1) Explain management spectrum or explain 4 p's of software system.

Effective software project management focuses on the four P's: people, product, process, and project.

The People

- People factor is very much important in the process of software development.
- There are following areas for software people like, recruiting, selection, performance management, training, compensation, career development, organization and work design, and team/culture development.
- Organizations achieve high levels of maturity in the people management area.

The Product

- Before a project can be planned, product objectives and scope should be established, alternative solutions should be considered and technical and management constraints should be identified.
- Without this information, it is impossible to define reasonable estimates of the cost, an effective assessment of risk, a realistic breakdown of project tasks, or a manageable project schedule.
- Objectives identify the overall goals for the product without considering how these goals will be achieved.
- Scope identifies the primary data, functions and behaviours that characterize the product.
- Once the product objectives and scope are understood, alternative solutions are considered. From the available various alternatives, managers and practitioners select a "best" approach.

The Process

- A software process provides the framework from which a comprehensive plan for software development can be established.
- A small number of frame-work activities are applicable to all software projects, regardless of their size or complexity.
- A number of different tasks, milestones, work products and quality assurance points enable the framework activities to be adapted to the characteristics of the software project and the requirements of the project team.
- Finally, umbrella activities such as software quality assurance, software configuration management, and measurement overlay the process model.

The Project

- We conduct planned and controlled software projects for one primary reason it is the only known way to manage complexity.
- A software project manager and the software engineers who build the product must avoid a set of common warning signs, understand the critical success factors that lead to good project management, and develop a common sense approach for planning, monitoring and controlling the project.

2) Explain various roles and responsibility of people. Or define four ps for project management and explain them in detail.

- People are the most important element for the success of software project. People participate in the project with different roles and responsibilities.

- People participate in the project with different roles and responsibilities.Let us consider these The Players
- The software process (and every software project) is populated by players who can be categorized into one of five constituencies:

1. Senior managers who define the business issues that often have significant influence on the project.
2. Project (technical) managers who must plan, motivate, organize, and control the practitioners who do software work.
3. Practitioners who deliver the technical skills that are necessary to engineer a product or application.
4. Customers who specify the requirements for the software to be engineered and other stakeholders who have a peripheral interest in the outcome.
5. End users who interact with the software once it is released for production use.

Team Leaders
There are some characteristics of team leader.
1. Motivation The ability to encourage technical people to produce to their best ability.
2. Organization The ability to mold existing processes that will enable the initial concept to be translated into a final product.
3. Ideas or innovation The ability to encourage people to create and feel creative even when they must work within bounds established for a particular software product or application.

Effective project manager emphasizes on 4 different key points:

Problem solving
An effective software project manager can diagnose thetechnical and organizational issues that are most relevant, systematically structure a solution or properly motivate other practitioners to develop the solution.

Managerial identity
A good project manager must take charge of the project. She must have the confidence to assume control when necessary and the assurance to allow good technical people to follow their instincts.

Achievement
To optimize the productivity of a project team, a manager mustreward initiative and accomplishment and demonstrate through his own actionsthat controlled risk taking will not be punished.

Influence and team building
An effective project manager must be able to“read” people; she must be able to understand verbal and nonverbal signalsand react to the needs of the people sending these signals.

Software Team
The best team structure depends on 3 things
- Management style of organization
- Number of people involved in the project and their skill levels.
- Overall problem difficulty
- Following factors to be considered for planning the structure of software engineering teams
  The difficulty of the problem to be solved.

The size of the software to be considered.
The time that the team will stay together.
The flexibility of delivery date.
The degree of communication required for the project.

- There are 4 organizational paradigms for software engineering teams

1. Closed Paradigm

It represents traditional hierarchy of team. The team work well for producing software based on past efforts but it fails to work for innovative ideas.

2. Random Paradigm

This is a loosely structured and depends upon individual initiative of team members. This performs well when innovations are required but performs poorly when orderly performance is required.

3. Open Paradigm

It attempts to structure a team in such a way that some controls are achieved using closed paradigm. But innovations are required using the random paradigm.

4. Synchronous Paradigm

It structures the team using natural compartmentalization of the problem. This team work on the piece of problem by establishing proper communication among them.

Agile team

- Small motivated team is called agile team. Group collaboration is critical success for the team. Agile teams are self-organizing team. There is no fixed team paradigm that can be applied to the agile, instead of that the agile team can use elements of random, open, closed and synchronous paradigms.
- Many agile models automate the project management and technical decisions required for project accomplishment.
- Agile team is allowed to select its own approach for software development. The only condition is that the business requirements and organizational standards must get satisfied during software development.

Coordination and Communication Issues

1) Scale

The scale of development efforts may be very large which leads to complexity and confusion in coordinating the team members.

2) Uncertainty

Continuous changes occur in the project, due to which uncertainty is common in modern software.

