

Specifikacija softverskih sistema

Predavanje br. 1 – Uvod

Gordana Milosavljević

Katedra za informatiku, FTN, Novi Sad
2022.

Agenda

- Osnovni podaci o predmetu
- Ukratko o UML-u
- Uvod u analizu zahteva

Nastavu izvode

- Predavanja:

Gordana Milosavljević, milosavljevic.g@gmail.com

- Vežbe:

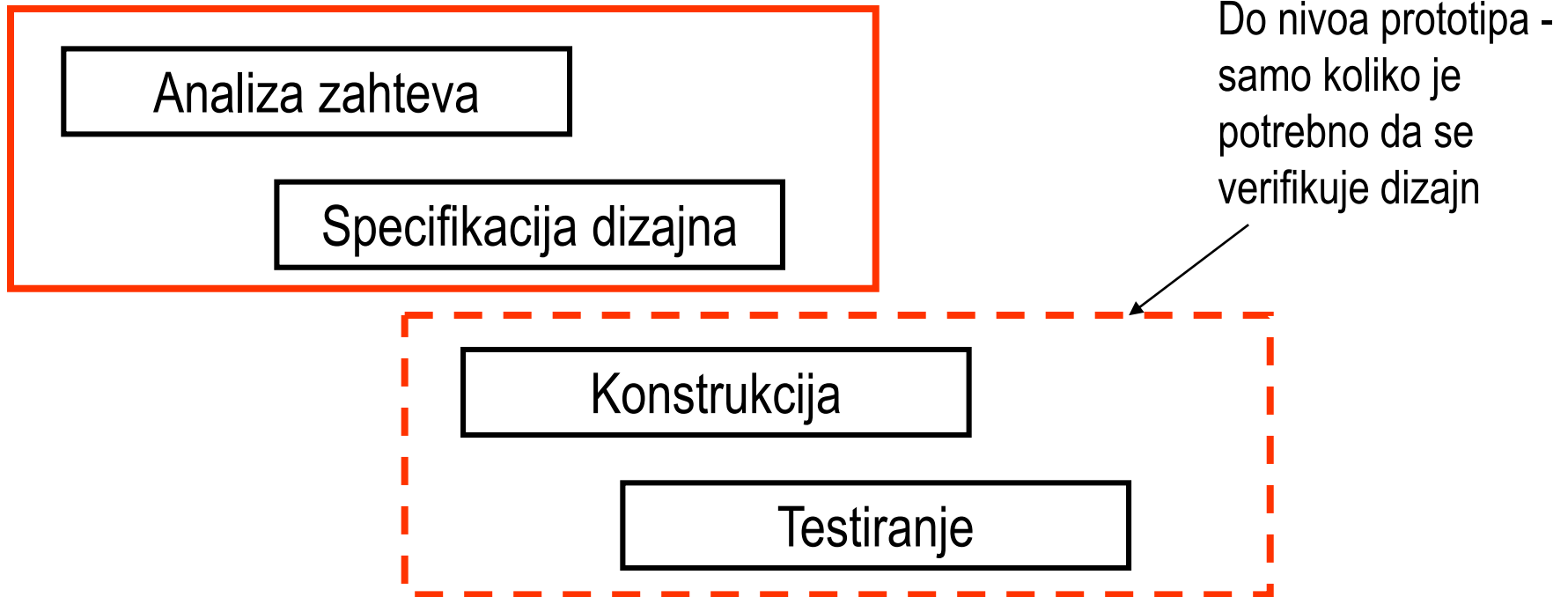
Mina Medić, mina.medic@uns.ac.rs

Filip Volarić, filip.kresa@gmail.com

Cilj predmeta

- Osposobiti studente za analizu i modelovanje realnih sistema
 - UML (*Unified Modeling Language*)
 - Alati za modelovanje (PowerDesigner, MagicDraw)

Fokus predmeta



Koju fazu je po vašem mišljenju najteže sprovesti?

Metodologija koju ćemo koristiti

- Na vežbama ćemo se bazirati na iterativno-inkrementalnom načinu rada i principima agilnog modelovanja:
 “PLANIRAJ MALO,
 ANALIZIRAJ MALO,
 PROJEKTUJ MALO,
 IMPLEMENTIRAJ MALO!”
- Pogodno za sisteme sa velikim stepenom neodređenosti
- Napomena:
 - Naučeno se može primeniti u okviru bilo koje metodologije, s obzirom da sve poseduju navedene faze razvoja

Podaci o predmetu

- Način polaganja:
 - kolokvijum: 50 poena
 - Moguće je polaganje u toku semestra i u prva dva sledeća termina (januarski i aprilski rok)
 - Kolokvijum služi da proverimo poznavanje i realnu primenu naučenih UML dijagrama
 - Projekat: 50 poena
 - dodatni bonusi: aktivnost, kvalitet projekta...

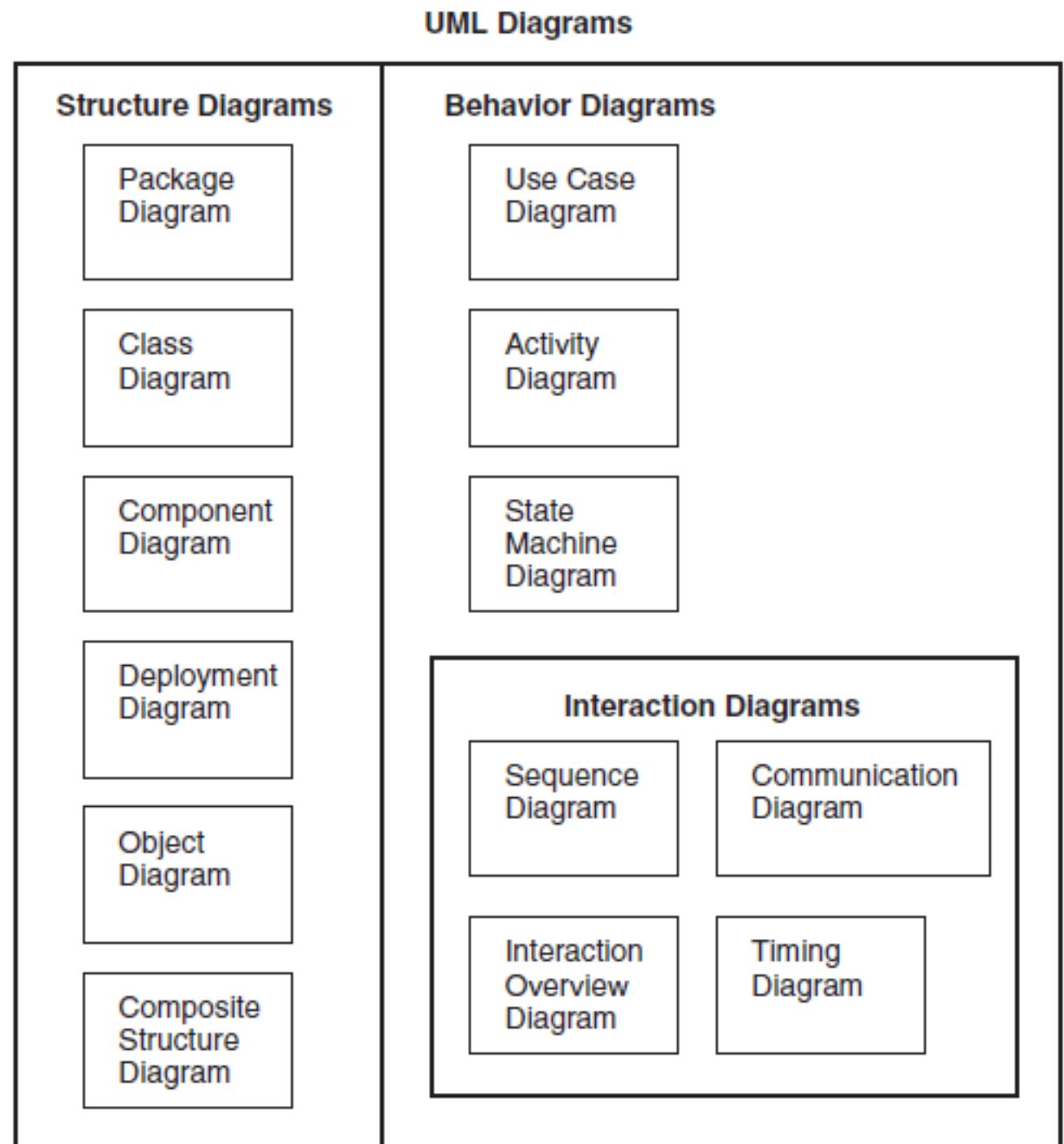
Projekat

- Platforma:
 - PowerDesigner ili MagicDraw
 - java ili python (po želji)
 - Svn ili git
- Način rada:
 - u timovima od 3-4 člana

Udžbenici

- Gordana Milosavljević, [Uvod u modelovanje softvera](#)
- Scott W. Ambler, [The Object Primer: Agile Model-Driven Development with UML 2.0](#), Cambridge University Press, 2004.
- James Rumbaugh, Ivar Jacobson, Grady Booch, [The Unified Modeling Language Reference Manual](#), Second Edition, Addison-Wesley, 2004.

