西安电子科技大学
XIDIAN UNIVERSITY

03 系统分析与设计

（System Analysis and Design）

Information Systems Development

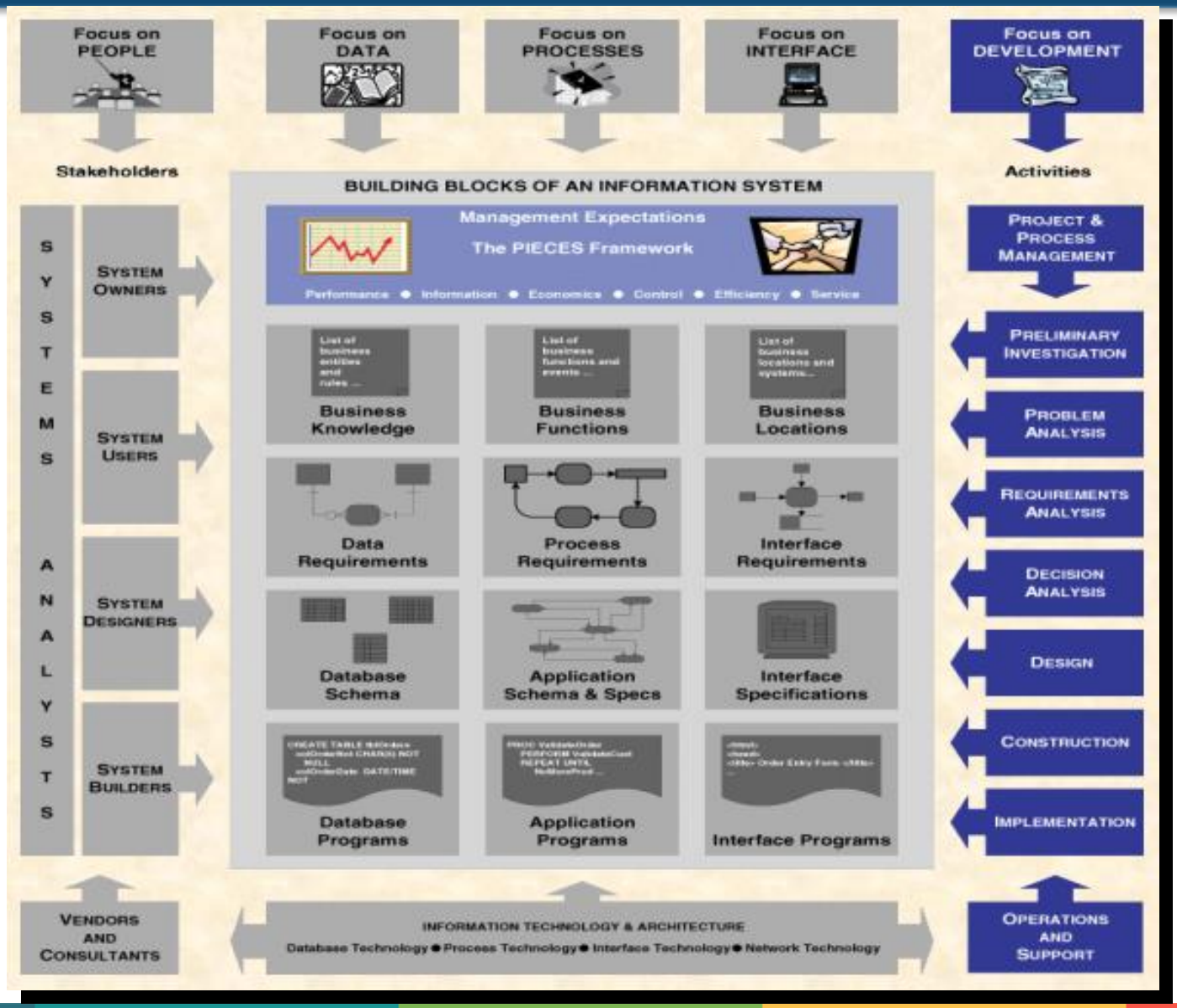
Content Structure

- The Story
  - 一次由技术人员参加的项目计划会议以启动项目开发过程。Bob初步了解了这个公司的开发中心的职能和这个公司所采用的开发方法论 – FAST。
  - 会议还体现出开发人员与机构之间的角色区别。
- Business analyst
- Development Center
- Model-driven approach
- System models

Content Structure

- **Process of Systems Development**
  - 系统开发过程的概念；对系统开发过程的管理和控制；系统生存周期；系统开发的方法论和原理。
- A **Systems Development Methodology**
  - 一种系统开发方法论 – FAST；一种问题求解框架 – PIECES；系统开发过程主要阶段的划分，以及一些跨阶段的开发活动。
- **Alternative Route and Methods**
  - 几种系统开发路线：模型驱动；快速原型；购置可用商品软件；混合路线；维护与再工程。
- Automated Tools and Technology
  - CASE： ADE工具；过程与项目管理工具。

Chapter Map


Focus on PEOPLE
Focus on DATA
Focus on PROCESSES
Focus on INTERFACE
Focus on DEVELOPMENT
Stakeholders
Activities
BUILDING BLOCKS OF AN INFORMATION SYSTEM
Management Expectations
The PIECES Framework
Performance • Information • Economics • Control • Efficiency • Service
SYSTEMS ANALYSTS
System Owners
System Users
System Designers
System Builders
Business Knowledge
Business Functions
Business Locations
Data Requirements
Process Requirements
Interface Requirements
Database Schema
Application Schema & Specs
Interface Specifications
Database Programs
Application Programs
Interface Programs
Project & Process Management
Preliminary Investigation
Problem Analysis
Requirements Analysis
Decision Analysis
Design
Construction
Implementation
Vendors and Consultants
INFORMATION TECHNOLOGY & ARCHITECTURE
Database Technology • Process Technology • Interface Technology • Network Technology
Operations and Support

The Process of Systems Development

Process of System Development

- A **System Development Process** is a set of activities, methods, best practices, deliverables, and automated tools that stakeholders (Chapter 1) use to develop and maintain information systems and software.

The CMM Process Management Model

- The **Capability Maturity Model** (CMM) is a framework to assess（评定） the maturity level of an organization's information system development and management processes and products. It consists of five levels of maturity as measured by a set of guidelines called the key process areas（关键过程领域）.

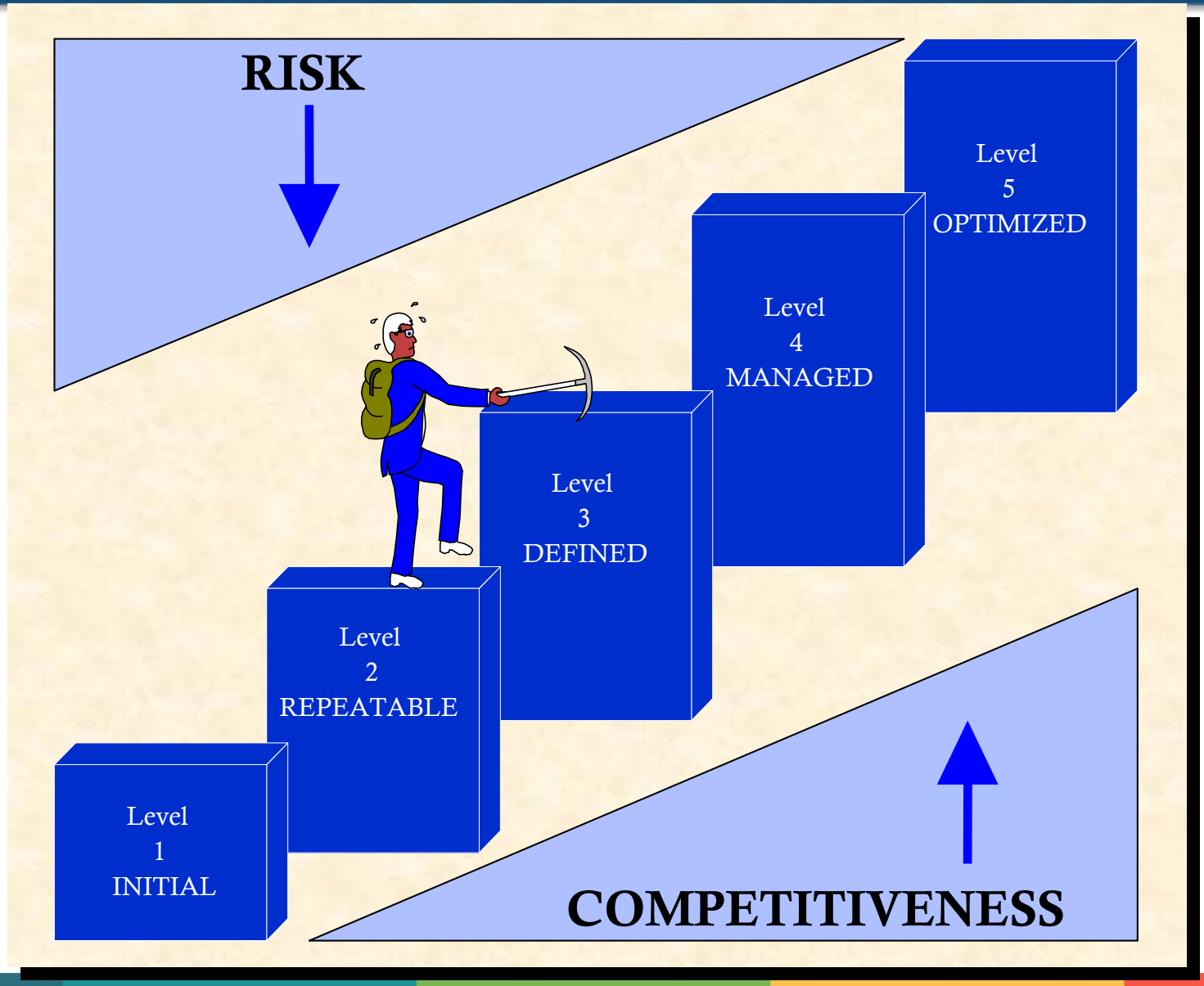
The CMM Process Management Model

- **Five levels of maturity:**
  - Level 1—Initial（初始级）: System development projects follow no prescribed process.
  - Level2—Repeatable（可重复级）: Project management processes and practices are established to track project costs, schedules, and functionality.
  - Level 3—Defined（已定义级）: A standard system development process (sometimes called a “methodology”) is purchased or developed, and integrated throughout the information systems/services unit of the organization. The process is stable, predictable and repeatable.

The CMM Process Management Model

- **Five levels of maturity:**
  - Level 4—Managed（已管理级）: Measurable goals for quality and productivity are established.
  - Level 5—Optimizing（优化级）: The standardized system development process is continuously monitored and improved based on measures and data analysis established in Level 4.

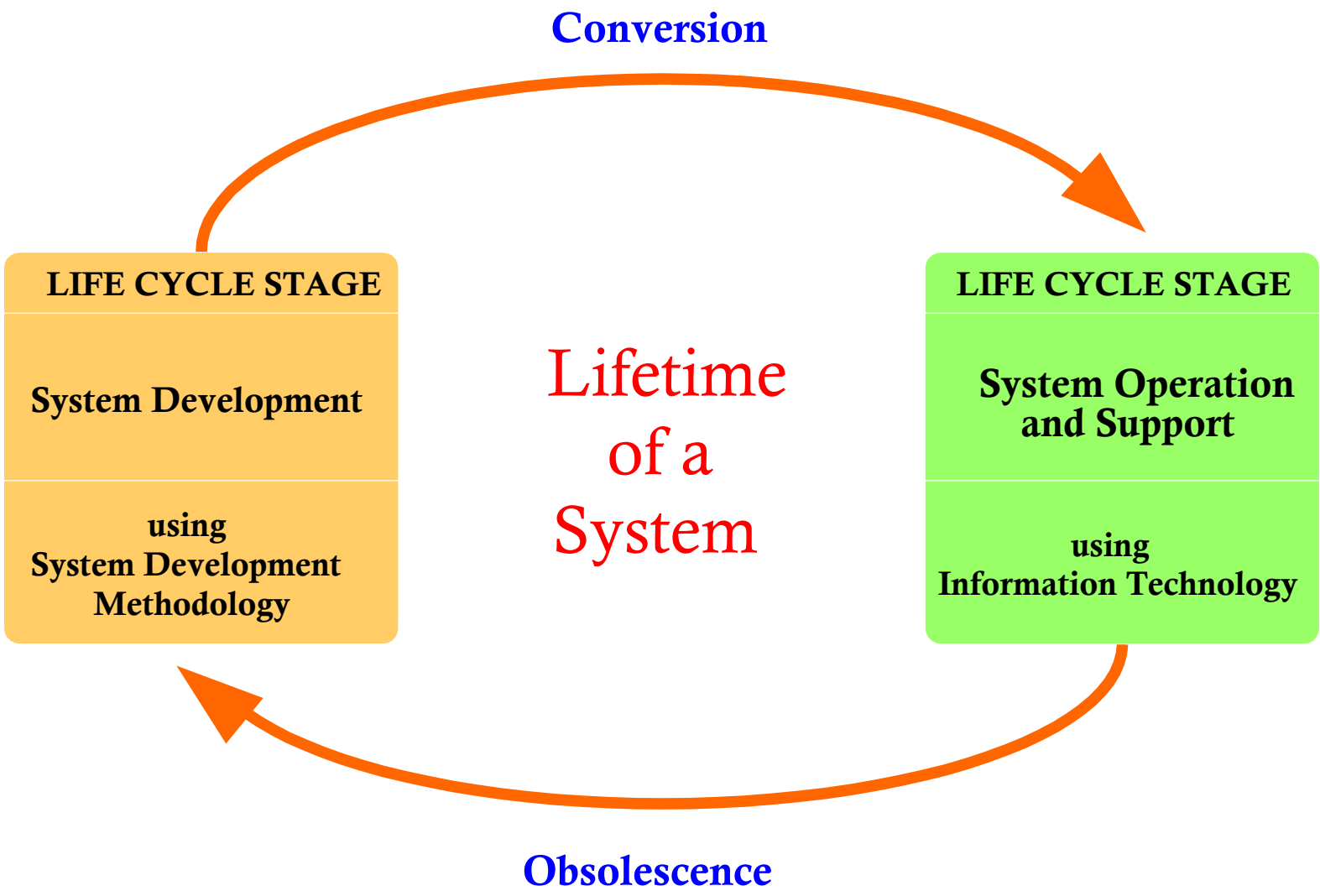
Capability Maturity Model (CMM)



