Software Productivity Research LLC

SOFTWARE QUALITY IN 2008:

A SURVEY OF THE STATE OF THE ART

Capers Jones
Founder and Chief Scientist Emeritus

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

January 30, 2008

SOURCES OF SPR'S QUALITY DATA

SPR clients from 1984 through 2008

- About 650 companies (150 clients in Fortune 500 set)
- About 35 government/military groups
- About 12,500 total projects
- New data = about 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	

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	

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	
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

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	

SOFTWARE QUALITY HAZARDS IN TEN INDUSTRIES

INDUSTRY	HAZARD
Telecommunications	**Service disruption hazards**
Intercontinental switching disrupted	
Domestic call switching disrupted	
Billing records in error	

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.**

3. **Software executives, managers, and technical personnel are regarded by many CEO's as a painful necessity rather than top professionals.**

4. **Improving software quality is a key topic for all industries.**

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**

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).

GOOD QUALITY RESULTS > 90% SUCCESS RATE

- **Formal Inspections (Requirements, Design, and Code)**
- **Joint Application Design (JAD)**
- **Software Six-Sigma methods (tailored for software projects)**
- **Quality Metrics using function points**
- **Quality Metrics using IBM's Orthogonal classification**
- **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 (CMM)**
- **Formal Test Plans for Major Projects**
- **Quality Estimation Tools**
- **Automated Test Support Tools**
- **Testing Specialists**
- **Root-Cause Analysis**

MIXED QUALITY RESULTS: < 50% SUCCESS RATE

- **Total Quality Management (TQM)**
- **Independent Verification & Validation (IV & V)**
- **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**
- **Clean-room methods**
- **Cost of quality without software modifications**

POOR QUALITY RESULTS: < 25% SUCCESS RATE

- **ISO 9000 - 9004 Quality Standards**
- **Informal Testing**
- **Agile development methods > 5000 function points**
- **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)**
- **Rapid Application Development (RAD) > 5000 func. Pts.**

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**

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%**
- **Reviews or inspections plus formal testing -- are often more than 95% efficient; have hit 99%**
- **Reviews and Inspections -- 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 FAILURES**	**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

U. S. SOFTWARE QUALITY AVERAGES CIRCA 2008

(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

U. S. SOFTWARE QUALITY AVERAGES CIRCA 2008

(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)

