



Uniwersytet
Wrocławski













Git jest git

Zbigniew Koza

Wydział Fizyki i Astronomii

Systemy kontroli wersji

„Lokalny system kontroli wersji”













	11_kwietnia.pdf	349,4 kB	Dokument	13 kwi 2016
	13_kwietnia.pdf	349,4 kB	Dokument	13 kwi 2016
	ankieta_7_10.pdf	199,1 kB	Dokument	7 paź 2017
	ankieta_18_2017.pdf	184,5 kB	Dokument	18 sie 2017
	ankieta_27.01.17.pdf	183,5 kB	Dokument	27 sty 2017
	ankieta-2.10.2017.pdf	195,2 kB	Dokument	2 paź 2017
	ankieta-7.10.2017.pdf	199,1 kB	Dokument	7 paź 2017
	ankieta-15.05.2017.pdf	183,7 kB	Dokument	20 maj 2017
	ankieta-15.11.16.pdf	114,8 kB	Dokument	15 lis 2016
	ankieta-23.05.2017.pdf	183,7 kB	Dokument	23 maj 2017
	ankieta-29.09.2017.pdf	186,3 kB	Dokument	29 wrz 2017
	ankieta-zk-8.10.2017.pdf	199,4 kB	Dokument	8 paź 2017

„Lokalny system kontroli wersji”

Piki z innego projektu

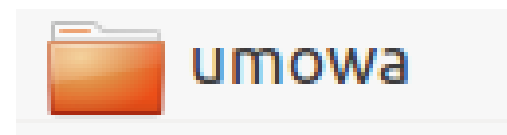
Brak plików źródłowych











Brak kopii bezpieczeństwa

	11_kwietnia.pdf	349,4 kB	Dokument	13 kwi 2016
	13_kwietnia.pdf	349,4 kB	Dokument	13 kwi 2016
	ankieta_7_10.pdf	199,1 kB	Dokument	7 paź 2017
	ankieta_18_2017.pdf	184,5 kB	Dokument	18 sie 2017
	ankieta_27.01.17.pdf	183,5 kB	Dokument	27 sty 2017
	ankieta-2.10.2017.pdf	195,2 kB	Dokument	2 paź 2017
	ankieta-7.10.2017.pdf	199,1 kB	Dokument	7 paź 2017
	ankieta-15.05.2017.pdf	183,7 kB	Dokument	20 maj 2017
	ankieta-15.11.16.pdf	114,8 kB	Dokument	15 lis 2016
	ankieta-23.05.2017.pdf	183,7 kB	Dokument	23 maj 2017
	ankieta-29.09.2017.pdf	186,3 kB	Dokument	29 wrz 2017
	ankieta-zk-8.10.2017.pdf	199,4 kB	Dokument	8 paź 2017

„Lokalny system kontroli wersji”

**Stosowany
w administracji
etc.**



	UMOWA UWr 10.04.2015.doc	121,9 kB	Dokument	10 kwi 2015
	UMOWA UW 2.04.2015.doc	121,9 kB	Dokument	10 kwi 2015
	UMOWA UW 10.04.2015.doc	121,9 kB	Dokument	10 kwi 2015
	UMOWA UW 26.03.2015.doc	118,3 kB	Dokument	31 mar 2015
	UMOWA UW_ śledz. zm..doc	145,9 kB	Dokument	14 kwi 2015
	UMOWA UW_ śledz. zm..pdf	350,3 kB	Dokument	14 kwi 2015
	UMOWA UW_ tekst jedn..doc	141,8 kB	Dokument	14 kwi 2015
	UMOWA UW_ tekst jedn..pdf	321,9 kB	Dokument	14 kwi 2015
	uwagi_radców_UWr.pdf	808,0 kB	Dokument	2 kwi 2015
	załącznik nr 1 do umowy. 30....	46,6 kB	Dokument	31 mar 2015

Archiwum zip (lub GDrive)



Informatyka_
Stosowana_-_
Dokumenty-20...



Informatyka_
Stosowana_-_
Dokumenty-20...



Informatyka_
Stosowana_-_
Dokumenty-20...



Informatyka_
Stosowana_-_
Dokumenty-
2014-12-02.zip

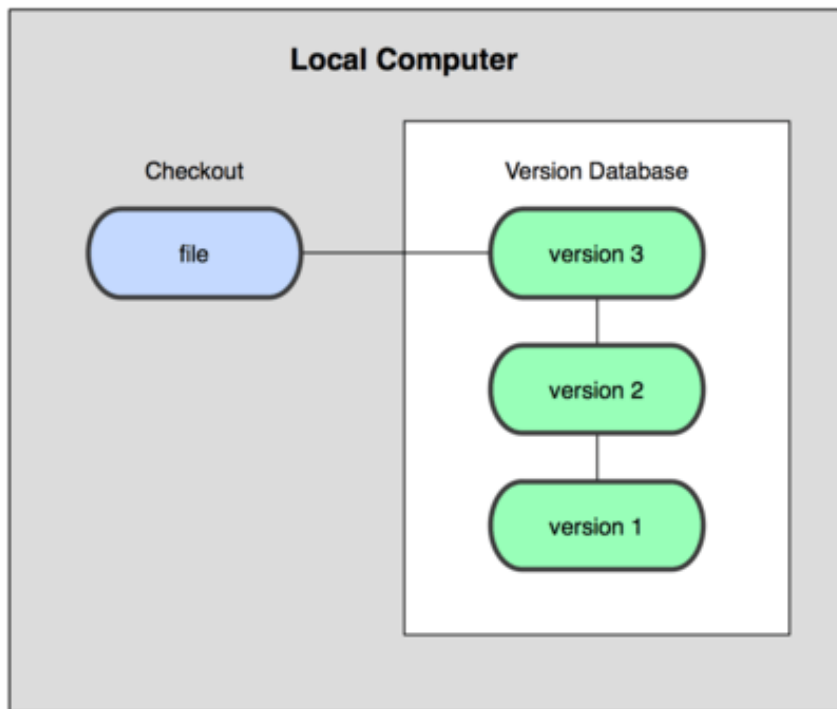


Informatyka_
Stosowana_-_
Dokumenty-
2014-12-10.zip



Informatyka_
Stosowana_-_
Dokumenty-
2014-12-23.zip

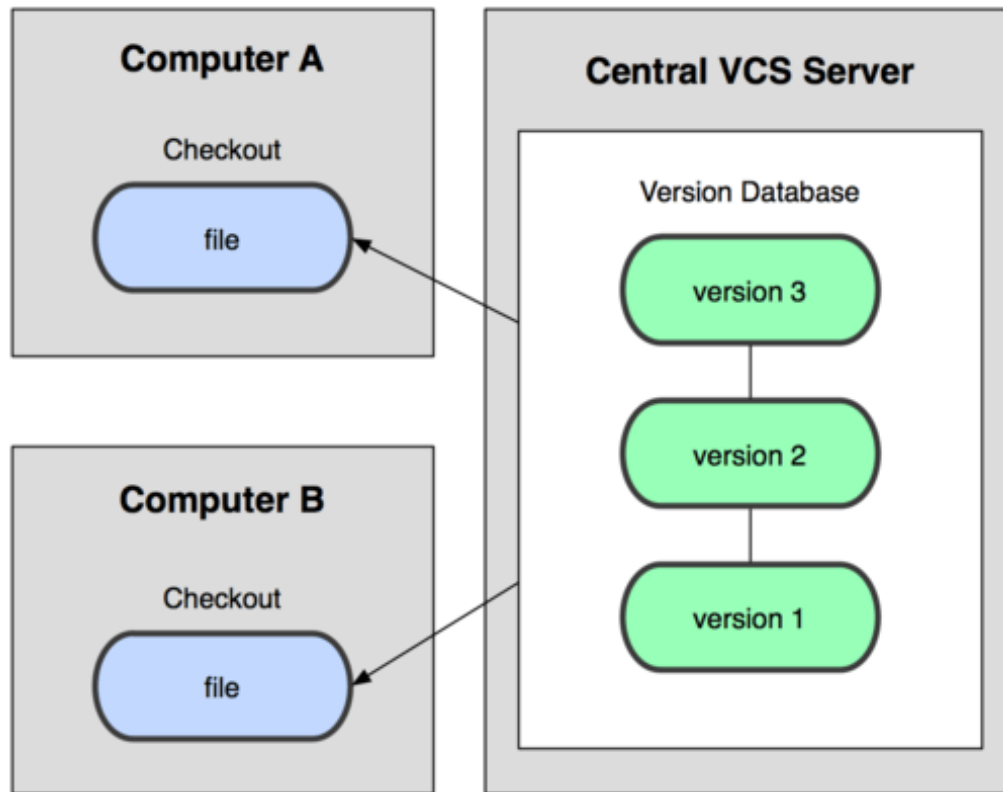
Lokalny system kontroli wersji



- `rcs` (1982-)
- Działa na pojedynczych plikach
- Jeden użytkownik naraz

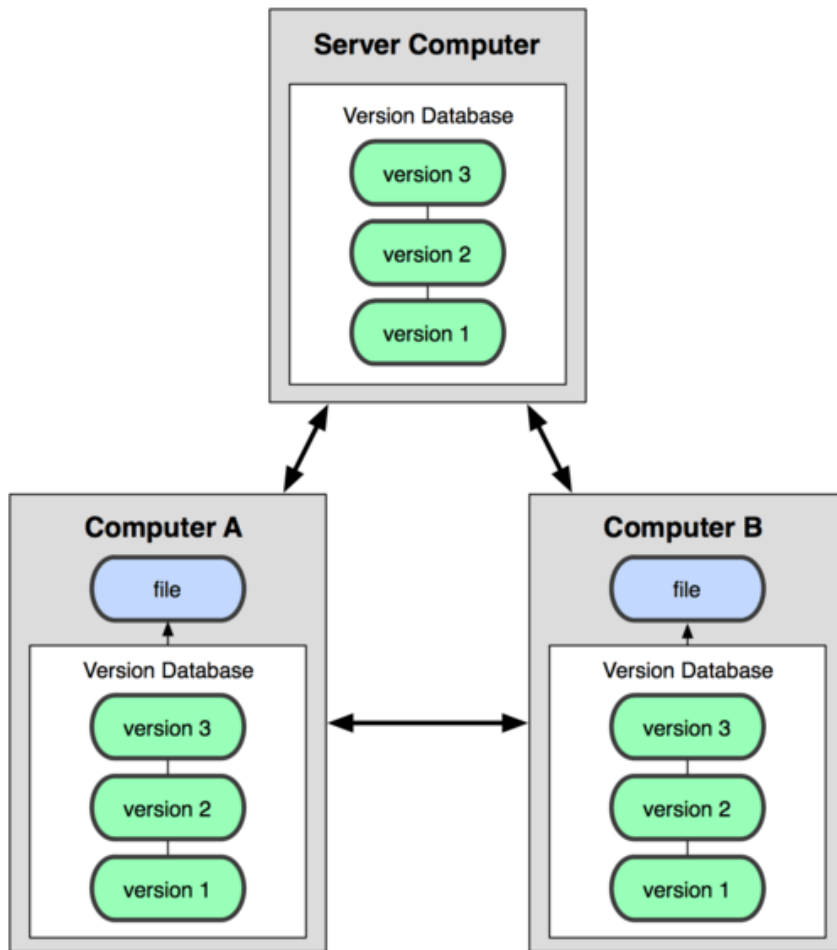
Systemy profesjonalne...