UML – Unified Modeling Language

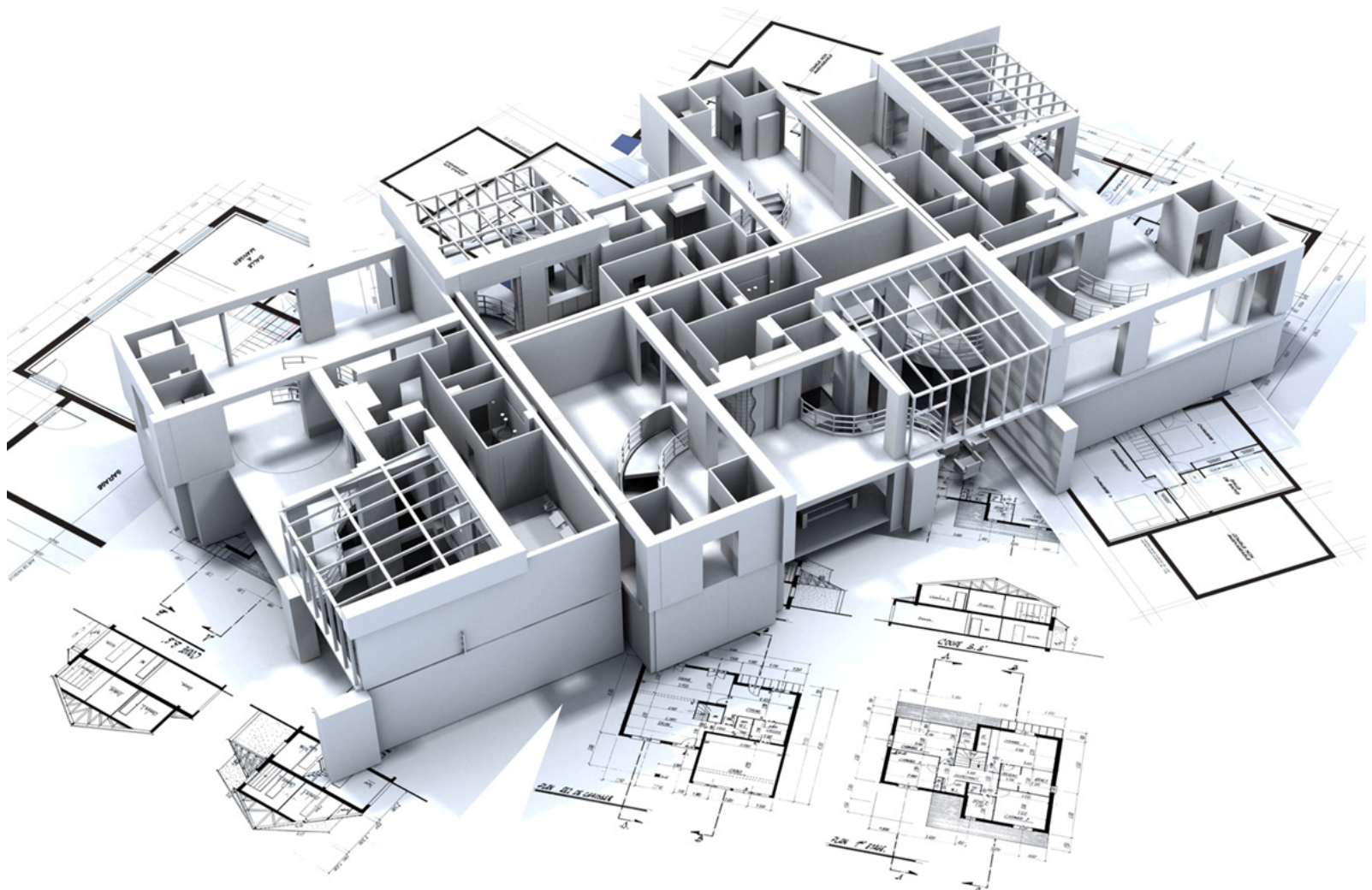


UML - Unified Modeling Language

- Objedinjeni jezik za modelovanje
 - Inicijalno je kreiran sa namerom da objedini i standardizuje različite notacije koje su korišćene za modelovanje
 - Prvu verziju su razvili Grady Booch, Ivar Jacobson i James Rumbaugh u okviru korporacije Rational Software 1995. godine
 - Od 1997. godine je njegov dalji razvoj i održavanje preuzela Object Management Group (OMG): www.omg.org/
 - Sve verzije UML-a se nalaze ovde: www.omg.org/spec/UML/
 - Poslednja verzija je UML 2.5 iz juna 2015.

Model

Model je pojednostavljena predstava realnog sistema kreirana sa određenim ciljem



Zašto modelujemo?

- Radi boljeg razumevanja sistema koji se implementira
- Radi lakše komunikacije
- Radi ispitivanja budućih karakteristika sistema koji se gradi
- Radi specificiranja implementacije

Vrste modela

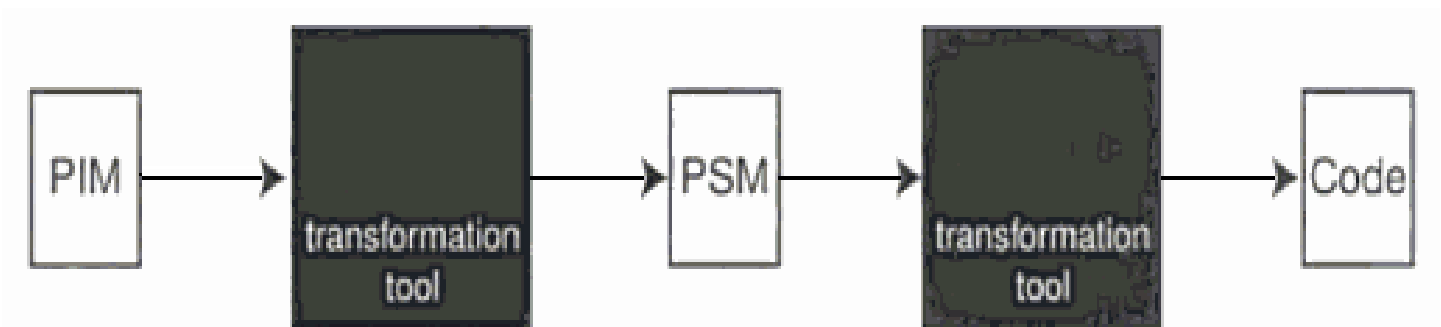
- Deskriptivni
 - služe za bolje razumevanje pojava, komunikaciju i predviđanje pojava i ponašanja
- Preskriptivni
 - služe kao specifikacija za izgradnju krajnjeg sistema

Izvršivi modeli

- Modeli zapisani korišćenjem dobro definisanog jezika koji podržava automatsku interpretaciju ili generisanje koda od strane računara.
- Koriste se u okviru Inženjerstva softvera vođenog modelima (*Model Driven Software Engineering* - MDSE)

MDSE – Model Driven Software Engineering

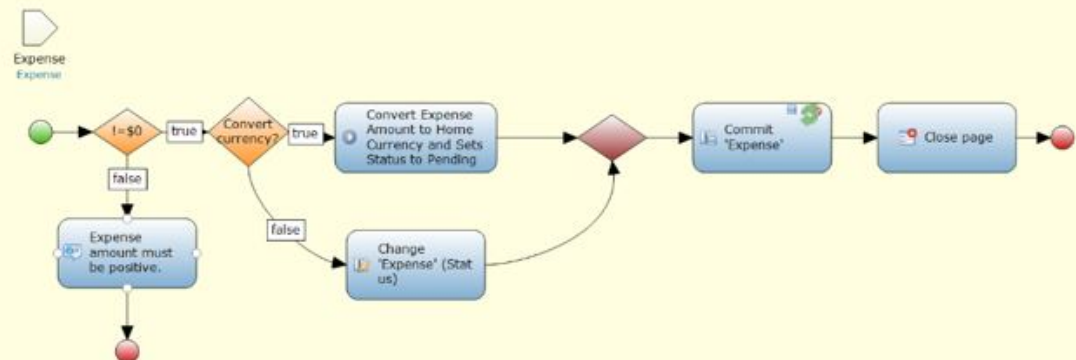
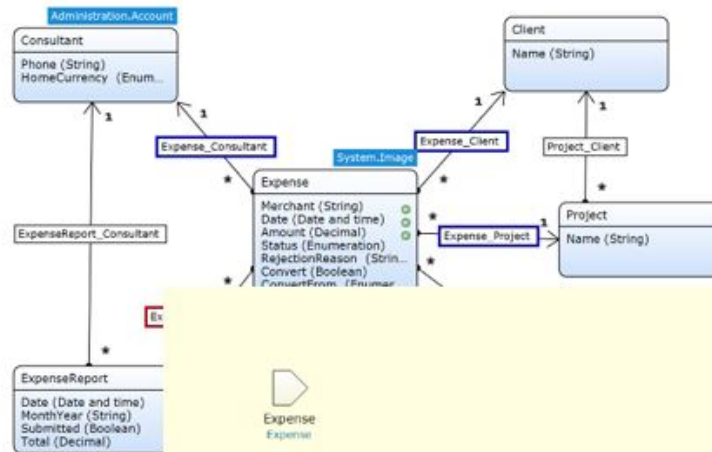
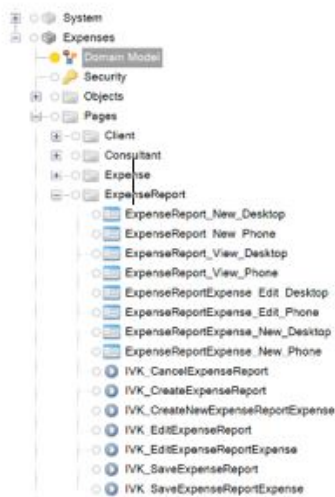
- MDSD – Model Driven Software Development
 - DSL – Domain Specific Languages, jezici specifični za domen na osnovu kojih se generiše kod
- MDA – Model Driven Architecture
 - Modeli nezavisni od platforme (*Platform Independent Model* – PIM)
 - Modeli zavisni od platforme (*Platform Specific Model* – PSM)



Low Code platforme

- Low Code razvojna okruženja
 - obično grafička okruženja
 - izbegava se klasično programiranje
 - akcenat je na kreiranju grafičke specifikacije za različite aspekte sistema
 - automatsko generisanje koda na osnovu specifikacije ili interpretiranje specifikacije
- Implementiraju MDSE principe
- Mendix
- Cuba
-

Mendix – primer modela



Mendix – dizajn ekrana

The screenshot displays the Mendix IDE interface with a dashboard design for 'ASSET Manager'. The dashboard is titled 'Dashboard' and features a 'Tasks' section on the left and a 'CONTAINER' section on the right.

