

Software Requirements Engineering

Lecture 04

Topics to be discussed

- Outlines
 - Requirements
 - Functional Requirements
 - Non-Functional Requirements

Objectives and outcomes

- *Objectives*

- The objective of this chapter is to introduce software requirements and
- to discuss the processes involved in discovering and documenting these requirements.

- *Outcomes*

- After taking this lecture students should be able to:
- Understand requirements
- Learn different types of requirements

References

- Software Engineering: Ian Sommerville chapter 6th

What is Requirement

- **Defining Requirement**

- Requirement is a condition or capability possessed by the software or system component in order to solve a real world problem.
- The problems can be to automate a part of a system, to correct shortcomings of an existing system, to control a device, and so on.

Types of Requirements

- **Requirement Types**
 1. Functional Requirements
 2. Non-Functional Requirements
 3. Software Requirements
 4. Hardware Requirements
 5. Domain Requirements
 6. User Requirements
 7. System Requirements

Types of Requirements

- ***Functional Requirement:***

- A functional requirement is a **function or feature** that must be included in a system in order to satisfy the business need and be acceptable to the users.

- **Examples: Inputs, Outputs, Processes, Stored data**

Functional Requirement Example

- **Functional Requirement Examples**

- Software should be able to help **students view marks, attendance**
- Software should be have **online fee payment facility** etc.
- Software should be able to help **teachers view students' attendance, to upload marks, and perform other daily activities** etc.

Types of Requirements

- ***Non-Functional Requirement:***

- Non-functional requirements, as the name suggests, are requirements that are not directly concerned with the specific functions delivered by the system.
- **Non-Functional Requirements:** are also called **Quality Requirements**

- **Examples:**

Performance, Security, Efficiency, usability, maintainability, availability

Types of Non Functional Requirement

1. Product Requirements

1. Efficiency
2. Reliability
3. Safety
4. Portability
5. Usability

2. Organizational Requirement

1. Delivery Requirements
2. Implementation Requirements
3. Standard Requirements

3. External Requirements

1. Ethical Requirements
2. Legislative Requirements
3. Interoperability Requirements

1. Product Requirements

- **Product requirements:**

- These requirements specify how software product performs. Product requirements comprise the following.

- 1. Efficiency requirements:**

- Describe the extent to which the software makes **optimal use of resources**, the speed with which the system executes, and the memory it consumes for its operation. For example, the system should be able to operate at least three times faster than the existing system.

- 2. Reliability requirements:**

- Describe the **acceptable failure rate** of the software. For example, the software should be able to operate even if a hazard occurs.

1. Product Requirements

3. Safety Requirements:

- Sometimes the database may get crashed at any certain time due to virus or operating system failure.
- Therefore, it is required to take the database backup

4. Portability requirements:

- Describe the ease with which the software can be transferred from one platform to another.
- For example, it should be easy to port the software to a different operating system without the need to redesign the entire software.

1. Product Requirements

5. Usability requirements:

- Describe the ease with which users are able to operate the software.
- For example, the software should be able to provide access to functionality with fewer **keystrokes and mouse clicks.**

2. Organizational Requirements

- **Organizational requirements:**

- These requirements are derived from the policies and procedures of an organization. Organizational requirements comprise the following.

1. **Delivery requirements:**

- Specify when the software and its documentation are to be delivered to the user.

2. **Implementation requirements:**

- Describe requirements such as programming language.

3. **Standards requirements:**

- Describe the process standards to be used during software development.
- For example, the software should be developed using standards specified by the ISO and IEEE standards.

3. External Requirements

- **External requirements:**
 - **External requirements** are things that **come from outside the software** or its development team but **still affect how it must be built, function, or behave**. These are usually **rules, standards, or expectations** set by other systems, organizations, or authorities.
 - External requirements comprise the following.
 1. Interoperability Requirements
 2. Ethical Requirements
 3. Legislatives Requirements

Non-Functional Requirement

1. Interoperability requirements:

- These define how the software must **interact or work with other systems**.
- For example: If your software must connect to a government database or work with another company's app, you need to follow certain rules or standards so both systems can "understand" each other.
- *Example: "The website must integrate with PayPal, Visa, and MasterCard payment gateways for processing customer transactions."*

