



## **Ред Платформа**

**Описание жизненного цикла и сопровождение продукта**

**46.98898398.506180-01 97 1-1**

Версия 1.6

Москва, 2019

## Перечень сокращений

<b>БД</b>	База данных
<b>ИБ</b>	Информационная безопасность
<b>ОС</b>	Операционная система
<b>ПК</b>	Персональный компьютер
<b>ПО</b>	Программное обеспечение
<b>РП</b>	Ред Платформа
<b>СУБД</b>	Система управления базами данных
<b>УК</b>	Управление конфигурацией

## Оглавление

1. Описание системы контроля версий .....	5
1.1. Возможности Subversion .....	5
1.2. Модель работы.....	6
1.3. Доступ к репозиторию.....	6
1.4. Рабочий цикл .....	7
1.5. Ветвление .....	7
1.6. Сборка программного кода РП .....	8
2. Элементы конфигурации РП .....	9
3. Маркировка документации .....	10
3.1. Требования к составу и маркировке программной документации .....	10
3.2. Процедура формирования серийного номера продукта .....	10
3.3. Правила присвоения серийного номера программного продукта .....	11
4. Система отслеживания ошибок .....	13
4.1. Общее описание.....	13
4.2. Техническая архитектура.....	13
4.3. Общий алгоритм обновление РП .....	16
4.4. Обновление изделия с комплекта поставки на физических носителях ....	16
4.5. Обновление изделия с использованием репозитория .....	17
4.6. Порядок установки критических обновлений .....	18
4.7. Характеристики репозитория изделия.....	19
4.8. Верификация изделия.....	20
5. Способы распространения изделия .....	22
6. Описание штатной структуры проекта .....	23

## Введение

В настоящем документе представлены описания системы управления конфигурацией, которая используется в ООО «РЕД СОФТ» при разработке Ред Платформы (далее – РП, платформа), список конфигурации и метод уникальной идентификации элементов конфигурации.

Управление конфигурацией помогает обеспечить сохранение целостности РП, устанавливая и контролируя определенный порядок процессов уточнения и модификации РП и предоставления связанной с ними информации. УК предотвращает несанкционированную модификацию, добавление или уничтожение составляющих РП, обеспечивая тем самым доверие, что оцениваются именно те РП и документация, которые подготовлены к распространению.

Управление конфигурацией - один из методов или способов установить, что в созданном РП реализованы функциональные требования и спецификации. УК отвечает этим целям, предъявляя требования дисциплины и контроля в процессе уточнения и модификации РП и связанной с ним информации. Системы УК используют для обеспечения целостности частей РП, которые они контролируют, предоставляя метод отслеживания любых изменений, и для того, чтобы все изменения были санкционированы.

## 1. Описание системы контроля версий

При разработке РП используется система управления версиями Subversion.

Subversion (также известная как «SVN») — свободная централизованная система управления версиями, официально выпущенная в 2004 году компанией CollabNet Inc. Цель проекта — заменить собой распространенную на тот момент систему Concurrent Versions System (CVS), которая ныне считается устаревшей. Subversion реализует все основные функции CVS и свободна от ряда недостатков последней.

### 1.1. Возможности Subversion

Основными функциональными возможностями Subversion являются:

- хранение полной истории изменений отслеживаемых объектов (файлов, каталогов, символьных ссылок) в централизованном хранилище (репозитории), в том числе при изменении атрибутов («метаданных»), перемещении, переименовании и удалении;
- копирование объектов с разветвлением истории — при копировании в хранилище появляются два отдельных объекта с общей историей;
- поддержка переноса изменений между копиями объектов, в том числе полного слияния копий (в рабочей копии, без объединения истории);
- поддержка ветвления: создания ветвей (копированием директорий) и работы с ними или слияние ветвей (переносом изменений);
- поддержка меток (копированием директорий);
- история изменений и копии объектов (в том числе ветви и метки) хранятся в виде связанных разностных копий при создании и хранении;
- поддержка конкурентной (в том числе одновременной, с изоляцией транзакций) многопользовательской работы с хранилищем и, в большинстве случаев, автоматическим слиянием изменений различных разработчиков (в рабочей копии);
- фиксации изменений в хранилище (в том числе многообъектные) организуются в виде атомарных транзакций;
- сетевой обмен между сервером и клиентом предусматривает передачу только различий между рабочей копией и хранилищем;
- обеспечивается одинаково эффективная работа как с текстовыми, так и с двоичными файлами.
- различные варианты доступа к хранилищу, в том числе: непосредственный доступ на локальной файловой системе, по собственному сетевому протоколу,

