Razvoj informacijskih sistemov

Informatika in podatkovne tehnologije

UNI, 2. letnik

Študijsko leto 2025 / 2026

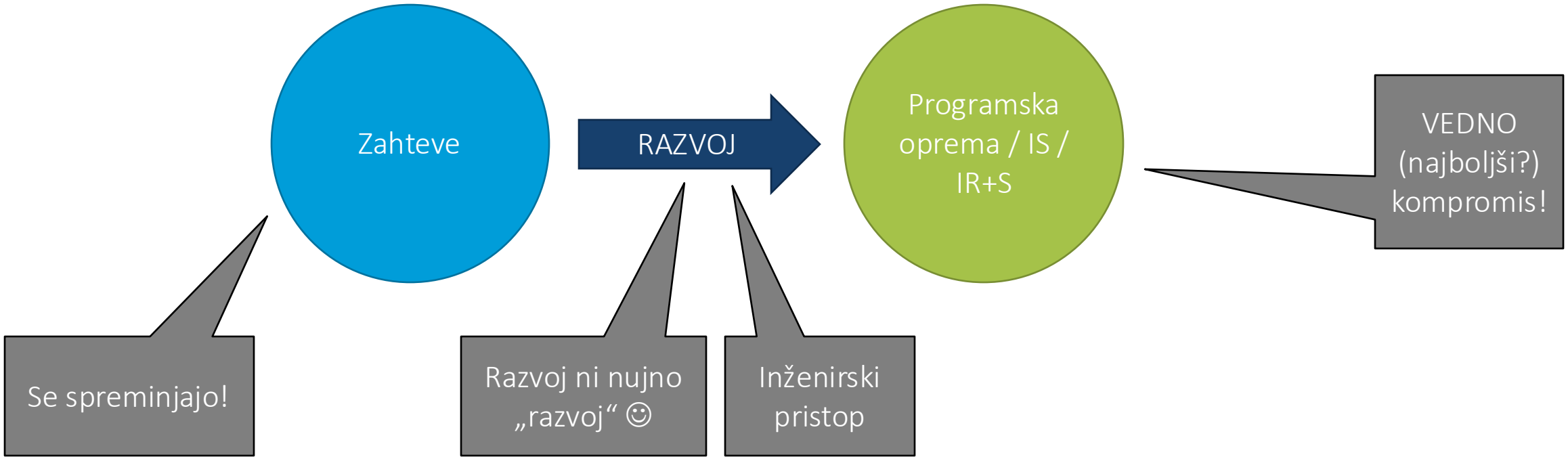
Luka Pavlič, UM, FERI

Zajem uporabniških zahtev

7. in 14. 11. 2025

- Vizija
- Besednjak
- Funkcionalne in nefunkcionale zahteve
- Standardni zapisi vizije in besednjaka
- UML na primeru UC, activity d., package d.
- Ostali načini zapisa zahtev: user stories

Razvoj informacijskih sistemov!



Pobrskajmo po spominu…

Zbiranje in analiza zahtev

Že poznamo!

Vrste zahtev?
- Funkcionalne / nefunkcionalne
- …

Zapis zahtev?
- Vizija
- Besednjak
- Primeri uporabe
- …

Standardizirano?
- UML UC?
- Predloge iz RUP?

Zakaj že...?

Causes of failed software projects [Standish Group]

Incomplete requirements	13.1%
Lack of user involvement	12.4%
Lack of resources	10.6%
Unrealistic expectations	9.9%
Lack of executive support	9.3%
Changing requirements & specifications	8.8%
Lack of planning	8.1%
System no longer needed	7.5%

Najbolj pogosta napaka je, da zgradimo napačen sistem.

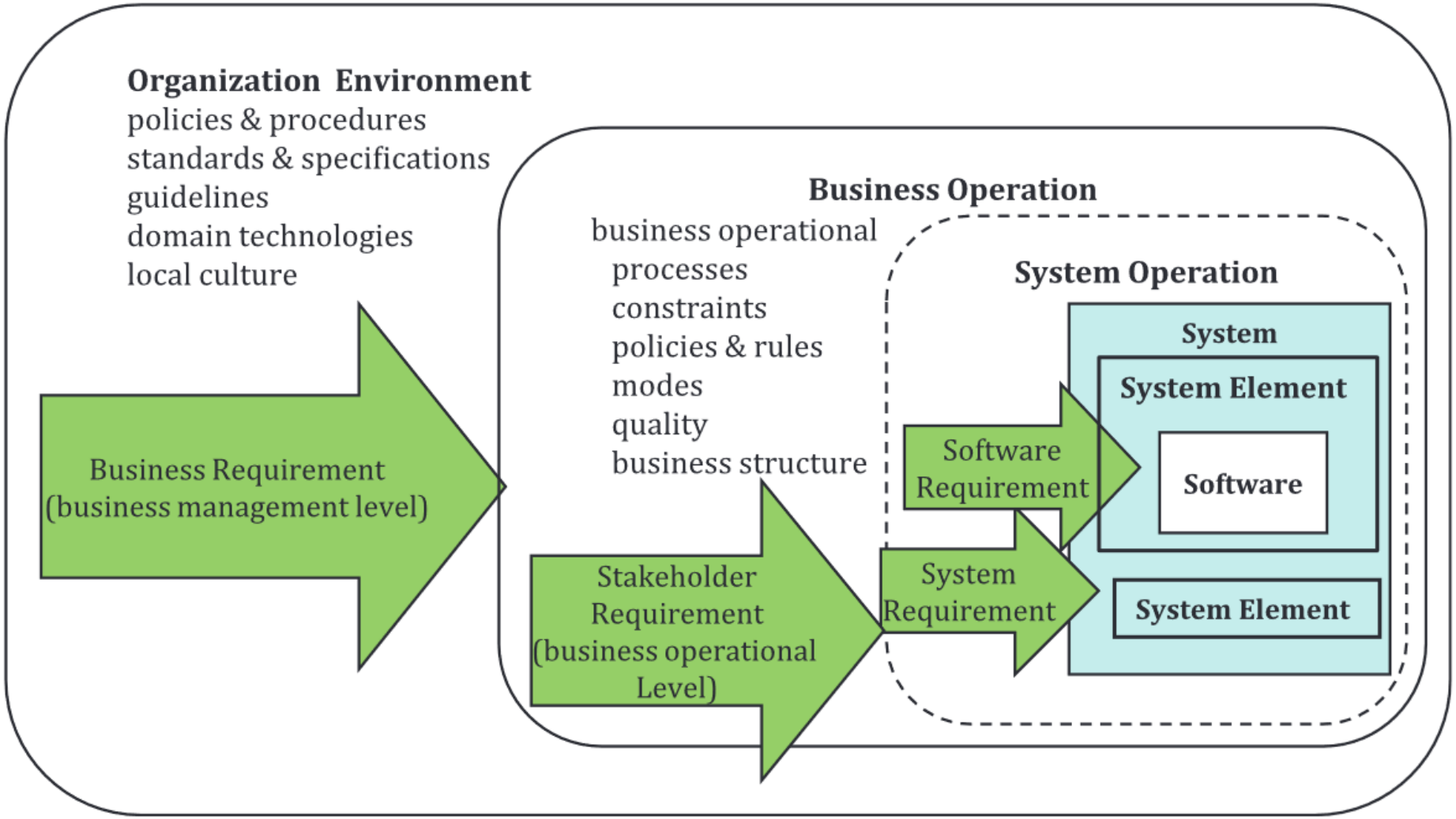
Real-world software projects tend to suffer from two primary requirements-related problems:

1. incompleteness: stakeholder requirements, and necessary detail, exist that are not revealed and communicated to the software engineers;
2. ambiguity: requirements are communicated in a way that is open to multiple interpretations, with only one possible interpretation being correct.

SWEBOK®

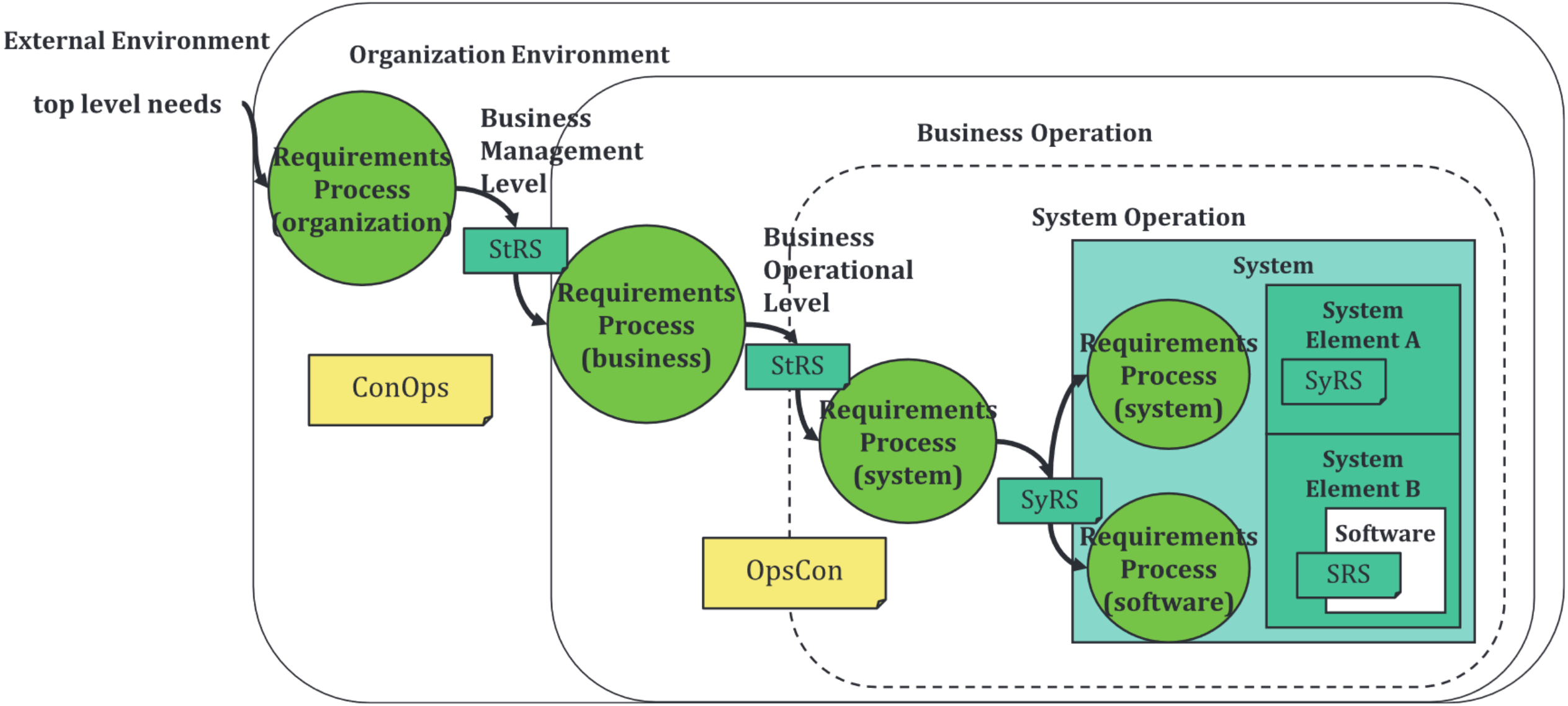
Vloga zahtev

External Environment
market trends
laws & regulations
legal liabilities
social responsibilities
technology base
labor pool
competing products
standards & specifications
public culture
Physical/natural environment

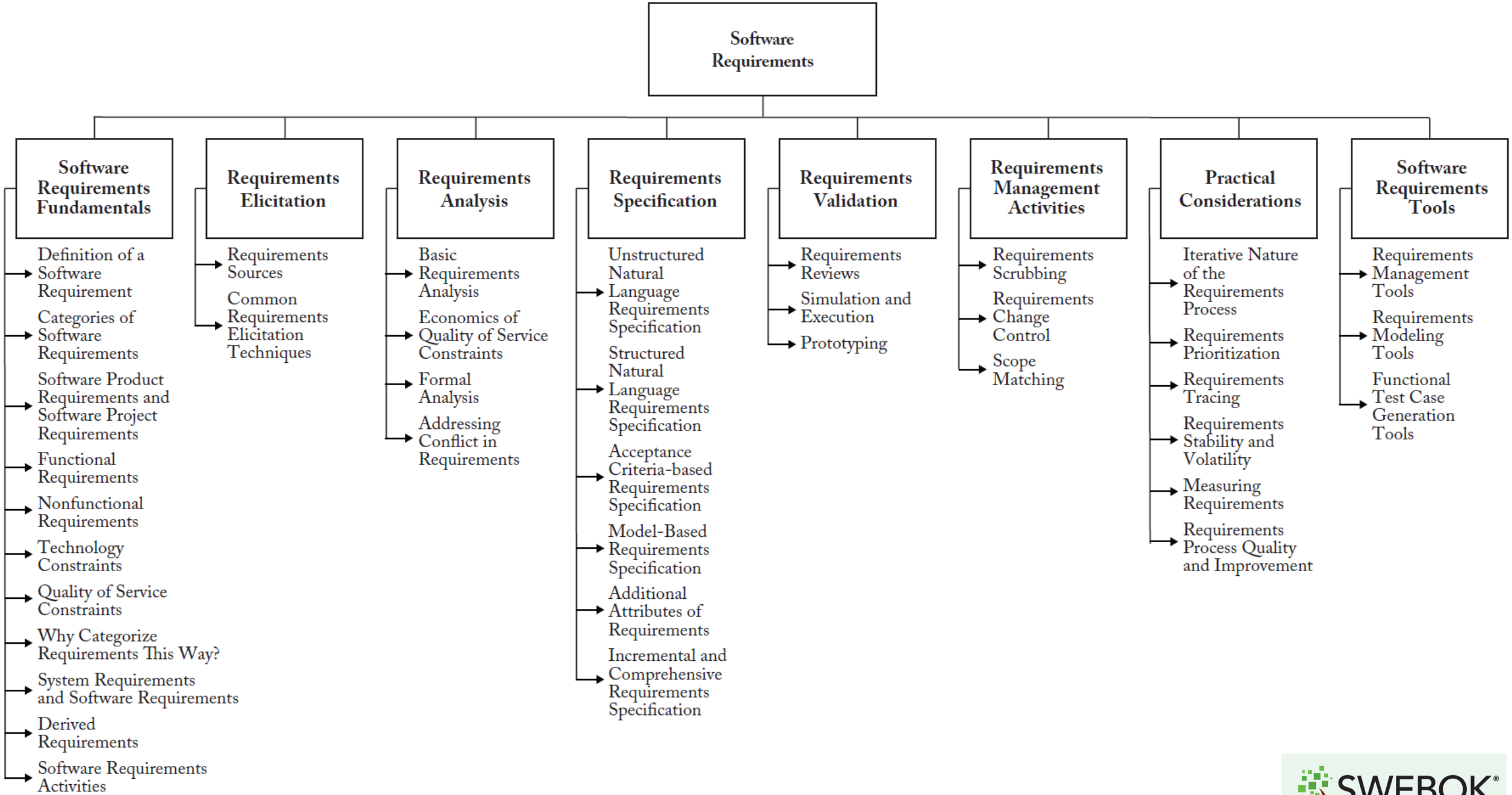


ISO/IEC/IEEE 29148:2018 - Systems and software engineering — Life cycle processes — Requirements engineering

Vrste specifikacij

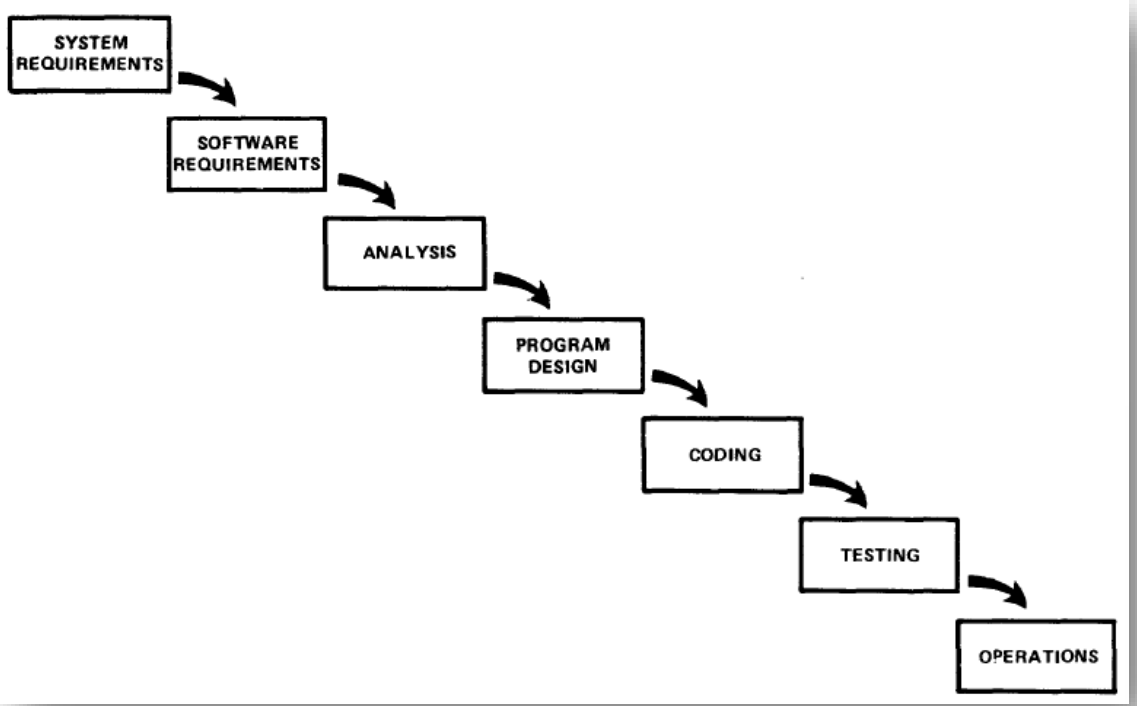


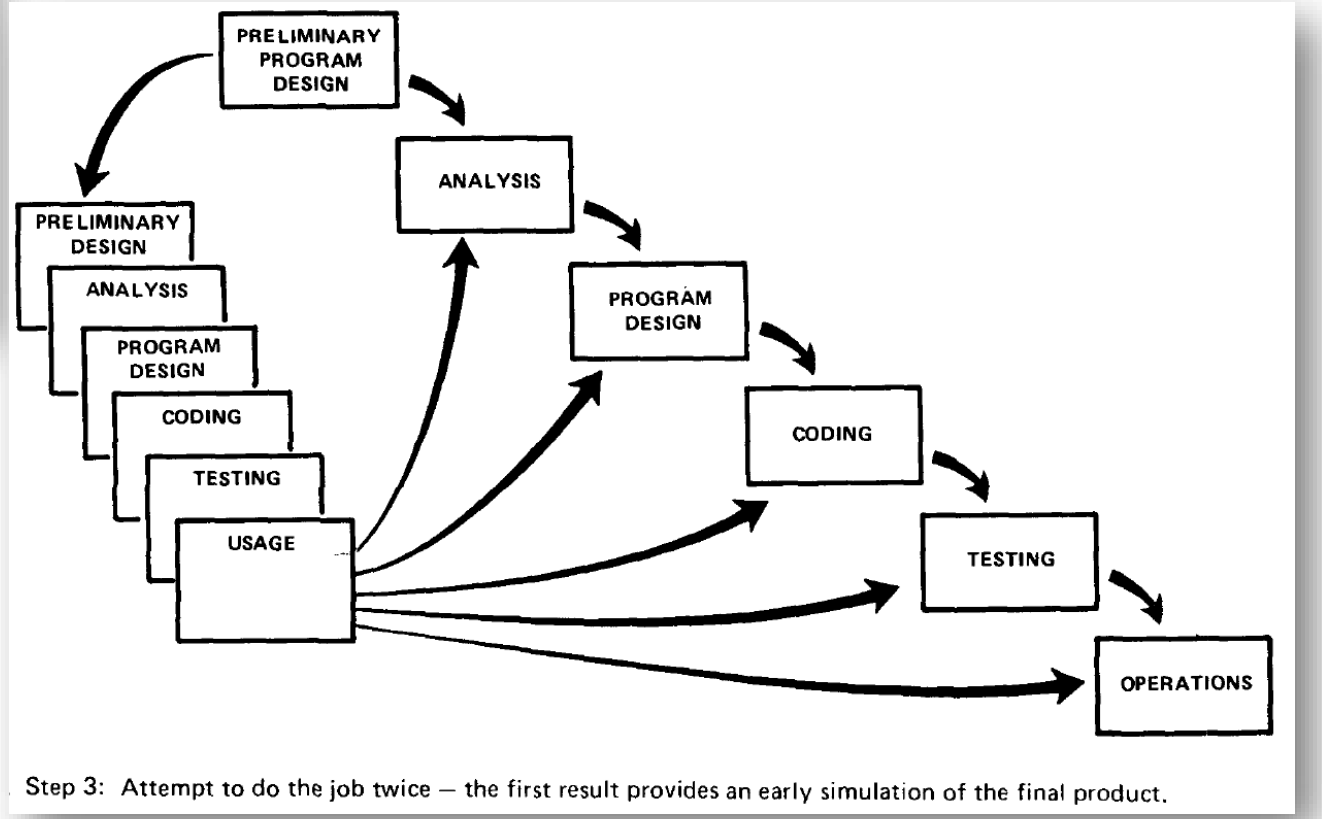
ISO/IEC/IEEE 29148:2018 - Systems and software engineering — Life cycle processes — Requirements engineering


