**Software Project Management**

The job pattern of an IT company engaged in software development can be seen split in two parts:

* Software Creation
* Software Project Management

A project is well-defined task, which is a collection of several operations done in order to achieve a goal (for example, software development and delivery). A Project can be characterized as:

1. Every project may has a unique and distinct goal.
2. Project is not routine activity or day-to-day operations.
3. Project comes with a start time and end time.
4. Project ends when its goal is achieved hence it is a temporary phase in the lifetime of an organization.
5. Project needs adequate resources in terms of time, manpower, finance, material and knowledge-bank.

**Software Project**

A Software Project is the complete procedure of software development from requirement gathering to testing and maintenance, carried out according to the execution methodologies, in a specified period of time to achieve intended software product.

Need of software project management

Software is said to be an intangible product. Software development is a kind of all new stream in world business and there’s very little experience in building software products.

Most software products are tailor made to fit client’s requirements. The most important is that the underlying technology changes and advances so frequently and rapidly that experience of one product may not be applied to the other one.

All such business and environmental constraints bring risk in software development hence it is essential to manage software projects efficiently.



The image above shows triple constraints for software projects. It is an essential part of software organization to deliver quality product, keeping the cost within client’s budget constrain and deliver the project as per scheduled.

There are several factors, both internal and external, which may impact this triple constrain triangle. Any of three factor can severely impact the other two.

Therefore, software project management is essential to incorporate user requirements along with budget and time constraints.

Software Project Manager

A software project manager is a person who undertakes the responsibility of executing the software project. Software project manager is thoroughly aware of all the phases of SDLC that the software would go through. Project manager may never directly involve in producing the end product but he controls and manages the activities involved in production.

A project manager closely monitors the development process, *prepares and executes various plans, arranges necessary and adequate resources, maintains communication among all team members in order to address issues of cost, budget, resources, time, quality and customer satisfaction.*

responsibilities that a project manager shoulders -

Managing People

1. Act as project leader
2. Liaison with stakeholders
3. Managing human resources
4. Setting up reporting hierarchy etc.

Managing Project

1. Defining and setting up project scope
2. Managing project management activities
3. Monitoring progress and performance
4. Risk analysis at every phase
5. Take necessary step to avoid or come out of problems
6. Act as project spokesperson

**Software Management Activities**

Software project management comprises of a number of activities, which contains planning of project, deciding scope of software product, estimation of cost in various terms, scheduling of tasks and events, and resource management. Project management activities may include:

* **Project Planning**
* **Scope Management**
* **Project Estimation**

Project Planning

Software project planning is task, which is performed before the production of software actually starts. It is there for the software production but involves no concrete activity that has any direction connection with software production; rather it is a set of multiple processes, which facilitates software production. Project planning may include the following:

Scope Management

It defines the scope of project; this includes all the activities, process need to be done in order to make a deliverable software product.

Scope management is essential because it creates boundaries of the project by clearly defining what would be done in the project and what would not be done. This makes project to contain limited and quantifiable tasks, which can easily be documented and in turn avoids cost and time overrun.

During Project Scope management, it is necessary to -

1. Define the scope
2. Decide its verification and control
3. Divide the project into various smaller parts for ease of management.
4. Verify the scope
5. Control the scope by incorporating changes to the scope

**Project Estimation**

For an effective management accurate estimation of various measures is a must. With correct estimation managers can manage and control the project more efficiently and effectively.

Project estimation may involve the following:

* **Software size estimation**

Software size may be estimated either in terms of KLOC (Kilo Line of Code) or by calculating number of function points in the software. Lines of code depend upon coding practices and Function points vary according to the user or software requirement.

* **Effort estimation**

The managers estimate efforts in terms of personnel requirement and man-hour required to produce the software. For effort estimation software size should be known. This can either be derived by managers’ experience, organization’s historical data or software size can be converted into efforts by using some standard formulae.

* **Time estimation**

Once size and efforts are estimated, the time required to produce the software can be estimated. Efforts required is segregated into sub categories as per the requirement specifications and interdependency of various components of software.

Software tasks are divided into smaller tasks, activities or events by Work Breakthrough Structure (WBS). The tasks are scheduled on day-to-day basis or in calendar months.

The sum of time required to complete all tasks in hours or days is the total time invested to complete the project.

* **Cost estimation**

This might be considered as the most difficult of all because it depends on more elements than any of the previous ones. For estimating project cost, it is required to consider -

* + Size of software
  + Software quality
  + Hardware
  + Additional software or tools, licenses etc.
  + Skilled personnel with task-specific skills
  + Travel involved
  + Communication
  + Training and support

**Project Estimation Techniques**

We discussed various parameters involving project estimation such as size, effort, time and cost.

Project manager can estimate the listed factors using two broadly recognized techniques –

There are two main models -

* **Line of Code** Estimation is done on behalf of number of line of codes in the software product.
* **Function Points** Estimation is done on behalf of number of function points in the software product.

**Major project estimation techniques**

Here are six common estimating methods in project management:

**1. Top-down estimate**

A top-down estimating technique assigns an overall time for the project and then breaks it down into discrete phases, work, and tasks — usually based on your project’s [work breakdown structure (WBS)](https://www.pmi.org/learning/library/applying-work-breakdown-structure-project-lifecycle-6979).

If a client tells you the project has to be done within six months, a top-down approach allows you to take that overall timeline and estimate how much time you can take for each [activity](https://www.wrike.com/project-management-guide/faq/what-is-an-activity-in-project-management/) within the project and still complete it on time.

**2. Bottom-up estimate**

A bottom-up estimate is the reverse of top-down. Using this estimation technique, you start by estimating each individual [task](https://www.wrike.com/project-management-guide/faq/what-is-a-task-in-project-management/) or aspect of the project. Then you combine all those separate estimates to build up the overall project estimate.

Since each activity is being assessed individually, this type of estimate tends to be more accurate than the top-down approach. But it also takes more time.

**3. Expert judgment**

Expert judgment is one of the most popular estimation techniques, as it tends to be quick and easy. This technique involves relying on the experience and gut feel of experts to estimate projects.

It’s most useful when you’re planning a standard project that is similar to projects your team has completed before. Expert judgment can be used for creating top-down or bottom-up estimates.

**4. Comparative or analogous estimation**

Comparative estimation uses past project data combined with a top-down approach to [estimate project duration](https://www.wrike.com/blog/how-estimate-project-duration/). If the average completion time of similar projects was eight months, you’d assume the current one will take eight months. Then you can break those eight months down across tasks and activities to get your lower-level work estimates.

**5. Parametric model estimating**

[Parametric modeling](https://www.pmlearningsolutions.com/blog/analogous-estimating-vs-parametric-estimating-pmp-concept-4) also uses past project data, but it attempts to adjust the data to reflect each project's differences. This technique takes the detail of past projects and pro-rates it to estimate the current project.

Imagine your company builds houses. Parametric modeling could take the cost of all past construction projects divided by each project's square footage to come up with an average project cost per square foot of the home. Then, you’d multiply that number by the planned square footage of the current home to create your overall project budget.

