

Maciej Aniserowicz

Grupa .NET Białystok, 28.03.2012

Spojrzenie na Gita (okiem niewolnika TFS)

O sobie samym

- architekt / programista / team leader
- blogger (www.maciejaniserowicz.com)
- ekspert CodeGuru.pl
- moderator devPytania.pl
- Microsoft Certified Professional
- Microsoft MVP
- 3 etaty w Warszawie
- ponad 2 lata freelance
- aktualnie: team leader w Predica

Agenda

- TFS
- Git
- Git-TFS

Ja i kontrola wersji

- 2007 – 2009 (studia i 3 etaty)
 - TFS->SVN->TFS->SVN
- 2009 – 2012 (freelance, praca)
 - SVN->Git->Mercurial (HG)->Git->HG->TFS-> Git

TFS

TFS dla...

- ...„czytelnika raportów”: 10/10
 - PM / product owner / ...
 - klient
 - team leader
- ...„modyfikatora work itemów”: 5/10
 - team leader
 - programista
- ...„pisarza/czytelnika kodu”: 1/10
 - programista
 - team leader

Problemy z TFS (1/2)

- model scentralizowany
- online-only („local workspaces” z TFS v11 nic tu nie zmieniają)
- wydajność
- promowanie złych praktyk
 - wielka parująca kupa... kodu

Problemy z TFS (2/2)

- diff / merge
- exclusive checkouts – strach przed merge
- do wszystkiego potrzebny VS
- pliki readonly
- niezmienione pliki w „pending changes”
- kolumna „last check-in” w „source control”
- ...

Git

Korzyści z Gita (1/2)

- praca offline
 - commity
 - gałęzie
 - diff
 - log
- nie da się nic zepsuć/stracić/nadpisać

Korzyści z Gita (2/2)

- „multitasking”
 - praca nad 2 zadaniami na zmianę
 - szybka poprawka buga „w międzyczasie”
- szczegółowa historia
- „czysta/działająca” wersja co X minut
- niezależna praca kilku programistów w tym samym obszarze
- ogólne zadowolenie zespołu

Podstawowe pojęcia

- „remote” – link do zdalnego repozytorium
 - ssh, https, share, lokalny folder...
- „origin” – domyślnie zdalne repo źródłowe
- „master” – domyślnie gałąź „główna”
 - „main” w TFS
 - „trunk” w SVN
 - „default” w HG
- „HEAD” – ostatni commit gałęzi
- „git bash” – konsola gita
- git [command] -- help

Komendy

komunikacja zdalna

- clone – ściągnięcie repozytorium
- pull – ściągnięcie zmian ze zdalnego repo
 - = fetch + merge
- push – wysłanie zmian do zdalnego repo

Komendy lokalne repozytorium

- `init` – założenie repozytorium
 - `--bare` – „centralne” repo (bez working copy)
- `checkout` – zmiana wersji working copy
 - `-b` – nowa gałąź
 - z nazwą pliku – ściąga konkretną wersję pliku
- `branch` – wypisuje gałęzie
 - `-d` – kasuje gałąź
- `status` – aktualny stan plików w working copy

Git demo 1

Git submodules

- jak svn:externals... tylko lepiej
- wykorzystanie konkretnej wersji innego repo
 - współdzielona biblioteka
 - projekt open source
 - własny rozwijany niezależnie moduł...

Komendy submodules

- submodule
 - init – dodanie modułów do konfiguracji
 - update – ściągnięcie odpowiedniej wersji zarejestrowanych modułów

Git demo 2

Git index

- „staging area”
- między working copy a nowym commitem
- dokładna kontrola nad zawartością commita
- dowolna manipulacja zawartością indeksu

Komendy praca z kodem

- add – dodanie zmian do indexu
 - -A – dodanie wszystkich zmian
 - -i – „interactive add”
 - -i -> patch – dodanie części zmian z jednego pliku
- commit – stworzenie commita
- clean – czyszczenie nieśledzonych plików
- reset – czyszczenie indeksu
 - --hard – czyszczenie indeksu i zmian z working copy
- stash – zachowanie zmian na później
 - lokalny „shelve”

Git demo 3

Komendy

modyfikacja historii

- `commit --amend` – modyfikacja ostatniego commita
- `rebase` - „nałożenie” aktualnych zmian na inną gałąź
 - `-i` – „interactive rebase”: zmiana commit msgs, kolejności i zawartości commitów...
- nie wolno modyfikować publicznej historii

Git demo 4

.gitignore

- wzorce nazw plików do zignorowania
- wersjonowany wraz z projektem
- *.user, *.suo, *.cache
- bin/, obj/
- *.resharper, *_ReSharper*, *_Resharper*
- może ignorować samego siebie!

TFS i Git

Gdzie TFS a gdzie Git?

- TFS...

- wszędzie gdzie był
- raporty
- buildy
- zarządzanie zadaniami

- Git

- dev-maszyny
- uwolnienie dev od mąk kontroli wersji w TFS

Git-TFS

- „proxy” między lokalnym Gitem a zdalnym TFSem
- niewidoczne dla reszty zespołu...
- ...choć największe korzyści jeśli wszyscy używają

Git-TFS workflow

- `git tfs clone https://<Host>/<ProjectCollection>$/<ProjectName>`
- `git tfs pull`
- `git checkout -b dev-branch`
- `edit -> git add -> git commit -> ...`
- `git checkout master`
- `git tfs pull`
- `git checkout dev-branch`
- `git rebase master`
- `git tfs checkintool --build-default-comment...`
- `... lub git rcheckin -w <task id>`
- `git checkout master`
- `git tfs pull`
- `git branch -d dev-branch`

Bonusy

Komendy

Wow-factor

- `git bisect` - identyfikacja commita powodującego błąd w systemie, na przestrzeni długiej historii
- `git cherry-pick` - przenoszenie commita pomiędzy gałęziami
- `git reflog` - historia WSZYSTKICH operacji

Hosting

- GitHub
 - publiczne - darmowe
- Bitbucket
 - publiczne i prywatne - darmowe
- Codeplex (od kilku dni)
 - tylko publiczne (?)

Narzędzia

- wbudowane
 - git bash (konsola)
 - gitk (log-viewer)
 - git gui (wykrywa konieczność Git GC)
- Tortoise Git <http://code.google.com/p/tortoisegit>
- Git Extensions
<https://github.com/spdr870/gitextensions>
- Git source control provider
<http://visualstudiogallery.msdn.microsoft.com/63a7e4od-4d71-4fbb-a23b-d262124b8f4c>
- Smart Git
<http://www.syntevo.com/smartgit/index.html>

Podsumowanie

- Wady TFSa
- Zalety Gita
- Jak korzystać z Gita
- Git w TFS

Linki

■ Git

- Git <http://git-scm.com>
- Git-tfs <https://github.com/git-tfs/git-tfs>
- Git - dokumentacja <http://git-scm.com/documentation>
- Git - tutorial <http://schacon.github.com/git/gittutorial.html>
- Git community book <http://book.git-scm.com>
- <http://whygitisbetterthanx.com>
- Linus Torvalds o Gicie: <http://www.youtube.com/watch?v=4XpnKHJAok8>
- A successful Git branching model - <http://nvie.com/posts/a-successful-git-branching-model>
- Git Magic - <http://www-cs-students.stanford.edu/~blynn/gitmagic>

■ Meine Blog

- <http://www.maciejaniserowicz.com/?tag=/git>
- <http://www.maciejaniserowicz.com/?tag=/tfs>
- <http://www.maciejaniserowicz.com/?tag=/version+control>

Pytania...?

Dziękuję i do widzenia

pptx: <https://github.com/maniserowicz/git-talk>

e-mail: maciej.aniserowicz@gmail.com

blog: <http://www.maciejaniserowicz.com/>