Namcook Analytics LLC

SOFTWARE QUALITY IN 2012:

A SURVEY OF THE STATE OF THE ART

Capers Jones, CTO

Web: www.Namcook.com
Email: Capers.Jones3@GMAILcom

May 1, 2012

SOURCES OF QUALITY DATA

Data collected from 1984 through 2012

- **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**

BASIC DEFINITIONS OF SOFTWARE QUALITY

- **Functional Software Quality**
  Software that combines low defect rates and high levels Of user satisfaction. The software should also meet all user requirements and adhere to international standards.
- **Structural Software Quality**
  Software that exhibits a robust architecture and can operate In a multi-tier environment without failures or degraded performance. Software has low cyclomatic complexity levels.
- **Aesthetic Software Quality**
  Software with elegant and easy to use commands and Interfaces, attractive screens, and well formatted outputs.

ECONOMIC DEFINITIONS OF SOFTWARE QUALITY

- **"Technical debt"**
  The assertion (by Ward Cunningham in 1992) that quick and careless development with poor quality leads to many years of expensive maintenance and enhancements.
- **Cost of Quality (COQ)**
  The overall costs of prevention, appraisal, internal failures, and external failures. For software these mean defect prevention, pre-test defect removal, testing, and post-release defect repairs. (Consequential damages are usually not counted.)
- **Total Cost of Ownership (TCO)**
  The sum of development + enhancement + maintenance + support from day 1 until application is retired. (Recalculation at 5 year intervals is recommended.)

FUNDAMENTAL SOFTWARE QUALITY METRICS

- **Defect Potentials**
  - **Sum of requirements errors, design errors, code errors, document errors, bad fix errors, test plan errors, and test case errors**
- **Defect Discovery Efficiency (DDE)**
  - **Percent of defects discovered before release**
- **Defect Removal Efficiency (DRE)**
  - **Percent of defects removed before release**
- **Defect Severity Levels (Valid unique defects)**
  - **Severity 1 = Total stoppage**
  - **Severity 2 = Major error**
  - **Severity 3 = Minor error**
  - **Severity 4 = Cosmetic error**

FUNDAMENTAL SOFTWARE QUALITY METRICS (cont.)

- **Standard Cost of Quality**
  - **Prevention**
  - **Appraisal**
  - **Internal failures**
  - **External failures**
- **Revised Software Cost of Quality**
  - **Defect Prevention**
  - **Pre-Test Defect Removal (inspections, static analysis)**
  - **Testing Defect Removal**
  - **Post-Release Defect Removal**
- **Error-Prone Module Effort**
  - **Identification**
  - **Removal or redevelopment**
  - **repairs and rework**

QUALITY MEASUREMENT PROBLEMS

- **Cost per defect penalizes quality!**
- **(Buggiest software has lowest cost per defect!)**
- **Lines of code penalize high-level languages!**
- **Lines of code ignore non-coding defects!**
- **Most companies don't measure all defects!**
- **Most common omissions are requirement bugs, design bugs, and bugs found by desk checks and unit testing. Real bugs can outnumber measured bugs by more than 5 to 1!**

COST PER DEFECT PENALIZES QUALITY

	Case A High quality	Case B Low quality
Defects found	50	500
Test case creation	$10,000	$10,000
Test case execution	$10,000	$10,000
Defect repairs	$10,000	$70,000
TOTAL	$30,000	$90,000
Cost per Defect	$600	$180
$ Cost savings	$60,000	$0.00

LINES OF CODE HARM HIGH-LEVEL LANGUGES

	Case A JAVA	Case B C
KLOC	50	125
Function points	1,000	1,000
Code defects found	500	1,250
Defects per KLOC	10.00	10.00
Defects per FP	0.5	1.25
Defect repairs	$70,000	$175,000
$ per KLOC	$1,400	$1,400
$ per Defect	$140	$140
$ per Function Point	$70	$175
$ cost savings	$105,000	$0.00

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)
(Code defects = 35% of total 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 or in TSP projects)

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 water fall projects > 10,000 Function Points).

