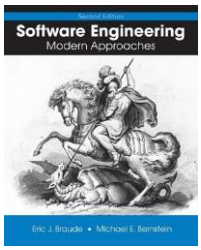
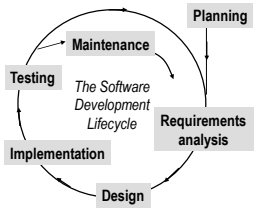
Software Engineering

Modern Approaches



Eric Braude and Michael Bernstein

Chapter 5: Quality in Software Process



Phases relevant to this chapter are shown in bold

Learning goals of this chapter

- What are principles of managing quality?
- How do you plan for "quality?"
- What are inspections and how do you carry them out?
- How do Quality Assurance personnel carry out reviews and audits?
- How do you measure and improve software processes?
- In what way does CMMI assess organizational quality?
- What does a software quality plan look like for a case study?

Quality in the Software Process

- Quality principles
  - overarching quality guidelines
- Quality planning
  - quality plan defines overall approach to quality
- Inspections
  - peer processes focused on quality
- Reviews and Audits
  - external quality assessment
- Defect management
  - identification, tracking and resolution of defects
- Process improvement
  - continuous upgrading of process effectiveness
- Organizational quality
  - engineering competence levels (e.g. CMMI/MPS.Br)

Quality Principles

- Focus continuously on quality
- A quality assurance process must be defined
- The organization must follow its quality assurance process
- Find and repair defects as early in the development process as possible

Defect Repair Cost

Reason (by one estimate):	If defect found ...	
	… soon after creation	… at integration time
Hours to ..		
.. detect	0.7 to 2	0.2 to 10
.. repair	0.3 to 1.2	9+
Total	**1.0 to 3.2**	**9.2 to 19+**

Needed from a Quality Plan

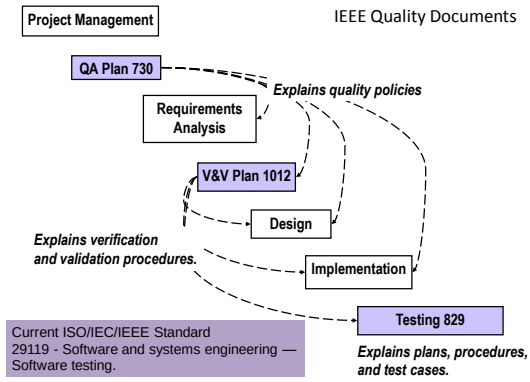
- **Who** will be responsible for quality?
  A person, a manager, a group, an organization, etc.
- What **documentation** will be generated to guide development, verification and validation, use and maintenance of the software?
- What **standards** will be used to ensure quality?
  Documentation standards, coding standards, etc.
- What **metrics** will be used to monitor quality?
  Product and process metrics.
- What **procedures** will be used to **manage** the quality process?
  Meetings, audits, reviews, etc.
- What kind of **testing** will be performed?
- What quality assurance **techniques** will be used?
  Inspections, proofs of correctness, tests, etc.
- How will **defects** be handled?

QA Personnel

QA Person vs. Developer Person-Hours

- 1 QA person per 3-7 developers
- Excludes developer testing
  counted as developer time
- Includes post-developer testing
- Ideally performed by external QA personnel

IEEE Quality Documents



Current ISO/IEC/IEEE Standard 29119 - Software and systems engineering — Software testing.

IEEE Quality Assurance Plan

IEEE 730-2002 Software Quality Assurance Plan
Table of Contents *1 of 2*

IEEE 730-2014 - IEEE Standard for Software Quality Assurance Processes

IEEE Quality Assurance Plan (cont.)

IEEE 739-2002 Software Quality Assurance Plans
Table of Contents *2 of 2*

Inspections

- Quality technique - focus on reviewing the details of a project artifact (requirements, designs, code etc.)
- Note: *not just code*
- Purpose: assure artifact's correctness by seeking defects.
- A meeting of inspectors is held at which defects are identified.

