

Requirements Elicitation

Requirements Elicitation

- It is a process of gathering requirements from the client; also called as Requirements gathering
- Requirement elicitation can be done by communicating with stakeholders directly or by doing some research, experiments.

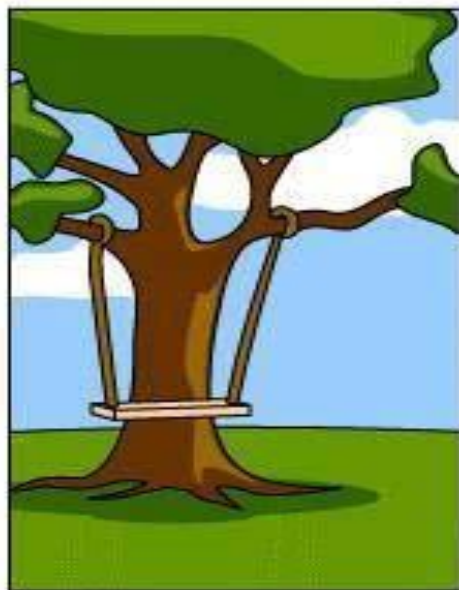
Stakeholders are the one who are showing interest in the project/ benefited from the project.

Requirements

- A requirement is a statement that identifies a capability/functionality of a system in order for it to have value and utility to a customer or user.
- Requirements are important because they provide the basis for all of the development work that follows.
- Once the requirements are set, developers initiate the other technical work: system design, development, testing, implementation, and operation.



How the customer explained it



How the Project Leader understood it



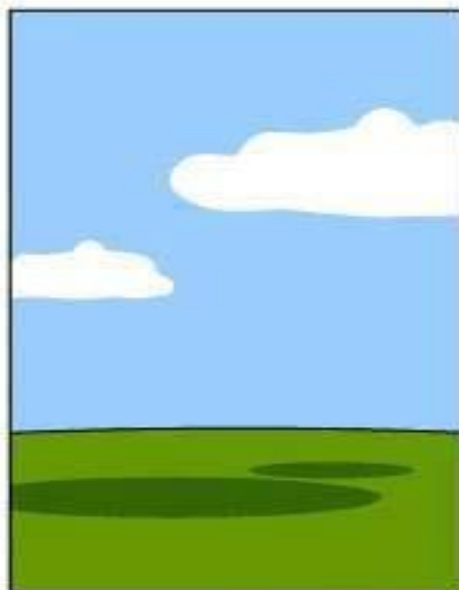
How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



