

Git – część 2

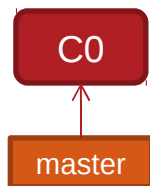
System kontroli wersji

Daniel Kossakowski

1. Rozgałęzianie

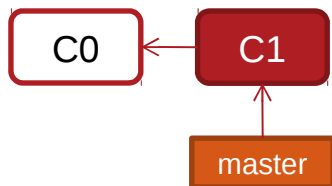
Rozgałęzianie

Jak działa branch?



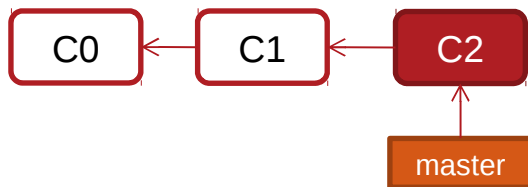
Rozgałęzianie

Jak działa branch?



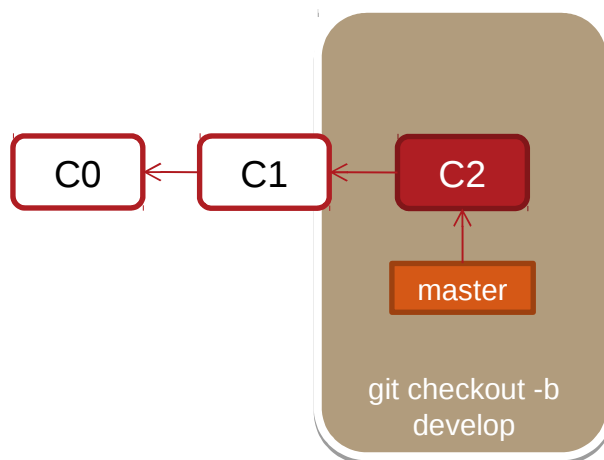
Rozgałęzianie

Jak działa branch?



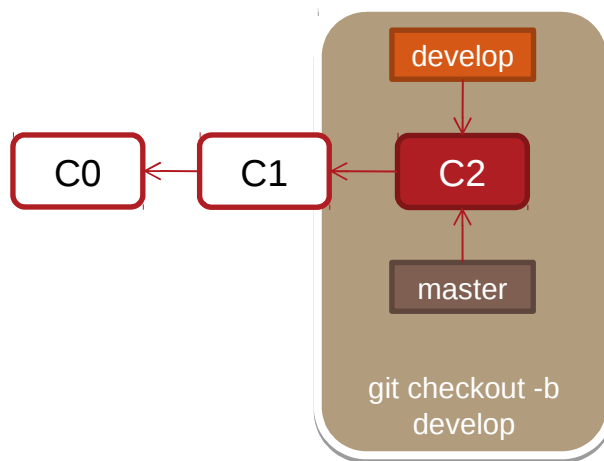
Rozgałęzianie

Jak działa branch?



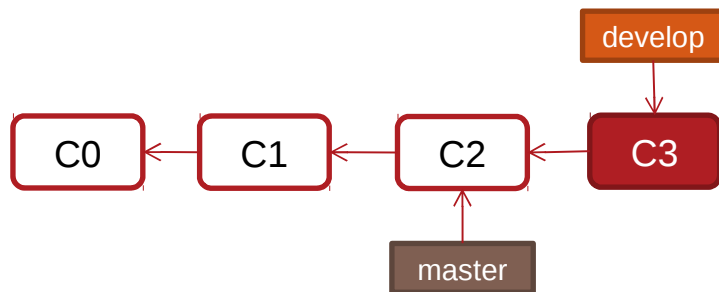
Rozgałęzianie

Jak działa branch?



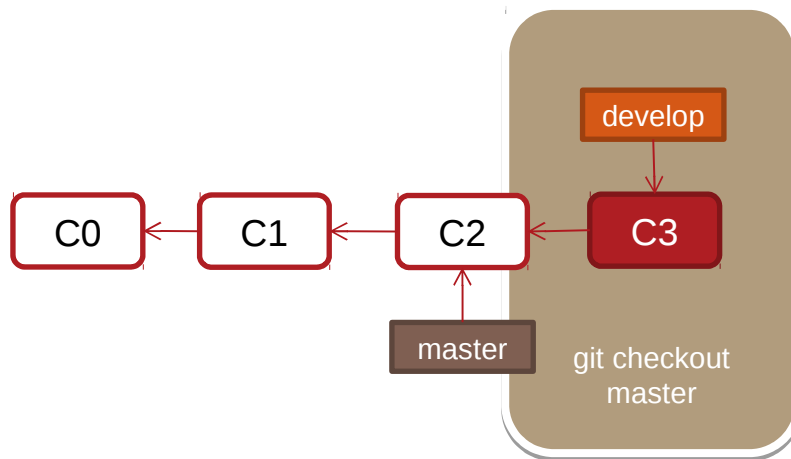
Rozgałęzianie

Jak działa branch?



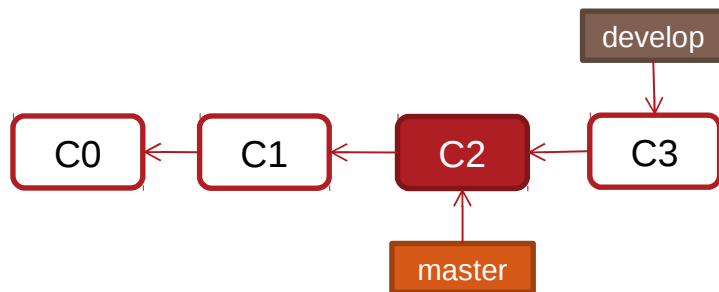
Rozgałęzianie

Jak działa branch?



