

# GIT CODE REVIEW

**ZOPTYMALIZUJ SWÓJ FLOW SPRYTNYMI TRICKAMI,  
DOBRYMI PRAKTYKAMI I PRZYDATNYMI NARZĘDZIAMI**



# TOMASZ SKRASKOWSKI



- Backend developer at Pirios S.A.
- Writes code since 12
- Currently .Net (Core), previously C++
- Interested in decentralized systems
- [www.SoftwareDeveloper.Blog](http://www.SoftwareDeveloper.Blog)

# KRZYSZTOF MORCINEK



- [@ Pragmatic Coders](#)
- 10 years in .NET stacks
- Last year in JAVA
- [krzysztofmorcinek.wordpress.com](http://krzysztofmorcinek.wordpress.com)
- Screencast: [Show me the code](#)

# GITWARSZTATY W TWOJEJ FIRMIE



[www.GitWarsztaty.pl](http://www.GitWarsztaty.pl)

# AGENDA:

Pytajcie w trakcie na chacie - odpowiemy async po skończonych sekcjach

- Powtarzające się uwagi - automatyzować, a może nie reagować?
- Jak unikać flame Wars?
- Kod jest !#@\$ - jak dawać konstruktywny feedback.
- Kod działa, ale nie jest umieszczony w odpowiednich "miejscach". W jakich warunkach taki kod przepuścić?

# AGENDA CD.

- Kandydat na nie robienie: <!--Zła kolejność argumentów - istotna czytelność, czy fanaberia?-->
- TLDR - za długie CR czy może coś, co trzeba bez CR "przepchnąć"?
- Jak namawiać do braku CR?
- Konieczny refactoring w trakcie feature'a - jak to reviewować?
- Przekrój CI: od startu projektu do code review

# NASZ KONTEKST

*it's better to ask forgiveness than permission*

projekty gdzie dostarcza się business value, a nie security critical - różne części kodu będą miały różne wymagania w Code Review

zespoły same się organizują


# POWTARZAJĄCE SIĘ UWAGI

9 + }

10 +

11 + `changeToBetterProduction(provinceName: string, pattern: BuildingPattern)`


12 + {

**Tometchy** @tometchy · 10 seconds ago

Maintainer ✓ 😊 💬 ✎ ⋮


Przenieś klamrę do wyższej linii

▼ Collapse replies

**Krzysztof Morcinek** @krzysztof.morcinek · just now


Maintainer ✓ 😊 ✎ ⋮

ok



Reply...

Resolve thread ↻



Start a new discussion...

