



Softversko inženjerstvo

Elektronski fakultet Niš

Agilne metode za
razvoj softvera



Elektronski fakultet u Nišu



Sadržaj

- Principi agilnog razvoja
- Agilne metode
- Scrum



Elektronski fakultet u Nišu



Agilni razvoj softvera

- Tradicionalne nasuprot agilnim metodama
- Agilno = aktivnost, hitnost, spremnost za pokret
- Iterativni i inkrementalni razvoj
- Razlika
 - Inkrementalne metode 3 - 6 meseci
 - Agilne metode 1 - 4 nedelje



Elektronski fakultet u Nišu



“Agile Manifesto” Definicija agilnog programiranja

- Više vrede:
 - Pojedinci i interakcije nego procesi i alati
 - Programska podrška koja radi nego dokumentacija
 - Saradnja s klijentima nego ugovori
 - Odgovor na promene nego praćenje plana



Principi

- Zadovoljstvo korisnika brзом isporukom korisnog softvera
- Mogućnost promene zahteva, čak i u poodmakloj fazi razvoja
- Česta isporuka softvera, u razmaku od par nedelja
- Ispravan softver je osnovna mera napretka
- Razvoj koji je u stanju da održi konstantan tempo
- Bliska saradnja između projekatanta i poslovnih saradnika



Principi

- Najbolji tip komunikacije je komunikacija “licem-u-lice”
- Projekti se izvode u okruženju u kojem su motivisani pojedinci, u koje se može imati poverenja
- Kontinualno usmeravanje pažnje ka tehničkoj veštini i dobrom dizajnu
- Jednostavnost
- Samoorganizovani timovi
- Prilagođavanje promenljivim okolnostima



Praksa nije uvek idilična

- Kupac ne može uvek biti umešan u razvoj
- Osobine članova tima
- Različiti prioriteti
- Jednostavnost zahteva dodatni trud
- Stare navike kompanija



Agilne metode

- **Ekstremno programiranje (XP)**
- Industrijsko ekstremno programiranje (IXP)
- **Scrum**
- Crystal
- Adaptivni razvoj softvera (ASD)
- Feature driven development (FDD)
- Metode dinamičkog razvoja sistema (DSDM)
- **Lean Development (LD)**
- Agilno modelovanje (AM)



Extreme Programming (XP)

- Najpopularnija agilna metoda, nastala 1999. god
- Mali timovi
- Proizvodnja dugotrajnog softvera
- Sposobnost reagovanja na promene zahteva



Životni ciklus XP-a

1. Istraživanje (pisanje priča, upoznavanje tehnologija)
2. Planiranje (utvrđivanje potrebnog vremena, prioriteta, stvaranje)
3. Iteracije (pisanje, procena i dodeljivanje prioriteta novim zadacima, testiranje na kraju svake iteracije)
4. Proizvodnja (testiranje pre isporuke, evidentiranje novih zahteva)
5. Održavanje (dodavanje novih funkcionalnosti, ispravke grešaka)
6. Kraj (zahtevi ispunjeni, završetak projekta)



Aktivnosti u XP-u

- Da bi se dostigle vrednosti ekstremnog programiranja sprovode se četiri vrste aktivnosti pomoću kojih se te vrednosti implementiraju u ponašanje i rad tima:
 - Kodiranje
 - Slušanje
 - Testiranje
 - Projektovanje



Uloge u XP-u

- Menadžer - formira tim i upravlja njime
- Trener - uči članove tima o XP metodologiji
- Tragač - prikuplja korisničke zahteve i prati napredak testova
- Programer - piše testove, projektuje i kodira
- Tester – pomaže kupcu da piše i razvija testove
- Kupac - piše zahteve i testove, te bira zahteve koji će se raditi u određenoj iteraciji.



Uslovi u XP-u

- Otvorena radna sredina
- Oglasna tabla
- Normalno radno vreme
- Dnevni sastanci
- Nedeljni sastanci
- Tromesečni sastanci



Lean Development (LD)

- Metodologija inspirisana sistemom proizvodnje u japanskim fabrikama automobila
- Vreme razvoja u Tojoti je duplo kraće, a produktivnost radnika je 4 puta veća, uz znatno manji procenat grešaka nego u fabrikama u SAD-u
- LD je skoncentrisan na vrh preduzeća i strategiju upravljanja, ne na detalje razvojnog procesa na niskom nivou
- Pošto se LD bavi više filozofijom upravljanja nego konkretnim razvojnim procesom, detalji razvoja nisu precizno definisani, kao ni veličina tima i način njegovog organizovanja



Principi LD-a

- Najveći prioritet je zadovoljiti očekivanja korisnika
- Potrebno je obezbediti najbolje moguće rešenje za novac korisnika
- Uspeh zavisi od aktivnog učešća korisnika
- Svaki LD projekat je rezultat timskog rada
- Sve se može menjati
- Rešenje primenjivo na širi problemski domen, a ne samo za konkretni problem
 - Realizacija "unikatnog" rešenja je preskupa



