

# Rozproszony system kontroli wersji

## GIT

Piotr Macuk <piotr@macuk.pl>

# Parę słów o mnie

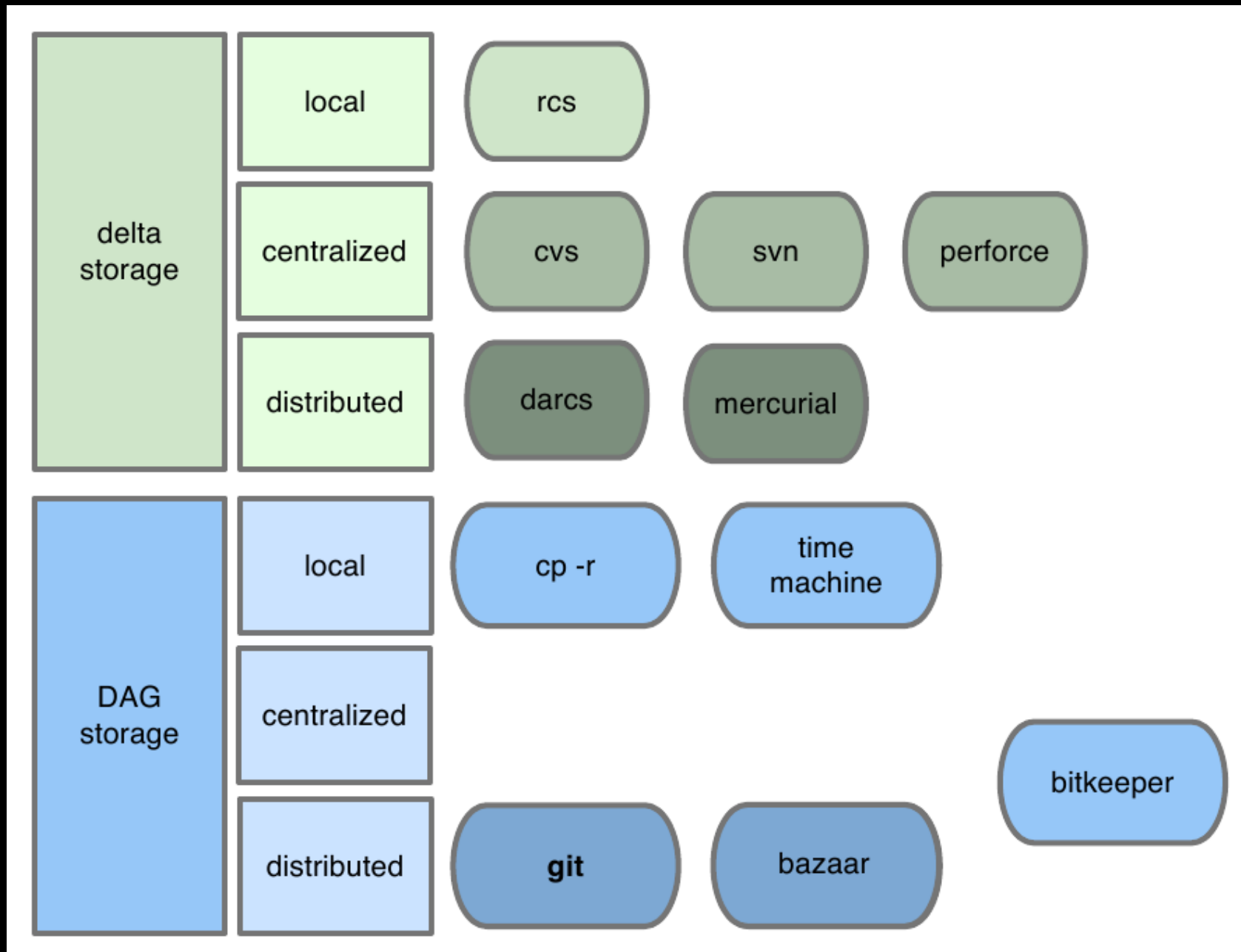
- Programowanie 19 lat
- Linux + vim 12 lat
- Kontrola wersji 9 lat
- Ruby (on Rails) 5 lat
- Git 2 lata

# Co to jest system kontroli wersji?

System kontroli wersji (ang. version/revision control system) służy do śledzenia zmian głównie w kodzie źródłowym oraz pomocy programistom w łączeniu i modyfikacji zmian dokonanych przez wiele osób w różnych momentach.

