



Система управления инфраструктурой «РЕД АДМ»

Описание жизненного цикла и сопровождение продукта

46.98898398. 501140-01 96 1-1

Перечень сокращений

| | |
|----|--|
| ОО | Объект оценки |
| ОС | Операционная система |
| ПО | Программное обеспечение |
| УК | Управление конфигурацией |
| РА | Система управления инфраструктурой «РЕД АДМ» |

Оглавление

| | |
|---|----|
| 1. Введение | 4 |
| 2. Описание системы контроля версий | 5 |
| 2.1. Общее описание системы контроля версий | 5 |
| 2.2. Возможности системы контроля версий..... | 5 |
| 2.3. Модель работы..... | 6 |
| 2.4. О репозиториях | 6 |
| 2.5. Доступ к репозиторию | 7 |
| 2.6. Стадии разработки РА..... | 7 |
| 2.6.1 Пре-альфа..... | 7 |
| 2.6.2 Альфа..... | 8 |
| 2.6.3 Бета..... | 8 |
| 2.6.4 Релиз-кандидат..... | 8 |
| 2.6.5 Релиз..... | 9 |
| 2.7. Сборка программного кода РА..... | 9 |
| 2.8. Технология создания РА | 9 |
| 2.9. Элементы конфигурации РА | 10 |
| 2.10. Документация УК..... | 11 |
| 2.10.1 Требования к составу и маркировке программной документации | 11 |
| 2.10.2 Процедура формирования серийного номера продукта | 11 |
| 2.10.3 Правила присвоения серийного номера программного продукта..... | 12 |
| 3. Система отслеживания ошибок и порядок обновления РА | 14 |
| 3.1. Общее описание и назначение | 14 |
| 3.2. Техническая архитектура | 14 |
| 3.3. Общий алгоритм обновление РА | 17 |
| 3.4. Характеристики репозитория изделия..... | 17 |
| 4. Способы распространения изделия | 19 |

1. Введение

Система управления инфраструктурой «РЕД АДМ» (далее - РА) является программным продуктом, полностью разработанным компанией «РЕД СОФТ». При разработке использовались языки Python, JavaScript, фреймворк Django и библиотека React.

РЕД АДМ имеет модульную структуру и агрегирует в себе множество модулей администрирования различного назначения.

В настоящем документе представлены описания системы управления жизненным циклом РА, которая используется в ООО «РЕД СОФТ» при разработке РА, список конфигурации и метод уникальной идентификации элементов конфигурации.

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

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

2. Описание системы контроля версий

2.1. Общее описание системы контроля версий

При разработке РА для обеспечения контроля версий применяется программный продукт с открытым исходным кодом GitLab, разрабатываемый компанией GitLab Inc.

В ООО «РЕД СОФТ» при разработке РА используется система GitLab версии 14.6.2.

2.2. Возможности системы контроля версий

GitLab - веб-инструмент жизненного цикла DevOps с открытым исходным кодом, предоставляющая работу с системой контроля версий Git. Она основана на отслеживании истории изменений. Когда разработчики вносят изменения в проект, любая более ранняя версия проекта может быть восстановлена в любое время.

Разработчики могут просмотреть историю проекта, чтобы узнать:

- Какие изменения были внесены;
- Кто внес изменения;
- Когда были внесены изменения;
- Зачем нужны были изменения;

GitLab дает каждому участнику единое и последовательное представление о проекте, выявляя работу, которая уже выполняется. Просмотр прозрачной истории изменений, сведений о том, кто их внес, и того, какой вклад они вносят в разработку проекта, помогает членам команды сохранять согласованность действий при независимой работе.

В распределенной системе контроля версий каждый разработчик имеет полную копию проекта и историю проекта. В отличие от когда-то популярных централизованных систем управления версиями, распределенные не требуют постоянного подключения к центральному репозиторию.

GitLab позволяет разработчикам видеть всю хронологию своих изменений, решений и хода выполнения любого проекта в одном месте. С момента доступа к истории проекта у разработчика есть весь контекст, необходимый для того, чтобы понять его и начать вносить свой вклад.

Разработчики работают во всех часовых поясах. Совместная работа может происходить в любое время при сохранении целостности исходного кода. Используя ветки, разработчики могут безопасно предлагать изменения в производственном коде.

2.3. Модель работы

GitLab размещает репозитории Git и предоставляет разработчикам инструменты для отправки более качественного кода с помощью функций командной строки, вопросов (поточковых обсуждений), запросов на получение, проверки кода.

