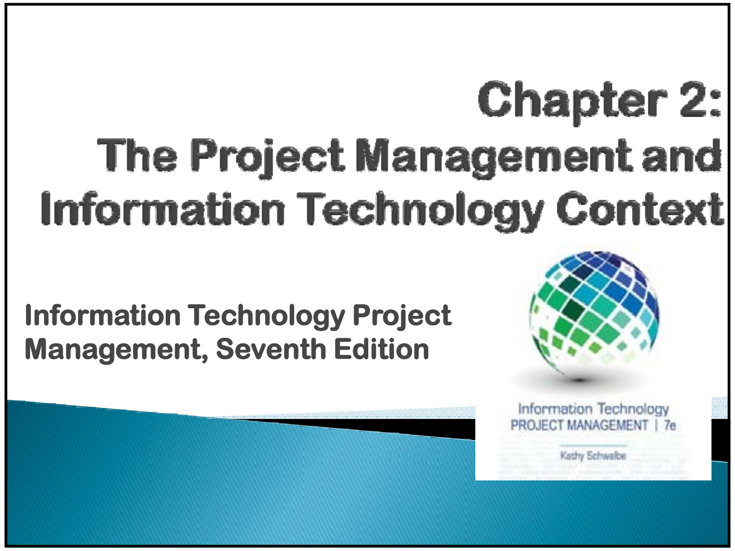
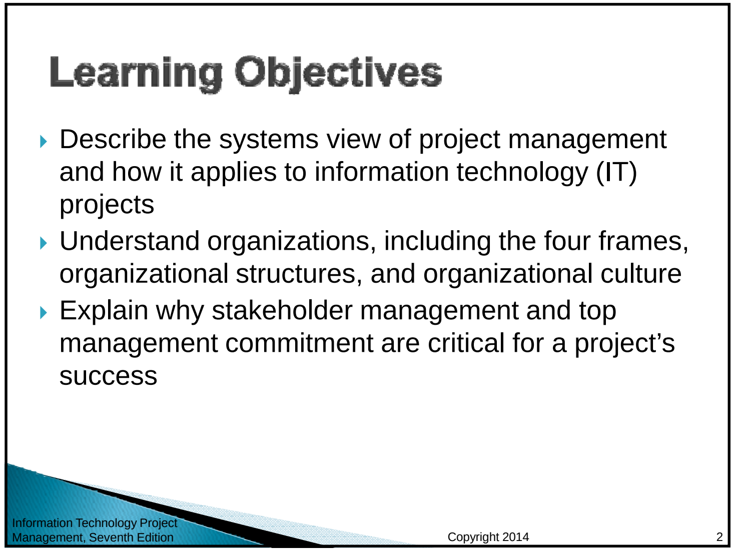


Chapter 2: The Project Management and Information Technology Context

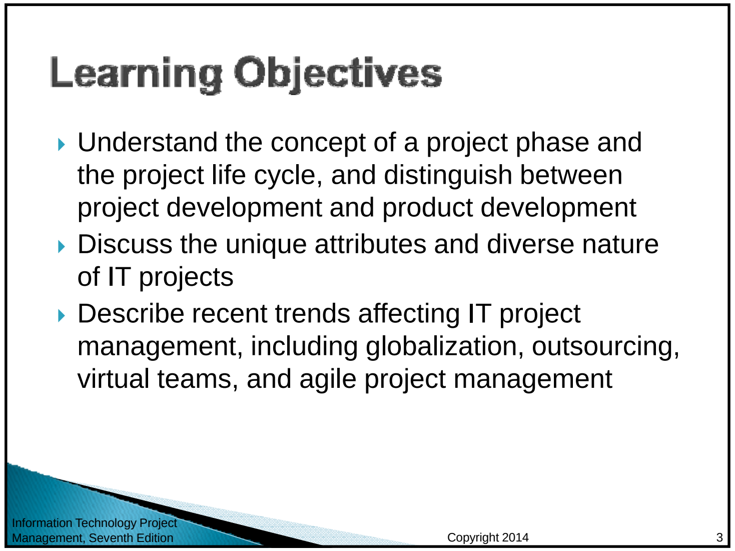
Information Technology Project Management, Seventh Edition

Information Technology
PROJECT MANAGEMENT | 7e
Kathy Schwalbe



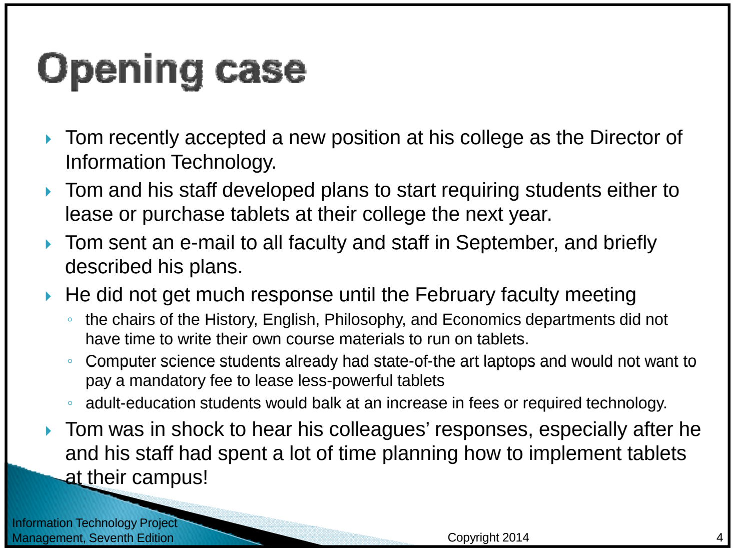
Learning Objectives

- Describe the systems view of project management and how it applies to information technology (IT) projects
- Understand organizations, including the four frames, organizational structures, and organizational culture
- Explain why stakeholder management and top management commitment are critical for a project's success



Learning Objectives

- Understand the concept of a project phase and the project life cycle, and distinguish between project development and product development
- Discuss the unique attributes and diverse nature of IT projects
- Describe recent trends affecting IT project management, including globalization, outsourcing, virtual teams, and agile project management



Opening case

- Tom recently accepted a new position at his college as the Director of Information Technology.
- Tom and his staff developed plans to start requiring students either to lease or purchase tablets at their college the next year.
- Tom sent an e-mail to all faculty and staff in September, and briefly described his plans.
- He did not get much response until the February faculty meeting
  - the chairs of the History, English, Philosophy, and Economics departments did not have time to write their own course materials to run on tablets.
  - Computer science students already had state-of-the art laptops and would not want to pay a mandatory fee to lease less-powerful tablets
  - adult-education students would balk at an increase in fees or required technology.
- Tom was in shock to hear his colleagues' responses, especially after he and his staff had spent a lot of time planning how to implement tablets at their campus!

Projects Cannot Be Run In Isolation

- Projects must operate in a broad organizational environment
- Project managers need to use **systems thinking:**
  - taking a holistic view of carrying out projects within the context of the organization
- Senior managers must make sure projects continue to support current business needs

A Systems View of Project Management

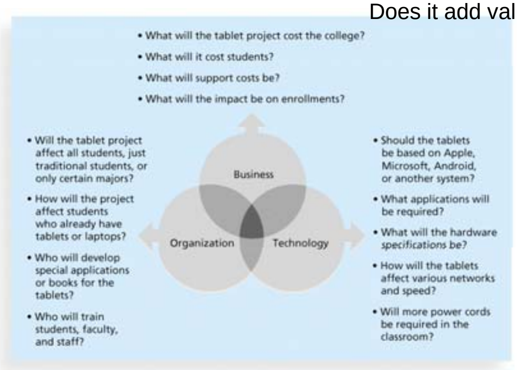
- A **systems approach** emerged in the 1950s to describe a more analytical approach to management and problem solving
- It includes:
  - **Systems philosophy**: an overall model for thinking about things as *systems*
  - **Systems analysis**: problem-solving approach(identify scope, decompose into components, identify problems, opportunities, constraints, needs, analise alternative solutions to identify a satisfactory solution considering the entire system)
  - **Systems management**: address *business*, *technological*, and *organizational* issues associated with creating, maintaining, and modifying a system

Systems approach

- Using a systems approach is *critical* to successful project management.
- Top management and project managers must follow a *systems philosophy* to understand how projects relate to the whole organization.
  - They must use systems analysis to address needs with a problem-solving approach.
  - They must use systems management to identify *key business, technological, and organizational issues* related to each project in order to identify and satisfy key *stakeholders* and do what is best for the *entire organization*.

Figure 2-1. Three Sphere Model for Systems Management

Does it add value?

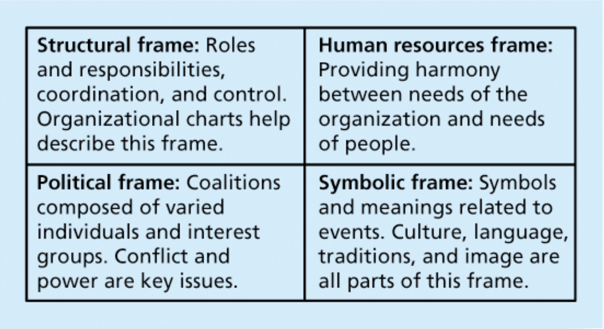


- What will the tablet project cost the college?
- What will it cost students?
- What will support costs be?
- What will the impact be on enrollments?

- Will the tablet project affect all students, just traditional students, or only certain majors?
- How will the project affect students who already have tablets or laptops?
- Who will develop special applications or books for the tablets?
- Who will train students, faculty, and staff?

- Should the tablets be based on Apple, Microsoft, Android, or another system?
- What applications will be required?
- What will the hardware specifications be?
- How will the tablets affect various networks and speed?
- Will more power cords be required in the classroom?

Understanding Organizations: focusing on different perspectives



Structural frame: Roles and responsibilities, coordination, and control. Organizational charts help describe this frame.	**Human resources frame:** Providing harmony between needs of the organization and needs of people.
Political frame: Coalitions composed of varied individuals and interest groups. Conflict and power are key issues.	**Symbolic frame:** Symbols and meanings related to events. Culture, language, traditions, and image are all parts of this frame.

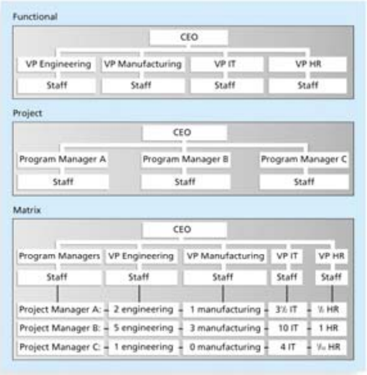
Organization

- **Definition Organization**: A set of *organizational* units and their different *relationships* with each other.
- Organizational units can be organized according to many different categories, for example by function or by project type. Typical examples of organizational units:
  - Functional organization: Research, Development, Marketing, Sales
  - Project organization: Project 1, Project 2, ….
- An organization usually has 3 different types of relationships between organizational units.
  - Reporting structure: To report status information
  - Decision structure: To propagating decisions
  - Communication structure: To exchange of information

Organizational Structures

- 3 basic organization structures
  - **Functional:** functional managers (VPs) report to the CEO; specialized skills
  - **Project:** program managers report to the CEO; staff with a variety of skills, project focused
  - **Matrix:** middle ground between functional and project structures;
    - personnel often report to two or more bosses (a VP and some PMs, different projects);
    - structure can be weak, balanced, or strong matrix (control made by PM)

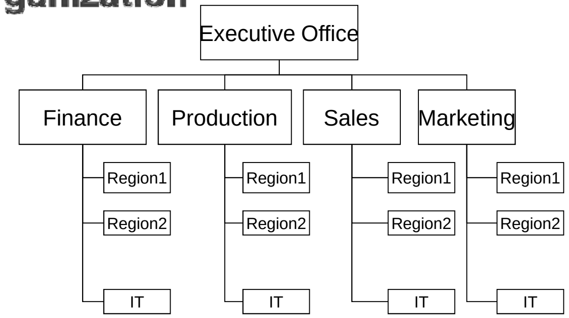
Functional, Project, and Matrix Organizational Structures



Functional Organization

- **Definition**: In a **functional organization** participants are grouped into so-called **departments**, each of which addresses a function.
- Examples of departments:
  - Traditional businesses: Research, development, production, sales, finance.
  - In software companies the departments correspond to the activities in the software process: Analysis, design, integration, testing departments.
- Key properties:
  - Projects are usually pipelined through the departments of a functional organization. The project starts in research, then it moves to development, then it moves to production, ….
  - Only a few participants are involved in the complete project.
  - Separate departments often address the same cross-functional needs (Examples: configuration management, IT infrastructure)

Example of a Functional Organization



Line organization of a „traditional business“

Properties of Functional Organizations

- Advantages:
  - Members of a department have a good understanding of the functional area they support.
  - Departments don‘t compete with another to get the support of their support teams
- Disadvantages:
  - Because each department has its own support team, different work procedures and reporting systems are the rule.
  - It is difficult to make major investments in equipment and facilities.
    - Example: Two departments with a budget of 50,000 Euro each need a printer that costs 100,000 Euro.
    - Both need only 50% of the maximum capacity.
    - Neither department can buy it, because they don‘t have sufficient funds.
  - High chance for overlap or duplication of work among departments

