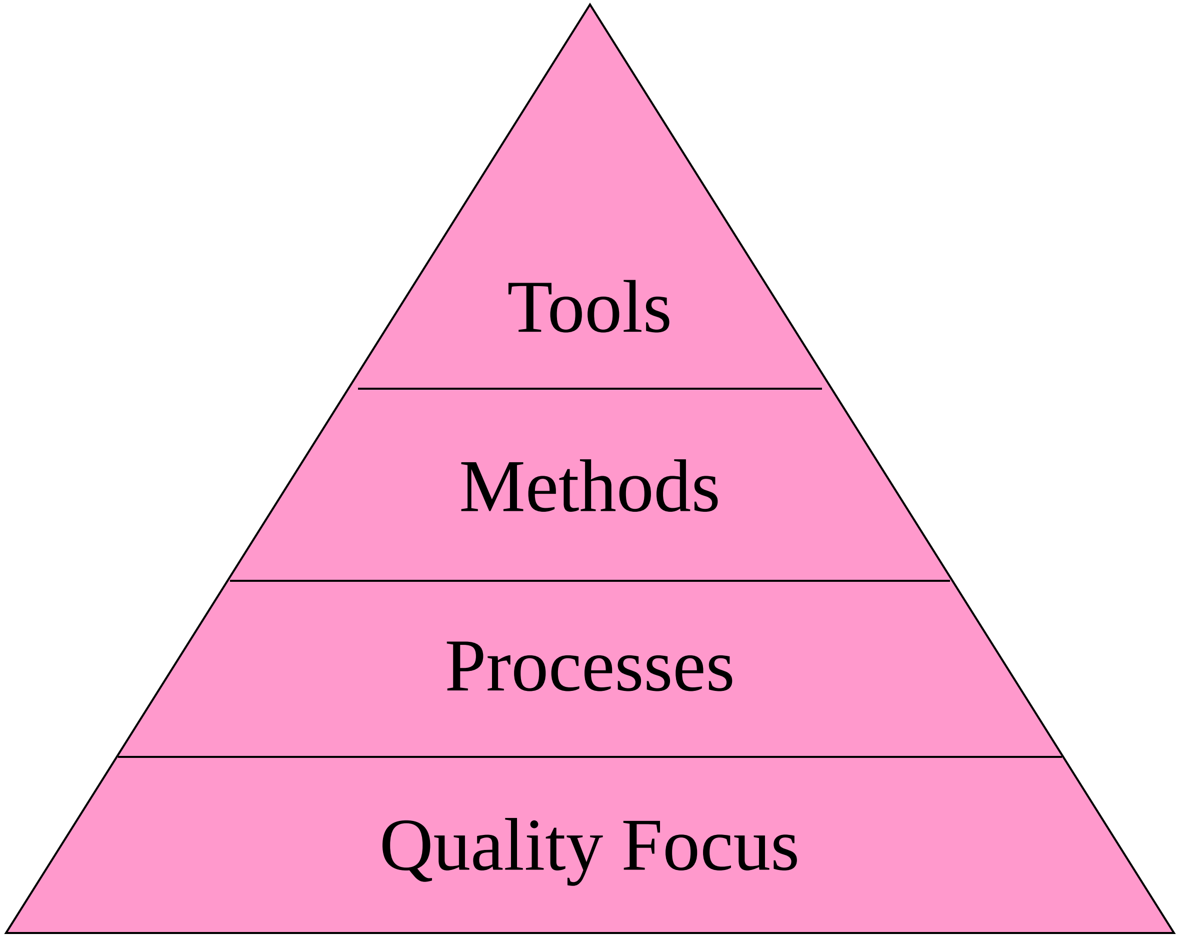
# Software Engineering

# The Software Process

# Software Engineering - Defined

- (1969) Software engineering is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines
- (IEEE) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software

