ГОСТ Р МЭК 62304-2013

Группа Р20

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ
ИЗДЕЛИЯ МЕДИЦИНСКИЕ

Программное обеспечение. Процессы жизненного цикла

Medical devices. Software. Life cycle processes

ОКС 11.040

Дата введения 2015-01-01

Предисловие

1 ПОДГОТОВЛЕН Закрытым акционерным обществом "МЕДИТЕСТ" на основе собственного аутентичного перевода на русский язык международного стандарта, указанного в пункте 4

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 436 "Управление качеством медицинских изделий"

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 8 ноября 2013 г. N 1492-ст

4 Настоящий стандарт идентичен международному стандарту МЭК 62304:2006* "Программное обеспечение медицинских изделий. Процессы жизненного цикла программного обеспечения" (IEC 62304:2006 "Medical device software - Software life cycle processes").

---

* Доступ к международным и зарубежным документам, упомянутым в тексте, можно получить, обратившись в Службу поддержки пользователей. - Примечание изготовителя базы данных.

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

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

5 ВВЕДЕН ВПЕРВЫЕ

Правила применения настоящего стандарта установлены в ГОСТ Р 1.0-2012 *(раздел 8). Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе "Национальные стандарты", а официальный текст изменений и поправок - в ежемесячном информационном указателе "Национальные стандарты". В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя "Национальные стандарты". Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования - на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (gost.ru)*

Введение

Программные средства часто являются неотъемлемой частью технологии МЕДИЦИНСКИХ ИЗДЕЛИЙ. Создание БЕЗОПАСНОГО и результативного МЕДИЦИНСКОГО ИЗДЕЛИЯ, содержащего ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, требует знания о том, для чего ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ предназначено и доказательств того, что использование ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ выполняет свое назначение, не создавая недопустимых РИСКОВ.

Настоящий стандарт определяет основу ПРОЦЕССОВ жизненного цикла совместно с ДЕЯТЕЛЬНОСТЬЮ (ДЕЙСТВИЯМИ) и ЗАДАЧАМИ, необходимыми для проектирования и технического обслуживания безопасного ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКИХ ИЗДЕЛИЙ. Настоящий стандарт определяет требования для каждого ПРОЦЕССА жизненного цикла. Каждый ПРОЦЕСС жизненного цикла далее подразделяется на некую совокупность видов ДЕЯТЕЛЬНОСТИ, большинство из которых, в свою очередь, разделены на ЗАДАЧИ.

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 ИСО 14971:2007. Поэтому настоящий стандарт использует ссылки на ИСО 14971:2007. 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 ОПАСНОСТИ, определяется во время ДЕЯТЕЛЬНОСТИ по идентификации ОПАСНОСТИ в ПРОЦЕССЕ МЕНЕДЖМЕНТА РИСКА. ОПАСНОСТИ, которые могут быть косвенно вызваны ПРОГРАММНЫМ ОБЕСПЕЧЕНИЕМ (например, предоставляя вводящую в заблуждение информацию, которая может вызвать неверную реакцию администрирования), рассматриваются при определении того, 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 в настоящем стандарте, должен быть включен в ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА изделия согласно ИСО 14971:2007.

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

Рисунок 1 - Краткий обзор ПРОЦЕССОВ разработки

ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ и РАБОТ

Рисунок 1 - Краткий обзор ПРОЦЕССОВ разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ и РАБОТ

Рисунок 2 - Краткий обзор ПРОЦЕССОВ обслуживания ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ и РАБОТ

Рисунок 2 - Краткий обзор ПРОЦЕССОВ обслуживания ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ и РАБОТ

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).

Настоящий стандарт не устанавливает организационную структуру ИЗГОТОВИТЕЛЯ или какое именно структурное подразделение организации должно осуществлять выполнение ПРОЦЕССА, ДЕЯТЕЛЬНОСТИ или ЗАДАЧИ. Требование состоит в том, чтобы ПРОЦЕСС, ДЕЯТЕЛЬНОСТЬ или ЗАДАЧА были завершены для соответствия требованиям настоящего стандарта.

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 предоставляет объяснение пунктов настоящего стандарта. Приложение В содержит рекомендации по положениям настоящего стандарта.

Для целей настоящего стандарта:

- "должен" означает необходимость полного соответствия требованиям стандарта;
- "следует" означает, что соответствие требованиям рекомендуется, но не является обязательным;
- "может" используется, чтобы описать допустимый способ достижения соответствия требованиям;
- "установить" означает определять, документировать и осуществлять выполнение; и
- там, где в настоящем стандарте используется термин "если применимо", в сочетании с требуемым ПРОЦЕССОМ, ДЕЯТЬНОСТЬЮ, ЗАДАЧЕЙ или продукцией, то изготовитель должен использовать процесс, деятельность, задачу или продукцию, если не может документировано опровергнуть необходимость применения.

1 Область применения

1.1 Цель

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

1.2 Применимость

Настоящий стандарт применяют при разработке и технической поддержке ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКИХ ИЗДЕЛИЙ.

Настоящий стандарт применим при разработке и технической поддержке ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКИХ ИЗДЕЛИЙ, когда ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ само по себе является МЕДИЦИНСКИМ ИЗДЕЛИЕМ, или когда ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ прилагается к готовому МЕДИЦИНСКОМУ ИЗДЕЛИЮ или является неотъемлемой его частью.

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

1.3 Взаимосвязь с другими стандартами

Настоящий стандарт в отношении жизненного цикла ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКИХ ИЗДЕЛИЙ обычно используется совместно с другими применимыми стандартами при разработке МЕДИЦИНСКИХ ИЗДЕЛИЙ. В приложении С представлена взаимосвязь между настоящим стандартом и другими уместными стандартами.

1.4 Соответствие

Соответствие настоящему стандарту устанавливается как выполнение всех ПРОЦЕССОВ, ДЕЯТЕЛЬНОСТИ и ЗАДАЧ, указанных в настоящем стандарте, в соответствии с классом БЕЗОПАСНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.

Примечание - Классы БЕЗОПАСНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, назначенные каждому требованию, указываются в тексте настоящего стандарта, в конце соответствующих пунктов с требованиями.

Соответствие устанавливается с помощью проверки всей документации, требуемой настоящим стандартом, включая ФАЙЛ МЕНЕДЖМЕНТА РИСКА, и оценки ПРОЦЕССОВ, ДЕЯТЕЛЬНОСТИ и ЗАДАЧ, требуемых классом БЕЗОПАСНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ. (см. приложение D).

Примечание 1 - Эти оценки могут быть сделаны путем внешнего или внутреннего аудита.

Примечание 2 - Хотя и должны быть выполнены указанные ПРОЦЕССЫ, ДЕЯТЕЛЬНОСТЬ и ЗАДАЧИ, существует определенная гибкость в методах осуществления этих ПРОЦЕССОВ и выполнения ДЕЯТЕЛЬНОСТИ и ЗАДАЧ.

Примечание 3 - Если в требованиях указывается "соответствующим образом", но эти требования не выполнены, для оценки необходимо предоставить документацию, объясняющую причины отступления от требований настоящего стандарта.

Примечание 4 - Термин "соответствие", используемый в стандарте ИСО/МЭК 12207, применен в настоящем стандарте таким же образом.

2 Нормативные ссылки

В настоящем стандарте использована нормативная ссылка на следующий стандарт:

ИСО 14971:2007* Изделия медицинские. Применение менеджмента риска к медицинским изделиям (ISO 14971:2007, Medical devices - Application of risk management to medical devices)

_____

* Таблицу соответствия национальных стандартов международным см. по ссылке. - Примечание изготовителя базы данных.

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. или от чьего-то восприятия или опыта. АНОМАЛИИ могут быть обнаружены во время проверки, тестов, анализа, компиляции или использования ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ или прилагаемой документации, или в других случаях.

[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 выполнено в ПРОГРАММНОМ ОБЕСПЕЧЕНИИ.

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

Примечание - Основано на ИСО/МЭК 12207, определение 3.6.

3.6 **РЕЗУЛЬТАТ** (DELIVERABLE): Требуемый исход или готовая продукция (включая документацию) ДЕЯТЕЛЬНОСТИ или ЗАДАЧИ.

3.7 **ОЦЕНИВАНИЕ** (EVALUATION): Систематическое определение степени соответствия объекта установленным критериям.

[ИСО/МЭК 12207:1999, определение 3.9]

3.8 **ВРЕД** (HARM): Нанесение физического повреждения или другого вреда здоровью людей, или вреда имуществу или окружающей среде.

[ИСО/МЭК Руководящие указания 51:1999, определение 3.3]

3.9 **ОПАСНОСТЬ** (HAZARD): Потенциальный источник ВРЕДА.

[ИСО/МЭК Руководящие указания 51:1999, определение 3.5]

3.10 **ИЗГОТОВИТЕЛЬ** (MANUFACTURER): Физическое или юридическое лицо, ответственное за проектирование, изготовление, упаковывание и/или маркирование 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.

[ИСО 14971:2007, определение 2.6]

3.11 **МЕДИЦИНСКОЕ ИЗДЕЛИЕ** (MEDICAL DEVICE): Любой инструмент, аппарат, прибор, устройство, оборудование, имплантат, *in vitro* реагент или калибратор, программное обеспечение, материал или 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).

[ИСО 13485, определение 3.7].

Примечание 2 - В определениях, используемых в регулирующих отраслях разных стран, могут возникать некоторые различия.

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

3.13 **ОТЧЕТ О ПРОБЛЕМАХ** (PROBLEM REPORT): Запись о фактическом или возможном поведении ПРОГРАММНОГО ПРОДУКТА, из которой пользователь или заинтересованное лицо могут узнать о том, что является опасным, несоответствующим предусмотренному назначению или о том, что противоречит спецификации.

Примечание 1 - Настоящий стандарт не требует, чтобы каждый ОТЧЕТ О ПРОБЛЕМАХ приводил к изменениям в ПРОГРАММНОМ ПРОДУКТЕ. ИЗГОТОВИТЕЛЬ может от

клонить ОТЧЕТ О ПРОБЛЕМАХ для неверно понятого, ошибочного или несущественного события.

Примечание 2 - ОТЧЕТ О ПРОБЛЕМАХ может относиться к готовому ПРОГРАММНОМУ ПРОДУКТУ или к ПРОГРАММНОМУ ПРОДУКТУ, находящемуся в ПРОЦЕССЕ разработки.

Примечание 3 - Настоящий стандарт требует от ИЗГОТОВИТЕЛЯ осуществлять некоторые дополнительные шаги для каждого ОТЧЕТА О ПРОБЛЕМАХ, относящегося к уже выпущенному продукту, чтобы убедиться в том, что регулирующие действия идентифицированы и осуществлены.

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

[ИСО 9000:2008, определение 3.4.1]

Примечание - Термин "ДЕЯТЕЛЬНОСТЬ" включает и использование ресурсов.

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

[ИСО/МЭК 9003:2004, определение 3.11]

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

[ИСО/МЭК Руководящие указания 51:1999, определение 3.2]

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

[ИСО/МЭК Руководящие указания 51:1999, определение 3.10]

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

[ИСО 14971:2007, определение 2.16]

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

[ИСО 14971:2007, определение 2.18]

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

[ИСО 14971:2007, определение 2.19]

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

[ИСО/МЭК Руководящие указания 51:1999, определение 3.1]

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

[ИСО/МЭК 12207:1999, определение 3.25]

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

- несет угрозу жизни;
- приводит к стойкому ухудшению функционирования организма или к постоянному ущербу (необратимому повреждению) структуры тела, или
- требует медицинского или хирургического вмешательства с целью предотвращения стойкого ухудшения функционирования организма или постоянному ущербу (необратимому повреждению) 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 запуска в производство, которая:

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

Примечание - Заимствовано из ИСО/МЭК 12207, определение 3.11.

3.25 **ПРОГРАММНЫЙ ЭЛЕМЕНТ** (SOFTWARE ITEM): Любая идентифицируемая (выделяемая) часть компьютерной программы.

[ИСО/МЭК 9003:2004, определение 3.14]

Примечание - 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 МОДУЛЬ. Все уровни композиции, включая верхний и нижний уровни, можно назвать ПРОГРАММНЫМИ ЭЛЕМЕНТАМИ. Тогда ПРОГРАММНАЯ СИСТЕМА состоит из одного или более ПРОГРАММНЫХ ЭЛЕМЕНТОВ, и каждый ЭЛЕМЕНТ, в свою очередь, состоит из одного или более ПРОГРАММНЫХ МОДУЛЕЙ или подразделенных ПРОГРАММНЫХ ЭЛЕМЕНТОВ. Ответственность за обеспечение разделения и степень детализации ПРОГРАММНЫХ ЭЛЕМЕНТОВ и ПРОГРАММНЫХ МОДУЛЕЙ возлагается на ИЗГОТОВИТЕЛЯ.

3.26 **ПРОГРАММНЫЙ ПРОДУКТ** (SOFTWARE PRODUCT): Совокупность компьютерных программ, процедур и, по возможности, связанных с ними документации и данных.

[ИСО/МЭК 12207:1999, определение 3.26]

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

3.28 **ПРОГРАММНЫЙ МОДУЛЬ** (SOFTWARE UNIT): ПРОГРАММНЫЙ ЭЛЕМЕНТ, который не может быть разделен на более мелкие части.

Примечание - ПРОГРАММНЫЕ МОДУЛИ могут быть использованы в целях управления конфигурацией ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ или для его тестирования.

3.29 **ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ НЕИЗВЕСТНОГО ПРОИСХОЖДЕНИЯ**; ПОНП (software of unknown provenance; SOUP): ПРОГРАММНЫЙ ЭЛЕМЕНТ, который уже разработан и общедоступен, но не был предназначен для включения в состав МЕДИЦИНСКОГО ИЗДЕЛИЯ (также известен как "готовое ПО"), или программное обеспечение, разработанное ранее, для которого недоступны требуемые записи ПРОЦЕССОВ разработки.

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.

[ИСО/МЭК 12207:1999, определение 3.31]

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]

3.33 **ВЕРИФИКАЦИЯ** (VERIFICATION): Подтверждение на основе представления объективных свидетельств того, что установленные требования были выполнены.

Примечание 1 - Термин "верифицировано" используется для обозначения соответствующего статуса.

[ИСО 9000, 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 ЭЛЕМЕНТА КОНФИГУРАЦИИ.

Примечание 1 - Изменение ВЕРСИИ ПРОГРАММНОГО ПРОДУКТА, приводящее к появлению новой ВЕРСИИ, требует действий по управлению конфигурацией ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.

Примечание 2 - Заимствовано из ИСО/МЭК 12207, определение 3.37.

4 Общие требования

4.1 Система менеджмента качества

ИЗГОТОВИТЕЛЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКИХ ИЗДЕЛИЙ должен быть способен продемонстрировать его соответствие требованиям потребителя и применимым регулирующим требованиям.

Примечание 1 - Демонстрация этой способности может быть осуществлена с помощью СИСТЕМЫ менеджмента качества, которая соответствует следующим требованиям:

- ИСО 13485 (раздел 7), или
- национальному стандарту на систему менеджмента качества, или
- системе менеджмента качества, требуемой национальным регулированием.

Примечание 2 - Руководство, как применить требования менеджмента качества к ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ, можно найти в ИСО/МЭК 9003 [11].

4.2 МЕНЕДЖМЕНТ РИСКА

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

4.3 Классификация ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ в отношении БЕЗОПАСНОСТИ

а) Каждой ПРОГРАММНОЙ СИСТЕМЕ ИЗГОТОВИТЕЛЬ должен присвоить класс БЕЗОПАСНОСТИ (*A, B* или *C*) согласно возможным воздействиям на пациента, пользователя или иных лиц, исходя из ОПАСНОСТИ, возникновению которой может способствовать СИСТЕМА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.

Классы БЕЗОПАСНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ должны быть разделены по степени тяжести следующим образом:

- класс *A*: Невозможны никакие травмы или ущерб здоровью;
- класс *B*: Возможны НЕЗНАЧИТЕЛЬНЫЕ ТРАВМЫ;
- класс *C*: Возможны СЕРЬЕЗНЫЕ ТРАВМЫ или смерть.

Если ОПАСНОСТЬ может происходить из-за отказа в работе ПРОГРАММНОЙ СИСТЕМЫ, то вероятность такого отказа должна быть принята как стопроцентная.

Если РИСК смерти или СЕРЬЕЗНОЙ ТРАВМЫ, проистекающий от отказа ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, впоследствии уменьшается до допустимого уровня (как определено в ИСО 14971) с помощью аппаратных мер УПРАВЛЕНИЯ РИСКОМ или снижением последствий отказа, или снижением вероятности смерти или СЕРЬЕЗНОЙ ТРАВМЫ, являющейся результатом этого отказа, класс БЕЗОПАСНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ может быть снижен с *C* до *B*; и если РИСК НЕЗНАЧИТЕЛЬНОЙ ТРАВМЫ, проистекающий от отказа ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, подобным образом уменьшен до допустимого уровня при помощи аппаратных мер УПРАВЛЕНИЯ РИСКОМ, класс БЕЗОПАСНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ может быть снижен с *B* до *A*.

c) ИЗГОТОВИТЕЛЬ обязан документировать класс БЕЗОПАСНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, присвоенный каждой ПРОГРАММНОЙ СИСТЕМЕ, в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА.

d) Если ПРОГРАММНАЯ СИСТЕМА подразделяется на ПРОГРАММНЫЕ ЭЛЕМЕНТЫ, и в дальнейшем ПРОГРАММНЫЕ ЭЛЕМЕНТЫ, в свою очередь, подразделяются на ПРОГРАММНЫЕ МОДУЛИ, то такие ПРОГРАММНЫЕ ЭЛЕМЕНТЫ должны наследовать класс БЕЗОПАСНОСТИ первоначального ПРОГРАММНОГО ЭЛЕМЕНТА (или СИСТЕМЫ), если только ИЗГОТОВИТЕЛЬ не обосновывает в документации присвоение другого класса БЕЗОПАСНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ. Это обоснование должно объяснять, почему ПРОГРАММНЫЕ ЭЛЕМЕНТЫ являются изолированными настолько, что могут быть классифицированы отдельно.

e) ИЗГОТОВИТЕЛЬ должен документировать класс БЕЗОПАСНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ каждого ПРОГРАММНОГО ЭЛЕМЕНТА, если этот класс отличается от класса ПРОГРАММНОГО ЭЛЕМЕНТА, из которого он был выделен при разложении ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ на уровни.

f) Для соответствия настоящему стандарту там, где для ПРОГРАММНЫХ ЭЛЕМЕНТОВ конкретной классификации требуется ПРОЦЕСС, и этот ПРОЦЕСС необходимо применить к группе ПРОГРАММНЫХ ЭЛЕМЕНТОВ, ИЗГОТОВИТЕЛЬ должен использовать ПРОЦЕССЫ и ЗАДАЧИ, которые требуются для классификации ПРОГРАММНОГО ЭЛЕМЕНТА, оцененного наиболее высоко из всей группы, если только ИЗГОТОВИТЕЛЬ в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА не приводит документированные обоснования для использования более низкого класса БЕЗОПАСНОСТИ.

g) Каждой ПРОГРАММНОЙ СИСТЕМЕ, если ей не присвоен класс БЕЗОПАСНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, по умолчанию, должен быть присвоен класс *C.*

Примечание - В требованиях, приведенных далее, классы БЕЗОПАСНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, для которых данное требование должно выполняться, будут указаны после требования в виде (класс...).

5 ПРОЦЕСС разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

5.1 Планирование разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

5.1.1 План разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

ИЗГОТОВИТЕЛЬ должен установить план (или планы) разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ с целью провести всю необходимую ДЕЯТЕЛЬНОСТЬ в отношении ПРОЦЕССА разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, соответствующую области, важности и классу БЕЗОПАСНОСТИ разрабатываемой ПРОГРАММНОЙ СИСТЕМЫ. МОДЕЛЬ ЖИЗНЕННОГО ЦИКЛА РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ должна быть либо полностью определена, либо должна ссылаться на план (или планы). План должен содержать:

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

b) РЕЗУЛЬТАТЫ (включая документацию) ДЕЯТЕЛЬНОСТИ и ЗАДАЧ;

c) ПРОСЛЕЖИВАЕМОСТЬ между требованиями СИСТЕМЫ, требованиями ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, испытанием ПРОГРАММНОЙ СИСТЕМЫ и мерами УПРАВЛЕНИЯ РИСКОМ, включенными в программное обеспечение;

d) конфигурацию ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ и управление изменениями, включая ЭЛЕМЕНТЫ КОНФИГУРАЦИИ ПОНП и ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, используемого для поддержки разработки;

e) программное решение проблем для обработки проблем, обнаруженных в ПРОГРАММНЫХ ПРОДУКТАХ, 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 РЕЗУЛЬТАТЫ) для различных ПРОГРАММНЫХ ЭЛЕМЕНТОВ, в соответствии с классами БЕЗОПАСНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ для каждого ПРОГРАММНОГО ЭЛЕМЕНТА ПРОГРАММНОЙ СИСТЕМЫ.

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

Примечание 3 - Другие ПРОЦЕССЫ изложены 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) В плане разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ в качестве входных данных ИЗГОТОВИТЕЛЬ должен указать требования СИСТЕМЫ;

b) В план разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИЗГОТОВИТЕЛЬ должен включить или дать ссылки на процедуры для координации проектирования и разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, а также на деятельность по валидации, необходимой для соответствия требованиям 4.1 (классы *A, B, C*).

Примечание - Может не существовать различий между требованиями ПРОГРАММНОЙ СИСТЕМЫ и требованиями СИСТЕМЫ, если ПРОГРАММНАЯ СИСТЕМА является отдельной СИСТЕМОЙ (например, если изделие состоит только из ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ).

5.1.4 Стандарты, методы и инструменты планирования разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В план разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИЗГОТОВИТЕЛЬ должен включить или дать ссылки:

- на стандарты;
- методы;
- инструменты, связанные с разработкой ПРОГРАММНЫХ ЭЛЕМЕНТОВ класса *C* (класс *C*).

5.1.5 Программная интеграция и планирование тестирования интеграции

В плане развития ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИЗГОТОВИТЕЛЬ должен указать или дать ссылки на план интеграции ПРОГРАММНЫХ ЭЛЕМЕНТОВ (включая ПОНП) и осуществление тестирования во время интеграции (классы *B, C*).

Примечание - Возможно комбинирование тестирования интеграции и тестирования ПРОГРАММНОЙ СИСТЕМЫ в единый план и совокупность ДЕЯТЕЛЬНОСТИ.

5.1.6 Планирование ВЕРИФИКАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В план разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИЗГОТОВИТЕЛЬ должен включить или дать ссылки на следующую ВЕРИФИЦИРУЕМУЮ информацию:

a) РЕЗУЛЬТАТЫ, требующие ВЕРИФИКАЦИИ;

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

c) контрольные точки, на которых РЕЗУЛЬТАТЫ ВЕРИФИЦИРОВАНЫ;

d) критерии приемки для ВЕРИФИКАЦИИ РЕЗУЛЬТАТОВ (классы *A, B, C*).

5.1.7 Планирование МЕНЕДЖМЕНТА РИСКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В план разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИЗГОТОВИТЕЛЬ должен включить или дать ссылки на план осуществления ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, в отношении ДЕЯТЕЛЬНОСТИ и ЗАДАЧ, включая МЕНЕДЖМЕНТ РИСКА, применяемый к ПОНП (классы *A, B, C*).

