Introduction to the SEI Capability Maturity Model® (SEI/CMM®)

Jeff Jacobs
Jeffrey Jacobs & Associates
phone: 650.571.7092
email: jmjacobs@jeffreyjacobs.com
http://www.jeffreyjacobs.com

Survey

- Does your organization have a well defined software development methodology/process?
- Does your organization use RUP?
- Does your organization have a *software engineering process group?*

Agenda

- What is SEI/CMM®?
- History
- Overview and characteristics of the 5 levels
- Structure
- Examples focus on Level 2
  - *Level 2 is the hardest to attain, as it involves fundamental cultural changes*
- A sample spreadsheet/questionnaire
- Caveats and clarifications
- Questions

What is the SEI/CMM®?

- Well defined process and framework for assessing or evaluating the maturity level of an organization
- Organizations may receive a formal assessment from SEI licensed assessors
  - DoD and other organizations may require a formal assessment rating for contractors and partners
- Organizations may use the SEI/CMM® materials to perform internal evaluations and as a basis for improvement
- Describes an evolutionary improvement path to migrate an organization from an ad-hoc, immature organization to mature, disciplined organization

Descriptive, not Prescriptive

- Describes *what* processes and practices should be in place to achieve a maturity level, but not *how* they should be performed
  - “Software Configuration Management” should be performed, but no guidelines on how to do it

History

- The SEI (Software Engineering Institute) is a Federally Funded Research and Development Center (FFRDC) established in 1984 at Carnegie Mellon University by US Department of Defense
  - *To solve the problem of why software projects were always late, over budget and full of bugs*
- November 1986, SEI, in conjunction with MITRE Corporation, began developing a process maturity framework that would help organizations improve their software process
- http://www.sei.cmu.edu

More History

- CMM conceived by Watts Humphrey
- SEI maturity questionnaire — 1987
- Humphrey, W.S. (1989), *Managing the Software Process*, Addison-Wesley, Reading, MA.
- Paulk, M.C., Weber, C., Chrissis, M.B., Garcia, S., & Bush, M. (1992), *Key Practices of the Capability Maturity Model: Version 1.1,* Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.
- http://www.sei.cmu.edu

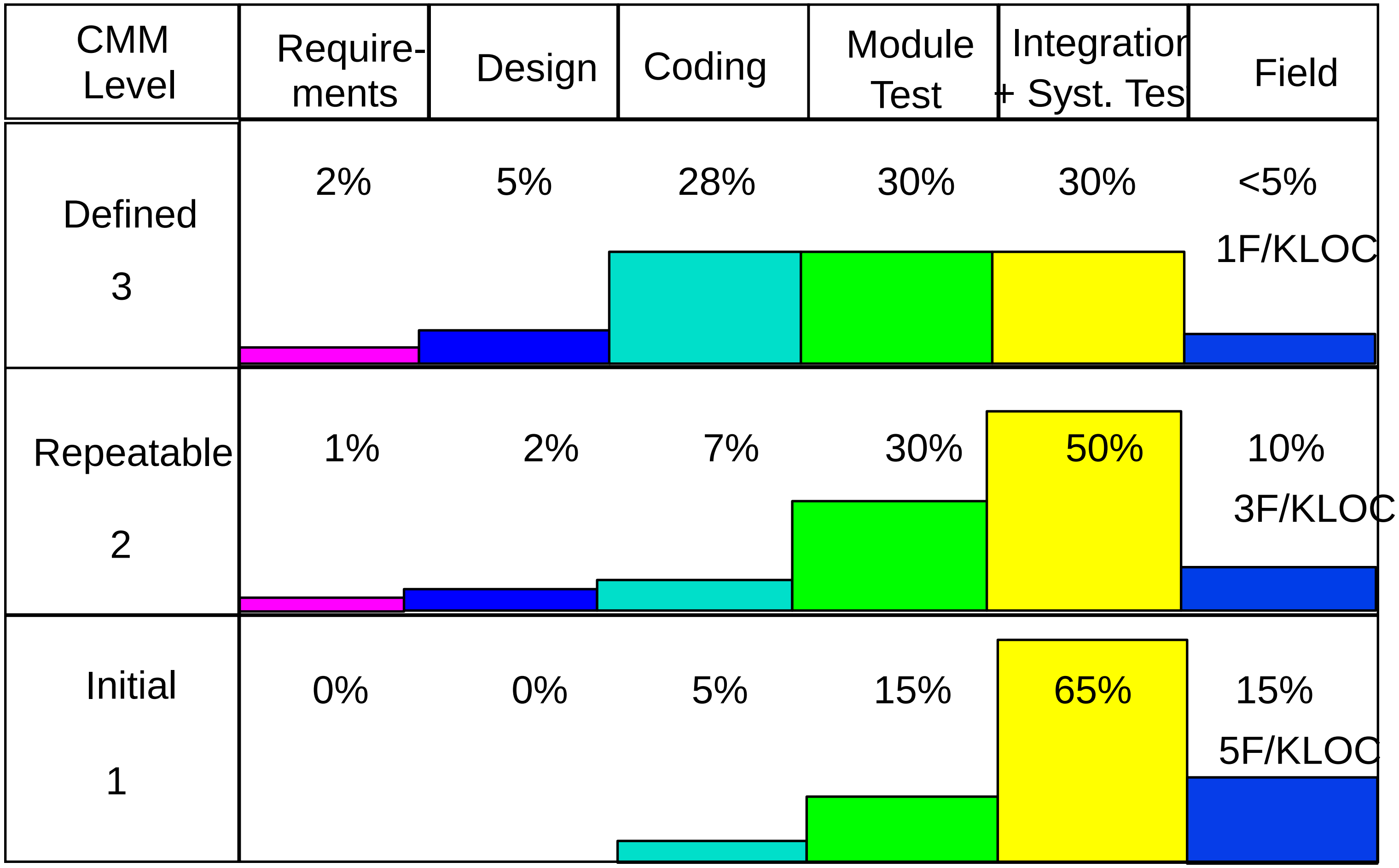
Why Should You Care?