13 + `this.clicker.changeProvinceProduction(provinceName, pattern.changeTo);`

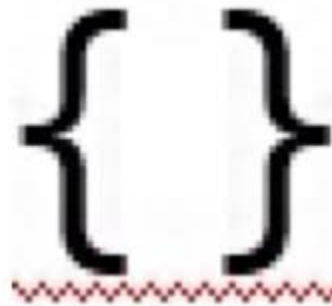
14 + }

15 + }



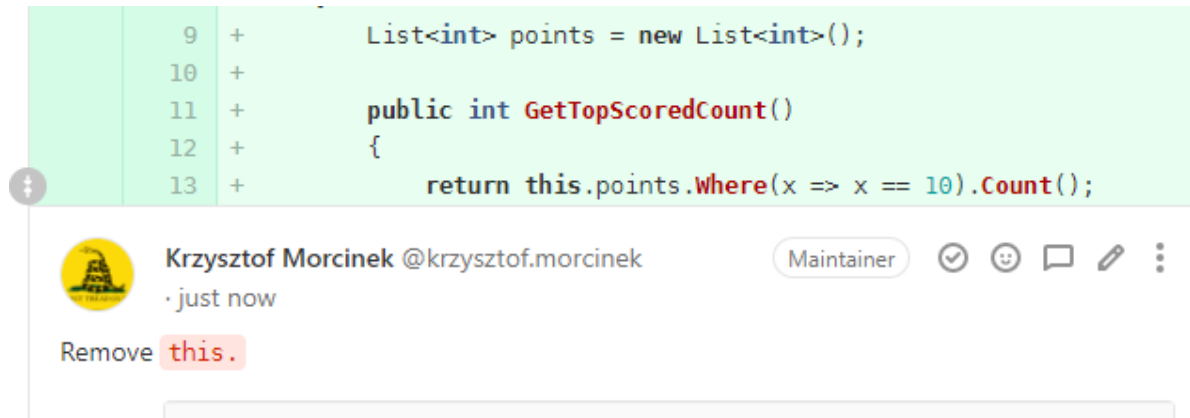
DEMO

# STYLECOP (PRETTIER.IO, CHECKSTYLE)



37	66		protected UIImage LoadPdfPage(int pageIndex)
38		-	{
	67	+	{
39	68		if (Images[pageIndex] == null)
40	69		{

# STYLECOP CD.



PrefixLocalCallsWithThis - przykład że coś trzeba wybrać i się stosować.

... w dowolną stronę (z **thisem**, albo bez **thisa**)

# UNIWERSALNE NARZĘDZIE DO FORMATOWANIA

[prettier.io](https://prettier.io)

Kolega ustawił indentation na 2 spacje - nigdy w życiu nie miałem takiego ustawienia. Ale dzięki temu że jest to wymuszone to nie narzekam i się przyzwyczaiłem (choć przeformatowało cały projekt)

# TESTY KONWENCJI

3 kawałki:

- pętla po wszystkich plikach w projekcie
- szukanie problemów w jednym pliku
- odpalenie tych *testów* podczas buildu

Przykład: wymóg aby wszystkie kolekcje były niemutowalne  
w C# (*ICollection<T>*)


## **POWTARZAJĄCE SIĘ UWAGI**

- automatyzować (jeśli się da)
- może nie reagować

# JAK UNIKAĆ FLAME WARS

39 +

if (productionGoal !== null)


**Tometchy** @tometchy · 23 hours ago

Maintainer

✓ 😊 💬 ✎ ⋮

Przenieś klamrę do poprzedniej linii (już ostatnio było to ustalone)


▼ Collapse replies

**Krzysztof Morcinek** @krzysztof.morcinek · 23 hours ago

Maintainer

✓ 😊 ✎ ⋮


ale nie wszędzie tak jest ...

**Tometchy** @tometchy · 23 hours ago

Maintainer

✓ 😊 ✎ ⋮


taką mamy konwencję, nowy kod jest tylko tak pisany

**Krzysztof Morcinek** @krzysztof.morcinek · 22 hours ago

Maintainer

✓ 😊 ✎ ⋮


ale ostatnio Marcin wrzucił i tak nie było

**Tometchy** @tometchy · 22 hours ago

Maintainer

✓ 😊 ✎ ⋮


jak to nie było, to może przeoczenie jakieś i też trzeba poprawić

**Krzysztof Morcinek** @krzysztof.morcinek · 22 hours ago

Maintainer

✓ 😊 ✎ ⋮


ja nie będę po Marcinie poprawiał. Jego przeszedł a ja od niego kopiowałem. Ja już tutaj chcę mergować.

**Tometchy** @tometchy · 22 hours ago

Maintainer

✓ 😊 ✎ ⋮

...

**Krzysztof Morcinek** @krzysztof.morcinek · 22 hours ago

Maintainer

✓ 😊 ✎ ⋮

...

# TABS VS SPACES



Wy sami wiecie co odpuścić. Zachęcam żeby czasem odpuszczać.



# **KOD JEST !#@ \$ - JAK DAWAĆ KONSTRUKTYWNY FEEDBACK**

Nie oceniamy osoby, merytorycznie opisujemy problem.

Proponujemy możliwe rozwiązania.

W CR nie tylko doszukujemy się potencjalnych problemów,  
dostrzegamy i doceniamy również to co ktoś napisał  
elegancko ;)

# KOD DZIAŁA, ALE NIE JEST UMIESZCZONY W ODPOWIEDNICH "MIEJSCACH". W JAKICH WARUNKACH TAKI KOD PRZEPUŚCIĆ?

Czy wymóc dostosowanie się? Jak zwykle - to zależy.  
Jaki jest impact?