GitLab встраивает совместную работу непосредственно в процесс разработки. Работа организована в репозитории, где разработчики могут наметить требования или направления и установить ожидания для членов команды. Затем, используя поток GitLab, разработчики просто создают ветку для работы с обновлениями, фиксируют изменения, чтобы сохранить их, открывают запрос на получение, чтобы предлагать и обсуждать изменения, и объединяют запросы на получение, когда все находятся на одной странице.

Есть два основных способа совместной работы на GitLab:

- Общий репозиторий
- Форки репозитория

В общем репозитории отдельные лица и группы явно назначаются участниками с правами чтения, записи или администратора. Эта простая структура разрешений в сочетании с такими функциями, как защищенные ветки, помогает командам быстро развиваться при переходе на GitLab.

Для проекта с открытым исходным кодом или для проектов, в которые может внести свой вклад любой желающий, управление отдельными разрешениями может быть сложной задачей, но модель разветвления и извлечения позволяет любому, кто может просматривать проект, внести свой вклад. Форк — это копия проекта находящаяся под полным контролем создавшего его разработчика. Каждый разработчик имеет полный контроль над своим форком и может реализовать исправление или новую функцию. Работа, выполненная в разветвлениях, либо хранится отдельно, либо возвращается в исходный проект через запрос на получение. Там сопровождающие могут просмотреть предложенные изменения перед их слиянием.

Для разработки РА используется подход форков, в которых разрабатываются новые модули. После тестирования они объединяются в общий проект.

2.4. О репозиториях

Репозиторий или проект Git включает в себя всю коллекцию файлов и папок, связанных с проектом, а также историю изменений каждого файла. История файлов отображается в виде моментальных снимков во времени, называемых

фиксациями. Коммиты могут быть организованы в несколько линий разработки, называемых ветвями. Поскольку Git — это DVCS, репозитории — это автономные единицы, и любой, у кого есть копия репозитория, может получить доступ ко всей кодовой базе и ее истории. Используя командную строку или другие простые в использовании интерфейсы, репозиторий Git также позволяет: взаимодействовать с историей, клонировать репозиторий, создавать ветки, фиксировать, объединять, сравнивать изменения в разных версиях кода и многое другое.

Благодаря таким платформам, как GitLab, Git также предоставляет больше возможностей для прозрачности проектов и совместной работы. Публичные репозитории помогают командам работать вместе над созданием наилучшего конечного продукта.

2.5. Доступ к репозиторию

В ООО «РЕД СОФТ» доступ к репозиториям системы GitLab обеспечивается из контролируемых зон объектов ООО «РЕД СОФТ» удаленно с использованием протокола SSH и HTTP. Доступ разработчиков обеспечивается внутри корпоративной виртуальной частной сети (VPN), организуемой сертифицированными средствами межсетевого экранирования и построения VPN каналов.

2.6. Стадии разработки РА

В разработке программного обеспечения, стадии разработки программного обеспечения используются для описания степени готовности программного продукта. Также стадия разработки может отражать количество реализованных функций, запланированных для определённой версии программы. Стадии либо могут быть официально объявлены и регламентируются разработчиками, либо иногда этот термин используется неофициально для описания состояния продукта. Следует отметить, что стадии Beta и Alpha (Pre-Alpha) не являются показателями нестабильности релиза, так как присваиваются программе один раз или один раз за серию (серией, в данном случае, считается число до первой точки). Они могут присваиваться нескольким релизам подряд. Релизом в данном случае считается завершённая версия.

2.6.1 Пре-альфа

Начальная стадия разработки — Период времени со старта разработки до выхода стадии Альфа. Также так называются программы, не вышедшие еще в

стадию альфа или бета, но прошедшие стадию разработки, для первичной оценки функциональных возможностей в действии. В отличие от альфа и бета версий пре-альфа может включать в себя не весь спектр функциональных возможностей программы. В этом случае подразумеваются все действия, выполняемые во время проектирования и разработки программы вплоть до тестирования. К таким действиям относятся - разработка дизайна, анализ требований, собственно разработка приложения, а также отладка отдельных модулей.

2.6.2 Альфа

Внутреннее тестирование — Стадия начала тестирования программы в целом специалистами-тестерами, обычно не разработчиками программного продукта, но, как правило, внутри организации или сообществе разрабатывающих продукт. Также это может быть стадия добавления новых функциональных возможностей. Программы на данной стадии могут применяться только для ознакомления с будущими возможностями.

