

Agilne metodologije i Scrum

Inženjering informacionih sistema

Sadržaj

- Tradicionalni i agilni pristup razvoju softvera
- *Agile Manifesto* – vrednosti i principi
- Pregled agilnih metodologija
- *Scrum* – uvod i osnove
- *Scrum* uloge
- *Scrum* artefakti
- *Scrum* eventi
- Sprint – životni ciklus
- *Product Backlog* i *User Stories*
- Estimacija kompleksnosti – *Story Points*
- *Definition of Done*
- *Scrum* na našem projektu

Tradicionalni i agilni pristup

- Tradicionalni (*Waterfall*) pristup:
 - Sekvencijalne faze: zahtevi → dizajn → implementacija → testiranje → isporuka
 - Svi zahtevi se definišu unapred – teška adaptacija na promene
 - Isporuka vrednosti tek na kraju projekta
 - Visok rizik: greška otkrivena kasno košta mnogo više
- Agilni pristup:
 - Iterativni i inkrementalni razvoj – vrednost se isporučuje u kratkim ciklusima
 - Zahtevi evoluiraju tokom projekta – promene su dobrodošle
 - Stalna saradnja sa korisnikom i redovne povratne informacije
 - Greške se otkrivaju i ispravljaju brzo, u okviru tekuće iteracije

Agile Manifesto – četiri vrednosti (2001)

- Individue i interakcije > procesi i alati
 - Ljudi koji saraduju donose bolji softver od krutih procedura
- Funkcionalan softver > sveobuhvatna dokumentacija
 - Dokumentacija postoji, ali radni softver je primarni cilj
- Saradnja s klijentom > pregovori oko ugovora
 - Korisnik je partner u procesu razvoja, ne samo potpisnik ugovora
- Reagovanje na promene > praćenje plana
 - Fleksibilnost je prednost, ne slabost

Agile Manifesto – 12 principa (izbor)

- Rana i kontinuirana isporuka vrednosti korisniku je prioritet
- Promene zahteva su dobrodošle čak i kasno u razvoju
- Funkcionalan softver isporučivati često – od nekoliko nedelja do nekoliko meseci
- *Product Management* deo tima i programeri moraju svakodnevno saradivati tokom projekta
- Projekti se grade oko motivisanih individua – dati im okruženje, podršku i poverenje
- Najefikasniji prenos informacija je razgovor licem u lice
- Funkcionalan softver je primarna mera napretka
- Agilni procesi promovišu održiv razvoj – stalan i ujednačen ritam rada
- Stalna pažnja na tehničku izvrsnost i dobro dizajn poboljšava agilnost
- Jednostavnost – umetnost maksimizovanja količine neobavljenog posla – je suštinska

Pregled agilnih metodologija

- *Scrum* – najpopularniji okvir; iterativni razvoj u sprintovima; jasno definisane uloge i eventi
- *Kanban* – vizualizacija toka rada; limit WIP (Work In Progress); nema fiksnih iteracija
- *Extreme Programming (XP)* – fokus na tehničkim praksama: TDD, pair programming, CI/CD
- *SAFe (Scaled Agile Framework)* – skaliranje agilnosti na nivo više timova i organizacije
- *LeSS (Large-Scale Scrum)* – proširenje *Scrum*-a za više timova koji rade na istom proizvodu
- *Kanban vs. Scrum* – ključna razlika:
 - *Scrum*: vremenski ograničeni sprintovi, fiksne uloge, planirani eventi
 - *Kanban*: kontinualni tok, bez fiksnih iteracija, fleksibilniji
- Na ovom predmetu koristimo: *Scrum* metodologiju uz *Jira* alat

Scrum – uvod

- *Scrum* je okvir za razvoj složenih proizvoda u složenim okruženjima
- Nije metodologija niti skup alata – *Scrum* definiše okvir, a tim popunjava detalje
- Temelje čine tri stuba empirizma:
 - Transparentnost – svi aspekti procesa moraju biti vidljivi svim akterima
 - Inspekcija – *Scrum* artefakti i napredak ka ciljevima se redovno pregledaju
 - Adaptacija – ako inspekcija pokaže devijaciju, proces se prilagođava što pre
- Pet vrednosti *Scrum*-a: predanost, fokus, otvorenost, poštovanje, hrabrost
- Osnova: kratke iteracije (sprintovi), samooorganizovani timovi i stalne povratne informacije

