

Requirements Engineering (Part 1)

Lecture 4

Agenda

- Requirements Definition vs. Requirements Specification.
- Functional vs. Non-functional Requirements
- Case Study
- Requirements Engineering Process:
 - Requirements Elicitation:
 - Elicitation Techniques
 - Elicitation preparation
 - Conduct the elicitation
 - Analyze elicitation results
 - Difficulties of Requirements Elicitation
- To be continued next Lecture:
 - Other Phases of Requirements Engineering Process
 - Requirement Specifications

Requirements Definition and Specification

- ***Requirements definition***

- Abstract description of the **services** which the system should provide and the **constraints** under which the system must operate;
- Should be written in such a way that it is understandable by customers without knowledge of specialized notations

- ***Requirements specification*** – Structured document which sets out the system service in detail; Should be precise.

Requirements Definition



Functional Requirements

- **Functional requirements (WHAT)**
 - Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks.
 - So, it's important to make them clear both for the development team and the stakeholders.
 - Requirements should not contain design and implementation details (***HOW***)

Functional Requirements (contd.)

Functional Requirements should include the following:

- Details of operations conducted in every screen
- Data handling logic should be entered into the system
- It should have descriptions of system reports or other outputs
- Complete information about the workflows performed by the system
- It should clearly define who will be allowed to create/modify/delete the data in the system
- How the system will fulfill applicable regulatory and compliance needs should be captured in the functional document

Functional Requirements (contd.)

- Here's an example list of functional requirements for a user interacting with an Automated Teller Machine (ATM):
 - The system shall prevent further interaction if it's out of cash or is unable to communicate with the financial institution.
 - The system shall validate that the inserted card is valid for financial transactions on this ATM.
 - The system shall validate that the PIN number entered by the user is correct.
 - The system shall dispense the requested amount of money, if it is available, and debit the user's account by the same amount.
 - The system shall notify the user if the transaction could not be completed. In that case, no money shall be taken from the user's account.

Non Functional Requirements

- **Nonfunctional requirements:**
 - Describe the general characteristics of a system.
 - They are also known as *quality attributes*.
 - Failing to meet non-functional requirements can result in systems that fail to satisfy user needs.

Non Functional Requirements (contd.)

- **Non-Functional Requirements should include the following (but are not limited to) :**
 - **Performance.** How fast does the system return results?
 - **Scalability.** How much will this performance change with higher workloads?
 - **Portability.** Which hardware, operating systems, browsers, and their versions does the software run on?
 - **Compatibility.** Does it conflict with other applications and processes within these environments?
 - **Reliability,** How often does the system experience critical failures? and
 - **Availability.** How much time is it available to users against downtimes?
 - **Security.** How are the system and its data protected against attacks?
 - **Usability.** How easy is it for a customer to use the system?

Non Functional Requirements (contd.)

Here are examples list of non-functional requirements:

- If Employees try to update their salary information, such attempt should be reported to the security administrator.
- Every unsuccessful attempt by a user to access an item of data shall be recorded on an audit trail.
- A website should be capable enough to handle 20 million users without affecting its performance
- The software should be portable. So moving from one OS to other OS does not create any problem.
- Privacy of information, the export of restricted technologies, intellectual property rights, etc. should be audited.

A Decision Support System for Office Space Management

Case Study

Case Study Aim

- In any sizeable organization, managing the massive amount resources can become an expensive and a time consuming task.
- One of these many resources is the building(s) which must be adequate to both house and allow efficient operation of all entities, an entity being a member of staff or a facility.
- Allocating enough building space for every required entity in an organization can become problematic with the many restrictions and limited space.
- Many institutes have planning and layout guidelines which must be followed, plus each room may also have special requirements. Being able to plan a suitable layout can take considerable time and any future changes will consume yet more time and money.
- Making an optimized floor plan that wastes the minimal amount of space, can accommodate future changes in both staff and facilities and meets all restrictions is the problem.

