Presentation
Paper
Bios
Return to Main Menu

PRESENTATION

F3

Friday, May 5, 2000
10:15AM

TEST METRICS WITHOUT TEARS

David Vaughan and Joe Elledge
The SIM Group

International Conference On
Software Testing Analysis & Review
May 1-5, 2000
Orlando. FL. USA

TEST METRICS WITHOUT TEARS

David Vaughan

&

Joe Elledge

©2000, Systems Integration Management Ltd.

Background

Year 2000 remediation, and, in Europe the Euro, have necessitated massive & unprecedented testing effort. This has led to:

- Testing gaining greater exposure and having a much higher profile
- Jan 2 2000: *“Did we need to do so much Y2k work?”*
- More than ever before, Testers must justify their existence

After Year 2000
What's left to test?

- Postponed new software development
- Re-engineering existing applications & making enhancements to them
- Multimedia & convergence applications

(*how do you test an interactive television?)*

- More package applications to be implemented
- Systems software & middleware upgrades
- New e-commerce applications & lots of them

©2000, Systems Integration Management Ltd.

What is an acceptable level of quality?

- It is determined by the risks to the organisation if the software fails.
- Metrics can help in risk assessment by providing the costs of testing
- To be weighed against the potential costs of not testing
- People risks should be averted not costed

We test too much!

- Many people hold the belief that:
  - testers do far too much testing
  - the bugs we identify are minor problems & typing errors
- We should not scare-monger – we should test commensurate with the risk
- The message is “Make no mistake. Software testing is critical to an organisation’s success”

©2000, Systems Integration Management Ltd.

Q:How can we prove that we are giving a good service?

A: METRICS

- It is not sufficient to produce software defect metrics – like how many raised, how many outstanding/ fixed by severity & urgency etc.
- We must prove that the activities of the test team are a cost-effective & efficient use of staff

Test Preparation statistics

reported by Test Phase or Development Stage

- No of test conditions identified & documented, by criticality/priority
- How many tests are required to exercise these test conditions
- Total number of tests to be built for Manual & Automated execution

Test Preparation statistics

- Number and or percentage of tests to be automated
- Test Environment preparation details including
  - the quantities of Referential & Parameter data required to be set up
  - Test Environment management procedures & utilities to be developed

©2000, Systems Integration Management Ltd.

Test Execution Statistics

- How long it takes to execute both manual and automated tests
- How many tests have been executed to date
- How many tests have passed or failed
- Where, why and the severity of all tests that have failed
- How many retests have been executed and how many failed on retest

©2000, Systems Integration Management Ltd.

Test team activity metrics

- how long to:
  - Plan stage or phase of testing
  - Produce test systems design
  - Prepare Test environments & develop utilities to maintain & recreate them
  - Prepare test scenarios, scripts, data & expected results

©2000, Systems Integration Management Ltd.

Test team activity metrics

- how long to:
  - Write automated test scripts
  - Execute tests at each significant stage of testing
  - Raise Defect reports and manage issues
  - Retest fixed defects
  - Manage the testing process

©2000, Systems Integration Management Ltd.

Part 2
How do we deliver metrics more easily

Wouldn't it be nice if Test Management tools produced all these metrics for us

Automatically

To some extent they do!

Reviewed 4 Test Management Tools

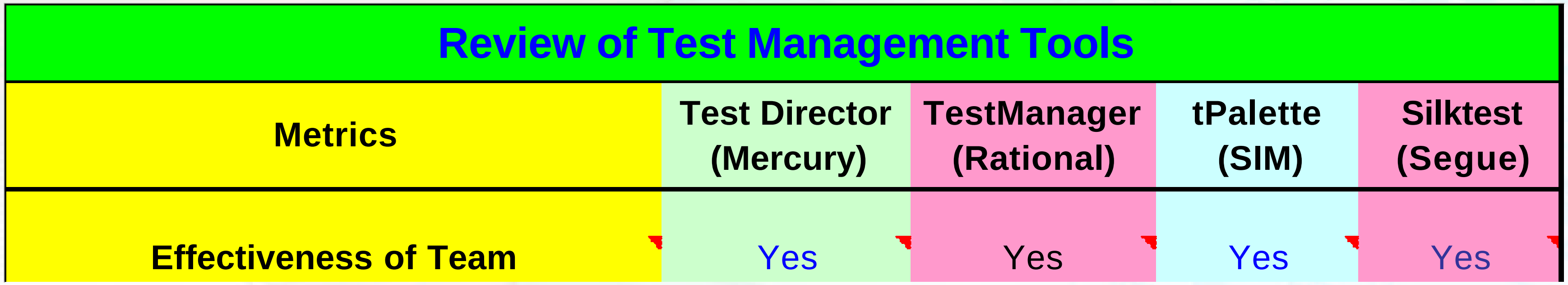
- Test Director
- Silktest

- TestManager
- *t*Palette

- Do these tools provide the metrics we need ?

- Do they put them in a format we can use?

Effectiveness of test team

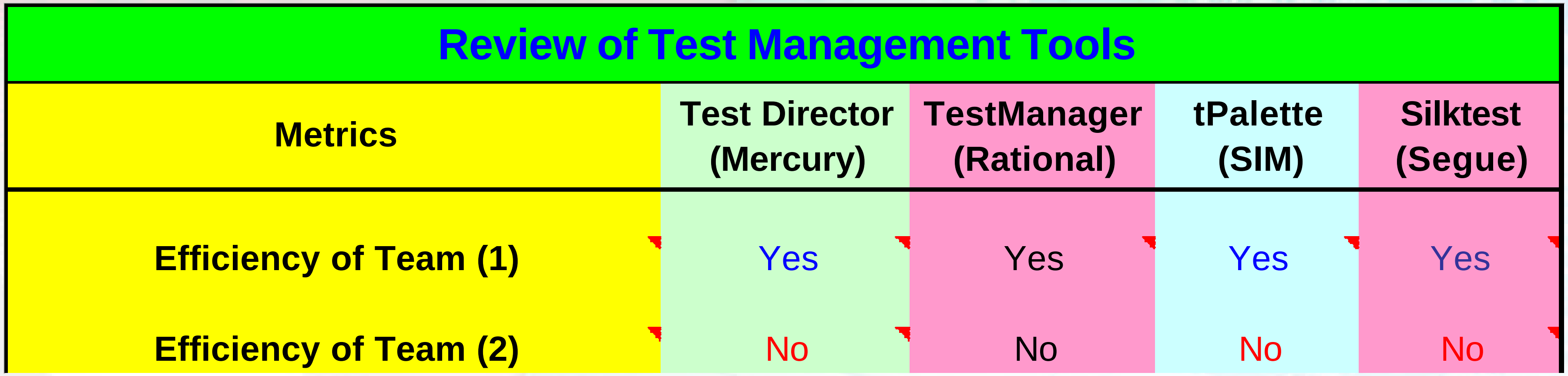


Review of Test Management Tools				
Metrics	Test Director (Mercury)	TestManager (Rational)	tPalette (SIM)	Silktest (Segue)
Effectiveness of Team	Yes	Yes	Yes	Yes

- Definition: The number of errors found in a given test phase related to the number found subsequently
- Statistics are not automatically produced

©2000, Systems Integration Management Ltd.

Efficiency of Test Team - Part 1



Review of Test Management Tools				
Metrics	Test Director (Mercury)	TestManager (Rational)	tPalette (SIM)	Silktest (Segue)
Efficiency of Team (1)	Yes	Yes	Yes	Yes
Efficiency of Team (2)	No	No	No	No

Definitions:

- (Part 1) Ratio of clean tests achieved to the number planned
- Statistics are not automatically produced

©2000, Systems Integration Management Ltd.

Efficiency of Test Team - Part 2

Review of Test Management Tools				
Metrics	Test Director (Mercury)	TestManager (Rational)	tPalette (SIM)	Silktest (Segue)
Efficiency of Team (1)	Yes	Yes	Yes	Yes
Efficiency of Team (2)	No	No	No	No

Definition: (Part 2) The cost of the testing stage

- Important ? YES. Available ? NO.

©2000, Systems Integration Management Ltd.

Analysis of Testing Phase

- Effectiveness of Phase - Quantity of defects found in the phase as a proportion of the whole.
- Tests completed / not completed.
- Causal Analysis; Breakdown of error type
  - by where it was caused
  - Where it was detected
  - Where it could have been prevented

Quantification of Effort

- Time:
  - To plan, design, analyse and develop tests
  - To execute tests
  - To report defects
  - To retest..
- Other Individual and Group activities within the phase.

Metrics Produced

- What do the Test Management Tools provide right now?
- Traditional Test Metrics
  - Quantity of tests Prepared / Performed
  - Quantity of Defects found
  - Type of Defects found
- Are they reported automatically?

Metrics Required

What additional assistance could the tools provide ?

- Test Project Metrics
  - Time taken - To Plan, Develop and Execute tests.
  - Cost of defect management; bug reporting, fixing and re-testing

THE COST OF TESTING -

OR NOT TESTING

Test Metrics without Tears
By
David Vaughan, Test Consultant

1. The profile of Testing has been raised significantly in the last 2-3 years, principally due to Year 2000 issues. Increasingly organisations accept that they need systematically and thoroughly to test software, but they are demanding the same levels of efficiency that they expect from all other areas of their businesses. To survive, the Test Manager has to demonstrate the effectiveness of the Testing Process. The evidence to show this is required to be more than software defect metrics – how many raised, how many outstanding/ fixed by severity and urgency etc. It needs to be a comprehensive measurement of the code being tested and of the testing service provided to test it.

2. It has always been desirable, but now it is necessary to provide evidence, not just that the software is fit for purpose, but that the activities of the test team are an effective and efficient use of staff. This paper suggests what metrics are required and how this further evidence may be obtained and reported with significantly lower overhead on testing resources.

3. The Test Manager should present a comprehensive set of statistics that inform Project and Business Management in several categories, as exemplified below:

3.1. Test Preparation figures, to quantify the work to be done before test execution – reported by Test Phase and or Development Stage

- How many test conditions have been identified and documented, by criticality/priority
- How many tests are required to exercise these test conditions
- Total number of tests to be built for Manual and Automated execution
- Percentage to be automated
- Test Environment preparation details including the quantities of Referential and Parameter data required to be set up
- Test Environment management procedures & utilities to be developed

3.2. Test Execution statistics that report the rate at which tests are being applied and the failures arising

- How long it takes to create, catalogue and execute both manual and automated tests
- How many tests have been executed to date
- How many tests have passed, failed
- Where, why and the severity of all tests that have failed
- How many retests have been executed and how many failed on retest

3.3. Test team activity metrics ie how long to:

- Plan stage or phase of testing
- Write the Test systems Design
- Prepare Test environment and develop utilities
- Prepare test scenarios, scripts, data and expected results
- Write automated test scripts
- Execute tests for each significant stage of testing
- Raise Defect reports and manage issues
- Retest fixed defects

©2000, Systems Integration Management Ltd.

4. We already produce graphs that depict defect detection rates, by severity, hopefully slowing down to the point where the software is considered robust enough to ship – and that it has passed all acceptance criteria. We must produce similar graphs depicting the team productivity for test development and execution.

5. Test resources are scarce and can be more productively used. So, the recording and presentation of testing metrics should be done automatically - ideally as a by-product of software we are using for other reasons, such as Test Management Tools that we are already using to record and report:
   - Tests built to date, sub-catalogued as necessary
   - Test execution metrics
   - Test failure metrics

6. I believe we should look to produce a complete metrical picture of testing. To do this efficiently, it should be possible to customise the test management tool to allow us to record and report individuals' test activity by category as shown in section 3 above.

7. At present test management tools can capture and report some metrics about test execution and defects; e.g. tests prepared; tests executed; tests failed; test execution times and so on. Most testing projects are based on three main phases. Specifically **Planning**, **Preparation** and **Execution**. For automated testing the distinction between preparation and execution may sometimes be blurred as it is necessary to execute a proportion of test data to verify the automated test script.

8. There are few if any facilities within test management tools to record and report test teams activities during the 1st two phases. But these metrics are needed to report progress; help next phase and future project estimates; prove value of activities and identify individuals' capabilities.

