УДК 004.056

МОДЕЛЬ ПРОЦЕССА БЕЗОПАСНОЙ РАЗРАБОТКИ ПРОГРАММНЫХ СРЕДСТВ КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ ИНФОРМАЦИИ

В. А. ГЕРАСИМОВ, А. О. ВОРОЖЦОВ

Научно-производственное республиканское унитарное предприятие «Научно-исследовательский институт технической защиты информации», г. Минск, 220088, Республика Беларусь

Введение

Средства криптографической защиты информации (далее — СКЗИ) являются механизмом, который обеспечивает конфиденциальность, целостность, подлинность критической информации. При разработке СКЗИ требуется повышенное внимание к безопасности, так как ошибка может напрямую создавать угрозу безопасности обрабатываемой информации: раскрыть промежуточные этапы шифрования, экспортировать личные ключи или внедрить уязвимость.

Классический жизненный цикл разработки программного обеспечения (Software Development Life Cycle, SDLC) создан прежде всего для эффективного выполнения функциональных требований с предсказуемыми сроками. Вопросы безопасности нередко переносятся на стадию тестирования или развертывания. Для СКЗИ такой порядок неприемлем: найденная уязвимость может относиться не просто к части кода, а к криптографической границе изделия, архитектуре управления ключами или криптографическим алгоритмам. В этом случае устранение несоответствия требует возврата к ранним этапам, вплоть до перепроектирования.

Цель — обосновать внедрение процесса безопасной разработки программного обеспечения (Secure Software Development Life Cycle, SSDLC) при разработке СКЗИ и предложить модель процесса, согласованную с законодательством Республики Беларусь, международными практиками и опытом внедрения стандартов. Под SSDLC понимается не отдельная каскадная методология, а встраивание требований безопасности во все этапы SDLC.

1. Нормативные предпосылки безопасной разработки СКЗИ

В Республике Беларусь для выпуска средств защиты информации (далее — СЗИ) в обращение применяется подтверждение соответствия требованиям технического регламента ТР 2013/027/BY в форме сертификации или декларирования. Сертификация обязательна, в частности, для средств, используемых при защите государственных секретов, информации ограниченного распространения, критически важных объектов информатизации, а также при обеспечении целостности и подлинности электронных документов [1]. Требования уточняются приказами Оперативно-аналитического центра при Президенте Республики Беларусь (далее — ОАЦ) и государственными стандартами серии СТБ 34.101.

Порядок государственной экспертизы, установленный приказом ОАЦ от 26 августа 2013 г. № 60 (в редакции с учетом приказа от 27 мая 2026 г. № 98), имеет последовательный характер: каждый последующий этап выполняется при положительном результате предыдущего. К заявке на сертификацию должны прилагаться конструкторская, технологическая и программная документация, описание всех реализованных функций защиты и структуры файлов, а также требования нормативно-правовых актов (далее — НПА), на соответствие

которым проводится экспертиза. Идентификатором образца является контрольная характеристика (хэш-значение). Отрицательные результаты испытаний приводят к приостановлению или прекращению экспертизы, а изменения продукта, документации, технологии изготовления либо обнаруженные уязвимости могут стать основанием для приостановления или отмены экспертного заключения [2].

Указанные положения фактически требуют управляемости конфигурации, воспроизводимости сборки, однозначного сходства версии исходного кода с испытанным образцом, полноты программной документации и формализованного учета изменений. Вышеперечисленные пункты возможно реализовать непосредственно на этапе тестирования, но в таком случае создание полноценной документации будет на порядки труднее, чем при написании и поддержке с самого начала разработки.

Приказ ОАЦ от 12 марта 2020 г. № 77 связывает подтверждение соответствия с конкретными стандартами. Для ряда программных СКЗИ установлены требования уровней 1 и 2 по СТБ 34.101.27-2022, а для программно-аппаратных — уровни 3 или 4; отдельно учитываются компоненты с открытыми исходными текстами, управление объектами внутри криптографической границы и реализация удаленного доступа [3]. В зависимости от назначения средства применяются СТБ 34.101.31 (алгоритмы шифрования и контроля целостности), СТБ 34.101.45 (электронная цифровая подпись и транспорт ключа), СТБ 34.101.47-2017 (генерация псевдослучайных чисел), СТБ 34.101.65 или СТБ 34.101.90 (протокол транспортного уровня, TLS) и другие документы серии [4-8]. Приказы ОАЦ № 66 и № 195 дополняют нормативный контекст для критически важных объектов информатизации и защиты персональных данных [9, 10]. Следовательно, нормативные требования не могут рассматриваться лишь в конце цикла разработки, но должны рассматриваться еще на этапе проектирования.

