УЧЕБНОЕ ПОСОБИЕ, версия 3

ГОСТ IEC 62304-2022 «Изделия медицинские. Программное обеспечение. Процессы жизненного цикла»

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

Сведения о стандарте

1 Подготовлен рабочей группой, состоящей из представителей ООО «Медитест», ООО «Аурига», ООО «Компания «ЭЛТА» на основе собственного перевода на русский язык англоязычной версии.

2 Принят Межгосударственным советом по стандартизации, метрологии и сертификации (протокол от 30.09.2022 г. № 154-П). За принятие проголосовали: Беларусь, Киргизия, Россия, Узбекистан.

3 Приказом ФАТРиМ № 1196- введён в действие в качестве национального стандарта РФ **с 01.09.2023 г**.

Оглавление

Рисунки

Список таблиц

Введение

Программное обеспечение (далее – «**ПО**») часто является неотъемлемой частью технологии медицинского изделия (далее – «**МИ**»). Создание безопасного и результативного МИ, содержащего ПО, требует знаний о его предназначении, а также доказательств того, что ПО надёжно функционирует, не создавая недопустимых рисков.

Настоящий стандарт определяет основу процессов (3.14) жизненного цикла совместно с деятельностью (3.1) и задачами (3.31) - (далее «**ДиЗ**»)[1], необходимыми для проектирования и техподдержки (обслуживания) безопасного ПОМИ. Настоящий стандарт определяет требования для каждого процесса жизненного цикла. **Каждый процесс жизненного цикла состоит из совокупности видов деятельности, причём большинство видов деятельности, в свою очередь, состоят из набора задач**[2].

В качестве основной концепции полагается, что ПОМИ проектируется и обслуживается с использованием систем менеджмента качества (далее «**СМК**») (см. 4.1) и систем менеджмента риска (см. 4.2). Процесс менеджмента риска уже достаточно хорошо описан в международном стандарте ISO 14971:2019[3]. Поэтому настоящий стандарт использует ссылки на этот стандарт. Некоторые незначительные дополнительные требования к менеджменту риска необходимы для ПО, особенно в области определения вклада факторов ПО, связанных с опасностями. Эти требования установлены в § 7 как процесс менеджмента риска ПО.

Является ли ПО фактором, способствующим опасной ситуации, определяется во время деятельности по идентификации опасности в процессе менеджмента риска. Опасные ситуации, которые могут быть косвенно вызваны ПО (например, предоставляя вводящую в заблуждение информацию, которая может вызвать неверную реакцию администрирования), рассматриваются, когда определяется, является ли ПО способствующим фактором. Решение подвергнуть ПО управлению риском принимается в течение деятельности по управлению риском в процессе менеджмента риска. Процесс менеджмента риска ПО, требуемый настоящим стандартом, должен быть включён в процесс менеджмента риска изделия согласно ISO 14971:2019.

Процесс разработки ПО состоит из множества действий. Эти действия показаны на рис.1 и описаны в § 5. Поскольку множество инцидентов в этой области связано с обслуживанием или техподдержкой систем МИ, включая неподходящие обновления ПО и его модернизации, процесс техподдержки (обслуживания) ПО считается столь же важным, как и процесс разработки ПО. Процесс техподдержки ПО очень похож на процесс разработки ПО. Это показано на рис. 2[4] и описано в разделе 6.

[1] В стандарте словосочетание «**Деятельность и/или Задачи**» с теми или иными вариациями встречается более 10 раз. Возможно разработчики стандарта хотели этим подчеркнуть **системный и частный характер** процессного подхода? Поэтому автор учебного пособия решил, что будет полезно заменить два этих слова аббревиатурой "**ДиЗ**" чтобы сразу по тексту можно было про это вспомнить. Также предлагаем обязательно ознакомиться с предложенными нами вариантами для всех ключевых терминов процессного подхода – они даны в рамках в § 3 после каждого из ключевых терминов

[2] См. примеч.1 – тут всё с ног на голову тут написали ☹ Методологические совершенно неверное мысле-построение! Более подробно смотри на эту тему мой ролик на Ютубе – https://youtu.be/6iWWSkcW16Q..

[3] В оригинале везде была указана старая версия ISO 14971:2007. Редактор пособия заменил ссылки на новую версию. Кроме того, к нему дополнительно есть Технический отчёт ISO/TR 24971:2020 «Изделия медицинские. Руководство по применению ISO 14971» (кому нужно учебное пособие – обращайтесь!)

[4] Мы объединили два рисунка в один. Кроме того, в оригинальной версии ГОСТа содержится ошибка во втором рисунке с номерами разделов с 6.1 по 6.3.

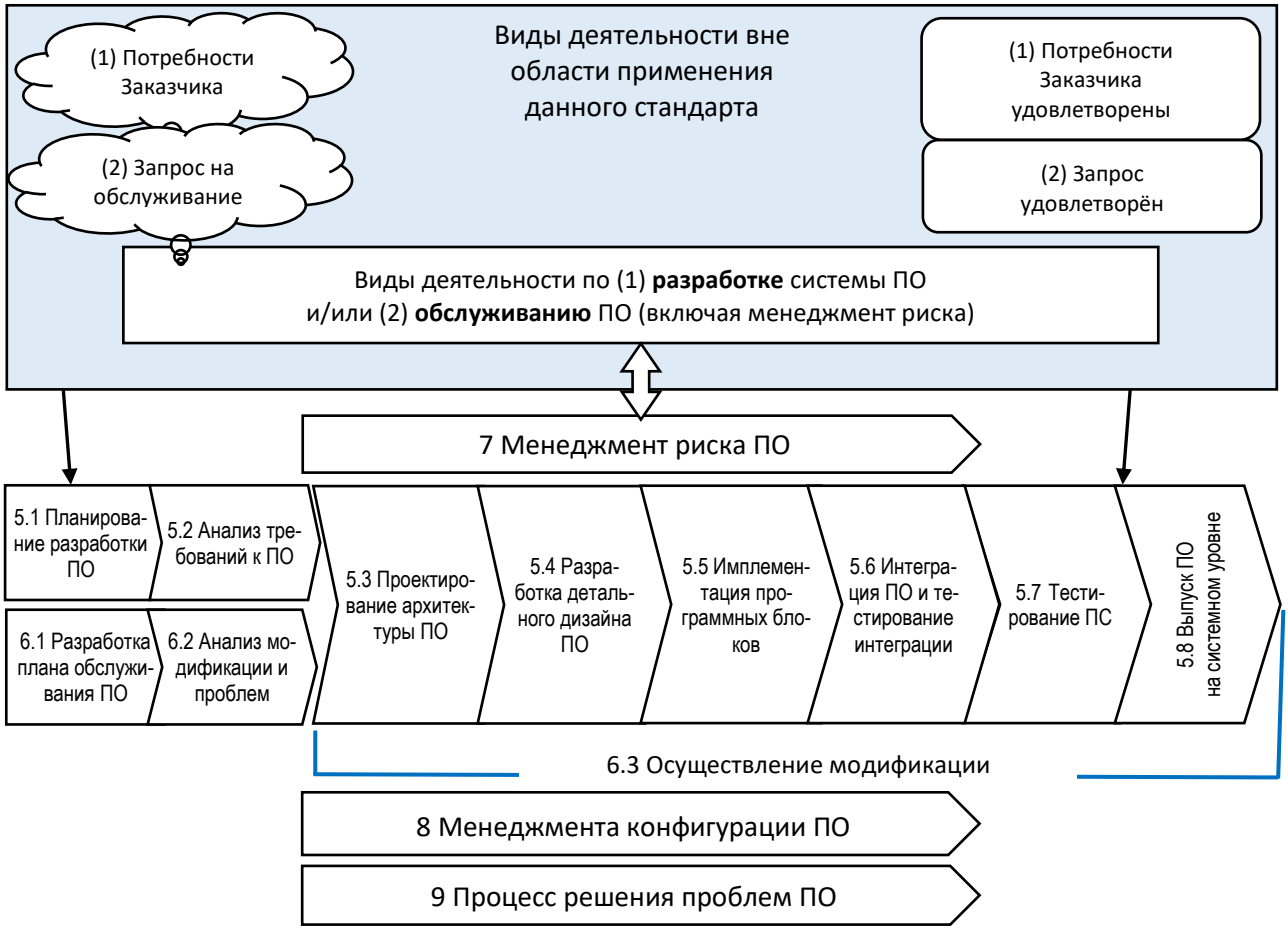


Рисунок 1 – Краткий обзор процессов и деятельности по разработке ПО

Рисунок 2 – Краткий обзор процессов и деятельности по техподдержке ПО

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

Настоящий стандарт идентифицирует два дополнительных процесса, которые считаются важными для разработки безопасного ПОМИ. Это процесс **менеджмента конфигурации ПО** (§ 8) и **процесс разрешения проблем ПО** (§ 9).

Изменение (Amd.) 1 **добавляет в настоящий стандарт требования к устаревшему/наследуемому ПО**, если разработка ПО была проведена до появления текущей версии, с целью оказания помощи изготовителям, которые должны продемонстрировать соответствие настоящему стандарту для соответствия Евродирективам. Изменения в классификации безопасности ПО включают уточнение требований и обновление классификации безопасности ПО, а также подход, основанный на оценке рисков.

Настоящий стандарт не устанавливает организационную структуру изготовителя или то, какое структурное подразделение организации должно осуществлять выполнение процесса, ДиЗ. Требование состоит в том, что в целях соответствия настоящему стандарту процесс, ДиЗ должны быть завершены.

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

Настоящий стандарт **не предписывает конкретную модель жизненного цикла**. Пользователи ответственны за выбор модели жизненного цикла для проекта ПО и за отображение процессов, ДиЗ настоящего стандарта применительно к этой модели.

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

– «должен» означает, что соответствие требованиям или испытаниям обязательно для соответствия

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

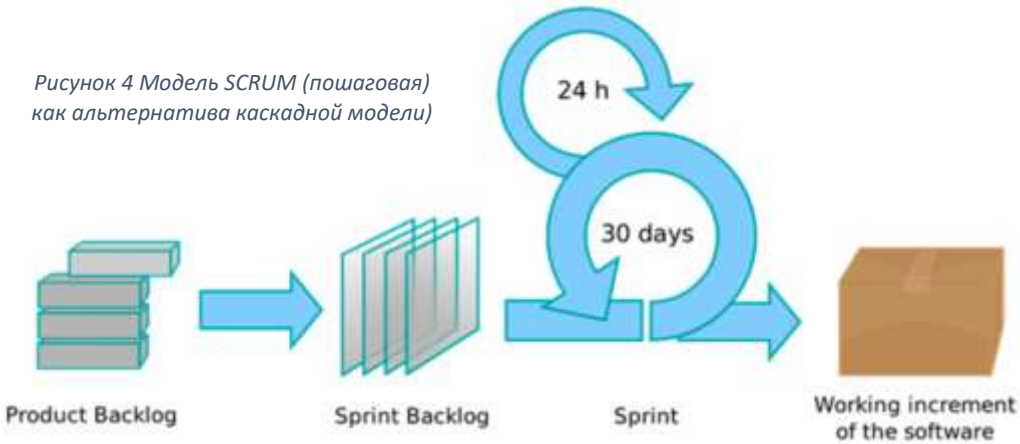
1 Область применения

1.1 Цель

Настоящий стандарт устанавливает требования к жизненному циклу ПОМИ. Совокупность процессов, ДиЗ, описанных в данном стандарте, устанавливает общую основу для процессов жизненного цикла ПОМИ[5].

Рисунок 3 Модель "водопад"

Рисунок 4 Модель SCRUM (пошаговая) как альтернатива каскадной модели)



Цель настоящего стандарта состоит в том, чтобы обеспечить процесс разработки, который позволит последовательно создавать высококачественное и безопасное ПОМИ. Для достижения этой цели настоящий стандарт устанавливает минимальную ДиЗ, которые необходимо выполнить для уверенности в том, что разработанное таким образом ПО (далее – ПО) позволяет создавать высоконадёжное и безопасное ПОМИ.

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

Следует отметить, что в настоящем стандарте деятельность выполняется в рамках процессов, а задачи опреде-

[5] Здесь и далее в две колонки шрифтом Times вставлен комментирующий текст из приложения В.

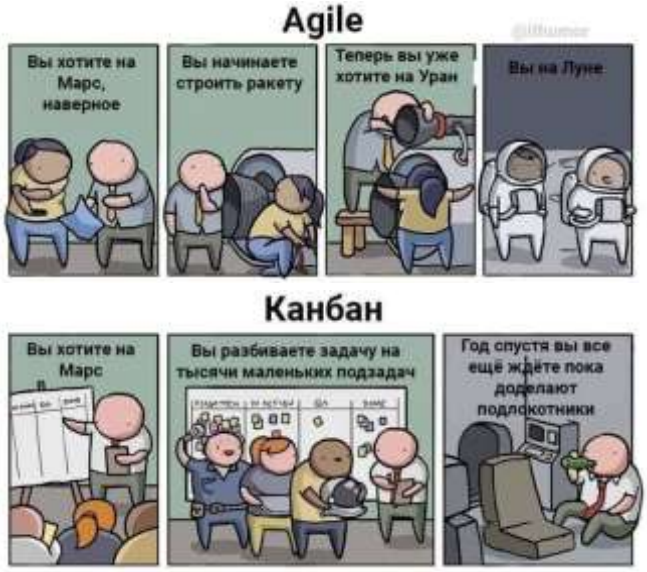
ляются при осуществлении деятельности. Например, деятельностью, выполняемой в рамках процесса разработки ПО, являются планирование разработки ПО, анализ требований к ПО, проектирование архитектуры ПО, разработка детального дизайна ПО, имплементация программных блоков и их верификация, интеграция и тестирование интеграции ПО, тестирование ПС и выпуск ПО. задачи являются конкретными требованиями при осуществлении деятельности.

Настоящий стандарт не требует использования определённой модели жизненного цикла разработки ПО. Тем не менее соответствие настоящему стандарту подразумевает наличие зависимости между процессами, поскольку входные данные одного процесса являются выходами другого процесса. Например, классификация безопасности ПО для ПС должна быть завершена после того, как процесс анализа рисков установит, какой вред может быть причинён в результате отказа ПС.

Из-за указанных логических зависимостей между процессами в настоящем стандарте целесообразней описывать процессы в последовательности, подразумевающей «водопадную» или «сквозную» модель жизненного цикла. Однако можно использовать и другие модели. Некоторые стратегии разработки (модели) определены в стандарте ISO/IEC 12207[A] и включают (см. также таблицу B. 1)[6]:

- **Водопад**. «Сквозная» стратегия, также называемая «водопад», состоит в выполнении процесса разработки за один раз. Нужно установить потребности потребителя, определить требования, разработать проект (дизайн) системы, имплементировать систему, протестировать, исправить и осуществить поставку;

- **Пошаговую**. Она устанавливает потребности потребителя и определяет требования системы. Далее разработка осуществляется в виде последовательной сборки. Первая сборка реализует часть запланированных возможностей, следующая добавляет ещё часть возможностей, и так далее, до тех пор, пока система не будет завершена;



- **Эволюционную**. Она так же развивает систему в сборке, но в отличие от пошаговой стратегии признавая, что нужды потребителя до конца не изучены и все требования не могут быть определены заранее. В этой стратегии нужды заказчика и системные требования определяются заранее, а затем актуализируются при каждой последующей сборке.

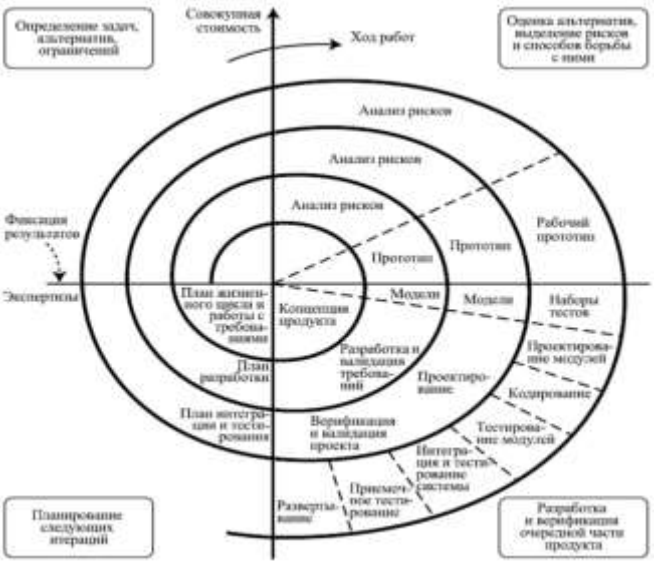


Таблица 1(B.1) – Разработка стратегий (моделей), как это определено в ISO/IEC 12207

Стратегия разработки	С самого начала определяет все требования?	Многократные циклы разработки?	Поставляет временное ПО?
Водопад (сквозная)	Да	Нет	Нет
Пошаговая (предварительно запланированное улучшение продукции)[7]	Да	Да	Возможно
Эволюционная	Нет	Да	Да

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

Другие жизненные циклы, особенно эволюционные, позволяют процессу вырабатывать выходные данные до того,

[6] См. хорошую статью на Хабре

[7] Рекомендация по применению Agile методов при проектировании МИ см. у FDA.

как будут доступны все входные данные для этого процесса. Например, новая ПСЧ может быть определена, классифицирована, имплементирована и верифицирована до того, как будет закончена целая программная архитектура. Такие жизненные циклы несут в себе риск того, что изменение или разработка выходных данных одного процесса сделает недействительными выходные данные другого процесса. Как бы там ни было, все жизненные циклы используют комплексную систему управления конфигурацией, чтобы убедиться, что выходные данные всех процессов доводятся до согласованного состояния и поддерживаются все необходимые зависимости.

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

- все выходные данные процесса должны поддерживаться в согласованном состоянии; каждый раз, когда выходные данные процесса создаются или меняются, все выходные данные всех связанных с ними процессов должны обновляться, чтобы поддерживать их согласованность друг с другом и поддерживать все зависимости, явные или подразумевающиеся, требуемые настоящим стандартом;
- все выходные данные процесса должны быть доступны в случае необходимости в качестве входных данных для дальнейшей работы над ПО;
- перед тем, как любое ПОМИ 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.

1.2 Применимость

Настоящий стандарт применяется при разработке и техподдержке ПОМИ, когда ПО само по себе является МИ[8] или когда ПО является встроенной или неотъемлемой частью готового МИ.

Примеч. 1 – Настоящий стандарт может применяться при разработке и обслуживании ПО, которое само по себе является МИ. Однако, прежде чем этот тип ПО может быть введён в эксплуатацию, необходимо осуществить дополнительную деятельность по разработке на системном уровне. Эта системная деятельность не входит в область применения настоящего стандарта, но её описание приведено в IEC 82304-1[B].

Настоящий стандарт описывает процессы, предназначенные для применения к ПО, которое выполняется на процессоре или другим ПО (например, интерпретатором), выполняемым на процессоре.

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

Настоящий стандарт применяется независимо от способа доставки ПО (например: передача по сети или электронной почте, оптический диск, флеш-память или EEPROM[9]). **Сам способ доставки ПО не считается ПОМИ**. Настоящий стандарт не затрагивает вопросы валидации и окончательного утверждения МИ, даже когда МИ полностью состоит из ПО.

*Примеч. 2 – Если МИ включает **встроенное ПО, предназначенное для работы на процессоре**, то к ПО применяются требования настоящего стандарта, включая требования, касающиеся ПО неизвестного происхождения (см. 8.1.2).*

Примеч. 3 – Валидацию и другую деятельность по разработке необходимо проводить на системном уровне, прежде чем ПО и МИ могут быть введены в эксплуатацию. Эта системная деятельность не входит в область применения настоящего стандарта, но её описание приведено в соответствующих стандартах на продукцию (например, IEC 60601-1, IEC 82304-1 и т. д.).

Настоящий стандарт применяется для разработки и техподдержки ПОМИ, а также для разработки и техподдержки МИ, которые содержат ПОНП.

Использование настоящего стандарта требует от изготовителя выполнения менеджмента риска МИ в соответствии с ИСО 14971. Следовательно, когда архитектура системы МИ включает приобретённый компонент (это может быть закупленный компонент или компонент неизвестного происхождения), такой как принтер/плоттер, который содержит ПОНП, этот приобретённый компонент становится ответственностью изготовителя и должен быть включён в менеджмент риска МИ. Считается, что посредством надлежащего выполнения менеджмента риска МИ изготовитель поймёт этот компонент и признает, что он содержит ПОНП. изготовитель, применяющий настоящий стандарт, должен ввести процесс менеджмента риска ПО как часть полного процесса менеджмента риска МИ.

Техподдержка выпущенного ПОМИ относится к постпроизводственному опыту[10] работы с ПОМИ. Техподдержка ПО состоит из сочетания всех технических и административных средств, включая действия по наблюдению для обработки отчётов о проблемах, чтобы сохранить или восстановить элемент в состоянии, в котором он может выполнять требуемую функцию, а также запросы на модификацию, связанные с выпущенным ПОМИ. Например, это включает исправление проблемы, регламентированную отчётность, повторную валидацию и предупреждающие действия. См. ISO/IEC 14764[C].

[8] Так называемое «Software as Medical Device - SaMD».

[9] EEPROM (англ. Electrically Erasable Programmable Read-Only Memory) — электрически стираемое перепрограммируемое ПЗУ (ЭСППЗУ), один из видов энергонезависимой памяти (типа PROM и EPROM).

[10] См. Технический отчёт ISO/TR 20416:2020 «Изделия медицинские — Послепродажное наблюдение для производителей» (кому нужно учебное пособие – обращайтесь!)

1.3 Взаимосвязь с другими стандартами

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

1.4 Соответствие

Соответствие настоящему стандарту определяется как выполнение всех установленных в нем процессов, ДиЗ, в соответствии с классом безопасности ПО.

Примеч. – Классы безопасности ПО, назначенные каждому требованию, указываются в нормативном тексте, следующем за требованиями.

Соответствие устанавливается посредством проверки всей документации, требуемой настоящим стандартом, включая файл менеджмента риска, и оценки процессов, ДиЗ, требуемых согласно классу безопасности ПО.

Примеч. 1 – Данные оценки могут быть сделаны путём внешнего или внутреннего аудита.
Примеч. 2 – Несмотря на то что должны быть выполнены указанные процессы, ДиЗ, существует определённая гибкость в методах осуществления этих процессов и выполнения ДиЗ.
Примеч. 3 – Если какие-либо требования, содержащие словосочетание «соответствующим образом», не были выполнены, то для проведения оценки необходимо предоставить документированные обоснования.
Примеч. 4 – Термин «соответствие», используемый в стандарте ИСО/МЭК 12207 (см. вставку ниже), применяется в настоящем стандарте таким же образом.
Примеч. 5 – Соответствие устаревшего/унаследованного ПО см. в подраздел 4.4 настоящего стандарта.