Non-Functional Requirement

1. Ethical requirements:

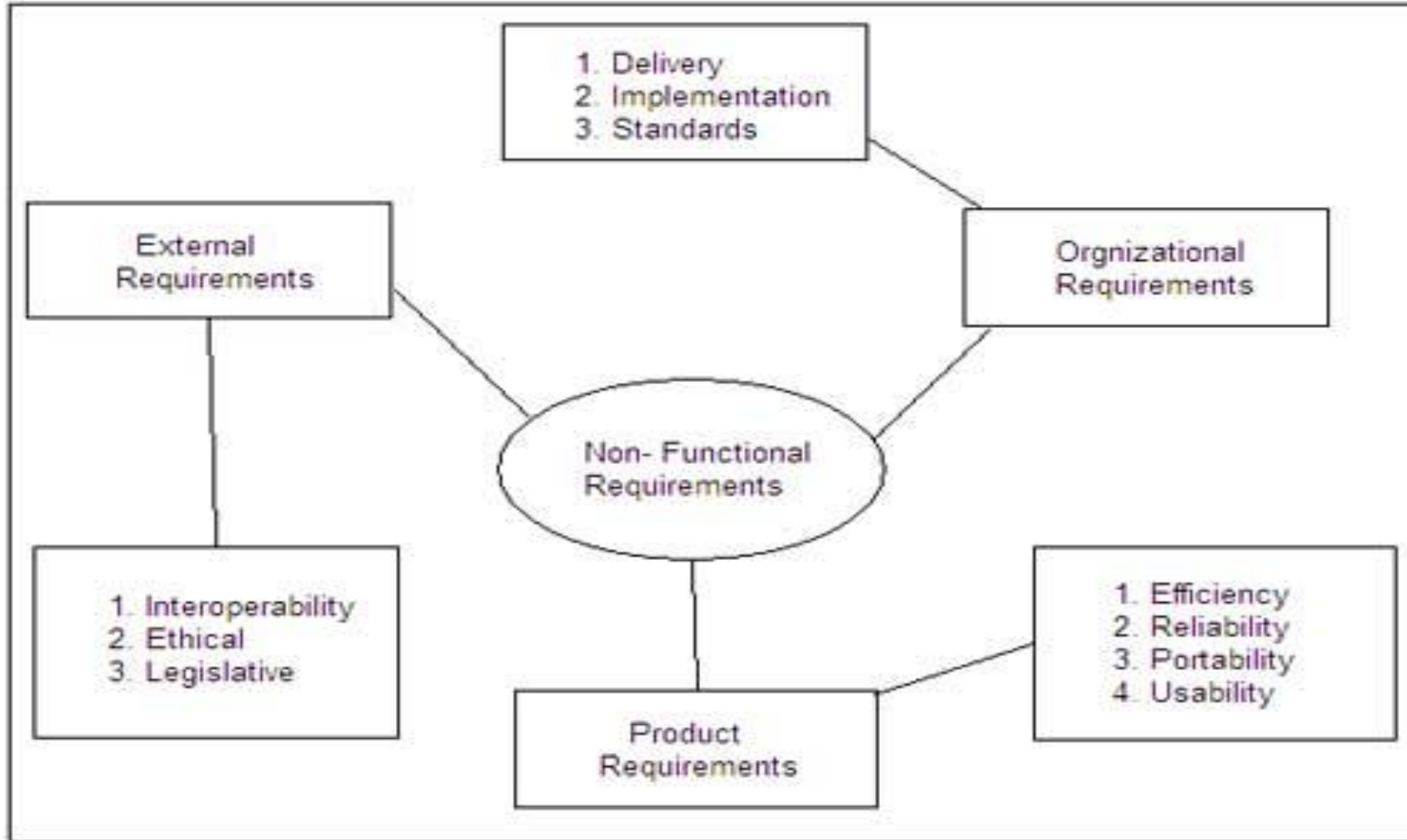
- These are rules about how to **behave responsibly and fairly**, especially with sensitive information.
- For example: If your software collects personal data, it must not misuse it, must protect it from unauthorized access, and must respect user privacy.
- Think about fairness, transparency, and respecting user rights.
- Example: An app should not collect more user data than it really needs.

Non-Functional Requirement

1. Legislative requirements:

- These are **laws and regulations** that the software must comply with. They often depend on where the software is used.
- For example: GDPR (General Data Protection Regulation) in the EU requires user data to be handled in specific ways.
- If your software breaks these laws, the company could face **legal action** or **penalties**.
- Example: Financial software must comply with anti-money laundering laws.

Non-Functional Requirement



Domain Requirement

- *Domain Requirements*
- **Domain requirements** are specific rules, behaviors, or constraints that come from the **problem area (domain)** where the software will be used.
- These are **not general software rules** — they are **specific to the industry or field**.
- They come from **domain experts, business rules, or operational needs**.

Domain Requirement

Examples of Domains:

- **Healthcare**
- **Banking**
- **Education**
- **E-commerce**
- **Telecommunication**

User Requirement

- *What is User Requirements?*
- **User requirements** describe **what the user needs or expects** the software to do.
They are usually written in **plain language** (non-technical) so **both developers and clients** can understand them.
- Think of them as the "**wish list**" from the users' point of view.
- Example: "The user should be able to check their account balance from the mobile app at any time."

