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

<table>
<thead>
<tr>
<th>INDUSTRY</th>
<th>HAZARD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airlines</td>
<td>Safety hazards</td>
</tr>
<tr>
<td></td>
<td>Air traffic control problems</td>
</tr>
<tr>
<td></td>
<td>Flight schedule confusion</td>
</tr>
<tr>
<td></td>
<td>Navigation equipment failures</td>
</tr>
<tr>
<td></td>
<td>Maintenance schedules thrown off</td>
</tr>
<tr>
<td></td>
<td>Delay in opening Denver airport</td>
</tr>
<tr>
<td></td>
<td>Passengers booked into non-existent seats</td>
</tr>
<tr>
<td></td>
<td>Passengers misidentified as terror suspects</td>
</tr>
</tbody>
</table>
## SOFTWARE QUALITY HAZARDS IN TEN INDUSTRIES

<table>
<thead>
<tr>
<th>INDUSTRY</th>
<th>HAZARD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defense</td>
<td>Security hazards</td>
</tr>
</tbody>
</table>

- 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
<table>
<thead>
<tr>
<th>INDUSTRY</th>
<th>HAZARD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finance</td>
<td>Financial transaction hazards</td>
</tr>
<tr>
<td></td>
<td>Interest calculations in error</td>
</tr>
<tr>
<td></td>
<td>Account balances thrown off</td>
</tr>
<tr>
<td></td>
<td>Credit card charges in error</td>
</tr>
<tr>
<td></td>
<td>Funds transfer thrown off</td>
</tr>
<tr>
<td></td>
<td>Mortgage/loan interest payments in error</td>
</tr>
<tr>
<td></td>
<td>Hacking and identity theft due to software security flaws</td>
</tr>
<tr>
<td></td>
<td>Denial of service attacks due to software security flaws</td>
</tr>
</tbody>
</table>
## SOFTWARE QUALITY HAZARDS IN TEN INDUSTRIES

<table>
<thead>
<tr>
<th>INDUSTRY</th>
<th>HAZARD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Health Care</td>
<td>Safety hazards</td>
</tr>
<tr>
<td></td>
<td>Patient monitoring devices malfunction</td>
</tr>
<tr>
<td></td>
<td>Operating room schedules thrown off</td>
</tr>
<tr>
<td></td>
<td>Medical instruments malfunction</td>
</tr>
<tr>
<td></td>
<td>Prescription refill problems</td>
</tr>
<tr>
<td></td>
<td>Hazardous drug interactions</td>
</tr>
<tr>
<td></td>
<td>Billing problems</td>
</tr>
<tr>
<td></td>
<td>Medical records stolen or released by accident</td>
</tr>
<tr>
<td>INDUSTRY</td>
<td>HAZARD</td>
</tr>
<tr>
<td>----------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>Insurance</td>
<td>Liability, benefit hazards</td>
</tr>
<tr>
<td></td>
<td>Policy due dates in error</td>
</tr>
<tr>
<td></td>
<td>Policies cancelled in error</td>
</tr>
<tr>
<td></td>
<td>Benefits and interest calculation errors</td>
</tr>
<tr>
<td></td>
<td>Annuities miscalculated</td>
</tr>
<tr>
<td></td>
<td>Errors in actuarial studies</td>
</tr>
<tr>
<td></td>
<td>Payment records in error</td>
</tr>
</tbody>
</table>
## SOFTWARE QUALITY HAZARDS IN TEN INDUSTRIES