Inspections – Guiding Principles

- Peer process
- Specified roles – strong moderator
- Artifact readiness
- Adequate preparation
- Defect *detection* instead of defect *repair*
  - don't design on the fly
- Use of checklists
- Metrics collection
- Record action items for follow up
- Time limit

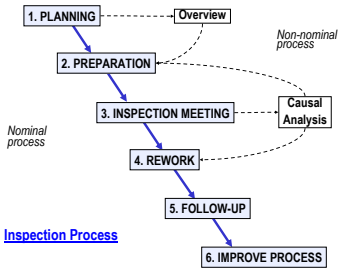
Inspection Roles

- Moderator
- Author
- Recorder
- Reader
- Inspector

Inspection Metrics

- Number of defects discovered, by severity and type
- Number of defects discovered by each category of stakeholder inspecting the artifact
- Number of defects per page reviewed
- Review rate (number of pages/hour)

Inspection Process



Inspection Process

Inspection Costs

Time/Costs per 100 Non-Comment Lines of Code

one company's estimates

Planning	**1 hr**	**×**	**(1 person)**
[Overview	*1 hr*	×	*(3-5 people)]*
Preparation	**1 hr**	**×**	**(2-4 people)**
Inspection meeting	**1 hr**	**×**	**(3-5 people)**
Rework	**1 hr**	**×**	**(1 person)**
[Analysis	*1 hr*	×	*(3-5 people)]*
Total:	**7 - 21 person-hours**		

Options for QA Reviews

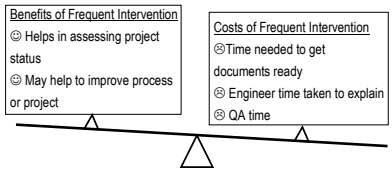
- Participate in all meetings
  Including formative sessions and inspections
- Review all documents
  Participate in all inspections
  (but do not attend all meetings)
- Attend final reviews and review all completed documents
- Review select completed documents
  But do not participate otherwise

Options for QA Audit

- Audit at unrestricted random times
  - Includes visiting with engineers
  - Includes inspecting any document at any time
- Audit random meetings
- Audit randomly from a specified list of meetings
- Audit with notice
- No auditing

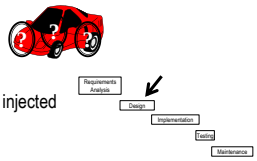
QA Intervention

Frequent vs. Occasional QA Intervention



Overall Defect Classification

- **Severity**
  How serious ☹
- **Priority**
  Order in which defects will be repaired
- **Type**
  What kind of problem
- **Source**
  - Phase during which injected



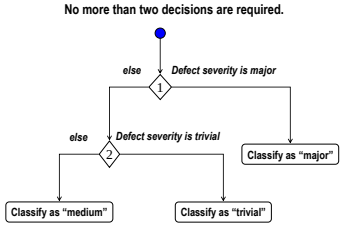
Defect Severity

Triage Severity Classifications

- **Major**
  Causes a requirement to be unsatisfied
- **Medium**
  Neither *major* nor *trivial*
- **Trivial**
  Defect, but doesn't affect operation or maintenance

Defect Severity Triage

Triage Decision Method Applied to Defect Severity Classification



IEEE Severity Classification

IEEE Severity Classification (1044.1)

- **Urgent**
  Failure causes system crash, unrecoverable data loss; or jeopardizes personnel
- **High**
  Causes impairment of critical system functions, and no workaround solution does exist
- **Medium**
  Causes impairment of critical system functions, though a workaround solution does exist
- **Low**
  Causes inconvenience or annoyance
- **None**
  None of the above

Common Defect Types

Common Defect Types Across All Artifacts

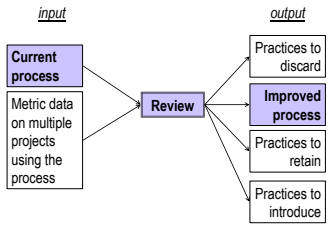
- **Omission**
  Something is missing.
- **Unnecessary**
  The part in question can be omitted.
- **Non-conformance with standards**
- **Inconsistency**
  The part in question contradicts other part(s).
- **Unclassified**
  None of the above

Defect Tracking

Defect Tracking								
Name	**Description**	**Discov-ering enginr.**	**Respon-sible enginr.**	**Date opened**	**Source**	**Seve-rity**	**Type**	**Status**
Check-out flicker	Checkout screen 4 flickers when old DVDs are checked out by hitting the Checkout button.	Kent Bain	Fannie Croft	1/4/04	Inte-gration	Med	GUI	Being worked begun 2/10/04
Bad fine	Fine not correct for first-run DVD's checked out for 2 weeks, as displayed on screen 7.	Fannie Croft	April Breen	1/4/06	Re-quire-ments	High	Math	Not worked yet
…	…	…	…	…	…	…	…	Tested with
…	…	…	…	…	…	…	…	Resolved
…	…	…	…	…	…	…	…	…

