Software Productivity Research LLC

SOFTWARE QUALITY IN 2010:

A SURVEY OF THE STATE OF THE ART

Capers Jones
Founder and Chief Scientist Emeritus
President, Capers Jones & Associates LLC

http://www.spr.com
cjonesiii@cs.com

November 2, 1010

SOURCES OF QUALITY DATA

Data collected from 1984 through 2010

- **About 675 companies (150 clients in Fortune 500 set)**
- **About 35 government/military groups**
- **About 13,500 total projects**
- **New data = about 50-75 projects per month**
- **Data collected from 24 countries**
- **Observations during more than 15 lawsuits**

SOFTWARE QUALITY HAZARDS IN TEN INDUSTRIES

INDUSTRY	HAZARD
Airlines	Safety hazards

Air traffic control problems

Flight schedule confusion

Navigation equipment failures

Maintenance schedules thrown off

Delay in opening Denver airport

Passengers booked into non-existent seats

Passengers misidentified as terror suspects

Suspicious shipments may not be identified

 SWQUAL08\3

SOFTWARE QUALITY HAZARDS IN TEN INDUSTRIES

INDUSTRY	HAZARD
Defense	Security hazards

Base security compromised

Computer security compromised

Strategic weapons malfunction

Command, communication network problems

Aircraft maintenance records thrown off

Logistics and supply systems thrown off

Satellites malfunction

 SWQUAL08\4

SOFTWARE QUALITY HAZARDS IN TEN INDUSTRIES

INDUSTRY	**HAZARD**
Finance	**Financial transaction hazards**

Interest calculations in error

Account balances thrown off

Credit card charges in error

Funds transfer thrown off

Mortgage/loan interest payments in error

Mortgage paperwork lost in cyberspace

Hacking and identity theft due to software security flaws

Denial of service attacks due to software security flaws

SOFTWARE QUALITY HAZARDS IN TEN INDUSTRIES

INDUSTRY	**HAZARD**
Health Care	**Safety hazards**

Patient monitoring devices malfunction

Operating room schedules thrown off

Medical instruments malfunction

Prescription refill problems

Hazardous drug interactions

Billing problems

Medical records stolen or released by accident

SOFTWARE QUALITY HAZARDS IN TEN INDUSTRIES

INDUSTRY	HAZARD
Insurance	Liability, benefit hazards

- Policy due dates in error
- Policies cancelled in error
- Benefits and interest calculation errors
- Annuities miscalculated
- Errors in actuarial studies
- Payment records in error

SOFTWARE QUALITY HAZARDS IN TEN INDUSTRIES

INDUSTRY	HAZARD
State, Local Governments	Local economic hazards

- School taxes miscalculated
- Jury records thrown off
- Real-estate transactions misfiled
- Divorce, marriage records misfiled
- Alimony, child support payment records lost
- Death records filed for wrong people
- Traffic light synchronization thrown off
- Errors in property tax assessments

SOFTWARE QUALITY HAZARDS IN TEN INDUSTRIES

INDUSTRY	HAZARD
Manufacturing	**Operational hazards**

Subcontract parts fail to arrive

Purchases of more or less than economic order quantities

Just-in-time arrivals thrown off

Assembly lines shut down

Aging errors for accounts receivable and cash flow

Aging errors for accounts payable and cash flow

Pension payments miscalculated

SOFTWARE QUALITY HAZARDS IN TEN INDUSTRIES

INDUSTRY	HAZARD
National Government	**Citizen record hazards**

Tax records in error

Annuities and entitlements miscalculated

Social Security payments miscalculated or cancelled

Disbursements miscalculated

Retirement benefits miscalculated

Personal data stolen or released by accident

Voting errors or hacking of vote tabulations

SOFTWARE QUALITY HAZARDS IN TEN INDUSTRIES

INDUSTRY	HAZARD
Public Utilities	**Safety hazards**

Electric meters malfunction

Gas meters malfunction

Distribution of electric power thrown off

Billing records in error

Nuclear power plants malfunction

SWQUAL08\11

SOFTWARE QUALITY HAZARDS IN TEN INDUSTRIES

INDUSTRY	HAZARD
Telecommunications	**Service disruption hazards**

Intercontinental switching disrupted

Domestic call switching disrupted

Billing records in error

Errors in recharging prepaid call cards

SWQUAL08\12

SOFTWARE QUALITY HAZARDS ALL INDUSTRIES

1. **Software is blamed for more major business problems than any other man-made product.**
2. **Poor software quality has become one of the most expensive topics in human history: > $150 billion per year in U.S.; > $500 billion per year world wide.**
3. **Projects cancelled due to poor quality >15% more costly than successful projects of the same size and type.**
4. **Software executives, managers, and technical personnel are regarded by many CEO's as a painful necessity rather than top professionals.**
5. **Improving software quality is a key topic for all industries.**

