Module-04

Chapter:-1 - Introduction to Project Management

Introduction

1. Comparison of Software and Other Projects:

- Evaluates whether software project management differs significantly from managing other types of projects.

2. Key Ideas in Software Project Management:

- Focuses on planning, monitoring, and controlling software projects.

3. Objective of All Projects:

- All projects, including software projects, aim to meet specific objectives and satisfy real needs.

4. Stakeholders and Objectives:

- Identifying the project's stakeholders and their objectives is crucial.
- Ensuring that these objectives are met is the primary aim of project management.

5. Assessing the Project's Progress:

- Knowing the current state of the project is essential to predict whether it will meet its objectives in the future.

Project and Importance of Project Management

Definition of a Project:

- A project is a planned activity, according to dictionary definitions.

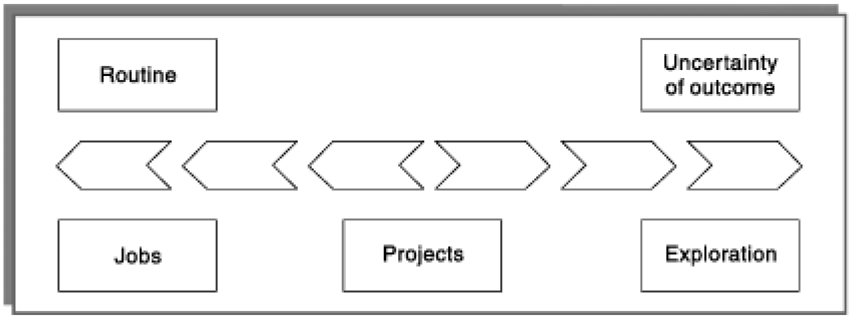

FIGURE 1.1 Activities most likely to benefit from project management

Planning Emphasis:

- Planning involves determining how to carry out a task before starting.
- Essential for careful consideration even with uncertain or exploratory projects, where plans should be seen as provisional.

Routine vs. Non-Routine Activities:

- Routine maintenance tasks are often well-known and documented, making detailed planning unnecessary.
- Documentation ensures consistency and aids newcomers.

Optimal Use of Conventional Project Management:

- Most beneficial for activities that fall between routine and highly uncertain exploratory projects.

Boundary between Routine and Non-Routine:

- There is a hazy boundary where the first instance of a routine task can resemble a project.
- Conversely, developing a system similar to previous ones can have significant routine elements.

Importance of Project Management:

- Technical students might resist studying project management as it diverts them from coding.

Financial Stakes:

- Significant amounts of money are involved in ICT projects.
- Example: In the UK during 2002-2003, the central government spent more on ICT projects (£2.3 billion) than on road contracts (£1.4 billion).
- The Department for Work and Pensions alone spent over £800 million on ICT.

Consequences of Mismanagement:

- Poor management of ICT projects can lead to reduced funding for essential services like hospitals.

Project Success Rates:

- A 2003 report by the Standish Group in the US analysed 13,522 projects:
  - Only a third were successful.
  - 82% were late.
  - 43% exceeded their budget.

Causes of Project Failures:

- Mismanagement is often the root cause.
- The National Audit Office in the UK highlighted a 'lack of skills and a proven approach to project management and risk management' as factors contributing to project failures.

The following characteristics distinguish projects:

- ❖ non-routine tasks are involved;
- ❖ planning is required;
- ❖ specific objectives are to be met or a specified product is to be created;
- ❖ the project has a predetermined time span;
- ❖ work is carried out for someone other than yourself;
- ❖ work involves several specialisms;
- ❖ people are formed into a temporary work group to carry out the task;
- ❖ work is carried out in several phases;
- ❖ the resources that are available for use on the project are constrained;
- ❖ The project is large or complex.

Project Size and Difficulty:

- Larger projects (e.g., 20 developers) are disproportionately more difficult than smaller ones (e.g., 10 developers) due to increased coordination needs.
- Examples and exercises in the book focus on smaller projects for ease of understanding, but the discussed techniques are relevant to larger projects as well.

Nature of Projects as Temporary Sub-organizations:

- Projects are temporary sub-organizations formed to carry out specific tasks.
- This setup can disrupt existing organizational authority but allows specialists to focus on a single important task.

Advantages and Disadvantages:

- **Advantage**: A specialized group can concentrate on a specific task.
- **Disadvantage**: Projects can be seen as disruptive by others within the organization.

- **Disadvantage**: Expertise gained during the project may be lost when the team disbands at the project's end.