Najlepiej dla konwencji również mieć przygotowany automat, np. obiekty domenowe muszą być w projekcie kończącym się na *Domain*.

# TLDR - ZA DŁUGIE CR CZY MOŻE COŚ, CO TRZEBA BEZ CR "PRZEPCHNAĆ"?


- olbrzymi rename - nikt nie chce tego przeglądać
- olbrzymi rename i gdzieś logika się zmieniła




# **OCZYSZCZENIE COMMITU (COMMITÓW) PRZED CR**

- Fix typo
- Poprawa formatowania
- Zmiana namespace'u / folderu / pakietu


Set "noImplicitAny"

 kmorcinek committed 19 hours ago

Remove not needed casting to "any"


 kmorcinek committed 20 hours ago

Verify newProvince in history has the same resources


 kmorcinek committed 20 hours ago

Commits on Dec 1, 2019


Run app from js script in bookmark #17

 kmorcinek committed 2 days ago ✓

Clean up README

 kmorcinek committed 2 days ago


Bump version

 kmorcinek committed 2 days ago ✓


Add FarmPrediction and use in HUD #1

 kmorcinek committed 2 days ago


Add FarmHistoryChecker and predict next farm turn #1

 kmorcinek committed 3 days ago

Add IPrediction


 kmorcinek committed 2 days ago ✓

Turn starts at 1

 kmorcinek committed 2 days ago ✓


Add ProvinceHistory

Add ProvinceHistory

 kmorcinek committed 2 days ago ✓

Commits on Nov 30, 2019

Extend max line to 120

 kmorcinek committed 3 days ago ✓

# JAK NAMAWIAĆ DO BRAKU CR?

Przykłady gdzie warto się zastanowić

- jakiś projekt na wymarcu gdzie nie chcemy inwestować swojego czasu
- 1 osoba w nowym projekcie i nikt nie kwapi się do przeglądania (jeśli chodzi o dzienne review)
- ... !!! ale z pewnością co jakiś czas ktoś powinien doglądać w którą stronę to ogólnie idzie
- 2 osoby które ze sobą rozmawiają
- jeśli mam przeglądać ogromny MR TechLeada, który zmienia coś we frameworku (co mnie nie dotyczy)

# WIĘC JAK NAMAWIAĆ DO BRAKU CR?

Przemyśleć sobie czy da nam to więcej zysku niż zainwestowany w to czas.

- zakładam że proces jest w rękach zespołu i możemy to sobie zmienić
- ... albo robić wyjątki
- może się okazać, że po pewnym czasie zespół się dociera i już na CR nie są wyłapytawne kardynalne błędy
- ... decyzje odnośnie architektury są wystarczająco dobrze przegadane przy kawach i kod powstaje w odpowiednich miejscach



**KONIECZNY REFACTORING W TRAKCIE  
FEATURE'A - JAK TO REVIEWOWAC?**

**DEMO**

# PORÓWNANIE DWÓCH PODEJŚĆ

	Wystawiający		Code-Review'ujący	
	Wady	Zalety	Wady	Zalety
nie rozdzielam MR (1 MR)		szybko, mało dodatkowej pracy, wszystko wypycham razem	gdy zmian jest dużo, to refaktoring zaciemnia zmiany dodające nowego feature	
rozzdzielam MR (2 MR)	więcej pracy, trzeba przygotować dwa osobne Mry			mogę się skupić tylko na nowym feature oraz w osobym CR tylko na refakroringu

# PORÓWNANIE DWÓCH PODEJŚĆ CD.

	Wystawiający		Code-Review'ujący	
	Wady	Zalety	Wady	Zalety
nie rozdzielam MR (1 MR)		mniej pracy	więcej trudnej pracy	
rozdzielam MR (2 MR)	więcej pracy			mniej łatwej pracy

# OD STARTU PROJEKTU DO CODE REVIEW



# WYBÓR FLOW COMMITÓW

😊 Jeden branch długo-żyjący

😞 Wiele branchy długo-żyjących

# WYBÓR FLOW DOŁĄCZANIA FEATURE BRANCHA DO CORE BRANCHA

- Basic merge
- Squash merge
- Rebase and fast-forward
- Rebase with merge commit