 SWQUAL08\13

BASIC DEFINITIONS

SOFTWARE QUALITY	**Software that combines the characteristics of low defect rates and high user satisfaction**
USER SATISFACTION	**Clients who are pleased with a vendor's products, quality levels, ease of use, and support**
DEFECT PREVENTION	**Technologies that minimize the risk of making errors in software deliverables**
DEFECT REMOVAL	**Activities that find and correct defects in software deliverables**
BAD FIXES	**Secondary defects injected as a byproduct of defect repairs**

 SWQUAL08\14

FUNDAMENTAL SOFTWARE QUALITY METRICS

- **Defect Potentials**
  - **requirements errors, design errors, code errors, document errors, bad fix errors, test plan errors, and test case errors**
- **Defects Removed**
  - **by origin of defects**
  - **before testing**
  - **during testing**
  - **during deployment**
- **Defect Removal Efficiency**
  - **ratio of development defects to customer defects**
- **Defect Severity Levels (Valid defects)**
  - **fatal, serious, minor, cosmetic**

FUNDAMENTAL SOFTWARE QUALITY METRICS (cont.)

- **Duplicate Defects**
- **Invalid Defects**
- **Defect Removal Effort and Costs**
  - **preparation**
  - **execution**
  - **repairs and rework**
  - **effort on duplicates and invalids**
- **Supplemental Quality Metrics**
  - **complexity**
  - **test case volumes**
  - **test case coverage**
  - **IBM’s orthogonal defect categories**

FUNDAMENTAL SOFTWARE QUALITY METRICS (cont.)

- **Standard Cost of Quality**
  - Prevention
  - Appraisal
  - Failures
- **Revised Software Cost of Quality**
  - Defect Prevention
  - Non-Test Defect Removal
  - Testing Defect Removal
  - Post-Release Defect Removal
- **Error-Prone Module Effort**
  - Identification
  - Removal or redevelopment
  - repairs and rework

U.S. AVERAGES FOR SOFTWARE QUALITY

(Data expressed in terms of defects per function point)

Defect Origins	Defect Potential	Removal Efficiency	Delivered Defects
Requirements	1.00	77%	0.23
Design	1.25	85%	0.19
Coding	1.75	95%	0.09
Documents	0.60	80%	0.12
Bad Fixes	0.40	70%	0.12
TOTAL	5.00	85%	0.75

(Function points show all defect sources - not just coding defects)

BEST IN CLASS SOFTWARE QUALITY

(Data expressed in terms of defects per function point)

Defect Origins	Defect Potential	Removal Efficiency	Delivered Defects
Requirements	0.40	85%	0.08
Design	0.60	97%	0.02
Coding	1.00	99%	0.01
Documents	0.40	98%	0.01
Bad Fixes	0.10	95%	0.01
TOTAL	2.50	96%	0.13

OBSERVATIONS

Most often found in systems software > SEI CMM Level 3

POOR SOFTWARE QUALITY - MALPRACTICE

(Data expressed in terms of defects per function point)

Defect Origins	Defect Potential	Removal Efficiency	Delivered Defects
Requirements	1.50	50%	0.75
Design	2.20	50%	1.10
Coding	2.50	80%	0.50
Documents	1.00	70%	0.30
Bad Fixes	0.80	50%	0.40
TOTAL	8.00	62%	3.05

OBSERVATIONS

Most often found in large client-server projects (> 5000 FP).

Software Quality Result Comparisons: 1000 FP

(Data expressed in terms of defects per function point)

Method	Defect Potential	Removal Efficiency	Delivered Defects
TSP	2.70	97%	0.08
CMMI 5	3.00	96%	0.12
RUP	3.90	95%	0.20
CMMI 3	4.50	93%	0.32
XP	4.50	92%	0.38
Agile	4.70	91%	0.42
CMMI 1	5.00	85%	0.75

OBSERVATIONS

TSP keeps high levels of removal efficiency above 10,000 function points.

Agile removal efficiency declines above 1,000 function points.

