Module 5

Maintenance, Risk Management and Project Management

MAINTENANCE

Ques 1) What is software maintenance? What is the need for software maintenance?

Or

Why is software maintenance required? Discuss with examples.

Ans: Software Maintenance

Software maintenance is a task that every development group has to face when the software is delivered to the customer's site, installed and is operational. Therefore, delivery or release of software inaugurates the maintenance phase of the life cycle.

Software maintenance is a very broad activity that includes error corrections, enhancements of capabilities, deletion of obsolete capabilities, and optimization. Because change is inevitable, mechanisms must be developed for evaluating, controlling and making modifications. So any work done to change the software after it is in operation is considered to be maintenance work. The purpose is to preserve the value of software over time. The value can be enhanced by expanding the customer base, meeting additional requirements, becoming easier to use, more efficient and employing newer technology.

Software maintenance is becoming an important activity of a large number of organizations. When the hardware platform changes and a software product perform some low-level functions, maintenance is necessary. Also, whenever the support environment of a software product changes, the software product requires re-work to cope with the newer interface.

Software maintenance is the activity associated with keeping operational computer system continuously in tune with the requirements of users and data processing operation. Software Maintenance Process is expensive and risky and is very challenging.

Need for Software Maintenance

Software maintenance is needed due to following **reasons**:

1) Control on software system functions, i.e., keeping it in tune for reliable and useful functioning.

2) Control system modifications, i.e., planning, execution, and control of modifications to maintain original software system integrity, reliability, and performance meeting original software quality goals.

3) Make changes (Not asked by the users) for achieving better quality of software system functioning, improving response, documentation, help and usefulness and so on.

4) Prevent software system decline on performance, and other tangible quality goals.

For example, a bank decides to offer a new mortgage product. This will have to be included in the system so that mortgage interest and payments can be calculated or the Government recently changed the VAT rate from x% to y%. This change meant that many organizations had to make alterations to their systems.

Examples of software maintenance include Re-organizing data sets within a database so they can be searched faster or use less storage or providing shortcuts commands that experts can use instead of the slower standard menu system.

Ques 2) What are the different phases of maintenance process? Draw the diagram also.

Ans: Maintenance Process

Once particular maintenance objective is established, the maintenance personnel must first understand what they are to modify. They must then modify the program to satisfy the maintenance objectives. After modification, they must ensure that the modification does not affect other portions of the program. Finally, they must test the program. These activities can be accomplished in the four phases as shown in **figure 5.1**.

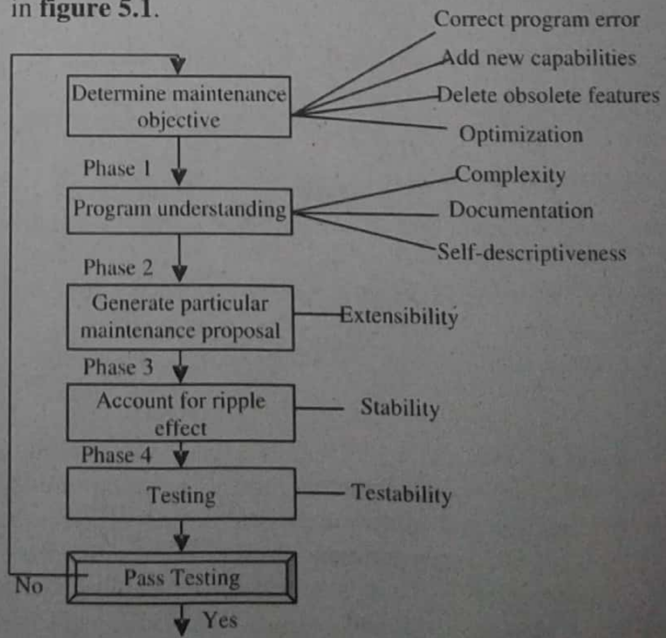

Figure 5.1: Software Maintenance Process

Ques 3) What are the different categories of software maintenance?

Or

Write short note on the following maintenance activities:
1) **Preventive Maintenance**
2) **Adaptive Maintenance**
3) **Corrective Maintenance**
4) **Perceptive Maintenance**

Ans: Categories of Maintenance
Categories of maintenance are as follows:
1) **Preventive Maintenance:** Corrective, adaptive and perfective changes lead to long term effects which increases the complexity of software and reflecting deteriorating structure. This is reduced or maintain by the maintenance activity. This maintenance is referred as preventive maintenance. Software does not wear away like hardware and requires no maintenance to retain functionality.

   Maintenance activity is initiated from the maintenance organisation so as to make programs easier to understand. This is done by the code restructuring, optimization, and updating documentation.

   Quick fixing of software can give rise to the complexity of its source code thereby justifying complete re-structuring of the code.

2) **Corrective Maintenance:** This type of maintenance is carried out to correct error in the new system. These errors may be of the following types:
   i) The first corrective maintenance is 'program logic errors'. Following are the some well-known logic errors:
      a) Poor Data Validation Procedures,
      b) Data Type Errors,
      c) Incorrect Usage of Variable Names, and
      d) Infinite Loop, etc.
   ii) Hardware problems such as disk crash, memory failure, etc. are the system errors.
   iii) When errors occur because of the operating staff member then it is known as 'operational errors'. **For example,** incorrect loading of tape file, improper back-up of computer files, forgetting to change the printer ribbons, etc. are the operational errors.
   iv) User errors result from lack of training and user manuals.

   All errors are rectified by the maintenance programmers by re-designing the error prone modules of the system. When, a user wish for a service request then adaptive maintenance begins. Then a committee assesses the request, approves it and then the maintenance programmers bring in to the new changes. All the changes are done by changing in the logical design, technical design, and revision of data dictionary to record new terms, if any, etc.

3) **Perceptive Maintenance:** When user's changes or want to add the new requirements to enhance the system then perceptive maintenance is taken into consideration. Software documentation is very important because it is very difficult for the user to make changes into the working software without the developer's approval. Software developed for the website has the highest possibility of change so it always has a copy of the documentation on its site.

4) **Adaptive Maintenance:** This includes changes to software to match up to the ever changing environment of business rules, government policies, work patterns, software and hardware operating platforms. These are external influences which when changed lead to software modification.

Ques 4) Which models are used for estimating cost of maintenance?

Or

Discuss the Boehm, Belady and Lehman Model with example.

Ans: Cost of Maintenance/Estimation of Maintenance Cost
Maintenance efforts constitute about 60% of the total life cycle cost for a typical software product. However, maintenance costs vary widely from one application domain to another. For embedded systems, the maintenance cost can be as much as 2 to 4 times the development cost.

Following two models are used for estimating maintenance cost:
1) **Boehm Model:** Boehm proposed a formula for estimating maintenance costs as part of his COCOMO cost estimation model. Boehm's maintenance cost estimation is made in terms of a quantity called **Annual Change Traffic (ACT)**.

   Boehm defined ACT as "the fraction of a software product's source instructions which undergo change during a typical year either through addition or deletion. Hence ACT is shown as:

$$ACT = \frac{KLOC_{added} + KLOC_{deleted}}{KLOC_{total}}$$

   Where, $KLOC_{added}$ is the total kilo lines of source code added during maintenance. $KLOC_{deleted}$ is the total $KLOC_{deleted}$ during maintenance. Thus, the code that is changed should be counted in both the code added and the code deleted.

   The annual change traffic (ACT) is multiplied with the total development cost to arrive at the maintenance cost:

   Annual Maintenance Effort (AME) = ACT × SDE.

Where,
SDE: Software development effort in person-months
and
ACT: Annual change traffic

Most maintenance cost estimation models, however, give only approximate results because they do not take into account several factors such as the experience level of engineers, and familiarity of engineers with the product, hardware requirements, software complexity, etc.

2) **Belady and Lehman Model:** This model indicates that the effort and cost can increase exponentially if poor software development approach is used and the person or group that used the approach is no longer available to perform maintenance. The basic equation is given below:

$$\mathbf{M = P + K\, e^{(c-d)}}$$

Where

M: Total effort expended.

P: Productive effort that involves analysis, design, coding, testing and evaluation.

K: An empirically determined constant.

c: Complexity measure due to lack of good design and documentation.

d: Degree to which maintenance team is familiar with the software.

In this relation, the value of 'c' is increased if the software system is developed without use of a software engineering process. 'c' will be higher for a large software product with a high degree of systematic structure than a small one with the same degree.

