

Course: Software Project Management

Week 11: Change Control and Configuration Management

Lecturer: Yimer Amedie (MSc.)

Addis Ababa Science and Technology University, Ethiopia

Contents

- Introduction
- Change control in Software Projects
- Configuration management
- Contract management



Learning Outcomes

After completing this lesson, you will be able to:

- Explain the role of change control and configuration management.
- Describe key processes in change control.
- Develop a basic change and configuration management plan
- Demonstrate understanding of contract management in software projects

Introduction

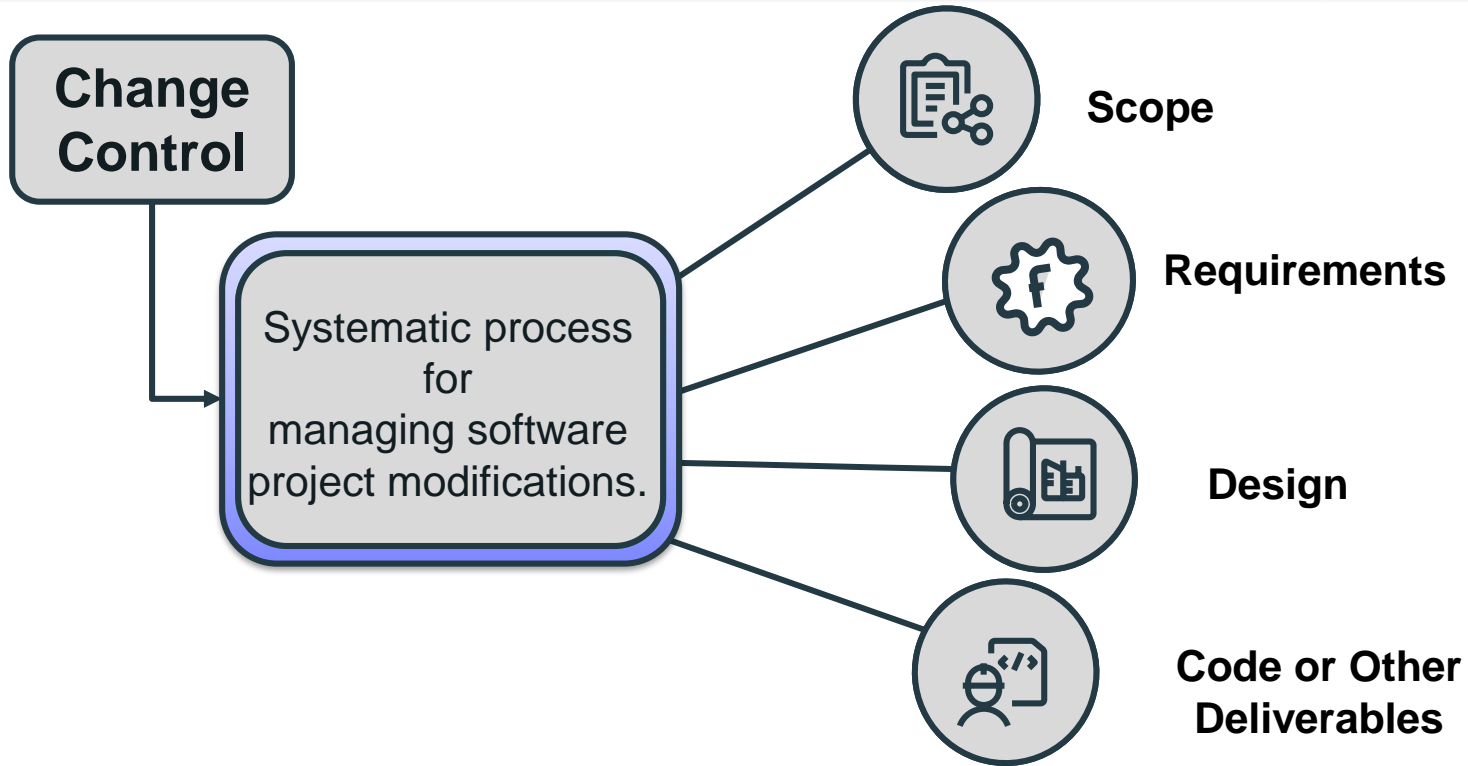
- In a project environment, change is inevitable (PMI, 2013), especially within the dynamic landscape of software development.
- The changes may be driven by
 - **Bugs or technical issue**
 - **Feedback from testing and iterations**
 - **Evolving customer needs**
 - **Technological advancements or**
 - **Internal process improvements /organizational priorities.**

Change Control in Software Projects

What is change in software projects?

