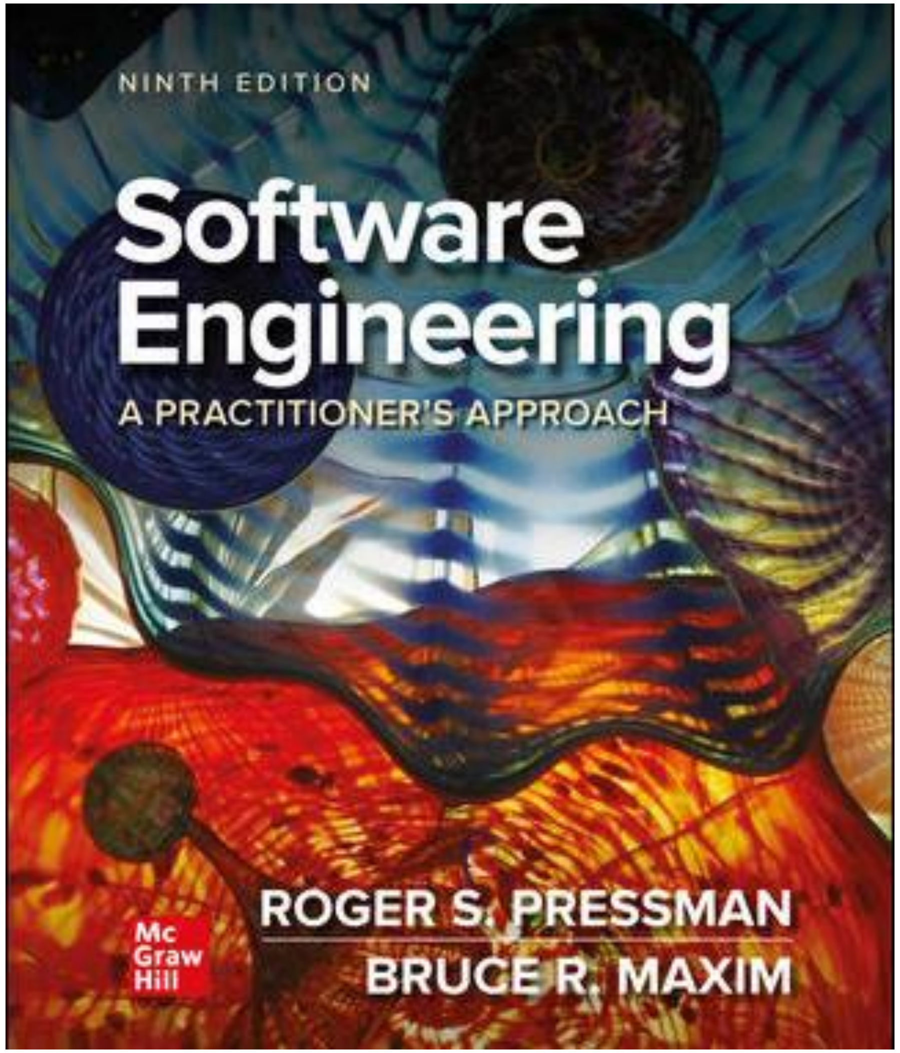
Software Engineering

Vinaya Sathyanarayana

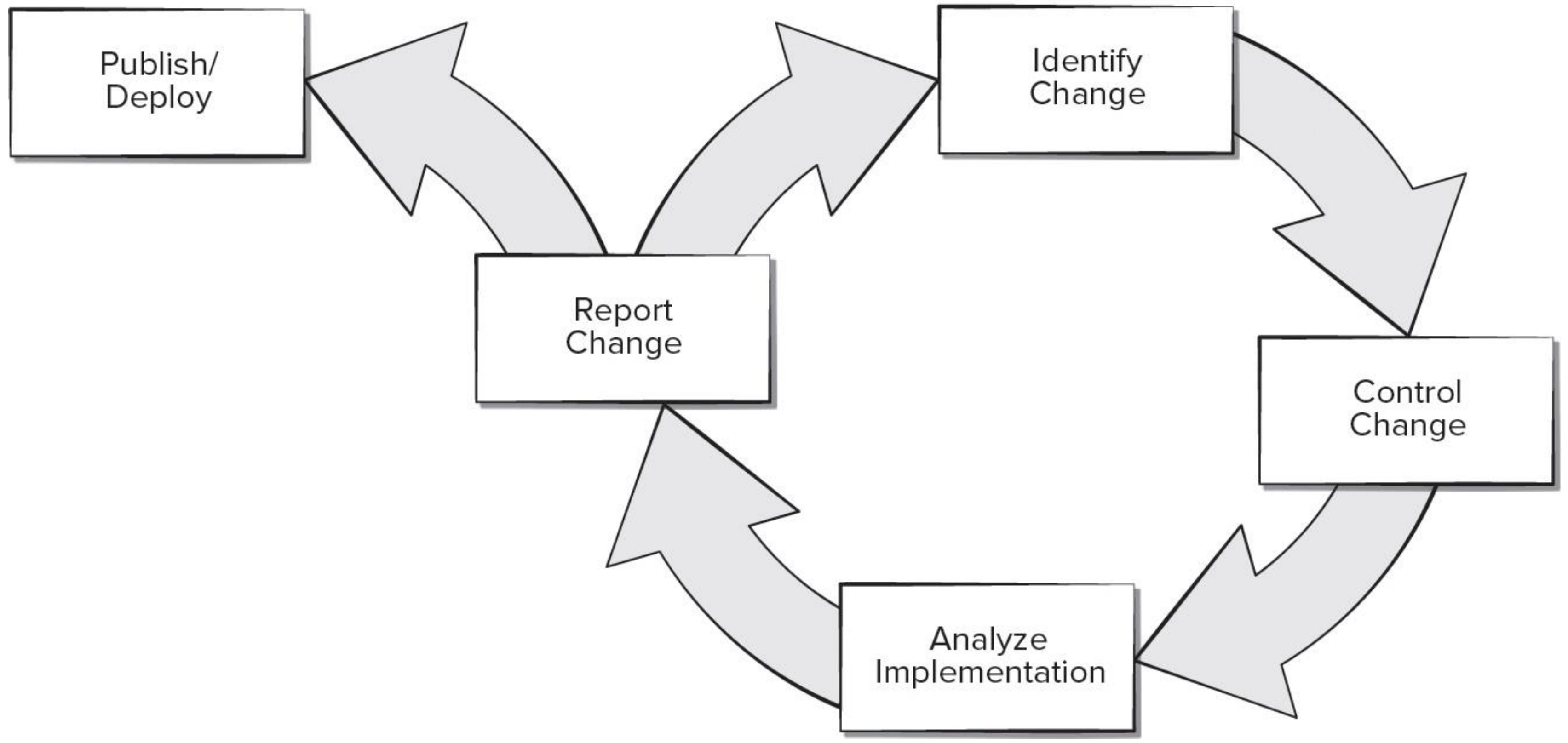
Chapter 22

Software Configuration Management.

- **Part Three – Quality and Security.**



