ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 55544— 2013/IEC/TR 80002-1:2009

Программное обеспечение медицинских изделий

Часть 1

Руководство по применению ИСО 14971 к программному обеспечению медицинских изделий

IEC/TR 80002-1:2009
Medical devices software — Part 1: Guidance on the application of ISO 14971 medical devices software
(IDT)

Издание официальное

Москва
Стандартинформ
2014

Предисловие

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

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

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 28 августа 2013 г. № 620-ст

4 Настоящий стандарт идентичен международному документу МЭК/ТО 80002-1:2009 «Программное обеспечение медицинских изделий. Часть 1. Руководство по применению ИСО 14971 к программному обеспечению медицинских изделий» (IEC/TR 80002-1:2009 «Medical devices software — Part 1: Guidance on the application of ISO 14971 medical devices software»).

При применении настоящего стандарта рекомендуется использовать вместо ссылочных международных стандартов соответствующие им национальный стандарт Российской Федерации и межгосударственный стандарт, сведения о которых приведены в дополнительном приложении ДА

5 ВВЕДЕН ВПЕРВЫЕ

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

Содержание

Введение

Программное обеспечение часто является неотъемлемой частью МЕДИЦИНСКИХ ИЗДЕЛИЙ. Для установления БЕЗОПАСНОСТИ и результативности МЕДИЦИНСКИХ ИЗДЕЛИЙ, содержащих программное обеспечение, требуется должное понимание назначения программного обеспечения, а также демонстрация того, что исполнение программного обеспечения отвечает этому назначению, не вызывая никаких недопустимых РИСКОВ.

Важно понимать, что само по себе программное обеспечение не является ОПАСНОСТЬЮ, но может способствовать ОПАСНЫМ СИТУАЦИЯМ. Программное обеспечение всегда должно рассматриваться в перспективе СИСТЕМЫ, а МЕНЕДЖМЕНТ РИСКА программного обеспечения также не может осуществляться в отрыве от этой СИСТЕМЫ.

Сложные конструкции программного обеспечения могут допускать сложные последовательности событий, которые могут способствовать ОПАСНЫМ СИТУАЦИЯМ. Значительная часть ЗАДАЧИ по МЕНЕДЖМЕНТУ РИСКА программного обеспечения состоит в определении тех последовательностей событий, которые могут приводить к ОПАСНЫМ СИТУАЦИЯМ, а также в определении точек, в которых эта последовательность может быть прервана с целью предотвращения ВРЕДА или уменьшения вероятности его возникновения.

В программном обеспечении, последовательности событий, которые способствуют ОПАСНЫМ СИТУАЦИЯМ, могут быть разделены на две категории:

а) последовательности событий, приводящие к непредвиденным программным ответам на входные данные (ошибки в спецификации программного обеспечения);

б) последовательности событий, возникающие из-за неправильного кодирования (ошибки в реализации программного обеспечения).

Эти категории характерны для программного обеспечения, так как возникают из-за трудности правильного определения и реализации сложной СИСТЕМЫ и полной ВЕРИФИКАЦИИ сложной СИСТЕМЫ.

Поскольку очень сложно оценивать вероятность АНОМАЛИЙ в программном обеспечении, которые могут способствовать ОПАСНЫМ СИТУАЦИЯМ, и поскольку программное обеспечение не отказывает в ПРОЦЕССЕ использования случайным образом, например, из-за износа, то при проведении АНАЛИЗА РИСКА основное внимание должно уделяться идентификации потенциальной функциональности программного обеспечения, а также должно уделяться внимание АНОМАЛИЯМ, которые могут приводить к ОПАСНЫМ СИТУАЦИЯМ, а не оценке возникновения вероятности. РИСКИ, возникающие из-за АНОМАЛИЙ программного обеспечения, чаще всего нужно оценивать только по ТЯЖЕСТИ ВРЕДА.

МЕНЕДЖМЕНТ РИСКА — это всегда сложная задача, которая становится еще более сложной, если это касается программного обеспечения. Нижеследующие пункты содержат дополнительную информацию относительно особенностей программного обеспечения и служат основой для понимания ИСО 14971:2007 в отношении программного обеспечения.

Структура стандарта

Настоящий стандарт построен таким образом, чтобы следовать структуре ИСО 14971:2007 и служить руководством для каждого вида деятельности по МЕНЕДЖМЕНТУ РИСКА в отношении программного обеспечения.

Существует некоторая преднамеренная избыточность в информации, которая связана с итеративным характером деятельности по МЕНЕДЖМЕНТУ РИСКА применяемого к ЖИЗНЕННОМУ ЦИКЛУ программного обеспечения.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Программное обеспечение медицинских изделий

Часть 1

Руководство по применению ИСО 14971 к программному обеспечению медицинских изделий

Medical devices software. Part 1. Guidance on the application of ISO 14971 medical devices software

Дата введения — 2014—08—01

1 Общие положения

1.1 Область применения

Настоящий стандарт предоставляет собой руководство по применению требований, содержащихся в ИСО 14971 «*Медицинские изделия. Применение менеджмента риска к медицинским изделиям*» *(Medical devices — Application of risk management to medical devices)* к программному обеспечению медицинских изделий, со ссылкой на МЭК 62304 «*Программное обеспечение медицинских изделий. Процессы жизненного цикла программного обеспечения*» *(Medical device software — Software life cycle processes)*. Настоящий стандарт ничего не добавляет и не изменяет в ИСО 14971 или МЭК 62304.

Настоящий стандарт предназначен для исполнителей МЕНЕДЖМЕНТА РИСКА, которым необходимо осуществить МЕНЕДЖМЕНТ РИСКА, когда программное обеспечение включено в МЕДИЦИНСКОЕ ИЗДЕЛИЕ или СИСТЕМУ, а также для инженеров по программному обеспечению, которые должны понимать, как выполнить требования по МЕНЕДЖМЕНТУ РИСКА, содержащиеся в ИСО 14971.

ИСО 14971, повсеместно признанный регулирующими органами, широко применяется в качестве основного стандарта, используемого при осуществлении МЕНЕДЖМЕНТА РИСКА МЕДИЦИНСКИХ ИЗДЕЛИЙ. МЭК 62304 дает нормативную ссылку на ИСО 14971, требуя его применения. Содержание этих двух стандартов является основой для настоящего стандарта.

Следует отметить, что, хотя требования ИСО 14971 и настоящего стандарта применяются к МЕДИЦИНСКИМ ИЗДЕЛИЯМ, он может использоваться для осуществления ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА в отношении БЕЗОПАСНОСТИ для всего программного обеспечения в области здравоохранения независимо от того, классифицировано ли оно как МЕДИЦИНСКОЕ ИЗДЕЛИЕ или нет.

Настоящий стандарт не включает:

- области, охваченные существующими или планируемыми стандартами, например, тревожную сигнализацию, проектирование эксплуатационной пригодности, работу в сети и т. д.,
- аспекты производства или системы менеджмента качества программного обеспечения; или
- инструменты разработки программного обеспечения.

Настоящий стандарт не предназначен для использования в качестве основы для проверки регулирующих требований или деятельности по сертификации.

Для целей настоящего стандарта термин «следует» (should) используется, чтобы показать, что среди нескольких возможностей удовлетворения требования одна рекомендуется как особенно пригодная, без упоминания или исключения других, или чтобы показать, что определенный курс действий предпочтительнее, но необязателен. Этот термин не должен интерпретироваться как требование.

1.2 Нормативные ссылки

В настоящем стандарте использованы ссылки на следующие стандарты:

МЭК 62304:2006 «*Программное обеспечение медицинских изделий — процессы жизненного цикла программного обеспечения*» (IEC 62304:2006, Medical device software — Software life cycle processes)

Издание официальное

ИСО 14971:2007 «*Медицинские изделия — Применение менеджмента риска к медицинским изделиям*» (ISO 14971:2007, Medical devices — Application of risk management to medical devices)

В случае датированных ссылок применяется только приведенное издание. Для недатированных ссылок применяется последнее издание документа (включая все поправки), на который приведена ссылка.

2 Термины и определения

В настоящем стандарте применены термины и определения, данные в ИСО 14971, МЭК 62304, а также нижеследующие термины и определения.

Примечание — Список определенных терминов можно найти на странице 58.

2.1 **РАЗНООБРАЗИЕ** (DIVERSITY): Форма избыточности, при которой избыточные элементы используют разные (различающиеся) компоненты, технологии или методы с целью снизить вероятность того, что все элементы откажут одновременно по общей причине.

2.2 **ИЗБЫТОЧНОСТЬ** (REDUNDANCY): Обеспечение многочисленных компонентов или механизмов для выполнения одной и той же функции таким образом, чтобы отказ одного или более компонентов или механизмов не препятствовал выполнению функции.

2.3 **СВЯЗАННОЕ С БЕЗОПАСНОСТЬЮ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ** (SAFETY-RELATED SOFTWARE): Программное обеспечение, которое может вызывать ОПАСНЫЕ СИТУАЦИИ, или программное обеспечение, используемое для реализации мер по УПРАВЛЕНИЮ РИСКОМ.

3 Общие требования к МЕНЕДЖМЕНТУ РИСКА

3.1 ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА

3.1.1 Общие положения

Текст ИСО 14971

3 Общие требования к МЕНЕДЖМЕНТУ РИСКА

3.1 ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА

ИЗГОТОВИТЕЛЬ должен устанавливать, документировать и поддерживать в рабочем состоянии непрерывный ПРОЦЕСС идентификации ОПАСНОСТЕЙ, связанных с МЕДИЦИНСКИМ ИЗДЕЛИЕМ, определения и оценивания сопутствующих РИСКОВ, управления данными РИСКАМИ и мониторинга результативности такого управления на протяжении всего ЖИЗНЕННОГО ЦИКЛА МЕДИЦИНСКОГО ИЗДЕЛИЯ. Этот ПРОЦЕСС должен включать следующие элементы:

- АНАЛИЗ РИСКА;
- ОЦЕНИВАНИЕ РИСКА;
- УПРАВЛЕНИЕ РИСКОМ;
- производственную и ПОСТПРОИЗВОДСТВЕННУЮ информацию.

Документированные ПРОЦЕССЫ жизненного цикла продукции (см. раздел 7 [3]) должны включать соответствующие элементы ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА.

Примечания

1 Документированный ПРОЦЕСС системы менеджмента качества может быть использован для обеспечения системного подхода к рассмотрению проблем БЕЗОПАСНОСТИ, позволяющего, в частности, идентифицировать на ранних стадиях ОПАСНОСТИ и ОПАСНЫЕ СИТУАЦИИ, связанные со сложными МЕДИЦИНСКИМИ ИЗДЕЛИЯМИ и системами.

2 ПРОЦЕСС МЕНЕДЖМЕНТА РИСКА схематически представлен на рисунке 1.

В зависимости от конкретной стадии ЖИЗНЕННОГО ЦИКЛА МЕДИЦИНСКОГО ИЗДЕЛИЯ отдельным элементам МЕНЕДЖМЕНТА РИСКА может быть уделено особое внимание. Деятельность по МЕНЕДЖМЕНТУ РИСКА может осуществляться *итеративно* или поэтапно в зависимости от рассматриваемого МЕДИЦИНСКОГО ИЗДЕЛИЯ. Приложение В содержит более подробный обзор этапов ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА.

Соответствие требованиям данного подраздела проверяют путем контроля необходимых документов.

Рисунок 1 — Схематичное представление ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА

БЕЗОПАСНОСТЬ является свойством СИСТЕМЫ (в данном случае всего МЕДИЦИНСКОГО ИЗДЕЛИЯ), которая включает программное обеспечение. МЕНЕДЖМЕНТ РИСКА должен осуществляться в рамках СИСТЕМЫ, включая программное обеспечение и всю ее аппаратную среду. Деятельность по МЕНЕДЖМЕНТУ РИСКА программного обеспечения не должна проводиться в отрыве от СИСТЕМЫ.

Поскольку аспекты МЕНЕДЖМЕНТА РИСКА программного обеспечения не могут быть эффективно выполнены в отрыве от всего МЕНЕДЖМЕНТА РИСКА МЕДИЦИНСКОГО ИЗДЕЛИЯ, существуют действия, являющиеся составной частью ЖИЗНЕННОГО ЦИКЛА программного обеспечения, которые наилучшим образом могут быть выполнены инженерами-программистами. Существуют также элементы программного обеспечения, которое требует бóльшего внимания и особого разъяснения, чем то, которое приве-

дено в ИСО 14971 для полного МЕНЕДЖМЕНТА РИСКА МЕДИЦИНСКОГО ИЗДЕЛИЯ. Важно подчеркнуть, что с целью обеспечения эффективности даже программные аспекты МЕНЕДЖМЕНТА РИСКА нуждаются в ориентации на РИСКИ, связанные с МЕДИЦИНСКИМ ИЗДЕЛИЕМ.

Примечание 1 — Аспекты МЕНЕДЖМЕНТА РИСКА программного обеспечения не могут быть эффективно выполнены в отрыве от всего МЕНЕДЖМЕНТА РИСКА МЕДИЦИНСКОГО ИЗДЕЛИЯ из-за взаимозависимости отказов аппаратных средств, отказов программного обеспечения и аппаратных и программных мер УПРАВЛЕНИЯ РИСКОМ.

Примечание 2 — Например, все систематические, а не случайные отказы программного обеспечения (как и многие типы аппаратных отказов или поломок) и их вероятность не могут быть точно оценены. Вот почему способ, которым компонент вероятности РИСКА применяется к программному обеспечению, существенно различается. См. 4.4.3.

Для инженеров-программистов существует много способов обеспечения общей БЕЗОПАСНОСТИ МЕДИЦИНСКИХ ИЗДЕЛИЙ на ранних стадиях проектирования. Роль программного обеспечения в БЕЗОПАСНОСТИ МЕДИЦИНСКИХ ИЗДЕЛИЙ должна быть рассмотрена перед тем, как проектирование СИСТЕМЫ будет завершено. Участвуя в ПРОЦЕССЕ проектирования МЕДИЦИНСКОГО ИЗДЕЛИЯ по мере развития проекта инженер-программист может способствовать принятию решений, СВЯЗАННЫХ С БЕЗОПАСНОСТЬЮ, относительно РИСКА, связанного с программным обеспечением. Такие решения могут включать, по крайней мере, следующие действия:

- выбор подходящих аппаратных средств для поддержки программного обеспечения;
- разделение функций между программными и аппаратными средствами;
- предусмотренное применение всего МЕДИЦИНСКОГО ИЗДЕЛИЯ, а также предусмотренное применение пользовательских интерфейсов программного обеспечения;
- исключение сложного программного обеспечения, не являющегося необходимым.

3.1.2 Итерация

Типичный цикл разработки ЖИЗНЕННОГО ЦИКЛА программного обеспечения часто использует повторные действия (итерацию). Использование итерации делает возможным:

- исследование осуществимости других проектов программного обеспечения;
- разработку различных ПРОГРАММНЫХ ЭЛЕМЕНТОВ в различные сроки;
- обеспечение выпуска различных ВЕРСИЙ программного обеспечения;
- исправление ошибок, сделанных во время ПРОЦЕССА разработки программного обеспечения.

МЭК 62304 требует итерации деятельности по МЕНЕДЖМЕНТУ РИСКА, а также координации с деятельностью по проектированию СИСТЕМЫ на всем протяжении ЖИЗНЕННОГО ЦИКЛА программного обеспечения. Например, во время разработки программного обеспечения п. 5.2.4 МЭК 62304 требует повторного ОЦЕНИВАНИЯ РИСКА МЕДИЦИНСКОГО ИЗДЕЛИЯ, когда установлены требования к программному обеспечению. Такое переоценивание может вызвать потребность в обновлении спецификаций требований к СИСТЕМЕ и ОЦЕНКЕ РИСКА МЕДИЦИНСКОГО ИЗДЕЛИЯ. ОЦЕНИВАНИЕ РИСКА должно повторяться на всех стадиях от требований к АРХИТЕКТУРЕ и проекту до реализации программного обеспечения.

ИСО 14971 не устанавливает ПРОЦЕСС проектирования и разработки и в основном требует, чтобы действия по МЕНЕДЖМЕНТУ РИСКА были предприняты до и после (не во время) осуществления проектирования (включая меры по УПРАВЛЕНИЮ РИСКОМ). Например, когда меры по МЕНЕДЖМЕНТУ РИСКА уже были выполнены, ИСО 14971 требует, чтобы они были проанализированы на предмет того, что они не вызвали дополнительных ОПАСНОСТЕЙ и ОПАСНЫХ СИТУАЦИЙ. Это не должно рассматриваться как инструкция по выполнению анализа только после того, как выполнение мер завершено. Это должно рассматриваться с точки зрения пользы для реагирования на любые дополнительные ОПАСНОСТИ, как только они становятся очевидными. Это подразумевает итерацию в рамках реализации мер по УПРАВЛЕНИЮ РИСКОМ.

Важно, чтобы все РЕЗУЛЬТАТЫ в любой момент времени находились и сохранялись согласованными и непротиворечивыми. Итерация является угрозой согласованности РЕЗУЛЬТАТОВ. Именно поэтому важно использовать четкое управление конфигурацией с целью обеспечения того, что все последствия

изменений идентифицированы и все связанные с ними РЕЗУЛЬТАТЫ обновлены после изменения. Это особенно важно, если предусмотрено сложное программное обеспечение, поскольку оно может быстро меняться, и любое небольшое с виду изменение может иметь неожиданные побочные эффекты. Вся информация, относящаяся к программному обеспечению, должна обновляться немедленно, чтобы не возникало недопонимания между инженерами. Предложения в отношении изменений программного обеспечения проверяются на побочные эффекты, особенно на те, которые влияют на БЕЗОПАСНОСТЬ. Это может привести к итерации этапов ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА.

3.1.3 Упреждающий или реактивный подход к проектированию БЕЗОПАСНОСТИ

МЕНЕДЖМЕНТ РИСКА должен начинаться на ранних стадиях проектирования с надежных входных данных в отношении спецификации МЕДИЦИНСКОГО ИЗДЕЛИЯ с учетом БЕЗОПАСНОСТИ. Упреждающий подход к проекту предпочтительнее реактивного. При упреждающем подходе БЕЗОПАСНОСТЬ учитывается вместе с другими потребностями клиента и принимается в качестве начального требования. Хотя реактивный метод иногда неизбежен (например, когда обновляется уже выпущенная в обращение продукция), упреждающий подход — обычно самый эффективный, быстрый и дешевый способ получить безопасное МЕДИЦИНСКОЕ ИЗДЕЛИЕ.

Преимущества упреждающего проектирования БЕЗОПАСНОСТИ:

- с самого начала спецификация СИСТЕМЫ включает не только то, что МЕДИЦИНСКОЕ ИЗДЕЛИЕ должно делать, но и определяет поведение СИСТЕМЫ, которого следует избегать с целью снижения РИСКА;

- с самого начала АРХИТЕКТУРА СИСТЕМЫ может планироваться с целью обеспечения возможности демонстрации того, что обеспечиваются желаемые характеристики и при этом исключаются или предупреждаются небезопасные состояния;

- в то время как АРХИТЕКТУРА завершена до состояния готового проекта, меры по УПРАВЛЕНИЮ РИСКОМ могут разрабатываться без доработки;

- выбор подходов к БЕЗОПАСНОСТИ и мер по УПРАВЛЕНИЮ РИСКОМ может быть сделан достаточно рано (например, БЕЗОПАСНОСТЬ, обусловленная проектом, может быть максимально увеличена, а информация по БЕЗОПАСНОСТИ сведена к минимуму).

3.1.4 Характеристики безопасности СИСТЕМ, включающих программное обеспечение

Крайне желательные характеристики безопасности СИСТЕМ включают:

a) использование простых аппаратных механизмов обеспечения БЕЗОПАСНОСТИ во избежание чрезмерных требований к ПРОГРАММНЫМ ЭЛЕМЕНТАМ, СВЯЗАННЫХ С БЕЗОПАСНОСТЬЮ;

b) использование только очень простых ПРОГРАММНЫХ ЭЛЕМЕНТОВ, СВЯЗАННЫХ С БЕЗОПАСНОСТЬЮ;

c) распределение СВЯЗАННЫХ С БЕЗОПАСНОСТЬЮ ПРОГРАММНЫХ ЭЛЕМЕНТОВ между некоторым числом независимых процессоров;

d) достаточные аппаратные возможности для запуска всего СВЯЗАННОГО С БЕЗОПАСНОСТЬЮ программного обеспечения, когда это необходимо и без одобрения;

e) использование детерминированного подхода к срокам проектирования программного обеспечения;

f) надлежащая обработка отказа, например:

1) предупреждение пользователя об отказе и предоставление возможностей для информированного вмешательства;

2) обеспечение ограниченной функциональности в условиях отказа;

3) безопасное отключение, когда это возможно в условиях отказа;

4) быстрое восстановление после отказа.

g) средства, препятствующие изменению программного кода в среде его выполнения или через самомодификацию, или в результате ввода данных;

h) средства определения и (или) предотвращения вмешательства в СВЯЗАННЫЕ С БЕЗОПАСНОСТЬЮ данные.

3.2 Ответственность высшего руководства

Текст ИСО 14971

3.2 Ответственность высшего руководства

ВЫСШЕЕ РУКОВОДСТВО должно обеспечивать свидетельства своей приверженности ПРОЦЕССУ МЕНЕДЖМЕНТА РИСКА посредством.

- обеспечения необходимыми ресурсами;
- назначения квалифицированного персонала (см. 3.3) для целей МЕНЕДЖМЕНТА РИСКА.

ВЫСШЕЕ РУКОВОДСТВО должно:

- разрабатывать и документировать политику установления критериев допустимости РИСКА. Данная политика должна гарантировать, что установленные критерии основаны на применимых национальных и региональных нормативных документах и соответствующих международных стандартах, а также учитывают доступную информацию, такую как современный уровень научно-технического развития и потребности заинтересованных сторон;
- проводить анализ пригодности ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА в запланированные промежутки времени для обеспечения постоянной результативности данного ПРОЦЕССА и документировать все решения и предпринятые действия. Если ИЗГОТОВИТЕЛЬ имеет действующую систему менеджмента качества, то данный анализ может быть частью анализа его системы менеджмента качества.

Примечание — Вышеуказанные документы могут быть включены в документы системы менеджмента качества ИЗГОТОВИТЕЛЯ, и на них могут быть ссылки в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА.

Соответствие требованиям данного подраздела проверяют путем контроля необходимых документов.

Как ИСО 14971, так и МЭК 62304 предполагают наличие системы менеджмента качества. Требования к ВЫСШЕМУ РУКОВОДСТВУ при МЕНЕДЖМЕНТЕ РИСКА перечислены в ИСО 14971, п. 3.2.

Примечание — п. 3.1 ИСО 14971 устанавливает, что МЕНЕДЖМЕНТ РИСКА может быть составной частью системы менеджмента качества, а п. 4.1 МЭК 62304 устанавливает, что демонстрация способности ИЗГОТОВИТЕЛЯ последовательно выполнять требования потребителей и применимые регулирующие требования может осуществляться при помощи системы менеджмента качества, соответствующей ИСО 13485, или системы менеджмента качества, требуемой национальным регулированием. МЭК 62304 также содержит рекомендации по положениям приложения В.4, 4.1, устанавливая, что необходимо определить менеджмент риска как неотъемлемую часть системы менеджмента качества как общих рамок для применения подходящих инженерных методов и техник программирования.

ВЫСШЕЕ РУКОВОДСТВО отвечает за внедрение необходимой организационной структуры, достаточные ресурсы, отчетность и подготовку (см. 3.3) для эффективного ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА, а также для БЕЗОПАСНОСТИ проекта и технической поддержки ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКОГО ИЗДЕЛИЯ.

ИЗГОТОВИТЕЛЬ может рассмотреть вопрос аутсорсинга ПРОЦЕССОВ разработки или технического обслуживания программного обеспечения (например, проектирования, воплощения, тестирования или технического обслуживания). В таких ситуациях ВЫСШЕЕ РУКОВОДСТВО все равно полностью ответственно за обеспечение того, что соответствующие действия по МЕНЕДЖМЕНТУ РИСКА были выполнены при аутсорсинге ПРОЦЕССОВ разработки или технической поддержки программного обеспечения, а также ответственно за то, что меры по УПРАВЛЕНИЮ РИСКОМ применены надлежащим образом.

Если разработка программного обеспечения передана на аутсорсинг, ИЗГОТОВИТЕЛИ с помощью подходящих договорных соглашений должны в достаточной мере обеспечить управление программным обеспечением и его проектированием, чтобы выполнить все требования по МЕНЕДЖМЕНТУ РИСКА, установленные ИСО 14971, на протяжении всего ЖИЗНЕННОГО ЦИКЛА МЕДИЦИНСКОГО ИЗДЕЛИЯ включая коррекцию АНОМАЛИЙ после того, как программное обеспечение уже выпущено.

ИЗГОТОВИТЕЛЬ должен рассмотреть вопрос об установлении требований к поставщикам (см. ИСО 13485) [1], п. 7.4 по управлению поставщиками), требуя от них продемонстрировать, например:

- эффективный МЕНЕДЖМЕНТ РИСКА в соответствии с ИСО 14971;
- эффективную практику разработки программного обеспечения в соответствии с МЭК 62304;

- способность предоставить ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МЕДИЦИНСКИХ ИЗДЕЛИЙ, которое равным образом удовлетворяет требованиям клиента и применимым регулирующим требованиям.

Если разработаны меры по УПРАВЛЕНИЮ РИСКОМ, примененные к аутсорсинговым ПРОЦЕССАМ или продуктам, эти меры и их значимость должны быть документированы и четко установлены в пределах договорного соглашения с поставщиком.

3.3 Квалификация персонала

3.3.1 Общие положения

Текст ИСО 14971

3.3 Квалификация персонала

Персонал, выполняющий задачи по МЕНЕДЖМЕНТУ РИСКА, должен иметь знания и опыт, обеспечивающие возможность выполнения данных задач. Квалификация персонала должна включать знание рассматриваемых (или подобных) МЕДИЦИНСКИХ ИЗДЕЛИЙ, опыт их применения, а также владение применяемыми технологиями и методами МЕНЕДЖМЕНТА РИСКА. ЗАПИСИ о необходимой квалификации персонала следует поддерживать в рабочем состоянии.

Примечание — Задачи по МЕНЕДЖМЕНТУ РИСКА могут решать представители разных функциональных подразделений с учетом имеющихся у них специальных знаний.

Соответствие требованиям данного подраздела проверяют путем контроля вышеуказанных записей.

Члены группы, участвующие в разработке и поддержании ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СИСТЕМЫ, должны иметь знания и опыт, соответствующие поставленным перед ними ЗАДАЧАМ. Чрезвычайно важно, чтобы лицо, назначенное для выполнения ЗАДАЧ, связанных с обработкой (импликацией) данных МЕНЕДЖМЕНТА РИСКА, обладало необходимыми знаниями по МЕНЕДЖМЕНТУ РИСКА. Участие междисциплинарной группы, включая клинических экспертов (например, экспертов по клинической поддержке и техническому обслуживанию, экспертов по другим связанным областям), инженеров-программистов, системных проектировщиков, экспертов по проектированию эксплуатационной пригодности и в других областях, а также степень и тип их взаимодействия при программировании и испытаниях, должно рассматриваться в отношении МЕНЕДЖМЕНТА РИСКА.