3) Interoperability

It is the key characteristics of many systems. New software must be able to communicate with existing software.

3) Explain software process with framework activities. Also explain process decomposition.

The generic phases that characterize the software processdefinition, development, and support are

applicable to all software.

- The problem is to select the processmodel that is appropriate for the software to be engineered by a project team.
- The project manager must decide which process model is most appropriate for the customers who have requested the product and the people who will do the work, the characteristics of the product itself, the project environment in which the software team works.
- When a process model has been selected, the team thendefines a preliminary project plan based on the set of common process framework activities.

Melding the Product and the Process

Project planning begins with the melding of the product and the process. Each function to be engineered by the software team must pass through the set of frameworkactivities that have been defined for a software organization. Assume that the organization has adopted the following set of framework activities:

Customer communication—tasks required to establish effective requirementselicitation between developer and customer.
Planning—tasks required to define resources, timelines, and other project-related information.
Risk analysis—tasks required to assess both technical and management risks.
Engineering—tasks required to build one or more representations of theapplication.
Construction and release—tasks required to construct, test, install, and pro-vide user support.
Customer evaluation—tasks required to obtain customer feedback based onevaluation of the software representations created during the engineering activity and implemented during the construction activity.

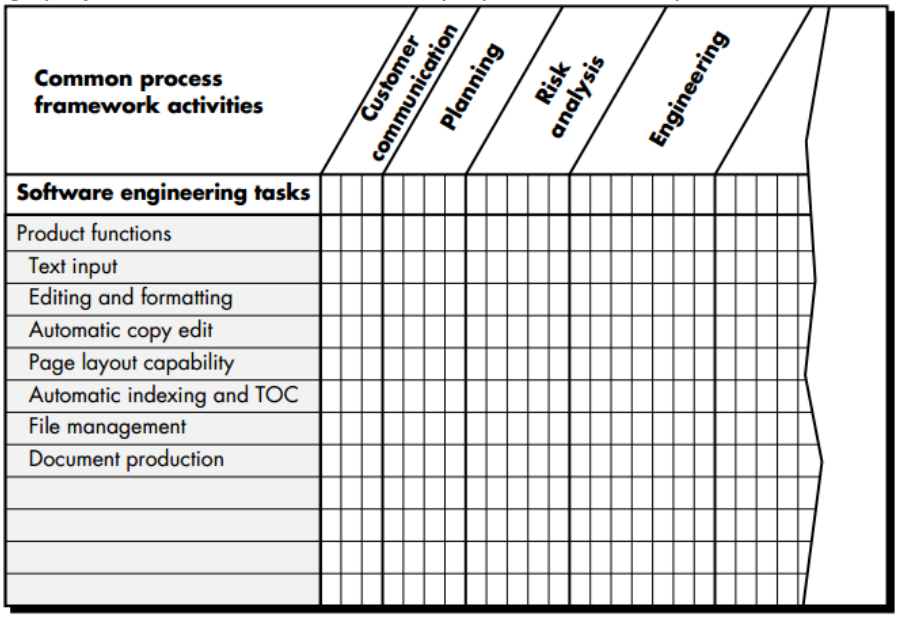
- The team members who work on a product function will apply each of the frame-work activities to it.
- In essence, a matrix similar to the one shown in Figure iscreated. Each major product function is listed in the left-hand column.
- Frameworkactivities are listed in the top row. Software engineering work taskswould be entered in the following row.
- The job of the project manager is to estimate resource requirements for each matrixcell, start and end dates for the tasks associated with each cell, and work products to be produced as a consequence of each task.

Process Decomposition

- The software development team as a greater flexibility for choosing the software process model which is best suitable for the project.
- After decision of process model the software engineering tasks are decided.
- When a project manager raises a question “How do we accomplish framework activities?” then project decomposition commences. For example a small project might require following work tasks for performing the communication activity-

1) A list of clarification issues is to be prepared.
2) For addressing the clarification issues conduct meetings with customers.
3) The developer and customer together must prepare statement of scope.
4) Review the statement of the scope.
5) Perform modifications in the statement of the scope if required.

Similarly for the large projects the work task list can be prepared which may contain some additional tasks.



Melding the Problem and the Process

4) Explain W^{5}HH principle.

Boehm suggests an approach that addresses project objectives, mile-stones and schedules, responsibilities, management and technical approaches, and required resources.