§2 из ГОСТ Р ИСО/МЭК 12207-2010: Соответствие

§2.1 Предполагаемое соответствие. *Требования изложены в §§ 6, 7 и в приложении А настоящего стандарта. Настоящий стандарт устанавливает требования для ряда процессов, приемлемых для использования в течение всего жизненного цикла программного продукта или услуги. Допускается, что в отдельных проектах или в некоторых организациях может не возникнуть потребность применять все процессы, приведённые в настоящем стандарте. В этом случае применение настоящего стандарта обычно сводится к выбору ряда процессов, подходящих для организации или проекта. Существует два способа такого выбора, выполнение которых может потребовать соответствия с положениями настоящего стандарта. Любое заявление о соответствии должно быть оформлено только в одной из двух приведённых ниже форм.*

§2.2 Полное соответствие. *В заявлении о полном соответствии перечисляют процессы, которые удовлетворяют требованиям настоящего стандарта. Для доказательства полного соответствия процессов положениям настоящего стандарта демонстрируют результаты процессов.*

§2.3 Адаптированное соответствие. *В случае использования стандарта как основы для установления какой-либо совокупности процессов, которые не могут быть квалифицированы как полностью соответствующие, положения настоящего стандарта выбирают или модифицируют согласно процессу адаптации, приведённому в приложении А. Формируют адаптированный текст, в отношении которого заявляют о соответствии в результате адаптации. Соответствие в результате адаптации достигается путём доказательства того, что требования к адаптированным процессам были удовлетворены, приводя в качестве доказательства результаты процессов.*

Примеч.1 - При использовании настоящего стандарта для разработки соглашения между приобретающей стороной и поставщиком определённые положения стандарта могут быть отобраны для включения в соглашение с изменениями или без изменений. В таком случае для приобретающей стороны и поставщика более приемлемо заявлять о соответствии соглашению, нежели о соответствии настоящему стандарту.

Примеч.2 - Любой организации, использующей стандарт в качестве условия при торговле, следует конкретизировать и сделать общеизвестным минимальный набор требуемых процессов, действий и задач, определяющих соответствие поставщиков стандарту.

2 Нормативные ссылки

В настоящем стандарте использована нормативная ссылка на следующий стандарт [для датированной ссылки применяют только указанное издание ссылочного стандарта, для недатированной – последнее издание (включая все изменения)]: ISO 14971[3], Medical devices – Application of risk management to medical devices (Изделия медицинские. Применение менеджмента риска к МИ)

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

3 Термины и определения

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

Для описания декомпозиции программной системы (**ПС**) (высший уровень) настоящий стандарт использует три термина. ПС, которая впоследствии становится ПОМИ, может быть подсистемой МИ (см. IEC 60601-1-4[D]) или сама по себе являться МИ, которое затем становится программным МИ. Самым нижним уровнем, ниже которого дальнейшая декомпозиция для целей тестирования или менеджмента конфигурации ПО не проводится, является программными блок. Все уровни композиции, включая верхний и нижний уровни, могут быть названы программной составной частью (**ПСЧ**). Таким образом, ПС состоит из одного или нескольких ПСЧ, а каждая ПСЧ – из программных блоков или разделяемых ПСЧ. Ответственность за определение и степень детализации ПСЧ и программных блоков возлагается на изготовителя. Отсутствие чёткого определения этих терминов позволяет применять их ко многим разным методам разработки и типам ПО, используемым в МИ.

[11]В настоящем стандарте применены следующие термины с соответствующими определениями.

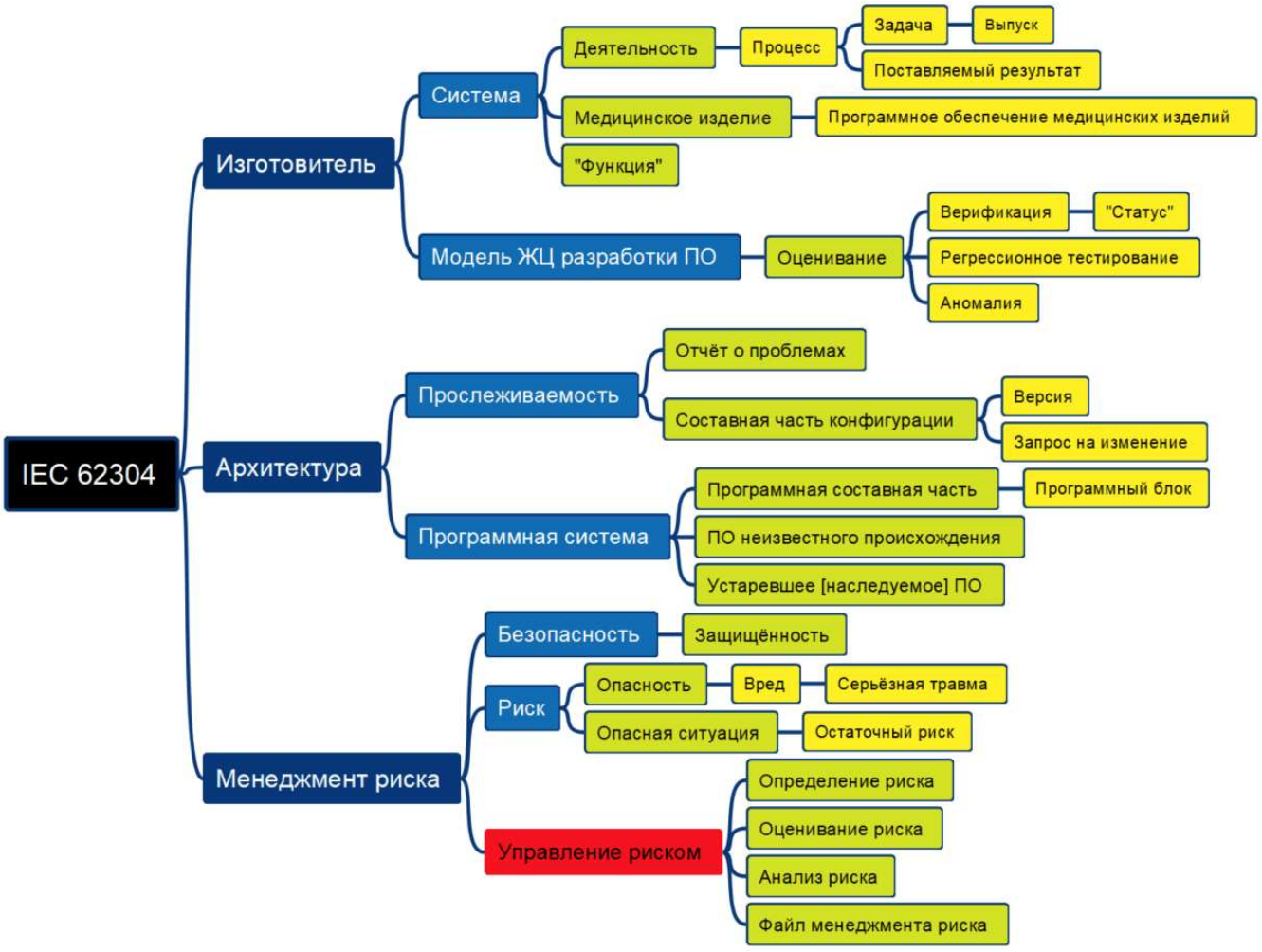
3.10 Изготовитель (Manufacturer):

Физическое или юридическое лицо, ответственное за проектирование, изготовление, упаковывание и/или маркировку МИ; установку, сборку или монтаж системы; или адаптацию МИ 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.

Примеч. 1 – К определению изготовителя могут применяться положения национального или регионального регулирования.

Примеч. 2 – Определение маркировки см. в ISO 13485:2016[12], определение 3.8.

*Наше уточнение в тему: **Организация** - лицо или группа лиц, имеющая возможность ставить собственные цели и обладающая собственными функциями с обязанностями, полномочиями и взаимосвязями для достижения этих целей*



[11] Определения перетасовал чтобы расположить их в более-менее логическом порядке…

[12] В оригинале везде стоял старый стандарт. Пришлось обновить и год и ссылки кое-где поменять…

Схема от автора пособия 1: В попытке хоть как-то структурировать понятия...

3.11 Медицинское изделие (МИ) (Medical device):

Любой инструмент, аппарат, прибор, устройство, оборудование, имплантат, *in vitro* реагент или калибратор, ПО, материал либо иные подобные или связанные с ними изделия, предназначенные изготовителем для применения к человеку по отдельности или в сочетании друг с другом в целях:

- диагностики, профилактики, мониторинга, лечения или облегчения заболеваний;
- диагностики, мониторинга, лечения, облегчения или компенсации последствий травмы;
- исследования, замещения или изменения анатомического строения или физиологических процессов;
- поддержания или сохранения жизни;
- управления зачатием;
- дезинфекции МИ;
- получения информации для медицинских целей посредством исследования *in vitro* проб, взятых из тела человека,

при условии, что их предполагаемое воздействие на человеческий организм не реализуется за счёт фармакологических, иммунологических или метаболических средств, но может поддерживаться такими средствами.

Примеч. 2 – Определения, используемые в регулировании разных стран, могут иметь некоторые различия.
Примеч. 3 – В сочетании с IEC 60601-1:2005 и IEC 60601-1:2005/AMD1:2012 термин «МИ» имеет то же значение, что и МЕ оборудование или МЕ система (которые определены терминами IEC 60601-1).

***Качество** – применительно к медицинским изделиям - свойство сохранять заданную безопасность и функциональные характеристики при доказанном уровне клинической результативности*

3.12 Программное обеспечение медицинских изделий (ПОМИ) (Medical device software):

ПС, разработанная как составная часть разрабатываемого МИ или предназначенная для использования в качестве МИ.

Примеч. – В это определение входит МИ – программный продукт, который сам по себе является МИ8.

3.24 Модель жизненного цикла разработки ПО (Software development life cycle model):

Концептуальная структура, охватывающая существование ПО от определения требований до выпуска ПО, которая:

- определяет процессы, ДиЗ, включённые в разработку ПОМИ;
- описывает последовательность и взаимозависимость между ДиЗ;
- идентифицирует этапы, на которых верифицируется полнота конкретных поставляемых результатов.

Примеч. – Основано на ISO/IEC 12207:1995, определение 3.11.

3.30 Система (System):

Интегрированный комплекс[14], состоящий из одного или более процессов, аппаратных средств, ПО, людей и средств, которая обеспечивает способность удовлетворить заявленную потребность или цель.

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

***Система менеджмента** – совокупность взаимосвязанных и взаимодействующих элементов организации для разработки политик и постановки целей, выделения и распределения функций, а также создания норм управления деятельностью для достижения этих целей.*

3.1 Деятельность (Activity):

Совокупность из одной или более взаимосвязанных или взаимодействующих задач.

[13] ***Деятельность** - способ отношения человека или организации (субъекта) к внешнему миру, состоящий в преобразовании и подчинении его целям субъекта. Характерные черты деятельности субъекта:*

- ***Сознательный характер**: субъект сознательно выдвигает цели деятельности и предвидит её результаты, продумывает наиболее целесообразные способы их достижения.*
- ***Продуктивный характер**: направлена на получение результата (продукта).*
- ***Преобразующий характер**: субъект изменяет окружающий мир (воздействует на среду специально созданными средствами труда, которые усиливают возможности субъекта).*
- ***Общественный характер**: субъект в процессе деятельности, как правило, вступает в разнообразные отношения с другими субъектами – людьми и организациями.*

[13] В данной рамочке даны определения от автора пособия.

В основе деятельности лежат потребности человека или организации. Наличие целеполагания здесь крайне обязательно. Именно наличием цели деятельность отличается от поведения.

3.14 Процесс (Process):

Совокупность взаимосвязанных и взаимодействующих видов **деятельности,** преобразующая входы в выходы.

Примеч. – Термин «Деятельность» включает использование ресурсов.

***Процесс** - устойчивая и целенаправленная СТРУКТУРА взаимосвязанных и взаимодействующих действий (элементарных актов деятельности – операций см. 4.1.1а), которая отображает в знаковой или текстовой форме преобразование по определённой технологии материала деятельности (входов в процесс) в результат деятельности (выход из процесса) - заранее запланированный продукт или услугу, представляющую ценность для потребителя при помощи инструментов деятельности (ресурсы процесса) и под управлением норм деятельности (регламентация процесса).*

***Процедура** – регламентация процесса, установленный способ выполнения процесса сформулированный в виде нормы управления деятельностью.*

3.31 Задача (task):

Отдельная часть работы, которую необходимо выполнить.

*Правильнее было бы назвать «задачу» набором операций для достижения некоторой промежуточной цели в рамках деятельности (с её главной целью), где: **операция** – это элементарный акт деятельности. Отдельное действие в ряду подобных, фаза процесса*

3.7 Оценивание (Evaluation):

Систематическое определение степени соответствия объекта установленным критериям.

[ISO/IEC 12207:2008, 4.12]

3.33 Верификация (синоним – «подтверждение соответствия») (Verification):

Подтверждение посредством предоставления объективных свидетельств того, что установленные требования были выполнены.

Прим. I Объективное свидетельство, необходимое для верификации, может быть результатом контроля или иных форм определения, таких как альтернативные расчёты или анализ документов.

Прим. II Действия, выполняемые при верификации, иногда называются процессом квалификации.

Прим. III Слово «верифицировано» («verified») используется для указания соответствующего статуса.

[Примеч. редактора пособия: определение полностью с примечаниями взято из ISO 9000:2015[E], § 3.8.13]

Примеч. 2 – При проектировании и разработке верификация относится к процессу проверки результатов конкретной деятельности, чтобы определить соответствие требованиям, установленным к этой деятельности.

*Удивительно, но термин «функция» в этом стандарте отсутствует 😳, как, впрочем, и в ISO 9000, поэтому дадим наше определение: **Функция** - использование в рамках процесса того или иного механизма взаимодействия работников (экономического, социального, информационного, административно-нормативного…) для достижения запланированной цели процесса или реализации определённых ценностей в деятельности организации.*

3.6 Поставляемый результат (Deliverable):

Требуемый итог или выход (включая документацию) деятельности или задачи.

3.3 Архитектура (Architecture):

Организационная структура системы или компонента. [IEEE 610.12:1990][F]

3.27 Программная система (ПС) (Software system):

Совокупность ПСЧ, предназначенных для выполнения конкретной функции или набора функций.

3.5 Составная часть конфигурации (Configuration item):

Объект, который может быть однозначно определён в данной конкретной точке.

Примеч. – Основано на ISO/IEC 12207:2008, 4.7.

3.29 ПО неизвестного происхождения (ПОНП) (Software of unknown provenance (SOUP)):

ПСЧ, которая уже разработана и общедоступна, но не была предназначена для включения в состав МИ (также известное как «готовое ПО») или ПСЧ, разработанная ранее, для которой недоступны требуемые записи процессов разработки.

Примеч. – ПС МИ сама по себе не может считаться ПОНП.

3.36 Устаревшее [наследуемое] ПО (Legacy software):

ПОМИ, которое было легально выпущено в обращение и по-прежнему доступно на рынке, но для которого недостаточно объективных свидетельств того, что оно было разработано в соответствии с текущей версией настоящего стандарта.

3.25 Программная составная часть - ПСЧ (Software item) (ранее переводили «Программный элемент»):

Любая идентифицируемая (выделяемая) часть компьютерной программы, т. е. исходный код, объектный код, управляющий код, управляющие данные или набор этих элементов.

*Примеч. 1 – Разделение программы на составные части можно охарактеризовать тремя терминами. **Верхний уровень** – ПС. **Самый нижний уровень**, ниже которого подразделение на составные части не осуществляется, – программный блок. Все уровни композиции, включая верхний и нижний уровни, можно назвать ПСЧ. Тогда ПС состоит из одного или более ПСЧ, и каждая ПСЧ в свою очередь состоит из одного или более программных блоков или подразделённых ПСЧ. Ответственность за обеспечение степени детализации ПСЧ и программных блоков возлагается на изготовителя.*

Примеч. 2 – Основано на ISO/IEC 90003:2004[J], 3.14 и ISO/IEC 12207:2008, 4.41.

3.26 Не используется (тут была фраза “SOFTWARE PRODUCT”).

3.28 Программный блок (Software unit) (ранее переводили как «Программный модуль»):

ПСЧ, которая не может быть разделена на более мелкие части.

Примеч. – Уровень детализации программных блоков определяется изготовителем (см. В.3).

3.15 Регрессионное тестирование (Regression testing):

Испытание, необходимое для определения того, что изменение компонента системы не повлияло отрицательно на функциональность, надёжность или производительность и не внесло дополнительных дефектов[14]. [ISO/IEC 90003:2004[J], определение 3.11]

3.4 Запрос на изменение (Change request):

Документированная спецификация изменения, которое будет сделано в ПОМИ.

3.22 Защищённость (Security):

Защита информации и данных таким образом, чтобы неавторизованные лица или системы не могли читать или изменять их, и чтобы авторизованным лицам или системам не было отказано в доступе к ним[14].

Примеч. – Основано на ISO/IEC 12207:2008, 4.39.

3.37 Выпуск (Release):

Конкретная версия составной части конфигурации, доступная для определённой цели.

Примеч. – Основано на ISO/IEC 12207:2008, определение 4.35.

3.34 Версия (Version):

Идентифицируемый отдельный вариант составной части конфигурации.

Примеч. 1— Изменение версии ПОМИ, приводящее к появлению новой версии, требует действий по управлению конфигурацией ПО. Примеч. 2 – Основано на ISO/IEC 12207:2008, 4.56.

3.32 Прослеживаемость (Traceability):

Степень, до которой может быть установлена взаимосвязь между двумя или более результатами (продуктами) процесса разработки. [IEEE 610.12:1990][F]

Примеч. – Требования, архитектура, меры по управлению риском и т. д. являются примерами поставляемых результатов процесса разработки.

3.2 Аномалия (Anomaly):

Любое условие или состояние, которое отклоняется от ожиданий, основанных на требованиях спецификаций, проектно-конструкторских документов, стандартов и т. д., либо от чьего-то восприятия или опыта. аномалии могут быть обнаружены во время проверки, тестов, анализа, компиляции, использования ПОМИ или прилагаемой документации либо в других случаях.

Примеч. – Основано на IEEE 1044:1993[G], определение 3.1.

[14] Заменил на свой перевод, т.к. ГОСТовский не нравится… 😊

3.13 Отчёт о проблемах (Problem report):

Запись о фактическом или возможном поведении ПОМИ, из которого пользователь или заинтересованное лицо могут узнать о том, что является опасным, не соответствующим предусмотренному назначению, или о том, что противоречит спецификации.

Примеч. 1 – Настоящий стандарт не требует, чтобы каждый отчёт о проблемах приводил к изменениям в ПОМИ. Изготовитель может отклонить отчёт о проблемах для неверно понятого, ошибочного или несущественного события.

Примеч. 2 – Отчёт о проблемах может относиться к готовому ПОМИ или к ПОМИ, ещё находящемуся в процессе разработки.

Примеч. 3 – Настоящий стандарт требует от изготовителя осуществлять некоторую дополнительную последовательность действий (см. § 6) по каждому отчёту о проблемах, относящемуся к уже выпущенному продукту, с целью обеспечения идентификации и выполнения предписанных действий.

3.21 Безопасность (Safety):

[15]Отсутствие недопустимого (*неприемлемого – так будет точнее*) риска [ISO 14971:2019 3.26].

3.19 Менеджмент риска (Risk management):

Систематическое применение политики, процедур и практики менеджмента к задачам анализа, оценки, контроля ~~и мониторинга~~ рисков. [ISO 14971:2019, 3.24, модифицировано – Фраза «и мониторинга» была удалена[16]]

3.16 Риск (Risk):

Сочетание вероятности причинения вреда и тяжести этого вреда. [ISO 14971:2019 3.18]

3.9 Опасность (Hazard):

Потенциальный источник вреда

3.8 Вред (Harm):

[17]Травма или ущерб здоровью людей, а также ущерб имуществу или окружающей среде.

3.23 Серьёзная травма (Serious injury):

Повреждение или заболевание, которое:

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

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

3.17 Анализ риска (Risk analysis):

Систематическое использование имеющейся информации для идентификации опасности и определения риска. [ISO 14971:2019 3.19]

3.18 Управление риском (Risk control):

Процесс, в рамках которого принимаются решения и реализуются меры, с помощью которых риски снижаются или поддерживаются в пределах установленных уровней. [ISO 14971:2019 3.21]

3.20 Файл менеджмента риска (Risk management file):

Совокупность записей и других документов, подготовленных в процессе менеджмента риска. [3.25]

3.35 Опасная ситуация (Hazardous situation):

Обстоятельство, при котором люди, имущество или окружающая среда подвергаются воздействию одной или нескольких опасностей. [Источник: ISO 14971:2019, 3.5]

[15] **Безопасность**, по сути — это главная задача риск-менеджмента.

[16] В чём заключается сакральный смысл удаления слова «мониторинг» из понятия «менеджмент риска» в контексте управления жизненным циклом ПО…в стандарте не поясняется 😐

[17] Термины связанные с рисками где-то были переведены также как в ISO 14971 на которой они ссылались, а где-то незначительно отличались, поэтому, не мудрствуя лукаво, я их просто скопировал из ISO 14971 чтобы обеспечить однозначную преемственность 😊

3.39 Определение риска (Risk estimation):

Процесс, используемый для присвоения значений вероятности наступления вреда и тяжести этого вреда. [ISO 14971:2019, 3.22]

3.40 Оценивание риска (Risk evaluation):

Процесс сравнения уже определённого риска с заданными критериями риска для определения допустимости риска. [ISO 14971:2019, 3.23]

3.38 Остаточный риск (Residual risk):

Риск, остающийся после выполнения мер по управлению риском. [ISO 14971:2019, 3.17]

Примеч. 2 – ISO/IEC Guide 51[H], определение 3.9 использует термин «защитные меры», а не «меры по управлению риском». Однако в контексте настоящего стандарта «защитные меры» являются лишь одним из вариантов[18] управления риском, как описано в 7.2 [ISO 14971:2019].

4 Общие требования

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

Есть три главных принципа, которые способствуют обеспечению безопасности ПОМИ:

- менеджмент риска;
- менеджмент качества;
- разработка ПО.

Для разработки и техподдержки безопасного ПОМИ необходимо установить менеджмент риска как неотъемлемую часть СМК, как общий каркас для приложения соответствующих методов и техник разработки ПО. Комбинация этих трех принципов позволяет изготовителю МИ следовать последовательно повторяемому процессу принятия решений, способствующему безопасности ПОМИ.

