Chapter 10
Project Management

Jason C. H. Chen, Ph.D.
Professor of MIS
School of Business Administration
Gonzaga University
Spokane, WA 99258
chen@jepson.gonzaga.edu

INTRODUCTION

- This chapter provides an overview of what a project is and how to manage one
- It discusses the aspects of IT-intensive projects that make them uniquely challenging
- Finally, it identifies the issues that shape the role of the general manager in such projects and help them to manage risk

Learning Objectives

- List the elements of a good project.
- Understand why many IT projects fail to meet their targeted goals.
- Explain the relationship between time, scope, and cost of a project.
- Explain why Gantt charts are popular for planning schedules.
- Define RAD and explain how it compares to the SDLC.
- Be able to identify when it is time to pull the plug on a project.

Real World Example

- Rural Payments Agency (RPA), UK, blamed poor planning and lack of system testing for delays in paying out 1.5billion pounds of EU subsidies.
  - Only 15% were paid out by the end of 2006.
- The RPA had to make substantial changes to the system post implementation.
  - Testing did not take into account the real environment, leading to unanticipated work to populate the database in the first place.
- The system had not been properly managed.
  - Costs were at 122 million pounds, and were originally estimated at 46.5 million.

WHAT DEFINES A PROJECT

What Defines a Project?

- Organizations combine two types of work—**projects** and **operations** (Figure 10.1).
- Both types are performed by people and require a flow of limited resources.
- Both are planned, executed, and controlled.
- Figure 10.1 compares characteristics of both project and operational work.
- *"[A] project is a temporary endeavor undertaken to create a unique product or service. Temporary means that every project has a definite beginning and a definite end. Unique means that the product or service is different in some distinguishing way from all similar products or services."*
  - -Project Management Institute (1996)

Projects

- Companies use *projects* and *operations* to generate revenue.
- *Projects* are temporary endeavors that have a fixed start and stop date and time.
- *Operations* are ongoing, repetitive tasks that are performed until they are changed or replaced.
- Project managers may break projects into sub-projects depending upon the work.
- Figure 11.1 show the differences between operational and project based work.

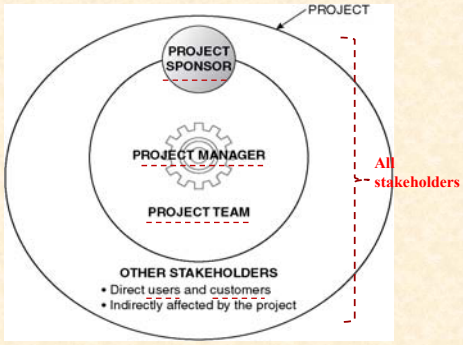
Figure 10.1 Characteristics of operational and project work.

Characteristics	Operations	Projects
Labor skills	Low	High
Training time	Low	High
Worker autonomy	Low	High
Compensation system	Hourly or weekly wage	Lump sum for project
Material input requirements	High certainty	Uncertain
Suppler ties	Longer duration	Shorter duration
	More formal	Less formal
Raw Materials inventory	Large	Small
Scheduling complexity	Lower	Higher
Quality control	Formal	Informal
Information flows	Less important	Very important
Worker-management communication	Less important	Very important
Duration	Ongoing	Temporary
Product or service	Repetitive	Unique

Project Stakeholders

- All projects have **stakeholders**.
  - Project **stakeholders** are the individuals and organizations that are either involved in the project, or whose interests may be affected as a result of the project.
    - ✓Include the project manager, project team, and the project sponsor (a general manager who provides the resources).
- The customer is an important stakeholder group.
  - Individuals or organizations who use the project's product.
  - Multiple layers of customers may be involved.
- The relationships among the project stakeholders are displayed in Figure 10.2.

Figure 10.2 Relationships among project stakeholders.

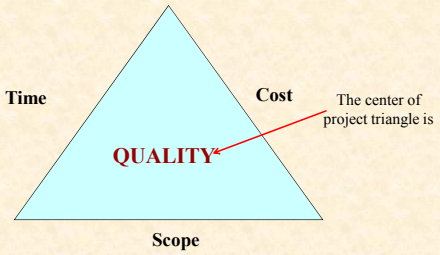


Organizing the Project

- A project **manager** can divide the project into **subprojects**.
- Subprojects can be based on distinct activities (e.g., quality control testing).
- This organizing method enables the project manager to contract certain kinds of work externally.
  - Provides a framework for managing crucial project resources, competing resource requirements, and shifting priorities among a set of projects.

- **What are the three elements in the "*Project Triangle*"?**
- **What is the center in the triangle?**

Figure 10.3 Project Triangle (Project Management Trade-offs)



The objective of the PM is to define project's scope realistically and ultimately deliver quality of product/service ***on time, on budget and within scope.***

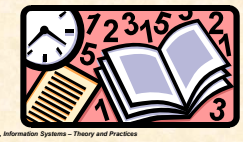
What is Project Management?

