

SKILLFACTORY

Git и Github. Практика



Кирилл Крайнов

iOS-разработчик

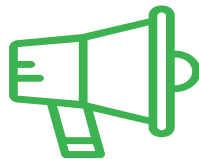
Правила вебинара



**Отложите все
дела на
ближайший
час**



**Фиксируйте
важную
информацию
на вебинаре**



**Задавайте
вопросы в чат**



**Делитесь
мнением в
чате**



**Ура! Вы
превосходны!**

Запись вебинара будет выложена на платформе lms.skillfactory.ru

План вебинара

1. Что такое Git и зачем он нужен
 - а. принципы работы
2. Существующие Git-системы
3. Ответы на вопросы с теоретического Git-вебинара
4. Github
 - а. в Xcode
 - б. команды
5. Связки

БОНУС: Полезные ссылки по GIT

6. Ваши вопросы

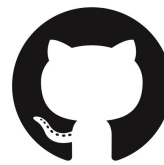
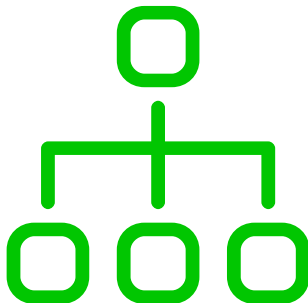


Небольшое пояснение



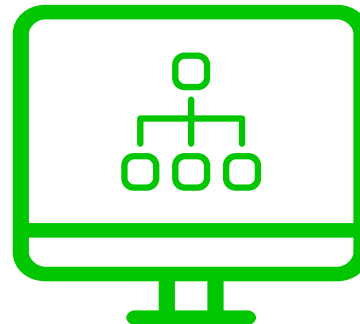
Git

**Git - система управления
версиями**



Github

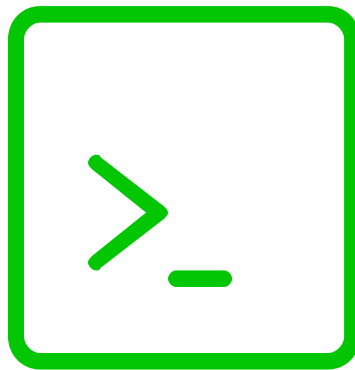
Одна из Git-оболочек



Что такое Git и зачем он нужен



- Система контроля версий
- Изначально помогала изобретателю Linux управлять разработкой ядра
- В 2005 году вышел первый релиз
- “Абсолютный” бэкенд - до оболочек типа Github люди ориентировались на терминал
- Хранит репозитории и их изменения
- Есть устойчивое соглашение структуры



