ORGANIZATIONAL FRAMEWORKS

Organizational Frameworks

Learning Over Time

How do we build models in a consistent manner over time?
How do we build baselines?
How do we begin to recognize the variations in projects?
How do we create and identify a set of like projects that we can use as a basis for comparison?
How do we evolve our knowledge systematically over time?
How do we store our accumulated knowledge?

How do we select processes?
How do we make recommendations for improvement?

How does an organization take advantage of its measurement program?
How do we integrate our goals?

How do we “experiment” in a way that promotes learning?
Organizational Frameworks

Learning Over Time

We need to build an organizational framework that allows us to:
- evolve knowledge, build models, experiment, learn

We propose a set of mechanisms to do this in the context of a single organization using:
- an evolutionary learning approach based upon the scientific method
  *Quality Improvement Paradigm*
- an organizational structure that collects, stores, analyzes and synthesizes experience over time
  *Experience Factory Organization*

Remember we want

**Practitioners** provided with
- the ability to control and manipulate project solutions
- based upon the environment and goals set for the project
- knowledge based upon empirical and experimental evidence
- of what works and does not work and under what conditions

**Researchers** provided laboratories for experimentation
This will require a research plan that will take place over many years
- coordinating experiments
- evolving with new knowledge
Organizational Frameworks

Quality Improvement Paradigm

Characterize the current project and its environment with respect to models and metrics.

Set quantifiable goals for successful project performance and improvement.

Choose the appropriate process model and supporting methods and tools for this project.

Execute the processes, construct the products, collect, validate, and analyze the data to provide real-time feedback for corrective action.

Analyze the data to evaluate the current practices, determine problems, record findings, and make recommendations for future project improvements.

Package the experience in the form of updated and refined models and other forms of structured knowledge gained from this and prior projects and save it in an experience base to be reused on future projects.
**Quality Improvement Paradigm**

**Step 1: Characterizing the Project and Environment**

**Build models to**
- help us understand what we are doing
- provide a basis for defining goals
- provide a basis for measurement

**Build models of**
- people, processes, products
- and study their interactions

**Use models to**
- classify the current project
- distinguish the relevant project environment
- find the class of projects with similar characteristics and goals

**Models provides a context for**
- Goal Definition
- Reusable Experience/Objects
- Process Selection
- Evaluation/Comparison
- Prediction

---

**Characterization**

**Project Characteristics and Environmental Factors**

**People Factors:** number of people, level of expertise, group organization, problem experience, process experience, ...

**Problem Factors:** application domain, newness to state of the art, susceptibility to change, problem constraints, ...

**Process Factors:** life cycle model, methods, techniques, tools, programming language, other notations, ...

**Product Factors:** deliverables, system size, required qualities, e.g., reliability, portability, ...

**Resource Factors:** target and development machines, calendar time, budget, existing software, ...
Quality Improvement Paradigm

Step 2: Goal Setting and Measurement

Need to establish goals for the processes and products

Goals should be measurable, driven by the models

Goals should be defined from a variety of perspectives:

Customer: predictable schedule, correct functionality
Project: quality controllable process, adherence to schedule
Corporation: reusable experiences, improved quality/productivity over time

There are a variety of mechanisms for defining measurable goals:
Goal/Question/Metric Paradigm (GQM)
Software Quality Metrics Approach (SQM)
Quality Function Deployment Approach (QFD)

Quality Improvement Paradigm

Step 3: Choosing the Processes

We need to choose and tailor an appropriate generic process model, integrated set of methods, and integrated set of techniques

We need to define their goals and give its definitions (models)

Choosing and tailoring are always done in the context of the environment, project characteristics, and goals established for the products and other processes

Examples:

If problem and solution well understood
choose waterfall process model

If high number of faults of omission expected
emphasize traceability reading approach embedded in design inspections

