

Системы управления версиями.

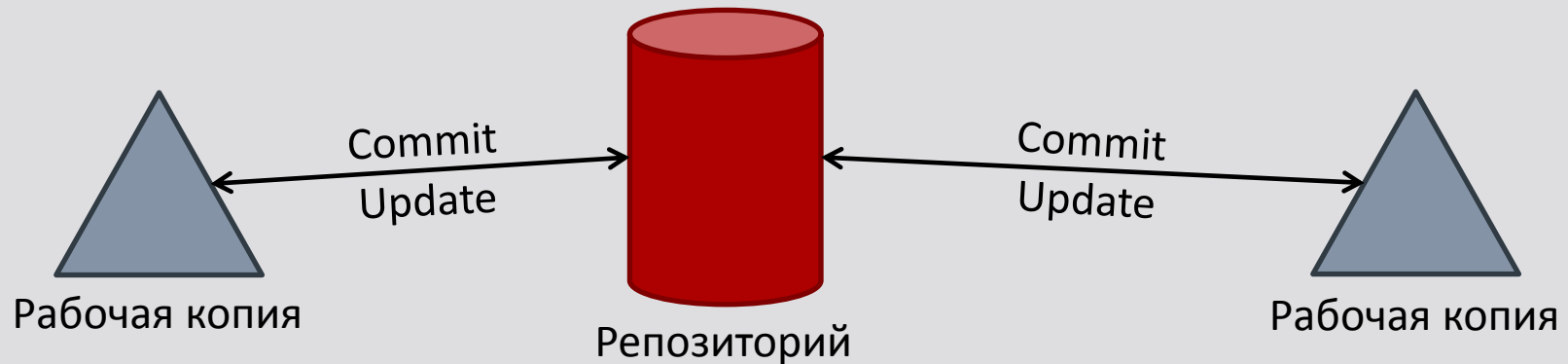
Назначение

- Организация одновременной работы нескольких разработчиков
- Поддержка нескольких направлений разработки и обеспечение их взаимодействия
- Управление версиями
- Отслеживание изменений

История

Поколение	Сетевая модель	Примеры
Первое	Нет	RCS, SCSS
Второе	Централизованная	CVS, SVN
Третье	Распределенная	Git, Mercurial