- Project management:
  - Applying knowledge, skills, tools, and techniques to project activities in order to meet or exceed stakeholder needs and expectations.
  - Involves continual trade-offs managed by the project **manager**.
- **Trade-offs** can be subsumed in the *project triangle* (Fig. 10.3).
  - **1) Scope** may be divided into:
    - ✓ ***Product*** scope - the detailed description of the product's quality, features, and functions.
    - ✓ ***Project*** scope - the work required to deliver a product or service with the intended product scope.
  - 2) **Time** refers to the time required to complete the project.
  - 3) **Cost** encompasses all the resources required to carry out the project.
- Cost vs. Quality
  - The quality of a system will normally impact its cost.

Project Management versus Process Management

"Ultimately, the parallels between process and project management give way to a fundamental difference: ***process*** management seeks to eliminate ***variability*** whereas ***project*** management must accept ***variability*** because each project is unique."

Elton, J. & J. Roe. "Bringing Discipline to Project Management" *Harvard Business Review*



Success vs. Failure

- What is the difference between "Success" and "Failure"?
  - **"I DO NOT HAVE TIME"**

Why do Projects Fail?

Studies have shown that the following factors contribute significantly to project failure:

- Improper focus of the project management system
- Fixation on first estimates
- Wrong level of detail
- Lack of understanding about project management tools; too much reliance on project management software
- Too many people
- Poor communication
- Rewarding the wrong actions

Failed IS Projects

- Standish Group found that 67 percent of all software projects are challenged
  - Late, over budget or fail to meet performance criteria.
- Managing a business project means managing an information systems project.
  - **Why?**
  - Many systems use or integrate the Internet.

Successful IS Projects

- To succeed, "a" general manager must be a project manager and must learn how to manage this type of risk.
- Executive management no longer has an option but to consider skilled IT project management as fundamental to business success.

The Need for Project Management

- Critical for companies today: the ability to adapt existing business processes faster than the competition
- Typical adaptation projects include:
  - "***Rightsizing***" the organization (*what is another name?*)
  - Reengineering business processes
  - Adopting more comprehensive, integrative processes

Project Management

- Changes in any one of the sides of the **triangle** affect one or both of the other sides.
  - Scope creep - increasing the scope after a project has begun.
- The project stakeholders decide on the overriding "**key success factor**" (i.e., time, cost, or scope).
  - The project manager is responsible for demonstrating to stakeholders the impact of the key success factors on the project.
  - Stakeholders are concerned about all facets of the project.
  - Measuring and tracking progress by tracking time, cost, scope, resources, quality, and risks.

Project Management - Business Case

- The business case spells out the components of the project and sets the foundation.
  - Argues resources for the project.
  - Clearly articulates the details of the project and contingency plans.
  - Implementation issues, areas of concern, and gaps are first identified in the planning phase.
- A **strong business plan** gives the project team a reference document to help guide decisions and activities.
- Project management software (e.g., Microsoft Project, Intuit Quickbase, Basecamp):
  - Tracks team members, deliverables, schedules, budgets, priorities, tasks, and other resources.
  - Provides a dashboard of key metrics.

PROJECT ELEMENTS

Essential Project Elements

- There are *four* components essential for any project and necessary to assure a high probability of **project success**:

1. Project **management**.
   - A project sponsor and a project manager are needed so that project can be coordinated and executed appropriately
2. A project **team**.
   - to ensure all parts of the project come together effectively and correctly (make sure to clearly define the teams objectives).
3. A project cycle ***plan.***
   - The methodology and schedule to execute the project (Gantt charts, CPM, and PERT diagrams).
   - The sequential steps of organizing and tracking the work of the team.
   - Method and schedule
4. A common project **vocabulary**.
   - so all team members can communicate effectively (very important as many are new

Project leadership

- Lack of **leadership** can result in unmotivated or confused people.
- Strong project leaders skillfully manage team composition, reward systems, and other techniques to focus, align, and motivate team members.
- Figure 10.4 reflects the inverse relationship between the magnitude of the project leader's role and the experience and **commitment** of the team.
- Factors influencing the project managers and team's performance:
  - Organizational culture.
  - Socioeconomic influences.

Project Team

- A project team consists of those people who work together to complete the project.
- Teams fail because members don't understand the nature of the work required.
- **Teamwork** should:
  - Clearly define the team's objectives.
  - Define each member's role in achieving these objectives.
  - Have norms about conduct, shared rewards, a shared understanding of roles, and team spirit.
- Project managers should leverage team member **skills**, knowledge, experiences, and capabilities.
- Team members should share information about their departments.