5.1.8 Документация планирования

В план разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИЗГОТОВИТЕЛЬ должен включить или дать ссылки 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) предусмотренные пользователи документа;

d) процедуры и ответственность за разработку, анализ, одобрение и модификацию (классы *A, B, C*).

5.1.9 Планирование менеджмента конфигурации ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В план разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИЗГОТОВИТЕЛЬ должен включить или дать ссылки 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) организационную структуру (структуры), отвечающие за осуществление менеджмента конфигурации ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ и ДЕЯТЕЛЬНОСТИ;

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).*

5.1.10 Поддержка элементов, подлежащих управлению

Элементы, подлежащие управлению, должны включать в себя инструменты, элементы или параметры, используемые для разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКИХ ИЗДЕЛИЙ, которые могут воздействовать на ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МЕДИЦИНСКИХ ИЗДЕЛИЙ (классы *B, C*).

Примечание - Примеры подобных элементов включают компиляторные/ассемблерные версии, созданные файлы, командные файлы и определенные параметры настройки сторонних устройств.

5.1.11 Управление ЭЛЕМЕНТАМИ КОНФИГУРАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ до ВЕРИФИКАЦИИ

ИЗГОТОВИТЕЛЬ должен запланировать размещение ЭЛЕМЕНТОВ КОНФИГУРАЦИИ под управление документированного менеджмента конфигурации, прежде чем они будут ВЕРИФИЦИРОВАНЫ.

5.2 Анализ требований к ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

5.2.1 Определение и документирование требований к ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ в зависимости от требований СИСТЕМЫ

Для каждой ПРОГРАММНОЙ СИСТЕМЫ к МЕДИЦИНСКОМУ ИЗДЕЛИЮ ИЗГОТОВИТЕЛЬ должен определить и документировать требования ПРОГРАММНОЙ СИСТЕМЫ, исходя из требований уровня СИСТЕМЫ (классы *A, B, C*).

Примечание - Могут отсутствовать различия между требованиями ПРОГРАММНОЙ СИСТЕМЫ и требованиями СИСТЕМЫ, если СИСТЕМА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ является отдельной СИСТЕМОЙ (создается только программное обеспечение).

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 в себя:

- эксплуатационные характеристики (например, цели ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, координация требований);
- физические характеристики (например, язык машинного кода, платформу, операционную СИСТЕМУ);
- компьютерные характеристики (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) средства взаимодействия между ПРОГРАММНОЙ СИСТЕМОЙ и другими СИСТЕМАМИ;

d) ПРОГРАММНЫЕ СРЕДСТВА управления предупреждением и оповещением оператора;

e) требования к ЗАЩИЩЕННОСТИ.

Примечание 3 - Например:

- связанные с компромиссом относительно конфиденциальной информации;
- идентификация;
- авторизация;
- контрольный журнал;
- коммуникационная целостность;

f) требования к разработке требований к удобству и простоте использования (эксплуатационной пригодности), которые чувствительны к человеческим ошибкам и обучению.

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 - Для информации относительно требований к разработке удобства и простоты использования (эксплуатационной пригодности) см. МЭК 60601-1-6;

g) определение данных и требований к базе данных.

Примечание 6 - Например:

- форма;
- размерность;
- функция;

h) установление и принятие требований к поставляемому ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ МЕДИЦИНСКИХ ИЗДЕЛИЙ для работ и технической поддержки сайта или сайтов;

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

j) разрабатываемая документация для пользователя;

k) требования пользователя к технической поддержке;

l) регулирующие требования (классы *A, B, C*).

Примечание 7 - Все эти требования могут не иметься в наличии на момент начала разработки.

Примечание 8 - ИСО/МЭК 9126-1 [8] обеспечивает 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 включить в требования меры УПРАВЛЕНИЯ РИСКОМ, осуществленные в отношении ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, как соответствующие ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ МЕДИЦИНСКИХ ИЗДЕЛИЙ (классы *B, C*), на случай отказа аппаратных средств и потенциальных дефектов ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.

Примечание - Эти требования могут быть недоступны в начале процесса разработки и могут быть изменены по мере того, как создается программное обеспечение и устанавливаются дальнейшие меры УПРАВЛЕНИЯ РИСКА.

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 к СИСТЕМЕ

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 Проверка требований к ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

ИЗГОТОВИТЕЛЬ должен проверить и документировать, что требования к ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ:

a) выполняют требования к СИСТЕМЕ, включая требования, относящиеся к УПРАВЛЕНИЮ РИСКОМ;

b) не противоречат друг другу;

c) выражены в терминах, которые не допускают двусмысленности;

d) сформулированы в терминах, которые позволяют установить критерии испытаний и осуществить их, а также определить, были ли удовлетворены установленные критерии испытаний;

e) могут быть идентифицированы уникальным образом;

f) обеспечивали ПРОСЛЕЖИВАЕМОСТЬ 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 ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКИХ ИЗДЕЛИЙ в документированную АРХИТЕКТУРУ, которая описывает структуру ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ и идентифицирует ПРОГРАММНЫЕ ЭЛЕМЕНТЫ (классы *B, C).*

5.3.2 Разработка АРХИТЕКТУРЫ для интерфейсов ПРОГРАММНЫХ ЭЛЕМЕНТОВ

ИЗГОТОВИТЕЛЬ должен разработать и документировать АРХИТЕКТУРУ для интерфейсов между ПРОГРАММНЫМИ ЭЛЕМЕНТАМИ и компонентами, внешними по отношению к ПРОГРАММНЫМ ЭЛЕМЕНТАМ (как к ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ, так и к аппаратным средствам) и между ПРОГРАММНЫМИ ЭЛЕМЕНТАМИ (классы *B, C*).

5.3.3 Определение требований к функциональным и эксплуатационным характеристикам элементов ПОНП

Если ПРОГРАММНЫЙ ЭЛЕМЕНТ идентифицирован как ПОНП, ИЗГОТОВИТЕЛЬ должен определить требования к функциональным и эксплуатационным характеристикам элемента ПОНП, которые необходимы для использования его согласно предусмотренному назначению (классы *B, C*).

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

Если ПРОГРАММНЫЙ ЭЛЕМЕНТ идентифицирован 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 указать обособленность ПРОГРАММНЫХ ЭЛЕМЕНТОВ, которые существенны для УПРАВЛЕНИЯ РИСКОМ, и установить, как можно удостовериться в том, что такая обособленность результативна (класс *C*).

Примечание - В качестве примера разделения можно взять ПРОГРАММНЫЕ ЭЛЕМЕНТЫ, выполняемые другими процессорами. В результативности обособленности можно удостовериться путем отсутствия общих ресурсов у разных процессоров.

5.3.6 Проверка АРХИТЕКТУРЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

ИЗГОТОВИТЕЛЬ должен проверить и документировать, что:

a) АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ выполняет требования к СИСТЕМЕ и ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ, включая требования, относящиеся к УПРАВЛЕНИЮ РИСКОМ;

b) АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ способна поддерживать взаимодействие между ПРОГРАММНЫМИ ЭЛЕМЕНТАМИ, а также между ПРОГРАММНЫМИ ЭЛЕМЕНТАМИ и аппаратными средствами;

c) АРХИТЕКТУРА МЕДИЦИНСКИХ ИЗДЕЛИЙ поддерживает правильную работу любых элементов ПОНП (классы *B, C*).

5.4 Детализированная разработка ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

5.4.1 Развитие АРХИТЕКТУРЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ в ПРОГРАММНЫЕ МОДУЛИ

ИЗГОТОВИТЕЛЬ должен развивать АРХИТЕКТУРУ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, пока она не будет представлена в виде ПРОГРАММНЫХ МОДУЛЕЙ (классы *B, C*).

5.4.2 Разработка детализированного проекта для каждого ПРОГРАММНОГО МОДУЛЯ

ИЗГОТОВИТЕЛЬ должен разработать и документировать детализированный проект для каждого ПРОГРАММНОГО МОДУЛЯ ПРОГРАММНОГО ЭЛЕМЕНТА (класс *C*).

5.4.3 Разработка детализированного проекта для интерфейсов

ИЗГОТОВИТЕЛЬ должен разработать и документировать детализированный проект для всех интерфейсов между ПРОГРАММНЫМИ МОДУЛЯМИ и внешними компонентами (аппаратными или программными средствами), а также для интерфейсов между ПРОГРАММНЫМИ МОДУЛЯМИ (класс *C*).

5.4.4 Проверка детализированного проекта

ИЗГОТОВИТЕЛЬ должен проверять и документировать, что детализированный проект ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ:

a) соответствует АРХИТЕКТУРЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ;

b) не вступает в противоречия с АРХИТЕКТУРОЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (класс *C*).

5.5 Исполнение и проверка ПРОГРАММНЫХ МОДУЛЕЙ

5.5.1 Исполнение каждого ПРОГРАММНОГО МОДУЛЯ

ИЗГОТОВИТЕЛЬ должен исполнять каждый ПРОГРАММНЫЙ МОДУЛЬ (классы *A, B, C*).

5.5.2 Установление ПРОЦЕССА ВЕРИФИКАЦИИ ПРОГРАММНОГО МОДУЛЯ

ИЗГОТОВИТЕЛЬ должен определить стратегии, методы и процедуры для ВЕРИФИКАЦИИ каждого ПРОГРАММНОГО МОДУЛЯ. Там, где ВЕРИФИКАЦИЯ осуществляется с помощью испытаний, правильность процедур их проведения должна быть ОЦЕНЕНА (классы *B, C*).

Примечание - Возможно объединение общего испытания и испытания ПРОГРАММНОЙ СИСТЕМЫ в единый план ДЕЯТЕЛЬНОСТИ.

5.5.3 Критерии приемки ПРОГРАММНЫХ МОДУЛЕЙ

ИЗГОТОВИТЕЛЬ должен установить критерии приемлемости для ПРОГРАММНЫХ МОДУЛЕЙ до их объединения в более крупные ПРОГРАММНЫЕ ЭЛЕМЕНТЫ соответствующим образом и удостовериться, что ПРОГРАММНЫЕ МОДУЛИ соответствуют критериям приемки (классы *B, C*).

Примечание - Примеры критериев приемки:

- отвечает ли программный код требованиям, включая меры УПРАВЛЕНИЯ РИСКАМИ (РИСКОМ)?
- нет ли в программном коде противоречий с интерфейсами, документированными в детализированном проекте ПРОГРАММНЫХ МОДУЛЕЙ?
- соответствует ли программный код процедурам программирования или стандартам кодирования?

5.5.4 Дополнительные критерии приемки ПРОГРАММНЫХ МОДУЛЕЙ

ИЗГОТОВИТЕЛЬ должен включить в существующий проект дополнительные критерии приемки, предназначенные:

a) для соответствующей последовательности событий;

b) потока данных и текущего контроля;

c) планируемого распределения ресурсов;

d) работы с ошибками (определение ошибки, локализация и восстановление);

e) инициализации переменных;

f) самодиагностики;

g) управления памятью и переполнений памяти;

h) граничных условий (класс *C*).

5.5.5 ВЕРИФИКАЦИЯ ПРОГРАММНЫХ МОДУЛЕЙ

ИЗГОТОВИТЕЛЬ должен выполнять ВЕРИФИКАЦИЮ ПРОГРАММНЫХ МОДУЛЕЙ и документировать РЕЗУЛЬТАТЫ (классы *B, C*).

5.6 Программная интеграция и испытания в отношении интеграции

5.6.1 Интеграция ПРОГРАММНЫХ МОДУЛЕЙ

ИЗГОТОВИТЕЛЬ должен интегрировать ПРОГРАММНЫЕ МОДУЛИ согласно плану интеграции (см. 5.1.5) (классы *B, C*).

5.6.2 Проверка программной интеграции

ИЗГОТОВИТЕЛЬ должен проверить и осуществить записи в отношении следующих аспектов интеграции ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ в соответствии с планом интеграции (см. 5.1.5):

a) ПРОГРАММНЫЕ МОДУЛИ должны быть интегрированы в ПРОГРАММНЫЕ ЭЛЕМЕНТЫ и ПРОГРАММНЫЕ СИСТЕМЫ;

b) аппаратные элементы, ПРОГРАММНЫЕ ЭЛЕМЕНТЫ и поддержка ручного управления (например, интерфейс, приспособленный для человека, диалоговые меню подсказки, распознавание речи, речевое управление СИСТЕМОЙ), интегрированные в СИСТЕМУ (классы *B, C*).

Примечание - Данная проверка заключается только в том, чтобы проверить, что элементы были интегрированы согласно плану, а не в том, что они выполняют предусмотренное назначение. ВЕРИФИКАЦИЯ должна осуществляться как некоторая форма инспектирования.

5.6.3 Испытания интегрированного ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

ИЗГОТОВИТЕЛЬ должен испытать интегрированные ПРОГРАММНЫЕ ЭЛЕМЕНТЫ в соответствии с планом интеграции (см. 5.1.5) и документировать РЕЗУЛЬТАТЫ (классы *B, C*).

5.6.4 Содержание испытаний в отношении интеграции

При испытаниях, в отношении интеграции ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ИЗГОТОВИТЕЛЬ должен установить, что ПРОГРАММНЫЕ ЭЛЕМЕНТЫ функционируют в соответствии с предусмотренным назначением (классы *B, C*).

Примечание 1 - Примерами можно считать:

- требуемую функциональность ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ;
- реализацию мер УПРАВЛЕНИЯ РИСКОМ;
- определенную синхронизацию и другие режимы работы;
- определенное функционирование внутренних и внешних интерфейсов;
- испытания в режиме неисправности, включая прогнозируемое неправильное применение.

Примечание 2 - Возможно объединение испытания интеграции и испытания ПРОГРАММНОЙ СИСТЕМЫ в единый план ДЕЯТЕЛЬНОСТИ.

5.6.5 Проверка процедур испытаний в отношении интеграции

ИЗГОТОВИТЕЛЬ должен ОЦЕНИТЬ правильность процедуры испытаний в отношении интеграции (классы *B, C*).

5.6.6 Проведение регрессионных испытаний

Если элементы ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ подлежат интеграции, ИЗГОТОВИТЕЛЬ должен провести РЕГРЕССИОННУЮ ПРОВЕРКУ, подходящую для демонстрации того, что в ранее интегрированном ПРОГРАММНОМ ОБЕСПЕЧЕНИИ не были обнаружены дефекты (классы *B, C*).

5.6.7 Содержание записей в отношении регрессионных испытаний

ИЗГОТОВИТЕЛЬ должен:

a) документировать РЕЗУЛЬТАТЫ испытаний (соответствует, не соответствует и перечень АНОМАЛИЙ);

b) сохранить существенные записи с целью сделать возможным проведение повторных испытаний;

c) указать лицо, проводившее испытания (классы *B, C*).

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

- характеристик условий проведения конкретного испытания, показывающих требуемые действия и ожидаемые РЕЗУЛЬТАТЫ;
- составления перечня используемого оборудования;
- записей внешних устройств (включая ПРОГРАММНЫЕ инструменты), используемых при проведении испытаний.

5.6.8 Использование программного ПРОЦЕССА решения проблем

АНОМАЛИИ, обнаруженные во время интеграции ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ и испытаний интеграции, ИЗГОТОВИТЕЛЬ должен ввести в программный ПРОЦЕСС решения проблем (классы *B, C*).

Примечание - См. раздел 9.

5.7 Испытания СИСТЕМЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

5.7.1 Установление испытаний в отношении требований ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Для проведения испытаний СИСТЕМЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИЗГОТОВИТЕЛЬ должен определить и выполнить перечень испытаний, выраженных как входные данные, ожидаемые РЕЗУЛЬТАТЫ, критерии приемки и процедуры, с целью учета всех требований ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (классы *B, C*).

Примечание 1 - Возможно объединение испытаний интеграции и испытания СИСТЕМЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ в единый план ДЕЯТЕЛЬНОСТИ. Также допустимо проверять требования ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ на более ранних стадиях.

Примечание 2 - Могут быть проведены не только тестирования отдельных требований, но и тестирования комбинаций требований, особенно если между требованиями существуют зависимости.

5.7.2 Использование программного ПРОЦЕССА решения проблем

АНОМАЛИИ, обнаруженные во время испытаний СИСТЕМЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ИЗГОТОВИТЕЛЬ должен ввести в программный ПРОЦЕСС решения проблем (классы *B, C*).

5.7.3 Повторные испытания после внесения изменений

Если во время испытаний СИСТЕМЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ вносятся изменения, ИЗГОТОВИТЕЛЬ обязан:

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

b) провести испытания, необходимые для демонстрации отсутствия возникновения непреднамеренных побочных эффектов;

c) выполнить соответствующую ДЕЯТЕЛЬНОСТЬ по УПРАВЛЕНИЮ РИСКОМ, как установлено в 7.4 (классы *B, C*).

5.7.4 ВЕРИФИКАЦИЯ испытаний СИСТЕМЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

ИЗГОТОВИТЕЛЬ должен удостовериться, что:

a) стратегии ВЕРИФИКАЦИИ и используемые процедуры испытаний являются соответствующими;

b) процедуры испытаний СИСТЕМЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ прослеживаются до требований ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ;

c) все требования ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ были подвергнуты испытаниям или ВЕРИФИЦИРОВАНЫ любым иным способом;

d) РЕЗУЛЬТАТЫ испытаний отвечают требуемым критериям приемки (классы *B, C*).

5.7.5 Содержание отчета по испытаниям СИСТЕМЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

ИЗГОТОВИТЕЛЬ должен:

a) документировать РЕЗУЛЬТАТЫ испытаний (соответствует, не соответствует и список АНОМАЛИЙ);

b) сохранить отчеты, важные для обеспечения возможности повторения испытаний;

c) идентифицировать лицо, проводившее испытания (классы *B, C*).

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

- характеристик условий проведения конкретного испытания, показывающих требуемые действия и ожидаемые РЕЗУЛЬТАТЫ;
- составления перечня используемого оборудования;
- записей внешних устройств (включая ПРОГРАММНЫЕ инструменты), используемых при проведении испытаний.

5.8 Выпуск ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

5.8.1 Обеспечение полного завершения ВЕРИФИКАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

ИЗГОТОВИТЕЛЬ до выпуска ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ в обращение должен обеспечить, что ВЕРИФИКАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ была полностью завершена, а РЕЗУЛЬТАТЫ ОЦЕНЕНЫ (классы *B, C*).

5.8.2 Документирование известных остаточных АНОМАЛИЙ

ИЗГОТОВИТЕЛЬ должен задокументировать все известные остаточные АНОМАЛИИ (классы *B, C*).

5.8.3 ОЦЕНИВАНИЕ известных остаточных АНОМАЛИЙ

ИЗГОТОВИТЕЛЬ должен удостовериться, что 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 ПРОДУКТА, которая будет выпущена (классы *A, B, C*).

5.8.5 Документирование создания выпущенного ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

ИЗГОТОВИТЕЛЬ должен документировать процедуру и программную среду, которые использовались при создании выпущенного ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (классы *B, C*).

5.8.6 Обеспечение полного завершения деятельности и задач

ИЗГОТОВИТЕЛЬ должен обеспечить, что вся ДЕЯТЕЛЬНОСТЬ и все ЗАДАЧИ полностью завершены, а также связанная с ними документация (классы *B, C*).

5.8.7 Архивирование ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

ИЗГОТОВИТЕЛЬ должен хранить в архиве в течение как минимум срока службы изделия, установленного ИЗГОТОВИТЕЛЕМ, или в течение срока, установленного соответствующими регулирующими требованиями, следующее:

a) ПРОГРАММНЫЕ ПРОДУКТЫ и ЭЛЕМЕНТЫ КОНФИГУРАЦИИ;

b) документацию (классы *B, C*).

5.8.8 Обеспечение воспроизводимости выпуска ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

ИЗГОТОВИТЕЛЬ должен установить процедуры, обеспечивающие, что выпущенный ПРОГРАММНЫЙ ПРОДУКТ будет поставлен пользователю (к месту его применения) без искажения или несанкционированного изменения. Эти процедуры должны распространяться на производство и обращение со средствами, содержащими ПРОГРАММНЫЙ ПРОДУКТ, и включать в себя, если применимо:

- создание копии;
- средства маркировки;
- упаковку;
- защиту;
- хранение;
- поставку (классы *B, C*).

6 ПРОЦЕСС технической поддержки 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 включать в себя:

a) процедуры:

1) для получения (установления);

2) документирования;

3) оценивания;

4) принятия решения;

5) отслеживания;

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

b) критерии для определения проблем с обратной связью;

c) использование ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ;

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

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

f) процедуры по ОЦЕНИВАНИЮ и осуществлению:

1) обновления;

2) местоположения ошибок;

3) исправления, вносимые в коды ("заплатки", "патчи");

4) устаревания;

5) 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 выпущенного ПРОГРАММНОГО ПРОДУКТА как внутри своей собственной организации, так и от пользователей (классы *A, B, C*).

6.2.1.2 Документирование обратной связи

Обратная связь должна быть документирована и ОЦЕНЕНА с целью определения существования проблемы в выпущенном ПРОГРАММНОМ ПРОДУКТЕ. Любая такая проблема должна быть зарегистрирована в ОТЧЕТЕ О ПРОБЛЕМАХ (см. раздел 9). ОТЧЕТ О ПРОБЛЕМАХ должен включать в себя фактические или потенциальные неблагоприятные события и отклонения от спецификации (классы *A, B, C*).

6.2.1.3 ОЦЕНИВАНИЕ влияния ОТЧЕТА О ПРОБЛЕМАХ на БЕЗОПАСНОСТЬ

Каждый ОТЧЕТ О ПРОБЛЕМАХ должен быть ОЦЕНЕН с целью определения его влияния на БЕЗОПАСНОСТЬ выпущенного ПРОГРАММНОГО ПРОДУКТА и необходимость изменений в выпущенном ПРОГРАММНОМ ПРОДУКТЕ (классы *A, B, C*).

6.2.2 Использование программного ПРОЦЕССА решения проблем

ИЗГОТОВИТЕЛЬ должен использовать программный ПРОЦЕСС решения проблем (см. раздел 9) в отношении ОТЧЕТОВ О ПРОБЛЕМАХ (классы *A, B, C*).

Примечание - Когда эта ДЕЯТЕЛЬНОСТЬ уже выполнена, любое изменение класса БЕЗОПАСНОСТИ ПРОГРАММНОЙ СИСТЕМЫ или ее ПРОГРАММНОГО ЭЛЕМЕНТА должно быть известно.

6.2.3 Анализ ЗАПРОСОВ НА ИЗМЕНЕНИЕ

В дополнение к анализу, требуемому в разделе 9, ИЗГОТОВИТЕЛЬ должен анализировать каждый ЗАПРОС НА ИЗМЕНЕНИЕ с целью определения его влияния на конструкцию выпущенных ПРОГРАММНЫХ ПРОДУКТОВ и СИСТЕМ, с которыми он взаимодействует (классы *B, C*).

6.2.4 Одобрение ЗАПРОСА НА ИЗМЕНЕНИЕ

ИЗГОТОВИТЕЛЬ должен ОЦЕНИТЬ и одобрить ЗАПРОСЫ НА ИЗМЕНЕНИЯ, которые модифицируют выпущенные ПРОГРАММНЫЕ ПРОДУКТЫ (классы *A, B, C*).

6.2.5 Информирование пользователей и регулирующих органов

ИЗГОТОВИТЕЛЬ должен идентифицировать одобренные ЗАПРОСЫ НА ИЗМЕНЕНИЯ, которые влияют на выпущенные ПРОГРАММНЫЕ ПРОДУКТЫ.