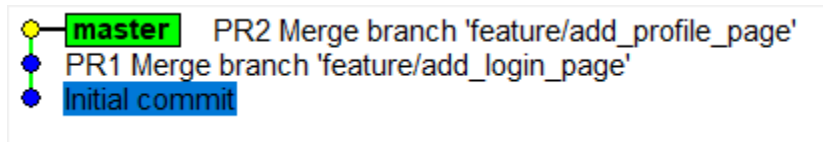
# BASIC MERGE

Da się wyciągnąć informację z historii, ale jest to trudne, czasami bardzo trudne.



# SQUASH MERGE

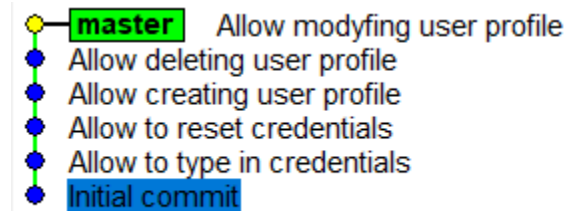
Gdy feature branche są bardzo małe, da się coś wyciągnąć z historii, ale stosunkowo niewiele.  
Ważna wiedza spoczywa poza repo - w zewnętrznym narzędziu.





# REBASE AND FAST-FORWARD

Historia jest przejrzysta, można wyciągnąć z niej bardzo dużo.  
Tracimy informacje o pullrequestach i 'zarys', które commity były w ramach jednego feature brancha.



# REBASE WITH MERGE COMMIT

Historia jest przejrzysta, można wyciągnąć z niej wszystko.

## Flow: Rebase with merge-commit

```
tom@DESKTOP-6VOJSCB MINGW64 ~/Downloads/temp (master)
$ git log --oneline --graph
* f2d13ff (HEAD -> master) PR2 Merge branch 'feature/add_profile_page'
| \
| * b7bb118 Allow modifying user profile
| * d62e375 Allow deleting user profile
| * d36af9d Allow creating user profile
| /
* 70fcc86 PR1 Merge branch 'feature/add_login_page'
| \
| * 4e3e399 Allow to reset credentials
| * 8a796c2 Allow to type in credentials
| /
* 9ebcc7a Initial commit
```

Widok pełnej historii- jakie commity realizowały jaki feature branch i w jakim PR można doczytać więcej informacji.

```
tom@DESKTOP-6VOJSCB MINGW64 ~/Downloads/temp (master)
$ git log --oneline --no-merges
b7bb118 Allow modifying user profile
d62e375 Allow deleting user profile
d36af9d Allow creating user profile
4e3e399 Allow to reset credentials
8a796c2 Allow to type in credentials
9ebcc7a Initial commit
```

Widok tylko commitów 'roboczych'.  
Efekt identyczny jak w 'Rebase and ff' flow.

```
tom@DESKTOP-6VOJSCB MINGW64 ~/Downloads/temp (master)
$ git log --oneline --merges
f2d13ff (HEAD -> master) PR2 Merge branch 'feature/add_profile_page'
70fcc86 PR1 Merge branch 'feature/add_login_page'
```

Widok tylko merge commitów.  
Efekt identyczny jak w 'Squash' flow.

```
tom@DESKTOP-6VOJSCB MINGW64 ~/Downloads/temp (master)
$ |
```

# WYBÓR POLITYKI OCHRONNEJ DLA CORE BRANCHA



# BLOKADA BEZPOŚREDNICH PUSHY DO CORE BRANCHA

Wszelkie zmiany możliwe tylko przez PR.

Polecane nawet, gdy pracujesz z samymi seniorami-wymiataczami,  
a nawet gdy prowadzisz dany projekt samemu.



# BRANCH POLICY - SPOSÓB DOŁĄCZANIA FEATURE BRANCHA

## ☒ **Limit merge types**

Control branch history by limiting the available types of merge when pull requests are completed.

