



Git - kontrola wersji projektu

Gniewomir Łuczak
Program Developer, IAI S.A.

Kontakt: gluczak@iai-sa.com





Wersja

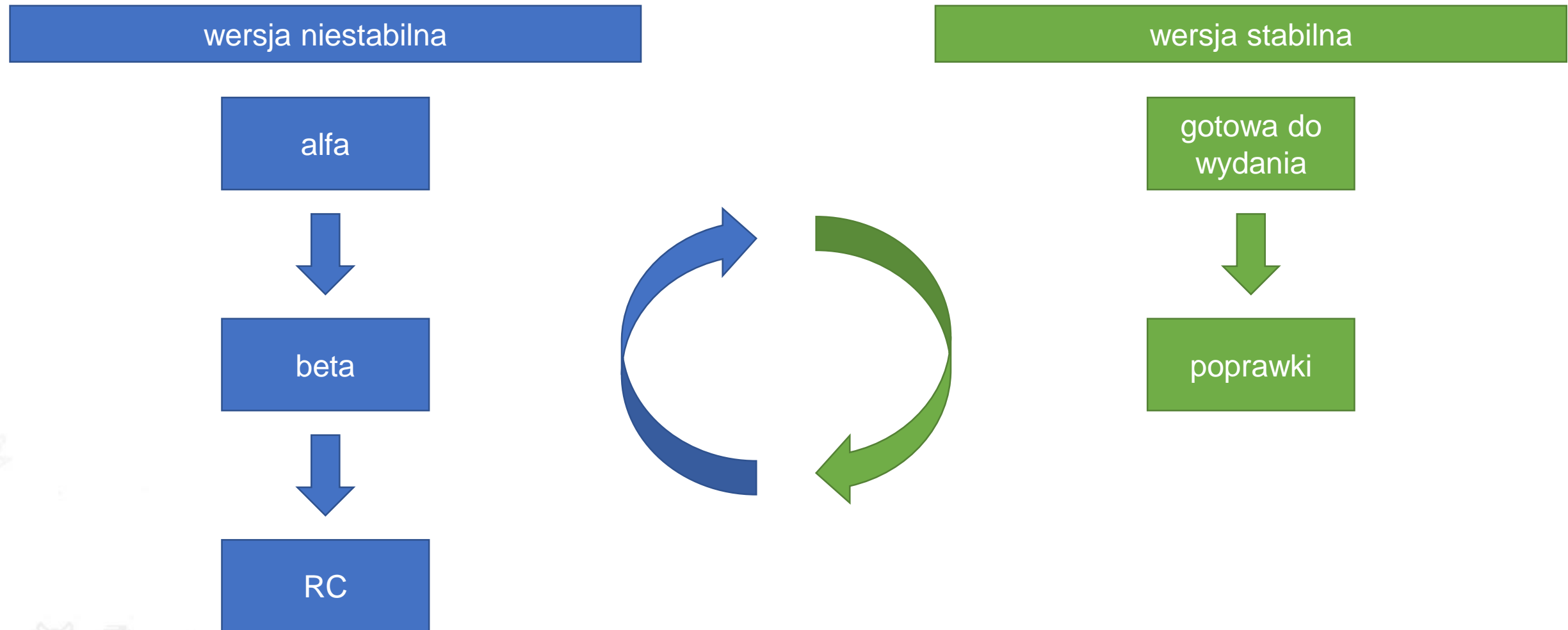
„jedna z kilku odmian jakiegoś produktu”

<https://sjp.pwn.pl/szukaj/wersja.html>





Cykl życia programu





WHEN YOU HEAR THIS:



*YOU KNOW YOU'RE IN A
SOFTWARE PROJECT*



<https://git-scm.com>





Linus Torvalds, 2015

Git

Rozproszony system kontroli wersji

Wydany w 2005 roku, stworzony przez Linusa Torvaldsa. Opublikowany na licencji GNU GPL.
Pierwotnie stworzony jako narzędzie wspomagające rozwój jądra Linuksa.