Scentralizowane systemy kontroli wersji



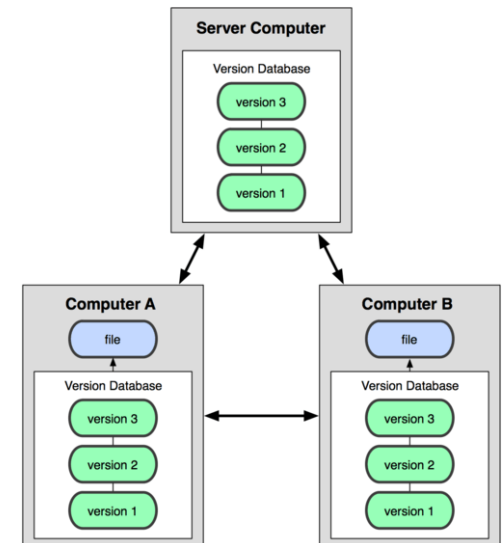
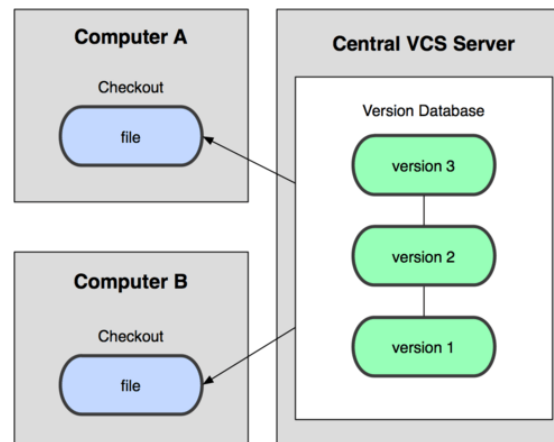
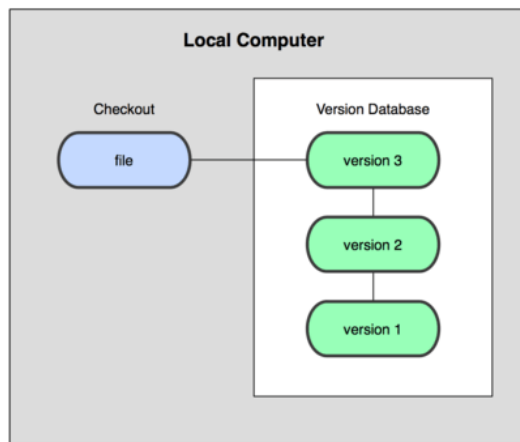
- CVS (1990-2008)
- **SVN** (2000-)
- Perforce (1995-)
- Obsługują całe **projekty** (np.. gcc, apache)
- Dostęp przez sieć, e-mail, etc.

Rozproszone systemy kontroli wersji



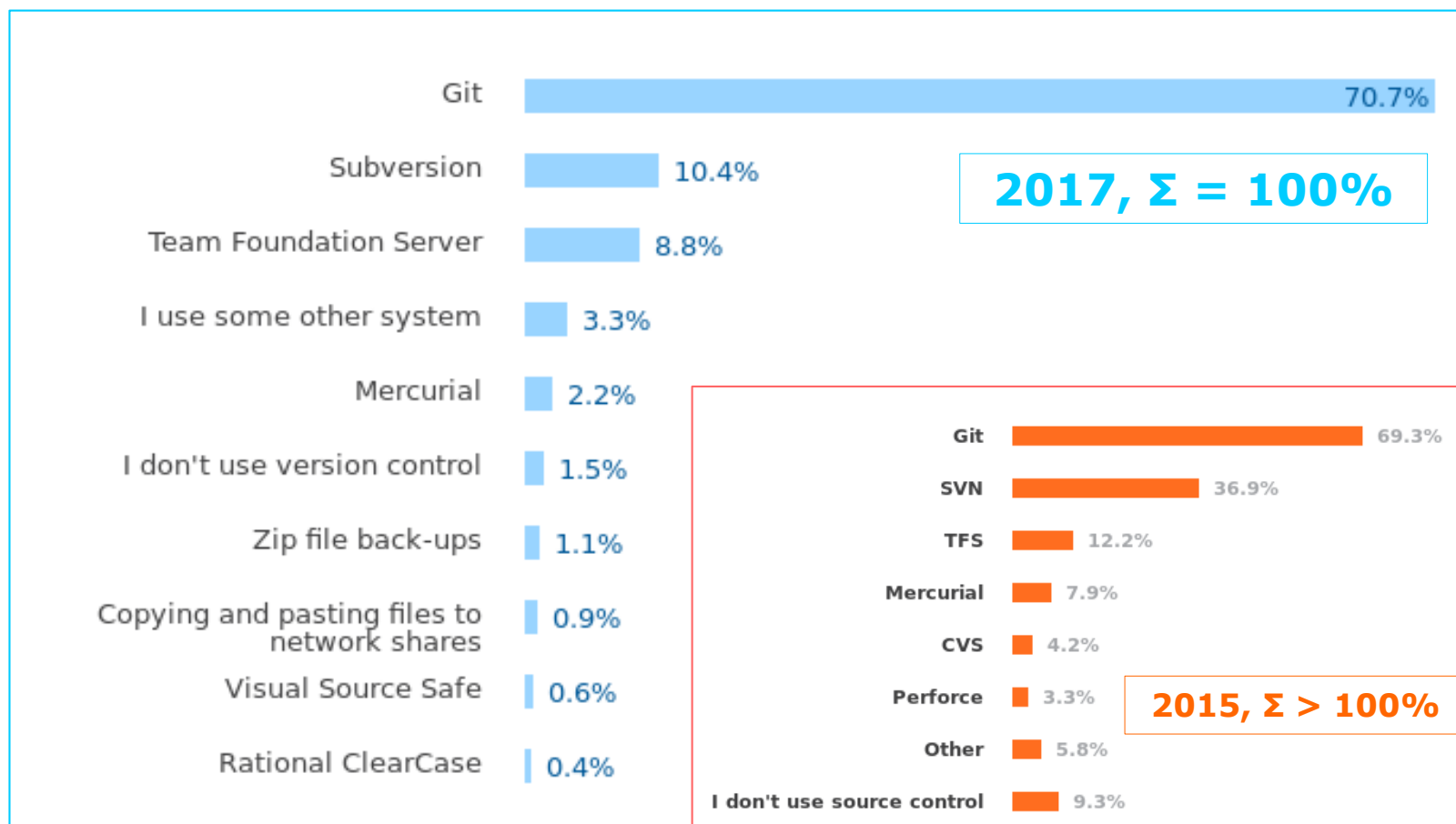
- Git
- Mercurial (Hg)
- Wszystkie węzły równorzędne
- To, który jest „serwerem” jest kwestią umowną

3 generacje



- Plusy: Coraz większe możliwości
- Minusy: Coraz bardziej złożone systemy, coraz trudniejsza nauka

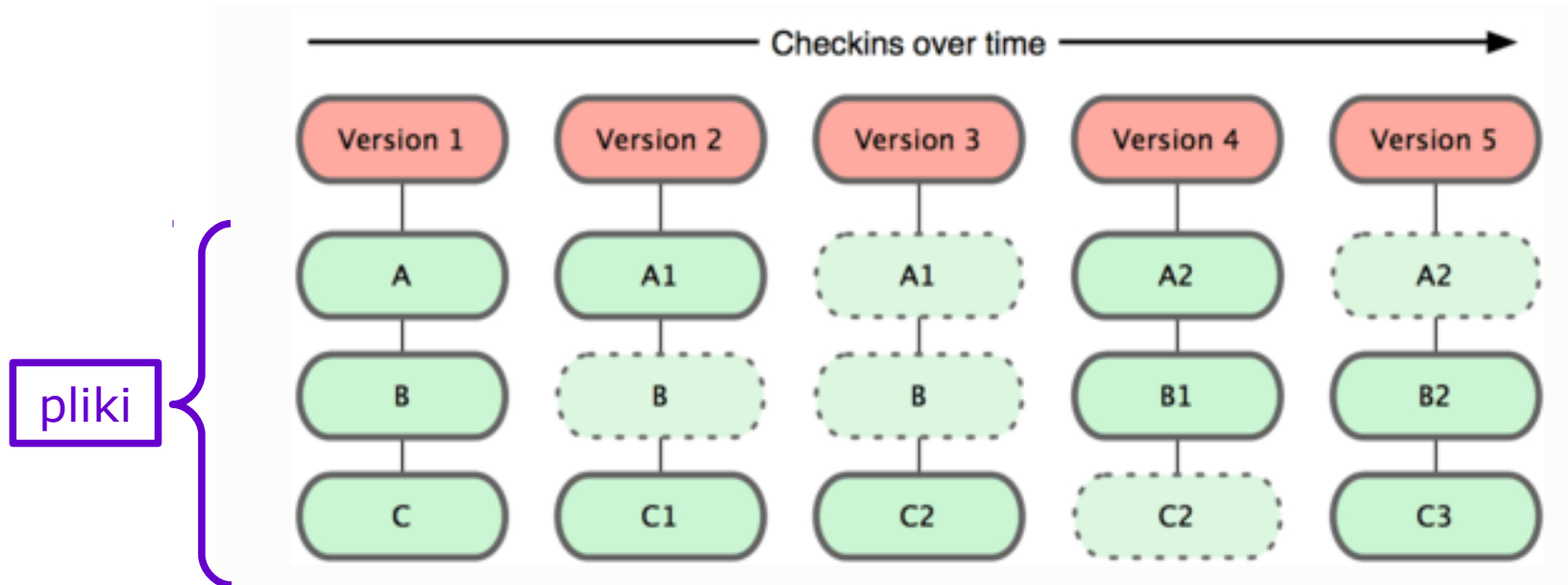
Popularność systemów kontroli wersji



Czyli Git...