Software Projects versus Other Types of Project

General Project Management Techniques:

- Many general project management techniques are applicable to software project management.
- Fred Brooks identified unique characteristics of software projects that add difficulty.

Characteristics of Software Projects:

- **Invisibility**:
  - Unlike physical artifacts (e.g., bridges), software progress is not immediately visible.
  - Software project management involves making the invisible progress visible.
- **Complexity**:
  - Software products are more complex per dollar/pound/euro spent compared to other engineered artifacts.
- **Conformity**:
  - Traditional engineers work with consistent physical laws (e.g., cement and steel).
  - Software developers must conform to the inconsistent requirements of human clients and organizations.
  - Organizations can exhibit "organizational stupidity" due to lapses in memory, communication, or decision-making.
- **Flexibility**:
  - Software's ease of change is a strength but also a challenge.

- Software is expected to adapt to changes in physical or organizational systems, making it particularly subject to change.

Contract Management

In-House Projects:

- Users and developers work for the same organization.

Outsourced Projects:

- Organizations increasingly contract out ICT development to external developers.
- The client organization appoints a project manager to supervise the contract.
- This project manager delegates many technical decisions to the contractors and focuses on budget and timeline.

Roles and Responsibilities:

- Client-side project managers prioritize overall project budget and schedule.
- Supplier-side project managers handle technical issues.

Focus of the Book:

- The book primarily addresses the concerns of technical project managers on the supplier side.

Activities Covered by Software Project Management

1. **Scope of Software Projects**:
    - Software projects involve more than just writing software.
    - Even when using "off the shelf" software, the project remains a software project due to the presence of other software-related activities.
2. **Successive Processes**:
    - There are typically three successive processes that bring a new system into being (as depicted in Figure 1.2).

Would you like more details on what those three processes typically involve, or any specific details about Figure 1.2 if you have it?

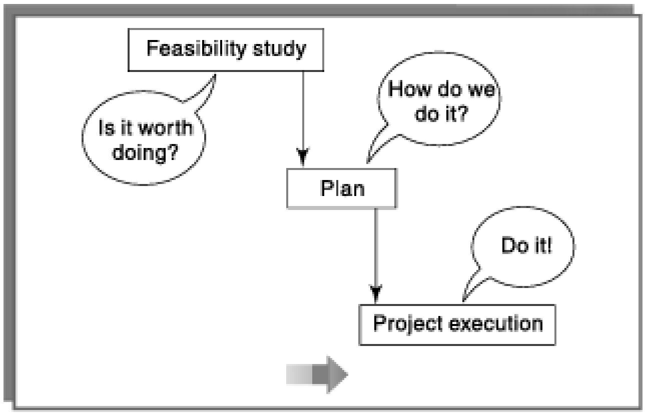

FIGURE 1.2 The feasibility study/plan/execution cycle

The feasibility study and subsequent phases of project development are critical steps in determining whether a project is worth pursuing and ensuring its successful execution. Here's a structured overview based on the information provided:

Feasibility Study

- **Purpose**: To assess whether a project has a valid business case and is worth starting.
- **Activities**:
  - Gather information about the requirements of the proposed application.
  - Identify the aims of the stakeholders and determine the means to achieve them.
  - Estimate developmental and operational costs.
  - Evaluate the value of the benefits of the new system.
  - For large systems, the feasibility study itself could be a separate project with its own plan.
  - It could also be part of a strategic planning exercise examining a range of potential software developments.
  - Assess a program of development that includes multiple projects.

Planning

- **Initiation**: Begins if the feasibility study indicates that the project is viable.
- **Approach**:
    - Create an outline plan for the entire project.
    - Develop a detailed plan for the first stage.
    - Planning for later stages is postponed until more detailed and accurate information is available after the earlier stages are completed.

Project Execution

- **Execution Phase**: Involves design and implementation sub-phases.
- **Challenges**:
    - New project planners often find the boundary between design and planning to be hazy.
    - Design involves making decisions about the form of the products to be created, such as the user interface and internal architecture.
    - The plan details the activities to create these products, which can be influenced by design decisions.
    - Detailed planning and design are interconnected, as design decisions can determine planning activities.

Sequence of Software Development Activities (ISO 12207)

- **Standard**: ISO 12207 provides a recommended sequence of software development activities.
- **Scope**:
    - Some activities are concerned with the system as a whole.
    - Others are specific to software development.
    - Software development may be only one part of a broader project, which could also include:

- Installation of ICT infrastructure.
- Design of user jobs.
- User training.

