Systemy kontroli wersji – GIT

Łukasz Ogan
lukasz.ogan@gmail.com
http://lukaszogan.com

System kontroli wersji

Jest to oprogramowanie służące do śledzenia zmian głównie w kodzie źródłowym oraz pomocy programistom w łączeniu zmian dokonanych w plikach przez wiele osób w różnym czasie.

Systemy kontroli wersji umożliwiają (ang. **VCS**, **V**ersion **C**ontrol **S**ystem):

- śledzenie wszystkich zmian dokonywanych na pliku/plikach
- przywołanie dowolnej wcześniejszej wersji pliku/plików
- porównywanie wprowadzonych zmian
- łączenie zmian dokonanych na plikach

git

Atlassian Bitbucket

GIT

Git – rozproszony system kontroli wersji. Stworzył go Linus Torvalds jako narzędzie wspomagające rozwój jądra Linux. Git stanowi wolne oprogramowanie i został opublikowany na licencji GNU GPL w wersji 2.

Linus Torvalds – autor Linuxa

Linux - rodzina uniksopodobnych systemów operacyjnych opartych na jądrze Linux. Linux jest jednym z przykładów wolnego i otwartego oprogramowania (jego kod źródłowy może być dowolnie wykorzystywany, modyfikowany i rozpowszechniany)

