**Student Name:** Sameer kamble

**Course:** Software Project Management

**Journal URL:** <https://github.com/sameer1130/SPM-journals->

**Week 4:** Feb 11

**Date: feb 11, 12**

**Key Concepts Learned:**When we refer to the configuration of a system, we mean the technical description of a system and how its component parts inter-connect.

So, what is Configuration Management?

We define Configuration Management in our free PM Glossary: ‘Be on the Inside. Decode the Jargon of Project Management’, as:

'Defining, controlling, releasing, changing, documenting, and reporting the configuration items in a system.'

We can think of that system as, for example, a technology solution we are implementing, or as the project itself. In the latter case, configuration management concerns itself with the project scope, quality management, and change control.

However, most often, we are concerned with the configuration of a solution we are building or adapting within our project. Therefore, configuration management is an important DevOps process - DevOps is a set of practices that combines software development and IT operations.

The purpose of configuration management is to ensure that we can properly track how a system is configured through its whole life, from development to retirement.

Without proper configuration management, we cannot build, test, install, accept, operate, maintain or support the system.

A big part of it is tracking changes, so we need to collect and structure a lot of data, such as:

• Date

• Version number

• Status

• Product owner

• Features

• Components

• Connections and interfaces

• Relationships with other products

It is beyond our scope to look at the methodologies and how we carry out configuration management. However, I will list the five principal components:

• Configuration Management Planning to create your CM plan document

• Configuration Identification – establishing and maintaining a baseline configuration against which changes can e tracked and recorded

• Configuration Control – like Project Change Control, this is the process of evaluating and accepting or rejecting changes to the configuration

• Configuration Status Accounting – recording and reporting on the configuration, as it changes

• Configuration Verification and Audit – independent (or quasi-independent) review of processes, artifacts, and compliance

As you might expect, there is a big choice of software tools to support configuration management

For us, as project managers, we need to ensure that our Project Plans include any configuration management plan. It will be part of your scope management process.  
  
• Software Configuration Management (SCM) is process to systematically manage, organize.

& control changes in documents, codes & other entities during the Software Development

Life Cycle.

• The primary goal is to increase productivity with minimal mistakes.

• SCM Tools: Puppet, ConfigHub, Saltstack, Ansible, Git, BitBucket, Docker & CHEF etc.  
  
1. It ensure changes to software system are properly planned, tested & integrated into the final

product.

2. Helps teams to collaborate & coordinate their work, everyone working from the same

version of the software system.

3. It manage & track different versions of the system and to revert to earlier versions if

necessary.

4. It ensure that software systems can be easily replicated & distributed to other environments

such as test, production & customer sites.

5. It improve quality & reliability of software systems, as well as increase efficiency and

reduce the risk of errors.  
  
SCM process

1. Planning & Identification:

• This method determining the scope of the software system.

• This is accomplished by having meetings and brainstorming sessions with your team.

• Identifying items like test cases, specification requirements, modules & schedule time.

• Identifying each computer software configuration items in the process.

• Group basic details of why, when and what changes will be made and who will be in charge

of making them

2. Version Control Process or Baselines:

• The aim of this step is to control the alteration and modification done to the product.

• It handle different version of configuration objects that are generated during the software

process.

• Also focuses on developing way to track the hierarchy of different versions of the software.

• Developing standardized label scheme for all products, revisions and files so that everyone is

on the same page.

1. Software Team send changes to the Software Configuration Manager

2. SCM checking examining the overall impact they will have on the pr

3. Making approved changes or explaining why change requests were d

4. If it is approved them implement all necessary changes.

5. After that Review or Reporting it.

4. Configuration Auditing Process:

• This process is used to ensure that application will develop as per the project plan and

test/verify the application as per scope.

• The audit confirms the completeness, correctness and consistency of modified items in the

SCM system and track action items from the audit to closure.

• It mentioned what is new in each version and why the changes were necessary.

• It ensures that what is built is what is delivered.

5. Review and Status Reporting Process:

• It is a technical review on the Application workflow, Process, Configuration items and

Change requests etc.

• It generate the accurate status report in every phase of SDLC process.

• Configuration Status report provide to the Developers, Testers, End users, Customers and

Stakeholders.

• It develop some application-related documents like User manual, Installation process guide,

Configuration guide, Do's and Don't Do's etc.  
  
Scm repository tools:  
1. Versioning: Save & retrieve all repository object based on there version number.

2. Dependency Tracking & Change Management: Track & manage to the changes in

relationship of all objects in repository. Ex. UML diagram.

