

Q.2 a. What is the role of software engineering? (4)

Answer:

Role of software engineering with reference to producing good quality software , maintainable software, and on time within budget.

b. Explain the waterfall model in detail. In what kind of projects this model is applicable? (12)

Answer:

Waterfall model is a sequential model. Its phases like requirement analysis & specification, design, implementation & unit testing, Integration and system testing, maintenance with diagram

The model is applicable to only those projects where requirements have been settled as it expects complete and accurate requirements.

Q.3 a. What is requirement elicitation? (2)

Answer:

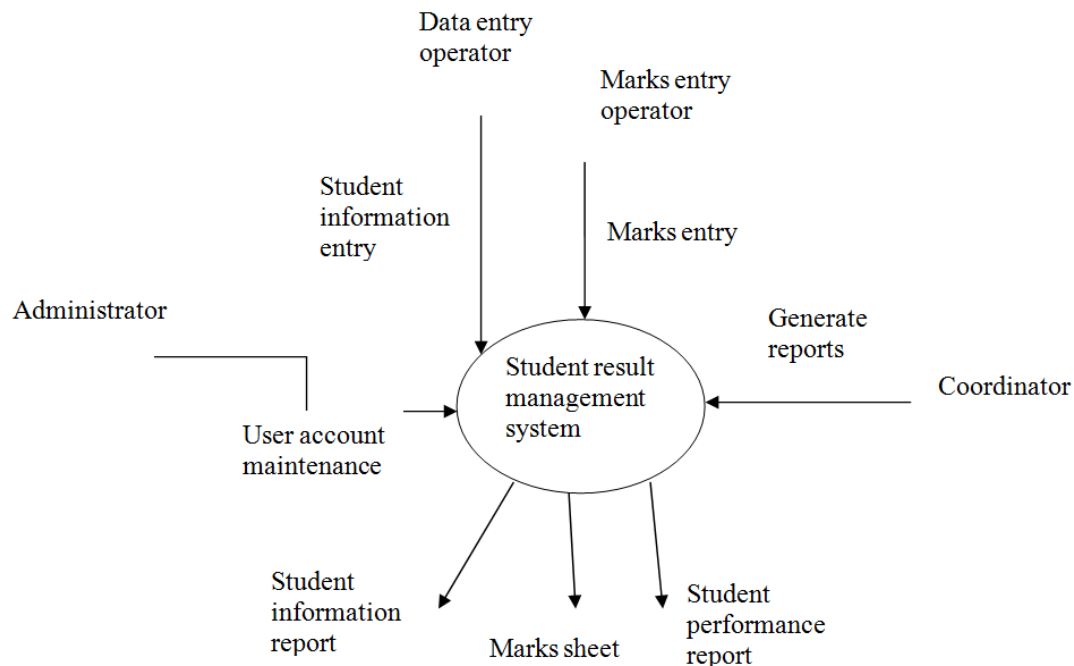
Activity that helps to understand the problem to be solved, requirements are gathered by various methods
Most communication intensive activity of software development.
Name all methods of requirement elicitation

b. Explain different symbols used for DFD. (4)

Answer: symbols for data flow, process, source/sink, data store.

c. There is need to automate student result management system of an institute. Draw context diagram for this system and give the explanation. (10)

Answer:



Q.4 a What is the need of a software prototype? (2)

Answer:

The prototype is used to refine the requirements and prepare the final specifications document. It is evaluated by the customer and it is reasonable to expect that the resulting specification document will be correct. When the prototype is created, it is reviewed by the customer. Typically this review gives feedback to the developers that help to recover uncertainties in the requirements of the software.

b. What are the features of Agile model? (4)

Answer: Iterative, customer-feedback, test driven development

c. Discuss in detail RAD model for software development. (10)

Answer: Four phases of the model: planning phase, user description, construction phase, and cut over phase

Q.5 a. What is module cohesion? Explain its types. (10)

Answer: Measure of degree to which the elements of a module are functionally related.

Types: Functional, sequential, communicational, procedural, temporal, logical and coincidental.

b. What should be a good strategy for designing a software? (6)

Answer: Abstraction, modularity, structural partitioning, information hiding, etc.

Q.6 a. What are the steps to analyze and design an object oriented system? Explain its features. (12)

Answer:

Create use case model

Draw activity diagram

Draw interaction diagram

Draw class diagram

Design state chart diagram

Draw component and development diagram

b. Discuss polymorphism and encapsulation in object oriented design. (4)

Answer:

Discuss polymorphism and encapsulation in object oriented design. (4)

Polymorphism is abstracting just the interface of an operation and leave the implementation to subclasses.

Encapsulation also known as information hiding consists of separation of the external aspects of an object from the internal implementation details of the object.

Q.7 a. What are design guidelines for component based design? Explain briefly. (8)

Answer:

The guidelines apply to components, their interface and the dependencies and inheritance characteristics that have an impact on the resulting design.

b. Mention the design issues in user interface design. (8)

Answer:

The design issues are response time, help facilities, error handling, menu and command labeling, application accessibility and internationalization.