Если предусмотрено региональными регулирующими требованиями, ИЗГОТОВИТЕЛЬ должен информировать пользователей и регулирующие органы:

a) о любых проблемах в отношении выпущенных ПРОГРАММНЫХ ПРОДУКТОВ и последствиях длительного использования неизмененного продукта;

b) о характере любых доступных изменений в выпущенных ПРОГРАММНЫХ ПРОДУКТАХ 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 использовать программный ПРОЦЕСС разработки (см. раздел 5) или установленный ПРОЦЕСС технической поддержки для осуществления модификации (классы *A, B, C*).

Примечание - Требования для изменений ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, относящиеся к МЕНЕДЖМЕНТУ РИСКА, - см. 7.4.

6.3.2 Повторный выпуск модифицированной ПРОГРАММНОЙ СИСТЕМЫ

ИЗГОТОВИТЕЛЬ должен выпускать модифицированные ПРОГРАММНЫЕ СИСТЕМЫ согласно 5.8. Модификации могут быть реализованы как часть полной повторно выпущенной ПРОГРАММНОЙ СИСТЕМЫ или как набор модификаций, включающий измененные ПРОГРАММНЫЕ ЭЛЕМЕНТЫ, а также инструменты, необходимые для установки изменений, как модификации существующей ПРОГРАММНОЙ СИСТЕМЫ (классы *A, B, C*).

7 ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

7.1 Анализ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, способствующего возникновению опасных ситуаций

7.1.1 Идентификация ПРОГРАММНЫХ ЭЛЕМЕНТОВ, которые могут способствовать возникновению опасных ситуаций

ИЗГОТОВИТЕЛЬ должен идентифицировать ПРОГРАММНЫЕ ЭЛЕМЕНТЫ, которые могут способствовать возникновению опасных ситуаций, идентифицированных при осуществлении ДЕЯТЕЛЬНОСТИ по АНАЛИЗУ РИСКА МЕДИЦИНСКОГО ИЗДЕЛИЯ, которая должна быть проведена в соответствии с ИСО 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, приводящих к опасным ситуациям

ИЗГОТОВИТЕЛЬ должен идентифицировать потенциальные причины в ПРОГРАММНОМ ЭЛЕМЕНТЕ, которые определены в предыдущем пункте как содействующие возникновению опасных ситуаций.

ИЗГОТОВИТЕЛЬ должен рассматривать потенциальные причины, включая, если применимо:

a) неправильную или неполную спецификацию функциональности;

b) дефекты ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, идентифицированные в определенных функциях ПРОГРАММНОГО ЭЛЕМЕНТА;

c) отказы или неожидаемые РЕЗУЛЬТАТЫ, исходящие от ПОНП;

d) отказы аппаратных средств или другие дефекты ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, которые могут привести к непредсказуемым операциям ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ;

e) обосновано прогнозируемое неправильное применение (классы *B, C*).

7.1.3 ОЦЕНКА опубликованных списков АНОМАЛИЙ ПОНП

Если отказ или неожидаемые РЕЗУЛЬТАТЫ, исходящие от ПОНП, являются потенциальной причиной того, что ПРОГРАММНЫЙ ЭЛЕМЕНТ может содействовать возникновению опасных ситуаций, 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 возникновения ПРОГРАММНОГО ЭЛЕМЕНТА, который может содействовать возникновению опасных ситуаций (см. ИСО 14971) (классы *B, C*).

7.1.5 Документирование последовательности событий

ИЗГОТОВИТЕЛЬ должен документировать в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА последовательности событий, идентифицированых в 7.1.2, которые могут привести к опасной ситуации (классы *B, C*).

7.2 Меры по УПРАВЛЕНИЮ РИСКОМ

7.2.1 Выбор мер по УПРАВЛЕНИЮ РИСКОМ

В отношении каждой потенциальной причины в ПРОГРАММНОМ ЭЛЕМЕНТЕ, зарегистрированной в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА, которая может содействовать возникновению опасных ситуаций, ИЗГОТОВИТЕЛЬ должен определить и документировать меры по УПРАВЛЕНИЮ РИСКОМ (классы *B, C*).

Примечание - Меры по УПРАВЛЕНИЮ РИСКОМ могут быть осуществлены в аппаратных средствах, ПРОГРАММНОМ ОБЕСПЕЧЕНИИ, рабочей внешней среде или инструкциях пользователя.

7.2.2 Меры по УПРАВЛЕНИЮ РИСКОМ, осуществляемые в ПРОГРАММНОМ ОБЕСПЕЧЕНИИ

Если меры по УПРАВЛЕНИЮ РИСКОМ осуществляются как часть функций ПРОГРАММНОГО ЭЛЕМЕНТА, ИЗГОТОВИТЕЛЬ должен:

a) включить меры по УПРАВЛЕНИЮ РИСКОМ в требования к ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ;

b) назначить класс БЕЗОПАСНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ для ПРОГРАММНОГО ЭЛЕМЕНТА, основанный на возможных последствиях ОПАСНОСТИ, которой управляет мера по УПРАВЛЕНИЮ РИСКОМ;

c) разработать ПРОГРАММНЫЙ ЭЛЕМЕНТ в соответствии с разделом 5 (классы *B, C*).

Примечание - Это требование обеспечивает дополнительное уточнение требований по УПРАВЛЕНИЮ РИСКОМ ИСО 14971.

7.3 ВЕРИФИКАЦИЯ мер по УПРАВЛЕНИЮ РИСКОМ

7.3.1 Проверка мер по УПРАВЛЕНИЮ РИСКОМ

Выполнение каждой меры по УПРАВЛЕНИЮ РИСКОМ, документированной в 7.2, должно быть верифицировано, а сама ВЕРИФИКАЦИЯ должна быть документирована (классы *B, C*).

7.3.2 Документирование любых новых последовательностей событий

Если мера по УПРАВЛЕНИЮ РИСКОМ осуществляется как ПРОГРАММНЫЙ ЭЛЕМЕНТ, ИЗГОТОВИТЕЛЬ должен ОЦЕНИТЬ меру по УПРАВЛЕНИЮ РИСКОМ с целью идентифицировать и документировать в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА любые новые последовательности событий, которые могут привести к возникновению опасной ситуации (классы *B, C*).

7.3.3 Документирование ПРОСЛЕЖИВАЕМОСТИ

ИЗГОТОВИТЕЛЬ должен документировать ПРОСЛЕЖИВАЕМОСТЬ в отношении ОПАСНОСТЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ соответствующим образом. Например, если применимо:

a) от опасной ситуации до ПРОГРАММНОГО ЭЛЕМЕНТА;

b) от ПРОГРАММНОГО ЭЛЕМЕНТА до конкретной причины в ПРОГРАММНОМ ОБЕСПЕЧЕНИИ;

c) от причины в ПРОГРАММНОМ ОБЕСПЕЧЕНИИ до мер по УПРАВЛЕНИЮ РИСКОМ;

d) от мер по УПРАВЛЕНИЮ РИСКОМ до ВЕРИФИКАЦИИ мер по УПРАВЛЕНИЮ РИСКОМ (классы *B, C*).

Примечание - См. ИСО 14971 - отчет по МЕНЕДЖМЕНТУ РИСКА.

7.4 МЕНЕДЖМЕНТ РИСКА в отношении изменений ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

7.4.1 Анализ изменений ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКИХ ИЗДЕЛИЙ в отношении БЕЗОПАСНОСТИ

ИЗГОТОВИТЕЛЬ обязан анализировать изменения в ПРОГРАММНОМ ОБЕСПЕЧЕНИИ МЕДИЦИНСКИХ ИЗДЕЛИЙ (включая ПОНП), с целью определения:

a) существования не выявленных ранее причин, способствующих возникновению опасной ситуации;

b) 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 7.1-7.3, основанную на результатах проведенных анализов (классы *B, C*).

8 ПРОЦЕСС менеджмента конфигурации ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

8.1 Идентификация конфигурации

8.1.1 Установление средств для идентификации ЭЛЕМЕНТОВ КОНФИГУРАЦИИ

ИЗГОТОВИТЕЛЬ должен установить схему уникальной идентификации ЭЛЕМЕНТОВ КОНФИГУРАЦИИ и их ВЕРСИЙ с целью управления проектом. Эта схема должна включать в себя ПРОГРАММНЫЕ ПРОДУКТЫ или объекты, такие как ПОНП, а также документацию (классы *A, B, C*).

8.1.2 Идентификация ПОНП

Для каждого ЭЛЕМЕНТА КОНФИГУРАЦИИ ПОНП, который будет использоваться, включая список стандартов, ИЗГОТОВИТЕЛЬ должен документировать:

a) наименование;

b) ИЗГОТОВИТЕЛЯ;

c) уникальный указатель (обозначение) ПОНП;

d) для каждого используемого ЭЛЕМЕНТА КОНФИГУРАЦИИ 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 ЭЛЕМЕНТОВ 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 ЭЛЕМЕНТЫ КОНФИГУРАЦИИ 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. Это положение требует только того, что одобрение изменения предшествовало его выполнению.

Примечание 2 - В отношении ЗАПРОСОВ НА ИЗМЕНЕНИЯ на разных стадиях жизненного цикла могут быть использованы различные ПРОЦЕССЫ одобрения, как это установлено в планах, см. 5.1.1, перечисление e), и 6.1, перечисление 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 ЭЛЕМЕНТОВ (классы *A, B, C*).

Примечание - Этот пункт устанавливает, как должно осуществляться изменение, чтобы достигнуть соответствующего управления. Это не подразумевает, что осуществление является неотъемлемой частью ПРОЦЕССА управления. В осуществлении следует использовать запланированные ПРОЦЕССЫ, см. 5.1.1, перечисление e), и 6.1, перечисление e).

8.2.3 ВЕРИФИКАЦИЯ изменений

ИЗГОТОВИТЕЛЬ должен проверить изменения, включая повторение любой ВЕРИФИКАЦИИ, которая стала недействительной после внесения изменений, а также уделить внимание пунктам 5.7.2 и 9.7 (классы *A, B, C*).

Примечание - Этот пункт требует только того, чтобы изменения были ВЕРИФИЦИРОВАНЫ. Он не подразумевает того, что ВЕРИФИКАЦИЯ - неотъемлемая часть ПРОЦЕССА управления изменениями. ВЕРИФИКАЦИЯ должна использовать запланированные ПРОЦЕССЫ, см. 5.1.1, перечисление e), и 6.1, перечисление e).

8.2.4 Обеспечение средствами для ПРОСЛЕЖИВАЕМОСТИ изменений

ИЗГОТОВИТЕЛЬ должен создать контрольный журнал, посредством которого в отношении каждого:

a) ЗАПРОСА НА ИЗМЕНЕНИЕ,

b) соответствующего ОТЧЕТА О ПРОБЛЕМАХ,

c) одобрения ЗАПРОСА НА ИЗМЕНЕНИЕ,

d) может быть осуществлена ПРОСЛЕЖИВАЕМОСТЬ (классы *A, B, C*).

8.3 Учет статуса конфигурации

ИЗГОТОВИТЕЛЬ должен сохранять восстанавливаемые записи о истории управляемых ЭЛЕМЕНТОВ КОНФИГУРАЦИИ, включая конфигурацию СИСТЕМЫ (классы *A, B, C*).

9 Программный ПРОЦЕСС решения проблем

9.1 Подготовка ОТЧЕТОВ О ПРОБЛЕМАХ

ИЗГОТОВИТЕЛЬ должен подготовить ОТЧЕТ О ПРОБЛЕМАХ в отношении каждой проблемы, обнаруженной в ПРОГРАММНОМ ПРОДУКТЕ. ОТЧЕТЫ О ПРОБЛЕМАХ должны быть классифицированы:

a) по типу.

Пример 1 - Корректирующие, предупреждающие или в целях адаптации к новым внешним условиям;

b) масштабу.

Пример 2 - Размер изменения, число затронутых моделей устройств, затронутые поддерживаемые приспособления, вовлеченные ресурсы, время изменения;

9.2 Исследование проблемы

ИЗГОТОВИТЕЛЬ должен:

a) исследовать проблему и, если возможно, определить причины;

b) ОЦЕНИТЬ влияние проблемы на БЕЗОПАСНОСТЬ, используя ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА (см. раздел 7);

c) документировать РЕЗУЛЬТАТЫ исследования и оценки;

d) 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 программному ПРОЦЕССУ решения проблем, при условии, что проблема не является важной для 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. 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 (см. 7.4) (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:

a) была ли проблема решена и был ли завершен ОТЧЕТ О ПРОБЛЕМЕ;

b) были ли преодолены неблагоприятные тенденции;

c) был ли ЗАПРОС НА ИЗМЕНЕНИЯ реализован в соответствующих ПРОГРАММНЫХ ПРОДУКТАХ и ДЕЯТЕЛЬНОСТИ;

d) появились ли дополнительные проблемы (классы *A, B, C*).

9.8 Содержание документации по испытаниям

При испытании, повторном испытании или РЕГРЕССИОННОМ ИСПЫТАНИИ ПРОГРАММНЫХ ЭЛЕМЕНТОВ и СИСТЕМ, следующих за изменением, ИЗГОТОВИТЕЛЬ должен включить в документацию по испытаниям:

a) РЕЗУЛЬТАТЫ испытаний;

b) обнаруженные АНОМАЛИИ;

c) ВЕРСИЮ испытуемого ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ;

d) соответствующие аппаратные и ПРОГРАММНЫЕ испытуемые конфигурации;

e) испытательное оборудование и средства измерений;

f) данные по испытаниям;

g) идентификацию лица, проводившего испытания (классы *A, B, C*).

Приложение А (справочное). Логическое обоснование требований настоящего стандарта

Приложение А
(справочное)

Логическое обоснование требований настоящего стандарта приводится в настоящем приложении.

А.1 Логическое обоснование

Главным требованием настоящего стандарта является то, что совокупность ПРОЦЕССОВ, которые надлежит применять при разработке и технической поддержке ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ, и выбор ПРОЦЕССОВ должны быть приемлемы в отношении РИСКА для пациентов и других людей. Это следует из утверждения, что тестирования ПО не достаточно, чтобы определить, что оно безопасно в работе.

ПРОЦЕССЫ, требуемые настоящим стандартом, могут быть разделены на две категории:

- ПРОЦЕССЫ, которые требуются для определения РИСКА, возникающего от действия каждого ПРОГРАММНОГО ЭЛЕМЕНТА в ПО;
- ПРОЦЕССЫ, которые требуются для достижения приемлемо низкой возможности отказа ПО для каждого ПРОГРАММНОГО ЭЛЕМЕНТА, выбранного на основе этих определенных РИСКОВ.

Настоящий стандарт требует, чтобы первая категория выполнялась для любого ПО МЕДИЦИНСКИХ ИЗДЕЛИЙ, а вторая категория выполнялась только для выбранных ПРОГРАММНЫХ ЭЛЕМЕНТОВ.

Следовательно, требования соответствия настоящему стандарту должны включать в себя документированный АНАЛИЗ РИСКОВ, который идентифицирует предсказуемые последовательности событий, связанные с наличием ПО, и могущие привести к опасной ситуации (см. ИСО 14971). ОПАСНОСТЬ, которая может быть косвенно вызвана программным обеспечением (например, создавая рассогласование информации, которое может вызвать неправильную реакцию управления), должна быть включена в настоящий АНАЛИЗ РИСКОВ.

Вся деятельность, которая требуется в рамках первой категории ПРОЦЕССОВ, нормативно определена как классы *А, В, С*, указывая на то, что она необходима вне зависимости от класса БЕЗОПАСНОСТИ ПО, к которому эти классы относятся.

Эта деятельность требуется для классов *А, В* и *С* по следующим причинам:

- деятельность создает план, относящийся к МЕНЕДЖМЕНТУ РИСКА или классификации ПО по безопасности;
- деятельность дает результат, который является входными данными для МЕНЕДЖМЕНТА РИСКА или классификации БЕЗОПАСНОСТИ ПО;
- деятельность - часть МЕНЕДЖМЕНТА РИСКА или классификации БЕЗОПАСНОСТИ ПО;
- деятельность устанавливает систему управления, документации или ведения записей СИСТЕМЫ, которые поддерживают МЕНЕДЖМЕНТ РИСКА или классификацию БЕЗОПАСНОСТИ ПО;
- деятельность обычно имеет место, когда классификация связанного с ней ПО неизвестна;
- деятельность может вызвать изменение, которое может сделать недействительным класс БЕЗОПАСНОСТИ связанного с ней ПО. Эти изменения включают обнаружение и анализ БЕЗОПАСНОСТИ родственных проблем после выпуска.

Другие ПРОЦЕССЫ, требующиеся только для ПРОГРАММНЫХ СИСТЕМ или для ПРОГРАММНЫХ ЭЛЕМЕНТОВ, классифицируются по БЕЗОПАСНОСТИ как классы *В* или *С*. Деятельность, требующаяся как часть этих ПРОЦЕССОВ, указана в нормативном тексте как класс *В, С* или класс *С*, указывая на то, что она требуется выборочно, в зависимости от классификации по классу БЕЗОПАСНОСТИ ПО, к которому она применяется.

Деятельность требуется выборочно для ПО классов *В* и *С* по следующим

причинам:

- деятельность повышает надежность ПО, требуя более детального или более точного проектирования, тестирования или другой ВЕРИФИКАЦИИ;
- деятельность является управленческой, поддерживающей другую деятельность, требуемую для классов *B* и *C*;
- деятельность поддерживает коррекцию проблем, связанных с БЕЗОПАСНОСТЬЮ;
- деятельность обеспечивает ведение записей по проектированию, внедрению, ВЕРИФИКАЦИИ и выпуску ПО, связанного с БЕЗОПАСНОСТЬЮ ПО.

Деятельность требуется выборочно для ПО класса *C* по следующей причине:

- деятельность еще больше повышает надежность СИСТЕМЫ, требуя более тщательного или более точного, более внимательного отношения к отдельным вопросам проектирования, тестирования или другой ВЕРИФИКАЦИИ.

Следует отметить, что все ПРОЦЕССЫ и действия, указанные в настоящем стандарте, считаются значимыми для того, чтобы обеспечивать разработку и техническую поддержку высококачественного ПО. Упущение многих из этих ПРОЦЕССОВ и действий в качестве требований для ПО класса *A*, которое по определению не может быть причиной ОПАСНОСТИ, не подразумевает, что эти ПРОЦЕССЫ и действия не являются важными или не рекомендуются. Их пропуск нацелен на то, чтобы определить, что для ПО, которое не может быть причиной ОПАСНОСТИ, можно легко удостовериться в БЕЗОПАСНОСТИ и результативности первоначально через совокупную деятельность по валидации в течение процесса проектирования МЕДИЦИНСКИХ ИЗДЕЛИЙ (которое не входит в область применения настоящего стандарта) и через некоторые простые элементы управления жизненным циклом ПО.

А.2 Обобщение требований класса

В таблице А.1 показано, какие классы БЕЗОПАСНОСТИ ПО назначены каждому требованию ПО. Эта таблица носит справочный характер и приведена только для удобства пользователей. В нормативном разделе указаны классы БЕЗОПАСНОСТИ ПО для каждого требования.

Таблица А.1 - Обобщение требований класса ПО

Пункты и подпункты		Класс *A*	Класс *B*	Класс *C*
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		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
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

5.8	5.8.4	X	X	X
	5.8.1, 5.8.2, 5.8.3, 5.8.5, 5.8.6, 5.8.7, 5.8.8		X	X
6.1	6.1	X	X	X
6.2	6.2.1, 6.2.2, 6.2.4, 6.2.5	X	X	X
	6.2.3		X	X
6.3	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
Примечание - Знак X указывает на обязательность выполнения требования.				

Приложение В (справочное). Руководство по положениям настоящего стандарта

Приложение В
(справочное)

В.1 Обзор

В.1.1 Цель

Цель настоящего стандарта состоит в том, чтобы обеспечить ПРОЦЕСС разработки, который соответственно производит высококачественное, безопасное ПО МЕДИЦИНСКИХ ИЗДЕЛИЙ. Чтобы достигнуть этого, настоящий стандарт определяет минимально необходимые деятельность и ЗАДАЧИ, которые следует осуществить, чтобы быть уверенным в том, что ПО разработано до той степени, которая обеспечивает производство надежного и безопасного ПРОГРАММНОГО ПРОДУКТА.

Настоящее приложение обеспечивает руководство для дополнительных требований стандарта. Оно не дополняет или не изменяет требования настоящего стандарта. Настоящее приложение может быть использовано для лучшего понимания требований стандарта.

Следует отметить, что в настоящем стандарте деятельность является подпунктами, вызываемыми ПРОЦЕССАМИ, а ЗАДАЧИ определены внутри деятельности. Например, деятельность, определенная для ПРОЦЕССА разработки ПО, - это планирование разработки ПО, анализ требований ПО, проектирование АРХИТЕКТУРЫ ПО, детализированное проектирование ПО, реализация ПРОГРАММНЫХ МОДУЛЕЙ и их ВЕРИФИКАЦИЯ, интеграция ПО и тестирование интеграции, тестирование ПРОГРАММНОЙ СИСТЕМЫ и выпуск ПО. Деятельность внутри этих ЗАДАЧ определяется индивидуальными требованиями.

Настоящий стандарт не требует использования определенной МОДЕЛИ ЖИЗНЕННОГО ЦИКЛА РАЗРАБОТКИ ПО.

Однако соответствие настоящему стандарту подразумевает наличие зависимости между ПРОЦЕССАМИ, поскольку входные данные для одного ПРОЦЕССА генерируются другим ПРОЦЕССОМ. Например, классификация БЕЗОПАСНОСТИ ПО для ПРОГРАММНОЙ СИСТЕМЫ должна быть завершена после того, как ПРОЦЕСС АНАЛИЗА РИСКОВ установит, какой ВРЕД приносит отказ ПРОГРАММНОЙ СИСТЕМЫ.

Из-за таких логических зависимостей между процессами, легче всего описывать ПРОЦЕССЫ в этом стандарте в последовательности, подразумевающей модель жизненного цикла "водопад" (прямоточная). Однако также могут быть использованы и другие жизненные циклы. Некоторые стратегии разработки (модели), как определено в ИСО/МЭК 12207 [9], включают в себя (см. также таблицу В.1):

- "Водопад" - прямоточная стратегия, также называемая "водопад", состоит в выполнении ПРОЦЕССА разработки один раз. Проще говоря, следует определить нужды заказчика, определить требования, спроектировать СИСТЕМУ, исполнить ее, протестировать, установить и поставить заказчику.

- "Пошаговая" - это стратегия определяет нужды заказчика и определяет требования СИСТЕМЫ, затем осуществляет остаток разработки 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.

- "Эволюционная" - это стратегия также развивает СИСТЕМУ в сборке, но в отличие от пошаговой стратегии признает, 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 В.1 - Разработка стратегий (моделей), как это определено в ИСО/МЭК 12207

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

Какой бы жизненный цикл не 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 жизненные циклы, разрешают ПРОЦЕССУ вырабатывать выходные данные до того, как будут доступны все входные данные для этого процесса. Например, новый ПРОГРАММНЫЙ ЭЛЕМЕНТ может быть определен, классифицирован, исполнен и ВЕРИФИЦИРОВАН до того, как будет закончена программная АРХИТЕКТУРА в целом. Такие жизненные циклы несут в себе РИСК того, что изменение или разработка выходных данных одного процесса сделает недействительными выходные данные другого ПРОЦЕССА. При этом все жизненные циклы используют комплексную систему управления конфигурацией, чтобы убедиться, что выходные данные всех ПРОЦЕССОВ доведены 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 быстро 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 ПО 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 ПО МЕДИЦИНСКИХ ИЗДЕЛИЙ, а также для разработки и технической поддержки МЕДИЦИНСКИХ ИЗДЕЛИЙ, которые содержат ПОНП.

