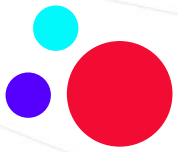


GIT – Part 3



HELLO

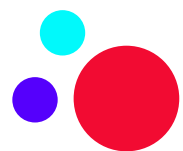
Damian Filipkowski





Agenda

1. Pull Requests
2. Dobre praktyki
3. Zaawansowany Git Flow



Pull Request

Powtórka

Pull Request



W uproszczonym ujęciu **Pull Request** jest mechanizmem powiadamiania członków zespołu przez programistę, że ukończył on jakąś część kodu i chce zmergować zmiany do brancha głównego. Bardzo często, zanim to się stanie jego kod przejdzie tzw. **Code Review**.

Technicznie, gdy branch na którym nowa funkcjonalność będzie gotowy, programista tworzy nowy pull request za pośrednictwem GIT GUI (GitHub, GitLab, BitBucket etc.)





Pull Request

 infoshareacademy / jpydzt4-materialy-podstawy-programowania

infoshareacademy / jpydzt4-materialy-podstawy-programowania

Q Type to search

>

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Filters Search all issues

Labels 9 Milestones 0

New pull request

Clear current search query, filters, and sorts

☐ 0 Open 8 Closed

Author Label Projects Milestones Reviews Assignee Sort

☐ prezentacje day_2 and day_3 added
#8 by kodyliszek was merged 13 minutes ago

☐ Mutable vs immutable
#7 by kodyliszek was merged 4 days ago

☐ instructions
#6 by kodyliszek was merged 5 days ago

☐ operacje na stringach
#5 by kodyliszek was merged 5 days ago

☐ reference_counting.py
#4 by kodyliszek was merged 5 days ago

☐ dodana algorytmy i skrypt
#3 by kodyliszek was merged 2 weeks ago

☐ dodana algorytmy i skrypt
#2 by kodyliszek was merged 2 weeks ago

☐ dodana prezentacja
#1 by kodyliszek was merged 2 weeks ago





Dobre praktyki – branche

- Używamy separatorów `optimize-data-analysis` or `optimize/data/analysis`
- Zaczynij od kategorii hotfix, bugfix, feature, test etc.
- Użyj numeru zadania
- Unikaj samych cyfr
- Unikaj długich nazw
- Niech nazwa coś znaczy
- Bądź spójny

<https://tilburgsciencehub.com/building-blocks/collaborate-and-share-your-work/use-github/naming-git-branches/>



Dobre praktyki – komentarze

- Tytuł i treść oddzielone pustym wierszem. (git log --oneline)
- Długość nie większa niż 50 znaków
- Tytuł pisany Wielką literą
- Brak kropki na końcu tytułu
- Użyj trybu rozkazującego w tytule
- Długość wiersza do 72 znaków
- Komentarz odpowiada na pytnia co i dlaczego. Na pytanie jak odpowiada kod.

