PROJECT

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:

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

A project is a temporary endeavor(try, work, effort, venture etc.) designed to produce a unique product, service, or result with a defined beginning and end (usually time-constrained, and often constrained by funding or staffing) undertaken to meet unique goals and objectives, typically to bring about beneficial change or added value

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.

MANAGEMENT

The best definition of **Management** refers to the optimal way to accomplish tasks and achieve goals, using Planning, Organizing, Staffing, Directing, and Controlling functions or processes **Management** (or **managing**) is the administration of an organization, whether it is a business, a not-for-profit organization, or government body.

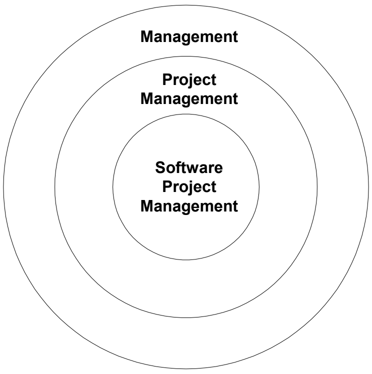
Management includes the activities of setting the strategy of an organization and coordinating the efforts of its employees (or of volunteers) to accomplish its objectives through the application of available resources, such as financial, natural, technological, and human resources

PROJECT MANAGEMENT

Project management is the process of leading the work of a team to achieve goals and meet success criteria at a specified time. The primary challenge of project management is to achieve all of the project goals within the given constraints.[1] This information is usually described in project documentation, created at the beginning of the development process. The primary constraints are scope, time, budget.[2] The secondary challenge is to optimize the allocation of necessary inputs and apply them to meet pre-defined objectives.

Project management, then, is the application of knowledge, skills, tools, and techniques to project activities to meet the project requirements.

SOFTWARE PROJECT MANAGEMENT



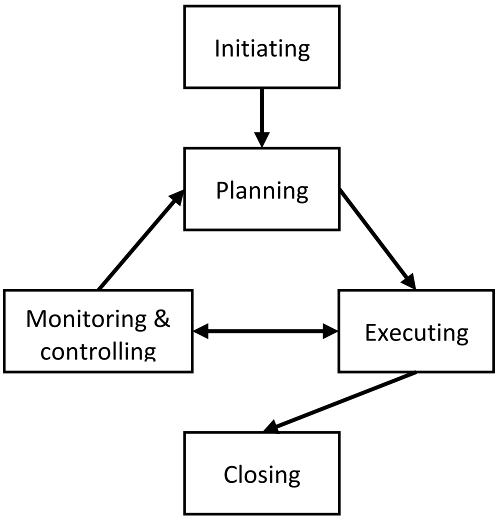
Is an art and science of planning and leading software projects. It is a sub-discipline of project management in which software projects are planned, implemented, monitored and controlled.

Software project management refers to the branch of project management dedicated to the planning, scheduling, resource allocation, execution, tracking and delivery of software and web projects.

Project management in software engineering is distinct from traditional project management in that software projects have a unique lifecycle process that requires multiple rounds of testing, updating, and customer feedback. Most IT-related projects are managed in the agile style, in order to keep up with the increasing pace of business, and iterate based on customer and stakeholder feedback.

PROJECT MANAGEMENT PROCESS GROUPS

Project Management Process Group is a logical grouping of project management processes to achieve specific project objectives. Process Groups are independent of project phases. Project management processes are grouped into the following five Project Management Process Groups:



1. **Initiating Process Group:**
   Those processes performed to define a new project or a new phase of an existing project by obtaining authorization to start the project or phase.

2. **Planning Process Group**:
   Those processes required to establish the scope of the project, refine the objectives, and define the course of action required to attain the objectives that the project was undertaken to achieve.

3. **Executing Process Group:**
   Those processes performed to complete the work defined in the project management plan to satisfy the project requirements.

4. **Monitoring and Controlling Process Group**:
   Those processes required to track, review, and regulate the progress and performance of the project; identify any areas in which changes to the plan are required; and initiate the corresponding changes.

5. **Closing Process Group**: Those processes performed to formally complete or close the project, phase, or contract.

PROJECT MANAGEMENT KNOWLEDGE AREAS

In addition to Process Groups, processes are also categorized by Knowledge Areas. A Knowledge Area is an identified area of project management defined by its knowledge requirements and described in terms of its component processes, practices, inputs, outputs, tools, and techniques. Although the Knowledge Areas are interrelated, they are defined separately from the project management perspective.

1. **Project Integration Management**. Includes the processes and activities to identify, define, combine, unify, and coordinate the various processes and project management activities within the Project Management Process Groups.

2. **Project Scope Management**. Includes the processes required to ensure the project includes all the work required, and only the work required, to complete the project successfully.

3. **Project Schedule Management.** Includes the processes required to manage the timely completion of the project.

4. **Project Cost Management**. Includes the processes involved in planning, estimating, budgeting, financing, funding, managing, and controlling costs so the project can be completed within the approved budget.

5. **Project Quality Management**. Includes the processes for incorporating the organization's quality policy regarding planning, managing, and controlling project and product quality requirements, in order to meet stakeholders' expectations.

6. **Project Resource Management**. Includes the processes to identify, acquire, and manage the resources needed for the successful completion of the project.

7. **Project Communications Management** Includes the processes required to ensure timely and appropriate planning, collection, creation, distribution, storage, retrieval, management, control, monitoring, and ultimate disposition of project information.

8. **Project Risk Management.** Includes the processes of conducting risk management planning, identification, analysis, response planning, response implementation, and monitoring risk on a Project.

9. **Project Procurement Management.** Includes the processes necessary to purchase or acquire products, services, or results needed from outside the project team.

10. **Project Stakeholder Management**. Includes the processes required to identify the people, groups, or organizations that could impact or be impacted by the project, to analyze stakeholder expectations and their impact on the project, and to develop appropriate management strategies for effectively engaging stakeholders in project decisions and execution.

FOUR Ps IN SOFTWARE PROJECT MANAGEMENT

For a successful project, there's a very important concept that we all should know in software project planning while developing a software product. There are 4 critical components in software project planning which are known as the 4P's namely:

1. **People**

   The most important component of a product and its successful implementation is human resources. This is virtually any type of stakeholder who is involved in the project. In building a proper product, a well-managed team with clear-cut roles defined for each person/team will lead to the success of the product. We need to have a good team in order to save our time, cost, and effort. Some assigned roles in software project planning are **project manager, team leaders, stakeholders, analysts,** and other **IT professionals**. Managing people successfully is a tricky process which a good project manager can do.

