SOLUTION-2016

Q.1)

(TANAYA JAISWAL 1729088)

a)What is meant by the term life cycle model of software development. ?

b)Differentiate between Heuristic and Empirical methods of software effort estimation.

c)What is meant by structured programming ?

d)What is meant by a software process ? What is the difference between a methodology and a process ?

e)If the phase containment of errors principle is not followed during software development, then would the development cost increase ? Justify your answer.

a) A software development life cycle (SDLC) model is a conceptual framework describing all activities in a software development project from planning to maintenance.

This process is associated with several models, each including a variety of tasks and activities. It provides:

i. Feasibility study
ii. Requirements analysis and specification
iii. Design
iv. Coding
v. Testing

vi. Maintenance

b) A heuristic technique, often called simply a heuristic, is any approach to problem solving, learning, or discovery that employs a practical method not guaranteed to be optimal or perfect, but sufficient for the immediate goals.

Empirical methods main motivation is that it is needed from an engineering perspective to allow for informed and well-grounded decision.

c) Structured programming is a logical programming method that is considered a precursor to object-oriented programming (OOP).Structured programming facilitates program understanding and modification and has a top-down design approach, where a system is divided into compositional subsystems.

d) A software development process, also known as a software development lifecycle, is a structure imposed on the development of a software product. A software process is represented as a set of work phases that is applied to design and build a software product. The software that meets the specification is produced.

A methodology is an approach to “doing something” with a defined set of rules, methods, tests activities, deliverables, and processes which typically serves to solve a specific problem.

A process is simply a well-defined set of steps and decisions points for executing a specific task.

e) A software development life cycle has different distinct development phases. Phase containment of errors means detect and correct the errors within the phase where it's actually lives.

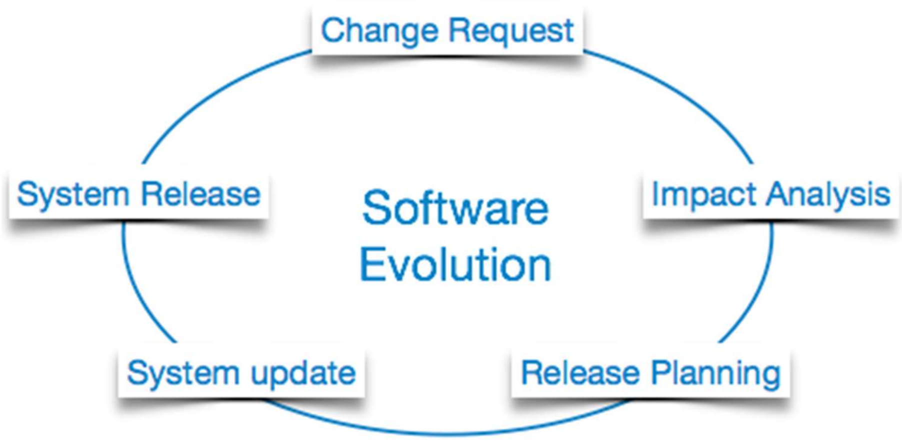
That is a design error should be detected and corrected within the design phase itself rather than detecting

it in the coding phase.

The development cost will increase as each and every step will have to be repeated again increasing not only the cost, but also the time required to complete the project as every change in the code will reflect at each and every step.

Q.2 a) Discuss the evolution of software engineering techniques over the years. (TANISHA BHARDWAJ 1729089)

The process of developing a software product using software engineering principles and methods is referred to as **software evolution.** This includes the initial development of software and its maintenance and updates, till desired software product is developed, which satisfies the expected requirements.



Evolution starts from the requirement gathering process. After which developers create a prototype of the intended software and show it to the users to get their feedback at the early stage of software product development. The users suggest changes, on which several consecutive updates and maintenance keep on changing too. This process changes to the original software, till the desired software is accomplished.

Even after the user has desired software in hand, the advancing technology and the changing requirements force the software product to change accordingly. Re-creating software from scratch and to go one-on-one with requirement is not feasible. The only feasible and economical solution is to update the existing software so that it matches the latest requirements.

Software Evolution Laws

Lehman has given laws for software evolution. He divided the software into three different categories:

- **S-type (static-type) -** This is a software, which works strictly according to defined specifications and solutions. The solution and the method to achieve it, both are immediately understood before coding. The s-type software is least subjected to changes hence this is the simplest of all. For example, calculator program for mathematical computation.
- **P-type (practical-type) -** This is a software with a collection of procedures. This is defined by exactly what procedures can do. In this software, the specifications can be described but the solution is not obvious instantly. For example, gaming software.

