

Metodologije razvoja softvera

Agilne metodologije razvoja softvera

dr Milan Stojkov

Katedra za informatiku

2022.



Fakultet tehničkih nauka
Univerzitet u Novom Sadu

Proizvodnja I

- **Problem I** - proizvodnja mobilnih telefona na pokretnoj traci
 - Moguće je jednoznačno definisati postupke i korake
 - Nakon pravljenja nekoliko telefona moguće je prilično pouzdano procenti vreme i organizovati njihovu proizvodnju

Proizvodnja II

- **Problem II** - gradnja kuće
 - Vlasnik želi da sagradi kuću koristeći „zelene“ materijale i tehnologije ali nije baš siguran šta tačno želi pa će tokom gradnje menjati ili pojasniti svoje odluke zavisno od troškova, tempa gradnje, itd.

Proizvodnja III

- U razvoju telefona postoje problemi sa malim stepenom noviteta ili izmena i visok stepen ponavljanja identičnih ili sličnih operacija
- Ovakva proizvodnja se obično naziva **masovna** ili **prediktivna proizvodnja**
- Proizvodnja sa visokim stepenom noviteta, kreativnosti i izmena se naziva **razvoj novih proizvoda (*new product development*)** ili **inventivni projekti**

Proizvodnja IV

Prediktivna proizvodnja	Razvoj novih proizvoda
Moguće je prvo napraviti specifikaciju pa onda razviti proizvod.	Jako retko je moguće unapred napraviti detaljnu specifikaciju koja se neće menjati.
Na početku proizvodnje moguće je prilično pouzdano proceniti trošak i trud.	Na početku proizvodnje nije moguće. Kako broj empirijskih podataka raste moguće je kvalitetnije vršiti procene i planiranja.
Moguće je identifikovati, definisati, rasporediti i organizovati sve aktivnosti proizvodnje.	Na samom početku ovo nije moguće. Neophodni su adaptivni koraci vođeni <i>build-feedback</i> ciklusom.
Prilagodljivost nepredvidivim izmenama nije pravilo i stope promene su relativno male.	Očekuje se prilagodljivost nepredvidivim izmenama. Zahtevi za promenama su visoki.

Proizvodnja (razvoj) softvera I

- Većina razvoja softvera nije prediktivna ili masovna proizvodnja
- Razvoj softvera spada u razvoj novih proizvoda (*new product development*)

Proizvodnja (razvoj) softvera II

- Faktori koji utiču na kreiranje pouzdane specifikacije pre samog razvoja (implementacije) softvera
 - Korisnici nisu sigurni šta tačno žele
 - Imaju problem da iskažu ono šta znaju i šta očekuju
 - Mnogi detalji zahteva će se obelodaniti tek tokom razvoja
 - Detalji su često jako komplikovani korisnicima
 - Kako se softver razvija korisnici menjaju mišljenje
 - Spoljašnji uticaj na promenu zahteva (promene na tržištu, konkurenciji, ...)

Proizvodnja (razvoj) softvera III

- Kompleksnost, visok stepen izmena i nepredvidivost u razvoju softvera predstavljaju osnovnu motivaciju za primenu agilnih metoda u njegovom razvoju

Iterativni razvoj

- Iterativni razvoj predstavlja pristup u razvoju softvera (ili drugih stvari) u kome je globalni tok proizvodnje sastavljan od nekoliko iteracija u sekvenci

Iterativni razvoj - Iteracija I

- Svaka iteracija je samosadržan mini projekat koji se sastoji od aktivnosti poput analize zahteva, dizajna, programiranja, testiranja, itd.
- Cilj iteracije je **iteration release** – stabilan, integrisan i testiran deo čitavog sistema
- Većina iteration *release*-ova su interni, a ne produkcion
- Rezultat poslednje iteracije je finalan proizvod koji se predaje klijentima

