Introduction to Project Management

Introduction

Evaluates whether software project management differs significantly from managing other types of projects.

Focuses on planning, monitoring, and controlling software projects.

All projects, including software projects, aim to meet specific objectives and satisfy real needs.

Identifying the project's stakeholders and their objectives is crucial.

Ensuring that these objectives are met is the primary aim of project management.

Knowing the current state of the project is essential to predict whether it will meet its objectives in the future.

A project is a planned activity, according to dictionary definitions.

Why is Software Project Management Important?

Information and communication technology (ICT) is expensive.

One of the main causes of these project failures is poor project management. Although criticisms have been levelled at the companies, the broad impression of the frequency of ICT project failure remains.

Software Development Life Cycle

The Software Development life cycle is a methodology that also forms the framework for planning and controlling the creation, testing, and delivery of an information system.

The software development life cycle concept acts as the foundation for multiple different development and delivery methodologies, such as the Hardware development life -cycle and software development life -cycle .

While Hardware development life -cycle deal specially with hardware and Software development life -cycle deal with software, a systems development life - cycle differs from each in that it can deal with any combination of hardware and software , as a system can be composed of hardware only , software only, or a combination of both.

Four Project Dimensions:

People

Process

Product

Technology

The 5 Variables of Project Control

1. Time: amount of time required to complete the project.
2. Cost: calculated from the time variable
3. Quality: The amount of time put into individual tasks determines the overall quality of the project.
4. Scope: Requirements specified for the end result.
5. Risk: Potential points of failure.

What is a Project?

A project is defined as being planned if, for the most part, we can figure out how we are going to complete a task before we begin.

Certain exploratory ventures might find this to be particularly challenging.

Planning is just considering things carefully before moving forward, and it is still worthwhile to do so even for uncertain projects, provided that it is acknowledged that the final plans may contain speculative and preliminary aspects.

Certain tasks, like regular maintenance, may have been completed so frequently that all parties involved are aware of exactly what has to be done.

Planning barely appears necessary in these situations, but procedures may need to be documented to make sure.

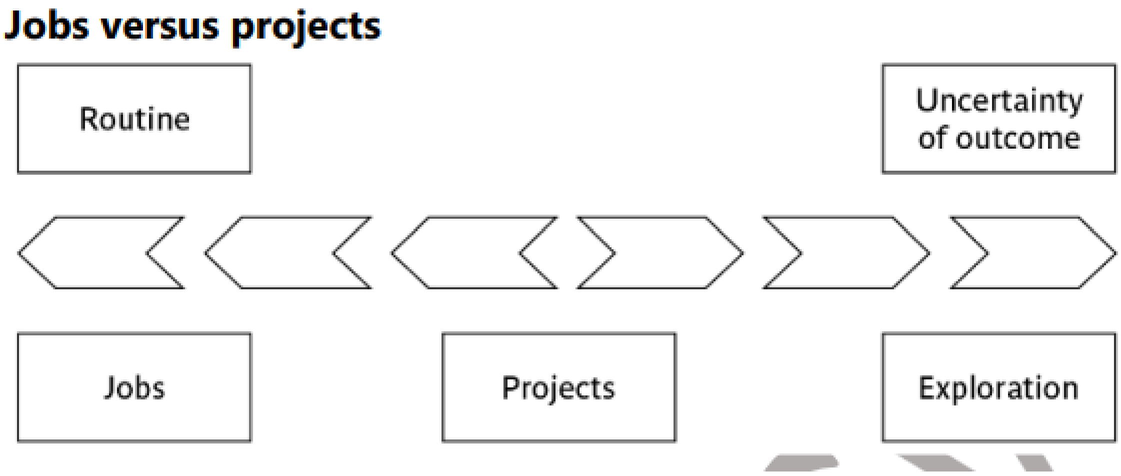
Definitions of ‘project’ include:

- A specific plan or design
- A planned undertaking
- A large undertaking e.g. a public works scheme” Key points above are planning and size of task Here are some definitions of ‘project’.

No doubt there are other ones: for example, ‘Unique process, consisting of a set of coordinated and controlled activities with start and finish dates, undertaken to achieve an objective conforming to specific requirements, including constraints of time, cost and resources.

There is a hazy boundary between the non-routine project and the routine job. The first time you do a routine task, it will be like a project.

On the other hand, a project to develop a system similar to previous ones you have developed will have a large element of the routine.



Characteristics of Project

- Non-routine tasks are involved
- Planning is required
- Specific objectives are to be met or a specified product is to be created
- The project has a pre-determined time span
- Work is carried out for someone other than yourself
- Work involves several specialisms
- Work is carried out in several phases
- The resources that are available for use on the project are constrained
- The project is large or complex.

The project that employs 20 developers is likely to be disproportionately more difficult than one with only 20 staff because of the need for additional coordination.

Software Projects versus Other Types of Project: Many of the techniques of general project management are applicable to software project management. One way of perceiving software project management is as the process of making visible that which is invisible.