Główne założenia:

1. Wziąć przykład z CVS, czego **nie** robić.
2. System powinien być rozproszony.
3. System powinien być chroniony przed błędami w repozytorium (przypadkowe lub zamierzone).
4. System powinien być szybki.



Git

Główne cechy

- Wsparcie dla rozgałęzionego procesu tworzenia oprogramowania;
- Praca lokalna (off-line);
- Wsparcie dla protokołów sieciowych takich jak HTTP(S), FTP, SSH, rsync;
- Efektywna praca z dużymi projektami;
- Każda rewizja to obraz całego projektu;
- Przenośność (GNU/Linux, Unix-based, Windows z Cygwin lub MinGW)





Google Trends: „git download”





Instalacja Git

Windows **<https://git-scm.com/download/win>**

Mac OS **<https://git-scm.com/download/mac>**

Linux **<https://git-scm.com/download/linux>**





```
Bash Git for Windows - /c

Gniewko@PHPCamp /c
$ git --version
git version 2.18.0.windows.1

Gniewko@PHPCamp /c
$ .....
```





git --version

git version 2.18.0.windows.1





IAI S.A. THE BEST
ONLINE
SELLING SOLUTIONS



Gdzie szukać pomocy?

git help

git <command> --help

<https://git-scm.com/docs>

<https://stackoverflow.com/questions/tagged/git>





git config --global user.name
git config --global user.email





git config --global user.name „Imie Nazwisko”
git config --global user.email adres@email.com





git init

Initialized empty Git repository in C:/Workshop/IAI/PHPCamp18/.git/





git add .

Dodaje **wszystkie** pliki w katalogu i podkatalogach jako śledzone pliki repozytorium





git commit -m „opis zmiany”

Zatwierdza **wszystkie** zmiany w śledzonych plikach





Trzy stany w których mogą znajdować się pliki

1. **Zmodyfikowany** – plik został zmieniony, ale zmiany nie zostały jeszcze zachowane w bazie danych
2. **Śledzony** – plik zostanie zatwierdzony tak-jak-jest w najbliższej operacji commit
3. **Zatwierdzony** – zmiany zostały bezpiecznie zachowane w lokalnej bazie danych





Struktura Git'a

Working directory (Katalog roboczy)

Katalogi i pliki
Dowolne operacje na plikach (w tym edycja)

Staging area (Przechowalnia)

Prosty plik w katalogu .git
Aktualne zmiany, niezatwierdzone

.git (Repozytorium)

Metadane oraz baza danych
(obiektoowa baza plików)





Trzy stany w których mogą znajdować się pliki

1. **Zmodyfikowany** – zmiany nie są śledzone (working directory)
2. **Śledzony** – plik zmodyfikowany, a zmiany są śledzone (staging area)
3. **Zatwierdzony** – znajduje się w repozytorium (katalog .git)





