

GIT

Piotr Kowalski

Wprowadzenie

Rodzina  
systemów

Składowe

GIT w życiu  
codziennym  
GitHub

Podsumowanie

Pytania

Linki

Autor

# GIT - system kontroli wersji

Piotr Kowalski

Wyższa Szkoła Informatyki Stosowanej i Zarządzania

11 czerwca 2023

# Wprowadzenie

## GIT

Piotr Kowalski

Wprowadzenie

Rodzina  
systemów

Składowe

GIT w życiu  
codziennym  
GitHub

Podsumowanie

Pytania

Linki

Autor

- Co to jest system kontroli wersji?
- Dlaczego zdobył serca wielu programistów?

## System kontroli wersji

Oprogramowanie służące do śledzenia zmian głównie w kodzie źródłowym oraz pomocy programistom w łączeniu zmian dokonanych przez wiele osób w różnych momentach.

# Scentralizowane

## GIT

Piotr Kowalski

Wprowadzenie

Rodzina  
systemów

Składowe

GIT w życiu  
codziennym  
GitHub

Podsumowanie

Pytania

Linki

Autor

- RCS
- CVS
- Subversion
- GNU CSSC, klon SCCS
- JEDI VCS

# Rozproszone

## GIT

Piotr Kowalski

Wprowadzenie

Rodzina  
systemów

Składowe

GIT w życiu  
codziennym

GitHub

Podsumowanie

Pytania

Linki

Autor

- Bazaar
- Codeville
- Darcs
- GIT
- GNU Arch
- Mercurial
- Monotone
- svk

# Zamknięte (własnościowe) systemy kontroli wersji

## GIT

Piotr Kowalski

Wprowadzenie

Rodzina  
systemów

Składowe

GIT w życiu  
codziennym  
GitHub

Podsumowanie

Pytania

Linki

Autor

- BitKeeper firmy BitMover
- Code Co-op firmy Reliable Software
- Perforce firmy Perforce Software
- Rational ClearCase firmy IBM
- Sublime firmy Lucent Technologies
- StarTeam firmy Borland
- Visual SourceSafe firmy Microsoft
- Visual Studio Team Foundation Server firmy Microsoft

# Informacje ogólne

## GIT

Piotr Kowalski

Wprowadzenie

Rodzina  
systemów

Składowe

GIT w życiu  
codziennym  
[GitHub](#)

Podsumowanie

Pytania

Linki

Autor

## GIT

Rozproszony system kontroli wersji. Stworzył go Linus Torvalds jako narzędzie wspomagające rozwój jądra Linux. GIT stanowi wolne oprogramowanie i został opublikowany na licencji GNU GPL w wersji 2.

# Historia

## GIT

Piotr Kowalski

Wprowadzenie

Rodzina systemów

Składowe

GIT w życiu codziennym  
GitHub

Podsumowanie

Pytania

Linki

Autor

- Prace nad GItem rozpoczęły się po tym, jak BitKeeper, używany wtedy do rozwoju Linuksa, przestał być darmowy dla projektów o otwartym kodzie źródłowym.
- Prace nad GItem rozpoczęły się 3 kwietnia 2005 roku, projekt został ogłoszony 6 kwietnia, 7 kwietnia GIT obsługiwał kontrolę wersji swojego własnego kodu, 18 kwietnia pierwszy raz wykonano łączenie kilku gałęzi kodu, 27 kwietnia GIT został przetestowany pod względem szybkości z wynikiem 6,7 łat na sekundę, a 16 czerwca Linux 2.6.12 był hostowany przez GITa.

# Założenia

## GIT

Piotr Kowalski

Wprowadzenie

Rodzina  
systemów

Składowe

GIT w życiu  
codziennym  
GitHub

Podsumowanie

Pytania

Linki

Autor

Torvalds szukał rozproszonego systemu kontroli wersji, który mógłby być użyty zamiast BitKeepera, głównymi kryteriami wyboru były:

- Wziąć przykład z CVS, czego nie robić.
- System powinien być rozproszony.
- System powinien być chroniony przed błędami w repozytorium (przypadkowymi, jak awaria twardego dysku, jak i złośliwymi, wprowadzonymi przez kogoś).
- System powinien być szybki.

