Q.1(a) Attempt any THREE of the following : [12]

Q.1(a) (i) What is software? What are its characteristics? [4]

(A) Software can be defined as various kind of programs used to operate computers and related devices.

The different characteristics of software are :

i) Correctness : The software should meet all the specifications stated by the customer.
ii) Usability : The amount of efforts or time required to learn how to use the software should be less. It should be user-friendly.
iii) Reliability : The software product should not have any defects, nor should it fail at the time of execution.
iv) Integrity : A software should not have any side-effects i.e. it should not affect the working of another application.
v) Efficiency : The software should make effective use of the storage space and execute commands as per desired timing requirements.
vi) Security : The software shouldn't have ill effects on data or hardware.
vii) Safety : The software should not be hazardous to the environment.
viii) Maintainability : Maintenance of the software should be easy for any kind of user.
ix) Flexibility : Changes in the software should be easy to make.
x) Extensibility : It should be easy to increase the functions performed by it.
xi) Scalability : It should be very easy to upgrade it for more work.
xii) Testability : Testing the software should be easy.
xiii) Modularity : If the software is divided into separate independent parts that can be modified, tested separately, it has high modularity.
xiv) Reusability : If we are able to use the software code with some modifications for different purpose then we call software to be reusable.
xv) Portability : The ability of software to perform some functions across all platforms is called portability.
xvi) Interoperability : Interoperability is the ability of software to exchange information with other applications and make use of information transparently.

Q.1(a) (ii) Define software testing. State principles of software testing. [4]

(A) i) Software testing is an umbrella activity in which the developed software is test with the intension of finding an error, faults, defects present in the software (syntactical & logical errors).
ii) Testing objectives
   a) Testing is a process of execute a program with the intension of finding an error.
   b) A good test-case is one that has high probability of finding as an yet undiscovered errors.
   c) A successful test is the one that uncoveres an undiscovered errors.
iii) Testing principles
   a) All test cases should be traceable to the customer's requirements.
   b) Test should be planned long before the testing begins.
   c) Exhaustive testing is not possible.
   d) To be most effective testing should be conducted by independent third party.

Q.1(a) (iii) With an example, explain how CPM & PERT are useful in software project management. **[4]**

(A) Program Evaluation Review Technique (PERT)

(i) PERT, is used in projects that have unpredictable tasks and activities such as in research and development projects.

(ii) It utilizes three estimates of the time to complete the project: the most probable, the most promising, and the most unfavorable.

(iii) It is a probabilistic tool using several estimates to determine the time completion of the project and to control the activities involved in the project so that it will be completed faster and at a lower cost.

Critical Path Method (CPM)

(i) CPM is a technique that is used in projects that have predictable activities and tasks such as in construction projects.

(ii) It allows project planners to decide which aspect of the project to reduce or increase when a trade-off is needed.

(iii) It is a deterministic tool and provides an estimate on the cost and the amount of time to spend in order to complete the project.

(iv) It allows planners to control both the time and cost of the project.

Disadvantage of PERT and CPM

(i) The Program Evaluation and Review Technique (PERT) is a project management technique or tool which is suitable for projects that have unpredictable activities while the Critical Path Method (CPM) is a project management tool which is suitable for projects that have predictable activities.

(ii) CPM uses a single estimate for the time that a project can be completed while PERT uses three estimates for the time that it can be completed.

(iii) CPM is a deterministic project management tool while PERT is a probabilistic project management tool.

(iv) CPM allows project management planners to determine which aspect of the project to sacrifice when a trade-off is needed in order to complete the project while PERT does not.

Q.1(a) (iv) Explain STD with suitable example. **[4]**

(A) State Transition Diagram

i) In STD we need to identify the state in which the system can work & what event we need to perform to change the state.

ii) STD is used to model the state of the object.

iii) The elements of STD are :

a) Initial state : It denotes the start state of the system. Initial state is denoted by filled or solid circle • : IT is compulsory state.

b) State : It denotes the current state of an object. The state is denoted by rectangle with round corner. ▭

c) Transition : It is used to move from one state to another state when a particular event or action occurs. It is denoted by a directed line & the name of the event & action written on the top of transition arrow.

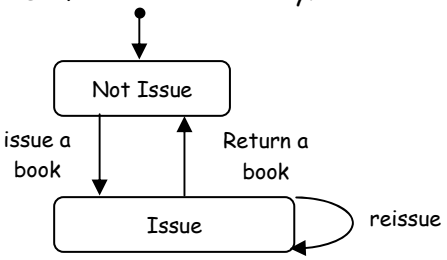
event / action →

Event & action are optional field that means we can write any one either action or action or both.

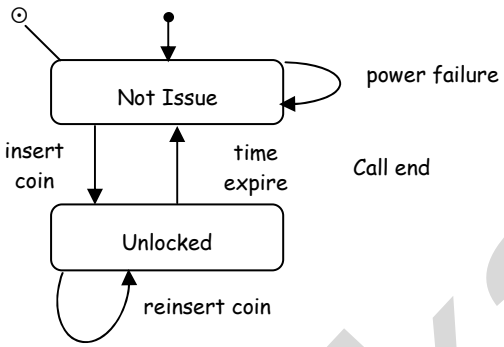
d) End state : It is used to show the end of state of the STD. It is denoted by bulls eye. ⊙

Final state is optional.

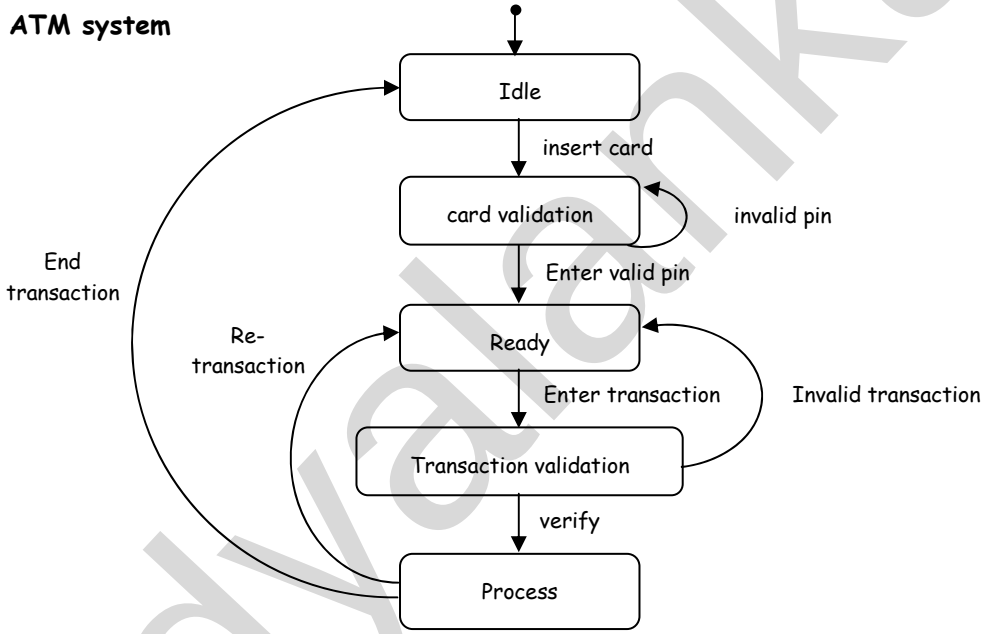
Consider the example of STD of a book in a library.



Coinbox / Turn Around



ATM system



Q.1(b) Attempt any ONE of the following : **[6]**

Q.1(b) (i) Explain the steps involved in requirement engineering. **[6]**

(A) (i) Generally customers & software engineers work together to develop a software but reality is quite different i.e. customer may be located in a different city, customer may have less technical knowledge, customer doesn't have complete idea about the software & so on. Hence, requirement engineering process is required.

The steps required in requirement engineering process are :

1) Identify the stake holders

a) A stake holder is a person who is benefited from the software either directly or indirectly.

b) There are various types of stake holders involved in software development like customer, end-users, business managers, project managers, software engineers, developers, testers, etc.

c) During requirement engineering create a list of people. This list will grow as the software development progresses.

