Arhitekta softvera: Kako efikasnije proizvoditi kvalitetniji (i monetarno vrijedniji) softver?

Jasmin Jahić

Sarajevo, Bosna i Hercegovina

8. maj 2024

Teme

- Pozicioniranje uloga u IT projektima (menadžeri, arhitekte, programeri)
- Problemi van opsega odgovornosti programera i menadžera
- Komunikacija i organizacija u IT projektima
- Vježba: odluke i komunikacija
- Uvod u softverske arhitekture
- Uloga arhitekte u digitalnim kompanijama

Ciljevi

- Prepoznati izazove koji se pojavljuju u IT projektima a van opsega su onoga što rade programeri i menadžeri
- Prepoznati vrijednost dokumentacije: kvalitet umjesto kvantiteta
- Prepoznati mogućnosti za unaprijeđenje kvaliteta softvera
- Unprijeđenje poslovanja u malim i srednjim biznisima
- Jasno predstavljanje projekata na odgovarajućem nivou apstrakcije

Zašto?

Kako efikasnije proizvoditi kvalitetniji (i monetarno vrijedniji) softver?

Previše mali za podjele, previše kvalitetni i žilavi da nestanemo.

Vaše iskustvo?

Stanje na tržištu?

Budućnost? Vlastiti proizvodi ili outsourcing?

Jasmin Jahić

- Direktor studija za kompjuterske nauke, Queens' College, University of Cambridge
- ARM, Arhitekta za softverska riješenja, softverski definisana vozila
- Izvršni direktor, YNOT istraživački institut, University of Cambridge
- Istraživač, Odsjek za informatiku, University of Cambridge
- **Menadžer projekata i istraživač, Fraunhofer Institute for Experimental Software Enginering**, Kaiserlsuatern, Njemačka
- Gostujući istraživač, Free University of Bolzano, Bolzano, Italy
- Softverski inženjer, AtlantBH, Sarajevo

- PhD, University of Kaiserslautern, Germany
- Master of Robotics, Ecole Centrale de Nantes, France
- Fakultet Elektrotehnike, Tuzla
- Srednja Elektrotehnička škola, Tuzla
- Osnovna škola u Kalesiji

RASPORED

9:00 — Apstrakcija i komunikacija
- Uloge u IT projektima i njihove odgovornosti
- Izazovi van rokova, upravljanja ljudskim resursima, te standardnih inženjerskih aktivnosti
- Apstrakcija u IT projektima
- Komunikacija u IT projektima

10:15 — Vježba 1: Komunikacija

Pauza: 10:00 - 10:15

10:45 — Indikatori kvaliteta u projektima
- Referentne arhitekture i domensko znanje
- Kvalitet funkcionalnosti u datom kontekstu
- Interni i eksterni indikatori kvaliteta
- IT i ne-IT klijenti: razlike i sličnosti

11:30 — Vježba 2: Kvalitet IT projekata

Pauza: 11:30 - 11:40

12:00 — Sesija 3: Arhitekta softvera: Kako efikasnije proizvoditi kvalitetniji (i monetarno vrijedniji) softver?

12:30

+ Diskusija

Reference

- The Software Architect Elevator: Redefining the Architect's Role in the Digital Enterprise, Gregor Hohpe, 2020
- Jan Bosch, https://janbosch.com/blog/

Prezentacije i kontakt

- Pošaljite email na: jasmin.jahic@gmail.com

Apstrakcija i komunikacija

Izazovi u IT projektima

- Sastanci sa potencijalnim kupcima
- Razumijevanje korisničkih zahtjeva
- Pregovaranje oko cijene i opsega/veličine projekta
- Ubijediti kupce u vrijednost vašeg rada (zašto je cijena veća nego kod konkurencije)
- Sastavljanje ugovora
- Planiranje i arhitektura softverskog sistema/temeljne odluke
- Programiranje, testiranje, CI/CD
- Paralelizacija razvoja softvera (više progamera/timova mogu raditi paralelno)
- Redovna komunikacija sa kupcima
- Komunikacija sa različitim vrstama uloga u projektu (marketing, biznis, pogrameri, ljudski resursi)
- Prenos znanja između članova istog tima ili između timova
- Zapošljavanje novih programera, proces osposobljavanja za rad (onboarding)

- Osigurati da svi u projektu razumiju sistem
- Osigurati da svi u projektu razumiju trenutno stanje razvoja
- Preuzimanje postojećih projekata
- Integracija softvera razvijenog od strane različitih timova
- Komunikacija između različitih timova programera
- Dokumentacija (šta dokumentovati, kako održavati)
- Osiguravanje kvaliteta softvera
- Podrška
- Rokovi
- Sastanci
- Plan razvoja karijere (za uposlenike)
- Unaprijeđenja

Da li nešto nedostaje?

Napisati na panelu

Uloge u IT projektima (stakeholders, „role“)

KUPCI

BIZNIS (CEO, CTO)

MARKETING

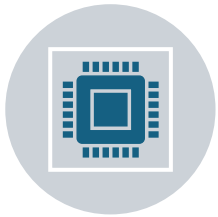
MENADŽERI

LJUDSKI RESURSI (HR)

A stakeholder is anyone who is affected by or concerned about outcome of a project

VOĐE DIVIZIJA/ODJELA

VOĐE TIMOVA



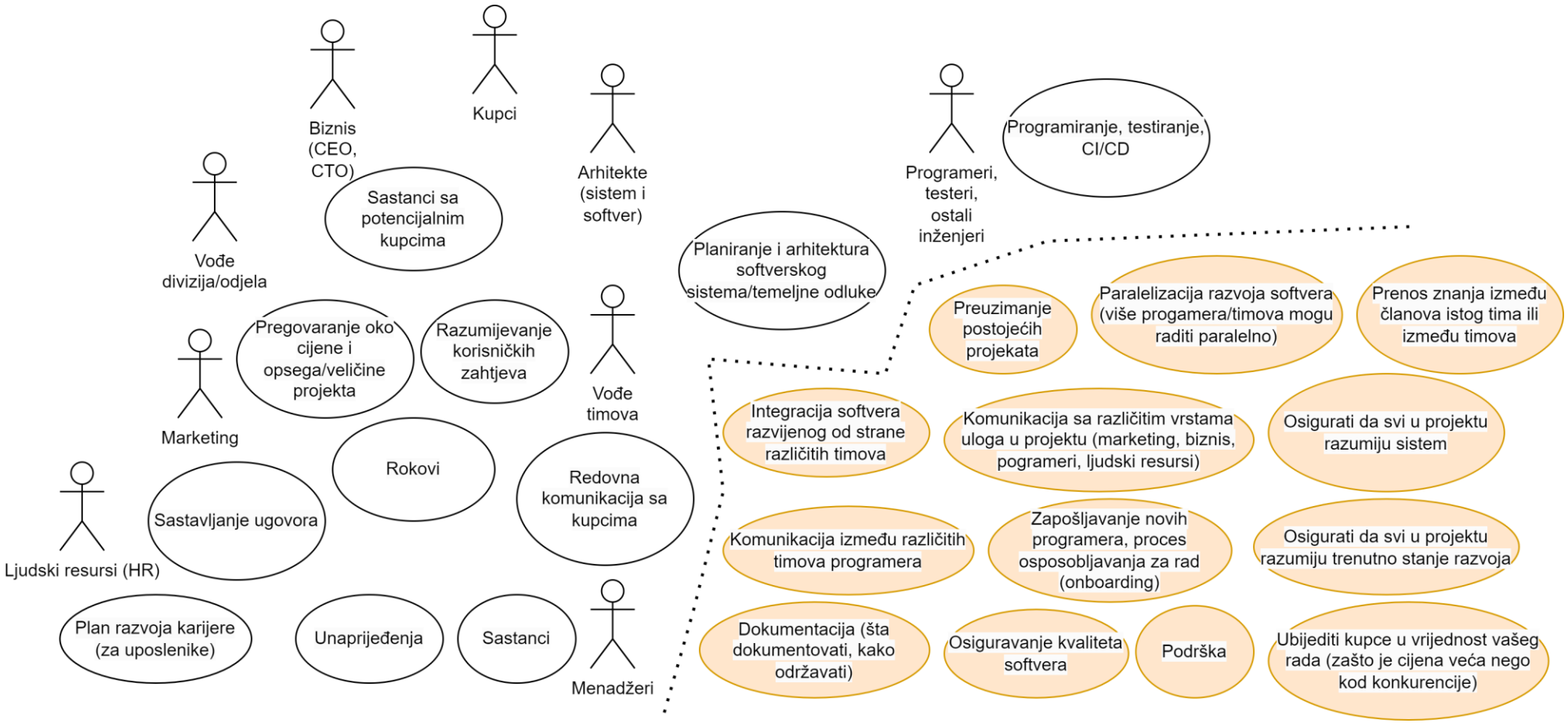
ARHITEKTE (SISTEM I SOFTVER)

PROGRAMERI, TESTERI, OSTALI INŽENJERI

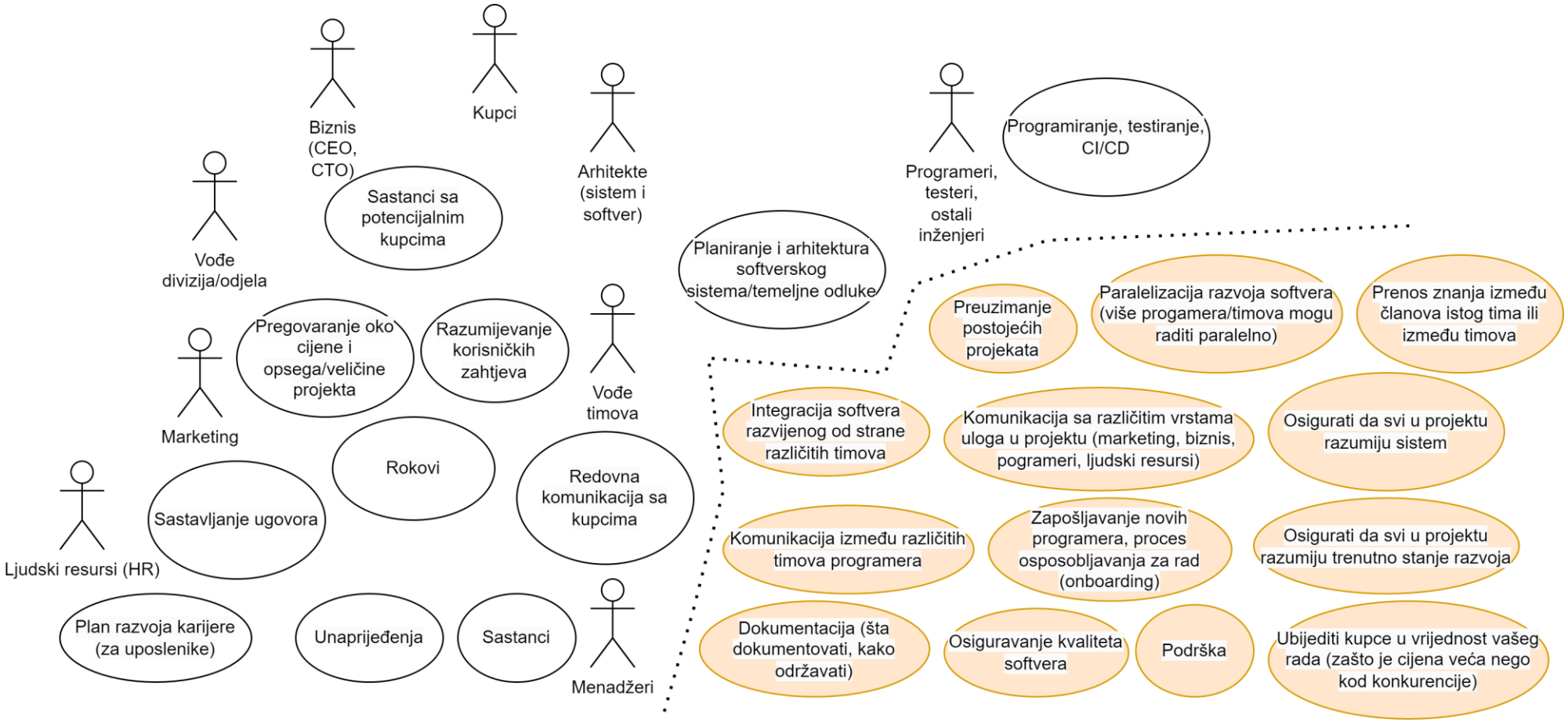
Koje uloge nedostaju?

Napisati na panelu

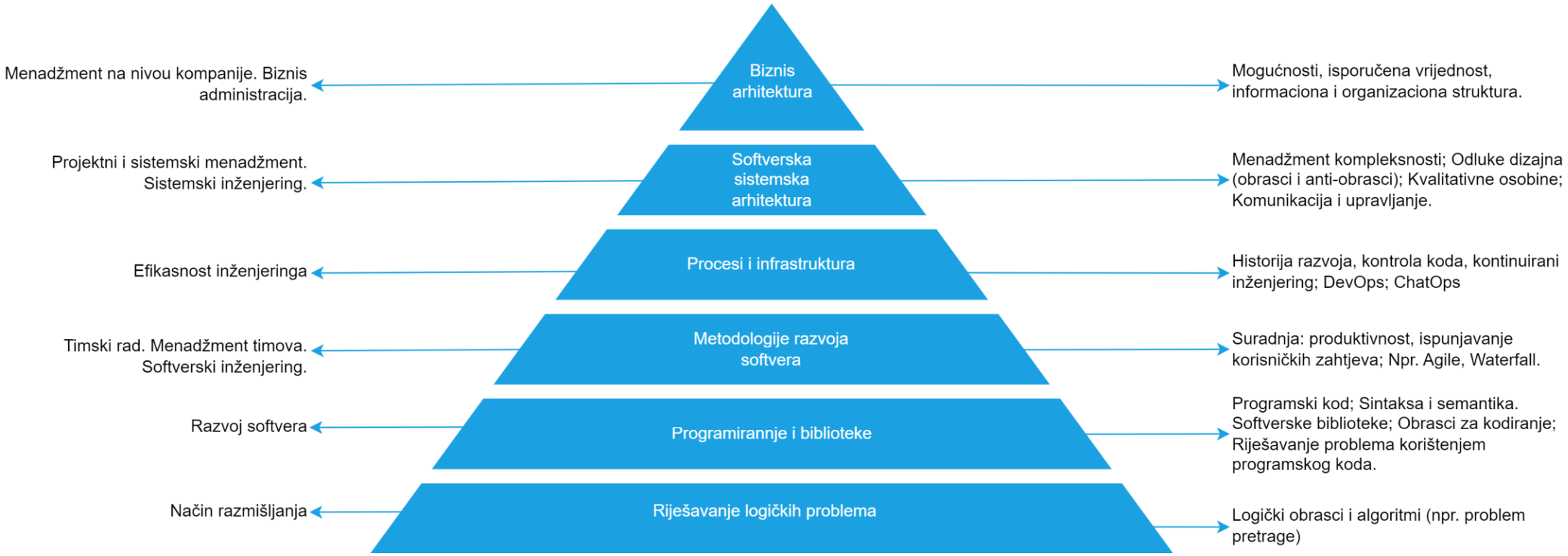
Mapiranje izazova na uloge – Da li se slažete?



