Carnegie Mellon University
Software Engineering Institute

A History of the Capability Maturity Model® for Software

Mark C. Paulk

mcp@sei.cmu.edu -or- Mark.Paulk@ieee.org
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890

Carnegie Mellon University
Software Engineering Institute

Topics

→ **Setting Context: the "Prehistory" of the CMM**

"Versions" of the Software CMM

Related Work: ISO Standards

Software CMM v2

CMM Integration

Closing Comments

What Is the Software CMM?

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

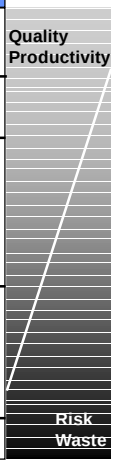
A *community-developed* guide

A model for *organizational* improvement

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

Software CMM v1.1 Key Process Areas

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



Carnegie Mellon University
Software Engineering Institute

A History of the Software CMM

The history of the CMM can be summarized by the different incarnations it has gone through.

- **several challenging decisions were made at different points in CMM development**
- **some issues were identified that remain problematic**
- **"issue slides" were retrieved from the archives**

A fuller picture includes some alternate approaches that were unsuccessful and why.

- **software process domains**
- **normative model**

Carnegie Mellon University
Software Engineering Institute

Inspirations for the Software CMM

Dissatisfaction with known, consistent software problems

Total Quality Management (TQM) successes

Crosby's maturity grid

IBM's process grid

The State of the Practice

"I'd rather have it wrong than have it late. We can always fix it later."

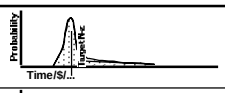
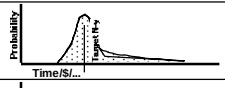
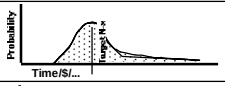
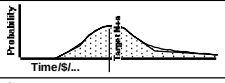
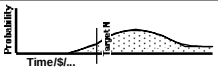
- **A senior software manager (industry)**

"The bottom line is schedule. My promotions and raises are based on meeting schedule first and foremost."

- **A program manager (government)**

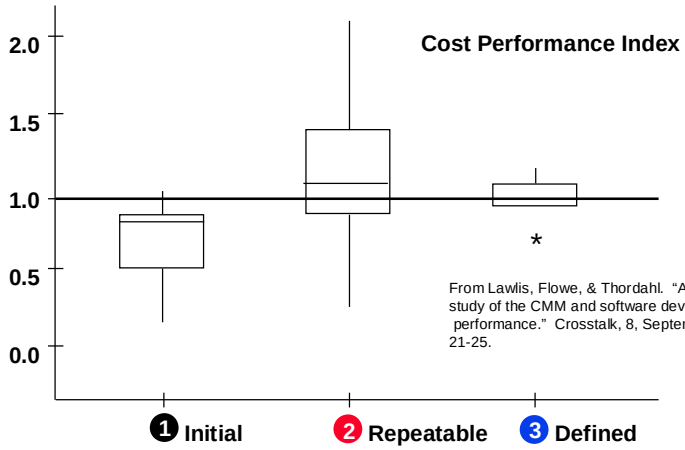
Or see any shrinkwrap software warranty…

Evolution of Process Capability

Level	Process Characteristics	Predicted Performance
5 Optimizing	Process improvement is institutionalized	
4 Managed	Product and process are quantitatively controlled	
3 Defined	Software engineering and management processes defined and integrated	
2 Repeatable	Project management system in place; performance is repeatable	
1 Initial	Process is informal and unpredictable	

Carnegie Mellon University
Software Engineering Institute

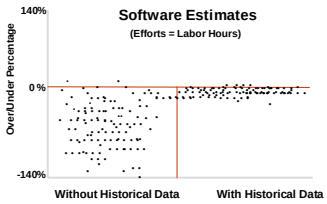
AFIT Study

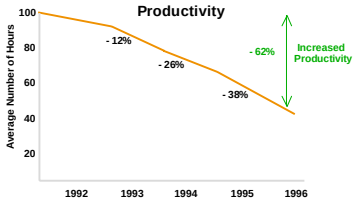


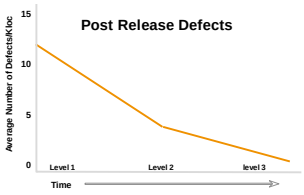
From Lawlis, Flowe, & Thordahl. "A correlational study of the CMM and software development performance." Crosstalk, 8, September 1995, pp. 21-25.

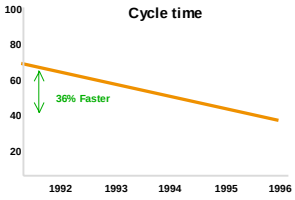
Carnegie Mellon University
Software Engineering Institute

Impact of Software Process Improvement: Boeing Data









John Vu, Boeing, keynote talk at SEPG '97, "Software Process Improvement Journey (From Level 1 to Level 5)"

"Trends" in Quality Results

Maturity Level	Design Faults / KSLOC (Keene)	Delivered Defects / FP (Jones)	Shipped Defects / KSLOC (Krasner)	Relative Defect Density (Williams)	Shipped Defects (Rifkin)
5	0.5	0.05	0.5	0.05	1
4	1	0.14	2.5	0.1	5
3	2	0.27	3.5	0.2	7
2	3	0.44	6	0.4	12
1	5-6	0.75	30	1.0	61

Samuel Keene, "Modeling Software R&M Characteristics." Unpublished report.
Capers Jones, "Software Benchmarking," IEEE Computer, October 1995, pp. 102-103.
Herb Krasner, "Self-Assessment Experience at Lockheed," Third Annual SEPG Workshop, 7 November 1990.
Karl D. Williams, "The Value of Software Improvement... Results! Results! Results!" SPIRE97, 4 June 1997.
Stan Rifkin, "The Business Case for Software Process Improvement," Fifth SEPG National Meeting, 26-29 April 1993.

Impact on Effort

In COCOMO II, the PMAT variable factors in maturity level in terms of decreasing effort/cost.

