







UNIVERSITY

ЛЕКЦІЯ 2

“Знайомство з системами контролю версій”

2

Знайомство з системами контролю версій

Огляд систем контролю версій

Основи роботи з GIT

GIT та xCode

GIT та SourceTree

Знайомство з системами контролю версій

Огляд систем контролю версій

Основи роботи з GIT

GIT та xCode

GIT та SourceTree

Огляд систем контролю версій

Це система, яка реєструє зміни у одному або декількох файлах, завдяки чому виникає поняття доступу до попередніх версій файлу (-ів).

Це потужний інструмент, який дозволяє одночасно, без завад один одному, проводити роботу над групами об'єктів (файлів).

Огляд систем контролю версій - GIT

Створена Лінус Торвальдсом

Випущена під ліцензією GNU GPL версії 2.

Має як командний так і графічний інтерфейс

Використовується віддалено з git-demon, SSH або ж HTTP

Одна з найнадійніших систем

Основні функції

Збереження декількох версій одного й того самого документу.

Збереження історії розробки.

Можливість повернення до більш ранній версій документу.

Визначення того, хто і коли зробив зміни до документа.

Синхронізація змін, виконаних різними членами команди.

Реалізація альтернативних варіантів вмісту документу.

Типи систем контролю версій

Локальна

Централізована

Розподілена

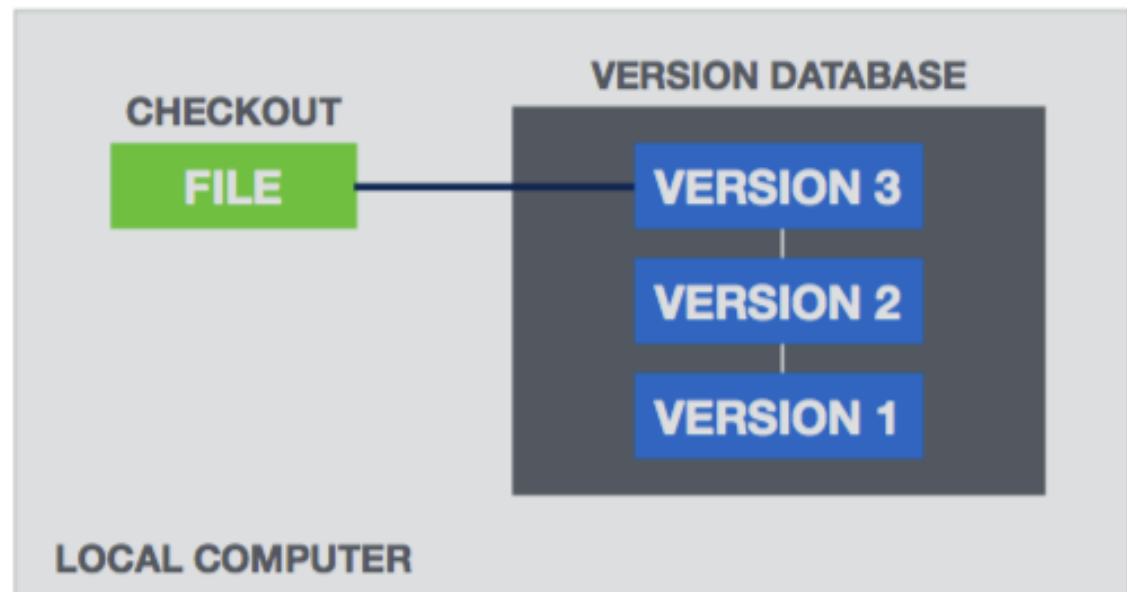
Типи систем контролю версій

Локальна

Централізована

Розподілена

Базується на базі даних, яка містить історію змін файлів.