Pierwsze dwa punkty wyeliminowały wszystko prócz Monotone'a, a czwarty punkt wyeliminował wszystko, więc Torvalds postanowił napisać własny system kontroli wersji.



# Rewizja

## GIT

Piotr Kowalski

Wprowadzenie

Rodzina  
systemów

Składowe

GIT w życiu  
codziennym  
GitHub

Podsumowanie

Pytania

Linki

Autor

Do wprowadzenia zmian w projekcie służy specjalna **operacja zatwierdzania** (*ang. commit*). Nie jest ona nigdy wykonywana przez GIT automatycznie. Jeśli uznamy, że bieżący stan plików lub folderów jest istotny, należy samodzielnie wykonać operację zatwierdzania.

Operację zatwierdzania możemy w pewnym uproszczeniu traktować jako zapisanie bieżącego stanu wszystkich plików i folderów projektu w danej chwili.

Wykonanie operacji zatwierdzania powoduje zapisanie **rewizji** (*ang. commit, revision*).

# Rewizja - dalej

## GIT

Piotr Kowalski

Wprowadzenie

Rodzina  
systemów

Składowe

GIT w życiu  
codziennym  
GitHub

Podsumowanie

Pytania

Linki

Autor

## Utworzenie rewizji

```
$ git add plik.txt
```

```
$ git commit -m "Dodaj plik.txt"
```

```
[master 66601e8] Dodaj plik.txt
```

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

```
create mode 100644 plik.txt
```

# Branch

## GIT

Piotr Kowalski

Wprowadzenie

Rodzina  
systemów

Składowe

GIT w życiu  
codziennym

GitHub

Podsumowanie

Pytania

Linki

Autor

Gałąź (*ang. branch*) jest to inna ścieżka dla rewizji.

Nazwa domyślnej gałęzi to **master**.

Branche nie są ze sobą połączone - tj. można pracować na nich niezależnie.

Stworzona rewizja istnieje tylko w aktualny branchu. Jeśli chcemy skopiować pojedynczą rewizję, to możemy użyć polecenia **git cherry-pick**, natomiast, będzie to kopia zmian, więc **commit hash** będzie inny.

# Branch - local

GIT

Piotr Kowalski

Wprowadzenie

Rodzina  
systemów

Składowe

GIT w życiu  
codziennym  
GitHub

Podsumowanie

Pytania

Linki

Autor

## Stworzenie nowego oraz przełączenie się na niego

```
$ git checkout -b simple-feature
```

```
Switched to a new branch 'simple-feature'
```

## Usunięcie brancha lokalnie

```
$ git branch -D simple-feature
```

```
Deleted branch simple-feature (was 2b325c2).
```

## Wypchnięcie brancha na serwer

```
$ git push origin simple-feature
```

```
Total 0 (delta 0), reused 0 (delta 0)
```

```
To git@github.com:piecioshka/test.git
```

```
* [new branch] simple-feature -> simple-feature
```

# Branch - remote

## GIT

Piotr Kowalski

Wprowadzenie

Rodzina  
systemów

Składowe

GIT w życiu  
codziennym  
GitHub

Podsumowanie

Pytania

Linki

Autor

## Wyświetlenie wszystkich gałęzi projektu (local & remote)

```
$ git branch -la
```

```
master
```

```
* simple-feature
```

```
remotes/origin/master
```

```
remotes/origin/simple-feature
```

## Usunięcie brancha z serwera

```
$ git push origin :simple-feature
```

```
To git@github.com:piecioshka/test.git
```

```
- [deleted] simple-feature
```

## GIT

Piotr Kowalski

Wprowadzenie

Rodzina  
systemów

Składowe

GIT w życiu  
codziennym  
GitHub

Podsumowanie

Pytania

Linki

Autor

Znacznik (*ang. tag*) jest to rewizja która jest specjalnie zapisana w repozytorium. Rewizja ta zostaje oznakowana specjalnym komentarzem, aby w przyszłości szybciej dotrzeć to odpowiedniego etapu projektu.

