ПРИМЕНЕНИЕ ГОСТ IEC 62304–2022 В РАЗРАБОТКЕ МЕДИЦИНСКОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ: АНАЛИЗ И ПЕРСПЕКТИВЫ

Ключевые слова: медицинские изделия, программное обеспечение для медицины, жизненный цикл программного обеспечения, создание безопасного программного обеспечения, обзор требований ГОСТ IEC 62304–2022.

Михаил ВИНОГРАДОВ, Сергей СОЛОННИКОВ

УДК 658.562.014:006.354

Аннотация. Рассматриваются требования ГОСТ IEC 62304–2022 как основы для создания безопасного программного обеспечения (ПО) для применения в здравоохранении в современных условиях разработки, включая использование наследуемого ПО и ПО неизвестного происхождения (ПОНП).

ВВЕДЕНИЕ

Разработка программного обеспечения (ПО) для применения в медицинской практике требует строгого соблюдения стандартов безопасности, что обусловлено высокими рисками для жизни и здоровья пациентов. Инцидент с использованием аппарата лучевой терапии *Therac-25* (1985–1987), который привел к гибели пациентов из-за сочетания ряда факторов, таких как ошибки в ПО и недоработки в аппаратной части, стал катализатором формирования современных нормативов, включая международный стандарт IEC 62304. В настоящее время действует первая редакция этого стандарта с поправкой от 2015 года – *IEC 62304:2006 + Amd.1:2015, Medical device software – Software life cycle processes*, а в Российской Федерации действует идентичный межгосударственный стандарт ГОСТ IEC 62304–2022 «Изделия медицинские. Программное обеспечение. Процессы жизненного цикла» [2].

В данной статье проводится обзор основных требований ГОСТ IEC 62304–2022 в отношении обеспечения безопасности медицинского ПО. Фрагменты текста в оригинальной редакции стандарта выделены **полужирным шрифтом**.

ОТПРАВНАЯ ТОЧКА

Инцидент с аппаратом *Therac-25* стал ключевым событием, которое привело к формированию основ регулирования в области медицинского ПО. *Therac-25* – линейный ускоритель для лучевой терапии, разработанный в 1980-х годах, представлял собой инновационное устройство с полностью автоматизированным управлением, однако его использование привело к нескольким случаям получения пациентами смертельных доз облучения (до 17 000 рад).

В ходе расследования было установлено, что основными причинами трагических событий стали [6]:

– ошибки в ПО, которые не были выявлены из-за недостаточного тестирования;

– трудно интерпретируемые сообщения об ошибках в ПО, что затрудняло диагностику проблем;

– отсутствие избыточности [4] и дублирования систем безопасности: в отличие от предыдущих моделей (*Therac-6* и *Therac-20*), *Therac-25* полагался исключительно на управление при помощи ПО без включения механических резервных систем блокирования;

– ошибки проектирования: поворотный диск, отвечающий за смену режимов облучения, не имел надежного контроля своего положения. Отсутствовали механизмы для отслеживания и оповещения об опасных ситуациях, таких как поломка микропереключателя и неправильное положение диска;

– плохое документирование наследуемого ПО: в *Therac-25* использовалось ПО, применявшееся в младших моделях, однако все изменения, касающиеся адаптации ПО к новой модели ускорителя, не были задокументированы должным образом;

IMPLEMENTATION OF GOST IEC 62304–2022 IN MEDICAL SOFTWARE DEVELOPMENT: ANALYSIS AND FUTURE PERSPECTIVES

Mikhail A. VINOGRADOV, Sergey V. SOLONNIKOV

Abstract. The requirements of GOST IEC 62304–2022 are examined as a foundation for safe medical devices software developing under modern development conditions, including the use of legacy software and software of unknown provenance (SOUP).

Keywords: medical devices, medical software, software life cycle, creation of secure software, software as medical device, review of GOST IEC 62304–2022 requirements.

– недостаточно тщательная проработка рисков и неполноценная стратегия обеспечения безопасности изделия: анализ рисков был проведен только для аппаратной части, причем в ограниченном объеме, без учета ПО.

Инцидент стал поводом для проведения серьезного расследования со стороны *FDA* (Управления по санитарному надзору за качеством пищевых продуктов и медикаментов США) и послужил отправной точкой развития законодательства в области медицинского ПО.

