1. SOFTWARE PROJECT MANAGEMENT

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. 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 goalof 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 andtracking 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 existingservices.

SOFTWARE CONFIGURATION MANAGEMENT

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.

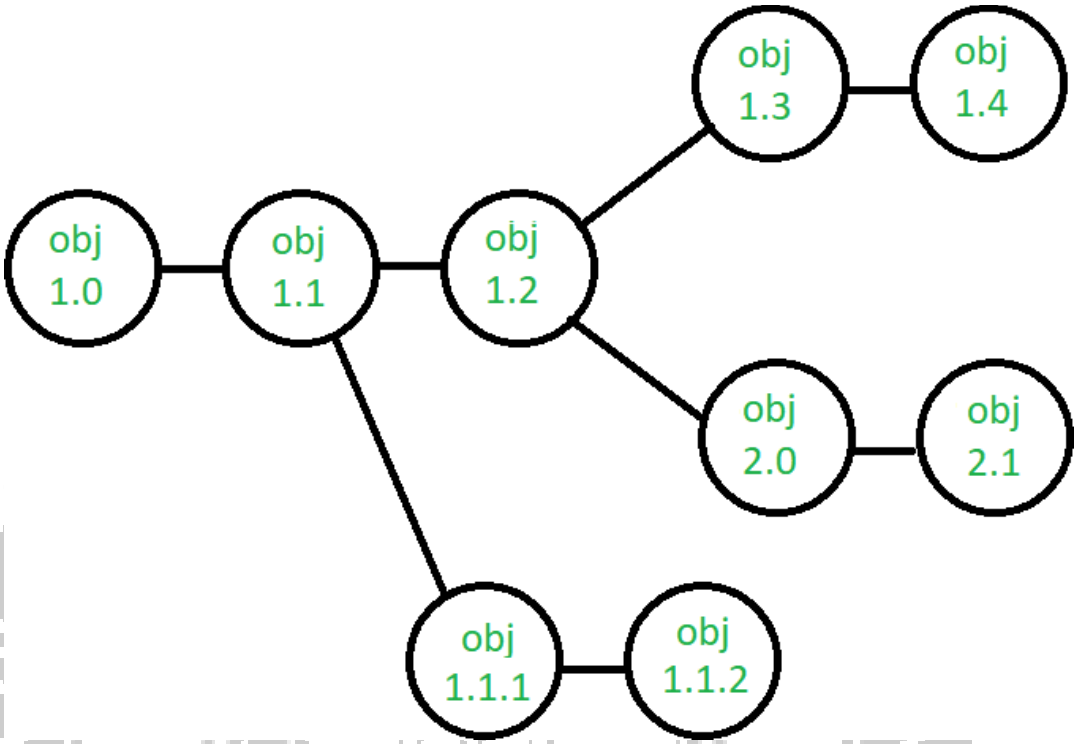
Requirements keep on changing daily so we need to keep on upgrading our systems based on the current requirements and needs to meet desired outputs. Changes should be analyzed before they are made to the existing system, recorded before they are implemented, reported to have details of before and after, and controlled in a manner that will improve quality and reduce error.

This is where the need for System Configuration Management comes. **System Configuration Management (SCM)** isan 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.

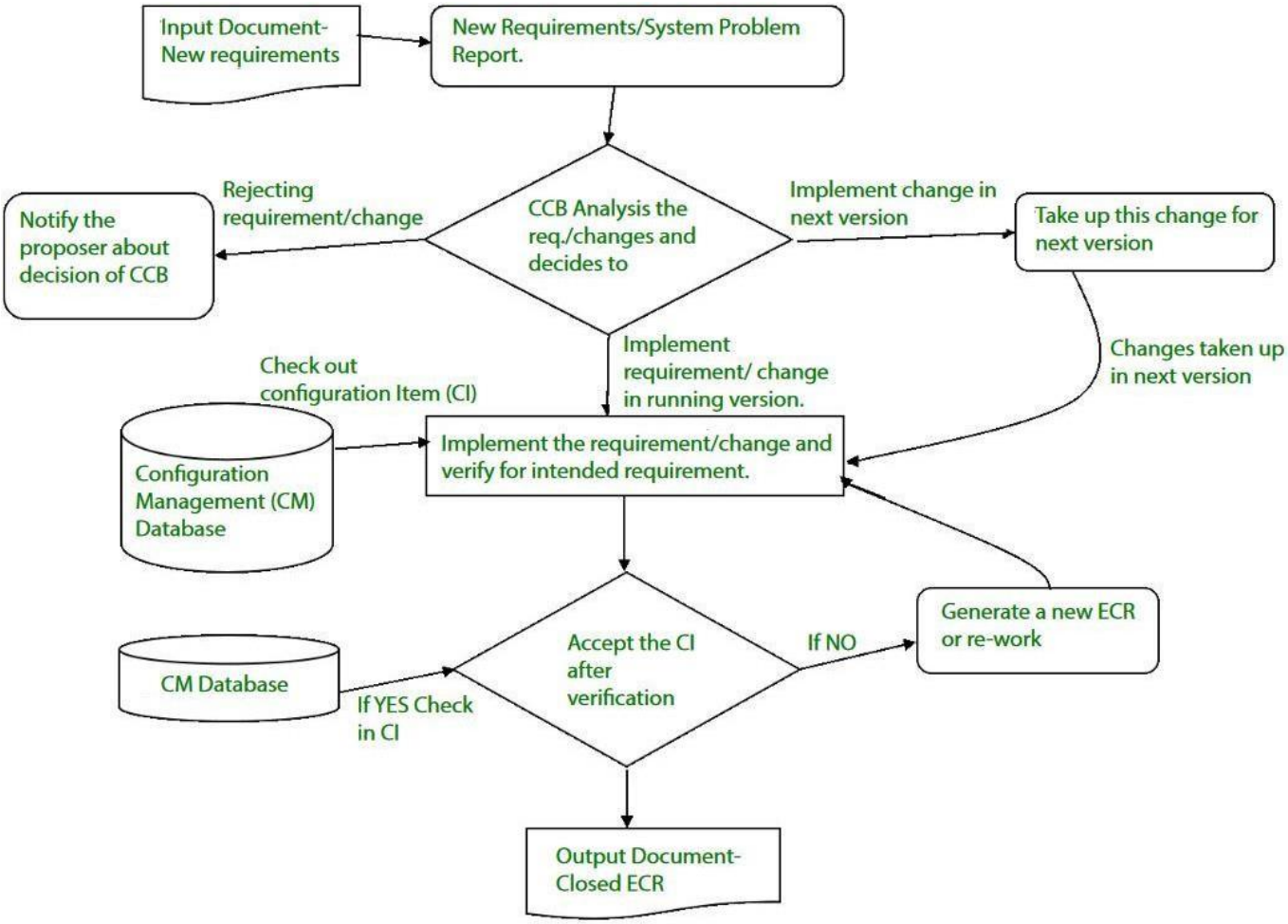
Processes involved in SCM – Configuration management provides a disciplined environment for smooth control of work products.
It involves the following activities:

1. **Identification and Establishment –** Identifying the configuration items from products that compose baselines atgiven 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 Configurationitems (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.

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 ofitems in the SCM system and tracks action items from the audit to closure.

4. **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 weremade, and manage the integration of these changes into the final product 0 seconds of 17 seconds Volume 0%

Importance of Software Configuration Management

1. Effective Bug Tracking: Linking code modifications to issues that have been reported, makes bug tracking moreeffective.
2. Continuous Deployment and Integration: SCM combines with continuous processes to automate deployment and testing, resulting in more dependable and timely softwaredelivery.
3. Risk management: SCM lowers the chance of introducingcritical flaws by assisting in the early detection and correction of problems.

4. Support for Big Projects: Source Code Control (SCM) offersan 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.

Key objectives of SCM

1. **Control the evolution of software systems:** SCM helps to ensure that changes to a software system are properly planned, tested, and integrated into the final product.
2. **Enable collaboration and coordination:** SCM helps teams to collaborate and coordinate their work, ensuring that changes are properly integrated and that everyone is working from the same version of the software system.
3. **Provide version control:** SCM provides version control for software systems, enabling teams to manage and track different

versions of the system and to revert to earlier versions if necessary.

4. **Facilitate replication and distribution:** SCM helps to ensure that software systems can be easily replicated and distributed to other environments, such as test, production,and customer sites.
5. SCM is a critical component of software development, and effective SCM practices can help to improve the quality and reliability of software systems, as well as increase efficiencyand reduce the risk of errors.

The main advantages of SCM

1. Improved productivity and efficiency by reducing the timeand effort required to manage software changes.
2. Reduced risk of errors and defects by ensuring that allchanges were properly tested and validated.
3. Increased collaboration and communication among team members by providing a central repository for software artifacts.
4. Improved quality and stability of software systems byensuring that all changes are properly controlled and managed.

The main disadvantages of SCM

1. Increased complexity and overhead, particularly in large software systems.
2. Difficulty in managing dependencies and ensuring that allchanges are properly integrated.
3. Potential for conflicts and delays, particularly in large development teams with multiple contributors.

PROJECT SCHEDULING

Project-task scheduling is a significant project planning activity. It comprises deciding which functions would be taken up when. To schedule the project plan, a software project manager wants to do the following:

1. Identify all the functions required to complete the project.
2. Break down large functions into small activities.

3. Determine the dependency among various activities.
4. Establish the most likely size for the time duration required tocomplete the activities.
5. Allocate resources to activities.
6. Plan the beginning and ending dates for different activities.
7. Determine the critical path. A critical way is the group of activities that decide the duration of the project.

The first method in scheduling a software plan involves identifying allthe functions required to complete the project. A good judgment of theintricacies of the project and the development process helps the supervisor to identify the critical role of the project effectively.

Next, the large functions are broken down into a valid set of small activities which would be assigned to various engineers. The work breakdown structure formalism supports the manager to breakdown the function systematically after the project manager has broken down the purpose and constructs the work breakdown structure; he has to find the dependency among the activities.

Dependency among the various activities determines the order in which the various events would be carried out. If an activity A necessary the results of another activity B,then activity A must be scheduled after activity B. In general, the function dependencies describe a partial ordering among functions, i.e., each service may precede a subset of other functions, but some functions might not have any precedence ordering describe between them (called concurrent function). The dependency among the activities is defined in the pattern of an activity network.

Once the activity network representation has been processed out, resources are allocated to every activity. Resource allocation is usually done using a Gantt chart. After resource allocation is completed, a PERT chart representation is developed.

The PERT chart representation is useful for program monitoring and control. For task scheduling, the project plan needs to decompose the project functions into a set of activities. The time frame when every activity is to be performed is to be determined. The end of every action is called a milestone.

The project manager tracks the function of a project by audit the timely completion of the milestones. If he examines that the

Milestones start getting delayed, and then he has to handle the activities Carefully so that the complete deadline can still be met.