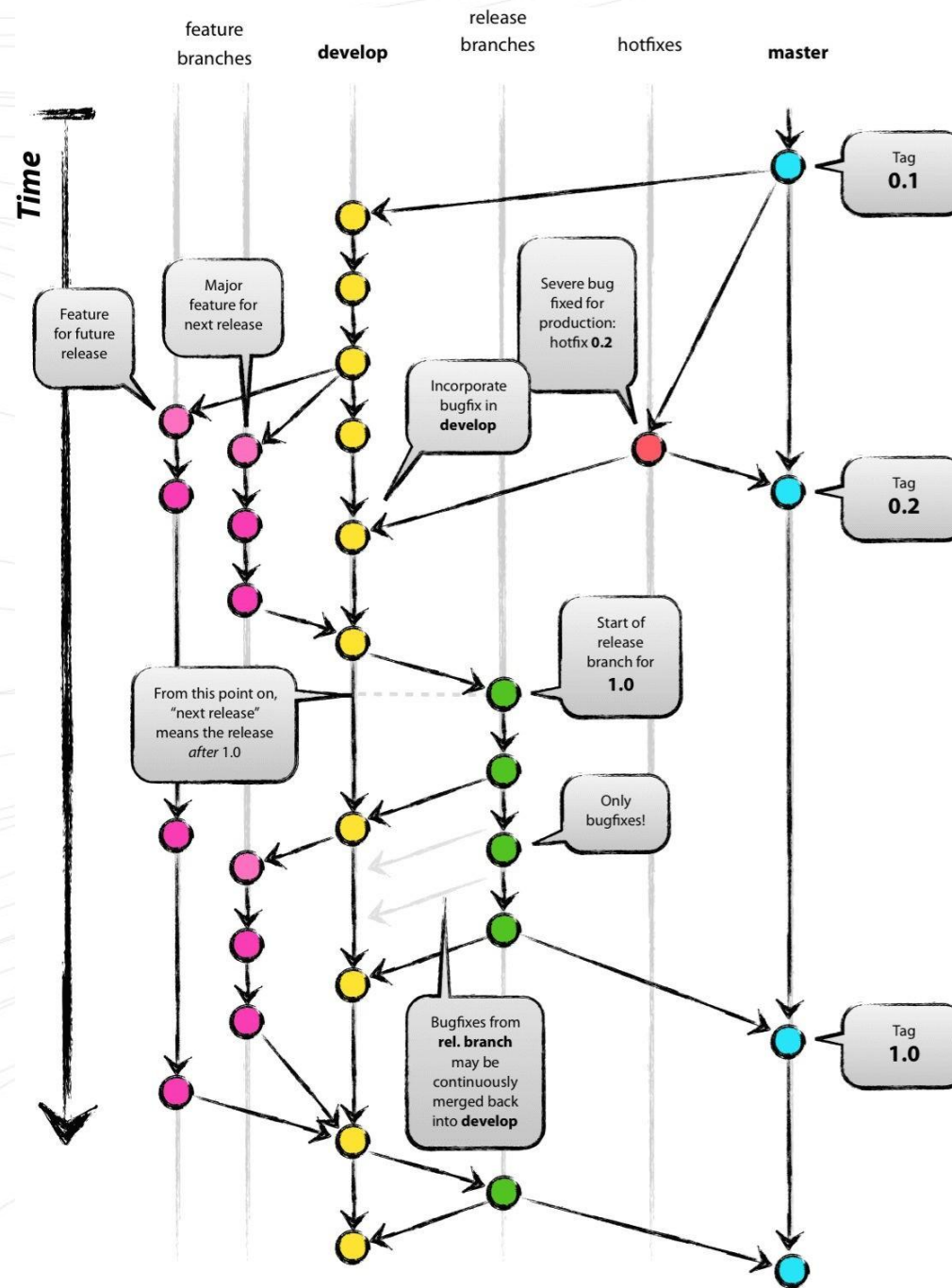
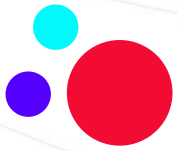
<https://cbea.ms/git-commit/#imperative>

https://wiki.openstack.org/wiki/GitCommitMessages#Information_in_commit_messages



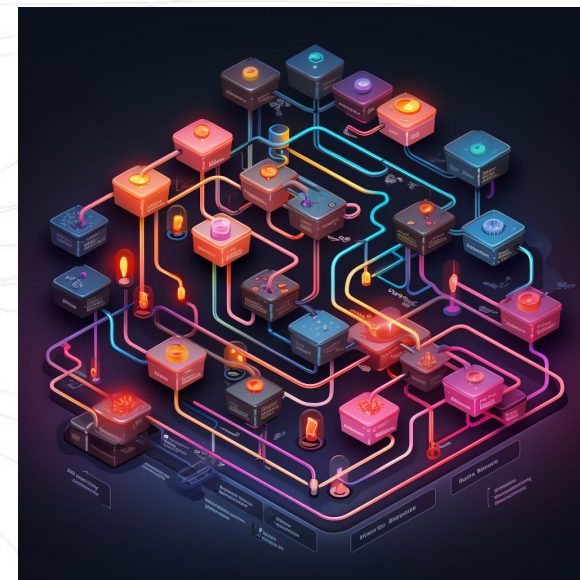
Zaawansowany GIT Flow

infoShare
ACADEMY



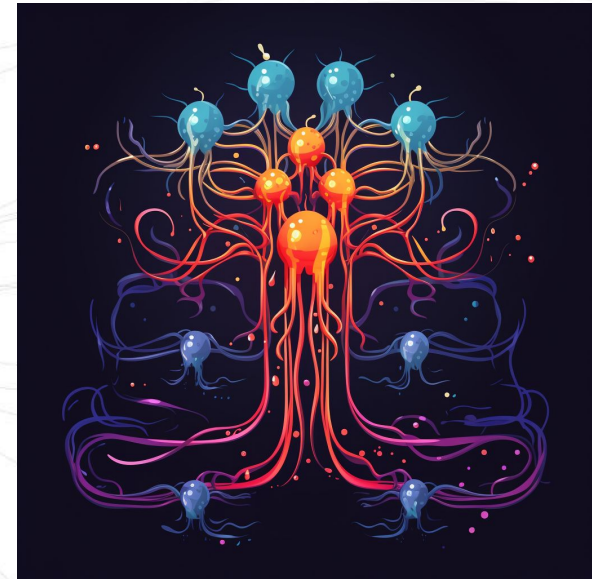
GIT Flow Quest – Pull Request

1. Sprawdź na jakim branchu jesteś
2. Przenieś się na branch **develop**
3. Stwórz branch **feature** z brancha **develop**
4. Przenieś się na branch **feature**
5. Dodaj plik **story.py** do repozytorium
6. Zrób kilka **commitów**
7. Wypchnij branch feature na **GitHuba** (remote)
8. Stwórz **Pull Request** twojego branch **feature** do **develop** i wyślij link na slacku na kanale merytoryka



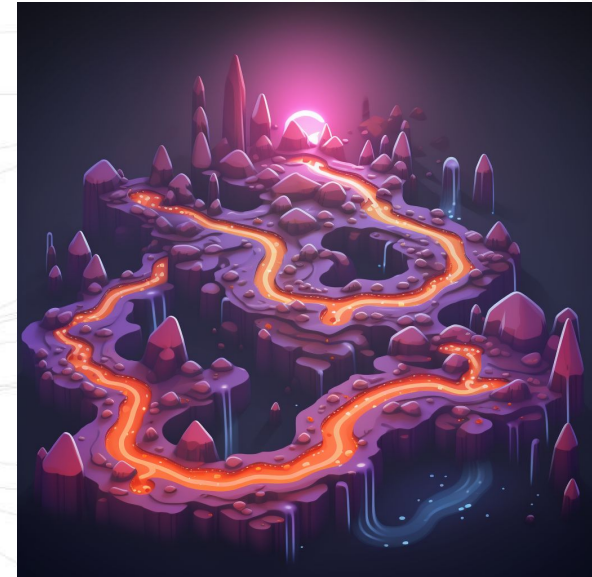
GIT Flow Quest – Release Branch

1. Sprawdź na jakim branchu jesteś
2. Przenieś się na branch **develop**
3. Stwórz **release** branch z brancha **develop**
4. Przenieś się na branch **release**
5. Dodaj zmiany w pliku **story.py**
6. Zrób kilka **commitów**
7. Zastanów się czym są te comity (diagram) ?
8. Co powinniśmy z nimi zrobić (prv slack) ?
9. Wypchnij release branch na **GitHuba** (remote)
10. Stwórz **Pull Request** twojego **release** branch do **master/main** i wyślij link na slacku na kanale merytoryka



GIT Flow Quest – Hot Fix

1. Sprawdź na jakim branchu jesteś
2. Przenieś się na branch **master/main**
3. Stwórz **fix** branch z brancha **master**
4. Przenieś się na branch **fix**
5. Napraw błąd w pliku **story.py**
6. Zrób **commit**
7. Wypchnij **fix** branch na **GitHuba** (remote)
8. Stwórz **Pull Request** twojego **release** branch do **master/main** i wyślij link na slacku na kanale merytoryka





Zestaw instrukcji rozszerzających GIT-a pozwalające na łatwiejsze zarządzanie flow repozytorium.

Narzędzia te można ściągnąć z: <https://github.com/nvie/gitflow>

Jak używać: <http://jeffkreeftmeijer.com/2010/why-arent-you-using-git-flow/>

Praca z feature branchem

git flow feature start MYFEATURE

Utworzy nowy branch MYFEATURE na bazie brancha develop oraz przełączy HEAD na nowo utworzonego brancha.

git flow feature finish MYFEATURE

Wykona merge do brancha develop, usunie brancha MYFEATURE oraz przełączy HEAD na branch develop

Praca z release branchem

git flow release start RELEASE [BASE]

Tworzy nowego brancha release
(jeśli podamy id commita w
miejscu base to od tego commita)
z gałęzi develop oraz przełączy
HEAD na gałąź RELEASE

git flow release finish RELEASE

Wykona merge do gałęzi develop i
master(main) przy czym podczas
merge'a do maina doda tag z nazwą
gałęzi, usunie branch RELEASE oraz
przełączy HEAD na develop

Praca z hotfix branchem

git flow hotfix start VERSION

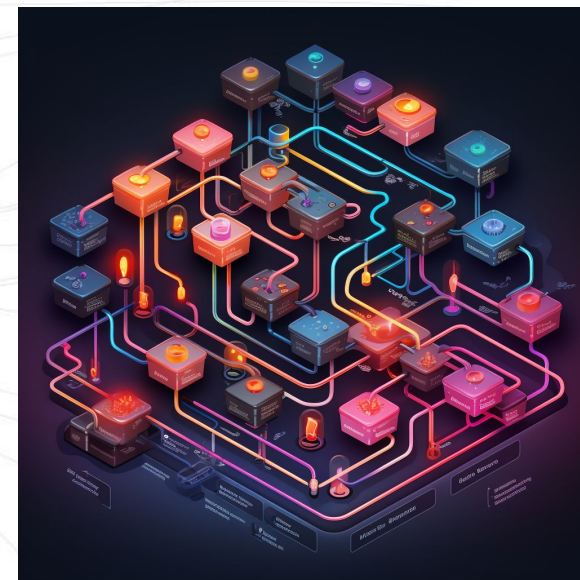
Utworzy nowego brancha VERSION i przełączy HEAD'a na niego.

