Unit-5

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. This article focuses on discussing Software Project Management (SPM).

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.

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

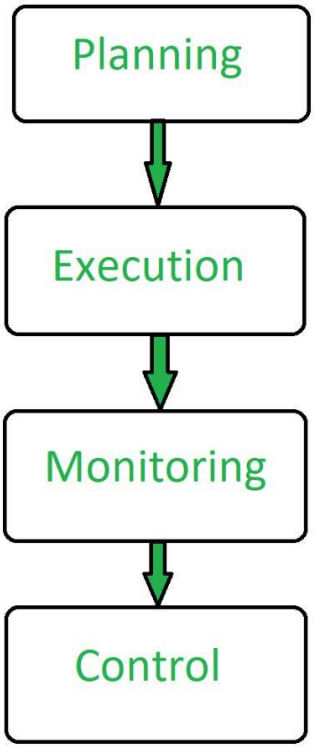
Abiding by a timeline is crucial to completing deliverables successfully. This is especially difficult when managing software projects because changes to the original project charter are unavoidable over time. To assure progress in the face of blockages or changes, software project managers ought to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

People Metrics and Process Metrics in Software Engineering

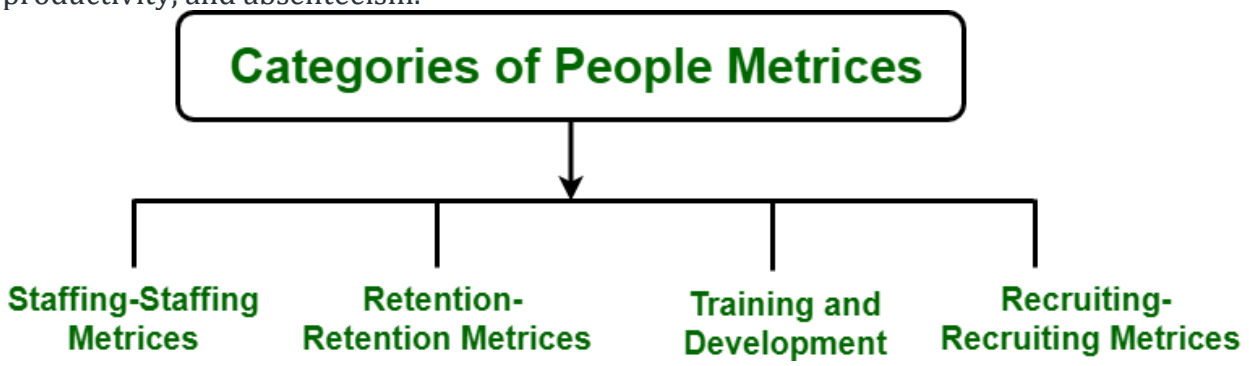
People Metrics and Process Metrics, both play important roles in software development. People Metrics helps in quantifying the useful attributes whereas Process Metrics creates the body of the software. We are going to deal with both Metrics in detail.

Table of Content

People Metrics

People metrics play an important role in **software project management**. These are also called personnel metrics. Some authors view resource metrics to include personnel metrics, software metrics, and hardware metrics but most of the authors mainly view resource

metrics to consist of personnel metrics only. In the present context, we also assume resource metrics to include mainly personnel metrics.
People metrics quantify useful attributes of those generating the products using the available processes, methods, and tools. These metrics tell you about the attributes like turnover rates, productivity, and absenteeism.



People Metrics Categories

The goal of the people metrics is to keep staff happy, motivated, and focused on the task at hand. These metrics are as:

Programming Experience Metrics

- Programming language experience
- Development methods experience
- Management experience

Communication Level Metrics

- Teamwork experience
- Communication hardware software level.
- Personal availability.

Productivity Metrics

- Size productivity
- Productivity statistics
- Quality vs. Productivity

Team Structure Metrics

- Hierarchy metrics.
- Team stability metrics

People metrics are very helpful in assisting the appropriate allocation of resources amongst various software project activities.
For more, refer to Most Important People Metrics.