GOOD QUALITY RESULTS > 90% SUCCESS RATE

- Formal Inspections (Requirements, Design, and Code)
- Text static analysis
- Code static analysis (for about 25 languages out of 2,500 in all)
- Joint Application Design (JAD)
- Requirements modeling
- Functional quality metrics using function points
- Structural quality metrics such as cyclomatic complexity
- Defect Detection Efficiency (DDE) measurements
- Defect Removal Efficiency (DRE) measurements
- Automated defect tracking tools
- Active quality Assurance (> 3% SQA staff)
- Mathematical test case design based on design of experiments
- Quality estimation tools
- Testing specialists (certified)
- Root-Cause Analysis

MIXED QUALITY RESULTS: < 50% SUCCESS RATE

- CMMI level 3 or higher (some overlap among CMMI levels:
  Best CMMI 1 groups better than worst CMMI 3 groups)
- ISO and IEEE quality standards (Prevent low quality;
  Little benefit for high-quality teams)
- Six-Sigma methods (unless tailored for software projects)
- Quality function deployment (QFD)
- Independent Verification & Validation (IV & V)
- Quality circles in the United States (more success in Japan)
- Clean-room methods for rapidly changing requirements
- Kaizan (moving from Japan to U.S. and elsewhere)
- Cost of quality without software modifications
- Pair programming

POOR QUALITY RESULTS: < 25% SUCCESS RATE

- **Testing as only form of defect removal**
- **Informal Testing and uncertified test personnel**
- **Testing only by developers; no test specialists**
- **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**
- **Quality measurement "leakage" such as unit test bugs**

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%**
- **Static analysis –often more than 65% efficient; has topped 95%**
- **Inspections, static analysis, and testing combined lower costs and schedules by > 20%; lower total cost of ownership (TCO) by > 45%**

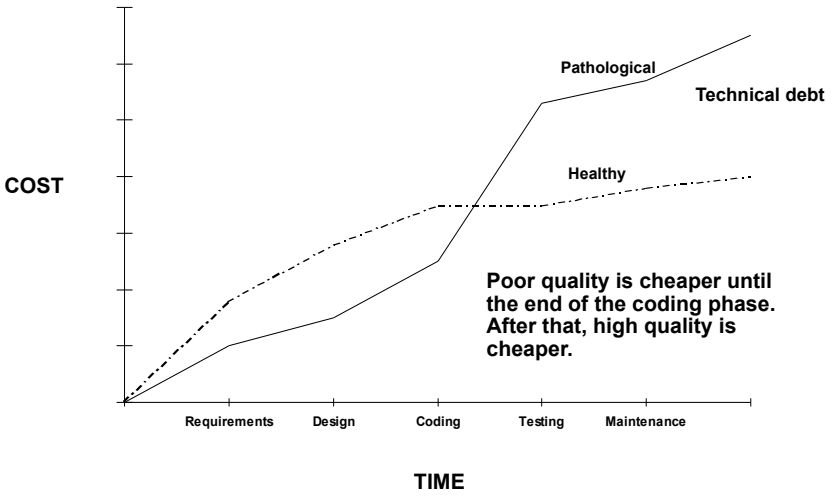
SOFTWARE DEFECT ORIGINS

1. Requirements	Hardest to prevent and repair
2. Requirements creep	Very troublesome source of bugs
3. Architecture	Key to structural quality
4. Design	Most severe and pervasive
5. Source code	Most numerous; easiest to fix
6. Security flaws	Hard to find and hard to fix
7. Documentation	Can be serious if ignored
8. Bad fixes	Very difficult to find
9. Bad test cases	Numerous but seldom measured
10. Data errors	Very common but not measured
11. Web content	Very common but not measured
12. Structure	Hard to find by testing; inspections and static analysis can identify multi-tier platform defects

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
STRUCTURAL	MULTI-TIER DEFECTS	15% of reports
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



U. S. SOFTWARE QUALITY AVERAGES CIRCA 2012

(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	0.36	0.50	1.22	0.28	0.42
First Year Discovery Rate	65%	70%	30%	75%	60%
First Year Reported Defects	0.23	0.35	0.36	0.21	0.25

U. S. SOFTWARE QUALITY AVERAGES CIRCA 2012

(Defects per Function Point)

	Web Software	Embedded Software	SEI-CMM 3 Software	SEI-CMM 1 Software	Overall Average
Defect Potentials	4.0	5.5	5.0	5.75	5.1
Defect Removal Efficiency	72%	95%	95%	83%	86.7%
Delivered Defects	1.12	0.3	0.25	0.90	0.68
First Year Discovery Rate	95%	90%	60%	35%	64.4%
First Year Reported Defects	1.06	0.25	0.15	0.34	0.42

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	5.25	80%	1.05
SEI CMMI 2	5.00	85%	0.75
SEI CMMI 3	4.75	90%	0.48
SEI CMMI 4	4.50	93%	0.32
SEI CMMI 5	4.25	96%	0.17