Iterativni razvoj - Iteracija II

- Iterativni pristup podržava mogućnost za izmene, ali ne i kaos tokom iteracije – usred čestih izmena potrebno je obezbediti određen nivo stabilnosti kako bi razvoj mogao da napreduje, što se postiže pravilom:
 - *Kada se odaberu zahtevi za iteraciju i kada je iteracija u toku niko sa strane (uključujući i klijente) ne može da unosi promene u te zahteve*

Iterativni razvoj - Prioriteti I

- Dominantne su dve vrste prioriteta: **risk-driven** i **client-driven**
- U risk-driven pristupu biraju se najrizičniji, najkompleksniji elementi za prve iteracije
 - Na ovaj način visoko rizični elementi isplivaju na površinu u ranim fazama razvoja
 - **Problem:** kako definisati rizike i dodeliti ih elementima

Iterativni razvoj - Prioriteti II

- Client-driven pristup podrazumeva da klijent bira elemente koji se realizuju u sledećoj iteraciji
 - Na ovaj način klijent usmerava projekat, iteraciju po iteraciju, zahtevajući realizaciju elemenata za koje smatra da su u tom trenutku najbitniji
- Preporuka je da se primenjuju paralelno oba pristupa
 - Klijenti ne mogu uvek predvideti šta je tehnički komplikovano ili riskantno implementirati
 - Programeri ne moraju znati koje funkcije su bitne

Vremenski ograničen iterativni razvoj

- Vremensko ograničenje iteracije predstavlja fiksiranje datuma kraja iteracije bez mogućnosti njegove promene
- Čitav projekat takođe može biti vremenski ograničen
- Trajanje svake iteracije ne mora biti identično

Evolutivni i adaptivni razvoj I

- Evolutivni iterativni razvoj podrazumeva da analiza zahteva, planiranje, estimacija i specifikacija evoluiraju ili se razrađuje tokom iteracija, umesto da se sve definiše na samom početku projekta i da se ne menja
- Evolutivne metode su konzistentne sa nepredvidivošću i izmenama u razvoju novog proizvoda (*new product development*)

Evolutivni i adaptivni razvoj II

- Adaptivni razvoj podrazumeva da se elementi prilagode u odnosu na povratnu informaciju prethodnog posla (povratna informacija od korisnika, programera, testera, ...)
- Namera adaptivnog razvoja je identična kao i kod evolutivnog razvoja, samo što se ovde naglašava povratna informacija (*feedback*) u postupku evolucije

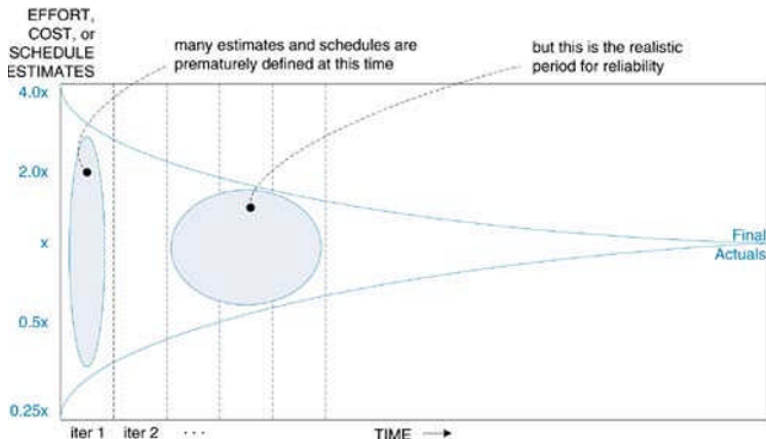
Evolutivna i adaptivna analiza zahteva

- U evolutivnom i adaptivnom razvoju nije slučaj da se zahtevi stalno i često menjaju
- Obično definisanje i prečišćavanje zahteva se dešava tokom prvih iteracija, gde se akcenat daje na tehnološki zahtevne ili poslovno važne zahteve

Evolutivno i adaptivno planiranje

- Slično kao i kod analize zahteva, procene i rasporedi nisu trajno nedefinisani
- U početnim fazama postoji određen nivo neodređenosti koji se smanjuje kako vreme prolazi i kako se informacije akumuliraju – **kupa neodređenosti**

Kupa neodređenosti



Slika preuzeta iz: *Agile and Iterative Development: A Manager's Guide*, Craig Larman, Addison-Wesley Professional, 2003.