- Any addition, modification, or removal that affects
 - **Project scope**
 - **Schedule**
 - **Budget**

Change Control in Software Projects



Change Control in Software Projects

Goal of change control

- To ensure that
 - Changes are
 - **Well-documented**
 - **Assessed for impact**
 - **Approved** and
 - **Implemented** without introducing unnecessary risks.
 - No unnecessary changes are made.

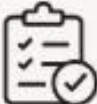

Change Control in Software Projects



Objectives:

- Control scope
- Assess impacts
- Communicate decisions
- Maintain traceability and history

Change Control in Software Projects

Types of Changes in Software Projects

	Functional Requirements
	Interface or Design
	Schedule or Timeline

	Budget or Resources
	Technology Stack

Change Control in Software Projects

Roles in Change Control



**Change Control
Board (CCB)**



**Project
Manager**



**Developers/
Testers**



Stakeholders

Change Control in Software Projects

- **The Challenge of Change in Software**
 - Projects Change is inevitable in modern software development.
 - Complexity arises from evolving requirements, shifting priorities, and rapid tech evolution.
 - Teams need structured methods to manage these dynamics effectively.



Change Control in Software Projects

- **Role of change management Frameworks**

- Provide systematic, organization-wide approach to change.

- Popular example:

- **ADKAR Model**

- **Focus on:**

- Stakeholder engagement

- Clear communication

- Risk mitigation during change implementation.



Change Control in Software Projects



Agile Methodologies and Flexibility

- Agile is built for adaptability and speed.
- Common frameworks: **Scrum, Kanban**
- Encourage:
 - Iterative development
 - Continuous feedback
 - Prioritization through sprints and backlogs



Change Control in Software Projects

Combining Agile with Change Management

Agile = executional flexibility

Change management = organizational alignment

Together, they:

- Ensure rapid response and strategic control
- Turn disruption into a competitive advantage

