

Osnove inovacija – Skram (drugi deo)

Inženjerska etika i inovacije – Auditorne vežbe 3

Nikola Petrović, asistent

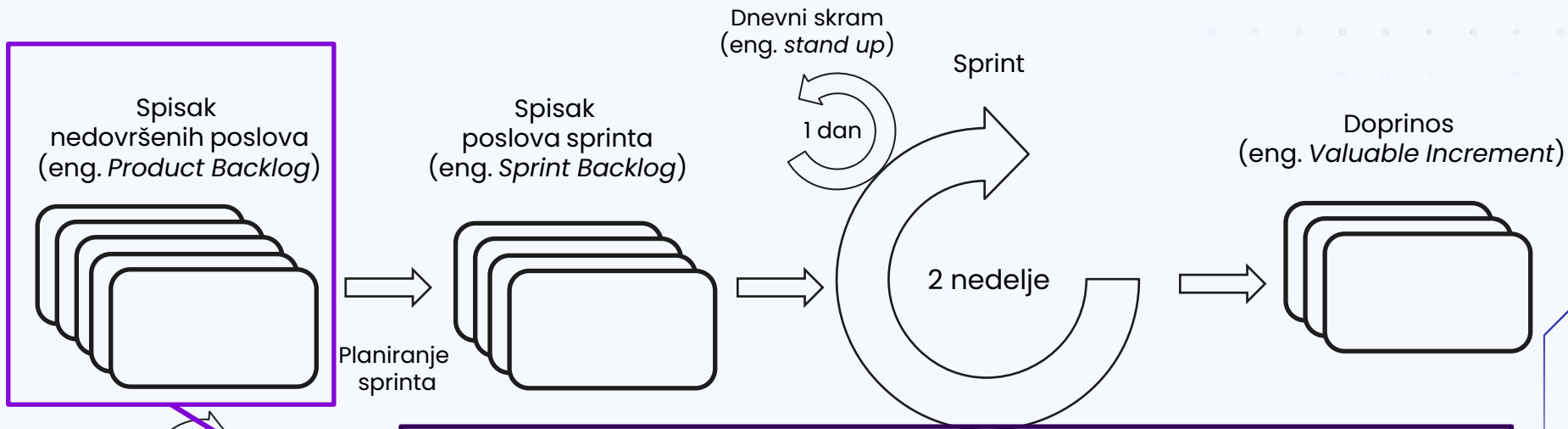
Email: nvpetrovic@mas.bg.ac.rs

Konsultacije: Sreda, 13.00–14.00, kab. 406

„Svaka organizacija koja se bavi dizajnom sistema neminovno preslikava sopstvenu komunikacionu strukturu u strukturu sistema koji razvija.”

—Melvin Konvej

Spisak nedovršenih poslova



Uređivanje spiska
nedovršenih poslova

- ❑ Sadrži sve zadatke na projektu koji još nisu dovršeni.
- ❑ Zadaci (zahtevi) su poređani prema svojoj važnosti (raspored vrši Vlasnik proizvoda – *Product owner*)
 - ❑ Nivo detalja zahteva raste zajedno sa njegovim prioritetom

Obrada novih zahteva

Novi zahtevi

Zahtev 1

Zahtev 2

Zahtev 3

Zahtev 4

Zahtev 5

Magacin

Spisak nedovršenih poslova

Uređivanje spiska nedovršenih poslova (eng. *Backlog Refinement*)

- Zahtevi se sortiraju prema važnosti (prioritetu).
 - Zahtev mora da bude dovoljno dobro definisan – razumljiv i da sadrži dovoljno informacija, kako bi ga razvojni tim potencijalno uzeo u obzir za naredni sprint.
 - Dodeljivanje težinskih poena svakom zahtevu,
 - Zahtevi koji nose preveliki broj poena se moraju raščlaniti na jednostavnije.
- Po uređivanju, odeljak za Nove zahteve bi trebalo da bude prazan. Svi prihvaćeni zahtevi bi trebalo da budu “spremni za sprint”.