Allowed merge types:

- ☐ Basic merge (no fast-forward)  
Preserves nonlinear history exactly as it happened during development.
- ☐ Squash merge  
Creates a linear history by condensing the source branch commits into a single new commit on the target branch.
- ☐ Rebase and fast-forward  
Creates a linear history by replaying the source branch commits onto the target without a merge commit.
- ☒ Rebase with merge commit  
Creates a semi-linear history by replaying the source branch commits onto the target and then creating a merge commit.

# BRANCH POLICY - BUILD PIPELINE

## Add build policy

Build pipeline \*

Frontend build

ⓘ No build pipelines were found

Path filter (optional) ⓘ It's worth to exclude triggering when changed files don't touch given build pipeline

/src/frontend

Trigger

Build starts automatically

☒ Automatic (whenever the source branch is updated)

☐ Manual

Policy requirement

☒ Required Green build required


Build must succeed in order to complete pull requests.


☐ Optional

Build failure will not block completion of pull requests.

Build expiration

Build invalidated when code changes

☒ Immediately when  master is updated

☐ After  hours if  master has been updated

☐ Never

Display name

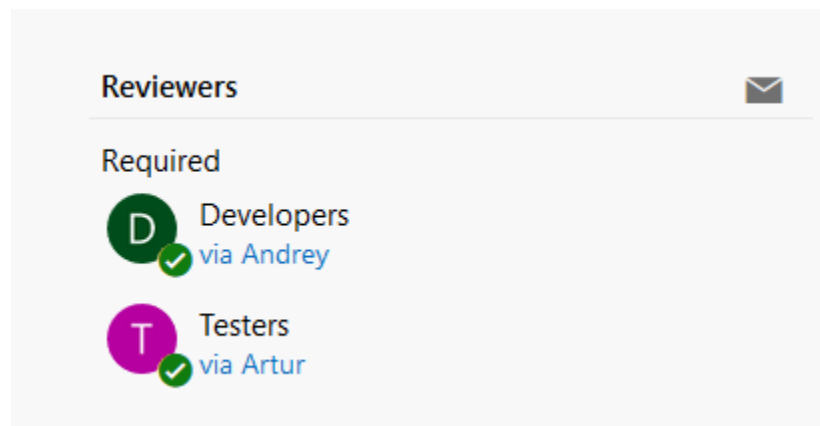
Frontend build

SaveCancel

# BRANCH POLICY - REQUIRED APPROVE

Minimum 1 wymagany approve (ale nie kliknięty przez siebie samego).

Ja polecam: wymagany 1 approve od kogoś z grupy *Developers* i 1 approve od kogoś z grupy *Testers*.




# BRANCH POLICY - ADDITIONAL APPROVE FOR SPECIFIC PROJECT PARTS

Gdy zmienione są pliki ze wskazanej ścieżki, wymagany dodatkowy approve od użytkownika ze specyficznej grupy/roli. Przykładowo:

- Dla ścieżki `src/data-processing/*` wymagany dodatkowy approve z grupy *Performance specialist*
- Dla ścieżki `src/core-engine/*` wymagany dodatkowy approve z grupy *Senior developer*
- Dla ścieżki `src/ssl-library/*` wymagany dodatkowy approve z grupy *Security specialist*



# BRANCH POLICY - ALL COMMENTS RESOLVED



**TS** **saga.ts** [View original diff](#)

21.04.2020

/src/WebClient/ElevenQ.WebClient/ClientApp/src/containers/PatientsMonitoring/saga.ts

^ 45 +

46 +     const latestTimeStamp = conversations.reduce((previousValue, currentValue) => {

47 +         const currentDate = new Date(+currentValue.timeStamp);

48 +         return currentDate > previousValue ? currentDate : previousValue;

49 +     }, new Date(0));

50 +

51 +     return { conversations, timestamp: latestTimeStamp };

 **Przemysław** 21.04.2020

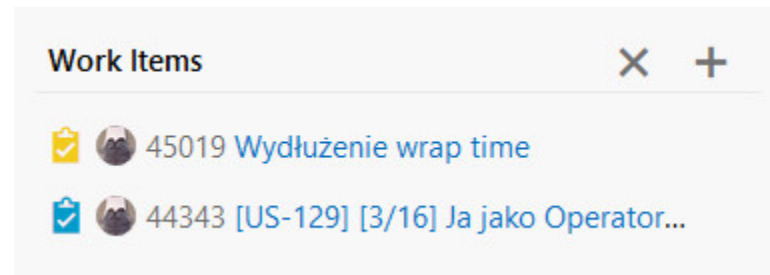
Active ▾

no dobra a jak pojdzie get i przyjdzie event , po drodze sie cos powyprzedza w rezultacie moga aupdatey nalozyć sie nie wtej kolejności. Nie wiem czy informacja o wersji stanu tez nie będzie potrzebna w ewencie integracyjnym.


 Write a reply...


Resolve





# BRANCH POLICY - LINKED WORK ITEM



# PRZYKŁADOWY PR Z BRANCH POLICY

 2202 COMPLETED Merge pr\_ts/add\_graphql\_to\_api to master



 Tomasz [pr\\_ts/add\\_graphql\\_to\\_api](#) into [master](#)


Delete source branch ...


[Overview](#) [Files](#) [Updates](#) [Commits](#)


Tomasz completed the pull request with a squash merge on 27.06.2019 11:32.



[dd54c995](#)   Merged PR 2202: Merge pr\_ts/add\_graphql\_to\_api to master...

Description


Show everything 

 Add a comment...

 Tomasz completed the pull request 27.06.2019

 Approved by  Artur 27.06.2019

2



 Tomasz pushed 1 commit creating [update 2](#) 27.06.2019

Policies

Required



- ✓ 2 reviewers approved
- ✓ Required reviewers have approved
- ✓ Work items linked
- ✓ All comments resolved
- ✓ [Build succeeded](#)
- ✓ [Build succeeded](#)

Work Items × +

  31705 Wystawienie api graphql

Reviewers ✉

Required

-  Developers [via Andrey](#)
-  Testers [via Artur](#)

Optional

# ZACZYNAMY NOWY FEATURE

1. Tworzymy i pushujemy nowego feature brancha.
2. Tworzymy nowy draft pull request, który będzie na bieżąco w trakcie pracy raportował nam stan z CI.
3. Pracujemy tworząc i pushując małe commity często. Akceptowalne WIP/TEMP commity i push --force-with-lease. Często też robimy `git fetch && git rebase origin/master`.
4. Porządkujemy historię, układamy commity w odpowiedniej kolejności, łączymy gdy trzeba, usuwamy niepotrzebne.
5. Opublikowujemy drafta jako czekającego pull requesta i informujemy zespół.

# OPUBLIKOWUJEMY PULL REQUESTA

1. Upewniamy się, że description PR wprowadza reviewującego w temat
2. Upewniamy się, że nie ma konfliktów lub czerwonych buildów
3. Po raz ostatni przeglądamy diffa wszystkich plików
4. Po raz ostatni przeglądamy listę commitów, które udostępniamy
5. Jeżeli serwer to umożliwia, ustawiamy automat do zamknięcia PR



Tomasz set the pull request to automatically complete when the

Cancel auto-complete

following policies succeed:

Minimum number of reviewers

**0 of 2 reviewers approved**

Required reviewers

**Required reviewers have not approved**

Work item linking

**No work items linked**

Build backend

**Build queued**

Build frontend

**Build queued**

Changes will be **squashed** into master. Branch feature/login\_form will be deleted. Work items will be **unchanged**.