2. Сравнительный анализ SDLC и SSDLC

Основное различие подходов состоит в способе принятия решений по безопасности. Традиционный SDLC рассматривает безопасность как выходная характеристика средней важности. В SSDLC она закладывается с самого начала и контролируется на каждом этапе. Сравнение приведено в таблице 1.

Таблица 1 — Сравнение обычного SDLC и SSDLC применительно к СКЗИ

Критерий	**SDLC**	**SSDLC**
Готовность продукта	Функции реализованы, приемочные тесты пройдены	Функции и требования безопасности выполнены; доказательства воспроизводимы
Требования	Соответствие НПА рассматривается на этапе тестирования	Матрица применимости НПА и измеряемые требования уже на этапе проектирования
Архитектура	Оптимизация функциональности и производительности	Модель угроз, криптографическая граница, минимизация доверенной базы, безопасные отказ и обновление
Код и зависимости	Общий code review и функциональные тесты	Независимый анализ критичного кода, SAST/SCA, контроль секретов, SBOM и разрешенный состав зависимостей
Сборка и выпуск	Артефакт формируется типовым CI/CD	Изолированная и воспроизводимая сборка, подпись, снятие хэш-значения, манифест и прослеживаемость версии

Критерий	SDLC	SSDLC
Испытания	Преимущественно после завершения реализации, тестирование стабильности работы	Проверки на каждом этапе: негативные тесты, fuzzing, тесты ключевого цикла и независимая верификация
Сопровождение	Исправление дефектов по обращениям	Мониторинг уязвимостей и стандартов, анализ влияния изменений, регрессия и управляемый выпуск обновлений

SSDLC не исключает итеративную, гибкую или каскадную организацию работ. Он дополняет критерии завершения этапа проверкой безопасности принятых решений. Такой подход позволяет совершить Shift Left, что означает перенос данных решений на более ранние этапы разработки, упрощая исправление ошибок и ведение документации.

3. Модель SSDLC для разработки СКЗИ

Предлагаемая модель включает шесть взаимосвязанных этапов и контрольные точки. Состав работ уточняется с учетом вида СКЗИ, требуемого уровня безопасности, среды эксплуатации и применимой схемы оценки соответствия.

3.1. Требования безопасности. Формируется матрица применимости нормативных документов: назначение средства, условия применения, требуемый уровень безопасности согласно СТБ 34.101.27, используемые алгоритмы и протоколы, требования к генерации случайных чисел, жизненному циклу ключей, криптографической границе, регистрации событий, обновлению и восстановлению. Каждое требование задается в проверяемой форме, определяется конкретный исполнитель, способ подтверждения и ссылка на будущий протокол испытаний.

3.2. Безопасное проектирование. Разрабатываются модели нарушителя и угроз для самого продукта и среды разработки. В состав защищаемых объектов включаются ключевая информация, источники энтропии, исходные тексты, сценарии сборки, бинарные файлы и инфраструктура обновления. Определяются границы доверия, минимизируется доверенная программная база, проектируются очистка чувствительных данных, защита от отката версии, безопасное состояние при ошибке исполнения, контроль целостности и исключение выдачи секретов. Криптографические примитивы выбираются из нормативно установленных; самостоятельные алгоритмы допустимы только при отдельном научном обосновании и независимом анализе.

3.3. Безопасная реализация и цепочка поставок. Для изменений, затрагивающих критические участки кода, применяется независимый обзор кода. В конвейер включаются статический анализ (SAST), композиционный анализ зависимостей (SCA), поиск секретов, проверка лицензий и формирование перечня компонентов (SBOM). Версии зависимостей фиксируются, проверяются их источники и контрольные характеристики. Репозитории защищаются разграничением прав, обязательным review и подписанными тегами. Сборочная среда изолируется, а процедура сборки описывается как контролируемый и по возможности воспроизводимый процесс.

3.4. Верификация безопасности. Функциональные тесты дополняются тестами известных ответов для криптографических алгоритмов, негативными и граничными сценариями, фаззингом интерфейсов и форматов, проверкой управления ключами, отказоустойчивости, обновления и восстановления. Для реализаций с повышенным уровнем безопасности, анализируются временные и иные побочные каналы.

Результаты SAST, SCA, динамического анализа и ручной экспертизы объединяются в единый реестр замечаний. Методики и протоколы должны быть связаны с исходными требованиями и применяться к идентифицированному образцу.