<table>
<thead>
<tr>
<th>INDUSTRY</th>
<th>HAZARD</th>
</tr>
</thead>
<tbody>
<tr>
<td>State, Local Governments</td>
<td>Local economic hazards</td>
</tr>
<tr>
<td></td>
<td>School taxes miscalculated</td>
</tr>
<tr>
<td></td>
<td>Jury records thrown off</td>
</tr>
<tr>
<td></td>
<td>Real-estate transactions misfiled</td>
</tr>
<tr>
<td></td>
<td>Divorce, marriage records misfiled</td>
</tr>
<tr>
<td></td>
<td>Alimony, child support payment records lost</td>
</tr>
<tr>
<td></td>
<td>Death records filed for wrong people</td>
</tr>
<tr>
<td></td>
<td>Traffic light synchronization thrown off</td>
</tr>
<tr>
<td></td>
<td>Errors in property tax assessments</td>
</tr>
<tr>
<td>INDUSTRY</td>
<td>HAZARD</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>Manufacturing</td>
<td>Operational hazards</td>
</tr>
<tr>
<td></td>
<td>Subcontract parts fail to arrive</td>
</tr>
<tr>
<td></td>
<td>Purchases of more or less than economic order quantities</td>
</tr>
<tr>
<td></td>
<td>Just-in-time arrivals thrown off</td>
</tr>
<tr>
<td></td>
<td>Assembly lines shut down</td>
</tr>
<tr>
<td></td>
<td>Aging errors for accounts receivable and cash flow</td>
</tr>
<tr>
<td></td>
<td>Aging errors for accounts payable and cash flow</td>
</tr>
<tr>
<td></td>
<td>Pension payments miscalculated</td>
</tr>
</tbody>
</table>
SOFTWARE QUALITY HAZARDS IN TEN INDUSTRIES

INDUSTRY

National Government

HAZARD

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
<table>
<thead>
<tr>
<th>INDUSTRY</th>
<th>HAZARD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public Utilities</td>
<td>Safety hazards</td>
</tr>
<tr>
<td></td>
<td>Electric meters malfunction</td>
</tr>
<tr>
<td></td>
<td>Gas meters malfunction</td>
</tr>
<tr>
<td></td>
<td>Distribution of electric power thrown off</td>
</tr>
<tr>
<td></td>
<td>Billing records in error</td>
</tr>
<tr>
<td></td>
<td>Nuclear power plants malfunction</td>
</tr>
<tr>
<td>INDUSTRY</td>
<td>HAZARD</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>Telecommunications</td>
<td>Service disruption</td>
</tr>
<tr>
<td></td>
<td>Hazards</td>
</tr>
<tr>
<td></td>
<td>Intercontinental switching disrupted</td>
</tr>
<tr>
<td></td>
<td>Domestic call switching disrupted</td>
</tr>
<tr>
<td></td>
<td>Billing records in error</td>
</tr>
</tbody>
</table>
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.
<table>
<thead>
<tr>
<th>BASIC DEFINITIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOFTWARE QUALITY</td>
</tr>
<tr>
<td>USER SATISFACTION</td>
</tr>
<tr>
<td>DEFECT PREVENTION</td>
</tr>
<tr>
<td>DEFECT REMOVAL</td>
</tr>
<tr>
<td>BAD FIXES</td>
</tr>
</tbody>
</table>
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
• 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)

<table>
<thead>
<tr>
<th>Defect Origins</th>
<th>Defect Potential</th>
<th>Removal Efficiency</th>
<th>Delivered Defects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>1.00</td>
<td>77%</td>
<td>0.23</td>
</tr>
<tr>
<td>Design</td>
<td>1.25</td>
<td>85%</td>
<td>0.19</td>
</tr>
<tr>
<td>Coding</td>
<td>1.75</td>
<td>95%</td>
<td>0.09</td>
</tr>
<tr>
<td>Documents</td>
<td>0.60</td>
<td>80%</td>
<td>0.12</td>
</tr>
<tr>
<td>Bad Fixes</td>
<td>0.40</td>
<td>70%</td>
<td>0.12</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>5.00</strong></td>
<td><strong>85%</strong></td>
<td><strong>0.75</strong></td>
</tr>
</tbody>
</table>

(Function points show all defect sources - not just coding defects)
### BEST IN CLASS SOFTWARE QUALITY

(Data expressed in terms of defects per function point)

