



**SOFTWARE ENGINEERING**

**Week 6**

**System Modeling and Requirements Engineering**



## Agenda

- 1. System Engineering
- 2. Requirement Engineering
- 3. Software Requirements Document
- 4. Requirement Engineering Processes

System Modeling and Requirements Engineering

1.2

1. System Engineering ←  
2. Requirement Engineering  
3. Software Requirements Document  
4. Requirement Engineering Processes

# System Engineering

☞ 6.1 ☞

System Modeling and Requirements Engineering

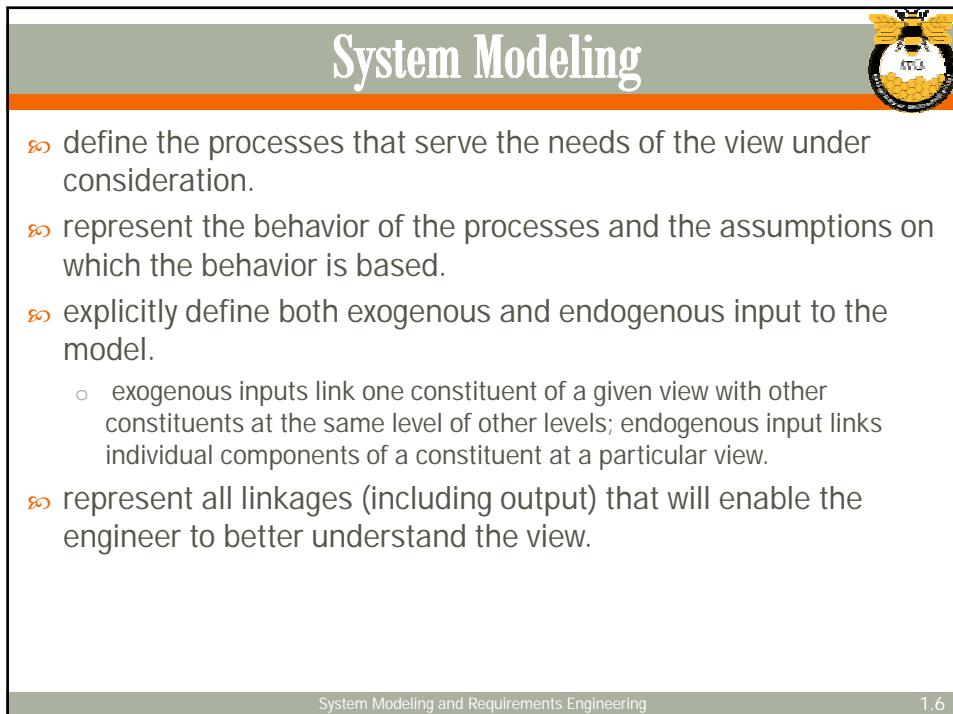
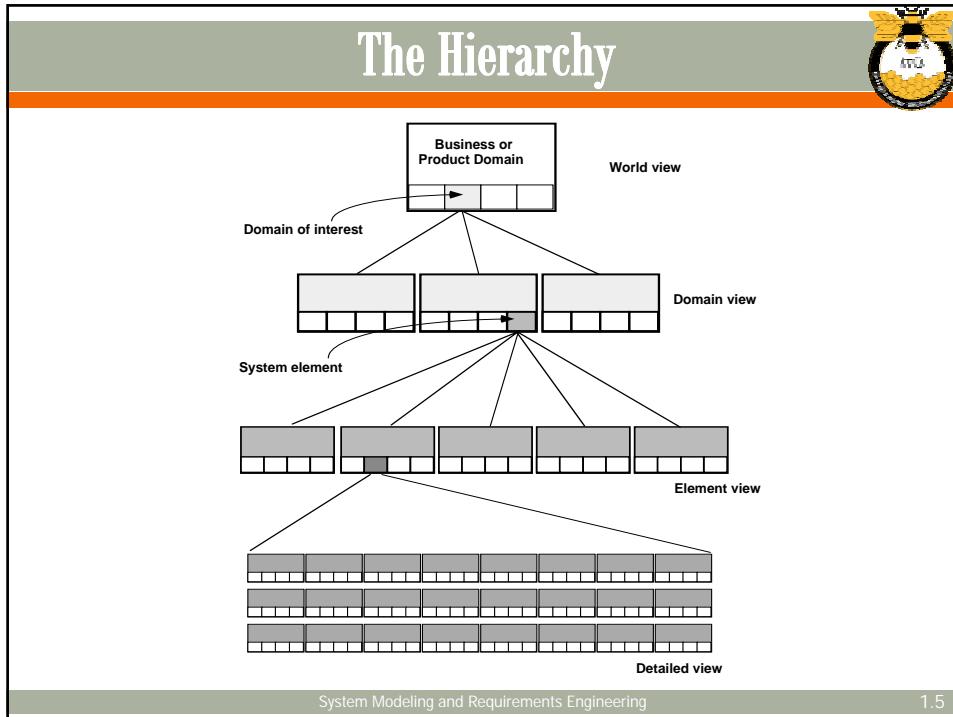
## System Engineering



