# INTRODUCTION TO SOFTWARE PROJECT MANAGEMENT


## SOFTWARE ENGINEERING 2 | CS6300

JOHN PAUL M. MANUEL | ACLC COLLEGE OF TAYTAY

ACLC COLLEGE OF TAYTAY
(A Member of AMA Education System)

VERSION 01-250201-2425

# WHAT IS SOFTWARE PROJECT MANAGEMENT?

Software Project Management is the art and science of planning, organizing, and leading software projects. It is a specialized discipline within project management that focuses on ensuring the successful execution of software projects by implementing planning, monitoring, and control strategies.


SOFTWARE PROJECT MANAGEMENT

# PRINCIPLES OF MODERN SOFTWARE MANAGEMENT

- **Architecture-First Approach** – Prioritize system architecture to manage complexity.
- **Iterative Life-Cycle Process** – Identify and mitigate risks early.
- **Component-Based Development** – Emphasize reusability in design methods.
- **Change Management** – Establish a structured environment for modifications.
- **Round-Trip Engineering** – Use tools to allow seamless forward and reverse engineering.

SOFTWARE PROJECT MANAGEMENT

# PRINCIPLES OF MODERN SOFTWARE MANAGEMENT

- **Model-Based Notation –** Document designs rigorously.
- **Objective Quality Control –** Integrate measurable assessment techniques.
- **Demonstration-Based Approach –** Validate intermediate results through prototypes.
- **Incremental Releases –** Deliver functional features in stages.
- **Configurable Process –** Ensure scalability and adaptability.

SOFTWARE PROJECT MANAGEMENT

# ORGANIZATIONAL & TEAM STRUCTURE IN SOFTWARE DEVELOPMENT

## Organizational Structure

Software organizations assign teams to different projects based on structural models, each with advantages and disadvantages. The choice of structure impacts efficiency, communication, and project success.

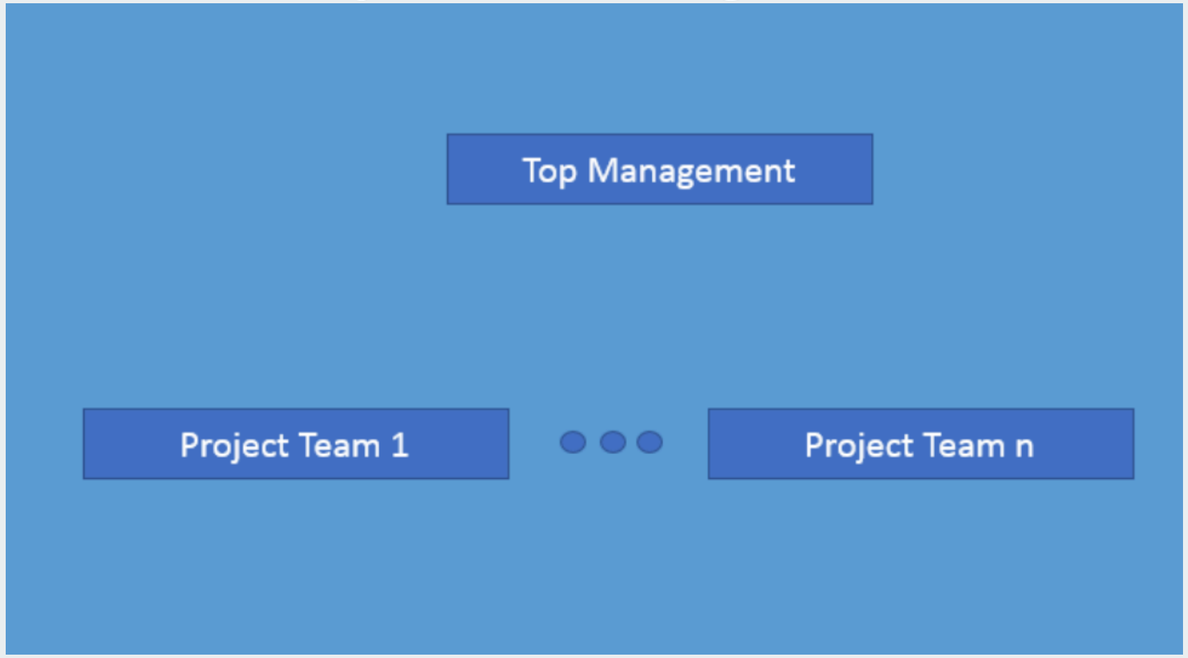
## Functional vs. Project-Based Structure

- **Functional Organization:** Teams are divided based on expertise (e.g., UI, backend, testing).
- **Project-Based Organization:** Teams are formed per project, fostering cross-functional collaboration.