Process Metrics

- Process Metrics are the measures of the development process that create a body of software. A common example of a **process metric** is the length of time that the process of software creation tasks.
- Based on the assumption that the quality of the product is a direct function of the process, process metrics can be used to estimate, monitor, and improve the reliability and quality of software. ISO- 9000 certification, or "**Quality Management Standards**",

is the generic reference for a family of standards developed by the **International Standard Organization (ISO).**

- Often, Process Metrics are tools of management in their attempt to gain insight into the creation of a product that is intangible. Since the software is abstract, there is no visible, traceable artifact from software projects. Objectively tracking progress becomes extremely difficult. Management is interested in measuring progress and productivity and being able to make predictions concerning both.
- Process metrics are often collected as part of a model of software development. Models such as Boehm's COCOMO (**Constructive Cost Model**) make cost estimations for software projects. The boat's COPMO makes predictions about the need for additional effort on large projects.
- Although valuable management tools, process metrics are not directly relevant to program understanding. They are more useful in measuring and predicting such things as resource usage and schedule.

Types of Process Metrics

- **Static Process Metrics:** Static Process Metrics are directly related to the defined process. For example, the number of types of roles, types of artifacts, etc.
- **Dynamic Process Metrics:** Dynamic Process Metrics are simply related to the properties of process performance. For example, how many activities are performed, how many artifacts are created, etc.)
- **Process Evolution Metrics:** Process Evolution Metrics are related to the process of making changes over a period of time. For example, how many iterations are there within the process)

Software Project Estimation: The First & Foremost Step To Success

JANUARY 29, 2024 BY VANCE DUONG

Software project estimation approaches assist project managers in effectively estimating critical project parameters such as cost and scope. PMs can then use these estimation strategies to give clients more accurate projections as well as budget the funds and resources they'll require for a project's success.

In this article, we'll go over which project elements should be estimated, the many available projects estimating approaches, and how to get started with estimation techniques that greatly help with software project estimation in software engineering.

1. What Is Software Project Estimation?

project estimation is a complex process that revolved around predicting the time, cost, and scope that a project requires to be deemed finished. But in terms of software development or software engineering, it also takes the experience of the **software outsourcing company**, In most cases, the whole estimation process would cost the company rather considerable cost & time at the very first stage of developing a brand new website, app, or software. However, this will act as the stepping stone to make the final result more credible, realistic, and customer-satisfying.

Whether big or small, every project is advised to employ project estimation as a crucial step to avoid unpredictable failure in the future.

Learn More On: The Essential Guide To Software Development Services

2. Which Estimations Take Place During A Project?

Ultimately, there are six critical elements of a project that benefit from the use of project estimating techniques.

2.1. Cost

In project management, cost is one of the three primary constraints. The project will fail if you do not have sufficient funds to complete it. You can help set client expectations and ensure you have enough money to complete the work if you can accurately estimate project costs early on. Estimating costs entails determining how much money you'll need and when you'll need it.

2.2. Time

Another of the project's three main constraints is the lack of time. It is critical for project planning to be able to estimate both the overall project duration and the timing of individual tasks.

You can plan for people and resources to be available when you need them if you estimate your project schedule ahead of time. It also enables you to manage client expectations for key deliverables.

2.3. Size Or Scope

The third major project constraint is scope. The project scope refers to all of the tasks that must be completed in order to complete the project or deliver a product. You can ensure that you have the right materials and expertise on the project by estimating how much work is involved and exactly what tasks must be completed.

Three sides of a triangle are often used to describe the three main constraints. This is because any changes to one constraint will inevitably have an effect on the other two. You need to know the scope and schedule to accurately estimate the budget. If one of the three estimates turns out to be higher or lower than you anticipated, the other two are likely to be off as well.

2.4. Risk

Any unforeseen event that could positively or negatively impact your project is referred to as project risk. Estimating risk entails predicting what events will occur during the project's life cycle and how serious they will be.

You can better plan for potential issues and create risk management plans if you estimate what risks could affect your project and how they will affect it.

2.5. Resources

The assets you'll need to complete the project are known as project resources. Tools, people, materials, subcontractors, software, and other resources are all examples of resources. Resource management ensures that you have all of the resources you require and make the best use of them.

It's challenging to plan how you'll manage resources without knowing what you'll need and when. This can result in people sitting around doing nothing or materials arriving weeks after you need them.

Learn More On: Top 10 Software Development Companies in Vietnam

2.6. Quality

Quality is concerned with the completion of project deliverables. Products that must adhere to stringent quality standards, such as environmental regulations, may require more money, time, and other resources than those with lower standards.