If the software is maintained without an understanding of the structure, function, and purpose of the software, then the value of 'd' will be low.

Ques 5) Annual Change Traffic (ACT) for a software system is 15% per year. The development effort is 600 PMs. Compute an estimate for Annual Maintenance Effort (AME). If life time of the project is 10 years, what is the total effort of the project?

Ans: The development effort = 600 PM
Annual Change Traffic (ACT) = 15%
Total duration for which effort is to be calculated = 10 years.

The maintenance effort is a fraction of development effort and is assumed to be constant.

$$AME = ACT \times SDE$$
$$= 0.15 \times 600 = 90 \text{ PM}$$

Maintenance effort for 10 years = 10 x 90 = 900 PM
Total effort = 600 + 900 = 1500 PM

Ques 6) The development effort for a software project is 500 person-months. The empirically determined constant (K) is 0.3. The complexity of the code is quite high and is equal to 8. Calculate the total effort expended (M) if:
1) **Maintenance team has good level of understanding of the project (d = 0.9)**
2) **Maintenance team has poor understanding of project (d = 0.1).**

Ans: Development effort (P) = 500 PM
K = 0.3
C = 8

Maintenance team has good level of understanding (d = 0.9)

$$M = P + K\, e^{(c-d)}$$
$$= 500 + 0.3\, e^{(8-0.9)}$$
$$= 500 + 363.59 = 863.59 \text{ PM}$$

Maintenance team has poor level of understanding (d = 0.1)

$$M = P + K\, e^{(c-d)}$$
$$= 500 + 0.3e^{(8-0.1)}$$
$$= 500 + 809.18 = 1309.18 \text{ PM}$$

Hence, it is clear that effort increases exponentially, if poor software engineering approaches are used and understandability of the project is poor.

RISK MANAGEMENT

Ques 7) What is risk management? What are the objectives of software risk management?
Or
What are the risks found in developing Software Projects?
Or
What is Risk management?

Ans: Software Risk Management
Risk management is the process of identifying, addressing, and eliminating these potential problems before they can damage our project. Software risk management is the process of identifying software risks and planning to avoid those risks or to minimize their effects if they cannot be avoided.

Risk Management is a software engineering practice with processes, methods, and tools for managing risks in a project. It provides a disciplines environment for proactive decision-making to:
1) Access continuously what can go wrong (risks).
2) Determine what risks are important to deal with.
3) Implement strategies to deal with those risks.

Objectives of Software Risk Management
The objectives of software risk management are to identify, address and eliminate software risk items before

they become threats to successor major sources of rework. Some objectives include:

1) **Identify Software Risks:** Identify potential problems and deal with them when it is easier and cheaper to do so (before they become crisis).
2) **Alignment with Objectives:** Focus on the project's objective and consciously look for things that may affect quality throughout the production process.
3) **Proactive Strategy:** Allow the early identification of potential problems (proactive strategy) and provide input into management decisions regarding resource allocation.
4) **Provides Staff:** Involve personnel at all levels of the projects focus their attention on a shared product vision, and provide a mechanism for achieving it.
5) **Enhances Software Success:** Increase the chances of project success.

Risks in Software Development Projects

Software development projects are subject to many risks, including:

1) Poorly defined requirements
2) Client requirements changes
3) Poor techniques for cost estimation
4) Rapid technological changes
5) Dependence on skills of individual developers
6) Extreme mobility of developers.

Ques 8) What is software risk? Classify them.

Or

What are the different software risks?

Ans: Software Risk

According to Webster, "Risk is the possibility of suffering loss".

Risk in a project or program is a measure of the ability to achieve objectives within cost, schedule and constraints. In a development project, the loss describes the impact to the project which could be in the form of diminished quality of the end product, increased costs, delayed completion, or failure.

Risk involves the following two characteristics:

1) **Uncertainty:** The risk may or may not happen; that is, there are no 100% probable risks.
2) **Loss:** If the risk becomes a reality, unwanted consequences or losses will occur.

When risks are analysed, it is important to quantify the level of uncertainty and the degree of loss associated with each risk.