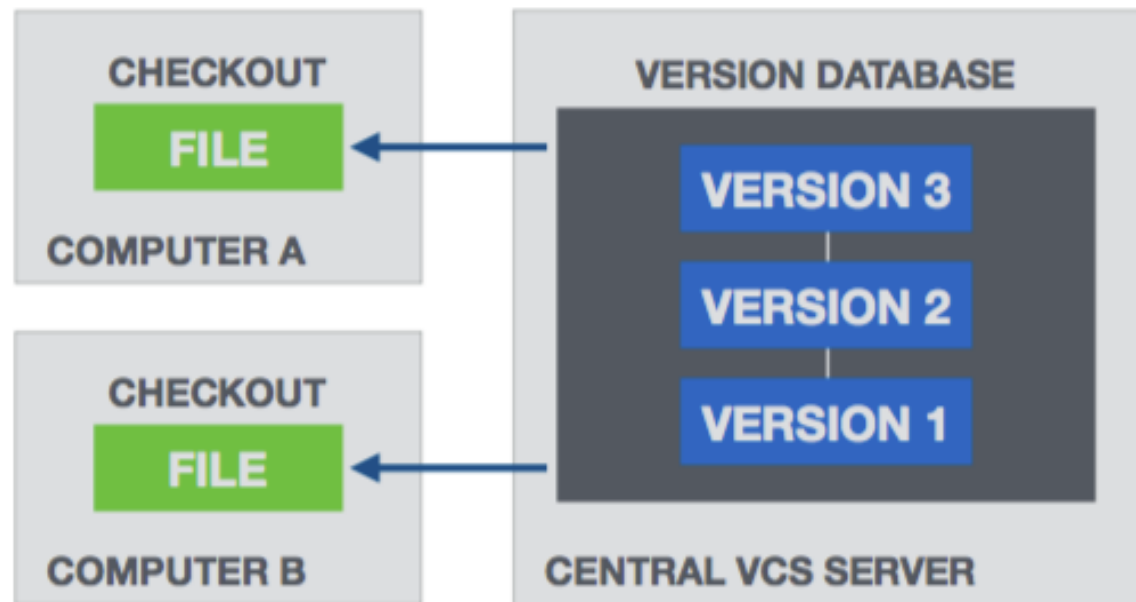
Типи систем контролю версій

Локальна

Централізована

Розподілена

Базується на центральному сервері даних, який зберігає всі файли під контролем версій; а також клієнтських комп'ютерах, які взаємодію з цим сервером.



