

3

Information Systems Development

Overview

Chapter 3 provides a comprehensive introduction to information system development. The chapter's intent is to introduce principles and processes used to develop information systems. The information system development process reinforces the information system building blocks that were introduced in Chapter 2. A hypothetical methodology called *FAST* (*Framework for the Application of System Techniques*) is used to teach a representative, implementation of a system development process. This chapter provides a “phase-level” overview. Later chapters provide “activity-level” descriptions and coverage of tools and techniques.

The methodology as presented supports all contemporary paradigms including structured methods, information engineering, and rapid application development, as well as emerging object-oriented methods. As the readers progress through the book, they will learn how to “plug and play” various tools and techniques into the methodological framework provided by FAST.

Chapter to Course Sequencing

This chapter should follow Chapter 2 in all but exceptional course scenarios. Chapter 2 provides the fundamental building blocks of information systems, and Chapter 3 immediately reinforces those building blocks with a system development methodology.

What's Different Here and Why?

The following changes have been made to the seventh edition of the information system development chapter:

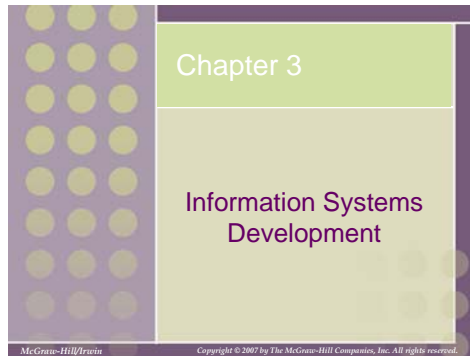
1. As with all chapters, we have streamlined the SoundStage episode into a quick narrative introduction to the concepts presented the chapter. We believe this streamlined version will be more readable and thus more useful.
2. The relationship between the classic SDLC phases and the FAST phases has been clarified.
3. The discussion of waterfall (sequential) vs. iterative development has been expanded and moved into this chapter to better focus this chapter on methodology.

4. Key terms have been added to this chapter for logical model and physical model. These concepts will also be discussed later, but it is useful to have them here to make sure students see the big picture.

Lesson Planning Notes for Slides

The following instructor notes, keyed to slide images from the PowerPoint repository, are intended to help instructors integrate the slides into their individual lesson plans for this chapter.

Slide 1



slide appearance after initial mouse click
in slide show mode

This repository of slides is intended to support the named chapter. The slide repository should be used as follows:

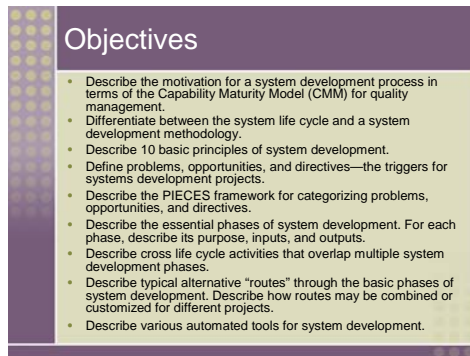
Copy the file to a unique name for your course and unit.

Edit the file by deleting those slides you don't want to cover, editing other slides as appropriate to your course, and adding slides as desired.

Print the slides to produce transparency masters or print directly to film or present the slides using a computer image projector.

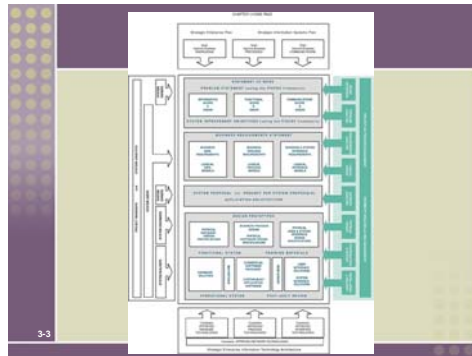
Most slides include instructor notes. In recent versions of PowerPoint, notes by default display in a window under the slide. The instructor notes are also reprinted below.

Slide 2



No additional notes

Slide 3

**Teaching Notes**

This slide uses the framework taught in chapter 2 and emphasizes how the 8 FAST methodology phases relate to it. FAST is a representative methodology used throughout the remainder of the textbook. Each chapter will focus on one or more aspects of the framework.

Slide 4

Process of System Development

System development process – a set of activities, methods, best practices, deliverables, and automated tools that stakeholders (Chapter 1) use to develop and continuously improve information systems and software (Chapters 1 and 2).

- Many variations
- Using a consistent process for system development:
 - Create efficiencies that allow management to shift resources between projects
 - Produces consistent documentation that reduces lifetime costs to maintain the systems
 - Promotes quality

Teaching Notes

In teaching this course we sometimes begin by asking students to imagine they have to develop some complex information system, such as a new course registration system for the university. Then we ask them how they would proceed to determine what needs to be included in this information system. After a little discussion we ask them to compare this hypothetical process to what they have done with assignments in programming classes. The point is that programming assignments are not at all like real world information systems, which need a much more detailed development process. That sells why students need this course.

Some textbooks use the term systems development life cycle. We elected not to use that term because it invokes negative connotation for many instructors. Some associate it with a pure waterfall development approach (which we consider unfair).

We differentiate between development and operation (sometimes called production). This chapter focuses on development and different methodologies and strategies employed for system development.

Slide 5

CMM Process Management Model

Capability Maturity Model (CMM) – a standardized framework for assessing the maturity level of an organization's information system development and management processes and products. It consists of five levels of maturity:

- **Level 1—Initial:** System development projects follow no prescribed process.
- **Level 2—Repeatable:** Project management processes and practices established to track project costs, schedules, and functionality.
- **Level 3—Defined:** Standard system development process (methodology) is purchased or developed. All projects use a version of this process.
- **Level 4—Managed:** Measurable goals for quality and productivity are established.
- **Level 5—Optimizing:** The standardized system development process is continuously monitored and improved based on measures and data analysis established in Level 4.

3-5

Teaching Notes

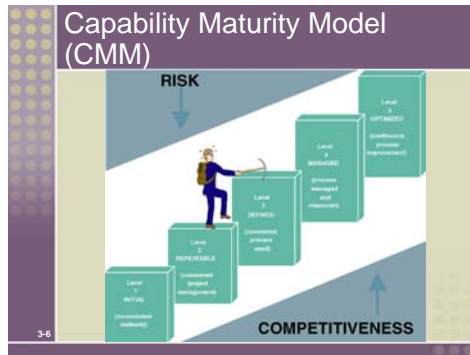
We feel that CMM is very important. CMM is the information technology response to the total quality management initiative. CMM breathed new life into the importance of a system development process. The term "process" in CMM is equivalent to the term "methodology" as popularized in systems analysis and design methods.

Recognize that each level is a prerequisite for the next level.

Most organizations pursuing the CMM are targeting Level 3, that is, consistently using a standardized process or methodology to develop all systems.

CMM Level 2 deals with project management. CMM Level 3 deals with what has come to be known as process management.

Slide 6

**Teaching Notes**

This diagram makes clear that you reach any given CMM level only by first climbing through the other levels.