2. **Product**

   As the name inferred, this is the deliverable or the result of the project. The product doesn't necessarily refer to a physical product. It is simply whatever represents the goal of the project. The project manager should clearly define the product scope to ensure a successful result, control the team members, as well technical hurdles that he or she may encounter during the building of a product. The product can consist of both tangible or intangible such as shifting the company to a new place or getting a new software in a company.

   The resources, materials, quality, and everything else that defines the development of the project belong to the P of product. Having a profound understanding of all the elements related to the product will help the project manager to define and control the deliverables throughout the project life cycle.

3. **Process**

   In every planning, a clearly defined process is the key to the success of any product. This involves the exact tasks and activities that need to be completed, as well as putting a specific project management methodology in place. It regulates how the team will go about its development in the respective time period. The Process has several steps involved like, documentation phase, implementation phase, deployment phase, and interaction phase.

4. **Project**

   In this phase, the project manager plays a critical role. The project manager coordinates and controls the whole project through its life cycle, maintaining an overview at all times. Closely monitoring everything, the project manager makes sure that all team members have the data and resources they need to keep the project moving forward. He is responsible to guide the team members to achieve the project's target and objectives, helping & assisting them with issues, checking on cost and budget, and making sure that the project stays on track with the given deadlines.

   Ultimately, the P for project can be defined as the structure or framework that is governed by the project manager in order to achieve the project's objectives.

ROLE OF THE PROJECT MANAGER (From Book)

The project manager is the person assigned by the performing organization to lead the team responsible for achieving the project objectives. The project manager's reporting relationships are based on the organizational structure and project governance.

In addition to any specific technical skills and general management proficiencies required for the project, project managers should have at least the following attributes:

- Knowledge about project management, the business environment, technical aspects, and other information needed to manage the project effectively;
- Skills needed to effectively lead the project team, coordinate the work, collaborate with stakeholders, solve problems, and make decisions;
- Abilities to develop and manage scope, schedules, budgets, resources, risks, plans, presentations, and reports; and
- Other attributes required to successfully manage the project, such as personality, attitude, ethics, and leadership.

Project managers accomplish work through the project team and other stakeholders. Project managers rely on important interpersonal skills, including, but not limited to:

- Leadership
- Team building
- Motivating
- Communicating
- Influencing
- Decision making
- Political and cultural awareness
- Negotiating
- Facilitating
- Managing conflict
- Coaching

The project manager is successful when the project objectives have been achieved. Another aspect of success is stakeholder satisfaction. The project manager should address stakeholder needs, concerns and expectations to satisfy relevant stakeholders. To be successful, the project manager should tailor the project approach, life cycle, and project management processes to meet the project and product requirements.

AREAS OF EXPERTISE

AREAS OF EXPERTISE
Application Knowledge, standards & regulations
Understanding the Project Environment
Management Knowledge and skills
Interpersonal Skills

Areas of Expertise a project manager should bring to the project team

APPLICATION KNOWLEDGE; STANDARDS AND REGULATIONS

- By standards, we mean guidelines or preferred approaches that are not necessarily mandatory.
- In contrast, when referring to regulations we mean mandatory rules that must be followed such as Government imposed requirements through laws.
- It should go without saying that as a professional, you're required to follow all applicable laws and rules that apply to your industry, organization, or project.
- It will not only help the project to unfold smoothly, but will also allow for effective risk analysis.
- Some projects require specific skills in certain application areas. These application areas are usually concerned with disciplines, regulations and the specific needs of the project, the customer, or the industry You need to stay up-to-date regarding your industry so that you can apply your knowledge effectively.

UNDERSTANDING THE PROJECT ENVIRONMENT

There are many factors that need to be understood within your project environment. At one level you need to understand your project environment by thinking in terms of the cultural and the social environment. In this region we think of people, demographics and education. The international and political environment is where you need to understand about different countries cultural influences. Then we move on to the physical environment; here we think about time zones, think about different countries and how differently your project will be executed whether it is just in your country or whether you have an international project team that is distributed throughout the world in five different countries.

PROJECT ENVIRONMENT	
Cultural	Social
International	Political
Physical	

The important factors to consider within the project environment

Of all the factors the physical ones are the easiest to understand, and it is the cultural and international factors that are often misunderstood or ignored. How we deal with clients, customers, or project members from other countries can be critical to the success of the project. For example, the culture of the United States values accomplishments and individualism. Americans tend to be informal and call each other by first names, even if having just met. Europeans tend to be more formal, using surnames instead of first names in a business setting, even if they know each other well. In addition, their communication style is more formal than in the US, and while they tend to value individualism, they also value history, hierarchy, and loyalty. The Japanese, on the other hand, tend to communicate indirectly and consider themselves part of a group, not as individuals. The Japanese value hard work and success, as most of us do.

How a product is received can be very dependent on the international cultural differences. For example, in the nineties, when many large American and European telecommunications companies were cultivating new markets in Asia, their customer's cultural differences often produced unexpected situations. Western companies planned their telephone systems to work the same way in Asia as they did in Europe and America. But the protocol of conversation was different. Call-waiting, a popular feature in the West, is considered impolite in some parts of Asia. This cultural blunder could have been avoided had the team captured the project environment requirements and involved the customer.

It is often the simplest things that can cause trouble since unsurprisingly in different countries people do things differently. One of the most notorious examples of this is also one of the most simple: date formats. What day and month is 2/8/2009? Of course it depends where you come from; in North America it is February 8th while in Europe (and much of the rest of the world) it is 2nd August. Clearly, when schedules and deadlines are being defined it is important that everyone is clear on the format used.

The diversity of practices and cultures and its impact on products in general and on software in particular, goes well beyond the date issue. You may be managing a project to create a new website for a company that sells products worldwide. There are language and presentation style issues to take into consideration; converting the site into different languages isn't enough. It is obvious to ensure that the translation is correct, however, the presentation layer will have its own set of requirements for different cultures. The left side of a web site may be the first focus of attention for an American; the right side would be the initial focus for anyone from the Middle East, as both Arabic and Hebrew are written from right to left. Colors also have different meanings in different cultures. White, which is a sign of purity in America (e.g., a bride's wedding dress), and thus would be a favored background color in North America, signifies death in Japan (e.g., a burial shroud). Table summarizes different meanings of common colors.

Project managers in multicultural projects must appreciate the culture dimensions and try to learn relevant customs, courtesies, and business protocols before taking responsibility for managing an international project. A project manager must take into consideration these various cultural influences and how they may affect the project's completion, schedule, scope and cost.

MANAGEMENT KNOWLEDGE AND SKILLS

In this area we are thinking of items like your ability to plan the project, to execute the project properly and of course to control the project and bring it to a successful conclusion with the ability to guide the project team while achieving project objectives and balancing the project constraints.