Основными уроками, которые индустрия разработки ПО для медицины вынесла из инцидента с *Therac-25,* стали следующие [6, 11, 12]:

1. Все этапы жизненного цикла ПО требуют тщательного и подробного документирования, начиная с проектирования и разработки МИ и заканчивая его выводом из эксплуатации.

2. При разработке медицинского ПО (особенно для использования в составе программно-аппаратных комплексов) следует применять избыточность и дублирование, то есть критически важные системы должны иметь резервные механизмы обеспечения безопасности.

3. ПО должно проходить тщательное тестирование, особенно в условиях, имитирующих использование в реальной среде применения (валидацию).

4. Должен быть учтен человеческий фактор: интерфейсы ПО и сообщения об ошибках должны быть понятными для операторов и исключать неоднозначное толкование информации.

Таким образом, инцидент с *Therac-25* четко обозначил важность разработки и строгого соблюдения стандартов безопасности при разработке медицинского ПО, что нашло отражение в современных международных нормативных документах, таких как ГОСТ IEC 62304–2022. Этот стандарт, по сути, создан на основе уроков, извлеченных из подобных трагедий, и направлен на предотвращение подобных инцидентов в будущем.

ОСНОВНЫЕ ПОЛОЖЕНИЯ ГОСТ IEC 62304–2022

ГОСТ IEC 62304–2022 (далее – Стандарт) закладывает основу процессов жизненного цикла, включая виды деятельности и задачи, необходимые для проектирования и технической поддержки безопасного ПО, применяемого в медицине.

Этот стандарт применим как к самостоятельному медицинскому ПО (когда оно само по себе классифицируется как медицинское изделие (МИ), так называемое *Software as Medical Device – SaMD*), так и к встроенному ПО медицинских изделий (то есть представляет собой часть медицинского устройства, например управляющее ПО томографа или анализатора крови, *Software in Medical Device, SiMD*).

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

Примечание. В это определение входит МЕДИЦИНСКОЕ ИЗДЕЛИЕ – программный продукт, который сам по себе является МЕДИЦИНСКИМ ИЗДЕЛИЕМ» [2].

Стандарт регламентирует требования к ПО медицинского назначения, независимо от типа носителя, используемого для его хранения (включая, но не ограничиваясь жесткими дисками, оптическими носителями, постоянной и флеш-памятью), а также от способа его распространения (посредством сетевых технологий, электронной почты, оптических носителей, флеш-накопителей или *EEPROM*). Вместе с тем следует отметить, что средства доставки ПО сами по себе не

подпадают под категорию ПО медицинских изделий.

Стандарт использует классический процессный подход, хорошо описанный в ГОСТ ISO 13485–2017. Основная концепция Стандарта предполагает, что разработка и обслуживание медицинского ПО осуществляются с использованием систем менеджмента качества (СМК) на основе ГОСТ ISO 13485–2017 и систем менеджмента рисков на основе ГОСТ ISO 14971–2021 [1–4].

Необходимо отметить, что процесс управления рисками, описанный в ГОСТ ISO 14971–2021, интегрирован в Стандарт с дополнительными требованиями, специфичными для ПО. Эти требования касаются определения вклада программных факторов в возникновение опасных ситуаций и изложены в разделе 7 Стандарта.

Опасные ситуации, которые могут быть вызваны ПО (например, предоставление пользователю вводящей в заблуждение информации, ведущей к неправильным действиям), анализируются в рамках процесса менеджмента риска. Процесс менеджмента риска для ПО должен быть интегрирован в общий процесс менеджмента риска всего МИ.

ГОСТ IEC 62304–2022 определяет два дополнительных процесса, необходимых для разработки безопасного ПО:

– процесс менеджмента конфигурации ПО (раздел 8 Стандарта), который обеспечивает управление изменениями и идентификацию компонентов ПО;

– процесс разрешения проблем ПО (раздел 9 Стандарта), направленный на выявление, анализ и устранение проблем, возникающих в ходе разработки и эксплуатации.

В настоящее время действует версия IEC 62304:2006, которая содержит Поправку 1 *(Amd.1:2015)* от 2015 года. Эта поправка вводит дополнительные требования к устаревшему/наследуемому (легаси) ПО, разработанному до появления второй версии Стандарта. Эта поправка была введена для того, чтобы помочь производителям продемонстрировать соответствие Стандарту в контексте Европейских директив, а ныне – Регламента Европейского Союза для медицинских изделий *(MDR)*. Поправкой были внесены обновленные требования к классификации безопасности ПО и внедрен риск-ориентированный подход.