GOOD QUALITY RESULTS > 90% SUCCESS RATE

- Formal Inspections (Requirements, Design, and Code)
- Static analysis (for about 25 languages out of 2,500 in all)
- Joint Application Design (JAD)
- Software Six-Sigma methods (tailored for software projects)
- Quality Metrics using function points
- Quality Function Deployment (QFD)
- Defect Removal Efficiency Measurements
- Automated Defect tracking tools
- Active Quality Assurance (> 5% SQA staff)
- Utilization of TSP/PSP approaches
- => Level 3 on the SEI capability maturity model (CMMI)
- Virtualization for reuse and debugging
- Quality Estimation Tools
- Automated Test Support Tools + testing specialists
- Root-Cause Analysis

MIXED QUALITY RESULTS: < 50% SUCCESS RATE

- Informal test case design without mathematical analysis
- Independent Verification & Validation (IV & V)
- Total quality management (TQM)
- Independent quality audits
- Six-Sigma quality programs (without software adjustments)
- Baldrige Awards
- IEEE Quality Standards
- Testing only by Developers
- DOD 2167A and DOD 498
- Reliability Models
- Quality circles in the United States (more success in Japan)
- Clean-room methods
- Cost of quality without software modifications

POOR QUALITY RESULTS: < 25% SUCCESS RATE

- ISO 9000 - 9004 Quality Standards
- Informal Testing
- Passive Quality Assurance (< 3% QA staff)
- Token Quality Assurance (< 1% QA staff)
- LOC Metrics for quality (omits non-code defects)
- Cost per defect metric (penalizes quality)
- Failure to estimate quality or risks early

A PRACTICAL DEFINITION OF SOFTWARE QUALITY (PREDICTABLE AND MEASURABLE)

- **Low Defect Potentials (< 2.5 per Function Point)**
- **High Defect Removal Efficiency (> 95%)**
- **Unambiguous, Stable Requirements (< 2.5% change)**
- **Explicit Requirements Achieved (> 97.5% achieved)**
- **High User Satisfaction Ratings (> 90% "excellent")**
  - **Installation**
  - **Ease of learning**
  - **Ease of use**
  - **Functionality**
  - **Compatibility**
  - **Error handling**
  - **User information (screens, manuals, tutorials)**
  - **Customer support**
  - **Defect repairs**

SOFTWARE QUALITY OBSERVATIONS

Quality Measurements Have Found:

- **Individual programmers -- Less than 50% efficient in finding bugs in their own software**
- **Normal test steps -- often less than 75% efficient (1 of 4 bugs remain)**
- **Design Reviews and Code Inspections -- often more than 65% efficient; have topped 90%**
- **Inspections, static analysis, virtualization, plus formal testing – are often more than 95% efficient; have hit 99%**
- **Reviews, Inspections, static analysis, and virtualization -- lower costs and schedules by as much as 30%**

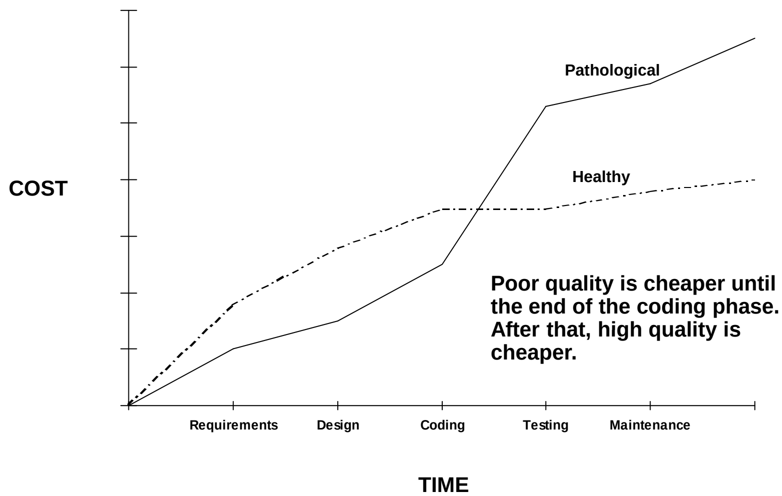
SOFTWARE DEFECT ORIGINS

- 1) Requirements: Hardest to prevent and repair
- 2) Design: Most severe and pervasive
- 3) Code: Most numerous; easiest to fix
- 4) Documentation: Can be serious if ignored
- 5) Bad Fixes: Very difficult to find
- 6) Bad Test Cases: Common and troublesome
- 7) Data quality: Common but hard to measure
- 8) Web content: Unmeasured to date

SOFTWARE DEFECT SEVERITY CATEGORIES

Severity 1:	TOTAL FAILURE S	1% at release
Severity 2:	MAJOR PROBLEMS	20% at release
Severity 3:	MINOR PROBLEMS	35% at release
Severity 4:	COSMETIC ERRORS	44% at release
INVALID	USER OR SYSTEM ERRORS	15% of reports
DUPLICATE	MULTIPLE REPORTS	30% of reports
ABEYANT	CAN'T RECREATE ERROR	5% of reports

HOW QUALITY AFFECTS SOFTWARE COSTS



 SWQUAL08\29

U. S. SOFTWARE QUALITY AVERAGES CIRCA 2010

(Defects per Function Point)