- Does this sound familiar?
  - Success is dependant on heroic efforts, long hours and death marches
  - People complain about workload and stress
  - Processes are improvised and reinvented for every project
  - Constant fire fighting
  - People are rewarded for solving problems, not preventing them
  - There is no time to do it right, but there is always time to do it over

Higher Level Organizations are More Effective

- Fewer defects/higher quality
- Reduced schedule misses
- Fewer/lower cost overruns
- Lower personnel turnover
- Better predictability
- Numerous study results can be found at http://www.sei.cmu.edu

Improved Defect Detection Results



CMM Level	Require-ments	Design	Coding	Module Test	Integratio + Syst. Tes	Field
Defined 3	2%	5%	28%	30%	30%	<5% 1F/KLOC
Repeatable 2	1%	2%	7%	30%	50%	10% 3F/KLOC
Initial 1	0%	0%	5%	15%	65%	15% 5F/KLOC

Source: Möller '95, Jones '96, Curtis '96

Characteristics of a Mature Organization

- Possesses an organization-wide ability to manage software development and maintenance processes
- The processes are *documented* and *communicated* to existing staff and new employees
- Work activities are carried out according to *documented* procedures and process
- The defined processes are *fit for use* and consistent with the way work actually gets done
- The defined processes are updated when necessary; improvements are developed through controlled pilot test or cost/benefit analysis

