Test Metrics:
A Practical Approach to Tracking & Interpretation

Presented By:
Shaun Bradshaw
Questcon Technologies
sbradshaw@questcon.com
Objectives

- Why Measure?
- Definition
- Metrics Philosophy
- Types of Metrics
- Interpreting the Results
- Metrics Case Study
- Q & A
“Software bugs cost the U.S. economy an estimated $59.5 billion per year. An estimated $22.2 billion could be eliminated by improved testing that enables earlier and more effective identification and removal of defects.”

- US Department of Commerce (NIST)
It is often said, “You cannot improve what you cannot measure.”
Test Metrics:

- Are a standard of measurement.
- Gauge the effectiveness and efficiency of several software development activities.
- Are gathered and interpreted throughout the test effort.
- Provide an objective measurement of the success of a software project.
When tracked and used properly, test metrics can aid in software development process improvement by providing pragmatic & objective evidence of process change initiatives.
Keep It Simple

- Measure the basics first
- Clearly define each metric
- Get the most “bang for your buck”
Metrics are useless if they are meaningless (use GQM model)

- Must be able to interpret the results
- Metrics interpretation should be objective
Incorporate metrics tracking into the Run Log or defect tracking system

Automate tracking process to remove time burdens

Accumulate throughout the test effort & across multiple projects
Metrics Philosophy

- Interpret the results
- Provide feedback to the Project Team
- Implement changes based on objective data

Keep It Simple

Make It Meaningful

Track It

Use It
### Base Metrics
- Raw data gathered by Test Analysts
- Tracked throughout test effort
- Used to provide project status and evaluations/feedback

### Examples
- # Test Cases
- # Executed
- # Passed
- # Failed
- # Under Investigation
- # Blocked
- # 1st Run Failures
- # Re-Executed
- Total Executions
- Total Passes
- Total Failures
## Types of Metrics

### Base Metrics
- Raw data gathered by Test Analyst
- Tracked throughout test effort
- Used to provide project status and evaluations/feedback

### # Blocked
- The number of distinct test cases that cannot be executed during the test effort due to an application or environmental constraint.
- Defines the impact of known system defects on the ability to execute specific test cases

### Examples
- # Test Cases
- # Executed
- # Passed
- # Failed
- # Under Investigation
- # Blocked
- # 1st Run Failures
- # Re-Executed
- Total Executions
- Total Passes
- Total Failures
**Types of Metrics**

**Calculated Metrics**
- Tracked by Test Lead/Manager
- Converts base metrics to useful data
- Combinations of metrics can be used to evaluate process changes

**Examples**
- % Complete
- % Test Coverage
- % Test Cases Passed
- % Test Cases Blocked
- 1st Run Fail Rate
- Overall Fail Rate
- % Defects Corrected
- % Rework
- % Test Effectiveness
- Defect Discovery Rate
**Types of Metrics**

### Calculated Metrics
- Tracked by Test Lead/Manager
- Converts base metrics to useful data
- Combinations of metrics can be used to evaluate process changes

### Examples
- % Complete
- % Test Coverage
- % Test Cases Passed
- % Test Cases Blocked
- **1st Run Fail Rate**
- Overall Fail Rate
- % Defects Corrected
- % Rework
- % Test Effectiveness
- Defect Discovery Rate

**1st Run Fail Rate**
- The percentage of executed test cases that failed on their first execution.
- Used to determine the effectiveness of the analysis and development process. Comparing this metric across projects shows how process changes have impacted the quality of the product at the end of the development phase.
## Sample System Test

<table>
<thead>
<tr>
<th>TC ID</th>
<th>Run Date</th>
<th>Actual Results</th>
<th>Run Status</th>
<th>Current Status</th>
<th># of Runs</th>
</tr>
</thead>
<tbody>
<tr>
<td>SST-001</td>
<td>01/01/04</td>
<td>Actual results met expected results.</td>
<td>P</td>
<td>P</td>
<td>1</td>
</tr>
<tr>
<td>SST-002</td>
<td>01/01/04</td>
<td>Sample failure</td>
<td>F</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>SST-003</td>
<td>01/02/04</td>
<td>Sample multiple failures</td>
<td>F</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>SST-004</td>
<td>01/02/04</td>
<td>Actual results met expected results.</td>
<td>P</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>SST-005</td>
<td>01/02/04</td>
<td>Actual results met expected results.</td>
<td>P</td>
<td>P</td>
<td>1</td>
</tr>
<tr>
<td>SST-006</td>
<td>01/03/04</td>
<td>Sample Under Investigation</td>
<td>U</td>
<td>U</td>
<td>1</td>
</tr>
<tr>
<td>SST-007</td>
<td>01/03/04</td>
<td>Actual results met expected results.</td>
<td>P</td>
<td>P</td>
<td>1</td>
</tr>
<tr>
<td>SST-008</td>
<td>01/03/04</td>
<td>Sample Blocked</td>
<td>B</td>
<td>B</td>
<td>0</td>
</tr>
<tr>
<td>SST-009</td>
<td>01/03/04</td>
<td>Sample Blocked</td>
<td>B</td>
<td>B</td>
<td>0</td>
</tr>
<tr>
<td>SST-010</td>
<td>01/03/04</td>
<td>Actual results met expected results.</td>
<td>P</td>
<td>P</td>
<td>1</td>
</tr>
<tr>
<td>SST-011</td>
<td>01/03/04</td>
<td>Actual results met expected results.</td>
<td>P</td>
<td>P</td>
<td>1</td>
</tr>
<tr>
<td>SST-012</td>
<td>01/03/04</td>
<td>Actual results met expected results.</td>
<td>P</td>
<td>P</td>
<td>1</td>
</tr>
<tr>
<td>SST-013</td>
<td>01/03/04</td>
<td>Actual results met expected results.</td>
<td>P</td>
<td>P</td>
<td>1</td>
</tr>
<tr>
<td>SST-014</td>
<td>01/03/04</td>
<td>Actual results met expected results.</td>
<td>P</td>
<td>P</td>
<td>1</td>
</tr>
<tr>
<td>SST-015</td>
<td>01/03/04</td>
<td>Actual results met expected results.</td>
<td>P</td>
<td>P</td>
<td>1</td>
</tr>
</tbody>
</table>
## Sample Run Log