Case Study Functional Requirements

1. Creation and editing of 2D floor plans for one building floor using straight lines only
2. Creation and editing of entities
3. Creation and editing of constraints
4. A user interface encompassing the floor plan, entities, and constraints
5. A print option or a exportable format suitable for printing which can produce a document suitable for use as a floor layout diagram
6. The ability to manually edit entities assigned rooms
7. The ability to manually assign room names
8. The ability to save and load Space Allocation data
9. Automatically edit rooms and assign entities based upon the constraints, entities, and the floor plan

Case Study Non Functional Requirements

1. Runs on Microsoft Windows XP
2. Allow quick, precise, understandable, and easily accessible editing of in-program data
3. Provide an accessible and usable User Interface to someone who is has basic computer knowledge
4. Has a data representation that allows AI algorithms to propose an optimized floor plan
5. To provide extensive documentation for both a user and a systems developer
6. Have a clean object oriented design allowing good maintainability

Requirements Engineering

- ***(1) Requirements elicitation***

process through which the customers, buyers, or users of a software system discover, reveal, articulate, and understand their requirements

- ***(2) Requirements analysis and negotiation***

process of reasoning about the requirements that have been elicited; examining requirements for **conflicts** or **inconsistencies**, **combining related requirements**, and identifying **missing** requirements

- ***(3) Requirements specification***

process of recording the requirements in one or more **forms**, including **natural** language and **formal, symbolic**, or **graphical** representations;

Also, the product – document produced by the process

Requirements Engineering

(1) Requirements Elicitation

Requirements Elicitation

- In requirements engineering, **requirements elicitation** is the practice of researching and discovering the requirements of a system from users, customers, and other stakeholders.
- The practice is also sometimes referred to as "**requirement gathering**".
- It's doubtful that any single elicitation technique will always work for you.
- It is generally accepted that an individual requirements elicitation technique or approach cannot possibly be suitable for all projects.
- So how do you decide which will give you the most bang for your buck (limited time), and why?
 - Your organization's makeup,
 - political climate,
 - the nature of your project,
 - your personal strengths and preferenceswill have a lot to do with which methods work best for you.

Requirement Elicitation Process

- **Step 1: Choose the optimal elicitation approach**
- Step 2: Elicitation preparation
- Step 3: Conduct the elicitation
- Step 4: Analyze elicitation results

Requirements Elicitation Techniques

- There are a myriad of requirements elicitation methods:



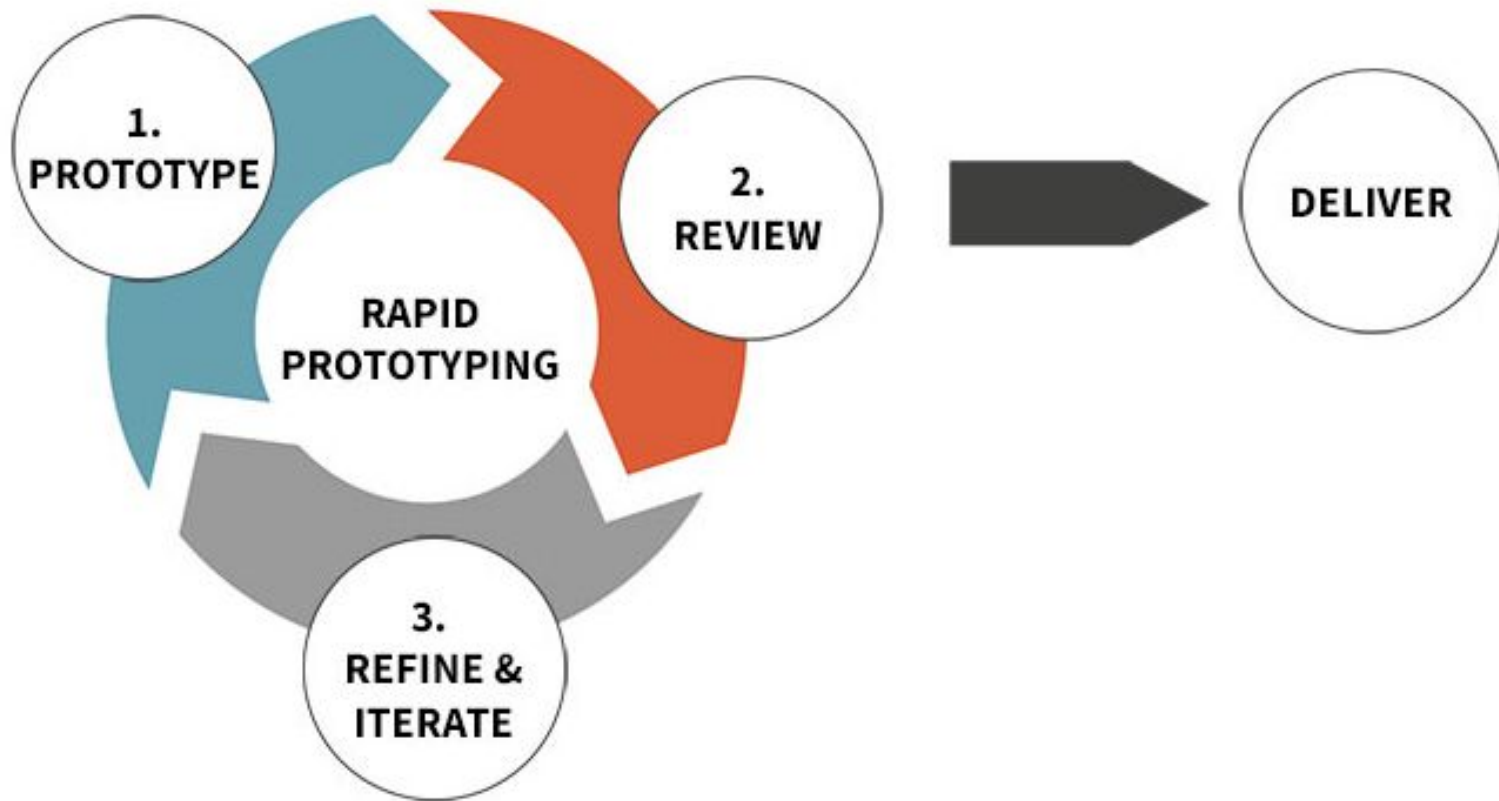
- but there are many more methods out there...

The Top 5 Elicitation Techniques



- **#1 Prototyping**
- **#2 Workshops**
- **#3 Interviews**
- **#4 Surveys**
- **#5 Observation**

Prototyping



- **Benefit:** You can make sure that what you're designing is really what people need while you still have time to change it.

Prototyping Example

MLCAT - Podcasting Lectures onto Mobies

select an icon
MLCAT

Create/Edit Session

Course Title:

Description:

Date:

Upload Files:

Upload files takes you to another screen which allows files to be added.

Requirements Workshops

- In a requirements workshop, you ask everyone to sit down and hammer out the requirements with you.
- “A requirements workshop:
 - is a highly productive focused event
 - attended by carefully selected key stakeholders and subject matter experts for a short,
 - intensive period (typically one or a few days).”
- **Benefit: You can get your basic requirements done in a hurry. Also, everyone you invite can become invested in the project.**

Requirements Workshops



Requirements Workshops

For your workshop to be successful, you will need to:

- **Select the right participants:**
 - Think about this carefully before inviting your group. (Do not involve too many participants ☹ slow down the workshop process)
 - Conversely, collecting input from too few participants can lead to overlooking requirements.”
- **Get everyone on the same page regarding the purpose of the workshop ahead of time** (defining scope, unearthing business requirements, etc.)
- **Conduct the workshop like an interview**, with open-ended questions presented to the room.
- **Document everything.** Get a recorder or get someone besides you (or whomever will be busy facilitating) to write everything down.