There is more to project management than just getting the work done. Inherent to the process of project management are the general management skills that allow the project manager to complete the project with some level of efficiency and control. In some respects, managing a project is similar to running a business: there are risk and rewards, finance and accounting activities, human resource issues, time management, stress management, and a purpose for the project to exist.

INTERPERSONAL SKILLS

Last but not least you also have to bring the ability onto the project to manage personal relationships as well as dealing with issues as they arise. Here we are talking about interpersonal skills

INTERPERSONAL SKILLS	
Communication	Influence
Leadership	Motivation
Negotiation	Problem Solving

Communication:

- Project managers must be good communicators, promoting clear unambiguous exchange of information.
- As a project manager, it is your job to keep a number of people well informed.
- If you are uncertain of what the customer expects of you, then the project will not even get off the ground.

Influence:

- Project management is about getting things done.
- Every organization is different in its policies, modes of operations and underlying culture. There are political alliances, differing motivations, confecting interest, and power struggles within every organization.
- A project manager must understand all of the unspoken influences at work within an organization.

Leadership:

- Leadership is the ability to motivate and inspire individuals to work towards expected results.
- Leaders inspire vision and rally people around common goals.
- A good project manager can motivate and inspire the project team to see the vision and value of the project.

- The project manager as a leader can inspire the project team to find a solution to overcome the perceived obstacles to get the work done.

Motivation:

- Motivation helps people work more efficiently and produce better results.
- Motivation is a constant process that the project manager must have to help the team move towards completion with passion and a profound reason to complete the work.
- Motivating the team is accomplished by using a variety of team building techniques and exercises. Team building is simply getting a diverse group of people to work together in the most efficient and effective manner possible.
- Recognition and rewards are an important part of team motivations.

Negotiation:

- Project managers must negotiate for the good of the project.
- In any project, the project manager, the project sponsor, and the project team will have to negotiate with stakeholders, vendors, and customers to reach a level of agreement acceptable to all parties involved in the negotiation process.

Problem Solving:

- Problem solving is the ability to understand the heart of a problem, look for a viable solution, and then make a decision to implement that solution.
- The premise for problem solving is problem definition. Problem definition is the ability to understand the cause and effect of the problem; this centers on root cause analysis.
- If a project manager treats only the symptoms of a problem rather than its cause, the symptoms will perpetuate and continue through the project life.
- Root cause analysis looks beyond the immediate symptoms to the cause of the symptoms, which then affords opportunities for solutions.
- Once the root of a problem has been identified, a decision must be made to effectively address the problem.
- Solutions can be presented from vendors, the project team, the project manager or various stakeholders.

PROJECT STAKEHOLDERS

A stakeholder is an individual, group, or organization that may affect, be affected by, or perceive itself to be affected by a decision, activity, or outcome of a project. Project stakeholders may be internal or external to the project, they may be actively involved, passively involved, or unaware of the project. Project stakeholders may have a positive or negative impact on the project, or be positively or negatively impacted by the project.

STAKEHOLDER VS KEY STAKEHOLDERS

Project stakeholders in general can be single individual or organizations who are affected by the execution or outcome of project. It does not matter the project affects them positively or negatively

Key stakeholders are those that have influence or authority to dictate whether a project is successful or not. These are people whose objective must be satisfied as they have the power to make or break the project. Even if all the deliverables in the budget are met, if the stakeholder is not happy, the project cannot be considered successful.

Examples of stakeholders include but are not limited to

INTERNAL STAKEHOLDERS

Internal stakeholders are people whose interest in a company comes through a direct relationship, such as employment, ownership, or investment. The project directly impacts them as they serve and are employed by the organization managing it.

They include top management, project team members, co-workers, peers and internal customers.

Sponsor:

- A project sponsor is a person or group who owns the project and provides resources and support for the project, program or portfolio in order to enable its success.
- Every project has at least one project sponsor. They are the reason for the project. While they don’t manage the day-to-day operations of a project, they are above the project manager in terms of the project hierarchy. Most likely, the project sponsor has been involved with the project from the very beginning. They were the ones who helped conceive it and advocated for it.
- They design criteria for project success.
- They aligns project with business strategy, goals and objectives.

Resource Manager

A resource manager is the one who helps project managers with planning and allocating resources for a project; determines an organization’s demand for resources and ensures its capacity to meet staffing needs of projects; assigns employees to a project’s tasks; and participates in the hiring process.

Project Management Office:

- A project management office (PMO) is a group or department that defines, maintains and ensures project management standards across an organization.

- A project management office keeps documentation on projects and offers direction and key metrics in the execution of the projects under its governance.
- It leads the development team.

Portfolio Steering Committee:

- An advisory group providing guidance on key decisions which includes the sponsor, executives, and key stakeholders from the organization.
- One of the key tasks for the portfolio steering committee is to decide which projects and programmes are undertaken and which priority every component of the portfolio has.

Program manager:

A program manager acts as a coordinator between multiple projects at a business or organization to be sure they're benefiting each other and aligning with overall business goals. They are different from project managers because they do not directly oversee individual projects.

Team Members:

- They Performs the actual work of project under the PM including development, testing etc.
- Developers as stakeholders are responsible for quality product delivered on time and under estimation.
- Based on experience, developers can provides stakeholders with advice on business idea implementation.

EXTERNAL STAKEHOLDERS

External stakeholders are those people who are outside of the organization and are indirectly impacted by the project. They are not a part of your organization, might include government entities, contractors, suppliers.

Customers:

- Customers are the stakeholders for whom project is being built.
- Customers determine the main requirements and project scope and signs contract with main project performers.
- They always interact with development team, approve or supplement the plan with new implementation points.
- They hold the power to accept or reject the work.
- Project manager will need to negotiate, clarify, document project specifications and deliverables with them. After the project begins, the project manager must stay tuned to the customers concern and issues and keep them informed about the progress

End Users:

- Every product has its target audience, whose needs and desires affect the design and functionality of the system.
- Users can be engaged as stakeholders, in testing product beta version to provide initial feedback.
- They point out missing features and contribute to user experience improvement.

Suppliers:

- There are times when organizations don't have expertise or resource available in house and work is framed out to contractors or sub-contractors.
- They can be network consultant, construction management foremen, architect or anyone who is not an employee but working for the project.
- Many goods are provided by suppliers , any module so, the cost, schedule, quality of the product or resource provided by the third part must be managed.

Shareholders:

Someone who will get some portion incase of success of project . People who have invested money or resources in the project.

Regulatory Bodies:

The state, represented by regulatory authorities can also be a stakeholder. Regulators adopt international standards that influence the development of a software product and impose fines for non-compliance with the rules.

Competitors:

Competitors always implement new features and create industry trends affecting the market. Thus, your competitors influence the prioritization of tasks creating new unexpected challenges to respond to.