**6. Three-point estimating**

Three-point estimating is a technique sometimes used for creating bottom-up estimates. Rather than assuming one duration for a task, you may assign three: optimistic, pessimistic, and most likely. These three numbers are averaged to create your actual estimate.

The [PERT (Program Evaluation and Review Technique)](https://www.wrike.com/blog/what-is-a-pert-chart/) method uses three-point estimating, but it takes a weighted average of the three points, with the ‘most likely’ guess carrying more weight

**Project Scheduling**

Project Scheduling in a project refers to roadmap of all activities to be done with specified order and within time slot allotted to each activity.

Project managers tend to define various tasks, and project milestones and arrange them keeping various factors in mind.

They look for tasks lie in critical path in the schedule, which are necessary to complete in specific manner (because of task interdependency) and strictly within the time allocated. Arrangement of tasks which lies out of critical path are less likely to impact over all schedule of the project.

For scheduling a project, it is necessary to -

1. Break down the project tasks into smaller, manageable form
2. Find out various tasks and correlate them
3. Estimate time frame required for each task
4. Divide time into work-units
5. Assign adequate number of work-units for each task
6. Calculate total time required for the project from start to finish

**Resource management**

All elements used to develop a software product may be assumed as resource for that project. This may include human resource, productive tools and software libraries.

The resources are available in limited quantity and stay in the organization as a pool of assets. The shortage of resources hampers the development of project and it can lag behind the schedule.

Allocating extra resources increases development cost in the end. It is therefore necessary to estimate and allocate adequate resources for the project.

**Resource management includes -**

1. Defining proper organization project by creating a project team and allocating responsibilities to each team member
2. Determining resources required at a particular stage and their availability
3. Manage Resources by generating resource request when they are required and de-allocating them when they are no more needed.

**Project Risk Management**

Risk management involves all activities pertaining to identification, analyzing and making provision for predictable and non-predictable risks in the project. Risk may include the following:

1. Experienced staff leaving the project and new staff coming in.
2. Change in organizational management.
3. Requirement change or misinterpreting requirement.
4. Under-estimation of required time and resources.
5. Technological changes, environmental changes, business competition.

**Risk Management Process**

A software project can be concerned with a large variety of risks. In order to be adept to systematically identify the significant risks which might affect a software project, it is essential to classify risks into different classes. The project manager can then check which risks from each class are relevant to the project.

There are three main classifications of risks which can affect a software project:

1. Project risks
2. Technical risks
3. Business risks

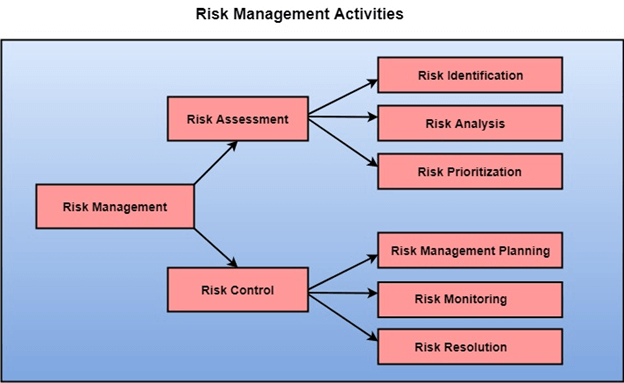
**1. Project risks:** Project risks concern differ forms of budgetary, schedule, personnel, resource, and customer-related problems. A vital project risk is schedule slippage. Since the software is intangible, it is very tough to monitor and control a software project. It is very tough to control something which cannot be identified. For any manufacturing program, such as the manufacturing of cars, the plan executive can recognize the product taking shape.

**2. Technical risks:** Technical risks concern potential method, implementation, interfacing, testing, and maintenance issue. It also consists of an ambiguous specification, incomplete specification, changing specification, technical uncertainty, and technical obsolescence. Most technical risks appear due to the development team's insufficient knowledge about the project.

**3. Business risks:** This type of risks contain risks of building an excellent product that no one need, losing budgetary or personnel commitments,

**Principle of Risk Management**

1. **Global Perspective:** In this, we review the bigger system description, design, and implementation. We look at the chance and the impact the risk is going to have.
2. **Take a forward-looking view:** Consider the threat which may appear in the future and create future plans for directing the next events.
3. **Open Communication:** This is to allow the free flow of communications between the client and the team members so that they have certainty about the risks.
4. **Integrated management:** In this method risk management is made an integral part of project management.
5. **Continuous process:** In this phase, the risks are tracked continuously throughout the risk management paradigm.



Risk Assessment

The objective of risk assessment is to division the risks in the condition of their loss, causing potential. For risk assessment, first, every risk should be rated in two methods:

* The possibility of a risk coming true (denoted as r).
* The consequence of the issues relates to that risk (denoted as s).

Based on these two methods, the priority of each risk can be estimated:

                  p = r \* s63.5Mo Java Program for Beginners

Where p is the priority with which the risk must be controlled, r is the probability of the risk becoming true, and s is the severity of loss caused due to the risk becoming true.

If all identified risks are set up, then the most likely and damaging risks can be controlled first, and more comprehensive risk abatement methods can be designed for these risks.

**1. Risk Identification:** The project organizer needs to anticipate the risk in the project as early as possible so that the impact of risk can be reduced by making effective risk management planning.

A project can be of use by a large variety of risk. To identify the significant risk, this might affect a project. It is necessary to categories into the different risk of classes.

There are different types of risks which can affect a software project:

1. **Technology risks:** Risks that assume from the software or hardware technologies that are used to develop the system.
2. **People risks:** Risks that are connected with the person in the development team.
3. **Organizational risks:** Risks that assume from the organizational environment where the software is being developed.
4. **Tools risks:** Risks that assume from the software tools and other support software used to create the system.
5. **Requirement risks:** Risks that assume from the changes to the customer requirement and the process of managing the requirements change.
6. **Estimation risks:** Risks that assume from the management estimates of the resources required to build the system

**2. Risk Analysis:** During the risk analysis process, you have to consider every identified risk and make a perception of the probability and seriousness of that risk.

There is no simple way to do this. You have to rely on your perception and experience of previous projects and the problems that arise in them.

It is not possible to make an exact, the numerical estimate of the probability and seriousness of each risk. Instead, you should authorize the risk to one of several bands:

1. The probability of the risk might be determined as very low (0-10%), low (10-25%), moderate (25-50%), high (50-75%) or very high (+75%).
2. The effect of the risk might be determined as catastrophic (threaten the survival of the plan), serious (would cause significant delays), tolerable (delays are within allowed contingency), or insignificant.

Risk Control

It is the process of managing risks to achieve desired outcomes. After all, the identified risks of a plan are determined; the project must be made to include the most harmful and the most likely risks. Different risks need different containment methods. In fact, most risks need ingenuity on the part of the project manager in tackling the risk.