Централизованные VCS

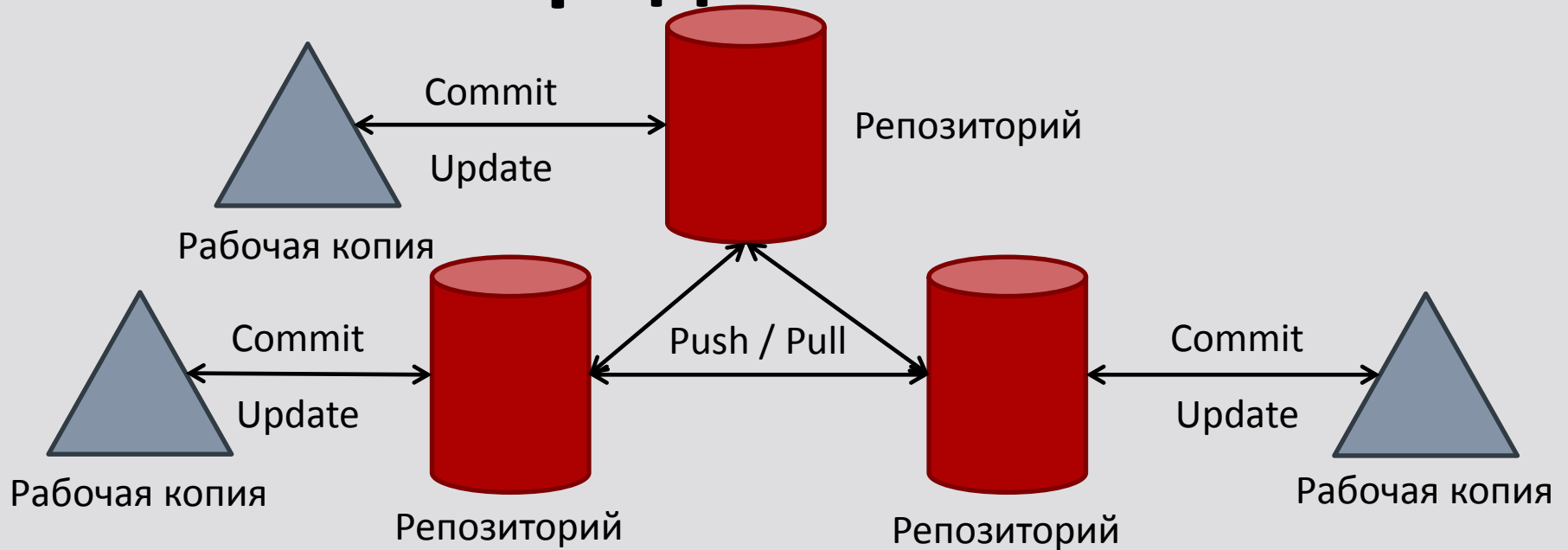


Клиент-серверная модель: один центральный репозиторий, с которым разработчики взаимодействуют по сети.

Централизованные VCS: Примеры

- CVS- одна из первых систем второго поколения (1986г.). Обладает множеством недостатков и считается устаревшей.
- Subversion (SVN) – система второго поколения, созданная для замены CVS. Одна из самых распространенных систем контроля версий.

Распределенные VCS



В отличие от централизованной модели, может существовать несколько экземпляров репозитория, которые время от времени синхронизируются между собой.

Распределенные VCS: примеры

- Git- распределенная система управления версиями, созданная Л. Торвальдсом для управления разработкой ядра Linux.
- Mercurial- другая распределенная VCS. Создана в 2005 году М. Макалом с практически одновременно с началом разработке git'а и с аналогичными целями.

Распределенные VCS: достоинства

- Наличие «личной» копии репозитория
- Скорость
- Возможность автономной работы
- Более гибкий рабочий процесс
- Более простое слияние
- Неявное резервное копирование
- Распределение нагрузки на сервер

Распределенные VCS: недостатки

- Отсутствие блокировок
- Неудобства при работе с большими репозиториями
- Проблемы с интеграцией с некоторыми другими инструментами.
- Сложно *полностью* удалить что-то из репозитория
- Сложное администрирование
- Отсутствие контроля доступа
- Сложнее в использовании и понимании, чем централизованные системы.

Основные операции: создание репозитория

Subversion	Git	Mercurial
svnadmin create	hg init	git init

Репозиторий- место, где система управления версиями хранит каталоги и файлы, историю их изменений и некоторые служебные данные. В дальнейшем содержимое репозитория изменяется не напрямую, а опосредованно через *рабочие копии*.

Основные операции: получение рабочей копии

Subversion	Git	Mercurial
svn checkout	-	-

Для работы с содержимым репозитория каждый разработчик имеет собственную *рабочую копию*.
Рабочая копия- «снимок» содержимого репозитория, плюс некоторая служебная информация.

Основные операции: фиксация изменений

Subversion	Git	Mercurial
svn commit	git commit	hg commit

Изменив содержимое рабочей копии, разработчик *фиксирует* сделанные изменения в репозитории. Как правило, фиксация сопровождается небольшим текстовым комментарием, описывающим сделанные изменения.

Основные операции: обновление рабочей копии

Subversion	Git	Mercurial
svn update	git checkout / git pull	hg update

Для применения последних изменений из репозитория к своей рабочей копии, разработчик *обновляет* её.

Основные операции: добавление файла

Subversion	Git	Mercurial
svn add	git add	hg add

Для помещения некоторого файла под управление системой контроля версий, его требуется *добавить*. Добавленный файл будет внесен в репозиторий при следующей фиксации.

Основные операции: изменение файла

Subversion	Git	Mercurial
-	git add	-

Как правило изменение файла не требует взаимодействия с VCS. Изменения файлов автоматически отслеживаются и становятся частью следующего коммита, однако, некоторые системы требуют явных указаний на этот счёт.

Основные операции: удаление файла

Subversion	Git	Mercurial
svn remove	git rm	hg remove

Если некоторый файл больше не нужен, то он может быть *удален* из репозитория. Файл будет сразу же удален из рабочей копии. Из репозитория файл будет удален во время следующего коммита. Следует учесть, что из репозитория файл не удаляется полностью: он остается доступен как часть предыдущих версий содержимого.

Основные операции: переименование файла

Subversion	Git	Mercurial
svn move	git mv	hg rename

Файл может быть *переименован*. При этом, система контроля версий сохраняет историю его предыдущих изменений.

Основные операции: перемещение файла

Subversion	Git	Mercurial
svn move	git mv	hg rename

Некоторые VCS не разделяют *перемещение* файла и его переименование, в то время как другие считают имя файла и его каталог отдельными атрибутами.