Inkrementalna isporuka

- Inkrementalna isporuka predstavlja neprestano isporučivanje softvera u produkciju, gde se pri svakoj novoj isporuci isporučuju nove/nadograđene funkcionalnosti

Evolutivna isporuka

- Predstavlja nadograđen postupak inkrementalne isporuke gde je akcenat na dobijanju *feedback*-a u vezi prethodne instalacije i njegovom korišćenju za narednu isporuku

Agilne metodologije

- Agilne metodologije primenjuju vremenski ograničen iterativni i evolutivni razvoj, adaptivno planiranje, promovišu evolutivnu isporuku i uključuju druge principe, prakse i vrednosti koji pospešuju **agilnost** – brz i fleksibilan odgovor na izmene
- Osim ovoga promovišu praksu i principe koji podstiču agilnost kroz jednostavnost, lakoću, komunikaciju, autonomiju timova, programiranje iznad dokumentovanja, ...

Agilni manifest i principi

- 2001. godine grupa zainteresovana za agilne metode se sastala da bi definisali zajedničku osnovu – <http://www.agilealliance.org>
- Kao rezultat nastaju **agilni manifest i agilni principi**
- Agilno upravljanje projektima treba da bude vođeno agilnim principima

Agilni manifest

- *Pojedinci i interakcije pre procesa i alata*
- *Funkcionalan softver pre detaljne dokumentacije*
- *Kolaboracija sa klijentom pre pregovaranja o ugovoru*
- *Odgovor na izmene pre praćenja plana*

Agilni principi I

- 1 Zadovoljan klijent je naš vrhunski prioritet, koji ostvarujemo blagovremenom i kontinuiranom isporukom vrhunskog softvera.
- 2 Spremno prihvatamo promene zahteva, čak i u kasnoj fazi razvoja. Agilni procesi omogućavaju uspešno prilagođavanje izmenjenim zahtevima što za rezultat ima prednost naših klijenata u odnosu na konkurenciju.
- 3 Redovno isporučujemo primenljiv softver, u periodu od nekoliko nedelja do nekoliko meseci, dajući prednost kraćim intervalima.

Agilni principi II

- 4 Poslovni ljudi i developeri svakodnevno da sarađuju u toku celokupnog trajanja projekta.
- 5 Projekte ostvarujemo uz pomoć motivisanih pojedinaca. Obezbeđujemo im ambijent i podršku koja im je potrebna i prepuštamo im posao s poverenjem.
- 6 Za najproduktivniji i najefikasniji metod prenosa informacije do i unutar razvojnog tima smatramo kontakt licem u lice.

Agilni principi III

- 7 Primenljiv softver je osnovno merilo napretka.
- 8 Agilni procesi promovišu održivi razvoj. Pokrovitelji, developeri i korisnici moraju biti u stanju da kontinuirano rade usklađenim tempom, nezavisno od perioda trajanja projekta.
- 9 Stalna posvećenost vrhunskom tehničkom kvalitetu i dobar dizajn pospešuju agilnost.

Agilni principi IV

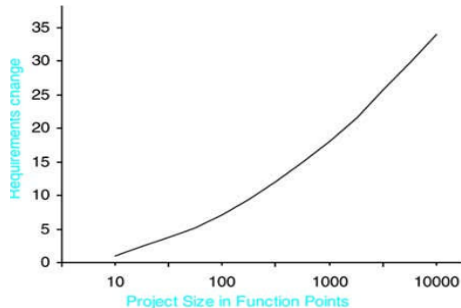
- 10 Jednostavnost – veština dovođenja do najvišeg stepena količine rada koji nije potrebno uraditi – je od suštinske važnosti.
- 11 Najbolje arhitekture, zahtevi i dizajn, rezultat su rada samo–organizovanih timova.
- 12 Timovi u redovnim intervalima razmatraju načine kako da postanu efikasniji, zatim se usklađuju i na osnovu tih zaključaka prilagođavaju dalje postupke.