When embedding traceability reading in design inspections, make sure traceability matrix exists
<table>
<thead>
<tr>
<th>Choose The Process</th>
<th>Choosing the Technique: Reading</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input object:</td>
<td>Requirements, specification, design, code, test plan,...</td>
</tr>
<tr>
<td>Output object:</td>
<td>set of anomalies</td>
</tr>
<tr>
<td>Approach:</td>
<td>Sequential, path analysis, stepwise abstraction, ...</td>
</tr>
<tr>
<td>Formality:</td>
<td>Reading, correctness demonstrations, ...</td>
</tr>
<tr>
<td>Emphasis:</td>
<td>Fault detection, traceability, performance, ...</td>
</tr>
<tr>
<td>Method:</td>
<td>Walk-throughs, inspections, reviews, ...</td>
</tr>
<tr>
<td>Consumers:</td>
<td>User, designer, tester, maintainer, ...</td>
</tr>
<tr>
<td>Product qualities:</td>
<td>Correctness, reliability, efficiency, portability...</td>
</tr>
<tr>
<td>Process qualities:</td>
<td>Adherence to method, integration into process,...</td>
</tr>
<tr>
<td>Quality view:</td>
<td>Assurance, control, ...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Choose The Process</th>
<th>Choosing the Technique: Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input object:</td>
<td>System, subsystem, feature, module,..</td>
</tr>
<tr>
<td>Output object:</td>
<td>Test results</td>
</tr>
<tr>
<td>Approach:</td>
<td>structural, functional, error-based, statistical testing,..</td>
</tr>
<tr>
<td>Formality:</td>
<td>Full adherence, partial adherence, ...</td>
</tr>
<tr>
<td>Emphasis:</td>
<td>Fault detection, new features, reliability, performance,...</td>
</tr>
<tr>
<td>Method:</td>
<td>As specified in the test plan</td>
</tr>
<tr>
<td>Consumers:</td>
<td>Various classes of customer/hardware configurations,</td>
</tr>
<tr>
<td>Product qualities:</td>
<td>Reliability, efficiency, ...</td>
</tr>
<tr>
<td>Process qualities:</td>
<td>Adherence to method, integration into process,...</td>
</tr>
<tr>
<td>Quality view:</td>
<td>Assurance, control</td>
</tr>
</tbody>
</table>
Quality Improvement Paradigm

Step 4: Executing the Processes

The development process must support the access and reuse of packaged experience.

Data items must be defined by the models and driven by the goals.

Data collection must be integrated into the processes, not an add on, e.g., defect classification forms part of configuration control mechanism.

Data validation important and necessary. e.g., defect data is error prone.

Education and training in data collection are necessary, everyone must understand the models.

Some analysis must be done in close to real time for feedback for corrective action.

The suppliers of the data need to gain from the data too.

Automated support is necessary to:
- support mechanical tasks
- deal with large amounts of data and information needed for analysis

However, the collection of the most interesting data cannot be automated.

---

Executing The Processes

Kinds of Data Collected

**Resource Data:**
- Effort by activity, phase, type of personnel
- Computer time
- Calendar time

**Change/Defect Data:**
- Changes and defects by various classification schemes

**Process Data:**
- Process definition
- Process conformance
- Domain understanding

**Product Data:**
- Product characteristics
  - logical, e.g., application domain, function
  - physical, e.g., size, structure
  - dynamic, e.g., reliability, coverage
- Use and context information, e.g., design method used
Quality Improvement Paradigm

Step 5: Analyzing the Data

Based upon the goals, we interpret the data that has been collected. We can use this data to:

characterize and understand, e.g.,
what project characteristics effect the choice of processes, methods and techniques?
which phase is typically the greatest source of errors?

evaluate and analyze, e.g.
what is the statement coverage of the acceptance test plan?
does the Cleanroom Process reduce the rework effort?

predict and control, e.g.,
given a set of project characteristics, what is the expected cost and reliability, based upon our history?

motivate and improve, e.g.,
for what classes of errors is a particular technique most effective

Quality Improvement Paradigm

Step 6: Packaging the Experience

Resource Models and Baselines,
e.g., local cost models, resource allocation models

Change and Defect Baselines and Models,
e.g., defect prediction models, types of defects expected for application

Product Models and Baselines,
e.g., actual vs. expected product size and library access over time