**There are three main methods to plan for risk management:**

1. **Avoid the risk:** This may take several ways such as discussing with the client to change the requirements to decrease the scope of the work, giving incentives to the engineers to avoid the risk of human resources turnover, etc.
2. **Transfer the risk:** This method involves getting the risky element developed by a third party, buying insurance cover, etc.
3. **Risk reduction:** This means planning method to include the loss due to risk. For instance, if there is a risk that some key personnel might leave, new recruitment can be planned.

**Risk Leverage:** To choose between the various methods of handling risk, the project plan must consider the amount of controlling the risk and the corresponding reduction of risk. For this, the risk leverage of the various risks can be estimated.

Risk leverage is the variation in risk exposure divided by the amount of reducing the risk.

**Risk leverage = (risk exposure before reduction - risk exposure after reduction) / (cost of reduction)**

**1. Risk planning:** The risk planning method considers each of the key risks that have been identified and develop ways to maintain these risks.

For each of the risks, you have to think of the behavior that you may take to minimize the disruption to the plan if the issue identified in the risk occurs.

You also should think about data that you might need to collect while monitoring the plan so that issues can be anticipated.

Again, there is no easy process that can be followed for contingency planning. It rely on the judgment and experience of the project manager.

**2. Risk Monitoring:** Closely monitor the potential risks and their early symptoms. Also monitor the effects of steps taken to mitigate or avoid them.

**Project Execution & Monitoring**

In this phase, the tasks described in project plans are executed according to their schedules.

Execution needs monitoring in order to check whether everything is going according to the plan. Monitoring is observing to check the probability of risk and taking measures to address the risk or report the status of various tasks.

These measures include -

* **Activity Monitoring -**All activities scheduled within some task can be monitored on day-to-day basis. When all activities in a task are completed, it is considered as complete.
* **Status Reports -**The reports contain status of activities and tasks completed within a given time frame, generally a week. Status can be marked as finished, pending or work-in-progress etc.
* **Milestones Checklist -**Every project is divided into multiple phases where major tasks are performed (milestones) based on the phases of SDLC. This milestone checklist is prepared once every few weeks and reports the status of milestones.

Effective communication plays vital role in the success of a project. It bridges gaps between client and the organization, among the team members as well as other stake holders in the project such as hardware suppliers.

Communication can be oral or written. Communication management process may have the following steps:

* **Planning**- This step includes the identifications of all the stakeholders in the project and the mode of communication among them. It also considers if any additional communication facilities are required.
* **Sharing**- After determining various aspects of planning, manager focuses on sharing correct information with the correct person on correct time. This keeps every one involved the project up to date with project progress and its status.
* **Feedback**- Project managers use various measures and feedback mechanism and create status and performance reports. This mechanism ensures that input from various stakeholders is coming to the project manager as their feedback.
* **Closure**- At the end of each major event, end of a phase of SDLC or end of the project itself, administrative closure is formally announced to update every stakeholder by sending email, by distributing a hardcopy of document or by other mean of effective communication.

After closure, the team moves to next phase or project.

**Configuration Management**

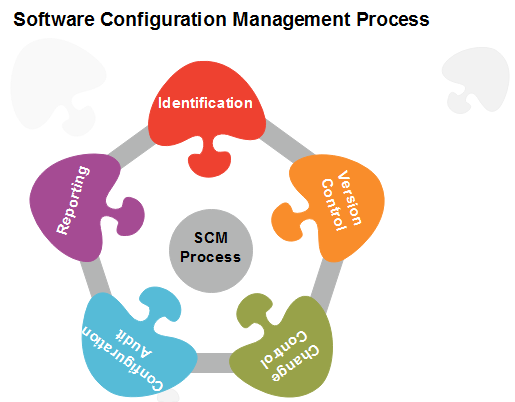
Configuration management is a process of tracking and controlling the changes in software in terms of the requirements, design, functions and development of the product.

Definition two “the process of identifying and defining the items in the system, controlling the change of these items throughout their life cycle, recording and reporting the status of items and change requests, and verifying the completeness and correctness of items”.

SCM Process

It uses the tools which keep that the necessary change has been implemented adequately to the appropriate component. The SCM process defines a number of tasks:

* Identification of objects in the software configuration
* Version Control
* Change Control
* Configuration Audit
* Status Reporting



1. **Identification**

**Basic Object:** Unit of Text created by a software engineer during analysis, design, code, or test.

**Aggregate Object:** A collection of essential objects and other aggregate objects. Design Specification is an aggregate object.

1. **Version Control**

Version Control combines procedures and tools to handle different version of configuration objects that are generated during the software process.

Configuration management allows a user to specify the alternative configuration of the software system through the selection of appropriate versions. This is supported by associating attributes with each software version, and then allowing a configuration to be specified [and constructed] by describing the set of desired attributes.

1. **Change Control**

We worry about change because a small confusion in the code can create a big failure in the product. But it can also fix a significant failure or enable incredible new capabilities.

We worry about change because a single rogue developer could sink the project, yet brilliant ideas originate in the mind of those rogues, and

A burdensome change control process could effectively discourage them from doing creative work.

A change request is submitted and calculated to assess technical merit; potential side effects, the overall impact on other configuration objects and system functions, and projected cost of the change.

The results of the evaluations are presented as a change report, which is used by a change control authority (CCA) - a person or a group who makes a final decision on the status and priority of the change.

The "check-in" and "check-out" process implements two necessary elements of change control-**access control** and **synchronization control**.

**Access Control** governs which software engineers have the authority to access and modify a particular configuration object.

**Synchronization Control** helps to ensure that parallel changes, performed by two different people, don't overwrite one another.

1. **Configuration Audit**

SCM audits to verify that the software product satisfies the baselines requirements and ensures that what is built and what is delivered.

SCM audits also ensure that traceability is maintained between all CIs and that all work requests are associated with one or more CI modification.

SCM audits are the "**watchdogs**" that ensures that the integrity of the project's scope is preserved.

1. **Status Reporting**

Configuration Status reporting (sometimes also called status accounting) providing accurate status and current configuration data to developers, testers, end users, customers and stakeholders through admin guides, user guides, FAQs, Release Notes, Installation Guide, Configuration Guide,

Configuration management is a discipline of organization administration, which takes care of occurrence of any change (process, requirement, technological, strategical etc.) after a phase is baselined. CM keeps check on any changes done in software.

**Change Control**

Change control is function of configuration management, which ensures that all changes made to software system are consistent and made as per organizational rules and regulations.

A change in the configuration of product goes through following steps -

* **Identification** - A change request arrives from either internal or external source. When change request is identified formally, it is properly documented.
* **Validation** - Validity of the change request is checked and its handling procedure is confirmed.
* **Analysis** - The impact of change request is analyzed in terms of schedule, cost and required efforts. Overall impact of the prospective change on system is analyzed.
* **Control** - If the prospective change either impacts too many entities in the system or it is unavoidable, it is mandatory to take approval of high authorities before change is incorporated into the system. It is decided if the change is worth incorporation or not. If it is not, change request is refused formally.
* **Execution** - If the previous phase determines to execute the change request, this phase take appropriate actions to execute the change, does a thorough revision if necessary.
* **Close request** - The change is verified for correct implementation and merging with the rest of the system. This newly incorporated change in the software is documented properly and the request is formally is closed.

