

System kontroli wersji git

Adam Prochownik

13 października 2016



KoPiAR

KOLEGIUM NAUKOWE ROBOTYKÓW

Plan prezentacji

- 1 **Wstęp**
 - Materiały
 - Po co kontrola wersji ?
 - Dlaczego git?
- 2 **Podstawowe operacje**
 - Konfiguracja
 - Tworzymy lokalne repozytoria
 - Zatwierdzamy zmiany
- 3 **Praca grupowa**
 - Gałęzie
- 4 **Podsumowanie**
 - Gdzie szukać pomocy/więcej?



KoPaR

KOLEGIUM NAUKOWE ROBOTYKÓW

Wi-fi

- Nazwa sieci: KoNaR
- Hasło: WarsztatyRobotyczne



KoNaR

KOLEGIUM NAUKOWE ROBOTYKÓW

Materiały wykorzystane w tej prezentacji

- Prezentacja Błażeja Kowalczyka z 2014
- "Git. Rozproszony system kontroli wersji" Włodzimierz Gajda
- <https://git-scm.com/book/pl/v1>



KoPaR

KOLEGIUM NAUKOWE ROBOTYKÓW

Kto tak miał?



Warcaby



Warcaby_05_06_14



Warcaby_v2_09_06_
14_10_09



Warcaby_v2_11_06_
14_06_50



Warcaby_v2_
100614_0958



Warcaby_v2_
100614_1950



Warcaby_wersja
oddana



Kompilująca się
wersja_10_06_01_
25.rar



KoNaR

KOLEGIUM NAUKOWE ROBOTYKÓW

Kilka innych powodów

- Historia (cofanie) zmian



KoNaR

KOLEGIUM NAUKOWE ROBOTYKÓW

Kilka innych powodów

- Historia (cofanie) zmian
- Kopia zapasowa



KoPiA R

KOŁO NAUKOWE ROBOTYKÓW

Kilka innych powodów

- Historia (cofanie) zmian
- Kopia zapasowa
- Współpraca



KOnAR

KOLEGIUM NAUKOWE ROBOTYKÓW

Kilka innych powodów

- Historia (cofanie) zmian
- Kopia zapasowa
- Współpraca
- Wersje robocze



KOPAR

KOLEGIUM NAUKOWE ROBOTYKÓW

Dlaczego git?

- Nie wymaga serwera do działania



KoPiA R

KOŁO NAUKOWE ROBOTYKÓW

Dlaczego git?

- Nie wymaga serwera do działania
- Błyskawiczne operacje na historii i gałęziach



KOnAR

KOLEGIUM NAUKOWE ROBOTYKÓW

Dlaczego git?

- Nie wymaga serwera do działania
- Błyskawiczne operacje na historii i gałęziach
- Każdy posiada pełną kopie zapasową (łącznie z historią)



KoNaR

KOLEGIUM NAUKOWE ROBOTYKÓW

Dlaczego git?

- Nie wymaga serwera do działania
- Błyskawiczne operacje na historii i gałęziach
- Każdy posiada pełną kopie zapasową (łącznie z historią)
- Rozdzielenie kontrolowania zmian i dzielenia się nimi



KoNaR

KOLEGIUM NAUKOWE ROBOTYKÓW

Podstawowa konfiguracja

- Dane użytkownika

```
git config --global user.name "Jan Kowalski"  
git config --global user.email jan.kowalski@gmail.com
```

- Edytor

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

- Wyświetlenie konfiguracji

```
git config -l
```



KOnPaR

KOLEGIUM NAUKOWE ROBOTYKÓW

Inicjalizacja repozytorium

Wydajemy polecenia

```
mkdir repo  
cd repo  
git init
```

- Stworzony został folder `.git`, który przechowuje bazę z całą historią repozytorium
- Skopiowanie całej zawartości folderu `repo` spowoduje skopiowanie również całej historii



KOnAR

KOLEGIUM NAUKOWE ROBOTYKÓW

Klonujemy zdalne repozytorium

- Znajdujemy adres naszego repozytorium

Wydajemy polecenie

```
git clone  
nazwaUzytkownika@protkol://nazwaSerwera:port/sciezkaDoRepoztorium
```