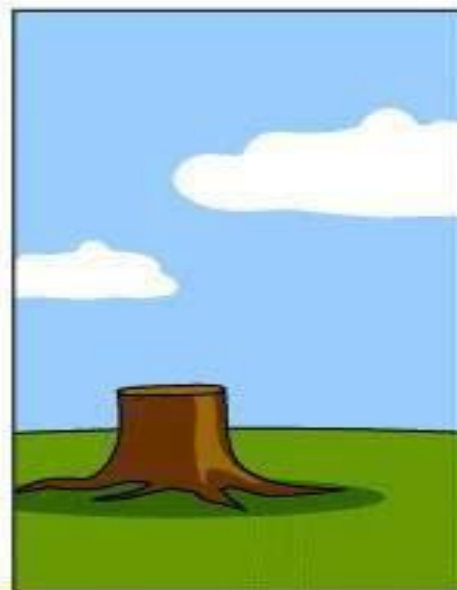
How the project was documented



What operations installed



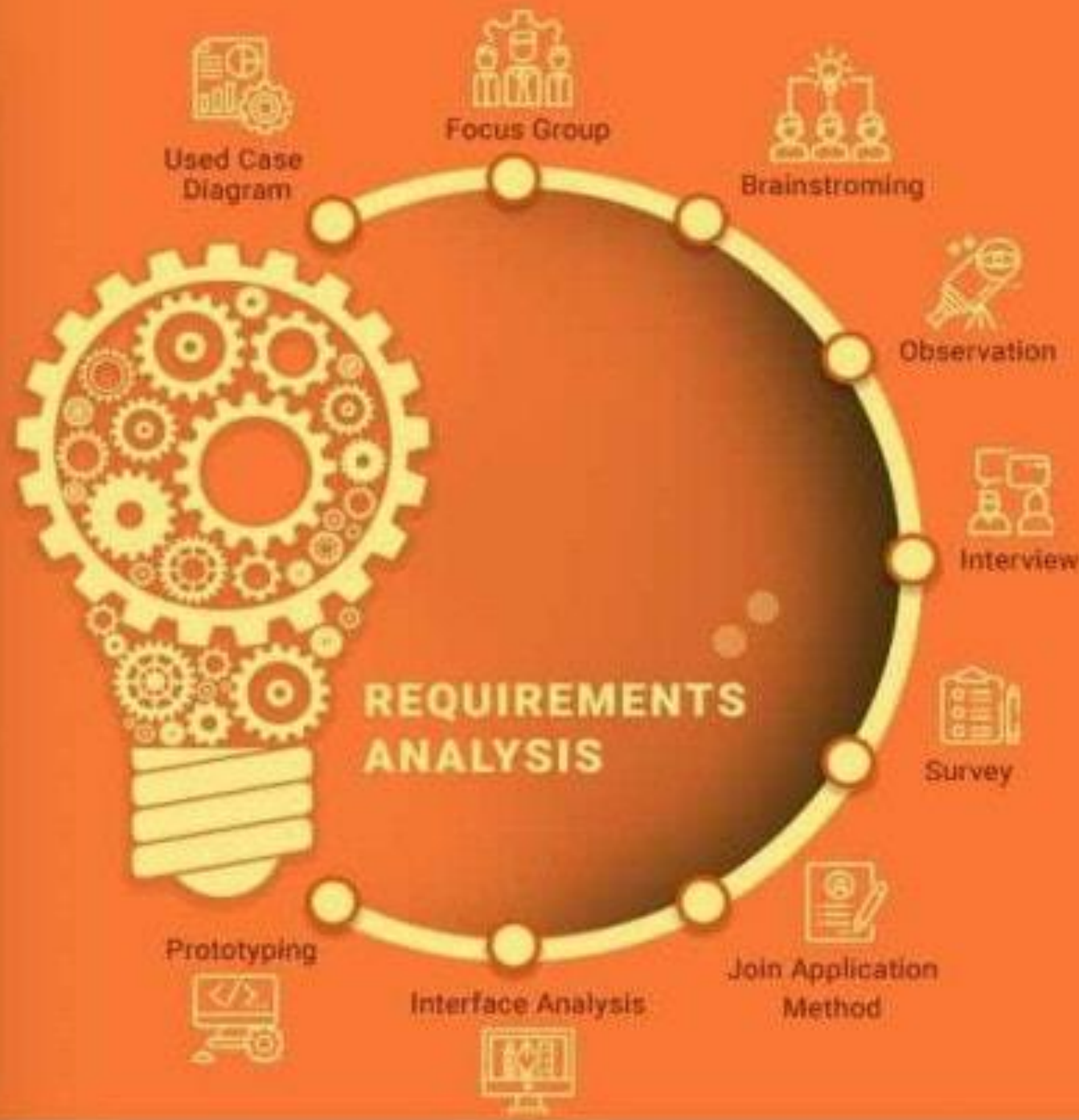
How the customer was billed



How it was supported

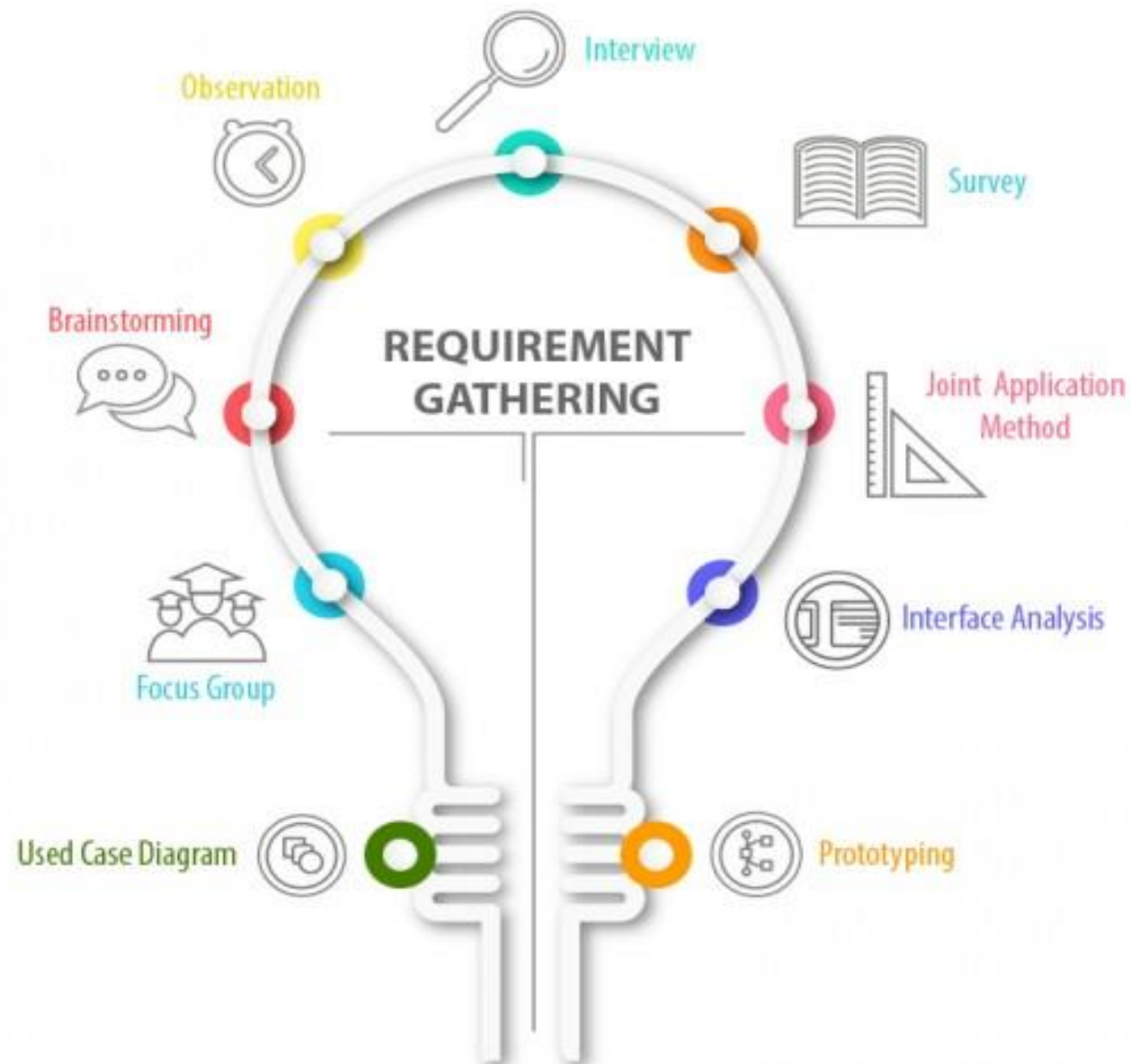


What the customer really needed



REQUIREMENTS ANALYSIS

Understanding the basics- Definition, Process, & Requirement Analysis Techniques



#1 Interview

- Interview turns out to be one of the most effective techniques for requirement gathering.
- In this method, the business analyst talks to the user and clients who are unable to give out detailed information as they are not aware of the system development and related functionalities.
- It is the responsibility of the business analyst to extract relevant information from them which can be achieved by interview.

#1 Interview

- One to one interview is the most commonly used technique
 - Structured interview
 - predefined set of questions
 - Unstructured interview
 - not having any particular format or any specific questions
 - For an effective interview, consider the **5 Why technique**.
<https://kanbanize.com/lean-management/improvement/5-whys-analysis-tool>
 - When you get an answer to all Whys then you are done with your interview process

5 Whys method

- The 5 Whys method is part of the Toyota Production System. Developed by Sakichi Toyoda, a Japanese inventor and industrialist, the technique became an integral part of the Lean philosophy.
- “The basis of Toyota’s scientific approach is to ask why five times whenever we find a problem. By repeating why five times, the nature of the problem as well as its solution becomes clear.”
- 5 Whys analysis aims to inspect a certain problem in depth until it shows you the real cause.
- Keep in mind that “5” is just a number. Ask “Why” as many times as you need to complete the process and take appropriate actions.

Example of applying the 5 Whys

1. Why did the project fail?

Answer: The project failed due to inadequate planning and lack of resources.

2. Why was there inadequate planning?

Answer: The project team did not have a clear understanding of the scope of the project and did not allocate enough time and resources to complete it.

3. Why did the project team not have a clear understanding of the scope of the project?

Answer: The project team did not receive adequate training and guidance from senior management.

4. Why did the project team not receive adequate training and guidance from senior management?

