Introduction
To The
The Software Engineering Institute’s
Capability Maturity Model®
for Software

The Software Engineering Institute (SEI)
A federally funded research and development center
Affiliated with Carnegie Mellon University (CMU)
SEI Vision and Mission

Vision:
To bring engineering discipline to the development and maintenance of software

Mission:
To provide leadership in advancing the state-of-the-practice of software engineering, to improve the quality of systems that depend on software
To facilitate higher quality communication among academia, government, and industry

Process & Method & Methodology

Process
A series of actions, changes, or functions that achieve an end or result.

Method
A reasonably complete set of rules that establishes a precise and repeatable way of performing a task and arriving at a desired result.

Methodology
A collection of methods, procedures, and standards that defines an integrated synthesis of engineering approaches to the development and maintenance of a product.
Why Is Software Process The Right Focus?

Software Is Needed Everywhere

This presentation was assembled by P. Wolfgang from material prepared by J. Vu, Boeing Shared Services Group
Customer Problems In Software

- Low visibility into progress and quality
- Unpredictable performance
- Constant surprises
- Excessive maintenance costs

Software Size Growth

Software Size: 20 K, 500 K, 20,000 K, 100,000 K, 200,000 K

This presentation was assembled by P. Wolfgang from material prepared by J. Vu, Boeing Shared Services Group
Software Complexity Growth

Scalability Problems

Order of magnitude growth in software size - Every 5 years

Scaling requires fundamental process changes:
- Can’t go from 6 mph to 60 mph by trying harder
- Can’t build skyscraper using carpenter hand tools
The Focus On Process

Focusing on product alone misses:

• Scalability issues
• Knowledge of how to do it better

Focusing on process predicts:

• Repeatability of outcomes
• Project trends
• Product characteristics

If builders built like programmer program, a woodpecker could destroy civilization!

Gerald Weinberg

The Definition Of Process

The logical organization of people, tools, procedures, and standards into work activities designed to produce a specified end result
Process Management Premise

The quality of a software product is governed by the quality of the process used to develop and maintain it.

To improve quality of the product, one must improve the quality of the process that creates the product.

---

Context

Software Development & Maintenance Process:
The development & maintenance of products that meet the needs of customers and markets

Software Improvement process:
Application of technology to improve software development and maintenance process
Process Improvement Objectives

Improve the quality, productivity, and reduce the cost of software activities.

Provide faster service, deliver higher quality products, and achieve customer satisfaction.

Typical Problems of Improvement Efforts

Improvement effort becomes unfocussed and vanishes

New methods are forgotten over time
Possible Causes

**Misunderstanding** of the problem being solved

**Lack of consensus** in the organization on which problem to solve first

**Too many changes** happening at the same time

Newly introduced solution disrupts current process

Everybody busy "Fighting Fires"

→ **Lack of a framework for process improvement**

### Capability Maturity Model (CMM)

A **conceptual framework** to help organizations:

- Characterize the maturity of their process (As Is)
- Establish **goals** for process improvement (To Be)
- Set **priorities** for immediate actions (Transition)
- Manage and sustain change in organizations (Stabilize)
- Introduce changes incrementally, to avoid disrupting current processes
Immature Software organizations

Processes are ad hoc, and occasionally chaotic
Product quality is unpredictable
Costs and schedules are usually exceeded
Testing and reviews usually curtailed under stress
Success rides on individual talent

Mature Software Organizations

Processes are defined and documented
Management plans, monitors, and communicates
Roles and responsibilities are clear
Product and process are measured
Quality, cost and schedules are predictable
Level 1 - The initial Process

Unstable environment lacking project management disciplines

Practitioners argue against engineering discipline under the guise of "Individual creativity"

Standard and practices often sacrificed to schedule

Process capability is unpredictable

Schedule, cost, and quality target are rarely met

Success rides on individual talent and heroic effort
**Maturity Level 2**
*(Repeatable)*

- Initial Level
- Repeatable Level

Requirements Management
- Software Project Planning
- Software Project Tracking and Oversight
- Software Configuration Management
- Software Quality Assurance
- Software Subcontract Management

**Disciplined Process**
*(Basic Project Management)*

---

**Level 2 - The Repeatable Process**

Basic project management discipline is installed to ensure that software engineering practices are followed and measured.