**project scope management**

Project scope management is a process that helps in determining and documenting the list of all the project goals, tasks, deliverables, deadlines, and budgets as a part of the planning process. In [project management](https://kissflow.com/project/project-management-basics/), it is common for a big project to have modifications along the way.

**Project scope vs. product scope**

Project scope should not be confused with product scope. Product scope defines the capabilities, characteristics, features and functions of the deliverables at the end of the project.

Project leaders should create a separate product scope statement. They should use both the [project scope and the product scope](https://www.techtarget.com/searchsoftwarequality/feature/Compare-a-product-vs-project-mindset-for-software-development) statements to support each other and establish a clear understanding of what every project aims to achieve.

**Importance of project scope management**

For a [project manager](https://kissflow.com/project/roles-and-responsibilities-of-project-manager/), managing the expectations of the stakeholders and clients is one of the most challenging tasks. With a definite project scope, managers can easily stay on track and ensure that all the deadlines are being followed throughout the [project life cycle.](https://kissflow.com/project/five-phases-of-project-management/)

***A well-defined project scope management helps avoid common issues like:***

* Constantly changing requirements
* Pivoting the project direction when you are already mid-way
* Realizing that the final outcome isn’t what was expected
* Going over the discussed budget
* Falling behind the project deadlines

Effective project scope management gives a clear idea about the time, labor, and cost involved in the project. It helps to distinguish between what is needed and what isn’t needed for accomplishing the project. Scope in project management also establishes the control factors of the project to address elements that might change

**How is the project scope defined**

Project scope is a part of the [project planning process](https://kissflow.com/project/steps-to-create-successful-project-plan/) that documents specific goals, deliverables, features, and budgets. The scope document details the list of activities for the successful completion of the project.

The scope is defined by understanding the project requirements and the client’s expectations. The scope statement usually contains,

* [project objectives](https://kissflow.com/project/project-objectives/)
* [project deliverables](https://kissflow.com/project/project-deliverables/)
* exclusions
* [project constraints](https://kissflow.com/project/project-management-constraints/) and
* [project assumptions](https://kissflow.com/project/glossary/project-assumptions/).

**Scope statement in project management**

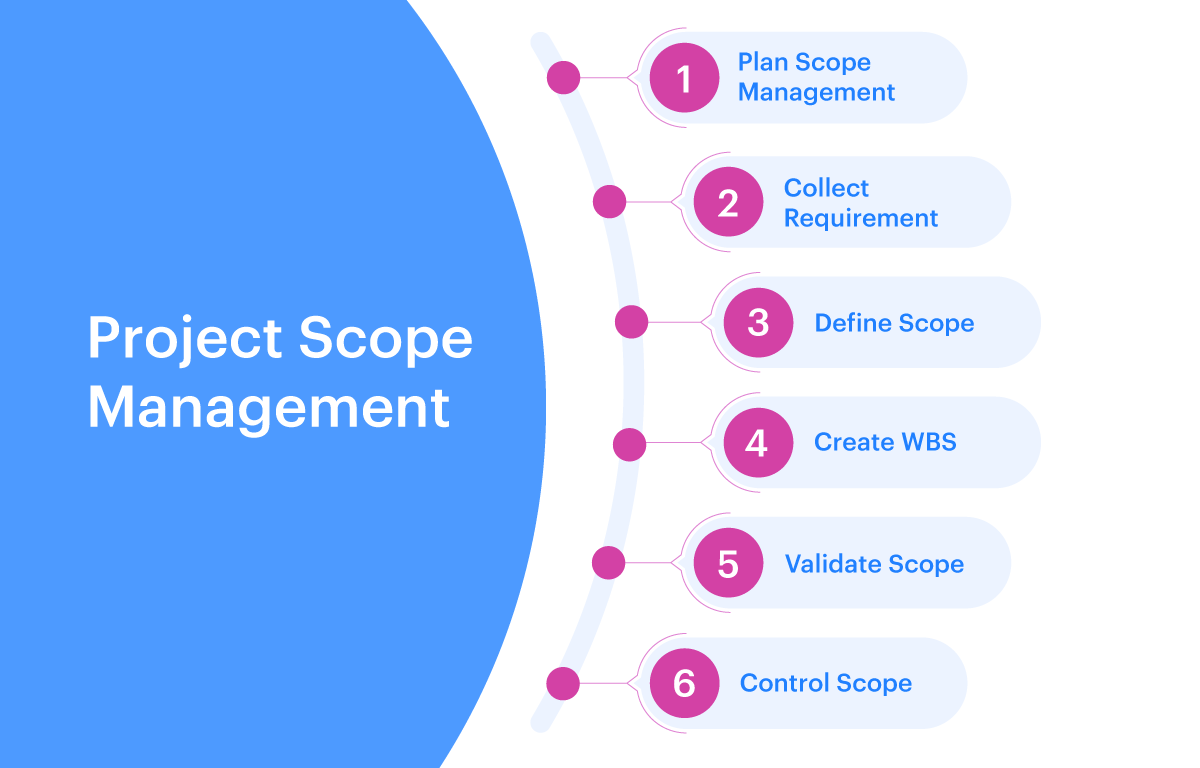
The project’s scope statement is also called its scope document or statement of work. The project scope statement

* details all the boundaries of the project while also establishing the responsibilities of the team,
* defines all the procedures that need to be followed for verifying and approving the finished work, and,
* gives team members a definitive guideline for making project-related decisions.

When documenting the scope of a project, team members and stakeholders have to be as specific as possible to avoid [scope creep](https://kissflow.com/project/avoid-scope-creep-in-project/), a situation where some parts of the project end up taking more time and effort than initially discussed due to miscommunication or poor planning.

**Project scope management process**

*Let’s discuss the six processes involved in accurately identifying the project scope management:*



1**. Planning scope management**

In the first process in project scope management, you create a scope plan document that you can refer to in the later stages. The document mainly helps in defining, managing, validating, and controlling the project’s scope.

**It includes:**

* Detailed project scope statement
* Breakdown of all the project requirements
* Expected project deliverables
* [Project change control process](https://kissflow.com/project/project-monitoring-and-controlling/)

The document doesn’t have to be very detailed, it just has to fit the purpose. You can also use a previous project’s scope management plan as a reference for this.

**2. Collecting requirements**

The next step is to work out stakeholder requirements and expectations. You will be required to document all the project requirements, expectations, budgets, and deliverables through interviews, surveys, and focus groups.

This is a rather important step because more often than not, stakeholders can have unrealistic requirements or expectations and the project managers would be required to step in to find a solution that is acceptable by everyone from avoiding [project delays](https://kissflow.com/project/how-to-avoid-project-delays/).

**At the end of the collection requirements stage, you should have the following:**

* Functional as well as non-functional requirements
* Stakeholder requirements
* Business requirements
* Support and training requirements
* Project requirements

**3. Defining the scope**

At this step, you need to turn your requirements into a well-detailed description of the service or product that you are trying to deliver through the project. You will then have a project scope statement that you can then refer to throughout your project.

While it is important to list what is in the scope of the project, it is just as important to note down what is out of the project scope. Any kind of inclusions to the scope would then have to go through the entire change control process to ensure the team is only working on things that they are supposed to work on.

With a defined scope, you get a reference point for your [project team](https://kissflow.com/project/project-team-management/) and anyone else involved. In case there is something that is not involved in the scope, it doesn’t need to be completed by the team.

4**. Making a project breakdown structure**

A [project breakdown structure](https://kissflow.com/project/glossary/work-breakdown-structure-wbs/) is a document that breaks down all the work which needs to be done in the project and then assigns all the tasks to the team members. It lists the deliverables that need to be completed and their respective deadlines as well.

You can use [project management software](https://kissflow.com/project/) for this step of the process to assign and [prioritize project tasks](https://kissflow.com/project/team/how-to-prioritize-project-tasks/) which will make it easier to track the entire progress of the project and avoid any unnecessary bottlenecks.

**5. Validating scope**

In this step, the scope and deliverables that you have recorded need to be sent to project executives and stakeholders to get the necessary approvals. Scope validation needs to be done before starting the project to ensure that if something goes wrong then it is easy to find where it went wrong.

**6. Controlling scope**

Project managers need to ensure that as the project begins, it always stays within the defined scope. In case there are some things that need to change, then the proper change control process should be followed.

*Project scope is defined in the scope statement, a document that provides the objectives, schedules, tasks and deliverables of a project. Scope statements align stakeholders' expectations and give projects a framework for success.*

**Software Crisis** is a term used in computer science for the difficulty of writing useful and efficient computer programs in the required time.

The software crisis was due to using the same workforce, same methods, same tools even though rapidly increasing in software demand, the complexity of software, and software challenges.

With the increase in the complexity of software, many software problems arise because existing methods were insufficient. If we will use the same workforce, same methods, and same tools after the fast increase in software demand, software complexity, and software challenges, then there arise some problems like software budget problems, software efficiency problems, software quality problems, software managing and delivering problem, etc. This condition is called a **software crisis**



**Causes of Software Crisis:**

* The cost of owning and maintaining software was as expensive as developing the software
* At that time Projects were running over-time
* At that time Software was very inefficient
* The quality of the software was low quality
* Software often did not meet user requirements
* The average software project overshoots its schedule by half
* At that time Software was never delivered
* Non-optimal resource utilization.
* Difficult to alter, debug, and enhance.
* The software complexity is harder to change.

Let’s now understand **which factors are contributing to the software crisis.**

* Poor project management.
* Lack of adequate training in software engineering.
* Less skilled project members.
* Low productivity improvements.

**Solution of Software Crisis:**

There is no single solution to the crisis. One possible solution to a software crisis is *Software Engineering* because software engineering is a systematic, disciplined, and quantifiable approach. For preventing software crises, there are some guidelines:

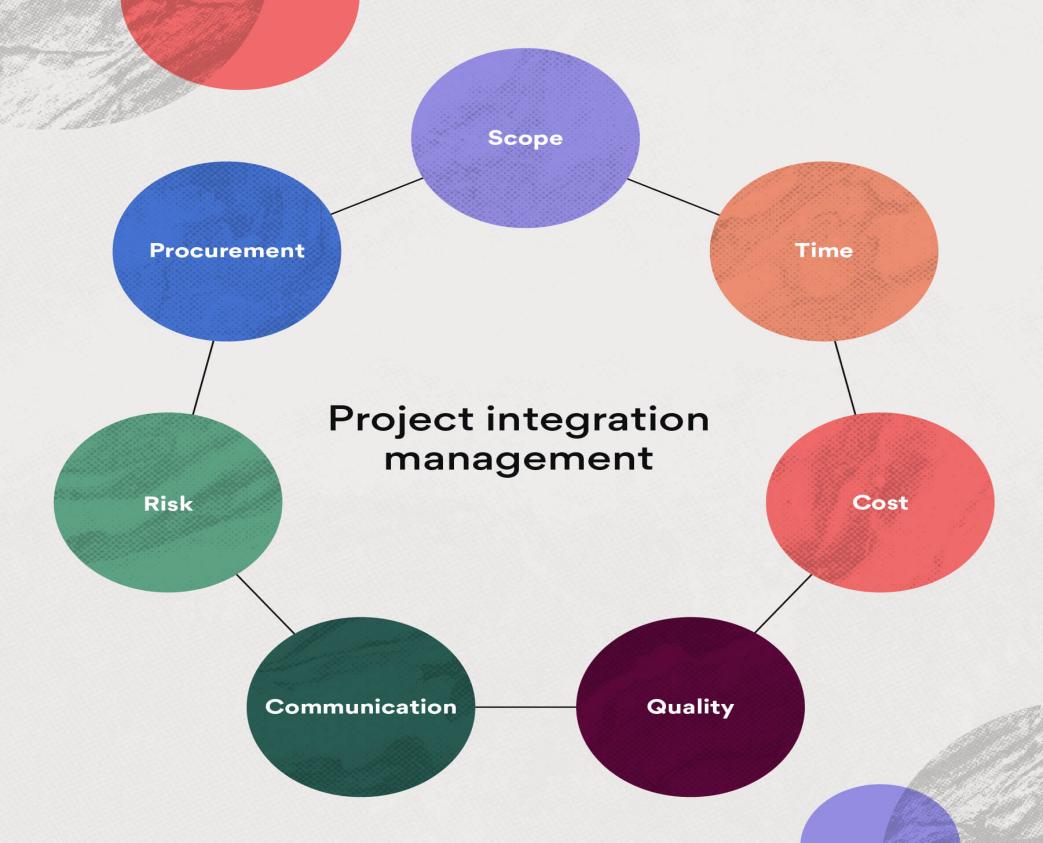
* Reduction in software over budget.
* The quality of software must be high.
* Less time is needed for a software project.
* Experienced and skilled people working over the software project.
* Software must be delivered.
* Software must meet user requirements.

**Integration Management**

It is to keep the complete interconnected project as one unified set of activities for effective execution of role that determine the success factors of the project.

Project integration management involves coordinating all elements of a project, including tasks, resources, stakeholders, and deliverables.

The purpose of project integration management is to ensure that processes run efficiently and meet predefined goals

.

**When do we need to integration Management?**

1. when there are interactions between different processes and teams.
2. When projects have competing objectives or scheduling conflicts
3. it helps one make cost or schedule trade-offs that enable the team to complete the project and meet stakeholder expectations
4. In general it helps in maintaining equilibrium in all areas of a project (time, scope, cost quality, human resource, communication, risk procurement, stakeholders etc.

