совокупность ПРОЦЕССОВ разработки ПО включает организационные ПРОЦЕССЫ, такие как менеджмент, инфраструктура, улучшение и обучение. Для исключения дублирования и с целью фокусировки внимания пользователя настоящего стандарта на разработке ПО данные ПРОЦЕССЫ не рассматриваются.

Эти ПРОЦЕССЫ устанавливаются системой менеджмента качества. ISO 13485 [8] специально предназначен для применения концепции менеджмента качества к МЕДИЦИНСКИМ ИЗДЕЛИЯМ. Соответствие требованиям системы менеджмента качества ISO 13485 не означает автоматического соответствия национальным или региональным регулирующим требованиями. Ответственность за определение и установление соответствия применимым регулирующим требованиям лежит на ИЗГОТОВИТЕЛЕ.

B.4.2 МЕНЕДЖМЕНТ РИСКА

Разработка ПО является частью ДЕЯТЕЛЬНОСТИ по МЕНЕДЖМЕНТУ РИСКА и обеспечивает рассмотрение всех обоснованно прогнозируемых РИСКОВ, связанных с ПРОГРАММНЫМ ОБЕСПЕЧЕНИЕМ МЕДИЦИНСКОГО ИЗДЕЛИЯ.

Вместо того, чтобы пытаться определить подходящий ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА, в настоящем стандарте требуется, чтобы ИЗГОТОВИТЕЛЬ применил установленный в ISO 14971 ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА, который непосредственно относится к МЕНЕДЖМЕНТУ РИСКА для МЕДИЦИНСКИХ ИЗДЕЛИЙ. Конкретные виды ДЕЯТЕЛЬНОСТИ по МЕНЕДЖМЕНТУ РИСКА ПО, возникающего в результате ОПАСНЫХ СИТУАЦИЙ, причиной которых является ПО, указана во вспомогательном ПРОЦЕССЕ, описанном в разделе 7.

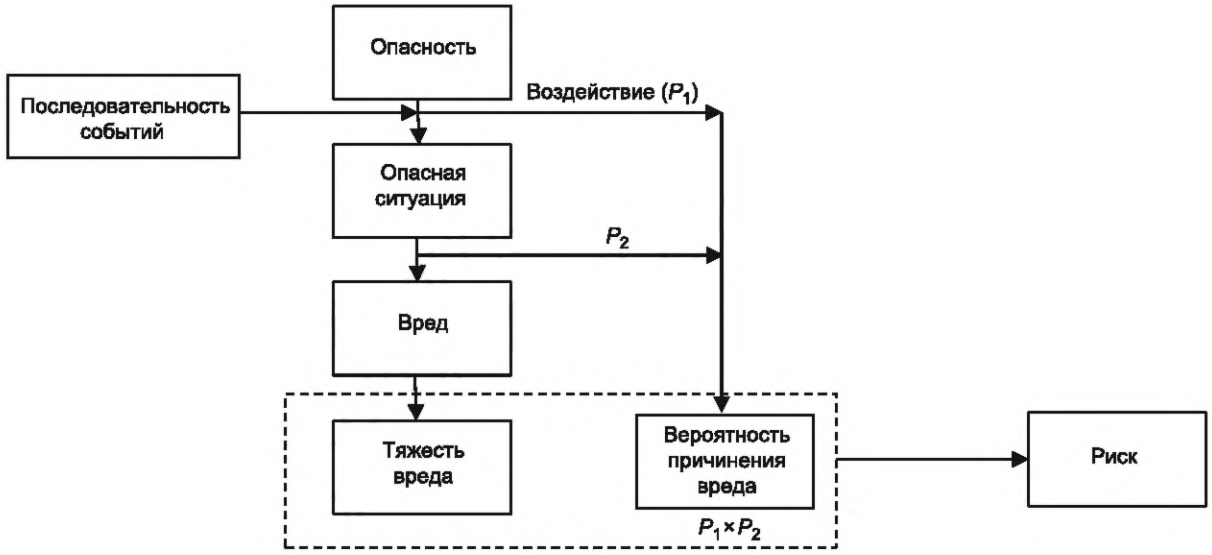
B.4.3 Классификация безопасности ПО

РИСК, связанный с ПО (как с неотъемлемой частью МЕДИЦИНСКОГО ИЗДЕЛИЯ, или как с прикладываемым к МЕДИЦИНСКОМУ ИЗДЕЛИЮ, или как с самостоятельным МЕДИЦИНСКИМ ИЗДЕЛИЕМ), используется в качестве входных данных для схемы классификации безопасности ПО, которая затем устанавливает ПРОЦЕССЫ, использующиеся при разработке и технической поддержке ПО.

РИСК рассматривается как комбинация тяжести ВРЕДА и вероятности его возникновения. Не существует общепризнанного метода количественного определения вероятности возникновения отказа программного обеспечения. Если ПО задействовано в последовательности или комбинации событий, приводящих к ОПАСНОЙ СИТУАЦИИ, то вероятность возникновения отказа программного обеспечения не может быть учтена при определении РИСКА от ОПАСНОЙ СИТУАЦИИ. В таких ситуациях целесообразно рассмотреть вероятность наихудшего случая и установить вероятность возникновения отказа ПО равной 1. Если есть возможность определить вероятность для остальных событий в последовательности (что возможно, если они не являются программными), то эта вероятность может быть использована для определения вероятности возникновения ОПАСНОЙ СИТУАЦИИ (P1 на рисунке B. 2).

В некоторых случаях невозможно определить вероятность остальных событий в последовательности и РИСК следует ОЦЕНИВАТЬ только на основе характера ВРЕДА (вероятность возникновения ОПАСНОЙ СИТУАЦИИ должна быть установлена равной 1). ОПРЕДЕЛЕНИЕ РИСКА в этих случаях должно быть сосредоточено на ТЯЖЕСТИ ВРЕДА, причиненного в результате ОПАСНОЙ СИТУАЦИИ. Субъективные оценки вероятности могут быть сделаны на основе клинических знаний, чтобы отличить очевидные для медицинского специалиста отказы от тех, которые не будут обнаружены и с большей вероятностью приведут к причинению ВРЕДА.

Определение вероятности возникновения ОПАСНОЙ СИТУАЦИИ, приводящей к ПРИЧИНЕНИЮ ВРЕДА (P2 на рисунке B. 2), как правило, требует клинических знаний для проведения разграничения между ОПАСНЫМИ СИТУАЦИЯМИ, в которых клиническая практика вероятней всего предотвратит ВРЕД, и ОПАСНЫМИ СИТУАЦИЯМИ, которые с большей вероятностью приведут к причинению ВРЕДА.



Примечание

P_1 — вероятность возникновения опасной ситуации;

P_2 — вероятность возникновения опасной ситуации, приводящей к причинению вреда.

Рисунок В.2 — Наглядное представление взаимосвязи ОПАСНОСТИ, последовательности событий, ОПАСНОЙ СИТУАЦИИ и ВРЕДА — заимствовано из ISO 14971:2007, приложение Е

Если ПРОГРАМНАЯ СИСТЕМА подразделяется на ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ, то каждая ПРОГРАММНАЯ СОСТАВНАЯ ЧАСТЬ может иметь свой собственный класс безопасности ПО. Определить РИСК, связанный с отказом ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ, можно только:

- если АРХИТЕКТУРА СИСТЕМЫ и АРХИТЕКТУРА ПО определяют роль ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ с точки зрения ее назначения и взаимодействия с другими программными и аппаратными элементами;
- если изменения в СИСТЕМЕ находятся под управлением;
- после выполнения АНАЛИЗА РИСКА для данной АРХИТЕКТУРЫ и установления мер по УПРАВЛЕНИЮ РИСКОМ.

Настоящий стандарт требует минимального количества ДЕЯТЕЛЬНОСТИ, направленной на выполнение вышеуказанных условий для всех классов ПО.

Завершение ДЕЯТЕЛЬНОСТИ по построению АРХИТЕКТУРЫ ПО является первым этапом разработки, когда определен полный набор ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ и в ходе ДЕЯТЕЛЬНОСТИ по МЕНЕДЖМЕНТУ РИСКА установлено, как ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ связаны с БЕЗОПАСНОСТЬЮ. Следовательно, это самый первый этап в разработке, на котором ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ могут быть окончательно классифицированы согласно их роли в обеспечении БЕЗОПАСНОСТИ.

Данный этап соответствует точке, с которой в ISO 14971 начинается УПРАВЛЕНИЕ РИСКОМ.

До этого этапа ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА идентифицирует меры по УПРАВЛЕНИЮ РИСКОМ в отношении АРХИТЕКТУРЫ, например добавление защитных подсистем или уменьшение возможностей причинения ВРЕДА отказами ПО. После этого этапа ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА использует ПРОЦЕССЫ, направленные на снижение вероятности отказа ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ. Другими словами, классификация ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ определяет меры по УПРАВЛЕНИЮ РИСКОМ, основанные на ПРОЦЕССАХ, которые должны к ней применяться.

ИЗГОТОВИТЕЛИ могут счесть полезным классифицировать ПО до данного этапа, например, чтобы сосредоточить внимание на нуждающихся в исследовании областях, но такая классификация должна рассматриваться как предварительная и не использоваться для обоснования пропуска ПРОЦЕССОВ.

Схема классификации безопасности ПО не предназначена для согласования с классификацией РИСКОВ согласно ISO 14971. Если в ISO 14971 классификация РИСКОВ осуществляется в соответствии с их тяжестью и вероятностью возникновения, то схема классификации безопасности ПО классифицирует ПРОГРАММНЫЕ СИСТЕМЫ и ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ в соответствии с ПРОЦЕССАМИ, которые будут применяться при их разработке и технической поддержке.

По мере развития проекта могут стать очевидными новые РИСКИ. Следовательно, МЕНЕДЖМЕНТ РИСКА должен применяться как неотъемлемая часть ПРОЦЕССА разработки. Это позволяет разработать проект АРХИТЕКТУРЫ, устанавливающий полный набор ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ, включая необходимые для

правильного функционирования с целью обеспечения безопасности, а также предотвращающие причинение ВРЕДА из-за отказов.

АРХИТЕКТУРА программного обеспечения должна способствовать изоляции программных элементов (составных частей), которые требуются для безопасной работы, и описывать методы, используемые для обеспечения результативного разделения этих ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ. Разделение не ограничивается физической изоляцией (раздел процессора или памяти) и содержит любой механизм, который предотвращает негативное влияние одной ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ на другую. Достаточность разделения определяется на основе связанных с этим РИСКОВ и обоснования, которое требуется задокументировать.

Как установлено в В.3, настоящий стандарт использует три термина для описания декомпозиции ПРОГРАММНОЙ СИСТЕМЫ (высший уровень).

Рисунок В.1 иллюстрирует возможное разделение ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ в рамках ПРОГРАММНОЙ СИСТЕМЫ и то, как классы безопасности ПО будут применяться к группе ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ в процессе декомпозиции.

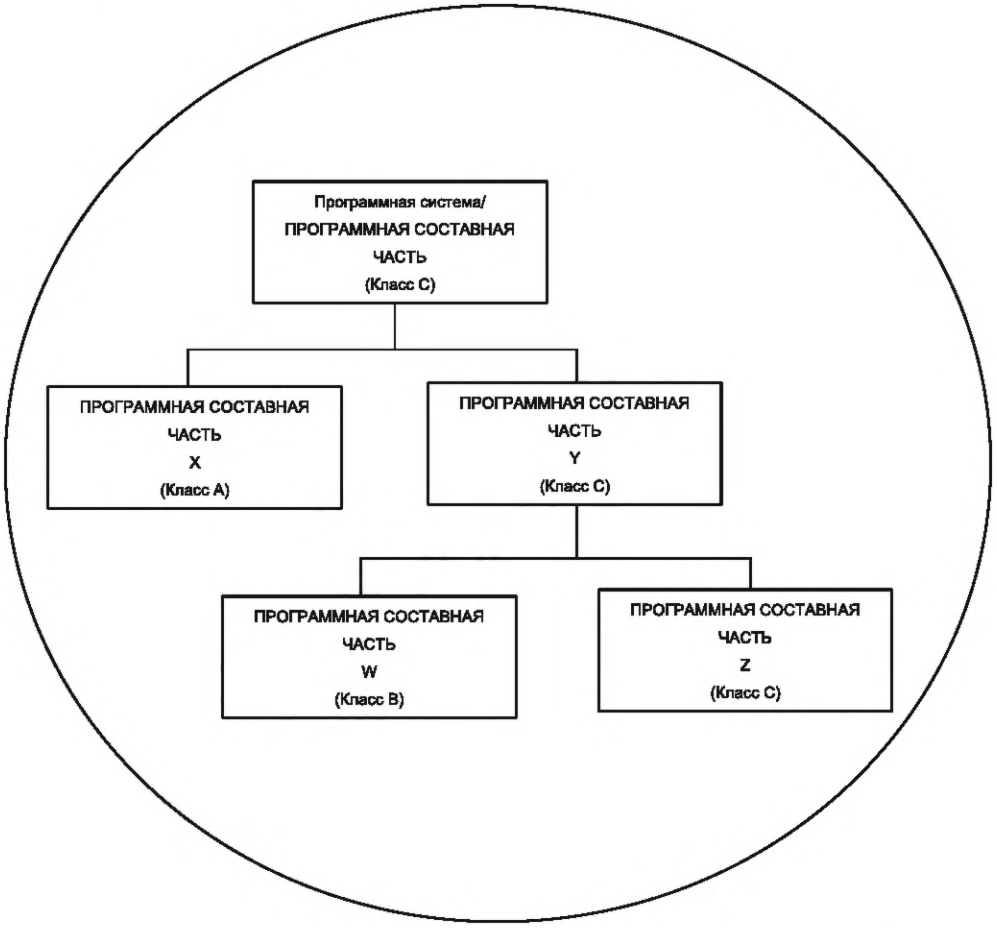


Рисунок В.1 — Пример разделения ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ

В приведенном примере ИЗГОТОВИТЕЛЬ знает благодаря типу разрабатываемого ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКОГО ИЗДЕЛИЯ, что ПРОГРАММНАЯ СИСТЕМА по предварительной классификации безопасности ПО относится к классу С. При проектировании АРХИТЕКТУРЫ ПО ИЗГОТОВИТЕЛЬ решает разделить СИСТЕМУ на три ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ — X, W и Z. ИЗГОТОВИТЕЛЬ может разделить весь вклад ПРОГРАММНОЙ СИСТЕМЫ в возникновение ОПАСНЫХ СИТУАЦИЙ на приводящие к возможной смерти или СЕРЬЕЗНОЙ ТРАВМЕ, в ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ Z, и вклад всей оставшейся ПРОГРАММНОЙ СИСТЕМЫ в ОПАСНЫЕ СИТУАЦИИ, не приводящие к возможной СЕРЬЕЗНОЙ ТРАВМЕ, в ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ W. ПРОГРАММНАЯ СОСТАВНАЯ ЧАСТЬ W классифицируется как класс безопасности В, а ПРОГРАММНАЯ СОСТАВНАЯ ЧАСТЬ Z относится к классу безопасности С. ПРОГРАММНАЯ СОСТАВНАЯ ЧАСТЬ Y, следовательно, должна быть отнесена к классу С (см. 4.3 d). В соответствии с этим требованием ПРОГРАММНАЯ СИСТЕМА также получает класс безопасности С. ПРОГРАММНАЯ СОСТАВНАЯ ЧАСТЬ X относится к классу безопасности А. ИЗГОТОВИТЕЛЬ может документировать обоснование разделения ПРОГРАММНЫХ СОСТАВНЫХ

ЧАСТЕЙ X и Y, а также ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ W и Z, чтобы обеспечить целостность разделения. Если разделение между ПРОГРАММНЫМИ СОСТАВНЫМИ ЧАСТЯМИ X и Y невозможно, то ПРОГРАММНАЯ СОСТАВНАЯ ЧАСТЬ X должна быть отнесена к классу безопасности C.

B.4.4 Устаревшее/наследуемое ПО

Подраздел 4.4 устанавливает процесс применения настоящего стандарта к УСТАРЕВШЕМУ ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ. В некоторых регионах может потребоваться, чтобы ИЗГОТОВИТЕЛЬ продемонстрировал соответствие стандарту для получения одобрения регулирующих органов ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКОГО ИЗДЕЛИЯ, даже если это программное обеспечение было разработано до появления текущей версии стандарта (УСТАРЕВШЕЕ ПО). В этом случае требования раздела 4.4 предоставляют ИЗГОТОВИТЕЛЮ способ продемонстрировать соответствие УСТАРЕВШЕГО ПО настоящему стандарту.

ИЗГОТОВИТЕЛЬ может определить, что ретроспективная документация уже завершенного жизненного цикла разработки, выполняемая как изолированная деятельность, не приводит к снижению РИСКА, связанного с использованием продукта. Этот процесс приводит к идентификации подмножества видов ДЕЯТЕЛЬНОСТИ, определенных в настоящем стандарте, что действительно приводит к снижению РИСКА. Некоторые дополнительные цели, подразумеваемые в этом процессе, заключаются в следующем:

- необходимая ДЕЯТЕЛЬНОСТЬ и итоговая документация должны основываться на существующей документации и использовать ее, где это возможно,
- ИЗГОТОВИТЕЛЬ должен использовать ресурсы для максимально результативного снижения РИСКА.

В дополнение к плану, определяющему подмножество видов ДЕЯТЕЛЬНОСТИ, которые необходимо выполнить, результатом процесса являются объективные свидетельства, подтверждающие безопасное дальнейшее использование УСТАРЕВШЕГО ПО, а также краткое обоснование этого вывода.

РИСКИ, связанные с планируемым продолжением использования УСТАРЕВШЕГО ПО, зависят от контекста, в котором оно будет использоваться для создания ПРОГРАММНОЙ СИСТЕМЫ. ИЗГОТОВИТЕЛЬ будет документировать все выявленные ОПАСНОСТИ МЕДИЦИНСКОГО ИЗДЕЛИЯ, связанные с УСТАРЕВШИМ ПО.

