

# Лекция 8. Системы контроля версий

Александр Смаль

CS центр  
8 апреля 2013  
Санкт-Петербург

## Системы контроля версий

- Системы управления версиями (Version Control Systems, VCS) или Системы управления исходным кодом (Source Management Systems, SMS) — важный аспект разработки современного ПО.
- VCS предоставляет следующие возможности:
  - Поддержка хранения файлов в репозитории.
  - Поддержка истории версий файлов в репозитории.
  - Нахождение конфликтов при изменении исходного кода и обеспечение синхронизации при работе в многопользовательской среде разработки.
  - Отслеживание авторов изменений.

## Классификация систем контроля версий

- Централизованные/распределённые — в централизованных системах контроля версий вся работа производится с центральным репозиторием, в распределённых — у каждого разработчика есть локальная копия репозитория.
- Блокирующие/не блокирующие — блокирующие системы контроля версий позволяют наложить запрет на изменение файла, пока один из разработчиков работает над ним, в неблокирующих один файл может одновременно изменяться несколькими разработчиками.
- Для текстовых данных/для бинарных данных — для VCS для текстовых данных очень важна поддержка слияния изменений, для VCS с бинарными данными важна возможность блокировки.

## Ежедневный цикл работы

Обычный цикл работы разработчика в течение дня выглядит следующим образом.

1. **Обновление рабочей копии.**

Разработчик выполняет операцию обновления рабочей копии (update) насколько возможно

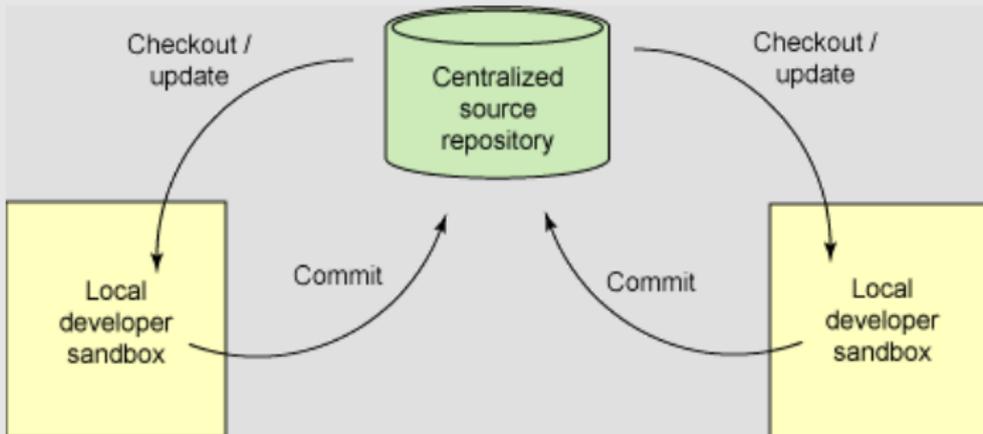
2. **Модификация проекта.**

Разработчик локально модифицирует проект, изменяя входящие в него файлы в рабочей копии.

