# CITS5501 Software Testing and Quality Assurance
# Standards and quality control systems

Unit coordinator: Arran Stewart

April 17, 2018

# Overview

Based on material from:

- Stephen Dannelly, Winthrop University
- Pressman, R., *Software Engineering: A Practitioner's Approach*, McGraw-Hill, 2005

Questions:

- If I am working in an organization that adheres to the ISO 9001 standard, what is involved in that?

# What ISO 9000 mandates

ISO:

The requirements for a quality system have been standardized - but many organizations like to think of themselves as unique. So how does ISO 9001:2008 allow for the diversity of say, on the one hand, a “Mr. and Mrs.” enterprise, and on the other, to a multinational manufacturing company with service components, or a public utility, or a government administration?

The answer is that ISO 9001:2008 lays down **what** requirements your quality system must meet, but does **not** dictate **how** they should be met in any particular organization. This leaves great scope and flexibility for implementation in different business sectors and business cultures, as well as in different national cultures.

# Insuring compliance

ISO:

1. The standard requires the organization itself to audit its quality system to verify that it is managing its processes effectively - or, to put it another way, to check that it is fully in control of its activities.
2. In addition, the organization may invite its clients to audit the quality system in order to give them confidence that the organization is capable of delivering products or services that will meet their requirements.
3. Lastly, the organization may engage the services of an independent quality system certification body to obtain an ISO 9001:2008 Certificate of Conformity. This last option has proved extremely popular in the market-place because of the perceived credibility of an independent assessment.

# ISO 9001 Contents

- Section 4 — General Requirements
- Section 5 — Management Responsibility
- Section 6 — Resource Management
- Section 7 — Product Realization
- Section 8 — Measurement, Analysis and Improvement

# ISO 9003

- ISO/IEC 90003 – "Software engineering – Guidelines for the application of ISO 9001:2008 to computer software"
- Generally, mirrors the structure of ISO 9001

# ISO 90003

Section 7 - Product Realization

```
7.1 Product Realization Planning
7.2 Customer Processes
  7.2.2 Review of Software Product Requirements
     7.2.2.1 Review Product Requirements related to
             Customer Contract
7.3 Software Design and Development
7.4 Purchasing Parts and Components
7.5 Product and Service Provisions
    * tracking builds, deliveries, releases
7.6 Monitoring and Measuring
```

see http://www.praxiom.com/iso-90003.htm

# ISO 90003

## Section 8 - Measurement, Analysis, and Improvement

8.1 Carry out remedial processes
  * Plan how monitoring, measuring, and analytical processes will be used to demonstrate conformity.
  * Use monitoring, measuring, and analytical processes to demonstrate conformance.

8.2 Monitor and measure quality
  8.2.1 Monitor and measure customer satisfaction.
  8.2.2 Plan and perform regular internal audits.
  8.2.3 Monitor and measure quality processes.
  8.2.4 Monitor and measure product characteristics.

8.3 Control your nonconforming software products
  * Prevent the delivery or use of nonconforming software products.

8.4 Analyze quality information

8.5 Take required remedial actions

see: http://www.praxiom.com/iso-90003.htm

# 9001 Required Documents

1. Quality Policy
2. Control of Documents
3. Control of Records
4. Internal Audits
5. Control of Nonconforming Product / Service
6. Corrective Action
7. Preventive Action

These may go in a single “Quality Manual”.

see http://en.wikipedia.org/wiki/ISO_9000

# Documentation

- **Quality policy**
  - intended for all levels of employees
  - linked to business plan, marketing plan, customer needs
  - measurable objectives
- **Records**
  - allows problems to be traced back to causes
  - includes
    - test results, customer comments, etc.
    - actions taken to improve
- **Internal Audits**
  - is the system working?
  - what improvements can be made?

# Is it worth it?

- Good business judgment is needed to determine ISO9001's proper role for a company.
- Is certification important to the marketing plans of the company? If not, do not rush to certification.
- Even without certification, companies should utilize the ISO 9001 model as a benchmark to assess the adequacy of its quality programs.

— Frank Barnes

# Capability Maturity Model

- **1986** - Effort started by SEI and MITRE Corporation
  - assess capability of DoD contractors
- First version published in 1991
- closely related to **TQM**
  - goal is customer satisfaction
    - not required that customer be "delighted"
- Process improvement is based on small steps, rather than revolutionary innovation.
- CMM is not exhaustive or dictatorial.
- CMM focuses on processes that are of value across the organization.