Types of Software Risks

Software risk can be classified as below:

1) **Classification I**
   i) **Project Risks:** Project risks threaten the project plan. That is, if project risks become real, it is likely that project schedule will slip and that costs will increase. Project risks identify potential budgetary, schedule, personnel (staffing and organization), resource, customer, and requirements problems and their impact on a software project. Project complexity, size, and the degree of structural uncertainty were also defined as project (and estimation) risk factors.

   ii) Technical Risks: Technical risks threaten the quality and timeliness of the software to be produced. If a technical risk becomes a reality, implementation may become difficult or impossible.

   Technical risks identify potential design, implementation, interface, verification, and maintenance problems. In addition, specification ambiguity, technical uncertainty, technical obsolescence, and "leading-edge" technology are also risk factors. Technical risks occur because the problem is harder to solve than we thought it would be.

   iii) **Business Risks:** Business risks threaten the viability of the software to be built. Business risks often jeopardize the project or the product. The top five business risks are:
      a) Building an excellent product or system that no one really wants (market risk),
      b) Building a product that no longer fits into the overall business strategy for the company (strategic risk),
      c) Building a product that the sales force doesn't understand how to sell
      d) Losing the support of senior management due to a change in focus or a change in people (management risk),
      e) Losing budgetary or personnel commitment (budget risks).

2) **Classification II**
   i) **Known Risks:** Known risks are those that can be uncovered after careful evaluation of the project plan, the business and technical environment in which the project is being developed, and other reliable information sources (**for example,** unrealistic delivery date, lack of documented requirements or software scope, poor development environment).
   ii) **Predictable Risks:** Predictable risks are extrapolated from past project experience (**for example,** staff turnover, poor communication with the customer, dilution of staff effort as ongoing maintenance requests are serviced).
   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.

Ques 9) How are project risks different from technical risks?

Ans: Difference between Project Risks and Technical Risks

Table below shows the difference between project risks and technical risks:

Table 1: Project Risks *vs.* Technical Risks

Project Risks	Technical Risks
Project risks are those that occur during the execution or building of the project.	Technical risks are risks that occur during the operation of a project after it is completed.
Project risks threaten the project plan.	Technical risks threaten the quality and timeliness of the software to be produced.
Generally groups such as construction, purchasing, finance, and human resources are responsible for managing project risks.	This type of risk should be identified during the design of the project and mitigated by modifying the design.
Project risks identify potential budgetary, schedule, personnel (staffing and organization), resource, customer, and requirements problems and their impact on a software project.	Technical risks identify potential design, implementation, interface, verification, and maintenance problems.
For example, suppose a piece of equipment such as a specialized filter needed for a smelter can only be purchased from a limited number of suppliers. If there is high demand for this product and none of the suppliers can supply it on time, the timelines for completing the project will be threatened.	**For example,** a tank is designed and built-in such as way as to create a risk of overflowing and spilling into a river during the operation of the plant.

Ques 10) What are the seven risk principles?

Ans: Risk Principles

There are seven principles, which provide a framework for effective risk management:

1) Global perspective
2) Forward-looking view
3) Open communications
4) Integrated management
5) Continuous process
6) Shared product vision
7) Teamwork

These principles are embodied within risk management products and services which addresses the need to establish a baseline set of risks in a project or program (Software Risk Evaluation), the need to create and implement a continuous processes for the effective management of risk (Continuous Risk Management), and the need to include all parts of the program (contractors, customers, etc.) in the joint management of risks (Team Risk Management).

Ques 11) What are the risk strategies?

Ans: Risk Strategies

1) **Reactive Risk Strategies:** A reactive strategy monitors the project for likely risks. Resources are set aside to deal with them, should they become actual problems. More commonly, the software team does nothing about risks until something goes wrong. Then, the team flies into action in an attempt to correct the problem rapidly. This is often called a fire-fighting mode. When this fails, "crisis management" takes over and the project is in real jeopardy.

2) **Proactive Risk Strategies:** A proactive strategy begins long before technical work is initiated. Potential risks are identified, their probability and impact are assessed, and they are ranked by importance. Then, the software team establishes a plan for managing risk. The primary objective is to avoid risk, but because not all risks can be avoided, the team works to develop a contingency plan that will enable it to respond in a controlled and effective manner.

Ques 12) Discuss the risk management Activities in detail.

Or

What is risk assessment and risk control?

Or

Discuss the risk management process in detail.

Or

What is risk identification? Also discuss about the risk monitoring?