Tasks Section:

- Alt: TaskTitle**
Alt: TaskMessage
Due 3/10/2018
- Analyse the Alert**
Temperature is too high
Due 3/10/2018
- Analyse the Alert**
Temperature is too high
Due 3/10/2018
- Analyse the Alert**
Temperature is too high
Due 3/10/2018
- Analyse the Alert**
Temperature is too high
Due 3/10/2018
- Analyse the Alert**
Temperature is too high
Due 3/10/2018

CONTAINER Section:

Analyse the Alert

Priority:

Due date:

Performer:

Assigned To:

Alert Description:

Alert date:

States:

Meter:

Open meter:

Notes:

Properties Panel:

Properties

ATTRIBUTE

Name:

Approved:

Documentation:

Type:

Toolbox:

Layout

- Layout Grid
- Container
- Scroll Container

Grids

- Data grid
- Listview
- Template Grid

Form Elements

- Dataview
- Textbox
- Textarea
- Dropdown
- Datapicker
- Label

Typography

- Heading
- Paragraph
- Text

Buttons

- Action

Media

- Static Image
- Dynamic Image

Entity Diagrams:

Employee

Attribute	Type
FullName	String
Phone	String
Email	String
Submitted	String

Conversion

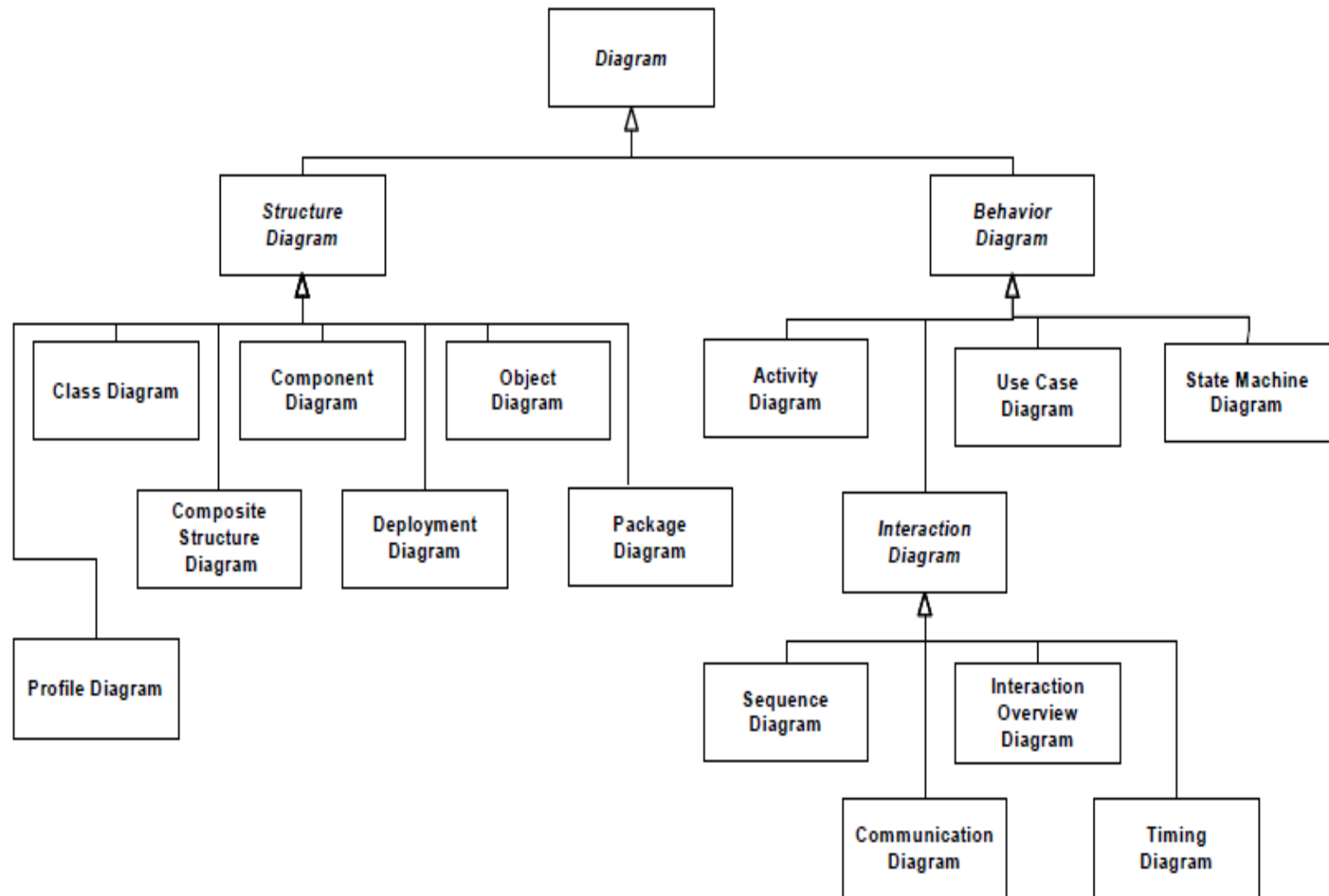
Attribute	Type
FromAmount	String
ToAmount	String
ToAmountFormatted	String

Currency

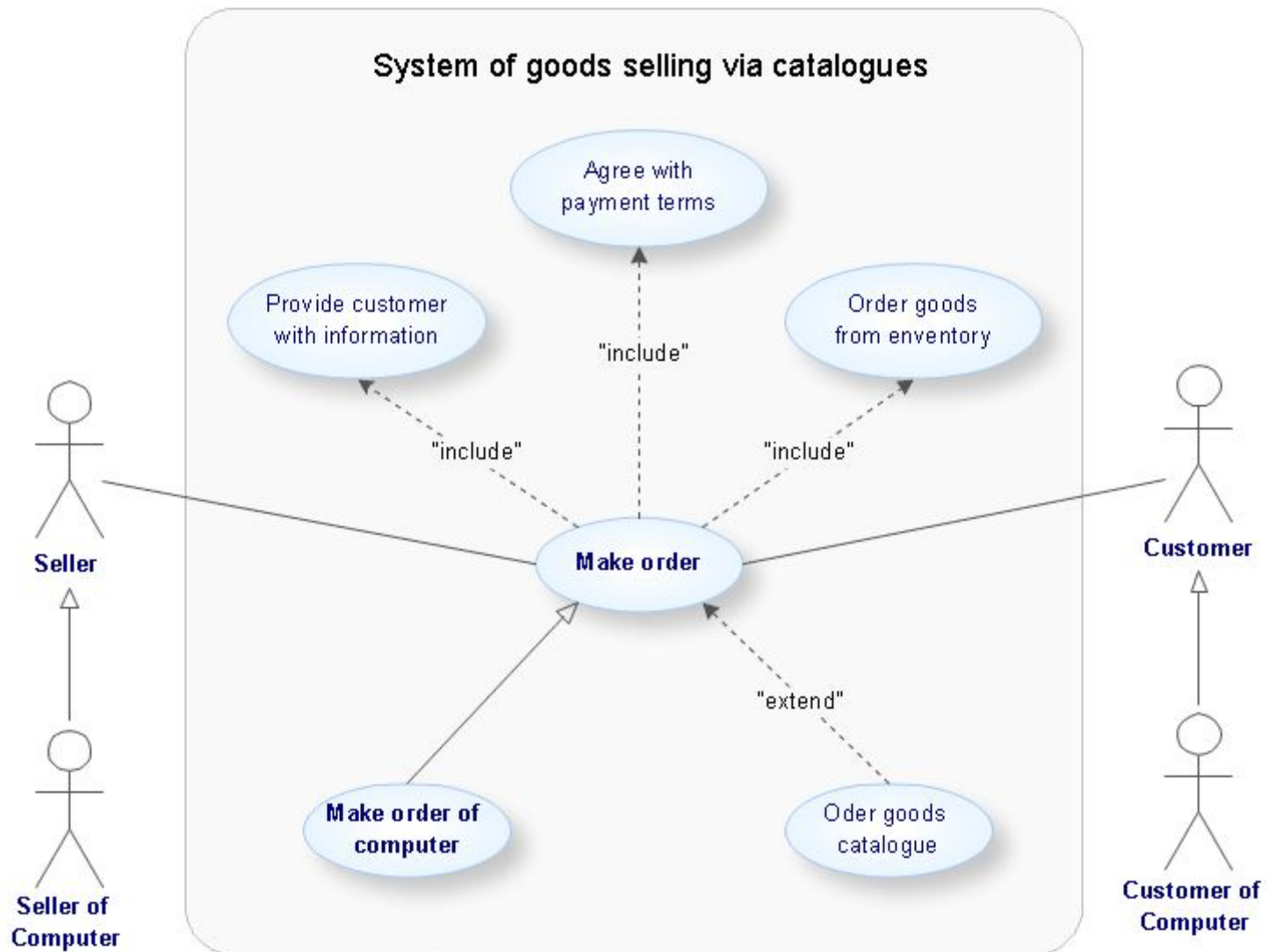
Attribute	Type
Code	String
Character	String
Name	String