# Levels

1. Initial
2. Repeatable
3. Defined
4. Managed
5. Optimizing

# Level 1: The Initial Level

- ad hoc, sometimes chaotic
- overcommitment leads to a series of crises
- during a crisis, projects abandon plans
- capability is characteristic of individuals, not the organization
  - when a good manager leaves, the success leaves with them

# Level 2: The Repeatable Level

- Planning is based on experience with similar projects
  - past successes can be repeated
- Policies for managing and implementation
  - installed basic management controls
  - track costs and schedules
  - notice and deal with problems as they arise

# Level 3: The Defined Level

- Standard processes defined across the organization and used by all projects
  - standard set of roles, activities, quality tracking, etc
  - each project uses a tailored version of this standard process
- Training program is in place to ensure everyone has the skills required for their assigned role

# Level 4: The Managed Level

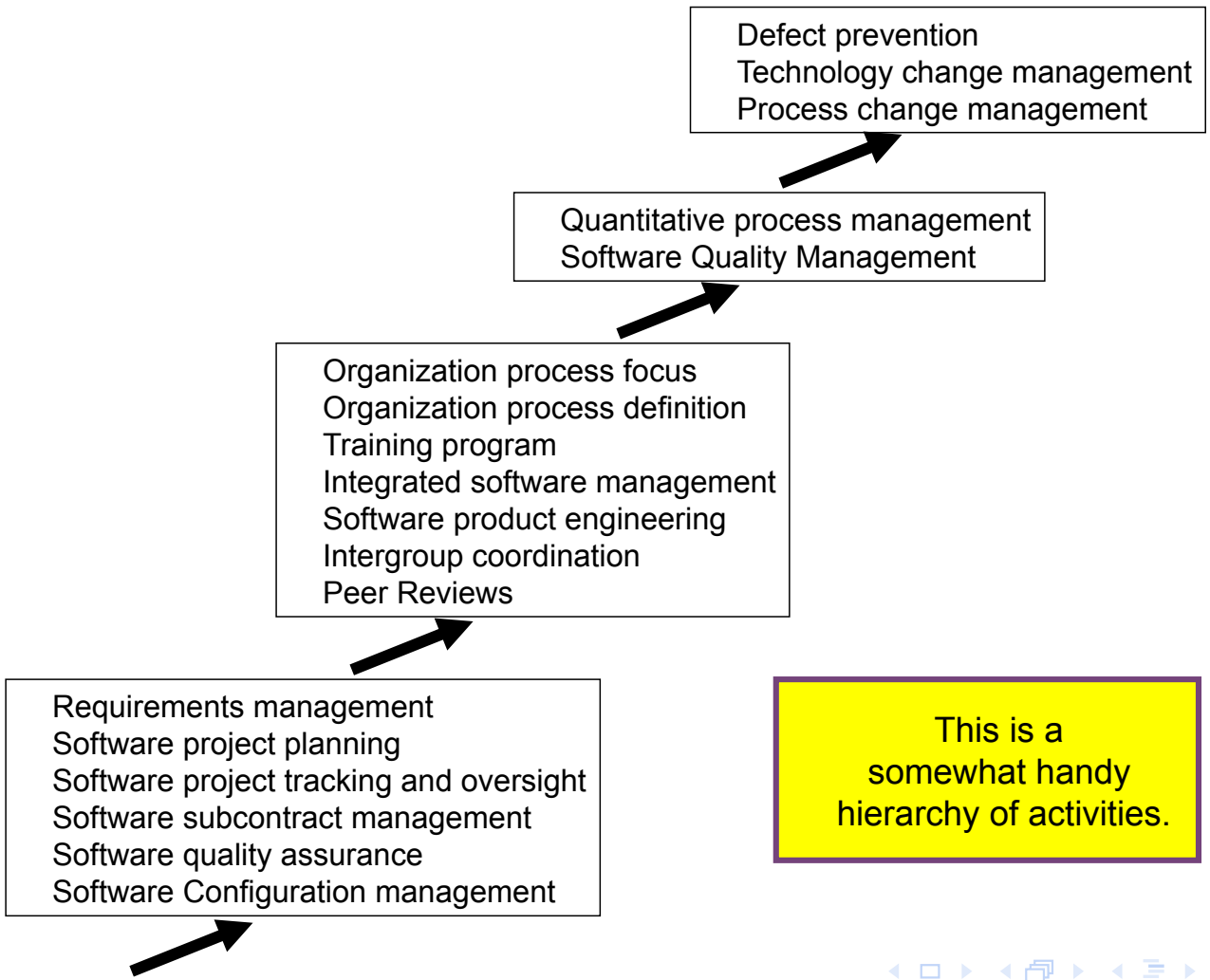
- Quantitative quality goals
  - for both Products and Processes
- Organization-wide process database
  - meaningful variations in process performance can be distinguished from random noise
  - actions are then taken to correct the situation
- Products are of predictably high quality

