

Process Improvement Driven by Maturity Frameworks

Books: N. Fenton, S. Pfleeger, *Software Metrics: A Rigorous and Practical Approach*, PWS, 1996
 Other source: B. Kitchenham, *Measurement for Process Improvement*, 1995, out of print

Process Improvement Frameworks

- Some development processes are more mature than others, as noted by the US Software Engineering Institute's (SEI's). While some organizations have clearly-defined processes, others are more variable, changing significantly with the people who work on the projects.
- We look at the most popular process assessment framework (CMM). Because this is being mandated by corporations or governments, it is important to understand it and to be aware of measurement's role within them
- There are other less popular frameworks, e.g., Bootstrap, SPICE (ISO standard)
- Such frameworks are complementary to measurement-driven improvement: They guide earlier improvement steps and provide a context for measurement.

Capability Maturity Model

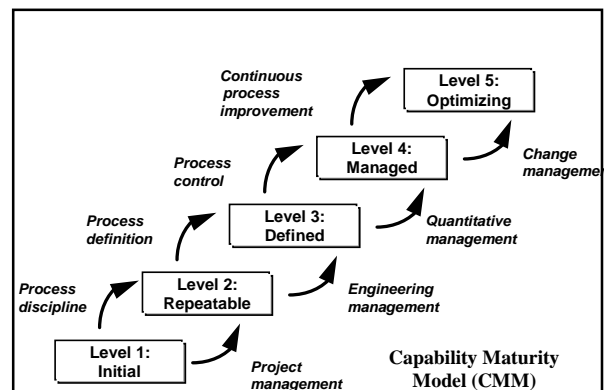
- The *Capability Maturity Model* (CMM) was developed by the U.S. Software Engineering Institute (SEI) to assist the Department of Defense in assessing the quality of its contractors.
- The CMM describes principles and practices that are assumed to lead to better software products, and the model organizes them into five levels, providing a path to more process visibility and control, and to the improved products that result.
- The model is used in two ways: by potential customers, to identify the strengths and weaknesses of their suppliers, and by software developers themselves, to assess their capabilities and set a path toward improvement.

Some Definitions

- **Software Process:** a set of interrelated activities, methods, practices, and transformations that people use to develop and maintain software.
- **Software Process Capability:** It describe a range of expected results that can be achieved by following a software process.
- **Software Process Maturity:** The extent to which a specific process is explicitly defined, managed, measured, controlled, and effective. It implies a potential growth in capability.

Capability Maturity Model

- Each of the five capability levels is associated with a set of *key process areas* on which an organization should focus as part of its improvement activities.
- The first level of the maturity model, *initial*, describes a software development process that is *ad hoc* or even chaotic. Few processes are defined, and the success of development depends on individual efforts, not on team accomplishments.



Maturity and Capability

- There are a number of assumptions in the CMM
 - An organization's software process maturity helps to predict a project's ability to meet its objectives
 - As an organization is going up the maturity ladder, its effectiveness increases and its variability across projects decreases (more predictability)
- Some published case studies indicate there are significant improvements in both quality and productivity
- The return on investment seems to typically be in the 4:1 to 8:1 range for *successful* process improvement efforts.
- Increases in productivity range within 9-67% and decreases in cycle time range within 15-23%.

CMM Key Process Areas

CMM level	Key process areas
Initial	none
Repeatable	Requirements management Software project planning Software project tracking and oversight Software subcontract management Software quality assurance Software configuration management
Defined	Organization process focus Organization process definition Training program Integrated software management Software product engineering Intergroup coordination Peer reviews
Managed	Quantitative process management Software quality management
Optimizing	Defect prevention Technology change management Process change management