Projektovanje čelične noseće konstrukcije akvarijuma

Kao glavnom majstoru,
Potrebna mi je kompletna tehnička dokumentacija noseće konstrukcije akvarijuma,
Kako bih započeo proizvodnju konstrukcije foto-bioreaktora.

Detalji i Pretpostavke

- Poznata su sva relevantna opterećenja i željeni materijali.
- Projektovanje je u skladu sa važećim tehničkim i bezbednosnim standardima.

Kriterijum prihvatanja

Dato (imajući u vidu) da su poznati svi projektni zahtevi i opterećenja,
Kada se završi projektovanje konstrukcije,
Onda mora postojati kompletna tehnička dokumentacija koja je potrebna za proizvodnju.

Angažovani

- bez angaž.

Oznake

- bez oznaka

Prekretnica (cilj)

- bez cilja

Procena težine

- bez procene

Oznake zahteva u GitHub-u

bug

- nešto ne radi
kako treba

help wanted

- potrebno dodatno
angažovanje

documentation

- potrebna poboljšanja ili
dopune dokumentacije

question

- potrebna dodatna
pojašnjenja

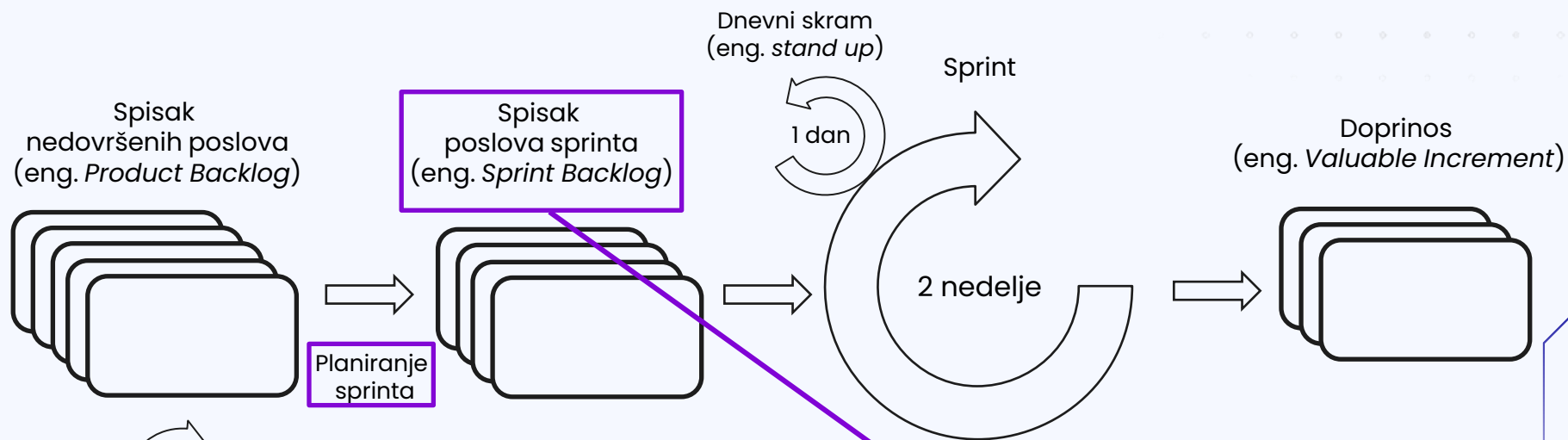
technical debt

- svesno prihvaćeno
privremeno rešenje radi
brzog dostavljanja
funkcionalnosti

enhancement

- funkcionalno
unapređenje ili zahtev

Spisak nedovršenih poslova



- ☐ Sastavlja se na sastanku planiranja sprinta.
- ☐ Sadrži sve zahteve koje treba ispuniti u narednom sprintu.
- ☐ Ukupan broj težinskih poena ne bi trebalo da pređe ukupne kapacitete tima (eng. *team velocity*).