Na przykład

```
git clone git@bitbucket.org:konar/warsztaty-robotyczne.git
```



KoNaR

KOLEO NAUKOWE ROBOTYKÓW

Dodajemy nowy plik

Wydajemy polecenia

```
echo "Zmiana nr 1" » plik1.txt  
git status
```

- Powstał plik plik1.txt, który jest nie jest jeszcze śledzony przez gita

Wydajemy polecenia

```
git add plik1.txt  
git status
```

Nasz plik jest teraz śledzony przez gita



KoPaR

KOLEGIUM NAUKOWE ROBOTYKÓW

Zatwierdzamy zmiany

- Pliki które są śledzone zostaną włączone do następnego commita

Wydajemy polecenie

```
git commit
```

- Zostanie uruchomiony nasz edytor w celu napisania komentarza, albo

Wydajemy polecenie

```
git commit -m "Komentarz do commita"
```



KOnAR
KOLEGIUM NAUKOWE ROBOTYKÓW

Co zrobić jak mamy 100 plików w repo?

Stwórzmy kilka plików

```
echo "Tekst 2" » plik2.txt
```

```
echo "Tekst 3" » plik3.txt
```

```
echo "Tekst 4" » plik4.txt
```

Zobaczmy co na to git:

```
git status
```

Teraz możemy dodać wszystkie wszystkie pliki do śledzenia:

```
git add -A
```



KoPaR

KOLEGIUM NAUKOWE ROBOTYKÓW

Przeglądamy historie commit'ów

Wydajemy polecenie

```
git log
```

- Ujrzymy listę ostatnich commitów

Wydajemy polecenie

```
git log --name-status
```

- Wyświetlona zostanie lista commitów razem z listą plików zmodyfikowanych



KOnAR

KOLEGIUM NAUKOWE ROBOTYKÓW

.gitignore

Jeżeli nie chcemy aby pewne pliki były widziane przez git'a możemy wymienić je w pliku .gitignore (np. pliki od środowiska programistycznego, wyniki kompilacji, pliki tymczasowe)

- Plik jest parsowany z góry do dołu (reguły z dołu nadpisują te z góry)
- # oznacza komentarz
- ! to wyjątek od poprzedniej reguły



KoPaR

KOŁO NAUKOWE ROBOTYKÓW

Przykładowy plik .gitignore

```
# żadnych plików .a
*.a
# ale uwzględniaj lib.a
!lib.a
# ignoruj plik TODO, ale nie podkatalog /TODO
/TODO
# ignoruj wszystkie pliki znajdujące się w katalogu build/
build/
```



KOnAR
KOLEGIUM NAUKOWE ROBOTYKÓW

Wysyłamy zmiany na serwer

- Nasze dotychczasowe commity trafiły tylko naszej lokalnej wersji repozytorium



KoNaR

KOLEGIUM NAUKOWE ROBOTYKÓW

Wysyłamy zmiany na serwer

- Nasze dotychczasowe commity trafiły tylko naszej lokalnej wersji repozytorium
- Teraz musimy umieścić je na serwerze



KOnAR

KOLEGIUM NAUKOWE ROBOTYKÓW

Wysyłamy zmiany na serwer

- Nasze dotychczasowe commity trafiły tylko naszej lokalnej wersji repozytorium
- Teraz musimy umieścić je na serwerze

Wydajemy polecenie

```
git push origin master
```



KOnAR

KOLEGIUM NAUKOWE ROBOTYKÓW

Pobieramy zmiany z serwera

- Gdy ktoś inny dokona zmiany na zdalnym repozytorium, musimy takie zmiany pobrać



KONAR

KOŁO NAUKOWE ROBOTYKÓW

Pobieramy zmiany z serwera

- Gdy ktoś inny dokona zmiany na zdalnym repozytorium, musimy takie zmiany pobrać

Wydajemy polecenie

```
git pull origin master
```



KOPAR
KOLEGIUM NAUKOWE ROBOTYKÓW

Cofanie commitów

Wydajemy polecenie

```
git reset NrCommitta
```

Aby wycofać zmiany do konkretnego *NrCommitta* (znikają z naszej historii)