3. Requirement Tracing: Trace design & construction of components & there deliverable

result (Forward Tracing). As per the work product, which requirement is responsible for

which feature. (Backward Tracing).

4. Configuration Management: Track a series of configuration representing specific project

milestone or production release.

5. Audit Trails: Establish information when, why & by whom changes are made in the

repository.

**Application in Real Projects:**

Given that CM is crucial in both project management and DevOps, its application requires seamless integration between project management teams and DevOps practices. This ensures that the configuration of software and systems is consistently managed from development through to operations.

Implementing version control processes allows teams to manage changes systematically. This is crucial in real-world applications where multiple versions of a product may be in development, testing, or production at any given time. Establishing baselines ensures that there is a stable version to revert to, which is essential for maintaining system integrity.

Potential challenges:  
In fast-paced development environments, managing configuration changes and maintaining documentation up-to-date can be challenging. Automating as much of the process as possible can help, but it requires investment in the right tools and training.

Benefits:  
CM facilitates better collaboration between teams by ensuring everyone is working from the same version of the software system. This can lead to improved efficiency and productivity, as well as a reduction in errors.Having a well-documented configuration and the ability to revert to previous versions makes it easier to recover from issues and adapt to changes. This flexibility is crucial in today’s dynamic project environments, where requirements and objectives can shift rapidly.

**Peer Interactions:**

As, we use Github as our main configuration management tool. We decided to start working on an project in which all the other students in other teams can also download our work and update and then we can download and work on the project that the others have updated it.

**Challenges Faced:**

Trying to lean JIRA, as I have never used it. Faced issues when I was uploading my file to it.

**Personal development activities:  
Worked on some mcq type problems**

1. What is the main purpose of configuration management in software projects?

(a) To track changes made to software requirements (Incorrect)

(b) To manage the development process itself (Incorrect)

**(c) To provide a central repository for storing and accessing software artifacts (Correct)**

**(d) To ensure that all team members are working on the same version of the software (Incorrect)**

1. What is the role of continuous integration in software development?

(a) To automate the testing of new code (Incorrect)

**(b)To integrate new code into an existing build frequently (Correct)**

(c) To track and manage software requirements (Incorrect)

(d) To deploy software to production environments (Incorrect)░

1. What is the potential risk of not managing software versions properly?

           (a) The software may not be built correctly. (Correct)

(b) Team members may work on the wrong version of a document. (Correct)

(c) The software may not be released on time. (Correct)

**(d) All of the above (Correct)**

4) What is the purpose of a configuration management system?

 (A) To manage project teams efficiently

 (B) To secure access to project information

 (C) To maintain version control of project items

**(D) All of the above**

5) How is access control managed in a configuration management system?

 (A) Each folder has a single access permission for all roles.

**(B) Roles are defined centrally with permissions for each folder.**

 (C) Access permissions are granted individually for each item.

 (D) Access control is not managed in a configuration management system.

3) Why is version control important in configuration management systems?

 (A) To ensure smooth working of teams from different locations

 (B) To prevent chaos and confusion in managing documents

 (C) To facilitate integration between components developed by different teams

**(D) All of the above**

6) What problems can arise from using decentralized configuration management systems?

 (A) Difficulty in controlling document versions

 (B) Confusion in managing work among teams

 (C) Challenges in integrating components developed by different teams

**(D) All of the above**

7) Which of the following is a recommended best practice for configuration management systems?

**(A) Continuous integration of software build with smoke test facility**

 (B) Manual synchronization of document versions across systems

 (C) Complex branching mechanism for software versioning

 (D) Lack of audit facility

8) What mechanism is described to ensure the integrity of software builds?

 (A) Continuous Integration

**(B) Automated Smoke Testing**

 (C) Version Control

 (D) Manual Code Review

9) How is the functionality of automatic smoke testing achieved in the described scenario?

 (A) By manual execution of test scripts

**(B) By using Cruise Control**

 (C) By sending emails to developers

 (D) By rewriting code multiple times

10) What happens if the smoke test fails after a new piece of code is checked into the build?

 (A) Success message is sent to the current developer

**(B) Failed message is sent to the current developer**

 (C) Code is automatically rectified

 (D) Build is marked as successful

**Goals for the Next Week:**

Learning the basics of Jira is crucial for effectively managing projects, especially in software development and IT operations where tracking issues, tasks, and workflows is essential. In Jira, everything is an issue, which could be a project task, a bug, a user story, or a support ticket.