PROCESS IMPROVEMENT

Primary goal – Build better products

Secondary goal – We don't know how to build better products, so develop good techniques that result in good products. We still are not very good at that, so …

Current goal – Determine HOW people work, so when something better shows up, you can take advantage of that. This is commonly called Process Improvement, and Continuous Process Improvement has been the current fad in software engineering research. "We don't know what to do, but when we see something better we can recognize it."

CMM

Most popular method today – Capability Maturity Model. Major developer Watts Humphrey

Began in 1987 at Software Engineering Institute (SEI). Two projects:

- Software Capability Evaluation (SCE) – a self assessment of an organization
- Capability Maturity Model (CMM) – A mechanism to evaluate the "maturity" of a development group

They were soon merged into one process.

CMM Overview

An evaluation is based upon a 5-level rating. Each level represents a more mature organization:

- Initial – No predefined process. Development is ad hoc. No established rules. (Note: This does not mean bad practices, but usually means it can be greatly improved. There are good development practices followed by some level 1 organizations.)
- Repeatable – Policies are in place to manage software development. Each project follows some process that is generally similar to others.
- Defined – Processes for software development fit into a corporate standard for software development. Development processes are well documented.
- Managed – Quantitative goals for products and processes are set. Software process attributes are quantifiable.
- Optimizing – Goals for continuous process improvement are in place.

CMM Key Process Areas (KPAs)

CMM evaluations are based upon KPAs. 18 KPAs govern the CMM. In order to be at level N, all the KPAs up to that level must be satisfied.

Initial – None – This is the initial state.

Repeatable – Requirements management; Project planning; Project tracking; Subcontract management; 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 – Software quality management; Quantitative process management [Is it really level 4?]

Optimizing – Defect prevention; Technology change management; Process change management

FOR EACH KPA …

For each KPA, need to address commitment:

- What is commitment to perform this KPA? What policies and management leadership are behind this KPA?
- What is the ability to perform this KPA? What mechanisms are in place?
- What activities are performed? What are the procedures in place to perform that KPA?
- What is the measurement of the KPA and the analysis of its adherence?
- What verifies the implementation? How to ensure compliance?

KPA ISSUES

Each KPA provides needed features for good development, but there are issues with CMM:

- Companies often look only one level ahead and ignore ultimate goal of level 5.
- Management only. Role of technology mostly ignored
- Expensive, especially for small companies
- Process is “bottom up.” “One size fits all.” (Contrast with QIP next.)
- “Level 3” is often a “requirement” for a DoD contract even though not fully demonstrated higher level is an accurate predictor of better quality products.

CMMs

CMM have become a growth industry …

- SW-CMM - Capability Maturity Model for Software
- P-CMM - People Capability Maturity Model
- SA-CMM - Software Acquisition Capability Maturity Model
- SE-CMM - Systems Engineering Capability Maturity Model
- IPD-CMM - Integrated Product Development Capability Maturity Model

CMMI proposed to integrate various models

CMMI

The CMM Integration project was formed to address the problem of having to use multiple Capability Maturity Models. The initial mission of the project was to combine three source models:

- Capability Maturity Model for Software (SW-CMM) v2.0 draft C
- Electronic Industries Alliance Interim Standard (EIA/IS) 731, Systems Engineering Capability Model (SECM)
- Integrated Product Development Capability Maturity Model (IPD-CMM) v0.98

into a single model for use by organizations pursuing enterprise-wide process improvement.

CMMI OVERVIEW

Six levels (0-5):

- Incomplete – Does not implement all of the capability level1practices
- Performed – Can perform all of the level 1 capability
- Managed
- Defined
- Quantitatively managed
- Optimizing

COMPONENTS OF MODEL

- Required – Required of an organization
- Expected – Organizations typically implement these components
- Informative – Help model users understand the goals of the model
- Process area – a collection of practices that collectively help achieve an objective
- Capability levels – Focus on maturing an organization's ability to achieve certain process goals

Still too new to assess impact of CMMI on CMM and software development practices in general

ISO 9000 Certification

ISO standard 9001 (software component) for standard quality of a product. (Note that "good" quality is not required, only that quality can be accurately determined. Concrete life preservers can be "ISO certified," but not be very useful.)