Основные операции: получение состояния

Subversion	Git	Mercurial
svn status	git status	hg status

Разработчик может получить информацию о текущем состоянии рабочей копии: список измененных файлов, список конфликтов, изменения, которые войдут в следующий коммит и так далее.

Основные операции: получение изменений

Subversion	Git	Mercurial
svn diff	git diff	hg diff

Также можно получить сами изменения, а не только их список.

Основные операции: отмена изменений

Subversion	Git	Mercurial
svn revert	git checkout	hg revert

Изменения, внесенные в рабочую копию, можно *отменить*. В этом случае рабочая копия будет приведена в состояние, соответствующее последнему коммиту в репозитории.

Основные операции: получение истории изменений

Subversion	Git	Mercurial
svn log	git log	hg log

Разработчик может запросить истории сделанных изменений, которая включает в себя собственно список изменений и некоторую дополнительную информацию, такую как: имя того, кто внес изменение, время внесения и комментарий.

Основные операции: создание метки

Subversion	Git	Mercurial
svn copy	git tag	hg tag

Метки используются для присвоения некоторой версии содержимого репозитория осмысленного имени.

Основные операции: создание ветви

Subversion	Git	Mercurial
svn copy	git branch	hg branch

Ветвь – независимое направление разработки. Изменения, вносимые в одну ветвь, не влияют на остальные ветви. Ветви используются, например, для внесения изменений, которые могут дестабилизировать код или для поддержки релизов. Изменения могут быть перенесены между ветвями посредством *слияния*.

Основные операции: слияние

Subversion	Git	Mercurial
svn merge	git merge	hg merge

Слияние – объединение независимых изменений в одну версию документа. Слияние может происходить при обновлении рабочей копии, коммите, затрагивающем измененные другим пользователем файлы, или при переносе изменений из одной ветви в другую. Если VCS не может автоматически объединить изменения, то говорят о наличии *конфликта*.

Основные операции: разрешение конфликта

Subversion	Git	Mercurial
svn resolve	-	hg resolve

В некоторых случаях слияние изменений не может быть сделано автоматически. В этом случае VCS извещает об этом пользователя и предлагает ему самому разрешить возникший *конфликт*. Если конфликт касается содержимого файла, то система управления версиями может вызвать внешнюю программу, позволяющую выполнить слияние конфликтных участков вручную.

Основные операции: блокировка

Subversion	Git	Mercurial
svn lock	-	-

В некоторых случаях требуется получить эксклюзивное право на работу с файлом. Это может потребоваться, например, при работе с бинарными файлами, либо когда изменения значительно изменяют содержимое файла.

Основные операции: клонирование репозитория

Subversion	Git	Mercurial
-	git clone	hg clone

Для создания нового экземпляра репозитория в распределенных VCS требуется *клонировать* существующий репозиторий. Экземпляры репозитория синхронизируются между собой, посылая друг другу свои изменения и получая изменения из других экземпляров.

Основные операции: отправка изменений в удаленный репозиторий

Subversion	Git	Mercurial
-	git push	hg push

Изменения, сделанные в одном экземпляре репозитория, могут быть отправлены в другой экземпляр. Экземпляры при этом не обязательно становятся одинаковыми.

Основные операции: получение изменений из удаленного репозитория

Subversion	Git	Mercurial
-	git pull / git fetch	hg pull

Один экземпляр репозитория может получить изменения, сделанные в другом экземпляре. Экземпляры при этом не обязательно становятся одинаковыми.