2) Recognizing the multiple viewpoints

a) There are various types of stake holders & each stake holder has its own views for example,
   i) Customer requires a software which is fully functional.
   ii) End users requires a software which is easy to use & learn.
   iii) Business manager requires a software which is developed within a budget but more profitable
   iv) Software developer requires a software which is implemented using well-known programming languages
   v) Sales manager requires a software which is easy to sell in the market.

3) Working toward collaboration

a) Customers must collaborate with software engineer if a successful system is to result.
b) The job of requirement engineer is to identify the areas where stake holders and developers agrees & the areas where there is a conflict between stake holders.
c) Collaboration doesn't mean that the decision will be taken by committee members rather than it should be taken by project champion.

For example, expert engineer, senior technologist, project manager, etc.

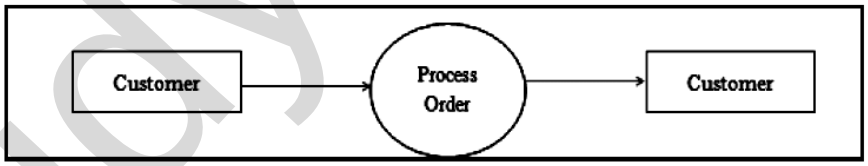
4) *Asking the question*

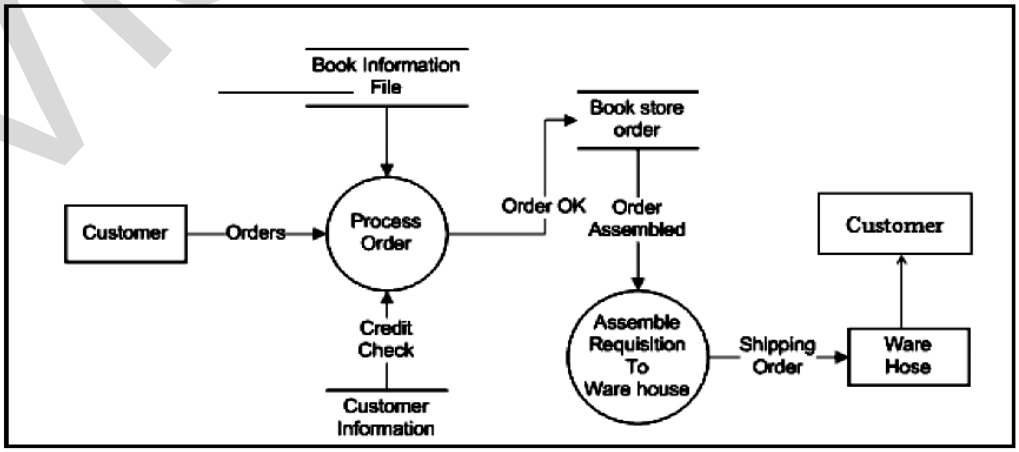
The final set of questions prepared to focuses on effective communication.
   i) Am I asking too many question ?
   ii) Are my questions relevant ?
   iii) Are you right person to answer this question ?
   iv) Are your questions official ?
   v) Can anybody else provide additional information ?

These are the set of questions needs to be answered during requirement engineering.

Q.1(b) (ii) Draw a data flow diagram level 0 and level 1 for a Book Publishing House. **[6]**

(A)





Q.2 Attempt any FOUR of the following : [16]

Q.2(a) Explain the principles of elements of analysis modeling. [4]

(A) (i) The number of analysis modeling methods such as control specification process specification data object description have been developed. The basic principle behind analysis modeling are :

a) The information domain of a problem must be represented & understood.
b) The functions that the software is to perform must be defined.
c) The behavior of the software must be represented.
d) The model that depicts information function & behavior must be partitioned in a manner that uncovers details in the layered fashion.
e) The analysis process should move from essential information towards implementation details.

(ii) The analysis model achieves 3 basic objectives :

a) To describe what the customer requires
b) To establish a basis for the creation of software design.
c) To define a set of requirements that can be validated once software is built.

(iii) The analysis model can be represented as follows :

[Analysis Model Diagram]

(iv) The data dictionary lies at the centre of the model. It is the collection of all the data objects used by the software. The other diagrams that are built around the data dictionary are ER-Diagram DFD & STD.

(v) The ERD shows relationship between data objects.

(vi) DFD is used to perform functional modeling. It serves 2 purposes to provide an indication of how data is transformed through the system & also show the functions that can be performed by the system.

(vii) STD indicates how the system behaves in response to external events. The STD represents the behavior of the system by using the states & the transition of the state.

Q.2(b) What are the generic characteristics of good software testing? [4]

(A) Testing generic characteristics

There are various software testing strategies like unit testing, integration testing, validation testing, system testing and each one strategy must have following generic characteristic.

(i) To perform effective testing a software team should conduct effective formal technical review.

(ii) Testing begins at component level and work towards outwards the integration the entire software.

(iii) Different test strategies are appropriate at different point of time.

(iv) Testing is conducted by the developer of the software and professional software tester group.

(v) Testing and debugging are different activities but debugging must be accommodate in any testing strategy.

The attributes or characteristics of good testing :

(i) A good test case has high probability of finding an error. To achieve this goal software tester must understand the software and develop the test cases that fails the software.

(ii) A good test case is not redundant and there is no point in conducting the test cases that produces the same result. Hence, every test case must have different purpose.

(iii) A good test case should be neither too simple nor too complex. Although it is possible to combine a set of test cases, it may produce the bugs. Hence, it is advisable to apply test cases separately.

Q.2(c) Explain six sigma strategy. **[8]**

(A) Six Sigma Strategy

- The primary goal of Six sigma is to improve customer satisfaction and thereby profitability, by reducing and eliminating defects. Defects may be related to any aspect of customer satisfaction: high product quality, schedule adherence, cost minimization.
- The Six Sigma drive for defect reduction, process improvement and customer satisfaction is based on the "statistical thinking" paradigm, i.e. :
  1. Everything is a process.
  2. All processes have inherent variability.
  3. Data is used to understand the variability and drive process improvement decisions.
- **Core Steps :**
  - **DMAIC (Define, Measure, Analyze, Improve, Control),** used when existing processes have fallen below acceptable levels and incremental improvement is desired; and
  - **DMADV (Define, Measure, Analyze, Design, Verify),** used when new processes or dramatic improvement in existing processes are desired.

DMAIC Method :

D : Define	**Identify and State the Practical Problem.** 1. Initiate the project. 2. Write the project charter (includes a business case, problem/opportunity statement, goal statement, constraints/assumptions, scope, players/rules and a preliminary plan). 3. Identify the customer and translate the **"voice of the customer"** into requirements. 4. Create a high level process diagram/design.
M : Measure	**Validity the Practical Problem by Collecting Data.** 1. Gather data on outputs/outcomes, processes, and inputs. 2. Discover and prioritize customer needs. 3. Identify facts and data that offer clues to quality issues. 4. Create an early sigma measure of the process i.e. measure baseline performance.
A : Analyze	**Convert the Practical Problem to a Statistical One Define Statistical Goal and Identify Potential Statistical Solution.** 1. Analyze the data, using advanced statistics and tools as needed. 2. Find the root cause of quality issues. 3. Develop design alternatives.
I : Improve	**Confirm and Test the Statistical Solution.** 1. Solution and action stage : solve the problem and act on it. 2. May go back to the charter to modify problem/goal statement to reflect discoveries. 3. May confirm with the management. 4. May modify the scope of the project. 5. Implement, manage, and test solutions. Usually, solutions will be thoroughly piloted and tested before full implementation.
C : Control	**Convert the Statistical Solution to a Practical Solution.** 1. Develop and implement monitoring process to track changes and results. 2. Create response plan in case solutions do not work as intended. 3. Help management focus on appropriate metrics to get info on outcomes and processes. 4. Handoff responsibilities to day-to-day operations staff. 5. Ensure management support for long-term goals.

