

Týmová infrastruktura

Y36SI3 - Realizace programových
systémů

Ondřej Macek

Obsah přednášky

- Komunikace
- Repositáře
- Bug tracking

KOMUNIKACE

Co patří do komunikace?

Jak realizovat „komunikaci“?

- Dokumenty
- Modely
- Zdrojový kód

version control a revision control

SPRÁVA KÓDU

Správa zdrojového kódu s centrální repository

- CVS
- Subversion
- Rational ClearCase
- Microsoft Visual Source Safe, Microsoft TFS

Distribovaná správa zdrojového kódu

- Git
- Mercurial
- Darcs (<http://darcs.net/>)
- 1 programátor = 1 repository
 - operace pull
 - operace push
 - reserved edit - možnost zamykání

Základní koncept SCM

- Check out
- Update (Pull, Fetch)
- Commit (Push)
- Revert
- Trunk, Branch
- Merge
- Patch/Changeset
- Blame

CVS

- Rok 1986
- Vzniklo z RCS (verzování jednoho souboru)
- Concurrent Versions System
 - nemá atomický Commit
 - nepodporuje přejmenování a přesun souborů
 - jméno souboru v historii
 - musí se udělat ručně

Microsoft VSS, Microsoft TFS

- Microsoft Visual Source Safe
 - sdílení souborů
 - navrženo pro LAN, na Internetu pomalé
- Microsoft Team Foundation Server
 - Team Foundation Version Control
 - ukládání změn na SQL Server

Subversion

- Rok 2000
 - atomický commit
 - přejmenování a přesun souborů v repository
 - TortoiseSVN – Windows klient
 - svn blame
 - Pro každý řádek ukáže kdy byl naposledy změněn a kým

Git

- Distribuovaná správa zdrojové
- Rok 2005
- Linus Torvalds
- Lokální operace
- Rychlá operace merge
- Název revize kóduje předchozí změny
 - chrání před změnou předcházející verze

Bude updatováno na přednášce +
spousta živých ukázek

Veřejné repository

- <http://sourceforge.net> - CVS, Subversion, Git
<http://tigris.org> – Subversion
- <http://code.google.com/hosting> - Subversion
- <http://www.assembla.com> – Subversion, Git
- <http://www.codeplex.com> - Microsoft TFS
- <http://repo.or.cz> – Git

Code Swarm

- Vizualizace aktivity v softwarovém projektu
- Michael Ogawa
 - <http://vis.cs.ucdavis.edu/~ogawa/codeswarm>

BUILD

Automatický build

- Skript který stáhne aktuální verzi z repository a zkompiluje ji
- Může pustit statické nástroje na analýzu zdrojového kódu
 - Fortify (Java)
 - FxCop (.NET)
- Po kompilaci může pustit unit testy

Příklady skriptovaných buildů

- *make, gnumake, nmake*
- *Ant* (<http://ant.apache.org>)
- *Maven* (<http://maven.apache.org>)

Číslování verzí

XX.YY.ZZ

YY - sudé => stable

YY – liché => development

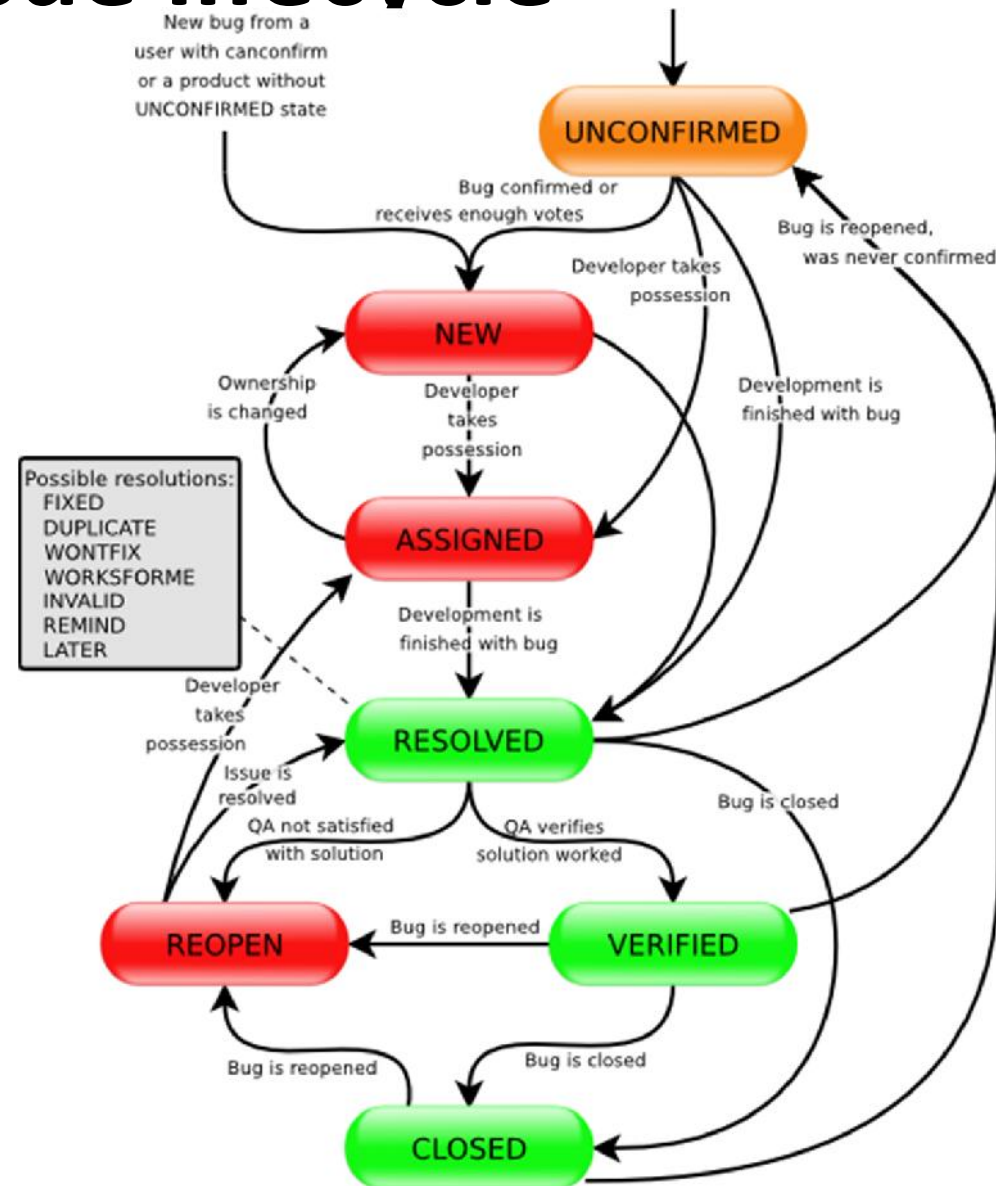
Stable vs. Development

ZZ – ukazuje na opravenou chybu

Issue and Bug tracking

SPRÁVA ÚKOLŮ A CHYB

Bug/issue lifecycle



Painless Bug Tracking

1. Minimální počet kroků v životním cyklu chyby
 - Fixed,
 - Won't fix
 - Postponed
 - Not Repro
 - Duplicate
 - By Design
2. Bug smí zavřít jenom ten kdo ho otevřel
3. Verze software

Painless Bug Tracking (2)

6. Programátor přijme pouze bug v databázi, ne jinou cestou
7. Pokud kolegové nepoužívají databázi, stejně jim přiřazujte bugy
8. Jako vedoucí týmu naučíte programátory používat databázi zadáváním nových požadavků
9. Nepřidávejte nová pole
10. Vše jasně popište

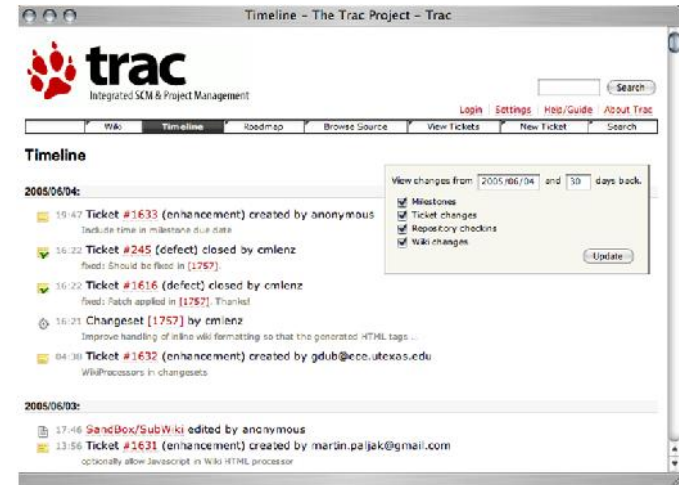
Nástroje

- <http://www.bugzilla.org/>
- <http://www.mantisbt.org/>
-

WIKI

Trac

- Rok 2006
- Wiki
- SCM
 - Subversion
 - moduly pro Git,Mercurial,Darcs
- Bug tracking



Zajímavá literatura

SPOSKY, Joel. *Joel on Software* [online]. 2000 [cit. 2010-09-08]. Painless Bug Tracking. <<http://www.jelonsoftware.com/articles/fog0000000029.html>>.

- SVN <http://subversion.apache.org/>
- Git <http://git-scm.com/>
- Mercurial <http://mercurial.selenic.com/>
- Bugzilla <http://www.bugzilla.org/>
- Mantis <http://www.mantisbt.org/>
- Ant <http://ant.apache.org>
- Maven <http://maven.apache.org>
- Trac <http://trac.edgewall.org/>
- Wiki http://en.wikipedia.org/wiki/Comparison_of_wiki_software