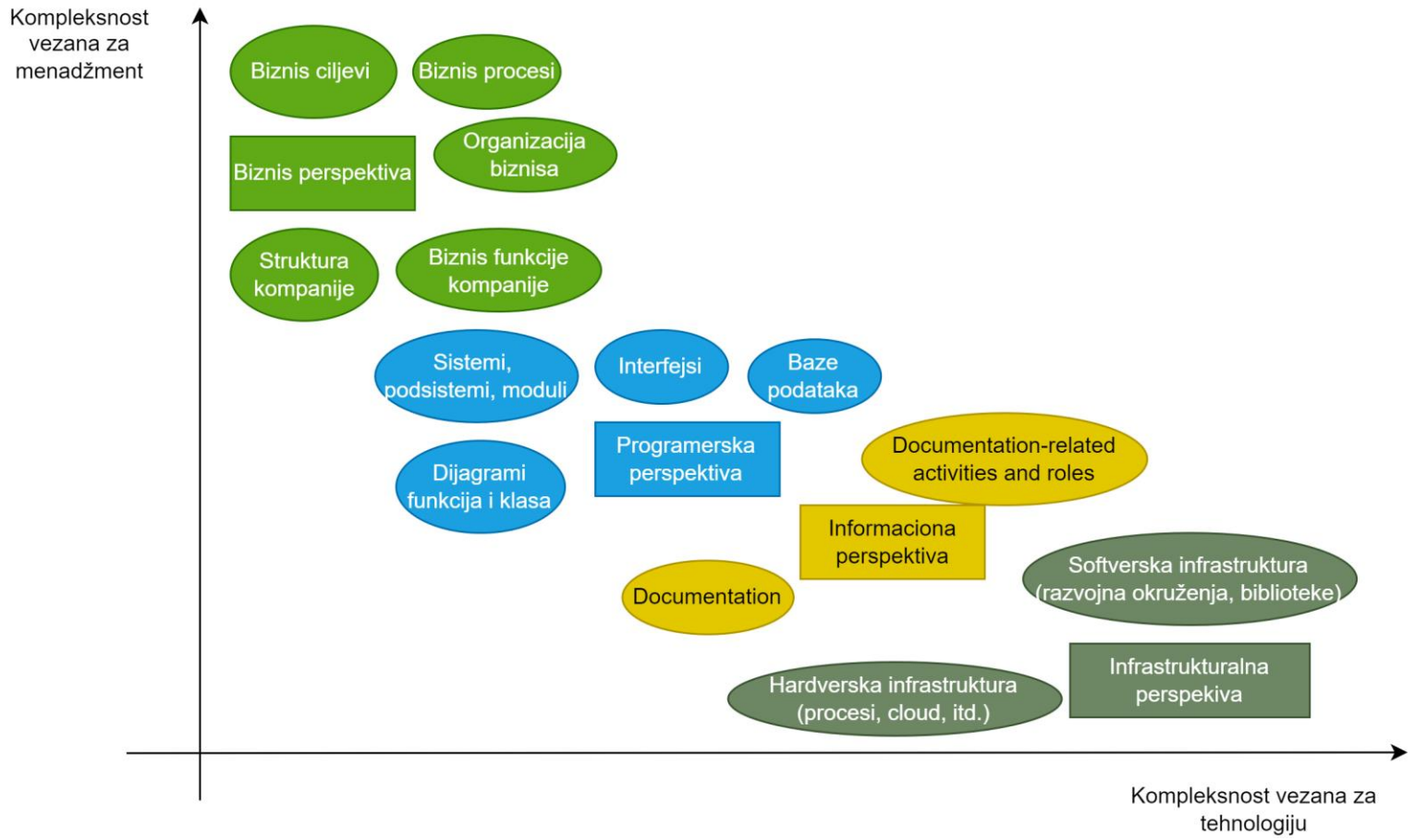
Mapiranje izazova na uloge – Potencijalni problemi sa narandžastim izazovima?



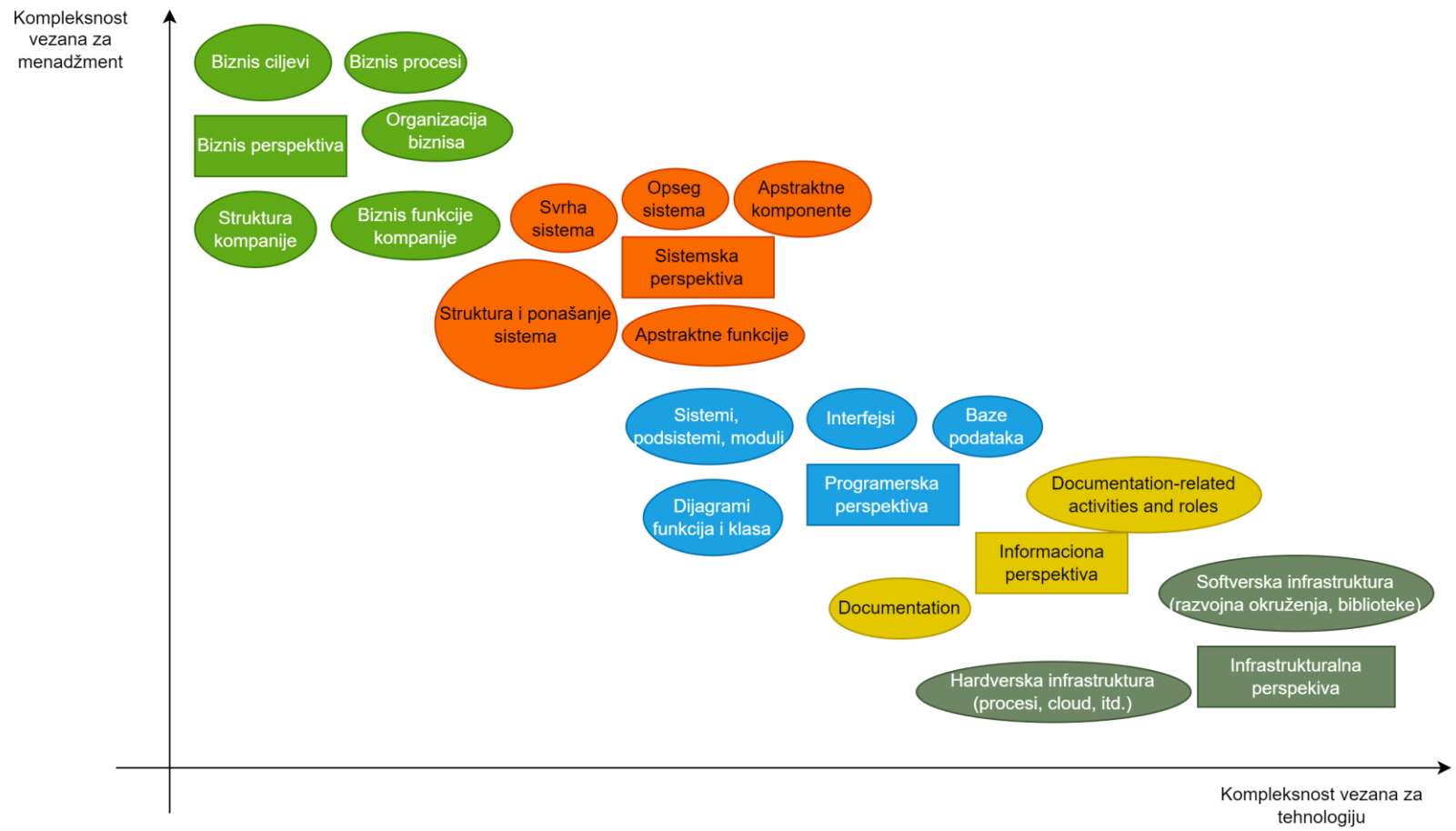
Proizvodnja softvera



Strma tranzicija od biznisa perspektive ka programerskoj perspektivi



Uloga sistemske perspektive



Ciljevi softverske sistemske arhitekture

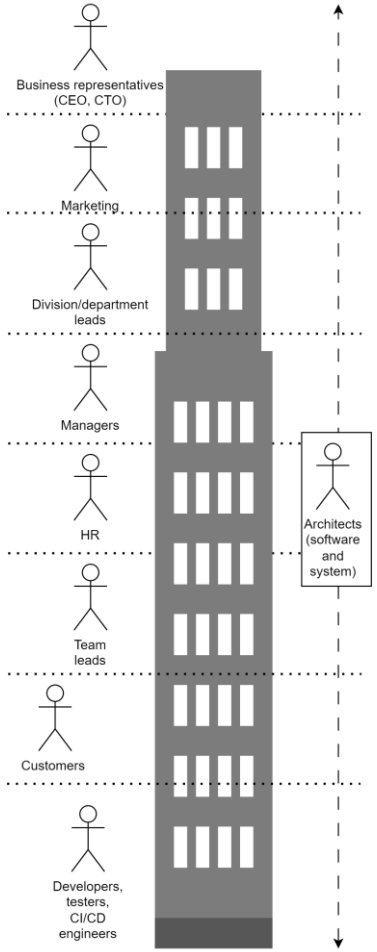
NAMETNUTI RIJEŠENJA (NEKI DETALJI NISU BITI, NEKI DETALJI SU KLJUČNI I KRITIČNI ZA USPJEH PROJEKTA)

OMOGUĆITI PARALELNI RAZVOJ SOFTVERA

UVJERITI NE-TEHNIČNE ULOGE U PROJEKTU DA RIJEŠENJE ODGOVARA ZAHTJEVIMA KUPACA

UVJERITI TEHNIČKE ULOGE U PROJEKTU DA RIJEŠENJE ODGOVARA ZAHTJEVIMA KUPACA

KOMUNIKACIJA



The Software Architect Elevator: Redefining the Architect's Role in the Digital Enterprise, Gregor Hohpe, 2020

Odgovornosti arhitekte softvera

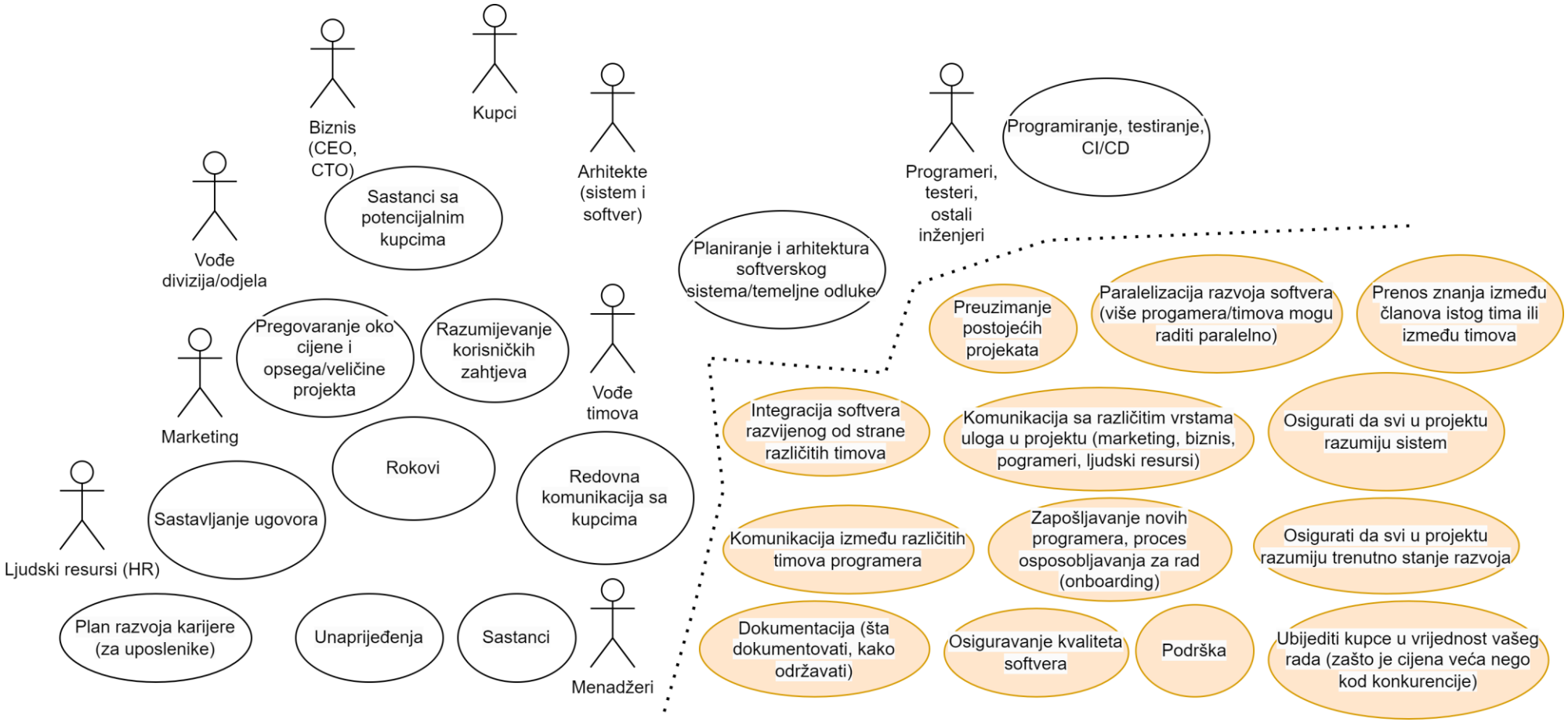
- Povezuje sistemske zahtjeve sa biznis ciljevima
- Učestvuje u odabiru ključnih tehnologija
- Komunikacija tehnoloških principa svim nivoima menadžmenta
- Osigurava da sistem ima željene kvalitativne osobine
- Radi dekompoziciju sistema na podsisteme, module, komponente, servise, itd.
- Održava apstraktni model sistema razumljiv svim učesnicima u projektu
- Kreira dizajn sistema koji trasira strategiju razvoja
- Predviđa i riješava probleme sa integracijom različitih dijelova sistema
- Planira budućnost sistema (uključujući i ljudske koji održavaju sistem živim, osigurati prenos znanja)
- Provodi vrijeme u „arhitektonskom liftu“

Kako se vi nosite sa ovim izazovima?

Koje izazove u ovoj oblasti želite riješiti?

Napisati na panelu

Koje od ovih izazova bi željeli riješiti? (napisati na panelu)



Svi navedeni izazovi su teški problemi

- Opaki (wicked) problemi
  - https://www.stonybrook.edu/commcms/wicked-problem/about/What-is-a-wicked-problem#:~:text=As%20described%20by%20Rittel%20and,false%2C%20only%20good%20or%20bad.
- Dileme u generalnoj teoriji planiranja
  - Problemi planiranja su opaki (wicked) problemi
  - https://www.jstor.org/stable/4531523
- Problemi koje je teško razumijeti
- Još teže riješiti

Karakteristike opakih (wicked) problema

Nemaju konačnu formulaciju.