# Level 5: The Optimizing Level

- Organization has the means to identify weaknesses and strengthen the process proactively
- teams analyze defects to determine their cause, and disseminate lessons learned throughout the organization
- major focus on eliminating waste
  - e.g. reduce amount of rework

# Process areas

Key process areas by maturity level:



# Level Comparison - Risk

- Level 1
  - Just do it
- Level 2
  - problems are recognized and corrected as they occur
- Level 3
  - problems are anticipated and prevented, or impacts minimized
- Levels 4 and 5
  - sources of problems are understood and eliminated

# Level Comparison - People

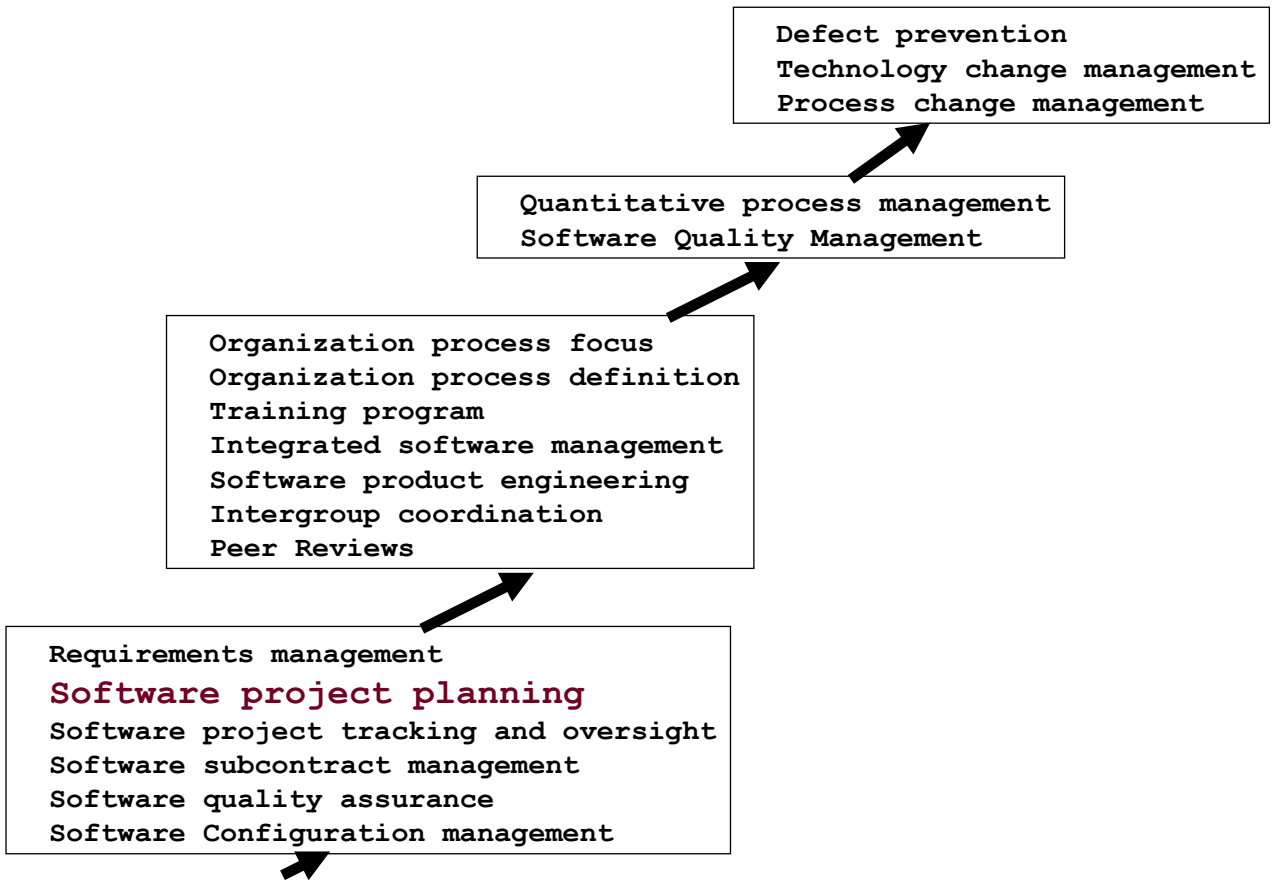
- Level 1
  - success depends on individual heroics
  - fire fighting is the way of life
- Level 2
  - success depends on individuals
  - efforts are supported by management
- Level 3
  - people are trained for their role(s)
  - groups work together
- Levels 4
  - strong sense of teamwork in every project
- Level 5
  - strong sense of teamwork across the organization
  - everyone does process improvement