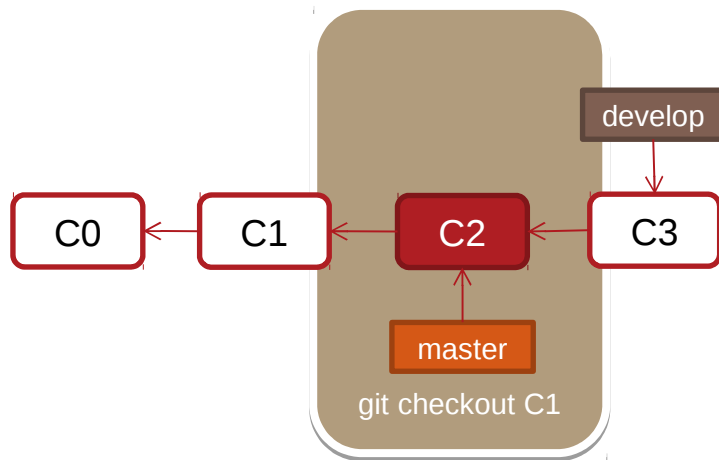
Rozgałęzianie

Jak działa branch?



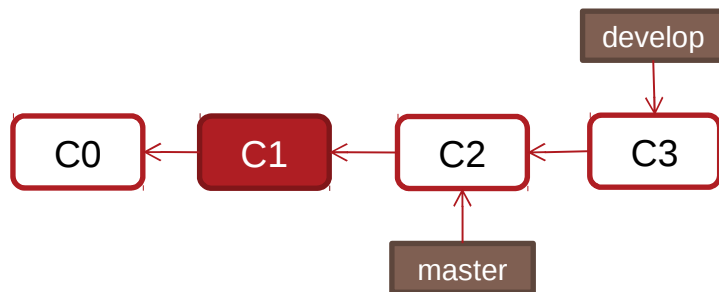
Rozgałęzianie

Detached HEAD



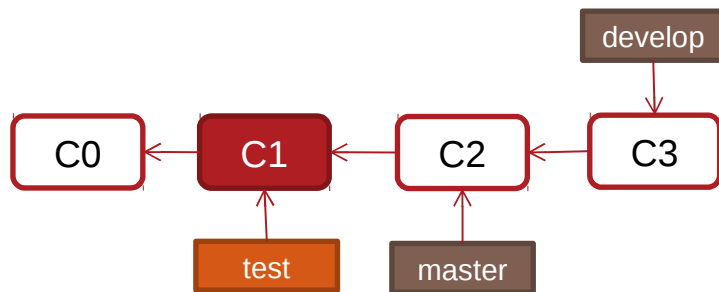
Rozgałęzianie

Detached HEAD



Rozgałęzianie

Powrót do historii



```
linux@iSA ~ $ git checkout master  
linux@iSA ~ $ git checkout -b test 56a4e5c08
```

ĆWICZENIE!



Utwórz folder o nazwie *test5* w swoim katalogu domowym.
Zainicjuj w nim nowe repozytorium.
Sprawdź czy został utworzony katalog *.git*.

2. Scalanie

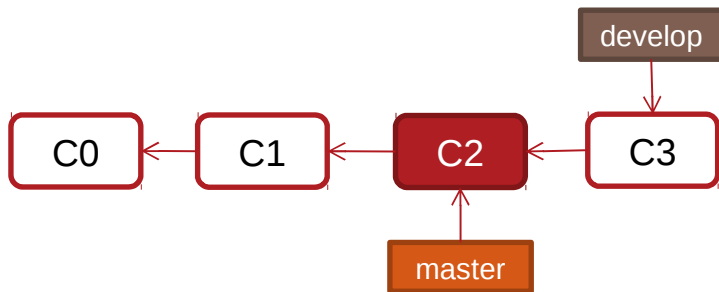
Scalanie

Podstawy

- Dwie metody scalania:
 - git merge
 - ▶ Fast forward
 - ▶ Non-fast forward
 - ▶ 3-way merge
 - git rebase

Scalanie

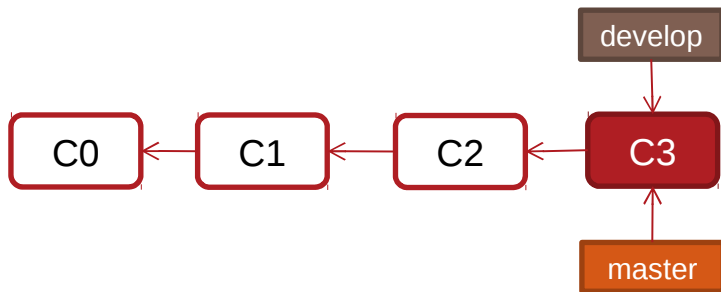
Merge (fast forward)



```
linux@iSA ~ $ git checkout master
linux@iSA ~ $ git merge develop
Updating 79d1630..9466f25
Fast-forward
 C | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 C
```

Scalanie

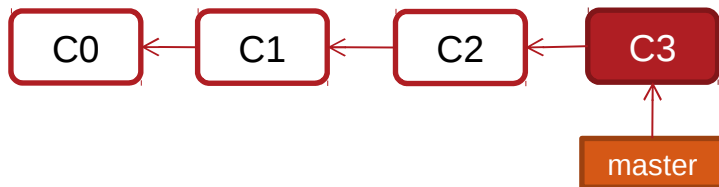
Merge (fast forward)



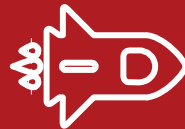
```
linux@iSA ~ $ git branch -D develop  
Deleted branch develop (was 9466f25).
```

Scalanie

Merge (fast forward)



ĆWICZENIE!



Przełącz się na branch *master*.

Utwórz plik A i commit z tytułem A oraz plik B oraz commit z tytułem B.

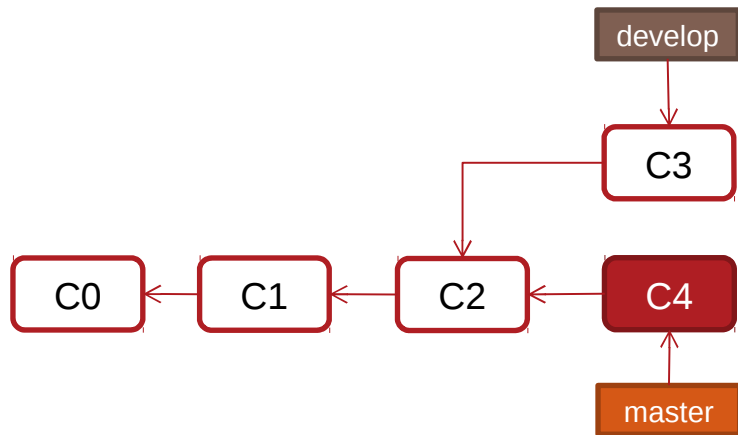
Przełącz się na branch *develop*, utwórz plik C i commit z tytułem C.

