



SmartCard-Service

Акционерное общество «СмартКард-Сервис»

127106, г. Москва, Алтуфьевское шоссе, д. 1

Телефон: +7 (495) 981-12-10, 8 (800) 100-31-64, факс: +7 (495) 981-12-11

E-mail: reception@scserv.ru, site: www.scserv.ru

УТВЕРЖДЕНО

Генеральный директор

АО «СмартКард-Сервис»

_____ В.А. Васильев

№ _____ «_____» _____ 20__ г.

Программное обеспечение «AT-NFC»

ПОЛОЖЕНИЕ О КОНФИГУРАЦИОННОМ УПРАВЛЕНИИ ПРИ РАЗРАБОТКЕ ПО И ДОКУМЕНТАЦИИ

Файл: положение о конфигурационном управлении.doc

СОГЛАСОВАНО

Технический директор

_____ В.В. Петров

«_____» _____ 20__ г.

СОГЛАСОВАНО

Заместитель

Генерального директора

_____ О.В. Андряков

«_____» _____ 20__ г.

Москва
2023

СОДЕРЖАНИЕ

1. Сведения о документе.....	3
2. Список терминов и сокращений	3
3. Введение.....	4
3.1. Цель документа	4
3.2. Сфера применения	4
3.3. Нормативная основа	4
3.4. Процедура сопровождения Положения.....	4
4. Основные положения.....	5
5.1. Цели конфигурационного управления	5
5.2. Участники конфигурационного управления.....	5
5.3. Использование Положения о конфигурационном управлении.....	6
5.4. Идентификация объектов конфигурационного управления	7
5.5. Структура информации в базе данных КУ.....	7
5.6. Версии и модификации	8
5.7. Распределение прав доступа и обязанностей.....	9
5.8. Интеграция проекта	10
5. Процедуры конфигурационного управления	11
6.1. Открытие нового проекта КУ	11
6.2. Конфигурационное управление в процессе разработки	11
6.3. Формирование версии проекта.....	12
6. Инструкции участников КУ	13
6.1. Архитектор проекта.....	13
6.2. Проектировщик.....	13
6.3. Менеджер конфигураций.....	14
6.4. Интегратор.....	14
6.5. Разработчик	14
6.6. Группа Контроля Качества (ГКК)	15
6.7. Менеджер по качеству	15
7. История изменений документа	16

1. СВЕДЕНИЯ О ДОКУМЕНТЕ

Номер версии:	01.00
Дата выпуска:	18.12.2023 г.
Дата утверждения:	
Порядок обновления:	1 раз в год

2. СПИСОК ТЕРМИНОВ И СОКРАЩЕНИЙ

Сокращение, термин	Расшифровка сокращения или термина
КУ	Конфигурационное Управление
ОКУ	Объект Конфигурационного Управления
План КУ	План конфигурационного управления проектом
ПО	Программное обеспечение
Проект	Проект разработки программного обеспечения
Подпроект	Проект, являющийся частью проекта
Проект КУ	Проект, состоящий из объектов КУ в базе данных конфигурационного управления
Подпроект КУ	Проект КУ, являющийся составной частью общего проекта КУ в базе данных конфигурационного управления
Положение	Положение о конфигурационном управлении при выполнении проектов разработки прикладного программного обеспечения и документации
ГКК	Группа Контроля Качества
Роль	Множество обязанностей, которое возлагается на сотрудника в рамках проекта. Один сотрудник может совмещать несколько ролей в проекте. Также одну роль в проекте могут выполнять несколько специалистов. В этом случае, группа специалистов, выполняющая одну роль, должна быть структурирована с выделением ведущего члена рабочей группы, ответственного за организацию работ по данному направлению

3. ВВЕДЕНИЕ

3.1. Цель документа

Целью «Положения о конфигурационном управлении при разработке программного обеспечения и документации» (далее — Положение) является регламентация процедур конфигурационного управления при разработке и сопровождении создаваемого прикладного программного обеспечения с целью обеспечения необходимого уровня качества проводимых работ.

