



# Using the Software CMM<sup>®</sup> in Small Projects and Small Organizations

**Mark C. Paulk**

**February 1999**

**Software Engineering Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213-3890**

SMIDEAL, Personal Software Process, PSP, Team Software Process, and TSP are service marks of Carnegie Mellon University.  
© Capability Maturity Model and CMM are registered in the U.S. Patent and Trademark Office.  
This work is sponsored by the U.S. Department of Defense.  
©1999 by Carnegie Mellon University.



## Topics

**→ The Current State of Affairs**

**Small Projects & Organizations**

**Software CMM Interpretation**

**Examples of Interpreting the Software CMM**

**Abusing the Software CMM**

**Concluding Thoughts**



## Software Crisis Headlines

Are meeting schedules, budgets, and requirements important to small projects? To small organizations?

Software crisis headlines focus on large projects.

Most software projects are comparatively small.

- size of software growing rapidly
- tools and support environment helping software professionals do more
- **many small problems sum to crisis!**

3



## The State of the Practice

Is this the state of affairs in your organization?

- **“I'd rather have it wrong than have it late.”**
  - A senior software manager (industry)
- **“The bottom line is schedule. My promotions and raises are based on meeting schedule first and foremost.”**
  - A program manager (government)

If it is, are managers and practitioners unhappy with the status quo?

- sufficiently unhappy to change things?
- willing and able to attack the known problems?

4



## What Is the CMM?

A **common-sense** application of process management and quality improvement concepts to software development and maintenance

A **community-developed** guide

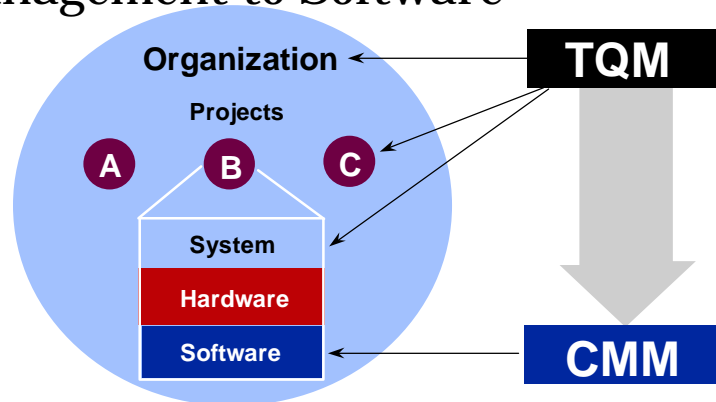
A model for **organizational** improvement

The underlying structure for **reliable and consistent** CMM-based appraisal methods

5



## Applying Total Quality Management to Software



Process improvement fits in an overall business context—CMM applies to software.

6



## SW-CMM v1.1 Key Process Areas

Level	Focus	Key Process Areas	
5 Optimizing	<i>Continuous process improvement</i>	Defect Prevention Technology Change Management Process Change Management	Quality Productivity    Risk Waste
4 Managed	<i>Product and process quality</i>	Quantitative Process Management Software Quality Management	
3 Defined	<i>Engineering processes and organizational support</i>	Organization Process Focus Organization Process Definition Training Program Integrated Software Management Software Product Engineering Intergroup Coordination Peer Reviews	
2 Repeatable	<i>Project management processes</i>	Requirements Management Software Project Planning Software Project Tracking & Oversight Software Subcontract Management Software Quality Assurance Software Configuration Management	
1 Initial	<i>Competent people and heroics</i>		



## What the CMM Can Do

A model, such as the CMM, can

- aid communication by establishing a common language
- focus your attention
- provide general recommendations
- help prioritize actions
  - the roadmap in the maturity levels
- provide a framework for measurement, tracking, and benchmarking



## Universal Value?

The CMM can be (and has been) successfully used to provide significant value.

- by customers, suppliers, even individuals in the form of the Personal Software Process<sup>SM</sup>!
- for any size organization or project, any application domain, any business context

... but you do have to apply **professional judgment**.