Principi LD-a

- Bolje je prilagoditi gotovo rešenje, nego raditi svaki put iz početka
- Bolje je imati 80% rešenja danas, nego 100% rešenja u budućnosti
- Minimalizam je esencijalan za uspeh projekta
- Potreba određuje tehnologiju, ne obrnuto

Uvod u Scrum

Agilno upravljanje projektima



Poreklo scrum-a

- “Pristup razvoju proizvoda kao u štafetnim trkama može biti u suprotnosti sa željama za maksimalnom brzinom i fleksibilnošću. Umesto toga, holistički ili ‘ragbi’ pristup (gde tim pokušava da pređe put kao celinu, dodajući loptu napred-nazad) može bolje poslužiti savladavanju današnjih zahteva.”
 - Hirotaka Takeuchi and Ikujiro Nonaka, “The New New Product Development Game”, Harvard Business Review, January 1986.

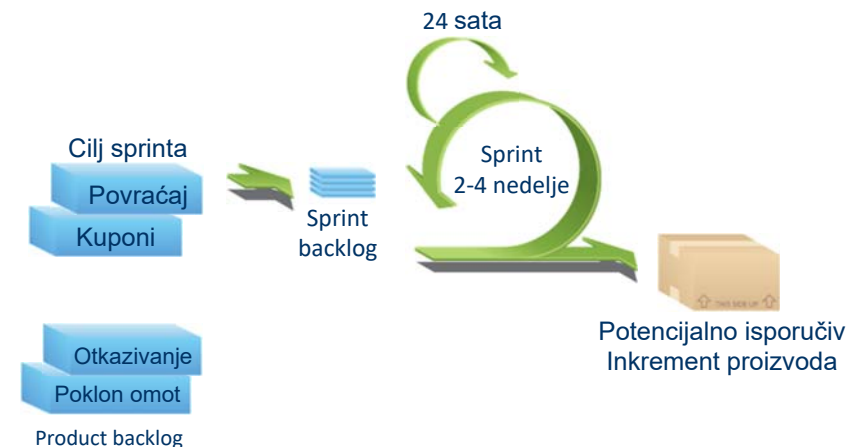


Scrum u 100 reči

- Scrum je agilni proces koji nam omogućava da se fokusiramo na isporuku najviših poslovnih vrednosti u najkraćem roku
- Omogućava nam da brzo i često unapređujemo aplikacije (svake 2 nedelje do jednog meseca)
- Posao postavlja prioritete. Timovi se samoorganizuju da bi odredili najbolji način za isporuku funkcionalnosti koje imaju najviši prioritet.
- Svake dve nedelje do mesec dana svako može da vidi potpuno funkcionalan softver i odluči se za puštanje u produkciju ili da nastavi unapređivanje u sledećem *sprintu*.



Scrum proces



Sprintovi

- Scrum projekti ostvaruju napredak kroz niz sprintova
- Tipično trajanje je 2-4 nedelje ili najviše jedan kalendarski mesec
- Konstantno trajanje omogućava bolji ritam
- Za vreme sprinta, proizvod se dizajnira, kodira i testira
- Za vreme sprinta nema promena!
 - Planirajte trajanje sprinta shodno tome koliko dugo možete izdržati bez promena

Scrum framework

Uloge

- Vlasnik proizvoda
- Scrum master
- Tim

Sastanci

- Planiranje sprinta
- Pregled sprinta
- Retrospektiva sprinta
- Dnevni scrum sastanak

Artefakti

- Product backlog
- Sprint backlog
- Burndown dijagrami

Uloge

- Vlasnik proizvoda
- Scrum master
- Tim

Vlasnik proizvoda

- Definiše funkcionalnosti proizvoda
- Odlučuje o datumu i sadržaju *release-a*
- Odgovoran za profitabilnost proizvoda (ROI)
- Prioritizuje funkcionalnosti u skladu sa značajem za tržište
- Prema potrebi, koriguje funkcionalnosti i prioritet u svakoj iteraciji
- Prihvata ili odbacuje rezultate rada



Scrum master

- Predstavlja menadžment projekta
- Odgovoran za sprovođenje Scrum vrednosti i prakse
- Uklanja smetnje
- Osigurava da je tim potpuno funkcionalan i produktivan
- Omogućava blisku saradnju između svih uloga i funkcija
- Štiti tim od spoljnih ometanja



Tim

- Obično 5-9 članova
- Multi-funkcionalni:
 - Programeri, testeri, dizajneri itd.
- Članovi bi trebalo da budu angažovani puno radno vreme
 - Može biti izuzetaka (npr. DBA)
- Timovi se samoorganizuju
 - Idealno bi bilo bez funkcija, ali teško ostvarivo
- Sastav bi trebalo menjati samo između sprintova



Sastanci

- Planiranje sprinta
- Dnevni scrum sastanak
- Pregled sprinta
- Retrospektiva sprinta



Planiranje sprinta

- Tim bira stavke iz *product backlog*-a za koje može da se obaveže da ih može završiti
- Kreira se *Sprint backlog*
 - Identifikuju se zadaci i svaki se procenjuje (1-16 sati)
 - Uz saradnju, ne samo Scrum master
- Uzima se u obzir *high-level* dizajn rešenja