По аналогии с ГОСТ ISO 13485–2017, ГОСТ IEC 62304–2022 не предписывает конкретную организационную структуру компании или подразделения, ответственные за выполнение процессов, деятельности или задач. Основное требование – выполнение всех требований, необходимых для соответствия Стандарту. ГОСТ IEC 62304–2022 также не устанавливает строгие требования к формату или содержанию документации, оставляя разработчикам свободу в выборе способа документирования задач. Также этот документ не навязывает конкретную модель жизненного цикла, предоставляя разработчикам возможность самостоятельного выбора модели, наиболее подходящей для их проекта, при условии отражения процессов, деятельности и задач, описанных в Стандарте.

ГОСТ IEC 62304–2022 содержит пять приложений:

– приложение A включает обоснование требований стандарта;

– B предоставляет рекомендации по применению положений Стандарта;

– C описывает взаимосвязи с другими стандартами;

– D приводит обзор практических моментов реализации требований в процессах изготовителя;

– ДА содержит ссылочные стандарты.

ТЕРМИНОЛОГИЯ ГОСТ IEC 62304–2022 (РАЗДЕЛ 3 СТАНДАРТА)

ГОСТ IEC 62304–2022 широко заимствует терминологию ГОСТ ISO 14971–2021 в части менеджмента риска. Другая часть терминологической базы была заимствована из документа *Standard Glossary of Software Engineering Terminology*, разработанного Институтом инженеров электротехники и электроники *(IEEE)* в 1990-х годах [7]. Стандарт наследует не только терминологию, но и инженерный подход к разработке ПО, характерный для того периода. Такой подход рассматривает ПО с инженерной точки зрения, аналогично электротехническим изделиям, и отражает методы разработки, распространенные в 1990-х годах (образно говоря, это «эпоха *Windows 3.0*», выпуск которой состоялся в мае 1990 года). Это создает разрыв между подходом Стандарта и современными методологиями гибкой разработки ПО, что часто вызывает критику со стороны разработчиков [11]. В настоящее время готовится вторая редакция IEC 62304, в которой предполагается обновить подходы и уточнить терминологическую базу для соответствия современным требованиям к разработке ПО.

Среди терминов, которые содержатся в стандарте, следует особо выделить те, которые описывают составные части ПО, а именно: 3.27 Программная система (*Software System*), 3.25 Программная составная часть (ПСЧ) (*Software Item*) и 3.28 Программный блок (*Software Unit*).

Взаимоотношение этих элементов состоит в следующем: программная система (ПС) включает программные составные части (ПСЧ), которые, в свою очередь, складываются из программных блоков (ПБ). Программный блок представляет собой наименьшую элементарную единицу ПО.

Еще один крайне важный термин – 3.29 ПОНП – Программное обеспечение неизвестного происхождения *(SOUP software of unknown provenance)*. Под ПОНП в Стандарте понимается «**ПРОГРАММНАЯ СОСТАВНАЯ ЧАСТЬ, которая уже разработана и общедоступна, но не была предназначена для включения в состав МЕДИЦИНСКОГО ИЗДЕЛИЯ (также известное как «готовое ПО») или ПРОГРАММНАЯ СОСТАВНАЯ ЧАСТЬ, разработанная ранее, для которой недоступ-**

ны требуемые записи ПРОЦЕССОВ разработки» [2]. Следует отметить, что разработчики смешали понятия ПОНП и готового ПО (которое в зарубежной литературе именуется *Off-the-Shelf Software (OTS)*), хотя между ними существует принципиальная разница.

Для ПОНП характерны следующие признаки:

– отсутствие полностью задокументированных процессов жизненного цикла (например, процессов проектирования и разработки, результатов тестирования, анализа рисков), что представляет серьезный риск для включения такого ПО в состав программных систем, разрабатываемых с учетом требований ГОСТ IEC 62304–2022;

– ограниченная поддержка (разработчик не всегда может или имеет возможность исправлять ошибки в своем ПО);

– отсутствие гарантий безопасности кода, его киберзащищенности в широком смысле.