Understanding and properly executing these phases are essential for the successful development and implementation of a project, ensuring that it meets stakeholder needs and achieves its intended benefits.

Requirements Analysis

Start:

- Begins with requirements elicitation or requirements gathering.

Purpose:

- Establish what potential users and their managers require of the new system.

Types of Requirements:

1. **Functional Requirements:**
    - What the system should do.
    - Example: Dispatching an ambulance in response to an emergency call.
2. **Quality Requirements:**
    - How well the functions must work.
    - Example: Transaction time affected by hardware, software performance, and human operation speed.

System Requirements:

- Training operators to use the system efficiently.
- Resource requirements related to application development costs.

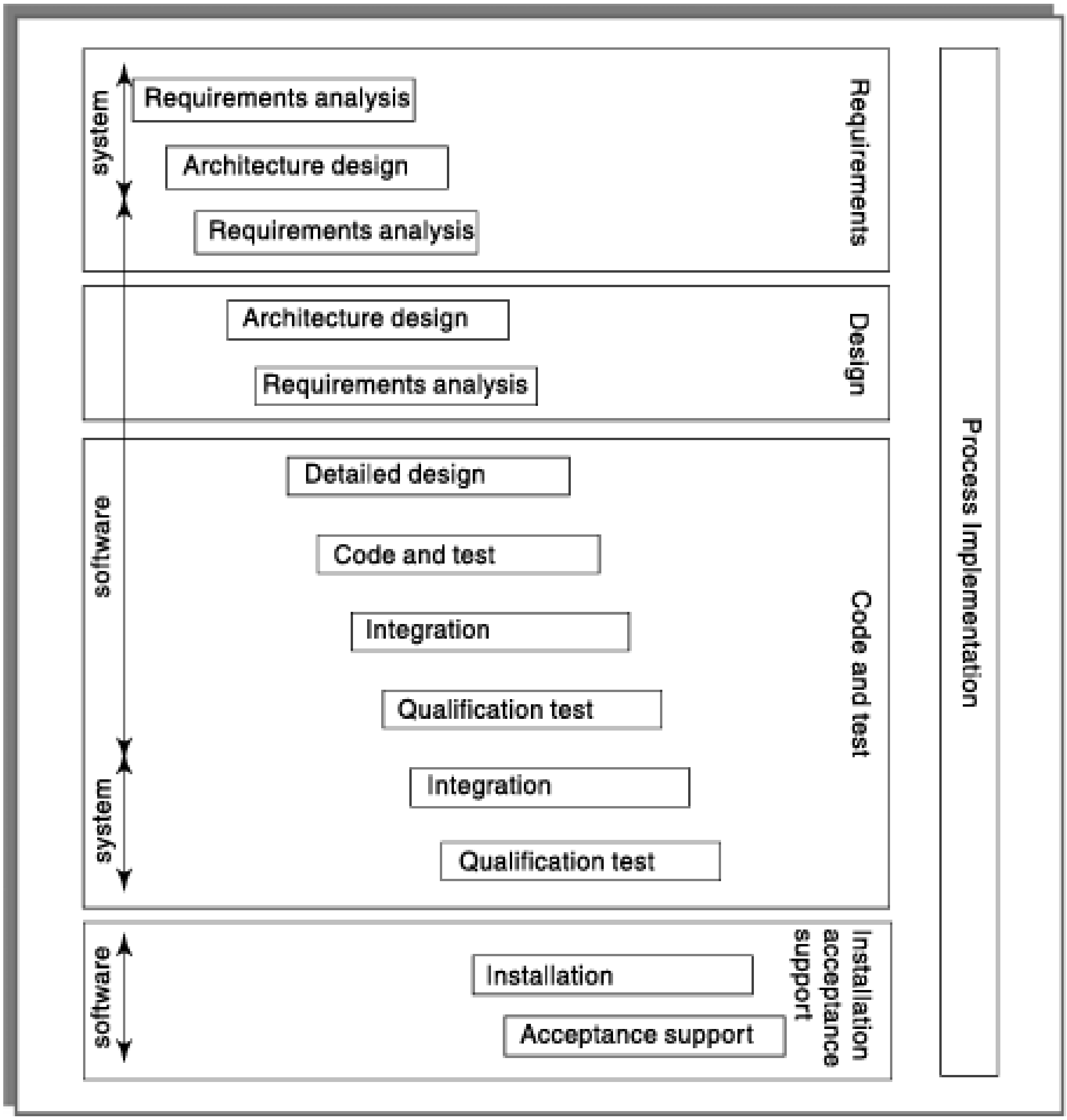

FIGURE 1.3 The ISO 12207 software development life cycle

Architecture Design

Purpose:

- Identify components of the new system that fulfill each requirement.