через веб-сервер по протоколу WebDAV/DeltaV.

- вывод клиента командной строки одинаково удобен и для чтения, и для разбора программами;
- возможность зеркалирования хранилища;
- два возможных внутренних формата хранилища: база данных или набор обычных файлов;
- интернационализированные сообщения программы (используются настройки локали);
- библиотеки для языков PHP, Python, Perl, Java позволяют встроить функциональность клиента Subversion в программы, написанные на этих языках;
- многоуровневая архитектура библиотек, изначально рассчитанная на клиент-серверную модель.

## 1.2. Модель работы

Subversion — централизованная система, то есть данные хранятся в едином хранилище. Хранилище может располагаться на локальном диске или на сетевом сервере.

Работа в Subversion мало отличается от работы в других централизованных системах управления версиями. Клиенты копируют файлы из хранилища, создавая локальные рабочие копии, затем вносят изменения в рабочие копии и фиксируют эти изменения в хранилище. Несколько клиентов могут одновременно обращаться к хранилищу. Для совместной работы над файлами в Subversion преимущественно используется модель копирование — изменение — слияние. Кроме того, для файлов, не допускающих слияние (различные бинарные форматы файлов), можно использовать модель блокирование — изменение — разблокирование.

При сохранении новых версий используется дельта-компрессия: система находит отличия новой версии от предыдущей и записывает только их, избегая дублирования данных.

## 1.3. Доступ к репозиторию

Subversion предоставляет следующие способы доступа к репозиторию:

- прямой доступ к репозиторию на диске (на локальной или сетевой файловой системе);
- удалённый доступ по протоколу WebDAV/DeltaV поверх HTTP (или HTTPS) с использованием модуля `mod_dav_svn` для веб-сервера Apache 2;
- удалённый доступ с использованием собственного протокола SVN: на

выделенном сетевом соединении (по умолчанию на TCP-порту 3690), через стандартный ввод-вывод (в том числе через средства удаленного CLI, например SSH).

Все эти способы могут быть использованы для работы с репозиториями. Для доступа к одному и тому же репозиторию могут одновременно использоваться разные способы.

#### 1.4. Рабочий цикл

Типичная итерация рабочего цикла с Subversion включает следующие этапы:

– обновление рабочей копии из хранилища (`svn update`) или ее создание (`svn checkout`);

– изменение рабочей копии. Изменения директорий и информации о файлах производится средствами Subversion, в изменении же (содержимого) файлов Subversion никак не задействован — изменения производятся программами, предназначенными для этого (текстовые редакторы, средства разработки и т. п.):

а) новые (еще не зафиксированные в хранилище) файлы и директории нужно добавить (команда `svn add`), то есть передать под управление версиями;

б) если файл или директорию в рабочей копии нужно удалить, переименовать, переместить или скопировать, необходимо использовать средства Subversion (`svn mkdir`, `svn delete`, `svn move`, `svn copy`);

в) просмотр состояния рабочей копии и локальных (ещё не зафиксированных) изменений (`svn info`, `svn status`, `svn diff`);

г) любые локальные изменения, если они признаны неудачными, можно откатить (`svn revert`).

– при необходимости — дополнительное обновление, для получения изменений, зафиксированных в хранилище другими пользователями и слияния этих изменений со своими (`svn update`);

– фиксация своих изменений (и/или результатов слияния) в хранилище (`svn commit`).