- **one level change results in 15-21% decrease in effort**

Bradford K. Clark, "Quantifying the Effects on Effort of Software Process Maturity," IEEE Software, November/December 2000.

Donald E. Harter, Mayuram S. Krishnan, and Sandra A. Slaughter, "Effects of Process Maturity on Quality, Cycle Time, and Effort in Software Product Development," Management Science, April 2000.

Carnegie Mellon University
Software Engineering Institute

Myth: Software Problems Are "Technical" Problems

Examined real-life case studies

- **Defense Science Board Task Force on Military Software report, 1987**
- **"Bugs in the Program" report, 1989**
- **red teams, assessments, evaluations, …**

Well-known, consistent problems – revealing a major gap between the state-of-the-art and the state-of-the-practice

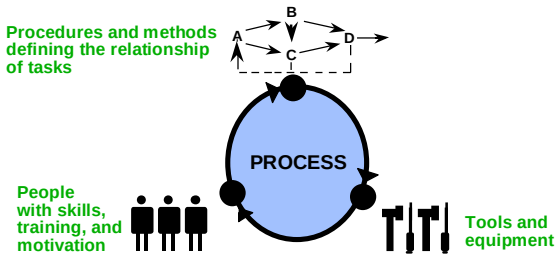
The major problems in software development are managerial – not technical.

Carnegie Mellon University
Software Engineering Institute

Definition of Software Process

Process **– a sequence of steps performed for a given purpose (IEEE)**

Software process **– a set of activities, methods, practices, and transformations that people use to develop and maintain software and the associated products (SEI)**



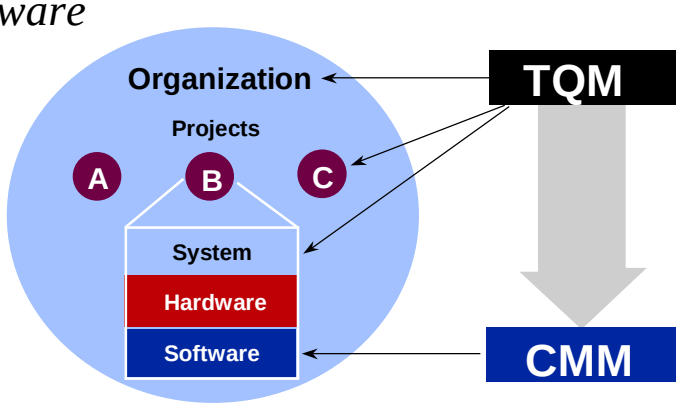
Process Management Premise

The quality of a (software) system is largely governed by the quality of the process used to develop and maintain it.

This premise implies focus on process as well as product.

The value of this premise is visible world-wide in the Total Quality Management movements in the manufacturing and service industries.

Applying Total Quality Management to Software



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

Crosby

Philip Crosby, Quality is Free, 1979.

Quality is measured by the cost of quality which is the expense of nonconformance - the cost of doing things wrong.

Stages
- **Stage 1: Uncertainty**
- **Stage 2: Awakening**
- **Stage 3: Enlightment**
- **Stage 4: Wisdom**
- **Stage 5: Certainty**

Measurement Categories
- **management understanding and attitude**
- **quality organization status**
- **problem handling**
- **cost of quality as percent of sales**
- **quality improvement actions**
- **summation of company quality posture**

IBM Maturity Grid

R.A. Radice, J.T. Harding, et al, "A Programming Process Study," IBM Systems Journal, Vol. 24, No.2, 1985.

Five-Point Scale
- **Traditional**
- **Awareness**
- **Knowledge**
- **Skill & wisdom**
- **Integrated management system**

Process Stages
- **requirements**
- **product level design**
- **component level design**
- **module level design**
- **code**
- **unit test**
- **functional verification test**
- **product verification test**
- **system verification test**
- **package and release**
- **early support program**
- **general availability**

Attributes
- **process**
- **methods**
- **adherence to practices**
- **tools**
- **change control**
- **data gathering**
- **data communication and use**
- **goal setting**
- **quality focus**
- **customer focus**
- **technical awareness**

Carnegie Mellon University
Software Engineering Institute

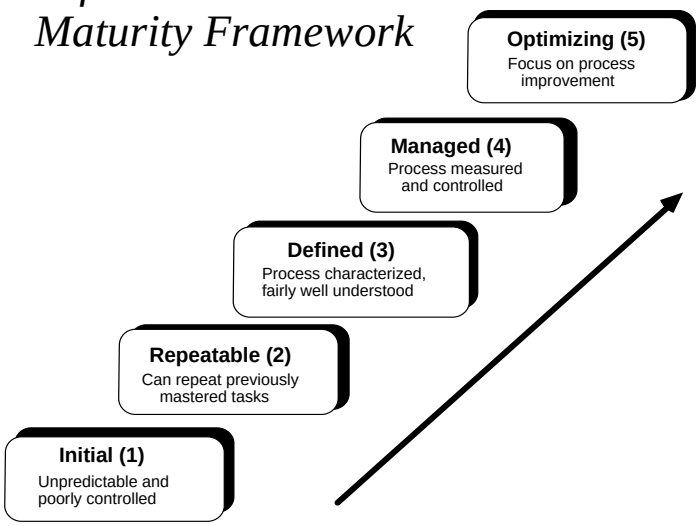
1987: Characterizing the Software Process