- Previously successful processes are repeatable in stable environment
- Reasonable commitments are planned
- Process capability exists for meeting business needs and schedules
- Project metrics are collected
- Foundation for continuous quality improvement is in place
Maturity Level 3
(Defined Process)

Repeatable Level → Defined Level

Organization Process Focus
Organization Process Definition
Peer Reviews
Training Program
Integrated Software Management
Intergroup Coordination
Software Product Engineering

Standard, Consistent Process
(Process Architecture)

Level 3 - The Defined Process

Common software processes are defined, documented, and applied throughout the organization

Process of sharing successful practices is in place

Roles and responsibilities are well defined

Organization processes are measured

Process capability exists to meet schedule, cost, and functionality target within established product lines

Organization metrics are collected and used for CQI
Maturity level 4
(Managed)

Defined → Process Measurement & Analysis → Product Quality Management → Managed

Predictable Process
(Management by Facts and Data)

Level 4 - The Managed Process

Statistical process control principles are used to address special causes of process variation

Process measurement and product measurement are available and correlated for "fine tuning" of existing processes

Standard software processes are established based on facts and data collected on software engineering practices

Process capability exists to perform within narrowly defined quantitative limits - Targets are predictable.

Business enterprise metrics are collected and used for CQI
Level 5 - The Optimizing Process

- Software process is continually improved
- Chronic causes of poor performance are eliminated
- New technologies are prototyped, piloted, and if successful, introduced into existing process
- Process capability is continually raised
Key Concept

Maturity levels cannot be skipped because each level forms the necessary foundation from which to achieve the next level.

A level 1 organization that tries to implement a defined process (level 3) before it has established a repeatable process (level 2) is usually unsuccessful because project managers are overwhelmed by schedule and cost pressures.

It is suggested that the organization focuses on management processes before engineering processes.

Although it seems easier to define and implement an engineering process than a management process, without management discipline, the engineering process is sacrificed to schedule and cost pressures.
Principles Of Process Improvement

1) Improvement direction must **start at the top**
2) **Everyone must be involved** in the improvement process
3) Effective improvement requires **knowledge of current process**
4) Improvement is **continuous**
5) Improvement requires **investment**

Direction Must Start At The Top

**Process Improvement requires leadership**

**Process Improvement is a business strategy**

- Set challenging goals
- Monitor progress
- Insist on performance
Everyone must be involved

Process Improvement is a **team effort**
Fix the process, **not** the people
Process Improvement is **everyone’s job**

Effective Improvement Requires Knowledge

If you don’t know where you are at, a map won’t help
*Chinese Proverb*
Improvement Is Continuous

Reactive “Quick Fix” makes things worse
Every defect is an improvement opportunity
Error prevention is better than error recovery

Plan → Do → CQI → Act → Check

Improvement Requires Investment

It takes time, skills, and resources to improve the process
Unplanned process improvement is wishful thinking
Automation of a poorly defined process will produce poorly defined results - Faster
Process Improvement should be made in small steps
Cost of Implementation Failure *

<table>
<thead>
<tr>
<th>Short Term</th>
<th>Long Term</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Direct</strong></td>
<td><strong>Indirect</strong></td>
</tr>
<tr>
<td>Wasted resources:</td>
<td></td>
</tr>
<tr>
<td>• Money</td>
<td>• Morale suffers</td>
</tr>
<tr>
<td>• Time</td>
<td>• Job security threatened</td>
</tr>
<tr>
<td>• People</td>
<td></td>
</tr>
<tr>
<td>Business goal not achieved</td>
<td>Business strategies not accomplished</td>
</tr>
<tr>
<td>• Lower confidence in leadership</td>
<td>• Resistance to change increase</td>
</tr>
<tr>
<td>• Resistance to change increase</td>
<td>• Next change more likely to fail</td>
</tr>
</tbody>
</table>