Planiranje sprinta

Novi zahtevi

Magacin

Zahtev 1

Zahtev 5

Spisak nedovršenih
poslova

Zahtev 3

Zahtev 2

Zahtev 4

Spisak poslova
sprinta

Sprint

Spisak poslova
sprinta

Zahtev 3

Zahtev 2

Zahtev 4

Radi se

Provera
kvaliteta/QA

Završeno

Sprint

Spisak poslova
sprinta

Radi se

Provera
kvaliteta/QA

Završeno

Zahtev 4

Zahtev 3

Zahtev 2

Razmatranje i retrospektiva sprinta

- Pri kraju sprinta obavezno se održava sastanak svih članova tima i zainteresovanih strana gde se vrši **RAZMATRANJE** sprinta (eng. *Sprint Review*):
 - ☐ Prezentacija isporučene funkcionalnosti i ostvarenih ciljeva,
 - ☐ Provera usklađenosti rezultata sa korisničkim zahtevima,
 - ☐ Prikupljanje povratnih informacija za dalju doradu proizvoda.
- Potom, kao zaključna faza održava se **RETROSPEKTIVA** sprinta (eng. *Sprint Retrospective*) gde učestvuju samo članovi tima:
 - ☐ Analiza procesa rada i timske saradnje,
 - ☐ Identifikacija problema, prepreka i tehničkog duga,
 - ☐ Definisanje konkretnih unapređenja za naredni sprint.

Kontinuirani napredak

Radi lakše evaluacije uspešnosti, najbolji timovi koriste konkretne metrike (indikatore performansi) koje im pomažu da ocene svoj učinak.

1) Mean Lead Time

Vreme koje protekne od nastanka ideje do njenog plasiranja u produkciju.

2) Release Frequency

Učestalost isporuke promena, odnosno koliko često se nove funkcionalnosti ili izmene puštaju u rad.

3) Change Failure Rate

Procenat promena koje rezultuju greškama, otkazima ili zahtevaju hitne ispravke.

4) Mean Time To Recovery (MTTR)

Prosečno vreme potrebno da se sistem oporavi nakon otkaza ili zastoja.

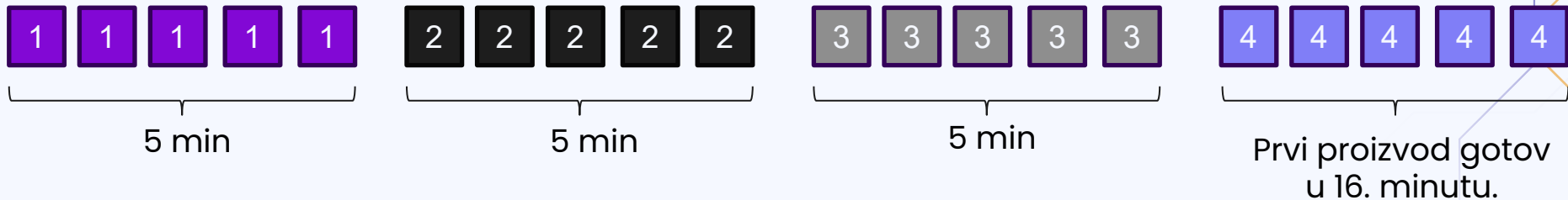


Glavni agilni principi

1. Rad u malim serijama

- ❑ Primer: Potrebno je poslati 1000 promo brošura potencijalnim kupcima.
 - 1) Savijanje brošura.
 - 2) Ubacivanje brošure u koverat.
 - 3) Zatvaranje koverti.
 - 4) Lepljenje markice.

Seriya od 50 brošura → za svaku korak treba 6 sekundi (npr. savijemo $10 \frac{\text{brošura}}{\text{min}}$)



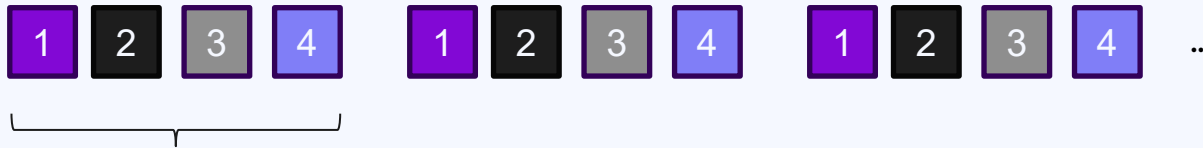
Šta ako koverti nisu imale lepak? Šta ako postoji štamparska greška na koverti?