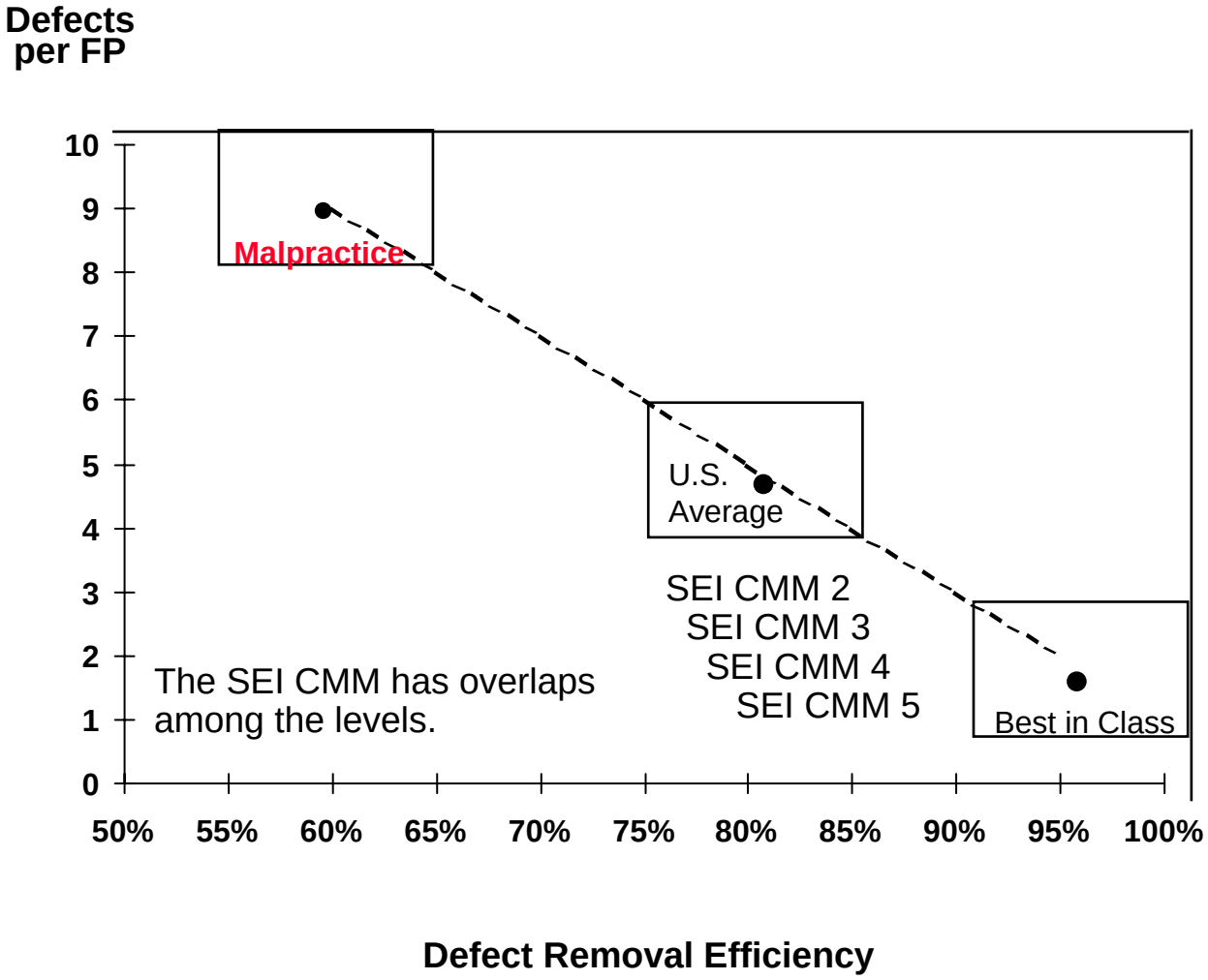
SEI CMM Levels	Defect Potentials	Removal Efficiency	Delivered Defects
SEI CMM 1	5.00	80%	1.00
SEI CMM 2	4.00	90%	0.40
SEI CMM 3	3.00	95%	0.15
SEI CMM 4	2.00	97%	0.08
SEI CMM 5	1.00	99%	0.01
SEI CMM 6 (TSP/PSP)	1.00	99.5%	<0.01

