

Using version control systems in software development

S.S.Kasimov, O.N.Djuraev
Tashkent University of Information Technologies
108, str. A.Temur, Tashkent, Uzbekistan

Использование систем управления версиями при разработке программных продуктов

С.С.Касымов, О.Н.Джураев
Ташкентский университет информационных технологий
108, ул. А.Темур, г. Ташкент, Узбекистан

Аннотация – В статье рассмотрены системы управления версиями (СУВ) при разработке программных продуктов. Анализируются возможности и основные концепции СУВ.

Abstract – In the article version control systems (VCS) in software development are considered. Opportunity and main conceptions of VCS are analyzed.

Ключевые слова: Программное обеспечение, Version Control System, Revision Control System, Concurrent Versions System, Subversion, Rational ClearCase, Microsoft Visual SourceSafe, Perforce, Team Foundation Server, Darcs, Git, Mercurial.

Keywords: Software, Version Control System, Revision Control System, Concurrent Versions System, Subversion, Rational ClearCase, Microsoft Visual SourceSafe, Perforce, Team Foundation Server, Darcs, Git, Mercurial.

Системы управления версиями - класс программных продуктов, нацеленных на решение ряда задач, с которыми повседневно сталкивается каждый программист. С помощью СУВ вы следите за изменениями кода вашего программного продукта в ходе его разработки, и можете управлять различными его состояниями: новая версия, работа над которой идет прямо сейчас; старая версия, которую придется поддерживать еще некоторое время; или же старая версия, интересная только историкам.

СУВ (*Version Control System, VCS* или *Revision Control System*) — программное обеспечение для облегчения работы с изменяющейся информацией. СУВ позволяет хранить несколько версий одного и того же документа, при необходимости, возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение и многое другое.

Такие системы наиболее широко применяются при разработке программного обеспечения, для хранения исходных кодов разрабатываемой программы. Однако, они могут с успехом применяться и в других

областях, в которых ведётся работа с большим количеством непрерывно изменяющихся электронных документов, в частности, они всё чаще применяются в САПР. Управление версиями используется в инструментах конфигурационного управления (*Software Configuration Management Tools*).

Программисты, чьи исходники контролируются СУВ, чем-то неуволимо отличаются от остальных программистов. Они в каждый момент рабочего дня точно знают, что именно было сделано за день, а после исправления ошибки могут точно сказать, в каком именно месте кода была ошибка. Они не подвержены синдрому "работает - не трогай", потому что могут совершенно безболезненно ударяться в самые сложные эксперименты со своей программой. Они твердо знают, что в любой момент могут вернуться к "исходникам, которые работали", сколько бы экспериментов с новым кодом не было проведено. Более того, если пользователь вдруг захочет небольшое, крошечное изменение, когда программа находится в многообещающем, но совершенно нерабочем состоянии, то все, что для этого потребуется - переключиться на стабильную ветку, исправить там, что надо и отдать пользователю, затем переключиться обратно на ствол разработки.

Рассмотрим некоторые СУВ при разработке программных продуктов.

1. RCS (*Revision Control System*) является одной из самых первых СУВ. Для каждого файла, зарегистрированного в системе, она хранит полную историю изменений, причём для текстовых файлов используется эффективный алгоритм дельта-компрессии, когда хранится только последняя версия и все межверсионные изменения. Система позволяет также хранить версии бинарных файлов, но без использования этого механизма, то есть каждая версия бинарного файла хранится полностью.

RCS разработана в 1980-х годах Уолтером Тиши из Университета Пердью как свободная и более развитая

альтернатива популярной тогда Source Code Control System. В настоящее время, является частью Проекта GNU, однако всё ещё поддерживается Университетом Пердью.

RCS не имеет средств для коллективной работы над набором файлов - эти средства появились в системе-наследнице — CVS, использующей форматы и алгоритмы RCS для учёта версий, но имеющей также интерфейсы для коллективной работы.

