Elicitation of requirements

Slides used in the video https://polimi365-my.sharepoint.com/:v:/g/personal/10143828_polimi_it/EQzGESnSzdBMnTIQvI9XG6QBV0zp715j6HqGqhHJ2Vx7qQ?e=Rx4Cab

Requirements elicitation in the RE process

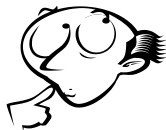**Requirements Models**

stakeholders

existing systems

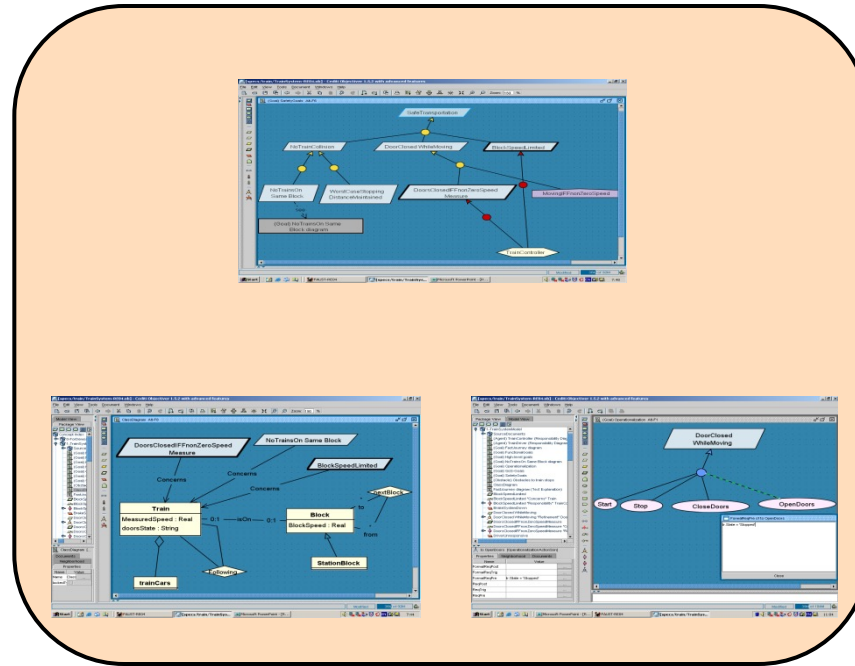*documents*
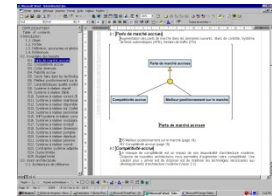
elicitation & modelling

generation of RE deliverables

requirements document

analysis & validation

Complexity example: rules and procedures for loan approval

- **Implicit knowledge:**
  - *There is no document in which the rules for approving loans are written down or the documents are ambiguous, incomplete, outdated*
- **Conflicting information:**
  - *Different members of the department have different ideas about what the rules are*
- **Bias:**
  - *The loan approval officers fear that your job is to computerize their jobs out of existence, so they are deliberately emphasizing the need for case-by-case discretion (to convince you it has to be done by a human!)*
- **Probe effect:**
  - *The loan approval process used by the officers while you are observing is different from the one they normally use*

How to cope with this complexity

- Adopting different approaches and strategies (listening, observing, studying, …) and combining the results achieved with all of them

- Being as close as possible to stakeholders

- Letting stakeholders describing their viewpoints

Scenarios

- "A narrative description of what people do and experience as they try to make use of computer systems and applications" [M. Carrol, Scenario-based Design, Wiley, 1995]

- A concrete, focused, informal description of a single feature of the system to be

Scenario example: warehouse on fire

- *Bob, driving down main street in his patrol car notices smoke coming out of a warehouse. His partner, Alice, reports the emergency from her car.*

- *Alice enters the address of the building, a brief description of its location (i.e., north west corner), and an emergency level. In addition to a fire unit, she requests several paramedic units on the scene given that area appears to be relatively busy. She confirms her input and waits for an acknowledgment.*

- *John, the Dispatcher, is alerted to the emergency by a beep of his workstation. He reviews the information submitted by Alice and acknowledges the report. He allocates a fire unit and two paramedic units to the Incident site and sends their estimated time of arrival (ETA) to Alice.*

- *Alice received the acknowledgment and the ETA.*

Observations

- Concrete scenario
  - Describes a single instance of reporting a fire incident
  - Does not describe all possible situations in which a fire can be reported

- Participating actors
  - Bob, Alice, and John

Heuristics for finding scenarios

- Ask yourself or the client the following questions:
  - Which user groups are supported by the system to perform their work?
  - What are the primary tasks that the system needs to perform?
  - What data will the actor create, store, change, remove or add in the system?
  - What external changes does the system need to know about?
  - What changes or events will the actor of the system need to be informed about?
- However, don't rely on questionnaires alone
  - Insist on **task observation** if the system already exists (interface engineering or reengineering)
  - Ask to speak to the end user, not just to the software contractor
  - Expect resistance and try to overcome it

So far…

- Scenarios provide a nice summary of what the requirements analysis team can derive from
  - Observation
  - Interviews
  - Analysis of documentation
- … they can be very specific
- How to abstract from details and specificities?
  - … Use cases!

