



M03 PRZYGOTOWANIA L05 Narzędzia

(Nie samą konsolą człowiek żyje... czy jednak... ?)

Maciej Aniserowicz

Najlepszy sposób na pracę z Gitem?

Bezapelacyjnie: **KONSOLA!**

... ale spokojnie, bez nerwów ...

”

Ja byłem bardzo Zagorzałym Windowsiarzem,
uzależnionym od okienek.

Bałem się konsoli.



devstyle.pl

~%. kurs **gita**

Konsola

Konsola: obawy

- :(pewnie trzeba będzie dużo pisać
- :(pewnie niewygodnie będzie szukać okienka konsoli
- :(pewnie nie zapamiętam wszystkich komend
- :(pewnie będzie tak bardzo wolno i nieproduktywnie

Konsola: kiedy się **nie sprawdza?**

Nie jesteśmy masochistami ani fanatykami! A... pragmatykami.

UI bywa lepsze w tych przypadkach:

- przeglądanie historii projektu
- porównywanie zmian (DIFF)
- rozwiązywanie konfliktów w plikach (merge)

... Do tego wszystkiego dojdziemy.

Konsola: dwa konieczne **tricki**

Warunki do efektywnego wykorzystania konsoli:

1. **Okno konsoli** zawsze pod ręką, na każde zwołanie
2. Zminimalizowanie **uderzeń w klawiaturę**
(pisanie = pomyłki)

WTEDY pokochasz pracę z konsolą <3.



Konsola: trick 01

01 Konsola zawsze pod ręką

Zaleta: szybciej niż kliknięcie w jakiekolwiek UI!

U mnie:

- Windows: WIN+3
- MacOS: CTRL+OPT+3

01 Konsola. Jak? [Windows]

Konsola Git Bash przypięta (*pin*) do *taskbar*.

Zaleta: szybciej niż kliknięcie w jakiekolwiek UI.

01 Konsola. Jak? [Windows]

- 
1. Konsola Git Bash przypięta (*pin*) do *taskbar*
 2. Katalog startowy: Twój aktualny projekt
(lub ogólny katalog z projektami)

01 Konsola. Jak? [MacOS] (1/2)

1. Automator.app

2. Service -> no input -> run AppleScript

3. Save

```
on run {input, parameters}
    tell application "Terminal"
        reopen
        activate
    end tell
end run
```

01 Konsola. Jak? [MacOS] (2/2)

- 
4. System Preferences -> Keyboard -> Shortcuts -> Services
 5. Wybierz swój workflow -> Add Shortcut
 6. Ustaw skrót
- 

01 Konsola. Jak? [MacOS] (alternatywnie)

1. Dodatkowy program (u mnie: Better Touch Tool)
(i tak warto mieć; < \$10)
2. Konfiguracja



Shortcut	On Do... ^	Assigned Action	Enabled
⌃⌥ 3	<input checked="" type="checkbox"/>	Launch Terminal.app	<input checked="" type="checkbox"/>



Konsola: trick 02

02 Zminimalizować pisanie

Zaleta: komendy piszą się „same”! Nie popełniamy błędów.

Git posiada „aliasy” - sposób na skrócenie komend...

(do tego jeszcze dojdziemy)

...ale dla nas **aliasy to za mało.**

Dlaczego? Bo wywołanie aliasa wymaga napisania „git”!

(wieeeeele razy dziennie)

02 (prawie) Bez pisania. Czyli...?

Po pierwsze: **autocomplete**.

Ale to już mamy!

02 (prawie) Bez pisania. Czyli...?

- ga[spacja] => git add -p
- gb[spacja] => git branch
- gmnff[spacja] => git merge --no-ff -e -m "" -

Rozpieszczeni przez IntelliSense w IDE?

W konsoli „piszemy głównie spacją”!

02 Bez pisania. Jak?

Rozwiążanie: **Text Expanding**

Komendy rozwijane „w locie”, podczas pisania.

Za pomocą zewnętrznych narzędzi, niezależnych od Gita.

02 Bez pisania. Jak? [Windows]

U mnie: **AutoHotKey** (AHK)

autohotkey.com

::gc::git commit

::gr::git rebase -i

::gai::git add -i

Alternatywa: dowolny „text expander”

kursgita.ahk

Mój skrypt AutoHotKey z
przydatnymi rozwinięciami
komend Gita.