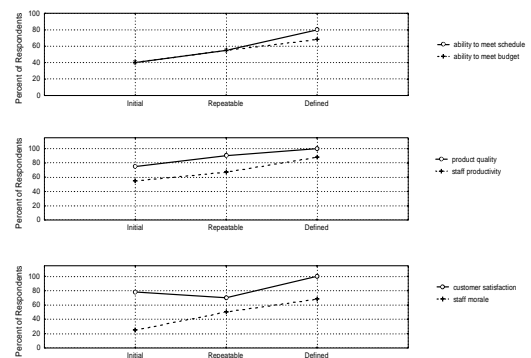
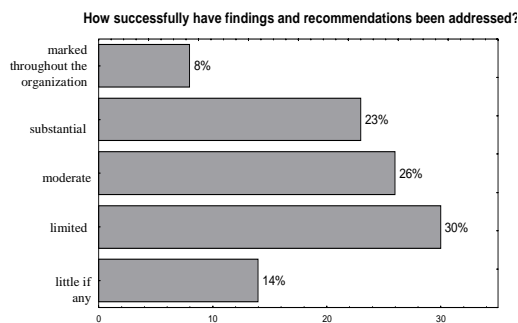
Key Process Area Examples

- **Requirements Management:** Establish a common understanding between the customer and the software project of the customer's requirements that will be addressed by the software project. This agreement is the basis for planning and managing the software project.
- **Software Project Planning:** Establish reasonable plans for performing the software engineering and for managing the software project. These plans are the necessary foundations for managing software projects.
- **Software Quality Assurance:** Provide management with appropriate visibility into the process being used by the software project and of the product being built.

Empirical Studies of the CMM

- Is there any evidence that higher maturity leads to greater project and organizational effectiveness?
- Herbsleb et al tracked the experience of a large number of organizations. A survey was run on 138 persons from 56 organizations in the US and Canada
- The CMM assessment took place approximately one to three years before the survey
- Organizations that vary in size and sector within the software industry, and that vary in the success they have experienced in their process improvement efforts
- Different people, with different roles were surveyed not to introduce any bias

Results



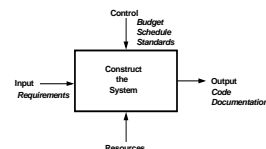
Maturity and Measurement

- The Capability Maturity Model emphasizes quantitative control of the process, and the higher levels direct the organization to use measurement and quantitative analysis.
- A measurement plan using the CMM framework can derive many of its goals directly from the key process areas and practices.
- At each maturity level, measurement and visibility are closely related: a developer can measure only what is visible in the process, and measurement helps to enable and increase visibility.
- Thus, the five-level maturity scale such as the one employed by the SEI is a convenient context for determining what to measure first and how to plan an increasingly comprehensive measurement program.

Level 1: Adhoc

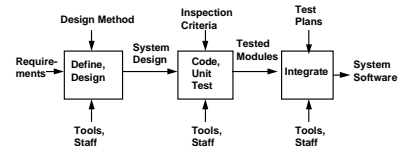
- For this level of process maturity, it is difficult even to write down or depict the overall process; the process is so reactive and ill-defined that visibility is nil and comprehensive measurement difficult.
- You may have goals relating to improved quality and productivity, but you do not know what current levels of quality and productivity are.
- For this level of process maturity, baseline measurements are needed to provide a starting point for measuring improvement as maturity increases.
- If your organization is at level 1, you should concentrate on imposing more structure and control on the process, in part to enable more meaningful measurement.

Level 2: Repeatable



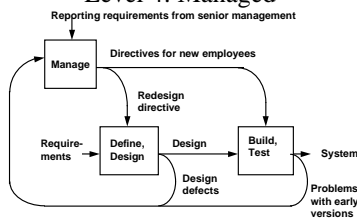
We can associate measurements with each arrow in the process diagram. Thus, for a repeatable process, measures of the input might include the size and volatility of the requirements. The output may be measured in terms of system size (functional or physical), the resources as overall staff effort, and the constraints as cost and schedule in dollars and days, respectively.

Level 3: Defined



Defects discovered in each type of product can be tracked, and you can compare the defect density of each product with planned or expected values. In particular, early product measures can be useful indicators of later product measures. For example, the quality of the requirements or design can be measured and used to predict the quality of the code.