Process Definitions and Models,
e.g., process models for Cleanroom, Ada

Method and Technique Evaluations,
e.g., best method for finding interface faults

Products, e.g., Ada generics for simulation of satellite orbits

Quality Models,
e.g., reliability models, defect slippage models, ease of change models

Lessons Learned, e.g., risks associated with an Ada development
Packaging Experience

Forms of Packaged Experience

Equations defining the relationship between variables,
e.g. Effort = 1.48*KSLOC^0.98, 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,

Quality Improvement Paradigm

Reuse Inhibitors

Need to reuse more than just code, need to reuse all kinds of experience
Experience requires the appropriate context definition for to be reusable
Experience needs to be identified and analyzed for its reuse potential
Experience cannot always be reused as is, it needs to be tailored
Experience needs to be packaged to make it easy to reuse
Reuse of experience has been too informal, not supported by the organization
Reuse has to be fully incorporated into the development or maintenance process models
Project focus is delivery, not reuse,
i.e., reuse cannot be a byproduct of software development

Need a separate organization to support the reuse of local experience
Quality Improvement Paradigm

Activity Support for Improvement

Improving the software process and product requires

Learning
- the continual accumulation of evaluated experiences

Experience models
- in a form that can be effectively understood and modified

Experience base
- stored in a repository of integrated experience models

Reuse
- accessible and modifiable to meet the needs of the projects being developed by the organization

Quality Improvement Paradigm

Activity Support For Improvement

Systematic learning requires support for
- recording, off-line generalizing, tailoring, synthesizing and formalizing experience

Packaging and modeling useful experience requires
- a variety of models and formal notations that are tailorable, extendible, understandable, flexible and accessible

An effective experience base must contain
- accessible and integrated set of models that capture the local experiences

Systematic reuse requires support for
- using existing experience
- on-line generalizing or tailoring of candidate experience
Quality Improvement Paradigm

Organizational Support for Improvement

This combination of ingredients requires an organizational structure that supports:

- A software evolution model that supports reuse
- Processes for learning, packaging, and storing experience
- The integration of these two functions

It requires separate logical or physical organizations:

- with different focuses/priorities,
- process models,
- expertise requirements

Quality Improvement Paradigm

Organizational Support for Experience Reuse

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

Experience Factory
- 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

PROJECT ORGANIZATION

Characterize
Set Goals
Choose Process

Execute Process

Execution Plans

EXPERIENCE FACTORY

Tailorable goals, processes, tools
products, resource models, defect models, ... from similar projects

data, lessons learned, ...

experience/environment characteristics

EXPERIENCE FACTORY

Analyze
(Analysis)

Product Support

Experience Base

Package

Generalize
Tailor
Formalize

(Synthesis)

project characteristics
models, baselines, tools, consulting, ...

direct project feedback

products, data, lessons learned, models, ...

12
THE EXPERIENCE FACTORY ORGANIZATION

**Project Organization**

1. Characterize
2. Set Goals
3. Choose Process

Execution plans

4. Execute Process

**Experience**

5. Analyze

Analysis, models, projects, lessons learned

6. Package
Generalize
Tailor
Formalize
Disseminate

**Experience Factory Organization**

**A Different Paradigm**

<table>
<thead>
<tr>
<th><strong>Project Organization</strong></th>
<th><strong>Experience Factory</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem Solving</td>
<td>Experience Packaging</td>
</tr>
</tbody>
</table>

**D**ecomposition of a problem into simpler ones

**Unification of different solutions and re-definition of the problem**

**I**ntentiation

**G**eneralization, **F**ormalization

**D**esign/Implementation process

**A**nalysis/Synthesis process

**V**alidation and **V**erification

**E**xperimentation
AN EXAMPLE EXPERIENCE FACTORY
The Software Engineering Laboratory
Established 1976

Participating Organizations

NASA/Goddard Space Flight Center
University of Maryland
Computer Sciences Corporation

Goals

Understand the software process in at NASA/GSFC
Determine the impact of available technologies
Infuse identified/refined methods back into the development process