Components:

- Could be existing or new.
- May include software, hardware, or work processes.

Considerations:

- Integration with existing legacy systems.
- System architecture design influences software requirements.
- A second architecture design process maps software requirements to software components.

Detailed Design

Purpose:

- Each software component consists of multiple software units.

Process:

- Detailed design of these units.
- Code and test each unit separately.
- Initial testing to debug individual software units.

Integration

Purpose:

- Test components together to meet overall requirements.

Process:

- Combine different software components.
- Test software with hardware platforms and user interactions.

Qualification Testing

Purpose:

- Ensure all requirements have been fulfilled.

Process:

- Thorough testing of the entire system, including software components.

Installation

Purpose:

- Make the new system operational.

Activities:

- Set up standing data (e.g., employee details in a payroll system).
- Configure system parameters.
- Install software onto hardware platforms.
- Provide user training.

Acceptance Support

Purpose:

- Resolve problems with the newly installed system.

Activities:

- Correct errors.
- Implement agreed extensions and improvements.
- Software maintenance as a series of minor software projects.

- Most software development in many environments is maintenance.

Plans, Methods and Methodologies

Planning an Activity

Foundation:

- A plan for an activity is based on a method of work.

Example - Testing Software:

1. **Analyze the requirements** for the software.
2. **Devise and write test cases** to check that each requirement is satisfied.
3. **Create test scripts and expected results** for each test case.
4. **Compare the actual results with the expected results** and identify discrepancies.

Method vs. Plan:

- **Method:**
  - Relates to a type of activity in general.
  - Provides a systematic approach, e.g., steps to test software.
- **Plan:**
  - Converts methods into real activities.
  - Identifies details for each activity:
    - **Start and end dates.**
    - **Responsible personnel.**
    - **Tools and materials** needed, including information.

Interrelationship:

- The output from one method might be the input to another.
- Methods and techniques are often grouped into **methodologies** (e.g., object-oriented design).

Some Ways of Categorizing Software Projects

Purpose:

- Identify project characteristics that affect planning and management.

Categories

1. **Compulsory vs. Voluntary Users**
   - **Compulsory Users:**
     - Users must use the system to perform tasks (e.g., recording a sale).
     - Easier to elicit precise requirements from users.
   - **Voluntary Users:**
     - Users choose to use the system (e.g., computer games).
     - Difficult to elicit precise requirements.
     - Relies on developers' ingenuity, market surveys, focus groups, and prototype evaluation.
2. **Information Systems vs. Embedded Systems**
   - **Information Systems:**
     - Enable staff to carry out office processes.
     - Example: Stock control system.
   - **Embedded Systems:**
     - Control machines.
     - Example: Air conditioning control in a building.
   - **Hybrid Systems:**

- Combine elements of both information and embedded systems.
- Example: Stock control system that also controls an automated warehouse.

Outsourced Projects

1. **Commercial Sense of Outsourcing**: Companies may outsource parts of a project if they lack expertise or find it cost-effective.
2. **Project Characteristics**: Outsourced projects are typically small and need to be completed within a few months.
3. **Management Challenges**: Managing outsourced projects entails special challenges due to their size and time constraints.
4. **Indian Software Companies**: Indian companies are renowned for executing outsourced software projects and are beginning to focus on product development.
5. **Revenue Impact**: Generic software products provide long-term revenue, while outsourced projects offer one-time revenue.

Objective-Driven Development

1. **Project Distinctions**: Projects can aim to produce a product or meet certain objectives.
2. **Client Responsibility**: In product creation, the client justifies the product.
3. **Objective-Driven Projects**: These projects identify solutions to problems, leading to product recommendations.
4. **Two-Stage Projects**: Initial objective-driven stage leads to recommendations, followed by product creation if needed.
5. **Technical Work by External Group**: Useful when user needs are unclear, allowing for a preliminary design and subsequent implementation based on agreed requirements.

Stakeholders

Definition: People who have a stake or interest in the project.

Early Identification: Important for setting up adequate communication channels.

Categories of Stakeholders:

- **Internal to the Project Team**: Under direct managerial control of the project leader.
- **External to the Project Team but within the Same Organization**: For example, users assisting with system testing; requires negotiated commitment.
- **External to Both the Project Team and the Organization**: Includes customers or users benefiting from the system and contractors working on the project; relationships based on contracts.

