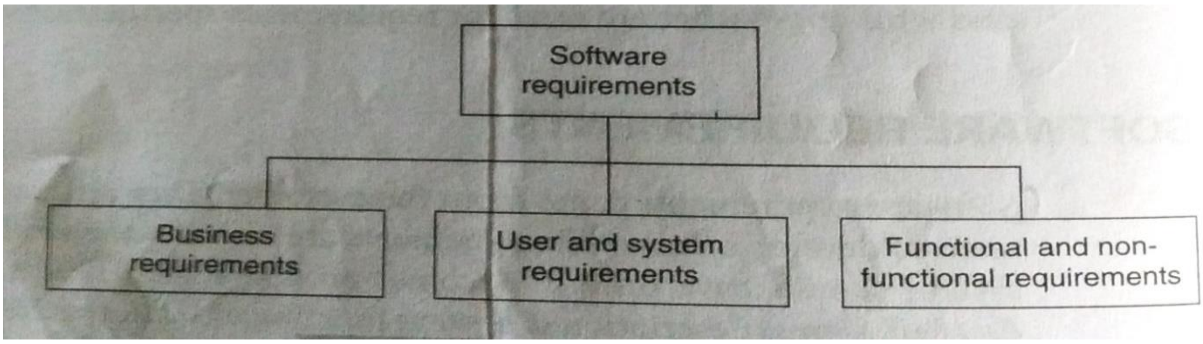
Explain about Software requirements?

- Software requirements come from the customer. They are used to development to develop software. A requirement is a detailed, formal description of system functionalities. A requirement specifies a function that s system or component must able to perform for customer satisfaction.


Software requirements
Business requirements
User and system requirements
Functional and non-functional requirements

Business Requirements: It defines the project goal and the expected business needs and opportunities for the product. Business needs also describe the product domain and product demand. It focuses on data and process analysis which helps to carry out the further software development life cycle. The Business analyst is well versed in understanding the concept of business flow as well as the process being followed in the organisation

User and System Requirements:

- User requirements are the high level abstract stalemates supplied by the customer, end users. These requirements are the functionalities that the system is expected to provide within the certain constraints or environment. These requirements are translated into system requirements keeping in mind user's view. These are generally represented in some natural languages with pictorial representation or tables to understand the requirements. User requirements may be ambiguous or incomplete in description with less product specification and little/software configurations are stated in the user requirements.
- System requirements are the detailed and technical functionalities written in a systematic manner that are implemented in the business process to achieve the goal of user requirements. It can also be set of functional and non-functional requirements. System requirements are often expressed as documents in a structured manner using technical representations.

- In an ATM machine, user requirements allow users to withdraw and deposit cash. The requirements consider customer ID, account type, bank name, consortium, PIN, communication link, hardware, and software. Also, an ATM will service one customer at a time.

Functional Requirements

- Requirements, which are related to functional aspect of software fall into this category. They define functions and functionality within and from the software system. Functional requirements are the behavior or functions that the system must support. Each function receives some user input, processes it, and provides certain out comes. These are the attributes that characterize what the software does to fulfil the needs of the customer.

EXAMPLES:

- Search option given to user to search from various invoices.
- User should be able to mail any report to management.
- Users can be divided into groups and groups can be given separate rights.
- Should comply business rules and administrative functions.
- Software is developed keeping downward compatibility intact.

Non-Functional Requirements

- Non-functional Requirements specify how a system must behave. These are qualities, standards, constraints upon the system services that are specified with respect to a product, organization and external environment. It specifies quality characteristics of a system. Functional requirements are actions that must be performed by the system under development.
- **Non-functional requirements include -**

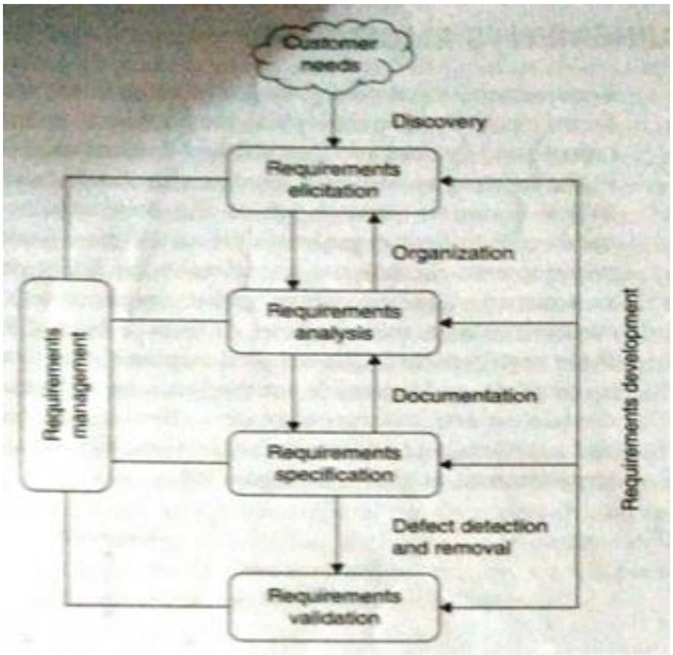
- Security
- Logging
- Storage
- Configuration
- Performance
- Cost
- Interoperability
- Flexibility
- Disaster recovery
- Accessibility

Explain about Requirement Engineering process?

