



C H A P T E R

# 3

## INFORMATION SYSTEMS DEVELOPMENT

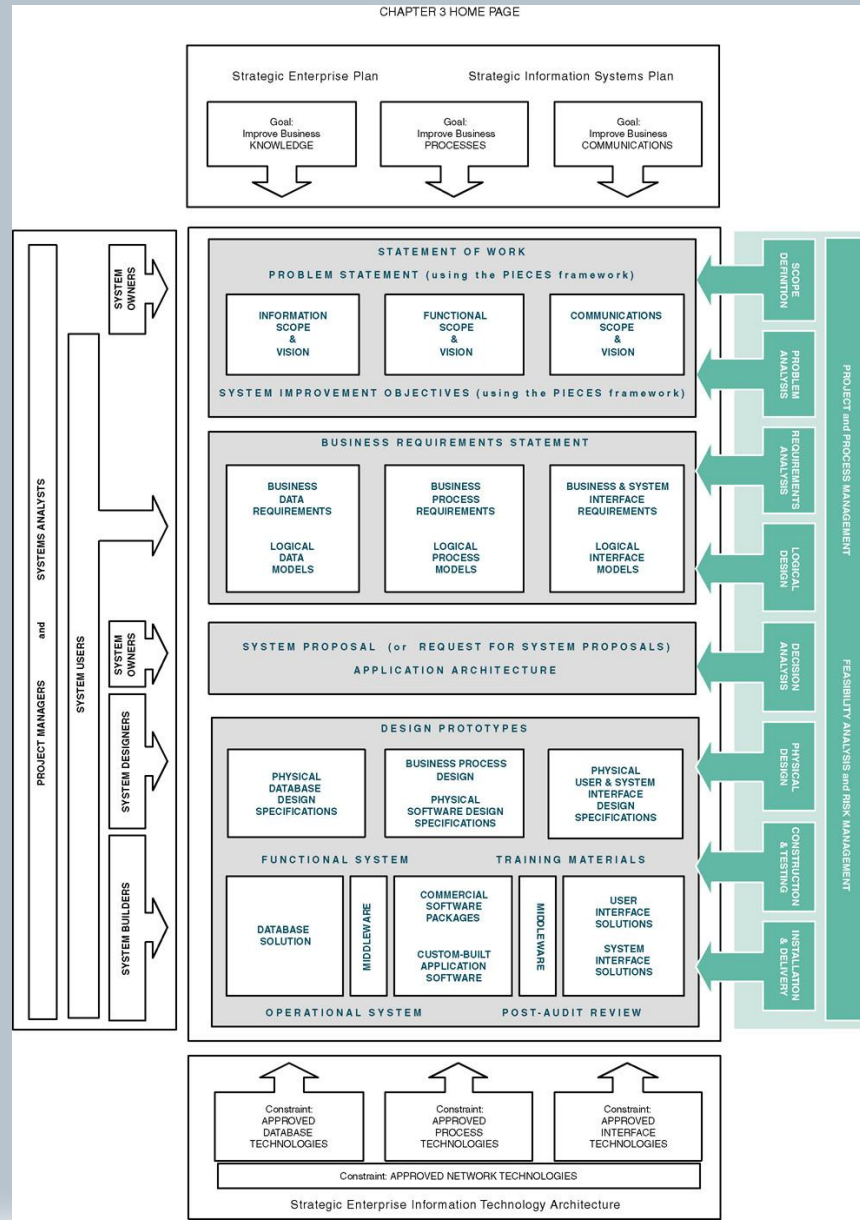


# Chapter Three Information System Development

- Describe the motivation for a system development process in terms of the Capability Maturity Model (CMM) for quality management.
- Differentiate between the system life cycle and a system development methodology.
- Describe 10 basic principles of system development.
- Define problems, opportunities, and directives—the triggers for systems development projects.
- Describe the PIECES framework for categorizing problems, opportunities, and directives.
- Describe the essential phases of system development. For each phase, describe its purpose, inputs, and outputs.
- Describe cross life cycle activities that overlap multiple system development phases.
- Describe typical alternative “routes” through the basic phases of system development. Describe how routes may be combined or customized for different projects.
- Describe various automated tools for system development.