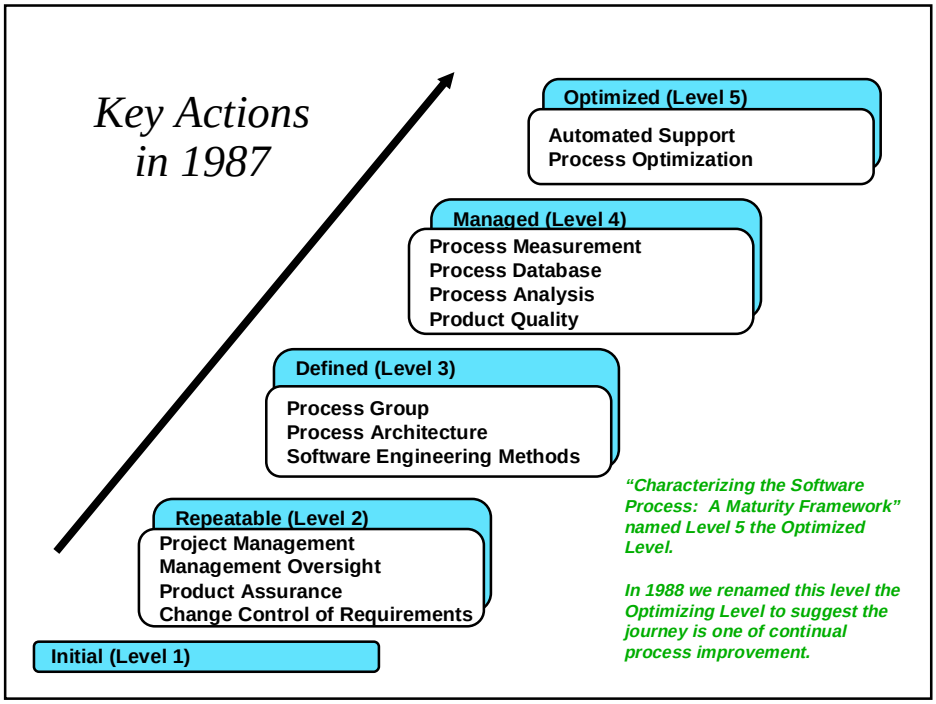
W.S. Humphrey, "Characterizing the Software Process: A Maturity Framework," Software Engineering Institute, CMU/SEI-87-TR-11, DTIC Number ADA182895, June 1987.

W.S. Humphrey and W.L. Sweet, "A Method for Assessing the Software Engineering Capability of Contractors", Software Engineering Institute, CMU/SEI-87-TR-23, DTIC Number ADA187320, September 1987.

Carnegie Mellon University
Software Engineering Institute

Software Process Maturity Framework





Carnegie Mellon University
Software Engineering Institute

Issue: Questionnaire as Model

The 1987 maturity questionnaire was widely viewed as the model.
- **85 process questions**
- **16 technology stage questions**

Technology stage questions were effectively never used.

Appraisal based solely on questionnaire would be superficial.
- ***Bollinger and McGowan, "A Critical Look at Software Capability Evaluations," IEEE Software, July 1991***
- **neither assessments nor appraisals ever relied solely on the questionnaire (as trained)**

Issue: Level of Abstraction

The 1987 software process maturity framework was abstract and incomplete.

Difficult to explain why questions were at a particular maturity level.

Significant interpretation issues existed.
- **"technical reviews" in the questionnaire versus "inspections"**
- **etc.**

Issue: Scoring

If the maturity questionnaire is not the sole basis for scoring, what is the algorithm?

Essentially relied on finding problems, then mapping to related questions as necessary to adjust score.
- **if questions are good probes, there will be some related to any problems**

Issue: Reliability & Consistency

Appraisals were based on an "expert judgment" approach.

Danger of inconsistent results between teams (e.g., two-level differences for the same organization).

Danger of unreliable appraisals by the same team, based on normal human variation.

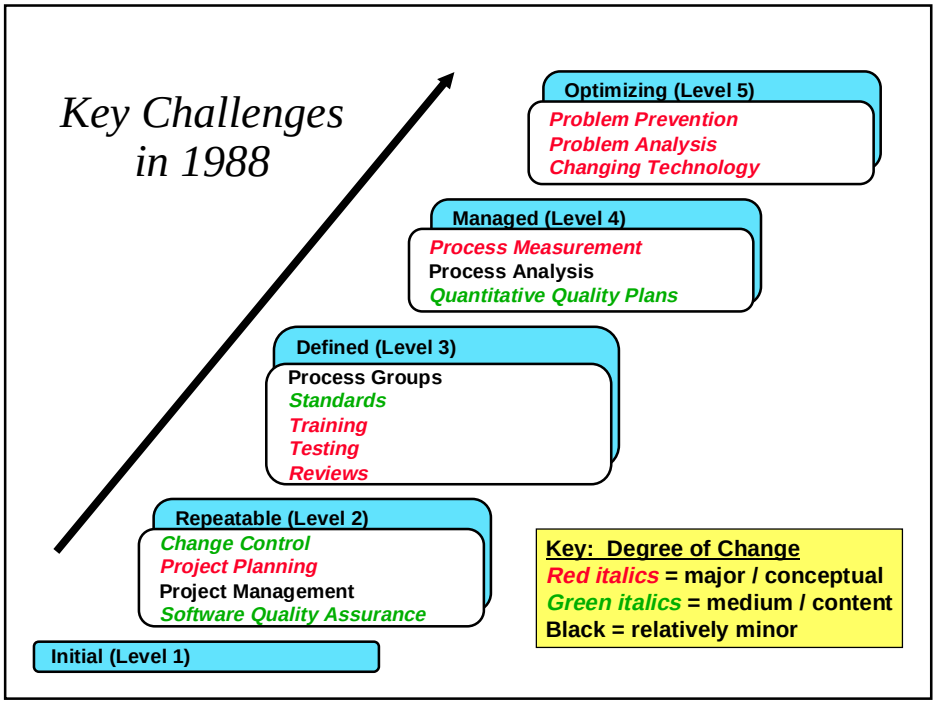
1988: SCE Team Training

In 1988 the SEI developed training courses for software capability evaluation teams.

A two-hour module in the SCE team training described the key challenges in moving between levels.

Concern about reliability and consistency of evaluations drove the inclusion of this module.

- **no equivalent in software process assessment training**



Carnegie Mellon University
Software Engineering Institute

Issue: Level of Detail

SCE training on the maturity model was only two hours.