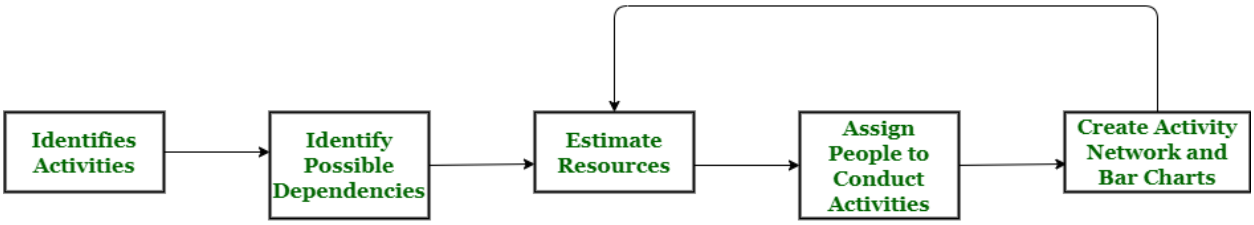
Estimating the level of quality required by the customer aids in the planning and estimating the remaining five aspects of your project. Because all six project factors are interconnected, forecasts for one can have an impact on forecasts for the other five.

As a result, applying the same software project estimation techniques to all six areas can help you improve your accuracy.

Short note on Project Scheduling

A schedule in your project's time table actually consists of sequenced activities and milestones that are needed to be delivered under a given period of time.

Project schedule simply means a mechanism that is used to communicate and know about that tasks are needed and has to be done or performed and which organizational resources will be given or allocated to these tasks and in what time duration or time frame work is needed to be performed. Effective project scheduling leads to success of project, reduced cost, and increased customer satisfaction. Scheduling in project management means to list out activities, deliverables, and milestones within a project that are delivered. It contains more notes than your average weekly planner notes. The most common and important form of project schedule is Gantt chart.



Project Scheduling Process

Process :

The manager needs to estimate time and resources of project while scheduling project. All activities in project must be arranged in a coherent sequence that means activities should be arranged in a logical and well-organized manner for easy to understand. Initial estimates of project can be made optimistically which means estimates can be made when all favorable things will happen and no threats or problems take place.

The total work is separated or divided into various small activities or tasks during project schedule. Then, Project manager will decide time required for each activity or task to get completed. Even some activities are conducted and performed in parallel for efficient performance. The project manager should be aware of fact that each stage of project is not problem-free.

Problems arise during Project Development Stage :

- People may leave or remain absent during particular stage of development.
- Hardware may get failed while performing.
- Software resource that is required may not be available at present, etc.

The project schedule is represented as set of chart in which work-breakdown structure and dependencies within various activities are represented. To accomplish and complete project within a given schedule, required resources must be available when they are needed. Therefore, resource estimation should be done before starting development.

Resources required for Development of Project :

- Human effort
- Sufficient disk space on server
- Specialized hardware
- Software technology
- Travel allowance required by project staff, etc.

Advantages of Project Scheduling :

There are several advantages provided by project schedule in our project management:

- It simply ensures that everyone remains on same page as far as tasks get completed, dependencies, and deadlines.
- It helps in identifying issues early and concerns such as lack or unavailability of resources.
- It also helps to identify relationships and to monitor process.
- It provides effective budget management and risk mitigation.

What is Risk Management?

A risk is a probable problem; it might happen, or it might not. There are main two characteristics of risk.

- **Uncertainty:** the risk may or may not happen which means there are no 100% risks.
- **Loss:** If the risk occurs in reality, undesirable results or losses will occur.

In this Article we will understand Risk Management in detail.

Table of Content

What is Risk Management?

Risk Management is a systematic process of recognizing, evaluating, and handling threats or risks that have an effect on the finances, capital, and overall operations of an organization. These risks can come from different areas, such as financial instability, legal issues, errors in strategic planning, accidents, and natural disasters.

The main goal of risk management is to predict possible risks and find solutions to deal with them successfully.

Why is risk management important?

Risk management is important because it helps organizations to prepare for unexpected circumstances that can vary from small issues to major crises. By actively understanding, evaluating, and planning for potential risks, organizations can protect their financial health, continued operation, and overall survival.

Let's Understand why risk management important with an example.

Suppose In a software development project, one of the key developers unexpectedly falls ill and is unable to contribute to the product for an extended period.

One of the solution that organization may have , The team uses collaborative tools and procedures, such as shared work boards or project management software, to make sure that each member of the team is aware of all tasks and responsibilities, including those of their teammates.