Process Improvement

The *Process Improvement* Meta-Process



Measure Process Effectiveness

Process →	Waterfall	Waterfall + Incremental	Pro-cess U	Pro-cess V
Average over 10 projects:				
Major defects identified within first 3 months per 1000SLOC in delivered product	1.3	0.9	0.7	2.1
Development cost per detailed requirement	$120	$100	$85	$135
Developer satisfaction index (1 to 10=best)	4	3	4	3
Customer satisfaction index (1 to 10=best)	4	6	6	2
Cost per maintenance request	$130	$140	$95	$165
Variance in schedule on each phase: 100 × *actual duration - projected duration / projected duration*	+20%	+70%	-10%	+80%
Variance in cost:: 100 × *actual cost - projected cost / projected cost*	+20%	+65%	-5%	+66%
Design fraction: *total design time / total programming time* Humphrey: Should be at least 50%.	23%	51%	66%	20%

A process for Gathering Process Metrics

1. **Identify & define** metrics team will use by phase; include
   ... time spent on 1. research, 2. execution, 3. review
   … size (e.g. lines of code)
   … # defects detected per unit (e.g., lines of code)
   include source
   … quality self-assessment of each on scale of 1-10
   maintain bell-shaped distribution
2. **Document** these in the SQAP
3. **Accumulate** historical **data** by phase
4. Decide **where** the metric data will be **placed**
   - as the project progresses SQAP? SPMP? Appendix?
5. **Designate engineers** to manage collection by phase
   - QA leader or phase leaders (e.g., design leader)
6. **Schedule reviews** of data for lessons learned
   - Specify when and how to feed back improvement

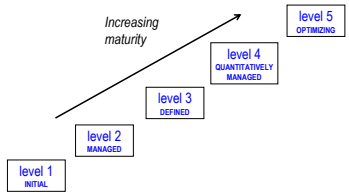
Requirements Document: 200 detailed requirements	Meeting	Research	Execution	Personal Review	Inspection
Hours spent	0.5 x 4	4	5	3	6
% of total time	10%	20%	25%	15%	30%
% of total time: norm for the organization	*15%*	*15%*	*30%*	*15%*	*25%*
Self-assessed quality 1-10	2	8	5	4	6
Defects per 100	N/A	N/A	N/A	5	6
Defects per 100: organization norm	*N/A*	*N/A*	*N/A*	*3*	*4*
Hours spent per detailed requirement	0.01	0.02	0.025	0.015	0.03
Hours spent per detailed requirement: organization norm	0.02	0.02	0.04	0.01	0.03
Process improvement	**Improve strawman brought to meeting**		**Spend 10% more time executing**		**Project Metric Collection for Phases**
Summary	**Productivity: 200/22 = 9.9 detailed requirements per hour**				

CMMI

- Software Engineering Institute (SEI) 1980s
- Measure process capability and maturity
- Two kinds of assessments: *staged* and *continuous*
- Builds on a long history of use that includes case studies and data that demonstrate return on investment

CMMI Staged

***CMMI* Model for Organization with Staged Processes**



CMMI Maturity Levels

Level, Title, and Summary	Expected Outcome	Characteristics
1. INITIAL		
Undefined; ad hoc	Unpredictable; depends entirely on team individuals.	Organizations often produce products, but they are frequently over budget and miss their schedules.
2. MANAGED	Preceding level plus:	
Measurement and control	Project outcomes are qualitatively predictable	Respect organizational policies Follow established plans Provide adequate resources Establish responsibility and authority Provide training Establish configuration management Monitor and control: Take corrective action Evaluate process and product relative to plans: Address deviations

CMMI Maturity Levels

3. DEFINED	Preceding level plus:	
Processes standardized	Projects consistent across the organization; qualitatively predictable	Establish process objectives Ensure process objectives met Establish orderly tailoring Describe processes rigorously Be proactive
4.QUANTI-TATIVELY MANAGED	Preceding level plus:	
Processes measured	Metrics available on process; quantitatively predictable	Set quantitative goals for key sub-processes Control key sub-processes with statistical techniques. Identify and remedy variants
5. OPTIMIZING	Preceding level plus:	
Improvement meta-process	Processes improved and adapted using process metrics	Establish quantitative process improvement objectives Identify and implement innovative process improvements Identify and implement incremental process improvements Evaluate process improvements against quantitative objectives.