- the CMM user needs knowledge and experience in software engineering

9



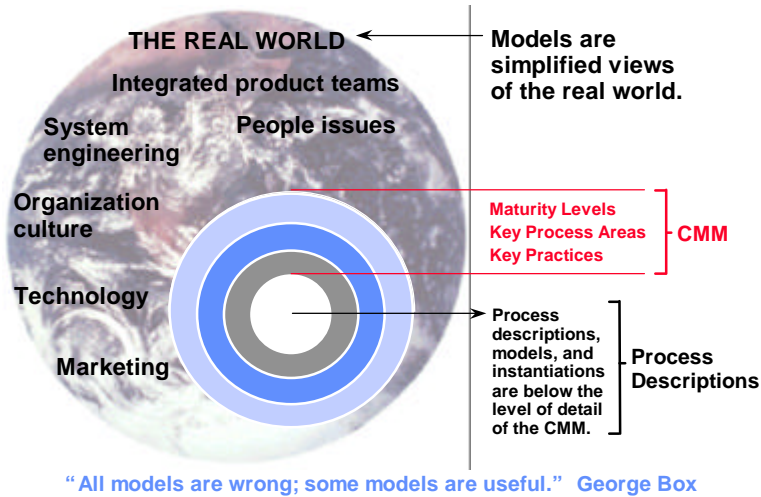
## A Need for Improvement?

Why is the organization interested in using the Software CMM?

- **desire to improve process**
  - direct tie to business objectives
  - willingness to invest in improvement
- **flavor of the month**
  - prescription for disaster!
- **customer concerns about process performance**
  - leading to collaborative improvement?
- **concern about software capability evaluations**
  - cost-effective for small organizations?

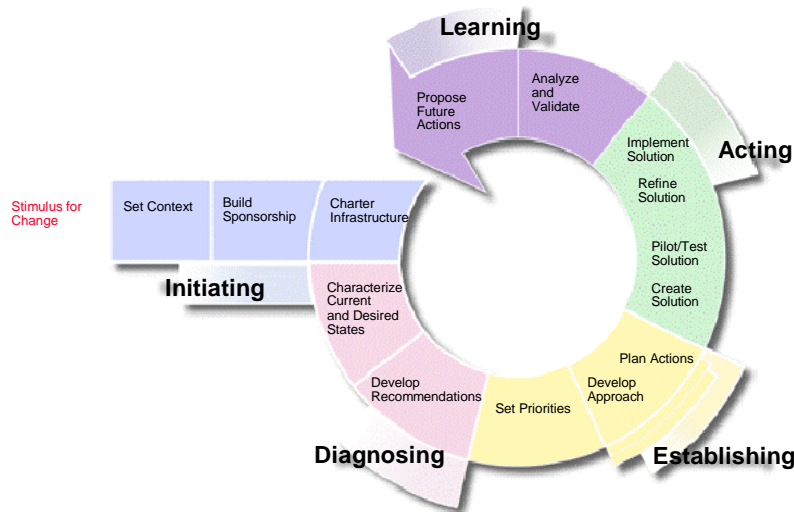
10

## “M” is for Model



11

## SEI’s IDEAL<sup>SM</sup> Approach



12



## Topics

The Current State of Affairs

→ **Small Projects & Organizations**

Software CMM Interpretation

Examples of Interpreting the Software CMM

Abusing the Software CMM

Concluding Thoughts

13



## An Example Context for Using the CMM

Frequently asked question:

**Can the Software CMM be used for small projects (or small organizations)?**

14



## Defining “Small”

Is a small project (or team)

- 2-3 professionals? 4-7? fewer than 25?

Operating for a small period of time

- 2-3 months? 5-6? less than a year?

For a small organization

- fewer than 10 employees? 25? 100?

Result of CMM Tailoring workshop (1995) was conclusion that we could not even agree on what “small” really meant!

