SOFTWARE PROJECT MANAGEMENT MID-TERM QUESTIONS

## Set A

1. What is the role of critical path in project planning? (Unit 6: Risk Management)

Project management generally requires directing a series of time-limited activities and tasks to meet project objectives. Proper scheduling, resource management and monitoring is critical. One of the major planning and scheduling techniques for project management is referred to as the critical path method or critical path analysis. It was first developed in the 1950s and was soon adopted as a tool to manage complex federal defense projects. Now it is used in business organizations to plan and manage a wide variety of projects and programs.

## Activities and Tasks

1. Critical path analysis starts with developing a list of all activities and tasks required to complete an objective. In addition to staff performance requirements, project activities might include, for example, any staff training or outsourced activities that may be required to achieve the objective. Each activity and task is lettered for purposes of later linkage with interdependent activities and tasks.

## Timescales

1. In critical path planing, timescales are used to identify the amount of time necessary to complete a given task or activity, which is generally necessary to develop a realistic estimate for completing the final objective. Alternatively, if you have a 10-week deadline to complete a project objective, you will then need to schedule realistic deadlines for each task and activity that works within the deadline set for the project. On a critical path analysis chart, time estimates are recorded alongside the list for each task or activity.

## Sequencing

1. One of the major concepts in critical path analysis relates to sequencing activities and tasks. In planning, this acknowledges that some activities must first be completed before others can be performed. These are called parallel activities. Time sequencing involves identifying parallel activities and tasks and properly ordering them within a project schedule. In critical path analysis planning, this makes it easy for planners to next develop precise sequencing and time schedules for parallel tasks and activities.

## Benefits

1. Critical path analysis is an effective technique for project management. It assists project managers in developing timescales and deadlines for tasks and activities. Also, by identifying the interrelationship between activities and tasks, project managers are better able to develop time sequences and set priorities to reduce the potential for project delays. For complex projects, it can be useful in evaluating problems that arise during the implementation stage of the project plan.

2. Explain the different characteristics of a Project Manager. (Unit1: Introduction to SPM)

1. **Objectives oriented:** Project management is focused on achieving specific project objectives with customer satisfaction. It is results-oriented.
2. **Change oriented:** Project management is a vehicle for planning and managing change in an organized manner. It adopts flexibility in doing things in a risky environment.
3. **Single Responsibility Center:** The project manager is the single responsibility center accountable for project outcomes. The role of project manager is crucial from inception to completion of the project. He is a project leader and champion. He motivates team members to excel.
4. **Team-based:** Project management consists of a multi-disciplinary project team with a wide range of skill and experiences. The team has project dedication. Each member has responsibility and accountability for a unit of work. Self-management is emphasized. So is member participation. The team membership is flexible and changes with project needs.
5. **Functional Coordination:** Project management requires coordination along functional lines. The work flow is both vertical and horizontal in a matrix organization structure.
6. **Planning and Control:** Project management required integrated planning and control systems for continuous improvement.
7. **Constraints:** Project management achieves results within the constraints of time, cost and quality. It is a time and resources limited activity. It is focused on customer needs.
8. **Body of Knowledge:** Project management consists of a body of knowledge like;

* Scope Management
* Time Management
* Change Management
* Integration Management
* Cost Management
* Quality Management
* Human Resource Management
* Conflict Management
* Risk Management
* Procurement Management

3.What is the difference between project monitoring and project controlling? (Unit 5:

Project Planning, Monitoring & Control)

Monitoring and controlling is essentially required in any project simply because things don’t always go according to plan no matter how much we prepare and to detect and react appropriately to deviations and changes to the plan.

**Monitoring** is concerned with the gathering of information and connecting them with the project plans and objectives. Project Monitoring refers to the method of keeping track of all project-related metrics including team performance and task duration, identifying possible problems, and taking remedial actions necessary to ensure that the project is within scope, on budget, and meets the stipulated deadlines.It answers questions like “have input and resources been provided as planned?” “Have activities been completed as planned?” “Have output been produced as anticipated?” “Is the work of the project progressing as planned?”. It indicates to the project manager where the [project performance](https://www.projecttimes.com/george-pitagorsky/measuring-in-progress-project-performance.html) is in terms of time, [quantity](https://www.constructionplacements.com/quantity-surveyor-job-description-and-salary/), quality, [risk](https://www.constructionplacements.com/types-of-risks-in-construction-projects/), money, and other areas of project progress.

**Control** is the corrective action that is undertaken if the desired result is not achieved. They are three separate actions but go hand in hand as tools for assessing the status and success of a project. Control uses the monitored [data](https://www.constructionplacements.com/data-science-for-construction-architecture-and-engineering-free-online-course/) and information to bring actual performance into an agreement with the plan. It Involves comparing actual performance with planned performance and taking appropriate corrective action that will yield the desired outcome in the project when a significant difference exists.

Changes to the original project execution plan are inevitable. Changes will always occur in the project. but we must be able to manage the changes as they occur.It is an important job of the project manager to identify all changes from the original project scope & Plan and manage them. Managing changes is one of the most challenging areas of [construction management](https://www.constructionplacements.com/construction-management/) and if left unchecked can result in the project run off the track.

4.What is a Work Breakdown Structure (WBS) and how does it help in preparing a good

plan? (Unit5: Project Planning, Monitoring and Control)

A work breakdown structure (WBS) is a tool that can be used for projects, programs, and even initiatives to understand the work that has to be done to successfully produce a deliverable. A work breakdown structure (WBS) lets project managers plan their work more efficiently. A project is characterized by time-limited activities and is assigned fixed time frames and costs. When it is finished, a project must fulfill the stakeholder needs it was designed to address. The project management has to plan for the schedule, the fixed costs and the functional completeness of the project and assign responsibilities. The WBS helps make this planning consistent and provides for effective project execution.

The main purpose of a WBS is to reduce complicated activities to a collection of tasks. This is important for the project manager because she can oversee the tasks more effectively than the complex activities. Tasks must be measurable and independent, with clearly defined limits. All the project work must be included in one of the tasks and the tasks must not include any non-project work.

5. How will you identify stakeholders of software project and why do you know more

about them? (Unit 8: Requirements Management)

Project stakeholders are any group of people that can have an influence or can be influenced by the project. In other words, they have a stake in the project. After completion, whether the project failed or succeeded depends on how much the stakeholders are satisfied.