Change Control in Software Projects

CHANGE REQUEST FORM

Change Title: _____

Date: _____

Requestor: _____

Contact Information: _____

Priority: _____

Description of Change: _____

Reason for Change: _____

Impact of Change: _____

RFC



Change Request Form



Change Log

Tools for Change Control



Impact Analysis Template

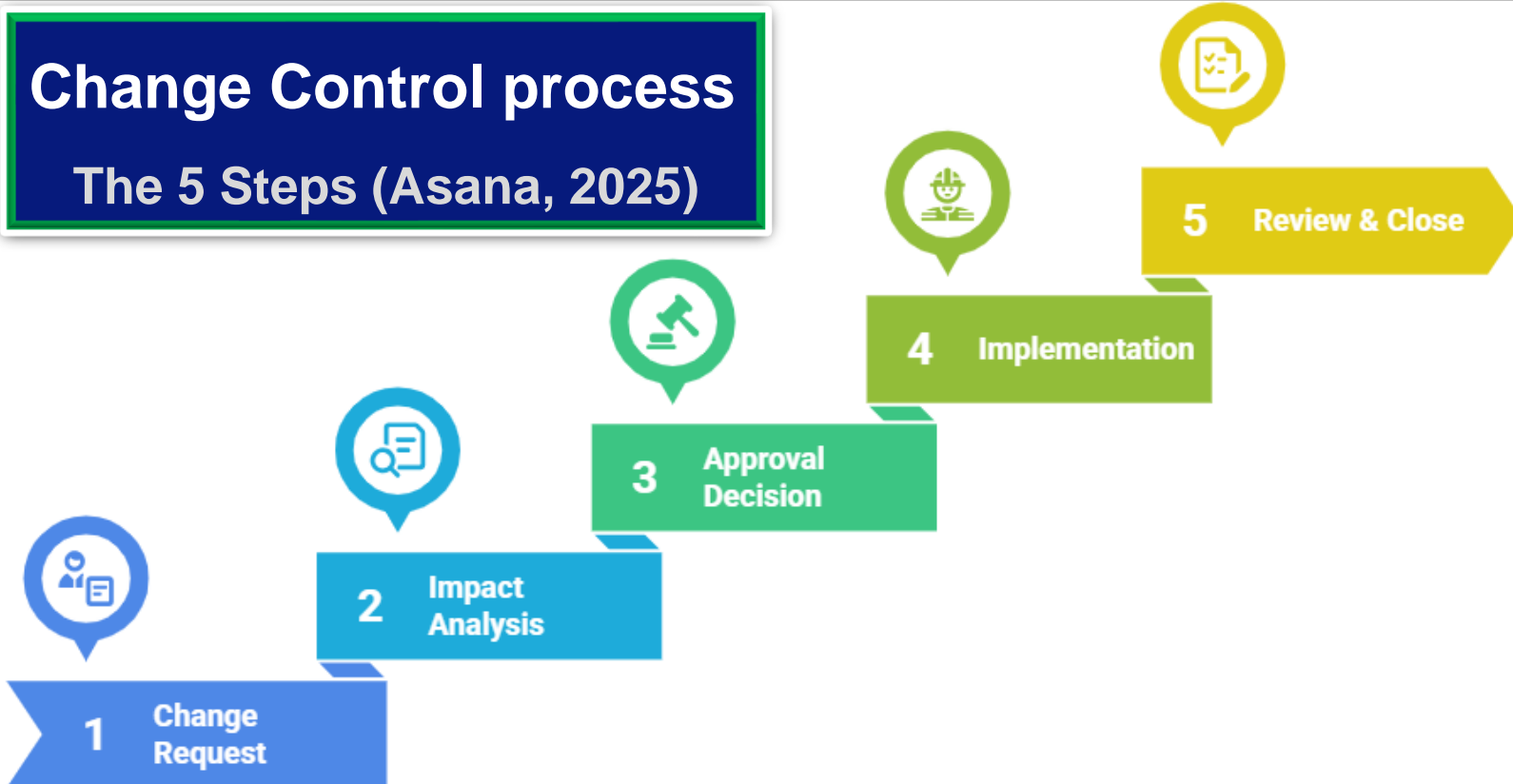


Version Control Systems

Change Control in Software Projects

Change Control process

The 5 Steps (Asana, 2025)



Change Control in Software Projects

1. Change request submission

- A stakeholder identifies a need for a change and submits a formal Change Request (CR).
- The request should include:
 - Description of the change
 - Reason/justification
 - Impact on scope, schedule, cost, quality, risks
 - Proposed solution or alternatives

Change Control in Software Projects

Example OLMS Change request: **Zoom Integration Requested**

- **By:** Yimer A., OLMS Admin
- **Date:** May 30, 2025
- **Change:** Add Zoom video conferencing to OLMS for seamless virtual classes.
- **Description:** To improve online teaching capabilities,
- **Reason:** Current system lacks built-in video tools; users rely on external apps.
- **Benefits:** Simplified virtual class management.
- **Urgency:** Medium, needed before next semester.

Change Control in Software Projects

2. Impact Analysis

- The project team evaluates:
 - **Scope impact:** Will it alter deliverables?
 - **Schedule impact:** Will it cause delays?
 - **Cost impact:** Will it increase the budget?
 - **Risk impact:** Does it introduce new risks?
 - **Resource impact:** Are additional skills/materials needed?
- A recommendation is made (approve, reject, or defer).

Change Control in Software Projects

3. Approval/Reject Decision

- The Change Control Board (CCB) or authorized decision-maker reviews the assessment and
 - **Approves**
 - **Rejects** or
 - **Requests modifications.**
- Approved changes move to implementation planning.
- Rejected changes are documented with reasons

Change Control in Software Projects

4. Implementation and Monitoring

- The project team updates plans (schedule, budget, resources).
- The change is executed following proper procedures.
- Stakeholders are informed of the change.
- The team monitors the change implementation to ensure it aligns with the plan.
- Progress is tracked using KPIs, status reports, meetings.

Change Control in Software Projects

5. Review, Document and Close Change

- After implementation, a **Post-Implementation Review (PIR)** checks:
 - Was the change successful?
 - Did it meet objectives?
 - Were there unexpected issues?
 - Are lessons learned documented?
- Adjustments are made if needed.

Change Control in Software Projects

5. Review, Document and Close Change

- All change records (requests, approvals, impact analyses, reviews) are archived.
- The **Change Log** is updated for future reference.
- The change is formally closed.

