CHAPTER

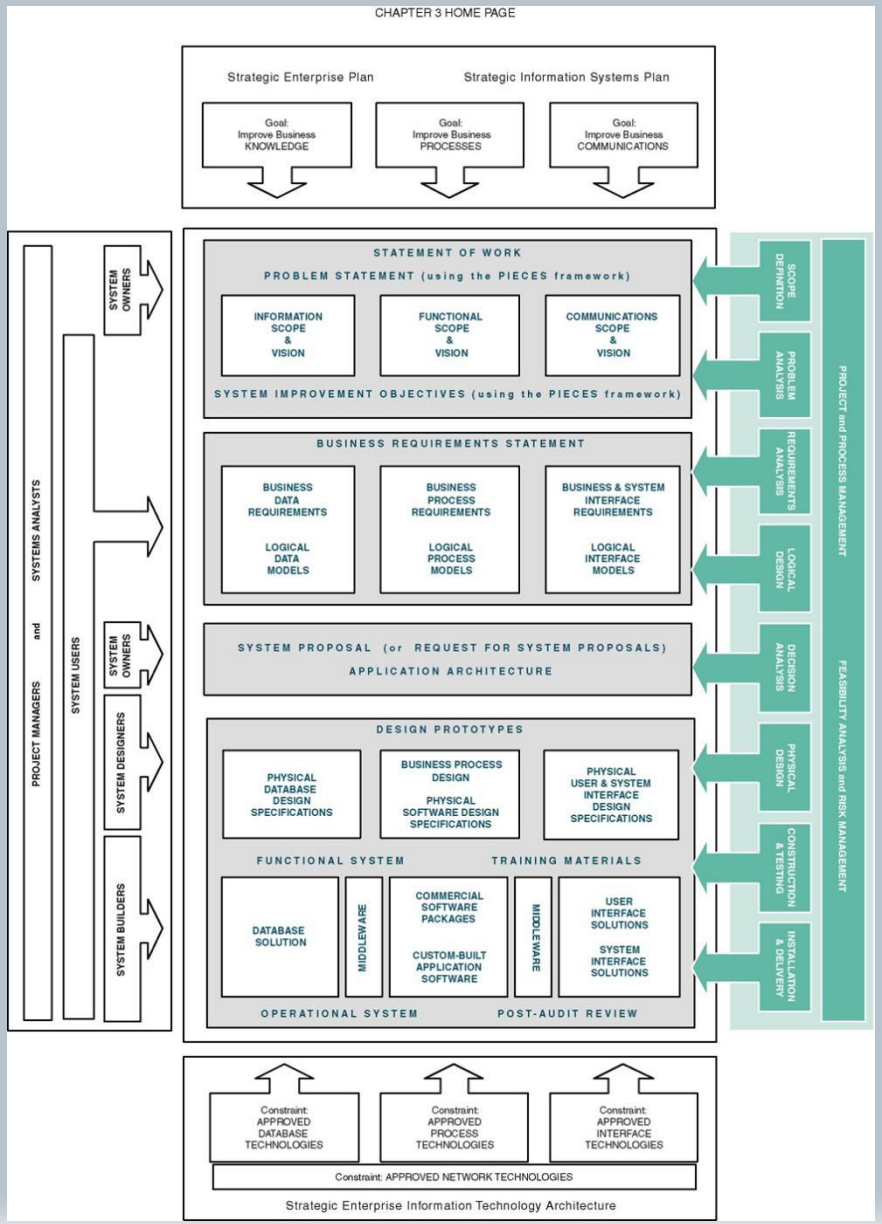
3

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



System Development Process

Systems development process:

- a set of activities, methods, best practices, deliverables, and automated tools that stakeholders use to develop and continuously improve information systems and software .

Using a standardized process:

- Create efficiencies that allow management to shift resources between projects
- Produces consistent documentation that reduces lifetime costs to maintain the systems
- Promotes quality: All US Gov. systems development requires certain quality management requirements **such as CMM model**.