Answer: Senior management did not provide the necessary support and resources to ensure the project team had the necessary skills and knowledge to complete the project.

5. Why did senior management not provide the necessary support and resources?

Answer: Senior management did not prioritize the project and did not allocate enough resources to ensure its success.

- **Example: A software application experiences frequent crashes.**
 - **Why does the application crash?** There are several bugs in the code.
 - **Why are there bugs in the code?** The testing process was inadequate.
 - **Why was the testing process inadequate?** The test cases were not comprehensive.
 - **Why were the test cases not comprehensive?** The testing team was understaffed.
 - **Why was the testing team understaffed?** The importance of thorough testing was not prioritized.

- Example: A project is continually behind schedule, which impacts profitability.
 - Why is the project behind schedule? The team is constantly missing deadlines.
 - Why is the team missing deadlines? They are struggling with workload prioritization.
 - Why are they struggling with workload prioritization? There is a lack of clear project milestones.
 - Why are there no clear project milestones? The project manager lacks experience in defining them.
 - Why does the project manager lack experience? They were promoted without proper training.

Decision: Implement project management training for the project manager and establish clear milestones for the team to follow.

Why?

There are not enough customer service representatives

Why?

Budget constraints limit the number of staff members

Why?

Revenue has decreased, affecting the budget

Root cause: The organization needs to find ways to increase revenue to hire more staff and reduce wait times for customers.

Problem: Customer complaints about long wait times

Problem: Project deadlines are consistently missed

Why?

Tasks take longer than expected to complete

Why?

Team members are unsure of their roles and responsibilities

Why?

There is a lack of clear communication and project management

Root cause: Improve project management and communication to ensure that deadlines are met.



brain STORMING

#Brainstorming

- Requirements gathering is critical to any project, ensuring that all stakeholders' needs and expectations are well understood.
- One highly effective technique in this process is brainstorming.

#Brainstorming

Brainstorming is a creative and collaborative approach that encourages participants to generate a wide range of ideas, solutions, and requirements.