- **still superficial coverage**
- **still relied heavily on expert judgment**

Carnegie Mellon University
Software Engineering Institute

Issue: Availability of Training

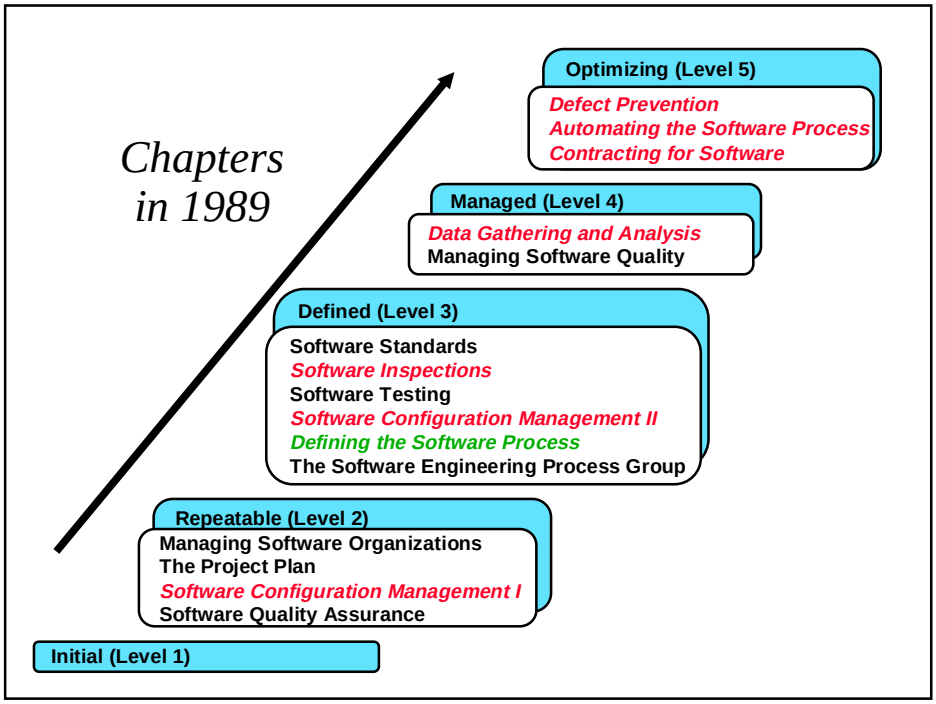
For all practical purposes, SCE training was restricted to government employees performing evaluations.

- **limited number of classes**
- **training materials not available outside class**

Carnegie Mellon University
Software Engineering Institute

1989: Managing the Software Process

Watts Humphrey, Managing the Software Process, Addison-Wesley, Reading, MA, 1989.



Carnegie Mellon University
Software Engineering Institute

Alternative Approaches

Two alternative approaches were attempted for formalizing the software process maturity framework

- **software process domains**
- **normative model**

Both approaches were part-time efforts within the Contractor Software Engineering Capability Assessment (CSECA) project (later SCE project).

Both approaches were abandoned after prototypes were reviewed within the program and/or with external reviewers.

July 1988: Software Process Domains

Planning

Measurement and analysis

Process definition

Verification and validation

Subcontract management

Software management

Technology insertion

Software engineering methods

Training

Selection and retention (of staff)

Issues With Domains

Short domain descriptions, approximately 1-2 pages apiece ⇒ *still ambiguous*

Domains spanned maturity levels ⇒ *complex and ambiguous*

Interdependencies between different domains, especially between practices at different maturity levels in different domains ⇒ *complex and ambiguous*

1989: Normative Model

"Orthogonal" representation of maturity in terms of stability factors and maturity indices applied to unit operations.

Stability Factors
- resources
- training
- tools
- plans
- policies
- responsibility

Unit Operations
- staffing
- committing
- planning
- tracking
- executing
- documenting
- verifying

Maturity Indices
- existence
- review
- selection
- metrics
- analysis
- monitoring

Issue: Usability

While the normative model led to some significant insights affecting the later development of the CMM, it was very difficult to explain.

It was deemed too artificial, or mathematically formal, to be useful for technology transition.

It failed the "usability" test (not easily comprehensible).

Carnegie Mellon University
Software Engineering Institute

Topics

Setting Context: the "Prehistory" of the CMM

→ "Versions" of the Software CMM

Related Work: ISO Standards

Software CMM v2

CMM Integration

Closing Comments

Carnegie Mellon University
Software Engineering Institute

January 1990: CMM v0.2

The first draft of the Software CMM distributed for review outside the SEI.

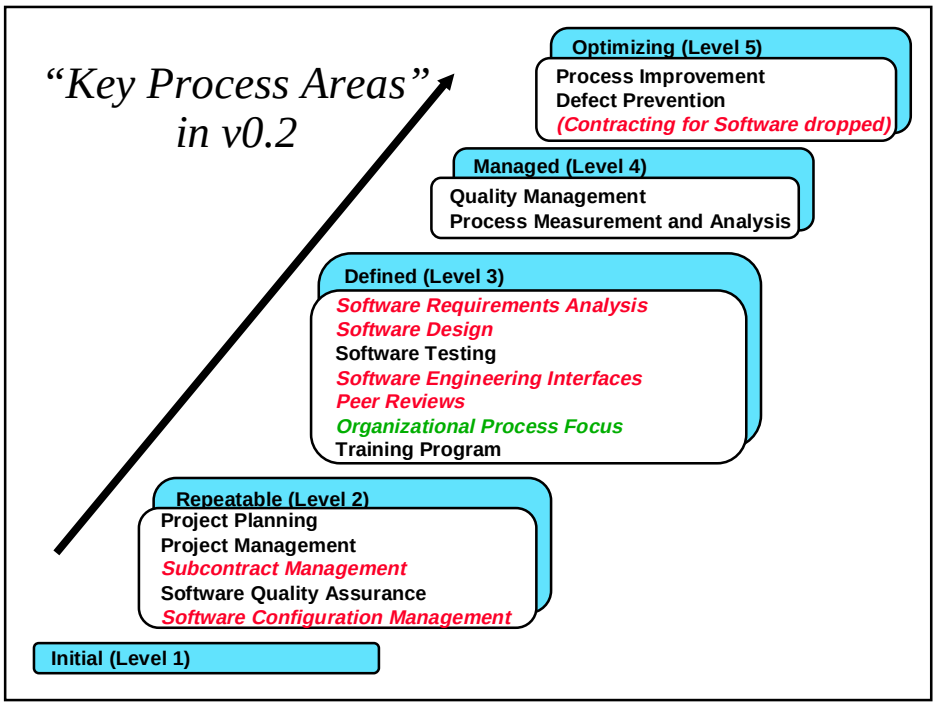
Watts Humphrey was the Program Director for the Software Process Program but had identified Bill Curtis as his replacement.