- Companies want ISO certification as a requirement to sell in Europe.
- Based upon 20 clauses. "ISO certification" is comparable to CMM level 2 or level 3, but processes but different.

ISO 9000 CLAUSES

Management	**Design control**
Quality team	**Purchasing**
Contract review	**Process control**
Training	**Service**
Control of customer supplied product	**Product identification and traceability**
Document and data control	**Inspection and test cases**
Control of nonconforming products	**Control of quality record**
Integration and testing	**Control of inspection, measuring, test**
Corrective and preventative actions	**Handling, storage, packaging, delivery**
Interval quality audits	**Statistical techniques**

ISO 9000 TASKS

- writing a quality manual, describing the organization's quality system at a high level
- writing procedure documents describing the work is carried out in the organization
- creating a system to control distribution and re-issue of documents
- identifying training needs for most positions in the organization
- training people in the organization on operation of the quality system
- planning and conducting internal audits

ISO 9000 STANDARDS

ISO 9000, "Quality management and quality assurance standards - Guidelines for selection and use," clarifies the distinctions and interrelationships between quality concepts and provides guidelines for the selection and use of a series of international standards on quality systems that can be used for internal quality management purposes (ISO 9004) and for external quality assurance purposes (ISO 9001, 9002 and 9003).

ISO 9001, "Quality systems - Model for quality assurance in design/development, production, installation, and servicing." Of the ISO 9000 series, it is the standard that is most pertinent to software development and maintenance.

ISO 9002, "Quality systems - Model for quality assurance in production and installation."

ISO 9003, "Quality systems - Model for quality assurance in final inspection and test"

ISO 9004, "Quality management and quality system elements – Guidelines."

ISO 9000-3, provides "Guidelines for the application of ISO 9001 to the development, supply, and maintenance of software."

GOALS OF ISO 9000

- An organization should achieve and sustain the quality of the product or service produced so as to meet continually the purchaser's stated or implied needs.
- An organization should provide confidence to its own management that the intended quality is being achieved and sustained.
- An organization should provide confidence to the purchaser that the intended quality is being, or will be, achieved in the delivered product or service provided.

Plan-Do-Check-Act Cycle (PDCA)

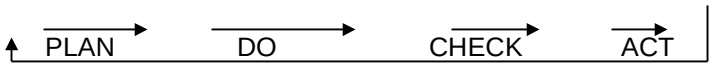
Based upon work by W. A. Shewart and made popular by W. E. Deming

Goal: optimize and improve a single process model/production line

Approach: uses such techniques as

- feedback loops
- statistical quality control
- design of experiments
- data models based upon multiple replications

Result: predictive models of the relationship between process and product



Note: that any application of the process produces a large quantity of products, sufficient to generate an accurate statistical model

DEMING'S 14 POINTS

Deming's approach toward improved management.

Are they used in the software industry?

- Create constancy of purpose toward improvement of product and service
- Adopt a new philosophy – Can no longer live with delays, mistakes, defective workmanship
- Cease dependence on mass inspection. Use statistical evidence of quality
- End practice of awarding business on price alone. Depend upon good measures of quality
- Improve constantly production and service

DEMING'S 14 POINTS - 2

- Institute a vigorous program of education and training
- Adopt and institute leadership
- Drive out fear so everyone works effectively
- Break down barriers between departments
- Eliminate numerical goals, slogans, posters without providing new methods
- Eliminate work standards that prescribe numerical quotas
- Remove barriers that stand between the hourly worker and his pride of workmanship
- Encourage education and self-improvement for everyone
- Create a structure in top management that will work daily on the above 13 points.

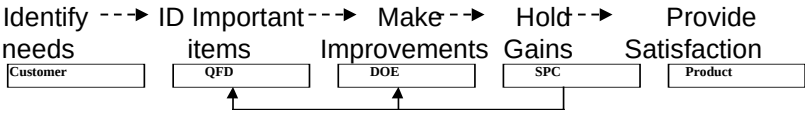
Total Quality Management

Term coined by Navy in 1985. Based upon work by Feigenbaum, Taguchi, ...

