

Requirements Modeling in UML

Jesper Andersson

Linnæus University



Software development

Problem solving

- ✓ Understand the problem
- ✓ Derive a solution
- ✓ Execute the plan
- ✓ Evaluate

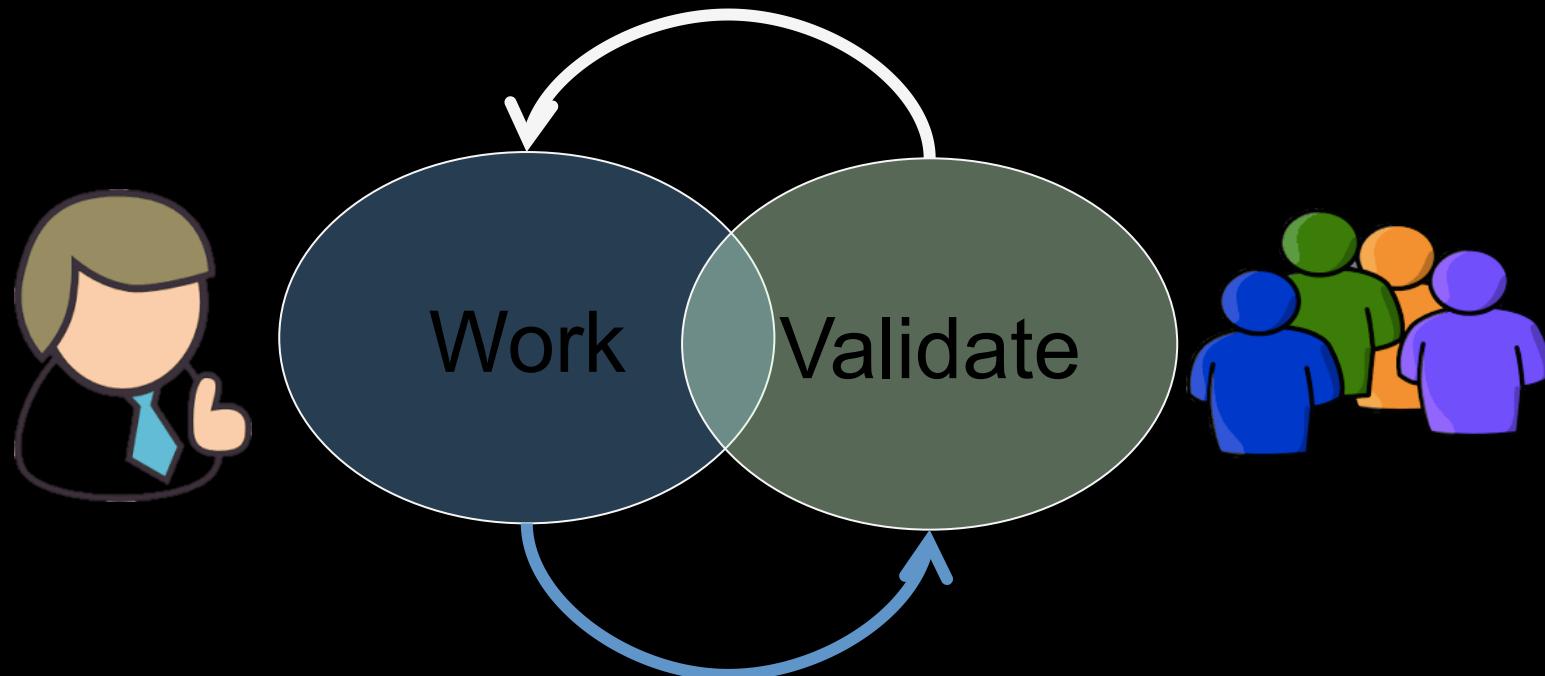
Software Development

- ✓ Requirements
- ✓ Design
- ✓ Implementation
- ✓ Testing



Linnæus University
Sweden

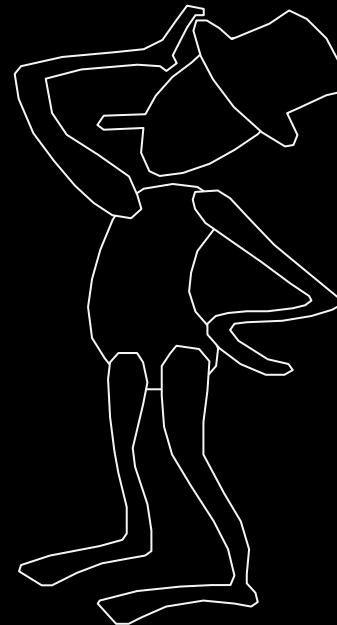
Understanding the Problem – Shared understanding



Requirements



Internal



External

Requirements engineering

1. Elicit information
 - a. Users' wish lists.
 - b. Additional documentation.
2. Analyze and identify requirements
 - a. Functionality
 - b. Quality
 - c. Process
3. Prioritize requirements
4. Specify the requirements for the product to be developed



Exercise

- ✓ Write down 3 most important requirements for a cell-phone?