Это может потребовать разработки учебных программ для отдельных лиц в целях обеспечения полного понимания требуемой деятельности и проводимых мероприятий.

Кроме того, должна быть рассмотрена квалификация каждого участника группы по МЕНЕДЖМЕНТУ РИСКА по отношению к программному обеспечению, в результате чего может потребоваться специальная подготовка.

Следующие подпункты должны обеспечивать обзор области требуемых знаний, которые следует учитывать.

3.3.2 ПРЕДУСМОТРЕННОЕ ПРИМЕНЕНИЕ — знания предметной области

На всех стадиях проектирования МЕДИЦИНСКОГО ИЗДЕЛИЯ важно использовать знания о ПРЕДУСМОТРЕННОМ ПРИМЕНЕНИИ. Это особенно важно как для разработчиков программного обеспечения, так и для персонала, осуществляющего МЕНЕДЖМЕНТ РИСКА программного обеспечения. Сложное поведение программного обеспечения может легко способствовать неправильному применению или введению в заблуждение пользователей, что может привести к непредвиденным ранее ОПАСНОСТЯМ и ОПАСНЫМ СИТУАЦИЯМ. Тщательная оценка клинической практики позволяет менеджерам по РИСКУ идентифицировать ОПАСНОСТИ и ОПАСНЫЕ СИТУАЦИИ, а также позволит инженерам-программистам избегать ОПАСНОСТЕЙ и ОПАСНЫХ СИТУАЦИЙ и разрабатывать меры по УПРАВЛЕНИЮ РИСКОМ.

ИЗГОТОВИТЕЛИ должны обеспечить, чтобы клинические эксперты (например, эксперты по клинической поддержке и техническому обслуживанию, эксперты по другим связанным областям) были способны участвовать в деятельности по проектированию и деятельности по МЕНЕДЖМЕНТУ РИСКА или, по крайней мере, давать консультации.

Кроме того, ИЗГОТОВИТЕЛЯМ следует рассматривать необходимость обучения менеджеров по РИСКУ и инженеров-программистов клиническому применению МЕДИЦИНСКОГО ИЗДЕЛИЯ.

3.3.3 Опыт программирования и подход

Опытные разработчики и лица, проводящие испытания программного обеспечения, учатся реалистичному отношению к сложности обнаружения всех дефектов программного обеспечения во время испытаний и, следовательно, к относительной концентрации дефектов, остающихся после испытаний. Важно включать опытных сотрудников в группу по разработке программного обеспечения и давать им соответствующие полномочия наставника, чтобы обучать, наблюдать и давать задания менее опытным сотрудникам.

Назначение опытных сотрудников особенно важно для выполнения следующих ЗАДАЧ в отношении программного обеспечения:

- идентификации того, каким образом программное обеспечение может выйти из строя;
- анализа РИСКОВ, связанных с отказом программного обеспечения;
- идентификации мер по УПРАВЛЕНИЮ РИСКОМ;
- анализа сообщений о проблемах, полученных после выпуска программного обеспечения;
- разработки и внедрения изменений, особенно тех, которые осуществляются после выпуска программного обеспечения.

Во всех этих ЗАДАЧАХ опыт приводит к пониманию того, что может пойти неправильно в программном обеспечении и в ПРОЦЕССАХ разработки программного обеспечения, а также к пониманию сложностей в осуществлении изменений при поддержании целостности проекта программного обеспечения.

3.4 План МЕНЕДЖМЕНТА РИСКА

3.4.1 Общие положения

Текст ИСО 14971

3.4 План МЕНЕДЖМЕНТА РИСКА

Деятельность по МЕНЕДЖМЕНТУ РИСКА необходимо планировать. Ввиду этого ИЗГОТОВИТЕЛЬ должен составить и документировать план МЕНЕДЖМЕНТА РИСКА для рассматриваемого МЕДИЦИНСКОГО ИЗДЕЛИЯ в соответствии с ПРОЦЕССОМ МЕНЕДЖМЕНТА РИСКА. План МЕНЕДЖМЕНТА РИСКА должен быть частью файла МЕНЕДЖМЕНТА РИСКА.

Данный план должен включать как минимум:

a) объем применения запланированной деятельности по МЕНЕДЖМЕНТУ РИСКА, в том числе идентификацию и описание МЕДИЦИНСКОГО ИЗДЕЛИЯ и стадий его жизненного цикла, к которым применим каждый элемент плана;

b) распределение ответственности и полномочий;

c) требования к анализу деятельности по МЕНЕДЖМЕНТУ РИСКА;

d) критерии допустимости РИСКА, основанные на политике ИЗГОТОВИТЕЛЯ по установлению допустимого РИСКА, включая случаи, когда вероятность причинения ВРЕДА не может быть определена;

e) действия по ВЕРИФИКАЦИИ;

f) действия по сбору и анализу информации, относящейся к МЕНЕДЖМЕНТУ РИСКА, на производственной и ПОСТПРОИЗВОДСТВЕННОЙ стадиях.

Примечания

1 Руководящие указания по разработке плана МЕНЕДЖМЕНТА РИСКА см. в приложении F.

2 Необязательно все элементы плана МЕНЕДЖМЕНТА РИСКА разрабатывать одновременно. План или его элементы можно разрабатывать поэтапно.

3 Критерии допустимости РИСКА имеют большое значение для определения конечной результативности ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА. Для каждого плана МЕНЕДЖМЕНТА РИСКА изготовителю следует выбирать надлежащие критерии допустимости РИСКА.

Среди прочих можно рассматривать следующие варианты:

- построение матрицы (рисунки D.4 и D.5), иллюстрирующей допустимые и недопустимые комбинации вероятности причинения ВРЕДА и ТЯЖЕСТИ ВРЕДА;
- дальнейшее подразделение области матрицы («пренебрежимо малый РИСК», «допустимый РИСК при условии его минимизации» и т. д.) с требованием к РИСКАМ, чтобы они сначала были уменьшены, насколько это практически осуществимо, прежде чем признать их допустимыми (см. D.8).

Любой из вариантов следует выбирать в соответствии с политикой ИЗГОТОВИТЕЛЯ в отношении установления критериев допустимости РИСКА и на основании применимых национальных или региональных нормативных документов, а также соответствующих международных стандартов с учетом доступной информации, такой как современный уровень научно-технического развития и интересы заинтересованных сторон (см. 3.2). Руководство по установлению данных критериев см. в D.4.

При внесении изменений в план МЕНЕДЖМЕНТА РИСКА в течение ЖИЗНЕННОГО ЦИКЛА МЕДИЦИНСКОГО ИЗДЕЛИЯ необходимо сделать запись об изменениях в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА.

Соответствие требованиям данного подраздела проверяют путем контроля файла МЕНЕДЖМЕНТА РИСКА.

План МЕНЕДЖМЕНТА РИСКА должен учитывать тот факт, что программное обеспечение является частью МЕДИЦИНСКОГО ИЗДЕЛИЯ и должно включать:

- описание МЕДИЦИНСКОГО ИЗДЕЛИЯ и то, какие функции МЕДИЦИНСКОГО ИЗДЕЛИЯ будут выполняться программным обеспечением;
- заявление о том, что программное обеспечение будет разрабатываться в соответствии с МЭК 62304;
- ссылку на аспекты разработки программного обеспечения, которые являются специфичными для осуществления МЕНЕДЖМЕНТА РИСКА программного обеспечения (см. примечание);
- критерии допустимости РИСКА в отношении РИСКОВ, вызываемых программным обеспечением или управляемых программным обеспечением, если они отличаются от критериев допустимости для других компонентов МЕДИЦИНСКОГО ИЗДЕЛИЯ.

Примечание — Ссылка на план разработки программного обеспечения может быть самым простым способом для включения специфичных аспектов разработки программного обеспечения для осуществления МЕНЕДЖМЕНТА РИСКА. См. также 3.4.2 и 3.4.3, в которых обсуждается взаимосвязь между планом МЕНЕДЖМЕНТА РИСКА и планом разработки программного обеспечения, а также определенные связанные с РИСКОМ темы плана разработки программного обеспечения согласно МЭК 62304.

Одной из причин, по которой критерии допустимости РИСКОВ, вызываемых или управляемых программным обеспечением, могут отличаться от критериев допустимости для других компонентов, является то, что вероятность ВРЕДА не может быть оценена. В этом случае критерии допустимости РИСКА определяются исходя из ТЯЖЕСТИ ВРЕДА. (См. 4.4.3 для обсуждения вероятности ВРЕДА, вызванного программным обеспечением). Если можно считать, что ОПАСНОСТЬ имеет небольшое практическое последствие, РИСК может быть оценен как допустимый, и в таком случае не требуется никаких мер по УПРАВЛЕНИЮ РИСКОМ. Однако в случае существенных ОПАСНОСТЕЙ, то есть ОПАСНОСТЕЙ, которые могут привести к ВРЕДУ высокой ТЯЖЕСТИ, не может быть определен уровень подверженности ОПАСНОСТИ, который бы соответствовал настолько низкому РИСКУ, чтобы РИСК являлся допустимым. В этом случае должны быть разработаны меры по УПРАВЛЕНИЮ РИСКОМ.

Критерии допустимости РИСКА для ОСТАТОЧНОГО РИСКА, где вероятность не может быть определена, следует принимать с учетом мер по УПРАВЛЕНИЮ РИСКОМ, которые были осуществлены, а также результативности этих мер по УПРАВЛЕНИЮ РИСКОМ в снижении вероятности возникновения вреда. Меры по УПРАВЛЕНИЮ РИСКОМ должны включать все приемлемые практически осуществимые меры, удовлетворять применимым стандартам и регулирующим требованиям, а также быть современными (см. ИСО 14971, приложение D. 4).

При планировании работ, связанных со сбором и анализом производственной и ПОСТПРОИЗВОДСТВЕННОЙ информации, должны приниматься во внимание следующие специфичные для программного обеспечения аспекты:

- если используется ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ НЕИЗВЕСТНОГО ПРОИСХОЖДЕНИЯ (ПОНП), то должны быть запланированы активный мониторинг, а также ОЦЕНИВАНИЕ общедоступного списка АНОМАЛИЙ и информации об эксплуатационных характеристиках ПОНП. Где возможно, это должно поддерживаться в соответствии с договорным соглашением с поставщиком ПОНП, заключаемым при его приобретении. Если пользователи МЕДИЦИНСКОГО ИЗДЕЛИЯ могут (преднамеренно или нет) модифицировать ПОНП МЕДИЦИНСКОГО ИЗДЕЛИЯ самостоятельно (например, применяя исправления ПОНП или обновления), то следует тщательно рассмотреть вопрос о проведении мониторинга при выпуске в обращение новых версий ПОНП. См. раздел 9 относительно ПОНП и ПОСТПРОИЗВОДСТВЕННОГО мониторинга;
- изготовитель должен по запросу идентифицировать версию программного обеспечения и сообщить ее инициатору жалобы.

3.4.2 Взаимосвязь между планом МЕНЕДЖМЕНТА РИСКА и планом разработки программного обеспечения

Требования ИСО 14971 для плана МЕНЕДЖМЕНТА РИСКА и требования МЭК 62304 для плана разработки программного обеспечения не должны рассматриваться в качестве требований конкретных документов с конкретными наименованиями. Планирование элементов может быть воплощено в любых документах, которые подходят для системы менеджмента качества ИЗГОТОВИТЕЛЯ, при условии, что:

- комбинация документов по планированию удовлетворяет требованиям обоих стандартов и осуществлена таким способом, который поддается проверке;
- все планы совместимы друг с другом;
- все планы доступны для использования в установленные сроки;
- все планы постоянно обновляются, чтобы отражать изменяющиеся обстоятельства.

3.4.3 Специфические связанные с РИСКОМ темы плана разработки программного обеспечения согласно МЭК 62304

План разработки программного обеспечения должен обеспечить, чтобы ПРОЦЕСС разработки программного обеспечения, стандарты, методы и средства, связанные с разработкой программного обеспечения (описанные в плане разработки программного обеспечения согласно МЭК 62304, раздел 4), были эффективными мерами по УПРАВЛЕНИЮ РИСКОМ (см. 6.2.2.6 для обсуждения ПРОЦЕССА как меры по УПРАВЛЕНИЮ РИСКОМ). Это может быть достигнуто путем предоставления доказательств другими организациями, поставщиками, а также от других проектов в рамках организации. Если это неизвестно, осуществляется планирование и ВЕРИФИКАЦИЯ результативности внутри проекта.

При разработке ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА МЕДИЦИНСКОГО ИЗДЕЛИЯ должны учитываться аспекты, специфичные для МЕНЕДЖМЕНТА РИСКА программного обеспечения, такие как стандарты безопасности кодирования, методы ВЕРИФИКАЦИИ (например, формальные доказательства, экспертные оценки, сквозные, моделирующие и т. д.) и использоваться синтаксические и логические проверки. Если такие аспекты рассматриваются как меры по УПРАВЛЕНИЮ РИСКОМ, они также подлежат ВЕРИФИКАЦИИ (см. таблицу В.2 для примеров верификации мер по УПРАВЛЕНИЮ РИСКОМ).

Деятельность по МЕНЕДЖМЕНТУ РИСКА программного обеспечения должна осуществляться на каждой стадии разработки МЕДИЦИНСКОГО ИЗДЕЛИЯ в планах, ПРОЦЕДУРАХ, обучении, если применимо.

3.5 ФАЙЛ МЕНЕДЖМЕНТА РИСКА

Текст ИСО 14971

3.5 ФАЙЛ МЕНЕДЖМЕНТА РИСКА

Для рассматриваемого МЕДИЦИНСКОГО ИЗДЕЛИЯ ИЗГОТОВИТЕЛЬ должен создать и поддерживать в рабочем состоянии ФАЙЛ МЕНЕДЖМЕНТА РИСКА. В дополнение к требованиям других разделов настоящего стандарта ФАЙЛ МЕНЕДЖМЕНТА РИСКА должен обеспечивать возможность прослеживания каждой идентифицированной ОПАСНОСТИ при:

- АНАЛИЗЕ РИСКА;
- ОЦЕНИВАНИИ РИСКА;
- выполнении и ВЕРИФИКАЦИИ мер по УПРАВЛЕНИЮ РИСКОМ;
- оценивании допустимости любого(ых) ОСТАТОЧНОГО(ЫХ) РИСКА(ОВ).

Примечания

1 ЗАПИСИ и другие документы, составляющие ФАЙЛ МЕНЕДЖМЕНТА РИСКА, могут быть частью других документов и файлов, требуемых, например, системой менеджмента качества ИЗГОТОВИТЕЛЯ. ФАЙЛ МЕНЕДЖМЕНТА РИСКА необязательно должен непосредственно включать все записи и другие документы, относящиеся к настоящему стандарту. Однако он должен содержать, по меньшей мере, ссылки или указания на все требуемые документы. Изготовителю следует своевременно собрать ссылочную информацию в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА.

2 ФАЙЛ МЕНЕДЖМЕНТА РИСКА может быть представлен в любой форме или на любом носителе информации.

ПРОЦЕСС программного обеспечения должен установить систему ПРОСЛЕЖИВАЕМОСТИ, которая начинается с идентификации ОПАСНОСТЕЙ, связанных с программным обеспечением, и мер по УПРАВЛЕНИЮ РИСКОМ программного обеспечения. Эта система должна прослеживать выполнение требований, связанных с БЕЗОПАСНОСТЬЮ программного обеспечения и требований к ЭЛЕМЕНТАМ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.

Все вышеупомянутое должно прослеживаться до ВЕРИФИКАЦИИ программного обеспечения (см. МЭК 62304, п. 7.3.3).

Поскольку программное обеспечение может часто изменяться во время разработки и быть реализовано в различных ВЕРСИЯХ, часть файла МЕНЕДЖМЕНТА РИСКА, относящаяся к программному обеспечению, может подвергаться изменениям и существовать во множестве ВЕРСИЙ.

В таблице 1 перечислены требования МЭК 62304 к документации, которую следует включать в ФАЙЛ МЕНЕДЖМЕНТА РИСКА в дополнение к требованиям ИСО 14971.

Таблица 1 — Требования МЭК 62304 к документации, которую следует включать в ФАЙЛ МЕНЕДЖМЕНТА РИСКА в дополнение к требованиям ИСО 14971.

Пункты и подпункты МЭК 62304	Документация ФАЙЛА МЕНЕДЖМЕНТА РИСКА
4.3 c)	Класс БЕЗОПАСНОСТИ программного обеспечения, назначенный каждой ПРОГРАММНОЙ СИСТЕМЕ
4.3 f)	Логическое объяснение для использования более низкого класса БЕЗОПАСНОСТИ (чем у ПРОГРАММНОЙ СИСТЕМЫ) для ПРОГРАММНОГО ЭЛЕМЕНТА в ПРОГРАММНОЙ СИСТЕМЕ, который не осуществляет СВЯЗАННЫХ С БЕЗОПАСНОСТЬЮ функций
7.1.4	Возможные причины, по которым ПРОГРАММНЫЙ ЭЛЕМЕНТ может способствовать ОПАСНОЙ СИТУАЦИИ
7.2.1	Меры по УПРАВЛЕНИЮ РИСКОМ, определенные для каждой возможной причины, по которой ПРОГРАММНЫЙ ЭЛЕМЕНТ может способствовать ОПАСНОЙ СИТУАЦИИ
7.3.2	Если мера по УПРАВЛЕНИЮ РИСКОМ реализуется как ПРОГРАММНЫЙ ЭЛЕМЕНТ, то ИЗГОТОВИТЕЛЬ должен оценить данную меру по УПРАВЛЕНИЮ РИСКОМ с целью идентификации и документирования любых новых последовательностей событий, которые могут привести к ОПАСНОЙ СИТУАЦИИ
9.5	ИЗГОТОВИТЕЛЬ должен поддерживать ЗАПИСИ СООБЩЕНИЙ О ПРОБЛЕМАХ и их разрешении, включая их ВЕРИФИКАЦИЮ. Если применимо, ИЗГОТОВИТЕЛЬ должен обновлять ФАЙЛ МЕНЕДЖМЕНТА РИСКА

4 АНАЛИЗ РИСКА

4.1 ПРОЦЕСС АНАЛИЗА РИСКА

Текст ИСО 14971

4 АНАЛИЗ РИСКА

4.1 ПРОЦЕСС АНАЛИЗА РИСКА

АНАЛИЗ РИСКА для рассматриваемого МЕДИЦИНСКОГО ИЗДЕЛИЯ необходимо проводить в соответствии с 4.2—4.4. Деятельность по запланированному АНАЛИЗУ РИСКА, а также результаты АНАЛИЗА РИСКА должны быть зарегистрированы в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА.

Примечания

1 Если доступны результаты АНАЛИЗА РИСКА или другая относящаяся к РИСКУ информация для подобного МЕДИЦИНСКОГО ИЗДЕЛИЯ, то их можно использовать в качестве отправной точки при новом анализе. Степень сопоставимости зависит от различий между изделиями и от того, могут ли данные различия стать источником новых ОПАСНОСТЕЙ или существенных различий в готовой продукции, характеристиках, функционировании или результатах применения. Возможность применения уже имеющегося АНАЛИЗА РИСКА основана также на систематическом оценивании влияния изменений на развитие ОПАСНЫХ СИТУАЦИЙ.

2 Некоторые методы АНАЛИЗА РИСКА описаны в приложении G.

3 Дополнительные руководящие указания по методам АНАЛИЗА РИСКА МЕДИЦИНСКИХ ИЗДЕЛИЙ для диагностики *in-vitro* приведены в приложении H.

4 Дополнительные руководящие указания по методам АНАЛИЗА РИСКА в отношении токсикологических ОПАСНОСТЕЙ приведены в приложении I.

В дополнение к записям, требуемым в 4.2—4.4, документы, относящиеся к проведению и результатам АНАЛИЗА РИСКА, должны включать как минимум:

a) описание и идентификацию рассматриваемого МЕДИЦИНСКОГО ИЗДЕЛИЯ;

b) идентификацию лица (лиц) и организации, выполнивших АНАЛИЗ РИСКА;

c) область применения и дату проведения АНАЛИЗА РИСКА.

5 Область применения АНАЛИЗА РИСКА может быть очень широкой (например, разработка нового изделия, в отношении применения которого у ИЗГОТОВИТЕЛЯ мало опыта или данный опыт вообще отсутствует) или ограниченной (например, анализ влияния привносимых изменений на выпускаемое изделие, о котором в файлах ИЗГОТОВИТЕЛЯ имеется обширная информация).

Соответствие требованиям данного подраздела проверяют путем контроля файла МЕНЕДЖМЕНТА РИСКА.

Как установлено в ИСО 14971, АНАЛИЗ РИСКА является термином, который используется, чтобы охватить три различных вида деятельности:

- идентификацию ПРЕДУСМОТРЕННОГО ПРИМЕНЕНИЯ;
- идентификацию известных или прогнозируемых ОПАСНОСТЕЙ (и их причин);
- определение РИСКА от каждой ОПАСНОСТИ и ОПАСНОЙ СИТУАЦИИ.

Важно заметить, что АНАЛИЗ РИСКА с целью обеспечения его эффективности осуществляется как неотъемлемая часть всего ПРОЦЕССА разработки программного обеспечения, а не как одно или два отдельных события потому, что информация об ОПАСНОСТИ и виде отказа накапливается по мере осуществления ПРОЦЕССА разработки ЖИЗНЕННОГО ЦИКЛА программного обеспечения и должна учитываться на каждой стадии проектирования.

Поскольку очень трудно определить вероятность АНОМАЛИЙ программного обеспечения, которые могут способствовать ОПАСНЫМ СИТУАЦИЯМ, центр программных аспектов АНАЛИЗА РИСКА направлен на идентификацию потенциальных функций программного обеспечения и АНОМАЛИЙ, которые могут привести к ОПАСНЫМ СИТУАЦИЯМ, а не на определение вероятности. См. 4.4.3 для более подробной информации об определении вероятности.

ТЯЖЕСТЬ ВРЕДА в условиях наихудшего случая, для которого программное обеспечение является способствующим фактором, является первичной информацией для установления уровня тщательности ПРОЦЕССОВ разработки программного обеспечения (см. МЭК 62304, п. 4.3). Информация, приведенная в 4.2, 4.3 и 4.4, предназначена, чтобы помочь идентифицировать определенные для программного обеспечения аспекты эффективного ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА. Кроме того, программные аспекты АНАЛИЗА РИСКА должны быть идентифицированы в создаваемой документации и должны включать как программное обеспечение, используемое для осуществления мер по УПРАВЛЕНИЮ РИСКОМ в отношении отказа аппаратных средств, так и программное обеспечение, вызывающее ОПАСНОСТИ и связанные с ними меры по УПРАВЛЕНИЮ РИСКОМ.

4.2 ПРЕДУСМОТРЕННОЕ ПРИМЕНЕНИЕ и определение характеристик, относящихся к БЕЗОПАСНОСТИ МЕДИЦИНСКОГО ИЗДЕЛИЯ

4.2.1 Общее

Текст ИСО 14971

4.2 ПРЕДУСМОТРЕННОЕ ПРИМЕНЕНИЕ и определение характеристик, относящихся к БЕЗОПАСНОСТИ МЕДИЦИНСКОГО ИЗДЕЛИЯ

Для рассматриваемого МЕДИЦИНСКОГО ИЗДЕЛИЯ ИЗГОТОВИТЕЛЬ должен документировать все случаи ПРЕДУСМОТРЕННОГО ПРИМЕНЕНИЯ и обоснованно прогнозируемого неправильного применения. ИЗГОТОВИТЕЛЬ должен также идентифицировать и документировать все качественные и количественные характеристики, которые могут повлиять на БЕЗОПАСНОСТЬ МЕДИЦИНСКОГО ИЗДЕЛИЯ и при необходимости указать их предельно допустимые значения. Данные документы необходимо поддерживать в рабочем состоянии в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА.

Примечания

1 В контексте настоящего стандарта неправильное применение означает непредусмотренное или ненадлежащее применение МЕДИЦИНСКОГО ИЗДЕЛИЯ.

2 Приложение С содержит вопросы, относящиеся к применению МЕДИЦИНСКОГО ИЗДЕЛИЯ, которые могут стать полезным ориентиром при идентификации характеристик данного изделия, способных повлиять на его БЕЗОПАСНОСТЬ.

Соответствие требованиям данного подраздела проверяют путем контроля файла МЕНЕДЖМЕНТА РИСКА.

Хотя каждое МЕДИЦИНСКОЕ ИЗДЕЛИЕ имеет ПРЕДУСМОТРЕННОЕ ПРИМЕНЕНИЕ, должна быть рассмотрена возможность преднамеренного или непреднамеренного неправильного применения. Хотя это касается не только программного обеспечения, но его использование может привести к повышенному РИСКУ неправильного применения из-за следующего:

- поведение МЕДИЦИНСКОГО ИЗДЕЛИЯ является более сложным и поэтому более трудным для управления или понимания;
- пользователь может чрезмерно полагаться на программное обеспечение, не понимая его возможностей и ограничений;
- МЕДИЦИНСКОЕ ИЗДЕЛИЕ может быть настраиваемым, и пользователь может быть не знаком с текущей конфигурацией;
- МЕДИЦИНСКИЕ ИЗДЕЛИЯ могут взаимодействовать с медицинскими или НЕМЕДИЦИНСКИМИ ИЗДЕЛИЯМИ способом, который ИЗГОТОВИТЕЛЬ МЕДИЦИНСКОГО ИЗДЕЛИЯ не может предвидеть во всех деталях.

Лицо, ответственное за создание системных требований, и разработчик программного обеспечения совместно ответственны за ЗАПИСИ в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА о ПРЕДУСМОТРЕННОМ ПРИМЕНЕНИИ СИСТЕМЫ, включая программное обеспечение, а также за все системные и программные требования, которые относятся к БЕЗОПАСНОСТИ и безопасному применению. Разработчик программного обеспечения конкретно отвечает за идентификацию аспектов ПРЕДУСМОТРЕННОГО ПРИМЕНЕНИЯ, которые являются слишком трудноуловимыми, чтобы быть очевидными на уровне СИСТЕМЫ.

4.2.2 Интерфейс пользователя

Наличие программного обеспечения делает возможным проектирование гибких пользовательских интерфейсов, которые могут повлиять на поведение пользователей, приводя к новым формам обоснованно прогнозируемого неправильного применения. Общие случаи неправильного применения возникают из-за недопонимания чрезмерно сложного пользовательского интерфейса и слишком большой уверенности в возможностях программного обеспечения избегать ошибок и опасных состояний. Важно предвидеть подобное неправильное применение и изменять конструкцию, чтобы избежать его насколько возможно.

В связи с этим следует использовать маркировки на нескольких языках, особенно когда такая маркировка является мерой по УПРАВЛЕНИЮ РИСКОМ. Особое внимание следует обратить на:

a) различную потребность в размере памяти для различных языков;

b) использование различных кодировок;

c) использование букв вместо символов;

d) использование разных единиц измерения, которые могут потребовать дополнительного масштабирования числовых результатов;

e) формат данных и пунктуацию в числах;

f) различные требования к расположению для различных языков и (или) кодировок;

g)поддержку валидации.