### Base Metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total # of TCs</td>
<td>100</td>
</tr>
<tr>
<td># Executed</td>
<td>13</td>
</tr>
<tr>
<td># Passed</td>
<td>11</td>
</tr>
<tr>
<td># Failed</td>
<td>1</td>
</tr>
<tr>
<td># UI</td>
<td>1</td>
</tr>
<tr>
<td># Blocked</td>
<td>2</td>
</tr>
<tr>
<td># Unexecuted</td>
<td>87</td>
</tr>
<tr>
<td># Re-executed</td>
<td>1</td>
</tr>
<tr>
<td>Total Executions</td>
<td>15</td>
</tr>
<tr>
<td>Total Passes</td>
<td>11</td>
</tr>
<tr>
<td>Total Failures</td>
<td>3</td>
</tr>
<tr>
<td>1st Run Failures</td>
<td>2</td>
</tr>
</tbody>
</table>

### Calculated Metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>% Complete</td>
<td>11.0%</td>
</tr>
<tr>
<td>% Test Coverage</td>
<td>13.0%</td>
</tr>
<tr>
<td>% TCs Passed</td>
<td>84.6%</td>
</tr>
<tr>
<td>% TCs Blocked</td>
<td>2.0%</td>
</tr>
<tr>
<td>% 1st Run Failures</td>
<td>15.4%</td>
</tr>
<tr>
<td>% Failures</td>
<td>20.0%</td>
</tr>
<tr>
<td>% Defects Corrected</td>
<td>66.7%</td>
</tr>
<tr>
<td>% Rework</td>
<td>100.0%</td>
</tr>
</tbody>
</table>
**Issue:**
The test team tracks and reports various test metrics, but there is no effort to analyze the data.

**Result:**
Potential improvements are not implemented leaving process gaps throughout the SDLC. This reduces the effectiveness of the project team and the quality of the applications.
Solution:
- Closely examine all available data
- Use the objective information to determine the root cause
- Compare to other projects
  - Are the current metrics typical of software projects in your organization?
  - What effect do changes have on the software development process?

Result:
Future projects benefit from a more effective and efficient application development process.
Volvo IT of North America had little or no testing involvement in its IT projects. The organization’s projects were primarily maintenance related and operated in a COBOL/CICS/Mainframe environment. The organization had a desire to migrate to newer technologies and felt that testing involvement would assure and enhance this technological shift.

While establishing a test team we also instituted a metrics program to track the benefits of having a QA group.
Project V

- Introduced a test methodology and metrics program
- Project was 75% complete (development was nearly finished)
- Test team developed 355 test scenarios
- 30.7% - 1st Run Fail Rate
- 31.4% - Overall Fail Rate
- Defect Repair Costs = $519,000
Project T

- Instituted requirements walkthroughs and design reviews with test team input
- Same resources comprised both project teams
- Test team developed 345 test scenarios
- 17.9% - 1st Run Fail Rate
- 18.0% - Overall Fail Rate
- Defect Repair Costs = $346,000
### Metrics Case Study

#### Reduction of 1st Run Fail Rate

<table>
<thead>
<tr>
<th></th>
<th>Project V</th>
<th>Project T</th>
<th>Reduction of</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st Run Fail Rate</td>
<td>30.7%</td>
<td>17.9%</td>
<td>41.7%</td>
</tr>
<tr>
<td>Overall Failure Rate</td>
<td>31.4%</td>
<td>18.0%</td>
<td>42.7%</td>
</tr>
<tr>
<td>Cost of Defects</td>
<td>$519,000.00</td>
<td>$346,000.00</td>
<td>$173,000.00</td>
</tr>
</tbody>
</table>

Reduction of **33.3%** in the cost of defect repairs

Every project moving forward, using the same QA principles can achieve the same type of savings.
Q & A

Questions & Answers

No one tries to fail.