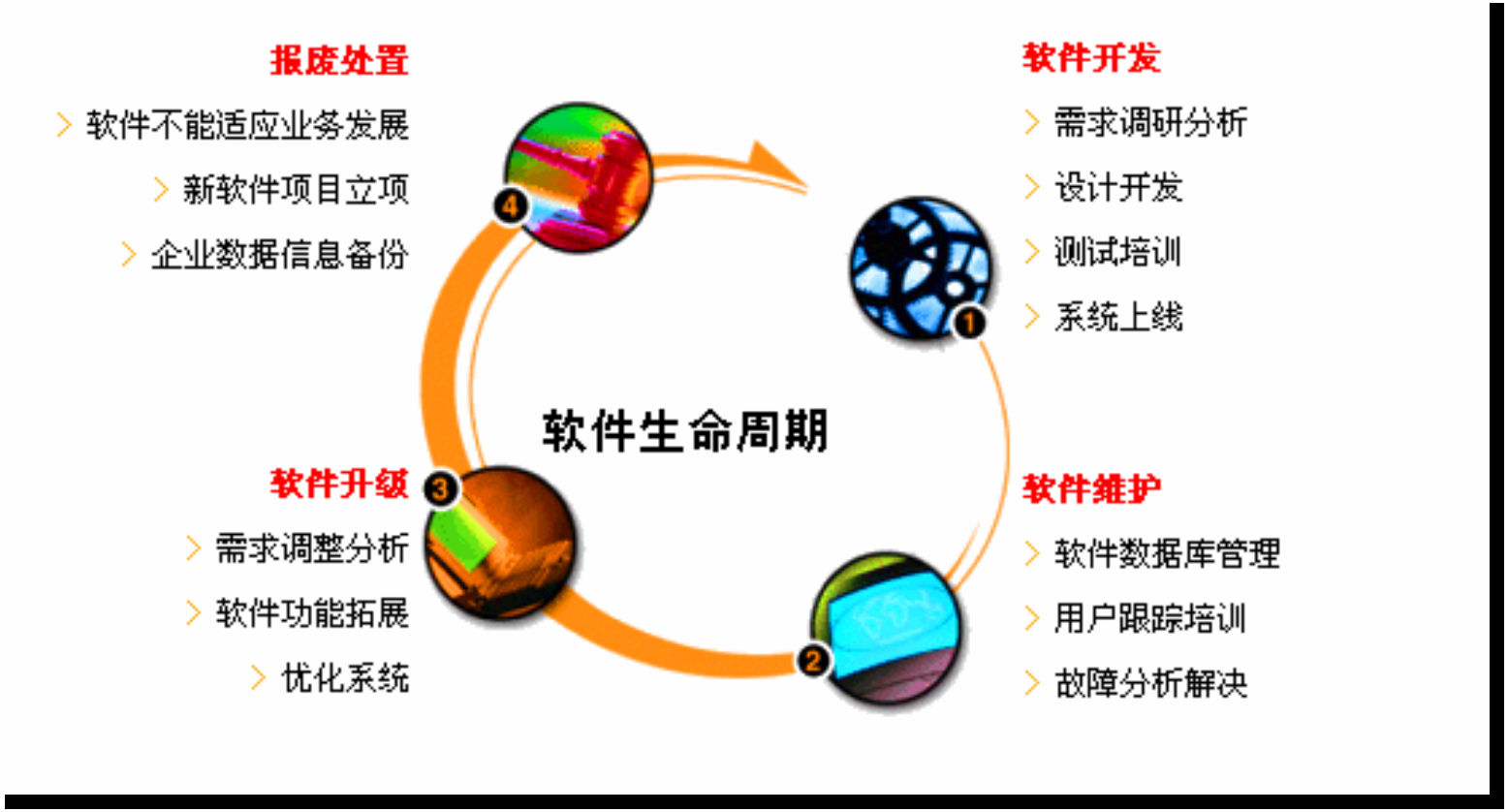
Life Cycle versus Methodology

- A **System Life Cycle**（系统生命周期） divides the life of an information system into two stages, systems development and systems operation and support.
- A **System Development Methodology**（系统开发方法论） is a very formal and precise system 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 maintain information systems and software.

A System Life Cycle



A Software Life Cycle



Principles of System Development

- Get the owners and users involved. (以人为本)
- Use a problem-solving approach. (有理有据)
- Establish phases and activities. (分步实施)
- Establish standards. (建立标准)
- Justify systems as capital investments. (最优决策)
- Don’t be afraid to cancel or revise scope. (悬崖勒马)
- Divide and conquer. (分而治之)
- Design systems for growth and change. (高瞻远瞩)

Principles of System Development - 1

- **Get the owners and users involved.**
  - Although analysts and programmers work hard to create technologically impressive solutions, those solutions often backfire（适得其反） because they don't address the real organization problems or they introduce new problems.
  - For this reason, system owner and user involvement is necessary for successful systems development.

Principles of System Development - 2

- **Use a problem-solving approach.**
  - The classic problem-solving approach:

    1. Study and understand the problem and its context.
    2. Define the requirements of a suitable solution.
    3. Identify candidate solution and select the "best" solution.
    4. Design and/or implement the solution.
    5. Observe and evaluate the solution's impact, and refine the solution accordingly.

Principles of System Development - 2

- **Use a problem-solving approach.**
  - Inexperienced problem solvers tend to eliminate or abbreviate one or more of the above steps. The result can range from

    (1) solving the wrong problem, to

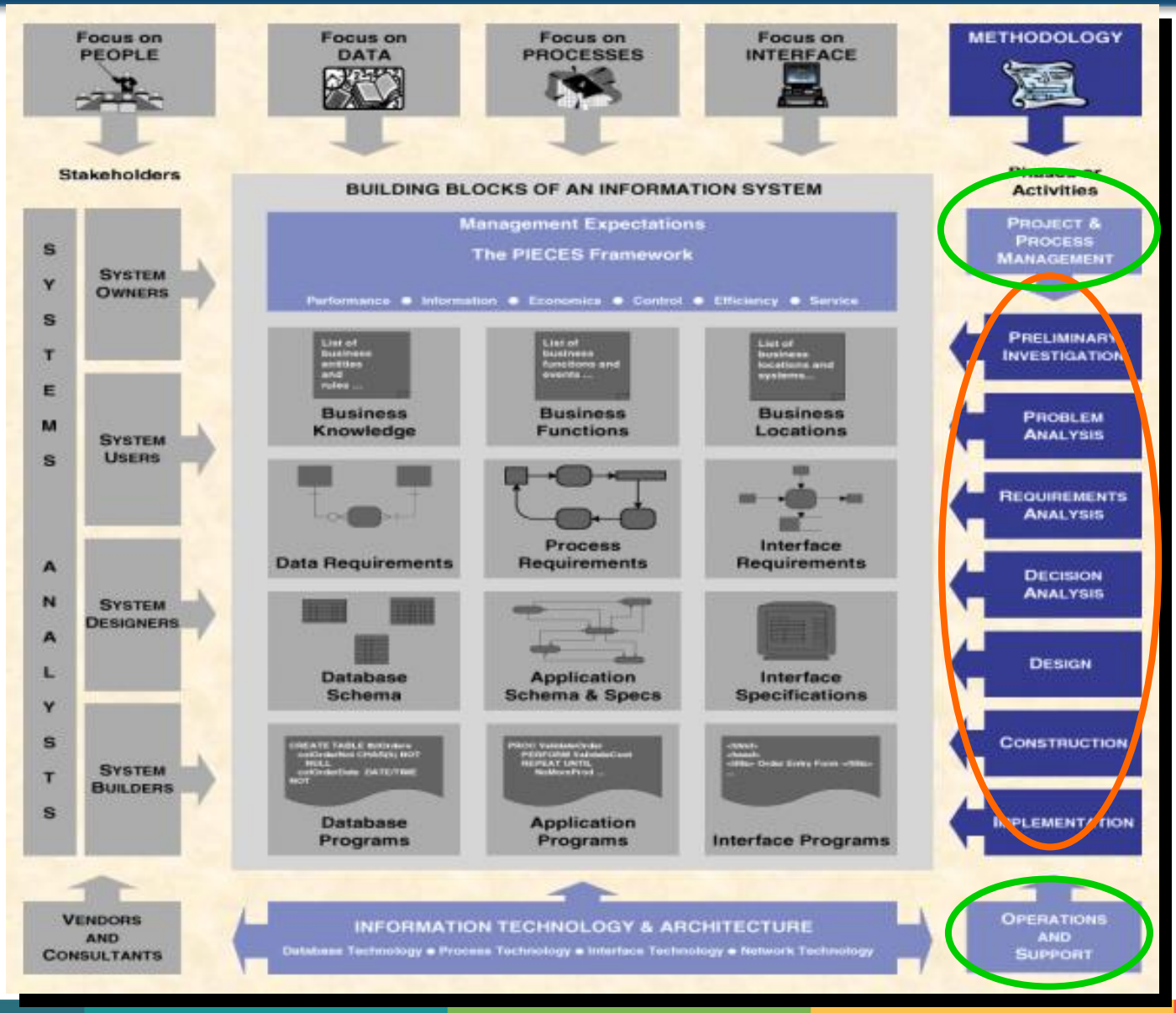
    (2) incorrectly solving the problem, to

    (3) picking the wrong solution.
  - A methodology's problem-solving orientation can reduce or eliminate the above risks.

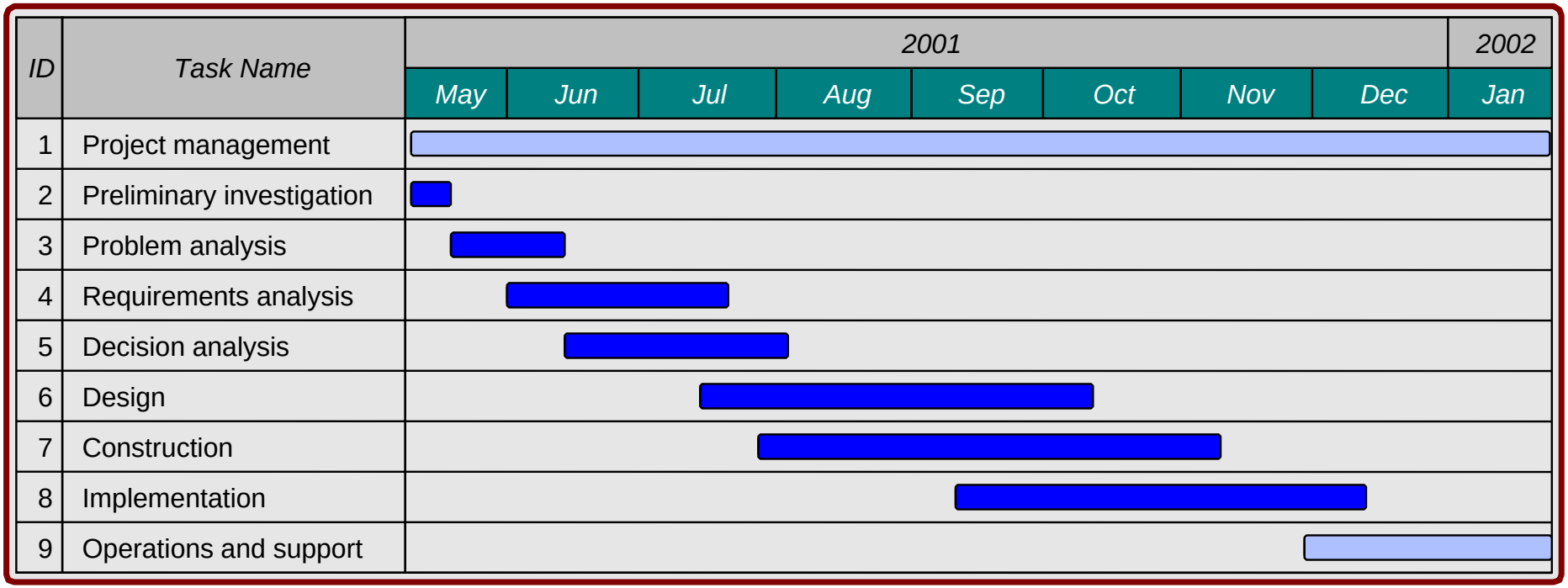
Principles of System Development - 3

- **Establish phases and activities.**
  - Phases:
    - Preliminary investigation
    - Problem analysis
    - Requirements analysis
    - Decision analysis
    - Design
    - Construction
    - Implementation
  - Activities:
    - Project and process management
    - Operations and Support

Principles of System Development - 3



Principles of System Development - 3



Principles of System Development - 4

- **Establish standards.**
  - An organization should embrace standards for both information systems and the process used to develop those systems（包括信息系统和开发过程的标准）.
  - Standards should minimally encompass the following
    - Documentation
    - Quality
    - Automated tools
    - Information technology

Principles of System Development - 5

- **Justify systems as capital investments.**
  - For any problem, there are likely to be several possible solutions. The analyst (or users) should not necessarily accept the first solution that comes to mind.
  - After identifying alternative solutions, the systems analyst should evaluate each possible solution for feasibility, especially for cost-effectiveness and risk management.

