



**PROGRAMSKO
INŽENJERSTVO**

AGILNE METODOLOGIJE

Što nas čeka danas?

Uvod u Agile metodologije

Zašto je waterfall outdated?

Lean principi

Toyota filozofija za software dev

Principi Agilnog razvoja

12 pravila koje morate znati

Quick Poll!

Što mislite - utječu li rokovi i dokumentacija na način rada?

Da, puno! 

Ne baš 

Tradicionalni pristup = 🤪

⊗ **Vodopadni model**

- ✓ Jasni rokovi (ali nerealni)
- ✓ Fiksna cijena (teoretski)
- ✗ Kasna isporuka
- ✗ Bugovi se otkriju prekasno
- ✗ Tone dokumentacije

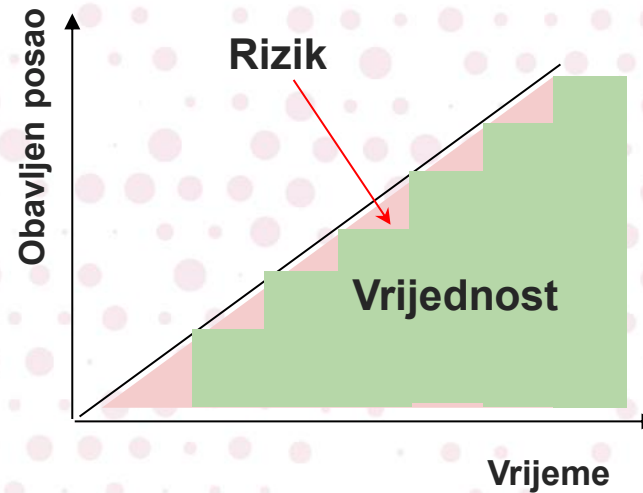
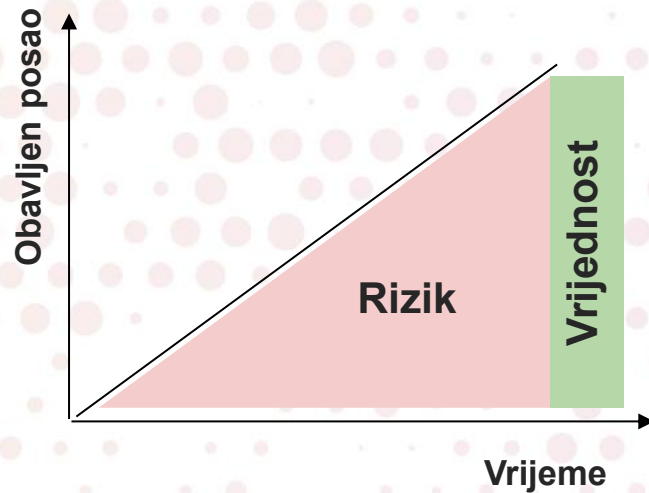


**Rizik raste kako projekt traje
duže!**

Sve ili ništa na kraju

Nedostaci tradicionalnog pristupa

Velika isporuka



**Primjer natječaj za
eHealth u RH, tko sve
ispunjava uvjete?**

Veliki dizajn unaprijed

Big Design Up Front (BDUF)

Planirati sve prije nego što započnemo

Filozofija pristupa

Vjerovanje da možemo predvidjeti i planirati sve unaprijed, prije nego što započnemo s radom. Sve mora biti savršeno isplanirano na papiru prije prvog reda koda.

Pretpostavke koje (ne) drže vodu

1



Problem je dobro definiran

Pretpostavka da neće biti promjena zahtjeva i da sve znamo od početka

2



Možemo predvidjeti probleme

Dizajn može predvidjeti sve probleme prije prototipiranja ili testiranja

3



Planiranje je jeftinije

Cijena planiranja je manja od cijene popravke kroz eksperimentiranje

Realnost izgleda drugačije



 **Cijena
discipliniranog procesa**

>

 **Cijena pokušaja i
popravka**

⊘ MIT: Više planiranja = Bolji rezultat

Što brojke govore?