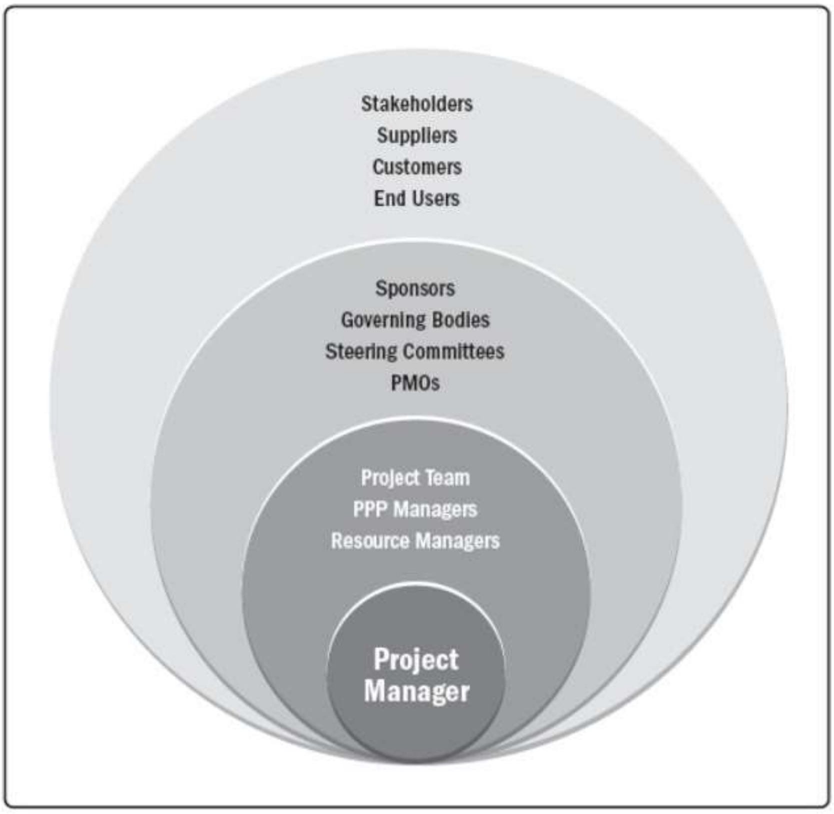


Figure 1-4. Examples of Project Stakeholders

A CIRCULAR WORKING MODE

Traditional models of project planning assume a linear working mode, proposing a stepwise procedure for planning with pre-determined phases. Activities of one phase are supposed to be completed before entering the subsequent phase. Such an approach can only be employed in well-known project environments with little uncertainty and for specific, well-defined types of projects.

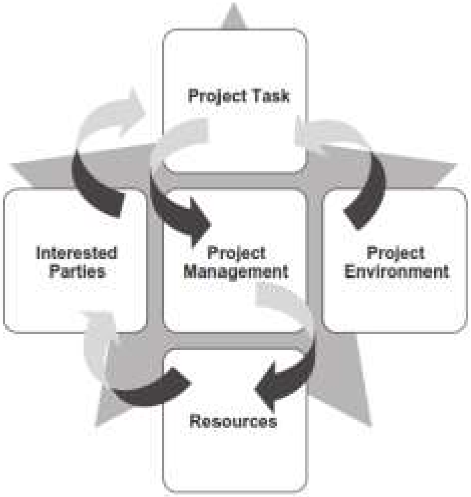


Figure 1.2. The Planning Process is Circular.

However, most projects face a great deal of uncertainties, especially at the beginning of forming a project. In addition, the complex nature of the project task suggests developing an overall vision of a new project and the way it will be carried out. This calls for a different approach than the linear working mode.

The five-by-five model supports a 'circular' planning process or, more correctly, a 'spiral' process. It is possible to start at each of the basic elements and to continue working with the other elements in repetitive circles, proposing ideas and asking questions until a coherent solution emerges.

A central feature of working with the five-by-five model in a circular process is a series of questions and answers. For example, if the planning team is addressing the project task and has decided to explore a given idea, a series of questions immediately arises, such as: Will it be possible to gather support from stakeholders for the idea, is it realistic to implement the idea, will it be possible to provide qualified persons for the project? These questions suggest moving to one or more of the five elements in the model, giving rise to further planning efforts in these elements. In turn, this may lead to questions addressing the project task element. For example, stakeholders propose to alter the direction of the project and also to lower the level of ambition, if necessary. Or an analysis of how the project might be realized suggests that the length of the project be extended, while partial results may be obtained.

The question-and-answer dialogue between the five elements may spur creative thinking and stimulate development of new ideas. Furthermore, the proposed circular planning process using the five-by-five model allows for a 'soft' planning of a project in the very early phase where playing with ideas is encouraged. Through the question-and-answer dialogue, many ideas may be tried out.

Eventually, the dialogue will converge on a holistic conception of a feasible project – a project vision. If no convergence is possible, then perhaps it is not a good idea to initiate a project. The five-by-five model may also be useful during the project as a check of feasibility of the project plan, in view of changes taken place.

FOUR PROJECT DIMENSIONS

PEOPLE

The most important component of a product and its successful implementation is human resources. In building a proper product, a well-managed team with clear-cut roles defined for each person/team will lead to the success of the product. We need to have a good team in order to save our time, cost, and effort. Some assigned roles in software project planning are **project manager, team leaders, stakeholders, analysts,** and other **IT professionals**. Managing people successfully is a tricky process which a good project manager can do.

The Players:

The software process (and every software project) is populated by players who can be categorized into one of five constituencies:

1. Senior managers who define the business issues that often have significant influence on the project.

2. Project (technical) managers who must plan, motivate, organize, and control the practitioners who dosoftware work.

3. Practitioners who deliver the technical skills that are necessary to engineer a product or application.

4. Customers who specify the requirements for the software to be engineered and other stakeholders whohave a peripheral interest in the outcome.

5. End-users who interact with the software once it is released for production use.

Importance Of People Factor:

There is a saying, ***"It's always a people problem."***

Usually the things that make or break a project are process and people issues. The way that you work on a day-to-day basis. Who your architects are, who your managers are, and who you are working with on the programming team. How you communicate, and most importantly how you solve processes and people problems when they come up.

For improving the readiness of a software organization to undertake big and complex applications, it is important to attract, grow, motivate, deploy, and retain the talent of the people working for the organization.

Some key practice areas for software people:

Recruiting, selection, performance management, training, compensation, career development, organization and work design, and team/culture development. Organizations that achieve high levels of maturity in the people management areas have a higher likelihood of implementing effective software engineering practices.