В 4.4 требуется всесторонняя оценка имеющихся постпроизводственных данных, полученных для УСТАРЕВШЕГО ПО за время его производства и применения. Типичные источники данных на стадии постпроизводства включают:

- неблагоприятные события, связанные с изделием,
- отзывы, полученные от пользователей изделия, и
- АНОМАЛИИ, обнаруженные ИЗГОТОВИТЕЛЕМ.

Хотя не существует единого мнения в отношении метода перспективного количественного определения вероятности возникновения отказа программного обеспечения, для УСТАРЕВШЕГО ПО может быть доступна уместная информация по постпроизводству. Если в таких случаях возможно количественно определить вероятность событий в последовательности, то количественное значение может быть использовано для выражения вероятности возникновения всей последовательности событий. Если данная количественная оценка невозможна, целесообразно учитывать вероятность наихудшего случая и принять вероятность возникновения сбоя программного обеспечения равной 1.

Определение ИЗГОТОВИТЕЛЕМ того, как УСТАРЕВШЕЕ ПО будет использоваться в общей АРХИТЕКТУРЕ СИСТЕМЫ МЕДИЦИНСКОГО ИЗДЕЛИЯ, является вкладом в оценку РИСКА. РИСКИ, которые необходимо учитывать, соответственно различаются.

- Когда УСТАРЕВШЕЕ ПО было безопасно и надежно использовано и ИЗГОТОВИТЕЛЬ желает продолжить его применение, то обоснование дальнейшего использования должно базироваться в первую очередь на оценке РИСКА, проведенной на основе постпроизводственных записей.
- При повторном использовании УСТАРЕВШЕГО ПО для создания новой ПРОГРАММНОЙ СИСТЕМЫ предполагаемое использование УСТАРЕВШЕГО ПО может отличаться от его первоначального предназначения. В этом случае оценка РИСКА должна учитывать измененный набор ОПАСНЫХ СИТУАЦИЙ, которые могут возникнуть из-за отказов УСТАРЕВШЕГО ПО.
- Повторно используемое УСТАРЕВШЕЕ ПО может использоваться по аналогичному назначению, но без интеграции в новую ПРОГРАММНУЮ СИСТЕМУ. В этом случае оценка РИСКА должна учитывать изменение мер по УПРАВЛЕНИЮ РИСКОМ архитектуры в соответствии с 5.3.

Когда УСТАРЕВШЕЕ ПО будет изменено и использовано в новой ПРОГРАММНОЙ СИСТЕМЕ, ИЗГОТОВИТЕЛЮ следует рассмотреть вопрос о том, как существующие записи о безопасной и надежной работе могут быть признаны недействительными в результате изменений.

Изменения в УСТАРЕВШЕМ ПО должны выполняться в соответствии с разделами 4—9 настоящего стандарта, включая оценку воздействия мер по УПРАВЛЕНИЮ РИСКАМИ в соответствии с 7.4. В случае УСТАРЕВШЕГО ПО существующие меры по УПРАВЛЕНИЮ РИСКАМИ могут быть не полностью документированы и особое внимание следует уделить ОЦЕНКЕ потенциального воздействия изменений, используя имеющиеся документированные проектные записи, а также опыт лиц, обладающих знаниями о системе.

Согласно 4.4 ИЗГОТОВИТЕЛЬ проводит анализ разрывов (пробелов), чтобы определить доступную документацию, включая объективные свидетельства выполненных ЗАДАЧ, созданную во время разработки УСТАРЕВ-

ШЕГО ПО, и сравнить ее с 5.2, 5.3, 5.7 и разделом 7. Типичные шаги для выполнения данного анализа разрывов (пробелов) включают:

а) идентификацию УСТАРЕВШЕГО ПО, включая ВЕРСИЮ, редакцию и любые другие средства, необходимые для четкой идентификации;

б) ОЦЕНКУ существующих ПОСТАВЛЯЕМЫХ РЕЗУЛЬТАТОВ, соответствующих результатам, требуемым 5.2, 5.3, 5.7 и разделом 7;

с) ОЦЕНКУ имеющихся объективных свидетельств, документирование ранее применявшейся модели жизненного цикла разработки программного обеспечения (при необходимости);

d) ОЦЕНКУ адекватности существующей документации по МЕНЕДЖМЕНТУ РИСКА с учетом ISO 14971.

Принимая во внимание проведенный анализ разрывов (пробелов), ИЗГОТОВИТЕЛЬ оценит потенциальное снижение РИСКА в результате создания недостающих ПОСТАВЛЯЕМЫХ РЕЗУЛЬТАТОВ и связанной с ними ДЕЯТЕЛЬНОСТИ, а также разработает план выполнения ДЕЯТЕЛЬНОСТИ и создания недостающих пока ПОСТАВЛЯЕМЫХ РЕЗУЛЬТАТОВ для закрытия этих разрывов.

Снижение РИСКА должно уравновешивать преимущества применения процесса разработки программного обеспечения в соответствии с разделом 5 с возможностью того, что модификация УСТАРЕВШЕГО ПО без полного знания истории его разработки может привести к появлению новых дефектов, которые увеличивают риск. Некоторые элементы раздела 5 могут быть оценены как имеющие незначительное влияние или полностью не влияющие на снижение РИСКА, когда это делается постфактум. Например, детальное проектирование и верификация модуля снижают РИСК в первую очередь в процессе разработки нового программного обеспечения или рефакторинга существующего программного обеспечения. Если эти цели не запланированы, изолированное выполнение ДЕЯТЕЛЬНОСТИ может привести к созданию документации, но не приведет к снижению РИСКА.

Как минимум, план устранения разрывов (пробелов) касается отсутствующих записей тестирования ПРОГРАММНЫХ СИСТЕМ. Если они отсутствуют или не подходят для обоснования продолжения использования УСТАРЕВШЕГО ПО, план устранения разрывов должен включать создание требований к ПРОГРАММНОЙ СИСТЕМЕ на функциональном уровне в соответствии с 5.2 и тестирование в соответствии с 5.7.

Документированное обоснование дальнейшего использования УСТАРЕВШЕГО ПО основывается на имеющихся объективных свидетельствах и результатах анализа, полученных в ходе оценки РИСКА и разработки плана устранения разрывов, соответствующих контексту повторного использования УСТАРЕВШЕГО ПО.

Обоснование дает положительный аргумент в пользу безопасной и надежной работы УСТАРЕВШЕГО ПО в контексте планируемого повторного использования, принимая во внимание как записи о постпроизводстве, доступные для УСТАРЕВШЕГО ПО, так и МЕРЫ ПО УПРАВЛЕНИЮ РИСКОМ, связанные с заполнением пробелов в процессе.

После повторного использования УСТАРЕВШЕГО ПО в соответствии с 4.4 те части УСТАРЕВШЕГО ПО, для которых сохраняются разрывы (пробелы) в ПОСТАВЛЯЕМЫХ РЕЗУЛЬТАТАХ, продолжают оставаться УСТАРЕВШИМ ПО и могут быть рассмотрены для дальнейшего повторного использования в соответствии с 4.4. Когда разрывы (пробелы) в результатах устраняются путем изменения УСТАРЕВШЕГО ПО, изменения должны выполняться в соответствии с разделами 4—9 настоящего стандарта.

B.5 ПРОЦЕСС разработки ПО

B.5.1 Планирование разработки ПО

Целью данной деятельности является планирование ЗАДАЧ разработки ПО для уменьшения РИСКОВ, вызываемых программным обеспечением, сообщение задач и целей участникам группы разработки, а также обеспечение выполнения требований к качеству СИСТЕМЫ для ПО МЕДИЦИНСКИХ ИЗДЕЛИЙ.

ДЕЯТЕЛЬНОСТЬ по планированию разработки ПО может документировать ЗАДАЧИ в едином плане или в различных планах. Некоторые ИЗГОТОВИТЕЛИ могут устанавливать политики и процедуры, которые применяются к разработке всего ПО для своих МЕДИЦИНСКИХ ИЗДЕЛИЙ. В этом случае план может просто ссылаться на существующие политики и процедуры. Некоторые ИЗГОТОВИТЕЛИ могут подготовить план или набор планов для разработки каждого ПО МЕДИЦИНСКИХ ИЗДЕЛИЙ, которые влекут за собой детально установленные виды ДЕЯТЕЛЬНОСТИ и ссылаются на общие процедуры. Другая возможность состоит в том, что план или набор планов приспособлен для разработки каждого ПО МЕДИЦИНСКИХ ИЗДЕЛИЙ. Планирование следует определять на уровне детализации, необходимой для осуществления ПРОЦЕССА разработки, и он должен быть пропорционален РИСКУ. Например, СИСТЕМЫ или элементы с более высокой степенью РИСКА должны подчиняться ПРОЦЕССУ разработки с более строгими требованиями, а ЗАДАЧИ следует излагать более детально.

Планирование является итеративной ДЕЯТЕЛЬНОСТЬЮ, которую следует пересматривать и обновлять по мере развития разработки. План может развиваться, чтобы включать большую и лучшую информацию, по мере того как больше узнают о СИСТЕМЕ и уровне усилий, необходимых для развития СИСТЕМЫ. Например, начальная классификация безопасности ПО СИСТЕМЫ может измениться в результате осуществления ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА и развития АРХИТЕКТУРЫ программного обеспечения. В некоторых случаях может быть принято решение о включении ПОНП в СИСТЕМУ. Важно, чтобы планы обновлялись с целью отразить в них текущие знания о СИСТЕМЕ и уровне усилий, необходимом для СИСТЕМЫ или ее элементов, чтобы обеспечить надлежащее управление ПРОЦЕССОМ разработки.

B.5.2 Анализ требований к ПО

Данная деятельность требует от ИЗГОТОВИТЕЛЯ установить и верифицировать требования к программному обеспечению для ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ. Установление верифицируемых требований крайне важно для определения того, что должно быть создано, для определения, что ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ функционирует должным образом, а также для демонстрации, что завершенное ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ готово к использованию. С целью демонстрации имплементации требований согласно замыслу, каждое требование должно быть установлено таким образом, чтобы можно было установить объективные критерии для проверки правильной имплементации. Если ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА изделия предъявляет требования к ПО для управления выявленными РИСКАМИ, эти требования должны быть идентифицированы в требованиях к программному обеспечению таким образом, чтобы сделать возможным прослеживание мер по УПРАВЛЕНИЮ РИСКОМ до требований к программному обеспечению. Все требования к программному обеспечению следует определять таким образом, чтобы сделать возможной демонстрацию ПРОСЛЕЖИВАЕМОСТИ между требованием и тестированием ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СИСТЕМЫ. Если регулирующие требования некоторых стран требуют соответствия специальным нормам или международным стандартам, данное требование должно быть документировано в требованиях к программному обеспечению. Поскольку требования к программному обеспечению устанавливают, что должно быть имплементировано в программное обеспечение, оценка требований требуется до завершения ДЕЯТЕЛЬНОСТИ по анализу требований.

Областью частых недоразумений является различие между потребностями потребителя, входными данными проектирования, требованиями к программному обеспечению, функциональными спецификациями программного обеспечения и спецификациями проекта (дизайна) программного обеспечения. Входные данные проектирования являются преобразованием потребностей потребителя в официально документированные требования к МЕДИЦИНСКОМУ ИЗДЕЛИЮ. Требования к программному обеспечению — это официально документированные спецификации того, что программное обеспечение отвечает потребностям потребителя и входным данным проектирования. Функциональные спецификации программного обеспечения часто включены в требования к программному обеспечению и определяют в деталях, что программное обеспечение делает, чтобы соответствовать этим требованиям, даже если много других альтернативных вариантов может так же соответствовать этим требованиям. Спецификации проекта программного обеспечения определяют, как ПО будет проектироваться и раскладываться на составные части, чтобы имплементировать эти требования и функциональные спецификации.

Традиционно требования к программному обеспечению, функциональные спецификации и спецификации проекта оформляются как набор из одного и более документов. В настоящее время возможно оформление этой информации как элементы данных внутри общей базы данных. Каждый элемент может иметь один или более признаков, которые определяют его цель и его соединение с другими элементами в базе данных. Этот подход допускает представление и печать различных видов информации, которая лучше всего подходит для каждой группы предполагаемых пользователей (например, продавцов, ИЗГОТОВИТЕЛЕЙ, тестировщиков, аудиторов) и поддерживает ПРОСЛЕЖИВАЕМОСТЬ, чтобы демонстрировать соответствие имплементации и степени, до которой тестовые задания проверяют требования. Инструменты, поддерживающие этот подход, могут быть такими же простыми, как гипертекстовый документ, использующий гиперссылки HTML, или столь же сложными, как CASE (computer aided software engineering — разработка компьютерного программного обеспечения) и инструменты анализа требований.

ПРОЦЕСС определения требований к СИСТЕМЕ лежит вне области применения настоящего стандарта. Однако решение об имплементации функционала МЕДИЦИНСКИХ ИЗДЕЛИЙ с программным обеспечением обычно осуществляется по время проектирования СИСТЕМЫ. Некоторые или все требования к СИСТЕМЕ выделяются с целью имплементации в программное обеспечение. ДЕЯТЕЛЬНОСТЬ по анализу требований к программному обеспечению заключается в анализе требований предъявляемых к программному обеспечению ПРОЦЕССОМ определения требований к СИСТЕМЕ, и в получении полного набора требований к программному обеспечению, отражающих выделенные требования.

Чтобы обеспечить целостность СИСТЕМЫ, ИЗГОТОВИТЕЛЬ должен предусмотреть механизм согласования внесения изменений и уточнения СИСТЕМНЫХ требований для исправления непрактичности, несоответствий или двусмысленностей либо в исходных СИСТЕМНЫХ требованиях, либо в требованиях к программному обеспечению.

ПРОЦЕСС сбора и анализа СИСТЕМНЫХ и программных требований может быть итеративным.

Настоящий стандарт не предполагает жесткого разделения ПРОЦЕССОВ на два уровня. На практике АРХИТЕКТУРА СИСТЕМЫ и АРХИТЕКТУРА программного обеспечения часто описываются одновременно, а требования к СИСТЕМЕ и программному обеспечению впоследствии документируются в многоуровневой форме.

B.5.3 АРХИТЕКТУРА программного обеспечения

Эта деятельность требует, чтобы ИЗГОТОВИТЕЛЬ определил главные структурные компоненты программного обеспечения и определил их основную зону ответственности, а также их внешне видимые свойства и взаимосвязь между ними. Если функционирование компонента может влиять на другие компоненты, то оно должно быть описано в АРХИТЕКТУРЕ программного обеспечения. Это описание особенно важно для аспектов функционирования, которые могут повлиять на компоненты МЕДИЦИНСКОГО ИЗДЕЛИЯ, находящиеся вне программного обеспечения (см. 5.3.5 и B.4.3). АРХИТЕКТУРНЫЕ решения чрезвычайно важны для осуществления мер по

УПРАВЛЕНИЮ РИСКАМИ. Без понимания (и документирования) аспектов функционирования компонента, которые могут повлиять на другие компоненты, почти невозможно доказать, что СИСТЕМА безопасна. АРХИТЕКТУРА программного обеспечения необходима для обеспечения правильной имплементации требований к ПО. АРХИТЕКТУРА программного обеспечения не считается завершенной, если все требования к ПО не могут быть реализованы определенными ПРОГРАММНЫМИ СОСТАВНЫМИ ЧАСТЯМИ. Поскольку дизайн и имплементация программного обеспечения зависят от АРХИТЕКТУРЫ, АРХИТЕКТУРА ВЕРИФИЦИРУЕТСЯ для завершения этой ДЕЯТЕЛЬНОСТИ. Как правило, ВЕРИФИКАЦИЯ АРХИТЕКТУРЫ выполняется путем технического ОЦЕНИВАНИЯ.

Классификация безопасности программного обеспечения в отношении ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ в процессе ДЕЯТЕЛЬНОСТИ по разработке АРХИТЕКТУРЫ программного обеспечения создает основание для последующего выбора программных ПРОЦЕССОВ. Записи о классификации находятся под управлением изменениями в составе ФАЙЛА МЕНЕДЖМЕНТА РИСКА.

Множество последующих событий может сделать классификацию недействительной. Они включают, например:

- изменения спецификации СИСТЕМЫ, программной спецификации или АРХИТЕКТУРЫ;
- обнаружение ошибок в АНАЛИЗЕ РИСКОВ, особенно непредвиденных ОПАСНОСТЕЙ; и
- обнаружение неосуществимости требования, особенно меры по УПРАВЛЕНИЮ РИСКОМ.

Поэтому во время всех видов ДЕЯТЕЛЬНОСТИ, следующих за разработкой АРХИТЕКТУРЫ программного обеспечения, классификация ПРОГРАММНЫХ СИСТЕМ и ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ должна ПЕРЕОЦЕНИВАТЬСЯ и, если нужно, пересматриваться. Это вызывает доработку с применением дополнительных ПРОЦЕССОВ к отдельной ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ в результате обновления до более высокого класса. ПРОЦЕСС менеджмента конфигурации программного обеспечения (раздел 8) используется для обеспечения уверенности в том, что все необходимые доработки были идентифицированы и завершены.

B.5.4 Детальный дизайн программного обеспечения

Данная ДЕЯТЕЛЬНОСТЬ требует от ИЗГОТОВИТЕЛЯ усовершенствования ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ и интерфейсов, определенных в АРХИТЕКТУРЕ, чтобы создать ПРОГРАММНЫЕ БЛОКИ и их интерфейсы. Хотя ПРОГРАММНЫЕ БЛОКИ часто считаются единичными функциями или модулями, эта точка зрения не всегда является приемлемой. Настоящий стандарт определяет ПРОГРАММНЫЙ БЛОК как ПРОГРАММНУЮ СОСТАВНУЮ ЧАСТЬ, не делимую на более мелкие элементы. ПРОГРАММНЫЕ БЛОКИ могут проверяться отдельно. ИЗГОТОВИТЕЛЮ следует определить уровень детализации ПРОГРАММНОГО БЛОКА. Детальный дизайн определяет алгоритмы представления данных и взаимодействия между ПРОГРАММНЫМИ БЛОКАМИ и структурами данных. Детальный дизайн также должен касаться формирования ПРОГРАММНОГО ПРОДУКТА. Необходимо определить конструкцию ПРОГРАММНЫХ БЛОКОВ и интерфейсов достаточно подробно, чтобы можно было объективно ВЕРИФИЦИРОВАТЬ их БЕЗОПАСНОСТЬ и результативность, если это может быть обеспечено с использованием других требований или документации по разработке. Он должен быть достаточно полным, чтобы программисту не требовалось принимать исключительных проектных решений. Детальный дизайн также должен быть связан с архитектурой ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКОГО ИЗДЕЛИЯ.