3.2. Сфера применения

Положение предназначено для использования в области разработки прикладного программного обеспечения и документации.

Положение является частью документационного обеспечения процесса разработки программного обеспечения.

Утвержденное Положение имеет статус внутреннего стандарта и обязательно для исполнения во всех проектах разработки прикладного программного обеспечения.

3.3. Нормативная основа

В качестве нормативной основы при разработке данного Положения использован стандарт: IEEE Std 1042-1987 «IEEE Guide to Software Configuration Management».

3.4. Процедура сопровождения Положения

Положение о конфигурационном управлении при выполнении проектов разработки прикладного программного обеспечения разрабатывается в Группе Контроля Качества и утверждается Генеральным Директором.

Сопровождение и версионный контроль Положения осуществляет Группа Контроля Качества. Номер версии присваивается в процессе ввода Положения в действие.

Группа Контроля Качества осуществляет сбор предложений и замечаний, которые формируются в ходе выполнения проектов разработки программного обеспечения. По результатам обобщения и анализа предложений и замечаний разрабатываются новые версии Положения.

При исправлении ошибок или несоответствий, Положению присваивается следующий по порядку вспомогательный номер версии (после разделительной точки); при изменении и вводе в действие новых элементов организации или технологии работ по конфигурационному управлению Положению присваивается следующий по порядку основной номер.

При вводе в действие новой версии Положения, заинтересованные сотрудники уведомляются о выпуске новой версии по электронной почте.

4. ОСНОВНЫЕ ПОЛОЖЕНИЯ

4.1. Цели конфигурационного управления

Целью конфигурационного управления является:

- обеспечение целостности разрабатываемого ПО;
- обеспечение возможности контроля изменений ПО;
- поддержка процессов сопровождения версий разрабатываемого ПО;
- обеспечение детального контроля над процессом разработки.

4.2. Участники конфигурационного управления

Для описания процессов конфигурационного управления выделяются следующие роли.

Менеджер проекта — лицо, ответственное за управлением проектом. Менеджер проекта контролирует выполнение данного Положения и дисциплины конфигурационного управления при выполнении проекта.

Архитектор проекта — лицо, ответственное за ежедневное управление проектом, командой проекта. Он также несет ответственность за полноту и целостность состояния разрабатываемого проекта в целом.

Проектировщик — разработчик, член рабочей группы, ответственный за разработку подпроекта. Он несет ответственность за полноту и целостность состояния разрабатываемого подпроекта в базе данных конфигурационного управления. Для небольших проектов, не требующих введения руководителей подпроектов, их обязанности по настоящему Положению выполняет Архитектор проекта.

Разработчик — член рабочей группы, непосредственно ведущий разработку. В настоящем Положении под разработчиками также понимаются специалисты, разрабатывающие документацию, инсталляционное программное обеспечение и т.д., то есть все сотрудники, принимающие непосредственное участие в создании программного продукта. Разработчики — несут ответственность за ведение всех работ по проекту в рамках дисциплины конфигурационного управления.

Менеджер по управлению конфигурациями — член рабочей группы, ответственный за установление конфигурационной политики и среды конфигурации. Контролирует выполнение дисциплины конфигурационного управления всеми участниками проекта.

Интегратор — член рабочей группы, ответственный за обеспечение результирующей сборки программного продукта. Интегратор должен быть информирован о состоянии разработки в целом. Он обеспечивает единый подход к стандартизации имен ОКУ, следит за целостностью и непротиворечивостью информации в базе данных конфигурационного управления. Проводит анализ измененного программного кода по отношению к предыдущему варианту. Контролирует выполнение дисциплины конфигурационного управления всеми участниками проекта.

Тестировщик — член рабочей группы, осуществляющий разработку тестов и ведущий тестирование разрабатываемого продукта. Разработанные тесты являются частью создаваемого программного продукта и должны находиться под конфигурационным контролем наряду с файлами разработчиков.