4.1 Система менеджмента качества (СМК)

Изготовитель ПОМИ должен быть способен продемонстрировать его соответствие требованиям потребителя и **применимым регулирующим требованиям**[19].

*Примеч. 1 – Демонстрация этой способности может быть осуществлена **при помощи СМК**, которая соответствует следующим требованиям:*
- *ISO 13485:2016[I] или*
- *национальному стандарту на систему менеджмента качества ГОСТ ISO 13485-2017 или*
- *СМК, требуемой национальным регулированием.*

Примеч. 2 – Руководство по применению требований СМК к ПО можно найти в ISO/IEC 90003[J].

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

Эти процессы устанавливаются СМК. ISO 13485[I] специально предназначен для применения концепции менеджмента качества к МИ. Соответствие требованиям СМК ISO 13485 не означает автоматического соответствия национальным или региональным регулирующим требованиями. Ответственность за определение и установление соответствия применимым регулирующим требованиям лежит на изготовителе.

4.2 Менеджмент риска

Изготовитель должен **применять процесс менеджмента риска в соответствии с ISO 14971**.

Разработка ПО является частью деятельности по менеджменту риска и обеспечивает рассмотрение всех обоснованно прогнозируемых рисков, связанных с ПОМИ.

Вместо того, чтобы пытаться определить подходящий процесс менеджмента риска, в настоящем стандарте требуется, чтобы изготовитель применил установленный в ISO 14971 процесс менеджмента риска, который непосредственно относится к менеджменту риска для МИ. Конкретные виды деятельности по менеджменту риска ПО, возникающего в результате опасных ситуаций, причиной которых является ПО, указана во вспомогательном процессе, описанном в § 7.

4.3 Классификация ПО в отношении безопасности

a) Изготовитель должен присвоить каждой ПС класс безопасности ПО (A, B или C) согласно риску причинения вреда пациенту, пользователю или иным лицам, исходя из опасной ситуации, в которую ПС может внести свой вклад в наихудшем сценарии, как показано на рисунке 4.

[18] Варианты «…a) изначально наиболее безопасная конструкция и производство, b) защитные меры в самом МИ или в процессе его изготовления, c) информация по безопасности и, при необходимости, обучение пользователей…».

[19] В основном по работе с ПО см. Приказ МинЗдрава РФ от 19 января 2017 года N 11н «Об утверждении требований к содержанию документации производителя МИ» (с изменениями (с изменениями на 20 ноября 2020 года). Также см. Постановления Правительства РФ от 09.02.2022 г. № 135 и № 136, Решение №106 от 10.11.17 г. Совета ЕЭК.

ПС относится к **классу безопасности ПО А**, если:

- ПС не может способствовать возникновению опасной ситуации;
- ПС может способствовать возникновению опасной ситуации, которая не приводит к недопустимому Риску после рассмотрения мер по управлению риском, внешних по отношению к ПС.

ПС относится к классу **безопасности ПО В**, если:

- ПС может способствовать возникновению опасной ситуации, которая приводит к недопустимому риску после рассмотрения мер по управлению риском, внешних по отношению к ПС, и вытекающий из этого возможный вред не является серьёзной травмой.

ПС относится к классу **безопасности ПО С**, если:

- ПС может способствовать возникновению опасной ситуации, которая приводит к недопустимому риску, после рассмотрения мер по управлению риском, внешних по отношению к ПС, и в результате возможным вредом является смерть или серьёзная травма.

Для ПС, первоначально классифицированной как класс безопасности ПО В или С, изготовитель может реализовать дополнительные меры по управлению риском, внешние по отношению к ПС (включая пересмотр архитектуры системы, содержащей ПС), и впоследствии присвоить ПС новую классификацию безопасности ПО.

Примеч. 1 – Внешними мерами по управлению риском могут быть аппаратные средства, независимая ПС, медицинские процедуры или другие средства, позволяющие свести к минимуму способность ПО приводить к возникновению опасной ситуации.

Примеч. 2 – Определение допустимости риска см. в § 4.2 «Ответственность руководства» ISO 14971:2019.

b) Не используется.

c) Изготовитель обязан документировать **класс безопасности ПО, присвоенный каждой ПС**, в файле менеджмента риска.

d) Если ПС подразделяется на ПСЧ и в дальнейшем ПСЧ в свою очередь подразделяются на ПСЧ, то такие ПСЧ должны наследовать класс безопасности первоначальной ПСЧ (или ПС), если только изготовитель не обосновывает в документации присвоение другого класса безопасности ПО (классы безопасности ПО, присвоенные в соответствии с подразделом 4.3 a), заменяя «программную систему» на «ПСЧ»). Это обоснование должно объяснять, **почему ПСЧ являются изолированными настолько**, что могут классифицироваться отдельно.

e) Изготовитель должен документировать класс безопасности ПО каждой ПСЧ, если этот класс отличается от класса ПСЧ, из которой он был выделен при декомпозиции.

f) Для соответствия настоящему стандарту там, где настоящий стандарт применяется к группе ПСЧ, изготовитель должен использовать процессы и задачи, которые требуются для классификации ПСЧ с наивысшей категорией из всей группы, если только изготовитель в файле менеджмента риска не приводит документированные обоснования для использования более низкой классификации.

g) Каждой ПС, если ей не присвоен класс безопасности ПО, п**о умолчанию должен присваиваться Класс С**.

h) *Примеч. – В последующих разделах и подразделах классы безопасности ПО, к которым применяется конкретное требование, определяются в соответствии с требованием в форме [Класс...].*

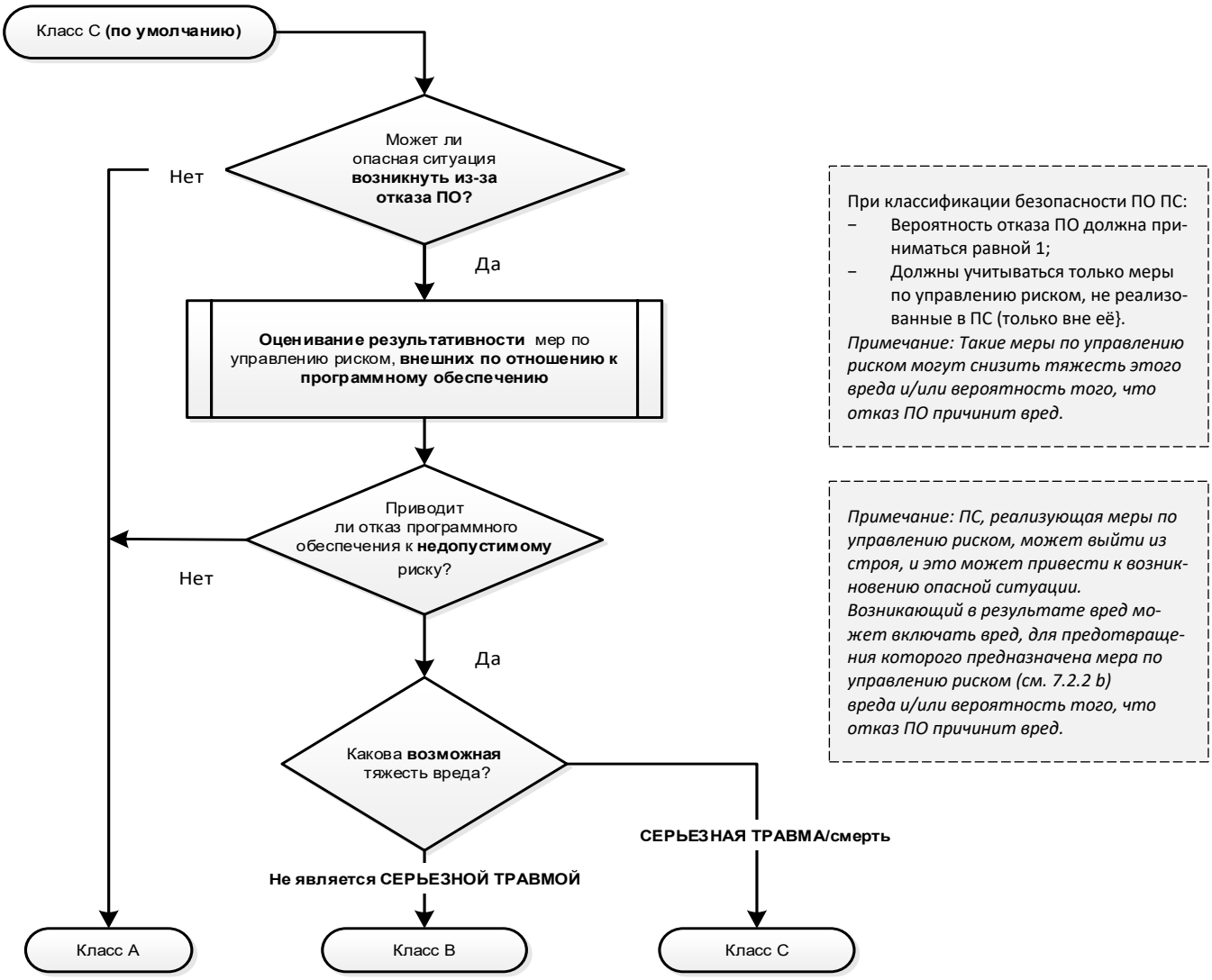


Рисунок 5 — Присвоение класса безопасности ПО

Риск, связанный с ПО (как с неотъемлемой частью МИ, или как с прикладываемым к МИ, или как с самостоятельным МИ), используется в качестве входных данных для схемы классификации безопасности ПО, которая затем устанавливает процессы, использующиеся при разработке и техподдержке ПО.

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

В некоторых случаях невозможно определить вероятность остальных событий в последовательности и риск следует оценивать только на основе характера вреда (вероятность возникновения опасной ситуации должна быть установлена равной 1). Определение риска в этих случаях должно быть сосредоточено на тяжести вреда, причинённого в результате опасной ситуации. Субъективные оценки вероятности могут быть сделаны на основе клинических знаний, чтобы отличить очевидные для медицинского специалиста отказы от тех, которые не будут обнаружены и с большей вероятностью приведут к причинению вреда.

Определение вероятности возникновения опасной ситуации, приводящей к причинению вреда (P2 на рисунке B. 2), как правило, требует клинических знаний для проведения разграничения между опасными ситуациями, в которых клиническая практика вероятней всего предотвратит вред, и опасными ситуациями, которые с большей вероятностью приведут к причинению вреда.

Если ПС подразделяется на ПСЧ, то каждая ПСЧ может иметь свой собственный класс безопасности ПО. Определить риск, связанный с отказом ПСЧ, можно только:

- если архитектура системы и архитектура ПО определяют роль ПСЧ сточки зрения её назначения и взаимодействия с другими программными и аппаратными элементами;
- если изменения в системе находятся под управлением;
- после выполнения анализа риска для данной архитектуры и установления мер по управлению риском.

Настоящий стандарт требует минимального количества

деятельности, направленной на выполнение вышеуказанных условий для всех классов ПО.

Завершение деятельности по построению архитектуры ПО является первым этапом разработки, когда определён полный набор ПСЧ и в ходе деятельности по менеджменту риска установлено, как ПСЧ связаны с безопасностью. Следовательно, это самый первый этап в разработке, на котором ПСЧ могут быть окончательно классифицированы согласно их роли в обеспечении безопасности.

Данный этап соответствует точке, с которой в ISO 14971 начинается управление риском.

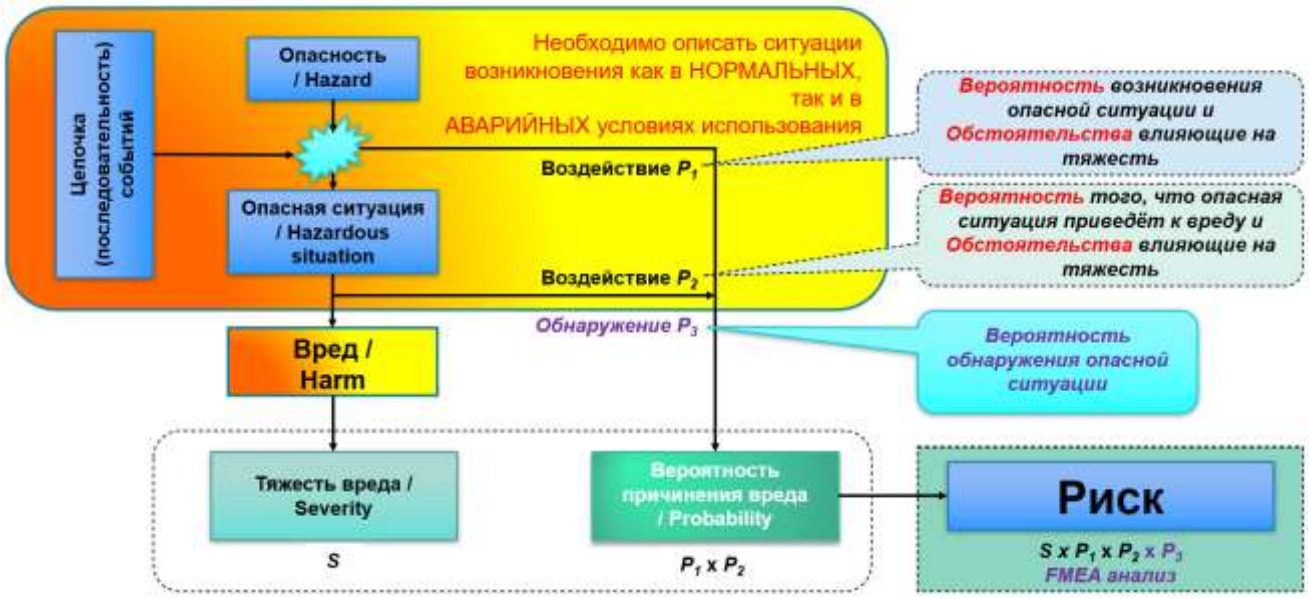


Рисунок 6(B.2) – Наглядное представление взаимосвязи опасности, последовательности событий, опасной ситуации и вреда – заимствовано из ISO 14971:2019, приложение E

До этого этапа процесс менеджмента риска идентифицирует меры по управлению риском в отношении архитектуры, например добавление защитных подсистем или уменьшение возможностей причинения вреда отказами ПО. После этого этапа процесс менеджмента риска использует процессы, направленные на снижение вероятности отказа ПСЧ. Другими словами, классификация ПСЧ определяет меры по управлению риском, основанные на процессах, которые должны к ней применяться.

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

Схема классификации безопасности ПО не предназначена для согласования с классификацией рисков согласно ISO 14971. Если в ISO 14971 классификация рисков осуществляется в соответствии с их тяжестью и вероятностью возникновения, то схема классификации безопасности ПО классифицирует ПС и ПСЧ в соответствии с процессами, которые будут применяться при их разработке и техподдержке.

По мере развития проекта могут стать очевидными новые риски. Следовательно, менеджмент риска должен применяться как неотъемлемая часть процесса разработки. Это позволяет разработать проект архитектуры, устанавливающий полный набор ПСЧ, включая необходимые для

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

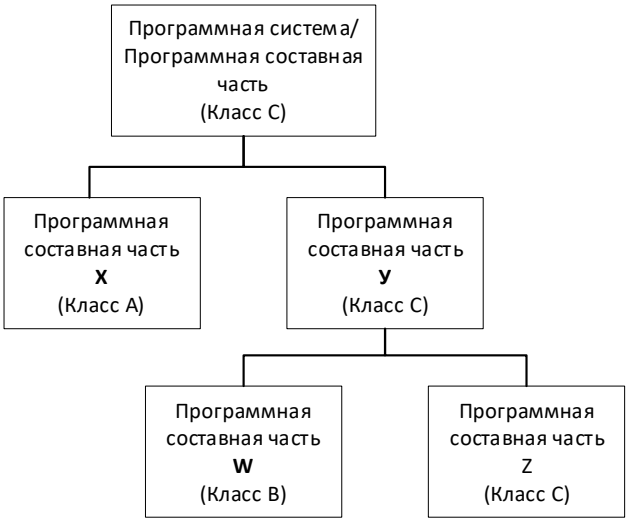


Рисунок 7(B.1) – Пример разделения ПСЧ

Архитектура ПО должна способствовать изоляции ПЭ (составных частей), которые требуются для безопасной работы, и описывать методы, используемые для обеспечения результативного разделения этих ПСЧ. Разделение не ограничивается физической изоляцией (§ процессора или памяти) и содержит любой механизм, который предотвращает негативное влияние одной ПСЧ на другую. Достаточность разделения определяется на основе связанных с этим рисков и обоснования, которое требуется задокументировать.

Как установлено в B.3, настоящий стандарт использует три термина для описания декомпозиции ПС (высший уровень).

Рис. B.1 иллюстрирует возможное разделение ПСЧ в рамках ПС и то, как классы безопасности ПО будут применяться к группе ПСЧ в процессе декомпозиции.

В приведённом примере изготовитель знает благодаря типу разрабатываемого ПОМИ, что ПС по предварительной классификации безопасности ПО относится к классу С. При проектировании архитектуры ПО изготовитель решает разделить систему на три ПСЧ – X, W и Z. Изготовитель может разделить весь вклад ПС в возникновение опасных ситуаций на приводящие к возможной смерти или серьёзной травме на вклад ПСЧ Z, и вклад всей оставшейся ПС в опасные ситуации, не приводящие к возможной серьёзной травме на вклад ПСЧ W. ПСЧ W классифицируется как класс безопасности В, а ПСЧ Z относится к классу безопасности С. ПСЧ Y, следовательно, должна быть отнесена к классу С (см. 4.3 d). В соответствии с этим требованием ПС также получает класс безопасности С. ПСЧ X относится к классу безопасности А. изготовитель может документировать обоснование разделения ПСЧ X и Y, а также ПСЧ W и Z, чтобы обеспечить целостность разделения. Если разделение между ПСЧ-ми X и Y невозможно, то ПСЧ X должна быть отнесена к классу безопасности С.

4.4 Устаревшее/наследуемое ПО

4.4.1 Общие сведения

В качестве альтернативы применению §§ 5—9 настоящего стандарта соответствие устаревшего/наследуемого ПО может быть продемонстрировано, как указано в 4.4.2—4.4.5.

4.4.2 Деятельность по менеджменту риска (Risk management activities)

В соответствии с 4.2 настоящего стандарта изготовитель должен:

a) Оценить любую обратную связь, включая пост-производственную информацию, об устаревшем ПО, касающуюся инцидентов и/или близких к ним ситуаций как внутри своей организации, так и/или от пользователей;
b) выполнить деятельность по управлению риском, связанную с продолжением использования устаревшего ПО, с учётом следующих аспектов:
   - интеграции устаревшего ПО в общую архитектуру МИ;
   - постоянной пригодности мер по управлению риском, реализованных в рамках устаревшего ПО;
   - идентификации опасных ситуаций, связанных с продолжением использования устаревшего ПО;
   - идентификации потенциальных причин, по которым устаревшее ПО может привести к возникновению опасной ситуации;
   - определения мер по управлению риском для каждой потенциальной причины, по которой устаревшее ПО способствует возникновению опасной ситуации.

4.4.3 Анализ несоответствия ожидаемых и реализованных результатов (GAP analysis)

Основываясь на классе безопасности устаревшего ПО (см. 4.3), изготовитель должен провести анализ несоответствия ожидаемых и реализованных поставленных результатов по сравнению с теми, которые требуются в соответствии с 5.2, 5.3, 5.7 и § 7.

a) Изготовитель должен оценить непрерывную пригодность полученных результатов.
b) В случае идентификации несоответствий ожидаемых и реализованных результатов изготовитель должен оценить возможное снижение риска в результате получения отсутствующих результатов и связанной с ними деятельности.
c) На основе этого оценивания изготовитель должен определить результаты, которые должны быть получены, а также связанную с ними деятельность, которая должна быть выполнена. Минимальным поставляемым результатом должны быть записи тестирования ПС (см. 5.7.5).

Примеч. – Анализ несоответствий ожидаемых и реализованных результатов должен обеспечивать включение в требования к ПО мер по управлению риском, реализованных в устаревшем ПО.

4.4.4 Деятельность по устранению несоответствий ожидаемых и реализованных результатов

a) Изготовитель должен разработать и выполнить план по созданию идентифицированных поставляемых результатов. Там, где это возможно, объективные свидетельства могут быть использованы для формирования требуемых поставляемых результатов без выполнения деятельности, установленной в 5.2, 5.3, 5.7 и § 7.

Примеч. – План устранения идентифицированных несоответствий ожидаемых и реализованных результатов может быть включён в план техподдержки ПО (см. 6.1).

b) План должен предусматривать использование процесса решения проблем для обработки проблем, обнаруженных в устаревшем ПО и поставляемых результатах, в соответствии с § 9.
c) Изменения в устаревшем ПО должны быть выполнены в соответствии с § 6.

4.4.5 Обоснование использования устаревшего/наследуемого ПО

Изготовитель **должен задокументировать версию устаревшего ПО вместе с обоснованием дальнейшего**

использования устаревшего ПО на основе результатов 4.4.

Примеч. – Выполнение 4.4 позволяет в дальнейшем использовать устаревшее ПО в соответствии с IEC 62304.

Подраздел 4.4 устанавливает процесс применения настоящего стандарта к устаревшему ПО. В некоторых регионах может потребоваться, чтобы изготовитель продемонстрировал соответствие стандарту для получения одобрения регулирующих органов ПОМИ, даже если это ПО было разработано до появления текущей версии стандарта (устаревшее ПО). В этом случае требования § 4.4 предоставляют изготовителю способ продемонстрировать соответствие устаревшего ПО настоящему стандарту.

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

- необходимая деятельность и итоговая документация должны основываться на существующей документации и использовать её, где это возможно,
- изготовитель должен использовать ресурсы для максимально результативного снижения риска.

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

Риски, связанные с планируемым продолжением использования устаревшего ПО, зависят от контекста, в котором оно будет использоваться для создания ПС. изготовитель будет документировать все выявленные опасности МИ, связанные с устаревшим ПО.

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

- неблагоприятные события, связанные с изделием,
- отзывы, полученные от пользователей изделия, и
- аномалии, обнаруженные изготовителем.

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

Определение изготовителем того, как устаревшее ПО будет использоваться в общей архитектуре системы МИ, является вкладом в оценку риска. Риски, которые необходимо учитывать, соответственно различаются.

- Когда устаревшее ПО было безопасно и надёжно использовано и изготовитель желает продолжить его применение, то обоснование дальнейшего использования должно базироваться в первую очередь на оценке риска, проведённой на основе пост-производственных записей.
- При повторном использовании устаревшего ПО для создания новой ПС предполагаемое использование устаревшего ПО может отличаться от его первоначального предназначения. В этом случае оценка риска должна учитывать изменённый набор опасных ситуаций, которые могут возникнуть из-за отказов устаревшего ПО.
- Повторно используемое устаревшее ПО может использоваться по аналогичному назначению, но без интеграции в новую программную систему. В этом случае оценка риска должна учитывать изменение мер по управлению риском архитектуры в соответствии с 5.3.