- ✓ In a real project: How would you “elicit” similar requirements?

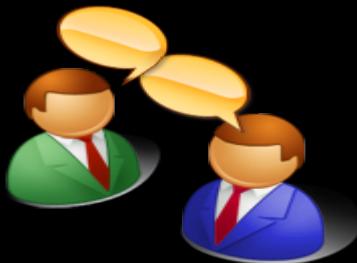
Remember!

- ✓ Requirements should be
 - Correct
 - Unambiguous
 - Complete
 - Verifiable
 - Consistent
 - Design independent
 - Traceable

Is this possible!?!

Requirements Engineering is Difficult

- ✓ People
- ✓ Communication
 - Different cultures
 - Different experiences
- ✓ Tacit knowledge
- ✓ Different agendas



Why do we Model System Goals?

- ✓ Capture and Communicate Goals among Stakeholders
- ✓ Understand the Problem and Each other!
- ✓ Bridge End-user perspective and Developer Perspective

Why we need Models



Guidelines – Writing Requirements

- ✓ Invent or find a standard format and use it for all requirements.
- ✓ Use language in a consistent way.
- ✓ Use “*shall*” or “*will*” for mandatory requirements, “*should*” for desirable requirements.
- ✓ *Do not use vague phrases* (e.g., “around a month,” “have basic knowledge of”)
- ✓ Every requirement must be *verifiable*.

- ✓ Group the requirements
- ✓ Prioritize
- ✓ Put everything else on the “future feature list”

Quality requirements

- ✓ "System shall be fast"
- ✓ "System shall be secure "
- ✓ What to do?
- ✓ A specification must be
 - Precise
 - Unambiguous
 - **Verifiable ... how can you test if something is fast and secure?**

These are also Requirements!

- ✓ Implementation (resource limitation, language and tools, hardware)
- ✓ Interface (constraints posed by interfacing with external systems)
- ✓ Operations (system management in its operational setting)
- ✓ Packaging (for example, a physical box)
- ✓ Legal (licensing)