02 Bez pisania. Jak? [MacOS]

U mnie: aText (\$5)

Opcja: „zmieniaj tylko w aplikacji Terminal”

Alternatywy:

- Text Expander (drooogo)
- Systemowa podmiana



(„u mnie nie działa”)

Git Command	Shortcut
git commit	gc
git commit --allow-empty	gcae
git status	gs
git add -p	ga
git add -i	gai
git log	gl
git diff	gd
git merge --no-ff -e -m "" -	gmnff
git fetch	gf
git stash	gst
git checkout	gch
git branch	gb
git rebase -i	gr



Konsola: trick 03

(bonus)

BONUS 03 Z konsoli do schowka

Często się przyda do kopiowania do schowka niektórych informacji z Gita (głównie identyfikatorów).

- Windows: clip (np.: dir | clip)
- MacOS: pbcopy (np.: ls | pbcopy)



Konsola: jaką?

Jaka konsola?

U mnie:

- Windows: Git Bash
- MacOS: Terminal (+autocomplete)

Ale używaj jakiej chcesz

(iTerm2, Powershell, Cmder, Console2, ConEmu, ...)



devstyle.pl

~%. kurs **gita**

UI

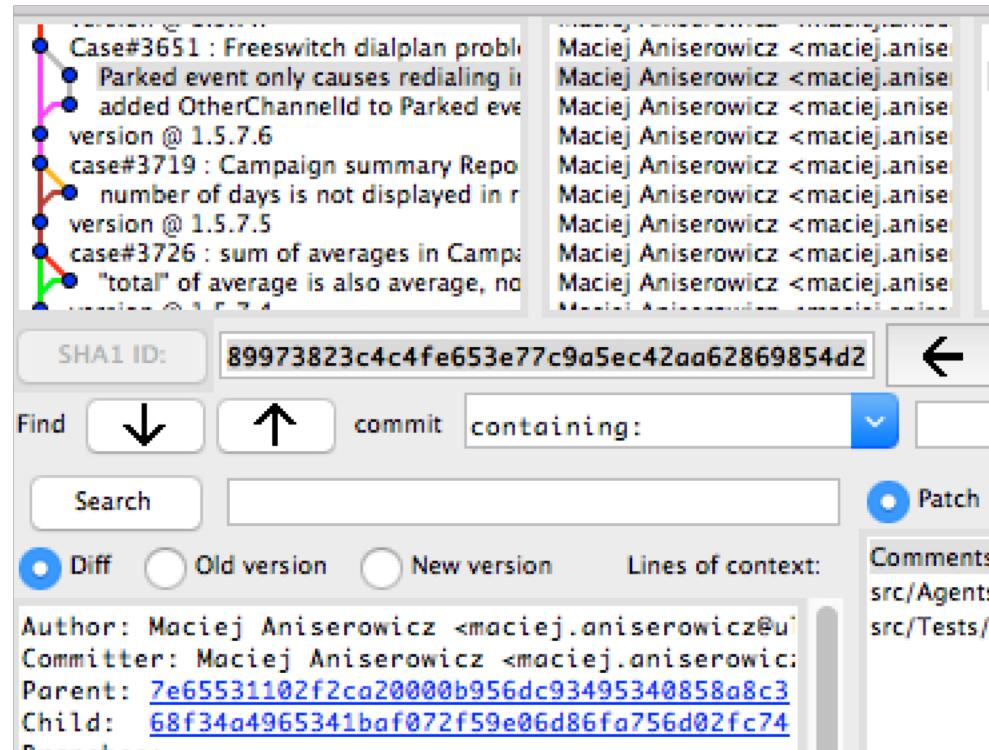
Gitk - Git repository browser

Instalowany razem z Gitem.

- gałęzie
- diff
- commity

Pokazuje historię.