Change Control in Software Projects

Change Request Submission

Initiating the change process by submitting a request



Impact Analysis

Evaluating the potential effects of the change



Approval/Reject Decision

Deciding whether to proceed with the change



Implementation and Monitoring

Executing the change and tracking its progress



Review, Document and Close Change

Finalizing the change by reviewing and documenting

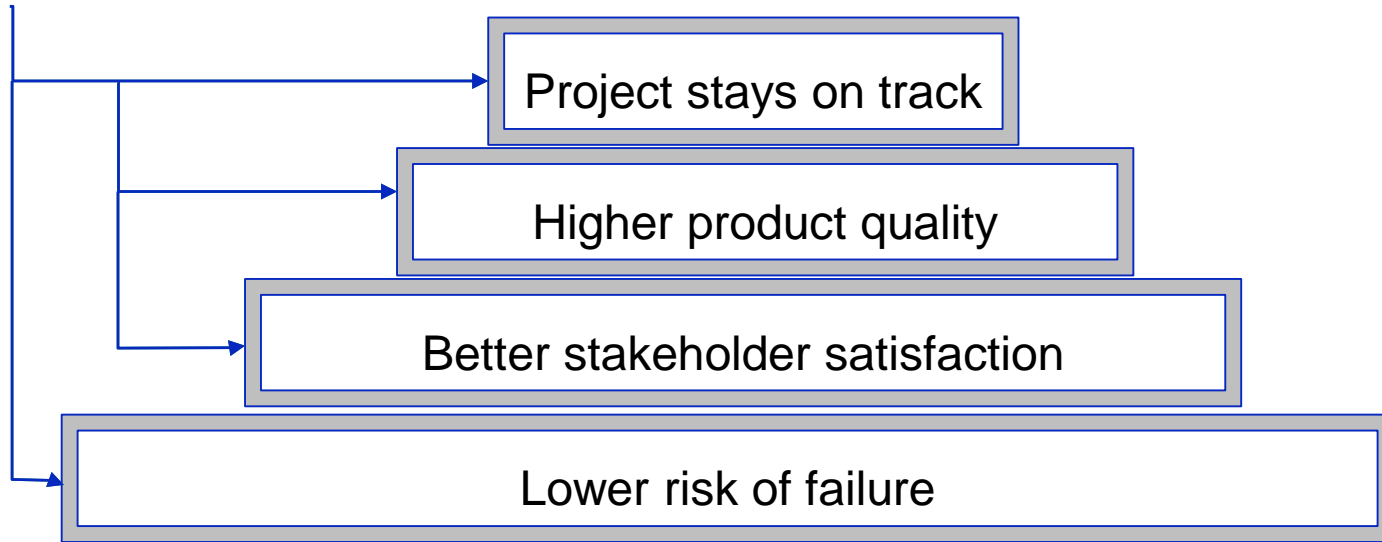


Change Control process Flow



Change Control in Software Projects

Advantages



Change Control in Software Projects

Best Practices:-

Use formal change
request process

Maintain detailed change
log

Evaluate impacts before
approval

Communicate with all
stakeholders

Overview of Software Configuration Management (SCM)

SCM is the process of identifying, controlling, and tracking software artifacts (code, design, documents)

SCM

- During the development phase, the **work products** get **modified** as development activities are carried out.
 - Maintenance phase → due to enhancement or adaption (bug fix)
- Thus, the state of the work products continually change.
 - The state of all work products **at any point of time** is called the **configuration of the software product** (Bob Hughes, 2011)

SCM

- Therefore, SCM deals with
 - Effectively tracking and controlling the configuration of a software product during its entire life cycle.

SCM

Why SCM?

- **Manual Change Control**
 - Suitable for small projects or minor updates
 - Changes handled by a designated person/team
 - Uses simple tools (**e.g., spreadsheets, emails**)
- However, as software systems grow in complexity, especially when multiple work products, components, or product variants are involved
 - manual methods become insufficient

SCM

- **Challenges**

- **Error-prone** in complex environments
- **Difficult to track** multiple product variants
- **Lacks consistency** and **traceability**
- **Risk** of **overwriting** or **losing** changes

SCM

Example: *what will happen and what's the role of SCM?*

- If you are managing a software project with multiple modules.
 - If one team **updates** the requirements while another modifies the design or code based on **old specs**,
 - **you'll have inconsistencies and rework.**
 - SCM prevents this by ensuring everyone is working from the same, controlled versions of all work products.