# Chapter Map



# 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



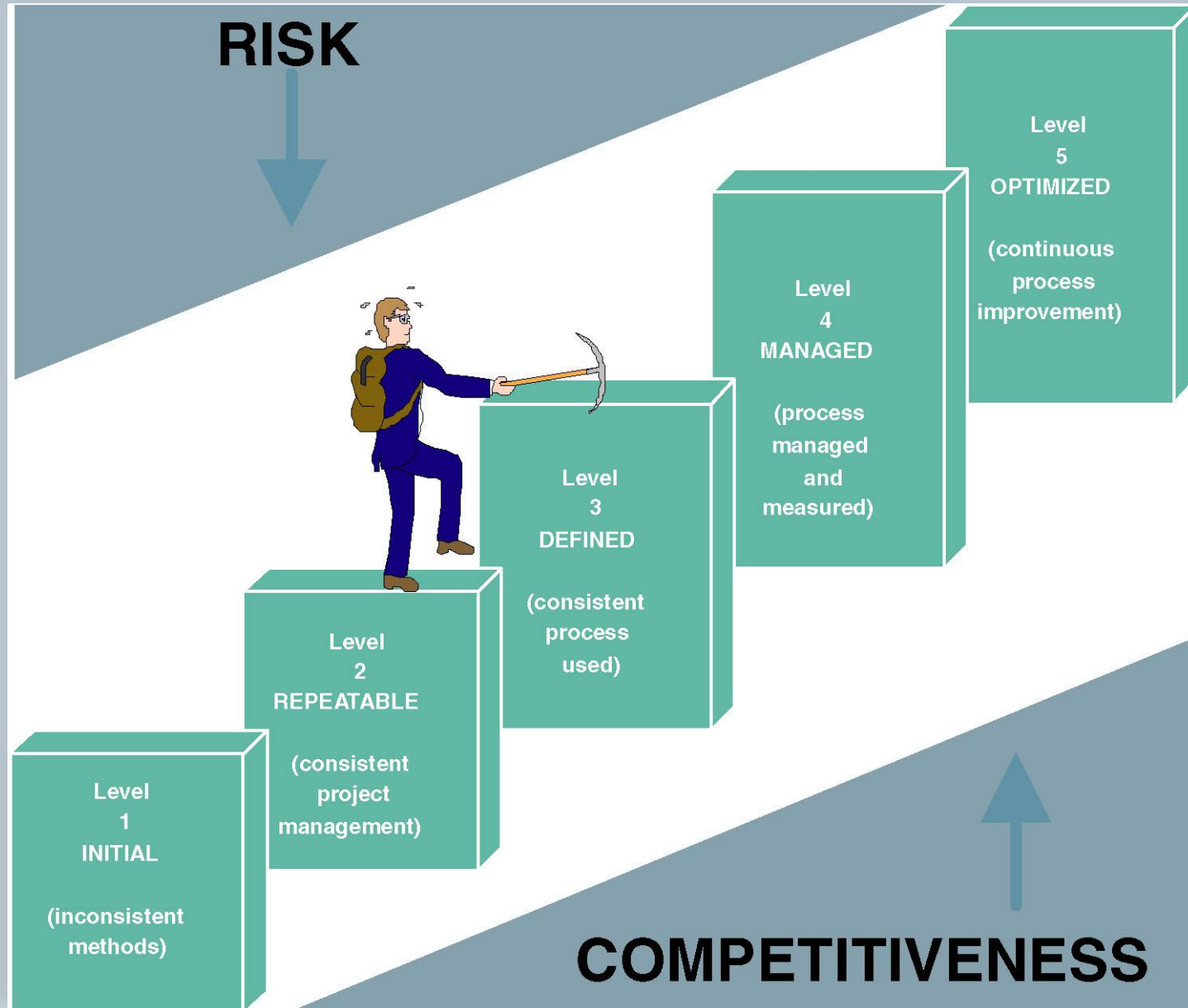
# The CMM Process Management Model

**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 are established to track project costs, schedules, and functionality.
- **Level 3—Defined:** A standard system development process (sometimes called a “methodology”) is purchased or developed. All projects use a version of this process to develop and maintain information systems and software.
- **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.



# Capability Maturity Model (CMM)



# 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



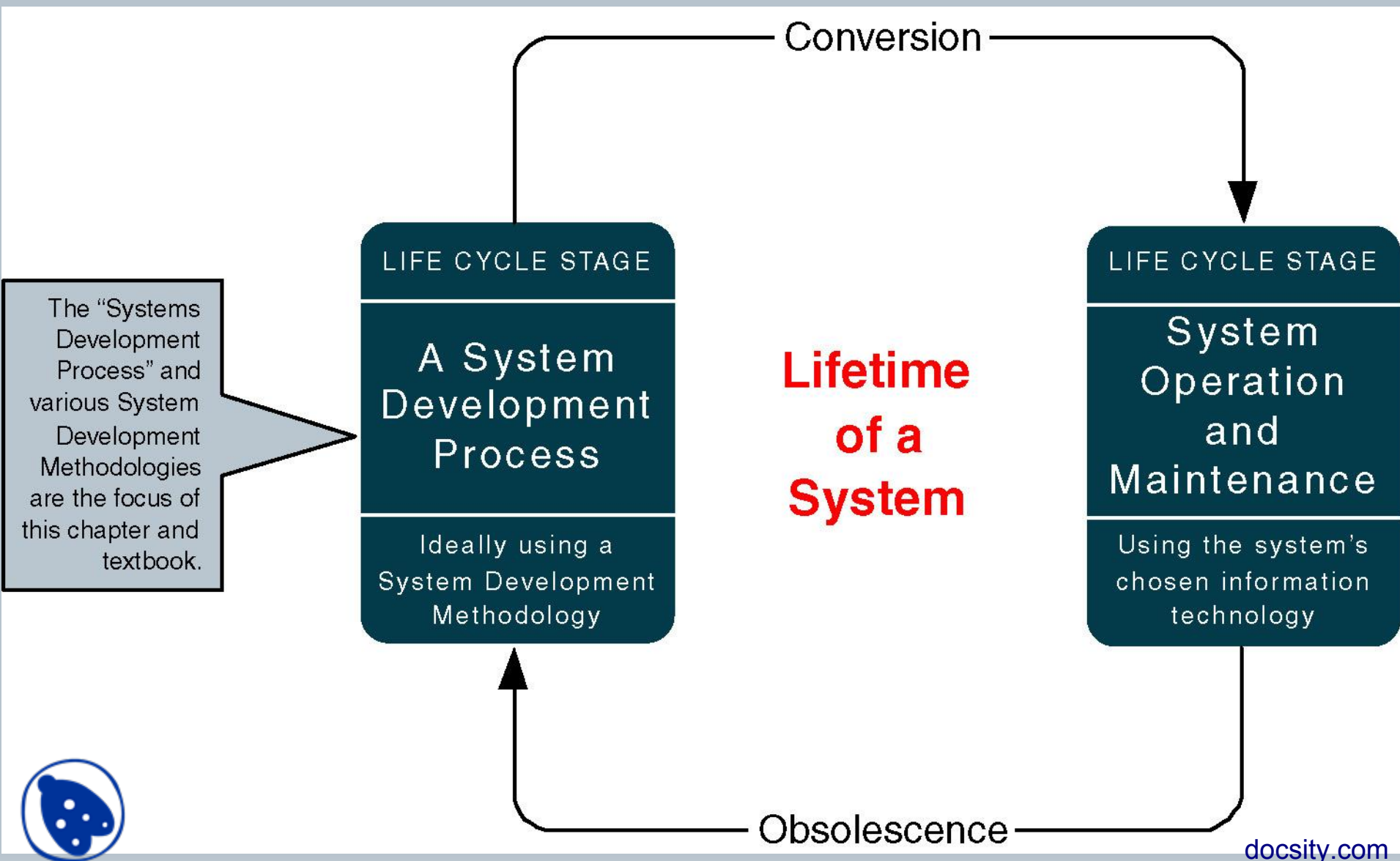
# Life Cycle versus Methodology

- **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 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.