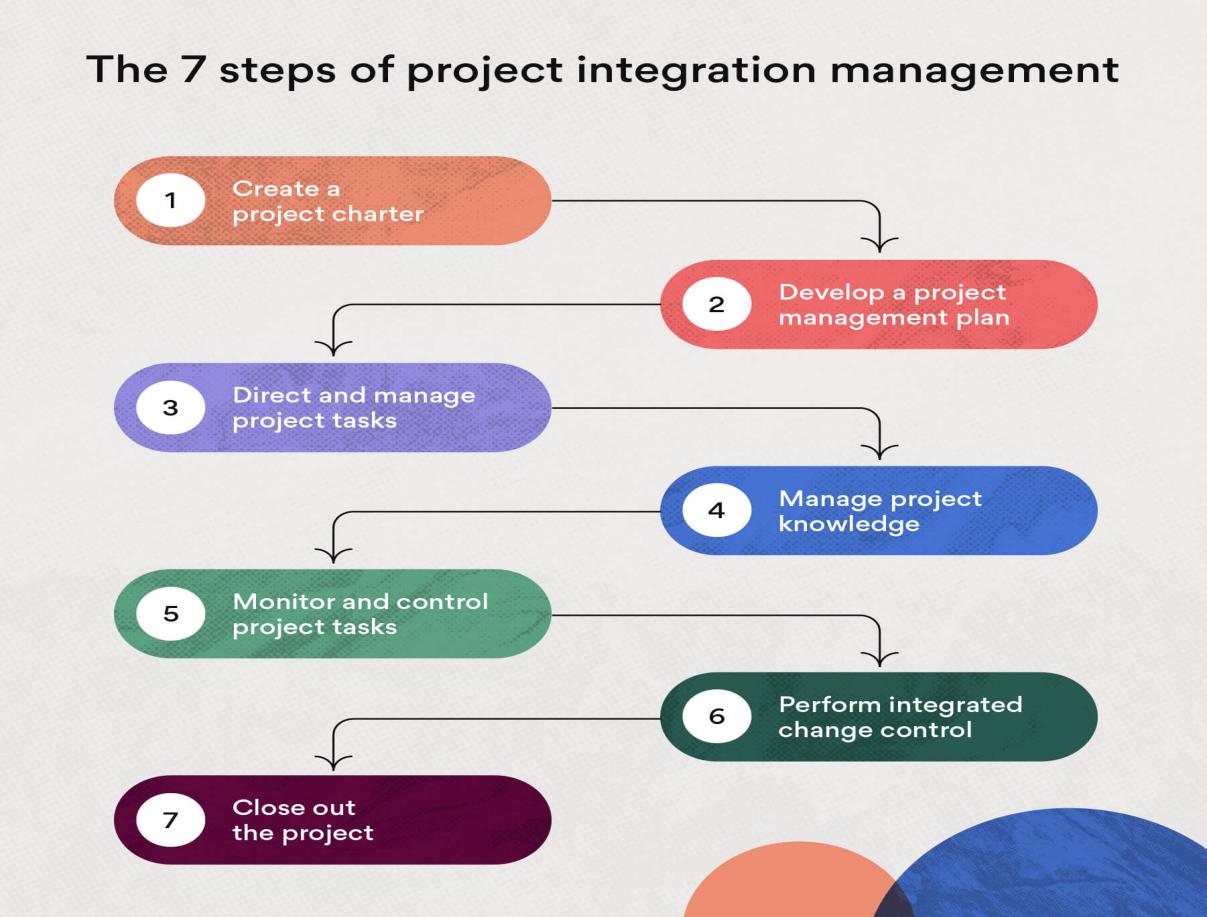
Example:

 Let’s say the marketing team requires data from the sales team in order to produce a report for stakeholders. Project integration management is used to ensure that the data from the sales team is properly handed off to the marketing team and that the deliverable meets requirements.

These five phases include:

1. **Project initiation:** The aim of this phase is to establish the vision and goals for the project and secure stakeholder approval through project objectives. This phase consists of creating the project charter to provide an overview of the project, a clear road map, and the stakeholder register to specify the stakeholders involved.
2. **Project planning:** The planning process is where you set up the project infrastructure to help you achieve the project goals within time, budget, and resource constraints.
3. **Project execution**: This is when you’ll put your project plan into action and get the project under way. Most of the budget will be spent during this phase to produce the deliverables. It also includes activities such as stakeholder engagement and communications, quality assurance, and team development.
4. **Project performance:** This phase involves supervising the progress of the project and comparing it against the original plan. It means taking corrective action when there are blockers or delays.
5. **Project closure**: This is where you formally close out the project by getting approval from the client or stakeholder. Records and lessons learned from the project should be archived for future reference.

**How do we integrate the project**



1. **Create project charter**

Projects typically start out with the creation of a project charter, a short document that provides an overview of the project, identifies the project manager and key stakeholders.

A project charter includes the following:

* Scope
* Objectives and deliverables
* Project team members
* Project risks
* Benefits or returns on investment
* Budget
* Business case

A project charter essentially acts as a foundation on which you can further plan your project. It also helps you gain buy-in from stakeholders, which enables you to move forward quickly and autonomously within the project scope.

1. **Develop project management plan**

The next step involves developing a more detailed project plan, which specifies the project scope statement, deliverables, time line, milestones, and metrics to evaluate success.

The project plan is used to direct the execution of the project to meet overall requirements and objectives.

Here are the steps to creating a project management methodology:

1. Meet with stakeholders to set project requirements, deliverables, and objectives.
2. Define the project scope.
3. Create a work breakdown structure (WBS) to delegate tasks and assign resources.
4. Create a project schedule.
5. Conduct a risk assessment and develop [contingency plans](https://asana.com/resources/contingency-plan).
6. Come up with a performance measurement baseline to assess project performance.
7. Develop additional plans for any of the following: scope management, cost management, resource management, change management, stakeholder management, or risk management.
8. **Direct and manage project work**

The next phase is project execution, in which the project manager takes charge of the day-to-day work that must be done, such as:

* Directing the project team
* Holding stakeholder meetings
* Tracking project progress

This phase ensures that tasks are being carried out effectively according to the project plan and scope statement.

1. **Manage project knowledge**

Project knowledge management refers to the process of using existing information or obtaining additional knowledge to reach project goals. This step ensures team members have all the information they need to produce the required deliverables.

Any knowledge or expertise gained during this step contributes to the company’s overall body of knowledge, which is useful for future strategic endeavours.

1. **Monitor and control project work**

The purpose of this step is to keep the project on track. If there are any deviations from the project plan, they need to be identified and corrected.

There are three approaches to this:

1. Preventive action: an action performed to reduce the negative impact of project risks
2. Corrective action: an action performed to bring the future project performance back in line with the project plan
3. Defect repair: an action to repair or replace a documented project defect

A common technique to measure performance is to do an earned-value analysis to assess the current status of the project’s budget and schedule. If project changes need to be made and are approved, project documents such as the work performance report may need to be updated.

Monitoring the project ensures that it is on track to meet objectives.