# Software Engineering is a Layered Technology



# PROCESS, METHODS, AND TOOLS

- Process
  - Provides the glue that holds the layers together; enables rational and timely development; provides a framework for effective delivery of technology; forms the basis for management; provides the context for technical methods, work products, milestones, quality measures, and change management
- Methods
  - Provide the technical "how to" for building software; rely on a set of basic principles; encompass a broad array of tasks; include modeling activities
- Tools
  - Provide automated or semi-automated support for the process and methods (i.e., CASE tools)

# Generic Process Framework

- Communication
  - Involves communication among the customer and other stake holders; encompasses requirements gathering
- Planning
  - Establishes a plan for software engineering work; addresses technical tasks, resources, work products, and work schedule
- Modeling (Analyze, Design)
  - Encompasses the creation of models to better understand the requirements and the design
- Construction (Code, Test)
  - Combines code generation and testing to uncover errors
- Deployment
  - Involves delivery of software to the customer for evaluation and feedback

# Umbrella Activities

- Software requirements management
- Software project planning
- Software project tracking and oversight
- Software quality assurance
- Software configuration management
- Software subcontract management
- Formal technical reviews
- Risk management
- Measurement – process, project, product
- Reusability management (component reuse)
- Work product preparation and production

# What is a Process?

- (Webster) A system of operations in producing something; a series of actions, changes, or functions that achieve an end or a result
- (IEEE) A sequence of steps performed for a given purpose

# What is a Software Process?

- (SEI)  A set of activities, methods, practices, and transformations that people use to develop and maintain software and the associated products (e.g., project plans, design documents, code, test cases, and user manuals)
- As an organization matures, the software process becomes better defined and more consistently implemented throughout the organization
- Software process maturity is the extent to which a specific process is explicitly defined, managed, measured, controlled, and effective

# Capability Maturity Model (SW-CMM)

- Developed in 1987 by the Software Engineering Institute (SEI) at Carnegie-Mellon University under the sponsorship of DARPA
- Described in the book Managing the Software Process in 1989 by Watts Humphrey
- Published as a separate document: Capability Maturity Model for Software in 1991

# Immature Software Organizations

- Software processes are generally improvised
- If a process is specified, it is not rigorously followed or enforced
- The software organization is reactionary
- Managers only focus on solving immediate (crisis) problems
- Schedules and budgets are routinely exceeded because they are not based on realistic estimates
- When hard deadlines are imposed, product functionality and quality are often compromised
- There is no basis for judging process quality or for solving product or process problems
- Activities such as reviews and testing are curtailed or eliminated when projects fall behind schedule