Podela UML dijagrama

- Za opis strukture (statičkih aspekata sistema)
- Za opis ponašanja (dinamičkih aspekata sistema)

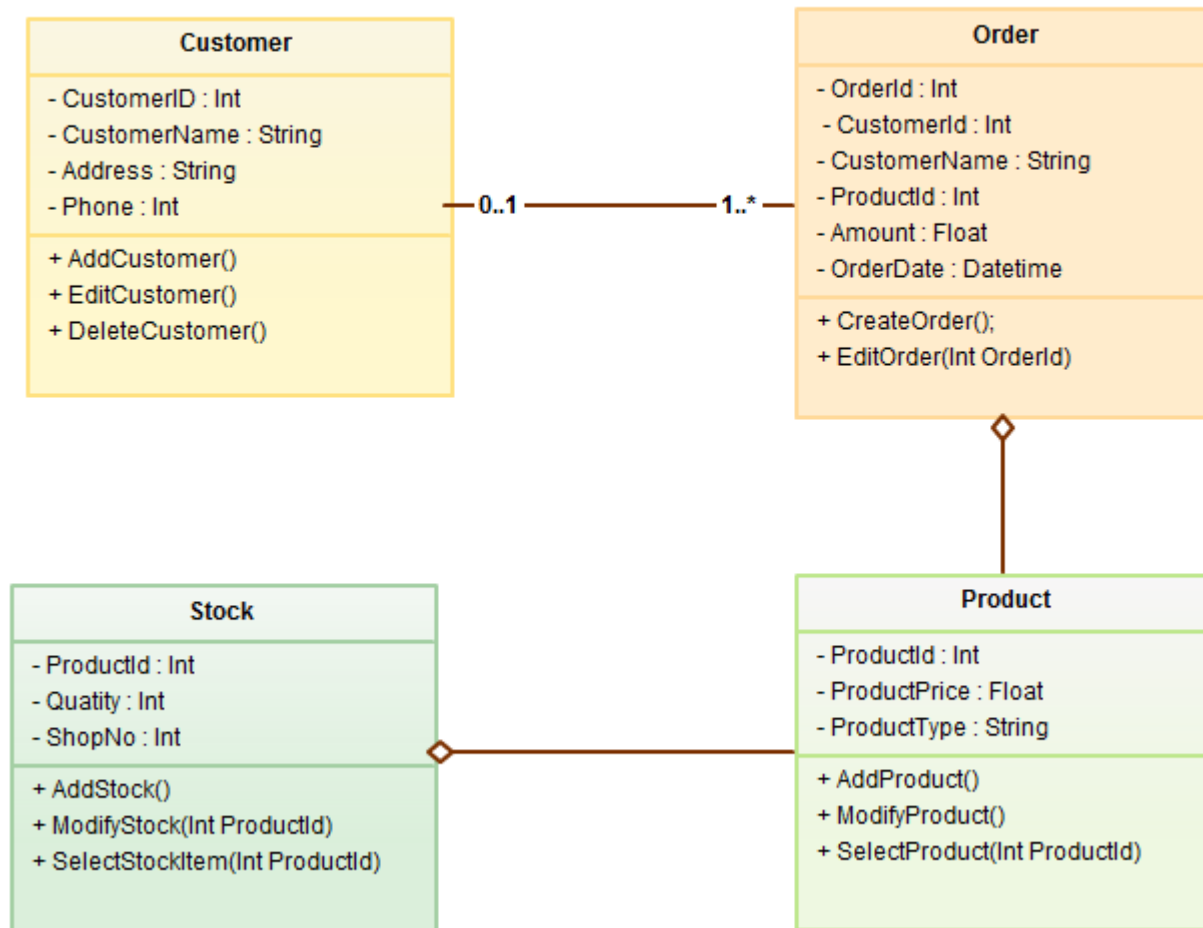


Primer – UML Use-Case diagram

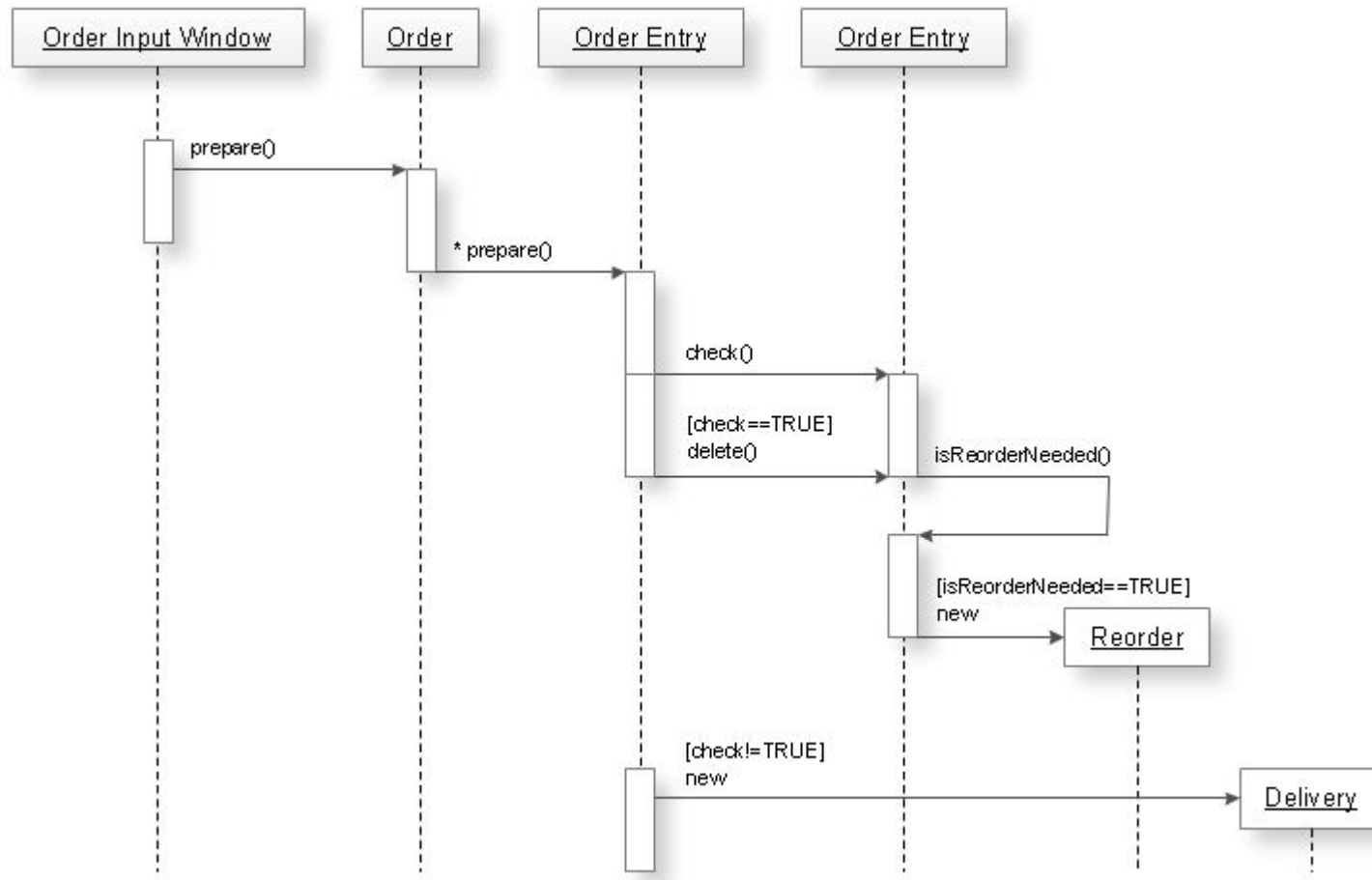


Primer – UML diagram kelas

Class Diagram for Order Processing System



Primer – UML diagram sekvenci



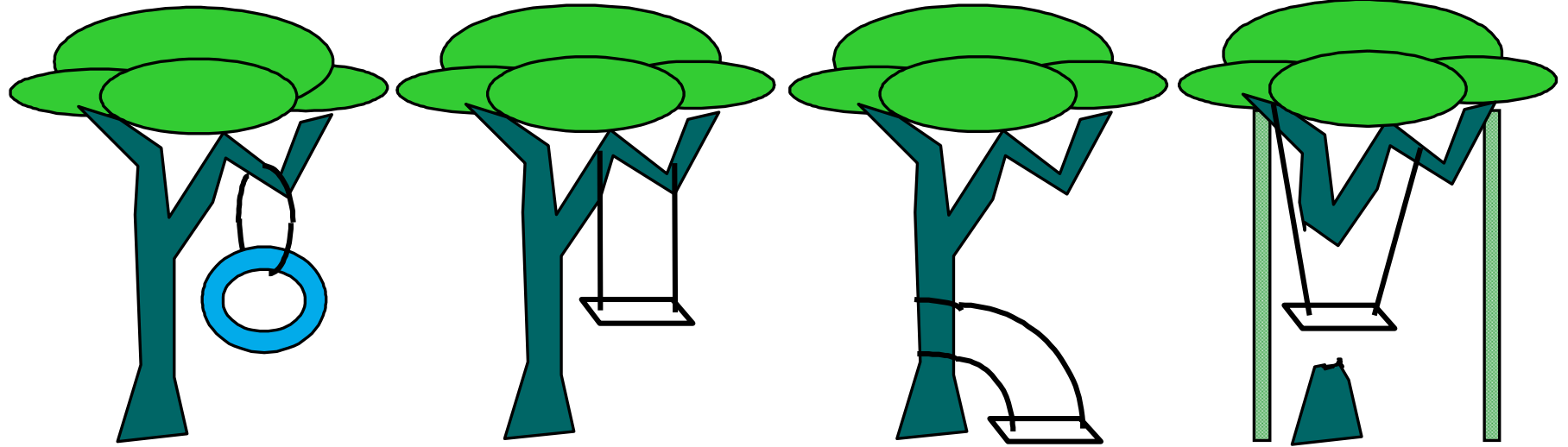
Problemi u razvoju softvera

CHAOS Summary 2009

A promotional banner for the CHAOS Summary 2009 report. It features a blue header with white text stating 'NEW! Standish Group report shows more projects failing and less successful projects.' Below this, a large yellow arrow points down to the text 'New CHAOS numbers show STARTLING RESULTS!' in red and yellow. Underneath, it says 'Boston, MA, - April 23 2009'. On the right side, there is a small image of the report cover, which includes a bar chart and the title 'CHAOS Summary 2009'.

- 32% projekata je realizovano na vreme, u okviru predviđenog budžeta i sa svom potrebnom funkcionalnošću
- 44% projekata nije realizovano na vreme i u okviru predviđenog budžeta i/ili nije imalo zahtevanu funkcionalnost
- 24% projekata je prekinuto

Tipičan razvoj softvera...



Očekivanja korisnika

Kako to korisnik opisuje

Kako to analitičar specificira

Šta programer implementira

Problemi

- složenost realnih sistema za koje se razvija softver
- socio-psihološki
- u komunikaciji
- tehnički
- spoljni uticaj
- ...

Prvi korak u razvoju softvera

- Prikupljanje zahteva
- Osnovni cilj je otkriti koje su potrebe svih zainteresovanih strana (*stakeholder*) vezano za softver

Zainteresovane strane

- Pojedinci ili organizacije koji su aktivno uključeni u projekat ili su zainteresovani za njegove rezultate
 - Naručioci
 - Investitori
 - Korisnici
 - Projektni tim
 - Različite organizacije uključene u projekat
 -

Efektivno prikupljanje zahteva

- Koristiti sve raspoložive izvore informacija!
 - intervju sa korisnicima različitih nivoa
 - brainstorming sesije
 - posmatranje aktivnosti korisnika (etnografski pristup)
 - analiza realnih dokumenata koje koriste
 - analiza funkcija postojećeg softvera, ako postoji
 - šeme, ekrani, izveštaji
 - analiza podataka postojećeg softvera, ako postoji
 - prezentacija prototipova
 -
- Ne potcenjivati problem!

Zahtevi

- Funkcionalni
 - Koje funkcije softver treba da obezbedi
- Nefunkcionalni (tehnički)
 - Performansa
 - Pouzdanost
 - Bezbednost
 - Dostupnost
 -
- Poslovna pravila
 - Uslovi koji moraju biti zadovoljeni tokom izvršavanja funkcija softvera

Kada se zahtevi prikupljaju

- Ne samo na početku
 - Obično tokom celokupnog životnog ciklusa projekta
- Na početku je potrebno izolovati zahteve „u grubim crtama“ radi sticanja kompletne slike i mogućnosti dekompozicije
- Kasnije se za svaku dekomponovanu celinu radi detaljno prikupljanje zahteva

Kako se prikupljaju zahtevi

- Zavisno od metodologije
 - Korisničke priče (user-stories)
 - Lista zahteva (*Product Backlog List*)
 - Slučajevi korišćenja (use cases)
 - Esencijalni slučajevi korišćenja
 - Sistemski slučajevi korišćenja
 - UML dijagrami slučajeva korišćenja
 - Prototipovi
 -

Korisničke priče (user-stories)

- tehnika ekstremnog programiranja (XP)
- Korisnici:
 - pišu “korisničke priče” (*user stories*) na kartonskim karticama
 - svaka priča predstavlja određenu elementarnu funkciju
 - kartice poseduju takve dimenzije da spreče pisanje predugačkih priča (obično 3X5 inča)

Korisničke priče

- tehnika ekstremnog programiranja
- Šablon:
- “Kao <uloga> želim da <cilj> [tako da <ostvarena korist>]“
- Primer:
 - Kao magacioner želim da vidim magacinsku karticu sa detaljima prometa
 - Kao referent kadrovske službe, želim da mi se obezbedi kreiranje podloge za godišnje odmore.

Lista zahteva (*Product Backlog List*)

- Tehnika SCRUM metodologije
- formira se prikupljanjem zahteva raspoloživih u datom trenutku od strane svih zainteresovanih:
 - direktnih korisnika
 - uprave
 - odeljenja prodaje i marketinga
 - odeljenja za pružanje podrške korisnicima
 - članova razvojnog tima
- Lista se ažurira isključivo od strane *vlasnika liste* (najčešće iz redova korisnika):
 - definiše prioritete
 - kontinuirano dodaje nove zahteve u saradnji sa zainteresovanim stranama.
- Projektni tim vlasniku liste obezbeđuje procene vremena i energije potrebne za realizaciju pojedinačnih zahteva