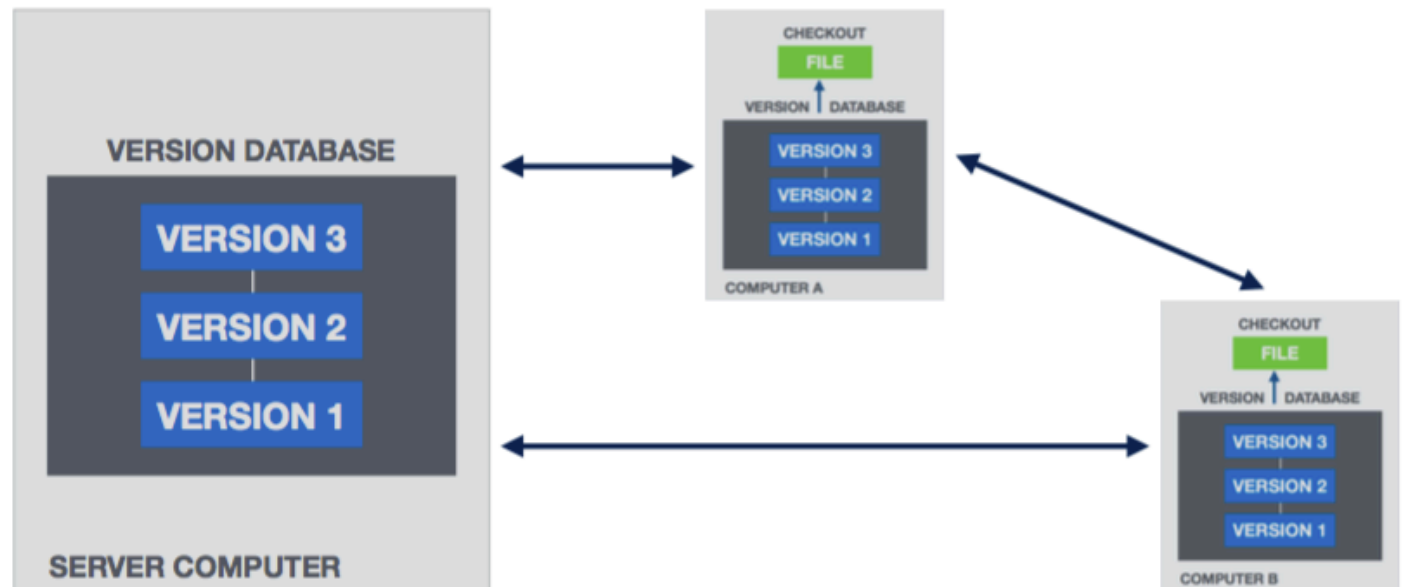
Типи систем контролю версій

Локальна

Централізована

Розподілена

Базується на сервері, а також клієнтських комп'ютерах. Зберігають на останні версії файлів, а весь репозиторій.



Знайомство з системами контролю версій

Огляд систем контролю версій

Основи роботи з GIT

GIT та xCode

GIT та SourceTree

Знайомство з системами контролю версій

Огляд систем контролю версій

Основи роботи з GIT

GIT та xCode

GIT та SourceTree

Git

Це розподілена система контролю версій.

Є однією з найефективніших, надійних і високопродуктивних систем керування версіями, що надає гнучкі засоби нелінійної розробки, що базуються на відгалуженні і злитті гілок.

Особливості Git

Швидкість.

Простота архітектури.

Підтримка нелінійної (паралельної) розробки.

Повна розподіленість.

Ефективні в роботі з великими проектами.

Специфіка роботи з Git

Майже всі операції можна виконувати локально, завдяки чому забезпечується робота без доступу до сервера даних, що веде до підвищення швидкості у порівнянні даних.

Автоматична підтримка цілісності даних, на основі контрольної суми, що являє собою індекс файлу (40 шістнадцяткових символів).

Можливість повернутися до попередніх даних.

Поняття «станів» документу Git

Зафіксований: документ зі змінами збережено в локальній базі версій.

Підготовлений: підготовлення документу до внесення через зафіксування (commit).

Змінений: фізично змінений документ, але ще не зафіксований.