Da li to znači da je Agile bolji za sve projekte?



📈 Porast broja projekata kojima se upravlja agilnim načinom vođenja

⚠️ Česti su tradicionalni elementi u Agilnom vođenju projekata



Waterfall

✖ Neuspjeli 29%

29%

⚠️ Izazovni 14%

14%

✓ Uspješni 57%

57%



Agile

🏆 WINNER

✖ Neuspjeli 9%

9%

⚠️ Izazovni 42%

42%

✓ Uspješni 49%

49%

Zaokret prema Agile metodama

Od tradicionalnog ka modernom



1

Praktični pristup

Fokus na stvarnim rezultatima umjesto beskonačnog planiranja. Djeluj odmah, uči usput.



2

Minimalizam procesa

Manje dokumentacije, više komunikacije. Ukloni sve što ne dodaje vrijednost.



3

Eksperimentiranje i observacija

Testiraj hipoteze, skupljaj podatke, uči iz grešaka. Istraživanje umjesto pretpostavki.



4

Stalna prilagodba

Fleksibilnost je ključ uspjeha. Prilagodi se promjenama, ne bori se protiv njih.



Ključ uspješne transformacije

Od **prediktivnog planiranja** ka **adaptivnom izvođenju**. Svaki princip gradi na prethodnom, stvarajući kompletnu Agile filozofiju.

🍷 Komplementarni pristupi

Lean + Agile = ❤️

Lean i Agile stoje na istim ili sličnim principima. Ne konkuriraju jedno drugome i ne isključuju jedno drugo.





Lean roots 🌱
Toyota, Japan



Lean



1950-1990

Izvorno nastao kao japanska proizvodna filozofija u Toyoti



Utjecaj na ljude

Veliki utjecaj na ljude koji su utemeljili agilni razvoj



Adaptacija za softver

Mary i Tom Poppendieck Lean principe prenijeli u svijet softvera



Agile razvoj



Krovni pojam

Agilni razvoj je krovni pojam za metode razvoja softvera temeljene na posebnim principima



Današnja praksa

Danas se mnogi Lean principi smatraju općeprihvaćenim praksama u agilnom razvoju



Međusobno jačanje

Lean i Agile se međusobno nadopunjuju i jačaju



Ključna spoznaja

Lean i Agile nisu konkurenti - oni su saveznici u borbi protiv tradicionalnih, nefleksibilnih pristupa. Zajedno stvaraju ekosistem koji omogućuje brže, efikasnije i kvalitetnije rezultate.

Lean: 3M Principi



Muri

(無理)

Preopterećenost

OVERBURDEN

Kad te šef zaspe sa 10 taskova odjednom



Mura

(斑)

Nejednakost

UN-EVENNESS

Jedan task traje 1h, drugi 40h



Muda

(無駄)

Gubitak

WASTE

Meetinzi koji su mogli biti email



Goal: Minimiziraj gubitak, balansiraj workload!



Lean u Software Engineeringu



Borba protiv Mude

Što je waste u dev-u?



Nepotrebne ekstra funkcionalnosti



Čekanje na druge timove/procese



Prevelika dokumentacija



Uzaludni sastanci

 **Gemba Walk**

Odi na mjesto gdje se radi i uoči waste!



Borba protiv Mure

Kako balansirati?



