



PROGRAMSKO INŽENJERSTVO 5. LABORATORIJSKE VJEŽBE

Agilno planiranje i Product
Backlog

Plan-driven Development

- Očekuje od nas da predvidimo kompletan plan koji prethodi razvoju
- Dizajn softvera također se radi unaprijed, a očekuje se da će implementacija biti u skladu s ovim dizajnom
- Uspjeh se mjeri prema tome koliko dobro razvoj slijedi definirani plan

Takav industrijski proces je predvidljiviji te bolje reagira na izmjene ljudi.

Agile Development

- Agilni planovi su osnova koju koristimo kako bismo kontrolirali promjene
- Agilni timovi planiraju jednako pažljivo kao i tradicionalni timovi, ali planovi se neprestano preispituju kako bi odražavali stvari koje se nauče tijekom projekta
- Uspjeh se temelji na vrijednosti koja se isporučuje

Agile razvoj softvera vidi prvenstveno kao ljudsku aktivnost, gdje su uključeni ljudi i način na koji se povezuju kao tim primarni pokretač uspjeha.

Tradicionalni vs Agilni pristup

Plan-driven (Waterfall)

- ▶ Sve se planira unaprijed
- ▶ Dizajn prije implementacije
- ▶ Uspjeh = slijeđenje plana
- ▶ Teško je napraviti izmjene
- ▶ Dokumentacija je ključna

Primjer: Izgradnja mosta - plan mora biti savršen od početka!



Agile Development

- ▶ Iterativno planiranje
- ▶ Kontinuirano učenje
- ▶ Uspjeh = isporučena vrijednost
- ▶ Promjene su dobrodošle
- ▶ Tim je najvažniji

Primjer: Razvoj mobilne aplikacije - stalno se uče potrebe korisnika!



Agile Manifesto

Ljudi i interakcije

...ispred procesa i alata



Funkcionalan softver

...ispred opsežne dokumentacije



Suradnja s klijentom

...ispred pregovaranja o ugovoru



Reagiranje na promjene

...ispred slijeđenja plana

★ Ključni Agile principi za planiranje

- ▶ **Zadovoljstvo korisnika** kroz pravovremenu isporuku vrijednog softvera
- ▶ **Promjene su dobrodošle** čak i u kasnoj fazi razvoja
- ▶ **Isporuka funkcionalnog softvera** svakih nekoliko tjedana
- ▶ **Dnevna suradnja** između developera i poslovnih ljudi
- ▶ **Razgovor uživo** je najučinkovitiji način komunikacije
- ▶ **Funkcionalan softver** je osnovno mjerilo napretka
- ▶ **Održiv ritam** - tim treba raditi dugoročno
- ▶ **Tehnička izvrsnost** i dobar dizajn pospješuju agilnost
- ▶ **Jednostavnost** - minimizirati posao koji ne treba raditi



Što je Product Backlog?



Product Backlog

Prioritizirana lista svega što treba napraviti

★ Najvažnije stavke na vrhu

Tim zna što prvo implementirati

👤 Product Owner je odgovoran

Upravlja i prioritizira backlog

👥 Pull, ne Push sistem

Tim povlači zadatke kada ima kapacitet, PO ne gura posao

🔄 Živi dokument

Konstantno se ažurira i prilagođava



Two R's: Roadmap & Requirements

Temelji za kreiranje Product Backloga



Roadmap

Plan isporuka
Dugoročna vizija
Strateški ciljevi



Requirements

Funkcijski zahtjevi
Specifične potrebe
Korisničke priče



Product Backlog

Prioritizirana lista taskova za implementaciju



Product Backlog

Prioritizirana lista svega što treba napraviti



Primjer: Mobilna Gaming App "QuizMaster"