Использование настоящего стандарта требует от изготовителя выполнять МЕНЕДЖМЕНТ РИСКА МЕДИЦИНСКИХ ИЗДЕЛИЙ, как это указано в ИСО 14971. Следовательно, когда АРХИТЕКТУРА СИСТЕМЫ МЕДИЦИНСКИХ ИЗДЕЛИЙ включает в себя 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 исполнения МЕНЕДЖМЕНТА РИСКА МЕДИЦИНСКИХ ИЗДЕЛИЙ ИЗГОТОВИТЕЛЬ идентифицирует этот компонент и признает, что он содержит ПОНП. ИЗГОТОВИТЕЛЬ, использующий настоящий стандарт, должен ввести ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА ПО как часть ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА МЕДИЦИНСКОГО ИЗДЕЛИЯ.

Техническая поддержка выпущенного ПО МЕДИЦИНСКИХ ИЗДЕЛИЙ относится к постпроизводственному опыту работы с ПО МЕДИЦИНСКИХ ИЗДЕЛИЙ. Техническая поддержка ПО включает в себя сочетание всех технических и управленческих средств, в т.ч. действия контроля, чтобы реагировать на ОТЧЕТ О ПРОБЛЕМАХ, сохраняя элемент или восстанавливая его до состояния, в котором он может осуществлять требуемые функции, а также запросы на модификацию, относящуюся к выпущенному ПРОГРАММНОМУ ПРОДУКТУ (ПРОДУКТАМ). Например, это включает исправление проблемы, регламентированную отчетность, повторные проверки и профилактические действия. См. ИСО/МЭК 14764 [10].

B.2 Нормативные ссылки

ИСО/МЭК 90003 [11] представляет собой руководство для применения систем менеджмента качества к разработке ПО. Это руководство не требуется настоящим стандартом, но рекомендуется.

B.3 Термины и определения

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

Настоящий стандарт выбирает для использования три термина, чтобы описывать декомпозицию ПРОГРАММНОЙ СИСТЕМЫ (верхний уровень). ПРОГРАММНАЯ СИСТЕМА может быть разделена на подсистемы МЕДИЦИНСКОГО ИЗДЕЛИЯ (см. [2]) или МЕДИЦИНСКОЕ ИЗДЕЛИЕ само по себе. Самый нижний уровень, ниже которого дальнейшее разложение на составные части не осуществляется, для целей тестирования или управления конфигурацией ПО - это ПРОГРАММНЫЙ МОДУЛЬ. Все уровни композиции, включая верхний и нижний уровни, могут быть названы ПРОГРАММНЫМИ ЭЛЕМЕНТАМИ. ПРОГРАММНАЯ СИСТЕМА состоит из одного или более ПРОГРАММНЫХ ЭЛЕМЕНТОВ, и каждый ПРОГРАММНЫЙ ЭЛЕМЕНТ состоит из ПРОГРАММНЫХ МОДУЛЕЙ или разложимых ПРОГРАММНЫХ ЭЛЕМЕНТОВ. Ответственность за обеспечение определения и степени детализации ПРОГРАММНЫХ ЭЛЕМЕНТОВ и ПРОГРАММНЫХ МОДУЛЕЙ возложена на ИЗГОТОВИТЕЛЯ. То, что эти термины остаются неопределенными, допускает их применение ко многим разным методам разработки и типам ПО, используемым в МЕДИЦИНСКИХ ИЗДЕЛИЯХ.

B.4 Общие требования

Не существует метода, чтобы обеспечить 100% БЕЗОПАСНОСТЬ для ПО любого класса.

Существует три главных принципа, которые способствуют БЕЗОПАСНОСТИ ПО МЕДИЦИНСКИХ ИЗДЕЛИЙ:

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

Для разработки и технической поддержки безопасного ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ необходимо установить МЕНЕДЖМЕНТ РИСКА как неотъемлемую часть СИСТЕМЫ менеджмента качества, как общий каркас для приложения соответствующих методов и техник проектирования ПО. Комбинация из этих трех понятий позволяет ИЗГОТОВИТЕЛЮ МЕДИЦИНСКИХ ИЗДЕЛИЙ следовать ясно структурированному и последовательно повторяемому ПРОЦЕССУ принятия решений, чтобы способствовать БЕЗОПАСНОСТИ ПО МЕДИЦИНСКИХ ИЗДЕЛИЙ.

B.4.1 Система менеджмента качества

Дисциплинированная и эффективная совокупность ПРОЦЕССОВ ПО включает в себя организационные ПРОЦЕССЫ, такие как управление, инфраструктура, улучшение и подготовка. Чтобы избежать дублирования и сфокусировать настоящий стандарт на проектировании, в настоящем стандарте эти ПРОЦЕССЫ не указаны.

Эти ПРОЦЕССЫ описаны в СИСТЕМЕ менеджмента качества ИСО 13485 [7] - это международный стандарт, который специально предназначен для применения понятий менеджмента качества к МЕДИЦИНСКИМ ИЗДЕЛИЯМ. Соответствие требованиям системы менеджмента качества ИСО 13485 не означает автоматического соответствия национальным или местным нормативным требованиям. На ответственности ИЗГОТОВИТЕЛЯ остается идентификация и установление соблюдения соответствующим нормативным актам.

B.4.2 МЕНЕДЖМЕНТ РИСКА

Включение разработки ПО в деятельность по МЕНЕДЖМЕНТУ РИСКА является достаточным, чтобы убедиться, что все обоснованно прогнозируемые РИСКИ, связанные с ПО МЕДИЦИНСКИХ ИЗДЕЛИЙ, учтены.

Вместо того, чтобы пытаться определить подходящий ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА, в настоящем стандарте по разработке ПО требуется, чтобы изготовитель применил ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА, совместимый с ИСО 14971, который непосредственно относится к МЕНЕДЖМЕНТУ РИСКА для МЕДИЦИНСКИХ ИЗДЕЛИЙ. Определенная деятельность по МЕНЕДЖМЕНТУ РИСКА ПО, вытекающая из ОПАСНОСТЕЙ, относящихся к ПО, способствующих РИСКУ, определена во вспомогательном ПРОЦЕССЕ, описанном в разделе 7.

В.4.3 Классификация БЕЗОПАСНОСТИ ПО

РИСК, связанный с ПО, как с частью МЕДИЦИНСКОГО ИЗДЕЛИЯ, как с приложением к МЕДИЦИНСКОМУ ИЗДЕЛИЮ или как с самостоятельным МЕДИЦИНСКИМ ИЗДЕЛИЕМ, используется в качестве входных данных для схемы классификации БЕЗОПАСНОСТИ ПО, которая после этого определяет ПРОЦЕССЫ, используемые во время разработки и технической поддержки ПО.

РИСК считается комбинацией тяжести вреда и вероятности его возникновения. Однако нет никакого единого мнения по вопросу о том, как определить вероятность возникновения отказа ПО, используя традиционные статистические методы. Поэтому в настоящем стандарте классификация ПРОГРАММНЫХ СИСТЕМ основана на тяжести опасности, проистекающей от отказа ПО, предполагая, что отказ обязательно произойдет. ПРОГРАММНЫЕ СИСТЕМЫ, которые способствуют выполнению мер по УПРАВЛЕНИЮ РИСКОМ, классифицируют на основе степени тяжести ОПАСНОСТИ, которую они контролируют.

Если ПРОГРАММНАЯ СИСТЕМА делится на ПРОГРАММНЫЕ ЭЛЕМЕНТЫ, тогда каждый ПРОГРАММНЫЙ ЭЛЕМЕНТ может иметь свой собственный класс БЕЗОПАСНОСТИ ПО. Можно определить РИСК, связанный с отказом ПРОГРАММНОГО ЭЛЕМЕНТА, только:

- если СИСТЕМНАЯ АРХИТЕКТУРА и АРХИТЕКТУРА ПО определяют роль ПРОГРАММНОГО ЭЛЕМЕНТА с точки зрения его цели и взаимодействия с другими программными и аппаратными элементами;
- если изменения в СИСТЕМЕ контролируются;
- после того, как был выполнен АНАЛИЗ РИСКОВ для данной АРХИТЕКТУРЫ, и определены меры по УПРАВЛЕНИЮ РИСКОМ.

Настоящий стандарт требует минимального количества действий, которые направлены на достижение вышеупомянутых условий для ПО всех классов.

Окончание деятельности по построению АРХИТЕКТУРЫ ПО - самая первая точка в разработке, когда определен полный набор ПРОГРАММНЫХ ЭЛЕМЕНТОВ, и деятельность по МЕНЕДЖМЕНТУ РИСКА уже определила, как ПРОГРАММНЫЕ ЭЛЕМЕНТЫ влияют на БЕЗОПАСНОСТЬ. Следовательно, это самая первая точка, на которой ПРОГРАММНЫЕ ЭЛЕМЕНТЫ могут быть окончательно классифицированы согласно их роли относительно БЕЗОПАСНОСТИ.

Эта точка соответствует точке, где по ИСО 14971 начинается УПРАВЛЕНИЕ РИСКОМ.

До этой точки ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА определяет АРХИТЕКТУРНЫЕ меры по УПРАВЛЕНИЮ РИСКОМ, например, добавляя защитные подсистемы или уменьшая возможность возникновения ВРЕДА из-за отказа в работе ПО. После этой точки ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА использует ПРОЦЕССЫ, направленные на уменьшение вероятности отказа в работе ПРОГРАММНЫХ ЭЛЕМЕНТОВ. Другими словами, классификация ПРОГРАММНОГО ЭЛЕМЕНТА определяет основанные на ПРОЦЕССНОМ ПОДХОДЕ меры по УПРАВЛЕНИЮ РИСКОМ, которые следует применить к этому элементу.

Ожидается, что ИЗГОТОВИТЕЛЬ посчитает полезным классифицировать ПО до данной точки, например, чтобы сфокусировать внимание на областях, которые нужно исследовать, но такая классификация должна считаться предварительной и не должна использоваться для обоснования пропуска ПРОЦЕССОВ.

Схема классификации БЕЗОПАСНОСТИ ПО не предназначена для согласования с классификацией РИСКОВ по ИСО 14971. Если в ИСО 14971 классификация РИСКОВ осуществляется относительно их тяжести и вероятности возникновения, то схема классификации БЕЗОПАСНОСТИ ПО разделяет ПРОГРАММНЫЕ СИСТЕМЫ и ПРОГРАММНЫЕ ЭЛЕМЕНТЫ в соответствии с ПРОЦЕССАМИ, которые будут применены для разработки и технической поддержки.

По мере развития проекта могут стать очевидными новые РИСКИ. Следовательно, МЕНЕДЖМЕНТ РИСКА следует применять как неотъемлемую часть ПРОЦЕССА разработки. Это допускает разработку АРХИТЕКТУРНОГО проекта, определяющего полный набор ПРОГРАММНЫХ ЭЛЕМЕНТОВ, включая те, от которых требуется правильное функционирование, чтобы обеспечить безопасную работу, и те, которые предотвращают причинение ВРЕДА из-за отказов в работе.

АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ должна способствовать изоляции ПРОГРАММНЫХ ЭЛЕМЕНТОВ, которые требуются для безопасной работы, и должна описывать методы, используемые для обеспечения эффективного разделения этих ПРОГРАММНЫХ ЭЛЕМЕНТОВ.

Как установлено в В.3, настоящий стандарт выбирает для применения три термина, чтобы описывать декомпозицию ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СИСТЕМЫ (верхний уровень).

На рисунке В.1 проиллюстрировано возможное разделение ПРОГРАММНЫХ ЭЛЕМЕНТОВ внутри ПРОГРАММНОЙ СИСТЕМЫ и как классы БЕЗОПАСНОСТИ ПО могут быть применены к группе ПРОГРАММНЫХ ЭЛЕМЕНТОВ в декомпозиции.

Рисунок В.1 - Пример разделения ПРОГРАММНЫХ ЭЛЕМЕНТОВ

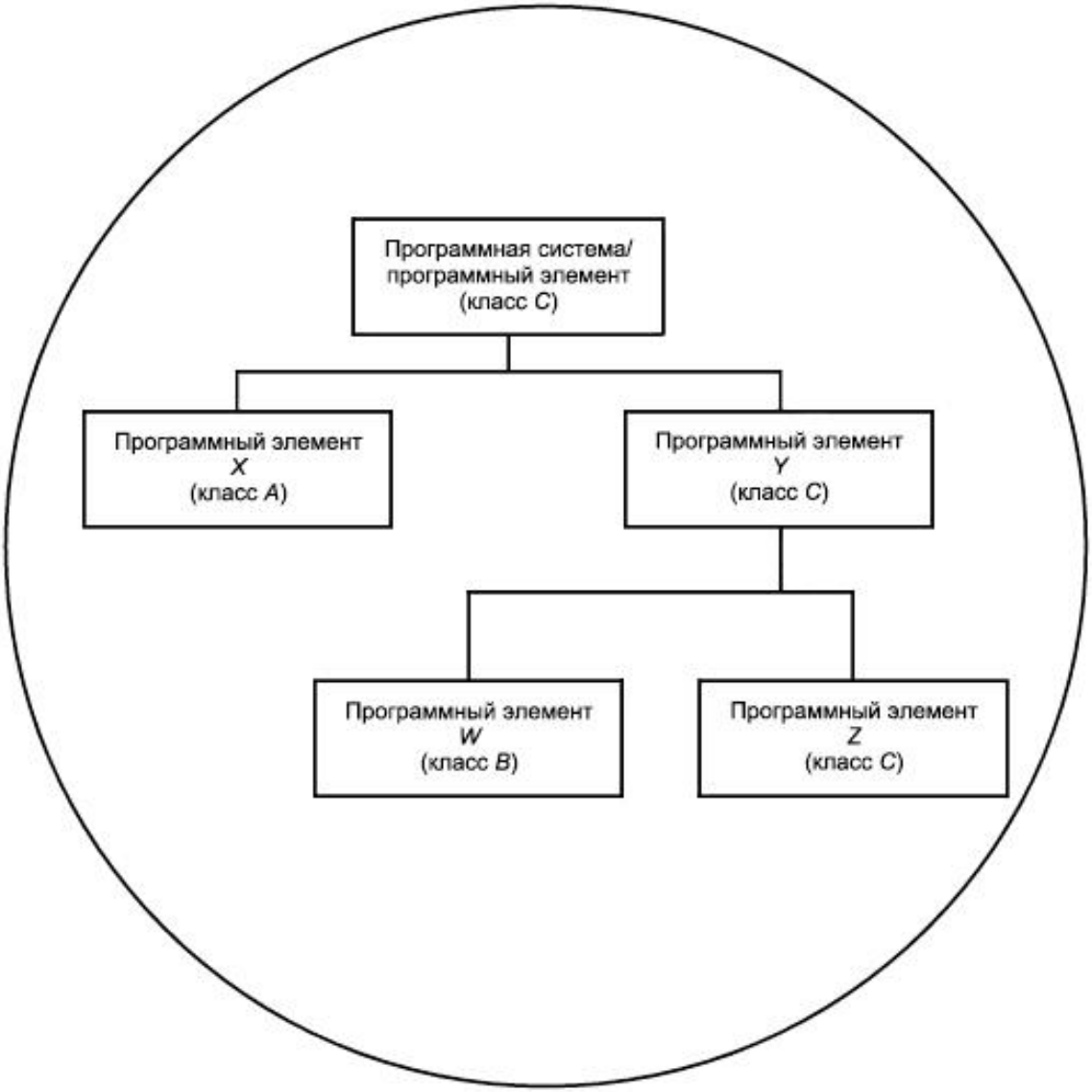

Рисунок В.1 - Пример разделения ПРОГРАММНЫХ ЭЛЕМЕНТОВ

Например, ИЗГОТОВИТЕЛЬ знает благодаря типу разрабатываемого ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ, что по предварительной классификации БЕЗОПАСНОСТЬ ПО для ПРОГРАММНОЙ СИСТЕМЫ относится к классу *C* по БЕЗОПАСНОСТИ ПО. Во время проектирования программной АРХИТЕКТУРЫ изготовитель решает разделить СИСТЕМУ, как это показано, на три программных уровня - *X, W* и *Z*. ИЗГОТОВИТЕЛЬ может выделить все вклады ПРОГРАММНОЙ СИСТЕМЫ в ОПАСНОСТИ, которые могут привести к смерти или СЕРЬЕЗНОМУ УЩЕРБУ, в программный уровень *Z*, а все оставшиеся вклады ПРОГРАММНОЙ СИСТЕМЫ в ОПАСНОСТИ, которые приводят к НЕСЕРЬЕЗНОМУ УЩЕРБУ, в программный уровень *W*. Программный уровень *W* получает класс БЕЗОПАСНОСТИ ПО *B*, программный уровень *Z* - класс БЕЗОПАСНОСТИ ПО *C*. Программный уровень *Y*, следовательно, должен быть отнесен к классу *C*, по 4.3, перечисление g). В соответствии с этим требованием ПРОГРАММНАЯ СИСТЕМА также получает программный класс БЕЗОПАСНОСТИ *C*. Программный уровень *X* относится к программному классу БЕЗОПАСНОСТИ *A*. ИЗГОТОВИТЕЛЬ может документально подтвердить разумное объяснение для разделения программных уровней *X* и *Y* так же, как и для программных уровней *W* и *Z*, чтобы обеспечить целостность разделения. Если разделение не является возможным, программные уровни *X* и *Y* должны быть отнесены к программному классу БЕЗОПАСНОСТИ *C*.

B.5 ПРОЦЕСС разработки ПО

B.5.1 Планирование разработки ПО

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

Деятельность по планированию разработки ПО и задач может быть документирована в едином плане или в различных планах. Некоторые ИЗГОТОВИТЕЛИ могут устанавливать политики и процедуры, которые применяются к разработке всего их ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ. В этом случае план может просто ссылаться на существующие политики и процедуры. Некоторые ИЗГОТОВИТЕЛИ могут подготовить план или набор планов для разработки каждого ПРОГРАММНОГО ПРОДУКТА МЕДИЦИНСКИХ ИЗДЕЛИЙ, который влечет за собой детально определенные действия и ссылается на общие процедуры. Другая возможность состоит в том, что план или набор планов приспособлен для разработки каждого ПРОГРАММНОГО ПРОДУКТА МЕДИЦИНСКИХ ИЗДЕЛИЙ. Планирование следует определять на уровне детализации, необходимой для осуществления ПРОЦЕССА разработки, и оно должно быть пропорционально РИСКУ. Например, СИСТЕМЫ или элементы с более высокой степенью РИСКА должны быть подчинены ПРОЦЕССУ разработки с более строгими требованиями, а задачи должны быть изложены более детально.

Планирование является итеративной РАБОТОЙ, 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 ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СИСТЕМЫ может измениться в результате осуществления ПРОЦЕССА УПРАВЛЕНИЯ РИСКОМ и развития АРХИТЕКТУРЫ ПО. Также может быть принято решение о включении ПОНП в СИСТЕМУ. Важно, чтобы план (планы) был обновлен, чтобы отражать текущие знания о СИСТЕМЕ и уровне точности, необходимом для СИСТЕМЫ или элементов, и чтобы сделать возможным надлежащий контроль над ПРОЦЕССОМ разработки.

B.5.2 Анализ требований к ПО

Эта деятельность требует от ИЗГОТОВИТЕЛЯ установить и верифицировать ПРОГРАММНЫЕ требования для ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ. Установление верифицируемых требований существенно для определения того, что должно быть создано, для определения, что ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ ведет себя приемлемым образом, и для демонстрации, что завершенное ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ готово к использованию. Чтобы продемонстрировать, что требования были осуществлены согласно замыслу, каждое требование должно быть указано таким образом, чтобы можно было установить объективные критерии для проверки того, правильно ли оно осуществлено. Если ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА изделия диктует требования к ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ для УПРАВЛЕНИЯ выявленными РИСКАМИ, эти требования должны быть идентифицированы в программных требованиях таким образом, чтобы сделать возможным прослеживание мер по УПРАВЛЕНИЮ РИСКОМ до программных требований. Все ПРОГРАММНЫЕ требования 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, это соответствие требованиям должно быть документировано в программных требованиях. Поскольку ПРОГРАММНЫЕ требования устанавливают, что должно быть реализовано в ПО, оценка требований требуется до завершения деятельности по анализу требований.

Областью частых недоразумений является различие между потребностями заказчика, входными данными проектирования, программными требованиями, функциональными спецификациями ПО и спецификациями проекта ПО. Входные данные проектирования являются преобразованием потребностей заказчика в официально документированные требования к МЕДИЦИНСКОМУ ИЗДЕЛИЮ. ПРОГРАММНЫЕ требования - это официально документированные спецификации того, что ПО отвечает требованиям заказчика и входным данным проектирования. Функциональные спецификации ПО часто включены в программные требования и определяют в деталях, что ПО выполняет, чтобы соответствовать этим требованиям, даже если другие альтернативные варианты могут также соответствовать этим требованиям. Спецификации проекта ПО определяют, как ПО будет спроектировано и разложено на составные части, чтобы осуществить эти требования и функциональные спецификации.

Традиционно программные требования, функциональные спецификации и спецификации проекта записаны как набор из одного и более документов. В настоящее время возможно рассматривать эту информацию 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, тестеров, аудиторов) и поддерживает ПРОСЛЕЖИВАЕМОСТЬ, чтобы демонстрировать соответствующее выполнение тестовых заданий проверяемым требованиям до установленной степени. 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 ПО).

ПРОЦЕСС определения требований к СИСТЕМЕ находится вне области применения настоящего стандарта. Однако решение о внедрении функциональных МЕДИЦИНСКИХ ИЗДЕЛИЙ с ПО обычно осуществляется во время проектирования СИСТЕМЫ. Некоторые или все требования к СИСТЕМЕ выделяются, чтобы быть осуществленными в ПО. Анализ требований к программному обеспечению заключается в анализе требований, выделяемых на программное обеспечение ПРОЦЕССОМ определения требований к СИСТЕМЕ, и в получении полного набора требований к ПО, отражающих выделенные требования.

Чтобы гарантировать целостность СИСТЕМЫ, ИЗГОТОВИТЕЛЬ должен обеспечить механизм для ведения переговоров об изменениях и уточнения требований СИСТЕМЫ, чтобы исправлять непрактичность, несоответствия или двусмысленности в требованиях любой родительской СИСТЕМЫ или в программных требованиях.

Процесс охвата и анализа СИСТЕМЫ и требований ПО может быть итеративным.

Настоящий стандарт не требует, чтобы ПРОЦЕССЫ были строго разделены на два слоя. На практике СИСТЕМНАЯ АРХИТЕКТУРА и АРХИТЕКТУРА ПО часто обрисовываются в общих чертах одновременно, и СИСТЕМНЫЕ и программные требования впоследствии документируются в форме слоев.

B.5.3 Проектирование АРХИТЕКТУРЫ ПО

Эта деятельность требует, чтобы ИЗГОТОВИТЕЛЬ определил главные структурные компоненты ПО, их внешне видимые свойства и взаимосвязь между ними. Если поведение компонента может оказывать влияние на другие компоненты, то такое поведение должно быть описано в АРХИТЕКТУРЕ ПО. Это описание особенно важно для поведения, которое может затронуть компоненты МЕДИЦИНСКОГО ИЗДЕЛИЯ, находящиеся вне ПО. АРХИТЕКТУРНЫЕ решения чрезвычайно важны для осуществления мер по УПРАВЛЕНИЮ РИСКАМИ. Без понимания и документирования поведения компонента, которое может оказать влияние на другие компоненты, почти невозможно доказать, что СИСТЕМА безопасна. АРХИТЕКТУРА ПО необходима, чтобы гарантировать правильное выполнение программных требований. АРХИТЕКТУРА ПО не считается завершенной, если все требования ПО не могут быть осуществлены определенными ЭЛЕМЕНТАМИ ПО. Поскольку проект и выполнение ПО зависят от АРХИТЕКТУРЫ, АРХИТЕКТУРА ВЕРИФИЦИРУЕТСЯ для завершения этой деятельности. Как правило, ВЕРИФИКАЦИЮ АРХИТЕКТУРЫ выполняют путем технического ОЦЕНИВАНИЯ.