Przełącz się na branch *master*, utwórz plik D oraz commit z tytułem D.

Włącz branch *develop* do *mastera*.

Scalanie

Merge (3-way merge)



```
linux@iSA ~ $ git checkout master
```

```
linux@iSA ~ $ git merge develop
```

```
(...)
```

```
Merge made by the 'recursive' strategy.
```

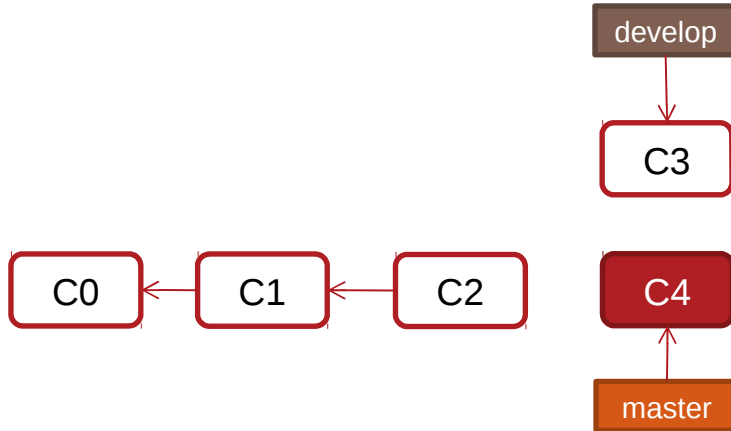
```
D | 0
```

```
1 file changed, 0 insertions(+), 0 deletions(-)
```

```
create mode 100644 D
```

Scalanie

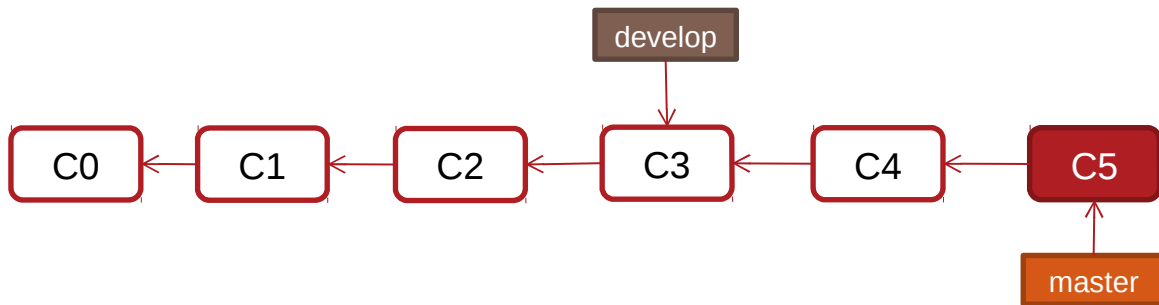
Merge (3-way merge)



```
linux@iSA ~ $ git checkout master
linux@iSA ~ $ git merge develop
(...)
Merge made by the 'recursive' strategy.
D | 0
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 D
```

Scalanie

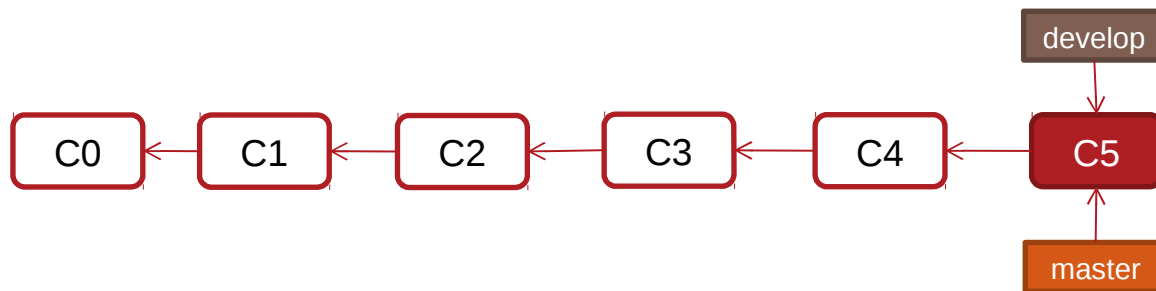
Merge (3-way merge)



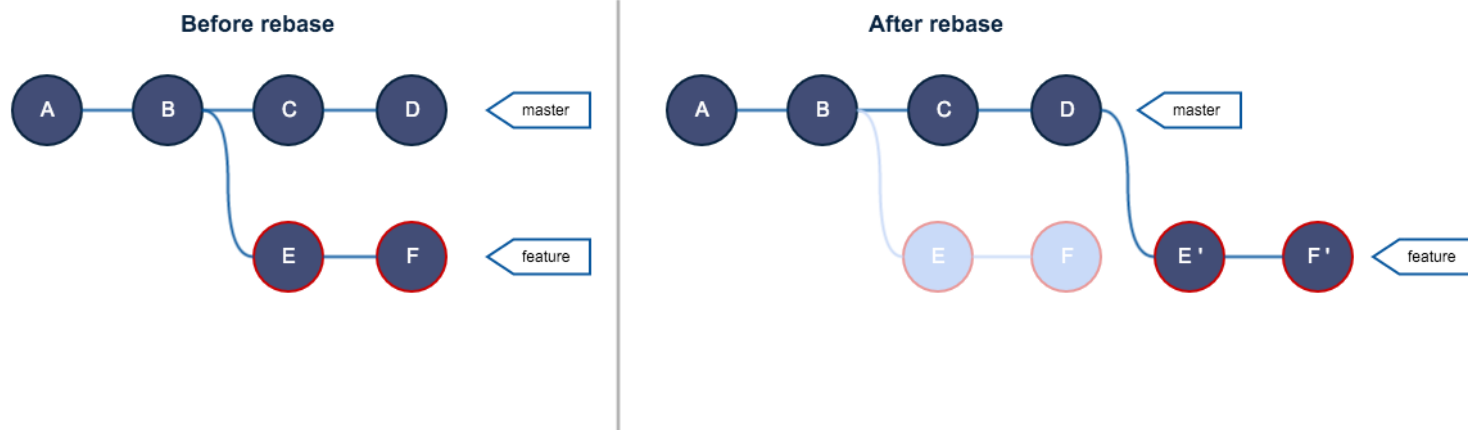
```
linux@iSA ~ $ git branch -D develop  
Deleted branch develop (was 15d7b46).
```

Scalanie

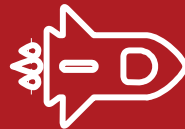
Merge (3-way merge)



Scalanie Rebase



ĆWICZENIE!



