

WYCIŚNIJ SOKI Z GITA

PRZESTAŃ UŻYWAĆ JAK ZIPA



TOMASZ SKRASKOWSKI



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

GITWARSZTATY W TWOJEJ FIRMIE



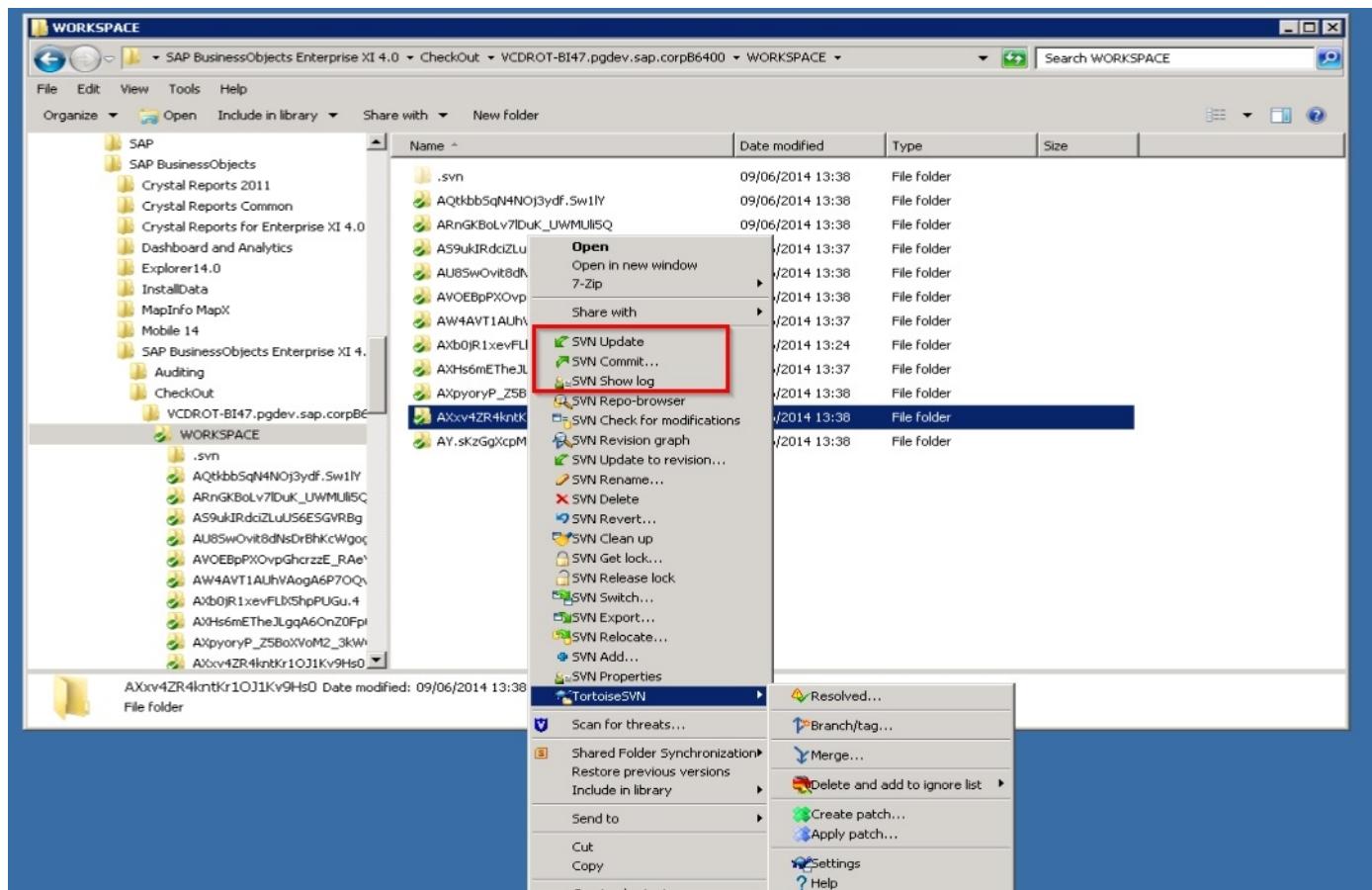
www.gitwarsztaty.pl

www.GitWarsztaty.pl

MOŻNA ODEBRAĆ TELEFON TAK...



SVN UPDATE... SVN COMMIT...



**MOŻNA OBRABIAĆ MATERIAŁ PROSTYMI
NARZĘDZIAMI...**

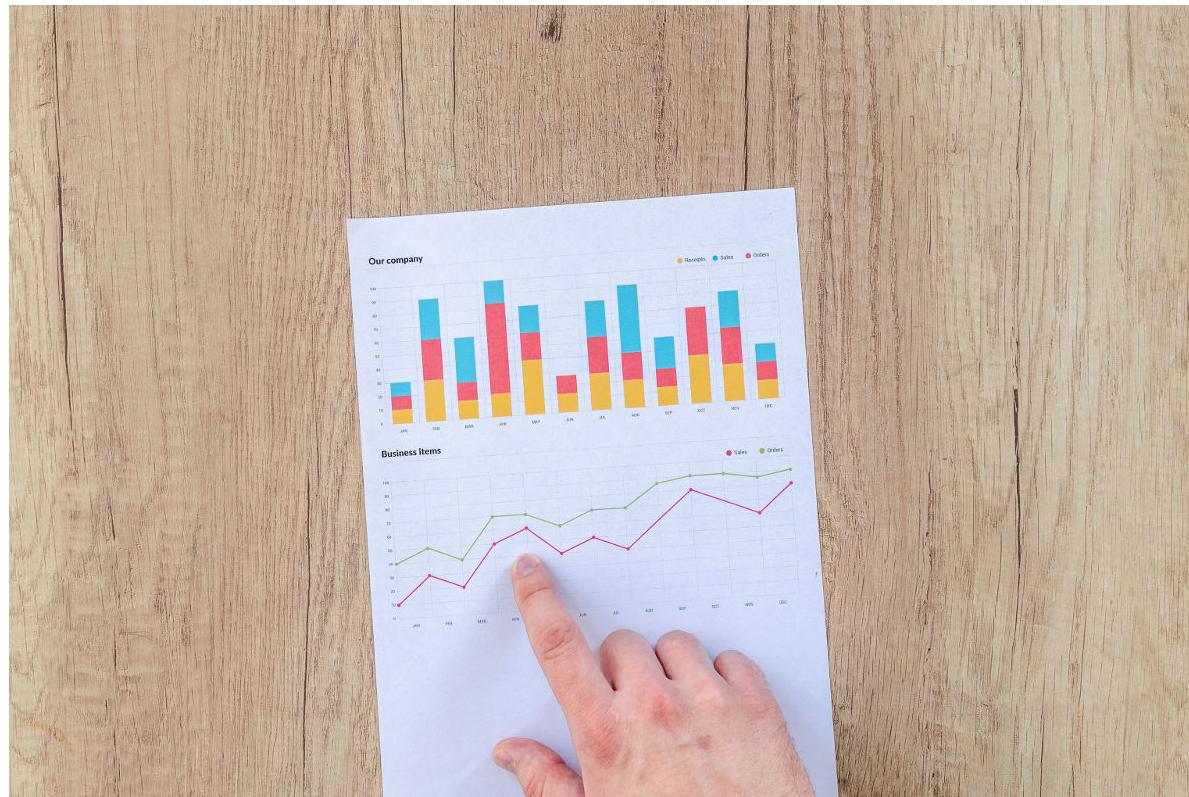


**MOŻNA OBRABIAĆ MATERIAŁ
SKOMPLIKOWANYMI NARZĘDZIAMI...**



GIT WYKRACZA POZA WERSJONOWANIE I KOLABORACJĘ

**GIT TO NARZĘDZIE DO EDYCJI I ZARZĄDZANIA KODEM
PROGRAMUJ SZYBCIEJ I TWÓRZ MNIEJ BŁĘDÓW**



ZAAWANSOWANE NARZĘDZIA MAJĄ PRÓG WEJŚCIA

Nie rozmawiałem jeszcze z nikim, kto...