Student startup koji razvija kviz aplikaciju za učenje

🔥 Registracija korisnika 5 SP

Kao korisnik želim se registrirati pomoću emaila ili Google računa

🔥 Kreiranje kviza 8 SP

Kao profesor želim kreirati kviz s pitanjima različitih tipova

⚡ Prijenos kvizova 3 SP

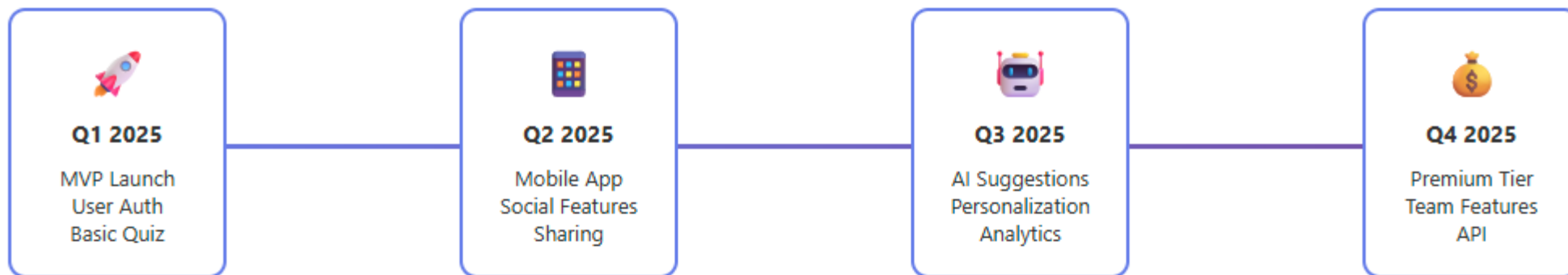
Kao korisnik želim poslati kviz prijatelju

💡 Ljestvica najboljih 5 SP

Kao korisnik želim vidjeti ranking najboljih igrača

Product Roadmap - Praktični primjer

QuizMaster App - 2025 Roadmap



Roadmap → Backlog

Svaki dio roadmapa se pretvara u Epics i User Stories u Product Backlogu

Epic vs User Story



Epic

- ▶ Veliki komad posla
- ▶ Rastavlja se na User Stories
- ▶ Obuhvaća više sprintova
- ▶ Može uključiti više timova
- ▶ Evolucija kroz razvoj



User Story

- ▶ Manja jedinica posla
- ▶ Implementira se u jednom sprintu
- ▶ Iz perspektive korisnika
- ▶ Netehnički jezik
- ▶ Pruža kontekst timu

Primjer:

Epic: "User Authentication System"

User Stories:

- Registracija s emailom
- Google OAuth login
- Reset password funkcionalnost
- Email verifikacija



Epic Breakdown

Epic: User Authentication

Kompletna implementacija autentifikacije korisnika

21 SP

Visok prioritet

Epic: Quiz Management

Kreiranje i upravljanje kvizovima

34 SP

Visok prioritet

Epic: Social Features

Društvene funkcionalnosti i dijeljenje

13 SP

Srednji prioritet

Epic: User Authentication


Kompletna implementacija autentifikacije korisnika

21 SP

Visok prioritet

 Kao korisnik, želim se registrirati s emailom 5 SP

 Kao korisnik, želim se prijaviti Google računom 3 SP

 Kao korisnik, želim resetirati lozinku 5 SP

 Kao korisnik, želim primiti email za verifikaciju 8 SP

Epic: Quiz Management

Kreiranje i upravljanje kvizovima

34 SP

Visok prioritet

 Kao profesor, želim kreirati novi kviz 8 SP

 Kao profesor, želim dodati različite tipove pitanja 13 SP

 Kao profesor, želim urediti postojeći kviz 5 SP

 Kao profesor, želim obrisati kviz 3 SP

 Kao profesor, želim vidjeti statistiku kviza 5 SP

Epic: Social Features

Društvene funkcionalnosti i dijeljenje

13 SP

Srednji prioritet

 Kao korisnik, želim podijeliti kviz prijatelju 3 SP

 Kao korisnik, želim vidjeti ljestvicu najboljih 5 SP

 Kao korisnik, želim dodati komentare na kviz 5 SP



Kako napisati User Story?