Diverse Objectives: Different stakeholders have different objectives that need to be recognized and reconciled by the project leader (e.g., ease of use for end-users vs. staff savings for managers).

Theory W: Proposed by Boehm and Ross, where the project manager aims to create win-win situations for all parties involved.

Overlooked Stakeholders: Important stakeholder groups can sometimes be missed, especially in unfamiliar business contexts.

Communication Plan: Recommended practice to create a communication plan at the start of a project to coordinate stakeholder efforts effectively.

Setting Objectives

1. **Project Owners**: Stakeholders who own the project control its financing and set the objectives.
2. **Objectives Definition**: Objectives define what the project team must achieve for project success and identify shared intentions among stakeholders.
3. **Outcome Focused**: Objectives focus on desired outcomes (post-conditions) rather than specific tasks.
   - Example: "The project will be a success if customers can order our products online" vs. "to build an e-commerce website".
4. **Multiple Routes to Success**: There are often several ways to meet an objective, which is advantageous.
5. **Project Authority**: When multiple stakeholders claim project ownership, a project steering committee (or similar body) with overall authority is needed to set, monitor, and modify objectives.
6. **Project Manager's Role**: The project manager runs the project daily and reports regularly to the steering committee.

Sub-Objectives and Goals

1. **Effective Objectives**: Must be within the control of the individual or team responsible.
   - Example: A business objective might be to reduce staff costs, but a software developer's sub-objective would be to keep development costs within a budget.
2. **Steps to Achieving Objectives**: Sub-objectives or goals are steps toward achieving the main objective.
   - Informal statement example: "To reach objective X, the following must be in place..."

SMART Objectives

1. **Specific**: Objectives should be concrete and well-defined.
   - Example: Instead of "to improve customer relations", use "to reduce customer complaints".
2. **Measurable**: There should be measures of effectiveness to determine success.
   - Example: "Did we install the new software by June 1?" (Yes/No question)
3. **Achievable**: The objective must be within the power of the individual or group to achieve.
4. **Relevant**: The objective must be relevant to the true purpose of the project.
5. **Time Constrained**: There should be a defined point in time by which the objective should be achieved.

Measures of Effectiveness

1. **Purpose**: Provide practical methods to check if an objective has been met.
2. **Example**: 'Mean time between failures' (MTBF) is used to measure reliability.
   - This is a performance measurement that can only be taken once the system is operational.
3. **Predictive Measures**: Used during the construction of the system to estimate future performance.
   - Example: A large number of errors found during code inspections might indicate potential reliability problems later.

Business Case

Justification or Business Case

1. **Purpose**: Ensure that the effort and expense of the project are worthwhile in terms of eventual benefits.
2. **Cost-Benefit Analysis**: Part of the project's feasibility study that itemizes and quantifies costs and benefits.

3. **Impact of Completion Date**: Sooner completion means sooner benefits.
4. **Business Model**: Explains how the new application can generate the claimed benefits.
   - **Example**: A new web-based application allowing global customers to order products online, increasing sales and profits.

Ensuring the Business Case

1. **Control Development Costs**: Ensure costs do not rise to a level that exceeds the value of benefits.
2. **Maintain System Features**: Do not reduce features to a level where expected benefits cannot be realized.
3. **Adhere to Delivery Dates**: Avoid delays that result in an unacceptable loss of benefits.

Project Success and Failure

1. **Ensuring Project Success**: The project plan should aim to preserve the business case for the project.
2. **Problems in Projects**: Every non-trivial project will have problems, and stakeholders may have different views on success and failure.

Distinguishing Objectives

1. **Project Objectives**: Targets the project team is expected to achieve, typically including:
   - Delivering the agreed functionality
   - Meeting the required level of quality
   - Completing on time
   - Staying within budget
2. **Business Objectives**: Ensure that the project's deliverables meet the business case (i.e., the value of benefits exceeds the costs).

Project and Business Success

1. **Project Success**: A project can be deemed successful if it meets its project objectives.
2. **Business Failure**: Despite meeting project objectives, the application might not meet the business case (e.g., a product doesn't sell, or a website isn't used).
3. **Business Success**: A project could be late and over budget, yet still be a business success if the deliverables generate long-term benefits that outweigh the costs.

Reducing the Gap between Project and Business Concerns

1. **Broader View**: Incorporate business issues into project management (e.g., market surveys, competitor analysis, focus groups, prototyping, and evaluation by potential users).
2. **Sequence of Projects**: Recognize that projects are often part of a sequence, where technical skills and expertise developed in earlier projects benefit later ones.