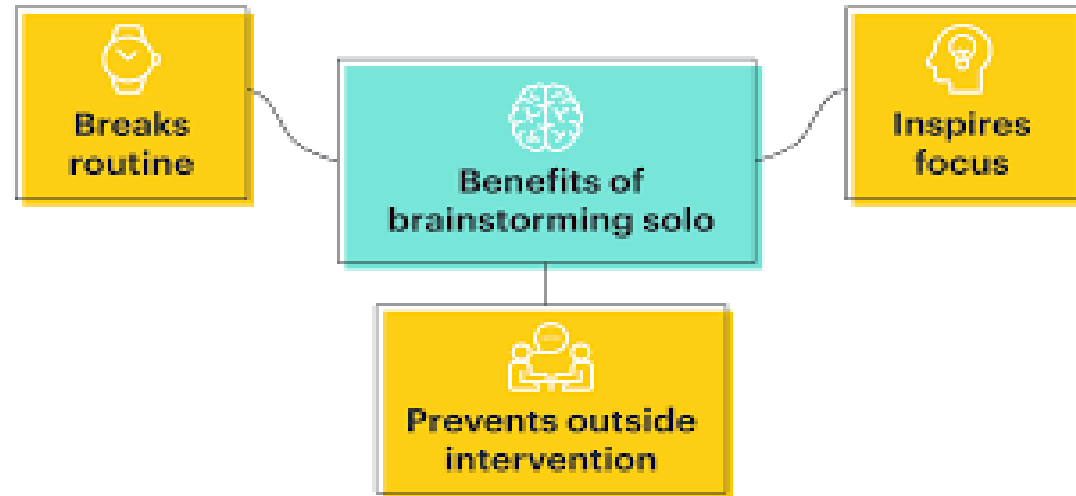


#Brainstorming

- Brainstorming encourages participants to think freely, without judgment or criticism, and explore various possibilities. This technique fosters creativity and can uncover valuable insights that might not emerge through other methods.
- As a Business Analyst, the success of a brainstorming session depends on how well it's organized, facilitated, and documented.

#Brainstorming - Steps

1. Define Clear Objectives
2. Identify Key Participants
3. Create an Agenda
4. Choose the Right Location
5. Prepare Materials
6. Set Ground Rules
7. Use Facilitation Techniques
8. Encourage Diverse Input
9. Record and Cluster Ideas
10. Prioritize and Discuss
11. Document and Share
12. Create Actionable Plans
13. Follow-Up



Solo BrainStorming

#3 Document Analysis/Review

- This technique is used to gather business information by reviewing/examining the available materials that describe the business.
- This analysis is helpful to validate the implementation of current solutions and is also helpful in understanding the business needs.
- This technique is important in identifying the gaps in the system i.e. to compare the AS-IS process with the TO-BE process.

AS-IS process : TO-BE process

- As-is and To-be process flows help you visualize how a process currently “as is”, giving you the opportunity to redesign the process to what you want it “to be”.
- The **As-is** process flow gives you a detailed visualization of a process as it exists today; take time to research all the steps currently in the process, not worrying about improvements at this point.
- By taking the time to understand the as-is process flow; can identify areas where improvements can be made, which you will highlight in the To-be process flow.

AS IS TO BE TEMPLATE

AS - IS → *Where are you now*

As - Is

We are not currently known for innovation, however our software does contain a unique feature

As - Is

The stand-alone version of software

TO- BE → *What are you focused on?*

To - be

To be recognized as one of the most innovation platforms in the industry

To - be

To be able to be accessed by everyone

GAP → *Difference between as is and to be*

Gap

Limited resources

Gap

Absence of Innovation checkpoint

#4 Observation

- Observation is an elicitation technique that helps in collecting information by observing process flows and work environments of stakeholders.
- It helps business analysts in situations where users are unable to explain requirements clearly.
- Observation provides practical insights into the actual real-world activities and even enables the identification of informally performed tasks. It also facilitates the identification of areas that can potentially be improved.

#5 Focus Group

- It is a way to collect information, opinions, and feedback from a group of people with diverse backgrounds and perspectives.
- Focus groups are used to understand the needs and expectations of stakeholders, end users, or subject matter experts.
- The ideal size of a focus group is generally between 6 and 8 participants. Because it's small enough to ensure that everyone has a chance to speak and share their views, but large enough to provide a variety of perspectives.
 - The representative here has a broad idea pertaining to the needs of the users and clients. Under focus group, the idea is