Классификация ПРОГРАММНЫХ ЭЛЕМЕНТОВ во время активности АРХИТЕКТУРЫ ПО создает основание для последующего выбора ПО ПРОЦЕССОВ. Записи о классификации находятся в составе ФАЙЛА МЕНЕДЖМЕНТА РИСКА.

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

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

Поэтому, во время всех видов деятельности, следующих за проектом АРХИТЕКТУРЫ ПО, классификация ПРОГРАММНЫХ СИСТЕМ и ПРОГРАММНЫХ ЭЛЕМЕНТОВ должна быть ПЕРЕОЦЕНЕНА и, если необходимо, пересмотрена. Это вызывает доработку с применением дополнительных ПРОЦЕССОВ к отдельному ЭЛЕМЕНТУ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ в результате его модернизации до более высокого класса. ПРОЦЕСС менеджмента конфигурации ПО (см. раздел 8) используется для обеспечения уверенности в том, что все необходимые доработки были идентифицированы и завершены.

В.5.4 Детализация проекта ПО

Эта деятельность требует от ИЗГОТОВИТЕЛЯ усовершенствовать ПРОГРАММНЫЕ ЭЛЕМЕНТЫ и интерфейсы, определенные в АРХИТЕКТУРЕ, чтобы создать ПРОГРАММНЫЕ МОДУЛИ и их интерфейсы. Хотя ПРОГРАММНЫЕ МОДУЛИ часто считаются единичными функциями или модулями, эта точка зрения не всегда является приемлемой. ПРОГРАММНЫЙ модуль был определен как ПРОГРАММНЫЙ ЭЛЕМЕНТ, не делимый на более мелкие элементы. ПРОГРАММНЫЕ МОДУЛИ могут быть проверены отдельно. ИЗГОТОВИТЕЛЮ следует определить уровень детализации ПРОГРАММНОГО МОДУЛЯ. Детализированный проект определяет алгоритмы представления данных и взаимодействия между ПРОГРАММНЫМИ МОДУЛЯМИ и структурами данных. Детализированный проект также должен касаться и формирования ПРОГРАММНОГО ПАКЕТА. Необходимо документировать проект для каждого ПРОГРАММНОГО МОДУЛЯ и его интерфейса так, чтобы ПРОГРАММНЫЙ МОДУЛЬ мог быть реализован правильно. Детализированный проект заполняет детали, необходимые для проектирования ПО. Он должен быть достаточно полным, чтобы программисту не требовалось принимать специальных проектных решений.

ПРОГРАММНЫЕ ЭЛЕМЕНТЫ могут быть разложены на уровни так, что только немногие из новых ПРОГРАММНЫХ ЭЛЕМЕНТОВ выполняют связанные с БЕЗОПАСНОСТЬЮ требования оригинального ПРОГРАММНОГО ЭЛЕМЕНТА. Оставшиеся ПРОГРАММНЫЕ ЭЛЕМЕНТЫ не осуществляют связанных с БЕЗОПАСНОСТЬЮ функций и могут быть повторно классифицированы с присвоением более низкого класса БЕЗОПАСНОСТИ ПО. Однако принятие такого решения само по себе является частью ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА и документируется в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА.

Поскольку реализация зависит от детализированного проекта, необходимо верифицировать детализированный проект до завершения деятельности. ВЕРИФИКАЦИЮ детализированного проекта, как правило, осуществляют путем технического ОЦЕНИВАНИЯ. 5.4.4 требует от ИЗГОТОВИТЕЛЯ ВЕРИФИЦИРОВАТЬ выходные данные деятельности по детализированному проектированию. Проект определяет, какие требования должны быть реализованы. Если проект будет содержать дефекты, то код не будет правильно осуществлять требования.

Если это имеет место в проекте, ИЗГОТОВИТЕЛЬ должен проверить характеристики проекта, которые считает важными для БЕЗОПАСНОСТИ. Примеры таких характеристик включают:

- осуществление намеченных событий, входных и выходных данных, интерфейсов, логической схемы, распределения ресурсов процессора, распределения ресурсов памяти, ошибок и исключений изоляции и устранения ошибок;
- определение состояния по умолчанию, в котором все отказы, могущие привести к возникновению опасной ситуации, охвачены, включая события и переходы;
- инициализация переменных, управление памятью; и
- "холодная" и "теплая" перезагрузки, режим ожидания и другие изменения состояния, которые могут оказать влияние на меры по УПРАВЛЕНИЮ РИСКОМ.

B.5.5 Реализация и ВЕРИФИКАЦИЯ ПРОГРАММНОГО МОДУЛЯ

Эта деятельность требует от ИЗГОТОВИТЕЛЯ записать и проверить код для ПРОГРАММНЫХ МОДУЛЕЙ.

Детализированный проект преобразовывается в исходный код. Кодирование представляет собой момент, в котором заканчивается декомпозиция спецификаций и начинается составление реализуемого 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, чтобы убедиться, что он функционирует, как определено в детализированном проекте, и что он соответствует указанным стандартам кодирования.

В 5.5.5 от ИЗГОТОВИТЕЛЯ требуется проверять код. Если код не реализует проект правильно, ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ не будет работать так, как предполагалось.

В.5.6 Программная интеграция и тестирование интеграции

Эта деятельность требует от ИЗГОТОВИТЕЛЯ планировать и реализовывать интеграцию ПРОГРАММНЫХ МОДУЛЕЙ в составные ПРОГРАММНЫЕ ЭЛЕМЕНТЫ так же, как и интеграцию ПРОГРАММНЫХ ЭЛЕМЕНТОВ в более сложносоставные ПРОГРАММНЫЕ ЭЛЕМЕНТЫ, и проверять, что полученный в результате ПРОГРАММНЫЙ ЭЛЕМЕНТ функционирует так, как предназначено.

Подход к интеграции может колебаться от непошаговой интеграции до любой формы пошаговой интеграции. Свойства собираемого ПРОГРАММНОГО ЭЛЕМЕНТА диктуют выбираемый метод интеграции.

Тестирование интеграции ПО направлено на передачу данных и управление всего ПРОГРАММНОГО ЭЛЕМЕНТА через внешние и внутренние интерфейсы. Внешние интерфейсы - те, которые имеют другое ПО, включая работающее системное ПО 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 ПО для потенциально опасных функций и с ролью определенных ПРОГРАММНЫХ ЭЛЕМЕНТОВ в функциях изделия с большей степенью РИСКА. Например, несмотря на то, что все ПРОГРАММНЫЕ ЭЛЕМЕНТЫ должны быть протестированы, элементы, которые влияют на БЕЗОПАСНОСТЬ, следует подвергать более точным, тщательным и подробным тестам.

В соответствующих случаях тестирование интеграции демонстрирует поведение программы на границах ее входных и выходных доменов (областей) и подтверждает ПРОГРАММНЫЕ реакции на недействительные, неожиданные и специальные входные данные. Действия программы обнаруживаются при введении комбинации входных данных, 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, в которой используется точное знание внутренней работы ТЕСТИРУЕМОГО ЭЛЕМЕНТА, чтобы выбирать данные тестирования. Тестирование "белого ящика" использует определенные знания о ПРОГРАММНОМ ЭЛЕМЕНТЕ, чтобы проверять выходные данные. Это тестирование является точным только в том случае, если тестер "знает", что ПРОГРАММНЫЙ ЭЛЕМЕНТ должен делать. Тогда тестер может видеть, когда ПРОГРАММНЫЙ ЭЛЕМЕНТ отклоняется от его намеченной цели. Тестирование методом "белого ящика" не может гарантировать, что была осуществлена полная спецификация, поскольку оно сфокусировано на тестировании реализации ПРОГРАММНОГО ЭЛЕМЕНТА. Тестирование "черного ящика", также известное как поведенческое, функциональное, тестирование непрозрачного ящика или тестирование закрытого ящика, сфокусировано на тестировании функциональной спецификации и не может гарантировать, что были протестированы все реализованные части. Таким образом, тестирование "черного ящика" является тестированием на спецификацию и обнаруживает дефекты пропусков, определяя, какая часть спецификации не была выполнена. Тестирование "белого ящика" является тестированием на реализацию и обнаруживает дефекты выполнения, указывая, какая часть реализации неисправна. Чтобы полностью

протестировать ПРОГРАММНЫЙ ПРОДУКТ, могут потребоваться как тестирование "черного ящика", так и тестирование "белого ящика".

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

Тестирование интеграции ПО может быть осуществлено в моделируемой среде, на имеющемся оборудовании, или на полном МЕДИЦИНСКОМ ИЗДЕЛИИ.

В 5.6.2 от ИЗГОТОВИТЕЛЯ требуется ВЕРИФИЦИРОВАТЬ выходные данные деятельности по интеграции ПО. Выходные данные деятельности интеграции ПО - это интегрированные (встроенные) ПРОГРАММНЫЕ ЭЛЕМЕНТЫ.

Эти интегрированные ПРОГРАММНЫЕ ЭЛЕМЕНТЫ должны правильно функционировать для всего ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ, чтобы оно функционировало правильно и надежно.

B.5.7 Тестирование ПО СИСТЕМЫ

Эта РАБОТА требует от ИЗГОТОВИТЕЛЯ проверить функциональность ПО путем проверки того, что требования к ПО были успешно выполнены.

Тестирование ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СИСТЕМЫ демонстрирует, что указанные функции существуют. Это тестирование ВЕРИФИЦИРУЕТ функциональность и характеристики программы как встроенной в отношении требований к ПО. Тестирование ПРОГРАММНОЙ СИСТЕМЫ ориентировано на функциональное ("черный ящик") тестирование, хотя может быть желательно использовать метод "белого ящика" (см. выше), чтобы эффективно выполнять определенные тесты, выделять стрессовые условия или дефекты, или увеличивать охват кодом квалификационных тестов. Организация тестирования по типам и этапам теста является гибкой, но охват требований, УПРАВЛЕНИЕ РИСКОМ, эксплуатационная пригодность и типы тестов (например, отказ, установка, работа в сложных ситуациях) должны быть продемонстрированы и задокументированы.

Тестирование ПРОГРАММНОЙ СИСТЕМЫ проверяет интегрированное ПО и может быть выполнено в моделируемой среде, на имеющемся оборудовании или на полном МЕДИЦИНСКОМ ИЗДЕЛИИ.

Когда в ПРОГРАММНУЮ СИСТЕМУ вносят изменения (даже небольшие), должна быть определена степень РЕГРЕССИОННОГО ТЕСТИРОВАНИЯ (но не только тестирования отдельных изменений), чтобы удостовериться, что не были введены никакие непредусмотренные побочные эффекты. Это РЕГРЕССИОННОЕ ТЕСТИРОВАНИЕ (и обоснование для неполностью повторяемого тестирования ПРОГРАММНОЙ СИСТЕМЫ) должно быть запланировано и документировано.

Ответственность за тестирование ПРОГРАММНОЙ СИСТЕМЫ может быть распределена, осуществляться в разных местах и быть проведена различными организациями. Однако, независимо от распределения ЗАДАЧ, договорных отношений, источника компонентов или среды (условий) разработки, ИЗГОТОВИТЕЛЬ изделия сохраняет окончательную ответственность за обеспечение правильного функционирования ПО в соответствии с предназначенным применением.

Если АНОМАЛИИ, обнаруженные в течение тестирования, могут повториться, но было принято решение не фиксировать их, тогда эти АНОМАЛИИ должны быть ОЦЕНЕНЫ в отношении анализа ОПАСНОСТИ, чтобы убедиться, что они не оказывают влияние на БЕЗОПАСНОСТЬ изделия. Причина и признаки АНОМАЛИЙ должны быть поняты, и должно быть в наличии документированное объяснение того, что эти АНОМАЛИИ не фиксируются.

В 5.7.4 требуется, чтобы РЕЗУЛЬТАТЫ тестирования ПРОГРАММНОЙ СИСТЕМЫ были ОЦЕНЕНЫ для обеспечения получения ожидаемых РЕЗУЛЬТАТОВ.

B.5.8 Выпуск ПО

Эта деятельность требует от ИЗГОТОВИТЕЛЯ документировать ВЕРСИЮ выпускаемого ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ, указать, как оно было создано, и следовать необходимым для выпуска ПО процедурам.

ИЗГОТОВИТЕЛЬ должен быть способен продемонстрировать, что ПО, которое было разработано, с использованием ПРОЦЕССА разработки, - это то ПО, которое было выпущено. ИЗГОТОВИТЕЛЬ должен иметь возможность восстановить ПО и инструменты, использованные для его создания, в случае, если это будет необходимо в будущем, и хранить, упаковывать и доставлять ПО 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 ПО отличается от ПРОЦЕССА разработки ПО по двум следующим пунктам:

- ИЗГОТОВИТЕЛЮ разрешается использовать ПРОЦЕСС меньший, чем полный ПРОЦЕСС разработки ПО, 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 требования локальными регулирующими актами (обычно запуская активную схему наблюдения для сбора данных о проблеме и ее области и для общения с пользователями и регулирующими органами о проблеме).

6.1 требует, чтобы эти ПРОЦЕССЫ были установлены в плане обслуживания.

Эта деятельность требует от ИЗГОТОВИТЕЛЯ создания или идентификации процедуры для реализации деятельности и ЗАДАЧ по технической поддержке. Чтобы выполнять корректирующие действия, управлять изменениями во время технической поддержки и управлять выпуском предусмотренного ПО, ИЗГОТОВИТЕЛЮ следует документировать и решить требуемые проблемы и запросы потребителей, а также управлять модификациями ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ. Этот ПРОЦЕСС активизируется, когда ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ подвергается модификациям кода или сопутствующей документации из-за проблем или потребности в улучшении или адаптации. Цель состоит в том, чтобы модифицировать выпущенное ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ, сохраняя его целостность. Этот ПРОЦЕСС включает помещение ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ во внешнюю среду или на платформы, для которых он не был первоначально приспособлен. Деятельность, предусмотренная настоящим пунктом, характерна для ПРОЦЕССА технической поддержки; однако ПРОЦЕСС технической поддержки может использовать другие ПРОЦЕССЫ, указанные в настоящем стандарте.

ИЗГОТОВИТЕЛЮ нужно планировать деятельность и ЗАДАЧИ ПРОЦЕССА технической поддержки.

B.6.2 Анализ проблем и модификаций

Данная деятельность требует от ИЗГОТОВИТЕЛЯ анализировать последствия обратной связи; проверять сообщения о проблемах и рассматривать, выбирать и одобрять подходящие для выполнения возможные варианты модификаций.

Проблемы и другие ЗАПРОСЫ НА ИЗМЕНЕНИЯ могут оказывать влияние на исполнение, БЕЗОПАСНОСТЬ и оформление МЕДИЦИНСКИХ ИЗДЕЛИЙ регулирующими органами. Анализ необходим, чтобы определить, существуют ли какие-нибудь последствия в результате СООБЩЕНИЯ О ПРОБЛЕМАХ и появятся ли какие-нибудь последствия из-за модификаций, чтобы устранить проблему или реализовать запрос. Для проверки посредством анализа кривых или регрессионного анализа особенно важно, чтобы встроенные в изделие меры по УПРАВЛЕНИЮ РИСКОМ не были изменены или модифицированы программным изменением, которое осуществляется как часть деятельности по технической поддержке. Также важно убедиться, что модифицированное ПО не вызывает опасности или смягчает РИСК. Классификация БЕЗОПАСНОСТИ ПО ПРОГРАММНОГО ЭЛЕМЕНТА может быть изменена, если модификация ПО в настоящий момент может вызывать ОПАСНОСТЬ или уменьшать РИСК.

Важно различать техническую поддержку ПО (см. раздел 6) и решение проблем ПО (см. раздел 9).

Главным в ПРОЦЕССЕ технической поддержки ПО является достаточный ответ на обратную связь, возникающую после выпуска ПРОГРАММНОГО ПРОДУКТА. В рамках МЕДИЦИНСКОГО ИЗДЕЛИЯ ПРОЦЕССУ технической поддержки ПО следует обеспечить уверенность в том, что:

- ОТЧЕТЫ О ПРОБЛЕМАХ, связанные с БЕЗОПАСНОСТЬЮ, адресуются регулирующим органам и заинтересованным потребителям и доводятся до них;
- ПРОГРАММНЫЕ ПРОДУКТЫ повторно одобряются и выпускаются после модификации и официального контроля, которые обеспечивают устранение проблемы и предотвращение дальнейших проблем;
- ИЗГОТОВИТЕЛЬ рассматривает, какие другие ПРОГРАММНЫЕ ПРОДУКТЫ могут быть затронуты, и предпринимает соответствующие действия.

Особое внимание при решении проблем ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ должно быть уделено функционированию комплексной СИСТЕМЫ управления, которая:

- анализирует ОТЧЕТЫ О ПРОБЛЕМАХ и идентифицирует все последствия этой проблемы;
- принимает решения по ряду изменений и определяет все их побочные эффекты;
- осуществляет изменения, сохраняя при этом согласованность ПРОГРАММНЫХ ЭЛЕМЕНТОВ КОНФИГУРАЦИИ, в том числе ФАЙЛА МЕНЕДЖМЕНТА РИСКА;
- ВЕРИФИЦИРУЕТ осуществление изменений.

Процесс технической поддержки ПО использует ПРОЦЕСС решения проблем ПО. Отчет о проблемах считается высшим уровнем в ПРОЦЕССЕ технической поддержки ПО (существует ли проблема, оказывает ли она существенное влияние на БЕЗОПАСНОСТЬ, какие изменения необходимы и когда осуществлять их), который использует ПРОЦЕСС решения проблем ПО для анализа ОТЧЕТА О ПРОБЛЕМАХ, чтобы обнаружить все последствия, а также для создания возможных ЗАПРОСОВ НА ИЗМЕНЕНИЯ, которые идентифицируют все ЭЛЕМЕНТЫ КОНФИГУРАЦИИ, требующие изменения, а

также 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 определен, для осуществления модификации могут быть использованы 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. Должно быть сделано обоснование, оправдывающее число РЕГРЕССИОННЫХ ТЕСТОВ, которые будут выполнены для обеспечения уверенности в том, что части еще не модифицированного ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ продолжают работать так, как и до выполнения модификации.

B.7 ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА ПО

МЕНЕДЖМЕНТ РИСКА ПО - это часть полного МЕНЕДЖМЕНТА РИСКА МЕДИЦИНСКОГО ИЗДЕЛИЯ, и не может надлежащим образом быть рассмотрена отдельно. Настоящий стандарт требует использования ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА, совместимого с ИСО 14971. Как определено в ИСО 14971, МЕНЕДЖМЕНТ РИСКА представляет собой основу для результативного МЕНЕДЖМЕНТА РИСКА в отношении МЕДИЦИНСКИХ ИЗДЕЛИЙ. Один из разделов ИСО 14971 относится к УПРАВЛЕНИЮ идентифицированными РИСКАМИ, связанными с каждой ОПАСНОСТЬЮ, выявленной в ходе АНАЛИЗА РИСКА. ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА ПО в настоящем стандарте предназначен для установления дополнительных требований к УПРАВЛЕНИЮ РИСКОМ для ПО, включая ПО, которое было определено в ходе АНАЛИЗА РИСКОВ как потенциально способствующее возникновению опасных ситуаций, или ПО, которое используется для УПРАВЛЕНИЯ РИСКОМ МЕДИЦИНСКОГО ИЗДЕЛИЯ. ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА ПО включен в настоящий стандарт по двум следующим причинам:

a) целевая аудитория настоящего стандарта должна осознавать минимальные требования для мер по УПРАВЛЕНИЮ РИСКОМ в ПРОГРАММНОМ ОБЕСПЕЧЕНИИ, являющемся зоной их ответственности;

b) общий стандарт по МЕНЕДЖМЕНТУ РИСКА, ИСО 14971, приведенный в настоящем стандарте в качестве нормативной ссылки, не охватывает специально УПРАВЛЕНИЕ РИСКОМ ПО и место УПРАВЛЕНИЯ РИСКОМ в жизненном цикле разработки ПО.

МЕНЕДЖМЕНТ РИСКА ПО - это часть общего МЕНЕДЖМЕНТА РИСКА МЕДИЦИНСКОГО ИЗДЕЛИЯ. Планы, процедуры и документация, требуемые для деятельности по МЕНЕДЖМЕНТУ РИСКА ПО, могут быть серией отдельных документов или одним документом, или могут быть интегрированы в деятельность и в документацию по МЕНЕДЖМЕНТУ РИСКА МЕДИЦИНСКИХ ИЗДЕЛИЙ при условии, что все требования настоящего стандарта выполняются.

B.7.1 Анализ ПО, способствующего возникновению опасных ситуаций

Ожидается, что анализ ОПАСНОСТИ изделия будет определять опасные ситуации и соответствующие меры УПРАВЛЕНИЯ РИСКОМ для уменьшения вероятности и/или тяжести вреда этих опасных ситуаций до приемлемого уровня. Также предполагается, что меры по УПРАВЛЕНИЮ РИСКОМ будут возложены на ПРОГРАММНЫЕ функции, которые, как ожидается, будут выполнять эти меры по УПРАВЛЕНИЮ РИСКОМ.

Однако вряд ли можно ожидать, что все опасные ситуации изделия могут быть определены до того, как будет подготовлена программная АРХИТЕКТУРА. В то же время известно, как функции ПО 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 ПО;
- новые опасные ситуации, возникающие из-за ПО, например, опасные ситуации, связанные с человеческим фактором.

АРХИТЕКТУРА ПО должна включать в себя надежные стратегии для разделения компонентов ПО таким образом, чтобы минимизировать их опасное взаимодействие.

B.8 ПРОЦЕСС менеджмента конфигурации ПО

ПРОЦЕСС менеджмента конфигурации ПО - это ПРОЦЕСС применения административных и технических процедур на протяжении жизненного цикла ПО для идентификации и определения ПРОГРАММНЫХ ЭЛЕМЕНТОВ, включая документацию, в СИСТЕМЕ, управление изменениями и выпуском элементов, документирование и сообщение о состоянии элементов и ЗАПРОСОВ НА ИЗМЕНЕНИЯ. Управление конфигурацией ПО необходимо, чтобы обновить ПРОГРАММНЫЙ ЭЛЕМЕНТ, 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 ЭЛЕМЕНТОВ КОНФИГУРАЦИИ ПО и их ВЕРСИЙ. Эта идентификация необходима, чтобы определять ЭЛЕМЕНТЫ КОНФИГУРАЦИИ ПО и ВЕРСИИ, которые включены в ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ.

B.8.2 Управление изменениями

Данная деятельность требует от ИЗГОТОВИТЕЛЯ управлять изменениями ЭЛЕМЕНТОВ КОНФИГУРАЦИИ ПО 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 ЭЛЕМЕНТЫ КОНФИГУРАЦИИ ПО и что одобренные ЗАПРОСЫ НА ИЗМЕНЕНИЯ были полностью осуществлены и ВЕРИФИЦИРОВАНЫ.