Example Gains

Decreased development defect rates by 75% (87 - 91) 37%(91 - 95)
Reduced Cost by 55% (87 - 91) and 42% (91 - 95)
Improved reuse by 300% (87 - 91) and 6% (91 - 95)
Increased functionality five-fold (76 - 92)

AN EXAMPLE EXPERIENCE FACTORY

SEL STRUCTURE

DEVELOPERS

(SOURCE OF EXPERIENCE)

STAFF
275-300 developers
TYPICAL PROJECT SIZE
100-300 KSLOC
ACTIVE PROJECTS
6-10 (at any given time)
PROJECT STAFF SIZE
5-25 people
TOTAL PROJECTS
(1976-1994)
120

PROCESS ANALYSTS

(STAFF)

FUNCTION
- Set goals/questions/metrics
- Design studies/experiments
- Analysis/Research
- Refine software process
- Produce reports/findings

PRODUCTS

(1976-1994)
300 reports/documents

DATA BASE SUPPORT

(MAINTAIN OR EXPERIENCE INFORMATION)

STAFF 3-6 support staff

FUNCTION
- Process forms/data
- QA all data
- Record/archive data
- Maintain SEL data base
- Operate SEL library
The Experience Factory Organization

Some Important Characteristics

The QIP process is **iterative**
- don’t be overly concerned with perfecting any step on the first pass
- the better your initial guess at the baselines, the sooner it will converge

No method is “packaged” that hasn’t been tried:
- applied, analyzed, tailored

Experience Factory provides a way to evaluate
- process conformance and domain understanding
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
  and 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

Iterating 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
  process conformance and domain understanding

Establish **new baselines**

**Repeat** the process and find the next opportunity for improvement
The Experience Factory Organization

Comparison with Other Approaches to Quality

Plan-Do-Check-Act
a quality improvement process based upon a feedback cycle for optimizing a single process model/production line

Total Quality Management
a management approach to long term success through customer satisfaction based on the participation of all members of an organization

SEI Capability Maturity Model
staged process improvement based upon assessment with regard to a set of key process areas until you reach a level 5 which represents a continuous process improvement

Lean Enterprise Management
principle supporting the concentration of production on “value added” activities and the elimination or reduction of “not value added” activities

Approaches To Quality

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
**Approaches To Quality**

**PDCA vs. EF**

**Similarities**
- scientific method
- feedback loops from product to process
- learn from experiments

**Differences**
- PDCA based upon production
  - it attempts to optimize a single process model/production line
  - based upon continual repetition of the same process
  - can collect sufficient data to develop quantitative models
  - can evaluate/predict accurately effects of the process
  - can use accurate models for statistical quality control
- EF based upon development,
  - rarely replicate the same thing twice
  - 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

**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

*Source: Michael Deutsch at Hughes*
## Approaches To Quality

### 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

## APPROACHES TO QUALITY

### 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
- decentralized organization
- quality management
- supplier and customer integration
- internationalization/regionalization

**Result:** A set of processes individualized for each particular product line
**Approaches To Quality**

**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

**Approaches To Quality**

**SEI Capability Maturity Model (CMM)**

Organizational/quality management maturity models by R. Likert/P. Crosby, Software model by R. Radice, made popular by Watts Humphrey at SEI

**Goal:** a level 5 maturity rating, implying continuous process improvement via defect prevention, technology innovation, and process change management

**Approach:** A 5 level process maturity model defined. Maturity level defined based on repeated assessment of an organization’s capability in key process areas. Improvement achieved by action plans for poorly assessed processes

<table>
<thead>
<tr>
<th>Level Focus</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 Optimizing Continuous Process Improvement</td>
</tr>
<tr>
<td>4 Managed Product &amp; Process Quality</td>
</tr>
<tr>
<td>3 Defined Engineering Process</td>
</tr>
<tr>
<td>2 Repeatable Project Management</td>
</tr>
<tr>
<td>1 Initial Heroes</td>
</tr>
</tbody>
</table>

**Result:** A set of well-defined key processes
Approaches To Quality

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

Approaches to Quality

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

Evolution of Process Capability

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

Source: Carnegie Mellon University, Software Engineering Institute
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

Experience Factory 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
Experience Factory Organization

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