Q.8 a. What is the difference between verification and validation? (4)

Answer:

The purpose of verification is to check the software with its specification at every development phase such that any defect can be detected at early stage of testing and will not be allowed to propagate further. That is why verification can be applied to all the stages of the SDLC. So verification refers to the set of activities that ensure that software correctly implements its function. However, as we progress down to completion of one module or system development, the scope of verification decreases. The validation process starts replacing the verification in the later stages of SDLC. The validation is a very general term to test the software as a whole in conformance with the customer expectations.

b. Compare static and dynamic testing. (4)

Answer:

Static Testing Technique

It is a technique for assessing the structural characteristics of source code, design specifications, or any notational representation that conforms to well-defined syntactic rules [16]. It is called as *static* as we never execute the code in this technique. For example, the structure of code is examined by the teams but the code is not executed.

Dynamic Testing Technique

All the methods to test the software which execute the code, are known as dynamic testing techniques. In this technique, code is run on a number of inputs provided by the user and corresponding results are checked.

c. What is black box testing? Explain boundary value analysis method for testing. (8)

Answer:

Black Box Testing takes care of input given to the system and output are received after processing in the system. What is being processed in the system? How is the system performing operation? Black box testing is not concerned with these questions. It checks functionality of the system only. That is why, the term black-box, is used for the system. It is also known as functional testing. This type is used for system testing under validation.

Since effective test case design demands that test case should be designed such that they maximize the probability of finding the errors. BVA technique addresses this issue. With the experience of testing team, it has been observed that test cases, which are designed with boundary input values, have a high chance to find the errors. It means that most of the failures happen at boundary values.

Thus, BVA is considered as a technique that will uncover the bugs at the boundary of input values. Here boundary means the maximum or minimum value taken by the input domain. For example, if A is an integer between 10 and 255, then boundary checking can be on 10(9,10,11) and on 255(256,255,254). Similarly, B is another integer variable between 10 and 100, then boundary checking can be on 10(9,10,11) and 100(99,100,101)

In this method, the test cases are designed by holding one variable at its extreme value and other variables at their nominal values in the input domain. The variable at its extreme value can be selected at:

- a) Minimum value (Min)
- b) Value just above the minimum value (Min^+)
- c) Maximum value (Max)
- d) Value just below the maximum value (Max^-)

Let us take the example of two variables A and B. If we consider all above combinations with nominal values, then following test cases (see Fig.) can be designed:

1. $A_{\text{nom}}, B_{\text{min}}$
2. $A_{\text{nom}}, B_{\text{min}^+}$
3. $A_{\text{nom}}, B_{\text{max}}$
4. $A_{\text{nom}}, B_{\text{max}^-}$
5. $A_{\text{min}}, B_{\text{nom}}$
6. $A_{\text{min}^+}, B_{\text{nom}}$
7. $A_{\text{max}}, B_{\text{nom}}$
8. $A_{\text{max}^-}, B_{\text{nom}}$
9. $A_{\text{nom}}, B_{\text{nom}}$

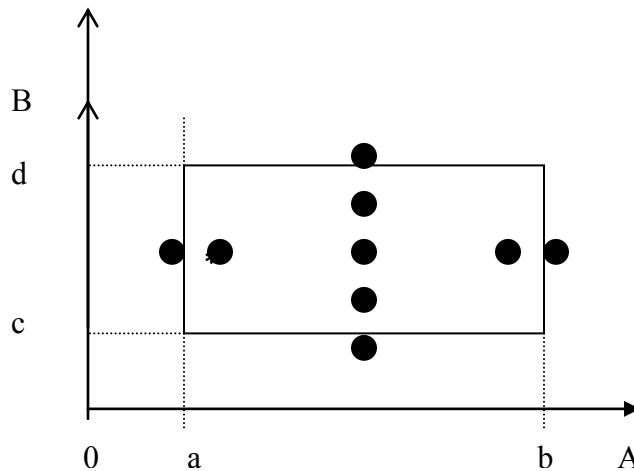


Fig. Boundary Value checking

It can be generalized that for n variables in a module, $4n+1$ test cases can be designed with boundary value checking method.

Q.9 a. Explain the change control process as a part of software configuration management. (8)

Answer: Change control process flow starting from the need for change to distribute the new version.

b. What is the difference between quality control and quality assurance? (2)

Answer:

Quality Control is basically related to software product such that there is minimum variation, according to the desired specifications. This variation is checked at each step of development. Quality control may include the following activities: Reviews, Testing using manual techniques or with automated tools (V & V).

Quality Assurance is largely related to the process. In addition, quality assurance activities are in management zone. Therefore auditing and reporting of quality based on quantitative measurements are also performed. The goal is to provide the data necessary to be informed about quality to the management. It is management's responsibility to address the problems.

c. Explain the process of FPA metric to calculate the size of software. (6)

Answer:

The process used to calculate the function points is given below :

1. Determine the type of project for which function point count is to be calculated. For example, Development Project(A new project), Enhancement Project
2. Identify the counting scope and the application boundary.

TEXT BOOK

- I. Software Engineering, Ian Sommerville, 7th edition, Pearson Education, 2004