#### 1.5. Ветвление

Ветвление является важным аспектом работы систем управления версиями, поскольку типичные приёмы управления версиями (по крайней мере, при разработке программного обеспечения) подразумевают использование ветвей. Subversion обладает развитыми возможностями для ветвления и слияния (однако

не поддерживает слияние переименованных файлов и директорий).

На рисунке 1 условно показан пример эволюции ветвей в хранилище. Зелёным цветом показана основная линия разработки проекта, жёлтым — ветви, синим — метки, пурпурным — ветвь, разработка которой прекращена. Красными стрелками показаны слияния изменений.

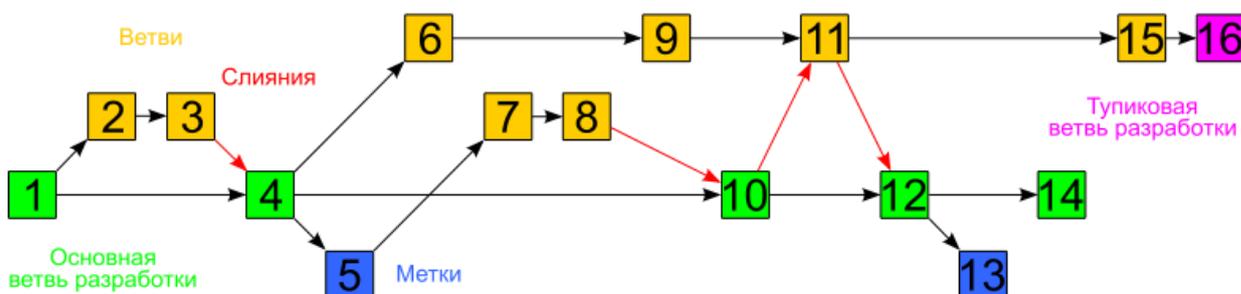


Рисунок 1 - Эволюция ветвей в хранилище

## 1.6. Сборка программного кода РП

Сборка программного кода РП производится на отдельной виртуальной ОС, размещенной на специальном выделенном сервере. Доступ к виртуальной ОС имеют только определенные лица, в круг обязанностей которых входит работа со сборкой программного кода или иных задач по контролю за работой над разработкой кода РП. Доступ осуществляется на основе логина и пароля пользователя.

Решение о необходимости сборки программного кода принимает руководитель отдела разработки после соответствующей проверки и тестирования всех компонентов РП.

Каждая сборка РП имеет уникальный идентификатор, состоящий из номера версии РП и номера сборки.

## 2. Элементы конфигурации РП

Элементами конфигурации РП являются:

- исходный код РП;
- проектная документация (пояснительные записки, частные технические задания, ведомости, программы испытаний, постановки задач, функциональная спецификация, проект верхнего уровня);
- тестовая документация;
- документация управления конфигурацией;
- руководство пользователя;
- руководство администратора.

### 3. Маркировка документации

#### 3.1. Требования к составу и маркировке программной документации

Виды, комплектность и обозначение документации, создаваемой и сопровождаемой в рамках разработки ОС, определяются ГОСТ 34.201-89 «Виды, комплектность и обозначение документов при создании автоматизированных систем».

Состав программной документации на различных этапах создания и сопровождения ОС определяется ГОСТ 19.101-77 «Виды программ и программных документов».

Маркировка программной документации на ОС соответствует ГОСТ 19.103-77 «Обозначения программ и программных документов» и представляет из себя следующую структуру:

А.В.СССС-DD EE FF-G,

где А – код страны разработчика (равен 46);

В – код организации-разработчика (равен 98898398);

СССС – регистрационный номер программного продукта по классификатору (равен 506180 – программные средства для автоматизации делопроизводства, документооборота и управленческой деятельности);

DD – номер издания (для программы) или номер редакции (для документа);

EE – код вида документа;

FF – номер документа данного вида;

G – номер части документа.

Например, документ «Руководство пользователя» маркируется следующим образом:

46.98898398.506108-01 34 1-1

#### 3.2. Процедура формирования серийного номера продукта

Потребители и заказчики формируют заявки на поставку комплектов РП. Заявки направляются в адрес отдела маркетинга ООО «РЕД СОФТ».