См. МЭК 62366 [5] дополнительно к ИСО 14971 для ПРОЦЕССА эксплуатационной пригодности.

4.2.3 МЕДИЦИНСКИЕ И ВЗАИМОСВЯЗАННЫЕ С НИМИ ИЗДЕЛИЯ

Использование программного обеспечения в МЕДИЦИНСКИХ ИЗДЕЛИЯХ делает возможным ряд взаимосвязей и коммуникаций между МЕДИЦИНСКИМИ и НЕМЕДИЦИНСКИМИ ИЗДЕЛИЯМИ. Такие связи и коммуникации делают возможными новые области применения (и неправильное применение) СИСТЕМЫ, включающей МЕДИЦИНСКИЕ ИЗДЕЛИЯ и взаимосвязанные устройства. Хотя легко предвидеть, что новые виды применения и неправильного применения могут иметь место, для ИЗГОТОВИТЕЛЯ МЕДИЦИНСКИХ ИЗДЕЛИЙ нелегко определить все виды применения и неправильного применения, если взаимосвязи и коммуникации не ограничены.

Вот почему для ИЗГОТОВИТЕЛЕЙ важно определить ограниченный перечень ПРЕДУСМОТРЕННОГО ПРИМЕНЕНИЯ для коммуникационных интерфейсов МЕДИЦИНСКОГО ИЗДЕЛИЯ и проектировать интерфейсы так, чтобы в максимально возможной степени ограничить взаимосвязи и коммуникации только теми из них, которые являются безопасными.

Например, программное обеспечение может, используя встроенный интерфейс МЕДИЦИНСКОГО ИЗДЕЛИЯ, проверить обработку введенных данных на согласованность и достоверность, основываясь на идентификации пользователя, пациента и контекста подготовки данных. Если данные подготовлены где-либо в другом месте и введены в МЕДИЦИНСКОЕ ИЗДЕЛИЕ с использованием сетевой коммуникации, то может оказаться невозможным применить подобную проверку. В таких случаях ИЗГОТОВИТЕЛЬ может проверить программное обеспечение доступными для пользователей сети средствами применения сетевого приложения и (или) ограничения импорта данных проверенными источниками, а также созданием подробного руководства для ответственных лиц, которые отвечают за сетевые коммуникации в условиях медицинского учреждения.

МЭК 80001-1 [6] устанавливает требования к интеграции МЕДИЦИНСКИХ ИЗДЕЛИЙ в информационной сети медицинского учреждения. В частности, устанавливает ответственность ИЗГОТОВИТЕЛЯ и лица, которое интегрирует МЕДИЦИНСКОЕ ИЗДЕЛИЕ в информационную сеть.

4.3 Идентификация ОПАСНОСТЕЙ

Текст ИСО 14971

4.3 Идентификация ОПАСНОСТЕЙ

ИЗГОТОВИТЕЛЬ должен составить перечень известных или прогнозируемых ОПАСНОСТЕЙ, связанных с рассматриваемым МЕДИЦИНСКИМ ИЗДЕЛИЕМ, как для нормальных условий, так и для условий отказа.

Данный перечень необходимо поддерживать в рабочем состоянии в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА.

Примечание — Примеры возможных ОПАСНОСТЕЙ, приведенные в Е.2 и Н.2.4, могут быть использованы как руководство для ИЗГОТОВИТЕЛЯ, выполняющего идентификацию ОПАСНОСТЕЙ.

Соответствие требованиям данного подраздела проверяют путем контроля файла МЕНЕДЖМЕНТА РИСКА.

Цель идентификации ОПАСНОСТИ состоит в том, чтобы сделать возможным анализ всех прогнозируемых ОПАСНОСТЕЙ с целью разработки и реализации эффективных мер по УПРАВЛЕНИЮ РИСКОМ.

В отличие от высокой температуры, электрической энергии и подвешенных грузов, программное обеспечение само по себе не является ОПАСНОСТЬЮ (потенциальным источником ВРЕДА), так как контакт с программным обеспечением не может вызвать травму. Однако программное обеспечение может стать причиной того, что человек подвергается ОПАСНОСТИ, другими словами, оно может способствовать созданию ОПАСНОЙ СИТУАЦИИ. Отказы программного обеспечения (любого рода) часто способствуют переходу ОПАСНОСТИ в ОПАСНУЮ СИТУАЦИЮ.

Таким образом, хотя программное обеспечение редко вводит новые ОПАСНОСТИ, оно часто изменяет ОПАСНУЮ СИТУАЦИЮ. Что еще более важно для ИЗГОТОВИТЕЛЯ, оно может переложить ответственность за предотвращение ОПАСНЫХ СИТУАЦИЙ от пользователя на ИЗГОТОВИТЕЛЯ.

Например, скальпель представляет очевидную ОПАСНОСТЬ порезаться. Однако ИЗГОТОВИТЕЛЬ обычно не несет ответственности за эту ОПАСНОСТЬ вне пределов эргономичного дизайна, поскольку ОПАСНОСТЬ, как предполагается, находится полностью под контролем хирурга. Если скальпель является частью дистанционно управляемой хирургической системы, подобная ОПАСНОСТЬ все же существует, но ответственность за то, чтобы избежать ОПАСНОСТИ порезаться, теперь разделяется с ИЗГОТОВИТЕЛЕМ, который поставляет программное обеспечение, управляющее этим скальпелем.

Это означает, что УПРАВЛЕНИЕ РИСКОМ в отношении некоторых ОПАСНОСТЕЙ, которые раньше, в отсутствие программного обеспечения, зависели исключительно от профессионального применения МЕДИЦИНСКОГО ИЗДЕЛИЯ, теперь переходит на МЕНЕДЖМЕНТ РИСКА со стороны ИЗГОТОВИТЕЛЯ.

Наглядным примером является ОПАСНОСТЬ неправильного лечения из-за ошибочной обработки данных. Это всегда являлось ОПАСНОСТЬЮ, но, когда данные обрабатывались вручную, это не являлось ответственностью ИЗГОТОВИТЕЛЯ. Сейчас многие МЕДИЦИНСКИЕ ИЗДЕЛИЯ используют программное обеспечение для сбора, хранения, обращения или использования данных, что делает такую ОПАСНОСТЬ частью ответственности ИЗГОТОВИТЕЛЯ.

Программное обеспечение может способствовать ОПАСНЫМ СИТУАЦИЯМ несколькими способами, включая некоторые из нижеследующих (см. также приложение В):

- программное обеспечение может правильно реализовывать небезопасные СИСТЕМНЫЕ требования, приводя к поведению, небезопасная природа которого не ощущается до тех пор, пока не возникает фактический ВРЕД;

- спецификация программного обеспечения может неправильно реализовывать СИСТЕМНЫЕ требования, приводя к нежелательному поведению, которое является правильным согласно спецификации программного обеспечения;

- проектирование программного обеспечения и его реализация могут быть ошибочными, приводя к поведению, которое противоречит спецификации программного обеспечения. Очевидные недостатки могут являться результатом неправильного понимания спецификации программного обеспечения и ошибок при преобразовании спецификации в код. Менее очевидные недостатки могут быть обусловлены непредвиденными взаимодействиями между ПРОГРАММНЫМИ ЭЛЕМЕНТАМИ, а также между программным обеспечением и его инфраструктурой, включая аппаратные средства и операционную СИСТЕМУ.

В МЕДИЦИНСКОМ ИЗДЕЛИИ, содержащем программное обеспечение, тщательная и всесторонняя идентификация ОПАСНОСТИ может приводить (на более поздних стадиях ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА) к следующим важным результатам:

- аппаратные меры по УПРАВЛЕНИЮ РИСКОМ, которые могут предотвратить нанесение ВРЕДА программным обеспечением;

- удаление потенциально опасных функций программного обеспечения из спецификации программного обеспечения;

- меры по УПРАВЛЕНИЮ РИСКОМ, которые внедрены в программное обеспечение для предотвращения причинения ВРЕДА (см. МЭК 62304 п. 5.2.3);

- идентификация частей программного обеспечения, которые должны реализовываться с низкой плотностью дефектов, и частей спецификации программного обеспечения, которые должны подвергаться специальным испытаниям (МЭК 62304, п. 4.3);

- идентификация ПРОГРАММНЫХ ЭЛЕМЕНТОВ с более высоким классом БЕЗОПАСНОСТИ, которые должны быть отделены от других ПРОГРАММНЫХ ЭЛЕМЕНТОВ (в программном обеспечении с более низким классом БЕЗОПАСНОСТИ), чтобы предотвратить ВРЕД, возникающий из-за неожиданных побочных эффектов (см. МЭК 62304, п. 4.3 и п. 5.3.5). Дальнейшее обсуждение этого вопроса см. в п. 6.2.2.2.4.

Чтобы полностью идентифицировать ОПАСНОСТИ, нужно хорошо понимать клиническое применение МЕДИЦИНСКОГО ИЗДЕЛИЯ. Кроме того, программное обеспечение представляет собой проблему особой сложности, включая возможные сложные пользовательские интерфейсы. Поэтому идентификация ОПАСНОСТЕЙ программного обеспечения не может быть выполнена без привлечения внешних специалистов. Она должна выполняться на СИСТЕМНОМ уровне междисциплинарной группой, включающей клинических экспертов (таких как эксперты по клинической поддержке и техническому обслуживанию), инженеров-программистов, системных проектировщиков и экспертов по разработке эксплуатационной пригодности (см. также 3.3).

Идентификация ОПАСНОСТИ должна учитывать ВРЕД, который может быть результатом самой природы МЕДИЦИНСКОГО ИЗДЕЛИЯ (например, порез, облучение или поражение пациента электрическим током), а также дополнительные ОПАСНОСТИ, связанные с использованием программного обеспечения. Дополнительные ОПАСНОСТИ могут включать в себя, например:

- предоставление неверной информации врачу или пациенту;

- неправильная идентификация пациента (в случаях, если МЕДИЦИНСКОЕ ИЗДЕЛИЕ хранит информацию о пациенте или рецепты);

- отклонение запроса или его задержка, вызванные АНОМАЛИЕЙ программного обеспечения.

Примечание — Для многих медицинских изделий задержка или отклонение запроса не подразумевают ВРЕДА пациенту.

Идентифицированные ОПАСНОСТИ должны включать в себя как ОПАСНОСТИ, относящиеся к программному обеспечению, которое работает в соответствии с его спецификацией, так и ОПАСНОСТИ, относящиеся к АНОМАЛИЯМ программного обеспечения (см. 6.1).

Часто пользовательский интерфейс МЕДИЦИНСКОГО ИЗДЕЛИЯ становится более сложным из-за программного обеспечения. В частности, МЕДИЦИНСКОЕ ИЗДЕЛИЕ, которое включает программное обеспечение, наверняка будет управлять информацией. Поскольку это может быть оправдано с точки зрения выгоды для пациента, следует учитывать дополнительные ОПАСНОСТИ, которые связаны с неправильной или неправильно используемой информацией, например:

- неправильный ввод данных;
- неправильное чтение пользователем отображаемых результатов;
- перегрузка пользователей чрезмерными данными или излишним числом тревожных сообщений (см. МЭК 62366 [5]).

4.4 Определение РИСКА(ОВ) для каждой ОПАСНОЙ СИТУАЦИИ

4.4.1 Общие положения

Текст ИСО 14971

4.4 Определение РИСКА(ОВ) для каждой ОПАСНОЙ СИТУАЦИИ

Необходимо рассматривать обоснованно прогнозируемые последовательности или комбинации событий, приводящие к возникновению ОПАСНОЙ СИТУАЦИИ, и регистрировать возникающую(ие) ОПАСНУЮ(ЫЕ) СИТУАЦИЮ(И).

Примечания

1 Для идентификации не выявленных ранее ОПАСНЫХ СИТУАЦИЙ можно использовать системные методы, применимые в конкретной ситуации (см. приложение G).

2 Примеры ОПАСНЫХ СИТУАЦИЙ приведены в H.2.4.5 и E.4.

3 Источником ОПАСНЫХ СИТУАЦИЙ могут стать промахи, упущения и заблуждения.

Для каждой идентифицированной ОПАСНОЙ СИТУАЦИИ необходимо определять связанный(е) с ней РИСК(И), используя для этого доступную информацию или данные. Для ОПАСНЫХ СИТУАЦИЙ, в отношении которых не может быть определена вероятность причинения ВРЕДА, должен быть подготовлен перечень возможных последствий применения МЕДИЦИНСКОГО ИЗДЕЛИЯ, используемый при ОЦЕНИВАНИИ и УПРАВЛЕНИИ РИСКОМ. Результаты этой деятельности должны быть зарегистрированы в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА.

Любая система, используемая для качественной или количественной градации вероятности причинения ВРЕДА или ТЯЖЕСТИ ВРЕДА, также должна быть зарегистрирована в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА.

Примечания

1 ОПРЕДЕЛЕНИЕ РИСКА для МЕДИЦИНСКОГО ИЗДЕЛИЯ включает анализ вероятности возникновения РИСКА и последствий применения данного изделия. В зависимости от применения МЕДИЦИНСКОГО ИЗДЕЛИЯ, возможно, достаточно рассмотреть лишь некоторые элементы ПРОЦЕССА ОПРЕДЕЛЕНИЯ РИСКА. Например, в отдельных случаях нет необходимости идти дальше анализа исходной ОПАСНОСТИ и последствий применения (см. также D.3).

2 ОПРЕДЕЛЕНИЕ РИСКА может быть количественным или качественным. Методы ОПРЕДЕЛЕНИЯ РИСКА, в том числе являющиеся следствием систематических отказов, описаны в приложении D. Приложение H содержит информацию, полезную при ОПРЕДЕЛЕНИИ РИСКОВ для МЕДИЦИНСКИХ ИЗДЕЛИЙ для диагностики *in-vitro*.

3 Информацию или данные для ОПРЕДЕЛЕНИЯ РИСКОВ можно получить из:

a) опубликованных стандартов;

b) научно-технической информации;

c) данных о применении подобных МЕДИЦИНСКИХ ИЗДЕЛИЙ, включая опубликованные сведения об инцидентах;

d) данных испытаний эксплуатационной пригодности типичными пользователями;

e) клинических данных;

f) результатов соответствующих исследований;

g) экспертных заключений;

h) схем внешней оценки качества.

Соответствие требованиям данного подраздела проверяют путем контроля файла МЕНЕДЖМЕНТА РИСКА.

Чтобы определить РИСК, связанный с программным обеспечением, необходимо идентифицировать ОПАСНЫЕ СИТУАЦИИ, которые включают в себя программное обеспечение. Программное обеспечение может быть или причиной, инициирующей последовательность событий, приводящих к ОПАСНОЙ СИТУАЦИИ, или являться причиной ОПАСНОЙ СИТУАЦИИ в любом другом месте последовательности событий, как в случае программного обеспечения, предназначенного для обнаружения сбоя аппаратных средств. Программное обеспечение может включать компоненты ПОНП или повторное использование ранее разработанных компонентов.

ОПРЕДЕЛЕНИЕ РИСКА основано на вероятности ВРЕДА и на ТЯЖЕСТИ ВРЕДА от каждой идентифицированной ОПАСНОЙ СИТУАЦИИ. Поскольку очень трудно определить вероятность ВРЕДА, являющегося результатом АНОМАЛИИ программного обеспечения (см. 4.4.3), вероятность появления АНОМАЛИИ программного обеспечения должна использоваться с осторожностью при определении РИСКА от ОПАСНОЙ СИТУАЦИИ, включая АНОМАЛИИ программного обеспечения в последовательности событий, приводящих к причинению ВРЕДА.

4.4.2 Методы идентификации

Для идентификации потенциальной роли программного обеспечения в ОПАСНЫХ СИТУАЦИЯХ может использоваться множество методов. Эти методы используют различные подходы и могут быть полезны на различных этапах разработки программного обеспечения. Ни один из них не является единственно правильным методом. См. также приложение G для информации о некоторых доступных методах АНАЛИЗА РИСКА.

Анализ дерева неисправностей (FTA) является традиционным методом анализа «сверху вниз» (см. МЭК 61025 [3]), часто используемым для анализа начиная с МЕДИЦИНСКОГО ИЗДЕЛИЯ в целом. Анализ дерева неисправностей чаще всего используется для анализа причины ВРЕДА. Постулируется, что ВРЕД нанесен и используется Булева логика, чтобы идентифицировать события или обстоятельства, которые должны произойти для того, чтобы вред наступил. События или обстоятельства анализируются с все возрастающей детализацией, пока не будет достигнута точка, где могут быть определены одна или более мер по УПРАВЛЕНИЮ РИСКОМ, которые будут предотвращать ВРЕД. Анализ дерева неисправностей может использоваться, чтобы определять ПРОГРАММНЫЕ ЭЛЕМЕНТЫ, которые вовлечены в последовательность событий, которая приводит к ОПАСНОЙ СИТУАЦИИ.

Анализ видов и последствий отказов (FMEA-анализ) является методом анализа «снизу вверх» (см. МЭК 60812 [2]), который начинается с компонента или подсистемы (для программного обеспечения это ПРОГРАММНЫЙ ЭЛЕМЕНТ в МЭК 62304) и ставит вопрос: «Если этот элемент откажет, какими будут последствия?».

Ввиду трудности предвидения того, какие дефекты программного обеспечения будут присутствовать в каждом ПРОГРАММНОМ ЭЛЕМЕНТЕ, в стартовой точке для FMEA-анализа следует перечислить все СВЯЗАННЫЕ С БЕЗОПАСНОСТЬЮ требования каждого ПРОГРАММНОГО ЭЛЕМЕНТА и рассмотреть вопрос: «Если это требование не выполняется, каковы будут последствия?».

Это приводит к идентификации ЭЛЕМЕНТОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, отказ которых может вызвать вред, и идентификации того, какие типы отказа должны быть предотвращены.

При идентификации последовательностей или комбинаций событий, которые могут привести к ОПАСНОЙ СИТУАЦИИ, самым легким является необходимость сосредоточиться на программном обеспечении, непосредственно связанном с эксплуатационной пригодностью МЕДИЦИНСКОГО ИЗДЕЛИЯ (например, алгоритмы, которые вычисляет уровень содержания глюкозы в крови) и отдельных причинах связанных с ними ОПАСНОСТЕЙ. Также важно рассмотреть программные причины, которые могут приводить к трудно уловимым способам отказа и поэтому приводить к одной или более ОПАСНОСТЯМ МЕДИЦИНСКОГО ИЗДЕЛИЯ. См. приложение B для ознакомления с примерами программных причин.

Примечание — Отдельным видом причин являются дефекты программного обеспечения, функциональность которого явно связана с клинической функциональностью МЕДИЦИНСКОГО ИЗДЕЛИЯ и приводит к одной из ОПАСНОСТЕЙ, присущих изделию. В качестве примера можно привести дефект в алгоритме вычисления результатов теста.

Поскольку трудно точно предсказать, что может отказать в ПРОГРАММНОМ ЭЛЕМЕНТЕ, возможно идентифицировать категории дефектов для каждой из которых имеются хорошо известные меры по УПРАВЛЕНИЮ РИСКОМ. Например, повреждение данных — класс отказов, которые могут быть обнаружены и подтверждены использованием контрольной процедуры. См. приложение B с примерами программных причин и предлагаемыми способами их обработки. ИЗГОТОВИТЕЛИ должны поддерживать свои собственные перечни категорий дефектов программного обеспечения, относящегося к их собственной продукции.

4.4.3 Вероятность

АНОМАЛИИ программного обеспечения в отдельной ВЕРСИИ программного обеспечения будут присутствовать во всех его копиях. Однако вероятность АНОМАЛИИ, приводящей к отказу программного обеспечения, очень трудно определить из-за случайной природы входных данных для каждой отдельной копии программного обеспечения.

Не существует единого мнения в отношении метода определения вероятности возникновения отказа программного обеспечения. Когда программное обеспечение присутствует в последовательности событий, приводящих к ОПАСНОЙ СИТУАЦИИ, вероятность возникновения отказа программного обеспечения не может быть обоснована при определении РИСКА от ОПАСНОЙ СИТУАЦИИ. В таких случаях принцип наихудшего случая для определения вероятности считается более подходящим и вероятность возникновения отказа программного обеспечения следует принимать равной 1. Когда возможно определить вероятность для остальных событий в последовательности (как это может быть, если они не являются программным обеспечением), эта вероятность может быть применена для определения вероятности возникновения ОПАСНОЙ СИТУАЦИИ (P_1 на рисунке 1). Если это невозможно, вероятность возникновения ОПАСНОЙ СИТУАЦИИ следует принимать равной 1.

Определение вероятности того, что ОПАСНАЯ СИТУАЦИЯ приведет к причинению ВРЕДА (P_2 на рисунке 1), обычно требует знаний в области практической медицины, чтобы отличить ОПАСНЫЕ СИТУАЦИИ, в которых медицинская практика вероятнее всего предотвратит ВРЕД, от ОПАСНЫХ СИТУАЦИЙ, которые более вероятно причинят ВРЕД.

Примечание — P_1 — это вероятность возникновения ОПАСНОЙ СИТУАЦИИ. P_2 — это вероятность того, что ОПАСНАЯ СИТУАЦИЯ приведет к причинению ВРЕДА.

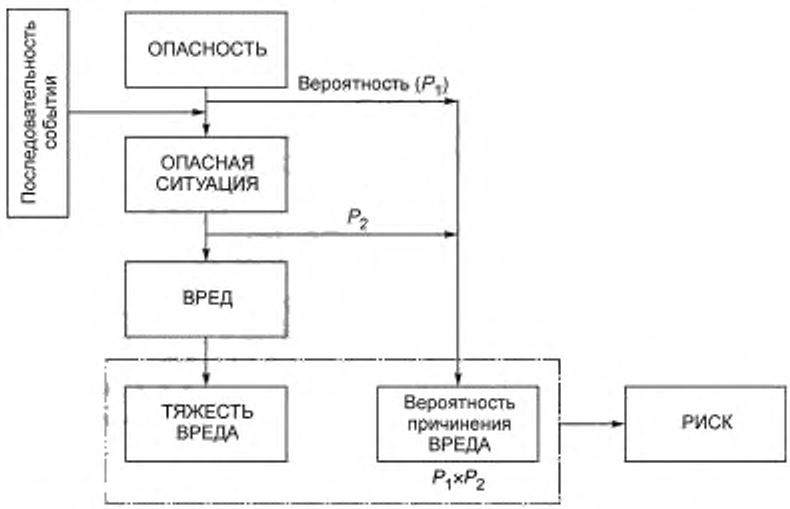

Рисунок 1 — Графическое представление взаимосвязи ОПАСНОСТИ, последовательности событий, ОПАСНОЙ СИТУАЦИИ и ВРЕДА (ИСО 14971, приложение Е)

Во многих случаях определение вероятности возникновения ВРЕДА может быть невозможным, и тогда РИСК должен оцениваться только на основе ТЯЖЕСТИ ВРЕДА. ОПРЕДЕЛЕНИЕ РИСКА в таких случаях должно фокусироваться на ТЯЖЕСТИ ВРЕДА, возникающего от ОПАСНОЙ СИТУАЦИИ.

Хотя может быть невозможным определить вероятность отказа программного обеспечения, очевидно, что многие меры по УПРАВЛЕНИЮ РИСКОМ снижают вероятность, что такой отказ может привести к ОПАСНОЙ СИТУАЦИИ. Рассмотрим, например, повреждение памяти, которое произошло из-за АНОМАЛИИ программного обеспечения. Контрольная сумма памяти может обнаружить отказ и снизить вероятность ОПАСНОЙ СИТУАЦИИ. Контрольная сумма не гарантирует, что любое возможное искажение будет обнаружено, хотя она обнаружит бóльшую часть таких искажений и этим снизит РИСК до допустимого уровня. Хотя вероятность ОПАСНОЙ СИТУАЦИИ не может быть определена как до, так и после применения контрольной суммы, можно утверждать, что вероятность ОПАСНОЙ СИТУАЦИИ после того,

как контрольная сумма поступит, ниже, чем вероятность перед выполнением контрольной суммы. Ответственностью ИЗГОТОВИТЕЛЯ является демонстрация того, что меры по управлению РИСКОМ эффективны в удовлетворении критериев допустимости для ОСТАТОЧНОГО РИСКА, которые были установлены в плане МЕНЕДЖМЕНТА РИСКА.

В итоге деятельность по ОПРЕДЕЛЕНИЮ РИСКА программного обеспечения в первую очередь должна быть направлена на ТЯЖЕСТЬ и относительную вероятность ВРЕДА, если отказ может произойти, а не на попытки определить вероятность каждого возможного отказа программного обеспечения.

Примечание — Это обеспечивает различие между ОПАСНОСТЯМИ, приводящими к одинаковой ТЯЖЕСТИ, позволяющее обращать больше внимания на те ОПАСНОСТИ, у которых более высокая вероятность фактического ВРЕДА.

4.4.4 ТЯЖЕСТЬ

Определение ТЯЖЕСТИ для связанного с программным обеспечением РИСКА оказывает влияние на использующийся ПРОЦЕСС разработки программного обеспечения. Согласно МЭК 62304 степень тщательности этого ПРОЦЕССА зависит от ТЯЖЕСТИ ВРЕДА, который может причинить программное обеспечение.

Поскольку за ИЗГОТОВИТЕЛЕМ остается определение уровня ТЯЖЕСТИ с целью ОЦЕНИВАНИЯ РИСКА согласно ИСО 14971, полезно определить эти уровни ТЯЖЕСТИ так, чтобы существовала взаимосвязь с классификацией программного обеспечения по БЕЗОПАСНОСТИ в соответствии с МЭК 62304. В противном случае может быть необходимо классифицировать ТЯЖЕСТЬ дважды: один раз для ОЦЕНИВАНИЯ РИСКА, что необходимо для общего ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА, а второй раз для установления класса БЕЗОПАСНОСТИ программного обеспечения согласно МЭК 62304.

5 ОЦЕНИВАНИЕ РИСКА

Текст ИСО 14971

5 ОЦЕНИВАНИЕ РИСКА

Для каждой идентифицированной ОПАСНОЙ СИТУАЦИИ ИЗГОТОВИТЕЛЬ должен принять решение о необходимости уменьшения РИСКА с учетом критериев, установленных в плане МЕНЕДЖМЕНТА РИСКА. Если в уменьшении РИСКА нет необходимости, то требования 6.2—6.6 для данной ОПАСНОЙ СИТУАЦИИ не применяют (т. е. переходят к 6.7). Результаты такого ОЦЕНИВАНИЯ РИСКА должны быть зарегистрированы в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА.

Примечания

1 Руководящие указания по принятию решения о допустимости риска приведены в D.4.

2 Применение соответствующих стандартов как один из критериев проектирования МЕДИЦИНСКИХ ИЗДЕЛИЙ может являться частью деятельности по УПРАВЛЕНИЮ РИСКОМ, что отвечает требованиям 6.3—6.6.