Существующие Git-системы

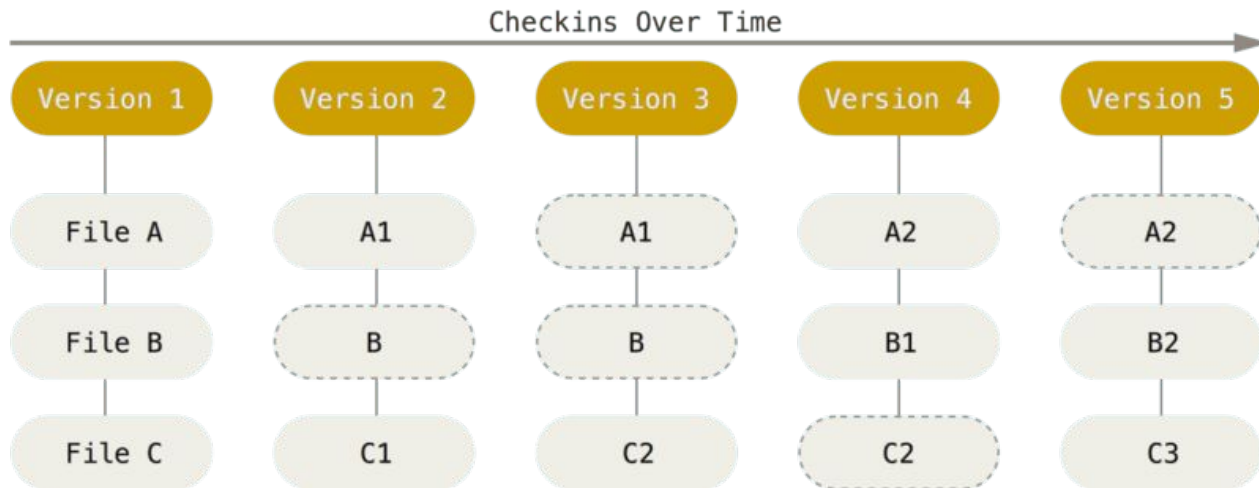


- Gitlab
- **Github**
- Bitbucket
- Gitea
- SourceForge



В 98% случаев используются
эти системы при разработке
ПО на территории РФ

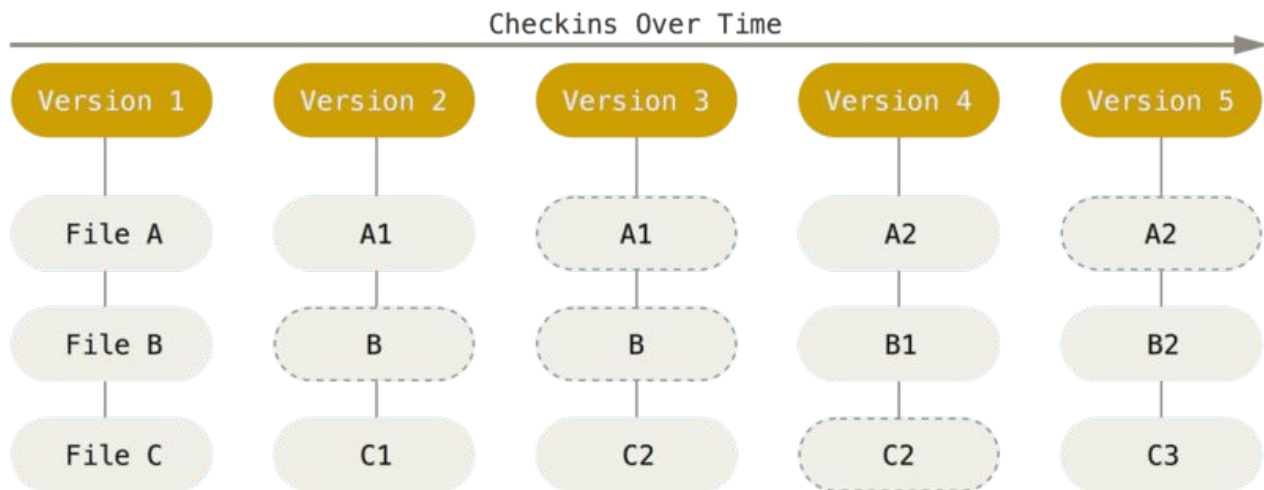
Git. Принципы работы



Источник:

<https://git-scm.com/book/ru/v2/%D0%92%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5-%D0%A7%D1%82%D0%BE-%D1%82%D0%B0%D0%BA%D0%BE%D0%B5-Git%3F>

Git. Принципы работы



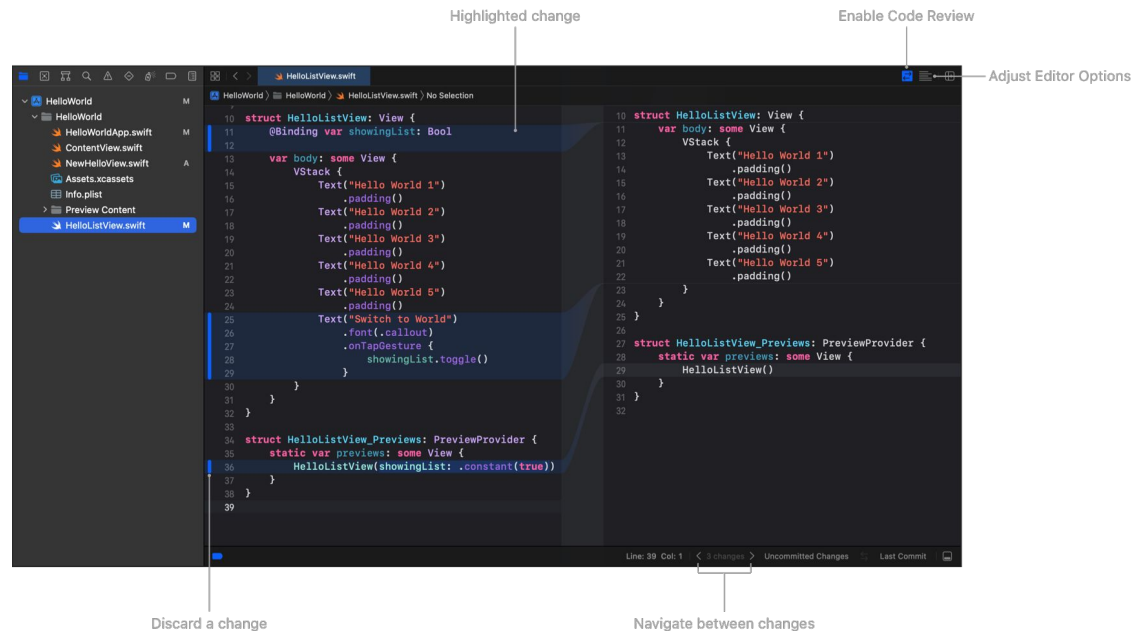
- Система запоминает, как выглядит каждый файл в этот момент, и сохраняет ссылку на этот снимок
- Сохраняет только изменённые файлы - не дублирует весь проект
- Предоставляет данные в виде потока снимков файловой системы