ПРОГРАММНАЯ СОСТАВНАЯ ЧАСТЬ может подразделяться на уровни так, что только немногие из новых ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ имплементируют требования, связанные с БЕЗОПАСНОСТЬЮ исходной ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ. Оставшиеся ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ не имплементируют функции, связанные с БЕЗОПАСНОСТЬЮ, и могут быть повторно классифицированы с присвоением более низкого класса безопасности программного обеспечения. Однако принятие такого решения само по себе является частью ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА и документируется в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА.

Поскольку имплементация зависит от детального дизайна, необходимо проверить данный детальный дизайн до завершения ДЕЯТЕЛЬНОСТИ. ВЕРИФИКАЦИЯ детализированного дизайна, как правило, осуществляется путем ОЦЕНИВАНИЯ технических характеристик. Пункт 5.4.4 требует от ИЗГОТОВИТЕЛЯ верифицировать выходные данные ДЕЯТЕЛЬНОСТИ по детализированному дизайну. Дизайн определяет, какие требования должны быть реализованы. ВЕРИФИКАЦИЯ дизайна обеспечивает уверенность в том, что дизайн правильно реализует АРХИТЕКТУРУ программного обеспечения и не противоречит АРХИТЕКТУРЕ программного обеспечения.

Если дизайн будет содержать дефекты, то код не будет правильно реализовывать требования.

Если дизайн содержит дефекты, то ИЗГОТОВИТЕЛЬ должен проверить те характеристики дизайна, которые он считает важными для обеспечения БЕЗОПАСНОСТИ. Примеры таких характеристик включают:

- реализацию предусмотренных событий, входных и выходных данных, интерфейсов, логической схемы, а также вопросы распределения ресурсов процессора, распределения ресурсов памяти, определения ошибок и исключений, изоляции ошибок и исключений и восстановления после ошибок;
- определение состояния по умолчанию, в котором устраняются все отказы, которые могут привести к опасной ситуации, включая события и переходы;
- инициализацию переменных, управление памятью; и
- «холодную» и «теплую» перезагрузки, режим ожидания и другие изменения состояния, которые могут влиять на меры по УПРАВЛЕНИЮ РИСКОМ.

B.5.5 Имплементация и верификация ПРОГРАММНОГО БЛОКА

Данная ДЕЯТЕЛЬНОСТЬ требует от ИЗГОТОВИТЕЛЯ записать и проверить код для ПРОГРАММНЫХ БЛОКОВ.

Детальный дизайн преобразовывается в исходный код. Кодирование представляет собой момент, в котором заканчивается декомпозиция спецификаций и начинается составление реализуемого исполняемого ПО. Чтобы последовательно достигать желаемых характеристик кода, должны использоваться стандарты кодирования для определения предпочитаемого стиля кодирования. Примеры стандартов кодирования включают требования к понятности, правила использования языка или ограничений и сложность управления. Код для каждого модуля ВЕРИФИЦИРУЕТСЯ на предмет функционирования как определено в детальном дизайне, и что он соответствует установленным стандартам кодирования.

Пункт 5.5.5 требует от ИЗГОТОВИТЕЛЯ проверять код. Если код не реализует дизайн правильно, программное обеспечение МЕДИЦИНСКОГО ИЗДЕЛИЯ не будет функционировать так, как предназначено.

B.5.6 Интеграция и тестирование интеграции программного обеспечения

Данная ДЕЯТЕЛЬНОСТЬ требует от ИЗГОТОВИТЕЛЯ планировать и реализовывать интеграцию ПРОГРАММНЫХ БЛОКОВ в ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ так же, как и интеграцию ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ в более сложносоставные ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ, и проверять, что полученные в результате данной сборки ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ функционируют так, как предназначено.

Подход к интеграции может варьироваться от неинкрементной интеграции до любой формы пошаговой интеграции. Свойства собираемой ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ диктуют выбираемый метод интеграции.

Тестирование интеграции ПО направлено на передачу данных и управление всей ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ через внешние и внутренние интерфейсы. Внешние интерфейсы — это те, которые имеют другое программное обеспечение, включая программное обеспечение операционной системы и аппаратные средства МЕДИЦИНСКОГО ИЗДЕЛИЯ.

Точность тестирования интеграции и уровень детализации документации, связанной с тестированием интеграции, должны быть соизмеримы с РИСКОМ, связанным с изделием, с зависимостью изделия от программного обеспечения для потенциально опасных функций, а также с ролью определенных ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ в функционале изделия с большей степенью РИСКА. Например, несмотря на то, что все ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ должны быть протестированы, элементы, которые влияют на БЕЗОПАСНОСТЬ, должны подвергаться более направленным, всесторонним и детальным тестам.

В соответствующих случаях тестирование интеграции демонстрирует поведение программы на границах ее входных и выходных доменов (областей) и подтверждает реакцию ПО на недействительные, неожиданные и специальные входные данные. Действия программы обнаруживаются при введении комбинации входных данных или неожиданной последовательности входных данных, или когда нарушены определенные требования синхронизации. Требования тестирования в плане должны включать, соответственно, типы тестирования методом «белого ящика», чтобы быть выполненными как часть интеграционного тестирования.

Тестирование методом «белого ящика», также известное как тестирование «стеклянного ящика», «структурное», «прозрачного ящика» и «открытого ящика», — это техника тестирования, где используется точное знание внутреннего функционирования ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ, чтобы выбирать данные тестирования. Тестирование методом «белого ящика» использует определенные знания о ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ, чтобы проверять выходные данные. Это тестирование является точным, только если тестовый инженер знает, что ПРОГРАММНАЯ СОСТАВНАЯ ЧАСТЬ должна делать. Тогда тестовый инженер может видеть, когда ПРОГРАММНАЯ СОСТАВНАЯ ЧАСТЬ отклоняется от его намеченной цели. Тестирование методом «белого ящика» не может гарантировать, что была реализована полная спецификация (на программное обеспечение), поскольку оно фокусируется на тестировании реализации ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ. Тестирование методом «черного ящика», также известное как «поведенческое», «функциональное», тестирование «непрозрачного ящика» или тестирование «закрытого ящика», фокусируется на тестировании функциональной спецификации и не может гарантировать, что были протестированы все реализованные части. Таким образом, тестирование методом «черного ящика» является тестированием на спецификацию и обнаруживает дефекты пропусков, определяя, какая часть спецификации не была выполнена. Тестирование методом «белого ящика» является тестированием на реализацию и обнаруживает дефекты выполнения, указывая, какая часть реализации неисправна. Чтобы полностью протестировать ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МЕДИЦИНСКОГО ИЗДЕЛИЯ, могут потребоваться как тестирование методом «черного ящика», так и тестирование методом «белого ящика».

Планы и документация тестирования, определенные в подразделах 5.6 и 5.7, могут быть отдельными документами, привязанными к конкретным стадиям разработки или эволюционным прототипам. Они могут быть объединены в единый документ или набор документов, охватывающих требования множества подразделов. Все документы или часть документов могут быть включены в проектные документы более высокого уровня, такие как план обеспечения качества проекта или ПО, или план комплексного тестирования, который охватывает все аспекты тестирования аппаратных средств и программного обеспечения. В таких случаях следует создавать перекрестную ссылку, которая определяет, как различные документы проекта связаны с каждой из ЗАДАЧ интеграции программного обеспечения.

Тестирование интеграции программного обеспечения может осуществляться в моделируемой среде, на имеющемся оборудовании, или на полноценном МЕДИЦИНСКОМ ИЗДЕЛИИ.

Пункт 5.6.2 требует от ИЗГОТОВИТЕЛЯ верифицировать выходные данные ДЕЯТЕЛЬНОСТИ по интеграции программного обеспечения. Выходные данные ДЕЯТЕЛЬНОСТИ по интеграции программного обеспечения — это интегрированные (встроенные) ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ.

Данные интегрированные ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ должны функционировать должным образом, чтобы все программное обеспечение МЕДИЦИНСКОГО ИЗДЕЛИЯ функционировало правильно и безопасно.

B.5.7 Тестирование ПРОГРАММНОЙ СИСТЕМЫ

Данная ДЕЯТЕЛЬНОСТЬ требует от ИЗГОТОВИТЕЛЯ проверить функциональность программного обеспечения путем проверки того, что требования к программному обеспечению были успешно реализованы.

Тестирование ПРОГРАММНОЙ СИСТЕМЫ демонстрирует, что указанная функциональность действительно существует. Тестирование ВЕРИФИЦИРУЕТ функциональность и характеристики программы, как разработанной в соответствии с требованиями к программному обеспечению.

Тестирование ПРОГРАММНОЙ СИСТЕМЫ ориентировано на функциональное тестирование («черный ящик»), хотя более предпочтительным может оказаться использование метода «белого ящика» (см. B.5.6), чтобы эффективней выполнять определенные тесты, выделять стрессовые условия или дефекты либо увеличивать покрытие исходного кода квалификационных тестов. Организация тестирования по типам и этапам является гибкой, но покрытие требований, УПРАВЛЕНИЕ РИСКОМ, эксплуатационная пригодность и типы тестов (например, негативные, инсталляционные, стресс) должны быть продемонстрированы и задокументированы.

Тестирование ПРОГРАММНОЙ СИСТЕМЫ проверяет интегрированное программное обеспечение и может быть выполнено в моделируемой среде на имеющемся оборудовании или на полноценном МЕДИЦИНСКОМ ИЗДЕЛИИ.

Когда в ПРОГРАММНУЮ СИСТЕМУ вносятся изменения (даже небольшие), должна быть определена степень РЕГРЕССИОННОГО ТЕСТИРОВАНИЯ (но не только тестирования отдельных изменений), чтобы удостовериться в отсутствии непредусмотренных побочных явлений. Данное РЕГРЕССИОННОЕ ТЕСТИРОВАНИЕ (и обоснование для не полностью повторяемого тестирования ПРОГРАММНОЙ СИСТЕМЫ) должно быть запланировано и документировано. (См. B.6.3).

Ответственность за тестирование ПРОГРАММНОЙ СИСТЕМЫ может быть распределена, происходить в разных местах и проводиться различными организациями. Однако, независимо от распределения ЗАДАЧ, договорных отношений, источника компонентов или среды (условий) разработки, ИЗГОТОВИТЕЛЬ изделия сохраняет окончательную ответственность за обеспечение правильного функционирования программного обеспечения в соответствии с предусмотренным назначением.

Если при тестировании были обнаружены способные к повторению АНОМАЛИИ, но было принято решение не устранять их, то данные АНОМАЛИИ должны быть ОЦЕНЕНЫ в соответствии с анализом РИСКА, чтобы убедиться, что они не влияют на БЕЗОПАСНОСТЬ изделия. Необходимо понять первопричину и особенности проявления АНОМАЛИЙ, а также задокументировать причины, по которым они не устраняются.

Пункт 5.7.4 требует, чтобы результаты тестирования ПРОГРАММНОЙ СИСТЕМЫ были ОЦЕНЕНЫ, чтобы обеспечить получение ожидаемых результатов.

B.5.8 Выпуск ПО

Данная ДЕЯТЕЛЬНОСТЬ требует от ИЗГОТОВИТЕЛЯ документировать ВЕРСИЮ выпускаемого ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ, указать, как оно было создано, и следовать соответствующим для выпуска программного обеспечения процедурам.

ИЗГОТОВИТЕЛЬ должен быть способен продемонстрировать, что программное обеспечение, созданное с использованием ПРОЦЕССА разработки, — это то программное обеспечение, которое было выпущено. ИЗГОТОВИТЕЛЬ должен иметь возможность восстановить программное обеспечение и инструменты, использованные для его создания, в случае если это понадобится в будущем. Он должен хранить, упаковывать и доставлять программное обеспечение способом, минимизирующим возможность повреждения или неправильного применения. Должны быть установлены определенные процедуры, чтобы обеспечить выполнение ЗАДАЧ надлежащим образом и с последовательными результатами.

B.6 ПРОЦЕСС технической поддержки ПО

B.6.1 Установление плана технической поддержки ПО

ПРОЦЕСС технической поддержки программного обеспечения отличается от ПРОЦЕССА разработки программного обеспечения двумя пунктами:

- ИЗГОТОВИТЕЛЮ разрешается использовать ПРОЦЕСС, меньший, чем полный ПРОЦЕСС разработки программного обеспечения, чтобы осуществлять быстрые изменения в ответ на неотложные проблемы;
- в ответ на программные ОТЧЕТЫ О ПРОБЛЕМАХ, относящихся к выпущенному продукту, ИЗГОТОВИТЕЛЬ не только решает проблему, но еще и выполняет локальные регулирующие требования (обычно запуская активную схему наблюдения для сбора данных о проблеме и ее области и для общения с пользователями и регулирующими органами о проблеме).

Подраздел 6.1 требует, чтобы эти ПРОЦЕССЫ были установлены в плане технической поддержки.

Данная ДЕЯТЕЛЬНОСТЬ требует от ИЗГОТОВИТЕЛЯ создания или идентификации процедуры для реализации ДЕЯТЕЛЬНОСТИ и ЗАДАЧ по технической поддержке. Чтобы выполнять корректирующие действия, управлять изменениями при технической поддержке и управлять выпуском обновленного ПО, ИЗГОТОВИТЕЛЮ следует документировать и решить проблемы и запросы потребителей, а также управлять модификациями ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ. Этот ПРОЦЕСС активизируется, когда из-за проблем либо потребности в улучшении или адаптации ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ подвергается модификациям кода или изменяется сопутствующая документация. Цель состоит в сохранении целостности выпущенного ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ при его модификации. Этот ПРОЦЕСС включает перемещение ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ в среду или на платформы, для которых оно первоначально не было выпущено. ДЕЯТЕЛЬНОСТЬ, предусмотренная настоящим пунктом, характерна для ПРОЦЕССА технической поддержки, однако ПРОЦЕСС технической поддержки может использовать другие ПРОЦЕССЫ настоящего стандарта.

ИЗГОТОВИТЕЛЮ нужно планировать ДЕЯТЕЛЬНОСТЬ и ЗАДАЧИ ПРОЦЕССА технической поддержки.

B.6.2 Анализ проблем и модификаций

Данная ДЕЯТЕЛЬНОСТЬ требует от ИЗГОТОВИТЕЛЯ анализировать обратную связь на предмет ее значимости; проверять сообщения о проблемах и рассматривать, выбирать и одобрять подходящие для выполнения возможные варианты модификаций.

Проблемы и другие запросы на внесение изменений могут повлиять на функциональные характеристики, БЕЗОПАСНОСТЬ или регистрацию МЕДИЦИНСКОГО ИЗДЕЛИЯ в регулирующих органах. Анализ необходим для определения каких-нибудь последствий из-за ОТЧЕТА О ПРОБЛЕМАХ и появятся ли какие-нибудь последствия из-за модификации, а также для устранения проблемы или выполнения запроса. Для проверки посредством анализа прослеживаемости или регрессионного анализа особенно важно, чтобы встроенные в изделие меры по УПРАВЛЕНИЮ РИСКОМ не были негативным образом изменены или модифицированы программным обеспечением, которое внедряется как часть ДЕЯТЕЛЬНОСТИ по технической поддержке ПО. Также важно убедиться, что измененное программное обеспечение не создает ОПАСНОЙ СИТУАЦИИ или снижает РИСК в программном обеспечении, которое ранее не создавало ОПАСНОЙ СИТУАЦИИ или снижало РИСКИ. Классификация безопасности программного обеспечения для ПРОГРАММНОЙ СОСТАВНОЙ ЧАСТИ может быть изменена, если модификация программного обеспечения в настоящий момент может вызывать ОПАСНОСТЬ или уменьшать РИСК.

Важно различать техническую поддержку программного обеспечения (раздел 6) и решение проблем программного обеспечения (раздел 9).

Главным в ПРОЦЕССЕ технической поддержки программного обеспечения является достаточный ответ на обратную связь, возникающую после выпуска ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ. Как часть МЕДИЦИНСКОГО ИЗДЕЛИЯ, ПРОЦЕСС технической поддержки программного обеспечения должен обеспечить уверенность в том, что:

- ОТЧЕТЫ О ПРОБЛЕМАХ, связанные с БЕЗОПАСНОСТЬЮ, рассматриваются и доводятся до сведения соответствующих регулирующих органов и затронутых пользователей;
- ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МЕДИЦИНСКОГО ИЗДЕЛИЯ повторно одобряется и повторно выпускается после модификации и официального контроля, которые обеспечивают устранение проблемы и предотвращение дальнейших проблем;
- ИЗГОТОВИТЕЛЬ рассматривает, какое другое ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МЕДИЦИНСКОГО ИЗДЕЛИЯ может быть затронуто и предпринимает соответствующие действия.

Центром внимания решения проблем ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ является функционирование комплексной системы управления, которая:

- анализирует ОТЧЕТЫ О ПРОБЛЕМАХ и идентифицирует все последствия этой проблемы;
- принимает решения по ряду изменений и определяет их любое побочное воздействие;
- осуществляет изменения, сохраняя при этом согласованность ПРОГРАММНЫЕ СОСТАВНЫЕ ЧАСТИ КОНФИГУРАЦИИ, в том числе в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА;
- ВЕРИФИЦИРУЕТ осуществление изменений.