Готовое ПО *(OTS)* – это коммерческое официально реализуемое ПО, производимое специализированными разработчиками; оно поставляется в виде готового продукта, сопровождается эксплуатационной документацией, имеет документированные процессы (к готовым ПО относятся, например, операционные системы, СУБД, лицензированные библиотеки);

Для готового ПО характерны следующие признаки:

– процессы жизненного цикла ПО задокументированы, но эта документация может не соответствовать требованиям IEC 62304, поскольку разработчик мог не запланировать медицинское применение для своего ПО;

– готовое ПО официально поддерживается разработчиком (выпускаются обновления, дополнения, исправления уязвимостей, предоставляется техническая поддержка);

– такое ПО может иметь специальную поддержку для медицинских изделий (например, операционная система *Windows 10 IoT* или *MediTUX OS*).

Итак, использование ПОНП в программной системе сопряжено со значительными рисками и требует больших усилий для проведения проверки, документирования, но ПОНП часто незаменимо (например, специализированные *open-source*-библиотеки и модули). Что касается готового ПО, то оно более безопасно, но также может требовать усилий по его дополнительной оценке соответствия требованиям IEC 62304. Готовое ПО может быть удобнее, но при этом более дорогостоящим или избыточным по функционалу [8–11].

Также мы хотели бы рассмотреть термин «устаревшее (наследуемое) ПО», и подчеркнуть его значение для создания ПО с учетом требований ГОСТ IEC 62304–2022:

«3.36 УСТАРЕВШЕЕ [НАСЛЕДУЕМОЕ] ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ (LEGACY SOFTWARE): ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МЕДИЦИНСКОГО ИЗДЕЛИЯ, которое было легально выпущено в обращение и по-прежнему доступно на рынке, но для которого недостаточно объективных свидетельств того, что оно было разработано в соответствии с текущей версией настоящего стандарта.» [2].

Устаревшим (наследуемым) ПО в настоящее время считается медицинское ПО, выпущенное до 2015 года, то есть до выхода стандарта с Поправкой 1. Для всего устаревшего ПО, выпущенного до 2015 года и используемого до сего дня как медицинское изделие (или в составе МИ), должно быть установлено соответствие IEC 62304:2006 + Amd.1:2015, чтобы ситуация, аналогичная инциденту, связанному с *Therac-25*, никогда не повторилась. Зачастую это требует серьезного ретроспективного анализа и дополнительного тестирования. В подобных случаях производитель медицинских изделий должен оценить риски и принять решение – сконцентрировать усилия на доказательстве соответствия своего наследуемого ПО требованиям стандарта или разработать новое ПО с нуля.

КЛАССИФИКАЦИЯ БЕЗОПАСНОСТИ ПО – КРАЕУГОЛЬНЫЙ КАМЕНЬ ГОСТ IEC 62304–2022

Стандарт требует, чтобы изготовитель присвоил каждой программной системе класс безопасности A, B или C на основе оценки риска причинения вреда пациенту, пользователю или другим лицам в наихудшем сценарии. Классификация зависит от степени тяжести вреда, который может быть нанесен в результате создания опасной ситуации, которой способствует рассматриваемая программная система.

Класс безопасности A: программная система не способствует возникновению опасной ситуации или, если система может способствовать созданию опасной ситуации, степень опасности уменьшается до допустимого уровня путем реализации внешних по отношению к рассматриваемой программной системе мероприятий по управлению рисками.

Класс B: хотя программная система может способствовать опасной ситуации, приводящей к недопустимому риску даже после принятия внешних мер по управлению риском, возможный вред не выражается серьезной травмой.

Класс C: программная система может способствовать опасной ситуации, приводящей к недопустимому риску даже после выполнения внешних мероприятий по управлению риском. Возможный вред представляет летальный исход пациента или серьезную травму.

Изготовитель может пересмотреть классификацию рассматриваемой программной системы, первоначально отнесенной к классу B или C, путем реализации дополнительных внешних мер управления рисками (например, посредством изменения архитектуры системы). Это позволяет присвоить рассматриваемой системе новый класс безопасности, если риски будут снижены до допустимого уровня.

ЖИЗНЕННЫЙ ЦИКЛ ПО

ГОСТ IEC 62304–2022 выделяет следующие основные этапы жизненного цикла медицинского ПО:

1. Процесс разработки требований.