Когда устаревшее ПО будет изменено и использовано в новой ПС, изготовителю следует рассмотреть вопрос о том, как существующие записи о безопасной и надёжной работе могут быть признаны недействительными в результате изменений.

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

Согласно 4.4 изготовитель проводит анализ разрывов (пробелов), чтобы определить доступную документацию, включая объективные свидетельства выполненных задач, созданную во время разработки устаревшего ПО, и сравнить её с 5.2, 5.3, 5.7 и § 7. Типичные шаги для выполнения данного анализа разрывов (пробелов) включают:

a) идентификацию устаревшего ПО, включая версию, редакцию и любые другие средства, необходимые для чёткой идентификации;
b) Оценку существующих поставляемых результатов, соответствующих результатам, требуемым 5.2, 5.3, 5.7 и § 7;
c) Оценку имеющихся объективных свидетельств, документирование ранее применявшейся модели жизненного цикла разработки ПО (при необходимости);
d) Оценку адекватности существующей документации по менеджменту риска с учётом ISO 14971.

Принимая во внимание проведённый анализ разрывов (пробелов), изготовитель оценит потенциальное снижение риска в результате создания недостающих поставляемых результатов и связанной с ними деятельности, а также разработает план выполнения деятельности и создания недостающих пока поставляемых результатов для закрытия этих разрывов.

Снижение риска должно уравновешивать преимущества применения процесса разработки ПО в соответствии с

§ 5 с возможностью того, что модификация устаревшего ПО без полного знания истории его разработки может привести к появлению новых дефектов, которые увеличивают риск. Некоторые элементы § 5 могут быть оценены как имеющие незначительное влияние или полностью не влияющие на снижение риска, когда это делается постфактум. Например, детальное проектирование и верификация модуля снижают риск в первую очередь в процессе разработки нового ПО или рефакторинга существующего ПО. Если эти цели не запланированы, изолированное выполнение деятельности может привести к созданию документации, но не приведёт к снижению риска.

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

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

Обоснование даёт положительный аргумент в пользу безопасной и надёжной работы устаревшего ПО в контексте планируемого повторного использования, принимая во внимание как записи о постпроизводстве, доступные для устаревшего ПО, так и меры ПО управлению риском, связанные с заполнением пробелов в процессе.

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

5 Процесс разработки ПО

5.1 Планирование разработки ПО

5.1.1 План разработки ПО

Изготовитель должен создать план (или планы) разработки ПО с целью провести всю необходимую деятельность в отношении процесса разработки ПО, соответствующий области, значимости и классу безопасности разрабатываемой ПС. Модель жизненного цикла разработки ПО должна быть либо полностью определена, либо указана в плане (или в планах). **План должен содержать**:

a) Процессы, которые будут использоваться при разработке ПС (см. Примеч. 4);
b) Поставляемые результаты (включая документацию) ДиЗ;
c) Прослеживаемость между требованиями системы, требованиями ПО, испытанием ПС и мерами по управлению риском, включёнными в ПО;
d) конфигурацию ПО и менеджмент изменений, включая составные части конфигурации ПОНП (ПО неизвестного происхождения), и ПО, используемого для поддержки разработки;
e) решение проблем с ПО для обработки проблем, обнаруженных в ПОМИ, поставляемых результатах и деятельности на каждой стадии жизненного цикла. [Классы A, B, C]

Примеч. 1— Модель жизненного цикла разработки ПО может определять различные элементы (процессы, деятельность, задачи и поставляемые результаты) для различных ПСЧ в соответствии с классами безопасности ПО для каждой программной составной части программной системы.

Примеч. 2 – ДиЗ могут перекрываться или взаимодействовать и могут выполняться итеративно или рекурсивно. Это не подразумевает того, что должна использоваться определённая модель жизненного цикла.

Примеч. 3 – Другие процессы описываются в настоящем стандарте отдельно от процесса разработки. Это не подразумевает того, что они должны быть реализованы в виде отдельной ДиЗ. ДиЗ других процессов могут быть включены в процесс разработки.

Примеч. 4 – План разработки ПО может ссылаться на существующие процессы или определять новые.

Примеч. 5 – План разработки ПО может быть включён в план разработки общей системы.

5.1.2 Поддержание плана разработки ПО в актуальном состоянии (Keep software development plan updated)

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

5.1.3 План разработки ПО относительно проектирования и разработки системы (Software development plan reference to system design and development)

a) Изготовитель должен указать требования системы в плане разработки ПО в качестве входных данных.
b) В план разработки ПО изготовитель должен включать или ссылаться на процедуры по координации разработки ПО с разработкой системы, необходимой для выполнения требований 4.1 (например, системная интеграция, верификация и валидация). [Классы A, B, C]

Примеч. – Может не существовать различий между требованиями ПС и требованиями системы, если ПС является отдельной системой (например, если ПО само по себе является изделием).

5.1.4 Стандарты, методы и инструменты планирования разработки ПО

В план разработки ПО изготовитель должен включать или ссылаться:

a) на стандарты;
b) методы;
c) инструменты, связанные с разработкой ПСЧ класса С. [Класс С]

5.1.5 Программная интеграция и планирование тестирования интеграции

Изготовитель должен включить в план разработки ПО план интеграции ПСЧ (включая ПОНП) или сослаться на него, а также выполнить тестирование во время интеграции. [Классы В, С]

Примеч. 1 – Допускается объединение тестирования интеграции и тестирования ПС в единые план и совокупность деятельности.

Примеч. 2 – См. 5.6.

5.1.6 Планирование верификации ПО

В план разработки ПО изготовитель должен включить или сослаться на следующую информацию по верификации:

a) Поставляемые результаты, требующие верификации;
b) требуемые верификационные задачи для каждой деятельности в жизненном цикле;
c) контрольные точки, на которых верифицируются поставляемые результаты;
d) критерии приёмки для верификации поставляемых результатов. [Классы А, В, С]

5.1.7 Планирование менеджмента риска ПО

В план разработки ПО изготовитель должен включать или ссылаться на план осуществления процесса менеджмента риска ПО в отношении ДиЗ, включая менеджмент риска, применяющийся к ПОНП. [Классы А, В, С]

5.1.8 Документация по планированию (Documentation planning)

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

a) титульный лист, наименование или обозначение;
b) цель;
c) процедуры и ответственность за разработку, анализ, одобрение и модификацию. [Классы А, В, С]

Примеч. – Информация для рассмотрения менеджмента конфигурации документации приведена в § 8.

5.1.9 Планирование менеджмента конфигурации ПО (Software configuration management planning)

В план разработки ПО изготовитель должен включать или ссылаться на информацию о менеджменте конфигурации ПО. Эта информация должна содержать или ссылаться:

a) на классы, типы, категории или списки элементов, подлежащих управлению;
b) ДиЗ по менеджменту конфигурации ПО;
c) структуру (структуры), отвечающую(ие) за деятельность по менеджменту конфигурации ПО;
d) их взаимосвязь с другими структурами, такими как разработка или техподдержка ПО;
e) случаи, когда элементы должны находиться под управлением конфигурации;
f) случаи, когда следует использовать процесс решения проблем. [Классы А, В, С]

Примеч. – См. § 8.

5.1.10 Поддержка элементов, подлежащих управлению

Элементы, подлежащие управлению, должны включать инструменты, элементы или настройки, использующиеся для разработки ПОМИ, которые могут воздействовать на ПОМИ. [Классы В, С]

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

Примеч. 2 – См. § 8.

5.1.11 Управление составными частями конфигурации ПО до верификации

Изготовитель должен запланировать размещение составной части конфигурации под управление менеджмента конфигурации прежде, чем они будут верифицированы. [Классы B, C]

5.1.12 Идентификация и предотвращение распространённых дефектов ПО

Изготовитель должен включить или указать в плане разработки ПО процедуру:

a) для определения категорий дефектов, которые могут быть введены на основе выбранной технологии программирования и имеют отношение к их ПС;
b) документирования свидетельств, демонстрирующих неспособность этих дефектов приводить к недопустимому риску.

Примеч. – Примеры категорий дефектов или причин, способствующих возникновению опасных ситуаций, см. в приложении B из IEC/TR 80002-1:2009.

[Классы B, C]

Целью данной деятельности является планирование задач разработки ПО для уменьшения рисков, вызываемых ПО, сообщение задач и целей участникам группы разработки, а также обеспечение выполнения требований к качеству системы для ПОМИ.

Деятельность по планированию разработки ПО может документировать задачи в едином плане или в различных планах. Некоторые изготовители могут устанавливать политики и процедуры, которые применяются к разработке всего ПО для своих МИ. В этом случае план может просто ссылаться на существующие политики и процедуры. Некоторые изготовители могут подготовить план или набор планов для разработки каждого ПОМИ, которые влекут за собой детально установленные виды деятельности и ссылаются на общие процедуры. Другая возможность состоит в том, что план или набор планов приспособлен для разработки каждого ПОМИ. Планирование следует определять на уровне детализации, необходимой для осуществления процесса разработки, и он должен быть пропорционален риску. Например, системы или элементы с более высокой степенью риска должны подчиняться процессу разработки с более строгими требованиями, а задачи следует излагать более детально.

Планирование является итеративной деятельностью, которую следует пересматривать и обновлять по мере развития разработки. План может развиваться, чтобы включать большую и лучшую информацию, по мере того как больше узнают о системе и уровне усилий, необходимых для развития системы. Например, начальная классификация безопасности ПО системы может измениться в результате осуществления процесса менеджмента риска и развития архитектуры ПО. В некоторых случаях может быть принято решение о включении ПОНП в систему. Важно, чтобы планы обновлялись с целью отразить в них текущие знания о системе и уровне усилий, необходимом для системы или её элементов, чтобы обеспечить надлежащее управление процессом разработки.

5.2 Анализ требований к ПО

5.2.1 Отделение и документирование требований к ПО на основе требований системы

Для каждой ПС МИ изготовитель должен определить и документировать требования к ПС исходя из требований уровня системы. [Классы A, B, C]

Примеч. – Может не существовать различий между требованиями ПС и требованиями системы, если ПС является отдельной системой (например, если ПО само по себе является изделием).

5.2.2 Содержание требований к ПО (Software requirements content)

Как применимые и подходящие в отношении ПОМИ, изготовитель должен включать в требования к ПО:

В соответствии с требованиями к ПОМИ, изготовитель должен включить в требования к ПО:

a) требования к потенциальным возможностям и функциональности.

Примеч. 1 – Примеры включают:
- *характеристики, связанные с выполнением (например, цели/назначение ПО, требования по синхронизации);*
- *физические характеристики (например, язык машинного кода, платформу, операционную систему);*
- *компьютерное окружение (например, аппаратные средства, размер памяти, процессор, часовой пояс, инфраструктуру сети);*
- *необходимость совместимости с модернизациями или многими ПОНП или другими версиями изделий;*

b) входные и выходные данные ПС.

Примеч. 2 – Например:
- *характеристики данных (например, цифровые, буквенно-цифровые, формат);*
- *диапазоны;*
- *пределы;*
- *значения по умолчанию;*

c) интерфейсы между ПС и другими системами;

d) программные средства управления сигналами тревоги, предупреждениями и оповещением оператора;
e) требования к защищённости.

Примеч. 3 – Например:
- *связанные с компромиссом относительно конфиденциальной информации;*
- *идентификация;*
- *авторизация;*
- *системный журнал;*
- *непрерывность связи;*
- *система безопасности/защита от взлома;*

f) требования пользовательского интерфейса, реализованные в ПО.

Примеч. 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.*

Примеч. 5 – Информацию относительно требований к разработке удобства и простоты использования (эксплуатационной пригодности) можно найти в IEC 62366-1[K] среди прочих стандартов (например, IEC 60601-1-6[L]);

g) определение данных и требований к базе данных.

Примеч. 6 – Например:
- *форма;*
- *размерность;*
- *функция;*

h) требования по инсталляции и приёмке поставляемого ПОМИ для разработки и техподдержки сайта или сайтов;
i) требования, относящиеся к методам разработки и техподдержки;
j) требования, связанные с аспектами информационных сетей.

Примеч. 9[20] – Примеры включают аспекты, которые связаны:
- *с сетевыми сигналами тревоги, предупреждения и сообщения оператора;*
- *сетевыми протоколами;*
- *обработкой недоступности сетевых услуг;*

k) требования к поддержке пользователей;
l) регулирующие требования.

Примеч. 10 – Требования с a) до l) могут пересекаться.

[Классы A, B, C]

Примеч. 7 – Все эти требования могут не иметься в наличии на момент начала разработки.

Примеч. 8 – Среди прочих, ISO/IEC 25010[M] приводит информацию о качественных характеристиках, которая может быть полезна при определении требований к ПО.

5.2.3 Включение мер управления риском в требования к ПО

Изготовитель должен включать меры по управлению риском, реализованные в ПО в требования, соответствующие ПОМИ. [Классы B, C]

Примеч. – Эти требования могут быть недоступны в начале процесса разработки и изменяться по мере того, как создаётся ПО и устанавливаются дальнейшие меры по управлению риском.

5.2.4 Переоценивание анализа риска МИ (Re-evaluate medical device risk analysis)

Изготовитель должен переоценить анализ риска МИ, когда требования к ПО установлены, и, соответственно, обновить эти требования по результатам переоценки. [Классы A, B, C]

5.2.5 Обновление требований

Изготовитель должен удостовериться, что существующие требования, включая требования к системе, переоценены и обновлены, в соответствии с результатами деятельности по анализу требований к ПО. [Классы A, B, C]

[20] Нумерация примечаний такая сбитая из-за того что были внесены изменения – в старой версии стандарта в пункте 5.2.3 было всего восемь примечаний, Amd. добавило 9 и 10 примечания.

5.2.6 Верификация требований к ПО

Изготовитель должен верифицировать и документировать, что требования к ПО:

a) включают требования к системе, в том числе относящиеся к управлению риском;
b) не противоречат друг другу;
c) выражены в терминах, которые избегают двусмысленности;
d) формулируются в терминах, которые позволяют установить критерии тестирования и осуществить тестирование;
e) могут быть идентифицированы уникальным образом;
f) являются прослеживаемыми в отношении требований к системе или к другому источнику.

[Классы A, B, C]

Примеч. – Настоящий стандарт не требует использования формально установленного языка.

Данная деятельность требует от изготовителя установить и верифицировать требования к ПО для ПОМИ. Установление верифицируемых требований крайне важно для определения того, что должно быть создано, для определения, что ПОМИ функционирует должным образом, а также для демонстрации, что завершённое ПОМИ готово к использованию. С целью демонстрации имплементации требований согласно замыслу, каждое требование должно быть установлено таким образом, чтобы можно было установить объективные критерии для проверки правильной имплементации. Если процесс менеджмента риска изделия предъявляет требования к ПО для управления выявленными рисками, эти требования должны быть идентифицированы в требованиях к ПО таким образом, чтобы сделать возможным прослеживание мер по управлению риском до требований к ПО. Все требования к ПО следует определять таким образом, чтобы сделать возможной демонстрацию прослеживаемой между требованием и тестированием ПО системы. Если регулирующие требования некоторых стран требуют соответствия специальным нормам или международным стандартам, данное требование должно быть документировано в требованиях к ПО. Поскольку требования к ПО устанавливают, что должно быть имплементировано в ПО, оценка требований требуется до завершения деятельности по анализу требований.

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

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

Процесс определения требований к системе лежит вне области применения настоящего стандарта. Однако решение об имплементации функционала МИ с ПО обычно осуществляется по время проектирования системы. Некоторые или все требования к системе выделяются с целью имплементации в ПО. деятельность по анализу требований к ПО заключается в анализе требований предъявляемых к ПО процессом определения требований к системе, и в получении полного набора требований к ПО, отражающих выделенные требования.

Чтобы обеспечить целостность системы, изготовитель должен предусмотреть механизм согласования внесения изменений и уточнения системных требований для исправления непрактичности, несоответствий или двусмысленностей либо в исходных системных требованиях, либо в требованиях к ПО.

Процесс сбора и анализа системных и программных требований может быть итеративным.

Настоящий стандарт не предполагает жёсткого разделения процессов на два уровня. На практике архитектура системы и архитектура ПО часто описываются одновременно, а требования к системе и ПО впоследствии документируются в многоуровневой форме.

5.3 Проектирование архитектуры ПО

5.3.1 Преобразование требований к ПО в архитектуру

Изготовитель должен преобразовать требования к ПОМИ в документированную архитектуру, которая описывает структуру ПО и идентифицирует ПСЧ. [Классы B, C]

5.3.2 Разработка архитектуры для интерфейсов ПСЧ

Изготовитель должен разработать и документировать АРХИТЕКТУРУ для интерфейсов между ПСЧ-ми и компонентами, внешними по отношению к ПСЧ (как для программной, так и для аппаратной части), а также между ПСЧ-ми. [Классы B, C]

5.3.3 Определение требований к функциональным и эксплуатационным характеристикам элементов ПОНП

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

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

Если ПСЧ определяется как ПОНП, изготовитель должен определить аппаратные и программные средства системы, необходимые для поддержания правильной работы элемента ПОНП. [Классы B, C]

Примеч. – Примеры включают тип и скорость процессора, тип и размер памяти, тип ПО системы, коммуникационные и дисплейные требования.

5.3.5 Идентификация обособленности, необходимой для управления риском

Изготовитель должен идентифицировать любую обособленность ПСЧ, которая необходима для управления риском, и указать, как обеспечить результативность созданной обособленности. [Класс C]

Примеч. – В качестве примера создания обособленности можно привести ПСЧ, выполняемые на разных процессорах. Результативность обособленности может быть обеспечена за счёт отсутствия общих ресурсов у разных процессоров. Другие способы создания обособленности могут применяться, когда результативность может быть обеспечена посредством разработки архитектуры ПО (см. B. 4.3).

5.3.6 Верификация архитектуры ПО

Изготовитель должен верифицировать и документировать, что:

a) Архитектура ПО реализует требования к системе и к ПО, включая относящиеся к управлению риском;
b) Архитектура ПО способна поддерживать взаимодействие между ПСЧ-ми, а также между ПСЧ-ми и аппаратными средствами;
c) Архитектура МИ поддерживает правильную работу любых элементов ПОНП. [Классы B, C]

Примеч. – Для выполнения требования a) может быть использован анализ прослеживаемой архитектуры к требованиям по программному обеспечению.

Эта деятельность требует, чтобы изготовитель определил главные структурные компоненты ПО и определил их основную зону ответственности, а также их внешне видимые свойства и взаимосвязь между ними. Если функционирование компонента может влиять на другие компоненты, то оно должно быть описано в архитектуре ПО. Это описание особенно важно для аспектов функционирования, которые могут повлиять на компоненты МИ, находящиеся вне ПО (см. 5.3.5 и B.4.3). архитектурные решения чрезвычайно важны для осуществления мер по управлению рисками. Без понимания (и документирования) аспектов функционирования компонента, которые могут повлиять на другие компоненты, почти невозможно доказать, что система безопасна. архитектура ПО необходима для обеспечения правильной имплементации требований к ПО. архитектура ПО не считается завершённой, если все требования к ПО не могут быть реализованы определёнными ПСЧ-ми. Поскольку дизайн и имплементация ПО зависят от архитектуры, архитектура верифицируется для завершения этой деятельности. Как правило, верификация архитектуры выполняется путём технического Оценивания.

Классификация безопасности ПО в отношении ПСЧ в процессе деятельности по разработке архитектуры ПО создаёт основание для последующего выбора программных процессов. Записи о классификации находятся под управлением изменениями в составе файла менеджмента риска.

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

- изменения спецификации системы, программной спецификации или архитектуры;
- обнаружение ошибок в анализе рисков, особенно непредвиденных опасностей; и
- обнаружение неосуществимости требования, особенно меры по управлению риском.

Поэтому во время всех видов деятельности, следующих за разработкой архитектуры ПО, классификация ПС и ПСЧ должна переоцениваться и, если нужно, пересматриваться. Это вызывает доработку с применением дополнительных процессов к отдельной ПСЧ в результате обновления до более высокого класса. процесс менеджмента конфигурации ПО (§ 8) используется для обеспечения уверенности в том, что все необходимые доработки были идентифицированы и завершены.

5.4 Разработка детального дизайна ПО

5.4.1 Дробление ПО на программные блоки

Изготовитель должен дробить ПО, пока оно не будет представлено в виде программных блоков. [Классы B, C]

Примеч. – Некоторые ПС далее не могут быть раздроблены.

5.4.2 Разработка детального дизайна для каждого программного блока

Изготовитель должен документировать дизайн с достаточной степенью детализации, с целью обеспечения правильной реализации каждого программного блока. [Класс C]

5.4.3 Разработка детального дизайна для интерфейсов

Изготовитель должен документировать дизайн любых интерфейсов между программным блоком и внешними компонентами (аппаратными или программными средствами), а также любых интерфейсов между программный блоками, который должен быть достаточно подробным для правильной реализации каждого программного блока и его интерфейсов. [Класс C]

5.4.4 Верификация детального дизайна

Изготовитель должен верифицировать и документировать, что детальный дизайн ПО:

a) реализует архитектуру ПО;
b) не вступает в противоречия с архитектурой ПО. [Класс C]

Примеч. – Для выполнения требования a) может быть использован анализ прослеживаемости архитектуры к детальному дизайну ПО.

Данная деятельность требует от изготовителя усовершенствования ПСЧ и интерфейсов, определённых в Архитектуре, чтобы создать программные блоки и их интерфейсы. Хотя программные блоки часто считаются единичными функциями или модулями, эта точка зрения не всегда является приемлемой. Настоящий стандарт определяет программный блок как программную составную часть, не делимую на более мелкие элементы. программные блоки могут проверяться отдельно. изготовителю следует определить уровень детализации программного блока. Детальный дизайн определяет алгоритмы представления данных и взаимодействия между программными блоками и структурами данных. Детальный дизайн также должен касаться формирования программного продукта. Необходимо определить конструкцию программных блоков и интерфейсов достаточно подробно, чтобы можно было объективно верифицировать их безопасность и результативность, если это может быть обеспечено с использованием других требований или документации по разработке. Он должен быть достаточно полным, чтобы программисту не требовалось принимать исключительных проектных решений. Детальный дизайн также должен быть связан с Архитектурой ПОМИ.