Nemaju pravilo za „kraj“. Ne postoji način da se odredi kada su riješeni.

Riješenja nisu tačna ili netačna (true or false) već dobra ili loša (u arhitekturi, adekvatna ili neadekvatna).

Ne postoji način da se testira riješenje za ovakve probleme.

Ne mogu se posmatrati kroz pokušaje i greške. Svaki pokušaj je bitan.

Ne postoji konačan broj riješenja ili pristupa riješenjima za ove probleme.

Svi ovakvi problemi su jedinstveni.

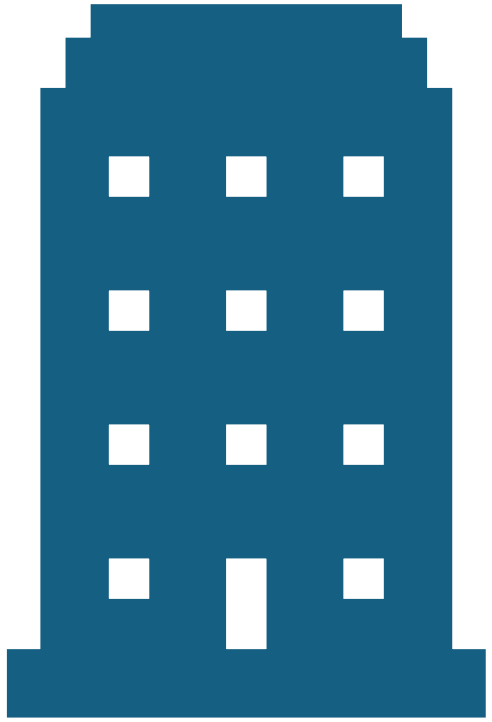
Uvijek mogu biti opisani kao simptomi nekih drugih problema.

Dilemmas in a General Theory of Planning, Horst W. J. Rittel and Melvin M. Webber, Policy Sciences, Vol. 4, No. 2 (Jun., 1973), pp. 155-169

Način na koji je problem opisan na neki način određuje njegovo moguće riješenje.

Oni koji planiraju riješenja za ovakve probleme su odgovorni za posljedice riješenja (posljedice mogu biti šarolike i široke).

Imamo li uopšte ikakve šanse?



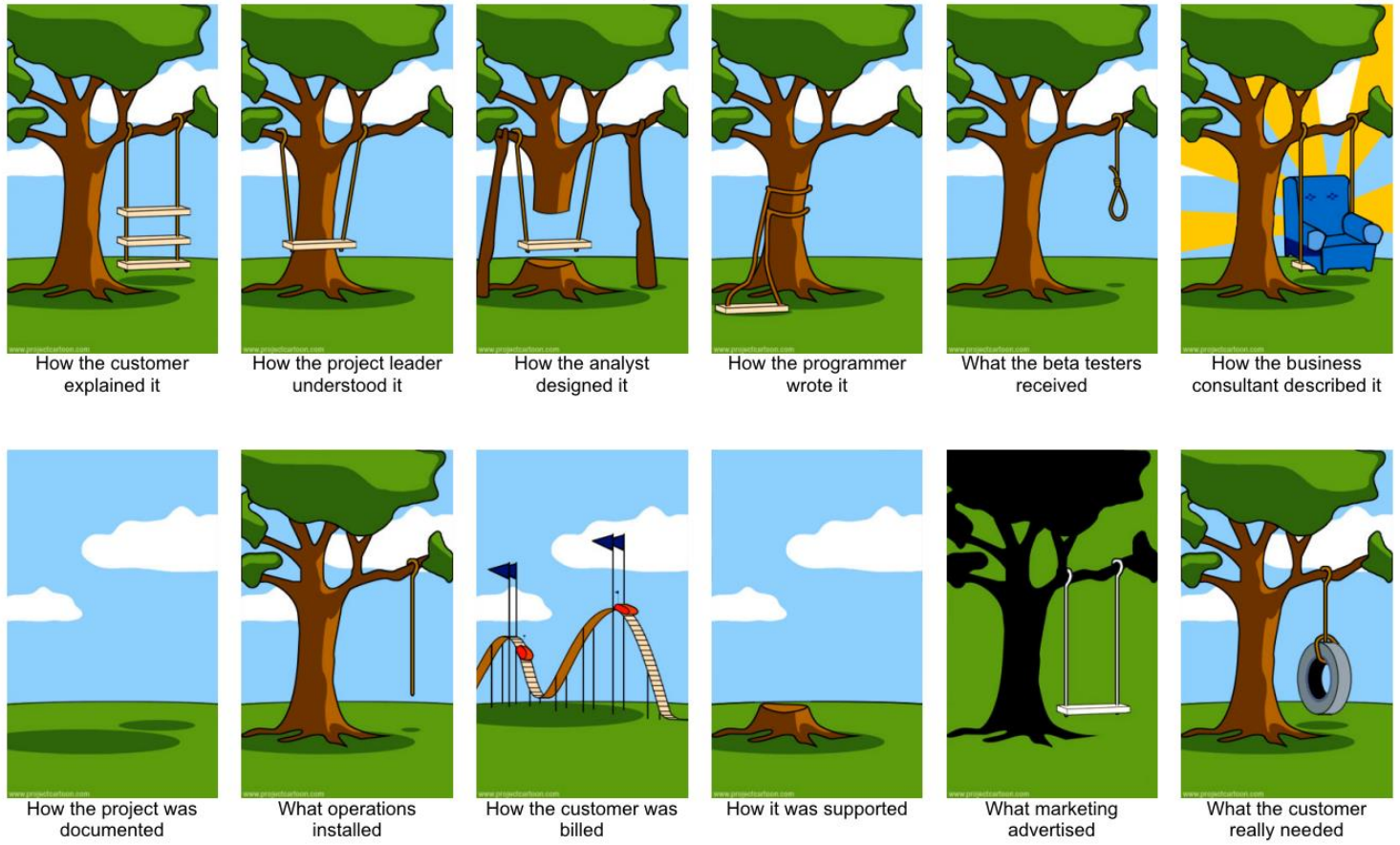
Arhitekta softverskih sistema

- Adekvatno, umjesto tačno ili optimalno riješenje
- Fokus nije na posebnoj poziciji, već na riješavanju izazova koji je neko (arhitekta) mora riješiti
- Jedna osoba može biti arhitekta
- Timovi koji su sastavljeni samo od arhitekata
- Ili, izazovi su raspoređeni između više osoba

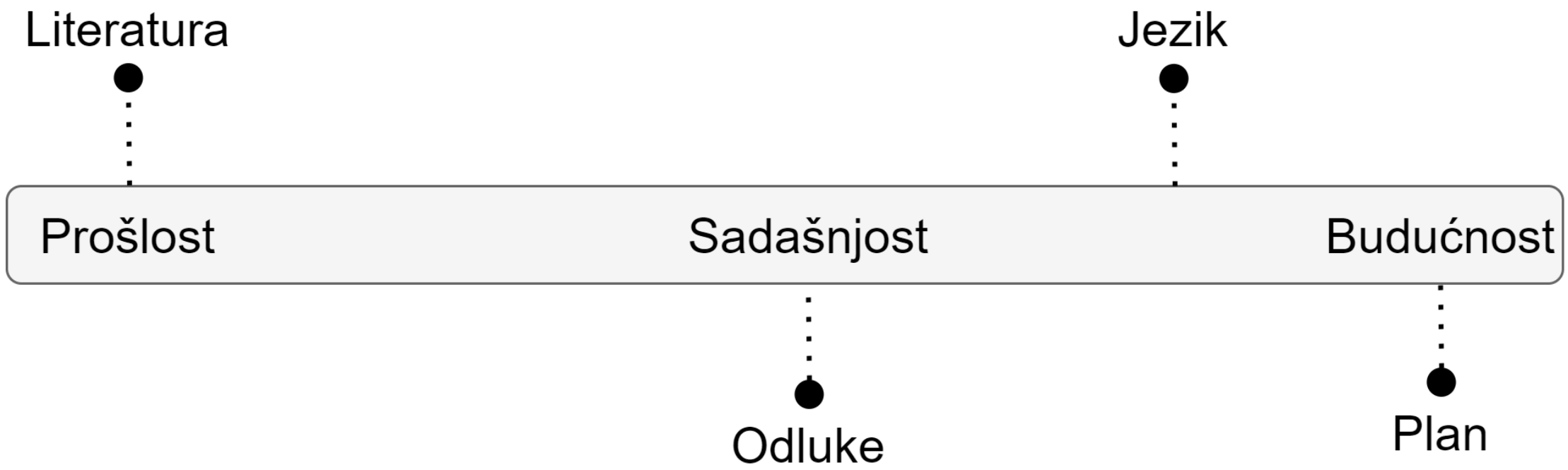
- Eksplicitno definišite ko je odgovoran za ove izazove
  - U suprotnom, niko nije odgovoran
  - Ili, neko će preuzeti odgovornost
  - Šta je gore od ova dva izbora?

Problem sa komunikacijom

- Arhitekte prije svega omogućuju komunikaciju i diskusiju
- Različiti: jezik, fokus, interesovanja, razumijevanje, profesija



Arhitektura kao sredstvo komunikacije



Apstrakcija omogućava komunikaciju i diskusiju

Šta za vas predstavlja apstrakcija?

Uvijek mogu objasniti moj kod!

- Ne možete!
- Ljudi zaborave kako nešto funkcioniše
- Ljudi zaborave zašto baš tako treba da funkcioniše
- Zaborave svrhu implementacije
- Kod predstavlja jako ograničen pogled na sistem (previše detalja)
- Iskreno, nikoga nije briga za Vaš kod (uglavnom)
- Kupci razmišljaju o biznis mogućnostima koje će funkcionalnost softvera omogućiti i zaradi koja proizilazi iz toga

Apstrakcija

Apstrakcija nebitnih detalja	Fokus na bitnim osobinama sistema	Šta je bitno za arhitekturu?	Sve ono što može biti od koristi učesnicima u projektu
	• Fokus na osobinama bitnim za arhitekturu		• Šta nije bitno? Svaki detalj koji u datom kontekstu ne služi ničemu

Kako predstaviti apstrakciju?

Tekst

Dijagrami

Animacije

Vrijeme potrebno za:

- Kreiranje prve verzije dokumentacije
- Pronalaženje i razumijevanje potrebnih informacija
- Održavanje i promjena

Jedina gora stvar od nepostojanja reprezentacije ili dokumentacije jeste previše (zastarjele) dokumentacije

Jezik

Tehnički pojmovi

- Klase, funkcije, pokazivači, interfejsi, naslijeđivanje, itd.
- Testiranje, pokrivenost koda testiranjem

Biznis pojmovi

- Bruto, neto, profit, rabat, dionice, porez, učešće, početna uplata, rast, broj sati, itd.
- Dodatni profit
- Bonus

Domensko znanje

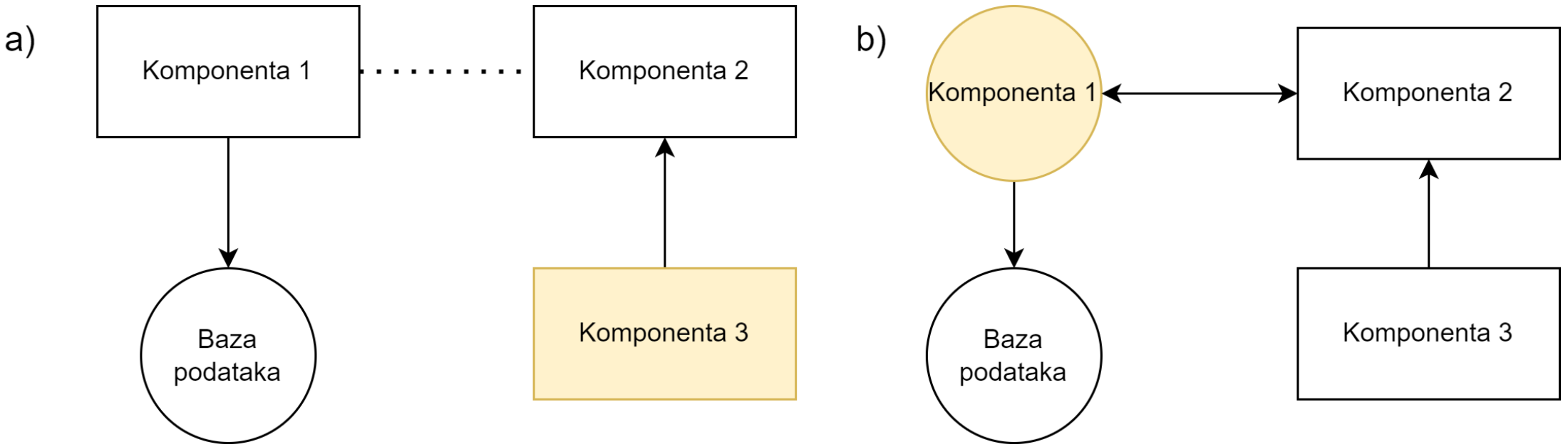
- Šarke, ventili, cijevi, traktori, kombajni, bušel, galoni, litre, stope
- Ruža, ljiljan, tulipan, orhideja, suncokret, kamilica, gerber, lavanda, iris, magnolija
- Baget, čabata, focaccia, pita, raženi hljeb, kiseli hljeb, hljeb od cijelog zrna, brioš, nan, čala

Profili za „modelovanje“

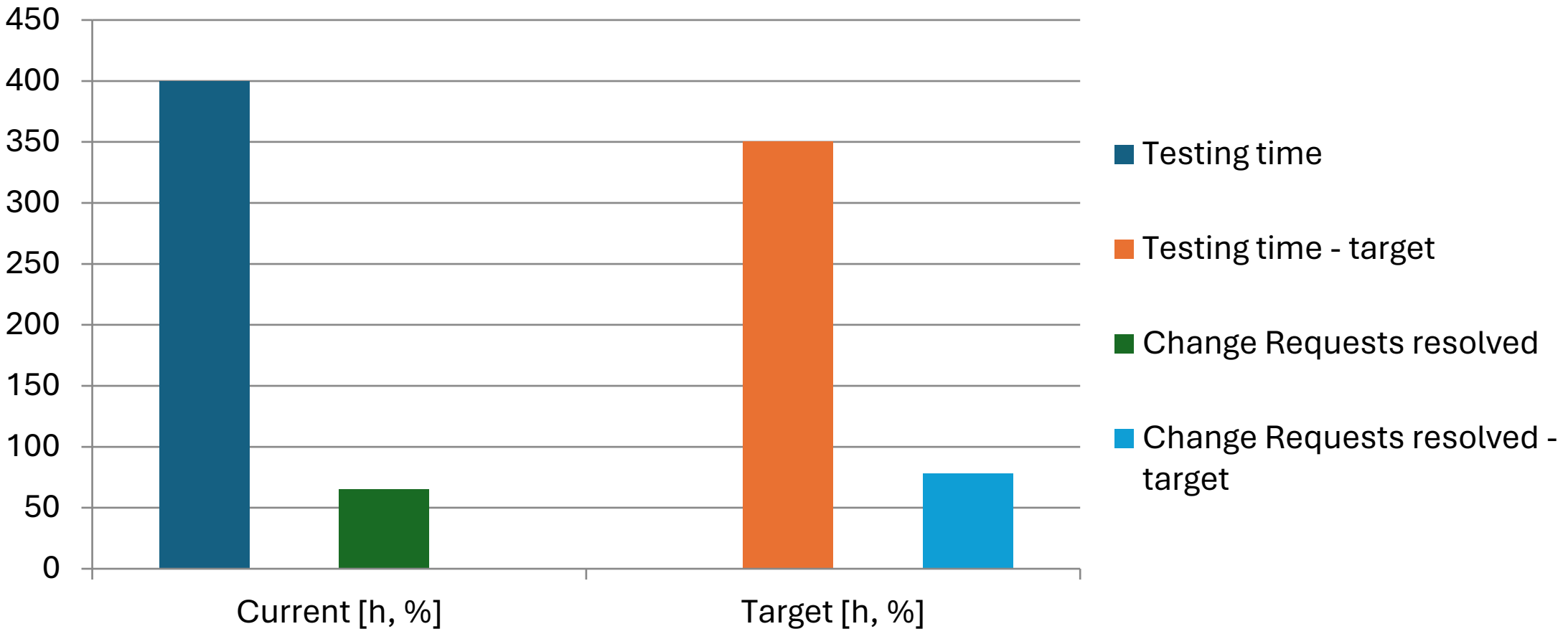
- Simboli
- Boje
- Strelice
- Fontovi

Imaju značenje!