The CMM Process Management Model

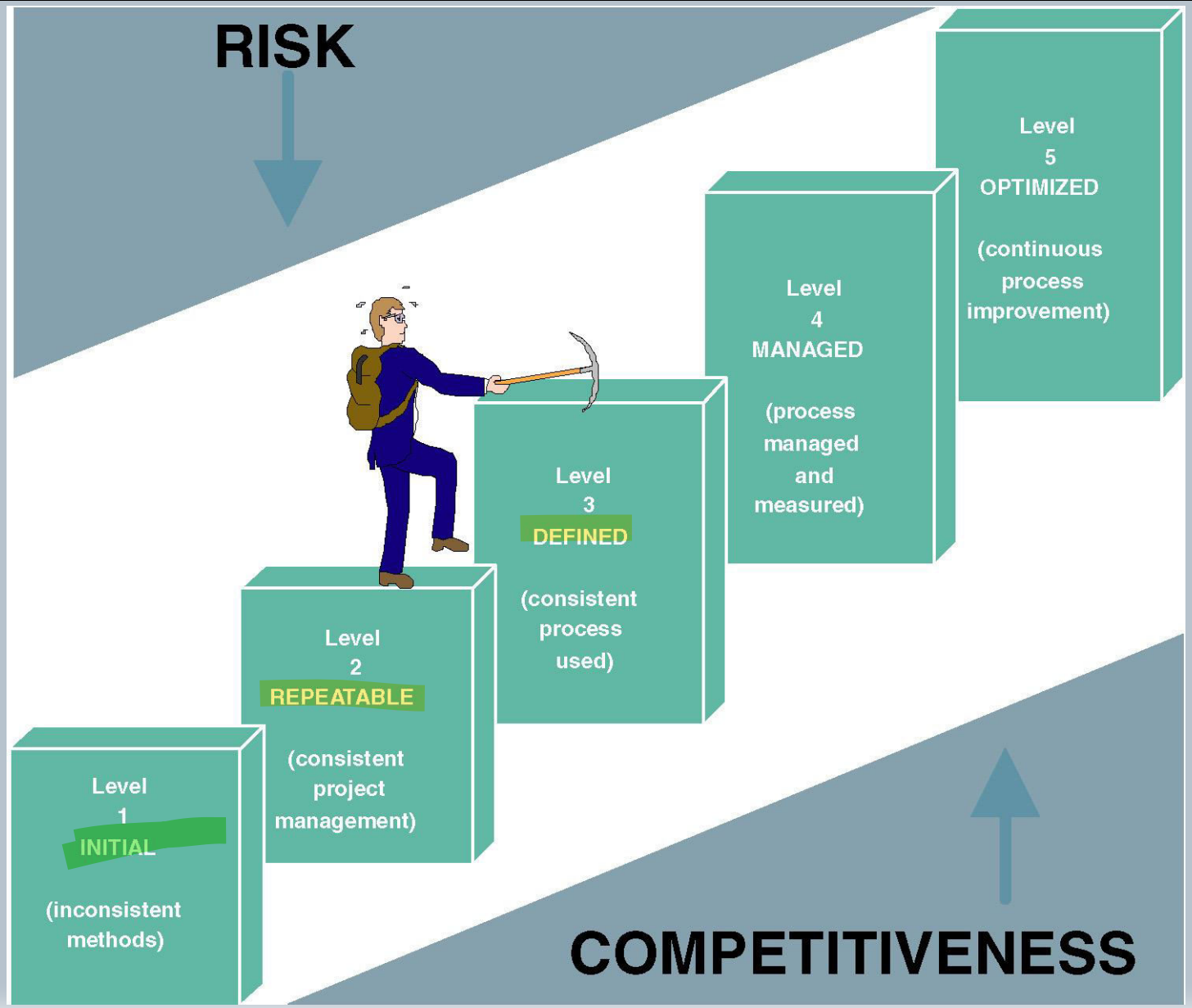
Capability Maturity Model (CMM) – a standardized framework for assessing the maturity level of an organization's information system development and management processes.

- Highly recognized and famous model
- Developed by The Software Engineering Institute (SEI)
- Is being used to qualify IS developers for US federal Gov. projects.

CMM consists of five levels of maturity.

- **Each level is a prerequisite for the next level.**

Capability Maturity Model (CMM)



The CMM Process Management Model (con't)

- **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 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 (Not even MS….)
- **Level 5—Optimizing**: The standardized system development process is continuously monitored and improved based on measures and data analysis established in Level 4.

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

- Significant improvements in terms of schedule and cost at level 3
- Thus, most organizations pursuing the CMM are targeting level 3.

More about CMM model

- Implications
  - You CANNOT skip level
    - It takes time to move from one level to the next
    - Not many organization are above level 1
    - Highly difficult to start at the level 3
  - Esoteric new technology should be avoided at the lower levels (especially level 1 and 2)
  - Starting level for the off-the-shelf systems
  - The levels are already becoming important for contracts.

10 Commandments (Principles) of System Development

1. Get the system users involved.
2. Use a problem-solving approach (slide)
3. Establish phases and activities (slide)
4. Document through development
5. Establish standards
6. Manage the process and projects (slide)
7. Justify systems as capital investments (slide)
8. Don't be afraid to cancel or revise scope (slide)
9. Divide and conquer.
10. Design systems for growth and change.

