

**Программный комплекс
«РЕДШЛЮЗ»**

**Управление конфигурацией
Код документа:**

Версия 1.0

**МОСКВА
2017**

ТЕРМИНЫ И СОКРАЩЕНИЯ

ОС	Операционная система
ПО	Программное обеспечение
УК	Управление конфигурацией

Содержание

ТЕРМИНЫ И СОКРАЩЕНИЯ	2
Введение	4
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	6
1.5 Ветвление.....	7
1.6 Сборка программного кода «РЕДШЛЮЗ»	8
1.7 Маркировка документации	8
2 Система отслеживания ошибок	9
Заключение.....	11

Введение

В настоящем документе представлены описания системы управления конфигурацией, которая используется в ООО «Ред Софт» при разработке «РЕДШЛЮЗ», список конфигурации и метод уникальной идентификации элементов конфигурации.

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

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

1 Описание системы контроля версий

При разработке «РЕДШЛЮЗ» 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.

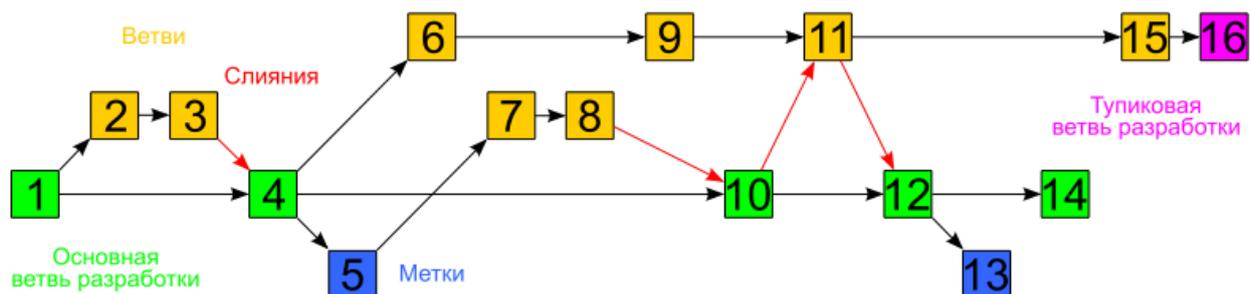


Рисунок 1 - Эволюция ветвей в хранилище

1.6 Сборка программного кода «РЕДШЛЮЗ»

Сборка программного кода «РЕДШЛЮЗ» 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 «РЕДШЛЮЗ». 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 «РЕДШЛЮЗ».

Каждая сборка «РЕДШЛЮЗ» имеет уникальный идентификатор, состоящий из номера версии ПО и номера сборки.

1.7 Маркировка документации

Вся документация, создаваемая и сопровождаемая в рамках разработки «РЕДШЛЮЗ», имеет уникальную маркировку, имеющую следующий формат:

АВ.00000000.00000-00 11-2

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

11 – код документа

2 – версия документа.

Если документ создается вновь, ему присваивается стандартизированный номер системы и код документа. Таким образом, обеспечивается уникальность номеров документов и универсальность в их идентификации, обеспечивающая удобство поиска и работы с документами.

Если документ уже существует и в него только вносятся изменения, документу присваивается новая версия. Изменения фиксируются в листе изменений.

Все зарегистрированные документы размещаются в репозитории продукта, откуда могут быть получены пользователями. Таким образом, достигается доверие, что документация может быть модифицирована только под контролем разработчика и в соответствии с необходимыми полномочиями.

2 Система отслеживания ошибок

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

Программа «АСУП» обеспечивает контроль исполнения любых производственных задач, заявок, документов организационно-распорядительного характера на стадии разработки и исполнения.

Программа 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.

Программа «АСУП» 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, 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 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 «РЕДШЛЮЗ». 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.