Dekompozicija na male dijelove



Jedinica rada = 1 dan



Kontinuirane isporuke



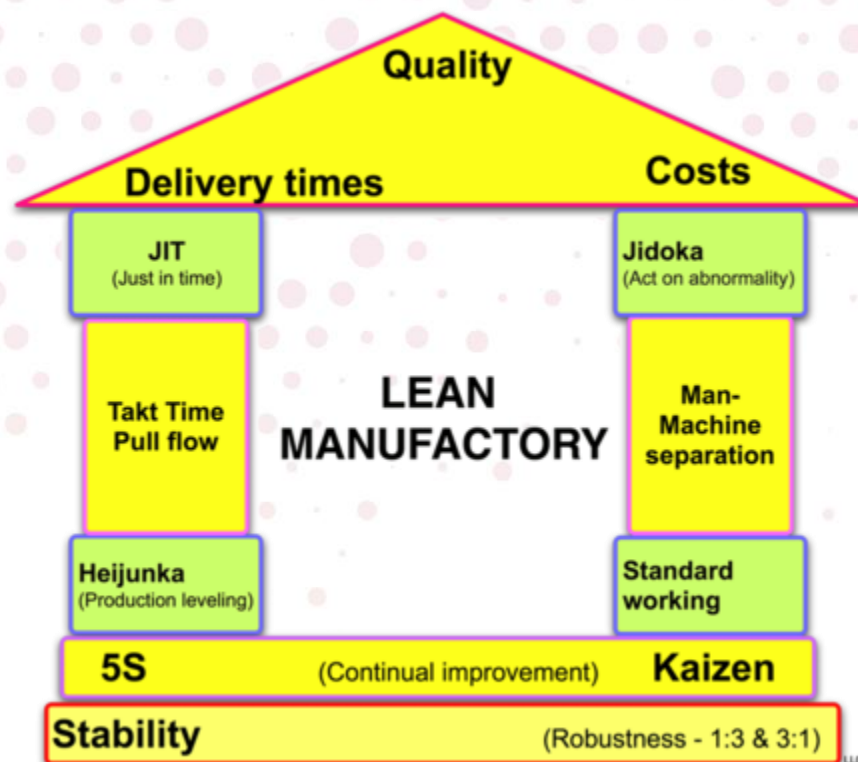
Jednake težine taskova

Lean = manje procesa, više valuea! 

Lean House

Postoji puno više od principa - cijela metodologija





Nas zanima samo utjecaj na Agile softverski proces







Agile Manifest (2001)

17 legendi se sastalo i reklo: "Mora postojati bolji način!"

VIŠE vrijednosti dajemo: ✨

-  Ljudima i interakcijama
-  Funkcionalnom softveru
-  Suradnji s klijentom
-  Reagiranju na promjene

Nego: 😬

-  Procesima i alatima
-  Dokumentaciji
-  Pregovaranju oko ugovora
-  Praćenju plana

Metodologija agilnog razvoja

Agilni manifest

Agile Alliance, <http://agilemanifesto.org/> , 2001



12 Principa - TL;DR verzija



Zadovoljan klijent = win



Welcome promjene (da, i u zadnji čas)



Často isporučuj (tjedni/mjeseci)



Dev + Business = BFF



Motivirani ljudi > sve



Face-to-face komunikacija FTW



Working software = progress



Održiv tempo (no burnout)



Tehnička izvrsnost je must



Jednostavnost je art



Self-organizing timovi



Retrospektiva i improve

Waterfall vs Agile

Waterfall



Plan EVERYTHING first

Opseg: FIKSAN 


Kvaliteta: $\sqrt{\text{(")}}\sqrt{\text{(")}}$


"Nadamo se da smo sve dobro isplanirali..."

Agile



Adapt as you go

Opseg: FLEKSIBILAN 






Kvaliteta: HIGH 

"Prilagođavamo se na temelju feedbacka"

Real World Example

Spotify, Netflix, Amazon...

Zašto uspijevaju? ✨

-  Brza isporuka novih featuresa
-  Kontinuirane iteracije
-  Autonomni timovi
-  Data-driven odluke
-  Eksperimentiraj i uči



**Move fast & don't break
things**

(ali ako se nešto slomi, brzo popravi)

Scrum = Agile GOAT 🐐

56% projekata koristi Scrum!