Architektura

- Git \approx minisystem plików



Każda wersja reprezentuje stan systemu plików w danej chwili czasu

Architektura: lokalna

- Wszystko (poza synchronizacją) robi się lokalnie
 - Szybkość pracy
 - Możliwość pracy bez dostępu do sieci

3D

literatura

submit_PRE

cpp

results




Tu jest całe, pełne repozytorium

Bezpieczeństwo danych

- Każdy obiekt ma przyporządkowany 40-znakowy klucz SHA-1 wyznaczany na podstawie zawartości tego obiektu
- Klucz ten jest identyfikatorem obiektu (np. migawki systemu plików w danym momencie)

```
commit 43ee714130058bbbe1bfb90dd5d18dcd66e70108  
Author: Zbigniew Koza <zbigniew.koza@uwr.edu.pl>  
Date:   Wed Jul 19 23:03:32 2017 +0200
```



```
git diff 43ee7
```



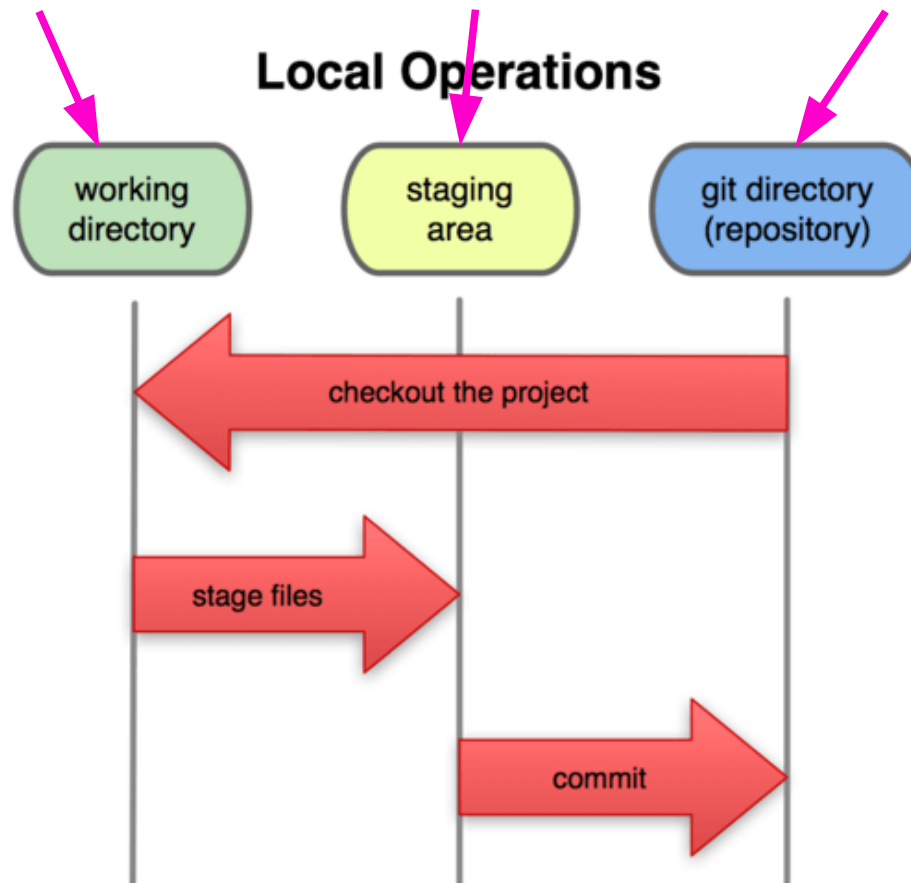
```
git diff 43ee714
```


Bezpieczeństwo danych

- Git zasadniczo tylko dodaje informacje
- Nawet usunięte pliki można łatwo przywrócić

