

Unit-V

Risk and Configuration Management



Presented by,
Prof. Barkha M Shahaji
Department of Computer Engineering,
Trinity College of Engineering and research, Pune-48.

Syllabus

| Unit V | Risks and Configuration Management | (07 Hours) |
|--|---|------------|
| <p>Risk Management: Software Risks, Risk Identification, Risk Projection, Risk Refinement, Risk Mitigation, Monitoring, and Management, The RMMM Plan.</p> <p>Software Configuration Management: Software Configuration Management, The SCM Repository The SCM Process, Configuration Management for any suitable software system.</p> <p>Suggested Free Open Source Tools: CF Engine Configuration Tool, Puppet Configuration Tool.</p> | | |
| <u>#Exemplar/Case Studies</u> | Risk management in Food delivery software | |

What is Software Risk ?

- Software risks are nothing but uncertain things that may occur in software and leads to heavy losses.
- Risks may harm cost, schedule or technical success of the project and the quality of our software device or project team.

Types of Software Risk

1. Project Risk:- budget, staff, resources, requirements, schedule and customer related problems
2. Technical Risk:- implementation, implementing the interfaces, testing & maintenance
3. Business Risk:-
 - Market Risks
 - Strategic Risks
 - Management Risks
 - Sale Risks
 - Budget Risks

Risk Management

Four activities in Risk Management

1. Risk Identification
2. Risk Projection
3. Risk Refinement
4. RMMM (Mitigation, Monitoring and Management)

Risk Identification

- Risk identification is related to discovering possible risks to the project.
- Two types of risks
 1. Generic Risk Identification
 - It includes potential threat identification to software project.
 2. Product Specific risk identification
 - It includes Product Specific threat identification by understanding people technology and working environment in which the product gets build.

Risk Projection

- Risk Projection is also called as risk estimation.
- Two methods:
 1. Probability that risk is real.
 2. Consequences of the risk

Risk Projection

- Steps involved in risk projection:
 - i. Building risk table
 - ii. Enlist the consequences of risks
 - iii. Estimate the impact of those risks on our project
 - iv. Maintaining over all accuracy of risk projection

| Components Category | | Performance | Support | Cost | Schedule |
|------------------------|---|---|---|--|--------------------------------|
| Catastrophic | 1 | Failure to meet the requirement would result in mission failure | | Failure results in increased costs and schedule delays with expected values in excess of \$500K | |
| | 2 | Significant degradation to nonachievement of technical performance | Nonresponsive or unsupportable software | Significant financial shortages, budget overrun likely | Unachievable IOC |
| Critical | 1 | Failure to meet the requirement would degrade system performance to a point where mission success is questionable | | Failure results in operational delays and/or increased costs with expected value of \$100K to \$500K | |
| | 2 | Some reduction in technical performance | Minor delays in software modifications | Some shortage of financial resources, possible overruns | Possible slippage in IOC |
| Marginal | 1 | Failure to meet the requirement would result in degradation of secondary mission | | Costs, impacts, and/or recoverable schedule slips with expected value of \$1K to \$100K | |
| | 2 | Minimal to small reduction in technical performance | Responsive software support | Sufficient financial resources | Realistic, achievable schedule |
| Negligible | 1 | Failure to meet the requirement would create inconvenience or nonoperational impact | | Error results in minor cost and/or schedule impact with expected value of less than \$1K | |
| | 2 | No reduction in technical performance | Easily supportable software | Possible budget underrun | Early achievable IOC |

Note: (1) The potential consequence of undetected software errors or faults.
 (2) The potential consequence if the desired outcome is not achieved.

Risk Refinement

- Process of explaining the risk in detailed way.
- It is represented by CTC (Condition Transition Consequences)
- CTC format:
<condition>then there is concern that
<consequences>

RMMM

- Risk Mitigation, Monitoring and Management
- Risk Mitigation:
It will prevent the risk from occurrence.
- Risk Monitoring:
It will happened after a risk has been occurred.
- Risk Management:
If mitigation done properly then management becomes easy.

