

# SOFTWARE ENGINEERING

Week 5

## System Modeling and Requirements Engineering

Yard. Doç. Dr. A. Cüneyd TANTUĞ      Yard. Doç. Dr. Tolga OVATMAN  
Istanbul Technical University  
Computer Engineering Department

## Agenda

1. System Engineering
2. Requirements Engineering
3. Requirement Engineering Processes

System Modeling and Requirements Engineering 1.2

1. System Engineering ←
2. Requirements Engineering
3. Requirement Engineering Processes
4. The Case Study

# System Engineering

5.1

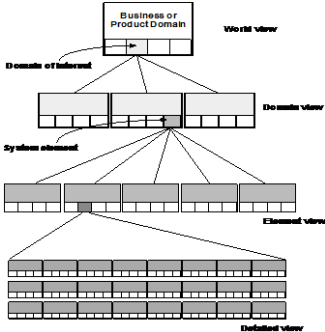
System Modeling and Requirements Engineering

## System Engineering

- Elements of a computer-based system
  - Software
  - Hardware
  - People
  - Database
  - Documentation
  - Procedures
- Systems
  - A hierarchy of macro-elements

System Modeling and Requirements Engineering 4

## The Hierarchy



System Modeling and Requirements Engineering 1.5

1. System Engineering
2. Requirements Engineering ←
3. Requirement Engineering Processes
4. The Case Study

# Requirements Engineering

5.2

System Modeling and Requirements Engineering

## Requirement Engineering

- The process of establishing the services that the customer requires from a system and the constraints under which it operates and is developed.
- The requirements themselves are the descriptions of the system services and constraints that are generated during the requirements engineering process.

System Modeling and Requirements Engineering

1

## Types of Requirement

- User requirements
  - Statements in natural language plus diagrams of the services the system provides and its operational constraints. Written for customers.
- System requirements
  - A structured document setting out detailed descriptions of the system's functions, services and operational constraints. Defines what should be implemented so may be part of a contract between client and contractor.

System Modeling and Requirements Engineering

1.8

## User and System Requirements

### User requirement definition

1. The MHC-PMS shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month.

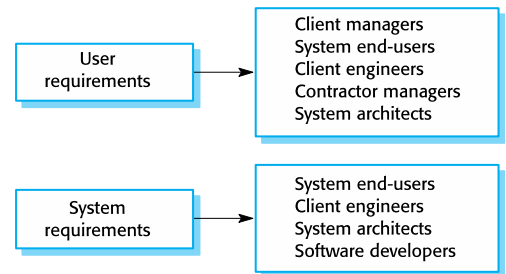
### System requirements specification

- 1.1 On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.
- 1.2 The system shall automatically generate the report for printing after 17.30 on the last working day of the month.
- 1.3 A report shall be created for each clinic and shall list the individual drug names, the total number of prescriptions, the number of doses prescribed and the total cost of the prescribed drugs.
- 1.4 If drugs are available in different dose units (e.g. 10mg, 20 mg, etc.) separate reports shall be created for each dose unit.
- 1.5 Access to all cost reports shall be restricted to authorized users listed on a management access control list.

System Modeling and Requirements Engineering

1.9

## Users of Requirements



System Modeling and Requirements Engineering

1.10

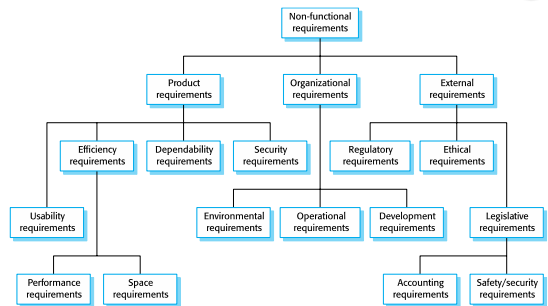
## Functional and Non-functional Requirements

- Functional requirements
  - Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.
  - May state what the system should not do.
- Non-functional requirements
  - Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.
  - Often apply to the system as a whole rather than individual features or services.
- Domain requirements
  - Constraints on the system from the domain of operation

System Modeling and Requirements Engineering

1.11

## Types of Non-functional Requirements



System Modeling and Requirements Engineering

1.12

## Requirements Specification

- ⇒ The process of writing down the user and system requirements in a requirements document.
- ⇒ User requirements have to be understandable by end-users and customers who do not have a technical background.
- ⇒ System requirements are more detailed requirements and may include more technical information.
- ⇒ The requirements may be part of a contract for the system development
  - It is therefore important that these are as complete as possible.

