Software Requirement 22/3/23

* IEEE defines requirement as -

(1) Condition or capability needed by a user to solve problem or achieve an objective.

(2) A condition or capability that must be made or by a system to satisfy a contract.

The description of services and constraints are the requirements for the system and the process of finding out, analysis, documenting and the checking the services is called requirement engineering.

* Different types of Requirements :

(a) User Requirements

User requirements are statements in a natural language plus diagrams of what service the system is expected to provide and the constraints under which it must operate.

System requirements set out the system services and constraints in detail. The system requirement document or Functional requirement should be precise. It may serve as a contract between the System buyer and a software developer.

Software Design Specification

A software design specification is an abstract description of the software design which is a basis for more detail design and implementation.

Two kinds of requirment documents:

(i) User requirement definition document URD

(ii) System " specification " SRS

(i) User req. def. doc. : The req. def. is a complete listing of everything the customer proposed

It represents an understanding bet^n the customer and developer.

(ii) System req. specification : SRS restates the user requirement definition in technical terms needed for the development of a system. Technical counterpart to user requirement definition which is written by requirement analysts.

Functional & Non functional Requirement

Software System Requirements are often classified as functional or non functional requirements.

Functional Requirement: A func. req. describes an interaction betn the system and its environment. For eg: to determine functional requirement we decide what states are acceptable once for the system to be in. further, func. req. describes how the system should behave on certain given certain stimuli.

Functional system req. describes the system in detail, the input, function output, exceptions etc

Non Functional Req: These are req. which are not directly concerned with the specific functions delivered by the system. They may relate to emergent system properties such as reliability, response time and store occupancy.

Alternatively they may define constraints on the system such as the capabilities of input device and output device data representations used in system interfaces.

Non functional req. are not always concerned with software systems to be developed.

There are different types of non functional req.:

(1) Product requirement: Requirement which specify product behaviour.

POE

(ii) Organisational Requirement

These are derived from policies and procedures in the customer and developer into process standards which must be used.

(iii) External Requirement

These are all requirements which are derived from factors external to the system and its development system.

23/3/23

Requirement Engineering Process

identifying, specifying, eliciting, analyzing, validating

① ~~Inception~~ : Task where the req. eng.

iasevm

② ~~Eli~~

Requirements Engineering is the process of identifying, eliciting, analyzing, specifying, validating and managing the needs and expectations of stakeholders for a software system. It is an iterative process and involves several steps:

(i) Requirements Elicitation: process of gathering information about needs and expectations of stakeholders for software system. involves surveys, interviews, focus groups to gather info from stakeholders.

(ii) Requirements Analysis: involves analyzing the info gathered in the req. eli step to identify high level goals and obj of software system. involves identifying high level goals and obj of soft. sys.

(iii) Requirements Specification: involves documenting the requirements identified in the analysis step in a clear, concise and unambiguous manner. involves prioritizing and grouping the req. into manageable chunks.

(iv) Requirements Validation: involves checking that the req. are complete, consistent and accurate. involves checking that req. are testable and meet needs & expectations of Stakeholders.

(v) Requirements Management: involves managing the requirements throughout the software development life cycle, including tracking and controlling changes and ~~(vi)~~ ensuring that the req. are still valid and relevant

Tools involved in RE: EASVM

(i) abs report (ii) Questionnaire (survey, poll)

(iii) Use cases (iv) User stories (v) Req. workshop

(vi) Mind mapping (vii) Role playing (viii) Prototyping

Requirement Engineering Process

Advantages

(i) Can improve communication and collaboration between development team and stakeholders

(ii) Ensures software is developed in a cost-effective and efficient manner

Disadvantages

(i) Time consuming and costly if the req. gat. process is not well managed

(ii) Difficult to ensure all stakeholder's needs and expectations

Cost Estimation Technique 29/3/23

Cost estimation technique is a technique to find out cost estimate. The cost estimate is the financial spend that is done on the efforts to develop and test software in SE.

Various techniques or models are available for cost estimation. There are three types of cost estimation:

(i) Empirical Estimation Technique
(ii) Heuristic Technique
(iii) Analytical Estimation Technique

(i) In EET we use formula to find cost using data from project development data.
(ii) For project parameters, we express it with mathematical equations (based on proposed practical model)

Flexible and simple.
Quick decision.

Constructive Cost model undergoes heuristic technique
↓
COCOMO model

(iii) We break down project into smaller parts.
Standard time for resource. ~~under~~

EET: This is a technique or model in which empirically derived formulas are used for predicting the data that are required essential part of the

software project planning step.

HT: This technique is used for solving problems, learning and discovering practical methods which are used for achieving immediate goals.

AET: It is a type of technique that is used to measure work. In this technique first the task is decided or broken down into its basic component operations. If standard time is available from some other source then these sources are applied to each element. ~~It is~~ If there is no such time available then it is based on experience of the work.

Cocomo Model :→ Cocomo model was ~~was~~ proposed by BOSCHE in 1991

→ It is the most complemented and thoroughly implemented used in effort estimation. The model provide the detailed formulae for determining the time scheduling, as well as maintainence effort.

→ ~~The~~ ~~The primary~~

The primary effort factor is the number of source line of code expressed in thousands of delivered source instructions. These instructions include all program instructions, format statements, and job control language statements. They exclude comments and unmodified utility software.

The software development project can be classified into one of the following three categories based on development complexity.

(i) Organic

(ii) Embedded
(iii) Semidetach

(i) Organic: This category encompasses all systems that are small relative to project size and team size and have a stable environment, familiar surroundings and relaxed interfaces.

(ii) Embedded: This class of system is characterized by tight constraints, change in environment and unfamiliar surroundings. Projects of embedded type are novel to the company and usually exhibit temporary constraints.

(iii) Semidetach: The software systems falling under this category are a mix of those of organic and embedded nature.

The basic form of the Cocomo model is based on exclusively on program size expressed in thousands of delivered source instructions.

The formula as used the form

$$Effort = a * KDLOC^{b} \quad \cancel{KSLOC^{b}}$$

and a, b are the two parameters whose specific values are selected upon the class of the software system.

Estimation of development effort

For embedded systems, effort = $3.6 * KDLOC^{1.20}$ per month

For organic, Effort = $2.4 * KDLOC^{1.05}$ "

For semidetach, = $3.0 * KDLOC^{1.12}$ "

Estimation of Development Time

For embedded system, $M = 2.5 * Effort^{0.32}$ month

$M = 2.5 * Effort^{0.33}$ month (organic)

$M = 2.5 * Effort^{0.35}$ month (semi detach)

Q1 Using COCOMO, estimate time required for the following:

(I) A semidetached model of software project of 2000 lines

(II) An embedded model of software of 30,000 lines

(III) An organic model of software of 1,00,000 lines

(IV) An organic model of software of 10,00,000

Q2 Consider a project having 30,000 lines of code which is an embedded software with critical area. Find out the estimation of effort and time.

KLOC = ~~30,000~~ 30

Effort = $3.6 * KDLOC^{1.20}$ per month

$M = 2.5 * Effort^{0.32}$ month

Eff = $3.6 \times 30^{1.20}$ pm = 213.23

$M = 2.5 \times 213.23^{0.32}$ month

= 14 month

Q1 Compare and contrast waterfall model ~~with~~ and spiral model.

Q2 ~~Weber~~ Draw an ER diagram for the relationship of bike manufacturer and dealership.