ПСЧ может подразделяться на уровни так, что только немногие из новых ПСЧ имплементируют требования, связанные с безопасностью исходной ПСЧ. Оставшиеся ПСЧ не имплементируют функции, связанные с безопасностью, 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 имплементация зависит от детального дизайна, необходимо проверить данный детальный дизайн до завершения деятельности. Верификация детализированного дизайна, как правило, осуществляется путём оценивания технических характеристик. Пункт 5.4.4 требует от изготовителя верифицировать выходные данные деятельности по детализированному дизайну. Дизайн определяет, какие требования должны быть реализованы. верификация дизайна обеспечивает уверенность в том, что дизайн правильно реализует архитектуру ПО и не противоречит архитектуре ПО.

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

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

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

5.5 Имплементация программных блоков

5.5.1 Имплементация каждого программного блока

Изготовитель должен имплементировать каждый программными блок. [Классы A, B, C]

5.5.2 Установление процесса верификации программного блока

Изготовитель должен установить стратегии, методы и процедуры для верификации программных блоков. Там, где верификация осуществляется посредством тестирования, правильность процедур проведения тестирования должна быть оценена на адекватность. [Классы B, C]

Примеч. – Допускается объединение интеграционного тестирования и тестирование ПС в единый план и совокупность деятельности.

5.5.3 Критерии приёмки программных блоков

Изготовитель должен установить критерии приёмки для программных блоков до их интеграции в более крупные ПСЧ и удостовериться, что программные блоки соответствуют критериям приёмки. [Классы B, C]

Примеч. – Примеры критериев приёмки:
- *Имплементированы ли требования в программном коде, включая меры по управлению риском?*
- *Нет ли в программном коде противоречий с проектом (дизайном) интерфейса программных блоков?*
- *Соответствует ли программный код процедурам программирования или стандартам кодирования?*

5.5.4 Дополнительные критерии приёмки программных блоков

Если детальный дизайн разработан, изготовитель должен включить в дизайн дополнительные критерии приёмки, предназначенные:

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

[Класс C]

5.5.5 Верификация программных блоков

Изготовитель должен выполнять верификацию программных блоков и документировать результаты. [Классы B, C]

Данная деятельность требует от изготовителя записать и проверить код для программных блоков.

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

Пункт 5.5.5 требует от изготовителя проверять код. Если код не реализует дизайн правильно, ПОМИ не будет функционировать так, как предназначено.

5.6 Интеграция ПО и тестирование интеграции

5.6.1 Интеграция программных блоков

Изготовитель должен интегрировать программные блоки согласно плану интеграции (см. 5.1.5). [Классы B, C]

5.6.2 Верификация интеграции ПО

Изготовитель должен верифицировать, что программные блоки были интегрированы в ПСЧ и/или программную систему в соответствии с планом интеграции (см. 5.1.5), а также сохранить записи, свидетельствующие о проведении такой верификации. [Классы B, C]

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

5.6.3 Интеграционное тестирование ПО

Изготовитель должен тестировать интегрированные ПСЧ в соответствии с планом интеграции (см. 5.1.5) и документировать полученные результаты. [Классы B, C]

5.6.4 Содержание тестирования интеграции ПО

При тестировании интеграции ПО изготовитель должен установить, что интегрированная ПСЧ функционирует в соответствии с предусмотренным назначением. [Классы B, C]

Примеч. 1 – Примерами могут служить:
- *требуемая функциональность ПО;*
- *выполнение мер по управлению риском;*
- *определённая синхронизация и другие режимы работы;*
- *определённое функционирование внутренних и внешних интерфейсов;*
- *тестирование в ненормальных условиях, включая обоснованно прогнозируемое неправильное применение.*

Примеч. 2 – Возможно объединять тестирование интеграции и тестирование ПС в единый план и совокупность деятельности.

5.6.5 Оценивание процедур тестирования интеграции ПО

Изготовитель должен оценивать процедуры тестирования интеграции на адекватность. [Классы B, C]

5.6.6 Проведение регрессионного тестирования

По завершении интеграции ПСЧ, изготовитель должен провести регрессионное тестирование, подходящее для демонстрации того, что в ранее интегрированном ПО не были обнаружены дефекты. [Классы B, C]

5.6.7 Содержание записей в отношении регрессионного тестирования

Изготовитель должен:

a) документировать результаты тестирования (соответствует, не соответствует и перечень аномалий);
b) сохранить существенные записи с целью сделать возможным повторное тестирование;
c) указать лицо, которое проводило тестирование.

[Классы B, C]

Примеч. – Требование b) может быть выполнено путём сохранения, например:
- *спецификаций тестового примера, показывающих требуемые действия и ожидаемые результаты;*
- *записей об оборудовании;*
- *записей о тестовом окружении (включая программные инструменты), используемом при проведении тестирования.*

5.6.8 Использование процесса решения проблем с ПО

Изготовитель должен вводить аномалии, обнаруженные во время интеграции ПО и тестирования интеграции, в процесс решения проблем с ПО. [Классы B, C]

Примеч. – см. § 9.

Данная деятельность требует от изготовителя планировать и реализовывать интеграцию программных блоков в ПСЧ также, как и интеграцию ПСЧ в более сложносоставные ПСЧ, и проверять, что полученные в результате данной сборки ПСЧ функционируют так, как предназначено.

Подход к интеграции может варьироваться от неинкрементной[21] интеграции до любой формы пошаговой интеграции. Свойства собираемой ПСЧ диктуют выбираемый метод интеграции.

Тестирование интеграции ПО направлено на передачу данных и управление всей ПСЧ через внешние и внутренние интерфейсы. Внешние интерфейсы – это те, которые имеют другое ПО, включая ПО операционной системы и аппаратные средства МИ.

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, а также с ролью определённых ПСЧ в функционале изделия с большей степенью риска. Например, несмотря на то, что все ПСЧ должны быть протестированы, элементы, которые влияют на безопасность, должны подвергаться более направленным, всесторонним и детальным тестам.

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

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

[21] Подход, применяемый при тестировании или интеграции ПО когда производится независимое тестирование каждого из модулей по отдельности с последующим их объединением в единую программу, называется неинкрементным (жаргонное название — «большой взрыв»)

тестировать ПОМИ, могут потребоваться как тестирование методом «черного ящика», так и тестирование методом «белого ящика».

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

Тестирование интеграции ПО может осуществляться в моделируемой среде, на имеющемся оборудовании, или на полноценном МИ.

Пункт 5.6.2 требует от изготовителя верифицировать выходные данные деятельности по интеграции ПО. Выходные данные деятельности по интеграции ПО – это интегрированные (встроенные) ПСЧ.

Данные интегрированные ПСЧ должны функционировать должным образом, чтобы все ПОМИ функционировало правильно и безопасно.

5.7 Тестирование ПС

5.7.1 Установление тестирования в отношении требований к ПО

a) Для проведения тестирования ПС изготовитель должен установить и выполнить перечень тестов, выраженных как входные данные, ожидаемые результаты, критерии приёмки и процедуры, с целью учёта и охвата тестированием всех требований к ПО. [Классы A, B, C]

Примеч. 1 – Допускается объединять тестирование интеграции и тестирование ПС в единый план и совокупность деятельности. Также допустимо тестировать ПО на более ранних стадиях.

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

b) изготовитель должен оценить адекватность стратегий проведения верификации и тестовых процедур.

5.7.2 Применение процесса решения проблем с ПО

Изготовитель должен ввести аномалии, обнаруженные во время испытаний ПС, в процесс решения проблем с ПО. [Классы A, B, C]

5.7.3 Повторное тестирование после внесения изменений

При внесении изменений в ходе тестирования ПС изготовитель должен:

a) повторить тестирование, выполнить модифицированные тесты или дополнительные тесты, если применимо, с целью проверки результативности вносимых изменений для исправления проблем;
b) провести соответствующее тестирование, необходимое для демонстрации отсутствия возникновения непреднамеренных побочных эффектов;
c) выполнить соответствующую деятельность по менеджменту риска, как установлено в 7.4.

[Классы A, B, C]

5.7.4 Оценивание тестирования ПС

Изготовитель должен оценить целесообразность стратегий верификации и процедур тестирования.

Изготовитель должен проверить, что:

a) все требования к ПО были протестированы или иным образом верифицированы;
b) ведутся записи по прослеживаемой между требованиями к ПО и тестами или другой верификацией;
c) результаты тестирования соответствуют требуемым критериям приёмки.

[Классы A, B, C]

5.7.5 Содержание отчёта по тестированию ПС

Для обеспечения повторяемости тестов изготовитель должен документировать:

a) ссылки на конкретные процедуры тестирования с указанием требуемых действий и ожидаемых результатов;
b) результаты тестирования (соответствует, не соответствует и список аномалий);
c) версию тестируемого ПО;
d) соответствующие конфигурации тестируемого аппаратного и ПО;
e) соответствующие средства тестирования;
f) дату выполненного тестирования;
g) идентификацию лица, ответственного за проведение тестирования и запись его результатов. [Классы A, B, C]

Данная деятельность требует от изготовителя проверить функциональность ПО путём проверки того, что требования к ПО были успешно реализованы.

Тестирование ПС демонстрирует, что указанная функциональность действительно существует. Тестирование верифицирует функциональность и характеристики программы, как разработанной в соответствии с требованиями к ПО.

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

Тестирование ПС проверяет интегрированное ПО и может быть выполнено в моделируемой среде на имеющемся оборудовании или на полноценном МИ.

Когда в программную систему вносятся изменения (даже небольшие), должна быть определена степень регрессионного тестирования (но не только тестирования отдельных изменений), чтобы удостовериться в отсутствии непредусмотренных побочных явлений. Данное регрессионное тестирование (и обоснование для не полностью повторяемого тестирования ПС) должно быть запланировано и документировано. (См. B.6.3).

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

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

Пункт 5.7.4 требует, чтобы результаты тестирования ПС были оценены, чтобы обеспечить получение ожидаемых результатов.

5.8 Выпуск ПО на системном уровне

5.8.1 Обеспечение завершённости верификации ПО

Изготовитель до выпуска ПО в обращение должен обеспечить, чтобы деятельность по верификации всего ПО была полностью завершена, а результаты были оценён. [Классы A, B, C]

5.8.2 Документирование известных остаточных аномалий

Изготовитель должен задокументировать все известные остаточные аномалии. [Классы A, B, C]

5.8.3 Оценивание известных остаточных аномалий

Изготовитель должен обеспечить, чтобы все известные остаточные аномалии были оценён, с целью обеспечения отсутствия их способности содействовать возникновению недопустимых рисков. [Классы B, C]

5.8.4 Документирование выпущенных версий

Изготовитель должен документировать версию ПОМИ, которая будет выпускаться. [Классы A, B, C]

5.8.5 Документирование создания выпущенного ПО

Изготовитель должен документировать процедуру и окружение (среду), используемые для создания выпущенного ПО. [Классы B, C]

5.8.6 Обеспечение полного завершения деятельности и задач (ДиЗ)

Изготовитель должен обеспечить выполнение всей деятельности и всех задач, входящих в состав плана разработки (или плана техподдержки) ПО, наряду со связанной с ними документацией. [Классы B, C]

Примеч. —См. 5.1.3.b).

5.8.7 Архивирование ПО

Изготовитель должен хранить в архиве:

a) ПОМИ и составной части конфигурации;
b) документацию

в течение как минимум срока службы ПОМИ, установленного изготовителем, или в течение срока, установленного соответствующими регулирующими требованиями.

[Классы A, B, C]

5.8.8 Обеспечение надёжной поставки выпущенного ПО

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

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

[Классы A, B, C]

Данная деятельность требует от изготовителя документировать версию выпускаемого ПОМИ, указать, как оно было создано, и следовать соответствующим для выпуска ПО процедурам.

Изготовитель должен быть способен продемонстрировать, что ПО, созданное 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 понадобится в будущем. Он должен хранить, упаковывать и доставлять ПО способом, минимизирующим возможность повреждения или неправильного применения. Должны быть установлены определённые процедуры, чтобы обеспечить выполнение задач надлежащим образом и с последовательными результатами.

6 Техподдержка ПО

6.1 Установление плана техподдержки ПО

Изготовитель должен установить план (или планы) техподдержки ПО для выполнения ДиЗ процесса техподдержки. Этот план должен содержать:

a) процедуры:
   - для получения (установления),
   - документирования,
   - оценивания,
   - исправления,
   - отслеживания

   по обратной связи, возникающей (устанавливаемой) после выпуска ПОМИ;
b) критерии для определения того, что обратная связь является проблемой;
c) использование процесса менеджмента риска ПО;
d) использование процесса решения проблем ПО для анализа и принятия решений по проблемам, возникающим после выпуска ПОМИ;
e) использование процесса менеджмента конфигурации ПО (§ 8) для управления модификациями существующей ПС;
f) процедуры по Оцениванию и проведению:
   - обновления,
   - исправления ошибок,
   - исправлений, вносимых в коды («заплатки», «патчи»),
   - признания ПО устаревшим, обращения с ПОНП.

[Классы A, B, C]

Процесс техподдержки ПО отличается от процесса разработки ПО двумя пунктами:

- изготовителю разрешается использовать процесс, меньший, чем полный процесс разработки ПО, чтобы осуществлять быстрые изменения в ответ на неотложные проблемы;
- в ответ на программные отчёты о проблемах, относящихся к выпущенному продукту, изготовитель не только решает проблему, но ещё и выполняет локальные регулирующие требования (обычно запуская активную схему наблюдения для сбора данных о проблеме и её области и для общения с пользователями и регулирующими органами о проблеме).

Подраздел 6.1 требует, чтобы эти процессы были установлены в плане техподдержки.

Данная деятельность требует от изготовителя создания или идентификации процедуры для реализации ДиЗ по техподдержке. Чтобы выполнять корректирующие действия, управлять изменениями при техподдержке и управлять выпуском обновлённого ПО, изготовителю следует документировать и решить проблемы и запросы потребителей, а также управлять модификациями ПОМИ. Этот процесс активизируется, когда из-за проблем либо потребности в улучшении или адаптации

ПОМИ подвергается модификациям кода или изменяется сопутствующая документация. Цель состоит в сохранении целостности выпущенного ПОМИ при его модификации. Этот процесс включает перемещение ПОМИ в среду или на платформы, для которых оно первоначально не было выпущено. деятельность, предусмотренная настоящим пунктом, характерна для процесса техподдержки, однако процесс техподдержки может использовать другие процессы настоящего стандарта.

Изготовителю нужно планировать ДиЗ процесса техподдержки.

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 ПОМИ, выпущенного для использования по назначению. [Классы A, B, C]

6.2.1.2 Документирование и Оценивание обратной связи

Обратная связь должна быть документирована и оценён с целью определения существования проблемы в выпущенном ПОМИ. Любая такая проблема должна быть зарегистрирована в отчёте о проблемах (см. § 9). отчёт о проблемах должен содержать фактические или возможные неблагоприятные события и отклонения от спецификации. [Классы A, B, C]

6.2.1.3 Оценивание влияния отчётов о проблемах на безопасность

Каждый отчёт о проблемах должен быть оценён с целью определения его влияния на безопасность ПОМИ, выпущенного для использования по назначению (см. 9.2), и требуется ли изменение этого ПО для решения проблемы. [Классы A, B, C]

6.2.2 Использование процесса решения проблем ПО

Изготовитель должен использовать процесс решения проблем ПО (см. § 9) в отношении отчётов о проблемах. [Классы A, B, C]

Примеч. – Проблема может указывать на то, что ПС или ПСЧ не были правильно отнесены к классу безопасности ПО. Процесс решения проблемы может состоять в изменении класса безопасности ПО. После завершения процесса любое изменение класса безопасности ПС или её ПСЧ должно быть известно и документировано.

6.2.3 Анализ запросов на изменение

В дополнение к анализу, требуемому § 9, изготовитель должен анализировать каждый запрос на изменение с целью определения его влияния на организацию, ПОМИ, выпущенного для использования по назначению, и системы, с которыми оно взаимодействует. [Классы A, B, C]

6.2.4 Одобрение запроса на изменение

Изготовитель должен оценить и одобрить запросы на изменения, которые модифицируют выпущенное ПОМИ. [Классы A, B, C]

6.2.5 Информирование пользователей и регулирующих органов

Изготовитель должен идентифицировать одобренные запросы на изменения, которые влияют на выпущенное ПОМИ. Если требуется региональным регулированием, изготовитель должен информировать пользователей и регулирующие органы:

a) о любых проблемах в отношении выпущенного ПОМИ и последствиях длительного использования неизменённого продукта;
b) о характере любых доступных изменений в выпущенном ПОМИ и о том, как получить и установить эти изменения.

[Классы A, B, C]

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

Проблемы и другие запросы на внесение изменений могут повлиять на функциональные характеристики, безопасность или регистрацию МИ в регулирующих органах. Анализ необходим для определения каких-нибудь последствий из-за отчёта о проблемах и появятся ли какие-нибудь последствия из- за модификации, а также для устранения проблемы или выполнения запроса. Для проверки посредством анализа прослеживаемости или регрессионного анализа особенно важно, чтобы встроенные в изделие меры по управлению риском не были негативным образом изменены или модифицированы ПО, которое внедряется как часть деятельности по техподдержке ПО. Также важно убедиться, что изменённое ПО не создаёт опасной ситуации или снижает риск в ПО, которое ранее не создавало опасной ситуации или снижало риски. Классификация безопасности ПО для ПСЧ может быть изменена, если модификация ПО в настоящий момент может вызывать опасность или

уменьшать риск.

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

Главным в процессе техподдержки ПО является достаточный ответ на обратную связь, возникающую после выпуска ПОМИ. Как часть МИ, процесс техподдержки ПО должен обеспечить уверенность в том, что:

- отчёты о проблемах, связанные с безопасностью, рассматриваются и доводятся до сведения соответствующих регулирующих органов и затронутых пользователей;
- ПОМИ повторно одобряется и повторно выпускается после модификации и официального контроля, которые обеспечивают устранение проблемы и предотвращение дальнейших проблем;
- Изготовитель рассматривает, какое другое ПОМИ может быть затронуто и предпринимает соответствующие действия.

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

- анализирует отчёты о проблемах и идентифицирует все последствия этой проблемы;
- принимает решения по ряду изменений и определяет их любое побочное воздействие;
- осуществляет изменения, сохраняя при этом согласованность ПСЧ конфигурации, в том числе в файле менеджмента риска;
- верифицирует осуществление изменений.

Процесс техподдержки ПО использует процесс решения проблем ПО. процесс техподдержки ПО рассматривает отчёт о проблемах на высоком уровне (существует ли проблема, имеет ли она существенное влияние на безопасность, какие изменения необходимы и когда их осуществлять) и использует процесс решения проблем ПО для анализа отчёта о проблемах с целью обнаружения любых последствий и создания возможных запросов на изменения, которые идентифицируют все нуждающиеся в изменении составные части конфигурации, а также все необходимые шаги по верификации.

6.3 Осуществление модификации

6.3.1 Использование установленного процесса осуществления модификации

Изготовитель должен идентифицировать и выполнять любую деятельность, указанную в § 5, которую необходимо повторить в результате модификации. [Классы A, B, C]

Примеч. – Требования в отношении менеджмента риска для изменений ПО см. в 7.4.

6.3.2 Повторный выпуск модифицированной ПС

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

Примеч. – Модификации могут быть реализованы как часть полной повторно выпущенной ПС или как набор модификаций, включающий изменённые ПСЧ, а также инструменты, необходимые для установки изменений как модификации существующей ПС.

Данная деятельность требует, чтобы изготовитель использовал установленные процессы для выполнения модификации. Если процесс техподдержки не был установлен, для осуществления модификации могут использоваться подходящие задачи процесса разработки. изготовитель должен также обеспечить уверенность в том, что модификация не вызывает отрицательного влияния на другие части ПОМИ. Если ПОМИ не рассматривается как новая разработка, необходим анализ влияния модификации на все ПОМИ. Регрессионный анализ и тестирование используются для обеспечения уверенности в том, что изменение не создало проблем в других частях ПО

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

7 Процесс менеджмента риска ПО

Менеджмент риска ПО – это часть полного менеджмента риска МИ, которая не может надлежащим образом быть рассмотрена изолировано. Настоящий стандарт требует использования процесса менеджмента риска, соответствующего ИСО 14971:2019. Как определено в ИСО 14971:2019, менеджмент риска представляет собой основу для результативного менеджмента риска в отношении МИ. Одна из частей ИСО 14971:2019 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 опасным ситуациям, или ПО, которое используется для управления риском МИ. процесс менеджмента риска ПО включён в настоящий стандарт по двум причинам:

а) целевая аудитория данного стандарта должна понимать минимальные требования в отношении мер по управлению риском в зоне их ответственности – ПО;

б) общий стандарт по менеджменту риска, ИСО 14971:2019, приведённый в качестве нормативной ссылки к настоящему стандарту, не охватывает специально управление риском ПО и место управления

риском в жизненном цикле разработки ПО.

Менеджмент риска ПО – это часть общего менеджмента риска МИ. Планы, процедуры и документация, требуемые для деятельности по менеджменту риска ПО, могут быть серией отдельных документов или одним документом, или они могут быть интегрированы в деятельность по менеджменту риска МИ и в документацию, при условии, что выполняются все требования настоящего стандарта.

7.1 Анализ ПО, способствующего опасным ситуациям

7.1.1 Идентификация ПСЧ, которые могут способствовать возникновению опасных ситуаций

Изготовитель должен идентифицировать ПСЧ, которые могут способствовать возникновению опасных ситуаций, идентифицированных при осуществлении деятельности по анализу риска МИ, которая должна проводиться согласно ISO 14971[22] (см. 4.2). [Классы B, C]

Примеч. – Опасные ситуации могут являться прямым следствием отказа ПО или возникнуть в результате отказа мер по управлению рисками, которые включены в ПО.

7.1.2 Идентификация потенциальных причин, способствующих возникновению опасных ситуаций

Изготовитель должен идентифицировать потенциальные причины, по которым указанные в предыдущем пункте ПСЧ могут способствовать возникновению опасных ситуаций.

Изготовитель должен рассмотреть потенциальные причины, включая, если применимо:

- неправильную или неполную спецификацию функциональности;
- дефекты ПО, идентифицированные в определённых функциях ПСЧ;
- отказы или результаты, исходящие от ПОНП которые не ожидались;
- отказы аппаратных средств или другие дефекты ПО, которые могут привести к непредсказуемым операциям ПО;
- обоснованно прогнозируемое неправильное применение.

[Классы B, C]

7.1.3 Оценивание опубликованных списков аномалий ПОНП

Если отказ или исходящие от ПОНП результаты которые не ожидались являются потенциальной причиной того, что ПСЧ может способствовать возникновению опасных ситуаций, то изготовитель должен оценивать как минимум любой список аномалий, опубликованный поставщиком элементов ПОНП, используемых в МИ, чтобы определить,

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

7.1.4 Документирование потенциальных причин

Изготовитель должен документировать в файле менеджмента риска потенциальные причины, по которым ПСЧ может способствовать возникновению опасной ситуации (см. ISO 14971). [Классы B, C]

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

Однако вряд ли можно ожидать, что все опасные ситуации присущие изделию могут быть идентифицированы до того, как будет подготовлена программная архитектура. В то же время известно, как функции ПО будут воплощены в программных компонентах, и может быть оценён 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 надёжные стратегии для разделения компонентов ПО таким образом, чтобы они не взаимодействовали опасным способом.