15



## Variations of “Small”

**Small** = 3-5 people 6-month project

**Very small** = 2-3 people 4-month project

**Tiny** = 1-2 people 2-month project

**Individual** = 1 person 1-week project

**Ridiculous** = 1 person 1-hour project

- *distinguish between a task and a project!*

Team Software Process (TSP)

Personal Software Process (PSP)

16





## Challenges for the “Small”

The primary business objective of small organizations? **Survive!**

Problems in initiating process improvement?

- deciding the status quo is unsatisfactory....
- deciding process improvement will help....
- **finding the resources and assigning responsibility for process improvement!**

Problems in following through?

- **finding the resources and assigning responsibility for defining and deploying processes!**

17



## Small Organization Culture?

We are all **competent**....

- people were hired to do the job
- we can't afford training - time or money

We all **communicate** with one another....

- osmosis works because we're so “close”

We are all **heroes**....

- we do whatever needs to be done
- rules don't apply to us (they just get in the way of getting the job done)
- we live with short cycle times and high stress

18



## Challenges for the “Small”

Handling requirements -- documented?

Managing projects -- management experience?

Allocating resources -- we're small!

Providing training -- everyone is competent here!

Conducting reviews -- who's qualified/available?

Generating documentation -- when there's time .....

Measuring progress -- the great unknown ....

19



## Small Is Beautiful

Although there are massive problems that may require large numbers of people to solve . . . .

Small teams can be much more productive than large teams.

- teams jell quicker
- fewer communication problems
- ideal team size fewer than 10 people

**Is process discipline needed for small teams?**

- what do we mean by discipline?

20



## Assessing “Small” Organizations

**Use a streamlined assessment process.**

- **a two-week CBA IPI is probably excessive**
  - less rigorous assessments should identify important problems, but may miss some
- **focus on institutionalization practices appropriate to the organization**
- **remember to look beyond the Software CMM**
  - business needs, not just an appraisal
  - people and technology issues
- **perform a readiness survey before trying to begin the improvement cycle**
  - dissatisfaction with the status quo is needed to drive change

21



## Where the Rubber Meets the Road

**Use the Software CMM as a **guide**, not a **dictate**.**

- **tie process improvement to business goals**

**CMMs are about management, communication, and coordination.**

**Keep process documentation concise and simple.**

- **cannot eliminate basic process definition**

22



## Improving “Small” Projects

**Watts Humphrey is currently working on the Team Software Process (TSP).**

**The Personal Software Process (PSP) demonstrates the applicability and validity of the process discipline for individual efforts.**

**TSP and PSP are applications of CMM concepts to the micro-level of the organization.**

- **demonstrate that we can be “level 5 professionals!”**

23



## Topics

**The Current State of Affairs**

**Small Projects & Organizations**

**→ Software CMM Interpretation**

**Examples of Interpreting the Software CMM**

**Abusing the Software CMM**

**Concluding Thoughts**

24



## Where Does CMM Apply?

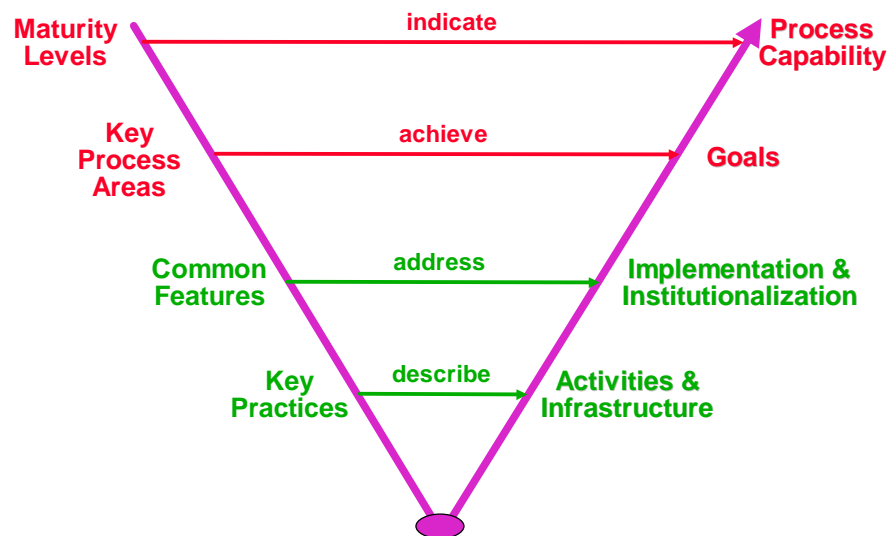
Software CMM was written to provide good software engineering and management practices for any project in any environment.