Interviews

- Interviews help you dig through your users' knowledge base, so you can understand what they understand and think.
- One writer notes, "Interviews provide an efficient way to collect large amounts of in-depth data quickly,"
- **Benefit: By exploring someone's knowledge and needs in-depth, one-on-one, you ensure you understand the real, not just the perceived, need.**

-

Interviews



Interviews (contd.)

For your interview to be effective, you must decide how structured or unstructured you want your interview to be.

- **Structured Interviews:**

- These are interviews that strictly adhere to the use of an interview protocol to guide the researcher.
- It is a more rigid interview style, in that only the questions on the interview protocol are asked.
- As a result, there are not a lot of opportunities to probe and further explore topics that participants bring up when answering the interview questions.
- **This method can be advantageous when researchers have a comprehensive list of interview questions**

Interviews (contd.)

- **Semi structured interviews:**

- These are interviews that use an interview protocol to help guide the researcher through the interview process.
- It does maintain some structure (hence the name semi structured),
- But it also provides the researcher with the ability to probe the participant for additional details.
- If you decide to choose this interview method, understand that it offers a great deal of flexibility for you as a researcher.
- You do not have to worry about needing to conduct several rounds of interviews **because your interview protocol will keep you focused on gathering all the information** that you need to answer your research question.

Interviews (contd.)

- **Unstructured interviews:**

- These are interviews that take place with few, if any, interview questions.
- They often progress in the manner a normal conversation would, however it concerns the research topic under review.
- It is a relatively formless interview style that researchers use to establish rapport and comfort with the participant, and is extremely helpful when researchers are discussing sensitive topics.
- If you select this interview style, just keep in mind that you **may have to conduct several rounds of interviews** with your participants in order to gather all the information you need.
- Since you do not use a standard interview protocol, sometimes participant's narratives maneuver the conversation away from other aspects of the research topic you want to explore;

Surveys / Questionnaires

- Questionnaires take into consideration data to be evoked from **numerous individuals**.
- The ideal approach to this technique is by making a basic Google Form and offering it to the correct individuals, and whenever required, determining a due date.
- You have to know what you are attempting to accomplish precisely with the study, and the questions must not to be uncertain.
- Misunderstanding of inquiries can prompt useless and pointless answers.

The format for Questionnaires

- **Fixed Format:**

- Fixed format surveys consist of questions that need a variety of predefined responses from people.
- Respondents have to choose an answer from a series of answers provided.
- A reply from this format of the questionnaire is a lot simpler to interpret.
- In any case, then again, it is increasingly latent; respondents can't give their answers or opinion other than presented in the survey.

- **Free Format:**

- Free format surveys will enable users to answer openly for each inquiry.
- A question is proposed, and the respondent enters the appropriate response in the space given after the query.

Fixed Format

NETFLIX

15%

How would you describe your satisfaction with the movies and TV shows on Netflix?

Select one response per row

	Not at all Satisfied 1	2	3	4	5	6	Extremely Satisfied 7	Not Applicable
Selection of Netflix Original movies (produced by Netflix)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Selection of Netflix Original TV shows (produced by Netflix)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Selection of movies and TV shows for children available	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Selection of locally produced movies and TV shows	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Selection of movies available	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Selection of TV shows available	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Continue »

Free Format

hotjar

SURVEY

1. *What was your biggest concern or fear before buying our product?*

2. *What should we have done to improve your experience?*

3. *How likely are you to recommend us to a family or friend?*

0

1

2

3

4

5

6

7

8

9

10

Observation

- Observation is primarily useful for capturing what's already in existence and enables several other types of requirements tools.
- The analyst can document what he/she observes through numerous types of diagramming and business process models.
- “The effectiveness of observation . . . can vary as users have a tendency to adjust the way they perform tasks when knowingly being watched.”
- Make sure the people you observe know that you are not there to judge what they do, but to make their work easier in the long run.
- Ask them what they like and don't like about it, and about any workarounds they've created on their own.
- **Benefit: You can figure out exactly where users are at the start of your project, and you can use your strengths to document it.**