2.6.3 Бета

Публичное тестирование - Стадия активного бета-тестирования и отладки программы, прошедшей альфа-тестирование (если таковое было). Программы этого уровня могут быть использованы другими разработчиками программного обеспечения для испытания совместимости. Тем не менее, программы этого этапа могут содержать достаточно большое количество ошибок.

Поскольку бета-продукт не является финальной версией, и публичное тестирование производится на страх и риск пользователя, производитель не несёт никакой ответственности за ущерб, причинённый в результате использования бета-версии. Таким образом, многие производители уходят от ответственности, предоставляя пользователям только бета-версии продукта.

2.6.4 Релиз-кандидат

Релиз-кандидат или RC (англ. release candidate), Пре-релиз или Pre - стадия-кандидат на то, чтобы стать стабильной. Программы этой стадии прошли комплексное тестирование, благодаря чему были исправлены все найденные критические ошибки. Но в то же время существует вероятность выявления ещё некоторого числа ошибок, не замеченных при тестировании.

2.6.5 Релиз

Релиз или RTM (англ. release to manufacturing промышленное издание) — издание продукта, готового к тиражированию. Это стабильная версия программы, прошедшая все предыдущие стадии, в которых исправлены основные ошибки, но существует вероятность появления новых, ранее не замеченных, ошибок. RTM предшествует общей доступности (GA), когда продукт выпущен для общественности.

2.7. Сборка программного кода РА

Сборка программного кода РА производится на отдельной виртуальной ОС, размещенной на специальном выделенном сервере в контролируемой зоне производственного объекта ООО «РЕД СОФТ». Доступ к виртуальной ОС имеют только определенные лица администраторов отделения разработки, в круг обязанностей, которых входит работа со сборкой программного кода или иных задач по контролю за работой над разработкой кода РА . 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 РА и на основании решения аналитического отдела, который, в свою очередь, принимает решение на основании отчетов о проведении тестирования версии РА отделом тестирования.

Каждая сборка РА имеет уникальный идентификатор ревизии, состоящий из номера ревизии РА и номера версии сборки в ревизии.

2.8. Технология создания РА

При создании РА используются следующие средства технологического оснащения:

- персональный компьютер с архитектурой процессора x86_64;
- операционная система сервера сборки;
- koji-1.8.0 - система построения и отслеживания RPMS. Пакет содержит общие библиотеки и интерфейс командной строки;
- mock-1.1.38 - средство построения сборочного chroot;

- rpm-build-4.8.0 - пакет сценариев и исполняемых программ, которые используются для построения пакетов с помощью диспетчера пакетов RPM;
- rpmdevtools-7.5 - пакет сценариев и файлов поддержки для разработки RPM пакетов;
- комплект сборочных репозиторияев.

Методы эксплуатации средств технологического оснащения при создании РА :

- создаётся новый src.rpm пакет или модифицируется существующий с помощью пакетов rpm и rpmbuild;
- полученный src.rpm передается в koji с ключом --scratch, что обеспечивает сборку бинарного пакета без включения результата в базу данных koji;
- проверяется бинарный пакет на работоспособность и функциональность;
- повторяется сборка пакета в koji, но уже без ключа --scratch. Собранный пакет появляется в базе данных koji и его репозитории;

Приемы эксплуатации средств технологического оснащения при создании ПО:

- mock используется как низкоуровневая сборочная среда, управляемая koji.

Правила эксплуатации средств технологического оснащения при создании РА :

- необходимо соблюдение правил электро-технической безопасности при использовании сборочного персонального компьютера;
- появление сообщений об ошибках в работе mock свидетельствует о некорректной настройке средств технологического оснащения при создании РА.

2.9. Элементы конфигурации РА

Элементами конфигурации РА являются:

- исходный код РА;
- проектная документация;
- тестовая документация;
- руководство пользователя;
- документация УК.

2.10. Документация УК

2.10.1 *Требования к составу и маркировке программной документации*

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

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

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

A.B.CCCCC-DD EE FF-G

, где А – код страны разработчика (равен 46);

В – код организации-разработчика (равен 98898398);

CCCCC – регистрационный номер программного продукта по классификатору (равен 501140 – Программные средства расширения операционных систем);

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. 501140-01 34 1-1

2.10.2 *Процедура формирования серийного номера продукта*

Потребители и заказчики формируют заявки на поставку комплектов «РЕД АДМ». 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 2.10.3.