# Level Comparison - Measurement

- Level 1
  - ad hoc (if any) data collection and analysis
- Level 2
  - individual projects use planning data
- Level 3
  - data collected for all processes
  - data shared across projects
- Levels 4
  - data standardized across the organization
- Level 5
  - data used for process improvement

# Key process areas

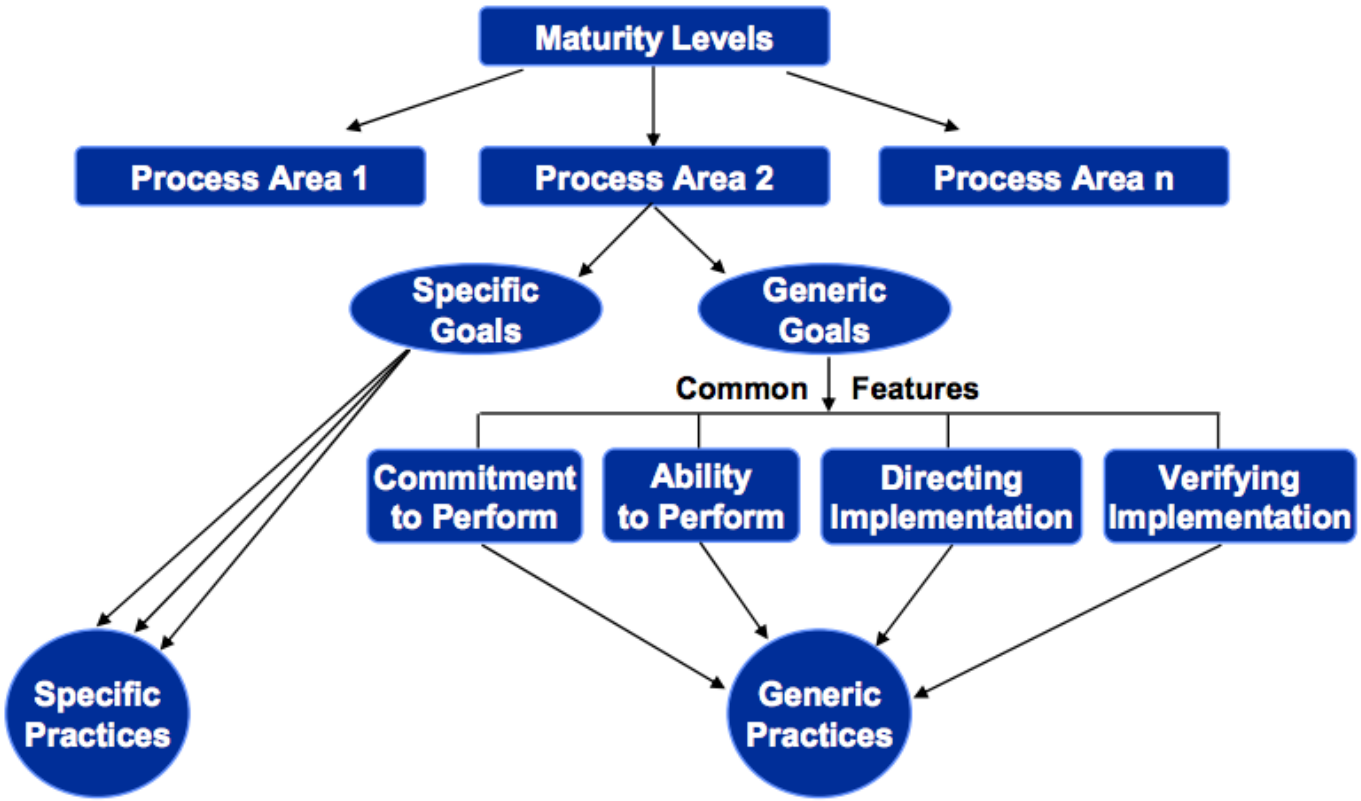
Key process areas by maturity level:



# CMMI vs CMM

- The Software Engineering Institute at Carnegie Mellon University developed Capability Maturity Model Integration (CMMI) in 2006 to integrate and standardize the separate models of CMM, and to eradicate other drawbacks of CMM.
- CMMI documents industry best practices categorized on separate areas of interests rather than separate functions. Organizations choose from any of the 22 available models depending on the business objectives, and each model covers all the functional areas.
- CMM is a certification and CMMI is an appraisal.
- Both involve a lot of paperwork.

# CMMI Structure



# CMMI Common Features

- Commitment to Perform: Commitment to Perform describes the actions the organization must take to ensure that the process is established and will endure.
- Ability to Perform: Ability to Perform describes the preconditions that must exist in the project or organization to implement the software process competently.
- Activities Performed: Activities Performed describes the roles and procedures necessary to implement a key process area.
- Measurement and Analysis: Measurement and Analysis describes the need to measure the process and analyze the measurements.
- Verifying Implementation: Verifying Implementation describes the steps to ensure that the activities are performed in compliance with the process that has been established.

# Example: Configuration Management (Level 2)

**Purpose:** The purpose of Configuration Management (CM) is to establish and maintain the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits.

**Specific Practices by Goal:**

- SG 1 Establish Baselines
  - SP 1.1 Identify Configuration Items
  - SP 1.2 Establish a Configuration Management System
  - SP 1.3 Create or Release Baselines
- SG 2 Track and Control Changes
  - SP 2.1 Track Change Requests
  - SP 2.2 Control Configuration Items
- SG 3 Establish Integrity
  - SP 3.1 Establish Configuration Management Records
  - SP 3.2 Perform Configuration Audits

# Change Management

- Also called software configuration management (SCM)
- It is an umbrella activity that is applied throughout the software process
- Its goal is to maximize productivity by minimizing mistakes caused by confusion when coordinating software development
- SCM identifies, organizes, and controls modifications to the software being built by a software development team
- SCM activities are formulated to identify change, control change, ensure that change is being properly implemented, and report changes to others who may have an interest

# Change management (2)

- SCM is initiated when the project begins and terminates when the software is taken out of operation
- The Output from the software process makes up the software configuration
  - Computer programs (both source code files and executable files)
  - Work products that describe the computer programs (documents targeted at both technical practitioners and users)
  - Data (contained within the programs themselves or in external files)
- The major danger to a software configuration is change
  - First Law of System Engineering: "No matter where you are in the system life cycle, the system will change, and the desire to change it will persist throughout the life cycle"

# Origins of Software Change

- Errors detected in the software need to be corrected
- New business or market conditions dictate changes in product requirements or business rules
- New customer needs demand modifications of data produced by information systems, functionality delivered by products, or services delivered by a computer-based system
- Reorganization or business growth/downsizing causes changes in project priorities or software engineering team structure
- Budgetary or scheduling constraints cause a redefinition of the system or product

# Elements of a Configuration Management System

- Configuration elements
  - A set of tools coupled with a file management (e.g., database) system that enables access to and management of each software configuration item
- Process elements
  - A collection of procedures and tasks that define an effective approach to change management for all participants
- Construction elements
  - A set of tools that automate the construction of software by ensuring that the proper set of valid components (i.e., the correct version) is assembled
- Human elements
  - A set of tools and process features used by a software team to implement effective SCM

# Baseline

- An SCM concept that helps practitioners to control change without seriously impeding justifiable change
- IEEE Definition: A specification or product that has been formally reviewed and agreed upon, and that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures
- It is a milestone in the development of software and is marked by the delivery of one or more computer software configuration items (CSCIs) that have been approved as a consequence of a formal technical review
- A CSCI may be such work products as a document (as listed in MIL-STD-498), a test suite, or a software component