Project Cycle Plan

- The project cycle plan organizes discrete project activities, sequencing them into steps along a time line.
  - Therefore, the project delivers according to the requirements of customers and stakeholders.
- Identifies critical beginning and ending dates and breaks the work spanning these dates into phases
- The three most common approaches (and software tools) are:
  - Project Evaluation and Review Technique (PERT) (Figure 10.5):
    - **Estimates about the time needed to complete project tasks, calculating the optimistic, most probable, and pessimistic time requirements for completing each task.**
  - Critical Path Method (CPM): *deterministic* task times.
    - If any activity on critical path *delayed,* the overall project time will be *increased*
  - Gantt chart: displaying time relationships of project tasks and monitoring the progress toward project completion (Figure 10.6)
- Figure 10.7 compares both a generic project cycle plan and the Project Management Institute's project life cycle.

Project Management - Key Players

- The project **sponsor**:
  - liaises between the project team and other stakeholders.
  - is a project champion providing leadership.
  - is a senior C-level executive with influence with the key stakeholders and C-level team.
  - provides the financial resources for the project.
- The project **manager**:
  - Requires a range of management skills to make the project successful.

The Project Manager Skills

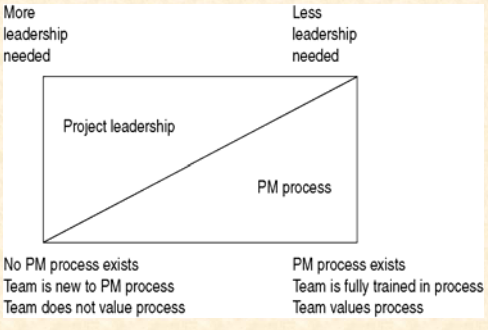
- A Project Manager's skills include:

1. *Identifying requirements* of the systems to be delivered.
2. Providing *organizational integration* by defining the team's structure.
3. Assigning *team* members to work on the project (*team mgt.*)
4. Managing *risks* and leveraging *opportunities*.
5. Measuring the project's status, outcomes, and exception to provide *project control*.
6. Making the *project visible* to general management and other stakeholders (*visibility*)
7. Measuring *project status* against the plan, often using project management software.
8. Taking *corrective action* when necessary to get the project back on track.
9. Project ***leadership***.

guide

The major focus of the status element of management is "*proactive*" as there is a need "strong" of project leaders to help the organization develop project competency to begin with.

Figure 10.4 Project leadership vs. project management (PM) process.



Project Manager's Role

- The project manager will typically be involved in:
  - Ensuring progress of the project according to defined metrics..
  - **Identifying risks.**
  - Ensuring progress toward deliverables within **time** and **resource constraints**
  - Running coordination meetings.
  - Negotiating for resources on behalf of the project.

But, not to determining the best fit of the project in the organizations vision

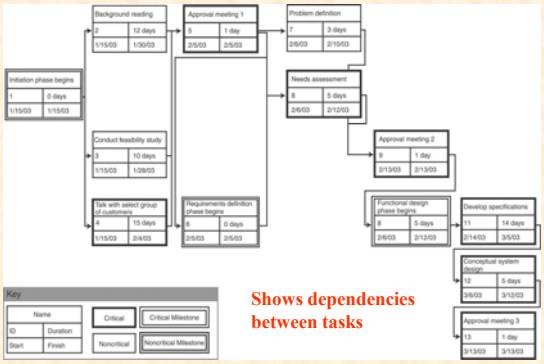
- Business projects are often initiated because of a successful business case.
  - A successful project begins with a well-written business case (i.e., spells out components of the project.)

PROJECT CYCLE PLAN (cont.)

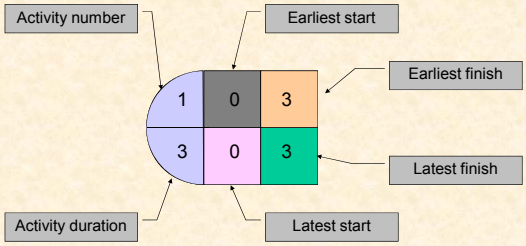
Project Cycle Plan Software

- PERT:
  - Identifies the **tasks**, orders the tasks in a time sequence, identifies their interdependencies, and estimates the time required to complete the task.
    - ✓**Critical tasks** - must be performed individually; together they account for the total elapsed time of the project.
    - ✓Non-critical tasks - can be built into the schedules without affecting the duration of the entire project.
- CPM:
  - A tool that is similar to PERT.
  - Incorporates a capability for identifying **relationships** between costs and the completion date of a project as well as the amount and value of resources that must be applied in alternative situations.