Процесс технической поддержки программного обеспечения использует ПРОЦЕСС решения проблем программного обеспечения. ПРОЦЕСС технической поддержки программного обеспечения рассматривает ОТЧЕТ О ПРОБЛЕМАХ на высоком уровне (существует ли проблема, имеет ли она существенное влияние на БЕЗОПАСНОСТЬ, какие изменения необходимы и когда их осуществлять) и использует ПРОЦЕСС решения проблем программного обеспечения для анализа ОТЧЕТА О ПРОБЛЕМАХ с целью обнаружения любых последствий и создания возможных ЗАПРОСОВ НА ИЗМЕНЕНИЯ, которые идентифицируют все нуждающиеся в изменении СОСТАВНЫЕ ЧАСТИ КОНФИГУРАЦИИ, а также все необходимые шаги по ВЕРИФИКАЦИИ.

B.6.3 Осуществление модификации

Данная ДЕЯТЕЛЬНОСТЬ требует, чтобы ИЗГОТОВИТЕЛЬ использовал установленные ПРОЦЕССЫ для выполнения модификации. Если ПРОЦЕСС технической поддержки не был установлен, для осуществления модификации могут использоваться подходящие ЗАДАЧИ ПРОЦЕССА разработки. ИЗГОТОВИТЕЛЬ должен также обеспечить уверенность в том, что модификация не вызывает отрицательного влияния на другие части ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ. Если ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ не рассматривается как новая разработка, необходим анализ влияния модификации на все ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ. Регрессионный анализ и тестирование используются для обеспечения уверенности в том, что изменение не создало проблем в других частях ПО

МЕДИЦИНСКОГО ИЗДЕЛИЯ. Регрессионный анализ — это определение влияния изменения на основе анализа соответствующей документации (например, спецификаций требований к программному обеспечению, спецификаций разработки программного обеспечения, исходного кода, планов тестирования, тестовых примеров, тестовых сценариев и т. д.) для определения необходимых регрессионных тестов, которые необходимо выполнить. Регрессионное тестирование — это повторный запуск тестовых случаев, которые программа ранее выполнила правильно, и сравнение текущего результата с предыдущим результатом для выявления непреднамеренных последствий изменения программного обеспечения. Должно быть сделано обоснование, оправдывающее количество РЕГРЕССИОННЫХ ТЕСТОВ, которое будет выполняться для обеспечения уверенности в том, что части еще не модифицированного ПО МЕДИНСКОГО ИЗДЕЛИЯ продолжают работать так, как и до выполнения модификации.

B.7 ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА программного обеспечения

МЕНЕДЖМЕНТ РИСКА программного обеспечения — это часть полного МЕНЕДЖМЕНТА РИСКА МЕДИЦИНСКОГО ИЗДЕЛИЯ, которая не может надлежащим образом быть рассмотрена изолировано. Настоящий стандарт требует использования ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА, соответствующего ИСО 14971:2007. Как определено в ИСО 14971:2007, МЕНЕДЖМЕНТ РИСКА представляет собой основу для результативного МЕНЕДЖМЕНТА РИСКА в отношении МЕДИЦИНСКИХ ИЗДЕЛИЙ. Одна из частей ИСО 14971:2007 относится к УПРАВЛЕНИЮ идентифицированными РИСКАМИ, связанными с каждой ОПАСНОСТЬЮ, выявленной в ходе АНАЛИЗА РИСКА. ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА программного обеспечения в настоящем стандарте предназначен для установления дополнительных требований к УПРАВЛЕНИЮ РИСКОМ для программного обеспечения, включая программное обеспечение, которое было определено в ходе АНАЛИЗА РИСКОВ как потенциально способствующее опасным ситуациям, или программное обеспечение, которое используется для УПРАВЛЕНИЯ РИСКОМ МЕДИЦИНСКОГО ИЗДЕЛИЯ. ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА программного обеспечения включен в настоящий стандарт по двум причинам:

- целевая аудитория данного стандарта должна понимать минимальные требования в отношении мер по УПРАВЛЕНИЮ РИСКОМ в зоне их ответственности — программном обеспечении;
- общий стандарт по МЕНЕДЖМЕНТУ РИСКА, ИСО 14971:2007, приведенный в качестве нормативной ссылки к настоящему стандарту, не охватывает специально УПРАВЛЕНИЕ РИСКОМ ПО и место УПРАВЛЕНИЯ РИСКОМ в жизненном цикле разработки ПО.

МЕНЕДЖМЕНТ РИСКА программного обеспечения — это часть общего МЕНЕДЖМЕНТА РИСКА МЕДИЦИНСКОГО ИЗДЕЛИЯ. Планы, процедуры и документация, требуемые для ДЕЯТЕЛЬНОСТИ по МЕНЕДЖМЕНТУ РИСКА программного обеспечения, могут быть серией отдельных документов или одним документом, или они могут быть интегрированы в ДЕЯТЕЛЬНОСТЬ по МЕНЕДЖМЕНТУ РИСКА МЕДИЦИНСКОГО ИЗДЕЛИЯ и в документацию, при условии, что выполняются все требования настоящего стандарта.

B.7.1 Анализ программного обеспечения, способствующего опасным ситуациям

Ожидается, что анализ ОПАСНОСТИ изделия будет идентифицировать опасные ситуации и соответствующие меры по УПРАВЛЕНИЮ РИСКОМ для уменьшения вероятности и/или тяжести причинения вреда от этих опасных ситуаций до допустимого уровня. Также предполагается, что меры по УПРАВЛЕНИЮ РИСКОМ будут возложены на программный функционал, который, как ожидается, будет реализовывать данные меры по УПРАВЛЕНИЮ РИСКОМ.

Однако вряд ли можно ожидать, что все опасные ситуации присущие изделию могут быть идентифицированы до того, как будет подготовлена программная АРХИТЕКТУРА. В то же время известно, как функции программного обеспечения будут воплощены в программных компонентах, и может быть ОЦЕНЕНА практичность мер по УПРАВЛЕНИЮ РИСКОМ, назначенных функциям ПО. Также следует пересмотреть анализ ОПАСНОСТИ изделия, чтобы включить:

- пересмотренные опасные ситуации;
- пересмотренные меры по УПРАВЛЕНИЮ РИСКОМ и требования к программному обеспечению;
- новые опасные ситуации, возникающие из-за программного обеспечения, например опасные ситуации, связанные с человеческим фактором.

АРХИТЕКТУРА программного обеспечения должна включать надежные стратегии для разделения компонентов программного обеспечения таким образом, чтобы они не взаимодействовали опасным способом.

B.8 ПРОЦЕСС менеджмента конфигурации программного обеспечения

ПРОЦЕСС менеджмента конфигурации программного обеспечения — это ПРОЦЕСС применения административных и технических процедур на протяжении жизненного цикла программного обеспечения для идентификации и определения ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ, включая документацию, в СИСТЕМЕ; управление изменениями и выпуском элементов; документирование и сообщение о состоянии элементов и ЗАПРОСОВ НА ИЗМЕНЕНИЯ. Управление конфигурацией программного обеспечения необходимо, чтобы обновить ПРОГРАММНУЮ СОСТАВНУЮ ЧАСТЬ, идентифицировать его составные части и предоставить историю изменений, которые были в нем осуществлены.

B.8.1 Идентификация конфигурации

Данная ДЕЯТЕЛЬНОСТЬ требует от ИЗГОТОВИТЕЛЯ однозначной идентификации СОСТАВНЫХ ЧАСТЕЙ КОНФИГУРАЦИИ программного обеспечения и их ВЕРСИЙ. Эта идентификация необходима, чтобы определять СОСТАВНЫЕ ЧАСТИ КОНФИГУРАЦИИ ПО и ВЕРСИИ, которые включены в ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ.

B.8.2 Управление изменениями

Данная ДЕЯТЕЛЬНОСТЬ требует от ИЗГОТОВИТЕЛЯ управлять изменениями СОСТАВНЫХ ЧАСТЕЙ КОНФИГУРАЦИИ программного обеспечения и регистрировать информацию, определяющую ЗАПРОСЫ НА ИЗМЕНЕНИЯ и предоставление документации об их местонахождении. Данная ДЕЯТЕЛЬНОСТЬ необходима, чтобы обеспечить уверенность в том, что несанкционированные или непреднамеренные изменения не были внесены в СОСТАВНЫЕ ЧАСТИ КОНФИГУРАЦИИ программного обеспечения и что одобренные ЗАПРОСЫ НА ИЗМЕНЕНИЯ были полностью осуществлены и ВЕРИФИЦИРОВАНЫ.

ЗАПРОСЫ НА ИЗМЕНЕНИЯ могут быть одобрены группой по управлению изменениями, менеджером или техническим руководством согласно плану управления конфигурацией программного обеспечения. Одобренные ЗАПРОСЫ НА ИЗМЕНЕНИЕ прослеживаются до фактической модификации и ВЕРИФИКАЦИИ программного обеспечения. Необходимо, чтобы каждое фактическое изменение было связано с ЗАПРОСОМ НА ИЗМЕНЕНИЕ и существовала документация, показывающая, что ЗАПРОС НА ИЗМЕНЕНИЕ был одобрен. Документация может быть изменена группой по управлению изменениями, подписью или записью в базе данных.

B.8.3 Учет статуса конфигурации

Данная ДЕЯТЕЛЬНОСТЬ требует от ИЗГОТОВИТЕЛЯ поддерживать записи истории СОСТАВНЫХ ЧАСТЕЙ КОНФИГУРАЦИИ программного обеспечения. Эта ДЕЯТЕЛЬНОСТЬ необходима, чтобы определять, когда и где были сделаны изменения. Доступ к этой информации нужен для обеспечения уверенности в том, что СОСТАВНЫЕ ЧАСТИ КОНФИГУРАЦИИ ПО содержат только разрешенные модификации.

B.9 ПРОЦЕСС решения проблем программного обеспечения

ПРОЦЕСС решения проблем программного обеспечения — это ПРОЦЕСС для анализа и решения проблем (включая несоответствия), вне зависимости от их природы или источника, включая те, которые обнаружены по время выполнения ПРОЦЕССОВ разработки, технической поддержки и других. Цель состоит в предоставлении своевременных и документально подтвержденных средств обеспечения того, что обнаруженные проблемы анализируются и решаются и что тенденции замечены. Данный ПРОЦЕСС в литературе, касающейся разработки программного обеспечения, иногда называется «отслеживание дефекта». В ИСО/МЭК 12207 [9] и МЭК 60601-1-4 [2], поправка 1, он называется «решение проблем». Для целей настоящего стандарта был принято решение называть ПРОЦЕСС «решение проблем программного обеспечения».

Данная ДЕЯТЕЛЬНОСТЬ требует от ИЗГОТОВИТЕЛЯ использовать ПРОЦЕСС решения проблем, когда определены проблема или несоответствие. Данная деятельность необходима, чтобы обеспечить уверенность в том, что обнаруженные проблемы проанализированы и ОЦЕНЕНЫ на возможное отношение их к БЕЗОПАСНОСТИ (как определено в ИСО 14971:2007).

План (планы) или процедуры разработки программного обеспечения, как требуется в 5.1, состоят в том, как будут обработаны проблемы или несоответствия. Это включает определение на каждой стации жизненного цикла аспектов ПРОЦЕССА решения проблем программного обеспечения, которые будут надлежащим образом оформлены и зарегистрированы тогда, когда проблемы и несоответствия будут введены в ПРОЦЕСС решения проблем программного обеспечения.

Приложение С
(справочное)

Взаимосвязь с другими стандартами

С.1 Общие положения

Настоящий стандарт применяется к разработке и технической поддержке программного обеспечения МЕДИЦИНСКИХ ИЗДЕЛИЙ. Программное обеспечение может быть подсистемой МЕДИЦИНСКОГО ИЗДЕЛИЯ или являться самостоятельным МЕДИЦИНСКИМ ИЗДЕЛИЕМ. Настоящий стандарт предназначен для использования совместно с другими подходящими стандартами на процессы разработки МЕДИЦИНСКИХ ИЗДЕЛИЙ.

Стандарты по менеджменту МЕДИЦИНСКИХ ИЗДЕЛИЙ, такие как ISO 13485:2003 [8] (см. С.2 и приложение D) и ISO 14971:2007, обеспечивают менеджмент окружения (среды), что закладывает основу для организации разработки продукции. Стандарты по БЕЗОПАСНОСТИ, такие как IEC 60601-1 [1] (см. приложение С.4) и IEC 61010-1 [5] (см. приложение С.5), дают определенное руководство по созданию безопасных МЕДИЦИНСКИХ ИЗДЕЛИЙ. Когда программное обеспечение является составной частью таких МЕДИЦИНСКИХ ИЗДЕЛИЙ, настоящий стандарт содержит более детальное руководство относительно требований к разработке и поддержанию безопасности ПО МЕДИЦИНСКИХ ИЗДЕЛИЙ. Многие другие стандарты, такие как ISO/IEC 12207 [9] (см. приложение С.6), IEC 61508-3 [4] (см. приложение С.7) и ISO/IEC 90003 [15], могут рассматриваться как источники методов, инструментов и техник, которые следует использовать для выполнения требований настоящего стандарта. Рисунок С.1 показывает взаимосвязь между этими стандартами.

Если цитируются положения или требования других стандартов, используемые термины в цитируемых элементах терминов являются терминами, которые определены в другом стандарте и не определены в настоящем стандарте.

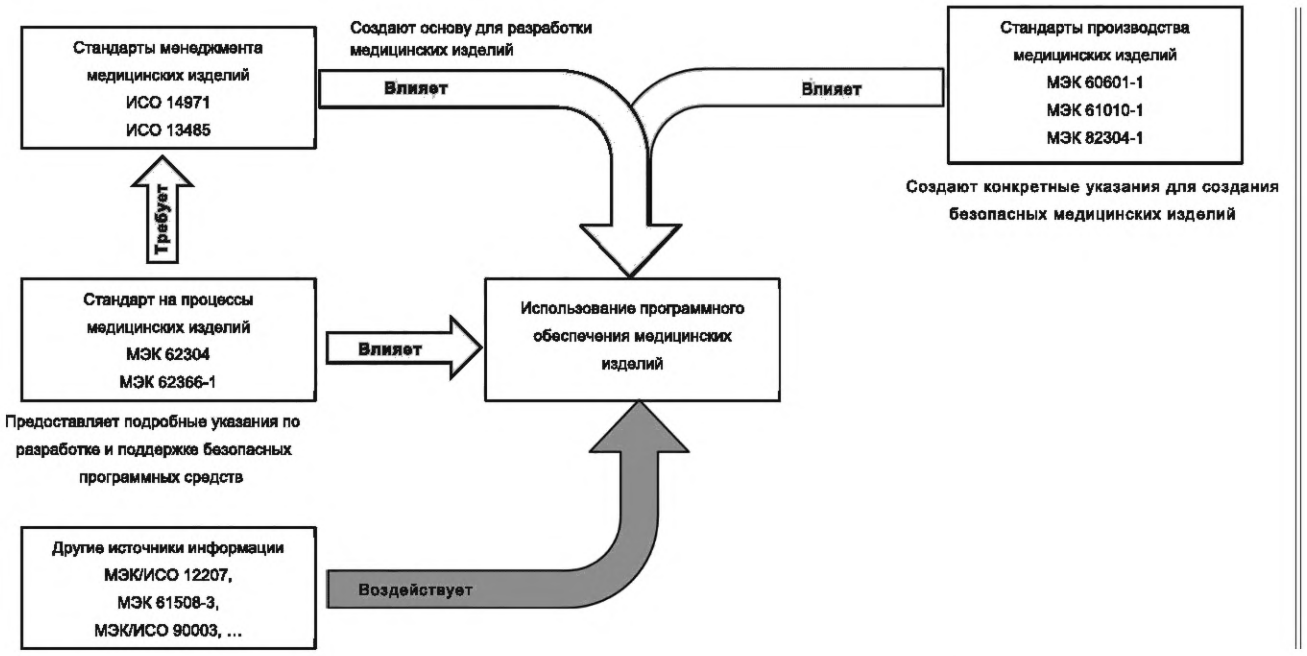


Рисунок С.1 — Взаимосвязь ключевых стандартов на МЕДИЦИНСКИЕ ИЗДЕЛИЯ с IEC 62304

С.2 Взаимосвязь с ISO 13485

Настоящий стандарт требует, чтобы изготовитель использовал систему менеджмента качества. Когда изготовитель использует ISO 13485 [8], требования настоящего стандарта непосредственно связаны с требованиями ISO 13485:2003, как это показано в таблице С.1.

Таблица С.1 — Взаимосвязь с ISO 13485:2003

Разделы/подразделы настоящего стандарта	Соответствующие пункты ISO 13485:2003
1	2
5.1 Планирование разработки программного обеспечения	7.3.1 Планирование проектирования и разработки
5.2 Анализ требований к программному обеспечению	7.3.2 Входные данные для проектирования и разработки
5.3 Проектирование АРХИТЕКТУРЫ программного обеспечения	
5.4 Разработка детального дизайна программного обеспечения	
5.5 Имплементация ПРОГРАММНЫХ БЛОКОВ	
5.6 Интеграция программного обеспечения и тестирование интеграции	
5.7 Тестирование ПРОГРАММНОЙ СИСТЕМЫ	7.3.3 Выходные данные проектирования и разработки 7.3.4 Анализ проекта и разработки
5.8 Выпуск программного обеспечения на системном уровне	7.3.5 ВЕРИФИКАЦИЯ проектирования и разработки 7.3.6 Валидация проектирования и разработки
6.1 Установление плана технической поддержки программного обеспечения	7.3.7 Управление изменениями проекта и разработки
6.2 Анализ модификации и проблем	
6.3 Осуществление модификации	7.3.5 ВЕРИФИКАЦИЯ проектирования и разработки 7.3.6 Валидация проектирования и разработки
7.1 Анализ программного обеспечения, способствующего опасным ситуациям	
7.2 Меры по УПРАВЛЕНИЮ РИСКОМ	
7.3 ВЕРИФИКАЦИЯ мер по УПРАВЛЕНИЮ РИСКОМ	
7.4 МЕНЕДЖМЕНТ РИСКА в отношении изменений программного обеспечения	
8.1 Идентификация конфигурации	7.5.3 Идентификация и ПРОСЛЕЖИВАЕМОСТЬ
8.2 Управление изменениями	7.5.3 Идентификация и ПРОСЛЕЖИВАЕМОСТЬ
8.3 Учет статуса конфигурации	
9 Процесс решения проблем программного обеспечения	

С.3 Взаимосвязь с ISO 14971:2007

Таблица С.2 показывает области, где настоящий стандарт усиливает требования к ПРОЦЕССУ МЕНЕДЖМЕНТА РИСКА, требуемого ISO 14971.