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

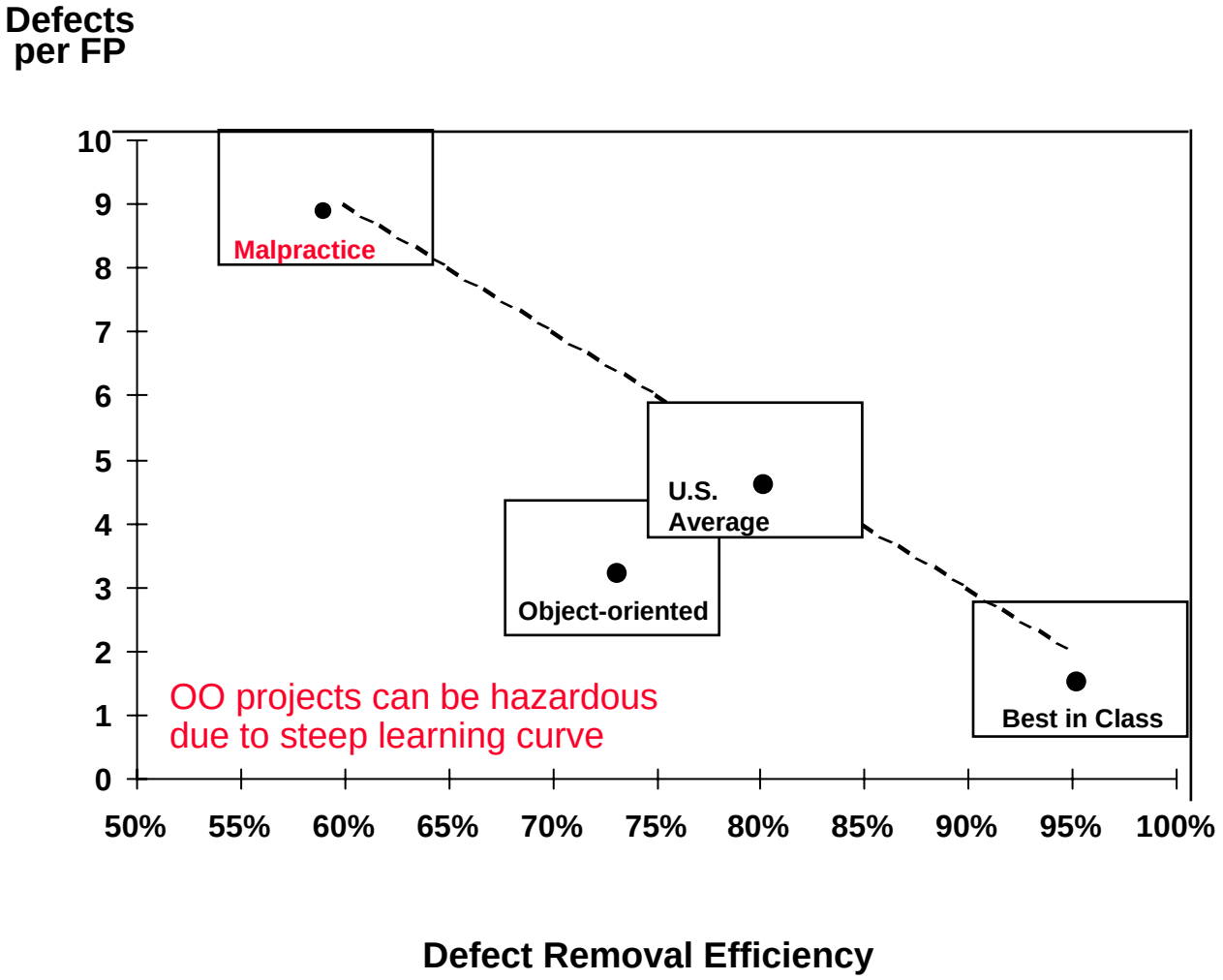
(Data Expressed in Terms of Defects per Function Point
For projects > 5000 function points in size)

SEI CMM Levels	Defect Potentials	Removal Efficiency	Delivered Defects
SEI CMM 1	5.50	73%	1.48
SEI CMM 2	4.00	90%	0.40
SEI CMM 3	3.00	95%	0.15
SEI CMM 4	2.50	97%	0.008
SEI CMM 5	2.25	98%	0.005
SEI CMM 6 (TSP/PSP)	2.00	99%	0.004

MAJOR SOFTWARE QUALITY ZONES



SOFTWARE QUALITY IMPROVEMENT (cont.)



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**

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 prevention technologies**
     - **Structured design and JAD**
     - **Structured code**
     - **Reuse of certified components**

  2. **Defect removal technologies**
     - **Design inspections**
     - **Code inspections**
     - **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 CMM**
- **IBM “clean room” methods**

