Системы контроля версий. Документация проекта.

Елена Кантонистова

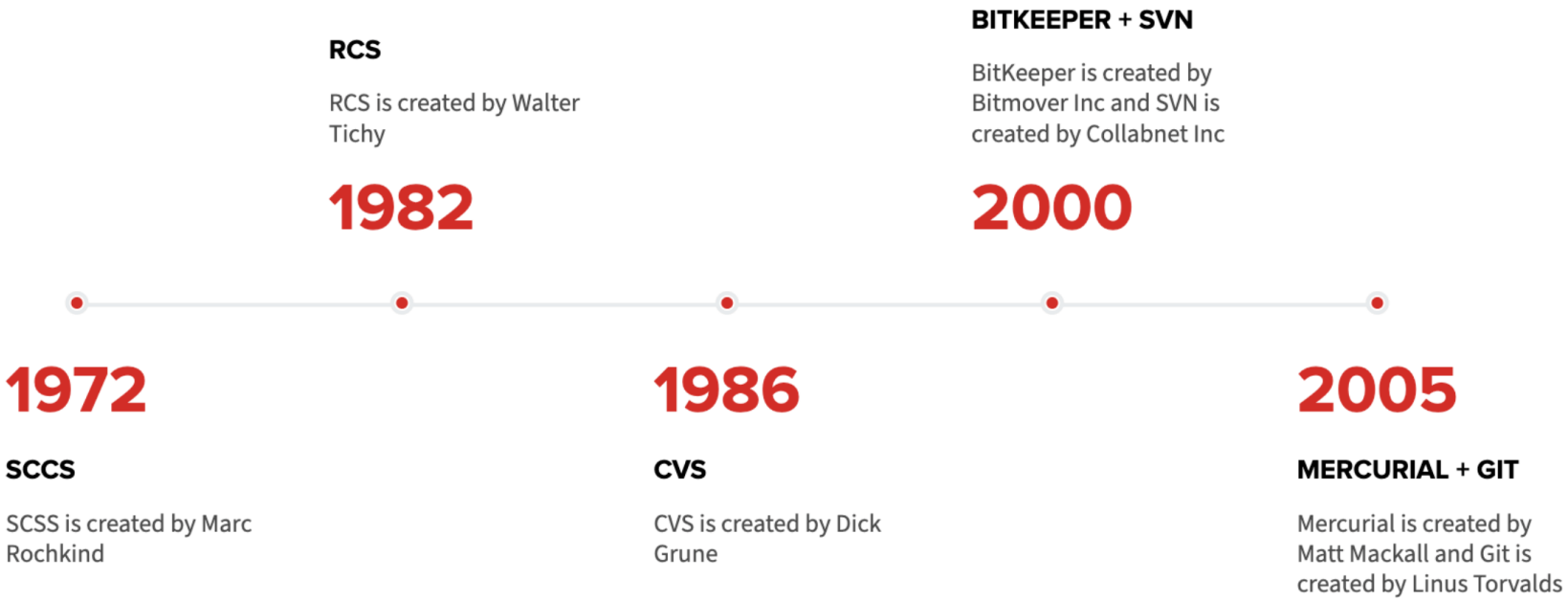
Системы контроля версий

- Если вы работаете над проектом, тем более в составе команды, необходимо сохранять промежуточные этапы.

Система контроля версий – это система, записывающая изменения в файл или набор файлов в течение времени и позволяющая вернуться позже к определенной версии.

Системы контроля версий

- **Локальные (**RCS**)**
- **Централизованные (**CVS, Subversion и Perforce**)**
- **Распределенные (**Git, Mercurial, Bazaar или Darcs**)**



Системы контроля версий

Локальные (RCS**)**

- для каждого файла, зарегистрированного в системе, хранится полная история изменений

Недостатки?

Системы контроля версий

Локальные (RCS**)**

- для каждого файла, зарегистрированного в системе, хранится полная история изменений

Недостатки?

- Не подходит для работы в команде

Системы контроля версий

Централизованные (CVS, Subversion и Perforce**)**

- Для организации такой системы контроля версий используется единственный сервер, который содержит все версии файлов
- Клиенты, обращаясь к этому серверу, получают из этого централизованного хранилища.

Недостатки?

Системы контроля версий

Централизованные (CVS, Subversion и Perforce**)**

- Для организации такой системы контроля версий используется единственный сервер, который содержит все версии файлов
- Клиенты, обращаясь к этому серверу, получают из этого централизованного хранилища.

Недостатки?

- Если сервер упал, то работа встанет

Системы контроля версий

Распределенные (Git, Mercurial, Bazaar или Darcs**)**

- клиенты не просто выгружают последние версии файлов, а полностью копируют весь репозиторий (место, где хранятся и поддерживаются какие-либо данные)
- можно выделить центральный репозиторий (условно), в который будут отправляться изменения из локальных и, с ним же эти локальные репозитории будут синхронизироваться

Git

Стандарт де-факто на сегодня – распределенная система контроля версий Git.

GitHub

Платформа GitHub позволяет создавать удалённые версии репозитория для работы с коллегами, а также делиться своими проектами с другими.

Основы работы с Git

В Git обширный функционал, но для знакомства с инструментом понадобится только несколько основных команд:

- git clone – клонирует существующий репозиторий на локальную машину
- git commit – добавляет файлы в список отслеживаемых для регистрации истории изменений
- git push – добавление локальных изменений в проект на GitHub
- git pull – загрузка изменений с GitHub на локальную машину

GitHub Desktop

Что делаем

- Регистрируемся на сайте github
- Клонируем репозиторий с материалами курса

- ДЗ: создаем репозиторий командного проекта + readme
- Клонируем его
- Загружаем туда файл с данными из БД
- Загружаем туда ноутбук с экспериментами