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

2.10.3 *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

RED ADM - указывается: RAD

РЕДАКЦИЯ

Open - указывается: OPN

Enterprise - указывается: ENT

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: 31 августа 2022 года — 310822.*

ПОРЯДКОВЫЙ НОМЕР

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

ПРИМЕР СЕРИЙНОГО НОМЕРА:

S3-RAD-ENT-00000018-310822-011

Серийный номер присвоен клиенту технической поддержки (расширенный уровень с дополнительными опциями) на продукт «РЕД АДМ» промышленной редакции, версия 1.8, сформирован 31 августа 2022 года, 11 заказ за день.

3. Система отслеживания ошибок и порядок обновления РА

3.1. Общее описание и назначение

При разработке РА использована система отслеживания ошибок «Ред Контроль» версии 2.0.45.1807 производства ООО «РЕД СОФТ». Система «Ред Контроль» предназначена для отслеживания одного или нескольких проектов, в которых циркулируют информационные элементы, которыми могут являться обращения клиентов, задачи департамента разработки, поручения руководителя, данные о сбоях, обнаруженных в продукте при контроле качества и т.д. Цикл обработки информационных элементов может проходить при участии различных программных комплексов в организации и персонала. «Ред Контроль» легко масштабируется и адаптируется для применения в любой отрасли.

Система «Ред Контроль» основана на СУБД «Ред База Данных», которая имеет сертификат соответствия ФСТЭК России по требованиям обеспечения информационной безопасности. Система «Ред Контроль» так же предназначена для оперативного управления бизнес-процессами и проектами, а также, для накопления фактов и их массивов, управления ими в организации.

Система «Ред Контроль» служит расширенным решением для хранения данных и обмена данными с пользователем бизнес-процессов и обеспечивает двунаправленные коммуникации, которые позволяют без усилий интегрироваться с другими решениями и источниками данных.

Ключевой сущностью в системе «Ред Контроль» являются информационные элементы, факты, которые определяют следующие группы:

- формализованные данные (поля, подструктуры, классификация)
- неформализованные данные (обсуждения, связи и т.п.);
- формальный жизненный цикл.

3.2. Техническая архитектура

Система «Ред Контроль» реализована с применением трехзвенной клиент-серверной архитектуры (Client GUI + AS + DB). Клиентская часть имеет собственную встроенную БД, синхронизируемую с сервером при наличии связи. Это позволяет работать с продуктом мобильным пользователям. Решение легко интегрируется в существующую ИТ-структуру ООО «РЕД СОФТ» благодаря серверу приложений системы «Ред Контроль», имеющему открытые интерфейсы на основе CORBA, XML и других стандартов коммуникации и обмена информацией.

Система «Ред Контроль» может быть использована как:

- система управления проектами (Issue tracking system) – предоставляет расширенную функциональность в управлении и контроле над проектами и процессами в разных бизнес-потоках;
- CRM – Система управления взаимодействием с клиентами;
- система управления и контроля над разработкой SW решений (SW development tracking and supervising tool, buqtracking);
- система управления и контроля над проектами и т.д.

Система «Ред Контроль» обеспечивает в режиме реального времени отслеживание закрепленных за ответственными лицами задач, что делает ее незаменимой для руководителей при организации эффективного труда сотрудников, а также при организации взаимодействия сотрудников между собой.

Система «Ред Контроль» позволяет работать со справочниками данных. Справочники программы, используются для исключения заполнения полей вручную. Из справочника могут быть взяты следующие данные:

- статус заявки;
- автор записи;
- тип записи;
- название подсистемы;
- приоритет;
- дата обнаружения;
- тип обращения;
- срок исполнения заявки.

Использование справочников позволяет избежать ошибок при наборе реквизитов заявки (задачи), так как в справочниках хранится достоверная информация по заполнению полей в заявке. Если справочник был изменен, то при первом же запуске программы система сама автоматически обновит этот справочник.

Общая схема жизненного цикла задачи приведена на рисунке 1.

Задача регистрируется в рамках работы структурных подразделений разработчика РА и линии технической поддержки пользователей. Это может быть задача от аналитика, связанная с доработкой по требованиям заказчика; ошибка, выявленная при тестировании РА; проблема, возникшая у пользователя или сообщение об ошибке от системы, обнаруженное пользователем в процессе эксплуатации РА. Так же задача может возникнуть в рамках работы группы разработки ПО или анализа ПО при его согласовании 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.