Each time an improvement effort fails to achieve its stated objectives, it incurs both short-term and long-termed costs

Software Process Improvement (SPI) Phases

The IDEAL Model
Maturity Levels

Initial (1)

Repeatable (2)

Defined (3)

Managed (4)

Optimizing (5)

COI (Continuous Improving Process)

Management by Facts & Data (Measurable Process)

Process Architecture (Consistent Process)

Basic Project Management (Disciplined Process)

Structure of The CMM

Process capability indicate Maturity levels 5 levels

Goals Achieve Contain Key Process Areas 18 KPAs

Implementation Address Organized by Common Features 5 common features

Infrastructure or Activities Describe Contain Key Practices 316 Key Practices

This presentation was assembled by P. Wolfgang from material prepared by J. Vu, Boeing Shared Services Group
Maturity Levels & Process Capability

• A maturity level is a well-defined evolutionary plateau toward achieving a mature software process

• Software process capability describes the range of expected results that can be achieved by following a software process

Key Process Areas

A cluster of related activities that, when performed collectively, achieve a set of goals considered important for enhancing process capability

Defined to reside at a single maturity level

Identify the issues that must be addressed to achieve a maturity level
Common Features

Attributes that indicate whether the implementation and institutionalization of a key process area is effective, repeatable, and lasting.

- Commitment to perform
- Ability to perform
- Activities performed
- Measurement and Analysis
- Verifying Implementation

Importance of Institutionalization

The organization outlives those who leave it

The organizational culture must convey the process

The CMM has four common features that focus on institutionalizing the process:

- Commitment to perform
- Ability to perform
- Measurement and analysis
- Verifying implementation

Activities performed focus on implementing the process
Common Features

Commitment to perform

Ability to perform

Resources and Funding

Training

Tools

Activities performed

Guidelines

Procedures

Standards

Roles and Responsibilities

Skills & Knowledge

Measurement & Analysis

Verifying Implementation

Key Practices

The infrastructures and activities that contribute most to the implementation and institutionalization of a Key Process Area

Describe “What” is to be done, but they should not be interpreted as mandating “how”

316 Key Practices in the CMM
Examples

Key Practices:

The project’s software development plan is developed according to a documented procedure (PP Ac 6)

Estimates for the size of the software work products (or change to the size of software work products) are derived according to a documented procedure (PP Ac 9)

The software work products to be placed under configuration management are identified (CM Ac 4)

The SQA group audits designated software work products to verify compliance (SQA Ac 5)

WHAT not HOW

The Repeatable Level

At level 1, an organization gets the job done

At level 2, a software project management system is in place

The organization sets expectations via Policies/Directions

Level 2 projects have disciplined processes and an infrastructure to get the job done
Maturity Level 2
(Repeatable)

Initial Level

Requirements Management
Software Project Planning
Software Project Tracking and Oversight
Software Configuration Management
Software Quality Assurance
Software Subcontract Management

Repeatable Level

Disciplined Process
(Basic Project Management)

Requirements Management

The purpose is to establish a common understanding between the customer and the software project.

Involves

- Document and control customer requirements (i.e. Specs, SOW..etc)
- Plans, products, and activities are kept consistent with the requirements.
What Do We Mean By Requirements?

* Customer may be external or internal
  The customer and/or end user identifies

Allocated Requirements

This presentation was assembled by P. Wolfgang from material prepared by J. Vu, Boeing Shared Services Group
Requirements Are Documented

The system requirements allocated to software must be documented.

Documenting requirements can be as simple as a Memo, Trouble Reports, Change Request, or as elaborate as a multi-volume specification.