	System Software	Commercial Software	Information Software	Military Software	Outsource Software
Defect Potentials	6.0	5.0	4.5	7.0	5.2
Defect Removal Efficiency	94%	90%	73%	96%	92%
Delivered Defects	0.4	0.5	1.2	0.3	0.4
First Year Discovery Rate	65%	70%	30%	75%	60%
First Year Reported Defects	0.26	0.35	0.36	0.23	0.30

 SWQUAL08\30

U. S. SOFTWARE QUALITY AVERAGES CIRCA 2010

(Defects per Function Point)

	Web Software	Embedded Software	SEI-CMM 3 Software	SEI-CMM 1 Software	Overall Average
Defect Potentials	4.0	5.5	3.0	5.5	5.1
Defect Removal Efficiency	72%	95%	95%	73%	86.7%
Delivered Defects	1.1	0.3	0.15	1.5	0.68
First Year Discovery Rate	95%	90%	60%	35%	64.4%
First Year Reported Defects	1.0	0.27	0.09	0.52	0.43

SOFTWARE SIZE VS DEFECT REMOVAL EFFICIENCY

(Data Expressed in terms of Defects per Function Point)

Size	Defect Potential	Defect Removal Efficiency	Delivered Defects	1st Year Discovery Rate	1st Year Reported Defects
1	1.85	95.00%	0.09	90.00%	0.08
10	2.45	92.00%	0.20	80.00%	0.16
100	3.68	90.00%	0.37	70.00%	0.26
1000	5.00	85.00%	0.75	50.00%	0.38
10000	7.60	78.00%	1.67	40.00%	0.67
100000	9.55	75.00%	2.39	30.00%	0.72
AVERAGE	*5.02*	*85.83%*	*0.91*	*60.00%*	*0.38*

SOFTWARE DEFECT POTENTIALS AND DEFECT REMOVAL EFFICIENCY FOR EACH LEVEL OF SEI CMM

(Data Expressed in Terms of Defects per Function Point For projects nominally 1000 function points in size)

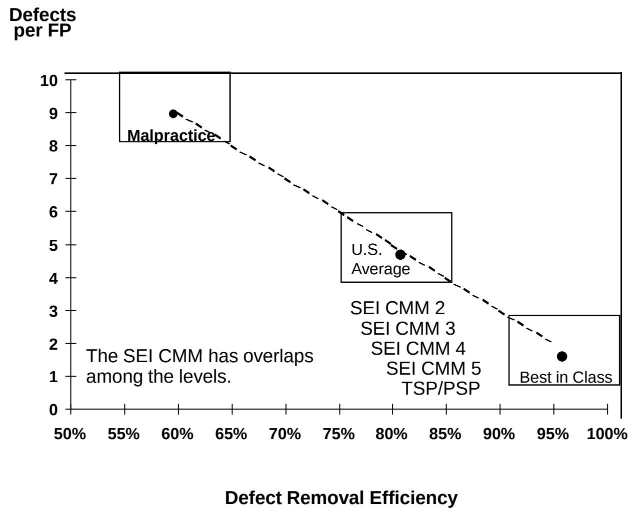
SEI CMM Levels	Defect Potentials	Removal Efficiency	Delivered Defects
SEI CMMI 1	5.00	85%	0.75
SEI CMMI 2	4.70	87%	0.60
SEI CMMI 3	4.50	93%	0.32
SEI CMMI 4	4.00	95%	0.20
SEI CMMI 5	3.00	96%	0.12
SEI CMMI 6 (TSP/PSP)	2.70	97%	0.08

SOFTWARE DEFECT POTENTIALS AND DEFECT REMOVAL EFFICIENCY FOR EACH LEVEL OF SEI CMM

(Data Expressed in Terms of Defects per Function Point For projects > 10,000 function points in size)

SEI CMM Levels	Defect Potentials	Removal Efficiency	Delivered Defects
SEI CMMI 1	6.00	80%	1.20
SEI CMMI 2	5.50	85%	0.82
SEI CMMI 3	5.00	90%	0.50
SEI CMMI 4	4.60	93%	0.32
SEI CMMI 5	4.25	95%	0.20
SEI CMMI 6 (TSP/PSP)	4.00	97%	0.12

MAJOR SOFTWARE QUALITY ZONES



SWQUAL08\35

INDUSTRY-WIDE DEFECT CAUSES

Ranked in order of effort required to fix the defects:

1. **Requirements problems (omissions; changes, errors)**
2. **Design problems (omissions; changes; errors)**
3. **Interface problems between modules**
4. **Logic, branching, and structural problems**
5. **Memory allocation problems**
6. **Testing omissions and poor coverage**
7. **Test case errors**
8. **Stress/performance problems**
9. **Bad fixes/Regressions**
10. **Documentation errors**

SWQUAL08\36

OPTIMIZING QUALITY AND PRODUCTIVITY

Projects that achieve 95% cumulative Defect Removal Efficiency will find:

1) **Minimum schedules**
2) **Maximum productivity**
3) **High levels of user and team satisfaction**
4) **Low levels of delivered defects**
5) **Low levels of maintenance costs**
6) **Low risk of litigation**

INDUSTRY DATA ON DEFECT ORIGINS

Because defect removal is such a major cost element, studying defect origins is a valuable undertaking.