Отсутствие коллективной работы на практике выглядит так, что только тот пользователь, который произвел действие «Lock» над файлом или файлами, может вносить изменения. Другие пользователи запросить эти же файлы на редактирование не могут, пока первый пользователь не закончит работу с ними [1].

2. CVS (Concurrent Versions System, «Система Одновременных Версий») — программный продукт, относящийся к разряду СУВ. Хранит историю изменений определённого набора файлов, как правило, исходного кода программного обеспечения, и облегчает совместную работу группы людей над одним проектом. CVS популярна в мире открытого программного обеспечения. Система распространяется на условиях лицензии GNU GPL.

В настоящее время активная разработка системы прекращена, в исходный код вносятся только небольшие исправления.

Некоторые источники считают CVS устаревшей системой, потому что CVS имеет ряд недостатков, и имеются более молодые альтернативные СУВ, свободные от большинства недостатков CVS.

CVS использует архитектуру клиент-сервер. Обычно клиент и сервер соединяются через локальную сеть или через Интернет, но могут работать и на одной машине, если необходимо вести историю версий локального проекта. Серверное ПО обычно работает под управлением Unix, тогда как CVS клиенты доступны во всех популярных операционных системах.

Сервер хранит в специальном хранилище текущую версию проекта и историю изменений, а клиент соединяется с ним, чтобы получить нужную ему версию или записать новую. Получив с сервера нужную версию, клиент создаёт локальную копию проекта - так называемую рабочую копию. После того как в файлы, находящиеся в рабочей копии, внесены необходимые изменения, они пересылаются на сервер [2].

3. *Subversion* — свободная централизованная СУВ, созданная в 2000 г. компанией CollabNet Inc.

Название Subversion является игрой слов: его можно перевести и как «подверсия», и как «свержение». Subversion обладает всеми основными функциями CVS и свободна от ряда её недостатков.

Subversion используется многими сообществами разработчиков открытого программного обеспечения. В их числе такие известные проекты, как Apache, KDE, GCC, Free Pascal, Python, Ruby, Mono, FreeBSD, Haiku OS. Хостинг Subversion для проектов с открытым кодом предоставляют SourceForge.net и Tigris.org. Subversion используется в системах Google Code и BountySource.

Также Subversion широко используется в корпоративной сфере.

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 СУВ. 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, 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.

При использовании доступа с помощью WebDAV также поддерживается прозрачное управление версиями — если любой клиент WebDAV открывает для записи и затем сохраняет файл, хранящийся на сетевом ресурсе, то автоматически создаётся новая версия [3].

4. Rational ClearCase - СУВ разрабатываемая подразделением Rational Software компании IBM.

ClearCase была разработана компанией Atria Software и выпущена в 1992 году сначала для Unix, а позже и для Windows. Некоторые сотрудники Atria принимали участие в работе над более ранней системой DSEE (Domain Software Engineering Environment) в компании Apollo Computer. Позднее в результате слияния Atria с Pure Software образовалась компания PureAtria, которая затем объединилась с Rational Software, которая в свою очередь была куплена IBM. IBM продолжила разработку и продвижение на рынке системы ClearCase, которая широко используется компаниями, производящими программное обеспечение.

В DSEE были предложены многие концепции используемые сейчас в ClearCase. Файловая система Apollo Domain позволяла специальной управляющей программе вмешиваться в процесс доступа к файлам и DSEE стала использовать эту возможность для незаметного создания копии текущей версии каждого отдельного открытого файла [4].

5. Microsoft Visual SourceSafe (Visual SourceSafe, VSS) — программный продукт компании Майкрософт, файл-серверная СУВ, предназначенная для небольших команд разработчиков. VSS позволяет хранить в общем хранилище файлы, разделяемые несколькими пользователями, для каждого файла хранится история версий.