Long-Term Benefits

1. **Technical Learning**: Increases costs on early projects but provides long-term benefits through quicker, cheaper, and more accurate deployment of learned technologies.
2. **Reusable Assets**: Additional software assets, such as reusable code, provide long-term value.
3. **Customer Relationships**: Building trust with clients over multiple projects can lead to repeat business, which is more cost-effective than acquiring new clients.

Outsourcing Considerations

1. **Immediate Savings vs. Long-Term Benefits**: Outsourcing can provide immediate savings, but may result in the loss of long-term benefits of increased expertise and customer relationships.

2. **Astute Management**: Managers should assess areas of technical expertise beneficial to develop in-house versus outsourcing.

Management and Management Control

What is Management?

Management in the context of software project management involves several key activities:

1. **Planning**: Deciding what is to be done.
2. **Organizing**: Making arrangements.
3. **Staffing**: Selecting the right people for the job.
4. **Directing**: Giving instructions.
5. **Monitoring**: Checking on progress.
6. **Controlling**: Taking action to remedy hold-ups.
7. **Innovating**: Coming up with new solutions.
8. **Representing**: Liaising with clients, users, developers, suppliers, and other stakeholders.

Focus of Project Managers

Project managers spend most of their time on three primary activities:

1. **Project Planning**: Creating an initial plan, estimating project attributes (cost, duration, effort), and planning subsequent activities.
2. **Monitoring**: Continuously tracking the progress of the project.
3. **Control**: Ensuring that the project proceeds as planned and making necessary adjustments to address any deviations.

Project Management Stages

Project management is carried out over three well-defined stages, regardless of the methodology used:

1. **Project Initiation**:
    - **Initial Planning**: Undertaken immediately after the feasibility study and before starting requirements analysis and specification.
    - **Estimations**: Estimating cost, duration, and effort.
    - **Scheduling**: Developing schedules for manpower and resources based on estimations.
    - **Staffing**: Organizing staff and making staffing plans.
    - **Risk Management**: Identifying, analyzing, and planning to mitigate risks.
    - **Miscellaneous Plans**: Creating quality assurance plans, configuration management plans, etc.
2. **Project Execution**:
    - **Monitoring**: Tracking the progress of the project.
    - **Control**: Taking corrective actions to ensure the project stays on track.
3. **Project Closing**:
    - Completing all activities logically.
    - Formally closing all contracts.

Iterative Planning and Adjustment

- As the project progresses, more data becomes available, leading to a better understanding of project complexities and risks.
- Project parameters are re-estimated periodically, and plans are adjusted accordingly to ensure more accurate future planning.
- This iterative process involves continuous feedback between monitoring, control, and plan revision activities.

Importance of Initial Project Planning

- Initial project planning is crucial and occurs right after the feasibility study phase.
- Involves estimating project attributes such as cost, duration, and effort, which influence subsequent project activities.
- Regular revision of initial plans as the project progresses ensures alignment with the project's evolving requirements and constraints.

By focusing on planning, monitoring, and control, project managers aim to guide projects to successful completion while adapting to changes and resolving issues as they arise.

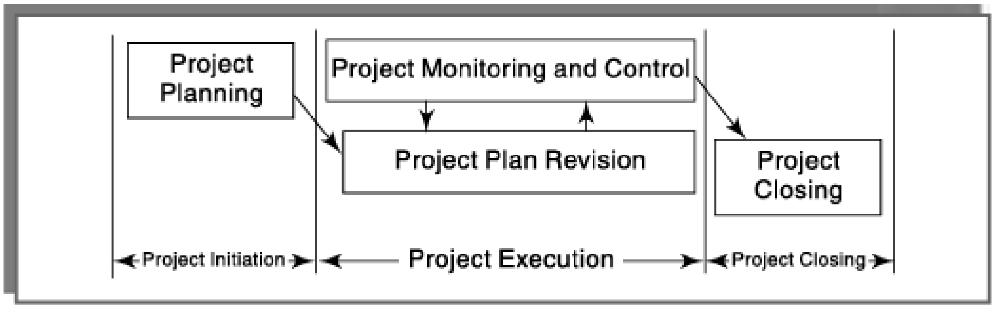

FIGURE 1.4 Principal project management processes

Project Planning Responsibilities of the Project Manager

Project planning is a critical responsibility of the project manager, involving several well-defined activities.