3.5. Безопасный выпуск и подготовка к оценке соответствия. После прохождения контрольной точки фиксируется кандидат на выпуск: бинарные файлы, исходная версия, зависимости, параметры компилятора, скрипты сборки и документация. Формируются подписанный манифест, хэш-значения, ведомость комплектности и доказательство соответствия исходных текстов поставляемому образцу. После отбора образцов любые изменения выполняются только через процедуру оценки влияния и, при необходимости, повторные испытания.

3.6. Безопасное сопровождение. Организуется прием сообщений об уязвимостях, мониторинг зависимостей, изменений стандартов и условий эксплуатации, классификация инцидентов, выпуск исправлений и регрессионное тестирование. Для каждого изменения определяется, влияет ли оно на характеристики, подтвержденные при экспертизе или сертификации. Архивируются исходные тексты, инструменты и материалы, позволяющие воспроизвести ранее выпущенную версию.

Для управления полнотой доказательств предлагается использовать матрицу сквозной прослеживаемости «нормативное требование → архитектурное решение→ элемент реализации → тест → выпускной артефакт». Ее формальный показатель может быть задан выражением:

$$K_{\text{ТР}} = N_{COV} / N_{APP},$$

где N_{APP} — число применимых требований, N_{COV} — число требований, для которых существуют утвержденное решение и положительное подтверждение. Для допуска к выпуску должно выполняться $K_{\text{ТР}}$=1, а критические и высокие риски должны быть устранены или минимизированы.

4. Международный опыт

Международная практика предполагает разделение требований к процессу разработки и криптографической поддержке. NIST SP 800-218 (SSDF 1.1) предлагает набор высокоуровневых практик, которые могут быть встроены в любую реализацию SDLC. Суть практик в организации процесса разработки для минимизации риска несанкционированного изменения, контроля безопасности программного обеспечения и реагирования на уязвимости после этапа развертывания [11]. ISO/IEC 27034-1 рассматривает интеграцию безопасности в процессы управления приложениями, включая собственную и заказную разработку [12]. Существует некоммерческая организация OWASP, занимающаяся формализацией требований к безопасной разработке программного обеспечения (далее — ПО).

Для криптографических модулей ISO/IEC 19790:2025 устанавливает четыре уровня безопасности и требования по одиннадцати областям, а ISO/IEC 24759:2025 определяет объективные методы испытаний и состав подтверждающих материалов, предоставляемых изготовителем лаборатории [13, 14]. Данная модель полностью соотносится с существующим процессом разработки и сертификации, что позволят внедрять ее без крупных изменений организации тестирования.

В Российской Федерации подход закреплен в ГОСТ Р 56939-2024, устанавливающем общие требования к созданию безопасного программного обеспечения и устранению уязвимостей [15]. ГОСТ Р 58412-2019 систематизирует угрозы, возникающие в среде и в ходе разработки [16], а ГОСТ Р 71207-2024 уточняет порядок внедрения и выполнения статического анализа [17]. Для шифровальных (криптографических) средств Положение ПКЗ-2005, утвержденное приказом Федеральной службы безопасности Российской Федерации от 9

февраля 2005 г. № 66, охватывает разработку, производство, реализацию и эксплуатацию [18]. Таким образом, российский опыт также объединяет управление процессом, специальные требования к СКЗИ и независимый контроль.

Прямое копирование зарубежных моделей нецелесообразно в силу отличий НПА. Однако универсальные практики — моделирование угроз, защита репозитория и сборки, независимый анализ, прослеживаемость, управление уязвимостями не зависят от конкретных требований и НПА, что позволяет их внедрять в любой процесс разработки.

5. Эффект внедрения и ограничения подхода

Главное преимущество SSDLC состоит не в увеличении количества проверок, а в изменении момента принятия решений. Архитектурные несоответствия находятся на этапе проектирования за счет работы специалиста по безопасности, уже на этом этапе риск провала испытаний снижается за счет закладывания требований с самого начала работы, идентификация образца упрощается благодаря контролю конфигурации и воспроизводимости, изменения после развертывания отслеживаются по контрольным характеристикам. За счет данного контроля усложняется проведение атаки на цепочку поставок, так как подобные изменения будут заметны в процессе непрерывной интеграции и непрерывного развертывания (CI/CD).