IBM Corporation (MVS)

45%	Design errors
25%	Coding errors
20%	Bad fixes
5%	Documentation errors
5%	Administrative errors
100%	

SPR Corporation (client studies)

20%	Requirements errors
30%	Design errors
35%	Coding errors
10%	Bad fixes
5%	Documentation errors
100%	

TRW Corporation

60%	Design errors
40%	Coding errors
100%	

Mitre Corporation

64%	Design errors
36%	Coding errors
100%	

Nippon Electric Corp.

60%	Design errors
40%	Coding errors
100%	

SOFTWARE QUALITY AND PRODUCTIVITY

- **The most effective way of improving software productivity and shortening project schedules is to reduce defect levels.**

- **Defect reduction can occur through:**

1. Defect <u>prevention</u> technologies
Structured design and JAD
Structured code
Use of inspections, static analysis
Reuse of certified components

2. Defect <u>removal</u> technologies
Design inspections
Code inspections, static analysis
Virtualization
Formal Testing

DEFECT PREVENTION METHODS

DEFECT PREVENTION

- **Joint Application Design (JAD)**
- **Quality function deployment (QFD)**
- **Software reuse (high-quality components)**
- **Root cause analysis**
- **Six-Sigma quality programs for software**
- **Usage of TSP/PSP methods**
- **Climbing > Level 3 on the SEI CMMI**
- **Virtualization, static analysis, inspections**

DEFECT PREVENTION - Continued

DEFECT PREVENTION

- **Total quality management (TQM)**
- **Quality measurements**
- **Quality Circles**
- **Orthogonal defect analysis**
- **Defect tracking tools**
- **Formal design inspections**
- **Formal code inspections**
- **Embedding users with development team (Agile methods)**

DEFECT REMOVAL METHODS

DEFECT REMOVAL

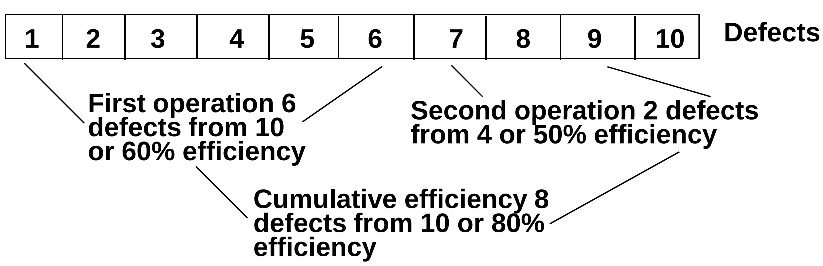
- **Requirements inspections**
- **Design inspections**
- **Test plan inspections**
- **Test case inspections**
- **Static analysis (C, Java, COBOL, SQL etc.)**
- **Code inspections**
- **Automated testing (unit, performance)**
- **All forms of manual testing (more than 40 kinds of test)**

Copyright © 2010 by Capers Jones. All Rights Reserved.

DEFECT REMOVAL EFFICIENCY

- **Defect removal efficiency is a key quality measure**
- **Removal efficiency =** $\dfrac{\textbf{Defects found}}{\textbf{Defects present}}$
- **“Defects present” is the critical parameter**

DEFECT REMOVAL EFFICIENCY - continued



Defect removal efficiency = **Percentage of defects removed by a single level of review, inspection or test**

Cumulative defect removal efficiency = **Percentage of defects removed by a series of reviews, inspections or tests**

DEFECT REMOVAL EFFICIENCY EXAMPLE

DEVELOPMENT DEFECTS	
Inspections	500
Testing	400
Subtotal	900
USER-REPORTED DEFECTS IN FIRST 90 DAYS	
Valid unique defects	100
TOTAL DEFECT VOLUME	
Defect totals	1000
REMOVAL EFFICIENCY	
Dev. (900) / Total (1000) =	90%