#6 Prototyping

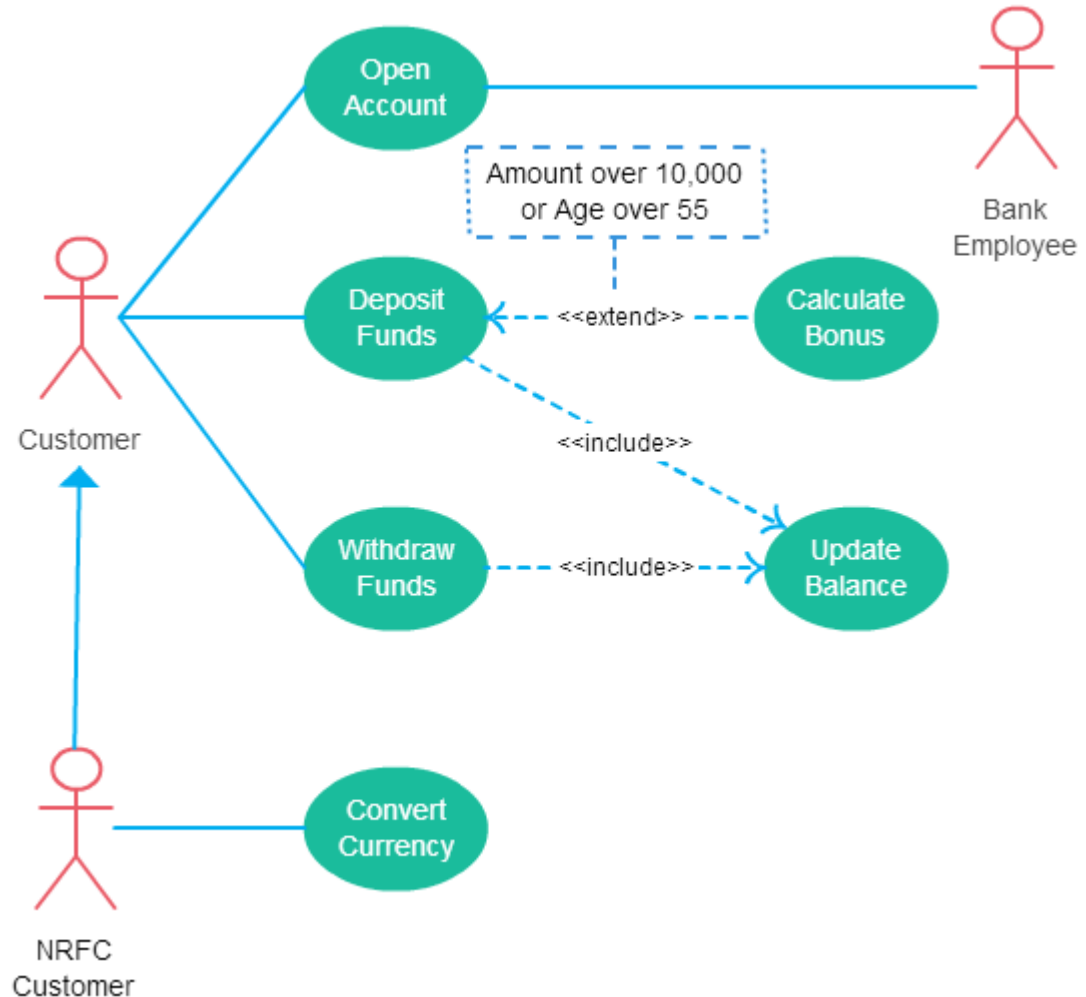
- This is a technique of building a model of software which helps in uncovering and capturing software requirements from client.
- The output can be broad mockups or sketch formats of software.

#7 Survey

- It is another effective method to collect information and requirements within a short frame of time.
- Under this technique, it is advisable to first ascertain the goal of the survey and thereafter draft the questionnaire.
- Once your questions list is ready, it should be delivered to the user as well as the stakeholder for answers.
- A responsible business analyst would study the answers and then document them.

#9 Use case diagram

- Use case diagram is a technique that shows how people interact with software. It shows what a system does.
- Even, any of the model describes the current system will be useful one.



#10 Joint Application Method

- JAD is used as a technique for developing business system requirements
- The purpose of JAD is to bring together all stakeholders in a structured workshop setup; to extract judgemental based system requirements.
- These stakeholders attend the workshops till the time the desired goal is accomplished.