VSS входит в состав пакета Microsoft Visual Studio и интегрирован с продуктами этого пакета. Доступен только для платформы Windows. Версию для Unix поддерживает компания MainSoft.

В ноябре 2005 года вышла обновлённая версия продукта — Visual SourceSafe 2005, обещающая повышенную стабильность и производительность, улучшенный механизм слияния для XML-файлов и файлов в Юникоде, а также работу через HTTP.

Visual SourceSafe нацелен на индивидуальных разработчиков либо небольшие команды разработчиков. Там где VSS недостаточно, ему на замену предлагается новый продукт Майкрософт — Team Foundation Server, входящий в состав Visual Studio Team System [5].

6. Perforce - коммерческая СУБД. Разработана компанией Perforce Software, основанной в 1995 году. Система имеет клиент-серверную архитектуру. Сервер Perforce может одновременно иметь несколько репозиторий, называемые «депо».

Сервер Perforce может быть установлен на операционные системы Unix, Mac OS X, Microsoft Windows.

Клиент предоставляет графический интерфейс и широкий набор утилит для работы из командной строки. Клиентская часть реализована для широкого набора операционных систем. Также разработан большой набор плагинов, позволяющих интегрироваться с широким кругом сред разработки программного обеспечения и приложений других разработчиков: IntelliJ IDEA, XCode, Autodesk 3D Studio Max, Maya, Adobe Photoshop, Microsoft Office, Eclipse, emacs. Помимо этого система предоставляет множество других возможностей — различного вида извещения, создание и обслуживание ветвей проекта, с мощной системой слияний веток, точки отката в базе данных и взаимодействие с системами отслеживания ошибок.

В настоящее время насчитывается более 320 000 пользователей Perforce в 5 000 компаниях [6].

7. Team Foundation Server - продукт корпорации Microsoft, представляющий собой комплексное решение, объединяющее в себе СУБД, сбор данных, построение отчетов, отслеживание статусов и изменений по проекту, и предназначенное для совместной работы над проектами по разработке программного обеспечения. Данный продукт доступен как в виде отдельного приложения, так и в виде серверной платформы для Visual Studio Team System (VSTS).

Team Foundation Server работает по трёхуровневой архитектуре: клиентский уровень, прикладной уровень и уровень данных. Клиентский уровень используется для создания и управления проектами, а также для доступа к хранимым и управляемым элементам проекта. На этом уровне TFS не содержит никаких пользовательских интерфейсов, как и веб-сервисов, которые могут быть использованы клиентскими приложениями для самостоятельной интеграции в функциональность TFS. Эти веб-сервисы используются такими приложениями, как Visual Studio Team System для применения TFS в качестве серверной инфраструктуры хранилища информации или выделенного TFS управления приложениями, наподобие включенного приложения Team Foundation Client. Сами

веб-сервисы находятся на прикладном уровне. Прикладной уровень также включает в себя веб-портал и репозиторий документации, поддерживаемые Windows SharePoint Services. Веб-портал, называемый Team Project Portal, выступает в роли центра взаимодействия для проектов, управляемых TFS. Репозиторий документов используется как для элементов проекта, так и для отслеживания ревизий, а также для накопления и обработки данных и генерации отчетов. Уровень данных, основывающийся в первую очередь на установленном SQL Server 2005 Standard Edition, обеспечивает сервисы постоянного хранения данных для репозитория документов. Уровень данных и уровень приложений могут существовать на различных физических или виртуальных серверах при использовании Windows Server 2003 или более специализированных версий. Уровень данных не взаимодействует с клиентским уровнем напрямую, только через прикладной уровень.