Поняття «областей» документу Git

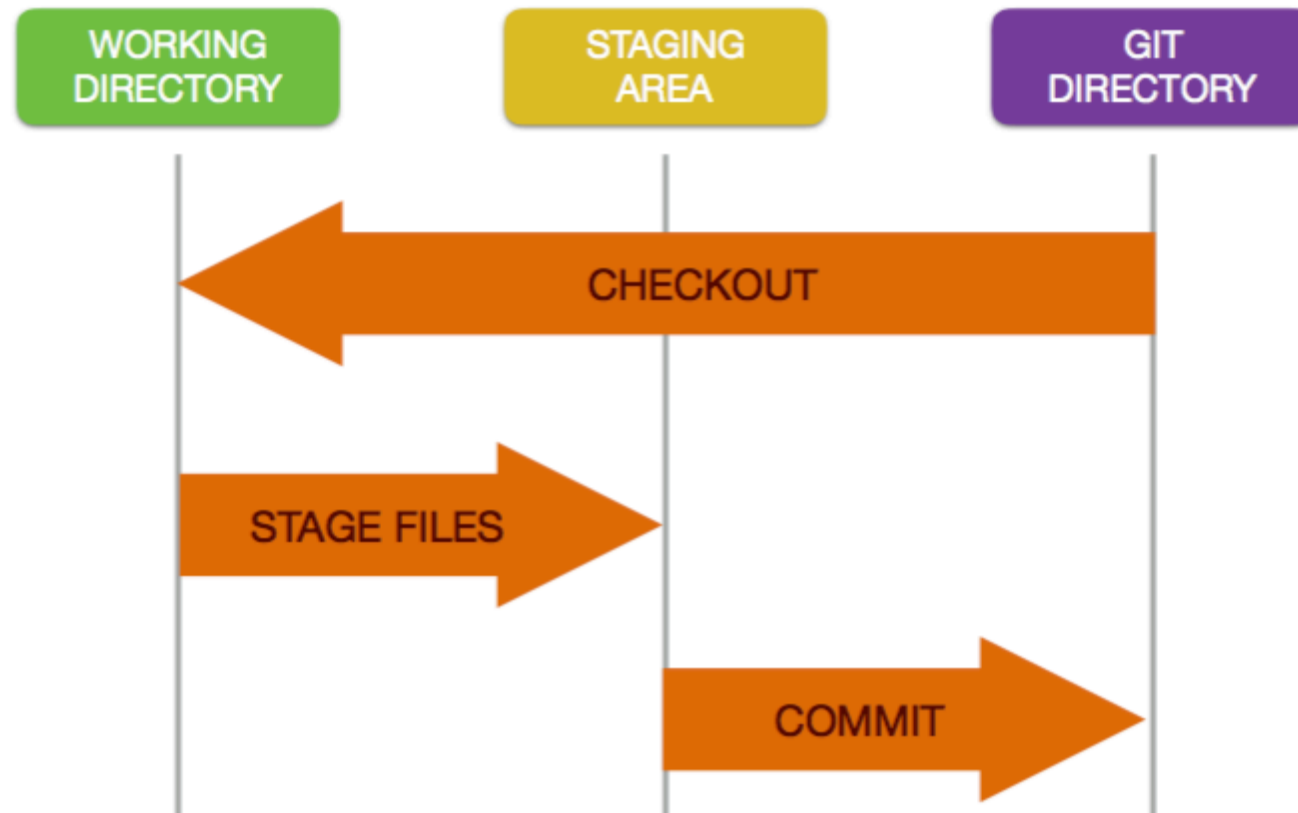
Git Directory (Repository) - каталог (репозиторій) Git.

Working Directory - робочий каталог.

Staging Area - область підготовлених файлів.

Fork - копія сховища. Можна розглядати як зовнішню гілку

Локальні операції Git



Процес роботи з Git

Внесення змін до файлів, що знаходяться у робочому каталозі.

Підготовлення файлів, додавання файлів у область підготовлених файлів.

Фіксування файлів за індексом у каталозі Git.

Операції з Git

Інсталювання.



```
$ ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"  
$ brew doctor  
  
$ brew install git  
  
$ git  
  
$ git --version
```

Операції з Git

Налаштування.



```
$ git config
usage: git config [<options>]

Config file location
--global          use global config file
--system          use system config file
--local           use repository config file
-f, --file <file> use given config file
--blob <blob-id>  read config from given blob object

Action
--get             get value: name [value-regex]
--get-all        get all values: key [value-regex]
....
```


Операції з Git

Утиліта порівняння



```
$ git config --global merge.tool vimdiff
```

```
git config merge.conflictstyle diff3
```

```
git config mergetool.prompt false
```

Операції з Git

Створення репозиторію



```
#init
$ cd /path/to/your/existing/code
$ git init

#clone
$ git clone <repo url>
$ git clone <repo url> <dir name>
```

Операції з Git

Індексація файлів



```
#add to index  
$ cd /path/to/your/existing/code  
$ git add <filename>  
$ git add -n . a  
  
#remove  
$ git remove <filename>
```

Операції з Git

Статус



```
$ cd /path/to/your/existing/code  
$ git status
```

Операції з Git

Статус

untracked: містить неіндексовані файли(виконання add).

unmodified: містить індексовані файли (виконання remove/edit).

modified: містить змінені файли.

staged: містить зафіксовані файли (виконання commit/reset).

