



GitHub, Bitbucket i git

czyli najlepsi pomocnicy programisty

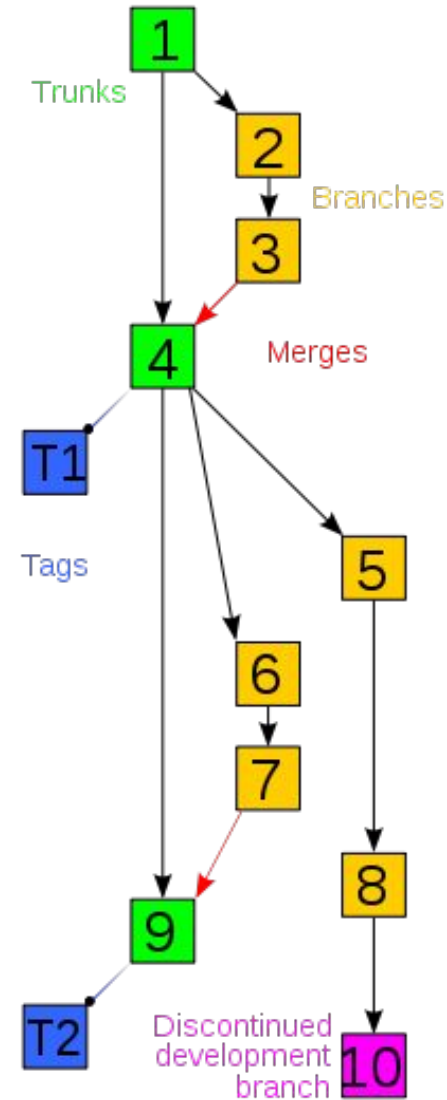


Łukasz Szeremeta 2017

System kontroli wersji

- ▶ oprogramowanie służące do śledzenia zmian głównie w kodzie źródłowym oraz pomocy programistom w łączeniu zmian dokonanych w plikach przez wiele osób w różnych momentach czasowych

source: Wikipedia under CC BY-SA 3.0 license



graphic credits: Traced by Stannered,
original by Sami Kerola derivative work:
Moxfyre derivative work: Echion2 under CC
BY-SA 3.0 license

I jest git

najpopularniejszy obecnie system kontroli wersji



Pierwotne założenia



Linus Torvalds - twórca gita
photo by Krd under CC BY-SA 4.0 license

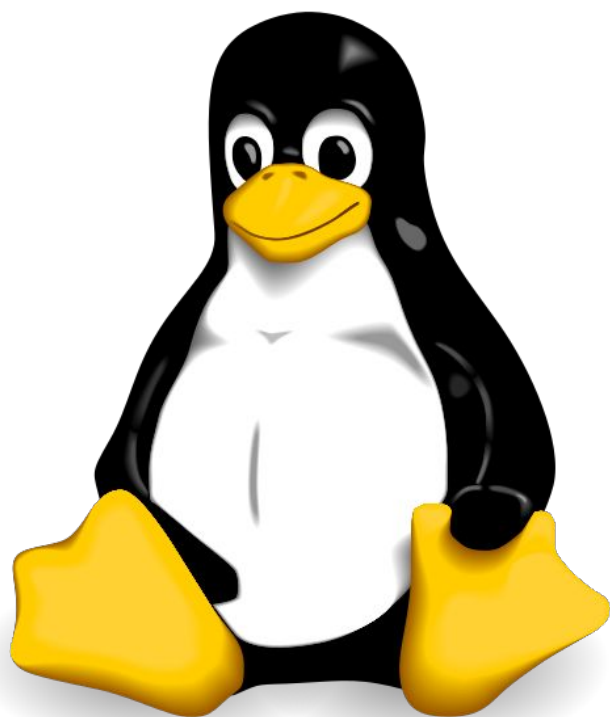
1. Wziąć przykład z CVS, czego nie robić
2. System powinien być rozproszony
3. System powinien być chroniony przed błędami w repozytorium (przypadkowymi, jak awaria twardego dysku, jak i złośliwymi, wprowadzonymi przez kogoś)
4. System powinien być szybki

source: Wikipedia under CC BY-SA 3.0 license

Główne cechy

- ▶ **Efektywna praca z dużymi projektami** - system Git według zapewnień Torvaldsa, a także według testów fundacji Mozilla, jest o rzędy wielkości szybszy niż niektóre konkurencyjne rozwiązania
- ▶ **Dobre wsparcie dla rozgałęzionego procesu tworzenia oprogramowania** - jest dostępnych kilka algorytmów łączenia zmian z dwóch gałęzi, a także można dodawać własne algorytmy
- ▶ **Praca off-line** - każdy programista posiada własną kopię repozytorium, do której może zapisywać zmiany bez połączenia z siecią; następnie zmiany mogą być wymieniane między lokalnymi repozytoriami
- ▶ **Wsparcie dla istniejących protokołów sieciowych** - dane można wymieniać przez HTTP(S), FTP, rsync, SSH
- ▶ **Każda rewizja to obraz całego projektu** - w przeciwieństwie do innych systemów kontroli wersji, Git nie zapamiętuje zmian między kolejnymi rewizjami, lecz kompletne obrazy. Z jednej strony wymaga to nieco więcej pracy aby porównać dwie rewizje, z drugiej jednak pozwala np. na automatyczną obsługę zmian nazw plików.