**Git nie jest kopią bezpieczeństwa
Twojego projektu**





Git jest historią zmian Twojego projektu





Rewizja, czyli stan (snapshot) projektu

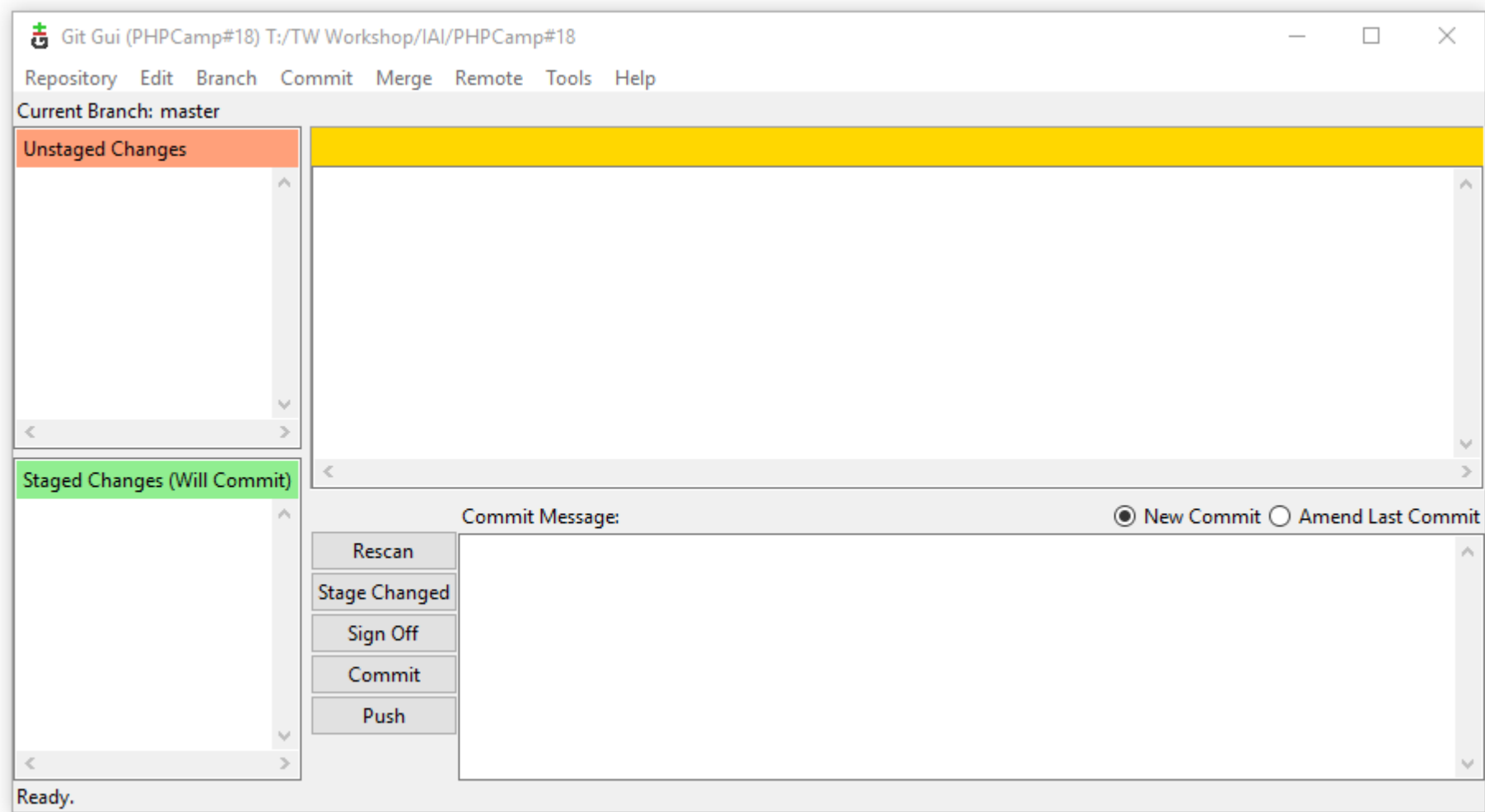