Next

Use a Problem-Solving Approach

Back

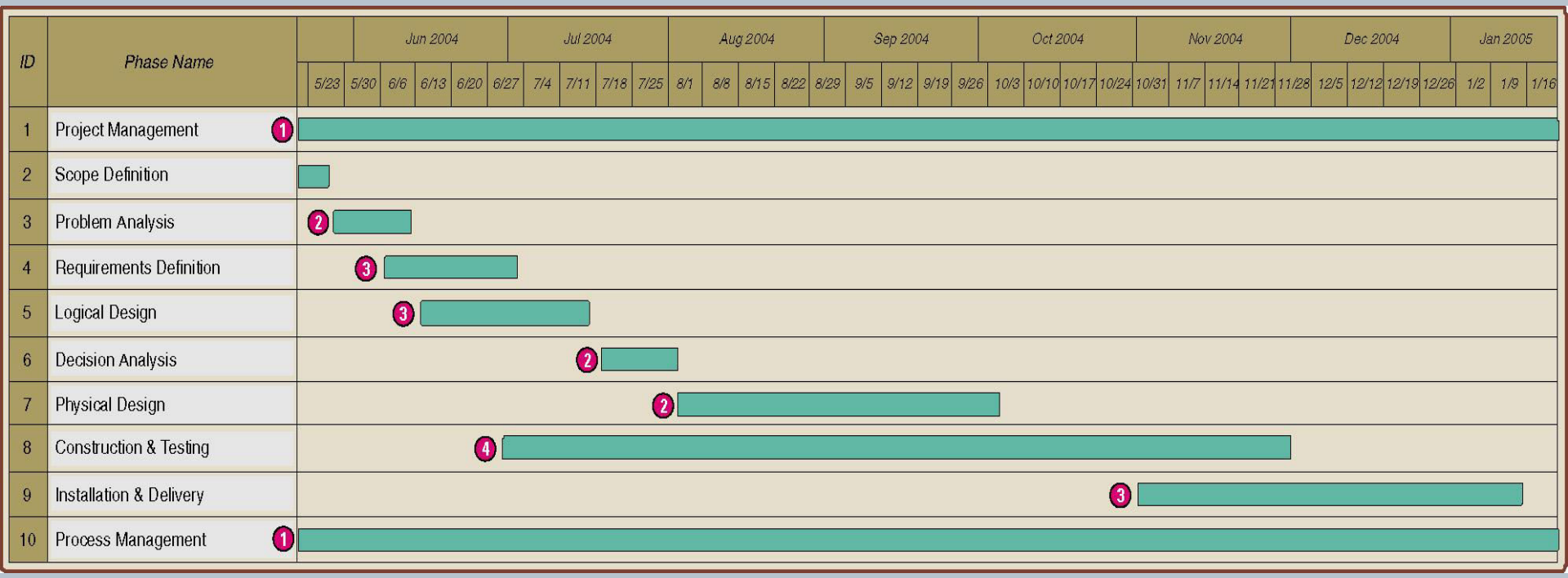
Classical Problem-solving approach

1. Study and understand the problem, its context, and its impact.
2. Define the requirements that must be meet 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

Back

Overlap of System Development Phases



In real world, the phases tend to overlap one another.

Manage the Process and Projects

Back

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

Back

- Cost-benefit analysis through out the development process
  - At each phase, reevaluate cost effectiveness, risk, and feasibility.
  - **Most system owners want more of their systems than they can afford or are willing to pay.**
- Other extra advices..
  - Should not blindly accept the first solution suggested
  - Always evaluate each possible solution

Don't Be Afraid to Cancel or Revise Scope

Back

- Cancel the project if it is no longer feasible
  - possible???
- Reevaluate and adjust the costs and schedule if project scope is to be increased.
  - someone should be blamed…
- Reduce the scope if the project budget and schedule are frozen and not sufficient to cover all project objectives.
- Manage feasibility throughout the project in order to control risks.

A System Development Process

- Systems owners and users usually initiate projects
- Usually use a framework to identify problems
  - **PIECES**
    - Practical framework to identify problems
    - The categories of the PIECES framework tend to overlap one another.

The PIECES Problem-Solving Framework

P the need to improve performance

I the need to improve information (and data)