Principles of System Development - 6

- **Don't be afraid to cancel or revise scope.**
  - There is often a temptation（诱惑） to continue with a project only because of the investment already made. In the long run, canceled projects are less costly than implemented disasters!
  - Using the creeping commitment approach（分步提交法）, multiple feasibility checkpoints（可行性检查点） are built into any systems development methodology. At each checkpoint, the analyst should consider the following options:
    - Cancel the project if it is no longer feasible. Reevaluate and adjust the costs and schedule if project scope is to be increased. Reduce the scope if the project budget and schedule are frozen and not sufficient to cover all project objectives.

Principles of System Development - 7

- **Divide and conquer.**
  - "If you want to learn anything, you must not try to learn everything – at least not all at once."
  - By dividing a larger problem (system) into more easily managed pieces (subsystems), the analyst can simplify the problem-solving process.
  - The divide and conquer approach also complements communication and project management by allowing different pieces of the system to be delegated to different stakeholders.

Principles of System Development - 8

- **Design systems for growth and change.**
  - 热力学第二定律在原理上已经陈述：一个封闭系统的无序状态不可能减少，而只能是增长，或者可能保持不变。这种无序状态的一种度量标准是熵（entropy）。关于软件的定律（Lehman）：（1）一个有用的程序总会要被修改。（2）倘若没有采取实际的防范措施，在修改一个程序时就将增加这个程序的复杂性。
  - 一个系统的生存跨度取决于这个系统初始时构造的好坏（即初始熵是小还是大）。一旦达到了特定的熵，继续使用该系统在经济上就不再合理，因为这将带来不合理的修改代价。

Principles of System Development - 8

- **Design systems for growth and change.**
  - The system entropy can be managed. Today's tools and techniques make it possible to design systems that can grow and change as requirements grow and change.
  - It's more important to recognize that flexibility and adaptability do not happen by accident—they must be built into a system.

A Systems Development Methodology

Project Identification and Initiation

- **Problems** are undesirable situations that prevent the organization from fully achieving its purpose, goals, and/or objectives.
- **Opportunities** are chances to improve the organization even in the absence of specific problems.
- **Directives** are new requirements that are imposed（施加） by management, government, or some external influence.

The PIECES Problem-Solving Framework

P the need to improve performance（性能）

I the need to improve information（信息） (and data)

E the need to improve economics（经济性）, control costs, or increase profits

C the need to improve control（可控制程度） or security

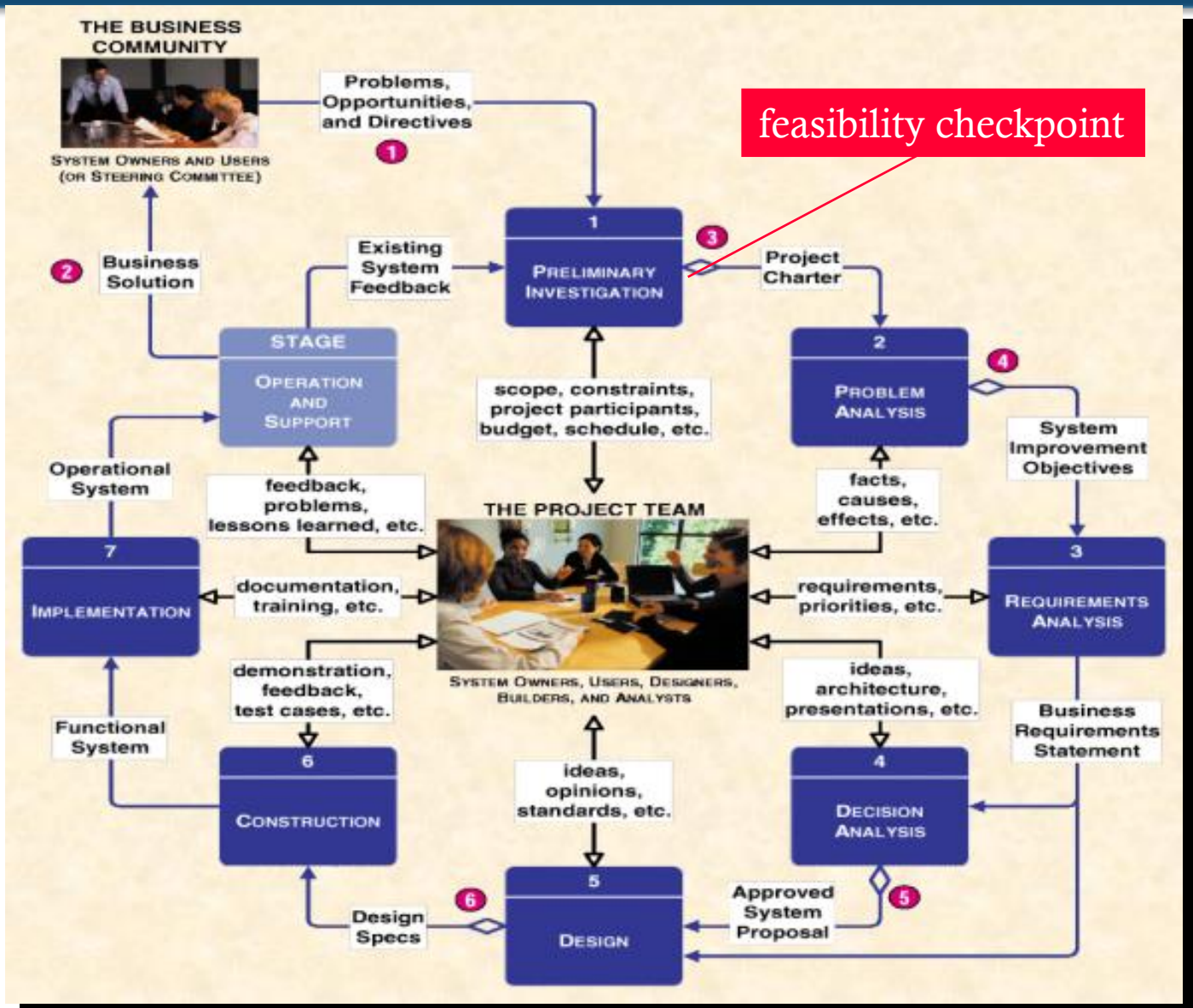
E the need to improve efficiency（效率） of people and processes

S the need to improve service（服务质量） to customers, suppliers, partners, employees, etc.

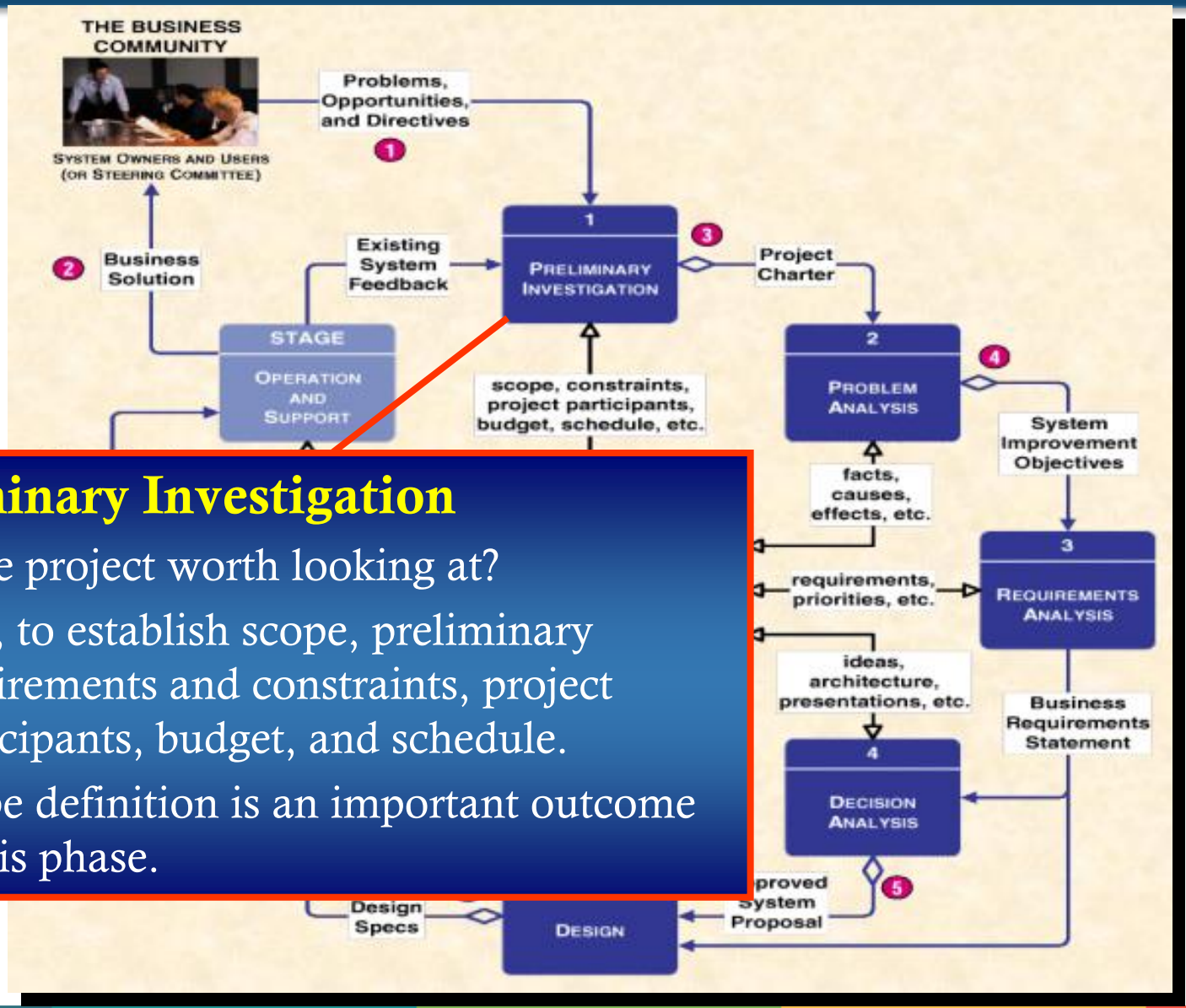
FAST

- **F**ramework for the **A**pplication of **S**ystems **T**echniques
- Not means to develop a system quickly or to use the prototyping approach only.