SCM

- **Purpose of using SCM**
 - Systematic control of change
 - Undoing change
 - System accounting
 - Handling variants
 - Accurate determination of project status
 - Prevent unauthorized access to work products

SCM

Configuration Identification

Determine items under control and assign identifiers



Configuration Control

Establish change control process and manage requests



Configuration Status Accounting

Record and report the status of configuration items

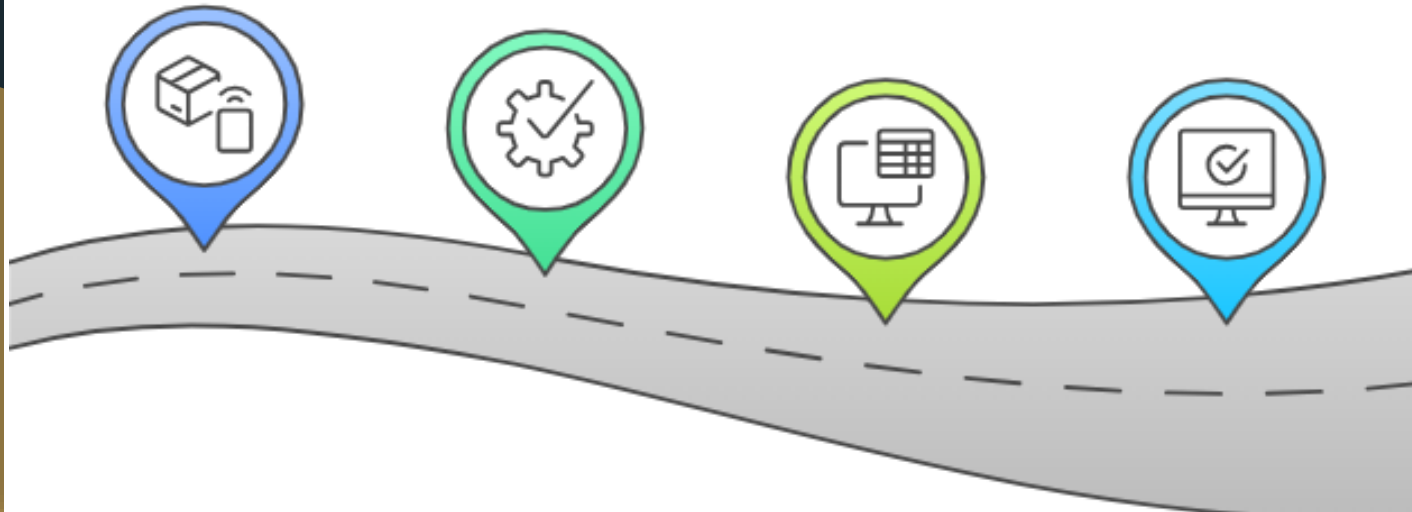


Configuration Verification and Audit

Ensure items meet requirements and conduct audits



Process of SCM



SCM - Tools

- **VCS**

- **Git**: Most widely used; supports distributed development, branching, and merging.