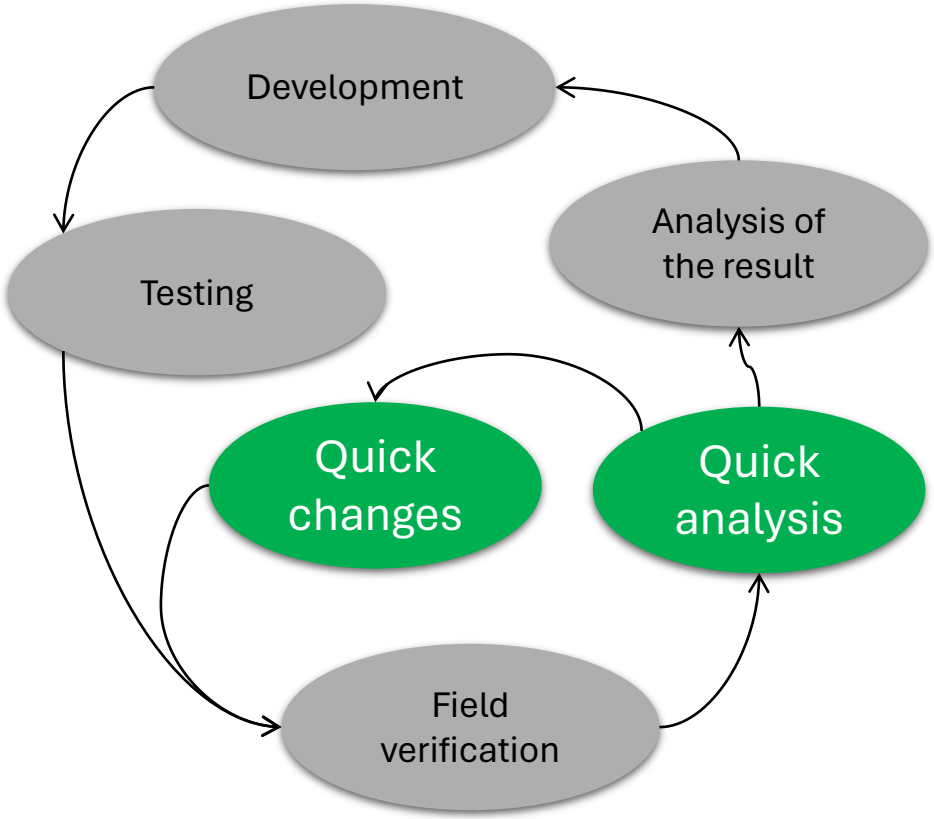
Koja je razlika između ove dvije slike?



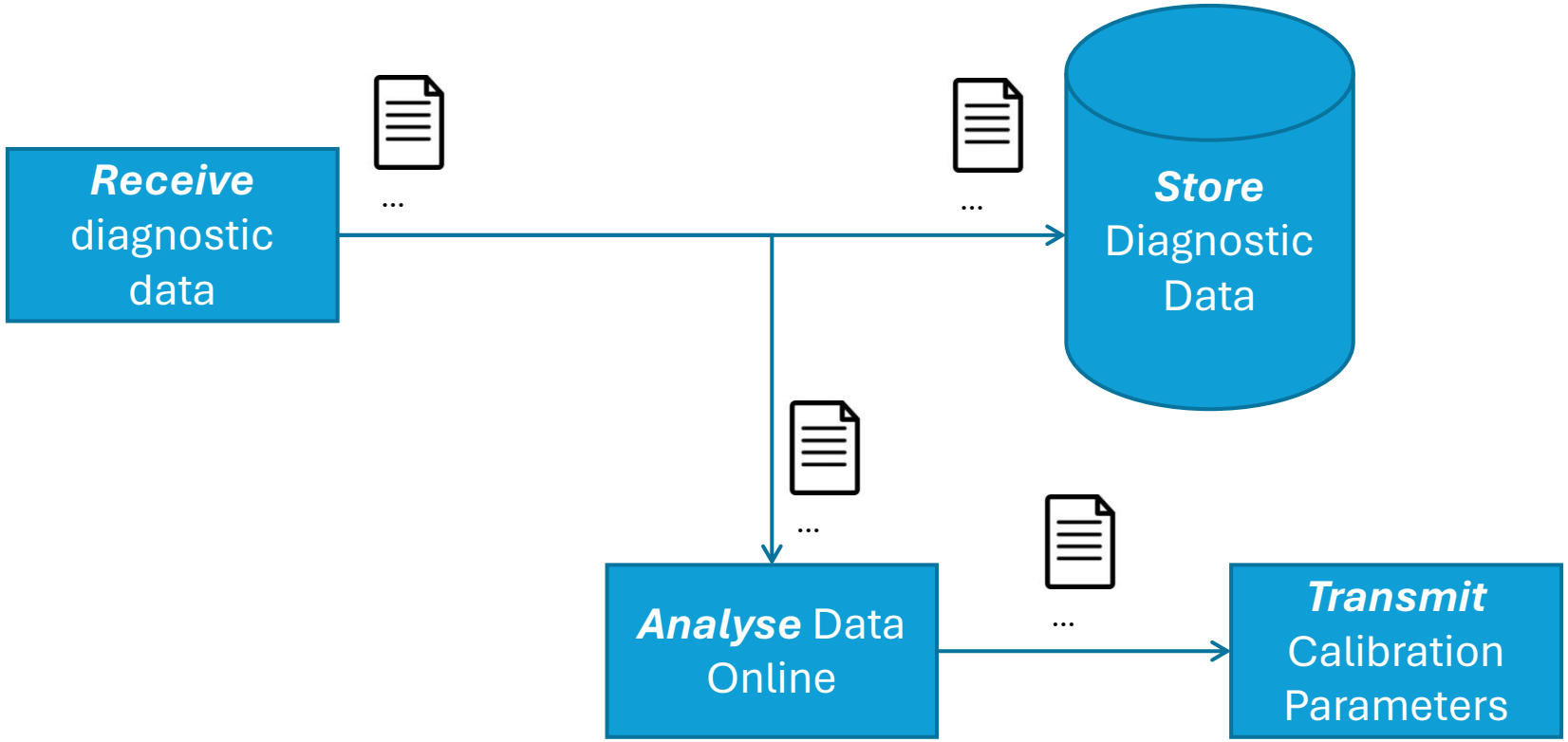
Za koga je namijenjena ova perspektiva?



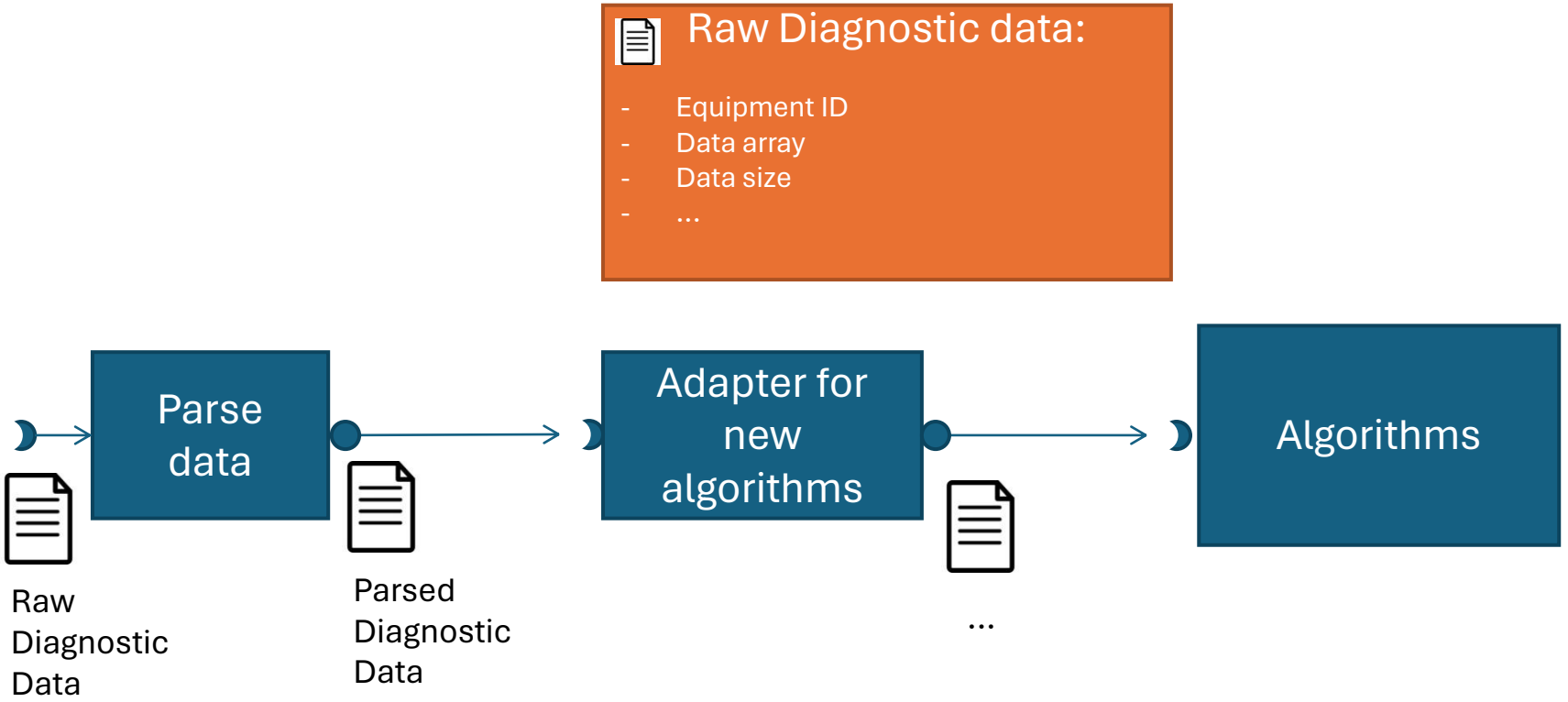
Za koga je namijenjena ova perspektiva?



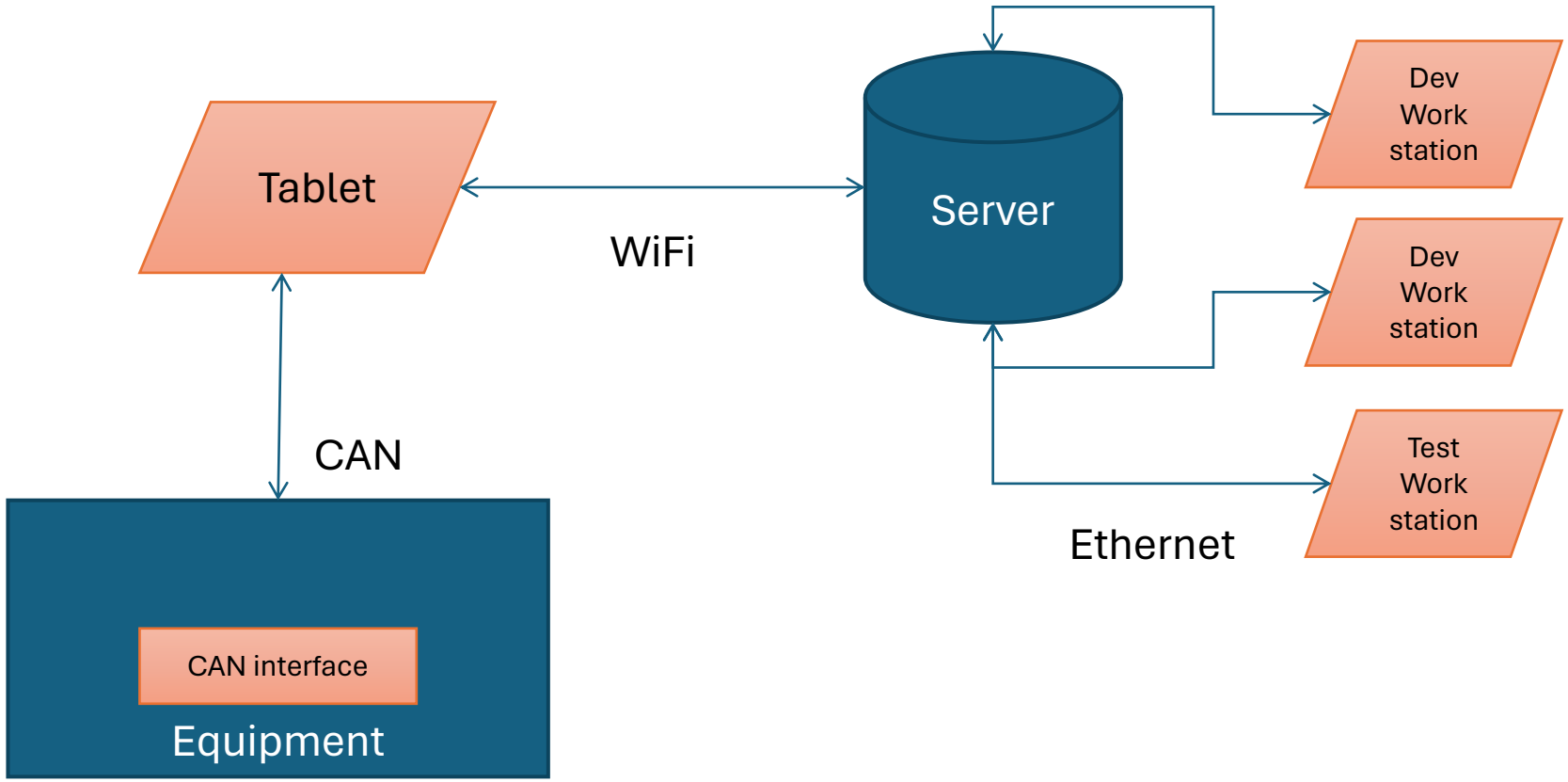
Za koga je namijenjena ova perspektiva?



