



## Requirements Analysis

---

## Requirements Analysis

- encompasses all of the tasks that go into the instigation, scoping and definition of a new or altered system
- Analysts, along with software developers, identify the needs or requirements of a client.
- Once the client's requirements have been identified, the system designers are then in a position to design a solution.

## Requirements Analysis

- Also known as:
  - requirements engineering
  - requirements gathering
  - requirements capture
  - operational concept documenting
  - systems analysis
  - requirements specification

3

## Three Types of Activity

1. **Eliciting requirements:**
  - the task of communicating with customers and users to determine what their requirements are.
2. **Analyzing requirements:**
  - determining whether the stated requirements are unclear, incomplete, ambiguous, or contradictory and resolving these issues.
3. **Recording requirements:**
  - Requirements may be documented in various forms, such as natural-language documents, use cases, or user stories.

4

## Problems

### ■ Stakeholder issues

- Users don't understand what they want
- Users won't commit to a set of written requirements
- Users insist on new requirements after the cost and schedule have been fixed.
- Communication with users is slow
- Users often do not participate in reviews or are incapable of doing so.
- Users are technically unsophisticated
- Users don't understand the development process.

5

## Problems

### ■ Engineer/developer issues

- Technical personnel and end users may have different vocabularies. Consequently, they can believe they are in perfect agreement until the finished product is supplied.
- Engineers and developers may try to make the requirements fit an existing system or model, rather than develop a system specific to the needs of the client.
- Analysis may be often carried out by engineers or programmers, rather than personnel with the people skills and the domain knowledge to understand a client's needs properly.

6

## Main Techniques

- **Stakeholder interviews**
  - Stakeholder interviews are a common method used in requirement analysis.
  - Some selection is usually necessary, cost being one factor in deciding whom to interview.
- **Requirement workshops**
  - In some cases it may be useful to gather stakeholders together in "requirement workshops".
  - These workshops are more properly termed Joint Requirements Development (JRD) sessions, where requirements are jointly identified and defined by stakeholders.

7

## Main Techniques

- **Contract-style requirement lists**
  - One traditional way of documenting requirements has been contract style requirement lists.
  - In a complex system such requirements lists can run to hundreds of pages.

8

## Main Techniques

### ■ Measurable goals

- Best practices take the composed list of requirements merely as clues and ask why, repeatedly, until the actual business purposes are discovered.
- Then stakeholders and developers can devise tests to measure what level of each goal has been achieved so far.
- Once the small set of critical, measured goals has been established, rapid prototyping and short iterative development phases may proceed to deliver actual stakeholder value long before the project is half over.

9

## Main Techniques

### ■ Prototypes

- Prototypes help users get an idea of what the system will look like, and make it easier for users to make design decisions without waiting for the system to be built.
- Early views of the screens led to fewer changes later and hence reduced overall costs considerably.

10

## Main Techniques

### ■ Use cases

- A use case is a technique for capturing the potential requirements of a new system or software change.
- Each use case provides one or more scenarios that convey how the system should interact with the end user or another system to achieve a specific business goal.
- Use cases typically avoid technical jargon, preferring instead the language of the end user or domain expert.
- Use cases are often co-authored by software developers and end users.

11

## Main Techniques

- Use cases are deceptively simple tools for describing the behavior of the software.
- A use case contains a textual description of all of the ways which the intended users could work with the software through its interface.
- Use cases do not describe any internal workings of the software, nor do they explain how that software will be implemented.
- They simply show the steps that the user follows to use the software to do his work.
- All of the ways that the users interact with the software can be described in this manner.

12

## Main Techniques

- A use case should:
  - describe a business task to serve a business goal
  - be at an appropriate level of detail
  - be short enough to implement by one software developer in a single release

13

## Software Requirements Specification

- A SRS is a complete description of the behavior of the system to be developed.
- It includes a set of use cases that describe all of the interactions that the users will have with the software.
- In addition to use cases, the SRS contains functional requirements and nonfunctional requirements.

14

## Software Requirements Specification

- Functional requirements define the internal workings of the software:
  - the calculations,
  - technical details,
  - data manipulation and processing,
  - and other specific functionality that shows how the use cases are to be satisfied.
- Nonfunctional requirements impose constraints on the design or implementation (such as performance requirements, quality standards, or design constraints).

15

## Now, in your project ...

- Interview with us
  - Apart from the project meetings
  - Put a message to the newsgroup indicating your appropriate time-slots.
  - We will answer you.
- We suggest UML

16

## Now, in your project ...

- In your report, consider to include
  - Introduction
    - Project Definition
    - Goals and Scope
    - Constraints
  - Market Research
    - Similar systems, if any.

17

## Now, in your project ...

- Requirements
  - Functional, Non Functional, Hardware, Software, Data, Operational, Structural, etc.
- Usage Scenerios
- Diagrams
  - DFD, Use Case, CFD, State Transition Diagram, etc.
- Project Plan
  - Work packages (e.g. WBS)
  - Schedule (e.g. Gantt Chart)

18

## Now, in your project ...

- Some examples documents at
  - "<http://senior.ceng.metu.edu.tr/2004/>"
  - "<http://senior.ceng.metu.edu.tr/2005/>"
  - "<http://senior.ceng.metu.edu.tr/>"