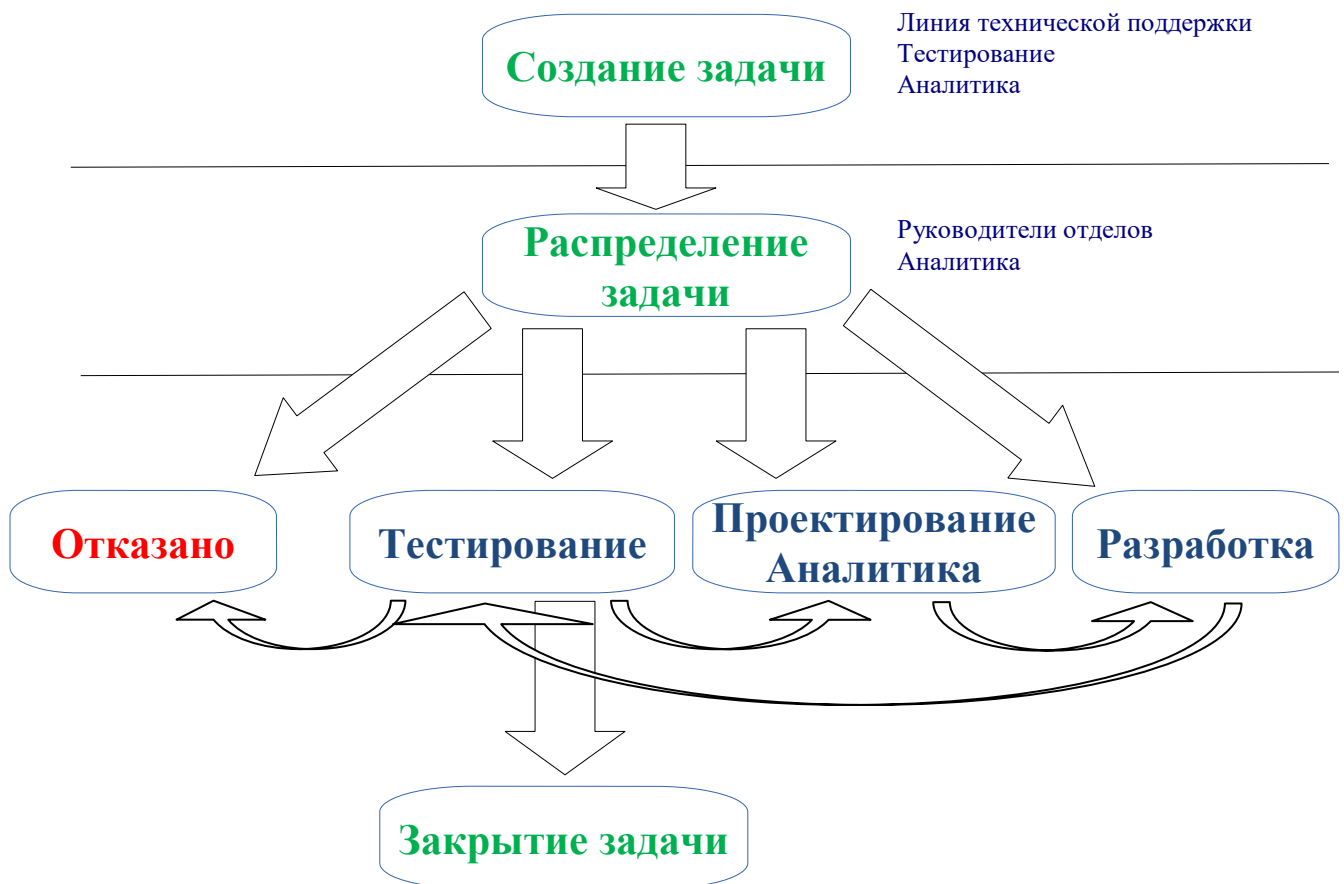


Рисунок 1 - Жизненный цикл задачи в системе отслеживания ошибок «Ред Контроль»

3.3. Общий алгоритм обновление РА

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 РА с последующей установкой обновленной версии РА в полном соответствии с программной документацией;
- Производится обновление с использованием сервисов цифровой дистрибуции и обновления репозитория изделия.

3.4. 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 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.

4. 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 копии изделия на односторонние идентичные носители информации: накопители на жестких магнитных дисках и твердотельные накопители.