ЗАПРОСЫ НА ИЗМЕНЕНИЯ могут быть одобрены советом по управлению изменениями, менеджером или техническим руководством согласно плану управления конфигурацией ПО. Обеспечивается ПРОСЛЕЖИВАЕМОСТЬ одобренных ЗАПРОСОВ НА ИЗМЕНЕНИЯ до фактической модификации и ВЕРИФИКАЦИИ ПО. Необходимо, чтобы каждое фактическое изменение было связано с ЗАПРОСОМ НА ИЗМЕНЕНИЕ и чтобы была в наличии документация, показывающая, что ЗАПРОС НА ИЗМЕНЕНИЕ был одобрен. Документация может быть изменена протоколом совета управления изменениями, подтвержденным подписью или записью в базе данных.

B.8.3 Учет статуса конфигурации

Данная деятельность требует от ИЗГОТОВИТЕЛЯ поддерживать записи истории ЭЛЕМЕНТОВ КОНФИГУРАЦИИ ПО. Эта работа необходима, чтобы определять, когда и где были сделаны изменения. Доступ к этой информации необходим для обеспечения уверенности в том, что ЭЛЕМЕНТЫ КОНФИГУРАЦИИ ПО содержат только разрешенные модификации.

B.9 ПРОЦЕСС решения проблем ПО

ПРОЦЕСС решения проблем ПО - это ПРОЦЕСС для анализа и решения проблем (включая несоответствия), вне зависимости от их природы или источника, включая те, которые обнаружены во время выполнения ПРОЦЕССОВ разработки, технической поддержки и других. Цель состоит в том, чтобы обеспечивать своевременные, ответственные и документально подтвержденные средства обеспечения того, что обнаруженные проблемы анализируются и решаются и что тенденции замечены. Данный ПРОЦЕСС в литературе, относящейся к разработке ПО, иногда называется "отслеживание дефекта". В ИСО/МЭК 12207 [9] и МЭК 60601-1-4 [2], поправка 1, он назван "решение проблем". Для настоящего стандарта было принято решение называть ПРОЦЕСС "решение проблем ПО".

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

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

Приложение С (справочное). Взаимосвязь с

другими стандартами

Приложение С
(справочное)

С.1 Общие положения

Настоящий стандарт применяется к разработке и технической поддержке ПО МЕДИЦИНСКИХ ИЗДЕЛИЙ. ПО считается подсистемой МЕДИЦИНСКОГО ИЗДЕЛИЯ или самостоятельным МЕДИЦИНСКИМ ИЗДЕЛИЕМ. Настоящий стандарт предназначен для использования совместно с другими подходящими стандартами в ПРОЦЕССЕ разработки МЕДИЦИНСКИХ ИЗДЕЛИЙ.

Стандарты по менеджменту МЕДИЦИНСКИХ ИЗДЕЛИЙ, такие как ИСО 13485 [7] (см. С.2 и приложение D) и ИСО 14971 обеспечивают менеджмент окружающей среды, что закладывает основу для организации разработки продукции. Стандарты БЕЗОПАСНОСТИ, такие как МЭК 60601-1 [1] (см. С.4) и МЭК 61010-1 [4] (см. С.5), дают определенное руководство по созданию безопасных МЕДИЦИНСКИХ ИЗДЕЛИЙ. Когда ПО является составной частью МЕДИЦИНСКИХ ИЗДЕЛИЙ, настоящий стандарт предлагает более детальное руководство относительно требований к разработке и поддержанию БЕЗОПАСНОСТИ ПО МЕДИЦИНСКИХ ИЗДЕЛИЙ. Многие другие стандарты, такие как ИСО/МЭК 12207 [9] (см. С.6), МЭК 61508-3 [3] (см. С.7) и ИСО/МЭК 90003 [11], могут рассматриваться как источники методов, инструментов и техник, которые могут быть использованы, чтобы осуществить требования настоящего стандарта. На рисунке С.1 показано отношение между этими стандартами.

Там, где цитируются положения или требования других стандартов, используемые термины в цитируемых элементах являются терминами, которые определены в других стандартах и не определены в настоящем стандарте.

Рисунок С.1 - Взаимосвязь ключевых стандартов на МЕДИЦИНСКИЕ ИЗДЕЛИЯ с настоящим стандартом

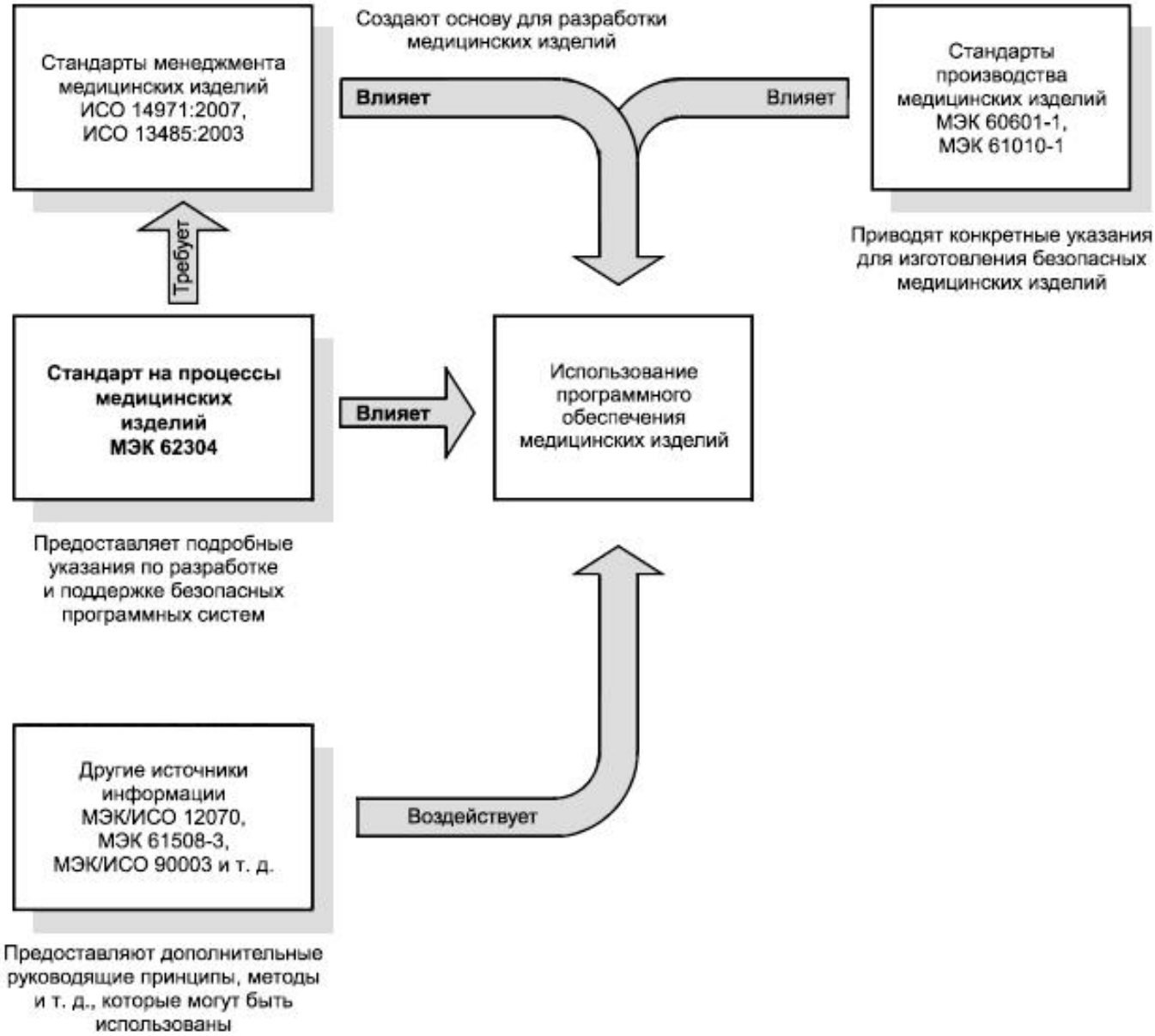

Рисунок С.1 - Взаимосвязь ключевых стандартов на МЕДИЦИНСКИЕ ИЗДЕЛИЯ с настоящим стандартом

С.2 Взаимосвязь с ИСО 13485

Настоящий стандарт требует, чтобы ИЗГОТОВИТЕЛЬ использовал систему менеджмента качества. Когда ИЗГОТОВИТЕЛЬ использует ИСО 13485 [7], требования настоящего стандарта непосредственно связаны с требованиями ИСО 13485, как это показано в таблице С.1.

Таблица С.1 - Взаимосвязь с ИСО 13485:2003

Раздел/пункт настоящего стандарта	Соответствующий пункт ИСО 13485:2003
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 Детализированная разработка ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	-
5.5 Исполнение и проверка ПРОГРАММНЫХ МОДУЛЕЙ	-
5.6 Программная интеграция и испытания в отношении интеграции	-
5.7 Испытания СИСТЕМЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	7.3.3 Выходные данные проектирования и разработки 7.3.4 Анализ проекта и разработки
5.8 Выпуск ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	7.3.5 ВЕРИФИКАЦИЯ проекта и разработки 7.3.6 Валидация проекта 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 проекта и разработки 7.3.6 Валидация проекта и разработки
7.1 Анализ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, способствующего возникновению опасных ситуаций	-
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 Программный ПРОЦЕСС решения проблем	-

С.3 Взаимосвязь с ИСО 14971

В таблице С.2 показаны области, где настоящий стандарт усиливает требования к ПРОЦЕССУ МЕНЕДЖМЕНТА РИСКА, требуемого ИСО 14971.

Таблица С.2 - Взаимосвязь с ИСО 14971:2007

Раздел/пункт ИСО 14971:2007	Соответствующий пункт настоящего стандарта
4.1 Процедура анализа РИСКА	-
4.2 Предусмотренное применение/ предусмотренное назначение и определение характеристик, относящихся к БЕЗОПАСНОСТИ медицинского изделия	-
4.3 Идентификация известных или прогнозируемых ОПАСНОСТЕЙ	7.1 Анализ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, способствующего возникновению опасных ситуаций
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	7.2.1 Выбор мер по УПРАВЛЕНИЮ РИСКОМ
6.3 Выполнение мер по УПРАВЛЕНИЮ РИСКОМ	7.2.2 Меры по УПРАВЛЕНИЮ РИСКОМ, осуществляемые в ПРОГРАММНОМ ОБЕСПЕЧЕНИИ 7.3.1 Проверка мер по УПРАВЛЕНИЮ РИСКОМ
6.4 Оценивание остаточного РИСКА	-
6.5 Анализ соотношения риск/польза	-

6.6 Другие возможные опасности	7.3.2 Документирование любых новых последовательностей событий
6.7 Полнота оценивания РИСКА	-
7 ОЦЕНИВАНИЕ полного остаточного РИСКА	-
8 Отчет по МЕНЕДЖМЕНТУ РИСКА	7.3.3 Документирование ПРОСЛЕЖИВАЕМОСТИ
9 Постпроизводственная информация	7.4 МЕНЕДЖМЕНТ РИСКА в отношении изменений ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

С.4 Взаимосвязь ПЭМС с требованиями МЭК 60601-1:2005

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 МЭК 60601-1 [1] для ПЭМС. Поскольку ПЭМС включают в себя элементы, не являющиеся ПО, не все требования МЭК 60601-1 для ПЭМС изложены в настоящем стандарте.

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 ПО до интеграции ПРОГРАММНЫХ ЭЛЕМЕНТОВ в ПРОГРАММНУЮ СИСТЕМУ. Эта ПРОГРАММНАЯ СИСТЕМА - часть программируемой электрической подсистемы (ПЭСС), являющейся в свою очередь частью ПЭМС.

Рисунок С.2 - ПО как часть V-модели

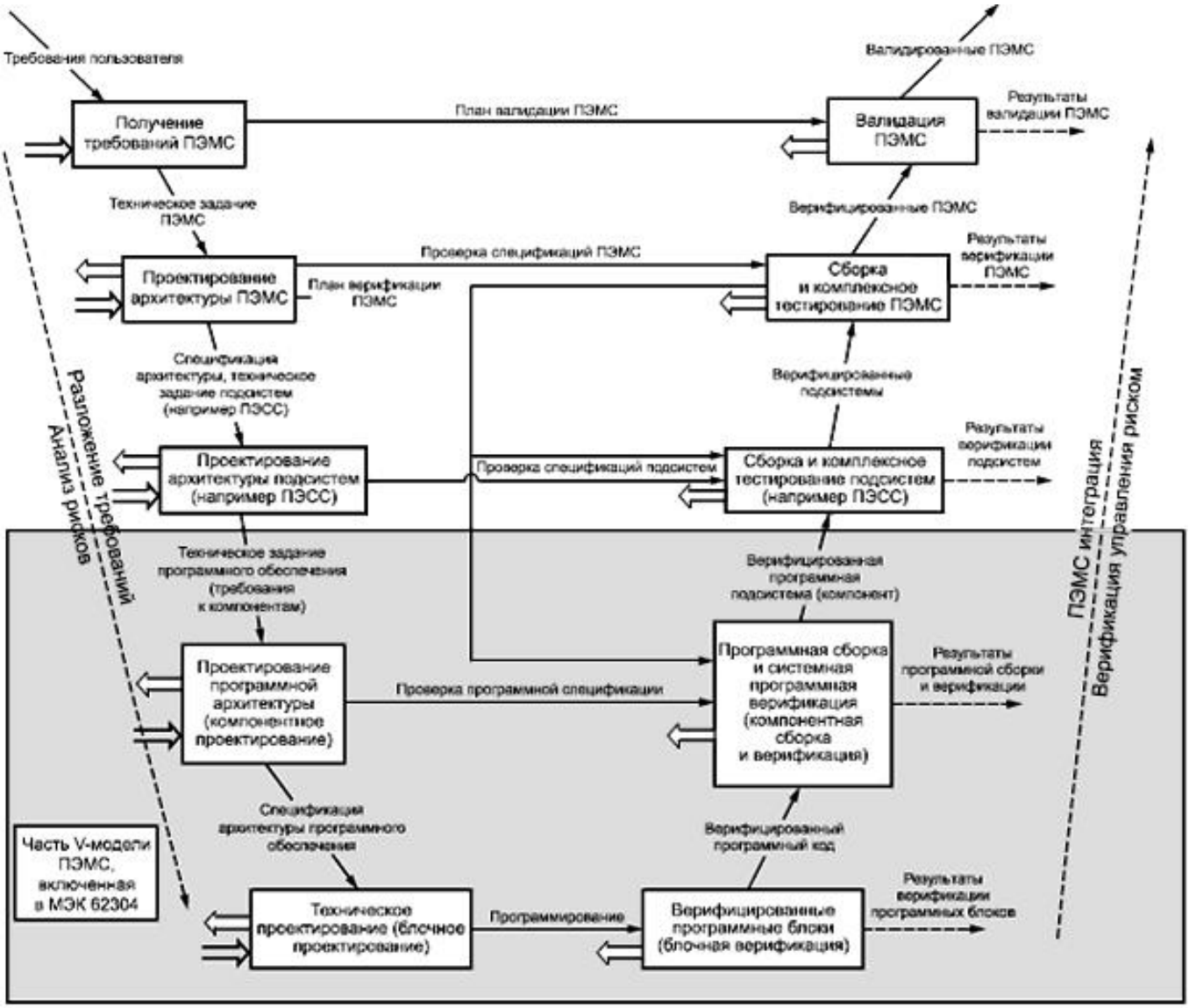

Рисунок С.2 - ПО как часть V-модели

С.4.3 ПРОЦЕСС разработки

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

С.4.4 ПРОЦЕСС технического обслуживания

Соответствие ПРОЦЕССУ технического обслуживания ПО в настоящем стандарте (раздел 6) требует, чтобы процедуры были установлены и соблюдались, когда в ПО вносятся изменения. Эти требования соответствуют требованиям МЭК 60601-1 для модификации ПЭМС. Требования настоящего стандарта относительно технического обслуживания ПО предоставляют более подробную информацию о том, что должно быть сделано для технической поддержки ПО, чем требования для модификации ПЭМС в МЭК 60601-1.

С.4.5 Прочие ПРОЦЕССЫ

Прочие ПРОЦЕССЫ в настоящем стандарте определяют дополнительные требования к ПО, сверх подобных требований для ПЭМС в МЭК 60601-1. В большинстве случаев существует общее требование для ПЭМС в МЭК 60601-1, которое расширяет ПРОЦЕССЫ, указанные в настоящем стандарте.

ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА ПО в настоящем стандарте соответствует дополнительным требованиям МЕНЕДЖМЕНТА РИСКА, определенными для ПЭМС в МЭК 60601-1.

ПРОЦЕСС решения проблем ПО в настоящем стандарте соответствует требованию к решению проблем для ПЭМС в МЭК 60601-1.

ПРОЦЕСС менеджмента конфигурации ПО в настоящем стандарте определяет дополнительные требования, которые не представлены для ПЭМС в МЭК 60601-1, за исключением документации.

С.4.6 Охват требований к ПЭМС в МЭК 60601-1

В таблице С.3 приведены требования к ПЭМС в МЭК 60601-1 и соответствующие им требования в настоящем стандарте.

Таблица С.3 - Взаимосвязь с МЭК 60601-1

Требования к ПЭМС в МЭК 60601-1:2005	Требования настоящего стандарта, связанные с программным обеспечением подсистемы ПЭМС
14* Программируемые электрические медицинские СИСТЕМЫ (ПЭМС) 14.1* Общие положения Требования данного пункта должны применяться к ПЭМС, за исключением случаев, когда: - программируемая электронная подсистема (ПЭСС) не задействована в обеспечении ОСНОВНОЙ БЕЗОПАСНОСТИ или ОСНОВНЫХ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК, или - применение ИСО 14971:2007 показывает, что отказ ПЭСС не приводит к возникновению недопустимого РИСКА	4.3 Классификация ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ в отношении безопасности Требования к ПЭМС, установленные в МЭК 60601-1, применимы только к программному обеспечению классов безопасности *B* и *C*. Настоящий стандарт содержит некоторые требования в отношении программного обеспечения класса безопасности *A*
14.2* Документирование В дополнение к ЗАПИСЯМ и документам, требуемым ИСО 14971:2007, разработанные при применении раздела 14 документы должны сохраняться в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА, составляя его часть	4.2 МЕНЕДЖМЕНТ РИСКА

Документы, требуемые в соответствии с разделом 14, должны рассматриваться, утверждаться, выпускаться и изменяться в соответствии с документированной процедурой управления документацией	5.1 Планирование разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В дополнение к определенным требованиям в отношении ДЕЯТЕЛЬНОСТИ по планированию разработки программного обеспечения, документы, которые являются частью ФАЙЛА МЕНЕДЖМЕНТА РИСКА, должны поддерживаться в соответствии с ИСО 14971:2007. Кроме того, в соответствии с ИСО 13485:2003, требуется управление в отношении документации системы качества
14.3* План МЕНЕДЖМЕНТА РИСКА План МЕНЕДЖМЕНТА РИСКА, требуемый согласно ИСО 14971:2007, пункт 3.5, должен также включать ссылку на план ПРОВЕРКИ СООТВЕТСТВИЯ ПЭМС (см. 14.11)	Специальные требования отсутствуют. Не существует никакого определенного плана валидации программного обеспечения. План валидации ПЭМС относится к уровню СИСТЕМЫ и, таким образом, находится вне области применения настоящего стандарта на программное обеспечение. Настоящий стандарт требует ПРОСЛЕЖИВАЕМОСТИ от ОПАСНОСТИ до определения причин возникновения опасности, связанных с программным обеспечением, мер по УПРАВЛЕНИЮ РИСКОМ и до ВЕРИФИКАЦИИ мер по УПРАВЛЕНИЮ РИСКОМ (см. 7.3)

14.4* ЖИЗНЕННЫЙ ЦИКЛ РАЗРАБОТКИ ПЭМС ЖИЗНЕННЫЙ ЦИКЛ РАЗРАБОТКИ ПЭМС должен быть документально оформлен	5.1 Планирование разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 5.1.1 План разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ Пункты, на которые ссылается план разработки программного обеспечения, составляют разработку жизненного цикла программного обеспечения
ЖИЗНЕННЫЙ ЦИКЛ РАЗРАБОТКИ ПЭМС должен состоять из набора определенных этапов	-
На каждом этапе должны быть определены работы, которые должны быть завершены, а также методы ВЕРИФИКАЦИИ, которые должны применяться в отношении этих работ	5.1.6 Планирование ВЕРИФИКАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ Должны быть запланированы ЗАДАЧИ ВЕРИФИКАЦИИ, этапы и критерии приемки
Каждая работа должна быть определена с указанием входных и выходных характеристик	5.1.1 План разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ Вся ДЕЯТЕЛЬНОСТЬ определена в настоящем стандарте. Документация, которая должна быть разработана, определена для каждого вида ДЕЯТЕЛЬНОСТИ

На каждом этапе должны быть определены 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 ПО должна быть документирована в плане разработки. Это означает, что план разработки должен содержать разработку конкретного жизненного цикла
ЖИЗНЕННЫЙ ЦИКЛ РАЗРАБОТКИ ПЭМС должен быть составлен 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 планирования

_____

* Текст документа соответствует оригиналу. - Примечание изготовителя базы данных.