Отдел маркетинга ООО «РЕД СОФТ» рассматривает поступающие заявки и после рассмотрения и согласования заявки предоставляет руководителю отдела материально-технического обеспечения информацию о поступлении заявки. Сведения предоставляются по служебной электронной почте в виде приложения к

письму.

По информации в письме отдел материально-технического обеспечения, руководитель отдела материально-технического обеспечения формирует необходимое количество дистрибутивных комплектов программных продуктов, формирует необходимое количество серийных номеров программных продуктов, производит маркировку дистрибутивных комплектов программных продуктов, производит запись в журнале формирования и выдачи программных продуктов стенда тиражирования. Правила присвоения серийных номеров приведены в разделе 3.3.

Далее отдел материально-технического обеспечения формирует письму-уведомление для клиента о готовности дистрибутивных комплектов, после чего осуществляет доставку дистрибутивных комплектов оговоренными в заявке средствами.

### 3.3. Правила присвоения серийного номера программного продукта

Серийный номер состоит из набора атрибутов (таблица 1), отражающих информацию о продукте, его версии и дате выпуска дистрибутивного комплекта. В общей строке серийного номера атрибуты детерминируются разделительными символами – дефисами.

Серийный номер продукта имеет вид **AA-BBB-CCC-12345678-123456-123** и состоит из:

Тип серийного номера	-	Тип продукта	-	Редакция	-	Номер сборки	-	Дата формирования дистрибутивного комплекта	-	Порядковый номер комплекта за текущую дату
2 знака(2 буквы или буква и цифра)	-	3 буквы латинского алфавита	-	3 буквы латинского алфавита	-	8 цифр	-	6 цифр	-	3 цифры

#### ТИП СЕРИЙНОГО НОМЕРА

Может принимать следующие значения:

PR — серийный номер для продукта. Данный тип серийных номер присваивается при продаже продукта без продажи техподдержки.

S0 - S9 — серийный номер для технической поддержки. Данный тип серийных номеров присваивается при продаже техподдержки для продукта, то есть продажа техподдержки вместе с продуктом, или продажа техподдержки к ранее купленному продукту. Цифра, стоящая после буквы S определяет уровень технической поддержки клиента. Чем больше число, тем выше уровень поддержки. Стандартному уровню технической поддержки присваивается код S0, расширенному - S1. Чем больше опций приобретает клиентом, тем больше увеличивается число.

#### ТИП ПРОДУКТА

*Ред Платформа* - указывается: **NCR**

#### РЕДАКЦИЯ

*Open* - указывается: **OPN**

*Standart* - указывается: **STN**

*Enterprise* - указывается: **ENT**

*Developer* - указывается: **DEV**

*Отсутствует* - указывается: **NED**

#### НОМЕР СБОРКИ

Указывается 8 цифр номера сборки продукта без разделителей. Недостающие позиции заполняются нулями. Например: 00000052 или 2601234

#### ДАТА ФОРМИРОВАНИЯ СЕРИЙНОГО НОМЕРА

Указывается 6 цифр даты, без разделителей. Для чисел менее 10 указывается вместе с нулем. Год - указывается 2 последние цифры. Например: 6 февраля 2014 года — 060214.

#### ПОРЯДКОВЫЙ НОМЕР

Указывает порядковый номер заказанного продукта за текущий день. Например: 001, 002 или 018

#### ПРИМЕР СЕРИЙНОГО НОМЕРА:

S3-NCR-NED-00000073-100117-011

Серийный номер присвоен клиенту технической поддержки (расширенный уровень с дополнительными опциями) на продукт РП без указания редакции, версия 7.3, сформирован 10 января 2017 года, 11 заказ за день.

## 4. Система отслеживания ошибок

### 4.1. Общее описание

При разработке РП использована система отслеживания ошибок Redmine, предназначенная для учета и контроля ошибок, найденных в программах, пожеланий пользователей, а также для отслеживания процесса устранения этих ошибок и выполнения или невыполнения пожеланий.