More Maturity…

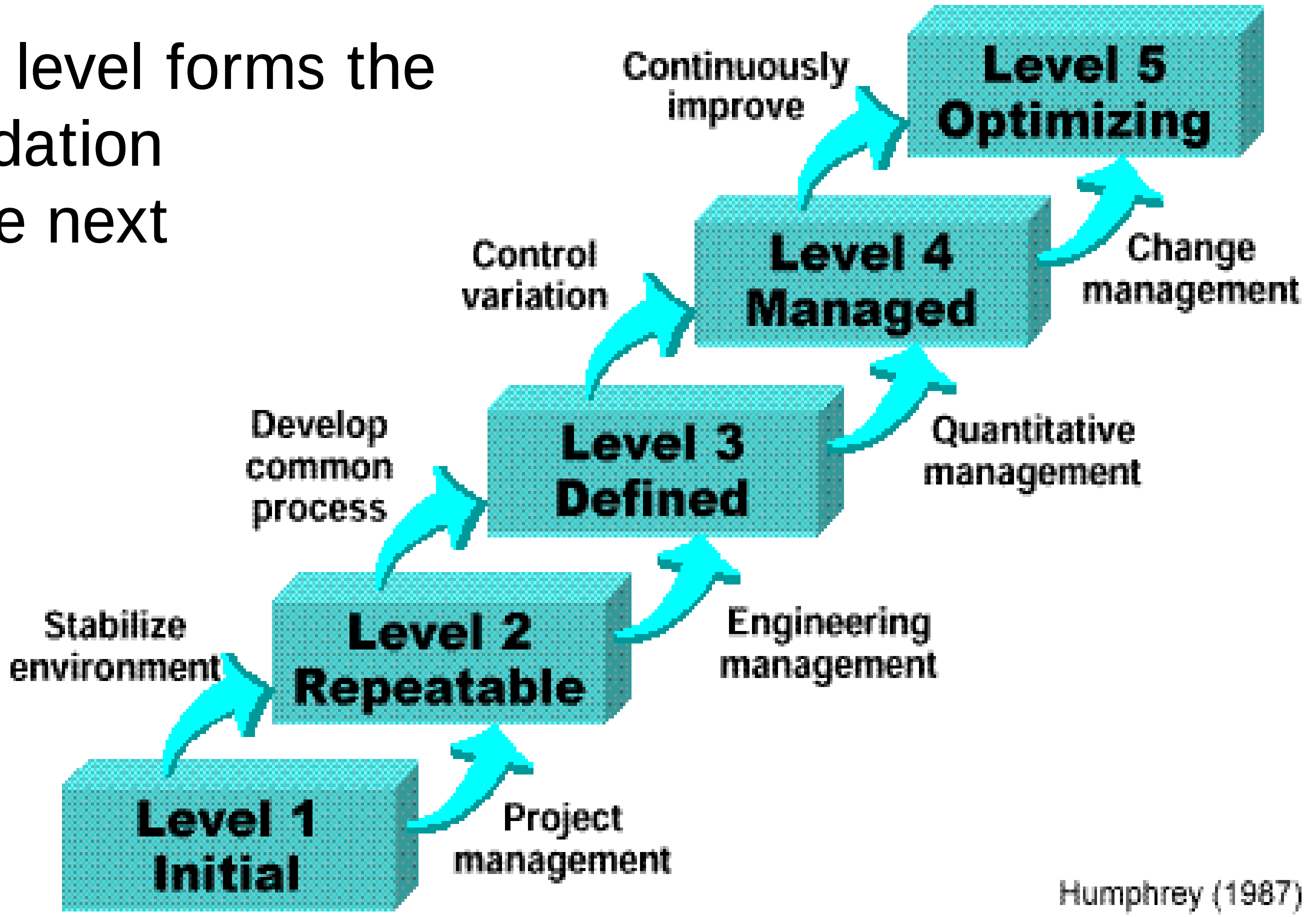
- Roles and responsibilities are clear throughout projects and across the entire organization
- Managers monitor the quality of the software products and customer satisfaction
- There is an objective, quantitative basis for judging product quality and for analyzing problems with the product or process
- Schedules and budgets are based on historical performance and are realistic
- The expected results for cost, schedule, functionality and quality of the product are usually achieved

More Maturity…

- A disciplined process is followed because everyone understands the benefits of doing so and the infrastructure is in place to support it

Overview of the 5 Levels

- Evolves from one level to next
- Each level forms the foundation of the next



Characteristics of Level 1 - Initial

- Ad hoc, chaotic work environment
- Success depends on heroic effort, overtime
- Few processes defined
    - *Even fewer processes and procedures are documented*
- Processes often ignored under schedule pressure
- Missed deadlines, poor quality
- *Most organizations are Level 1*

Characteristics of Level 2 - Repeatable

- Ability to repeat earlier success based on prior experience
- Basic project management processes are established
  - Project planning
  - Project tracking
  - Subcontractor management
- Basic software development processes are in place
  - Requirements management
  - Software quality assurance
  - Software configuration management
- Processes are project oriented