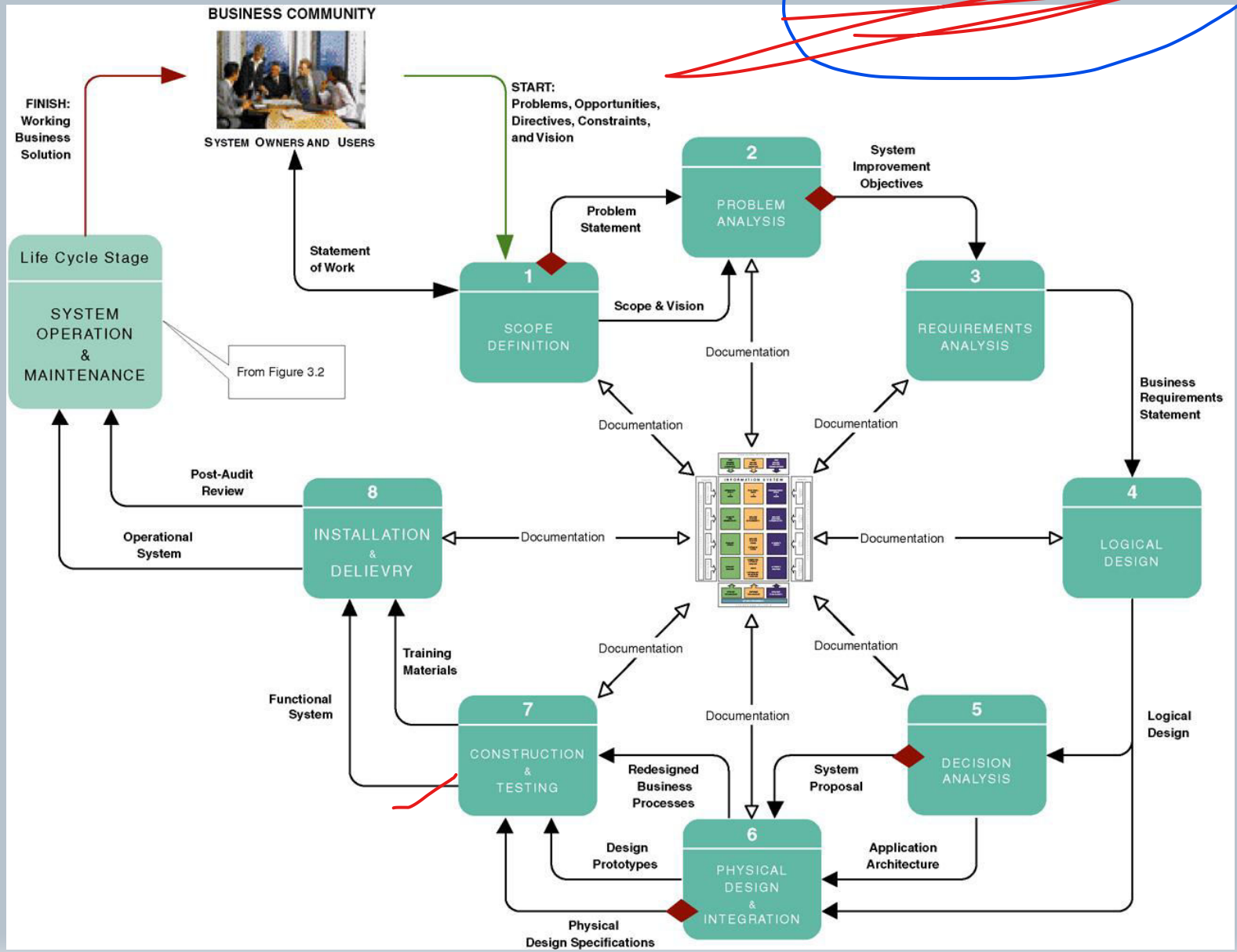
E the need to improve economics, control costs, or increase profits

C the need to improve control or security

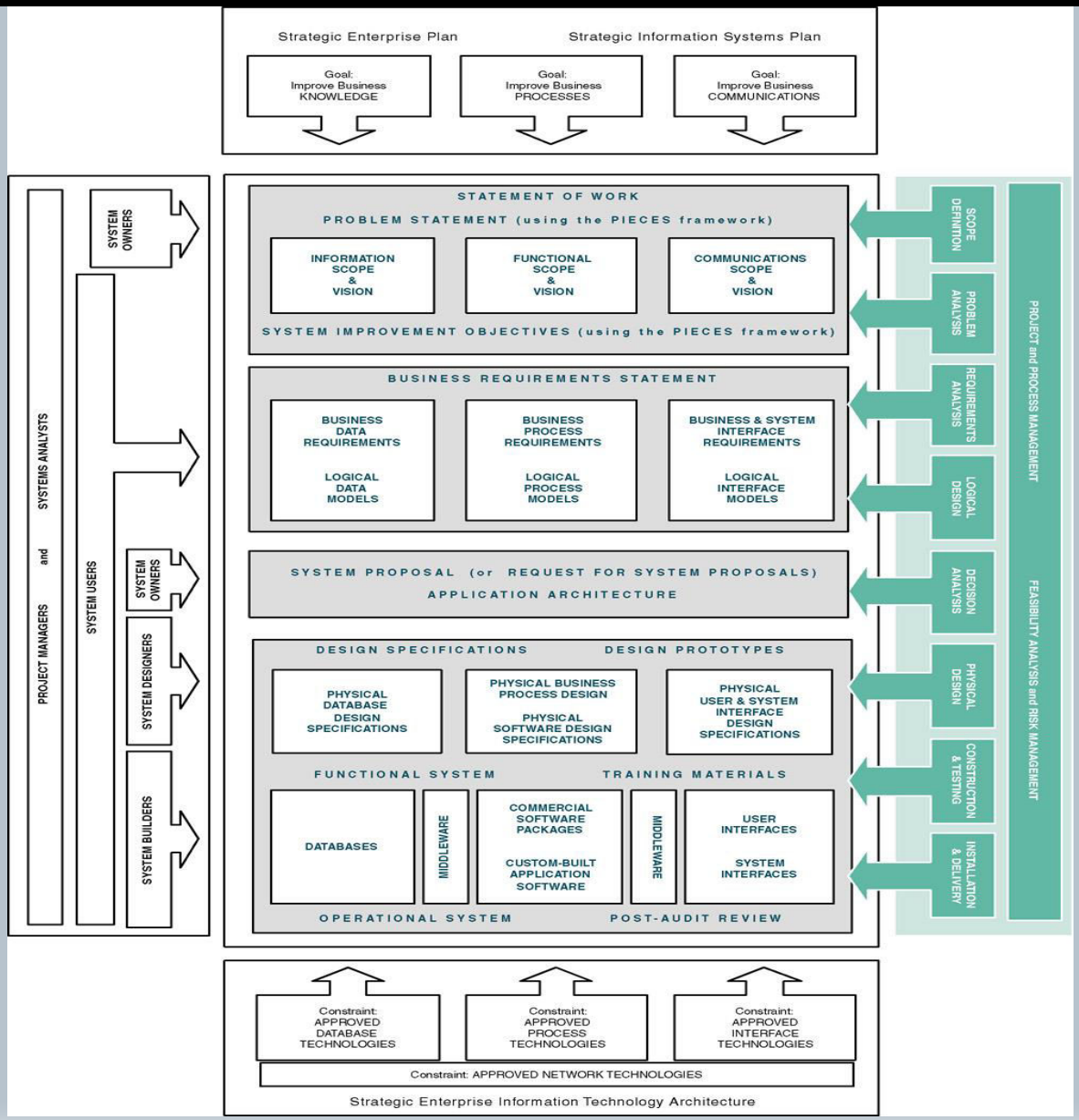
E the need to improve efficiency of people and processes