Группа контроля качества — специалисты, которые несут ответственность за сопровождение и сохранность базы данных конфигурационного управления, методологическую поддержку разработок и контроль качества.

Менеджер по качеству — член рабочей группы, осуществляющий контроль качества разрабатываемого ПО.

4.3. Использование

Положения о конфигурационном управлении

Настоящий документ содержит положения общие для всех проектов разработки ПО. На основе данного положения для каждого проекта должен быть разработан план конфигурационного управления, в котором будут конкретизироваться пункты Положения.

План конфигурационного управления разрабатывает участник проекта, ответственный за целостность информации о разрабатываемом продукте — Менеджер по управлению конфигурациями, согласовывает его с Архитектором проекта и Менеджером по качеству, и утверждает его у Генерального директора.

План конфигурационного управления должен соответствовать настоящему Положению, как стандарту компании. При разработке плана конфигурационного управления необходимо конкретизировать те пункты Положения, которые требуют уточнения.

План конфигурационного управления для проекта разрабатывается с целью:

- утвердить стандарт именования ОКУ;
- утвердить собственную процедуру формирования версий проекта;
- определить структуру подпроектов общего проекта и их ответственных исполнителей;
- назначить сотрудника (Интегратора), ответственного за сборку разрабатываемого продукта и конфигурационный контроль проекта;
- распределить полномочия доступа членов рабочей группы к подпроектам КУ в базе данных конфигурационного управления;
- определить платформы разработки и функционирования продукта;
- зафиксировать инструментальные средства разработки, используемые библиотеки программ, компоненты и т.п. с обязательным указанием применяемых версий, а также другой информации, необходимой для однозначного определения конфигураций подпроектов КУ;
- определить порядок сборки и развертывания продукта;
- определить общие настройки и ограничения использования средств конфигурационного управления;
- определить регулярность и условия актуализации информации в базе данных конфигурационного управления.

В плане конфигурационного управления могут присутствовать все или ряд из перечисленных пунктов, а также могут быть решены другие организационные и технологические задачи проекта, связанные с конфигурационным управлением.

Таким образом, в целом, регламентация процедур конфигурационного управления для каждого проекта осуществляется на основе двух документов: настоящего Положения и Плана конфигурационного управления. При этом пункты Положения, которые носят рекомендательный характер, будут являться обязательными для выполнения, если иное специально не оговорено в Планах конфигурационного управления.

4.4. Идентификация объектов конфигурационного управления

Объект конфигурационного управления — единица информации, поставленная под контроль конфигурационного управления. В качестве ОКУ рекомендуется принимать файлы. ОКУ подвергаются изменениям в процессе разработки ПО. При этом с помощью средства конфигурационного управления регистрируется история изменений и поддерживается целостность версий разрабатываемого продукта, состоящего из множества файлов.

Для каждого проекта выбирается свой набор ОКУ в зависимости от особенностей проекта и используемых средств разработки.

Набор ОКУ должен содержать полную информацию о разработке, то есть информацию, которая *полностью* и *однозначно* определяет состояние разрабатываемого продукта. Полный набор ОКУ для всех проектов разработки ПО должен храниться и поддерживаться в актуальном состоянии в базе данных конфигурационного управления.

При этом в базе данных КУ рекомендуется хранить только первичные файлы разработки, определяющие состояние разрабатываемого программного продукта: файлы исходных кодов, экранных форм, скрипты настроек и т.п. Производные файлы, то есть файлы, которые полностью определяются исходной информацией (например, продукты компиляции) ставить под конфигурационный контроль не рекомендуется.

Ответственность за полноту и актуальность информации по проекту, хранящейся в базе данных КУ, лежит на архитекторе проекта. Разработчики несут ответственность перед архитектором проекта за постановку под конфигурационный контроль разрабатываемых ими файлов в базе данных КУ. Разработчикам рекомендуется ежедневно в конце рабочего дня (если иное не оговорено в плане конфигурационного управления проектом) записывать в базу данных КУ новые версии модифицируемых файлов.