Characteristics of Level 3 - Defined

- Management and engineering processes are standardized and documented across the *organization*
- Projects use approved, tailored versions of the standard process definitions
- A training program exists
- Level 3 focuses on organization wide issues
  - A formal focus for process improvement exists
  - Processes are documented *and maintained*

Characteristics of Level 4 - Managed

- The software process is under statistical control
- Quantitative quality goals for products are established and met
- Strong knowledge and use of metrics and statistical techniques throughout the organization
- Schedule and quality performance is predictable

Characteristics of Level 5 - Optimizing

- "Continuous process improvement is enabled by quantitative feedback and from piloting innovative ideas and new technologies"

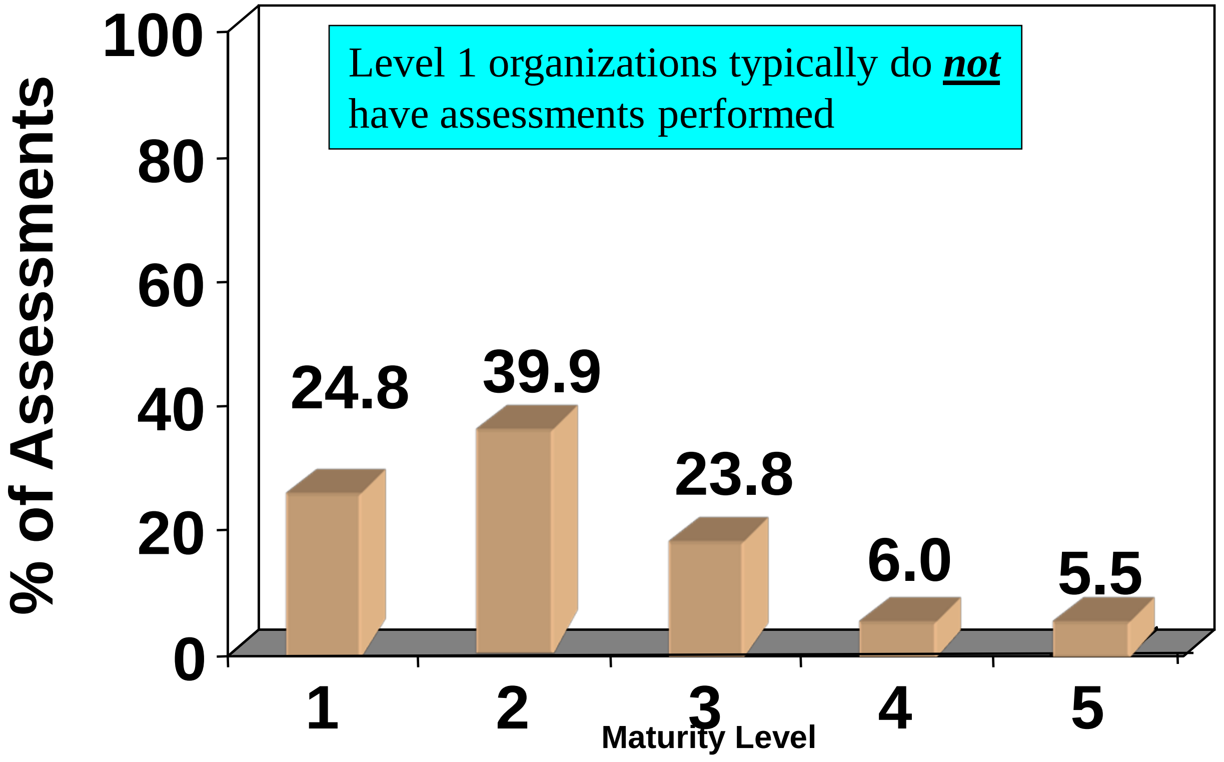
CMM Maturity Distribution

Assessments conducted from 1997 - December 2001

1158	**organizations**
365	**companies**
5624	**projects**
38.0%	**offshore organizations**

