

Tehnologija programske opreme

- Predavatelj: doc. Bojan Kverh
(bojan.kverh@fri.uni-lj.si)
- Govorilne ure: četrtek, 15h-16h
- Asistent: dr. Aleš Jaklič
- eUčilnica (<http://ucilnica.fri.uni-lj.si>):
prosojnice, novice, forum...

Ocenjevanje

- Seminarske naloge (50%)
 - analiza
 - načrt
 - aplikacija
 - testiranje
- Pisni izpit, naloge (35%)
- Teorija (15%)
 - testi v okviru predavanj

Predavanja

- Sreda, 16h-19h, PR06
- Prosojnice
- Spoznavanje teorije in različnih pristopov k razvoju kompleksne programske opreme
- Ilustracija s praktičnimi primeri
- Vprašanja
- Diskusija

Vsebina predavanj (1/2)

- Uvod
- Proces razvoja PO
- Različni modeli procesov
- Agilne tehnologije
- Praksa razvoja programske opreme
- Zbiranje zahtev
- Analiza problema

Vsebina predavanj (2/2)

- Načrt rešitve problema
- Arhitekturni načrt
- Načrt uporabniških vmesnikov
- Testiranje
- Kvantitativno ocenjevanje programske opreme
- Tehnike za razvoj spletnih aplikacij
- Upravljanje projektov

Literatura

- Roger S. Pressman: Software Engineering, A practitioner's approach
- Steve McConnell: Rapid development
- Robert C. Martin: Agile Software Development: Principles, Patterns and Practices
- Martin Fowler: Patterns of Enterprise Application Architecture
- Franc Solina: Projektno vodenje razvoja programske opreme

Seminarske naloge

- Kompleksen projekt (1 do 2 projekta)
- Več skupin (skupina = 5-6 študentov)
- Vsaka skupina poskrbi za svoj del sistema
- Od vsake skupine se pričakuje nek končen izdelek
- Analiza problema, načrt, aplikacija!
- Integracija in testiranje!

1. Uvod

- **Motivacija:** veliko kompleksnih projektov propade zaradi slabega razumevanja problema ali slabega načrtovanja
- Tehnologije in postopki za hitrejši, natančnejši in bolj sistematičen razvoj programske opreme in njeno vzdrževanje
- Programska oprema je najpomembnejša tehnologija sodobnega sveta!
- Včasih: 1 projekt – 1 programer
- Danes: 1 projekt – več skupin specialistov za posamezne dele projekta, ki skupaj tvorijo delujočo aplikacijo

Vprašanja:

- 1) Zakaj traja razvoj programske opreme tako dolgo?
- 2) Zakaj je razvoj programske opreme tako drag?
- 3) Zakaj ne moremo najti vseh napak v aplikaciji, preden aplikacijo damo končnim uporabnikom?
- 4) Zakaj porabimo veliko časa in truda za vzdrževanje obstoječih aplikacij?
- 5) Zakaj je težko meriti napredek pri razvoju programske opreme?

Programska oprema

- Zbirka ukazov (aplikacija), ki ob izvršitvi opravi želene operacije v dovolj kratkem času
- Podatkovne strukture, ki omogočajo aplikacijam da ustrezno obdelajo informacije
- Dokumentacija, ki ustrezno opisuje operacije in uporabo aplikacij

Programska oprema je logični in ne fizični element

Programska oprema vs strojna oprema

- Programska oprema ni izdelana v klasičnem smislu!
- Programska oprema se ne iztroši, vendar se vseeno poslabšuje
- Čeprav v industriji prevladujejo standardne komponente, je programska oprema večinoma razvita za vsak projekt posebej

Vrste programske opreme

- Sistemska programska oprema
- Aplikativna programska oprema
- Inženirska/znanstvena programska oprema
- Vgrajena programska oprema
- Splošnonamenska programska oprema
- Programska oprema namenjena zabavi
- Spletne aplikacije
- Umetna inteligenca

Stara programska oprema

- Slabša kvaliteta (kvaliteta kode, arhitektura, dokumentacija, dnevnik sprememb, ...)
- Dokler dela dovolj zanesljivo, je vse v redu
- Do sprememb mora priti ob naslednjih dogodkih:
 - PO mora biti prilagojen za potrebe novih okolij in tehnologij
 - PO mora biti nadgrajen zaradi novih poslovnih zahtev
 - PO mora biti nadgrajen zato, da lahko sodeluje z novimi sistemi in PB
 - PO mora biti prilagojena za delo v mreži
- Vsaka PO se s časom razvija (spreminja)! Popravljanje napak, novo okolje, nove zahteve, predelava.

Popularni miti (upravljalci)

- Imamo že knjigo standardnih postopkov za razvoj aplikacij. To bo mojim ljudem zagotovilo vse, kar morajo vedeti za razvoj aplikacije
- Če bomo s projektom v zaostanku, lahko vedno začasno zaposlimo nove programerje in projekt bo dokončan v roku.
- Če razvoj aplikacije prepustimo zunanjim izvajalcem, se lahko mirno lotimo drugih stvari, saj bodo oni poskrbeli zato, da bo stvar uspešno narejena.