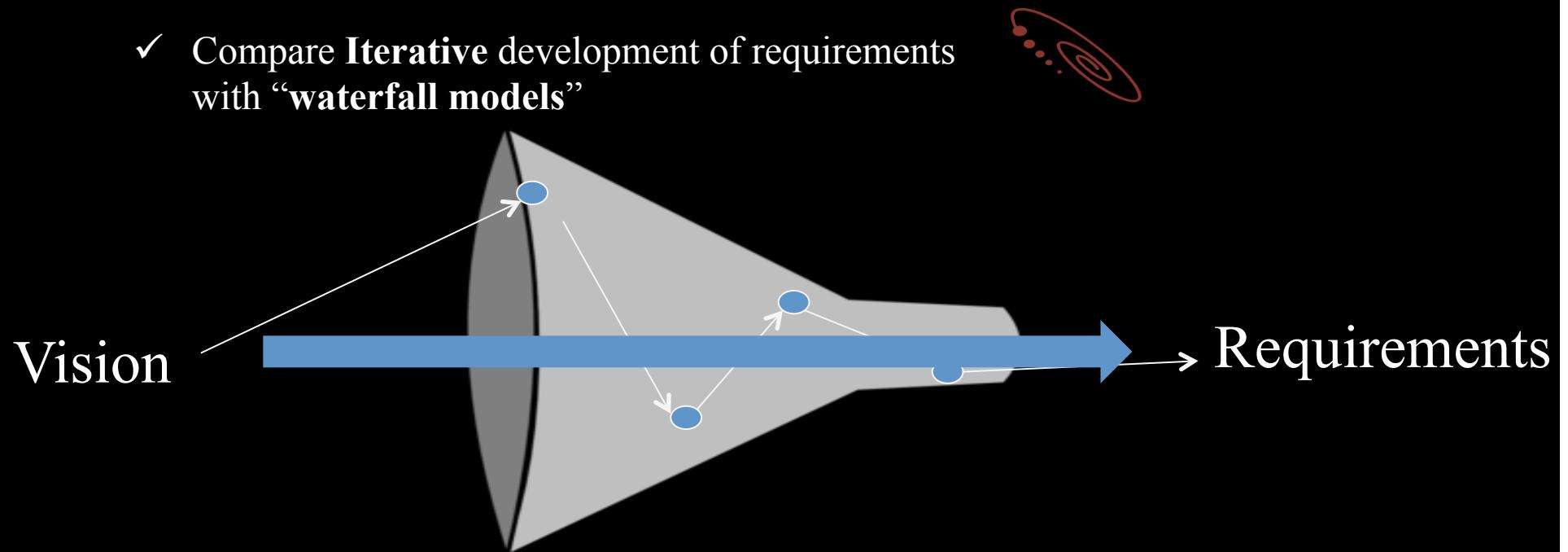
Requirements must Evolve

- ✓ Make use of iterations and increments!
- ✓ The difficulties can be found in
 - Communication!
 - Stakeholders do not understand each other.
- ✓ How can we make stakeholders agree and share the same vision?

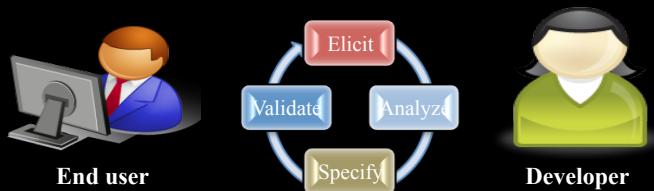


Requirements must Evolve

- ✓ Compare **Iterative** development of requirements with “**waterfall models**”

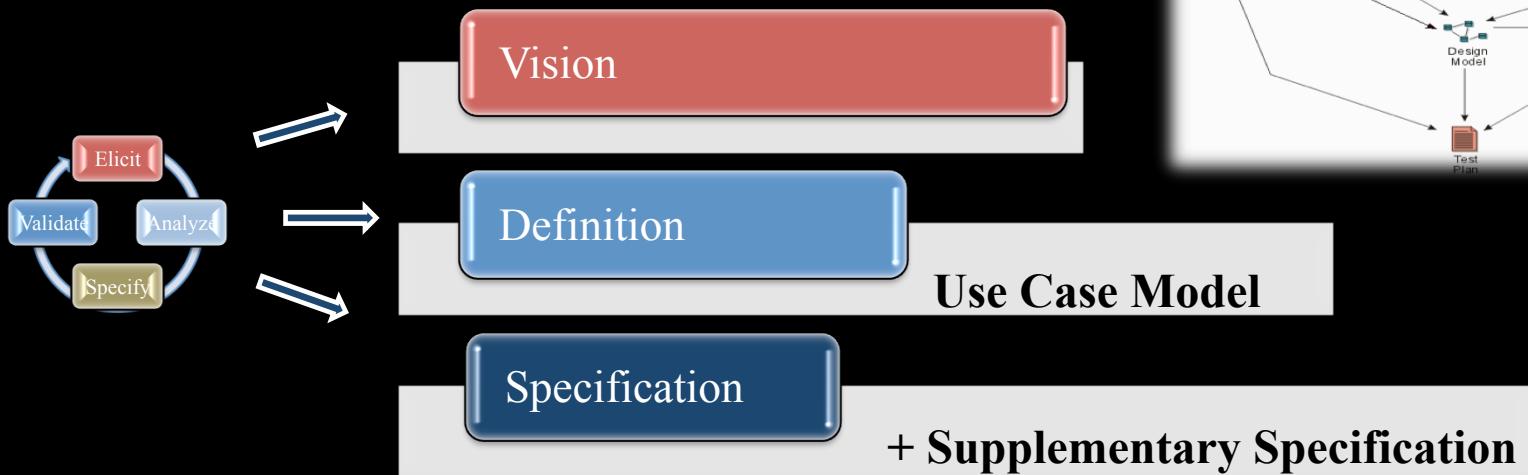


Iterative and Incremental



**Vision
Feasibility Study**
Use Case model
Supplementary specification

An Incremental approach



So we start with a Vision

- ✓ This vision defines the stakeholders' view on the technical solution to be developed.
- ✓ The system is defined in terms stakeholders' key needs and features
- ✓ The vision outlines the systems envisioned core requirements.