Каждый ОКУ должен иметь *уникальное mnemonic имя* в проекте. Следует отметить, что в процессе работы многие файлы могут иметь разделяемое представление в проектах КУ. Использование одинаковых имен в различных подпроектах может затруднить работу.

Именованье файлов должно быть стандартизировано. Причем, чем больше проект и чем больше разработчиков участвует в работе, тем более жесткой должна быть стандартизация. Уникальность и мнемоника имен файлов в больших проектах обеспечивается стандартом именования, утвержденным в плане конфигурационного управления проектом. В проектах, не имеющих утвержденного стандарта именования, уникальность и мнемоника имен обеспечивается Интегратором проекта.

В базе данных КУ не должно содержаться лишней информации, то есть файлов, не относящихся к формированию разрабатываемого продукта. Возможность версионного хранения информации, предусмотренная в других программных средствах, рекомендуется не применять при использовании автоматизированного средства конфигурационного управления.

4.5. Структура информации в базе данных КУ

Информация в системе конфигурационного управления организуется в иерархическом виде по проектам КУ. Проект в контексте КУ представляет собой совокупность, связанных по условиям разработки, файлов и имеет аналогию с понятием каталогов файловой системы. Проекты КУ имеют иерархическую вложенную структуру.

Для проекта разработки в соответствии с архитектурой разрабатываемой системы должна использоваться структура проектов КУ, состоящая из нескольких уровней. Первый — головной уровень, который содержит проект КУ всей системы. Проект включает в себя ОКУ, обслуживающие весь проект, и набор подпроектов, соответствующих составу физических подсистем.

Структура подпроектов должна соответствовать архитектуре разрабатываемой системы. Ввод подпроектов в головной проект разработки осуществляет архитектор проекта. Ввод следующих уровней подпроектов осуществляют ответственные за подпроекты члены рабочей группы — проектировщики. Таким образом, для каждого проекта (подпроекта) в базе данных КУ должен быть ответственный член рабочей группы, который ведет или возглавляет разработку в данном проекте (подпроекте). Именование подпроектов должно быть мнемоническим отражением архитектурных названий.

4.6. Версии и модификации

В процессе разработки в согласованные моменты времени в соответствии с календарным планом разработки производится сборка готовой версии проекта. Готовому для сборки проекту присваивается уникальный номер версии и номер сборки. Присвоение номера версии и проставление метки (label) на соответствующий проект КУ производится Интегратором.

Модифицируемому проекту присваивается новый номер версии, если была изменена функциональность проекта, или новый номер сборки, если изменения вносились только с целью исправления ошибок или улучшения нефункциональных характеристик ПО (эффективности, надежности и т.п.).

Версии обозначаются номерами, имеющими следующий формат XX.YY.Z...Z, где:

XX — основной номер;

YY — средний номер;

Z...Z — HotFix и номер сборки.

Нумерация версий и сборок начинается от нуля.

Таблица 1

Нумерация версий и сборок

Типы изменений	Критерии	Примеры	Изменения версии
High-impact	<p>Внесение в ПО изменений, удовлетворяющих <i>любому</i> из условий:</p> <ul style="list-style-type: none"> • внесены существенные изменения в продукт • внесены изменения в архитектуру, исходный код или компоненты (функции, ресурсы), которые обрабатывают чувствительные данные или взаимодействуют с ними 	<p>Примеры:</p> <ul style="list-style-type: none"> • крупное изменение в ПО • изменения в функциях, реализующих обработку чувствительных данных • изменения в функциях, влияющих на выполнение требований стандарта 	<p>Изменяется первая цифра (например, с 1.0.0 на 2.0.0)</p>

