Software Process and Product Quality

Conclusions
Course Summary

• Quality engineering concepts and principles
  – Quality engineering activities
  – Measurement fundamentals
  – Seven basic quality tools

• Focus on product quality
  – Defects as a quality indicator
  – Other quality attributes
  – Customer perspectives

• Focus on process quality
  – Project-level and activity-level
  – Process capability assessment and improvement
    • Quality system frameworks
Premise of a Quality Focus

Process Quality → Product Quality → Organization Success
Measure and Improve

Process Quality → Product Quality → Organization Success

Measure quality, analyze results, and identify improvements
Implementing a Quality Improvement System

Definition: Identify data for quality control

Operational definition, metrics

Measurements

Data analysis and interpretation

Data collection

Execution: Measure, analyze, and control quality
Quality Engineering

• Balance cost of quality with cost of poor quality in the context of business objectives

• The ultimate measure of quality is customer satisfaction

• Quality systems and frameworks (principles and practices) guide quality activities, but they cannot guarantee quality
  – Only people can deliver excellence
Scope of Software Quality Engineering

- Quality planning, process improvement
- Inspection, formal methods, defect prevention, fault tolerance, etc.
- Execute software, compare observed behavior to specs.

Software Quality Engineering

Quality Assurance

Testing
Cost of Quality, Cost of Poor Quality

• The cost of getting “it” right is high
• The cost of getting “it” wrong and having to fix may be higher
• Murphy: “There is never enough time to do it right, but there is always enough time to do it over”

• But, up front, can you anticipate where it might be wrong?
  – The cost of focusing on the wrong things might be higher than the cost of fixing it when it is wrong
    • Especially schedule impact costs
  – Software is easy to fix
    • Embrace change

• As an engineer (and business-person), balance the cost of getting it wrong with the cost of wasted effort
Quality Engineering Process

Entry

Quality Planning

Selected QA activities

Quality Assurance Activities

Feedback & adjustments

No

Selected measurements & models

Quality Assessment & Improvement

Analysis & modeling results

Yes

Exit

Quality goals satisfied?
Quality Planning Activities

• Set quality goals by balancing customer expectations with project economics (cost, schedule, scope, risk)
  – Identify customer quality views and attributes
    • Including customer balance of their cost of quality
  – Select direct measures of the quality attributes
  – Set achievable and acceptable goal values of the quality measures

• For the stated quality goals ...
  – Select specific QA activities to achieve quality goals
    • Balance the cost and benefit of the QA activities against the quality goals
  – Select direct and indirect product and process quality measurements and quality models for quality assessment and analysis
Quality Assurance Activities

• Defect prevention
  – Remove (human) error sources
  – Block defects from being injected into software artifacts

• Defect reduction
  – Detect defects
    • Inspection
    • Testing
  – Remove defects
    • “Debugging”—iterate on the software engineering activity
      – Rework requirements, design, code, etc.

• Defect containment
  – Fault tolerance
  – Fault containment
Quality Assessment and Improvement Activities

• Measurement
  – Defect and other product quality measurements
  – Process measurements
• Quality analysis and modeling
  – Analyze measurement data
  – Fit data to analytical models of quality
    • Estimate current and future quality (quality trends)
    • Identify problematic software components or process activities
• Feedback for immediate process improvement
  – Adjust quality goals, project plan, QA plan
  – Adjust quality models
• Feedback for organizational process improvement
  – Improve techniques for quality assurance, quality engineering, and overall software engineering process
Implementing a Quality Improvement System – GQM Approach

The following is based on Goal-Question-Metric Software Acquisition Gold Practice at the DACS Gold Practices Web Site (https://www.goldpractices.com/practices/gqm/)
Integration of GQM Process within the Quality Improvement Paradigm (QIP)

1. Characterize
   - GQM 1: Develop business goals and associated measurement goals for both the project and organization

2. Set goals
   - GQM 2: Generate questions to define or refine the goals

3. Choose process
   - GQM 3: Specify the measures needed to answer the questions and track progress toward goals

4. Execute
   - GQM 4: Develop mechanisms for data collection

5. Analyze
   - GQM 5: Collect, validate and analyze data in real time to provide feedback to projects for corrective action

6. Package
   - GQM 6: Analyze the data in a post mortem fashion to assess conformance to goals and make recommendations for future improvements

Based on: Basili, "Using Measurement to Build Core Competencies in Software", DACS Course, 2005
Phases of GQM Implementation

Source: Solingen, “Experiences in Using the Goal/Question/Metric Paradigm”, 1998
Six Steps of GQM

• Steps 1-3: Definition
  – Use business goals to drive identification of the right metrics

• Steps 4-6: Data Collection and Interpretation
  – Gather the measurement data and make effective use of the measurement results to drive decision making and improvements
Six Steps of GQM
Steps 1-3: Definition

Use business goals to drive identification of the right metrics

1. **Develop** a set of corporate, division and project business goals and associated measurement goals for productivity and quality

2. **Generate questions** (based on models) that define those goals as completely as possible in a quantifiable way

3. **Specify** the measures needed to be collected to answer those questions and track process and product conformance to the goals
Six Steps of GQM

Steps 4-6: Data Collection and Interpretation

Gather the measurement data and make effective use of the measurement results to drive decision making and improvements

4. **Develop mechanisms** for data collection

5. **Collect, validate and analyze** the **data in real time** to provide feedback to projects for corrective action

6. **Analyze the data** in a **postmortem** fashion to assess conformance to the goals and to make recommendations for future improvements
Key Practices of GQM (p. 1 of 3)

• Get the right people involved in the GQM process
• Set explicit measurement goals and state them explicitly
• Don’t create false measurement goals (for example, matching metrics you already have or are easy to get)
• Acquire implicit quality models from the people involved
Key Practices of GQM (p. 2 of 3)

• Consider context
• Derive appropriate metrics
• Stay focused on goals when analyzing data
• Let the data be interpreted by the people involved
• Integrate the measurement activities with regular project activities
Key Practices of GQM (p. 3 of 3)

• Do not use measurements for other purposes (such as to assess team member productivity)
• Secure management commitment to support measurement results
• Establish an infrastructure to support the measurement program
• Ensure that measurement is viewed as a tool, not the end goal
• Get training in GQM before going forward
Measurement Fundamentals

• Tie measurements to the concept of interest
  – Indicators
  – Measurements vs. Metrics
• Measurement scales (nominal, ordinal, interval, ratio) and proper use of measures
• Correlation and Causation
• Reliability and Validity; Systematic and Random Error

Reliable but not valid  Valid but not reliable  Valid and reliable
The Seven Basic Quality Tools

- Checklists (Checksheets)
- Pareto Diagrams
- Histograms
- Run Charts
- Scatter Diagrams (Scatter Plots)
- Control Charts
- Cause-and-Effect (Fishbone) Diagrams
Product Quality
Defects and Failures

Defect prevention and reduction

- Human (developer) Error
- Software Defect (bug)

Fault detection and containment

- System Fault
- System Failure

Build time

- Latent (dormant) defect

Run time
Defects

• Defect measures and metrics
  – Size and defect counts → various “density” metrics
  – Defect type classifications

• Multiple stages of defect removal
  – Inspections at all stages (requirements, design, implementation)
  – Multiple stages of testing (unit, integration, system)
  – Defect removal/containment effectiveness
Software Reliability Engineering

- Customer perspective (failures) vs. developer perspective (defects)
- Operational profiles
  - Identify the most frequently used product features
  - Focus design and testing on the frequently used features
  - And the “important” features (high cost if getting it wrong)
- Predict reliability growth
Product Quality: Big Q vs. little q

- Quality includes many more attributes than just absence of defects

**Reliability**
- Maturity
- Fault-tolerance
- Recoverability

**Functionality**
- Suitability
- Accurateness
- Interoperability
- Compliance
- Security

**Usability**
- Understandability
- Learnability
- Operability

**Portability**
- Adaptability
- Installability
- Conformance
- Replaceability

**Efficiency**
- Time behavior
- Resource behavior

**Maintainability**
- Analyzability
- Changeability
- Stability
- Testability
Process Quality

• Project-level metrics
  – Cycletime, Productivity, Staffing, Requirements volatility, Reuse, Estimation accuracy, Progress

• Activity-specific metrics
  – Requirements, Design, Coding, Testing, Maintenance, Configuration Management, Quality Engineering
Process Maturity and CMMI

- Achieving each level...
  - Establishes a different component in the software process
  - Increases the process capability of the organization

Figure 3.2 The Key Process Areas by Maturity Level
More Mature → Better Way to Run a Business

- Reduced variability
- Improved productivity

Figure 2.4  Process Capability as Indicated by Maturity Level
Assessments and Quality Frameworks

• Assessments baseline quality processes; Frameworks provide benchmarks

• ISO 9000 Family of Standards
  – A general international standard for organizational quality systems
  – Oriented towards assessment and certification

• Malcolm-Baldrige Assessment Discipline
  – A set of criteria for the (US) Malcolm-Baldrige Quality Award.
  – Designed to encourage and recognize excellence

• SEI CMM (Capability Maturity Model family)
  – A software-specific model for improving the maturity of software development practices
  – Oriented towards self-assessment and improvement

• Total Quality Management (TQM)
  – A philosophy and practices for improving quality
  – Focuses on building an organizational quality culture
Applying the Concepts

• Creating a Quality Systems Improvement Plan
  – Quality objectives
  – Quality approach
  – Product quality
    • Defects
    • Other quality attributes
    • Customer satisfaction
  – Process quality
  – Incremental implementation plans
  – Assessment plans
Conclusion

A systematic approach to measuring and improving software product and process quality will help ensure organizational success.

Process Quality → Product Quality → Organization Success

Measure quality, analyze results, and identify improvements.