git flow hotfix finish VERSION

Wykona merge do gałęzi develop i master(main) przy czym podczas merge'a do maina doda tag z nazwą gałęzi, usunie branch VERSION oraz przełączy HEAD na develop.

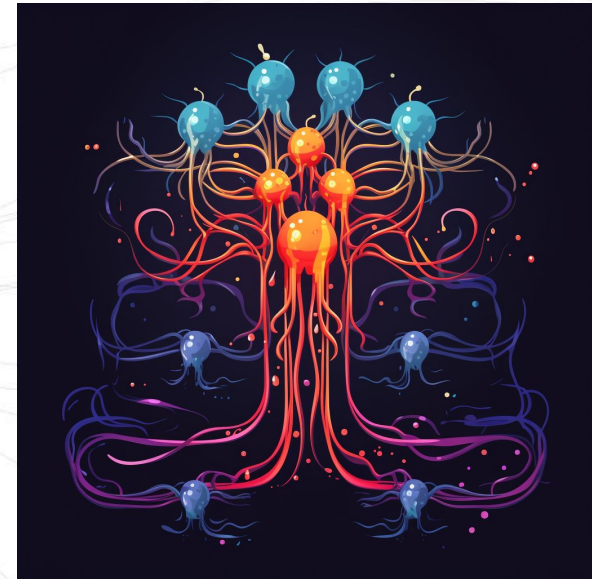
GIT Flow Quest – Pull Request

1. Sprawdź na jakim branchu jesteś
2. Przenieś się na branch **develop**
3. Stwórz branch **feature** z brancha **develop**
4. Przenieś się na branch **feature**
5. Dodaj plik **story.py** do repozytorium
6. Zrób kilka **commitów**
7. Wypchnij branch feature na **GitHuba** (remote)
8. Stwórz **Pull Request** twojego branch **feature** do **develop** i wyślij link na slacku na kanale merytoryka



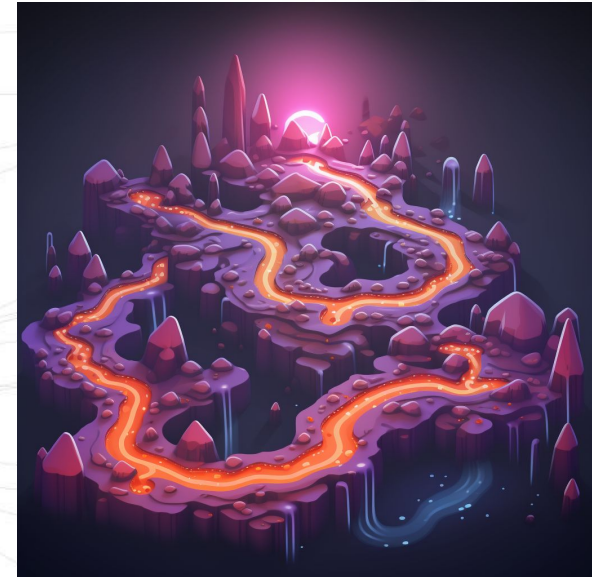
GIT Flow Quest – Release Branch

1. Sprawdź na jakim branchu jesteś
2. Przenieś się na branch **develop**
3. Stwórz **release** branch z brancha **develop**
4. Przenieś się na branch **release**
5. Dodaj zmiany w pliku **story.py**
6. Zrób kilka **commitów**
7. Zastanów się czym są te comity (diagram) ?
8. Co powinniśmy z nimi zrobić (prv slack) ?
9. Wypchnij release branch na **GitHuba** (remote)
10. Stwórz **Pull Request** twojego **release** branch do **master/main** i wyślij link na slacku na kanale merytoryka



GIT Flow Quest – Hot Fix

1. Sprawdź na jakim branchu jesteś
2. Przenieś się na branch **master/main**
3. Stwórz **fix** branch z brancha **master**
4. Przenieś się na branch **fix**
5. Napraw błąd w pliku **story.py**
6. Zrób **commit**
7. Wypchnij **fix** branch na **GitHuba** (remote)
8. Stwórz **Pull Request** twojego **release** branch do **master/main** i wyślij link na slacku na kanale merytoryka



**Dziękuję za
uwagę!**