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

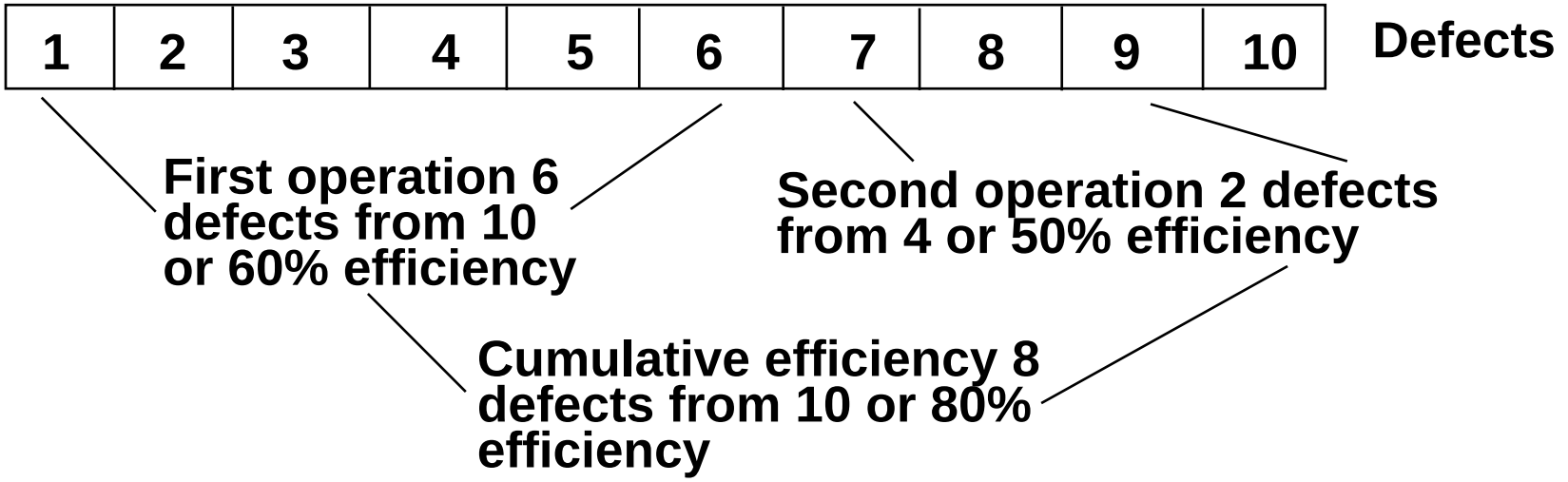
DEFECT REMOVAL

- **Requirements inspections**
- **Design inspections**
- **Test plan inspections**
- **Test case inspections**
- **Code inspections**
- **User manual inspections**
- **Data quality inspections**
- **All forms of testing (more than 15 kinds of test)**

DEFECT REMOVAL EFFICIENCY

- **Defect removal efficiency is a key quality measure**
- **Removal efficiency =** $\dfrac{\text{Defects found}}{\text{Defects present}}$
- **“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**