1104 organizations reported size

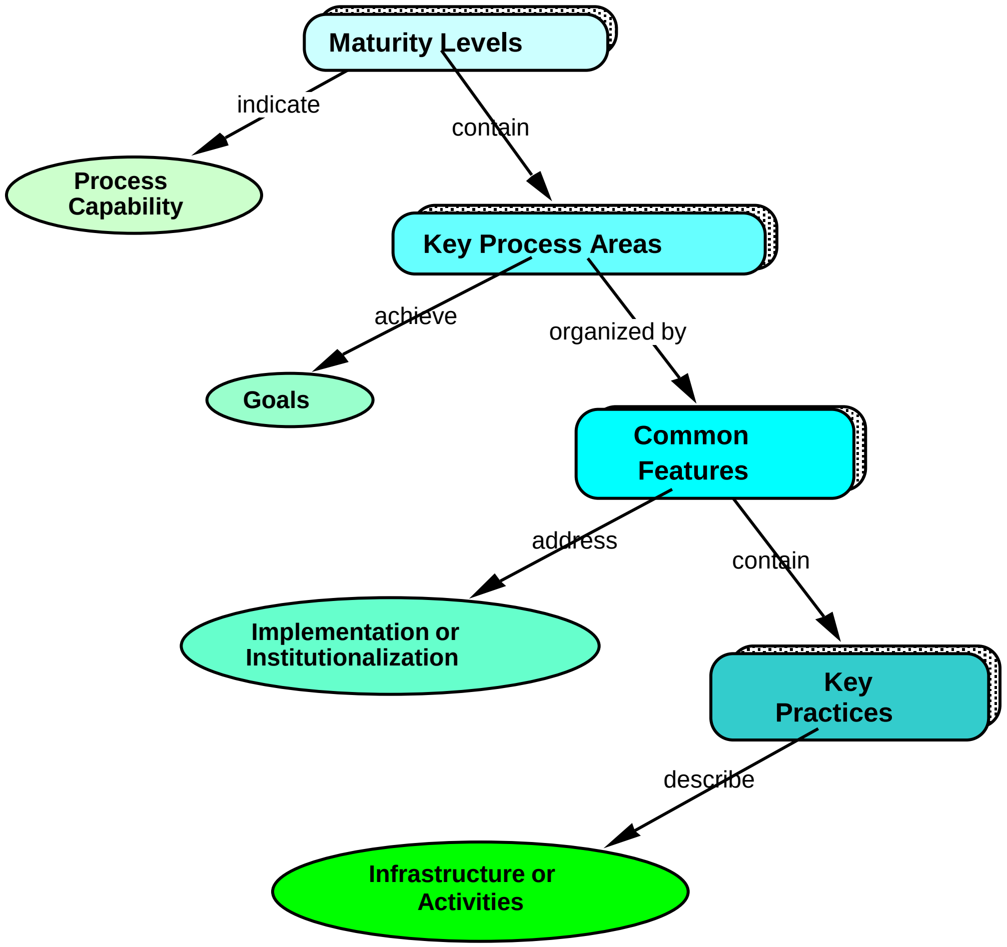
30.3%	**201 SW people or more**
23.8%	**101 to 200**
45.9%	**1 to 100**



Level 1 organizations typically do ***not*** have assessments performed

Source: SEI, March 2002 Update

Structure


Maturity Levels
indicate
Process Capability
contain
Key Process Areas
achieve
Goals
organized by
Common Features
address
Implementation or Institutionalization
contain
Key Practices
describe
Infrastructure or Activities

Key Process Areas (KPA)

- Each Level (except 1) consists of several *Key Process Areas*
  - Requirements Management (L2)
- “Key” implies other processes may be needed, but are not part of CMM, e.g. “requirements elicitation” is *not* a CMM KPA
  - Just manage ‘em after you find them :-)

Key Process Areas

Level	Focus	Key Process Area
5 Optimizing	Continuous process Improvement	Defect Prevention Technology Change Management Process Change Management
4 Managed	Product and process quality	Quantitative Process Management Quality Management
3 Defined	Defined engineering process	Organization Process Focus Organization Process Definition Integrated Software Management Product Engineering Training Program Intergroup Coordination Peer Review
2 Repeatable	Project management and commitment process	Requirements Management Project Planning Project Tracking Quality Assurance Configuration Management Subcontract Management
1 Initial	Heroes and massive effort	None