	<ul style="list-style-type: none"> добавлены новые поддерживаемые платформы, не заявленные в рамках прошлой оценки стандарта PCI SSS в оценку добавлены или удалены модули стандарта PCI SSS 	PCI SSS	
Low-impact	Внесение в ПО функциональных изменений, не затрагивающих критерии для типа изменений «High impact»	Примеры: <ul style="list-style-type: none"> добавление нового типа функциональности, не затрагивающего чувствительные данные обновление ПО, не затрагивающее функции обработки чувствительных данных перекompиляция исходного кода с использованием нового компилятора или новых настроек компилятора 	Изменяется первая или вторая цифра (например, с 1.0.0 на 1.1.0)
No impact	Исправление ошибок и уязвимостей в ПО, а также исправление уязвимостей в сторонних компонентах, не влияющее на безопасность ПО и не влияющие на выполнение требований стандарта PCI SSS	Примеры: <ul style="list-style-type: none"> изменение в пользовательском интерфейсе ПО исправление ошибок в ПО 	Изменяется инкрементируемая часть, не влияющая на сертификацию (например, с 1.0.5 на 1.0.6)
Administrative	Изменения, не связанные с разработкой ПО	Примеры: <ul style="list-style-type: none"> изменение наименования ПО изменение наименования вендора 	Версия ПО не меняется

В документе «План конфигурационного управления» может быть предусмотрен другой порядок присвоения версий готовому продукту.

После сборки дистрибутива программного продукта проводится его тестирование. В ходе тестирования используются только тестовые карты. По окончании тестирования установленное на устройствах программное обеспечение удаляется и никаким образом не может быть использовано в промышленной эксплуатации. Для установки на «боевых» устройствах используются дистрибутивы программного обеспечения.

4.7. Распределение прав доступа и обязанностей

В базе данных конфигурационного управления происходит разделение сред разработки программного обеспечения, сборки версий и тестирования. При разделении сред

используется принцип параллельных потоков в организации работы в среде версионного контроля. Для группы разработчиков определен свой поток разработки. Для сборки релизов будущего программного продукта определен общий поток - интеграционный.

В рамках своих потоков пользователям базы данных конфигурационного управления по каждому проекту КУ также присваиваются различные общие права доступа:

- право доступа только на чтение;
- право редактирования объектов, входящих в данный проект КУ;
- право редактирования самого проекта КУ, право полного контроля.

Права доступа для членов рабочей группы проекта определяет архитектор проекта.

По умолчанию права доступа к проекту КУ автоматически распространяются на все подчиненные подпроекты.

Разделение обязанностей происходит на уровне ролей.

4.8. Интеграция проекта

Обязательным условием целостности информации в базе КУ является выполнение интеграции (сборки) проекта только на основе информации, хранящейся в базе данных КУ.

5. ПРОЦЕДУРЫ КОНФИГУРАЦИОННОГО УПРАВЛЕНИЯ

В разделе приведены процедуры конфигурационного управления обязательные для выполнения всеми участниками проектов конфигурационного управления.

6.1. Открытие нового проекта КУ

Процедура открытия нового проекта КУ является обязательной для инициализации каждого нового проекта разработки программного обеспечения.

№	Действие	Исполнители	Требования
1.	Разработка плана конфигурационного управления проекта	Менеджер по управлению конфигурациями	План КУ должен соответствовать настоящему положению
2.	Согласование и утверждение плана КУ	Архитектор проекта, ГKK	План КУ должен быть согласован в ГKK и архитектором проекта, и утвержден Генеральным директором
3.	Введение нового проекта в базу КУ	ГKK	Имя проекта в базе КУ должно соответствовать идентификатору проекта.
4.	Формирование структуры подпроектов в базе КУ	Архитектор проекта	Структура подпроектов должна соответствовать архитектуре проекта.
5.	Регистрация прав доступа членов рабочей группы в базе КУ	ГKK	По запросу архитектора проекта выполняется корректировка распределения полномочий в течение проекта

6.2. Конфигурационное управление в процессе разработки

Данная процедура обязательна для регулярного выполнения в процессе разработки программного обеспечения.