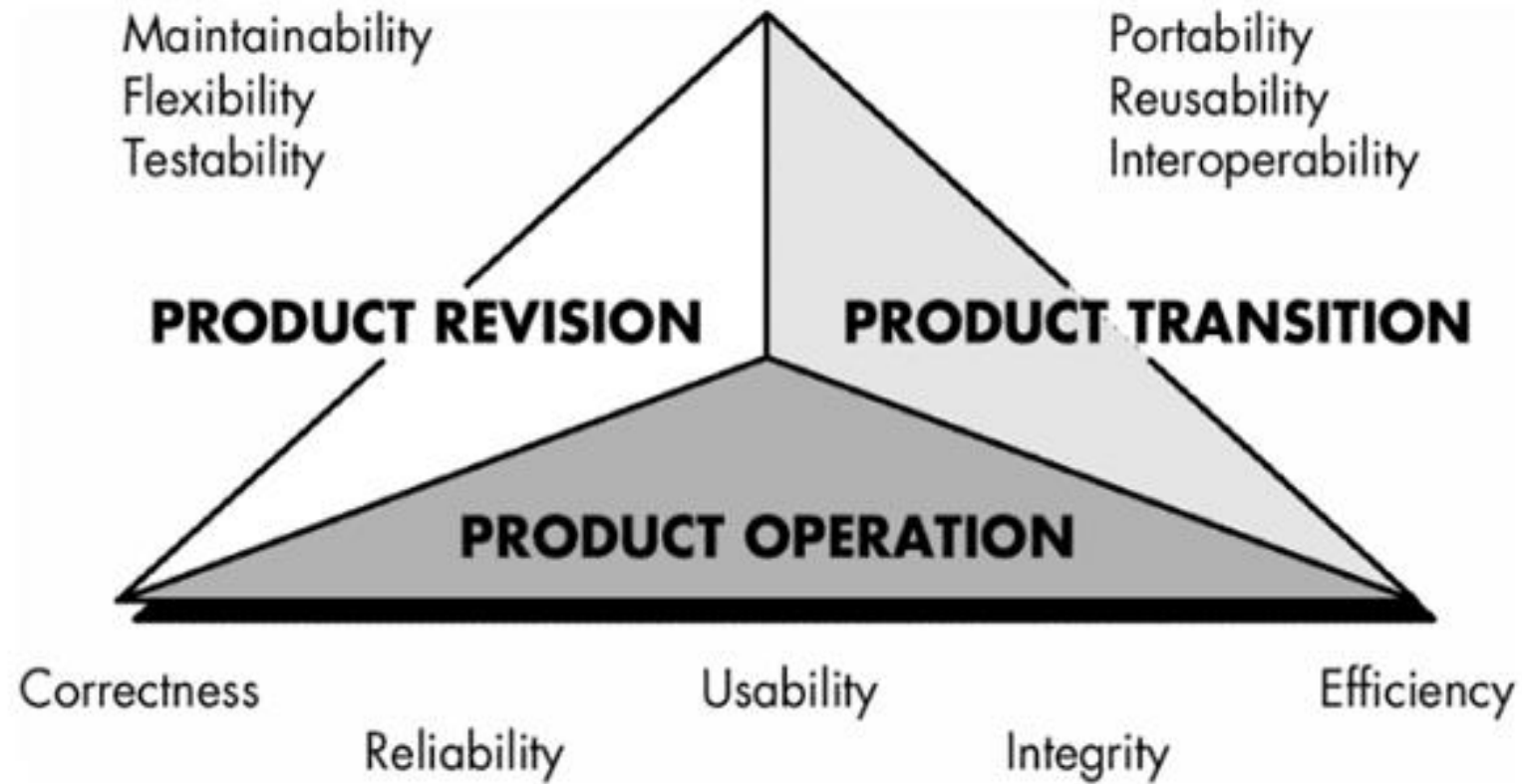
Requirements Elicitation Techniques

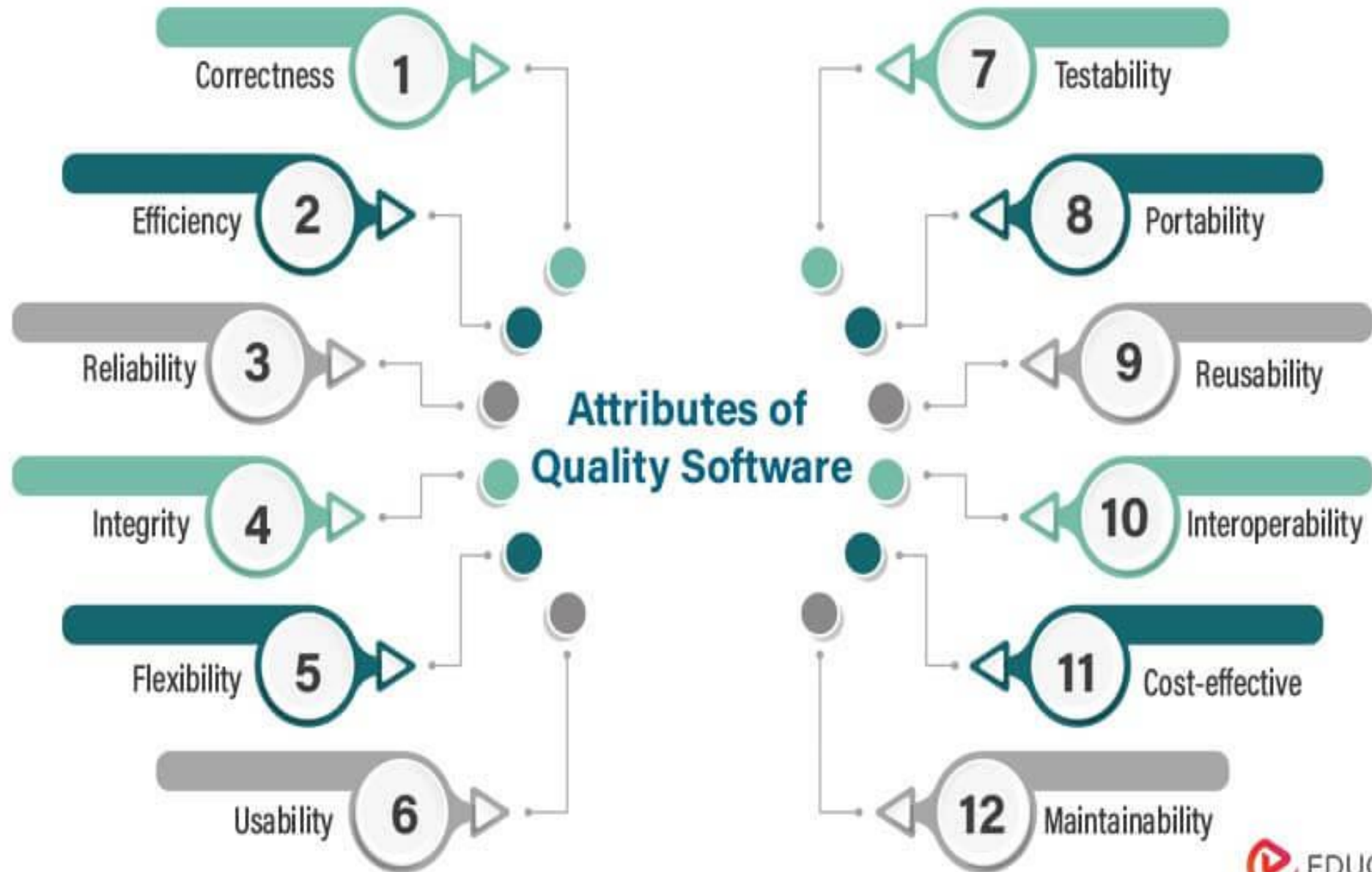
- To select an appropriate technique for requirements gathering, there are a few factors that need to be considered:
 - Availability and location of the stakeholders
 - Client and development team's knowledge on the problem
 - Client and development team's knowledge of the development process and methods to resolve the same.
 - Documents related to the existing system