1. **Perform integrated change control**

Changes to projects can sometimes be stressful if not handled properly, but with a change control process in place they don’t have to be.

That’s why change requests must be assessed to ensure they don’t exceed the scope or approach scope creep, which refers to the increase in requirements during the project life cycle. Some companies even have a dedicated change control board to review change requests related to budget, time lines, and resources, for example.

An example of project change is if the client asks for additional assets, in which case you’ll need to evaluate the level of impact on the project.

1. **Close out the project**

After all project work is complete and deliverables are shipped and approved by the client, it’s time to close the project.

Project closure involves processes such as:

* Holding a final stakeholder meeting
* Conducting a formal review of the project
* Closing the contract
* Organizing and archiving project materials for future use

Project closure serves as a reference for future works and provides insight on how to improve the project integration management system.

**Benefit of integration**

1. Avoiding duplication of efforts
2. Making more effective use of senior management time.
3. Using resources to implement and manage systems in a more efficient manner.
4. Achieve more cost efficient certification.
5. Reducing audit fatigue

**Software Quality Assurance**

Software Quality Assurance, is a process or a role of a software engineer to make sure there is no slippage occurring in the software application with respect to the requirement provided by the customer.

The four phases of Software Quality Assurance are ***Plan, Do, Check and Act***, where **Plan** is to plan the measures necessary to keep the application standards in high quality, **Do** is to the development process that involves the build and testing processes, **Check** is to observe and examine the implementation routes, and **Act** is to act upon the activities required to maintain the application quality.

**Software Quality Assurance Activities**

Below given are some of the activities of Software Quality Assurance.

**1. Setting the Checkpoint**

SQA team sets the checkpoints after specific time intervals in order to check the progress, quality, performance of software, and whether the software quality work is done on time as per the schedule and documents.

**2. Measure Change Impact**

For a defect reported by QA and fixed by the developer, it is very important to retest [the defect](https://www.educba.com/what-is-defect/) fix and to verify whether the fixed defect does not introduce new defects in the working software. For this, test metrics are maintained and observed by managers and developers to check for newly generated defects by the introduction of new functionality or the fix of any defect.

**3. Having Multiple Testing Strategy**

One should not rely on a single testing approach and strategy for testing software. Multiple testing strategies should be implemented in software so as to test it from different angles and cover all the areas. For an e-commerce website, security testing, performance testing, load testing, database testing all should be done to ensure a better quality of software.

**4. Maintaining Records and Reports**

It is important to keep all the records and documents of the QA and share them on time to time to stakeholders. Test cases executed, test cycles, defects logged, defects fixed, test cases created, change in requirements from a client for a specific test case, all should be properly documented for future reference

**5. Managing Good Relations**

Managing good relations between the testers and developers plays an important role in the project. As the role of developer and tester contradict each other but this should not be taken on a personal level. The main aim of both teams should be the delivery of good quality projects with minimum risks of failure.

**6. SQA Management Plan**

This includes finding ways how the SQA will work in the new project in the most effective way. Think of SQA strategies, software engineering processes that could be implemented as per the project requirements, and the individual skills of team members.

**Components of SQA System**

SQA components can be classified into 6 classes:

**1. Pre-Project Components**

This assures that the commitment of the project has been defined clearly regarding the time estimation, clarification of customer requirements, total budget of the project, evaluation of development risks, total staff required for that particular project. It also assures that development and quality plans have been clearly defined.

**2. Software Project Life Cycle Components**

This component includes the review, expert opinions, software testing, software maintenance components. In the project development life cycle, it includes components like reviews, expert opinions, and finding defects in software design and programming, whereas in the software maintenance life cycle it includes specializing in maintenance components and development life cycle components for enhancing maintenance tasks.

**3. Infrastructure Components for Error Prevention and Improvements**

This component includes staff training, certification, configuration management, preventive and corrective measures in order to reduce the rate of errors in software based on the organization’s accumulated SQA experience.

**4. Management SQA Components**

This class includes software quality metrics, software quality costs, which includes control of maintenance and development activities, and the introduction of managerial involvement in order to reduce the risk of quality, schedule, and budget in the project.

**5. Components of Standardization, Certification, and SQA System Assessment**

The main objective of this class is the utilization of professional international knowledge, which helps in the coordination between the different organization quality systems at a professional level.

**6. Organizing for SQA Human Components**

This base includes managers, testers, and other SQA practitioners that are interested in SQA. The main objective is to support and initiate the SQA activities, detect the gaps/deviations in them, and suggest improvements for that.

**SQA Techniques**

There are several SQA techniques. Some of them are mentioned below:

**1. Reviewing**

In reviewing, a meeting is held by both internal and external stakeholders to review the whole project who analyses the whole software and if finds an issue, distinguishes whether it is testing, development, requirement or a design The main objective is to measure the quality of software and ensure that whether it meets the customer expectations or not.

**2. Auditing**

In Auditing, the whole work product and all the data are inspected by stakeholders to check whether it follows the standard processes or not.

**3. Functional Testing**

In functional testing, the functionality of the whole software is tested whether it is functioning as expected or not. It checks “what the system works” without knowing “how the system works”. It is [like the black box testing](https://www.educba.com/black-box-testing/) of an application in which the user knows the expected output without knowing how it is produced.

**4. Standardization**

It assures that everything in the software should be standardized, i.e. it follows all the standards either the standards in documentation, development, quality control. It reduces ambiguity and hence improves the quality of software.

**5. Code Inspection**

Code Inspection is one of the most formal kinds of review with the main objective of finding defects in the code and highlighting any issues in the Code Inspection is led by a trained Moderator rather than the Author of the code. Meeting has proper entry and exit criteria. Users must need complete preparation before the meeting in order to have complete knowledge of documents and all before raising their points.

**6. Walkthroughs**

Software walkthrough is a kind of informal process and usually, it is initiated by the Author to read the document or code and the peer members write down their suggestions or errors in it and submits them. It is not formally documented like Inspection and moderator is not necessary for the meeting. Its main objective is to know the status of code completed to date and collecting suggestions from peers for a better quality of software.

**7. Stress Testing**

Stress testing is done to check how the system works under heavy load. This testing plays an important role in software quality as in e-commerce applications, stress and load testing are done properly in order to test the capacity of the software (how many maximum numbers of users can access an application at a time).

**8. Design Inspection**

Design Inspection is done to check the various areas of software using the checklist like functional and interface design, conventions, general requirements and design, requirement traceability, logic, coupling, and cohesion.

**Advantages of SQA**

Let’s discuss the advantages of SQA.

**1.  Increases Client’s Confidence**

Proper quality check at different levels of software like review, Inspection, Auditing, etc and with the involvement of both internal and external stakeholder increases the confidence of clients in the submission of the Weekly reports of the defect and requirement metrics also helps a lot in assuring the client that the work is being done on time.