System Modeling and Requirements Engineering

1.13

## Guidelines For Writing Requirements

- ⇒ Invent a standard format and use it for all requirements.
- ⇒ Assign a unique number and source (mostly people) for each requirement.
- ⇒ Use language in a consistent way. Use shall for mandatory requirements, should for desirable requirements.
- ⇒ Use text highlighting to identify key parts of the requirement.
- ⇒ Avoid the use of computer jargon.
- ⇒ Include an explanation (rationale) of why a requirement is necessary.

System Modeling and Requirements Engineering

1.14

## An Example Requirement Document

- ⇒ [Click here for an example requirements document.](#)

System Modeling and Requirements Engineering

1.15

1. System Engineering
2. Requirements Engineering
3. Requirements Engineering Processes
4. The Case Study

## Requirements Engineering Processes

⇒ 5.3 ⇐

System Modeling and Requirements Engineering

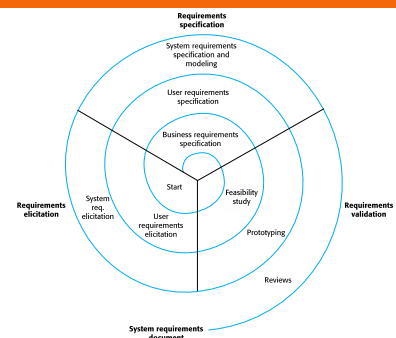
## Requirements Engineering Processes

- ⇒ **Inception**—ask a set of questions that establish ...
  - basic understanding of the problem
  - the people who want a solution
  - the nature of the solution that is desired, and
  - the effectiveness of preliminary communication and collaboration between the customer and the developer
- ⇒ **Elicitation**—elicit requirements from all stakeholders
- ⇒ **Elaboration**—create an analysis model that identifies data, function and behavioral requirements
- ⇒ **Negotiation**—agree on a deliverable system that is realistic for developers and customers
- ⇒ **Specification**—can be any one (or more) of the following:
  - A written document
  - A set of models
  - A formal mathematical
  - A collection of user scenarios (use-cases)
  - A prototype
- ⇒ **Validation**—a review mechanism that looks for
  - errors in content or interpretation
  - areas where clarification may be required
  - missing information
  - inconsistencies (a major problem when large products or systems are engineered)
  - conflicting or unrealistic (unachievable) requirements.
- ⇒ **Requirements management**

System Modeling and Requirements Engineering

1.17

## A Spiral View Of The Requirements Engineering Process



System Modeling and Requirements Engineering

1.18

## Problems Of Requirements Analysis

- ❏ Stakeholders don't know what they really want.
- ❏ Stakeholders express requirements in their own terms.
- ❏ Different stakeholders may have conflicting requirements.
- ❏ Organisational and political factors may influence the system requirements.
- ❏ The requirements change during the analysis process. New stakeholders may emerge and the business environment may change.

System Modeling and Requirements Engineering

1.19

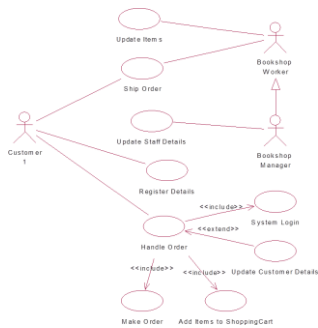
## Use Cases

- ❏ Use case diagrams are used to visualize, specify, construct, and document the (intended) behavior of the system, during requirements capture and analysis.
- ❏ Provide a way for developers, domain experts and end-users to Communicate.
- ❏ Serve as basis for testing.
- ❏ Main authors: *Booch, Rumbaugh, and Jacobson*
- ❏ The Object Management Group (OMG) is responsible for standardization. ([www.omg.org](http://www.omg.org))
- ❏ Current version is UML 2.0

Introduction &amp; UML

1.20

## Example : Use Case Diagram



Introduction &amp; UML

1.21

## Example : Use Case (Money Withdraw) - I

- ❏ **Use Case:** Withdraw Money
- ❏ **Author:** ZB
- ❏ **Date:** 1-OCT-2004
- ❏ **Purpose:** To withdraw some cash from user's bank account
- ❏ **Overview:** The use case starts when the customer inserts his credit card into the system. The system requests the user PIN. If the validation succeeded, the customer can choose the withdraw operation else alternative 1 – validation failure is executed. The customer enters the amount of cash to withdraw. The system checks the amount of cash in the user account, its credit limit. If the withdraw amount in the range between the current amount + credit limit the system dispense the cash and prints a withdraw receipt, else alternative 2 – amount exceeded is executed.
- ❏ **Cross References:** R1.1, R1.2, R7