Invisibility: When a physical artefact such as a bridge or road is being constructed the progress being made can actually be seen. With software, progress is not immediately visible.

Complexity: Software products contain more complexity than other engineered artifacts. **Conformity:** The 'traditional' engineer is usually working with physical systems and physical materials like cement and steel. These physical systems can have some complexity, but are governed by physical laws that are consistent. Software developers have to conform to the requirements of human clients. It is not just that individuals can be inconsistent.

Flexibility: The ease with which software can be changed is usually seen as one of its strengths. However, this means that where the software system interfaces with a physical or

organizational system, it is expected that, where necessary, the software will change to accommodate the other components rather than vice versa.
This means the software systems are likely to be subject to a high degree of change.

Contract Management

ln-house projects are where the users and the developers of new software work for the same organization.

However, increasingly organizations contract out ICT development to outside developers. Here, the client organization will often appoint a 'project manager' to supervise the contract who will delegate many technically oriented decisions to the contractors.

Thus, the project manager will not worry about estimating the effort needed to write individual software components as long as the overall project is within budget and on time.

On the supplier side, there will need to be project managers who deal with the more technical issues.

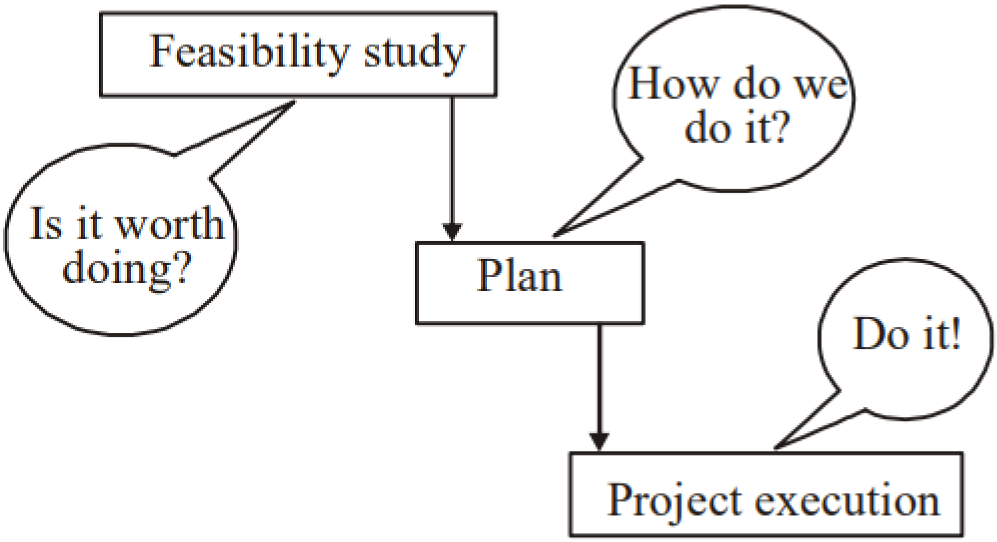
Contract management is the process of managing the creation, execution, and analysis of contracts to maximize operational and financial performance and minimize risk.

It involves various activities from the initial request for a contract, through negotiation, execution, compliance, and renewal.

Effective contract management ensures that all parties to a contract fulfill their obligations as efficiently as possible.

Activities Covered by Software Project Management

The activities covered by Software Project management are diagrammatically illustrated as follows:



The

Feasibility Study: This study aims to determine if a potential project has a strong enough business case to warrant further development.

Data on the specifications of the suggested application are obtained.

The predicted costs include the expected development and operating expenses as well as the benefits of the new system's worth.
The study might be a component of a strategic planning process that looks at and ranks several possible software advancements.

Planning: Project planning can begin if the feasibility assessment yields results that suggest the potential project seems feasible.

