

Systemy Kontroli Wersji

zarządzanie źródłami

Marcin Gryszkalis

dagoon@math.uni.lodz.pl

Wydział Matematyki Uniwersytetu Łódzkiego
Łódź 2003

-
-
-

VCS a SCM



-
-
-
-
-
-
-
-
-
-

VCS a SCM

- VCS = *Version Control System*
- SCM = *Software Configuration Management*
- VCS \in SCM

Systemy SCM

- Zarządzanie kodem źródłowym (VCS)

Systemy SCM

- Zarządzanie kodem źródłowym (VCS)
- Zarządzanie zmianami i obsługa błędów

Systemy SCM

- Zarządzanie kodem źródłowym (VCS)
- Zarządzanie zmianami i obsługa błędów
- Zarządzanie budowaniem

Systemy SCM

- Zarządzanie kodem źródłowym (VCS)
- Zarządzanie zmianami i obsługa błędów
- Zarządzanie budowaniem
- Obsługa procesów tworzenia (zatwierdzanie itp.)

Systemy SCM

- Zarządzanie kodem źródłowym (VCS)
- Zarządzanie zmianami i obsługa błędów
- Zarządzanie budowaniem
- Obsługa procesów tworzenia (zatwierdzanie itp.)
- Zarządzanie projektem

Systemy VCS

- Repozytorium plików

Systemy VCS

- Repozytorium plików
- Śledzenie zmian

Systemy VCS

- Repozytorium plików
- Śledzenie zmian
- Odtwarzanie stanów

Systemy VCS

- Repozytorium plików
- Śledzenie zmian
- Odtwarzanie stanów
- *Regression bugs*

Systemy VCS

- Repozytorium plików
- Śledzenie zmian
- Odtwarzanie stanów
- *Regression bugs*
- Współpraca wielu autorów (programistów) potrzebujących dostępu do repozytorium

Jak działa system VCS

CVS: login

- `login` - logowanie do repozytorium
- konieczne przy zdalnym dostępie
- może bazować na różnych metodach autentykacji (`pserver`, `GSSAPI`)

```
$ cvs -d :pserver:user@example.pl:/home/cvsroot login
Logging in to :pserver:user@example.pl:/home/cvsroot
CVS password:
$ cvs -d :pserver:user@example.pl:/home/cvsroot checkout foo
```

CVS: login

- login - logowanie do repozytorium
- konieczne przy zdalnym dostępie
- może bazować na różnych metodach autentykacji (pserver, GSSAPI)

```
$ export CVSROOT=:pserver:user@example.pl:/home/cvsroot
$ cvs login
Logging in to :pserver:user@example.pl:/home/cvsroot
CVS password:
$ cvs checkout foo
```


CVS: checkout

- `checkout` - pobranie pliku/projektu z repozytorium
- można operować na całych drzewach katalogów lub tzw. modułach
- można operować na różnych *wersjach* pliku
- `update` - pobranie pliku z repozytorium (uaktualnienie)

CVS: commit

- `commit` - zapisanie zmian w repozytorium
- czasami określane jako "check in"
- automatycznie zwiększa numer wersji
- oczekuje komentarza

Wersje

- pliki są w różnych wersjach (*ang. revision*)
- aplikacje są w różnych wydaniach (*ang. release*)

CVS: diff

- `diff` - porównanie różnych wersji pliku
- domyślnie porównuje lokalną kopię z aktualną wersją w repozytorium

CVS: log

- log - historia zmian danego pliku
- wyswietla liste zmian (komentarzy, które zostały podany podczas operacji `commit`)

CVS: annotate

- `annotate` - zawartość pliku, z oznaczeniem kto co zrobił
- czasami oznaczane jako 'blame'

Gałęzie

- Jeden projekt może być rozwijany w kilku wydaniach jednocześnie
- np. jądro Linuxa: 2.2.x, 2.4.x, 2.5.x
- możliwość wielokrotnego rozdzielania gałęzi
- możliwość łączenia gałęzi

Wydania - tagi

- Oznaczenie konkretnego momentu w życiu projektu
- Pozwala odnosić się do konkretnego wydania
- Pozwala wygodnie rozgałęziać
- tag HEAD - oś rozwoju

Słowa kluczowe

- Specjalne wzorce podmieniane w czasie operacji zapisu (`commit`)
- `Id` - charakterystyczny identyfikator
`$Id: actions.txt,v 1.9 2003/04/10 01:22:55 mg Exp $`
- `$Date$` - data zapisu wersji
- `$Author$` - autor (kto zapisał)
- `$Revision$` - numer wersji