Software Configuration Management Work Flow



- Access the text alternative for slide images.

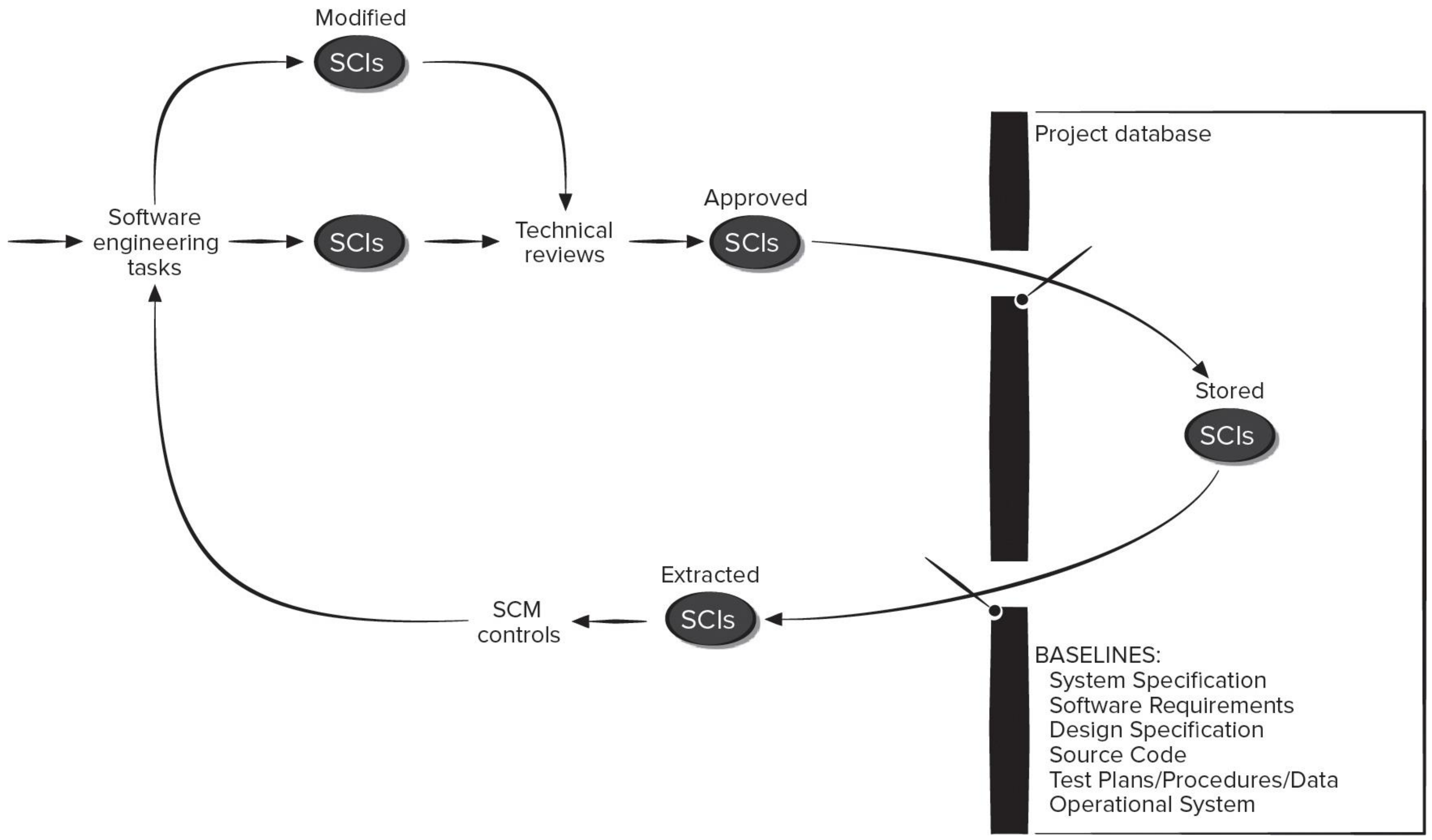
Configuration Management System Elements