№	Действие	Исполнители	Требования
1.	Ввод новых подпроектов и файлов в базу данных КУ	Архитекторы подпроектов, Разработчики	Имена подпроектов и файлов должны соответствовать стандарту именования или согласовываться с Интегратором проекта. Структура подпроектов должна соответствовать структуре, утвержденной в Плане конфигурационного управления для проекта
2.	Считывание файлов из базы данных КУ для редактирования (Check Out)	Разработчики	Операция выполняется только при необходимости редактирования файла
3.	Считывание файлов из базы данных КУ для использования	Разработчики	Операция выполняется только для использования требуемых версий файлов без их изменения
4.	Запись файлов после редактирования в базу данных КУ (Check In)	Разработчики	Операция выполняется для всех считанных файлов (Checked Out). Рекомендуется проводить эту операцию в конце рабочего дня. Иной порядок может быть утвержден в Плане конфигурационного управления для проекта

№	Действие	Исполнители	Требования
5.	Анализ измененного программного кода по отношению к предыдущему варианту	Интегратор	После внесения разработчиками изменений в программный код, интегратор выполняет процедуру анализа кода по отношению к предыдущему варианту. Выявляются изменения, сделанные в программном обеспечении, и они проверяются на соответствие требованиям безопасности, на предмет выявления «закладок», неустойчивых и критических алгоритмов. После анализа и утверждения изменений они попадают в сборку очередного релиза

6.3. Формирование версии проекта

Данная процедура обязательна для выполнения при завершении работы над проектом в целом.

№	Действие	Исполнители	Требования
1.	Интеграция проекта	Интегратор	Интеграция проекта должна осуществляться только из файлов, которые находятся в базе данных КУ
2.	Тестирование	ГКК, Тестировщики	Изменения, которые вносятся в файлы в период тестирования, должны записываться в базу данных КУ. Анализ исправлений и повторная интеграция их должна вестись из базы данных КУ
3.	Проверка целостности информации в базе данных КУ	Архитектор проекта, Интегратор	Проверка включает проверку целостности и непротиворечивости информации в базе данных КУ
4.	Квалификационные испытания разработанной системы	ГКК, Тестировщики	В данный период изменения информации в соответствующем проекте КУ запрещены. Если принято решение о необходимости внесения изменений в проект, то выполняется переход к п. 2
5.	Фиксация версии проекта	Архитектор проекта, ГКК	Присвоение версии должно соответствовать утвержденному плану КУ. При фиксации версии должен быть внесен комментарий

6. ИНСТРУКЦИИ УЧАСТНИКОВ КУ

В разделе приводится описание сфер ответственности участников конфигурационного управления.

6.1. Архитектор проекта

Контроль полноты информации — архитектор проекта должен обеспечить полноту информации о разрабатываемом продукте в базе данных КУ.

Контроль стандартизации — архитектор проекта должен контролировать выполнение стандартизации имен файлов, подпроектов, названий версий всеми участниками проекта и наличие необходимых комментариев для файлов и подпроектов в базе данных КУ.

Контроль интеграции — архитектор проекта должен обеспечить выполнение сборки разрабатываемой системы только на основе информации из базы данных КУ.

Структура подпроектов — архитектор проекта должен обеспечить соответствие структуры подпроектов в базе данных КУ архитектурным решениям разрабатываемой системы.

Распределение прав доступа — архитектор проекта должен поддерживать в актуальном состоянии распределение прав доступа к базе данных КУ среди членов рабочей группы. При изменении в составе рабочей группы или изменении распределения обязанностей, Архитектор проекта должен своевременно подавать запрос на изменение прав доступа в ГKK.

Контроль версий — архитектор проекта должен обеспечить целостность и непротиворечивость фиксируемых версий в базе данных КУ.

Контроль испытаний — архитектор проекта должен контролировать соблюдение правил конфигурационного управления при испытаниях готовой системы.

6.2. Проектировщик

Контроль полноты информации — на проектировщика возлагается ответственность за обеспечение постановки под конфигурационный контроль всех разрабатываемых файлов подпроекта.

Контроль интеграции — проектировщик должен обеспечить выполнение сборки разрабатываемого подпроекта только на основе информации из базы данных КУ.