Q.2(d) Explain the RAD with in advantages and disadvantages. [4]

(A) Rapid application development (RAD) is an incremental software development process model that emphasizes an extremely short development cycle. The RAD model is a "high-speed" adaptation of the linear sequential model in which rapid development is achieved by using component-based construction. If requirements are well understood and project scope is constrained, the RAD process enables a development team to create a "fully functional system" within very short time periods (e.g., 60 to 90 days). Used primarily for information systems applications, the RAD approach encompasses the following phases.

(i) Business modeling
(ii) Data modeling
(iii) Process modeling
(iv) Application generation
(v) Testing and turnover

Advantages:

(i) Changing requirements can be accommodated and progress can be measured.
(ii) Powerful RAD tools can reduce development time.
(iii) Productivity with small team in short development time and quick reviews risk control increases reusability of components, better quality.
(iv) Risk of new approach only modularized systems are recommended through RAD.
(v) Suitable for scalable component based systems.

Disadvantages:

(i) RAD model success depends on strong technical team expertise and skills.
(ii) Highly skilled developers needed with modeling skills.
(iii) User involvement throughout life cycle. If developers & customers are not committed to the rapid fire activities necessary to complete the System in a much-abbreviated time frame, RAD projects will fail.
(iv) May not be appropriate for very large scale systems where the technical risks are high.

Q.2(e) What are communication principles? Explain their meaning. [4]

(A) (i) During communication activity requirements are collected from the customer & establish the goals & objectives of communication.

(ii) The core principles of communication are :

1) Listen : Try to focus on speaker's words & if something is unclear then ask for clarification.
2) Prepare : Before communication do some research regarding the topics which are discussed in communication. For example, if you want to conduct a meeting then prepare an agenda before the meetings.
3) Facilitate : Facilitator has the responsibility to keep the communication on right track.
4) Face to face communication : It is always better than any other communication because many concepts gets cleared when we do face to face communication.
5) Take down the notes & document decision : A person involved in communication (recorder) who take down the notes & important points for future purpose.
6) Strive for collaboration : Even a small collaboration helps in project success & build a trust among the team members.
7) Stay focused : If there are more no. of people of involved in communication then it will bounce from one topic to another. Hence it is responsibility of team leader to stay focused with the basic concepts.
8) Unclear picture : If something is unclear then draw it's picture. It's always better to draw a picture if something is unclear in verbal communication.
9) Move on : If you agree to something then move on.

10) Negotiation : If features & functions are not understood & cannot clarify at that moment then simply move on to the next context where some conditions which are more complex in the project then negotiate on that terms conditions. Negotiation is not a context, it works best when both the parties win.

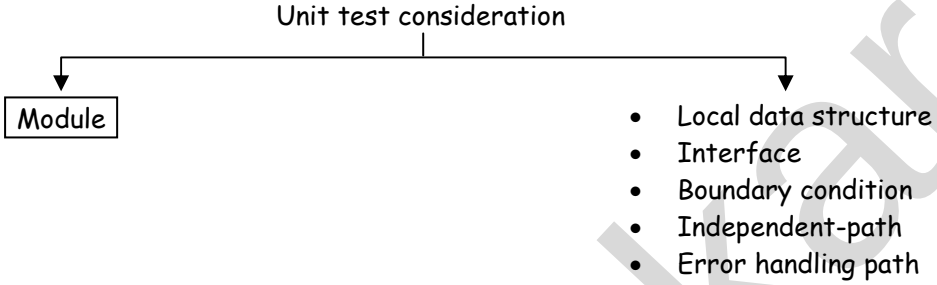
Q.2(f) Explain unit testing. **[4]**

(A) Unit testing

i) A software is divided into modules and each module is separately tested by using unit testing instead complete software.

ii) Once the modules are tested they are integrated using integration testing and finally complete software is tested.

iii)



a) In unit test consideration, the software tester has a set of test cases using that test cases, software tester tries to eliminate the errors such as interface error, boundary condition error, independent path error, etc. in unit testing.

b) Interface error : Interface is tested to ensure that information flows properly into the out of a program.

c) Local data structure : Local data structures are examined to ensure that data stored temporarily maintains it's integrity.

d) Boundary condition : Boundary conditions are tested that ensures the modules works properly at the boundaries because more errors are generates at the boundaries rather than at the centre.

e) Independent path : Independent paths are tested to ensure that all statements within the module must be executed at least once.

f) Finally all error handling paths are test to ensure their validity.

iv) Unit testing is used to identify various types of errors in computation. They are as follows :

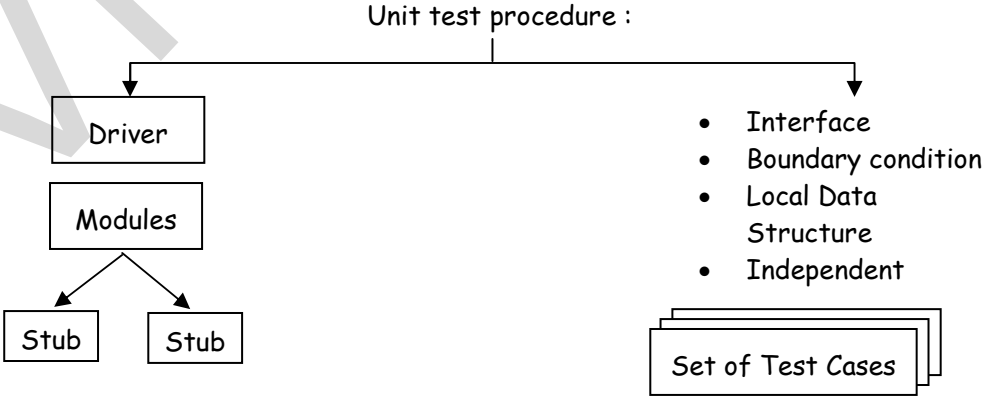
a) Improper operator precedence

b) Improper initialization

c) Precision in accuracy

d) Incorrect symbolic representation

v)



a) Since module is not a standalone program hence two procedures are written i.e. driver and stubs.

b) In most applications driver is nothing but main program that accept input data and pass those data to the module and finally gives the relevant result.

c) Stub is a dummy sub-program which used to serve to replace the module or pass user input.
d) Drivers and stubs represents overhead if drivers and stubs are kept complex but if they are simple then unit testing can be performed fast.

Q.3 Attempt any FOUR of the following : [16]

Q.3(a) Explain McCall's quality factors. [4]

(A) McCall's quality factors

McCall, Richards, and Walters propose a useful categorization of factors that affect software quality. These software quality factors, shown in Figure , focus on three important aspects of a software product: its operational characteristics, its ability to undergo change, and its adaptability to new environments.

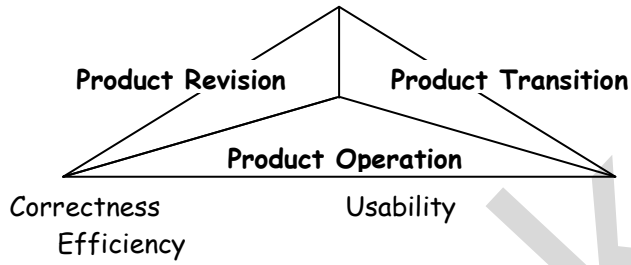


Fig.: McCall's software quality factors.

Correctness : The extent to which a program satisfies its specification and fulfills the customer's mission objectives.
Reliability : The extent to which a program can be expected to perform its intended function with required precision.
Efficiency : The amount of computing resources and code required by a program to perform its function.
Integrity : Extent to which access to software or data by unauthorized persons can be controlled.
Usability : Effort required to learn, operate, prepare input, and interpret output of a program.
Maintainability : Effort required to locate and fix an error in a program.
Flexibility : Effort required to modify an operational program.
Testability : Effort required to test a program to ensure that it performs its intended function.
Portability : Effort required to transfer the program from one hardware and/or software system environment to another.
Reusability : Extent to which a program [or parts of a program] can be reused in other applications—related to the packaging and scope of the functions that the program performs.
Interoperability : Effort required to couple one system to another.

Q.3(b) Describe the principles of deployment. [4]

(A) Deployment principle:

1) **Principle 1:"Manage customer's expectations".**
It always happens that customer wants more than he has started earlier as his requirements. It may be the case that customer gets disappointed, even after getting all his requirements satisfied. Hence at time of delivery developer must have skills to manage customer's expectations.

2) **Principle 2: Assembly and test complete delivery package.**
It is not the case that the deliverable package is 'only software'. The customer must get all supporting and essential help from developer's side.

3) **Principle 3: Record-keeping mechanism must be established for customer support.**
'customer support' is important factor in deployment phase. If proper support is not provided, customer will not be satisfied. Hence support should be well planned and with record-keeping mechanism.

4) **Principle 4: Provide essential instructions, documentations and manual.**
Many times, developer thinks "when project is successful deliverable part is only working program". But realty is that working program is just part of software product.
Actual project delivery includes all documentations, help files and guidance for handling the software by user.

5) **Principle 5: Don't deliver any defective or buggy software to the customer.**
In incremental type of software, software organizations may deliver some defective software to the customer by giving assurance that the defects will be removed in next increment.

Q.3(c) What do you mean by software risk? What are reactive and proactive risk strategies? [4]

(A) Software risk

Risk originate from the Italian work "resicare it means to dare. Risk is a possibility of loss. It may occur or may not occur. So good idea is to identify it, assess its impact and probability and estimate a plan to minimize the consequences of a risk.

Reactive vs Proactive strategy

i) The best method to tackle any kind of risk is to stop them from occurring. Practically this may not be always possible and hence strategies are designed to tackle them at various stages.
ii) Reactive strategy is the one that is decided upon once the risk actually appears. Generally this is in case where the risks are difficult to identify before their occurrence. Once the risk appears after analyzing the various factors, the steps to be taken are decided.
iii) Proactive strategies on the other hand are a better method of tackling risk. The strategies are designed even before the occurrence of a risk. Though they might not be able to stop the risk from occurring entirely but can minimize the impact of the risk once it occurs.

General Categorization of Risk

i) **Known risks :** Known risks are those risks that can be uncovered after careful evaluation of the project plan.
For example unrealistic delivery date, lack of documented requirements of software scope, poor development environment, communication gap between software development team.
ii) **Predictable risks :** Predictable risks are extracted from past project experience.
For example poor communication with customer less effort from team members, etc. Predictable risks can be eliminated by using proactive strategy.
iii) **Unpredictable risks :** Unpredictable risks are the joker in the deck. They can and do occur, but they are extremely difficult to identify in advance. If unpredictable risk occurs then the impact of that risk is high. For unpredictable risk we use reactive strategy.
iv) The software risk can also be categorize depending on the characteristics or properties of the domain of a risk.
   1) **Project risk :** Project risk affect the project plan i.e. if project risk becomes real then it is likely that the project schedule will slip and the project cost will increase. Project risk identify the potential budget, schedule, staffing and organization, resource customer and their impact on the software project.

2) **Technical risk :** Technical risk affect the quality & performance of the software to be produced / developed. If technical risk becomes a reality then implementation may become difficult or say impossible. The examples of technical risks are the problem in potential design, the problem in implementation, the problem in interfacing, validation problem and maintenance issue.
3) **Business risk :** Business risk affect the business related issues of the software. The examples of business risk :
   i) Market risk – Building an excellent product or system that no one really wants.
   ii) Strategic risk – Building a product that no longer fits into the overall business strategies for the company.
   iii) Building a product that the sales department doesn't understand how to sell.
   iv) Losing the support of senior management due to change in focus or change in people (management risk).
   v) Losing the budget i.e. if we expect the x no. of users for the product in the market but in reality the no. of users are much less than x.

Q.3(d) What are the principles used for project scheduling and explain their meaning. [4]

(A) 1) Compartmentalization

The software : must be compartmentalize into a number of manageable activities and task.

2) Effort validation

Each and every project has a define number of people on the software team. As time allocation occurs, you must ensure that no more than the allocated number of people has been scheduled at any given time.

3) Inter-dependency

The inter-dependency of each every task must be determined before scheduling. Some task may occur in sequence while some are occur in parallel. Some tasks can not commence until the work product is produced by another task is available (such are called as inter-dependent task). So before preparing a schedule we need to identify all the inter-dependency amongst the task.

4) Defined milestones

Every task or group of task should be associated with a project milestone. A milestone is accomplished when one or more work products has been reviewed for quality and has been approved. In short milestones are quality outcomes or quality workable product after the completion of task.

5) Defined outcomes

Each and every task that is scheduled should have defined outcomes.

6) Defined responsibilities

Each and every task that is scheduled should be assigned to a specific team members.

Q.3(e) Explain project management spectrum 4P's. [4]

(A) Project management spectrum as 4P's

(i) People (ii) Product (iii) Process (iv) Project

People

- The factor people is so important that the SE institute (SEI) has developed a people management maturity model to enhance a people management maturity model to enhance the readiness of software organization.

- The organization that achieves high level maturity in people management have higher probability of implementing effective software product.
- The people included in software development are (hierarchy wise)

Stack Holder
- Senior Manager
- Project Manager
- Team Leader
- Software Programmer
- Customer

Product

- Before a product can be plant the following things needs to be concentrated
  a) Objective and scope of product should be established.
  b) Alternative solutions should be considered.
  c) Technical and management constraints should be identified.
- Before starting the development of the product the team tries to find answers of following :
  a) What are the basic requirements of a customer w.r.t. product?
  b) Should the product implemented with all the requirements?
  c) The development team can be able to implement the product?
- Without defining the objectives and scope and considering alternative solution and identifying the object the real product implementation is difficult.

Process

- The software process provides the framework from which an effective plan for software development can be established.
- A small number of framework activities called common process framework are applicable to all software immaterial of its size and complexity.
- However, while defining a process for the product the number of difficult tasks such as task, milestones, work product, deliverables and quality controls need to be considered.
- While selecting the best process model project management considers the facts :
  a) The customer and people who will do the work.
  b) The characteristics of product itself.
  c) The project environment in which software works.

Project

- We conduct planned and controlled software project development since it is the only one known way to manage complexity. Although the success rate of software project has improved so that the project can be effectively developed by the development team.
- **The reason why it fails**
  a) Unrealistic deadline of project
  b) Continuous changing customer requirement
  c) The effort applied to complete the task is not 10%.
  d) Predictable or unpredictable risk
  e) Technical difficulties
  f) Miscommunication amongst the team members and team leader.
- **How to avoid failure of project**
  A software manager and software engineers should perform following activities.
  a) Gather requirements from customer at very beginning stage.
  b) Understand the critical success factor that needs to good project management.
  c) Develop a commonsense approach for planning, monitoring, controlling software.

Q.4(a) Attempt any THREE of the following : [12]

Q.4(a) (i) What are different debugging strategies? Explain any one in brief. [4]

(A) i) Debugging and testing are two different activities. Debugging occurs as a consequence of testing i.e. initially testing is performed to identify the bugs which are uncovered in debugging process.

ii) The debugging strategies are :

1) Brute Force 2) Back Tracking 3) Cause Elimination

1) Brute Force :
   i) It is a most common but least efficient debugging strategy for isolating the cause of a software error.
   ii) This strategy is generally applied whenever everything else fails.
   iii) Using "let computer find bugs" philosophy is used in brute force which accept a large amount of inputs (trial and error) from user and try one inputs at a time and hope that somewhere in large amount of information we get a correct answer.
   iv) Information produced by computer is very large and that is waste of effort and time.

2) Back Tracking :
   i) It is fairly common and efficient method for isolating the cause of an error.
   ii) Beginning at the site where symptoms had occurred and the program is trace in the backward direction until the cause is found.
   iii) As the number of lines of a program increases, the number of back-tracking paths also increases which are impossible to manage.