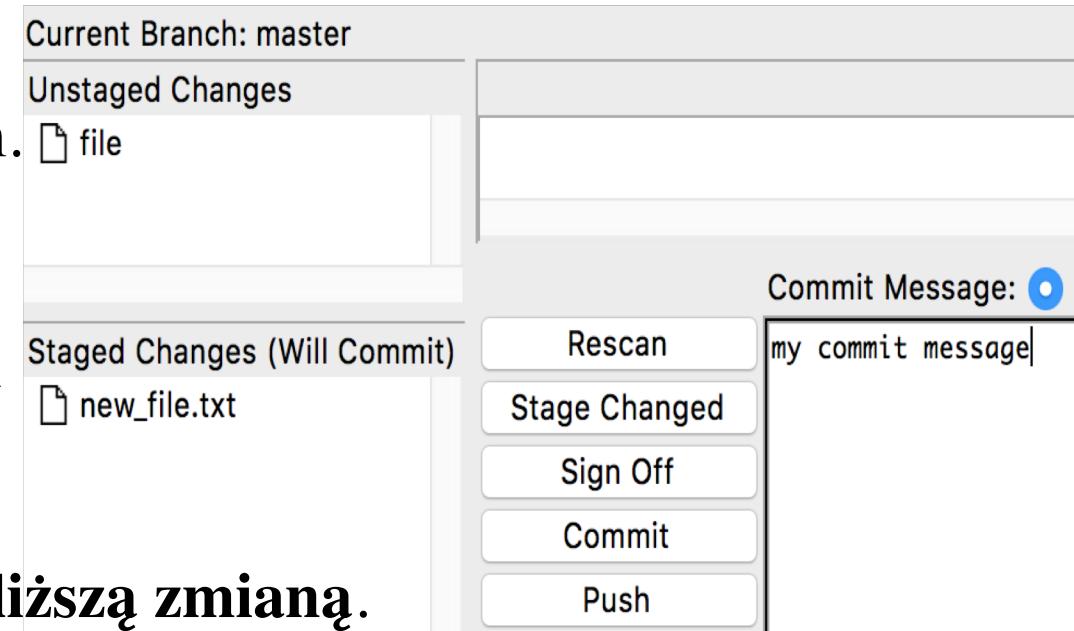
Będziemy często używać.



Git GUI - Git Graphical Interface

Instalowany razem z Gitem.

- aktualne zmiany
- możliwość zapisu zmian
- ...



Umożliwia pracę nad **najbliższą zmianą**.

(prawie) Nie będziemy używać.

IDE - integracja

- Visual Studio
- IntelliJ
- Eclipse
- Webstorm
- VS Code
- ...

IDE - po co?

- symbole pokazujące stan plików
- metadane o kodzie (np. kto autorem metody)
- „ultra-szybki” commit

Nie będziemy używać.

Po kursie - Twoja decyzja.

IDE i “ultra-szybki” commit

Wiem, że stworzenie prostego commita z IDE będzie
szybsze niż z konsoli (jeden skrót).

ALE to może zredukować Gita do roli „*maszynki do robienia ZIPów*”... a **tego nie chcemy**.

Wkrótce pokażę jak (i dlaczego) „powinno się” robić commity.
(też będzie szybko!)

Inne narzędzia: "standalone UI"

- TortoiseGit
- GitKraken
- Tower
- SourceTree
- GitHub Desktop
- Git Extensions
- SmartGit
- gmaster

... Narzędziowy Zawrót Główy ...

Inne narzędzia: "standalone UI"

Narzędzi jest dużo.

Różnych.

Jedne darmowe, inne płatne.

Który najlepszy???

Z konsolą ZROZUMIESZ Gita. A potem podejmiesz decyzję.

[KG]-Git_Clients.pdf

Zestawienie
najpopularniejszych
UI do obsługi Gita.





Jeszcze jedno narzędzie...

Konsolowy edytor tekstu: **VIM**

Spoiler alert!

Po co? Dlaczego?

Spokojnie, **bez paniki.**

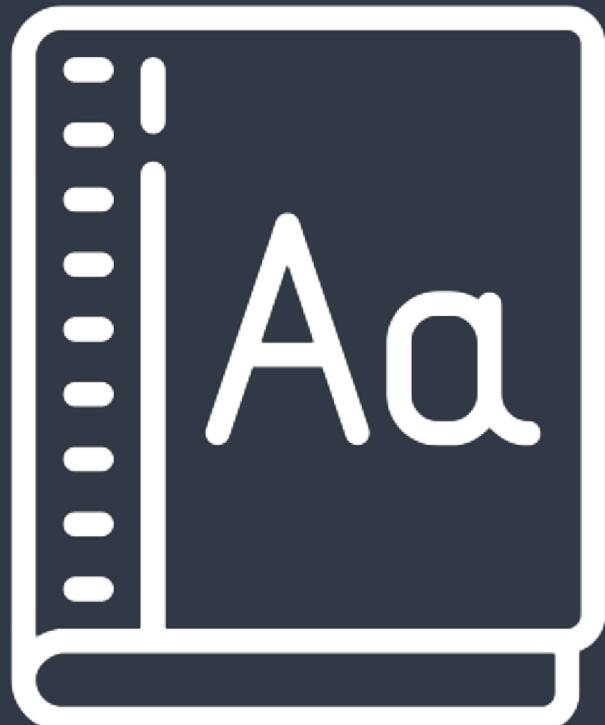
Wszystko się wyjaśni.



VIM

Magiczny edytor tekstu, z
którego wychodzi się
poprzez restart komputera.

[tak naprawdę to nie]



ZADANIE

- ✓ Ustaw skrót do pokazania konsoli
- ✓ Rozważ „text expanding”