Slide 7

Impact of System Development "Process" on Quality

CMM Project Statistics for a Project Resulting in 200,000 Lines of Code

Organization's CMM Level	Project Duration (months)	Project Person-Months	Number of Defects Shipped	Median Cost (\$ millions)	Lowest Cost (\$ millions)	Highest Cost (\$ millions)
1	30	600	61	5.5	1.8	100+
2	18.5	143	12	1.3	.96	1.7
3	15	80	7	.728	.518	.933

3-7

Teaching Notes

This table dramatically illustrates the value of following a methodology. Note the drop in project duration, person-months, defects, and costs as a result of following a consistent methodology

Slide 8

Life Cycle versus Methodology

- **System life cycle** – the factoring of the lifetime of an information system into two stages, (1) systems development and (2) systems operation and maintenance.
- **System development methodology** – a formalized approach to the systems development process; a standardized development process that defines (as in CMM Level 3) a set of activities, methods, best practices, deliverables, and automated tools that system developers and project managers are to use to develop and continuously improve information systems and software.

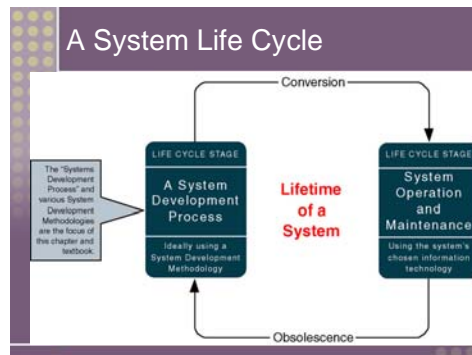
3-8

Teaching Notes

Point out that these terms are not synonymous. A system life cycle just happens. A system development methodology is planned and purposely carried out during the development stage of the system life cycle.

A common synonym for system development methodology is *system development process*.

Slide 9

**Teaching Notes**

This slide formally differentiates between the life cycle and a systems development methodology that is used to execute the development stage of the life cycle.

A common synonym for "system operation" is "production."

Slide 10

Representative System Development Methodologies

- Architected Rapid Application Development (Architected RAD)
- Dynamic Systems Development Methodology (DSDM)
- Joint Application Development (JAD)
- Information Engineering (IE)
- Rapid Application Development (RAD)
- Rational Unified Process (RUP)
- Structured Analysis and Design
- eXtreme Programming (XP)

3-10

Teaching Notes

Structured Analysis and Design is old but still occasionally encountered

Note that many other commercial methodologies and software tools exist based on these general methodologies

Slide 11

Principles of System Development

- Get the system users involved.
- Use a problem-solving approach.
- Establish phases and activities.
- Document through development.
- Establish standards.
- Manage the process and projects
- Justify systems as capital investments.
- Don't be afraid to cancel or revise scope.
- Divide and conquer.
- Design systems for growth and change.

3-11

Teaching Notes

Note that these principles apply regardless of which methodology is used.

Slide 12

Use a Problem-Solving Approach

Classical Problem-solving approach

1. Study and understand the problem, its context, and its impact.
2. Define the requirements that must be met by any solution.
3. Identify candidate solutions that fulfill the requirements, and select the "best" solution.
4. Design and/or implement the chosen solution.
5. Observe and evaluate the solution's impact, and refine the solution accordingly.

3-12

Teaching Notes

It can be useful to walk through a non-IS example of problem-solving, such as diagnosing a car problem or going to the restaurant because you are hungry.

Slide 13

Establish Phases and Activities

Overlap of System Development Phases

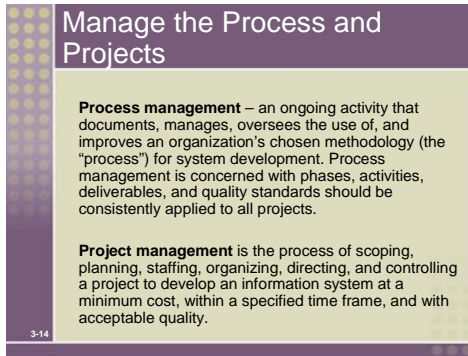
3-13

Teaching Notes

Every project is different depending on the size, complexity, and development methodology or route. The key point to emphasize in the figure is that the phases occur in parallel. It is important that students not misinterpret that the phases in this chapter are sequential.

Note that project and process management are illustrated as ongoing activities that last the duration of a project. (The next chapter will focus on project and process management as well as the construction and use of Gantt Charts such as the one above.)

Slide 14



Manage the Process and Projects

Process management – an ongoing activity that documents, manages, oversees the use of, and improves an organization's chosen methodology (the "process") for system development. Process management is concerned with phases, activities, deliverables, and quality standards should be consistently applied to all projects.

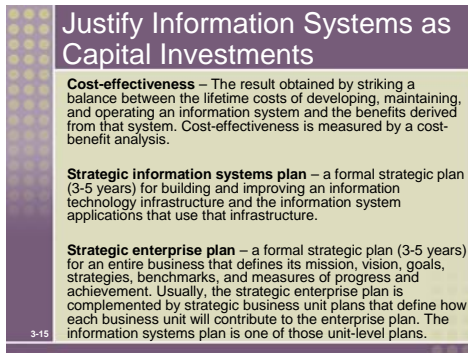
Project management is the process of scoping, planning, staffing, organizing, directing, and controlling a project to develop an information system at a minimum cost, within a specified time frame, and with acceptable quality.

3-14

Teaching Notes

Most organizations pursuing the CMM are targeting Level 3, that is, consistently using a standardized process or methodology to develop all systems. CMM Level 2 deals with project management. CMM Level 3 deals with what has come to be known as process management.

Slide 15



Justify Information Systems as Capital Investments

Cost-effectiveness – The result obtained by striking a balance between the lifetime costs of developing, maintaining, and operating an information system and the benefits derived from that system. Cost-effectiveness is measured by a cost-benefit analysis.

Strategic information systems plan – a formal strategic plan (3-5 years) for building and improving an information technology infrastructure and the information system applications that use that infrastructure.

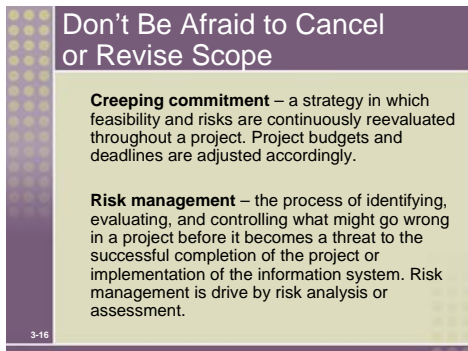
Strategic enterprise plan – a formal strategic plan (3-5 years) for an entire business that defines its mission, vision, goals, strategies, benchmarks, and measures of progress and achievement. Usually, the strategic enterprise plan is complemented by strategic business unit plans that define how each business unit will contribute to the enterprise plan. The information systems plan is one of those unit-level plans.

3-15

Teaching Notes

The cost justification of system development projects is an ongoing activity. At any time during the process it could be discovered that the project no longer makes economic sense.

Slide 16



Don't Be Afraid to Cancel or Revise Scope

Creeping commitment – a strategy in which feasibility and risks are continuously reevaluated throughout a project. Project budgets and deadlines are adjusted accordingly.

Risk management – the process of identifying, evaluating, and controlling what might go wrong in a project before it becomes a threat to the successful completion of the project or implementation of the information system. Risk management is driven by risk analysis or assessment.

3-16

No additional notes

Slide 17

Where Do Systems Development Projects Come From?

