

Введение в Git

Роман Чепляка

18 июня 2010
EPAM Systems

- 1 `svn checkout svn.epam.com`
- 2 `work`
- 3 `svn update`
- 4 `resolve conflicts`
- 5 `svn commit`
- 6 `goto 2`

- Коммиты сразу идут на сервер
 - Коммит не делается до того, как функциональность готова и протестирована
- Бранчи существуют только на сервере
 - Создание бранча – дорогая операция

- 1 `git clone git.epam.com`
- 2 `work`
- 3 `git commit -a`
- 4 `git pull --rebase`
- 5 resolve conflicts
- 6 `goto 2`
- 7 `git push`

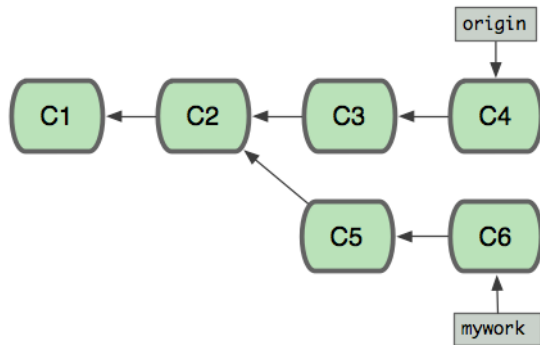
Workflow в Git: преимущества

- Коммиты записываются локально
- Коммиты можно редактировать перед тем, как запустить
 - Коммитить можно что угодно
 - Коммитить можно (и нужно!) часто
- Локальные ветки
 - Создание ветки – дешевая операция
 - Ветки можно (и нужно!) создавать для экспериментов

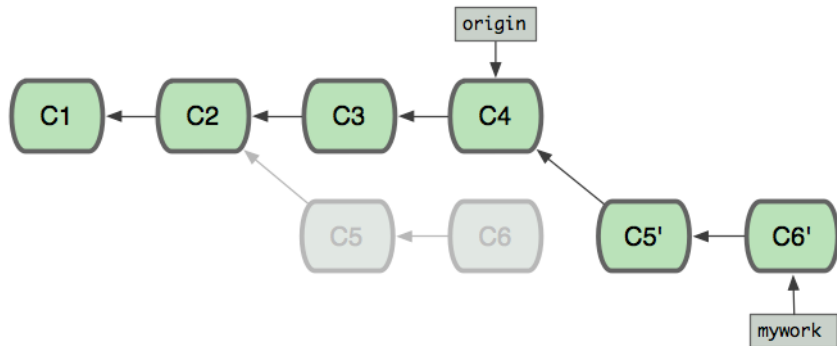
Причины делать маленькие коммиты

- Легче манипулировать (например, выбросить и заменить другим)
- Легче читать
- Легче cherry-pick'нуть
- Безопаснее

Rebase



Rebase



- Ветка – ссылка на commit
- Бывают локальные (`git branch`) и удаленные (`remote`, `git branch -r`)
- Локальные – для экспериментов, отдельных фич
- Удаленные ветки – для отслеживания удаленных репозиториях
- Не делайте коммиты в удаленные ветки!

- Подмодули (submodules) позволяют разбивать проект на компоненты.
- Подмодуль – автоматический клон внешнего репозитория в поддиректории
- Застрахованы от изменений во внешнем репозитории – явно указываем, какой коммит нам нужен

- Выступление Линуса Торвальдса о Git на Google TechTalk
- Git Community Book
- Pro Git: professional version control
- Git user's manual
- Рекомендуется создать свою wiki для документирования процессов

Скорая помощь:

- Roman_Cheplyaka@epam.com