User Requirement

- User requirements are often the **first step** in the software development process.
- They are later turned into more detailed **system requirements** or **function**.
- Example : “As a [type of user], I want to [do something], so that [reason/benefit].”
- “As a student, I want to download my exam schedule, so that I can prepare in advance.”

System Requirements

- *What is System Requirements?*

- System requirements are more detailed descriptions of the software system's functions, services, and operational limits.
- The system requirements document (sometimes called a functional specification) should **define exactly what is to be implemented**.
- It may be part of the contract between the system buyer and the software developers.

Software Requirement Examples

- **For development, software's used are:**

- Operating System : Windows-XP Professional
- Version : Version 2002, service pack-2,
- Platform : .Net
- Language : C#(C sharp)
- Backend : SQL server 2007
- Editor : Microsoft word processor 2000
- For Development : visual studio .Net setup tools
- For Design : HTML, Visual studio designer

Software Requirement Examples

- **For deployment, software's used are :**

- Operating system : Windows-xp professional
- Version : Version 2002, service pack-2
- Framework : .Net 2005
- IIS (Internet Information Services) : Configured
- Backend : sql server 2000

Hardware Requirements Example

- *For Development, Hardware's used are:*

- Core i7 (core i7 processor operating at 3.6 GHz)
- 1TB hard disk
- 32GB RAM

Hardware Requirements Example

- *For Deployment, Hardware's used are :*

- Core I3 or better processor
- 20 GB hard disk
- 4 GB RAM

User interface Requirements

- *Good user Interface*

- A system is said be good if it provides means to use it efficiently. User interface requirements are briefly mentioned below –
 - Content presentation
 - Easy Navigation
 - Simple interface
 - Responsive
 - Consistent UI elements
 - Feedback mechanism
 - Default settings
 - Strategic use of color and texture.
 - Provide help information
 - User centric approach

Requirement Engineering

Requirement Engineering

- **Definition:** The process to gather the software requirements from client, analyze and document them is known as requirement engineering.
- **Goal:** The goal of requirement engineering is to develop and maintain sophisticated and descriptive 'System Requirements Specification' document.

Requirements Engineering

- It is essential that the software engineering team understand the requirements of a problem before the team tries to solve the problem.
- RE is software engineering actions that start with communication activity and continues into the modeling activity.
- RE establishes a solid base for design and construction. Without it, resulting software has a high probability of not meeting customer needs.

Why is Getting Good Requirements Hard?

- 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 RE process.
- New stakeholders may emerge and the business environment change.

Requirements Engineering Tasks

- Inception
- Elicitation
- Elaboration (Highly structured)
- Negotiation
- Specification
- Validation
- Requirements management

Inception

- Inception— Ask “context-free” questions that establish ...
 - Basic understanding of the problem
 - The people who want a solution
 - The nature of the solution that is desired, and
- Through these questions, the requirements engineer needs to...
- Identify the **stakeholders**
 - Recognize multiple **viewpoints**

Initiating Requirements Engineering Process

- **Identify stakeholders**

- Stakeholder can be “anyone who benefits in a direct or indirect way from the system which is being developed”

Ex. Business manager, project manager, marketing people, software engineer, support engineer, end-users, internal-external customers, consultants, maintenance engineer.

- Each one of them has different **view of the system.**

Elicitation

- Elicitation – elicit (draw) requirements from customers, users and others.
 - Find out from customers, users and others what the • what is to product objectives are
be done
 - how the product fits into business needs, and
 - how the product is used on a day to day basis

Elaboration

- Focuses on developing a refined technical model of software functions, features, and constraints using the information obtained during **inception and elicitation**
- Create an **analysis model** that **identifies data, function and behavioral** requirements.

Negotiation

- **Agree on a deliverable** system that is **realistic for developers** and **customers**
 - Requirements are **categorized** and organized into subsets
 - Requirements reviewed for **correctness**
 - Requirements **prioritized** based on customer needs
 - Negotiation about requirements, **project cost and project timeline.**
 - There should be **no winner and no loser** in effective negotiation.

Specification

- It can be
 - Written Document
 - A set of graphical models,
 - A formal mathematical models
 - Collection of usage scenario.
 - A prototype
- For large systems, **written document, language descriptions, and graphical models** may be the best approach.
- For small systems or products, **usage scenarios**

Requirement Management

- Set of activities that help project team to **identify, control, and track requirements and changes as project proceeds**

Validation

- Requirements Validation – a review mechanism that looks for
 - Errors in content or **interpretation**
 - Areas where **clarification** may be required
 - **Missing** information
 - **Inconsistencies**
 - **Conflicting or unrealistic** (unachievable) requirements.