Figure 10.5 PERT chart



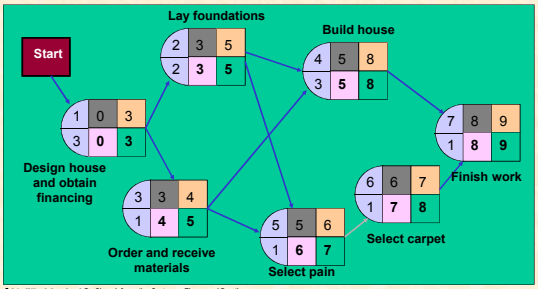
CPM - Node Configuration



Activity of Latest Start and Finish Times

#2, LF = min(LS followings) = min(5,6) = 5
LS = LF-t = 5-2 = 7

#9, LF =9
LS=LF-t=9-1



Activity of Latest Start and Finish Times

#2, LF = min(LS followings) = min(5,6) = 5
LS = LF-t = 5-2 = 7

#9, LF =9
LS=LF-t=9-1

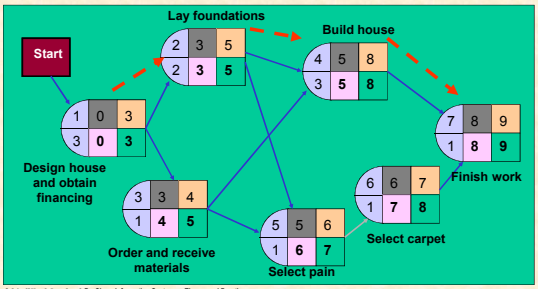


Figure 10.6 Gantt Chart

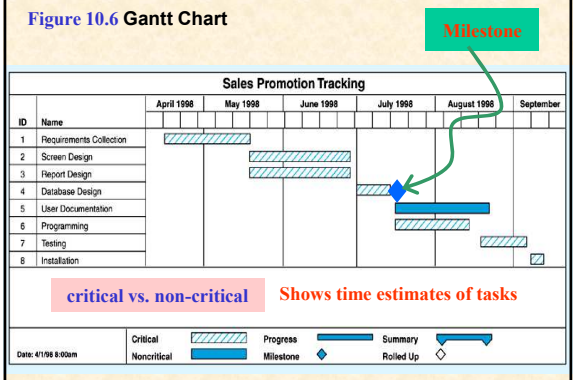


Figure 10.7 Project cycle template.

Requirements Definition Period			Production Period			Deployment/ Dissemination Period	
Investigation Task Force							
User requirement definition	Research concept definition	Information use specification	Collection planning phase	Collection and analysis phase	Draft report phase	Publication phase	Distribution phase
Typical High Tech Commercial Business							
Product requirement phase	Product definition phase	Product proposal phase	Product develop-ment phase	Engineer model phase	Internal test phase	Production phase	Manufacturing, sales & support phase
Generic Project Cycle Template							
User require-ment definition phase	Concept definition phase	System specificatio n phase	Acquisitio n planning phase	Source selectio n phase	Developmen t phase	Verificatio n phase	Deploy-ment or production phase	Operations/ maintenanc e or sales support phase	Deactivat e phase

Project Milestone

- A milestone represents an event or condition that marks the completion of a group of related tasks or the completion of a phase of the project.
- It is an interim goal or checkpoint in the project. It is like a task with duration of zero.
- Purpose: Milestones help us organize tasks into logical groups or sequences. They also help us note the progress of the project.

IT PROJECT DEVELOPMENT METHODOLOGIES

IT Project Development Methodologies and Approaches

- The choice of development methodologies and managerial influences distinguish IT projects from other projects.
- The systems development life cycle (**SDLC**) - a traditional tool for developing IS or implementing software developed by an *outsourcing* provider or software developer.
- Other development approaches:
  - **Agile** development
  - **Prototyping**
  - Rapid applications development (RAD)
  - Joint applications development (JAD)

Project Development Methodologies

- The choice of development methodologies and managerial influences distinguish IT projects from other projects.
- There are four main methodologies IT professionals use to manage the technology projects:
  - Systems Development Life Cycle (SDLC)
  - Prototyping
  - Rapid applications development (RAD)
  - Joint applications development (JAD)

Systems Development Life Cycle

- **Systems Development:** a set of activities used to create an IS.
- **Systems Development Life Cycle (SDLC):** the process of designing and delivering the entire system.
- The SDLC generally is used in one of two distinct ways:
  - ➢ as a general project **plan** of all activities required for the entire system to operate.
    - ✓Plan includes the analysis and feasibility study, the development or acquisition of components, the implementation activities, the maintenance activities, and the retirement activities.
  - ➢ as a process to **design** and **develop** system software.
    - ✓Process is highly structured, disciplined, and formal.