RMMM Plan

- It is a document in which all the risk analysis activities are described
- Risk Information Sheet is used for describing each risk

Risk Information Sheet format

| | | | |
|--------------------------|-------------|--------------|--------|
| Project Name | | | |
| Risk ID | Date | Probability | Impact |
| Origin | Assigned To | | |
| Description | | | |
| Refinement/context | | | |
| Mitigation/monitoring | | | |
| Trigger/contingency plan | | | |
| Status | | | |
| Originator | | closing date | |

SCM

Software Configuration Management:

- When we develop software, the product (software) undergoes many changes in their maintenance phase; we need to handle these changes effectively.
- It is a set of activities carried out for identifying, organizing, and controlling changes throughout the computer software life cycle.

SCM

What is SCM?

- 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.



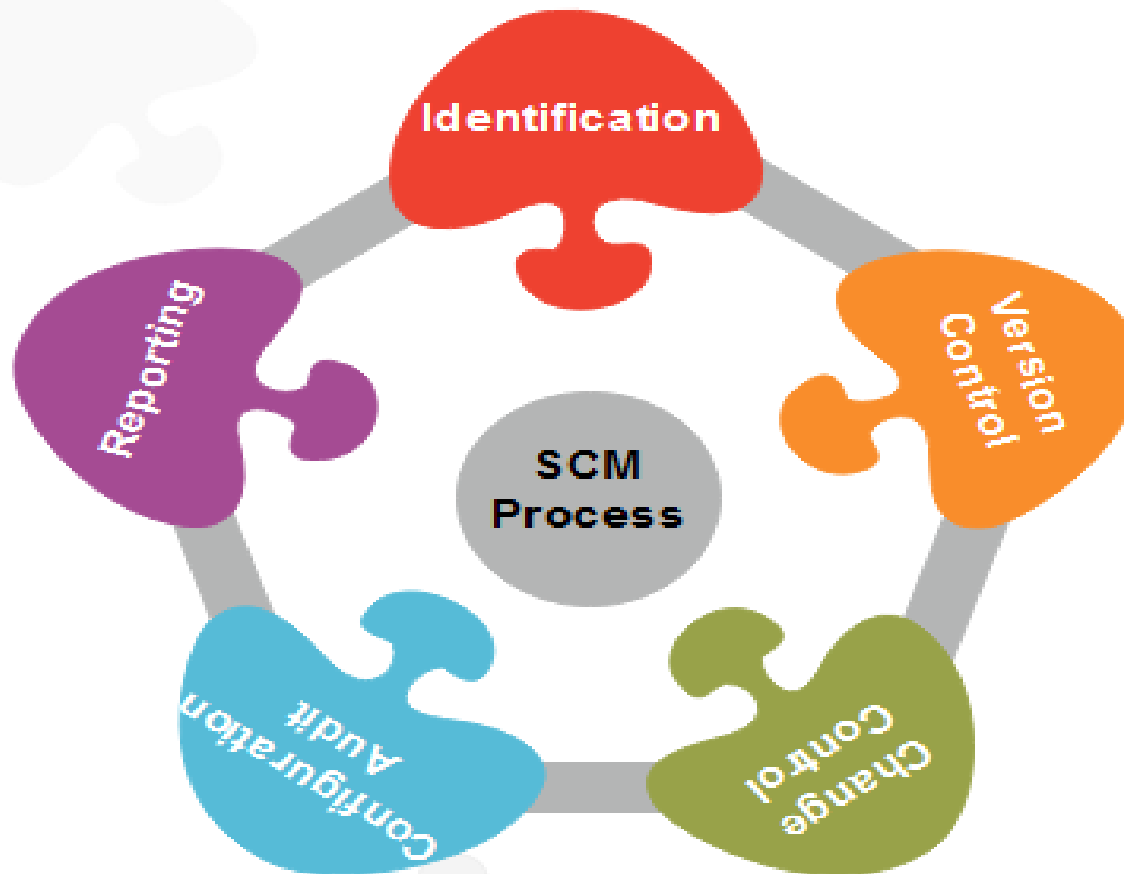
Importance of SCM

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.