- **E-type (embedded-type) -** This software works closely as the requirement of real-world environment. This software has a high degree of evolution as there are various changes in laws, taxes etc. in the real world situations. For example, Online trading software.

b)Describe the Delphi cost estimation technique.

Delphi Technique for Software Estimation :

In software estimation, the project specifications are allotted to the experts and they convey their views/opinions about the same. The total number of experts chosen depends on their availability and the size of the project. Delphi in software estimation takes into account the following key points :

- *Expert's Selection*
- *Briefing about the project to the experts*
- *Gathering an idea/estimate from the experts*
- *Assimilate the ideas and finalize it*

Selection of Experts :

Expert selection must be based on the relevant amount of experience they have in software development. The experts can be within or without the organisation. The number of experts varies according to their availability in the required knowledge domain, complexity involved in the project etc.

Briefing Experts :

Once the team is set, it is time to brief the experts regarding the various aspects of the project like objectives of the estimation, explaining the scope of the project, competition involved in the project, estimated deadline and expected deliverables from the experts. According to the information , the experts prepare their schedule and devise plan to carry forward the software estimation.

Gather idea/estimate from the experts :

Based on the inputs provided, the experts offer an approximation about the size, effort and time to be allocated to the project.

Expert Name	**Size**	**Effort**
Expert 1	x	K
Expert 2	1	A

Assimilation of ideas :

Now when the estimates are defined, we can arrive at a conclusion to combine these estimates. Based on high and low estimates, an average estimate can be drawn, which can be actually used to serve the intended purpose.

Q.3 a) Explain the different phases of the classical waterfall model. Identify the phase in which maximum effort is required?
(TUNEER BHATTACHARYA 1729091)

Following are the phases in a classical waterfall model:-

1)Feasibility study- The main focus of the feasibility study stage is to determine whether it would be *financially* and *technically feasible* to develop the software. The feasibility study involves carrying out several activities such as collection of basic information relating to the software such as the different data items that would be input to the system, the processing required to be carried out on these data, the output data required to be produced by the system, as well as various constraints on the development.

2)Requirements analysis and specification- The aim of the requirements analysis and specification phase is to understand the exact requirements of the customer and to document them properly. This phase consists of two distinct activities, namely requirements gathering and analysis, and requirements specification.

3)Design- The goal of the design phase is to transform the requirements specified in the SRS document into a structure that is suitable for implementation in some programming language. In technical terms, during the design phase the *software architecture* is derived from the SRS document. Two distinctly different design approaches are popularly being used at

present—the procedural and object-oriented design approaches.

1) **Coding and unit testing**- The purpose of the coding and unit testing phase is to translate a software design into source code and to ensure that individually each function is working correctly. The coding phase is also

sometimes called the implementation phase, since the design is implemented into a workable solution in this phase.

5)Integration and system testing- Integration of different modules is undertaken soon after they have been coded and unit tested. During the integration and system testing phase, the different modules are integrated in a planned manner.

6)Maintenance-

The total effort spent on maintenance of a typical software during its

operation phase is much more than that required for developing the

software itself.

Maintenance is the phase in which maximum effort is required.

b)Discuss the circumstances under which the following Software development models would be appropriate (i) Waterfall model (ii) Prototyping model (iii) Incremental Model (iv) Spiral Model.

(I) WATERFALL MODEL:

Projects which not focus on changing the requirements, for example, projects initiated from a request for proposals (RFPs), the customer has a very clear documented requirements

(II)PROTOTYPING MODEL:

This process can be used with any software developing life cycle model. While this shall be chosen when you are developing a system has user interactions.It refers to the activity of creating prototypes of software applications, for example, incomplete versions of the software program being developed. It is an activity that can occur in software development and It used to visualize some component of the software to limit the gap of misunderstanding the customer requirements by the development team.

(III)SPIRAL MODEL:

It is used in the large applications and systems which built-in small phases or segments.The spiral model is favored for large, expensive, and complicated projects.Estimates (i.e. budget, schedule, etc.) become more realistic as work progressed because important issues are discovered earlier.