After the scenarios are formulated…

- Generalize scenarios
  - Example: from "Warehouse on fire" we can generalize a "Report Emergency" use case
- Structure the description in terms of
  - Participating actors
  - Describe the Entry Condition
  - Describe the Flow of Events
  - Describe the Exit Condition
  - Describe Exceptions
  - Describe Special Requirements (Constraints, Nonfunctional Requirements)

Use case example: ReportEmergency

- Participating actors:
  - Field Officer (Bob and Alice in the Scenario)
  - Dispatcher (John in the Scenario)

After the scenarios are formulated…

- Generalize scenarios
  - Example: from "Warehouse on fire" we can generalize a "Report Emergency" use case
- Structure the description in terms of
  - Participating actors
  - Describe the Entry Condition
  - Describe the Flow of Events
  - Describe the Exit Condition
  - Describe Exceptions
  - Describe Special Requirements (Constraints, Nonfunctional Requirements)

Use case example: ReportEmergency

- Participating actors:
  - Field Officer (Bob and Alice in the Scenario)
  - Dispatcher (John in the Scenario)

- Entry Condition: True (an emergency can be reported all times)

After the scenarios are formulated…

- Generalize scenarios
  - Example: from "Warehouse on fire" we can generalize a "Report Emergency" use case
- Structure the description in terms of
  - Participating actors
  - Describe the Entry Condition
  - Describe the Flow of Events
  - Describe the Exit Condition
  - Describe Exceptions
  - Describe Special Requirements (Constraints, Nonfunctional Requirements)

Use case example: ReportEmergency

- Flow of events

  - The **FieldOfficer** activates the "Report Emergency" function of her terminal.

  - **FRIEND** (the system to be developed) responds by presenting a form to the officer.

  - The FieldOfficer fills the form, by selecting the emergency level, type, location, and brief description of the situation. The FieldOfficer also describes possible responses to the emergency situation. Once the form is completed, the FieldOfficer submits the form

  - At which point, the **Dispatcher** is notified.

  - The Dispatcher reviews the submitted information and allocates resources by invoking the AllocateResources use case. The Dispatcher selects a response and acknowledges the emergency report.

After the scenarios are formulated…

- Generalize scenarios
  - Example: from "Warehouse on fire" we can generalize a "Report Emergency" use case
- Structure the description in terms of
  - Participating actors
  - Describe the Entry Condition
  - Describe the Flow of Events
  - Describe the Exit Condition
  - Describe Exceptions
  - Describe Special Requirements (Constraints, Nonfunctional Requirements)

Use case example: ReportEmergency

- Participating actors: Field Officer, Dispatcher

- Entry Condition: True

- Flow of events: ….

- Exit condition: The FieldOfficer has received the acknowledgment and the selected response.

After the scenarios are formulated…

- Generalize scenarios
  - Example: from "Warehouse on fire" we can generalize a "Report Emergency" use case
- Structure the description in terms of
  - Participating actors
  - Describe the Entry Condition
  - Describe the Flow of Events
  - Describe the Exit Condition
  - Describe Exceptions
  - Describe Special Requirements (Constraints, Nonfunctional Requirements)

Use case example: ReportEmergency

- Exceptions:
  - The FieldOfficer is notified immediately if the connection between her terminal and the control room is lost
  - The Dispatcher is notified immediately if the connection between any logged in FieldOfficer and the control room is lost
- Special Requirements:
  - The FieldOfficer's report is acknowledged within 30 seconds;
  - The selected response arrives no later than 30 seconds after it is sent by the Dispatcher

Example of a poor use case

- Use case name: Accident   (not an action, i.e., a verb)
- Participating Actors:
  - Field Officer   (Dispatcher actor is missing here but mentioned in the next section)
- Flow of Events:
  - 1. The Field Officer reports the accident
  - 2. An ambulance is dispatched   (by whom?)
  - 3. The Dispatcher is notified when the ambulance arrives on site

  (Who notifies the Dispatcher?)

  Are we done at this point?

Tips to define proper use cases

- Use cases named with verbs that indicate what the user is trying to accomplish

- Actors named with nouns

- Use cases steps in active voice

- The causal relationship between steps should be clear

- A use case per user transaction

- Separate description of exceptions

- Keep use cases small (no more than two/three pages)

- The steps accomplished by actors and those accomplished by the system should be clearly distinguished

  - This helps us in identifying the boundaries of the system

Use case example: AllocateResources

- Actors:
  - ***Dispatcher:*** the Dispatcher allocates a resource to an Emergency if the resource is available (of course, he also updates and removes Emergency Incidents, Actions, and Requests in the system)

  - ***Resource Allocator:*** The Resource Allocator is responsible for allocating resources in case they are scarce

  - ***Resources:*** The Resources that are allocated to the Emergency

Allocate a resource (1)

- Use case name: AllocateResources
- Participating Actors:
  - Dispatcher (John in the Scenario)
  - Resource Allocator
  - Resources
- Entry Condition
  - An Incident has been opened
- Flow of Events
  - The Dispatcher selects the types and number of Resources that are needed for the incident
  - FRIEND replies with a list of Resources that fulfill the Dispatcher's request
  - The Dispatcher selects the Resources from the list and allocates them for the incident
  - FRIEND automatically notifies the Resources
  - The Resources send a confirmation

Allocate a resource (2)

- Exit Condition
  - The use case terminates when the resource is committed.
  - The selected Resource is now unavailable to any other Emergences or Resource Requests
- Exceptions
  - If the list of Resources provided by FRIEND is insufficient to fulfill the needs of the emergency, the Dispatcher informs the Resource Allocator
  - The Resource Allocator analyzes the situation and selects new Resources by decommitting them from their previous work
  - FRIEND automatically notifies the Resources and the Dispatcher
  - The Resources send a confirmation