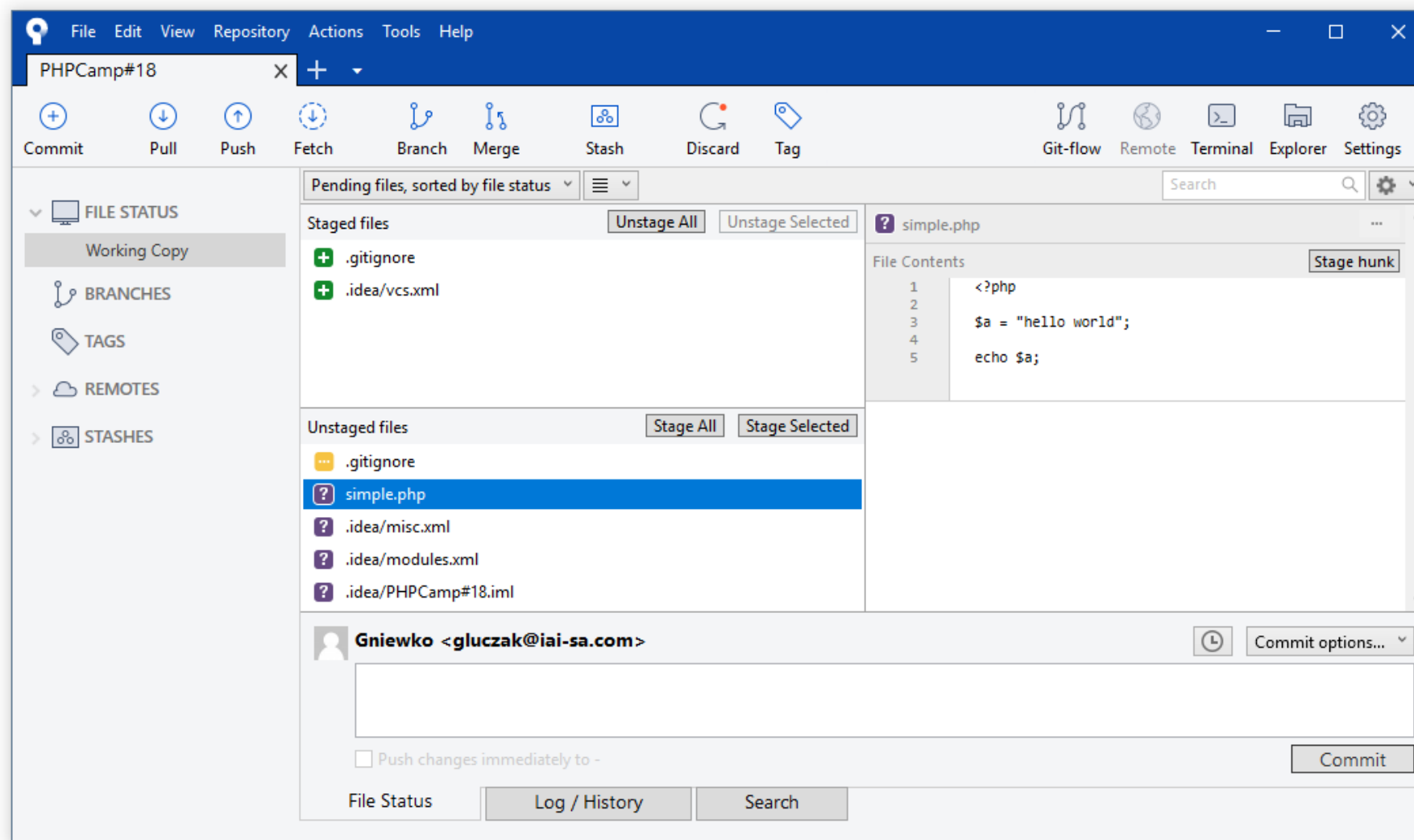


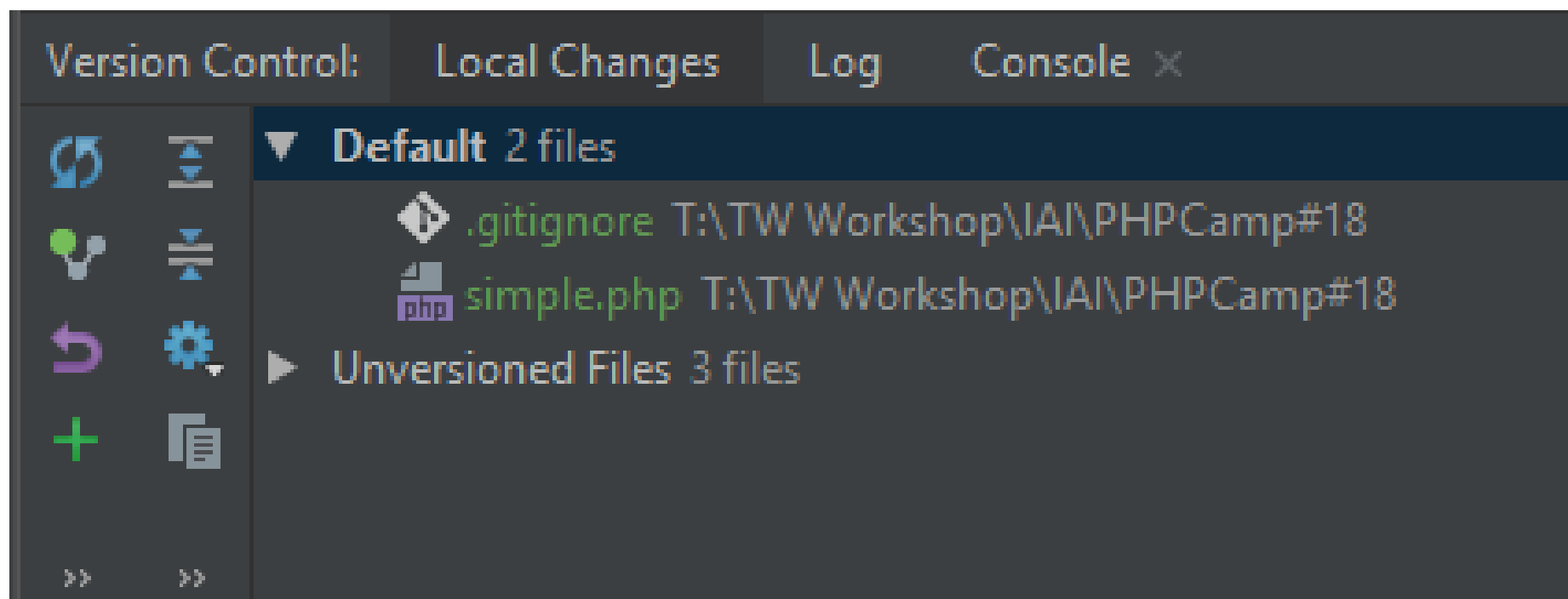
git log

Pokazuje ostatnie zatwierdzone zmiany (commit-y)











<https://www.sourcetreeapp.com/>

Uwaga: do uruchomienia programu wymagana jest darmowa rejestracja na stronie producenta





Branch

Czyli „piaskownica developera”