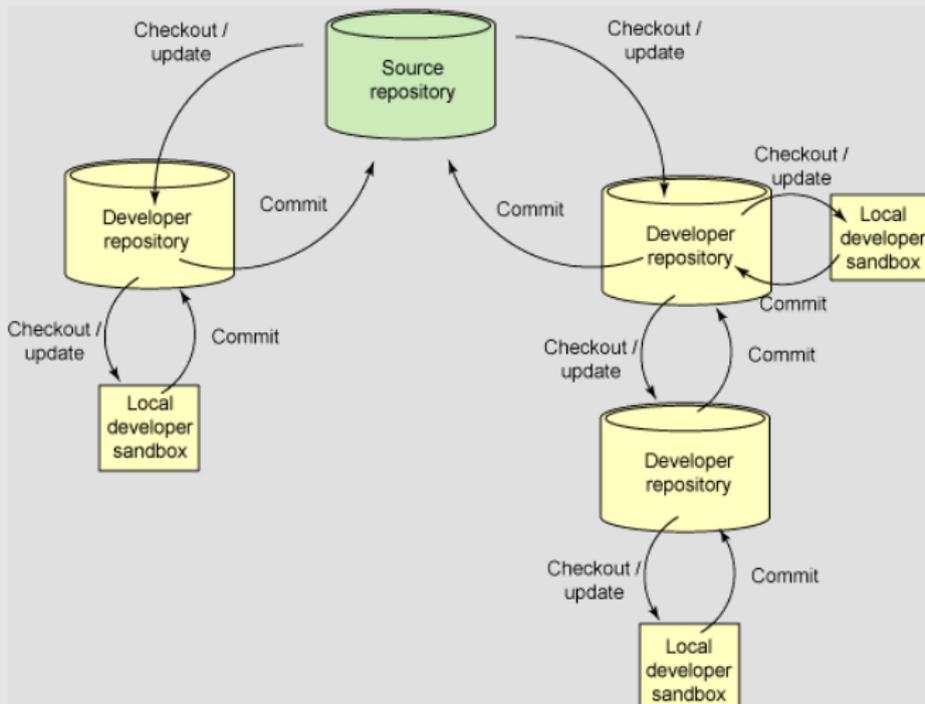
3. **Фиксация изменений.**

Завершив очередной этап работы над заданием, разработчик фиксирует (commit) свои изменения, передавая их на сервер. VCS может требовать от разработчика перед фиксацией выполнить обновление.

## Централизованные VCS



## Распределённые VCS



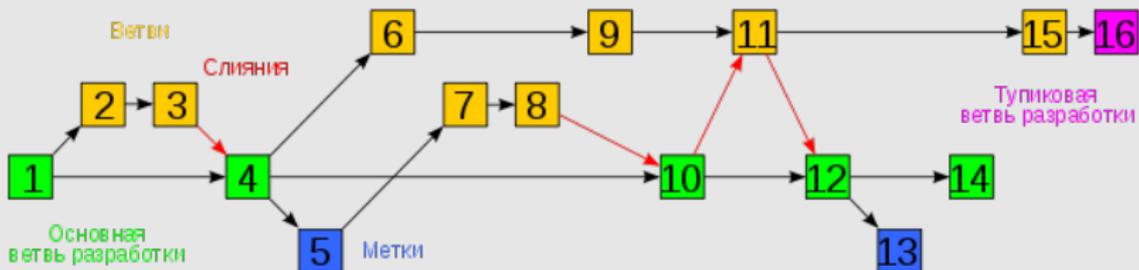
## Основные термины

- working copy — рабочая (локальная) копия документов.
- repository, depot — хранилище.
- revision — версия документа. Новые изменения (changeset) создают новую ревизию репозитория.
- check-in, commit, submit — фиксация изменений.
- check-out, clone — извлечение документа из хранилища и создание рабочей копии.
- update, sync — синхронизация рабочей копии до некоторого заданного состояния хранилища (в т.ч. и к более старому состоянию, чем текущее).
- merge, integration — слияние независимых изменений.
- conflict — ситуация, когда несколько пользователей сделали изменения одного и того же участка документа.
- head — самая свежая версия (revision) в хранилище.

## Ветвление

- Ветвь (branch) — направление разработки проекта, независимое от других.
- Ветвь представляет собой копию части (как правило, одного каталога) хранилища, в которую можно вносить свои изменения, не влияющие на другие ветви.
- Документы в разных ветвях имеют одинаковую историю до точки ветвления и разные — после неё.
- Изменения из одной ветви можно переносить в другую.
- Ствол (trunk, mainline, master) — основная ветвь разработки проекта.

## Пример эволюции ветвей в проекте



## CVS

Одна из наиболее старых систем контроля версий. Создана в 1984 году как развитие RCS (Revision Control System), которая не поддерживала совместную работу.

