Projektni razvoj aplikacija

Predavanje 8

Držanje zahtjeva pod kontrolom

Uvod

- Zahtjevi koji upadaju u kasnijim fazama projekta
  - Ključan faktor u prekoračenjima budžeta, resursa, rokova
  - Otkazivanje projekta
- Šefovi, korisnici, … programeri?
- *Reći NE mirne savjesti*

Podjela

A. **Kontrola u ranoj fazi**
   - Da ostanemo u skladu sa željenim rokom i/ili budžetom

B. **Kontrola** ulaska novih zahtjeva u projekt **tijekom razvoja**

C. **Odustajanje od zahtjeva** u projektu koji kasni
   - Kako bi se dostigao rok i/ili budžet

A. Kontrola u ranoj fazi

- Ne stavljati nepotrebne zahtjeve u specifikaciju (aplikaciju, proizvod)
- Preporuke:
  1. Minimalna specifikacija
  2. Izbacivanje zahtjeva
  3. Razvoj u fazama

A.1. Minimalna specifikacija

- Koliko detaljna specifikacija mora biti?
- Pretjerano detaljna specifikacija
  - Zastarjelost zahtjeva
  - Netko drugi odlučuje o stvarima o kojima bi programer trebao

A.1. Minimalna specifikacija

Mogućnosti?

- Jednokratna specifikacija
- Upute za korištenje
- Prototip korisničkog sučelja
- Slike/skice iz razgovora

A.1. Minimalna specifikacija

Prednosti

- To postaje proizvod programera
  - Uključen je u osmišljavanje proizvoda
- Ne gubi se vrijeme na specificiranje nečega što programeru možda uopće nije važno

A.1. Minimalna specifikacija

Rizici

- Izostavljanje ključnih zahtjeva
- Nejasni ili neostvarivi ciljevi
- Prirodna težnja programera da „poboljša” proizvod
  - Sprečavanje: pregled koda
- Nemogućnost izvođenja paralelnih aktivnosti
  - Programiranje, testiranje, pisanje specifikacije

A.1. Minimalna specifikacija

Rizici

- Programer vezan za određenu funkcionalnost
- Korištenje minimalne specifikacije iz krivih razloga
  - Lijenost u prikupljanju i pisanju
- Korisnik će pokušati „ugurati” nešto što se „podrazumijeva”
  - Potrebno poznavati korisnika i njegovo ponašanje

A.1. Minimalna specifikacija

Preduvjeti za uspjeh

- Prilagodljivi zahtjevi
- Uočiti ključne zahtjeve i njih specificirati
- Pronaći ključne korisnike i njih uključiti u pisanje specifikacije i kontrolu razvoja
- Fokus na skice, slike, dijagrame umjesto čistog teksta

A.2. Izbacivanje zahtjeva

- Što ranije – to bolje
  - Ušteda na specificiranju, dizajnu, programiranju, testiranju, dokumentiranju
- Manje rizično korištenja minimalne specifikacije
  - Radi se nad gotovom specifikacijom

A.2. Izbacivanje zahtjeva

- Postupak:
  - Izbaciti sve što nije apsolutno nužno
  - Pojednostavniti sve zahtjeve koji su kompleksniji od onog što je potrebno
  - Zamijeniti sve što možemo jeftinijom varijantom zahtjeva

A.3. Razvoj u fazama

- Umjesto potpunog izbacivanja – odgoda funkcionalnosti za kasnije faze
- Ako se zahtjevi u međuvremenu promjene – nismo gubili vrijeme na njihov razvoj

B. Kontrola ulaska tijekom projekta

- Ako smo dobro specificirali zahtjeve – mislimo da imamo projekt (zahtjeve) pod kontrolom
- 25% zahtjeva se promijeni
  - Dodatna/izmijenjena funkcionalnost
  - Bolje razumijevanje poslovnog područja
  - Konkurencija ne spava
  - Programeri sami imaju potrebu mijenjati

B. Kontrola ulaska tijekom projekta

- Nagovaranje pojedinog programera da im se napravi omiljeni/potrebni zahtjev
- Marketing – nakon analize tržišta inzistira na još nekom zahtjevu
- Programeri implementiraju svoje kad ih se ne kontrolira
- 1% mjesečno, nakupi se tijekom dužih projekata