S the need to improve service to customers, suppliers, partners, employees, etc.

The Classic Project Phases (using FAST methodology)



Building Blocks View of System Development



Shows corresponding participants for each phase

Scope Definition

Two objectives

- Is this problem worth looking at?
- If the problem is worth looking at, establish the size and boundaries of the project, the project vision, any constraints or limitations, the required project participants, and, finally, the budget and schedule.

- Participants
  - systems owners, SA, and project managers
- Deliverables
  - A problem statement and an initial scope statement

Problem Analysis

Objective

Study the existing system and analyzes the findings to provide the project team with a more thorough understanding of the problems that triggered the project.

– Answer the most important question,"Will the benefits of solving these problems exceed the costs of building the system to solve these problems?"

– Participants
  - systems owners, SA, and project managers

– Deliverables
  - A system improvement objective
  - Ex) reduce the time between order processing and shipping by three days

Requirements Analysis

Objective

Define and prioritize the business requirement: find out what users need or want out of the new system.

- Perhaps the most important phase of systems development.

Errors and omissions in requirements analysis →

- result in user dissatisfaction with the final system and costly modifications.

– Participants

- system users, SA, and project managers

– Deliverables

- Business requirements statement

Logical Design

Objective

Translate business user requirements into a system model

- Develop a conceptual model
  - Process model
  - Data Model
  - interface model

  - Participants
    - SA (draw model)
    - System users (validate model)
    - Project managers
  - Deliverables
    - Logical system models and specifications

Decision Analysis

- Evaluate candidate solutions 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?
- Participants
  - system owners, SA, and project managers
- Deliverables
  - System proposal and application architecture

Physical Design & Integration

Objective

Translation of business user requirements into a system model that depicts a technical implementation of the users' business requirements.

Two 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.

- Participants
  - system desingers, system builders, SA, and project managers
- Deliverables
  - Design prototypes and physical specifications