Key Process Areas

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

Focused on a single maturity level, but practices could span maturity levels (especially subpractices)

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



Carnegie Mellon University
Software Engineering Institute

Issue: KPAs Spanning Levels

Should key process areas span maturity levels?

Feedback that (sub)practices at different levels were confusing.

- **difficult to understand, implement, and transition**
- **focus on moving up maturity levels was obscured**

Encouraged debates over why a practice was at a particular maturity level, debate the precedence of KPAs.

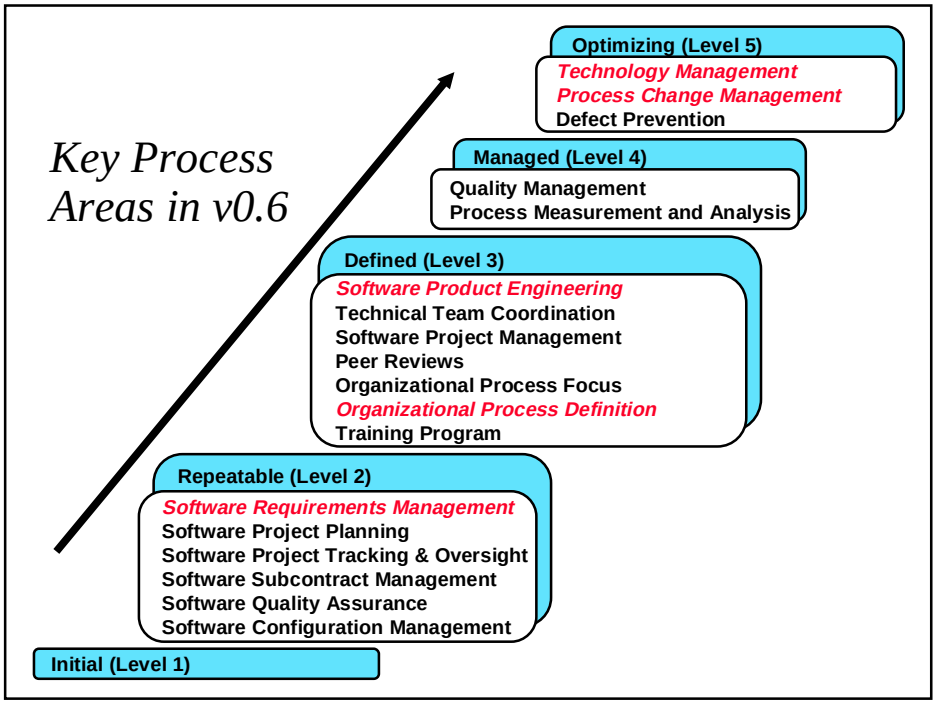
Note that this observation (and this slide) preceded the SEI's work with ISO on continuous architectures and CMMI.

Carnegie Mellon University
Software Engineering Institute

June 1990: CMM v0.6

One of the later drafts of the Software CMM.

Bill Curtis had become Program Director for the Software Process Program.



Carnegie Mellon University
Software Engineering Institute

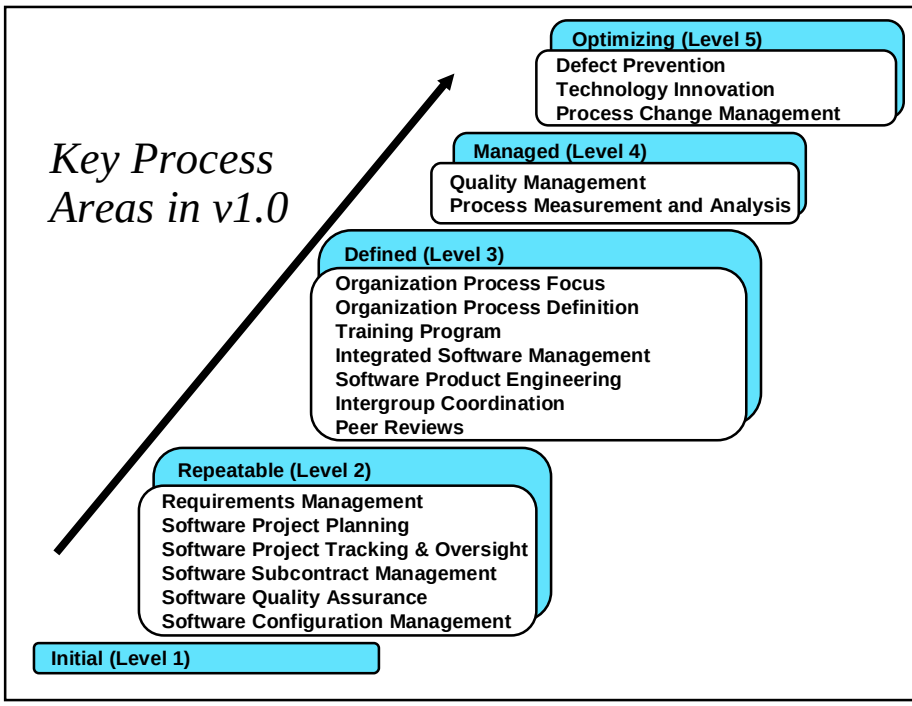
"In earlier drafts of the CMM, key process areas spanned maturity levels but they did not span all maturity levels… The decision was made with version 0.6 to redefine the key process areas as residing at a single maturity level. As a result of defining key process areas as residing at a single maturity level, Software Project Management *[later renamed Integrated Software Management]* was added at level 3 to address software project planning and management issues at Maturity Level 3. This has been one of the more controversial decisions in defining the structure of the CMM… If key process areas span levels, then a more complete picture is provided, but the 'vital few' issues that dominate organizational improvement may be lost in the detail. Also, the emphasis on organization improvement in the CMM means that some processes are prioritized before others. When providing a detailed set of guidelines for process improvement, such as the key practices in the CMM, this unevenness in (described) process capability at different maturity levels becomes more visible and distracting than when the emphasis is explicitly placed on the 'key' areas that build organizational capability. Both perspectives are valuable, but the CMM's explicit target is organizations."