(IV)INCREMENTAL MODEL:

It is used in shrink-wrap application and large system which built-in small phases or segments. Also, can be used in a system has separated components, for example, ERP system. Which we can start with the budget module as a first iteration and then we can start with the inventory module and so forth.

Q4 (a) Differentiate among basic COCOMO , intermediate COCOMO and complete COCOMO models . (SAYAHNA SENGUPTA 1729158)

Basic COCOMO

(1) Basic COCOMO is good for quick, early, rough order of magnitude estimates of software costs.

(2) It does not account for differences in hardware constraints, personnel quality and experience, use of modern tools and techniques, and other project attributes known to have a significant influence on software costs, which limits its accuracy.

(b) Intermediate COCOMO

(1) Basic COCOMO model assumes that effort and development time are functions of the product size alone.

(2) The intermediate COCOMO model refines the initial estimate obtained using the basic COCOMO by using a set of 15 cost drivers

.

(c) Complete COCOMO

(1) Major shortcoming of both the basic and

intermediate COCOMO models is that they consider

a software product as a single homogeneous entity.

(2) For example, some sub-systems may be considered

as organic type, some semidetached, and some

embedded.

(3) This approach reduces the margin of error in the final

estimate.

(b)Assume that the size of an organic type software product has been estimated to be 32,000 lines of source code. Assume the average salary of software engineers to be Rs. 35,000/- per month. Estimate the nominal (i) cost and (ii) time to develop the software product.

Values ofparameters: Effort (2.4, 1.05); Time (2.5, 0.38)

Effort = a1(KLOC)^a2

Time to develop = b1(Effort)^b2

Effort = 2.4 X 32^1.05 = 91.33 per month

(a) Time to develop = 2.5 X 91.33^0.38 = 13.89 (14 months approx)

(b)Nominal cost = 14 X 35000 = Rs 490000

Q.5 a) List the major responsibilities of a software project manager. (SUSHMITA ROY 1729180)

The major responsibilities of a software project manager are as follows:

1.Activity and resource planning. ...

2.Organizing and motivating a project team. ...

3.Controlling time management. ...

4.Cost estimating and developing the budget. ...

5.Ensuring customer satisfaction. ...

6.Analyzing and managing project risk. ...

7.Monitoring progress.

b)Discuss the different ways in which project teams are organized.

1. Outline Project Requirements

Software development projects are complex. Therefore, many of today's software development teams rely on agile software development.

2. Define Software Project Team – Roles and Responsibilities

3. Project Team Structure

Q.6 a) Differentiate betweenfunctional and non-functional requirements, giving examples from the operation of an ATM.
(SHREYANGI SAXENA 1729219)

NON-FUNCTIONAL REQUIREMENTS:

These requirements are qualities or standards that the system under development must have or comply with, but which are not tasks that will be automated by the system.

1.The keypad is a set of buttons that includes the following: 10 buttons labeled with numbers 0 to 9, an OKAY button, a CLEAR button, a CANCEL button, and dynamic buttons that perform different actions.
2.A PIN must be entered within 20 seconds.
3.The User must enter the PIN correctly within three attempts.
4.ATM suspends further access using a particular card if the associated PIN is entered incorrectly 3 times in succession.

FUNCTIONAL REQUIREMENTS:

A functional requirement document defines the functionality of a system or one of its subsystems. It also depends upon the type of software, expected users and the type of system where the software is used.

1.The card reader determines the account number from the entered card.
2.The ATM is in the idle state when there is no operation.
3.The User is prompted to enter a PIN after a card is entered.
4.A menu is displayed to the User with the following options: Withdraw, Deposit, Pay Bill, Account Update, and Exit.
5.The card is ejected when the session is completed.
6.The ATM defaults to the idle state.
7.The cash dispenser is aware of the cash amount available.
8.The keypad accepts input from the User.
9.Transactions can be canceled at any prompt by the User pressing the CANCEL button.

b) Describe the various purposes for which the SRS document can be used.

To plan a software or project first an idea is needed i.e. what functions it will perform and how it will perform.Next technical expertise needed to develop it.The product should be enough understandable to both the developers and users.All these details comprises in a Software Requirements Specification(SRS).

In other words we can say that a Software Requirements Specification (**SRS)** is a document that lays out the description of the software that is to be developed as well as the intention of the software under development. Software requirements specification shows **what** the software is supposed to do as well as **how** it is supposed to perform. It is written down before the actual software development work starts.