Systems Development Life Cycle

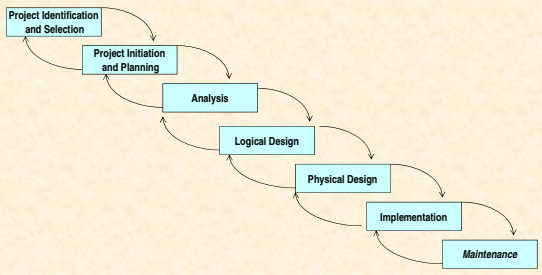
SDLC typically consists of seven phases

1. Initiation of the project
2. The requirements definition phase
3. The functional design phase
4. The system is actually built
5. Verification phase
6. The "cut over" where the new system is put in operation and all links are established. Possible conversion methods
   a) *Parallel*
   b) *Direct*
   c) *Phased in/out*
   d) *pilot*

   Which one is the best approach?
   Sabre mini case
7. The maintenance and review phase

See Figure 10.8 for more information on each step.

Systems Development Life Cycle: Another View



Traditional SDLC Methodology Issues

- Several problems arise with using traditional SDLC methodology:
  - ➢ Many systems projects fail to meet objectives.
    - ✓The **skills** needed to estimate costs and schedules are difficult to obtain.
    - ✓Each project is unique.
  - ➢ The objectives may reflect a scope that is too broad or too narrow.
    - ✓The problem the system was designed to solve may still exist.
  - ➢ Organizations need to respond quickly.
    - ✓Not enough time available to adequately do each step of the SDLC for each IT project.
  - ➢ Newer methodologies designed to address these concerns use an ***iterative*** **approach** (Figure 10.9).

Figure 10.9 Iterative approach to systems development.

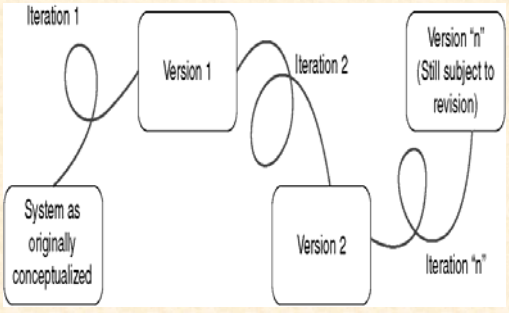
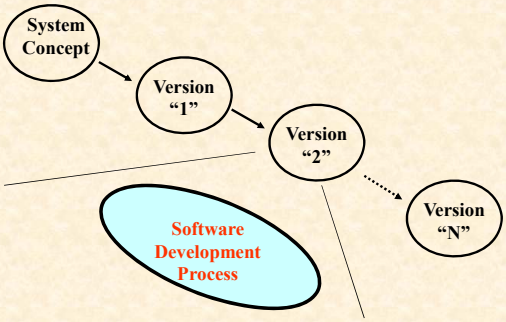


Figure 10.9 Iterative approach to systems development



Agile Development

- Agile development methodologies were developed for situations where a predictable development process **cannot** be followed.
  - E.g., XP (Extreme Programming), Crystal, Scrum, Feature-Driven Development, and Dynamic System Development Method (DSDM).
- Agile development is **people-oriented** rather than process oriented.
  - Adapts to changing requirements by iteratively developing systems in small stages and then testing the new code extensively.
  - The mantra for agile programming is "Code a little; test a little."
- DSDM is an extension of Rapid Application Development (RAD) used in the UK and is based on the underlying principles of active user interaction, frequent deliveries, and **empowered teams**.

Agile Development (cont.)

- DSDM:
  - incorporates a project planning technique that divides the schedule into a number of separate time periods (timeboxes) with each part having its own deliverables, deadline, and budget.
  - is based on four types of **iterations**:
    - ✓Study (business and feasibility).
    - ✓Functional model.
    - ✓Design and build.
    - ✓Implementation.
- XP is a more prescriptive agile methodology.
  - XP revolves around 12 **practices**, including pair programming, test-driven development, simple design, and small releases.
- Some disadvantages include difficulty estimating the required effort easily getting off track if the customer is unclear about final outcomes.

Prototyping

- **Prototyping**:
  - is a type of evolutionary development.
  - builds a ***fast***, ***high-level*** version of the system at the beginning of the project.
- **User involvement**.
  - Users see the day-to-day growth of the system and contribute frequently to the development process.
- Prototyping can be used as a phase in the SDLC to capture project requirements.
- **Drawbacks** to prototyping:
  - Documentation may be difficult to write.
  - Users may not understand the realistic scope of the system.
    - ✓The final prototype may not be scalable to an operational version.
  - An operational version may be difficult to complete.
  - The process can be difficult to manage.
  - Difficult to integrate across a broad range of requirements.
    - ✓Suitable for "quick-and dirty" types of systems.
  - System design flaws may be more prevalent.
- Various approaches are summarized in Figure 10.10.