...nie widzi w R# niczego szczególnego i
zna go dobrze

...nie widzi w Vimie* niczego
szczególnego i zna go dobrze

*Vimie w IDE(VS, Rider, VSCode)



...nie widzi w Gicie niczego szczególnego i
zna go dobrze

WYCISKANIE GITA

PODSTAWOWY PRÓG WEJŚCIA - KONSOLA

```
└─ bash --init-file $BASH_IT/bash_it.sh -i

Homebase: ~
→ cd workspace
ls
cd jockeyjs
git
Homebase: ~/workspace
→ ls
cd jockeyjs
git statusAlcatraz          apartmentlist           friend_memories
AutolayoutCollectionViewIssue apartmentlist-ios      jockeyjs
Contentment                  apartmentlist.sublime-project keyboard_typer
DynamicResizeUITextView      apartmentlist.sublime-workspace misc-js-css-extension
JustANudge                   bash-it
TestuserDefaults              bash-it-theme-screenshots
WhereTheHellAreYouiOS        drugs-and-booze-check
XcodeDocExpander             facebook_event_gcal      ocstyle
                                         resume.rb
                                         run_distance
                                         scripts

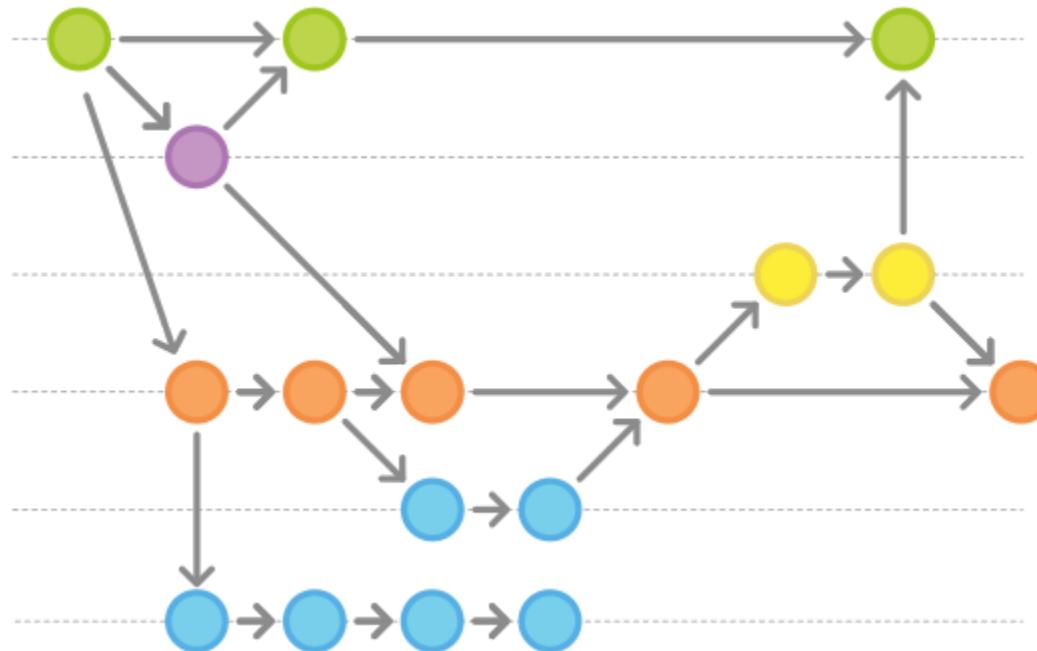
Homebase: ~/workspace
→ cd jockeyjs
Homebase: ~/workspace/jockeyjs |master ✘
→ git status
# On branch master
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       bobby-black.jpg
nothing added to commit but untracked files present (use "git add" to track)

Homebase: ~/workspace/jockeyjs |master ✘
→ cd ~/Desktop
screencapture ba
Homebase: ~/Desktop
→ screencapture bakke-black.jpg
```

(Główne trudność dla użytkowników Windowsa)

WYCISKANIE GITA

DRUGI PRÓG WEJŚCIA - FRUWAJĄCE COMMITY



Zrozumienie i przyzwyczajenie się do rozproszonej architektury Gita,
w tym do commitów latających między branchami i repozytoriami

WYCISKANIE GITA

TRZECI PRÓG WEJŚCIA - MNOGOŚĆ KOMEND I PRZEŁĄCZNIKÓW

--all-match
Limit the commits output to ones that match all given `--grep`, instead of ones that match at least one.

-invert-grep
Limit the commits output to ones with log message that do not match the pattern specified with `--grep=<pattern>`.

-i

-regexp-ignore-case
Match the regular expression limiting patterns without regard to letter case.

-basic-regexp
Consider the limiting patterns to be basic regular expressions; this is the default.

-E

-extended-regexp
Consider the limiting patterns to be extended regular expressions instead of the default basic regular expressions.

-F

-fixed-strings
Consider the limiting patterns to be fixed strings (don't interpret pattern as a regular expression).

-P

-perl-regexp
Consider the limiting patterns to be Perl-compatible regular expressions.
Support for these types of regular expressions is an optional compile-time dependency. If Git wasn't compiled with support for them providing this option will cause it to die.

--remove-empty
Stop when a given path disappears from the tree.

--merges
Print only merge commits. This is exactly the same as `--min-parents=2`.

--no-merges
Do not print commits with more than one parent. This is exactly the same as `--max-parents=1`.

--min-parents=<number>

--max-parents=<number>

--no-min-parents

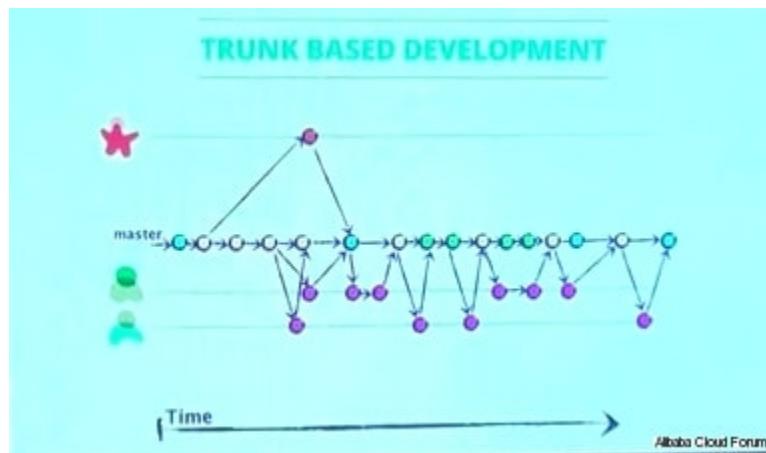
--no-max-parents

WYCISKANIE GIT

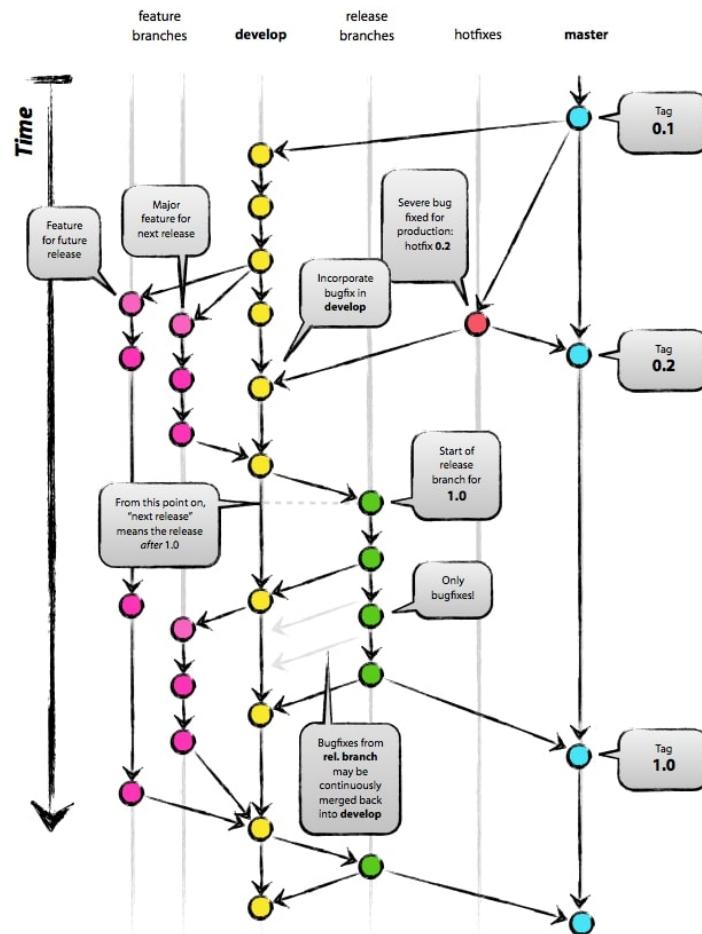
CZWARTY PRÓG WEJŚCIA - WYBÓR I ODNALEZIENIE SIĘ W
WORKFLOW



TRUNK BASED DEVELOPMENT



GIT FLOW



GIT GUITAR HERO



FEATURE TOGGLE (AKA FEATURE FLAG)