FAST System Development phases



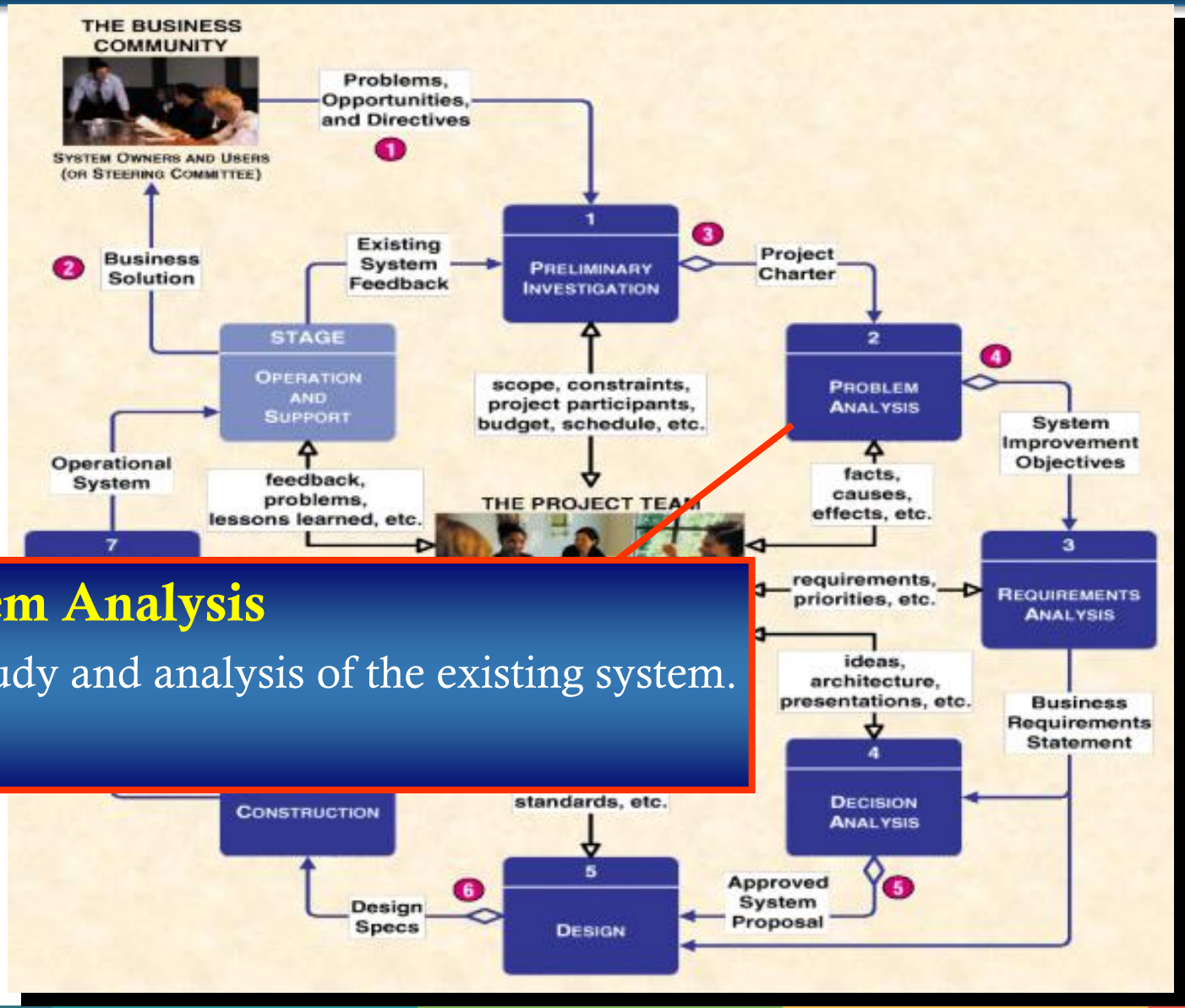
FAST System Development phases



Preliminary Investigation

- Is the project worth looking at?
- If so, to establish scope, preliminary requirements and constraints, project participants, budget, and schedule.
- Scope definition is an important outcome of this phase.

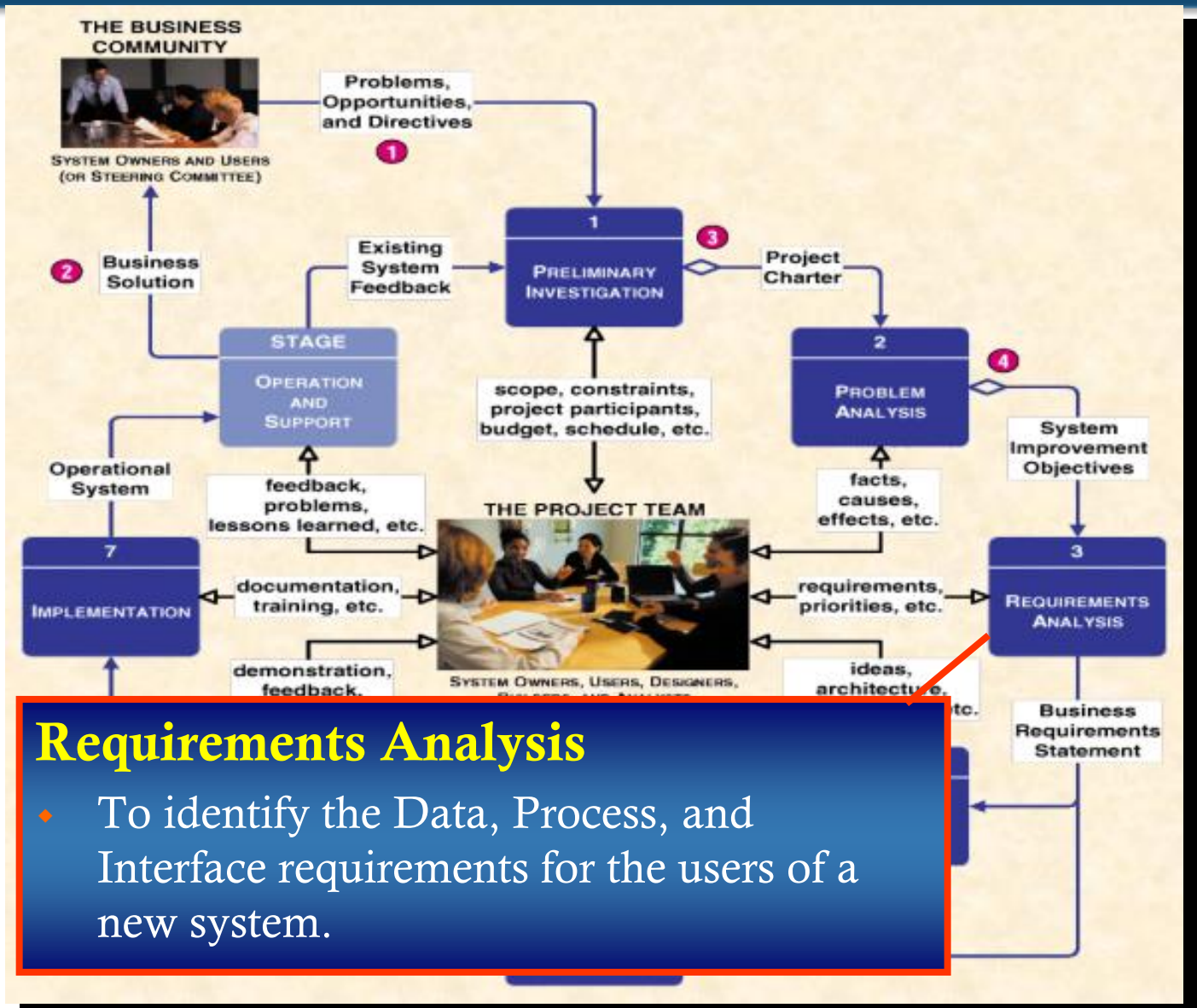
FAST System Development phases



Problem Analysis

- A study and analysis of the existing system.

FAST System Development phases



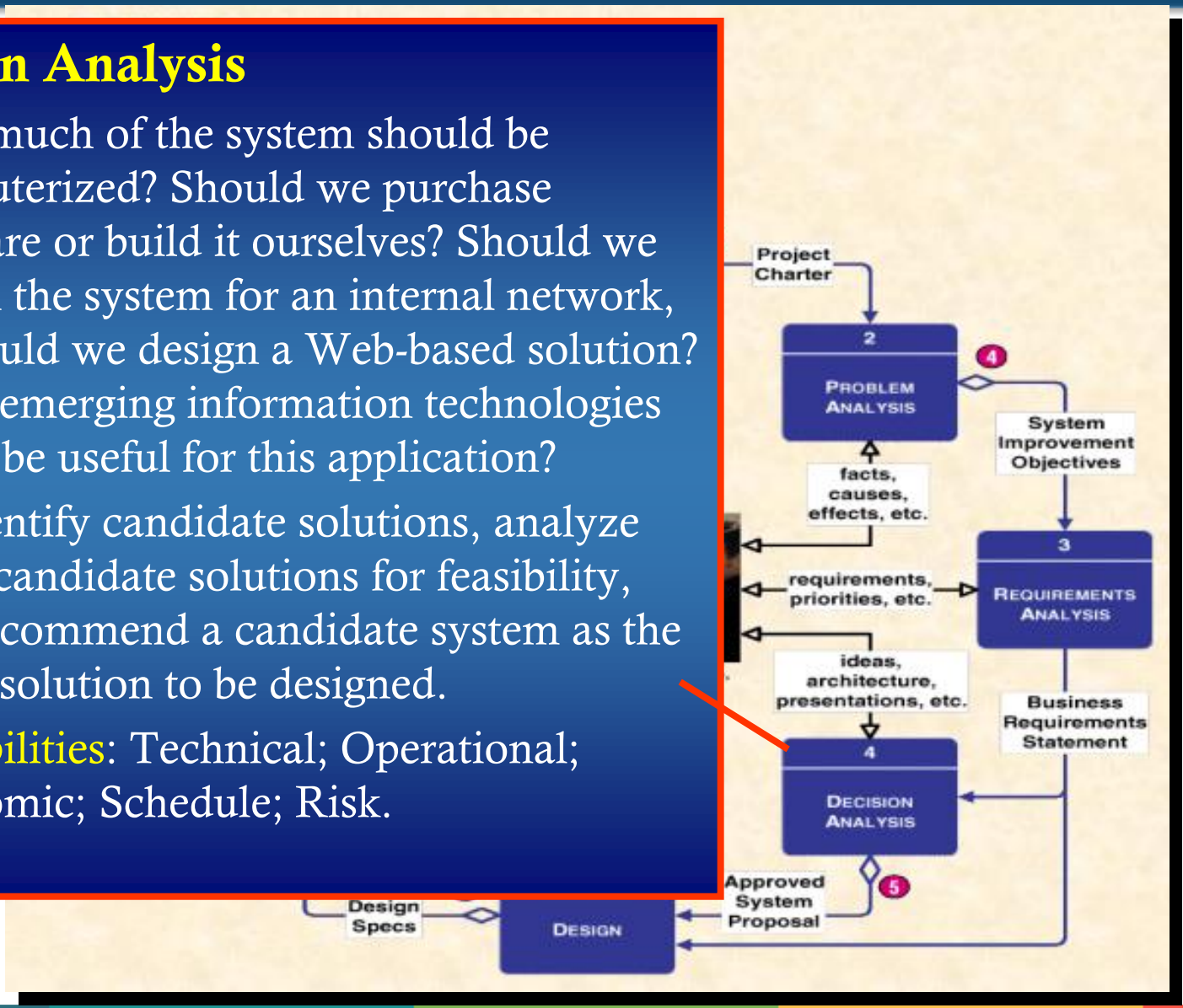
Requirements Analysis

- To identify the Data, Process, and Interface requirements for the users of a new system.

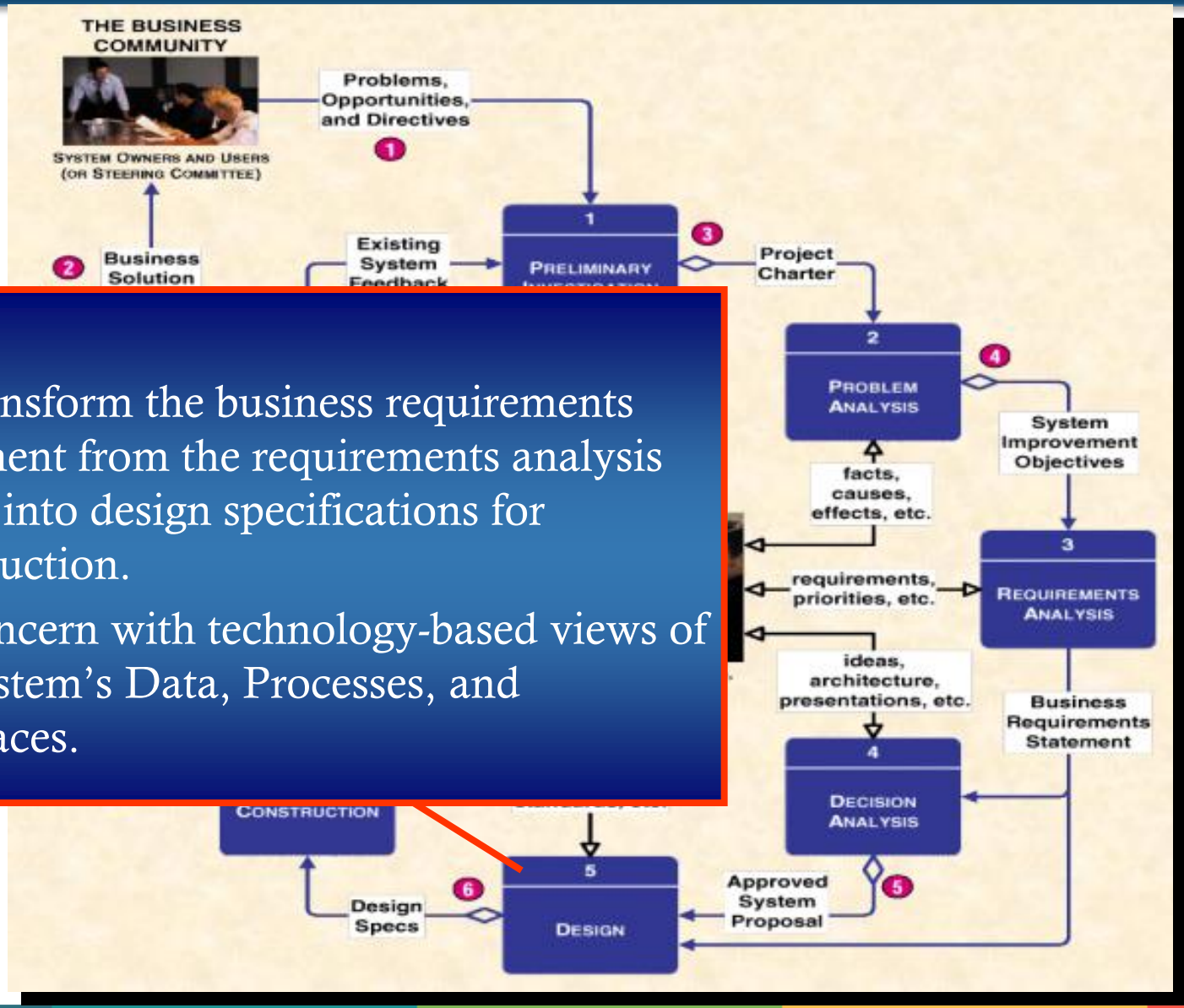
FAST System Development phases

Decision Analysis

- How much of the system should be computerized? Should we purchase software or build it ourselves? Should we design the system for an internal network, or should we design a Web-based solution? What emerging information technologies might be useful for this application?
- To identify candidate solutions, analyze those candidate solutions for feasibility, and recommend a candidate system as the target solution to be designed.
- Feasibilities: Technical; Operational; Economic; Schedule; Risk.