Team Foundation Server реализует репозиторий управления исходным кодом, называемый Team Foundation Version Control (TFVC). В отличие от предыдущего решения Microsoft по контролю кода, Visual SourceSafe (VSS), которое основывалось на механизме файлового хранения, Team Foundation хранит весь код, равно как и запись обо всех изменениях кода в базе данных под управлением SQL Server. Поддерживаются такие особенности, как например, одновременная множественная блокировка кода для изменений, разрешение конфликтов, откладывание внесения изменений, ветвление и слияние, а также возможность устанавливать уровни доступа на любом уровне дерева исходного кода, наряду с наиболее очевидными возможностями отслеживания версий документации, блокировок, откатов и операций подтверждения микроизменений. Механизм контроля исходного кода напрямую связан с рабочими элементами Team System; При подтверждении изменений разработчик может определять взаимосвязь его кода с одним или более определенными рабочими элементами для указания какие именно проблемы решает данное подтверждение. Администраторы TFS могут принудительно указывать политики подтверждений, которые предусматривают удовлетворение требованиям Code Analysis, а кроме того можно поставить обязательное указывание рабочих элементов, связанных с данным подтверждением, или обновление статуса связанных рабочих элементов. Отдельные версии файлов могут отмечать специальными метками, а все файлы с одинаковыми метками образуют релиз-группу. В отличие от VSS, репозиторий контроля кода TFS не обладает ни поддержкой привязки к элементу из нескольких мест структуры каталогов исходного кода, ни поддерживает «закрепления» элемента [7].

8. Darcs - распределённая СУБД с широкими возможностями, может быть использована для замены CVS.

Darcs написана на языке Haskell, и может использоваться в GNU/Linux, Mac OS X, FreeBSD,

NetBSD, OpenBSD и Microsoft Windows. Darcs включает CGI-скрипт для просмотра репозитория через web.

В противоположность CVS и Subversion, но подобно Arch и Monotone, Darcs является «распределённой» СУБ. Репозитории - это не синхронизированные друг с другом «острова», а система патчей Darcs управляет потоками изменений между ними. В этом смысле Darcs принадлежит тому же поколению СУБ, что и Mercurial, Bazaar, Git.

Поскольку в Darcs записываются и хранятся патчи, и есть алгебра их вычислений, то легко реализуются такие интересные возможности как спонтанные автоматические ветви, перестановка патчей и ряд других [8].

9. Git - распределённая СУБ файлов. Проект был создан Линусом Торвалдсом для управления разработкой ядра Linux. На сегодняшний день поддерживается Джунио Хамано.

Примерами проектов, использующих Git, являются Linux kernel, Cairo, GNU Core Utilities, Mesa, Wine, Chromium, Compiz Fusion и некоторые дистрибутивы GNU/Linux.

Программа является свободной и выпущена под лицензией GNU GPL версии 2.

Система спроектирована как набор программ, специально разработанных с учётом их использования в скриптах. Это позволяет удобно создавать специализированные системы контроля версий на базе Git или пользовательские интерфейсы. Например, Cogito является именно таким примером фронтенда к репозиториям Git, а StGit использует Git для управления коллекцией патчей.

Git поддерживает быстрое разделение и слияние версий, включает инструменты для визуализации и навигации по нелинейной истории разработки. Как и Darcs, BitKeeper, Mercurial, SVK, Bazaar и Monotone, Git предоставляет каждому разработчику локальную копию всей истории разработки, изменения копируются из одного репозитория в другой.

Удалённый доступ к репозиториям Git обеспечивается git-daemon, SSH- или HTTP-сервером. TCP-сервис git-daemon входит в дистрибутив Git и является наряду с SSH наиболее распространённым и надёжным методом доступа. Метод доступа по HTTP, несмотря на ряд ограничений, очень популярен в контролируемых сетях, потому что позволяет использование существующих конфигураций сетевых фильтров [9].

10. Mercurial - кроссплатформенная распределённая СУБ, разработанная для эффективной работы с очень большими репозиториями кода. Mercurial первоначально был написан для Linux, позже портирован под Windows, Mac OS X и большинство Unix-систем. В первую очередь он является консольной программой. Все его операции запускаются параметрами программы hg, название которой взято от обозначения химического знака ртути.