Glavni agilni principi

1. Rad u malim serijama

- ❑ Primer: Potrebno je poslati 1000 promo brošura potencijalnim kupcima.
 - 1) Savijanje brošura.
 - 2) Ubacivanje brošure u koverat.
 - 3) Zatvaranje koverti.
 - 4) Lepljenje markice.

Serijski od 50 brošura → Formiranje gotovog proizvoda po komadu (eng. *Single Piece Flow*)

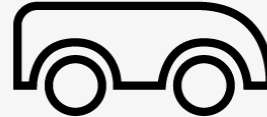


Prvi proizvod gotov posle 24 sekunde.

Glavni agilni principi

2. Minimalno održiv proizvod (eng. *Minimum Viable Product – MVP*)

- Cilj nije što pre dostaviti proizvod, već dobiti povratne informacije od tržišta i shvatiti šta se tačno traži.
- Promeni pravac ili istraj (eng. *pivot or persevere*)



Glavni agilni principi

3. Razvoj vođen ponašanjem (eng. *Behavior Driven Development – BDD*)

- Pristup razvoju fokusiran na ponašanje sistema iz ugla korisnika.
- Cilj: razviti tačno ono što je potrebno, bez pogrešnih pretpostavki.

Osnovna ideja:

- Zahtevi se opisuju kroz konkretne scenarije ponašanja,
- Svi akteri (biznis, razvoj, testiranje) koriste zajednički način izražavanja.

Gherkin sintaksa

- ☐ **Kao** <uloga u timu>,
- ☐ **Potrebno mi je** <zahtev>,
- ☐ **Kako bi** <benefit, nova funkcionalnost>.

- ☐ **Dato** – početni uslovi,
- ☐ **Kada** – akcija (korisnika),
- ☐ **Onda** – očekivani rezultat

Prednosti BDD-a

- ☐ Jasniji zahtevi i manje nesporazuma,
- ☐ Bolja saradnja tima,
- ☐ Lakša verifikacija ispravnosti sistema.

Glavni agilni principi

4. Razvoj vođen testovima (eng. *Test Driven Development – TDD*)

- Produkcioni kod se piše sa ciljem da zadovolji testove jedinica koda.
 - ❑ Test : Za ulaz x, funkcija mora da vrati y.
 - ❑ Produkcioni kod: Način na koji od izračunavamo y iz x.
- Obrnut redosled u odnosu na tradicionalni pristup
 - ❑ Prvo se pišu i uređuju testovi, a potom produkioni kod koji mora da ih prođe.
- *Refaktorisanje*
 - ❑ Sredstvo za borbu protiv opadanja kvaliteta koda
 - ❑ Predstavlja preuzimanje niza malih transformacija koda, kojima se unapređuje njegova struktura, bez vidljivih promena ponašanja sistema.

Praksa i problemi

- ❑ **Površna ili pogrešna primena Skrama**
 - ~47% timova navodi da Skram koriste delimično ili uz značajne modifikacije *
- ❑ **Organizaciona kultura ne podržava agilni način rada**
 - ~43% ispitanika navodi kulturu organizacije kao glavni problem *
- ❑ **Nedostatak jasno definisanih zahteva i prioriteta**
 - ~31% neuspešnih projekata ne uspe zbog loše definisanih zahteva **
- ❑ **Tehnički dug i pritisak kratkih rokova**
 - ~25–30% kapaciteta sprinta se troši na ispravke i refaktorisanje ***

* - Digital.ai. (2022). *State of Agile Report 2022*. Digital.ai. <https://digital.ai/resource-center/analyst-reports/state-of-agile-report>

** - Standish Group. (2020). *CHAOS Report 2020*. Standish Group International.

*** - McKinsey & Company. (2020). *Tech debt: Reclaiming tech equity*. McKinsey Global Institute.
<https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/tech-debt-reclaiming-tech-equity>