Checklist

- The problem behind the problem has been fully explored
- The problem statement is correctly formulated
- The list of stakeholders is complete and correct
- The features solve the identified problems
- The features are consistent with the identified constraints
- Everyone agrees on the definition of the system boundaries
- Constraints on the system have been sufficiently explored
- All key features of the system been identified and defined

A Vision Document Template

<Project Name> Vision <i>Usage note: There is procedural guidance within this template that appears in a style named InfoBlue. This style has a hidden font attribute allowing you to toggle whether it is visible or hidden in this template. Use the Word menu Tools->Options->View->Hidden Text checkbox to toggle this setting. A similar option exists for printing Tools->Options->Print.</i> 1. Introduction 2. Positioning 2.1 Problem Statement <i>(Provide a statement summarizing the problem being solved by this project. The following format may be used:)</i> <table border="1"><tr><td>The problem of</td><td>[describe the problem]</td></tr><tr><td>affects</td><td>[the stakeholders affected by the problem]</td></tr><tr><td>The impact of which is</td><td>[what is the impact of the problem?]</td></tr><tr><td>a successful solution would be</td><td>[list some key benefits of a successful solution]</td></tr></table> 2.2 Product Position Statement <i>(Provide an overall statement summarizing, at the highest level, the unique position the product intends to fill in the marketplace. The following format may be used:)</i> <table border="1"><tr><td>For</td><td>[target customer]</td></tr><tr><td>Who</td><td>[environment of the need or opportunity]</td></tr><tr><td>The (product name)</td><td>[is a [product category]]</td></tr><tr><td>That</td><td>[associates of key benefit; that is, the compelling reason to buy]</td></tr><tr><td>Unlike</td><td>[provides competitive alternative]</td></tr><tr><td>Our product</td><td>[associates of primary differentiation]</td></tr></table> <i>[A product position statement communicates the intent of the application and the importance of the project to all concerned personnel.]</i> 3. Stakeholder Descriptions 3.1 Stakeholder Summary <table border="1"><tr><td>Name</td><td>Description</td><td>Responsibilities</td></tr><tr><td>[Name the stakeholder type.]</td><td>[Briefly describe the stakeholder.]</td><td>[Summarize the stakeholder's key responsibilities with regard to the system being developed; that is, their interest as a stakeholder. For example, this stakeholder: [assumes that the system will be maintainable [assumes that there will be a market demand for the product's features]</td></tr></table>			The problem of	[describe the problem]	affects	[the stakeholders affected by the problem]	The impact of which is	[what is the impact of the problem?]	a successful solution would be	[list some key benefits of a successful solution]	For	[target customer]	Who	[environment of the need or opportunity]	The (product name)	[is a [product category]]	That	[associates of key benefit; that is, the compelling reason to buy]	Unlike	[provides competitive alternative]	Our product	[associates of primary differentiation]	Name	Description	Responsibilities	[Name the stakeholder type.]	[Briefly describe the stakeholder.]	[Summarize the stakeholder's key responsibilities with regard to the system being developed; that is, their interest as a stakeholder. For example, this stakeholder: [assumes that the system will be maintainable [assumes that there will be a market demand for the product's features]
The problem of	[describe the problem]																											
affects	[the stakeholders affected by the problem]																											
The impact of which is	[what is the impact of the problem?]																											
a successful solution would be	[list some key benefits of a successful solution]																											
For	[target customer]																											
Who	[environment of the need or opportunity]																											
The (product name)	[is a [product category]]																											
That	[associates of key benefit; that is, the compelling reason to buy]																											
Unlike	[provides competitive alternative]																											
Our product	[associates of primary differentiation]																											
Name	Description	Responsibilities																										
[Name the stakeholder type.]	[Briefly describe the stakeholder.]	[Summarize the stakeholder's key responsibilities with regard to the system being developed; that is, their interest as a stakeholder. For example, this stakeholder: [assumes that the system will be maintainable [assumes that there will be a market demand for the product's features]																										
3.2 User Environment <i>(Detail the working environment of the target user. Here are some suggestions: Number of people involved in completing the task? Is this changing? How long is a task cycle? Amount of time spent in each activity? Is this changing? Any unique environmental constraints: mobile, outdoors, in-flight, and so on? Which system platforms are in use today? Future platforms? What other applications are in use? Does your application need to integrate with them? This is where extracts from the Business Model could be included to outline the task and roles involved, and so on.)</i>	<table border="1"><tr><td>Name</td><td>Description</td><td>Responsibilities</td></tr><tr><td></td><td></td><td><i>monitors the project's progress approves funding and so forth</i></td></tr></table>	Name	Description	Responsibilities			<i>monitors the project's progress approves funding and so forth</i>																					
Name	Description	Responsibilities																										
		<i>monitors the project's progress approves funding and so forth</i>																										
4. Product Overview 4.1 Needs and Features <i>(Avoid design. Keep feature descriptions at a general level. Focus on capabilities needed and why (not how) they should be implemented. Capture the stakeholder priority and planned release for each feature.)</i>	<table border="1"><tr><td>Need</td><td>Priority</td><td>Features</td><td>Planned Release</td></tr><tr><td></td><td></td><td></td><td></td></tr></table>	Need	Priority	Features	Planned Release																							
Need	Priority	Features	Planned Release																									
5. Other Product Requirements <i>(At a high level, list applicable standards, hardware, or platform requirements; performance requirements; and environmental requirements. Define the quality ranges for performance, robustness, fault tolerance, usability, and similar characteristics that are not captured in the Feature Set. Note any design constraints, external constraints, assumptions or other dependencies that, if changed, will alter the Vision document. For example, an assumption may state that a specific operating system will be available for the hardware designated for the software product. If the operating system is not available, the Vision document will need to change. Define any specific documentation requirements, including user manuals, online help, installation, labeling, and packaging requirements. Define the priority of these other product requirements. Include, if useful, attributes such as stability, benefit, effort, and risk.)</i>	<table border="1"><tr><td>Requirement</td><td>Priority</td><td>Planned Release</td></tr><tr><td></td><td></td><td></td></tr></table>	Requirement	Priority	Planned Release																								
Requirement	Priority	Planned Release																										