# A System Life Cycle



# 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)



# 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.



# Use a Problem-Solving Approach

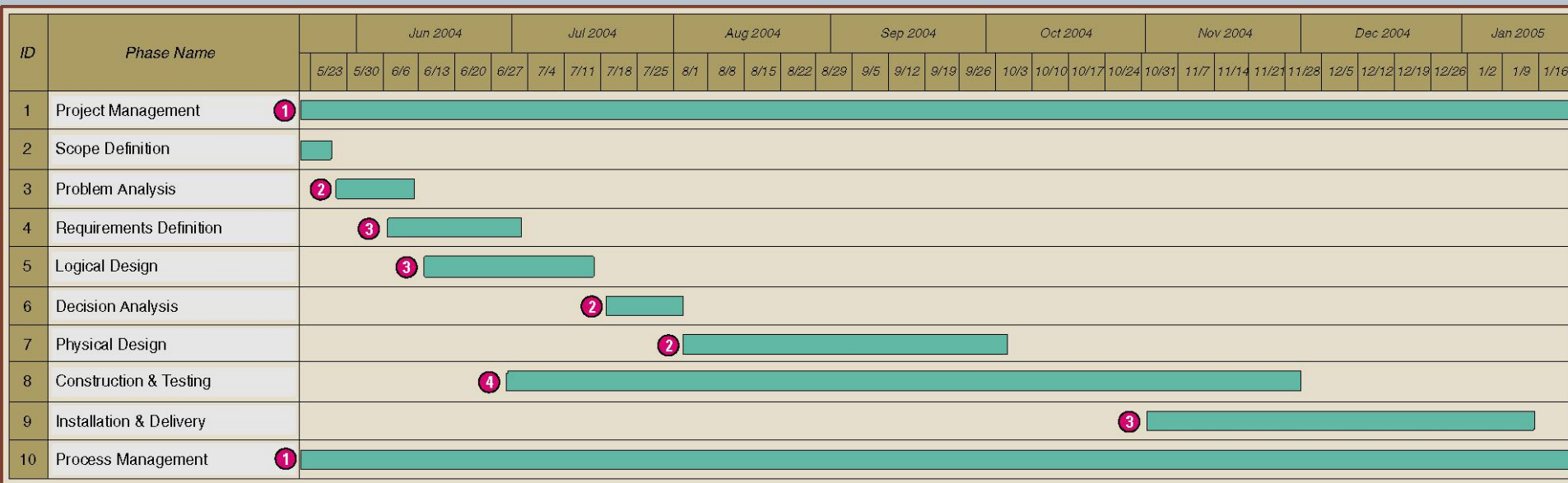
## Classical Problem-solving approach

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.



# Establish Phases and Activities

## Overlap of System Development Phases



# 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.



# 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.



# 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 drive by risk analysis or assessment.





# 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.



# 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. Now it is time to redesign the supporting information system for those redesigned business processes.