- **Component elements.** A set of tools coupled within a file management system (for example, a database) 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 (and related activities) for all constituencies involved in the management, engineering, and use of computer software.
- **Construction elements.** A set of tools that automate the construction of software by ensuring that the proper set of validated components (that is, the correct version) have been assembled.
- **Human elements.** A set of tools and process features (encompassing other CM elements) used by the software team to implement effective SCM.

Baselines

- The IEEE defines a baseline as:
  A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures.
- A baseline is a milestone in the development of software that is marked by the delivery of one or more software configuration items and the approval of these S C I that is obtained through a formal technical review.

Baselined Software Configuration Items

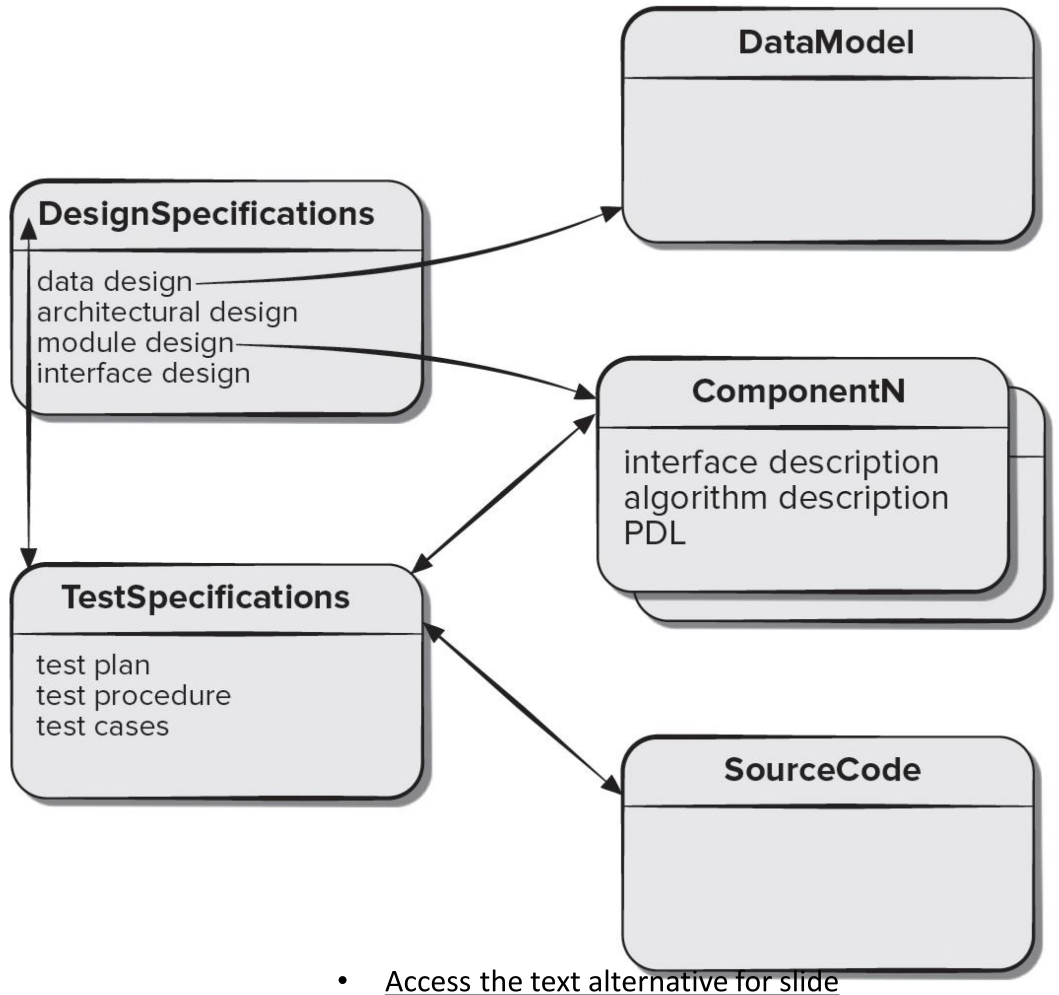


- Access the text alternative for slide images.