So far...

- ✓ We talked about the importance of understanding the problem
 - For planning
 - For doing the right things
 - For developing the right product for the customer
- ✓ We talked about dealing with complexity using
 - Abstraction, and
 - Decomposition

So far

- ✓ We discussed requirements types such as,
 - Functional and
 - Quality requirements
- ✓ And how difficult it was to
 - Find them and
 - Specify them

Behavioral Modeling

- ✓ Avoid the imperative form!
 - Shall do this!
 - Shall do that!
- ✓ Expressing **system dynamics/semantics** instead
- ✓ Several levels
 - Use Cases
 - Business Processes
 - Use-case realizations
 - Object life-cycle
 - Object – to – Object Interaction

Use cases

- ✓ A use case **tells a story** of how to fulfill a goal, or a set of stories of both **reaching and failing**.
- ✓ A use case is performed by actors in a timely manner, i.e., there is a **beginning** and an **end** (multiple) that should be **reached** in a **reasonable time**.
- ✓ Examining the **Goals** for a system is a good starting point for identifying **requirements**



Interactions

- ✓ *Actors interact with the system in scenarios defined by use cases.*
- ✓ What is a *scenario*?

sce·nar·i·o

[si-nair-ee-oh, -nahr-] noun, plural -nar·i·os.

1. an outline of the plot of a dramatic work, giving particulars as to the scenes, characters, situations, etc.
2. the outline or the manuscript of a motion picture or television program, giving the action in the order in which it takes place, the description of scenes and characters, etc.
3. ***an imagined or projected sequence of events***, especially any of several detailed plans or possibilities: One scenario calls for doubling profits by increasing our advertising, the other by reducing costs.

Where we start? – Scenarios

- ✓ “An outline of events that elicits some system behavior“
- ✓ Attractive because you can involve en users directly.
- ✓ We have many similar behaviors in a system!
- ✓ We need abstractions that models “similar” behaviors!
 - Several Scenarios → ??
 - Text, lots of “ifs”, “thens” and “loops”
 - Activity diagrams



Example: ATM

- ✓ What are the goals for an ATM?
- ✓ Withdraw Money
- ✓ Account Statement
- ✓ "Scenarios" or stories that describe how these goals can be achieved. Based on your own experiences wish.
 - Descriptive (is)
 - Prescriptive (should)



Scenarios – some examples

#1 The card holder inserts the bank card into the ATM. The ATM asks for a PIN code and the card holder enters her code. The code is correct so the ATM asks for an amount to withdraw. The card holder selects one of the preset amounts (300 RNB) which are displayed on the screen. The ATM ejects the card, provides the money to the card holder together with a receipt.

Scenarios – some examples

#2 The card holder inserts the bank card into the ATM. The ATM asks for a PIN code and the card holder enters her code. The code is correct so the ATM asks for an amount to withdraw. The card holder enters an amount (400 RNB) using the numeric key pad. The ATM ejects the card, provides the money to the card holder together with a receipt.

Scenarios – some examples

- ✓ #3 *The card holder inserts the bank card into the ATM. The ATM asks for a PIN code and the card holder enters her code. The code is correct so the ATM asks for an amount to withdraw. The card holder enters an amount (450 RNB) using the numeric key pad. The ATM rejects the amount. The card holder decides to abort. The ATM ejects the card.*

Scenarios and Use cases

- ✓ Brainstorm and document scenarios, they
 - Collect details
 - Communicate with end-users
 - Perfect starting point for test cases
- ✓ How many stories do we have?