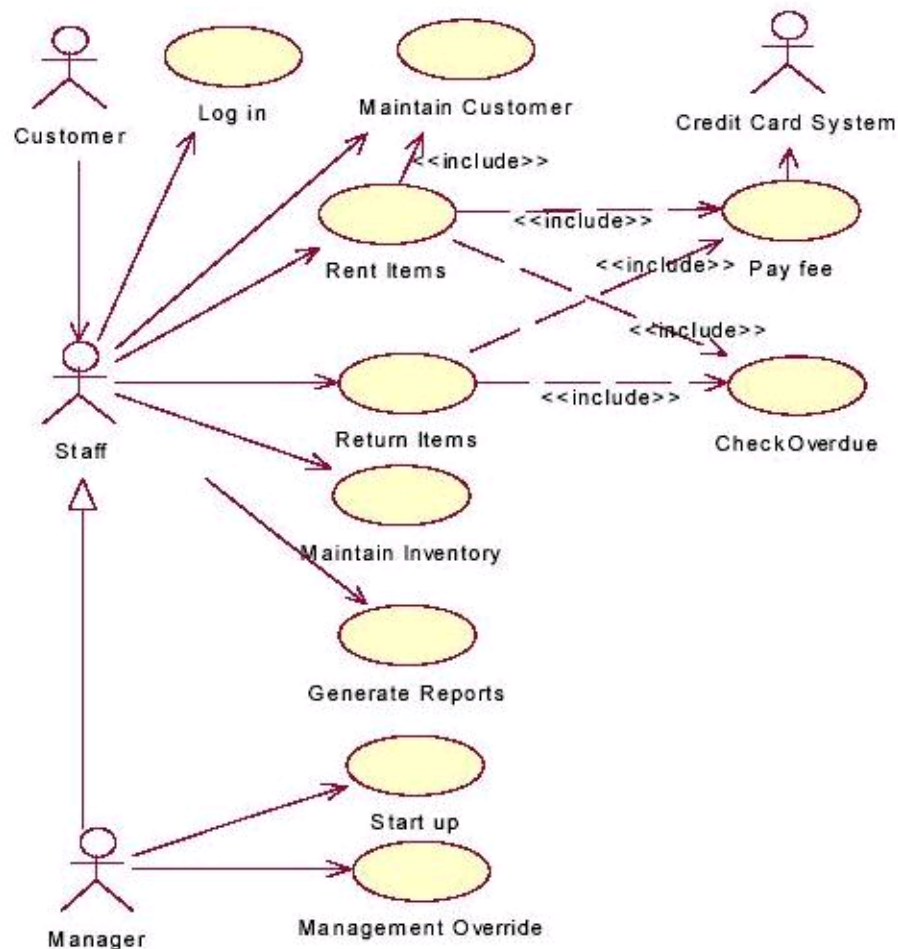
Primer liste

Hobby Site Product Backlog Items chosen for this Sprint

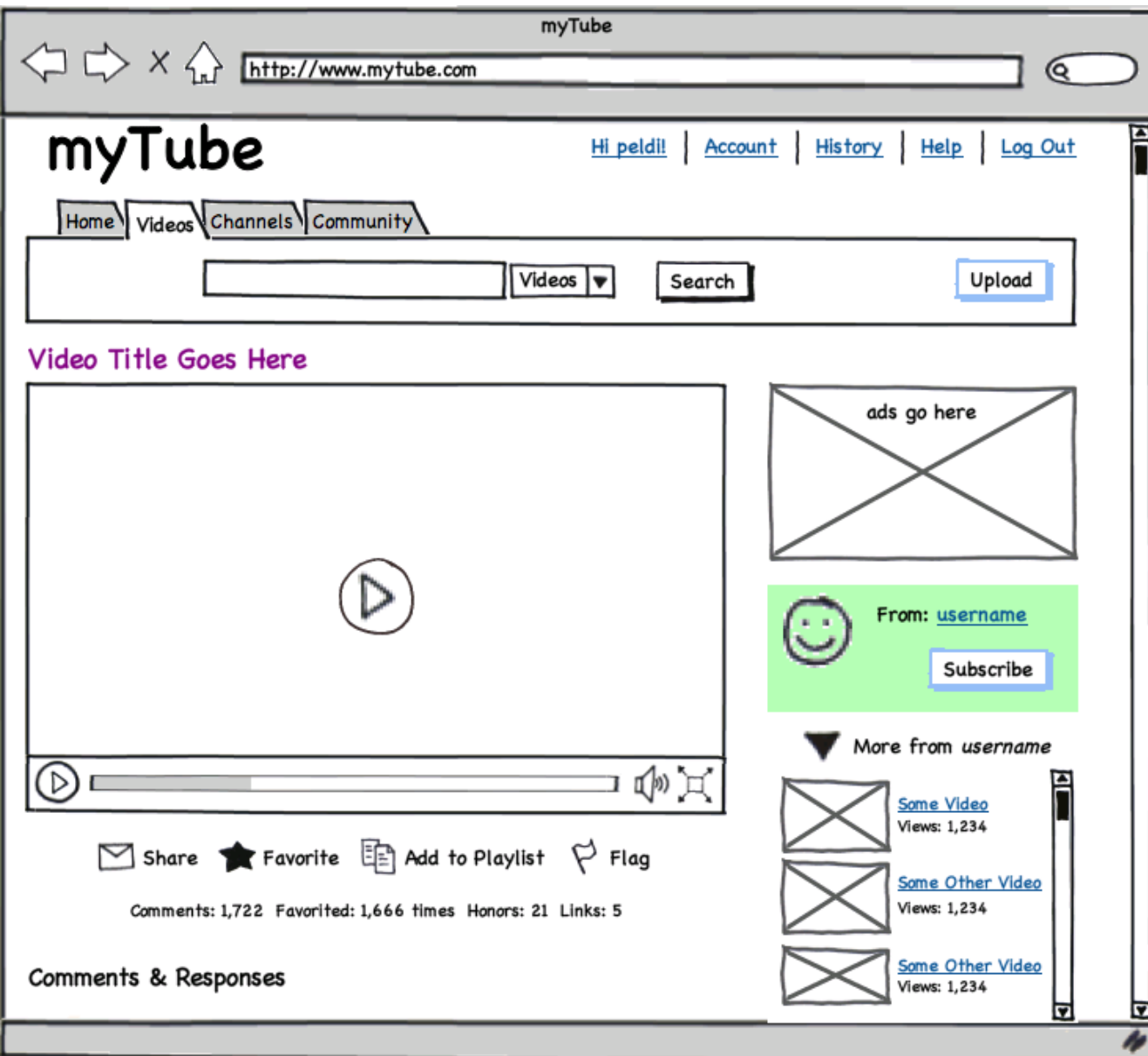
Item ID	Bug ID	Summary	Rank	Category	Accomplished?
1		Search other hobby sites: bring back results in a list	1	Search	X
2		Search hobby content & bring back results in a list	3	Search	X
3		Show direct display of hobby content in results	14	Search	
4		Use variety of methods for relevancy	4	Search	X
5		Keep track of queries for buckets	37	Search	X
6		Include user-generated content (message boards) in index	19	Search	X
7		Include Ads on search	16	Search	X
8		Include sites to index	2	Search	X
9		Search infrastructure/ops/machines		Search	X
10		"Did you mean?" support for misspellings, etc.	18	Search	X
11		Media for Message boards	23	Boards	
12		View/post to boards	6	Boards	X
13		View list of forums		Boards	X
14		See a list of threads on this board	7	Boards	X
15		Show different views of threads	40	Boards	X
16		Show messages/all messages in thread		Boards	X
17		Sort list of threads	43	Boards	X
18		Sort postings by rating	42	Boards	X
19		Signed in users can rate another user's posted message	41	Boards	X
20		Signed in users can start a thread/message	8	Boards	X
21		Signed in users can preview messages before submitting	22	Boards	X
22		Find all posts by another user	45	Boards	
23		Show user info for each message	24	Boards	

