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.

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.

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. 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.

The PIECES Problem-Solving Framework (Motivation)
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.

Classic Phases (from Chapter 1)
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 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.

X

Problem Analysis

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?

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 Phase (a system analysis transition 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?

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.

Construction and Testing Phase
Construct and test system components

Software

Purchased

Custom-built

Databases

User and System Interfaces

Hardware

Networks

Installation and Delivery Phase
Deliver the system into operation (production)

Deliver User training

Deliver completed documentation

Convert existing data