# 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.

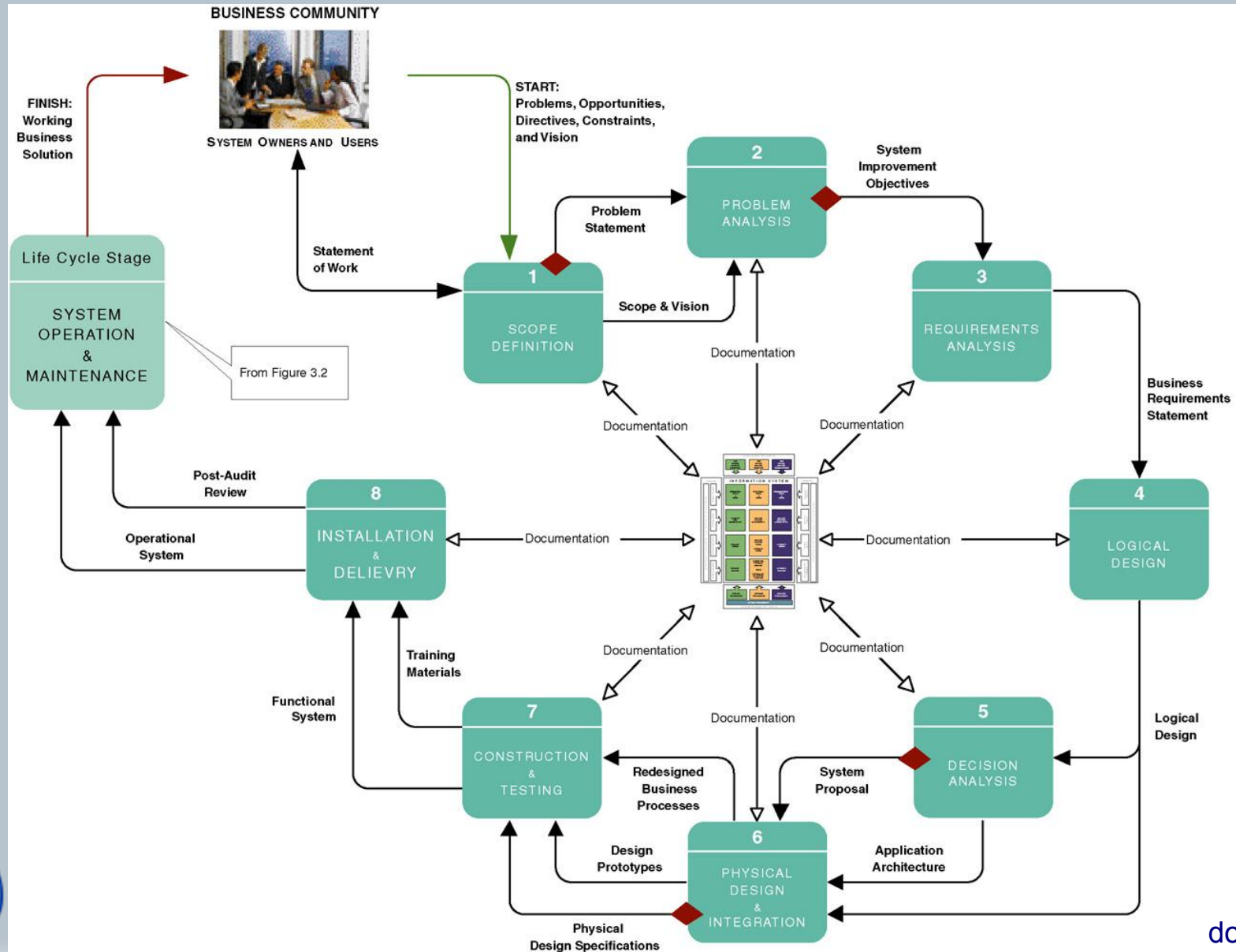


# The **PIECES** Problem-Solving Framework

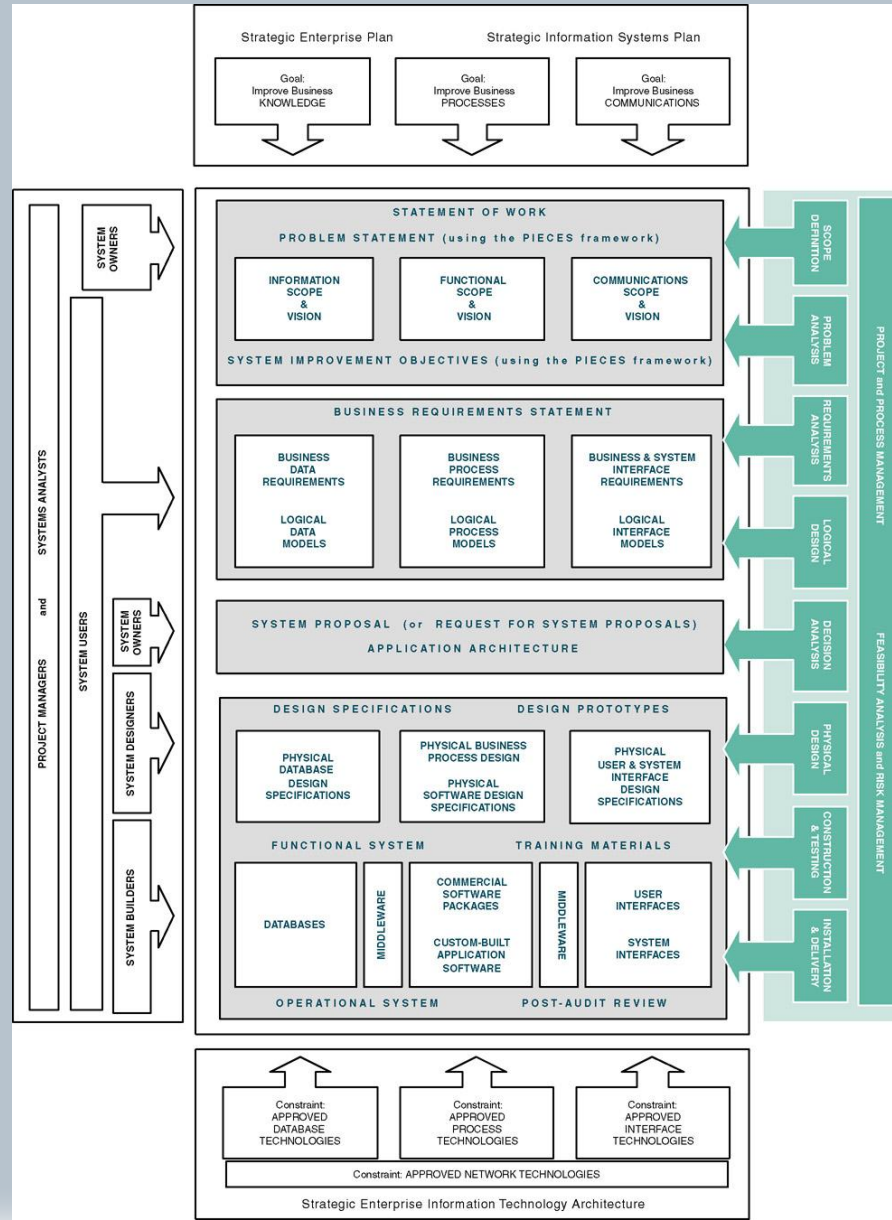
- P** the need to improve performance
- I** the need to improve information (and data)
- E** the need to improve economics, control costs, or increase profits
- C** the need to improve control or security
- E** the need to improve efficiency of people and processes
- S** the need to improve service to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



# The Classic Project Phases



# Building Blocks View of System Development



# Scope Definition

**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*.



# Logical Design

**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.





# Decision Analysis

- 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?



# Physical Design & Integration

**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.



# 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.



# Cross Life-Cycle Activities

**Cross life-cycle activity** – any activity that overlaps many or all phases of the systems development process.

- **Fact-finding**

- **Fact-finding** - the formal process of using research, interviews, meetings, questionnaires, sampling, and other techniques to collect information about system problems, requirements, and preferences.

- **Documentation and presentation**

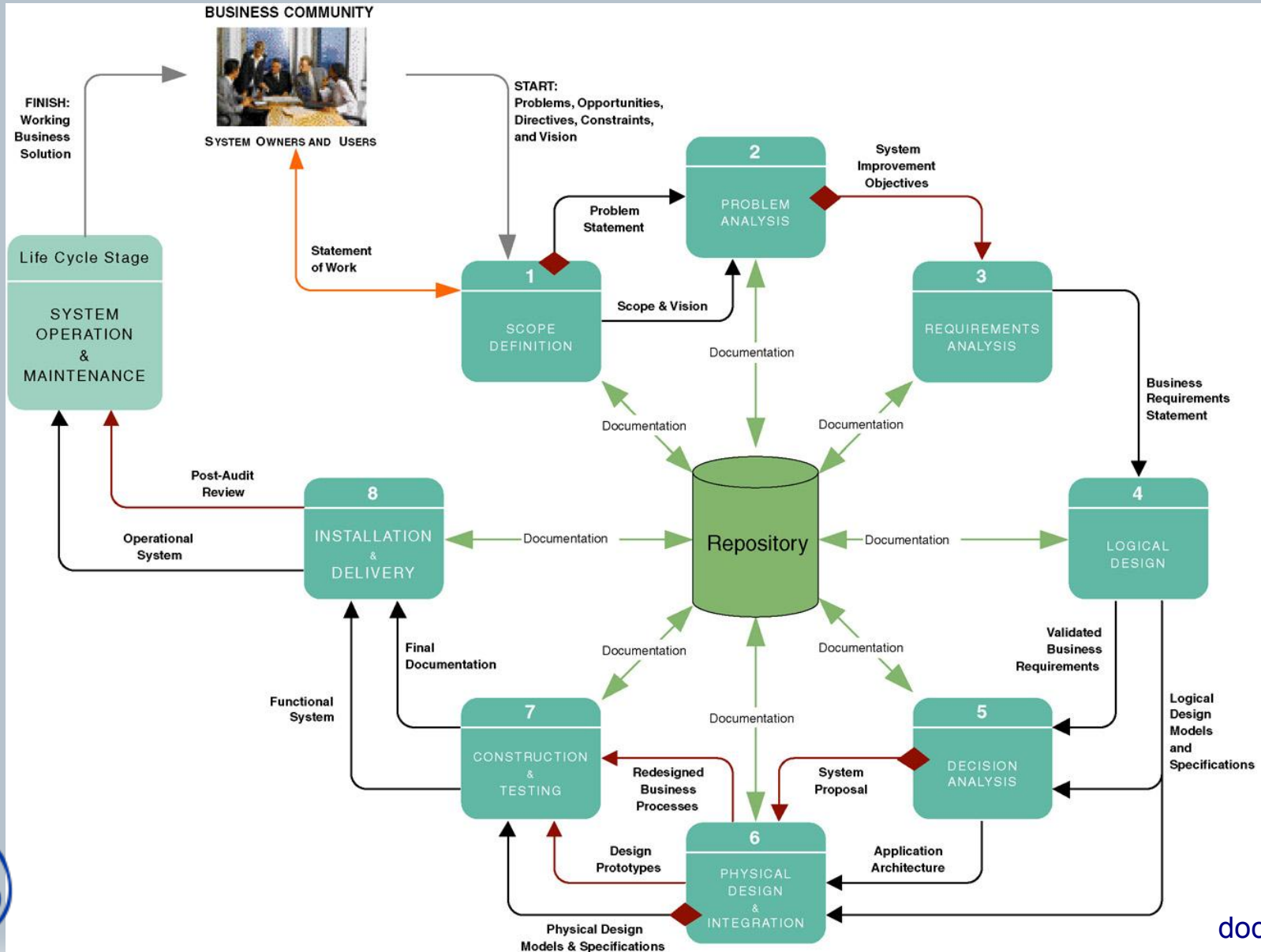
- **Documentation** – the ongoing activity of recording facts and specifications for a systems for current and future reference.
- **Presentation** – the ongoing activity of communicating findings, recommendations, and documentation for review by interested users and managers.
- **Repository** – a database and/or file directory where system developers store all documentation, knowledge, and artifacts for one or more information systems or projects.