Types of Requirements

- Functional Requirements
 - Functional means providing particular service to the user. For example, in context to banking application the functional requirement will be when customer selects “View Balance” they must be able to look at their latest account balance.
- Non-Functional Requirements
 - Software requirement can also be a non-functional, it can be a performance requirement. For example, a non-functional requirement is where every page of the system should be visible to the users within 5 seconds.

Quality Attribute Triangle





Product Operation

- **Correctness** – The extent to which a software meets its requirements specification.
- **Efficiency** – The amount of hardware resources and code the software, needs to perform a function.
- **Integrity** – The extent to which the software can control an unauthorized person from the accessing the data or software.
- **Reliability** – The extent to which a software performs its intended functions without failure.
- **Usability** – The extent of effort required to learn, operate and understand the functions of the software.

Product Revision

- Maintainability – The effort required to detect and correct an error during maintenance phase.
- Flexibility – The effort needed to improve an operational software program.
- Testability – The effort required to verify a software to ensure that it meets the specified requirements.

Product Transition

- **Portability** – The effort required to transfer a program from one platform to another.
- **Re-usability** – The extent to which the program's code can be reused in other applications.
- **Interoperability** – The effort required to integrate two systems with one another.

Functional vs. Non-Functional

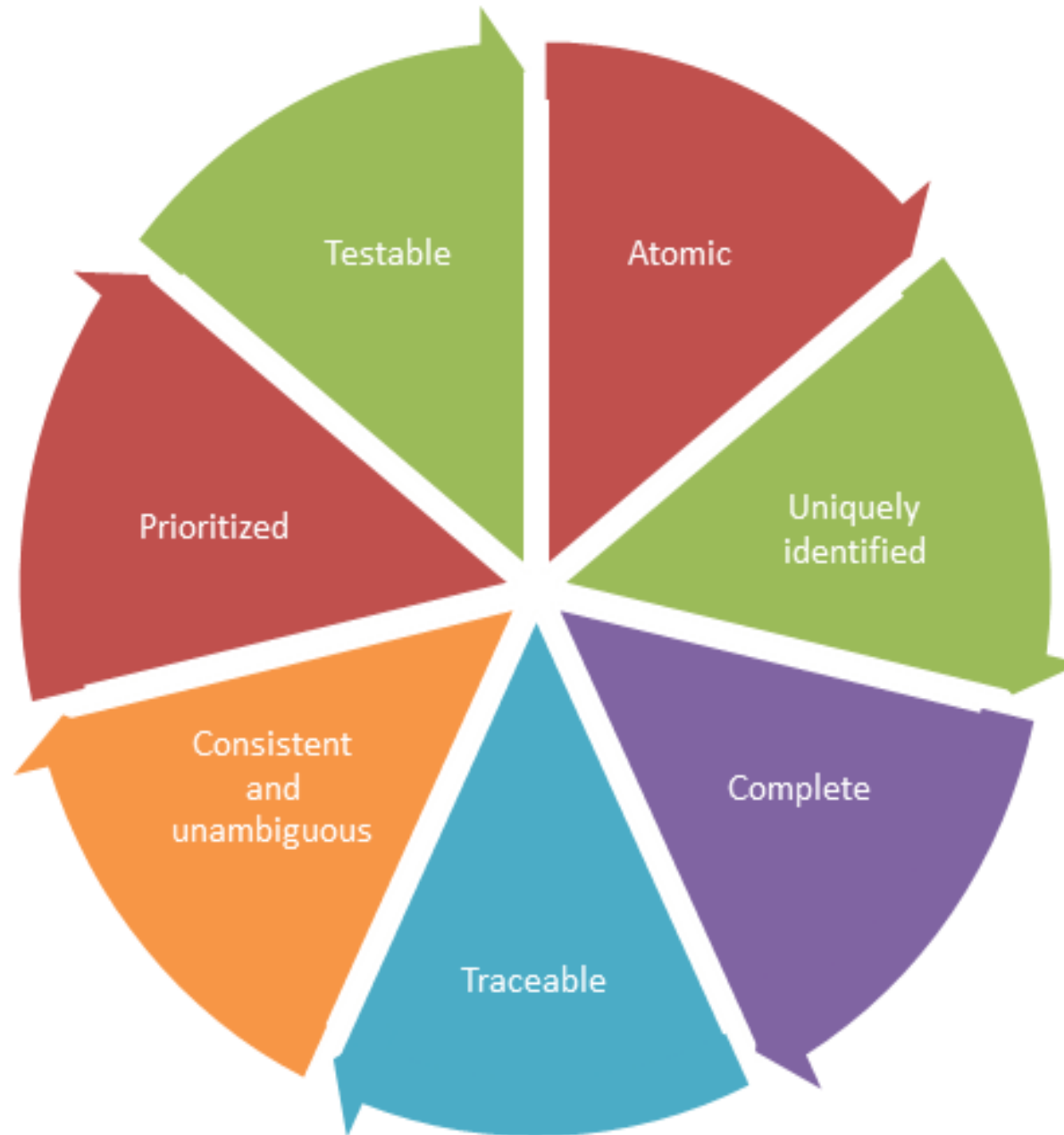
Functional	Non-Functional
Defines a system or its component	Defines the performance attribute of a software system.
Focus on user requirement	Concentrates on the user's expectation.
Features of software product	Properties of software product
Helps to verify the functionality of the software.	Helps to verify the performance of the software.
Describes what the product does	Describes how the product works
Easy to implement	Hard to implement