<table>
<thead>
<tr>
<th>Defect Origins</th>
<th>Defect Potential</th>
<th>Removal Efficiency</th>
<th>Delivered Defects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>0.40</td>
<td>85%</td>
<td>0.08</td>
</tr>
<tr>
<td>Design</td>
<td>0.60</td>
<td>97%</td>
<td>0.02</td>
</tr>
<tr>
<td>Coding</td>
<td>1.00</td>
<td>99%</td>
<td>0.01</td>
</tr>
<tr>
<td>Documents</td>
<td>0.40</td>
<td>98%</td>
<td>0.01</td>
</tr>
<tr>
<td>Bad Fixes</td>
<td>0.10</td>
<td>95%</td>
<td>0.01</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>2.50</strong></td>
<td><strong>96%</strong></td>
<td><strong>0.13</strong></td>
</tr>
</tbody>
</table>

**OBSERVATIONS**

Most often found in systems software > SEI CMM Level 3
POOR SOFTWARE QUALITY - MALPRACTICE

(Data expressed in terms of defects per function point)

<table>
<thead>
<tr>
<th>Defect Origins</th>
<th>Defect Potential</th>
<th>Removal Efficiency</th>
<th>Delivered Defects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>1.50</td>
<td>50%</td>
<td>0.75</td>
</tr>
<tr>
<td>Design</td>
<td>2.20</td>
<td>50%</td>
<td>1.10</td>
</tr>
<tr>
<td>Coding</td>
<td>2.50</td>
<td>80%</td>
<td>0.50</td>
</tr>
<tr>
<td>Documents</td>
<td>1.00</td>
<td>70%</td>
<td>0.30</td>
</tr>
<tr>
<td>Bad Fixes</td>
<td>0.80</td>
<td>50%</td>
<td>0.40</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>8.00</strong></td>
<td><strong>62%</strong></td>
<td><strong>3.05</strong></td>
</tr>
</tbody>
</table>

OBSERVATIONS

Most often found in large client-server projects (> 5000 FP).
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 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 (CMMI)
- Virtualization for reuse and debugging
- Quality Estimation Tools
- Automated Test Support Tools + testing specialists
- Root-Cause Analysis
MIXED QUALITY RESULTS: < 50% SUCCESS RATE

- Quality Function Deployment (QFD)
- 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**

<table>
<thead>
<tr>
<th>Severity 1:</th>
<th>TOTAL FAILURE S</th>
<th>1% at release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Severity 2:</td>
<td>MAJOR PROBLEMS</td>
<td>20% at release</td>
</tr>
<tr>
<td>Severity 3:</td>
<td>MINOR PROBLEMS</td>
<td>35% at release</td>
</tr>
<tr>
<td>Severity 4:</td>
<td>COSMETIC ERRORS</td>
<td>44% at release</td>
</tr>
<tr>
<td>INVALID USER OR SYSTEM ERRORS</td>
<td>15% of reports</td>
<td></td>
</tr>
<tr>
<td>DUPLICATE</td>
<td>MULTIPLE REPORTS</td>
<td>30% of reports</td>
</tr>
<tr>
<td>ABEYANT</td>
<td>CAN’T RECREATE ERROR</td>
<td>5% of reports</td>
</tr>
</tbody>
</table>
Poor quality is cheaper until the end of the coding phase. After that, high quality is cheaper.
## U. S. SOFTWARE QUALITY AVERAGES CIRCA 2010

(Defects per Function Point)

<table>
<thead>
<tr>
<th></th>
<th>System Software</th>
<th>Commercial Software</th>
<th>Information Software</th>
<th>Military Software</th>
<th>Outsource Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defect Potentials</td>
<td>6.0</td>
<td>5.0</td>
<td>4.5</td>
<td>7.0</td>
<td>5.2</td>
</tr>
<tr>
<td>Defect Removal Efficiency</td>
<td>94%</td>
<td>90%</td>
<td>73%</td>
<td>96%</td>
<td>92%</td>
</tr>
<tr>
<td>Delivered Defects</td>
<td>0.4</td>
<td>0.5</td>
<td>1.2</td>
<td>0.3</td>
<td>0.4</td>
</tr>
<tr>
<td>First Year Discovery Rate</td>
<td>65%</td>
<td>70%</td>
<td>30%</td>
<td>75%</td>
<td>60%</td>
</tr>
<tr>
<td>First Year Reported Defects</td>
<td>0.26</td>
<td>0.35</td>
<td>0.36</td>
<td>0.23</td>
<td>0.30</td>
</tr>
</tbody>
</table>
## U. S. SOFTWARE QUALITY AVERAGES CIRCA 2010