- **Problem** – an undesirable situation that prevents the organization from fully achieving its purpose, goals, and/or objectives.
- **Opportunity** – a chance to improve the organization even in the absence of an identified problem.
- **Directive** – a new requirement that is imposed by management, government, or some external influence.

3-17

Teaching Notes

Emphasize that problems, opportunities, and directives can either be planned or unplanned.

Slide 18

Where Do Systems Development Projects Come From?

- **Planned Projects**
 - An **information systems strategy plan** has examined the business as a whole to identify those system development projects that will return the greatest strategic (long-term) value to the business
 - A **business process redesign** has thoroughly analyzed a series of business processes to eliminate redundancy and bureaucracy and to improve efficiency and value added. Not it is time to redesign the supporting information system for those redesigned business processes.

3-18

No additional notes.

Slide 19

Where Do Systems Development Projects Come From?

- **Unplanned projects**
 - Triggered by a specific problem, opportunity, or directive that occurs in the course of doing business.
 - **Steering committee** – an administrative body of system owners and information technology executives that prioritizes and approves candidate system development projects.
 - **Backlog** – a repository of project proposals that cannot be funded or staffed because they are a lower priority than those that have been approved for system development.

3-19

No additional notes.

Slide 20

The PIECES Problem-Solving Framework

- P** the need to improve performance
- I** the need to improve information (and data)
- E** the need to improve economics, control costs, or increase profits
- C** the need to improve control or security
- E** the need to improve efficiency of people and processes
- S** the need to improve service to customers, suppliers, partners, employees, etc.

Teaching Notes

We really emphasize PIECES as a useful way to characterize all problems. Later, we teach our students to use PIECES to analyze requirements and solutions as well.

It can be useful to apply the PIECES framework to a project from the instructor's professional background.

Refer to the figure in the text for the PIECES checklist.

Slide 21

Project Phases

- FAST** - (Framework for the Application of Systems Thinking) a hypothetical methodology used throughout this book to demonstrate a representative systems development process.
- Each methodology will use different project phases.

FAST Phases	Classic Phases (from Chapter 1)			
	Project Initiation	System Analysis	System Design	System Implementation
Scope Definition	X			
Problem Analysis	X	X		
Requirements Analysis		X		
Logical Design		X		
Decision Analysis		(a system analysis transition phase)		
Physical Design and Integration			X	
Construction and Testing			X	X
Installation and Delivery				X

Conversion Notes

This is a new slide in the seventh edition

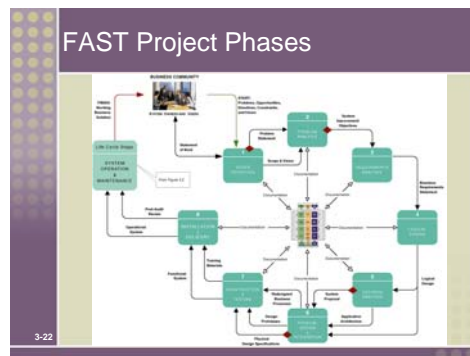
Teaching Notes

The classic SDLC phases were presented in chapter 1. Now we present the FAST phases. This switch often confuses students. FAST is taught for two reasons:

1. The classic SDLC phases are really two general for teaching.
2. FAST allows students to see a methodology in action.

It is important for students to see that FAST is a methodology and how it fits into the classic SDLC phases.

Slide 22

**Teaching Notes**

This is not meant to be interpreted as a "waterfall" model.

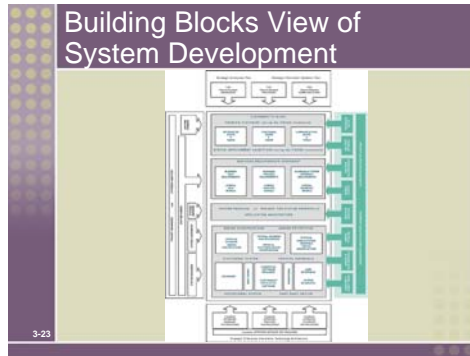
Make sure that the students recognize that this slide represents one methodology that the authors call *FAST*. There are many methodologies and variations on this slide.

Note that documentation goes on at every phase. These are the "deliverables" of the project.

The diamonds reflect typical "go" or "no go" feasibility checkpoints consistent with the creeping commitment philosophy.

At the center of the figure is the Building Blocks View of System Development, shown on the next slide.

Slide 23

**Teaching Notes**

This figure details all the deliverables of the *FAST* methodology throughout the project as well as the interaction with the project of all system users and the timing of each *FAST* phase. Note also the role of the strategic enterprise plan, strategic information systems plan, and strategic enterprise information technology architecture. You can use this figure to walk students through the syllabus, showing where each chapter and each case study deliverable fits in.

Slide 24

Scope Definition Phase

Problem statement – a statement and categorization of problems, opportunities, and directives; may also include constraints and an initial vision for the solution. Synonyms include *preliminary study* and *feasibility assessment*.

Constraint – any factor, limitation, or restraint that may limit a solution or the problem-solving process.

Scope creep – a common phenomenon wherein the requirements and expectations of a project increase, often without regard to the impact on budget and schedule.

Statement of work – a contract with management and the user community to develop or enhance an information system; defines vision, scope, constraints, high-level user requirements, schedule, and budget. Synonyms include *project charter*, *project plan*, and *service-level agreement*.

3-24

Teaching Notes

If any of these will be a later assignment, it can be helpful to make note of it.

Slide 25

Requirements Analysis Phase

- What capabilities should the new system provide for its users?
- What data must be captured and stored?
- What performance level is expected?
- What are the priorities of the various requirements?

3-25

Conversion Notes

This slide is new in the seventh edition.

Slide 26

Logical Design Phase

Logical design – the translation of business user requirements into a system model that depicts only the business requirements and not any possible technical design or implementation of those requirements. Common synonyms include *conceptual design* and *essential design*.

System model – a picture of a system that represents reality or a desired reality. System models facilitate improved communication between system users, system analysts, system designers, and system builders.

Analysis paralysis – a satirical term coined to describe a common project condition in which excessive system modeling dramatically slows progress toward implementation of the intended system solution.

3-26

Teaching Notes

After finishing this chapter, students are liable to think that this course is teaching "analysis paralysis," since their prior experience may consist of short programming assignments. But without the right amount of analysis you will never deliver what the users want and need.

Slide 27

Decision Analysis Phase

- Candidate solutions evaluated in terms of:
 - Technical feasibility** – Is the solution technically practical? Does our staff have the technical expertise to design and build this solution?
 - Operational feasibility** – Will the solution fulfill the users' requirements? To what degree? How will the solution change the users' work environment? How do users feel about such a solution?
 - Economic feasibility** – Is the solution cost-effective?
 - Schedule feasibility** – Can the solution be designed and implemented within an acceptable time?
 - Risk feasibility** – What is the probability of a successful implementation using the technology and approach?

3-27

Teaching Notes

These will be covered in detail in Chapter 11.

Slide 28

Physical Design & Integration Phase

Physical design – the translation of business user requirements into a system model that depicts a technical implementation of the users' business requirements. Common synonyms include *technical design* or *implementation model*.