Utwórz nowe repozytorium – test6.

Utwórz plik A i commit z tytułem A oraz plik B oraz commit z tytułem B.

Przełącz się na branch *develop*, utwórz plik C i commit z tytułem C.

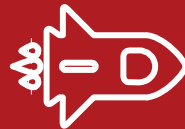
Przełącz się na branch *master*, utwórz plik D oraz commit z tytułem D.

Przełącz się znów na branch *develop*.

Przeprowadź rebase *developa* do *mastera*.

3. Konflikty

ĆWICZENIE!



Utwórz nowe repozytorium – test7.

Utwórz plik A i commit z tytułem A.

Przełącz się na branch *develop*, wpisz do pliku A treść: *Jestem develop*.

Zrób commit.

Przełącz się na branch *master*, wpisz do pliku A treść: *Jestem master*.

Zrób commit.

Przełącz się na *master* i scal do niego gałąź *develop*.

Konflikty

Skąd się biorą?

- Czasami algorytm nie potrafi sobie poradzić
- Nie należy bać się konfliktów!

```
linux@iSA ~ $ git merge develop
Auto-merging A
CONFLICT (content): Merge conflict in A
Automatic merge failed; fix conflicts and then commit the result.
cat A
<<<<<< HEAD
Jestem master
=====
Jestem develop
>>>>>> develop
```

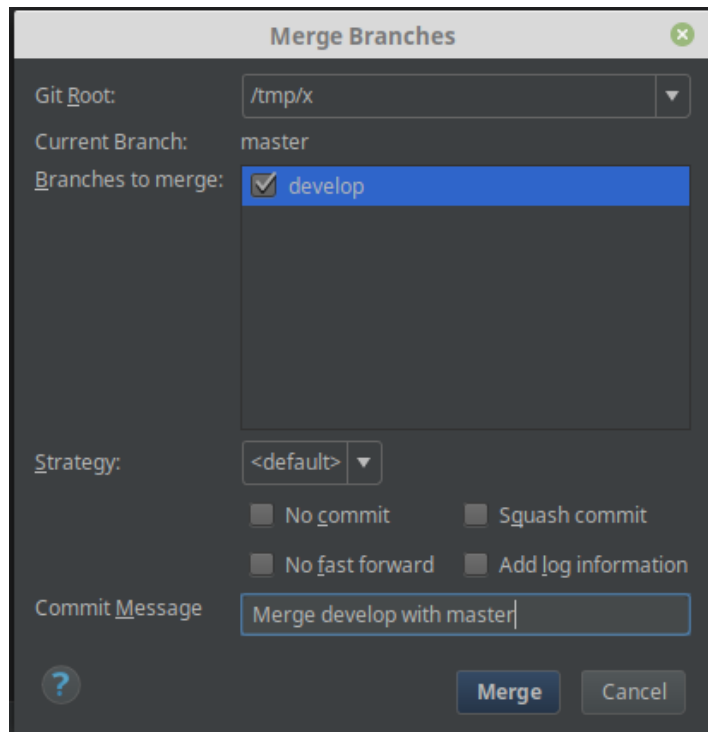
Konflikty

Rozwiązywanie

- Git listuje pliki z konfliktami
- W każdym pliku zapisywana jest wersja z obu branchy
- Należy usunąć konflikty i stworzyć nową zmianę

Konflikty

Rozwiązywanie

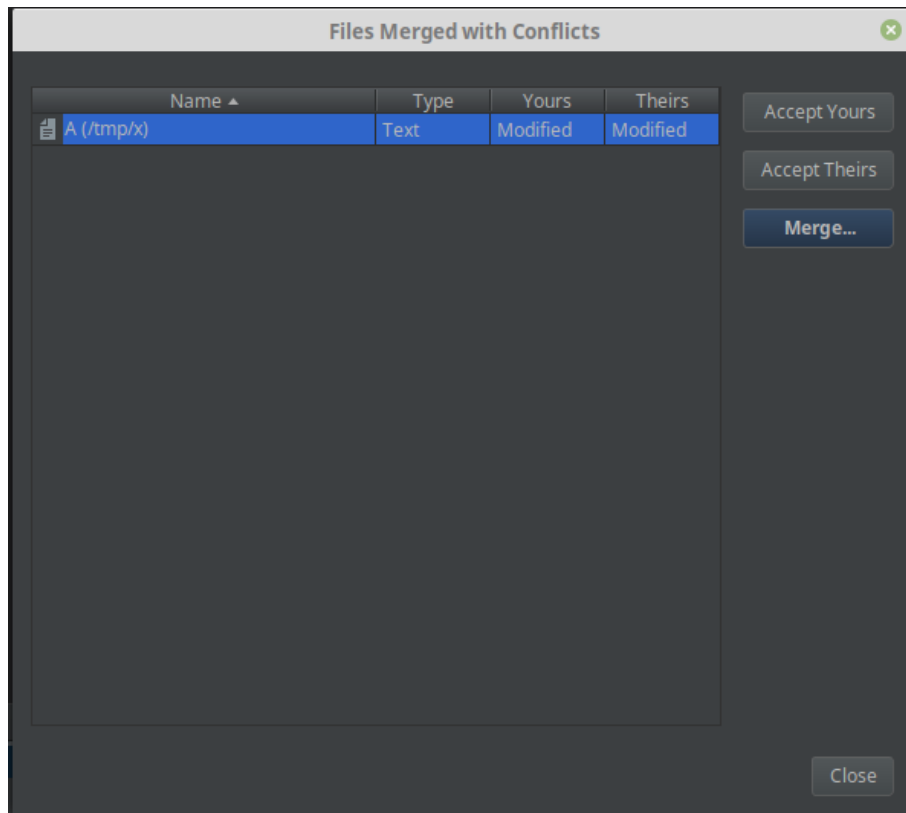


The screenshot shows a 'Merge Branches' dialog box with the following fields and options:

- Git Root:** /tmp/x
- Current Branch:** master
- Branches to merge:** A list containing 'develop' with a checked checkbox.
- Strategy:** <default>
- Options:**
 - ☐ No commit
 - ☐ Squash commit
 - ☐ No fast forward
 - ☐ Add log information
- Commit Message:** Merge develop with master
- Buttons:** A help icon (?), a 'Merge' button, and a 'Cancel' button.