- **Feasibility analysis**

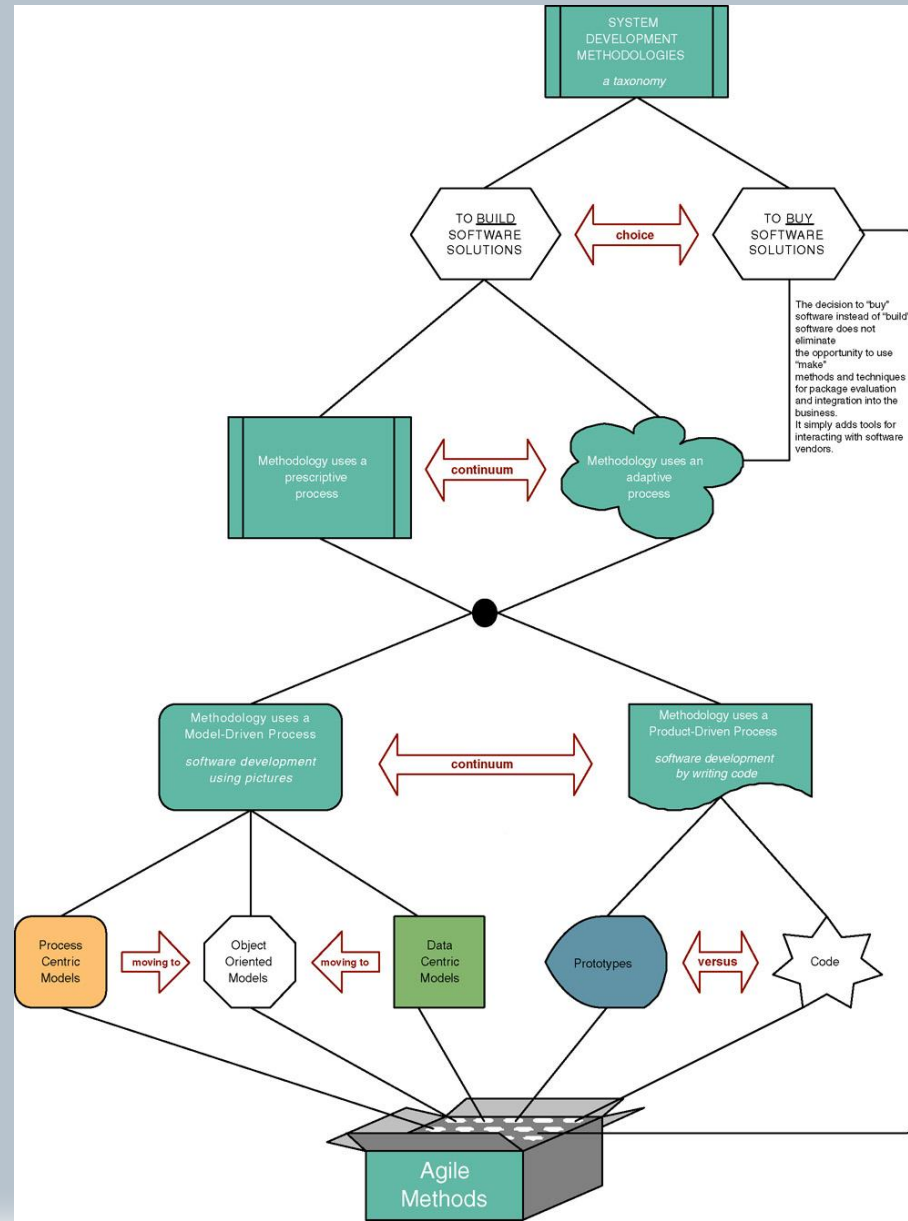
- **Process and project management**



# System Development Documentation, Repository, and Presentations



# A Taxonomy for System Development Methodologies & Strategies



# 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.





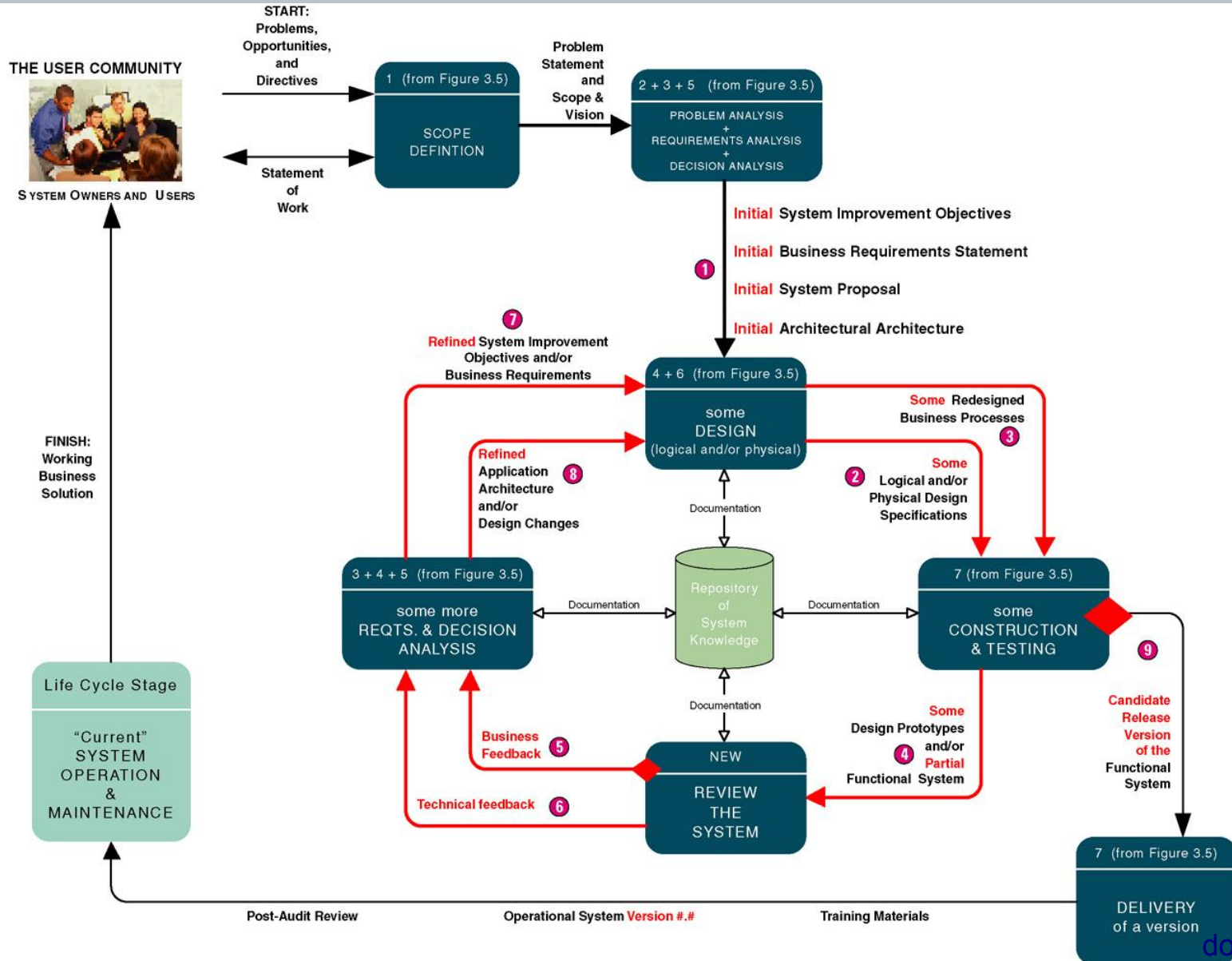


# Rapid Application Development Strategy

- **Rapid application development (RAD)** – a system development strategy that emphasizes speed of development through extensive user involvement in the rapid, iterative, and incremental construction of series of functioning prototypes of a system that eventually evolves into the final system.
  - **Prototype** – a small-scale, representative, or working model of the users' requirements or a proposed design for an information system.
  - **Time box** – the imposition of a nonextendable period of time, usually 60-90 days, by which the first (or next) version of a system must be delivered into operation.