MartinFowler.com/articles/feature-toggles.html

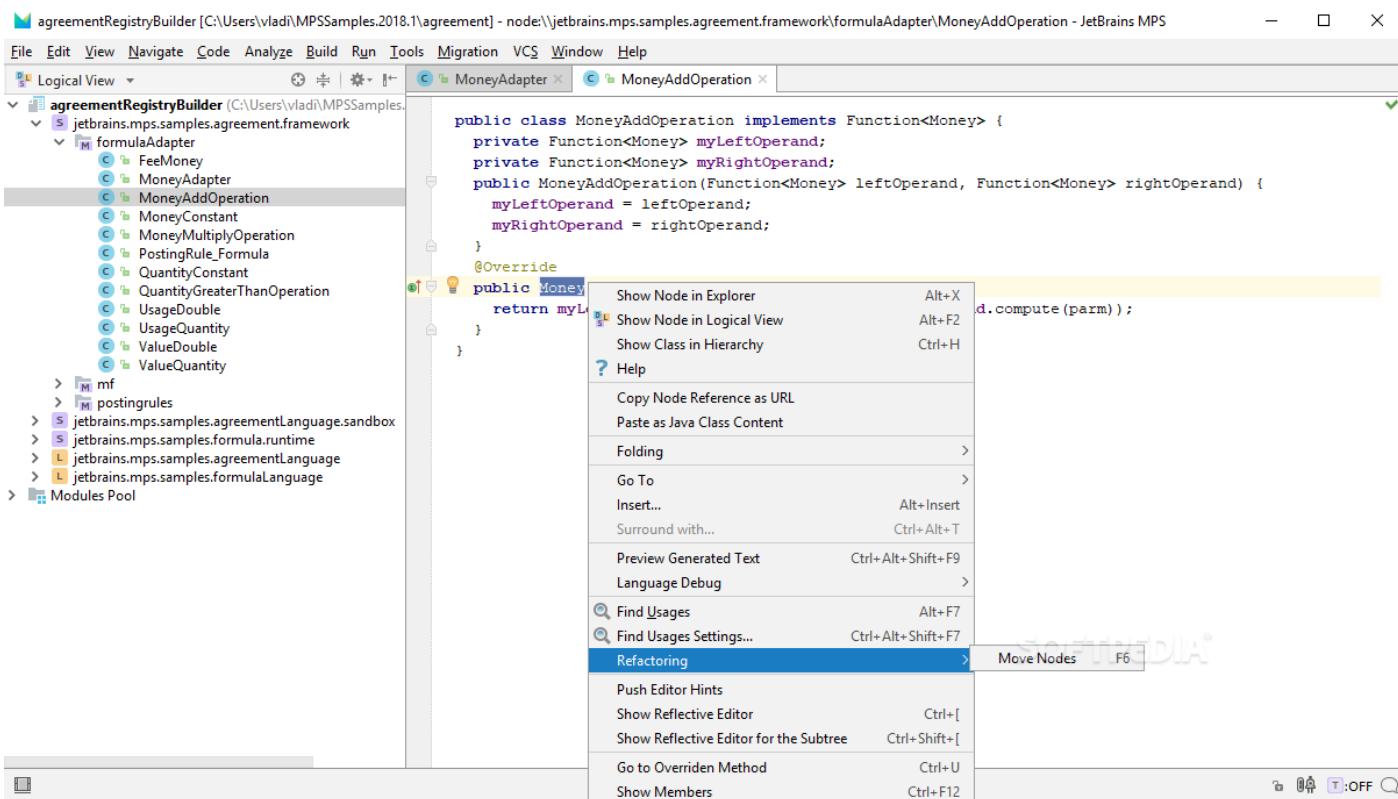
OBAWY PRZED KONSOLĄ

- Pewnie trzeba się strasznie opisać, a to będzie wolno szło
- Pewnie będzie problem spamiętać komendy
- To jest jakiś niewygodny przeżytek

KIEDY NIE WARTO KURCZOWO TRZYMAĆ SIĘ KONSOLI?

- Tool bywa lepszy do diffa
- Tool lepiej wizualizują historię branchów
- W dobrym IDE wygodniej się patrzy na *Git Blame*
- Nie warto rozwiązywać konfliktów bez toola

MYSZKA JEST DO EKSPLOROWANIA



SĄ TOOLE Z DOBRZE OGARNIĘTYMI SKRÓTAMI KŁAWISZOWYMI

Fork Keyboard Shortcuts #333

① Open

TanyaPristupova opened this issue on Jun 20 · 40 comments

The screenshot shows a GitHub issue page for "Fork Keyboard Shortcuts #333". The page has a green "Open" button and a comment from TanyaPristupova. Below the comment, there are two sections of keyboard shortcuts:

General Navigation

- `Ctrl 1` - Show Changes view (second press will focus commit field)
- `Ctrl 2` - Show All Commits view (second press will jump to HEAD)
- `Ctrl P / Ctrl Shift 0` - Show Quick Launch window
- `Ctrl Tab` - Select next tab
- `Ctrl Shift Tab` - Select previous tab
- `Ctrl T` - Open new tab
- `Ctrl W` - Close current tab
- `Ctrl = / Ctrl -` Zoom in/ Zoom out
- `Ctrl ,` - Open Fork Preferences

All Commits View

- `Ctrl 2` - jump to HEAD
- `Ctrl F` - Commit search
- `Ctrl Enter, F3` - Jump to next search result
- `Shift Enter, Shift F3` - Jump to previous search result
- `Ctrl C` - Copy commit info
- `Delete` - Remove Branch/Stash

ALE DOBRZE OGARNIĘTA KONSOLA WYGRYWA

- Z dobrze ogarniętą konsolą pracuje się wydajniej
- Konsola zawsze ma więcej możliwości - żaden tool nie pokryje wszystkich casów
- Konsola zachęca do lepszego zrozumienia Gita
- Nie ma toola działającego szybciej niż konsola
- Toola często zachęcają do złych praktyk i budują złe nawyki
- Toola często zachęcają do klikania zamiast używania skrótów klawiszowych
- Wiele (popularnych) tooli działa wolno i odrywa od *flow*

CO TO ZNACZY DOBRZE OGARNIĘTA KONSOLA?

- Konsola zawsze pod ręką
 - Zdecydowanie najlepiej gdy jest zintegrowana w IDE
 - Opcjonalnie wyskakująca od skrótu klawiszowego
(choć to mało przy otwartych na raz dwóch lub więcej projektach)
- Konsola z wydajnym autouzupełnianiem
- Zdecydowanie dobrze, gdy konsola ma wydajny *reverse search*
- Ustawione *Git aliasy* i aliasy na poziomie shella

JAK W KILKA MINUT PRZYGOTOWAĆ KONSOLĘ NA START?

