Requirements Gathering

What is this?

- Process to gather software requirements from client, analyse and document them
- Goal of this process: Develop and maintain sophisticated and descriptive ‘System Requirements Specification’ document
- Two types of specification:
  - Functional: What the system should do
  - Non-functional: What are the constraints in development of the system

Requirement Elicitation Techniques

- Requirement elicitation is the process to find out the requirements for an intended system by communicating with client, users and others who have a stake in the software system development
- Data gathering techniques:
  - Interview
  - Questionnaires
  - Surveys
  - Task analysis
  - Domain analysis
  - Brainstorming
  - Prototyping
  - Observation

Interview

- Strong medium to collect requirements
- Several types of interview:
  - Structured interview (closed): Every information to gather is decided in advance
  - Non-structured (open): Information to gather is not decided. More flexible
  - Oral/written interview
  - One to one interview/group interview

Surveys

- Conducted among stakeholders by querying their expectation and requirements
- Conduct short meeting to ask for opinions

Questionnaires

- Pre-defined set of objective questions with respective options
- Handed over to participant, they fill it up and returned to be compiled
- Shortcoming: if an option for some is not mentioned in the questionnaire, the issue might be left unattended

Task analysis

- Team of engineers and developers may analyse the operation for which the system is required.
- Study existing software to perform certain operation and collect the requirements proposed

Domain analysis

- Expert people in the domain help to analyse general and specific requirements

Brainstorming

- Informal discussion or debate among various stakeholders and all the inputs are recorded for further requirement analysis

Prototyping

- Build user interface without adding detail functionality for user to interpret the features on intended software product. Feedback from the clients are noted

Observation

- Team of experts visit the client’s organization or workplace.
- Observe the actual working of the existing system
- Draw own conclusion from these observations.

Characteristics of the gathered requirements

- Clear
- Correct
- Consistent
- Coherent
- Comprehensible
- Modifiable
- Verifiable
- Unambiguous
- Traceable
- Credible source

Types of Requirements

- Different types of requirements will be gathering from the elicitation process
  - Functional Requirements
    - Requirements related to functional aspect of system. They define functions and functionality within and from the system
    - Example: Search option given to user to search product listing
    - Example: User should be able to main report to management
    - Example: User is able to make his/her desired picture as the starred picture.

Non-functional Requirements

- Non-functional Requirements
  - Not related to function aspect of software
  - Implicit or expected characteristics of system which users make assumption of
  - Performance, security, usability, compatibility is not function of the system but are required characteristic.

Non-functional Requirements

- Performance requirements/Technical requirements
  - Requirements about resources required, response time, or anything else which has to do with performance
  - Example: User require Internet connection and a browser of xx version to access the web development platform,

Non-functional Requirements

- Operating Requirements
  - List any run-time constraints such as required software or system resources
  - The latest Java version 8 is required to run all the functionalities in the platform

Non-functional Requirements

- Platform Requirements
    - Discuss the target platform
    - Example: The developed system can only run on Android platform
    - Example: System can run on any version of Internet Explorer

Non-functional Requirements

- Usability Requirements
  - Requirements about how difficult it will be to learn and operate the system
  - Example: System must be intuitive or easily understood after reading the rules
  - Example: System interface must be intuitive and easily understood with affordance.
  - Example: Needs user friendly layout that accommodates beginners as well as veterans

Non-functional Requirements

- Security Requirements
  - Requirements about protection of the system and its data.
  - Example: User requires password to login and use certain functionalities such as xxx
  - Administrator login for admin to manage certain functions such as xxx
  - In order to use the system, user has to create a user account with password and username as login credentials