- model described in **hierarchy**
- detailed practices primarily support large, contracting software organizations
- “normative” components of the CMM are maturity levels, key process areas, and goals
- **all practices in the CMM are informative!**
- organizational learning prevents reinventing the wheel<sup>®</sup> repeatable processes

25



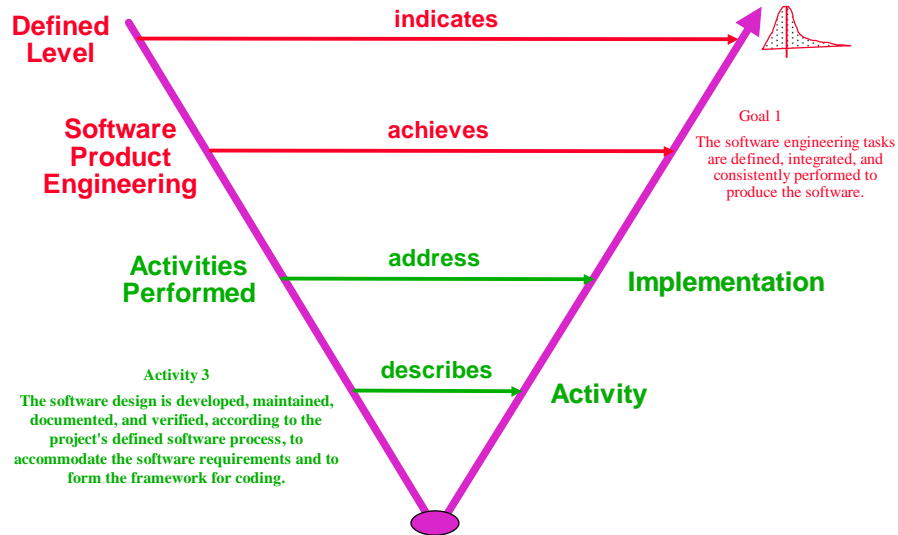
## The CMM Structure



26



## An Example of CMM Structure



27



## CMM Practice Granularity

### *CMMs*

### *Unknown? CMM Integration!*

<b>Maturity Levels</b>	<b>5</b>
<b>Key Process Areas</b>	<b>18</b>
<b>Goals</b>	<b>52 (2-4 per KPA)</b>
<b>Key Practices</b>	<b>316 total</b> <b>150 KPs at ML2/3</b> <b>62 Activities at ML2/3</b>

28



## The CMM Is 500 Pages Long!

Key practices, subpractices, examples, etc., provide guidance for interpreting the CMM.

- much of the guidance is directed at large projects and large organizations

Documentation is important.

- documents need not be lengthy or complex

Training, resources, tools, policies, oversight, measurement, etc., are important.

- institutionalization need not be intrusive
  - culture is “the way we do things around here”

29



## “What” Versus “How To”

Software CMM is intended to be

- **descriptive** of software engineering and management practices
- **prescriptive** for process improvement priorities

Key process areas describe “what” not “how.”