UML slučajevi korišćenja - primer

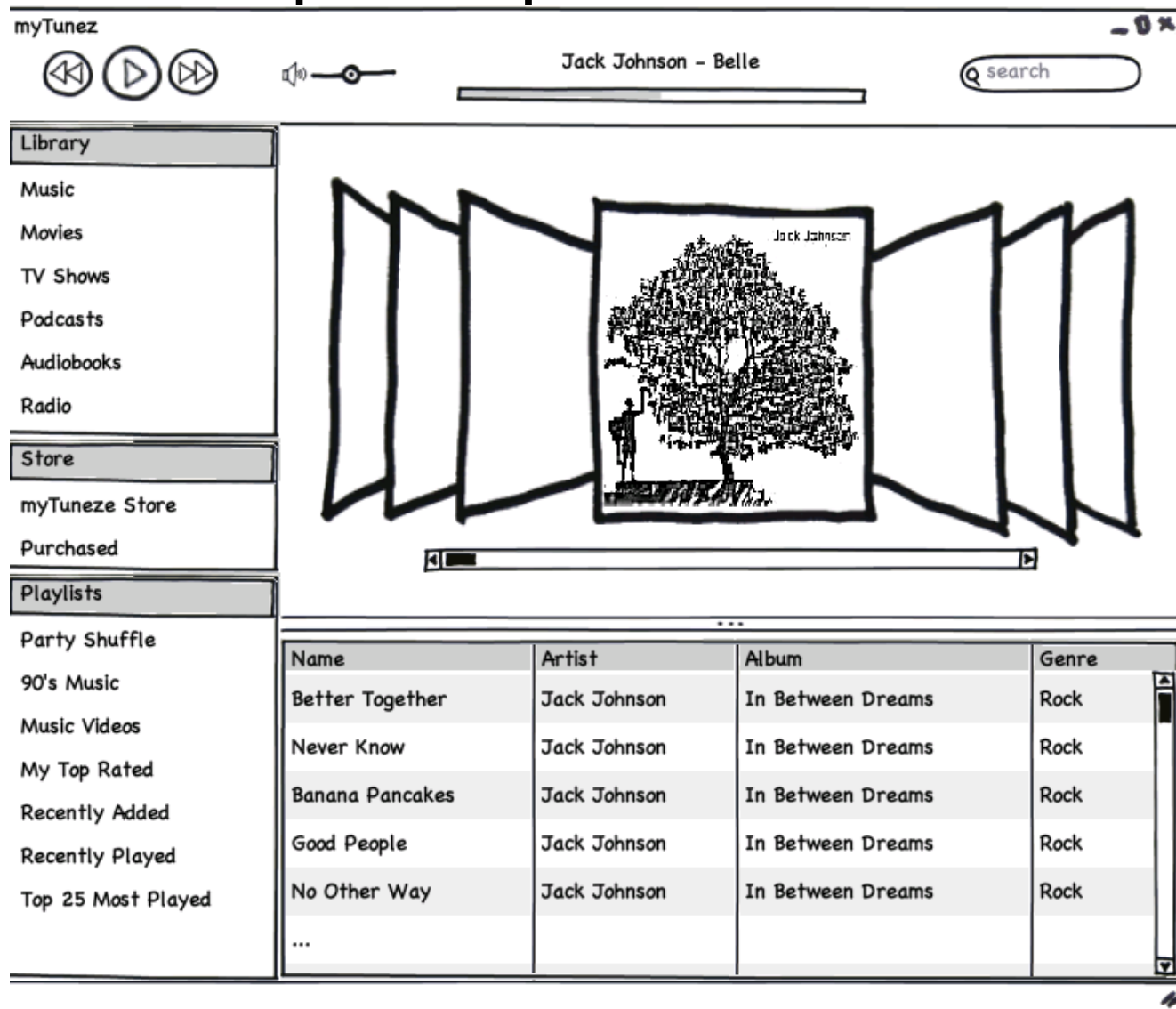
Video Rental Store Use Case Diagram



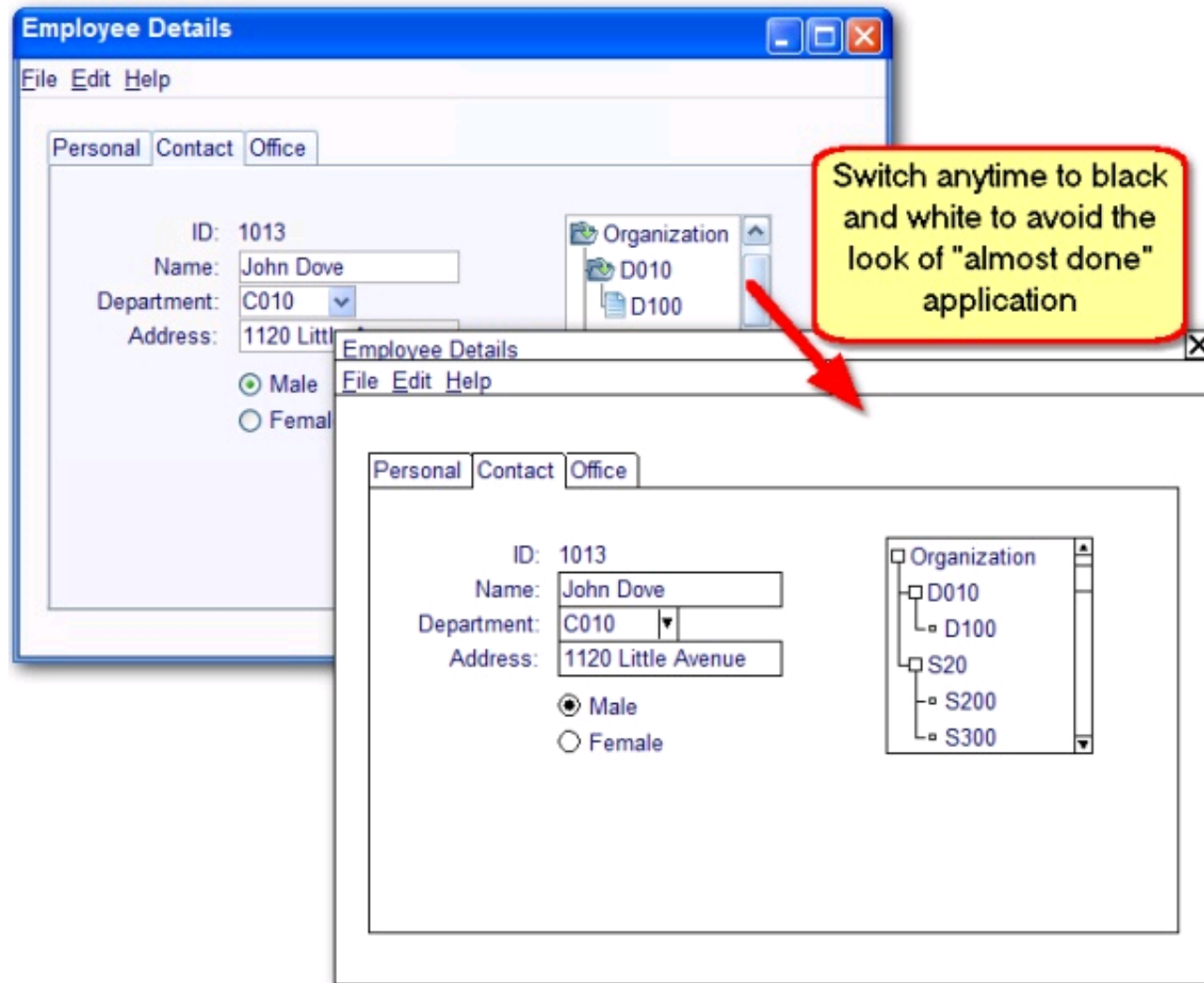
Prototipovi – primer 1



Prototipovi – primer 2



Prototipovi – primer 3

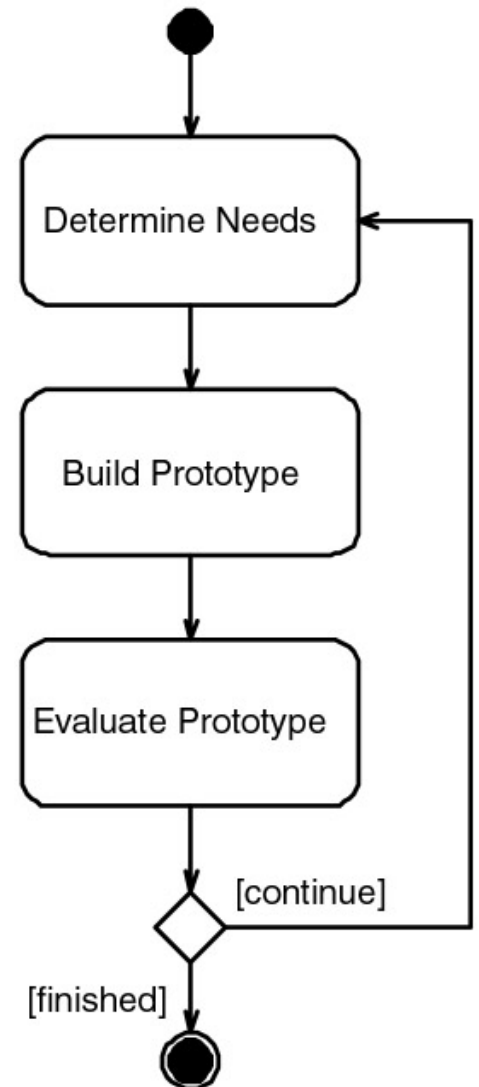


Prototipovi

- Osnovna namena prototipova je da pomognu korisnicima i razvojnom timu da bolje razumeju jedni druge
- Prototipovi korisničkog interfejsa pomažu i da se u ranim fazama oceni lakoća korišćenja softvera koji se razvija

Pravljenje prototipova

- U interakciji sa korisnicima
- U toku ili posle prikupljanja zahteva



Vrste prototipova

- Evolutivni prototipovi
 - Prototip se konstantno unapređuje, u saradnji sa korisnicima, dok ne preraste u konačno rešenje