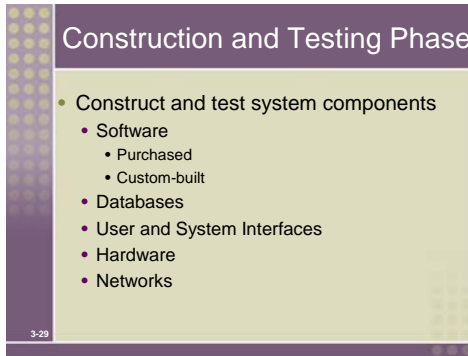
Two extreme philosophies of physical design

- Design by specification** – physical system models and detailed specification are produced as a series of written (or computer-generated) blueprints for construction.
- Design by prototyping** – Incomplete but functioning applications or subsystems (called *prototypes*) are constructed and refined based on feedback from users and other designers.

3-28

No additional notes.

Slide 29

The slide has a purple header with the title "Construction and Testing Phase" in white. The main content area is light green with a list of bullet points. The slide number "3-29" is in the bottom left corner.

Construction and Testing Phase

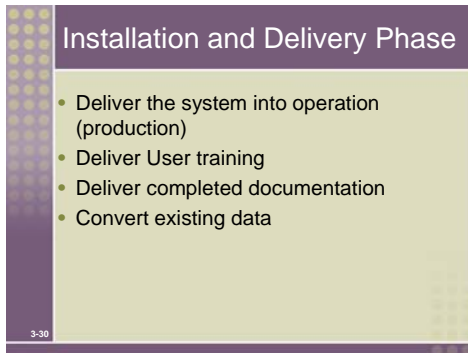
- Construct and test system components
 - Software
 - Purchased
 - Custom-built
 - Databases
 - User and System Interfaces
 - Hardware
 - Networks

3-29

Conversion Notes

This slide is new in the seventh edition.

Slide 30

The slide has a purple header with the title "Installation and Delivery Phase" in white. The main content area is light green with a list of bullet points. The slide number "3-30" is in the bottom left corner.

Installation and Delivery Phase

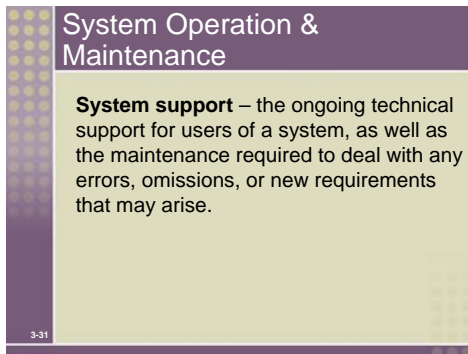
- Deliver the system into operation (production)
- Deliver User training
- Deliver completed documentation
- Convert existing data

3-30

Conversion Notes

This slide is new in the seventh edition.

Slide 31

The slide has a purple header with the title "System Operation & Maintenance" in white. The main content area is light green with a paragraph of text. The slide number "3-31" is in the bottom left corner.

System Operation & Maintenance

System support – the ongoing technical support for users of a system, as well as the maintenance required to deal with any errors, omissions, or new requirements that may arise.

3-31

No additional notes.

Slide 32

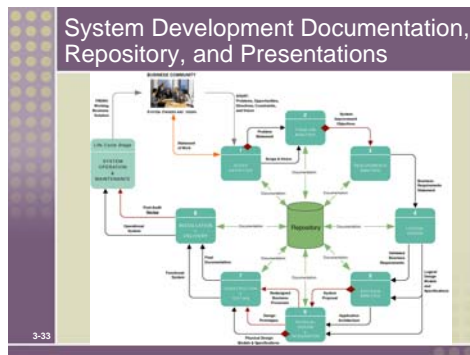
Cross Life-Cycle Activities

Cross life-cycle activity – activities that overlap multiple phases

- **Fact-finding** - formal process of using research, interviews, meetings, questionnaires, sampling, and other techniques to collect information about system problems, requirements, and preferences.
- **Documentation and presentation**
 - **Documentation** – recording facts and specifications for a systems for current and future reference.
 - **Presentation** – communicating findings, recommendations, and documentation for review by interested users and managers.
- **Repository** – database and/or file directory where system developers store all documentation, knowledge, and artifacts for information systems or project(s).
- **Feasibility analysis**
- **Process and project management**

No additional notes.

Slide 33

**Teaching Notes**

Some instructors prefer the term “dictionary” or “encyclopedia” depending on what their CASE tool calls its repository. The key point here is that system development documentation (“knowledge”) is shared via a repository as well as flowing between phases and people.

Slide 34

Sequential versus Iterative Development

Waterfall development approach an approach to systems analysis and design that completes each phase one after another and only once.

Iterative development approach an approach to systems analysis and design that completes the entire information system in successive iterations. Each iteration does some analysis, some design, and some construction. Synonyms include incremental and spiral.

Conversion Notes

In the sixth edition this slide appeared in chapter 1. These concepts were moved here to better focus this chapter on methodologies.

Teaching Notes

Sequential processes are one alternative. This is often called a “waterfall development” process. An iterative or incremental development process develops and placed into operation a portion of the new system as quickly as possible and then moves on to other parts of the system. Consider asking students to come up with pros and cons of each approach. For instance, the iterative approach gives the users some of what they want much sooner than the sequential approach. But the sequential approach is less likely to require fixes and redesigns to what has previously been implemented.

Slide 35

**Teaching Notes**

There can be multiple strategies or “routes” through the traditional phases. Thus, “one size does not fit all projects.”

We have included a few of the more common routes, but there are literally dozens of routes and hundreds of variations in many methodologies. A key precept is that, contrary to popular marketing and consulting hype, the routes are merely different implementations of the same basic phases already covered (usually cleverly disguised in proprietary languages and terminology). Different routes emphasize different phases, tools, and techniques.

Slide 36

Model-Driven Development Strategy

- **Model-driven development** – a system development strategy that emphasizes the drawing of system models to help visualize and analyze problems, define business requirements, and design information systems.
 - **Process modeling** – a process-centered technique popularized by the structured analysis and design methodology that used models of business process requirements to derive effective software designs for a system.
 - **Data modeling** – a data-centered technique used to model business data requirements and design database systems that fulfill those requirements.
 - **Object modeling** – a technique that attempts to merge the data and process concerns into singular constructs called objects. Object models are diagrams that document a system in terms of its objects and their interactions.

3-36

Teaching Notes

The model-driven “route” is most typically associated with “methodologies based on structured analysis and design, information engineering (data modeling), or object-oriented analysis and design (use-case, UML, etc.).

It was not the intent to teach the techniques in this chapter. That is why we elected not to include sample models that the students would not truly understand until they read the modeling chapters themselves.

Slide 37

Logical vs. Physical Models

Logical model - a pictorial representation that depicts what a system is or does.

Physical model - a technical pictorial representation that depicts what a system is or does and how the system is implemented.

3-37

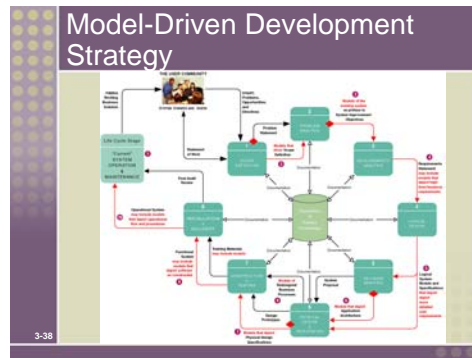
Conversion Notes

This slide is new to the seventh edition.