Операції з Git

Збереження змін



```
#save changes  
$ cd /path/to/project  
$ echo "test content for git description» >> NewFile.txt  
$ git add NewFile.txt  
$ git commit -m "added CommitTest.txt to the repo"
```

Операції з Git

Логування змін



```
$ cd /path/to/project  
$ git log  
$ git log -graph
```

Операції з Git

Відміна змін



```
$ git commit -amend #додати файл, зміни в коментарі (description)  
$ git reset HEAD <file> #відміна індексації  
$ git checkout --<file> #відміна всіх змін у файлі (!)  
$ git reset HEAD^ #відміна коміту з збереженням змін
```


Операції з Git

Зміни



```
$ git diff -ammend #змінено, але не проіндексовано  
$ git diff --staged #проіндексовано, ввійде в фіксацію (commit)
```

Операції з Git

Робота з віддаленим репозиторієм



```
$ git remote # список репозиторіїв  
$ git remote -v  
$ git remote add <name> <url> # додавання репозиторія  
  
$ git pull #приймання даних глобального репозиторію до локальний репозиторій  
$ git push #відправка даних локального репозиторію до глобального репозиторію  
$ git push origin master --force
```

Знайомство з системами контролю версій

Огляд систем контролю версій

Основи роботи з GIT

GIT та xCode

GIT та SourceTree

Знайомство з системами контролю версій

Огляд систем контролю версій

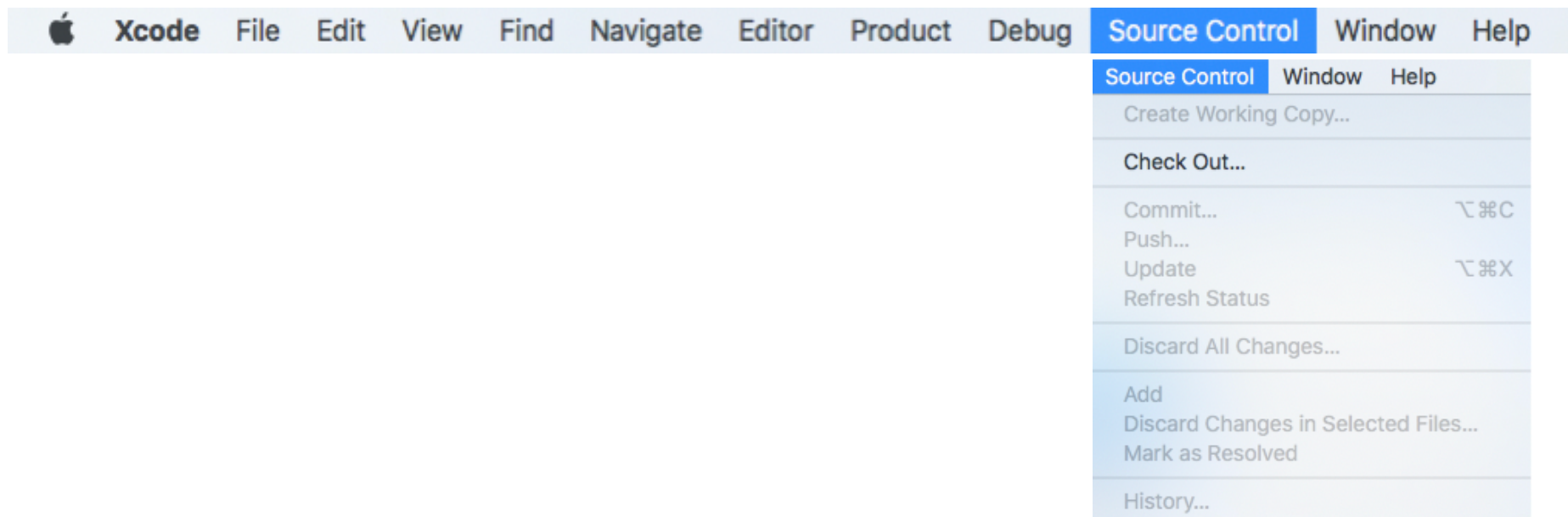
Основи роботи з GIT

GIT та xCode

GIT та SourceTree

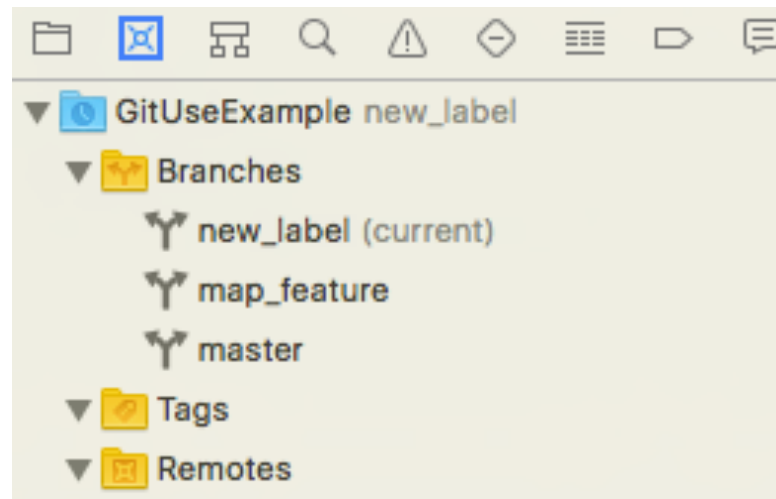
Git та xCode

Меню «SourceControl»