**2. SQA Saves Money**

Defects found in the early stage either in requirement gathering, code, testing are easy and cost-effective to Proper SQA done at several levels helps to reduce those risks as maximum defects have been uncovered and resolved in early stages and hence saves money to fix the faulty software after being presented to the client which can costs company’s reputation, users and clients too.

**3. Boost Customer Satisfaction**

Timely involvement of the [client in software development](https://www.educba.com/what-is-software-development/) and testing boost customer satisfaction that the quality software is being developed and as per the requirements and taking of suggestions in between consideration boosts customer satisfaction.

**4. Promotes Productivity and Efficiency**

When development and testing are done in parallel, defects found early just after the development of a single module is done and fixed by developers timely allows everyone to work in peace and in a more productive manner rather than be burdened with multiple bugs at once after the completion of the whole software.

**5. Prevents from Unforeseen Emergencies**

When developing corporate software, stakes are also very high. As the software deals with a lot of customer’s sensitive data, it needs to work as expected without any blackouts, corruption, or communication breakdowns. The software should be tested very rigorously so that it should work as expected.

**6. Reduces End Time Client Conflicts**

There are many cases found of disagreement of client and organizations later on regarding the change in requirements, time and budget fixed in the starting resulting in the cancellation of the project, money loss and bad impression of the company in the market (loss of client as it would create a bad reputation). In SQA everything is fixed at the starting of the project and documented properly without any ambiguity so that no conflicts would arise

**Disadvantages of SQA**

Let’s discuss the disadvantages of SQA.

**1. Sometimes Difficult to Implement**

As SQA defines all the activities and actions that should be taken at each step of software development in a very detailed manner, sometimes it becomes difficult to implement every single activity and process in development. So the person knows that it would be beneficial but focussing on each step in detail becomes difficult when working in large teams.

**2. Time Consuming**

Implementing each action in SQA is very time-consuming and sometimes it wastes more time in documentation and meetings rather than working on the actual development and testing of software.

**3. High Cost**

Through implementing SQA, though the cost of fixing the bugs in the later stages can be reduced by finding them and fixing in the starting only for the small projects with a low budget it is very difficult to implement SQA as the number of resources increases in the project so does the budget of a project. For small projects hiring the whole team of QA and implementing SQA cause a drastic increase in the cost of a project.

**Project management standards: CMM, ISO 12207**

**Project Management Tools**

The risk and uncertainty rises multifold with respect to the size of the project, even when the project is developed according to set methodologies.

There are tools available, which aid for effective project management. A few are described -

1. **Gantt Chart**

Gantt charts was devised by Henry Gantt (1917). It represents project schedule with respect to time periods. It is a horizontal bar chart with bars representing activities and time scheduled for the project activities.

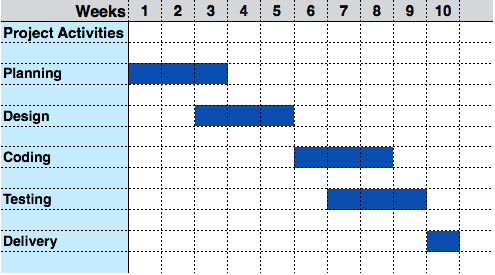
A [Gantt chart](https://www.ntaskmanager.com/blog/gantt-charts-planning-and-scheduling/) is a visual representation of all the tasks related to your project scheduled overtime. They are used to plan projects of all sizes and shapes.

Why? Because they are an incredible tool to show the work that is scheduled to be done on a project on a specific day. They also show the whole tenure of a particular project in one simple view.

Here are some of the features of a project you can monitor on a Gantt Chart.

1. The start and end dates of a project
2. What are the project tasks
3. Who are the team members involved in each project
4. Who is working on each individual task
5. What is the duration of each individual task
6. How all of the tasks are linked or are they dependent on each other

EXAMPLE



1. **Program Evaluation Review Technique (PERT) Chart**

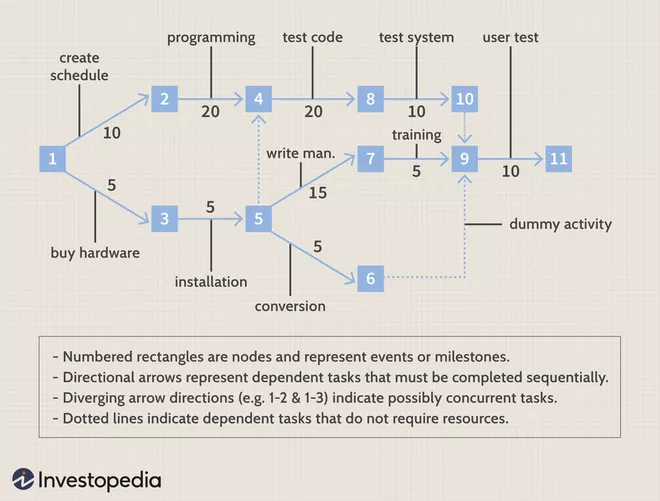
Program Evaluation Review Technique (PERT) is a project management planning tool used to calculate the amount of time it will take to realistically finish a project. PERT charts are used to plan tasks within a project — making it easier to schedule and coordinate team members.

PERT, you create three different time estimates for the project:

* The shortest possible amount of time each task will take
* The most probable amount of time
* The longest amount of time tasks might take if things don't go as planned

**PERT helps project planners identify:**

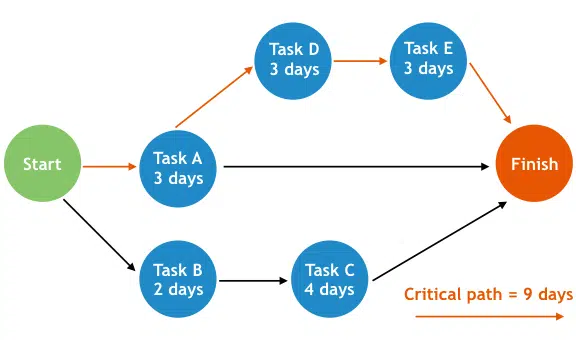
1. Start and end dates
2. Anticipated total required completion time
3. All activities, referred to as events on the chart, that impact the completion time
4. The required sequence of events
5. The probability of completion by a certain date



**Critical Path Analysis**

The Critical Path Method (CPM) is a simple but powerful technique for analyzing, planning, and scheduling large, complex projects. It is used to determine a project’s critical path—the longest sequence of tasks that must be finished for the entire project to be complete.

The events are arranged according to their earliest possible start time. Path between start and end node is critical path which cannot be further reduced and all events require to be executed in same order.



**Comparison Chart**

| **BASIS FOR COMPARISON** | **PERT** | **CPM** |
| --- | --- | --- |
| Meaning | PERT is a project management technique, used to manage uncertain activities of a project. | CPM is a statistical technique of project management that manages well defined activities of a project. |
| What is it? | A technique of planning and control of time. | A method to control cost and time. |

**3) Resource Histogram**

This is a graphical tool that contains bar or chart representing number of resources (usually skilled staff) required over time for a project event (or phase). Resource Histogram is an effective tool for staff planning and coordination.