Summary

- Problems with requirements practices
- Requirements engineering & Tasks
 - Inception
 - Elicitation
 - Elaboration
 - Negotiation
 - Specification
 - Validation
 - Requirements managements

Fact Finding Techniques

Fact Finding Techniques

- **What is fact finding?**

- Fact finding is process of **collection of data and information** based on techniques which contain **sampling of existing documents, research, observation, questionnaires, interviews, prototyping and joint requirements planning.**
- System analyst uses suitable fact-finding techniques to develop and implement the current existing system.
- Following are some of the techniques used for fact finding:

Sampling

• What is Sampling?

- Sampling is the process of selecting a **small group** from a **larger set** of data to **analyze and make conclusions** about the **whole set**.

ESTIMATING WORK SHEET											
PREPARED BY	E. B. JASON	MOJ. LOCATION	DIEGO GARCIA	SHEET	1	OF	5	DRAWING NO.	1,337,444/1/104/100	PROJ. TITLE	CANTONMENT AREA WATER SYSTEM BLDG.
CHECKED BY	F. J. AMERIKIAN	PROJ. SECTION	ANALYSIS/ESTIMATE	ACTIVITY NO.	NOOK 31 TO NOOK 34	BM NO.	D/W-112	MYO NO.		DATE PREPARED	19 FEB 85
ITEM NO.	DESCRIPTION	PREFAB FORMS	REFER TO DIM QD SECT. V. DRAW B1C PP 7 & 8	BM NO.	BM LINE ITEM NO.	UNIT OF ISSUE	TOTAL QTY	REMARKS	USE LOC. PROCEDURES TO		
		BUILDING FOOTING	L W T					SLAB/FOOTING			
1.	3/4" PLYWOOD	2(26'-8") + (26'-12" + 53'-11" + 45'-0") = 156'-8"	20'-0" 12"			SH	3	EDGE FORMS-TO BE USED AT TRANSMITTER SITE BLDG.			
	BB EXTERIOR TYPE	4' x 8'	8' x 4' PLYWOOD SUPPLY 12' x 32'								
			DB 33/2'S 3 SHEETS								
2.	LUMBER 1" x 4" x 9L	4 LENGTHS x 2 EA CORNER				BF	30	BLDG LAYOUT			
	GR 2 OR BETTER	x 4 CORNERS = 8 PCS / 6 LONG						BATTER BOARDS			
3.	LUMBER 2" x 4" x 9L	16' - 4R PCS - 16' x 2" x 4" x 8R PCS				BF	590	USE REUSABLE			
								2" x 4" AT TRANSMITTER SITE BUILDING			
		RAMP AND DOOR STOOP FORMS									
4.	3/4" PLYWOOD 4' x 8'	(15'-8") + (2(6') + 3(4')) = 37'-8"				SH	1	EDGE FORMS REUSE AT TRANSMITTER SITE BUILDING			
		RP PLYWOOD INTO 8" STRIPS = 6 X 8 x 48									
		BEAMS									
5.	3/4" PLYWOOD GR BB	B-1	2 EACH 26'-8" BOND BEAMS					B-1 SIDE FORMS			
	EXT TYPE 4' x 8'	26'-8" x 4 SIDES + 106'-8"				SH	10	REUSE AT TRANSMITTER SITE BUILDING			

BUNP013

(A) Item Name: DEXY CYCLES MONOR
A.R. NO.: 174 811715

(B) Details on Q. R. Note:

- 1) Manufactured by / supplied by Present / Not Present
- 2) Item Code / Item Name Present / Not Present
- 3) Approved Manufacturer / Supplier Present / Not Present
- 4) Chassis No. Present / Not Present
- 5) Purchase Order No. Present / Not Present
- 6) Q. R. N. Number / Doc. No. Present / Not Present
- 7) Date of Receipt Present / Not Present
- 8) No. of Containers Present / Not Present
- 9) Manufacturing Date Present / Not Present
- 10) Expiry date Present / Not Present
- 11) Pharmaceutical Grade Present / Not Present
- 12) Batch No. Present / Not Present

(C) Details on Consignment:

- 1) As per Standard Packing Present / Not Present
- 2) Physical Condition of the Consignment Present / Not Present
- 3) Labeling on the Consignment
 - a) Pharmaceutical Grade Present / Not Present
 - b) Product Name Present / Not Present
 - c) Batch No. Present / Not Present
 - d) Name of Manufacturer / Supplier Present / Not Present
 - e) Manufacturing Date Present / Not Present
 - f) Expiry Date / Reuse Date Present / Not Present
 - g) Colour Index No. (For colour only) Present / Not Present
- 4) Storage Present / Not Present

(D) Certificate:

- 1) SSE free certificate from manufacturer Present / Not Present
- 2) Checked the manufacturer's COA with respect to acceptance criteria and manufacturing and tested / expiry date Present / Not Present

REMARK:

Observation

- **What is observation?**

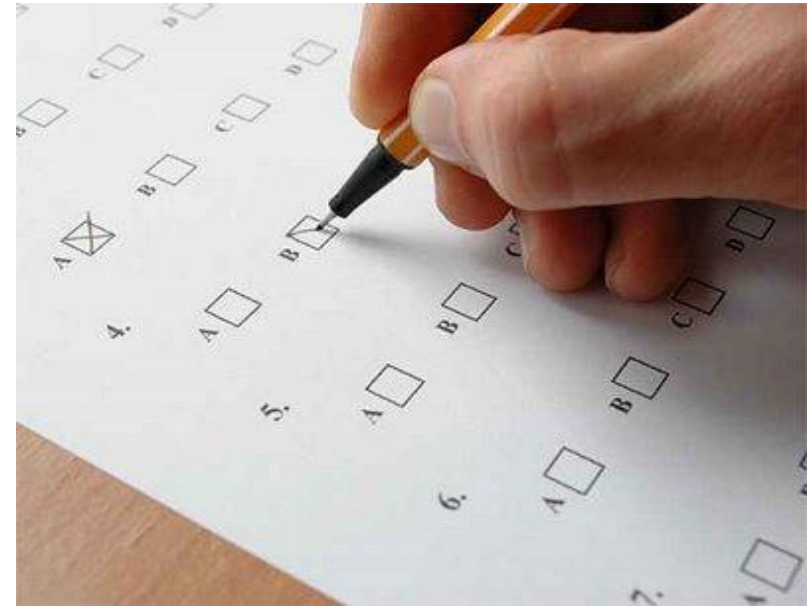
- Observation is the act of watching, noticing, or recording something carefully to learn or understand the system.



Questionnaires

- **What is Questionnaire?**

- A questionnaire is a set of written questions used to collect information from people.
- Questionnaires are special-purpose documents that allow the analyst to collect information and opinions from respondents.



Types of Questionnaires

- *Types of Questionnaire?*

- Free format Questionnaire: Free-format questionnaires offer the respondent greater freedom in the answer. A question is asked, and the respondent records the answer in the space provided after the question.
- Fixed Format Questionnaire: Fixed-format questionnaires contain questions that require selection of predefined responses from individuals.

Questionnaire Procedure

- *Procedure for questionnaire?*

1. Determine what facts and opinions those must be collected and from whom you should get them?
2. Based on the facts, determine whether free or fixed-format questionnaires will be required.
3. Write the questions.
4. Test the questions on a small sample of respondents.
5. Duplicate and distribute the questionnaire.

Interviews

What is Interviews?

- Interview is a fact-finding technique whereby the systems analysts collect information from individuals through face-to-face interaction.



Types of Interview Questions

- Types of Interview Questions

- Open-ended questions: Open-ended questions allow the interviewee to respond in any way that seems appropriate.
- Closed-ended questions: Closed-ended questions restrict answers to either specific choices or short, direct responses.

Procedure to Conduct an Interview

- **Procedure to conduct an interview**

1. Select Interviewees

2. Prepare for the Interview

- An interview guide is a checklist of specific questions the interviewer will ask the interviewee.

3. Conduct the Interview

4. Follow Up on the Interview

Joint Requirements Planning

- What is Joint Requirement Planning?