Соответствие требованиям данного раздела проверяют путем контроля файла МЕНЕДЖМЕНТА РИСКА.

Как описано в 4.4.3, сложно определить вероятность отказов программного обеспечения. Когда это приводит к невозможности определить вероятность ВРЕДА, тогда РИСК должен оцениваться только на основе ТЯЖЕСТИ ВРЕДА.

6 УПРАВЛЕНИЕ РИСКОМ

6.1 Уменьшение РИСКА

Текст ИСО 14971

6 УПРАВЛЕНИЕ РИСКОМ

6.1 Уменьшение РИСКА

При необходимости уменьшения РИСКА следует осуществлять деятельность по УПРАВЛЕНИЮ РИСКОМ, как описано в 6.2—6.7.

Программные аспекты уменьшения РИСКА рассматриваются в пунктах с 6.2 по 6.7.

6.2 Анализ возможностей УПРАВЛЕНИЯ РИСКОМ

Текст ИСО 14971

6.2 Анализ возможностей УПРАВЛЕНИЯ РИСКОМ

ИЗГОТОВИТЕЛЬ должен идентифицировать меру(ы) по УПРАВЛЕНИЮ РИСКОМ, необходимую(ые) для уменьшения РИСКА(ОВ) до допустимого уровня.

ИЗГОТОВИТЕЛЬ должен применять один или несколько способов УПРАВЛЕНИЯ РИСКОМ, перечисленных ниже в порядке приоритетов:

a) внутреннюю БЕЗОПАСНОСТЬ, обеспечиваемую проектом и конструкцией;

b) защитные меры, предусмотренные в самом медицинском изделии или в ПРОЦЕССЕ его изготовления;

c) информацию по БЕЗОПАСНОСТИ.

Примечания

1 При применении способов, приведенных в перечислениях b) и c), ИЗГОТОВИТЕЛЬ может рассмотреть обоснованные и практически осуществимые меры по УПРАВЛЕНИЮ РИСКОМ и выбрать способ, обеспечивающий необходимое уменьшение РИСКА, прежде чем будет установлено, является ли РИСК допустимым.

2 Меры по УПРАВЛЕНИЮ РИСКОМ могут уменьшить ТЯЖЕСТЬ ВРЕДА или вероятность причинения ВРЕДА, или и то и другое вместе.

3 Многие стандарты рассматривают вопросы внутренней БЕЗОПАСНОСТИ, обеспечиваемой проектом и конструкцией, защитные меры и информацию по БЕЗОПАСНОСТИ МЕДИЦИНСКИХ ИЗДЕЛИЙ. Кроме того, другие стандарты на МЕДИЦИНСКИЕ ИЗДЕЛИЯ рассматривают те элементы ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА (электромагнитную совместимость, эксплуатационную пригодность, биологическую совместимость), которые являются общими для МЕДИЦИНСКИХ ИЗДЕЛИЙ. При анализе возможностей УПРАВЛЕНИЯ РИСКОМ следует применять соответствующие стандарты.

4 В отношении РИСКОВ, для которых не может быть определена вероятность причинения ВРЕДА, см. D.3.2.3.

5 Руководящие указания в отношении информации по БЕЗОПАСНОСТИ приведены в приложении J.

Выбранные меры по УПРАВЛЕНИЮ РИСКОМ должны быть зарегистрированы в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА.

Если в ПРОЦЕССЕ анализа возможностей УПРАВЛЕНИЯ РИСКОМ ИЗГОТОВИТЕЛЬ устанавливает, что требуемое уменьшение РИСКА практически неосуществимо, то он должен выполнить анализ соотношения РИСК/польза для ОСТАТОЧНОГО РИСКА (переход к 6.5).

Соответствие требованиям данного подраздела проверяют путем контроля файла МЕНЕДЖМЕНТА РИСКА.

6.2.1 Выбор возможностей по УПРАВЛЕНИЮ РИСКОМ для сложных СИСТЕМ

6.2.1.1 Общие положения

В сложной СИСТЕМЕ может быть много последовательностей событий, которые способны привести к ОПАСНЫМ СИТУАЦИЯМ. Применить меры по УПРАВЛЕНИЮ РИСКОМ к каждому событию такой последовательности может быть не осуществимо. Достаточно применять меры по УПРАВЛЕНИЮ РИСКОМ к выбранным событиям, чтобы сократить общую вероятность ВРЕДА до допустимого уровня.

В следующих трех подпунктах дан обзор того, как три типа мер по УПРАВЛЕНИЮ РИСКОМ могут быть применены в программном обеспечении. Кроме того, обсуждается, какие события нуждаются в применении мер по УПРАВЛЕНИЮ РИСКОМ (см. 6.2.1.5).

6.2.1.2 Внутренняя БЕЗОПАСНОСТЬ, обеспечиваемая проектом и конструкцией

Внутренняя БЕЗОПАСНОСТЬ, обеспечиваемая проектом и конструкцией, обычно достигается устранением небезопасных особенностей МЕДИЦИНСКОГО ИЗДЕЛИЯ или изменением конструкции для реализации функции более безопасным способом, то есть таким образом, чтобы избежать или минимизировать ОПАСНЫЕ СИТУАЦИИ. Это часто влечет за собой упрощение конструкции, облегчающее реализацию проекта и упрощающее эксплуатацию изделия для оператора или пользователя.

Особенно это относится к возможностям, осуществляемым в программном обеспечении. В программных СИСТЕМАХ существует соблазн включать все возможные пожелания потребителя без ограничений. Это может приводить к чрезмерному множеству способов, которыми могут взаимодействовать компоненты

программного обеспечения, создавая тем самым неожиданные ОПАСНЫЕ СИТУАЦИИ. Этого можно избежать путем применения МЕНЕДЖМЕНТА РИСКА на ранних стадиях разработки МЕДИЦИНСКОГО ИЗДЕЛИЯ и его программного обеспечения, одновременно удовлетворяя требования большинства потребителей.

В большинстве случаев внутренняя БЕЗОПАСНОСТЬ, обеспечиваемая проектом и конструкцией, примененная к программному обеспечению, будет включать в себя:

- устранение характеристик, которые не являются необходимыми;
- изменение АРХИТЕКТУРЫ программного обеспечения с целью избегания последовательностей событий, которые приводят к ОПАСНЫМ СИТУАЦИЯМ;
- упрощение интерфейса пользователя для сокращения вероятности ошибок человека при использовании;
- установление правил проектирования программного обеспечения, чтобы избежать АНОМАЛИЙ программного обеспечения.

Примером к предыдущему перечислению может служить:

- использование только статического распределения памяти с целью избегания АНОМАЛИЙ программного обеспечения, связанных с динамическим распределением памяти;
- использование строгой (защищенной) ВЕРСИИ языка программирования, чтобы избегать структур, которые вероятно могут привести к ошибкам программирования.

6.2.1.3 Защитные меры

Защитные меры в отношении МЕДИЦИНСКОГО ИЗДЕЛИЯ, использующего программное обеспечение, могут быть осуществлены либо в аппаратных средствах, либо в программном обеспечении. Проект защитных мер должен продемонстрировать, что защитная мера независима от функции, к которой она применяется. Этого сравнительно легко достичь, если программные защитные меры применяются к аппаратным средствам или наоборот.

При выборе защитных мер, которые реализованы в программном обеспечении и применены к программному обеспечению, важно избегать вероятности множества отказов, происходящих по одной причине. Если защитные меры обнаруживают и (или) предотвращают ОПАСНУЮ СИТУАЦИЮ, ИЗГОТОВИТЕЛЬ должен продемонстрировать достаточное разделение между защитной мерой и текущим исполнением программного обеспечения, которое поддерживает основные эксплуатационные характеристики.

Например, программное обеспечение, обуславливающее лечение пациента, может быть задействовано на одном процессоре, в то время как программное обеспечение, которое реализует программные защитные меры, выполняется на другом.

6.2.1.4 Информация по БЕЗОПАСНОСТИ

Использование программного обеспечения в МЕДИЦИНСКОМ ИЗДЕЛИИ чаще всего приводит к более сложному поведению с точки зрения пользователя. Чаще всего это приводит к возросшему доверию к информации по БЕЗОПАСНОСТИ, варьирующейся от простых предупреждений на экране до сложных пользовательских инструкций и необходимости в обучении. Объем и сложность такого письменного материала могут быть уменьшены путем создания лучшей конструкции пользовательского интерфейса (см. МЭК 62366 [5]).

6.2.1.5 Какие события нуждаются в разработке мер по УПРАВЛЕНИЮ РИСКОМ?

Многие последовательности событий способны приводить к ОПАСНЫМ СИТУАЦИЯМ. Применить меры по УПРАВЛЕНИЮ РИСКОМ к каждому событию такой последовательности может быть неосуществимо. Достаточно применять меры по УПРАВЛЕНИЮ РИСКОМ к тщательно выбранным событиям, чтобы сократить общую вероятность ВРЕДА до допустимого уровня.

При решении, какие события должны быть выбраны, предотвращена или уменьшена вероятность их возникновения, полезно составить карту последовательностей событий, которые могут привести к ОПАСНЫМ СИТУАЦИЯМ. Поскольку анализ дерева отказов не показывает последовательности событий, он может быть использован для идентификации таких последовательностей. Правильное действие меры по УПРАВЛЕНИЮ РИСКОМ будет проявляться как входные данные «неправильно» на входе логического элемента «И», приводящие к предотвращению ВРЕДА независимо от других входных данных на входе «И». Рисунок 2 показывает часть FTA (анализ дерева неисправностей) диаграммы, на которой неверные выходные данные программного обеспечения (сам результат последовательности событий) предотвращают получение травмы пациентом посредством меры по УПРАВЛЕНИЮ РИСКОМ, которая разработана, чтобы обнаруживать небезопасные состояния и предотвращать вредное воздействие таких

выходных данных (например, путем прекращения работы). Неверные данные на выходе программного обеспечения могут нанести ущерб пациенту, только если откажет мера по УПРАВЛЕНИЮ РИСКОМ. Следует заметить, что последовательность событий, приводящая к неправильным данным на выходе программного обеспечения, не нуждается в детальном изучении, чтобы убедиться, что она не может привести к нанесению ущерба пациенту.

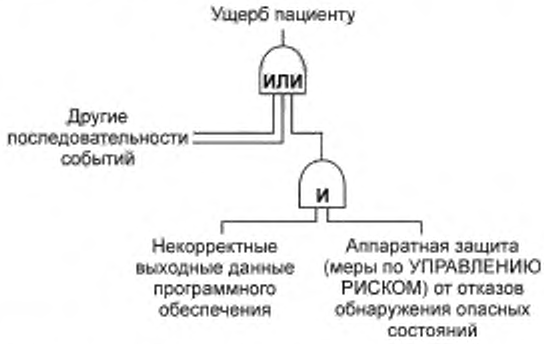

Рисунок 2 — FTA (анализ дерева неисправностей) показывает меру по УПРАВЛЕНИЮ РИСКОМ, которая препятствует неправильным выходным данным программного обеспечения причинить ВРЕД

Очевидные точки, в которых могут быть применены меры по УПРАВЛЕНИЮ РИСКОМ, включают:

- входные данные в ПРОГРАММНУЮ СИСТЕМУ в целом;
- выходы из ПРОГРАММНОЙ СИСТЕМЫ в целом;
- внутренние интерфейсы между программными модулями.

Мера по УПРАВЛЕНИЮ РИСКОМ, которая ограничивает диапазон входных данных в программное обеспечение, может предотвратить появление небезопасных выходных данных (результатов). Менее очевидно, что она может уменьшить вероятность появления входных данных, приводящих к причинению ВРЕДА АНОМАЛИЯМИ программного обеспечения, поскольку это снижает вероятность функционирования программного обеспечения непредсказуемым образом, что не может быть проверено испытанием (см. 4.4.3)

Ограничение диапазона входных данных программного обеспечения может быть выполнено с использованием программных или аппаратных мер по УПРАВЛЕНИЮ РИСКОМ. Например:

- программные меры по УПРАВЛЕНИЮ РИСКОМ могут проверять входные данные и отклонять небезопасные или противоречивые величины;
- аппаратные меры по УПРАВЛЕНИЮ РИСКОМ могут состоять из «автоматической блокировки», чтобы предотвращать введение данных несанкционированными людьми.

Меры по УПРАВЛЕНИЮ РИСКОМ, реализованные на выходе МЕДИЦИНСКОГО ИЗДЕЛИЯ или его программного обеспечения, могут верифицировать, что выходные данные программного обеспечения непротиворечивы и находятся в пределах безопасного диапазона и, если это так, могут предотвращать ВРЕД. Это может достигаться, например, путем:

- программной меры по УПРАВЛЕНИЮ РИСКОМ, которая проверяет диапазон выходных данных и препятствует их выходу за пределы безопасных значений;
- аппаратной меры по УПРАВЛЕНИЮ РИСКОМ, которая ограничивает энергию, поступающую к пациенту;
- мерой по УПРАВЛЕНИЮ РИСКОМ, состоящей из комбинации предупреждающей маркировки и аппаратного выключателя. Реализация этой меры по УПРАВЛЕНИЮ РИСКОМ предполагает, что компетентный оператор способен обнаружить ОПАСНУЮ СИТУАЦИЮ.

Кроме мер по УПРАВЛЕНИЮ РИСКОМ, применяемых к входным и выходным данным МЕДИЦИНСКОГО ИЗДЕЛИЯ или его программного обеспечения, меры по УПРАВЛЕНИЮ РИСКОМ могут также применяться к входным и выходным данным программных компонентов. Это позволяет проверять входные и выходные данные более мелких частей программного обеспечения и предотвращать ВРЕД.

Может быть неосуществимым указание единственного диапазона для параметра, внутри которого МЕДИЦИНСКОЕ ИЗДЕЛИЕ работает безопасно. Тем не менее, может быть осуществимым указание «безопасного рабочего диапазона», другими словами — комбинации параметров, формирующих границы,

внутри которых МЕДИЦИНСКОЕ ИЗДЕЛИЕ работает безопасно. Программное обеспечение может быть использовано для оценки того, что МЕДИЦИНСКОЕ ИЗДЕЛИЕ функционирует в пределах безопасного рабочего диапазона. Например, программное обеспечение может отслеживать время воздействия и температуру определенного участка тела для предотвращения возможности ожога у пациента.

Бывают случаи, когда безопасный диапазон для входных и выходных данных известен врачу, но его нельзя предвидеть в конструкции МЕДИЦИНСКОГО ИЗДЕЛИЯ. В таких случаях меры по УПРАВЛЕНИЮ РИСКОМ могут применяться для обеспечения выполнения МЕДИЦИНСКИМ ИЗДЕЛИЕМ именно того, что определено врачом. Аппаратные и программные меры по УПРАВЛЕНИЮ РИСКОМ могут использоваться для обнаружения несогласованности между входными и выходными данными программного обеспечения.

Например, врач может назначить лечение с использованием конкретного МЕДИЦИНСКОГО ИЗДЕЛИЯ, предусматривающее различные режимы его функционирования для различных пациентов. ОПАСНАЯ СИТУАЦИЯ не может быть обнаружена только путем анализа входных или выходных значений. Тем не менее, программная мера по УПРАВЛЕНИЮ РИСКОМ может быть применена для обеспечения точного соответствия входных и выходных значений МЕДИЦИНСКОГО ИЗДЕЛИЯ (соответствие назначенному лечению).

6.2.2 Методы УПРАВЛЕНИЯ РИСКОМ

6.2.2.1 Обзор

В целях результативного осуществления мер по УПРАВЛЕНИЮ РИСКОМ программного обеспечения понимание ПРОЦЕССА разработки продукции и ЖИЗНЕННОГО ЦИКЛА программного обеспечения следует тщательно продумывать. Некоторые типы мер по УПРАВЛЕНИЮ РИСКОМ очень легко осуществить на ранних этапах проектирования и невозможно или очень дорого — на последующих стадиях разработки. Если программное обеспечение не рассматривают с точки зрения МЕНЕДЖМЕНТА РИСКА уже на ранних стадиях ПРОЦЕССА разработки продукции, то для обеспечения БЕЗОПАСНОСТИ МЕДИЦИНСКОГО ИЗДЕЛИЯ могут быть приняты такие решения относительно аппаратных средств, которые будут чрезмерно зависеть от правильной работы программного обеспечения.

Это может быть полезно для разделения ПРОГРАММНЫХ ЭЛЕМЕНТОВ и присвоения классов БЕЗОПАСНОСТИ программного обеспечения ПРОГРАММНЫМ ЭЛЕМЕНТАМ, чтобы отличать критичные ПРОГРАММНЫЕ ЭЛЕМЕНТЫ (например, те, которые могут привести к смерти, если они являются дефектными) от тех, которые не влияют на БЕЗОПАСНОСТЬ. См. п. 4.3 МЭК 62304 для информации о классификации программного обеспечения по БЕЗОПАСНОСТИ.

Назначение классов БЕЗОПАСНОСТИ программного обеспечения может служить основой для обеспечения большего внимания к ВЕРИФИКАЦИИ и управлению конфигурацией для наиболее критичных ПРОГРАММНЫХ ЭЛЕМЕНТОВ. Если это будет сделано, побочные эффекты нужно тщательно рассмотреть, и менее критичные ПРОГРАММНЫЕ ЭЛЕМЕНТЫ должны быть оценены так же, как и более критичные, на которые они могут повлиять. Следует отметить, что МЭК 62304 допускает различные методы, которые будут использоваться в пределах деятельности или задачи (См. МЭК 62304, п. 5.1.4, в соответствии с которым определяются методы). ИЗГОТОВИТЕЛЬ может принять решение создать схему дифференцирования ПРОГРАММНЫХ ЭЛЕМЕНТОВ, которым по классификации БЕЗОПАСНОСТИ программного обеспечения присвоен класс С. Например, ИЗГОТОВИТЕЛЬ может использовать более формальный метод ВЕРИФИКАЦИИ (то есть тщательная проверка кода вместо обзора кода) для ПРОГРАММНЫХ ЭЛЕМЕНТОВ с более высокой сложностью.

ПРОГРАММНЫЕ ЭЛЕМЕНТЫ могут быть первоначально классифицированы как СВЯЗАННЫЕ С БЕЗОПАСНОСТЬЮ, но затем, при определенных мерах по УПРАВЛЕНИЮ РИСКОМ или проектных решениях, они могут рассматриваться как менее критичные. Правильно выполненный МЕНЕДЖМЕНТ РИСКА может приводить к уменьшению РИСКОВ, СВЯЗАННЫХ С БЕЗОПАСНОСТЬЮ программного обеспечения, до наименьших значений путем их изоляции и разработки изначально безопасной конструкции.

Обеспечение БЕЗОПАСНОСТИ программного обеспечения требует различных мероприятий по всему ЖИЗНЕННОМУ ЦИКЛУ разработки продукции. Такие методы определения надежности как официальные методы для анализа отказов не включают в себя методы полного МЕНЕДЖМЕНТА РИСКА. Также важно признать, что надежность и безопасность часто являются взаимосвязанными, но не идентичными понятиями. ПРОЦЕСС ЖИЗНЕННОГО ЦИКЛА, который сосредотачивается на надежности, может не достичь должной БЕЗОПАСНОСТИ.

Некоторые отдельные меры по УПРАВЛЕНИЮ РИСКОМ, а также руководство по определению причин отдельных РИСКОВ описаны более подробно в пп. 6.2.2.2, 6.2.2.3, 6.2.2.4, 6.2.2.5 и 6.2.2.6.

6.2.2.2 Меры по УПРАВЛЕНИЮ РИСКОМ и проектирование структуры программного обеспечения

6.2.2.2.1 Краткий обзор

АРХИТЕКТУРА программного обеспечения должна описывать особенности программного обеспечения, используемые для управления РИСКОМ путем изначально безопасной конструкции, а также программных механизмов реализации защитных мер для снижения РИСКА.

6.2.2.2.2 Изначально безопасная конструкция на основе особенностей АРХИТЕКТУРЫ

ОПАСНОСТИ, связанной с функцией программного управления, можно избежать, например, используя аппаратные средства для реализации этой функции. Точно также ОПАСНОСТИ, связанной с функцией аппаратных средств (изнашивание, усталость), можно избежать, используя программное обеспечение.

Иногда ОПАСНОСТЕЙ можно полностью избежать, используя проектное решение высокого уровня. Примером в отношении аппаратных средств, является решение использовать батарейки в качестве источника энергии вместо переменного тока, которое может устранить РИСК смерти от электрического тока. Подобным образом целый класс ошибок программирования, которые могут привести к ОПАСНОСТИ, может быть устранен на основе проектных решений высокого уровня. Например, утечек памяти можно избежать, используя только статические структуры данных.

Особой проблемой в СИСТЕМАХ, использующих программное обеспечение, является представление о том, что нет предела той степени, до которой различное программное обеспечение может совместно использовать общую физическую инфраструктуру. Это представление ложно.

Обычным правилом проектирования СИСТЕМЫ является то, что, когда требуется, система должна включать достаточные ресурсы для выполнения всех необходимых ЗАДАЧ. Это правило должно применяться к программному обеспечению и к оборудованию. Если ПРОГРАММНЫЙ ЭЛЕМЕНТ играет роль в обеспечении БЕЗОПАСНОСТИ, то ОЦЕНКА РИСКА должна охватывать следующие вопросы:

- может ли связанный с БЕЗОПАСНОСТЬЮ ПРОГРАММНЫЙ ЭЛЕМЕНТ получить доступ к своему процессору, когда необходимо?

- может ли связанному с БЕЗОПАСНОСТЬЮ ПРОГРАММНОМУ ЭЛЕМЕНТУ быть обеспечено достаточно времени процессора, чтобы завершить его ЗАДАЧУ до того, как небезопасное состояние перерастет в неблагоприятное событие?

- можно ли продемонстрировать, что никакой другой ПРОГРАММНЫЙ ЭЛЕМЕНТ не может прервать или вмешаться в работу СВЯЗАННОГО С БЕЗОПАСНОСТЬЮ ПРОГРАММНОГО ЭЛЕМЕНТА?

Если связанное с БЕЗОПАСНОСТЬЮ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ должно использовать процессор совместно с не СВЯЗАННЫМ С БЕЗОПАСНОСТЬЮ ПРОГРАММНЫМ ОБЕСПЕЧЕНИЕМ, то вышеупомянутые вопросы особенно важны, поскольку функции обеспечения БЕЗОПАСНОСТИ будут конкурировать с другими за ресурсы (см. п. 6.2.2.2.4 о разделении).

Методы разработки должны быть выбраны так, чтобы сделать все вышеупомянутые проблемы видимыми проектировщику. Например, недостаточно создать СВЯЗАННЫЙ С БЕЗОПАСНОСТЬЮ ПРОГРАММНЫЙ ЭЛЕМЕНТ, как ПРОЦЕСС, который запустится когда «все хорошо» и операционная система найдет для него время. Метод разработки должен намеренно поддерживать надлежащее проектирование планирования, приоритетов и сроков.

6.2.2.2.3 Отказоустойчивая АРХИТЕКТУРА

Для обеспечения БЕЗОПАСНОСТИ пациента или пользователя наличие некоторых функций МЕДИЦИНСКОГО ИЗДЕЛИЯ может носить обязательный характер. Такие функции могут включать медицинские процедуры, которые не должны быть прерваны или отсрочены, а также функции, которые осуществляют защитные меры по УПРАВЛЕНИЮ РИСКОМ.

Отказоустойчивая конструкция — это очень распространенный подход к улучшению надежности МЕДИЦИНСКОГО ИЗДЕЛИЯ (ссылки для специалистов-практиков по разработке программного обеспечения включают Паллума [7] и Банатре [8]). Целью отказоустойчивой конструкции является обеспечение того, что СВЯЗАННЫЕ С БЕЗОПАСНОСТЬЮ функции продолжают работать при наличии отказов компонентов, включая АНОМАЛИИ программного обеспечения.

Отказоустойчивая конструкция обычно обладает некоторой ИЗБЫТОЧНОСТЬЮ. Это может быть простым дублированием жизненно важного компонента для обеспечения продолжения функционирования, если один из компонентов откажет, или может состоять из дополнительных компонентов, которые обнаруживают отказ и переводят выполнение операции на альтернативный режим работы, возможно и с ограниченной функциональностью.

Отказоустойчивость может использоваться для поддержания существенной функции в случае отказа программного обеспечения. Тогда простая ИЗБЫТОЧНОСТЬ, использующая множество копий одной и той же программы, скорее всего недостаточна, поскольку тот же самый дефект будет присутствовать в каждой копии программы.

В таких случаях требуется РАЗНООБРАЗИЕ. Например, дополнительное программное обеспечение может использоваться для обнаружения ошибки программного обеспечения и выполнения программы восстановления. Дополнительное программное обеспечение не должно разделять любые функции с тем программным обеспечением, которое оно контролирует. Таким образом, устраняется возможность того, что один дефект вызывает отказ обеих программ.

В наиболее важных случаях два или несколько ПРОГРАММНЫХ ЭЛЕМЕНТОВ могут выполнять одну и ту же функцию, но они могут быть независимо разработаны и реализованы начиная с общей спецификации. Это называется «программирование РАЗНООБРАЗИЯ». Следует отметить, что есть тенденция совершать одни и те же ошибки разными инженерами-разработчиками. Эта тенденция делает РАЗНООБРАЗИЕ недействительным. Отметим также, что общая спецификация может содержать неправильные требования. Наконец, некоторые методы, такие как голосование, должны использоваться для обеспечения того, чтобы работающее со сбоями программное обеспечение не вызывало никаких эффектов. Как минимум три различных программных элемента были бы необходимы для реализации схемы голосования.

Используя ИЗБЫТОЧНОСТЬ с РАЗНООБРАЗИЕМ или без него, для обеспечения отказоустойчивости, важно дать понять пользователю, что произошел отказ. В противном случае может показаться, что отказоустойчивое МЕДИЦИНСКОЕ ИЗДЕЛИЕ работает безопасно, когда на самом деле, изделие работает с уменьшенной БЕЗОПАСНОСТЬЮ.

6.2.2.2.4 Разделение, уменьшающее РИСК, происходящий от программных причин

Возможно, что дефекты программного обеспечения могут приводить к ошибкам в не связанном с ним программном обеспечении, выполняемом на тех же аппаратных средствах. ИЗГОТОВИТЕЛЬ должен выбирать методы разделения СВЯЗАННЫХ С БЕЗОПАСНОСТЬЮ ПРОГРАММНЫХ ЭЛЕМЕНТОВ от не СВЯЗАННЫХ С БЕЗОПАСНОСТЬЮ таким образом, чтобы последние не могли вмешиваться в работу СВЯЗАННЫХ С БЕЗОПАСНОСТЬЮ ПРОГРАММНЫХ ЭЛЕМЕНТОВ (см. МЭК 62304, п. 5.3.5). ИЗГОТОВИТЕЛЬ должен продемонстрировать, что такое разделение эффективно. Это включает в себя демонстрацию подходящего использования ресурсов (физических или временны́х) ПРОГРАММНЫХ ЭЛЕМЕНТОВ для избежания непреднамеренной конкуренции между элементами.