*Wikipedia*

# Systemy kontroli wersji



# Centralne vs rozproszone

## Centralne

- Single Point of Failure
- Blokady
- Koniecznie on-line

## Rozproszone

- Wszystko lokalnie (prawie)
- Szybkość
- Każde repozytorium to pełen backup

# GIT – skąd się wziął?

- Linux kernel (GPL)
- 1991–2005 BitKeeper (\$\$\$)
- 2005 zmiana licencjonowania i koniec współpracy
- Linus daje sobie 2 tygodnie

# Cechy / założenia GIT-a

- Szybkość!
- Prostota
- Łatwość rozgałęzienia i łączenia kodu
- Rozproszenie pracy
- Praca off-line
- Integralność danych
- Obsługa dużych projektów

# Rewelacyjny wynik

- 3 kwietnia start prac
- 7 kwietnia kontrola wersji własnego kodu
- 18 kwietnia łączenie kilku gałęzi kodu
- 27 kwietnia test szybkości (6.7 łat na sekundę)
- 16 czerwca produkcyjny hosting jądra (Linux 2.6.12)



# GIT - kto go używa?

- Linux Kernel
- Perl
- Gnome
- Qt
- Ruby on Rails
- Apache
- Android
- Wine
- Fedora
- Debian
- X.org
- XFCE