3) Cause Elimination
   i) It is the least common but most efficient debugging strategy for isolating the cause of a software error.
   ii) This method is used directly to identify the prime cause of a bug and introduce the concept of binary portioning.
   iii) This method uses a "cause hypothesis" that can be proved using various method (cause hypothesis can be proved using inductive hypothesis).
   iv) Once the causes are identified they are eliminated one by one.
   v) Above three strategies are semi-automated because automated tools requires human intervention but debugging can also be possible by using automated tools without any human intervention called as automated debugging.

Q.4(a) (ii) Explain different approaches of integration testing. [4]

(A) Integration Testing

i) It is a systematic technique for construction of program structure while conducting test to uncover error associated with interfacing. The objective is to take unit tested module and to build a structure that is dictated by the design plan.

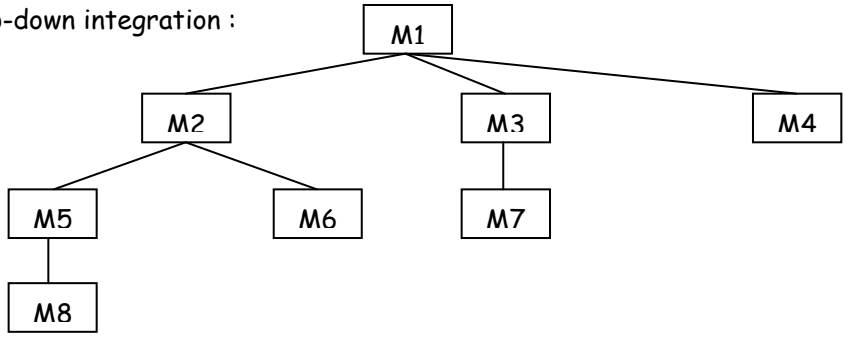
ii) There are 2 approaches used in integration testing
   a) Non-incremental integration
   b) Incremental integration

   In incremental integration, the program is tested in small modules where the errors are easier to locate and correct.

   In non-incremental approach all the modules are combined in advance and program is tested as a whole. This approach is difficult.

iii) There are 4 methods of integration testing
   1) Top-down integration 2) Bottom-up integration
   3) Regression testing 4) Smoke testing

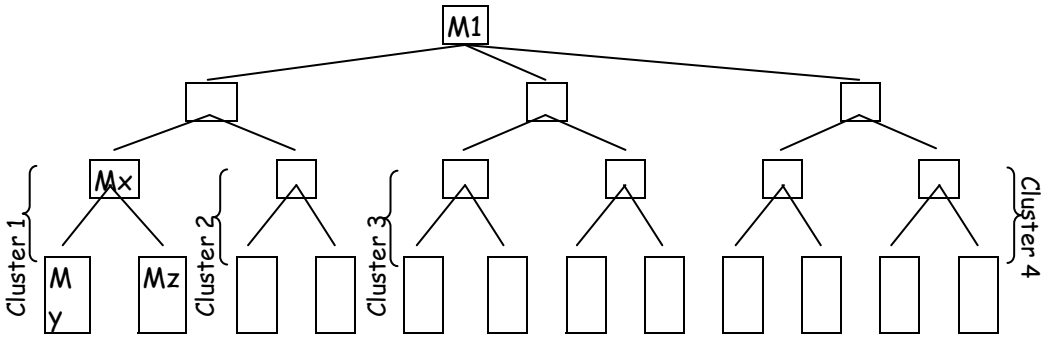
1) Top-down integration :



- As the name implies, modules are integrated by moving downwards through the control hierarchy beginning with the main control module.
- There are 2 approaches to perform top down integration :
  (a) Depth first : In this method modules are integrated by following the complete one sub-tree & then consider the other sub-trees. The possible depth first integration for the above structure are as follows :
  M1 M2 M5 M8 M6 M3 M7 M4
  M1 M4 M3 M7 M2 M6 M5 M5
  (b) Breadth first : In this method modules are integrated by moving the structure horizontally i.e. level-wise or breadth-wise. For the above structure breadth-first integration are as fill.
  M1 M2 M3 M4 M5 M6 M7 M8
  M1 M4 M3 M2 M7 M6 M5 M8
- Top down integration is performed in a series of 5 steps :
  (a) The main control module is used as a test driver and stubs are sub-divided all the components.
  (b) Depending on the integration approach (depth first or breadth first) stubs are replaced one at a time with actual component.
  (c) Test is conducted whenever module is integrated.
  (d) On completion of testing, other stub is replaced with real component.
  (e) Regression testing may be conducted ensure that no new errors have been generated.
- This testing verifies major control and decision points early in the design process.
- Top-down testing sounds relatively uncomplicated but in practice many problems may arises. Therefore 3 choices are used in top-down integration.
  (a) Delay many tests until stubs are replaced with actual modules.
  (b) Develop such a stub that performs limited functions
  (c) Integrate the software from bottom to upward direction.

2) Bottom up integration

- As the name implies modules are integrated from lower level modules and then moving in upward direction because modules are integrated from bottom up integration.
- Bottom up integration requires following steps.
  (a) Lower level modules are combined into clusters that performs specific network sub function.
  (b) For every cluster driver is returned to co-ordinate test cases input and output.
  (c) By using those test cases, cluster is tested.
  (d) Once the cluster is tested, driver is removed and combined by moving in upward direction.



In the above structure there are 4 clusters so we first combine the cluster and test them using the test cases derived by their drivers (parent).
Once the clusters are tested then integrate those clusters with their parents using bottom up integrations.

3) Regression Testing

- Regression testing is a type of integration testing which is used whenever there is a change in software, that means if some modules are newly added, some modules may get deleted or any structural change in software.
- Whenever a new module is added as a part of integration testing, the software may change, new data flow paths are established, new I/O may occur, new control logic may be generated. Therefore regression testing is performed.
- In the context of integration testing, regression testing is nothing but reexecution of some subset of test that have already conducted will be reconducted to ensure that new change does not affect the system performance.
- There are 2 ways to perform regress testing :
  (a) Manual testing : In this test are executed manual that requires more time.
  (b) Automatic : In this automated tools are used in which test cases are captured and the software is tested.
- Regression testing contains 3 sets of test cases.
  (a) A representative sample of test case that exercise all the functions.
  (b) Test cases are conducted on the software components that have been change.
  (c) Additional test are conducted on those functions which are affected by changes.
- As the integration testing proceeds the number of regression test can grow. Hence, it is not advisable to perform regression testing on new functions or on those functions which are affected by change.

4) Smoke Testing

- Smoke testing is an integration testing which is commonly used whenever software has to be developed.
- Smoke testing has following activities :
  (a) Software components which are translated into nodes are integrated to build a module component.
  (b) A series of test is conducted on that module to perform the intended function.
  (c) Once the entire software is integrated with all modules then the entire product is smoke tested.
- The benefit of smoke testing are as follows :
  (a) Integration risk is minimized because smoke testing is performed daily.
  (b) The quality of an end product is improved because smoke testing is performed to uncover both functional errors and components level design errors.
  (c) Progress of the software is easier to access and assess because smoke testing is performed on daily basis.

(d) Error diagnosis and correction are simplified because smoke testing is performed on each module and whenever error occurs then it is likely to be present in a new module.

- Smoke testing is generally performed after the stress testing because stress testing results into the system loopholes.