Mark C. Paulk, "The Evolution of the SEI's Capability Maturity Model for Software," Software Process: Improvement and Practice, Spring 1995.

1991: CMM v1.0

M.C. Paulk, B. Curtis, M.B. Chrissis, et al., "Capability Maturity Model for Software," Software Engineering Institute, CMU/SEI-91-TR-24, August, 1991.

C.V. Weber, M.C. Paulk, C.J. Wise, and J.V. Withey, "Key Practices of the Capability Maturity Model," Software Engineering Institute, CMU/SEI-91-TR-25, August, 1991.



Issue: Scope

What should the scope of the CMM be?

How far should we go into
- **people issues?**
- **technology issues?**
- **concurrent engineering?**
- **organization culture/change?**

Many TQM issues not addressed in CMM as scoping decision (software-specific)
- **issues that are important for effective change and improvement**

Issue: Requirements Elicitation

Whose job is it to identify what the software system should do?

Is requirements elicitation a systems engineering problem?

Which definition of quality will we use?
- **conformance to requirements?**
- **customer satisfaction?**
- **customer delight?**

Carnegie Mellon University
Software Engineering Institute

Issue: An Audit Checklist

Is the CMM too detailed?

There is an enormous amount of information in this 500-page document.

Danger of CMM being used as an audit checklist.

Danger of CMM being inappropriately used in environments different than that for which the key practices were written.

Carnegie Mellon University
Software Engineering Institute

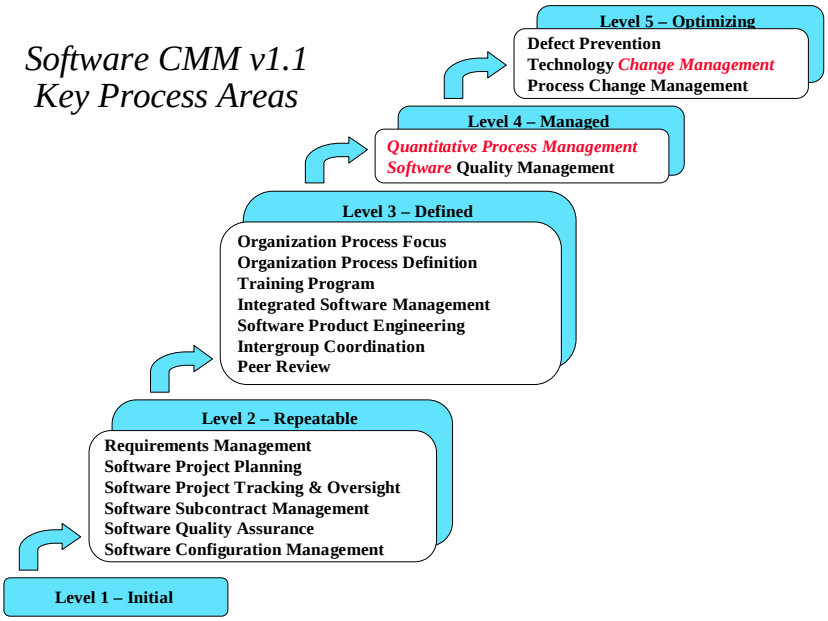
1993: Software CMM v1.1

CMM v1.1 was a "minor" revision of v1.0 ...

There were name changes of KPAs and common features, but the content did not change significantly.

Almost every practice in the CMM was revised to improve consistency and clarity.

Software CMM v1.1
Key Process Areas



Issue: The Customer's Maturity

Can a Level 1 acquisition agency effectively manage a Level 5 software supplier?

Without crippling the supplier's software process?

Issue: Maintenance

Approximately 75% of software life cycle costs are in maintenance (sustaining engineering).

The CMM addresses the maintenance environment when appropriately interpreted.

- **focus of practices is on the development environment**
- **as part of v1.1 revision, some practices were rewritten to make them more friendly to the maintenance environment**

Issue: Appraiser Qualifications

What should the requirements be for becoming a licensed CMM-based appraiser?

Knowledge of the CMM?

Taking a CMM course (i.e., the Introduction to CMM)?

Auditing, interviewing, and reading skills?

Issue: Focusing on Level Number

Some organizations focus on what their maturity level is.

- **senior executives may legitimately be interested in only the level of abstraction represented by the level**
- **most managers should be more interested in the key process area profile and the problems represented thereby**

Danger is focusing on a score rather than on software process improvement.

Topics

Setting Context: the "Prehistory" of the CMM

"Versions" of the Software CMM

→ **Related Work: ISO Standards**

Software CMM v2

CMM Integration

Concluding Comments

Carnegie Mellon University
Software Engineering Institute

Related Efforts – the "Quagmire"

"Children" of CMM / SPA / SCE
- **Trillium**
- **Software Technology Diagnostic**
- **Healthcheck**
- **etc.**

Government-procurement methods
- **SDC/CR**
- **SDCE**
- **etc.**

ISO standards – 9001, 15504, 12207,… 15288

Carnegie Mellon University
Software Engineering Institute

CMM Gaps Identified in ISO 9001

4.7 Purchaser-Supplied Product
- **purchaser-supplied and commercial-off-the-shelf (COTS) software addressed only in context of planning (ISM.AC.6.3) – use SSM**

4.15 Handling, Storage, Packaging, and Delivery
- **acceptance testing (SPE.AC.7) and release (SCM.AC.7) addressed**
- **installation process – including handling, storage, packaging, and delivery – not specifically addressed**

4.19 Servicing
- **ISO 9000-3 interprets servicing as maintenance (sustaining engineering)**
- **maintenance not separate component in CMM**

International Standards Organization (ISO), 1991 Plenary

Proposal from Alec Dorling of the U.K. for a study period to examine whether consensus existed for creating an international standard on software process assessment