Scrum uloge

- *Product Owner* (PO) – vlasnik proizvoda:
 - Odgovoran za maksimizovanje vrednosti proizvoda
 - Jedina osoba odgovorna za upravljanje *Product Backlog*-om
 - Predstavlja glas korisnika i svih zainteresovanih strana (*stakeholders*)
- *Scrum Master* (SM) – čuvar procesa:
 - *Servant-leader* tima – služi timu, a ne upravlja njime
 - Osigurava razumevanje i primenu *Scrum* vrednosti i prakse
 - Uklanja prepreke (*impediments*) koji usporavaju *Development Team*
- *Development Team* – razvojni tim:
 - Samoorganizovani i međufunkcionalni – poseduju sve veštine za isporuku Inkrementa
 - Preporučena veličina: 3–9 članova (optimalno 5–7)
 - Kolektivno odgovorni za rezultat sprinta

Scrum artefakti

- *Product Backlog* – lista svih zahteva za proizvod:
 - Jedini izvor zahteva za sve promene na proizvodu
 - Živi dokument – nikada nije potpun; evoluira tokom projekta
 - *Product Owner* je vlasnik; prioritizovan je po vrednosti, riziku, kompleksnosti
- *Sprint Backlog* – plan tekućeg sprinta:
 - Skup stavki *Product Backlog*-a odabranih za sprint + plan isporuke Inkrementa
 - Vidljiv plan u realnom vremenu; vlasništvo *Development Team*-a
 - Može se ažurirati tokom sprinta – samo tim može dodavati/uklanjati stavke
- Inkrement – rezultat sprinta:
 - Suma svih završenih *Product Backlog* stavki u tekućem i svim prethodnim sprintovima
 - Mora biti u upotrebljivom stanju i ispunjavati *Definition of Done*

Scrum događaji

- *Sprint* – vremenski okvir (time-box) od 1–4 nedelje; srce Scrum procesa
- *Sprint Planning* – planiranje sprinta:
 - Šta može biti isporučeno? (odabir iz *Product Backlog*-a)
 - Kako će odabrani posao biti urađen? (razrada plana)
 - *Time-box*: max 8 sati za mesečni sprint
- *Daily Scrum* – dnevni sastanak (15 min, svaki dan):
 - Šta sam uradio juče? Šta planiram danas? Ima li prepreka?
 - Cilj: inspekcija napretka i prilagođavanje *Sprint Backlog*-a
- *Sprint Review* – pregled rezultata (max 4 sata za mesečni sprint):
 - Tim demonstrira isporučeni inkrement *Product Owner*-u i *stakeholder*-ima
 - Ažuriranje *Product Backlog*-a na osnovu povratnih informacija
- *Sprint Retrospective* – poboljšanje procesa (max 3 sata za mesečni sprint):
 - Šta je išlo dobro? Šta se može poboljšati? Koje poboljšanje uvodimo odmah?

Sprint – životni ciklus

- *Sprint* traje fiksno vreme (*time-box*) – po pravilu 2 nedelje na našem projektu
- Tokom sprinta:
 - Ne menjaju se ciljevi sprinta, ne dodaju se novi zahtevi bez dogovora
 - Tim svakodnevno inspekcijom prati napredak (*Daily Scrum*)
 - Opseg se može precizirati i ponovog dogovarati između PO i *Development Team*-a
 - *Sprint* se može prekinuti jedino ako *Sprint Goal* postane zastareo – odlučuje PO
- *Burndown chart* – vizualizacija preostalog posla tokom sprinta
- *Velocity* – brzina tima; prosečan broj story points isporučenih po sprintu
- Na projektnom kursu: *Sprint* 1 (osnovne funkcionalnosti) → *Sprint* 2 (glavna funkcionalnost) → *Sprint* 3 (izveštaji)