Przykład poprawnie stworzonych znaczników:

- <https://github.com/adobe/brackets/tags>

# Tag - dalej

## GIT

Piotr Kowalski

Wprowadzenie

Rodzina  
systemów

Składowe

GIT w życiu  
codziennym  
GitHub

Podsumowanie

Pytania

Linki

Autor

### Stworzenie nowego taga

```
$ git tag v1.0 -m "new feature"
```

### Wyświetlenie wszystkich tagów projektu

```
$ git tags -l
```

### Wypchnięcie tagów na serwer

```
$ git push --tags
```

# Wyświetlanie rewizji

## GIT

Piotr Kowalski

Wprowadzenie

Rodzina  
systemów

Składowe

GIT w życiu  
codziennym  
GitHub

Podsumowanie

Pytania

Linki

Autor

### Wyświetlanie rewizji w sposób czytelny

```
$ git log --graph --pretty=oneline
```

### Wyświetlanie rewizji w sposób czytelny

```
$ gitk
```



# Przykład projektu

## GIT

Piotr Kowalski

Wprowadzenie

Rodzina systemów

Składowe

GIT w życiu codziennym

GitHub

Podsumowanie

Pytania

Linki

Autor



Rysunek 1: Rewizje, branche, tagi

# GitHub - rejestracja

GIT

Piotr Kowalski

Wprowadzenie

Rodzina  
systemów

Składowe

GIT w życiu  
codziennym

GitHub

Podsumowanie

Pytania

Linki

Autor

**GitHub** Search or type a command ⓘ ⚙ Explore Features Enterprise Blog Sign in

## Build software better, together.

Powerful collaboration, review, and code management for open source and private development projects.

Tip: use at least one number and at least 7 characters.

By clicking on "Sign up for free" below, you agree to the [Terms of Service](#) and the [Privacy Policy](#).

**Sign up for free**

[See plans and pricing](#)

**Great collaboration starts with communication.**

Review changes, comment on lines of code, report issues, and plan the future of your project with [flexible discussion tools](#).

**Manage and contribute from all your devices.**

Native clients for [Mac](#), [Windows](#), and [mobile](#) devices allow you to stay involved with your projects from anywhere.

**The world's largest open source community.**

Share your code with the world and contribute to [the millions of public repositories](#) hosted on GitHub.

Rysunek 2: Formularz rejestracji konta

# GitHub - logowanie

GIT

Piotr Kowalski

Wprowadzenie

Rodzina systemów

Składowe

GIT w życiu codziennym

GitHub

Podsumowanie

Pytania

Linki

Autor

The screenshot shows the GitHub homepage with the login form highlighted by a red rectangle. The form is titled "Sign in" and contains the following elements:

- Username or Email:** A text input field containing "pleciohka@gmail.com".
- Password (forgot password):** A text input field containing "\*\*\*\*\*".
- Sign in:** A button located below the password field.

The page header includes the GitHub logo, a search bar, and navigation links: Explore, Features, Enterprise, Blog, Sign up, and Sign in. The footer contains links for GitHub, Applications, Services, Documentation, and More, along with Terms of Service, Privacy, Security, and copyright information.