These activities include estimation, scheduling, staffing, risk management, and creating miscellaneous plans.

Estimation

The project manager estimates the following project attributes:

1. **Cost**: Determining the total cost to complete the project.

2. **Duration**: Estimating the time required to complete the project.
3. **Effort**: Assessing the amount of effort necessary to complete the project.

The accuracy of these estimations significantly impacts the effectiveness of subsequent activities such as scheduling and staffing.

Scheduling

- Based on the estimations of effort and duration, schedules for manpower and other resources are developed.

Staffing

- Organizing the project team and creating staffing plans.

Risk Management

- Includes risk identification, analysis, and planning to mitigate risks.

Miscellaneous Plans

- Creating additional plans such as quality assurance plans and configuration management plans.

Project Monitoring and Control

After the initiation of development activities, project monitoring and control activities are undertaken to ensure the project proceeds as planned. These activities involve:

- **Monitoring**: Tracking the progress of the project.
- **Control**: Taking corrective actions to address deviations from the plan.

Iterative Planning and Adjustment

- As the project progresses, the project manager gains more detailed knowledge about the project, allowing for more accurate planning.
- The project parameters are periodically re-estimated, incorporating new information and changes in project conditions.
- This iterative process involves continuous feedback between monitoring, control, and plan revision activities, allowing the project manager to adjust subsequent activities with increasing confidence.

Methodologies and Best Practices

- The Step Wise method, based on the PRINCE2 (Projects IN Controlled Environments) method, is extensively used in the UK and Europe.
- In the USA, the Project Management Institute's PMBOK (Project Management Body of Knowledge) offers similar best practices for software project management.

By focusing on these activities and following established methodologies, project managers can effectively plan, monitor, and control software development projects, ensuring that they meet their objectives within the constraints of cost, time, and effort.

Management Control

Management control involves setting objectives for a system and monitoring its performance. This process transforms raw data into useful information to guide decision-making. Here's a breakdown:

1. **Setting Objectives**:
   - Establish clear goals for the system or project.
2. **Monitoring Performance**:

   - Continuously track the performance of the system against the set objectives.
   - Use local managers for data collection to ensure accurate and up-to-date information.
3. **Data Collection**:
   - Collect raw data from various sources within the system.
   - Example: "Location X has processed 2000 documents."
4. **Data Processing**:
   - Transform raw data into meaningful information.
   - This involves calculating metrics that are useful for higher management decision-making.
5. **Transforming Data into Information**:
   - Calculate percentages: e.g., "Percentage of records processed."
   - Determine averages: e.g., "Average documents processed per day per person."
   - Estimate timelines: e.g., "Estimated completion date."

By following these steps, management can effectively control the system, ensuring it meets its objectives and identifying areas that require attention or improvement.