WG10 eventually established to build the standard

SPICE (Software Process Improvement and Capability dEtermination) is the name of the development and trialling group

Original SPICE Project Schedule

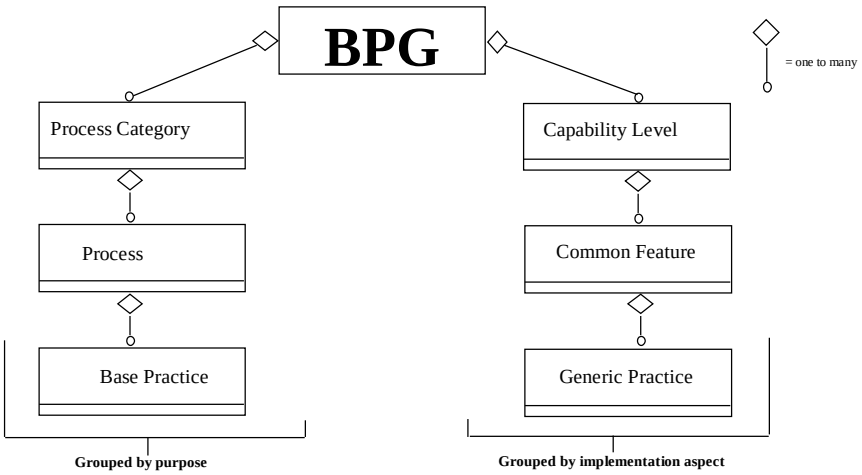
Developing drafts of standard components
- **2Q 1993 through 4Q 1994**

Key components of standard developed first
- **Baseline Practices Guide (BPG is reference model) and Process Assessment Guide completed 2Q 1994**

Release to WG10 in 4Q 1994 ***(done in 1995)***
- **2 months of reviews**
- **20+ months of trials (SPICE product testing and revision)**

International Standards ready for balloting in 1996 timeframe ***(type 2 technical reports finished in 1998, IS ballot planned for 2003)***

Introducing Continuous Architecture



Two Architectural Perspectives

A "staged" architecture, e.g., the Software CMM
- **focuses on building organizational capability**
- **identifies the vital few issues to focus on**
- **describes a roadmap for process improvement**

The staged architecture was designed for changing organizational behavior.

A "continuous" architecture, e.g., ISO/IEC 15504
- **focuses on building process capability**
- **provides a reference model for rating processes**
- **describes the terrain of process management**

The continuous architecture was designed for comparing different models.

Topics

Setting Context: the "Prehistory" of the CMM

"Versions" of the Software CMM, 1987 to 1993

Related Work: ISO Standards

→ **Software CMM v2**

CMM Integration

Closing Comments

Drivers for Software CMM v2

Address change requests from users.

Continual improvement of the Software CMM
- **respond to growing/changing needs**
- **improved understanding of "best practices"**
- **improved understanding of maturity levels 4 and 5**

Harmonize with relevant national and international standards (and other CMMs)
- **minimize unnecessary differences**
- **provide mappings**

Sources of Change

Change requests

Workshops
- **February 1995 brainstorming workshop**
- **November 1995 requirements workshop**
- **April 1996 maturity level 4 & 5 workshop**

Mappings to standards, other CMMs, working papers from community

Discussions with high maturity organizations

Pilots of prototypes

Software CMM Version 2

Balance between conflicting requirements
- **stability of CMM for organizations engaged in software process improvement**
- **need to continually improve CMM to address user needs**

Version 2 planned for release in November 1997

Integrating Other Maturity Models

A number of other models based on CMM being developed to address the specific needs of

- **systems engineering (SE-CMM)**
- **people (PM-CMM)**
- **software acquisition (SA-CMM)**
- **discipline of engineering (engineering maturity model or EMM)**

Both staged and continuous architecture models had been characterized as "CMMs."

Model integration working group at SEI was studying model integration and generalization.

The Revision Process

Revision of the Software CMM involved

- **collecting change requests**
- **identifying new practices and key process areas needed by users of the CMM**
- **distributing prototype key process areas for review**
- **pilot testing**
- **holding CMM workshops**
- **distributing draft versions of CMM v2 for review**
- **recommendations from CMM Advisory Board**
- **decisions by CMM Change Control Board**

Carnegie Mellon University
Software Engineering Institute

Software CMM v2 Release Halted

In October 1997, SEI's sponsor, the Office of the Under Secretary of Defense for Acquisition and Technology, directed that the Software CMM Version 2 release be halted in favor of work on CMM Integration.

One of the source documents for CMMI is Software CMM v2C.

Carnegie Mellon University
Software Engineering Institute

Topics

Setting Context: the "Prehistory" of the CMM

"Versions" of the Software CMM

Related Work: ISO Standards

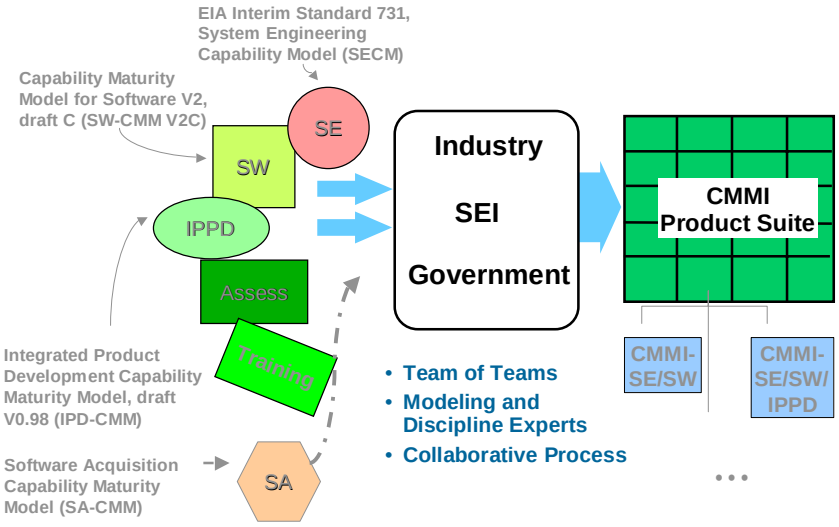
Software CMM v2

→ **CMM Integration**

Closing Comments

Carnegie Mellon University
Software Engineering Institute

CMMI Source Models



Carnegie Mellon University
Software Engineering Institute

Model Comparisons

Release	PAs/ FAs		Goals/ Themes*		Activities/ Practices**	
SW-CMM V1.1	18		52		316	
SW-CMM V2C	19	38	62	139	318	701
EIA/IS 731	19		77		383	
IPD-CMM V0.98	23		60		865	
CMMI V0.1 SE/SW	27		149		550	
CMMI V0.2 SE/SW	24		80		528	
CMMI V1.0 SE/SW	22		70		417	