Za koga je namijenjena ova perspektiva?



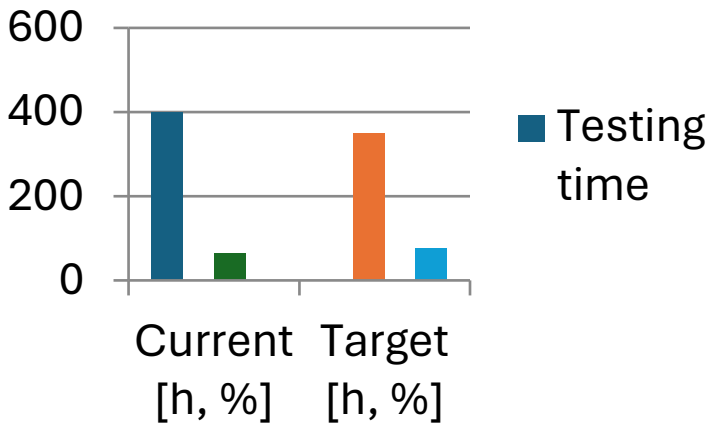
Za koga je namijenjena ova perspektiva?

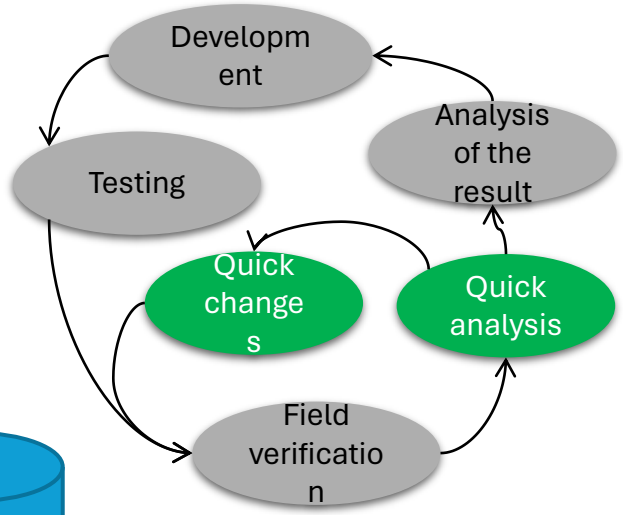


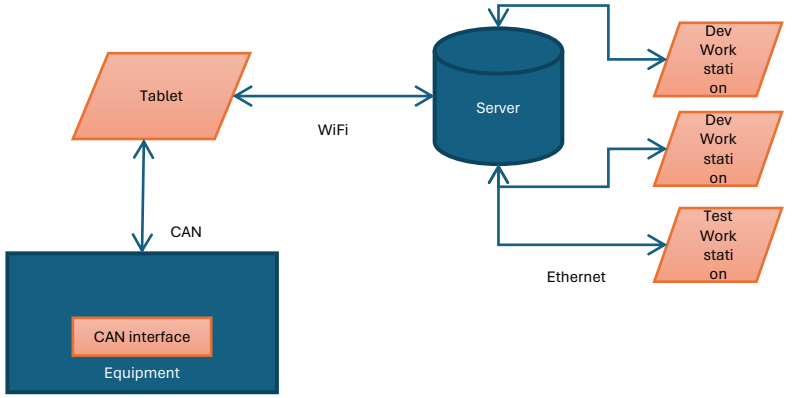
Svi moraju razumijeti sve detalje o sistemu

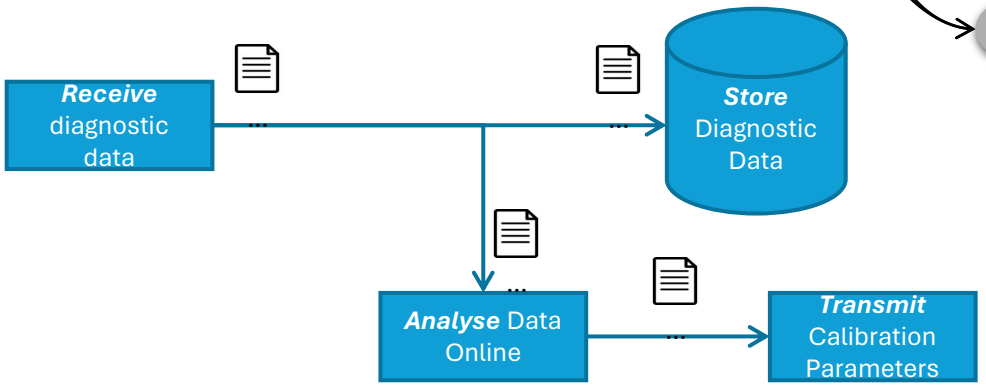
Ne baš...

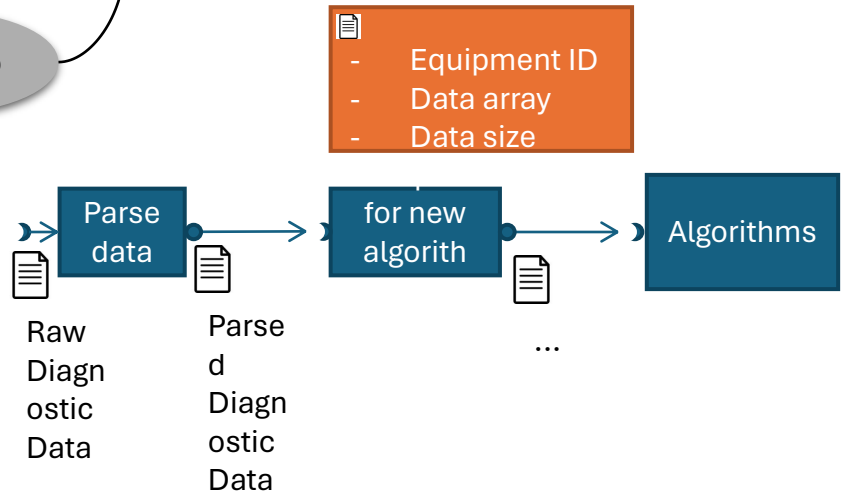
Ako sve dijagrame spojimo u jednu sliku...



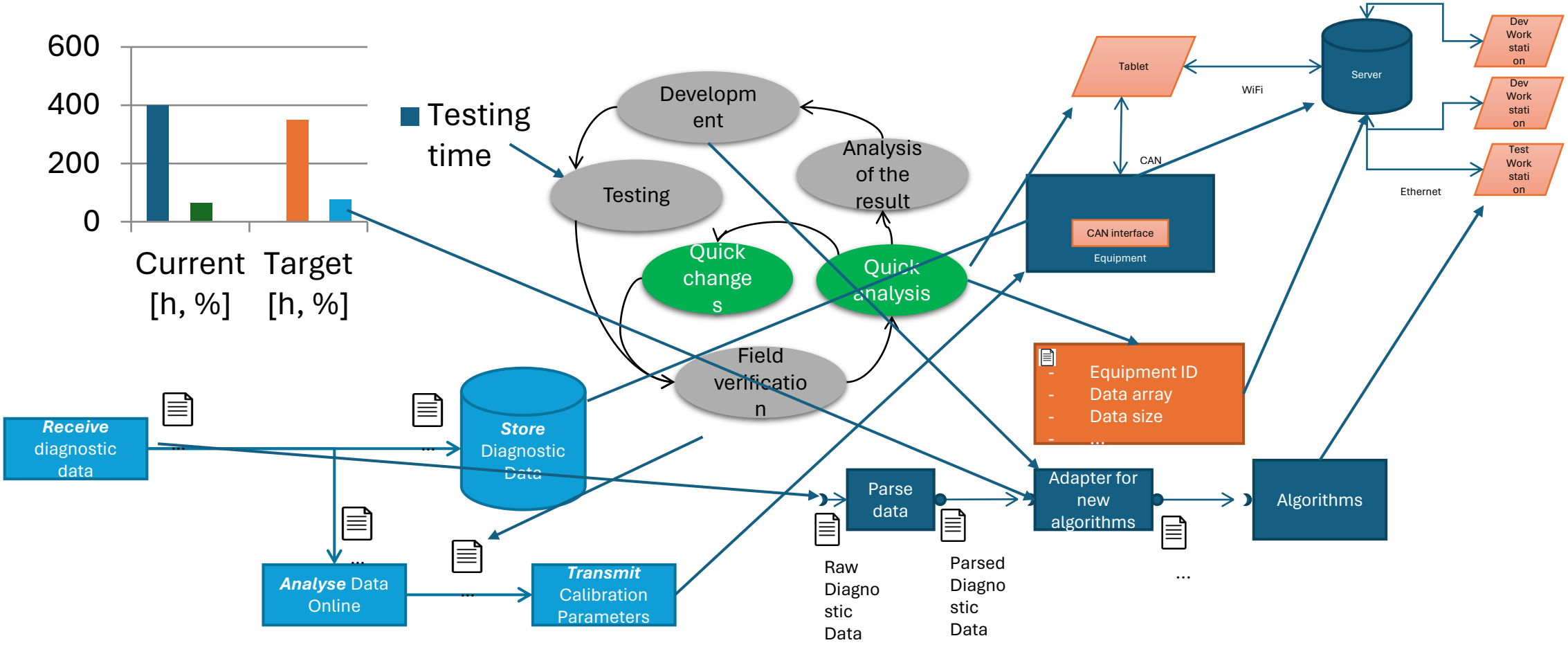




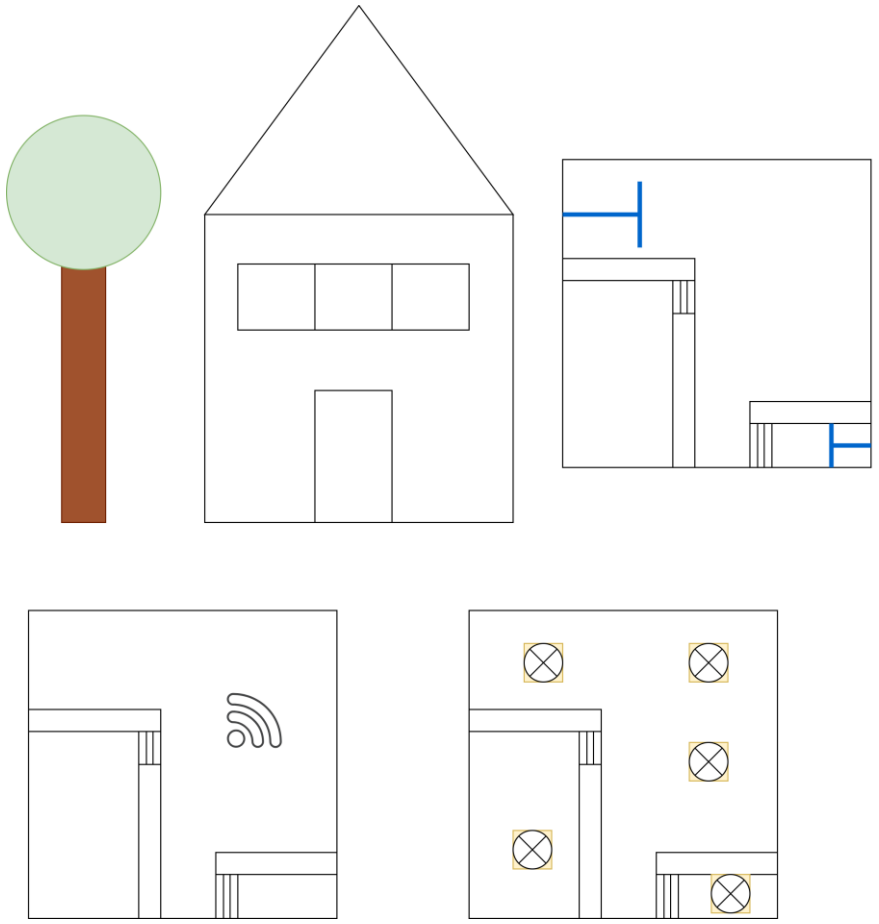




Ako ih povežemo...



Perspektive/pogledi na sistem (Architecture Views)



Perspektive/pogledi na sistem (Architecture Views)

Različite uloge trebaju različite poglede na sistem

Jasno definisani dijagrami

Svrha dijagrama

Šta je moguće naučiti iz dijagrama?

Šta je moguće učiniti sa tim znanjem?