Недостатки:

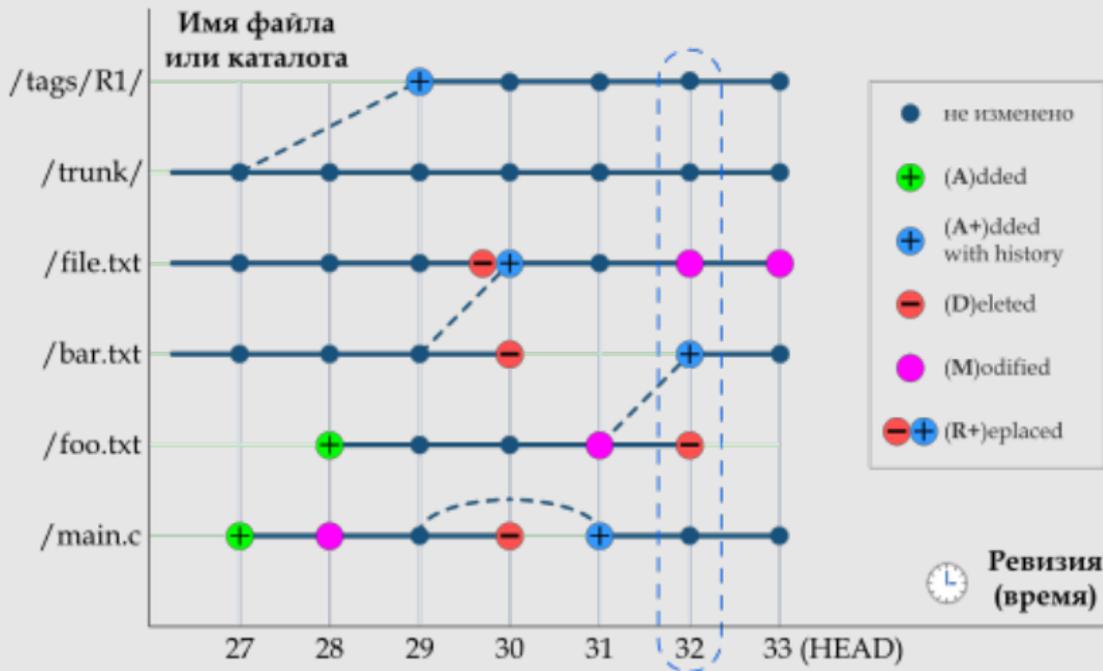
- Невозможно переименовать файл или директорию так, чтобы это изменение было отражено в истории.
- Ограниченная поддержка юникода и не-ASCII имен.
- Публикации изменений не атомарны.
- Наборы изменений не поддерживаются.
- Неэффективное хранение бинарных файлов.
- Оригинальный GNU CVS не поддерживает разграничения прав между пользователями репозитория.

## Subversion, SVN

Subversion, SVN — свободная централизованная система управления версиями, официально выпущенная в 2004 году компанией CollabNet Inc.

- Копирование объектов с разветвлением истории.
- Поддержка ветвления: создания ветвей (копированием директорий) и слияние ветвей (переносом изменений)
- Поддержка меток (копированием директорий).
- Поддержка разделение прав пользователей.
- История изменений и копии объектов (в том числе ветви и метки) хранятся в виде связанных разностных копий.
- Атомарная фиксации изменений в хранилище.