Prototyping

- SDLC may not work for all situations, requires a lot of planning and is difficult to implement quickly.
- Prototyping is a type of evolutionary development.
- Builds a ***fast***, ***high-level*** version of the system at the beginning of the project.
- Advantages include:
  - User involvement and comment early on and throughout the development process.
- Disadvantages include:
  - Documentation may be difficult to write.
  - Users may not understand the realistic scope of the system.
    - ✓The final prototype may not be scalable to an operational version.

Prototyping

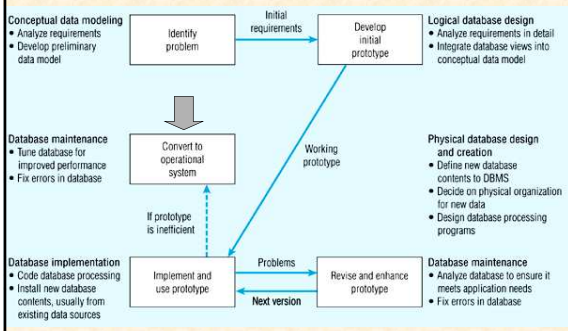
A prototyping is a small, but working system that contains only those important (not complete) features.

- Prototpying is one of the most popular rapid application development (RAD) methods.
- It is an iterative process of system development in which requirements are converted to a **working** system that is continually revised through close work between analysts and users.

Figure 10.10 Comparison of IT development methodologies.

Methodology	Advantages	Disadvantages
SDLC	• Structured approach with mile-stones and approvals for each phase • Uses system approach • Focuses on goals and trade-offs • Emphasizes documentation • Requires user sign-offs	• Systems often fail to meet objectives • Needed skills are often difficult to obtain • Scope may be defined too broadly or too narrowly • Very time consuming
Agile Development	• Good for adapting to changing requirements • Good for understanding and responding to changing user requirements • Allows face-to-face communica-tion and continuous inputs from users • Speeds up development process • Users like it	• Hard to estimate system deliverables at start of project • Under-emphasis of designing and documentation • Easy to get project off-track if user not clear about what the final outcome should be
Prototyping	• Improved user communications • Users like it • Speeds up development process • Good for eliciting system requirements • Provides a tangible model to serve as basis for production version	• Often under-documented • Not designed to be an operational version • Often creates unrealistic expectations • Difficult-to-manage development process • Integration often difficult • Design flaws more prevalent than in SDLC • Often hard to maintain

The *prototyping methodology*



Other Development Methodologies and Approaches

- Rapid applications development (**RAD**), joint applications development (**JAD**), Object-oriented analysis, design and development, and the open sourcing approach.
- **RAD** is similar to prototyping in that it is an interactive process in which tools are used to drastically speed up the development process.
  - Has tools for developing the user interface, graphical user interface (GUI), reusable code, code generation, and programming language testing and debugging.
  - Enables the developer to build a library of standard sets of code—or objects—used and reused in multiple applications.
  - "Drags and drops" objects into the design.
    - ✓ Automatically writes the code necessary to include that functionality.

Other Development Methodologies and Approaches

- Joint applications development (**JAD**):
  - Is a version of RAD or prototyping.
  - Has users that are more integrally involved.
  - Uses a group approach to elicit requirements by interviewing groups of users.
  - Is expensive due to travel and living expenses needed to coordinate participants.
- **Object-oriented development**:
  - Is a way to avoid the pitfalls of procedural methodologies.
  - Builds on the concept of objects—or reusable components.
    - ✓ An object encapsulates both the data stored about an entity and the operations that manipulate that data.

Open Sourcing Approach

- **Linux**:
  - Was created by Linus Torvalds and several thousand hackers around the world.
  - Is a world-class OS—a clone of **Unix**.
  - Was built using a development approach called open sourcing, which is the process of building and improving "**free**" software via an Internet community.
    - ✓ Eric Raymond suggests that the Linux community resembles a great bazaar of differing agendas and approaches (with submissions from anyone) out of which a coherent and stable system emerged.
- Software is open source software (**OSS**) if it is released under a license approved by the Open Source Initiative (OSI).
  - The most widely used OSI license is the general public license (GPL), which is based on the concept of free software.

Free Software

- Free software offers the following **freedoms** for the software users:
  - The freedom to run the program for any purpose.
  - The freedom to study how the program works and adapt it to your needs. Access to the source code is a precondition for this.
  - The freedom to distribute copies so that you can help your neighbor.
  - The freedom to improve and release your improvements to the public so that the whole community benefits. Access to source code is a precondition for this.
- A user who modifies the software must observe the rule of copyleft.
  - Copyleft - a user cannot add restrictions to deny others their central freedoms regarding the free software.