# Terminologia

## repozytorium

baza danych z pełną historią projektu

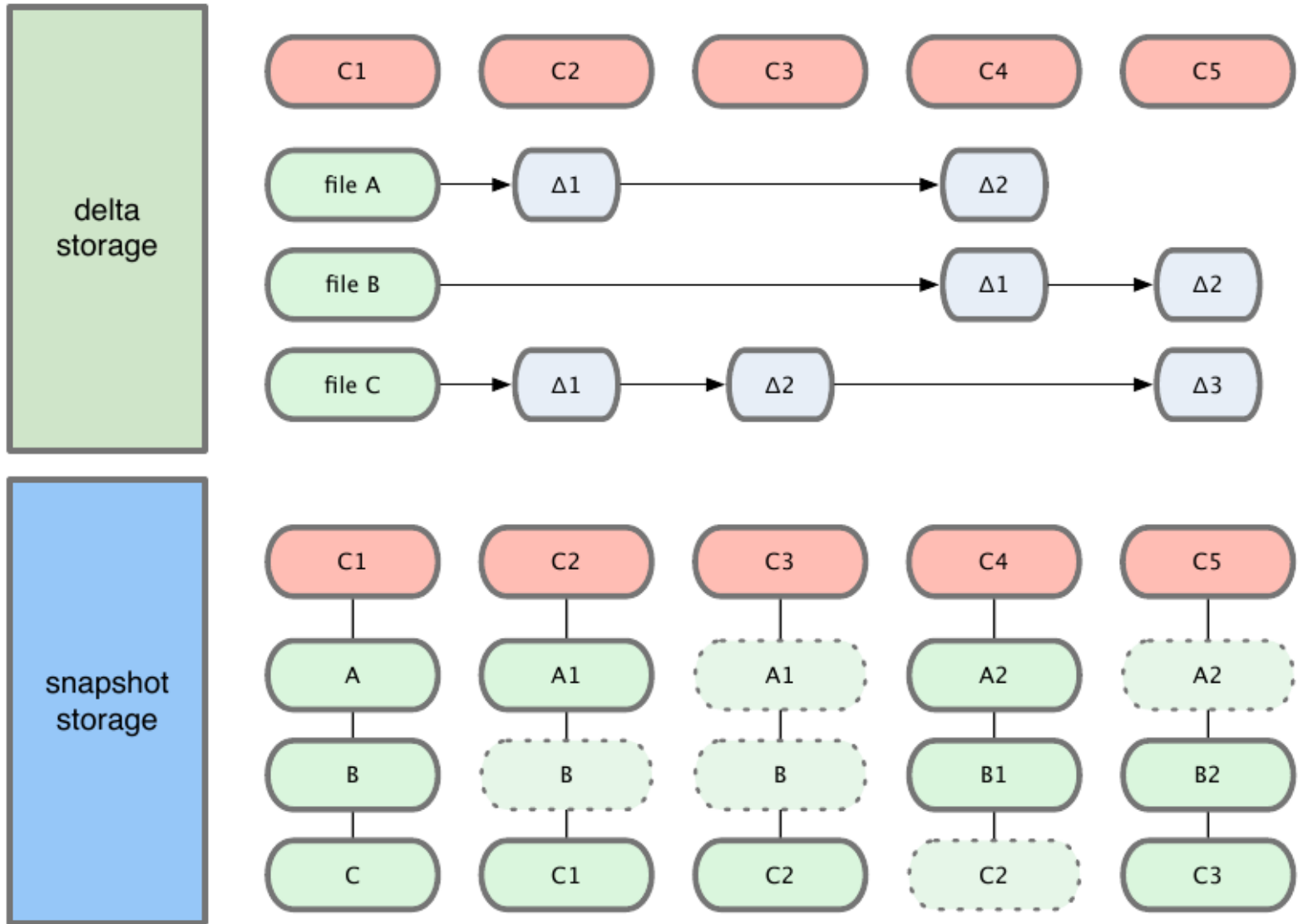
## kopia robocza

płiki we wskazanym momencie w czasie

## commit

przeniesienie zmian z kopii roboczej do repozytorium

# Migawki a nie różnice



# Praca off-line

- Różnice
- Historia
- Commit zmian
- Rozgałęzianie + łączenie
- Przemieszczanie się w czasie
- Przełączanie gałęzi

# Instalacja i konfiguracja

- /etc/gitconfig
- ~/.gitconfig
- .git/config
- git help config

```
$ git config --global user.name "Piotr Macuk"
```

```