7.2 Меры по управлению риском

7.2.1 Определение мер по управлению риском

В отношении каждого случая, зарегистрированного в файле менеджмента риска, при котором ПСЧ может способствовать возникновению опасной ситуации, изготовитель должен определить и документировать меры по управлению риском в соответствии с ISO 14971. [Классы B, C]

[22] См. также ГОСТ Р 55544-2013 (IEC/TR 80002-1:2009) «Программное обеспечение медицинских изделий. Часть 1. Руководство по применению ИСО 14971 к ПО МИ»

Примеч. – Меры по управлению риском могут быть реализованы в аппаратных средствах, ПО, рабочей среде или в инструкциях пользователя.

7.2.2 Меры по управлению риском, реализованные в ПО

Если мера по управлению риском реализуются как часть функций ПСЧ, то изготовитель должен:

a) включить меру по управлению риском в требования к ПО;
b) назначить каждой ПСЧ, которая способствует реализации меры по управлению риском, класс безопасности ПО, основанный на риске, которым управляет данная мера по управлению риском (см.4.3 a);
c) разработать программную составную часть в соответствии с § 5.

[Классы B, C]

Примеч. – Это требование обеспечивает дополнительное уточнение требований по рискам в ISO 14971.

7.3 Верификация мер по управлению риском

7.3.1 Проведение верификации мер по управлению риском

Выполнение каждой меры по управлению риском, документированной в 7.2, должно быть верифицировано, а сама верификация должна быть документирована. изготовитель должен проанализировать меры по управлению рисками и определить их способность привести к возникновению новой опасной ситуации. [Классы B, C]

7.3.2 Документирование любых новых последовательностей событий

Не применяется.

7.3.3 Документирование прослеживаемой

Изготовитель должен соответствующим образом документировать прослеживаемость в отношении опасностей ПО:

- от опасной ситуации до ПСЧ;
- от ПСЧ до конкретной причины в ПО;
- от причины в ПО до меры по управлению риском;
- от меры по управлению риском до верификации меры по управлению риском. [Классы B, C]

Примеч. – См. ISO 14971:2019, отчёт по менеджменту риска.

7.4 Менеджмент риска в отношении изменений ПО

7.4.1 Анализ изменений ПОМИ в отношении безопасности

Изготовитель обязан анализировать изменения в ПОМИ (включая ПОНП) с целью определения:

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

[Классы A, B, C]

7.4.2 Анализ влияния изменений ПО на выполненные меры по управлению риском

Изготовитель должен анализировать изменения ПО, включая изменения ПОНП, с целью определения возможности конфликта модифицированного ПО и выполненных мер по управлению риском. [Классы B, C]

7.4.3 Осуществление деятельности по менеджменту риска, основанной на результатах анализа

Изготовитель должен осуществить уместную деятельность по менеджменту риска, которая определена в 7.1,7.2 и 7.3, основанную на результатах проведённого анализа. [Классы B, C]

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 ПСЧ, 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 программную составную часть, идентифицировать его составные части и предоставить историю изменений, которые были в нем осуществлены.

8.1 Идентификация конфигурации

8.1.1 Установление средств идентификации составной части конфигурации

Изготовитель должен установить схему уникальной идентификации подлежащих управлению составных

частей конфигурации и их версий в соответствии с планированием разработки и конфигурации, установленной в 5.1. [Классы A, B, C]

8.1.2 Идентификация ПОНП

Для каждой составной части конфигурации ПОНП, который будет использоваться, включая библиотеки стандартов, изготовитель должен документировать:

a) наименование;
b) изготовителя;
c) уникальный указатель (обозначение) ПОНП.

[Классы A, B, C]

Примеч. – Уникальным указателем ПОНП может быть, например, версия, дата выпуска, номер патча или обозначение модернизации.

8.1.3 Идентификация документации конфигурации системы

Изготовитель должен документировать набор составных частей конфигурации и их версий, входящих в состав конфигурации ПС.

[Классы A, B, C]

Данная деятельность требует от изготовителя однозначной идентификации составных частей конфигурации ПО и их версий. Эта идентификация необходима, чтобы определять составные части конфигурации ПО и версии, которые включены в ПОМИ.

8.2 Управление изменениями

8.2.1 Одобрение запросов на изменения

Изготовитель может изменять составные части конфигурации, идентифицированные как подлежащие управлению согласно 8.1, только после того, как будет одобрен запрос на изменения. [Классы A, B, C]

Примеч. 1 – Решение одобрить запрос на изменения может быть частью процесса управления изменениями или частью другого процесса. Этот подпункт требует только того, чтобы одобрение изменения предшествовало его выполнению.

Примеч. 2 – В отношении запросов на изменения на разных стадиях жизненного цикла могут использоваться различные процессы одобрения, как это установлено в планах, см. 5.1.1 d) и 6.1 e).

8.2.2 Осуществление изменений

Изготовитель должен осуществить изменение так, как это определено в запросе на изменения. изготовитель должен идентифицировать и выполнить любую деятельность, которую нужно повторить из-за произведённых изменений, включая изменение класса безопасности ПС и ПСЧ. [Классы A, B, C]

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

8.2.3 Верификация изменений

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

Примеч. – Данный подпункт требует только верификации изменений. Он не подразумевает того, что верификация – неотъемлемая часть процесса управления изменениями. верификация должна использовать запланированные процессы, см. 5.1.1 д) и 6.1 e).

8.2.4 Обеспечение средствами для прослеживаемой изменений

Изготовитель должен поддерживать записи о взаимосвязях и зависимостях между:

a) запросами на изменение,
b) соответствующими отчётами о проблемах,
c) одобрениями запроса на изменение.

[Классы A, B, C]

Данная деятельность требует от изготовителя управлять изменениями составных частей 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 составные части конфигурации ПО и что одобренные запросы на изменения были полностью осуществлены и верифицированы.

Запросы на изменения могут быть одобрены группой по управлению изменениями, менеджером или техническим руководством согласно плану управления конфигурацией ПО. Одобренные запросы на изменение прослеживаются до фактической модификации и верификации ПО. Необходимо, чтобы каждое фактическое изменение было связано с запросом на изменение и существовала документация, показывающая, что запрос на изменение был одобрен. Документация может быть изменена группой по управлению изменениями, подписью или записью в базе данных.

8.3 Учёт статуса конфигурации

Изготовитель должен сохранять восстанавливаемые записи об истории управляемых составных частей конфигурации, включая конфигурацию системы. [Классы A, B, C]

Данная деятельность требует от изготовителя поддерживать записи истории составных частей конфигурации ПО. Эта деятельность необходима, чтобы определять, когда и где были сделаны изменения. Доступ к этой информации нужен для обеспечения уверенности в том, что составные части конфигурации ПО содержат только разрешённые модификации.

9 Процесс решения проблем ПО

9.1 Подготовка отчётов о проблемах

Изготовитель должен подготовить отчёт о проблемах в отношении каждой проблемы, обнаруженной в ПОМИ. отчёты о проблемах должны включать заключение о критичности (например, влияние на функциональные характеристики, безопасность или защищённость), а также другую информацию, которая может помочь в решении проблемы (например, затронутые устройства, затронутое вспомогательное оборудование). [Классы A, B, C]

Примеч. – Проблемы могут быть обнаружены до или после выпуска, внутри или вне организации изготовителя.

9.2 Исследование проблемы

Изготовитель должен:

- исследовать проблему и, если возможно, определить причины;
- оценить влияние проблемы на безопасность, используя процесс менеджмента риска (§ 7);
- документировать результаты исследования и оценки;
- создать запрос (запросы) на изменение в отношении действий, необходимых для исправления проблемы, или документировать объяснение того, почему никакие действия не предприняты. [Классы A, B, C]

Примеч. – Проблема не обязательно должна быть исправлена изготовителем, чтобы соответствовать процессу решения проблем ПО, при условии, что проблема не является важной для обеспечения безопасности.

9.3 Консультирование заинтересованных сторон

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

Примеч. – Проблемы могут быть обнаружены до или после выпуска, внутри организации изготовителя или вне её. изготовитель сам определяет заинтересованные стороны в зависимости от ситуации.

9.4 Использование процесса управления изменениями

Изготовитель должен одобрить и осуществить все запросы на изменения, соблюдая требования процесса управления изменениями (см. пункт 8.2). [Классы A, B, C]

9.5 Поддержание записей

Изготовитель должен поддерживать записи в отношении отчётов о проблемах и принятых решениях, включая их верификацию.

Если применимо, изготовитель должен обновлять файл менеджмента риска. [Классы A, B, C]

9.6 Анализ проблем на предмет выявления тенденций

Изготовитель должен проводить анализ с целью определения тенденции в отчётах о проблемах. [Классы A, B, C]

9.7 Верификация решения проблем ПО

Изготовитель должен верифицировать решения с целью определения:

a) была ли проблема решена и был ли завершён отчёт О проблеме;
b) были ли преодолены неблагоприятные тенденции;

c) был ли запрос на изменения реализован в соответствующем ПОМИ и деятельности;
d) появились ли дополнительные проблемы.

[Классы A, B, C]

9.8 Содержание документации по тестированию

При проведении тестирования, при повторном тестировании или регрессионном тестировании ПСЧ и систем, следующих за изменением, изготовитель должен включить в документацию по испытаниям:

a) результаты тестирования;
b) обнаруженные аномалии;
c) версию тестируемого ПО;
d) соответствующие аппаратные и тестовые конфигурации ПО;
e) соответствующие инструменты тестирования;
f) дату проведения тестирования;
g) идентификацию лица, проводившего тестирование.

[Классы A, B, C]

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

Данная деятельность требует от изготовителя использовать процесс решения проблем, когда определены проблема или несоответствие. Данная деятельность необходима, чтобы обеспечить уверенность в том, что обнаруженные проблемы проанализированы и оценён на возможное отношение их к безопасности (как определено в ИСО 14971:2019).

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

Приложение A (справочное). Обоснование требований настоящего стандарта

В данном приложении приведено обоснование положений настоящего стандарта.

A.1 Обоснование

Основным требованием настоящего стандарта является выполнение совокупности процессов, которые надлежит применять при разработке и техподдержке ПОМИ, а также выбор этих процессов, исходя из риска для пациентов и третьих лиц. Это следует из твёрдого убеждения в том, что для установления безопасности функционирования ПО одного лишь тестирования недостаточно.

Процессы, требуемые настоящим стандартом, можно разделить на **две категории**:

- процессы для определения рисков, возникающих от функционирования каждой ПСЧ в ПО;
- процессы для снижения вероятности отказа ПО для каждой ПСЧ, выбранные на основе этих определённых рисков.

Настоящий стандарт требует, чтобы первая категория процессов выполнялась для любого ПОМИ, а вторая категория – только для выбранных ПСЧ.

Следовательно, для соответствия настоящему стандарту должен быть реализован документированный анализ рисков, идентифицирующий предсказуемые последовательности событий, связанные с наличием ПО, которые могут привести к опасной ситуации (см. ISO 14971). Опасные ситуации, которые могут быть косвенно вызваны ПО (например, путём предоставления вводящей в заблуждение информации, которая может привести к назначению неверного лечения), должны быть включены в этот анализ рисков.

Вся деятельность требующаяся в рамках первой категории процессов, обозначена в нормативном тексте как «[Классы A, B, C]», указывая, что она требуется вне зависимости от класса безопасности ПО, к которому она относится.

Деятельность требуется для классов A, B и C по следующим причинам:

- деятельность создаёт план, имеющий отношение к менеджменту риска или классификации программного обеспечения по безопасности;

- деятельность производит результат, который является входными данными для менеджмента риска или классификации безопасности ПО;
- деятельность является частью менеджмента риска или классификации безопасности ПО;
- деятельность устанавливает систему управления, документации или ведения записей, которая поддерживает менеджмент риска или классификацию безопасности ПО;
- деятельность обычно имеет место, когда классификация связанного с ней ПО неизвестна;
- деятельность может вызвать изменение, приводящее к изменению класса безопасности связанного с ней ПО. Это включает обнаружение и анализ проблем связанных с безопасностью после выпуска.

Другие процессы, требующиеся только для ПС или для ПСЧ, классифицируются как классы безопасности B или C. деятельность, требующаяся как часть этих процессов, указана в нормативном тексте как «[Классы B, C]» или «[Класс C]», указывая, что она требуется в зависимости от класса безопасности ПО, к которому она относится.

Деятельность требуется выборочно для ПО классов B и C по следующим причинам:

- деятельность повышает надёжность ПО, требуя большей детальности или большей точности в дизайне, тестировании или другой верификации;
- деятельность является управленческой, поддерживающей другую деятельность, требуемую для классов B и C;
- деятельность поддерживает коррекцию связанных с безопасностью проблем;
- деятельность обеспечивает ведение записей по проекту (дизайну), имплементации, верификации и выпуску связанного с безопасностью ПО;
- деятельность требуется выборочно для класса C по следующим причинам:
- деятельность ещё больше повышает надёжность системы, требуя более тщательного, или более точного, или более внимательного отношения к отдельным вопросам проекта (дизайна), тестирования или другой верификации.

Следует отметить, что все процессы и деятельность, указанные в настоящем стандарте, являются значимыми для обеспечения разработки и техподдержки высококачественного ПО. Отсутствие многих из этих процессов и деятельности в качестве требований для ПО класса A не подразумевает, что эти процессы и деятельность не являются важными или не рекомендуются. Их отсутствие оправдано тем, что безопасность и результативность ПО, которое не может быть причиной опасности, можно обеспечить посредством совокупной деятельности по валидации в рамках проектирования МИ (что выходит за рамки области применения настоящего стандарта), а также посредством простых средств управления жизненным циклом ПО.

A.2 Краткое изложение требований по классификации

Таблица A.1 показывает, какие классы безопасности ПО назначены каждому требованию. Это информационная таблица и предоставлена она только для удобства. Нормативный § указывает классы безопасности ПО для каждого требования.

Таблица 2(A.1) – Краткое изложение требований в зависимости от классификации безопасности ПО

Пункты и подпункты		Класс A	Класс B	Класс C
§ 4	Все требования	X	X	X
5.1	5.1.1, 5.1.2, 5.1.3, 5.1.6, 5.1.7, 5.1.8 5.1.9	X	X	X
	5.1.5, 5.1.10, 5.1.11, 5.1.12		X	X
	5.1.4			X
5.2	5.2.1, 5.2.2, 5.2.4, 5.2.5, 5.2.6	X	X	X
	5.2.3		X	X
5.3	5.3.1, 5.3.2, 5.3.3, 5.3.4, 5.3.6		X	X
	5.3.5			X
5.4	5.4.1		X	X
	5.4.2, 5.4.3, 5.4.4			X
§ 4	Все требования	X	X	X
5.5	5.5.1	X	X	X
	5.5.2, 5.5.3, 5.5.5		X	X
	5.5.4			X
5.6	Все требования		X	X

Пункты и подпункты		Класс A	Класс B	Класс C
5.7	Все требования	X	X	X
5.8	5.8.1, 5.8.2, 5.8.4, 5.8.7, 5.8.8	X	X	X
	5.8.3, 5.8.5, 5.8.6,		X	X
6	Все требования	X	X	X
7.1	Все требования		X	X
7.2	Все требования		X	X
7.3	Все требования		X	X
7.4	7.4.1	X	X	X
	7.4.2, 7.4.3		X	X
§ 8	Все требования	X	X	X
§ 9	Все требования	X	X	X

Приложение B (справочное). Руководство по положениям настоящего стандарта

Все рекомендации из данного приложения перенесены в текст основного стандарта и оформлены там двухколоночной вёрсткой шрифтом Times New Roman.

Приложение C (справочное). Взаимосвязь с другими стандартами

C.1 Общие положения

Настоящий стандарт применяется к разработке и техподдержке ПОМИ. ПО может быть подсистемой МИ или являться самостоятельным МИ. Настоящий стандарт предназначен для использования совместно с другими подходящими стандартами на процессы разработки МИ.

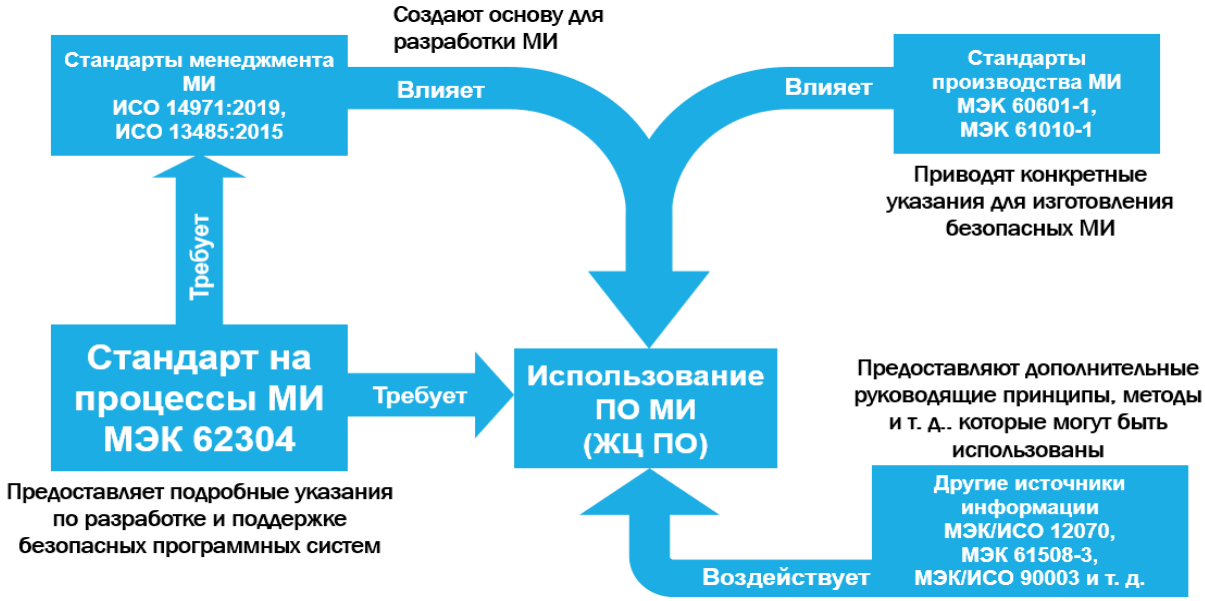


Рисунок 8(C.1) – Взаимосвязь ключевых стандартов на МИ с IEC 62304

Стандарты по менеджменту МИ, такие как ISO 13485[I] (см. C.2 и приложение D) и ISO 14971:2019, обеспечивают менеджмент окружения (среды), что закладывает основу для организации разработки продукции. Стандарты по безопасности, такие как IEC 60601-1[N] (см. приложение C.4) и IEC 61010-1[P] (см. приложение C.5), дают определённое руководство по созданию безопасных МИ. Когда ПО является составной частью таких МИ, настоящий стандарт содержит более детальное руководство относительно требований к разработке и поддержанию безопасности ПОМИ. Многие другие стандарты, такие как ISO/IEC 12207[A] (см. приложение C.6), IEC 61508-3[O] (см. приложение C.7) и ISO/IEC 90003[J], могут рассматриваться как источники методов, инструментов и техник, которые следует использовать для выполнения требований настоящего стандарта. Рисунок C.1 показывает взаимосвязь между этими стандартами.

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

C.2 Взаимосвязь с ISO 13485

Настоящий стандарт требует, чтобы изготовитель использовал систему менеджмента качества. Когда изготовитель использует ISO 13485[I], требования настоящего стандарта непосредственно связаны с требованиями ISO 13485:2016, как это показано в таблице C.1.

Таблица 3(C.1) – Взаимосвязь с ISO 13485:2016

IEC 62304	Соответствующие пункты ISO 13485:2016
1	2
5.1 Планирование разработки ПО	7.3.2 Планирование проектирования и разработки
5.2 Анализ требований к ПО	7.3.3 Входные данные для проектирования и разработки
5.3 Проектирование архитектуры ПО	
5.4 Разработка детального дизайна ПО	
5.5 Имплементация программных блоков	
5.6 Интеграция ПО и тестирование интеграции	
5.7 Тестирование ПС	7.3.4 Выходные данные проектирования и разработки 7.3.5 Анализ проекта и разработки
5.8 Выпуск ПО на системном уровне	7.3.6 Верификация проектирования и разработки 7.3.7 Валидация проектирования и разработки
6.1 Установление плана техподдержки ПО	7.3.8 Управление изменениями проекта и разработки
6.2 Анализ модификации и проблем	
6.3 Осуществление модификации	7.3.6 Верификация проектирования и разработки 7.3.7 Валидация проектирования и разработки
7.1 Анализ ПО, способствующего опасным ситуациям	
7.2 Меры по управлению риском	
7.3 Верификация мер по управлению риском	
7.4 менеджмент риска в отношении изменений ПО	
8.1 Идентификация конфигурации	7.5.8 Идентификация и 7.5.9 Прослеживаемость
8.2 Управление изменениями	7.5.8 Идентификация и 7.5.9 Прослеживаемость
8.3 Учёт статуса конфигурации	
9 Процесс решения проблем ПО	

C.3 Взаимосвязь с ISO 14971:2019

Таблица C.2 показывает области, где настоящий стандарт усиливает требования к процессу менеджмента риска, требуемого ISO 14971.

Таблица 4(C.2) – Взаимосвязь с ISO 14971:2019

Пункты ISO 14971:2019[23]	Соответствующие пункты IEC 62304
5.1 Процесс анализа риска	
§.4.2 в старом ISO 14971:2007 разбит в новой версии на два раздела: §5.2 «Предусмотренное применение и обоснованно прогнозируемое неправильное применение» и §5.3 «Определение характеристик, связанных с безопасностью»	
5.4 Идентификация опасностей и опасных ситуаций	7.1 Анализ ПО, способствующего опасным ситуациям
4.4 Определение риска	4.3 Классификация безопасности ПО
6 Оценивание риска	
7.1 Анализ возможностей управления риском	7.2.1 Определение мер по управлению риском

[23] Все заголовки из ISO 14971:2007 заменены на новые из версии 2019 года согласно табл.B.1 там же.