An organization must focus on providing resources to minimize the negative effects of possible events and maximize positive results in order to reduce risk effectively.

Organizations can more effectively identify, assess, and mitigate major risks by implementing a consistent, systematic, and integrated approach to risk management.

The risk management process

Risk management is a sequence of steps that help a software team to understand, analyze, and manage uncertainty. Risk management process consists of

- Risks Identification.
- Risk Assessment.
- Risks Planning.
- Risk Monitoring

Risk Management Process

Risk Identification

Risk identification refers to the systematic process of recognizing and evaluating potential threats or hazards that could negatively impact an organization, its operations, or its workforce. This involves identifying various types of risks, ranging from IT security threats like viruses and phishing attacks to unforeseen events such as equipment failures and extreme weather conditions.

Risk analysis

Risk analysis is the process of evaluating and understanding the potential impact and likelihood of identified risks on an organization. It helps determine how serious a risk is and how to best manage or mitigate it. Risk Analysis involves evaluating each risk's probability and potential consequences to prioritize and manage them effectively.

Risk Planning

Risk planning involves developing strategies and actions to manage and mitigate identified risks effectively. It outlines how to respond to potential risks, including prevention, mitigation, and contingency measures, to protect the organization's objectives and assets.

Risk Monitoring

Risk monitoring involves continuously tracking and overseeing identified risks to assess their status, changes, and effectiveness of mitigation strategies. It ensures that risks are regularly reviewed and managed to maintain alignment with organizational objectives and adapt to new developments or challenges.

Understanding Risks in Software Projects

A computer code project may be laid low with an outsized sort of risk. To be ready to consistently establish the necessary risks that could affect a computer code project, it's necessary to group risks into completely different categories. The project manager will then examine the risks from every category square measure relevant to the project.
There are mainly 3 classes of risks that may affect a computer code project:

1. **Project Risks:**
   Project risks concern various sorts of monetary funds, schedules, personnel, resources, and customer-related issues. A vital project risk is schedule slippage. Since computer code is intangible, it's tough to observe and manage a computer code project. It's tough to manage one thing that can not be seen. For any producing project, like producing cars, the project manager will see the merchandise taking form.
   For example, see that the engine is fitted, at the moment the area of the door unit is fitted, the automotive is being painted, etc. so he will simply assess the progress of the work and manage it. The physical property of the merchandise being developed is a vital reason why several computer codes come to suffer from the danger of schedule slippage.
2. **Technical Risks:**
   Technical risks concern potential style, implementation, interfacing, testing, and maintenance issues. Technical risks conjointly embody ambiguous specifications, incomplete specifications, dynamic specifications, technical uncertainty, and technical degeneration. Most technical risks occur thanks to the event team's lean information concerning the project.

3. **Business Risks:**
   This type of risk embodies the risks of building a superb product that nobody needs, losing monetary funds or personal commitments, etc.

Classification of Risk in a project

Example: Let us consider a satellite-based mobile communication project. The project manager can identify many risks in this project. Let us classify them appropriately.

- What if the project cost escalates and overshoots what was estimated? – **Project Risk**
- What if the mobile phones that are developed become too bulky to conveniently carry? **Business Risk**
- What if call hand-off between satellites becomes too difficult to implement? **Technical Risk**

Risk management standards and frameworks

Risk management standards and frameworks give organizations guidelines on how to find, evaluate, and handle risks effectively. They provide a structured way to manage risks, making sure that everyone follows consistent and reliable practices. Here are some well-known risk management standards and frameworks:

1. COSO ERM Framework:

COSO ERM Framework was introduce in 2004 and updated in 2017. Its main purpose is to addresses the growing complexity of Enterprise Risk Management (ERM).

- **Key Features**:
    - 20 principles grouped into five components: Governance and culture, Strategy and objective-setting, Performance, Review and revision, Information, communication, and reporting.
    - It promote integrating risk into business strategies and operations.

2. **ISO 31000**:

ISO 31000 was introduce in 2009, revised in 2018. It provides principles and a framework for ERM.

- **Key Features**:
    - It offers guidance on applying risk management to operations.
    - It focuses on identifying, evaluating, and mitigating risks.
    - It promote senior management's role and integrating risk management across the organization.

3. **BS 31100**:

This framework is British Standard for Risk Management and latest version issued in 2001. It offers a structured approach to applying the principles outlined in ISO 31000:2018, covering tasks like identifying, evaluating, and addressing risks, followed by reporting and reviewing risk management efforts.