Product Backlog i User Stories

- *User Story* – neformalni opis funkcionalnosti iz ugla korisnika:
 - Format: „Kao [korisnik], želim da [akcija], kako bih [cilj/vrednost]“
 - Primer: „Kao administrator, želim da upravljam korisničkim nalogima, kako bih osigurao bezbednost sistema“
- Elementi dobro napisane *User Story* – INVEST kriterijum:
 - *Independent* – nezavisna od ostalih priča
 - *Negotiable* – opseg je predmet dogovora, ne rigidan ugovor
 - *Valuable* – donosi vrednost korisniku ili biznisu
 - *Estimable* – tim može proceniti kompleksnost
 - *Small* – dovoljno mala da stane u jedan sprint
 - *Testable* – postoje jasni kriterijumi prihvatanja (*Acceptance Criteria*)
- *Epic* – veća celina koja se „razlaže“ na više *User Stories* (*Epic Breakdown*)
- Inicijativa → Epic → User Story → Subtask – hijerarhija posla na projektu

Estimacija kompleksnosti – *Story Points*

- Story Points su relativna, ne apsolutna mera – porede se priče međusobno
- Uključuju: složenost, rizik, neizvesnost i obim posla (nije isto što i sati rada!)
- Najčešće konvencije:
 - Fibonačijev niz: 1, 2, 3, 5, 8, 13, 21 – veći skokovi reflektuju rastuću nesigurnost
 - *T-shirt* veličine: XS, S, M, L, XL – intuitivan za početnike
- Tehnika *Planning Poker*:
 - Svaki član tima nezavisno procenjuje priču i otkriva kartu istovremeno
 - Razlike u procenama pokreću diskusiju i dolazak do zajedničkog razumevanja
 - Konvergencija ka konsenzusu – ponavlja se dok se tim ne usaglasi
- *Velocity* tima – ukupan broj *Story Points* završenih u jednom sprintu
 - Koristi se za planiranje budućih sprintova i procenu rokova isporuke
- *Backlog Grooming (Refinement)* – redovna sesija pripreme i estimacije *backlog* stavki

Definition of Done (DoD)

- *Definition of Done* je zajednički dogovor tima o tome kada je zadatak „završen“
- Osigurava zajednički standard kvaliteta za sve isporučene stavke
- Primeri kriterijuma DoD-a:
 - Kod je napisan i pregledan (code review)
 - Jedinični testovi su napisani i prolaze
 - Funkcionalnost je integrisana u glavnu granu (*main branch*)
 - Nema regresija u postojećim funkcionalnostima
- Korisničke priče (*User Stories*) imaju i kriterijum prihvatanja (*Acceptance Criteria*):
 - Uslovi koje rešenje mora ispuniti da bi se smatralo prihvaćenim od strane korisnika
 - Definišu se pre početka implementacije – „*Definition of Ready*“
- Veza sa projektom: DoD definiše tim zajedno na Sprint 0 / Sprint Planning sesijama

Scrum na našem projektu

- Struktura projekta prati *Scrum* okvir prilagođen akademskom kontekstu:
 - Timovi od 4 studenta – *Development Team* + rotacija PO/SM uloga
 - Svaki tim radi na sopstvenom podsistemu organizacionog sistema
 - *Jira* – PM alat za praćenje *Product Backlog*-a, sprintova i zadataka
- *Sprint 0* – priprema (6. termin):
 - *Epic Breakdown* – razlaganje zahteva na epike i *User Stories*
 - Priprema *Jira* dashboarda i postavljanje projekta
- *Sprint 1* (7–9. termin): osnovne funkcionalnosti – upravljanje korisnicima, CRUD
- *Sprint 2* (9–12. termin): glavna funkcionalnost podsistema
- *Sprint 3* (12–14. termin): izveštajna funkcija
- Demo sesije na kraju svakog sprinta – *Sprint Review* u akademskom okruženju
- Bodovanje: Sprint 1 (8p) + Sprint 2 (17p) + Sprint 3 (10p) = 35 poena

Kraj!

Hvala na pažnji!