Для эффективного разделения ПРОГРАММНЫХ ЭЛЕМЕНТОВ следует обратить внимание на описанные ниже возможные способы, при помощи которых ПРОГРАММНЫЕ ЭЛЕМЕНТЫ могут подвергаться неожиданному взаимодействию.

ПРОГРАММНЫЕ ЭЛЕМЕНТЫ могут взаимодействовать непредусмотренными способами, когда они конкурируют за время на общих аппаратных средствах (например, процессоры, устройства хранения данных и другие устройства входа/выхода). Это может препятствовать запуску ПРОГРАММНОГО ЭЛЕМЕНТА в назначенное время. Предоставление достаточных аппаратных средств является характерной архитектурной чертой или особенностью проекта (см. 6.2.2.2.2), которая должна быть установлена в соответствующей спецификации и подлежать планированию для обеспечения достаточного времени, насколько это возможно, чтобы запустить ПРОГРАММНЫЙ ЭЛЕМЕНТ в случае необходимости.

ПРОГРАММНЫЕ ЭЛЕМЕНТЫ могут сосуществовать в одной и той же памяти. Такая ситуация может привести к тому, что один ПРОГРАММНЫЙ ЭЛЕМЕНТ неожиданно изменяет данные, принадлежащие другому ПРОГРАММНОМУ ЭЛЕМЕНТУ. В чрезвычайных случаях один ПРОГРАММНЫЙ ЭЛЕМЕНТ может случайно изменить кодирование другого ПРОГРАММНОГО ЭЛЕМЕНТА. Многие процессоры и операционные системы предлагают поддерживаемые аппаратными средствами методы разделения использования памяти. Там, где такие методы существуют, они должны использоваться всегда. Большинство таких методов защиты будут принимать меры, против непреднамеренного взаимодействия, даже когда есть дефект в одном из ПРОГРАММНЫХ ЭЛЕМЕНТОВ.

ПРОГРАММНЫЕ ЭЛЕМЕНТЫ могут также взаимодействовать непреднамеренным способом, когда они делят переменные, включая глобальные переменные, переменные окружающей среды и параметры операционной системы. Такая ситуация может привести к непреднамеренной коммуникации между ПРОГРАММНЫМИ ЭЛЕМЕНТАМИ, если есть дефект в одном из ПРОГРАММНЫХ ЭЛЕМЕНТОВ.

Совместное использование переменных разными ПРОГРАММНЫМИ ЭЛЕМЕНТАМИ должно быть минимизировано. Если это необходимо, правила должны быть доведены до сведения всех инженеров, чтобы убедиться, что совместно используемые переменные изменяются только небольшим количеством определенных ПРОГРАММНЫХ ЭЛЕМЕНТОВ и что другие ПРОГРАММНЫЕ ЭЛЕМЕНТЫ только читают общие переменные, не изменяя их.

Самая строгая форма разделения состоит в выполнении ПРОГРАММНЫХ ЭЛЕМЕНТОВ, которые не должны взаимодействовать на отдельных процессорах. Однако проработанный архитектурный дизайн, как рекомендуется выше, может обеспечить необходимую степень разделения на одном процессоре.

Испытания СИСТЕМ в лабораторных условиях могут определить достаточные физические и временные ресурсы в отношении проводимых испытуемых случаев (условий), в то время как загрузка приложений или исполнение программно-аппаратной среды (другие ПРОЦЕССЫ, выполняемые в том же самом блоке) в реальных практических условиях вызывают отказ программного обеспечения, который приводит к ВРЕДУ.

С другой стороны, когда конкретные испытания в лаборатории действительно показывают, что существует низкая производительность и принимаются недопустимые меры для ускорения работы программного обеспечения, то эти меры, возможно, могут привести к нарушению конструкции и добавляют другие РИСКИ посредством непредвиденных побочных эффектов.

Эффективное разделение должно демонстрировать, что при нормальной работе:

a) искажение потока данных предотвращено: не СВЯЗАННЫЕ С БЕЗОПАСНОСТЬЮ ПРОГРАММНЫЕ ЭЛЕМЕНТЫ не могут изменять СВЯЗАННЫЕ С БЕЗОПАСНОСТЬЮ данные;

b) предотвращено изменение управляющего потока данных:

1) СВЯЗАННЫЕ С БЕЗОПАСНОСТЬЮ функции всегда могут выполняться в нужное время без воздействия на них не СВЯЗАННЫХ С БЕЗОПАСНОСТЬЮ ПРОГРАММНЫХ ЭЛЕМЕНТОВ;

2) не СВЯЗАННЫЕ С БЕЗОПАСНОСТЬЮ ПРОГРАММНЫЕ ЭЛЕМЕНТЫ не могут изменять СВЯЗАННЫЕ С БЕЗОПАСНОСТЬЮ ПРОГРАММНЫЕ ЭЛЕМЕНТЫ;

c) предотвращено искажение программной средой выполнения: частей ПРОГРАММНОЙ СИСТЕМЫ, использующей как СВЯЗАННЫЕ С БЕЗОПАСНОСТЬЮ, так и не СВЯЗАННЫЕ С БЕЗОПАСНОСТЬЮ ЭЛЕМЕНТЫ (например, регистры процессора, регистры изделия и привилегии доступа к памяти).

События, которые вызывают любое из упомянутых выше нарушений, например, отказ аппаратных средств, должны быть обнаружены и должны понуждать СИСТЕМУ к необходимым действиям для обеспечения сохранения БЕЗОПАСНОСТИ.

6.2.2.3 Подробности о защитных мерах

Во многих случаях нецелесообразно избегать всех ОПАСНОСТЕЙ путем создания изначально безопасной конструкции или обеспечивать отказоустойчивость в отношении всех возможных отказов. Следующим наилучшим подходом к управлению потенциальной ОПАСНОСТЬЮ в таких случаях являются защитные меры. Как правило, эти меры функционируют путем обнаружения потенциально ОПАСНОЙ СИТУАЦИИ. Далее эти меры либо инициируют вмешательство для смягчения последствий автоматически, либо генерируют тревожное сообщение для возможности вмешательства пользователя.

Например, терапевтическая рентгеновская СИСТЕМА может иметь блокирующую СИСТЕМУ, использующую программную логику или аппаратные средства которые блокируют лучевой генератор, если какая-нибудь из соответствующих дверей открыта. Блокирующая функция в терапии не играет никакой роли. Ее единственная цель состоит в том, чтобы смягчить ВРЕД от непреднамеренного облучения.

В некоторых случаях (например, когда потеря функциональности МЕДИЦИНСКОГО ИЗДЕЛИЯ не означает ОПАСНОСТИ) БЕЗОПАСНОСТЬ может достигаться за счет назначения требуемого действия. Например, отказ лабораторного анализатора крови может в некоторых случаях не быть опасным из-за отсутствия результата, но при этом существует возможность получения неверного результата. В этом примере отключение анализатора вместо продолжения работы, когда защитные программные проверки показывают неожиданные ошибки, уменьшает РИСКИ. В отказоустойчивой АРХИТЕКТУРЕ отказ СИСТЕМЫ или ее компонента или другое опасное условие могут привести к потере функции, но таким способом, который сохраняет БЕЗОПАСНОСТЬ операторов и пациентов. В случае сбоев операционной СИСТЕМЫ она может продолжать работать безопасно, но с ухудшенными эксплуатационными характеристиками (например, со снижением производительности или времени отклика).

6.2.2.4 Незамедлительное оповещение и предотвращение ОПАСНЫХ СИТУАЦИЙ

Важным классом мер по УПРАВЛЕНИЮ РИСКОМ является тот, который увеличивает вероятность предотвращения ОПАСНОЙ СИТУАЦИИ.

В дополнение к предотвращению ВРЕДА следует уделять внимание информированию пользователя об обнаруженном состоянии. При отсутствии такого информирования существует возможность причинения ВРЕДА, если последует отказ меры по УПРАВЛЕНИЮ РИСКОМ.

Следует также уделить внимание частоте, с которой будут выполняться меры по управлению РИСКОМ. Программные меры по УПРАВЛЕНИЮ РИСКОМ должны выполняться достаточно часто для обнаружения состояния, которое может привести к ВРЕДУ, прежде чем этот ВРЕД произойдет.

6.2.2.5 Меры по УПРАВЛЕНИЮ РИСКОМ для АНОМАЛИЙ программного обеспечения

Программное обеспечение представляет особую трудность: некоторые события в последовательности, приводящей к ОПАСНОЙ СИТУАЦИИ, могут возникнуть в результате необнаруженных АНОМАЛИЙ программного обеспечения, и трудно предсказать, где такие АНОМАЛИИ могут произойти или какие последствия вызвать.

Меры по УПРАВЛЕНИЮ РИСКОМ могут снижать вероятность ВРЕДА, происходящего от программных АНОМАЛИЙ. Обычно в АРХИТЕКТУРЕ программного обеспечения будут присутствовать области, где меры по УПРАВЛЕНИЮ РИСКОМ могут уменьшить вероятность ВРЕДА вне зависимости от природы предыдущих событий. Если это сделано тщательно, то нет необходимости выяснять точный характер программных АНОМАЛИЙ для предотвращения ВРЕДА, который может от них произойти.

В тех случаях, когда этот подход неприменим, например, если предупреждающая мера осуществлена в программном обеспечении, должны использоваться методы, обеспечивающие целостность программного обеспечения (см. 6.2.2.6).

6.2.2.6 ПРОЦЕСС — как мера по УПРАВЛЕНИЮ РИСКОМ

Если АНОМАЛИИ программного обеспечения могут способствовать последовательности событий, приводящей к ОПАСНОЙ СИТУАЦИИ, может оказаться невозможным разработать меры по УПРАВЛЕНИЮ РИСКОМ для предотвращения ВРЕДА от их возникновения. Наилучшим решением в таком случае будет конструктивно обусловленная БЕЗОПАСНОСТЬ, чтобы АНОМАЛИИ программного обеспечения не могли создать ОПАСНУЮ СИТУАЦИЮ.

Когда это неосуществимо, то с целью уменьшения вероятности возникновения АНОМАЛИЙ программного обеспечения может использоваться эффективный ПРОЦЕСС разработки программного обеспечения. Существует мнение, что меры по УПРАВЛЕНИЮ РИСКОМ ПРОЦЕССА выгодны, когда используются в сочетании с другими видами мер по УПРАВЛЕНИЮ РИСКОМ, если они определены в деталях.

Если установлено, что высока уверенность в том, что программное обеспечение будет выполнять свою предусмотренную функцию надежно, а также в том, что в программном обеспечении отсутствуют ошибки, то программное обеспечение можно рассматривать как компонент высокой целостности. Чтобы достичь этого высокого уровня надежности, ИЗГОТОВИТЕЛЬ должен продемонстрировать, что ПРОЦЕСС разработки программного обеспечения может создать высоконадежное, безотказное программное обеспечение. Использование такого ПРОЦЕССА может быть востребовано для уменьшения вероятности возникновения АНОМАЛИЙ программного обеспечения.

Установлено, что увеличение тщательности ПРОЦЕССА разработки программного обеспечения может сократить количество АНОМАЛИЙ программного обеспечения. Следует отметить, что, хотя испытания ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕДИЦИНСКОГО ИЗДЕЛИЯ могут сократить количество АНОМАЛИЙ программного обеспечения, нельзя утверждать, что, когда программное обеспечение проходит все запланированные тесты, в нем не остается никаких АНОМАЛИЙ. Это происходит, поскольку при клиническом использовании входные данные программного обеспечения будут включать последовательности, которые не входили в запланированный объем испытаний. Поскольку ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МЕДИЦИНСКОГО ИЗДЕЛИЯ является слишком сложным для проведения полномасштабных испытаний, то тщательный выбор и проведение конкретных испытаний может рассматриваться только как метод снижения возникновения вероятности ОПАСНЫХ СИТУАЦИЙ. Испытаний самих по себе недостаточно для обеспечения уверенности в том, что программное обеспечение можно рассматривать как компонент высокой целостности.

Отправной точкой для обеспечения тщательного ПРОЦЕССА разработки программного обеспечения может быть включение деятельности и ЗАДАЧ, определенных в МЭК 62304, в ПРОЦЕСС разработки программного обеспечения. Дополнительные элементы, которые следует учесть при рассмотрении тщательного ПРОЦЕССА разработки программного обеспечения, должны включать:

- компетентность персонала — навыки, квалификация, опыт и обучение (кто разрабатывает программное обеспечение?);
- методы — пригодность спецификации, проектирования, кодирования и методов испытаний (каков ПРОЦЕСС разработки?);

- точность, официальность, область анализа и тщательных проверок (сколько статических анализов выполняется?);

- инструменты — качество инструментов, таких как компиляторы, требования ПРОСЛЕЖИВАЕМОСТИ и инструменты управления конфигурацией (какие инструменты используются во время разработки программного обеспечения?).

Предсказуемость разработки программного обеспечения высокой целостности зависит от наличия воспроизводимости ПРОЦЕССА, который последовательно соблюдается.

Когда выполнение тщательного ПРОЦЕССА разработки используется, чтобы уменьшить РИСК ОПАСНЫХ СИТУАЦИЙ, проистекающих из программных АНОМАЛИЙ, должна быть продемонстрирована результативность мер по УПРАВЛЕНИЮ РИСКОМ при помощи сбора и анализа данных, показывающих частоту отказов из–за таких АНОМАЛИЙ. В поддержку соответствия требованию, что ПРОЦЕСС производит программное обеспечение высокой целостности, должны быть представлены доказательства, что отказы программного обеспечения отсутствуют или очень редки.

6.2.3 Рассмотрение программного обеспечения неизвестного происхождения (ПОНП)

Решение использовать ПОНП часто принимается во время проектирования СИСТЕМЫ. Чем выше потенциальный РИСК МЕДИЦИНСКОГО ИЗДЕЛИЯ, тем более внимательно должны анализироваться потенциальные виды отказов ПОНП и идентифицироваться меры по УПРАВЛЕНИЮ РИСКОМ. Обычно ПОНП не может быть модифицировано для включения новых мер по УПРАВЛЕНИЮ РИСКОМ, необходимых для мониторинга или изоляции ПОНП, чтобы оно не способствовало ОПАСНОЙ СИТУАЦИИ или ОПАСНОСТИ в случае отказа. При этом нет соответствующей информации о внутренней конструкции, доступной для определения потенциальных ОПАСНОСТЕЙ, вызываемых ПОНП. Поэтому СИСТЕМА и АРХИТЕКТУРА программного обеспечения должны быть разработаны так, чтобы обеспечить меры по УПРАВЛЕНИЮ РИСКОМ, необходимые для мониторинга или изоляции ПОНП для предотвращения ОПАСНОСТИ в случае его отказа.

Если ПОНП включено в МЕДИЦИНСКОЕ ИЗДЕЛИЕ, следует обратить внимание на предотвращение снижения БЕЗОПАСНОСТИ МЕДИЦИНСКОГО ИЗДЕЛИЯ. Такая ситуация может потребовать введения «оболочек» или промежуточной АРХИТЕКТУРЫ. Промежуточное программное обеспечение может:

a) предотвращать использование тех характеристик ПОНП, использование которых нежелательно;

b) выполнять логические проверки для обеспечения правильности информации, передаваемой между ПОНП и ПРОГРАММНЫМ ОБЕСПЕЧЕНИЕМ МЕДИЦИНСКОГО ИЗДЕЛИЯ, или

c) обеспечивать, в случае необходимости, МЕДИЦИНСКОЕ ИЗДЕЛИЕ дополнительной информацией.

Другой важной проблемой, связанной с ПОНП, является использование коммерческих операционных систем и коммуникаций. АРХИТЕКТУРА должна быть установлена так, чтобы допускать изменения (например, стабильность, ЗАЩИЩЕННОСТЬ) платформы программного обеспечения, основанной на полной ОЦЕНКЕ РИСКА без угрозы БЕЗОПАСНОСТИ. Такая ОЦЕНКА РИСКА должна включать в себя анализ частоты изменений, необходимых для обеспечения целостности БЕЗОПАСНОСТИ МЕДИЦИНСКОГО ИЗДЕЛИЯ, таких как, например, установка участков ЗАЩИЩЕННОЙ сети.

6.3 Выполнение мер по УПРАВЛЕНИЮ РИСКОМ

Текст ИСО 14971

6.3 Выполнение мер по УПРАВЛЕНИЮ РИСКОМ

ИЗГОТОВИТЕЛЬ должен выполнять меры по УПРАВЛЕНИЮ РИСКОМ, выбранные в соответствии с 6.2.

Выполнение каждой меры по УПРАВЛЕНИЮ РИСКОМ должно быть верифицировано. Данная ВЕРИФИКАЦИЯ должна быть зарегистрирована в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА.

Результативность мер по УПРАВЛЕНИЮ РИСКОМ должна быть верифицирована, а результаты ВЕРИФИКАЦИИ — зарегистрированы в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА.

Примечание — верификация результативности может включать действия по валидации.

Соответствие требованиям данного подраздела проверяют путем контроля файла МЕНЕДЖМЕНТА РИСКА.

Как только меры по УПРАВЛЕНИЮ РИСКОМ идентифицированы, они должны быть осуществлены, а их результативность верифицирована.

ВЕРИФИКАЦИЯ того, что меры по УПРАВЛЕНИЮ РИСКОМ были должным образом осуществлены и эффективны при управлении РИСКОМ, важна для программного обеспечения. Для этого может потребоваться как анализ, так и испытания. Основные аспекты для рассмотрения включают:

a) ПРОСЛЕЖИВАЕМОСТЬ для обеспечения уверенности в том, что все СВЯЗАННЫЕ С БЕЗОПАСНОСТЬЮ ПРОГРАММНЫЕ ЭЛЕМЕНТЫ идентифицированы, а так же вся связанная с БЕЗОПАСНОСТЬЮ функциональность определена, осуществлена и испытана во всех соответствующих ВЕРСИЯХ и вариантах программного обеспечения (например, для различных платформ, языков или моделей МЕДИЦИНСКОГО ИЗДЕЛИЯ);

b) большую тщательность и объем испытаний в отношении мер по УПРАВЛЕНИЮ РИСКОМ, включая испытания в широком диапазоне ненадлежащих и экстремальных условий;

c) сосредоточение на регрессионных испытаниях мер по УПРАВЛЕНИЮ РИСКОМ и связанной с БЕЗОПАСНОСТЬЮ функциональностью, когда вносятся изменения, даже если эти изменения не предусмотрены в отношении влияния на БЕЗОПАСНОСТЬ.

Если используется процесс, который считается достаточно тщательным для создания программного обеспечения высокой целостности, результаты все равно должны быть верифицированы.

Информация об АНОМАЛИИ, собранная во время деятельности по ВЕРИФИКАЦИИ и валидации, часто бывает полезна, если был выполнен анализ основных причин и значения (рейтинги) критичности аномалии (см. МЭК 62304, п. 9.1), связанной с БЕЗОПАСНОСТЬЮ, прослежены и оценены. АНОМАЛИИ совместно с последствиями, влияющими на БЕЗОПАСНОСТЬ, могут быть оценены с целью определения того, что они идентифицированы при ОЦЕНКЕ РИСКА и для них будут достаточны идентифицированные меры по УПРАВЛЕНИЮ РИСКОМ, которые однажды применены. Данные по АНОМАЛИЯМ программного обеспечения могут использоваться для демонстрации результативности ПРОЦЕССА разработки программного обеспечения или для определения аспектов ПРОЦЕССА разработки программного обеспечения, которые нуждаются в улучшении для снижения РИСКА.

Достаточность программных мер по УПРАВЛЕНИЮ РИСКОМ может быть не такой же очевидной, как достаточность аппаратных мер по УПРАВЛЕНИЮ РИСКОМ. По этой причине, имея дело с программным обеспечением, нужно учитывать, требует ли документация по оценке РИСКА других форматов, чем те, которые традиционно требуются. Один из способов, который может оказаться полезным, предполагает составление списка «случаев БЕЗОПАСНОСТИ» (см. приложение Е).

6.4 ОЦЕНИВАНИЕ ОСТАТОЧНОГО РИСКА

Текст ИСО 14971

6.4 ОЦЕНИВАНИЕ ОСТАТОЧНОГО РИСКА

Любой ОСТАТОЧНЫЙ РИСК, сохраняющийся после выполнения мер по УПРАВЛЕНИЮ РИСКОМ, необходимо оценивать в соответствии с критериями, установленными в плане МЕНЕДЖМЕНТА РИСКА. Результаты данного оценивания должны быть зарегистрированы в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА.

Если остаточный(е) РИСК(И) согласно установленным критериям оценен(ы) как недопустимый(е), то необходимо применить дополнительные меры по УПРАВЛЕНИЮ РИСКОМ (см. 6.2).

Если остаточные РИСКИ оценены как допустимые, то ИЗГОТОВИТЕЛЬ должен принять решение, о каких остаточных РИСКАХ необходимо информировать и какую именно информацию необходимо включить в эксплуатационные документы в целях привлечения внимания к остаточным РИСКАМ.

Примечание — Руководящие указания по информированию об остаточном РИСКЕ(АХ) приведены в приложении J.

Соответствие требованиям данного подраздела проверяют путем контроля файла МЕНЕДЖМЕНТА РИСКА и эксплуатационных документов.

ОСТАТОЧНЫЙ РИСК из-за программного обеспечения должен быть включен в ОСТАТОЧНЫЙ РИСК, связанный с МЕДИЦИНСКИМ ИЗДЕЛИЕМ на уровне СИСТЕМЫ. Учитывая трудность в определении вероятности АНОМАЛИЙ программного обеспечения, ОЦЕНИВАНИЕ ОСТАТОЧНОГО РИСКА обычно включает установление того, что для всех последовательностей событий, которые приводят к недопустимому РИСКУ, разработаны меры по УПРАВЛЕНИЮ РИСКОМ для снижения вероятности их возникновения или для ограничения ТЯЖЕСТИ ВРЕДА до допустимого уровня, установленного в плане МЕНЕДЖМЕНТА РИСКА (см. 3.4.1).

Любые идентифицированные (во время ВЕРИФИКАЦИИ и валидации) АНОМАЛИИ программного обеспечения, которые не исправлены, должны быть проанализированы для определения их влияния на связанное с БЕЗОПАСНОСТЬЮ программное обеспечение (см. МЭК 62304, п. 5.8.3). В этом случае РИСК от АНОМАЛИЙ должен быть оценен и использован при оценивании ОСТАТОЧНОГО РИСКА любой ОПАСНОЙ СИТУАЦИИ, на которую они могут оказывать влияние.

6.5 Анализ соотношения РИСК/польза

Текст ИСО 14971

6.5 Анализ соотношения РИСК/польза

Если ОСТАТОЧНЫЙ РИСК по критериям, установленным в плане МЕНЕДЖМЕНТА РИСКА, сочтен недопустимым, а дальнейшее УПРАВЛЕНИЕ РИСКОМ — практически неосуществимым, то ИЗГОТОВИТЕЛЬ может собрать и проанализировать имеющиеся данные и литературу, чтобы установить, превышает ли польза при предусмотренном применении МЕДИЦИНСКОГО ИЗДЕЛИЯ ОСТАТОЧНЫЙ РИСК. Если собранные доказательства свидетельствуют о том, что польза от предусмотренного применения МЕДИЦИНСКОГО ИЗДЕЛИЯ не превышает ОСТАТОЧНЫЙ РИСК, то РИСК считается недопустимым. Если польза от предусмотренного применения МЕДИЦИНСКОГО ИЗДЕЛИЯ превышает ОСТАТОЧНЫЙ РИСК, то можно перейти к выполнению требований 6.6.

В отношении РИСКА, где польза от применения МЕДИЦИНСКОГО ИЗДЕЛИЯ превышает РИСК, ИЗГОТОВИТЕЛЬ должен решить, какую информацию по БЕЗОПАСНОСТИ необходимо предоставить для информирования об остаточном РИСКЕ.

Результаты оценивания должны быть зарегистрированы в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА.

Примечание — См. также D.6.

Соответствие требованиям данного подраздела проверяют путем контроля файла МЕНЕДЖМЕНТА РИСКА.

Нет дополнительных разъяснений для программного обеспечения.

6.6 РИСКИ, возникающие вследствие мер по УПРАВЛЕНИЮ РИСКОМ

Текст ИСО 14971

6.6 РИСКИ, возникающие вследствие выполнения мер по УПРАВЛЕНИЮ РИСКОМ

Эффективность выполнения мер по УПРАВЛЕНИЮ РИСКОМ необходимо анализировать с точки зрения:

a) возникновения новых ОПАСНОСТЕЙ и ОПАСНЫХ СИТУАЦИЙ;

b) влияния выполненных мер по УПРАВЛЕНИЮ РИСКОМ на РИСКИ, определенные для ранее идентифицированных ОПАСНЫХ СИТУАЦИЙ.

Любыми новыми или возросшими РИСКАМИ необходимо управлять в соответствии с требованиями, приведенными в 4.4—6.5.

Результаты анализа мер по УПРАВЛЕНИЮ РИСКОМ следует зарегистрировать в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА.

Соответствие требованиям данного подраздела проверяют путем контроля файла МЕНЕДЖМЕНТА РИСКА.

Тщательный ПРОЦЕСС менеджмента конфигурации программного обеспечения, включая тщательное управление изменениями (см. МЭК 62304, п. 8.2), является важным, и поэтому введение мер по УПРАВЛЕНИЮ РИСКОМ должно тщательно исследоваться на предмет их влияния на другие части МЕДИЦИНСКОГО ИЗДЕЛИЯ (см. также МЭК 62304, п. 7.4).

ИСО 14971 не устанавливает ПРОЦЕСС проектирования и разработки. ИСО 14971 требует, что, когда мера по УПРАВЛЕНИЮ РИСКОМ осуществлена, она должна быть проанализирована еще раз, чтобы убедиться, что она не вызывает каких-либо дальнейших ОПАСНОСТЕЙ. Это не должно пониматься как указание исследовать этот вопрос только после того, как эта мера осуществлена.

В отношении мер по УПРАВЛЕНИЮ РИСКОМ особенно важно, чтобы этот анализ продолжался до тех пор, пока программное обеспечение не будет разработано. Как только мера по УПРАВЛЕНИЮ РИСКОМ определена, она должна подпадать под менеджмент конфигурации и проанализирована с целью обнаружения неблагоприятных побочных эффектов, включая неумышленное создание новых ОПАСНОСТЕЙ или ОПАСНЫХ СИТУАЦИЙ.

Выполнение меры по УПРАВЛЕНИЮ РИСКОМ, которая делает программную конструкцию значительно более сложной, может увеличить потенциал возникновения дополнительных АНОМАЛИЙ программного обеспечения или новых ОПАСНЫХ СИТУАЦИЙ. По возможности меры по УПРАВЛЕНИЮ РИСКОМ должны быть максимально простыми и всегда подлежат новой оценке РИСКА.

Этот анализ следует повторить (как минимум) после разработки программного обеспечения и после испытаний ПРОГРАММНОЙ СИСТЕМЫ.