# Rapid Application Development Strategy

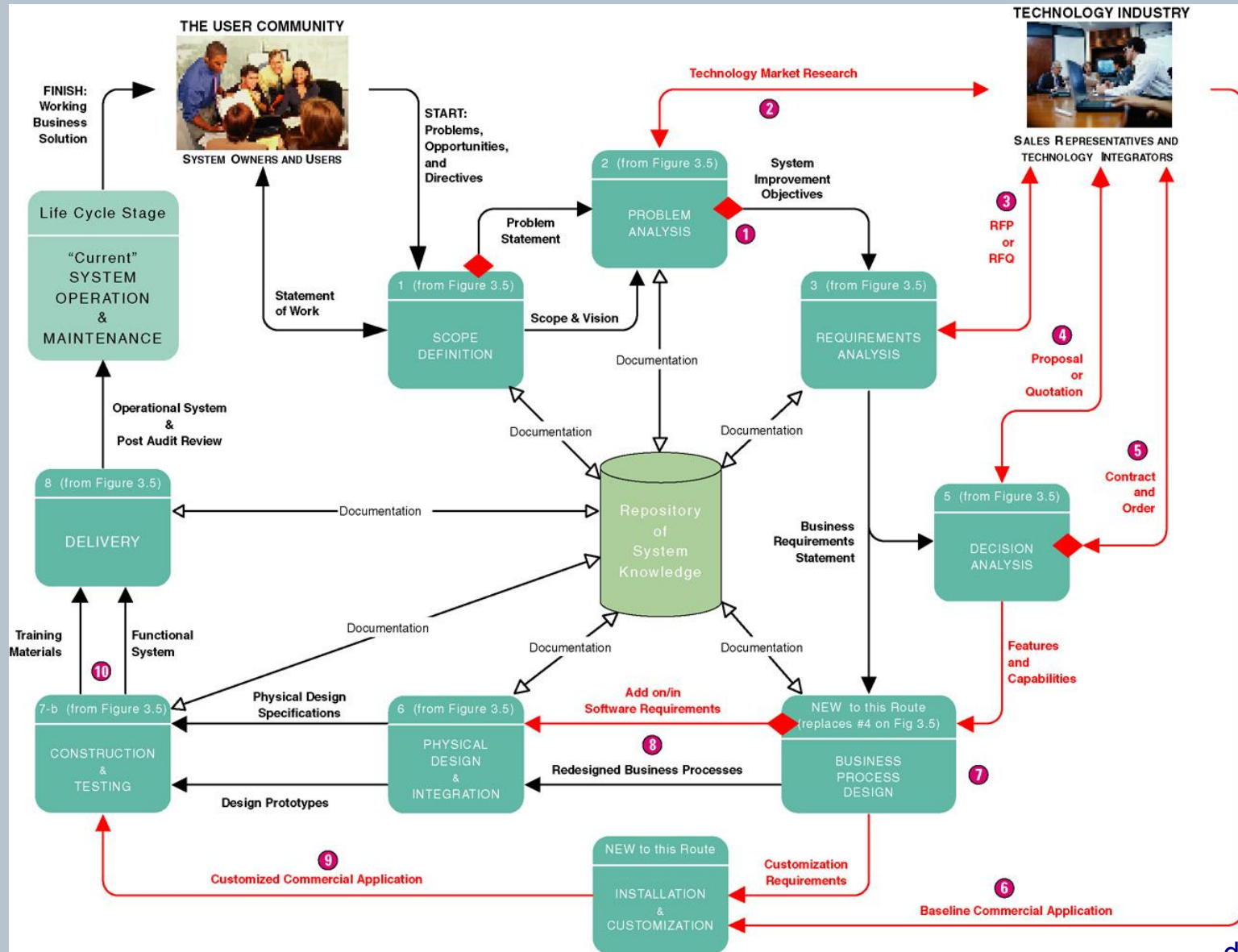


# Commercial Application Package Implementation Strategy

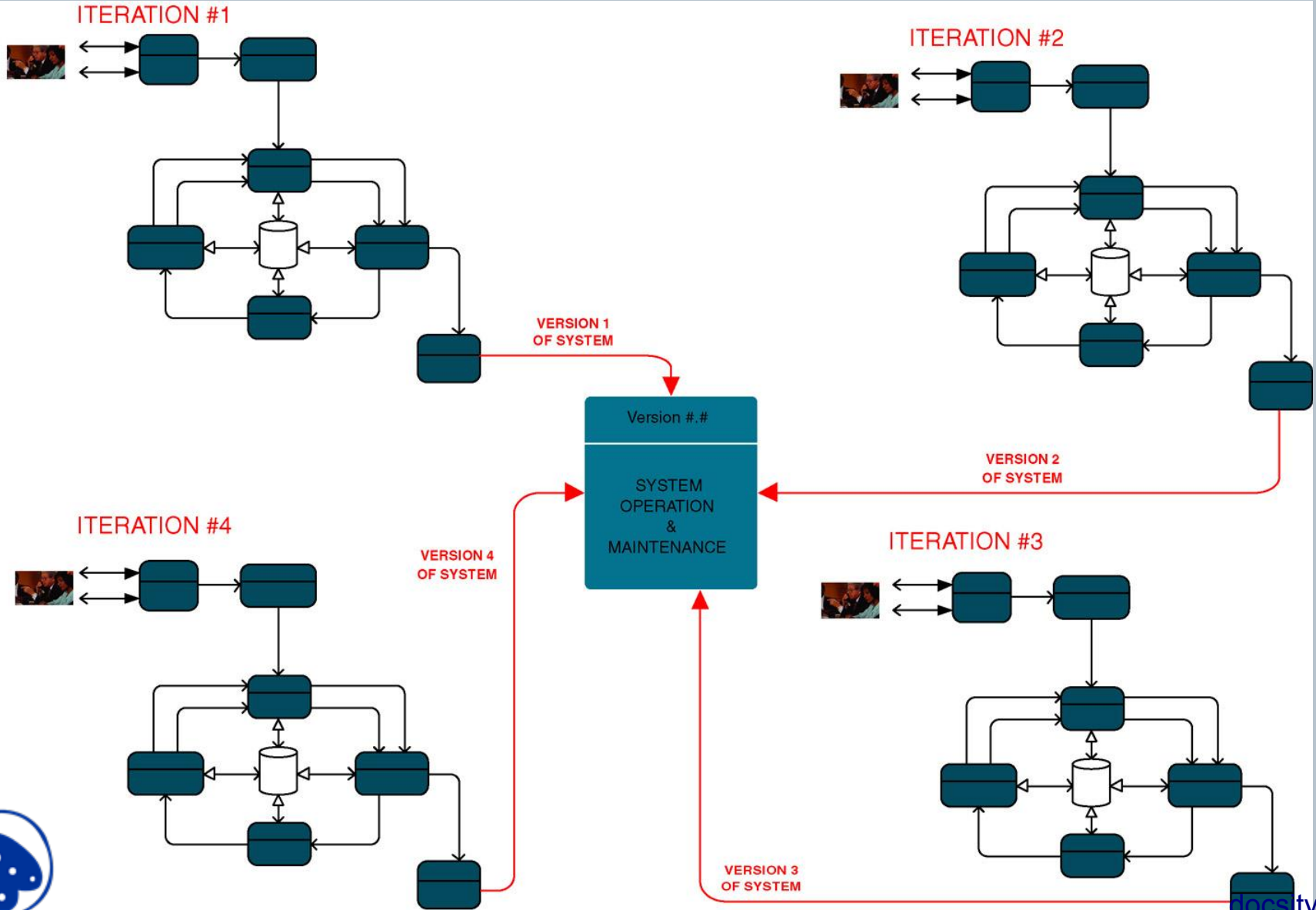
- **Commercial application package** – a software application that can be purchased and customized to meet the business requirements of a large number of organizations or a specific industry. A synonym is *commercial off-the-shelf* (COTS) system.
  - **Request for proposal (RFP)** – a formal document that communicates business, technical, and support requirements for an application software package to vendors that may wish to compete for the sale of that application package and services.