Legend:

* = Ratable components (Maturity Rating (e.g., Level n)

** = Key to implementation effort

Carnegie Mellon University
Software Engineering Institute

Comparing SW-CMM to CMMI Staged Level 2 (Managed)

SW-CMM v1.1 Key Process Areas (KPAs)	CMMI Process Areas (PAs)
Requirements Management	Requirements Management
Software Project Planning	Project Planning
Software Project Tracking & Oversight	Project Monitoring & Control
Software Subcontract Management	Supplier Agreement Management
Software Quality Assurance	Process & Product Quality Assurance
Software Configuration Management	Configuration Management
NEW	Measurement & Analysis

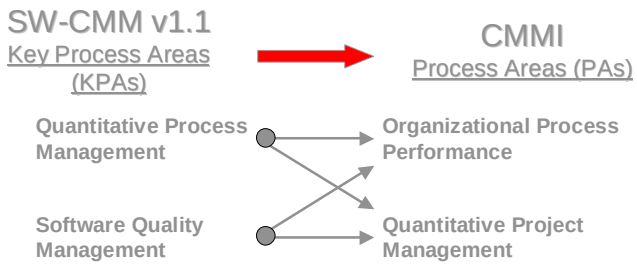
Carnegie Mellon University
Software Engineering Institute

Comparing SW-CMM to CMMI Staged Level 3 (Defined)

SW-CMM v1.1 Key Process Areas (KPAs)	CMMI Process Areas (PAs)
Organization Process Focus	Organizational Process Focus
Organization Process Definition	Organizational Process Definition
Training Program	Organizational Training
Integrated Software Management	Risk Management
	Integrated Project Management
Intergroup Coordination	
Software Product Engineering	Requirements Development
	Technical Solution
	Product Integration
	Validation
Peer Reviews	Verification
From SE CMM	Decision Analysis & Resolution

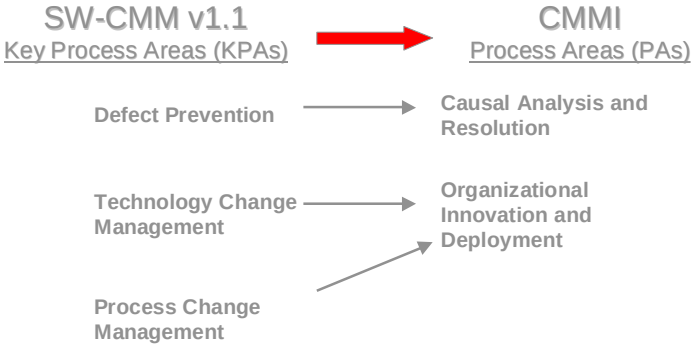
Carnegie Mellon University
Software Engineering Institute

Comparing SW-CMM to CMMI Staged Level 4 (Quantitatively Managed)



Carnegie Mellon University
Software Engineering Institute

Comparing SW-CMM to CMMI Staged Level 5 (Optimizing)



Carnegie Mellon University
Software Engineering Institute

The Stream of Continuous & Staged Architecture Models Leading to CMMI

1979 – Crosby's maturity grid (Quality is Free)
1985 – IBM maturity grid (Radice)
1987 – SEI software process maturity framework
1988 – SEI software process domains
1989 – SEI normative model
1990 – SEI Software CMM v0.2
1990 – SEI Software CMM v0.6
1991 – SEI Software CMM v1.0
1993 – SEI Software CMM v1.1
1995 – SPICE Baseline Practices Guide
1995 – Systems Engineering CMM
1997 – SEI Software CMM v2 Draft C
1998 – EIA 731 (Systems Engineering Capability Model)
1998 – ISO/IEC 15504 type 2 technical reports
2000 – SEI CMM Integration v1.0 (both)

Carnegie Mellon University
Software Engineering Institute

Topics

Setting Context: the "Prehistory" of the CMM

"Versions" of the Software CMM

Related Work: ISO Standards

Software CMM v2

CMM Integration

→ **Closing Comments**

Carnegie Mellon University
Software Engineering Institute

DoD Policy for CMMI

From Dr. Jack Ferguson, Director, Software Intensive Systems (OUSD (AT&L)), at Software Technology Conference, 2001

"CMMI will become the logical integrated successor for the CMM-SW for software engineering and EIA/IS 731 for systems engineering."

"CMMI will become the approved means of judging engineering maturity for procurements within two years."

Carnegie Mellon University
Software Engineering Institute

SEI Policy for CMMI

From the CMMI FAQ, as of 17 September 2001

"The models that are designated as the starting point for the CMMI Product Suite development and identified as source documents will no longer be updated or supported by the issuing organization. The Product Suite is intended to replace the source models. As other disciplines are incorporated into the CMMI Product Suite, they too would follow the same process. As improvements are incorporated into the CMMI Product Suite, the original source documents will become obsolete and less representative of industry practice.

The plan is that such replacement would take place three years after successful model development and full release of the CMMI Product Suite. This replacement is currently scheduled for Fall 2003."

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 15213-3890

Internet Access to SEI

www.sei.cmu.edu
www.sei.cmu.edu/cmm/
www.sei.cmu.edu/cmm/cmm.articles.html
www.sei.cmu.edu/cmm/slides/slides.html
www.sei.cmu.edu/cmmi/

www.sei.cmu.edu/cmm/slides/cmm-history.pdf
- ***www.sei.cmu.edu/cmm/slides/cmm-history-handout.pdf***