- ignorance of “how” to implement processes can lead to “ticking off” CMM practices
- particularly a problem for technical people promoted to management positions
  - different skill set than what they excel at

30



## Using the CMM Correctly

### Correct use of the CMM implies

- reflecting the reality of your **business environment**
  - tailoring (interpreting) the CMM to suit your context and needs
  - allowing for professional judgment
- **identifying problems** as objectively as possible
- **thinking** and analyzing how the CMM applies
  - doing and not just thinking!
  - not forcing foolish decisions!
- supporting **worker participation** and **empowerment**

31



## Using the CMM Effectively

### Effective use of the CMM implies

- **admitting** the existence of critical problems in your software process
  - understanding that there are non-CMM problems
- having the **will to change**
  - balancing stability and change in improvement
- doing **organizational learning**
  - do it, then improve it
- **investing** time and money to make change happen

32





## The Business Environment ... .

**Environments where interpretation and tailoring are needed**

- very large programs
- virtual projects or organizations
- geographically distributed projects
- rapid prototyping projects
- research and development organizations
- software services organizations
- small projects and organizations

... pretty much everywhere!

33



## The Interpretation Guidance for Small Projects & Organizations ....

**Is Also Applicable to  
Large Projects & Organizations!**

**All projects are different ....**

**All projects are the same ....**

**Organizational learning is the lesson!**

34



## Topics

The Current State of Affairs

Small Projects & Organizations

Software CMM Interpretation

➔ **Examples of Interpreting the Software CMM**

Abusing the Software CMM

Concluding Thoughts

35



## Interpreting and Tailoring

Develop a mapping between CMM terminology and the language used by the organization.

- **organizational structures**
  - independent groups (SQA, testing, SCM)
- **roles and relationships**
  - project manager
  - project software manager
  - customer (internal? external?)
- **formality**
  - frequency of periodic, event-driven
  - granularity of procedures, plans, etc.
  - scope of processes (e.g., subcontracting)

36



## Invariants of Process Discipline

**Assume key process areas and goals are always relevant to any environment.**

- **Software Subcontract Management** may be “not applicable” if no subcontracting
- in contrast, **Peer Reviews** cannot be reasonably tailored out for a level 3 organization

**Some “informative” practices should always be present, some are context-sensitive, and sometimes it depends ....**

- professional judgment and trained, experienced assessors are crucial

37



## Always Needed!

**Never seen a case where the following were not needed (though implementations differ)**

- **documented customer (system) requirements**
- **communication with customer (and end users)**
- **agreed-to commitments**
- **planning**
- **documented processes**
- **work breakdown structure**

38



## Context-Sensitive?

### “Large-project implementations”

- **SCM group and Change Control Board**
  - but configuration management necessary
- **independent SQA group**
  - but objective verification necessary
- **(independent) testing group**
  - but testing necessary

Many context-sensitive, large project implementation issues relate to organizational structure (read the CMM definition of “group”)

39



## It Depends .... Even for “Small”

### Use of historical data in planning

- use work packages directly in estimating small efforts

### Training

- may be through external sources rather than internally developed
- training on internal processes may still be necessary

### Risk management

- complete failure of the project may be a minor risk

40



## Management Sponsorship

Management must actively support improvement.

- dissatisfaction with status quo
- establish expectations
- provide improvement resources
- **pay attention!**

Alternative is ad hoc islands of improvement.

**However.... in small organizations the president/CEO is the primary role model, but a respected “champion” frequently has the influence to move the entire organization!**

41



## Planning

**The #1 factor in successful process definition and improvement is “planfulness.”**

- Bill Curtis, "The Factor Structure of the CMM and Other Latent Issues," 1996 Software Engineering Process Group Conference

**Planning is needed for every major software process.**

- within the bounds of reasonable judgment, the organization determines what is “major”
- **packaging of plans is an organizational decision**

42



## Risk Management

### Project management = risk management?

Both order and chaos have a place

- keep the system in balance so it can change and grow

**Incremental** and **evolutionary** life cycles

- phased approach to delivering the product
- address requirements volatility

43



## Process Focus

### Not just a level 3 concern

Software engineering process group

- competent, respected staff with good interpersonal skills
- focus for following through, action, change
- part-time participants (worker participation!)
  - small organizations may not have full-time SEPG staff at all

Align with any other TQM initiatives

44



## Focusing on Improvement

Identify the **process owner** - establish clear responsibility.

- process improvement team is primarily process implementers

SEPG has a **coordination** and **communication** oriented role.