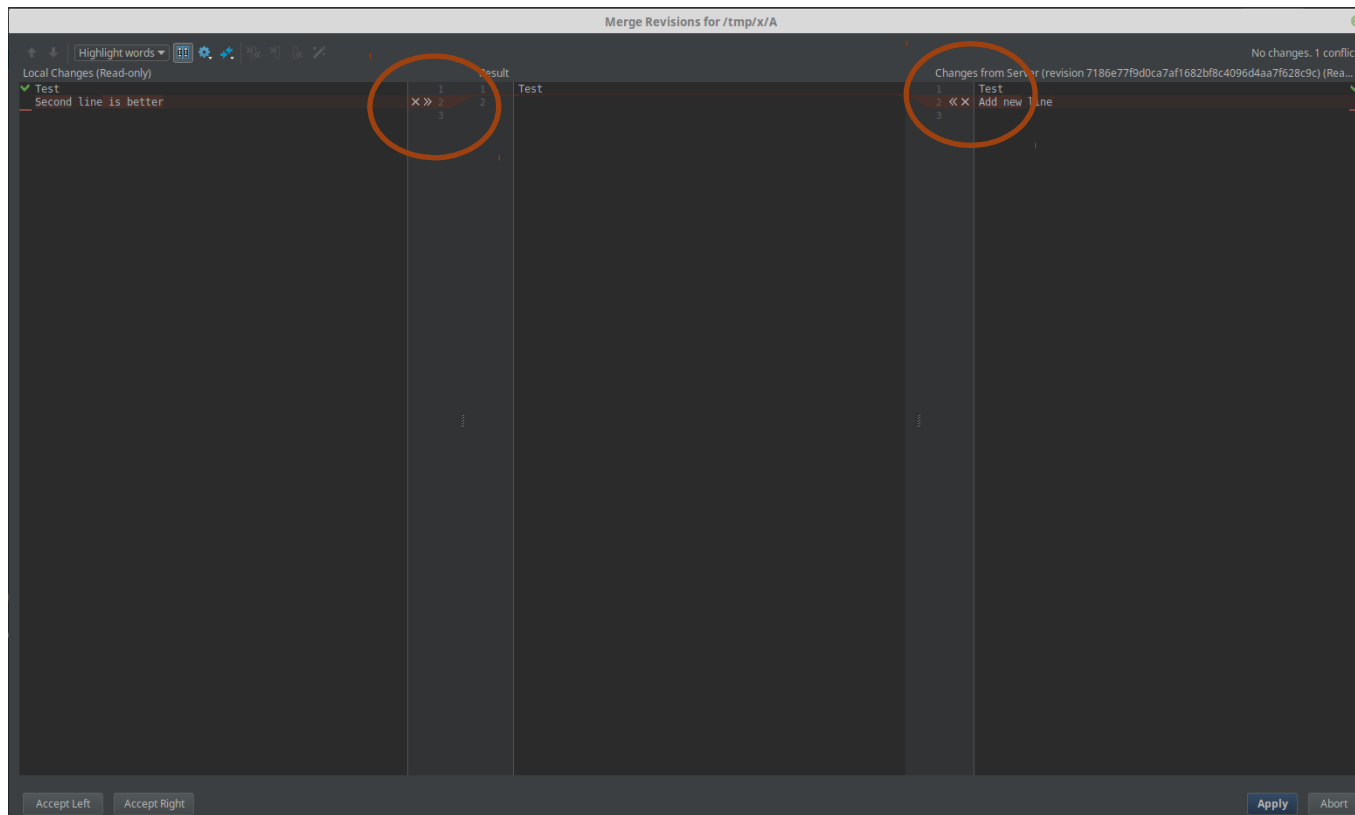
Konflikty

Rozwiązywanie



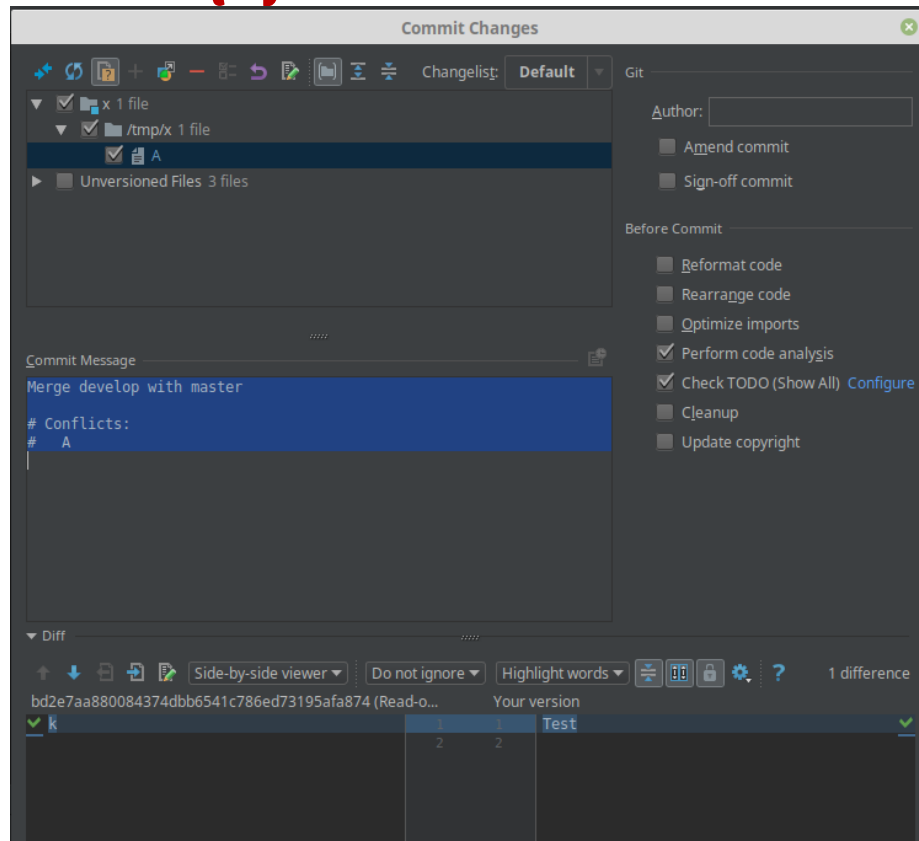
Konflikty

Rozwiązywanie



Konflikty

Rozwiązywanie



4. Git flow

Git flow

Po co w sieci?

- Idea branchów funkcjonalnych
- Utrzymanie porządku w repozytorium
- Sztywny model zarządzania gałęziami
- Zaprojektowany do regularnych release'ów