14.5* Решение проблем Когда это целесообразно, должна разрабатываться и поддерживаться зарегистрированная система решения проблем, возникающих на каждом этапе (и между ними) ЖИЗНЕННОГО ЦИКЛА РАЗРАБОТКИ ПЭМС	9 Программный ПРОЦЕСС решения проблем

В зависимости от типа изделия 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
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 СЕТЕВЫМИ/ИНФОРМАЦИОННЫМИ СРЕДСТВАМИ СВЯЗИ, компонентами сторонних изготовителей и стандартными подсистемами	7.1 Анализ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, способствующего возникновению опасных ситуаций Настоящий стандарт не дает ссылки на конкретное сопряжение сети и данных

14.6.2* УПРАВЛЕНИЕ РИСКОМ Нижеследующие требования к ПЭМС дополняют требования, содержащиеся в ИСО 14971:2007, подраздел 6.1. Для выполнения каждого мероприятия по УПРАВЛЕНИЮ РИСКОМ должны быть выбраны и идентифицированы соответствующим образом обоснованные методы и ПРОЦЕДУРЫ. Эти методы и ПРОЦЕДУРЫ должны быть пригодны для гарантии того, что каждое мероприятие по УПРАВЛЕНИЮ РИСКОМ уменьшает идентифицированные РИСКИ в достаточной степени	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 этой системе или подсистеме	5.2.1 Определение и документирование требований к ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ в зависимости 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 не устанавливает, чтобы требования, связанные с эксплуатационными характеристиками и мерами по УПРАВЛЕНИЮ РИСКОМ были отличными от других требований, однако устанавливает, чтобы все требования были идентифицированы уникальным образом
14.8* АРХИТЕКТУРА Для любой ПЭМС и каждой из ее подсистем должна быть определена АРХИТЕКТУРА, удовлетворяющая перечню требований	5.3 Проектирование АРХИТЕКТУРЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Когда это целесообразно для снижения риска до приемлемого уровня, в требованиях к структуре ПЭМС должны использоваться: a) компоненты с высокой степенью интеграции; b) устойчивые к отказам функции; c) избыточность; d) диверсификация; e) разделение функций;	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
f) защищенная структура, служащая, например, для ограничения представляющих потенциальную опасность эффектов путем ограничения допускаемой выходной мощности или введения устройств, ограничивающих свободный ход исполнительных устройств	
Структура ПЭМС должна также учитывать: a) распределение мероприятий по управлению риском в подсистемах и компонентах ПЭМС; b) режимы (состояния) 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 сетевым/информационным системам связи	Не включено в настоящий стандарт

14.9* Проектирование и реализация ПЭМС Когда это целесообразно, проектирование следует проводить для отдельных подсистем, каждая из которых должна иметь собственные требования к разработке и требования к испытаниям	5.4 Детализированная разработка ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 5.4.2 Разработка детализированного проекта для каждого ПРОГРАММНОГО МОДУЛЯ
Пояснения относительно условий проектирования должны включаться в ФАЙЛ МЕНЕДЖМЕНТА РИСКА	5.4.2 Разработка детализированного проекта для каждого ПРОГРАММНОГО МОДУЛЯ
14.10* ВЕРИФИКАЦИЯ ВЕРИФИКАЦИЯ требуется для всех функций, которые обеспечивают ОСНОВНУЮ БЕЗОПАСНОСТЬ, ОСНОВНЫЕ ФУНКЦИОНАЛЬНЫЕ ХАРАКТЕРИСТИКИ ИЛИ УПРАВЛЕНИЕ РИСКОМ	5.1.6 Планирование ВЕРИФИКАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ВЕРИФИКАЦИЯ требуется в отношении любого вида 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. Это требование считается установленным в ИСО 13485
ВЕРИФИКАЦИЯ должна выполняться в соответствии с планом верификации, а ее результаты должны документироваться	Требования ВЕРИФИКАЦИИ установлены к большинству видов ДЕЯТЕЛЬНОСТИ

14.11* ПРОВЕРКА СООТВЕТСТВИЯ (ВАЛИДАЦИЯ) ПЭМС План ПРОВЕРКИ СООТВЕТСТВИЯ (ВАЛИДАЦИИ) ПЭМС должен включать проверку ОСНОВНОЙ БЕЗОПАСНОСТИ И ОСНОВНЫХ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК, а также проверки на непредусмотренное функционирование ПЭМС	Настоящий стандарт не распространяется на валидацию программного обеспечения. ВАЛИДАЦИЯ ПЭМС является ДЕЯТЕЛЬНОСТЬЮ на уровне СИСТЕМЫ и находится вне области применения настоящего стандарта
ПРОВЕРКА СООТВЕТСТВИЯ (ВАЛИДАЦИЯ) ПЭМС должна выполняться в соответствии с планом ВАЛИДАЦИИ, а ее РЕЗУЛЬТАТЫ должны документироваться	Настоящий стандарт не распространяется на валидацию программного обеспечения. ВАЛИДАЦИЯ ПЭМС является ДЕЯТЕЛЬНОСТЬЮ на уровне СИСТЕМЫ и находится вне области применения настоящего стандарта
Лицо, несущее основную ответственность за ПРОВЕРКУ СООТВЕТСТВИЯ (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
Ни один из членов коллектива разработчиков ПЭМС не должен нести ответственность за процесс ПРОВЕРКИ СООТВЕТСТВИЯ (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 ПРОВЕРКЕ СООТВЕТСТВИЯ (ВАЛИДАЦИИ) ПЭМС, и членами коллектива разработчиков ПЭМС должны быть зарегистрированы 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 быть включена 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 ПРОЦЕСС технической поддержки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ Настоящий стандарт устанавливает, что обслуживание программного обеспечения должно быть запланировано, а реализация модификаций должна использовать ПРОЦЕСС разработки программного обеспечения или установленный ПРОЦЕСС обслуживания программного обеспечения

14.13* Соединение ПЭМС с другим изделием с помощью СЕТЕВЫХ/ИНФОРМАЦИОННЫХ СРЕДСТВ СВЯЗИ Если ПЭМС предназначена для соединения с помощью СЕТЕВЫХ/ИНФОРМАЦИОННЫХ СРЕДСТВ СВЯЗИ с другим изделием, которое не может контролироваться изготовителем ПЭМС, то в техническом описании: a) должны быть указаны характеристики СЕТЕВЫХ/ИНФОРМАЦИОННЫХ СРЕДСТВ СВЯЗИ, необходимые для предусмотренного применения ПЭМС; b) должен содержаться перечень ОПАСНЫХ СИТУАЦИЙ, возникающих из-за отказов СЕТЕВЫХ/ИНФОРМАЦИОННЫХ СРЕДСТВ СВЯЗИ, связанных с обеспечением их установленных характеристик; c) должны содержаться указания ОТВЕТСТВЕННОЙ ОРГАНИЗАЦИИ относительно того, что: 1) соединение ПЭМС с СЕТЕВЫМИ/ИНФОРМАЦИОННЫМИ СРЕДСТВАМИ СВЯЗИ, которое осуществляется с использованием другого оборудования, может приводить к ранее непредусмотренным РИСКАМ ДЛЯ ПАЦИЕНТОВ, ОПЕРАТОРОВ или третьих лиц; 2) ОТВЕТСТВЕННАЯ ОРГАНИЗАЦИЯ должна идентифицировать, анализировать, ОЦЕНИВАТЬ эти РИСКИ и управлять ими;	Требования в отношении сопряжения сети и данных не включены в настоящий стандарт	

3) последующие изменения СЕТЕВЫХ/ИНФОРМАЦИОННЫХ СРЕДСТВ СВЯЗИ могут приводить к появлению новых РИСКОВ и требовать дополнительного анализа; 4) последующие изменения СЕТЕВЫХ/ИНФОРМАЦИОННЫХ СРЕДСТВ СВЯЗИ могут включать: 1) изменения в их конфигурации; 2) подсоединение к ним дополнительных элементов; 3) отсоединение от них отдельных элементов; 4) модификацию соединенного с ними изделия; 5) модернизацию соединенного с ними изделия	

С.4.7 Взаимосвязь с МЭК 60601-1-4

МЭК 60601-1-4 будет продолжать использоваться до тех пор, пока переходный период для МЭК 60601-1 не будет завершен.

Таблица С.4 показывает требования МЭК 60601-1-4 [2] и соответствующие требования в настоящем стандарте. Это не свидетельствует о том, что соответствующие требования в настоящем стандарте полностью охватывают требования в МЭК 60601-1-4. Многие части требований МЭК 60601-1-4 охватываются путем соблюдения требований стандарта ИСО 14971. Некоторые требования МЭК 60601-1-4 не относятся к настоящему стандарту.

Таблица С.4 - Взаимосвязь с МЭК 60601-1-4

Требования МЭК 60601-1-4:1996 к ПЭМС, включая поправки 1:1999	Соответствующие требования настоящего стандарта
6.8 Сопроводительная документация	-
6.8.201	4.2 и 4.3, перечисление с)
52.201 Документация	-
52.201.1	4.1
52.201.2	4.1 и 4.2
52.201.3	4.2
52.202 ПЛАН МЕНЕДЖМЕНТА РИСКА	-
52.202.1	4.2
52.202.2	5.1.1, 5.1.5
52.202.3	4.1, 5.1.2
52.203 Жизненный цикл разработки	-
52.203.1	5.1.1
52.203.2	5.1.1
52.203.3	-
52.203.4	5.1.7

52.203.5	Раздел 7
52.204 ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА	-
52.204.1	4.2
52.204.2	4.2, раздел 7
52.204.3	-
52.204.3.1	-
52.204.3.1.1	4.2, 7.1
52.204.3.1.2	4.2, 7.1.2
52.204.3.1.3	4.2
52.204.3.1.4	4.2, 7.1.2, перечисление е)
52.204.3.1.5	4.2, 7.1.2
52.204.3.1.6	4.2, 7.1
52.204.3.1.7	4.2
52.204.3.1.8	4.2
52.204.3.1.9	4.2
52.204.3.1.10	4.2

52.204.3.2	-
52.204.3.2.1	4.2
52.204.3.2.2	4.2, 4.3
52.204.3.2.3	-
52.204.3.2.4	-
52.204.3.2.5	4.2
52.204.4	-
52.204.4.1	4.2
52.204.4.2	4.2
52.204.4.3	4.2
52.204.4.4	4.2
52.204.4.5	-
52.204.4.6	4.2
52.205 Квалификация персонала	4.1
52.206 Требования спецификации	-
52.206.1	5.2
52.206.2	7.2.2

52.206.3	-
52.207 АРХИТЕКТУРА	-
52.207.1	5.3.1
52.207.2	5.3
52.207.3	-
52.207.4	-
52.207.5	-
52.208 Проектирование и реализация	-
52.208.1	Раздел 5
52.208.2	-
52.209 ВЕРИФИКАЦИЯ	-
52.209.1	5.7.1
52.209.2	5.1.5, 5.1.6
52.209.3	5.2.6, 5.3.6, 5.4.4, 5.5.5, 5.6, 5.7
52.209.4	-
52.210 ВАЛИДАЦИЯ	-
52.210.1	4.1

52.210.2	4.1
52.210.3	4.1
52.210.4	-
52.210.5	-
52.210.6	-
52.210.7	-
52.211 Модификация	-
52.211.1	Раздел 6
52.211.2	4.1, раздел 6
52.212 ОЦЕНКА	-
52.212.1	4.1

С.5 Взаимосвязь с МЭК 61010-1

Сфера действия МЭК 61010-1 [4] охватывает измерительное оборудование и оборудование для электрических испытаний, оборудование электрического контроля и электрооборудование лаборатории. Только часть лабораторного электрооборудования используется в медицинской среде или в качестве оборудования для *in vitro* диагностики.

Из-за правовых норм или нормативных ссылок, оборудование для *in vitro* диагностики выделяется из МЕДИЦИНСКИХ ИЗДЕЛИЙ, однако, не попадая в сферу действия МЭК 60601-1 [1]. Это связано не только с тем, что инструменты для *in vitro* диагностики не являются МЕДИЦИНСКИМИ ИЗДЕЛИЯМИ, которые прямо контактируют с пациентами, но и с тем, что такую продукцию изготавливают для разных целей различных лабораторий. Использование в качестве инструментов для *in vitro* диагностики или принадлежностей к инструментам для *in vitro* диагностики встречается редко.

Если лабораторное оборудование используется в качестве оборудования для *in vitro* диагностики, полученные измеренные РЕЗУЛЬТАТЫ должны быть ОЦЕНЕНЫ в соответствии с медицинскими критериями. Применение ИСО 14971 требуется для МЕНЕДЖМЕНТА РИСКА. Если подобная продукция также содержит ПО, которое может привести к возникновению ОПАСНОСТИ, например отказу, вызванному ПО, что приводит к нежелательному изменению медицинских данных (измеряемых РЕЗУЛЬТАТОВ), следует учитывать требования МЭК 62302.

Блок-схема, приведенная на рисунке С.3, содержит полезную вспомогательную информацию, объясняющую принципиальный путь ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА и применения настоящего стандарта.

Рисунок С.3 - Применение настоящего стандарта совместно с МЭК 61010-1

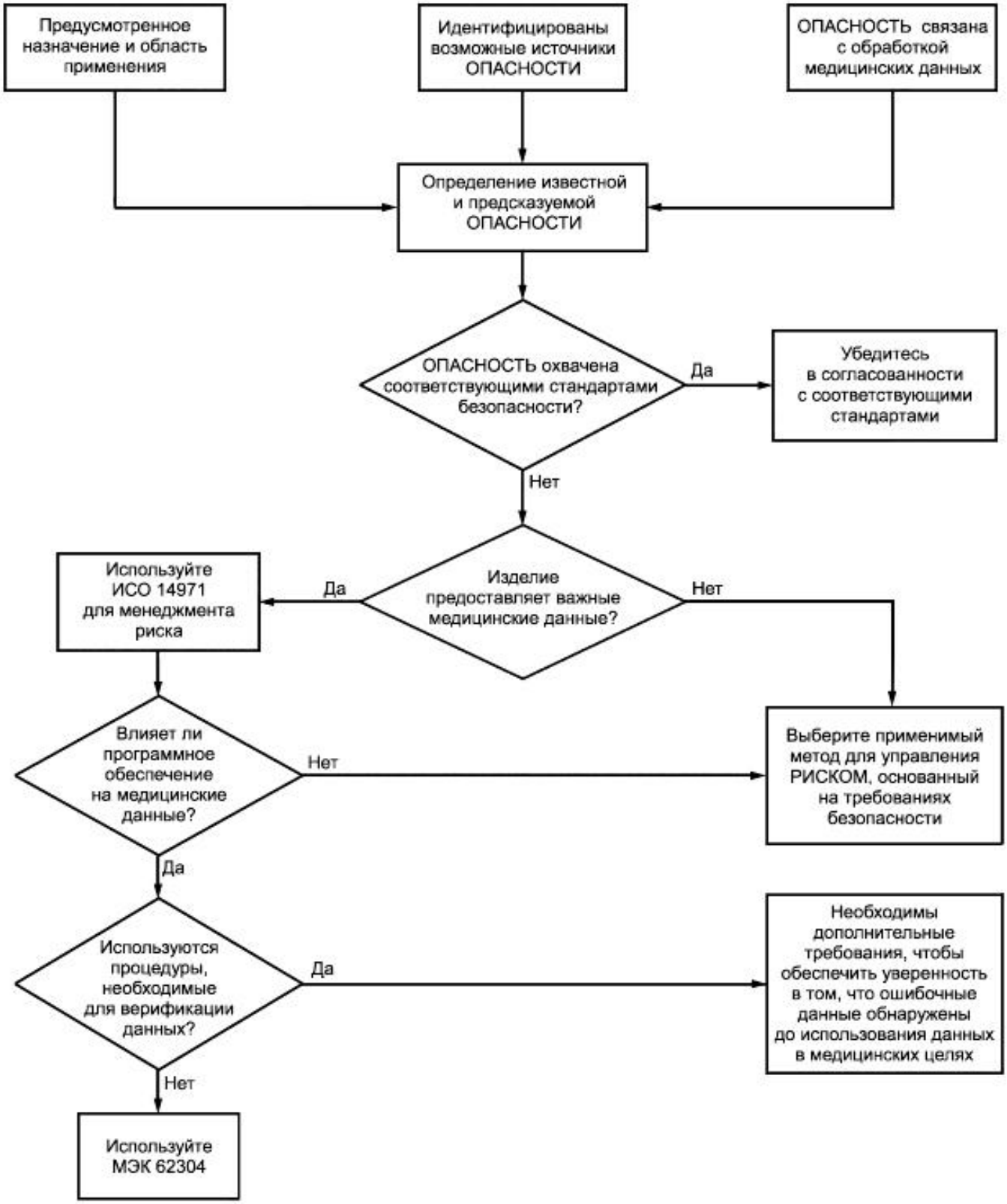

Рисунок С.3 - Применение настоящего стандарта совместно с МЭК 61010-1

С.6 Взаимосвязь с ИСО/МЭК 12207

Настоящий стандарт был разработан исходя из подходов и концепций ИСО/МЭК 12207 [9], который определяет требования для ПРОЦЕССОВ жизненного цикла ПО в общих чертах, то есть не ограничиваясь МЕДИЦИНСКИМИ ИЗДЕЛИЯМИ.

Настоящий стандарт отличается от ИСО/МЭК 12207 главным образом следующим:

- исключены аспекты СИСТЕМЫ, такие как системные требования, АРХИТЕКТУРА СИСТЕМЫ и валидация;
- исключены некоторые ПРОЦЕССЫ, рассматриваемые как дублирующая деятельность, представленные в различных изданиях для МЕДИЦИНСКИХ ИЗДЕЛИЙ;
- добавлены ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА (БЕЗОПАСНОСТЬ) и ПРОЦЕСС выпуска ПО;
- включены документирование и ВЕРИФИКАЦИЯ поддерживающих ПРОЦЕССОВ в ПРОЦЕССЫ разработки и технической поддержки;
- объединены реализация ПРОЦЕССОВ и планирование деятельности каждого ПРОЦЕССА в единую деятельность в ПРОЦЕССАХ разработки и технического обслуживания;
- классифицированы требования с учетом нужд БЕЗОПАСНОСТИ; и

ПРОЦЕССЫ явно не классифицированы на первостепенные или поддерживающие, и не сгруппированы ПРОЦЕССЫ, как это сделано в ИСО/МЭК 12207.

Большинство этих изменений были внесены из-за желания приспособить стандарт к нуждам сферы МЕДИЦИНСКИХ ИЗДЕЛИЙ:

- фокусируясь на аспектах БЕЗОПАСНОСТИ и МЕНЕДЖМЕНТА РИСКА МЕДИЦИНСКИХ ИЗДЕЛИЙ ИСО 14971;
- отбирая подходящие ПРОЦЕССЫ, полезные в регулируемой внешней среде;
- принимая во внимание, что разработка ПО включена в систему качества (которая охватывает некоторые ПРОЦЕССЫ и требования ИСО/МЭК 12207), и
- уменьшая уровень обобщения, чтобы сделать более легким использование.

Настоящий стандарт не противоречит ИСО/МЭК 12207 и может быть полезным в качестве вспомогательной информации для создания хорошо структурированной МОДЕЛИ ЖИЗНЕННОГО ЦИКЛА РАЗРАБОТКИ ПО, которая включает требования настоящего стандарта.

В таблице С.5, которая была подготовлена подкомитетом 7 ИСО/МЭК, показана взаимосвязь между настоящим стандартом и ИСО/МЭК 12207.

Таблица С.5 - Взаимосвязь с ИСО/МЭК 12207

Процессы настоящего стандарта		Процессы ИСО/МЭК 12207	
Деятельность	Задача	Деятельность	Задача
5 ПРОЦЕСС разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ		5.3 Процесс разработки 6.1 Процесс документирования 6.2 Процесс управления конфигурацией 6.4 Процесс верификации 6.5 Процесс валидации 6.8 Процесс решения проблем 7.1 Процесс менеджмента	

5.1 Планирование разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ		5.3.1 Подготовка процесса 5.3.3 Проектирование системной архитектуры 5.3.7 Программирование и тестирование программных средств 5.3.8 Сборка программных средств 5.3.9 Квалификационные испытания программных средств 5.3.10 Сборка системы 6.1.1 Подготовка процесса 6.2.1 Подготовка процесса 6.2.2 Определение конфигурации 6.4.1 Подготовка процесса 6.5.1 Подготовка процесса 6.8.1 Подготовка процесса 7.1.2 Планирование 7.1.3 Выполнение и контроль 7.2.2 Создание инфраструктуры		

		7.2.3 Сопровождение инфраструктуры	
	5.1.1 План разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	5.3.1 Подготовка процесса 7.1.2 Планирование	5.3.1.1 5.3.1.3 5.3.1.4 7.1.2.1
	5.1.2 Поддержание плана разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ в актуальном состоянии	7.1.3 Выполнение и контроль	7.1.3.3
	5.1.3 План разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ относительно проектирования и разработки СИСТЕМЫ	5.3.3 Проектирование системной архитектуры 5.3.10 Сборка системы 6.5.1 Подготовка процесса	5.3.3.1 5.3.10.1 6.5.1.4
	5.1.4 Стандарты, методы и инструменты планирования разработки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	5.3.1 Подготовка процесса	5.3.1.3 5.3.1.4
	5.1.5 Программная интеграция и планирование тестирования интеграции	5.3.8. Сборка программных средств	5.3.8.1

	5.1.6 Планирование ВЕРИФИКАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	6.4.1 Подготовка процесса 5.3.7 Программирование и тестирование программных средств 5.3.8 Сборка программных средств 5.3.9 Квалификационные испытания программных средств	6.4.1.4 6.4.1.5 5.3.7.5 5.3.8.5 5.3.9.3
	5.1.7 Планирование МЕНЕДЖМЕНТА РИСКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	Процесс менеджмента риска	
	5.1.8 Документация планирования	6.1.1 Подготовка процесса	6.1.1.1
	5.1.9 Планирование менеджмента конфигурации ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	6.2.1 Подготовка процесса 6.8.1 Подготовка процесса	6.2.1.1 6.8.1.1
	5.1.10 Поддержка элементов, подлежащих управлению	7.2.2 Создание инфраструктуры 7.2.3 Сопровождение инфраструктуры	7.2.2.1 7.2.3.1
	5.1.11 Управление ЭЛЕМЕНТАМИ КОНФИГУРАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ до ВЕРИФИКАЦИИ	6.2.2 Определение конфигурации	6.2.2.1

5.2 Анализ требований к ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ		5.3.3 Проектирование системной архитектуры 5.3.4 Анализ требований к программным средствам 6.4.2 Верификация	
	5.2.1 Определение и документирование требований к ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ в зависимости от требований СИСТЕМЫ	5.3.3 Проектирование системной архитектуры	5.3.3.1
	5.2.2 Содержание требований к ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ	5.3.4 Анализ требований к программным средствам	5.3.4.1
	5.2.3 Включение мер УПРАВЛЕНИЯ РИСКОМ в требования к ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ		
	5.2.4 ПЕРЕОЦЕНИВАНИЕ АНАЛИЗА РИСКА МЕДИЦИНСКОГО ИЗДЕЛИЯ		
	5.2.5 Обновление требований к СИСТЕМЕ	5.3.4 Анализ требований к программным средствам	Перечисления a) и b)

	5.2.6 Проверка требований к ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ	5.3.4 Анализ требований к программным средствам 6.4.2 Верификация	5.3.4.2 6.4.2.3
5.3 Проектирование АРХИТЕКТУРЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ		5.3.5 Проектирование программной архитектуры	
	5.3.1 Преобразование требований к ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ в АРХИТЕКТУРУ	5.3.5 Проектирование программной архитектуры	5.3.5.1
	5.3.2 Разработка АРХИТЕКТУРЫ для интерфейсов ПРОГРАММНЫХ ЭЛЕМЕНТОВ		5.3.5.2
	5.3.3 Определение требований к функциональным и эксплуатационным характеристикам элементов ПОНП		
	5.3.4 Определение требований к аппаратным и программным средствам СИСТЕМЫ, требуемых элементами ПОНП		
	5.3.5 Идентификация обособленности, необходимой для УПРАВЛЕНИЯ РИСКОМ		

	5.3.6 Проверка АРХИТЕКТУРЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	5.3.5 Проектирование программной архитектуры	
5.4 Детализированная разработка ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ		5.3.6 Техническое проектирование программных средств 6.4.2 Верификация	
	5.4.1 Развитие АРХИТЕКТУРЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ в ПРОГРАММНЫЕ МОДУЛИ	5.3.6 Техническое проектирование программных средств	5.3.6.1
	5.4.2 Разработка детализированного проекта для каждого ПРОГРАММНОГО МОДУЛЯ		
	5.4.3 Разработка детализированного проекта для интерфейсов		5.3.6.2
	5.4.4 Проверка детализированного проекта	6.4.2 Верификация	5.3.6.7
5.5 Исполнение и проверка ПРОГРАММНЫХ МОДУЛЕЙ		5.3.6 Техническое проектирование программных средств 5.3.7 Программирование и тестирование программных средств 6.4.2 Верификация	

	5.5.1 Исполнение каждого ПРОГРАММНОГО МОДУЛЯ	5.3.7 Программирование и тестирование программных средств	5.3.7.1
	5.5.2 Установление ПРОЦЕССА ВЕРИФИКАЦИИ ПРОГРАММНОГО МОДУЛЯ	5.3.6 Техническое проектирование программных средств 5.3.7 Программирование и тестирование программных средств	5.3.6.5 5.3.7.5
	5.5.3 Критерии приемки ПРОГРАММНЫХ МОДУЛЕЙ	5.3.7 Программирование и тестирование программных средств	5.3.7.5
	5.5.4 Дополнительные критерии приемки ПРОГРАММНЫХ МОДУЛЕЙ	5.3.7 Программирование и тестирование программных средств 6.4.2 Верификация	5.3.7.5 6.4.2.5
	5.5.5 ВЕРИФИКАЦИЯ ПРОГРАММНЫХ МОДУЛЕЙ	5.3.7 Программирование и тестирование программных средств	5.3.7.2