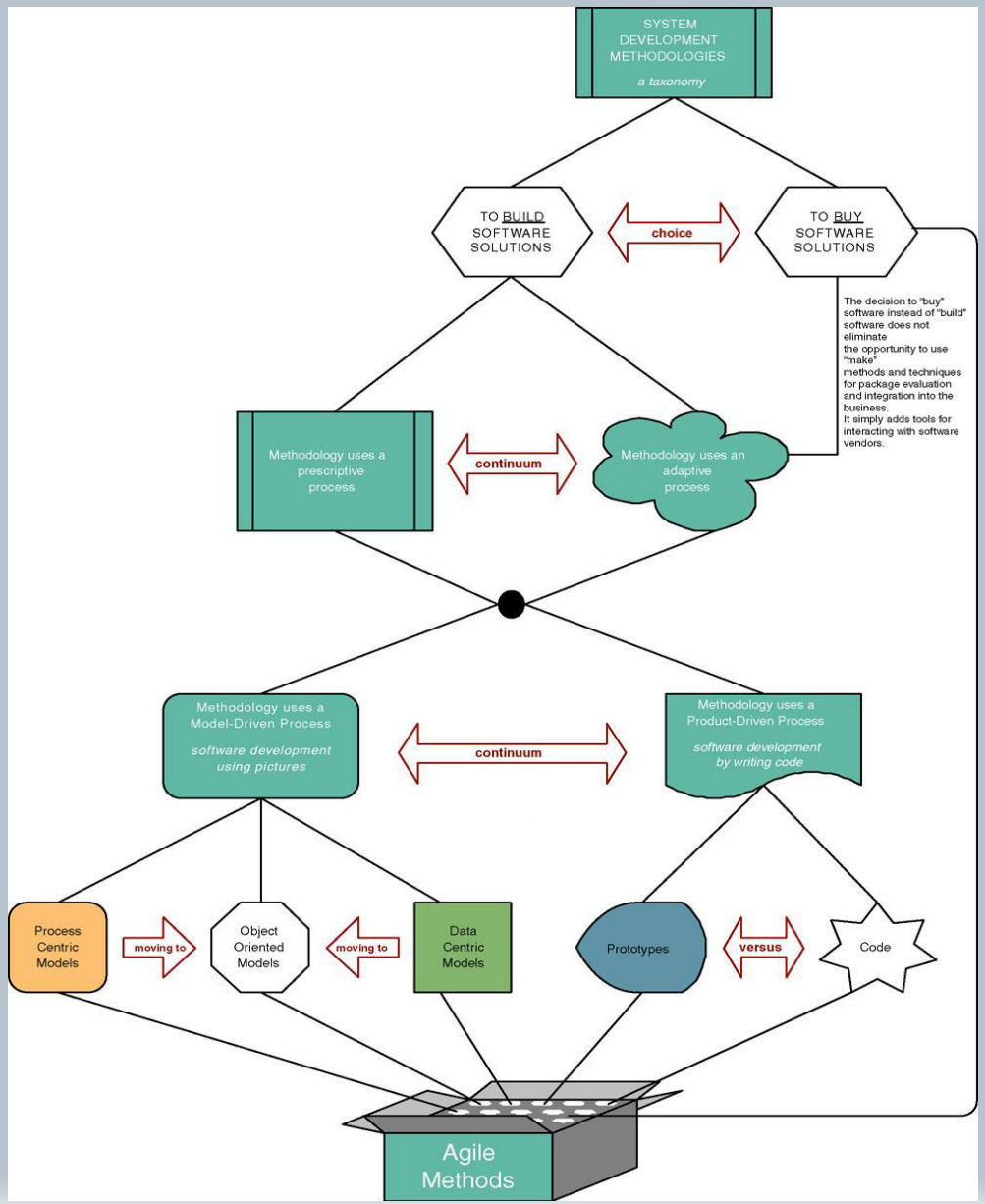
Rest of the development phases

- Construction and Testing
- Installation and Delivery
- System Operation and Maintenance

System Development Strategies (Will be discussed at Ch 5)

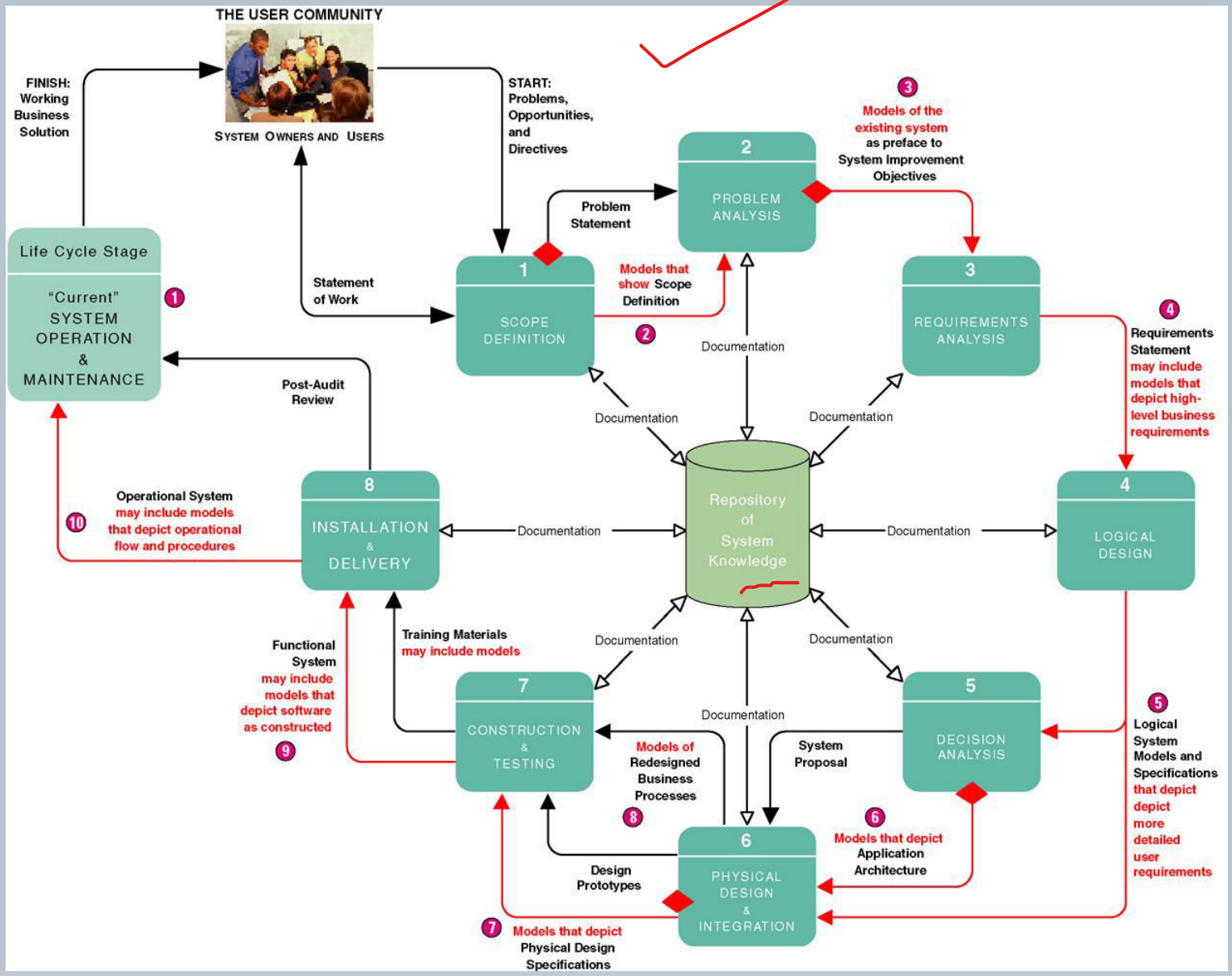
- A taxonomy for systems development methodologies and strategies (slide)
- **Model-driven Development Strategy** – a system development strategy that emphasizes the drawing of system models to help visualize and analyze problems, define business requirements, and design information systems.
  - Process modeling, Data modeling, and Object modeling
  - Often criticized for its time and effort intensity, model-driven development strategy work well with large/complex projects and structured domain problems.
  - Next