# Five Levels of Software Process Maturity

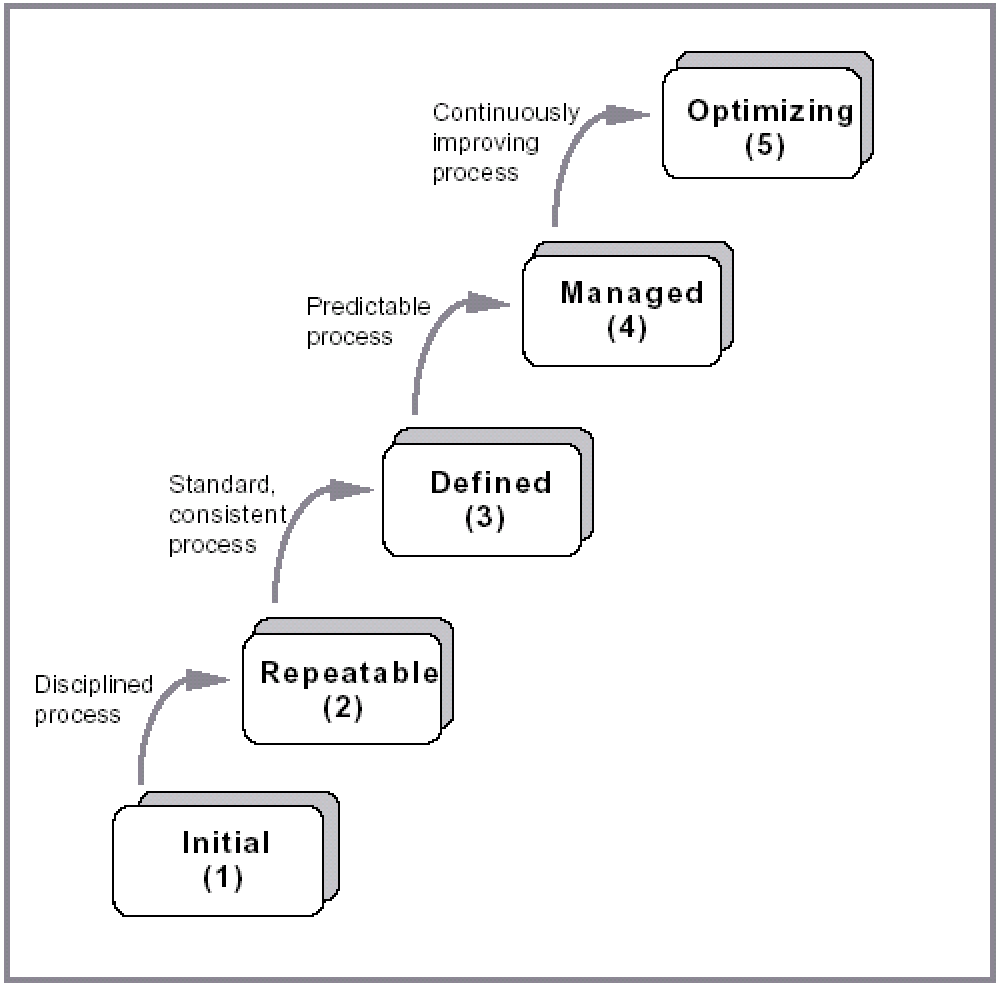


Figure 2.1 The Five Levels of Software Process Maturity

# Characteristics of Each Level

- Initial Level (Level 1)
  - Characterized as ad hoc, and occasionally even chaotic
  - Few processes are defined, and success depends on individual effort
- Repeatable (Level 2)
  - Basic project management processes are established to track cost, schedule, and functionality
  - The necessary process discipline is in place to repeat earlier successes on projects with similar applications

# Characteristics of Each Level (continued)

- Defined (Level 3)
  - The software process for both management and engineering activities is documented, standardized, and integrated into a standard software process for the organization
  - All projects use an approved, tailored version of the organization's standard software process for developing and maintaining software
- Managed (Level 4)
  - Detailed measures of the software process and product quality are collected
  - Both the software process and products are quantitatively understood and controlled

# Characteristics of Each Level (continued)

- Optimized (Level 5)
  - Continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies

# Visibility into the Software Process

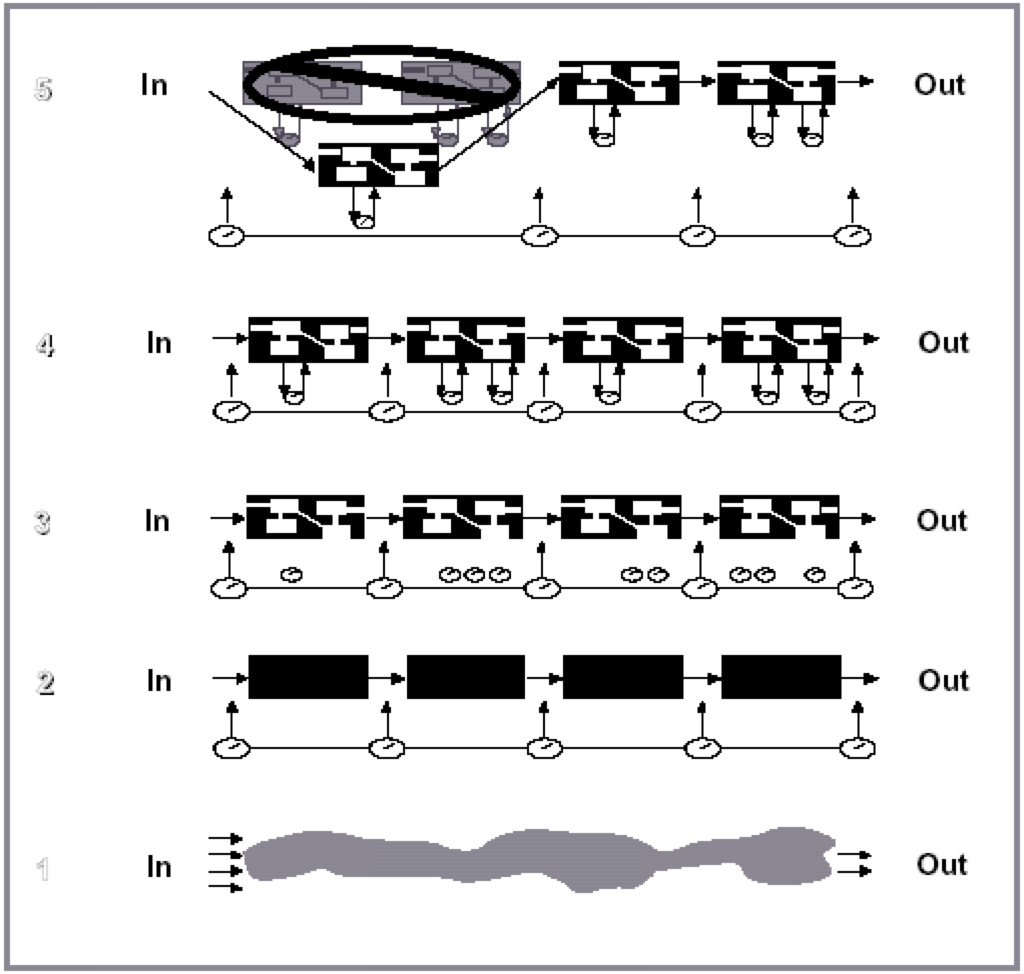
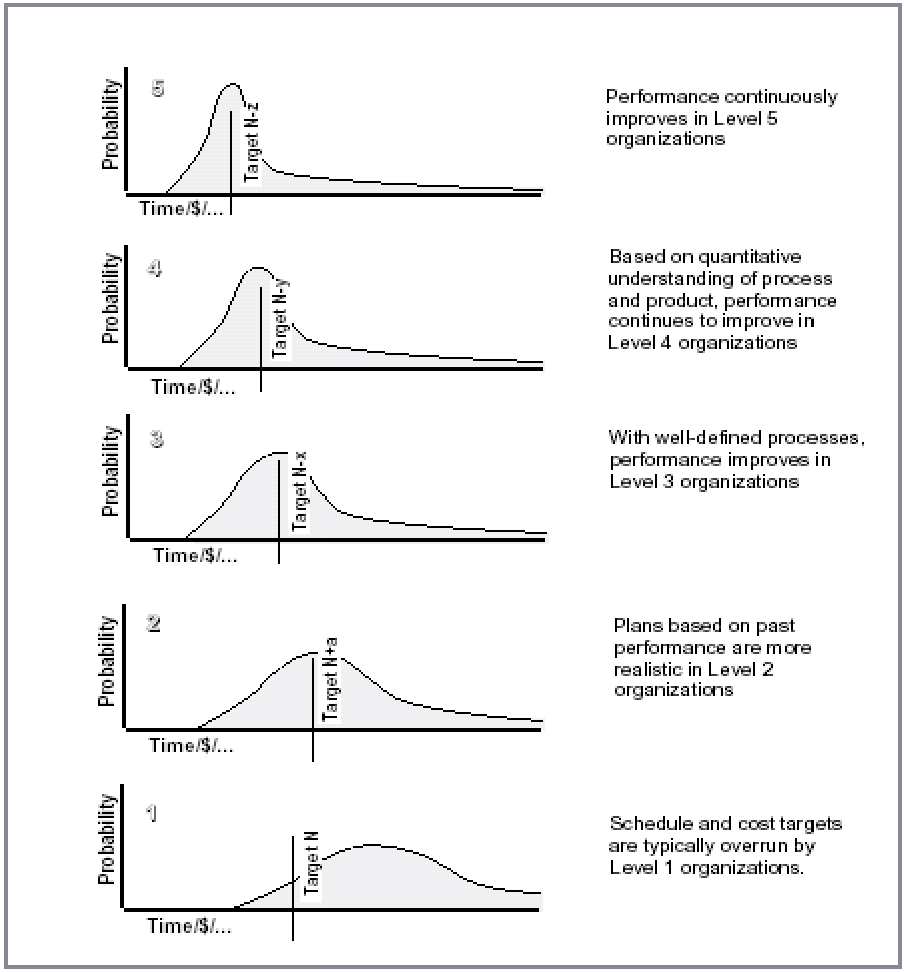