- in very small organizations, “process focus” may be assigned to an existing manager
- central, accessible repository for current process descriptions, e.g., Web, database

45



## Documented Processes

Granularity, scope, and detail of procedures and standards should be useful, not onerous.

- packaging and formality are organizational decisions
- if the process is “there,” then its existence can be demonstrated to an appraiser!
  - appraisers look for the audit trail
  - appraisers look for knowledge of the process
    - ® communication and consistency
- making the appraiser’s job easy is nice but not necessary
  - **address problems, not practices!**

46



## Process Definition

### Keep it simple!

- identify process owners
- rule of thumb: process descriptions should be **1-2 pages** long
  - reference subprocesses, procedures, standards, and checklists as needed
- rule of thumb: **2-3 tasks per week** at most in the bottom-level process description
  - procedures, standards, and checklists may be more detailed, but they are task-focused
- remember your software design principles
  - locality, information hiding, abstraction, ....
- mix of graphics and text
  - ETVX, EITVOX, IDEF0, Information Mapping, ....

47



## Deploying Processes

**Buy-in for the documented process is critical to successful deployment.**

- process implementers must be part of process definition and improvement

**Don't force implementation of a "bad" process.**

- find out what the problems are
  - "as is" versus "should be" processes
  - pilot processes before broadscale deployment
- know
  - where you are and where you want to be
  - how you're going to get there and how you will recognize success
- automate where possible

48





## Training

**Necessary to effectively deploy and support processes**

- **training (and mentoring) in the process is crucial to institutionalization**

**True need is **skills**, not training!**

- **crucial to professional development and employee retention**

**Choosing between internally developed and externally provided training is an organizational decision.**

49



## Customer-Supplier Relationship

**Talk to the customer.**

- **communication, coordination, and integrity**

**Software Capability Evaluations are driven by a customer need!**

- **build the supplier base -- even by industry**

**Communication and coordination are intrinsic to *Requirements Management* and *Intergroup Coordination*.**

50



## Peer Reviews

**Any kind is better than none**

- inspections
- structured walkthroughs

**No longer an argument over whether peer reviews are worthwhile**

- debates are over “how”
- recognizing the value does not mean that we do them systematically
  - need to “walk the walk,” not just “talk the talk”

51



## Successful Improvement

**Success is based on achieving business objectives.**

- customer satisfaction/delight
- decreased cycle time
- increased productivity

**Don't forget to build a product that customers will want to buy!**

52



## LOGOS Tailored CMM

**For small businesses, small organizations, and small projects**

- by Judith Brodman and Donna Johnson
- Version 1.0 published August 1996

**Modifications to the Software CMM**

- clarification of existing practices
- exaggeration of the obvious
- introduction of alternative practices
- alignment of practices with “small” structure and resources

53



## LOGOS Points to Remember

**Shortcuts, such as templates and checklists, for documentation**

**Alternative methods, such as spot checks and resource sharing, for activities**

**Combined roles for agents of activities**

**Manual methods or basic tools**

54



## Topics

The Current State of Affairs

Small Projects & Organizations

Software CMM Interpretation

Examples of Interpreting the Software CMM

→ **Abusing the Software CMM**

Concluding Thoughts

55



## Using the CMM Improperly

Improper uses of the CMM include

- checking off (sub)practices for conformance
- mandating processes from above: not involving the true process owners – the workers
- riding roughshod over reasonable concerns
- confusing

*Value judgments are embedded in the terminology you use to describe your processes!*

documented

guidance  
disciplined

measured

detailed  
onerous  
law  
inflexible  
bureaucracy  
judgmental

56



## Drivers for CMM Abuse

**Unwillingness or inability to interpret, tailor, or apply judgment within organization**

- easy to mandate the key practices
- judgment is needed even for large projects and organizations!
- paranoia about customer intentions and competence

**Ignorance by the customer**

- software capability evaluation (SCE) teams?
- **judgments may differ!**
  - ® risk profile rather than maturity level

57



## SCEs on Small Organizations?

**Questionable whether SCEs for small projects are cost-effective.**

**SCEs for small acquisitions may be of value when the customer wishes to**

- build a stronger supplier base
- build better customer-supplier relations
- understand software acquisition issues better

**SCEs impose a significant overhead on suppliers!**

58



## The Danger of Focusing on Score

“Standards” such as the CMM, ISO 15504 (SPICE), and ISO 9001 can help organizations improve their software process.