(Ostale metode su u minornoj ligi)



Sprints
(1-4 tjedna)



Daily Standups
(15 min)



Sprint Goals
(jasni ciljevi)



Retrospective
(što smo naučili)

It's like playing a video game with clear levels! 🎮

Kako radimo u Agile?

Zahtjevi



Dizajn



Dev



Test

Sve paralelno, u malim dijelovima! 

 **NE radimo:**

Čitavu bazu podataka pa tek onda frontend

 **Radimo:**

Jednu funkcionalnost od baze do UI-ja

Pojam Agile

Agilne metodologije

💡 Agile je kišobran pod kojim se nalazi mnoštvo različitih metodologija



Scrum

SPRINT-BASED FRAMEWORK



XP

EXTREME PROGRAMMING



DSDM

DYNAMIC SYSTEMS
DEVELOPMENT



FDD

FEATURE DRIVEN
DEVELOPMENT



Crystal

LIGHTWEIGHT METHODOLOGY



Kanban

VISUAL WORKFLOW

Koje se najviše koriste? ●

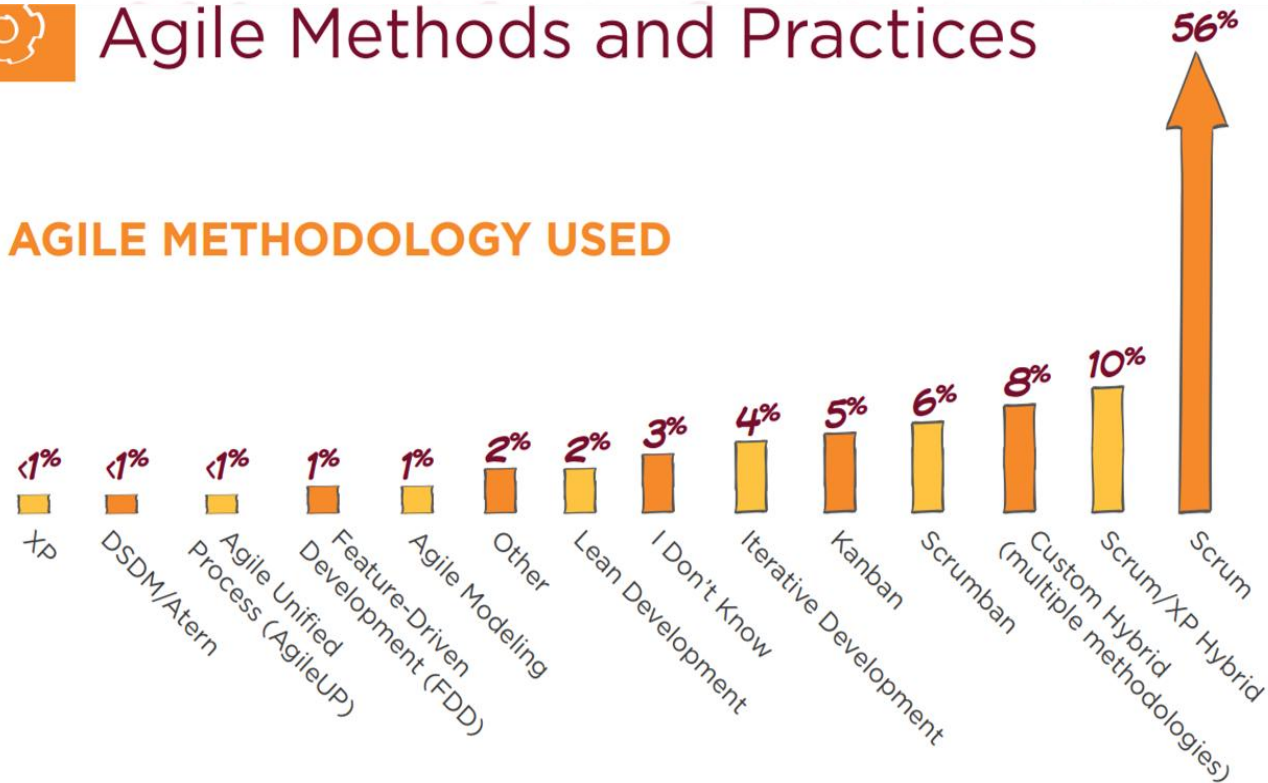
AGILE METHODOLOGY USED

👤 Scrum je drugo ime za Agile!