Teaching Notes

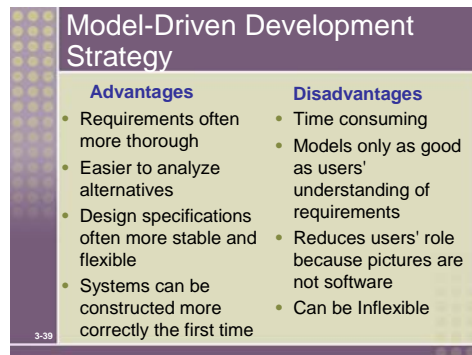
We always emphasize to students that logical models are built in the analysis phases while physical models are built in the design phases after the implementation architecture is selected.

Slide 38

**Teaching Notes**

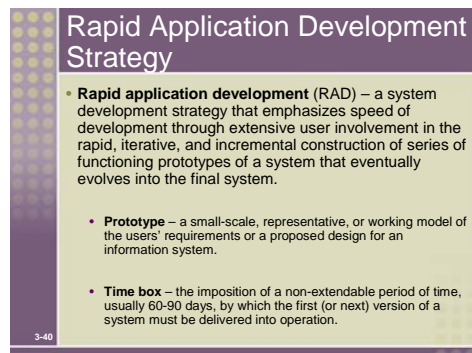
The model-driven approach (with the notable exception of OOAD) is most commonly associated with the “waterfall” approach to system development. While often criticized for its time and effort intensity, model-driven strategies still work well with large and unstructured problem domains.

Slide 39

**Conversion Notes**

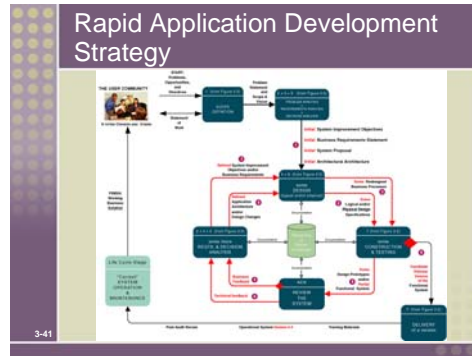
This slide is new to the seventh edition.

Slide 40

**Teaching Notes**

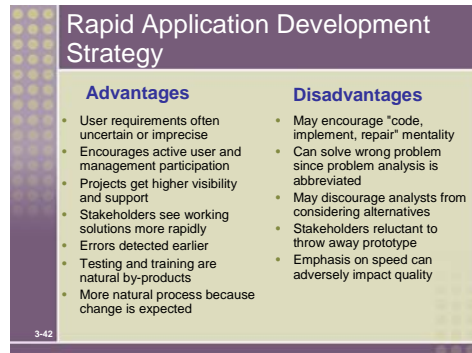
The rapid application development “route” is most typically associated with prototyping, JAD, and incremental or iterative approaches to system development.

Slide 41

**Teaching Notes**

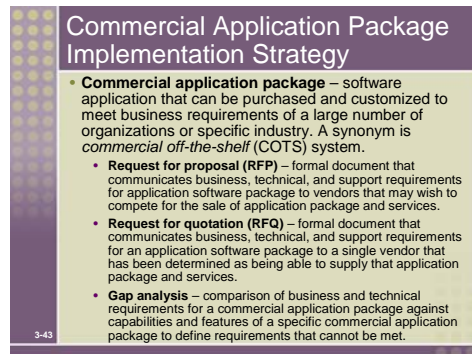
The rapid application development approach is most commonly associated with an incremental or iterative approach to system development. It is very popular for smaller and relatively structured projects in which requirements are fairly well understood from the beginning of the project.

Slide 42

**Conversion Notes**

This slide is new to the seventh edition.

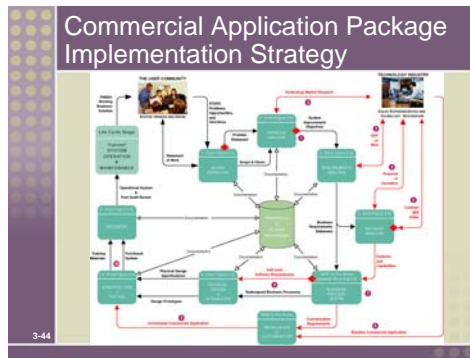
Slide 43

**Teaching Notes**

COTS has become extraordinarily important to aspiring systems analysts because an ever-increasing percentage of all information systems are purchased, not built in-house.

Emphasize to students that tomorrow's systems analysts will be as likely to participate in a software package selection and integration as they will in a traditional design-and-construction style project.

Slide 44

**Teaching Notes**

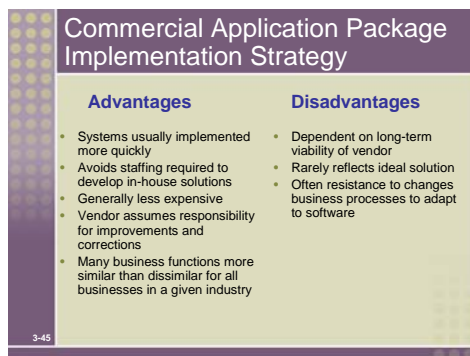
This slide depicts a typical project to select a software package and then integrate that package into a business (and its other existing information systems).

In the chapter, we specifically omitted ERP applications from this COTS route. ERP applications are so large and complex that their vendors typically provide a custom methodology (and consulting) to implement the ERP solution.

The COTS route looks more complex because few software packages fulfill 100 percent of an organizations requirements. Thus, a COTS project does not preclude traditional analysis, design, and construction activities to supplement capabilities not provided by the chosen software package.

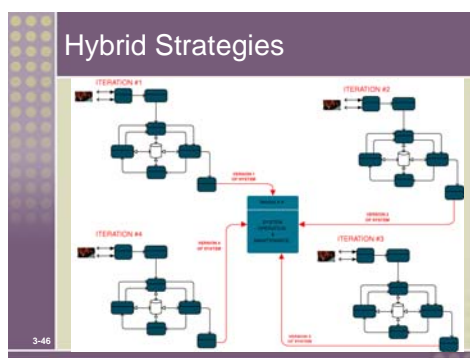
Additionally, most software packages require customization that requires additional requirements analysis, design customization, and programming within the application's embedded language.

Slide 45

**Conversion Notes**

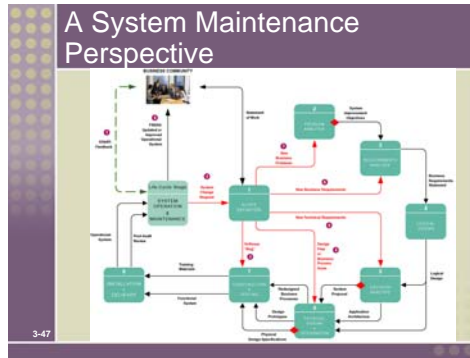
This slide is new to the seventh edition.

Slide 46

**Teaching Notes**

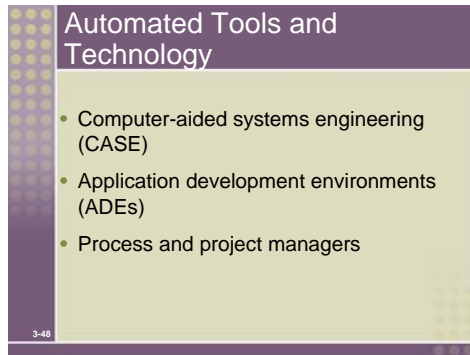
This is one possible hybrid strategy using an incremental strategy in combination with rapid application development.