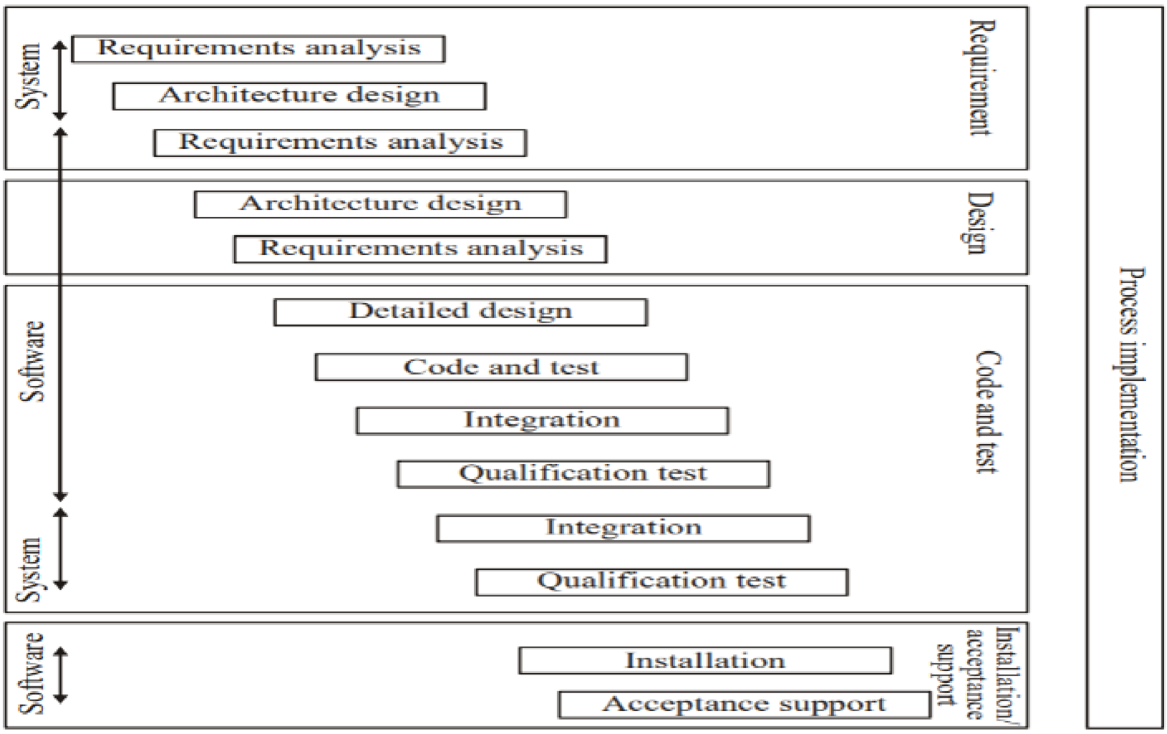
On the other hand, we wouldn't start a big project with all of our meticulous planning done at once. We would create a broad project plan and a more specific one for the initial phase.

The next stages would be planned in more detail as they got closer.

This is because closer to the beginning of the later stages, we would have more precise and in-depth information on which to base our preparations.

Carrying Out Projects: Now, the job can be carried out. Design and implementation phases are frequently included in the project execution process.

The international standard ISO 12207 recommends a typical sequence of software development operations, which is depicted in below figure



Analysis of Requirements: This begins with requirement elicitation, also known as requirement gathering, which determines what the project's consumers need from the system.

It is highly likely that some work in this direction was done during the project evaluation; however, the initial data must now be updated and expanded upon.

Specification: Detailed documentation of what the proposed system is to do.

Design: A design has to be drawn up which meets the specification. This design will be in two stages.

One will be the external or user design concerned with the external appearance of the application.

The other produces the physical design which tackles the way that the data and software procedures are to be structured internally.

> **Architecture Design**: This maps the requirements to the components of the system that is to be built. At the system level, decisions will need to be made about which processes in the new system will be carried out by the user and which can be computerized.
>
> This design of the system architecture thus forms an input to the development of the software requirements.

Detailed Design: Each software component is made up of a number of software units that can be separately coded and tested.

The detailed design of these units is carried out separately.

Coding: This may refer to writing code in a procedural language or an object-oriented language or could refer to the use of an application-builder. Even where software is not being built from scratch, some modification to the base package could be required to meet the needs of the new application.

Testing (Verification and Validation): Whether software is developed specially for the current application or not, careful testing will be needed to check that the proposed system meets its requirements.

Integration: The individual components are collected together and tested to see if they meet the overall requirements. Integration could be at the level of software where different software components are combined, or at the level of the system as a whole where the software and other components of the system such as the hardware platforms and networks and the user procedures are brought together.

Qualification Testing: The system, including the software components, has to be tested carefully to ensure that all the requirements have been fulfilled.

Implementation/ Installation: Some system development practitioners refer to the whole of the project after design as 'implementation' (that is, the implementation of the design) while others insist that the term refers to the installation of the system after the software has been developed.

Acceptance Support: Once the system has been implemented there is a continuing need for the correction of any errors that may have crept into the system and for extensions and improvements to the system. Maintenance and support activities may be seen as a series of minor software projects.

Plans, Methods and Methodologies

A plan for an activity must be based on some idea of a method of work. To take a simple example, if you were asked to test some software, even though you do not know anything about the software to be tested, you could assume that you would need to:

Analyse the requirements for the software

Devise and write test cases that will check that each requirement has been satisfied

Create test scripts and expected results for each test case

Compare the actual results and the expected results and identify discrepancies

While a method relates to a type of activity in general, a plan takes that method (and perhaps others) and converts it to real activities, identifying for each activity:

Its start and end dates

Who will carry it out?

What tools and materials will be used?

‘Materials’ in this context could include information, for example a requirements document.

Some ways of categorizing Software Projects

Distinguishing different types of projects is important as different types of tasks need different project approaches e.g.

Changes to the characteristics of software projects

Voluntary systems (such as computer games) versus compulsory systems e.g. the order processing system in an organization

Information systems versus embedded systems.