Management of Dependencies and Changes

- It is important for developers to maintain software work products to ensure the dependencies among the S C I are documented.
- Developers must establish discipline when checking items in and out of the SCM repository.
- Impact management involves two complementary aspects:
  1. Ensuring that developers employ strategies to minimize the impact of their colleagues' actions on their own work.
  2. Encouraging software developers to use practices that minimize the impact of their own work on that of their colleagues.

Software Configuration Items

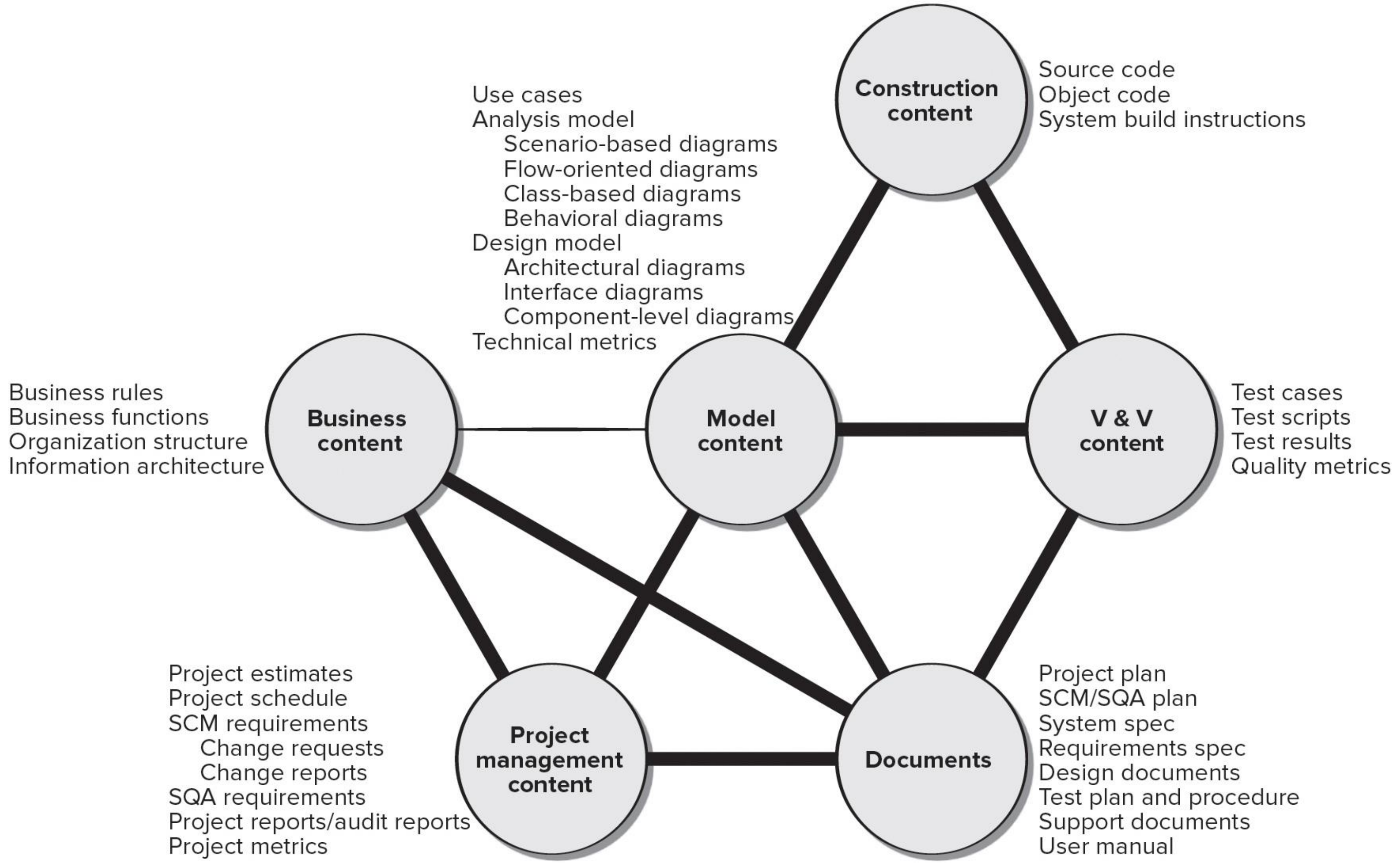


- Access the text alternative for slide images.