Kto używa Gita?



reddit

digg



Microsoft

Podstawy gita

Szybkie i efektywne komendy



GIT INIT? GIT CLONE?



source: memecrunch.com

Najprzydatniejsze komendy

git init

- ▶ Inicjacja repozytorium w bieżącym katalogu

git add -A

- ▶ Dodanie wszystkich ścieżek do monitorowania oraz sprawdzenie stanu plików już modyfikowanych

git clone ADRES

- ▶ Sklonowanie zdalnego repozytorium do nowego katalogu

Najprzydatniejsze komendy

git status

- ▶ Wyświetlenie statusu repozytorium
 - ▶ lista zmienionych plików
 - ▶ pliki, które zostały zmienione, ale nie są śledzone
 - ▶ inne...

git push

- ▶ Wysłanie zmian do zdalnego repozytorium

git commit -m "my commit"

- ▶ Utworzenie commita wraz z opisem wprowadzonych zmian

git log

- ▶ Wyświetla listę zmian (commitów)

Zarządzanie gałęziami

`git branch`

- ▶ Wyświetlanie listy wszystkich dostępnych gałęzi

`git branch nazwaGalezi`

- ▶ Utworzenie nowej gałęzi

`git checkout nazwaGalezi`

- ▶ Przełączenie się do innej gałęzi

`git merge testowy_branch`

- ▶ Scalanie zmian między gałęziami

Tagi

`git tag -a v1.1 -m 'opis'`

- ▶ Dodanie etykiety z opisem

`git tag v1.1`

- ▶ Dodanie etykiety bez opisu

Inne przydatne komendy

`git remote add origin ADRES`

- ▶ Dodanie zdalnego repozytorium

`gitk`

- ▶ Graficzne przeglądanie zmian



Mały pojedynek

GitHub vs BitBucket



GitHub

Mocne strony

- ▶ Społecznościowy, wyluzowany charakter
- ▶ Duża liczba użytkowników i projektów
- ▶ Barwny, przemyślany interfejs
- ▶ Interesujące dodatki (np. hosting statycznych stron internetowych)
- ▶ Program dla studentów

Słabe strony

- ▶ Płatne prywatne repozytoria
- ▶ Brak głosowania w systemie zgłaszania błędów

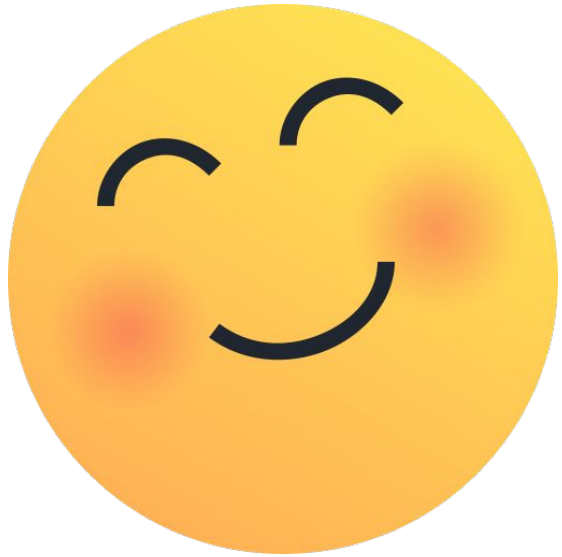
BitBucket

Mocne strony

- ▶ Bezpłatne prywatne repozytoria (w przypadku maksymalnie 5-osobowego zespołu)
- ▶ Oprócz gita obsługa Mercurial
- ▶ Większość najważniejszych możliwości GitHuba obecna

Słabe strony

- ▶ Stonowany, mniej “wystrzałowy” interfejs
- ▶ Brak możliwości przeszukiwania kodu źródłowego projektu wprost z serwisu
- ▶ Brak graficznych statystyk zmian w kodzie czy aktywności
- ▶ Mniejsza liczba możliwości niż w GitHubie



Czas na gita!

Podstawowa konfiguracja gita

```
git config --global color.ui true
```

```
git config --global user.name "Adam Nowak"
```

```
git config --global user.email "a.nowak@ex.com"
```

```
git config --global credential.helper cache
```

<http://bit.ly/git-conf>