

# Agilne metodologije

# Agilne metodologije

- u Šta je agilno razvijanje softvera?
- u Zvanični agilni manifest: <http://agilemanifesto.org/iso/sr/principles.html>

# Scrum

- υ Role
- υ Activities
- υ Artifacts
- υ Rules

# Scrum - Role

- υ Product owner
- υ Scrum Master
- υ Team

# Scrum - Activities

- u Sprint planning
- u Daily scrum meeting
- u Sprint review
- u Sprint retrospective

# Scrum - Artifacts

- u Product backlog
- u Sprint backlog
- u Increments

# Dodatni materijal

u Kako Spotify radi agilno:

<https://engineering.atspotify.com/2014/03/27/spotify-engineering-culture-part-1/>

# Korisničke priče



# Korisničke priče

- ▶ As a <user role> (or system), I want to <goal> so that <benefit>
- ▶ Iskazuju suštinu bez suvišnih detalja (detalje ustanove članovi tima)
- ▶ Kriterijumi prihvatljivosti (acceptance criteria)
- ▶ Identifikator
- ▶ How to Demo checklist

# INVEST

- ▶ Independent
- ▶ Negotiable
- ▶ Valuable
- ▶ Estimable
- ▶ Small
- ▶ Testable

# Primeri

- ▶ As a user, I want to post a resume.
- ▶ As a user, I want to add a new resume to the site.
- ▶ As a user, I want to edit a resume that is already on the site.
- ▶ As a user, I want to remove my resume from the site.
- ▶ As a user, I want to mark a resume as inactive so that employers cannot send me job offers for some time.
- ▶ As a user, I want to mark a resume as hidden from certain employers.

# Kriterijumi prihvatljivost

- ▶ Verifikacija da su korisničke priče implementirane tako da rade onako kako klijent to očekuje
- ▶ Primer: As a user, I want to pay for the items in my shopping cart with a credit card.
- ▶ Test with Visa, MasterCard and American Express (pass)
- ▶ Test with a Diner's Club (fail)
- ▶ Test with a Visa Debit card (pass)...

# Labele

User Story - epic

New Feature

Issue

Bug

Task

Improvement

# Prioritet

- ▶ Sortiranje kartica
- ▶ Dodavanje polja Importance
- ▶ Dodeljuje Product Owner

# Story poeni

- ▶ Koristi se modifikovan Fibonačijev niz (  $\frac{1}{2}$ , 1, 2, 3, 5, 8, 13, 20, 40, 100)
- ▶ Planning poker
- ▶ Brzina tima

# Task

- ▶ Nakon preuzimanja US, potrebno je dekomponovati ga na zadatke (taskove)
- ▶ Definišu se nakon US
- ▶ Opisuje kako nešto napraviti, a ne šta napraviti



# GitFlow

- ▶ Model grananja za Git koji organizuje tok posla oko razvoja novih funkcionalnosti, pripreme postavke softvera i hitnih ispravki.
- ▶ Pomaže u upravljanju velikim projektima sa više članova tima, osiguravajući strukturirani pristup kontroli verzija.
- ▶ Glavne grane:
  - ▶ Master – Predstavlja stabilni kod koji je spreman za produkciju
  - ▶ Develop – Grana u kojoj se integrišu nove funkcionalnosti. Sadrži najnoviju verziju koda (koji možda još nije spreman za produkciju)
- ▶ Sporedne grane:
  - ▶ Feature – Grana u kojoj se implementiraju nove funkcionalnosti (feature/naziv-funkcionalnosti)
  - ▶ Release – Kreira se iz develop grane nakon što je ona spremna za isporuku (release/x.x).
  - ▶ Hotfix – Služi za brzu ispravku grešaka koje su pronađene u produkciji (hotfix/opis-greške)