Benefits of risk management

Here are some benefits of risk management:

- Helps protect against potential losses.
- Improves decision-making by considering risks.
- Reduces unexpected expenses.
- Ensures adherence to laws and regulations.
- Builds resilience against unexpected challenges.
- Safeguards company reputation.

Limitation of Risk Management

Here are Some Limitation of Risk Management

- Too much focus on risk can lead to missed opportunities.
- Implementing risk management can be expensive.

- Risk models can be overly complex and hard to understand.
- Having risk controls might make people feel too safe.
- Relies on accurate human judgment and can be prone to mistakes.
- Some risks are hard to predict or quantify.
- Managing risks can take a lot of time and resources.

System configuration management – Software Engineering

Whenever software is built, there is always scope for improvement and those improvements bring picture changes. Changes may be required to modify or update any existing solution or to create a new solution for a problem.

. **System Configuration Management (SCM)** is an arrangement of exercises that controls change by recognizing the items for change, setting up connections between those things, making/characterizing instruments for overseeing diverse variants, controlling the changes being executed in the current framework, inspecting and revealing/reporting on the changes made. It is essential to control the changes because if the changes are not checked legitimately then they may wind up undermining a well-run programming. In this way, SCM is a fundamental piece of all project management activities.
Processes involved in SCM – Configuration management provides a disciplined environment for smooth control of work products. It involves the following activities:

1. **Identification and Establishment –** Identifying the configuration items from products that compose baselines at given points in time (a baseline is a set of mutually consistent Configuration Items, which has been formally reviewed and agreed upon, and serves as the basis of further development). Establishing relationships among items, creating a mechanism to manage multiple levels of control and procedure for the change management system.
2. **Version control –** Creating versions/specifications of the existing product to build new products with the help of the SCM system. A description of the version is given below:

   Suppose after some changes, the version of the configuration object changes from 1.0 to 1.1. Minor corrections and changes result in versions 1.1.1 and 1.1.2, which is followed by a major update that is object 1.2. The development of object 1.0 continues through 1.3 and 1.4, but finally, a noteworthy change to the object results in a new evolutionary path, version 2.0. Both versions are currently supported.
3. **Change control –** Controlling changes to Configuration items (CI). The change control

   process is explained in Figure below: A change request (CR) is submitted and evaluated to assess technical merit, potential side effects, the overall impact on other configuration objects and system functions, and the projected cost of the change. The

results of the evaluation are presented as a change report, which is used by a change control board (CCB) —a person or group who makes a final decision on the status and priority of the change. An engineering change Request (ECR) is generated for each approved change. Also, CCB notifies the developer in case the change is rejected with proper reason. The ECR describes the change to be made, the constraints that must be respected, and the criteria for review and audit. The object to be changed is "checked out" of the project database, the change is made, and then the object is tested again. The object is then "checked in" to the database and appropriate version control mechanisms are used to create the next version of the software.

4. **Configuration auditing –** A software configuration audit complements the formal technical review of the process and product. It focuses on the technical correctness of the configuration object that has been modified. The audit confirms the completeness, correctness, and consistency of items in the SCM system and tracks action items from the audit to closure.
5. **Reporting –** Providing accurate status and current configuration data to developers, testers, end users, customers, and stakeholders through admin guides, user guides, FAQs, Release notes, Memos, Installation Guide, Configuration guides, etc.

System Configuration Management (SCM) is a software engineering practice that focuses on managing the configuration of software systems and ensuring that software components are properly controlled, tracked, and stored. It is a critical aspect of software development, as it helps to ensure that changes made to a software system are properly coordinated and that the system is always in a known and stable state.

SCM involves a set of processes and tools that help to manage the different components of a software system, including source code, documentation, and other assets. It enables teams to track changes made to the software system, identify when and why changes were made, and manage the integration of these changes into the final product.

Importance of Software Configuration Management

1. Effective Bug Tracking: Linking code modifications to issues that have been reported, makes bug tracking more effective.
2. Continuous Deployment and Integration: SCM combines with continuous processes to automate deployment and testing, resulting in more dependable and timely software delivery.
3. Risk management: SCM lowers the chance of introducing critical flaws by assisting in the early detection and correction of problems.
4. Support for Big Projects: Source Code Control (SCM) offers an orderly method to handle code modifications for big projects, fostering a well-organized development process.
5. Reproducibility: By recording precise versions of code, libraries, and dependencies, source code versioning (SCM) makes builds repeatable.
6. Parallel Development: SCM facilitates parallel development by enabling several developers to collaborate on various branches at once.

Why need for System configuration management?

1. Replicability: Software version control (SCM) makes ensures that a software system can be replicated at any stage of its development. This is necessary for testing, debugging, and upholding consistent environments in production, testing, and development.
2. Identification of Configuration: Source code, documentation, and executable files are examples of configuration elements that SCM helps in locating and labeling. The

management of a system's constituent parts and their interactions depend on this identification.
3. Effective Process of Development: By automating monotonous processes like managing dependencies, merging changes, and resolving disputes, SCM simplifies the development process. Error risk is decreased and efficiency is increased because of this automation.

The Software Process Improvement (SPI) – Reward or Risk

Most of the Software companies large, medium, small, or startup usually face issues in their software development projects and its delivery. The issues can vary from lack of documentation, lack of following the process, lack of process governance, lack of the integration and collaboration between the teams, lack of requirements traceability, lack of technology management, …etc.

We have discussed in a previous post the trends of software projects and that large software projects on the average run 66% over budget and 33% over schedule; as many as 17% of projects go so badly that they can threaten the existence of the company. Therefore, some methods and techniques started to exist to tackle the software process issues to suggest different improvements and identify issues and inefficiencies in the process. These methods became a standard which the companies can follow to improve their software process. Moreover, each method established its ecosystem, from providing the training and certificates for the method to provide consultancy to help companies to improve based on actual practices.

In this article, It will be good to ask yourself if the software process improvement is a peril to have or a promise for a better change for the organization and to have a superior advantage in the market. For answering this, we will discuss in this article what is SPI?, what is the SPI process steps? what are the different methods?, the motivators and demotivators of SPI projects, what are the common success factors for SPI project implementation?

What is SPI?

Software Process Improvement (SPI) methodology is defined as a sequence of tasks, tools, and techniques to plan and implement improvement activities to achieve specific goals such as increasing development speed, achieving higher product quality or reducing costs.

This definition is combined from [1][2]. SPI can be considered as process re-engineering or change management project to detect the software development lifecycle inefficiencies and resolve them to have a better process. This process should be mapped and aligned with organizational goals and change drivers to have real value to the organization.

SPI mainly consists of 4 cyclic steps as shown in the figure below, while these steps can be broken down into more steps according to the method and techniques used. While in most cases the process will contain these steps.

SPI 4 cyclic steps

Current Situation Evaluation

This step is the initial phase of the process and it is mainly to assess the current situation of the software process by eliciting the requirements from the stakeholders, analyzing the current artifacts and deliverables, and identifying the inefficiencies from the software process. The elicitation can be conducted through different techniques. For example, individual interviews, group interview, use-case scenarios, and observations.

The key considerations in this step to identify organization goals and ask the solution-oriented questions. Moreover, identifying the measurement using the GQM (Goal – Question – Metric) technique that will help in measuring the current status and measuring the effectiveness of the improvement process.

Improvement Planning

After analyzing the current situation and the improvement goals, the findings should be categorized and prioritized according to which one is the most important or have the most severity. We should observe what is the new target level of improvements should look like.

Moreover, in this step, the gap between the current level and the target level should be planned in terms of a set of activities to reach that target. These activities should be prioritized with the alignment of the involved stakeholders and the organization goals, for example, if the project is using the CMMI model, the target could be reaching maturity level 4 and the company at level 3, in that case, the plan should be focused on the process areas and their activities which is related to that level of improvement with the alignment of the organization goal.

Improvement Implementation

In this step, the planned activities are executed and it puts the improvements into practice and spreads it across the organization, what can be effective at the 2nd, 3rd, and 4th step that planning and implementation could be an iterative way, for example, implementing improvement for improving requirements first, then implementing the reduction for testing process time, and so forth. This iterative way of implementation will help the organization to realize the early benefits from the SPI program early or even adopt the plan if there is no real impact measured from the improvement.

Improvement Evaluation

What is cannot be measured cannot be improved, that's why in this step, the impact measurement is applied compared with the GQM. The before improvement measures, after the improvement measures, and the target improvement measure. Measurement, in general, permits an organization to compare the rate of actual change against its planned change and allocate resources based on the gaps between actual and expected progress.