(Defects per Function Point)

<table>
<thead>
<tr>
<th></th>
<th>Web Software</th>
<th>Embedded Software</th>
<th>SEI-CMM 3 Software</th>
<th>SEI-CMM 1 Software</th>
<th>Overall Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defect Potentials</td>
<td>4.0</td>
<td>5.5</td>
<td>3.0</td>
<td>5.5</td>
<td>5.1</td>
</tr>
<tr>
<td>Defect Removal</td>
<td>72%</td>
<td>95%</td>
<td>95%</td>
<td>73%</td>
<td>86.7%</td>
</tr>
<tr>
<td>Efficiency</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delivered Defects</td>
<td>1.1</td>
<td>0.3</td>
<td>0.15</td>
<td>1.5</td>
<td>0.68</td>
</tr>
<tr>
<td>First Year</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discovery Rate</td>
<td>95%</td>
<td>90%</td>
<td>60%</td>
<td>35%</td>
<td>64.4%</td>
</tr>
<tr>
<td>First Year</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reported Defects</td>
<td>1.0</td>
<td>0.27</td>
<td>0.09</td>
<td>0.52</td>
<td>0.43</td>
</tr>
</tbody>
</table>
## SOFTWARE SIZE VS DEFECT REMOVAL EFFICIENCY

(Data Expressed in terms of Defects per Function Point)

<table>
<thead>
<tr>
<th>Size</th>
<th>Defect Potential</th>
<th>Defect Removal Efficiency</th>
<th>Delivered Defects</th>
<th>1st Year Discovery Rate</th>
<th>1st Year Reported Defects</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.85</td>
<td>95.00%</td>
<td>0.09</td>
<td>90.00%</td>
<td>0.08</td>
</tr>
<tr>
<td>10</td>
<td>2.45</td>
<td>92.00%</td>
<td>0.20</td>
<td>80.00%</td>
<td>0.16</td>
</tr>
<tr>
<td>100</td>
<td>3.68</td>
<td>90.00%</td>
<td>0.37</td>
<td>70.00%</td>
<td>0.26</td>
</tr>
<tr>
<td>1000</td>
<td>5.00</td>
<td>85.00%</td>
<td>0.75</td>
<td>50.00%</td>
<td>0.38</td>
</tr>
<tr>
<td>10000</td>
<td>7.60</td>
<td>78.00%</td>
<td>1.67</td>
<td>40.00%</td>
<td>0.67</td>
</tr>
<tr>
<td>100000</td>
<td>9.55</td>
<td>75.00%</td>
<td>2.39</td>
<td>30.00%</td>
<td>0.72</td>
</tr>
</tbody>
</table>

**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)

<table>
<thead>
<tr>
<th>SEI CMM Levels</th>
<th>Defect Potentials</th>
<th>Removal Efficiency</th>
<th>Delivered Defects</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEI CMMI 1</td>
<td>5.00</td>
<td>80%</td>
<td>1.00</td>
</tr>
<tr>
<td>SEI CMMI 2</td>
<td>4.00</td>
<td>90%</td>
<td>0.40</td>
</tr>
<tr>
<td>SEI CMMI 3</td>
<td>3.00</td>
<td>95%</td>
<td>0.15</td>
</tr>
<tr>
<td>SEI CMMI 4</td>
<td>2.00</td>
<td>97%</td>
<td>0.08</td>
</tr>
<tr>
<td>SEI CMMI 5</td>
<td>1.00</td>
<td>99%</td>
<td>0.01</td>
</tr>
<tr>
<td>SEI CMMI 6 (TSP/PSP)</td>
<td>1.00</td>
<td>99.5%</td>
<td>&lt;0.01</td>
</tr>
</tbody>
</table>
## 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)