A sample requirement

- If sales for current month are below target sales, then report is to be printed unless difference between target sales and actual sales is less than half of difference between target sales and actual sales in previous month, or if difference between target sales and actual sales for the current month is less than 5%.
- Problems in this requirement are
 - Difficult to read
 - Ambiguity: sales and actual sales, 5% of what?
 - Incomplete: what if sales are above target sales?

There is a need for the quality in requirements definition

How to Analyze Quality of the Requirements?



How to Analyze Quality of the Requirements

- Consider example of an educational software system where a student can register for different courses.

Unambiguous

- REQ: The system shall be implemented using ASP.
 - REQ: The system shall be implemented using Active Server Pages.
- REQ: On the books screen, the user can only view one book.
 - REQ: On the books screen, the system shall display only one book.

Testable (Verifiable)

- REQ: The user shall be able to search for books based on author's name, title, etc.
- REQ: The user shall be able to search for books based on author's name or title.

Concise

- REQ: Sometimes the user can search for books using author's name, but sometimes he should be able to search using the book title. Yet, other times, the user can enter both.
- REQ: The user shall be able to search for books based on author's name or title.

Correct

- **REQ: Based on bank regulations, currency amounts shall be calculated and stored with accuracy of two decimal places.**

Understandable

- Requirements should be
 - grammatically correct
 - Written in a consistent style e.g. the word “shall” should be used instead of “will”, “must”, “can”, or “may”
- REQ: The system shall remember customer data.
- REQ: The system shall display order details.

Feasible (Realistic)

- REQ: The system shall be able to understand commands given in Arabic language.

Independent

- **REQ:** The administrator shall be able to enter the list of best selling books.
- **REQ:** The system shall allow the user to view it.
- **REQ:** He shall be able to enter books related to a given book.



Atomic

- REQ: The system shall provide the ability to order books, browse the best-selling books, search for books, and view book information.

Necessary

- A requirement is unnecessary if
 - It is not needed by any stakeholder
 - Or removing it will not affect the system
- **REQ:** All requirements shall be implemented and tested.



Implementation-Free (Abstract)

- Requirements should not contain unnecessary design and implementation information.
- **REQ:** Customer information shall be stored in a text file.



Example

- Let understand this with an example, there are three columns in the table shown in the next slide;
 - The first column indicates- “requirement quality”
 - The second column indicates- “bad requirement with some problem”
 - The third column is same as second column but – “converted into a good requirement”.

Requirements Analysis

Requirement Quality	Example of bad requirement	Example of good requirement
Atomic	Students will be able to enroll to undergraduate and post graduate courses	Students will be able to enroll to undergraduate courses Students will be able to enroll to post-graduate courses
Uniquely identified	<ol style="list-style-type: none">1. Students will be able to enroll to undergraduate courses2. Students will be able to enroll to post-graduate courses	1.Course Enrolment 1.1 Students will be able to enroll to undergraduate courses 1.2 Students will be able to enroll to post-graduate courses
Complete	A professor user will log into the system by providing his username, password, and other relevant information	A professor user will log into the system by providing his username, password and department code

Requirements Analysis

Requirement Quality	Example of bad requirement	Example of good requirement
Consistent and unambiguous	A student will have either undergraduate courses or post-graduate courses but not both. Some courses will be open to both under-graduate and post-graduate	A student will have either undergraduate or post graduates but not both
Traceable	Maintain student information-mapped to BRD req.ID? [Business Requirements Document - BRD]	Maintain student information-Mapped to BRD req ID 4.1
Prioritized	Registered student-Priority 1 Maintain User Information-Priority 1 Enroll courses-Priority 1 View Report Card-Priority 1	Register Student-Priority 1 Maintain User Information-Priority 2 Enroll courses-Priority 1 View Report Card-Priority 3
Testable	Each page of the system will load in an acceptable time-frame	Register student and enroll courses pages of the system will load within 5 seconds

Questions?