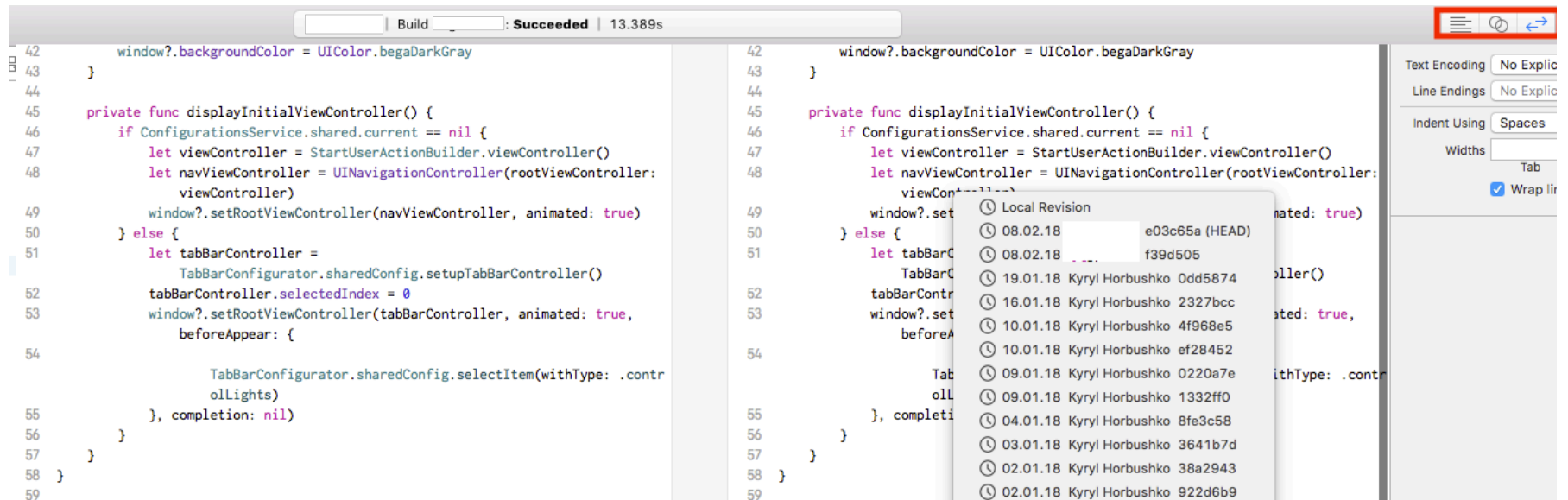
Git та xCode

Source Control NavigationPanel



Git та xCode

TimeTravel



Git та xCode

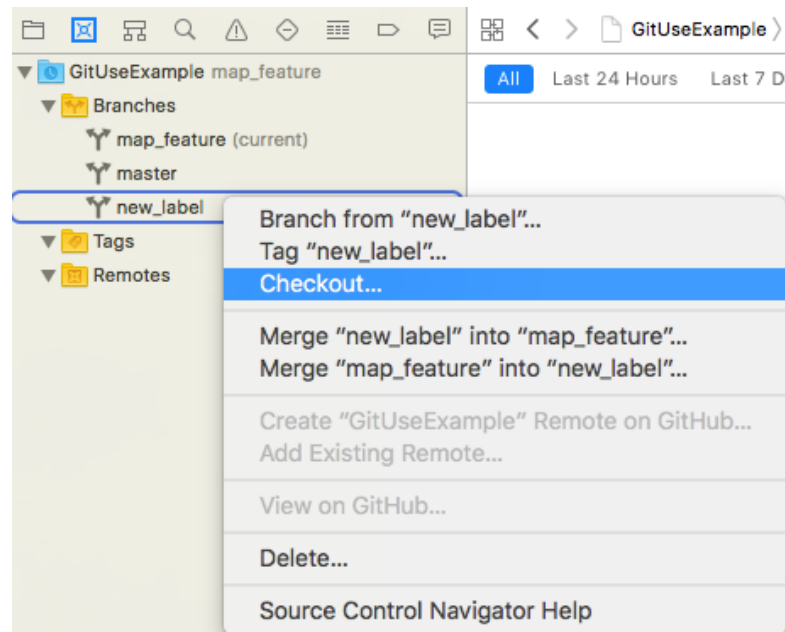
Author



```
1 //
2 // ATOnOffState.swift
3 //
4 //
5 // Created by Kirill Gorbushko on 29.01.18.
6 // Copyright © 2018 - present . All rights reserved.
7 //
8
9 import Foundation
10
11 enum ATSwitchMode: Int {
12
13     typealias RawValue = Int
14
15     case on = 1
16     case off = 0
17 }
18
```