Таблица С.2 — Взаимосвязь с ISO 14971:2007

Разделы/подразделы ISO 14971:2007	Соответствующие подразделы и пункты настоящего стандарта
4.1 Процесс АНАЛИЗА РИСКА	
4.2 Предусмотренное применение и определение характеристик, относящихся к БЕЗОПАСНОСТИ МЕДИЦИНСКОГО ИЗДЕЛИЯ	
4.3 Идентификация ОПАСНОСТЕЙ	7.1 Анализ программного обеспечения, способствующего опасным ситуациям
4.4 Определение РИСКОВ для каждой ОПАСНОЙ СИТУАЦИИ	4.3 *Классификация программного обеспечения в отношении безопасности
5 ОЦЕНИВАНИЕ РИСКА	
6.1 Уменьшение риска	
6.2 Анализ возможностей УПРАВЛЕНИЯ РИСКОМ	7.2.1 Определение мер по УПРАВЛЕНИЮ РИСКОМ
6.3 Выполнение мер по УПРАВЛЕНИЮ РИСКОМ	7.2.2 Меры по УПРАВЛЕНИЮ РИСКОМ, реализованные в программном обеспечении 7.3.1 Проверка мер по УПРАВЛЕНИЮ РИСКОМ
6.4 ОЦЕНИВАНИЕ ОСТАТОЧНОГО РИСКА	
6.5 Анализ соотношения риск/польза	
6.6 РИСКИ, возникающие в результате МЕР ПО УПРАВЛЕНИЮ РИСКАМИ	7.3.2 Не применяется
6.7 Полнота УПРАВЛЕНИЯ РИСКАМИ	
7 ОЦЕНИВАНИЕ допустимости совокупного ОСТАТОЧНОГО РИСКА	
8 Отчет по МЕНЕДЖМЕНТУ РИСКА	7.3.3 Документирование ПРОСЛЕЖИВАЕМОСТИ
9 Производственная и постпроизводственная информация	7.4 МЕНЕДЖМЕНТ РИСКА в отношении изменений программного обеспечения

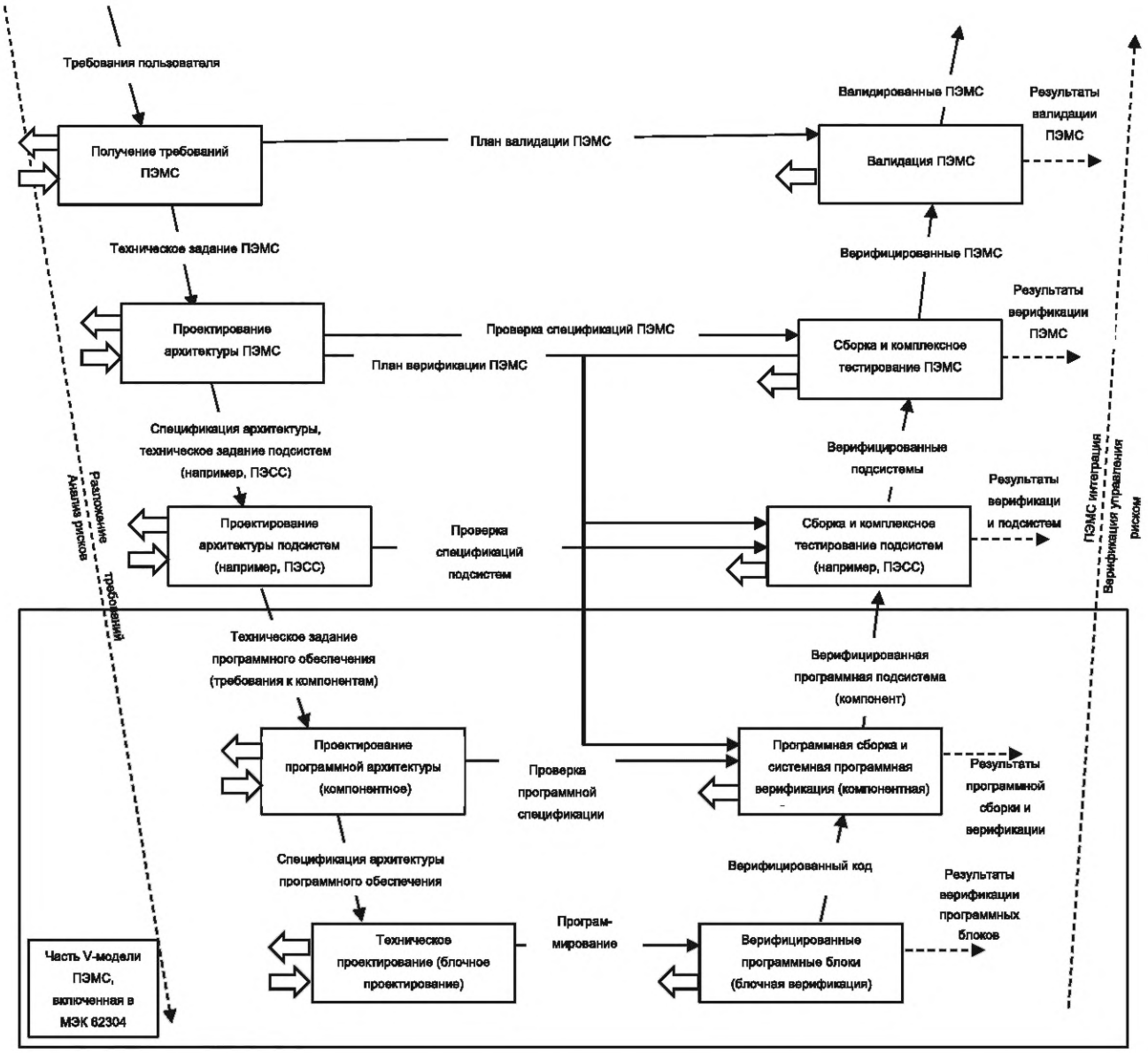
С.4 Взаимосвязь ПЭМС с требованиями МЭК 60601-1:2005 + МЭК 60601-1:2005/AMD1:2012

С.4.1 Общие положения

Требования к ПО — это подмножество требований к программируемой электрической медицинской системе (ПЭМС). Настоящий стандарт определяет требования к ПО, которые являются дополнительными, но не являются несовместимыми с требованиями IEC 60601-1:2005 + IEC 60601-1:2005 /AMD1:2012 [1] к ПЭМС. Поскольку ПЭМС включает элементы, не являющиеся ПО, не все требования IEC 60601-1:2005 + IEC 60601-1:2005/AMD1:2012 к ПЭМС отражены в настоящем стандарте. С публикацией IEC 60601-1:2005 + IEC 60601-1:2005 /AMD1:2012, IEC 62304 теперь является нормативным справочником IEC 60601-1, и соответствие пункту 14 IEC 60601-1:2005 + IEC 60601-1:2005/AMD1:2012 (и, следовательно, соответствие стандарту) требует соответствия частям IEC 62304 (не всему IEC 62304, потому что IEC 60601-1:2005 + IEC 60601-1:2005/AMD1:2012 не требует соблюдения требований к постпроизводству и техническому обслуживанию IEC 62304). Наконец, важно помнить, что IEC 60601-1:2005 + IEC 60601-1:2005/AMD1:2012 применяется только в том случае, если программное обеспечение является частью ПЭМС, а не если программное обеспечение само по себе является МЕДИЦИНСКИМ ИЗДЕЛИЕМ.

С.4.2 Взаимосвязь ПО с разработкой ПЭМС

Используя V-модель, показанную на рисунке С.2, для описания того, что происходит во время разработки ПЭМС, можно увидеть, что требования настоящего стандарта применяются на уровне компонентов ПЭМС, от спецификации требований программного обеспечения до интеграции ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ в ПРОГРАММНУЮ СИСТЕМУ. Эта ПРОГРАММНАЯ СИСТЕМА — часть программируемой электрической подсистемы (ПЭСС), являющейся, в свою очередь, частью ПЭМС.



Блоки — типичные виды деятельности жизненного цикла разработки; сплошные стрелки — типичные результаты, передаваемые в/из видов деятельности; пунктирные стрелки — результаты, относящиеся только к файлу менеджмента риска;

⇨ — результат процесса разрешения проблем; ⇦ — исходные данные процесса разрешения проблем

Рисунок С.2 — ПО как часть V-модели

С.4.3 ПРОЦЕСС разработки

Соответствие ПРОЦЕССУ разработки программного обеспечения в настоящем стандарте (раздел 5) требует, чтобы план разработки программного обеспечения был определен и соблюдался; это не требует, чтобы использовалась некая определенная модель жизненного цикла, но требует, чтобы план включал определенные виды ДЕЯТЕЛЬНОСТИ и имел определенные признаки. Эти требования соотносятся с требованиями ПЭМС в IEC 60601 к разработке жизненного цикла, спецификации требований, АРХИТЕКТУРЕ, проектированию и осуществлению, а также ВЕРИФИКАЦИИ. Требования в этом стандарте более детальны в области разработки ПО, чем требования IEC 60601-1.

С.4.4 ПРОЦЕСС технического обслуживания

Соответствие ПРОЦЕССУ технического обслуживания программного обеспечения в настоящем стандарте (раздел 6) требует, чтобы процедуры были установлены и соблюдались, когда в ПО вносятся изменения. Эти требования соответствуют требованиям IEC 60601-1 для модификации ПЭМС. Требования в настоящем стандарте относительно технического обслуживания программного обеспечения предоставляют более подробную информа-

цию о том, что должно быть сделано для технической поддержки программного обеспечения, чем требования для модификации ПЭМС в IEC 60601-1.

С.4.5 Прочие ПРОЦЕССЫ

Прочие ПРОЦЕССЫ в настоящем стандарте определяют дополнительные требования к программному обеспечению сверх подобных требований к ПЭМС в IEC 60601-1. В большинстве случаев существует общее требование к ПЭМС в IEC 60601-1, которое расширяет ПРОЦЕССЫ в настоящем стандарте.

ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА программного обеспечения в настоящем стандарте соответствует дополнительным требованиям к МЕНЕДЖМЕНТУ РИСКА, определенным для ПЭМС в IEC 60601-1.

ПРОЦЕСС решения проблем программного обеспечения в настоящем стандарте соответствует требованию к решению проблем для ПЭМС в IEC 60601-1.

ПРОЦЕСС менеджмента конфигурации программного обеспечения в настоящем стандарте устанавливает дополнительные требования, которые отсутствуют для ПЭМС в IEC 60601-1, за исключением документации.

С.4.6 Охват требований к ПЭМС в IEC 60601-1:2005 + IEC 60601-1:2005 /AMD1:2012

Таблица С.3 — Взаимосвязь с IEC 60601-1 (1 из 5)

Требования к ПЭМС в IEC 60601-1:2005 + IEC 60601-1:2005/AMD1:2012	Требования IEC 62304, связанные с программным обеспечением подсистемы ПЭМС
14.1 Общие положения Требования 14.2—14.12 (включительно) применяются к ПЭМС только в тех случаях, когда: - ни одна из ПРОГРАММИРУЕМЫХ ЭЛЕКТРОННЫХ ПОДСИСТЕМ (ПЭСС) не задействована в обеспечении ОСНОВНОЙ БЕЗОПАСНОСТИ или ОСНОВНЫХ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК или - применение МЕНЕДЖМЕНТА РИСКА как описано в 4.2, показывает, что отказ любой ПЭСС не приводит к возникновению недопустимого РИСКА. Требования 14.13 применимы к любым ПЭМС, предназначенным для включения в ИТ-СЕТЬ, независимо от применения требований 14.2—4.12. В случае применения требований 14.2—14.13, требования 4.3, разделов 5, 7, 8 и 9 IEC 62304:2006 также применяются к разработке или модификации программного обеспечения для каждого ПЭСС	4.3* Классификация программного обеспечения в отношении безопасности Требования к ПЭМС, установленные в IEC 60601-1, применимы только к программному обеспечению класса безопасности B и C. Настоящий стандарт содержит некоторые требования в отношении программного обеспечения класса безопасности A. ПРОЦЕСС разработки программного обеспечения, необходимый для соответствия IEC 60601-1, не включает последующий мониторинг и техническую поддержку, требуемые разделом 6 IEC 62304:2006
14.2 Документирование Документы, требуемые разделом 14, должны рассматриваться, утверждаться, выпускаться и изменяться в соответствии с официальной процедурой управления документацией	5.1* Планирование разработки программного обеспечения В дополнение к конкретным требованиям к ДЕЯТЕЛЬНОСТИ по планированию разработки программного обеспечения документы, которые являются частью ФАЙЛА МЕНЕДЖМЕНТА РИСКА, должны поддерживаться в соответствии с требованиями ISO 14971. Кроме того, ISO 13485 [8] требует управления документами системы качества
14.3 План МЕНЕДЖМЕНТА РИСКА План МЕНЕДЖМЕНТА РИСКА, требуемый согласно 4.2.2, должен включать ссылку на план ВЕРИФИКАЦИИ ПЭМС (см. 14.11)	Нет специальных требований. Не существует никакого определенного плана валидации программного обеспечения. План валидации ПЭМС относится к уровню СИСТЕМЫ и находится вне области применения настоящего стандарта на программное обеспечение. Настоящий стандарт требует ПРОСЛЕЖИВАЕМОСТИ от ОПАСНОСТИ определенного события с программным обеспечением до меры по УПРАВЛЕНИЮ РИСКОМ и к ВЕРИФИКАЦИИ меры по УПРАВЛЕНИЮ РИСКОМ (см. 7.3)

Продолжение таблицы С.3

Требования к ПЭМС в IEC 60601-1:2005 + IEC 60601-1:2005/AMD1:2012	Требования IEC 62304, связанные с программным обеспечением подсистемы ПЭМС
14.4 ЖИЗНЕННЫЙ ЦИКЛ РАЗРАБОТКИ ПЭМС ЖИЗНЕННЫЙ ЦИКЛ РАЗРАБОТКИ ПЭМС должен быть документально оформлен. ЖИЗНЕННЫЙ ЦИКЛ РАЗРАБОТКИ ПЭМС должен состоять из набора определенных этапов	5.1* Планирование разработки программного обеспечения 5.1.1 План разработки программного обеспечения Пункты, на которые ссылается план разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, составляют ЖИЗНЕННЫЙ ЦИКЛ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
На каждом этапе должна быть определена ДЕЯТЕЛЬНОСТЬ, которая должна быть завершена, а также методы ВЕРИФИКАЦИИ, которые должны применяться в отношении этой ДЕЯТЕЛЬНОСТИ	5.1.6 Планирование ВЕРИФИКАЦИИ программного обеспечения Должны быть запланированы ЗАДАЧИ ВЕРИФИКАЦИИ, этапы и критерии приемки
Каждая ДЕЯТЕЛЬНОСТЬ должна определяться с указанием входных и выходных параметров	5.1.1 План разработки программного обеспечения Вся ДЕЯТЕЛЬНОСТЬ определена в настоящем стандарте. Документация, которая должна быть разработана, определена для каждой ДЕЯТЕЛЬНОСТИ
На каждом этапе должны определяться работы по МЕНЕДЖМЕНТУ РИСКА, которые необходимо завершить перед этим этапом	5.1.1 План разработки программного обеспечения В соответствии с настоящим стандартом разработка жизненного цикла должна быть документирована в плане разработки. Это означает, что план разработки должен содержать разработку конкретного жизненного цикла
ЖИЗНЕННЫЙ ЦИКЛ РАЗРАБОТКИ ПЭМС должен составляться для каждой разработки путем создания планов, в которых уточняются работы, этапы и графики их выполнения	
ЖИЗНЕННЫЙ ЦИКЛ РАЗРАБОТКИ ПЭМС должен также включать требования к документации	5.1.1 План разработки программного обеспечения 5.1.8 Документация по планированию
14.5 Решение проблем Если целесообразно, должна быть разработана и поддерживаться документированная система решения проблем, возникающих на каждом этапе ДЕЯТЕЛЬНОСТИ (и между ними) ЖИЗНЕННОГО ЦИКЛА РАЗРАБОТКИ ПЭМС	9 ПРОЦЕСС решения проблем программного обеспечения
В зависимости от типа продукции СИСТЕМА решения проблем может: - регистрироваться как часть ЖИЗНЕННОГО ЦИКЛА РАЗРАБОТКИ ПЭМС; - позволять уведомлять о потенциальных или возникающих проблемах, затрагивающих ОСНОВНУЮ БЕЗОПАСНОСТЬ или ОСНОВНЫЕ ФУНКЦИОНАЛЬНЫЕ ХАРАКТЕРИСТИКИ ПЭМС; - включать оценку каждой проблемы с точки зрения связанных с ней РИСКОВ; - определять критерии завершения решения проблем; - определять работы, которые должны выполняться для решения каждой проблемы	5.1.1 План разработки программного обеспечения 9.1 Подготовка ОТЧЕТОВ О ПРОБЛЕМАХ