Goals

- Each KPA has associated *goals*
- The degree of achievement of each Level's KPA's Goals measure the level of maturity
  - Goals are achieved by satisfactorily implementing the activities

Level 2 KPA’s Goals (1)

- Requirements Management
  - Goal 1 - System requirements allocated to software are controlled to establish a baseline for software engineering and management use.
  - Goal 2 - Software plans, products, and activities are kept consistent with the system requirements allocated to software.
- Software Project Planning
  - Goal 1 - Software estimates are documented for use in planning and tracking the software project.
  - Goal 2 - Software project activities and commitments are planned and documented.
  - Goal 3 - Affected groups and individuals agree to their commitments related to the software project.

Level 2 KPA's Goals (2)

- Software Project Tracking and Oversight
  - Goal 1 - Actual results and performances are tracked against the software plans.
  - Goal 2 - Corrective actions are taken and managed to closure when actual results and performance deviate significantly from the software plans.
  - Goal 3 - Changes to software commitments are agreed to by the affected groups and individuals.

Level 2 KPA's Goals (3)

- Software Quality Assurance
  - Goal 1 - Software quality assurance activities are planned.
  - Goal 2 - Adherence of software products and activities to the applicable standards, procedures, and requirements is verified objectively.
  - Goal 3 - Affected groups and individuals are informed of software quality assurance activities and results.
  - Goal 4 - Noncompliance issues that cannot be resolved within the software project are addressed by senior management.

Level 2 KPA's Goals (4)

- Software Configuration Management
  - Goal 1 - Software configuration management activities are planned.
  - Goal 2 - Selected software work products are identified, controlled, and available.
  - Goal 3 - Changes to identified software work products are controlled.
  - Goal 4 - Affected groups and individuals are informed of the status and content of software baselines.

Level 2 KPA's Goals (5)

- Software Subcontract Management
  - Goal 1 - The prime contractor selects qualified software subcontractors. (CMM/SEI certified :-)
  - Goal 2 - The prime contractor and the software subcontractor agree to their commitments to each other.
  - Goal 3 - The prime contractor and the software subcontractor maintain ongoing communications.
  - Goal 4 - The prime contractor tracks the software subcontractor's actual results and performance against its commitments.

Common Features

- *Common Features* are attributes that indicate whether the implementation and *institutionalization* of a KPA is effective, repeatable, and lasting
- There are 5 features common to all KPAs
  - *Commitment to Perform*
  - *Ability to Perform*
  - *Activities to Perform*
  - *Measurement and Analysis*
  - *Verifying Implementation*

Common Feature #1 - Commitment to Perform

- *Commitment to Perform* describes actions the organization must take to ensure that the process is established and will endure.
- *This involves establishing organizational policies and senior management sponsorship.*
- Policy Statements
  - Refers to a *written* organizational policy for the practices of that key process area
  - May refer to the project or to the organization (level 3)
- Leadership
  - Assign a leadership *role* (e.g., project manager) or describes sponsorship activities necessary for the key process area to be successfully institutionalized

Common Feature #2 - Ability to Perform

- *Ability to Perform* describes preconditions that must exist in the project or organization to implement the software process competently
- Resources and *funding* for the activities covered by the key process area
- Training, formal and informal, for practioners
- *Orientation*
  - Indicates less depth of skill or knowledge than would be expected via training, e.g. overview or introduction to a topic for those overseeing or interfacing with the practitioners

Ability to Perform (2)

- Prerequisite Items
  - Either outputs from another KPA or from outside the project
- Software development plan is a prerequisite for *Project Tracking and Oversight* KPA
- CMM cites only those that are critical to implementing the KPA; there are many other issues that may be necessary for success

Common Feature #3 - Activities Performed

- *Activities Performed* describes the roles and procedures necessary to implement a key process area.
- This typically involves:
  - establishing plans and procedures
  - performing the work
  - tracking it
  - taking corrective actions as necessary.