If requirements change, the changes must be documented and all resulting necessary changes in other documents must be tracked and verified.
Requirements & Other Key Process Area

- Configuration Management
- Project Management
- Requirements Management
- Software Quality Assurance
- Subcontract Management
- Testing Group

Requirements Management

- Organization Direction
- Documented Requirements
- Resources
- Training
- Capture requirements
- Review requirements
- Analyze requirements
- Coordinate activities
- Approved Requirements
- Make/Buy/Reuse Decisions
- Roles & Responsibilities
- Metrics
- Reviews

This presentation was assembled by P. Wolfgang from material prepared by J. Vu, Boeing Shared Services Group.
Software Project Planning

Purpose is to establish reasonable plans for performing the software engineering and for managing the software project.

Involves:
- Developing estimates for the work to be performed
- Establishing the necessary commitments
- Defining the plan to perform the work

Managing based on a Plan *

Plan provides the basis for initiating the software effort and managing the work.

Other names for this plan included:
- Software Development Plan
- Software Project Management Plan
- Software Project Plan
- Project Management Plan
- Software Engineering management Plan

* Different customer requires different plan
What Is A Software Plan?

A software Plan may specifies many or all of the following:

- The project’s chosen software life cycle
- A list of products to be developed
- Schedules
- Estimates (Cost, size, effort etc.)
- Facilities, support tools, hardware
- Project risks

Plans are based on estimates

In creating estimates for size, effort, cost, schedule ..etc:

- Use historical data, where available
- Document assumptions and estimates

Good estimating depends on the skills and judgment of the estimator

High-Level Project Planning Process

![Diagram of project planning process]

Estimates

Actuals

Develop software
Project Planning

- Estimate product size, cost, effort, resources, and schedule.
- Identify risks
- Identify work products
- Identify software life cycle
- Develop project plan
- Record planning data

Project Plan:
- Estimates
- Risks
- Work Products
- Planning data

Metrics

Reviews

Project Tracking and Oversight

Purpose is to provide adequate visibility into actual progress so that management can take effective actions when performance deviates significantly from the plan.

Involves

- Tracking and reviewing software accomplishments and results against documented estimates, commitments, and plans
- Adjusting plans based on actual accomplishments and results
Manage to a Plan

Progress must be tracked against plans and specifications, including:

- Product size
- Project effort, cost, and schedule
- Activities
- Risks

Mechanism to track progress against plans include both internal reviews and formal reviews with the customer.

If and when discrepancies between plans and actual progress occur, a judgment must be made about whether to:

- Change the way the work is being done
- Adjust the plans

Project Tracking & Oversight

- Track activities planned in project plan
  - Actual Vs Planned
- Take corrective actions
- Track progress, size, cost, effort, resources, schedule and corrective actions
- Manage risks
- Record measurement and replanning data
- Conduct reviews

Revised project plan
- Estimates
- Risks
- Data

Action items from reviews

Metrics

Reviews

This presentation was assembled by P. Wolfgang from material prepared by J. Vu, Boeing Shared Services Group
Software Configuration Management

Purpose is to establish and maintain the integrity of the products of the software project throughout the software life cycle

Involves

- Identifying configuration items/units
- Systematically controlling changes
- Maintaining integrity and traceability of the configuration throughout the software life cycle

Controlling Change

Configuration Management provides a stable working environment

Uncontrolling change of work products is a chaotic process

Configuration Management provides a “memory” of the status of software work products via baselines

Baseline is a specification or product that has been formally reviewed and agreed on. It serves as the basis for further work, and can only be changed through formal change control procedure
High-Level Change Management Flow

Change
Change Board
Approve/Reject
Analyst
Change Board
Analyze
Estimates
Recommend
Programmer
Authorize
Implement
CM Librarian
Control/Storage
Tester/SQA
Verify
Change Board
Release
Approve
Authorize
Baseline