Project Organization

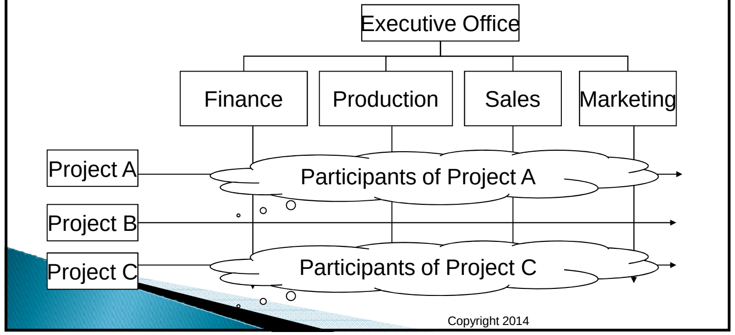
- In a **project organization** participants are grouped into **projects**, each of which has a problem to be solved within time and budget.
- **Key properties:**
  - Teams are assembled for a project as it is created. Each project has a project leader.
  - All participants are involved in the complete project.
  - Teams are disassembled when the project terminates

Properties of Project Organizations

- Advantages
  - Very responsive to new project requests (because the project is newly established and can be tailored around the problem)
  - New people can be hired/selected who are very familiar with the problem or who have special capabilities.
  - There is no waste of staff workload
- Disadvantages:
  - Teams cannot be assembled rapidly. Often it is difficult to manage the staffing/hiring process.
  - Because there are „no predefined lines“, roles and responsibilities need to be defined at the beginning of the project

Matrix Organization

- In a matrix organization, participants from different departments of the functional organization are assigned to work on projects as they are created.
- The project manager and team members may be assigned to the project for <= 100 % of their time



Properties of Matrix Organizations

- Advantages:
  - Teams for projects can be assembled rapidly
  - Scarce expertise can be applied to different projects as needed
  - Consistent work and reporting procedures can be used for projects of the same type.
- Disadvantages:
  - Team members usually are not familiar with each other
  - Team members have different working styles
  - Team members must get used to each other

New Challenges in Matrix Organizations

- Team members must respond to two different bosses with different focus:
  - Focus of the functional manager: Assignments to different projects, performance appraisal
  - Focus of the project manager: Work assignments, project team support
- Team members working on multiple projects have competing demands for their time
  - Team members working on more than one project have even more project members to report to
  - Some people who have claim on the team member‘s time may be at similar levels in the organization‘s hierarchy
- Multiple work procedures and reporting systems are used by different team members
  - Development of common procedures needs to be addressed at project kickoff time

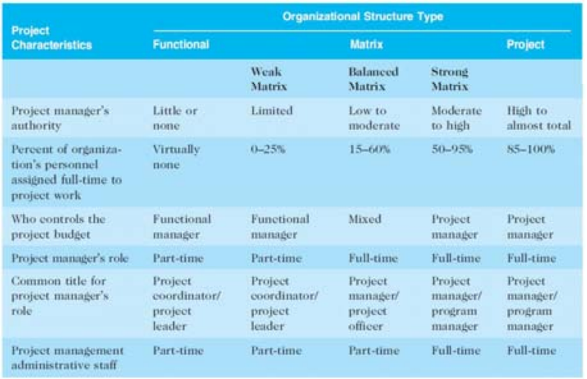
When to use a Functional Organization

- Projects with high degree of certainty, stability, uniformity and repetition.
  - Requires little communication
  - Role definitions are clear
- When?
  - The more people on the project, the more need for a formal structure
  - Customer might insist that the test team be independent from the design team
  - Project manager insists on a previously successful structure

When to Use a Project or Matrix Organization

- Project with degree of uncertainty
  - Open communication needed among members
  - Roles are defined on project basis
- When?
  - Requirements change during development
  - New technology develops during project

Table 2-1. Organizational Structure Influences on Projects

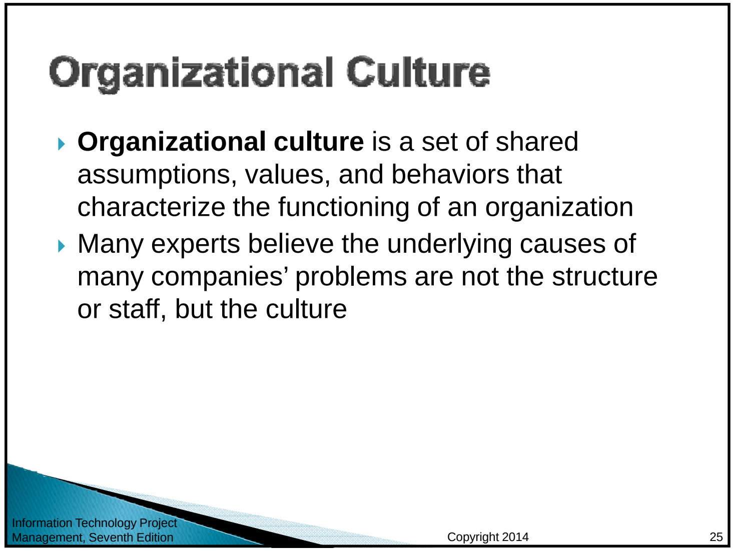


Project Characteristics	Organizational Structure Type				
	Functional	Matrix			Project
		Weak Matrix	Balanced Matrix	Strong Matrix	
Project manager's authority	Little or none	Limited	Low to moderate	Moderate to high	High to almost total
Percent of organization's personnel assigned full-time to project work	Virtually none	0–25%	15–60%	50–95%	85–100%
Who controls the project budget	Functional manager	Functional manager	Mixed	Project manager	Project manager
Project manager's role	Part-time	Part-time	Full-time	Full-time	Full-time
Common title for project manager's role	Project coordinator/ project leader	Project coordinator/ project leader	Project manager/ project officer	Project manager/ program manager	Project manager/ program manager
Project management administrative staff	Part-time	Part-time	Part-time	Full-time	Full-time

© Cengage Learning 2014

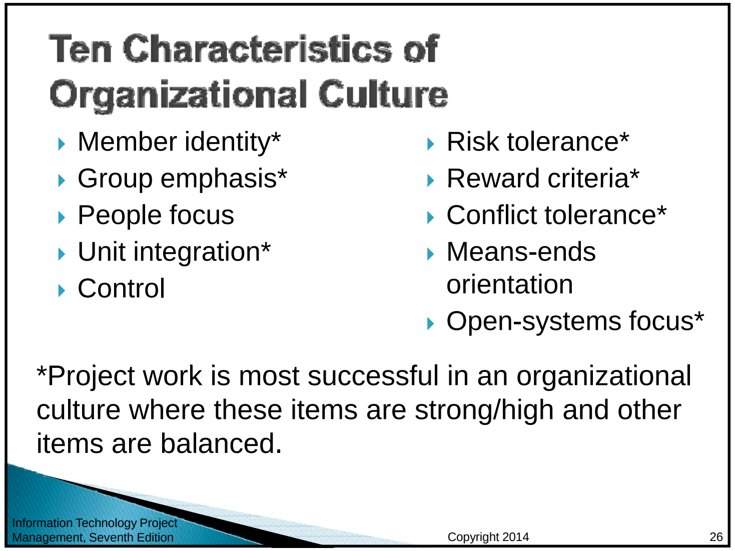
Organizational influence

- Functional organization
  - Project leader needs to ask top management sponsorship. Sponsor should
    - solicit the VPs for giving HR,
    - assign separate budget for project-related trips, meeting, training, incentives
- Project organization
  - Most authority but inefficient for the organization as a whole (underutilization and misallocation of resources)
- Systems approach
  - Hiring an independent contractor instead of using a full-time employee



Organizational Culture

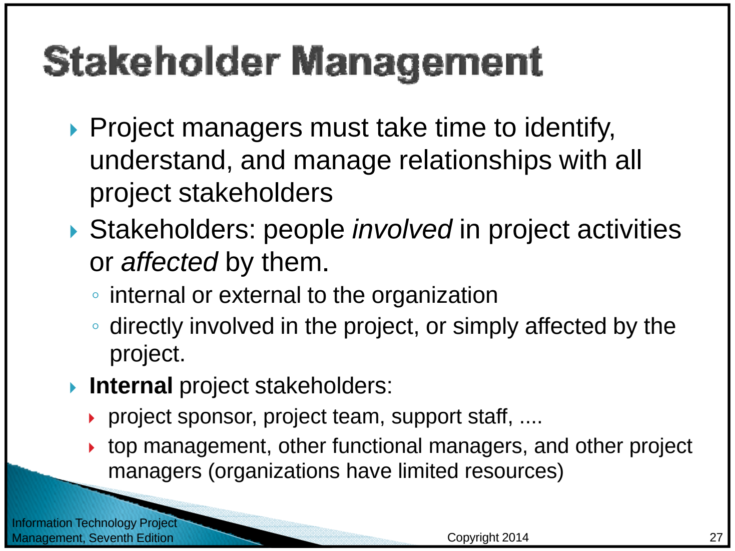
- **Organizational culture** is a set of shared assumptions, values, and behaviors that characterize the functioning of an organization
- Many experts believe the underlying causes of many companies' problems are not the structure or staff, but the culture



Ten Characteristics of Organizational Culture

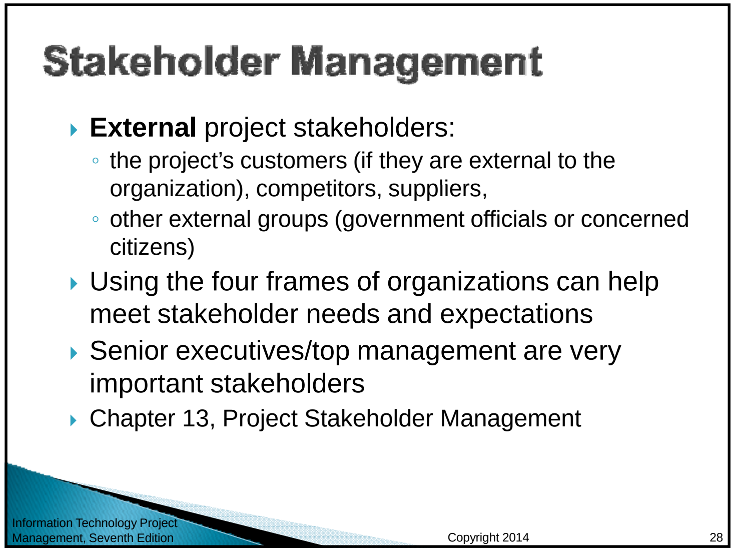
- Member identity*
- Group emphasis*
- People focus
- Unit integration*
- Control
- Risk tolerance*
- Reward criteria*
- Conflict tolerance*
- Means-ends orientation
- Open-systems focus*

*Project work is most successful in an organizational culture where these items are strong/high and other items are balanced.



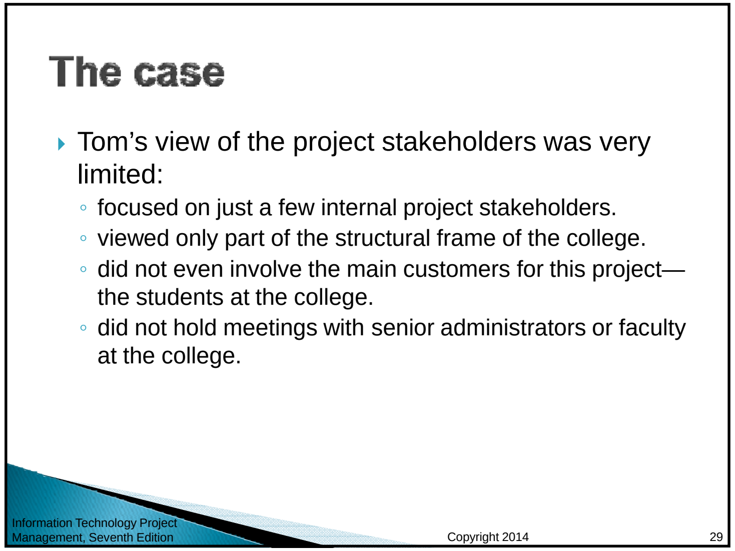
Stakeholder Management

- Project managers must take time to identify, understand, and manage relationships with all project stakeholders
- Stakeholders: people *involved* in project activities or *affected* by them.
  - internal or external to the organization
  - directly involved in the project, or simply affected by the project.
- **Internal** project stakeholders:
  - project sponsor, project team, support staff, ...
  - top management, other functional managers, and other project managers (organizations have limited resources)



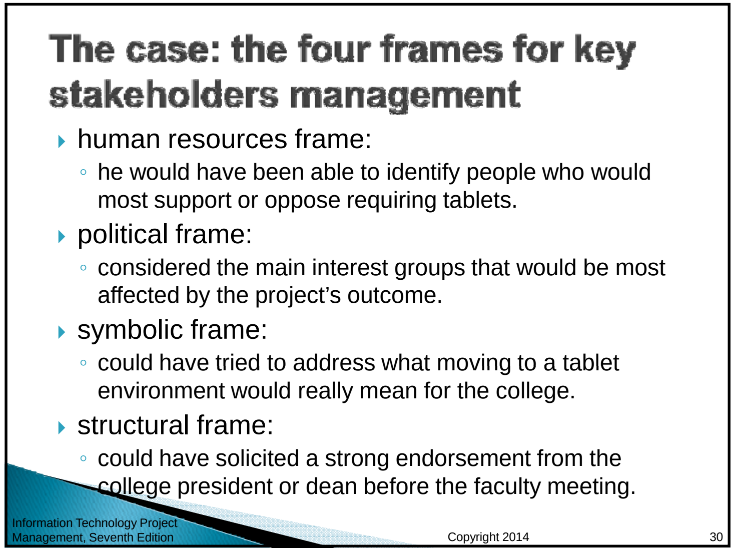
Stakeholder Management

- **External** project stakeholders:
  - the project's customers (if they are external to the organization), competitors, suppliers,
  - other external groups (government officials or concerned citizens)
- Using the four frames of organizations can help meet stakeholder needs and expectations
- Senior executives/top management are very important stakeholders
- Chapter 13, Project Stakeholder Management