Project stakeholders can be broadly categorized into two ways:



### 1. Internal stakeholders

These individuals or organizations are involved in the project from within the organization. They include (but are not limited to):

* **The project team:** They are a group of individuals who work on the project to ensure its [execution](https://kissflow.com/project/project-execution-phase/).
* **The project manager:** The project manager has the responsibility of [project planning](https://kissflow.com/project/steps-to-create-successful-project-plan/), procuring, and executing the project.
* **The sponsor:** Project sponsors own the project and ensure the provision of resources for the project. They hold responsibility for the project.
* **Internal teams:** Other teams can use the outcome of the project. For example, the sales team.

### 2. External stakeholders

External stakeholders are affected by the project’s outcome even though they are not involved in the project directly. They can include:

* The external clients
* The end-user
* The subcontractors
* The government
* The supplier
* The community

## How to identify stakeholders in a project

There has to be a significant effort toward identifying project stakeholders early on. This involves prioritizing stakeholders according to their vested interests in the project as well as their overall impact and influence on the project. The process of identifying stakeholders ideally starts when your sponsor approves the project charter.



There following documents and techniques can help you identify the stakeholders:

### 1. Project Charter

This document normally identifies the project head, key sponsors, clients, and influencers. The [project charter](https://kissflow.com/project/project-charter/) can provide useful insights for identifying the stakeholders.

### 2. Reviewing the Enterprise Environmental Factors

The enterprise environmental factors (EEF) may involve factors such as competition, the company’s structure, and industry standards. Reviewing these factors can help you identify the stakeholders.

### 3. Interviewing the influencers

Interviewing the [project management experts](https://kissflow.com/project/project-management-strategies-discussed/) and key influencers is also instrumental in identifying the key stakeholders.

### 4. Asking questions

Brainstorming sessions between the [project team members](https://kissflow.com/project/project-team-management/) and experts are a great way of getting the stakeholders involved. For getting a clearer picture, try answering these questions:

* Who are this project’s shareholders?
* Who is involved in the project? Directly or indirectly.
* Who gains if the project is successful?
* Who are the suppliers?
* Who can accept or reject the project?
* Who might be affected by the project’s outcome?

Answering questions like these can go a long way in successfully identifying your project’s stakeholders.

6. Explain the Software Configuration Management (SCM) with examples. (Unit10: SCM)

Software configuration management (SCM) is a software engineering discipline consisting of standard processes and techniques often used by organizations to manage the changes introduced to its software products. SCM helps in identifying individual elements and configurations, tracking changes, and version selection, control, and baselining.

SCM is also known as software control management. SCM aims to control changes introduced to large complex software systems through reliable version selection and version control.

When we develop software, the product (software) undergoes many changes in their maintenance phase; we need to handle these changes effectively.

Several individuals (programs) works together to achieve these common goals. This individual produces several work product (SC Items) e.g., Intermediate version of modules or test data used during debugging, parts of the final product.

The elements that comprise all information produced as a part of the software process are collectively called a software configuration.

As software development progresses, the number of Software Configuration elements (SCI's) grow rapidly.

**These are handled and controlled by SCM. This is where we require software configuration management.**

7.Explain the Black box testing. (Unit 9: Software test management)

Black box testing involves testing a system with no prior knowledge of its internal workings. A tester provides an input, and observes the output generated by the system under test. This makes it possible to identify how the system responds to expected and unexpected user actions, its response time, usability issues and reliability issues.

Black box testing is a powerful testing technique because it exercises a system end-to-end. Just like end-users “don’t care” how a system is coded or architected, and expect to receive an appropriate response to their requests, a tester can simulate user activity and see if the system delivers on its promises. Along the way, a black box test evaluates all relevant subsystems, including UI/UX, web server or application server, database, dependencies, and integrated systems.

An example of a [security technology that performs black box testing is Dynamic](https://www.imperva.com/learn/data-security/information-security-infosec/) [Application Security Testing](https://www.imperva.com/learn/application-security/application-security/) (DAST), which tests products in staging or production and provides feedback on compliance and security issues.

[](https://www.imperva.com/learn/wp-content/uploads/sites/13/2020/03/thumbnail_Black-box.jpg)

### Functional Testing

Black box testing can test specific functions or features of the software under test. For example, checking that it is possible to log in using correct user credentials, and not possible to log in using wrong credentials.

Functional testing can focus on the most critical aspects of the software (smoke testing/sanity testing), on integration between key components (integration testing), or on the system as a whole (system testing).

### Non-Functional Testing

Black box testing can check additional aspects of the software, beyond features and functionality. A non-functional test does not check “if” the software can perform a specific action but “how” it performs that action.

Black box tests can uncover if software is:

* Usable and easy to understand for its users
* Performant under expected or peak loads
* Compatible with relevant devices, screen sizes, browsers or operating systems
* Exposed to security vulnerabilities or common security threats

### Regression Testing

Black box testing can be used to check if a new version of the software exhibits a regression, or degradation in capabilities, from one version to the next. Regression testing can be applied to functional aspects of the software (for example, a specific feature no longer works as expected in the new version), or non-functional aspects (for example, an operation that performed well is very slow in the new version).

White box testing

Many practitioners combine black box testing with white box testing. [White box testing](https://www.imperva.com/learn/application-security/white-box-testing/) involves testing an application with detailed inside information of its source code, architecture and configuration. It can expose issues like security vulnerabilities, broken paths or data flow issues, which black box testing cannot test comprehensively or at all.

By combining black box and white box testing, testers can achieve a comprehensive “inside out” inspection of a software application and increase coverage of quality and security issues.

## Grey Box Testing

While white box testing assumes the tester has complete knowledge, and black box testing relies on the user’s perspective with no code insight, [grey box testing](https://www.imperva.com/learn/application-security/gray-box-testing/) is a compromise. It tests applications and environments with partial knowledge of internal workings. Grey box testing is commonly used for [penetration testing](https://www.imperva.com/learn/application-security/penetration-testing/), end-to-end system testing, and integration testing.

8. Explain the steps involved in Cost-benefit Analysis. (Unit 4: Software estimation)

The cost-benefit analysis compares the costs and benefits of a project and then makes a decision on whether or not to proceed with the project. The project’s costs and benefits are measured in monetary terms after adjusting for the time value of money, thus providing a true picture of the costs and benefits. Net Present Value and Benefit-Cost Ratio are the two most common methods of doing a cost-benefit analysis. The NPV model chooses the project with the highest NPV. The benefit-cost ratio model chooses the project with the highest benefit-cost ratio.