Rysunek 3: Formularz logowania użytkownika

# GitHub - tworzenie nowego repozytorium

GIT

Piotr Kowalski

Wprowadzenie

Rodzina systemów

Składowe

GIT w życiu codziennym

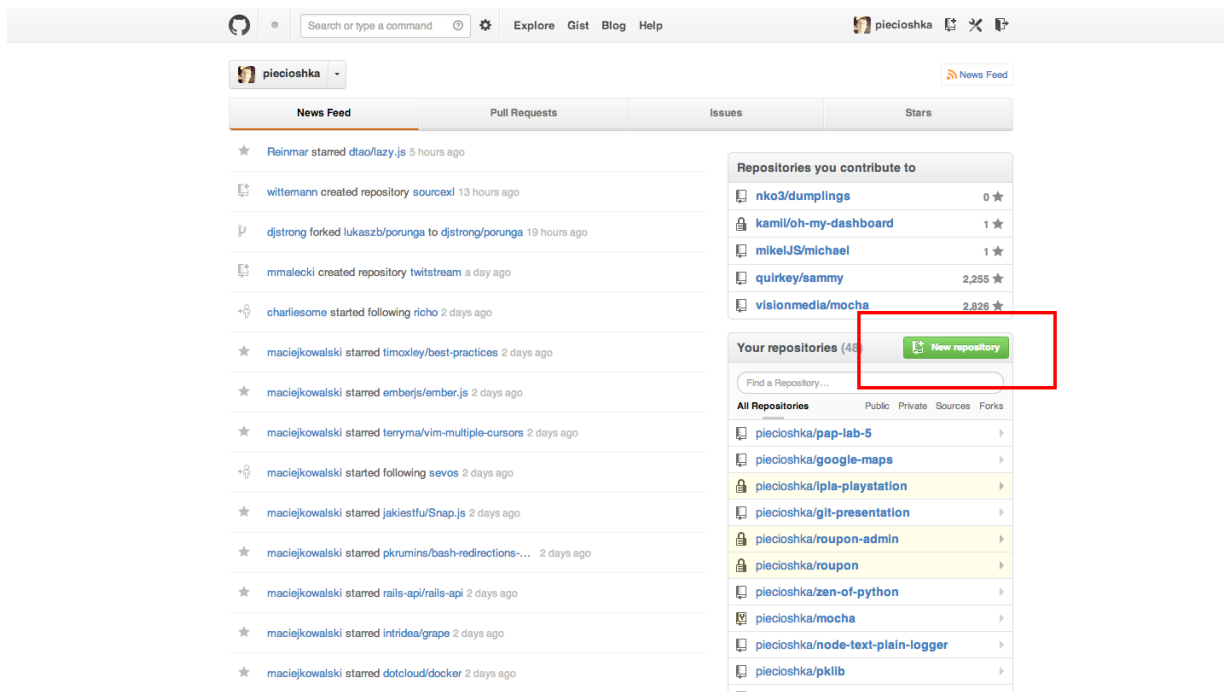
GitHub

Podsumowanie

Pytania

Linki

Autor



Rysunek 4: Link do utworzenia nowego repozytorium

# GitHub - nowe repozytorium

## GIT

Piotr Kowalski

Wprowadzenie

Rodzina systemów

Składowe

GIT w życiu codziennym

GitHub

Podsumowanie

Pytania

Linki

Autor

The screenshot shows the GitHub 'Create repository' interface. At the top, there's a navigation bar with the GitHub logo, a search bar, and links for 'Explore', 'Gist', 'Blog', and 'Help'. The user 'piecioszka' is logged in. The main form has a 'PUBLIC' label and a 'Repository name' field, which is highlighted with a red box. Below this, there's a hint: 'Great repository names are short and memorable. Need inspiration? How about `furry-octo-archer`.' The 'Description (optional)' field is empty. There are two radio buttons for visibility: 'Public' (selected) and 'Private'. Below these, there's a checkbox for 'Initialize this repository with a README'. At the bottom of the form is a green 'Create repository' button. The footer contains links for 'Terms of Service', 'Privacy', and 'Security', along with the GitHub logo and copyright notice.