Agile Methods and Practices

AGILE METHODOLOGY USED



Agilni Manifest

Metodologija agilnog razvoja



Izvor: Autoindustrija

Principi preuzeti iz automatizacije autoindustrije (Toyota, Lean proizvodnja)



Transformacija

Prilagodba i evolucija principa za softverski razvoj



Softverski razvoj

Prilagođeni razvoju softvera pod pretpostavkom idealnog svijeta



Analiza principa agilnog razvoja



Industrijski koreni

Agile principi nisu izmišljeni - izvorno dolaze iz desetljeća iskustva autoindustrije, posebno Toyota's Lean praksi i Just-In-Time proizvodnje.



Kreativna adaptacija

Softverski razvoj ima svoje specifičnosti - principi su prilagođeni za kreativni, intelektualni rad umjesto fizičke proizvodnje.



Dokazana učinkovitost

Statistika pokazuje **velike uštede** kod timova koji koriste Agile pristup. Brže isporuke, manje otpada, veća zadovoljstvo kupaca - principi koji su radili u Toyoti, rade i u razvoju softvera.

Vrijednosti proizvoda

Balans četiri ključne dimenzije



⚡ Utjecaj na isporuku proizvoda



Pravilo kompromisa

Ne možeš imati sve! **Izabire se i varira jedna od vrijednosti (ili više)**. Želiš brže? Morat ćeš podići cijenu ili smanjiti opseg. Želiš jeftinije? Potrebno je više vremena ili niža kvaliteta. To je umjetnost project managementa - **naći pravi balans**.

Iterativni pristup

Način rada

Tim radi sve po malo



Zahtjevi



Dizajn



Ostvarenje



Testiranje

Zahtjevi

Dizajn

Ostvarenje

Testiranje



Paralelni rad

Umjesto da čeka završetak jedne faze, tim radi **kontinuirano na svim fazama**. Dok se razvija jedan feature, istovremeno se pišu zahtjevi za drugi, testira treći, i dizajnira četvrti. Rezultat? **Brži feedback loop i brža isporuka!**



Način dekompozicije i isporuke



Ne graditi slojeve horizontalno

Tradicionalni pristup gradi cijeli sloj odjednom - svi frontend elementi, zatim svi backend elementi. Problem? Nema funkcionalnog proizvoda dok sve nije gotovo.



Graditi prema potrebi funkcionalnosti

Agile pristup reže vertikalno - svaka isporuka sadrži sve slojeve potrebne za jednu funkcionalnost. Rezultat? Funkcionalni proizvod nakon svake iteracije!

✗ Horizontalno



Prezentacijski sloj

Sloj poslovnog upravljanja

Podatkovni sloj

⚠ Nema funkcionalnosti dok sve nije gotovo

✓ Vertikalno



Feature 1

Feature 2

Feature 3

Feature 4

UI

UI

UI

UI

Logic

Logic

Logic

Logic

Data

Data

Data

Data

✨ Svaki feature je kompletan i funkcionalan



Prezentacijski sloj



Sloj poslovnog upravljanja



Podatkovni sloj



Ključna razlika

Vertikalni rezovi omogućuju kontinuiranu isporuku vrijednosti. Nakon svake iteracije imaš funkcionalni proizvod koji korisnik može koristiti i testirati. Horizontalni pristup znači mjeseci rada bez ikakve vrijednosti - sve ili ništa!

Key Takeaways



Agile = brzina + fleksibilnost



People > Process



Adapt, don't panic



Kvaliteta je prioritet

Pitanja? 

Hvala na pažnji!