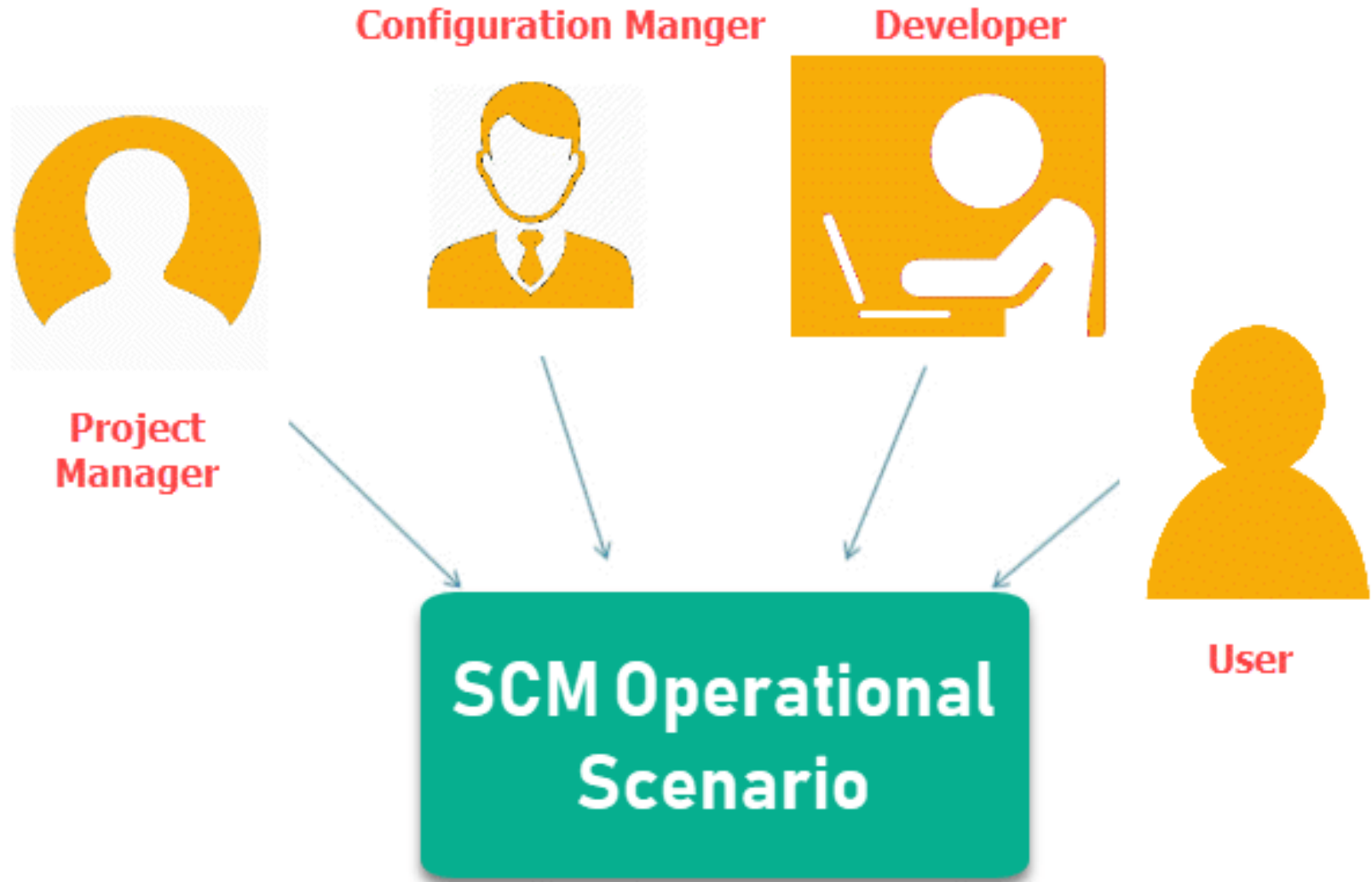


SCM Process

Software Configuration Management Process



Participant of SCM Process



Participant of SCM Process

1. Configuration manager

- He is the head who is responsible for identifying configuration items.
- Cm ensures team follows the SCM process
- He/she needs to approve or reject change request

2. Developer

- The developer needs to change the code as per standard development activities or change requests.
- He is responsible for maintaining configuration of code
- The developer should check the changes and resolves conflicts.

3. Auditor

- The auditor is responsible for SCM audits and reviews.
- Need to ensure the consistency and completeness of release.

4. Project manager

- Ensure that product is developed within a certain time frame.
- Monitors the progress of development and recognizes issues in the SCM process.
- Generate reports about the status of the software system
- Make sure that processes and policies are followed for creating, changing and testing

5. User

- The end user should understand the key SCM terms to ensure he has the latest version of the software

SCM

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

Examples:

1. Instead of naming a File login.php its should be named login_v1.2.php where v1.2 stands for the version number of the file
2. Instead of naming folder “Code” it should be named “Code_D” where D represents code should be backed up daily.