Rysunek 5: Ustal nazwę dla nowego repozytorium

# GitHub - nowe puste repozytorium

## GIT

Piotr Kowalski

Wprowadzenie

Rodzina systemów

Składowe

GIT w życiu codziennym

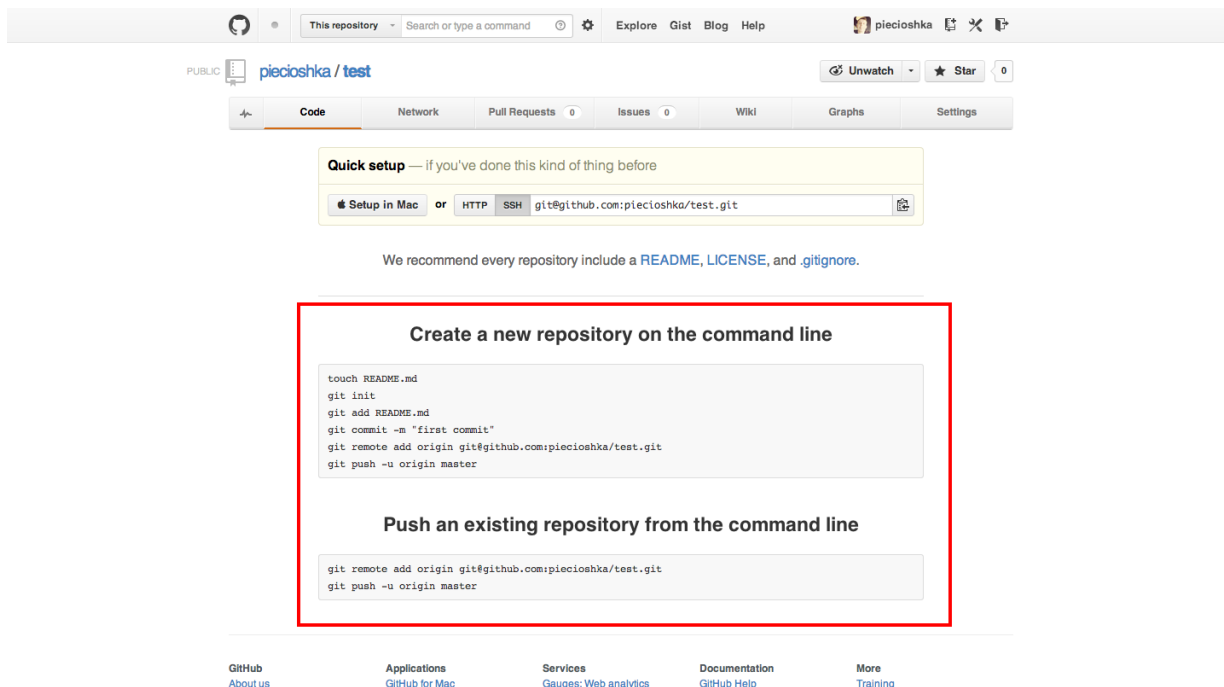
GitHub

Podsumowanie

Pytania

Linki

Autor



Rysunek 6: Początkowe ustawienia repozytorium

## GIT

Piotr Kowalski

Wprowadzenie

Rodzina  
systemów

Składowe

GIT w życiu  
codziennym

GitHub

Podsumowanie

Pytania

Linki

Autor

### Listing 1: Pierwszy projekt

```
$ mkdir test && cd test
$ touch README.md
$ git init
$ git add README.md
$ git commit -m "Pierwszy commit"
$ git remote add origin git@github.com:piecioshka/test.git
$ git push -u origin master
```