Slide 47

**Teaching Notes**

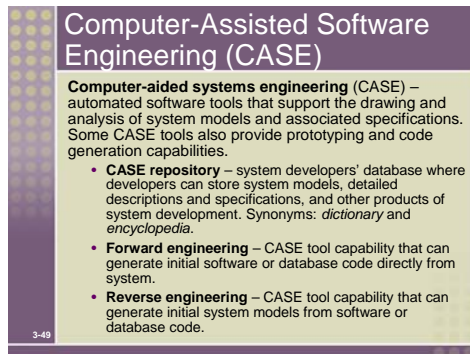
The key point illustrated in this slide is that system maintenance is merely a smaller-scale version of the development methodology that was used to create a system in the first place. Notice that maintenance and reengineering projects do not have to start in the same phase. It is important to recognize that any given phase will not require the same amount of time in a maintenance and reengineering project as it would for a new system development project. Thus, any phase illustrated may require hours or days in maintenance and reengineering versus days, weeks, or months in a new system development project.

Slide 48



No additional notes.

Slide 49

**Teaching Notes**

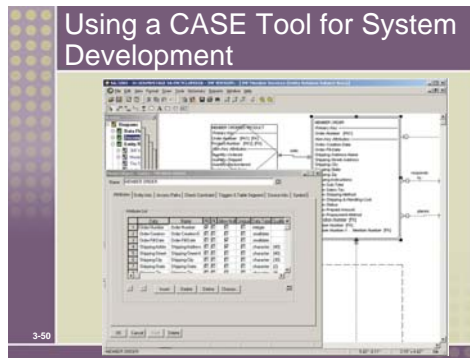
Different CASE tools may refer to their repository as a dictionary or encyclopedia.

Some CASE tools maintain a repository at a project-by-project level. Others provide or integrate into a project-independent repository to promote sharing of models and specifications between projects.

Most CASE tools interface with one or more ADEs to provide round-trip engineering that supports the full life cycle.

If the students will be using a CASE tool during the semester, this would be a good time to introduce it.

Slide 50

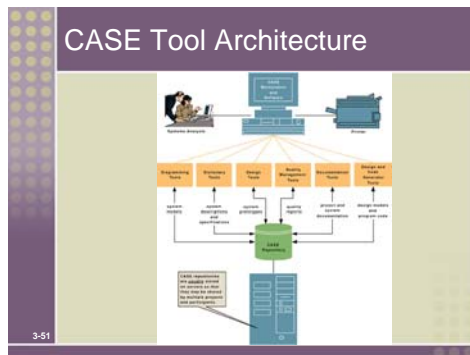
**Conversion Notes**

The sixth edition showed a photograph of a person using a case tool. We believe a screen shot is more useful.

Teaching Notes

At this point in the semester the students will not understand what this screen shot it showing. But you can at least note that this is part of the data design for an information system. If students have had a database course, they will probably relate to the ERD.

Slide 51

**Teaching Notes**

Map your course's CASE (and ADE) environment into this diagram to help your students better understand the automated tools that will be taught in your course.

Slide 52

Application Development Environments

Application development environments (ADEs) – an integrated software development tool that provides all the facilities necessary to develop new application software with maximum speed and quality. A common synonym is *integrated development environment (IDE)*

- ADE facilities may include:
 - Programming languages or interpreters
 - Interface construction tools
 - Middleware
 - Testing tools
 - Version control tools
 - Help authoring tools
 - Repository links

Teaching Notes

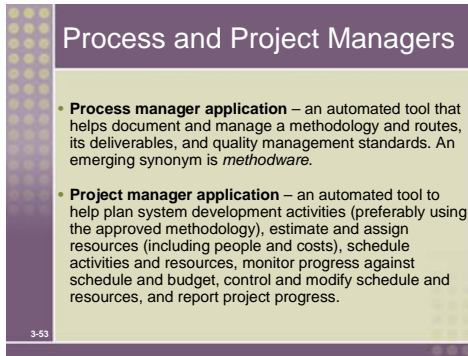
ADE is compatible with both model-driven and rapid application development methodologies. In contemporary literature, it is the basis for all RAD methodologies.

Many ADEs provide links to a repository to support sharing of program code.

Most ADEs either interface with one or more CASE tool repositories or they provide rudimentary CASE-like modeling tools within the ADE.

This allows developers to integrate RAD and MDD techniques as was demonstrated in the first hybrid route presented earlier in the chapter.

Slide 53

The slide has a purple header with the title "Process and Project Managers" in white. The main content area is light green with two bullet points. A vertical purple bar with a grid of small yellow circles is on the left. The slide number "3-53" is in the bottom left corner.

Process and Project Managers

- **Process manager application** – an automated tool that helps document and manage a methodology and routes, its deliverables, and quality management standards. An emerging synonym is *methodware*.
- **Project manager application** – an automated tool to help plan system development activities (preferably using the approved methodology), estimate and assign resources (including people and costs), schedule activities and resources, monitor progress against schedule and budget, control and modify schedule and resources, and report project progress.

3-53

Teaching Notes

Examples of process and project management technology will be covered extensively in Chapter 4.

Answers to End of Chapter Questions and Exercises

Review Questions

1. A standardized process for systems development creates efficiencies that allow management to shift resources between projects. Also, a standardized or consistent methodology will produce consistent documentation that reduces lifetime costs to maintain the system. Lastly, a consistent process promotes quality as well, which can increase competitive advantage.
2. The system life cycle is the lifetime of an information system, divided into two parts—1) systems development and 2) systems operation and maintenance.

System development methodology is a standardized development process that defines a set of activities, methods, best practices, deliverables, and automated tools that system developers and project managers are to use to develop and improve information systems.

They are related because the system development methodology is the means to execute the systems development part of the system life cycle.

3.
 - 1) Get the system users involved
 - 2) Use a problem-solving approach
 - 3) Establish phases and activities
 - 4) Document throughout the development
 - 5) Establish standards
 - 6) Manage the process and projects
 - 7) Justify information systems as capital investments
 - 8) Don't be afraid to cancel or revise scope
 - 9) Divide and conquer
 - 10) Design systems for growth and change
4. Documentation enhances communications and acceptance. It helps increase system user involvement and management confidence in the project. In addition, during the development process, stakeholders typically enter and leave the project. Thus, it is important to make sure other people joining the developing process will understand what is really going on.
5. Process management is an ongoing activity that documents, teaches, oversees the use of, with phases, activities, deliverables, and quality standards that should be consistently applied to all projects.

Project management is the process of scoping, planning, staffing, organizing, directing, and controlling a project to develop an information system at minimum cost, within a specified time frame, and with acceptable quality.

Process management and project management are necessary because we want to maintain high quality in the system development process. In fact, both process management and project management are highly impacted by the need for quality management. Quality standards are built into process to ensure that the activities and deliverables of each phase will contribute to the development of a high-quality information system.

6. Risk management is the process of identifying, evaluating, and controlling what might go wrong in a project before it becomes a threat to the successful completion of a project or implementation of the information system. Risk management is driven by risk analysis and assessment.

Risk management is necessary because it seeks to balance risk and reward, after all different organizations are more or less averse to risk.