- Aby spełnić wszystkie *prerekwizity*
 1. IDE z wbudowanym terminaliem
(np. Jet Brains Rider, działa reverse search)
 2. Bash ustawiony jako terminal w IDE
(na Windowsie GitBash)
 3. Aliasy (Gitowe i Bashowe) na start wzięte z internetu
(np. z tej prezentacji, później dostosowywanie pod siebie)
- Opcjonalnie
 - Dla IDE bez terminala
 - (Git)Bash przypięty do taskbarda pod stałą pozycją wywoływany *Win+pozycja*
 - Lub nakładka na (Git)Bash umożliwiająca wywołanie skrótem klawiszowym np. *Conemu*
 - W terminalu ustawiony katalog z projektami jako katalog startowy
 - *Text expandery* na poziomie OS np. AutoHotKey
 - Odradzam poleganie na *Alt+Tab*

PRZEKONAŁEM JUŻ NIEJEDENEGO ZATWARDZIAŁEGO WINDOWSiarZA :)

Ostatnio przeczytane

Krzysztof Porebski 16:37
powiem Ci, że zmiana na konsolce

w gitcie mocno na plus



mozesz agitowac

podajac mnie jako przyklad

bo sie bronilem rekami i nogami

16:37
Cieszy mnie to 😊

Wpisz nową wiadomość

A, C, 😊, GIF, 📸, 📁, ... ➡

CO DAJE GIT?

JAK PRZYZYDZIA SIĘ DO WIĘKSZEJ WYDAJNOŚCI PRACY?

**dlaczego?
po co? jak?**

GDY WSZYSTKO MUSI IŚĆ PO SIECI...



KONTROLA NAD SYTUACJĄ W IDE

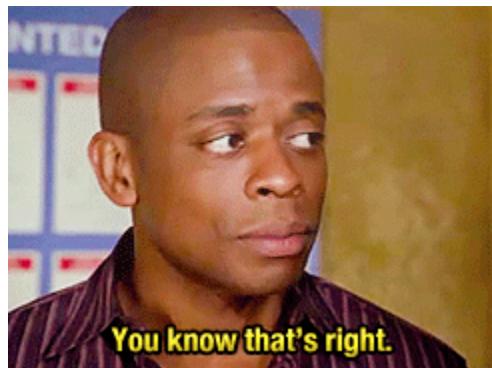


NOWE FUNKCJONALNOŚCI PO KAWAŁKU

JAKĄ MAŁĄ ZMIANĘ PROWADZĄCĄ DO CELU W TYM
MOMENCIE WPROWADZAM?

COMMIT!

(MAŁY I CZYSTY - SRP)



Zawsze później można się z niego wycofać.

W ORK | N P ROGRESS



```
wip = !git add --all && git commit -m "WIP"
```

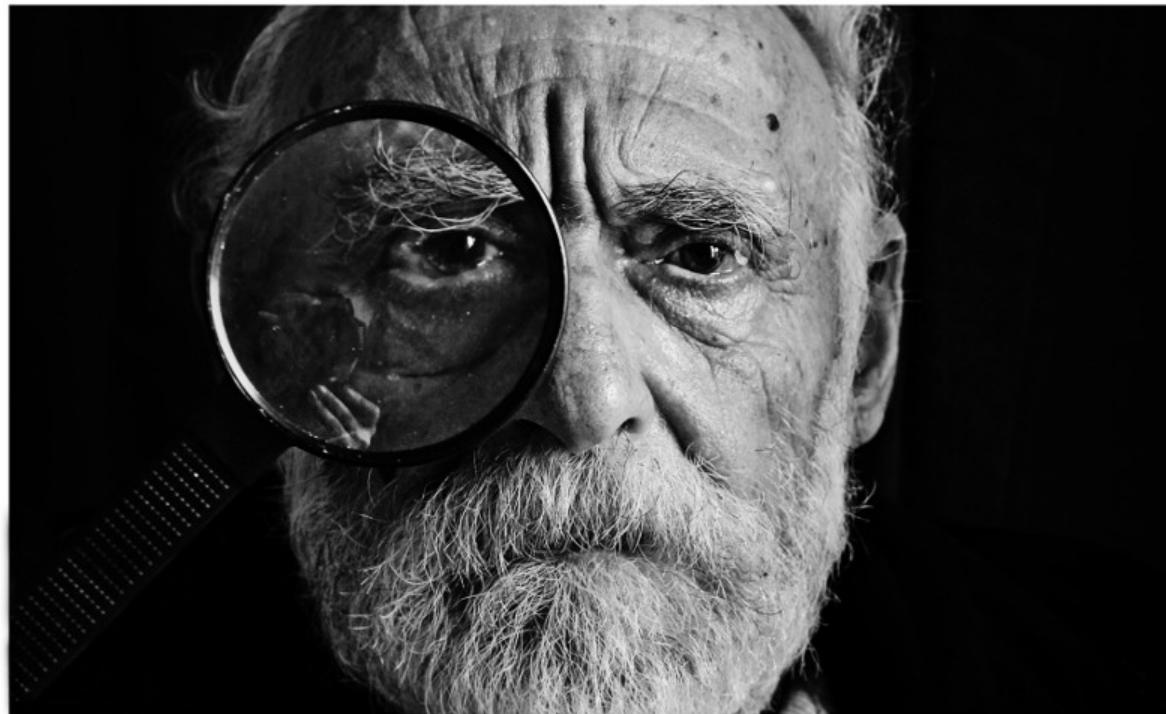
MONITOROWANIE ZMIAN

Jesteśmy spokojni, że nie przemycimy przypadkiem zmiany, która powstała nieuchiący lub tylko na chwilę.



MONITOROWANIE TOOLI

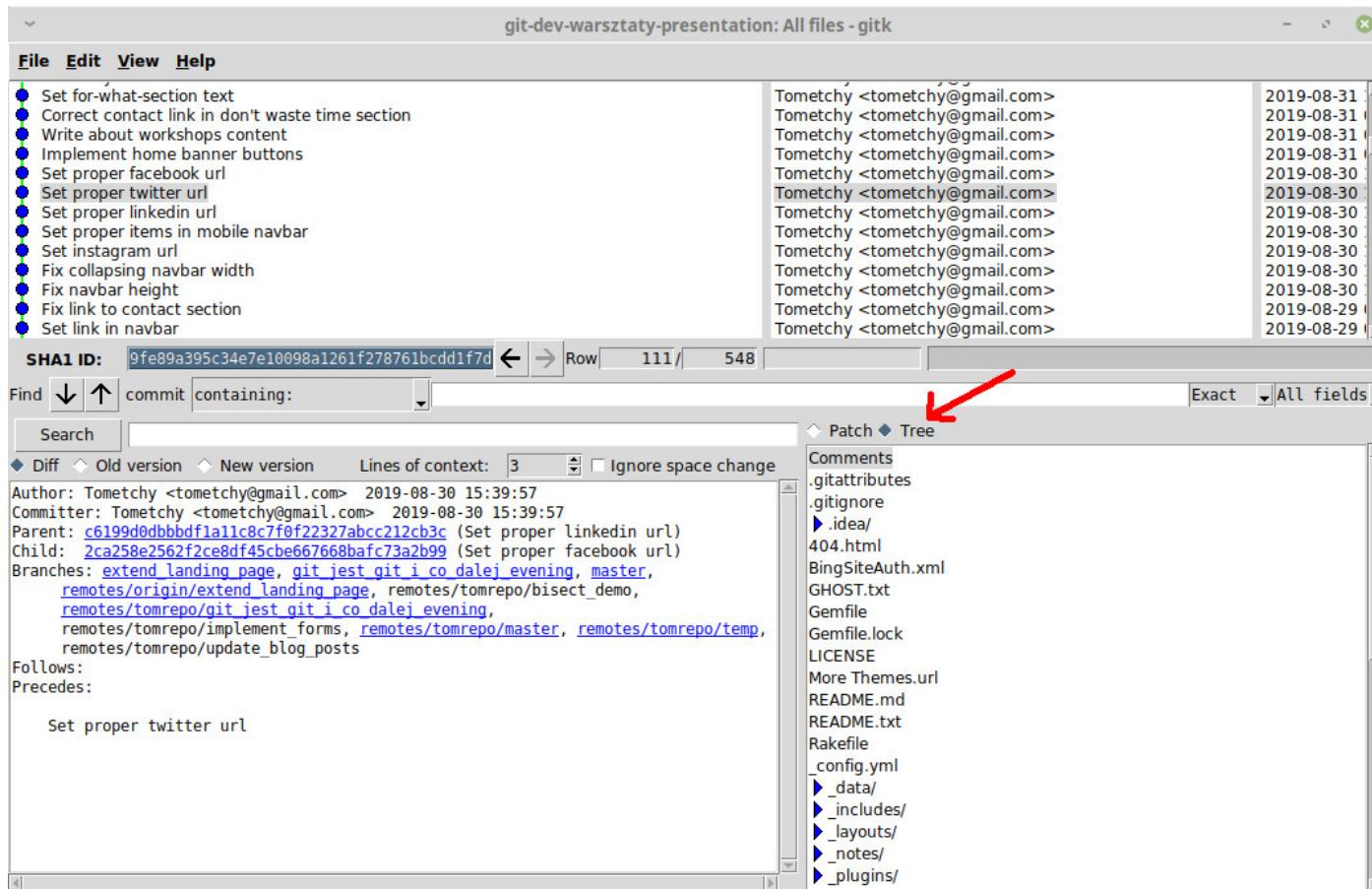
Tool did something over many files...
is it exactly what it was meant to be?