Software Requirements
Software Requirements Fundamentals
Definition of a Software Requirement
Categories of Software Requirements
Software Product Requirements and Software Project Requirements
Functional Requirements
Nonfunctional Requirements
Technology Constraints
Quality of Service Constraints
Why Categorize Requirements This Way?
System Requirements and Software Requirements
Derived Requirements
Software Requirements Activities
Requirements Elicitation
Requirements Sources
Common Requirements Elicitation Techniques
Requirements Analysis
Basic Requirements Analysis
Economics of Quality of Service Constraints
Formal Analysis
Addressing Conflict in Requirements
Requirements Specification
Unstructured Natural Language Requirements Specification
Structured Natural Language Requirements Specification
Acceptance Criteria-based Requirements Specification
Model-Based Requirements Specification
Additional Attributes of Requirements
Incremental and Comprehensive Requirements Specification
Requirements Validation
Requirements Reviews
Simulation and Execution
Prototyping
Requirements Management Activities
Requirements Scrubbing
Requirements Change Control
Scope Matching
Practical Considerations
Iterative Nature of the Requirements Process
Requirements Prioritization
Requirements Tracing
Requirements Stability and Volatility
Measuring Requirements
Requirements Process Quality and Improvement
Software Requirements Tools
Requirements Management Tools
Requirements Modeling Tools
Functional Test Case Generation Tools

SWEBOK®

Kdaj zbiramo zahteve - primer





Step 3: Attempt to do the job twice – the first result provides an early simulation of the final product.

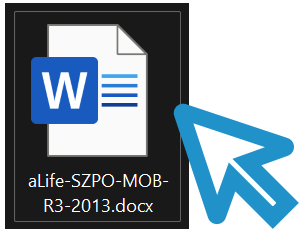
Zbiranje zahtev

Razumevanje zahtev

Specifikacija zahtev (razumljivo stranki in razvijalcem)

Dokumentiranje **ciljev**

Specifikacija zahtev je tipično tudi **pogodba** med stranko in razvijalci



Katere aktivnosti spadajo med zbiranje zahtev?

KAJ pričakujemo od sistema:

- Definiranje poslovnih ciljev, vizije
- Potrebe vlog uporabnikov
- Odvisnosti med drugimi rešitvami
- Definiranje besednjaka
- Definiranje vlog
- Definiranje modela primerov uporabe
- Določitev prioritet
- Podrobnejše opisovanje primerov uporabe
- Modeliranje GUI
- Prototipiranje GUI

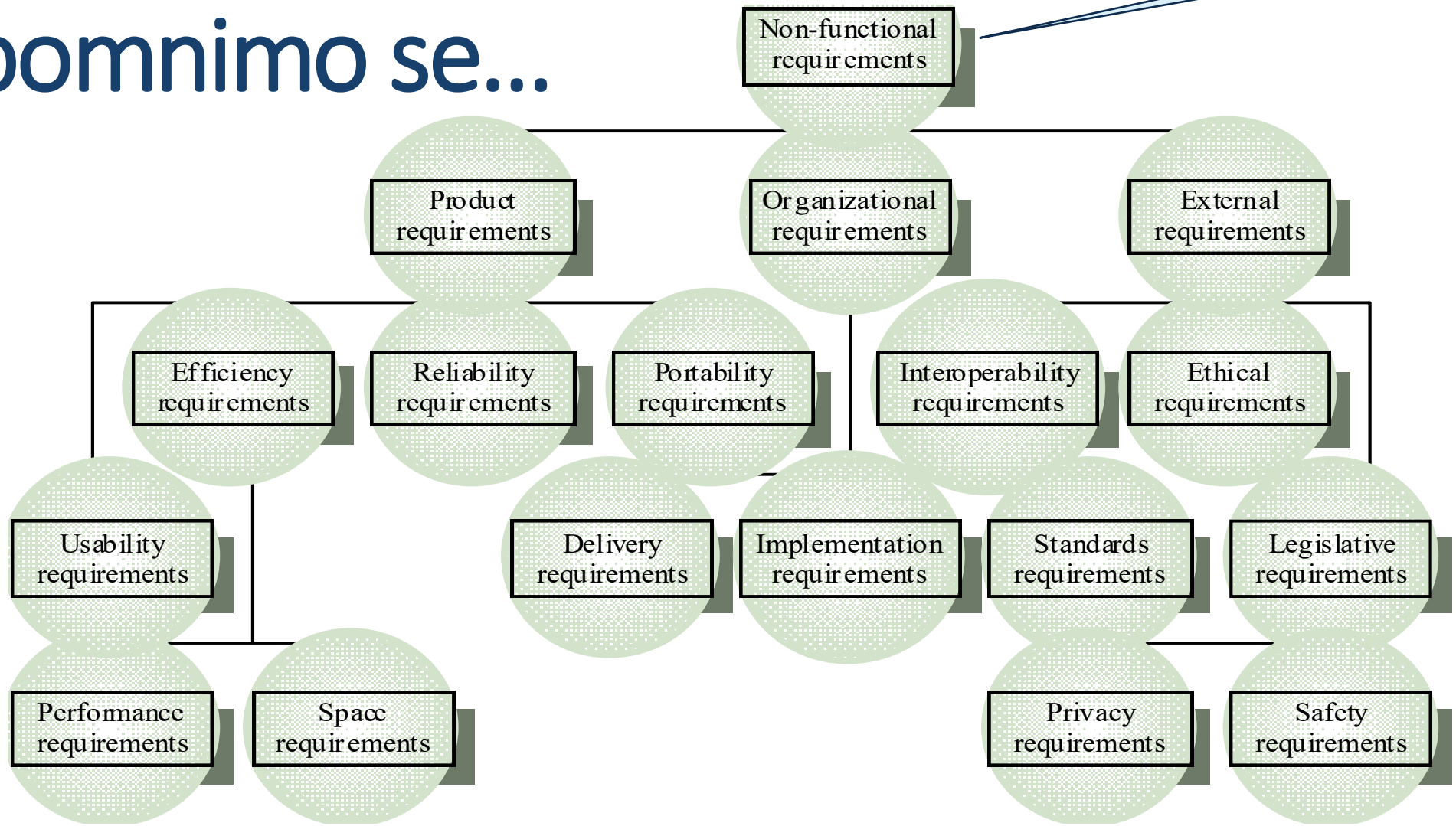
→ definiranje sistema iz vidika uporabnika

> Requirements are not limited to only coming from people. Other, non-person requirements sources can include:
>
> - documentation such as requirements for previous versions, mission statements, concept of operations;
> - other systems;
> - larger business context including organizational policies and processes;
> - computing environment.

SWEBOK®

Spomnimo se...

= omejitve



[Sommerville]

Nefunkcionalne zahteve

Nonfunctional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs. Also known as system qualities, nonfunctional requirements are just as critical as functional. They ensure the usability and effectiveness of the entire system. Failing to meet any one of them can result in systems that fail to satisfy internal business, user, or market needs, or that do not fulfill mandatory requirements imposed by regulatory or standards agencies. In some cases, non-compliance can cause significant legal issues (privacy, security, safety, to name a few).

[https://www.scaledagileframework.com/nonfunctional-requirements/]

In systems engineering and requirements engineering, a non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. They are contrasted with functional requirements that define specific behavior or functions. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture, because they are usually architecturally significant requirements.[1]

[https://en.wikipedia.org/wiki/Non-functional_requirement]

Zahteve: "Minimum"

Vizija

Besednjak

Popis vlog uporabnikov (in ostalih deležnikov)

Ter njihovih zahtev

V odvisnosti od modela razvoja:

Funkcionalne zahteve

Nefunkcionalne zahteve (omejitve)

Tipično iz poslovnega vidika

"Želimo omogočiti hitro dostavo hrane naročniku"

Da vemo o čem se pogovarjamo!
Kako lahko modeliramo zahteve, če osnovni koncepti niso jasni?

Razumemo razliko med konkretnimi uporabniki in vlogami?
Pa povezavami med vlogami?

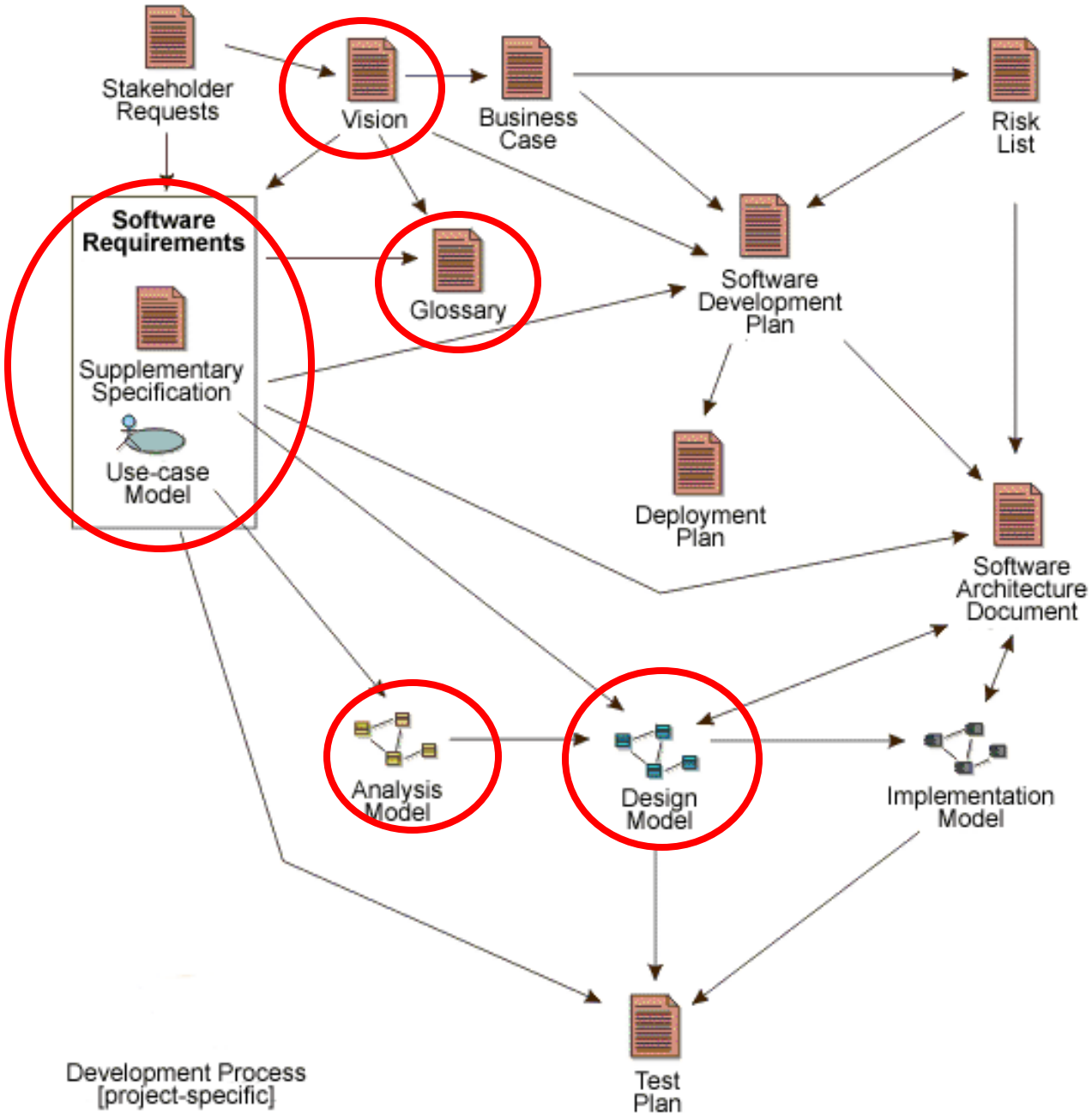
Kaj bi rešitev naj počela / zakaj jo potrebujemo / katere **funkcije** ponuja…

"Prenos osnovnega sredstva na drugega uporabnika"

Kaj sistem ("žal") mora ponujati, a ni **neposredno povezano s funkcionalnostmi**

"Prijava", "Reset gesla", "Hitro delovanje"…

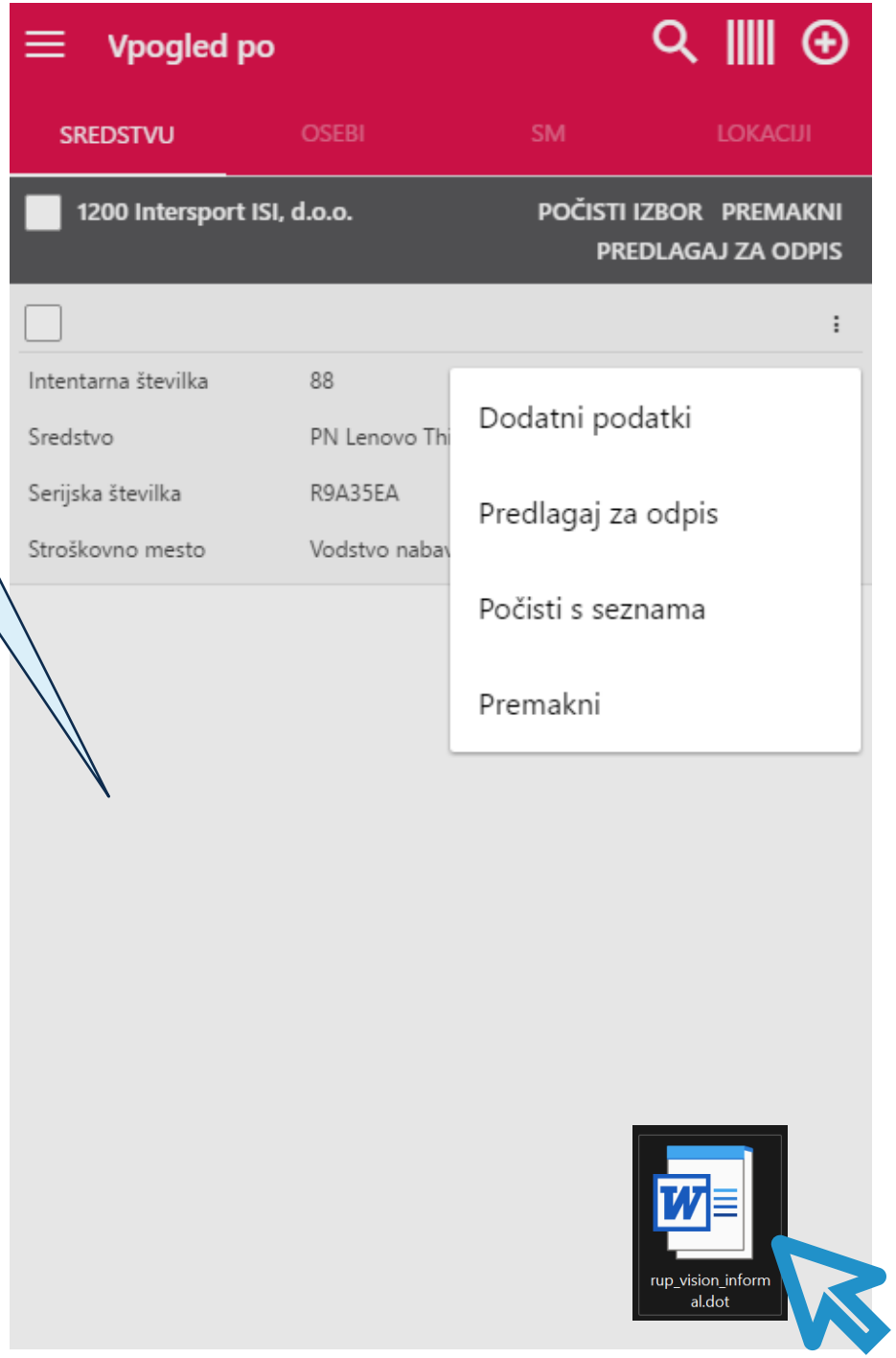
Sistem prioritet?



[Rational Method Composer]

Vizija

Vizija tega izdelka?



Kratek “plot” informacijske rešitve

Lahko razkrije “problem-za-problemom”

Vključuje **seznam deležnikov**

Tipično se vsi strinjajo z vizijo rešitve

Meje rešitve

Kaj vključujemo

(Pomembneje) Česa NE vključimo

Morebitne omejitve

Na poslovni ravni!

Ključne funkcionalnosti

Morda tudi knjučne alternative

…

Zgradba zavisi od uveljavljenih (uporabljenih) praks / standardov razvojne ekipe.

Page title

[https://www.atlassian.com/software/confluence/templates/vision-to-values]

Define your reason for being so you can form actionable business strategies

👁 Vision

Start by describing your company's main vision. This is the primary goal that unites each product goal and team member's work.

🚀 Mission

Explain the goal for each of your company's products and how it helps achieve your company's main vision. Each product goal should be measurable and achievable.

🔮 Strategies

Identify your company's competitive advantages. Then decide how to transform those strengths into actionable strategies to help you achieve product goals.

