



Программный продукт «РЕД ЕСАА»

Описание жизненного цикла и сопровождение продукта

Перечень сокращений

ОС	Операционная система
ПО	Программное обеспечение
УК	Управление конфигурацией

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	5
2.4. О репозиториях	6
2.5. Доступ к репозиторию	7
2.6. Стадии разработки РЕД ЕСАА	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 РЕД ЕСАА	9
2.8. Элементы конфигурации РЕД ЕСАА.....	9
2.9. Документация УК.....	10
2.9.1 Требования к составу и маркировке программной документации	10
3. Система отслеживания ошибок	11
3.1. Общее описание и назначение	11
3.2. Техническая архитектура	11
4. Способы распространения изделия	14
5. Описание штатной архитектуры проекта	15

1. Введение

Программный продукт «РЕД ЕСАА» (далее – РЕД ЕСАА) является программным продуктом, полностью разработанным компанией «РЕД СОФТ». При разработке использовался язык java.

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

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

Управление конфигурацией - один из методов или способов установить, что в созданном РЕД ЕСАА реализованы функциональные требования и спецификации. УК отвечает этим целям, предъявляя требования дисциплины и контроля в процессе уточнения и модификации РЕД ЕСАА и связанной с ним информации. Системы УК используют для обеспечения целостности частей РЕД ЕСАА, 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 РЕД ЕСАА для обеспечения контроля версий применяется программный продукт с открытым исходным кодом GitLab, разрабатываемый компанией GitLab Inc.

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

2.6. Стадии разработки РЕД ЕСАА

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 РЕД ЕСАА

Сборка программного кода РЕД ЕСАА 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 РЕД ЕСАА. 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 РЕД ЕСАА и на основании решения аналитического отдела, который, в свою очередь, принимает решение на основании отчетов о проведении тестирования версии РЕД ЕСАА отделом тестирования.

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

2.8. Элементы конфигурации РЕД ЕСАА

Элементами конфигурации РЕД ЕСАА являются:

- исходный код РЕД ЕСАА;
- проектная документация;
- тестовая документация;
- руководство пользователя;
- документация УК.

2.9. Документация УК

2.9.1 Требования к составу и маркировке программной документации

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

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

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

АВ.00000000.00000-00 11-2

где АВ.00000000.00000-00 – номер разрабатываемой системы или программного продукта

11 – код документа

2 – версия документа.

3. Система отслеживания ошибок

3.1. Общее описание и назначение

При разработке РЕД ЕСАА использована система отслеживания ошибок «Redmine» версии 2.0.45.1807 производства ООО «РЕД СОФТ». Система «Redmine» 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. «Redmine» легко масштабируется и адаптируется для применения в любой отрасли.

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

Система «Redmine» 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 «Redmine» 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 «Redmine» 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 «Redmine» 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 РЕД ЕСАА и линии технической поддержки пользователей. Это может быть задача от аналитика, связанная с доработкой по требованиям заказчика; ошибка, выявленная при тестировании РЕД ЕСАА; проблема, возникшая у пользователя или сообщение об ошибке от системы, обнаруженное пользователем в процессе эксплуатации РЕД ЕСАА. 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.

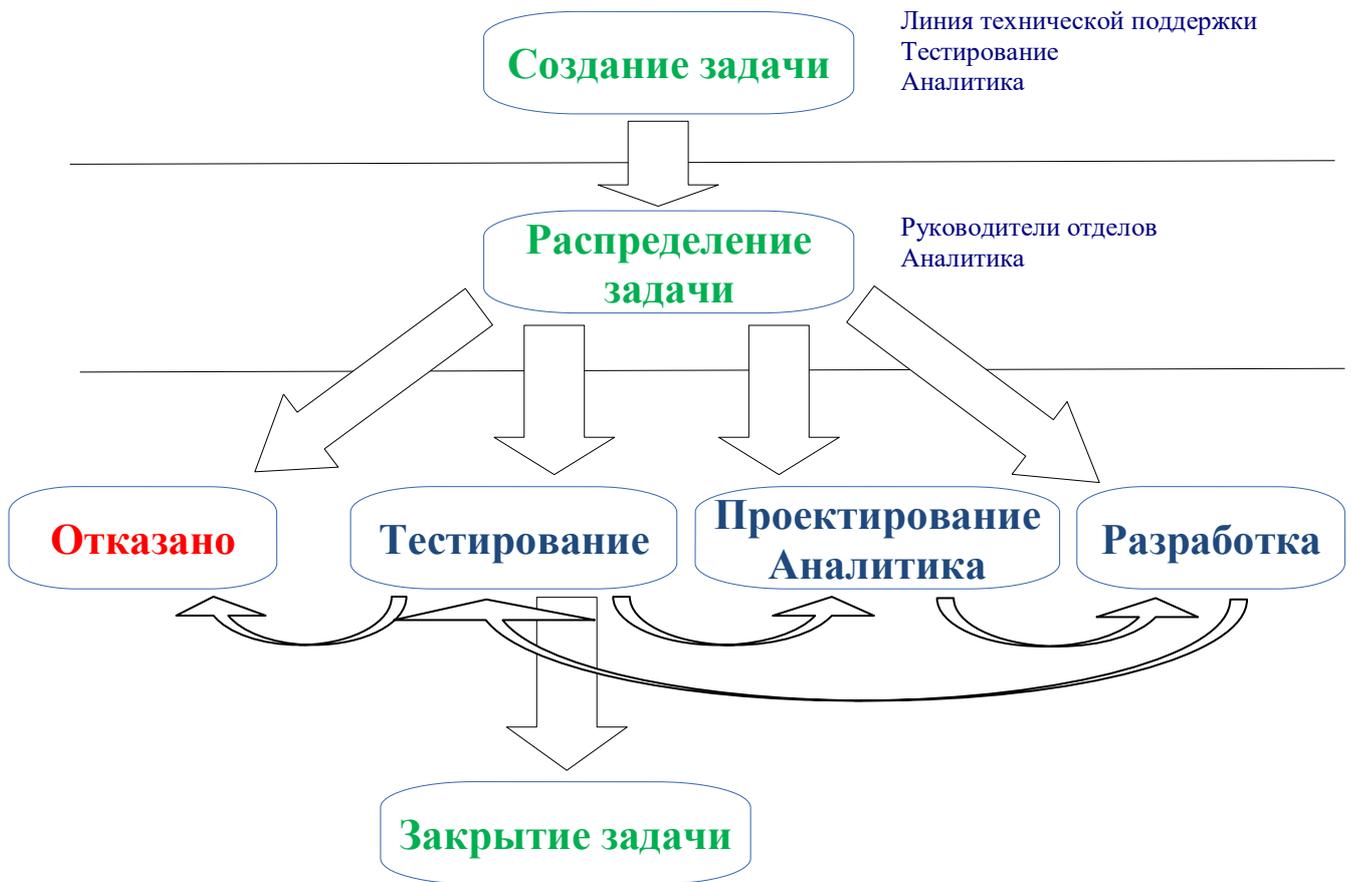


Рисунок 1 - Жизненный цикл задачи в системе отслеживания ошибок «Redmine»

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.

5. Описание штатной архитектуры проекта

Руководство проектом осуществляет директор проектного департамента ООО «РЕДСОФТ», Скорняков Василий Александрович.

Штатная структура проекта включает:

- Отдел управления проектами;
- Отдел аналитики;
- Отдел разработки;
- Отдел тестирования;
- Отдел технической поддержки;
- Отдел внедрения и эксплуатации систем.