Redmine — открытое серверное веб-приложение для управления проектами и задачами (в том числе для отслеживания ошибок).

Данный продукт предоставляет следующие возможности:

- ведение нескольких проектов;
- гибкая система доступа, основанная на ролях;
- система отслеживания ошибок;
- диаграммы Ганта и календарь;
- ведение новостей проекта, документов и управление файлами;
- оповещение об изменениях с помощью RSS-потоков и электронной почты;
- вики для каждого проекта;
- форумы для каждого проекта;
- учёт временных затрат;
- настраиваемые произвольные поля для инцидентов, временных затрат, проектов и пользователей;
- лёгкая интеграция с системами управления версиями (SVN, CVS, Git, Mercurial, Bazaar и Darcs);
- создание записей об ошибках на основе полученных писем;
- поддержка множественной аутентификации LDAP;
- возможность самостоятельной регистрации новых пользователей.

### 4.2. Техническая архитектура

Трекеры являются основной классификацией, по которой сортируются задачи в проекте. Само по себе понятие «трекер» восходит к системам учёта ошибок (англ. Bug tracking tool), представлявшим каждая в отдельности один проект.

По сути, в «Redmine» трекеры представляют собой аналог подклассов класса «Задача» и являются основой для полиморфизма разного рода задач, позволяя определять для каждого их типа различные поля. Примерами трекеров являются «Улучшение», «Ошибка», «Документирование», «Поддержка»,

Задачи являются центральным понятием всей системы, описывающим некую

задачу, которую необходимо выполнить. У каждой задачи в обязательном порядке есть описание и автор, в обязательном порядке задача привязана к трекеру.

Каждая задача имеет статус. Статусы представляют собой отдельную сущность с возможностью определения прав на назначение статуса для различных ролей (например, статус «отклонен» может присвоить только менеджер) или определение актуальности задачи (например, «открыт», «назначен» — актуальные, а «закрит», «отклонен» — нет).

Для каждого проекта отдельно определяются набор этапов разработки и набор категорий задач. Среди других полей интересны также «оцененное время», служащее основой для построения управленческих диаграмм, а также поле выбора наблюдателей за задачей. К задачам имеется возможность прикреплять файлы (имеется отдельная сущность «Приложение»).

Значения других перечислимых свойств (например, приоритетность) хранятся в отдельной общей таблице.

За отслеживание изменений параметров задач пользователями в системе отвечают две сущности: «Запись журнала изменений» и «Измененный параметр». Запись журнала отображает одно действие пользователя по редактированию параметров задачи и/или добавление комментария к ней. То есть служит одновременно инструментом ведения истории задачи и инструментом ведения диалога.

Сущность «Измененный параметр» привязана к отдельной записи журнала и предназначена для хранения старого и нового значения измененного пользователем параметра.

Задачи могут быть взаимосвязаны: например, одна задача является подзадачей для другой или предшествовать ей. Эта информация может быть полезна в ходе планирования разработки программы, за её хранение в Redmine отвечает отдельная сущность.

Система поддерживает учёт затраченного времени благодаря сущности «Затраченное время», связанной с пользователями и задачей. Сущность позволяет хранить затраченное время, вид деятельности пользователя (разработка, проектирование, поддержка) и краткий комментарий к работе. Эти данные могут быть использованы, например, для анализа вклада каждого участника в проект или для оценки фактической трудоемкости и стоимости разработки.

Redmine предоставляет возможность интеграции с различными системами управления версиями (репозиториями). Интеграция заключается в отслеживании изменений во внешнем репозитории, их фиксации в базе данных, анализе изменений с целью их привязки к определенным задачам. В инфологической структуре системы за интеграцию с внешними репозиториями отвечают три

сущности: Репозиторий, Редакция и Изменение.

Репозиторий — связанная с проектом сущность, хранящая тип подключенного репозитория, его местонахождение и идентификационные данные его пользователя.