Ans. Requirement Engineering is the disciplined, process oriented approach to the development and management of requirements.

It is a four step process, which includes –

- Feasibility Study
- Requirement Gathering
- Software Requirement Specification
- Software Requirement Validation



Feasibility study:

- ➢ When the client approaches the organization for getting the desired product developed, it comes up with rough idea about what all functions the software must perform and which all features are expected from the software.
- ➢ Referencing to this information, the analysts does a detailed study about whether the desired system and its functionality are feasible to develop.
- ➢ This feasibility study is focused towards goal of the organization. This study analyses whether the software product can be practically materialized in terms of implementation, contribution of project to organization, cost constraints and as per values and objectives of the organization. It explores technical aspects of the project and product such as usability, maintainability, and productivity and integration ability.
- ➢ The output of this phase should be a feasibility study report that should contain adequate comments and recommendations for management about whether or not the project should be undertaken.

Requirement Gathering

- If the feasibility report is positive towards undertaking the project, next phase starts with gathering requirements from the user. Analysts and engineers communicate with the client and end-users to know their ideas on what the software should provide and which features they want the software to include.

- **Requirement Analysis**
- Requirement analysis is carried out after gathering requirements in the requirement elicitation phase. Requirement analysis is an activity which is performed iteratively during the course of software development. Requirement analysis includes various activities, such as classification, organization, prioritization, negotiation and modelling requirements.

- **Software Requirement Specification**
- SRS is a document created by system analyst after the requirements are collected from various stakeholders.
- SRS defines how the intended software will interact with hardware, external interfaces, speed of operation, response time of system, portability of software across various platforms, maintainability, speed of recovery after crashing, Security, Quality, Limitations etc.
- The requirements received from client are written in natural language. It is the responsibility of system analyst to document the requirements in technical language so that they can be comprehended and useful by the software development team.
- SRS should come up with following features:

- User Requirements are expressed in natural language.
- Technical requirements are expressed in structured language, which is used inside the organization.
- Design description should be written in Pseudo code.
- Format of Forms and GUI screen prints.
- Conditional and mathematical notations for DFDs etc.

Software Requirement Validation

After requirement specifications are developed, the requirements mentioned in this document are validated. User might ask for illegal, impractical solution or experts may interpret the requirements incorrectly. This results in huge increase in cost if not nipped in the bud. Requirements can be checked against following conditions -

- If they can be practically implemented
- If they are valid and as per functionality and domain of software
- If there are any ambiguities
- If they are complete
- If they can be demonstrated

Write a short note on Requirement Specification?

- ➢ Requirements specification in systems engineering and software engineering is the direct result of a requirements analysis and can refer to SRS. Software requirement specification (SRS) is a document that completely describes what the proposed software should do without describing how software will do it. The basic goal of the requirement phase is to produce the SRS, Which describes the complete behaviour of the proposed software. SRS is also helping the clients to understand their own needs.

Advantages

Software SRS establishes the basic for agreement between the client and the supplier on what the software product will do.

1. A SRS provides a reference for validation of the final product.
2. A high-quality SRS is a prerequisite to high-quality software.
3. A high-quality SRS reduces the development cost.

Characteristics of SRS

- ➢ Software requirements specification should be accurate, complete, efficient, and of high quality, so that it does not affect the entire project plan. An SRS is said to be of high quality when the developer and user easily understand the prepared document. Other characteristics of SRS are discussed below.

Correct: SRS is correct when all user requirements are stated in the requirements document. The stated requirements should be according to the desired system. This implies that each requirement is examined to ensure that it (SRS) represents user requirements. Note that there is no specified tool or procedure to assure the correctness of SRS. Correctness ensures that all specified requirements are performed correctly.

Unambiguous: SRS is unambiguous when every stated requirement has only one interpretation. This implies that each requirement is uniquely interpreted. In case there is a term used with multiple meanings, the requirements document should specify the meanings in the SRS so that it is clear and easy to understand.

Complete: SRS is complete when the requirements clearly define what the software is required to do. This includes all the requirements related to performance, design and functionality.