SCM Repository 1

- The SCM repository is the set of mechanisms and data structures that provides the following functions that allow a software team to manage change:
- Data integrity.
- Information sharing.
- Tool integration.
- Data integration.
- Methodology enforcement.
- Document standardization.

SCM Repository 2



- Access the text alternative for slide images.

SCM Repository Features

- **Versioning. -** saves versions to manage product releases and allow developers to go back to previous versions.
- **Dependency tracking and change management. -** manages a wide variety of relationships among the data elements stored in it.
- **Requirements tracing. -** provides the ability to track all design and construction components and deliverables resulting from a specific requirement specification.
- **Configuration management. -** tracks series of configurations representing specific project milestones or production releases and provides version management.
- **Audit trails. -** establishes additional information about when, why, and by whom changes are made.

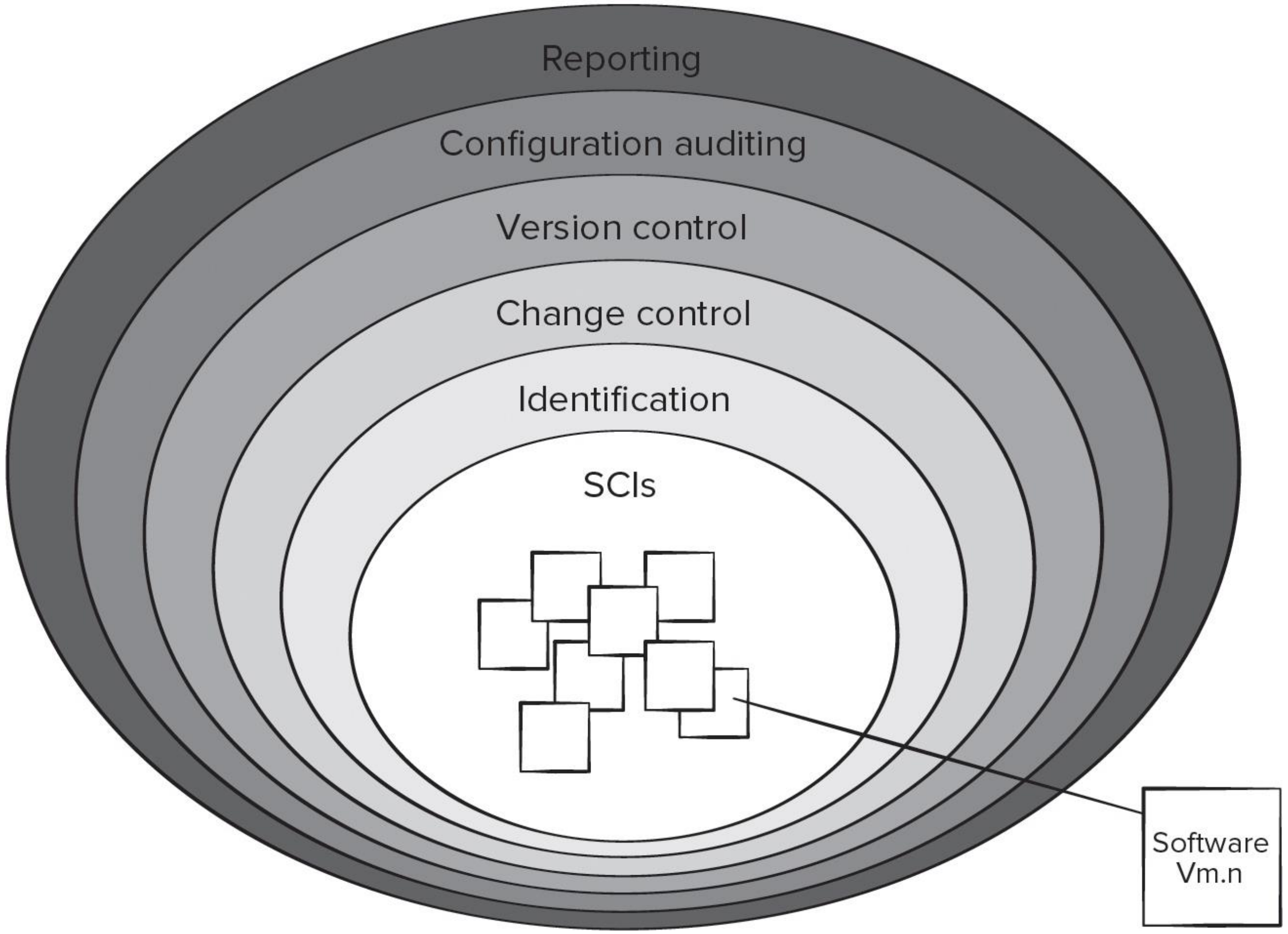
SCM Best Practices

- Keeping the number of code variants small.
- Test early and often.
- Integrate early and often.
- Use tools to automate testing, building, and code integration.