6.7 Полнота УПРАВЛЕНИЯ РИСКОМ

Текст ИСО 14971

6.7 Полнота УПРАВЛЕНИЯ РИСКОМ

ИЗГОТОВИТЕЛЬ должен обеспечить рассмотрение РИСКА(ОВ) для всех идентифицированных ОПАСНЫХ СИТУАЦИЙ. Результаты данной деятельности должны быть зарегистрированы в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА.

Соответствие требованиям данного подраздела проверяют путем контроля файла МЕНЕДЖМЕНТА РИСКА.

Если программное обеспечение является способствующим фактором в создании ОПАСНЫХ СИТУАЦИЙ, МЭК 62304, п. 7.3.3 должен учитываться для включения в полноценную деятельность по УПРАВЛЕНИЮ РИСКОМ.

7 ОЦЕНИВАНИЕ допустимости совокупного ОСТАТОЧНОГО РИСКА

Текст ИСО 14971

7 ОЦЕНИВАНИЕ допустимости совокупного ОСТАТОЧНОГО РИСКА

После выполнения и ВЕРИФИКАЦИИ всех мер по УПРАВЛЕНИЮ РИСКОМ ИЗГОТОВИТЕЛЬ должен принять решение о допустимости совокупного ОСТАТОЧНОГО РИСКА, создаваемого МЕДИЦИНСКИМ ИЗДЕЛИЕМ, с точки зрения критериев, установленных в плане МЕНЕДЖМЕНТА РИСКА.

Примечание — Руководящие указания по ОЦЕНИВАНИЮ совокупного ОСТАТОЧНОГО РИСКА приведены в D.7.

Если совокупный ОСТАТОЧНЫЙ РИСК по критериям, установленным в плане МЕНЕДЖМЕНТА РИСКА, оценен как недопустимый, то ИЗГОТОВИТЕЛЬ может собрать и проанализировать имеющиеся данные и литературу, чтобы установить, превышает ли польза при предусмотренном применении МЕДИЦИНСКОГО ИЗДЕЛИЯ совокупный ОСТАТОЧНЫЙ РИСК. Если собранные доказательства свидетельствуют о том, что польза от предусмотренного применения МЕДИЦИНСКОГО ИЗДЕЛИЯ превышает совокупный ОСТАТОЧНЫЙ РИСК, то РИСК считается допустимым. В противном случае совокупный ОСТАТОЧНЫЙ РИСК считается недопустимым.

В отношении совокупного ОСТАТОЧНОГО РИСКА, который оценен как допустимый, ИЗГОТОВИТЕЛЬ должен решить, какую информацию необходимо включить в ЭКСПЛУАТАЦИОННЫЕ ДОКУМЕНТЫ для информирования о совокупном остаточном РИСКЕ.

Примечание — Руководящие указания по информированию об остаточном РИСКЕ(АХ) приведены в приложении J.

Результаты оценивания совокупного ОСТАТОЧНОГО РИСКА должны быть зарегистрированы в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА.

Соответствие требованиям данного раздела проверяют путем контроля файла МЕНЕДЖМЕНТА РИСКА и ЭКСПЛУАТАЦИОННЫХ ДОКУМЕНТОВ.

Оценивание совокупного ОСТАТОЧНОГО РИСКА требует реализации всех мер по УПРАВЛЕНИЮ РИСКОМ. Это включает в себя программное обеспечение, оцениваемое в контексте каждой различной СИСТЕМНОЙ конфигурации, которую использует программное обеспечение.

Результаты всех испытаний СИСТЕМЫ (относительно всей функциональности программного обеспечения и аппаратного УПРАВЛЕНИЯ РИСКОМ) должны быть оценены в соответствии с критериями допустимости. Все оставшиеся остаточные АНОМАЛИИ программного обеспечения должны быть документированы в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА и оценены для обеспечения уверенности в том, что они не способствуют возникновению недопустимого РИСКА (см. МЭК 62304, п.п. 5.8.2 и 5.8.3). В тех случаях, где это необходимо, такое оценивание должно осуществляться посредством независимого междисциплинарного анализа с привлечением медицинских экспертов и экспертов по прикладным направлениям. Может быть также необходимым включение информации в ЭКСПЛУАТАЦИОННЫЕ ДОКУМЕНТЫ.

8 Отчет по МЕНЕДЖМЕНТУ РИСКА

Текст ИСО 14971

8 Отчет по МЕНЕДЖМЕНТУ РИСКА

Перед введением МЕДИЦИНСКОГО ИЗДЕЛИЯ в обращение ИЗГОТОВИТЕЛЬ должен провести анализ ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА. Данный анализ должен, по меньшей мере, свидетельствовать о том, что:

- план МЕНЕДЖМЕНТА РИСКА реализован должным образом;
- совокупный ОСТАТОЧНЫЙ РИСК является допустимым;
- применяют надлежащие способы получения необходимой производственной и ПОСТПРОИЗВОДСТВЕННОЙ информации.

Результаты анализа ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА должны быть зарегистрированы в отчете по МЕНЕДЖМЕНТУ РИСКА и включены в ФАЙЛ МЕНЕДЖМЕНТА РИСКА.

В плане МЕНЕДЖМЕНТА РИСКА должны быть указаны лица, имеющие необходимые полномочия и ответственные за проведение анализа ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА [см. 3.4, перечисление b)].

Соответствие данному разделу проверяют путем контроля файла МЕНЕДЖМЕНТА РИСКА.

Раздел 6 и пп. 7.3.3 МЭК 62304 должны быть рассмотрены для включения в состав анализа ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА.

9 Производственная и ПОСТПРОИЗВОДСТВЕННАЯ информация

Текст ИСО 14971

9 Производственная и ПОСТПРОИЗВОДСТВЕННАЯ информация

ИЗГОТОВИТЕЛЬ должен разработать, документировать и поддерживать в рабочем состоянии систему сбора и анализа информации о рассматриваемом МЕДИЦИНСКОМ ИЗДЕЛИИ или подобных изделиях на стадиях производства и постпроизводства.

При разработке системы сбора и анализа информации о рассматриваемом МЕДИЦИНСКОМ ИЗДЕЛИИ ИЗГОТОВИТЕЛЬ среди прочего должен учитывать:

a) механизмы, с помощью которых можно собирать и обрабатывать информацию, поступающую от операторов, пользователей или других лиц, ответственных за установку/монтаж, применение и поддержание в рабочем состоянии МЕДИЦИНСКОГО ИЗДЕЛИЯ;

b) новые или пересмотренные стандарты.

В рамках данной системы следует также собирать и анализировать общедоступную информацию, опубликованную о подобных МЕДИЦИНСКИХ ИЗДЕЛИЯХ, находящихся на рынке.

Данную информацию необходимо оценивать с точки зрения соответствия требованиям к БЕЗОПАСНОСТИ, особенно с учетом того:

- существуют ли не выявленные ранее ОПАСНОСТИ или ОПАСНЫЕ СИТУАЦИИ;
- не стал ли недопустимым РИСК(И), определенный(ые) ранее для какой-либо ОПАСНОЙ СИТУАЦИИ.

При положительном ответе на любой из данных вопросов необходимо:

- оценить влияние вышеуказанных факторов на ранее осуществленную деятельность по МЕНЕДЖМЕНТУ РИСКА и результаты оценивания вернуть на вход ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА;
- провести анализ файла МЕНЕДЖМЕНТА РИСКА по рассматриваемому МЕДИЦИНСКОМУ ИЗДЕЛИЮ; при наличии потенциальной возможности изменения ОСТАТОЧНОГО РИСКА(ОВ) или его (их) допустимости следует оценить воздействие вышеуказанных факторов на ранее выполненные меры по УПРАВЛЕНИЮ РИСКОМ.

Результаты оценивания должны быть зарегистрированы в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА.

Примечания

1 Отдельные аспекты ПОСТПРОИЗВОДСТВЕННОГО мониторинга являются предметом национальных или региональных регламентов. В таких случаях могут быть задействованы дополнительные меры (например, перспективное ПОСТПРОИЗВОДСТВЕННОЕ оценивание).

2 См. также [3], 8.2.

Соответствие требованиям данного раздела проверяют путем контроля файла МЕНЕДЖМЕНТА РИСКА и другой соответствующей документации.

МЕНЕДЖМЕНТ РИСКА программного обеспечения продолжается в течение всего ЖИЗНЕННОГО ЦИКЛА программного обеспечения, включая ПРОЦЕСС технической поддержки программного обеспечения (см. МЭК 62304, раздел 6) и ПРОЦЕСС решения проблем программного обеспечения (см. МЭК 62304, раздел 9).

Раздел 6 МЭК 62304 требует, чтобы ИЗГОТОВИТЕЛЬ установил планы поддержки программного обеспечения, которые охватывают использование процедур получения, документирования, оценивания, разрешения и отслеживания обратной связи после выпуска в обращение программного обеспечения МЕДИЦИНСКОГО ИЗДЕЛИЯ. План(ы) технического обслуживания необходимы для того, чтобы обратиться к использованию ПРОЦЕССА МЕНЕДЖМЕНТА РИСКА программного обеспечения и использованию ПРОЦЕССА решения проблем программного обеспечения для анализа и решения проблем, возникающих после выпуска программного обеспечения МЕДИЦИНСКОГО ИЗДЕЛИЯ.

Использование ПРОЦЕССА решения проблем программного обеспечения (см. МЭК 62304, раздел 9) объединяет деятельность по МЕНЕДЖМЕНТУ РИСКА в исследовании проблем программного обеспечения и оценивания актуальности проблемы в отношении БЕЗОПАСНОСТИ. Важно подключить к такому исследованию и оцениванию междисциплинарную команду, включая медицинских экспертов, разработчиков программного обеспечения, СИСТЕМНЫХ проектировщиков и экспертов по эксплуатационной пригодности (см. 3.3).

ПОНП является важным аспектом плана технической поддержки программного обеспечения и деятельности по МЕНЕДЖМЕНТУ РИСКА на стадии постпроизводства. Некоторое ПОНП по его характеристикам (например, вирусная защита программного обеспечения) может часто обновляться, и это обстоятельство ИЗГОТОВИТЕЛЬ должен учитывать в плане(ах) технической поддержки.

Отказы или неожиданные результаты ПОНП и устаревание ПОНП (прекращение поддержки) могут повлиять на допустимость совокупного ОСТАТОЧНОГО РИСКА МЕДИЦИНСКОГО ИЗДЕЛИЯ. Поэтому необходимо осуществлять работы по мониторингу и оцениванию ПОНП при разработке и техническом обслуживании ПРОГРАММНОЙ СИСТЕМЫ. Эта деятельность должна быть направлена на обновления ПОНП, модернизацию, исправление ошибок, таюке должно уделяться внимание патчам и устареванию. Активный мониторинг и оценивание общедоступных списков АНОМАЛИЙ и общей информации об эксплуатационных характеристиках в области ПОНП позволяют ИЗГОТОВИТЕЛЮ заранее определить, приводит ли какая-нибудь из известных АНОМАЛИЙ к последовательности событий, которые могут привести к ОПАСНОЙ СИТУАЦИИ (см. МЭК 62304, п.п. 6.1 f), 7.1.2 c), 7.1.3 и 7.4.2).

Выпущенные части (патчи) ПОНП или обновления от ИЗГОТОВИТЕЛЯ могут включать дополнительные функции, которые не важны для БЕЗОПАСНОСТИ и результативности МЕДИЦИНСКОГО ИЗДЕЛИЯ. Эти обновления ПОНП должны быть проанализированы на избыточные компоненты, которые могут быть исключены из медицинской версии программного обеспечения, чтобы избежать неожиданных изменений, которые могут привести к ОПАСНЫМ СИТУАЦИЯМ.

Как и при любом изменении ПРОГРАММНОГО ЭЛЕМЕНТА, ИЗГОТОВИТЕЛЬ должен знать, какие элементы программного обеспечения затронуты обновлением ПОНП, и выполнить регрессионные испытания (см. МЭК 62304, п.п. 7.4, 8.2 и 9.7).

Приложение А
(справочное)

Обсуждение определений

ОПАСНОСТЬ и ВРЕД являются одними из ключевых терминов ИСО 14971. Их правильное понимание очень важно для понимания всего стандарта. Вред определен как «физическое повреждение или ущерб здоровью людей, имуществу или окружающей среде». ОПАСНОСТЬ определена как «потенциальный источник ВРЕДА», и есть много причин для возникновения ОПАСНОСТИ.

Согласно определениям в ИСО 14971, ОПАСНОСТЬ не может привести к вреду до тех пор, пока последовательность событий или других обстоятельств (включая условия нормального применения) не приведет к возникновению ОПАСНОЙ СИТУАЦИИ (см. рисунок А.1). Эта последовательность событий может включать как единичные события, так и комбинации событий. Приложение D.2 ИСО 14971 дает представление об ОПАСНОСТЯХ и ОПАСНЫХ СИТУАЦИЯХ. Приложение Е ИСО 14971 дает представление о примерах ОПАСНОСТЕЙ, прогнозируемых последовательностях событий и ОПАСНЫХ СИТУАЦИЯХ.

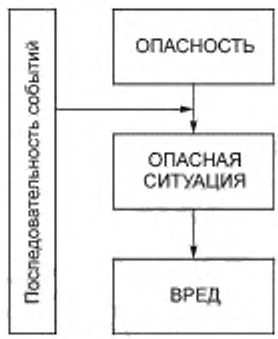

Рисунок А.1 — Взаимосвязь между последовательностью событий, ВРЕДОМ и ОПАСНОСТЬЮ

Причиной ОПАСНОСТИ или ОПАСНОЙ СИТУАЦИИ может быть любая последовательность событий, комбинация которых может привести к ОПАСНОЙ СИТУАЦИИ. Данная ОПАСНОСТЬ может иметь одну, несколько или множество возможных причин (последовательностей событий).

В отличие от высокой температуры, электроэнергии и подвешенных грузов, программное обеспечение само по себе не является ОПАСНОСТЬЮ (потенциальным источником ВРЕДА). Контакт с программным обеспечением не может нанести физических повреждений. Однако программное обеспечение может подвергнуть человека ОПАСНОСТИ, другими словами, может способствовать ОПАСНОЙ СИТУАЦИИ. Отказы программного обеспечения (любого рода) часто способствуют преобразованию ОПАСНОСТИ в ОПАСНУЮ СИТУАЦИЮ.

Программное обеспечение может вызывать ОПАСНЫЕ СИТУАЦИИ, способствуя, прежде всего, возникновению последовательности событий, которая переводит ОПАСНОСТЬ в ОПАСНУЮ СИТУАЦИЮ (См. таблицу А.1).

Таблица А.1 — Взаимосвязь между опасностями, прогнозируемыми последовательностями событий, опасными ситуациями и возможным ВРЕДОМ.

Опасность	Предсказуемая последовательность событий, включая программное обеспечение	Инициирующая причина	Опасная ситуация	Вред
Электрическая энергия	Выходные данные программного обеспечения, контролирующие выходной ток, подающийся на глазной имплантат, обуславливают слишком высокий ток	(1) Программный алгоритм имеет ограничения. (2) Программное обеспечение правильно определено, но имеет АНОМАЛИИ	Чрезмерный ток подается к глазу пациента через имплантат	Серьезные ожоги. Потеря зрения

Окончание таблицы А.1

Опасность	Предсказуемая последовательность событий, включая программное обеспечение	Инициирующая причина	Опасная ситуация	Вред
Потеря медицинской функции	Программное обеспечение отказывает при обеспечении функции поддержания жизни, для которой оно проектировалось	(1) Программное обеспечение не способно работать с необычными входными данными. (2) Программное обеспечение отказывает при обнаружении неверных установок оборудования. (3) Аппаратные средства не предоставляют достаточных ресурсов, чтобы поддерживать своевременную работу устройства	(1) Изделие не может обеспечить лечение, когда требуется. (2) Изделие работает при неправильной установке. (3) Изделие не может предупреждать об угрожающих жизни состояниях	Ухудшение состояния пациента. Смерть
Игнорирование пациента медицинским персоналом.	Входные и выходные данные программмного обеспечения путают или вводят в заблуждение пользователя	1) Интерфейс ПО-человек путает пользователя. (2) Выходные данные программы превышают способность пользователя реагировать. (3) Пользователь не понимает ограничений программного обеспечения	(1) Неверное лечение пациента. (2) Отсутствие своевременной реакции на чрезвычайные ситуации. (3) Чрезмерное доверие программному обеспечению заменяет личную инициативу	Ухудшение состояния пациента. Смерть

Приложение В
(справочное)

Примеры использования программных средств

Таблица В.1 перечисляет функциональные области программного обеспечения, часто связываемые с ОПАСНОСТЬЮ, и предоставляет примеры условий, которые являются потенциальными причинами ОПАСНОСТЕЙ. Также она приводит примеры вопросов, задаваемых во время разработки программного обеспечения, которые могут помочь достигнуть более тщательного УПРАВЛЕНИЯ РИСКОМ. Часть информации в этой таблице может относиться не ко всему программному обеспечению МЕДИЦИНСКОГО ИЗДЕЛИЯ. Уместность информации для отдельных МЕДИЦИНСКИХ ИЗДЕЛИЙ зависит от намеченного применения МЕДИЦИНСКОГО ИЗДЕЛИЯ, дизайна уровня системы МЕДИЦИНСКОГО ИЗДЕЛИЯ, роли программного обеспечения в МЕДИЦИНСКОМ ИЗДЕЛИИ и других факторов. Данная таблица предназначена только в качестве отправной точки.

Эта таблица не является исчерпывающей, но она может помочь при разработке безопасного и эффективного программного обеспечения.

Таблица В.1 — Примеры использования программных средств

Область функционирования программного обеспечения	Пример причин опасности	Задаваемые вопросы
Аварийные сигналы и предупреждения		
Приоритет	Сигнал тревоги с более низким приоритетом перекрывает отображение или аудио оповещение сигнала тревоги с более высоким приоритетом. Критические сообщения о сигналах тревоги не сохраняются	Определяют ли спецификации, как система реагирует на множественные аварийные условия? Есть ли несколько уровней сигналов тревоги? Перекрывает ли сигнал тревоги более высокого уровня аудиосигнал тревоги с более низким уровнем? Сохраняется ли какой-либо сигнал тревоги, пока пользователь не отреагирует на него?
Защитные меры	Защитные меры не определены для каждого состояния тревоги или категории тревог. Не определен сброс тревоги после срабатывания аварийной защиты	Создают ли защитные меры проблемы в использовании, т. е. может ли пользователь безопасно далее использовать изделие после защитного действия?
Отключение/безопасный режим/восстановление	Действия безопасного режима недостаточны. Действие безопасного режима создает новые ОПАСНОСТИ	Является ли действие безопасного режима подходящим для намеченного применения? Рассмотрел ли медицинский персонал сценарии безопасного режима? Очевидно ли состояние безопасного режима для пользователя?
Пользовательский интерфейс	Переполненный или плохой пользовательский интерфейс маскирует «реальное» аварийное условие. Действия, предпринимаемые в ответ на сигнал тревоги, не ясны	Рассмотрел ли медицинский персонал защитные меры для использования?
Журнал	Постоянная ошибка предупреждает о неминуемом отказе. Зарегистрированные тревоги связываются с неправильным пациентом	Обнаруженные ошибки зарегистрированы? Журнал является достаточно большим? Хранение журнала надежно? Как очищается журнал? Пользователь знает, когда журнал очищается?

Продолжение таблицы В.1

Область функционирования программного обеспечения	Пример причин опасности	Задаваемые вопросы
Слышимость	Фоновый шум подавляет сигнал тревоги. Сигнал тревоги настолько громкий, что операторы находят альтернативные методы, чтобы отключить сигнал тревоги. Аудиосистема отказывает, и пользователь не узнает об ошибке	Учитывается ли окружающая среда предназначенного применения при разработке аудиосигнала? Были ли пользователи вовлечены в разработку проекта пользовательского интерфейса? Как аудиосистема проверяется при включении или при работе с пациентом?
Критические состояния цикла электропитания		
Целостность данных	Энергонезависимая запись в ПРОЦЕССЕ при отключении. Критические параметры не сохраняются, чтобы возобновлять лечение после отключения электропитания. Критические параметры непреднамеренно перезаписываются из-за скрытой АНОМАЛИИ программного обеспечения во время его работы	Что происходит с записью памяти, которая была в ПРОЦЕССЕ, когда отключается электроэнергия? Программное обеспечение изготовлено с учетом возможного отключения электроэнергии? Энергонезависимое хранение проверено при включении напряжения? Критические параметры проверены перед использованием?
Перезагрузка	Ошибка синхронизации с компонентами при неожиданной перезагрузке. Постоянные перезагрузки могут не быть обнаружены, но могут сигнализировать о предстоящем отказе	Перезагрузка использовалась как мера УПРАВЛЕНИЯ РИСКОМ? Ставится ли вход/выход управления под угрозу во время цикла перезагрузки? Пользователь знает о перезагрузках?
Восстановление	МЕДИЦИНСКОЕ ИЗДЕЛИЕ недоступно для предназначенного применения, однако инициализация при включении происходит. Энергонезависимые отказы при включении — что вы делаете? Пользователь не знает, что критические настройки восстанавливаются заводскими настройками по умолчанию	Время восстановления является проблемой БЕЗОПАСНОСТИ? Доступность МЕДИЦИНСКОГО ИЗДЕЛИЯ является проблемой БЕЗОПАСНОСТИ? Как энергонезависимая память влияет на отказоустойчивые защитные меры?
Режимы энергопотребления	Аудиофункции или другие критические функции пользовательского интерфейса недоступны при режиме с низким энергопотреблением. Переход в состояние с низким энергопотреблением делает невозможным критическое прерывание	Любые меры УПРАВЛЕНИЯ РИСКОМ поставлены под угрозу при режимах с низкой мощностью? Можно ли восстановление после режимов с низкой мощностью рассматривать как стартовое состояние для деятельности по ВЕРИФИКАЦИИ и валидации?
Критические пользовательские средства управления/применимость		
Регулирование/навигация/выбор	ПРОЦЕСС программного обеспечения пользовательского интерфейса обрабатывает изменение величин, но управление ПРОЦЕССОМ никогда не получает нового значения.	Пользователь уведомлен, что корректировка получает новое значение, но никогда не выбирается и не подтверждается? Требует ли приспособляемый параметр двухступенчатой операции для изменения?

Продолжение таблицы В.1

Область функционирования программного обеспечения	Пример причин опасности	Задаваемые вопросы
Ввод данных	Пользовательские входные данные находятся вне диапазона значений. Пользовательские входные данные в пределах диапазона, но со случайными величинами	Следует ли побудить пользователя к подтверждению? Программное обеспечение проверяет вводимые данные на действительность? Требуется ли от пользователя регистрация в качестве администратора, чтобы подтвердить критические входные данные или отвергнуть ошибку?
Доступность	Скрытое управление отключением. Средства управления сенсорным экраном не функционируют при работе в хирургических перчатках.	Сколько «слоев» должен перейти пользователь, чтобы получить доступ к функциям, связанным с БЕЗОПАСНОСТЬЮ?
Изменения экрана	Сигнал тревоги «автоматически» вызывает показ экранов выключателя	Были ли оценены автоматические выключатели экрана?
Конструкция пользовательского интерфейса	Использование цветов игнорирует общие формы дальтонизма. Пользователь не может определить, какое условие вызвало тревогу	Как будет страдающий дальтонизмом оператор интерпретировать сообщение об ошибке? Были ли пользователи привлечены к разработке требований конструкции пользовательского интерфейса?
Отображение		
Диагностические изображения	Аннулирование ориентации. Изображение некорректно соотносится с пациентом	Используются ли методы, гарантирующие правильную ориентацию изображения? Как изображения связаны с пациентом?
Диагностические кривые	Неподходящий фильтр отображения. Совмещение имен, искажение, ошибки измерений, ошибки развертки, сжатие с потерями	Какое содержание частоты требуется для отображения? Медицинский персонал был ознакомлен с этим требованием? Был ли фильтр отображения полностью охарактеризован, то есть известно ли, что отклоняется и что пропускается сверх полного диапазона входных данных?
Аппаратные средства контроля		
Алгоритмы управления двигателем	Составное завершение, совмещение имен, синхронизация, переполнение, ошибка порта (последовательный порт/параллельный порт)	Какова частота выборки? При пропорционально-интегрально-дифференциальном (ПИД) регулировании интегральное звено ограничивается? Алгоритм был охарактеризован по всем вариациям выпускаемых аппаратных средств? При управлении с обратной связью, какие проверки осуществляются для подтверждения сигнала обратной связи? Все ли типы данных были оценены для микропроцессора и компилятора в использовании?

Продолжение таблицы В.1

Область функционирования программного обеспечения	Пример причин опасности	Задаваемые вопросы
Применяемая энергия	Отсутствие проверки всей энергии «входов» первоначально и непрерывно во время терапии. Система БЕЗОПАСНОСТИ отказывает, но пользователь об этом не осведомлён	Все утверждения постоянно верифицируются на запланированной основе? Может ли ошибка «общего режима» существовать в программном обеспечении управления терапией и программном обеспечении монитора БЕЗОПАСНОСТИ? Мониторы БЕЗОПАСНОСТИ проверены при скачке напряжения или в присутствии пациента?
Дискретность	«Застрявший» бит. Изменения бита остаются необнаруженными из-за интервала опроса	Программное обеспечение обнаруживает тот бит, который «застрял» (никогда не меняется)? Обсуждался ли интервал опроса с системным или аппаратным инженером?
Калибровка/ самотестирование Проверки диапазона Калибровка испытания (определенная калибровка программного обеспечения)	Плохие инструкции не позволяют пользователю правильно калибровать МЕДИЦИНСКОЕ ИЗДЕЛИЕ, что приводит к ошибочным калибровочным константам. Операция автоустановки нуля, при которой дан отличный от нуля сигнал, т. е. неожиданное давление на манжете или в линии, или сила на преобразователе	Выполняет ли программное обеспечение проверку обоснованности и достоверности калибровочных величин, т. е. отклонение или смещение? Пользователь знает об автокалибровке и автонуле?
Обнаружение ошибки аппаратных средств	Ошибка аппаратных средств может быть обнаружена, но о ней никогда не сообщается пользователю МЕДИЦИНСКОЕ ИЗДЕЛИЕ продолжает использоваться при таких условиях. Ошибка аппаратных средств происходит после запуска, программное обеспечение только проверяет на ошибки аппаратные средства при запуске	Пользователю сообщается обо всех ошибках аппаратных средств? Должно ли наличие ошибки аппаратных средств проверяться каждый раз при запуске, перед каждым обращением или сессионно, или на непрерывной основе, например, раз в секунду?
Самоочистка	Пользователь прерывает ПРОЦЕСС очистки в середине цикла	Программное обеспечение проводит завершение цикла? Может ли потерпеть неудачу программное обнаружение незавершенного цикла очистки?
Изменяющаяся поставка	Обнаружение неподходящей калибровки. Отсутствие проверки всех изменяющихся «входов» первоначально и во время терапии	Все утверждения постоянно подтверждаются на запланированной основе? Может ли система БЕЗОПАСНОСТИ потерпеть неудачу, например, помпа работает без трубки в зажиме БЕЗОПАСНОСТИ?
Жизнеобеспечение	Безопасное состояние никогда не определено. Из многих путей блокировки один не делает невозможными прерывания. Не делается резервных копий для функций жизнеобеспечения	Были полностью определены и проанализированы безопасные состояния, включая влияние задержки обращения и безопасные блокировки последовательностей для диапазона целевых групп населения (например, взрослые, новорожденные)? Программное обеспечение может поддерживать режим «ограниченной функциональности» и информировать пользователя об этой ситуации?