Ranked for importance/stability: All requirements are not equally important, hence each requirement is identified to make differences among other requirements. For this, it is essential to clearly identify each requirement. Stability implies the probability of changes in the requirement in future.

Modifiable: The requirements of the user can change, hence requirements document should be created in such a manner that those changes can be modified easily, consistently maintaining the structure and style of the SRS.

Traceable: SRS is traceable when the source of each requirement is clear and facilitates the reference of each requirement in future. For this, forward tracing and backward tracing are used. Forward tracing implies that each requirement should be traceable to design and code elements. Backward tracing implies defining each requirement explicitly referencing its source.

Verifiable: SRS is verifiable when the specified requirements can be verified with a cost-effective process to check whether the final software meets those requirements. The requirements are verified with the help of reviews. Note that unambiguity is essential for verifiability.

Consistent: SRS is consistent when the subsets of individual requirements defined do not conflict with each other. For example, there can be a case when different requirements can use different terms to refer to the same object.

Explain the Components of an SRS?

- ➢ Completeness of specifications is difficult to achieve and even more difficult to verify. Having guidelines about what different things an SRS should specify will help in completely specifying the requirements. The basic issues an SRS must address are:

- Functionality
- Performance

- Design constraints imposed on an implementation
- External interfaces

1. Functional Requirements: Functional requirements specify what output should be produced from the given inputs. So they basically describe the connectivity between the input and output of the system. For each functional requirement:

- 1. A detailed description of all the data inputs and their sources, the units of measure, and the range of valid inputs be specified:
- 2. All the operations to be performed on the input data obtain the output should be specified, and
- 3. Care must be taken not to specify any algorithms that are not parts of the system but that may be needed to implement the system.
- 4. It must clearly state what the system should do if system behaves abnormally when any invalid input is given or due to some error during computation. Specifically, it should specify the behavior of the system for invalid inputs and invalid outputs.

2. Performance Requirements (Speed Requirements)

- ➢ This part of an SRS specifies the performance constraints on the software system. All the requirements related to the performance characteristics of the system must be clearly specified. Performance requirements are typically expressed as processed transaction s per second or response time from the system for a user event or screen refresh time or a combination of these. It is a good idea to pin down performance requirements for the most used or critical transactions, user events and screens.

3. Design Constraints

- ➢ The client environment may restrict the designer to include some design constraints that must be followed. The various design constraints are standard compliance, resource limits, operating environment, reliability and security requirements and policies that may have an impact on the design of the system. An SRS should identify and specify all such constraints.

Standard Compliance: It specifies the requirements for the standard the system must follow. The standards may include the report format and according procedures.

Hardware Limitations: The software needs some existing or predetermined hardware to operate, thus imposing restrictions on the design. Hardware limitations can includes the types of machines to be used operating system availability memory space etc.

Fault Tolerance: Fault tolerance requirements can place a major constraint on how the system is to be designed. Fault tolerance requirements often make the system more complex and expensive, so they should be minimized.

Security: Currently security requirements have become essential and major for all types of systems. Security requirements place restriction s on the use of certain commands control access to database, provide different kinds of access, requirements for different people, require the use of passwords and cryptography techniques, and maintain a log of activities in the system.

4. External Interface Requirements

For each external interface requirements:

1. All the possible interactions of the software with people hardware and other software should be clearly specified,
2. The characteristics of each user interface of the software product should be specified and
3. The SRS should specify the logical characteristics of each interface between the software product and the hardware components for hardware interfacing.

Write a short note on Requirement Analysis?

- ➢ Requirement analysis is carried out after gathering requirements in the requirement elicitation phase. Requirement analysis is an activity which is performed iteratively during the course of software development. Requirement analysis includes various activities, such as classification, organization, prioritization, negotiation and modelling requirements.
- ➢ During Requirement analysis, models are prepared to analyse the requirements. Requirements analysis is critical to the success of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

The following analysis techniques are generally used for the modelling of requirements:

1. Structured Analysis-DFD
2. Data-oriented analysis-ER
3. Object- oriented analysis-Class
4. Prototyping – Trhowaway,Evolutionary