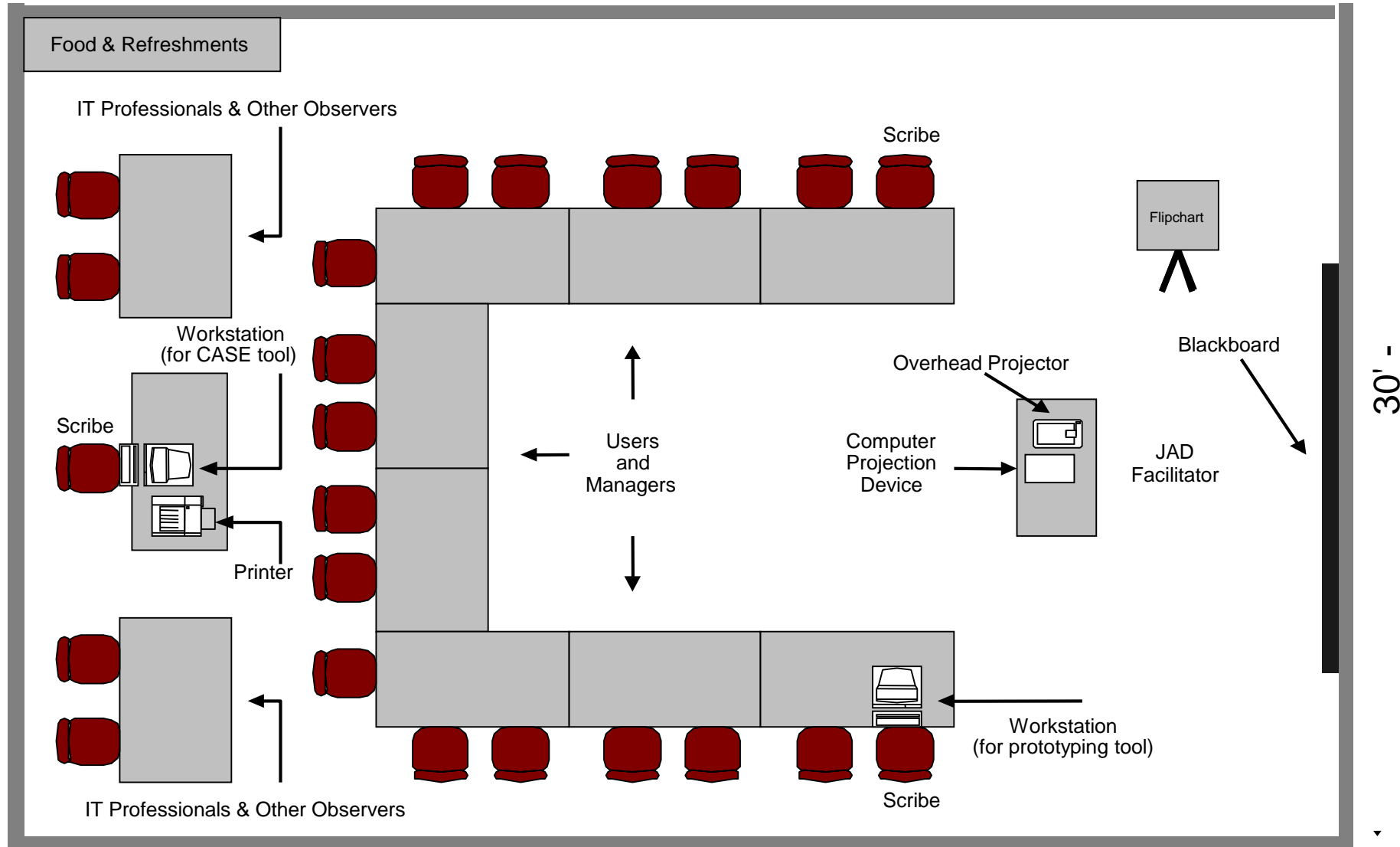
- Joint requirements planning (JRP) is a process whereby highly structured group meetings are conducted for the purpose of analyzing problems and defining requirements.



JRP Participants

- Who are the participants in JRP?
 - Sponsor
 - Facilitator
 - Users and Managers
 - Computers and other equipment
 - I.T. Staff

Typical room layout for JRP session



Benefits of JRP

- More better communication with all people involve in software development.
- Saves time