Начальный этап жизненного цикла предполагает формирование детализированных и измеримых требований к ПО, включая функциональные характеристики, эксплуатационные параметры и ограничения использования. Особое внимание уделяется анализу предполагаемых условий эксплуатации и потенциальных рисков, связанных с человеческим фактором. На данном этапе осуществляется первоначальная оценка рисков, определяющая класс безопасности ПО (A, B или C), результаты которой в дальнейшем влияют на глубину проработки последующих процессов. Особое значение приобретают вопросы интеграции ПОНП и наследуемого ПО, что требует специального рассмотрения в рамках каждого процесса жизненного цикла. Архитектурные решения должны предусматривать механизмы изоляции (обособления) потенциально ненадежных компонентов и минимизации их влияния на критические функции системы. Для наследуемых компонентов ПО, разработанных до внедрения ГОСТ IEC 62304–2022, требуется проведение ретроспективного анализа соответствия текущим требованиям безопасности. Результат процесса разработки требований, как правило, выражается в сформированном списке требований и способов оценки соответствия этим требованиям.

2. Проектирование архитектуры.

Проектирование архитектуры ПО осуществляется с учетом потенциально опасных ситуаций, выявленных на этапе разработки требований. Для ПО классов безопасности B и C обязательное требование заключается в документировании принятых решений, направленных на минимизацию выявленных рисков, включая резервирование критических функций и механизмы самодиагностики. Для ПОНП и наследуемых компонентов обязательно проведение углубленного анализа происхождения, включая проверку безопасности, лицензионной чистоты, истории обновлений и известных уязвимостей. Риск-ориентированный подход на данном этапе требует классификации таких компонентов по степени их критичности для функционирования МИ.

3. Реализация и кодирование.

Процесс кодирования (имплементации – в терминологии Стандарта) регламентируется внутренними стандартами разработки, включая требования к стилю программирования, комментированию кода и использованию безопасных практик. На данном этапе продолжается актуализация анализа рисков с учетом конкретных реализационных решений.

4. Верификация.

Верификационные мероприятия включают комплекс статических и динамических методов тестирования, направленных на подтверждение соответствия ПО установленным ранее требованиям. Объем верификации напрямую зависит от класса безопасности: для класса C обязательно проведение модульного, интеграционного и системного тестирования с покрытием всех критических путей выполнения. Результаты верификации документируются и учитываются при актуализации оценки рисков.

5. Управление конфигурацией.

Процесс управления конфигурацией обеспечивает контроль версий, отслеживание изменений и поддержание целостности ПО на протяжении всего жизненного цикла. Каждое изменение сопровождается оценкой его влияния на безопасность и эффективность МИ. Для критических изменений в ПО классов B и C требуется проведение повторной верификации и валидации.

Для ПОНП-компонентов устанавливаются особые правила управления версиями и обновлениями. Каждое изменение в конфигурации таких компонентов требует проведения полного цикла переоценки рисков. Документирование всех модификаций и их влияния на безопасность системы установлено обязательным требованием для МИ классов безопасности B и C.

6. Решение проблем и сопровождение.

Пострелизная фаза включает мониторинг эксплуатации ПО, анализ поступающих сообщений о проблемах и своевременное устранение выявленных недостатков. Каждая зарегистрированная проблема подвергается оценке риска, определяющей срочность и объем необходимых корректирующих действий. Критические обновления, влияющие на безопасность, требуют повторной валидации перед распространением.

7. Интеграция с управлением рисками.

Менеджмент риска является сквозным процессом, сопровождающим все этапы жизненного цикла ПО. На каждом этапе осуществляется идентификация новых опасностей и опасных ситуаций, оценка их последствий и разработка мер по снижению рисков. Особое внимание уделяется документированию всех решений, связанных с управлением рисками, что является обязательным требованием для ПО классов B и C.

ОСНОВНЫЕ ОГРАНИЧЕНИЯ ГОСТ IEC 62304–2022

Современная версия Стандарта основана, повторим, на IEC 62304:2006 с поправкой 2015 года *(Amd.1:2015)*, тогда как в 2020-х годах технологии и восприятие рисков значительно изменились. Стандарт не учитывает такие современные киберугрозы, как «уязвимости нулевого дня» и атаки на цепочки поставок ПО. Также следует отметить, что раздел Стандарта, посвященный ПОНП, требует актуализации, в частности, в отношении широко используемых *open-source*-библиотек, применение которых стало обычной практикой при разработке ПО с технологиями искусственного интеллекта (ИИ).