Software Products verses services

Product-development versus outsourced.

Object-driven development.

Compulsory versus Voluntary users

In workplaces there are systems that staff have to use if they want to do something, such as recording a sale.

However, use of a system is increasingly voluntary, as in the case of computer games.

Here it is difficult to elicit precise requirements from potential users as we could with a business system.

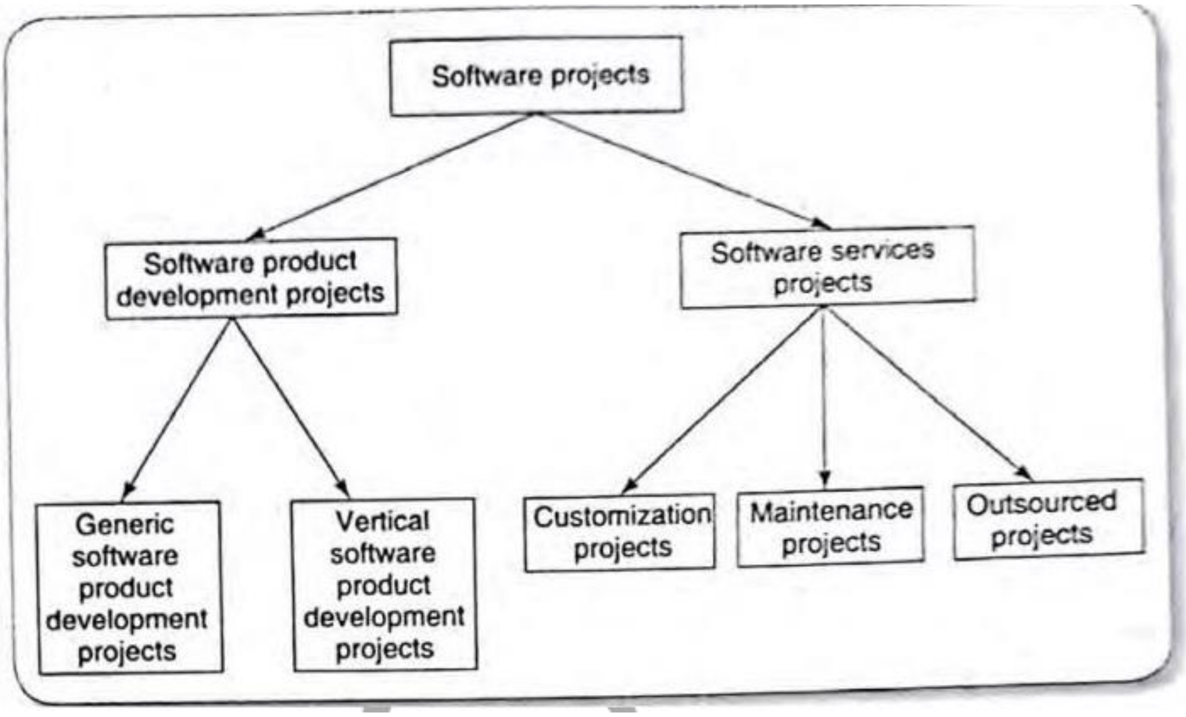
What the game will do will thus depend much on the informed ingenuity of the developers, along with techniques such as market surveys, focus groups and prototype evaluation.

Information Systems versus Embedded systems

A traditional distinction has been between information systems which enable staff to carry out office processes and embedded systems which control machines. A stock control system would be an information system. An embedded, or process control, system might control the air conditioning equipment in a building. Some systems may have elements of both where, for example, the stock control system also controls an automated warehouse.

Software Products verses Services

All types of software projects can broadly be classified into software product development projects and software services projects. It can be further classified as shown in below figure


Software projects
Software product development projects
Software services projects
Generic software product development projects
Vertical software product development projects
Customization projects
Maintenance projects
Outsourced projects

A software product development concerns developing the software by keeping the requirements to the general customers in mind and developed software is usually sold-off-the shelf to a large number of customers.

Software services cover a large gamut of software projects such as customization, outsourcing, maintenance, testing and consultancy. Projects may be distinguished by whether their aim is to produce a product or to meet certain objectives.

Outsourced Projects While developing a large project, it makes good commercial sense for a company to outsource some parts of its work to other companies. For example, A company may consider outsourcing as a good option, it if feels that it does not have sufficient expertise to develop some specific parts of the product or if it determines that some parts can be developed cost-effectively another company.

Object-driven development

Projects may be distinguished by whether their aim is to produce or to meet certain objective.

Stakeholders

These are people who have a stake or interest in the project.

It is important that they be identified as early as possible, because you need to set up adequate communication channels with them right from the start.

The project leader also has to be aware that not everybody who is involved with a project has the same motivation and objectives.

The end-users might, for instance, be concerned about the ease of use of the system while their managers might be interested in the staff savings the new system will allow.