PREDNOSTI RAZLIČITIH PERSPEKTIVA

Razdvajanje izazova

Zasebna analiza

Smanjenje kompleksnosti

Usmjeren fokus

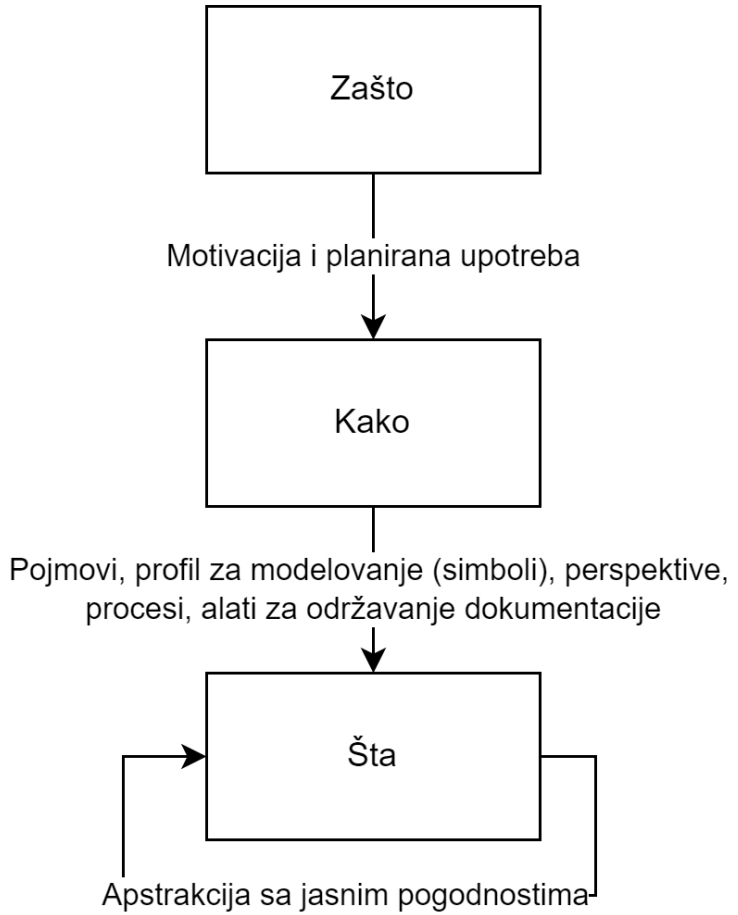
Rastavljanje sistema na manje dijelove

Funkcionalnost

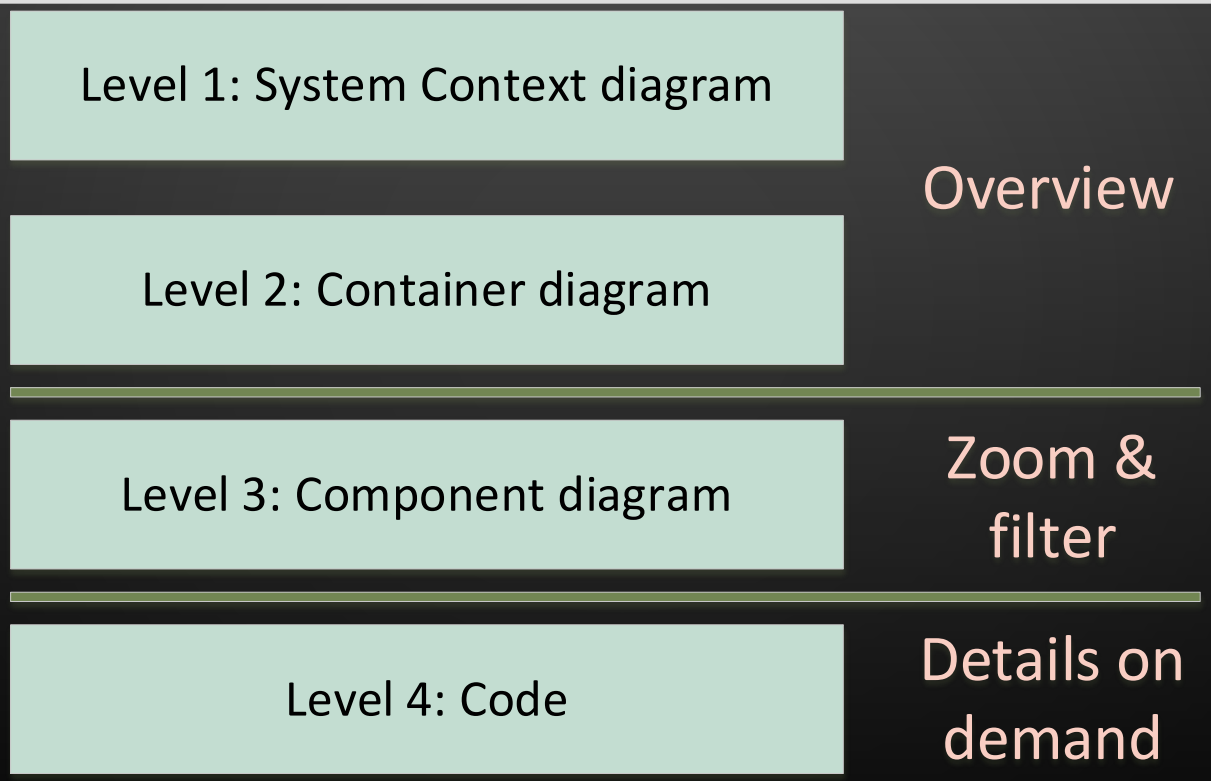
Kvalitet

Prilikom kreiranja bilo kakve apstrakcije

1. Onboarding novih članova tima
2. Prezentacija idejnog riješenja



C4 ARCHITECTURE VIEWS



https://c4model.com/

C4 ARCHITECTURE VIEWS

Level 1: System Context diagram

system as a box in the centre, surrounded by its users and the other systems that it interacts with.

Level 2: Container diagram

the high-level shape of the software architecture and how responsibilities are distributed across it (applications and data).

Level 3: Component diagram

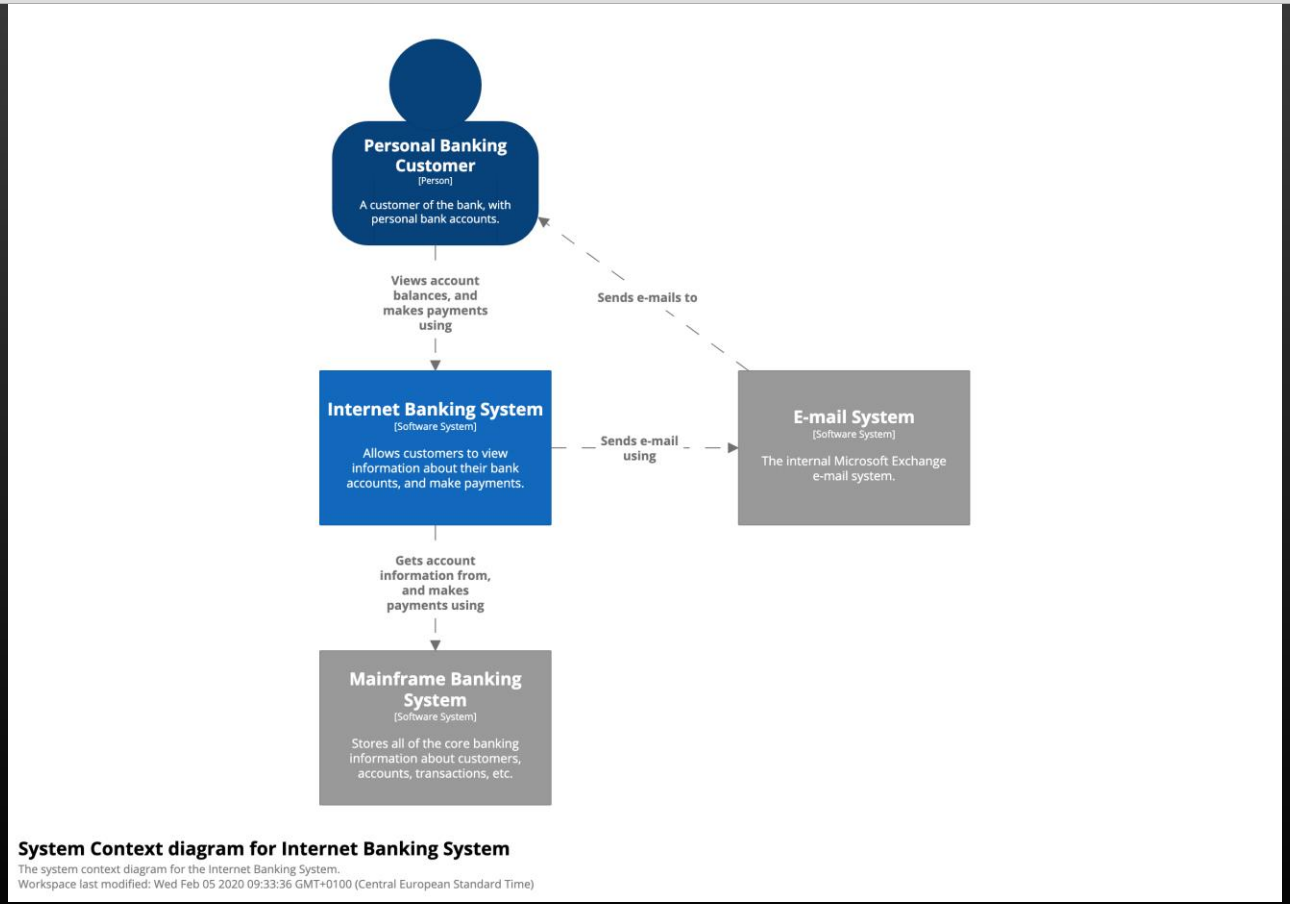
how a container is made up of a number of "components", what each of those components are, their responsibilities and the technology/implementation details.

Level 4: Code

UML class diagrams, entity relationship diagrams or similar.

https://c4model.com/

C4 ARCHITECTURE VIEWS - CONTEXT

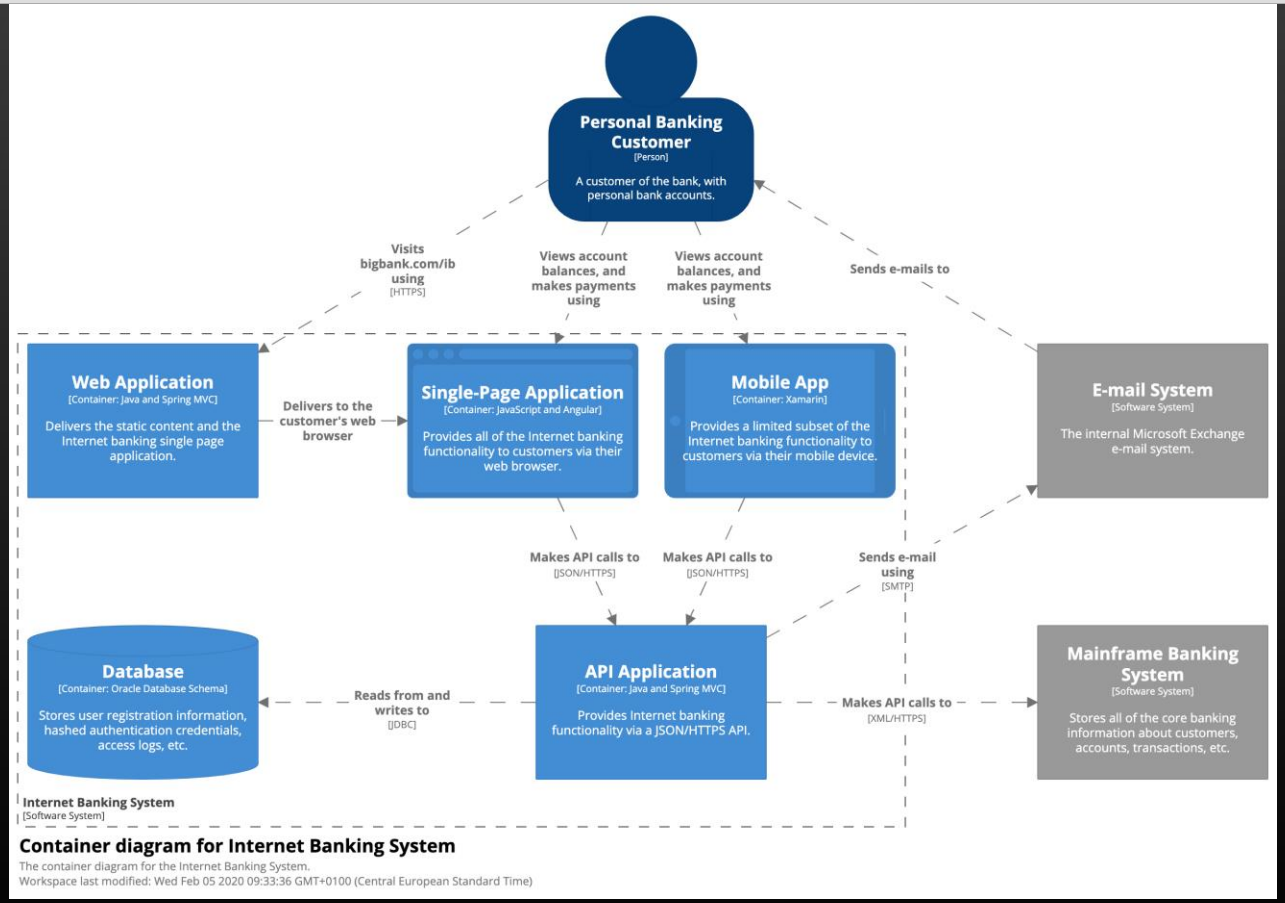


System Context diagram for Internet Banking System

The system context diagram for the Internet Banking System.
Workspace last modified: Wed Feb 05 2020 09:33:36 GMT+0100 (Central European Standard Time)