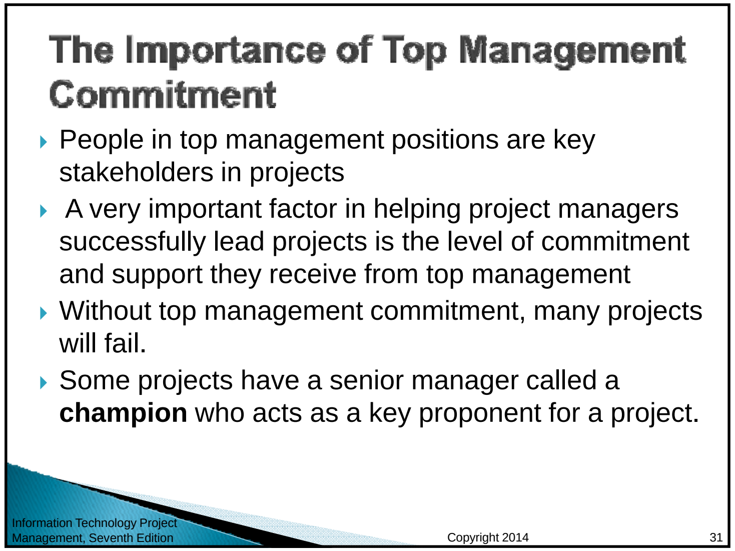
The case

- Tom’s view of the project stakeholders was very limited:
  - focused on just a few internal project stakeholders.
  - viewed only part of the structural frame of the college.
  - did not even involve the main customers for this project—the students at the college.
  - did not hold meetings with senior administrators or faculty at the college.



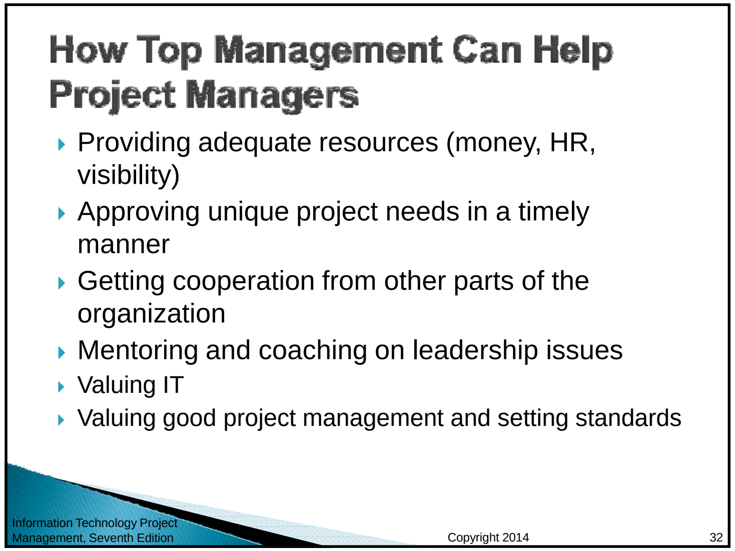
The case: the four frames for key stakeholders management

- human resources frame:
  - he would have been able to identify people who would most support or oppose requiring tablets.
- political frame:
  - considered the main interest groups that would be most affected by the project’s outcome.
- symbolic frame:
  - could have tried to address what moving to a tablet environment would really mean for the college.
- structural frame:
  - could have solicited a strong endorsement from the college president or dean before the faculty meeting.



The Importance of Top Management Commitment

- People in top management positions are key stakeholders in projects
- A very important factor in helping project managers successfully lead projects is the level of commitment and support they receive from top management
- Without top management commitment, many projects will fail.
- Some projects have a senior manager called a **champion** who acts as a key proponent for a project.

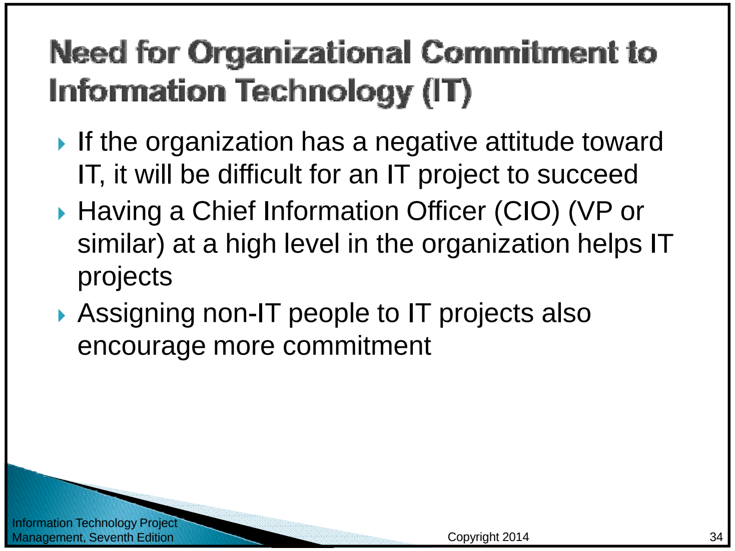


How Top Management Can Help Project Managers

- Providing adequate resources (money, HR, visibility)
- Approving unique project needs in a timely manner
- Getting cooperation from other parts of the organization
- Mentoring and coaching on leadership issues
- Valuing IT
- Valuing good project management and setting standards

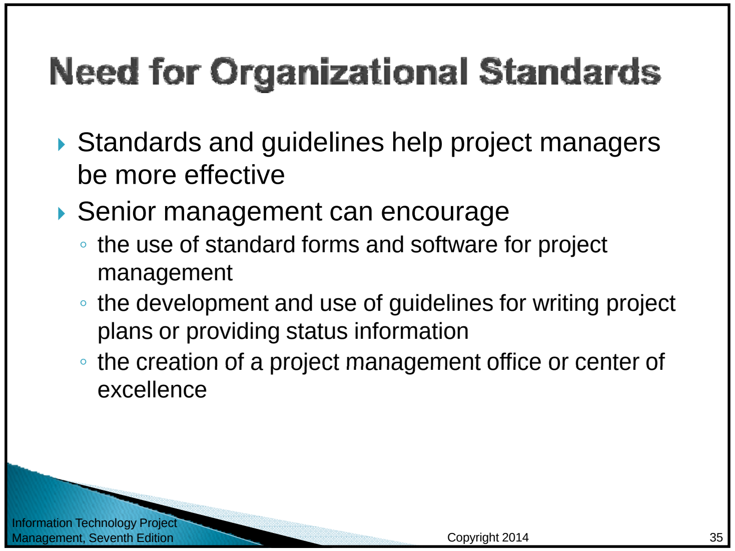
Best Practice

- **IT governance** addresses the authority and control for key IT activities in organizations, including IT infrastructure, IT use, and project management, in a uniform way
- A lack of IT governance can be dangerous, as evidenced by three well-publicized IT project failures in Australia (Sydney Water's customer relationship management system, the Royal Melbourne Institute of Technology's academic management system, and One.Tel's billing system)



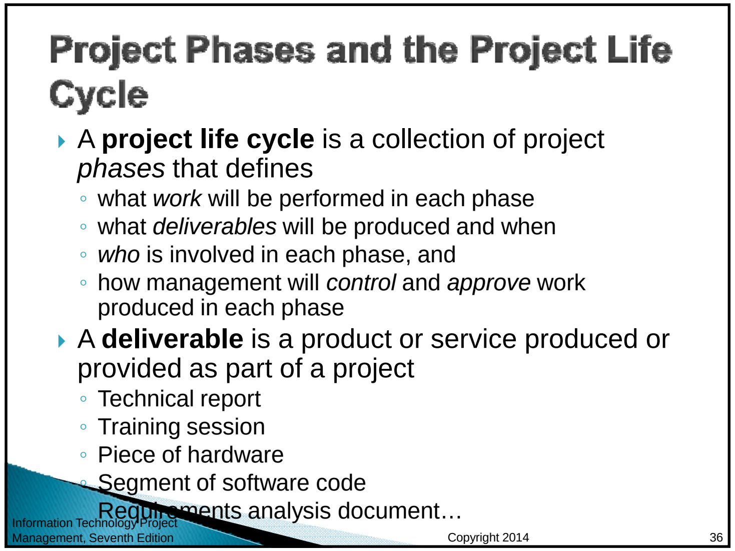
Need for Organizational Commitment to Information Technology (IT)

- If the organization has a negative attitude toward IT, it will be difficult for an IT project to succeed
- Having a Chief Information Officer (CIO) (VP or similar) at a high level in the organization helps IT projects
- Assigning non-IT people to IT projects also encourage more commitment



Need for Organizational Standards

- Standards and guidelines help project managers be more effective
- Senior management can encourage
  - the use of standard forms and software for project management
  - the development and use of guidelines for writing project plans or providing status information
  - the creation of a project management office or center of excellence



Project Phases and the Project Life Cycle

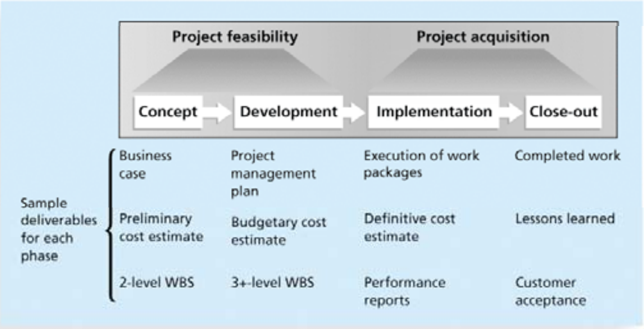
- A **project life cycle** is a collection of project *phases* that defines
  - what *work* will be performed in each phase
  - what *deliverables* will be produced and when
  - *who* is involved in each phase, and
  - how management will *control* and *approve* work produced in each phase
- A **deliverable** is a product or service produced or provided as part of a project
  - Technical report
  - Training session
  - Piece of hardware
  - Segment of software code
  - Requirements analysis document…

More on Project Phases

- In early phases of a project life cycle
  - resource needs are usually lowest
  - the level of uncertainty (risk) is highest
  - project stakeholders have the greatest opportunity to influence the project
- In middle phases of a project life cycle
  - the certainty of completing a project improves
  - more resources are needed
- The final phase of a project life cycle focuses on
  - ensuring that project requirements were met
  - the sponsor approves completion of the project
- Phases vary by project or industry

Phases of the Traditional Project Life Cycle

PMBOK: Starting project, organizing and preparing, carrying out the project work, finishing

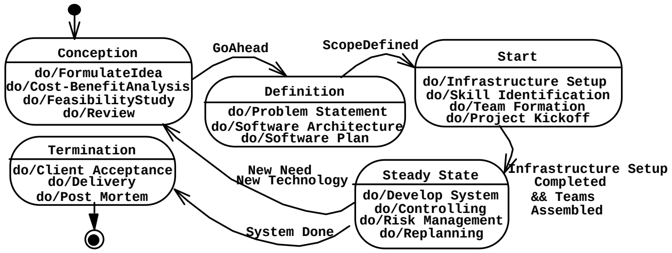


	Project feasibility		Project acquisition	
	Concept	Development	Implementation	Close-out
Sample deliverables for each phase	Business case	Project management plan	Execution of work packages	Completed work
	Preliminary cost estimate	Budgetary cost estimate	Definitive cost estimate	Lessons learned
	2-level WBS	3+-level WBS	Performance reports	Customer acceptance

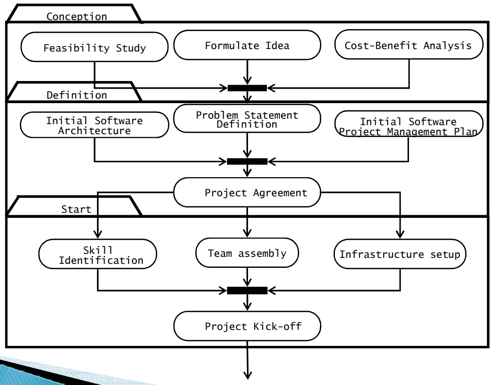
Some concepts

- Business case
  - the need for the project
  - basic underlying concepts
  - a preliminary cost estimate
  - an overview of the needed work
- Work breakdown structure (WBS)
  - Outlines project work decomposing the activities into different levels of taks

States of a Project



Management activities in a software project (continued on next slide).