# GitHub - projekt

## GIT

Piotr Kowalski

Wprowadzenie

Rodzina systemów

Składowe

GIT w życiu codziennym

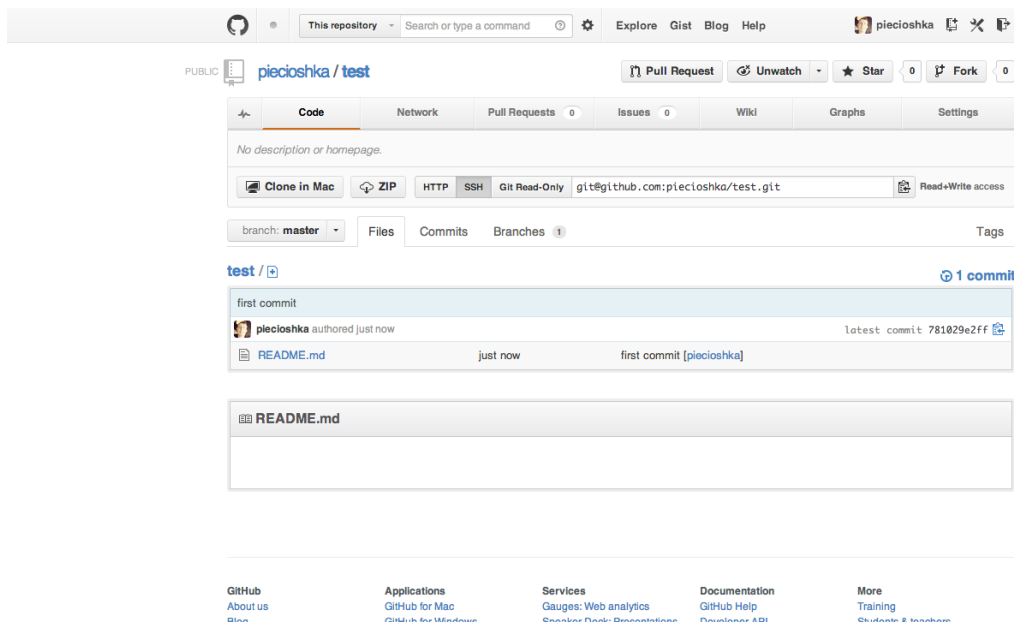
GitHub

Podsumowanie

Pytania

Linki

Autor



Rysunek 7: Projekt



## Listing 2: Praca lokalna

```
$ mkdir test && cd test  
$ touch README.md  
$ git init  
$ git add README.md  
$ git commit -m "Pierwszy commit"
```

## Listing 3: Praca zdalna

```
$ git remote add origin git@github.com:piecioshka/test.git  
$ git push -u origin master
```

## Listing 4: Najpopularniejsze polecenia

```
$ git add  
$ git blame  
$ git branch  
$ git checkout  
$ git diff  
$ git log  
$ git pull  
$ git push  
$ git reset  
$ git status  
$ git tag
```

# Podsumowanie

## GIT

Piotr Kowalski

Wprowadzenie

Rodzina  
systemów

Składowe

GIT w życiu  
codziennym  
GitHub

Podsumowanie

Pytania

Linki

Autor

GIT jest:

- wszechstronny i uniwersalny
- rozproszony
- przenośny
- multi-platformowy
- wygodny
- szybki
- bezpieczny
- darmowy

Czego więcej trzeba?

# Pytania

## GIT

Piotr Kowalski

Wprowadzenie

Rodzina  
systemów

Składowe

GIT w życiu  
codziennym  
GitHub

Podsumowanie

Pytania

Linki

Autor

- ❶ Jak się na to zapatrujecie?
- ❷ Czy jest sens, używania systemu kontroli wersji, podczas działań w pojedynkę?

# Przydatne źródła

## GIT

Piotr Kowalski

Wprowadzenie

Rodzina  
systemów

Składowe

GIT w życiu  
codziennym  
GitHub

Podsumowanie

Pytania

Linki

Autor

- <https://help.github.com/>
- <https://github.com/git/git>
- <https://git-scm.com/book/pl/Pierwsze-kroki-Wprowadzenie-do-kontroli-wersji>
- <https://bitbucket.org/>

Źródła do tej prezentacji:

- <https://github.com/piecioshka/slides-git-overview>

## GIT

Piotr Kowalski

Wprowadzenie

Rodzina  
systemów

Składowe

GIT w życiu  
codziennym  
GitHub

Podsumowanie

Pytania

Linki

Autor

- Piotr Kowalski
- <https://github.com/piecioshka>
- <https://twitter.com/piecioshka>