<table>
<thead>
<tr>
<th>SEI CMM Levels</th>
<th>Defect Potentials</th>
<th>Removal Efficiency</th>
<th>Delivered Defects</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEI CMMI 1</td>
<td>5.50</td>
<td>73%</td>
<td>1.48</td>
</tr>
<tr>
<td>SEI CMMI 2</td>
<td>4.00</td>
<td>90%</td>
<td>0.40</td>
</tr>
<tr>
<td>SEI CMMI 3</td>
<td>3.00</td>
<td>95%</td>
<td>0.15</td>
</tr>
<tr>
<td>SEI CMMI 4</td>
<td>2.50</td>
<td>97%</td>
<td>0.008</td>
</tr>
<tr>
<td>SEI CMMI 5</td>
<td>2.25</td>
<td>98%</td>
<td>0.005</td>
</tr>
<tr>
<td>SEI CMMI 6 (TSP/PSP)</td>
<td>2.00</td>
<td>99%</td>
<td>0.004</td>
</tr>
</tbody>
</table>
MAJOR SOFTWARE QUALITY ZONES

The SEI CMM has overlaps among the levels.

Defect Removal Efficiency

Malpractice

U.S. Average

SEI CMM 2
SEI CMM 3
SEI CMM 4
SEI CMM 5
TSP/PSP

Best in Class
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.

<table>
<thead>
<tr>
<th>IBM Corporation (MVS)</th>
<th>SPR Corporation (client studies)</th>
</tr>
</thead>
<tbody>
<tr>
<td>45% Design errors 25%</td>
<td>20% Requirements errors 30%</td>
</tr>
<tr>
<td>20% Coding errors 20%</td>
<td>35% Design errors 35%</td>
</tr>
<tr>
<td>20% Bad fixes 5%</td>
<td>35% Coding errors 10%</td>
</tr>
<tr>
<td>5% Documentation errors 5%</td>
<td>10% Bad fixes 5%</td>
</tr>
<tr>
<td>5% Administrative errors 5%</td>
<td>5% Documentation errors 5%</td>
</tr>
<tr>
<td>100% 100%</td>
<td>100% 100%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TRW Corporation</th>
<th>Mitre Corporation</th>
<th>Nippon Electric Corp.</th>
</tr>
</thead>
<tbody>
<tr>
<td>60% Design errors 40%</td>
<td>64% Design errors 36%</td>
<td>60% Design errors 40%</td>
</tr>
<tr>
<td>40% Coding errors 36%</td>
<td>36% Coding errors 40%</td>
<td>40% Coding errors 40%</td>
</tr>
<tr>
<td>100% 100%</td>
<td>100% 100%</td>
<td>100% 100%</td>
</tr>
</tbody>
</table>

Copyright © 2009 by Capers Jones. All Rights Reserved.
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
   - 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 17 kinds of test)
DEFECT REMOVAL EFFICIENCY

• Defect removal efficiency is a key quality measure

• Removal efficiency = \( \frac{\text{Defects found}}{\text{Defects present}} \)

• “Defects present” is the critical parameter
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

<table>
<thead>
<tr>
<th>Task</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inspections</td>
<td>500</td>
</tr>
<tr>
<td>Testing</td>
<td>400</td>
</tr>
<tr>
<td>Subtotal</td>
<td>900</td>
</tr>
</tbody>
</table>

USER-REPORTED DEFECTS IN FIRST 90 DAYS

<table>
<thead>
<tr>
<th>Category</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid unique defects</td>
<td>100</td>
</tr>
</tbody>
</table>

TOTAL DEFECT VOLUME

<table>
<thead>
<tr>
<th>Total</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defect totals</td>
<td>1000</td>
</tr>
</tbody>
</table>

REMOVAL EFFICIENCY

\[
\text{Dev. (900)} / \text{Total (1000)} = 90\%
\]
# Ranges of Defect Removal Efficiency