### Steps of Cost-Benefit Analysis

We all know it’s quite simple to make an investment decision when the benefits overshadow the costs, but only a few of us know the other key elements that go into the analysis. The steps to create a meaningful model are:

The steps to create a meaningful Cost-Benefit Analysis model are:

1. **Define the framework for the analysis.**

Identify the state of affairs before and after the policy change or investment on a particular project. Analyze the cost of this status quo. We need to first measure the profit of taking up this investment option instead of doing nothing or being on ground zero. Sometimes the status quo is the most lucrative place to be in.

1. **Identity and classify costs and benefits.**

It is essential to costs and benefits are classified in the following manner to ensure that you understand the effects of each cost and benefit.  
– Direct Costs (Intended Costs/Benefits)  
– Indirect Costs (Unintended Costs/Benefits),  
– Tangible (Easy To Measure And Quantify)/  
– Intangible (Hard To Identify And Measure), And  
– Real (Anything That Contributes To The Bottom Line Net-Benefits)/Transfer (Money Changing Hands)

1. **Drawing a timeline for expected costs and revenue.**

When it comes to decision making, timing is the most crucial element. Mapping needs to be done when the costs and benefits will occur and how much they will pan out over a phase. It solves two major issues. Firstly, a defined timeline enables businesses to align themselves with the expectations of all interested parties. Secondly, understanding the timeline allows them to plan for the impact that the cost and revenue will have on the operations. This empowers businesses to better manage things and take steps ahead of any contingencies.

1. **Monetize costs and benefits.**

We must ensure to place all costs and all benefits in the same monetary unit.

1. **Discount costs and benefits to obtain present values.**

It implies converting future costs and benefits into present value. It is also known as discounting the cash flows or benefits by a suitable discount rate. Every business tends to have a different discount rate.

1. **Calculate net present values.**

It is done by subtracting costs from benefits. The investment proposition is considered efficient if a positive result is obtained. However, there are other factors to be considered, as well.

## Set b

1. List out the activities used in SPM. (Unit 1: introduction to SPM)

Software Project Management consists of many activities, that includes planning of the project, deciding the scope of product, estimation of cost in different terms, scheduling of tasks, etc.

**The list of activities are as follows:**

1. Project planning and Tracking
2. Project Resource Management
3. Scope Management
4. Estimation Management
5. Project Risk Management
6. Scheduling Management
7. Project Communication Management
8. Configuration Management

Now we will discuss all these activities -

**1. Project Planning:** It is a set of multiple processes, or we can say that it a task that performed before the construction of the product starts.

**2. Scope Management:** It describes the scope of the project. Scope management is important because it clearly defines what would do and what would not. Scope Management create the project to contain restricted and quantitative tasks, which may merely be documented and successively avoids price and time overrun.

**3. Estimation management:** This is not only about cost estimation because whenever we start to develop software, but we also figure out their size(line of code), efforts, time as well as cost.

If we talk about the size, then Line of code depends upon user or software requirement.

If we talk about effort, we should know about the size of the software, because based on the size we can quickly estimate how big team required to produce the software.

If we talk about time, when size and efforts are estimated, the time required to develop the software can easily determine.

And if we talk about cost, it includes all the elements such as:

* Size of software
* Quality
* Hardware
* Communication
* Training
* Additional Software and tools
* Skilled manpower

**4. Scheduling Management:** Scheduling Management in software refers to all the activities to complete in the specified order and within time slotted to each activity. Project managers define multiple tasks and arrange them keeping various factors in mind.

**For scheduling, it is compulsory -**

* Find out multiple tasks and correlate them.
* Divide time into units.
* Assign the respective number of work-units for every job.
* Calculate the total time from start to finish.
* Break down the project into modules.

**5. Project Resource Management:** In software Development, all the elements are referred to as resources for the project. It can be a human resource, productive tools, and libraries.

Resource management includes:

* Create a project team and assign responsibilities to every team member
* Developing a resource plan is derived from the project plan.
* Adjustment of resources.

**6. Project Risk Management:** Risk management consists of all the activities like identification, analyzing and preparing the plan for predictable and unpredictable risk in the project.

Several points show the risks in the project:

* The Experienced team leaves the project, and the new team joins it.
* Changes in requirement.
* Change in technologies and the environment.
* Market competition.

**7. Project Communication Management:** Communication is an essential factor in the success of the project. It is a bridge between client, organization, team members and as well as other stakeholders of the project such as hardware suppliers.

From the planning to closure, communication plays a vital role. In all the phases, communication must be clear and understood. Miscommunication can create a big blunder in the project.

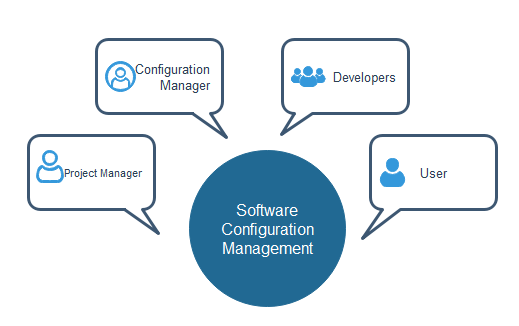
**8. Project Configuration Management:** Configuration management is about to control the changes in software like requirements, design, and development of the product.

The Primary goal is to increase productivity with fewer errors.

**Tasks perform in Configuration management:**

* Identification
* Baseline
* Change Control
* Configuration Status Accounting
* Configuration Audits and Reviews

**People involved in Configuration Management:**



2. List out the advantages and disadvantages of Rapid Application Development (RAD)

software development life cycle. (Unit 3: Software life Cycle Models)

Rapid Application Development (RAD) is a development model that prioritizes rapid prototyping and quick feedback over long drawn out development and testing cycles. With rapid application development, developers can make multiple iterations and updates to a software quickly without starting from scratch each time. This helps ensure that the final outcome is more quality-focused and is in alignment with the end-users’ requirements.

|  |  |
| --- | --- |
| **Advantages of RAD Model** | **Disadvantages of RAD Model** |
| Requirements can be changed at any time | Needs strong team collaboration |
| Encourages and priorities customer feedback | Cannot work with large teams |
| Reviews are quick | Needs highly skilled developers |
| Development time is drastically reduced | Needs user requirement throughout the life cycle of the product |
| More productivity with fewer people | Only suitable for projects which have a small development time |
| Time between prototypes and iterations is short | More complex to manage when compared to other models |
| Integration isn’t a problem, since it integrates from project inception | Only systems which can be modularised can be developed using Rapid application development. |

1. Briefly explain about risk planning and controlling. (Unit 6: Risk Management)

**Risk** is defined as:

* An uncertain event or condition that if it occurs, has a positive or negative effect on a Project’s Objectives (PMBOK).
* The effect of uncertainty on objectives (ISO 31000).
* The possibility that something bad or unpleasant will happen (Miriam-Webster)

**Risk planning** is the process of identifying, prioritizing, and managing risk.

Every project or initiative has ***objectives***, that is, goals that it seeks to accomplish. These are often called [Critical Success Factors](http://www.projectengineer.net/39-project-success-factors/) (CSF).