Conception

Feasibility Study

Formulate Idea

Cost-Benefit Analysis

Definition

Initial Software Architecture

Problem Statement Definition

Initial Software Project Management Plan

Project Agreement

Start

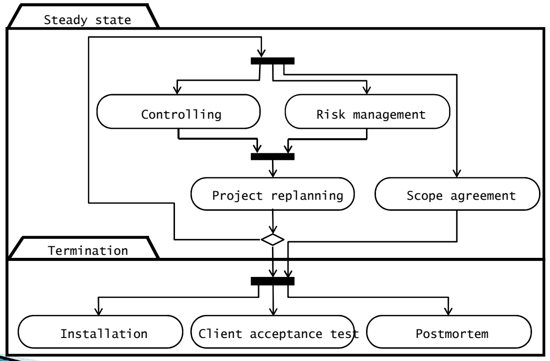
Skill Identification

Team assembly

Infrastructure setup

Project Kick-off

Management activities in a software project (continued).



Steady state

Controlling

Risk management

Project replanning

Scope agreement

Termination

Installation

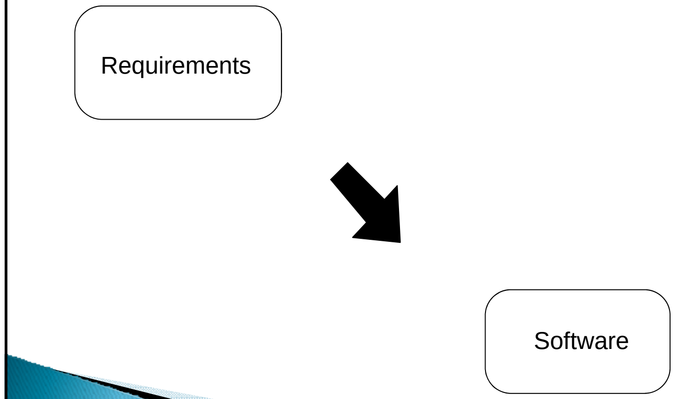
Client acceptance test

Postmortem

Product Life Cycles

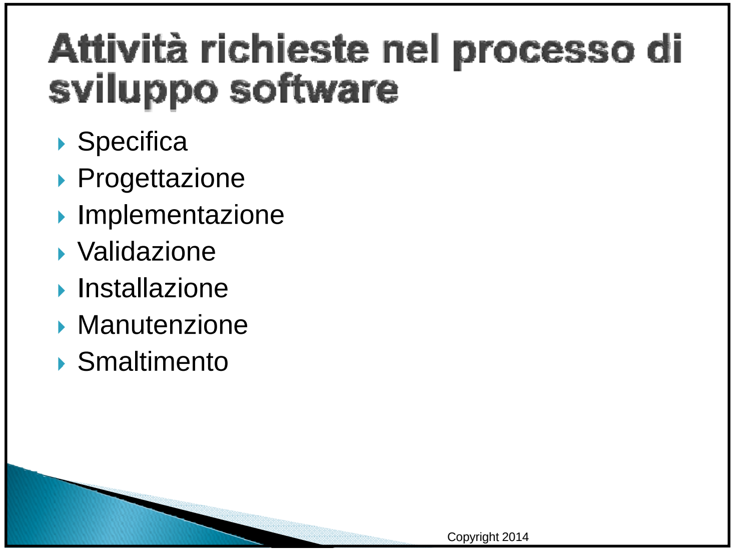
- Products also have life cycles
- The **Systems Development Life Cycle (SDLC)** is a term used to describe a *process* for:
  - developing (planning, creating, testing, and deploying)
  - maintaining sw
- Un processo è un particolare *metodo* per *fare qualcosa* costituito da una **sequenza di passi** che coinvolgono **attività, vincoli e risorse** (Pfleeger)
- ***Processo software:* un metodo per sviluppare del software**

What we intend



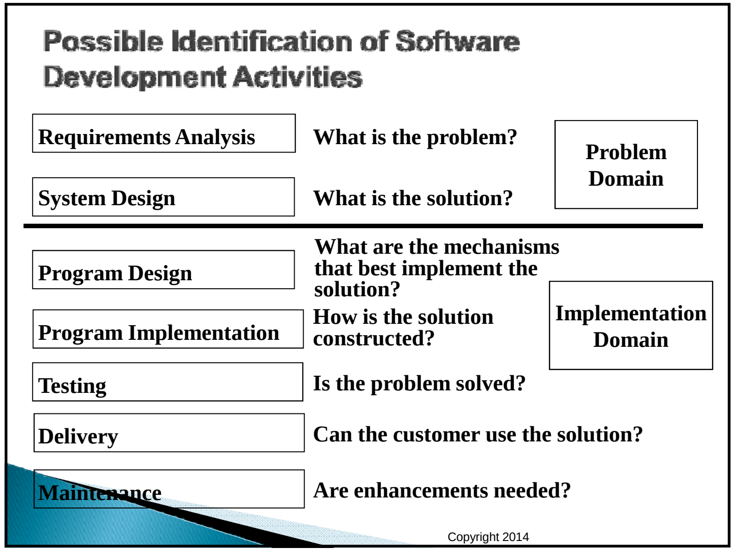
Requirements

Software



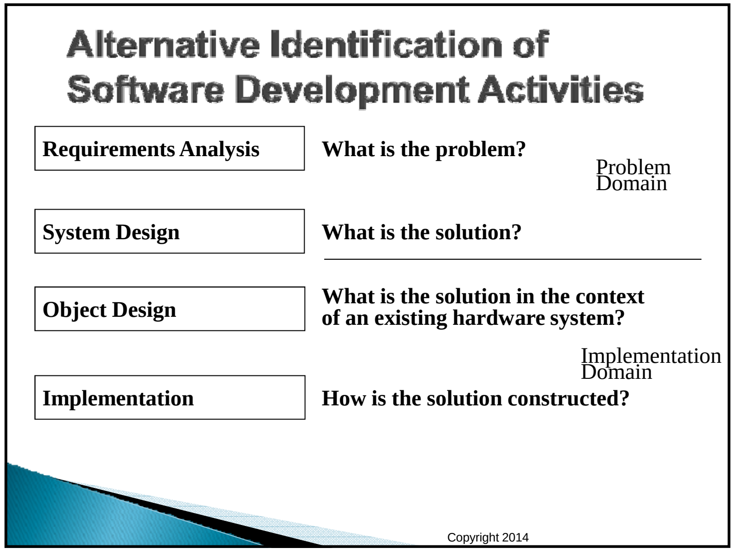
Attività richieste nel processo di sviluppo software

- Specifica
- Progettazione
- Implementazione
- Validazione
- Installazione
- Manutenzione
- Smaltimento



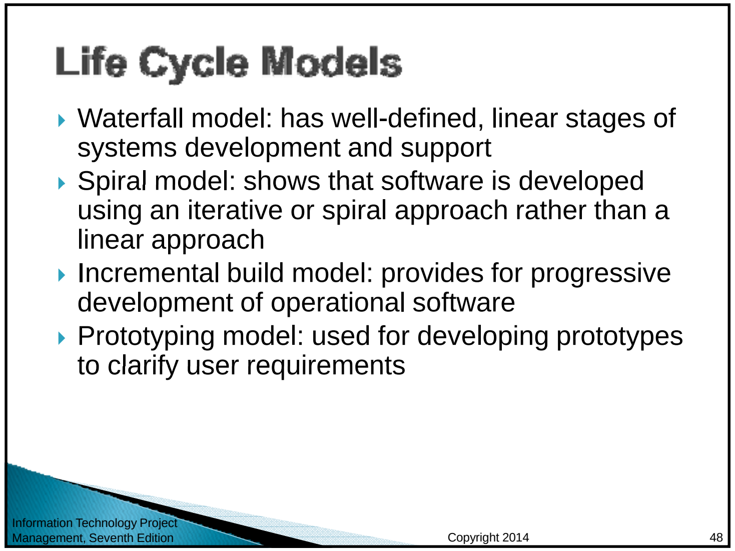
Possible Identification of Software Development Activities

Requirements Analysis	What is the problem?	Problem Domain
System Design	What is the solution?	
Program Design	What are the mechanisms that best implement the solution?	Implementation Domain
Program Implementation	How is the solution constructed?	
Testing	Is the problem solved?	
Delivery	Can the customer use the solution?	
Maintenance	Are enhancements needed?	



Alternative Identification of Software Development Activities

Requirements Analysis	What is the problem?	Problem Domain
System Design	What is the solution?	
Object Design	What is the solution in the context of an existing hardware system?	Implementation Domain
Implementation	How is the solution constructed?	



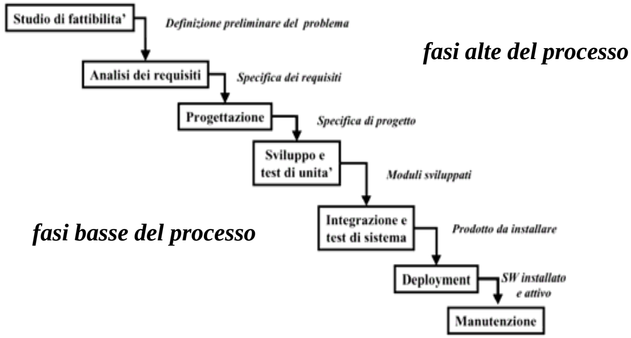
Life Cycle Models

- Waterfall model: has well-defined, linear stages of systems development and support
- Spiral model: shows that software is developed using an iterative or spiral approach rather than a linear approach
- Incremental build model: provides for progressive development of operational software
- Prototyping model: used for developing prototypes to clarify user requirements

Modelli a Cascata (Waterfall) - Royce '70

- Popolare negli anni '70
  - reazione al "code and fix" originario
  - ispirazione dall'industria manifatturiera
- Modello sequenziale lineare
  - progressione sequenziale (in cascata) di fasi, senza ricicli, al fine di meglio controllare tempi e costi
  - definisce e separa le varie fasi e attività del processo
    - nullo (o minimo) overlap fra le fasi
  - uscite intermedie: semilavorati del processo (documentazione di tipo cartaceo, programmi)
    - formalizzati in struttura e contenuti
  - consente un controllo dell'evoluzione del processo
    - attività trasversali alle diverse fasi

Modello a Cascata



Modelli a Cascata: organizzazione sequenziale delle fasi

- ogni fase raccoglie un insieme di attività omogenee per metodi, tecnologie, skill del personale, etc.
- ogni fase è caratterizzata dalle attività (tasks), dai prodotti di tali attività (deliverables), dai controlli relativi (quality control measures)
- la fine di ogni fase è un punto rilevante del processo (**milestone**)
- i semilavorati output di una fase sono input alla fase successiva
- i prodotti di una fase vengono "congelati", ovvero non sono più modificabili se non innescando un processo formale e sistematico di modifica
- Il limite del modello a cascata è la difficoltà ad effettuare cambiamenti nel corso del processo

Modelli a cascata e modelli di processo industriali (1)

- Un esempio: costruzione di una casa …
  - … vorrei un casa su 2 piani, con autorimessa e cantina ...
  - … prospetti, piantine, …
  - … planimetrie, piante assonometrie, ...
  - … calcoli statici travi, infrastruttura elettrica, idrica,...
  - … costruzione di pilastri, muri, impianto idrico ...
  - … verifica che il prodotto rispetti i desiderata del richiedente, nonché norme e/o standard, rilascio di certificati di idoneità all'uso
  - …il richiedente ci va a vivere ...
  - … dopo un po': riverniciatura delle pareti, riparazione del tetto, sopraelevazione, modifiche alla suddivisione in stanze,

Modelli a cascata e modelli di processo industriali (2)

- **Specifica Requisiti** - definizione di requisiti e vincoli del sistema (vorrei un casa su 2 piani, con autorimessa e cantina…)
- **Progetto di Sistema** - produrre un modello cartaceo (planimetrie, piante assonometrie, …)
- **Progetto di Dettaglio** - modelli dettagliati della parti (progetto dell'infrastruttura elettrica, idrica, calcoli statici travi...)
- **Costruzione** - realizzare il sistema
- **Test dei Componenti** - verifica le parti separatamente (impianto elettrico, idraulico, portata solai…)
- **Test di Integrazione** - integra le parti (il riscaldamento funziona...)
- **Test di Sistema** - verifica che il sistema rispetti le specifiche richieste
- **Installazione** - avvio e consegna del sistema ai clienti (ci vado a vivere)
- **Manutenzione** - (cambio la guarnizione al lavandino che perde …)

Studio di fattibilità

- Valutazione preliminare di costi e benefici
- Varia a seconda della relazione committente/ produttore
- **Obiettivo**
  - Stabilire se avviare il progetto, individuare le possibili opzioni e le scelte più adeguate, valutare le risorse umane e finanziarie necessarie
- **Output** : documento di fattibilita'
  - definizione preliminare del problema
  - scenari - strategie alternative di soluzione
  - costi, tempi, modalita' di sviluppo per ogni alternativa

Analisi dei requisiti

- Analisi completa dei bisogni dell'utente e dominio del problema
- Coinvolgimento di committente e ingegneri del SW
- **Obiettivo**
  - Descrivere le caratteristiche di qualita' che l'applicazione deve soddisfare