Ответы на вопросы с теоретического Git-вебинара

Имеет ли смысл качать GIT Desktop или лучше сразу учиться пользоваться Xcode?

В первую очередь, лучше освоить Git через терминал, разобрать логику его работы, а потом уже использовать какие-то визуальные средства

Github. В Xcode



Источник: <https://developer.apple.com/documentation/xcode/tracking-code-changes-in-a-source-control-repository>

Git и Github. Практика



Практический блок Xcode

Github. Команды



- **Подключение через терминал**
- **Превращаем папку с проектом в репозиторий**
- **Клонирование репозитория для внесения изменений**
 - Статус изменений
 - Создание коммита
 - Добавление файлов
 - Добавление в ветку
 - Подтверждение и отправка изменений в репозиторий
- **Переключение между ветками (“ориентирование” в Git)**

Ответы на вопросы с теоретического Git-вебинара

А нумерация версий (0.1, 0.2, 1.0 и т.д.) и отдельных релизов обновляется разработчиками вручную или Git как-то эту нумерацию помогает автоматизировать?

Какой-то автоматизированной версионности без настроек получить сложно. Если делать по хардкору, то можно снабжать каждый коммит номером версии.

По умолчанию, о версии мы можем получить только её уникальный ID или что-то вроде хэшкода.



Подключение через терминал

Это можно пропустить, если вы подключены через Xcode

Однако имеет смысл подключаться глобально, а не только через IDE

git config
git remote



Превращаем папку с проектом в репозиторий

Это может делать Xcode в самом начале проекта, но понятнее через терминал

git init



Клонирование репозитория для внесения изменений

- Статус изменений

git status



Клонирование репозитория для внесения изменений

- Создание коммита

git commit



Клонирование репозитория для внесения изменений

- Добавление файлов

git add



Клонирование репозитория для внесения изменений

- Добавление в ветку

git branch



Клонирование репозитория для внесения изменений

- Подтверждение и отправка изменений в репозиторий

git commit

git push



Переключение между ветками (“ориентирование” в Git)

git checkout

Git и Github. Практика



Слияние веток

git merge

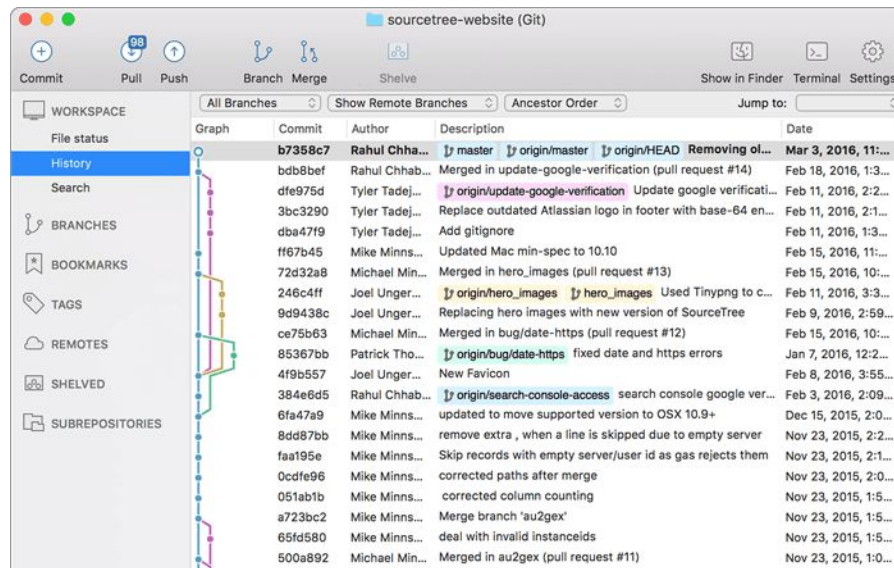
Ответы на вопросы с теоретического Git-вебинара

А если у нас произойдут изменения в ветке `main` ветке, то ветка `release` также обновиться ? Т.е у нас есть ветка `release`, которую мы не влили в мастер, но например другая команда изменила ветку `main`, то неопубликованная ветка `release` также обновиться ?