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

System Modeling and Requirements Engineering

4



## Business Process Engineering



- „ uses an integrated set of procedures, methods, and tools to identify how information systems can best meet the strategic goals of an enterprise
- „ focuses first on the enterprise and then on the business area
- „ creates enterprise models, data models and process models
- „ creates a framework for better information management distribution, and control

## System Architectures



- „ Three different architectures must be analyzed and designed within the context of business objectives and goals:
  - data architecture
  - applications architecture
  - technology infrastructure
- „ *data architecture* provides a framework for the information needs of a business or business function
- „ *application architecture* encompasses those elements of a system that transform objects within the data architecture for some business purpose
- „ *technology infrastructure* provides the foundation for the data and application architectures

1. System Engineering
2. Requirement Engineering ←
3. Software Requirements Document
4. Requirement Engineering Processes

# Requirement Engineering

☞ 6.2 ☞

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

10

## What is a Requirement?



- ∞ A requirement is a software capability needed by the user to solve a problem to achieve an objective.
- ∞ A software capability that must be met by a system to satisfy a contract, standard, specification, or other formally imposed documentation.
- ∞ It is a statement describing either
  - 1) an aspect of what the proposed system must do,
  - 2) a constraint on the system's development.

11

## 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.

# Examples of 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.

# Target Stakeholders of Requirements



**User requirements**

- » Client managers
- System end-users
- Client engineers
- Contractor managers
- System architects

**System requirements**

- » System end-users
- Client engineers
- System architects
- Software developers

# Functional and Non-functional Requirements



## Functional requirements

- o Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.
- o May state what the system should not do.

## Non-functional requirements

- o Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.
- o Often apply to the system as a whole rather than individual features or services.

## Domain requirements

- o Constraints on the system from the domain of operation

# Functional Requirements



Describe functionality or system services.

Depend on the type of software, expected users and the type of system where the software is used.

Functional user requirements may be high-level statements of what the system should do.

Functional system requirements should describe the system services in detail.

**Functional requirements specify what the system should do.**

Some examples are:

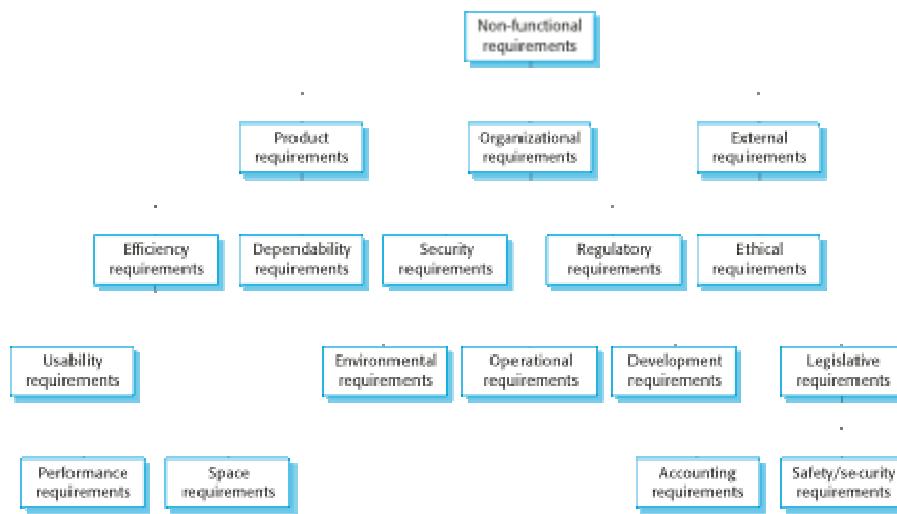
- o Calculate the discount at the rate of 14% for a customer spending amount
- o Number of significant digits to which accuracy should be maintained in all numerical calculations is 4
- o A book can be deleted from the Library Management System by the Database Administrator only

# Non-functional Requirements



- ☞ These define system properties and constraints e.g. reliability, response time and storage requirements. Constraints are I/O device capability, system representations, etc.
- ☞ Process requirements may also be specified mandating a particular IDE, programming language or development method.
- ☞ Non-functional requirements may be more critical than functional requirements. If these are not met, the system may be useless.
- ☞ **Non-functional requirements specify the overall quality attributes the system must satisfy.**
- ☞ Some examples are:
  - The response time of the system should always be less than 5 seconds
  - The software should be developed using C language on a UNIX based system
  - Experienced officers should be able to use all the system functions after a total training of two hours. After this training, the average number of errors made by experienced officers should not exceed two per day.