7. System owners and system users initiate most projects. The impetus for most projects is some combination of problems, opportunities, and directives.
8. The main participants are the system owners, project managers, and system analysts. Their goals are to see if the problem identified for the organization to implement a system is worth resolving. If so, this phase establishes the size and boundaries of the project, the project vision, limitations, required participants, budget and schedule.
9. The first one is the problem statement (also called preliminary study or feasibility assessment). It is a statement of the problems, opportunities, and/or directives that have given impetus to the project. The problem statement may also include the project vision and identify the constraints.

The second deliverable is the scope statement. It defines the boundaries for the project, i.e., “what is in and what is out.”

The third deliverable is the statement of work (also called project charter, project plan, or service-level agreement). It is a contract with management and system users to develop or enhance an information system. The statement of work defines vision, scope, constraints, high-level user requirements, schedule, and budget.

10. The main participants are the system users and system analysts. They are the main participants because the requirement analysis defines and prioritizes the business requirements, which is based upon what the users need or want out of the system. System users need to be involved because they

are the ones using the system; therefore, they must participate in order for the system analysts to get the requirements.

In addition to that, since the requirements phase is technology independent, system designers and builders are not included in this process.

11. Technical feasibility analysis, operational feasibility analysis, economic feasibility analysis, schedule feasibility analysis, and risk feasibility analysis.
12. It is a system development strategy that emphasizes the drawing of system models, which represent the current and/or desired reality, to help visualize and analyze problems, define business requirements, and design information systems.
13. Requirement specifications tend to be more thorough and better documented.

Business requirements and system designs are easier to conceptualize and validate with pictures than words.

Models make it easier to identify, conceptualize and analyze alternative technical solutions.

Design specifications tend to be more sound, stable, adaptable, and flexible because they are model-based and more thoroughly analyzed before they are built.

Systems can be constructed more accurately the first time when built from thorough and clear model-based specifications.

14. RAD is a system development strategy that emphasizes speed of development through extensive user involvement in the rapid, iterative, and incremental construction of a series of functioning prototypes of a system that eventually evolves into the final system.
15. It is useful for projects in which the user requirements are uncertain or imprecise.

It encourages active user and management participation.

Projects have higher visibility and support because of the extensive user involvement throughout the process.

Users and management see working, software-based solutions more rapidly than they do in model-driven development.

Errors and omissions tend to be detected earlier in prototypes than in system models.

Testing and training are natural by-products of the underlying prototyping approach.

The iterative approach is a more “natural” process because change is an expected factor during development.

16. CASE represents a set of automated tools to help system analysts better perform system modeling with engineering-like precision and rigor by supporting the drawing and analysis of system models.

Example: Computer Associates' Erwin, Oracle's Designer 2000, Popkin's System Architect, Rational Rose, and Visible systems' Visible Analyst

Problem and Exercises

1. Based upon studies, the cost and timelines of a project go down while productivity and quality go up as an organization is able to “mature” its information system development process. The CMM framework is designed to assist organizations and to help them evaluate their system development processes. It does this by categorizing an organization's system development process into five maturity levels, each of which is a steppingstone to the next level.
2. Level 1 – Initial: The organization does not have a standard development process other than chaos. Project success depends almost entirely upon the ability of the project team

Level 2 – Repeatable: The organization has established project management processes and follows a system development process, but this process may change from project to project; project success is still heavily dependent upon the abilities of the project team.

Level 3 – Defined: The organization has established a standard system development process which is stable, predictable and repeatable.

Level 4 – Managed: The organization has established quality and productivity goals and metrics for projects.

Level 5 – Optimizing: The organization has established a continuous project monitoring and improvement process.

3. By far the greatest percentage improvement for all factors – project duration, person-months, quality and cost – occurs when an organization matures and moves from CMM Level 1 to CMM Level 2. Reasons given for this vary, but most include that project costs, schedules and functionality begin to be tracked at CMM Level 2, and that the greatest magnitude of improvement occurs once a tracking system is established.
4. Systems development methodology refers to the methodology used to “execute” and govern the systems development process.

Systems life cycle refers to the “natural” life cycle stages that a system goes through, starting with development, then converting to operations and maintenance, and ending with redevelopment or retirement when obsolescence occurs.

5.
 - a. The organization uses the same systems development approach for all projects.
 - b. Risks associated with taking shortcuts or making mistakes in the development process are reduced.
 - c. Complete documentation in a standardized format is produced for all projects.
 - d. Developer resources can be readily reassigned to different projects because the same development processes are standardized throughout the organization.
 - e. Systems knowledge is maintained by the organization, and does not disappear with staff or contractor turnover.
6.
 - a. Principle 1: The methodology involves the users, in order to get their partnership and commitment, and to ensure that the solution addresses the real business problem and requirements.
 - b. Principle 2: The methodology employs a problem-solving approach to ensure that the right problem is being correctly solved with the right solution.
 - c. Principle 3: The methodology establishes and divides the project into phases to ensure that the systems development process is conducted in a structured, planned process.
 - d. Principle 4: The methodology includes documenting continuously through each phase, rather than waiting for the end of the project. This is to ensure that not only does documentation actually take place, but also to build communication and acceptance, keep management apprised of progress and issues, and increase user involvement.
 - e. Principle 5: The methodology establishes or uses established standards to ensure consistency with the organization’s IT architecture and to ensure effective system integration

- f. Principle 6: The methodology includes a project management process to ensure that the project meets quality standards, stays within budget and is completed on time.
 - g. Principle 7: The methodology views organizational information systems as capital investments. Accordingly, alternative solutions are evaluated based upon their cost-effectiveness, and risk and feasibility are considerations for all project decisions.
 - h. Principle 8: The methodology includes a strategy for continuous evaluation of project scope, feasibility and risk. As part of this strategy, the methodology includes a change management process for adjusting scope, and a risk management plan to identify and mitigate risks. The methodology also acknowledges that cancellation of the project may at times be the best business decision.
 - i. Principle 9: The methodology breaks the system into smaller, more manageable pieces in order to facilitate the problem-solving process.
 - j. Principle 10: The methodology acknowledges the reality of change and that system entropy is a natural occurrence; accordingly the methodology supports system design based upon growth and change.
7. The categories of problems are performance, information and data, economics (costs and profits), control or security, efficiency of people and processes, and service to stakeholders.
- a. Information, Efficiency or Control
 - b. Information or Economics
 - c. Efficiency
 - d. Control
 - e. Efficiency, Information, Service and possibly Control
8. Requirements analysis: A business requirements statement, which documents the business requirements that the new system must meet

Logical design: A non-technical design document that translates the business requirements into a conceptual design for the new system.

Physical design and integration phase: The redesigned business processes, design prototype, and the physical (technical) design specifications.

9. The scope definition phase occurs as the result of a business problem, opportunity, directive, or some combination of any or all of these three triggers.

The stakeholder participants are generally the system owners, project managers and systems analysts.

The first essential question is focused on “is the problem, issue, etc. worth looking at?” If so, the second essential question is what should the project vision be, and what is the project’s size and boundaries, participants, constraints, budget and schedule?