An SRS should address, among other things:

1. **Functionality of the software:** What the software will do
2. **External interfaces:** How the given software will interact with hardware, other software(s) and assumptions on these entities
3. **Required performance levels:** Required performance levels such as response rate, recovery rate etc. of the software
4. **Quality attributes:** The non-functional factors that are used to evaluate the performance of the software, such as security, safety, portability etc
5. **Design constraints:** Any operating system limitations (e.g.: the stock exchange software will only run on Windows), implementation language etc that will affect or limit the design of the software.

Software requirements specification(**SRS**) is important for developers because it **minimizes** the amount of time and effort developers have to expend to achieve desired software goals. It thus **reduces** development cost. This also **benefits** the client company because the lesser the development cost, the lesser the developers will charge from the client. And, if composed properly, an **SRS** ensures that there is less possibility of future redesigns as there is less chance of mistake on the part of developers as they have a clear idea on the functionalities and externalities of the software. It also helps **clear** any communication problems between the **client** and the **developer**. Furthermore, an **SRS** serves to form a foundation of **mutual agreement** between the client and the developer (supplier). It also serves as the document to **verify** the testing processes.

Q-7 Write short notes on any two:

A) Risk Management

B) Software configuration management

C) RAD model

D) Function point metric

E) Project Scheduling

(SAGNIK MISRA 1729152)

a) **Risk Management** - Risk management is the process of identifying, assessing and controlling threats to an organization's capital and earnings. These threats, or risks, could stem from a wide variety of sources, including financial uncertainty, legal liabilities, strategic management errors, accidents and natural disasters. IT security threats and data-related risks, and the risk management strategies to alleviate them, have become a top priority for digitized companies. As a result, a risk management plan increasingly includes companies' processes for identifying and controlling threats to its digital assets, including proprietary corporate data, a customer's personally identifiable information and intellectual property.

b) **Software Configuration Management** - A software is a collection of items like programs, data, documents etc. that can be changed easily. Throughout the software development cycle the software design documents, software code or software requirements document etc. are changed often and it is very important that the changes done in software are done in a controlled manner.

Software Configuration Management(SCM**)** is the discipline for systematically controlling the changes in software and supporting documents (like Test Cases, Test Plan, Design Documents, SRS etc.) during the software development life cycle.

Software configuration management is a process independent of the development process largely because most development models cannot accommodate change at any time during development. SCM can be considered as having three major components:

- Software Configuration Identification
- Change Control
- Status Accounting and Auditing

c) **RAD Model** - The RAD (Rapid Application Development) model is based on prototyping and iterative development with no specific planning involved. The process of writing the software itself involves the planning required for developing the product.

Rapid Application Development focuses on gathering customer requirements through workshops or focus groups, early testing of the prototypes by the customer using iterative concept, reuse of the existing prototypes (components), continuous integration and rapid delivery.

Rapid application development is a software development methodology that uses minimal planning in favor of rapid prototyping. A prototype is a working model that is functionally equivalent to a component of the product.

In the RAD model, the functional modules are developed in parallel as prototypes and are integrated to make the complete product for faster product delivery. Since there is no detailed preplanning, it makes it easier to incorporate the changes within the development process.

d) **Function Point Metric** - A Function Point (FP) is a unit of measurement to express the amount of business functionality, an information system (as a product) provides to a user. FPs measure software size. They are widely accepted as an industry standard for functional sizing.

A function point calculates software size with the help of logical design and performance of functions as per user requirements. It also helps in determining the business functionality of a software application. A function point has a number of benefits, including increase in productivity and reduction in the risk of inflation of created code. Function points can be derived from a software's requirements and can be estimated in the early phases of software development, before the actual lines of code can be determined. The number of function points in a code depends on function complexity.

e) **Project Scheduling** - The project schedule is the tool that communicates what work needs to be performed, which resources of the organization will perform the work and the timeframes in which that work needs to be performed. The project schedule should reflect all of the work associated with delivering the project on time. Without a full and complete schedule, the project manager will be unable to communicate the complete effort, in terms of cost and resources, necessary to deliver the project.

Online project management software allows project managers to track project schedules, resources, budgets and project related assets in real time. The project schedule can be viewed and updated by team members associated with the project, keeping everyone well informed on the overall project status.