Formula:

Kao [tip korisnika] ,
želim [funktionalnost] ,
kako bih [razlog/vrijednost]

✓ ODLIČAN primjer:

Kao **student**, želim **filtrirati kvizove po predmetu**, kako bih **brže pronašao relevantne materijale za učenje**

✓ Jasno tko, što i zašto

✗ LOŠ primjer:

"Dodati filter"

✗ Nema konteksta, ne jasno tko koristi i zašto

INVEST kriteriji

Independent • Negotiable • Valuable • Estimable • Small • Testable



Prioritizacija Backloga

Product Owner može birati cijeli Epic ili važnije taskove iz više Epica

Što utječe na prioritete?

- ▶ **Prioriteti klijenta** - što je najvažnije korisnicima?
- ▶ **Kompleksnost implementacije** - koliko je teško?
- ▶ **Hitnost feedbacka** - što trebamo testirati prvo?
- ▶ **Ovisnosti između taskova** - što treba prvo?



Primjer prioritizacije za QuizMaster:



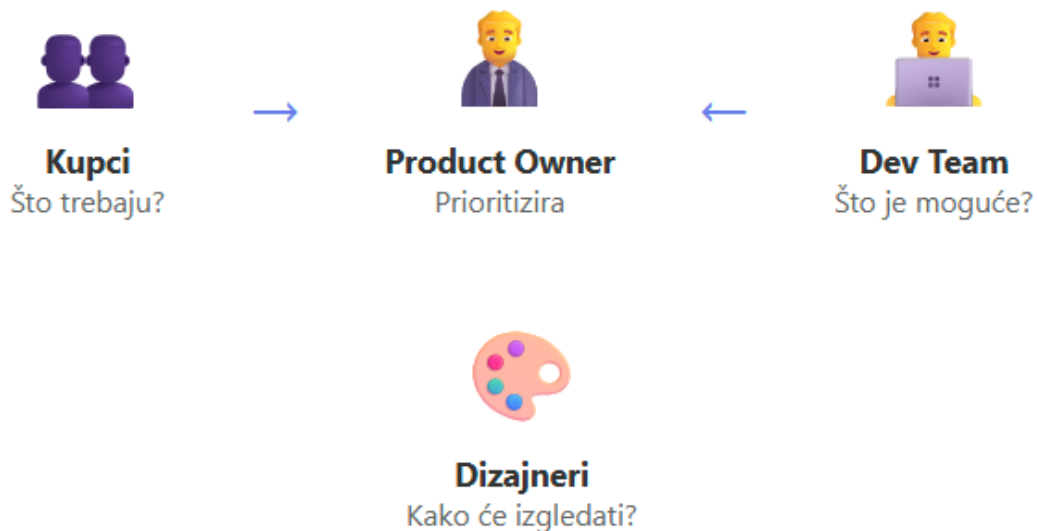
1. Visok **User Authentication** - Bez toga nitko ne može koristiti app!

2. Visok **Kreiranje kviza** - Osnovna funkcionalnost proizvoda

3. Srednji **Dijeljenje kvizova** - Važno ali može i kasnije



Product Owner ne radi u vakuumu



Ključna poruka:

Učinkoviti Product Owneri **traže input i feedback** od svih dionika kako bi optimizirali prioritete i isporuku proizvoda.







Održavanje Backloga

Product Backlog je živi dokument koji se mora redovito održavati

Kratkoročne stavke

Bliska budućnost (1-2 sprinta)

- ☒  Potpuno razrađene User Stories
- ☒  Suradnja s dizajnom završena
- ☒  Procjene od razvojnog tima
- ☒  Acceptance kriteriji definirani

Dugoročne stavke

Dalja budućnost (3+ sprinta)

- ☐  Mogu biti nejasne
- ☐  Okvirne procjene dovoljne
- ☐  Mogu se mijenjati
- ☐  "Rough" estimacije

Redovito održavanje

Product Owner treba pregledati backlog **prije svakog planiranja** kako bi osigurao točne prioritete i uključio novi feedback.