The three important deliverables are the:

- i. Problem statement, which documents and categorizes the problem, opportunity and/or directive, but doesn’t attempt to solve any of them.
 - ii. Scope statement, which identifies the boundaries of the project, i.e., “what’s out, what’s in.”
 - iii. Statement of work, which in essence the contract that agrees to develop the project and specifies the work to be done.
10. System users, systems analysts, and project managers typically are involved and participate in this phase. The primary focus during the requirements analysis phase is on what the system needs to do, not how it should do it. Each proposed requirement should be evaluated on whether it meets one or more of the system improvement objectives defined during the preceding problem analysis phase. The critical error that must be avoided is to cut the requirements analysis short before business needs are fully understood in order to work on the technical solution.
11. In general, system owners (unless they actually use or will be using the system) do not really understand what the system must do from a user perspective. Relying upon system owners for this information may give a false or incomplete picture of the requirements. Additionally, the presence of system owners, who are generally managers and executives, may make it more difficult to elicit candid responses from system users, who are generally at a lower level in the organization. System designers generally should be precluded from participation because of the inherent temptation to start discussing technological solutions before the business problems are fully understood.
12. The essential purpose of the logical design phase is to translate business requirements into system models (diagrams or pictures) which represent the current or desired state, and which are intended to facilitate communication among stakeholders and ensure that requirements have been completely and accurately met.

This stage does not include technological solutions; it is technology-independent.

Common synonyms for logical design include conceptual design and essential design.

Agile modeling, such as extreme programming, is intended to provide “just enough modeling” in order to avoid analysis paralysis (excessive modeling) which needlessly increases development time.

Deliverables are typically logical system models and specifications, which may vary in their level of detail in different methodologies.

The critical transition that takes place is the transition from the business-oriented focus of system owners and users to the technological focus of the system designers and builders.

13. The purpose of the physical design phase is to translate the business requirements into the technical specifications the system builders need to actually construct the system.

The two philosophies are design by specification and design by prototyping. Design by specification is similar in concept to using detailed blueprints to construct a building. Design by prototyping is to build the design shell based upon feedback from users and designers, then to refine and add to the design through an iterative process until it is complete.

Although a project could be cancelled during the physical design phase, this would be unusual unless the project is perceived as beyond redemption in terms of budget or schedule. More commonly, the scope of the project would probably be scaled down.

Most current methodologies provide for overlap between design and construction. The basis for this overlap is that the builders have enough technical information to begin construction before all elements of the design phase are completed, which can decrease overall development time without sacrificing quality.

14. The Rapid Application Development (RAD) strategy might be the best one to use in this situation where requirements are not well known and time-frames are tight, but the organization does not need the entire system delivered all at once. The RAD strategy is to use prototyping techniques to actively engage system users, to condense the typical time required for initial development by focusing on the most critical requirements, and to dramatically decrease the time to implementation of an operational system with basic functionality. Subsequently, iterative development will continue in order to gradually fill out and expand system functionality.

An alternative strategy might be to purchase and customize a commercial application package (also known as commercial off-the-shelf software or COTS). Since designing and building a new system from scratch would not

be required, initial implementation would be generally quicker. Also, COTS often has a “shotgun” or “one size fits all” approach to functionality in order to ensure that a broad range of possible business functions are met.

15. Employing the RAD methodology may increase lifetime system costs through what is frequently perceived as a philosophy of “code, implement and repair” because of its emphasis on speed of implementation. A COTS solution may end up costing far more than anticipated because system integration can be very difficult and time consuming. Further, a COTS solution may end up driving the business processes, rather than the other way around with the business processes driving the technological solution.

Project and Research

1. a. SEI supports three CMM products (as of April 2005):
 - [CMMI®](#)(Capability Maturity Model Integration)
 - [P-CMM](#) (People Capability Maturity Model)
 - [SA-CMM](#) (Software Acquisition Capability Maturity Model)
- b. All three of the CMM products are conceptually the same and are based upon the maturity levels described in the textbook. The CMMI product is focused on product and process improvement in an organization, the P-CMM on critical human resource issues, and the SA-CMM on benchmarking and improving the software acquisition process in an organization.
- c. Responses should indicate a maturity level consistent with the characteristics described for each level. Responses may indicate that an organization can be at a certain maturity level for certain processes, but at a different level for other processes. This is actually more common than not; relatively few organizations fall entirely within one maturity level. Further, the most common levels for organizations are Levels 1 through 3; very few organizations (in the United States at least) are at Level 4 completely and perhaps a handful may be at Level 5.
- d. Depending upon the current level(s) of the organization, the response should reflect the characteristics described for the next higher level.
- e. Responses are somewhat open-ended; however, in general, it would be very difficult to build a case that it is not cost-beneficial to advance to Level 3; the cost-benefits to move from Level 3 to Level 4 may vary depending upon the nature of the organization and the criticality of its information systems and the ROI may be longer. From Level 4 to Level 5,

the cost-benefits may be offset or outweighed by the extended ROI period.

2. Student responses should indicate that dropping attendance of system users and owners at meetings, particularly early in the project, is a serious sign of disengagement (or failure to initially engage). While this is certainly a common occurrence in all too many projects, it is not a sign of a healthy project, and disengagement requires prompt attention and correction at any phase in the project, and particularly so in the early stages. Answers should address finding out what has caused owners and users to disengage, e.g., the analysis is going in a direction that doesn't address what owners and users perceive to be the real organizational problems, the project team is inexperienced and does not know how to effectively conduct these types of meetings, the project initiation phase was inadequate, etc. Answers should also address remediation measures to attempt to re-engage owners and users, such as bringing in a consultant with project expertise to help the project manager, adopting a different methodology for project communications and meetings, etc.
3. Responses can be open-ended, but they should indicate that the student has made a serious search for other methodologies, and understands that no methodology is "carved in stone." The response should also indicate an understanding of whether the differences are cosmetic or fundamental, and the basis for these differences.
4. Responses should indicate that student understands how to use the PIECES framework to categorize problems. Most responses should find moderate to a great deal of commonality of problems, and that this leads to a understanding that different systems frequently have the same problems, once the different descriptions are applied against the PIECES categories. Most responses should also indicate that the PIECES framework is of the most value early in the requirements phase.
5. Responses are open-ended, but should be methodical, and conclusions, if any, should be supported by the survey responses. Responses should also indicate that the use of CASE tools varies widely between different organizations.
6. Responses are open-ended, but should be thoughtful and indicate an understanding that project abandonment may sometimes be necessary, that it doesn't always represent a failure, and that strategies exist to make project abandonment a structured process.

Mini-cases

1. Note to Professor: Scope creep is a common phenomenon. Encourage students to explore with the managers the impacts of scope creep on project success and management.
2. Project champions are generally loathe to give up on a “pet” project. This is even more so when they have already invested considerable time and money into its development. It is critical that at continuous stages of a project the program or system is re-evaluated to determine feasibility and probability for successful completion. Part of that consideration should be the availability of off-the-shelf software that can complete the job without custom program development.
3. What characteristics make a good project manager? In reality, when we say project manager do we mean project leader? There is a saying that managers “do things right” and leaders “do the right thing.” (reference unknown) Project management is an issue of both management of resources, rules, etc. and leadership of a team. Excellence as a manager does not necessarily equal excellence in project management. The same is true of leadership.
4. Answers will vary. However, consideration should be given to whether inventory management holds a strategic advantage for this company, as well as the expectation of success of “tweaking” the program.

Team and Individual Exercises

There are no answers to this section.