<table>
<thead>
<tr>
<th>Step</th>
<th>Lowest</th>
<th>Median</th>
<th>Highest</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Requirements review</td>
<td>20%</td>
<td>30%</td>
<td>50%</td>
</tr>
<tr>
<td>2 Top-level design reviews</td>
<td>30%</td>
<td>40%</td>
<td>60%</td>
</tr>
<tr>
<td>3 Detailed functional design reviews</td>
<td>30%</td>
<td>45%</td>
<td>65%</td>
</tr>
<tr>
<td>4 Detailed logic design reviews</td>
<td>35%</td>
<td>55%</td>
<td>75%</td>
</tr>
<tr>
<td>5 Code inspection or static analysis</td>
<td>35%</td>
<td>60%</td>
<td>85%</td>
</tr>
<tr>
<td>6 Unit tests</td>
<td>10%</td>
<td>25%</td>
<td>50%</td>
</tr>
<tr>
<td>7 New Function tests</td>
<td>20%</td>
<td>35%</td>
<td>55%</td>
</tr>
<tr>
<td>8 Integration tests</td>
<td>25%</td>
<td>45%</td>
<td>60%</td>
</tr>
<tr>
<td>9 System test</td>
<td>25%</td>
<td>50%</td>
<td>65%</td>
</tr>
<tr>
<td>10 External Beta tests</td>
<td>15%</td>
<td>40%</td>
<td>75%</td>
</tr>
</tbody>
</table>

**Cumulative Efficiency**

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>75%</td>
<td>97%</td>
<td>99.99%</td>
</tr>
</tbody>
</table>
NORMAL DEFECT ORIGIN/DISCOVERY GAPS

Defect Origins

- Requirements
- Design
- Coding
- Documentation
- Testing
- Maintenance

Defect Discovery

- Requirements
- Design
- Coding
- Documentation
- Testing
- Maintenance

Zone of Chaos
DEFECT ORIGINS/DISCOVERY WITH INSPECTIONS

Defect Origins

Requirements  Design  Coding  Documentation  Testing  Maintenance

Defect Discovery

Requirements  Design  Coding  Documentation  Testing  Maintenance
## SOFTWARE DEFECT REMOVAL RANGES

### WORST CASE RANGE

<table>
<thead>
<tr>
<th>TECHNOLOGY COMBINATIONS</th>
<th>DEFECT REMOVAL EFFICIENCY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. No Design Inspections</td>
<td>Lowest 30%</td>
</tr>
<tr>
<td>No Code Inspections or static analysis</td>
<td>Median 40%</td>
</tr>
<tr>
<td>No Quality Assurance</td>
<td>Highest 50%</td>
</tr>
<tr>
<td>No Formal Testing</td>
<td></td>
</tr>
</tbody>
</table>
## SOFTWARE DEFECT REMOVAL RANGES (cont.)

<table>
<thead>
<tr>
<th>TECHNOLOGY COMBINATIONS</th>
<th>DEFECT REMOVAL EFFICIENCY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lowest</td>
</tr>
<tr>
<td>2. No design inspections</td>
<td>32%</td>
</tr>
<tr>
<td>No code inspections or static analysis</td>
<td></td>
</tr>
<tr>
<td>FORMAL QUALITY ASSURANCE</td>
<td></td>
</tr>
<tr>
<td>No formal testing</td>
<td></td>
</tr>
<tr>
<td>3. No design inspections</td>
<td>37%</td>
</tr>
<tr>
<td>No code inspections or static analysis</td>
<td></td>
</tr>
<tr>
<td>No quality assurance</td>
<td></td>
</tr>
<tr>
<td>FORMAL TESTING</td>
<td></td>
</tr>
<tr>
<td>4. No design inspections</td>
<td>43%</td>
</tr>
<tr>
<td>CODE INSPECTIONS/STATIC ANALYSIS</td>
<td></td>
</tr>
<tr>
<td>No quality assurance</td>
<td></td>
</tr>
<tr>
<td>No formal testing</td>
<td></td>
</tr>
<tr>
<td>5. FORMAL DESIGN INSPECTIONS</td>
<td>45%</td>
</tr>
<tr>
<td>No code inspections or static analysis</td>
<td></td>
</tr>
<tr>
<td>No quality assurance</td>
<td></td>
</tr>
<tr>
<td>No formal testing</td>
<td></td>
</tr>
</tbody>
</table>
## SOFTWARE DEFECT REMOVAL RANGES (cont.)