STABILNA PRACA Z NIESTABILNYM KODEM

bda54862 Remove phone book button from main canvas
bb4efb64 Add phone book to shell resources
9cff0517 Add phone book image for slide panel
18b3ab24 Set phone book tab view model
251c65eb Remove not used catalog from phone book project
fa27d977 Use phone book view model from old module
836f9507 Clean phone book tab in side panel
98384dbb Move resources from old module to new phone book
b179f970 Move phone book view from old modules to side panel
882d2cb1 Remove references to old phone book module
3633f485 Merge old phone book module style with new in side panel
061441fb Set close phone book button to hide panel
9b54878f Remove unused button from phone book
153dbcd6 Remove no longer needed files from old phone book module
2895928d Remove old phone book module
6182277b Replace phone book buttons style to improve readability
49ff42b6 Remove shadows from phone book
7d6584da Update entity migrations instruction

ALTERNatywne wersje rzeczywistości



The screenshot shows a terminal window titled "git-dev-warsztaty-presentation: All files - gitk". The window displays a list of commits on the left and a file tree on the right. A red arrow points to the "Tree" tab at the bottom of the interface.

Commits:

- Set for-what-section text
- Correct contact link in don't waste time section
- Write about workshops content
- Implement home banner buttons
- Set proper facebook url
- Set proper twitter url
- Set proper linkedin url
- Set proper items in mobile navbar
- Set instagram url
- Fix collapsing navbar width
- Fix navbar height
- Fix link to contact section
- Set link in navbar

SHA1 ID: 9fe89a395c34e7e10098a1261f278761bcd1f7d

Find: commit containing:

Search:

Diff **Old version** **New version** Lines of context: 3 Ignore space change

Author: Tometchy <tometchy@gmail.com> 2019-08-30 15:39:57
Committer: Tometchy <tometchy@gmail.com> 2019-08-30 15:39:57
Parent: [c6199d0dbbd1fa11c8c7f0f22327abcc212cb3c](#) (Set proper linkedin url)
Child: [2ca258e2562f2ce8df45cbe667668bafc73a2b99](#) (Set proper facebook url)
Branches: [extend_landing_page](#), [git_jest_git_i_co_dalej_evening](#), [master](#), [remotes/origin/extend_landing_page](#), [remotes/tomrepo/bisect_demo](#), [remotes/tomrepo/git_jest_git_i_co_dalej_evening](#), [remotes/tomrepo/implement_forms](#), [remotes/tomrepo/master](#), [remotes/tomrepo/temp](#), [remotes/tomrepo/update_blog_posts](#)
Follows:
Precedes:
Set proper twitter url

Patch **Tree**

Comments:

- .gitattributes
- .gitignore
- .idea/
- 404.html
- BingSiteAuth.xml
- GHOST.txt
- Gemfile
- Gemfile.lock
- LICENSE
- More Themes.url
- README.md
- README.txt
- Rakefile
- _config.yml
- _data/
- _includes/
- _layouts/
- _notes/
- _plugins/

SZYBKIE SPRAWDZANIE STARYCH WERSJI

- Szybkie i precyzyjne przeglądanie historii
- Szybkie *przepinanie się* na starą wersję
- Szybkie *podglądarki* starych wersji
- Szybki *Blame* do każdej linii w pliku
- Szybkie odszukiwanie commitów



MODYFIKOWANIE KODU WPROST Z GITA

Częste scenariusze lokalnych zmian kodu **bez commitowania**

- Wywal zmiany z pliku Y (`git reset Y; git checkout Y`)
- Wyzeruj wszystkie zmiany (`git reset --hard`)
- Wyzeruj zmiany ale odłóż na potem (`git stash --include-untracked`)
- Zastaguj zmianę X z pliku Y (`git add add Y -p`), resztę z pliku wywal (`git checkout Y`)
- Zastaguj wszystkie zmiany z wszystkich plików (`git add .`), ale wyklucz zmianę X z pliku Y (`git reset Y -p`) i wywal (`git checkout .`)

GIT POSIADA LICZNE NARZĘDZIA I TRICKI USPRAWNIAJĄCE



ALIASY

Modyfikacje aliasów można wykonywać w gitconfigu

```
[alias]
st = status
ci = commit -v
```

Albo za pomocą polecenia

```
git config --global alias.st status
git config --global alias.ci 'commit -v'
```

Przykład użycia: `git st`

Aliases mogą być też na poziomie *Basha* - wówczas nie trzeba pisać *Git*.

PRZYKŁADY ALIASÓW

```
vd = difftool --tool=vimdiff  
  
vds = difftool --tool=vimdiff --staged  
  
wip = !git add --all && git commit -m WIP  
  
pfl = push --force-with-lease  
  
standup = log --since '1 day ago' --oneline --author krzysztof.morcinek@gmail.com # use your mail  
  
cam = commit -am  
  
jira = log --since '6am' --oneline --author krzysztof.morcinek@gmail.com # use your mail  
  
ls = log --pretty=format:"%C(yellow)%h%Cred%d\\ %Creset%s%Cgreen\\ [%cn]" --decorate  
  
mt = mergetool  
  
mff = merge --ff-only  
  
cpc = cherry-pick --continue  
  
rei = rebase --interactive  
  
rec = rebase --continue  
  
# list aliases  
  
a = "!f() { git alias; }; f"  
  
alias = ! git config --get-regexp ^alias\\. | sed -e s/^alias\\./ -e s/\\\\ /\\\\ =\\\\ /
```

PULL --REBASE

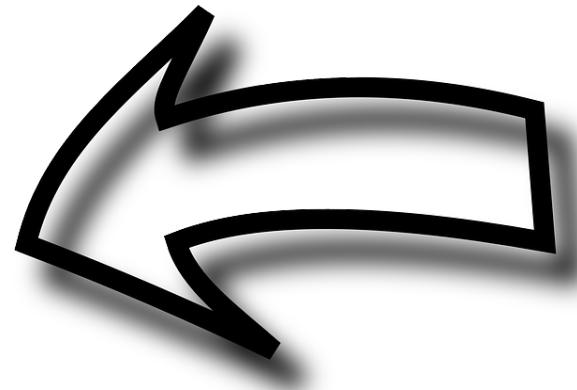
```
git pull --rebase
```

```
git config --global --bool pull.rebase true
```



CHECKOUT LAST USED BRANCH

git checkout -



INTENT TO ADD UNTRACKED FILES

-N

--intent-to-add

Np.

git add . -N

Przykładowe korzyści:

- działa git add -p
- można użyć git diff

Record only the fact that the path will be added later. An entry for the path is placed in the index with no content. This is useful for, among other things, showing the unstaged content of such files with git diff and committing them with git commit -a.