# ZACZYNAMY CODE REVIEW

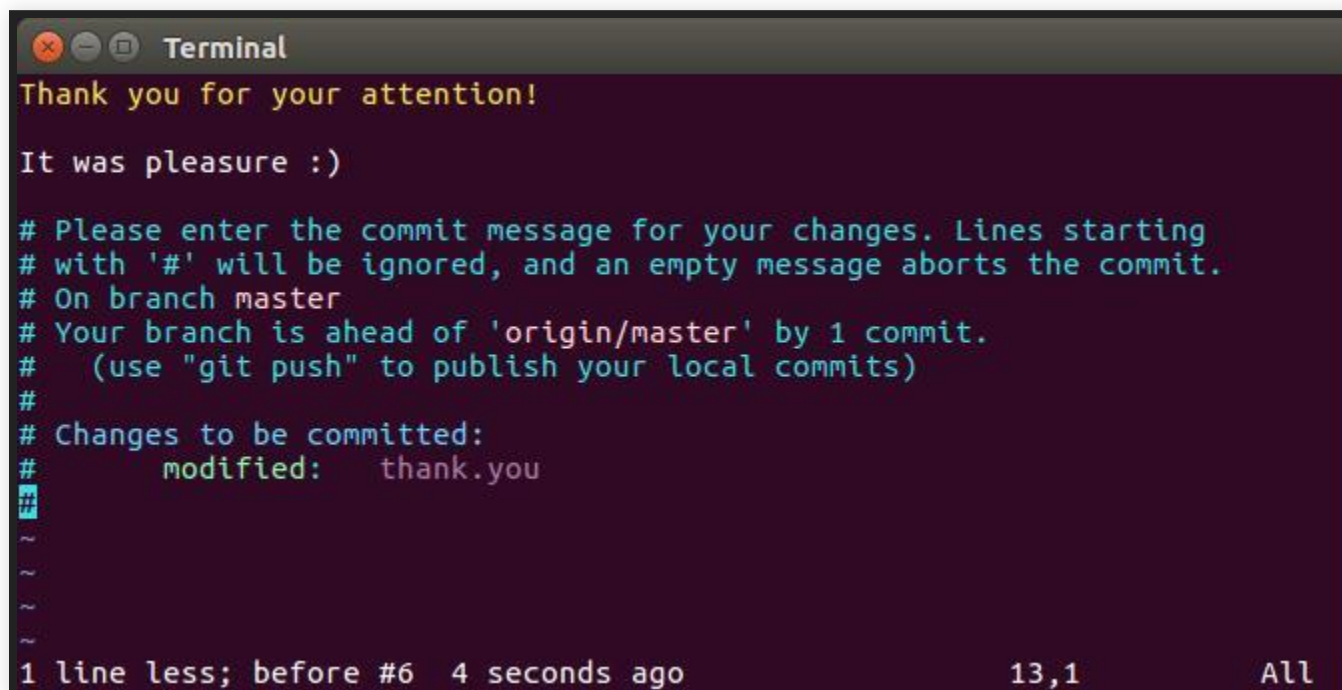
1. Nie tykamy, jeżeli są konflikty lub buildy świecą się na czerwono
2. Zapoznajemy się z descriptionem pull requesta
3. Rzut oka na listę commitów
4. Przeglądanie kodu
5. W niektórych serwerach, komentarze mogą wskazywać zaznaczony tekst
6. Jeżeli serwer to umożliwia, gdy wystawiający PR dostosuje się do uwag, my przeglądamy tylko nowo wypchnięte zmiany
7. Wystawiający PR na zakończenie pilnuje usunięcia brancha

# NINIEJSZA PREZENTACJA W FORMIE PDF

Prezentacja, na którą właśnie patrzysz, istnieje również w formie pdfa, niestety za pośrednictwem platformy webinarowej nie ma jak jej wysłać.

Jeżeli chciałbyś/chciałabyś ją sobie zachować, napisz do nas na [kontakt@GitWarsztaty.pl](mailto:kontakt@GitWarsztaty.pl) i ją podeślemy.

# DZIĘKUJEMY!

A terminal window titled "Terminal" with a dark background and light-colored text. It displays the output of a git commit command. The text is as follows:

```
Thank you for your attention!  
  
It was pleasure :)  
  
# Please enter the commit message for your changes. Lines starting  
# with '#' will be ignored, and an empty message aborts the commit.  
# On branch master  
# Your branch is ahead of 'origin/master' by 1 commit.  
#   (use "git push" to publish your local commits)  
#  
# Changes to be committed:  
#   modified:   thank.you  
~  
~  
~  
~  
~  
1 line less; before #6  4 seconds ago          13,1      All
```