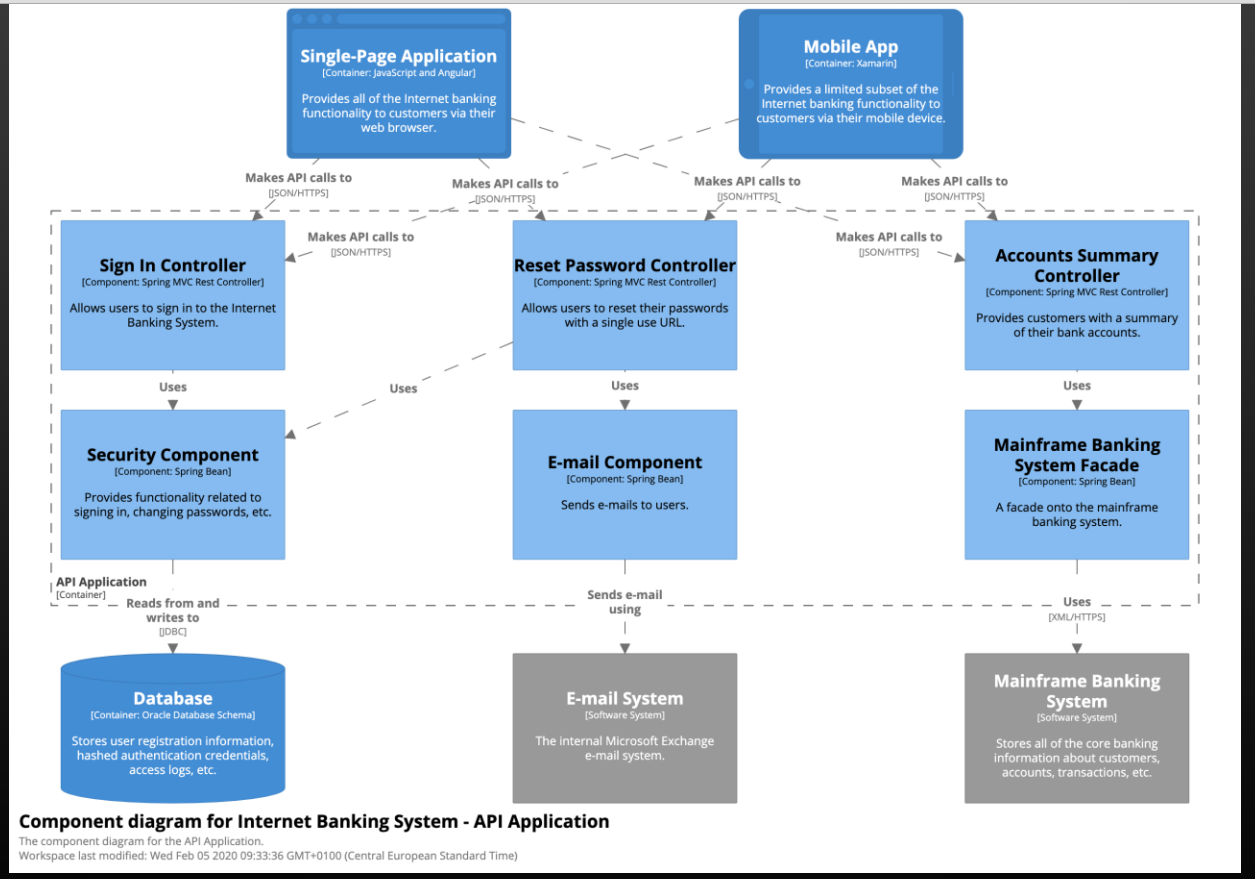
https://c4model.com/

C4 ARCHITECTURE VIEWS - CONTAINER



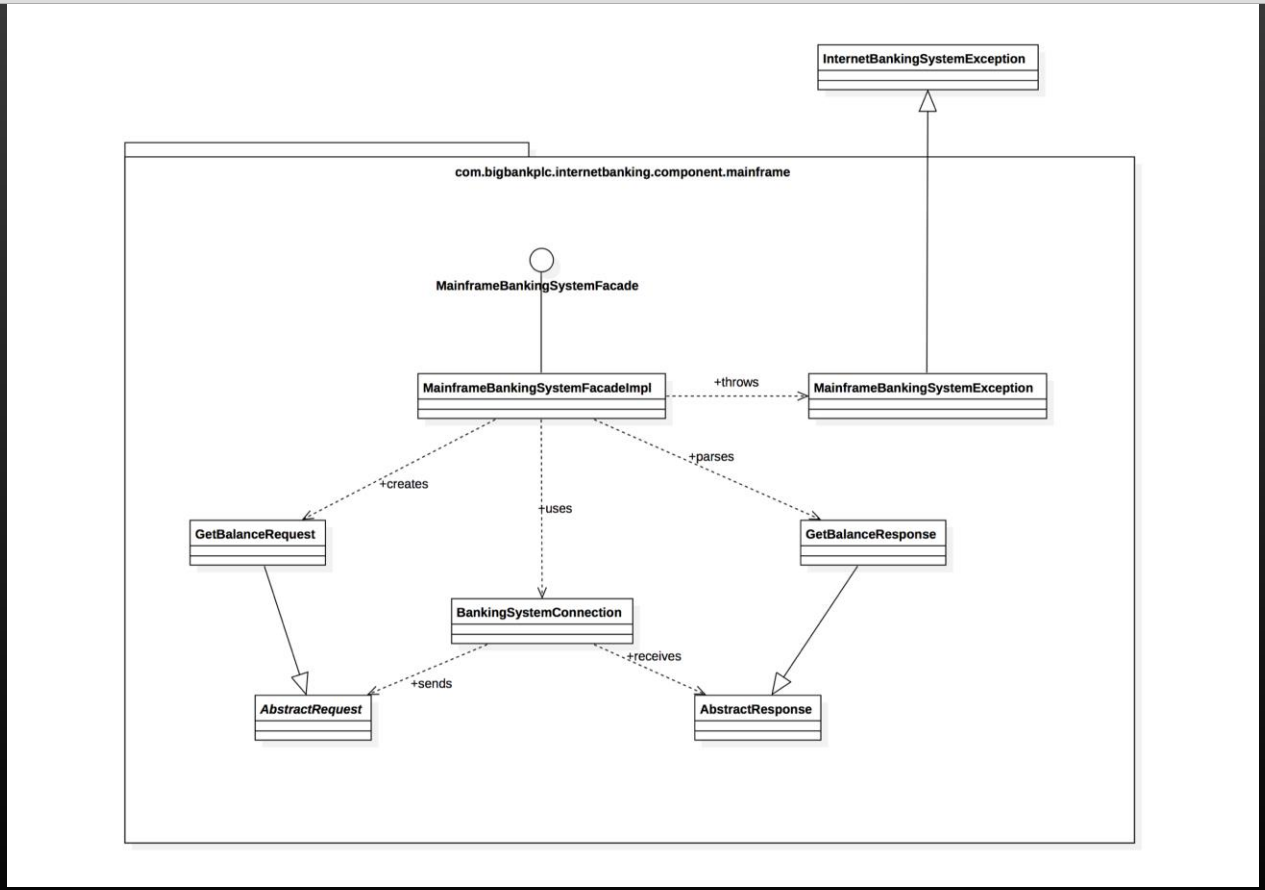
https://c4model.com/

C4 ARCHITECTURE VIEWS - COMPONENT



https://c4model.com/

C4 ARCHITECTURE VIEWS - CODE



https://c4model.com/

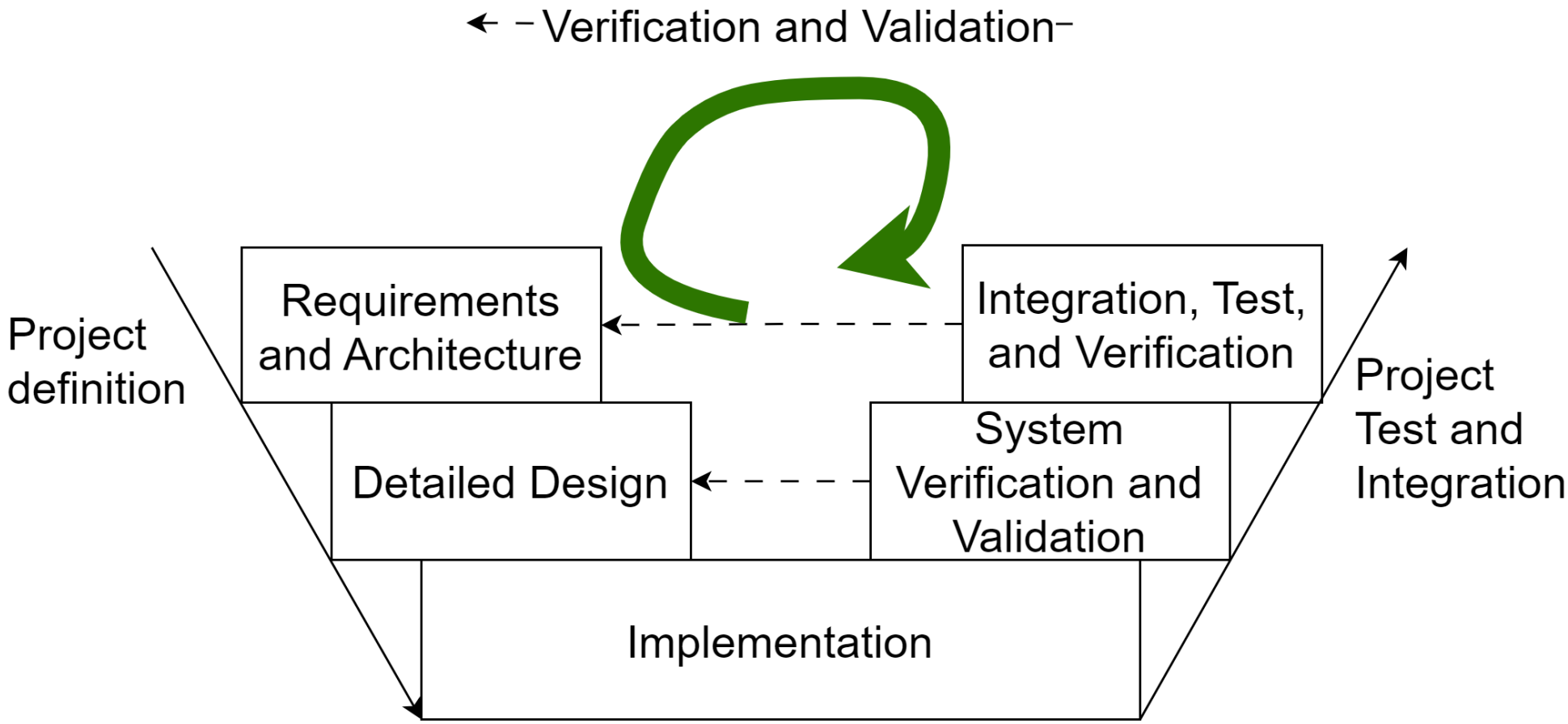
Vježba: Pokušajte skicirati vaš sistem

Vježba: Pokušajte skicirati vaš sistem

Pitajte osobu pored vas da komentariše vašu skicu. Da li shvataju šta ste željeli predstaviti?

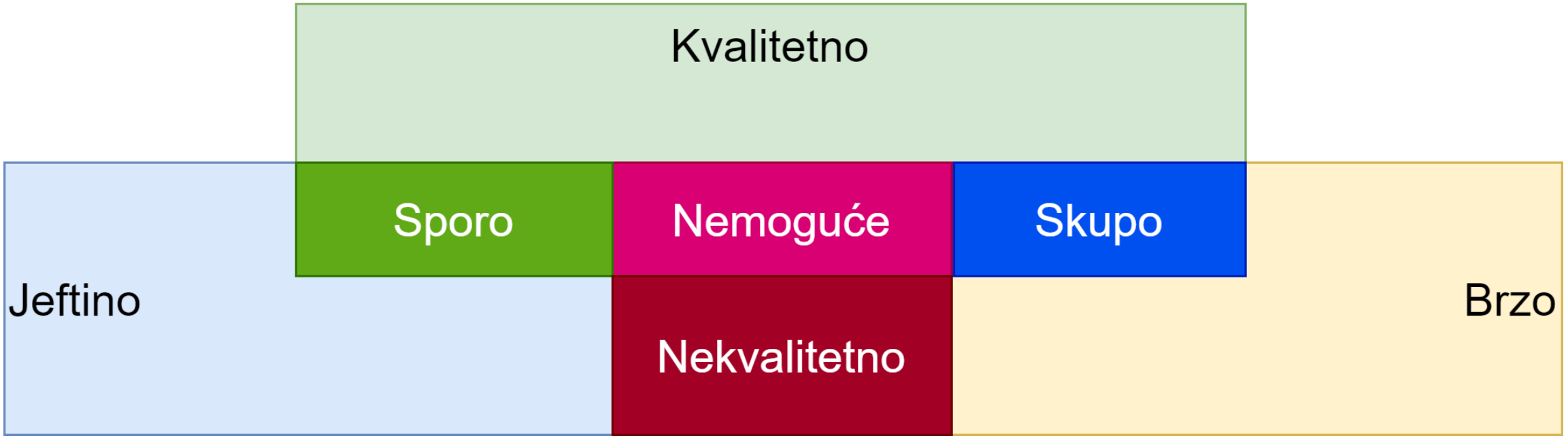

Pauza

Kako napraviti riješenje koje zadovoljava potrebe kupaca?



Kako napraviti riješenje koje zadovoljava potrebe kupaca?

- Da li riješenje uopšte radi ono što treba da radi?
- Da li iko razumije šta program radi i kome je to potrebno?
- Da li je riješenje dovoljno vrijedno, novo, standardno, inovativno?
- Da li je kupac navikao na slična riješenja?



Kada bi samo postojalo slično riješenje...

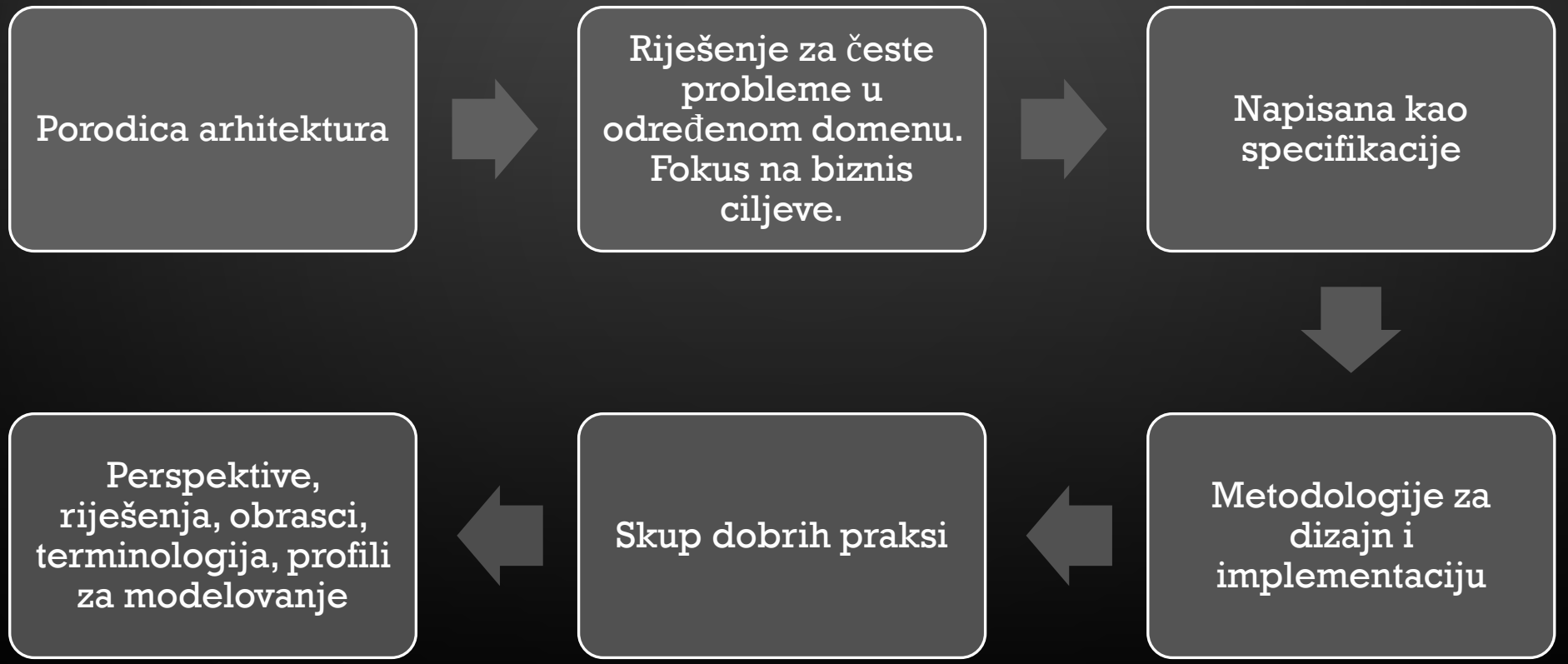
Zapravo, postoji i previše

Koristite postojeća znanja, često nema potrebe izmišljati „toplu vodu“

REFERENTNE ARHITEKTURE

- „Referentna arhitektura je porodica arhitektura koja sadrži upustva (razvoj, najbolje prakse) za kreiranje sistema, kroz riješavanje problema koji su karakteristični za tu porodicu softverskih sistema.“
- „Referentna arhitektura je struktura iz koje je moguće izvesti konkretnu arhitekturu u određenom domenu“; “*Reference Architecture and Product Line Architecture: A Subtle but Critical Difference”; Elisa Yumi Nakagawa, Pablo Oliveira Antonino, and Martin Becker*
- „Referentna arhitektura je u osnovi predefinisana arhitektura, dizajnirana za korištenje u određenom biznis i tehničkom kontekstu“; “The Rational Unified Process: An Introduction.”; Kruchten, P., 2000

REFERENTNE ARHITEKTURE



REFERENTNE ARHITEKTURE

- Znanje o određenoj oblasti
- Kako dizajnirati arhitekturu sistema u određenoj oblasti
- Pravila za biznis, stilovi, obrasci
- Dobre prakse
- Softverski elementi pogodni za određenu oblast
- Terminologija

"Reference Architecture and Product Line Architecture: A Subtle but Critical Difference"; E.Y. Nakagawa, et al. ECSA 2011

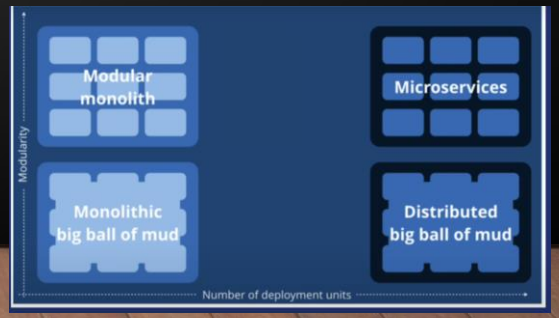
PRIMJERI REFERENTNIH ARHITEKTURA

- Shopify: An e-commerce reference architecture.
- BIAN: A banking industry architecture reference model.
- ACORD: Reference architecture for the insurance industry.
- eTOM: A business process framework for the telecommunications industry.
- FEAF: The federal enterprise architecture framework.