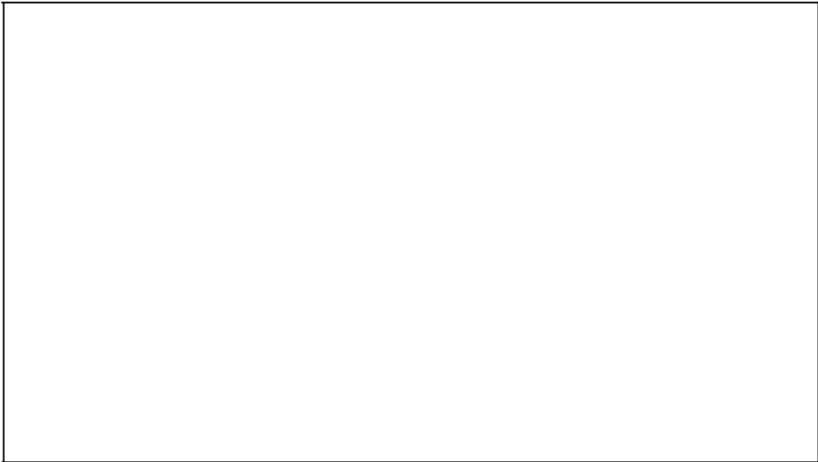
Storyboarding

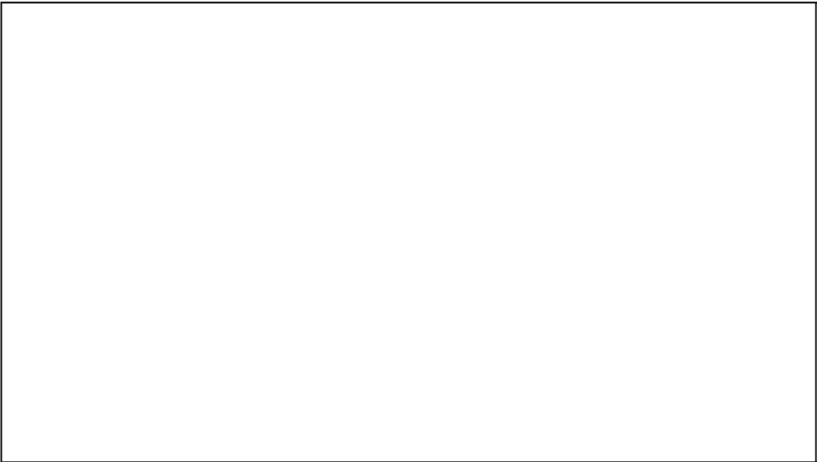
- Storyboarding is a powerful and simple technique for capturing ideas about the form and appearance of a system

Storyboarding

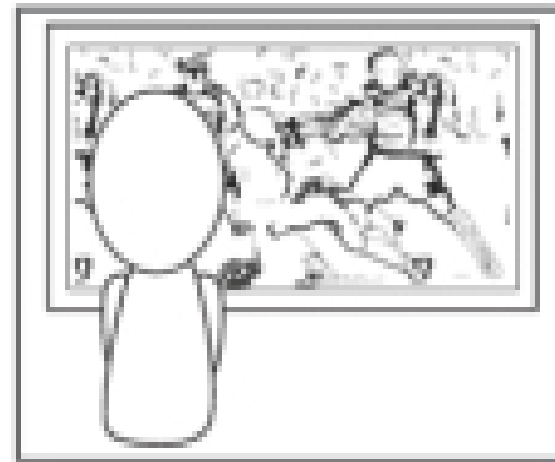
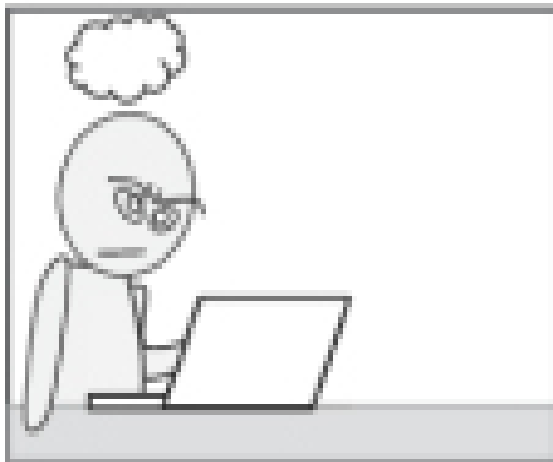
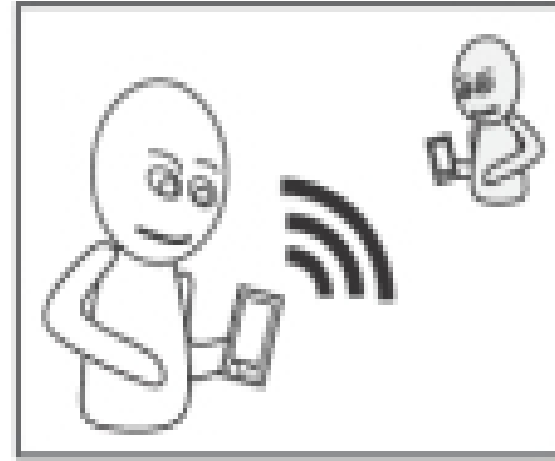
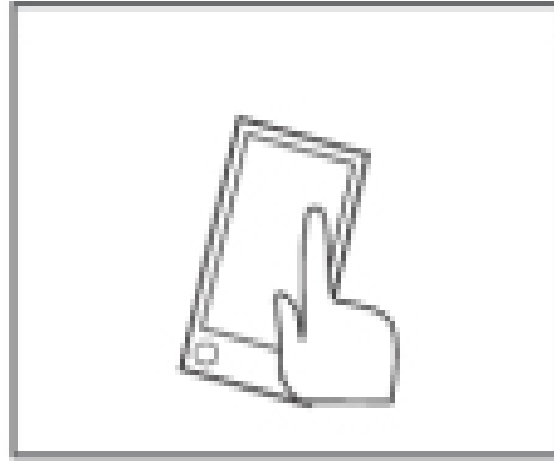
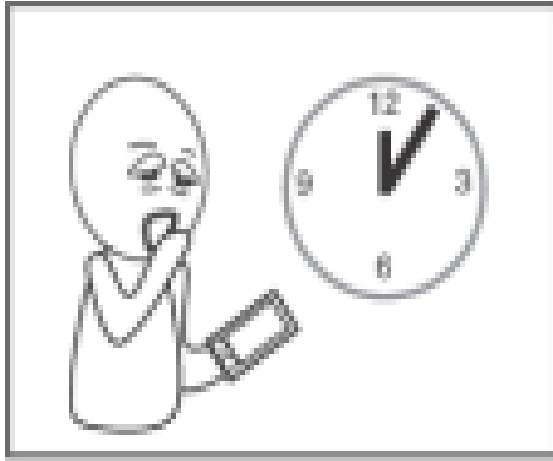
- Common uses
 - Data visualization
 - Report layout
 - Screen layouts
 - Screen flows