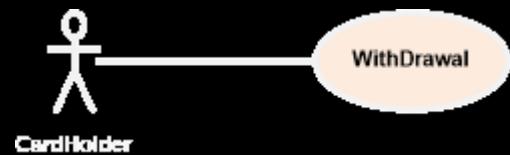
Scenarios and Use Cases

- ✓ So now we have many scenarios
 - Overlapping
 - Conflicting
- ✓ **We need some abstraction to reduce the number!**
- ✓ **The Use Case**
 - Scenarios are instances of a use case
 - *Scenario – Use case* similar to *Object – Class* relationship

Actors

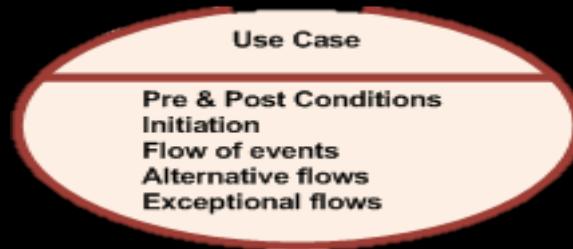
- ✓ Classifiers – represents a class, a type, an abstraction.
- ✓ Someone or some System that is involved in (interacts with) a use case.
- ✓ One actor may extend another (advanced)
- ✓ Actors in the ATM Case?
 - Customer
 - Bank
 - ATM Technician

Example: ATM Use case and Actor



Use Case Descriptions

- ✓ Details a use case with descriptions of
 - **Initiation**
 - **Pre and post conditions**
 - **Primary** flow of events
 - **Alternative and secondary** flows
 - Use case **relationships**
- ✓ Flows of events
 - Primary path to achieve a goal
 - Alternative paths to achieve a goal
 - Exceptional paths terminate without achieving a goal



Use Case Process:

- ✓ Common glossary
- ✓ Brainstorm **scenarios**
- ✓ **Identify** Actors and Use Cases
- ✓ **Describe** the Use Case model
 - Describe individual Use Cases
 - Structure the Use Case model
 - Prioritize the Use Case model
- ✓ **Review** the Use Case model



Write down scenarios for The Chat application

A chat system connects two or more users. A user uses the chat to interact with the chat system and the other users. Only registered users can use the system. New users must register with the server first. When a user wants to use the system, the user must first log on to the chat system. When a user successfully logs on to the chat server, the user is shown a list of “buddies” that are currently logged on to the chat server and a list of active chat rooms. Users may enter named chat rooms they have access to. In the chat rooms they write messages which are displayed to all users in that chat room. A user can “whisper” to another user in the same room, which means that the message is only forwarded to that particular user. Users may also create and enter private chat rooms for “face-to-face”, private conversations. A user may create a “buddy-list” which will notify whenever a user on that list logs on to the system. The buddy list can also be used to create private chat rooms.

Scenarios

- ✓ #1 Helen enters the registration page, fills the form with her **Helen Anderson, hean** as nickname, and **hean@neu.edu.cn** as her email address. There is also an optional field for a phone number that she decides to leave blank. When the form is filled out correctly (unique nickname, valid email address, etc) the system activates the submit button. The form data is submitted when Helen presses the submit. The system processes the form and sends out an email to Helen's email with logon information, including a one-time password and a URL to the first time logon page.

Describe Use Cases

- ✓ Iterate through the proposed Use Cases
- ✓ Use questions for validation
 - How does it **start** and **terminate**?
 - Pre and post conditions.
 - How does it interact with actors?
 - How does it exchange data with actors?
 - Are there any **optional** paths in the flow of events?
 - How can we describe the use case so a customer and user can understand it?

Example: ATM Use case Withdrawal

Initiation: Cardholder inserts card

1. Check if card is valid
2. Ask Cardholder for PIN
3. Check if PIN is correct
4. Ask for amount to withdraw
5. Ask if the Cardholder would like a receipt
6. Eject card
7. Present money and Receipt to the Cardholder



Please note that we do not consider the optional paths for this example!!

Pre and post conditions? Termination?

Use Case Specification

<project>	Use-case Specification: <use-case name>	Date: <dd/mm/yyyy>
-----------	-----------------------------------------	--------------------

<project>
Use-Case: <use-case name>

1 Brief Description
<brief description of use-case>

2 Actor Brief Descriptions

2.1 <Actor 1 Name>

3 Preconditions
<pre-condition 1>

4 Basic Flow of Events

1. The use case begins when <actor>, <does something>...
2. <basic flow step 1>
3. ...
4. <basic flow step n>
5. The use case ends.