B. Kontrola ulaska tijekom projekta

- Situacije koje pogoršavaju stanje:
  - Izrada superiorne aplikacije
  - Nejasni ili nedostižni ciljevi

- Svima jasno reći koliko prilagodljivu aplikaciju želimo, jasno reći prioritet

B. Kontrola ulaska tijekom projekta

- **Kako promjene utječu?**
- Većina ljudi opuštena vezano za kasne promjene na projektu
  - Ne razmišlja se o: dizajnu, programiranju, testiranju, dokumentiranju, korisničkoj podršci, planiranju, praćenju, …
- Glavni uzrok prekoračenja rokova
- Izuzetno važno kontrolirati nove zahtjeve

B. Savjeti za kontrolu novih zahtjeva

- Inkrementalni razvoj
- Dizajnirati/razvijati imajući na umu fleksibilnost (čak i kad korisnik ne kaže eksplicitno)
- „Odbijati” zahtjeve niskog, prihvaćati zahtjeve visokog prioriteta
- Inicijalno ne zaključati specifikaciju dok programeri ne daju svoje mišljenje
- Ne proglašavati zahtjeve stabilnima, ako to nisu

B. Načini kontrole

- Dozvoljavati samo promjene u cilju poboljšavanja proizvoda
- Dati svim stranama da procjene utjecaj promjena
- Stalno obavještavati da će se rokovi pomaknuti zbog analize novih zahtjeva i eventualne implementacije zahtjeva
- Inzistirati na pisanom zahtjevu za izmjenu, poslovnoj analizu utjecaja, primjerima za testiranje, …
- Umjesto „Ne”, „Da u sljedećoj verziji”

B. Kontrola ulaska tijekom projekta

Analiza zahtjeva

Analiza	Ako nemamo vremena za analizu, zahtjev vjerojatno nije potreban
Stupanj hitnosti	Donijeti odluku, ne odgađati
Grupiranje	Manje košta kada više srodnih zahtjeva grupiramo

C. Odustajanje od zahtjeva

- Kada projekt kasni
- Micanje zahtjeva niskog prioriteta
- Negativnosti:
  - Dio truda bacamo
  - Moguće da moramo uložiti dodatni napor kako bismo maknuli funkcionalnost

Oporavak projekta

Uvod

- Projektu treba prva pomoć
- Kako prepoznati takav projekt?
  - Većina uključenih ne može pretpostaviti kada će projekt biti dovršen
  - More grešaka
  - Ljudi rade prekovremeno
  - Korisnici izgubili povjerenje u razvojni tim
  - Odnosi su zategnuti
  - Projekt na rubu otkazivanja
  - Moral razvojnog tima na niskim granama

Što učiniti?

- Sitnice neće pomoći
- Krupan zaokret potreban
- Smanjiti veličinu
- Povećanje produktivnosti (kratkoročno)
- Suočavanje sa stanjem i odgoda uz posljedice
- Kombinacija

Plan oporavka

- Procjena situacije
  - Koliko je čvrst krajnji rok?
  - Koliko su korisnici spremni pregovarati o funkcionalnosti?
- Što?
  - Mi kao tim moramo napraviti da bismo uspjeli?
  - Što korisnik treba?
  - Kako sačuvati dobar odnos s korisnikom?
- Moguće odustajanje od projekta?
- Biti realan u obećanjima
  - Uteg prošlih neostvarenih obećanja

Ljudi

- Učiniti da se tim osjeća vrijedno
- Sastanci, pritisak nadređenih, …
- Postoje problematični član(ovi)?
  - U timu, voditelj projekta, tehnički voditelj tima
- Dodavanje novih članova?
- Oslobađanje od aktivnosti nepovezanih s projektom