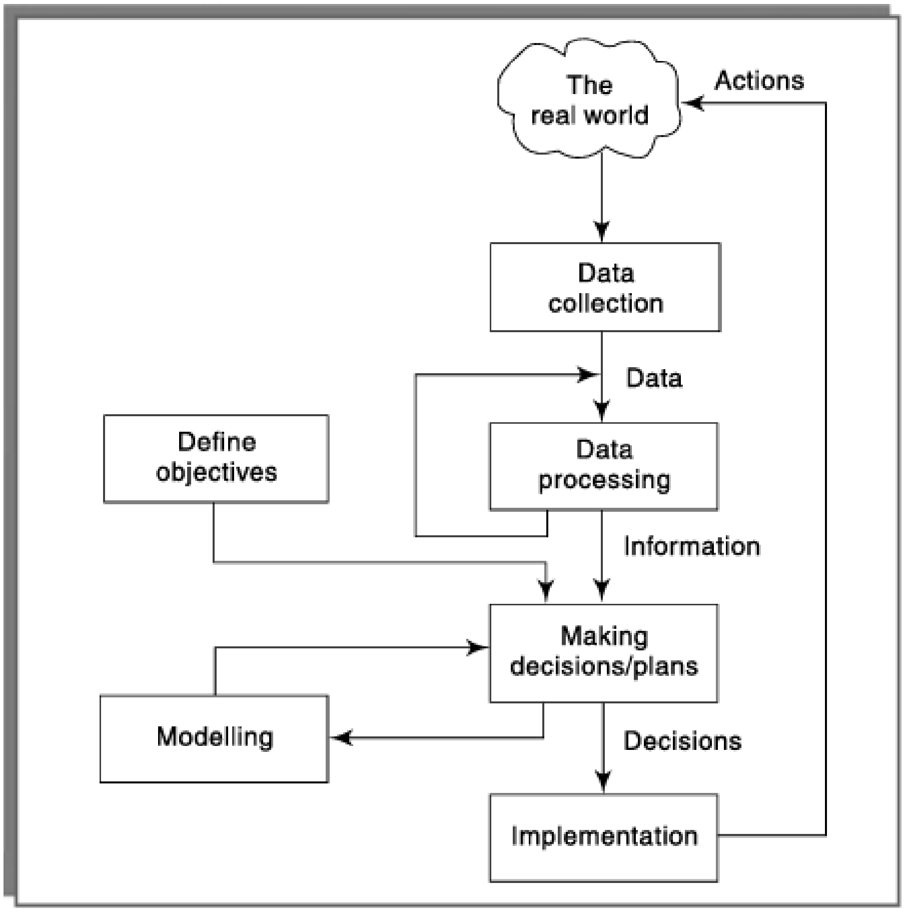

FIGURE 1.5 The project control cycle

Dynamic Project Management

In project management, especially in complex endeavors, continual monitoring and adjustment are crucial for success. Here’s how it works:

1. **Monitoring Performance Against Objectives**:
   - Project managers compare actual performance metrics, like the estimated completion date for data transfer in different branches, against established targets.
2. **Identifying Issues and Making Decisions**:
   - If some branches are falling behind, decisions need to be made. For instance, reallocating staff from one branch to another to meet deadlines.

- Managers must assess the impact of decisions carefully. While one branch might meet its deadline early, transferring staff could delay another branch.

3. **Modeling Consequences**:
   - Various solutions are modeled to predict their impact before implementation. This ensures informed decision-making and minimizes risks.
4. **Implementing Solutions**:
   - Once a decision is made, it's crucial to implement it effectively.
5. **Continuous Review and Adjustment**:
   - Progress is continually monitored and data processed to assess the effectiveness of implemented solutions.
   - Adjustments are made as new information becomes available or circumstances change.
6. **Dynamic Nature of Project Planning**:
   - Project plans are not static documents but evolve throughout the project lifecycle.
   - While extensive planning is essential, successful project execution depends on adaptive management and intelligent decision-making.

By maintaining flexibility and responsiveness, project managers can navigate challenges effectively and enhance the likelihood of project success despite unforeseen obstacles.

Traditional versus Modern Project Management Practices

The evolution from traditional to modern project management practices in software development reflects significant shifts in approach and focus.

Traditional Project Management Practices:

1. **Long-Term Planning**:
   - Projects were planned in detail upfront, often with extensive documentation and fixed requirements.

   - Emphasis was on creating a comprehensive project plan before execution started.
   - Monitoring and control focused on adhering strictly to the initial plan.
2. **Predictability and Stability**:
   - Projects were simpler and more predictable due to stable requirements and longer development cycles.
   - Changes to requirements post-sign-off were discouraged, aiming for stability throughout the project lifecycle.
3. **Customer Interaction**:
   - Customer involvement was limited mainly to requirement gathering and sign-off phases.
   - Customer feedback during development was minimal, with changes considered disruptive.
4. **Quality Management**:
   - Quality management was primarily about meeting predefined specifications and standards.
   - Assessing project progress and tracking quality were important but often measured against predefined benchmarks.

Modern Project Management Practices:

1. **Adaptive Short-Term Planning**:
   - Projects are planned in shorter cycles, focusing on incremental deliveries of functionalities.
   - Agile methodologies and extreme project management are adopted to accommodate changing requirements and customer feedback.
2. **Flexibility and Agility**:
   - Rapid application development and deployment strategies are prioritized to meet shorter project timelines.

- Projects are more flexible, allowing for continuous adaptation and adjustment based on evolving requirements and feedback.

3. **Customer Engagement and Feedback**:
   - Customers are actively involved throughout the development process, with their feedback encouraged and incorporated iteratively.
   - Incremental delivery models facilitate regular customer feedback loops, enabling continuous improvement and alignment with customer needs.
4. **Enhanced Quality Management**:
   - Quality management is integrated throughout the development process, focusing on iterative improvement and customer satisfaction.
   - Project managers play a crucial role in assessing project progress dynamically and ensuring quality in all intermediate artifacts.
5. **Change Management (Configuration Management)**:
   - Change management is integral to handling continuous customer feedback and evolving requirements.
   - Version control and baseline management become essential to manage multiple iterations and versions of the product effectively.