7 ZASAD DOBREGO COMMIT MESSAGE

1. Oddziel tytuł od ciała pustą linią
2. Ogranicz tytuł do 50 znaków
3. Stosuj wielkie litery w tytule
4. Nie kończ tytułu commita kropką
5. Zapisuj tytuł w trybie rozkazującym
6. Ogranicz 'szerokość' ciała do 72 znaków
7. Opisz *co* i *dłaczego*, a nie *jak*

Oprócz punktu 6. (szerokość ciała), wszystkie punkty robią znaczącą różnicę.
Testowałem każdy z nich w innych wariacjach, te naprawdę mają sens.

GIT COMMIT > GIT COMMIT -M

```
This title i a bit too long, see text colors in this editor Too long text indication
This text is in wrong line Text in wrong line indication
Here is fine. And bellow you can see which changes are to be committed and which are not.
Moreover with -v flag you can see introduced changes (bellow).
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch git_jest_git_i_co_dalej_evening
# Your branch is up to date with 'tomrepo/git_jest_git_i_co_dalej_evening'.
#
# Changes to be committed:
#       modified:   wycisnij-soki-z-gita/index.html
#
# Changes not staged for commit: See which files got staged, which didn't
#       modified:   index.html
#
# ----- >8 -----
# Do not modify or remove the line above.
# Everything below it will be ignored.
diff --git a/wycisnij-soki-z-gita/index.html b/wycisnij-soki-z-gita/index.html
index 9939fda..786c0d3 100644
--- a/wycisnij-soki-z-gita/index.html
+++ b/wycisnij-soki-z-gita/index.html
@@ -268,6 +268,10 @@ cover: img/prezentacja.jpg
           </aside>
         </section>
+        <section> See changes (-v flag required)
+          <h3><code>git commit</code> > <code>git commit -m</code></h3>
+        </section>
```

Those colors are configured in VIM by default

GIT COMMIT -V

```
Update meld slide
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch git_jest_git_i_co_dalej_evening
# Your branch is up to date with 'tomrepo/git_jest_git_i_co_dalej_evening'.
#
# Changes to be committed:
#       modified:   wycisnij-soki-z-gita/index.html
#
# Changes not staged for commit:
#       modified:   wycisnij-soki-z-gita/index.html
#
# ----- >8 -----
# Do not modify or remove the line above.
# Everything below it will be ignored.
diff --git a/wycisnij-soki-z-gita/index.html b/wycisnij-soki-z-gita/index.html
index d606ff7..17bc943 100644
--- a/wycisnij-soki-z-gita/index.html
+++ b/wycisnij-soki-z-gita/index.html
@@ -278,6 +278,12 @@ cover: img/prezentacja.jpg

        <section>
            <h3>Meld</h3>
+
            <small>
+
                <ul>
+
                    <li>Możliwość diffowania tekstów bez plików</li>
+
                    <li>Możliwość edytowania z poziomu diffa</li>
+
                </ul>
+
            </small>
```

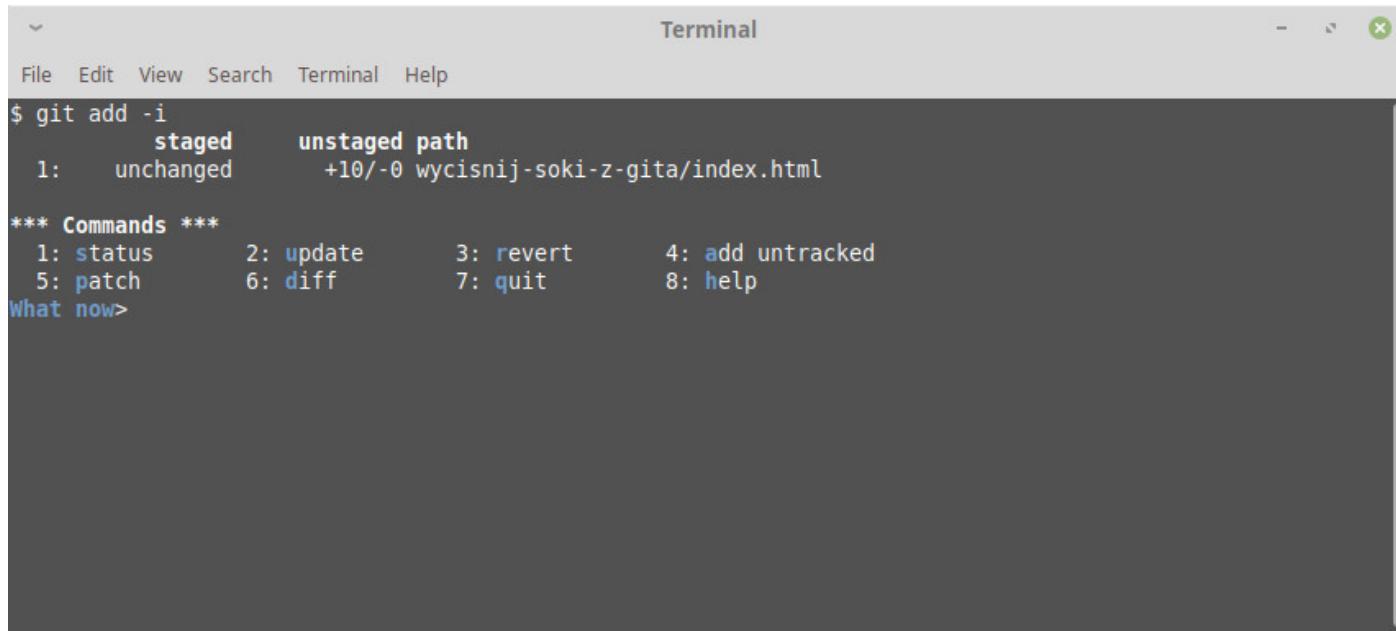
GIT STASH WITH NAME

git stash save "my stash"



GIT ADD INTERACTIVE

git add -i



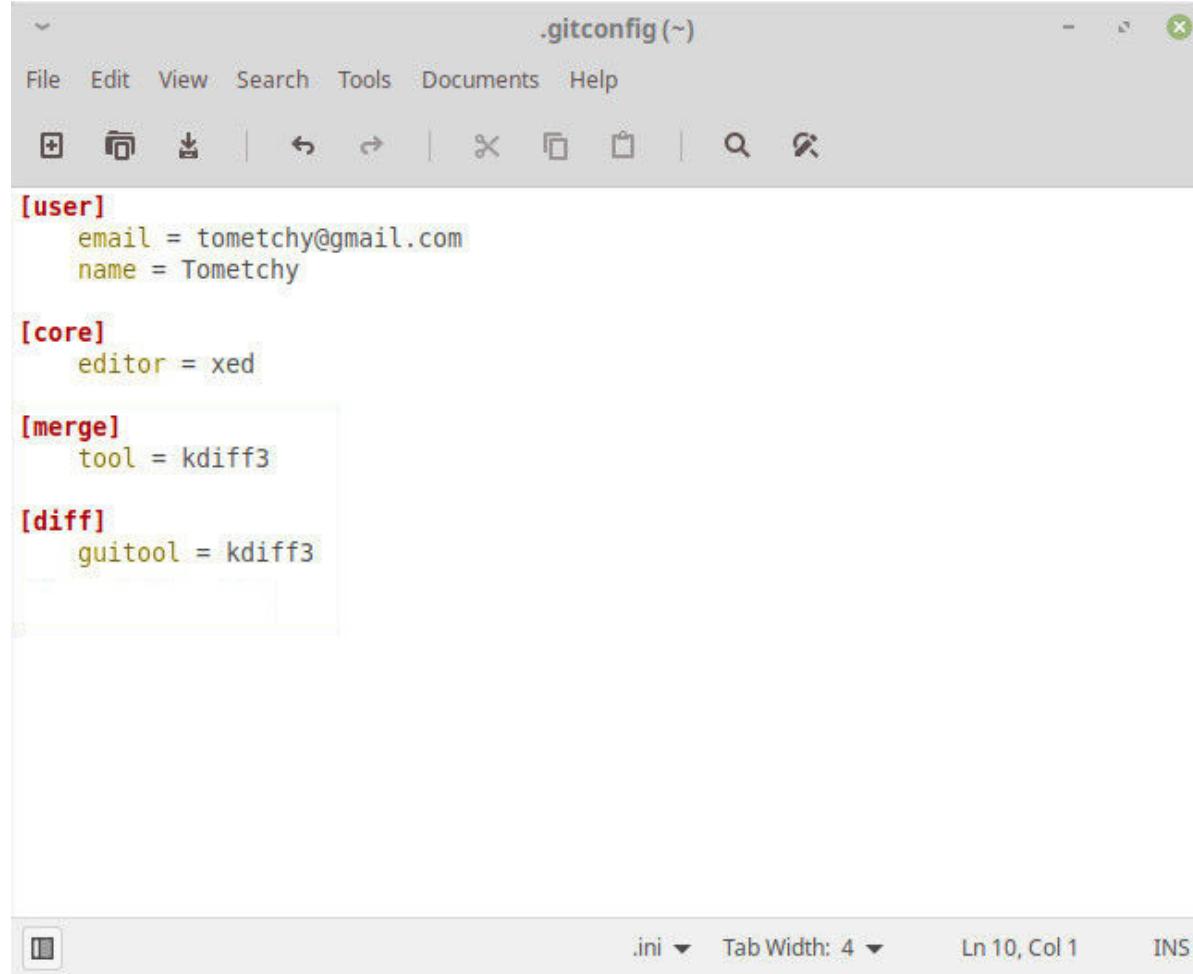
A screenshot of a terminal window titled "Terminal". The window has a standard OS X-style title bar with icons for minimizing, maximizing, and closing. The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The main terminal area shows the following output:

```
File Edit View Search Terminal Help
$ git add -i
      staged      unstaged path
 1:  unchanged      +10/-0 wycisnij-soki-z-gita/index.html

*** Commands ***
 1: status       2: update       3: revert       4: add untracked
 5: patch       6: diff        7: quit        8: help
What now>
```

OTWIERANIE GITCONFIGA Z KONSOLI

git config --global -e



The screenshot shows a text editor window titled ".gitconfig (~)". The file contains the following configuration:

```
[user]
email = tometchy@gmail.com
name = Tometchy

[core]
editor = xed

[merge]
tool = kdiff3

[diff]
guitool = kdiff3
```

The bottom status bar indicates ".ini" and "Tab Width: 4".

CIEKAWE PRZEŁĄCZNIKI DO GIT MERGE

--ff

--no-ff

--ff-only

--squash

STRATEGIE MERGOWANIA

- resolve
- recursive
 - ours
 - theirs
 - patience
 - diff-algorithm=[patience|minimal|histogram|myers]
 - ignore-space-change
 - ignore-all-space
 - ignore-space-at-eol
 - ignore-cr-at-eol
 - renormalize
 - no-renormalize
 - no-renames
 - **find-renames[=n]**
 - rename-threshold=*n*
 - subtree[=*path*]
- octopus
- ours
- subtree

GIT PATCH

Pozwala np. na łatwe przerzucanie zmian pomiędzy repozytoriami nie związanymi ze sobą.

Przykład:

```
git format-patch HEAD~2
# Output:
# 0001-make-stuff-more-awesome.patch
# 0002-allow-users-to-be-locked.patch

cp *.patch ANOTHER_REPO_PATH

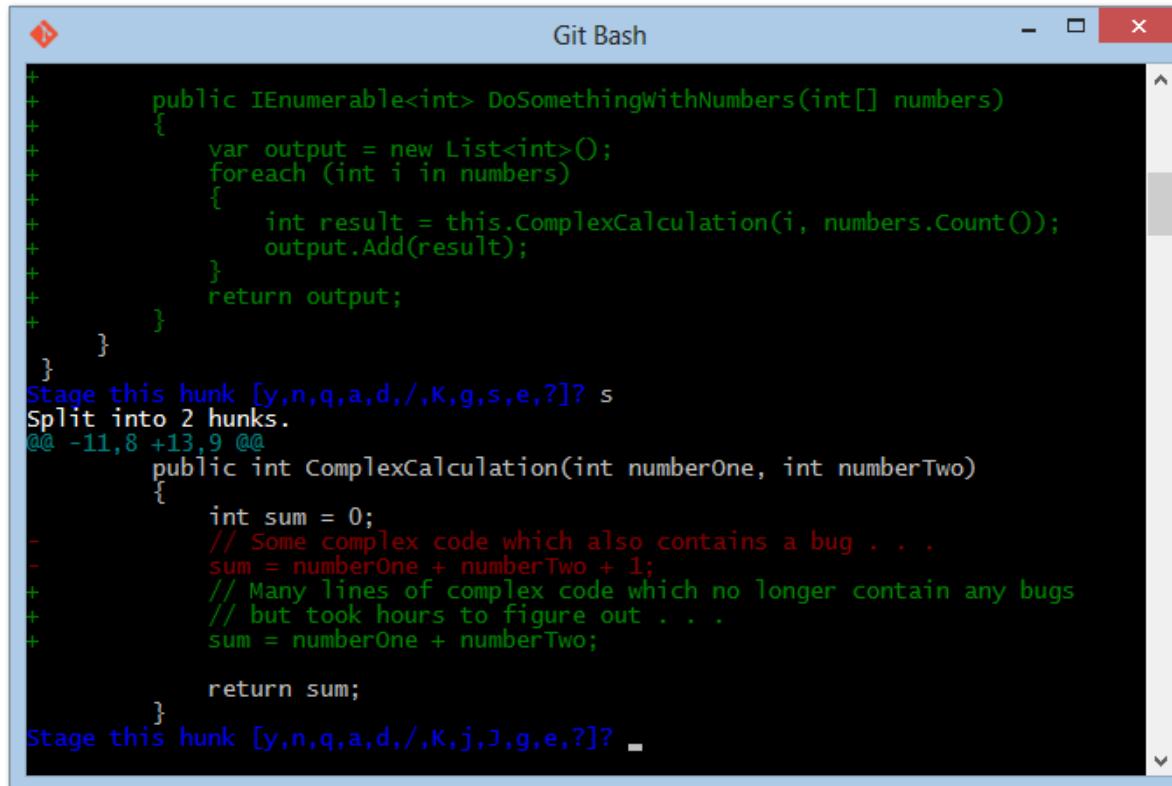
cd ANOTHER_REPO_PATH

git am *.patch
```

STAGOWANIE/RESETOWANIE PO KAWAŁKU

git add -p

git reset -p



The screenshot shows a Git Bash window with the title "Git Bash". The terminal displays a diff of a C# file. The top part of the diff shows a large hunk of code being added:

```
+     public IEnumerable<int> DoSomethingWithNumbers(int[] numbers)
+     {
+         var output = new List<int>();
+         foreach (int i in numbers)
+         {
+             int result = this.ComplexCalculation(i, numbers.Count());
+             output.Add(result);
+         }
+         return output;
+     }
}
```

Below this, there is a prompt: "Stage this hunk [y,n,q,a,d,/,K,g,s,e,?]? s". The terminal then continues with another hunk of code:

```
@@ -11,8 +13,9 @@
     public int ComplexCalculation(int numberOne, int numberTwo)
    {
        int sum = 0;
-        // Some complex code which also contains a bug . . .
-        sum = numberOne + numberTwo + 1;
+        // Many lines of complex code which no longer contain any bugs
+        // but took hours to figure out . . .
+        sum = numberOne + numberTwo;

        return sum;
    }

```

At the bottom of the terminal, there is another prompt: "Stage this hunk [y,n,q,a,d,/,K,j,J,g,e,?]? -".