Редакция — отображение редакции репозитория, и, кроме информационных полей, может быть привязана к конкретной задаче: для этого требуется указать в описании изменений «refs #NUM», где NUM — номер задачи), и к пользователю-автору редакции.

Изменение — хранит список измененных (добавленных, удаленных, перемещенных, модифицированных) файлов в каждой редакции.

Уведомления пользователей об изменениях, происходящих на сайте, осуществляется с помощью сущности «Наблюдатели», связывающей пользователей с объектами различных классов (проекты, задачи, форумы и др.). В базе данных хранятся также ключи доступа к подписке RSS, позволяющие получать уведомления посредством этой технологии, также уведомления рассылаются с помощью электронной почты.

Общая схема жизненного цикла задачи приведена на рисунке 2.

Задача регистрируется в рамках работы линии технической поддержки пользователей. Это может быть проблема, возникшая у пользователя или сообщение об ошибке от системы, обнаруженное пользователем в процессе эксплуатации РП. Так же задача может возникнуть в рамках работы группы разработки ПО или анализа ПО при его согласования с заказчиком продукта.

Далее, созданная задача поступает на рассмотрение руководителей отделов и ведущих аналитиков. Исходя из контекста, они принимают решение об отказе или назначении в работу поступившей задачи.

Принятая в работу задача поступает в группу тестирования для подтверждения факта обнаруженной ошибки или непосредственно в группу аналитики или разработки, если такое подтверждение не требуется. Если ошибка не будет подтверждена в результате тестирования, она отклоняется.

Подтвержденная тестировщиками ошибка поступает для согласования и утверждения в группу аналитики и далее непосредственным разработчикам программного продукта.

После доработки или разработки вновь задача возвращается на тестирование, и если будет подтверждено исправление обнаруженной ошибки или корректная работа созданного компонента, задача закрывается.



Рисунок 2 - Жизненный цикл задачи в системе отслеживания ошибок

#### 4.3. Общий алгоритм обновление РП

Обновления изделия, если при поставке обновлений не обговорено иное, осуществляются согласно условиям договора о поставке изделия и технического сопровождения в следующих вариантах:

- Производится полная деинсталляция текущей версии РП с последующей установкой обновленной сертифицированной версии РП в полном соответствии с программной документацией;
- Производится обновление с использованием сервисов цифровой дистрибуции и обновления репозитория сертифицированного изделия.

Оба варианта обновления должны завершаться процедурой верификации изделия, описанной в разделе 4.8.

#### 4.4. Обновление изделия с комплекта поставки на физических носителях

Производитель по результатам проведения оценок соответствия изделия в форме инспекционного контроля обеспечивает формирование комплектов поставки изделия.

Производитель направляет потребителям извещение об изменениях,

содержащее развернутый перечень изменений в изделии, или публикует данное извещение на официальном общедоступном ресурсе.

По запросу потребителя, производитель направляет в адрес потребителя дистрибутив в комплекте поставки, извещения об изменениях эксплуатационной документации изделия в бумажном виде, а также заверенную копию сертификата соответствия СЗИ с внесенными изменениями.

Потребитель в соответствии с эксплуатационной документацией на изделие обязан выполнить обновление изделия с использованием, полученного от производителя дистрибутивного комплекта поставки.

Потребитель после выполнения обновления обязан делать соответствующую отметку в эксплуатационной документации изделия с указанием типа, даты и времени обновления, а также с указанием фамилии лица, применившего его.

При возникновении нештатных ситуаций, сбоев и отказов в процессе установки или обновления изделия, потребитель, выявивший сбой, должен немедленно сообщить о проблеме производителю изделия путем обращения по телефону или другими доступными потребителю средствами связи.

#### 4.5. Обновление изделия с использованием репозитория

Производитель по результатам проведения оценок соответствия изделия в форме инспекционного контроля обеспечивает своевременное обновление репозитория изделия.

При обновлении и синхронизации уровней репозитория производитель использует только защищенные, т.е. доверенные каналы передачи данных.