# Types of Non-functional Requirements



## Non-functional Requirements Implementation



- ☞ Non-functional requirements may affect the overall architecture of a system rather than the individual components.
  - For example, to ensure that performance requirements are met, you may have to organize the system to minimize communications between components.
- ☞ A single non-functional requirement, such as a security requirement, may generate a number of related functional requirements that define system services that are required.
  - It may also generate requirements that restrict existing requirements.

## Non-functional Classifications



- ☞ Product requirements
  - Requirements which specify that the delivered product must behave in a particular way e.g. execution speed, reliability, etc.
- ☞ Organisational requirements
  - Requirements which are a consequence of organisational policies and procedures e.g. process standards used, implementation requirements, etc.
- ☞ External requirements
  - Requirements which arise from factors which are external to the system and its development process e.g. interoperability requirements, legislative requirements, etc.

## Examples Of Nonfunctional Requirements



**Product requirement**  
The MHC-PMS shall be available to all clinics during normal working hours (Mon–Fri, 0830–17.30). Downtime within normal working hours shall not exceed five seconds in any one day.

**Organizational requirement**  
Users of the MHC-PMS system shall authenticate themselves using their health authority identity card.

**External requirement**  
The system shall implement patient privacy provisions as set out in HStan-03-2006-priv.

System Modeling and Requirements Engineering      1.21

1. System Engineering  
 2. Requirement Engineering  
 3. Software Requirements Document ←  
 4. Requirement Engineering Processes

# Software Requirements Document

6.3 ↵

System Modeling and Requirements Engineering

## Gathering Requirements



- ↪ Gathering is a systematic approach to eliciting, organizing, and documenting the requirements of the system.
- ↪ The set of requirements as a whole represents a negotiated agreement among the stakeholders.
- ↪ A collection of requirements are written as a "*Requirements Document*".

23

## Techniques for Gathering Requirements



1. Observation
2. Interviewing
3. Brainstorming
4. Prototyping
5. Use case analysis

24

## 1. Observation



- » Read documents and discuss requirements with users
- » Following important potential users as they do their work
  - o ask the user to explain everything he is doing
- » Session voice recording

25

## 2. Interviewing



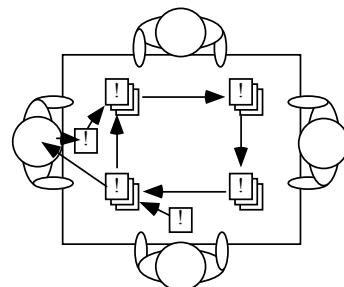
- The requirements team meet with the client and users to extract all relevant information
- Conduct a series of interviews
  - Ask about specific details
  - Ask about the stakeholder's vision for the future
  - Ask if they have alternative ideas
- After the interview, the interviewer must prepare a written report
  - Give a copy of the report to the person who was interviewed
- **Structured interview (Survey)**
  - » Specific preplanned questions are asked, frequently close-ended .
- **Unstructured interview**
  - » Questions are posed in response to the answers received, frequently open-ended.

26

## 3.Brainstorming



- *Joint Application Development (JAD)* technique
- Appoint an experienced moderator
- Arrange the attendees around a table
- Decide on a 'trigger question'
- Ask each participant to write an answer and pass the paper to its neighbour



27

## 4.Prototyping



- » The simplest kind: *paper prototype*.
  - a set of pictures of the system that are shown to users in sequence to explain what would happen
- » The most common: a mock-up of the system's user interface
  - Written in a rapid prototyping language (like Visual Basic)
  - Does *not* normally perform any computations, access any databases or interact with any other systems
  - May prototype a particular aspect of the system

28

## 5. Use case analysis



- » Determine the types of users (actors) that will use the facilities of this system
- » Determine the tasks that each actor will need to do with the system
- » Mostly used in Object-oriented system development