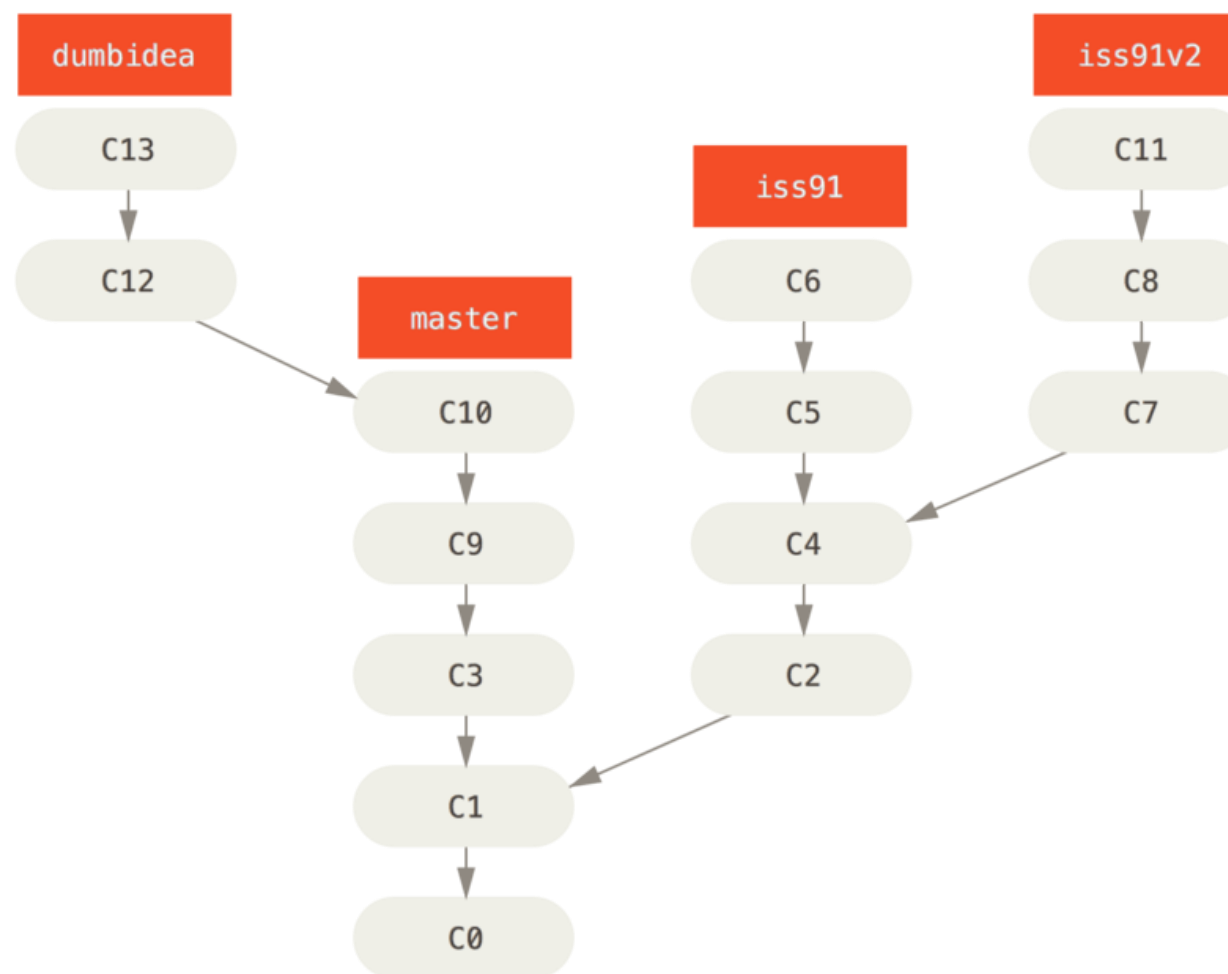
Gałąź (ang. branch), osobny pion historii rewizji głównego nurtu projektu, która po wprowadzeniu w niej zmian może, ale nie musi, zostać włączona do głównego pionu projektu.

Przykład: na nowej gałęzi projektu może być rozwijana eksperymentalna funkcja, która zagraża działaniu projektu, gdyby znalazła się na „produkcji”.





Branches / Gałęzie





git branch

Pokazuje listę gałęzi w lokalnym repozytorium





git branch <branch name>

Tworzy nową gałąź z obecnej zatwierdzonej rewizji





git checkout <branch name>

Zatwierdzanie rewizji (commit) będzie w ramach wskazanej gałęzi – inaczej „przełączenie się” na wskazany branch





git branch -m <new name>

Zmienia nazwę obecnego branch'a





git merge <branch_name>

Łączy dwie gałęzie, obecną ze wskazaną

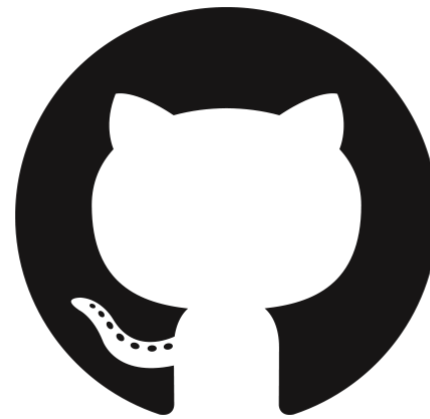




git tag

Pokazuje listę tagów – rozpoznawalne nazwy dla rewizji





<https://github.com>





git remote add origin <https://github.com/gniewkolu/gitdemo.git>





git clone <https://github.com/gniewkolu/gitdemo.git> .





git remote -v

Lista zdalnych repozytoriów





git branch -a

Lista gałęzi również zdalnych





git branch -vv

Lista gałęzi ze szczegółowymi informacjami np. śledzone branch-e





git fetch [remote_name]

Synchronizuje zmian repozytorium, jednakże Working Directory pozostaje bez zmian





git checkout -b my_branch origin/remote_branch





Repozytorium zdalne nie posiada Working Directory





git push -u <remote> <local_branch>:<new_remote_branch>

Wysyła wskazany branch jako nowy na zdalne repozytorium oraz ustawia śledzenie branch lokalnego





git pull = git fetch + git merge

Jeżeli nasz branch jest już śledzony z branchem zdalnym możemy zintegrować zdalne zmiany lokalnie jednym poleceniem





Kodowanie zespołowe

Mocne strony

Słabe strony

Szanse

Zagrożenia





Git zabezpiecza Cię przed utratą zmian w projekcie





Dlatego czasem powstają tzw. konflikty





Kiedy może powstać konflikt?

```
1 <?php
2
3 $a = "hello world";
4
5 echo $a;
```

```
1 <?php
2
3 $a = "hello world";
4
5 //echo $a;
```



git merge <remote>/<branch_name>

Auto-merging simple.php

CONFLICT (content): Merge conflict in simple.php

Automatic merge failed; fix conflicts and then commit the result.





```
<?php
```

```
$a = "hello world";
```

```
<<<<<<< HEAD
```

```
if(empty($a)){  
    echo 'N/A';  
}
```

```
else{  
    echo $a;
```

```
=====
```

```
if(!empty($a)){  
    echo $a;  
}
```

```
else{  
    echo 'N/A';
```

```
>>>>>> origin/new_feature_002  
}
```





<div>✓</div> <pre><?php \$a = "hello world"; if(empty(\$a)){ echo 'N/A'; } else{ echo \$a; }</pre>	<div></div> <pre>1 1 2 2 3 3 4 4 5 5 6 7 8 9 10</pre>	<div></div> <pre><?php \$a = "hello world"; echo \$a;</pre>	<div>✓</div> <pre>1 2 3 4 5 6 7 8 9 10</pre>	<div>✓</div> <pre><?php \$a = "hello world"; if(!empty(\$a)){ echo \$a; } else{ echo 'N/A'; }</pre>
--	---	---	--	---



git log

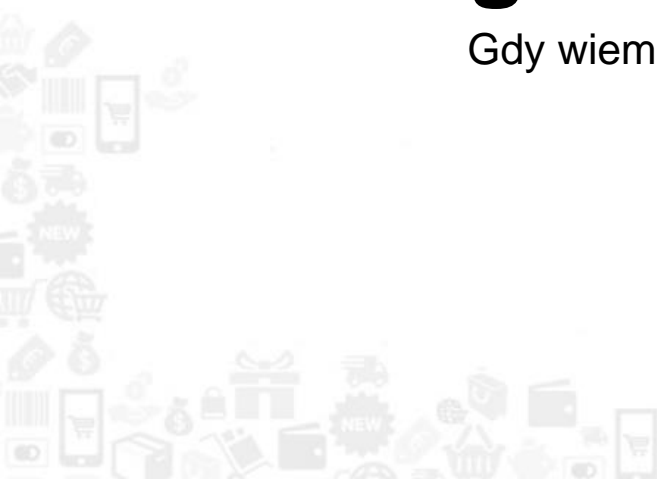
Gdy nie wiem kiedy ostatnio zmieniłem coś w repozytorium

git status

Gdy nie wiem, co ostatnio zmieniłem

git show / git diff

Gdy wiem co i kiedy, ale nie wiem jak





IAI S.A. THE BEST
ONLINE
SELLING SOLUTIONS



In case of fire



1. `git commit`



2. `git push`



3. `leave building`



Dziękuję za uwagę.

Gniewomir Łuczak
Program Developer, IAI S.A.

Kontakt: gluczak@iai-sa.com