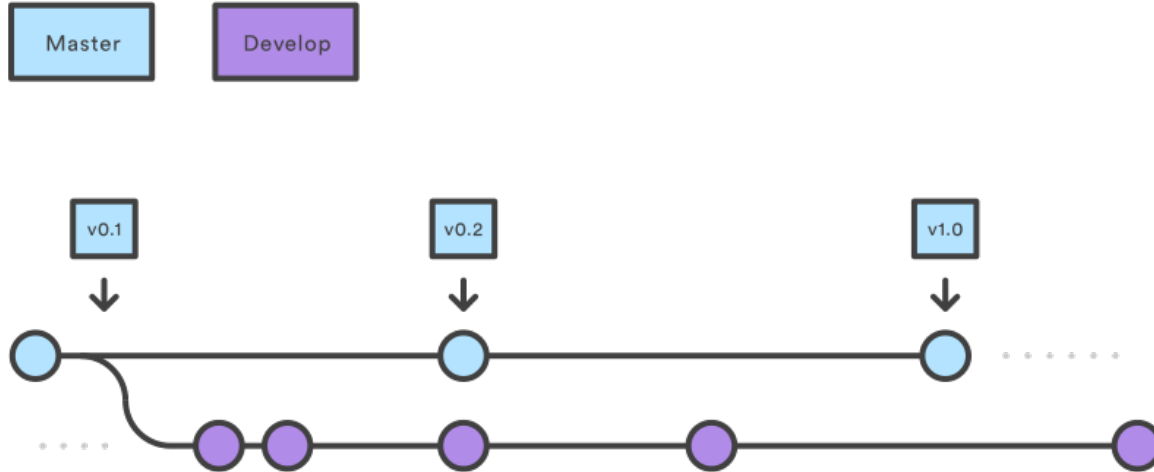
Git flow

Master i develop

- Dwie gałęzie do śledzenia historii
 - Master – master wydania produktu
 - Develop – integracja kodu zmian
- Zwykle brak możliwości przesyłania zmian do mastera

Git flow

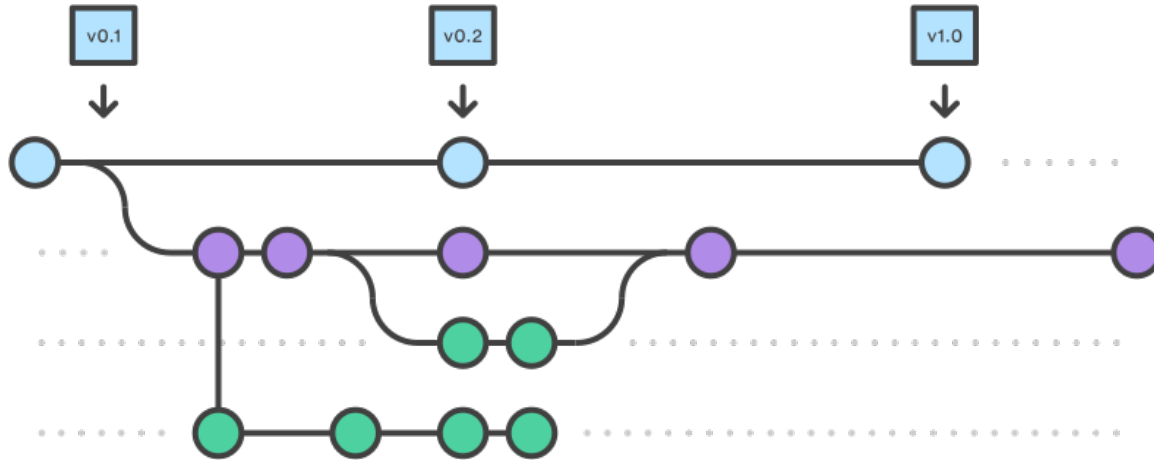
Master i Develop



```
linux@iSA ~ $ git branch develop  
linux@iSA ~ $ git push -u origin develop
```