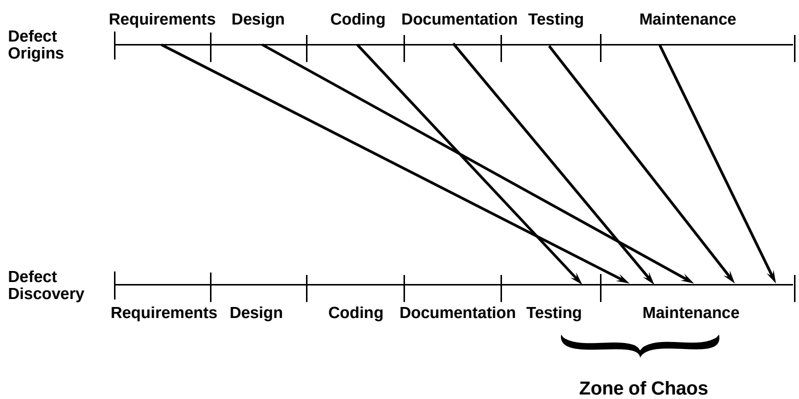
SWQUAL08\45

RANGES OF DEFECT REMOVAL EFFICIENCY

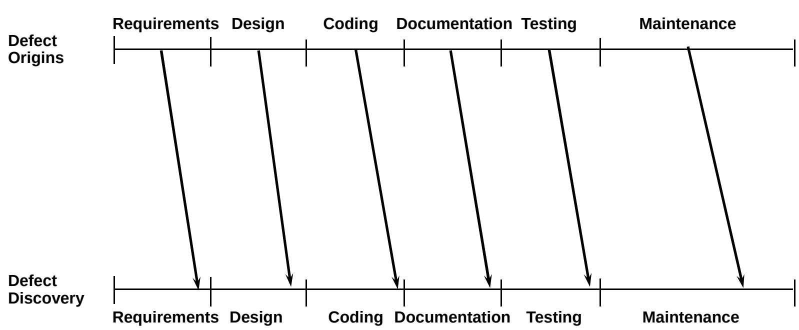
	Lowest	Median	Highest
1 Requirements review	20%	30%	50%
2 Top-level design reviews	30%	40%	60%
3 Detailed functional design reviews	30%	45%	65%
4 Detailed logic design reviews	35%	55%	75%
5 Code inspection or static analysis	35%	60%	85%
6 Unit tests	10%	25%	50%
7 New Function tests	20%	35%	55%
8 Integration tests	25%	45%	60%
9 System test	25%	50%	65%
10 External Beta tests	15%	40%	75%
CUMULATIVE EFFICIENCY	**75%**	**97%**	**99.99%**

SWQUAL08\46

NORMAL DEFECT ORIGIN/DISCOVERY GAPS



DEFECT ORIGINS/DISCOVERY WITH INSPECTIONS



SOFTWARE DEFECT REMOVAL RANGES

WORST CASE RANGE

TECHNOLOGY COMBINATIONS	DEFECT REMOVAL EFFICIENCY		
	Lowest	Median	Highest
1. No Design Inspections No Code Inspections or static analysis No Quality Assurance No Formal Testing	30%	40%	50%

SWQUAL08\49

SOFTWARE DEFECT REMOVAL RANGES (cont.)

SINGLE TECHNOLOGY CHANGES

TECHNOLOGY COMBINATIONS	DEFECT REMOVAL EFFICIENCY		
	Lowest	Median	Highest
2. No design inspections No code inspections or static analysis FORMAL QUALITY ASSURANCE No formal testing	32%	45%	55%
3. No design inspections No code inspections or static analysis No quality assurance FORMAL TESTING	37%	53%	60%
4. No design inspections CODE INSPECTIONS/STATIC ANALYSIS No quality assurance No formal testing	43%	57%	65%
5. FORMAL DESIGN INSPECTIONS No code inspections or static analysis No quality assurance No formal testing	45%	60%	68%

SWQUAL08\50

SOFTWARE DEFECT REMOVAL RANGES (cont.)

TWO TECHNOLOGY CHANGES

TECHNOLOGY COMBINATIONS	DEFECT REMOVAL EFFICIENCY		
	Lowest	Median	Highest
6. No design inspections No code inspections or static analysis FORMAL QUALITY ASSURANCE FORMAL TESTING	50%	65%	75%
7. No design inspections FORMAL CODE INSPECTIONS/STAT. AN. FORMAL QUALITY ASSURANCE No formal testing	53%	68%	78%
8. No design inspections FORMAL CODE INSPECTIONS/STAT.AN. No quality assurance FORMAL TESTING	55%	70%	80%

SWQUAL08\51

SOFTWARE DEFECT REMOVAL RANGES (cont.)

TWO TECHNOLOGY CHANGES - continued

TECHNOLOGY COMBINATIONS	DEFECT REMOVAL EFFICIENCY		
	Lowest	Median	Highest
9. FORMAL DESIGN INSPECTIONS No code inspections or static analysis FORMAL QUALITY ASSURANCE No formal testing	60%	75%	85%
10. FORMAL DESIGN INSPECTIONS No code inspections or static analysis No quality assurance FORMAL TESTING	65%	80%	87%
11. FORMAL DESIGN INSPECTIONS FORMAL CODE INSPECTIONS/STAT.AN. No quality assurance No formal testing	70%	85%	90%

SWQUAL08\52

SOFTWARE DEFECT REMOVAL RANGES (cont.)

THREE TECHNOLOGY CHANGES

TECHNOLOGY COMBINATIONS	DEFECT REMOVAL EFFICIENCY		
	Lowest	Median	Highest
12. No design inspections FORMAL CODE INSPECTIONS/STAT.AN. FORMAL QUALITY ASSURANCE FORMAL TESTING	75%	87%	93%
13. FORMAL DESIGN INSPECTIONS No code inspections or static analysis FORMAL QUALITY ASSURANCE FORMAL TESTING	77%	90%	95%
14. FORMAL DESIGN INSPECTIONS FORMAL CODE INSPECTIONS/STAT. AN. FORMAL QUALITY ASSURANCE No formal testing	83%	95%	97%
15. FORMAL DESIGN INSPECTIONS FORMAL CODE INSPECTIONS/STAT.AN. No quality assurance FORMAL TESTING	85%	97%	99%

SOFTWARE DEFECT REMOVAL RANGES (cont.)

BEST CASE RANGE

TECHNOLOGY COMBINATIONS	DEFECT REMOVAL EFFICIENCY		
	Lowest	Median	Highest
1. FORMAL DESIGN INSPECTIONS STATIC ANALYSIS FORMAL CODE INSPECTIONS FORMAL QUALITY ASSURANCE FORMAL TESTING	95%	99%	99.99%

DISTRIBUTION OF 1500 SOFTWARE PROJECTS BY DEFECT REMOVAL EFFICIENCY LEVEL

Defect Removal Efficiency Level (Percent)	Number of Projects	Percent of Projects
> 99	6	0.40%
95 - 99	104	6.93%
90 - 95	263	17.53%
85 - 90	559	37.26%
80 - 85	408	27.20%
< 80	161	10.73%
Total	1,500	100.00%

SOFTWARE QUALITY UNKNOWNS IN 2010

SOFTWARE QUALITY TOPICS NEEDING RESEARCH:

Errors in software test plans and test cases

Errors in web content such as graphics and sound

Correlations between security flaws and quality flaws

Supply chain defect removal

Error content of data bases, repositories, warehouses

Causes of bad-fix injection rates

Impact of complexity on quality and defect removal

Impact of creeping requirements

2010 QUALITY RESEARCH TOPICS

Quality levels of Agile projects (more data needed)

Quality levels of Extreme (XP) programming

Quality levels of object-oriented (OO) development

Quality levels of web applications

Quality levels of Microsoft applications

Quality levels of Linux and open source software

Quality levels or ERP applications

Effectiveness of automatic testing methods

CONCLUSIONS ON SOFTWARE QUALITY

- **No single quality method is adequate by itself.**
- **Six-Sigma provides the broadest quality focus**
- **Formal inspections, static analysis are most efficient**
- **Inspections + static analysis + testing > 97% efficient.**
- **Defect prevention + removal best overall**
- **Quality excellence has ROI > $15 for each $1 spent**
- **High quality benefits schedules, productivity, users**
- **Virtualization is also a quality tool**

REFERENCES ON SOFTWARE QUALITY

Gilb, Tom & Graham, Dorothy; Software Inspection; Addison Wesley, 1993.

Jones, Capers; Best Practices in Software Engineering; McGraw Hill, 2009

Jones, Capers; Estimating Software Costs, McGraw Hill, 2007.

Jones, Capers; Assessments, Benchmarks, and Best Practices, Addison Wesley, 2000.

Jones, Capers; Applied Software Measurement; McGraw Hill, 2008.

Jones, Capers; Software Quality: Analysis and Guidelines for Success; International Thomson; 1997.

Kan, Steve; Metrics and Models in Software Quality Engineering, Addison Wesley, 2003.

Radice, Ron; High-quality, Low-cost Software Inspections, Paradoxican Publishing, 2002.

Wiegers, Karl; Peer Reviews in Software, Addison Wesley, 2002.