Improvement:

People management can be improved by:

- **Team Selection:** Team should be selected while keeping the past experiences and the nature of the project in mind whether it requires people from the same or different departments etc.

- **Team Organization:** Once the team is selected, it should be organized in a way that everyone gets the tasks according to their expertise, no one gets overburdened, and everyone knows exactly what to do.

- **Motivation:** The ability to encourage technical people to produce to their best ability and become more creative.

Planning The Structure Of A Software Team:

- The difficulty of the problem to be solved.
- The size of the resultant program(s) in lines of code or function points
- The time that the team will stay together (team lifetime).
- The degree to which the problem can be modularized.
- The required quality and reliability of the system to be built.
- The rigidity of the delivery date.
- The degree of sociability (communication) required for the project.

Success Factors

For managing people in organizations or projects, below mentioned factors should be considered,

- **Matching People to tasks:** Assign tasks according to the expertise or according to the person's potential of learning (new things).

- **Career development:** Step by step promotion (Increase in designation and salary). Growth in terms of salary and designation and Technical growth (learning) both are important.

- **Balance:** individual & Team: Balance between individual role & team of people with whom a person is working, no dominance by a particular head ---increases productivity

- **Clear communication:** Use task management tools by generating alerts or emails, not verbal orders to employees as it can cause miscommunication, forgetfulness, excuses of not doing work etc.

PROCESS

A series of actions or steps taken in order to achieve a particular end is called a process. A project in execution is a process.

TYPES OF PROCESSES

Technical Aspects:

- Following points to be considered.
- How work is done.
- How document is made.
- Performance of algo, how much effective it is, selection of algo. They should be defined before in documentation.
- If documentation is good, it acts like a checklist.
- Be ready to adapt-When necessary, adapt your approach to constraints imposed by the problem, the people, and the project itself.
- Focus on quality at every step.
- Be agile-keep your technical approach as simple as possible, keep the work products you produce as concise as possible, and make decisions locally whenever possible
- Create work products that provide value for others- Be sure that the work product imparts the necessary information without ambiguity or omission.

Management aspects:

- Concerned with whether work is done or not. (document completed or not).
- Establish mechanisms for communication and coordination.
- Projects fail because important information falls into the cracks and/or stakeholders fail to coordinate their efforts to create a successful end product. These are management issues and they must be addressed.
- Assess risk. Lots of things can go wrong as software is being developed. It’s essential that you establish contingency plans.
- Build an effective team. Software engineering process and practice are important, but the bottom line is people. Build a self-organizing team that has mutual trust and respect.

DEVELOPMENT FUNDAMENTALS

Quality assurance

- Focus on quality at every step. The exit condition for every process activity, action, and task should focus on the quality of the work product that has been produced.
- Assess risk. Lots of things can go wrong as software is being developed. It’s essential that you establish contingency plans.

Risk Management:

Tasks required to assess both technical and management risks.

Lifecycle Planning

- Plan who will do what. (like Who will be in design team, programming team)

Avoid abuse by neglect.

Customer Orientation

- A customer is the person or group who originally requested the software to be built, defines overall business objectives for the software, provides basic product requirements,

- Staff is told what to do. We should involve customers too to verify the work done. Be it in a meeting or there can be a person from company who interacts with customer.

- Involve stakeholders in the planning activity. Stakeholders define priorities and establish project constraints. To accommodate these realities, software engineers must often negotiate order of delivery, time lines, and other project-related issues.

Process maturity improvement

- Process maturity is an indication of how close a developing process is to being complete and capable of continual improvement through qualitative measures and feedback.

- Example-In software testing->How much testing I am doing. Whether I am checking for all possible values (positive, negative, zero, infinity).

- The more the maturity the more the quality is achieved.

- As a project manager I will just give direction for quality and will not do itself.

Rework Avoidance

- Most companies keep less people and more work is taken from them. Due to which he will not get enough time to make product.

- Keep trainee.

- Use tools.

- It is caused by problems in your quality management process.

PRODUCT

The “tangible” dimension

Product size management

- If you can reduce a product’s feature set, you can reduce the product’s schedule.
- If the feature set is flexible, you might be able to use the 80/20 rule and develop the 80% of the product that takes 20% of the time.
- You can develop the other 20% later.

Product characteristics and requirements

Efficiency:

The software should not make wasteful use of system resources such as memory and processor cycles.

Maintainability:
It should be possible to evolve the software to meet the changing requirements of customers.

Dependability:
It is the flexibility of the software that ought to not cause any physical or economic injury within the event of system failure. It includes a range of characteristics such as reliability, security, and safety.

In time:
Software should be developed well in time.

Within Budget:
The software development costs should not overrun and it should be within the budgetary limit.

Functionality:
The software system should exhibit the proper functionality, i.e. it should perform all the functions it is supposed to perform.

Adaptability:
The software system should have the ability to get adapted to a reasonable extent with the changing requirements.
A product with ambitious goals will take longer to than a product without any goals for these characteristics.

Feature Creep Management
Customer gets educated once sees the product and then demands more/enhanced features

TECHNOLOGY

Often the least important dimension

Language and tool selection
Changing from less effective tools to more effective tools can also be a fast way to improve your development speed.

Value and Cost of Reuse
The current move toward component ware (VBXs and OCXs) might eventually produce dramatic results.

TRADEOFF / PROJECT MANAGEMENT / TRIPLE CONSTRAINT TRIANGLE:

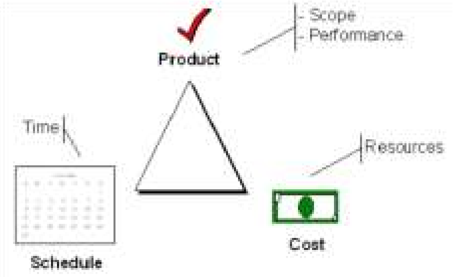
Scope
Schedule
Resources

The “project management triangle” or “tradeoff triangle” visualizes the problem of “triple constraints”—the need to balance scope, cost, and time in order to maintain a high-quality final product.

The project management triangle is made up of three variables that determine the quality of the project:

- Scope/Product
- Cost/Resources
- Time/Schedule

The triangle demonstrates how these three variables are linked—if one of the variables is changed, the other two must be adjusted in order to keep the triangle connected. If the triangle breaks—that is, if one point is moved without adjusting one or both of the other points along with it—the quality of the project will suffer.



No single variable of the project triangle can be changed without making tradeoffs with the other two points of the triangle. It’s the project manager’s job to balance all three elements in order to keep their project within budget and on deadline while still fulfilling the specifications of the project’s scope.

PRODUCT (SCOPE): Scope is the defined features and functions of a product, or the tasks needed to finish a project. Scope involves getting information required to start a project, including the features the product needs to meet its stakeholders' requirements.