Dnevni scrum sastanak

- Parametri
 - Dnevno
 - 15 minuta
 - Stand-up
- Nije za rešavanje problema
 - Svi su pozvani
 - Samo članovi tima, scrum master i vlasnik proizvoda smeju da pričaju
- Pomaže da se izbegnu ostali nepotrebnii sastanci



Dnevni sastanak

- Svi odgovaraju na 3 pitanja:
 1. Šta si radio juče?
 2. Šta ćeš raditi danas?
 3. Da li ti nešto stoji na putu?
- Ovo NIJE izveštaj o statusu scrum masteru već informisanje među jednakima

Pregled sprinta

- Tim prezentuje ono što je urađeno za vreme sprinta
- Tipično u formi demonstracije novih funkcionalnosti ili upotrebljene arhitekture
- Neformalno
 - Pravilo 2-satne pripreme
 - Bez slajdova
- Učestvuje ceo tim
- Svi su pozvani

Retrospektiva sprinta

- Periodično razmatranje šta je dobro, a šta ne
- Tipično 15–30 minuta
- Radi se posle svakog sprinta
- Ceo tim učestvuje
 - Scrum master
 - Vlasnik proizvoda
 - Tim
 - Moguće je učešće klijenata i ostalih



Artefakti

- Product backlog
- Sprint backlog
- Burndown dijagrami



Product backlog

- Zahtevi
- Lista svih zahtevanih poslova na proizvodu
- Idealno je ako su zahtevi takvi da svaki predstavlja vrednost za korisnika proizvoda
- Prioritizuje ih product owner
- Reprioritizovani na početku svakog sprinta



Primer product backlog-a

Stavka backlog-a	Procena
Omogućiti gostu da napravi rezervaciju	3
Kao gost, želim da otkažem rezervaciju	5
Kao gost, želim da promenim datume rezervacije	3
Kao zaposleni u hotelu, mogu da startujem RevPAR izveštaje (revenue-per-available-room)	8
Unaprediti obradu grešaka	8
...	30
...	50



Cilj sprinta

- Kratka izjava na šta će biti fokusiran rad za vreme sprinta
- Na primer:
 - Omogućiti da se aplikacija, uz Oracle, može startovati i na SQL Server-u.
 - Podržati funkcionalnosti neophodne za studije genetike stanovništva.
 - Podržati više tehničkih indikatora od kompanije ABC sa real-time podacima.

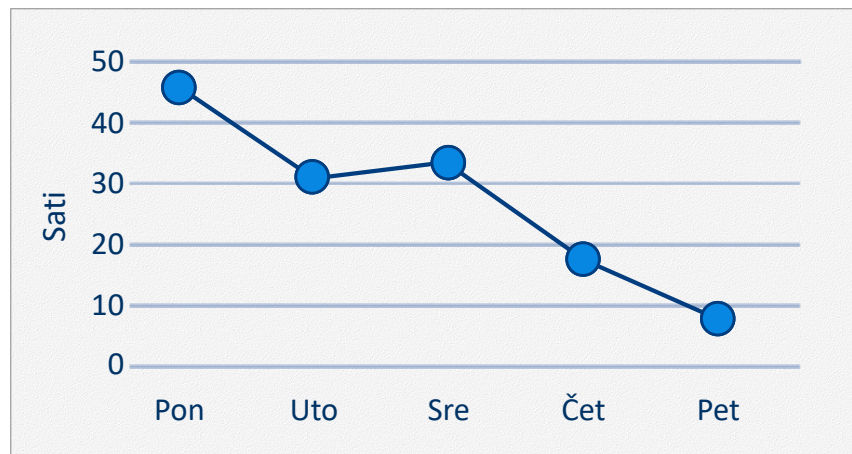
Upravljanje sprint backlog-om

- Pojedinci se sami opredeljuju za zadatke
 - Zadaci se ne dodeljuju
- Procena preostalog posla se radi svakodnevno
- Svaki član tima može dodavati, brisati ili menjati sprint backlog
- Posao se pojavljuje u toku sprinta
- Ako je posao nejasan, definiše se zadatak u sprint backlog-u sa više vremena i kasnije se usitnjava
- Preostali posao se procenjuje čim saznamo više

Primer sprint backlog-a

Zadaci	Pon	Uto	Sre	Čet	Pet
Kodiranje kor. interfejsa	8	4	8		
Kodiranje srednjeg sloja	16	12	10	7	
Testiranje srednjeg sloja	8	16	16	11	8
Pisanje online helpa	12				
	46	32	34	18	8

Sprint burndown dijagram



Skalabilnost

- Tipičan tim je 7 ± 2 ljudi
 - Skalabilnost se realizuje pomoću timova timova
- Činioci koji utiču na skalabilnost
 - Tip aplikacije
 - Veličina tima
 - Disperzija tima
 - Trajanje projekta
- Scrum je uspešno korišćen na više projekata sa više od 500 osoba

Skaliranje pomoću skrama skramova