### TWO TECHNOLOGY CHANGES

<table>
<thead>
<tr>
<th>TECHNOLOGY COMBINATIONS</th>
<th>DEFECT REMOVAL EFFICIENCY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lowest</td>
</tr>
<tr>
<td>6. No design inspections</td>
<td>50%</td>
</tr>
<tr>
<td>No code inspections or static analysis</td>
<td></td>
</tr>
<tr>
<td>FORMAL QUALITY ASSURANCE</td>
<td></td>
</tr>
<tr>
<td>FORMAL TESTING</td>
<td></td>
</tr>
<tr>
<td>7. No design inspections</td>
<td>53%</td>
</tr>
<tr>
<td>FORMAL CODE INSPECTIONS/STAT. AN.</td>
<td></td>
</tr>
<tr>
<td>FORMAL QUALITY ASSURANCE</td>
<td></td>
</tr>
<tr>
<td>No formal testing</td>
<td></td>
</tr>
<tr>
<td>8. No design inspections</td>
<td>55%</td>
</tr>
<tr>
<td>FORMAL CODE INSPECTIONS/STAT. AN.</td>
<td></td>
</tr>
<tr>
<td>No quality assurance</td>
<td></td>
</tr>
<tr>
<td>FORMAL TESTING</td>
<td></td>
</tr>
</tbody>
</table>
# SOFTWARE DEFECT REMOVAL RANGES (cont.)

## TWO TECHNOLOGY CHANGES - continued

<table>
<thead>
<tr>
<th>TECHNOLOGY COMBINATIONS</th>
<th>DEFECT REMOVAL EFFICIENCY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lowest</td>
</tr>
<tr>
<td>9. FORMAL DESIGN INSPECTIONS</td>
<td>60%</td>
</tr>
<tr>
<td>No code inspections or static analysis</td>
<td></td>
</tr>
<tr>
<td>FORMAL QUALITY ASSURANCE</td>
<td></td>
</tr>
<tr>
<td>No formal testing</td>
<td></td>
</tr>
<tr>
<td>10. FORMAL DESIGN INSPECTIONS</td>
<td>65%</td>
</tr>
<tr>
<td>No code inspections or static analysis</td>
<td></td>
</tr>
<tr>
<td>No quality assurance</td>
<td></td>
</tr>
<tr>
<td>FORMAL TESTING</td>
<td></td>
</tr>
<tr>
<td>11. FORMAL DESIGN INSPECTIONS</td>
<td>70%</td>
</tr>
<tr>
<td>FORMAL CODE INSPECTIONS/STAT.AN.</td>
<td></td>
</tr>
<tr>
<td>No quality assurance</td>
<td></td>
</tr>
<tr>
<td>No formal testing</td>
<td></td>
</tr>
</tbody>
</table>
## SOFTWARE DEFECT REMOVAL RANGES (cont.)