Графические интерфейсы

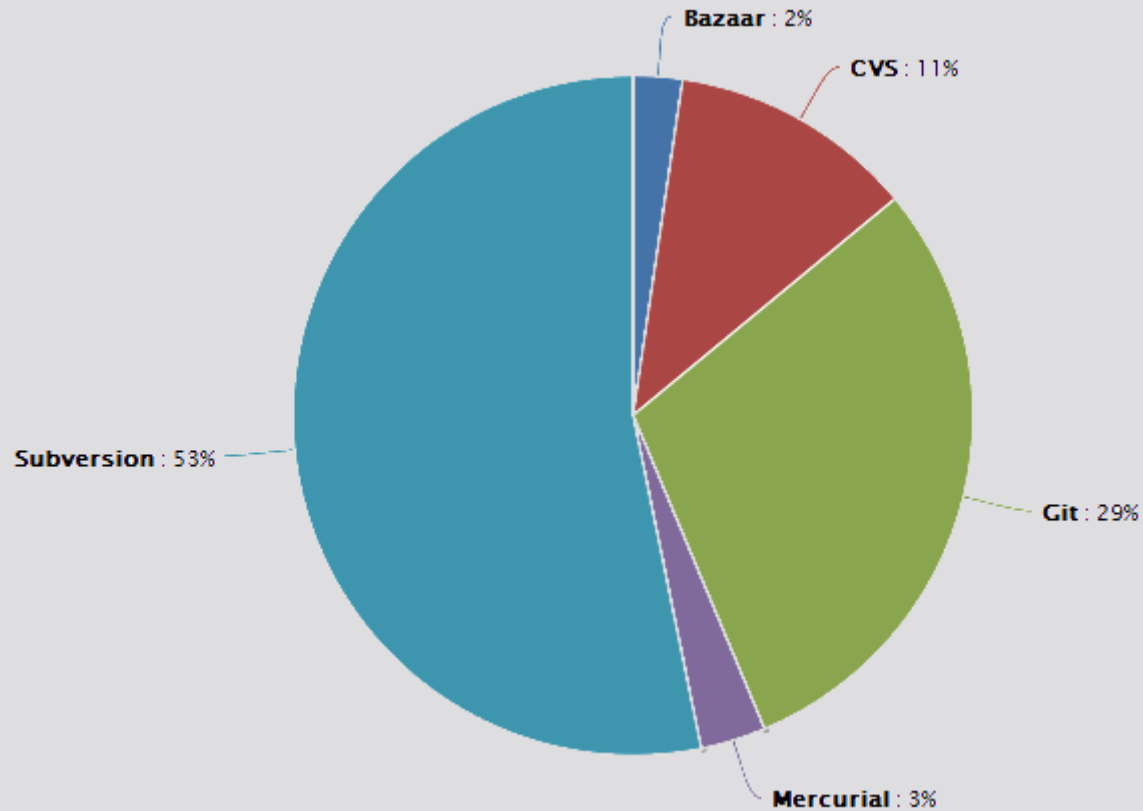
- SVN: TortoiseSVN, RapidSVN, VisualSVN
- Git: gitk, git-gui, TortoiseGit, SmartGit, etc
- Mercurial: Hgk, TortoiseHg

Помимо специализированных клиентов, средства для работы с VCS предоставляются некоторыми IDE.