Focusing on achieving a maturity level or certification without addressing the underlying process can cause **dysfunctional** behavior.

Maturity levels and certification should be measures of improvement, not goals of improvement.

- need to tie improvement to business needs

59



## Topics

The Current State of Affairs

Small Projects & Organizations

Software CMM Interpretation

Examples of Interpreting the Software CMM

Abusing the Software CMM

→ **Concluding Thoughts**

60



## The Bottom Line

Software process improvement should be done to help the business – not for its own sake.

Improvement means different things to different organizations.

- What are your business goals?
- How do you measure progress?

Improvement is a long-term, strategic effort.

- What is the expected impact on the bottom line?
- How will the impact be measured?

61



## Let Common Sense Prevail!

		Documented Process	
		Yes	No
Common Sense	Yes	<b>Quality</b>	<b><i>Creative Chaos</i></b>
	No	<b><i>Mindless Bureaucracy</i></b>	<b><i>Mindless Chaos</i></b>

With thanks to Sanjiv Ahuja, President and COO of Bellcore.

62



## General SEI Information

SEI Customer Relations +1 (412) 268-5800

SEI FAX number +1 (412) 268-5758

### Internet Address

customer-relations@sei.cmu.edu

### Mailing Address

Customer Relations  
Software Engineering Institute  
Carnegie Mellon University  
4500 Fifth Avenue  
Pittsburgh, PA 15213

63



## Internet Access to SEI

### SEI Web pages

- [www.sei.cmu.edu](http://www.sei.cmu.edu)
- [www.sei.cmu.edu/cmm/](http://www.sei.cmu.edu/cmm/)
- [www.sei.cmu.edu/cmm/cmm.articles.html](http://www.sei.cmu.edu/cmm/cmm.articles.html)

For the latest version of this presentation, see

- [ftp.sei.cmu.edu/pub/cmm/Misc/cmm-small.pdf](ftp://ftp.sei.cmu.edu/pub/cmm/Misc/cmm-small.pdf)



64





## Acronyms - 1

<b>CM</b>	<b>Configuration Management</b>
<b>CM</b>	<b>Capability Model (term used by CMMI)</b>
<b>CMM</b>	<b>Capability Maturity Model</b>
<b>CMMI</b>	<b>CMM Integration</b>
<b>IDEAL</b>	<b>Initiating, Diagnosing, Establishing, Acting, Learning model for continual process improvement</b>
<b>ISO</b>	<b>International Organization for Standardization</b>
<b>ISO 12207</b>	<b>ISO standard for software life cycle processes</b>
<b>ISO 15288</b>	<b>draft ISO standard (currently working draft) for system life cycle processes</b>
<b>ISO 15504</b>	<b>draft ISO standards (currently type 2 technical reports) for software process assessment</b>
<b>ISO 9000,</b> <b>ISO 9001</b>	<b>ISO standards for quality management systems</b>
<b>PSP</b>	<b>Personal Software Process</b>
<b>SCE</b>	<b>Software Capability Evaluation</b>
<b>SCM</b>	<b>Software Configuration Management</b>

65



## Acronyms - 2

<b>SEI</b>	<b>Software Engineering Institute</b>
<b>SEPG</b>	<b>Software Engineering Process Group</b>
<b>SPICE</b>	<b>Software Process Improvement and Capability dEtermination (aka ISO 15504 - Software Process Assessment)</b>
<b>SQA</b>	<b>Software Quality Assurance</b>
<b>SW-CMM</b>	<b>Capability Maturity Model for Software</b>
<b>TQM</b>	<b>Total Quality Management</b>
<b>TSP</b>	<b>Team Software Process</b>

66