Introduction &amp; UML

1.22

## Example : Use Case (Money Withdraw) - II

- ❏ **Actors:** Customer
- ❏ **Pre Condition:**
  - ❑ The ATM must be in a state ready to accept transactions
  - ❑ The ATM must have at least some cash on hand that it can dispense
  - ❑ The ATM must have enough paper to print a receipt for at least one transaction
- ❏ **Post Condition:**
  - ❑ The current amount of cash in the user account is the amount before the withdraw minus the withdraw amount
  - ❑ A receipt was printed on the withdraw amount
  - ❑ The withdraw transaction was audit in the System log file

Introduction &amp; UML

1.23

## Example : Use Case (Money Withdraw) - III

Typical Course of events:

Actor Actions	System Actions
1. Begins when a Customer arrives at ATM	
2. Customer inserts a Credit card into ATM	3. System verifies the customer ID and status
5. Customer chooses "Withdraw" operation	4. System asks for an operation type
7. Customer enters the cash amount	6. System asks for the withdraw amount
	8. System checks if withdraw amount is legal
	9. System dispenses the cash
	10. System deduces the withdraw amount from account
	11. System prints a receipt
13. Customer takes the cash and the receipt	12. System ejects the cash card

Introduction &amp; UML

1.24

## Example : Use Case (Money Withdraw) - IV

### Alternative flow of events:

- o Step 3: Customer authorization failed. Display an error message, cancel the transaction and eject the card.
- o Step 8: Customer has insufficient funds in its account. Display an error message, and go to step 6.
- o Step 8: Customer exceeds its legal amount. Display an error message, and go to step 6.

### Exceptional flow of events:

- o Power failure in the process of the transaction before step 9, cancel the transaction and eject the card

Introduction &amp; UML

1.25

## What is a User Story?

- o Simple, Clear, short description of customer valued functionality
- o User Stories are NOT part of the Scrum framework
- o User Stories are an eXtreme Programming technique
- o This may optionally be used to capture Product Backlog Items
- o The Product Backlog is the Scrum Artifact
- o User Stories capture Who, What and Why of any requirement
- o 3Cs – Card, Conversation, Confirmation
- o Conversation rather than documentation

## Leveraging User Roles and Personas

- o Write story from user's perspective
- o Understand the user's goal for the story
- o Understand the user's value from the story
- o Use human users
- o Avoid generic "as a user" or "as a customer"
- o If you have identified Personas, the story could be written from the point of view of this character/user

## User Story Template

Title: \_\_\_\_\_ Priority: \_\_\_\_\_

As a [type of user], I want [goal] so that [Value]

Notes:  
Assumptions:  
Constraints:

Estimate: \_\_\_\_\_

## User Story Example

Checkout Using Credit Card Priority: 25

As a book shopper, I can checkout using my credit card  
So that I can purchase a selected book.

Notes: Support mc, visa, amex

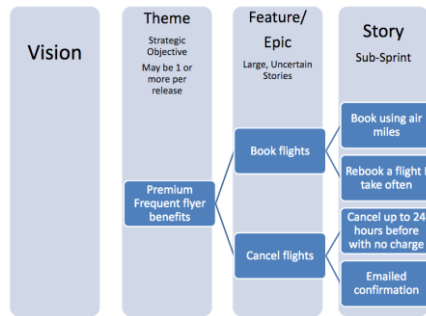
Constraints: Must use SBI payment gateway

Estimate: 13pts

## Acceptance Criteria

- o Given [context]
- o When [some event]
- o Then [outcome]

## Feature/Epic/User Story



System Modeling and Requirements Engineering

1.31

## User Story Examples

- » As a user I want emails with attachments to go faster so that I can work more efficiently
- » As a user I want to cancel a reservation so that I avoid being charged full rate

System Modeling and Requirements Engineering

1.32

## Tools For Requirements

- » OSRMT – Open Source Requirements Management Tool  
[www.sourceforge.net/projects/osrmt](http://www.sourceforge.net/projects/osrmt)
- » EasyRM – Cybernetic Intelligence  
[www.eary-rm.com](http://www.eary-rm.com)
- » Rational Requisite Pro  
[www.rational.com](http://www.rational.com)
- » OnYourMark - Omni-Vista  
[www.omni-vista.com](http://www.omni-vista.com)
- » RTM – Integratd Chipware  
[www.chipware.com](http://www.chipware.com)

System Modeling and Requirements Engineering

1.33

## Wrap-up

This week we present

- » System Engineering: How to model and understand the overall components of a software system
- » Requirements Engineering: How to manage and acquire the needs of customer
- » Requirement Process: What phases should be applied in effectively gathering requirements.

Introduction &amp; UML

1.34

## Next Week

- » We will introduce object oriented and classical approaches in Requirements Analysis!!!

Introduction &amp; UML

1.35