REQUIREMENT ANALYSIS

Assoc. Prof. Dr. Mehmet Akif Çifçi

What is Requirements Analysis?

- Requirements analysis is the process of identifying, defining and documenting the requirements of a software system. The goal of requirements analysis is to identify user needs and translate them into specific, measurable and achievable requirements that the software development team can use to design and develop the system. This process is also known as requirements engineering and is a critical step in the software development life cycle (SDLC) as it helps ensure that the final product meets the needs of stakeholders and users.

What is Requirements Analysis?

- Requirements analysis includes various activities such as gathering requirements from stakeholders, documenting and organizing requirements, validating requirements and creating a requirements specification document. This document serves as a blueprint for the software development team and is used to guide the design, development, testing and deployment of the software system.

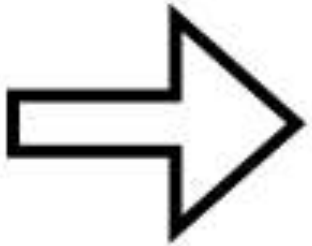
What is Requirements Analysis?

- It is important to remember that requirements analysis is an iterative process that continues throughout the SDLC and changes may occur as the project progresses. This means that the requirements analysis process needs to be flexible enough to adapt as changes and updates occur.

Purpose of Requirements Analysis?

- To clarify the functions and precise requirements of the system and consequently document them in a specific format.
- It allows developers to understand how customers want the system to work.
- Requirements tell us what features and functionality the final system will have.
- The requirements tell the testing team what they need to show to convince the user that the system presented is the desired system.

İhtiyaç



Gereksinim

A requirements analysis does the following:

- Describes the necessary features and overall vision of a new product.
- Clarifies stakeholder expectations for the relevant product.
- Avoids conflict and communication gaps during development and testing.
- It ensures that the final product conforms to the requirements, i.e. avoids going out of scope.

Requirements Analysis Steps

- Collect to requirements.
- Analyze the requirements.
- Improve the quality of requirements.
- Model the requirements.
- Document and review requirements.

1. Collect Requirements

- To gather requirements, you need to contact users. During the requirements gathering phase, you can interview users, observe your workplace and organize focus groups. Based on all the data you collect, you can create the requirements document.

2. Analyze Requirements

- What you need to do at this stage is to evaluate the system using the requirements document. You can identify whether the requirements listed are contradictory, incomplete or ambiguous and investigate what you can do to resolve these issues. The purpose of this phase is to decompose, analyze and elaborate the system into its requirements.

3. Improve the Quality of Requirements

You can improve the quality of requirements using the criteria we have listed:

- Visualization: Use tools such as visualization and simulation to better understand the final system being developed.
- Documentation: Document the relationships between requirements and assumptions.
- Use of Templates: Use consistent templates to memorize requirements in an easy-to-understand way.

4. Model Requirements

- You can model requirements using UML diagrams, flowcharts, graphs or models to make sure the system fits the needs of the business.

5. Document and Review Requirements

- To make the requirements easy for both developers and users to understand, you should save them in a document and check all requirements afterwards.

How are Requirements Classified?

- Requirements are divided into two categories: functional and non-functional requirements. Functional requirements are all the features that the people who will use the system expect from the system. Non-functional requirements are general software features that are not related to the functionality of the application.

1. Functional Requirements

- Defines what functionality the system or software should provide.
- It determines users' expectations, business processes and system behaviors.
- Example: Functional requirements such as user login, database management, report generation.

2. No Functional Requirements

- Non-functional requirements are processes that the user needs to solve independently of the problem.
- It includes performance metrics such as response times, height, processing speed.
- Example: Page load time, type of computer to be used, software development environment, database management system to be used.

What are the Types of Requirements?

- Physical Environment
- Interface
- User and Human factor
- Functionality
- Documentation

- Data
- Sources
- Security
- Quality Assurance


Arayüzler
Fiziksel Ortam
Kullanıcı
Kalite
GEREKSİNİMLER
İşlevsellik
Güvenlik
Belgelendirme
Kaynaklar
Veri

Physical Environment

- What are the devices on which the functions will be developed and operated?
- Will the system be in one place? Are there physically separate locations?
- Are there environmental constraints such as temperature, humidity or magnetic interference?

Interface

- Are the inputs coming from one or multiple systems?
- Do the outputs go to one or multiple systems?
- Is there a way how to format the data?
- Is there a recommended environment in which the data will be used?

User and Human Factor

- Who will use the system?
- Will there be different types of users?
- What is the skill level of each type of user?
- What kind of training is required for each type of user?
- To what extent is it difficult for a user to use the system maliciously?

Functionality

- What will the system do?
- When will the system make it happen?
- How and when can the system be changed and/or strengthened?
- Are there any limiting factors on speed of operation, response time or output?

Documentation

- How much documentation is required?
- Which user group is the documentation targeting?
- Is documentation of the maintenance process of the system to be developed required?

Data

- What should be the format of the data for both input and output?
- How often will this data be received or sent?
- What should be the accuracy of this data?
- What precision should the results of the calculations be?
- How much data will flow through the system?
- How long will the data be stored and processed?

Sources

- How much material, personnel and other resources are needed to install, use and maintain the system?
- What skills should developers have?
- How much physical space will the system take up?
- What are the constraints for power, heating and cooling?
- Is there a recommended timeline for development?

Security

- Should access to the system or information be controlled?
- How will one user's data be separated from another?
- How will user programs be kept separate from other programs and the operating system?
- How often will the system be backed up?
- Will backup copies be stored elsewhere?
- What measures will be taken against fire and theft?
- Is there internet access? Is security used?

Quality Assurance

- What are the requirements for reliability?
- How should the features of the system be transferred to the constructions?
- What is the predicted time interval between system crashes?
- What are the efficiency measures for resource utilization and response time?

Software Design and Development

- After the clarification of the requirements, the Design phase, i.e. the first model and representation of the product to be developed, is started. At this stage, pre-development activities of the product are carried out with the documentation called Technical Design, Solution Design, etc. by utilizing the requirements. In this respect, experience and knowledge have an important place in the realization of designs. The "design and development" stage in the software development process is the stage of transforming the demands transformed into requirements into a workable state.

Software Design Process

- Component design
- Algorithm design
- Architectural design
- Data structure design
- Interface design

Development Phase

- After the design phase is realized, the coding of the software, i.e. the development phase, begins. Design and development activities can be intertwined actions and differ according to software process models.

Verification Process

- Are the requirements set correctly?
- Are the requirements consistent?
- Are the requirements complete (External completeness / Internal completeness)?
- Are the requirements realistic?
- Does each requirement describe something desired by the user?
- Can the requirements be verified?
- Are the requirements traceable?

Doc. Dr. Mehmet Akif Cifci

- Technical University of Vienna (Austria)
- University of Klaipeda (Lithuania)
- Bandirma Seventeen Eylul University

To Follow and Connect

https://github.com/themanoftalent
https://www.linkedin.com/in/themanoftalent/
https://www.researchgate.net/profile/Mehmet-Akif-Cifci