Figure 2.3 A Management View of Visibility Into the Software Process at Each Maturity Level

# PROBABILITY OF SCHEDULE AND BUDGET



Performance continuously improves in Level 5 organizations

Based on quantitative understanding of process and product, performance continues to improve in Level 4 organizations

With well-defined processes, performance improves in Level 3 organizations

Plans based on past performance are more realistic in Level 2 organizations

Schedule and cost targets are typically overrun by Level 1 organizations.

**Figure 2.4 Process Capability as Indicated by Maturity Level**

# THE CMM STRUCTURE

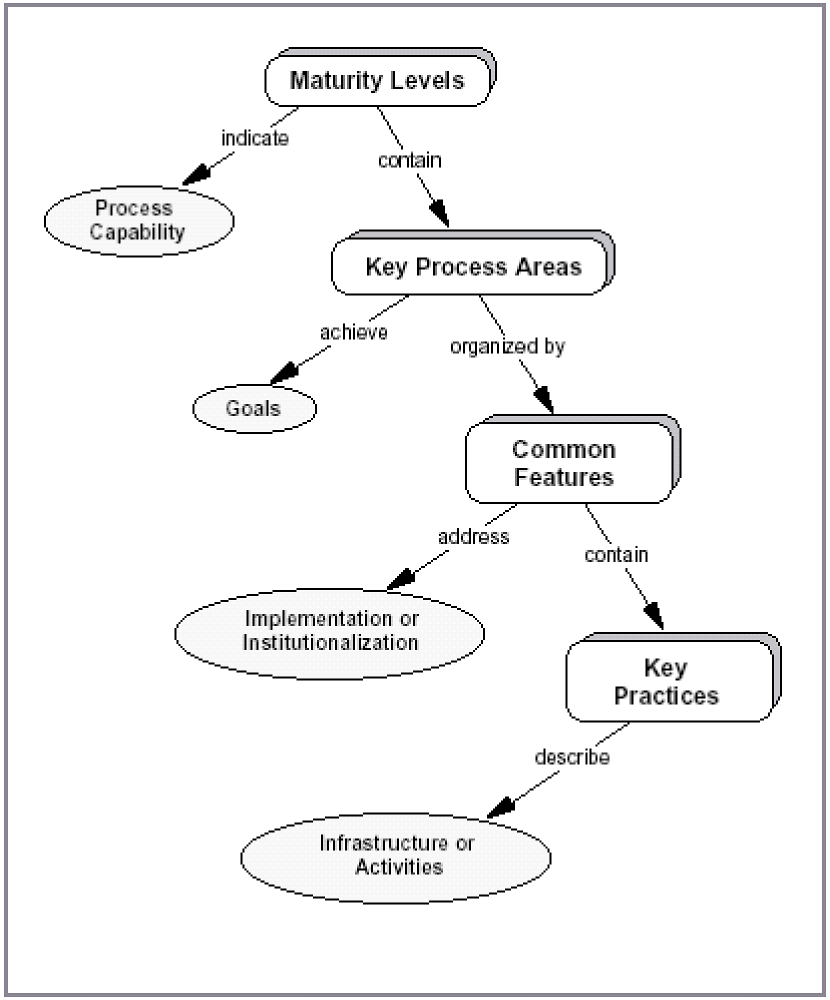
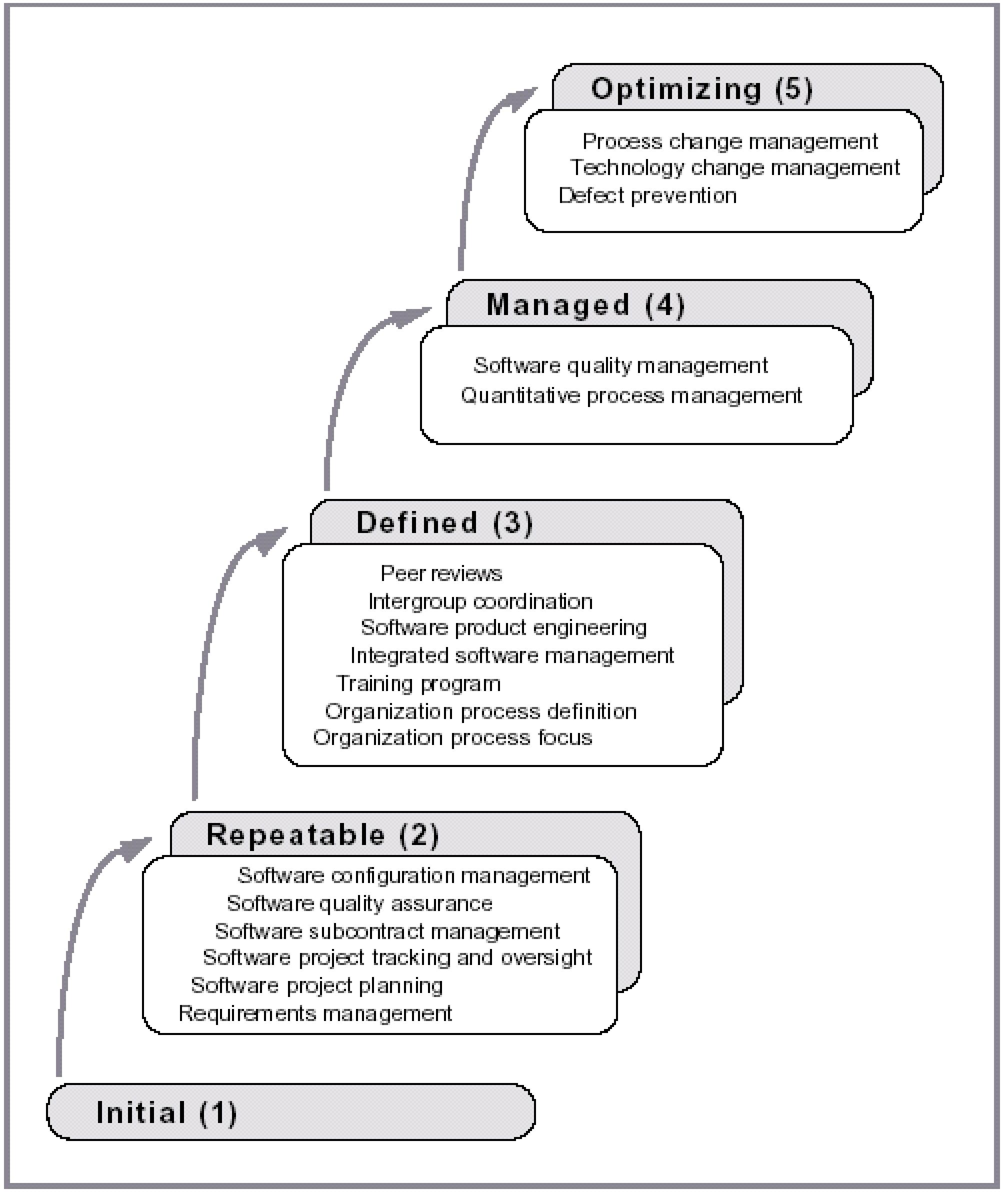