Open Sourcing Movement

- The Open Sourcing Movement.
  - Offers a **speedy** way to develop software that:
    - ✓is available to a whole community.
    - ✓uses widespread testing.
    - ✓is free.
- A number of managerial issues are associated with its use in a business organization.
  - Preservation of intellectual property.
  - Updating and maintaining open source code.
  - Competitive advantage.
  - Tech support.
  - Standards.

Popular Open Source Software

- Examples of popular open source:
  - Software Mozilla (a popular web browser core).
  - Apache (web server).
  - PERL (web scripting language).
  - OpenOffice (a Sun Microsystems-originated set of office applications that support the Microsoft Office suite formats).
  - PNG (graphics file format).
- Open source applications available on the **Internet**, including **Web 2.0** applications, are becoming part of the corporate infrastructure.

MANAGING IT PROJECT RISK

- IT projects are often distinguished from many non-IT projects on the basis of their high levels of **risk**.
- Risk is the possibility of additional cost or loss due to the choice of alternative.
  - Some alternatives have a lower associated risk than others.
  - Risk can be quantified by assigning a **probability** of occurrence and a financial consequence to each alternative.
- Risk is to be considered as a function of:
  - Complexity
  - Clarity
  - Size

Assessing Project Risk

		Clarity: Low Structure	Clarity: High Structure
Low Company-Relative Technology	Large Project	Low risk (very susceptible to mismanagement)	Low risk
	Small Project	Very low risk (very susceptible to mismanagement)	Very low risk
High Company-Relative Technology	Large Project	Very high risk	Medium risk
	Small Project	High risk	Medium-low risk

Managing IT Reosource Thru Strategic Partnerships; A Portfolio Approach to IT Development

Complexity

- The **complexity level** is the extent of difficulty and interdependent components of the project.
- Several factors contribute to greater complexity in IT projects:
  - The sheer pace of technological change.
  - The degree of uncertainty in identifying and agreeing on common goals.
- Complexity can be determined once the **context** of the project has been established.

Clarity

- Clarity is concerned with the ability to define the **requirements** of the system.
  - A project has low clarity if the users cannot easily state their needs or define what they want from the system.
  - A project with high clarity is one in which the systems requirements can be easily documented and do not change.

Size

- Size plays a big role in project **risk**.
- A project can be considered big if it has:
  - a large budget relative to other budgets in the organization.
  - a large number of **team** members or number of man-months.
  - a large number of organizational units involved in the project.
  - a large number of programs/components.
  - a large number of function points or lines of code.

Managing Project Risk Level

- The project's **complexity, clarity, and size** determine its risk.
  - Varying levels of these three determinants affect the amount of project risk.
  - Large, highly complex projects that are low in clarity are extremely risky.
  - Small projects that are low in complexity and high in clarity are usually low risk.
  - Everything else is somewhere in between.
- The level of risk determines how formal and detailed the project management system and planning should be.
- When it is difficult to estimate duration or expense of a project because it is complex or has low clarity, formal management practices or planning may be inappropriate.
- Formal planning tools may be useful in low-risk projects.

Managing the Complexity Aspects of Project Risk

- Strategies that may be adopted in dealing with complexity are:
  - **Leveraging** the **technical skills** of the team.
    - ✓ Having a leader or team members who have had significant work experience.
  - Relying on consultants and vendors.
    - ✓ Their work is primarily project-based and they usually possess the crucial IT knowledge and skills.
  - Integrating within the organization.
    - ✓ Having frequent team meetings, documenting critical project decisions, and holding regular technical status reviews.
    - ✓ Requires good communication among team members.

Managing Clarity Aspects of Project Risk

- When a project has low clarity, project managers need to:
  - rely more heavily upon the users to define system requirements.
  - manage **stakeholders**.
    - ✓ Managers must balance the goals of the various stakeholders to achieve desired project outcomes.
    - ✓ Often involves both the project manager and the general manager.
  - Sustain project **commitment** (Figure 10.11).
    - ✓ Four primary types of determinants of project commitment:
      - Project determinants.
      - Psychological determinants.
      - Social determinants.
      - Organizational determinants.

Figure 10.11 Determinants of commitment for IT projects.

Determinant	Description	Example
Project	Objective attributes of the project such as cost, benefits, expected difficulty, and duration.	Projects are more likely to have higher commitment if they involve a large potential payoff.
Psychological	Factors managers use to convince themselves things are not so bad, such as previous experience, personal responsibility for outcome, and biases.	Projects are more likely to have higher commitment when there is a previous history of success.
Social	Elements of the various groups involved in the process, such as rivalry, norms for consistency, and need for external validation.	Projects are more likely to have higher commitment when external stakeholders have been publicly led to believe the project will be successful.
Organizational	Structural attributes of the organization, such as political support, and alignment with values and goals.	Projects are more likely to have higher commitment when there is strong political support from executive levels.

