

# Mercurial SCM



Paweł Pacana  
<pawel.pacana@starware.com.pl>



# Intro (1)

## mercurial

(adj) relating to or having characteristics (eloquence, swiftness, cleverness) attributed to the god Mercury.

## Mercurial

(n) a fast, lightweight Source Control Management system designed for efficient handling of very large distributed projects.

# Intro (2)

- 2005, Matt Mackal
  - ohloh.net podaje 300+ kontrybutorów
- 2008, v1.0
- 2009, v1.4
  - time-based release plan
  - wydanie co 4 miesiące

# Intro (3)

- napisany w języku Python
- kluczowe wydajnościowe moduły implementowane w C
  - 11% kodu
  - diff, mpatch

Python	59%
shell script	28%
Other	13%



# Intro (4)

- podstawowy SCM
  - Mozilla
  - Python
  - NetBeans
  - OpenOffice
  - OpenJDK
  - OpenSolaris
  - MoinMoin
  - Xen
  - Symbian Platform
  - Go
  - Xemacs
- mirror
  - Linux Kernel
  - Rails
  - GNU Emacs
  - GCC
  - FreeBSD

# Intro (5) – Hot

Bram Moolenaar, 7 Sty, 17:28 napisał:

Od: Bram Moolenaar <B...@Moolenaar.net>

Data: Thu, 07 Jan 2010 17:28:02 +0100

Lokalna: Czw. 7 Sty 2010 17:28

**Temat: Mercurial repository available for Vim**

I have setup a Mercurial repository. It contains the same files that are in CVS, plus the updated runtime files.

# Filozofia (1)

- tiny core + extensions
  - to co robi ma robić dobrze i szybko
  - 1 binarka (144, Git?)
  - niestandardowe rzeczy w extensions (pluginy vs. kitchen sink design, Git?)
- wieloplatformowy
  - unix
  - mac os
  - windows
    - wspiera nawet „junction files” – dowiązania NTFS

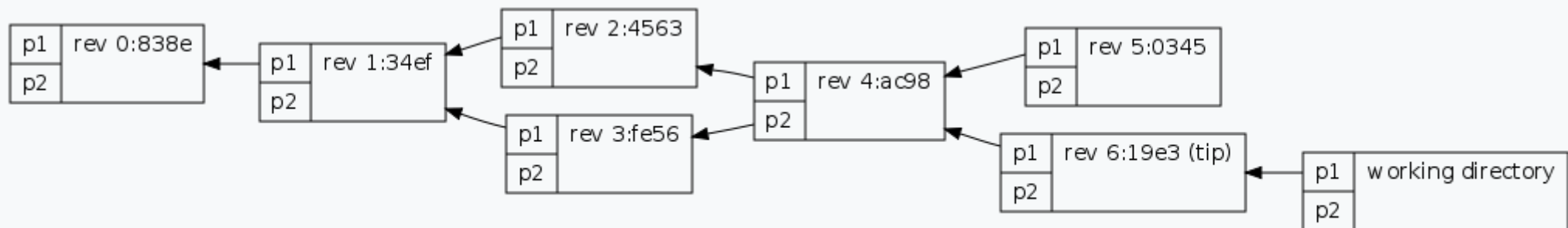
# Filozofia (2)

- *„history is sacred”*
  - core nigdy nie przepisuje historii
    - Git: *„rewriting history is a badge of honor”*
  - changeset ID jest wyliczany na podstawie ID przodków
    - zmiana przodka powoduje zmianę ID potomków
- prosty i przejrzysty UI
  - nie można zrobić sobie krzywdy
  - fałszywe przekonanie o słabości
    - extensions!



# Model

- historia jako DAG
- Changeset
  - head - bez dzieci
  - tip – najnowszy
  - root – bez przodków
- commit – 1 przodek
  - merge commit – 2 przodków
- branch name
  - z pierwszego rodzica
- lokalne numery rewizji oprócz SHA1



