

Software Project Management (SPM) – Software Engineering

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

Need for Software Project Management

Software is a non-physical product. Software development is a new stream in business and there is very little experience in building software products. Most of the software products are made to fit clients' requirements. The most important is that basic technology changes and advances so frequently and rapidly that the experience of one product may not be applied to the other one. Such types of business and environmental constraints increase risk in software development hence it is essential to manage software projects efficiently. It is necessary for an organization to deliver quality products, keep the cost within the client's budget constraint, and deliver the project as per schedule. Hence, in order, software project management is necessary to incorporate user requirements along with budget and time constraints.

Types of Management in SPM

1. Conflict Management

Conflict management is the process to restrict the negative features of conflict while increasing the positive features of conflict. The goal of conflict management is to improve learning and group results including efficacy or performance in an organizational setting. Properly managed conflict can enhance group results.

2. Risk Management

Risk management is the analysis and identification of risks that is followed by synchronized and economical implementation of resources to minimize, operate and control the possibility or effect of unfortunate events or to maximize the realization of opportunities.

3. Requirement Management

It is the process of analyzing, prioritizing, tracking, and documenting requirements and then supervising change and communicating to pertinent stakeholders. It is a continuous process during a project.

4. Change Management

Change management is a systematic approach to dealing with the transition or transformation of an organization's goals, processes, or technologies. The purpose of change management is to execute strategies for effecting change, controlling change, and helping people to adapt to change.

5. Software Configuration Management

Software configuration management is the process of controlling and tracking changes in the software, part of the larger cross-disciplinary field of configuration management. Software configuration management includes revision control and the inauguration of baselines.

6. Release Management

Release Management is the task of planning, controlling, and scheduling the built-in deploying releases. Release management ensures that the organization delivers new and enhanced services required by the customer while protecting the integrity of existing services.

Aspects of Software Project Management

The list of focus areas it can tackle and the broad upsides of Software Project Management is:

1. Planning

The software project manager lays out the complete project's blueprint. The project plan will outline the scope, resources, timelines, techniques, strategy, communication, testing, and maintenance steps. SPM can aid greatly here.

2. Leading

A software project manager brings together and leads a team of engineers, strategists, programmers, designers, and data scientists. Leading a team necessitates exceptional communication, interpersonal, and leadership abilities. One can only hope to do this effectively if one sticks with the core SPM principles.

3. Execution

SPM comes to the rescue here also as the person in charge of software projects (if well versed with SPM/Agile methodologies) will ensure that each stage of the project is completed successfully. measuring progress, monitoring to check how teams function, and generating status reports are all part of this process.

4. Time Management

Abiding by a timeline is crucial to completing deliverables successfully. This is especially difficult when managing software projects because changes to the original project charter are unavoidable over time. To assure progress in the face of blockages or changes, software project managers ought to be specialists in managing risk and

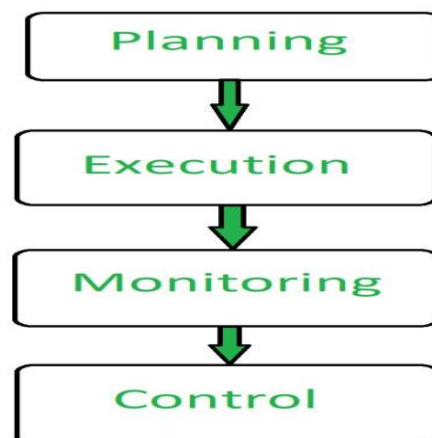
emergency preparedness. This Risk Mitigation and management is one of the core tenets of the philosophy of SPM.

5. Budget

Software project managers, like conventional project managers, are responsible for generating a project budget and adhering to it as closely as feasible, regulating spending, and reassigning funds as needed. SPM teaches us how to effectively manage the monetary aspect of projects to avoid running into a financial crunch later on in the project.

6. Maintenance

Software project management emphasizes continuous product testing to find and repair defects early, tailor the end product to the needs of the client, and keep the project on track. The software project manager makes ensuring that the product is thoroughly tested, analyzed, and adjusted as needed. Another point in favor of SPM.



Aspects of Project Management

Downsides of Software Project Management

Numerous issues can develop if a Software project manager lacks the necessary expertise or knowledge. Software Project management has several drawbacks, including resource loss, scheduling difficulty, data protection concerns, and interpersonal conflicts between Developers/Engineers/Stakeholders. Furthermore, outsourcing work or recruiting additional personnel to complete the project may result in hefty costs for one's company.

1. Costs are High

Consider spending money on various kinds of project management tools, software, & services if ones engage in Software Project Management strategies. These initiatives can be expensive and time-consuming to put in place. Because your team will be using them as well, they may require training. One may need to recruit subject-matter experts or specialists to assist with a project, depending on the circumstances.

Stakeholders will frequently press for the inclusion of features that were not originally envisioned. All of these factors can quickly drive up a project's cost.

2. Complexity will be increased

Software Project management is a multi-stage, complex process. Unfortunately, some specialists might have a propensity to overcomplicate everything, which can lead to confusion among teams and lead to delays in project completion. Their expressions are very strong and specific in their ideas, resulting in a difficult work atmosphere. Projects having a larger scope are typically more arduous to complete, especially if there isn't a dedicated team committed completely to the project. Members of cross-functional teams may lag far behind their daily tasks, adding to the overall complexity of the project being worked on.

3. Overhead in Communication

Recruits enter your organization when we hire software project management personnel. This provides a steady flow of communication that may or may not match a company's culture. As a result, it is advised that you maintain your crew as small as feasible. The communication overhead tends to skyrocket when a team becomes large enough. When a large team is needed for a project, it's critical to identify software project managers who can conduct effective communication with a variety of people.

4. Lack of Originality

Software Project managers can sometimes provide little or no space for creativity. Team leaders either place an excessive amount of emphasis on management processes or impose hard deadlines on their employees, requiring them to develop and operate code within stringent guidelines. This can stifle innovative thought and innovation that could be beneficial to the project. When it comes to Software project management, knowing when to encourage creativity and when to stick to the project plan is crucial. Without Software project management personnel, an organization can perhaps build and ship code more quickly. However, employing a trained specialist to handle these areas, on the other hand, can open up new doors and help the organization achieve its objectives more quickly and more thoroughly.

Question For Practice

1. If in a software project, the number of user input, user output, inquiries, files, and external interfaces are (15, 50, 24, 12, 8), respectively, with complexity average weighting factor. The productivity of effort = 70 percent-month is [ISRO CS 2015]

- (A) 110.54
- (B) 408.74
- (C) 304.78
- (D) 220.14

Solution: Correct Answer is (B). Show the working