B.E.

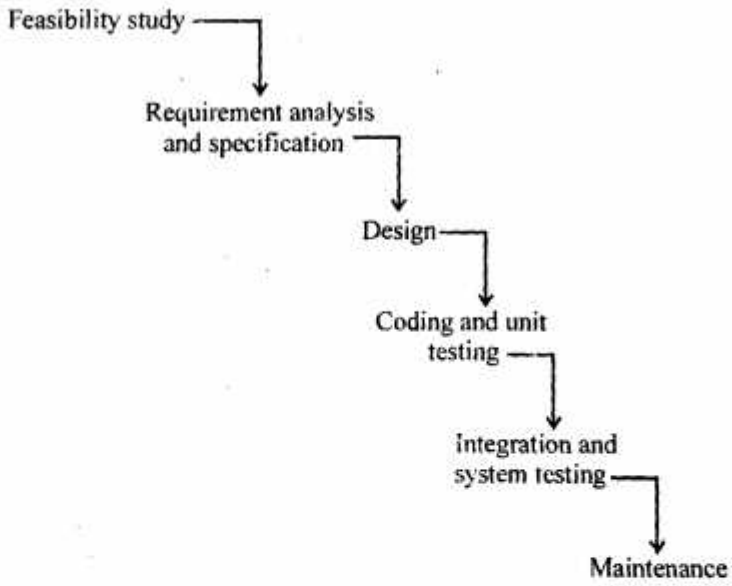
Sixth Semester Examination, Dec-2009

Principles of Software Engineering (CSE-302-E)

Note : Attempt any *five* questions.

Q. 1. (a) Explain water fall model of software development. Also discuss its merits and demerits.

Ans.



The different phases of this model are feasibility study, requirements analysis and specification, design coding and unit testing, integration and system testing and maintenance. The different phases starting from the feasibility study to the integration and system testing phase are known as the development phases. The part of life cycle model between the feasibility study and product testing and delivery is known as the development part.

Demerits :

* The classical waterfall model is an idealistic one since it assumes that no development error is ever committed by the engineers during any of the life cycle phases.
* If any error in above phase then travel one by one phase above and detect error.

Q. 1. (b) What are the various characteristics for the selection of a life cycle model?

Ans.

(i) It is necessary for a development team to adhere to a suitable life cycle model.

(ii) It encourages development of software in a systematic and disciplined manner.

(iii) A program is developed by a single programmer he has the freedom to decode the exact step through which he will develop the program.

(iv) When a software product is developed by a team. It is necessary to have a precise understanding among the team members as to when to do what.

(v) A software development problem is divided into several parts and the parts are assigned to the team members.

(vi) From then on, suppose the team members are allowed the freedom to develop the parts assigned to them whatever way they like.

(vii) It is possible that one member might start writing the code for his part, another might decide to prepare the test documents first and some other engineer might begin with the design phase of the parts assigned to him.

Q. 2. (a) What is SRS? What are the characteristics of a good SRS?

Ans. After the analyst has collected all the required information regarding the software to be developed and has removed incompleteness, inconsistencies and anomalies from the specification, he starts to systematically organize the requirements in the form of an SRS document (Software Requirements specification). The SRS document usually contains all the user requirements in an informal form.

Some of the important categories of users of SRS document and their needs are as follows :

1. User, customers and marketing personnel
2. Software developers
3. Test engineers
4. User documentation writers
5. Project Managers

Characteristics of a Good SRS Document :

1. **Concise :** The SRS document should be concise and consistent and complete.

2. **Structured :** The SRS document should be well structured, which is easy to understand and modify.

3. **Block-Box View :** This means that the SRS document should specify the external behaviour of the system and not discuss the implementation issues.

4. **Conceptual Integrity :** The SRS document should exhibit conceptual integrity so that the reader can easily understand the content.

5. **Response to Undesired Events :** The document should characterize acceptable response to undesired events.

6. **Verifiable :** This means that SRS document should be possible to determine whether or not requirement have been met in an implementation.