Popularni miti (naročniki)

- Seznam splošnih zahtev je dovolj za začetek programiranja; detajle bomo dodali kasneje
- Zahteve projekta se stalno spreminjajo, vendar pa se spremembe zlahka integrirajo v programsko opremo, saj je programska oprema zelo fleksibilna.
- Sedaj, ko je razvit uporabniški vmesnik, bo projekt zelo hitro dokončan

Popularni miti (razvijalci)

- Ko dokončamo program, je naše delo opravljeno
- Preden lahko zaženemo program, je nemogoče soditi o njegovi kvaliteti
- Edini produkt uspešnega projekta je delujoč program
- Z uvedbo postopkov za sistematičen razvoj PO (software engineering) bomo morali napisati ogromno nepotrebne dokumentacije, kar nas bo upočasnilo pri dokončanju projekta

2. Proces razvoja PO

- Proces razvoja PO = ogrodje za vse naloge, ki so potrebne za razvoj kvalitetne PO
- Software engineering = uporaba sistematičnih, discipliniranih in preverljivih pristopov k razvoju in vzdrževanje PO. Uporaba inženirskih pristopov v razvoju PO.
- Nivoji SE: orodja, metode, procesi, zagotavljanje kvalitete

Proces razvoja PO → aktivnosti, krovne aktivnosti

Aktivnost → SE akcije (zbirka povezanih akcij, ki proizvedejo del produkta)

SE akcija → posamezne delovne naloge

Generičen proces razvoja PO: komunikacija, planiranje, modeliranje, razvoj, inštalacija

Krovne aktivnosti: sledenje razvoja projekta, upravljanje s tveganji, zagotavljanje kvalitete, formalni tehnični pregledi, meritve, zagotavljanje ponovne uporabe, itd.

Modeliranje

- **Analiza:** zbiranje zahtev, razčlenjevanje zahtev, pogajanje z naročniki, formalna specifikacija zahtev
- **Načrtovanje:** načrtovanje podatkovnih struktur, arhitekturni načrt, načrt uporabniškega vmesnika, načrt komponent

Primer: enostaven projekt

- **Komunikacija:**
 - **ugotavljanje zahtev naročnikov:**
 - 1) Naredi seznam vseh naročnikov projekta
 - 2) Povabi vse naročnike na neformalni sestanek
 - 3) Vsakega naročnika prosi, naj naredi seznam željenih zahtev in funkcionalnosti
 - 4) Predebatiraj seznam željenih funkcionalnosti in naredi končni seznam
 - 5) Uredi funkcionalnosti po prioriteti
 - 6) Označi dvoumne zahteve

Primer: kompleksen projekt (1/2)

- **Komunikacija:**
 - **ugotavljanje zahtev naročnikov:**
 - 1) Naredi seznam vseh naročnikov projekta
 - 2) Povabi vsakega naročnika na individualen razgovor glede željenih zahtev in funkcionalnosti
 - 3) Naredi preliminarni seznam zahtev
 - 4) Splaniraj serijo sestankov z namenom poenostavitve in konsolidacije seznama zahtev
 - 5) Izvedi sestanke

Primer: kompleksen projekt (2/2)

- 6) Izdelaj neformalne uporabniške scenarije na osnovi izvedenih sestankov
- 7) Izboljšaj uporabniške scenarije na osnovi pripomb naročnikov
- 8) Izdelaj končni seznam naročniških zahtev
- 9) Uredi seznam po prioriteti zahtev
- 10) Grupiraj povezane zahteve tako, da jim lahko zadostimo v eni iteraciji razvoja
- 11) Naredi seznam omejitev aplikacije
- 12) Predebatiraj metode za zagotavljanje pravilnosti delovanja aplikacije

Procesni vzorci

Proces razvoja PO → zbirka vzorcev, ki definirajo aktivnosti, akcije in posamezne delovne naloge, ki so potrebni za razvoj PO

Vzorec: konsistentna, utečena metoda za opis pomembne faze procesa razvoja

Opis vzorca: ime, namen, vrsta, kontekst, problem, rešitev, končni kontekst, sorodni vzorci, primeri uporabe

Omogočajo hierarhičen opis procesa

Vzorci omogočajo ponovno uporabo

Primer vzorca (1/2)

Ime: prototip

Namen: izdelava modela (prototipa), katerega naročniki lahko uporabljajo zato, da bolj natančno opišejo svoje zahteve

Vrsta: vzorec faze

Kontekst: izpolnjeni morajo biti sledeči pogoji: (1) naročniki morajo biti znani, (2) vzpostavljena komunikacija med naročniki in razvijalci, (3) problem, ki naj bi ga rešili, je identificiran s strani naročnikov, (4) znani so namen projekta, osnovne poslovne zahteve in omejitve projekta

Problem: naročniške zahteve so nedoločene, vendar je jasno, da gre za problem, katerega naj bi rešili s PO. Naročniki niso prepričani, kaj želijo; svojih zahtev ne znajo natančno opisati.

Primer vzorca (2/2)

Rešitev: opis izdelave prototipa

Končni kontekst: prototip programske opreme, ki zadošča osnovnim zahtevam, je sprejet s strani naročnikov. Prototip se lahko razvije v končni produkt skozi serijo iteracij, lahko pa ga zavržemo in razvijemo končni produkt samo na osnovi izkušenj, pridobljenih s prototipom.

Sorodni vzorci: komunikacija z naročniki, iterativno načrtovanje, iterativni razvoj, identifikacija zahtev

Primeri uporabe: uporaba prototipov se priporoča, kadar zahteve naročnikov niso popolnoma jasne.