$ git config --global user.email piotr@macuk.pl
```

# Szybki start

```
$ rails app
```

```
$ cd app
```

```
$ git init
```

```
$ git add .
```

```
$ git commit -m 'Start'
```

# Szybki start - klon

```
$ git clone git://github.com/git/hello.git
```

```
$ cd hello
```

```
$ vim / emacs / ...
```

```
$ git add (files)
```

```
$ git commit -m 'Zmiana koloru tła'
```

```
$ git format-patch origin/master
```

# Cykl pracy

## Standardowo

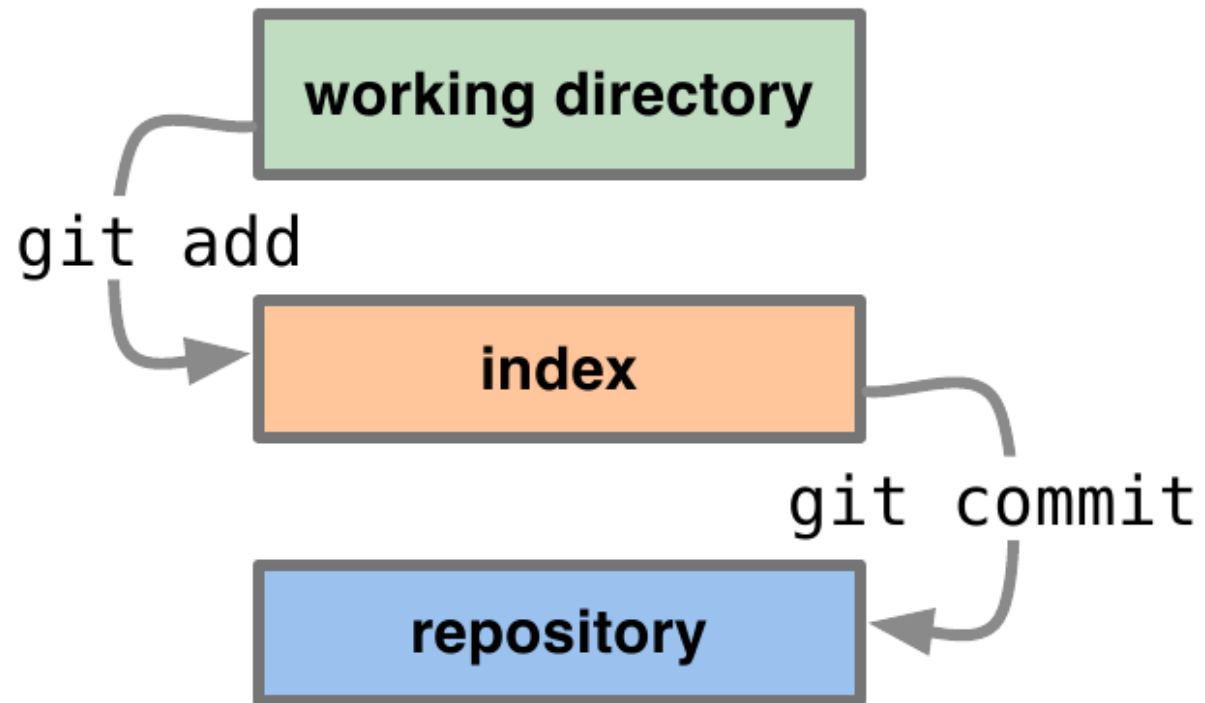
- vim / emacs / ...
- git add
- git status, git diff
- git commit

