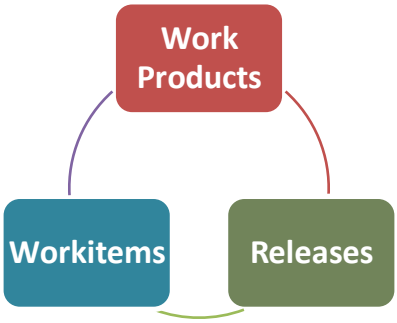
Software configuration management (SCM) is the task of tracking and controlling changes in the software and is one of the major Software Engineering topics. Good CM environments are invaluable enablers for high performance teams and organizations. However, it has received little attention in Agile software development; may be due to the heavy implementation of some CM practices specially in organizations with high level of discipline and compliance requirements (compliance with standards like ISO or CMMI).

1 SCM Definition

"In software engineering, software configuration management (SCM) is the task of tracking and controlling changes in the software" *–Wikipedia*

The basic idea behind SCM is to enable the team map workitems with work products which resulted from these workitems, and both with the software releases.



2 SCM Process Increments

2.1 What is a Process Increment?

A process increment is a process improvement chunk which can be implemented in a relatively small time (1-2 weeks) and still provide value for the organization. A process increment is independent of any other process increment, although it may have prerequisite ones.

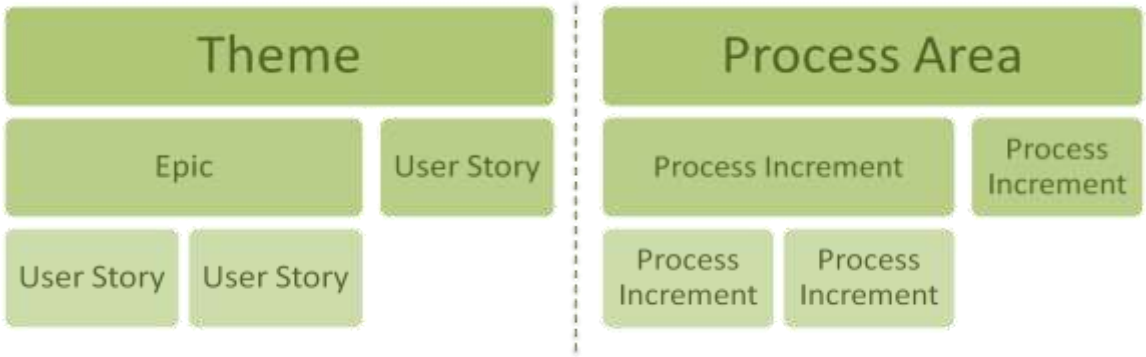
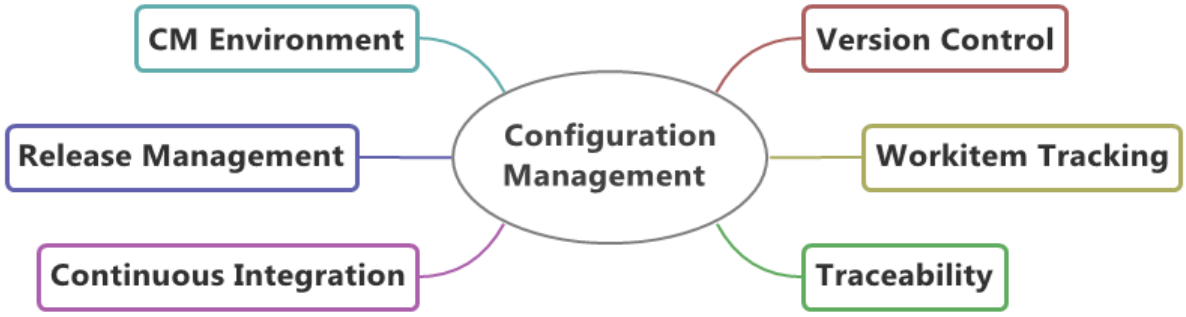


Figure 1 - Process Increments vs. User Stories

The concept of process increment in software process improvement (SPI) projects is almost identical to user stories in Agile development projects.

2.2 SCM Process Increments – A Practical Model for CM

Configuration Management can be defined in terms of the following Process Increments:



Next is a brief description for very increment.

Version Control

Project configuration items are under version control, and team is trained on basic copy-update-merge and lock-modify-unlock procedures

Workitem Tracking

Workitem types are identified, and workitems are managed and tracked

Traceability

Bi-directional traceability of requirements and work products is defined and enforced

Release Management

Release and release scope is identified; changes are received, prioritized, and planned; baselining, packaging, releasing and post-release verification procedures are enforced

CM Environment

Project structure is defined, access rights are enforced, backup/restore procedures are employed, and proper branching/merging techniques are in-action

Continuous Integration

Builds are automated; and integration between team members and between teams is automated and frequent