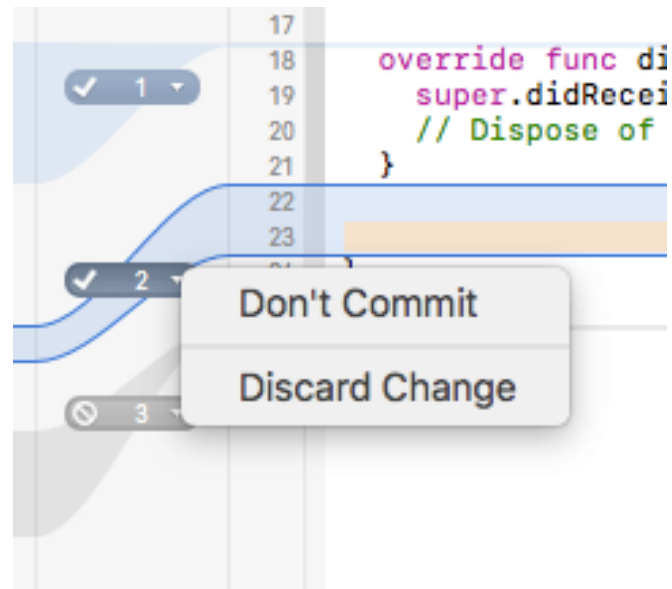

Git та xCode

Merge



Git та xCode

FileMerge



Знайомство з системами контролю версій

Огляд систем контролю версій

Основи роботи з GIT

GIT та xCode

GIT та SourceTree

Знайомство з системами контролю версій

Огляд систем контролю версій

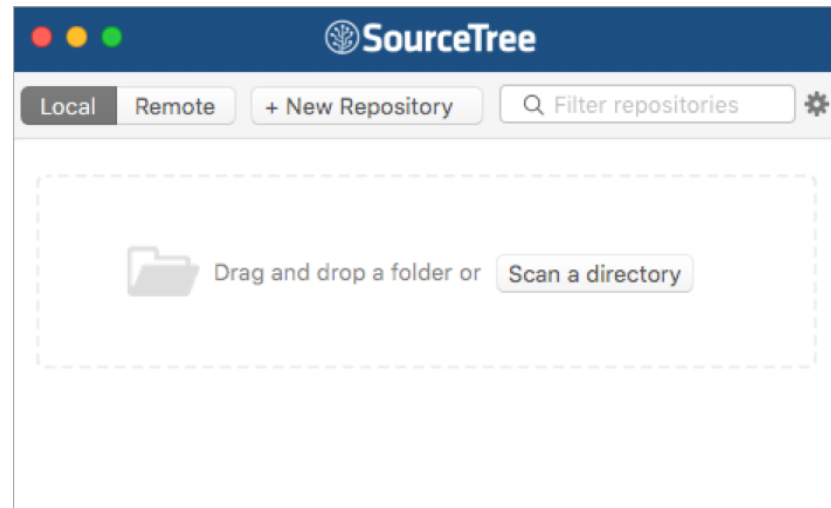
Основи роботи з GIT

GIT та xCode

GIT та SourceTree

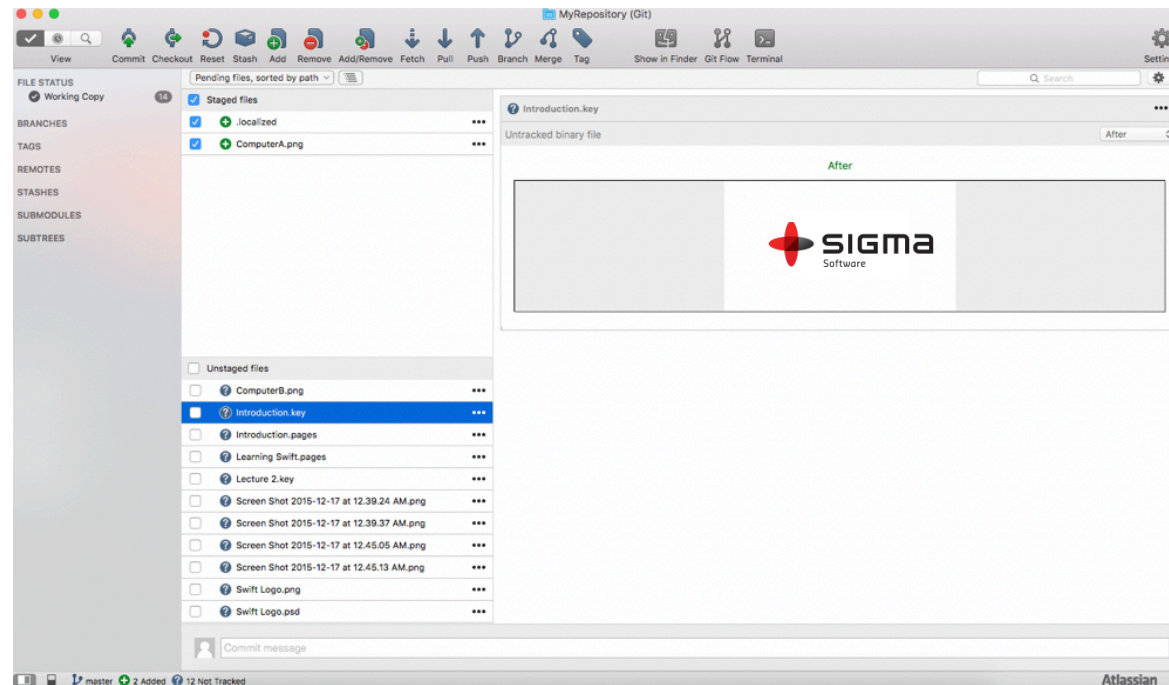
Git та SourceTree

Atlassian freeware cross-platform software



Git та SourceTree

Atlassian freeware cross-platform software



Список рекомендованих ресурсів до перегляду

Git Wiki - (<https://uk.wikipedia.org/wiki/Git>)

Git Install - (<https://www.atlassian.com/git/tutorials/install-git>)

Git merge tool - (<http://www.rosipov.com/blog/use-vimdiff-as-git-mergetool/>)

Git vs xCode - (<https://www.raywenderlich.com/153084/use-git-source-control-xcode-9>)

SourceTree - (<https://confluence.atlassian.com/bitbucket/tutorial-learn-sourcetree-with-bitbucket-cloud-760120235.html>)



UNIVERSITY