Continuous Integration Advantages

- **Accelerated feedback.** Notifying developers immediately when integration fails allows fixes when the number of changes is small.
- **Increased quality.** Building and integrating software whenever necessary provides confidence into the quality of the product.
- **Reduced risk.** Integrating components early avoids a long integration phase, design failures are discovered and fixed early.
- **Improved reporting.** Providing additional information (for example, code analysis metrics) allows for accurate configuration status accounting.
- CI is becoming a key technology as software organizations begin their shift to more agile software development processes.
- Early defect capture always reduces the development costs.

SCM Process Layers



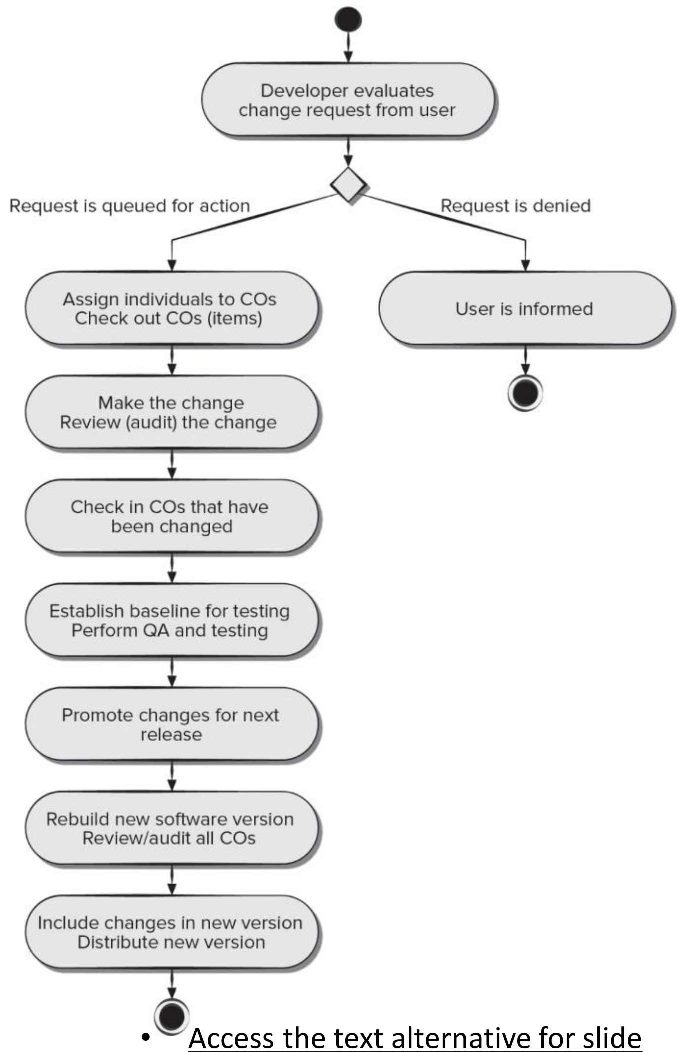
- Access the text alternative for slide images.

Change Management Objectives

- Software change management process defines a series of tasks that have four primary objectives:

1. To identify all items that collectively define the software configuration.
2. To manage changes to one or more of these items.
3. To facilitate the construction of different versions of an application.
4. To ensure that software quality is maintained as the configuration evolves over time.

Change Control Process


Developer evaluates change request from user
Request is queued for action
Request is denied
Assign individuals to COs Check out COs (items)
User is informed
Make the change Review (audit) the change
Check in COs that have been changed
Establish baseline for testing Perform QA and testing
Promote changes for next release
Rebuild new software version Review/audit all COs
Include changes in new version Distribute new version

- Access the text alternative for slide images.

Impact Management

- A web of software work product interdependencies must be considered every time a change is made.
- Impact management is accomplished with three actions:

1. An impact network identifies the stakeholders who might effect or be affected by changes that are made to the software based on its architectural documents.
2. Forward impact management assesses the impact of your own changes on the members of the impact network and then informs members of the impact of those changes.
3. Backward impact management examines changes that are made by other team members and their impact on your work and incorporates mechanisms to mitigate the impact.

Software Configuration Audit

- ***Software configuration audit*** complements a technical review by asking and answering the following questions:

1. Has the change specified in the ECO been made? Have any additional modifications been incorporated?
2. Has a technical review been conducted to assess technical correctness?
3. Has the software process been followed, and have software engineering standards been properly applied?
4. Has the change been “highlighted” in the S C I? Do the attributes of the configuration object reflect the change?
5. Have SCM procedures for noting the change, recording it, and reporting it been followed?
6. Have all related S C I been properly updated?