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	6.50	75%	1.63
SEI CMMI 2	6.25	82%	1.13
SEI CMMI 3	5.50	87%	0.71
SEI CMMI 4	5.25	90%	0.53
SEI CMMI 5	4.75	94%	0.29

DEFECTS AND SOFTWARE METHODOLGOIES

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

Software methods	Defect Potential	Removal Efficiency	Delivered Defects
Waterfall		5.50	80%
1.10			
Iterative	4.75	87%	0.62
Object-Oriented	4.50	88%	0.54
Agile with scrum	4.00	90%	0.40
Rational Unified Process (RUP)	4.25	94%	0.26
PSP and TSP	3.50	96%	0.14
Model-based	3.00	98%	0.06
85% Certified reuse	1.75	99%	0.02

DEFECTS AND SOFTWARE METHODOLGOIES

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

Software methods	Defect Potential	Removal Efficiency	Delivered Defects
Waterfall	7.00	75%	1.75
Iterative	6.25	82%	1.13
Object-Oriented	5.75	85%	0.86
Agile with scrum	5.50	87%	0.72
Rational Unified Process (RUP)	5.50	90%	0.55
PSP and TSP	5.00	94%	0.30
Model-based	4.00	96%	0.15
85% Certified reuse	2.25	96%	0.09

GLOBAL SOFTWARE QUALITY SAMPLES

Data Expressed in Terms of Defects per Function Point
Top countries for software quality out of 66

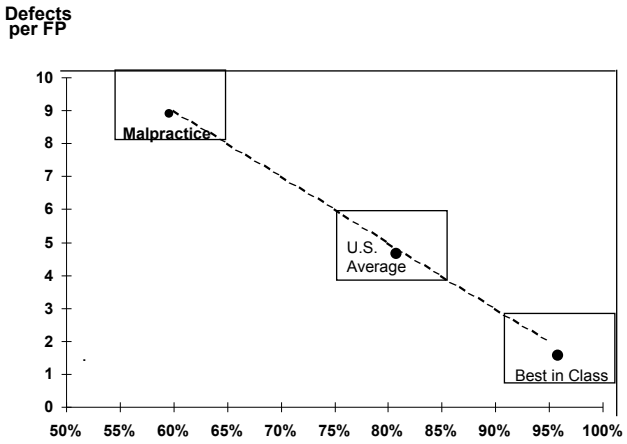
Country	Defect Potential	Removal Efficiency	Delivered Defects
Japan	4.50	93.50%	0.29
India	4.90	93.00%	0.34
Denmark	4.80	92.00%	0.38
Canada	4.75	91.75%	0.39
South Korea	4.90	92.00%	0.39
Switzerland	5.00	92.00%	0.40
United Kingdom	5.10	91.50%	0.40
Israel	5.10	92.00%	0.41

GLOBAL SOFTWARE QUALITY SAMPLES

Data Expressed in Terms of Defects per Function Point
Selected Countries out of 66 compared

Country	Defect Potential	Removal Efficiency	Delivered Defects
United States	4.82	90.15%	0.47
France	4.85	90.00%	0.49
Germany	4.95	88.00%	0.59
Italy	4.95	87.50%	0.62
Spain	4.90	86.50%	0.66
Russia	5.15	86.50%	0.70
China	5.20	86.50%	0.70
Ukraine	4.95	85.00%	0.74

MAJOR SOFTWARE QUALITY ZONES



DEFECT REMOVAL EFFICIENCY CASE 1

Inspections + static analysis + testing

DEVELOPMENT DEFECTS REMOVED

Static analysis	350
Inspections	390
Testing	250
Subtotal	990

USER-REPORTED DEFECTS IN FIRST 90 DAYS

Valid unique defects	10

TOTAL DEFECT VOLUME

Defect totals	1,000

DEFECT REMOVAL EFFICIENCY

Dev. (990) / Total (1,000) = 99.0%

DEFECT REMOVAL EXPENSES CASE 1

Inspections + static analysis + testing	
Static analysis	**$12,772**
Inspections	**$70,773**
Testing	**$220,889**
Subtotal	**$304,434**
Maintenance	**$6,629**
TOTAL	**$310,703 (lowest cost)**

