accenture technology

Learning and Knowledge Management

Module 1: Introduction to Configuration Management

Module Objectives

At the end of this module, you will be able to:

- Describe Configuration Management (CM), its aim and practices.
- Compare the features of Centralized Server and Distributed Server models of Configuration Management.
- Define the version control terminologies used in SCM.
- Visualize the usage of Configuration Management in SDLC.

Topic List

- Configuration Management
- Centralized & Distributed Model
- Version Control Terminologies
- Configuration Management in SDLC

Topic List

Configuration Management

Centralized & Distributed Model

Version Control Terminologies

Configuration Management in SDLC

Configuration Management (1)

What is Configuration Management ?

- Configuration Management (CM) is a branch for evaluation, co-ordination, acceptance or rejection and applying of changes in artifacts which are used to construct and maintain software system.
- CM is the management of artifacts through all the stages of SDLC which includes design, build, test, baselining, release, implementation and maintenance.
- Consists of a set of policies and standards which are institutionalized and well defined.

Purpose of CM

It is used to maintain different versions of artifacts without creating a mess.

Artifact in CM

An artifact in CM can be a piece of hardware or software or documentation.

Configuration Management (2)

Key Functionalities

Configuration Management looks after the following things:

Labelling of artifacts.

Setting up the artifacts (configuration items) under a control set.

Adding and deleting an artifact in the control set.

Maintaining the changes made to the artifacts.

Using CM tools to enable configuration management.

Tracking various versions of artifacts.

Configuration Management (3)

Software Configuration Management (SCM)

Software Configuration Management (SCM) is a subset of configuration management which monitors the changes in the software

Aim of SCM

- To identify configuration items
- To perform configuration changes through Change Control Board (CCB), status accounting and auditing
- To perform Build Management, Process Management & Environment Management
- To track the defects

Uses of SCM

- Maintaining the revision control
- Establishing the baselines
- Tracking what changes are made
- Tracking who made the change

Topic List

Configuration Management

Centralized & Distributed Model

Version Control Terminologies

Configuration Management in SDLC

Centralized & Distributed Model (1)

Comparison between Centralized & Distributed Models

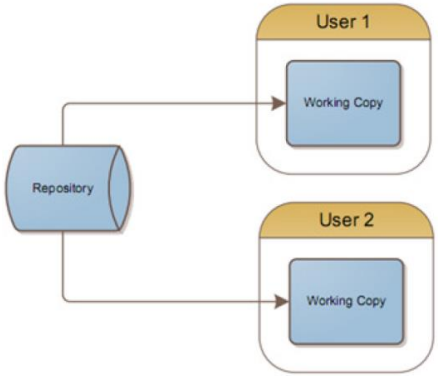
Centralized Server and Distributed Server are the types of SCM Server.

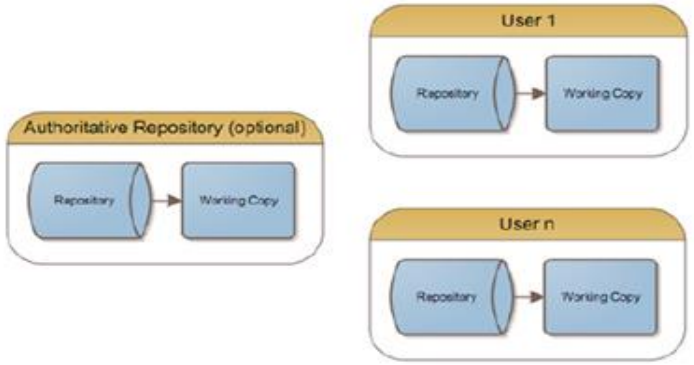
Centralized server

- In the Centralized server (client-server) model, developers use a single shared repository.
- Single server contains various versions of files.
- All the clients/users use the central place from where files are checked out. Users can do “Pessimistic” (centralised) locking during “reserved” checkouts.
- Single point of failure.

Distributed Server

- In the Distributed server model, developer have their own local repository.
- Changes are shared between repositories as a separate step/task.
- Latest snapshot of files are taken at a particular time of the repository.
- In case of failure of a server, any of the snapshots of the repositories which were checked out can be used to restore by copying back to the server. Thus each clone serves as a backup.





Centralized & Distributed Model (2)

Centralized Server Model Tools

Following are the open source Centralized Server model tools:

Centralized & Distributed Model (3)

Distributed Server Model Tools

Following are the open source Distributed Server model tools:

ArX

Topic List

Configuration Management

Centralized & Distributed Model

Version Control Terminologies

Configuration Management in SDLC

Version Control Terminologies

Key Terminologies

Following are the list of Version Control terminologies used in SCM:

Terms	Description
Repository	A central room for developer's work.
Trunk	A directory checked out for main development work on the project.
Tags	Descriptive names to specific the version (snapshot). Example: LASTSTABLECODE_ Release65 instead of the Repository UUID 7fdggf8cb-5678-40dd-a067-f216ec2e674.
Branches	Different separate spaces for developers. Example: If development process forks off into two different paths i.e when you release version 3.0, a branch is created to keep the development of 4.0 features separately from 3.0 bug-fixes.
Working copy	A snapshot of the repository. The repository is shared with all the teams, but is not modified directly. Instead, each developer checks out an individual working copy which acts as a private workplace for each developer’s work and is isolated from each others.
Commit changes	Updating changes from developer’s workplace to central server, thus the changes available to all. Other developers can pull these changes to update their working copy. At a commit, either the whole commit succeeds or is rolled back, thus is Atomic operation.