Dostępne systemy VCS i SCM

Systemy VCS

- Open Source: RCS, CVS, Subversion
- Pół-komercyjne: BitKeeper, Perforce
- Komercyjne: SCCS, Microsoft SourceSafe, Forte Code Management Software, Rational ClearCase, Borland StarTeam
- Inne: VMS Filesystem

SCCS

- *Source Code Control System*
- Stary komercyjny system VCS
- Obecnie znaczenie raczej historyczne
- CSSC - Odpowiednik z rodziny GNU
- <http://cssc.sourceforge.net>

RCS

- *Revision Control System*
- Klasyczny system kontroli wersji
- OpenSource, Licencja - GPL
- Bezpośredni dostęp do plików wersji
- Tylko lokalny
- Bazuje na blokadach
- <http://www.gnu.org/software/rcs/rcs.html>

CVS

- *Concurrent Versions System*
- OpenSource, Licencja - GPL
- Format plików "prawie" zgodny z RCS
- "Concurrent" - pozwala na jednoczesne wprowadzanie zmian przez wiele osób
- <http://cvshome.org>

Subversion

- Zaprojektowany jako następca CVSa
- OpenSource, Licencja - Apache/BSD-style
- Dużo rozszerzeń względem CVSa
- Cały czas w trakcie intensywnego rozwoju
- <http://subversion.tigris.org>

BitKeeper

- Komercyjny system kontroli wersji
- Udostępnia możliwość darmowej obsługi projektów OpenSource
- Jądro Linuksa
- <http://bitkeeper.com>

Perforce

- Komercyjny, nowoczesny SCM
- Używany przez: SAP, Nvidia, SGI
- Możliwość darmowego wykorzystania przy projektach OpenSource (m.in. OpenWatcom)
- <http://www.perforce.com>

Microsoft SourceSafe

- Komercyjny VCS
- Standard dla narzędzi MS
- Oparty na współdzielonym katalogu
- Microsoft zaleca - nie przekraczać 3GB
- Nie obsługuje stref czasowych
- <http://msdn.microsoft.com/ssafe/>

Forte Code Management Software

- Komercyjny produkt Sun Microsystems
- Dawniej Sun TeamWare, oparty na SCCS
- Kompletny system SCM
- <http://www.sun.com/software/...>

Rational ClearCase

- Komercyjny kompletny SCM
- Wspiera standard UCM (Unified Change Management)
- Prawdopodobnie jeden z najsilniejszych SCM wogóle
- Bardzo duże wymagania sprzętowe (serwer)
- <http://www.rational.com/products/clearcase/>

Borland StarTeam

- Komercyjny kompletny SCM
- <http://www.borland.com/starteam/index.html>

System plików VMS

- System plików ze wsparciem dla kontroli wersji
- Dostępny w systemach VMS, także OpenVMS (dla platform VAX i Alpha)
- Obejmuje wszystkie działania w systemie
- <http://www.openvms.org>

Wady i Zalety: cvs, snv, bk

CVS - cechy

Zalety

- Dojrzały system
- Bardzo szerokie wsparcie, integracja z narzędziami
- Standard

Wady

- Problemy z plikami binarnymi i nie-ASCII
- Brak obsługi rename

Subversion - cechy

- Wersjonowanie katalogów, rename, symlinks
- Atomic commits
- Apache/WebDAV jako podstawa komunikacji sieciowej
- Algorytmy obsługi plików binarnych (także diff!)
- Łatwo parsowalne wyjście
- Wsparcie dla plików tekstowych nie-ASCII (UTF-8 i inne)

BitKeeper - cechy

- Hierarchiczne repozytoria
- Changesets
- Rollback (nazwy, zawartość)
- Triggery
- Dostępny na większości platform (klient?)
- Sumy kontrolne

CVS - narzędzia

CVS - klienci

- command-line
- SmartCVS (Java, zamknięty, darmowy w wersji "community")
- WinCVS (GTK, GPL)
- TortoiseCVS (tylko Windows, GPL)
- pluginy

CVS - klienci www

- Bonsai
- ViewCVS (Python)
- CVSWeb

CVS - inne

- cvsup (klient i serwer)
- CVS-NT (serwer)

-
-
-



Pytania?



-
-
-
-
-
-
-
-
-
-

Prosper

Prezentacja wykonana przy użyciu pakietu

`prosper`

dla systemu $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

<http://prosper.sourceforge.net>