**Case Example 2:**

## Software Quality Control

Software quality control is a weak line in software engineering. Due to poor metrics choices and poor measurement practices, very few people have reliable data on effective quality control techniques.

Example 2 shows accurate quantified data on the effectiveness of defect prevention, pre-test defect removal such as inspections and static analysis, and the effectiveness of common forms of testing.

The goal of effective software quality control is to have defect potentials below 3.00 per function point combined with defect removal efficiency (DRE) above 99%. The current U.S. average is a defect potential of about 4.25 bugs per function point and only 92.50% DRE

| | | | | |
|---|---|---|---|---|
| **Example 2: How Software Risk Master (SRM) Evaluates Software Quality Results** | | | | |
| | | | | |
| | | | | |
| | **Java Language for all 3 Cases** | | | |
| | **1000 function points for all 3 Cases** | | | |
| | **$10,000 per month for all 3 Cases** | | | |
| | **Iterative development for all 3 Cases** | | | |
| | **132 effective work hours per month for all 3 Cases** | | | |
| | **Note: example uses round numbers for clarity.** | | | |
| | **Function points, defect potentials, and removal efficiency come from IBM** | | | |
| | **2017 is the 30th anniversary of IFPUG function point metrics** | | | |
| | | | | |
| | | | | |
| | | | | |
| | **Poor Quality Control** | **Average Quality Control** | **High Quality Control** | |
| | | | | |
| **Team Experience** | **Novice** | **Average** | **Expert** | **Expert teams are best in software quality control** |
| | | | | |
| **Defect potential per FP** | | | | |
| Requirements defects | 1.00 | 0.75 | 0.25 | **Defect potentials includes all defect sources** |
| Design defects | 1.50 | 1.00 | 0.50 | |
| Code defects | 2.25 | 1.15 | 0.75 | |
| Document defects | 0.80 | 0.60 | 0.40 | |
| Bad fixes | 0.70 | 0.40 | 0.10 | **Bad fix = new bugs in bug repairs** |
| **TOTAL DEFECTS** | **6.25** | **3.90** | **2.00** | **Expert teams have low defect potentials** |
| | | | | |
| **Defect potentials** | | | | |
| **Requirements defe** | **1,000** | **750** | **250** | |
| **Design defects** | **1,500** | **1,000** | **500** | |

| | | | | |
|---|---|---|---|---|
| **Code defects** | **2,250** | **1,150** | **750** | **Code defects are < 30% of total defects** |
| **Document defects** | **800** | **600** | **400** | |
| **Bad fixes** | **700** | **400** | **100** | |
| **TOTAL DEFECT PO** | **6,250** | **3,900** | **2,000** | **Expert teams have low defect potentials** |
| | | | | |
| **Defect Prevention Efficiency** | | | | |
| JAD | 0.00% | 22.50% | 26.00% | **JAD = Joint Application Design** |
| QFD | 0.00% | 0.00% | 28.00% | **QFD = Quality Function Deployment** |
| Prototype | 20.00% | 20.00% | 25.00% | |
| Models | 0.00% | 0.00% | 62.00% | **Models are cost effective and efficient** |
| **TOTAL** | **20.00%** | **37.02%** | **81.19%** | **Defect prevention can eliminate many bugs** |
| **Defects remaining** | **4,835** | **2,456** | **312** | **Defect prevention is cost effective and efficient** |
| | | | | |
| **Pre-Test Removal Efficiency** | | | | |
| Desk check | 20.00% | 25.00% | 26.00% | **Pre-test removal is key to good quality control** |
| Pair programming | 0.00% | 0.00% | 0.00% | **Pair programming is expensive and inefficient** |
| Static analysis | 0.00% | 55.00% | 59.00% | **Static analysis is cost effective and efficient** |
| Inspections | 0.00% | 0.00% | 85.00% | **Inspections are cost effective and efficient** |
| **TOTAL** | **20.00%** | **62.01%** | **94.81%** | **High quality removes < 90% of bugs before test** |
| **Defects remaining** | **4,081** | **933** | **15** | **Pre-test removal raises test efficiency too** |
| | | | | |
| **Test Removal Efficiency** | | | | |

| | | | | |
|---|---|---|---|---|
| Unit test | 27.50% | 30.00% | 32.40% | **Up to 18 different kinds of testing are known; most projects use only 7** |
| Function test | 30.50% | 33.00% | 35.64% | |
| Regression test | 9.50% | 12.00% | 12.96% | |
| Component test | 27.50% | 30.00% | 32.40% | |
| Performance test | 7.50% | 10.00% | 10.80% | |
| System test | 31.50% | 34.00% | 36.72% | |
| Acceptance test | 12.50% | 15.00% | 16.20% | |
| **TOTAL** | **76.80%** | **81.60%** | **88.13%** | **Most forms of testing are < 35% removal efficiency** |
| **Defects remaining** | **972** | **177** | **8** | |
| | | | | |
| **CUMULATIVE EFF** | **83.77%** | **95.46%** | **99.81%** | **High quality > 99% defect removal efficiency (DRE)** |
| | | | | |
| **DELIVERED DEFE** | **972** | **177** | **8** | **High quality has few delivered defects** |
| | | | | |
| **HIGH-SEVERITY I** | **107** | **11** | **0** | **High quality has few serious bugs** |
| | | | | |
| **SECURITY FLAWS** | **15** | **3** | **0** | **High quality has few security flaws** |
| | | | | |
| **DEFECT REMOVAI** | **$3,595,000** | **$1,035,750** | **$784,284** | **High quality is cheaper than poor quality** |
| | | | | |
| **Delivered defects p** | **0.97** | **0.18** | **0.01** | **Function points are best metric for quality data** |
| | | | | |
| **Defect removal $ p** | **$3,595.00** | **$1,035.75** | **$784.28** | **Function points are best metric for quality costs** |
| | | | | |

| | | | | |
|---|---|---|---|---|
| % of high-severity c | 764.29% | 100.00% | 0.00% | Low severe bugs are #1 sign of high quality |
| | | | | |
| % of delivered secu | 595.24% | 100.00% | 0.00% | Low security flaws are #2 sign of high quality |
| | | | | |
| % of average delive | 549.15% | 100.00% | 4.52% | Low delivered defects are #3 sign of high quality |
| | | | | |
| % of average defect | 160.26% | 100.00% | 51.28% | Low defect potentials are #4 sign of high quality |
| | | | | |
| % of average cost o | 347.09% | 100.00% | 75.72% | Low COQ is #5 sign of high quality |
| | | | | |
| % of average remov | 87.75% | 100.00% | 104.56% | High removal efficiency is #6 sign of high quality |
| | | | | |
| | | | | |
| | | END OF EXAMPLE | | |