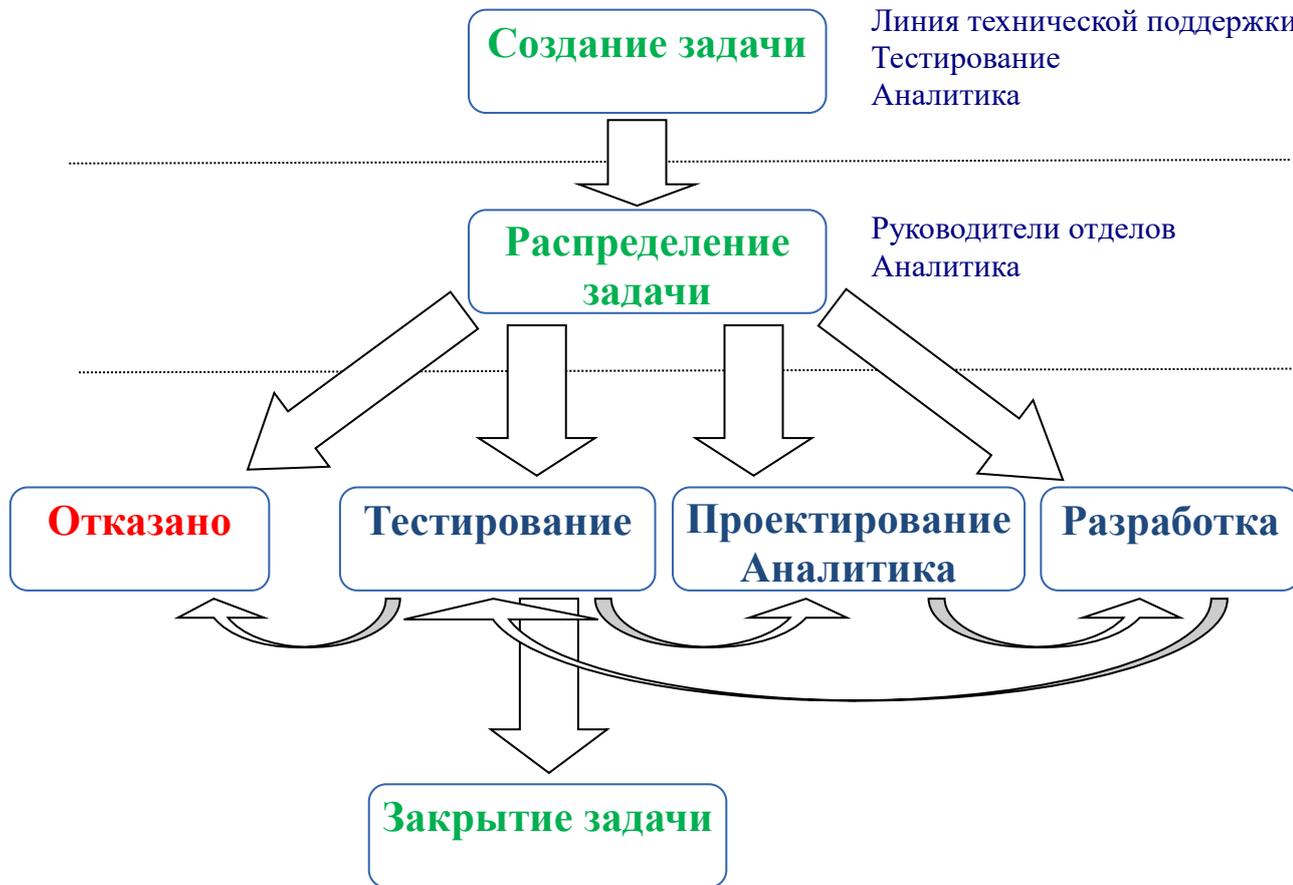


Рисунок 2 - Жизненный цикл задачи в системе отслеживания ошибок «АСУП»

Заключение

Использование системы управления версиями помогает предотвратить случайные или несанкционированные модификации элементов конфигурации. Система УК обеспечивает целостность «РЕДШЛЮЗ», начиная с ранних этапов проектирования и на протяжении всей последующей деятельности по сопровождению.

Система управления версиями:

а) обеспечение корректности и полноту «РЕДШЛЮЗ» к моменту представления его потребителю;

б) обеспечение, чтобы никакие элементы конфигурации не были пропущены в процессе оценки;

в) предотвращение несанкционированной модификации, добавления или удаления элементов конфигурации «РЕДШЛЮЗ».