Git flow

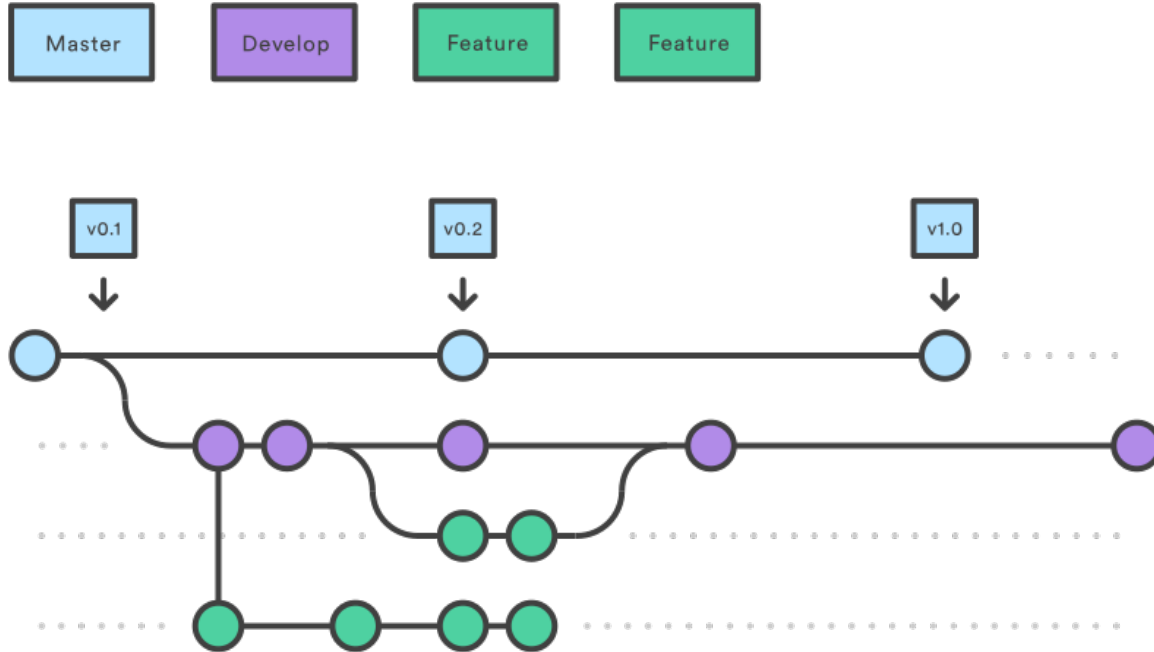
Branch: Feature (start)



```
linux@iSA ~ $ git checkout develop  
linux@iSA ~ $ git checkout -b feature_branch
```

Git flow

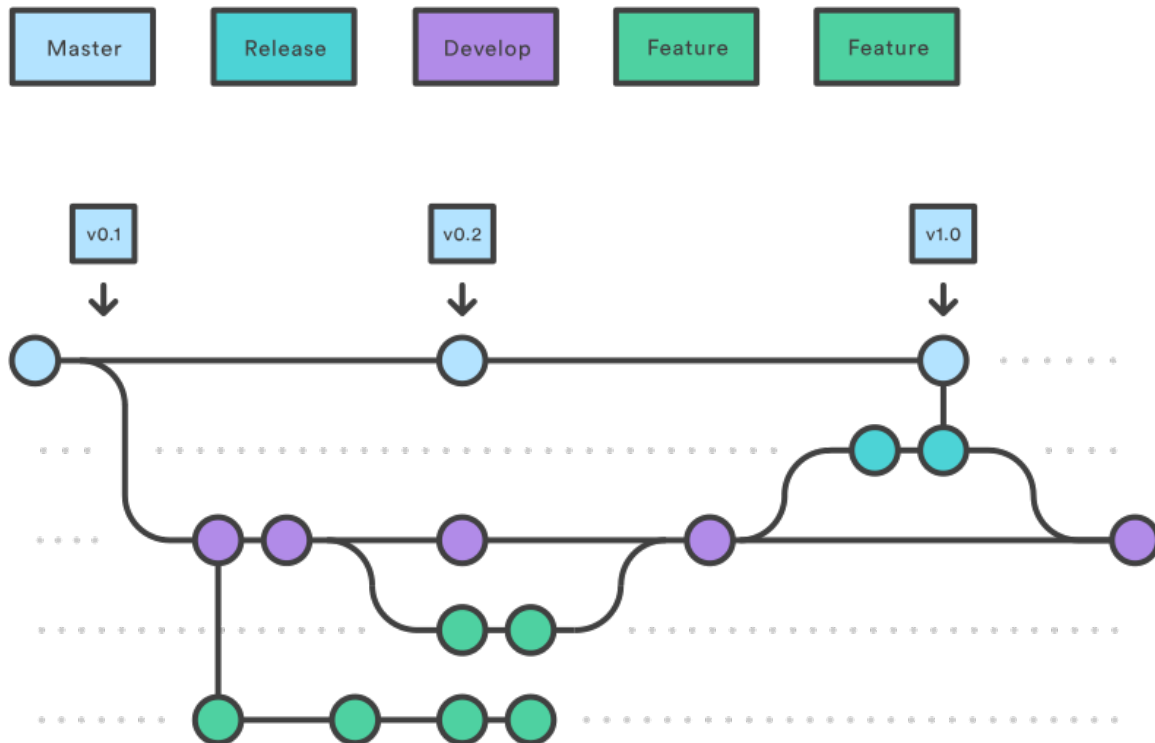
Branch: Feature (merge)



```
linux@iSA ~ $ git checkout develop  
linux@iSA ~ $ git merge feature_branch
```