A Taxonomy for System Development Methodologies & Strategies



Back

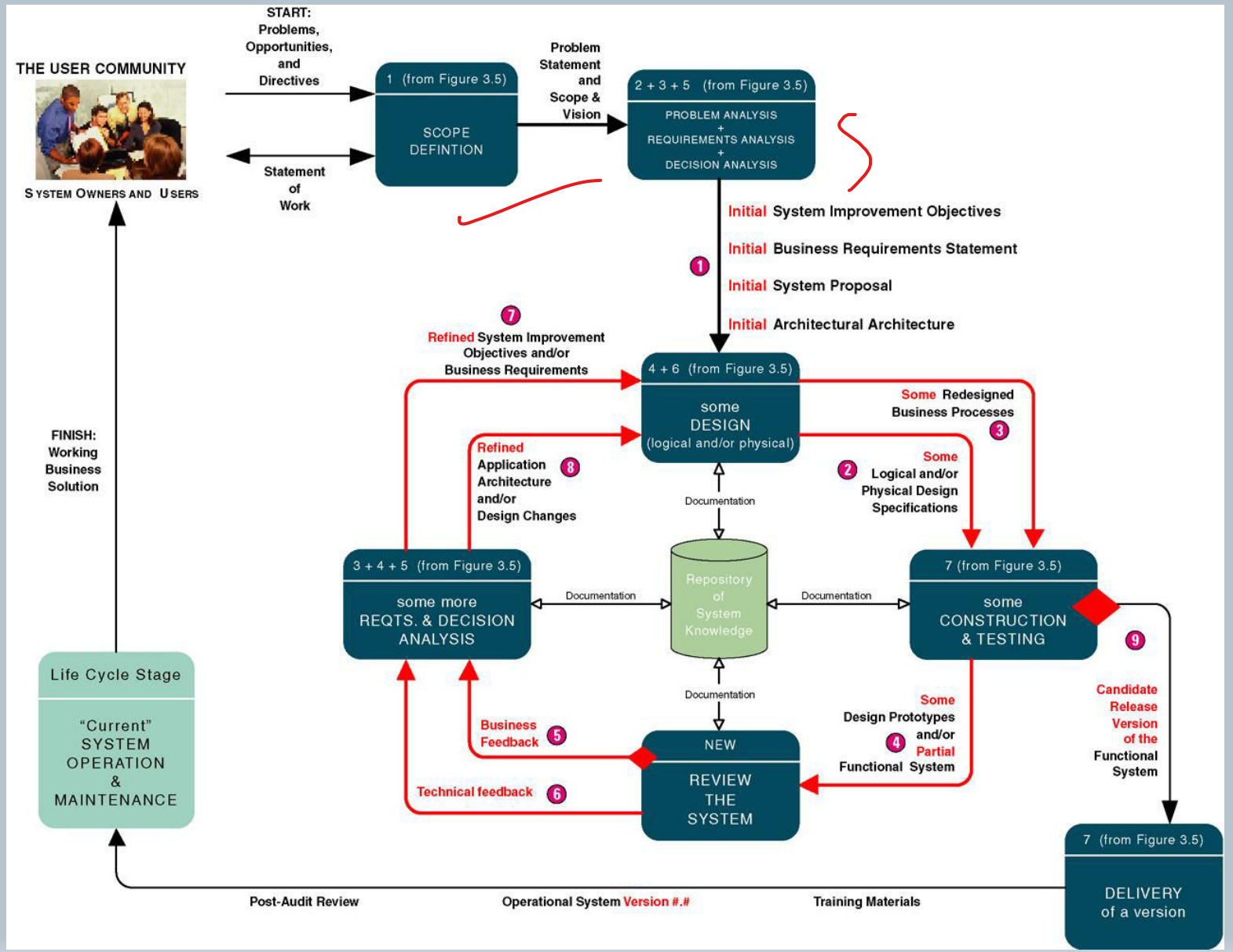
Model-Driven Development Strategy



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.
  - Suitable if domain is not well defined (semi or unstructured)
  - Useful if technical requirements are unclear.
  - Not appropriate for large and complex technical systems
  - Design problem due to rapid development