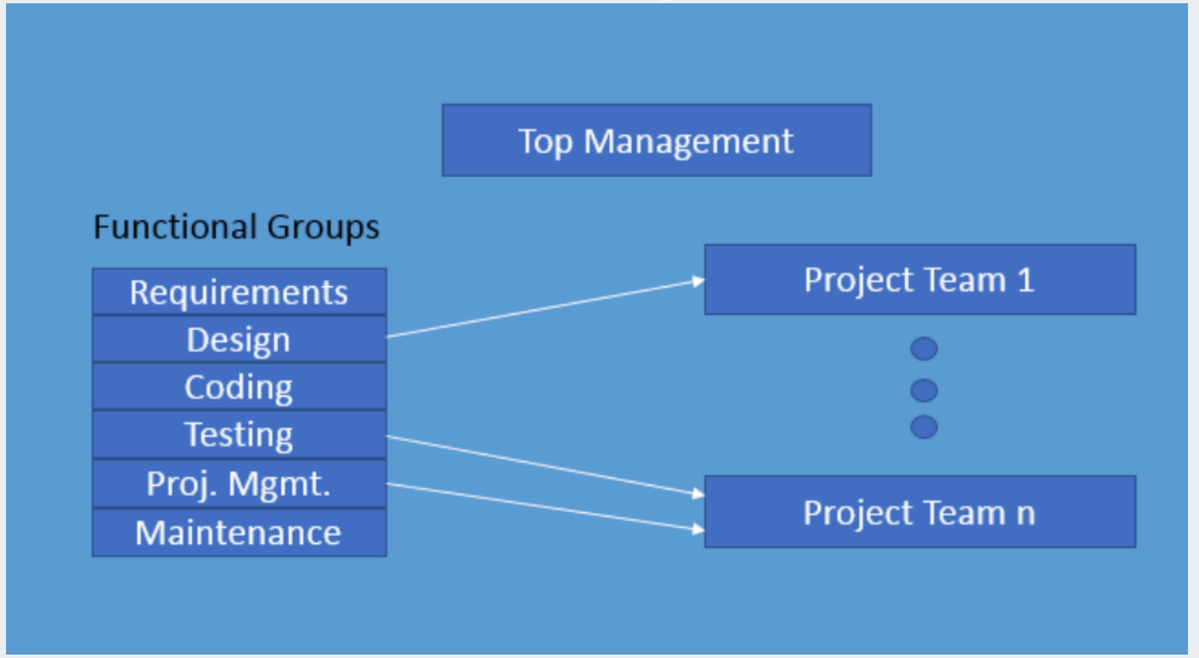
SOFTWARE PROJECT MANAGEMENT

# ORGANIZATIONAL & TEAM STRUCTURE IN SOFTWARE DEVELOPMENT

## Project-Based Organization




## Functional Organization



SOFTWARE PROJECT MANAGEMENT

# ADVANTAGES OF FUNCTIONAL ORGANIZATION

✓ **Specialization & Expertise** – Employees become highly skilled in their specific domain.

✓ **Efficient Resource Allocation** – Team members can be assigned across multiple projects efficiently.

✓ **Clear Career Growth** – Employees have a well-defined path within their function (e.g., UI specialists can advance within UI roles).

✓ **Knowledge Retention** – Expertise remains within the organization, even if projects change.

✓ **Standardized Processes** – Best practices are maintained within departments, improving consistency.

SOFTWARE PROJECT MANAGEMENT

# DISADVANTAGES OF FUNCTIONAL ORGANIZATION

❌ **Limited Cross-Functional Collaboration** – Lack of interaction between teams can cause communication gaps.

❌ **Slow Decision-Making** – Requires approval from multiple levels, leading to bureaucracy.

❌ **Less Project Focus** – Teams work on multiple projects simultaneously, reducing efficiency in specific projects.

❌ **Departmental Silos** – Teams may prioritize their functional goals over overall project success.