Configuration Status Reporting (CSR)

- ***Configuration status reporting*** is an SCM task that answers the following questions any time a change or audit occurs:

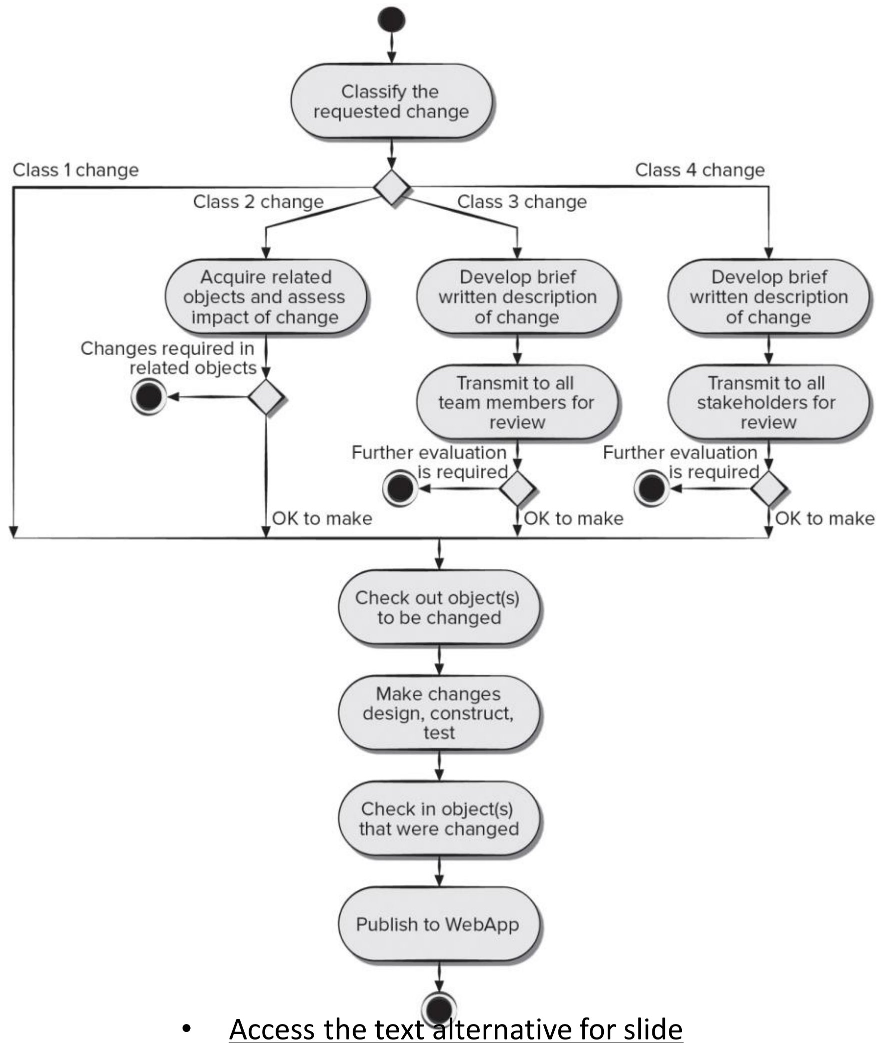
  1. What happened?
  2. Who did it?
  3. When did it happen?
  4. What else will be affected?

- Output from CSR may be placed in an online database or website, so that software developers or support staff can access change information by keyword category.

Mobility and Agile Change

- Mobile developers and game developers use an iterative, incremental process model that applies many agile principles.
- An engineering team often develops an increment in a very short time period using a customer-driven approach.
- Subsequent increments add additional content and functionality, and each is likely to implement changes that lead to enhanced content, better usability, improved aesthetics, better navigation, enhanced performance, and stronger security.
- Members of software teams that build apps or games must embrace change, but do not want excessive bureaucracy.
- Traditional SCM principles, practices, and tools must be molded them to meet the special needs of mobile projects.

Agile Change Management


Classify the requested change
Class 1 change
Class 2 change
Class 3 change
Class 4 change
Acquire related objects and assess impact of change
Develop brief written description of change
Develop brief written description of change
Changes required in related objects
Transmit to all team members for review
Transmit to all stakeholders for review
Further evaluation is required
Further evaluation is required
OK to make
OK to make
OK to make
Check out object(s) to be changed
Make changes design, construct, test
Check in object(s) that were changed
Publish to WebApp

- Access the text alternative for slide images.