example repository

# Internals

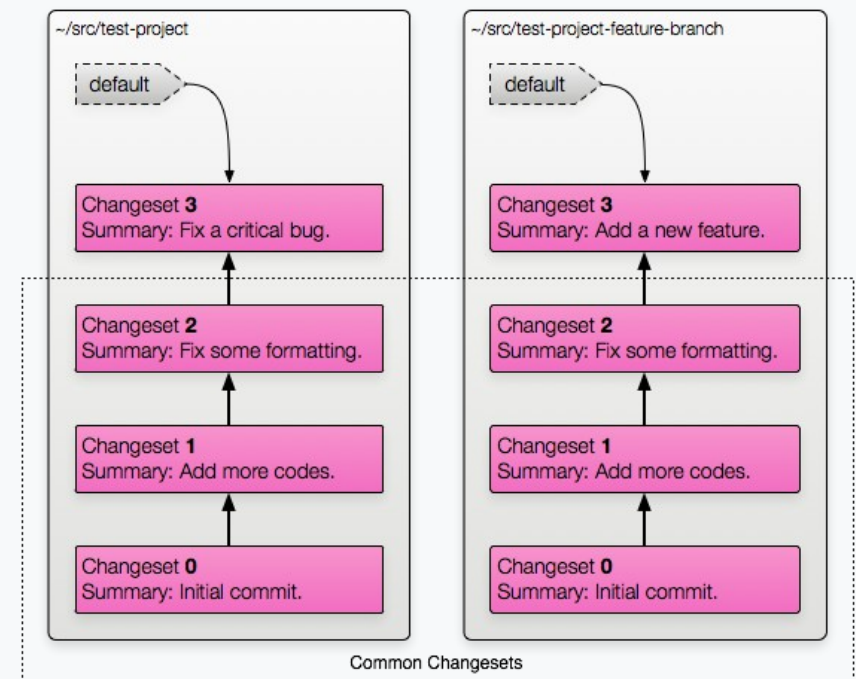
- delta i snapshoty
- append-only
- Wire protocol
  - kompresja

# Branch & Merge (1)

- branching w Mercurialu
  - „Heavy branch”
    - hg clone
  - „Named branch”
    - hg branch
  - „Anonymous branch”
    - hg update do dowolnego changesetu i hg commit
  - „Bookmark”
    - BookmarkExtension

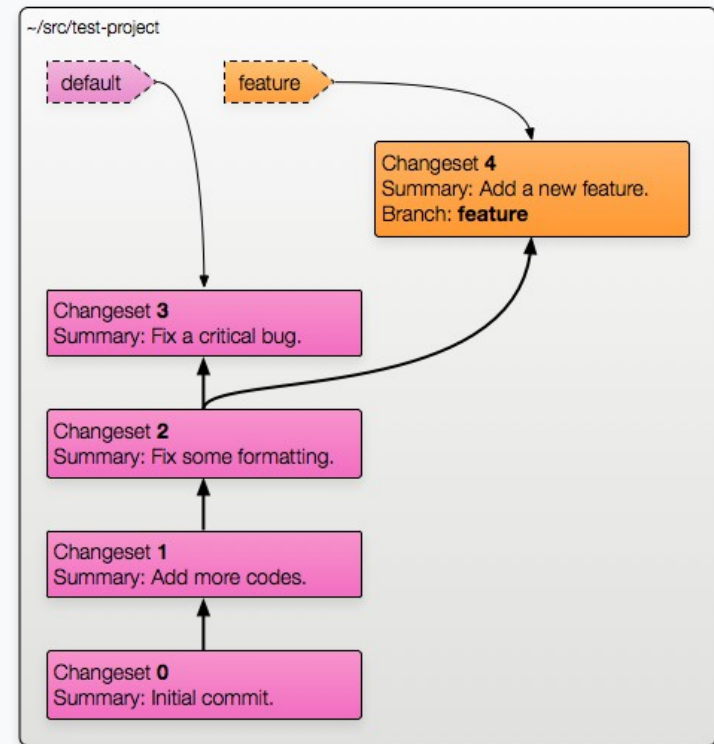
# Branch & Merge (2)

- „Heavy branch”
  - zewnętrzny
  - tworzy osobne repozytorium na dysku
    - `hg clone test-project test-project-feature-branch`
      - hardlinki, copy-on-write
  - całkowita izolacja gałęzi
  - usunięcie gałęzi
    - `rm -rf test-project-feature-branch`
  - zmiana ścieżek



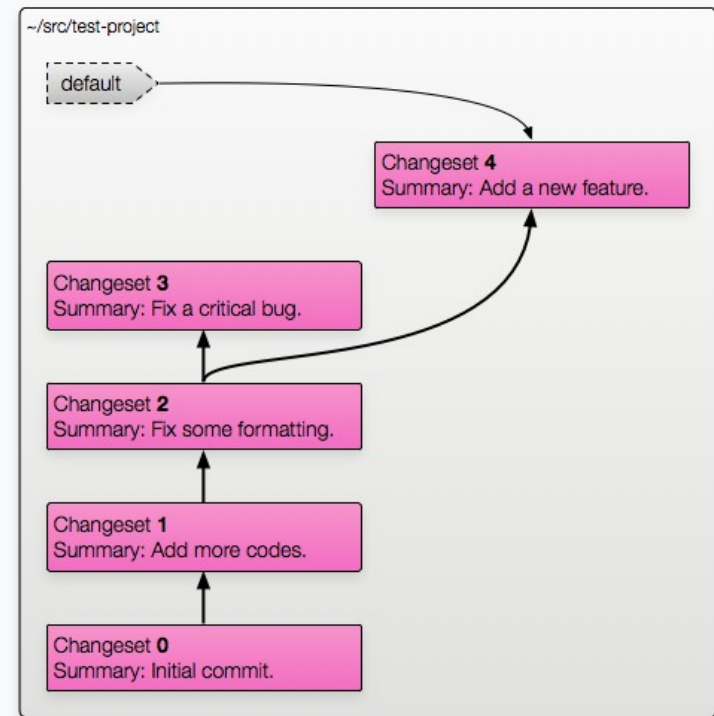
# Branch & Merge (3)