Topic List

Configuration Management

Centralized & Distributed Model

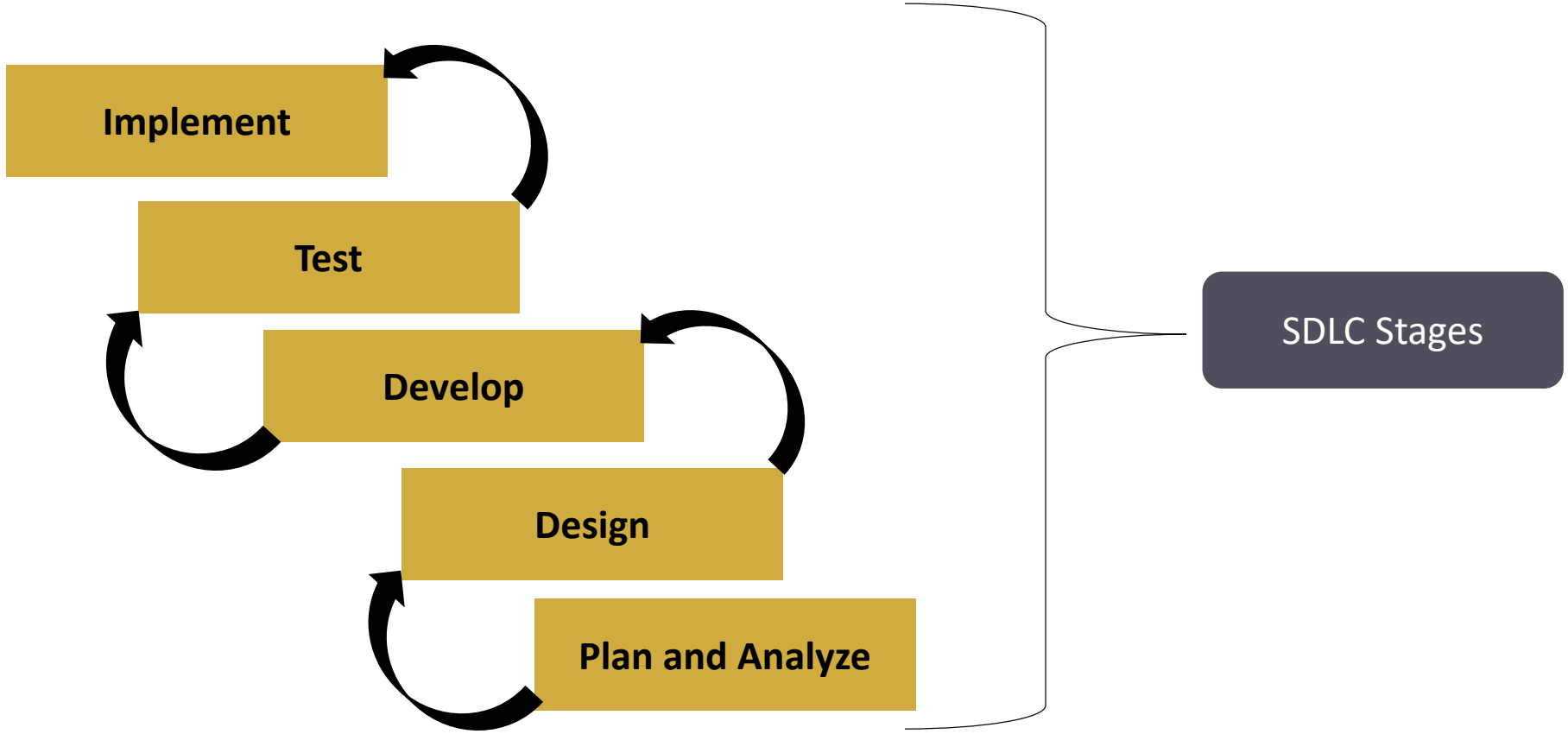
Version Control Terminologies

Configuration Management in SDLC

Configuration Management in SDLC

Overview

Configuration Management can be used in Plan, Design, Develop, Testing and Implementation stages of SDLC in order to maintain different versions of designs and code.



Knowledge Check

State True or False

The purpose of Configuration Management (CM) is to maintain different versions of artifacts without creating a mess.

- True
- False

Module Summary

Now, you should be able to:

- Describe Configuration Management (CM), its aim and practices.
- Compare the features of Centralized Server and Distributed Server models of Configuration Management.
- Define the version control terminologies used in SCM.
- Visualize the usage of Configuration Management in SDLC.

Reference

https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control

Thank You