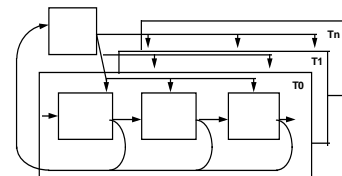
Level 4: Managed



Management oversight relies on a metrics database that can provide information about such characteristics as distribution of defects, productivity and effectiveness of tasks, allocation of resources, and the likelihood that planned and actual values will match.

Level 5: Optimizing

- Measures from activities are used to improve the process, possibly by removing and adding process activities and changing the process structure dynamically in response to measurement feedback.
- Results from one or more ongoing or completed projects may also lead to a refined, different development process for future projects.
- Measurements act as sensors and monitors, and the process is not only under your control but you can change it significantly in response to warning signs.



An optimizing process (level 5)

CMM and Measurement

Maturity Level	Characteristics	Type of Metrics to Use
5. Optimizing	Improvement fed back to the process	Process plus feedback for changing the process
4. Managed	Measured process	Process plus feedback for control
3. Defined	Process defined and institutionalized	Product
2. Repeatable	Process dependent on individuals	Project management
1. Initial	Ad hoc	Baseline

Fenton and Pfleeger et al, 1996

What does Process Maturity Measure?

- The real indicator of process maturity is the level of predictability of project performance (quality, cost, schedule).
 - Level 1: Random, unpredictable performance
 - Level 2: Repeatable performance from project to project
 - Level 3: Better performance on each successive project
 - Level 4: project performance improves on each subsequent project either
 - Substantially (order of magnitude) in one dimension of project performance
 - Significant in each dimension of project performance
 - Level 5: Substantial improvements across all dimensions of project performance.

Determining the Maturity of a Project

- Level 1 questions:
 - Has a process model been adopted for the Project?
- Level 2 questions:
 - Software size: How big is the system?
 - Personnel effort: How many person-months have been invested?
 - Technical expertise of the personnel:
 - What is the application domain experience
 - What is their design experience
 - Do they use tools?
 - Do they have experience with a design method?
 - What is the employee turnover?

Maturity Level 3 Questions

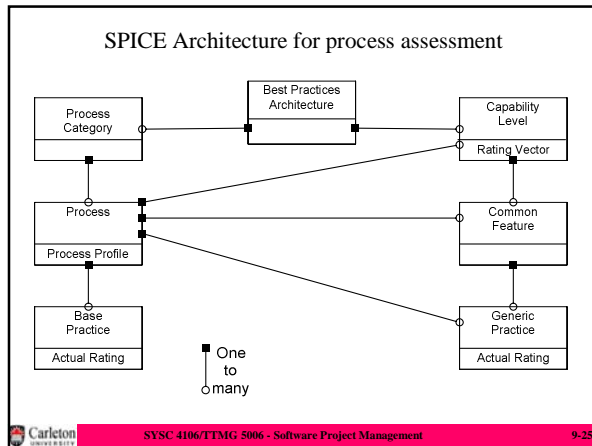
- What are the software development activities?
- Requirements complexity:
 - How many requirements are in the requirements specification?
- Design complexity:
 - Does the project use a software architecture? How many subsystems are defined? Are the subsystems tightly coupled?
- Code complexity: How many classes are identified?
- Test complexity:
 - How many unit tests, subsystem tests need to be done?
- Documentation complexity: Number of documents & pages
- Quality of testing:
 - Can defects be discovered during analysis, design, implementation? How is it determined that testing is complete?
 - What was the failure density? (Failures discovered per unit size)

Maturity Level 4 and 5 Questions

- Level 4 questions:
 - Has intra-project feedback been used?
 - Is inter-project feedback used? Does the project have a post-mortem phase?
 - How much code has been reused?
 - Was the configuration management scheme followed?
 - Were defect identification metrics used?
 - Module completion rate: How many modules were completed in time?
 - How many iterations were done per activity?
- Level 5 questions:
 - Did we use measures obtained during development to influence our design or implementation activities?