Proces

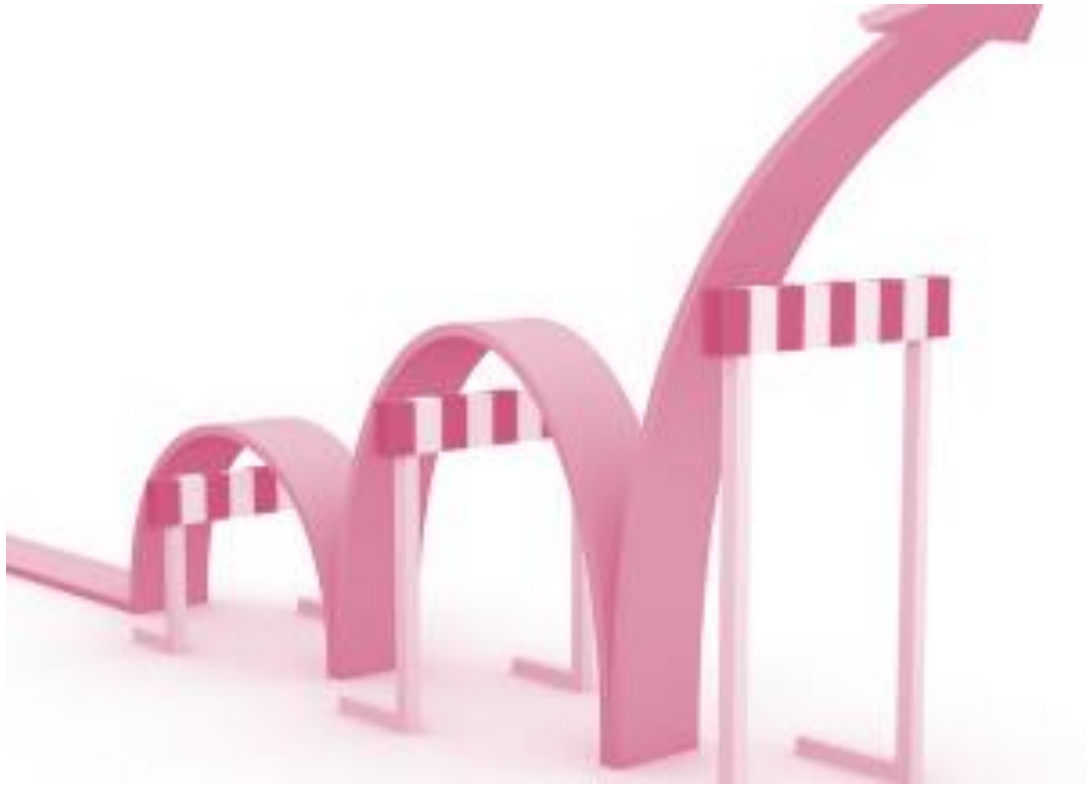
- Moguće greške u procesu:
  - Specifikacija proizvoda koja se mijenja
  - Loš dizajn / neodgovarajuća arhitektura
  - Nedovoljno praćenje tijeka razvoja
  - Srozavanje kvalitete kako bi se dostigao rok
  - Nerealan rok (nekoliko puta pomaknut)
  - Preopterećeni ljudi
  - Gunđalo koje kvari atmosferu ostalima

Popraviti što se može odmah

- Sustav za čuvanje verzija koda
- Sustav za prijavu i praćenje grešaka
- Izolirati tim ukoliko ima previše prekida
- Tim za kontrolu novih zahtjeva
- Davanje odgovora i donošenje odluka

Maleni miljokazi

- Na nivou dana je potrebno znati što mora biti napravljeno
- 1-0 (napravljeno – nije napravljeno)
- Ako postoje drugi lebdeći zadaci – uvrstiti ih u miljokaz
- Djeluje kao motivacijski faktor
- Podesiti plan projekta temeljen na miljokazima
- Definicija gotovog?
  - „Kada bismo uzeli taj kod i ne bismo ga dirali do kraja projekta, …?” – „99% gotovo”



Praćenje izvođenja

- Bilježenje razloga zašto rok nije dostignut
- Ako nekome treba 7 dana za 5 dana procjene, uzeti to u obzir
  - Bez naknadne nadoknade
- Ne davati nove rokove bez da smo se uvjerili da novi plan *drži vodu*

Proizvod

- Stabilizirati zahtjeve
- Smanjiti ih na minimalnu prihvatljivu količinu
- Pronaći *smeće* u kodu (module s puno grešaka)
- Fokus na rješavanje greška