DEFECT REMOVAL EFFICIENCY CASE 2

No static analysis. Inspections + testing

DEVELOPMENT DEFECTS REMOVED	
Static analysis	**0**
Inspections	**560**
Testing	**400**
Subtotal	**960**
USER-REPORTED DEFECTS IN FIRST 90 DAYS	
Valid unique defects	**40**
TOTAL DEFECT VOLUME	
Defect totals	**1,000**
DEFECT REMOVAL EFFICIENCY	
Dev. (960) / Total (1,000) =	**96.0%**

DEFECT REMOVAL EXPENSES CASE 2

No static analysis. inspections + testing

Static analysis	**$0**
Inspections	**$91,387**
Testing	**$230,203**
Subtotal	**$321,590**
Maintenance	**$13,932**
TOTAL	**$335,522**

DEFECT REMOVAL EFFICIENCY CASE 3

No inspections. Static analysis + testing

DEVELOPMENT DEFECTS REMOVED

Static analysis	**500**
Inspections	**0**
Testing	**425**
Subtotal	**925**

USER-REPORTED DEFECTS IN FIRST 90 DAYS

Valid unique defects	**75**

TOTAL DEFECT VOLUME

Defect totals	**1,000**

DEFECT REMOVAL EFFICIENCY

Dev. (925) / Total (1,000) = 92.5%

DEFECT REMOVAL EXPENSES CASE 3

No inspections; static analysis + testing

Static analysis	**$12,772**
Inspections	**$0**
Testing	**$264,045**
Subtotal	**$276,817**
Maintenance	**$41,796**
TOTAL	**$318,613**

DEFECT REMOVAL EFFICIENCY CASE 4

No inspections; no static analysis. Testing only.

DEVELOPMENT DEFECTS REMOVED	
Inspections	**0**
Static analysis	**0**
Testing	**850**
Subtotal	**850**
USER-REPORTED DEFECTS IN FIRST 90 DAYS	
Valid unique defects	**150**
TOTAL DEFECT VOLUME	
Defect totals	**1,000**
DEFECT REMOVAL EFFICIENCY	
Dev. (850) / Total (1,000) =	**85.0%**

DEFECT REMOVAL EXPENSES CASE 4

Inspections + static analysis + testing		
Static analysis	**$0**	
Inspections	**$0**	
Testing	**$326,089**	
Subtotal	**$326,089**	
Maintenance	**$92,879**	
TOTAL	**$418,968**	**(Highest cost)**

COSTS FOR CASES 1 THROUGH 4

	Removal	Repairs	Total	
Case 1	**$304,434**	**$6,289**	**$310,703**	**Best**
Case 2	**$321,590**	**$13,932**	**$335,522**	
Case 3	**$276,817**	**$41,796**	**$318,613**	
Case 4	**$326,089**	**$92,879**	**$418,698**	**Worst**

Note: Defect removal costs predicted by Software Risk Master™

REMOVAL FOR FOR CASES 1 THROUGH 4

	Efficiency	Removed	Delivered	
Case 1	99.0%	990	10	Best
Case 2	96.0%	940	40	
Case 3	92.5%	925	75	
Case 4	85.0%	850	150	Worst

Note: Defect removal efficiency predicted by Software Risk Master™

INDUSTRY DATA ON DEFECT ORIGINS

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

IBM Corporation (MVS)		SPR Corporation (client studies)	
45%	Design errors	20%	Requirements errors
25%	Coding errors	30%	Design errors
20%	Bad fixes	35%	Coding errors
5%	Documentation errors	10%	Bad fixes
5%	Administrative errors	5%	Documentation errors
100%		100%	

TRW Corporation		MITRE Corporation		Nippon Electric Corp.	
60%	Design errors	64%	Design errors	60%	Design errors
40%	Coding errors	36%	Coding errors	40%	Coding errors
100%		100%		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:**

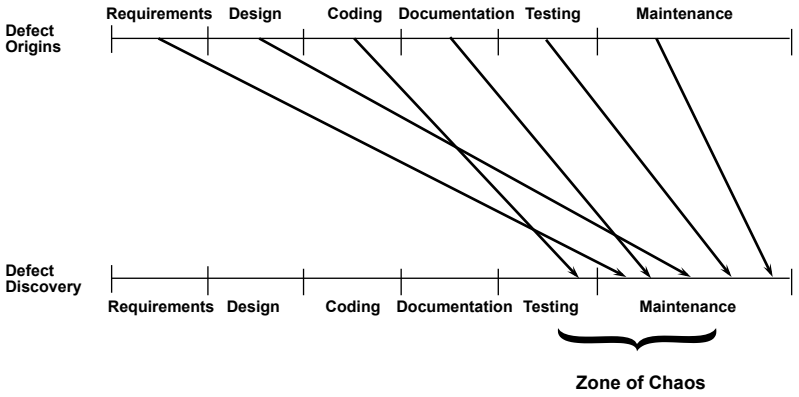
1. **Defect prevention technologies**
   - **Structured design and JAD**
   - **Structured code**
   - **Use of inspections, static analysis**
   - **Reuse of certified components**