CHE COSA ☑ COME ⊘

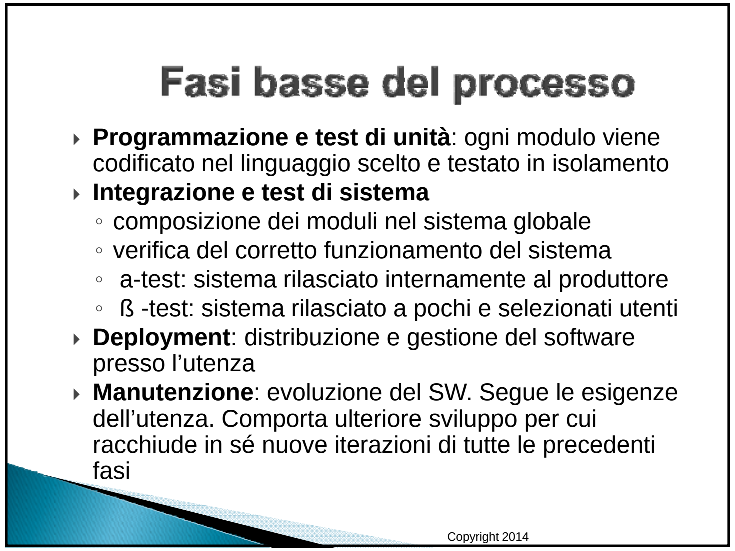
- **Output**: documento di specifica dei requisiti
  - manuale d'utente
  - piano di *acceptance test* del sistema

Progettazione

- Definizione di una struttura opportuna per il SW
- Scomposizione del sistema in componenti e moduli
  - allocazione delle funzionalità ai vari moduli
  - definizione delle relazioni fra i moduli
- Distinzione fra:
  - *architectural design*: struttura modulare complessiva (componenti)
  - *detailed design*: dettagli interni a ciascuna componente
- **Obiettivo**

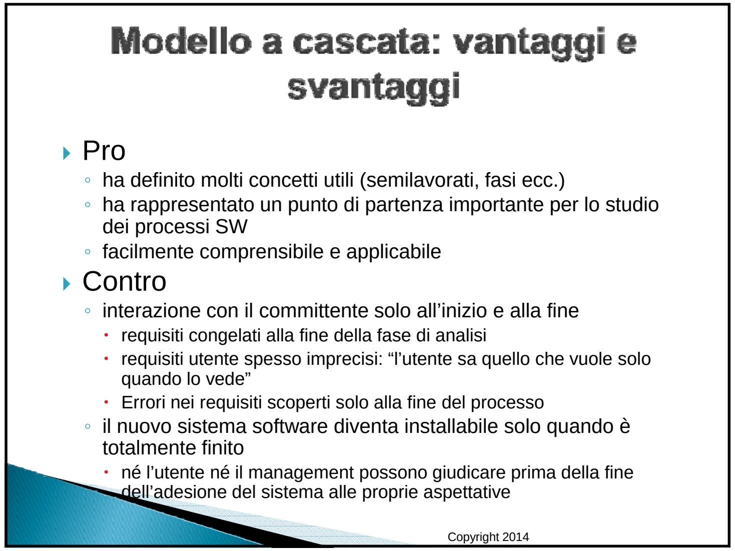
CHE COSA ⊘ ☑ COME

- **Output**: documento di specifica di progetto
  - possibile l'uso di linguaggi/ formalismi per la progettazione



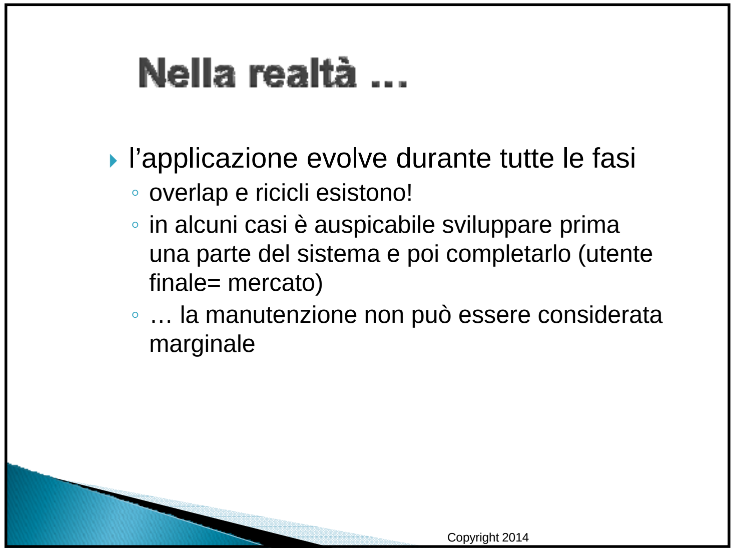
Fasi basse del processo

- **Programmazione e test di unità**: ogni modulo viene codificato nel linguaggio scelto e testato in isolamento
- **Integrazione e test di sistema**
  - composizione dei moduli nel sistema globale
  - verifica del corretto funzionamento del sistema
  - a-test: sistema rilasciato internamente al produttore
  - ß -test: sistema rilasciato a pochi e selezionati utenti
- **Deployment**: distribuzione e gestione del software presso l'utenza
- **Manutenzione**: evoluzione del SW. Segue le esigenze dell'utenza. Comporta ulteriore sviluppo per cui racchiude in sé nuove iterazioni di tutte le precedenti fasi



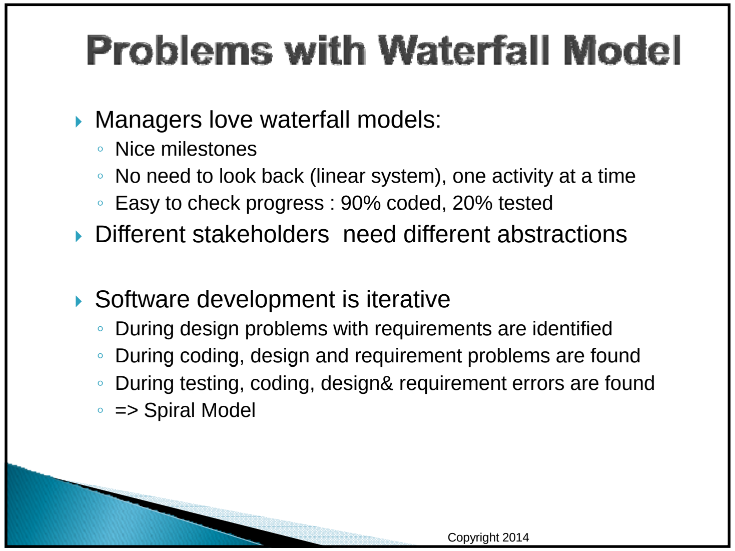
Modello a cascata: vantaggi e svantaggi

- Pro
  - ha definito molti concetti utili (semilavorati, fasi ecc.)
  - ha rappresentato un punto di partenza importante per lo studio dei processi SW
  - facilmente comprensibile e applicabile
- Contro
  - interazione con il committente solo all'inizio e alla fine
    - requisiti congelati alla fine della fase di analisi
    - requisiti utente spesso imprecisi: "l'utente sa quello che vuole solo quando lo vede"
    - Errori nei requisiti scoperti solo alla fine del processo
  - il nuovo sistema software diventa installabile solo quando è totalmente finito
    - né l'utente né il management possono giudicare prima della fine dell'adesione del sistema alle proprie aspettative



Nella realtà …

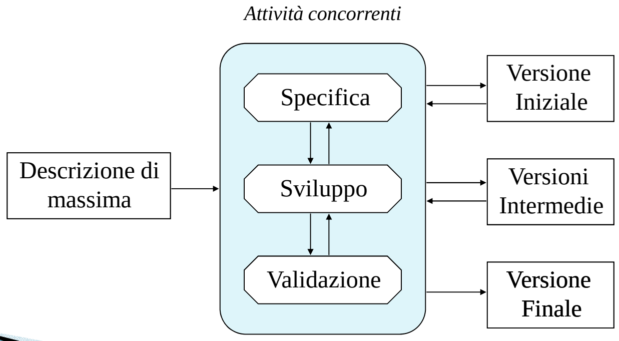
- l'applicazione evolve durante tutte le fasi
  - overlap e ricicli esistono!
  - in alcuni casi è auspicabile sviluppare prima una parte del sistema e poi completarlo (utente finale= mercato)
  - … la manutenzione non può essere considerata marginale



Problems with Waterfall Model

- Managers love waterfall models:
  - Nice milestones
  - No need to look back (linear system), one activity at a time
  - Easy to check progress : 90% coded, 20% tested
- Different stakeholders need different abstractions
- Software development is iterative
  - During design problems with requirements are identified
  - During coding, design and requirement problems are found
  - During testing, coding, design& requirement errors are found
  - => Spiral Model

Modelli evolutivi



Attività concorrenti

Descrizione di massima

Specifica

Sviluppo

Validazione

Versione Iniziale

Versioni Intermedie

Versione Finale

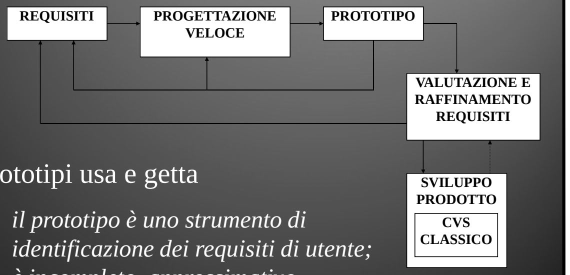
Modello evolutivo

- Prototipazione di tipo evolutivo (sviluppo esplorativo)
  - L'obiettivo è lavorare con il cliente ed evolvere verso il sistema finale a partire da una specifica di massima. Lo sviluppo inizia con le parti del sistema che sono già ben specificate, aggiungendo via via nuove caratteristiche
- Prototipazione di tipo usa e getta (throw-away)
  - L'obiettivo è capire i requisiti del sistema e quindi sviluppare una definizione migliore dei requisiti. Il prototipo sperimenta le parti del sistema che non sono ancora ben comprese

Modelli basati su prototipo throw-away (1)

- Un *prototipo* per aiutare a comprendere i requisiti o per valutare la fattibilità di un approccio
- Realizzazione di una prima implementazione (prototipo), più o meno incompleta da considerare come una 'prova', con lo scopo di:
  - accertare la **fattibilità** del prodotto
  - validare i **requisiti**
- *il prototipo è un mezzo attraverso il quale si interagisce con il committente per accertarsi di aver ben compreso le sue richieste, per specificare meglio tali richieste, per valutare la fattibilità del prodotto*
- dopo la fase di utilizzo del prototipo si passa alla produzione della versione definitiva del Sistema SW mediante un modello che, in generale, è di tipo waterfall

Modello con Prototipo Throw-away



REQUISITI

PROGETTAZIONE VELOCE

PROTOTIPO

VALUTAZIONE E RAFFINAMENTO REQUISITI

SVILUPPO PRODOTTO

CVS CLASSICO

- ◆ Prototipi usa e getta
  - *il prototipo è uno strumento di identificazione dei requisiti di utente; è incompleto, approssimativo, realizzato utilizzando parti già possedute o routines stub.*
  - *Il prototipo **deve essere gettato** !*

Modelli con prototipo Throw-away

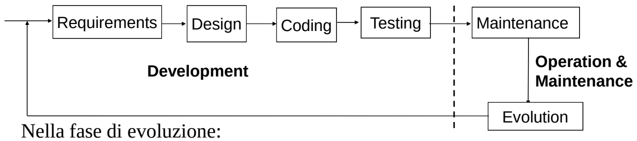
- 2 tipi di prototipazione:
  - **mock-ups**: produzione completa dell'interfaccia utente. Consente di definire con completezza e senza ambiguità i requisiti (si può, già in questa fase, definire il manuale di utente)
  - **breadboards**: implementazione di sottoinsiemi di funzionalità critiche del SS, nel senso dei vincoli pesanti che sono posti nel funzionamento del SS (carichi elevati, tempo di risposta, ...), senza le interfacce utente. Produce feedbacks su come implementare la funzionalità (in pratica si cerca di conoscere prima di garantire).

Prototyping vs Rapid Development

- Revolutionary prototyping is sometimes called rapid prototyping
- Rapid Prototyping is not a good term because it confuses *prototyping* with *rapid development*
  - **Prototyping is a technical issue: It is** a particular model in the life cycle process
  - **Rapid development is a management issue. It is a** particular way to control a project
- Prototyping can go on forever if it is not restricted
  - **"Time-boxed" prototyping limits the duration of the prototype development**

Sviluppo Esplorativo

- Il prodotto SW evolve continuamente (aggiunta di funzionalità, cambio di piattaforma, cambiamenti nella organizzazione dell'azienda che lo utilizza e quindi adeguamento del SW, ...).