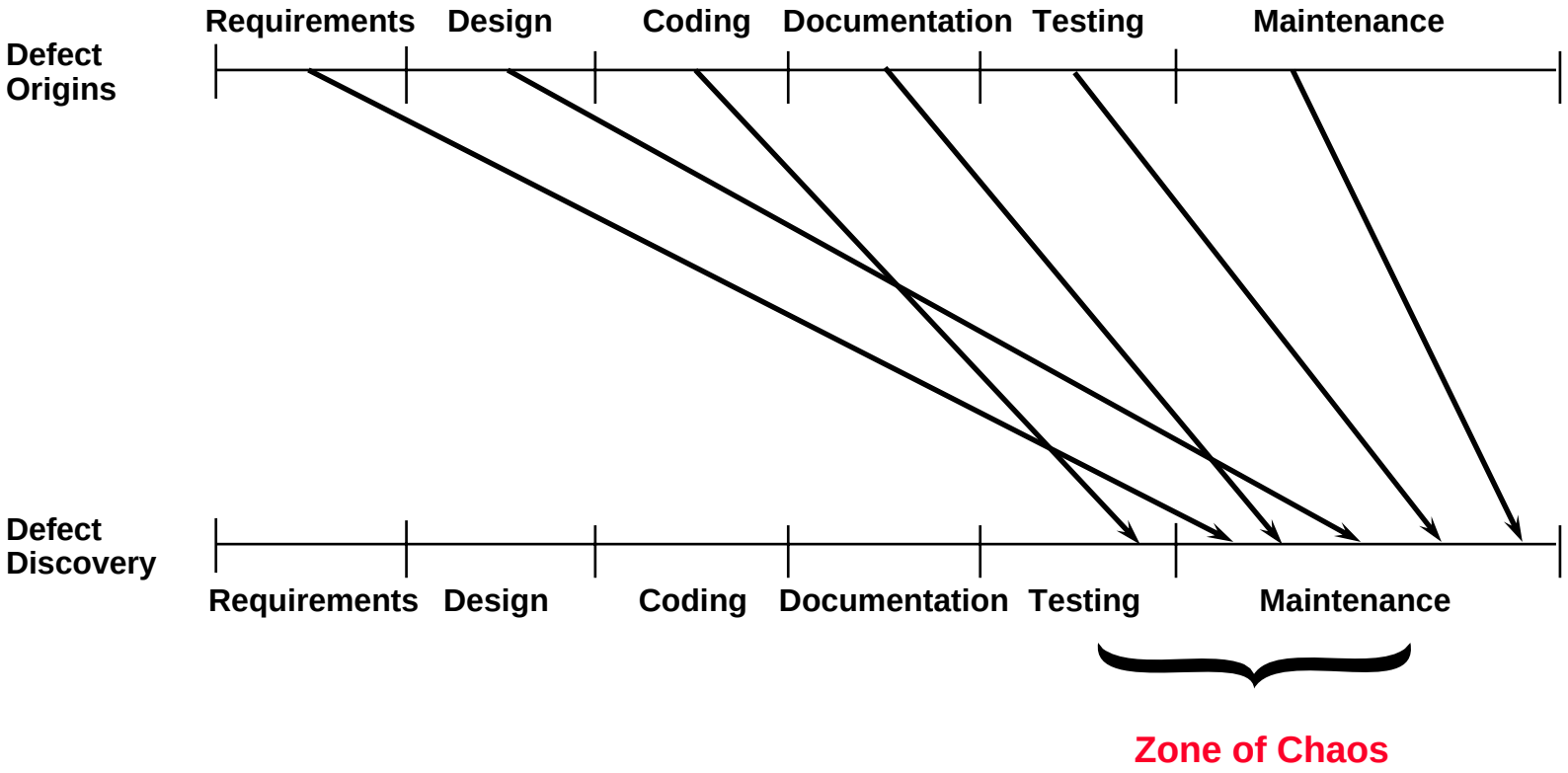
REMOVAL EFFICIENCY

Dev. (900) / Total (1000)	**=**	**90%**

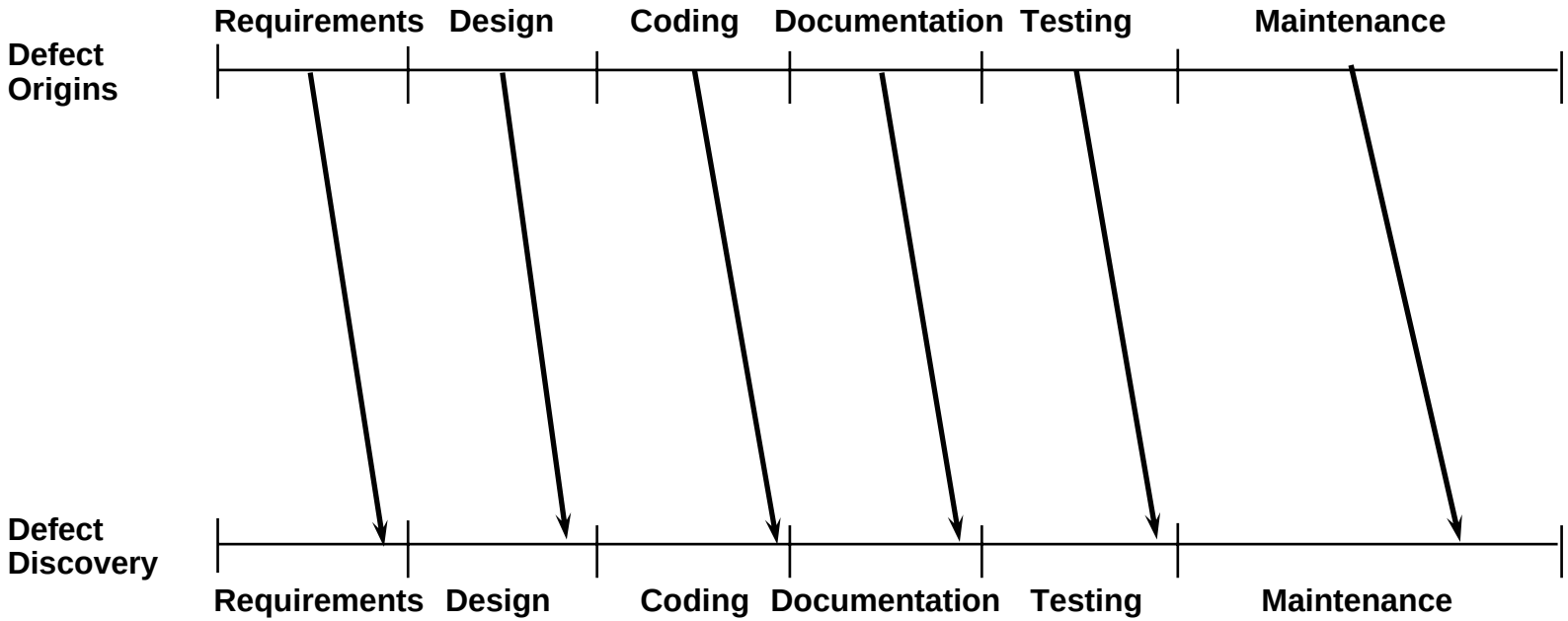
RANGES OF DEFECT REMOVAL EFFICIENCY

	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 inspections	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%

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 No Quality Assurance No Formal Testing	**30%**	**40%**	**50%**

SOFTWARE DEFECT REMOVAL RANGES (cont.)

SINGLE TECHNOLOGY CHANGES

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

SOFTWARE DEFECT REMOVAL RANGES (cont.)

TWO TECHNOLOGY CHANGES

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

SOFTWARE DEFECT REMOVAL RANGES (cont.)

TWO TECHNOLOGY CHANGES - continued

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

SOFTWARE DEFECT REMOVAL RANGES (cont.)

THREE TECHNOLOGY CHANGES

TECHNOLOGY COMBINATIONS	DEFECT REMOVAL EFFICIENCY		
	Lowest	Median	Highest
12. No design inspections FORMAL CODE INSPECTIONS FORMAL QUALITY ASSURANCE FORMAL TESTING	75%	87%	93%
13. FORMAL DESIGN INSPECTIONS No code inspections FORMAL QUALITY ASSURANCE FORMAL TESTING	77%	90%	95%
14. FORMAL DESIGN INSPECTIONS FORMAL CODE INSPECTIONS FORMAL QUALITY ASSURANCE No formal testing	83%	95%	97%
15. FORMAL DESIGN INSPECTIONS FORMAL CODE INSPECTIONS 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 **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 2008

SOFTWARE QUALITY TOPICS NEEDING RESEARCH:

Errors in software test plans and test cases

Errors in web content such as graphics and sound

Mass update costs and effectiveness

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

2008 QUALITY RESEARCH TOPICS

Quality levels of Agile projects

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 are most efficient**
- **Inspections + testing most often > 95% efficient.**
- **Defect prevention + removal best overall**
- **Quality excellence has ROI > $15 for each $1 spent**
- **High quality benefits both schedules and productivity**
- **High quality raises both customer and team morale!**

REFERENCES ON SOFTWARE QUALITY

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

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, 1996. (3rd edition due in the Spring of 2008)

Jones, Capers; Software Quality: Analysis and Guidelines for Success; International Thomson; 1995.

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.