  - **Request for quotation (RFQ)** – a formal document that communicates business, technical, and support requirements for an application software package to a single vendor that has been determined as being able to supply that application package and services.
  - **Gap analysis** – a comparison of business and technical requirements for a commercial application package against the capabilities and features of a specific commercial application package for the purpose of defining the requirements that cannot be met.



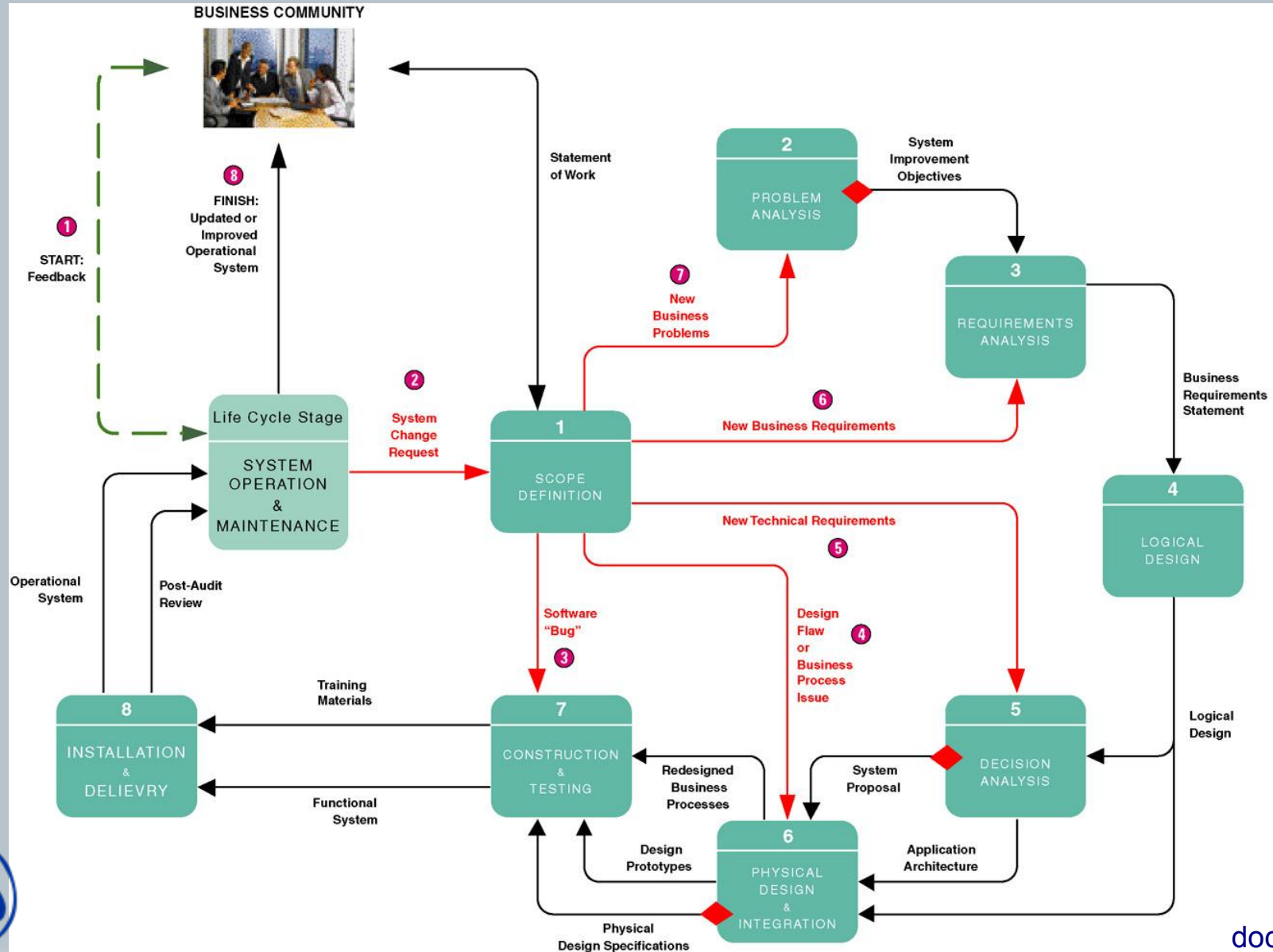
# Commercial Application Package Implementation Strategy