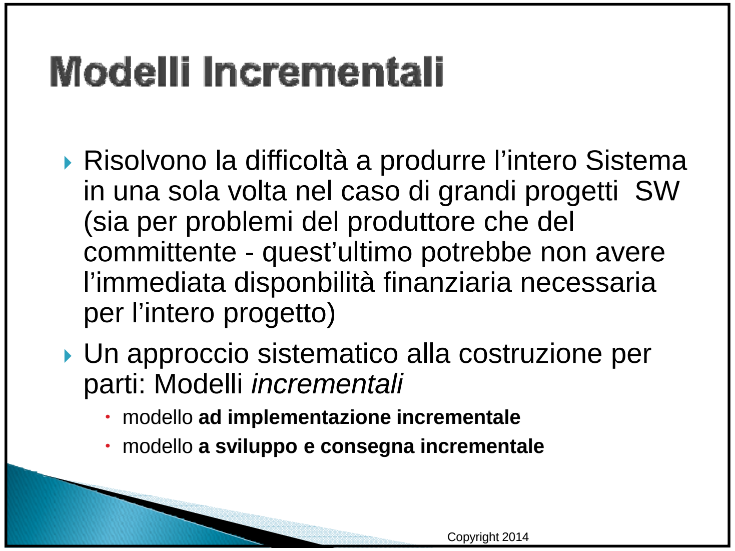


Nella fase di evoluzione:

- si analizza l'esperienza di uso del SW sul campo e si utilizza la maggiore conoscenza per definire nuovi obiettivi,
- si determinano esigenze e nuove funzionalità emerse o non coperti in precedenza,
- si riprende il ciclo dalla definizione dei requisiti alla messa in esercizio e manutenzione del nuovo SW nato dall'arricchimento ed evoluzione del precedente

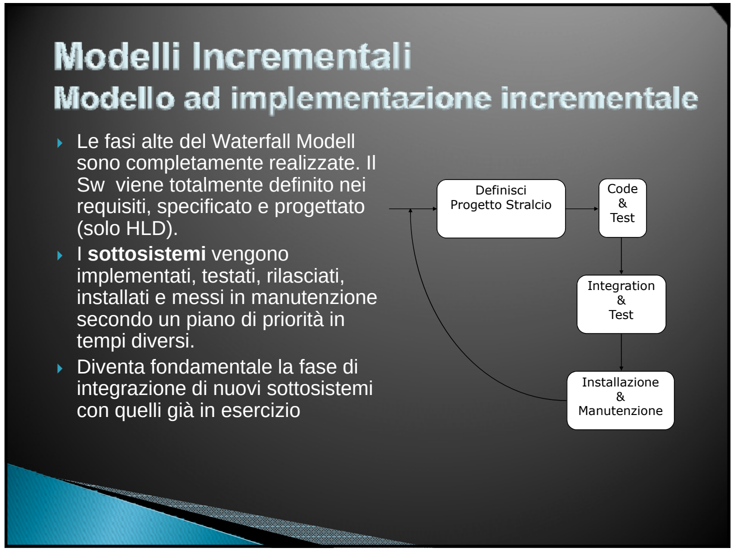
Modello evolutivo: Sviluppo esplorativo

- Problemi
  - Mancanza di visibilità del processo
  - Sistemi spesso poco strutturati
  - Possono essere richieste particolari capacità (ad esempio in linguaggi per prototyping)
- Applicabilità
  - Sistemi interattivi di piccola o media dimensione
  - Per parti di sistemi più grandi (es. interfaccia utente)
  - Per sistemi a vita breve



Modelli Incrementali

- Risolvono la difficoltà a produrre l'intero Sistema in una sola volta nel caso di grandi progetti SW (sia per problemi del produttore che del committente - quest'ultimo potrebbe non avere l'immediata disponbilità finanziaria necessaria per l'intero progetto)
- Un approccio sistematico alla costruzione per parti: Modelli *incrementali*
  - modello **ad implementazione incrementale**
  - modello **a sviluppo e consegna incrementale**



Modelli Incrementali
Modello ad implementazione incrementale

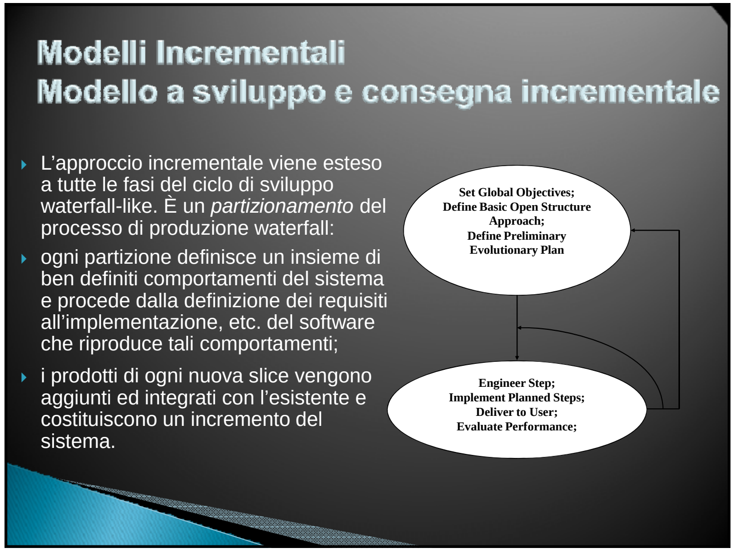
- Le fasi alte del Waterfall Modell sono completamente realizzate. Il Sw viene totalmente definito nei requisiti, specificato e progettato (solo HLD).
- I **sottosistemi** vengono implementati, testati, rilasciati, installati e messi in manutenzione secondo un piano di priorità in tempi diversi.
- Diventa fondamentale la fase di integrazione di nuovi sottosistemi con quelli già in esercizio

Definisci Progetto Stralcio

Code & Test

Integration & Test

Installazione & Manutenzione

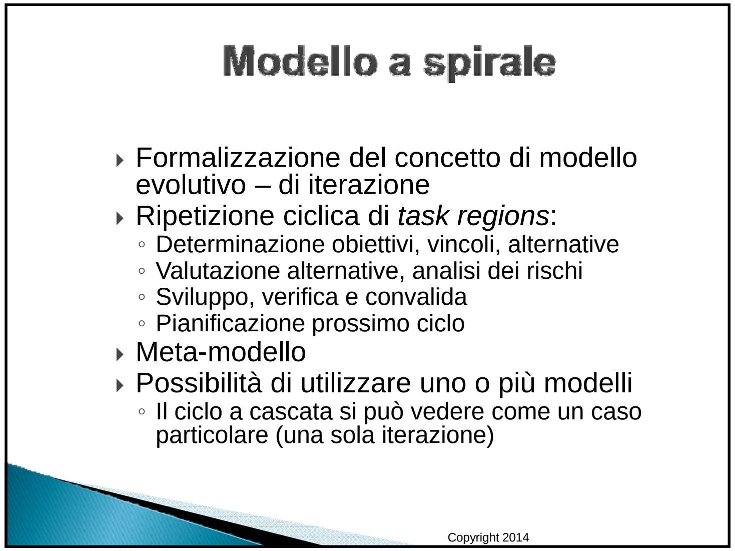


Modelli Incrementali
Modello a sviluppo e consegna incrementale

- L'approccio incrementale viene esteso a tutte le fasi del ciclo di sviluppo waterfall-like. È un *partizionamento* del processo di produzione waterfall:
- ogni partizione definisce un insieme di ben definiti comportamenti del sistema e procede dalla definizione dei requisiti all'implementazione, etc. del software che riproduce tali comportamenti;
- i prodotti di ogni nuova slice vengono aggiunti ed integrati con l'esistente e costituiscono un incremento del sistema.

Set Global Objectives; Define Basic Open Structure Approach; Define Preliminary Evolutionary Plan

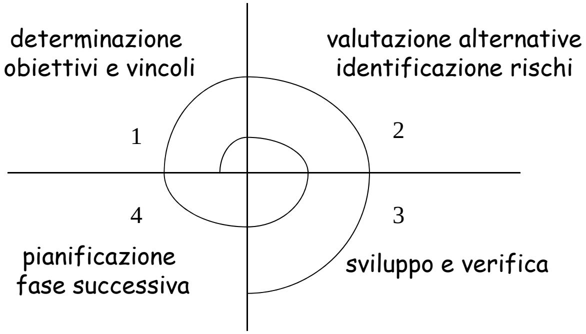
Engineer Step; Implement Planned Steps; Deliver to User; Evaluate Performance;



Modello a spirale

- Formalizzazione del concetto di modello evolutivo – di iterazione
- Ripetizione ciclica di *task regions*:
  - Determinazione obiettivi, vincoli, alternative
  - Valutazione alternative, analisi dei rischi
  - Sviluppo, verifica e convalida
  - Pianificazione prossimo ciclo
- Meta-modello
- Possibilità di utilizzare uno o più modelli
  - Il ciclo a cascata si può vedere come un caso particolare (una sola iterazione)

Modello a spirale di Boehm


determinazione obiettivi e vincoli
valutazione alternative identificazione rischi
1
2
4
3
pianificazione fase successiva
sviluppo e verifica

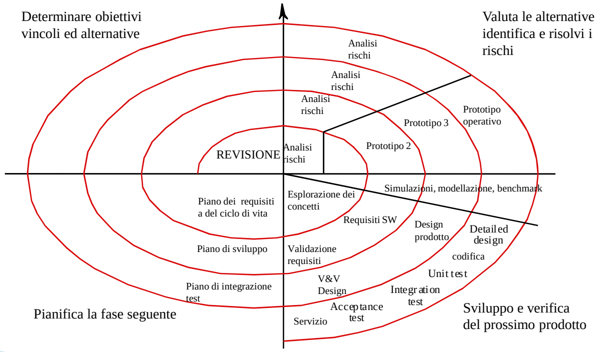
Gestione dei rischi

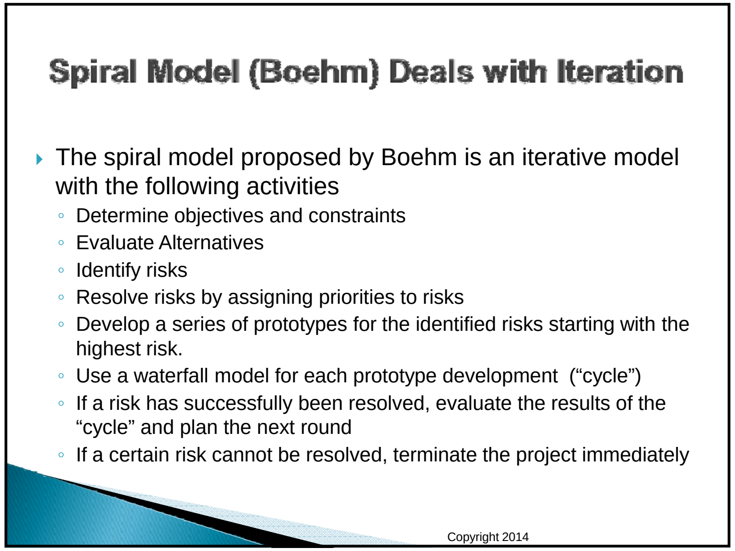
- Compito di chi gestisce (il manager) è minimizzare i rischi
- Tipologie di rischi: personale adeguato, scheduling, budget non realistico, sviluppo del sistema sbagliato …
- Il rischio è insito in tutte le attività umane ed è una misura dell’incertezza sul risultato dell’attività
- Alti rischi provocano ritardi e costi imprevisti
- Il rischio è collegato alla quantità e qualità delle informazioni disponibili: meno informazione si ha più alti sono i rischi

I modelli ed il rischio

- Cascata
  - alti rischi per sistemi nuovi, non familiari per problemi di specifica e progetto
  - Bassi rischi nello sviluppo di applicazioni familiari con tecnologie note
- Prototipazione
  - bassi rischi per le nuove applicazioni, specifica e sviluppo vanno di pari passo
  - alti rischi per la mancanza di un processo definito e visibile
- Trasformazionale
  - alti rischi per le tecnologie coinvolte e le professionalità richieste

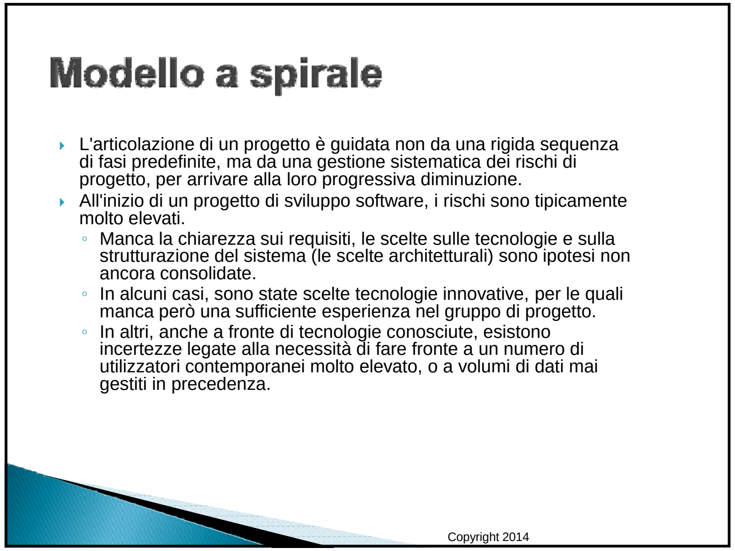
Il modello a spirale tradizionale


Determinare obiettivi vincoli ed alternative
Valuta le alternative identifica e risolvi i rischi
Analisi rischi
Analisi rischi
Analisi rischi
Prototipo 3
Prototipo operativo
REVISIONE
Analisi rischi
Prototipo 2
Simulazioni, modellazione, benchmark
Piano dei requisiti a del ciclo di vita
Esplorazione dei concetti
Requisiti SW
Design prodotto
Detailed design
Piano di sviluppo
Validazione requisiti
codifica
Unit test
V&V Design
Piano di integrazione test
Integration test
Acceptance test
Servizio
Pianifica la fase seguente
Sviluppo e verifica del prossimo prodotto