Продолжение таблицы С.3

Требования к ПЭМС в IEC 60601-1:2005 + IEC 60601-1:2005/AMD1:2012	Требования IEC 62304, связанные с программным обеспечением подсистемы ПЭМС
14.6 ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА	7 ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА программного обеспечения
14.6.1 Идентификация известных и прогнозируемых ОПАСНОСТЕЙ При составлении перечня известных или прогнозируемых ОПАСНОСТЕЙ ИЗГОТОВИТЕЛЬ должен учитывать те из них, которые связаны с программным обеспечением и особенностями аппаратных средств ПЭМС, включая связанные с подключением к ИТ-СЕТЕВЫМ РЕСУРСАМ, компонентами сторонних изготовителей и унаследованными подсистемами	7.1* Анализ программного обеспечения, способствующего опасным ситуациям Настоящий стандарт не ссылается на конкретное сопряжение с сетями и данными
14.6.2 УПРАВЛЕНИЕ РИСКОМ Для реализации каждой меры ПО УПРАВЛЕНИЮ РИСКОМ должны быть выбраны и идентифицированы надлежащим образом проверенные инструменты и ПРОЦЕДУРЫ. Эти инструменты и ПРОЦЕДУРЫ должны быть подходящими для обеспечения того, что каждая мера ПО УПРАВЛЕНИЮ РИСКОМ результативно снизит идентифицированный(е) РИСК(И)	5.1.4 Стандарты, методы и инструменты планирования разработки программного обеспечения Настоящий стандарт требует идентификации определенных инструментов и методов, которые используются как общепринятые при разработке, но не в отношении каждой меры ПО УПРАВЛЕНИЮ РИСКОМ
14.7 Перечень требований Для любой ПЭМС и каждой из ее подсистем должен быть разработан и задокументирован перечень требований	5.2 Анализ требований к программному обеспечению Настоящий стандарт применим только в отношении подсистем программного обеспечения ПЭМС
Перечень требований к системе или подсистеме должен включать и характеризовать все ОСНОВНЫЕ ФУНКЦИОНАЛЬНЫЕ ХАРАКТЕРИСТИКИ и все мероприятия по УПРАВЛЕНИЮ РИСКОМ, реализуемые в системе или подсистеме	5.2.1 Отделение и документирование требований к программному обеспечению от требований СИСТЕМЫ 5.2.2 Содержание требований к программному обеспечению 5.2.3 Включение мер УПРАВЛЕНИЯ РИСКОМ в требования к программному обеспечению Настоящий стандарт устанавливает, что требования, связанные с основными функциональными характеристиками и мерами по УПРАВЛЕНИЮ РИСКОМ, должны отличаться от других требований, но требует однозначной идентификации любых требований
14.8 АРХИТЕКТУРА Для ПЭМС и каждой из ее подсистем должна быть установлена АРХИТЕКТУРА, удовлетворяющая перечню требований	5.3 Проектирование АРХИТЕКТУРЫ программного обеспечения
Когда это целесообразно для снижения РИСКА до допустимого уровня, в спецификации требований к АРХИТЕКТУРЕ должны использоваться: a) КОМПОНЕНТЫ С ВЫСОКОЙ СТЕПЕНЬЮ ИНТЕГРАЦИИ; b) устойчивые к отказам функции; c) избыточность; d) диверсификация; e) разделение функций; f) защищенная конструкция, служащая, например, для ограничения представляющих потенциальную опасность эффектов путем ограничения допускаемой выходной мощности или введения устройств, ограничивающих свободный ход исполнительных устройств.	5.3.5 Идентификация обособленности, необходимой для УПРАВЛЕНИЯ РИСКОМ Разделение является единственным идентифицированным способом, и это только идентификация, потому что требование состоит в точном определении того, что целостность разделения обеспечена

Продолжение таблицы С.3

Требования к ПЭМС в IEC 60601-1:2005 + IEC 60601-1:2005/AMD1:2012	Требования IEC 62304, связанные с программным обеспечением подсистемы ПЭМС
Спецификация АРХИТЕКТУРЫ должна также учитывать: a) распределение мер по УПРАВЛЕНИЮ РИСКОМ в подсистемах и компонентах ПЭМС; b) виды отказов компонентов и их последствия; c) неспецифические отказы; d) систематические отказы; e) период проведения тестирования или диагностики; f) ремонтопригодность; g) защиту от прогнозируемых ошибок в применении; h) если применимо, требования к ИТ-СЕТЕВЫМ РЕСУРСАМ	Это не включено в настоящий стандарт
14.9 Проектирование и реализация Когда это целесообразно, проектирование должно проводиться для отдельных подсистем, каждая из которых должна иметь собственные требования к разработке и требования к испытаниям	5.4 Разработка детального дизайна программного обеспечения 5.4.2 Разработка детального дизайна для каждого ПРОГРАММНОГО БЛОКА Настоящий стандарт не требует спецификации испытаний для детализированного проекта
Пояснения относительно условий проектирования должны включаться в документацию	5.4.2 Разработка детального дизайна для каждого ПРОГРАММНОГО БЛОКА
14.10 ВЕРИФИКАЦИЯ ВЕРИФИКАЦИЯ требуется для всех функций, которые обеспечивают ОСНОВНУЮ БЕЗОПАСНОСТЬ, ОСНОВНЫЕ ФУНКЦИОНАЛЬНЫЕ ХАРАКТЕРИСТИКИ или меры по УПРАВЛЕНИЮ РИСКОМ	5.1.6 Планирование ВЕРИФИКАЦИИ программного обеспечения ВЕРИФИКАЦИЯ требуется в отношении любой ДЕЯТЕЛЬНОСТИ
План ВЕРИФИКАЦИИ должен формироваться для указания способов проверки этих функций и включать: - указания о том, на каком этапе (этапах) каждая функция должна проходить ВЕРИФИКАЦИЮ; - выбор и документирование принципов, мероприятий, методов и соответствующего уровня независимости персонала, выполняющего ВЕРИФИКАЦИЮ; - выбор и использование методов ВЕРИФИКАЦИИ; критерии ВЕРИФИКАЦИИ	5.1.6 Планирование ВЕРИФИКАЦИИ программного обеспечения Требование в отношении независимости персонала не включено в настоящий стандарт. Требование считается установленным в ISO 13485
ВЕРИФИКАЦИЯ должна выполняться в соответствии с планом ВЕРИФИКАЦИИ. Результаты ВЕРИФИКАЦИИ ДЕЯТЕЛЬНОСТИ должны документироваться	Требования по проведению ВЕРИФИКАЦИИ установлено к большинству видов ДЕЯТЕЛЬНОСТИ
14.11 ВАЛИДАЦИЯ ПЭМС План ВАЛИДАЦИИ ПЭМС должен включать валидацию ОСНОВНОЙ БЕЗОПАСНОСТИ и ОСНОВНЫХ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК	Настоящий стандарт не распространяется на валидацию программного обеспечения. Валидация ПЭМС является ДЕЯТЕЛЬНОСТЬЮ на уровне СИСТЕМЫ и находится вне области применения настоящего стандарта
Метод проведения ВАЛИДАЦИИ ПЭМС должен быть задокументирован	Настоящий стандарт не распространяется на валидацию программного обеспечения. Валидация ПЭМС является ДЕЯТЕЛЬНОСТЬЮ на уровне СИСТЕМЫ и находится вне области применения настоящего стандарта
ВАЛИДАЦИЯ ПЭМС должна выполняться в соответствии с планом ВАЛИДАЦИИ ПЭМС. Результаты ДЕЯТЕЛЬНОСТИ по ВАЛИДАЦИИ ПЭМС должны быть задокументированы	Настоящий стандарт не распространяется на валидацию программного обеспечения. Валидация ПЭМС является ДЕЯТЕЛЬНОСТЬЮ на уровне СИСТЕМЫ и находится вне области применения настоящего стандарта

Продолжение таблицы С.3

Требования к ПЭМС в IEC 60601-1:2005 + IEC 60601-1:2005/AMD1:2012	Требования IEC 62304, связанные с программным обеспечением подсистемы ПЭМС
Лицо, несущее основную ответственность за ВАЛИДАЦИЮ ПЭМС, должно быть независимым от коллектива разработчиков ПЭМС. ИЗГОТОВИТЕЛЬ должен задокументировать обоснование уровня его независимости	Настоящий стандарт не распространяется на валидацию программного обеспечения. Валидация ПЭМС является ДЕЯТЕЛЬНОСТЬЮ на уровне СИСТЕМЫ и находится вне области применения настоящего стандарта
Никакой член коллектива разработчиков ПЭМС не должен нести ответственность за процесс ВАЛИДАЦИИ ПЭМС их собственного проекта	Настоящий стандарт не распространяется на валидацию программного обеспечения. Валидация ПЭМС является ДЕЯТЕЛЬНОСТЬЮ на уровне СИСТЕМЫ и находится вне области применения настоящего стандарта
Все профессиональные взаимодействия между членами коллектива, выполняющего работы по ВАЛИДАЦИИ ПЭМС, и членами коллектива разработчиков ПЭМС должны регистрироваться в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА	Настоящий стандарт не распространяется на валидацию программного обеспечения. Валидация ПЭМС является ДЕЯТЕЛЬНОСТЬЮ на уровне СИСТЕМЫ и находится вне области применения настоящего стандарта
Ссылка на методы и РЕЗУЛЬТАТЫ ПРОВЕРКИ СООТВЕТСТВИЯ (ВАЛИДАЦИИ) ПЭМС должна включаться в ФАЙЛ МЕНЕДЖМЕНТА РИСКА	Настоящий стандарт не распространяется на валидацию программного обеспечения. Валидация ПЭМС является ДЕЯТЕЛЬНОСТЬЮ на уровне СИСТЕМЫ и находится вне области применения настоящего стандарта
14.12 Модификация Если часть или весь существующий проект является модификацией более раннего проекта, то к нему следует либо применять требования всего настоящего пункта так, как если бы эта модификация была новым проектом, либо с помощью задокументированной ПРОЦЕДУРЫ модификации в процессе внесения изменений ОЦЕНИВАТЬ возможность дальнейшего использования предыдущей проектной документации	6 Техническая поддержка программного обеспечения Настоящий стандарт устанавливает, что техническая поддержка программного обеспечения должна быть запланирована, а реализация модификаций должна использовать ПРОЦЕСС разработки программного обеспечения или установленный ПРОЦЕСС технической поддержки программного обеспечения
Когда программное обеспечение модифицируется, требования подраздела 4.3, раздела 5, раздела 7, раздела 8 и раздела 9 стандарта IEC 62304:2006 также применяются к модификации	
14.13 ПЭМС, предназначенные для подключения к ИТ-СЕТЕВЫМ РЕСУРСАМ Если ПЭМС предназначена для соединения с помощью ИТ-СЕТЕВЫХ РЕСУРСОВ с другим изделием, которое не может контролироваться ИЗГОТОВИТЕЛЕМ ПЭМС, то в техническом описании указывают: a) цель подключения ПЭМС к ИТ-СЕТЕВЫМ РЕСУРСАМ; b) требуемые характеристики ИТ-СЕТЕВЫХ РЕСУРСОВ, включающей ПЭМС; c) требуемая конфигурация ИТ-СЕТЕВЫХ РЕСУРСОВ, включающей ПЭМС; d) технические характеристики сетевого подключения ПЭМС, включая спецификации безопасности; e) предполагаемый поток информации между ПЭМС, ИТ-СЕТЕВЫМИ РЕСУРСАМИ и другими устройствами, а также предполагаемая маршрутизация.	Требования в отношении подключения к ИТ-СЕТЕВЫМ РЕСУРСАМ не включены в настоящий стандарт

Окончание таблицы С.3

Требования к ПЭМС в IEC 60601-1:2005 + IEC 60601-1:2005/AMD1:2012	Требования IEC 62304, связанные с программным обеспечением подсистемы ПЭМС
Примечание 1 — Это может включать аспекты результативности и безопасности данных и систем, связанные с ОСНОВНОЙ БЕЗОПАСНОСТЬЮ и ОСНОВНЫМИ ФУНКЦИОНАЛЬНЫМИ ХАРАКТЕРИСТИКАМИ (см. также раздел Н. 6 и IEC 80001-1:2010); f) перечень ОПАСНЫХ СИТУАЦИЙ, возникающих в результате неспособности ИТ-СЕТЕВЫХ РЕСУРСОВ обеспечить характеристики, необходимые для выполнения цели подключения PEMS к ИТ-СЕТЕВЫМ РЕСУРСАМ; В СОПРОВОДИТЕЛЬНОЙ ДОКУМЕНТАЦИИ ИЗГОТОВИТЕЛЬ должен проинструктировать ОТВЕТСТВЕННУЮ ОРГАНИЗАЦИЮ о том, что: - соединение ПЭМС с ИТ-СЕТЕВЫМИ РЕСУРСАМИ, которое производится с использованием другого оборудования, может приводить к ранее непредусмотренным РИСКАМ для ПАЦИЕНТОВ, ОПЕРАТОРОВ или третьих лиц; - ОТВЕТСТВЕННАЯ ОРГАНИЗАЦИЯ должна идентифицировать, анализировать, оценивать эти РИСКИ и управлять ими. Примечание 3 — IEC 80001-1:2010 содержит рекомендации для ОТВЕТСТВЕННОЙ ОРГАНИЗАЦИИ по обращению с такими рисками; - последующие изменения ИТ-СЕТЕВЫХ РЕСУРСОВ могут приводить к появлению новых РИСКОВ и требовать дополнительного анализа; - последующие изменения ИТ-СЕТЕВЫХ РЕСУРСОВ могут включать: изменения в их конфигурации; подсоединение к ним дополнительных элементов; отсоединение от них отдельных элементов; модификацию соединенного с ними оборудования; модернизацию соединенного с ними оборудования	

С.4.7 Взаимосвязь с IEC 60601-1-4

IEC 60601-1-4 был отменен.

С.5 — Взаимосвязь с IEC 61010-1

Область применения IEC 61010-1 [5] распространяется на измерительное оборудование и оборудование для электрических испытаний, оборудование электрического контроля и электрооборудование лаборатории. Только часть лабораторного электрооборудования используется в здравоохранении или в качестве изделий для *in vitro* диагностики.

В соответствии с правовым регулированием или нормативными ссылками изделия для диагностики *in vitro* относятся к МЕДИЦИНСКИМ ИЗДЕЛИЯМ, при этом не подпадая под область применения IEC 60601-1 [1]. Это связано не только с тем, что изделия для *in vitro* диагностики не вступают в прямой контакт с пациентами, как обычные МЕДИЦИНСКИЕ ИЗДЕЛИЯ, но и с тем, что эти изделия производятся для различных применений в различных лабораториях. Использование в качестве инструментов для *in vitro* диагностики или принадлежностей к изделиям для *in vitro* диагностики встречается редко.

Если лабораторное оборудование используется в качестве изделия для *in vitro* диагностики, то полученные результаты измерений должны быть ОЦЕНЕНЫ в соответствии с медицинскими критериями. Применение ISO 14971 требуется для осуществления МЕНЕДЖМЕНТА РИСКА. Если подобная продукция содержит программное обеспечение, способное привести к ОПАСНОЙ СИТУАЦИИ, например к нежелательному изменению медицинских данных (результатов измерений) вследствие отказа, вызванного ПО, то должны учитываться требования IEC 62304.

IEC 61010-1:2010 содержит общее требование к оценке рисков в разделе 17, которое является более упрощенным, чем полные требования ISO 14971 по менеджменту риска. Применение стандарта IEC 61010-1, раздела 17 само по себе не соответствует требуемым критериям менеджмента риска IEC 62304, который основан на полных требованиях ISO 14971 по менеджменту риска. Исходя из этого ожидается, что если *in vitro* медицинское изделие имеет связанные с ПО риски, то процесс менеджмента риска выполняется в соответствии с ISO 14971, а не только разделом 17 IEC 61010-1. Соответствие разделу 17 стандарта IEC 61010-1 будет достигнуто, как подробно описано в примечании к разделу 17 стандарта IEC 61010-1.

Примечание — Одна процедура оценки РИСКА описана в Приложении J. Другие процедуры оценки РИСКОВ содержатся в ISO 14971, SEMI S10-1296, IEC 61508, ISO 14121-1 и ANSI B11.TR3. Также могут использоваться другие установленные процедуры, которые реализуют аналогичные шаги.

Блок-схема на рисунке С.3 показывает применение IEC 62304 с IEC 61010-1, раздел 17.

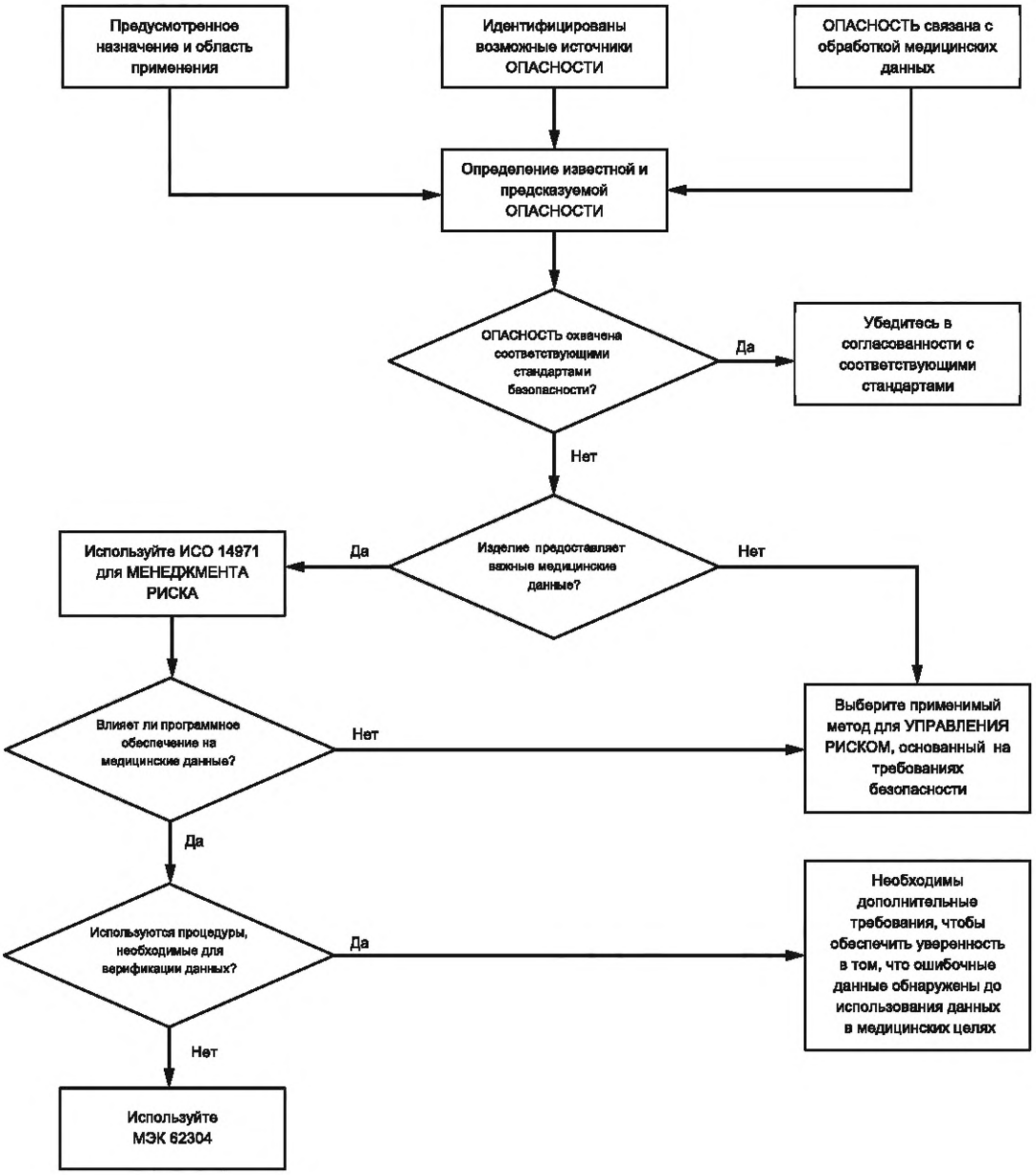


Рисунок С.3 — Применение IEC 62304 с IEC 61010-1

С.6 Взаимосвязь с ISO/IEC 12207

Настоящий стандарт был производным от подходов и концепций ISO/IEC 12207 [9], определяющим требования для ПРОЦЕССОВ жизненного цикла ПО в общих чертах, то есть не ограничиваясь МЕДИЦИНСКИМИ ИЗДЕЛИЯМИ.