- „Named branch”
  - wewnątrz repozytorium
  - `hg branch branchname`
  - `hg commit` dołącza changeset do brancha rodzica
  - domyślny branch to default
  - branchname jest dołączany jako metadana do changesetu
    - czytelniejsze
    - z metadanych można korzystać w hookach
    - nieusuwalne (bo historia jest nieusuwalna), ale można je zamknąć i nie pokażą się w `hg branches`



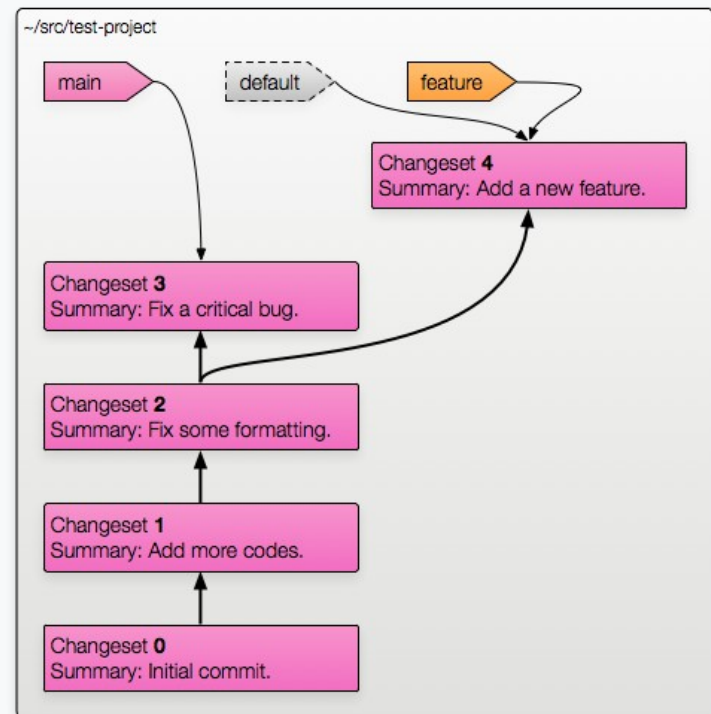
# Branch & Merge (4)

- „Anonymous branch”
  - hg update #cset i hg commit
    - hg update ustawia rodzica
  - przełączanie między gałęziami wymaga podania hasza/rewizji changesetu
  - hg pull (i inne) tworzą takie gałęzie
  - najszybszy i najłatwiejszy
    - wymaga pisania dobrych commit messages
    - lżejszy niż „Git lightweight branches” ;)



# Branch & Merge (5)

- „Bookmark”
  - wewnętrzne (takie jak branche anonimowe)
  - etykieta do changesetu
    - .hgbookmarks
    - można je usuwać
- lokalne
  - nie można ich transferować
  - Mercurial 1.5?