5 Alternative Flows

5.1 <alternate flow 1>
If in step <x> of the basic flow the <actor or system does something>, then

1. <describe flow>
2. The use case resumes at step <y>

6 Subflows

6.1 <subflow 1>

1. <subflow 1, step 1>
2. ...
3. <subflow 1, step n>

7 Key Scenarios

7.1 <scenario 1>

1. <scenario 1, step 1>
2. ...
3. <scenario 1, step n>

8 Post-conditions

8.1 <post-condition 1>

9 Special Requirements
<special requirement 1>

Example: Chat Application



- ✓ From Scenarios, identify and describe the use cases
- ✓ Put the scenarios for registering together into a use case.
- ✓ The description should include.
 - Initiation
 - Pre-post conditions
 - Flows
 - Primary
 - Secondary
 - Exceptional
 - Termination

Use Case – RegisterUser

Initiation: The user enters the RegisterUser page

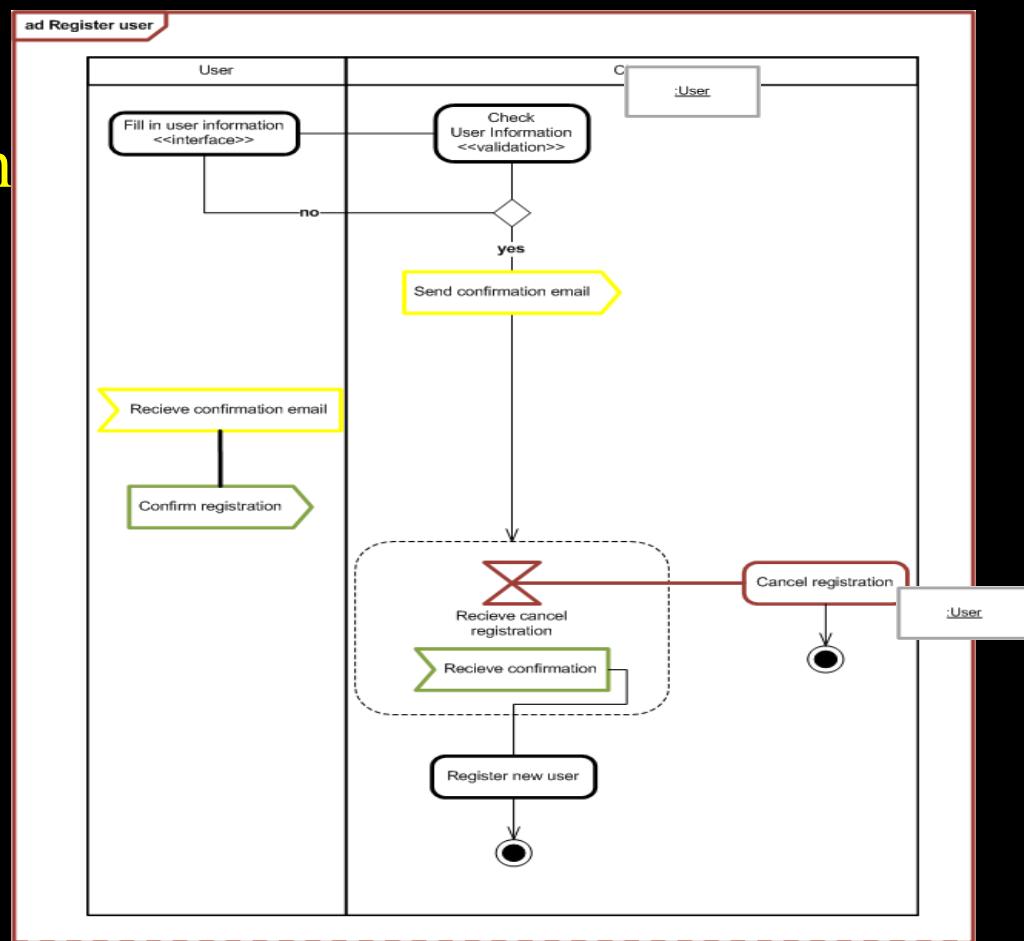
Primary flow

1. User fills out form with name, valid email, unique nickname, and optional phone number.
2. Submit the registration form.
3. System generates a user and sends out an email with logon information (username, one-time password) and a URL to the first-time login page.

Special requirement:

Submit button is deactivated. Page validates form in the background. Activates submit button when form is complete, name,nick, email.

