

# No i Git!

## część I

Marcin TORGiren Fabrykowski

AGH - University of Science and Technology

29 listopada 2012

# System kontroli wersji

główne zadanie

kontrola i zachowywanie wszystkich wersji kontrolowanych plików

# System kontroli wersji

## główne zadanie

kontrola i zachowywanie wszystkich wersji kontrolowanych plików

## zadania dodatkowe

# System kontroli wersji

## główne zadanie

kontrola i zachowywanie wszystkich wersji kontrolowanych plików

## zadania dodatkowe

- wielu użytkowników

# System kontroli wersji

## główne zadanie

kontrola i zachowywanie wszystkich wersji kontrolowanych plików

## zadania dodatkowe

- wielu użytkowników
- rozwijanie w różnych kierunkach

# System kontroli wersji

## główne zadanie

kontrola i zachowywanie wszystkich wersji kontrolowanych plików

## zadania dodatkowe

- wielu użytkowników
- rozwijanie w różnych kierunkach
- backup

# Podział

lokacja

# Podział

## lokacja

- scentralizowane



# Podział

## lokacja

- scentralizowane
- rozproszone

# Podział

## lokacja

- scentralizowane
- rozproszone

## przechowywanie wersji

# Podział

## lokacja

- scentralizowane
- rozproszone

## przechowywanie wersji

- jako zmiany

# Podział

## lokacja

- scentralizowane
- rozproszone

## przechowywanie wersji

- jako zmiany
- jako snapshoty

# Pierwszy commit

klonujemy repozytorium

```
git clone git@192.168.246.133:first.git
```

# Pierwszy commit

klonujemy repozytorium

```
git clone git@192.168.246.133:first.git
```

praca nad projektem

```
vim plik
```

# Pierwszy commit

klonujemy repozytorium

```
git clone git@192.168.246.133:first.git
```

praca nad projektem

```
vim plik
```

dodanie pliku

```
git add plik
```

# Pierwszy commit

klonujemy repozytorium

```
git clone git@192.168.246.133:first.git
```

praca nad projektem

```
vim plik
```

dodanie pliku

```
git add plik
```

commit

```
git commit -m "tresc commita"
```



# Pierwszy commit

klonujemy repozytorium

```
git clone git@192.168.246.133:first.git
```

praca nad projektem

```
vim plik
```

dodanie pliku

```
git add plik
```

commit

```
git commit -m "tresc commita"
```

wysalanie

```
git push origin master
```

# Uproszczony schemat pracy

pobranie danych z serwera

git fetch

# Uproszczony schemat pracy

pobranie danych z serwera

git fetch

mergowanie zmian

git merge origin/master

# Uproszczony schemat pracy

pobranie danych z serwera

```
git fetch
```

mergowanie zmian

```
git merge origin/master
```

praca nad plikiem

```
vim plik
```

```
git add plik
```

```
git commit -m "opis"
```

```
git push origin
```

# Podstawowa konfiguracja

przedstawiamy się

```
git config --global user.name "Jan Kowalski"  
git config --global user.email "moj@email.pl"
```

# Podstawowa konfiguracja

przedstawiamy się

```
git config --global user.name "Jan Kowalski"  
git config --global user.email "moj@email.pl"
```

cenimy sobie komfort pracy

```
git config --global core.editor vim
```

# Dwoimy się i troimy

tworzymy gałąź

```
git branch galaz1
```

# Dwoimy się i troimy

tworzymy gałąź

```
git branch galaz1
```

przełączamy się

```
git checkout galaz1
```



# Dwoimy się i troimy

tworzymy gałąź

```
git branch galaz1
```

przełączamy się

```
git checkout galaz1
```

normalna praca

```
vim plik
```

```
git add plik
```

```
git commit -m "opis"
```

# Czas na magię...

przechodzimy na gałąź główną

```
git checkout master
```

# Czas na magię...

przechodzimy na gałąź główną

```
git checkout master
```

owoc naszych prac...

```
cat plik
```

# Czas na magię...

przechodzimy na gałąź główną

```
git checkout master
```

owoc naszych prac...

```
cat plik
```

wracamy na galaz1

```
git checkout galaz1
```

```
cat plik
```

# Co człowiek rozłączył, git niech złączy...

przechodzimy na gałąź do której chcemy dołączyć  
git checkout master

# Co człowiek rozłączył, git niech złączy...

przechodzimy na gałąź do której chcemy dołączyć

```
git checkout master
```

gałęzie... łączcie się

```
git merge galaz1
```

# Co człowiek rozłączył, git niech złączy...

przechodzimy na gałąź do której chcemy dołączyć

```
git checkout master
```

gałęzie... łączcie się

```
git merge galaz1
```

i oto efekt

cat plik

# Mergowanie

Fast forward vs no-fast forward



# Gałęzie zdalne

“importowanie” gałęzi zdalnych

```
git checkout -t origin/galaz
```

# Gałęzie zdalne

"importowanie" gałęzi zdalnych

```
git checkout -t origin/galaz
```

wysyłanie zmian do gałęzi zdalnej

```
git push origin galaz:galaz
```

# Gałęzie zdalne

"importowanie" gałęzi zdalnych

```
git checkout -t origin/galaz
```

wysyłanie zmian do gałęzi zdalnej

```
git push origin galaz:galaz
```

usuwanie gałęzi zdalnej

```
git push origin :galaz
```

# Tematy

- git remote

# Tematy

- git remote
- git archive