CM Disciplines

<table>
<thead>
<tr>
<th>Discipline</th>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification</td>
<td>What is my software configuration ?</td>
<td>The software consists of the following items ....... each of which has its own version designation .....</td>
</tr>
<tr>
<td>Change Management</td>
<td>How are changes managed ?</td>
<td>The steps are .....</td>
</tr>
<tr>
<td>Status accounting</td>
<td>What changes have been made and when ?</td>
<td>The current configuration was changed as follows ....</td>
</tr>
</tbody>
</table>
Software Configuration Management

<table>
<thead>
<tr>
<th>Organization Direction</th>
<th>Develop CM plan</th>
<th>CM Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Board</td>
<td>Identify configuration items</td>
<td>Baselines</td>
</tr>
<tr>
<td>Resources</td>
<td>Establish CM library</td>
<td>Establish baselines</td>
</tr>
<tr>
<td>Training</td>
<td>Control changes to baselines</td>
<td>Releases</td>
</tr>
<tr>
<td>Metrics</td>
<td>Manage change requests &amp; problem reports</td>
<td>Configuration records</td>
</tr>
<tr>
<td>Reviews</td>
<td>Record status</td>
<td>CM reports</td>
</tr>
<tr>
<td></td>
<td>Develop CM reports</td>
<td></td>
</tr>
</tbody>
</table>

Software Quality Assurance

Purpose is to provide management with appropriate visibility into the process being used and the product being built

Involves

Reviewing and auditing the software products and activities to ensure that they comply with the applicable procedures, processes, and standards

Providing the software project and other appropriate managers with the results of those reviews and audits

The value of SQA is that it provides an independent view of project activities process and product
Provides Visibility

**SQA is a function (May or may not be an organization)**

The value of SQA is that it provides an independent view of project activities, process, and product.

SQA serves as the “eyes and ears” of management.

Without an independent SQA function, can the project demonstrate objective means of adherence?

---

Resolving Noncompliance

There are three possible ways for resolving a noncompliance issue:

- Make the product or process satisfy the standard, procedure, or requirement.
- Change the standard or procedure to make it usable.
- Make a executive decision not to satisfy the standard, procedure or requirement.

Proactive SQA functions as value-adding member of the project team.
Software Quality Assurance

Organization
Direction

Develop SQA Plan
Review activities
Resolve Non-compliance
Report Non-compliance
Conduct reviews with customer

Metrics
Reviews

SQA Plan

Review reports
Non-Compliance reports

Resources

Training

Barriers To SQA

Disbelief in the value of SQA
Lack of respect for SQA
Lack of resources & funding for SQA activities
No believe in standards & procedures
Unqualified & Untrained SQA member
Belief that SQA responsible for quality

This presentation was assembled by P. Wolfgang from material prepared by J. Vu, Boeing Shared Services Group
SQA Evolves

Proactive SQA functions as a value-adding member of the project team

- SQA start early in the project
- SQA helps prepare and review procedures, plans, and standards
- SQA help drive improvements of process and product

Current industry concept of Integrated Product Team (IPT)

Quality Is Built In

Design in Quality

In-process detection and prevention

Current

In process Inspection

Post release inspection

Future
Software Subcontract Management

Purpose is to select qualified software subcontractors and manage them effectively

Involves

- Selecting a software subcontractor
- Establishing commitments with the subcontractor
- Tracking and reviewing the subcontractor’s performance and results

*Prime contractor* is the organization responsible for building a system, and may contract out part of the work to another contractor, the *subcontractor*. Therefore, the performance of prime contractor may critically depend on performance of the subcontractor.

Selecting Subcontractors

The qualifications of a subcontractor may depend on many factors:

- Process capability
- Software engineering expertise
- Application domain knowledge
- Strategic business alliances

In selecting subcontractor, prime contractors must perform activities additional to the ordinary project management effort:

Specify the work to be performed and procedures to be followed by subcontractor:

- **Statement Of Work**
- **Requirements**
- **Products to be delivered**
- **Standards and procedures**
- **Criteria for success**
Managing a Subcontract

The prime contractor must manage the subcontract:

- Ensure subcontractors follow software development plans, standards, procedures, and processes
- Track progress via
  - Periodic technical and formal reviews
  - Monitor SQA by subcontractor
  - Monitor CM by the subcontractor

What About COTS?