- Prototipovi koji se odbacuju (throw-away)
 - Prototip koji se koristi samo za potrebe analize zahteva ili proveru nekih projektantskih odluka prilikom dizajna i zatim odbacuje

Cilj razvoja prototipa

- Evolutivnog prototipa
 - Da obezbedi komunikaciju uz oslonac na „nešto što radi“
 - Da dovede do finalnog rešenja uz minimalan utrošak vremena i energije
- Prototipa koji se odbacuje
 - Da obavi validaciju korisničkih zahteva ili projektantskih pretpostavki
 - Prototipovi koji se odbacuju ne smeju biti „skupi“ sa stanovišta vremena i energije

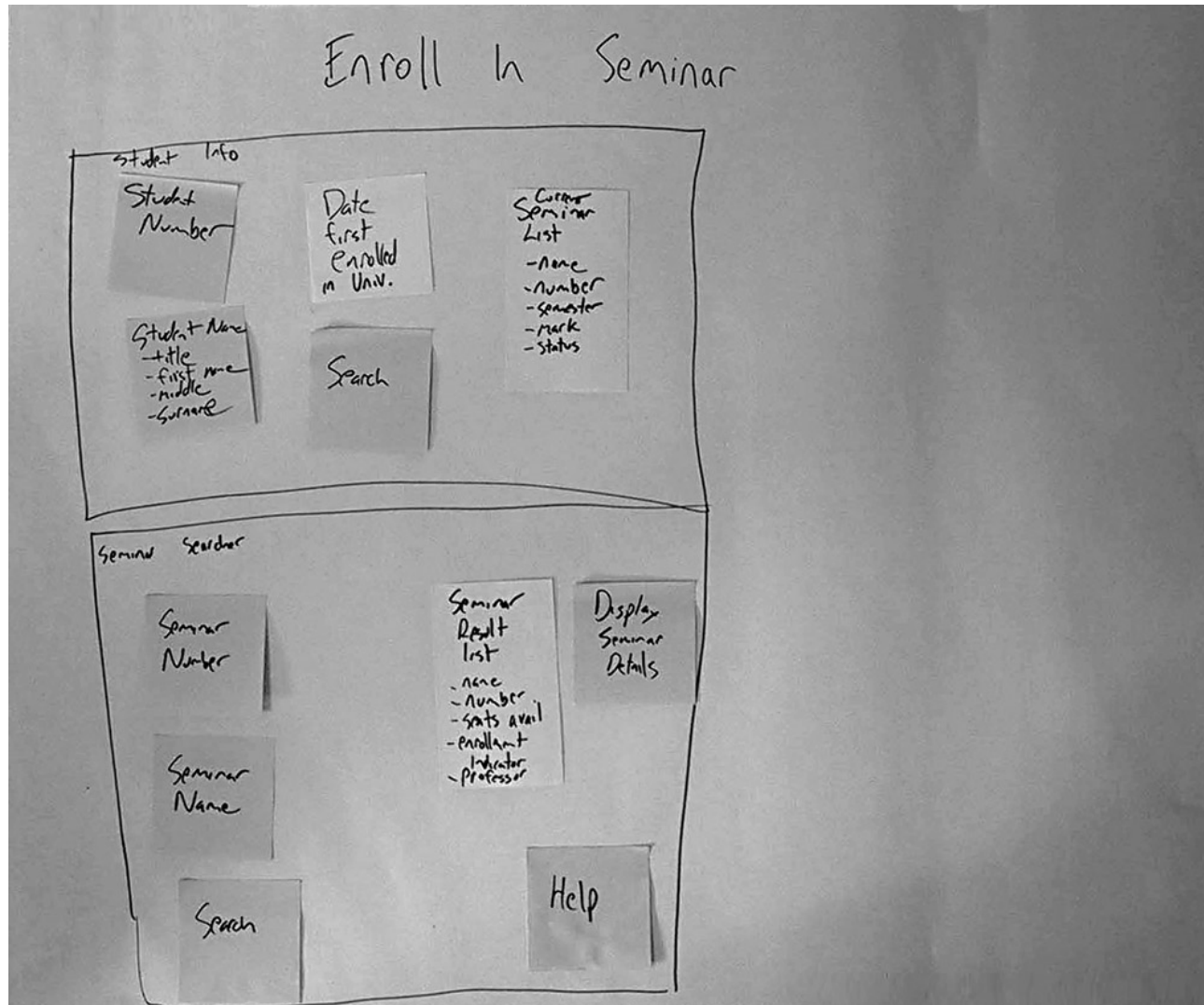
Prototipovi koji se odbacuju

- Za smanjivanje rizika usled nerazumevanja prilikom analize zahteva
- Za proveru različitih pretpostavki