К основным недостаткам Стандарта следует также отнести отсут-

ствие специализированных требований к ПО с технологиями ИИ и машинного обучения. Стандарт не учитывает специфику этих новых направлений, не устанавливает требования к алгоритмам машинного обучения, к валидации обучающих данных, не предлагает классификацию нейросетевых архитектур. На наш взгляд, крайне важным для Стандарта будет введение понятия «доверие к системе искусственного интеллекта» и описание требований для его обеспечения [5].

Ожидается, что указанные недостатки будут устранены во второй редакции Стандарта, которая планируется к выходу в 2025 году.

ЗАКЛЮЧЕНИЕ

ГОСТ IEC 62304—2022 устанавливает комплексный подход к управлению жизненным циклом медицинского ПО, обеспечивающий постоянное управление качеством и безопасностью на всех этапах — от разработки требований до вывода из эксплуатации. Интеграция процессов разработки с непрерывной оценкой рисков позволяет создавать медицинское ПО, соответствующее современным требованиям, что особенно критично для изделий классов B и C. Для должной реализации требований Стандарта необходимо создание пригодной и результативной системы менеджмента качества и документирование всех принятых решений.

ИСТОЧНИКИ

1. ГОСТ ISO 14971–2021. Изделия медицинские. Применение менеджмента риска к медицинским изделиям [Электронный ресурс]. Кодекс. URL: https://docs.cntd.ru/document/1200181438
2. ГОСТ IEC 62304–2022. Изделия медицинские. Программное обеспечение. Процессы жизненного цикла [Электронный ресурс]. Кодекс. URL: https://docs.cntd.ru/document/1200193845
3. ГОСТ ISO 13485–2017. Изделия медицинские. Системы менеджмента качества. Требования для целей регулирования [Электронный ресурс]. Кодекс. URL: https://docs.cntd.ru/document/1200146167
4. ГОСТ Р 55544-2013/IEC/TR 80002-1:2009. Программное обеспечение медицинских изделий. Часть 1. Руководство по применению ИСО 14971 к программному обеспечению изделий [Электронный ресурс]. Кодекс. URL: https://docs.cntd.ru/document/1200104471
5. ГОСТ Р 59276–2020. Системы искусственного интеллекта. Способы обеспечения доверия. Общие положения [Электронный ресурс]. Кодекс. URL: https://docs.cntd.ru/document/1200177291
6. Leveson N.G., Turner C.S. An Investigation of the Therac-25 Accidents // Computer. Vol. 26. No. 7. P. 18-41, July 1993, DOI: 10.1109/MC.1993.274940/
7. IEEE Std 610.12-1990. IEEE Standard Glossary of Software Engineering Terminology. IEEE Computer Society; 1990.
8. Höss A., Lampe C., Panse R., Ackermann B., Naumann J., Jäkel O. First experiences with the implementation of the European standard EN 62304 on medical device software for the quality assurance of a radiotherapy unit // Radiat Oncol. 2014 Mar 21;9:79. DOI: 10.1186/1748-717X-9-79. PMID: 24655818; PMCID: PMC3994433
9. Medical Device Coordination Group. MDCG 2019-11 Guidance on Qualification and Classification of Software in Regulation (EU) 2017/745 – MDR and Regulation (EU) 2017/746 – IVDR. 2019.
10. IMDRF SaMD Working Group. IMDRF/ SaMD WG/ N10FINAL2013 Software as a Medical Device. Key Definitions. International Medical Device Regulators Forumdev; 2013.
11. Engineering High Quality Medical Software. Regulations, standards, methodologies and tools for certification. Antonio Coronato. Published by The Institution of Engineering and Technology, London, United Kingdom. ISBN: 978-1-78561-248-0
12. Värri A, Kranz-Zuppan P, de la Cruz R. IEC 62304 Ed. 2: Software Life Cycle Standard for Health Software // Stud Health Technol Inform. 2019 Aug 21;264:868-872. DOI: 10.3233/SHTI190347. PMID: 31438048.