2. **Defect removal technologies**
   - **Design inspections**
   - **Code inspections, static analysis**
   - **Formal Testing using mathematical test case design**

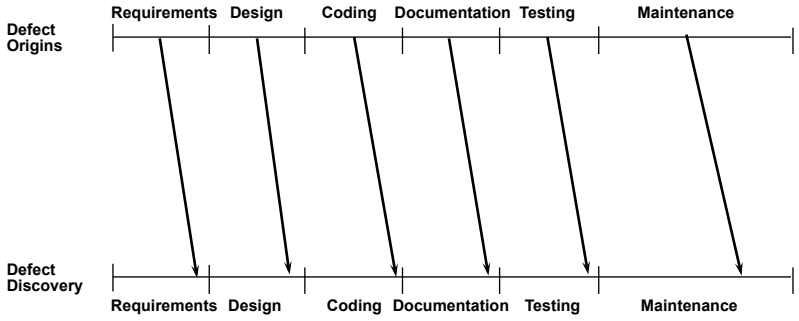
RANGES OF DEFECT REMOVAL EFFICIENCY

	Lowest	Median	Highest
1 Requirements review (informal)	20%	30%	50%
2 Top-level design reviews (informal)	30%	40%	60%
3 Detailed functional design inspection	30%	65%	85%
4 Detailed logic design inspection	35%	65%	75%
5 Code inspection or static analysis	35%	60%	90%
6 Unit tests	10%	25%	50%
7 New Function tests	20%	35%	65%
8 Integration tests	25%	45%	60%
9 System test	25%	50%	65%
10 External Beta tests	15%	40%	75%
CUMULATIVE EFFICIENCY	75%	98%	99.99%

NORMAL DEFECT ORIGIN/DISCOVERY GAPS



DEFECT ORIGINS/DISCOVERY WITH INSPECTIONS



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%

CONCLUSIONS ON SOFTWARE QUALITY

- **No single quality method is adequate by itself.**
- **Formal inspections, static analysis, models are effective**
- **Inspections + static analysis + testing > 97% efficient.**
- **Defect prevention + removal best overall**
- **QFD, models, inspections, & six-sigma prevent defects**
- **Higher CMMI levels, TSP, RUP, Agile, XP are effective**
- **Quality excellence has ROI > $15 for each $1 spent**
- **High quality benefits schedules, productivity, users**

REFERENCES ON SOFTWARE QUALITY

Black, Rex; Managing the Testing Process; Microsoft Press, 1999.

Crosby, Phiip B.; Quality is Free; New American Library, Mentor Books, 1979.

Gack Gary, Managing the Black Hole; Business Expert Publishing, 2009

Gilb, Tom & Graham, Dorothy; Software Inspections; Addison Wesley, 1983.

Jones, Capers & Bonsignour, Olivier; The Economics of Software Quality; Addison Wesley, 2011

Jones, Capers; Software Engineering Best Practices; McGraw Hill, 2010

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

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

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

REFERENCES ON SOFTWARE QUALITY

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

McConnell; Steve; Code Complete 2; Microsoft Press, 2004

Pirsig, Robert; Zen and the Art of Motorcycle Maintenance; Bantam; 1984

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.CASTSoftware.com	**(CAST Software Inc.)**
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.processfusion.net	**(Process Fusion)**
www.SEI.CMU.edu	**(Software Engineering Institute)**
www.SPR.com	**(Software Productivity Research LLC)**

REFERENCES ON SOFTWARE QUALITY

www.SSQ.org	**(Society for Software Quality)**
www.SEMAT.org	**(Software Engineering Methods and Theory)**
www.CISQ.org	**(Consortium for IT software quality)**
www.SANS.org	**Sans Institute listing of software defects**
www.eoqsg.org	**European Institute for Software Qualiy**
www,Galorath.com	**Galorath Associates**
www.Namcook.com	**Namcook Analytics LLC**

www.associationforsoftwaretesting.org
Association for Software Testing

www.qualityassuranceinstitute.com
Quality Assurance Institute (QAI)