- He calls it the WWWWWHH principle, after a series of questions that lead to a definition of key project characteristics and the resultant project plan:

- **Why is the system being developed?** The answer to this question enables all parties to assess the validity of business reasons for the software work. Stated in another way, does the business purpose justify the expenditure of people, time, and money?
- **What will be done?** The answers to these questions help the team to establish a project schedule by identifying key project tasks and the milestones that are required by the customer.
- **When will be done?** Answer to this question will help to prepare the project schedule with identified project task and milestones.
- **Who is responsible for a function?** Earlier in this chapter, we noted that the role and responsibility of each member of the software team must be defined. The answer to this question helps accomplish this.
- **Where they are organizationally located?** Not all roles and responsibilities reside within the software team itself. The customer, users, and other stakeholders also have responsibilities.
- **How will the job be done technically and managerially?** Once product scope is established, a management and technical strategy for the project must be defined.
- **How much of each resource is needed?** The answer to this question is derived by developing estimates based on answers to earlier questions.
- Boehm's W5HH principle is applicable regardless of the size or complexity of a software project. The questions noted provide an excellent planning outline for the project manager and the software team.

5) Explain symptoms to indicate why software project fail. Also explain point to overcome the problem in software project.

- In order to manage a successful software project, we must understand what can go wrong and how to do it right.
- In an excellent paper on software projects, John Reel defines ten signs that indicate why the software project fails.

1. Software people don't understand their customer's needs.
2. The product scope is poorly defined.
3. Changes are managed poorly.
4. The chosen technology changes.
5. Business needs change.
6. Deadlines are unrealistic.
7. Users are resistant.
8. Sponsorship is lost.
9. The project team lacks people with appropriate skills.
10. Managers avoid best practices and lessons learned.

- Reel suggests a five points to overcome the problems in the software projects.

1) Start on the right foot.
This is accomplished by working hard to understand the problem that is to be solved and then setting realistic objects and expectations for everyone who will be involved in the project.
It is reinforced by building the right team and giving the team the autonomy, authority, and technology needed to do the job.

2) Maintain momentum.
Many projects get off to a good start and then slowly disintegrate.
The project manager must provide reasons to keep turnover of personnel to an absolute minimum, the team should emphasize quality in every task it performs, and senior management should do everything possible to stay out of the team's way.

3) Track progress.
For a software project, progress is tracked as work products are produced and approved as part of a quality assurance activity.
In addition, software process and project measures can be collected and used to assess progress against averages developed for the software development organization.

4) Make smart decisions.
In essence, the decisions of the project managerand the software team should be to "keep it simple."
Whenever possible decide to use commercial off the shelf software or existing software compo-available, decide to identify and then avoid obvious risks.

5) Conduct a post-mortem analysis.
Establish a consistent mechanism for extracting lessons learned for each project.
Evaluate the planned and actual schedules, collect and analyse software project metrics, get feedback

from team members and customers, and record findings in written form.

6) Describe about the scope of software product and problem decomposition.

A software project manager has to examine the product in order to obtain the quantitative estimates and organizational plan.By examining the product the scope of the product can be decided.

Software scope

- The first step in software project management is to determine the scope of the software project. Following questions need to be answered for determining the scope of the project-

Context

- How does the software built fit into the larger system and business context?
- What are the constraints imposed on the context of the project.

Information Objectives

- What are the visible data objects that get produced as an output of the software?
- What are the data objects that are required for the input of the required for the input of the software?

Function and performance

- For transforming the input data to an output of the software what are the required functions?
- What are special performance characteristics?
- The software scope must be unambiguous and understandable at the management and technical levels. A statement of scope must be bounded. Constraints and limitations are noted and mitigating factors are described.

Problem decomposition

- It means partitioning or elaborating the problems.
- During the project scope not attempt is made to decompose the problem fully. The decomposition that is applied in based on two major areas:
- Functionalities that must be delivered.
- The process that will be used to deliver it.
- Normally “Divide and Conquer” strategy is applied to partition the product into smaller pieces which can be managed easily as project planning begins.
- The software function described in the scope are evaluated and refined to provide more details before the beginning of the estimation.
- As both cost and schedule are functionally oriented, some degree of decomposition is often useful.
- As the statement of scope evolves the first level of partitioning naturally occurs.

7) What is importance of team management?

For performance management the Airline Council has developed a list of critical policies-
Many successful software projects and organizations have used these practises. Following is a quick look up for critical practises.

Formal Risk Management

- What are the possible project risks? What is the chance that the risk will become the problem? And what will be the impact of those risks in the overall project?

Empirical cost and schedule management

- What will be the current estimated cost of the application software and how will it be derived?

Metric based project management

- Is there any metric program available that will indicate the evolving problems? If So, are those metrics computed using the activity network or task network?

Earned value tracking

- Do the earned value metrics tracked periodically?

Defect tracking

- Do you track the project periodically and inspect the number of defects that are occurring the project? Are there any reviews or meeting conducted for this purpose?

People aware project management

- What is the average staff turnover for last few months for each of the supplier per developer during project management?
- If the software project team cannot answer these questions with satisfactory justification then review of project practises is indicated.