

Development methodologies in Web 2.0 timeframe

Fakultetska znanja

- difference between bottom-up composition and top-down decomposition
- potreba za učenjem dovoljno teškog predmeta na fakultetima (LISP-olikog, FP) da potakne razmišljanje i filtrira studente
 - ◆ nekima više pa je znanstveni rad i teoretska istina a drugima praktični problemi
- ISO 9001 certifikacija ili kako system architect postaje poslovna
- prednost studentata FOI-ja je razumijevanje sustava end-to-end
 - ◆ papiri su dio IS-a
 - ◆ komunikologija, organizacija, ekonomija, marketing

Revival of 4GL-like tools for Web 2.0 world

- Service Oriented Architecture (SOA) - kockice od kojih se grade rješenja
 - ◆ biblioteke (.so, .dll)
 - ◆ komponente, moduli (Java .jar, CPAN za perl)
 - ◆ servisi (SOAP, XML/RPC, REST)
- Not Invented Here (NIH) simptom polako umire (kao posljedica uspjeha kolaborativnog Web-a)
- napokon RAD (Rapid Application Development) alati za Web
 - ◆ projekti od nekoliko dana, a ne mjeseci
 - ◆ CRUD (Create/Read/Update/Delete)
 - ◆ ORM (Object-Relational-Mappers)
 - ◆ RDBMS vs Object Storage (XML?) vs full-text search

Frameworks

- Ruby on Rails
- Maypole, Catalyst, Jifty
- Strix?

Model View Controller (MVC) paradigma

Zašto su skriptni jezici različiti?

Weakly typed, nema kompajliranja, brži razvoj

- JavaScript (u browseru)
- php (server, nema podjele logika/prikaz, sličan asp-u)
- perl (server, cgi, više različitih MVC frameworka)
- python (izgled striktno zadan, odličan end-to-end jezik client i server, jako dobar za početnike)
- ruby (potpuno OO, sličan JavaScript-u)

FLOSS

Najveći izvor kvalitetnog gotovog koda - don't re-invent the wheel

- Kultura
 - ◆ Eric S. Raymond: scratch own itch
 - ◆ Linus Torvalds: given enough eyeballs all bugs will be swallowed
- open source: katedral, bazar
- free software, GPL
- licence: BSD/GPL/OSI/dual-licensing/proprietary

Code re-use dileme

NaÅ in pisanja *upstream* koda uvelike odreÅ uje njegovu jednostavnost integracije

- proceduralno
- Object Oriented
- Multiple Inheritance
- Mix-in
- closures

Koliko su aktivni *upstream* developeri? Kako prihvaÅ aju promjene?

RaspoloÅivi kod Å esto odreÅ uje platformu implementacije u FLOSS svijetu!

Svaki programer treba znati viÅje alata da bi mogao odabrati najbolji:

- automatizacija - bash
- kernel modul - C
- web site - skriptni jezik
- GUI na klijentu - python, C#

Jezik mora biti dovoljno fleksibilan da moÅe pratiti promjenjive zahtjeve.

Principi za uspjeÅan Web 2.0 projekt

Kako je Internet promjenio pogled na metodologije razvoja informacijskih sustava?

Kolaboracija koriÅtenjem Interneta kao komunikacijskog medija

- atribucija (minimalni uvijet, motivacija)
- mailing liste
- wiki
- bug tracking (RT, Peregrine, SourceForge, trac)
- SCM (CVS, Subversion, git)

Diljenje znanja

- preduvijet za uspjeÅan projekt
- physical proxymity (15m rule)
- follow the sun - distribuirani development timovi
- projektna organizacija
- product manager, project manager - razlike u pogledima i funkcijama na projektu

Dobra dokumentacija

- osnova za dobru komunikaciju i dijeljenje znanja
- razlika između u programerske i korisničke dokumentacije

hackable/tweakable (mogućnost modifikacije)

- Proizlazi iz osnovne motivacije *scratch own itch*
- poboljšava mogućnost, povećava brojka contributora (povremenih) i developera

Testiranje

0 dokumentacija

0 testovi

0 pisanje koda

-> rezultat bolji API

Buzzwords

- Web 2.0, beta (forever)
- REST, AJAX

Pojednostavljenje produkcije i održavanja

IS-ovi se koriste mnogo duže nego što je planirano (Y2K, Cobol).

- KISS - anti NIH
- profit
- TCO

WWW - izvor informacija

- blogs
- SIG sites (www.perlmonks.org)
- www.itconversation.com
- podcasts

XP kolegij na FOI-ju

- Unit testing
- peer programming
- 20-30% programera
- 24.4.2006. 16-18 unutar kolegija programiranje