Boehm and Ross proposed a ‘Theory W’ of software project management where the manager concentrates on creating the role and format situations where all parties benefit from a project and therefore have an of communication interest in its success. (The 'W' stands for 'win-win'.)

Stakeholders might be internal to the project team, external to the project team but in the same organization, or totally external to the organization.

Internal to the project team: This means that they will be under the direct managerial control of the project leader.

External to the project team but within the same organization: For example, the project leader might need the assistance of the information management group in order to add some additional data types to a database or the assistance of the users to carry out systems testing. Here the commitment of the people involved has to be negotiated.

External to both the project team and the organization: External stakeholders may be customers (or users) who will benefit from the system that the project implements or contractors who will carry out work for the project.

Setting Objectives

The objectives should define what the project team must achieve for project success. Objectives focus on the desired outcomes of the project rather than the tasks within it-they are the ‘post-conditions’ of the project.

Objectives could be set of statements following the opening words ‘the project will be a success if ….’

To have a successful software project, the manager and the project team members must know what will constitute success.

This will make them concentrate on what is essential to project success.

There may be several sets of users of a system and there may be several different groups of specialists involved its development.

There is a need for well-defined objectives that are accepted by all these people. Where there is more than one user group, a project authority needs to be identified which has overall authority over what the project is to achieve.

This authority is often held by a *project steering committee* (or project board or project management board) which has overall responsibility for setting, monitoring and modifying objectives.

The project manager still has responsibility for running the project on a day-to-day basis, but has to report to the steering committee at regular intervals. Only the steering committee can authorize changes to the project objectives and resources.

Sub-objectives and Goals

Setting objectives can guide and motivate individuals and groups of staff. An effective objective for an individual must be something that is within the control of that individual.

An objective might be that the software application to be produced must pay for itself by reducing staff costs over two years.

For software developers it would be unreasonable as, though they can control development costs, any reduction in operational staff costs depends not just on them but on the operational management after the application has ‘gone live’.

What would be appropriate would be to set a goal or sub-objective for the software developers to keep development costs within a certain budget.

Thus, objectives will need be broken down into goals or sub-objectives.

The mnemonic SMART is sometimes used to describe well defined objectives:

Specific: Effective objectives are concrete and well defined. Vague aspirations such as ‘to improve customer relations’ are unsatisfactory. Objectives should be defined in such a way that it is obvious to all whether the project has been successful or not.

Measurable: Ideally there should be measures of effectiveness which tell us how successful the project has been. The measure can, in some cases, be an answer to simple yes/no questions, e.g. ‘Can we install the new software by 1 November 2011?’

Achievable: It must be within the power of the individual or group to achieve the objective.

Relevant: The objective must be relevant to the true purpose of the project.

Time constrained: There should be a defined point in time by which the objective should have been achieved.

Measures of effectiveness

Measures of effectiveness provide practical methods of ascertaining whether an objective has been met.

Mean time between failures’ (mtbf) is used to measure reliability.

A measure of effectiveness will usually be related to the installed operational system.

Business Case

Most projects need to have a justification or business case: the effort and expense of pushing the project through must be seen to be worthwhile in terms of the benefits that will eventually be felt.

The quantification of benefits will often require the formulation of a business model which explains how the new application can generate the claimed benefits.

Any project plan must ensure that the business case is kept intact. For example:

The development costs are not allowed to rise to a level which threatens to exceed the value of benefits.

The features of the system are not reduced to a level where the expected benefits cannot be realized.

The delivery date is not delayed so that there is an unacceptable loss benefit.

Project Success and Failure

The project plan should be designed to ensure project success preserving the business case for the project.

Different stakeholders have different interests, some stakeholders in a project might see it as a success while others do not.

The project objectives are the targets that the project team is expected to achieve—They are summarized as delivering:

The agreed functionality

To the required level of quality

In time

Within budget

A project could meet these targets but the application, once delivered could fail to meet the business case.

A computer game could be delivered on time and within budget, but might then not sell.

In business terms, the project is a success if the value of benefits exceeds the costs.

A project can be a success on delivery but then be a business failure, On the other hand, a project could be late and over budget, but its deliverables could still, over time, generate benefits that outweigh the initial expenditure.

The possible gap between project and business concerns can be reduced by having a broader view of projects that includes business issues.

Technical learning will increase costs on the earlier projects, but later projects benefit as the learnt technologies can be deployed more quickly cheaply and accurately.