Стандарт отличается от ISO/IEC 12207 главным образом в отношении того, что он:

- исключает аспекты СИСТЕМЫ, такие как СИСТЕМНЫЕ требования, АРХИТЕКТУРА СИСТЕМЫ и валидация;
- пропускает некоторые ПРОЦЕССЫ, рассматриваемые как дублирующая деятельность, представленные в различных изданиях для МЕДИЦИНСКИХ ИЗДЕЛИЙ;
- добавляет ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА (БЕЗОПАСНОСТЬ) и ПРОЦЕСС выпуска программного обеспечения;
- включает документирование и ВЕРИФИКАЦИЮ поддерживающих ПРОЦЕССОВ в ПРОЦЕССЫ разработки и технической поддержки;
- объединяет реализацию ПРОЦЕССОВ и планирование ДЕЯТЕЛЬНОСТИ по каждому ПРОЦЕССУ в единую ДЕЯТЕЛЬНОСТЬ по ПРОЦЕССАМ разработки и технического обслуживания;
- классифицирует требования с учетом БЕЗОПАСНОСТИ;
- не классифицирует ПРОЦЕССЫ как первостепенные или поддерживающие и не группирует ПРОЦЕССЫ, как это сделано в ISO/IEC 12207.

Большинство отличий были реализованы исходя из нужд и потребностей промышленности МЕДИЦИНСКИХ ИЗДЕЛИЙ:

- фокусироваться на аспектах безопасности и МЕНЕДЖМЕНТА РИСКА МЕДИЦИНСКИХ ИЗДЕЛИЙ, установленных в ISO 14971;
- выбрать подходящие ПРОЦЕССЫ, полезные в регулируемой внешней среде;
- принять во внимание, что разработка программного обеспечения включена в систему качества (которая охватывает некоторые ПРОЦЕССЫ и требования ISO/IEC 12207);
- уменьшить уровень обобщения, чтобы облегчить применение.

Настоящий стандарт не противоречит ISO/IEC 12207. ISO/IEC 12207 может быть полезным в качестве вспомогательной информации для создания правильно структурированной МОДЕЛИ ЖИЗНЕННОГО ЦИКЛА РАЗРАБОТКИ программного обеспечения, которая включает требования настоящего стандарта.

Таблица С.5, которая была подготовлена подкомитетом 7 ISO/IEC, показывает взаимосвязь между настоящим стандартом и ISO/IEC 12207.

Таблица С.5 — Взаимосвязь с процессами ISO/IEC 12207:2008

Процессы настоящего стандарта		Процессы ISO/IEC 12207:2008	
Деятельность	Задача	Процессы	Деятельность/Задача
5 ПРОЦЕСС разработки программного обеспечения			
5.1 Планирование разработки программного обеспечения	5.1.План разработки программного обеспечения	7.1.1 Реализация программного обеспечения	7.1.1.3.1 Стратегия реализации программного обеспечения 7.1.1.3.1.1 7.1.1.3.1.3 7.1.1.3.1.4 6.3.1.3.2 Планирование проекта 6.3.1.3.2.1
	5.1.2 Поддержание плана разработки программного обеспечения в актуальном состоянии	6.3.2 Оценка проекта и процесс управления	6.3.2.3.2 Управление проектом 6.3.2.3.2.1
	5.1.3 План разработки программного обеспечения относительно проектирования и разработки СИСТЕМЫ	6.4.3 Процесс проектирования архитектуры системы 6.4.5 Системная интеграция 7.2.5 Процесс валидации программного обеспечения	6.4.3.3.1 Установление архитектуры 6.4.3.3.1.1 6.4.5.3.1 Интеграция 6.4.5.3.1.1 7.2.5.3.1 Процесс реализации 7.2.5.3.1.4

Продолжение таблицы С.5

Процессы настоящего стандарта		Процессы ISO/IEC 12207:2008	
Деятельность	Задача	Процессы	Деятельность/Задача
5.1 Планирование разработки программного обеспечения	5.1.4 Стандарты, методы и инструменты планирования разработки программного обеспечения	7.1.1 Реализация программного обеспечения	7.1.1.3.1 Стратегия реализации программного обеспечения 7.1.1.3.1.3
	5.1.5 Программная интеграция и планирование тестирования интеграции	7.1.6 Интеграция программного обеспечения	7.1.6.3.1 Интеграция программного обеспечения 7.1.6.3.1.1
	5.1.6 Планирование ВЕРИФИКАЦИИ программного обеспечения	7.2.4 Верификация программного обеспечения 7.1.5 Процесс конструирования программных средств 7.1.6 Software Integration 7.1.7 Квалификационное тестирование программного обеспечения	7.2.4.3.1 Процесс реализации 7.2.4.3.1.4 7.2.4.3.1.5 7.1.5.3.1 Процесс конструирования программных средств 7.1.5.3.1.5 7.1.6.3.1 Интеграция программного обеспечения 7.1.6.3.1.5 7.1.7.3.1 Квалификационное тестирование программного обеспечения 7.1.7.3.1.3
	5.1.7 Планирование МЕНЕДЖМЕНТА РИСКА программного обеспечения	6.3.4 Процесс менеджмента риска	
	5.1.8 Документация по планированию	7.2.1 Менеджмент документации программного обеспечения	7.2.1.3.1 Процесс реализации 7.2.1.3.1.1
	5.1.9 Планирование менеджмента конфигурации программного обеспечения	7.2.2 Менеджмент конфигурации программного обеспечения 7.2.8 Процесс решения проблем в программном обеспечении	7.2.2.3.1 Процесс реализации 7.2.2.3.1.1 7.2.8.3.1 Процесс реализации 7.2.8.3.1.1
	5.1.10 Поддержка элементов, подлежащих управлению	6.2.2 Менеджмент инфраструктуры	6.2.2.3.2 Установление инфраструктуры 6.2.2.3.2.1 6.2.2.3.3 Поддержание инфраструктуры 6.2.2.3.3.1
	5.1.11 Управление СОСТАВНЫМИ ЧАСТЯМИ КОНФИГУРАЦИИ программного обеспечения до ВЕРИФИКАЦИИ	7.2.2 Менеджмент конфигурации программного обеспечения	7.2.2.3.2 Идентификация конфигурации 7.2.2.3.2.1
5.2 Анализ требований к программному обеспечению	5.2.1 Отделение и документирование требований к программному обеспечению на основе требований СИСТЕМЫ	6.4.3 Проектирование архитектуры системы	6.4.3.3.1 Установление архитектуры 6.4.3.3.1.1

Продолжение таблицы С.5

Процессы настоящего стандарта		Процессы ISO/IEC 12207:2008	
Деятельность	Задача	Процессы	Деятельность/Задача
5.2 Анализ требований к программному обеспечению	5.2.2 Содержание требований к программному обеспечению	7.1.2 Анализ требований к программному обеспечению	7.1.2.3.1 Анализ требований к программному обеспечению 7.1.2.3.1.1
	5.2.3 Включение мер по УПРАВЛЕНИЮ РИСКОМ в требования к программному обеспечению		
	5.2.4 ПЕРЕОЦЕНИВАНИЕ АНАЛИЗА РИСКА МЕДИЦИНСКОГО ИЗДЕЛИЯ	Нет	Нет
	5.2.5 Обновление требований к СИСТЕМЕ	7.1.2 Анализ требований к программному обеспечению	7.1.2.3.1 Анализ требований к программному обеспечению 7.1.2.3.1.1 a) и b)
	5.2.6 Верификация требований к программному обеспечению	7.2.4 Верификация программного обеспечения	7.2.4.3.2 Верификация 7.2.4.3.2.1
5.3 Проектирование АРХИТЕКТУРЫ программного обеспечения	5.3.1 Преобразование требований к программному обеспечению в АРХИТЕКТУРУ	7.1.3 Проектирование архитектуры программного обеспечения	7.1.3.3.1 Проектирование архитектуры программного обеспечения 7.1.3.3.1.1
	5.3.2 Разработка АРХИТЕКТУРЫ для интерфейсов ПРОГРАММНЫХ СОСТАВНЫХ ЧАСТЕЙ		7.1.3.3.1 Проектирование архитектуры программного обеспечения 7.1.3.3.1.2
	5.3.3 Определение требований к функциональным и эксплуатационным характеристикам элементов ПОНП	Нет	Нет
	5.3.4 Определение требований к аппаратным и программным средствам СИСТЕМЫ, требуемых элементами ПОНП	Нет	Нет
	5.3.5 Идентификация обособленности, необходимой для УПРАВЛЕНИЯ РИСКОМ	Нет	Нет
	5.3.6 Верификация АРХИТЕКТУРЫ программного обеспечения	7.1.3 Проектирование архитектуры программного обеспечения	7.1.3.3.1 Проектирование архитектуры программного обеспечения 7.1.3.3.1.6

Продолжение таблицы С.5

Процессы настоящего стандарта		Процессы ISO/IEC 12207:2008	
Деятельность	Задача	Процессы	Деятельность/Задача
5.4 Разработка детального дизайна программного обеспечения	5.4.1 Дробление программного обеспечения на ПРОГРАММНЫЕ БЛОКИ	7.1.4 Детализированная разработка программного обеспечения	7.1.4.3.1 Детализированная разработка программного обеспечения 7.1.4.3.1.1
	5.4.2 Разработка детального дизайна для каждого ПРОГРАММНОГО БЛОКА		
	5.4.3 Разработка детального дизайна для интерфейсов		7.1.4.3.1 Детализированная разработка программного обеспечения 7.1.4.3.1.2
	5.4.4 ВЕРИФИКАЦИЯ детального дизайна	7.1.4 Детализированная разработка программного обеспечения	7.1.4.3.1 Детализированная разработка программного обеспечения 7.1.4.3.1.7
5.5* Имплементация ПРОГРАММНЫХ БЛОКОВ	5.5.1 Имплементация каждого ПРОГРАММНОГО БЛОКА	7.1.5 Конструирование программного обеспечения	7.1.5.3.1 Конструирование программного обеспечения 7.1.5.3.1.1
	5.5.2 Установление ПРОЦЕССА ВЕРИФИКАЦИИ ПРОГРАММНОГО БЛОКА	7.1.4 Детализированная разработка программного обеспечения 7.1.5 Конструирование программного обеспечения	7.1.4.3.1 Детализированная разработка программного обеспечения 7.1.4.3.1.5 7.1.5.3.1 Конструирование программного обеспечения 7.1.5.3.1.5
	5.5.3 Критерии приемки ПРОГРАММНЫХ БЛОКОВ	7.1.5 Конструирование программного обеспечения	7.1.5.3.1 Конструирование программного обеспечения 7.1.5.3.1.5
	5.5.4 Дополнительные критерии приемки ПРОГРАММНЫХ БЛОКОВ	7.1.5 Конструирование программного обеспечения 7.2.4 Верификация программного обеспечения	7.1.5.3.1 Конструирование программного обеспечения 7.1.5.3.1.2
	5.5.5. ВЕРИФИКАЦИЯ ПРОГРАММНЫХ БЛОКОВ	7.1.5 Конструирование программного обеспечения	7.1.5.3.1 Конструирование программного обеспечения 7.1.5.3.1.2
5.6 Интеграция программного обеспечения и тестирование интеграции	5.6.1 Интеграция ПРОГРАММНЫХ БЛОКОВ	7.1.6 Интеграция программного обеспечения	7.1.6.3.1 Интеграция программного обеспечения 7.1.6.3.1.2
	5.6.2 Верификация интеграции программного обеспечения	7.1.6 Интеграция программного обеспечения 6.4.5 Системная интеграция	7.1.6.3.1 Интеграция программного обеспечения 7.1.6.3.1.2 6.4.5.3.1 Интеграция 6.4.5.3.1.2
	5.6.3 Интеграционное тестирование программного обеспечения	7.1.7 Квалификационное тестирование программного обеспечения	7.1.7.3.1 Квалификационное тестирование программного обеспечения 7.1.7.3.1.1

Продолжение таблицы С.5

Процессы настоящего стандарта		Процессы ISO/IEC 12207:2008	
Деятельность	Задача	Процессы	Деятельность/Задача
5.6 Интеграция программного обеспечения и тестирование интеграции	5.6.4 Содержание тестирования интеграции программного обеспечения	7.1.7 Квалификационное тестирование программного обеспечения	7.1.7.3.1 Квалификационное тестирование программного обеспечения 7.1.7.3.1.3
	5.6.5 Оценивание процедур тестирования интеграции программного обеспечения	Нет	Нет
	5.6.6 Проведение регрессионного тестирования	7.1.6 Интеграция программного обеспечения	7.1.6.3.1 Интеграция программного обеспечения 7.1.6.3.1.2
	5.6.7 Содержание записей в отношении регрессионного тестирования	7.1.6 Интеграция программного обеспечения	7.1.6.3.1 Интеграция программного обеспечения 7.1.6.3.1.2
	5.6.8 Использование ПРОЦЕССА решения проблем с программным обеспечением	7.2.4 Верификация программного обеспечения	7.2.4.3.1 Реализация процесса 7.2.4.3.1.6
5.7 Тестирование ПРОГРАММНОЙ СИСТЕМЫ	5.7.1 Установление тестирования в отношении требований к программному обеспечению	7.1.6 Интеграция программного обеспечения 7.1.7 Квалификационное тестирование программного обеспечения	7.1.6.3.1 Интеграция программного обеспечения 7.1.6.3.1.4 7.1.7.3.1 Квалификационное тестирование программного обеспечения 7.1.7.3.1.1
	5.7.2 Применение ПРОЦЕССА решения проблем с программным обеспечением	7.2.4 Верификация программного обеспечения	7.2.4.3.1 Реализация процесса 7.2.4.3.1.6
	5.7.3 Повторное тестирование после внесения изменений	7.2.8 Процесс решения проблем в программном обеспечении	7.2.4.3.1 Реализация процесса 7.2.4.3.1.1
	5.7.4 Оценивание тестирования ПРОГРАММНОЙ СИСТЕМЫ	7.1.7 Квалификационное тестирование программного обеспечения	7.1.7.3.1 Квалификационное тестирование программного обеспечения 7.1.7.3.1.3
	5.7.5 Содержание отчета по тестированию ПРОГРАММНОЙ СИСТЕМЫ	7.1.7 Квалификационное тестирование программного обеспечения	7.1.7.3.1 Квалификационное тестирование программного обеспечения 7.1.7.3.1.1
5.8 Выпуск программного обеспечения на системном уровне	5.8.1 Обеспечение завершенности ВЕРИФИКАЦИИ программного обеспечения	6.4.9 Функционирование программного обеспечения 7.2.2 Менеджмент конфигурации программного обеспечения	6.4.9.3.2 Активация и проверка функционирования 6.4.9.3.2.1 6.4.9.3.2.2 7.2.2.3.6 Поставка и менеджмент выпуска 7.2.2.3.6.1

Продолжение таблицы С.5

Процессы настоящего стандарта		Процессы ISO/IEC 12207:2008	
Деятельность	Задача	Процессы	Деятельность/Задача
5.8 Выпуск программного обеспечения на системном уровне	5.8.2 Документирование известных остаточных АНОМАЛИЙ		
	5.8.3 ОЦЕНИВАНИЕ известных остаточных АНОМАЛИЙ		
	5.8.4 Документирование выпущенных ВЕРСИЙ	7.2.2 Процесс менеджмента конфигурации программного обеспечения	7.2.2.3.6 Поставка и менеджмент выпуска 7.2.2.3.6.1
	5.8.5 Документирование создания выпущенного программного обеспечения		
	5.8.6 Обеспечение полного завершения деятельности и задач		
	5.8.7 Архивирование программного обеспечения		
	5.8.8 Обеспечение надежной поставки выпущенного программного обеспечения		
6 Техническая поддержка программного обеспечения		6.4.10 Процесс технической поддержки программного обеспечения	
6.1 Установление плана технической поддержки программного обеспечения		6.4.10 Техническая поддержка программного обеспечения	Нет
6.2 Анализ модификации и проблем	6.2.1 Документирование и оценивание обратной связи	Нет	Нет
	6.2.1.1 Мониторинг обратной связи	6.4.10 Техническая поддержка программного обеспечения	Нет
	6.2.1.2 Документирование и ОЦЕНИВАНИЕ обратной связи	6.4.10 Техническая поддержка программного обеспечения	Нет
	6.2.1.3 ОЦЕНИВАНИЕ влияния ОТЧЕТОВ О ПРОБЛЕМАХ на БЕЗОПАСНОСТЬ	6.4.10 Техническая поддержка программного обеспечения	Нет
	6.2.2 Использование ПРОЦЕССА решения проблем программного обеспечения	6.4.10 Техническая поддержка программного обеспечения	Нет