❌ **Lower Adaptability** – Changing requirements may be harder to implement due to rigid structures.

SOFTWARE PROJECT MANAGEMENT

# ADVANTAGES OF PROJECT-BASED ORGANIZATION

✓ **Stronger Project Focus** – Teams are fully dedicated to a single project, ensuring better efficiency.

✓ **Faster Decision-Making** – Decisions can be made within the project team, reducing bureaucratic delays.

✓ **Cross-Functional Collaboration** – Encourages teamwork and knowledge-sharing across different disciplines.

✓ **More Flexibility** – Teams can quickly adapt to project changes without being restricted by department structures.

✓ **Better Ownership & Accountability** – Team members feel more responsible for project success.

SOFTWARE PROJECT MANAGEMENT

# DISADVANTAGES OF PROJECT-BASED ORGANIZATION

❌ **Resource Duplication** – Since each project has its own team, expertise may not be efficiently shared across projects.

❌ **Higher Costs** – Maintaining separate teams for each project can increase operational expenses.

❌ **Knowledge Loss** – When a project ends, expertise and experience may not be retained effectively.

❌ **Career Uncertainty** – Employees may struggle with career progression as they move between projects.

❌ **Limited Long-Term Skill Development** – Employees may not develop deep expertise in one area since they work on multiple aspects of a project.

# SOFTWARE PROJECT TEAM STRUCTURES

- **Chief Programmer Team** – A strong leader (chief programmer) directs all development.
- **Democratic Team** – Equal participation and decision-making by all team members.
- **Mixed Team** – A blend of hierarchical and democratic approaches for flexibility.

SOFTWARE PROJECT MANAGEMENT

# KEY ATTRIBUTES OF A GOOD SOFTWARE ENGINEER

- ✓ Strong foundation in software engineering principles.
- ✓ Good domain knowledge and programming skills.
- ✓ Effective communication (oral, written, and interpersonal).
- ✓ High motivation and discipline.
- ✓ Ability to work in a team.
- ✓ Analytical thinking and problem-solving skills.

SOFTWARE PROJECT MANAGEMENT

# PROJECT PLANNING, MONITORING & CONTROL

## Project Planning

- The strategic process of defining goals, estimating resources, and setting timelines.
- Utilizes project management tools like Gantt charts for scheduling.

## Monitoring & Evaluation (M&E)

- Ensures project objectives are met through performance tracking.
- Identifies risks and deviations early for timely interventions.

## Software Quality Control

- Enforces quality standards throughout development.
- Ensures deliverables meet customer expectations and compliance requirements.

SOFTWARE PROJECT MANAGEMENT

# PROJECT LIFECYCLE: INITIATION & CLOSURE

**Project Initiation**

- The pre-planning phase defining scope, objectives, and stakeholders.
- Lays the foundation for execution and risk assessment.

**Project Closure**

- Formal completion of a project with deliverables handed over.
- Involves documentation finalization, stakeholder notification, and post-project evaluations.

SOFTWARE PROJECT MANAGEMENT

# TECHNICAL, QUALITY, & MANAGEMENT PLANS

**Technical Planning**

- Outlines design, development, and resource allocation strategies.
- Provides a roadmap for execution.
-

**Quality Planning**

- Establishes quality benchmarks for project deliverables.
- Defines verification and validation methodologies.

SOFTWARE PROJECT MANAGEMENT

# KEY COMPONENTS OF A QUALITY PLAN

- ✅ Management responsibilities
- ✅ Documentation control
- ✅ Requirements scope
- ✅ Design and development control
- ✅ Testing and quality assurance
- ✅ Risk mitigation strategies
- ✅ Quality audits and defect tracking
- ✅ Training needs assessment

SOFTWARE PROJECT MANAGEMENT

# PROJECT MANAGEMENT PLANNING PROCESS

- **Define Goals** – Set clear and measurable project objectives.
- **Identify Resources** – Determine the required manpower, tools, and budget.
- **Outline Key Tasks** – Break down the project into actionable steps.
- **Prioritize & Schedule** – Assign importance levels and create a timeline.
- **Allocate Responsibilities** – Assign tasks to team members based on skills.
- **Monitor & Evaluate** – Set up feedback mechanisms to track progress.
- **Adjust & Improve** – Identify alternative strategies to optimize outcomes.