Use Case Description *Activity Diagram* for RegisterUser

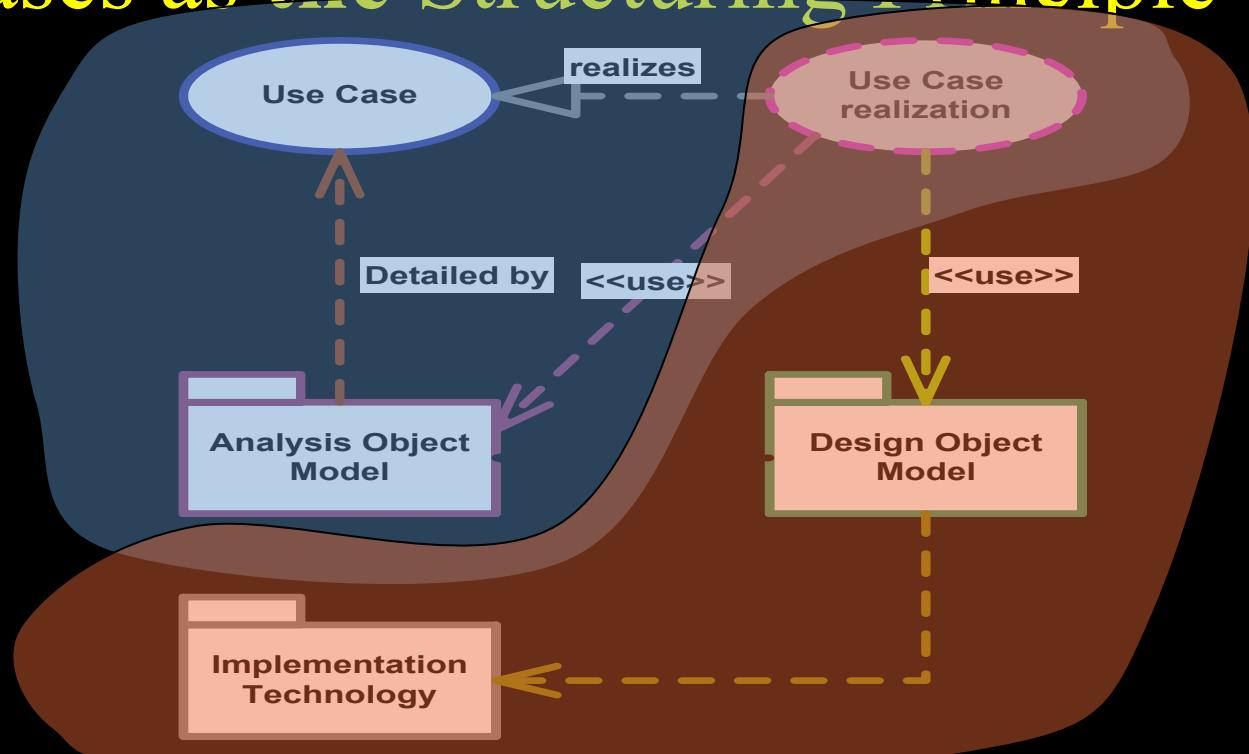


Use Cases - Known Problems/Issues

- ✓ Highly “functional approach” that tends to ignore quality attributes
- ✓ Looking for the positive first can mean never getting around to looking for Exceptions.
- ✓ “Fragmentation of problems”
- ✓ Drawing small blobs and stickmen but never writing good Scenarios



Use cases as the Structuring Principle



Other Use Cases for Use Cases

- ✓ Iteration **planning** during development
- ✓ Testing
 - Planning
 - Test case derivation – System and Subsystem testing
- ✓ Staff allocation

What about other requirements?

- ✓ Use cases are not enough.
- ✓ **Dictionary**, important (often neglected) for a shared understanding
- ✓ **Vision**, describe the target area for the project. An '*executive overview*'
- ✓ **Supplementary specification**, covers remaining requirements
 - FURPS+
 - Implementation constraints such as specific platforms, frameworks, etc.

FURPS+, variant from OpenUP process model

The System Wide Requirements Template

1. Introduction
 2. Statement of system-wide functional requirements
 - a. not expressed as use cases
 - b. Examples include auditing, authentication, printing, reporting .
- ✓ System Qualities - Qualities represent the URPS in FURPS+ classification of supporting requirements.

FURPS+

- ✓ **Usability** – Describe requirements for
 - easy of use,
 - easy of learning,
 - usability standards
 - Localization
- ✓ **Reliability** – system's ability to keep running under stress and adverse conditions.
 - Availability,
 - Frequency of severity of failures
 - Recoverability.

FURPS+

Performance –characteristics of the system should be outlined in this section.

- Response time, Throughput,
- Capacity, Memory footprint
- Startup or shutdown times.

Supportability –or maintainability of the system being built

- Adaptability, Upgrading, Compatibility
- Configurability, Scalability
- requirements regarding system installation
- Level of support, Maintenance

Today's takeaways

- ✓ Modeling Requirements
 - Behavioral Models for Functionality
 - Supplementary specifications for Quality
- ✓ UML Use Cases
- ✓ UML Activity Diagrams