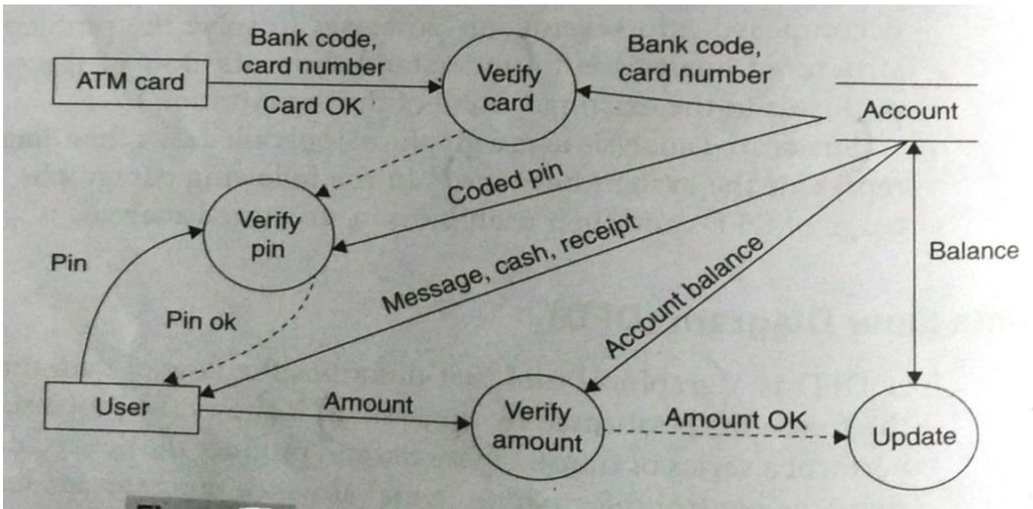
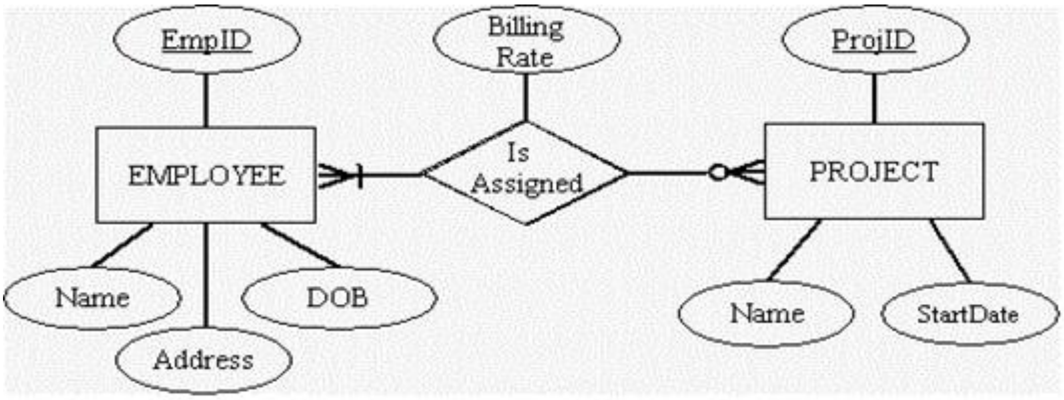


Figure for cash withdrawal from an ATM



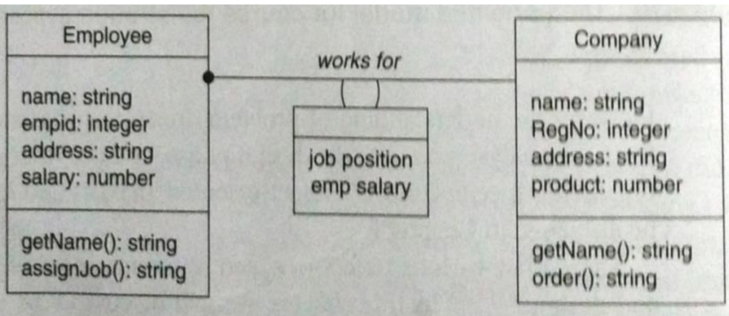


Figure Class diagram and association of employee in a company

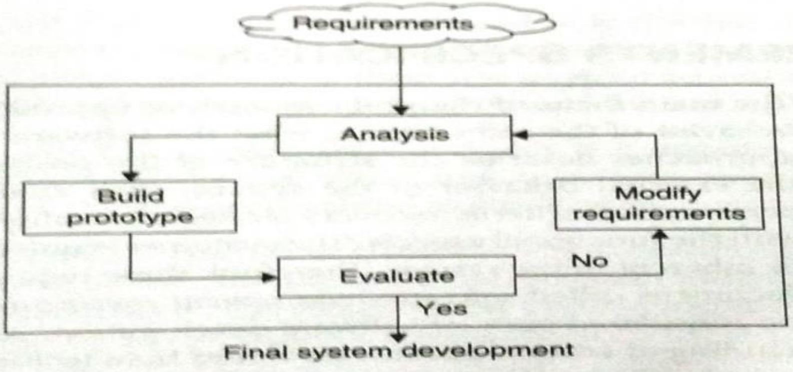


Figure Throwaway prototyping