***Risk events*** threaten the successful completion of these critical success factors. Thus, risk planning involves identifying the most important risk events in advance, prioritizing them, and developing the appropriate risk response plans.  There are three steps to risk planning:

1. Identifying Risks
2. Prioritizing Risks
3. Determining Response Plans

## Identifying Risks

A strong risk identification process is important to the successful completion of the critical success factors. This is particularly true for large or inherently risky projects, like nuclear power plants. But if it’s beneficial for large projects, an appropriately sized risk planning process will benefit small projects too.  A [Risk Management Plan](http://www.projectengineer.net/parts-of-a-risk-management-plan/) is prepared which includes items such as:

* Risk Register
* [Risk Breakdown Structure](http://www.projectengineer.net/do-you-need-a-risk-breakdown-structure/)
* [Risk Analysis](http://www.projectengineer.net/project-risk-analysis/)

The risk register is the itemized listing of most important risks and it becomes the cornerstone of the Risk Management Plan.  It requires careful consideration of the project risks and what could affect the project’s critical success factors.  Here are a few ideas to ensure that each risk is identified:

Identifying risks to a project’s success is a great first step that would benefit most projects that I’ve seen.  But to create a strong risk management plan, those risks must be [analyzed and prioritized](http://www.projectengineer.net/knowledge-areas/project-risk/risk-analysis/) to determine which require the project manager’s time and attention, how often, and what resources are required.

Stakeholders can be pretty sensitive to issues the project manager considers minor.  Some stakeholders seem to demand excessive communication requirements.  Prioritizing risks ensures that stakeholders recognize the importance placed on their areas of concern which goes a long ways toward placating them.

Since risk has two components:

***Risk = Probability x Impact***

Each of these factors should be prioritized.  The scale is not important, but it is often 1-10, low-medium-high, or a similar scale.

If your risk register is a table with the risks listed vertically (in rows), you would add two columns labelled probability and impact.  Each risk gets a ranking of 1-10, or whatever scale you choose.

After the initial ranking, an overall prioritization is often helpful to stakeholders.  You would multiply the probability and impact to get a risk level, and then sort the table from highest to lowest.  Clearly you will be able to see which risks to focus your attention on.

Even better, the risk register, together with its prioritizations, can be shared with stakeholders which will put each of their issues into perspective, keeping the stakeholders in line when problems arise.

An example risk register might look like this:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risk** | **Probability** | **Impact** | **Priority** | **Trigger** | **Response Plan** |
| Miss completion date due to inclement weather | 3 | 6 | 18 | Site foreman to decide | See below |
| Deficient materials arrive at site | 1 | 8 | 8 | Site foreman inspects material upon arrival and decides | Send material back for full refund |

## Determine Risk control/response Plans

The final piece of information that completes the risk register is a [risk response plan](http://www.projectengineer.net/creating-risk-response-plans/).  Now that you’ve identified the triggers that allow you to quickly identify when a risk has occurred (or is occurring), the response plan gives you a head start in the response.  Some responses occur at the beginning of the project (when the risk planning process is taking place) and others occur when the risk event occurs.  Still other occur at any applicable time during the project.

There are four possible responses to risk events:

1. **Avoid.**  Eliminate the threat.  For example, change the scope of the project, spin off a certain business unit, or change the objectives that the risk event is threatening.
2. **Transfer.**  Off-load the risk to a third party.  For example, buy insurance, issue a performance bond, or change the contract from a lump sum to a unit price (or vice versa).
3. **Mitigate.**  Reduce the probability or impact of the risk event.  For example, cover the project area to prevent work stoppages due to inclement weather, or purchase materials in advance to ensure they can be returned without threatening the project completion date.
4. **Accept.**  Sometimes there is no other alternative than to proceed with the project and accept the risk. But producing documentation, holding meetings, and communicating the risk with stakeholders can go a long ways toward minimizing the damage.
5. Explain the tools for collecting the requirements. (Unit 8: Requirements management)

The collect requirement is a process that determines, as well as documents and manages the needs and requirements of the stakeholders, to meet the objectives of the Project Management task. The documentation that takes place within the collect requirement process is considered important as it provides the foundation for defining and managing the scope of the project.

## **Collect Requirements Tools and Techniques**

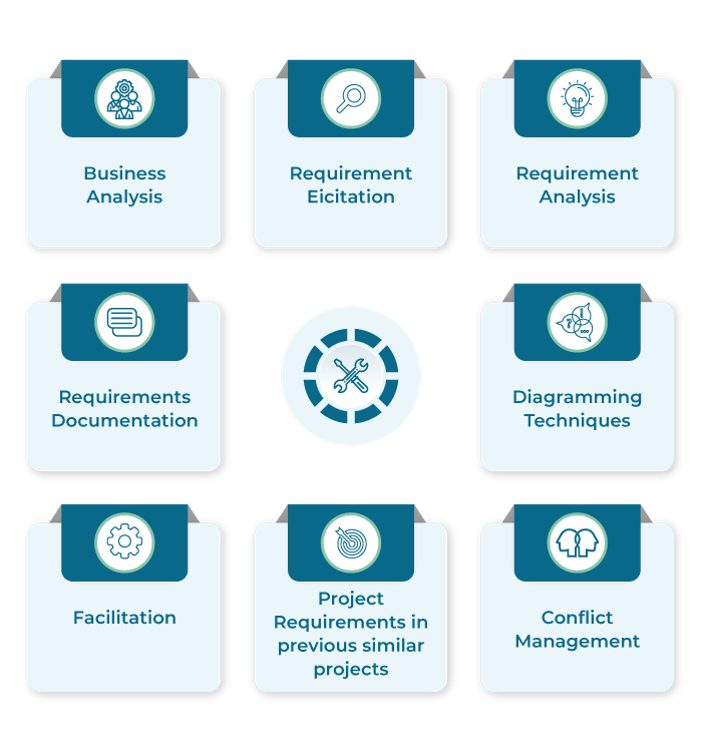
The following are included as the tools and techniques in the collect requirements process:

1. Expert Judgement
2. Data Gathering
3. Data Analysis
4. Decision Making
5. Data Representation
6. Interpersonal & Team Skills
7. Context Diagram
8. Prototypes

Let’s take a look at each of them in detail.