Ans: Risk Management Process/Activities

Major risk management activities include identify, analyse, plan, track and control risks. These activities serve as the foundation for the application of continuous risk management. Each risk normally goes through these activities sequentially. Risk management activities can be listed as under:

1) **Risk Assessment:** Risk assessment is the determination of quantitative or qualitative value of risk related to a concrete situation and a recognized threat (also called **hazard**).

   Risk assessment includes the following activities:

   i) **Risk Identification:** Risk identification is a systematic attempt to specify threats to the project plan (estimates, schedule, resource, etc.) By identifying known and predictable risks, the project manager takes a first step toward avoiding them when possible and controlling them when necessary.

Activities in Risk Identification Phase

Several activities that occur within identification phase are:

a) **Identify Risks:** Identify risks using checklists, interviews, brain storming meetings, reviews and survey.

b) **Define Risk Attributes:** After risks are identified, they are evaluated with the criteria like likelihood of occurrence, consequence and time frame for action. These values of initial estimations, which are, analysed more in next phase.

c) **Document:** The risks are then documented. Along with the name of the risks, a risk statement and context are to be specified.

d) **Communicate:** Spreading the knowledge to all the project members.

ii) **Risk Analysis:** After the risks have been identified, all items are analysed using different criteria's. The purpose of risk analysis is to assess the loss probability and magnitude of each risk item.

Activities of Risk Analysis

a) **Group Similar Risks:** Detect duplicates and find new risk items by grouping the identified risks into categories.

b) **Determine Risk Drivers:** The risk drivers are parameters that affect the identified risk.

c) **Determine Sources of Risks:** The sources of risks are determined by asking the question 'why?' and figure out the root cause of risk.

d) **Estimate Risk Exposure:** The risk exposure is a measure of the probability and the consequence of a risk item.

e) **Evaluate against Criteria:** Each risk item is evaluated using the predefined criteria, which are important for the specific project. Criteria's can be stated in terms of probability of occurrence, the consequence and the time frame.

iii) **Risk Prioritization:** Risk prioritization helps the project focus on its most severe risks by assessing the risk exposure. Exposure is the product of the probability of incurring a loss due to the risk and the potential magnitude of that loss.

Let (r) is the likelihood of a risk occurring.
(s) is the consequence of the problems associated with that risk.

Thus, based on r and s, the priority of each risk (P) can be calculated as:

$$P = r * s;$$

The higher the exposure, the more aggressively the risk should be tackled. It is easy to understand the probability and impact, if these factors are measured on a three point scale as low, medium and high.

2) **Risk Control:** Finally, a plan is developed to manage those risks with high probability and high impact. Risk control is the process of managing risks to achieve the desired outcomes. Risk control process involves the following activities:

i) **Risk Management Planning:** Risk management planning produces a plan for dealing with each significant risk, including mitigation approaches, actions and timelines. Risk planning is to identify strategies to deal with risk.

Strategies in Risk Management Planning

a) **Risk Avoidance:** If a software team adopts a proactive approach to risk, avoidance is always the best strategy. This is achieved by developing a plan for risk mitigation.

b) **Risk Monitoring:** As the project proceeds, risk monitoring activities commence. The project manager monitors factors that may provide an indication of whether the risk is becoming more or less likely. In the case of high staff turnover, the following **factors** can be monitored:

- General attitude of team members based on project pressures.
- The degree to which the team has jelled.
- Interpersonal relationships among team members.
- Potential problems with compensation and benefits.
- The availability of jobs within the company and outside it.

In addition to monitoring these factors, the project manager should monitor the effectiveness of risk mitigation steps.

c) **Risk Management and Contingency Planning:** It assumes that mitigation efforts have failed and that the risk has become a reality. Risk contingency plans are preparations for dealing with a risk, which has occurred.

ii) **Risk Resolution:** When a risk has occurred, it needs to be solved. Risk resolution is the execution of the plans for dealing with each risk. If the risk has triggered, the project manager need to execute the action plan. The project manager also reports the progress against the plan and corrects the deviation.