Capability Maturity Model Integration (CMMI)

Last Updated : 17 Aug, 2020

Prerequisite – Capability Maturity Model (CMM)
Capability Maturity Model Integration (CMMI) is a successor of CMM and is a more evolved model that incorporates best components of individual disciplines of CMM like Software CMM, Systems Engineering CMM, People CMM, etc. Since CMM is a reference model of matured practices in a specific discipline, so it becomes difficult to integrate these disciplines as per the requirements. This is why CMMI is used as it allows the integration of multiple disciplines as and when needed.
Objectives of CMMI :
1. Fulfilling customer needs and expectations.
2. Value creation for investors/stockholders.
3. Market growth is increased.
4. Improved quality of products and services.
5. Enhanced reputation in Industry.

CMMI Representation – Staged and Continuous :
A representation allows an organization to pursue a different set of improvement objectives. There are two representations for CMMI :

- **Staged Representation :**
    - uses a pre-defined set of process areas to define improvement path.
    - provides a sequence of improvements, where each part in the sequence serves as a foundation for the next.
    - an improved path is defined by maturity level.
    - maturity level describes the maturity of processes in organization.

    - Staged CMMI representation allows comparison between different organizations for multiple maturity levels.
- **Continuous Representation :**
    - allows selection of specific process areas.
    - uses capability levels that measures improvement of an individual process area.
    - Continuous CMMI representation allows comparison between different organizations on a process-area-by-process-area basis.
    - allows organizations to select processes which require more improvement.
    - In this representation, order of improvement of various processes can be selected which allows the organizations to meet their objectives and eliminate risks.

CMMI Model – Maturity Levels :

In CMMI with staged representation, there are five maturity levels described as follows :

1. **Maturity level 1 : Initial**
    - processes are poorly managed or controlled.
    - unpredictable outcomes of processes involved.
    - ad hoc and chaotic approach used.
    - No KPAs (Key Process Areas) defined.
    - Lowest quality and highest risk.
2. **Maturity level 2 : Managed**
    - requirements are managed.
    - processes are planned and controlled.
    - projects are managed and implemented according to their documented plans.
    - This risk involved is lower than Initial level, but still exists.
    - Quality is better than Initial level.
3. **Maturity level 3 : Defined**
    - processes are well characterized and described using standards, proper procedures, and methods, tools, etc.
    - Medium quality and medium risk involved.
    - Focus is process standardization.
4. **Maturity level 4 : Quantitatively managed**
    - quantitative objectives for process performance and quality are set.
    - quantitative objectives are based on customer requirements, organization needs, etc.
    - process performance measures are analyzed quantitatively.
    - higher quality of processes is achieved.
    - lower risk
5. **Maturity level 5 : Optimizing**
    - continuous improvement in processes and their performance.
    - improvement has to be both incremental and innovative.
    - highest quality of processes.
    - lowest risk in processes and their performance.

CMMI Model – Capability Levels

A capability level includes relevant specific and generic practices for a specific process area that can improve the organization's processes associated with that process area. For

CMMI models with continuous representation, there are six capability levels as described below :

1. **Capability level 0 : Incomplete**
   - incomplete process – partially or not performed.
   - one or more specific goals of process area are not met.
   - No generic goals are specified for this level.
   - this capability level is same as maturity level 1.
2. **Capability level 1 : Performed**
   - process performance may not be stable.
   - objectives of quality, cost and schedule may not be met.
   - a capability level 1 process is expected to perform all specific and generic practices for this level.
   - only a start-step for process improvement.
3. **Capability level 2 : Managed**
   - process is planned, monitored and controlled.
   - managing the process by ensuring that objectives are achieved.
   - objectives are both model and other including cost, quality, schedule.
   - actively managing processing with the help of metrics.
4. **Capability level 3 : Defined**
   - a defined process is managed and meets the organization's set of guidelines and standards.
   - focus is process standardization.
5. **Capability level 4 : Quantitatively Managed**
   - process is controlled using statistical and quantitative techniques.
   - process performance and quality is understood in statistical terms and metrics.
   - quantitative objectives for process quality and performance are established.
6. **Capability level 5 : Optimizing**
   - focuses on continually improving process performance.
   - performance is improved in both ways – incremental and innovation.
   - emphasizes on studying the performance results across the organization to ensure that common causes or issues are identified and fixed.