29

## The Software Requirements Document

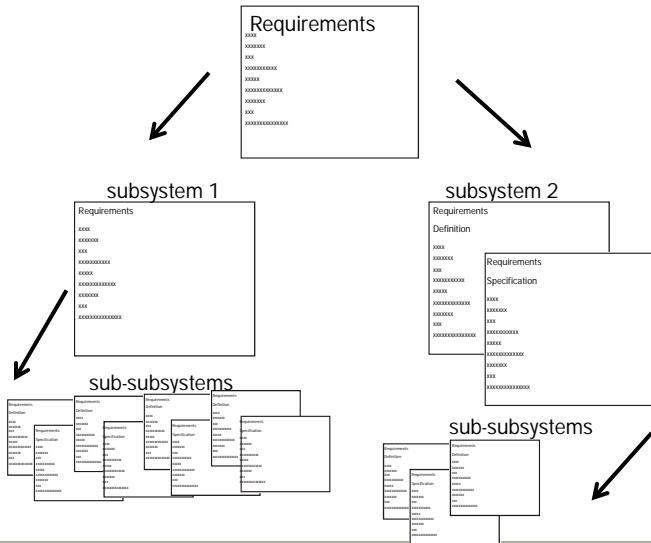


- » The software requirements document is the official statement of what is required of the system developers.
- » Should include both a definition of user requirements and a specification of the system requirements.
- » It is NOT a design document. As far as possible, it should set of WHAT the system should do rather than HOW it should do it.



## Hierarchy of Requirement Documents

- Requirements documents for large systems are normally arranged in a hierarchy.



31



## Agile Methods and Requirements

- Many agile methods argue that producing a requirements document is a waste of time as requirements change so quickly.
- The document is therefore always out of date.
- Methods such as XP use incremental requirements engineering and express requirements as 'user stories'
- This is practical for business systems but problematic for systems that require a lot of pre-delivery analysis (e.g. critical systems) or systems developed by several teams.

## Requirements Document Variability



- ↪ Information in requirements document depends on type of system and the approach to development used.
- ↪ Systems developed incrementally will, typically, have less detail in the requirements document.
- ↪ Requirements documents standards have been designed e.g. IEEE standard. These are mostly applicable to the requirements for large systems engineering projects.

## 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.



## 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.



## An Example Requirement Document

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

1. System Engineering  
 2. Requirement Engineering  
 3. Software Requirements Document  
 4. Requirement Engineering Processes ←

# Requirement Engineering Processes

6.4

System Modeling and Requirements Engineering

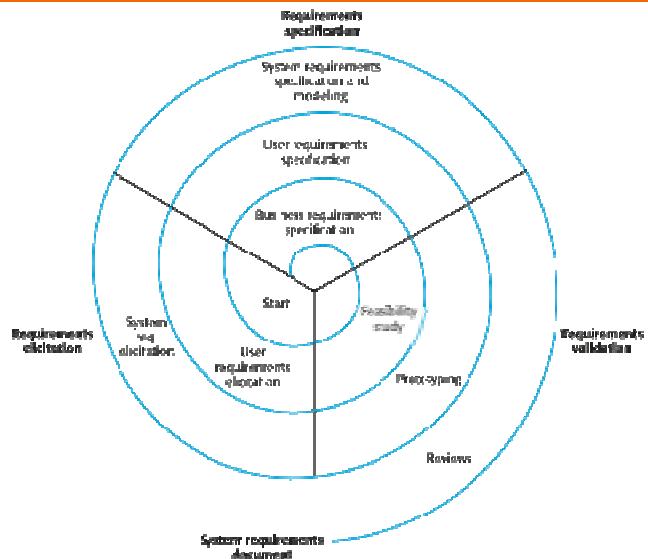
## Requirements Engineering Processes



- **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 1.38

## A Spiral View Of The Requirements Engineering Process



System Modeling and Requirements Engineering

1.39

## 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

1.40

# Inception



- ¤ Identify stakeholders
  - "who else do you think I should talk to?"
- ¤ Recognize multiple points of view
- ¤ Work toward collaboration
- ¤ The first questions
  - Who is behind the request for this work?
  - Who will use the solution?
  - What will be the economic benefit of a successful solution
  - Is there another source for the solution that you need?