Продолжение таблицы В.1

Область функционирования программного обеспечения	Пример причин опасности	Задаваемые вопросы
Мониторинг		
Решение	Общий режим ошибки в проверяемом программмном обеспечении. Работа нескольких программ вызывает неправильный результат решения	Программное обеспечение управления терапией и программное обеспечение мониторинга терапии разрабатывались отдельно? Программное обеспечение ограничивает или минимизирует условия для конфликта ресурсов в контрольной точке?
Деактивация	Система управления не оповещается о том, что система мониторинга отключает подсистему. Сетевая система регистрирует параметр, который деактивирован в медицинском изделии	Управляющая подсистема осведомлена о действиях подсистемы мониторинга? Как деактивируются параметры, связывающиеся пользователем или сетевыми системами?
Отображение	Отображаемые величины не были обновлены, но пользователь не осведомлен об этом. Записи отображения осуществляют на двух или более уровнях приоритета	Как пользователю сообщается о «замершем» дисплее? Видеоконтекст сохраняется перед приоритетным прерыванием?
Измерение	Неправильная синхронизация сбора данных или частота дискретизации	Подходит ли частота дискретизации для частотного содержимого сигнала? Величина измерения хранится в последовательных устройствах по всем программным слоям?
Интерфейсы (взаимодействия)		
Плохая передача данных	Функция передает значение в микролитрах, а драйвер ожидает значения в миллилитрах. Передается плохой указатель. Проходит указатель на энергозависимую память, и значения теряются до обработки	Каждая функция программного обеспечения проверяет прошедшие параметры? Язык программирования поддерживает более точную проверку типов? Программное обеспечение разрабатывается последовательными блоками для величин по всему пакету программ? Аргументы модифицируются в обрабатывающем слое более высокого приоритета?
Сеть	Программное обеспечение входит в бесконечный цикл, ожидания ответа от главного компьютера. МЕДИЦИНСКИМ ИЗДЕЛИЯМ в сети дают идентичные «имена», приводящие к потере данных. Организация сети обработки данных доминирует среди ПРОЦЕССОВ центрального процессора, отводя недостаточно ресурсов на БЕЗОПАСНОСТЬ или функции предназначенного применения	Было ли программное обеспечение разработано так, чтобы не реагировать на физические условия сети? Может ли дистанционная связь ухудшить работу системы, неоднократно посылая команды или фальсифицированные данные? Проверяет ли МЕДИЦИНСКОЕ ИЗДЕЛИЕ, что уже используется системное имя?

Продолжение таблицы В.1

Область функционирования программного обеспечения	Пример причин опасности	Задаваемые вопросы
Данные		
Медицинская информация	Система дает доступ к неправильным данным о пациенте, а отображение пользовательского интерфейса не представляет это очевидным. Система хранит данные о пациенте не в том архиве	Может ли произойти отображение множества независимых идентификаторов, чтобы закольцевать обнаружение путаницы? Могут ли быть включены критические идентификаторы с фактическими данными как двойная проверка?
Сообщения	Сообщение содержит неверные данные или определяет их в неверной последовательности или без единиц измерения	Какие сообщения будут использоваться в медицинских целях? Какова серьезность ВРЕДА, если данные неверны? Какова вероятность того, что врач не заметит проблемы?
Базы данных	Повреждение данных из-за побочных эффектов от отказов системного уровня или ПОНП	Как можно обнаружить повреждение данных до их использования? Можно ли это делать при каждом использовании, а не только при запуске?
Диагностика		
Принятие решений	Отображения обнаружения объекта, искажающего результаты исследования, подавляет отображение асистолии на дисплее.	Была ли тщательно рассмотрена иерархия определения тревоги, в том числе медицинским персоналом?
Преобразование данных	Арифметические ошибки точности приводят к неправильному результату. Алгоритм использует или отображает некорректные единицы	Какая арифметическая точность требуется? Как могут быть закодированы математические формулы, чтобы обеспечить надлежащую точность?
Автоматизированная профилактическая техническая поддержка	Базовая диагностика изменяет данные временно, но в то же время код приложения извлекает данные для фактического использования. Базовая диагностика мешает правильной синхронизации	Дополнительные ПРОЦЕССЫ заблокированы во время диагностики в соответствующее время? Диагностика заблокирована во время критических циклов?
БЕЗОПАСНОСТЬ		
Параметры настройки	Отсутствие или недостаточная защита доступа к критическим параметрам конфигурации или данным	Какие данные считаются критическими и не должны изменяться пользователями или должны требовать полномочий администратора, чтобы сделать это? Является ли аудиторская проверка необходимой?
Функциональный доступ	Отсутствие или недостаточная защита доступа к элементам управления терапией или к работе прибора	Должны ли операторы войти в систему под своим именем перед работой? Могут ли пациенты случайно управлять МЕДИЦИНСКИМ ИЗДЕЛИЕМ?
Интерфейс доступа	Отсутствие или недостаточная защита от данных и команд, предоставленных через коммуникационные интерфейсы или сеть	Что можно разрешить удалить? Стоит ли полагаться на предполагаемый дистанционный контроль системы или нет?

Окончание таблицы В.1

Область функционирования программного обеспечения	Пример причин опасности	Задаваемые вопросы
Исполнение		
Емкость/загрузка/время реагирования	Критическая синхронизация затрагивается при пиковых нагрузках. Последовательность транзакций/входных данных/выходных данных затрагивается, или данные теряются при пиковых нагрузках. Пиковые нагрузки системы затрагивают устройство управления	Во время пиковых нагрузок или когда пределы мощности достигнуты, данные или синхронизация будут потеряны или затронуты способами, которые нельзя обнаружить? Будут ли входные и выходные данные стоять в очереди в детерминированной последовательности при пиковых нагрузках? Были ли критическая функциональность и меры УПРАВЛЕНИЯ РИСКОМ проверены при таких напряженных условиях? Были ли осуществлены меры по УПРАВЛЕНИЮ РИСКОМ, чтобы обнаружить пределы? Могут ли устанавливаться прерывания для разделения критического момента, отделяющего одну функциональность от другой функциональности?

В дополнение к потенциальным причинам ОПАСНОСТЕЙ, связанным с функционированием программного обеспечения, показанным в таблице В.1, существуют некоторые типы программных причин, которые могут привести к побочным эффектам программного обеспечения, не связанным с тем программным обеспечением, в котором произошел отказ. Если определенный тип дефекта программного обеспечения может иметь непредсказуемое влияние на СВЯЗАННЫЕ С БЕЗОПАСНОСТЬЮ части программного обеспечения, эта возможность должна быть идентифицирована и должны быть разработаны стратегии УПРАВЛЕНИЯ РИСКОМ, а также определены меры по УПРАВЛЕНИЮ РИСКОМ.

Таблица В.2 идентифицирует примеры этих возможных программных причин ОПАСНОСТЕЙ и некоторые возможные меры по УПРАВЛЕНИЮ РИСКОМ. Требования, основанные на испытаниях системы, часто неэффективны при идентификации этих типов программных причин или ВЕРИФИКАЦИИ связанных с ними мер по УПРАВЛЕНИЮ РИСКОМ и результативности мер по УПРАВЛЕНИЮ РИСКОМ. Колонки справа в таблице дают представление о типе статического или динамического метода(ов) ВЕРИФИКАЦИИ, которые могут быть подходящими для каждого примера. Таблица В.3 предоставляет примеры для статического или динамического метода(ов) ВЕРИФИКАЦИИ.

Таблица В.2 — Примеры программных причин, которые могут вызвать побочные эффекты

Программные причины	Типы ВЕРИФИКАЦИИ	Анализ: Статический (С), Динамический (Д), Временной (В)		
		Испытание (модульное, интеграционное)		
		Проверка		
	Меры по УПРАВЛЕНИЮ РИСКОМ			
Вычислительные				
Деление на ноль	Использование ловушек отслеживания ошибок времени запуска, защищенное кодирование	x	x	Д
Выход результата за пределы диапазона переменной	Проверки диапазона, использование переменных с плавающей точкой	x	x	Д
Округление переменных с плавающей точкой	Применение четких алгоритмов	x	—	—
Некорректная проверка пределов диапазона переменных	Защищенное кодирование	x	x	С

Продолжение таблицы В.2

Программные причины	Типы ВЕРИФИКАЦИИ: Анализ: Статический (С), Динамический (Д), Временной (В); Испытание (модульное, интеграционное); Проверка; Меры по УПРАВЛЕНИЮ РИСКОМ			
Ошибка сравнения рациональной переменной с переменной целого типа (ОВО)	Защищенное кодирование	x	x	—
Связанные с аппаратными средствами				
Работа с ЭСППЗУ: длительное время ответа, выработка ресурса	Применение специальных режимов (поблочная запись), запись только при изменении данных, запись в кэш и обновление ЭСППЗУ только при сбоях питания	x	—	В
ЦП или аппаратные причины	Проверка ЦП при включении питания, СКС-тест исполняемого кода, тестирование оперативной памяти, проверка встроенного таймера и часов, проверка энергозависимой памяти, проверка времен ожидания ответов аппаратных модулей, проверка датчиков короткого замыкания, проверка датчиков по ответу на заданный сигнал	—	x	—
Шумы	Фильтрация цифровых и аналоговых входных сигналов, проверка процедур обработки прерывания для всех типов прерываний — как используемых, так и неиспользуемых	—	—	—
АНОМАЛИИ периферийного аппаратного обеспечения	Задержки запуска конвертации аналогового сигнала в цифровой или из цифрового в аналоговый, проверка времени ответа и других требований к интерфейсу, проверка причинности	x	—	В
Временные				
Условия работы с ресурсами	Идентификация и защита (блокировка использования) совместно с используемыми ресурсами в процессе их обновления, реализация одного нециклического ПРОЦЕССА для оценки совместно используемых ресурсов, анализ совместно используемых ресурсов	—	—	С
Пропущенные сроки ожидания ответа	Установка требований к времени ожидания ответа, использование правильных алгоритмов реального времени, структурное исключение инверсии приоритета ожидания ответа, избегание неопределенных времен ожидания, проверка выполнения всех допусков по временам ожидания для завершенного кода. Примечание — Неопределенные времена ожидания включают рекурсивные процедуры, ожидание ответа от аппаратного обеспечения, динамическое резервирование памяти, использование виртуальной памяти (с перебросом страниц с жесткого диска или на него)	—	—	В
Пропущенные прерывания	Использование простой АРХИТЕКТУРЫ (с наименьшим количеством приоритетов), использование коротких и быстрых процедур обработки прерываний, неиспользование или использование краткой блокировки прерываний, использование правильного кода в реальном времени	—	—	В
Высокая временная неравномерность при передаче данных	Использование коротких и быстрых процедур обработки прерываний, избегание неопределенных времен ожидания, использование правильного кодирования в реальном времени, обновление статуса всех прерываний на передачу данных при начале выполнения периодических ЗАДАЧ или процедуры обработки прерывания высокого приоритета	—	—	В

Продолжение таблицы В.2

Программные причины	Типы ВЕРИФИКАЦИИ Анализ: Статический (С), Динамический (Д), Временной (В) Испытание (модульное, интеграционное) Проверка Меры по УПРАВЛЕНИЮ РИСКОМ			
Таймеры	Определение времен сброса таймеров и установка таймера соответственно, определение максимального значения времени для таймера, не ведущего к ОПАСНОЙ СИТУАЦИИ	—	—	В
Режим работы				
Непредусмотренное завершение программы	Использование ловушек для отслеживания выходов из программы, ошибок времени запуска, правильная конструкция таймеров, проверка устойчивости к входным данным, самотестирование при включении. Удалить весь отладочный код и нерабочий код из релиза, обеспечивая невозможность непроизвольного запуска специальных режимов (обслуживание, изготовление).	—	—	Д
Проблемы при отключении / повторном включении / многократном включении	Самотестирование при включении для ЦП, оперативной памяти, таймера и часов, энергозависимой памяти, СКС-тест исполняемого кода, тестирование периферии и т. п. Соответствующая разработка в отношении статуса выполнения программы, инициализация переменных, усредняемых по времени, реинициализация периферийных устройств, инициализация статуса СИСТЕМЫ из энергозависимой памяти, внешний монитор напряжения или сброса питания	—	—	—
АНОМАЛИИ запуска / завершения работы	Проверки при включении (см. выше), правильная инициализация периферии и данных, правильное использование энергозависимой памяти, правильная разработка статуса выполнения программы	—	—	—
Вход/выход в режиме низкого напряжения питания	Правильная обработка прерываний	x	—	В
Ошибки в данных				
Искажение данных	Дублирование оперативной памяти, СКС или проверочные суммы для блоков данных. Работать с данными только через специальные функции, минимизировать глобальные данные, сохранять простую структуру данных, учитывать, как данные организуются в структуры компьютером, избегать смешения данных (также далее см. «ошибочные указатели» и «промежуточные данные»)	—	—	С
Конфликты в работы с ресурсами	Анализ совместно используемых ресурсов (также ранее см. «условия работы с ресурсами»)	—	—	—
Ошибочные указатели	Защищенное кодирование: проверка целостности данных перед снятием указателя, использование языков с сильными типами данных, минимизировать применение указателей, избегать смешения указателей	x	x	С
Ошибки при переводе данных: смешение типов, масштабирование	Избегание смешения типов, использование форматов с плавающей точкой	x	x	С
Некорректная инициализация	Преинициализирование переменных усредняемых по времени, обнуление всей памяти при запуске	x	x	С
Усредняемые данные за пределами диапазона	Проверка того, что достаточное число отсчетов взято до вычисления среднего (особенно при запуске) или преинициализирование среднего к известному (или последнему) корректному значению	x	x	С

Продолжение таблицы В.2

Программные причины	Типы ВЕРИФИКАЦИИ Меры по УПРАВЛЕНИЮ РИСКОМ	Анализ: Статический (С), Динамический (Д), Временной (В) Испытание (модульное, интеграционное) Проверка			
Переходящие данные	Проверка правильности переходов		x	x	—
Изменяемые данные	Проверка того, что изменяемый тип хранения данных использован для всех данных, изменяемых аппаратным обеспечением, процедурами обработки прерываний или другими приоритетными ЗАДАЧАМИ		x	—	С
Непредусмотренное дублирование	Считывание данных с частотой в 2 и более раз большей, чем наибольшая компонента сигнала (критерий Найквиста) ограничить полосу пропускания сигнала		—	—	—
Использование промежуточных данных Примечание — Последовательность вычислений никогда не должна выполняться с глобальной (или с общей) переменной, если вычисление может быть прервано или сброшено. Вместо этого следует выполнять все вычисления с временной переменной и обновлять значение глобальной переменной одной непрерываемой командой	Проверка того, что все данные, которые синхронизированы во времени, обновляются одновременно с проведением анализа совместно используемых ресурсов		x	—	—
Проблемы интерфейса					
Несвоевременное обновление изображения на дисплее	Постоянное обновление изображения вместо обновления при наступлении событий		—	—	—
Человеческий фактор: НЕПРАВИЛЬНОЕ ПРИМЕНЕНИЕ	Вести файлы истории работы для восстановления последовательности действий, использовать контекстную подсказку, использовать простой интерфейс		—	—	—
Сетевые проблемы, например, многопользовательский режим	Тестирование нагрузки со стороны сети		—	—	В
Ошибки конфигурации программного или аппаратного обеспечения или некорректные драйверы	Использовать ПРОЦЕСС разработки программного обеспечения, использовать средства управления конфигурацией		—	—	—
Нерабочие патчи или обновления	СКС-тест исполняемого кода и проверка ВЕРСИИ при запуске, проверка версий протоколов и сроков годности кода		—	—	—
Отказы ПОНП: зависание, отсутствие ответа, превышение времени ожидания прерываний и т. п.	Проанализировать файл истории ошибок ПОНП, использование Робастного проектирования (например, поставить таймеры передела времени ожидания на все блокирующие прерывания), заблокировать доступ к страницам памяти, которые часто используются процедурами обработки прерываний, использовать только необходимые элементы ПОНП, удалив все остальные		x	—	В
Вирус	Применение антивируса		—	—	—

Окончание таблицы В.2

Программные причины	Типы ВЕРИФИКАЦИИ Анализ: Статический (С), Динамический (Д), Временной (В) Испытание (модульное, интеграционное) Проверка Меры по УПРАВЛЕНИЮ РИСКОМ			
Несовместимость веббраузера	Интегративное тестирование версий при запуске, проверка совместимости	—	—	—
Разное				
Утечки памяти	Избегание динамического выделения массивов памяти	x	x	Д
Блокировка СИСТЕМЫ	Использование простых стратегий блокирования ресурсов (ПРОЦЕСС может заблокировать только один ресурс за один раз), анализ блокировки	—	—	С
Повторное применение	Подтверждение, что все функции, включая библиотеки третьих сторон, которые вызываются прерываниями (или множественными ЗАДАЧАМИ с разными приоритетами), предусматривают повторный запуск	—	—	Д
Переполнение стека	Использование флагов ошибок времени запуска стеков, флагов переполнения, анализ стеков	—	—	С
Логические ошибки/синтаксис	Использование средств анализа кода (таких как Lint) и/или максимального уровня предупреждений при компилировании. Использование удвоенного РАЗНООБРАЗИЯ и перекрестного тестирования в критических контрольных точках	x	x	С
Бесконечные циклы	Использование счетчиков циклов, таймеров циклов	x	x	—
Порча кода	СКС-тест исполняемого кода при запуске и вызове	—	—	—
Нерабочий код	Установка проверок ошибки, которая будет выдавать сообщение и закрывать программу, в случае если нерабочий код не был удален и начинает выполняться (для специального программного обеспечения или для компонентов стороннего программного обеспечения)	—	—	Д
Некорректно скомпилированный код	Подтверждение, что условия компиляции изменяются правильно и только при необходимости	x	—	—
Непредусмотренные побочные эффекты на макроуровне	Использование круглых скобок для всех макропараметров	—	—	С
Истощение ресурсов	Использование анализа стека, очереди, времен выполнения	—	—	В
Некорректный приоритет тревожных сигналов и опасностей	Стресс-тестирование	—	—	—
Несанкционированные применения (извлечение части кода, запуск кода непредусмотренным образом и т. п.)	Анализ разработки, анализ требований к коду, использование матриц для прослеживания	—	—	—
Некорректная последовательность операций/приоритетов	Разбивка кода на элементы, линейная структура кода	—	—	—
Состояние БЕЗОПАСНОСТИ	Использование независимых наблюдателей	—	—	—

Существует много методов с различной ресурсоемкостью для обеспечения результативности (работоспособности) МЕР ПО УПРАВЛЕНИЮ РИСКОМ. Ни один отдельный метод не является исчерпывающим. Некоторые из таких методов приведены далее в таблице В.3.

Таблица В.3 — Методы, способствующие обеспечению результативности МЕР ПО УПРАВЛЕНИЮ РИСКОМ

Статический анализ	Динамический анализ	Моделирование
Просмотр кода	Функциональное тестирование	Моделирование рабочего окружения
Анализ разработки	Временные тесты и тесты памяти	Симулирование во времени
Анализ работы скрытых частей	Анализ граничных значений	Сценарии применения/действий пользователя
—	Тестирование работоспособности	—
—	Стресс-тестирование	—
—	Статистическое тестирование	—
—	Прогнозирование ошибок	—
—	Тестирование пути выполнения кода	—
—	Тестирование применения кода	—
—	Кластерное тестирование	—

Вместо стремления сосредоточиться на испытаниях, связанных с функционалом ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, следует сфокусироваться исключительно на том, чтобы СВЯЗАННОЕ С БЕЗОПАСНОСТЬЮ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ было испытано в достаточном, адекватном диапазоне условий и состояний. Испытания должны состоять из разнообразных типов тестирования (например, стресс-тест, граничные значения, временные рамки, отказ электропитания, неисправности, отказ ПОНП и т. д.).

Приложение С
(справочное)

Потенциальные ошибки, связанные с программным обеспечением

В таблице С.1 перечислены потенциальные ошибки, связанные с программным обеспечением, которых следует избегать в работе по МЕНЕДЖМЕНТУ РИСКА (по пунктам ИСО 14971) и в работе по управлению ЖИЗНЕННЫМ ЦИКЛОМ программного обеспечения (по пунктам МЭК 62304).

Таблица С.1 — Потенциальные ошибки, связанные с программным обеспечением, которых следует избегать

ИСО 14971 Раздел 4: АНАЛИЗ РИСКА
- Применение неправдоподобно низких значений вероятности для программных отказов, что ведет к неправдоподобно низким значениям при определении РИСКА и несоответствующим мерам по УПРАВЛЕНИЮ РИСКОМ; - Расширение программных функций без выполнения АНАЛИЗА РИСКА для оценки того, возникли ли новые ОПАСНОСТИ или ОПАСНЫЕ СИТУАЦИИ, или причины в МЕДИЦИНСКОМ ИЗДЕЛИИ, или не нарушена ли результативность существующих мер по УПРАВЛЕНИЮ РИСКОМ (как при исходной разработке, так и после релиза в процессе технического сопровождения); - ПРОЦЕСС АНАЛИЗА РИСКА МЕДИЦИНСКОГО ИЗДЕЛИЯ установлен только на уровне СИСТЕМЫ и аппаратной части и не включает рассмотрение связи АНАЛИЗА РИСКА с программным обеспечением и не требует отдельного рассмотрения АНОМАЛИЙ программного обеспечения как потенциальных источников ОПАСНОСТЕЙ и ОПАСНЫХ СИТУАЦИЙ; Масштаб АНАЛИЗА РИСКА и процедуры разработки ЖИЗНЕННОГО ЦИКЛА программного обеспечения несоизмеримы с потенциальным ВРЕДОМ от МЕДИЦИНСКОГО ИЗДЕЛИЯ.
ИСО 14971 Пункт 4.1: ПРОЦЕСС АНАЛИЗА РИСКА
- ПРОЦЕСС АНАЛИЗ РИСКА определен только на уровне СИСТЕМЫ и аппаратной части. Программное обеспечение рассматривается только в случае, когда оно задействовано в мерах по УПРАВЛЕНИЮ РИСКОМ для аппаратных отказов; - Программное обеспечение рассматривается как часть АНАЛИЗА РИСКА только на последних этапах ЖИЗНЕННОГО ЦИКЛА разработки продукции.
ИСО 14971 Пункт 4.2: Идентификация ПРЕДУСМОТРЕННОГО ПРИМЕНЕНИЯ
- Рассмотрение только подмножества пользовательских интерфейсов/потенциальных платформ для компьютерных СИСТЕМ; - Неучет эволюции платформы или требований к установке патчей по БЕЗОПАСНОСТИ и других патчей ПОНП; - Некорректное рассмотрение случаев применения не по назначению и ошибок пользователя, которые ведут к потенциальным ОПАСНОСТЯМ, что в результате приводит к тому, что соответствующие меры по УПРАВЛЕНИЮ РИСКОМ неидентифицированы.
ИСО 14971 Пункт 4.3: Идентификация ОПАСНОСТЕЙ
a) Применение FMEA и FTA методологий, как если бы они были бы полностью достаточны для корректного МЕНЕДЖМЕНТА РИСКА; b) Применение FMEA и FTA методологий раздельно для аппаратной и программной частей; c) Игнорирование целого класса ОПАСНОСТЕЙ, таких как: 1) программные ошибки, которые имеют непредсказуемые эффекты; 2) ошибки в программной логике, которая применена как мера по УПРАВЛЕНИЮ РИСКОМ для аппаратных отказов; 3) ошибки в программной логике для реализации предусмотренного применения МЕДИЦИНСКОГО ИЗДЕЛИЯ (такие как алгоритмы для вычисления результатов); 4) отказы программных платформ — операционных СИСТЕМ, библиотек, ПОНП; 5) отказы компонентов компьютера и периферийных устройств; 6) отказы интерфейсов передачи данных; 7) человеческие ошибки. d) Проведение идентификации причин в предположении, что: 1) АНОМАЛИИ программного обеспечения будут влиять на функции только отдельных компонентов и не будут оказывать влияние на другие ПРОГРАММНЫЕ ЭЛЕМЕНТЫ или данные; 2) программное обеспечение будет работать корректно;

Продолжение таблицы С.1

3) потенциальные отказы программного обеспечения слишком многочисленны и непредсказуемы для идентификации, определения или УПРАВЛЕНИЯ РИСКОМ; 4) всегда достаточно применить меры по УПРАВЛЕНИЮ РИСКОМ в начале или в конце цепи программных событий, ведущих к ОПАСНЫМ СИТУАЦИЯМ.
ИСО 14971 Пункт 4.4: Определение РИСКА
- Предположение, что концепция единичного отказа применима к разработке программного обеспечения и к последовательностям событий; - Предположение, что тестирование, которое может быть и неполным, уменьшает вероятность определенного отказа до нуля; - Предположение, основанное на функционале изделия, о том, что некоторые ПРОГРАММНЫЕ ЭЛЕМЕНТЫ не связаны с БЕЗОПАСНОСТЬЮ, без учета возможных неожиданных побочных эффектов; - Задание значений ТЯЖЕСТИ ВРЕДА без достаточных клинических знаний и без привлечения тех, кто имеет представление о клинических аспектах (человеческих факторах) воздействия ОПАСНОСТЕЙ на всех потенциальных пользователей и групп пациентов; - Задание низких значений тяжести в предположении, что врачи обнаружат отказ, или на основе неверной информации; - Задание низких значений тяжести в предположении, что все пользователи будут следовать указаниям в маркировке МЕДИЦИНСКОГО ИЗДЕЛИЯ и в инструкции пользователя и не совершат непредвиденных ошибок; - При задании ТЯЖЕСТИ ВРЕДА предположение о том, что запланированы определенные меры по УПРАВЛЕНИЮ РИСКОМ для ОПАСНОСТЕЙ. Если предположения ошибочны, то низкая начальная ТЯЖЕСТЬ ВРЕДА может привести к некорректному УПРАВЛЕНИЮ РИСКОМ впоследствии; - Применение концепции потенциального ВРЕДА для пациента только для прямых ущербов без рассмотрения прочих целей применения информации, которую программное обеспечение выдает пользователю, задержек в лечении и других факторов, относящихся к результативности и основным эксплуатационным характеристикам МЕДИЦИНСКОГО ИЗДЕЛИЯ; - Предположение о том, что врач всегда будет перепроверять информацию, выданную программным обеспечением, и сможет определить неверную информацию, использование которой приведет к тому, что событиям придается низкая ТЯЖЕСТЬ ВРЕДА, что ведет к неприменению других мер по УПРАВЛЕНИЮ РИСКОМ.
ИСО 14971 Раздел 5: ОЦЕНИВАНИЕ РИСКА
- Субъективное задание вероятности появления АНОМАЛИИ программного обеспечения для определения того, требуются ли меры по УПРАВЛЕНИЮ РИСКОМ; - Исключение ОПАСНОСТИ из области программного обеспечения в связи с особенностями аппаратного обеспечения и при этом изменение аппаратного обеспечения впоследствии таким образом, что программное обеспечение является влияющим фактором, для которого не предусмотрены дополнительные меры по УПРАВЛЕНИЮ РИСКОМ; - Неучет потенциальной АНОМАЛИИ программного обеспечения как фактора ОПАСНОСТИ, поскольку предполагается, что программное обеспечение будет работать правильно или тестирование выявит все АНОМАЛИИ.
ИСО 14971 Пункт 6.3: Выполнение мер по УПРАВЛЕНИЮ РИСКОМ
- Меры по УПРАВЛЕНИЮ РИСКОМ верифицируются при нормальных или стандартизированных условиях, не отражающих широкого диапазона аномальных и стрессовых условий; - Программное обеспечение или данные, использованные при реализации мер по УПРАВЛЕНИЮ РИСКОМ, включены в компоненты или месторасположения, которые легкодоступны для другого программного обеспечения, что повышает риск побочных эффектов; - Меры по УПРАВЛЕНИЮ РИСКОМ верифицированы только на одной операционной системе или программной платформе; - Некоторые меры по УПРАВЛЕНИЮ РИСКОМ не верифицированы в связи со сложностью моделирования их возникновения (например, отказ памяти, конфликт при доступе к ресурсам, порча данных, переполнение стека); - Предположение о том, что все относящиеся к БЕЗОПАСНОСТИ АНОМАЛИИ будут обнаружены при разработке и что тестирование гарантирует корректную работу; - Реализация мер по УПРАВЛЕНИЮ РИСКОМ, которые делают программное обеспечение значительно более сложным. Сложность увеличивает вероятность появления дополнительных АНОМАЛИЙ и причин новых ОПАСНОСТЕЙ.