Rapid Application Development Strategy

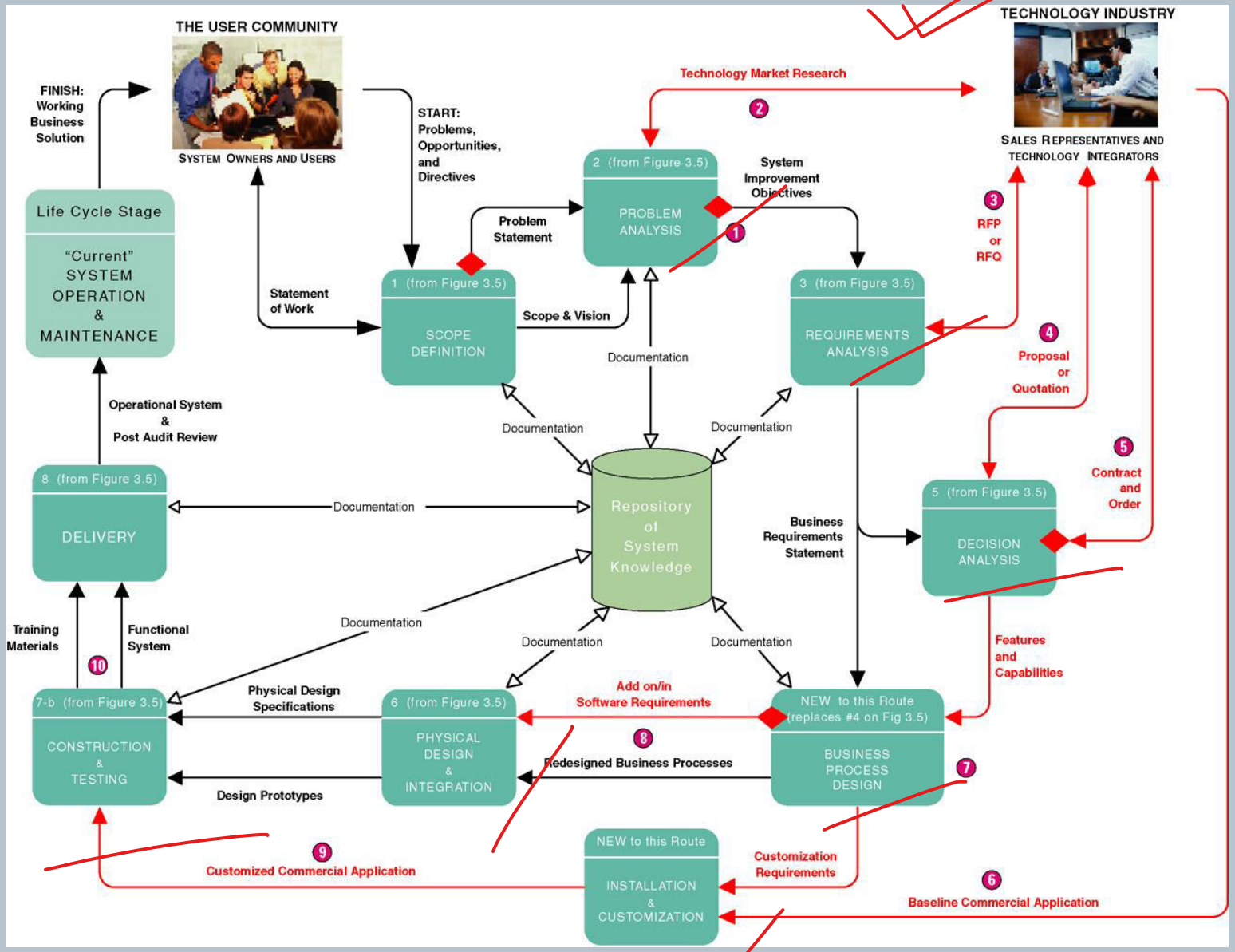


Commercial Application Package Implementation Strategy

- **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.
  - Save development costs
  - Existing capabilities are limited (e.g., short of technological ability and IS staff)
  - Systems that do not related with organization's secret
  - To learn about more advanced technologies

- ERP vendors provide their own implementation strategy.
  - Change of business processes after implementation

Commercial Application Package Implementation Strategy



Hybrid Strategies