Производитель публикует образы дистрибутивных дисков комплекта поставки изделия и бинарные установочные пакеты изделия в репозитории. Образы включают полный поставочный комплект документации на изделие и копию сертификата соответствия. При этом производитель обеспечивает усиленную квалифицированную электронную подпись образов дистрибутивных дисков и бинарных установочных пакетов комплекта поставки изделия квалифицированным сертификатом ключа проверки электронной подписи, выданным аккредитованным удостоверяющим центром.

После обновления изделия в репозиториях производитель направляет потребителям извещение об изменениях, содержащее развернутый перечень изменений в изделии. В случае необходимости, по запросу потребителя,

производитель дополнительно направляет в адрес потребителя комплект извещений об изменениях эксплуатационной документации изделия в бумажном виде, а также заверенную копию сертификата соответствия СЗИ с внесенными изменениями.

Порядок действий по синхронизации репозитория и обновлению СЗИ описан в эксплуатационной документации на изделие.

Потребитель после выполнения обновления обязан делать соответствующую отметку в эксплуатационной документации изделия с указанием типа, даты и времени обновления, а также с указанием фамилии лица, применившего его.

При возникновении нештатных ситуаций, сбоев и отказов в процессе загрузки, установки или обновления изделия, потребитель, выявивший сбой, должен немедленно сообщить о проблеме производителю изделия путем обращения по телефону или другими доступными потребителю средствами связи.

#### 4.6. Порядок установки критических обновлений

При возникновении необходимости установки критических обновлений производитель незамедлительно осуществляет доведение до потребителей информации о необходимости обновления и предоставляет возможность его получения по доверенному каналу.

Потребитель при получении указанной информации осуществляет получение обновления средства защиты информации и незамедлительно применяет его, о чем делает соответствующую отметку в эксплуатационной документации с указанием типа, даты и времени обновления, а также с указанием фамилии лица, применившего его.

При невозможности устранения уязвимостей средства защиты информации, в том числе путем применения обновления, производитель разрабатывает ограничения по применению средства защиты информации и согласовывает их с испытательной лабораторией. Если в соответствии с заключением испытательной лаборатории ограничение по применению позволит устранить уязвимость, производитель незамедлительно доводит его до потребителей. Потребитель реализует указанное ограничение по применению средства защиты информации. Если потребитель не может реализовать ограничение по применению средства защиты информации он прекращает его применение.

Производитель вносит необходимые изменения в эксплуатационную документацию и после завершения инспекционного контроля, получив во ФСТЭК

России сертификат соответствия с внесенными изменениями, доводит его копию, а также изменения в эксплуатационную документацию до всех потребителей.

#### 4.7. Характеристики репозитория изделия

Репозиторий изделия служит хранилищем дистрибутивных пакетов изделия, обеспечивает сервис онлайн-дистрибуции изделия для потребителей.

Физически репозиторий представляет собой набор сетевых ресурсов в сети передачи данных с идентифицированными для пользователей сетевыми именами. Идентификация доступного репозитория в сети передачи данных организуется на уровне сетевых протоколов.

Функционирование сервисов репозитория изделия в сети передачи данных осуществляется по следующим сетевым протоколам и портам:

- протокол HTTP, порт 80. На данном порту обеспечивается доступ потребителей к ресурсам репозитория и получение обновлений;
- протокол HTTPS, порт 443. Данный порт зарезервирован для обеспечения аутентификации и построения защищенного TLS-соединения на основании сертификата открытого ключа пользователя с использованием инфраструктуры PKI;
- протокол RSYNC, порт 873. Данный порт предназначен для обеспечения процессов синхронизации зеркалируемых репозиториях изделия.

Функции репозитория изделия заключаются в обеспечении следующих сервисов:

- сервис онлайн-дистрибуции изделия;
- сервис обновления изделия;
- сервис синхронизации зеркалируемых репозиториях;
- сервис верификации изделия.

Сервис онлайн-дистрибуции обеспечивает полную идентичность бинарных образов изделия, представленных на физических цифровых носителях информации (флеш-накопители, CD/DVD диски и т.д.), и бинарных образов изделия, размещаемых в ресурсах репозитория производителя и загружаемых потребителями по сети передачи данных.