Measuring Success

- At the start of the project, the general manager should consider several aspects based on achieving the business goals.
- Care is needed to prevent a too narrow or too broad set of goals.
- It is important that the goals be measurable so that they can be used throughout the project to provide the project manager with feedback.

MEASURES OF INFORMATION SYSTEM SUCCESS

1. HIGH LEVELS OF USE
2. USER SATISFACTION
3. FAVORABLE ATTITUDES
4. ACHIEVED OBJECTIVES
5. FINANCIAL PAYOFF

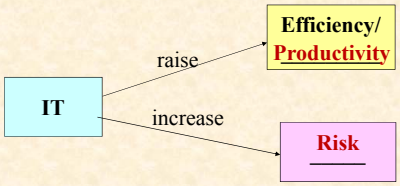
Gauging Success

- At the start of the project, the general **manager** should:
  - ➢consider several aspects based on achieving the business **goals**.
    - ✓The goals be measurable and should be used throughout the project to provide the project manager with feedback.
  - ➢assess if the system meets the specifications and project **requirements** laid out in the project scope.
    - ✓***Metrics*** may be derived specifically from the requirements and business needs.
    - ✓Four dimensions that are useful in determining if a project is successful or not (Figure 10.12):
      - – Resource constraints.
      - – Impact on customers.
      - – Business success.
      - – Prepare the future.

Figure 10.12 Success dimensions for various project types.

Success Dimension	Low Tech	Medium Tech	High Tech
	Existing technologies with new features	*Most technologies are new but available before the project*	*New, untested technologies*
Resource Constraint	Important	Overruns acceptable	Overruns most likely
Impact on Customers	Added value	Significantly improved capabilities	Quantum leap in effectiveness
Business Success	Profit; return on investment	High profits; market share	High, but may come much later; market leader
Prepare the Future	Gain additional capabilities	New market; new service	Leadership-core and future technologies

IT and its Influences



MANAGING IT PROJECT RISK

- IT projects are often distinguished from many non-IT projects on the basis of their high levels of **risk**.
- Risk is the possibility of additional cost or loss due to the choice of alternative.
  - ➢ Some alternatives have a lower associated risk than others.
  - ➢ Risk can be quantified by assigning a **probability** of occurrence and a financial consequence to each alternative.
- Risk is to be considered as a function of:
  - ➢ **Complexity**
  - ➢ **Clarity**
  - ➢ **Size**

Assessing Project Risk

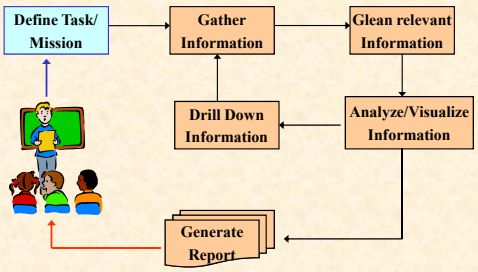
		Clarity: Low Structure	Clarity: High Structure
Low Company-Relative Technology	Large Project	Low risk (very susceptible to mismanagement)	Low risk
	Small Project	Very low risk (very susceptible to mismanagement)	Very low risk
High Company-Relative Technology	Large Project	Very high risk	Medium risk
	Small Project	High risk	Medium-low risk

Managing IT Reosource Thru Strategic Partnerships; A Portfolio Approach to IT Development

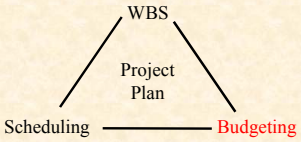
POOR PROJECT MANAGEMENT

- COST OVERRUNS
- TIME SLIPPAGE
- TECHNICAL SHORTFALLS IMPAIR PERFORMANCE
- FAILURE TO OBTAIN ANTICIPATED BENEFITS

Project Development Processes



Creating a Project Budget



The *budget is a plan* that identifies the resources, goals and schedule that allows a firm to achieve those goals

- Top-down
- Bottom-up
- Activity-based costing (ABC)

Activity-Based Costing

Projects use activities & activities use resources

1. Assign costs to activities that use resources
2. Identify cost drivers associated with this activity
3. Compute a cost rate per cost driver unit or transaction
4. Multiply the cost driver rate times the volume of cost driver units used by the project

Budget Contingencies

The allocation of extra funds to cover uncertainties and improve the chance of finishing on time.

Contingencies are needed because

- Project scope may change
- Murphy's Law is present
- Cost estimation must anticipate interaction costs
- Normal conditions are rarely encountered

Summary

- General manager fulfills an important role in project management.
- Project management involves continual trade-offs.
- Four important project elements: Common vocabulary, teamwork, project cycle plan, and project management.
- Important to understand the complexity of a project.
- SDLC, prototyping, JAD and RAD are used for development of IS systems.
- Manage project risk carefully.
- The PMO can be very useful.

END OF CHAPTER 11