Q. 2. (b) Explain COCOMO model in detail.

Ans. COCOMO Model :

COCOMO (Constructive Cost Estimation Model) was proposed by Boehm 1981. Boehm postulated that any software development project can be classified into 3 categories :

1. **Organic :** Development project will be of organic type, if the size of the development team is reasonably small and the team members are experienced in developing similar type of projects.

2. **Semidetached :** Development project can be considered to be of semidetached type, if the development team consist of a mixture of experienced and inexperienced staff. Team member may have limited experience on related systems.

3. Embedded : A development project is considered to be of embedded type, if the software being developed is strongly coupled to complex hardware.

Basic COCOMO Model : Basic COCOMO model gives an approximate estimate of the project parameters. The basic COCOMO estimation model is given by following expression :

$$\text{Effort} = a_1 \times (\text{KLOC})^{a_2} \text{ PM}$$

$$T_{dev} = b_1 \times (\text{effort})^{b_2} \text{ Months}$$

Where KLOC is the estimated size of the software product expressed in kilo lines of code (KLOC).

a_1, a_2, b_1, b_2 are constant for each category of software product.

T_{dev} is the estimated time to develop the software effort is the total effort required to develop the software.

Q. 3. (a) Draw level-1 DFD for the library management system.

Ans. Level-1 DFD : Level-1 DFD, examine the high level function requirements. If there are between three to seven high level function requirements, then these can be directly represented as bubbles in the level-1 DFD. We can then examine the input data to these functions and the data output by the these functions and represent them appropriately in the diagram.

If a system has more than seven high level requirements, then some of the related requirements have to be combined and represented in the form of a bubble in the level-1 DFD. These can be split in the lower DFD levels. If a system has less than three high level functional requirements then some of the high level requirements need to be split into their subfunctions so that we have roughly about five to seven bubbles on the diagram.

Q. 3. (b) What are different risk management activities?

Ans. Project Risk : Project risk concern various form of budgetary schedule, personnel, resources and customer related problems.

Technical Risk : Technical risk concern potential design implementation, interfacing testing and maintenance problems.

Business Risk : Business risk include risk of building an excellent product that no one wants, Closing budgetary or personnel commitments etc.

* It is necessary to anticipate and identify different risk that a project may be susceptible to, so that contingency plans can be prepared to contain the effects of each risk.
* The content, risk management aims at reducing the impact of all kinds of risk that might affect a project.
* Risk management consists of three essential activities risk identification, risk assessment and risk containment.

Q. 4. (a) What is software quality? Discuss the software quality attributes.

Ans. A quality management system is the principal methodology used by organization to ensure that the products they develop have the desired quality.

Quality System Activities : The quality system activities encompass the following :

* Auditing of the project

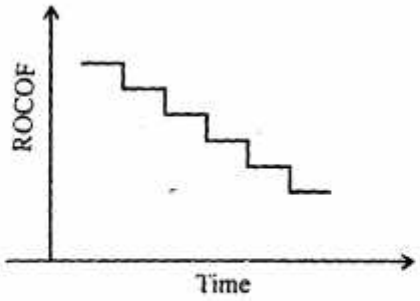
- Review of the quality system
- Development of standards, procedures and guidelines
- Production of reports for the top management summarizing the effectiveness of the quality system in the organization.

A good quality system must be well documented without a properly documented quality system, the application of quality controls and procedures became ad hoc, resulting in large variations in the quality of the product delivered. Also an undocumented quality systems sends clear messages to the staff about the attitude of the organization towards quality assurance.

Q. 4. (b) What is software reliability? Explain any one reliability model in detail.

Ans. Reliability of a software product essentially denotes its trustwothiness or dependently. Alternatively reliability of a software product can also be defined as the probability of the product working correctly over a given time of period.

Jelenski and Moranda Model : The simplest reliability growth model is a step function model where it is assumed that the reliability increase by a constant increment each time an error is detected and repaired.



The simplest model of reliability which implicitly assumes that all errors contribute equally to reliability growth is highly unrealistic since we already know that correction of different errors contribute differently to reliability growth.

Q. 5. (a) What are the key concepts in designing a software?

Ans. Different modules required to implement the design solution :

- Control relationship among the identified modules. The relationship is also known as the call relationship or invocation relationship among modules.
- Interface among different modules. The interface among different modules identifies the exact the data items exchanged among the modules.
- Data structures of the individual modules.
- Algorithms required to implement the indivisual module.
- A good software design is seldom achieved by using a single step procedure but requires several iterations.