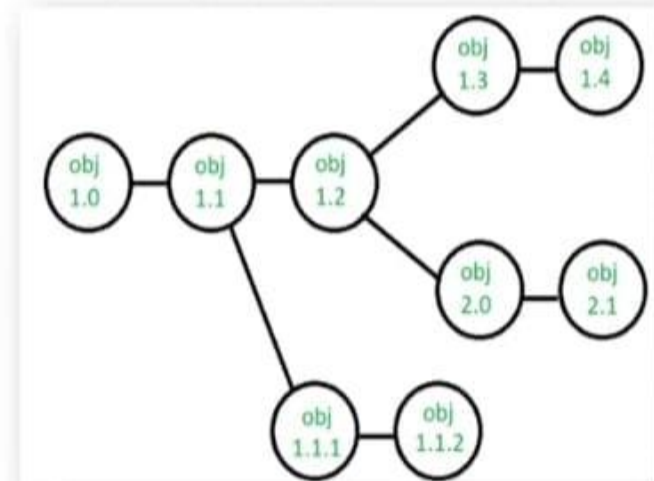
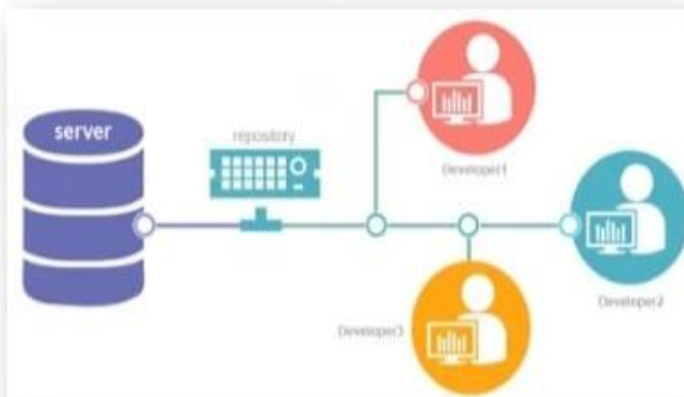
SCM

SCM Process

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.

Example:

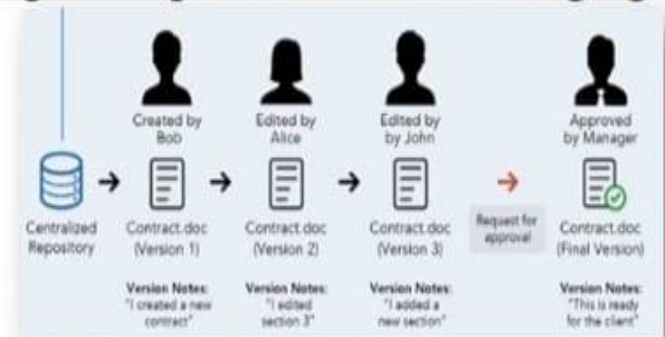


SCM

SCM Process

3. Change Control Process:

- This method used to ensure that any changes that are made are consistent with the rest of the project.
- **Examples:** To add or edit various configuration items, Change user permissions or Changing requirements of clients.



Process:

1. Software Team send changes to the Software Configuration Manager (SCM).
2. SCM checking examining the overall impact they will have on the project.
3. Making approved changes or explaining why change requests were denied to the team.
4. If it is approved them implement all necessary changes.
5. After that Review or Reporting it.

SCM

SCM Process

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:

- Its 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

- It manage version control, change control & release control process
- **SCM repository** Software configuration items are maintained in project repository or project library.
- It is basically database that act as a center for both accumulation and storage for software engineering information.