Backlog Refinement (Grooming)

Redovito preispitivanje i čišćenje neriješenih taskova

✖ Ukloni

Stories koji više nisu relevantni

Npr. feature koji je odbačen

+ Dodaj

Novi stories na temelju feedbacka

Novootkrivene potrebe korisnika

🎯 Procijeni

Dodaj ili ažuriraj Story Points

Procjena složenosti taskova

📅 Promijeni prioritet

Ažuriraj važnost taskova

Promijenjene potrebe biznisa

🔪 Razbij

Podijeli velike stories

Story s 13+ SP treba razbiti!

📄 Pojasni

Dodaj više detalja

Acceptance criteria, mockupi



Best practice: Refinement session jednom tjedno, 1-2 sata, s cijeli timom



Backlog Refinement - Proces

Prije Refinementsa: Backlog je neorganiziran, story-ji nejasni



Refinement Session

Tim zajedno pregleda backlog



Nakon Refinementsa: Backlog je čist, prioriteti jasni, stories spremni!

QuizMaster primjer:

Prije: "Dodati 社交 features" (13 SP, nejasno)

Nakon refinementsa:

- "Share quiz on social media" (3 SP)
- "Add friends feature" (5 SP)
- "Create leaderboards" (5 SP)



Vrste taskova u Product Backlogu

Feature Nove funkcionalnosti

Epics i User Stories - Nova mogućnost za korisnike

Npr: "Quiz sharing funkcionalnost"

Improvement Poboljšanja postojećih funkcionalnosti

Izmjene i optimizacije trenutnih features

Npr: "Ubrzati učitavanje liste kvizova"

Bug Fix Popravci bugova

Ispravljanje problema u kodu

Npr: "Login ne radi na Safari browseru"

Technical Debt Tehnički dug

Refactoring i tehničke optimizacije

Npr: "Refaktorirati auth modul"

Infrastructure Infrastrukturni taskovi

Setup, deployment, monitoring

Npr: "Postaviti CI/CD pipeline"



Balansiranje Product Backloga

Dobar Product Backlog ima balans između različitih vrsta taskova

60%

Features

Nove funkcionalnosti

20%

Bugs

Popravci problema

20%

Tech Debt

Tehničke optimizacije

⚠ Upozorenje:

- ▶ **Previše features:** Tehnički dug se gomila, kvaliteta koda pada
- ▶ **Previše bug fixa:** Nema novih mogućnosti, korisnici nezadovoljni
- ▶ **Previše tech debta:** Korisnicima ne vidljiva vrijednost

💡 **Best practice:** U svakom sprintu imaj mix različitih taskova!



Praktični savjeti za Product Backlog

1 Započni s roadmapom

Imaš li jasnu viziju što želiš postići u sljedećih 6-12 mjeseci?

2 Razbij na Epics

Iz roadmapa izvuci velike cjeline (Epics) koje možeš pratiti

3 Kreiraj User Stories

Svaki Epic razbij na manje User Stories koje se mogu završiti u jednom sprintu

4 Prioritiziraj

Što je najvažnije? Što donosi najviše vrijednosti korisnicima?

5 Održavaj redovito

Backlog refinement jednom tjedno - čisti, dodaj, promijeni prioritete

6 Komuniciraj s timom

Product Owner ne radi sam - traži feedback od svih!



Kako napisati User Story?

Formula:

Kao [tip korisnika] ,
želim [funktionalnost] ,
kako bih [razlog/vrijednost]

✓ ODLIČAN primjer:

Kao **student**, želim **filtrirati kvizove po predmetu**, kako bih **brže pronašao relevantne materijale za učenje**

✓ Jasno tko, što i zašto

✗ LOŠ primjer:

"Dodati filter"

✗ Nema konteksta, ne jasno tko koristi i zašto

INVEST kriteriji

Independent • Negotiable • Valuable • Estimable • Small • Testable



Story Points - Procjena složenosti

Story Points nisu vrijeme! Oni mjere kompleksnost, rizik i količinu posla.

Login gumbom

1 SP

Jednostavna funkcionalnost, poznata tehnologija

Validacija forme

3 SP

Srednje kompleksna, jasna implementacija

Real-time chat

8 SP

Kompleksna, potrebna nova infrastruktura

AI preporuke

13 SP

Vrlo kompleksna, nepoznata tehnologija, visok rizik

💡 **Fibonacci sekvenca:** 1, 2, 3, 5, 8, 13, 21...

Ako je story 13 ili više SP, razbij je na manje dijelove!





Scrum Framework

Najčešći Agile framework za timski rad



Product Owner

Definira prioritete, upravlja backlogom,
predstavlja korisnika



Scrum Master

Uklanja prepreke, vodi ceremonije, štiti
tim



Development Team

Samoorganizirajući, međufunkcionalni,
isporučuje softver



U QuizMaster timu:

Product Owner: Ana (osnivač, vizija proizvoda)

Scrum Master: Marko (iskusni developer, organizacija)

Dev Team: 3 developera + 1 dizajner





Sprint Planning - Kako planirati sprint?



🎯 QuizMaster - Sprint 1 (2 tjedna, 40 SP kapacitet) 💡

User registracija 5 SP

Login funkcionalnost 3 SP

Profil korisnika 5 SP

Kreiranje jednostavnog kviza 8 SP

Lista kvizova 5 SP

Ukupno: 26 SP (ostaje buffer za testiranje i bugove)

Daily Standup (Daily Scrum)

15 minuta, svaki dan, u isto vrijeme



Što sam jučer
napravio?



Što ću danas raditi?



Što mi smeta?

Primjer iz QuizMaster tima:



 **Luka:**

"Jučer sam završio backend za registraciju. Danas ću spojiti frontend. Čekam da Marko deployira test server."

 **Sara:**

"Završila sam dizajn login screena. Danas radim na responsive verziji. Sve ok."

 **Ivan:**

"Radio sam na bazi podataka. Danas nastavljam. Problem: nisam siguran kako strukturirati quiz tablice - trebam pomoć!"



Kanban Board

Vizualizacija rada i ograničenje WIP-a (Work In Progress)

🎮 Interaktivni Kanban za QuizMaster:



To Do

Dizajn login ekrana

API za kvizove

Unit testovi



In Progress (WIP: 2)

User registracija

Baza podataka



Done

Project setup

Wireframes





Kanban principi:

- ▶ **Ograniči WIP:** Ne radi previše stvari odjednom
- ▶ **Kontinuirani tok:** Nema fiksnih sprinteva
- ▶ **Vizualizacija:** Svi vide status projekta
- ▶ **Fleksibilnost:** Prioriteti se mogu mijenjati



Scrum vs Kanban - Što odabrati?

Aspekt	 Scrum	 Kanban
Iteracije	Fiksni sprintevi (1-4 tjedna)	Kontinuirani tok
Uloge	PO, SM, Dev Team	Nema propisanih uloga
Promjene	Ne tijekom sprinta	Bilo kada
WIP limit	Indirektno (sprint kapacitet)	Eksplisitno (max items po stupcu)
Najbolje za	Timove s jasnim ciljevima	Operativne timove s fleksibilnim prioritetima



Koristi Scrum kada:



- ▶ Razvijaš novi proizvod
- ▶ Imaš jasne ciljeve po iteracijama
- ▶ Tim je fokusiran na jedan projekt
- ▶ Potrebna je predvidljivost isporuke



Koristi Kanban kada:



- ▶ Radiš support/maintenance
- ▶ Prioriteti se često mijenjaju
- ▶ Različite vrste zadataka
- ▶ Potrebna je maksimalna fleksibilnost

Scrumban & Kanplan - Hibridni pristupi

Scrumban = Scrum + Kanban

Kombinira najbolje od oba svijeta

- ▶ **Od Scruma:** Sprintevi, uloge (PO, SM), ceremonije
- ▶ **Od Kanbana:** Vizualna tabla, WIP limiti, kontinuirani tok
- ▶ **Fleksibilnost:** Sprint backlog se može ažurirati

Kanplan (Jira način)

Kanban + Backlog grooming (bez sprinteva)

- ▶ Kontinuirani razvoj kao Kanban
- ▶ Ali s strukturanim backlogom
- ▶ Idealno za timove koji ne žele commitment sprinteva



Praktični savjeti za vaš projekt

1 Započnite s Product Backlogom

Napišite 10-15 user stories za vaš projekt. Ne morate sve, samo najvažnije!

2 Procijenite Story Points

Koristite Planning Poker s timom. Raspravljajte o razlikama u procjenama!

3 Planirajte prvi sprint

Izaberite najviše prioritetne stvari. Budite realni s kapacitetom (radite 10h/tjedan, ne 40!).

4 Koristite alate

Besplatni alati: Trello, GitHub Projects, Notion, ClickUp Free tier

5 Svakodnevna komunikacija

Daily standup preko Discord/Slack - 5 minuta je dovoljno za studentski projekt!

Jira - Alat za Product Backlog

Jira Software

Što je Jira?

- ▶ Najpopularniji alat za Agile timove
- ▶ Upravljanje Product Backlogom
- ▶ Scrum & Kanban boards
- ▶ Sprint planning i tracking
- ▶ Besplatna verzija za male timove

Ključne mogućnosti:

- ▶ **Backlog management** - organizacija Epics i Stories
- ▶ **Sprint planning** - povlačenje taskova u sprint
- ▶ **Board views** - Scrum ili Kanban prikaz
- ▶ **Reporting** - burndown charts, velocity tracking
- ▶ **Collaboration** - komentari, mentions, notifikacije



Alternativni alati za Product Backlog



Jira

- + Najpopularniji, kompletan feature set
- Može biti kompleksan za početnike



Trello

- + Jednostavan, vizualan, besplatan
- Manje features za kompleksne projekte



Notion

- + Fleksibilan, odličan za dokumentaciju
- Treba više setupanja



GitHub Projects

- + Integrirano s kodom, besplatno
- Osnovno za Agile management



ClickUp

- + All-in-one, besplatna verzija solidna
- Može biti overwhelming



Asana

- + User friendly, lijep interface
- Manje specifičan za Agile

Što trebate napraviti?



1. Kreirati Atlassian (Jira) account

Registrirajte se na: atlassian.com/software/jira/free

2. Kreirati Jira space

Naziv: {GRUPA}-{NAZIV_PROJEKTA}

Npr: "Grupa3-QuizMaster"

3. Dodati članove tima i nastavnika

Email nastavnika: aleksander.radovan@algebra.hr

4. Popuniti Product Backlog (15 - 1 bod)

- Kreirajte Epics za vaš projekt
- Raspisati User Stories za svaki Epic
- Dodati taskove i subtaskove

5. Prioritizirati i procijeniti (16 - 1 bod)

- Postaviti prioritete (High/Medium/Low)
- Dodati Story Points procjene
- Organizirati po modulima



Sažetak i sljedeći koraci



Što smo naučili:

- ▶ Product Backlog je prioritizirana lista svih taskova
- ▶ Roadmap i Requirements su temelj Backloga
- ▶ Epic = veliki task, User Story = mala jedinica
- ▶ Product Owner upravlja, ali NE radi sam
- ▶ Backlog je živi dokument koji se mora održavati
- ▶ Backlog Refinement je ključan proces



Sljedeći koraci:



- ▶ Kreirajte Jira account i projekt
- ▶ Popunite Product Backlog za vaš projekt
- ▶ Planirajte prvi sprint
- ▶ Održavajte backlog kroz cijeli semestar

Hvala na pažnji!