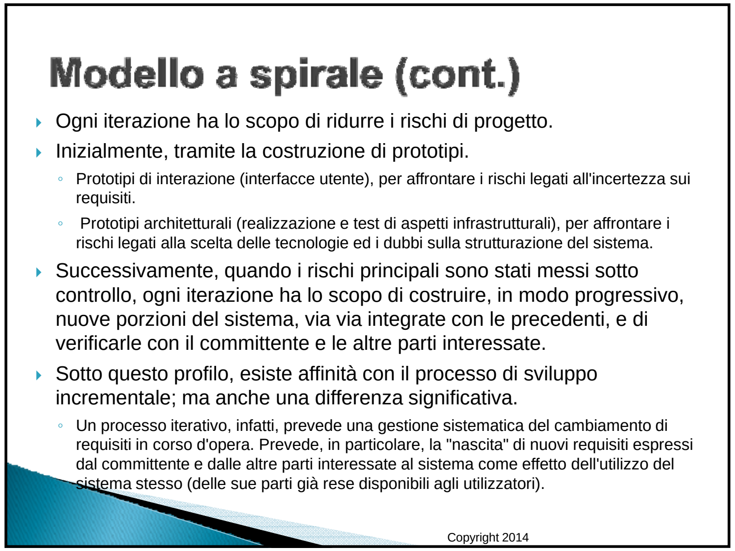
Spiral Model (Boehm) Deals with Iteration

- The spiral model proposed by Boehm is an iterative model with the following activities
  - Determine objectives and constraints
  - Evaluate Alternatives
  - Identify risks
  - Resolve risks by assigning priorities to risks
  - Develop a series of prototypes for the identified risks starting with the highest risk.
  - Use a waterfall model for each prototype development (“cycle”)
  - If a risk has successfully been resolved, evaluate the results of the “cycle” and plan the next round
  - If a certain risk cannot be resolved, terminate the project immediately



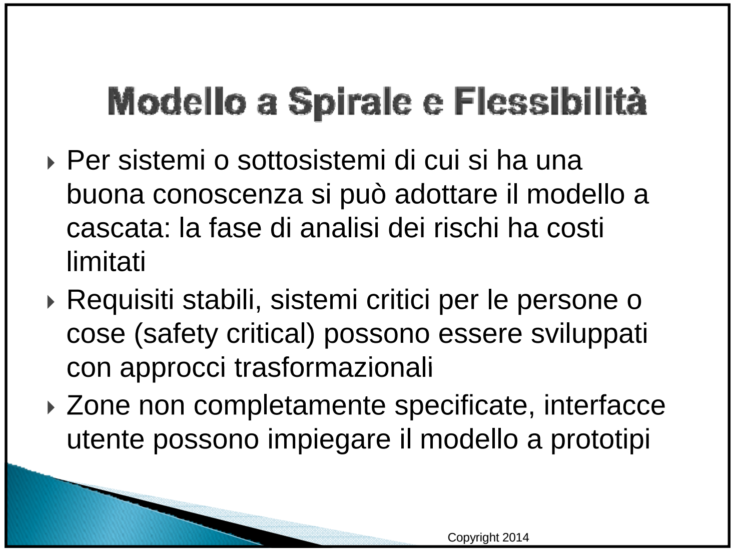
Modello a spirale

- L'articolazione di un progetto è guidata non da una rigida sequenza di fasi predefinite, ma da una gestione sistematica dei rischi di progetto, per arrivare alla loro progressiva diminuzione.
- All'inizio di un progetto di sviluppo software, i rischi sono tipicamente molto elevati.
  - Manca la chiarezza sui requisiti, le scelte sulle tecnologie e sulla strutturazione del sistema (le scelte architetturali) sono ipotesi non ancora consolidate.
  - In alcuni casi, sono state scelte tecnologie innovative, per le quali manca però una sufficiente esperienza nel gruppo di progetto.
  - In altri, anche a fronte di tecnologie conosciute, esistono incertezze legate alla necessità di fare fronte a un numero di utilizzatori contemporanei molto elevato, o a volumi di dati mai gestiti in precedenza.



Modello a spirale (cont.)

- Ogni iterazione ha lo scopo di ridurre i rischi di progetto.
- Inizialmente, tramite la costruzione di prototipi.
  - Prototipi di interazione (interfacce utente), per affrontare i rischi legati all'incertezza sui requisiti.
  - Prototipi architetturali (realizzazione e test di aspetti infrastrutturali), per affrontare i rischi legati alla scelta delle tecnologie ed i dubbi sulla strutturazione del sistema.
- Successivamente, quando i rischi principali sono stati messi sotto controllo, ogni iterazione ha lo scopo di costruire, in modo progressivo, nuove porzioni del sistema, via via integrate con le precedenti, e di verificarle con il committente e le altre parti interessate.
- Sotto questo profilo, esiste affinità con il processo di sviluppo incrementale; ma anche una differenza significativa.
  - Un processo iterativo, infatti, prevede una gestione sistematica del cambiamento di requisiti in corso d'opera. Prevede, in particolare, la "nascita" di nuovi requisiti espressi dal committente e dalle altre parti interessate al sistema come effetto dell'utilizzo del sistema stesso (delle sue parti già rese disponibili agli utilizzatori).



Modello a Spirale e Flessibilità

- Per sistemi o sottosistemi di cui si ha una buona conoscenza si può adottare il modello a cascata: la fase di analisi dei rischi ha costi limitati
- Requisiti stabili, sistemi critici per le persone o cose (safety critical) possono essere sviluppati con approcci trasformazionali
- Zone non completamente specificate, interfacce utente possono impiegare il modello a prototipi

Modello a spirale: vantaggi e svantaggi

- Vantaggi
  - Rende esplicita la gestione dei rischi
  - Aiuta a determinare errori nelle fasi iniziali
  - Obbliga a considerare gli aspetti di qualità
  - Integra sviluppo e manutenzione
- Svantaggi
  - Per contratto di solito si specifica a priori il modello di processo e i "deliverables". Lo sviluppo richiede un contratto nel contratto: vanno specificati vincoli, modello di processo, tempi di consegna e artefatti da consegnare
  - La pianificazione dei progetti condotti in modo iterativo è più complessa. Il piano di un processo iterativo evolve durante tutta la durata del progetto stesso, e richiede un controllo sistematico degli avanzamenti.
  - Un punto cruciale per il successo di un progetto iterativo è la collaborazione sistematica tra committenti (e altre parti interessate) e gruppo di progetto.
  - Richiede persone in grado di valutare i rischi
  - Per poter essere usato deve essere adattato alla realtà aziendale e/o al team

Agile Software Development

- Agile software development has become popular to describe new approaches that focus on close collaboration between programming teams and business experts
- Chapter 3 for more information on agile

The Importance of Project Phases and Management Reviews

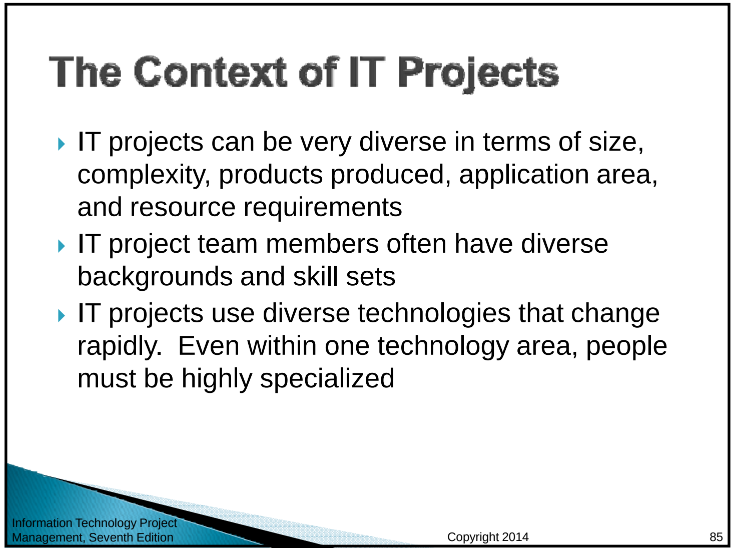
- A project should successfully pass through each of the project phases in order to continue on to the next
- Management reviews, also called **phase exits** or **kill points**, should occur after each phase to evaluate the project's progress, likely success, and continued compatibility with organizational goals

What Went Right?

"The real improvement that I saw was in our ability to—in the words of Thomas Edison—know when to stop beating a dead horse.…Edison's key to success was that he failed fairly often; but as he said, he could recognize a dead horse before it started to smell...In information technology we ride dead horses—failing projects—a long time before we give up. But what we are seeing now is that we are able to get off them; able to reduce cost overrun and time overrun. That's where the major impact came on the success rate."*

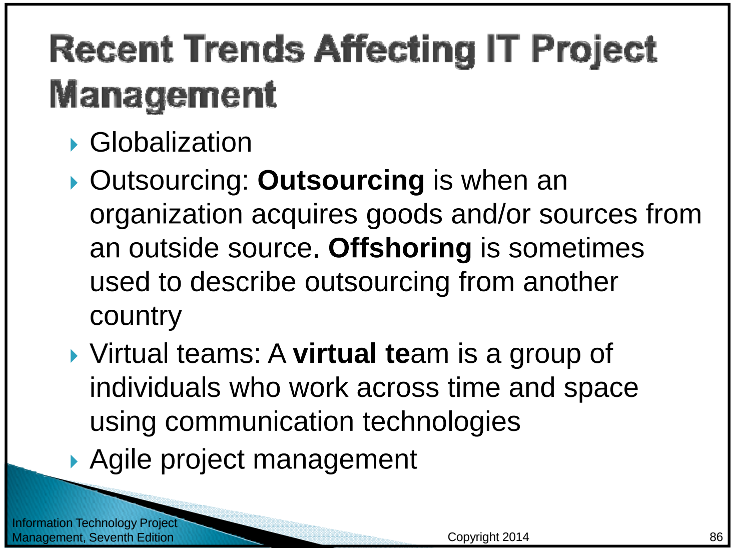
Many organizations, like Huntington Bancshares, Inc., use an **executive steering committee** to help keep projects on track.

*Cabanis, Jeannette, "'A Major Impact': The Standish Group's Jim Johnson On Project Management and IT Project Success," PM Network, PMI, Sep.1998, p. 7



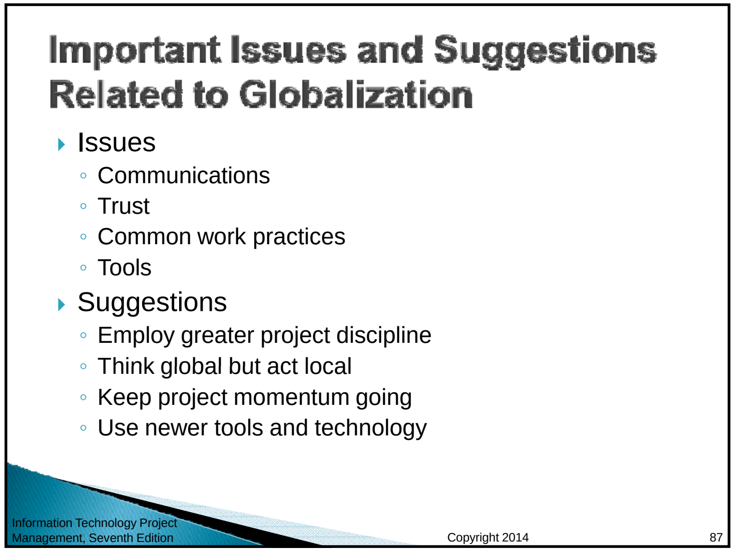
The Context of IT Projects

- IT projects can be very diverse in terms of size, complexity, products produced, application area, and resource requirements
- IT project team members often have diverse backgrounds and skill sets
- IT projects use diverse technologies that change rapidly. Even within one technology area, people must be highly specialized



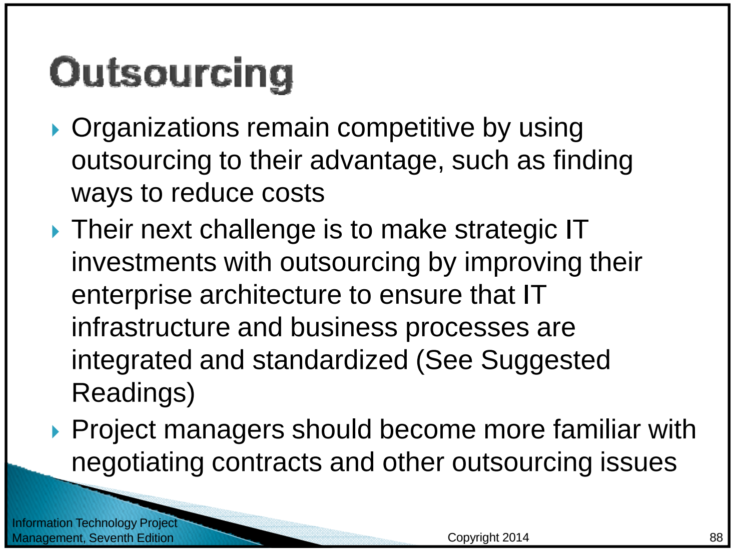
Recent Trends Affecting IT Project Management