Q.4(a) (iii) With reference to software design give the meanings of **[4]**

(1) Modularity (2) Functional independence
(3) Refactoring (4) Information hiding

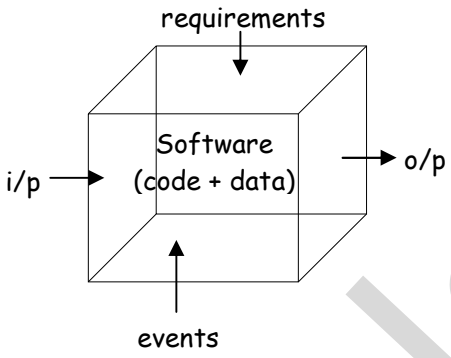
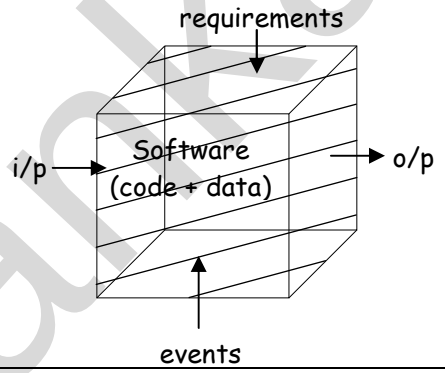
(A)

1. **Modularity :**
   Software architecture and design patterns embody modularity; that is, software is divided into separately named and addressable components, sometimes called modules that are integrated to satisfy problem requirements.
   It is the compartmentalization of data and function. It is easier to solve a complex problem when you break it into manageable pieces. "Divide-and-conquer".
   Don't over-modularize. The simplicity of each small module will be overshadowed by the complexity of integration "Cost".
2. **Functional Independence :**
   The concept of functional Independence is a direct outgrowth of modularity and the concepts of abstraction and information hiding.
   Design software such that each module addresses a specific sub-function of requirements and has a simple interface when viewed from other parts of the program structure.
   Functional independence is a key to good design, and to software quality. Independence is assessed using two qualitative criteria: cohesion and coupling.
3. **Refactoring :**
   It is a reorganization technique that simplifies the design of a component without changing its function or behavior. When software is re-factored, the existing design is examined for redundancy, unused design elements, inefficient or unnecessary algorithms, poorly constructed data structures, or any other design failures that can be corrected to yield a better design.
4. **Information Hiding :**
   Hiding implies that effective modularity can be achieved by defining a set of independent modules that communicate with one another only that information necessary to achieve software function.
   The use of Information Hiding as a design criterion for modular systems provides the greatest benefits when modifications are required during testing and later, during software maintenance. Because most data and procedures are hidden from other parts of the software, inadvertent errors introduced during modifications are less likely to propagate to other location within the software.

Q.4(a) (iv) Compare White-Box and Black-Box testing. **[4]**

(A)

	White Box Testing	Black Box Testing
1)	WBT is also called as open box, glass box, glass box & clear box testing	BBT is also called as closed box, functional & behavioral testing
2)	The code & data of a software are visible to the tester	The code & data of a software are invisible to the tester
3)	In WBT the software is tested line by line	In BBT the performance software is tested by applying the given set of test cases
4)	WBT is conducted by the developer of the software	BBT is conducted by the professional software tester group.

5)	High technical knowledge required	Technical knowledge not required
6)	WBT is used to identify errors in following categories : (a) Independent path (b) Logical conditions (c) Conditional loops	BBT is used to identify errors in following categories : (a) Missing functions (b) Interface error (c) Errors in data structure
7)	WBT is performed in the earlier stages of software development	BBT is performed in the later stages of software development i.e. after implementation
8)	WBT requires more time	BBT requires less time
9)	WBT is suitable for small scale programs	BBT is suitable for large scale programs or any software
10)	The types of WBT (a) Basis path testing (b) Control structure	The types of BBT are : (a) Graph based testing (b) Equivalence partitioning (c) Boundary value Analysis
11)	The scenario of WBT can be shown as : 	The scenario of BBT can be shown as : 

Q.4(b) Attempt any ONE of the following : [6]

Q.4(b) (i) Describe process of CMMI techniques. [6]

(A) CMMI (Capability Maturity Model Integration) is a proven industry framework to improve product quality and development efficiency for both hardware and software Sponsored by US Department of Defense in cooperation with Carnegie Mellon University and the Software Engineering Institute (SEI) Representation as :

- Staged
- Continuous

Level 0 : Incomplete the process area (e.g., requirements management) is either not performed or does not achieve all goals and objectives defined by the CMMI for level 1 capability for the process area.

Level 1 : Performed all of the specific goals of the process area (as defined by the CMMI) have been satisfied. Work tasks required to produce defined work products are being conducted.

Level 2 : **Managed** all capability level 1 criteria have been satisfied. In addition, all work associated with the process area conforms to an organizationally defined policy; all people doing the work have access to adequate resources to get the job done; stakeholders are actively involved in the process area as required; all work tasks and work products are monitored, controlled, and reviewed; and are evaluated for adherence to the process description.

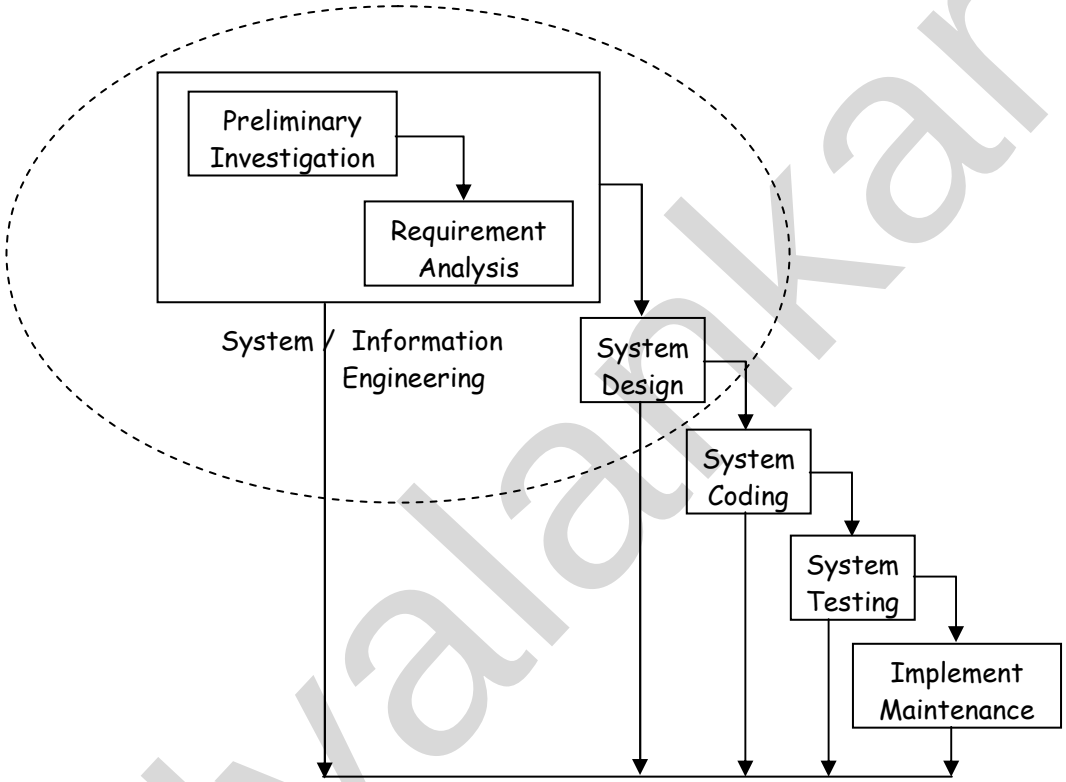
Level 3: Defined all capability level 2 criteria have been achieved. In addition, the process is tailored from the organization's set of standard processes according to the organization's tailoring guidelines, and contributes work products, measures, and other process-improvement information to the organizational process assets.

Level 4: Quantitatively managed all capability level 3 criteria have been achieved. In addition, the process area is controlled and improved using measurement and quantitative assessment. Quantitative objectives for quality and process performance are established and used as criteria in managing the process

Level 5: Optimized all capability level 4 criteria have been achieved. In addition, the process area is adapted and optimized using quantitative (statistical) means to meet changing customer needs and to continually improve the efficacy of the process area under consideration.

Q.4(b) (ii) Explain waterfall model with diagram. **[6]**

(A) Waterfall Model



i) Waterfall Model was developed by Winstone Royce.

ii) Waterfall Model suggest a systematic & sequential approach towards software development which begins with preliminary investigation, requirement analysis, system design, system coding, system testing & maintenance.

iii) The phases of waterfall model are :

(a) Preliminary Investigation : In this phase all the requirements or the software are gathered from customer or end-user or stake holder

(b) Requirement Analysis : To understand the nature of the programs or software to be built, the system analysis understand the information domain for the software as well as the required function, performance, interfacing & software as well as hardware requirements are analysized.