# Eliciting Requirements



- ¤ meetings are conducted and attended by both software engineers and customers
- ¤ rules for preparation and participation are established
- ¤ an agenda is suggested
- ¤ a "facilitator" (can be a customer, a developer, or an outsider) controls the meeting
- ¤ a "definition mechanism" (can be work sheets, flip charts, or wall stickers or an electronic bulletin board, chat room or virtual forum) is used
- ¤ the goal is
  - to identify the problem
  - propose elements of the solution
  - negotiate different approaches, and
  - specify a preliminary set of solution requirements

## Elicitation Work Products



- » a statement of need and feasibility.
- » a bounded statement of scope for the system or product.
- » a list of customers, users, and other stakeholders who participated in requirements elicitation
- » a description of the system's technical environment.
- » a list of requirements (preferably organized by function) and the domain constraints that apply to each.
- » a set of usage scenarios that provide insight into the use of the system or product under different operating conditions.
- » any prototypes developed to better define requirements.

## Elaboration



- » create an analysis model that identifies data, function and behavioral requirements
- » these activities are covered in «*Week 8 – Analysis Model*»

## Negotiating Requirements



### Identify the key stakeholders

- These are the people who will be involved in the negotiation

### Determine each of the stakeholders “win conditions”

- Win conditions are not always obvious

### Negotiate

- Work toward a set of requirements that lead to “win-win”

## Requirements Validation



Concerned with demonstrating that the requirements define the system that the customer really wants.

Requirements error costs are high so validation is very important

- Fixing a requirements error after delivery may cost up to 100 times the cost of fixing an implementation error.

### Requirements Checking

- **Validity.** Does the system provide the functions which best support the customer's needs?
- **Consistency.** Are there any requirements conflicts?
- **Completeness.** Are all functions required by the customer included?
- **Realism.** Can the requirements be implemented given available budget and technology
- **Verifiability.** Can the requirements be checked?



## Requirements Validation Techniques

- » Requirements reviews
  - Systematic manual analysis of the requirements.
- » Prototyping
  - Using an executable model of the system to check requirements.
- » Test-case generation
  - Developing tests for requirements to check testability.



## Requirements Reviews

- » Regular reviews should be held while the requirements definition is being formulated.
- » Both client and contractor staff should be involved in reviews.
- » Reviews may be formal (with completed documents) or informal. Good communications between developers, customers and users can resolve problems at an early stage.
- » Review Checks
  - **Verifiability**
    - Is the requirement realistically testable?
  - **Comprehensibility**
    - Is the requirement properly understood?
  - **Traceability**
    - Is the origin of the requirement clearly stated?
  - **Adaptability**
    - Can the requirement be changed without a large impact on other requirements?

## Requirements Management



- ☞ Requirements management is the process of managing changing requirements during the requirements engineering process and system development.
- ☞ New requirements emerge as a system is being developed and after it has gone into use.
- ☞ You need to keep track of individual requirements and maintain links between dependent requirements so that you can assess the impact of requirements changes. You need to establish a formal process for making change proposals and linking these to system requirements.

## Changing Requirements



- ☞ The business and technical environment of the system always changes after installation.
  - New hardware may be introduced, it may be necessary to interface the system with other systems, business priorities may change (with consequent changes in the system support required), and new legislation and regulations may be introduced that the system must necessarily abide by.
- ☞ The people who pay for a system and the users of that system are rarely the same people.
  - System customers impose requirements because of organizational and budgetary constraints. These may conflict with end-user requirements and, after delivery, new features may have to be added for user support if the system is to meet its goals.
- ☞ Large systems usually have a diverse user community, with many users having different requirements and priorities that may be conflicting or contradictory.
  - The final system requirements are inevitably a compromise between them and, with experience, it is often discovered that the balance of support given to different users has to be changed.

# Requirements Management Planning



- ∞ Establishes the level of requirements management detail that is required.
- ∞ Requirements management decisions:
  - **Requirements identification** Each requirement must be uniquely identified so that it can be cross-referenced with other requirements.
  - **A change management process** This is the set of activities that assess the impact and cost of changes. I discuss this process in more detail in the following section.
  - **Traceability policies** These policies define the relationships between each requirement and between the requirements and the system design that should be recorded.
  - **Tool support** Tools that may be used range from specialist requirements management systems to spreadsheets and simple database systems.

# Requirements change management



- ∞ Deciding if a requirements change should be accepted
  - **Problem analysis and change specification**
    - During this stage, the problem or the change proposal is analyzed to check that it is valid. This analysis is fed back to the change requestor who may respond with a more specific requirements change proposal, or decide to withdraw the request.
  - **Change analysis and costing**
    - The effect of the proposed change is assessed using traceability information and general knowledge of the system requirements. Once this analysis is completed, a decision is made whether or not to proceed with the requirements change.
  - **Change implementation**
    - The requirements document and, where necessary, the system design and implementation, are modified. Ideally, the document should be organized so that changes can be easily implemented.

## 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)