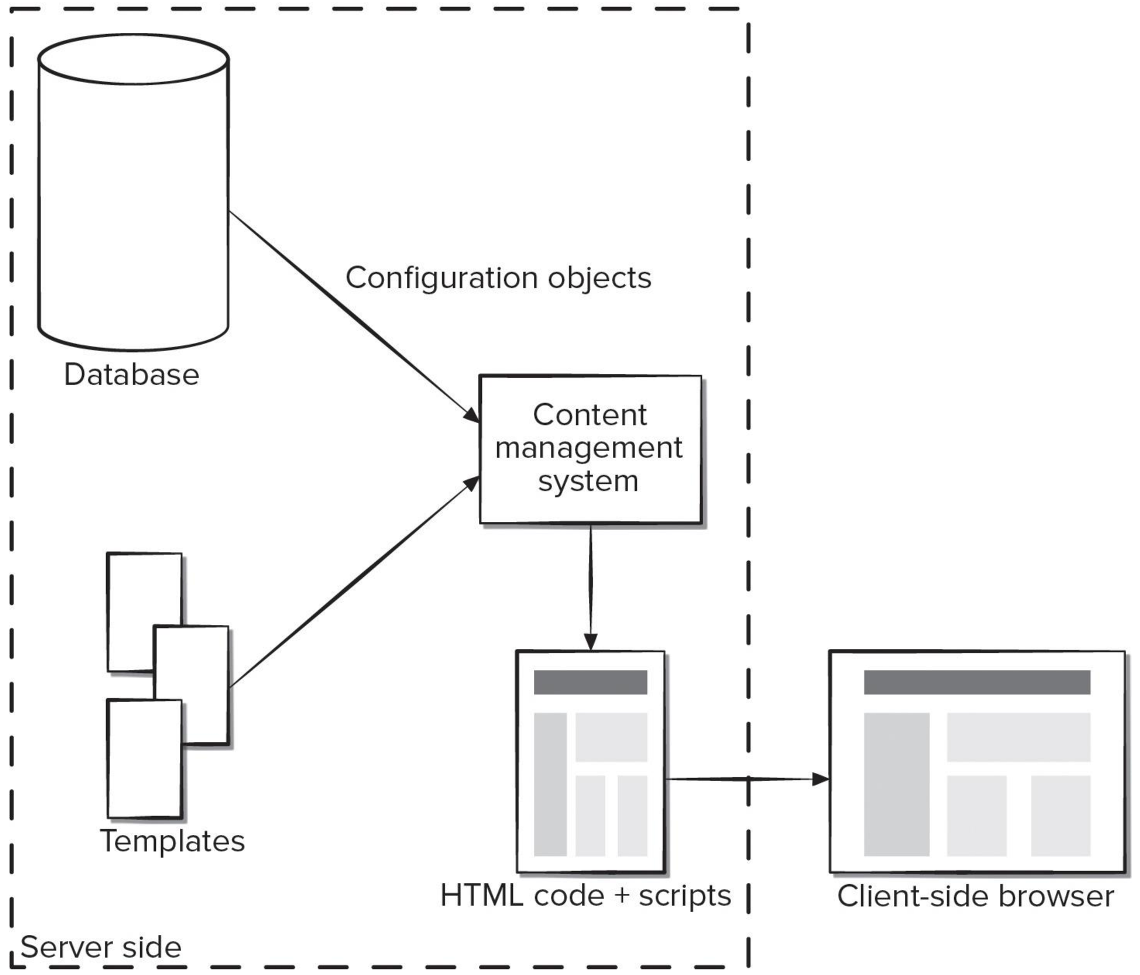
Content Management 1

- **Collection subsystem** encompasses all actions required to create or acquire content, and the technical functions that are necessary to:
  - Convert content into a form that can be represented by a mark-up language (for example, HTML, XML).
  - Organize content into packets that can be displayed effectively on the client-side.
- **Management subsystem** implements a repository that encompasses:
  - *Content database* - information structure to store all content objects.
  - *Database capabilities* - functions to search for content objects, store and retrieve objects, and manage the content file structure.
  - *Configuration management functions* - supports content object identification, version control, change management, change auditing, and reporting.

Content Management 2

- **Publishing subsystem -** extracts content from the repository, converts it to a publishable form, and formats it so that it can be transmitted to client-side browsers.
- The publishing subsystem uses a series of templates for each type:
  - *Static elements*—text, graphics, media, and scripts that require no further processing are transmitted directly to the client-side.
  - *Publication services*—function calls to specific retrieval and formatting services that personalize content (using predefined rules), perform data conversion, and build appropriate navigation links.
  - *External services*—provide access to external corporate information infrastructure such as enterprise data or “back-room” applications.

Content Management System



- Access the text alternative for slide images.

Version Control

- As apps and games evolve through a series of increments different versions may exist at the same time and version control process is required to avoid overwriting changes:

1. A central repository for the app or game project should be established.
2. Each developer creates his own working folder.
3. The clocks on all developer workstations should be synchronized.
4. As new configuration objects are developed or existing objects are changed, they are imported into the central repository.
5. As objects are imported or exported from the repository, an automatic, time-stamped log message is made.

Auditing and Reporting

- Interest of agility, auditing and reporting functions are deemphasized during development of games or apps.
- As objects are checked into or out of the repository the action is recorded in a log that can be reviewed.
- Complete log report can be generated at time one id needed o that all members of the team have a chronology of changes.
- An e-mail notification can be generated automatically any time an object is checked in or out of the repository.