Customer relationships can also be built up over a number of projects. If a client has trust in a supplier who has

Management and Management Control

Management

Management involves following activities:

Planning - deciding what is to be done

Organizing - making arrangements;

Staffing - selecting the right people for the job etc.;

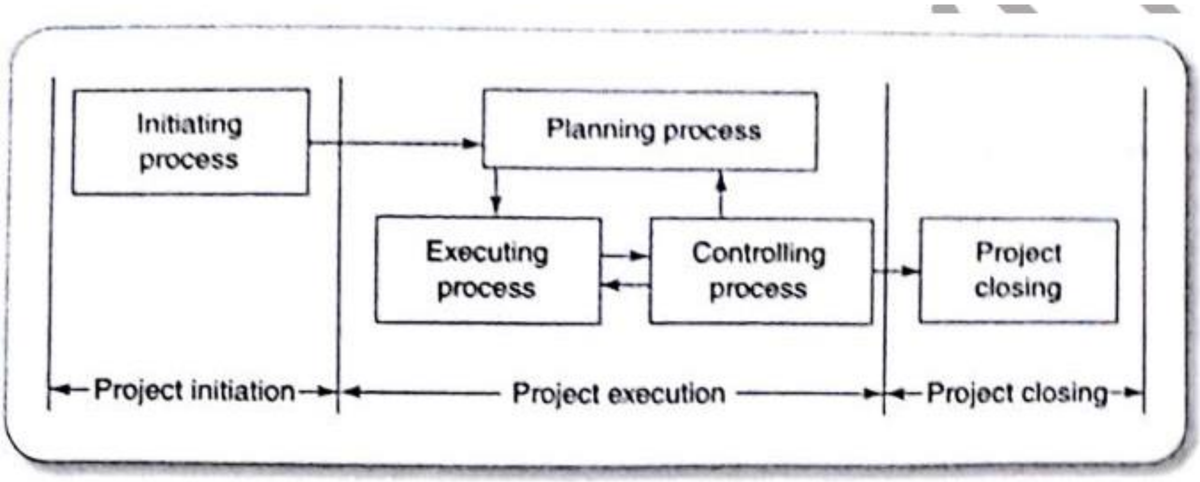
Directing - giving instructions;

Monitoring - checking on progress;

Controlling - taking action to remedy hold-ups;

Innovating - coming up with new solutions;

Representing - liaising with clients, users, developer, suppliers and other stakeholders.



Much of the project manager’s time is spent only in three activities , i.e. Project Planning , Monitoring and control.

This time period during which these activities are carried out is indicated in above figure.

It shows that project management is carried out over three well-defined stages or processes irrespective of the methodology used.

In the Project initiation stage, an initial plan is made. As a project starts, the project is monitored and controlled to process as planned. Initial plan is revised periodically to accommodate additional details and constraints about the project as they become available.

Finally, the project is closed.

Initial project is undertaken immediately after the feasibility study phase and before starting the requirement analysis and specification process.

Initial project planning involves estimating several characteristics of a project. Based on these estimates all subsequent project activities are planned.

The monitoring activity involves monitoring the progress of the project. Control activities are initiated to minimize any significant variation in the plan.

Project Planning is an important responsibility of the project Manager. During project planning, the project manager needs to perform a few well-defined activities that have been outlined below.

“A Gude to the Project Management Body of knowledge, is used.

Estimation: The following project attributes are estimated.

Cost: How much is it going to cost to complete the project.

Duration: How long is it going to take to complete the project.

Effort: How much effort would be necessary for completing the project?

The effectiveness of all activities such as scheduling and staffing are planned at later stage.

Scheduling: Based on estimations of effort and duration, the schedules for manpower and other resources are developed.

Staffing: Staff organization and staffing plans are made.

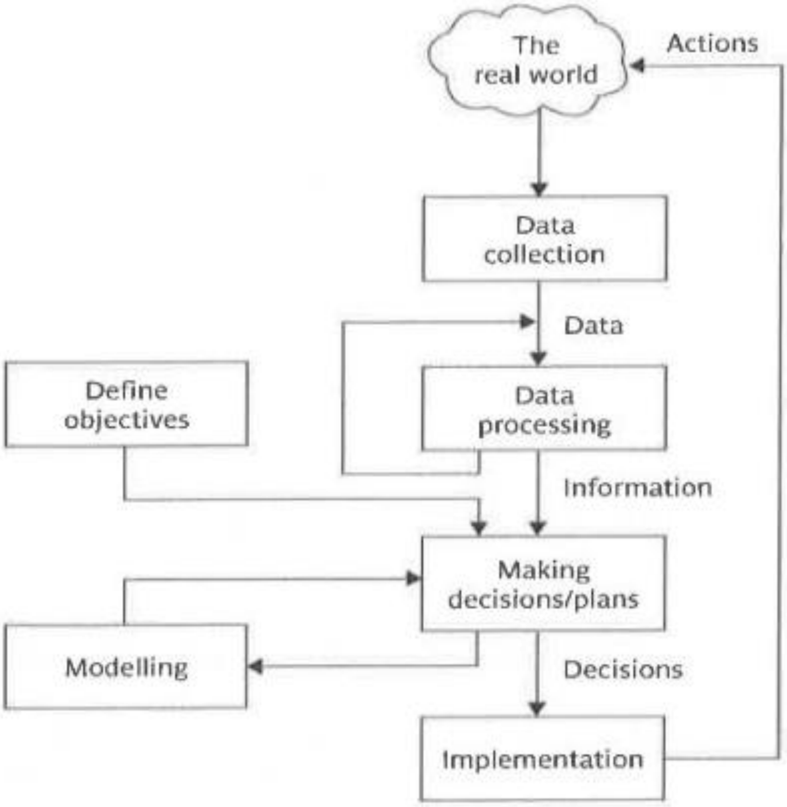
Risk Management: This activity includes risk identification, analysis, and abatement planning.