Motivacija za agilni pristup I

- Bolje ikakva nego nikakva metodologija.
- *If it ain't broken, don't fix it*
 - Ako organizacija primenjuje waterfall model (ili bilo koji drugi) i postiže visok stepen uspešnosti, produktivnosti, itd. onda nema potrebe da se menja
- Ako postoje problemi agilni pristup može biti solucija za njihovo rešenje

Motivacija za agilni pristup II

- *Jones C. - Applied Software Measurment, McGraw Hill*
 - X osa – veličina projekta u funkcionalnim poenima (mera kompleksnosti sistema nezavisna od programskog jezika)
 - Y osa – procena ukupnih promena zahteva
- Srednje složeni projekti imaju oko 25% promena
- Složeniji projekti imaju preko 35%
- Visok procenat promena zahteva predstavlja jedan od ključnih motiva za primenu agilnih metodologija



Ključni motivi za agilni pristup I

- Agilni pristup ima manji rizik za neuspešnost projekta od waterfall pristupa
 - Pokazano kroz naučna istraživanja i kroz realne projekte
- Rana detekcija i smanjenje rizika
 - Agilni model promoviše prvo pristup/realizaciju najriskantnijim zahtevima
- Prihvata i podržava rane zahteve za promenom
 - Agilni pristup podržava čestu promenu zahteva kao jednu od važnih karakteristika softverskih proizvoda
- Kontrolisana složenost
 - Step en neuspeha je veći i produktivnost je niža kod veoma složenih softverskih projekata. Agilni pristup dekomponuje složene projekte u manje celine čija je složenost upravljiva.

Ključni motivi za agilni pristup II

- Poverenje i zadovoljstvo
 - Kratke iteracije vode do relativno brzo urađenog zadatka i *feedback*-a, što pozitivno utiče na tim i takođe povećava poverenje klijenta u tim jer vrlo brzo ima uvid u progres projekta.
- Rano kreiranje dela projekta
 - U prvim fazama moguće je imati funkcionalan softver sa urađenim delovima definisanih funkcionalnosti
- Praćenje projekta – bolja prediktivnost
 - Mnogo realniji uvid u napredak projekta u odnosu na waterfall modele. U svakoj fazi/iteraciji dobija se deo softvera koji je funkcionalan što je verovatno najbolji način za praćenje progressa. Nakon par početnih iteracija mnogo bolje je moguće predvideti sledeće iteracije.

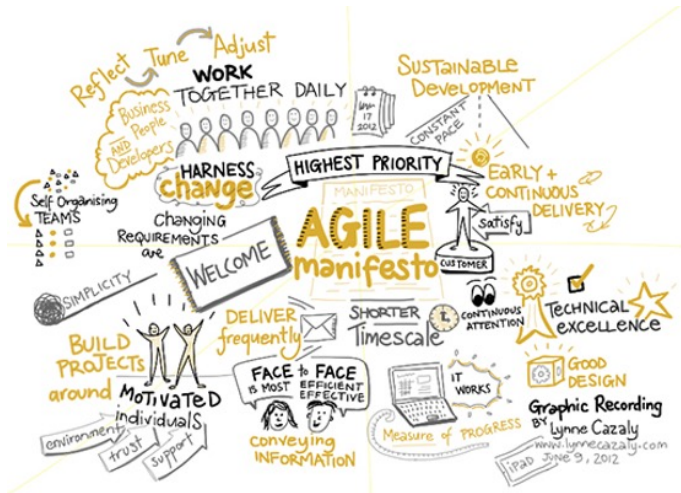
Ključni motivi za agilni pristup III

- Viši kvalitet, manje defekata
 - Agilne metode zahtevaju testiranje još u ranim fazama i često u svim mogućim dimenzijama (funkcionalnost, opterećenje, performanse, ...). Sami testovi mogu da se evaluiraju.
- Finalni proizvod više zadovoljava klijentove želje
 - Zbog rane evaluacije i *feedback*-a od strane klijenta softver je moguće lakše prolagoditi klijentovim očekivanjima.
- Rano i redovno poboljšanje procesa
 - Agilne metodologije stimulišu analizu (evaluaciju) programiranja, testiranja, zahteva, itd. nakon svake iteracije