**Figure 3.1 The CMM Structure**

# KEY PROCESS AREAS



Optimizing (5)
Process change management
Technology change management
Defect prevention
Managed (4)
Software quality management
Quantitative process management
Defined (3)
Peer reviews
Intergroup coordination
Software product engineering
Integrated software management
Training program
Organization process definition
Organization process focus
Repeatable (2)
Software configuration management
Software quality assurance
Software subcontract management
Software project tracking and oversight
Software project planning
Requirements management
Initial (1)

Figure 3.2 The Key Process Areas by Maturity Level

# Software Process Assessments

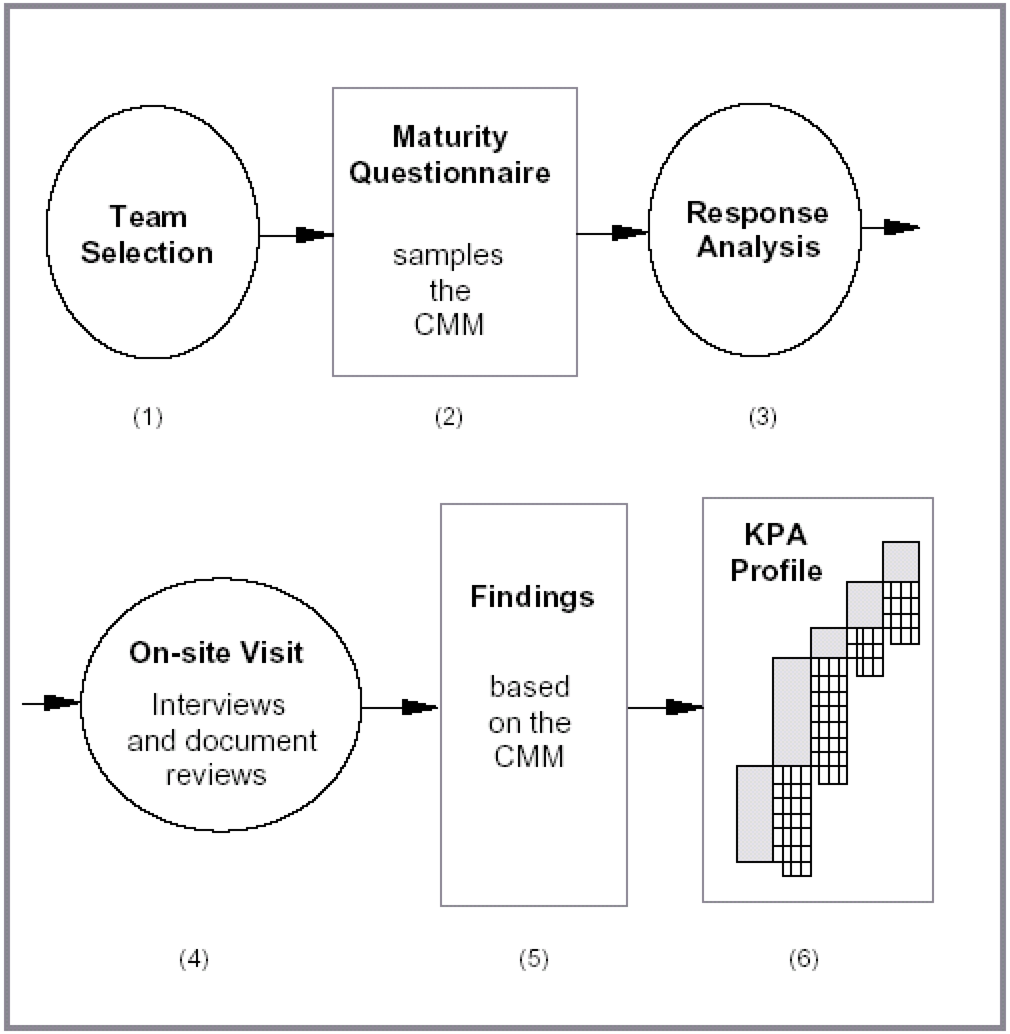


Figure 4.1 Common Steps in Software Process Assessments and Software Capability Evaluations

☺