Observation



Summary of Elicitation Techniques

Technique	Good for	Kind of data	Plus	Minus
Questionnaires	Answering specific questions	Quantitative and qualitative data	Can reach many people with low resource	The design is crucial. Response rate may be low. Responses may not be what you want
Interviews	Exploring issues	Some quantitative but mostly qualitative data	Interviewer can guide interviewee. Encourages contact between developers and users	Time consuming. Artificial environment may intimidate interviewee
Focus groups and workshops	Collecting multiple viewpoints	Some quantitative but mostly qualitative data	Highlights areas of consensus and conflict. Encourages contact between developers and users	Possibility of dominant characters
Naturalistic observation	Understanding context of user activity	Qualitative	Observing actual work gives insight that other techniques cannot give	Very time consuming. Huge amounts of data

Requirement Elicitation Process

- Step 1: Choose the optimal elicitation approach
- **Step 2: Elicitation preparation**
- Step 3: Conduct the elicitation
- Step 4: Analyze elicitation results

Elicitation Preparation

You should have established:

- An objective for elicitation activities
- The specific participants
- The resources or other supporting materials needed during the elicitation activities
- A predetermined set of questions and how stakeholder responses are to be recorded
- An agenda with definitive start and end times

Requirement Elicitation Process

- Step 1: Choose the optimal elicitation approach
- Step 2: Elicitation preparation
- **Step 3: Conduct the elicitation**
- Step 4: Analyze elicitation results

Conduct the Elicitation

Elicitation activities have four stages:

1.Introduction:

- You set the stage by introducing the purpose and goals of the elicitation activity: establish the tone of the interaction, discussion of activities, timing, etc.

2.Body:

- Depending on which tool or technique you chose, the body will differ.
- For example, if you're interviewing a stakeholder, you'll likely launch into your questions or if you're conducting a workshop, you'll transition the group into the scheduled activity.
- Questions may arise organically throughout the interaction.

Conduct the Elicitation (contd.)

3. Close:

- The transition to elicitation activity closure can be a smooth one if you've kept an eye on the time in relation to the flow of the activity.
- Ask closing questions, e.g., "Do you have any questions", "Is there any information you think is important about the product that we didn't discuss?", etc.
- Definitely ask if the stakeholder is open to follow up questions, new ones may surface as you analyze the elicitation results.

4. Follow-up:

- **This step isn't always needed**, but can prove to be extremely useful.
- Follow-ups provide you with the opportunity to ask clarifying questions and ensure that the information you did record is accurate.

Conduct the elicitation (contd.)

- Once all of your elicitation activities are completed, you've gathered all of the data/information into some sort of repository
 - this can be digital
 - handwritten notes,
 - drawings, etc.
- Your next step is to analyze everything you've collected.

Requirement Elicitation Process

- Step 1: Choose the optimal elicitation approach
- Step 2: Elicitation preparation
- Step 3: Conduct the elicitation
- **Step 4: Analyze elicitation results**

Analyze Elicitation Results

- Analysis can be as straightforward as reading through your notes and other documents then summarizing them.
- Identifying keywords and ideas that were derived from stakeholder elicitation activities.
- There are tools to support the analysis of requirements such as : QVscribe , NVivo.

Difficulties of Requirements Elicitation

(1) Articulation problems

- Users: may **not** be **aware** of their needs, **unable** to articulate them appropriately or **afraid** to articulate them
- Developers: may not really be **listening** to the users; may fail to **understand**, appreciate, or relate to the users; tend to overrule or **dominate** users

(2) Communication barriers:

- Users and developers have different professional vocabularies

(3) Knowledge and cognitive limitations:

- Requirements elicitor must have adequate domain knowledge
- People tend to state the problem in terms of the **favored solution**

Difficulties of Requirements Elicitation (contd.)

(4) Human behavior issues:

- Expectation or fear that installation of software will necessitate all kinds of changes in behavior, including the potential **loss of jobs**.

(5) Technical issues:

- **Requirements change** over time
- Software and hardware **technologies** are **changing** rapidly

To be Continued...