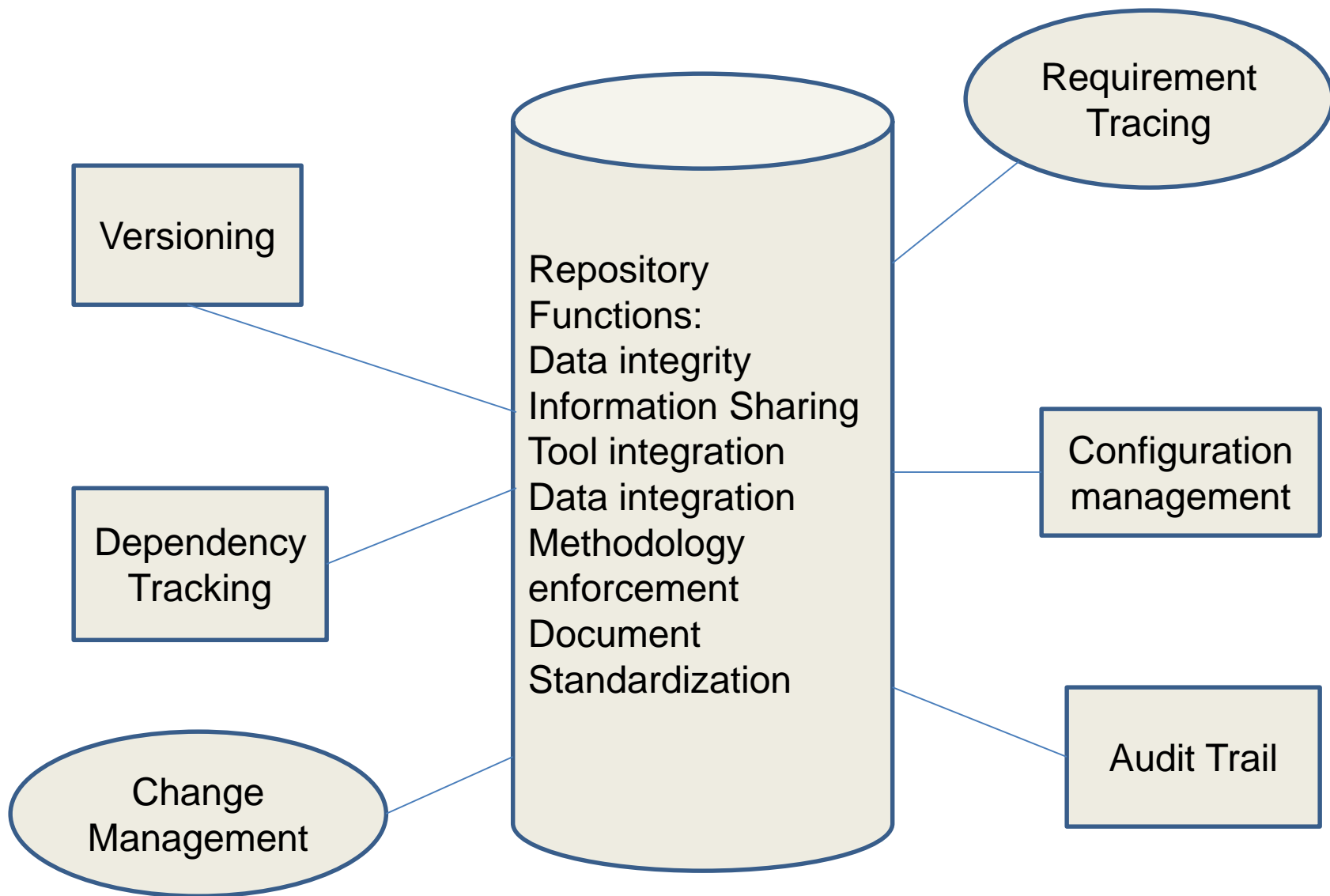


Fig: Functions of SCM repository

1. Versioning

- A project progresses various versions of individuals work products may get created
- The repository must be able to maintain all these versions and permit the developer to go back to previous versions during testing and debugging.

2. Dependency Tracking & change management

- Data elements those are stored in the repository are related to each other.
- The repository must have this ability to trace these relationships and maintain data integrity.

Features

3. Requirement Tracing

- If design of constructed components are traced then their particular requirements can be trace out.
- The repository must have this ability.

4. Configuration Management

- The repository must be able to keep the track of series of configurations representing project milestones and production releases.

controls multi user access.

3. Tool integration

- Establishes a data model that can be accessed by many software engineering tools, controls access to a data.

4. Data integration

- Allows various SCM tasks to be performed on one or more CSCIs.

5. Methodology enforcement

- Defines an entity-relationship model for the repository that implies a specific process model for software engineering.

Repository Functions: