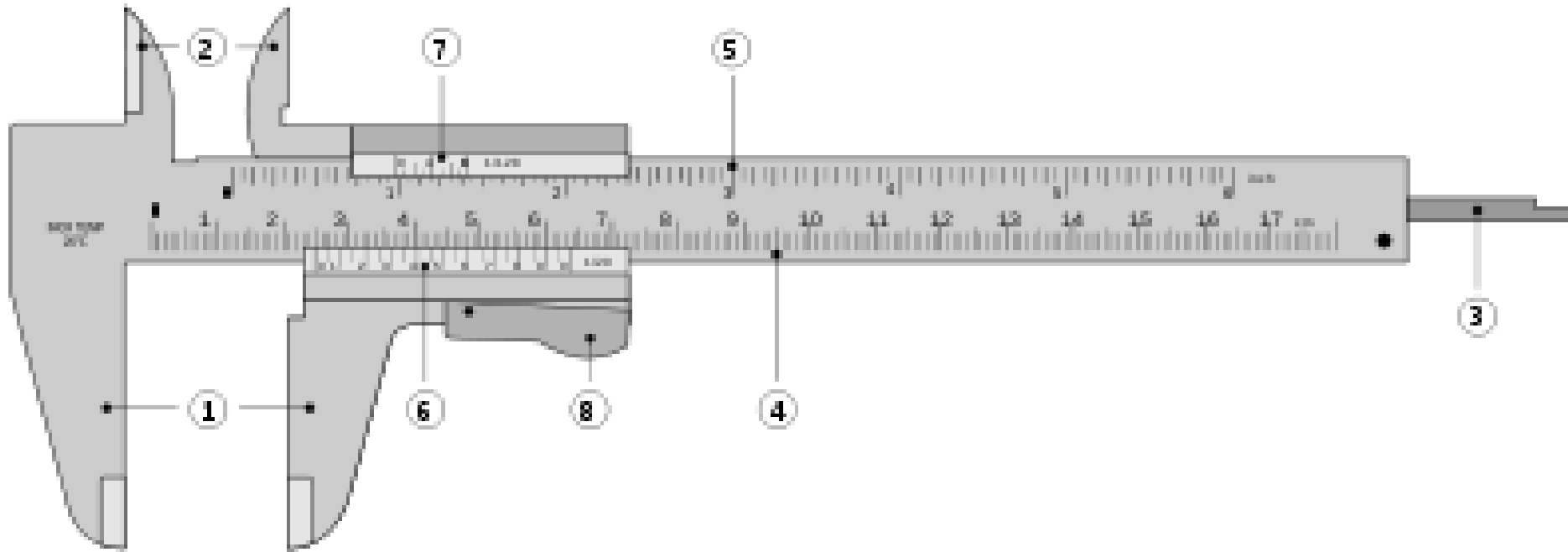


Metrics for Software Testing

Measuring Project, Process, and Product



RBCS

**TIME TESTED.
TESTING IMPROVED.**

www.RBCS-US.com



Metrics for Testing: Managing with Facts

- ❖ By using metrics, we can manage with facts and reality
 - ❖ How can we use metrics to manage testing?
 - ❖ What metrics can we use to measure the test process?
 - ❖ What metrics can we use to measure our progress in testing a project?
 - ❖ What do metrics tell us about the quality of the product?
- ❖ Let's see if we can find some answers...



Why Metrics?

- ❖ Managing with metrics – for testing or anything – allows us to manage with facts
- ❖ Subjective, uninformed opinions are not a sound basis for management
- ❖ What sounds reasonable can be wrong (e.g., Aristotle)



Why Testing Metrics?

- ❖ Testing by itself has no value...but it produces potentially valuable information
- ❖ Test information must be generated and communicated effectively to be valuable
- ❖ Common communication goals
 - ❖ Notify (“We have 24 bugs remaining to close”)
 - ❖ Enlighten (“See the time lost due to reopens”)
 - ❖ Influence (“We propose a bug triage meeting”)
- ❖ The metrics should be used as part of regular status reports (dashboards) as well as periodic analyses



How to Develop Metrics?

- ✚ Identify objectives
- ✚ Create questions about effectiveness, efficiency, and elegance of achievement of objectives
- ✚ Devise a measurable metric (or find a surrogate metric) to address the question
- ✚ Set a goal for each metric
- ✚ Implement improvements to improve goal
- ✚ Let's look at types of metrics and examples...



Example

- ❖ Objective: Find defects
- ❖ Effectiveness question: Have we finished finding new defects?
- ❖ Metric: Track trend in defect discovery during testing
- ❖ Goal: Watch for convergence (more later)
- ❖ Improvement: Find bugs earlier



Example

- ✦ Objective: Build confidence
- ✦ Note that we need a surrogate metric for confidence; one way is to measure coverage
- ✦ Coverage has multiple dimensions
- ✦ Effectiveness question: Do any requirements have identified failures?
- ✦ Metric: What % of requirements are completely tested without any failures, what % requirements have failures, what % are untested
- ✦ Goal: 100% requirements tested without must-fix failures
- ✦ Improvement: Reduce % of requirements that fail in testing



Good Metrics

- ❖ Define a useful, pertinent, and concise set of quality and test metrics
- ❖ Avoid too large a set of metrics
 - ❑ Too difficult to measure later
 - ❑ Confusing to participants
- ❖ Ensure uniform, agreed interpretations of these metrics, to minimize disputes and divergent opinions about measures of outcomes, analyses, and trends



Tracking Metrics

- ⊕ Automated tool support can reduce the time required to capture, track, analyze, report, and measure metrics
- ⊕ Apply objective and subjective analysis for specific metrics over time, especially when trends emerge that could allow for multiple interpretations of meaning
- ⊕ Be aware of and manage the tendency for people's interests to affect the interpretation they place on a particular metric or measure



Reporting Metrics and Measurements

- ❖ Reporting of metrics and measures should enlighten management and other stakeholders, not confuse or misdirect them
- ❖ Good testing reports based on metrics should be easily understood, not overly complex or ambiguous
- ❖ Good testing reports based on metrics should help management improve processes, guide the project to success, and manage product quality
- ❖ Metrics include snapshots of status at a moment in time, trends emerging over time, and analysis of causes and relationships between factors that influence testing and quality outcomes



Process Metrics for Testing

- ✚ Uses
 - ✚ To measure the effectiveness, efficiency, and elegance of the test process
 - ✚ To provide the insights to guide where process improvements should occur
- ✚ Part of certain test process assessments (e.g., using CTP framework)
- ✚ Not used as frequently as should be the case
- ✚ Not typically part of any dashboard (but can be)



Developing Good Process Metrics

- ❖ Determine key objectives for the test process
- ❖ Typical objectives
 - ❖ Finding bugs, especially important bugs
 - ❖ Reducing quality risk
- ❖ Decide which important effectiveness, efficiency, and elegance questions apply
- ❖ Create metrics that address those questions
- ❖ Set goals that reflect current or desired capability



Bug Finding Effectiveness

- ⊕ Objective: Finding bugs
- ⊕ Question: What percentage of bugs found?
- ⊕ Metric: Defect detection percentage
- ⊕ Goal: A typical number is 85%, though some teams do much better

$$DDP = \frac{\text{bugs detected}}{\text{bugs present}}$$

$$DDP (\text{for testing}) = \frac{\text{test bugs}}{\text{test bugs} + \text{production bugs}}$$



Bug Finding Focus

- ❖ Objective: Finding important bugs
- ❖ Question: Do we find more important bugs than less important bugs?
- ❖ Metric: Relation of DDPs (see below)
- ❖ Goal: Relationship shown below is always true

$$DDP(\textit{all bugs}) < DDP(\textit{critical bugs})$$



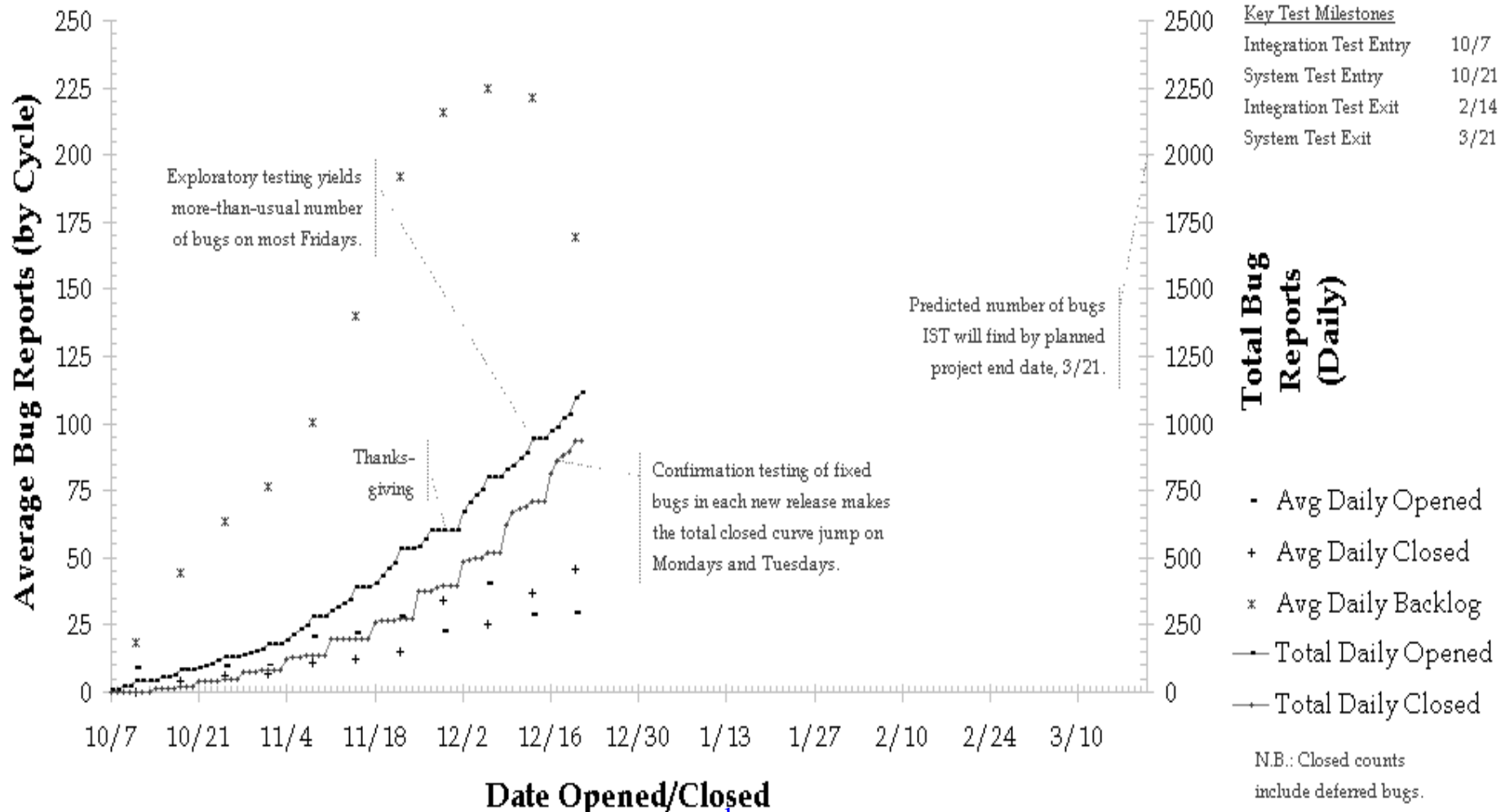
Project Metrics for Testing

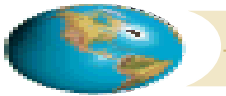
✚ Uses

- ✚ To measure the progress of the testing effort on a project
- ✚ To provide the insights to guide where project optimizations might be needed/useful
- ✚ Part of certain test process assessments (e.g., using CTP framework)
- ✚ The most commonly used test metrics
- ✚ Usually included in test dashboards (often over-relied upon)



Sumatra Integration and System Test Execution Phases System Quality Problems Analysis