REFERENCES

1. *GOST ISO 14971–2021. Izdeliya meditsinskie. Primenenie menedzhmenta riska k meditsinskim izdeliyam* [GOST ISO 14971–2021. Medical devices. Application of risk management to medical devices]. Kodeks, available at: https://docs.cntd.ru/document/1200181438
2. *GOST IEC 62304–2022. Izdeliya meditsinskie. Programmnoe obespechenie. Protsessy zhiznennogo tsikla* [GOST IEC 62304–2022. Medical devices. Software. Life cycle processes]. Kodeks, available at: https://docs.cntd.ru/document/1200193845
3. *GOST ISO 13485–2017. Izdeliya meditsinskie. Sistemy menedzhmenta kachestva. Trebovaniya dlya tseley regulirovaniya* [GOST ISO 13485–2017. Medical devices. Quality management systems. Requirements for regulatory purpose]. Kodeks, available at: https://docs.cntd.ru/document/1200146167
4. *GOST R 55544-2013/IEC/TR 80002-1:2009. Programmnoe obespechenie meditsinskikh izdeliy. Chast' 1. Rukovodstvo po primeneniyu ISO 14971 k programmnomu obespecheniyu izdeliy* [GOST R 55544-2013/IEC/TR 80002-1:2009. Medical devices software. Part 1. Guidance on the application of ISO 14971 medical devices software]. Kodeks, available at: https://docs.cntd.ru/document/1200104471
5. *GOST R 59276–2020. Sistemy iskusstvennogo intellekta. Sposoby obespecheniya doveriya. Obshchie polozheniya* [GOST R 59276–2020. Artificial intelligence systems. Methods for ensuring trust. General]. Kodeks, available at: https://docs.cntd.ru/document/1200177291
6. Leveson N.G., Turner C.S. An Investigation of the Therac-25 Accidents. *Computer*, vol. 26, N 7, pp. 18-41, July 1993, DOI: 10.1109/MC.1993.274940/
7. *IEEE Std 610.12-1990. IEEE Standard Glossary of Software Engineering Terminology*. IEEE Computer Society; 1990.
8. Höss A., Lampe C., Panse R., Ackermann B., Naumann J., Jäkel O. First experiences with the implementation of the European standard EN 62304 on medical device software for the quality assurance of a radiotherapy unit. *Radiat Oncol.*, 2014 Mar 21;9:79. DOI: 10.1186/1748-717X-9-79. PMID: 24655818; PMCID: PMC3994433
9. *Medical Device Coordination Group. MDCG 2019-11 Guidance on Qualification and Classification of Software in Regulation (EU) 2017/745 – MDR and Regulation (EU) 2017/746 – IVDR*. 2019.
10. *IMDRF SaMD Working Group. IMDRF/ SaMD WG/ N10FINAL2013 Software as a Medical Device. Key Definitions.* International Medical Device Regulators Forumdev; 2013.
11. *Engineering High Quality Medical Software. Regulations, standards, methodologies and tools for certification*. Antonio Coronato. Published by The Institution of Engineering and Technology, London, United Kingdom. ISBN: 978-1-78561-248-0
12. Värri A., Kranz-Zuppan P., de la Cruz R. IEC 62304 Ed. 2: Software Life Cycle Standard for Health Software. *Stud Health Technol Inform.*, 2019 Aug 21;264:868-872. DOI: 10.3233/SHTI190347. PMID: 31438048

Михаил Андреевич ВИНОГРАДОВ

инженер НИЛ электрофизиологии, Институт физиологии ФГАОУ ВО РНИМУ им. Н.И. Пирогова Минздрава России (Пироговский Университет), аудитор СМК организаций, работающих в области здравоохранения (ISO 9001, ISO 13485), член Технического комитета 436 «Менеджмент качества и общие аспекты медицинских изделий» Росстандарта

Mikhail A. VINOGRADOV

Research Laboratory of Electrophysiology, Institute of Physiology, Pirogov Russian National Research Medical University of the Ministry of Health of the Russian Federation (Pirogov University), Technical Committee 436 "Quality Management and General Aspects of Medical Devices" of Rosstandart. Moscow, Russian Federation. ORCID: 0009-0001-3096-4391, e-mail: mvinogradov@inbox.ru

Сергей Владимирович СОЛОННИКОВ

председатель Технического комитета 436 «Менеджмент качества и общие аспекты медицинских изделий» Росстандарта

Sergey V. SOLONNIKOV

Technical Committee 436 "Quality Management and General Aspects of Medical Devices" of Rosstandart. Moscow, Russian Federation, e-mail: ssv@meditest.ru