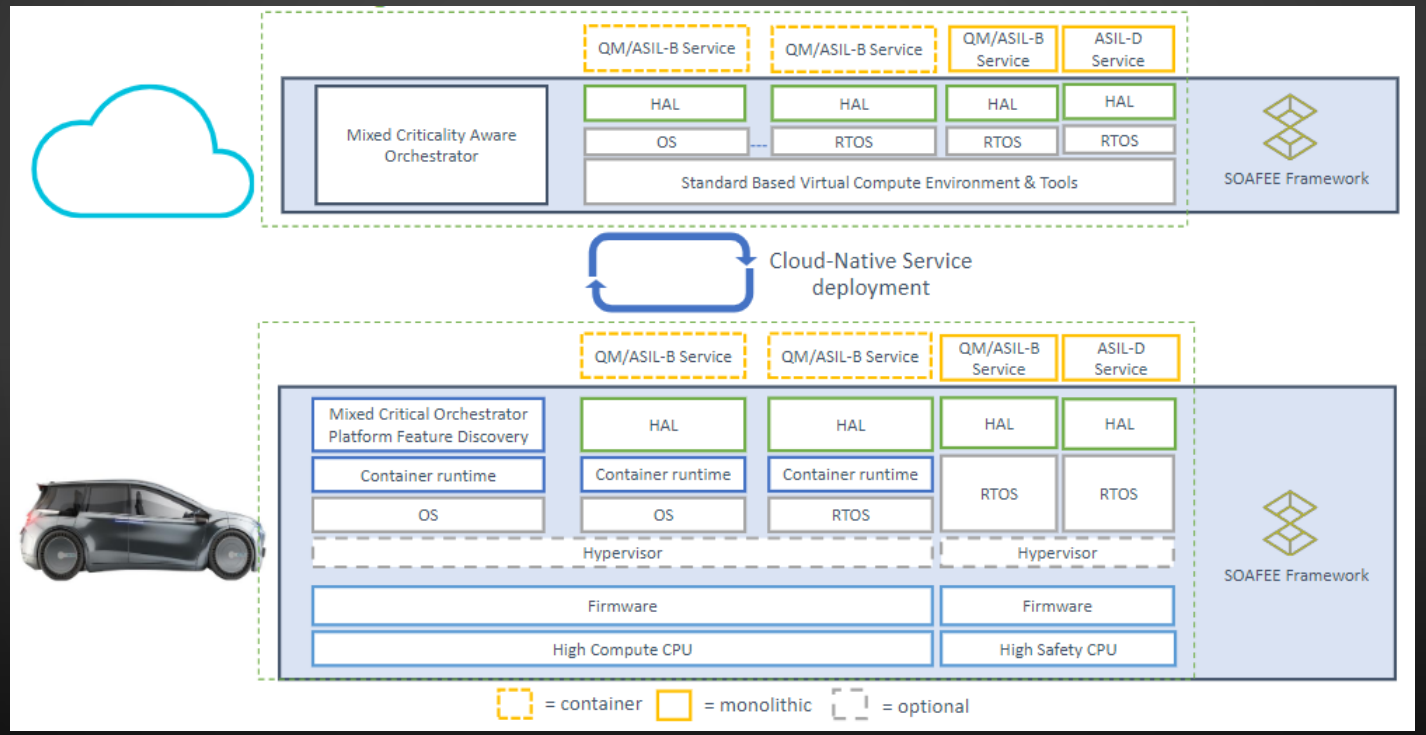
- AUTOSAR: An automotive open system architecture reference architecture.
- SOAFEE: Scalable open architecture for embedded edge.
- Siemens reference architecture for Condition Monitoring Systems
- NASA crew exploration vehicle reference architecture.
- RACS: The reference architecture for component-based simulation.

SHOPIFY

- youtube.com/watch?v=MV5Kdwzwcag
- youtube.com/watch?v=gyZDWsLk_Ac
- ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8491621
- ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10128209
- shopify.engineering/shopify-monolith
- shopify.engineering/deconstructing-monolith-designing-software-maximizes-developer-productivity
- youtube.com/watch?v=N8NWDHgWA28



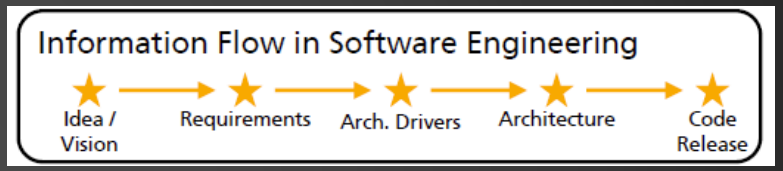
SOAFEE



https://www.soafee.io/

POKRETAČI ARHITEKTURE (ARCHITECTURAL DRIVERS)

- Biznis ciljevi
  - Finansijski ciljevi
  - Vizija
- Funkcionalni pokretači
- Ograničenja
- Ne-funkcionalni (kvalitativni) pokretači
  - Performance, Maintainability, Expendability, Safe, Secure, Accessibility, Deployability, Reliability, Scalability, … ISO/IEC 9126.
  - **Kvalitet uvijek ima kontekst!**



VRSTE POKRETAČA PROJEKTA

Inovativni

Neinovativni (infrastrukturalni)

Primjer: Konkurencija za ebay. Šta treba sadržavati? Šta je inovativno?

POKRETAČI ARHITEKTURE

Biznis	Funkcionalnost	Ograničenja	Kvalitet
Natural language **Links to documents** Increase sale for 15%. Increase a reputation. A unique functionality.	**Use Cases** **User Stories / Epics** **Template scenario** User registration. Web shop.	**Natural language** Use open source. Use Android. Do not use QR codes.	**Template scenario** Performance, Maintainability, Extendibility, Safety, Security, Accessibility, Deplorability, Reliability, Scalability

PREDLOŽAK SA SPECIFIKACIJU I KVANTIFIKACIJU KVALITETA

ID	*Unique identifier*	**Status**	*[Open, Defined, Solved, …]*
Name	*Name of scenario*	**Owner**	*Responsible for the scenario*
Quality	*Related quality attribute: exactly one attribute should be chosen.*	**Stakeholders**	*Stakeholders involved*
Environment	*Context applying to this scenario. May describe both context and status of the system.*		
Stimulus	*The event or condition arising from this scenario.*		
Response	*The expected reaction of the system to the scenario event.*		
Response measure	*The measurable effects showing whether the scenario is fulfilled by the architecture.*		

PREDLOŽAK SA SPECIFIKACIJU I KVANTIFIKACIJU KVALITETA

ID	Unique identifier	**Status**	[Open, Defined, Solved, …]
Name	Name of scenario	**Owner**	Responsible for the scenario
Quality	Related quality attribute: exactly one attribute should be chosen.	**Stakeholders**	Stakeholders involved
		Quantification	
Environment	Context applying to this scenario. May describe both context and status of the system.		
Stimulus	The event or condition arising from this scenario.		
Response	The expected reaction of the system to the scenario event.		

PRIMJER: PERFORMANSE

ID	#P_01	**Status**	Elicited
Name	Responsive under high load	**Owner**	Architect
Quality	Performance	**Stakeholders**	Users
Environment	The system is running under high system load, caused by background processing of computation-intensive operations of the application.		
Stimulus	A user is working in the GUI, entering data in forms.		
Response	The system is still responsive and lets the user work in the used workflow.		
Response measure	System response time is below 0.2s on user actions.		

VAŠ PRIMJER?

ID	----------	**Status**	----------
Name		**Owner**	----------
Quality		**Stakeholders**	----------
Environment			
Stimulus			
Response			
Response measure			

PARAMETRI KVALITETA (…ILITIES)

- Performance
- Responsiveness
- Safety
- Security

- Deployability
- Interoperability
- Maintainability
- Modifialibity
- Parallel development
- Portability
- Reusability
- Scalability
- Testability
- Availability

KOJE OSOBINE KVALITETA SU BITNE ZA KUPCA

Ne-IT (dobar softver, lijep, brz, siguran, koristan, itd.)	Orijentisani ka biznisu i digitalizaciji (https://janbosch.com/blog/) Eksterne osobine kvaliteta
IT (testiran, moguće brzo nadograditi i izmijeniti, skalira se jednostavno, lako čitanje koda, itd.)	Tehnički kvalitet, plan je preuzimanje projektnog koga u budućnosti i nastavak razvoja Interne osobine kvaliteta

PRIORITY OF REQUIREMENTS IN SOFTWARE PROJECTS FROM COMMISSIONERS' POINT OF VIEW, LUCERNE SCHOOL OF BUSINESS, SWITZERLAND

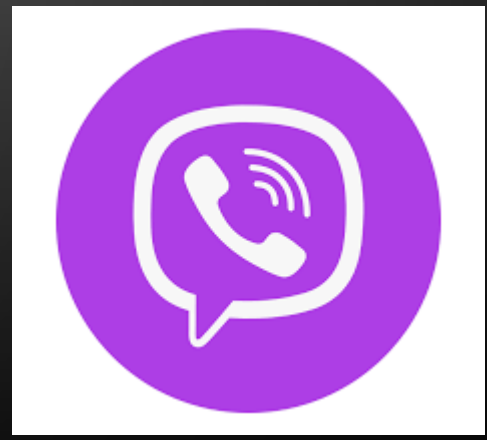
Kupci očekuju od inženjera da znaju njihove zahtjeve za softver (Customers expect from engineers to understand what they want without telling them because "engineers are the experts,,).

Domensko znanje je jako bitno (finansije, turizam, itd.)

Industry	No. of employees	Project setup	Interviewee background	Setup for software development
Finance	40	New applications	Business background	In-house
Carpentry	70	Extension of existing industry software	Business background	Outsourced with IT intermediator
Food Packaging	2	New website	Business background	Outsourced
Tourism	50	Improvement of existing website	Business background with basic technical knowledge	Outsourced, with design agency as 3rd party
Wine Trade	100	Digital transformation	Business background	Outsourced, with communication agency as 3rd party
Insurance	13000	Extension of existing software	Business background, IT intermediator in the interview	In-house, with internal IT intermediator

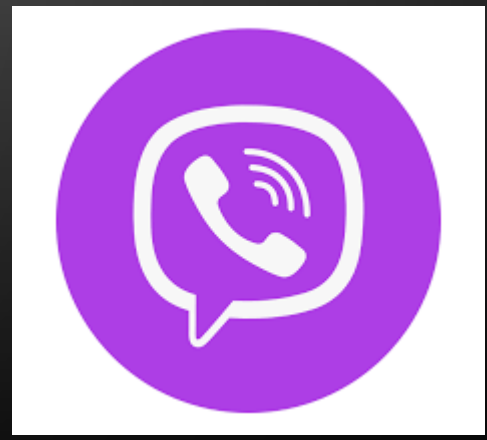
KOJU APLIKACIJU KORISTITE?

- Zašto?
- Po čemu se ove aplikacije razlikuju?



KOJU APLIKACIJU KORISTITE?

- Razlika nije u funkcionalnosti, već u kvalitetu, da li se slažete?



Vježba: Koje osobine kvaliteta su najvažnije u vašim projektima?

Vježba: Koje osobine kvaliteta su najvažnije u vašim projektima?

Da li su te osobine važne vama ili vašim kupcima?


Pauza

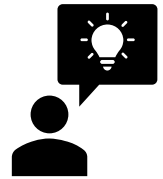
Primjena – Kako efikasnije proizvoditi kvalitetniji (i monetarno vrijedniji) softver?

- Razumijete potrebe za sistemom
  - Razumijete kvalitativne osobine softvera koje su važne kupcima
- Govorite njihovim jezikom (domensko znanje, tipični izrazi)
- Vaše riješenje riješava njihov problem
- U mogućnosti ste da ponudite više (dokumentacija, prezentacija, kvantifikacija testiranja, najbolje prakse iz industrije)
- Možete jasno objasniti vaše riješenje i kako ono riješava problem koji kupci imaju

Učesnici u projektu – želje i brige

Inženjeri

Marketing



Investitori

Biznis

Korisnici

Korisnici

POKRETAČI ARHITEKTURE

Biznis	Funkcionalnost	Ograničenja	Kvalitet
Natural language **Links to documents** Increase sale for 15%. Increase a reputation. A unique functionality.	**Use Cases** **User Stories / Epics** **Template scenario** User registration. Web shop.	**Natural language** Use open source. Use Android. Do not use QR codes.	**Template scenario** Performance, Maintainability, Extendibility, Safety, Security, Accessibility, Deplorability, Reliability, Scalability

Jezik kupaca

Softverska riješenja za biznis zahtjeve

Uglavnom niko ne mari za vaš kod

Kupci žele riješenja, ne kod

Kupci žele nešto što riješava njihove probleme

Vrijednost vašeg proizvoda

- Niste jeftini
- Nudite izvrstan kvalitet i znate o čemu govorite (domenski riječnik)
- Kvalitet na prvom mjestu (prilagođavate priču o kvalitetu u zavisnosti od vrste klijenta)
- Nudite dodatnu vrijednost (extras: dokumentacija za korištenje, dokumentacija za nove članove tima, kvantifikacija pokrivenosti koda testovima, itd.)
- Između ostalog, koristite najrelevantnije tehnologije u datom domenu

Kako vaše riješenja rješava problem?

Non-IT

- Ne interesuje ih riješenje
- Brinu o biznisu koji treba profitirati usvajanjem vašeg riješenja

IT

- Zainteresovani za tehničke detalje
- Zainteresovani za kvalitet (jer planiraju integirati i održavati vaš softver)

Vrijednost softvera

Vi ne prodajete funkcionalan softver

Vi nudite riješenje za poslovne probleme (npr. primjer digitalizacije)

Koje su to kvalitativne osobine softvera najbitnije vašim kupcima?

Šta je to dodatno što nudite?

- Standardi?
- Slijedite referentnu arhitekturu?
- Poznavanje industrijskog domena

Primjer: Auto softver, MC/DC code coverage, ISO 26262, ASIL-B

Jasna komunikacija je bitna za sve

- Kupci su sretni jer ih razumijete
- Vi (kao arhitekta) razumijete šta se očekuje od vas
- Vi (kao arhitekta) možete kreirati odgovarajuća riješenja
- Vi (kao arhitekta) možete jasno komunicirati sa inženjerima

Glavna poruka

Shvatite šta je važno vašim kupcima (kvalitativni zahtjevi)

Ubijedite ih da ste vi eksperti u datom domenu

Koje izazove ste odabrali na početku? Da li neke od tehnika koje smo diskutovali riješavaju vaše probleme?

Kontakt, ideje, suradnja

- jasmin.jahic@gmail.com