Пункты ISO 14971:2019[23]	Соответствующие пункты IEC 62304
7.2 Осуществление мер по управлению рисками	7.2.2 Меры по управлению риском, осуществлённых в ПО 7.3.1 Верификация мер по управлению риском
7.3 Оценивание остаточного риска	
7.4 Анализ соотношения риск/польза	
7.5 Риски, возникающие в результате мер по управлению рисками	7.3.2 Документирование любых новых последовательностей событий
7.6 Полнота управление рисками	
8 Оценивание совокупного остаточного риска	
9 Анализ менеджмента риска	7.3.3 Документирование прослеживаемости
10 Производственная и пост-производственная деятельность	7.4 Менеджмент риска изменений ПО

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

C.4.1 Общие положения

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

C.4.2 Взаимосвязь ПО с разработкой ПЭМС

Используя V-модель, показанную на рисунке C.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 ПСЧ в программную систему. Эта ПС – часть программируемой электрической подсистемы (ПЭСС), являющейся, в свою очередь, частью ПЭМС.

C.4.3 Процесс разработки

Соответствие процессу разработки ПО в настоящем стандарте (§ 5) требует, чтобы план разработки ПО был определён и соблюдался; это не требует, чтобы использовалась некая определённая модель жизненного цикла, но требует, чтобы план включал определённые виды деятельности и имел определённые признаки. Эти требования соотносятся с требованиями ПЭМС в IEC 60601 к разработке жизненного цикла, спецификации требований, архитектуре, проектированию и осуществлению, а также верификации. Требования в этом стандарте более детальны в области разработки ПО, чем требования IEC 60601-1.

C.4.4 Процесс технического обслуживания

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

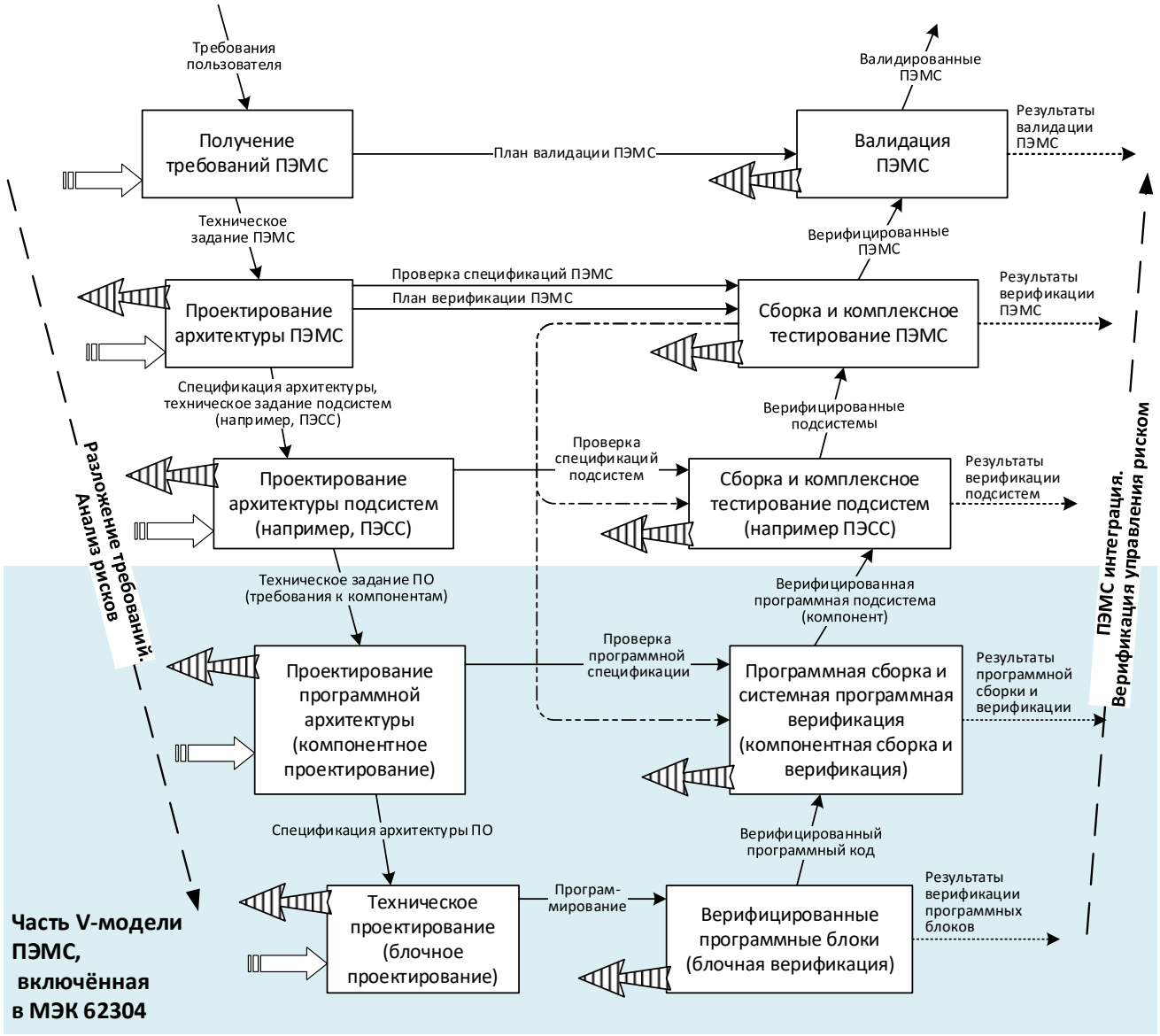
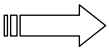


Рисунок 9(C.2) — ПО как часть V-модели

Блоки – типичные виды деятельности жизненного цикла разработки; Сплошные стрелки[24] – типичные результаты, передаваемые в/из видов деятельности; Пунктирные стрелки – результаты, относящиеся только к файлу менеджмента риска

 – Исходные данные процесса разрешения проблем

 – Результат процесса разрешения проблем

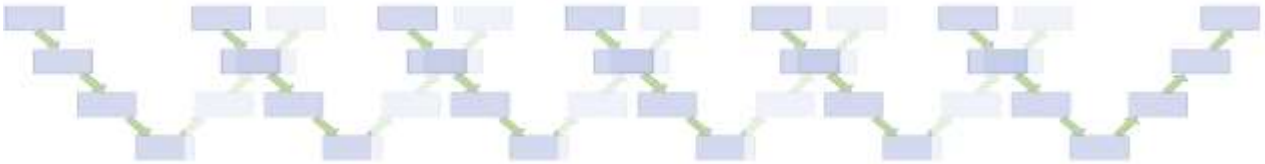


Рисунок 10 Рассуждения редактора: Как вам вариант V-модели при SCRUM-подходе?

C.4.5 Прочие процессы

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

Процесс менеджмента риска ПО в настоящем стандарте соответствует дополнительным требованиям к менеджменту риска, определённым для ПЭМС в IEC 60601-1.

Процесс решения проблем ПО в настоящем стандарте соответствует требованию к решению проблем для

[24] Необходимо заметить: стрелки являются функцией отношений между субъектами и объектами, а также субъектами и субъектами, а не функцией времени!

ПЭМС в IEC 60601-1.

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

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

Таблица 5(C.3) – Взаимосвязь с IEC 60601-1 (1 из 5)

Требования к ПЭМС в IEC 60601-1:2005 + IEC 60601-1:2005/AMD1:2012	Требования IEC 62304, связанные с ПО подсистемы ПЭМС
14.1 Общие положения Требования 14.2—14.12 (включительно) применяются к ПЭМС только в тех случаях, когда: – ни одна из программируемых электронных подсистем (ПЭСС) не задействована в обеспечении основной безопасности или основных функциональных характеристик или – применение менеджмента риска как описано в 4.2, показывает, что отказ любой ПЭСС не приводит к возникновению недопустимого риска.	**4.3 Классификация ПО в отношении безопасности** Требования к ПЭМС, установленные в IEC 60601-1, применимы только к ПО класса безопасности B и C. Настоящий стандарт содержит некоторые требования в отношении ПО класса безопасности A.
Требования 14.13 применимы к любым ПЭМС, предназначенным для включения в ИТ-СЕТЬ, независимо от применения требований 14.2—4.12. В случае применения требований 14.2—14.13, требования 4.3, §§ 5, 7, 8 и 9 IEC 62304:2006 также применяются к разработке или модификации ПО для каждого ПЭСС	Процесс разработки ПО, необходимый для соответствия IEC 60601-1, не включает последующий мониторинг и техническую поддержку, требуемые § 6 IEC 62304:2006
14.2 Документирование Документы, требуемые § 14, должны рассматриваться, утверждаться, выпускаться и изменяться в соответствии с официальной процедурой управления документацией	**5.1 Планирование разработки ПО** В дополнение к конкретным требованиям к деятельности по планированию разработки ПО документы, которые являются частью файла менеджмента риска, должны поддерживаться в соответствии с требованиями ISO 14971. Кроме того, ISO 13485[I] требует управления документами системы качества
14.3 План менеджмента риска План менеджмента риска, требуемый согласно 4.2.2, должен включать ссылку на план верификации ПЭМС (см. 14.11)	Нет специальных требований. Не существует никакого определённого плана валидации ПО. План валидации ПЭМС относится к уровню системы и находится вне области применения настоящего стандарта на ПО. Настоящий стандарт требует прослеживаемой от опасности определённого события с ПО до меры по управлению риском и к верификации меры по управлению риском (см. 7.3)
14.4 Жизненный цикл разработки ПЭМС жизненный цикл разработки ПЭМС должен быть документально оформлен.	**5.1 Планирование разработки ПО** **5.1.1 План разработки ПО Пункты, на которые ссылается план разработки ПО, составляют жизненный цикл разработки ПО**
Жизненный цикл разработки ПЭМС должен состоять из набора определённых этапов	
На каждом этапе должна быть определена деятельность, которая должна быть завершена, а также методы верификации, которые должны применяться в отношении этой деятельности	**5.1.6 Планирование верификации ПО** **Должны быть запланированы задачи верификации, этапы и критерии приёмки**
Каждая деятельность должна определяться с указанием входных и выходных параметров	**5.1.1 План разработки ПО** Вся деятельность определена в настоящем стандарте. Документация, которая должна быть разработана, определена для каждой деятельности

Требования к ПЭМС в IEC 60601-1:2005 + IEC 60601-1:2005/AMD1:2012	Требования IEC 62304, связанные с ПО подсистемы ПЭМС
На каждом этапе должны определяться работы по менеджменту риска, которые необходимо завершить перед этим этапом	**5.1.1 План разработки ПО** В соответствии с настоящим стандартом разработка жизненного цикла должна быть документирована в плане разработки. Это означает, что план разработки должен содержать разработку конкретного жизненного цикла
Жизненный цикл разработки ПЭМС должен составляться для каждой разработки путём 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 ПО** **5.1.8 Документация по планированию**
14.5 Решение проблем Если целесообразно, должна быть разработана и поддерживаться документированная система решения проблем, возникающих на каждом этапе деятельности (и между ними) жизненного цикла разработки ПЭМС	**9 Процесс решения проблем ПО**
В зависимости от типа продукции система решения проблем может: – регистрироваться как часть жизненного цикла разработки ПЭМС; – позволять уведомлять о потенциальных или возникающих проблемах, затрагивающих основную безопасность или основные функциональные характеристики ПЭМС; – включать оценку каждой проблемы с точки зрения связанных с ней рисков; – определять критерии завершения решения проблем; – определять работы, которые должны выполняться для решения каждой проблемы	**5.1.1 План разработки ПО** **9.1 Подготовка отчётов о проблемах**
14.6 Процесс менеджмента риска	**7 Процесс менеджмента риска ПО**
14.6.1 Идентификация известных и прогнозируемых опасностей При составлении перечня известных или прогнозируемых опасностей изготовитель должен учитывать те из них, которые связаны с ПО и особенностями аппаратных средств ПЭМС, включая связанные с подключением к ИТ-сетевым ресурсам, компонентами сторонних изготовителей и унаследованными подсистемами	**7.1 Анализ ПО, способствующего опасным ситуациям** Настоящий стандарт не ссылается на конкретное сопряжение с сетями и данными
14.6.2 Управление риском Для реализации каждой меры ПО управлению риском должны быть выбраны и идентифицированы надлежащим образом проверенные инструменты и процедуры. Эти инструменты и процедуры должны быть подходящими для обеспечения того, что каждая мера ПО управлению риском результативно снизит идентифицированный(е) риск(И)	**5.1.4 Стандарты, методы и инструменты планирования разработки ПО** Настоящий стандарт требует идентификации определённых 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 её подсистем должен быть разработан и задокументирован перечень требований	**5.2 Анализ требований к ПО Настоящий стандарт применим только в отношении подсистем ПО ПЭМС**
Перечень требований к системе или подсистеме должен включать и характеризовать все основные функциональные характеристики и все мероприятия по управлению риском, реализуемые в системе или подсистеме	**5.2.1 Отделение и документирование требований к ПО от требований системы** **5.2.2 Содержание требований к ПО** **5.2.3 Включение мер управления риском в требования к ПО** Настоящий стандарт устанавливает, что требования, связанные с основными функциональ-

Требования к ПЭМС в IEC 60601-1:2005 + IEC 60601-1:2005/AMD1:2012	Требования IEC 62304, связанные с ПО подсистемы ПЭМС
	ными характеристиками и мерами по управлению риском, должны отличаться от других требований, но требует однозначной идентификации любых требований
14.8 Архитектура Для ПЭМС и каждой из её подсистем должна быть установлена архитектура, удовлетворяющая перечню требований	**5.3 Проектирование архитектуры ПО**
Когда это целесообразно для снижения риска до допустимого уровня, в спецификации требований к архитектуре должны использоваться: a) компоненты с высокой степенью интеграции; b) устойчивые к отказам функции; c) избыточность; d) диверсификация; e) разделение функций; f) защищённая конструкция, служащая, например, для ограничения представляющих потенциальную опасность эффектов путём ограничения допускаемой выходной мощности или введения устройств, ограничивающих свободный ход исполнительных устройств.	**5.3.5 Идентификация обособленности, необходимой для управления риском** Разделение является единственным идентифицированным способом, и это только идентификация, потому что требование состоит в точном определении того, что целостность разделения обеспечена
Спецификация архитектуры должна также учитывать: a) распределение мер по управлению риском в подсистемах и компонентах ПЭМС; b) виды отказов компонентов и их последствия; c) неспецифические отказы; d) систематические отказы; e) период проведения тестирования или диагностики; f) ремонтопригодность; g) защиту от прогнозируемых ошибок в применении; h) если применимо, требования к ИТ-сетевым ресурсам	Это не включено в настоящий стандарт
14.9 Проектирование и реализация Когда это целесообразно, проектирование должно проводиться для отдельных подсистем, каждая из которых должна иметь собственные требования к разработке и требования к испытаниям	**5.4 Разработка детального дизайна ПО** **5.4.2 Разработка детального дизайна для каждого программного блока** Настоящий стандарт не требует спецификации испытаний для детализированного проекта
Пояснения относительно условий проектирования должны включаться в документацию	**5.4.2 Разработка детального дизайна для каждого программного блока**
14.10 Верификация Верификация требуется для всех функций, которые обеспечивают основную безопасность, основные функциональные характеристики или меры по управлению риском	**5.1.6 Планирование верификации ПО** Верификация требуется в отношении любой деятельности
План верификации должен формироваться для указания способов проверки этих функций и включать: – указания о том, на каком этапе (этапах) каждая функция должна проходить верификацию; – выбор и документирование принципов, мероприятий, методов и соответствующего уровня независимости персонала, выполняющего верификацию; – выбор и использование методов верификации; критерии верификации	**5.1.6 Планирование верификации ПО** Требование в отношении независимости персонала не включено в настоящий стандарт. Требование считается установленным в ISO 13485
Верификация должна выполняться в соответствии с планом верификации. Результаты верификации деятельности должны документироваться	Требования по проведению верификации установлено к большинству видов деятельности

Требования к ПЭМС в IEC 60601-1:2005 + IEC 60601-1:2005/AMD1:2012	Требования IEC 62304, связанные с ПО подсистемы ПЭМС
14.11 Валидация ПЭМС План валидации ПЭМС должен включать валидацию основной безопасности и основных функциональных характеристик	Настоящий стандарт не распространяется на валидацию ПО. Валидация ПЭМС является деятельностью на уровне системы и находится вне области применения настоящего стандарта
Метод проведения валидации ПЭМС должен быть задокументирован	Настоящий стандарт не распространяется на валидацию ПО. Валидация ПЭМС является деятельностью на уровне системы и находится вне области применения настоящего стандарта
Валидация ПЭМС должна выполняться в соответствии с планом валидации ПЭМС. Результаты деятельности по валидации ПЭМС должны быть задокументированы	Настоящий стандарт не распространяется на валидацию ПО. Валидация ПЭМС является деятельностью на уровне системы и находится вне области применения настоящего стандарта
Лицо, несущее основную ответственность за вариацию ПЭМС, должно быть независимым от коллектива разработчиков ПЭМС. Изготовитель должен задокументировать обоснование уровня его независимости	Настоящий стандарт не распространяется на валидацию ПО. Валидация ПЭМС является деятельностью на уровне системы и находится вне области применения настоящего стандарта
Никакой член коллектива разработчиков ПЭМС не должен нести ответственность за процесс валидации ПЭМС их собственного проекта	Настоящий стандарт не распространяется на валидацию ПО. Валидация ПЭМС является деятельностью на уровне системы и находится вне области применения настоящего стандарта
Все профессиональные взаимодействия между членами коллектива, выполняющего работы по валидации ПЭМС, и членами коллектива разработчиков ПЭМС должны регистрироваться в файле менеджмента риска	Настоящий стандарт не распространяется на валидацию ПО. Валидация ПЭМС является деятельностью на уровне системы и находится вне области применения настоящего стандарта
Ссылка на методы и результаты проверки соответствия (валидации) ПЭМС должна включаться в файл менеджмента риска	Настоящий стандарт не распространяется на валидацию ПО. 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 Техподдержка ПО Настоящий стандарт устанавливает, что техподдержка ПО должна быть запланирована, а реализация модификаций должна использовать процесс разработки ПО или установленный процесс техподдержки ПО
Когда ПО модифицируется, требования подраздела 4.3, § 5, § 7, § 8 и § 9 стандарта IEC 62304:2006 также применяются к модификации	
14.13 ПЭМС, предназначенные для подключения к ИТ-сетевым ресурсам	Требования в отношении подключения к ИТ-сетевым ресурсам не включены в настоящий стандарт
Если ПЭМС предназначена для соединения с помощью ИТ-сетевых ресурсов с другим изделием, которое не может контролироваться изготовителем ПЭМС, то в техническом описании указывают: a) цель подключения ПЭМС к ИТ-сетевым ресурсам; b) требуемые характеристики ИТ-сетевых ресурсов, включающей ПЭМС; c) требуемая конфигурация ИТ-сетевых ресурсов, включающей ПЭМС; d) технические характеристики сетевого подключения ПЭМС, включая спецификации безопасности; e) предполагаемый поток информации между ПЭМС, ИТ-сетевыми ресурсами и	

Требования к ПЭМС в IEC 60601-1:2005 + IEC 60601-1:2005/AMD1:2012	Требования IEC 62304, связанные с ПО подсистемы ПЭМС
другими устройствами, а также предполагаемая маршрутизация.	
Примеч. 1 – Это может включать аспекты результативности и безопасности данных и систем, связанные с основной безопасностью и основными функциональными характеристиками (см. также § H. 6 и IEC 80001-1:2010); f) перечень опасных ситуаций, возникающих в результате неспособности ИТ-сетевых ресурсов обеспечить характеристики, необходимые для выполнения цели подключения PEMS к ИТ-сетевым ресурсам; В сопроводительной документации изготовитель должен проинструктировать ответственную организацию о том, что: – соединение ПЭМС с ИТ-сетевыми ресурсами, которое производится с использованием другого оборудования, может приводить к ранее непредусмотренным рискам для пациентов, операторов или третьих лиц; – ответственная организация должна идентифицировать, анализировать, оценивать эти риски и – управлять ими. *Примеч. 3[25] – IEC 80001-1:2010 содержит рекомендации для ответственной организации по обращению с такими рисками;* – последующие изменения ИТ-сетевых ресурсов могут приводить к появлению новых рисков и требовать дополнительного анализа; – последующие изменения ИТ-сетевых ресурсов могут включать: • изменения в их конфигурации; • подсоединение к ним дополнительных элементов; • отсоединение от них отдельных элементов; • модификацию соединённого с ними оборудования; • модернизацию соединённого с ними оборудования	

C.4.7 Взаимосвязь с IEC 60601-1-4

IEC 60601-1-4 был отменен.

C.5 – Взаимосвязь с IEC 61010-1

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

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

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

IEC 61010-1:2010 содержит общее требование к оценке рисков в § 17, которое является более упрощённым, чем полные требования ISO 14971 по менеджменту риска. Применение стандарта IEC 61010-1, § 17 само по себе не соответствует требуемым критериям менеджмента риска IEC 62304, который основан на полных требованиях ISO 14971 по менеджменту риска. Исходя из этого ожидается, что если *in vitro* МИ имеет связанные с ПО риски, то процесс менеджмента риска выполняется в соответствии с ISO 14971, а не

[25] Нумерация примечаний в англоязычном оригинале тут также перепрыгнула ☹

только § 17 IEC 61010-1. Соответствие § 17 стандарта IEC 61010-1 будет достигнуто, как подробно описано в Примечании к § 17 стандарта IEC 61010-1.