Стандартизация — проектировщик должен поддерживать соответствие стандартам именования названий вводимых подпроектов и файлов.

Контроль лишней информации в базе данных КУ — проектировщик должен обеспечивать своевременное удаление ненужной информации из базы данных КУ.

Контроль версий — проектировщик должен обеспечить целостность и непротиворечивость фиксируемых версий подсистем в базе данных КУ.

6.3. Менеджер по управлению конфигурациями

Планирование — менеджер по управлению конфигурациями должен разработать План конфигурационного управления для проекта и поддерживать его в актуальном состоянии в течение всего жизненного цикла проекта.

Конфигурационная политика — менеджер по управлению конфигурациями устанавливает правила конфигурационного управления и контролирует соблюдение этих правил.

Среда конфигурации — менеджер по управлению конфигурациями настраивает проектный репозиторий и регулярно проводит его аудит по контролю его состояния.

6.4. Интегратор

Стандартизация — интегратор должен отслеживать выполнение стандартизации именования файлов и подпроектов в базе данных КУ.

Испытания — интегратор должен обеспечить отсутствие внесения изменений в проект в период испытаний.

Проверка целостности — интегратор должен организовать и провести проверку целостности и непротиворечивости информации о проекте в базе данных КУ.

Аналитика исходного кода — интегратор должен провести проверку исходного кода по отношению к предыдущему варианту на предмет выявления «закладок», неустойчивых и критических алгоритмов, а также на соответствие требованиям безопасности.

6.5. Разработчик

Актуализация — разработчик несет ответственность за регулярную актуализацию информации в базе данных КУ в соответствии с настоящим Положением и Планом конфигурационного управления проекта.

Полнота информации — разработчик несет ответственность за ведение всех разработок под конфигурационным управлением в базе данных КУ.

Стандартизация — разработчик должен поддерживать соответствие стандартам именования названий вводимых подпроектов и файлов.

Контроль лишней информации — разработчик должен не допускать хранение в базе данных КУ информации, не относящейся к проекту.

Процесс разработки приложений должен предупреждать возникновение общеизвестных уязвимостей программного кода, и включать следующее:

- Инъекции, в особенности, SQL-инъекции. Также следует учесть OS Command, LDAP и Xpath инъекции.
- Переполнение буфера.
- Небезопасное криптографическое хранилище.
- Небезопасная передача данных.
- Некорректная обработка ошибок.
- Все уязвимости, имеющие «Высокую» степень риска, найденные в процессе обнаружения уязвимостей.
- Межсайтовый скриптинг (XSS).

- Ошибки в контроле доступа (например, небезопасные прямые ссылки на объекты, невозможность ограничения доступа по URL и обход директорий).
- Подделка межсайтовых запросов (CSRF).

6.6. Группа Контроля Качества (ГКК)

Конфигурационное управление — ГКК обеспечивает организацию и поддержку системы конфигурационного управления при выполнении всех проектов разработки программного обеспечения.

Техническая поддержка — ГКК несет ответственность за техническую поддержку средств конфигурационного управления.

Сопровождение базы данных — ГКК несет ответственность за сопровождение и администрирование базы данных конфигурационного управления.

Испытания — ГКК организует и проводит квалификационные внутренние испытания разработанного программного обеспечения и участвует в проведении приемо-сдаточных испытаний.

Аудит — ГКК проводит аудит конфигурационного управления в соответствии с утвержденным планом работы.

6.7. Менеджер по качеству

Конфигурационное управление — менеджер по качеству обеспечивает организацию и поддержку системы конфигурационного управления при разработке программного обеспечения.

Испытания — менеджер по качеству проводит квалификационные внутренние испытания разработанного программного обеспечения.

Аудит — менеджер по качеству проводит аудит конфигурационного управления.

7. ИСТОРИЯ ИЗМЕНЕНИЙ ДОКУМЕНТА

Дата изменений	Версия док-та	Описание изменений
18.12.2023	01.00	Исходная редакция документа, разработанная с учетом требований PCI SSF и PCI SSS v.1.2.1