Система Mercurial написана на Python, хотя чувствительные к производительности части выполнены в качестве Python-расширений на C. Репозитории Mercurial управляются при помощи утилиты командной строки hg.

Наряду с традиционными возможностями систем контроля версий, Mercurial поддерживает полностью децентрализованную работу, ветвление, слияние репозитория. Поддерживается обмен данными между репозиториями через HTTP/HTTPS, SSH и вручную при помощи упакованных наборов изменений.

Mercurial использует SHA1-хеши для идентификации ревизий и позволяет присваивать отдельным ревизиям индивидуальные метки.

Утилита hg обладает компактным интерфейсом, и Mercurial считается более простой в освоении системой, чем, например, git [10].

11. Bazaar - распределённая СУБ, разработка которой спонсируется фирмой Canonical Ltd.. Система Bazaar разработана с целью облегчить работу над развитием свободных и открытых проектов для всех желающих.

Команда разработчиков фокусируется на том, чтобы сделать систему лёгкой в использовании, но при этом точной в деталях и очень гибкой, подстраиваемой под конкретные нужды пользователей. Также большое внимание уделяется вопросам работы со множеством веток и их последующим объединением. Bazaar может использоваться как одним разработчиком для работы над множеством локальных веток, так и группой разработчиков, совместно работающих в сети.

Система контроля версий Bazaar написана на языке программирования Python. Существуют установочные пакеты для основных дистрибутивов GNU/Linux, инсталляторы для Mac OS X и MS Windows. Bazaar — это свободное программное обеспечение, в настоящее время является частью проекта GNU.

Команды Bazaar очень похожи на команды, используемые в CVS или SVN. Для создания и поддержки нового проекта без использования специального сервера с репозиторием пользователю достаточно запустить команду `bzr init` в том каталоге, который нужно поместить под контроль версий.

В отличие от чисто распределённых систем контроля версий, которые не используют центральный сервер, Bazaar поддерживает работу как с сервером так и без него. Возможно даже использовать оба метода одновременно для одного и того же проекта. Сайты Launchpad и Sourceforge предлагают свободный хостинг для проектов, которые используют Bazaar.

Bazaar поддерживает работу напрямую с некоторыми другими системами контроля версий. Пользователи могут создавать новые ветки на основе репозитория других систем, делать локальные изменения и фиксировать их в Bazaar-ветке, и затем отправлять свои изменения назад в оригинальный репозиторий. Bazaar поддерживает базовые операции с Subversion, а также с Git. Также начата работа над поддержкой Mercurial. Плагин `bzr-hg` умеет пока немного, однако его функций достаточно, чтобы отобразить историю ревизий в графическом виде.

Bazaar поддерживает полный набор символов Unicode в именах файлов. Система также позволяет использовать

Unicode для составления комментариев к ревизиям, в именах авторов изменений и т. д. [11].

В заключение можно сказать что эта статья - поверхностный взгляд на СУВ в использовании на сегодняшний момент. СУВ редко используются в изоляции, и, таким образом, обычно выбираются командами больше, чем в индивидуальном порядке. Поэтому, играйте с возможностями и развивайтесь в разных направлениях. СУВ необычайно широко применяются при разработке подавляющего большинства современных проектов.

ЛИТЕРАТУРА

- [1] <http://www.cs.purdue.edu/homes/trinkle/RCS>
- [2] <http://www.nongnu.org/cvs>
- [3] <http://subversion.tigris.org>
- [4] <http://www-306.ibm.com/software/awdtools/clearcase>
- [5] <http://msdn.microsoft.com/en-us/library>
- [6] <http://www.perforce.com>
- [7] <http://msdn2.microsoft.com/en-us/library/ms364061.aspx>
- [8] <http://www.darcs.net>
- [9] <http://git-scm.com>
- [10] <http://hgbook.red-bean.com/read>
- [11] <http://bazaar-vcs.org>