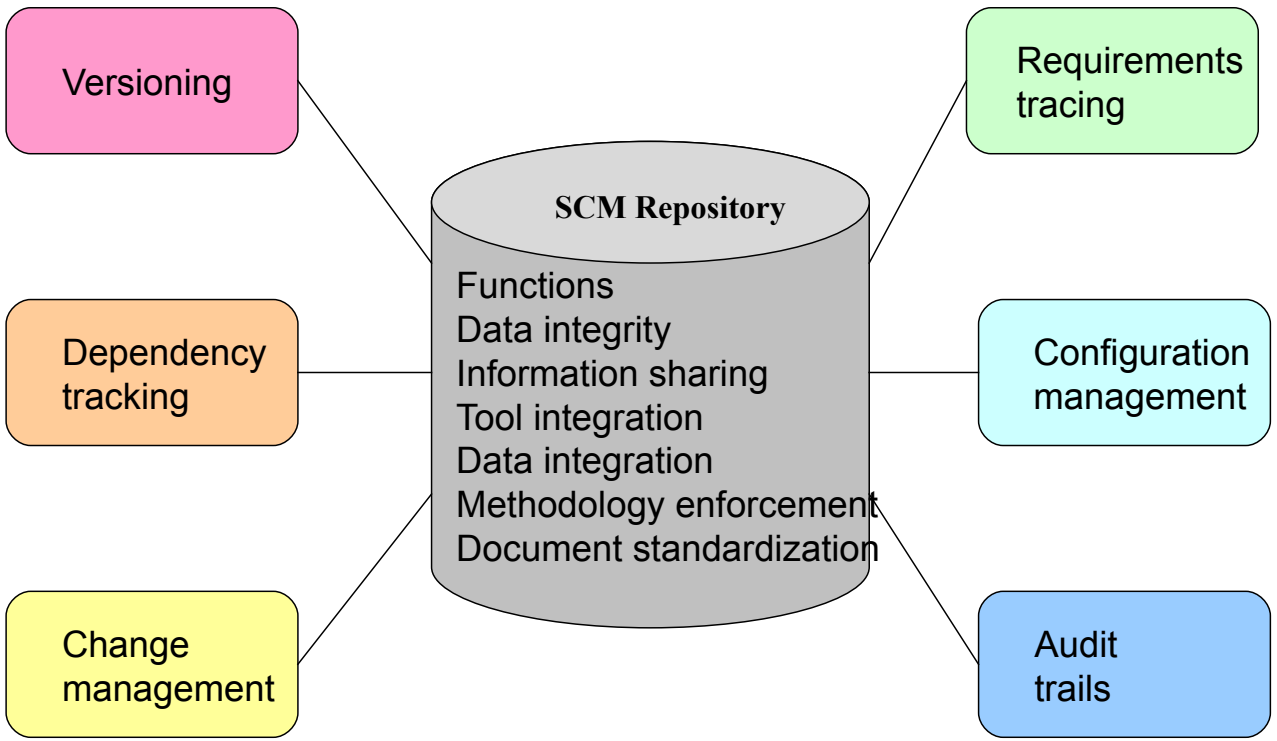
# Baselining Process

1. A series of software engineering tasks produces a CSCI
2. The CSCI is reviewed and possibly approved
3. The approved CSCI is given a new version number and placed in a project database (i.e., software repository)
4. A copy of the CSCI is taken from the project database and examined/modified by a software engineer
5. The baselining of the modified CSCI goes back to Step #2

# Paper-based vs. Automated Repositories

- Problems with paper-based repositories (i.e., file cabinet containing folders)
  - Finding a configuration item when it was needed was often difficult
  - Determining which items were changed, when and by whom was often challenging
  - Constructing a new version of an existing program was time consuming and error prone
  - Describing detailed or complex relationships between configuration items was virtually impossible
- Today's automated SCM repository
  - It is a set of mechanisms and data structures that allow a software team to manage change in an effective manner
  - It acts as the center for both accumulation and storage of software engineering information
  - Software engineers use tools integrated with the repository to interact with it

# Automated SCM Repository




1

# Functions of an SCM Repository

- Data integrity
  - Validates entries, ensures consistency, cascades modifications
- Information sharing
  - Shares information among developers and tools, manages and controls multi-user access
- Tool integration
  - Establishes a data model that can be accessed by many software engineering tools, controls access to the data
- Data integration
  - Allows various SCM tasks to be performed on one or more CSCIs
- Methodology enforcement
  - Defines an entity-relationship model for the repository that implies a specific process model for software engineering
- Document standardization
  - Defines objects in the repository to guarantee a standard approach for creation of software engineering documents