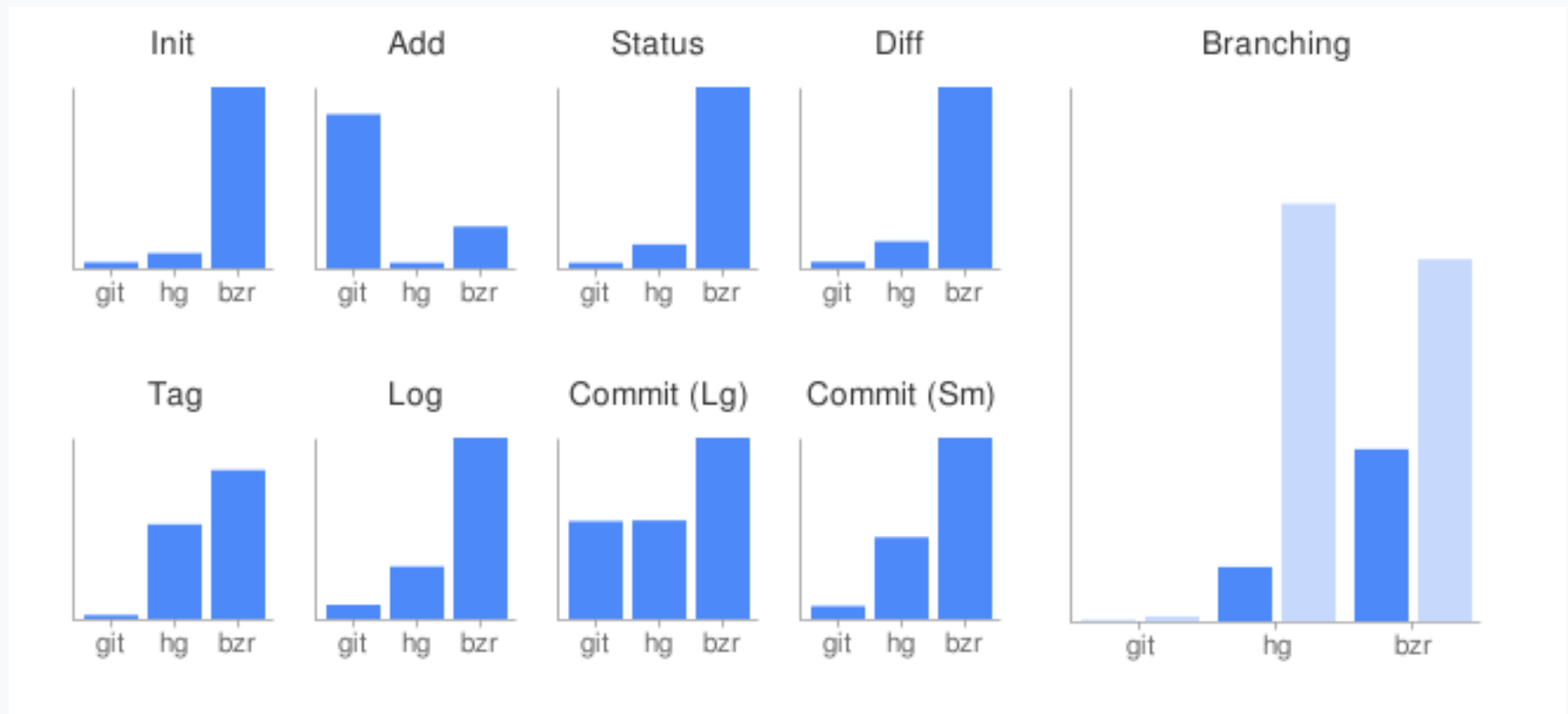


# Git vs. Hg (1)

- [whygitisbetterthanx.com](http://whygitisbetterthanx.com)
  - github
    - bitbucket
      - brakuje tylko masowego hype'a
  - cheap local branches
    - clone i hardlinks
    - BookmarkExtension
    - anonymous branching
  - staging area
    - dirstate
    - hg record?
- git concepts
  - historia jako DAG
  - tree – manifest
  - blob – file
  - commit - changeset



# Git vs. Hg (2)



# Subrepos

- podobne do externals z SVN
  - Hg
    - stabilne
  - SVN
    - w hg-crew
  - Git
    - w drodze, patche na ML

# Extensions (1)

- Transplant
- Rebase
- Histedit
- Bookmarks
- Bisect
- Record
- Codesmell
- Alias
- Kerberos
- Schemes
- Acl
- Hg-Git
- HgSubversion
- + 44 inne

# Extensions (2)

```
#!/usr/bin/env python
from mercurial import hg

def printparents(ui, repo, node, **opts):
    # coś się tu robi
    cmdtable = {
        # cmd name          function call
        "print-parents": (printparents,
                          [(['s', 'short', None, 'print short form'],
                            ['l', 'long', None, 'print long form'])],
                          "[options] REV")
    }
```

# Ekosystem (1)

- hosting
  - BitBucket
    - „pythonowy” GitHub
  - GoogleCode
  - SourceForge

# Ekosystem (2)

- continuous integration/bug tracking
  - Hudson
  - CruiseControl
  - BuildBot
  - Bamboo
  - TeamCity
  - JIRA
  - Trac
  - FogBugz
  - Redmine

# Ekosystem (3)

- IDE
  - Vim
  - Emacs
  - TextMate
  - Eclipse
  - NetBeans
  - IntelliJ
  - VisualStudio

# Ekosystem (4)

- konwersja innych SCM do Hg
  - CVS
  - Subversion
  - Git
  - Darcs
  - Monotone
  - Bzr
  - Arch
  - Mercurial
- można przemapować userów, pominąć pliki



# Mercurial – Happy End

Pytania?

Paweł Pacana  
<pawel.pacana@starware.com.pl>



# Mercurial - Epilog

- Pycon 2010
  - Scott Chacon – git evangelist
    - “Hg and Git : **Can’t we all just get along?**”