https://www.kernel.org/

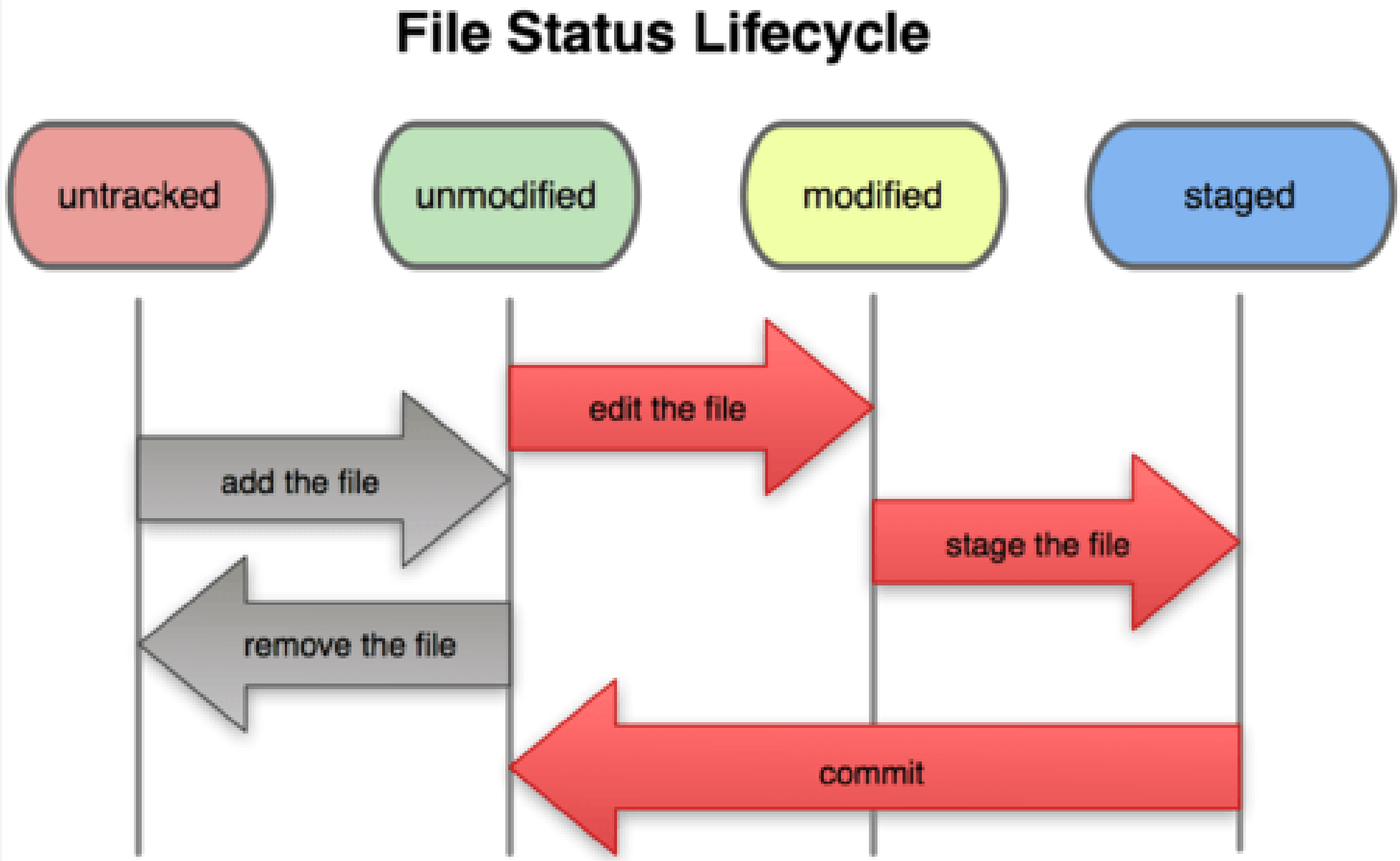
Zalety systemu GIT

Pozwala m.in. na:

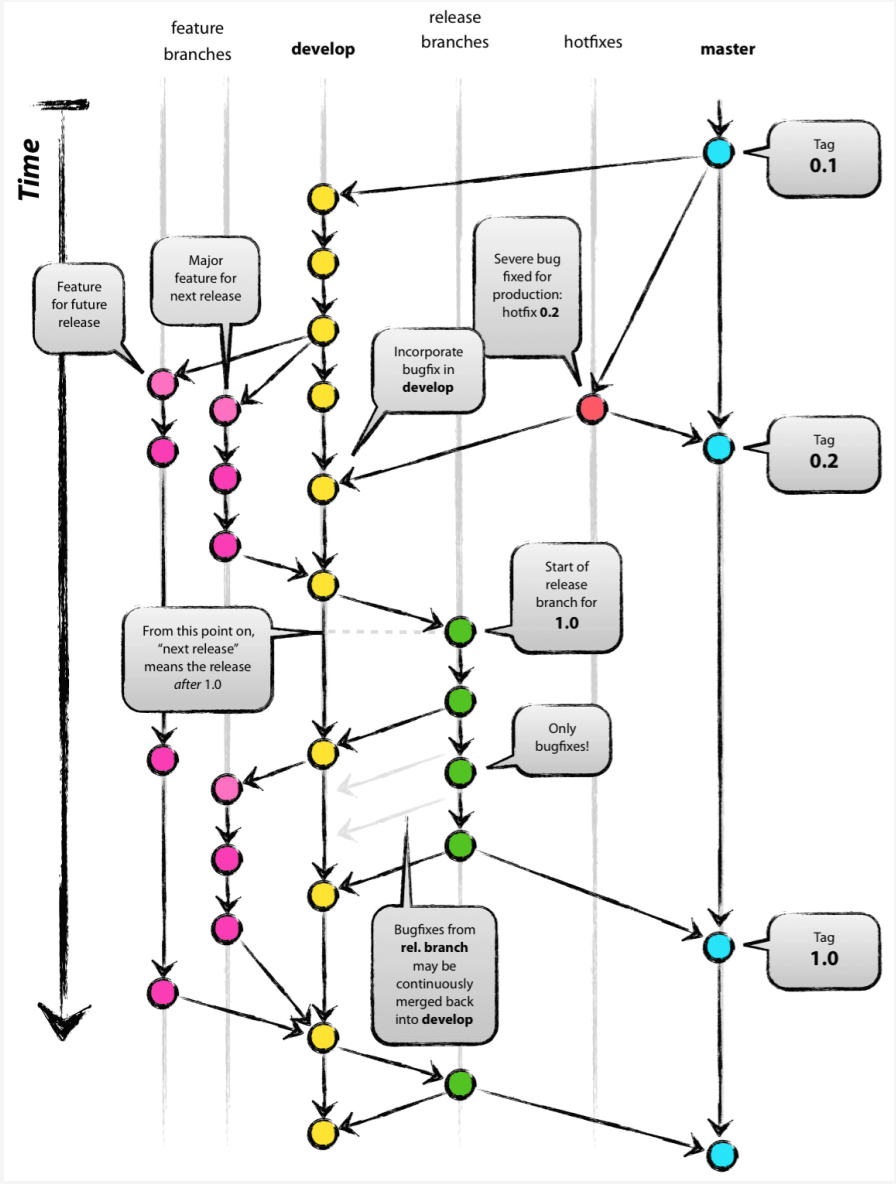
- śledzenie zmian w plikach
- łatwy powrót do wcześniejszej wersji pliku
- sprawną zdalną współpracę

Stany plików:

1. Nieśledzony
2. Śledzony
3. Zmodyfikowany
4. Zatwierdzony



GitFlow


feature branches
develop
release branches
hotfixes
master
Time
Tag 0.1
Major feature for next release
Feature for future release
Severe bug fixed for production: hotfix 0.2
Incorporate bugfix in develop
Tag 0.2
Start of release branch for 1.0
From this point on, "next release" means the release after 1.0
Only bugfixes!
Bugfixes from rel. branch may be continuously merged back into develop
Tag 1.0

Graficzne klienty Gita

- TortoiseGit
  - https://tortoisegit.org/
- GitHub
  - Windows: https://windows.github.com/
  - Mac: https://mac.github.com/
- SourceTree
  - https://www.sourcetreeapp.com/

Konfiguracja GITa

Do konfiguracji GITa służy komenda **git config**. Może ona przyjmować jedną z opcji:

- --local (konfiguracja dla kazdego repozytorium)
- --global (konfiguracja dla każdego użytkownika)
- --system(kondiguracja dla całego komputera)

Możliwe są również inne opcje, które zmieniają działanie komendy :

- --unset name
- --list

GIT polecenia

Inicjalizacja repozytorium

```
git init
```

Dodanie plików do repozytorium

```
git add .
```

Commit plików

```
git commit –m „add final files”
```

`git commit --amend –` napisuje ostatni commit (poprawa czegoś)

Dodanie adresu repozytorium

```
git remote add origin http://github.com/user/myrepo.git
```

Push plików

```
git push –u origin master
```

GIT gałęzie

`git branch` - **listuje** wszystkie lokalne gałęzie
`git checkout -b feature/1` - **utworzenie** i przejście do gałęzi **feature/1**
`git checkout develop` - **przejście** do gałęzi **develop**
`git branch -d feature/1` - **usunięcie** gałęzi **feature/1**

Git polecenia

Klonowanie istniejącego repozytorium

```
git clone http://github.com/user/repo.git
```

Sprawdzenie stanu plików

```
git status
```

Dodanie jednego pliku

```
git add README
```

Lista różnic

```
git diff
```

Nowy branch

```
git branch test
```

Przełączenie się na nowego brancha

```
git checkout test
```

Merge z masterem

```
git merge test
```

`git init`	Inicjalizacja repozytorium git
`git status`	Sprawdzanie aktualnego stanu repozytorium
`git remote add origin http://github.com/...`	To polecenie łączy nasze lokalne repozytorium z repozytorium stworzonym gdzieś na serwerze (np. na portalu Github - jak w naszym przypadku) o podanym adresie url.
`git config --global user.name "Imię nazwisko"`	Zmiana naszego imienia i nazwiska w ustawieniach Gita
`git config --global user.email "email@email.com"`	Zmiana naszego emaila w ustawieniach gita
`git push`	↑ - "Wypchnięcie" zakomitowanych zmian
`git push -u origin master`	↑ - To polecenie uruchamiamy przy pierwszym "wypchnięciu" naszych danych na zdalne repozytorium tak, aby git zapamiętał ustawienia odnośnie gałęzi (branch'a)
`git pull`	↓ - Pobranie zmian z repozytorium (ostatnio używana gałąź)

`git pull origin master`	↓ - Pobranie zmian ze zdalnego repozytorium z gałęzi master (jest to gałąź domyślna)
`git add *`	Dodanie wszystkich plików z danego katalogu do śledzenia zmian przez Gita
`git add plik1.txt`	Dodanie pliku plik1.txt do śledzenia zmian przez Gita
`git add plik1.txt plik2.txt`	Ta sama komenda co wyżej, tylko jednocześnie dodajemy 2 pliki
`git add '*.txt'`	A tutaj dodajemy wszystkie pliki z danego katalogu o rozszerzeniu .txt
`git commit -m "Drobne zmiany"`	Aby zapisać zmiany jakie dokonaliśmy uruchamiamy polecenie commit i podajemy w komunikacie informacje o tym co udało się zrobić

Cygwin

Jest to kolekcja narzędzi wolnego i otwartego oprogramowania, które udostępniają programom działającym pod systemem Windows funkcjonalność przypominającą system Linux, a także biblioteka dla systemu Windows standardu POSIX. Aplikacje używają głównie interfejsu tekstowego, ale dostępny jest również podsystem grafiki X.Org oraz graficzne środowiska GNOME i KDE. Projekt posiada wygodny w użyciu program instalacyjny.

Instalator: https://cygwin.com/install.html

Terminal – praktyczne polecenia

Podstawowe polecenia do wykonywania operacji na plikach i katalogach:
http://www.gabo.hi.pl/linux/polecenia.htm#II

ConEmu

Alternatywny klient terminala na system Windows. Pozwala między innymi tworzyć karty. https://conemu.github.io/