Commercial-Off-The-Shelf (COTS) items and Non-Developmental Items (NDI) are not explicitly addressed in the CMM

COTS usage:
- As-Is (No change)
- Modified (Tailored COTS)
- Contracted COTS (Modified by developer of COTS)
Software Subcontract Management

- Define subcontract work
- Select qualified subcontractor
- Manage subcontractor
- Review subcontractor plan
- Track subcontractor progress
- Monitor activities
- Conduct acceptance testing
- Evaluate subcontractor performance

Subcontract SOW

Subcontract plan

Status report

Action items

Report

Acceptance test

Subcontract product

Organization Direction

Resources

Training

Metrics

Reviews

Summary

Level 2 is only the beginning

“Change the process, not the people” means focus on changing the process rather than blaming individuals

To change the process we also must work to change the relationship styles and attitudes of the people

People have to change their behavior to adjust to the new process

From our perspective, level 1 organization are mostly running into walls, level 2 and 3 organizations are learning where the walls are, and level 4 and 5 organizations are building walls for the other guys to run into

John Major, Sr. Vice President
Motorola Co.
Institutionalized Processes

The building of infrastructure and culture that support methods, practices, and procedures so that they are the on-going way of doing business, even after those who originally defined them are gone.

“That’s the way we do thing around here”.

An infrastructure that contains effective, usable, and consistently applied processes across the organization.

---

Moving From Level 2 to Level 3

At level 2, the focus is on **project management**

At level 3, the emphasis shifts to **consistent process across organization**

- Best practices are gathered across the organization
- Processes are tailored as appropriate

The organization supports the projects by establishing:

- Common processes
- Common measurements
- Required training
Maturity Level 3
(Defined)

Repeatable Level → Defined Level

Organization Process Focus
Organization Process Definition
Peer Reviews
Training Program
Integrated Software Management
Intergroup Coordination
Software Product Engineering

Consistent Process
(Process Architecture)

Key Activities At Level 3

Define a **common set of processes** which can be tailored for any given project within the organization

The common set of processes is developed in a combined **bottom up** (Best Practices) and **top down** fashion (Software Architecture)

The common set of processes must be validated via process improvement phases (Define, Pilot, Refine, and Institutionalize)

Cross-Organization coordination activities must be planned, and measured to ensure consistency and avoid conflicts

Validated process will be automated and supported
Attributes
Processes in mature organization share a number of attributes:

Defined
Documented
Trained
Practiced
Measured
Improvable
Maintained
Supported
Controlled
Enforced

Essential Criteria
The Goals of each Key Process Area are
The Essential Criteria
The CMM Does Not
Imply a particular Life Cycle Model
Imply a specific Software Technology
Require a specific Set of Documents
Advocate a particular organization structure

Organization Maturity Overview

This presentation was assembled by P. Wolfgang from material prepared by J. Vu, Boeing Shared Services Group
Tying It All Together

Policies provide directions and expectations for the behavior of everyone in the organization.

Processes are “living” entities, which are supported and maintained.

Everyone is involved in continuous process improvement.

Processes are documented and practiced.

Training is a way of establishing consistent ways for performing an activity.

Measurement of both product and process is an integral part of every key process area.

Continuous improvement is based on business objectives and cost/benefit analysis.

Note: CMM is not a process but a process framework.

Measurement

Level 1 to 2: Projects collect data about cost, quality, size, time
Different projects may use different definitions

Level 2 to 3: Organization collect data from projects
Common definitions from one project to the next

Level 3 to 4: Data analysis is based on statistical process control
to measure the effectiveness of process as compare to product performance

Level 4 to 5: Continuing improvement to optimize process is based on business objective and cost/benefit analysis
This presentation was assembled by P. Wolfgang from material prepared by J. Vu, Boeing Shared Services Group