Uobičajeni pristup

- U ranim fazama se koriste papirni prototipovi i/ili skice na papiru
- U kasnijim fazama se koriste izvršivi prototipovi

Primer 1 – papirni prototip (esencijalni prototip)



Primer 2 – skica na papiru ili tabli

Student Information

Help

Student Number: 789-567-234

First Name: Scott

Middle: William

Surname: Ambler

Salutation: Mr. ☐

Date first Enrolled: June 14 2003

Seminars:

Seminar	Term	Mark	Status
CSC 100 Intro to CS	Fall 2003	A+	Passed
CSC 200 Intro to AM	Fall 2003	A	Passed
CSC 203 Advanced AM	Spring 2004	—	Enrolled

Add...

Cop...

Transcript

Close

Add a Seminar

Help

Seminar Number: CSC*

Name: :*Agile*

Search

Results

Seminar	Term	Seats Avail	Professor
CSC 250 Agile Techniques	Fall 2004	4	Smithy, J.
CSC 300 Agile EUP	Spring 2003	17	Jones, S.
CSC 310 Agile Database Techniques	Spring 2004	0	Johnson, H.

Course description:

CSC 310 Agile Database Techniques

This Course describes evolutionary development strategies for data oriented development. See www.agiledata.org for details.

This Course currently has 39 people waitlisted for it.

Close

Primer 3 – izvršivi prototip

Edit Student Information - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Reload Home Search Favorites Media Links

Student number: 789-456-123

First name:

Middle name:

Surname:

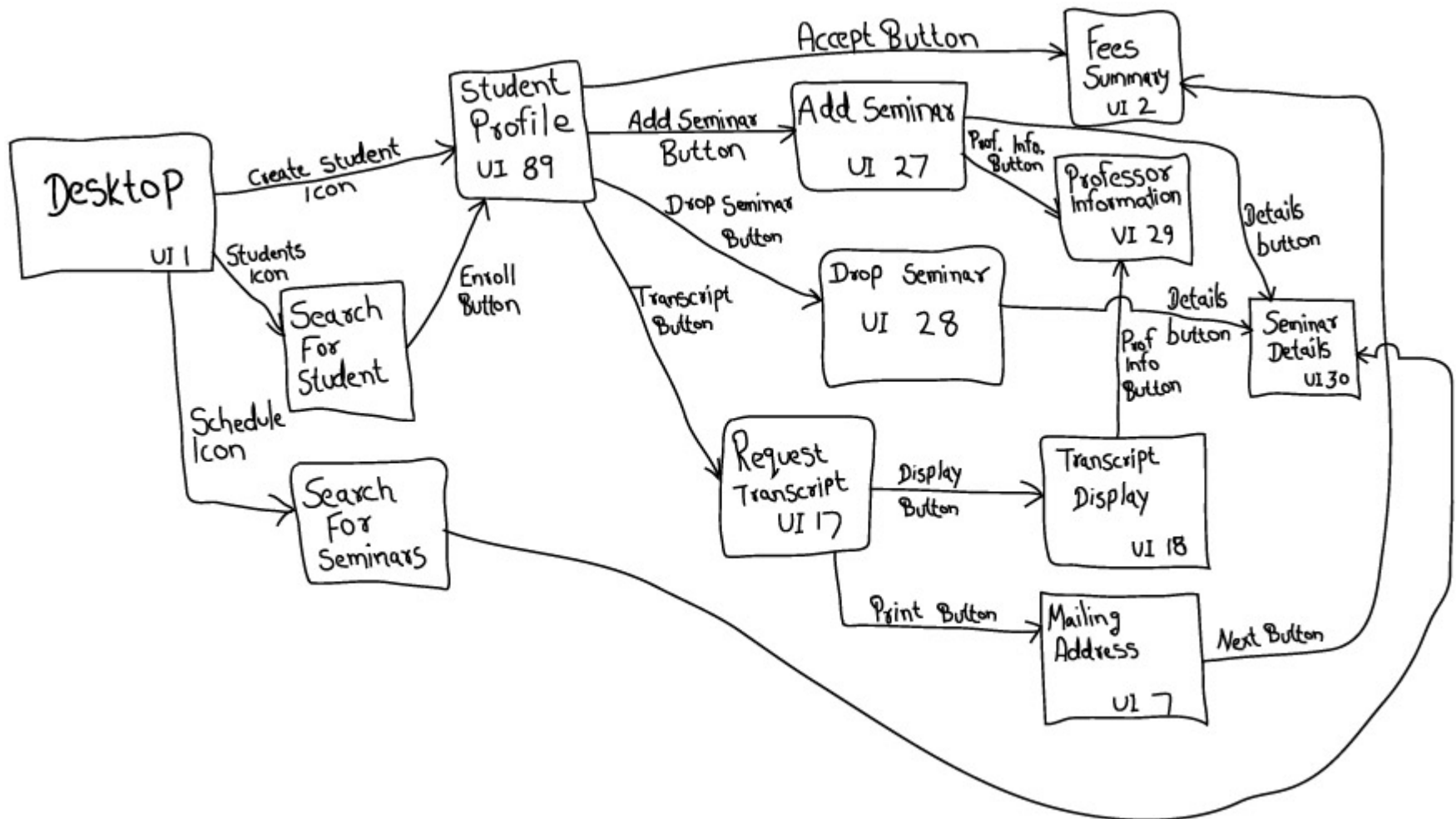
Salutation: ▼

First enrolled: June 14 2003

Schedule:

Seminar	Term	Mark	Status
CSC 100 Intro to C#	Fall 2003	A+	Passed
CSC 200 Intro to Agile Modeling	Fall 2003	B-	Passed
CSC 203 Advanced Agile Modeling	Spring 2004	-	Enrolled
CSC 220 Intro to Agile Databases	Spring 2004	-	Enrolled

Primer 4 - skica navigacije u okviru aplikacije



Zašto se koriste papirni prototipovi i/ili skice?

- Odnosno, zašto se odmah ne krene na kreiranje izvršivog prototipa?
 - Uglavnom je brže crtati skicu nego pisati kod
 - Može se brže menjati nego programski kod
 - Može se praviti više slika odjednom, sa više ili manje detalja (po želji)
 - Pogodnije za timski rad i brainstorming
 - Ovakvu vrstu prototipa ne moraju praviti programeri (može i korisnik)

Opasnosti razvoja prototipova

- Ukoliko su izvršivi, postoji opasnost da ih korisnik zameni sa finalnim rešenjem
- Ako nisu dovoljno detaljni, možda neće izazvati željenu reakciju korisnika
- Ako se previše vremena i energije potroši na prototipove koji se odbacuju, usporiće rad na projektu

Alati za razvoj prototipova

- Papir i olovka, nalepnice...
- Tabla i flomaster
- Alati za crtanje skica
 - Framebox, <http://framebox.org/>
 - Mockingbird, <https://gomockingbird.com/>
 - Balsamiq, <http://balsamiq.com/products/mockups/>
- Uobičajena razvojna okruženja i programski jezici
 - Java, Python, C#...
- CASE alati
- Jezici specifični za domen (Domain Specific Languages – DSL) i generatori koda
- ...

Prototipovi korisničkog interfejsa u cilju postizanja lakoće korišćenja

- Korisnički interfejs (User Interface – UI) je izuzetno važan
 - Korisnici sude o aplikaciji po korisničkom interfejsu
- Loše dizajniran UI može jako otežati rad i dovesti do katastrofalnih grešaka prilikom korišćenja
- Ako UI nije dobar, aplikacija se neće koristiti
 - Nezavisno od toga koliko korisnih funkcija nudi

Literatura

1. L. L. Constantine, Simplifying User Interfaces by Simplifying Use Cases, www.foruse.com/articles/simplifying.pdf
2. D. Avison, G. Fitzgerald, *Information Systems Development: Methodologies, Techniques and Tools*, Third Edition, McGraw-Hill Publishing, 2003.
3. M. L. Markus, "Power, Politics, and MIS Implementation", *Communications of the ACM*, June 1983, (26:6), pp. 430-444
4. P. M. Asaro, "Transforming Society by Transforming Technology: The Science and Politics of Participatory Design", University of Illinois, 1999.