We can broadly classify the design activities into two important parts :

- Preliminary.
- Detailed design.

Q. 5. (b) Explain different levels of CMM.

Ans. Level-1 Initial : A software, development organization at this level is characterized by adhoc activities for few or no processes are defined and followed.

Level 2 : Repeatable : At this level the basic project management practices such as tracking cost and schedule are established.

Level 3 : Defined : At this level the process for both management and development activities are defined and documented.

Level 4 : Managed : At this level, the focus on software metrics. Two types of metrices are collected.

Level 5 : Optimizing : At this stage, the process and the product metrics are collected. Process and product measurement data are analyzed for continuous process improvement.

Q. 6. (a) What is software testing? Why do we test? What are different testing principles?

Ans. Testing a program consists of subjecting the program to a set of test input and observing of the program behaves as expected. If the program fails to behave as expected, then the conditions under which failure occurs are noted for latter debugging and correction.

- A failure is a manifestation of an error. But the mere presence of an error may not necessary lead to failure.
- A test case is the triplet [I, S, O] where I is the data input to the system, S is the state of the system at which the data is input and 0 is expected output of the system.
- A test suit is the set of all cases with which a given software product is to be tested.
- The aim of testing process is to identify all defects existing a software product.

Q. 6. (b) Differentiate between the following :

(i) Black box and white box testing

(ii) Alpha and Beta testing

Ans. (i) Black Box and White Box Testing :

Black Box Testing : In black box testing test case are designed from an examination of the input/output values only and no knowledge of design or code is required. The following are the two main approaches to designing black box test cases :

- Equivalence class partitioning.
- Boundary value analysis.

White Box Testing : Each testing strategy is based on some heuristic, one white box testing strategy is said to be stronger than another strategy if all types of error detected by the first testing strategy (say B) are also detected by the second testing strategy say (A) and the second strategy additionally detects some more types of error.

(ii) Alpha and Beta Testing :

Alpha Testing : Alpha testing refers to the system testing carried out by the test team within the developing organization.

Beta Testing : Beta testing is the system testing performed by a select group of friendly customers.

Q. 7. (a) Design various test cases to find out the roots of a quadratic equation using various methods of functional testing.

Ans. The following sequence of ste : that need to undertaken for deriving the path coverage based test

cases of a program :

1. Draw the flow control graph.
2. Determine V(G).
3. Determine the basis set of linearly independent path.
4. Prepare the test case that will force execution of each path in the basis test.

Another interesting application of cyclometric complexity of program is follows experimental studies indicate there existing in the code as well as the time required to find and correct such errors. It is also generally accepted that the cyclomatic complexity of a program is a indication of the psychological complexity or the level of difficulty in understanding the program.

Q. 7. (b) Explain the concept of software re-engineering.

Ans. Software reverse engineering is the process of recovering the design and the requirement specification of a product from an analysis of a code. The purpose of reverse engineering is to facilitate maintenance work by improving the understandability of a system and produce the necessary documents for a legency system. Reverse engineering is becoming important, since legency software product lack proper documentation and are highly unstructured. Even well design products become legency software as their structure degrades because of a series of maintenance efforts implement over a period of use.

The first stage of reverse engineering usually focuses on carrying out cosmetic changes to the code to improve its readability, structure and understandability without changing any of its functionalities.

Q. 8. Write short note on the following :

(a) Coupling and Cohesion

(b) ISO 9000

(c) Data Dictionary

Ans. (a) Coupling and Cohesion : The primary characteristic of a neat module decomposition are high cohesion and low coupling. Cohesion is the measure of the functional strength of a module whereas the coupling between two modules is a measure of the degree of interdependence or interaction between the two modules.

A module having high cohesion and low coupling is said to be functionally independent of other modules.

Classification of Cohesiveness : Following are classes of cohesion :

1. Coincidental cohesion
2. Logical cohesion
3. Temporal cohesion
4. Procedural cohesion
5. Communicational cohesion
6. Sequential cohesion.