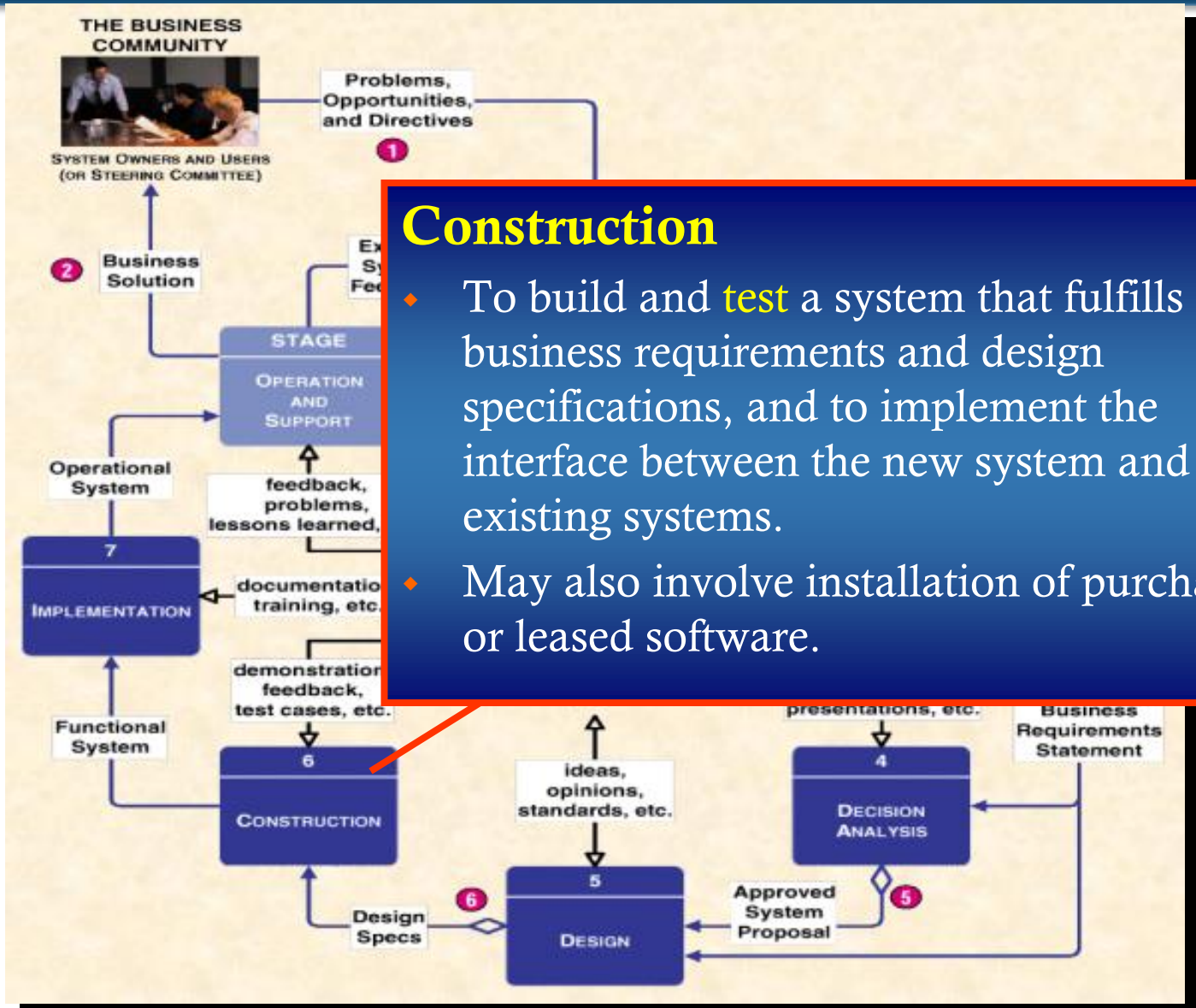
FAST System Development phases



Design

- To transform the business requirements statement from the requirements analysis phase into design specifications for construction.
- To concern with technology-based views of the system's Data, Processes, and Interfaces.

FAST System Development phases



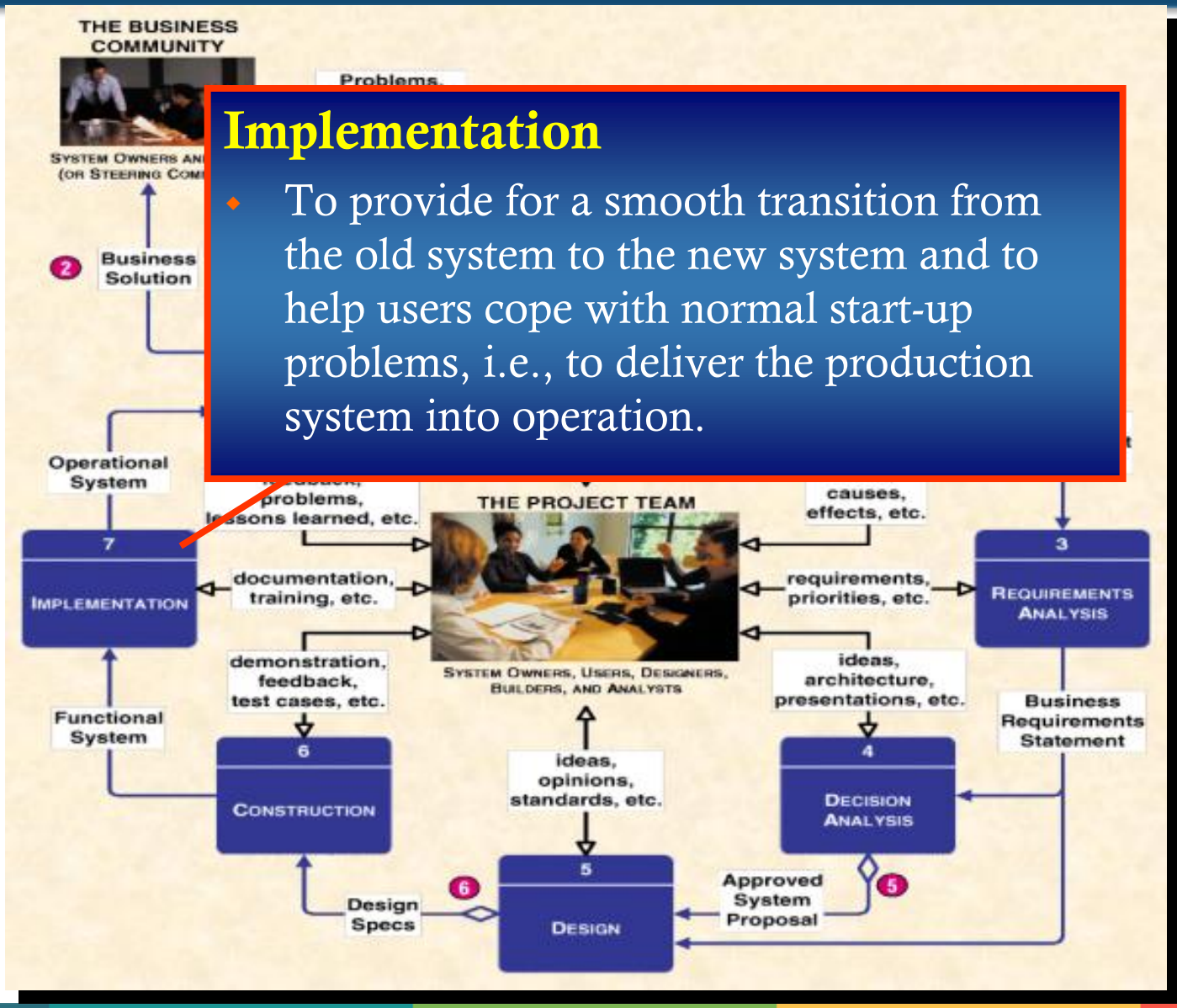
Construction

- To build and test a system that fulfills business requirements and design specifications, and to implement the interface between the new system and existing systems.
- May also involve installation of purchased or leased software.

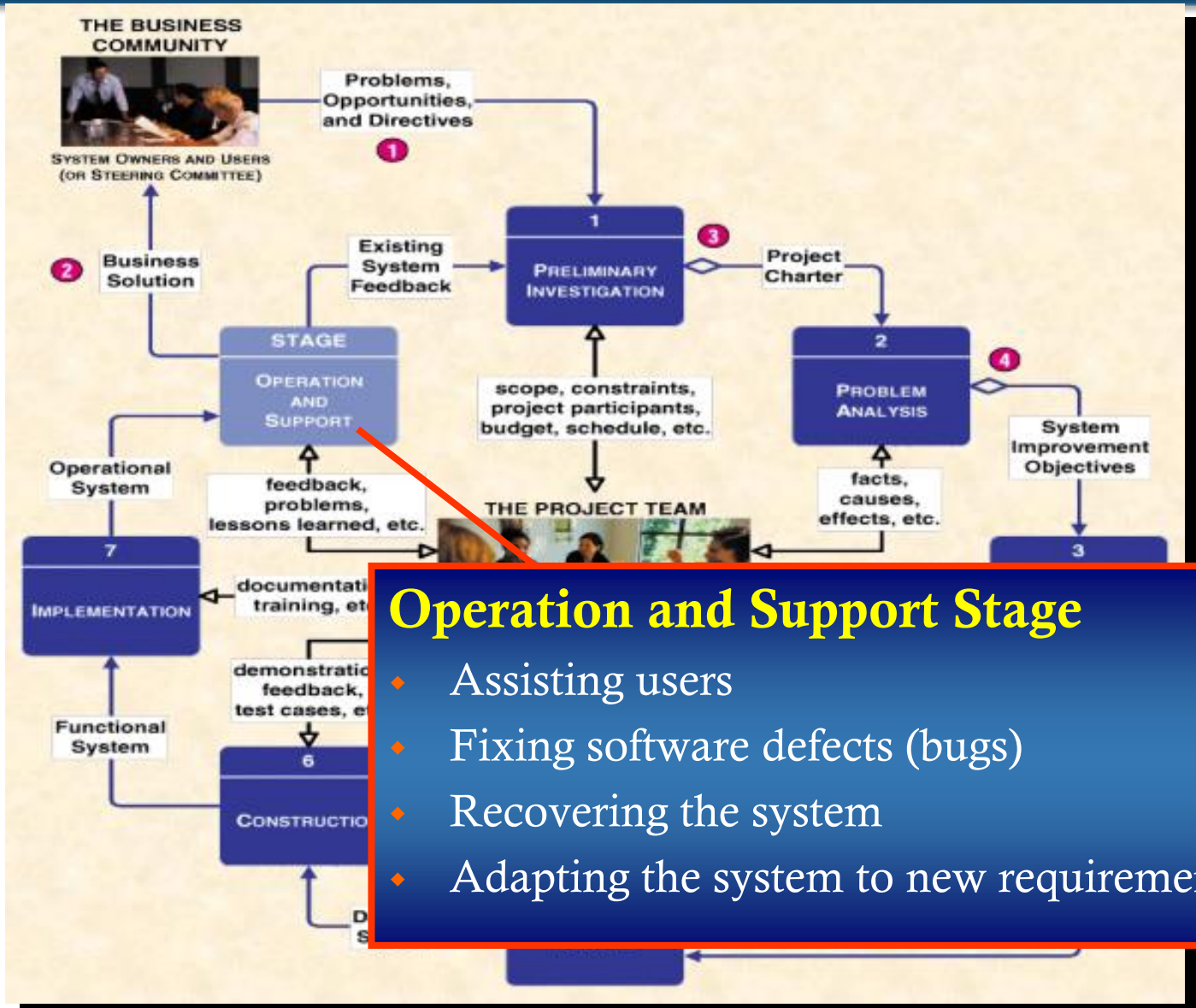
FAST System Development phases

Implementation

- To provide for a smooth transition from the old system to the new system and to help users cope with normal start-up problems, i.e., to deliver the production system into operation.