The storyboarding sample paper

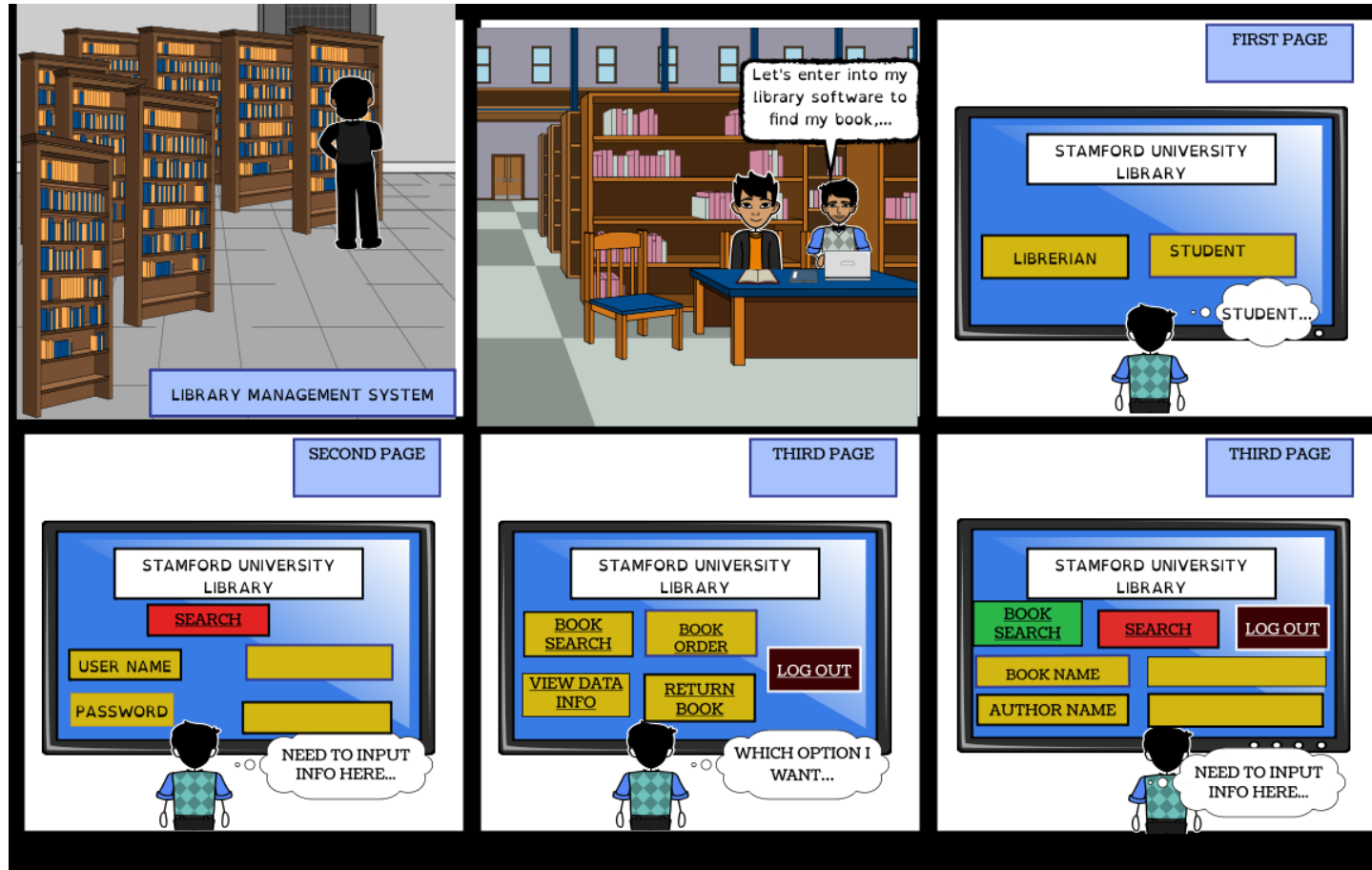




Story boarding example for animated movie



Library Management System



Thank You