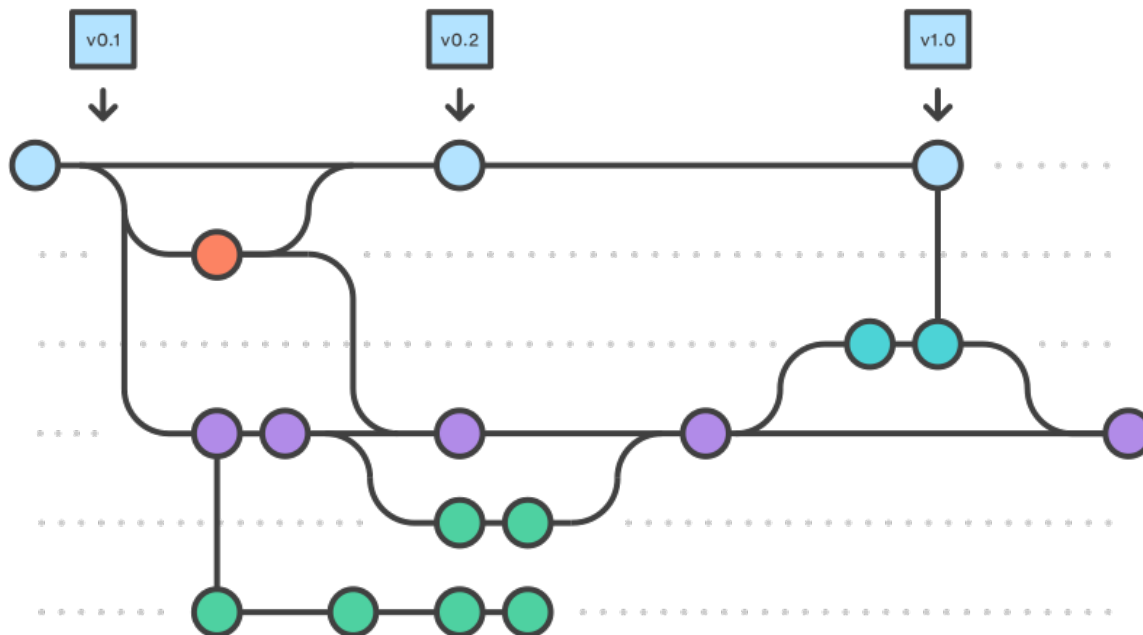

Git flow

Branch: Release



Git flow

Branch: Hotfix



5. Git w sieci

Git w sieci

Po co w sieci?

- Rozproszony Git
- Stały dostęp do kodu
- Pełna historia zmian
- Usługi hostujące GIT

Git w sieci

Gdzie trzymać kod?

	GitHub	Bitbucket	GitLab
Publiczne repa	Bez limitu	Bez limitu	Bez limitu
Prywatne repa	0	1 GB / projekt	10 GB / projekt
Użytkownicy	Bez limitu	Bez limitu	Bez limitu
Support	Email, Forum	Email, Forum	Forum
Powstanie	2008	2008	2011
Wspierane VCS	Git, SVN, HG	Git, SVN, HG	Git
Prywatne branche	NIE	TAK	TAK

Git w sieci

Gdzie trzymać kod?

- Każdy z wymienionych serwisów oferuje dodatkowo:
 - Wiki
 - Code review
 - Issue tracker

6. Pull request

Pull request

Idea pull requestów



Pull request

Idea pull requestów

- Wąskie grono osób z uprawnieniami zapisu
- Kontrola nad wchodzącymi zmianami
- Przeglądanie i ocena kodu
- Testy automatyczne

Pull request

Fork repozytorium

savonet / liquidsoap

Watch

57

Star

399

Fork

48

<> Code

Issues 105

Pull requests 7

Projects 1

Wiki

Insights

Audio and video streaming language <http://liquidsoap.fm/>

2,841 commits

19 branches

18 releases

15 contributors

GPL-2.0

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

rhizome and **toots** Reformat dependency lists (#644)

Latest commit 7f1368b 17 days ago

doc


Remove mentions of --dynamic-plugins-dir

2 months ago

Pull request

Fork repozytorium

 **dkossako / liquidsoap**
forked from [savonet/liquidsoap](#)

 Unwatch ▾

1

★ Star

0

 Fork


49

 Code

 Pull requests 0

 Projects 0

 Wiki

 Insights


 Settings


Audio and video streaming language <http://liquidsoap.fm/>

Edit

[Manage topics](#)

 2,841 commits

 19 branches

 18 releases

 15 contributors

 GPL-2.0

Branch: master ▾

New pull request

Create new file


Upload files

Find file

Clone or download ▾

This branch is even with savonet:master.

 Pull request  Compare

 **rhizome** and **toots** Reformat dependency lists ([savonet#644](#))

Latest commit 7f1368b 17 days ago

 [doc](#) Remove mentions of --dynamic-plugins-dir 2 months ago

 [examples](#) Update versions, use liquidsoap.info 2 months ago

Pull request

Wypchnięcie zmiany

- Kopia repo znajduje się na koncie użytkownika
- Możliwość dowolnego manewrowania zmianami
- Przygotowujemy zmianę i wypychamy do Githuba

```
linux@iSA ~ $ git clone https://github.com/dkossako/liquidsoap.git
linux@iSA ~ $ git checkout -b feature-branch
Switched to a new branch 'feature-branch'
linux@iSA ~ $ (...)
linux@iSA ~ $ git commit -a -m "Add some magic feature"
linux@iSA ~ $ git add && git commit
linux@iSA ~ $ git push origin feature-branch
remote: Create a pull request for 'feature-branch' on GitHub by visiting:
remote:   https://github.com/dkossako/liquidsoap/pull/new/feature-branch
To https://github.com/dkossako/liquidsoap.git
* [new branch]   feature-branch -> feature-branch
```

Pull request

Pierwszy pull request

savonet / liquidsoap
Watch 57
Star 399
Fork 49

Code
Issues 105
Pull requests 7
Projects 1
Wiki
Insights

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base fork: savonet/liquidsoap
base: master
head fork: dkossako/liquidsoap
compare: feature-branch

✓ **Able to merge.** These branches can be automatically merged.

Add some magic feature

Write
Preview
AA B i
“ ” < >
☰ ☷ ✓
@ 📌 ↶

Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.

☒ **Allow edits from maintainers.** [Learn more](#)

Create pull request

1 commit
1 file changed
0 commit comments
1 contributor

Pull request

Pierwszy pull request

- Pull request trafia do autora projektu źródłowego
- Od jego polityki zależy dalszy los zmiany
- Github pozwala prowadzić dyskusję nad PR

Pull request


Code review & Merge


Add some magic feature #650

Edit

 Open dkossako wants to merge 1 commit into `savonet:master` from `dkossako:feature-branch`

 Conversation 0

 Commits 1

 Checks 0

 Files changed 1


+0 -0



dkossako commented just now

+ 👤 ...

This is only test, ignore that.

 Add some magic feature

3d15950

Add more commits by pushing to the **feature-branch** branch on **dkossako/liquidsoap**.




Some checks haven't completed yet

[Hide all checks](#)

1 pending check



 **continuous-integration/travis-ci/pr** Pending — The Travis CI build is in progress

[Details](#)



This branch has no conflicts with the base branch

Only those with [write access](#) to this repository can merge pull requests.

Reviewers

No reviews

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Notifications



Koniec!

Dziękuję za uwagę.
Pytania?