Сервис обновления изделия круглосуточно обеспечивает обновление изделия для конечных потребителей изделия.

Сервисы онлайн-дистрибуции и обновления изделия обеспечивают для конечных потребителей однозначную идентификацию изделия по сборкам, версиям, релизам, точкам монтирования ресурса в репозитории и контрольным

суммам.

Сервис верификации изделия обеспечивает для конечных потребителей средства для установления подлинности изделия как сертифицированного продукта, его бинарных файлов и цифровых установочных образов дистрибутивов изделия. Установление подлинности изделия проводится автоматизированно методом подсчета контрольных сумм бинарных файлов и установочных образов изделия.

Сервис верификации обеспечивает верификацию СЗИ на каждом рабочем месте пользователя. Более подробно процессы сервиса верификации описаны в пункте 4.8 «Верификация изделия» настоящего документа.

Для организации идентификации ресурсов репозитория в сети передачи данных и обеспечения потребителей сервисом онлайн-дистрибуции производитель обеспечивает регистрацию уникального домена.

Репозиторий обеспечивает предоставление потребителям сетевого доступа к бинарным исполняемым файлам сертифицированных средств защиты информации, а также репозиторий содержит образы установочных дисков СЗИ.

#### 4.8. Верификация изделия

При подключении и обновлении изделия из сторонних репозиториев, оно теряет статус сертифицированного продукта. Поэтому потребителю запрещается подключение и использование сторонних репозиториев.

Сервис верификации изделия в составе репозитория изделия обеспечивает верификацию для загружаемых образов дистрибутивных дисков комплекта поставки и проверку соответствия комплектов изделия, установленных на рабочих местах, пакетам, расположенным в репозитории изделия.

Для обеспечения верификации для загружаемых образов дистрибутивных дисков комплекта поставки производитель обеспечивает усиленную квалифицированную электронную подпись файлов образов дистрибутивных дисков.

Потребитель производит загрузку образов дистрибутивных дисков комплекта поставки изделия, размещенных в репозитории производителя. После загрузки потребитель производит верификацию загруженного дистрибутивного комплекта изделия методом проверки усиленной квалифицированной электронной подписи и проверки контрольных сумм комплекта поставки. Верификацию для загружаемых из репозитория образов дистрибутивных дисков

комплекта поставки изделия потребитель проводит самостоятельно в соответствии с инструкцией, приведенной в эксплуатационной документации на изделие.

В случае неудовлетворительного результата верификации усиленной квалифицированной электронной подписи или контрольных сумм изделия, потребителю запрещается применение загруженного дистрибутивного комплекта.

## 5. Способы распространения изделия

Изделие распространяется следующими способами:

- По договору (акту, соглашению) о поставке изделия между потребителем и производителем. В данном случае поставка может осуществляться в виде комплектов поставки;
- Сервисом цифровой дистрибуции репозитория;
- При OEM-дистрибуции оборудования (от англ. original equipment manufacturer - производитель оригинального оборудования) в рамках договора о поставке изделия между потребителем и производителем. В данном случае распространение производится путем клонирования предустановленной сертифицированной копии изделия на однотипные идентичные носители информации: накопители на жестких магнитных дисках и твердотельные накопители. После клонирования производится верификация изделия способами, описанными в разделе 4.8.

## 6. Описание штатной структуры проекта

Проект «Ред Платформа» в соответствии с приказом ООО «РЕД СОФТ» разрабатывается департаментом развития системных продуктов ООО «РЕД СОФТ» с декабря 2015 года.

С 2017 года руководство проектом осуществляет технический директор ООО «РЕД СОФТ» Кислухин Роман Александрович.

Штатная структура проекта включает:

- Отдел маркетинга;
- Отдел аналитики;
- Отдел разработки;
- Отдел тестирования;
- Отдел технической поддержки;
- Отдел документирования.

Обеспечение процессов жизненного цикла РП осуществляется командой разработчиков и технических специалистов в составе около 20 человек.