5.6 Программная интеграция и испытания в отношении интеграции		5.3.8 Сборка программных средств 5.3.9 Квалификационные испытания программных средств 5.3.10 Сборка системы 6.4.1 Подготовка процесса 6.4.2 Верификация	
	5.6.1 Интеграция ПРОГРАММНЫХ МОДУЛЕЙ	5.3.8 Сборка программных средств	5.3.8.2
	5.6.2 Проверка программной интеграции	5.3.8 Сборка программных средств 5.3.10 Сборка системы	5.3.8.2 5.3.10.1
	5.6.3 Испытания интегрированного ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	5.3.9 Квалификационные испытания программных средств	5.3.9.1
	5.6.4 Содержание испытаний в отношении интеграции		5.3.9.3
	5.6.5 Проверка процедур испытаний в отношении интеграции	6.4.2 Верификация	6.4.2.2
	5.6.6 Проведение регрессионных испытаний	5.3.8 Сборка программных средств	5.3.8.2

	5.6.7 Содержание записей в отношении регрессионных испытаний	5.3.8 Сборка программных средств	5.3.8.2
	5.6.8 Использование программного ПРОЦЕССА решения проблем	6.4.1 Подготовка процесса	6.4.1.6
5.7 Испытания СИСТЕМЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ		5.3.8 Сборка программных средств 5.3.9 Квалификационные испытания программных средств 6.4.1 Подготовка процесса 6.4.2 Верификация 6.8.1 Подготовка процесса	
	5.7.1 Установление испытаний в отношении требований ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	5.3.8 Сборка программных средств 5.3.9 Квалификационные испытания программных средств	5.3.8.4 5.3.9.1
	5.7.2 Использование программного ПРОЦЕССА решения проблем	6.4.1 Подготовка процесса	6.4.1.6

	5.7.3 Повторные испытания после внесения изменений	6.8.1 Подготовка процесса	6 8.1.1
	5.7.4 ВЕРИФИКАЦИЯ испытаний СИСТЕМЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	6.4.2 Верификация 5.3.9 Квалификационные испытания программных средств	6.4.2.2 5.3.9.3
	5.7.5 Содержание отчета по испытаниям СИСТЕМЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	5.3.9 Квалификационные испытания программных средств	5.3.9.1
5.8 Выпуск ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ		5.3.9 Квалификационные испытания программных средств 5.4.2 Эксплуатационные испытания 6.2.5 Оценка конфигурации 6.2.6 Управление выпуском и поставка	
	5.8.1 Обеспечение полного завершения ВЕРИФИКАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	5.4.2 Эксплуатационные испытания 6.2.6 Управление выпуском и поставка	5.4.2.1 5.4.2.2 6.2.6.1

	5.8.2 Документирование известных остаточных АНОМАЛИЙ	6.2.5 Оценка конфигурации 5.3.9 Квалификационные испытания программных средств	6.2.5.1 5.3.9.3
	5.8.3 ОЦЕНИВАНИЕ известных остаточных АНОМАЛИЙ		
	5.8.4 Документирование выпущенных ВЕРСИЙ	6.2.6 Управление выпуском и поставка	6.2.6.1
	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 воспроизводимости выпуска ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ		
6 ПРОЦЕСС технической поддержки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ		5.5 Процесс сопровождения 6.2 Процесс управления конфигурацией	

6.1 Установление плана технической поддержки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ		5.5.1 Подготовка процесса	5.5.1.1
6.2 Анализ модификации и проблем		5.5.1 Подготовка процесса 5.5.2 Анализ проблем и изменений 5.5.3 Внесение изменений 5.5.5 Перенос	
	6.2.1 Документирование и ОЦЕНИВАНИЕ обратной связи		
	6.2.1.1 Мониторинг обратной связи 6.2.1.2 Документирование и ОЦЕНИВАНИЕ обратной связи	5.5.1 Подготовка процесса	5.5.1.1 5.5.1.2
	6.2.1.3 ОЦЕНИВАНИЕ влияния ОТЧЕТА О ПРОБЛЕМАХ на БЕЗОПАСНОСТЬ	5.5.2 Анализ проблем и изменений	5.5.2.1 5.5.2.2 5.5.2.3 5.5.2.4
	6.2.2 Использование программного ПРОЦЕССА решения проблем	5.5.1 Подготовка процесса	5.5.1.2

	6.2.3 Анализ ЗАПРОСОВ НА ИЗМЕНЕНИЕ	5.5.2 Анализ проблем и изменений	5.5.2.1
	6.2.4 Одобрение ЗАПРОСА НА ИЗМЕНЕНИЕ	5.5.2 Анализ проблем и изменений	5.5.2.5
	6.2.5 Информирование пользователей и регулирующих органов	5.5.3 Внесение изменений 5.5.5 Перенос	5.5.3.1 5.5.5.3
6.3 Осуществление модификации		5.5.3 Внесение изменений 6.2.6 Управление выпуском и поставка	
	6.3.1 Использование установленного ПРОЦЕССА осуществления модификации	5.5.3 Внесение изменений	5.5.3.2
	6.3.2 Повторный выпуск модифицированной ПРОГРАММНОЙ СИСТЕМЫ	6.2.6 Управление выпуском и поставка	6.2.6.1
7* ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ		ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА Процесс в настоящем стандарте адресован рискам/вопросам опасностей, которые не рассматриваются в ИСО/МЭК 12207. Существует некоторая общность (меры по управлению риском и т.д.), но направление анализа совершенно иное	
8 ПРОЦЕСС менеджмента конфигурации ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ		5.5 Процесс сопровождения 6.2 Процесс управления конфигурацией	

8.1* Идентификация конфигурации		6.2.2 Определение конфигурации	
	8.1.1 Установление средств для идентификации ЭЛЕМЕНТОВ КОНФИГУРАЦИИ	6.2.2 Определение конфигурации	6.2.2.1
	8.1.2 Идентификация ПОНП		
	8.1.3 Идентификация документации конфигурации СИСТЕМЫ	6.2.2 Определение конфигурации	6.2.2.1
8.2* Управление изменениями		5.5.3 Внесение изменений 6.2.3 Контроль конфигурации	
	8.2.1 Одобрение ЗАПРОСОВ НА ИЗМЕНЕНИЯ	6.2.3 Контроль конфигурации	6.2.3.1
	8.2.2 Осуществление изменений	5.5.3 Внесение изменений 6.2.3 Контроль конфигурации	5.5.3.2 6.2.3.1
	8.2.3 ВЕРИФИКАЦИЯ изменений	6.2.3 Контроль конфигурации	6.2.3.1
	8.2.4 Обеспечение средствами для ПРОСЛЕЖИВАЕМОСТИ изменений		

8.3 Учет статуса конфигурации		6.2.4 Учет состояний конфигурации	6.2.4.1
9 Программный ПРОЦЕСС решения проблем		5.5 Процесс сопровождения 6.2 Процесс управления конфигурацией 6.8 Процесс решения проблем	
9.1 Подготовка ОТЧЕТОВ О ПРОБЛЕМАХ		6.8.1 Подготовка процесса 6.8.2 Решение проблемы	6.8.1.1, перечисление b) 6.8.2.1
9.2 Исследование проблемы		6.8.2 Решение проблемы 6.8.1 Подготовка процесса	6.8.2.1 6.8.1.1, перечисление b)
9.3 Консультирование заинтересованных сторон		6.8.1 Подготовка процесса	6.8.1.1, перечисление a)
9.4 Использование процесса управления изменениями		6.2.3 Контроль конфигурации 5.5.3 Внесение изменений	
9.5 Поддержание записей		6.8.1 Подготовка процессса	6.8.1.1, перечисление a)
9.6 Анализ проблем на предмет выявления тенденций		6.8.1 Подготовка процесса 6.8.2 Решение проблемы	6.8.1.1, перечисление b) 6.8.2.1

9.7 ВЕРИФИКАЦИЯ решения проблем ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ		6.8.1 Подготовка процесса	6.8.1.1, перечисление d)
9.8 Содержание документации по испытаниям			Все тестируемые задачи в ИСО /МЭК 12207 требуют документирования

С.7 Взаимосвязь с МЭК 61508

Поднимался вопрос о том, должен ли настоящий стандарт, имея дело с ПО, критичным к БЕЗОПАСНОСТИ, следовать принципам МЭК 61508.

Позицию настоящего стандарта объясняет следующее.

МЭК 61508 охватывает три главных вопроса:

1) жизненный цикл МЕНЕДЖМЕНТА РИСКА и ПРОЦЕССЫ жизненного цикла;

2) определение уровней БЕЗОПАСНОСТИ эксплуатации оборудования;

3) рекомендуемые техники, инструменты и методы для разработки ПО и уровни независимости персонала, ответственного за выполнение различных ЗАДАЧ.

Вопрос 1) включен в настоящий стандарт в качестве нормативной ссылки на ИСО 14971 (стандарт по МЕНЕДЖМЕНТУ РИСКА для сферы МЕДИЦИНСКИХ ИЗДЕЛИЙ). Необходимость этой ссылки состоит в том, чтобы адаптировать подход ИСО 14971 к МЕНЕДЖМЕНТУ РИСКА как составной части ПРОЦЕССА РАЗРАБОТКИ ПО для МЕДИЦИНСКИХ ИЗДЕЛИЙ.

Для вопроса 2) настоящий стандарт принимает более простой подход, чем МЭК 61508. Последний классифицирует ПО на четыре "уровня БЕЗОПАСНОСТИ эксплуатации оборудования", определенные с точки зрения надежности. Цели надежности устанавливают после проведения АНАЛИЗА РИСКА, который определяет как тяжесть, так и вероятность ВРЕДА, вызванного отказом ПО.

Настоящий стандарт упрощает вопрос 2), запрещая рассмотрение вероятности отказа ПО до его классификации. Классификация на три класса БЕЗОПАСНОСТИ ПО основана только на тяжести этого ВРЕДА, вызванного отказом ПО. После классификации разные ПРОЦЕССЫ требуются для различных классов БЕЗОПАСНОСТИ ПО: намерение состоит в дальнейшем уменьшении вероятности отказа ПО.

Вопрос 3) не затронут в настоящем стандарте. Пользователям настоящего стандарта рекомендуется использовать МЭК 61508 в качестве источника современных программных методов, техник и инструментов, отмечая, что другие подходы, настоящие и будущие, могут обеспечить достаточно хорошие РЕЗУЛЬТАТЫ. Настоящий стандарт не дает никаких рекомендаций относительно независимости людей, ответственных за один вид деятельности в области ПО (например, ВЕРИФИКАЦИЮ) от тех, кто отвечает за другую (например, проектирование). В частности, настоящий стандарт не предлагает никаких требований для независимого эксперта по БЕЗОПАСНОСТИ, так как это рассматривается в ИСО 14971.

Приложение D (справочное). Применение

Приложение D
(справочное)

D.1 Введение

В настоящем приложении представлен обзор того, как настоящий стандарт может быть применен к ПРОЦЕССАМ ИЗГОТОВИТЕЛЯ. Также предполагается, что другие стандарты, такие как ИСО 13485 [7], требуют подходящих и сопоставимых ПРОЦЕССОВ.

D.2 Система менеджмента качества

В контексте настоящего стандарта для изготовителей МЕДИЦИНСКИХ ИЗДЕЛИЙ, включая ПО к МЕДИЦИНСКИМ ИЗДЕЛИЯМ, установление СИСТЕМЫ менеджмента качества требуется в соответствии с 4.1. Настоящий стандарт не требует, чтобы система менеджмента качества обязательно была сертифицирована.

D.3 ОЦЕНИВАНИЕ ПРОЦЕССОВ менеджмента качества

Рекомендуется ОЦЕНИВАТЬ то, насколько полно установленные и документированные ПРОЦЕССЫ СИСТЕМЫ менеджмента качества уже охватывают ПРОЦЕССЫ жизненного цикла ПО посредством аудита, проверок или анализа в рамках ответственности ИЗГОТОВИТЕЛЯ. Любые выявленные недостатки могут быть урегулированы путем расширения ПРОЦЕССА менеджмента качества или описаны отдельно. Если ИЗГОТОВИТЕЛЬ уже имеет доступные описания ПРОЦЕССОВ, которые регулируют разработку, ВЕРИФИКАЦИЮ и валидацию ПО, тогда они также должны быть ОЦЕНЕНЫ, чтобы определить, насколько они согласуются 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 СИСТЕМЕ менеджмента качества, или интеграцией новых ПРОЦЕССОВ. Настоящий стандарт не устанавливает того, как это должно быть сделано; ИЗГОТОВИТЕЛЬ может сделать это любым подходящим способом.

ИЗГОТОВИТЕЛЬ отвечает за обеспечение того, чтобы ПРОЦЕССЫ, описанные в настоящем стандарте, были надлежащим образом введены в действие, когда ПО МЕДИЦИНСКОГО ИЗДЕЛИЯ разработано ИЗГОТОВИТЕЛЯМИ оригинального оборудования или субподрядчиками, не имеющими своей собственной документированной СИСТЕМЫ менеджмента качества.

D.5 Контрольный список для небольших компаний, не имеющих сертифицированной СИСТЕМЫ менеджмента качества

ИЗГОТОВИТЕЛЬ должен определить самый высокий класс БЕЗОПАСНОСТИ ПО (*A, B* или *C*). В таблице D.1 приведены все виды деятельности, описанные в настоящем стандарте. Ссылка на ИСО 13485 должна помочь определить место в СИСТЕМЕ менеджмента качества. Основываясь на требуемом классе БЕЗОПАСНОСТИ ПО, ИЗГОТОВИТЕЛЮ следует ОЦЕНИТЬ каждое требуемое действие относительно уже существующих ПРОЦЕССОВ. Если требование уже охвачено, должны быть приведены ссылки на подходящие описания ПРОЦЕССА.

Если существует расхождение, необходимо принять меры для улучшения ПРОЦЕССА.

Список также может быть использован для оценивания ПРОЦЕССОВ после завершения действия.

Таблица D.1 - Контрольный список для небольших компаний без сертифицированной СИСТЕМЫ менеджмента качества

Вид деятельности	Соответствующий пункт ИСО 13485:2003	Охватывается существующими процедурами?	Если да: Ссылка	Предприня-тые 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 ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ		Да/нет		
5.4 Детализированный проект ПО		Да/нет		
5.5 Реализация и ВЕРИФИКАЦИЯ ПРОГРАММНОГО МОДУЛЯ		Да/нет		
5.6 Интеграция ПО и тестирование интеграции		Да/нет		
5.7 Тестирование ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СИСТЕМЫ	7.3.3 Выходные данные проектирования и разработки 7.3.4 Анализ проекта и разработки	Да/нет		

5.8 Выпуск ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	7.3.5 Верификация проекта и разработки 7.3.6 Валидация проекта и разработки	Да/нет		
6.1 Установление плана технической поддержки ПО	7.3.7 Управление изменениями проекта и разработки	Да/нет		
6.2 Анализ проблем и модификации		Да/нет		
6.3 Осуществление модификации	7.3.5 Верификация проекта и разработки 7.3.6 Валидация проекта 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 УПРАВЛЕНИЕ РИСКАМИ 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 (справочное). Сведения о соответствии ссылочных международных стандартов национальным стандартам Российской Федерации (и действующим в этом качестве межгосударственным стандартам)

Приложение ДА
(справочное)

Таблица ДА.1

Обозначение ссылочного международного стандарта	Степень соответствия	Обозначение и наименование соответствующего национального стандарта
ИСО 14971:2007	IDT	ГОСТ ISO 14971-2011 "Изделия медицинские. Применение менеджмента риска к медицинским изделиям"
Примечание - В настоящей таблице использовано следующее условное обозначение степени соответствия стандартов: - IDT - идентичные стандарты.		

Библиография

[1]	МЭК 60601-1:2005	Изделия медицинские электрические. Часть 1. Общие требования безопасности с учетом основных функциональных характеристик
	IEC 60601-1:2005	Medical electrical equipment - Part 1: General requirements for basic safety and essential performance
[2]	МЭК 60601-1-4:1996	Изделия медицинские электрические. Часть 1. Общие требования к безопасности. 4. Вспомогательный стандарт. Программируемые медицинские электрические системы
	IEC 60601-1-4:1996	Medical electrical equipment - Part 1: General requirements for safety - 4: Collateral standard: Programmable electrical medical systems
[3]	МЭК 61508-3	Системы электрические/электронные/программмируемые электронные, связанные с функциональной безопасностью. Часть 3. Требования к программному обеспечению
	IEC 61508-3	Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 3: Software requirements
[4]	МЭК 61010-1:2001	Требования к безопасности электрооборудования для проведения измерений, управления и лабораторного использования. Часть 1. Общие требования
	IEC 61010-1:2001	Safety requirements for electrical equipment for measurement, control, and laboratory use - Part 1: General requirements
[5]	ИСО 9000:2005	Системы менеджмента качества. Определения и словарь
	ISO 9000:2005	Quality management systems - Fundamentals and vocabulary

[6]	ИСО 9001:2000	Системы менеджмента качества. Требования
	ISO 9001:2000	Quality management systems - Requirements
[7]	ИСО 13485:2003	Медицинские изделия. Системы менеджмента качества. Системные требования для целей регулирования
	ISO 13485:2003	Medical devices - Quality management systems - Requirements for regulatory purposes
[8]	ИСО/МЭК 9126-1:2001	Программирование. Качество продукта. Часть 1. Модель качества
	ISO/IEC 9126-1:2001	Software engineering - Product quality - Part 1: Quality model
[9]	ИСО/МЭК 12207:1995	Информационные технологии. Процессы жизненного цикла программного обеспечения
	ISO/IEC 12207:1995	Information technology - Software life cycle processes Amendment 1 (2002)Amendment 2 (2004)
[10]	ИСО/МЭК 14764:1999	Информационные технологии. Сопровождение программного обеспечения
	ISO/IEC 14764:1999	(Information technology - Software maintenance)
[11]	ИСО/МЭК 90003:2004	Программирование. Руководство по применению ИСО 9001:2000 для компьютерного программного обеспечения
	ISO/IEC 90003:2000	Software engineering - Guidelines for the application of ISO 9001:2000 to computer software

[12]	ИСО/МЭК Руководящие указания 51:1999	Аспекты безопасности. Руководящие указания по включению их в стандарты
	ISO/IEC Guide 51:1999	Safety aspects - Guidelines for their inclusion in standards
[13]	IEEE 610.12:1990	Глоссарий терминологии программного обеспечения
	IEEE 610.12:1990	IEEE standard glossary of software engineering terminology
[14]	IEEE 1044:1993	Классификация аномалий программного обеспечения
	IEEE 1044:1993	IEEE standard classification for software anomalies
[15]	МЭК 60601-1-6	Изделия медицинские электрические. Часть 1-6. Общие требования безопасности. Эксплуатационная пригодность
	IEC 60601-1-6	Medical electrical equipment - Part 1-6: General requirements for safety - Collateral standard: Usability

Предметный указатель терминов

ДЕЯТЕЛЬНОСТЬ
Изменение управления
Изменение запроса
Завершение
Конфигурация идентификации
Управление конфигурацией
Конфигурация учета состояния
Определение
Поставка
Проектирование и обслуживание
Идентификация опасностей
Техническое обслуживание
Сопоставление
Модификация реализации
Планирование
Проблема анализа и модификации
Решение проблемы
Обязательства
Требования
Анализ требований
Анализ РИСКОВ
Управление РИСКАМИ
Программное обеспечение архитектурного проектирования
Программное обеспечение детального проектирования
Разработка программного обеспечения
Программное обеспечение интеграции
Интеграция программного обеспечения и интеграционное тестирование
Сопровождение программного обеспечения
Версия программного обеспечения
Требования к программному обеспечению анализа
Программное обеспечение системы тестирования
Программная реализация блока и тестирование
Проверка
АНОМАЛИЯ
Определение
АРХИТЕКТУРА
Определение
ИЗМЕНЕНИЕ ЗАПРОСА
Определение
ЭЛЕМЕНТ КОНФИГУРАЦИИ
Определение
ПОНП
ПОСТАВКА
Определение
ОЦЕНИВАНИЕ
Переоценивание
ВРЕД
Определение
ОПАСНОСТЬ
Определение
Непредвиденная
ИЗГОТОВИТЕЛЬ
Определение

МЕДИЦИНСКОЕ ИЗДЕЛИЕ
Определение
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МЕДИЦИНСКОГО ИЗДЕЛИЯ
Изменения
Определение
ОТЧЕТ О ПРОБЛЕМАХ
Классификация
Определение
ПРОЦЕСС
Применение
Управления изменениями
Классификация
Управление конфигурацией

Принятие решений
Определение
Развитие
Осуществление
Улучшение
Жизненный цикл
Техническое обслуживание
Сопоставление
Модификация
Упущение
Выходные данные
Физиология
Решение проблем
Менеджмент качества
Обязательства
Требования
Анализ РИСКОВ
МЕНЕДЖМЕНТ РИСКА
Программное обеспечение
Разработка программного обеспечения
Сопровождение программного обеспечения
Версия программного обеспечения
Системные требования
Верификация
РЕГРЕССИВНОЕ ТЕСТИРОВАНИЕ
Определение
РИСК
Определение
НЕСЕРЬЕЗНАЯ травма
Достаточный обзор
УПРАВЛЕНИЕ РИСКОМ
СЕРЬЕЗНАЯ ТРАВМА
ПОНП
Неприемлемый
АНАЛИЗ РИСКОВ
Определение
УПРАВЛЕНИЕ РИСКОМ
Деятельность
Определение
Измерительное оборудование
Измерение
Требования
Сегрегация
МЕНЕДЖМЕНТ РИСКА
Определение
Медицинское изделие
Отчет
Файл менеджмента РИСКА
Определение
БЕЗОПАСНОСТЬ
Определение
ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ
Определение

Требования
СЕРЬЕЗНАЯ ТРАВМА
Определение
НЕСЕРЬЕЗНАЯ ТРАВМА
РАЗРАБОТКА МОДЕЛИ ЖИЗНЕННОГО ЦИКЛА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
Определение
ПРОГРАММНЫЙ ЭЛЕМЕНТ
Изменения
Определение
Интеграция
Разделы
Производительность
Сегрегация
ПОНП
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ неизвестного происхождения

См. ПОНП
ПРОГРАММНЫЙ ПРОДУКТ
Определение
Релиз
СИСТЕМА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
Определение
Интеграция

Требования
Тестирование
ЭЛЕМЕНТ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
Определение
Интеграция
Проверка
Верификация элемента программного обеспечения
ПОНП
Изменения
Элемент конфигурации
Определение
Номенклатура
Элемент программного обеспечения
СИСТЕМА
Конфигурация
Определение
План разработки
Существование
Внедрение
Требования
ЗАДАЧА
Завершение
Управление конфигурацией
Определение
Поставка
Проектирование и обслуживание
Техническое обслуживание
Сопоставление
Требования
УПРАВЛЕНИЕ РИСКАМИ
Верификация
ПРОСЛЕЖИВАЕМОСТЬ
Определение
ВЕРИФИКАЦИЯ
Определение
ВЕРСИЯ
Определение

УДК 658:562.014:006.354 ОКС 11.040 Р20 ОКП 94 000

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

Электронный текст документа
подготовлен АО "Кодекс" и сверен по:
официальное издание
М.: Стандартинформ, 2015