Common Feature #4 - Measurement and Analysis

- *Measurement and Analysis* describes the need to measure the process and analyze the measurements. This typically includes examples of the measurements that could be taken to determine the status and effectiveness of the Activities Performed.

Common Feature #5 - Verifying Implementation (1)

- *Verifying Implementation* describes the steps to ensure that the activities are performed in compliance with the process that has been established. This typically encompasses reviews and audits by management and software quality assurance.
- Project management oversight on both a periodic and event-driven basis
  - Periodic - maintain an ongoing awareness of the status of the effort and be informed when significant events (e.g., design review) on the project occur

Verifying Implementation (2)

- Senior management oversight on a periodic basis
  - Provide awareness of and insight into software process activities
  - Adequate mechanisms for exception reporting
  - Covers different topics or similar topics at a higher level of abstraction than project management oversight reviews
- Software Quality Assurance activities
  - Particular activities considered appropriate for review and/or audit by the SQA group are described as a key practice

Activities

- Each KPA in each level has a list of *activities*

Level 2, Requirements Management Activities

- System requirements allocated to software are controlled to establish a baseline for software engineering and management use.
- Software plans, products, and activities are kept consistent with the system requirements allocated to software.
- The project follows a written, organizational policy for managing the system requirements allocated to software.

More Activities

- For each project, responsibility is established for analyzing the system requirements and allocating them to hardware, software, and other system components.
- The requirements are *documented*.
- Adequate resources and funding are provided for managing the requirements.
- Members of the software engineering group and other software-related groups are *trained* to perform their requirements management activities.

Still More

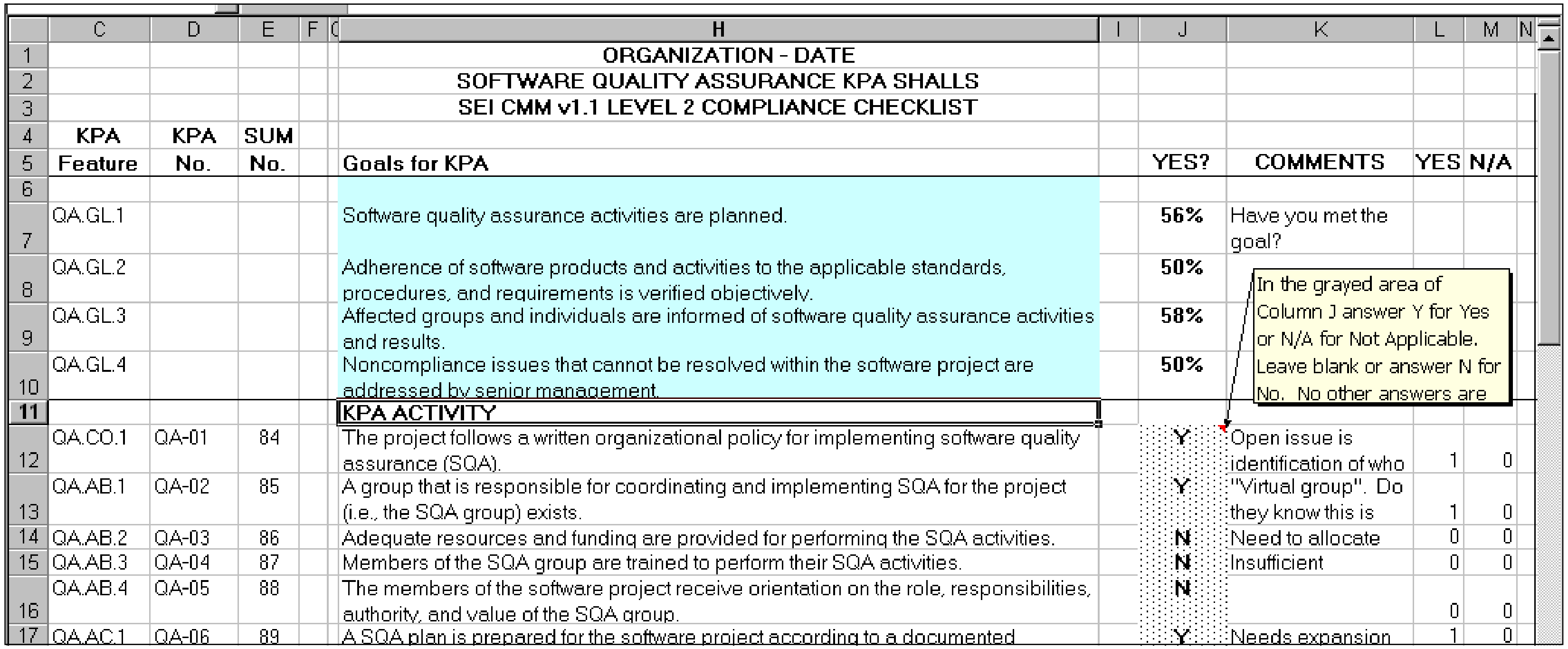
- The software engineering group reviews the requirements before they are incorporated into the software project.
- The software engineering group uses the requirements as the basis for software plans, work products, and activities.
- Changes to the requirements are reviewed and incorporated into the software project.
- Measurements are made and used to determine the status of the activities for managing the requirements.
- The activities for managing the requirements are reviewed with senior management on a periodic basis.