Goal: generate institutional commitment to success through customer satisfaction

Approach: varied, a philosophy supported by a variety of techniques, e.g.,

- Quality Function Deployment (QFD)
- design of experiments (DOE)
- statistical process control (SPC)



Result: An customer driven organization and a satisfied set of customers

Lean Enterprise Management (LEM)

Philosophy used to improve factory output. Book by Womack, et. al. (1989), on the application of lean enterprises in the automotive industry

Goal: to build products using the minimal set of activities needed, eliminating non essential steps, i.e., tailoring the process to the product needs

Approach: uses such concepts as

- technology management
- human centered management
- decentral organization
- quality management
- supplier and customer integration
- internationalization/regionalization

Result: A set of processes individualized for each particular product line

Quality Improvement Paradigm

Developed by Basili as part of NASA/GSFC Software Engineering Laboratory (SEL) research – General model emphasized this semester

QIP – 6 step process:

Characterize – Analyze and understand the environment. Need to understand current processes before improvement. Compare to CMM. Measurement crucial at beginning (not level 4) and each environment is unique.

Set goals – Quantifiable goals must be based on discovered characteristics.

Choose process – Based on characterization and goals, processes, methods, and tools have to be determined.

Execute the process – Perform the processes constructing the required products.

Analyze – At the end of each project, analyze the data gathered, determine problems, and make recommentations for improvement.

Package – Structure knowledge and store it an experience base for future reuse.

Forms of Packaged Experience

Equations defining the relationship between variables,
e.g. Effort = $1.48*KSLOC^{.98}$,
Number of Runs = 108 + 150*KSLOC

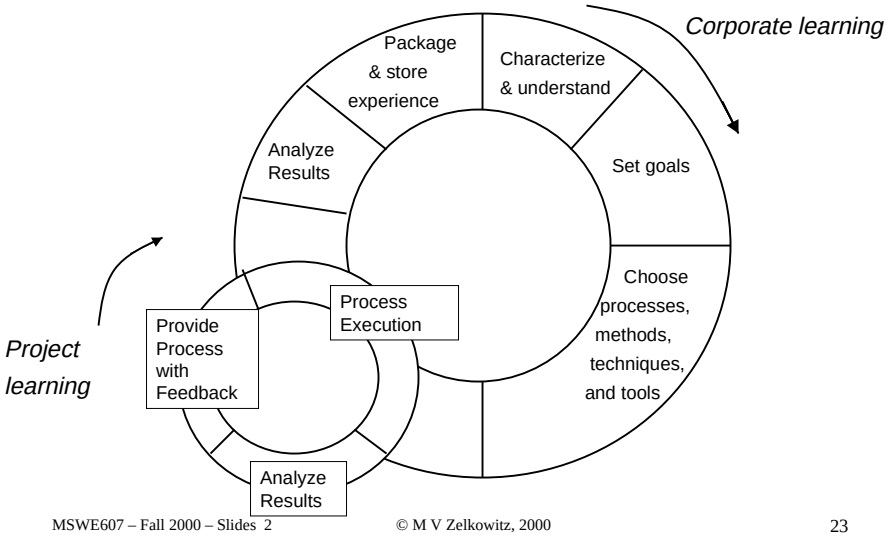
Histograms or pie charts of raw or analyzed data, e.g., Classes of Faults: 30% data, 24% interface, 16% control, 15% initialization, 15% computation
Effort Distribution: 23% design, 21% code, 30%test, 26% other

Graphs defining ranges of "normal" e.g.,
Fault Slippage Rate: halve faults after each test phase
(4, 2, 1, .5)

Specific lessons learned, e.g., an Ada design should use library units rather than a deeply nested structure;
minimize the use of tasking as its payoff is minimal in this environment;
size varies inversely with defect rate up to about 1KLOC per module

Processes descriptions (adapted to SEL), e.g., Recommended Approach, Manager's Handbook, Cleanroom Process Handbook, Ada Developer's Guide, Ada Efficiency Guide