Хостинг кода

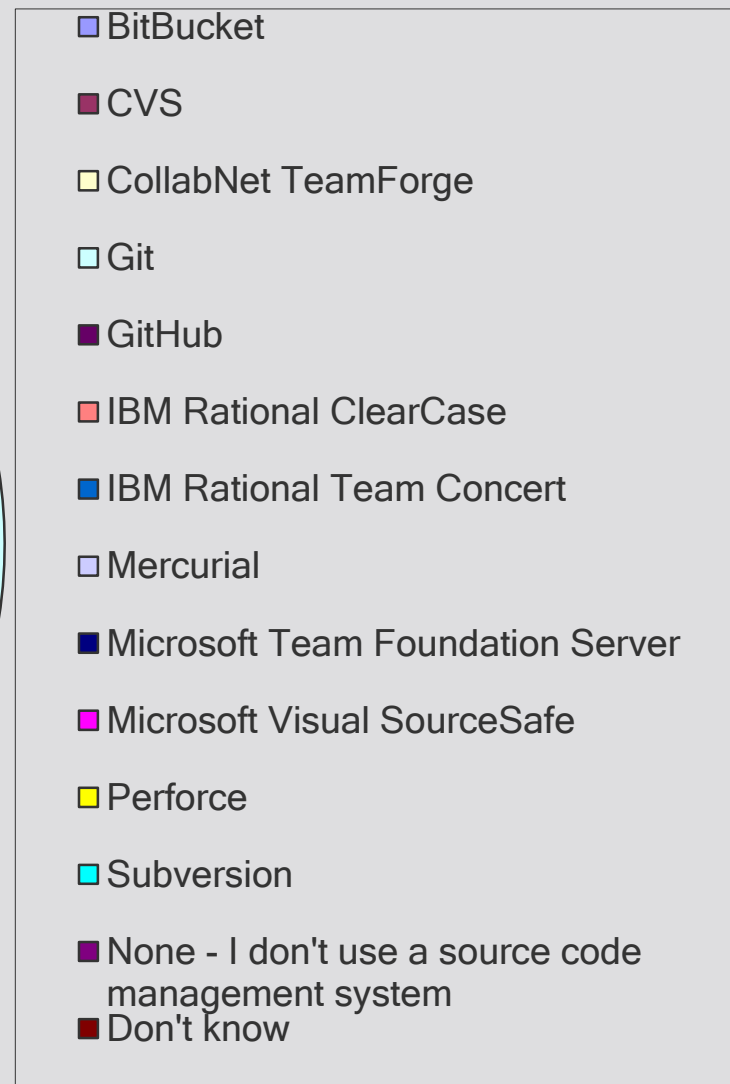
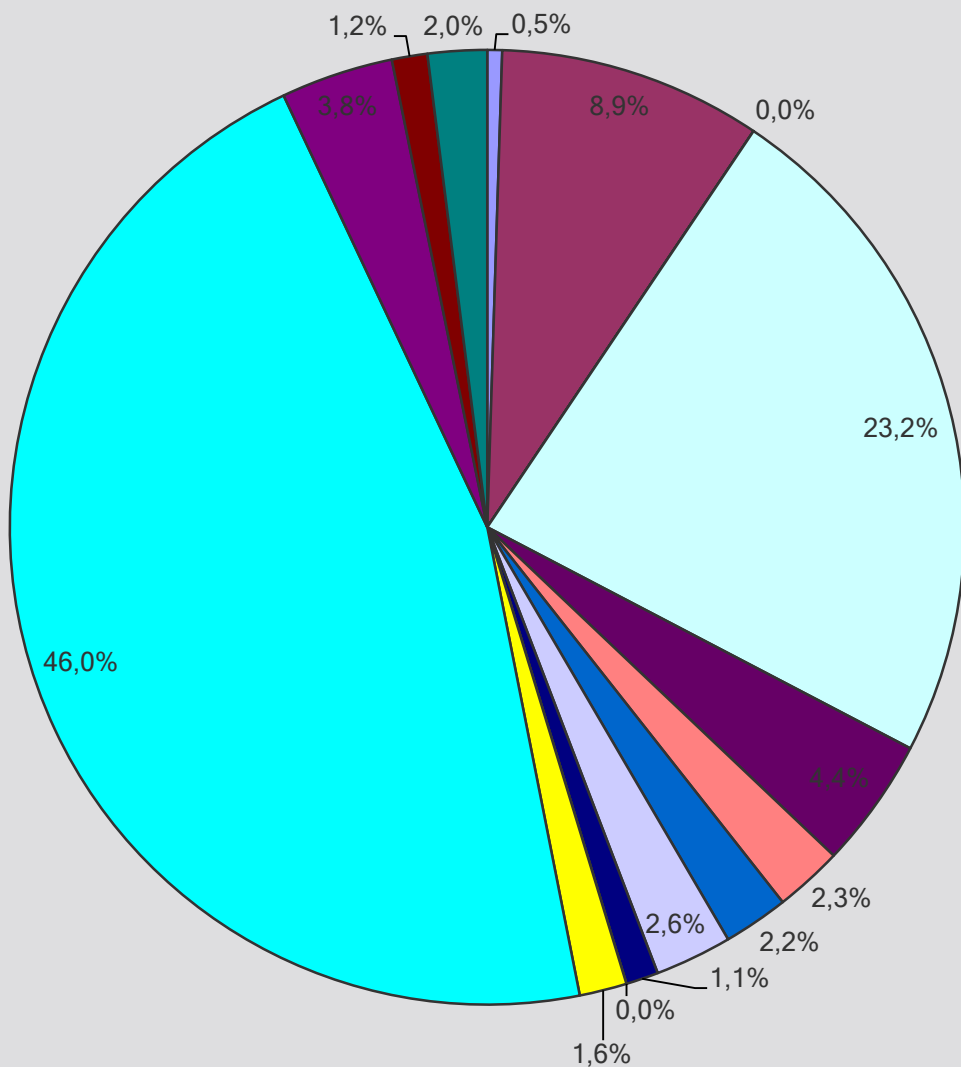
- SVN: Google Code, SourceForge
- Git: Github, Bitbucket, Google Code, Gitorious
- Mercurial: Bitbucket, Google Code

Распространенность



Распространенность различных VCS среди проектов с открытым исходным кодом
(<http://www.ohloh.net/repositories/compare>)

Распространенность



Что почитать

- Pro Git - <http://git-scm.com/book>
- Mercurial: The Definitive Guide - <http://hgbook.red-bean.com/>
- Version Control with Subversion - <http://svnbook.red-bean.com/index.en.html>
- Version Control by Example - <http://www.ericSink.com/vcbe/index.html>