Miscellaneous Plans: This includes making several other plans such as quality assurance plan, configuration management plan etc.

While carrying out project monitoring and control activities, a project manager may sometimes find it necessary to change the plan to cope with specific situations and make the plan more accurate as more project data becomes available.

Management Control

Management involves setting objectives for a system and monitoring the performance of the system.



In the above Fig, local mangers involve in data collection. Bare details such as “location X has processed 2000 documents” may not be useful to higher management.

Data processing is required to transform this raw data into useful information. This might be in such forms as “Percentage of records Processed”, average documents per day per person”, and estimated completion date”.

In this example , the project management might examine the “estimated completion date” for completing data transfer for each branch. They are comparing actual performance with overall project objectives.

They might find that one or two branches will fail to complete the transfer of details in time.

It can be seen that a project plan is dynamic and will need constant adjustment during the execution of the project.

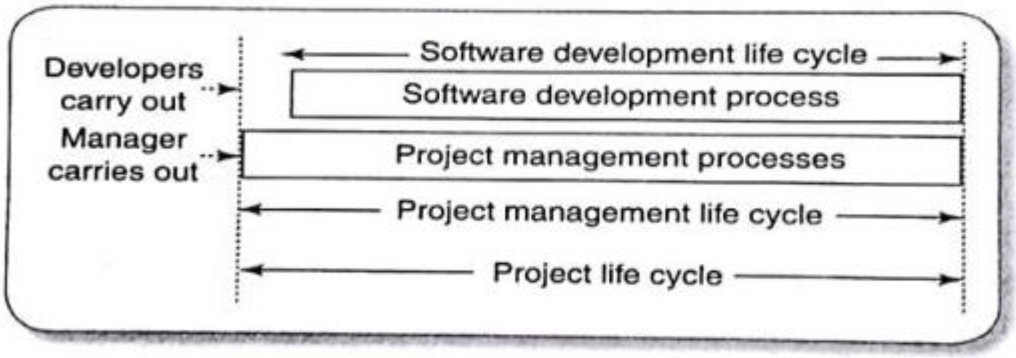
A good plan provides a foundation for a good project, but is nothing without intelligent execution.

Project Management Life Cycle

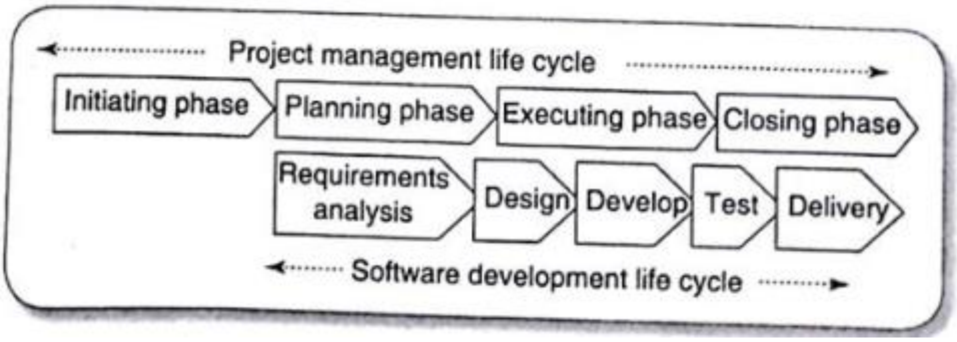
Software development life cycle denotes (SDLC) the stages through which a software is developed.

In contrast to SDLC, the project management life cycle typically starts well before the software development activities start and continues for the entire duration of SDLC.

In Project Management process, the project manager carries out project initiation, planning, execution, monitoring, controlling and closing.


Developers carry out
Manager carries out
Software development life cycle
Software development process
Project management processes
Project management life cycle
Project life cycle

The different phases of the project management life cycle are shown in figure below.



Project Initiation: The project initiation phase starts with project concept development. During concept development the different characteristics of the software to be developed are thoroughly understood, which includes, the scope of the project, the project constraints, the cost that would be incurred and the benefits that would accrue.

Based on this understanding, a feasibility study is undertaken to determine the project would be financially and technically feasible.

Based on feasibility study, the business case is developed. Once the top management agrees to the business case, the project manager is appointed, the project charter is written and finally project team is formed.

This sets the ground for the manager to start the project planning phase.

W5HH Principle: Barry Boehm, summarized the questions that need to be asked and answered in order to have an understanding of these project characteristics.

Why is the software being built?