- **CI/CD**

- **Jenkins**

- **Configuration as Code / Infrastructure Management**

- **Ansible**

- **Code Review and Collaboration Tools**

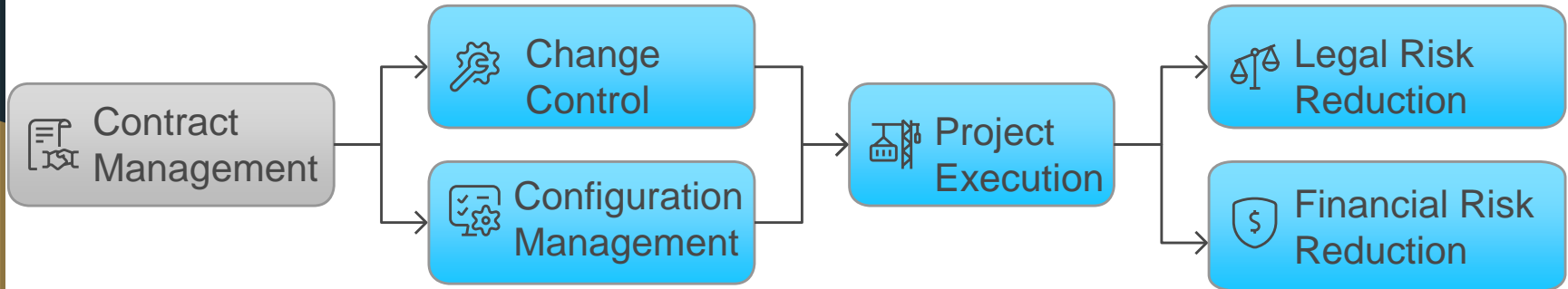
- **GitHub**

Contract Management

- According to (PMI, 2013), contract management is the process of **creating, executing, and overseeing** agreements between parties to ensure
 - **Deliverables**
 - **Timelines**
 - **Costs** and
 - **Quality standards are met.**

Contract Management

- When combined with **Change Control (CC)** and **Configuration Management (CM)**, Contract management ensures that contractual obligations align with **project execution, minimizing legal** and **financial risks**.



Contract Management

- **Types of contracts in software projects**
 - **Fixed-Price Contract**
 - Predetermined cost, scope changes require formal approvals (strong Change Control needed).
 - **Time & Materials (T&M)**
 - Pay-as-you-go, flexible but needs strict CM to track work.
 - **SLA-Based Contracts**
 - Includes penalties/rewards for meeting service levels (requires CM for compliance tracking).

Contract Management

- **Key contract components in software projects**
 - **Scope of Work (SOW):** Defines deliverables
 - **Payment Terms:** Tied to milestones (requires Change Control for scope adjustments).
 - **Change Request Clause:** Specifies how modifications are handled.
 - **Intellectual Property (IP) Rights:** Determines who owns code/configurations (CM tracks ownership)

Contract Management

Contract phases



Contract Management

How Contract management related with CC & CM?

Aspect	Contract Management	Change Control	Configuration Management	Relationship
Scope Changes	Contract defines change procedures	Evaluates & approves changes	Implements & versions changes	A contract change clause triggers CC, and CM updates baselines.
Compliance	Legal/financial obligations	Ensures changes follow rules	Tracks what was delivered	CM audits prove contractual compliance.

Contract Management

How Contract management related with CC & CM?

Aspect	Contract Management	Change Control	Configuration Management	Relationship
Deliverables	Lists expected outputs	Manages modifications to outputs	Controls versions of outputs	CM ensures deliverables match contract terms.
Disputes	Resolves breaches (e.g., missed deadlines)	Documents why changes occurred	Provides evidence (version history)	CM logs help settle disputes.

Contract Management

Why Integration of the three Matters?

- **Avoids cost overruns:** Unapproved changes (missing CC) can violate contract terms.
- **Ensures auditability:** CM provides proof of compliance.
- **Prevents legal risks:** Contracts define liability; CC+CM enforce accountability.
- **Supports agile contracts:** Flexible agreements still need CM for traceability.

Best Practices

✓ Embed Change Control in Contracts

- ✓ Define how modifications are handled.

✓ Use CM for Evidence

- ✓ Version history proves deliverables were met.

✓ Automate Tracking

- ✓ Tools like **Jira + DocuSign + Git** links contracts to execution.

✓ Regular Reviews

- ✓ Check if project aligns with contract terms.

Summary

- Change control and configuration management are integral components of successful software project management.
- By implementing robust processes and practices,
 - **Teams can navigate the complexities of software development while ensuring quality, stability, and compliance.**
- Emphasizing these concepts not only enhances project outcomes but also contributes to
 - **The long-term sustainability of software systems.**

Summary

- Contract management involves the creation, execution, and oversight of agreements between stakeholders in a software project to ensure mutual obligations are fulfilled,
 - particularly regarding deliverables, timelines, quality, and costs.
- Used to
 - Clarify roles, responsibilities, and expectations
 - Protect both client and vendor legally and financially
 - Ensure quality, risk management, and timely delivery

References

1. *PMI. (2013). A Guide to the Project Management Body of Knowledge (5th ed.). PMI, Inc.*
2. *Bob Hughes, M. C. (2011). Software Project Management (5th ed.). McGraw-Hill.*
3. *Asana, T. (2025, January 7). What is a change control process and how do you use it? Retrieved May 26, 2025, from Asana website: <https://asana.com/resources/change-control-process>*