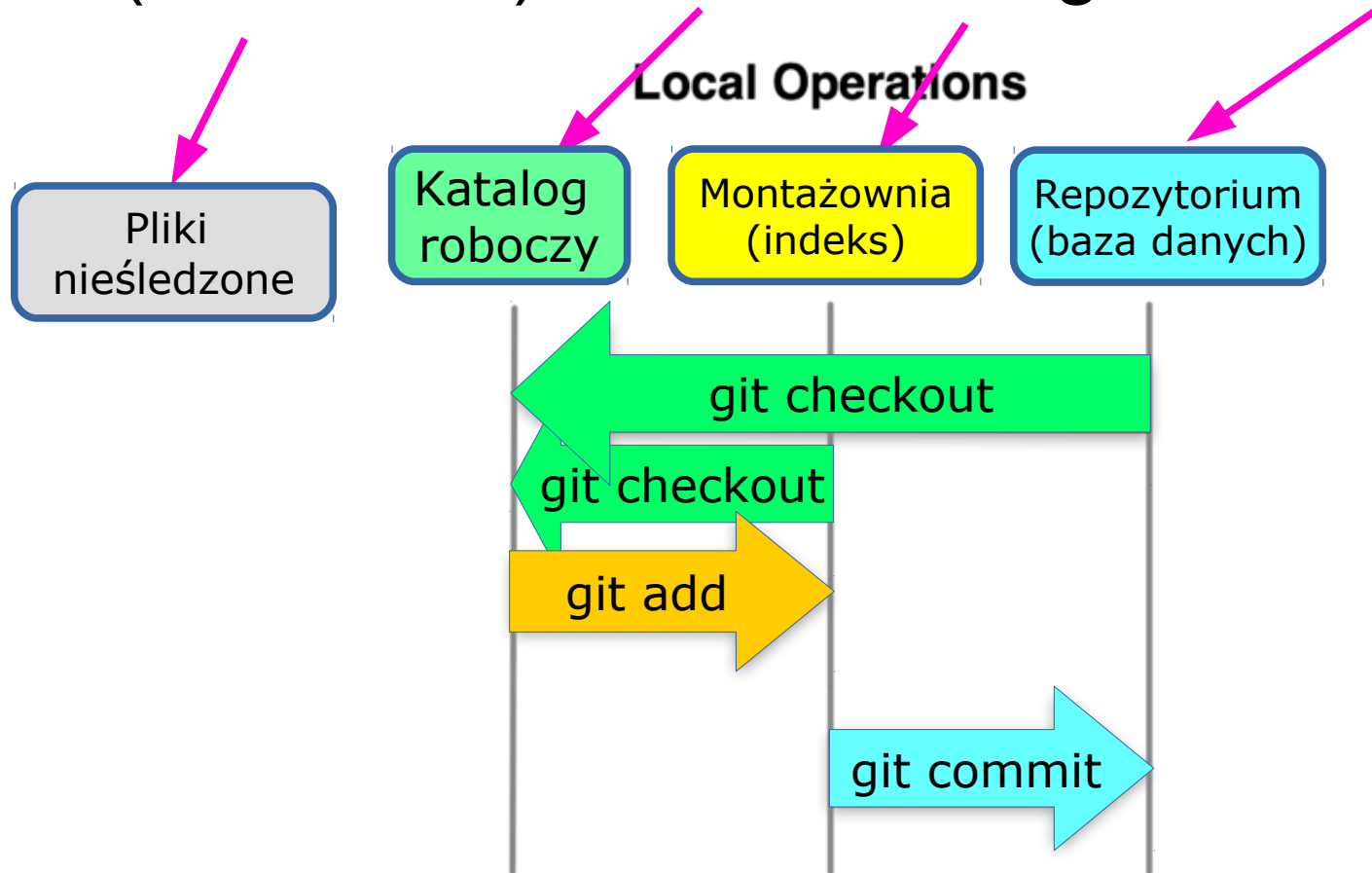
Architektura: trzy stany plików

- Zmodyfikowany, śledzony, zatwierdzony

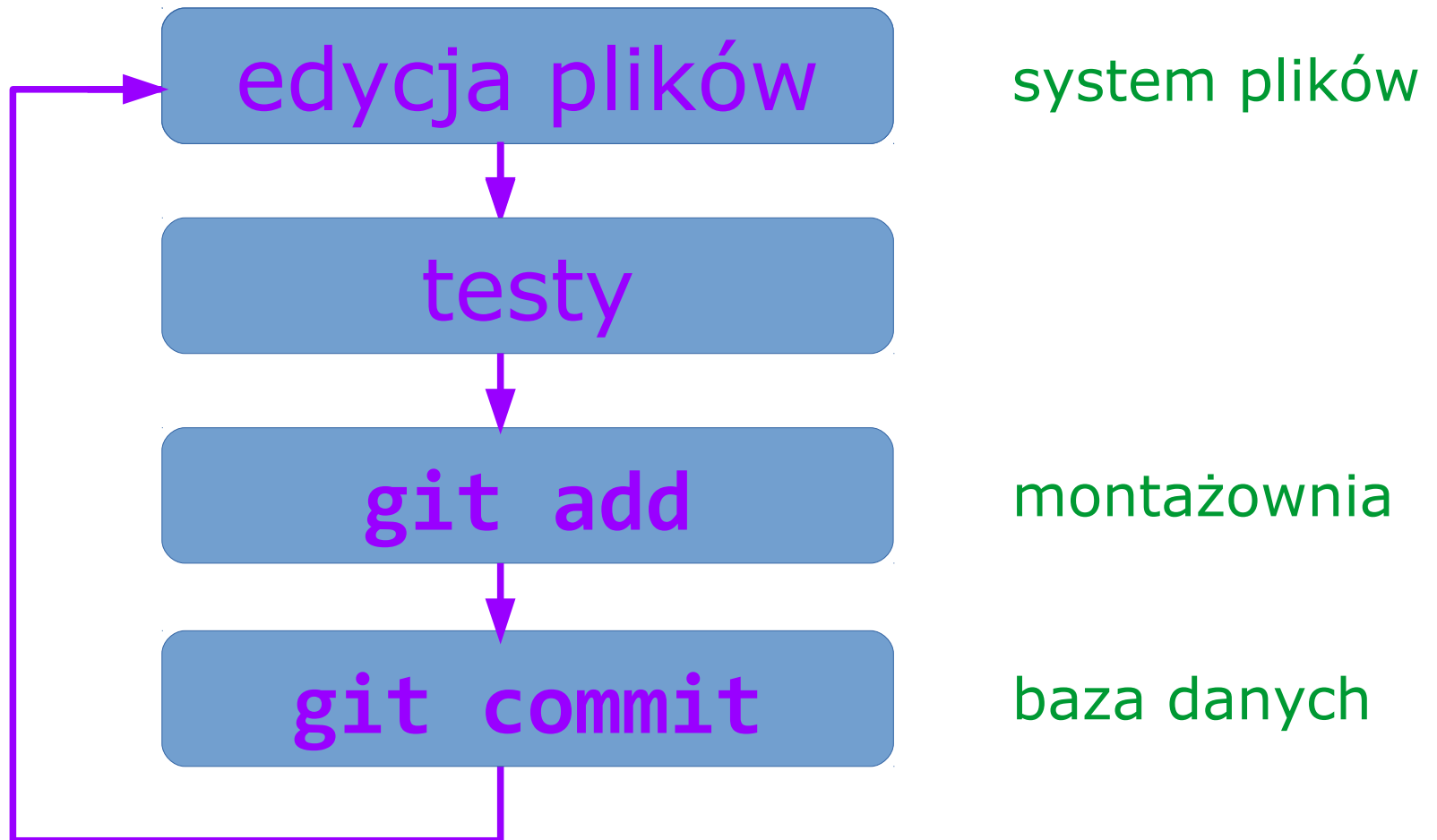


Architektura: trzy stany plików

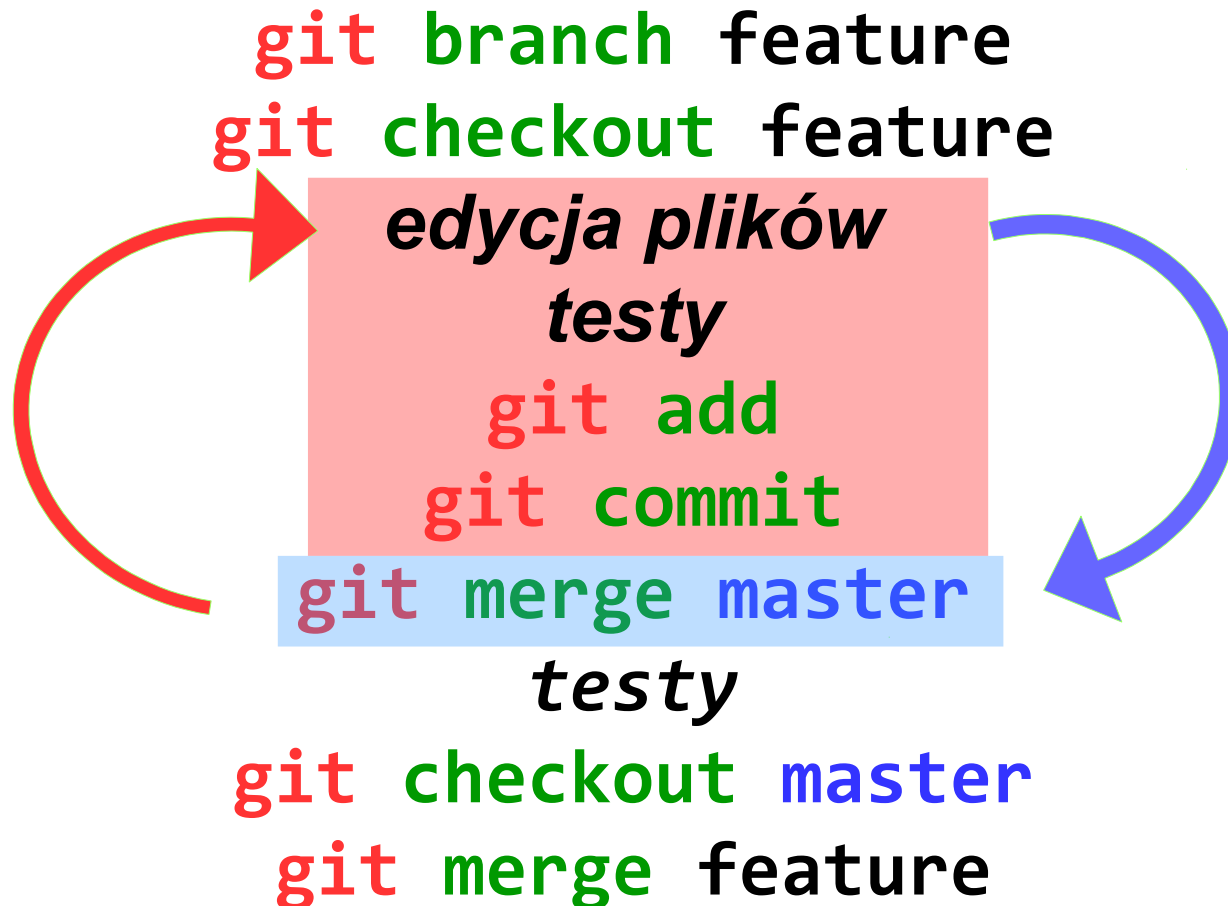
- (untracked), modified, staged, committed



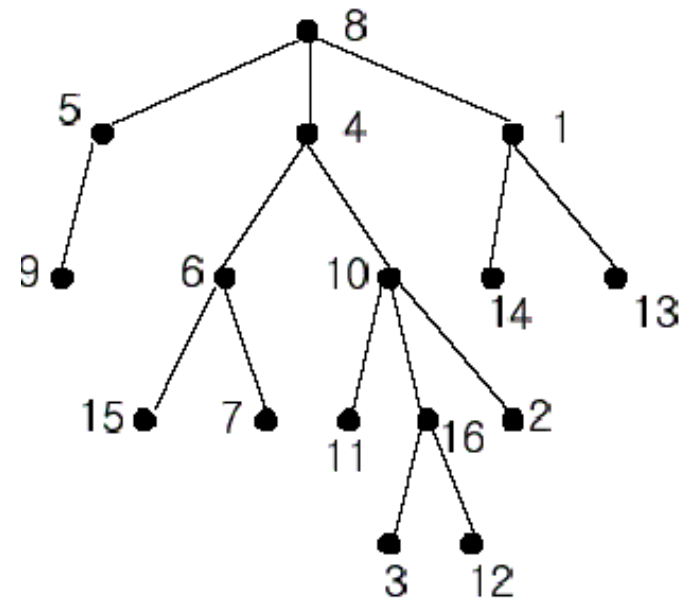
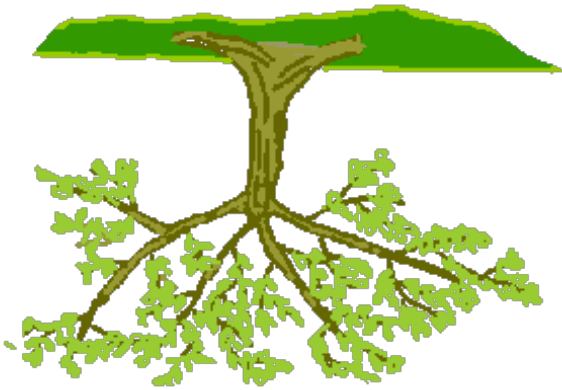
Podstawowy tryb pracy (bardzo mały projekt)



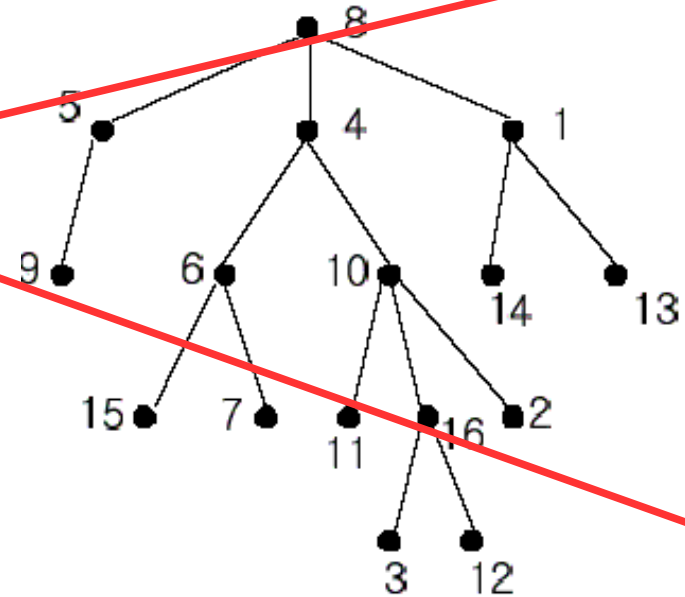
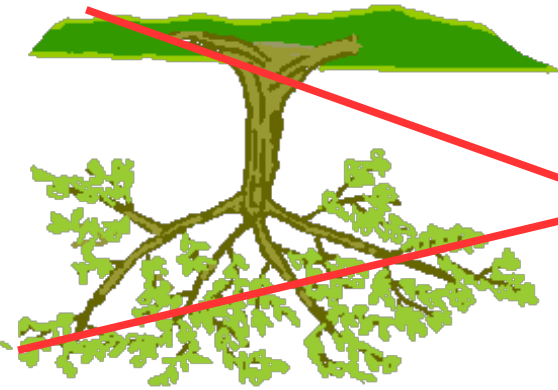
Typowy tryb pracy (projekt lokalny)



Gałęzie git-a

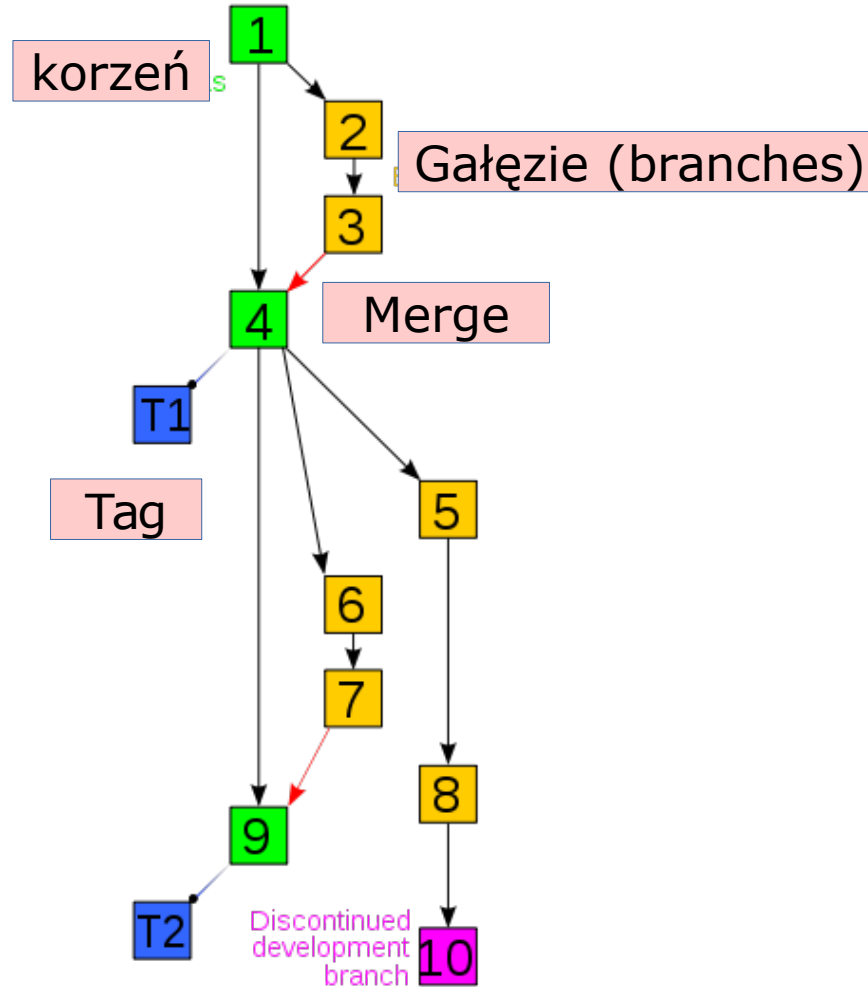


Gałęzie git-a

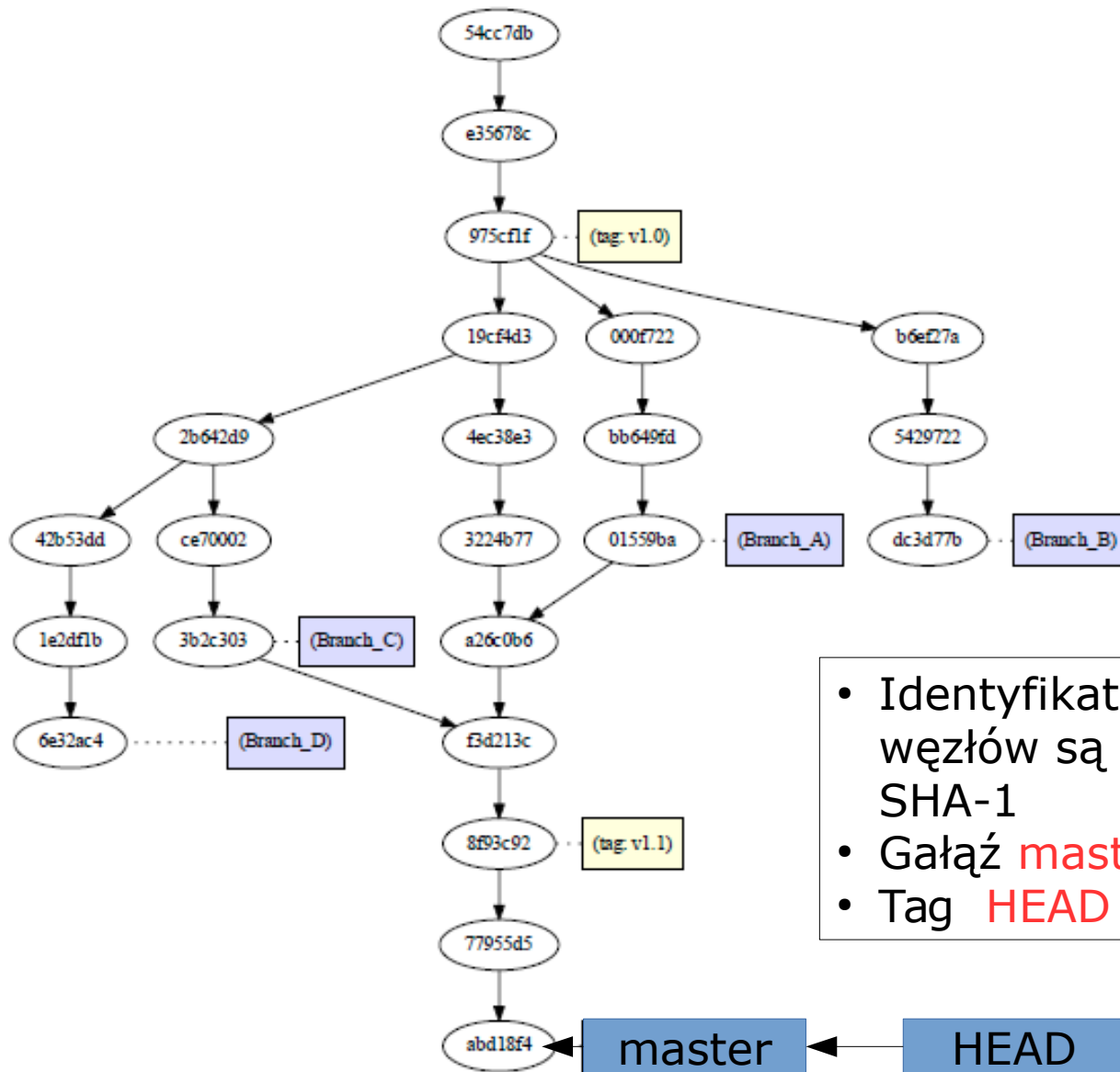


Ta struktura jest
dla VCS
bezużyteczna!

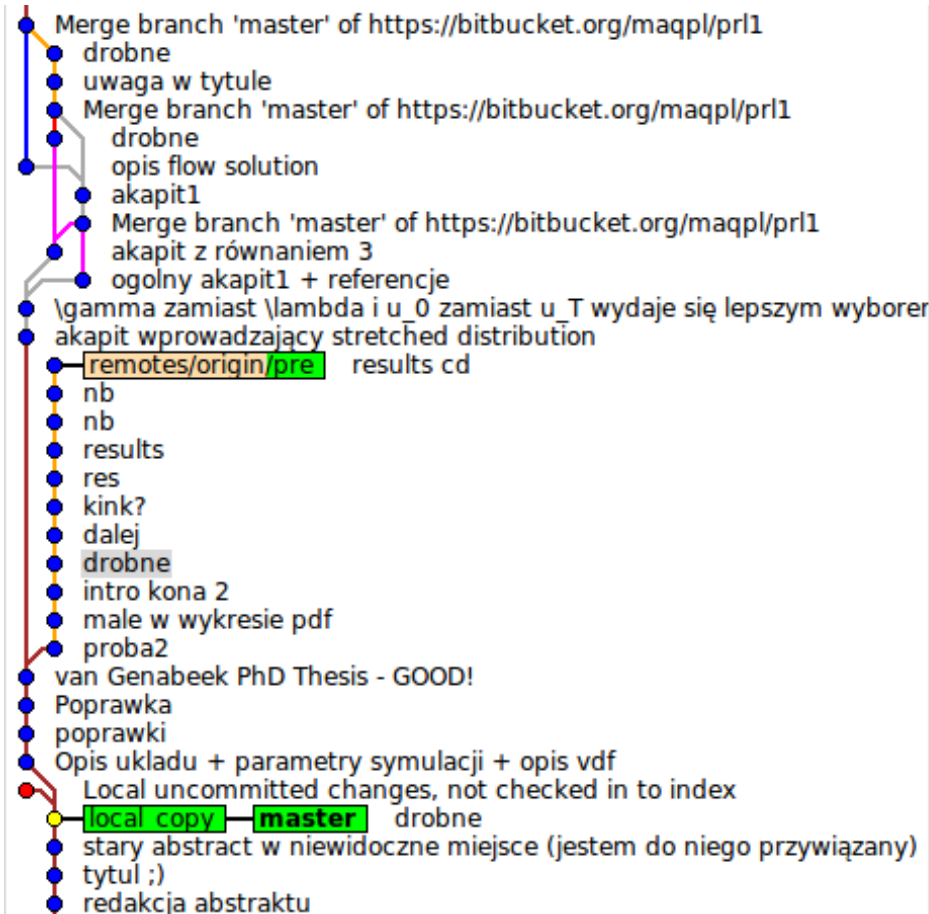
Gałęzie git-a: skierowany graf acykliczny



- Strzałki ilustrują zależności czasowe
- Przepływ informacji jest w odwrotną stronę (dziecko zna swojego rodzica)

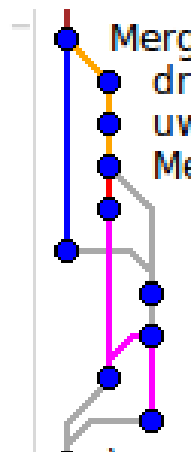


gitk



Maciej Matyka <maciej.matyka@gr	2015-07-30 11:45:23
Zbigniew Koza <zbigniew.koza@ift.	2015-07-30 11:44:50
Zbigniew Koza <zbigniew.koza@ift.	2015-07-30 11:42:00
Zbigniew Koza <zbigniew.koza@ift.	2015-07-30 11:36:12
Zbigniew Koza <zbigniew.koza@ift.	2015-07-30 11:35:46
Maciej Matyka <maciej.matyka@gr	2015-07-30 11:45:08
Maciej Matyka <maciej.matyka@gr	2015-07-30 11:34:39
Maciej Matyka <maciej.matyka@gr	2015-07-30 11:33:16
Zbigniew Koza <zbigniew.koza@ift.	2015-07-30 11:32:10
Maciej Matyka <maciej.matyka@gr	2015-07-30 11:32:21
Zbigniew Koza <zbigniew.koza@ift.	2015-07-30 10:47:50
Zbigniew Koza <zbigniew.koza@ift.	2015-07-30 01:54:14
Maciej Matyka <maciej.matyka@gr	2015-07-30 09:29:34
Maciej Matyka <maciej.matyka@gr	2015-07-29 15:11:49
Maciej Matyka <maciej.matyka@gr	2015-07-29 15:07:11
Maciej Matyka <maciej.matyka@gr	2015-07-29 10:14:57
Maciej Matyka <maciej.matyka@gr	2015-07-28 15:11:16
Maciej Matyka <maciej.matyka@gr	2015-07-28 11:44:42
Maciej Matyka <maciej.matyka@gr	2015-07-28 09:27:56
Maciej Matyka <maciej.matyka@gr	2015-07-27 15:21:55
Maciej Matyka <maciej.matyka@gr	2015-07-27 13:01:52
Maciej Matyka <maciej.matyka@gr	2015-07-24 15:10:17
Maciej Matyka <maciej.matyka@gr	2015-07-24 15:02:25
Maciej Matyka <maciej.matyka@gr	2015-07-16 11:34:29
Jarek <jarek.golembiewski@gamil.c	2015-06-18 17:22:46
Maciej Matyka <maciej.matyka@gr	2015-06-18 10:15:13
Jarek <jarek.golembiewski@gmail.c	2015-06-17 15:16:47
Zbigniew Koza <zbigniew.koza@ift.	2015-06-16 10:20:30
Zbigniew Koza <zbigniew.koza@ift.	2015-06-16 00:47:06
Maciej Matyka <maciej.matyka@gr	2015-06-12 11:44:39
Maciej Matyka <maciej.matyka@gr	2015-06-12 11:41:07

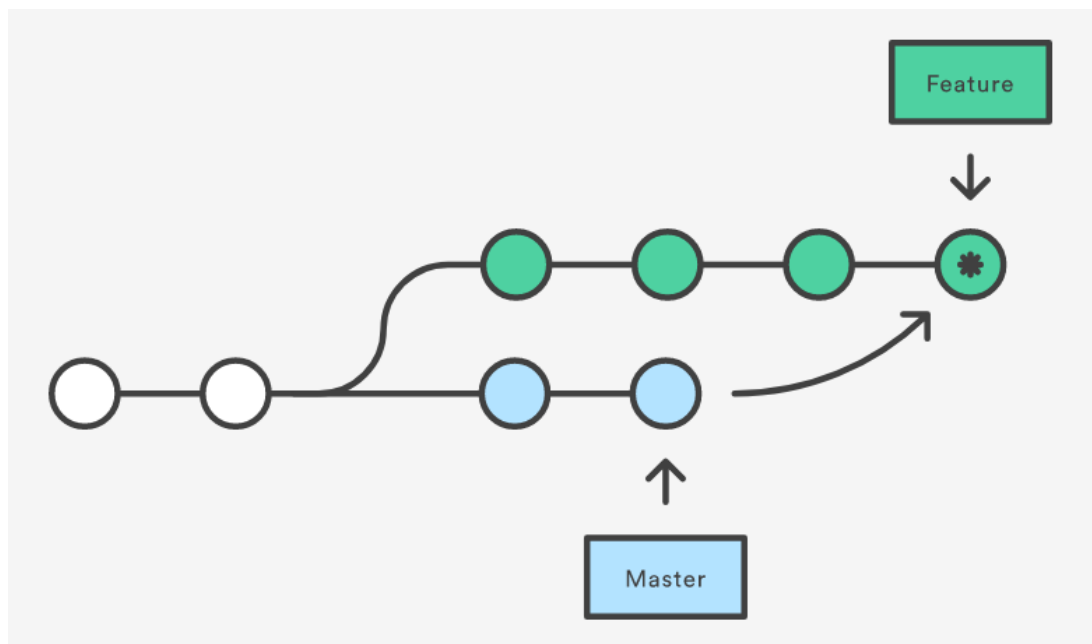
- W architekturze rozproszonej każdy użytkownik wprowadza własną gałąź



Po są są gałęzie?

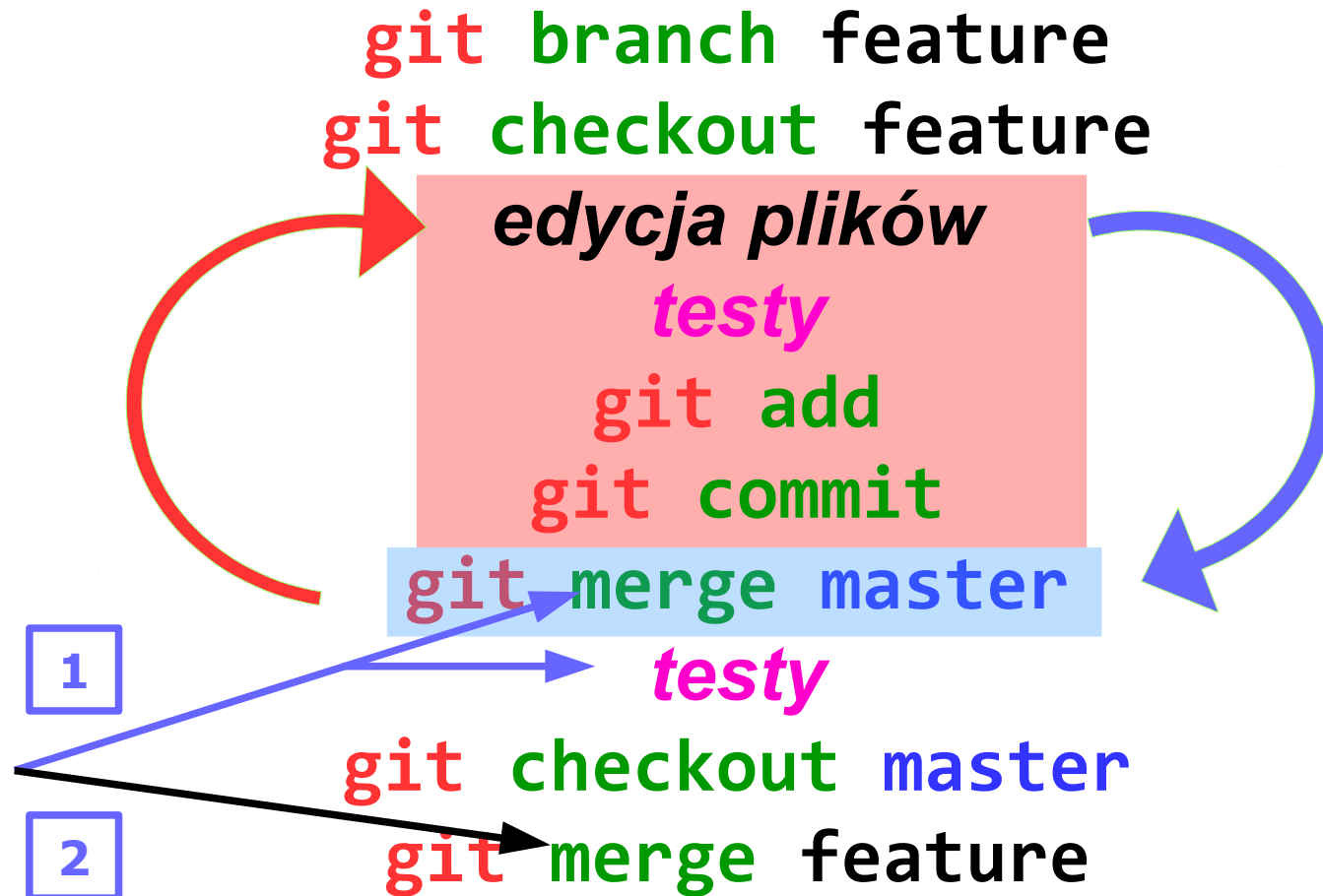
- Żeby uniknąć deweloperskich „race conditions”!
- Żeby móc pracować niezależnie od innych
- Żeby nie bombardować współpracowników swoimi eksperymentami w kodzie

Scalanie gałęzi (*merging*)



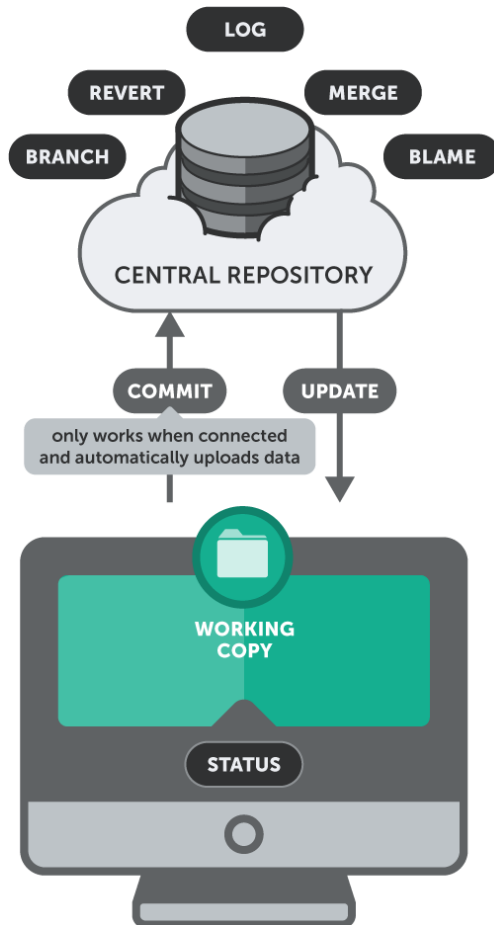
- Wprowadzenie poprawek wykonanych w jednej gałęzi (tu: master) do drugiej (tu: Feature)

Najpierw master->Feature

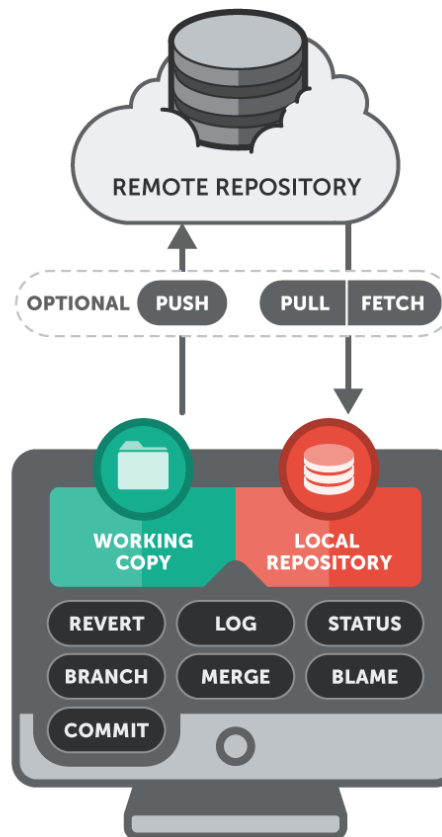


Wiele komputerów...

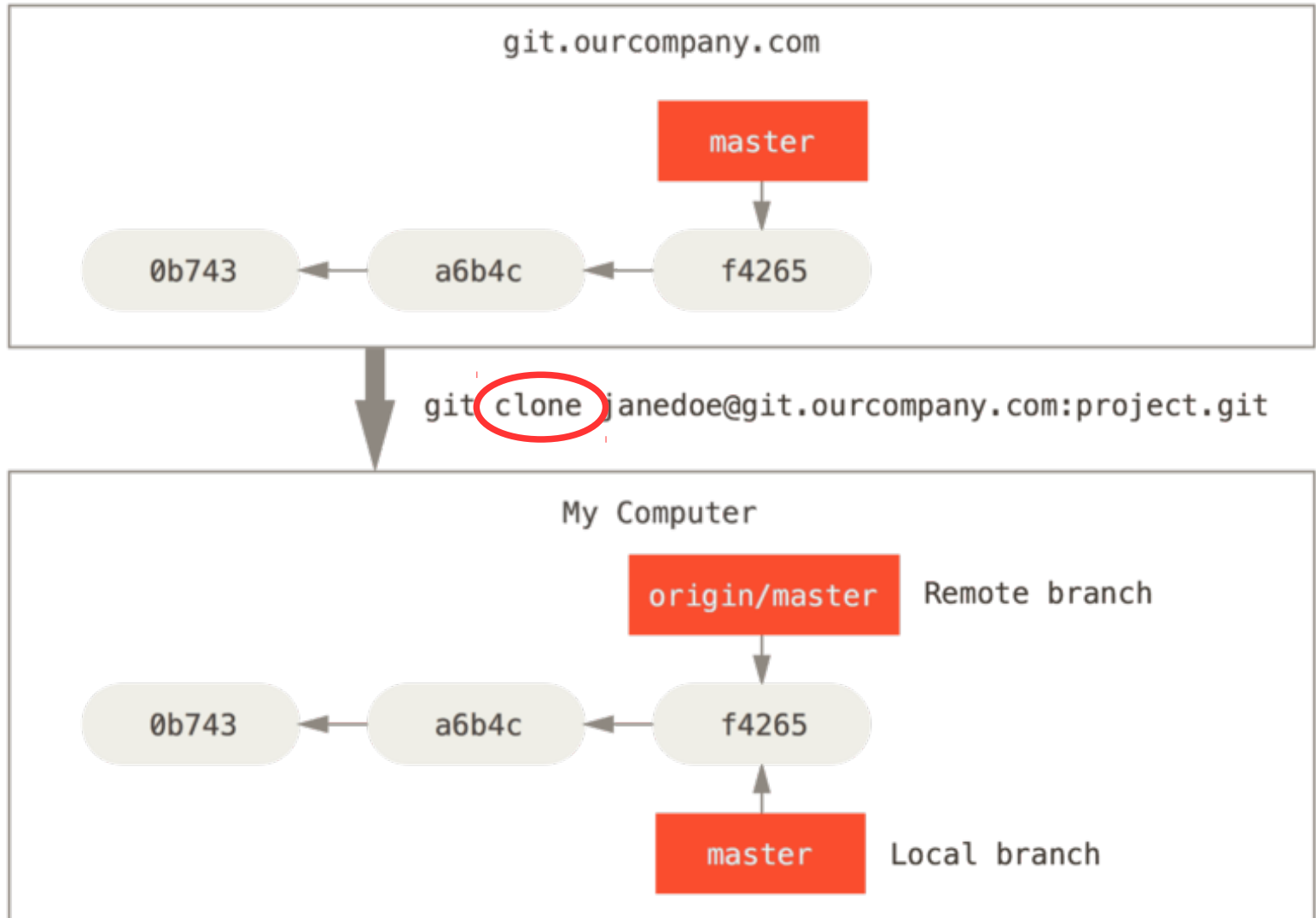
SUBVERSION



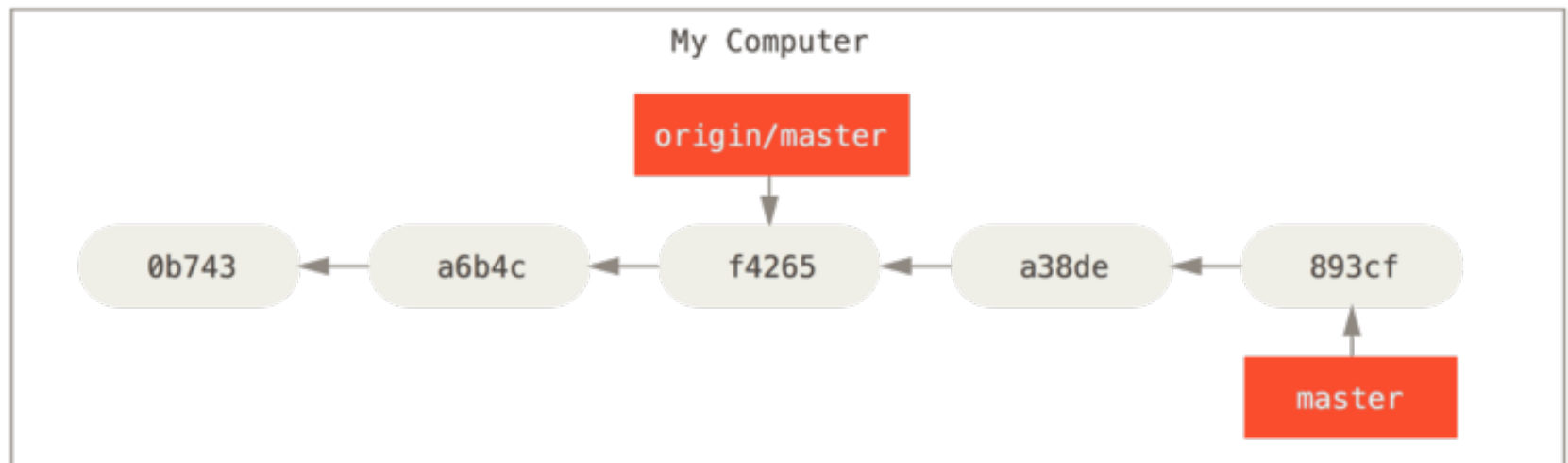
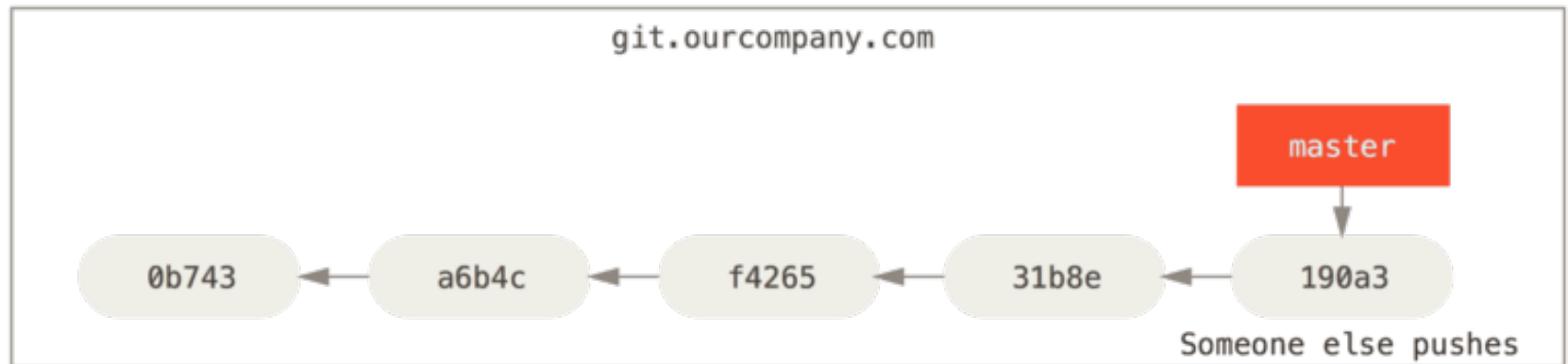
GIT



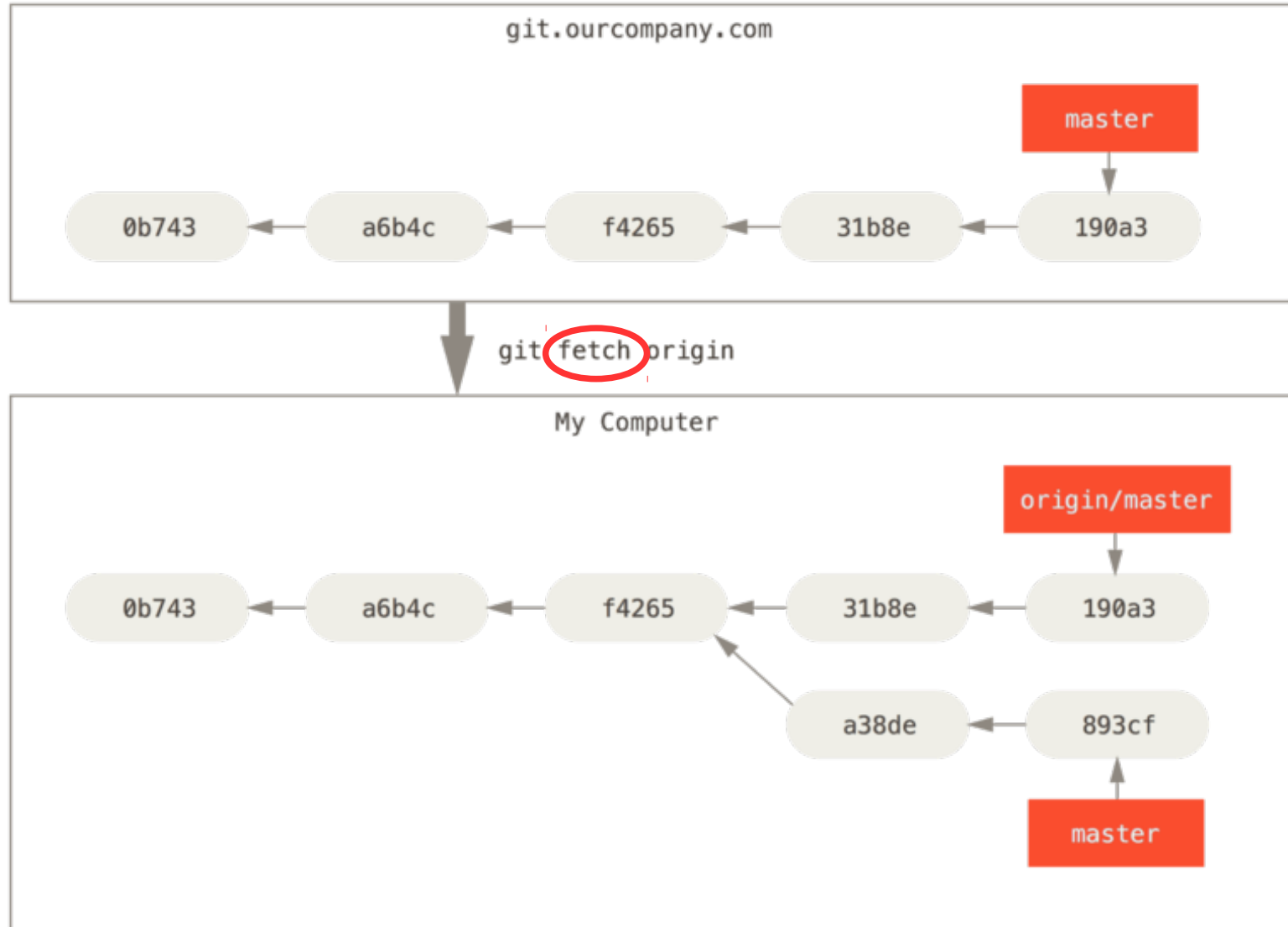
Klonowanie repozytorium



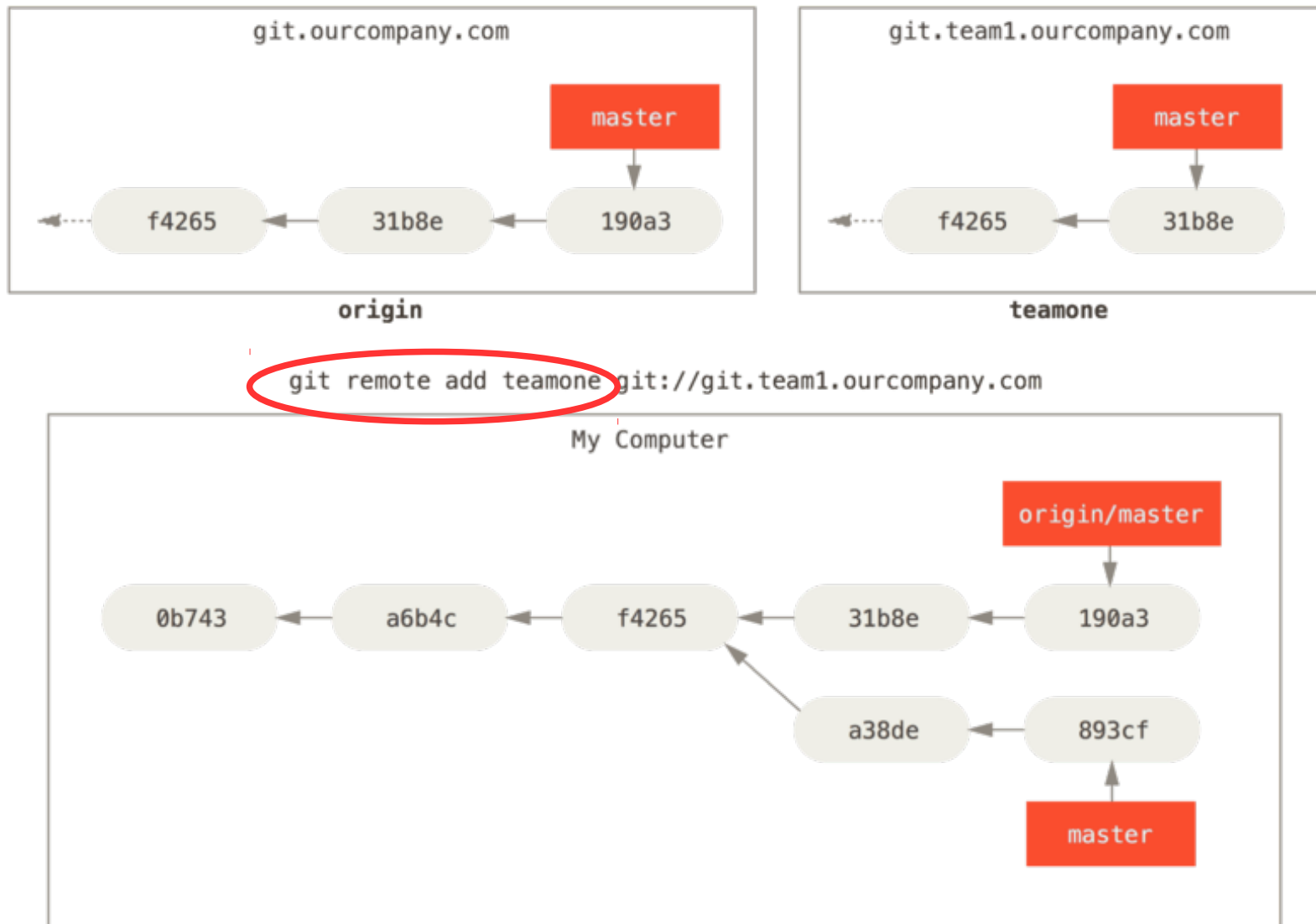
Repozytoria zwykle się rozchodzą



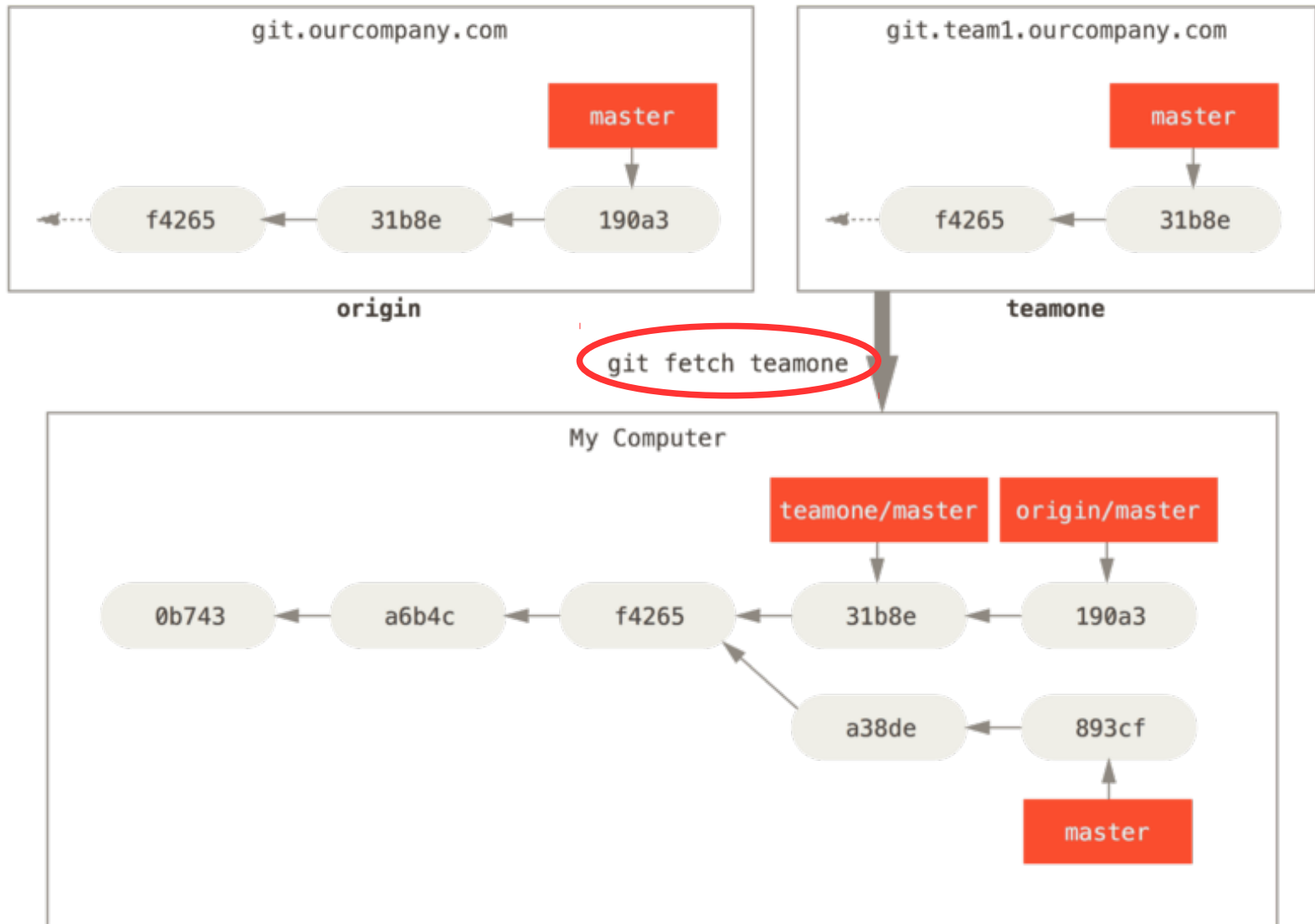
Ściągnięcie wersji zdalnej



Można śledzić kilka repozytoriów



Ściągnięcie danych z repo nr 2



pull

- `git pull = git fetch; git merge`

Tracking branches

```
$ git checkout serverfix  
Branch serverfix set up to track remote branch serverfix from origin.  
Switched to a new branch 'serverfix'
```

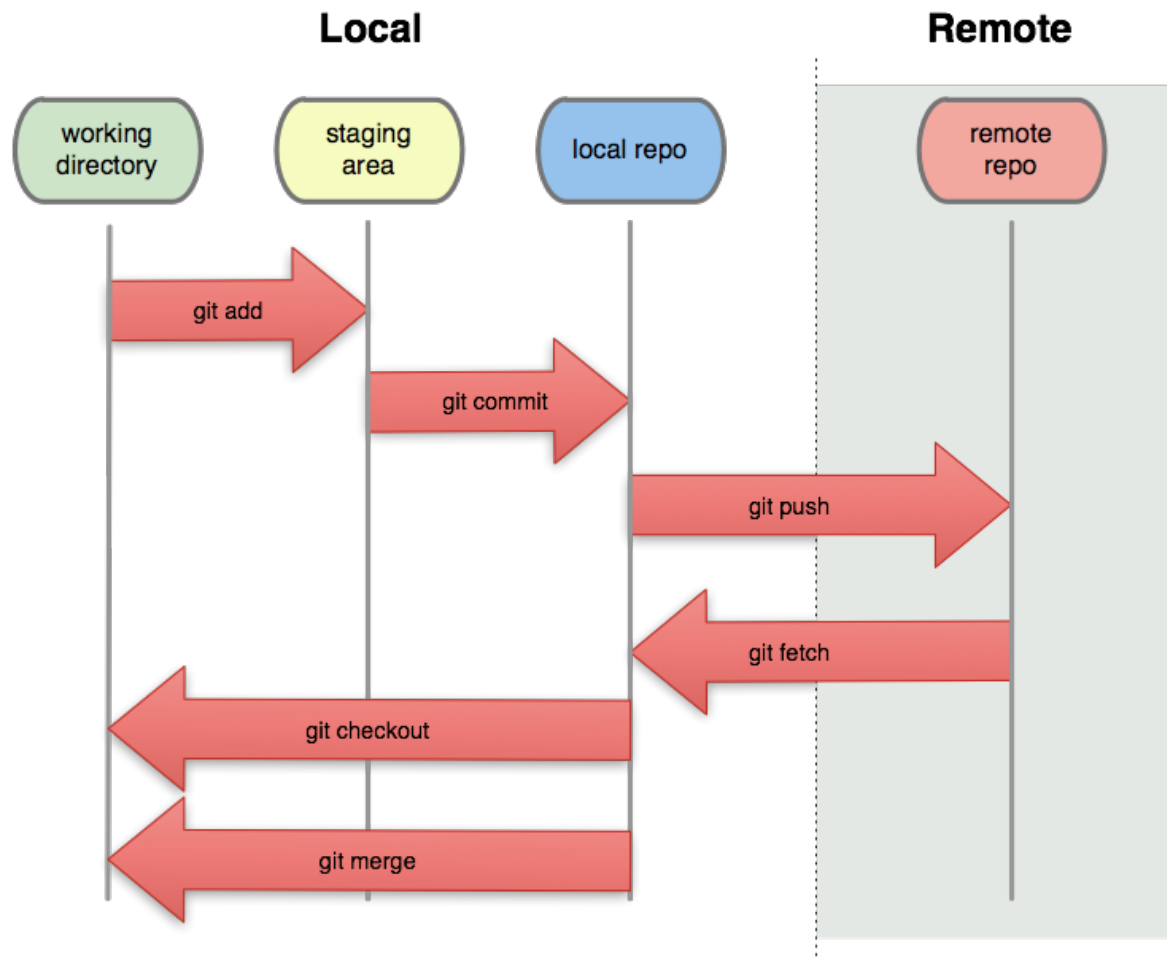
↑
Gałąź serverfix musi istnieć na serwerze
wskazywanym przez tag „origin”

- Gałąź śledząca to lokalna gałąź, która zostanie użyta do łączenia ze śledzoną gałęzią zdalną (*upstream branch*) w przypadku użycia `git pull`
- Por: `git branch -v`

push

- `git push <remote> <branch_name>`
- Przesyła lokalne zmiany (*commits*) do zdalnego repozytorium

Nieco pełniejszy obraz



Konflikty...

How to resolve merge conflicts in Git?



Is there a good way to explain how to resolve merge conflicts in Git?

3871

[git](#)

[git-merge](#)

[merge-conflict-resolution](#)

[git-conflict-resolution](#)



[share](#) [improve this question](#)

edited Jul 22 '16 at 15:47



Steve Chambers

asked Oct



Sn

```
<<<<<<<
```

```
Changes made on the branch that is being merged into. In most cases,  
this is the branch that I have currently checked out (i.e. HEAD).
```

```
|||||||
```

```
The common ancestor version.
```

```
=====
```

```
Changes made on the branch that is being merged in. This is often a  
feature/topic branch.
```

```
>>>>>>>
```

Dokumentacja

Rozbudowany system pomocy

- `> git help <polecenie>`
- `> git <polecenie> --help`
- `> man git<-polecenie>`
- `> man git`
- `> man gittutorial`
- `> man gittutorial-2`
- `> man gitglossary`
- `> man giteveryday`
- `> man gitcore-tutorial`

Liczna i życzliwa społeczność

🔒 <https://stackoverflow.com/questions/315911/git-for-beginners-the-def>

Git for beginners: The definitive practical guide

855 Ok, after seeing [this post](#) by PJ Hyett, I have decided to skip to the end and go with [Git](#).



4067

So what I need is a beginner's **practical** guide to Git. "Beginner" being defined as someone who knows how to handle their compiler, understands to some level what a [Makefile](#) is, and has touched source control without understanding it very well.

"Practical" being defined as this person doesn't want to get into great detail regarding what Git is doing in the background, and doesn't even care (or know) that it's distributed. Your answers might hint at the possibilities, but try to aim for the beginner that wants to keep a 'main' repository on a 'server' which is backed up and secure, and treat their local repository as merely a 'client' resource.

<https://stackoverflow.com/questions/315911/git-for-beginners-the-definitive-practical-guide>

Praca z git

Tworzenie repozytorium

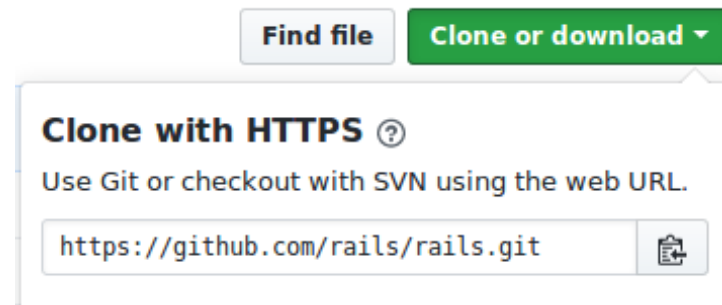
- > `git init`

Kopiowanie istniejącego repozytorium

- > **git clone ...**

```
git clone https://github.com/rails/rails.git
```

<https://github.com/rails/rails>



Tworzenie gałęzi

- `git checkout -b super_pomysl`
- `git branch super_pomysl`
- `git checkout super_pomysl`