<table>
<thead>
<tr>
<th>THREE TECHNOLOGY CHANGES</th>
<th>TECHNOLOGY COMBINATIONS</th>
<th>DEFECT REMOVAL EFFICIENCY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Lowest</td>
</tr>
<tr>
<td>12. No design inspections</td>
<td>FORMAL CODE INSPECTIONS/STAT. AN.</td>
<td>75%</td>
</tr>
<tr>
<td></td>
<td>FORMAL QUALITY ASSURANCE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FORMAL TESTING</td>
<td></td>
</tr>
<tr>
<td>13. FORMAL DESIGN INSPECTIONS</td>
<td>77%</td>
<td>90%</td>
</tr>
<tr>
<td>No code inspections or static analysis</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FORMAL QUALITY ASSURANCE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FORMAL TESTING</td>
<td></td>
</tr>
<tr>
<td>14. FORMAL DESIGN INSPECTIONS</td>
<td>83%</td>
<td>95%</td>
</tr>
<tr>
<td></td>
<td>FORMAL CODE INSPECTIONS/STAT. AN.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FORMAL QUALITY ASSURANCE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>No formal testing</td>
<td></td>
</tr>
<tr>
<td>15. FORMAL DESIGN INSPECTIONS</td>
<td>85%</td>
<td>97%</td>
</tr>
<tr>
<td></td>
<td>FORMAL CODE INSPECTIONS/STAT. AN.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>No quality assurance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FORMAL TESTING</td>
<td></td>
</tr>
</tbody>
</table>
### SOFTWARE DEFECT REMOVAL RANGES (cont.)

#### BEST CASE RANGE

<table>
<thead>
<tr>
<th>TECHNOLOGY COMBINATIONS</th>
<th>DEFECT REMOVAL EFFICIENCY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. FORMAL DESIGN INSPECTIONS</td>
<td>Lowest</td>
</tr>
<tr>
<td>STATIC ANALYSIS</td>
<td>95%</td>
</tr>
<tr>
<td>FORMAL CODE INSPECTIONS</td>
<td></td>
</tr>
<tr>
<td>FORMAL QUALITY ASSURANCE</td>
<td></td>
</tr>
<tr>
<td>FORMAL TESTING</td>
<td></td>
</tr>
</tbody>
</table>
### DISTRIBUTION OF 1500 SOFTWARE PROJECTS BY DEFECT REMOVAL EFFICIENCY LEVEL

<table>
<thead>
<tr>
<th>Defect Removal Efficiency Level (Percent)</th>
<th>Number of Projects</th>
<th>Percent of Projects</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; 99</td>
<td>6</td>
<td>0.40%</td>
</tr>
<tr>
<td>95 - 99</td>
<td>104</td>
<td>6.93%</td>
</tr>
<tr>
<td>90 - 95</td>
<td>263</td>
<td>17.53%</td>
</tr>
<tr>
<td>85 - 90</td>
<td>559</td>
<td>37.26%</td>
</tr>
<tr>
<td>80 - 85</td>
<td>408</td>
<td>27.20%</td>
</tr>
<tr>
<td>&lt; 80</td>
<td>161</td>
<td>10.73%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>1,500</strong></td>
<td><strong>100.00%</strong></td>
</tr>
</tbody>
</table>
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

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


# REFERENCES ON SOFTWARE QUALITY

<table>
<thead>
<tr>
<th>Website</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.ASQ.org">www.ASQ.org</a></td>
<td>(American Society for Quality)</td>
</tr>
<tr>
<td><a href="http://www.ISBSG.org">www.ISBSG.org</a></td>
<td>(Int. Software Bench. Standards Group)</td>
</tr>
<tr>
<td><a href="http://www.ISO.org">www.ISO.org</a></td>
<td>(International Organization for Standards)</td>
</tr>
<tr>
<td><a href="http://www.PMI.org">www.PMI.org</a></td>
<td>(Project Management Institute)</td>
</tr>
<tr>
<td><a href="http://www.SEI.org">www.SEI.org</a></td>
<td>(Software Engineering Institute)</td>
</tr>
<tr>
<td><a href="http://www.SPR.com">www.SPR.com</a></td>
<td>(Software Productivity Research LLC)</td>
</tr>
<tr>
<td><a href="http://www.SSQ.org">www.SSQ.org</a></td>
<td>(Society for Software Quality)</td>
</tr>
<tr>
<td><a href="http://www.semat.org">www.semat.org</a></td>
<td>(Software Engineering Methods and Tools)</td>
</tr>
<tr>
<td><a href="http://www.cisq.org">www.cisq.org</a></td>
<td>(Consortium for IT software quality)</td>
</tr>
</tbody>
</table>