Wydajemy polecenie

```
git revert NrCommitta
```

Aby stworzyć commit “naprostowujący” nasze modyfikacje



KoPaR

KOLEGIUM NAUKOWE ROBOTYKÓW

Cofanie nie zacommitowanych zmian

Wydajemy polecenie

```
git reset --hard HEAD
```

lub

Wydajemy polecenie

```
git checkout -f nazwaNaszegoBrancha
```

Aby stworzyć commit “naprostowujący” nasze modyfikacje



KoPaR

KOLEGIUM NAUKOWE ROBOTYKÓW

Co to są gałęzie?

- Duplikacja aktualnej wersji



KoPaR

KOLEGIUM NAUKOWE ROBOTYKÓW

Co to są gałęzie?

- Duplikacja aktualnej wersji
- Zmiany mogą być wprowadzane równoległe



KoPaR

KOLEGIUM NAUKOWE ROBOTYKÓW

Co to są gałęzie?

- Duplikacja aktualnej wersji
- Zmiany mogą być wprowadzane równoległe (również przez kilka osób naraz)



KoPaR

KOLEGIUM NAUKOWE ROBOTYKÓW

Co to są gałęzie?

- Duplikacja aktualnej wersji
- Zmiany mogą być wprowadzane równoległe (również przez kilka osób naraz)
- Zmiany można łączyć



KoPnR

KOLEGIUM NAUKOWE ROBOTYKÓW

Co to są gałęzie?

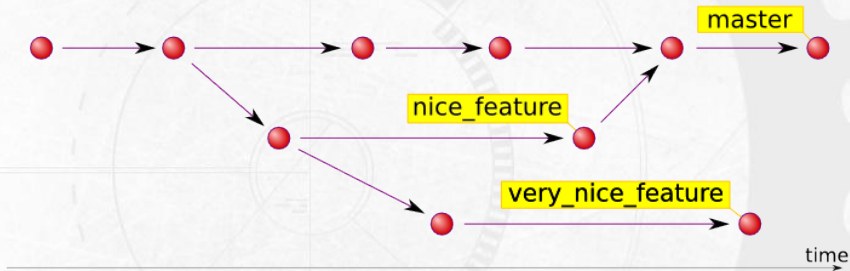
- Duplikacja aktualnej wersji
- Zmiany mogą być wprowadzane równoległe (również przez kilka osób naraz)
- Zmiany można łączyć (merge)



KoPnR

KOLEGIUM NAUKOWE ROBOTYKÓW

Odpowiedni obrazek

**KOnAR**

KOLEGIUM NAUKOWE ROBOTYKÓW

Kiedy robić branże?

● **Zawsze!**



KoPaR

KOLEGIUM NAUKOWE ROBOTYKÓW

Kiedy robić branche?

- **Zawsze!**
- Tak naprawdę, to najczęściej robi się gałąź per funkcjonalność



KoPaR

KOLEGIUM NAUKOWE ROBOTYKÓW

Jak to zrobić w gitcie

Wydajemy polecenie

```
git branch -a
```

Ujrzymy listę wszystkich branchy (łącznie z zdalnymi repozytoriami)

Wydajemy polecenie

```
git branch NazwaNowegoBrancha
```

Stworzymy nowego brancha



KoPiAR

KOLEGIUM NAUKOWE ROBOTYKÓW

Wracamy ze zmianami do głównej gałęzi

Wydajemy polecenie

```
git merge StaryBranch
```

Zmiany, z *StaryBranch* zostaną dołączone do aktualnej gałęzi



KoNaR

KOLEGIUM NAUKOWE ROBOTYKÓW

• <https://git-scm.com/book/pl/v1>



KoPaR

KOLEGIUM NAUKOWE ROBOTYKÓW

- <https://git-scm.com/book/pl/v1>
- <http://stackoverflow.com/>



KoNaR

KOLEGIUM NAUKOWE ROBOTYKÓW

- <https://git-scm.com/book/pl/v1>
- <http://stackoverflow.com/>

połączenie

```
git cośćam -help
```



KoNaR

KOLEGIUM NAUKOWE ROBOTYKÓW