Ključni motivi za agilni pristup IV

- Zahteva se komunikacija i angažovanje
 - Jedan od značajnih razloga za neuspeh projekata je nedostatak komunikacije između klijenta i tima kao i između samih članova tima i podtimova
- Zahteva se IKIWISI (I'll Know It When I See It)
 - Posebno izraženo kod softvera gde je UI dominantan

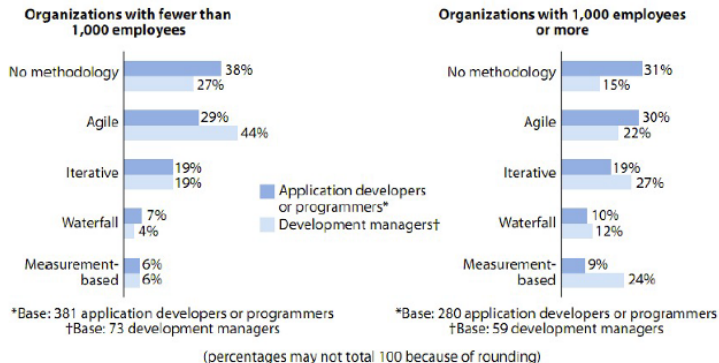
Agilne metodologije



Slika preuzeta iz: <https://www.planisware.com/hub/blog/how-can-non-it-or-non-software-teams-use-agile-methodologies>

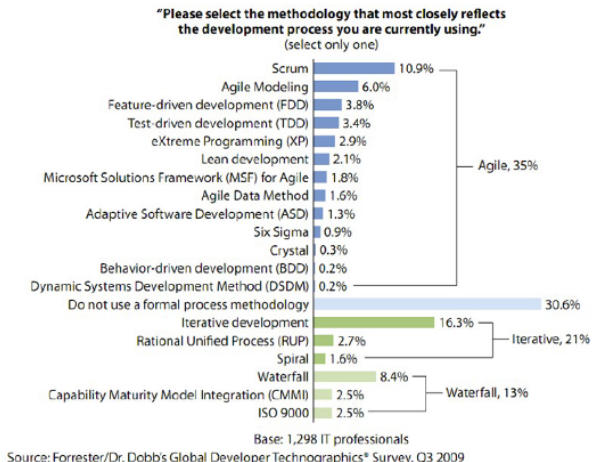
Anketa - korišćene metodologije

"Please select the methodology that most closely reflects the development process you are currently using."



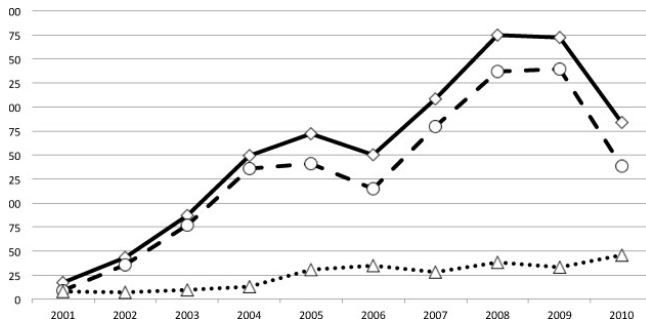
Source: Forrester/Dr. Dobb's Global Developer Technographics* Survey, Q3 2009

Anketa - korišćene agile metodologije



Istraživački radovi na temu MRS - 2001-2010

- svi radovi (na vrhu)
- radovi na konferencijama (u sredini)
- radovi u časopisu (na dnu)



Slika preuzeta iz: Torgeir et al, A decade of agile methodologies: Towards explaining agile software development, Journal of Systems and Software, Volume 85, Issue 6, 2012