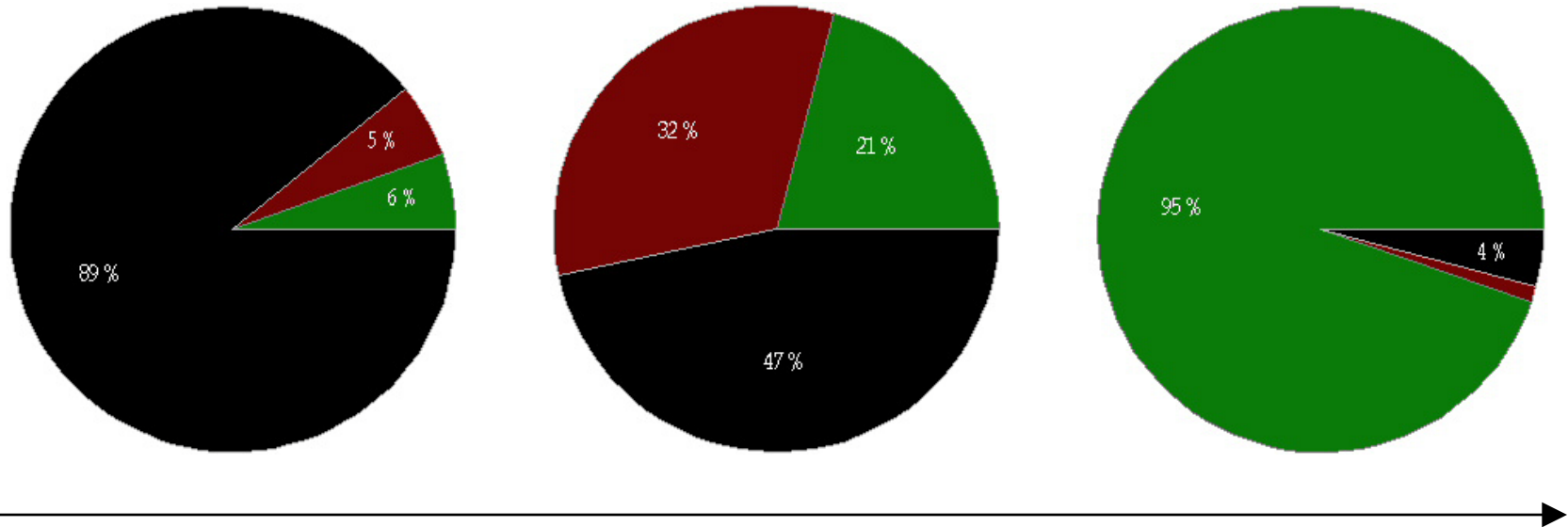


Product Metrics for Testing

- ✚ Uses
 - ✚ To measure the quality and residual quality risk in the product
 - ✚ To provide the insights to guide where product improvements should occur
- ✚ Good testing should measure the quality of the system
- ✚ Not always measured during testing (using project metrics instead)
- ✚ Not always part of testing dashboards (but should be)



Example: Reporting Residual Levels of Risk



Test execution period

Region in green represents risks for which all tests were run and passed and no must-fix bugs were found. Region in red represents risks for which at least one test has failed and at least one must-fix bug is known. Region in black represents other risks, which have no known must-fix bugs but still have tests pending to run.



Conclusions

- ⊕ Metrics can be useful to measure, understand, and manage process, project, and product attributes
- ⊕ Testing metrics are essential for a complete understanding of status
- ⊕ Use a small number of metrics
- ⊕ Use surrogate metrics where necessary
- ⊕ Ensure consistent stakeholder understanding of metrics
- ⊕ Define objectives, then questions, then metrics, then goals
- ⊕ Use the goals to drive meaningful process improvements



To Contact RBCS

For over a dozen years, RBCS has delivered services in consulting, outsourcing and training for software and hardware testing. Employing the industry's most experienced and recognized consultants, RBCS conducts product testing, builds and improves testing groups and hires testing staff for hundreds of clients worldwide. Ranging from Fortune 20 companies to start-ups, RBCS clients save time and money through improved product development, decreased tech support calls, improved corporate reputation and more. To learn more about RBCS, visit www.rbc-us.com.

Address: RBCS, Inc.
31520 Beck Road
Bulverde, TX 78163-3911
USA

Phone: +1 (830) 438-4830
Fax: +1 (830) 438-4831
E-mail: info@rbc-us.com
Web: www.rbc-us.com