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 do software work.

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

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

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

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.

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:

- 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.
- 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.
- 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.

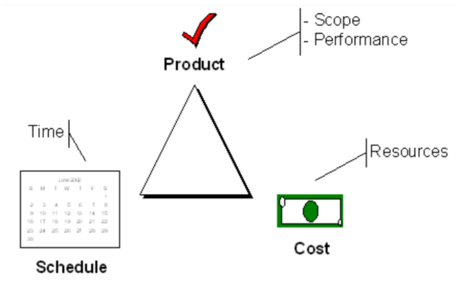
TRADEOFF / PROJECT MANAGEMENT / TRIPLE CONSTRAINT TRIANGLE:

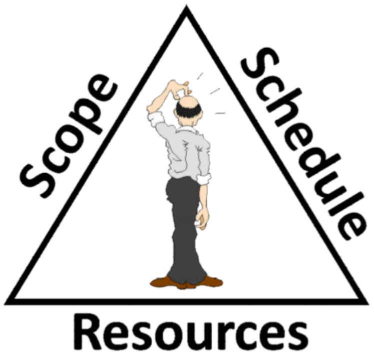
The “project management triangle” or “tradeoff triangle” visualizes the problem of “triple constraints”—the need to balance scope, cost, and time in order to maintain a high-quality final product.

The project management triangle is made up of three variables that determine the quality of the project: scope, cost, and time.

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 dont 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.