(c) System Design : In this phase all the requirements which are analysized from analysis phase gets converted into system flow or system design using flowchart, ER diagram, dataflow diagram (DFD).

(d) System coding : In this phase, the actual design is translated into a machine readable code i.e. software coding is done in this phase.

(e) System Testing : In this phase, the developed software is tested using valid & invalid test cases with the intention of finding errors or defects or faults present in the software.

(f) Implementation & Maintenance : This is an iterative process where the developed software is maintained throughout its process to give error-free performance. Some modifications & enhancement of the software can be done during the maintenance phase.

iv) Features of waterfall model
(a) Being one of the oldest model, therefore can be implemented for all size project.
(b) In this model the verification & testing is inherent in all the phases.

v) Advantages of waterfall model
(a) The most systematic & sequential approach towards software development.
(b) All the phases of waterfall model are clearly defined.

vi) Disadvantages of waterfall model :
(a) One of the time-tested / time-consuming model
(b) Most of the project or software rarely follow this approach.
(c) The time & cost is very high for every stage / phase of this model.
(d) Customer needs to specify all its requirements at the very beginning stage, which is often difficult for customer.

Q.5 Attempt any TWO of the following : [16]

Q.5(a) Explain core principles of software engineering. [8]

(A) David Hooker proposes seven core principal of software engineering which are require for software development.

i) KISS [Keep It Simple Stupid!]

- Before coding, all designs should be as simple as possible.
- Simple doesn't mean that the functionalities should be deleted rather it makes easily understood & maintain the system.
- Simple doesn't mean that it is quick or dirty rather it takes lot of thinking & large number of iterations.

ii) Reason It all exists :

- Software is developed for 1 reason i.e. to provide value to the customers.
- Before collecting system's requirements, before noting system's functionality, hardware or software requirements ask questions to yourself does it add a real value to the system ?" if the answer is no then don't do it otherwise start the development.

iii) Maintain the vision :

- A clear vision always leads to successful software development.
- If the software development team doesn't have a clear vision then software will fail & cannot be restarted from last failure point.
- Making the compromise in architectural vision, software becomes weak & finally break down.
- If the software compromise on requirement analysis then the software cannot be able to fulfill all the basic requirements.

iv) Whatever you produce other will consume :

- Software developer develop the software but it is used by the customer hence while developing a software think customer point of view.
- For any software there are large no. of customers therefore making their job simple is a key for software development.

v) Be open to the future :

- A software that has long life may provide a real value to the customer but in today's computing world software is changing & hardware becomes dependent.

- While developing a software & manufacturing software or hardware, always think future point of view, that means our developed software should be open for the future use which means software enhancement should be possible & reusability of components is also possible for developing a new software.

vi) Plan ahead for reuse :

- A software is divided into components that can be reused to develop new versions of software to minimize the time & cost.
- Object oriented technology uses the concept of inheritance in which existing functionality can be reused any number of times.
- There are many techniques of various levels for reusability but these levels should be properly understood by the software development team.

vii) Think :

- Placing a clear thinking always gives accuracy & gives the feeling that whatever you are doing is right.
- If you think & still software development fails then it's a good experience & start rethinking.
- If clear thinking has gone into the system a quality product will come out.

Q.5(b) Explain RMMM strategy in detail. [8]

(A) RMMM plan (Risk Mitigation, Monitoring, Management plan)

i) Risk management strategy can be included in the software project plan or the risk management step can be categorized into risk mitigation, risk monitoring and risk management plan.

ii) Risk Mitigation : It collects the steps to which try to avoid the chances of risk. It considers all the factors which can be used to minimize the chances of risk without affecting the working environment.

iii) Risk Monitoring :

a) Risk monitoring is a project tracking activity with 3 primary objectives
   1) To assess whether predicted risk do infact occur.
   2) To ensure that when the risk occurs then we have the proper steps to avoid the risk.
   3) To collect information that can be used further risk analysis.

b) In risk monitoring we design a step to avoid the risk such that the working environment may affect marginally and sometimes we may compromise on the performance.

Risk Management

This is required when the risk becomes reality. It defines the steps to be followed so that the system will be affected as less as possible. It also defines the alternate measures that can be taken so as to ensure running of the development process without much of the modification.

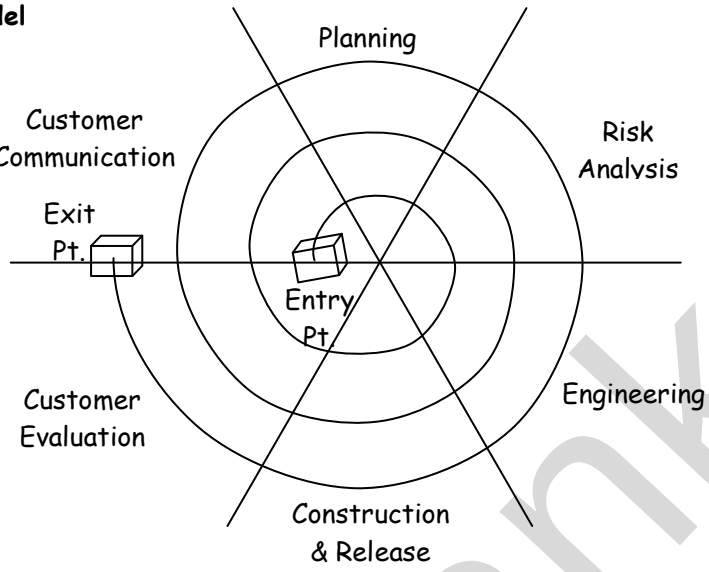
The RMMM plan documents all the work performed as a part of risk analysis and is used by project manager as a part of overall project plan. Some software team develop RMMM plan where each risk is documented individually using risk information sheet (RIS).

Risk Information Sheet (RIS)			
Risk-id : 101	Date : 4/6/2012	Probability : 60%	Impact : 1
Risk description : Funding will be lost that means no fund is available currently to complete the project.			
Risk Mitigation : i) Apply for the loan. ii) Try to avail funds from alternative resources.			

Risk Monitoring : i) Reduce the work-force. ii) Use local material iii) Increase the working hours iv) Cut down the expenses.	
Risk Management : i) Sell the shares. ii) Go for merging with other company.	
Current status : Now the risk is resolved & fund is available to complete the project.	
Originator : abc	Assigned : XYZ

Q.5(c) Explain spiral model with its advantages and disadvantages. **[8]**

(A) Spiral Model



i) Spiral model is one of evolutionary process model having most realistic approached to the development of large system.

ii) The spiral model is useful when the development of a project starts & it may possible that work may be stopped due to any reason, if such situation arises what steps must be taken to solve such problem should be known (i.e. because of risk analysis phase.)

iii) Spiral model is also useful in which path should be followed for a particular procedure must be known in advance to the development team because of this facility work will not be stopped in middle (i.e. because of planning phase.)

iv) The phases of spiral model are :
   a) Customer communication : In this phase effective communication is established between customer & developer to identify the requirement of a customer for the software under construction.
   b) Planning : It determines the objectives, alternatives & constraints of the portion of the product under consideration. In this phase, developer tries to prepare cost estimation & scheduling chart for the software.
   c) Risk Analysis : The alternatives are evaluated & potential risk areas are identified in this phase. Also determine the risk management strategy to resolve the risk or tries to minimize the impact of the risk.
   d) Engineering : By using the activities like simulation & prototyping the development & verification of the product is done in this phase.
   e) Construction & release : The actual implementation or system coding is done in this phase.
   f) Customer Evaluation : In this phase customer evaluates the working products & makes the suggestions & modifications.

v) Advantages of spiral model :
   a) Spiral model is a realistic approach for the development of large scale project or software product.
   b) Spiral model uses prototyping & evolutionary process for the software development.

vi) Disadvantages of spiral model :
   a) It is time consuming model because risk analysis & planning require more time as well as to perform planning & risk analysis we require expert persons
   b) As it is iterative it is never ending model.