COST (RESOURCES): The financial resources of a project and the resources required to complete the project like team members, equipment etc.

SCHEDULE (TIME): Project Schedule includes start date and finish date for each activity/phase, total time allocated for each activity/phase.

When you change or want to change any one of the sides, either or both of the other two will get impacted.

How tradeoff or project management triangle manages constraints?

The success of the project is impacted by the scope, cost and time of the project. But in most cases, it is difficult to build a project with minimum costs, complete scope and a limited schedule, while managing a project some of the variables can change and some remain fixed. In this scenario this tradeoff triangle is used to decide which of the constraints can be adjusted or we can tradeoff between these constraints.

There might be a case,

- Where we have to deliver a project in a limited time, so we can reduce the scope, exclude some features and allocate more resources and budget to complete the project earlier.

- Where we want to increase the scope, so it will require more resources and time to complete the project.

- Where we have limited resources and budget, so in this case we'll reduce the scope and exclude some features, and time can also exceed as we don't have the budget and resources resulting in an understaffed team.

Time, Cost, and Scope can all become high priorities. The conventional wisdom is 'pick two'. Let the third go. E.g. If you want a product on time with a low cost, the team will likely have to cut functions, open tolerances, or sacrifice quality.

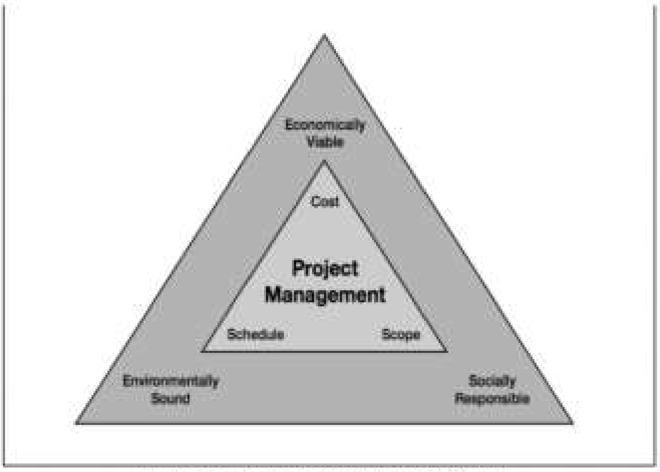
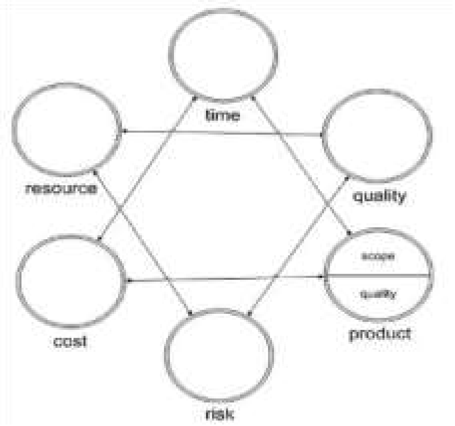


Figure 1.2 Project Management by Jennifer Russell

PROJECT MANAGEMENT STAR MODEL



Traditionally the Project Constraint Model recognised three key constraints; "Cost", "Time" and "Scope", but the project management star model evolved based on the previous triple constraint with 6 factors to be monitored and managed. This is illustrated as a 6 pointed Star that is the combination of two overlaid triangles.

It represents the separation and relationship between project inputs/outputs factors on one triangle and the project processes factors on the other. The star variables are:

1. Input-Output Triangle
    - Product (Scope & Quality)
    - Cost
    - Time

2. Process Triangle
    - Risk
    - Quality
    - Resources

We can infer that, in terms of project's output, "scope & quality" both can be adjusted to make up the product, while quality of process can also be monitored separately.

The 6 Project Constraints:

1. **Product (Scope & Quality):**
   What will be the outcome of the project in terms of scope and quality.

2. **Cost:**
   The budget available to complete the project.

3. **Time:**
   What is the schedule of the project, when will the project be completed?

4. **Risk:**
   What can go wrong and what can be done about it.

5. **Quality:**
   Is the expected outcome and actual outcome of the process same?

6. **Resources:**
   Who and what is required to do the work (team members, equipment etc.)

Description Of Star Model:
From the above diagram, it is clear that for cost, product (quality and scope) and schedule are dependent on each other, change in one will have an impact on others, either one or both. There might be a case,

- Where we have to deliver a project in a limited time, so we can reduce the scope, exclude some features and increase the budget to complete the project earlier.

- Where we want to increase the scope, so it will require more cost and time to complete the project.

- Where we have limited budget, so in this case we'll reduce the scope and exclude some features, and time can also exceed as we dont have the budget to acquire more staff and allocate them salaries.

We can also see that processes, resources, quality and risks are dependent on each other. There might be a case,

- Where we may have any risk like inaccurate time or resource estimation for a process, which could affect the quality, so to avoid the risk from turning to failure we'll have to adjust the allocation of resources.

- Where resources are not available, this might result in the risk of delay or exceeding the budget, or if we don't allocate more resources, then the quality would be affected.

- Where improving the quality of a process may result in using more resources which in return can again increase the risk of costs exceeding the limits or else if quality is compromised then it can result in the risk that stakeholders might not be satisfied by the progress of the project.

PROJECT PLANNING

Software project begins with a set of activities that are collectively called project planning.

Determine requirements

In planning, first requirements are gathered and analyzed and a detailed document is prepared in which all product specifications are mentioned.

Determine resources

Which resources are required? Is the product feasible to make? Cost and schedule is determined.

Select lifecycle model

Which lifecycle model will best suit the project requirements? Same lifecycle model is used for projects of same domain.

Determine product features strategy

A product strategy is the foundation of a product life cycle and the execution plan for further development.

PROJECT TRACKING

Cost, efforts and schedule

Once the tasks dictated by the software process model is refined based on the functionality of the system , effort and duration are allocated for each task and an activity network is created that allows the project to meets its deadlines. The work product of this activity is the project schedule and in order that it is accurate, it is required to check all tasks are covered in the activity network, effort and timing are appropriately allocated, interdependencies are correctly indicated, resources are allocated tasks in a right manner and closely spaced milestones are defined to track the project easily.

Planned vs actual

If in the end of project, deadline cannot be met, or the quality is not achieved, then the problems are determined. QA determines what problems occurred in the project lifecycle and which process was not conducted correctly. Project manager prepares a report regarding the issues in the process. Software tools must be used for tracking since it is an efficient way to track a process.

Tracking the project schedule can be done by conducting periodic project status meeting, evaluating result of reviews conducted at all stages of development life cycle, determining the completion of defined project milestones, comparison of actual and planned dates, using earned value analysis technique for performing quantitative analysis of program.