Продолжение таблицы С.1

ИСО 14971 Раздел 9: ПОСТПРОИЗВОДСТВЕННАЯ информация
- Игнорирование потенциально опасных контекстных событий от ОШИБОК ПОЛЬЗОВАТЕЛЯ, когда дополнительные меры по УПРАВЛЕНИЮ РИСКОМ могли бы быть применены; - Предположение, что начальные значения вероятности и ТЯЖЕСТИ являются правильными, без оценивания информации о применении изделия; - Упущение, что МЕДИЦИНСКОЕ ИЗДЕЛИЕ может быть применено не по назначению так, что меры по УПРАВЛЕНИЮ РИСКОМ будут неэффективны. Например, изделие *in-vitro* для тестирования на ВИЧ было предназначено для индивидуального применения, а используется для скрининга станции по переливанию крови.
МЭК 62304 Пункт 5.1 Планирование разработки программного обеспечения
- Деятельность по МЕНЕДЖМЕНТУ РИСКА не включена в планы программного обеспечения и в ПРОЦЕССЫ ЖИЗНЕННОГО ЦИКЛА; - Деятельность по МЕНЕДЖМЕНТУ РИСКА программного обеспечения не связана с деятельностью по МЕНЕДЖМЕНТУ РИСКА МЕДИЦИНСКОГО ИЗДЕЛИЯ; - ОЦЕНИВАНИЕ РИСКА программного обеспечения проводится только для одной стадии ЖИЗНЕННОГО ЦИКЛА; - Разработчики и испытатели программного обеспечения не обучены или не имеют опыта в МЕНЕДЖМЕНТЕ РИСКА; - МЕНЕДЖМЕНТ РИСКА программного обеспечения считается полностью включенным в деятельность по общему МЕНЕДЖМЕНТУ РИСКА; - РИСКИ программного обеспечения управляются неаккуратно; - Решения в отношении ПРОСЛЕЖИВАЕМОСТИ и БЕЗОПАСНОСТИ не установлены.
МЭК 62304 Аспекты программного обеспечения неизвестного происхождения (ПОНП)
- Пропуск разработки внутренних мер по УПРАВЛЕНИЮ РИСКОМ из-за отсутствия ОЦЕНИВАНИЯ РИСКА и управления при определении АРХИТЕКТУРЫ программного обеспечения; - Предположение о том, что тестирование сделает неэффективные АРХИТЕКТУРЫ достаточно безопасными; - Неполная идентификация аспектов БЕЗОПАСНОСТИ АРХИТЕКТУРЫ, что ведет к неизвестным РИСКАМ в отношении БЕЗОПАСНОСТИ в случаях, когда элементы архитектуры заменяют или удаляют.
МЭК 62304 Пункт 5.4 Детальная проработка программного обеспечения
- Концентрация на обработке только нормальных ситуаций (случаев) и предположение о том, что интерфейсы и параметры передачи между компонентами будут корректны, что исключает многоуровневую проверку устойчивости к ошибкам; - При «мозговом штурме» при проработке, детализации и последующих анализах нерассмотрение возможных отказов программного обеспечения, которые могут привести к ОПАСНОСТЯМ и ОПАСНЫМ СИТУАЦИЯМ и соответствующим мерам по УПРАВЛЕНИЮ РИСКОМ; - Игнорирование причин программных отказов (см. приложение В) в деятельности по МЕНЕДЖМЕНТУ РИСКА.
МЭК 62304 Пункт 5.5 Реализация и ВЕРИФИКАЦИЯ ПРОГРАММНОГО МОДУЛЯ
- Предположение о том, что наилучшие практики кодирования и/или ПРОЦЕССА тестирования, программные инструменты или персонал могут компенсировать ущербную, внутренне небезопасную или излишне сложную структуру; - Привлечение неопытных разработчиков для написания критических частей кода; - Отсутствие установления и требования применения специальных защищенных практик программирования; - Использование исключительно динамического тестирования без выполнения проверок кода и статического анализа, особенно для критических компонентов; - Отступление от хода разработки без понимания соотношения между требованиями к разработке и МЕНЕДЖМЕНТОМ РИСКА; - Однократное тестирование критических компонентов на ранних стадиях ПРОЦЕССА разработки без возвращения к ним в регрессионном тестировании; - Концентрация испытаний исключительно на динамическом тестировании, на модели черного ящика, на уровне СИСТЕМЫ и невыполнение статической и динамической ВЕРИФИКАЦИИ по модели белого ящика.

Окончание таблицы С.1

МЭК 62304 Пункты 5.6—5.7 Программная интеграция и испытания в отношении интеграции и испытание СИСТЕМЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
- Неприменение информации по ОЦЕНИВАНИЮ РИСКОВ при планировании тестирования и обучения испытателей; - Зависимость от испытания как меры по УПРАВЛЕНИЮ РИСКОМ — даже в том случае, когда 100 % тестирование невозможно; - Невоспроизведение случаев отказа СИСТЕМ и программного обеспечения при тестировании для верификации мер по УПРАВЛЕНИЮ РИСКОМ; - Применение инструментов для автоматического тестирования без должной квалификации и контроля и использование лишь результатов такого тестирования; - Неспособность корректно проанализировать код для определения АНОМАЛИЙ, которые испытатель не сможет обнаружить.
МЭК 62304 Пункт 5.8 Выпуск программного обеспечения
- Невозможность привести ВЕРСИЮ документации релиза в соответствие выпущенным программным обеспечением может ввести в заблуждение группу разработчиков и испытателей в отношении будущих релизов. Неаккуратные документирование и ПРОСЛЕЖИВАЕМОСТЬ могут привести к потере связей, пропуску ОПАСНОСТЕЙ и их причин, потере мер по УПРАВЛЕНИЮ РИСКОМ или невозможности корректно верифицировать ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, влияющее на БЕЗОПАСНОСТЬ; - Невозможность привлечь персонал с клиническими знаниями для оценки остаточных АНОМАЛИЙ; - Оценка значимости оставшихся АНОМАЛИЙ, основанная исключительно на найденных функциональных симптомах, без полного анализа главных причин для определения потенциальных побочных эффектов в различных ситуациях; - Неучет того факта, что как только третья сторона перестанет поддерживать определенную версию ПОНП, эта версия не будет более доступна для расследования отказов и коррекций; - Потеря определенной ВЕРСИИ программного обеспечения по причине того, что некоторые инструменты или ВЕРСИИ инструментов (например, компилятор) не были включены в архив; - Технический файл МЕДИЦИНСКОГО ИЗДЕЛИЯ может храниться дольше, чем жизненный цикл архиватора. Если ИЗГОТОВИТЕЛЬ заменяет старый архиватор на новый, он должен предусмотреть перенос всех задействованных частей программного обеспечения.
МЭК 62304 Пункт 6.1 Разработка плана технической поддержки программного обеспечения
- Разработка ПРОЦЕССА технической поддержки не имеет четкого подхода к МЕНЕДЖМЕНТУ РИСКА изменений; - Установление МЕНЕДЖМЕНТА РИСКА для изменений, который рассматривает только функциональные аспекты изменения и не рассматривает компоненты, участвующие в изменении, и связанные с ними РИСКИ.
МЭК 62304 Пункты 6.2–6.3 Анализ проблем и модификаций и их реализация
- Предположение, что небольшое функциональное изменение не повлияет на БЕЗОПАСНОСТЬ; - Расширение области применения МЕДИЦИНСКОГО ИЗДЕЛИЯ на новые группы пациентов, новые заболевания, новые типы пользователей (например, медицинские сестры вместо хирургов) или на новые программные платформы без пересмотра существующих мер по УПРАВЛЕНИЮ РИСКОМ и пригодности пользовательского интерфейса; - Установка приоритетности разрешения проблем на основе симптомов и контекста, без определения главных причин и потенциальных побочных эффектов; - Программное обеспечение для немедицинских применений (например, генерация счета) содержит клинические данные, которые впоследствии распространяются для клинических целей без соответствующего МЕНЕДЖМЕНТА РИСКА.

Приложение D
(справочное)

Матрица жизненный цикл/менеджмент риска

В таблице D.1 перечислены этапы разработки по МЭК 62304 и соответствующие этапы МЕНЕДЖМЕНТА РИСКА программного обеспечения. Следует отметить, что данная таблица не предполагает четкого последовательного представления этапов ЖИЗНЕННОГО ЦИКЛА.

Таблица D.1 — Матрица ЖИЗНЕННЫЙ ЦИКЛ/МЕНЕДЖМЕНТ РИСКА

МЭК 62304 Требования к ЖИЗНЕННОМУ ЦИКЛУ	ИСО 14971		
	4 АНАЛИЗ РИСКА	5 ОЦЕНИВАНИЕ РИСКА	6 УПРАВЛЕНИЕ РИСКОМ
5 ПРОЦЕСС разработки программного обеспечения			
5.1 Планирование разработки программного обеспечения	Планирование МЕНЕДЖМЕНТА РИСКА (3.4 из ИСО 14971) Спланировать и документировать: a) объем запланированной деятельности по МЕНЕДЖМЕНТУ РИСКА, в том числе идентификацию и описание МЕДИЦИНСКОГО ИЗДЕЛИЯ и стадий его ЖИЗНЕННОГО ЦИКЛА, к которым применим каждый элемент плана; b) распределение ответственности и полномочий; c) требования к анализу деятельности по МЕНЕДЖМЕНТУ РИСКА; d) критерии допустимости риска, основанные на политике ИЗГОТОВИТЕЛЯ по определению допустимого РИСКА, в том числе критерии допустимости РИСКА в тех случаях, когда вероятность причинения ВРЕДА не может быть определена; e) деятельность по ВЕРИФИКАЦИИ; f) деятельность по сбору и анализу информации, относящейся к МЕНЕДЖМЕНТУ РИСКА, на производственной и ПОСТПРОИЗВОДСТВЕННОЙ стадиях.		
5.2 Анализ требований к программмному обеспечению	- Проанализировать предусмотренное применение и обоснованно прогнозируемое неправильное применение, а также пользователей для определения ОПАСНОСТЕЙ; - Определить известные и возможные ОПАСНОСТИ в отношении врачей, пациентов, обслуживающего персонала и прочих людей, которые имеют контакт с МЕДИЦИНСКИМ ИЗДЕЛИЕМ; - Учесть этапы эксплуатации, такие как монтаж и сборка, обучение, применение, модернизация, техническое обслуживание; - Определить последовательности и комбинации событий, которые могут привести к ОПАСНЫМ СИТУАЦИЯМ; - Определить РИСКИ для каждой ОПАСНОЙ СИТУАЦИИ, учитывая ТЯЖЕСТЬ ВРЕДА от возможных последствий; - 4.3 Классифицировать программное обеспечение по БЕЗОПАСНОСТИ (требования МЭК 62304.	- Принять решение о том, требуется ли уменьшение РИСКА; - Определить, будут ли достаточны меры по УПРАВЛЕНИЮ РИСКОМ программного обеспечения, или требуются меры по УПРАВЛЕНИЮ РИСКОМ аппаратного обеспечения.	- Определить меры по УПРАВЛЕНИЮ РИСКОМ программного обеспечения для идентифицированных РИСКОВ (например, по причине отказов аппаратного обеспечения или ошибок пользователя); - Начальное определение мер по УПРАВЛЕНИЮ РИСКОМ для отказов программного обеспечения, которые влияют на конструкцию (например, внутренне безопасные конструкции или защитные меры); - Идентифицировать программное обеспечение для улучшения дифференциации, уменьшения ТЯЖЕСТИ ВРЕДА и/(или) снижения вероятности ОПАСНОЙ СИТУАЦИИ; - Проанализировать меры по УПРАВЛЕНИЮ РИСКОМ на наличие новых ОПАСНОСТЕЙ.

Продолжение таблицы D.1

МЭК 62304 Требования к ЖИЗНЕННОМУ ЦИКЛУ	ИСО 14971		
	4 АНАЛИЗ РИСКА	5 ОЦЕНИВАНИЕ РИСКА	6 УПРАВЛЕНИЕ РИСКОМ
5.3 Разработка архитектуры программного обеспечения	- Определить критические данные, компоненты и классы дефектов, которые могут привести к ОПАСНОСТЯМ. Особое внимание уделить программным причинам; - Определить ассоциированные причины; - Определить интерфейсы; что и когда передается; - Оценить критерии успешной работы и ограничения; - Классифицировать программное обеспечение по БЕЗОПАСНОСТИ (требования МЭК 62304).	- Провести переоценку приемлемости роли, отведенной программному обеспечению в контексте РИСКА; - Оценить приемлемость того, что мера по управлению риском будет подвергаться воздействию со стороны функционала, не связанного с БЕЗОПАСНОСТЬЮ.	- Определить меры по УПРАВЛЕНИЮ РИСКОМ на уровне архитектуры для того, чтобы выделить критические компоненты и предупредить или обнаружить специфические программные причины ОПАСНОСТЕЙ; - Уделить особое внимание обеспечению соответствующей ИЗБЫТОЧНОСТИ; - Определить ОПАСНОСТИ, связанные с ИЗБЫТОЧНОСТЬЮ; - Определить общие методы по распознаванию и управлению.
5.4 Детальная проработка программного обеспечения	- Определит дополнительные потенциальные причины ОПАСНОСТЕЙ; - Предположить наличие ошибок в данных, в коде и в передаче данных; - Предположить отказы аппаратного обеспечения.	- Провести повторную оценку корректности мер по УПРАВЛЕНИЮ РИСКОМ; - Определить границы между кодом, связанным с БЕЗОПАСНОСТЬЮ, и не связанным с БЕЗОПАСНОСТЬЮ кодом.	- Внедрить специальные меры по УПРАВЛЕНИЮ РИСКОМ и защищенные практики проектирования/программирования; - Завершить анализ ПРОСЛЕЖИВАЕМОСТИ и целостности, чтобы убедиться в выполнении мер по УПРАВЛЕНИЮ РИСКОМ; - Определить, был ли реализован какой-либо не указанный ранее функционал.
5.5 Реализация и ВЕРИФИКАЦИЯ ПРОГРАММНОГО МОДУЛЯ	- Определить дополнительные потенциальные причины ОПАСНОСТЕЙ; - Оценить отказ каждого теста на подобных реализациях кода.	- Повторно оценить корректность мер по УПРАВЛЕНИЮ РИСКОМ, проверяя меры по УПРАВЛЕНИЮ РИСКОМ во всем диапазоне условий и тестируя на типичных пользователях в типичных условиях.	- Верифицировать меры по УПРАВЛЕНИЮ РИСКОМ во всем диапазоне условий и рабочих платформ; - Провести регрессионное тестирование мер по УПРАВЛЕНИЮ РИСКОМ перед окончательным релизом; - Завершить анализ ПРОСЛЕЖИВАЕМОСТИ и целостности, чтобы убедиться в том, что все меры по УПРАВЛЕНИЮ РИСКОМ реализованы и протестированы.

Окончание таблицы D.1

МЭК 62304 Требования к ЖИЗНЕННОМУ ЦИКЛУ	ИСО 14971		
	4 АНАЛИЗ РИСКА	5 ОЦЕНИВАНИЕ РИСКА	6 УПРАВЛЕНИЕ РИСКОМ
5.6 Интеграция программного обеспечения и интеграционное тестирование	—	—	- Регрессионное тестирование мер по УПРАВЛЕНИЮ РИСКОМ перед окончательным релизом; - Завершить анализ ПРОСЛЕЖИВАЕМОСТИ и целостности, чтобы убедиться, что все меры по УПРАВЛЕНИЮ РИСКОМ реализованы и протестированы.
5.7 Тестирование на уровне СИСТЕМЫ	—	—	- Регрессионное тестирование мер по УПРАВЛЕНИЮ РИСКОМ перед окончательным релизом; - Завершить анализ ПРОСЛЕЖИВАЕМОСТИ и целостности, чтобы убедиться в том, что все меры по УПРАВЛЕНИЮ РИСКОМ реализованы и протестированы.
5.8 Выпуск программного обеспечения	- Определить план управления конфигурацией, включая конфигурацию элементов и взаимозависимости.	—	Верифицировать, что корректные ВЕРСИИ специального и ПОНП программного обеспечения включены в релиз; Верифицировать, что условия компиляции учтены в управлении конфигурацией.
6 процесс технической поддержки			
6.1 Определить план технической поддержки программного обеспечения	- Спланировать, как будет выполнен МЕНЕДЖМЕНТ РИСКА в отношении изменений, улучшений, исправлений ошибок и как будет собираться и анализироваться информация о применении для целей оценки корректности мер по УПРАВЛЕНИЮ РИСКОМ и возможности дополнительного снижения РИСКА.		
6.2 Анализ проблем и модификаций	- Проанализировать информацию о применении для цели определения ранее не идентифицированных или дополнительных ОПАСНОСТЕЙ и причин этих ОПАСНОСТЕЙ.	- Провести повторное оценивание ранжирования РИСКОВ и корректности мер по УПРАВЛЕНИЮ РИСКОМ.	- Определить, требуются ли дополнительные меры по УПРАВЛЕНИЮ РИСКАМИ или требуются изменения существующих мер.
6.3 Реализация модификаций	Аналогично ПРОЦЕССУ разработки, но с фокусом на результат действия изменений проанализировать: влияние на существующие меры по УПРАВЛЕНИЮ РИСКОМ; привнесение новых причин ОПАСНОСТЕЙ; привнесение нового функционала к ПРЕДУСМОТРЕННОМУ ПРИМЕНЕНИЮ, которое создает новые ОПАСНОСТИ; регрессионное тестирование кода, связанного с БЕЗОПАСНОСТЬЮ. Релиз программного обеспечения в соответствии с пунктом 5.8 МЭК 62304.		

Приложение Е
(справочное)

БЕЗОПАСНЫЕ случаи

БЕЗОПАСНЫЙ случай — это «структурированное обоснование, подкрепленное совокупностью свидетельств, формирующих убедительный, обоснованный и достоверный случай того, что МЕДИЦИНСКОЕ ИЗДЕЛИЕ является безопасным для данного ПРЕДУСМОТРЕННОГО ПРИМЕНЕНИЯ в данных условиях применения» (адаптировано из UK MoD Def Stan 00-56).

В то время как в таких областях, как СИСТЕМЫ вооружения, шельфовая нефтедобыча, железнодорожный транспорт и ядерная энергетика, концепция БЕЗОПАСНЫХ случаев широко известна, эта методика не является обязательной для индустрии МЕДИЦИНСКИХ ИЗДЕЛИЙ и настоящее приложение не имеет целью установление каких-либо дополнительных требований к ИСО 14971.

Этот стандарт выдвигает предположение, что БЕЗОПАСНЫЙ случай может быть средством структурирования, документирования и передачи для демонстрации соответствующего уровня БЕЗОПАСНОСТИ МЕДИЦИНСКОГО ИЗДЕЛИЯ. БЕЗОПАСНЫЙ случай также может способствовать обеспечению того, что БЕЗОПАСНОСТЬ поддерживается на протяжении всего срока службы МЕДИЦИНСКОГО ИЗДЕЛИЯ.

БЕЗОПАСНЫЙ случай использует результаты МЕНЕДЖМЕНТА РИСКА, чтобы обосновать, почему программное обеспечение достаточно безопасно для своего ПРЕДУСМОТРЕННОГО ПРИМЕНЕНИЯ и почему оно соответствует всем регулирующим требованиям (что может быть сделано в соответствующей регуляторной терминологии).

Можно рассматривать БЕЗОПАСНЫЙ случай как МЕНЕДЖМЕНТ РИСКА или сводку ОСТАТОЧНОГО РИСКА со ссылками на более детальные документы с подтверждающей информацией и свидетельствами в ФАЙЛЕ МЕНЕДЖМЕНТА РИСКА. ФАЙЛ МЕНЕДЖМЕНТА РИСКА может содержать перекрестные ссылки для демонстрации спецификаций и выполнения тестов для всех мер по УПРАВЛЕНИЮ РИСКОМ.

Для реализации БЕЗОПАСНОГО случая требуется наличие следующих компонентов:

- четкий перечень заявлений в отношении СИСТЕМЫ;
- представление подтверждающих свидетельств;
- перечень обоснований БЕЗОПАСНОСТИ, связывающих заявления со свидетельствами;
- предположения и заключения по базе обоснований;
- допущение различных точек зрения и уровней детальности.

Главными элементами БЕЗОПАСНОГО случая являются:

- заявление о свойстве СИСТЕМЫ или некоторой подсистемы;
- свидетельство, которое используется как база для обоснования БЕЗОПАСНОСТИ. Это могут быть или факты (например, основанные на известных научных принципах или предварительных исследованиях), допущения или заявления, полученные на более низком уровне аргументирования;
- обоснование связывания свидетельства и заявления, которое может быть детерминистским, вероятностным или качественным;
- логический вывод — метод, устанавливающий правила вывода суждений.

Для дальнейшей информации об элементах и структуре БЕЗОПАСНЫХ случаев см. [9].

Две публикации, представляющие собой хорошее введение в БЕЗОПАСНЫЕ случаи и структурированное по цели изложение см. [10] и [11].

Приложение ДА
(справочное)

Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации и действующим в этом качестве межгосударственным стандартам

Таблица ДА.1

Обозначение ссылочного международного стандарта	Степень соответствия	Обозначение и наименование соответствующего национального, межгосударственного стандарта
ИСО 14971:2007	IDT	ГОСТ ISO 14971—2011 «Изделия медицинские. Применение менеджмента риска к медицинским изделиям»
МЭК 62304:2006	IDT	ГОСТ Р МЭК 62304—2013 «Изделия медицинские. Программное обеспечение. Процессы жизненного цикла»

Примечание — В настоящей таблице использовано следующее условное обозначение степени соответствия стандартов:
- IDT — идентичные стандарты.

Список определенных терминов

Библиография

[1]	ИСО 13485:2003	Медицинские изделия. Системы менеджмента качества. Системные требования для целей регулирования
	ISO 13485:2003	Medical devices — Quality management systems — Requirements for regulatory purposes
[2]	МЭК 60812	Анализ методов технической надежности. Метод анализа видов и последствий отказов
	IEC 60812	Analysis techniques for system reliability — Procedure for failure mode and effects analysis (FMEA)
[3]	МЭК 61025	Анализ дерева неисправностей
	IEC 61025	Fault tree analysis (FTA)
[4]	МЭК 61882	Исследование опасности и работоспособности. Прикладное руководство
	IEC 61882	Hazard and operability studies (HAZOP studies) — Application guide
[5]	МЭК 62366	Медицинские изделия. Проектирование медицинских изделий с учетом эксплуатационной пригодности
	IEC 62366	Medical devices — Application of usability engineering to medical devices
[6]	МЭК 80001-1	Менеджмент рисков для информационных сетей, связанных с медицинскими приборами. Часть 1. Роли, ответственность и деятельность
	IEC 80001-1	Application of risk management to information technology (IT) networks incorporating medical devices — Part 1: Roles, responsibilities and activities
[7]	Паллум Л. Бостон: Артех хаус, 2001	Программное обеспечение отказоустойчивой техники и ее реализация. Бостон
	Pullum L. Boston: Artech House, 2001	Software fault tolerant techniques and implementation
[8]	Банатре М. Ли П. Берлин, Германия Спринджер-Велаг, 1994	Оборудование и программное обеспечение для архитектуры отказоустойчивости: Опыт и перспективы
	Banatre M., Lee P. (Eds)., Berlin, Germany: Springer-Verlag, 1994	Hardware and Software Architectures for Fault Tolerance: Experiences and Perspectives
[9]	Бишоп П., Блумфилд Р. (1998)	Симпозиум по методологии развития безопасности, безопасности критически важных систем
	Bishop P., Bloomfield R. (1998)	A Methodology for Safety Case Development, Safety-Critical Systems Symposium http://www.adelard.co.uk/resources/papers/pdf/sss98web.pdf
[10]	Келли Т.П., Детройт, Март 2004	Системный подход к управлению безопасностью дела. Материалы мирового конгресса SAE 2004 (Материалы опубликованы Обществом автомобильных инженеров)
	Kelly T. P. Detroit, March 2004	Systematic Approach to Safety Case Management, Proceedings of SAE 2004 World Congress (Proceedings published by the Society for Automotive Engineers)
[11]	Вивер Р.А., Келли Т.П., Июль 2004	Обозначенные цели структурирования — Обозначение безопасности аргументов, материалы по надежности систем и сетей
	Weaver R. A., Kelly T. P., July 2004	The Goal Structuring Notation — A Safety Argument Notation, Proceedings of the Dependable Systems and Networks 2004 Workshop on Assurance Cases

УДК 006.85 ОКС 11.040.01 Р20 ОКП 94 0000

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

Редактор *А. В. Барандеев*
Технический редактор *Е. В. Беспрозванная*
Корректор *Л. Я. Митрофанова*
Компьютерная верстка *В. Н. Романовой*

Сдано в набор 01.09.2014. Подписано в печать 12.11.2014. Формат 60×84 $^{1}/_{8}$. Бумага офсетная. Гарнитура Ариал. Печать офсетная. Усл. печ. л. 7,44. Уч.-изд. л. 6,80. Тираж 35 экз. Зак. 1456.

ФГУП «СТАНДАРТИНФОРМ», 123995 Москва, Гранатный пер., 4.
www.gostinfo.ru info@gostinfo.ru
Набрано и отпечатано в Калужской типографии стандартов, 248021 Калуга, ул. Московская, 256.