REFERENCES ON SOFTWARE QUALITY

www.ASQ.org	(American Society for Quality)
www.IFPUG.org	(Int. Func. Pt. Users Group)
www.ISBSG.org	(Int. Software Bench. Standards Group)
www.ISO.org	(International Organization for Standards)
www.ITMPI.org	(Infor. Tech. Metrics and Productivity Institute)
www.PMI.org	(Project Management Institute)
www.SEI.org	(Software Engineering Institute)
www.SPR.com	(Software Productivity Research LLC)
www.SSQ.org	(Society for Software Quality)
www.semat.org	(Software Engineering Methods and Tools)
www.cisq.org	(Consortium for IT software quality)

SQGNE Software Quality Group of New England

SQGNE is made possible by the support of our sponsors:

ASQ Software Division

Oracle and
Sun Integrated Systems

Software Quality Consulting
consulting • training • auditing

Logo design: Sarah Cole Design

Welcome to our 17th season!

- An all-volunteer group with no membership dues!
- Supported entirely by our sponsors…
- Over 700+ members
- Monthly meetings - Sept to July on 2nd Wed of month
- E-mail list - contact John Pustaver **pustaver@ieee.org**
- **SQGNE Web site: www.sqgne.org**

SQGNE Software Quality Group of New England

Volunteers / Hosts / Mission

Officers and Volunteers

- John Pustaver – President and Founder
- Steve Rakitin – VP and Programs
- Gene Freyberger – Annual Survey
- Howie Dow – Treasurer
- Dawn Wu – Clerk and official Greeter

Our gracious Hosts:

- Paul Ratty – Board of Dir
- Tom Arakel
- Margaret Shinkle
- Jack Guilderson

Mission

- To promote use of engineering and management techniques that lead to delivery of high quality software
- To disseminate concepts and techniques related to software quality engineering and software engineering process
- To provide a forum for discussion of concepts and techniques related to software quality engineering and the software engineering process
- To provide networking opportunities for software quality professionals

SQGNE Software Quality Group of New England

ASQ Software Division

- Software Quality Live - for ASQ SW Div members…
- Software Quality Professional Journal www.asq.org/pub/sqp/
- CSQE Certification info at www.asq.org/software/getcertified
- SW Div info at www.asq.org/software

SQGNE Software Quality Group of New England

SQGNE 20010-11 Schedule

Speaker	Affiliation	Date	Topic
Steve and Howie Dow		9/8/10	Test your Testing Aptitude!
Stan Wrobel	CSC	10/13/10	CMM vs. Agile - Finding the right fit for your project
Capers Jones	SPR	11/10/10	SOFTWARE QUALITY IN 2010: A SURVEY OF THE STATE OF THE ART
Linda McInnis		12/8/10	Career Paths for SQA Professionals
Robin Goldsmith	GoPro Management	1/12/11	Add Steak to Exploratory Testing's Parlor-Trick Sizzle
Rick Spiewak		2/9/11	A fundamental approach to improving software quality
Stephen P Berczuk		3/9/11	Build, SCM, and QA: Enablers for Agility
Johanna Rothman	Rothman & Assoc.	4/13/11	SQA in an agile environment
Damon Poole	AccuRev	5/11/11	Is Agile Any Better?
Marc Rene	MetLife Auto & Home	6/8/11	Maximizing the Value of Testing to the Business First Annual Election for SQGNE Board of Directors and At-large Members
Everyone		7/13/10	Annual Hot Topics Night…

SQGNE Software Quality Group of New England

Tonight's Speaker…

Software Quality in 2010: A Survey of the State of the Art

Capers Jones, Capers Jones & Associates

Software quality is a topic of importance throughout the world. Unfortunately software quality assurance groups are often understaffed for the work at hand, and also undercapitalized and under equipped. This presentation attempts to cover the known factors which influence software quality results, including methodologies, tools, and staffing levels. The presentation provides empirical data on the impact of major quality approaches, such as the six-sigma approach, TSO/TSP, clean-room methods, ISO certification, the Software Engineering Institute (SEI) capability (CMM) level concept, and other topics that can impact overall quality levels by as much as 1%. The presentation utilizes the function point metric for quantifying quality results. U.S. software averages about 5 defects per function point, with about 85% of these being removed prior to delivery. Empirical data is provided on software quality levels in a number of industries, and in the major industrialized countries.

Bio: Capers Jones is currently the President and CEO of Capers Jones & Associates LLC. He is also the founder and former chairman of Software Productivity Research LLC (SPR). He holds the title of Chief Scientist Emeritus at SPR. Capers Jones founded SPR in 1984. Before founding SPR Capers was Assistant Director of Programming Technology for the ITT Corporation at the Programming Technology Center in Stratford, Connecticut. He was also a manager and researcher at IBM in California. Capers is a well-known author and international public speaker.

SQGNE Software Quality Group of New England