How to handle when things go off plan?

Tracking is easy in agile development as faults can be tracked in the initial development but if the defects are identified after 50% of the project is done then delay and increase in cost is inevitable.

Following measures should be taken:

- Identify the key problem areas
- Re-evaluate the core objectives
- Examine the communication channels between the tram members
- Address stakeholders’ concerns

SCOPE CREEP MANAGEMENT

Scope Creep is not only Inevitable – It's Natural
Every IT project is executed with a set of deliverables, and has an expected closure time. Prior to this closure period, there are a predetermined set of tasks and activities to complete the project successfully. These tasks constitute the **scope** of a project. Since a project schedule is closely tied to the delivery timeline and the scope, a little variation in the scope can affect delivery and in turn affect the success of the project.

This inching forward of scope to introduce more requirements that are not included in the initial planning of the project whilst maintaining the same time frame for project delivery, is called **Scope Creep.** Scope Creep is the pejorative name given to the natural process by which clients discover what they really want.

The scope creep can be classified based on the users who creates these changes:

- Business Scope Creep
- Technology Scope Creep

BUSINESS SCOPE CREEP
Systems are designed to solve the business needs for a company. Due to continual changes in market trends, the requirements that are defined before, might change. The common reasons for these changes are:

1. Insufficient Requirements Analysis Definition resulting in business requirements that are not well defined.

2. Underestimating the complexity of the problem in an unknown industry.

3. Management failure in managing user expectations.

4. Involving the users only in later stages of project life cycle such as programming and testing.

In an IT project, regardless of whether it is outsourced or built in-house, the project team works with the client to gather the requirements. This requirement definition analysis phase involves meetings, interviews, and questionnaires with the client about the current system and their future needs. In most cases, clients are unable to specify exactly what they want in the beginning until they see the product. It is also often difficult for business users to visualize how the new system will be until they see it.

When the users do see the new system for the first time, changes may be needed because any new applications will be initially unfamiliar to users. Most of the time, the user perspective is to always look for things that won't work, rather than the things that do work in the system. The approach the business users have in mind is that,*"We're spending so much time and money anyway, so let's add this during the testing phase".* This expands the scope way beyond what you can accomplish or really need.

Solution to Business Scope Creep

1. Define the business requirements as **“must-haves”** and **“nice to haves”** and prioritize them. Identify the risks for each “must-have” requirement and get the stakeholders approval. Plan these prioritized requirements in the form of phased deliverables during the project life cycle.

2. Set project expectations with the customer stakeholders and get the buy-in from the customer to make them agree on the finalized requirements.

3. Decide and document the agreed project deliverables in the **Statement Of Work** (SOW) document and requirement areas that are NOT included.

4. Document requirements and review with the customers before any sign off.

5. Decide and document how the users will use the system in the form of test cases during the requirement analysis phase.

6. Make a flexible project plan allowing users to participate at the design phase and incorporate their suggestions. In case scope creep cannot be avoided, participate in rescoping.

7. Introduce a formal change management process that would allow the users to define the requests as **“Your Enhancement Submission”** (YES) form. It is surprising how effectively this cuts out low priority demands, when users have to initiate a change requisition. Follow the six steps for any changes or deviations from the initial set of requirements

   - Record
   - Assess
   - Plan
   - Build
   - Implement
   - Close.

8. Do an impact analysis and attach a cost and time for the new requirements. This is effective in getting the sponsor to revalidate the new requirements.

TECHNOLOGY SCOPE CREEP

The scope creep created by the technologists can be broadly classified into two categories –

Customer Pleasing

The project team or an individual who wants to please the customer and is reluctant to say “no” to a change in the requirements.

Solution

1. The project team’s responsibility is to let the business know that the requested change is considerably different from the requirements approved during the project scoping process. The team should provide the business sponsors with the options and explain to

them how these changes could impact the budget, timelines and resources. The options are

- Integrating the new set of requirements in a different phase
- Stop the project so that new additional requirements can be properly scoped and integrated rather than tacked on.
- Continue the project without rescoping.

2. Since the user can visualize the system, perform a visual walkthrough session during the requirements phase to define what the client wants before the system is built. This iterative **Prototyping Approach** or **Joint Application Development** (JAD) session with the client can help the team to identify the features and can deliver a final product close to the client's needs and will result in project success.

Technical Gold-Plating

The programmers/developers who add features and functionality that have not been specified in the approved requirements definition. The reason for these changes could be the business requirements are lacking the details or the programmer is a perfectionist.

Solution:

1. Specify the "must-have" requirements in the form of a checklist and track them through the development process. This process would help to check the deliverables from the developer.

2. Because they play a crucial role in the process, involve the developers during the Requirements Management stage to prevent the team from starting with incomplete or ambiguous requirements. To decrease the risk of destabilizing the project, the developer needs to incorporate the changes that have been approved by the team so that he knows that he is working on agreed upon functionality.

3. Introduce a **Change Control Board** (CCB) team that would evaluate the risk of implementing the changes done by the developers. The team would categorize the risk as high, medium, low and define a process that could capture these kinds of requirements in the early stages of the project. The team should suggest a reward mechanism for the developer if the feature introduced by the developer gets implemented.

Conclusion

The conclusion from this article is that "scope creep" is always a change or growth of project scope. Instead of preventing the changes completely, we should work as a team to effectively manage the changes by not affecting the project timelines and budget.

CLASSIC MISTAKES ENUMERATED

Some ineffective development practices have been chosen so often, by so many people, with such predictable, bad results that they deserve to be called “classic mistakes”. Developers, managers, and customers usually have good reasons for making the decisions they do, and the seductive appeal of the classic mistakes is part of the reason these mistakes have been made so often. But because they have been made so many times, their consequences have become easy to predict, and they rarely produce the results that people hope for.

PEOPLE

Here are some of the people-related classic mistakes.

1. **Undermined motivation**
   Motivation has a larger effect on productivity and quality than any other factor. Management undermined morale throughout the project by giving a pep talk at the beginning and then requiring overtime in the middle of the project and going on a long vacation while the team worked through the holidays.

2. **Weak personnel**
   After motivation, either the individual capabilities of the team members or their relationship as a team probably has the greatest influence on productivity. Hiring from the bottom of the barrel will threaten a rapid development effort. Personnel selections made with an eye toward who could be hired fastest instead of who would get the most work done over the life of the project gets the project off to a quick start but doesn’t set up for rapid completion.

3. **Uncontrolled problem employees**
   Failure to deal with the problem of personnel also threatens development speed. Failure to take action to deal with a problem employee is the most common complaint that team members have about their leaders.