### **Expert Judgment**

Listed below are the topics in which the individuals or groups should have specialized knowledge and expertise:



### **Data Gathering**

Data gathering techniques that can be used for the Collecting Requirements process are :

* 1. **Brainstorming**  
     A group thinking activity, where several people from various teams come together to list requirements for a project. And during the brainstorming session, new ideas are generated from existing plans, which helps to identify new requirements.
  2. **Interviews**  
     It is the first collect requirements technique. It can be done in both formal and informal ways. The critical feature of this process is that it helps the project manager to interview experienced project participants, sponsors, stakeholders and other executives, and subject matter experts who can aid in identifying and defining the features and functions of the desired product deliverables.
  3. **Focus Groups**

The focus groups bring together pre-qualified stakeholders and subject matter experts to learn about their expectations and attitudes about a proposed project. The Focus Group is a technique used to get a specific set of stakeholders’ requirements. For instance, first, the project manager can organize a meeting with executive directors of a company to get their requirements, and then arrange a separate meeting with the functional managers to understand their requirements.

* 1. **Questionnaires and Surveys**

This technique is best used when there are more stakeholders involved in a project. For example, if there are 200 stakeholders associated with the project, collecting information from each individual to assess their requirements will consume a lot of time. Hence, the project manager is requested to prepare a questionnaire and conduct surveys to collect their requirements list.

* 1. **Bench-marking**  
     It is a process that is used to compare the actual or planned practices – procedures and operations, to those of comparable organizations (internal or external) to identify best practices, generate ideas for improving the scope, and provide a framework for measuring the actual performance.

### **Data Analysis**

Data analysis mainly deals with the processes that are related to document analysis. The primary purpose of document analysis is to review and assess all the relevant document information. This process is used to obtain requirements by carefully analyzing the existing documents and identifying relevant details on the requirements.

### **Decision Making**

The decision-making techniques that can assist in the Collect requirement process are :

* **Voting**  
  It is the collective decision-making technique and an assessment process that has various alternatives with a defined outcome. These techniques are further used to generate, classify, and prioritize product requirements. Listed below are the examples of voting techniques:
* **Autocratic Decision Making**  
  As the title states, the decision is taken by a single individual who has the ultimate authority in the organization.
* **Multi-criteria Decision Analysis**  
  A technique in which a decision matrix is used to provide a systematic and analytical approach for determining criteria such as risk levels, uncertainty, and valuation, to evaluate and rank many ideas.

### **Data Representation**

The data representation techniques that can be used for the process of collecting requirements are as follows:

* 1. **Affinity Diagram**

A technique where all the ideas that are collected or gathered are segregated accordingly based on their similarities

* 1. **Idea/Mind Mapping**

The ideas that are generated through the brainstorming sessions are consolidated into a single map to filter out the conventional concepts and understand the differences in opinions which will help in developing new plans.

### **Interpersonal and Team Skills**

The interpersonal and team skills techniques that can be used for the process of collecting requirements are as follows:

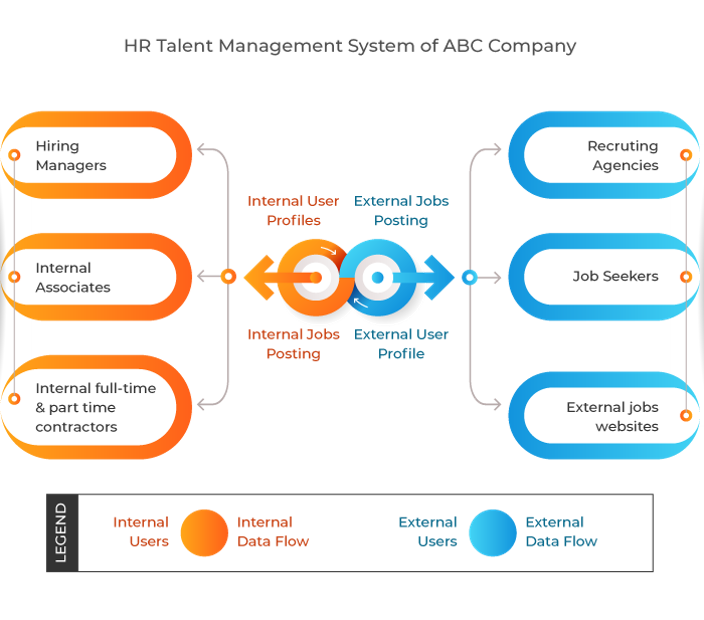
* 1. **Nominal Group Technique**

The technique uses skills to prioritize the already existing ideas rather than developing new ideas. In this process, the plans are ranked based on their value, and this helps the teams to focus on the essential concepts to generate the project requirements.

* 1. **Observations**  
     Also known as ‘Job Shadowing’, Observation is a process where an observer views a business expert performing his job. This process is mainly to strictly observe the activities taking place across various areas to find out what are the actual requirements of the consumer, stakeholder, sponsor, etc.
  2. **Facilitation**  
     It is a technique that has focused sessions that bring key stakeholders together to define product requirements. In general, each group of project stakeholders will look to the project from their very own perspective and express their requirements. Workshops are considered a primary technique for quickly defining cross-functional requirements and reconciling stakeholder differences.

### **Context Diagram**

To have actual knowledge of understanding the scope, the context diagram provides an example of the scope model, which will allow the project manager to visualize how a business system (Process, Equipment, and Computer systems) will work. This process is used to find out how the business system and the other users interact with each other.

****

### **Prototypes**

The technique, which involves the process of creating a model of the actual product, is to be achieved. The team will build this model of the product and pass it on to the stakeholders for collecting their feedback. As the model is a tangible product, the stakeholders can check it thoroughly and request for changes if they find any, instead of waiting until the end by only sharing and discussing abstract representations and ideas.