Finally

- The activities for managing the requirements are reviewed with the project manager on both a periodic and event-driven basis.
- The software quality assurance group reviews and/or audits the activities and work products for managing the allocated requirements and reports the results.

An Assessment Spreadsheet

- Simple “Y” for yes, “N” or blank for no, “N/A” for not applicable answers
- Percent of Goal achievement is calculated to determine level of achievement of KPA



	C	D	E	H	J	K	L	M
1				ORGANIZATION - DATE				
2				SOFTWARE QUALITY ASSURANCE KPA SHALLS				
3				SEI CMM v1.1 LEVEL 2 COMPLIANCE CHECKLIST				
4	KPA	KPA	SUM					
5	Feature	No.	No.	Goals for KPA	YES?	COMMENTS	YES	N/A
6								
7	QA.GL.1			Software quality assurance activities are planned.	56%	Have you met the goal?		
8	QA.GL.2			Adherence of software products and activities to the applicable standards, procedures, and requirements is verified objectively.	50%			
9	QA.GL.3			Affected groups and individuals are informed of software quality assurance activities and results.	58%			
10	QA.GL.4			Noncompliance issues that cannot be resolved within the software project are addressed by senior management.	50%			
11				KPA ACTIVITY				
12	QA.CO.1	QA-01	84	The project follows a written organizational policy for implementing software quality assurance (SQA).	Y	Open issue is identification of who	1	0
13	QA.AB.1	QA-02	85	A group that is responsible for coordinating and implementing SQA for the project (i.e., the SQA group) exists.	Y	"Virtual group". Do they know this is	1	0
14	QA.AB.2	QA-03	86	Adequate resources and funding are provided for performing the SQA activities.	N	Need to allocate	0	0
15	QA.AB.3	QA-04	87	Members of the SQA group are trained to perform their SQA activities.	N	Insufficient	0	0
16	QA.AB.4	QA-05	88	The members of the software project receive orientation on the role, responsibilities, authority, and value of the SQA group.	N		0	0
17	QA.AC.1	QA-06	89	A SQA plan is prepared for the software project according to a documented	Y	Needs expansion	1	0

In the grayed area of Column J answer Y for Yes or N/A for Not Applicable. Leave blank or answer N for No. No other answers are

Caveats and Clarifications

- Maturity assessment is not a guarantee of quality
- Activities belonging to a higher level's KPA are often needed and implemented by a lower level organization
  - Peer Review is a Level 3 KPA, but is a highly effective technique often used by organizations that would be rated Level 2
- Achievement of maturity takes time
  - Estimate of 2 years to fully transition from Level 1 to Level 2
  - Benefits accrue long before next level is reached
- Not a silver bullet

Summary

- SEI/CMM provides a well defined framework for assessing and evaluating a software development organization’s maturity
- Formal assessment is not required
- SEI/CMM provides a well defined framework for understanding what a mature software development organization should look like
- SEI/CMM is descriptive, not prescriptive
- SEI/CMM is comprehensive, but not complete

Questions?