- Globalization
- Outsourcing: **Outsourcing** is when an organization acquires goods and/or sources from an outside source. **Offshoring** is sometimes used to describe outsourcing from another country
- Virtual teams: A **virtual te**am is a group of individuals who work across time and space using communication technologies
- Agile project management



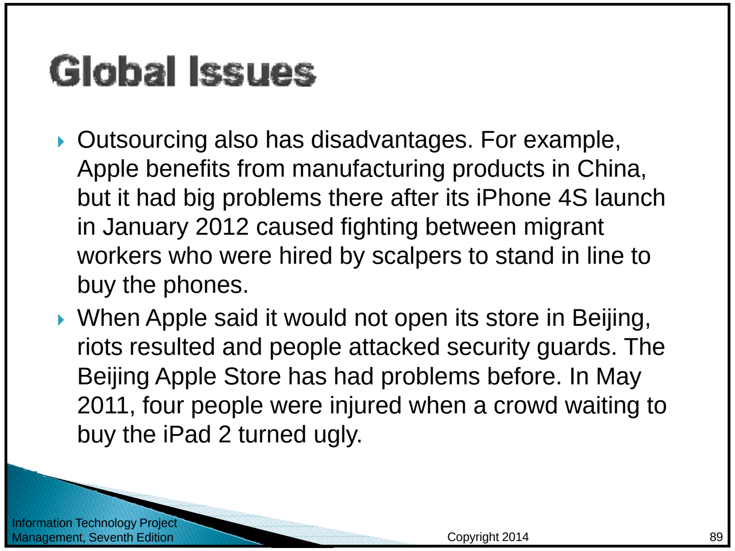
Important Issues and Suggestions Related to Globalization

- Issues
  - Communications
  - Trust
  - Common work practices
  - Tools
- Suggestions
  - Employ greater project discipline
  - Think global but act local
  - Keep project momentum going
  - Use newer tools and technology



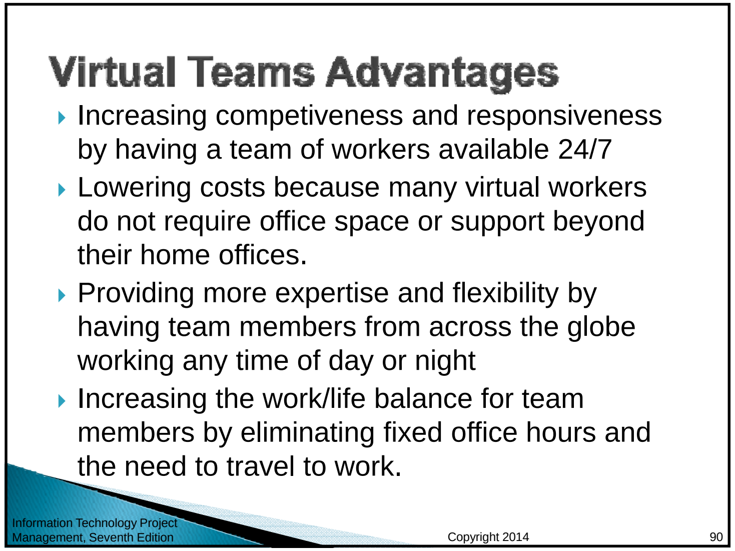
Outsourcing

- Organizations remain competitive by using outsourcing to their advantage, such as finding ways to reduce costs
- Their next challenge is to make strategic IT investments with outsourcing by improving their enterprise architecture to ensure that IT infrastructure and business processes are integrated and standardized (See Suggested Readings)
- Project managers should become more familiar with negotiating contracts and other outsourcing issues



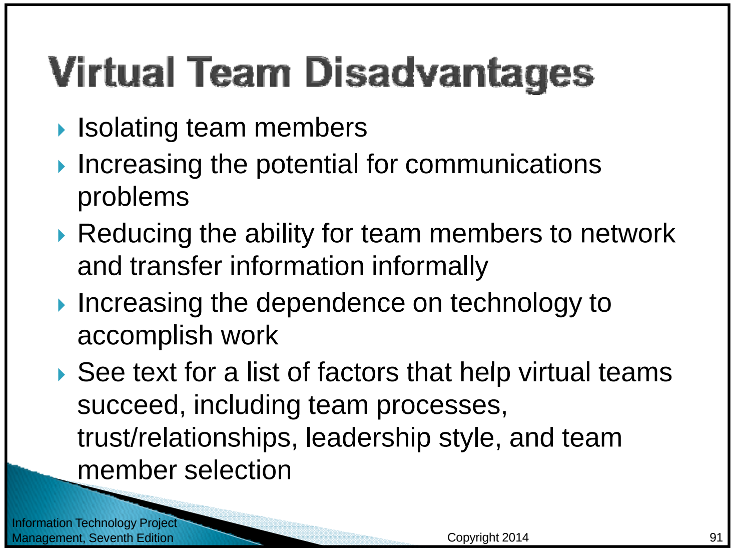
Global Issues

- Outsourcing also has disadvantages. For example, Apple benefits from manufacturing products in China, but it had big problems there after its iPhone 4S launch in January 2012 caused fighting between migrant workers who were hired by scalpers to stand in line to buy the phones.
- When Apple said it would not open its store in Beijing, riots resulted and people attacked security guards. The Beijing Apple Store has had problems before. In May 2011, four people were injured when a crowd waiting to buy the iPad 2 turned ugly.



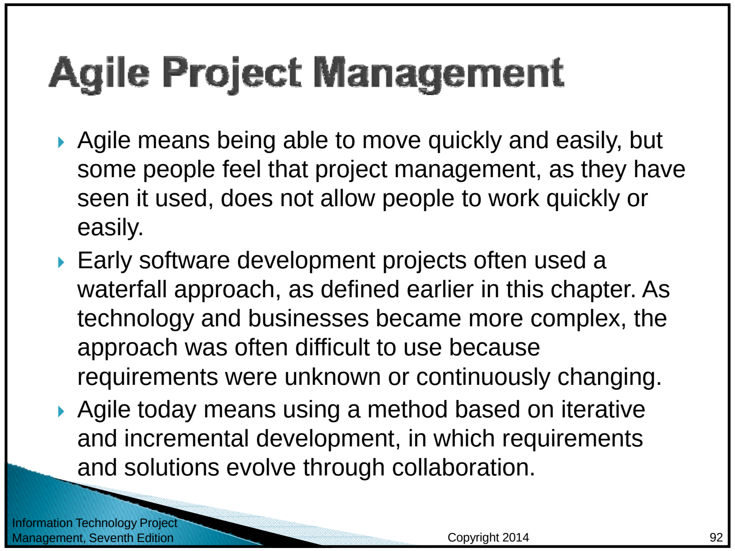
Virtual Teams Advantages

- Increasing competiveness and responsiveness by having a team of workers available 24/7
- Lowering costs because many virtual workers do not require office space or support beyond their home offices.
- Providing more expertise and flexibility by having team members from across the globe working any time of day or night
- Increasing the work/life balance for team members by eliminating fixed office hours and the need to travel to work.



Virtual Team Disadvantages

- Isolating team members
- Increasing the potential for communications problems
- Reducing the ability for team members to network and transfer information informally
- Increasing the dependence on technology to accomplish work
- See text for a list of factors that help virtual teams succeed, including team processes, trust/relationships, leadership style, and team member selection



Agile Project Management

- Agile means being able to move quickly and easily, but some people feel that project management, as they have seen it used, does not allow people to work quickly or easily.
- Early software development projects often used a waterfall approach, as defined earlier in this chapter. As technology and businesses became more complex, the approach was often difficult to use because requirements were unknown or continuously changing.
- Agile today means using a method based on iterative and incremental development, in which requirements and solutions evolve through collaboration.

Agile Makes Sense for Some Projects, But Not All

- Many seasoned experts in project management warn people not to fall for the hype associated with Agile.
- For example, J. Leroy Ward, Executive Vice President at ESI International, said that “Agile will be seen for what it is … and isn’t….Project management organizations embracing Agile software and product development approaches will continue to grow while being faced with the challenge of demonstrating ROI through Agile adoption.”*

*J. Leroy Ward, “The Top Ten Project Management Trends for 2011,” projecttimes.com (January 24, 2011).

Manifesto for Agile Software Development

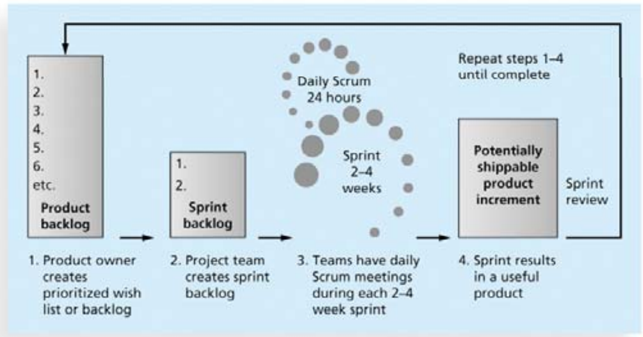
- In February 2001, a group of 17 people that called itself the Agile Alliance developed and agreed on the Manifesto for Agile Software Development, as follows:
- “We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:
- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan”*

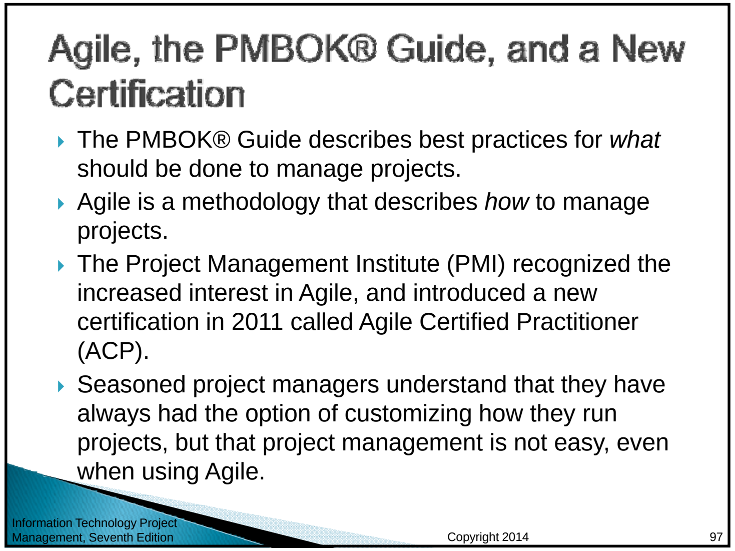
*Agile Manifesto, www.agilemanifesto.org.

Scrum

- According to the Scrum Alliance, Scrum is the leading agile development method for completing projects with a complex, innovative scope of work.
- The term was coined in 1986 in a Harvard Business Review study that compared high-performing, cross-functional teams to the scrum formation used by rugby teams.

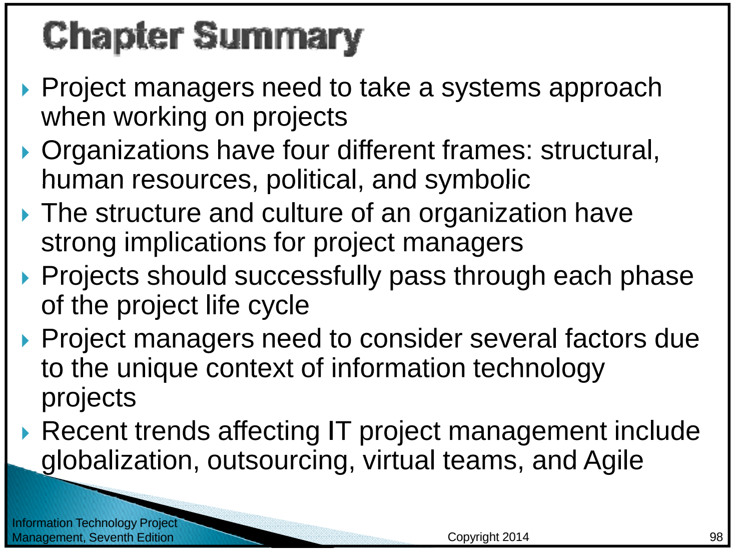
Figure 2-6. Scrum Framework


Repeat steps 1–4 until complete
1.
2.
3.
4.
5.
6.
etc.
Product backlog
1.
2.
Sprint backlog
Daily Scrum 24 hours
Sprint 2–4 weeks
Potentially shippable product increment
Sprint review
1. Product owner creates prioritized wish list or backlog
2. Project team creates sprint backlog
3. Teams have daily Scrum meetings during each 2–4 week sprint
4. Sprint results in a useful product



Agile, the PMBOK® Guide, and a New Certification

- The PMBOK® Guide describes best practices for *what* should be done to manage projects.
- Agile is a methodology that describes *how* to manage projects.
- The Project Management Institute (PMI) recognized the increased interest in Agile, and introduced a new certification in 2011 called Agile Certified Practitioner (ACP).
- Seasoned project managers understand that they have always had the option of customizing how they run projects, but that project management is not easy, even when using Agile.



Chapter Summary

- Project managers need to take a systems approach when working on projects
- Organizations have four different frames: structural, human resources, political, and symbolic
- The structure and culture of an organization have strong implications for project managers
- Projects should successfully pass through each phase of the project life cycle
- Project managers need to consider several factors due to the unique context of information technology projects
- Recent trends affecting IT project management include globalization, outsourcing, virtual teams, and Agile