Quality Improvement Paradigm - 2



Experience Factory

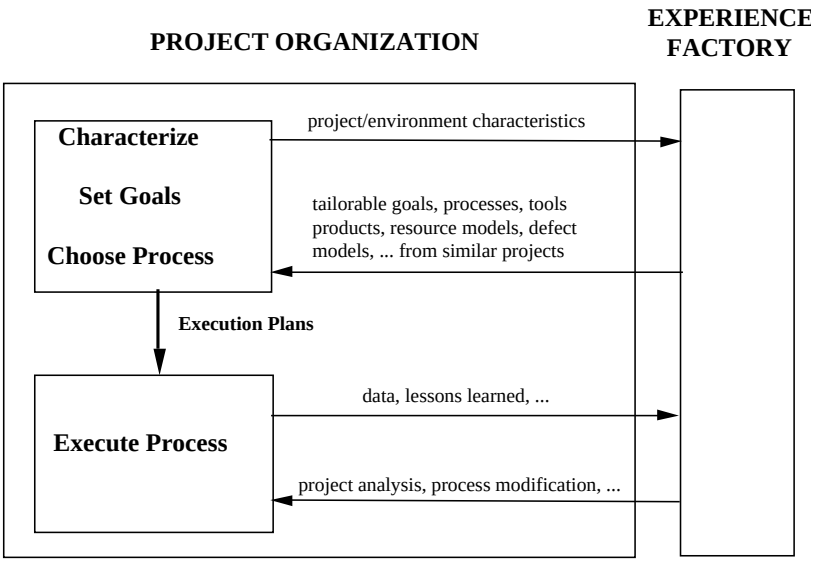
Separate Project Organization (that builds products) and Experience Factory that contains the corporate knowledge – models, data, experiences, source libraries, …

Project Organization - focus/priority is delivery, supported by packaged experiences

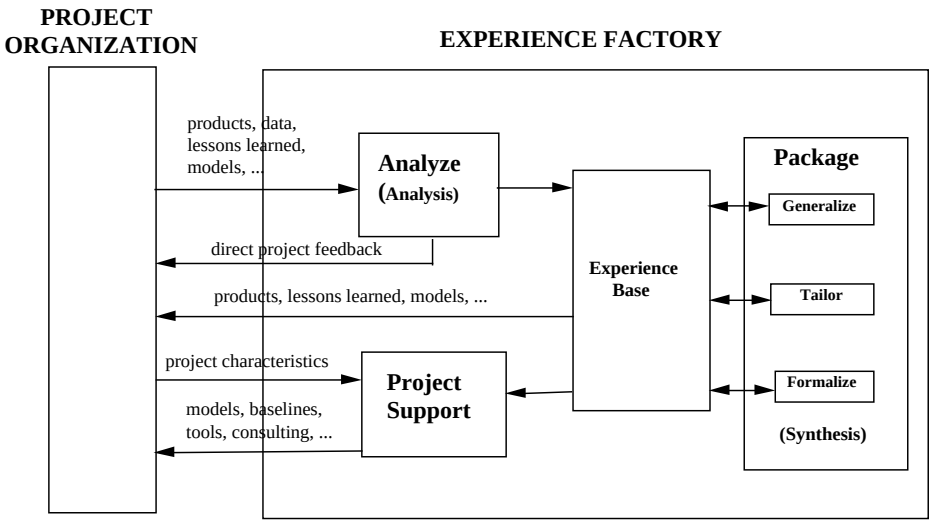
Experience Factory (EF) - focus is project development support analyzes and synthesizes all kinds of experience acts as a repository for such experience supplies that experience to various projects on demand.

The Experience Factory packages experience by building informal, formal or schematized, and productized models and measures of various software processes, products, and other forms of knowledge via people, documents, and automated support

Experience Factory Organization- Role of the Project Organization



Experience Factory Organization- Role of the Experience Factory



The Experience Factory Organization

Some Important Characteristics:

Everyone is part of the technology infusion process,
can be a developer on one project and an experimenter on another

Project personnel play the major role in the feedback mechanism if they are not using the technology right it can be because:

- they don't understand it / it wasn't taught right
- it doesn't fit/interface with other project activities
- it needs to be tailored
- it doesn't work
- you need the user to tell you how to change it

Technology infusion is motivated by the local problems, so people are more willing to try something new

The Experience Factory Organization - 2

Iterating with the QIP:

Get the commitment- Put the organization in place, collect data to establish baselines e.g., defects and resources that are process and product independent

Measure your strengths and weaknesses- Provides a focus and goals for improvement

Select and experiment with methods and techniques to improve process based upon product quality needs

Evaluate improvement based upon existing resource and defect baselines

Understand process characteristics and product qualities relationship
Manipulate process to achieve those product characteristics
Define and tailor better and measurable processes based upon

- experience and knowledge of the environment and
- process conformance and domain understanding

Establish new baselines- Repeat the process and find the next opportunity for improvement

CMM vs. EF

Similarities
- characterize processes

Differences

CMM
- goal is to improve process
- characterize processes
- baseline is process assessment
- common yardstick drives change (key process areas)
- change based upon assessment of processes
- measurement plays key role at level 4
- process emphasis is on management activities

EF
- goal is to improve product
- characterizes all kinds of experiences: products, defects, resources
- baseline is process and product understanding
- many goals drive change, e.g., customer satisfaction
- change based upon achieving goals
- measurement fundamental at all stages
- process emphasis is on technological and management activities

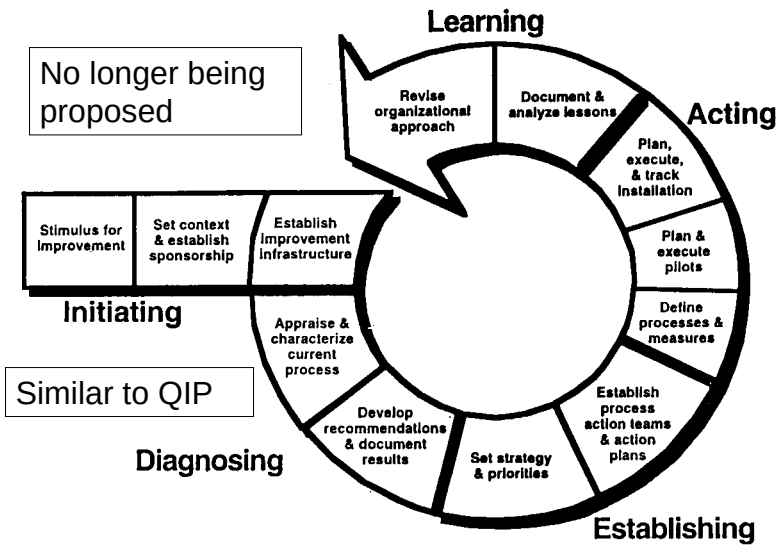
SEI Process Improvement Cycle

Initialize
- Establish Sponsorship
- Create vision and strategy
- Establish improvement structure

For Each Maturity Level
- Characterize current practice in terms of key process areas
- Assessment recommendations
- Revise strategy (generate action plans and prioritize key process areas)

For Each key Process Area
- Establish process action teams
- Implement tactical plan, define processes, plan and execute pilot(s), plan and execute institutionalization
- Document and analyze lessons
- Revise organizational approach

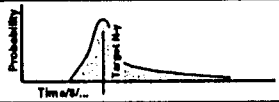
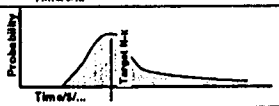
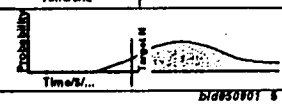
SEI's Ideal Improvement Model



No longer being proposed

Similar to QIP

Evolution of Process Capability

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

Experience Factory Organization

Can it make you a 5?

Using the Experience Factory Organization:

- You pull yourself up from the top rather than pushing up from the bottom
- At step 1 you start with a level 5 organization but not level 5 capabilities
- You are driven by an understanding of your business, your product and process problems, your business goals, your experience with methods, etc.
- You learn from your business, not on an external model of process
- You make process improvements based upon an understanding of the relationship between process and product in your organization

Can it make you a 5?

What does a level 5 organization mean?

It is an organization that can manipulate process to achieve various product characteristics.

This requires that we have a process and an organizational structure to help us:

- Understand our processes and products
- Measure and model the project and the organization
- Define and tailor process and product qualities explicitly
- Understand the relationship between process and product qualities
- Feedback information for project control
- Experiment with methods and techniques
- Evaluate our successes and failures
- Learn from our experiences
- Package successful experiences
- Reuse successful experiences

Can it make you a 5?