Результативность внедрения оценивается не самим фактом принятия решений, а их результатом. Безопасность ПО невозможно организовать только наличием локальных актов, здесь требуются конкретные решения всех участников процесса разработки. Результаты, полученные в рамках проведенного исследования применимости данного подхода приведены в таблице 2. Целевые значения устанавливаются политикой организации с учетом требований безопасности, однако для показателей полноты и отсутствия дефектов, с которыми недопустимо выпускать продукт, применимы однозначные критерии.

Таблица 2 — Показатели результативности SSDLC для СКЗИ

Показатель	**Содержание**	**Критерий перед выпуском**
K_{TP}	Доля применимых требований, связанных с решением и положительным подтверждением	1.0
D_{BLOCK}	Количество неустраненных критических и высоких дефектов без принятого решения	0
R_{BUILD}	Доля выпускных сборок, воспроизводимых из зафиксированных исходных данных и инструментария	100%
C_{DEP}	Доля зависимостей с утвержденным источником, версией и контрольной характеристикой	100%
C_{REG}	Доля измененных требований безопасности, покрытых регрессионными испытаниями	100%
T_{FIX}	Время от подтверждения уязвимости до корректирующего выпуска	Не более срока, установленного политикой риска

SSDLC требует затрат на обучение, участие специалистов по безопасности, инструменты анализа и защищенную инфраструктуру разработки. Избыточно формализованный процесс способен замедлить небольшую команду. Поэтому глубина контроля должна зависеть от риска

и уровня СКЗИ, повторяемые проверки следует автоматизировать в CI/CD, а бумажную работу следует оптимизировать в пределах разумного. Минимально необходимый набор включает нормативную матрицу, модель угроз, управление изменениями, независимый обзор критичного кода, SAST/SCA, идентифицируемую сборку, связанные с требованиями испытания и процедуру реагирования на уязвимости.

Сам по себе SSDLC не является гарантией качественного результата, он является именно методологией, позволяющей упростить и оптимизировать процесс обеспечения безопасности итогового продукта. Можно выстроить идеальный, отлаженный процесс разработки, но при этом оставить личный ключ в списке открытых объектов или применить уязвимую реализацию криптографического алгоритма, что уничтожит безопасность продукта. Требования процесса лишь позволяют обнаружить подобные проблемы на более ранних этапах.

Заключение

Для СКЗИ безопасная разработка должна рассматриваться как часть нормативно-технического обеспечения продукта. Требования НПА формируют не только набор характеристик готового средства, но и фактические требования к прослеживаемости, документации, управлению конфигурацией, идентификации образца и сопровождению. Применение этих требований в начале проектирования и контроль качества на каждом этапе позволяет получать более предсказуемый результат, без масштабных изменений в конце разработки.

Предложенная модель SSDLC объединяет формирование требований, безопасное проектирование, контролируемую реализацию и цепочку поставок, верификацию, выпуск и сопровождение. Практическое внедрение целесообразно начинать с рассмотрения текущего процесса разработки и поиска мест, где не хватает присутствия инженера по информационной безопасности. Изменения стоит отслеживать по вычисляемым метрикам, организационные изменения сами по себе не дадут никакого результата.

Список литературы

1. Технический регламент Республики Беларусь «Информационные технологии. Средства защиты информации. Информационная безопасность» (ТР 2013/027/BY) утв. постановлением Совета Министров Республики Беларусь от 15 мая 2013 г. № 375.
2. Положение о порядке проведения государственной экспертизы средств технической и криптографической защиты информации утв. приказом Оперативно-аналитического центра при Президенте Республики Беларусь от 26 авг. 2013 г. № 60.
3. Перечень государственных стандартов, взаимосвязанных с техническим регламентом Республики Беларусь «Информационные технологии. Средства защиты информации. Информационная безопасность» (ТР 2013/027/BY) утв. приказом Оперативно-аналитического центра при Президенте Республики Беларусь от 12 мар. 2020 г. № 77.
4. СТБ 34.101.27-2022. Информационные технологии и безопасность. Средства криптографической защиты информации. Требования безопасности: утв. постановлением Госстандарта Республики Беларусь от 03.03.2022 № 23. — Введ. 03.03.2022. — Минск: Госстандарт, 2022.
5. СТБ 34.101.31-2020. Информационные технологии и безопасность. Криптографические алгоритмы шифрования и контроля целостности: утв. постановлением Госстандарта Республики Беларусь от 01.10.2020 № 56. — Введ. 01.10.2020. — Минск: Госстандарт, 2020.
6. СТБ 34.101.45-2013. Информационные технологии и безопасность. Алгоритмы электронной цифровой подписи и транспорта ключа на основе эллиптических кривых: утв.