4. **Heroics**
   Some software developers place a high emphasis on project heroics, thinking that certain kinds of heroics can be beneficial . But emphasizing heroics in any form usually does more harm than good. An emphasis on heroics encourages extreme risk taking and discourages cooperation among the many stakeholders in the software-development process.

5. **Adding people to a late project**
   This is perhaps the most classic of the classic mistakes. When a project is behind, adding people can take more productivity away from existing team members than it adds through new ones.

6. **Noisy, crowded offices**
   Most developers rate their working conditions as unsatisfactory. About 60 percent report that they are neither sufficiently quiet nor sufficiently private . Workers who occupy quiet, private offices tend to perform significantly better than workers who occupy noisy, crowded work bays or cubicles.

7. **Friction between developers and customers**

   Friction between developers and customers can arise in several ways. Customers may feel that developers are not cooperative when they fail to deliver on their promises. Developers may feel that customers unreasonably insist on unrealistic schedules or requirements changes after requirements have been baselined.

   The primary effect of this friction is poor communication, and the secondary effects of poor communication include poorly understood requirements, poor user-interface design, and, in the worst case, customers refusing to accept the completed product. On average, friction between customers and software developers is so severe that both parties consider canceling the project. Such friction is time-consuming to overcome, and it distracts both customers and developers from the real work of the project.

8. **Unrealistic expectations**

   One of the most common causes of friction between developers and their customers or managers is unrealistic expectations. Project managers or developers ask for trouble by getting funding based on overly optimistic schedule estimates.

9. **Lack of effective project sponsorship**

   High-level project sponsorship is necessary to support many aspects of rapid development including realistic planning, change control, and the introduction of new development practices. Without an effective project sponsor, other high-level personnel in your organization can force you to accept unrealistic deadlines or make changes that undermine your project.

10. **Lack of stakeholder buy-in**

    The close cooperation that occurs only when you have complete buy-in from all stakeholders, that includes the executive sponsor, team leader, team members, marketing, end-users, customers, allows for precise coordination of a rapid development effort that is impossible to attain without good buy-in.

11. **Lack of user input**

    User input is essential for a successful project development..

12. **Politics placed over substance**

    Larry Constantine reported on four teams that had four different kinds of political orientations. Politicians, specialized in managing up,concentrating on relationships with their managers. Researchers concentrated on scouting out and gathering information.;Isolationists kept to themselves, creating project boundaries that they kept closed to non-team members.Generalists did a little bit of everything: they tended their relationships with their managers, performed research and scouting activities, and coordinated with other teams through the course of their normal workflow. Constantine reported that initially the political and generalist teams were both well regarded by top management. But after a year and a half, the political team was ranked dead last. Putting politics over results is fatal to speed-oriented development.

13. **Wishful thinking**
    Wishful thinking isn’t just optimism. It’s closing your eyes and hoping something works when you have no reasonable basis for thinking it will. Wishful thinking at the beginning of a project leads to big blowups at the end of a project. It undermines meaningful planning and may be at the root of more software problems than all other causes combined.

PROCESS

Here are some of the worst process-related mistakes.

4. **Overly optimistic schedules**
   The challenges faced by someone building a three-month application are quite different from the challenges faced by someone building a one-year application. Setting an overly optimistic schedule sets a project up for failure by under-scoping the project, undermining effective planning, and abbreviating critical upstream development activities such as requirements analysis and design. It also puts excessive pressure on developers, which hurts developer morale and productivity.

5. **Insufficient risk management**
   Failure to manage risks is one of the most common classic mistakes. If you don’t actively manage risks, only one thing has to go wrong to change your project from a rapid-development project to a slow-development one.

6. **Contractor failure**
   Companies sometimes contract out pieces of a project when they are too rushed to do the work in-house. But contractors frequently deliver work that’s late, that’s of unacceptably low quality, or that fails to meet specifications risks such as unstable requirements. If the contractor relationship isn’t managed carefully, the use of contractors can slow a project down rather than speed it up.

7. **Insufficient planning**
   If you don’t plan to achieve rapid development, you can’;t expect to achieve it.

8. **Abandonment of planning under pressure**
   Projects make plans and then routinely abandon them when they run into schedule trouble. The problem isn’t so much in abandoning the plan as in failing to create a substitute and then falling into code-and-fix mode instead.

9. **Wasted time during the fuzzy front end**
   The ”fuzzy front end” is the time before the project starts, the time normally spent in the approval and budgeting process. It’s not uncommon for a project to spend months or years in the fuzzy front end and then to come out of the gates with an aggressive schedule. It’s much easier and cheaper and less risky to save a few weeks or months in the fuzzy front end than it is to compress a development schedule by the same amount.

10. **Shortchanged upstream activities**
    Projects that are in a hurry try to cut out non-essential activities, and since requirements analysis, architecture, and design don't directly produce code, they are easy targets. Projects that skimp on upstream activities typically have to do the same work downstream at anywhere from 10 to 100 times the cost of doing it properly in the first place.

11. **Inadequate design**
    A special case of shortchanging upstream activities is inadequate design. Rush projects undermine design by not allocating enough time for. The design emphasis is on expediency rather than quality, so you tend to need several ultimately time-consuming design cycles before you finally complete the system.

12. **Shortchanged quality assurance**
    Projects that are in a hurry often cut corners by eliminating design and code reviews, eliminating test planning, and performing only perfunctory testing. This inefficiency undermines development speed.

13. **Insufficient management controls**
    In the case study, there were few management controls in place to provide timely warnings of impending schedule slips, and the few controls in place at the beginning were abandoned once the project ran into trouble. Before you can keep a project on track, you have to be able to tell whether it's on track.

14. **Premature or too frequent convergence**
    Shortly before a product is scheduled to be released there is a push to prepare the product for release--improve the product's performance, print final documentation, and so on. On rush projects, there is a tendency to force convergence early.The extra convergence attempts don't benefit the product. They just waste time and prolong the schedule.

15. **Omitting necessary tasks from estimates**
    If people don't keep careful records of previous projects, they forget about the less visible tasks, but those tasks add up. Omitted effort often adds about 20 to 30 percent to a development schedule.

16. **Planning to catch up later**
    Many projects simply plan to catch up later, but they never do. You learn more about the product as you build it, including more about what it will take to build it. That learning needs to be reflected in the schedule.Another kind of reestimation mistake arises from product changes. If the product you're building changes, the amount of time you need to build it changes too.

14. **Code-like-hell programming**
Some organizations think that fast, loose, all-as-you-go coding is a route to rapid development. If the developers are sufficiently motivated they can overcome any obstacles. This is far from the truth. The entrepreneurial model is often a cover for the old code-and-fix paradigm combined with an ambitious schedule, and that combination almost never works. It's an example of two wrongs not making a right.