Using EF may not get you a level 5 rating (depending on how it gets defined when you get there)

- because your technologies are not from the “key set of processes”
- but you are operating at a level 5 definition
- and have chosen and tailored processes to create a lean, optimizing, continuously improving organization

How does this fit in with the CMM?

- EF is not incompatible with the SEI CMM model
- can use key process assessments to evaluate where you stand (along with your internal goals, needs, etc.).
- Using the EF will move you up the maturity scale faster

  offers experience early on with an improvement-based organization
- can demonstrate product improvement benefits early

TQM vs. EF

Similarities

- cover goals that are customer satisfaction driven
- based upon the philosophy that quality is everyone's job
- everyone is part of the technology infusion process
- can be on project team on one project, experimenting team on another
- all the project personnel play the major role in the feedback mechanism

Differences

EF provides

specific steps, model types

more specific and aimed at software

LEM vs. EF

Similarities

- scientific method /PDCA philosophy
- feedback loops, learn from experiments, process /product relationship
- goal is to generate an optimum set of processes
- based upon tailoring a set of processes for particular product

Differences

- LEM based upon production model building based upon continual repetition of the same process
- can use accurate models for statistical quality control
- EF based upon development,
- must learn from one process about another
- models are less rigorous and more abstract
- processes more human based
- effects building, use, and accuracy of models built

Personal Software Process (PSP)

Developed by Watts Humphrey at SEI. Designed as aid for programmer activities in conjunction with corporate-wide CMM activities. Developed 1994.

Develop methods to improve: size estimates, effort estimates, software quality (defects)

Issue: Burden of data required to manage a project

PHASES OF PSP

- Gather data – Measures concerned about: size and time estimating errors, cost-performance index (planned vs. actual), defects injected and removed per hour, process yield (% defects fixed before testing), quality appraisal (% of time in design and code reviews), and appraisal to failure ratio.
- Estimating – PROBE is a proxy-based estimating method based on personal data from previous efforts. (Uses LOC, but can use function points, screens, chapters, etc.) Making repeated estimates on historical data improves accuracy.

PSP PHASES - continued

- Managing defects – All defects are counted – compiling, testing, desk checking. Defect management is a personal software engineer's responsibility to find and fix. (From early versions of PSP course, dropped from 116/KLOC to 49/KLOC, but test dropped from 34 to 10 and compile from 76 to 13)
- Managing yield – When start PSP program, engineering spend 30% time compiling and testing and 30%-50% during integration and testing phases. Improve process by thorough code reviews before compiling. (We will discuss code reviews and cleanroom later.)

PSP PHASES - continued

Controlling code quality – Compute appraisal costs (% time in design and code reviews), failure costs (% time in compile and test) and prevention costs (% time preventing defects before they occur – prototyping, formal specs, …)

- Appraisal to failure ratio – appraisal costs/failure costs. – A goal of PSP is to achieve a ration of 2
- Until quality goals are met.
- Humphrey's data shows that higher defect rates are not related to LOC/hour

PSP DEFECT TYPES

Documentation – comments, messages

Syntax – spelling, typos, punctuation

Build, package – change management, library

Assignment – declaration, dupl. Names, scope, limits

Interface – procedure calls and references, I/O, user formats

Checking – error messages, inadequate checks

Data – structure, content

Function – logic, pointers, loops, recursion, computation, function defects

System – configuration, timing, memory

Environment – design, compile, test, support system problems

PSP DESIGN ISSUES

Internal static – static structure of an object

Internal dynamic – Treat object as a state machine.

External static – a function—specification template, including inheritance

External dynamic – interaction of an object with other entities (e.g., call-return behavior)

Scale up – Process scales up to large developments

Process definition – ability to modify development processes to incorporate PSP activities.

PSP SUMMARY

- Principles PSP based on are sound
- Early results show that PSP seems to work
- High overhead to keep data. Not sure how all of it gets used
- Industry apparently neutral about using PSP
  - Results from Philip Johnson in 1999 – After users taught PSP in course, few (none?) continued to use it when not required to later.