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

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

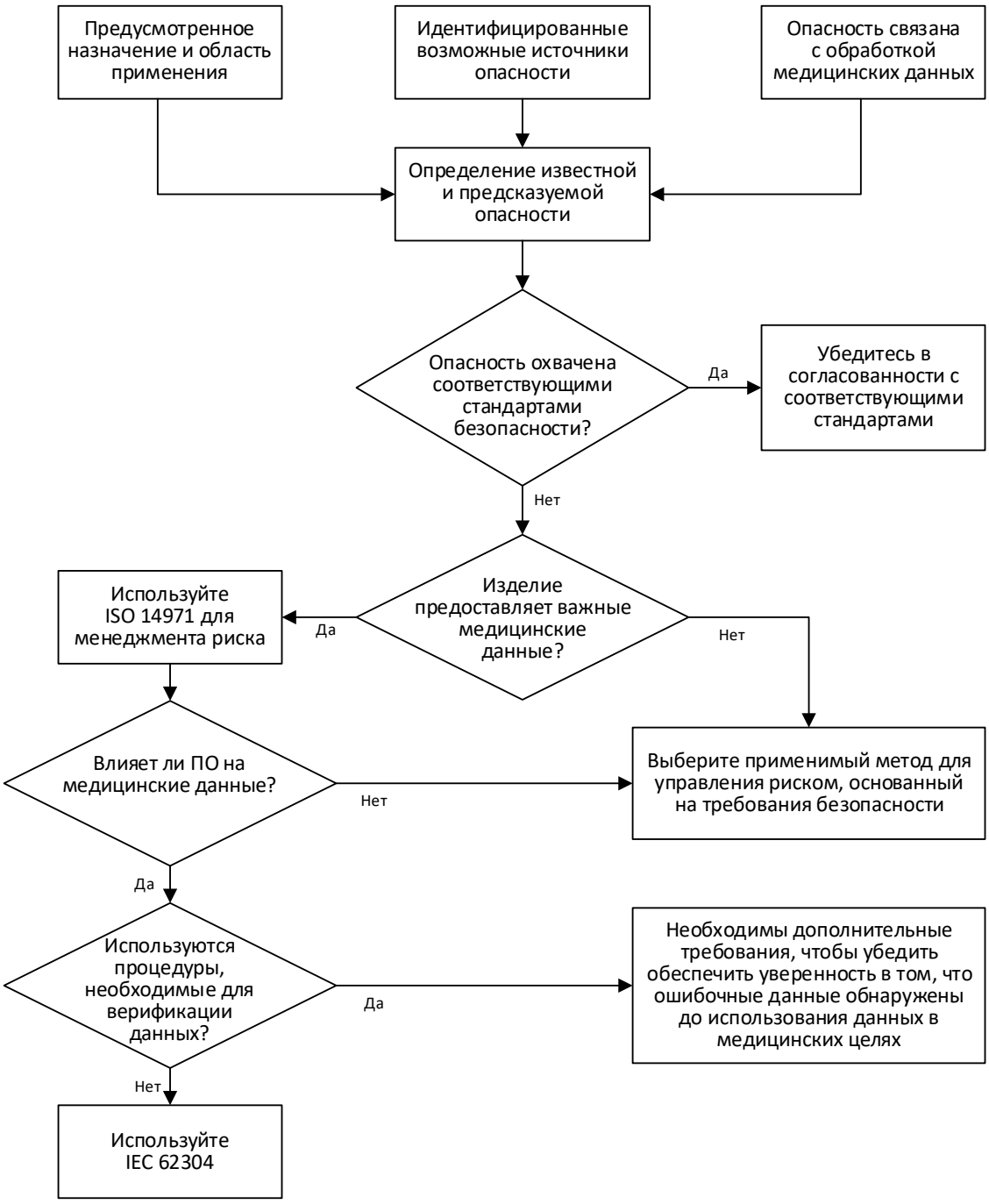


Рисунок 11(С.3) – Применение IEC 62304 с IEC 61010-1

С.6 Взаимосвязь с ISO/IEC 12207

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

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

- исключает аспекты системы, такие как системные требования, архитектура системы и валидация;
- пропускает некоторые процессы, рассматриваемые как дублирующая деятельность, представленные в различных изданиях для МИ;
- добавляет процесс менеджмента риска (безопасность) и процесс выпуска ПО;

- включает документирование и верификацию поддерживающих процессов в процессы разработки и техподдержки;
- объединяет реализацию процессов и планирование деятельности по каждому процессу в единую деятельность по процессам разработки и ТО;
- классифицирует требования с учётом безопасности;
- не классифицирует процессы как первостепенные или поддерживающие и не группирует процессы, как это сделано в ISO/IEC 12207.

Большинство отличий были реализованы исходя из нужд и потребностей промышленности МИ:

- фокусироваться на аспектах безопасности и менеджмента риска МИ, установленных в ISO 14971;
- выбрать подходящие процессы, полезные в регулируемой внешней среде;
- принять во внимание, что разработка ПО включена в систему качества (которая охватывает некоторые процессы и требования ISO/IEC 12207);
- уменьшить уровень обобщения, чтобы облегчить применение.

Настоящий стандарт не противоречит ISO/IEC 12207. ISO/IEC 12207 может быть полезным в качестве вспомогательной информации для создания правильно структурированной модели жизненного цикла разработки ПО, которая включает требования настоящего стандарта.

Таблица С.5[26], которая была подготовлена подкомитетом 7 ISO/IEC, показывает взаимосвязь между настоящим стандартом и ISO/IEC 12207.

Таблица 6(С.5) – Взаимосвязь с процессами ISO/IEC 12207:2008

Процессы настоящего стандарта		**Процессы ISO/IEC 12207:2008**	
Деятельность	**Задача**	**Процессы**	**Деятельность/Задача**
5 процесс разработки ПО			
5.1 Планирование разработки ПО	5.1.План разработки ПО	7.1.1 Реализация ПО	7.1.1.3.1 Стратегия реализации ПО 6.3.1.3.2 Планирование проекта
	5.1.2 Поддержание плана разработки ПО в актуальном состоянии	6.3.2 Оценка проекта и процесс управления	6.3.2.3.2 Управление проектом 6.3.2.3.2.1
	5.1.3 План разработки ПО относительно проектирования и разработки системы	6.4.3 Процесс проектирования Архитектуры системы 6.4.5 Системная интеграция 7.2.5 Процесс валидации ПО	6.4.3.3.1 Установление Архитектуры 6.4.5.3.1 Интеграция 7.2.5.3.1 Процесс реализации
5.1 Планирование разработки ПО	5.1.4 Стандарты, методы и инструменты планирования разработки ПО	7.1.1 Реализация ПО	7.1.1.3.1 Стратегия реализации ПО
	5.1.5 Программная интеграция и планирование тестирования интеграции	7.1.6 Интеграция ПО	7.1.6.3.1 Интеграция ПО
	5.1.6 Планирование верификации ПО	7.2.4 Верификация ПО 7.1.5 Процесс конструирования программных средств 7.1.6 Software Integration 7.1.7 Квалификационное тестирование ПО	7.2.4.3.1 Процесс реализации 7.1.5.3.1 Процесс конструирования программных средств 7.1.6.3.1 Интеграция ПО 7.1.6.3.1.5 7.1.7.3.1 Квалификационное тестирование ПО
	5.1.7 Планирование менеджмента риска ПО	6.3.4 Процесс менеджмента риска	
	5.1.8 Документация по планированию	7.2.1 Менеджмент документации ПО	7.2.1.3.1 Процесс реализации
	5.1.9 Планирование менеджмента конфигурации ПО	7.2.2 Менеджмент конфигурации ПО	7.2.2.3.1 Процесс реализации 7.2.8.3.1 Процесс реализации

[26] Удалил из колонки «Процессы ISO/IEC 12207:2008 / Деятельность/Задача» пустые пункты без подписей

Процессы настоящего стандарта		Процессы ISO/IEC 12207:2008	
Деятельность	**Задача**	**Процессы**	**Деятельность/Задача**
		7.2.8 Процесс решения проблем в ПО	
	5.1.10 Поддержка элементов, подлежащих управлению	6.2.2 Менеджмент инфраструктуры	6.2.2.3.2 Установление инфраструктуры 6.2.2.3.3 Поддержание инфра структуры
	5.1.11 Управление составными частями конфигурации ПО до верификации	7.2.2 Менеджмент конфигурации ПО	7.2.2.3.2 Идентификация конфигурации
5.2 Анализ требований к ПО	5.2.1 Отделение и документирование требований к ПО на основе требований системы	6.4.3 Проектирование Архитектуры системы	6.4.3.3.1 Установление Архитектуры
5.2 Анализ требований к ПО	5.2.2 Содержание требований к ПО	7.1.2 Анализ требований к ПО	7.1.2.3.1 Анализ требований к ПО
	5.2.3 Включение мер по управлению риском в требования к ПО		
	5.2.4 Переоценивание анализа риска МИ	Нет	Нет
	5.2.5 Обновление требований к системе	7.1.2 Анализ требований к ПО	7.1.2.3.1 Анализ требований к ПО 7.1.2.3.1.1 a) и b)
	5.2.6 Верификация требований к ПО	7.2.4 Верификация ПО	7.2.4.3.2 Верификация
5.3 Проектирование архитектуры ПО	5.3.1 Преобразование требований к ПО в архитектуру	7.1.3 Проектирование Архитектуры ПО	7.1.3.3.1 Проектирование Архитектуры ПО
	5.3.2 Разработка архитектуры для интерфейсов ПСЧ		7.1.3.3.1 Проектирование Архитектуры ПО
	5.3.3 Определение требований к функциональным и эксплуатационным характеристикам элементов ПОНП	Нет	Нет
	5.3.4 Определение требований к аппаратным и программным средствам системы, требуемых элементами ПОНП	Нет	Нет
	5.3.5 Идентификация обособленности, необходимой для управления риском	Нет	Нет
	5.3.6 Верификация Архитектуры ПО	7.1.3 Проектирование Архитектуры ПО	7.1.3.3.1 Проектирование Архитектуры ПО 7.1.3.3.1.6
5.4 Разработка детального дизайна ПО	5.4.1 Дробление ПО на программные блоки	7.1.4 Детализированная разработка ПО	7.1.4.3.1 Детализированная разработка ПО
	5.4.2 Разработка детального дизайна для каждого программного блока		
	5.4.3 Разработка детального дизайна для интерфейсов		7.1.4.3.1 Детализированная разработка ПО
	5.4.4 Верификация детального дизайна	7.1.4 Детализированная разработка ПО	7.1.4.3.1 Детализированная разработка ПО
5.5 Имплемен-	5.5.1 Имплементация каждого программного блока	7.1.5 Конструирование ПО	7.1.5.3.1 Конструирование ПО

Процессы настоящего стандарта		Процессы ISO/IEC 12207:2008	
Деятельность	**Задача**	**Процессы**	**Деятельность/Задача**
тация программных блоков	5.5.2 Установление процесса верификации программного блока	7.1.4 Детализированная разработка ПО 7.1.5 Конструирование ПО	7.1.4.3.1 Детализированная разработка ПО 7.1.5.3.1 Конструирование
	5.5.3 Критерии приёмки программных блоков	7.1.5 Конструирование ПО	7.1.5.3.1 Конструирование ПО
	5.5.4 Дополнительные критерии приёмки программных блоков	7.1.5 Конструирование ПО 7.2.4 Верификация ПО	7.1.5.3.1 Конструирование ПО
	5.5.5. Верификация программных блоков	7.1.5 Конструирование ПО	7.1.5.3.1 Конструирование ПО
5.6 Интеграция ПО и тестирование интеграции	5.6.1 Интеграция программных блоков	7.1.6 Интеграция ПО	7.1.6.3.1 Интеграция ПО
	5.6.2 Верификация интеграции ПО	7.1.6 Интеграция ПО 6.4.5 Системная интеграция	7.1.6.3.1 Интеграция ПО 6.4.5.3.1 Интеграция
	5.6.3 Интеграционное тестирование ПО	7.1.7 Квалификационное тестирование ПО	7.1.7.3.1 Квалификационное тестирование ПО
5.6 Интеграция ПО и тестирование интеграции	5.6.4 Содержание тестирования интеграции ПО	7.1.7 Квалификационное тестирование ПО	7.1.7.3.1 Квалификационное тестирование ПО
	5.6.5 Оценивание процедур тестирования интеграции ПО	Нет	Нет
	5.6.6 Проведение регрессионного тестирования	7.1.6 Интеграция ПО	7.1.6.3.1 Интеграция ПО
	5.6.7 Содержание записей в отношении регрессионного тестирования	7.1.6 Интеграция ПО	7.1.6.3.1 Интеграция ПО
	5.6.8 Использование процесса решения проблем с ПО	7.2.4 Верификация ПО	7.2.4.3.1 Реализация процесса 7.2.4.3.1.6
5.7 Тестирование ПС	5.7.1 Установление тестирования в отношении требований к ПО	7.1.6 Интеграция ПО 7.1.7 Квалификационное тестирование ПО	7.1.6.3.1 Интеграция ПО7.1.7.3.1 Квалификационное тестирование ПО
	5.7.2 Применение процесса решения проблем с ПО	7.2.4 Верификация ПО	7.2.4.3.1 Реализация процесса 7.2.4.3.1.6
	5.7.3 Повторное тестирование после внесения изменений	7.2.8 Процесс решения проблем в ПО	7.2.4.3.1 Реализация процесса 7.2.4.3.1.1
	5.7.4 Оценивание тестирования ПС	7.1.7 Квалификационное тестирование ПО	7.1.7.3.1 Квалификационное тестирование ПО
	5.7.5 Содержание отчёта по тестированию ПС	7.1.7 Квалификационное тестирование ПО	7.1.7.3.1 Квалификационное тестирование ПО
5.8 Выпуск ПО на системном уровне	5.8.1 Обеспечение завершённости верификации ПО	6.4.9 Функционирование ПО 7.2.2 Менеджмент конфигурации ПО	6.4.9.3.2 Активация и проверка функционирования 7.2.2.3.6 Поставка и менеджмент выпуска
5.8 Выпуск ПО на системном уровне	5.8.2 Документирование известных остаточных аномалий		
	5.8.3 Оценивание известных остаточных аномалий		
	5.8.4 Документирование выпущенных версий	7.2.2 Процесс ме-	7.2.2.3.6 Поставка и менеджмент выпуска

Процессы настоящего стандарта		Процессы ISO/IEC 12207:2008	
Деятельность	**Задача**	**Процессы**	**Деятельность/Задача**
	5.8.5 Документирование создания выпущенного ПО	неджмента конфигурации ПО	
	5.8.6 Обеспечение полного завершения ДиЗ		
	5.8.7 Архивирование ПО		
	5.8.8 Обеспечение надёжной поставки выпущенного ПО		
6 Техподдержка ПО		**6.4.10 Процесс техподдержки ПО**	
6.1 Установление плана техподдержки ПО		6.4.10 Техподдержка ПО	Нет
6.2 Анализ модификации и проблем	6.2.1 Документирование и оценивание обратной связи	Нет	Нет
	6.2.1.1 Мониторинг обратной связи	6.4.10 Техподдержка ПО	Нет
	6.2.1.2 Документирование и оценивание обратной связи	6.4.10 Техподдержка ПО	Нет
	6.2.1.3 Оценивание влияния отчётов о проблемах на безопасность	6.4.10 Техподдержка ПО	Нет
	6.2.2 Использование процесса решения проблем ПО	6.4.10 Техподдержка ПО	Нет
6.2 Анализ модификации и проблем	6.2.3 Анализ запросов на изменение	6.4.10 Техподдержка ПО	Нет
	6.2.4 Одобрение запроса на изменение	6.4.10 Техподдержка ПО	Нет
	6.2.5 Информирование пользователей и регулирующих органов	6.4.10 Техподдержка ПО	Нет
6.3 Осуществление модификации		Нет	Нет
	6.3.1 Использование установленного процесса осуществления модификации	6.4.10 Техподдержка ПО	Нет
	6.3.2 Повторный выпуск модифицированной ПС	7.2.2 Процесс менеджмента конфигурации ПО	Нет
7 Процесс менеджмента риска ПО	6.3.4 Процесс менеджмента риска Основан на ISO/IEC 16085. Несмотря на некоторую общность, в нем не рассмотрены конкретные требования к разработке ПОМИ в отношении менеджмента риска		
8 Процесс менеджмента конфигурации ПО			
8.1 Идентификация конфигурации	8.1.1 Установление средств идентификации составных частей конфигурации	7.2.2 Процесс менеджмента конфигурации ПО	Нет
	8.1.2 Идентификация ПОНП	Нет	Нет
	8.1.3 Идентификация документации конфигурации системы	7.2.2 Процесс менеджмента конфигурации ПО	Нет
8.2 Управление изменениями	8.2.1 Одобрение запросов на изменения	7.2.2 Процесс менеджмента конфигурации ПО	Нет
	8.2.2 Осуществление изменений	6.4.10 Техподдержка ПО	Нет
	8.2.3 Верификация изменений	7.2.2 Процесс менеджмента конфигурации ПО	Нет
	8.2.4 Обеспечение средствами для прослеживаемой изменений		
8.3 Учёт статуса конфигурации		7.2.2 Процесс менеджмента конфигурации ПО	Нет
9 Процесс решения проблем ПО			
9.1 Подготовка отчётов о проблемах		7.2.8 Процесс решения проблем в ПО	Нет
9.2 Исследование проблемы		7.2.8 Процесс решения проблем в ПО	Нет
9.3 Консультирование заинтересованных сторон		7.2.8 Процесс решения проблем в ПО	Нет

Процессы настоящего стандарта		Процессы ISO/IEC 12207:2008	
Деятельность	**Задача**	**Процессы**	**Деятельность/Задача**
9.4 Использование процесса управления изменениями		7.2.2 Процесс менеджмента конфигурации ПО 6.4.10 Техподдержка ПО	Нет
9.5 Поддержание записей		7.2.8 Процесс решения проблем в ПО	Нет
9.6 Анализ проблем на предмет выявления тенденций		7.2.8 Процесс решения проблем в ПО	Нет
9.7 Верификация решения проблем ПО		7.2.8 Процесс решения проблем в ПО	Нет
9.8 Содержание документации по тестированию		ISO 12207 требует документирования всех задач проведения тестирования	Нет

C.7 Взаимосвязь с IEC 61508

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

IEC 61508 сфокусирован на трех главных вопросах:

1) жизненном цикле менеджмента риска и процессах жизненного цикла;

2) определении уровней безопасности эксплуатации оборудования;

3) рекомендуемых техниках, инструментах и методах для разработки ПО и уровнях независимости персонала, ответственного за выполнение различных задач.

Вопрос 1) включён в настоящий стандарт нормативной ссылкой на ISO 14971 (стандарт по менеджменту риска для промышленности МИ). Влияние этой ссылки состоит в том, чтобы адаптировать подход ISO 14971 к менеджменту риска как составную часть процесса ПО для ПОМИ.

Для вопроса 2) настоящий стандарт принимает более простой подход, чем IEC 61508, классифицирующий ПО на четыре «уровня безопасности эксплуатации оборудования», определённые с точки зрения надёжности. Цели надёжности идентифицируют после анализа риска, который определяет как тяжесть, так и вероятность причинения вреда, вызванного отказом ПО.

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

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

Приложение D (справочное). Применение

D.1 Введение

В данном приложении приводится общий обзор того, как настоящий стандарт может быть реализован в процессах изготовителя. Предполагается, что другие стандарты, такие как ИСО 13485[I], требуют подходящих и сопоставимых процессов.

D.2 Система менеджмента качества (СМК)

В контексте настоящего стандарта для изготовителей МИ, включая ПОМИ, установление СМК (далее – СМК) требуется в соответствии с 4.1. Настоящий стандарт не требует, чтобы СМК обязательно была сертифицирована.

D.3 Оценивание процессов менеджмента качества

Рекомендуется оценивать, как установленные и документированные процессы СМК охватывают процессы жизненного цикла ПО посредством проведения аудитов, проверок инспекций или анализа, за которые

несёт ответственность изготовитель. Любые выявленные несоответствия могут быть устранены посредством расширения установленных процессов менеджмента качества или оформлены отдельными документами. Если изготовитель уже имеет документированные процессы, которые регламентируют разработку, верификацию и валидацию ПО, то данные процессы также должны быть оценён с целью определения согласованности с настоящим стандартом.

D.4 Интеграция требований настоящего стандарта в процессы менеджмента качества изготовителя

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

Изготовитель несёт ответственность за обеспечение надлежащего выполнения процессов, описанных в настоящем стандарте, когда ПОМИ разработано изготовителями оригинального оборудования (OEM) или субподрядчиками, не имеющими собственной документированной СМК.

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

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

При наличии несоответствий необходимо предпринять действия для улучшения процесса.

Список также может быть использован для оценивания процессов после выполнения действия.

Таблица 7(D.1) – Контрольный список для малых предприятий, не имеющих сертифицированной СМК

Деятельность	**Соответствующий § ИСО 13485:2016**	Охватывается существующими процедурами?	Если да: ссылка	Предпринятые действия
5.1 Планирование разработки ПО	7.3.1 Планирование проектирования и разработки	Да/Нет		
5.2 Анализ требований к ПО	7.3.2 Входные данные для проектирования и разработки	Да/Нет		
5.3 Проектирование архитектуры ПО		Да/Нет		
5.4 Разработка детального дизайна ПО		Да/Нет		
5.5 Имплементация программных блоков		Да/Нет		
5.6 Интеграция ПО и тестирование интеграции		Да/Нет		
5.7 Тестирование ПС	7.3.3 Выходные данные проектирования и разработки 7.3.4 Анализ проекта и разработки	Да/Нет		
5.8 Выпуск ПО на системном уровне	7.3.5 Верификация проектирования и разработки 7.3.6 Валидация проектирования и разработки	Да/Нет		
6.1 Установление плана техподдержки ПО	7.3.7 Управление изменениями проектирования и разработки	Да/Нет		
6.2 Анализ модификаций и проблем		Да/Нет		
6.3 Осуществление модификации	7.3.5 Верификация проектирования и разработки 7.3.6 Валидация проектирования и разработки	Да/Нет		
7.1 Анализ ПО, способствующего		Да/Нет		

Деятельность	Соответствующий § ИСО 13485:2016	Охватывается существующими процедурами?	Если да: ссылка	Предпринятые действия
опасным ситуациям				
7.2 Меры по управлению риском		Да/Нет		
7.3 Верификация мер по управлению риском		Да/Нет		
7.4 менеджмент риска в отношении изменений ПО		Да/Нет		
8.1 Определение конфигурации	7.5.3 Идентификация и прослеживаемость	Да/Нет		
8.2 Управление изменениями	7.5.3 Идентификация и прослеживаемость	Да/Нет		
8.3 Учёт статуса конфигурации		Да/Нет		
9 процесс решения проблем ПО		Да/Нет		

Библиография[27]

[7] ISO 9001:2008 Quality management systems – Requirements (СМК. Требования)

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

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

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

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

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

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

[B] [22] IEC 82304-1-2016 Healthcare Software Systems – Part 1: General requirements (ГОСТ Р МЭК 82304-1-2019 «Медицинское программное обеспечение. Часть 1. Общие требования к безопасности программных продуктов»).

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

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

[E] [6] ISO 9000:2005 Quality management systems – Fundamentals and vocabulary (СМК. Основные положения и словарь)

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

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

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

[I] [8] ISO 13485:2016 Medical devices – Quality management systems – Requirements for regulatory purposes (Изделия медицинские. СМК. Требования для целей регулирования).

[J] [15] ISO/IEC 90003:2014 Software engineering – Guidelines for the application of ISO 9001:2008 to computer software

[27] Часть библиографии идёт как она шла в ГОСТе (и в оригинале), а часть, более мелким шрифтом, выполнена с использованием механизма WOR - концевых сносок. Порядок там получился другой из-за того что данный список формировался по ходу упоминания документов в тексте стандарта. Но номера из оригинала мы также сохранили 😊

(Разработка программных продуктов. Руководящие указания по применению ИСО 9001:2008 при разработке программных продуктов).

[K] [21] IEC 62366-1:2015 Medical devices – Part 1: Application of usability engineering to medical devices (Изделия медицинские. Часть 1. Проектирование МИ с учётом эксплуатационной пригодности).

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

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

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

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

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