What will be done?

When will it be done?

Who is responsible for a function?

Where are they organizationally located?

How will the job be done technically and managerially?

How much of these each resource is needed.

Project Bidding: Once the top management is convinced by the business case, the project charter is developed. For some categories of projects, it may be necessary to have formal bidding process to select suitable vendor based on some cost-performance criteria.

The different types of bidding techniques are:

Request for quotation (RFQ) : An organization advertises an RFQ if it has good understanding of the project and the possible solutions.

Request for Proposal (RFP): An organization had reasonable understanding of the problem to be solved, however, it does not have good grasp of the solution aspects. i.e. may not have sufficient knowledge about different features to be implemented.

The purpose of RFP is to get an understanding of the alternative solutions possible that can be deployed and not vendor selection.

Based on the RFP process, the requesting organization can form a clear idea of the project solutions required, based on which it can form a statement work (SOW) for requesting RFQ for the vendors.

Request for Information (RFI): An organization soliciting bids may publish an RFI. Based on the vendor response to the RFI, the organization can assess the competencies of the vendors and shortlist the vendors who can bid for the work.

Project Planning: An importance of the project initiation phase is the project charter.

During the project planning the project manger carries out several processes and creates the following documents:

Project plan: This document identifies the project the project tasks and a schedule for the project tasks that assigns project resources and time frames to the tasks.

Resource Plan: It lists the resources , manpower and equipment that would be required to execute the project.

Functional Plan: It documents the plan for manpower, equipment and other costs.

Quality Plan: Plan of quality targets and control plans are included in this document.

Risk Plan: This document lists the identification of the potential risks, their prioritization and a plan for the actions that would be taken to contain the different risks.

Project Execution: In this phase the tasks are executed as per the project plan developed during the planning phase. Quality of the deliverables is ensured through execution of proper processes. Once all the deliverables are produced and accepted by the customer, the project execution phase completes and the project closure phase starts.

Project Closure: Project closure involves completing the release of all the required deliverables to the customer along with the necessary documentation.

All the Project resources are released and supply agreements with the vendors are terminated and all the pending payments are completed. Finally, a post implementation review is undertaken to analyze the project performance and to list the lessons for use in future projects.

Traditional Versus Modern Management Practices

Over the last two decades, the basic approach taken by the software industry to develop software has undergone a radical change.

Software is not developed from scratch any more, Software development projects are based on either tailoring some existing product or reusing certain pre-built libraries both will maximize code reuse and compression of project durations.

Other goals include facilitating and accommodating client feedback and client feedbacks and customer participation in project development work and incremental delivery of the product with evolving functionality.

Some Important difference between modern management practices and traditional practices are:

Planning Incremental Delivery: Earlier, projects were simpler and therefore more predictable than the present-day projects. In those days, projects were planned with sufficient detail much before the actual project execution started.

After the project initiation, monitoring and control activities were carried out to ensure that the project execution proceeded as per plan, Now, the projects are required to be completed over a much shorter duration, and rapid application development and deployment are considered key strategies.

Instead of making a long-term project completion plan, the project manager now plans all incremental deliveries with evolving functionalities.

This type of project management is often called extreme project management. Extreme project management is highly flexible approach that concentrates on human side of project management (e.g. managing project stakeholders).

Quality Management: Customer awareness about product quality has increased significantly. The key responsibility of a project manager now includes assessment of project progress and tracking the quality of all intermediate artefacts.

Change Management: Earlier, when the requirements were signed off by the customer, any changes to the requirements were rarely entertained. Customer suggestions are now actively solicited and incorporated throughout the development process. To facilitate customer feedback, incremental delivery models are popularly being used. Product development is being carried out through a series of product versions implementing increasingly greater functionalities. The Project manager plays a key role in product base lining and version control.

This has made change management a crucial responsibility of the project manager. Change Management is also known as configuration management.

Requirement Management: In older development methodologies, the requirements had to be identified upfront and these were ‘signed off’ by the customer and frozen before the development could start. At present, in most projects, the requirements change frequently during the development cycle. Requirement management has therefore become a systematic process of controlling changes, documenting , analyzing, tracing, prioritizing requirements and then communicating the changes to the relevant stakeholders.

Release Management: Release management concerns planning, prioritizing and controlling the different releases of a software. Modern development processes such as Agile development processes advocate frequent and regular releases of the software to be made to the customer during the software development. Starting with the release of basic or core functionalities of the software, more complete functionalities are made available to the customer every couple of weeks. Hence effective Release Management has become important.

Risk Management: In modern software development practices. Effective risk management is considered very important to the success of a project. A risk is any negative situation that may arise as the project progresses and may threaten the success of the project. Risk Management involves identification of risks, assessment of the impacts of various risks, prioritization of the risks and preparation of risk-containment plans.

Scope Management: Modern software development encourages customer to come up with change requests. While accepting the requests, three critical project parameters: scope , schedule and project cost are interdependent and related.