SPICE – Software Process Improvement and Capability dEtermination

- SPICE is intended to harmonize and extend the existing approaches.
- It is similar to CMM but, recommended for both process improvement and capability determination.
- The framework is built on an assessment architecture that defines desirable practices and process
- There are two different types of practices:
 - Base practices* are essential activities of a specific process
 - Generic practices* institutionalize or implement a process in a general way.
- The figure below shows how SPICE architecture ties the tow together.
- CMM addresses organizations, but SPICE addresses processes.



SPICE architecture for process assessment (Rout 1995)

- The left-hand side of the diagram represents functional practices involved in software development. This functional view considers five activities:
 - Customer-supplied*: Processes that affect the customer directly, support development and delivery of the products to the customer, and ensure correct operation and use
 - Engineering*: processes that specify, implement, or maintain the system and its documentation.
 - Project*: processes that establish the project, coordinate or manage resources, or provide customer services.
 - Support*: processes that enable or support performance of the other processes.
 - Organization*: processes that establish business goals and develop assets to achieve those goals.

Carleton University SYSC 4106/TTMG 5006 - Software Project Management 9-26

- The right-hand side of the figure shows a picture of management; the generic practices, applicable to all processes, are arranged in six levels of capability:
 - Not performed*: failure to perform and no identifiable workproducts.
 - Performed informally*: not planned and tracked, depends on individual knowledge and identifiable workproducts.
 - Planned and tracked*: verified according to specified procedures, workproducts conform to specified standards and requirement.
 - Well-defined*: process using approved, tailored versions of standard, documented processes.
 - Quantitatively controlled*: detailed performance measures, prediction capability, objective management, and workproducts evaluated quantitatively.
 - Continuously improving*: quantitative targets for effectiveness and efficiency based on business goals, quantitative feedback from defined processes, plus trying new ideas.

Carleton University SYSC 4106/TTMG 5006 - Software Project Management 9-27

Steps to Take in Using Metrics

Metrics are useful only when implemented in a careful sequence of process-related activities.

- Assess your current process maturity level
- Determine what metrics to collect
- Recommend metrics, tools and techniques
 - whenever possible implement automated support for metrics collection
- Estimate project cost and schedule and monitor actual cost and schedule during development
- Construct a project data base:
 - Design, develop and populate a project data base of metrics data.
 - Use this database for the analysis of past projects and for prediction of future projects.
- Evaluate cost and schedule for accuracy after the project is complete.
- Evaluate productivity and quality
 - Make an overall assessment of project productivity and product quality based on the metrics available.

Carleton University SYSC 4106/TTMG 5006 - Software Project Management 9-28

Pros and Cons of Process Maturity

- Problems:**
 - Need to watch a lot ("Big brother", "big sister")
 - Overhead to capture, store and analyse the required information
- Benefits:**
 - Increased control of projects
 - Predictability of project cost and schedule
 - Objective evaluations of changes in techniques, tools and methodologies
 - Predictability of the effect of a change on project cost or schedule

Carleton University SYSC 4106/TTMG 5006 - Software Project Management 9-29

Summary I

- It is important to note that process maturity does not involve a discrete set of five possible ratings. Instead, maturity represents relative locations on a continuum from 1 to 5.
- An individual process is assessed or evaluated along many dimensions, and some parts of the process can be more mature or visible than others. For example, a repeatable process may not have well-defined intermediate activities, but the design activity may indeed be clearly defined and managed.

Carleton University SYSC 4106/TTMG 5006 - Software Project Management 9-30

Summary II

- The guidelines presented here are meant only to give a general depiction of typical processes. It is essential that you begin any measurement program by examining the process model and determining what is visible.
- If one part of a process is more mature than the rest, measurement can enhance the visibility of that part and help to meet overall project goals, at the same time bringing the rest of the process up to a higher level of maturity. Thus, in a repeatable process with a well-defined design activity, design quality metrics may be appropriate and desirable.