FAST System Development phases



Operation and Support Stage

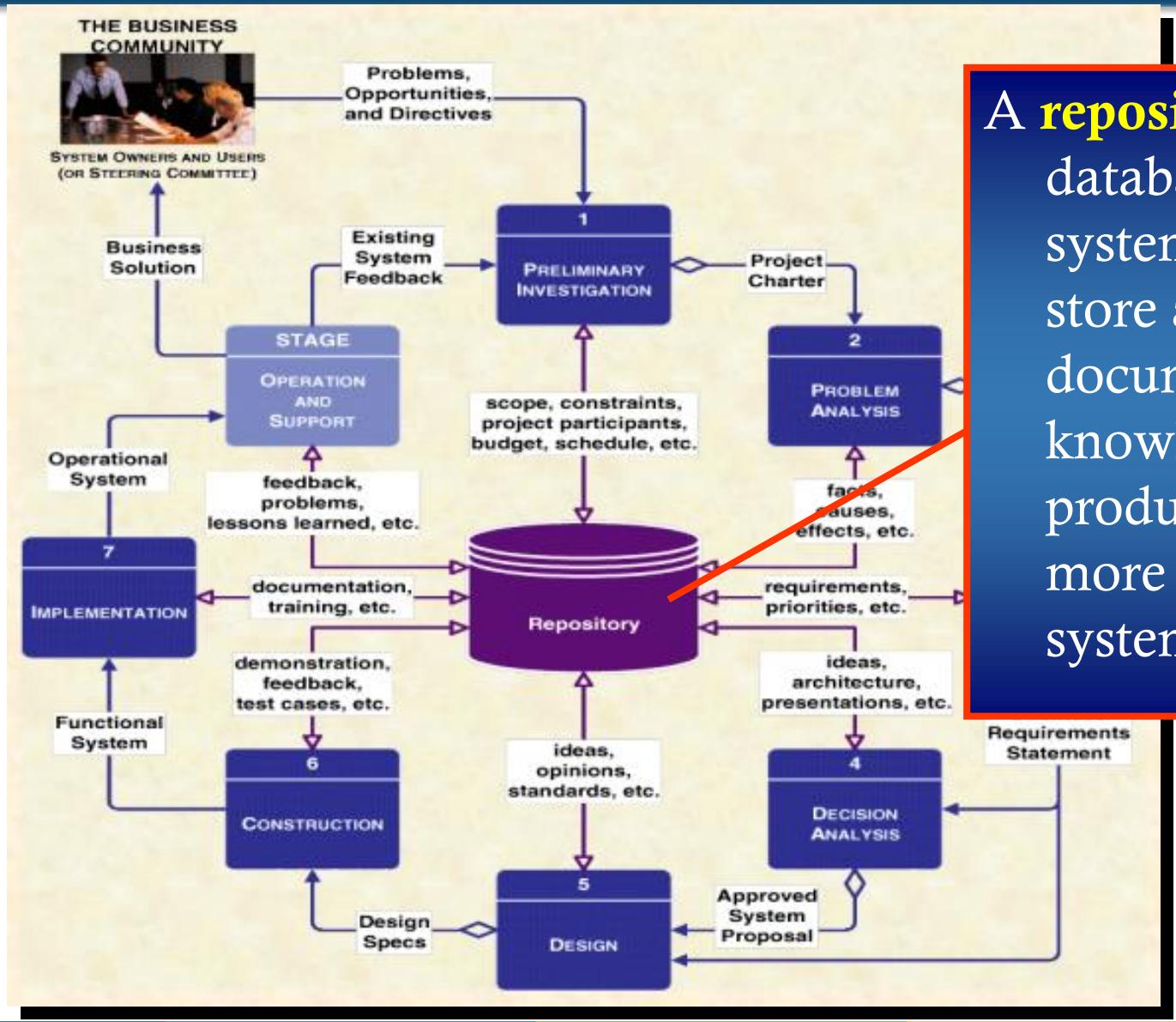
- Assisting users
- Fixing software defects (bugs)
- Recovering the system
- Adapting the system to new requirements

Cross Life Cycle Activities

Cross life cycle activities are activities that overlap many or all phases of the methodology.

- Fact-finding (Information gathering or data collection)
- Documentation and presentation
- Feasibility analysis
- Process and project management

Sharing Knowledge via a Repository



A **repository** is a database where system developers store all documentation, knowledge, and products for one or more information systems or projects.

Alternative Routes and Methods

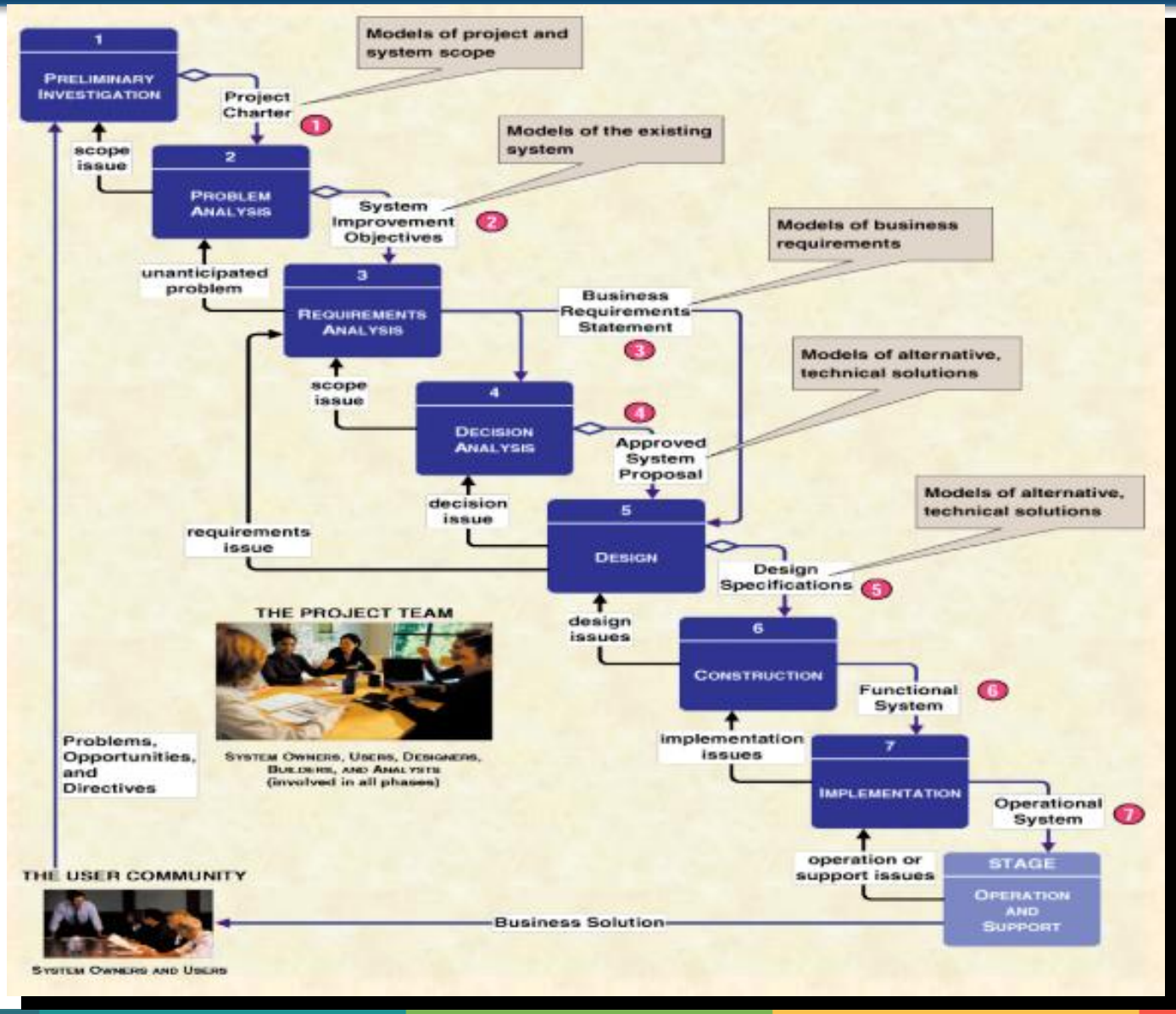
Alternative Routes through a Methodology

- Model-Driven Development (MDD)（模型驱动开发）
- Rapid Application Development (RAD)（快速原型开发）
- Commercial Off-the-Shelf Software (COTS) （购置商业软件）
- Maintenance and Reengineering（维护与再工程） or hybrids of the above

Model-Driven Development Route

- **Modeling** is the act of drawing one or more graphical representations (or pictures) of a system. Modeling is a communication technique based upon the old saying, "a picture is worth a thousand words."
- **Model-Driven Development** (MDD) techniques emphasize the drawing of models to help visualize and analyze problems, define business requirements, and design information systems.
  - Structured systems analysis and design — process-centered
  - Information engineering (IE) — data-centered, process-sensitive
  - Object-oriented analysis and design (OOAD) — object-centered (integration of data and process concerns)

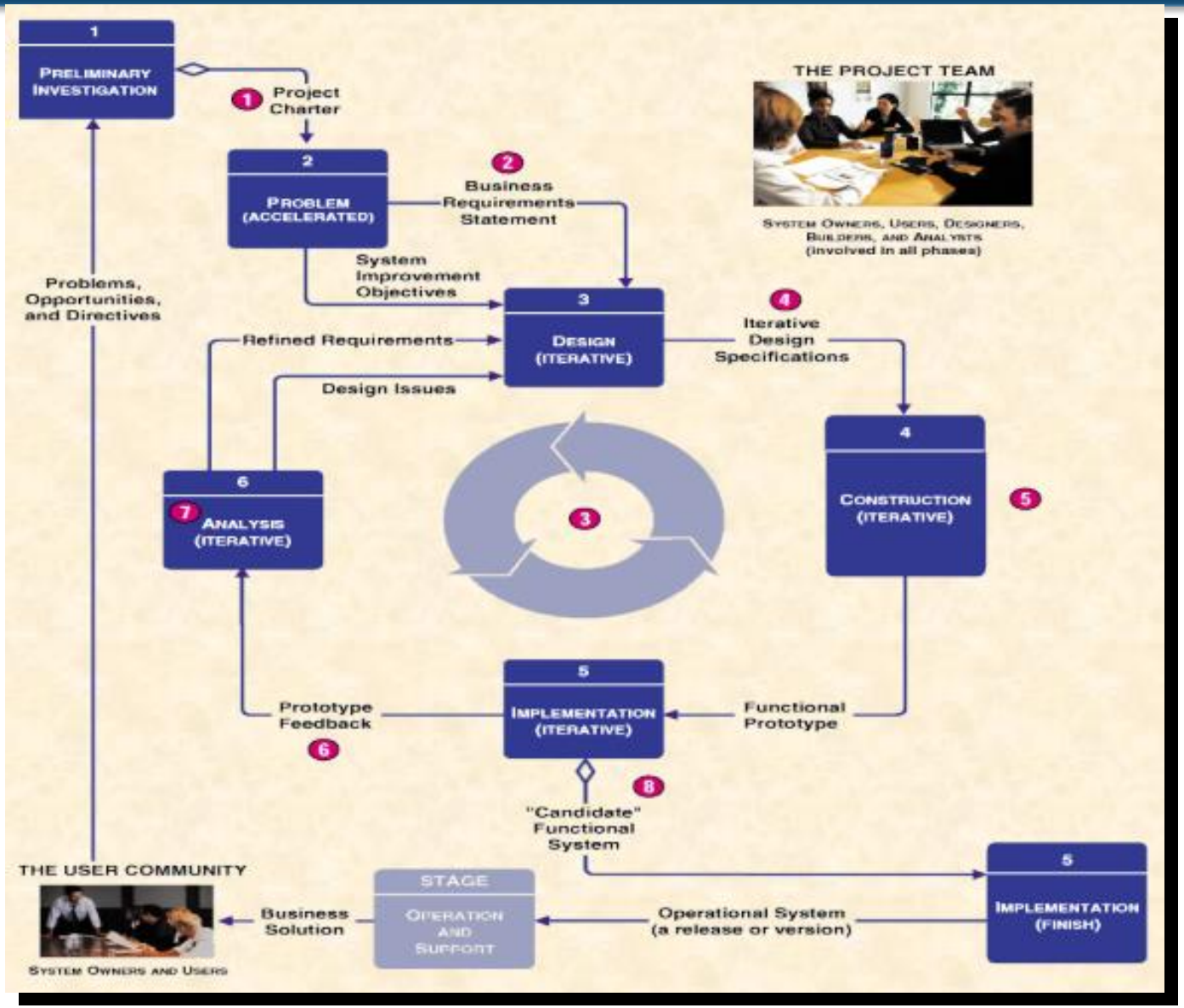
Model-Driven Development (MDD) Route



Rapid Application Development Route

- **Rapid Application Development** (RAD) techniques emphasize extensive user involvement in the rapid and evolutionary construction of working prototypes of a system to accelerate the system development process. RAD is sometimes called a spiral(螺旋线) approach.
- RAD is based on building **prototypes** that evolve into finished systems (often using **timeboxing** to limit the duration of the prototyping loop)