Если под вливанием в мастер подразумевается `merge` веток, то вряд ли какая-либо ветка получит изменения.



SourceTree для визуализации



Источник: <https://techrocks.ru/2018/08/06/best-tools-for-productive-development-2018/>

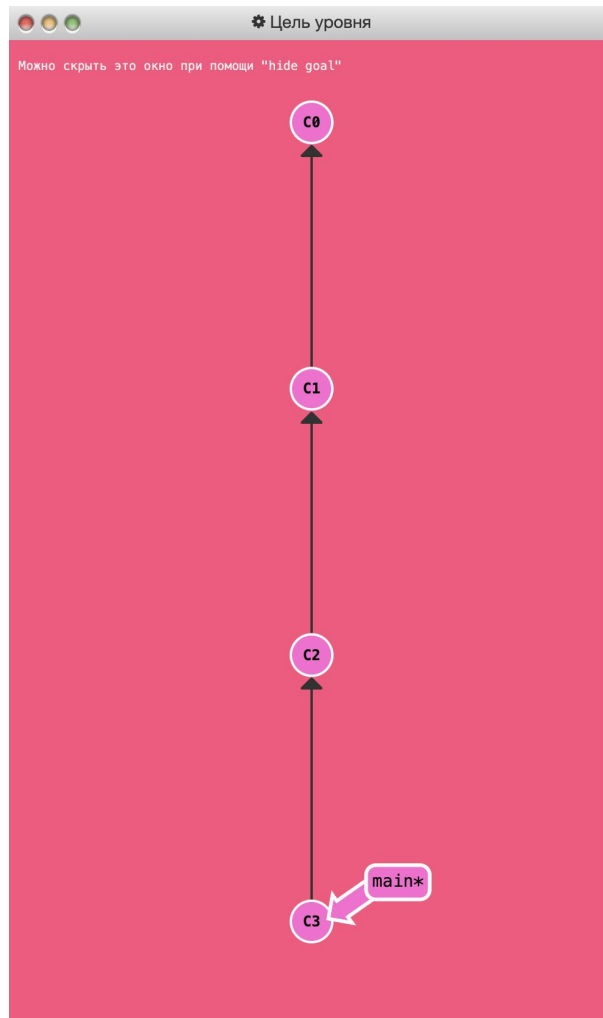
Git и Github. Практика

Git визуально



Git тренажер -

<https://learngitbranching.js.org/>



Задания



1. Создать свой аккаунт на Github
2. Потренироваться в связывании репозиториев
3. Опробовать git-команды на своих проектах

P.S. ошибаться разрешается, только так можно понять.

БОНУС: Полезные ссылки по Git

- **Apple. Инструкция по связыванию Xcode с Github -** <https://developer.apple.com/documentation/xcode/configuring-your-xcode-project-to-use-source-control>
- **Git book.** Документация на русском - <https://git-scm.com/book/ru/v2>
- **Git тренажер** - <https://learngitbranching.js.org/>
- **Git тренажер от Github** - <http://git-school.github.io/visualizing-git/>
- **Habr. Git для новичков.** Ч1 - <https://habr.com/ru/post/541258/>
- **Habr. Git для новичков.** Ч2 - <https://habr.com/ru/post/542616/>
- **Habr. 30 команд** - <https://habr.com/ru/company/ruvds/blog/599929/>
- **Habr. Gitflow** - <https://habr.com/ru/post/106912/>
- **Habr. Github без командной строки** - <https://habr.com/ru/post/233719/>
- **Bitworks Software. Gitflow** - <https://bitworks.software/2019-03-12-gitflow-workflow.html>

Ваши вопросы

Блок вопросов и ответов объявляется открытым!

Обратная связь по вебинару

Мы будем признательны за обратную связь с вашей стороны!

Приветствуются комментарии и пожелания, чтобы мы могли стать лучше!

Пожалуйста, ставьте активнее реакции, чтобы следующий вебинар состоялся :)