Q.6 Attempt any FOUR of the following : [16]

Q.6(a) Explain system testing methods. [4]

(A)
- System testing is used to exercise computer based system, that means system testing is used when software gets validated and now it is ready to upload on a system (i.e. either on client machine or server machine.)
- There are various system testing methods and each one has distinct purpose.
  The types of or methods of system testing are :
  (a) Recovery testing (b) Security testing
  (c) Stress testing (d) Performance testing
- Recovery testing
  (a) Most of the computer system must require recovery from the defaults & resumes processing within a pre-specified time.
  (b) Some computer systems are "fault tolerant" that means processing the fault doesn't affect on the computer system.
  (c) Recovery testing is a type of system testing that force the software to fail in various ways and determine whether the system is recovered or not.
  (d) There are 2 ways to perform recover testing.
    1) Automated testing : In this re-initialization, check-point mechanism, data recovery & restart parameter must be evaluated.
    2) Manual testing : In this meantime to repair is evaluated to determine whether the fault is in acceptable limit or not.

- Security Testing
  (a) Many computer system contains sensitive data that should not be accessed by unauthorized members.
  (b) It verifies that protection mechanism built into a system protect it from hackers.
  (c) During the security testing a tester plays a role of hacker and try to apply all the test cases to hack the system.
  (d) Given enough time and resources, good security testing will ultimately penetrate the system.

- Stress Testing
  (a) These testing execute a system in a manner that demand abnormal quantity frequency or volume that means stress testing is basically used to calculate the stress of a system (the number of user request can be handled without failure).
  (b) The example of stress testing are
    1) Ten interrupt occurs where one interrupt per second occurs. There we calculate the number of interrupts that can be handled by system without failure.
    2) Test cases requires memory but it is currently unavailable. Here we test the availability of a memory space for an operation.
    3) Input data rates suddenly increases. Here we check how much data can system accept as an initial set of inputs.
  (c) A variation of stress testing is called as sensitive testing in which a small amount of data within the large amount of data causes errors. Hence, it is advisable to detect that faulty data.

- Performance testing
  (a) It is designed to test run-time performance of a software within the integrated system.

(b) It is performed in 2 ways :
1) Unit level : In this individual components of a computer system are tested.
2) Integration level : In this entire computer system is tested as a whole.

(c) This testing is generally coupled with stress testing and requires both software & hardware performance test.

Q.6(b) What are PSP and TSP framework activities? Explain their meaning. [4]

(A) Personal Software Process (PSP)

i) PSP is an SEI technology that brings discipline of individual software engineer, dramatically improving the process quality.

ii) PSP is a structural software development process that can help software engineer to understand & improve their performance by using disciplined & data driven procedure.

iii) The PSP provide software engineer with disciplined method for improving personal software development processes. The PSP helps software engineer to
1) Improve their estimating & planning skills
2) Make commitments they can keep
3) Manage the quality of their projects
4) Reduce the no. of defects in their work

Team Software Process (TSP)

i) In combination with PSP, TSP provides a defined operational process framework that is designed to help teams of managers & engineers to organize & produce large scale software projects.

ii) The TSP is intended to improve the levels of quality & productivity of a software development team in order to help them to develop a quality project.

iii) The objectives of TSP are :
1) To provide improvement guidance to high maturity organizations.
2) To show managers how to motivate & coach their teams & how to help them to sustain at peak performance.

Q.6(c) Explain SCM process repository. [4]

(A) i) Software configuration management process :
1) SCM process includes the five tasks
a) Identification
b) Version control
c) Change control
d) Configuration auditing
e) Status reporting

2) To control and manage software configurations each configuration items must be separately named and defined. There are 2 types of objects that we need to identify in identification, these are
a) Basic objects :
They are the text that has been created by software engineer during analysis, design coding and testing.
For e.g. : A part of requirement specification, source listing of modules, a documentation.
b) Aggregate object :
It is collection of basic objects and other aggregated objects.
For e.g. : Design specification, implementation part of a module.

3) Version control combines the procedures and tool to manage different versions of configuration objects that are created during software engineering process.

4) Change control combines human procedures and automated tools to provide a mechanism for the control of change. Although many changes request are submitted during the software maintenance phase, request for change can occur at any during the software process.

5) Configuration audit controls the entire configuration of the software by ensuring the change is properly implemented or not in configuration audit a formal technical review is conducted by assessment a configuration object for characteristics that are generally implemented in the change.
6) Status reporting is an SCM task that answers the following questions.
   i) What happened ? ii) Who did it ?
   iii) When did it happen ? iv) What else will be affected ?

ii) SCM repository
1) In the early days of software engineering, software configuration items were maintained as a paper documents placed in file folders or the storage area where all files are kept.
2) This is a problematic approach because
   a) Finding a configuration item when it was needed was difficult.
   b) Determining which items were changed when and by whom was often challenging
   c) Constructing a new version of an existing program was time consuming.
   d) Describing detail on complex relationship between configuration items was virtually impossible.
3) Therefore today's SCI maintains a repository or database where all the items are stored.
4) The SCM repository is a set of mechanism in data structure that allow software team to manage a change in an effective manner.
5) The SCM repository provides all the features of database and repository performs the following functions :
   a) Data Integrity b) Information Sharing c) Tools Integration
   d) Data Integration e) Document Standardization
6) A repository performs the following activities for software engineer.
   a) Integrate with or directly support process management function.
   b) It supports the interface with other software engineering tools.
   c) Accommodate storage of sophisticated date object.

Q.6(d) Write in brief on ISO 9000 quality standard. [4]

(A) The ISO 9000 quality standards

- I The International Organization for Standardization established the term ISO 9000, which refers to a set of quality management standards.
- The ISO 9000 standards are maintained by ISO and administered by accreditation and certification bodies.
- ISO's purpose is to facilitate international trade by providing a single set of standards that is recognized everywhere.
- The ISO 9000 standards require :
  1. A standard language for documenting quality procedures.
  2. Documented procedures, covering all parts of the organization within the scope of registration, for ensuring that quality objectives are met.
  3. A system to track and manage evidence that these practices are instituted throughout the organization and
  4. A third party auditing model to assess, certify and maintain certification of organizations.
- ISO 9000 standards have been adopted by many countries including all members of the European Community, Canada, Mexico, the Unite States, Australia, New Zealand and the Pacific Rim.
- A quality assurance system may be defined as the organizational structure, responsibilities, procedures, processes and resources for implementing quality management.
- ISO 9000 describes quality assurance elements in generic terms that can be applied to any business regardless of the products or services offered.

- Quality assurance systems are created to help organizations ensure their products and services satisfy customer expectations by meeting their specifications.
- These quality assurance systems cover a wide variety of activities encompassing a product's entire life cycle including planning, controlling, measuring, testing and reporting and improving quality levels throughout the development and manufacturing process.
- To become registered to one of the quality assurance system models contained in ISO 9000, a company's quality system and operations are scrutinized by third party auditors for compliance to the standard and for effective operation.
- After adopting the standard, a country typically permits only ISO registered companies to supply goods and services to government agencies and public utilities.
- Upon successful registration, a company is issued a certificate from a registration body represented by the auditors. The manufacturers of these products often require their suppliers to become registered.

Q.6(e) What is SRS? Explain importance of SRS. [4]

(A) A Software requirements specification (SRS), a requirements specification for a software system, is a description of the behavior of a system to be developed and may include a set of use cases that describe interactions the users will have with the software. In addition it also contains non-functional requirements. Non-functional requirements impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

Software requirements specification establishes the basis for agreement between customers and contractors or suppliers (in market-driven projects, these roles may be played by the marketing and development divisions) on what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules.

The software requirements specification document enlists enough and necessary requirements that are required for the project development. To derive the requirements we need to have clear and thorough understanding of the products to be developed or being developed. This is achieved and refined with detailed and continuous communications with the project team and customer till the completion of the software.

❑ ❑ ❑ ❑ ❑