Outputs of Risk Resolution Phase

a) **Risk status:** Risk status is the progress of risk management

b) **Acceptable risks:** Acceptable risks are the ones that are not to be solved.

c) **Reduced rework:** Reduced rework is a measure of the benefit of using risk management.

d) **Corrective actions:** Corrective actions are procedures that are known solutions if a problem occur and are generally accepted within the project or organization.

e) **Problem prevention:** Problem prevention occurs when trying to avoid problems and thereby eliminating their result.

iii) **Risk Monitoring:** Risk Monitoring is the continually reassessing of risks as the project proceeds and conditions change.

For example, successful completion of beta testing means that the risk of the client organization rejecting the system is minimal, while large turnover in development staff usually increases project and product risks. A risk management strategy can be included in the software project plan or the risk management steps can be organized into a separate Risk Mitigation, Monitoring and Management Plan.

The Risk Mitigation, Monitoring and Management (RMMM) plan documents all work performed as part of risk analysis and is used by the project manager as part of the overall project plan.

Some software teams do not develop a formal RMMM document. Rather, each risk is documented individually using a risk information sheet (RIS). In most cases, the RIS is maintained using a database system, so that creation and information entry, priority ordering, searches, and other analysis may be accomplished easily.

Once RMMM has been documented and the project has begun, risk mitigation and monitoring steps commence. Risk mitigation is a problem avoidance activity. Risk monitoring is a project tracking activity with three primary objectives:

a) To assess whether predicted risks do, in fact, occur;

b) To ensure that risk aversion steps defined for the risk are being properly applied; and

c) To collect information that can be used for future risk analysis.

In many cases, the problems that occur during a project can be traced to more than one risk. Another job of risk monitoring is to attempt to locate the source (what risk(s) caused which problems throughout the project).

SOFTWARE PROJECT MANAGEMENT (SPM)

Ques 13) What is the software project management? What is the responsibility of the project manager in software project management?

Ans: Software Project Management

The process of planning, organising, staffing, monitoring, controlling lead to a software project is known as Software Project Management (SPM).

There must be a manager in every software project who performs the task of communicating with the senior management, suppliers and initiators and leads the development team.

Responsibilities of the Project Manager

Responsibilities of the project manager as given below:

1) Production of the Software Project Management Plan (SPMP).
2) Allocating staff after defining the various organisational roles.
3) Informing the staff about their part in the plan for easy control of the project.
4) Motivates the team members to perform well and makes major decisions to lead the project.
5) Measures the progress of the project by monitoring.
6) Communicates progress reports to senior managers and initiators.

Software Project Management deals with software projects and is concerned with their planning, monitoring and controlling. It is a sub-discipline of Project Management.

Ques 14) What are the different activities of software project management?

Ans: Activities of Software Project Management

1) **Feasibility Study:** This deals with examining the validity of the business case of the project, i.e., whether it is worthy of getting started. Information regarding the requirements of the proposed project is gathered.

2) **Planning:** Once the results of the feasibility study indicate that the project is viable, the planning of the project is started. In case of large projects, all the planning is not done at the beginning.

3) **Project Execution:** In the execution phase the project is designed and implemented. Design involves thinking and making decisions about the features and form of the product that needs to be created. In case of software products it may relate to the internal architecture and external appearance or user interface of the software.

4) **Requirement Analysis:** This phase is concerned with finding what the potential users, managers and

employers are expecting as qualities and features of the proposed system. A holistic view of the system is taken into consideration. **For example,** one of the quality requirements might be that a transaction is completed within a certain time period. The ability of the system developers to design the software is one of the technical requirements.

5) **Architecture Design:** In this phase the requirements are mapped to the components of the proposed system. The users will be identified and what process in the new system can be computerized is decided at the system level.

6) **Detailed Design:** The software consists of a number of sub-modules that can be independently coded and tested. Thus, the units are separately designed.

7) **Code and Test:** Coding refers to writing the instructions contained in each module with the help of programming languages such as Java or C# or with the help of an application-builder such as Microsoft Access. The individual units can be tested to remove the bugs during this phase.

8) **Integration:** In this phase the joining of the individual units is done to verify if they conform to the overall requirements. The integration could be carried out at various levels. It might be implemented at the system level where the software and other components of the system like the network and hardware are brought together.

9) **Qualification Testing:** A careful testing of the components of the software is carried out in order to ensure that the requirements have been achieved.

10) **Installation:** The process of bringing the new system to operation is known as installation. Activities such as setting up standing data are also included.