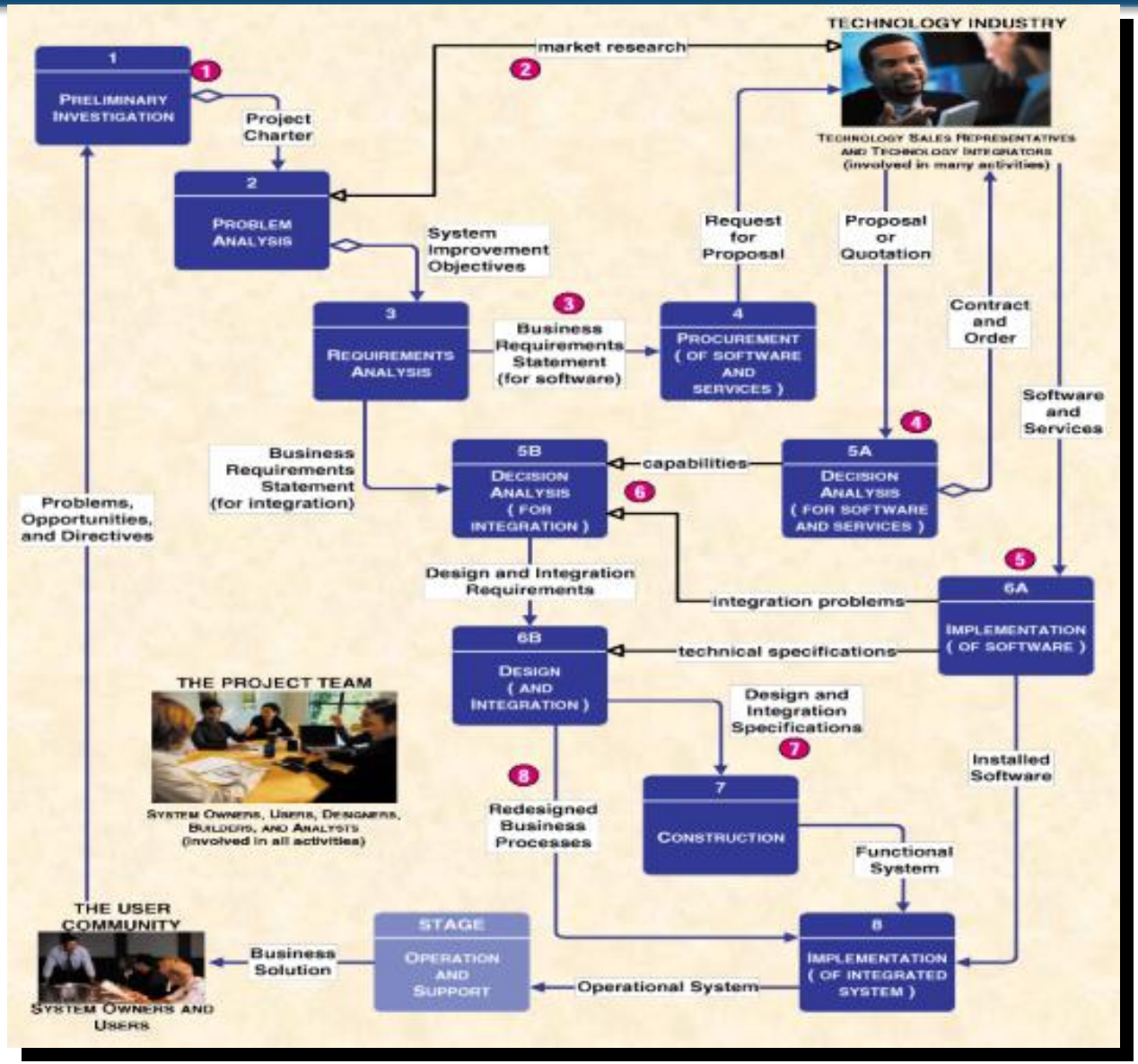
Rapid Application Development (RAD) Route



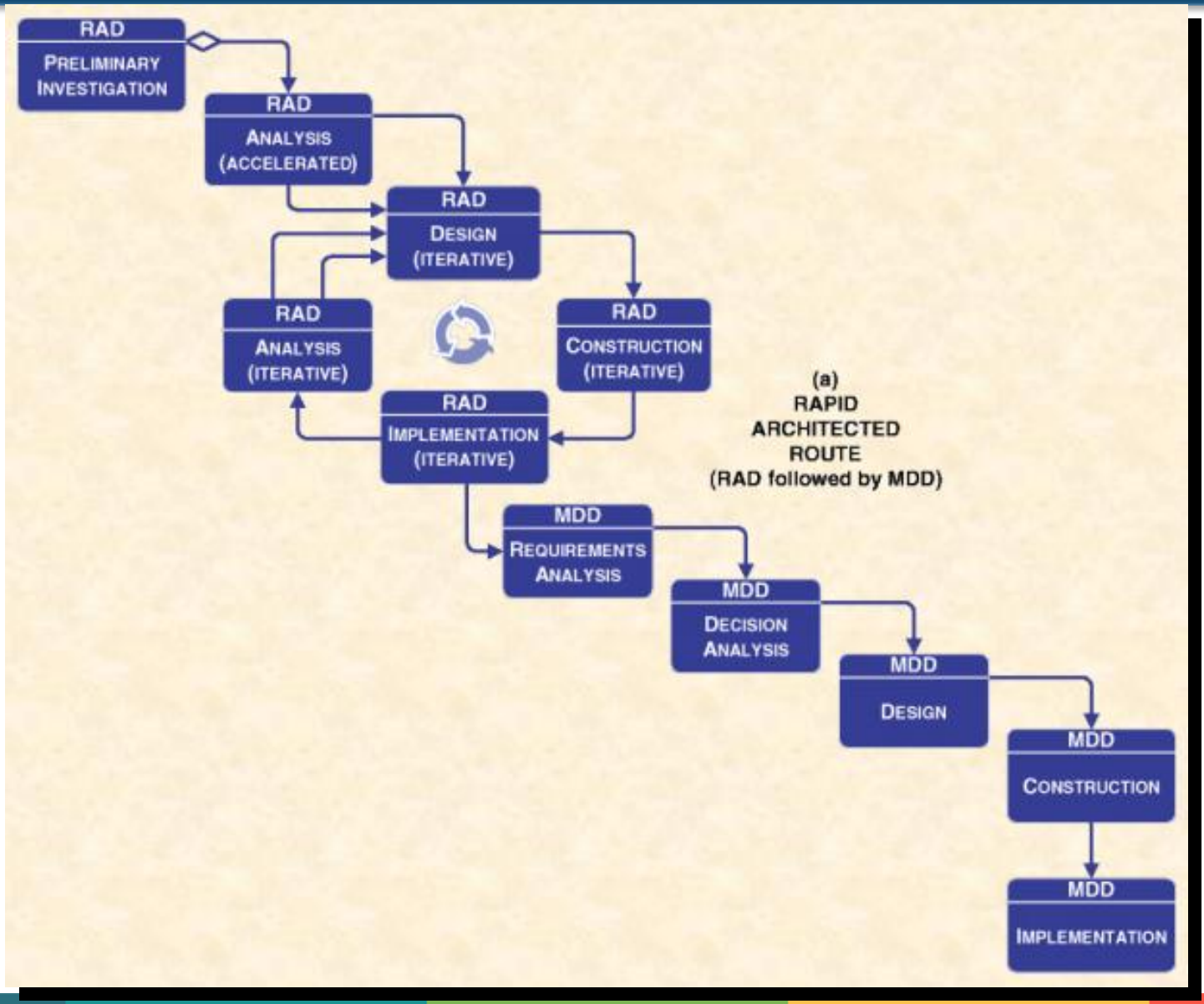
Commercial Off-the-Shelf Software Route

- **Commercial Off-The-Shelf** (COTS) software is a software package or solution that is purchased to support one or more business functions and information systems.
- The ultimate COTS software is ERP.

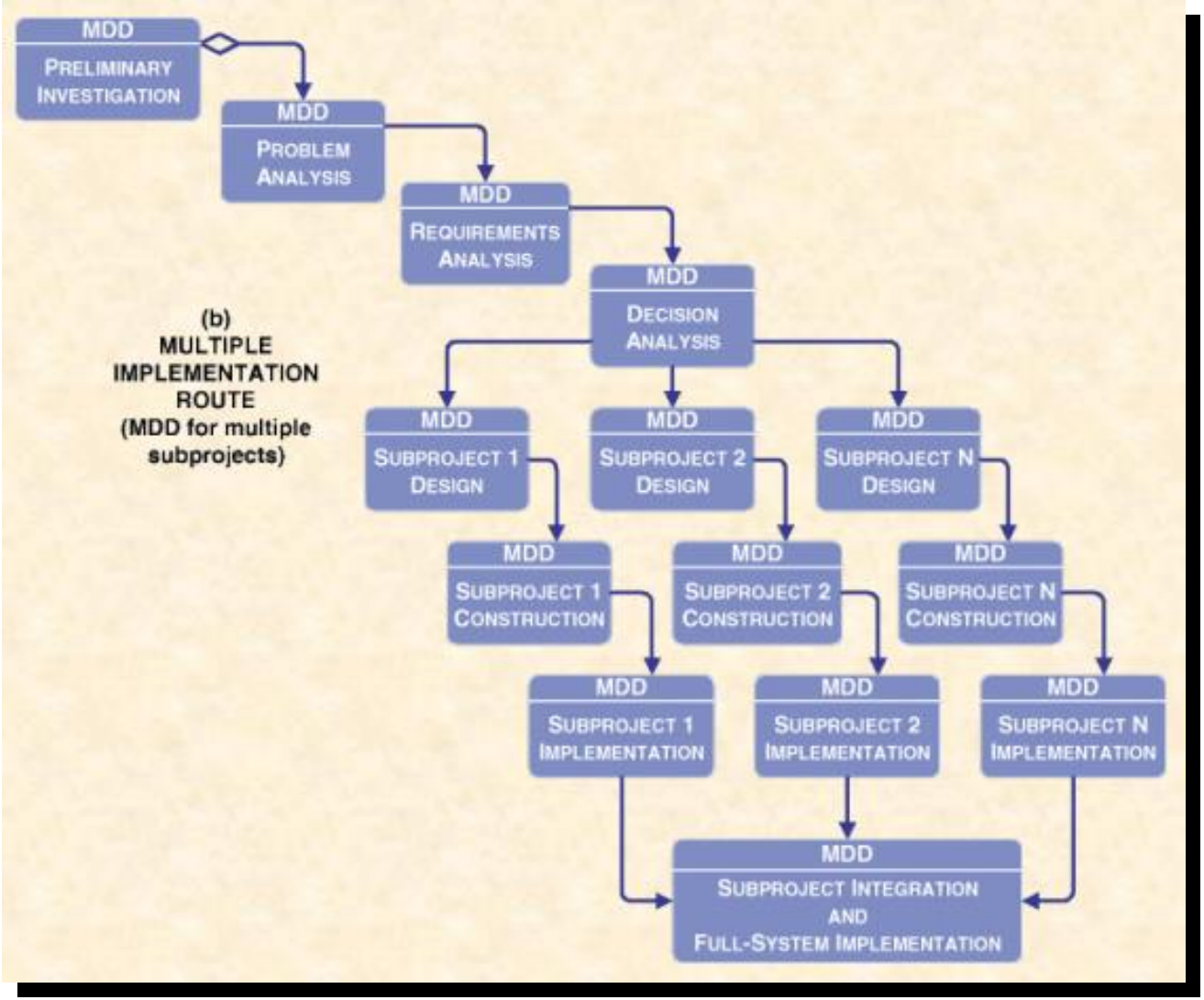
Commercial Off-the-Shelf (COTS) Software Route