* 1. How you can select the right person for the job? Explain. (Unit 5: Project planning)

Having the right people on staff is crucial to the success of an organization. Various selection devices help employers predict which applicants will be successful if hired. These devices aim to be not only valid, but also reliable. *Validity* is proof that the relationship between the selection device and some relevant job criterion exists. *Reliability* is an indicator that the device measures the same thing consistently. For example, it would be appropriate to give a keyboarding test to a candidate applying for a job as an administrative assistant. However, it would not be valid to give a keyboarding test to a candidate for a job as a physical education teacher. If a keyboarding test is given to the same individual on two separate occasions, the results should be similar. To be effective predictors, a selection device must possess an acceptable level of consistency.

For most employers, the application form is the first step in the selection process. Application forms provide a record of salient information about applicants for positions, and also furnish data for personnel research. Interviewers may use responses from the application for follow‐up questions during an interview.

These forms range from requests for basic information, such as names, addresses, and telephone numbers, to comprehensive personal history profiles detailing applicants' education, job experience skills, and accomplishments.

* 1. Define software quality and its importance. (Unit 12: SQA)

Software quality is defined as a field of study and practice that describes the desirable attributes of software products. There are two main approaches to software quality: defect management and quality attributes.

**Testing** is the act of checking the software code and design to detect errors and flaws. There are different types of testing procedures such as Grey Box, [Black Box, and White Box testing](https://www.goodcore.co.uk/blog/black-box-and-white-box-testing/) and they can be automated or conducted manually by the QA team to check the performance of a software’s code and design. Testing can be conducted during the testing phase in the [software development life cycle](https://www.goodcore.co.uk/blog/sdlc-methodologies/) or throughout the development process.

**Quality Control** refers to the process of validating that the developed product corresponds to the requirements and specifications. It is conducted on a completed or near-completed product to ensure that all its features perform perfectly together.

While testing focuses on errors in code and design, quality control focuses on verifying the quality of the product as a whole.

**Quality Assurance**, on the other hand, is a broad term that encompasses both testing and quality control and focuses on the quality of the overall process of software development. It refers to the consistent improvement of processes to ensure better quality control. Unlike testing and QC, QA is involved at every stage of the development process. It makes sure that each step of the process (Requirement gathering and analysis, planning, designing, coding and development, testing, and launch) is performed right so that quality is maintained throughout the project lifecycle.

## **Reasons Why Quality Assurance Is Crucial for Every Software Development Project**

The result of a thorough quality assurance approach is indeed a high-quality and fault-free software product. But quality assurance is much more than that. In addition to identifying errors and areas of improvement in a product, quality assurance is important for many other aspects of the business such as client relations and the business’ reputation in the market. Let’s shed light on some of the top reasons why quality assurance is essential for every software development project.

## **Saves Time and Money**

Consistent software quality assurance is necessary if you want to save time and money. It can be expensive if mistakes and bugs go unidentified till the later stages in the development process because fixing mistakes in an already designed and coded product is going to be time-consuming, hence cost more money.

As a software provider, it is your responsibility to maintain quality standards at every step along the development process through testing and quality assurance so errors can be identified and fixed early on. A well-thought-out quality assurance strategy will ensure that no mistakes are carried forward to the final stages to prevent any major loss for the software development company and the client.

## **Ensures Quality and Competitiveness of the Product**

The tech industry is unforgiving when it comes to software quality and performance. Examples of software failure mentioned above speak for the consequences a company could face for releasing a faulty software product.

Why is quality assurance important in software development? Well, a simple answer is that if you want your software products to compete in the cut-throat software business, then your products should always maintain quality and standard of performance to compete with the rest. This sheds light on the importance of quality assurance from the standpoint of market competitiveness. Better quality assurance maintains the quality and the competitiveness of the product before it is released to the customers and with every update rollout afterwards.

## **Ensures Security**

Security is always a big question for software products. Whilst a software application may meet its functional requirements and perform exactly as it is supposed to, it may not be completely secure. Vulnerabilities in a software system’s security can compromise the user data and credibility of the software.

This is one of the main reasons why software quality assurance is so important. It helps uncover vulnerabilities in a software’s security and helps build a product that is fully secure and trustworthy.

## **Maintains Goodwill of the Business**

As discussed in the points above, quality assurance is important to avoid loss of time, money, and competitiveness of software products. But software quality assurance is also important to protect a business’ reputation. The inconvenience and, in some cases, legal troubles and controversy caused by faulty software systems can negatively impact the goodwill and position of the software provider in the market. More so, it can jeopardize the relationship between the company and its clients if they a product of substandard quality is delivered.

This is where the importance of QA in software development is most evident. It ensures that your product serves as an apt brand ambassador for your business.

## **Guarantees User Satisfaction**

The software development approach always focuses on the end-users of the product, and it is necessary to guarantee the best user experience. A software system with bugs and errors can cause inconvenience for the users and ruin their experience.

Throughout the development process, QA engineers ensure that the product is developed without faults and performs its expected functions properly when in the hands of the end-users. As a result, users get a quality product that they are happy to use.

* 1. Explain about SCM tasks and tools. (Unit 10: SCM)

Change management in the software context is done using the disciplined approach of *software configuration management* (SCM). In software projects where changes are not managed by **implementing the SCM process**, there is a possibility of low productivity, more rework, and more errors.

To be able to **plan for SCM** as a part of project management, it is necessary to understand the tasks in the SCM process. The five tasks of the SCM process are configuration identification, change control, version control, configuration auditing, and reporting. These tasks relate to **software configuration items** (SCIs) and can be seen as concentric layers that apply to SCIs as the project progresses.

**The Tasks in the SCM Process** course includes:

* Configuration Identification
* Change Control
* Version Control
* Configuration Auditing
* Reporting

Changes are inevitable during a software project and can arise due to various reasons. Software configuration management (SCM) provides a disciplined approach to change management. In order to implement SCM in a software project, it is necessary to understand the SCM process.

There are **five tasks in the SCM process**. These are configuration identification, change control, version control, configuration auditing, and reporting.

**Configuration identification** is at the core of the SCM process. It sets the basis for the subsequent tasks of the SCM process. Configuration identification includes defining the basis for identifying software configuration items (SCIs), defining the method to depict the relationship between SCIs, defining the identification scheme for naming SCIs, identifying SCIs, identifying the baselines to be established and the corresponding SCIs, and defining the method for acquiring SCIs in the project repository.

In the **change control task** of the SCM process, a defined process is used to handle change requests from their initiation until the time the change is implemented and released to users. The change control process includes evaluating a change request, implementing the change, and verifying and releasing the change.

The **version control task** of the SCM process is executed to manage multiple versions of a configuration and the SCIs that constitute it. Revisions, variations, and variants are types of versions of a software product; and the principles of version control apply to all of them.

The **configuration auditing task** of the SCM process is a software quality assurance (SQA) function. It is executed formally and impartially to ensure that changes have been made correctly and quality has been maintained while implementing the changes. There are three types of configuration audits—functional configuration audit (FCA), physical configuration audit (PCA), and formal qualification review (FQR).

The **reporting task** of the SCM process provides timely information on the status of the changes requested and the SCIs to the people who may be affected by the changes. These could be the people requesting for changes, the developers, the project manager, and the senior management.

## long question

1. Suppose you are asked to develop a software for automation a lift operation. List out and

explain the approaches followed by you in order to gather the requirement of system.

(5+15)

The process to gather the software requirements from client, analyze and document them is known as requirement engineering.

The goal of requirement engineering is to develop and maintain sophisticated and descriptive ‘System Requirements Specification’ document.

## Requirement Engineering Process

It is a four step process, which includes –

* Feasibility Study
* Requirement Gathering
* Software Requirement Specification
* Software Requirement Validation

Let us see the process briefly -

### Feasibility study

When the client approaches the organization for getting the desired product developed, it comes up with rough idea about what all functions the software must perform and which all features are expected from the software.

Referencing to this information, the analysts does a detailed study about whether the desired system and its functionality are feasible to develop.

This feasibility study is focused towards goal of the organization. This study analyzes whether the software product can be practically materialized in terms of implementation, contribution of project to organization, cost constraints and as per values and objectives of the organization. It explores technical aspects of the project and product such as usability, maintainability, productivity and integration ability.

The output of this phase should be a feasibility study report that should contain adequate comments and recommendations for management about whether or not the project should be undertaken.

### Requirement Gathering

If the feasibility report is positive towards undertaking the project, next phase starts with gathering requirements from the user. Analysts and engineers communicate with the client and end-users to know their ideas on what the software should provide and which features they want the software to include.

### Software Requirement Specification

SRS is a document created by system analyst after the requirements are collected from various stakeholders.

SRS defines how the intended software will interact with hardware, external interfaces, speed of operation, response time of system, portability of software across various platforms, maintainability, speed of recovery after crashing, Security, Quality, Limitations etc.

The requirements received from client are written in natural language. It is the responsibility of system analyst to document the requirements in technical language so that they can be comprehended and useful by the software development team.

SRS should come up with following features:

* User Requirements are expressed in natural language.
* Technical requirements are expressed in structured language, which is used inside the organization.
* Design description should be written in Pseudo code.
* Format of Forms and GUI screen prints.
* Conditional and mathematical notations for DFDs etc.

### Software Requirement Validation

After requirement specifications are developed, the requirements mentioned in this document are validated. User might ask for illegal, impractical solution or experts may interpret the requirements incorrectly. This results in huge increase in cost if not nipped in the bud. Requirements can be checked against following conditions -

* If they can be practically implemented
* If they are valid and as per functionality and domain of software
* If there are any ambiguities
* If they are complete
* If they can be demonstrated

## Requirement Elicitation Process

Requirement elicitation process can be depicted using the folloiwng diagram:

Requirement elicitation process

* **Requirements gathering -** The developers discuss with the client and end users and know their expectations from the software.
* **Organizing Requirements -** The developers prioritize and arrange the requirements in order of importance, urgency and convenience.
* **Negotiation & discussion -** If requirements are ambiguous or there are some conflicts in requirements of various stakeholders, if they are, it is then negotiated and discussed with stakeholders. Requirements may then be prioritized and reasonably compromised.

The requirements come from various stakeholders. To remove the ambiguity and conflicts, they are discussed for clarity and correctness. Unrealistic requirements are compromised reasonably.

* **Documentation -** All formal & informal, functional and non-functional requirements are documented and made available for next phase processing.

2. Define risk and also explain in details how will you managing risks?

(5 + 15)

Risk is uncertain events associated with future events which have a probability of occurrence but it may or may not occur and if occurs it brings loss to the project. Risk identification and management are very important task during software project development because success and failure of any software project depends on it.

**Various Kinds of Risks in Software Development :**

1. **Schedule Risk :**  
   Schedule related risks refers to time related risks or project delivery related planning risks. The wrong schedule affects the project development and delivery. These risks are mainly indicates to running behind time as a result project development doesn’t progress timely and it directly impacts to delivery of project. Finally if schedule risks are not managed properly it gives rise to project failure and at last it affect to organization/company economy very badly.

Some reasons for Schedule risks –

* + Time is not estimated perfectly
  + Improper resource allocation
  + Tracking of resources like system, skill, staff etc
  + Frequent project scope expansion
  + Failure in function identification and its’ completion

1. **Budget Risk :**  
   Budget related risks refers to the monetary risks mainly it occurs due to budget overruns. Always the financial aspect for the project should be managed as per decided but if financial aspect of project mismanaged then there budget concerns will arise by giving rise to budget risks. So proper finance distribution and management are required for the success of project otherwise it may lead to project failure.

Some reasons for Budget risks –

* + Wrong/Improper budget estimation
  + Unexpected Project Scope expansion
  + Mismanagement in budget handling
  + Cost overruns
  + Improper tracking of Budget

1. **Operational Risks :**  
   Operational risk refers to the procedural risks means these are the risks which happen in day-to-day operational activities during project development due to improper process implementation or some external operational risks.

Some reasons for Operational risks –

* + Insufficient resources
  + Conflict between tasks and employees
  + Improper management of tasks
  + No proper planning about project
  + Less number of skilled people
  + Lack of communication and cooperation
  + Lack of clarity in roles and responsibilities
  + Insufficient training

1. **Technical Risks :**  
   Technical risks refers to the functional risk or performance risk which means this technical risk mainly associated with functionality of product or performance part of the software product.

Some reasons for Technical risks –

* + Frequent changes in requirement
  + Less use of future technologies
  + Less number of skilled employee
  + High complexity in implementation
  + Improper integration of modules

1. **Programmatic Risks :**  
   Programmatic risks refers to the external risk or other unavoidable risks. These are the external risks which are unavoidable in nature. These risks come from outside and it is out of control of programs.

Some reasons for Programmatic risks –

* + Rapid development of market
  + Running out of fund / Limited fund for project development
  + Changes in Government rules/policy
  + Loss of contracts due to any reason

To protect a project from unplanned risk, project managers typically follow an ongoing risk management process which helps them identify, understand, and respond to threats and opportunities. Before beginning this process, however, it’s important to fully understand your organization’s practices and how you will conduct your risk work for that project. This plan then will drive the following steps:

* **Identify** the risks that could potentially impact your project.
* **Assign ownership** of each identified risk to a team member who will be charged with overseeing that threat or opportunity. Although some project managers prefer to assign ownership after the risks have been analyzed and prioritized, taking this step early can be beneficial. “Many times I assign an owner to the risk very early on because I want that person to drive the analysis of the risk,” Emerson notes.
* **Analyze** each risk to fully understand the driving factors involved and potential impacts. Be sure to consider the breadth and depth of each threat at this stage in order to evaluate the severity of each risk in the context of the overall project.
* **Prioritize** project risks according to urgency and the severity of the impact they could cause.
* **Respond** to your identified risks in accordance with your risk management approach, either by taking steps to prevent the risk event from occurring or to minimize the impact if it does occur. This step should include building the response as well as taking action.
* **Monitor** your risk management strategy and make changes as needed.

3. Define Quality Assurance (QA) and Quality Control (QC). Explain the different

techniques used for enhancing the quality of software project. (10+10)

### Quality Assurance

Quality assurance can be defined as "part of quality management focused on providing confidence that quality requirements will be fulfilled." The confidence provided by quality assurance is twofold—internally to management and externally to customers, government agencies, regulators, certifiers, and third parties. An alternate definition is "all the planned and systematic activities implemented within the quality system that can be demonstrated to provide confidence that a product or service will fulfill requirements for quality."

### Quality Control

Quality control can be defined as "part of quality management focused on fulfilling quality requirements." While quality assurance relates to how a process is performed or how a product is made, quality control is more the inspection aspect of quality management. An alternate definition is "the operational techniques and activities used to fulfill requirements for quality."

1. Suppose you are assigned a project manager for a software project named e-attendance

system. Explain How you are going to allocate the resources require for each activity of

the project in order to complete it on time with desire quality. Feel free to make any

assumptions. (20)

2. What is software project management? How software project differs from other projects?

(10 + 10)

Software project management is an art and discipline of planning and supervising software projects. It is a sub-discipline of software project management in which software projects planned, implemented, monitored and controlled.

It is a procedure of managing, allocating and timing resources to develop computer software that fulfills requirements.

In software Project Management, the client and the developers need to know the length, period and cost of the project

**Software Project Management (SPM)** is a proper way of planning and leading software projects. It is a part of project management in which software projects are planned, implemented, monitored and controlled.

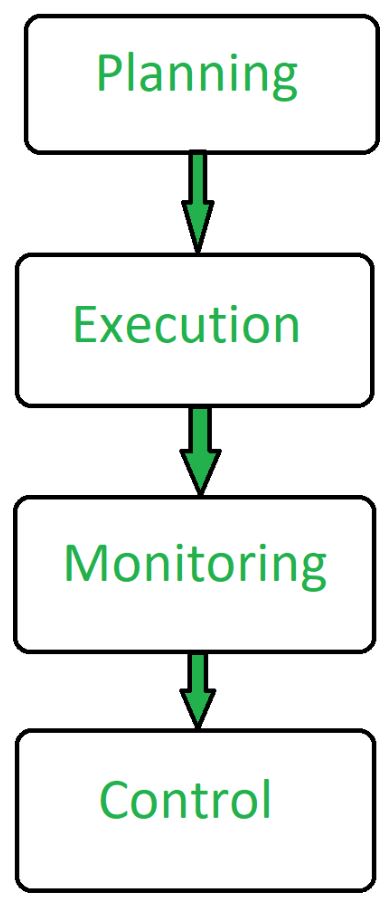
**Need of Software Project Management:**  
Software is an non-physical product. Software development is a new stream in business and there is very little experience in building software products. Most of the software products are made to fit client’s requirements. The most important is that the basic technology changes and advances so frequently and rapidly that experience of one product may not be applied to the other one. Such type of business and environmental constraints increase risk in software development hence it is essential to manage software projects efficiently.

It is necessary for an organization to deliver quality product, keeping the cost within client’s budget constrain and deliver the project as per scheduled. Hence in order, software project management is necessary to incorporate user requirements along with budget and time constraints.

**Software Project Management consists of several different type of managements:**

1. **Conflict Management:**  
   Conflict management is the process to restrict the negative features of conflict while increasing the positive features of conflict. The goal of conflict management is to improve learning and group results including efficacy or performance in an organizational setting. Properly managed conflict can enhance group results.
2. **Risk Management:**  
   Risk management is the analysis and identification of risks that is followed by synchronized and economical implementation of resources to minimize, operate and control the possibility or effect of unfortunate events or to maximize the realization of opportunities.
3. **Requirement Management:**  
   It is the process of analyzing, prioritizing, tracing and documenting on requirements and then supervising change and communicating to pertinent stakeholders. It is a continuous process during a project.
4. **Change Management:**  
   Change management is a systematic approach for dealing with the transition or transformation of an organization’s goals, processes or technologies. The purpose of change management is to execute strategies for effecting change, controlling change and helping people to adapt to change.
5. **Software Configuration Management:**  
   Software configuration management is the process of controlling and tracing changes in the software, part of the larger cross-disciplinary field of configuration management. Software configuration management include revision control and the inauguration of baselines.
6. **Release Management:**  
   Release Management is the task of planning, controlling and scheduling the build in deploying releases. Release management ensures that organization delivers new and enhanced services required by the customer, while protecting the integrity of existing services.

**Aspects of Software Project Management:**



**Advantages of Software Project Management:**

* It helps in planning of software development.
* Implementation of software development is made easy.
* Monitoring and controlling are aspects of software project management.
* It overall manages to save time and cost for software development.

3. Explain the different techniques for enhancing the quality of software project.