# Hybrid Strategies



# A System Maintenance Perspective



# Automated Tools and Technology

- Computer-aided systems engineering (CASE)
- Application development environments (ADEs)
- Process and project managers





# Computer-Assisted Software Engineering (CAS)

**Computer-aided systems engineering (CASE)** – the use of automated software tools that support the drawing and analysis of system models and associated specifications. Some CASE tools also provide prototyping and code generation capabilities.

- **CASE repository** – a system developers' database where developers can store system models, detailed descriptions and specifications, and other products of system development. Synonyms include *dictionary* and *encyclopedia*.
- **Forward engineering** – a CASE tool capability that can generate initial software or database code directly from system.
- **Reverse engineering** – a CASE tool capability that can generate initial system models from software or database code.

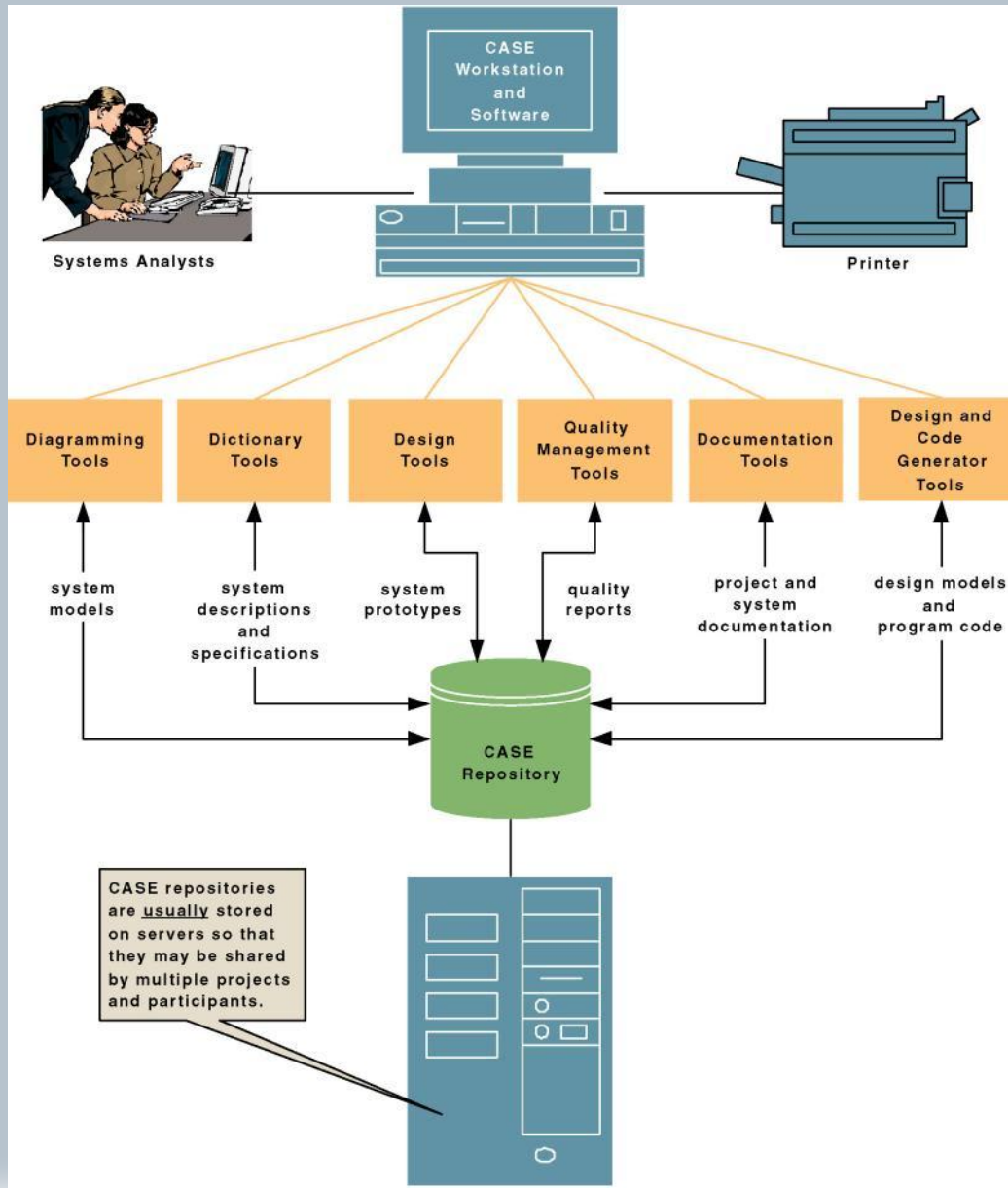




# Using a CASE Tool for System Development



# CASE Tool Architecture



# 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



# 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.