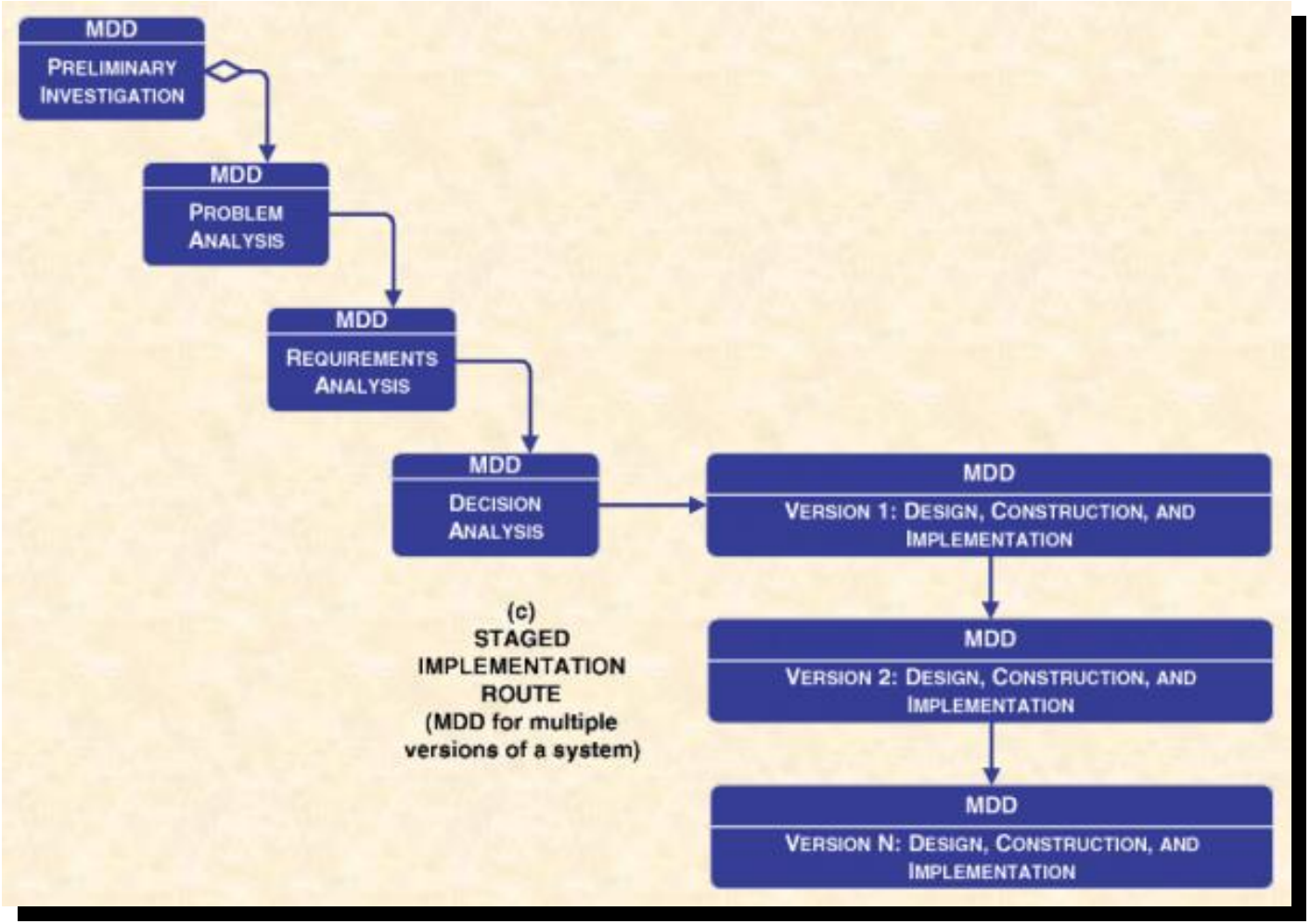
Hybrid: Rapid Architected Development



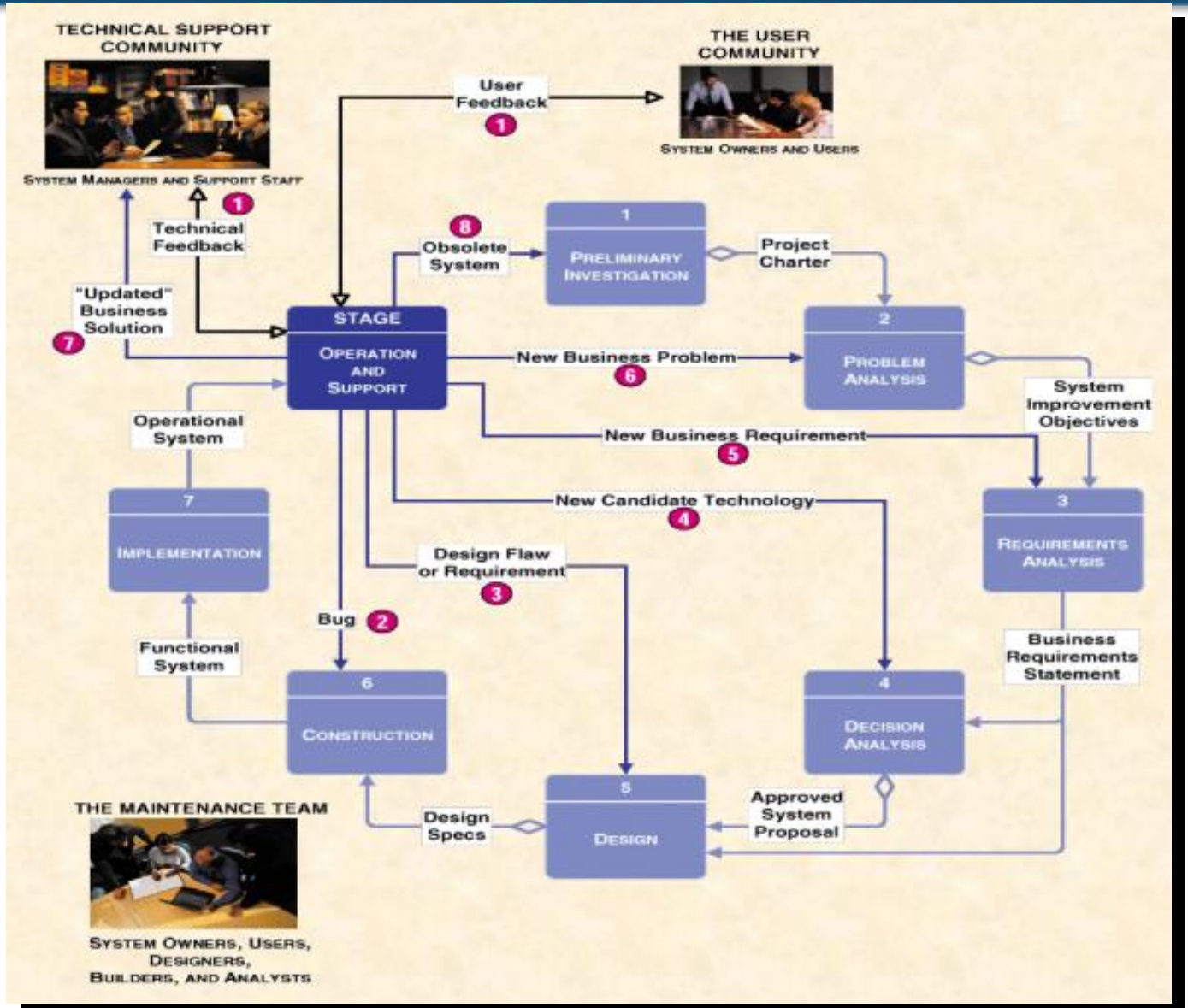
Hybrid: Multiple Implementation



Hybrid: Staged Implementation



Maintenance and Reengineering Route



Automatic Tools and Technology

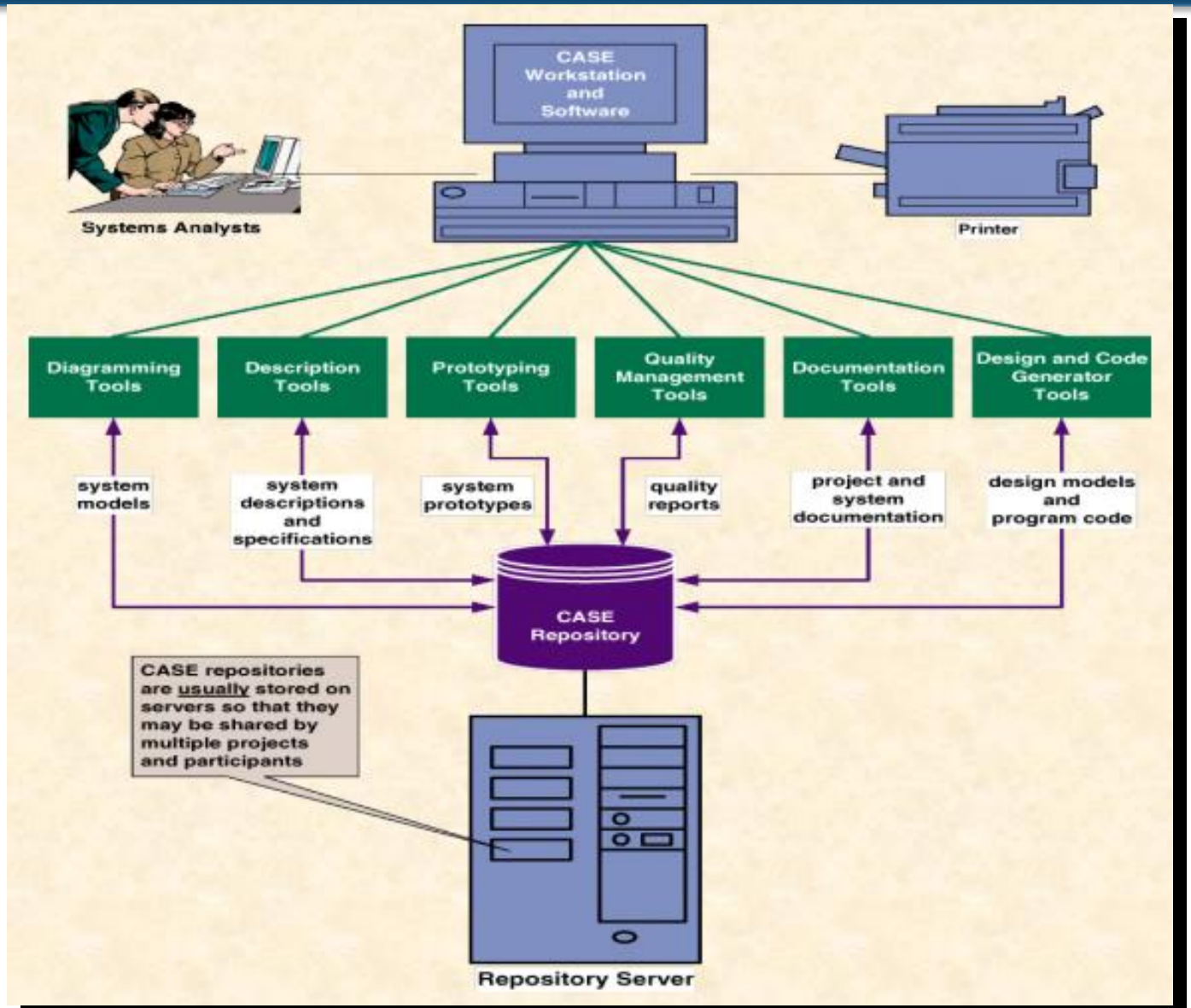
Automated Tools and Technology

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

CASE Tools

- Computer-Aided Systems Engineering (CASE) tools are software programs that automate or support the drawing and analysis of system models and provide for the translation of system models into application programs.
  - A CASE repository is a system developers' database. It is a place where developers can store system models, detailed descriptions and specifications, and other products of system development. Synonyms include dictionary and encyclopedia. Forward engineering requires the systems analyst to draw system models, either from scratch or from templates. The resulting models are subsequently transformed into program code. Reverse engineering allows a CASE tool to read existing program code and transform that code into a representative system model that can be edited and refined by the systems analyst.

CASE Architecture



CASE Tools (e.g.)

- Oracle's Designer
- Platinum's Erwin
- Rational's ROSE
- Popkin's System Architect
- Sterling's COOL product family
- Visible Systems' Visible Analyst
- Visio's Visio Enterprise

ADE Tools

- **Application Development Environments** (ADEs) are integrated software development tools that provide 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

ADE Tools(e.g.)

- Allaire's Cold Fusion
- IBM's Visual Age product family
- InPrise's Delphi and J Builder
- Microsoft's Visual Studio
- Microsoft Access
- Oracle's Designer
- Sybase's PowerBuilder
- Symantec's Visual Café

Process and Project Managers

- A **process manager** is an automated tool that helps to document and manage a methodology and routes, its deliverables, and quality management standards.
- A **project manager** is 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.

要点与引申

- 任何软件系统都不可能“永垂不朽” 。
- 可行性的判定是贯穿整个开发过程的。
- 尽量用数据（而不是程序）来体现系统中现在或将来可能出现的变数。
- “分而治之”是从秦始皇时代开始就被证明了的一种行之有效的战略。但是，过分的细分将带来过多的关联。
- 环境和工具只能用来支持开发工作，而不能取代开发者。

西安电子科技大学
XIDIAN UNIVERSITY

04 系统分析与设计

（System Analysis and Design）

Project Management

Content Structure

- **What is Project Management?**
  - 项目失败的原因；项目管理的知识结构和基本技术。
- **The Project Management Life Cycle**
  - 项目管理活动：确定项目的边界；认定项目中的各项任务；估计任务的持续期；规定任务之间的依赖关系；分配资源；指导项目组成员齐心协力地工作；监控开发过程；评估项目结果和总结经验。