Competitive advantage	Product strategy
e.g., The company has a strong AI engineering team	e.g., The company can leverage AI to develop new products

🎯 Objectives

Track your progress by aligning each product goal with a measurable metric.

Product goal	Metric
e.g., Increase revenue	e.g., 40% increase in y/y revenue

🧩 Tactics

List the tactics you're planning to execute your product strategies. Organize your list by prioritizing tactics that best help your company achieve its main vision.

Priority	Product goal	Tactic
1	e.g., Increase revenue	e.g. Launch premium tier

❤️ Customer values

List the reasons why customers should use your company's products or services. Organize your list by prioritizing values that are most important to your customers and most beneficial to your business.

Priority	Customer value
1	e.g., Customers can get started on document collaboration faster

👥 Customer segments

List and organize the customer segments you're targeting. This will help you focus product strategies and prioritize tactics.

Priority	Customer segment
1	e.g., Customers can get started on document collaboration faster

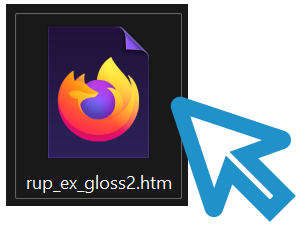
Besednjak

Tipično del SZPO

Ključni del dokumenta za sporazumevanje / razumevanje domene in rešitve

Sestavi razvojna ekipa v sodelovanju z domentskimi strokovnjaki

Funkcionalnosti in/ali vloge uporabnikov kot stranski produkt



[https://www.atlassian.com/software/confluence/templates/content-design-glossary]

Create an A-Z guide to help your team design effective content

Page title

Table of Contents

Table of Contents

The results of this element aren't visible when editing. Preview the page or publish it to see how it will look.

A A-D

Name	Definition	Guidelines	Example	
e.g., annual percentage rate (APR)	e.g., the rate of interest that a person pays on money that they borrowed	• e.g., only use the acronym • Type / to add a link to a DACI or other resource	DO	e.g., The company offers credit cards with an introductory 0 percent APR.
			DONT	e.g., The company offers credit cards with an introductory 0 percent annual percentage rate.

E E-K

Name	Definition	Guidelines	Example

L L-O

Name	Definition	Guidelines	Example

ATLASSIAN

Confluence

Zahteve - Primer kazala *[IEEE]* 1/3

Title, authors, legal statements…

Summary

Introduction

- Purpose of the document
- Goals
  - *Software will contain*
  - *Software will NOT contain*
  - *Benefits*
- Naming, vocabulary
- References to other documents (name, location etc.)
- Content overview

Zahteve - Primer kazala *[IEEE]* 2/3

- General description
  - Functions
  - Users
  - Limitations
    - *Software*
    - *Hardware*
    - *Environment*
  - Interdependence

Zahteve - Primer kazala *[IEEE]* 3/3

- Specific requirements
  - Functional Requirements
    - *Req #1*
    - *Req #2*
    - *Req #N*
  - Non-Functional Requirements
    - *Req #1*
    - *Req #2*
    - *Req #N*
  - Performance requirements
  - Usability requirements
  - Properties
    - *Availability*
    - *Security*
    - *Mantainability*
    - *Hardware, software requirements*
  - Other properties

Del dokumentacije je lahko tudi prototip!

Dokumentacija zahtev

Oblika dokumentacije variira, a tipično vsebuje:

General

- Purpose and scope of system
- Objectives and criteria for success
- List of terminology, organizations involved, etc.

Description of current system(s)

Requirements of proposed system

- Overview
- Functional Requirements
- Usability requirements
- Non-functional requirements

System models

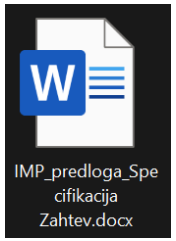
- Scenarios
- Use cases
- Models used during analysis

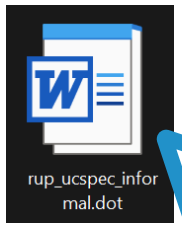
Detailed specifications

- Business rules, specifications, etc. (e.g., reference to an accounting standard)
- Data flow, sources of data, data validation
- etc., etc.,

[https://www.atlassian.com/software/confluence/templates/product-requirements]

Your pre-launch prep and post-launch progress check





Page Properties | _parentId = 294917

Target release	Type // to add a target release date
Epic	Type /Jira to add Jira epics and issues
Document status	DRAFT
Document owner	@ mention owner
Designer	@ designer
Tech lead	@ lead
Technical writers	@ writers
QA	

Objective

Provide context on this project and explain how it fits into your organization's strategic goals.

Success metrics

List project goals and the metrics you'll use to judge its success.

Goal	Metric
e.g., Simplify the user experience	e.g., Customer satisfaction score increases

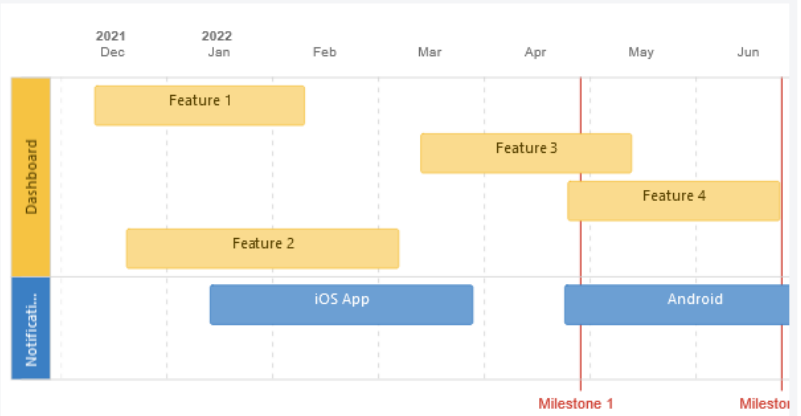
Assumptions

List any assumptions you have about your users, technical constraints, or business goals.

Milestones

Type /planner to create a visual roadmap and help your team stay on track. To edit workstreams or dates, select the placeholder and tap the pencil icon.

Roadmap Planner



Type /trello to add a card or board to this page or /jira to include a Jira issue, chart, or project.

Requirements

Requirement	User Story	Importance	Jira Issue	Notes
e.g., Must be mobile responsive	e.g., John is a PM who wants to check on his team's progress from the train station	HIGH		

User interaction and design

Type /image to add mockups, diagrams, and screenshots related to the requirements.

Open Questions

Question	Answer	Date Answered
e.g., How might we make users more aware of this feature?	e.g., We'll announce the feature with a blog post and a presentation	Type // to add a date

Out of Scope

List the features discussed which are out of scope or might be revisited in a later release.

•

ATLASSIAN Confluence

Alternative: User Stories

Se vrnemo čez nekaj tednov…

Stories act as a 'pidgin language,' where both sides (users and developers) can agree enough to work together effectively.

Bill Wake, co-inventor of Extreme Programming

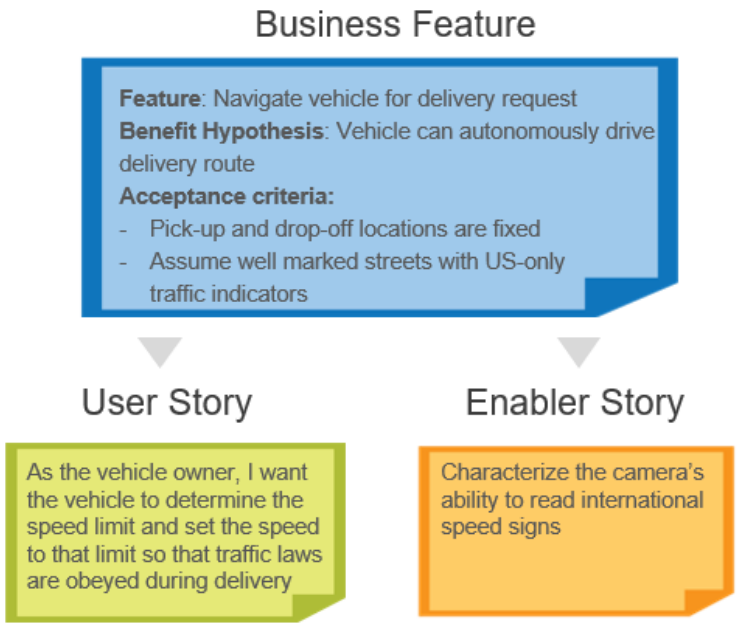
Modeliranje zahtev (z UML):

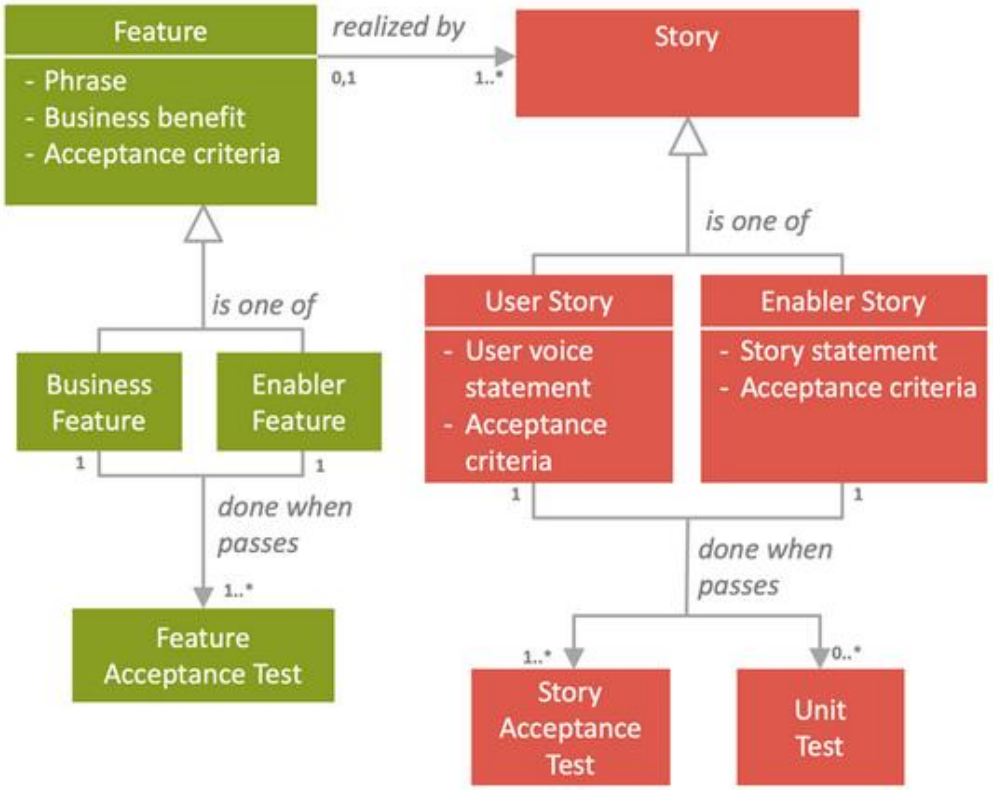
- Diagrami primerov uporabe (akterji, primeri uporabe, povezave, meje sistema…)
- Opisi primerov uporabe (besedilno, grafično)

Ima tudi mnoge alternative

- Npr. Epiki, uporabniške zgodbe..

User Story	@life weigth history	P	1
As an	user		
I want	see my weigth list and graph		
In order to	evaluate progress		
Comments			





https://www.scaledagileframework.com/story/

Tudi uporabniške zgodbe (Primer)

User Story	@life weigth history	P	1
As an	user		
I want	see my weigth list and graph		
In order to	evaluate progress		
Comments			

NAZIV
PRIORITETA
BESEDILO
DODATNE INFO (LINK)

NAZIV
(PRIORITETA)
OCENA
KRATEK OPIS
Definicija končanja (DoD)

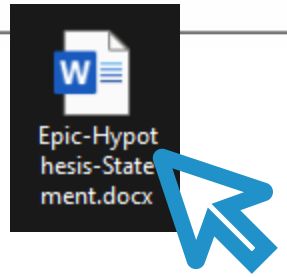
Task	Weigth list	P	1
Info	User want to see a table with dates and weigth. Table should be sorted according to date – descending (last on top).	T	
How to test?	Insert some weigths in mixed date oder. See table and inspect that all inserts are there and that the order is correct.		

Združevanje zahtev
(Uporabniških zgodb)

V realnosti uporabniške zgodbe včasih združujemo -> Epic

Epics	≠	Projects
Implemented by stable, cross-functional Value Streams and ARTs.		Implemented by temporary teams, which disband after work is completed.
No definitive start and end date; scope is variable. Continue until WSJF says otherwise.		Definitive start and end date; scope is fixed. All scope must be implemented.
Progress is measured as outcomes against the benefit hypothesis.		Progress is measured based on task completion.
Lean Business Case, based on benefit hypothesis and definition of an MVP.		Overly detailed business case, based on speculative ROI.
Implementation follows the build – measure – learn SAFe Lean Startup Cycle.		Implementation typically follows phase-gated, sequential (waterfall) process.
After the Lean Business Case is approved, commitment is to the evaluation of the MVP.		After business case is approved, up-front commitment is made to the entire project scope.

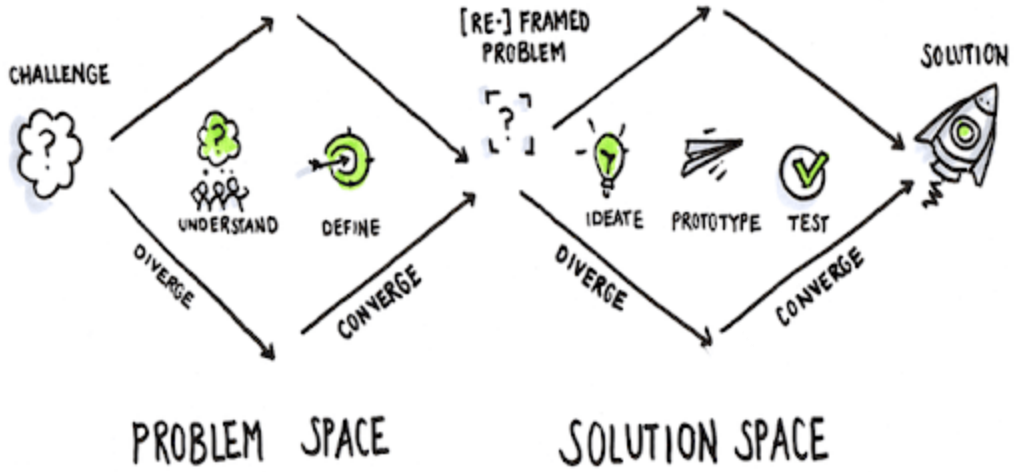
Epic Hypothesis Statement	
Funnel Entry Date:	<The date that the epic entered the funnel.>
Epic Name:	<A short name for the epic.>
Epic Owner:	<The name of the epic owner.>
Epic Description:	<An elevator pitch (value statement) that describes the epic in a clear and concise way.> **For** <customers> **who** <do something> **the** <solution> **is a** <something – the 'how'> **that** <provides this value> **unlike** <competitor, current solution or non-existing solution> **our solution** <does something better – the 'why'>
Business Outcomes:	<The measurable benefits that the business can anticipate if the epic hypothesis is proven to be correct.>
Leading Indicators:	<The early measures that will help predict the business outcome hypothesis. For more on this topic, see the Innovation Accounting advanced topic article.>
Nonfunctional Requirements (NFRs):	<Nonfunctional requirements (NFRs) associated with the epic.>



https://www.scaledagileframework.com/epic

[https://www.productboard.com/glossary/dual-track-agile]

Pomnimo: načrtovanje



KAJ → KAKO

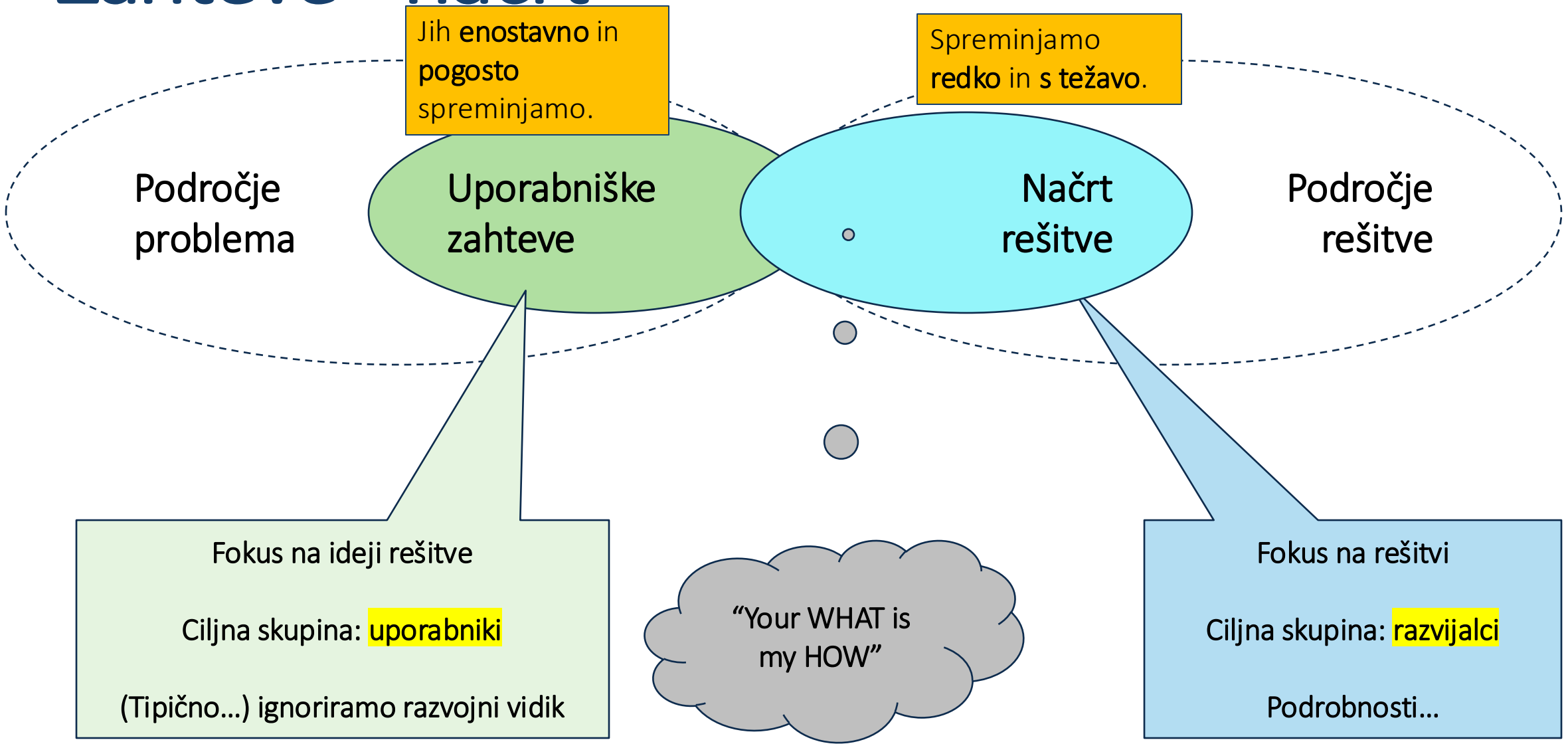
- Podrobnosti!
- Razredi, operacije
- Vmesniki, protokoli
- Algoritmi

Tipično vezano na konkretno platformo, programski jezik ipd.

OIS: "Načrt daje odgovor na vprašanje, KAKO izdelati sistem, da bo ustrezal zahtevam, ki smo jih evidentirali v fazi analize."

(konstruktiven) konflikt med
- problemom (zahtevami) in
- rešitvijo (njenim načrtom)?

Zahteve - načrt



[https://www.atlassian.com/software/confluence/templates/design-review]

Document requirements and run a productive design review

Status	NOT STARTED
Owner	@ mention owner
Contributors	@ contributors
Goals	Provide context for your team and explain the goals for your design review
Prototype	Paste a link to an InVision prototype
Jira tickets	Paste links to Jira tickets
On this page	Table of Contents Table of Contents The results of this element aren't visible when editing. Preview the page or publish it to see how it will look.

🎯 Requirements

Product requirements

Paste a link to a page you've made that explains the product requirementsproduct requirements

Problem statements

- Describe the problem you're trying to solve and explain why it's important to your customers

Success metrics

- Explain the metrics, signals, and customer behavior that define success

📸 Latest designs

Add designs ready to launch and explain the user context and conditions for each screen during your review. If you're still exploring design options, use theDesign decision template.

	Screens	Context	Conditions
1	Type /image to add a screenshot		
2			

Previous design options

Add previous design versions and options that weren't selected.

	Screens	Design decision
1	Type /image to add a screenshot	Explain why this option wasn't selected
2		

❓ Open issues

- List open issues or questions raised during the design review
-
-

🧪 Design test plan

Organize information and identify research opportunities.

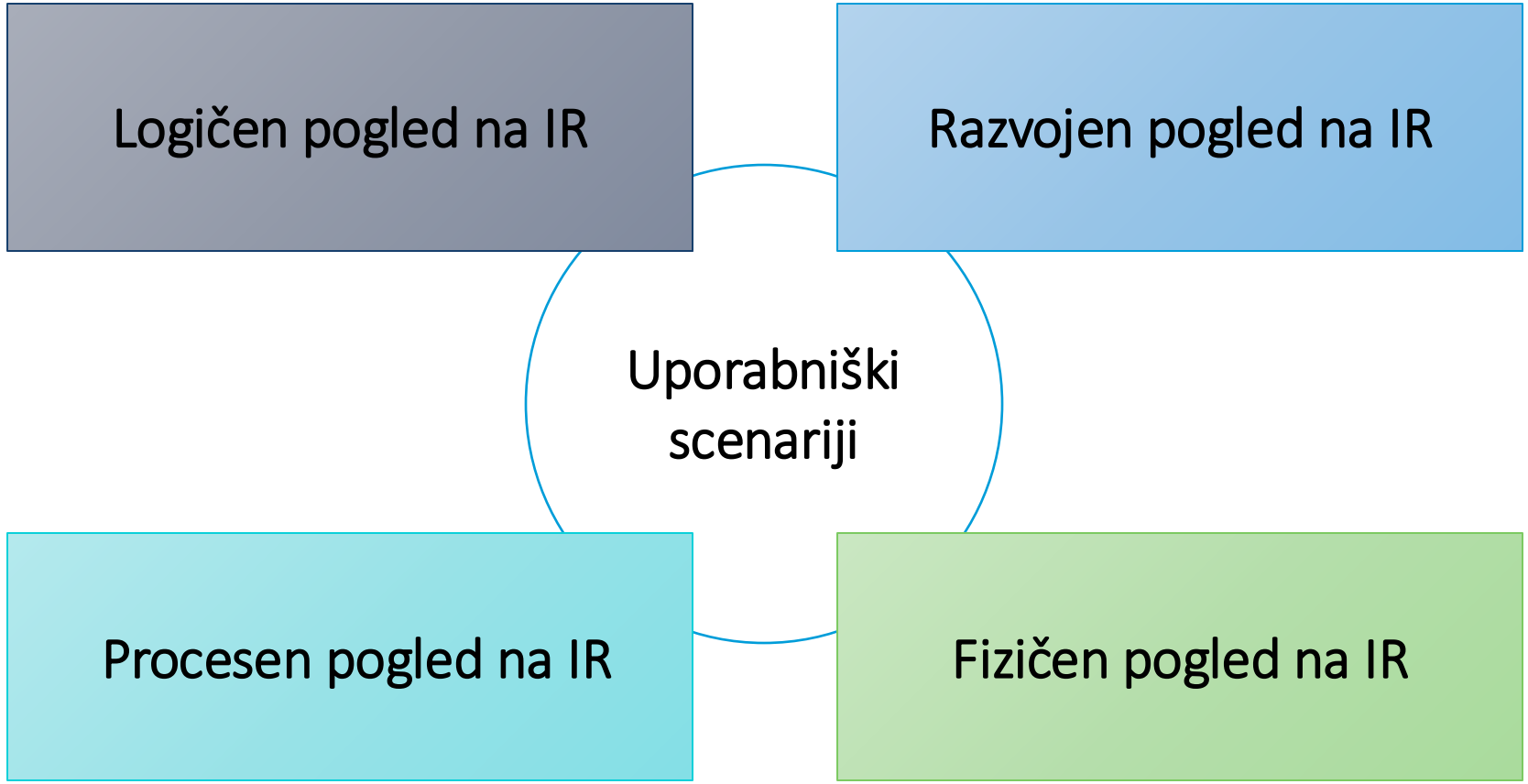
Scenario	Prototype	What we know	What we need to learn
e.g., Users scroll and browse through the homepage	Paste a link to an InVision prototype		

🚢 Deployment plan

Organize deployment plans and prepare for your design launch.

	Date	Plan type	Target segment	Notes	Action items
1	Type // to add a date	TEST / FEATURE FLAG / OTHER	e.g., Customers who upgraded to the premium service in Q1		Type your acti...
2					

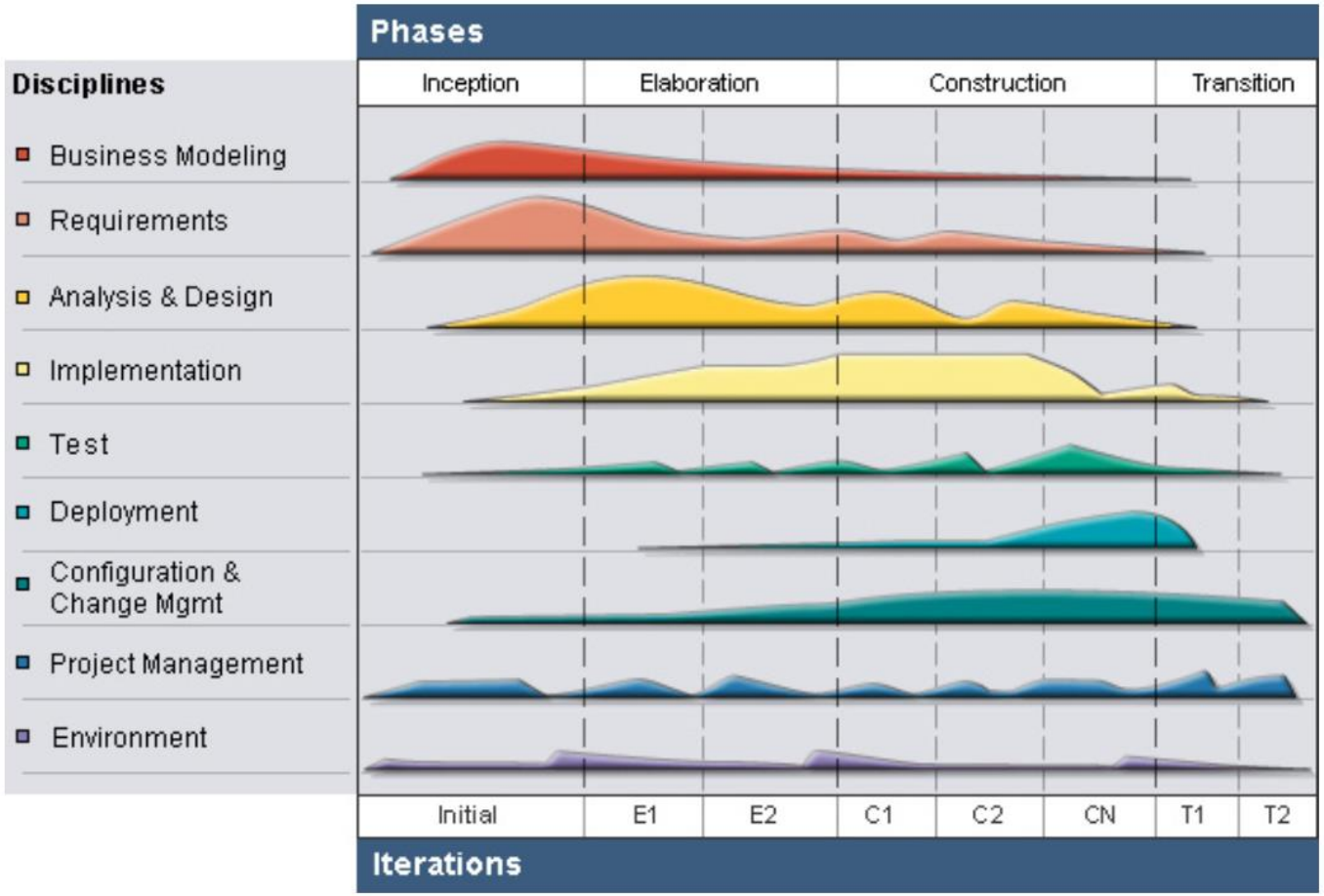
4+1 – Eden bolj celovitih pogledov na arhitekturo IR



Osnova so zahteve uporabnika!

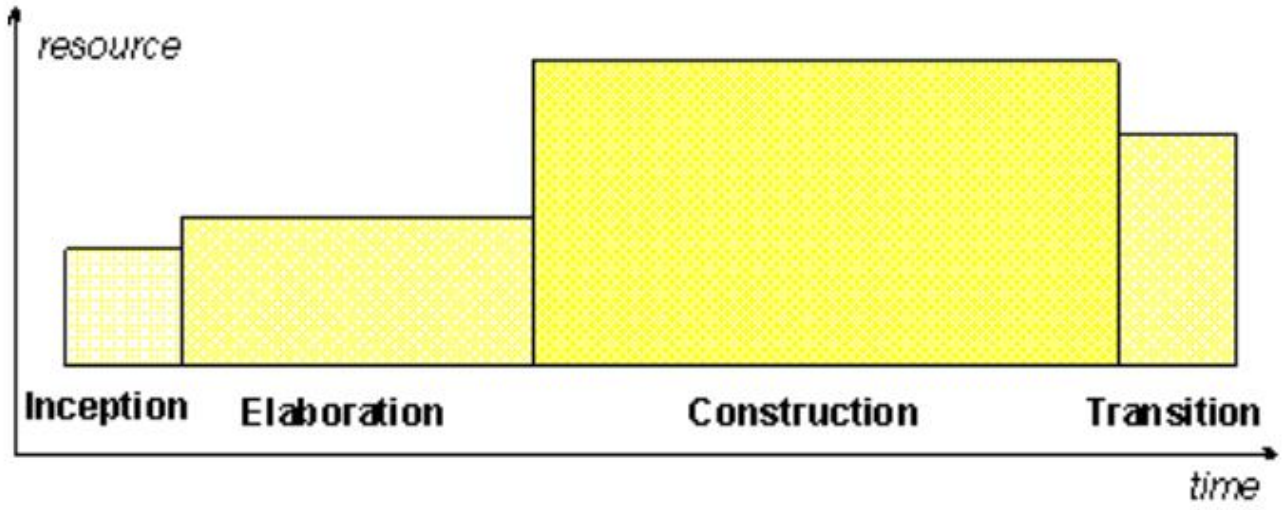
Izhaja iz ogrodja RUP, predvideva UML ipd. – a se lahko prilagodi

Kdaj zbiramo zahteve - primer



Čas, namenjen fazi zbiranja zahtev

	inception	elaboration	construction	transition
Effort	~5 %	20 %	65 %	10%
Schedule	10 %	30 %	50 %	10%



Effort Distribution per Activities for Small Software Development Project Uses Prototype Model

Sholiq
Department of Information Systems
Institut Teknologi Sepuluh Nopember
Kampus ITS Sukolilo-Surabaya 60111,
Surabaya, Indonesia
sholiq@is.its.ac.id

Arifin Puji Widodo
Departemen of Accounting
Institut Bisnis dan Informatika Stikom

Teguh Sutanto
Department of Information Systems
Institut Bisnis dan Informatika Stikom

No	Activity	Roles (%)						Total
		Project Manager	System Analyst/ Design	Programmer	Tester	Technical Support	Documenter	
1	**Software Development**	**8.11**	**14.46**	**36.12**	**2.84**	**1.46**	**1.82**	**64.81**
	Requirement	1.32	1.18	1.45	0.16	0.16	0.00	4.28
	Design	1.35	1.87	1.12	0.00	0.00	0.00	4.34
	Build Prototype	0.99	2.49	6.00	0.40	0.00	0.00	9.87
	Costumer Evaluation	0.79	0.65	2.11	0.49	0.00	0.00	4.05
	Coding	2.21	3.63	17.14	1.42	0.53	1.45	26.37
	Testing	0.46	3.27	3.36	0.36	0.00	0.00	7.46
	Acceptance & Deployment	0.99	1.37	4.94	0.00	0.77	0.37	8.44
2	**On Going Activity**	**7.94**	**3.66**	**4.81**	**0.96**	**1.22**	**2.80**	**21.39**
	Project management	5.44	1.47	0.99	0.00	0.00	0.79	8.69
	Configuration management	1.09	1.18	0.66	0.33	1.22	0.56	5.03
	Training and technical support	0.99	0.69	2.50	0.63	0.00	0.00	4.80
	Documentation	0.43	0.33	0.66	0.00	0.00	1.45	2.86
3	**Quality and maintenance**	**2.14**	**1.80**	**6.59**	**0.82**	**2.44**	**0.00**	**13.80**
	Quality assurance	2.14	1.80	2.97	0.82	0.53	0.00	8.26
	Evaluation and maintenance	0.00	0.00	3.63	0.00	1.91	0.00	5.54
	Total	**18.20**	**19.92**	**47.52**	**4.62**	**5.12**	**4.62**	**100.00**

LEH [21] AND

Primandari & Sholiq (%)
1.60
7.50
6.00
52.00
5.50
3.80
4.30
8.40
1.00
7.00
0.90
2.00
100.00

Phase
Design
Code
Test
Transition

TABLE I. I

Phase			
Requirements Analysis and Design	20	15	5
Detail Design, Code and Unit Test	57	10	5
SW Integration &Test	23	40	30
Relative Effort	100	65	40

Evaluation and Testing	20.84	2.00
Total	100.00	100.00

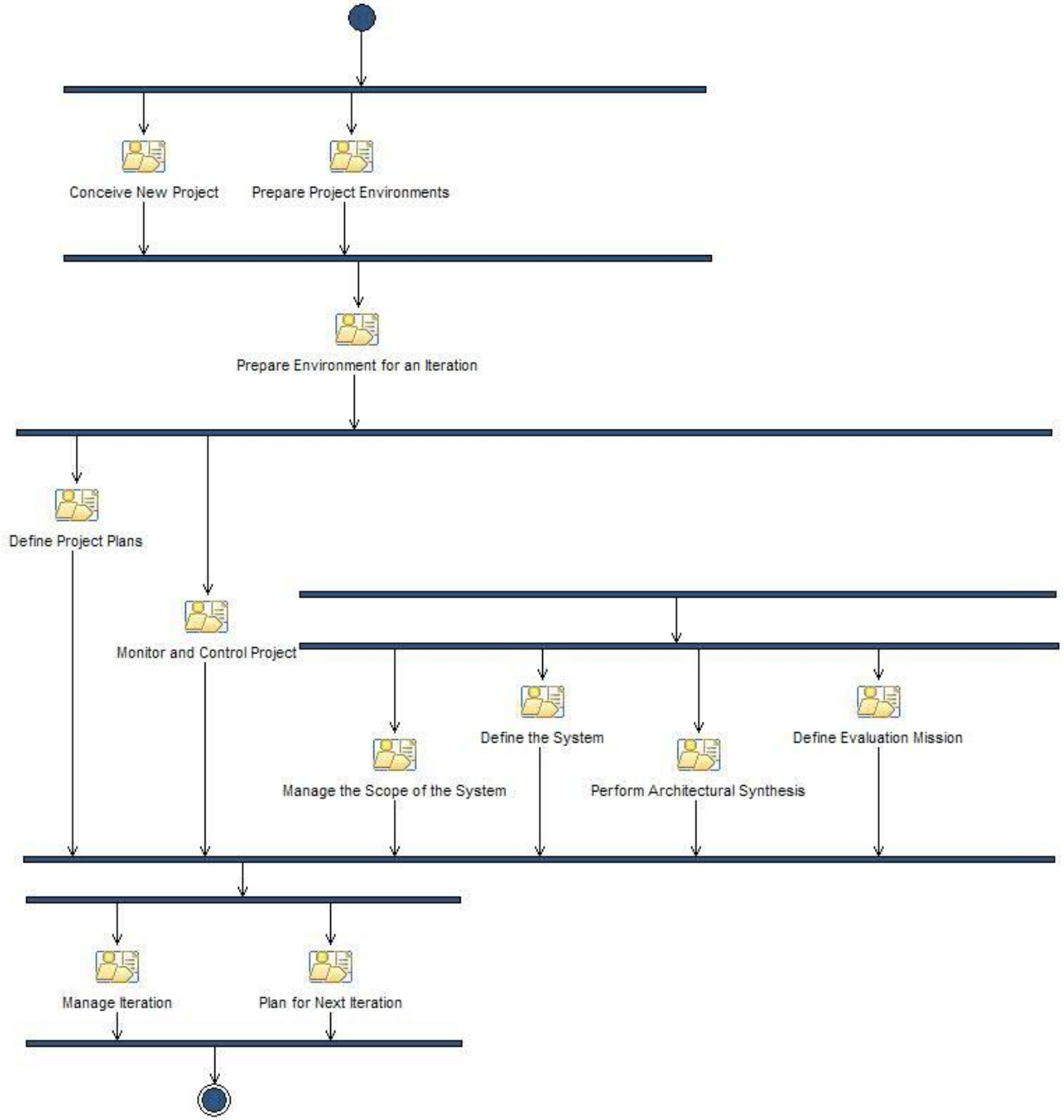
Primer faze – začetna faza (inception)

Rezultat je načrt, ki je iterativen in inkrementalen.

Osnovne zahteve vsebujejo nabor preliminarnih primerov uporabe.

Arhitektura je na začetku le oris podsistemov in funkcij, ki jih opisujejo – razširitev v nabor modelov sledi v kasnejših iteracijah.

Inception


Conceive New Project
Prepare Project Environments
Prepare Environment for an Iteration
Define Project Plans
Monitor and Control Project
Manage the Scope of the System
Define the System
Perform Architectural Synthesis
Define Evaluation Mission
Manage Iteration
Plan for Next Iteration

Aktivnosti - začetna faza

Definiranje obsega projekta

Priprava poslovnega načrta

Priprava kandidata arhitekture

Priprava projektnega okolja

Mejnik – začetna faza

Naročnik **soglaša z obsegom projekta** in oceno stroškov in trajanja.

Naročnik se **strinja z zahtevami** in razume njihovo ozadje.

Obstaja dogovor o ustrezni razporeditvi **prioritet**, **tveganjih** na projektu in **poteku dela** na projektu.

Določena so vsa **tveganja in načrt reševanja** ob morebitnem nastopu.

Primer faze – zbiranje zahtev (elaboration)

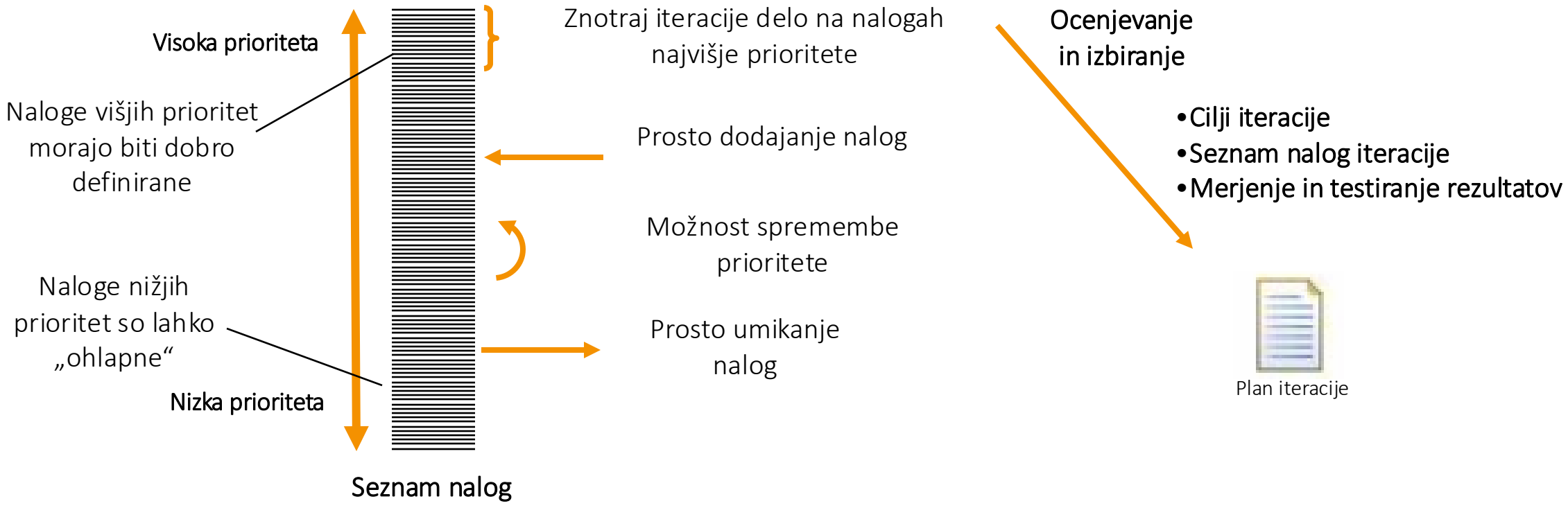
dodela in razširi preliminarne primere uporabe in razširi arhitekturo s petimi pogledi na informacijsko rešitev:

- model primerov uporabe,
- model zahtev,
- model načrtovanja,
- model implementacije in
- model predaje.

V določenih primerih lahko tu že dobimo prvi izvršljiv sistem, ki ga ne zavržemo in nam predstavlja osnovo za nadaljnje delo.

Izvedemo tudi prve spremembe načrta.

Ideje agilnih metod v OpenUP



Uporabniške zgodbe in INVEST

Independent

Negotiable

Valuable

Estimatable

Small

Testable

Konkretneje: DoR - Definition of Ready

Primer

Kontrolni seznam, kdaj je uporabniška zgodba primerna za razvoj

	Definition of Ready
❑	Business value is clearly articulated.
❑	Details are sufficiently understood by the development team so it can make an informed decision as to whether it can complete the PBI.
❑	Dependencies are identified and no external dependencies would block the PBI from being completed.
❑	Team is staffed appropriately to complete the PBI.
❑	The PBI is estimated and small enough to comfortably be completed in one sprint.
❑	Acceptance criteria are clear and testable.
❑	Performance criteria, if any, are defined and testable.
❑	Scrum team understands how to demonstrate the PBI at the sprint review.

OO modeli in UML (Unified Modeling Language)

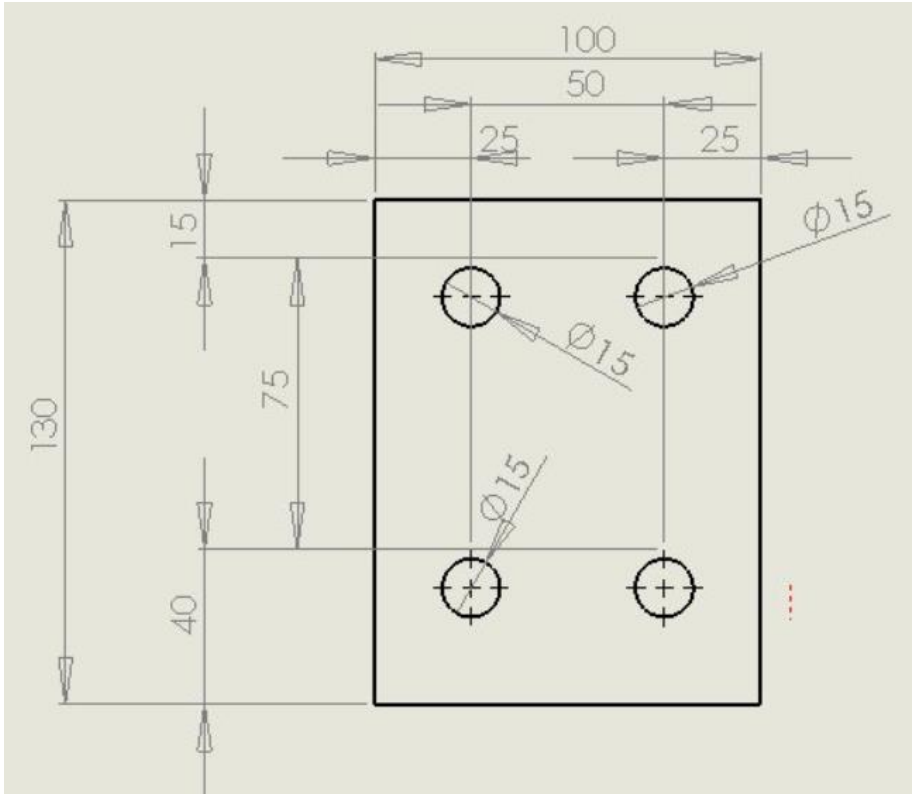
UML – standarden modelirni jezik - notacija

„Lepilo“ med specifikacijami in implementacijo
Modeli specifikacij, načrta, arhitekture, namestitve…

Specificiranje in dokumentiranje načrtov

Neodvisen od procesa in programskih platform

Še posebej primeren za OO sisteme
In uporabo v procesnih ogrodjih, kot je (R)UP



Unified Modeling Language

http://www.omg.org/spec/UML

http://www.uml.org

POPULAR OMG STANDARDS

WE SET THE STANDARDS®

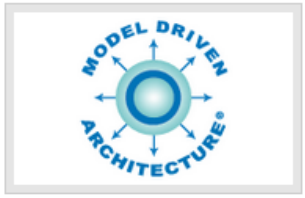
We are proud to have finalized our 250th specification! All OMG standards are free to download and use by anyone. We are proud that OMG standards are found in everything from cell phones to satellites, from manufacturing plants to jet engines. Our growing certification program allows individuals to demonstrate their understanding and mastery of some of our most well-known standards to potential employers or clients.

BPMN™
Business Process Model and Notation™

CORBA®
Common Object Request Broker Architecture™

CWM™
Common Warehouse Metamodel™

DDS™
Data-Distribution Service for Real-Time Systems™

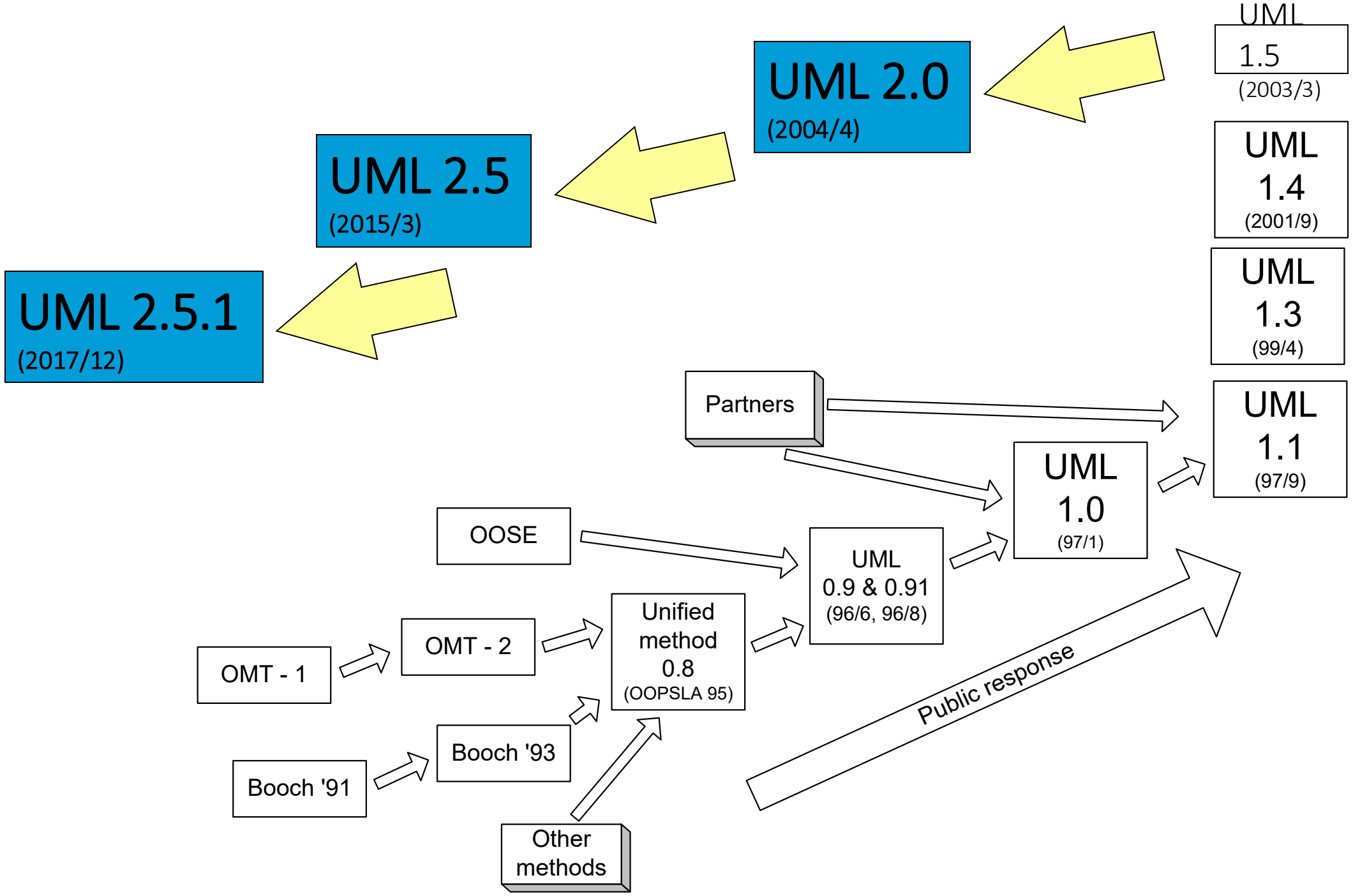


MDA®
Model Driven Architecture®

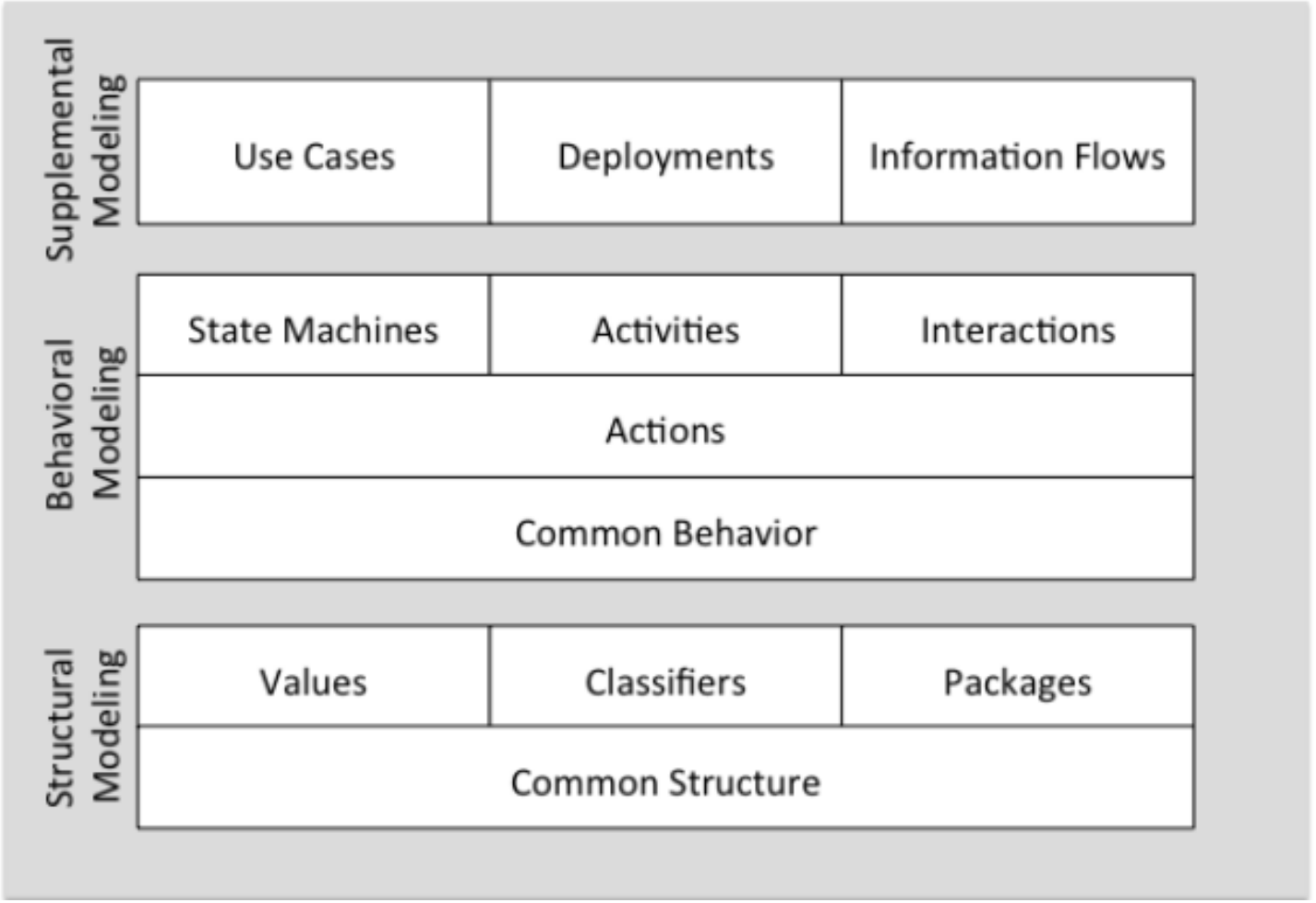
MOF™
MetaObject Facility Specification™

SYSML®
Systems Modeling Language™

UML®
Unified Modeling Language™


UML 2.5.1
(2017/12)
UML 2.5
(2015/3)
UML 2.0
(2004/4)
UML
1.5
(2003/3)
UML
1.4
(2001/9)
UML
1.3
(99/4)
UML
1.1
(97/9)
Partners
UML
1.0
(97/1)
OOSE
UML
0.9 & 0.91
(96/6, 96/8)
OMT - 1
OMT - 2
Unified
method
0.8
(OOPSLA 95)
Booch '91
Booch '93
Other
methods
Public response

Področja uporabe



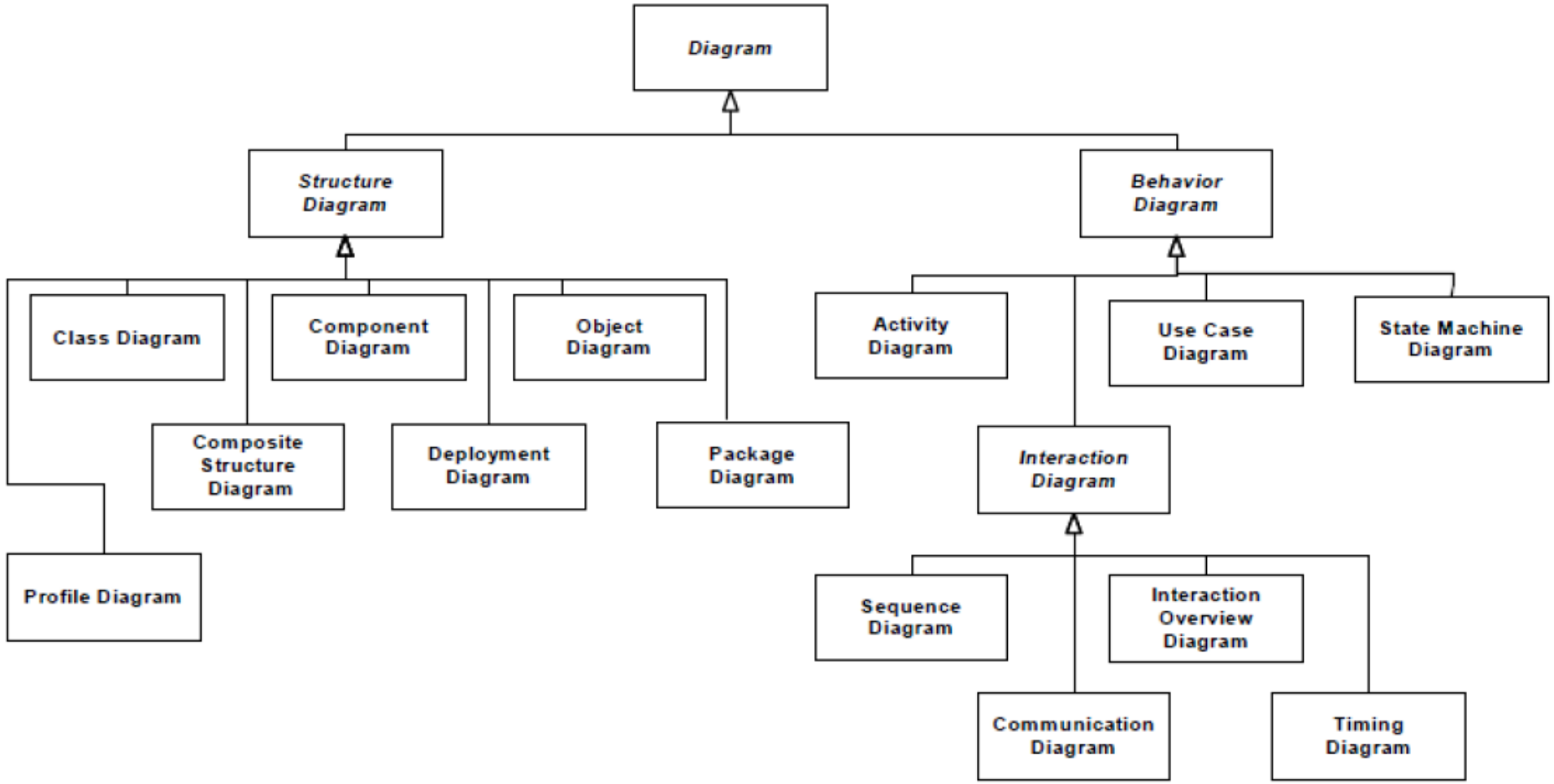
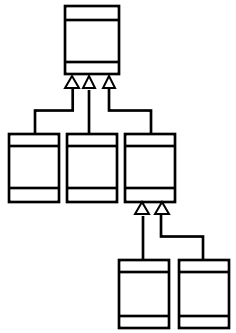


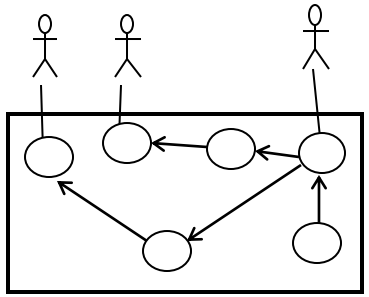
Diagram
Structure Diagram
Behavior Diagram
Class Diagram
Component Diagram
Object Diagram
Activity Diagram
Use Case Diagram
State Machine Diagram
Composite Structure Diagram
Deployment Diagram
Package Diagram
Interaction Diagram
Profile Diagram
Sequence Diagram
Interaction Overview Diagram
Communication Diagram
Timing Diagram

Razredni Diagrami



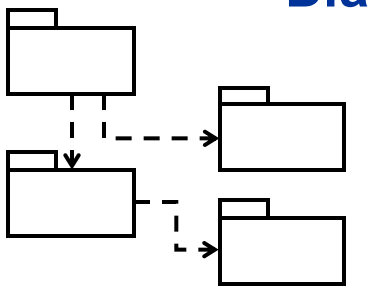
- Struktura
- Relacije
- Modularna zgradba

Primeri uporabe



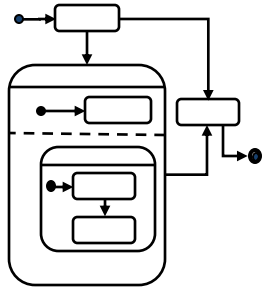
- Uporabnikov pogled
- Funkcionalnosti
- - vizualen pregled

Diagrami paketov



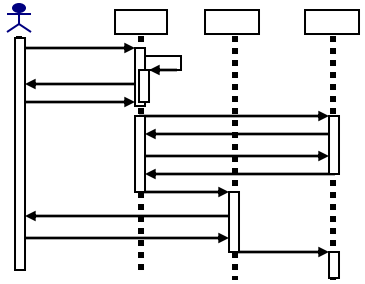
- Združevanje razredov
- Pregled arhitekture
- Sorodne stvari
- Odvisnosti

Diagrami stanj



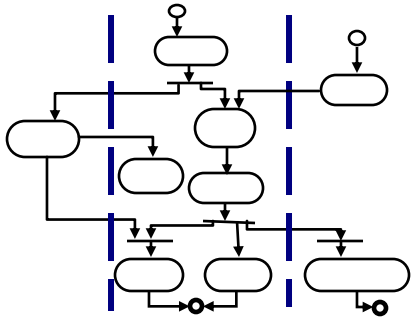
- Odzivi na dogodke
- Dinamično obnašanje
- Zaporedje dogodkov, dosegljivosti ipd.

Diagrami zaporedja



- Posamezni scenariji
- Interakcija med uporabniki in sistemom
- Zaporedja sporočil

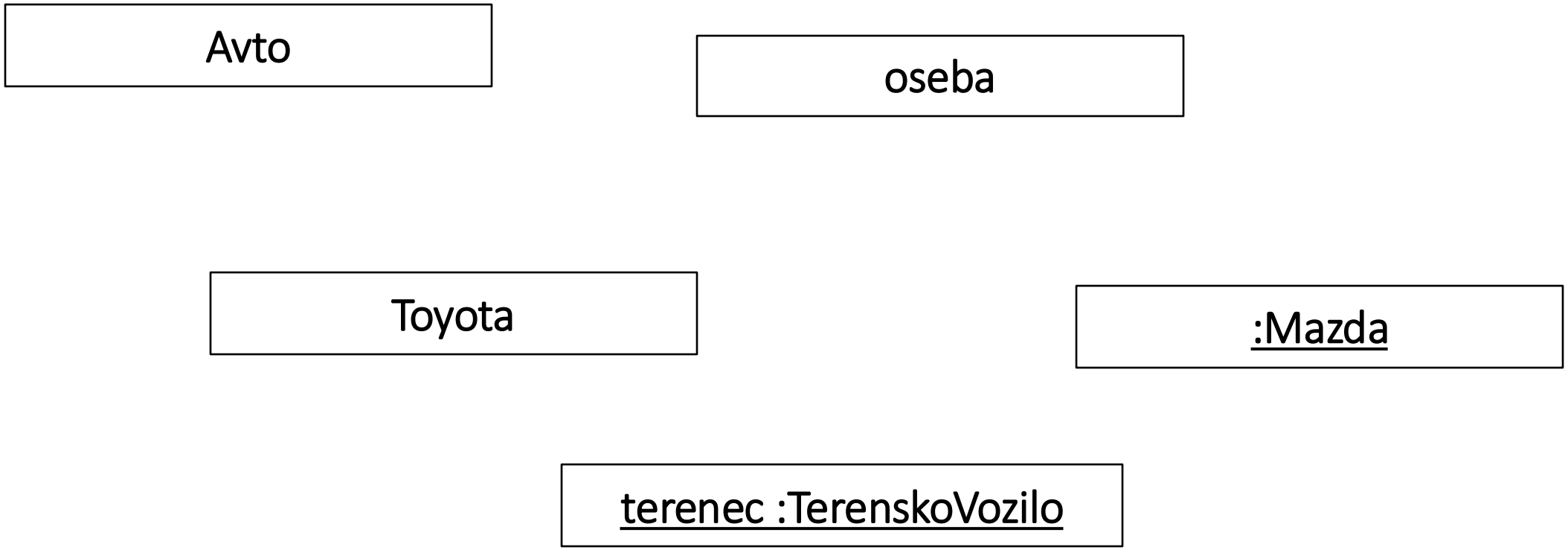
Diagrami aktivnosti



- Poslovni procesi
- Zaporedja
- Sočasnost in sinhronizacija
- Odvisnosti med nalogami

Razred

Koliko razredov je na prosojnici?

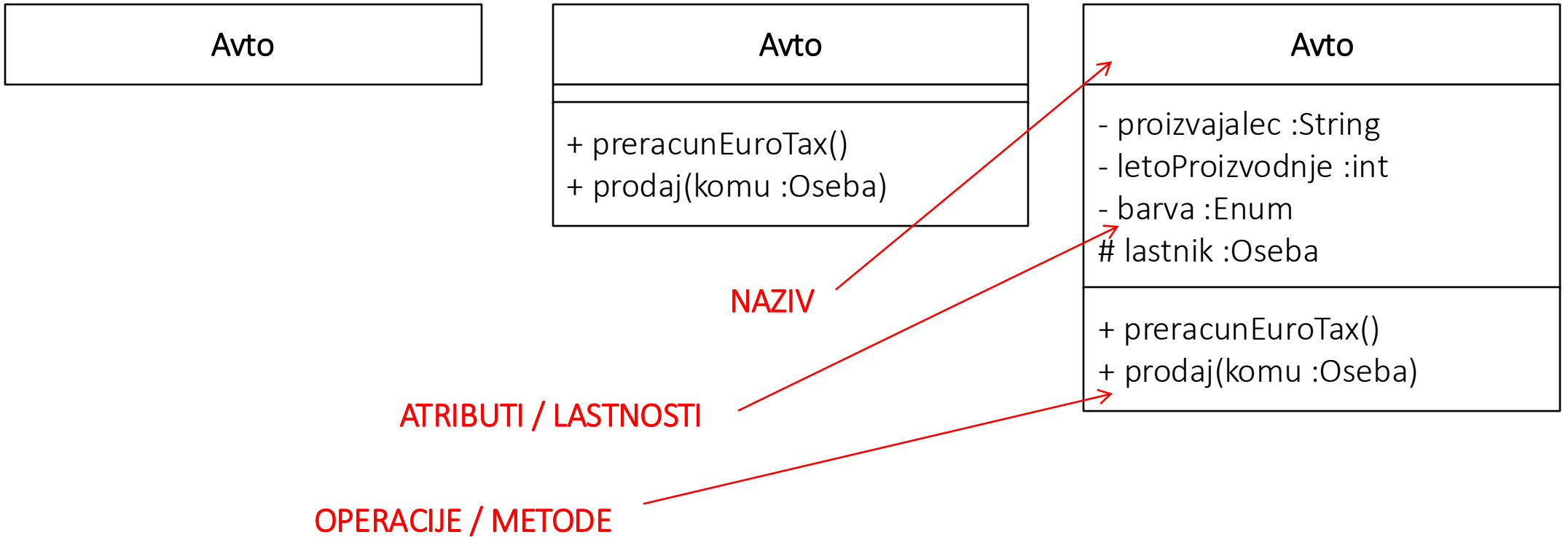


UML predstavitev razreda

Domenski pogled

Pogled na vmesnik

Podrobnosti



Gradniki razreda

Naziv

Italic – abstraktni razredi

:Underline - objekti

Atributi, Operacije

Možne sekcije **private** / **protected** / **public**

Ali oznaka neposredno pri elementu: **+, -, #**

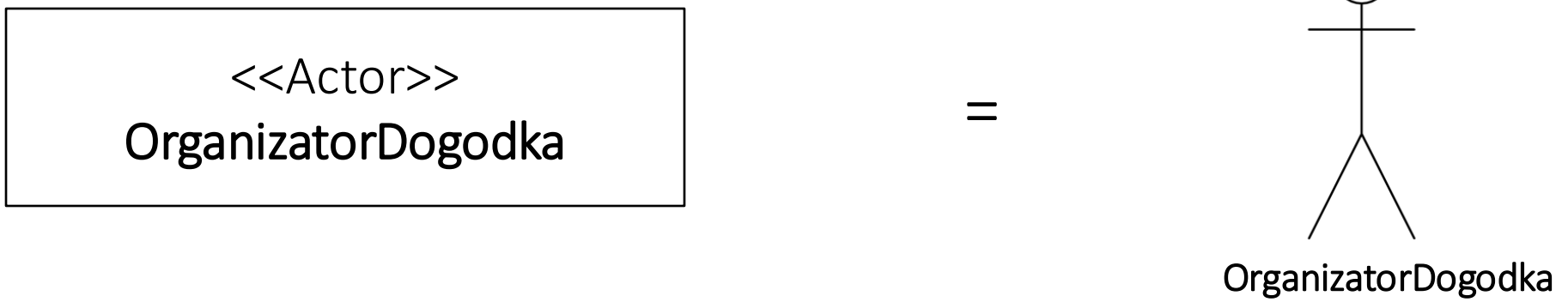
Stereotipi

Razširjanje standardnega besednjaka UML

Označevanje razredov / povezav s posebnim pomenom

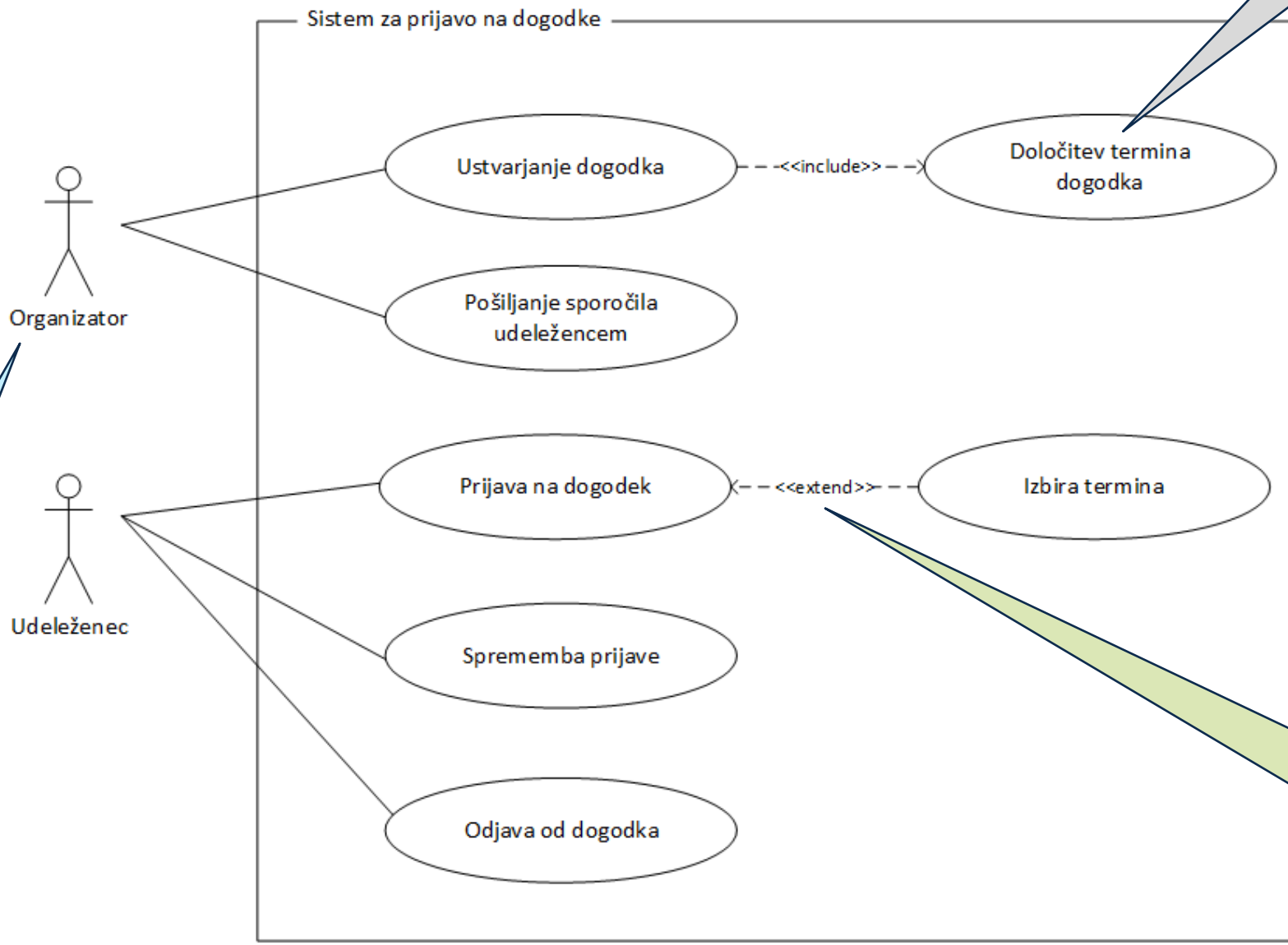
Razred z znanimi stereotipi lahko ima lastni grafično predstavitev

Nakej primerov: <<include>>, <<extend>>, <<UseCase>>, <<interface>> …



Kateri tip diagramov je na sliki? ☺

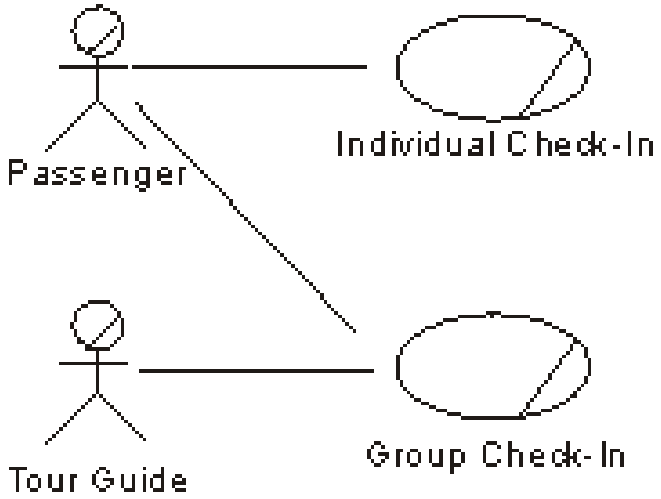
Funkcionalne zahteve!



Vloge – povezano tudi z omejitvami = nefunkcionalnimi zahtevami

Ne pretiravajmo!

Pazite: UML omogočat tudi poslovno modeliranje!



A passenger can either travel individually or with a group. When traveling with a group, a passenger is accompanied by a tour guide.

Poslovnega modeliranja NE naslavljamo tekom tega predmeta!

[IBM Rational Method Composer]

Povzetek diagramov primerov uporabe

- Meja sistema
  - Ne vedno
- Akter
  - Uporabnik
  - Tudi zunanji sistem
- Primer uporabe
  - Modelira funkcionalno zahtevo
- Asociacija
  - Akter -> Primer uporabe
- Odvisnost
  - Tipična stereotipa <<include>>, <<extend>>
- Generalizacija
  - Med akterji
  - Med primeri uporabe
  - Beremo: 1 IS-A 1

Diagrami paketov

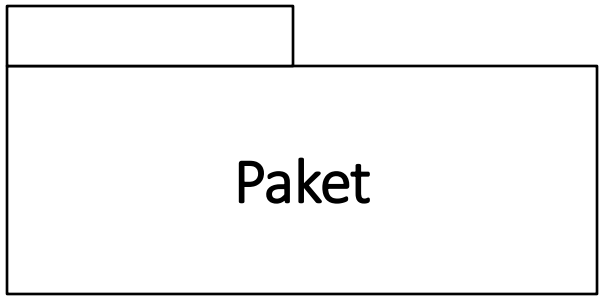
Paket je logična povezava med pomensko povezanimi elementi (razredi, komponentami, primeri uporabe ipd.)

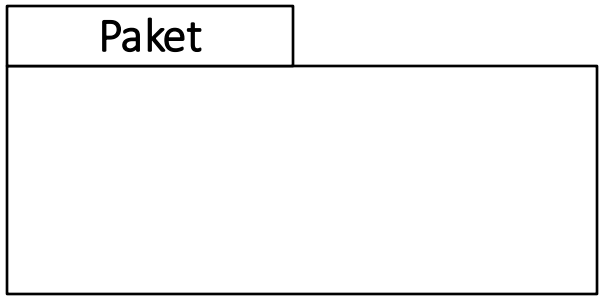
- V namestitvenih paketih so gradniki razdeljeni glede na izvajanje

UML omogoča tudi gnezdenje paketov

Tudi med paketi tipično uporabljamo povezave

- Asociacije
- Odvisnosti



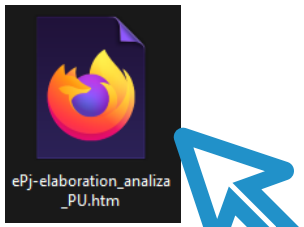


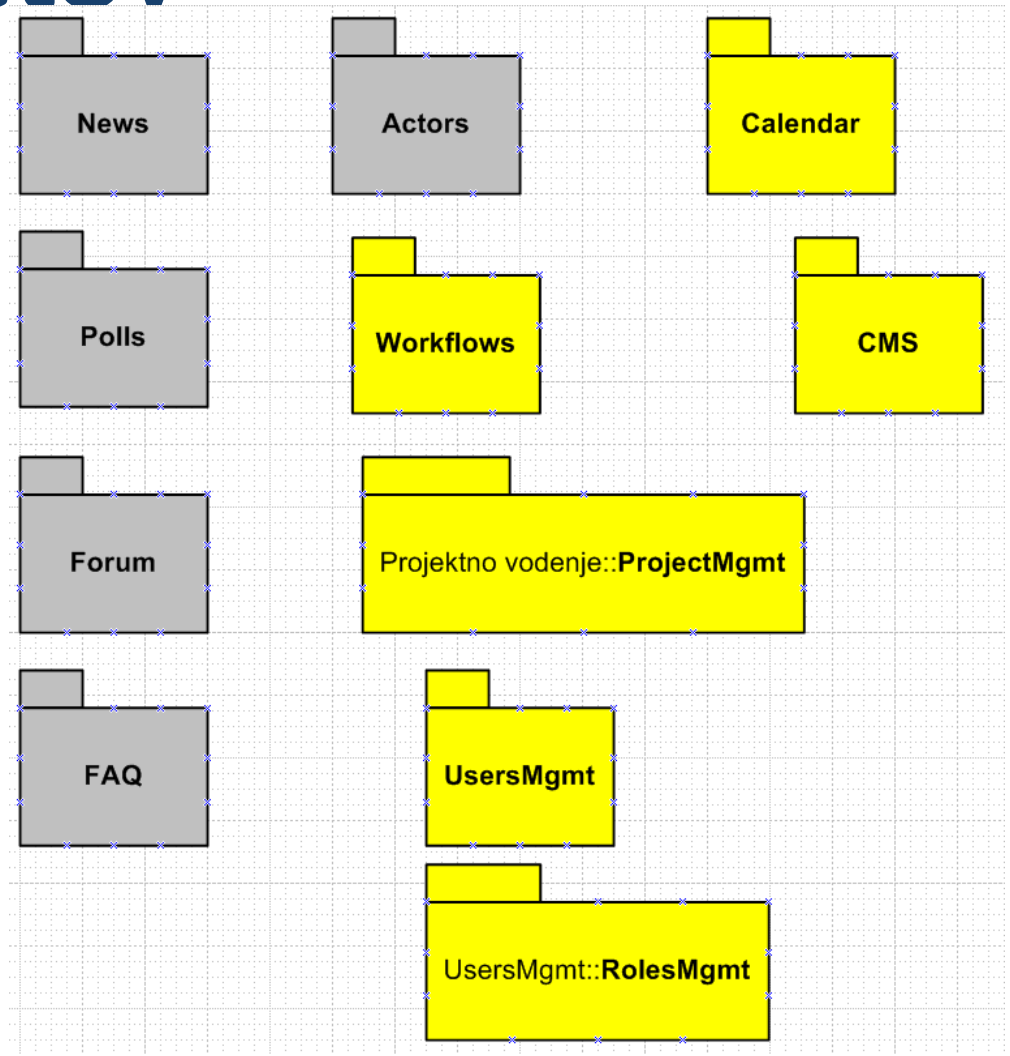
Uporaba diagramov paketov

UML diagrami paketov so uporabni pri smiselnem združevanju (npr. sorodnih primerov uporabe)

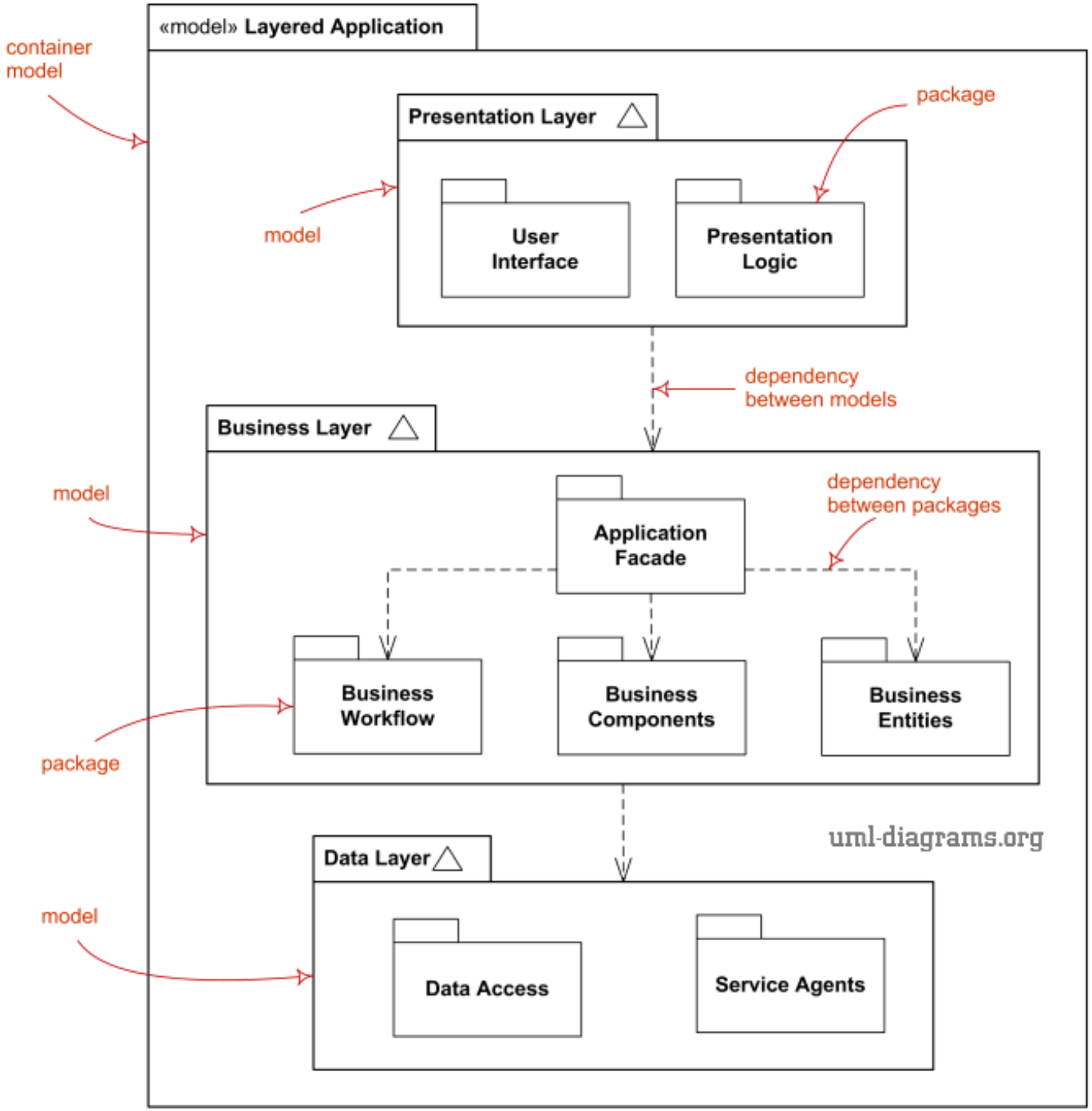
Vsak paket lahko vsebuje tudi diagram (npr. primerov uporabe)

Pazite: Enaki primeri uporabe, akterji, razredi… lahko nastopajo v več paketih!



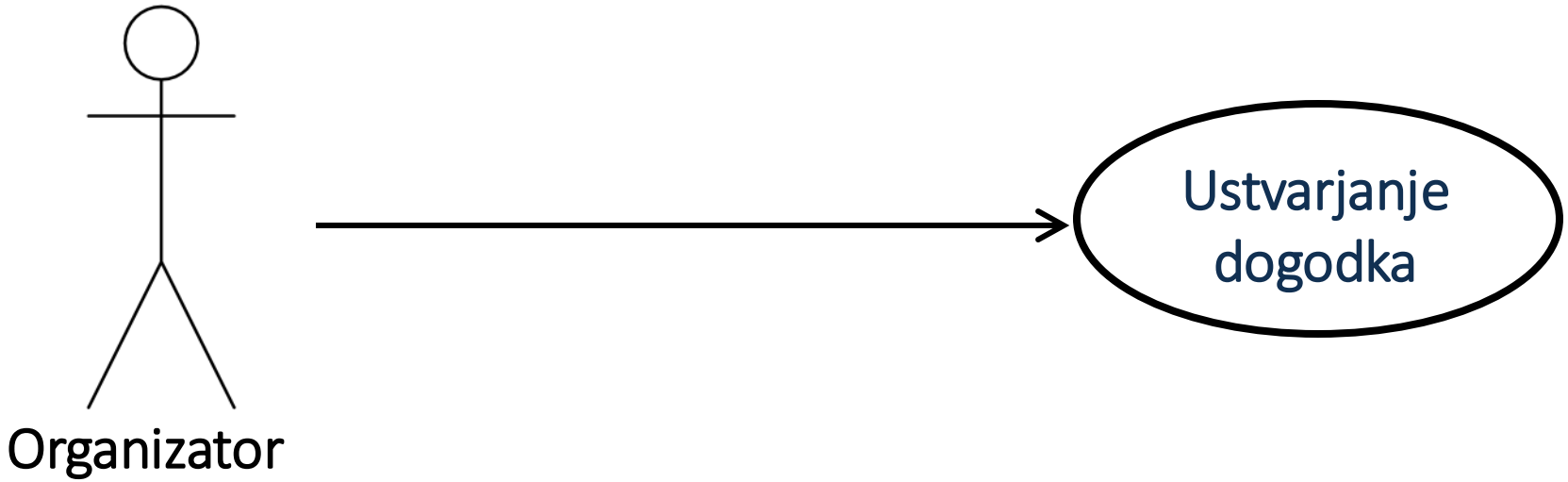


https://www.uml-diagrams.org/package-diagrams-overview.html



Specifikacija primerov uporabe

To še ni načrtovanje!



Je to dovolj?

Opis primera uporabe: Kombiniranje besedila in diagramov

Kombiniramo besedila in diagrame
- Pogosto za prikaz statične in dinamične slike

Ločimo:
- Pozitiven scenarij (koraki 1, 2, 3, 4, 5…)
- Alternative (3.a, 3.b …)

Primer besedilnega opisa:
- Naziv primera uporabe
- Vloge
- Splošen opis
- Pozitiven scenarij
- Alternative
- Posebne zahteve
- Predpogoji
- Popogoji

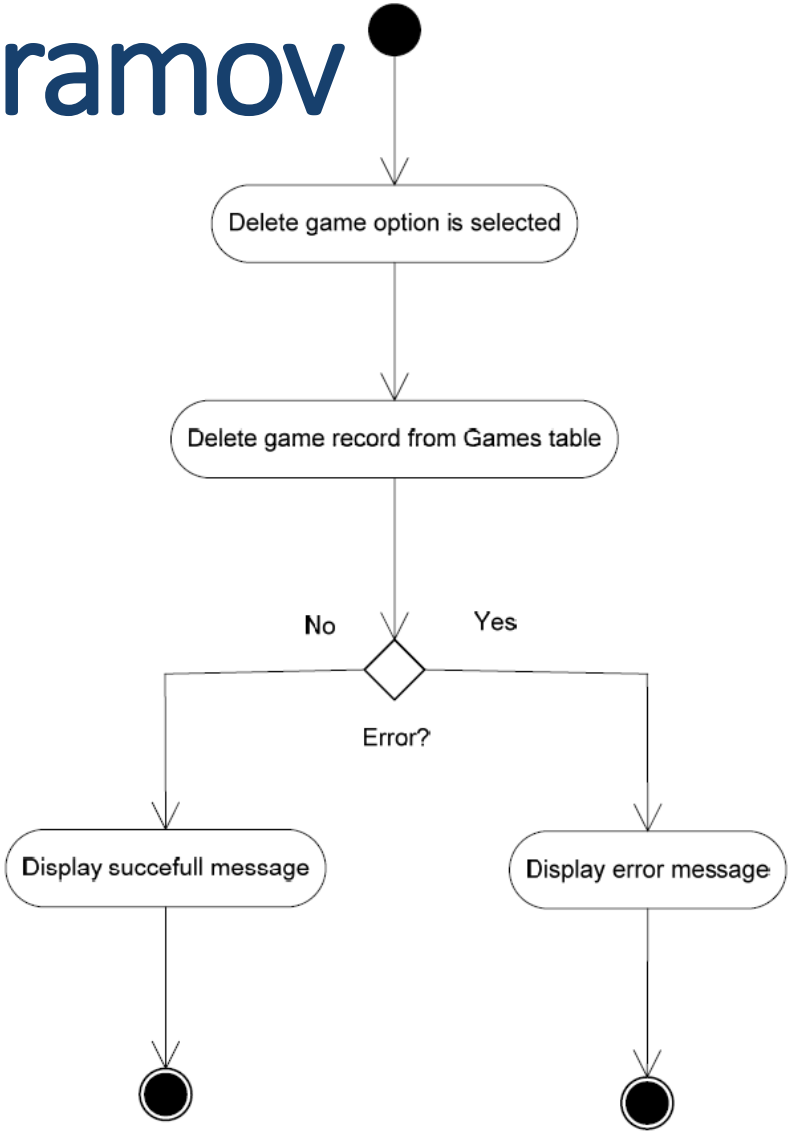


Figure 6: AD Remove game

Konkreten *(ne najboljši)* primer opisa PU 1/2

Name: TakeQuiz

Actor(s): QuizTaker

Flow of events:

1. QuizTaker connects to the Quiz server.
2. Quiz server checks whether student is already authenticated and transfer to Sidecar for authentication if necessary.
3. QuizTaker selects a quiz from a list of options.
4. QuizTaker repeatedly selects a question and either types in a solution, attaches a file with a solution, edits a solution or attaches a replacement file.

Konkreten *(ne najboljši)* primer opisa PU 1/2

Flow of events (continued):

5. QuizTaker either submits completed quiz or saves current state.
6. If a completed quiz is submitted, Quiz server checks that all questions have been attempted and either sends acknowledgement to QuizTaker, or saves current state and notifies QuizTaker of incomplete submission.
7. QuizTaker logs out.

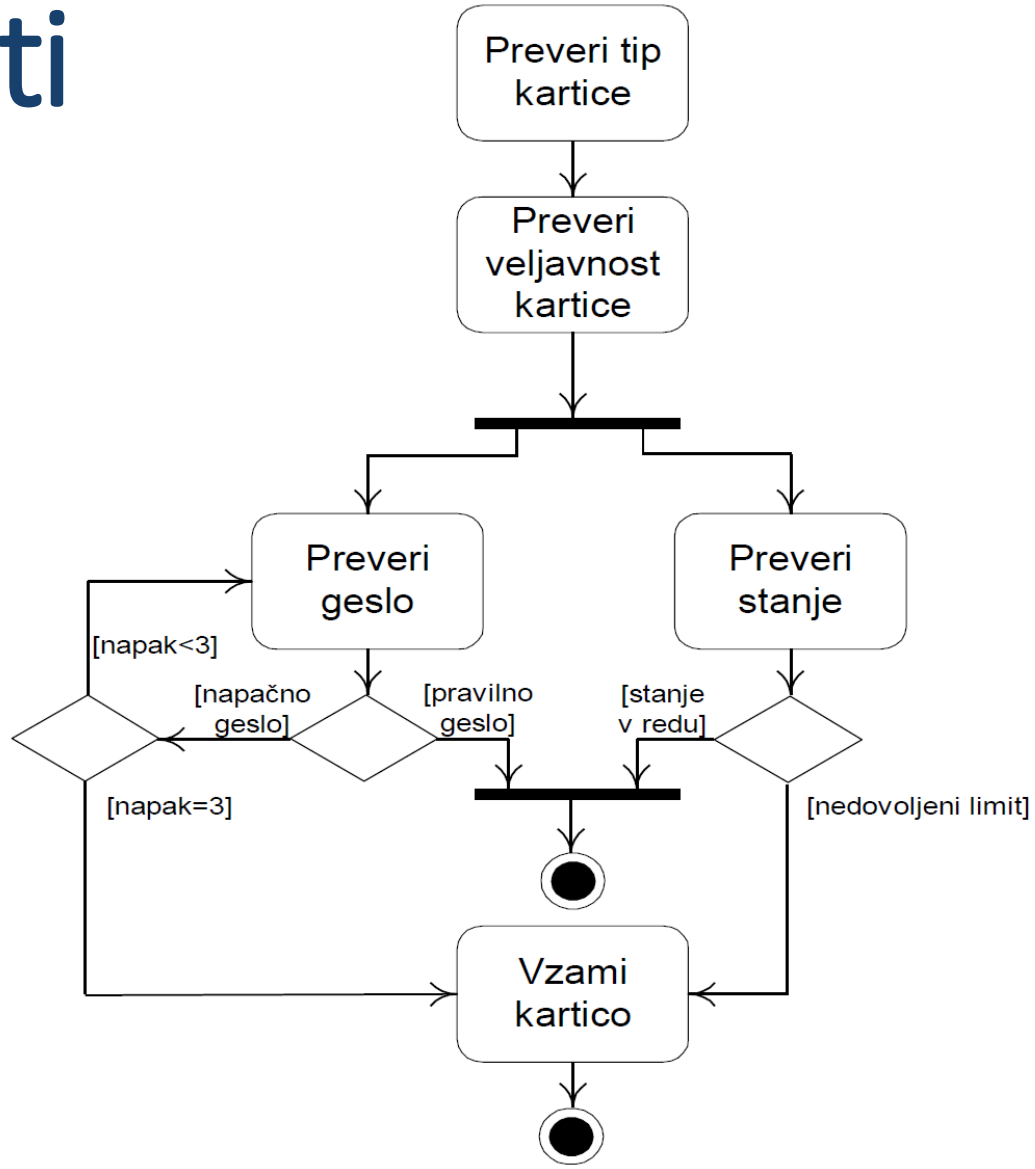
Entry conditions:

1. QuizTaker must have user ID.
2. Computing requirements: CIT supported browser and Sidecar

UML diagrami aktivnosti

Celoten diagram aktivnosti se nanaša na:

- razred,
- implementacijo operacij,
- primer uporabe
- …



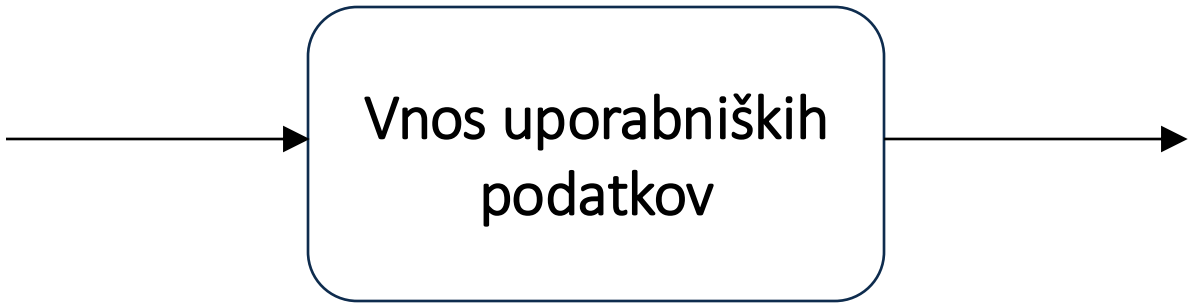
“Aktivnost”

Je stanje z notranjo aktivnostjo in vsaj eno izhodno transakcijo

Aktivnost naj ne bi imela notranjih prehodov ali izhodnih prehodov osnovanih na eksplicitnih dogodkih

v tem primeru uporabimo diagrame stanj

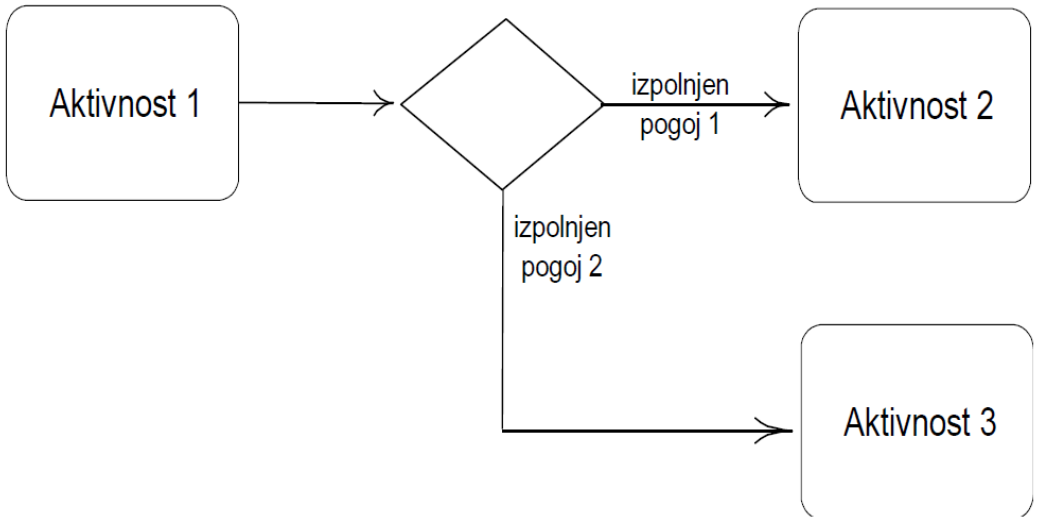
Normalna uporaba je modeliranje korakov pri izvajanju npr. algoritmov



“Vejitev”

Diagram aktivnosti izraža odločitev takrat, ko uporabimo pogoje za različne možne prehode.

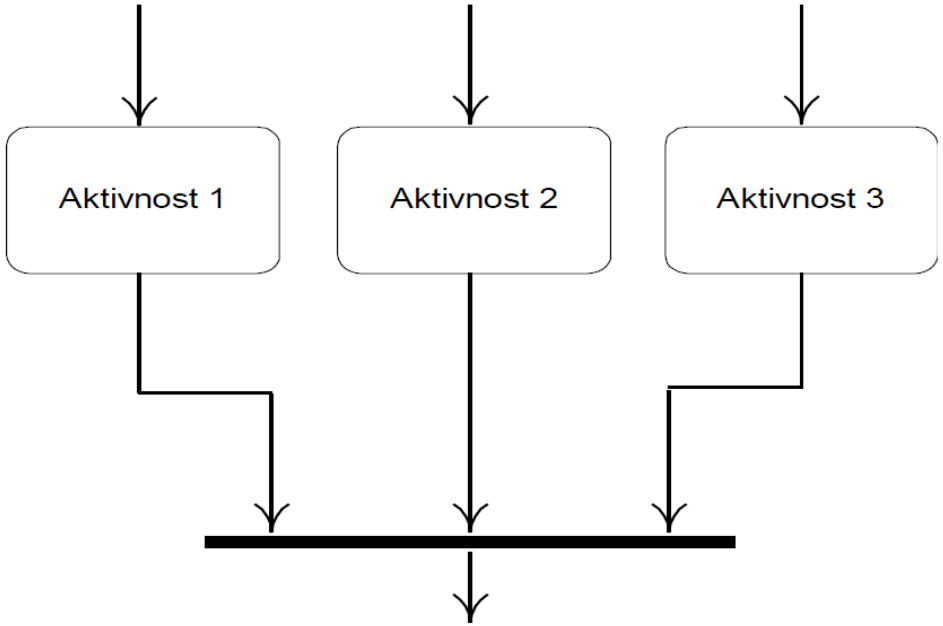
Prehodi so odvisni od logične vrednosti pogoja pripadajočega objekta.



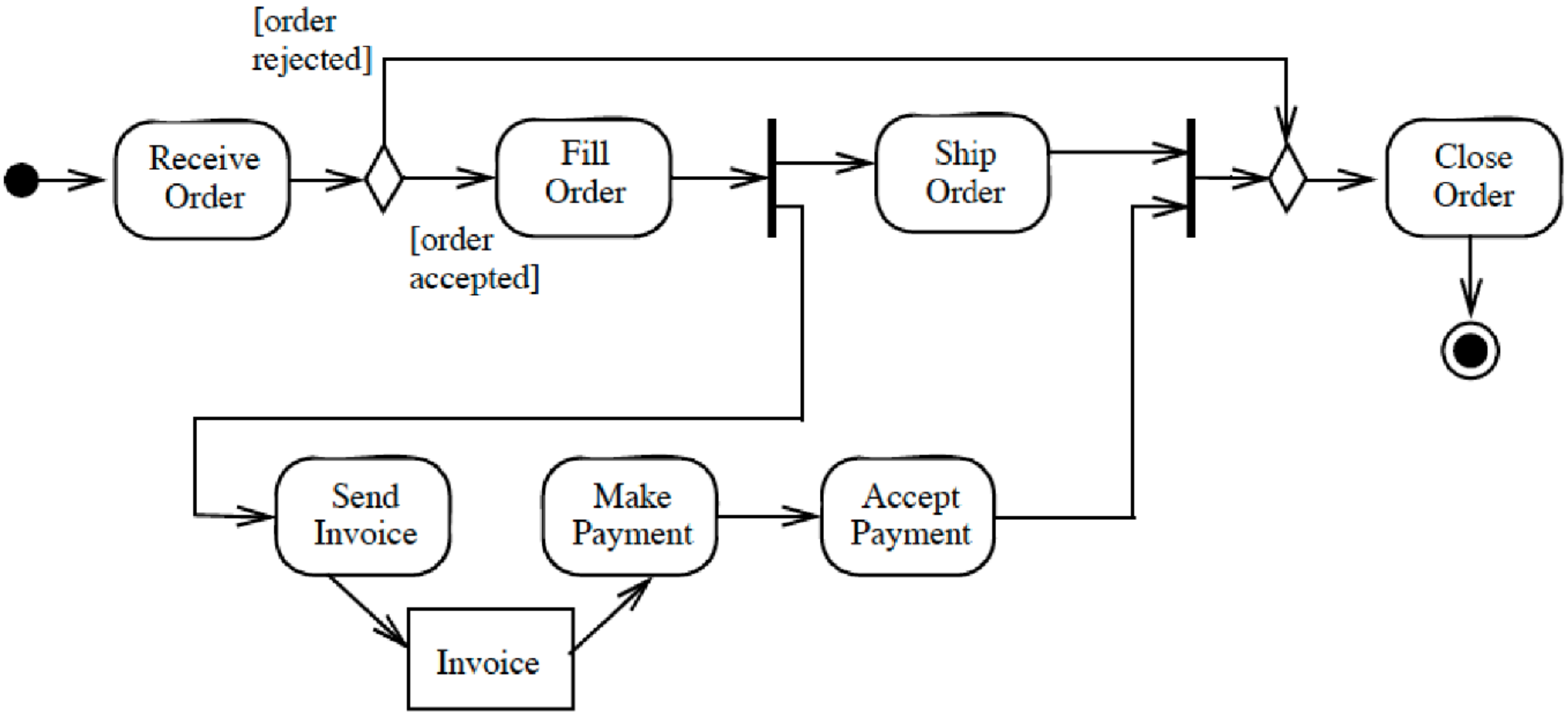
Sinhronizacija vzporednih aktivnosti

Uporabimo jo kot simbol sinhronizacije večih aktivnosti, ki potekajo vzporedno in se morajo pred pričetkom novih aktivnosti izvesti do konca ter se sinhronizirati.

Več aktivnosti se lahko izvaja paralelno, kar pomeni, da vrstni red aktivnosti ni pomemben.



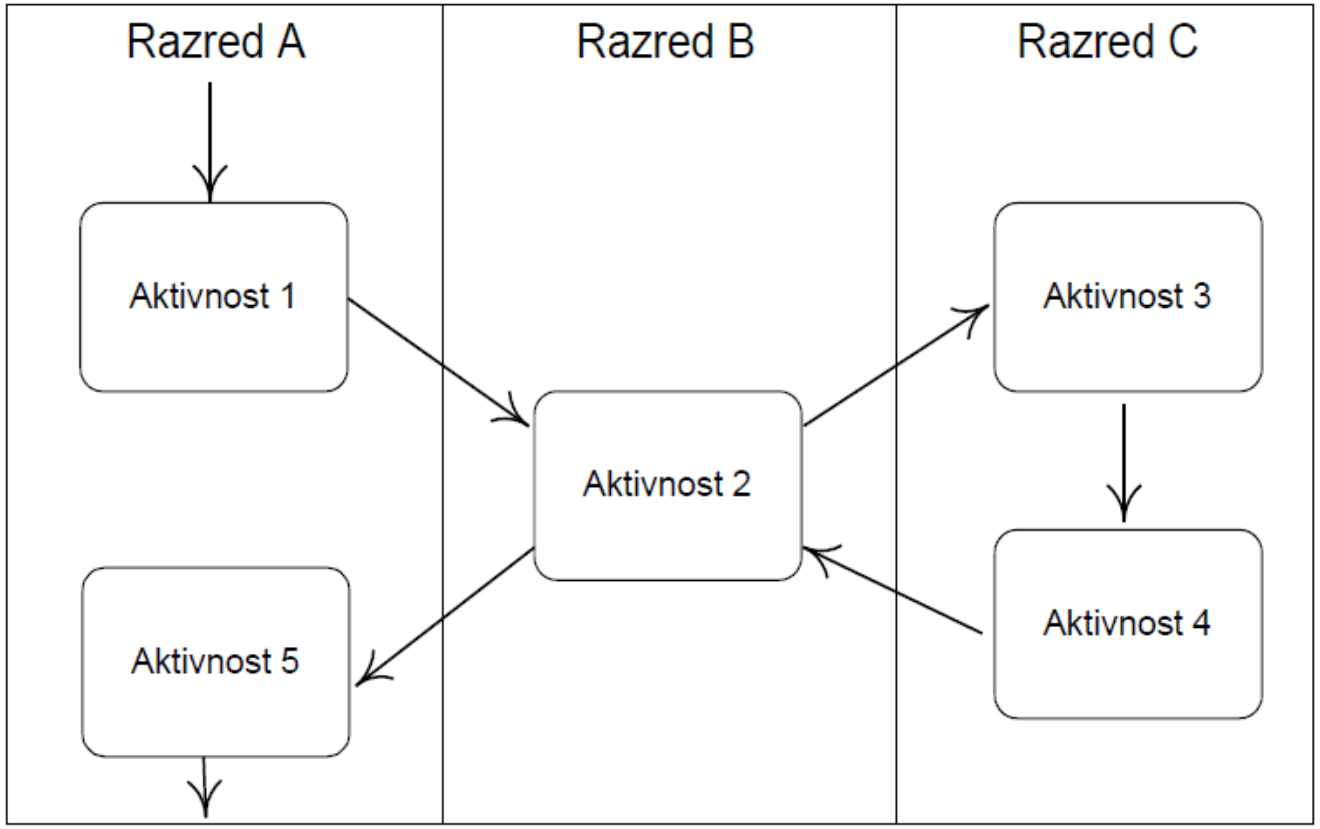
Primer


[order rejected]
Receive Order
Fill Order
[order accepted]
Ship Order
Close Order
Send Invoice
Make Payment
Accept Payment
Invoice

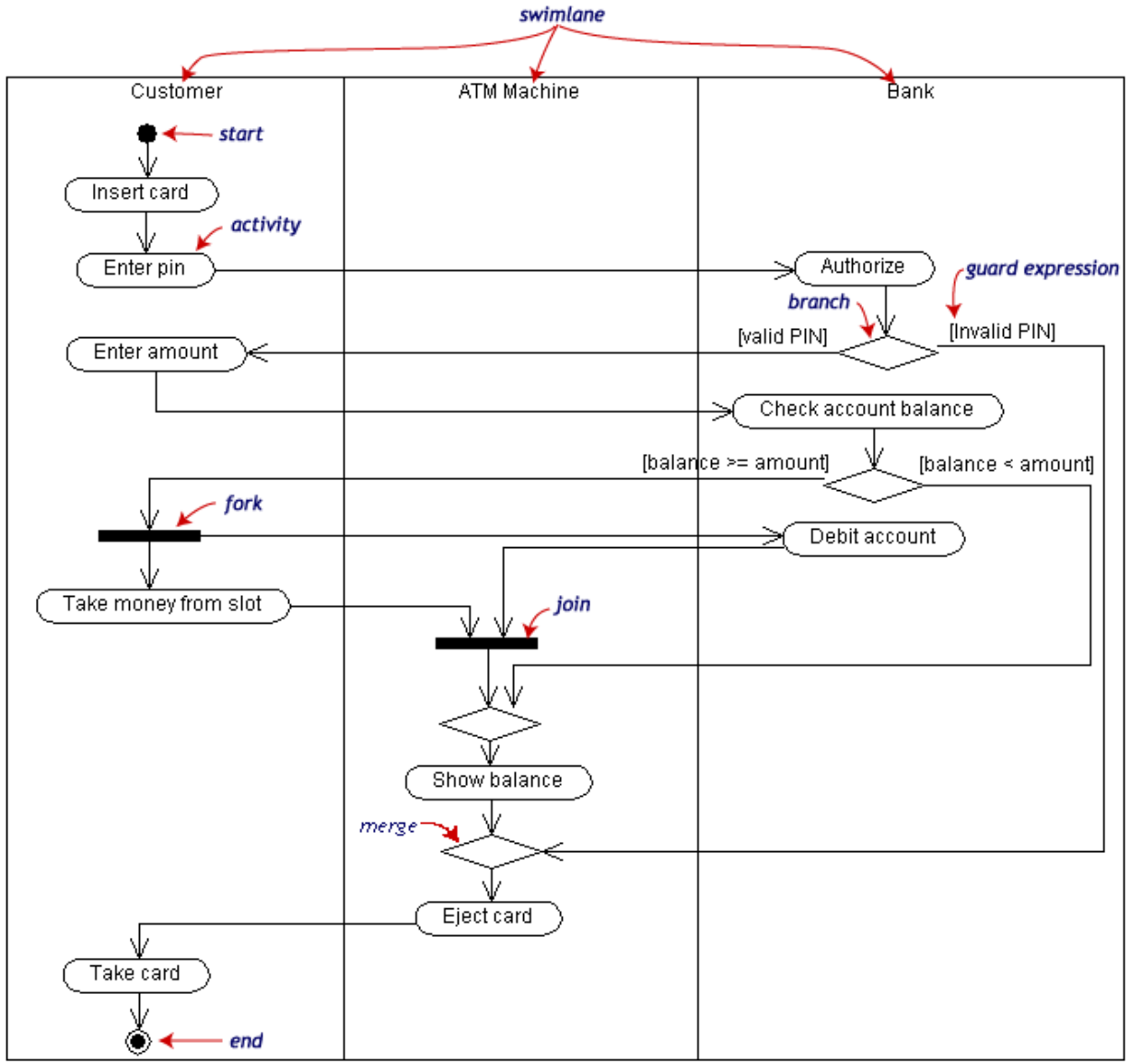
Uporaba stez

S stezami uredimo aktivnosti v vertikalna območja, ločena s črto.

Vsako območje predstavlja odgovornosti posameznega razreda ali osebe.



Primer diagrama aktivnosti



Konkreten primer

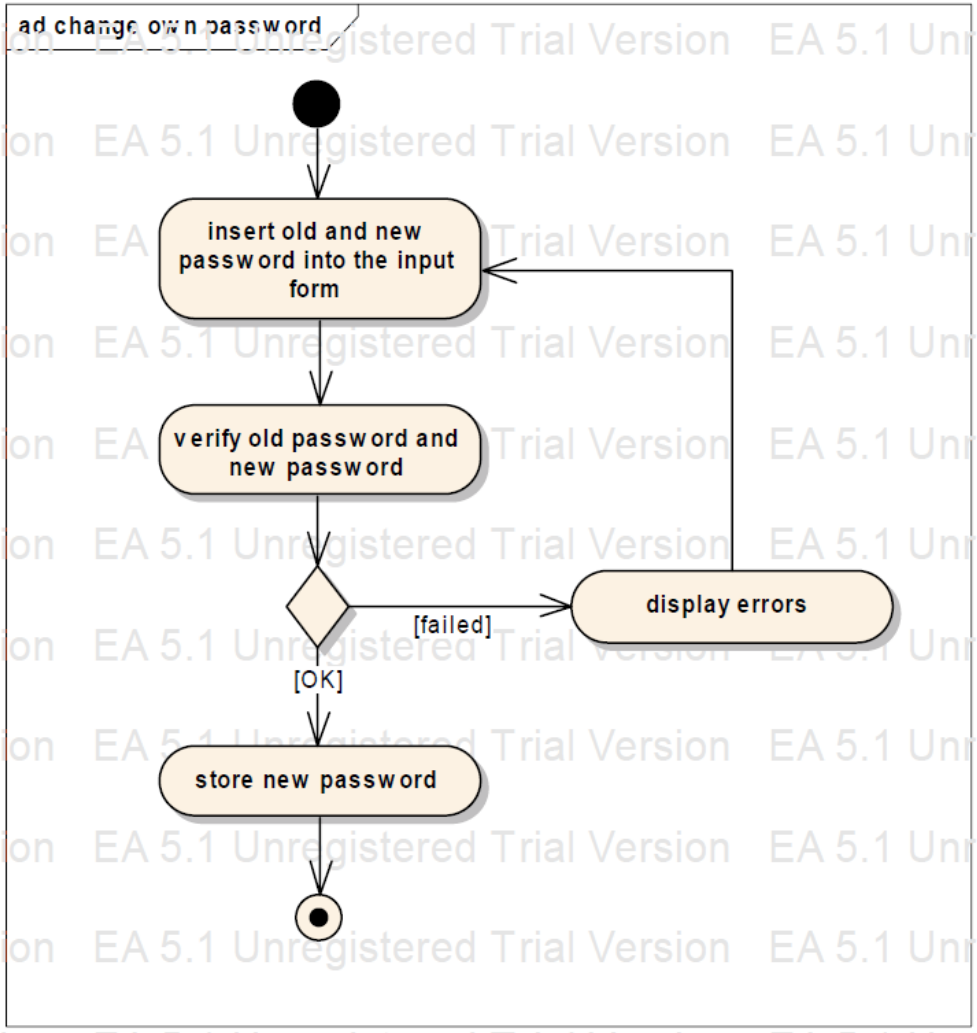


Figure 4 AD change own password

Pazite – ne zamenjujmo z diagrami stanj!