## Prościej:

- vim / emacs / ...
- git commit -a



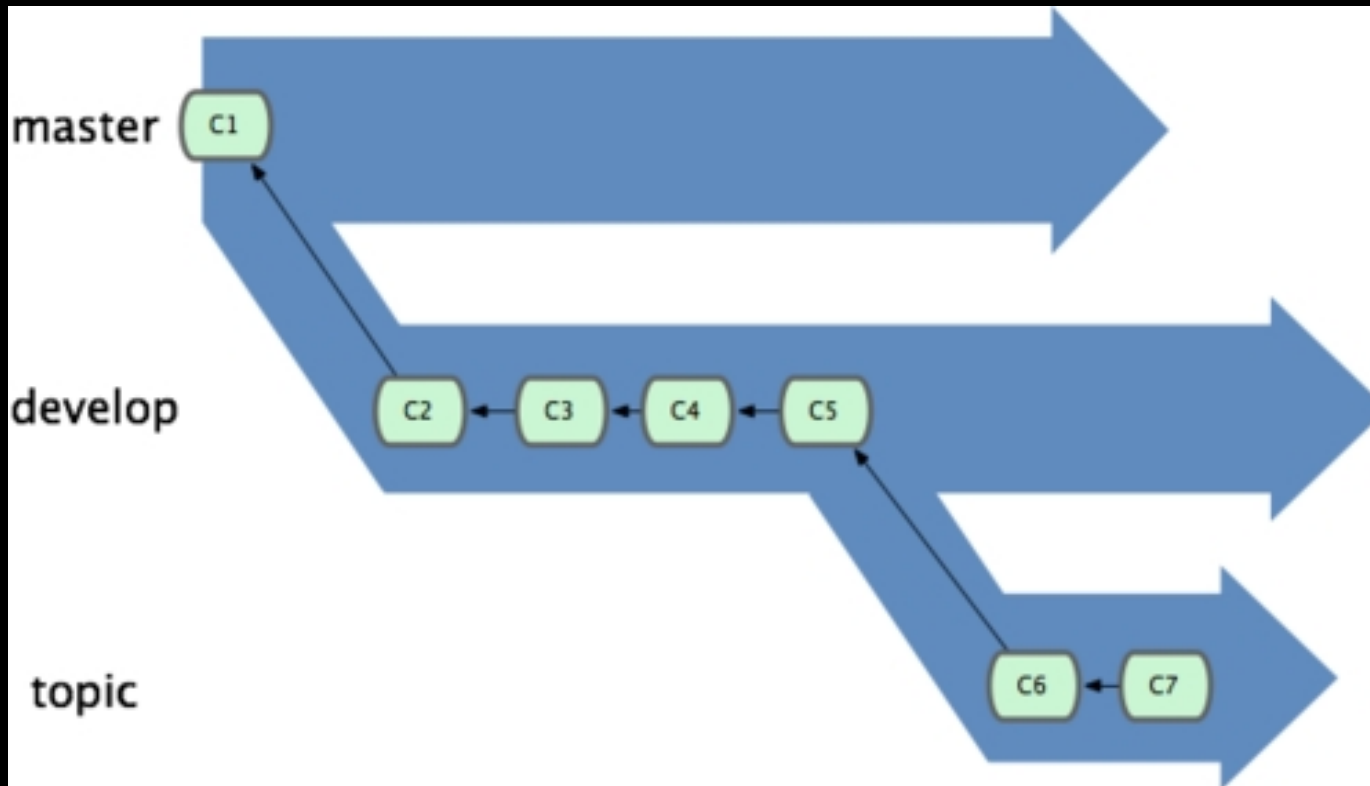
# Cykl pracy



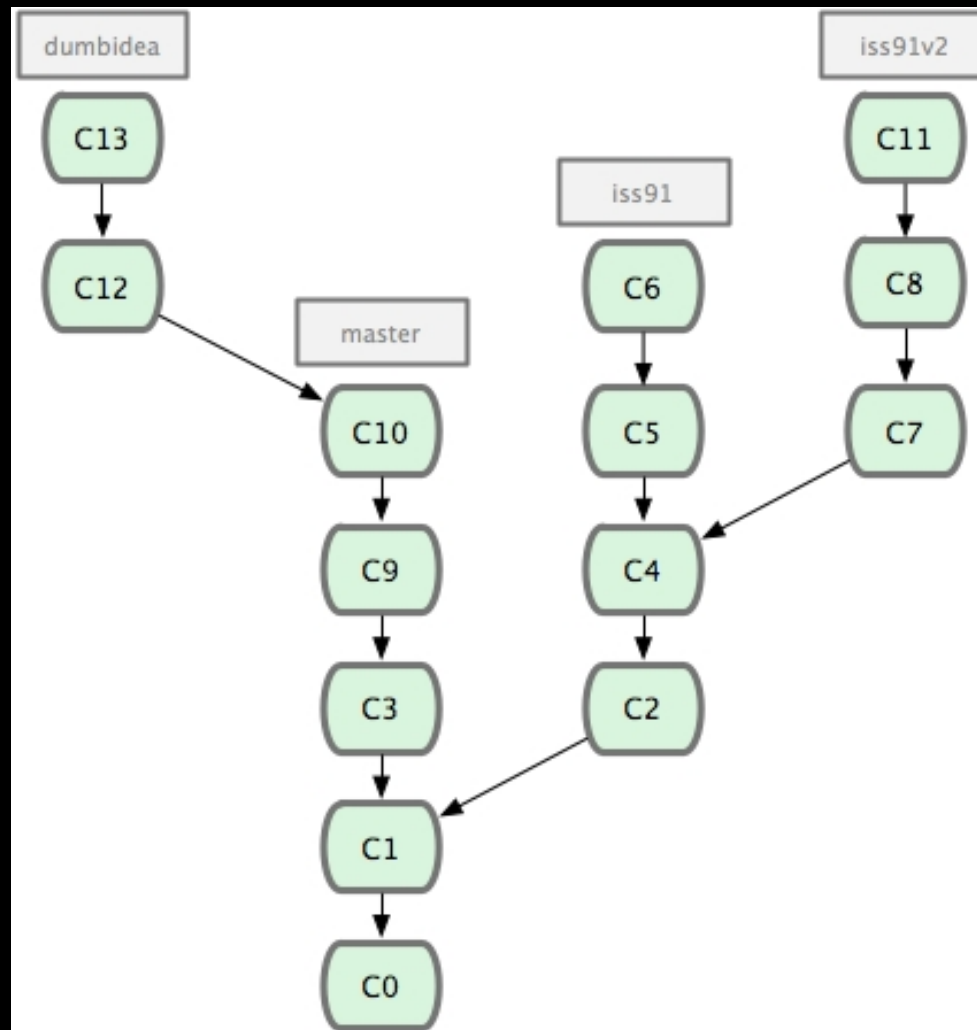
# Rozgałęzienia

- Lekkie, łatwe i przyjemne
- Do eksperymentów
- Do pracy nad kolejnymi funkcjami
- Do poprawiania błędów

# Rozgałęzienia



# Rozgałęzienia



Demo

# Narzędzia dodatkowe

- gitk (tcl/tk)
- tig (console)
- qgit (Qt)
- TortoiseGit (Windows)
- GitX (Mac OS X)
- git svn (import, proxy)

# Linki

1. YouTube: Linus Torvalds on git
2. <http://git-scm.com>
3. <http://gitcasts.com>
4. <http://progit.org>
5. <http://learn.github.com>

# GIT – podsumowanie

- Szybki, efektywny
- Prosty
- Łatwy w rozgałęzianiu i łączeniu kodu
- Rozproszony
- Integralny
- Open source (GPL)



Pytania?

Dziękuję za uwagę :)

Obrazki :

<http://github.com/schacon/git-presentations>

<http://progit.org>