Продолжение таблицы С.5

Процессы настоящего стандарта		Процессы ISO/IEC 12207:2008	
Деятельность	Задача	Процессы	Деятельность/Задача
6.2 Анализ модификации и проблем	6.2.3 Анализ ЗАПРОСОВ НА ИЗМЕНЕНИЕ	6.4.10 Техническая поддержка программного обеспечения	Нет
	6.2.4 Одобрение ЗАПРОСА НА ИЗМЕНЕНИЕ	6.4.10 Техническая поддержка программного обеспечения	Нет
	6.2.5 Информирование пользователей и регулирующих органов	6.4.10 Техническая поддержка программного обеспечения	Нет
6.3 Осуществление модификации		Нет	Нет
	6.3.1 Использование установленного ПРОЦЕССА осуществления модификации	6.4.10 Техническая поддержка программного обеспечения	Нет
	6.3.2 Повторный выпуск модифицированной ПРОГРАММНОЙ СИСТЕМЫ	7.2.2 Процесс менеджмента конфигурации программного обеспечения	Нет
7 ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА программного обеспечения		6.3.4 Процесс менеджмента риска Основан на ISO/IEC 16085. Несмотря на некоторую общность, в нем не рассмотрены конкретные требования к разработке программного обеспечения медицинских изделий в отношении менеджмента риска	
8 ПРОЦЕСС менеджмента конфигурации программного обеспечения			
8.1 Идентификация конфигурации	8.1.1 Установление средств идентификации СОСТАВНЫХ ЧАСТЕЙ КОНФИГУРАЦИИ	7.2.2 Процесс менеджмента конфигурации программного обеспечения	Нет
	8.1.2 Идентификация ПОНП	Нет	Нет
	8.1.3 Идентификация документации конфигурации СИСТЕМЫ	7.2.2 Процесс менеджмента конфигурации программного обеспечения	Нет
8.2 Управление изменениями	8.2.1 Одобрение ЗАПРОСОВ НА ИЗМЕНЕНИЯ	7.2.2 Процесс менеджмента конфигурации программного обеспечения	Нет
	8.2.2 Осуществление изменений	6.4.10 Техническая поддержка программного обеспечения	Нет
	8.2.3 ВЕРИФИКАЦИЯ изменений	7.2.2 Процесс менеджмента конфигурации программного обеспечения	Нет
	8.2.4 Обеспечение средствами для ПРОСЛЕЖИВАЕМОСТИ изменений		

Окончание таблицы С.5

Процессы настоящего стандарта		Процессы ISO/IEC 12207:2008	
Деятельность	Задача	Процессы	Деятельность/Задача
8.3 Учет статуса конфигурации		7.2.2 Процесс менеджмента конфигурации программного обеспечения	Нет
9 ПРОЦЕСС решения проблем программного обеспечения			
9.1 Подготовка ОТЧЕТОВ О ПРОБЛЕМАХ		7.2.8 Процесс решения проблем в программном обеспечении	Нет
9.2 Исследование проблемы		7.2.8 Процесс решения проблем в программном обеспечении	Нет
9.3 Консультирование заинтересованных сторон		7.2.8 Процесс решения проблем в программном обеспечении	Нет
9.4 Использование процесса управления изменениями		7.2.2 Процесс менеджмента конфигурации программного обеспечения 6.4.10 Техническая поддержка программного обеспечения	Нет
9.5 Поддержание записей		7.2.8 Процесс решения проблем в программном обеспечении	Нет
9.6 Анализ проблем на предмет выявления тенденций		7.2.8 Процесс решения проблем в программном обеспечении	Нет
9.7 ВЕРИФИКАЦИЯ решения проблем программного обеспечения		7.2.8 Процесс решения проблем в программном обеспечении	Нет
9.8 Содержание документации по тестированию		ISO 12207 требует документирования всех ЗАДАЧ проведения тестирования	Нет

С.7 Взаимосвязь с IEC 61508

В результате рассмотрения вопроса об использовании в настоящем стандарте, применяемом в отношении критически важного для БЕЗОПАСНОСТИ программного обеспечения, принципов IEC 61508, были учтены следующие соображения. Подход к безопасности в IEC 62304 принципиально отличается от подхода в IEC 61508. IEC 62304 учитывает, что результативность медицинских изделий оправдывает существование остаточных рисков, связанных с их применением. Позицию настоящего стандарта объясняет следующее.

IEC 61508 сфокусирован на трех главных вопросах:

1) жизненном цикле МЕНЕДЖМЕНТА РИСКА и ПРОЦЕССАХ жизненного цикла;

2) определении уровней безопасности эксплуатации оборудования;

3) рекомендуемых техниках, инструментах и методах для разработки программного обеспечения и уровнях независимости персонала, ответственного за выполнение различных ЗАДАЧ.

Вопрос 1) включен в настоящий стандарт нормативной ссылкой на ISO 14971 (стандарт по МЕНЕДЖМЕНТУ РИСКА для промышленности МЕДИЦИНСКОИХ ИЗДЕЛИЙ). Влияние этой ссылки состоит в том, чтобы адаптиро-

вать подход ISO 14971 к МЕНЕДЖМЕНТУ РИСКА как составную часть ПРОЦЕССА программного обеспечения для ПО МЕДИЦИНСКИХ ИЗДЕЛИЙ.

Для вопроса 2) настоящий стандарт принимает более простой подход, чем IEC 61508, классифицирующий программное обеспечение на четыре «уровня безопасности эксплуатации оборудования», определенные с точки зрения надежности. Цели надежности идентифицируют после АНАЛИЗА РИСКА, который определяет как тяжесть, так и вероятность причинения ВРЕДА, вызванного отказом ПО.

Настоящий стандарт упрощает вопрос 2) посредством установления классификации по трем классам безопасности программного обеспечения на основе РИСКА, вызванного отказом. После классификации для разных классов безопасности программного обеспечения требуются разные ПРОЦЕССЫ: намерение состоит в дальнейшем уменьшении вероятности (и/или тяжести последствий) отказа программного обеспечения.

Вопрос 3) не затронут в настоящем стандарте. Пользователям рекомендуется применять IEC 61508 в качестве источника программных методов, техник и инструментов, с учетом того, что другие подходы могут обеспечить одинаковые РЕЗУЛЬТАТЫ. Настоящий стандарт не содержит рекомендаций относительно независимости лиц, ответственных за один вид ДЕЯТЕЛЬНОСТИ в области программного обеспечения (например, за ВЕРИФИКАЦИЮ) от тех, кто отвечает за другой (например, за проектирование). В частности, настоящий стандарт не требует наличия независимого эксперта по безопасности, поскольку это относится к ISO 14971.

Приложение D
(справочное)

Применение

D.1 Введение

В данном приложении приводится общий обзор того, как настоящий стандарт может быть реализован в ПРОЦЕССАХ ИЗГОТОВИТЕЛЯ. Предполагается, что другие стандарты, такие как ИСО 13485 [8], требуют подходящих и сопоставимых ПРОЦЕССОВ.

D.2 Система менеджмента качества

В контексте настоящего стандарта для ИЗГОТОВИТЕЛЕЙ МЕДИЦИНСКИХ ИЗДЕЛИЙ, включая ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МЕДИЦИНСКИХ ИЗДЕЛИЙ, установление системы менеджмента качества (далее — СМК) требуется в соответствии с 4.1. Настоящий стандарт не требует, чтобы СМК обязательно была сертифицирована.

D.3 ОЦЕНИВАНИЕ ПРОЦЕССОВ менеджмента качества

Рекомендуется ОЦЕНИВАТЬ, как установленные и документированные ПРОЦЕССЫ СМК охватывают ПРОЦЕССЫ жизненного цикла программного обеспечения посредством проведения аудитов, проверок инспекций или анализа, за которые несет ответственность ИЗГОТОВИТЕЛЬ. Любые выявленные несоответствия могут быть устранены посредством расширения установленных ПРОЦЕССОВ менеджмента качества или оформлены отдельными документами. Если ИЗГОТОВИТЕЛЬ уже имеет документированные ПРОЦЕССЫ, которые регламентируют разработку, ВЕРИФИКАЦИЮ и валидацию ПО, то данные ПРОЦЕССЫ также должны быть ОЦЕНЕНЫ с целью определения согласованности с настоящим стандартом.

D.4 Интеграция требований настоящего стандарта в ПРОЦЕССЫ менеджмента качества ИЗГОТОВИТЕЛЯ

Настоящий стандарт может быть внедрен посредством адаптации или расширения ПРОЦЕССОВ, уже установленных в СМК, или интеграцией новых ПРОЦЕССОВ. Настоящий стандарт не устанавливает какой-либо способ, и ИЗГОТОВИТЕЛЬ может выполнить внедрение любым подходящим образом.

ИЗГОТОВИТЕЛЬ несет ответственность за обеспечение надлежащего выполнения ПРОЦЕССОВ, описанных в настоящем стандарте, когда ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МЕДИЦИНСКОГО ИЗДЕЛИЯ разработано ИЗГОТОВИТЕЛЯМИ оригинального оборудования (OEM) или субподрядчиками, не имеющими собственной документированной СМК.

D.5 Контрольный список для малых предприятий, ИЗГОТОВИТЕЛЕЙ, не имеющих сертифицированной СМК

Изготовитель должен установить самый высокий класс безопасности программного обеспечения (A, B или C). Таблица D.1 перечисляет все виды ДЕЯТЕЛЬНОСТИ, описанные в настоящем стандарте. Ссылка на ИСО 13485 должна помочь определить место данной деятельности в СМК. Основываясь на требуемом классе безопасности программного обеспечения, ИЗГОТОВИТЕЛЮ следует ОЦЕНИТЬ каждый требуемый вид ДЕЯТЕЛЬНОСТИ относительно уже существующих ПРОЦЕССОВ. Если требование уже выполнено, то должны быть сделаны ссылки на соответствующие ПРОЦЕССЫ.

При наличии несоответствий необходимо предпринять действия для улучшения ПРОЦЕССА.

Список также может быть использован для ОЦЕНИВАНИЯ ПРОЦЕССОВ после выполнения действия.

Таблица D.1 — Контрольный список для малых предприятий, не имеющих сертифицированной СМК

ДЕЯТЕЛЬНОСТЬ	Соответствующий раздел ИСО 13485:2003	Охватывается существующими процедурами?	Если да: ссылка	Предпринятые действия
5.1 Планирование разработки ПО	7.3.1 Планирование проектирования и разработки	Да/Нет		
5.2 Анализ требований к ПО	7.3.2 Входные данные для проектирования и разработки	Да/Нет		
5.3 Проектирование АРХИТЕКТУРЫ ПО		Да/Нет		
5.4 Разработка детального дизайна ПО		Да/Нет		

Окончание таблицы D.1

ДЕЯТЕЛЬНОСТЬ	Соответствующий раздел ИСО 13485:2003	Охватывается существующими процедурами?	Если да: ссылка	Предпринятые действия
5.5 Имплементация ПРОГРАММНЫХ БЛОКОВ		Да/Нет		
5.6 Интеграция ПО и тестирование интеграции		Да/Нет		
5.7 Тестирование ПРОГРАММНОЙ СИСТЕМЫ	7.3.3 Выходные данные проектирования и разработки 7.3.4 Анализ проекта и разработки	Да/Нет		
5.8 Выпуск ПО на системном уровне	7.3.5 Верификация проектирования и разработки 7.3.6 Валидация проектирования и разработки	Да/Нет		
6.1 Установление плана технической поддержки ПО	7.3.7 Управление изменениями проектирования и разработки	Да/Нет		
6.2 Анализ модификаций и проблем		Да/Нет		
6.3 Осуществление модификации	7.3.5 Верификация проектирования и разработки 7.3.6 Валидация проектирования и разработки	Да/Нет		
7.1 Анализ ПО, способствующего опасным ситуациям		Да/Нет		
7.2 Меры по УПРАВЛЕНИЮ РИСКОМ		Да/Нет		
7.3 ВЕРИФИКАЦИЯ мер по УПРАВЛЕНИЮ РИСКОМ		Да/Нет		
7.4 МЕНЕДЖМЕНТ РИСКА в отношении изменений ПО		Да/Нет		
8.1 Определение конфигурации	7.5.3 Идентификация и прослеживаемость	Да/Нет		
8.2 Управление изменениями	7.5.3 Идентификация и прослеживаемость	Да/Нет		
8.3 Учет статуса конфигурации		Да/Нет		
9 ПРОЦЕСС решения проблем ПО		Да/Нет		

Приложение ДА
(обязательное)

Сведения о соответствии ссылочных международных стандартов межгосударственным стандартам

Таблица ДА.1

Обозначение ссылочного международного стандарта	Степень соответствия	Обозначение и наименование соответствующего межгосударственного стандарта
ISO 14971	IDT	ГОСТ ISO 14971—2021 «Изделия медицинские. Применение менеджмента риска к медицинским изделиям»

Примечание — В настоящей таблице использовано следующее условное обозначение степени соответствия стандарта:

- IDT — идентичный стандарт.

Библиография

[1] IEC 60601-1:2005
IEC 60601-1:2005/AMD1:2012
Medical electrical equipment — Part 1: General requirements for basic safety and essential performance (Изделия медицинские электрические. Часть 1. Общие требования безопасности и основные характеристики)

[2] IEC 60601-1-4:1996
IEC 60601-1-4:1996/AMD1:1999
Medical electrical equipment — Part 1: General requirements for safety — 4.Collateral standard: Programmable electrical medical systems (withdrawn) (Изделия медицинские электрические. Часть 1-1. Общие требования безопасности. Дополнительный стандарт. Требования безопасности к медицинским электрическим системам)

[3] IEC 60601-1-6
Medical electrical equipment — Part 1-6: General requirements for basic safety and essential performance — Collateral standard: Usability (Изделия медицинские электрические. Часть 1-6. Общие требования безопасности с учетом основных функциональных характеристик. Дополнительный стандарт. Эксплуатационная пригодность)

[4] IEC 61508-3
Functional safety of electrical/electronic/programmable electronic safetyrelated systems — Part 3: Software requirements (Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. Часть 3. Требования к программному обеспечению)

[5] IEC 61010-1:2010
Safety requirements for electrical equipment for measurement, control, and laboratory use — Part 1: General requirements (Безопасность контрольно-измерительных приборов и лабораторного оборудования. Часть 1. Общие требования)

[6] ISO 9000:2005
Quality management systems — Fundamentals and vocabulary (Системы менеджмента качества. Основные положения и словарь)

[7] ISO 9001:2008
Quality management systems — Requirements (Системы менеджмента качества. Требования)

[8] ISO 13485:2003
Medical devices — Quality management systems — Requirements for regulatory purposes (Изделия медицинские. Системы менеджмента качества. Требования для целей регулирования)

[9] ISO/IEC 12207:2008
Systems and software engineering — Software life cycle processes (Системная и программная инженерия. Процессы жизненного цикла программных средств)

[10] ISO/IEC 14764:1999
Software Engineering — Software Life Cycle Processes — Maintenance (Информационная технология. Сопровождение программных средств)

[11] ISO/IEC 15504-5:2012
Information technology — Process assessment — Part 5: An exemplar software life cycle process assessment model (Информационные технологии. Оценка процессов. Часть 5. Пример модели оценки процесса)

[12] ISO/IEC 25010:2011
Systems and software engineering — System and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models (Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов)

[13] ISO/IEC 33001:2015
Information technology — Process assessment — Concepts and terminology (Информационные технологии. Оценка процесса. Понятия и терминология)

[14] ISO/IEC 33004:2015
Information technology — Process assessment — Requirements for process reference, process assessment and maturity modelss (Информационная технология. Оценка процесса. Требования к эталонным моделям процесса, моделям оценки процесса и завершенным моделям)

[15] ISO/IEC 90003:2014
Software engineering — Guidelines for the application of ISO 9001:2008 to computer software (Разработка программных продуктов. Руководящие указания по применению ИСО 9001:2008 при разработке программных продуктов)

[16] ISO/IEC Guide 51:2014 Safety aspects — Guidelines for their inclusion in standards (Аспекты безопасности. Руководящие указания по включению их в стандарты)

[17] IEEE 610.12:1990 IEEE standard glossary of software engineering terminology

[18] IEEE 1044:2009 IEEE standard classification for software anomalies

[19] U.S. Department Of Health and Human Services, Food and Drug Administration, Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices, May 11, 2005, <://www.fda.gov/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/ucm089543.htm>

[20] U.S. Department Of Health and Human Services, Food and Drug Administration, General Principles of Software Validation; Final Guidance for Industry and FDA Staff, January 11, 2002, <http://www.fda.gov/downloads/RegulatoryInformation/Guidances/ucm126955.pdf>

[21] IEC 62366-1:2015 Medical devices — Part 1: Application of usability engineering to medical devices (Изделия медицинские. Часть 1. Проектирование медицинских изделий с учетом эксплуатационной пригодности)

[22] 82304-1:—[3] Healthcare Software Systems — Part 1: General requirements (Изделия медицинские. Часть 1. Проектирование медицинских изделий с учетом эксплуатационной пригодности)

[3] В стадии разработки.

УДК 658.562.014:006.354 МКС 11.040.01

Ключевые слова: программное обеспечение, жизненный цикл, система менеджмента качества, менеджмент риска, изготовитель, медицинское изделие

Редактор *З.А. Лиманская*
Технический редактор *В.Н. Прусакова*
Корректор *О.В. Лазарева*
Компьютерная верстка *Е.А. Кондрашовой*

Сдано в набор 31.10.2022. Подписано в печать 23.11.2022. Формат 60×84⅛. Гарнитура Ариал.
Усл. печ. л. 8,37. Уч.-изд. л. 7,58.

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

Создано в единичном исполнении в ФГБУ «РСТ»
для комплектования Федерального информационного фонда стандартов,
117418 Москва, Нахимовский пр-т, д. 31, к. 2.
www.gostinfo.ru info@gostinfo.ru