# Представление файловой системы в SVN



## Пример работы SVN

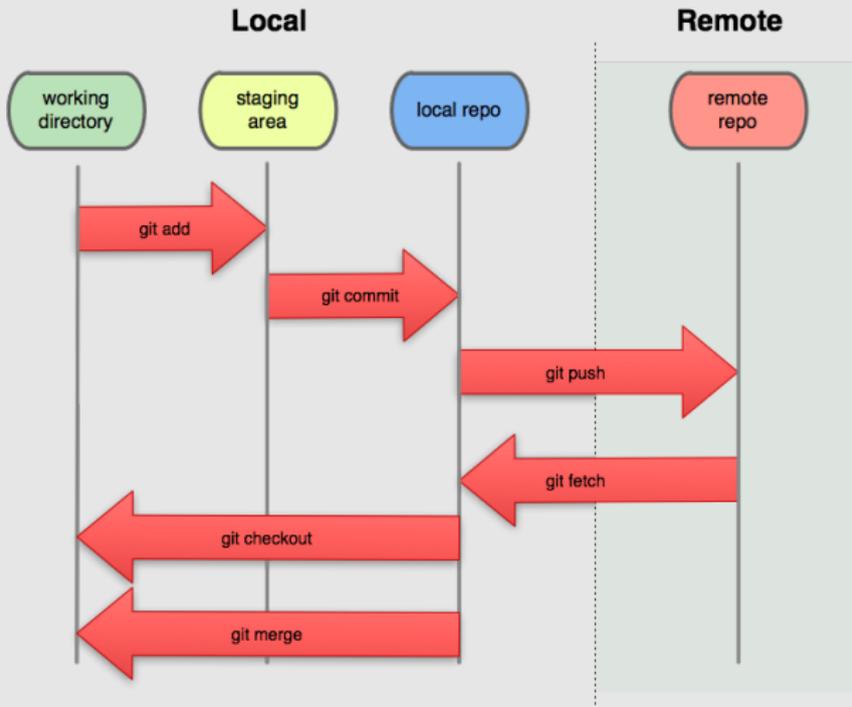
```
$ svn co http://projects.com/svn/myproject --username user
$ cd myproject
  ..change project..
$ svn log
$ svn st
$ svn add newfile.cpp
$ svn rm oldfile.cpp
$ svn commit -m"Mega enhancement"
  ..next day..
$ svn up
```

## GIT

Git — распределённая система управления версиями файлов. Проект был создан Линусом Торвальдсом для управления разработкой ядра Linux в 2005 году.

- + Высокая производительность.
- + Развитые средства интеграции с другими VCS.
- + Репозитории git могут распространяться и обновляться общесистемными файловыми утилитами, такими как rsync.
- Отсутствие переносимой на другие операционные системы поддержки путей в кодировке Unicode.
- Команды ориентированны на наборы изменений, а не на файлы.
- Использование для идентификации ревизий хешей SHA1.
- Отсутствие отдельной команды переименования/переноса файла, которая так отображалась бы в истории.

## Схема работы с Git



## Пример работы Git

```
$ git clone git@projects.com/myproject.git
```

```
$ cd myproject
```

```
..change project..
```

```
$ git log
```

```
$ git status
```

```
$ git add newfile.cpp
```

```
$ git commit -m "Mega enhancement"
```

```
..more commits..
```

```
$ git push
```

```
..next day..
```

```
$ git pull
```

## Mercurial

- Mercurial, он же Hg (от обозначения химического элемента ртути) — кроссплатформенная распределенная система управления версиями, разработанная для эффективной работы с очень большими репозиториями кода.
- Система Mercurial написана на Python.
- Mercurial считается более простой в освоении системой, чем, например, git.
- Предоставляет конкретные команды переименования и копирования, и сохранения этих действий как истории файла.
- Бережно относится к истории.

## Пример работы Mercurial

```
$ hg clone http://projects.com/repo/myproject
```

```
$ cd myproject
```

```
..change project..
```

```
$ hg log
```

```
$ hg status
```

```
$ hg add newfile.cpp
```

```
$ hg commit -m"Mega enhancement"
```

```
..more commits..
```

```
$ hg push
```

```
..next day..
```

```
$ hg pull
```

## Бесплатные VCS серверы

Есть много сервисов, которые предоставляют открытые репозитории для совместной работы:

- [sourceforge.net](http://sourceforge.net) — SVN, Git, Mercurial, Bazaar, CVS репозитории.
- [code.google.com](http://code.google.com) — SVN, Git, Mercurial репозитории.
- [github.com](http://github.com) — Git репозитории.
- [bitbucket.org](http://bitbucket.org) — Git и Mercurial репозитории.
- ...

Спасибо за внимание!