NOWA KONWENCJA --*DELETE*

Można używać zamiast starej :*name*

Np. usuwanie zdalnego brancha:

```
git push remote_name --delete branch_name
```

GIT PUSH --FORCE-WITH-LEASE



GIT RERERE

REuse REcorded REsolution of conflicted merges

What rerere does is save the resolution of a conflict so that it can be re-applied later if it sees the same conflict again. When Git sees the conflict which it already has a resolution recorded for, it will apply the resolution automatically for you, and give you the opportunity to accept the resolution as applied, or change it.

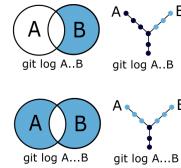
- `git config --global rerere.enabled true`
- `git rerere` (*przed lub po rozwiązaniu konfliktu*)

GIT LOG

Popularne przełączniki do git log

```
git log --after="2014-02-12T16:36:00-07:00"  
git log --before="2014-02-12T16:36:00-07:00"  
git log --since="2 weeks 3 days 2 hours 30 minutes 59 seconds ago"  
Wyszukiwanie commitów na podstawie czasu  
  
git log -4  
Oznaczanie liczby commitów (w tym przypadku 4 commity)  
  
git log --oneline  
Skompresowanie commitów jedna linijka per commit  
  
git log --graph  
Wizualizacja commitów w grafie  
  
git log --decorate  
Dodanie do commitów adnotacji o branchach/tagach itp.  
  
git log --stat  
Wylistowanie zmodyfikowanych/dodanych/usuniętych plików  
  
git log --patch  
Ukazanie diffów - zmian wprowadzonych przez dany commit (skrót: -p)  
  
git log origin/develop --not develop  
Commity z brancha origin/develop, których nie ma na branchu develop(„lokalnie”)  
  
git log develop..origin/develop  
Commity z brancha origin/develop, których nie ma na branchu develop(„lokalnie”)  
  
git log develop..origin/develop  
Commity z brancha origin/develop(„serwera”), których nie ma na branchu develop(„lokalnie”),  
oraz commity z brancha develop(„lokalnie”), których nie ma branch origin/develop(„serwera”)  
  
git log --author="Toma*"  
Commity, których autor treści pasuje do regexa „Toma*”  
  
git log --committer="Toma*"  
Commity, które zostały stworzone przez osobę pasującą do regexa „Toma*”  
  
git log -grep="scope*" -i  
Commity których wiadomość pasuje do regexa „scope*” (case insensitive; -i)  
  
git log -S"com.arellomobile.mvp.MvpPresenter" -i  
Commity które dodają/usuwają frazę „com.arellomobile.mvp.MvpPresenter”, tak zwany  
pickaxe (case insensitive; -i)  
  
git log -all  
Commity z wszystkich branchy (nie tylko z „obecnego”)  
  
git log --no-merges  
Wykluczanie commitów łączących z wyników  
  
git log --pretty="--FORMAT"  
Zmiana wyświetlanego formatu commita,  
np. git log --pretty="%an %H"  
https://devhints.io/git-log-format
```

www.GitWarsztaty.pl



GitWarsztaty.pl/materialy/git-log.pdf

GIT KILOF

Pickaxe - git log -S text

Warto łączyć z innymi, np. od razu podać *-p*



STAN CAŁEGO SYSTEMU Z DOWOLNEGO COMMITA

The screenshot shows a terminal window titled "git-dev-warsztaty-presentation: All files - gitk". The window displays a list of commits on the right and a detailed commit log on the left.

Commits:

- Set for-what-section text
- Correct contact link in don't waste time section
- Write about workshops content
- Implement home banner buttons
- Set proper facebook url
- Set proper twitter url
- Set proper linkedin url
- Set proper items in mobile navbar
- Set instagram url
- Fix collapsing navbar width
- Fix navbar height
- Fix link to contact section
- Set link in navbar

SHA1 ID: 9fe89a395c34e7e10098a1261f278761bcd1f7d

Commit Log (Left Panel):

```
Author: Tometchy <tometchy@gmail.com> 2019-08-30 15:39:57
Committer: Tometchy <tometchy@gmail.com> 2019-08-30 15:39:57
Parent: c6199d0dbbd1fa11c8c7f0f22327abcc212cb3c (Set proper linkedin url)
Child: 2ca258e2562f2ce8df45cbe667668bafc73a2b99 (Set proper facebook url)
Branches: extend_landing_page, git_jest_apis_dalej_evening, master,
          remotes/origin/extend_landing_page, remotes/tomrepo/bisect_demo,
          remotes/tomrepo/git_jest_apis_dalej_evening,
          remotes/tomrepo/implement_forms, remotes/tomrepo/master, remotes/tomrepo/temp,
          remotes/tomrepo/update_blog_posts
Follows:
Precedes:

  Set proper twitter url
```

File Tree (Right Panel):

- Comments
- .gitattributes
- .gitignore
- .idea/
- 404.html
- BingSiteAuth.xml
- GHOST.txt
- Gemfile
- Gemfile.lock
- LICENSE
- More Themes.url
- README.md
- README.txt
- Rakefile
- _config.yml
- _data/
- _includes/
- _layouts/
- _notes/
- _plugins/

A red arrow points to the "Tree" tab in the bottom right corner of the interface.

MELD

- Możliwość diffowania tekstów bez plików
- Możliwość edytowania z poziomu diffa

The screenshot shows the Meld diff tool interface. The top menu bar includes 'Meld', 'File', 'Edit', 'Changes', and 'View'. Below the menu is a toolbar with icons for '+' (New), '-' (Delete), 'Save', 'Undo', 'Redo', and 'X' (Close). The main window displays two code snippets side-by-side. A vertical dashed line separates the left and right panes. The left pane contains the following code:

```
return Expression.Lambda<Func<TGeometry,>
    Expression.Call(
        right,
        left,
        typeof(TGeometry).GetRuntimeMethod(
            typeof(TGeometry).GetRuntimeMethod(
                right),
                right),
        right,
        left,
        left,
        right);
```

The right pane contains a similar code snippet with some differences highlighted by colored arrows:

```
return Expression.Lambda<Func<TGeometry,>
    Expression.Call(
        left,
        left,
        typeof(TGeometry).GetRuntimeMethod(
            typeof(TGeometry).GetRuntimeMethod(
                right),
                right),
        right,
        left,
        left,
        right);
    right;
```

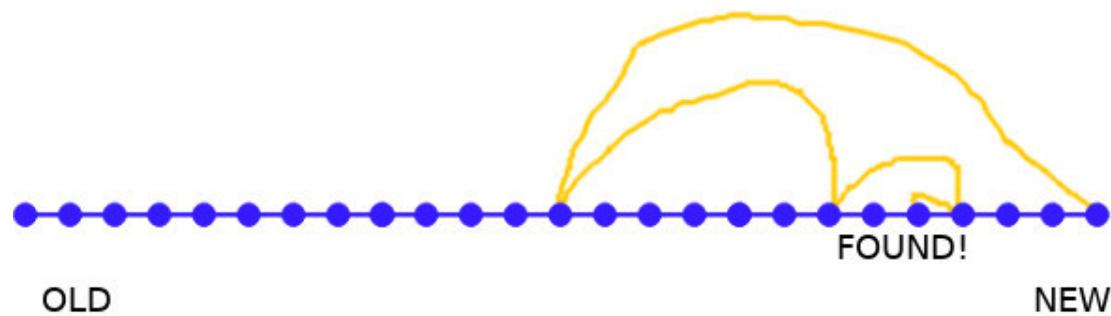
Annotations in the left pane highlight the first four lines of code. Annotations in the right pane highlight the first four lines of code. A green horizontal bar at the bottom of the right pane highlights the last two lines of code. The status bar at the bottom right of the window shows 'Ln 8, Col 15 INS'.

PRZYDATNE NARZĘDZIA

- GitLab pages
- GitHub pages
- Netlify
- GitHub Gist

GIT BISECT

Znowu sprawdziło się tworzenie małych commitów ;)



GITA NIE ODDAM!

Mam nadzieję, że od teraz

Ty również :)



DZIĘKUJĘ ZA UWAGĘ

Życzę wszystkim, żeby z pomocą Gita
praca posuwała się do przodu w zawrotnym tempie