11) **Acceptance Support:** This phase is concerned with the solving the problems that the new system shows. This includes the incorporation of any extended features and improvements and removal of bugs in the system.

Ques 15) What are the management concepts?

Or

Discuss about People, product, process and project in detail.

Ans: Management Spectrum (4 P's Concept)

Pressman proposes that in order to become effective a software project management should focus on 4Ps which are given below:

1) **People:** This factor is considered to be to be the most important one. This is evident from the fact that the Software Engineering Institute has developed the People Management Capability Maturity Model (P-CMM), "to enhance the readiness of software organisations to undertake development of increasingly complex applications by helping to attract, grow, motivate, deploy, and retain the talent needed to improve their software development capability".

Those organisations that have been able to achieve high levels of maturity in the management of the people have been able to manage the software projects in a much more efficient manner.

Following are the some people:

i) **Stakeholder:** There are five categories of people who take part in a software project:

a) **Senior Manager:** Carries out the task of defining those business issues that have notable influence on the software project.

b) **Project (Technical) Managers:** Involved with planning, motivating, organising and controlling the staff associated with the software work.

c) **Practitioners:** These people deliver the necessary technical skills that are required to design the application or product.

d) **Customers:** These people are stakeholders who have peripheral interest in the software product and specify the requirements.

e) **End-Users:** Once the software is implemented and released for production use they interact with it.

ii) **Team Leaders:** As the project management activity is people-intensive in nature, the people with adequate management capabilities become the team leader.

iii) **Software Team:** The number of human organisational structures is almost equal to the software development organisations. The team structure can be attributed to be 'best' when it is formed taking into consideration the difficulty of the problem, skill level of the people who will make up the team and the management style of the organisation.

A high performance team can be formed keeping in mind:

a) The team members must trust each other.

b) The skills must be distributed in proportion to the problem.

c) In order to maintain the cohesiveness of the team, mavericks must be excluded.

2) **Product:** In order to prepare a good project it is necessary to get a complete and reasonably accurate definition of the product. The various elements of the product that needs to be specified are as follows:

i) Objective and scope,

ii) Functional specifications,

iii) Performance levels, and

iv) Technical and management constraints.

In order to define the scope and objectives of the product the customer and the software developer must meet.

i) **Objectives:** These are associated with the identification of the overall goals of the product without considering the means of achieving them.

ii) **Scope:** It is an attempt to bind the characteristics of the software product quantitatively by identifying the behaviour, functions, primary data and characteristics of the system.

The alternative solutions are taken into consideration once the product scope and objectives are understood. Using these alternative solutions, users and managers are able to get the 'best' approach including the various constraints such as technical interfaces, budgetary restrictions, personnel availability, delivery deadlines and other factors.

3) **Process:** The structure of the software development project is determined by the development process model that is chosen. Typically a process includes three types of activities as listed below:

i) Activities common to all projects also known as the core activities,

ii) SQA, SCM, etc., popularly known as umbrella activities, and

iii) Activities that are project specific in nature.

With the help of the software process it is possible to establish a comprehensive plan for the software development. Thus, it is the task of the project manager to decide the most appropriate process model keeping in mind:

i) The people who will develop the product and the customers who have requested for the product.

ii) The inherent characteristics of the product.

iii) The environment where the team members of the project works.

4) **Project:** If the project has to be successfully completed then it is necessary to create good project plan. It not only helps in the monitoring and controlling the project but also serves as a guide for executing the project.

Though there has been a significant amount of improvement in the rate of failure of software projects still it remains higher than the expectation. So, it is necessary for the project managers and the software engineers to understand the critical factors that lead to success and identify the general warnings. They must develop a rational approach to plan, monitor and control the project. Thus, it is necessary to identify those facets of the software project that might go wrong and how to make them right to ensure the success of the project.

An information system project is at risk can be understood by looking at the following ten signs:

i) The development team fails to understand the needs of the customers.

ii) Ill-definition of the product scope.

iii) Changes in the chosen technology.

iv) Needs of the business change.

v) Unrealistic deadlines.

vi) Loss in the sponsorship.

vii) Resistance from the user's side.

viii) There is lack of people possessing appropriate skill in the project team.

ix) Poor management of the changes.

x) The best practices are avoided by the managers and practitioners.