9. It would take relatively little further effort to specify additional fields and configure the Test Management Tool to capture the additional information required. The problem might be to ensure that they are completed accurately…if at all. Making the completion of these additional fields mandatory would be a good first step.

10. When achieved, such comprehensive data capture and measurement reporting would provide management with:

    - Weaknesses in the software development and test life cycle arising from a causal analysis performed on the reasons why tests failed.
    - Measurement as to the effectiveness of various test stages in defect detection and cost
    - A solid basis for estimating further test activities
    - A baseline for further improvement within each phase of test activity.

    The table depicted below indicates the capability of some test management software. We examined the stated functionality of three commercial test management tools, usually supplied with test automation software and SIM's independent tool tPalette[1]. Sadly, most of the data collected have to be manipulated and reported manually. As stated previously, most of the information required for test preparation and team activity is not readily collectible.

[1] *t*Palette has been developed to operate in conjunction with most leading test automation software. SIM's consultants use it where there is no TM tool in situ at a client site. There is an ongoing programme to improve its functionality, based on our clients and our own internal needs.

Review of Test Management Tools

Usual Metrics	**Test Director (Mercury)**	**TestManager (Rational)**	**Silktest (Segue)**	***t*****Palette (SIM)**
Effectiveness of Team	Yes	Yes	Yes	Yes
Efficiency of Team (1)	Yes	Yes	Yes	Yes
Efficiency of Team (2)	No	No	No	No
Effectiveness of Stage	Yes	Yes	Yes	Yes
Speed (1)	Yes	No	No	No
Speed (2)	Yes	Yes	Yes	Yes
Causal Analysis by Stage	Yes	Yes	Yes	Yes
Analysis of outstanding incidents	Yes	Yes	Yes	Yes
Activity Metrics				
Test System Design activities (overall)	No	Yes	Yes	No
Test System Design Activities (automation)	No	Yes	Yes	No
Time to develop automated tests	No	No	No	No
Time to develop Manual tests	No	No	Yes	No
Time to complete automated test run	Yes	Yes	Yes	No
Time to complete manual test run	No	No	Yes	Yes
Individual & team activity times	No	No	Yes	No
Group activity times	No	No	Yes	No
General TM tool Requirements				
Can data be exported to Excel?	No	Yes	Yes	Yes
Customisable?	Yes	Yes	Yes	No
Mandatory Customisations	No	No	?	Yes
Can metrics be displayed graphically	Yes	Yes	Yes	Yes

Conclusion

Test Managers have purchasing power, we may not always have direct control over the budget, however it is unlikely that a test tool would be purchased without our recommendation. Collectively, we can exert pressure on the test tool manufacturers to make provision for input data capture and make it mandatory or automatic. We should also ask them to provide better facilities to collate the data in diverse ways to automatically report a complete set of test metrics, ideally graphically.

This day will come and soon. Metrics will be produced without pain - and hopefully keep the test team and its manager in a job!

©2000, Systems Integration Management Ltd.

David Vaughan

David is a Senior Consultant working for the SIM Group. SIM specializes in software testing and has put in place a number of highly efficient testing systems that automatically test sophisticated and mission-critical software systems. SIM is the UK leader in providing efficient automated solutions for software testing. SIM's work has had a profound impact on the way companies approach testing, and many improvements to testing have been realized with SIM's help.

SIM has managed the development of testing strategies for software projects and has implemented automated testing techniques for many different software environments.

David's experience includes:

- Over 34 years in IT, including operations and most areas of software development, specifically programming, systems analysis, business analysis, internal and external consultancy, and senior IT management.
- More than 15 years hands-on testing activity and testing project management
- Member of the BCS Sigist and BCS SIG for software quality
- Has developed Testing Methodologies for many aspects of testing
- Testing consultant to many significant multinational and British organizations
- Training and lecturing in software testing techniques for SIM and Price Waterhouse and other major organizations
  In the production of this paper David enlisted the help of Joe Elledge to organize the research into various test management tools

Joe Elledge

After 10 years as a commercial insurance lawyer, Joe needed a change of direction and a new challenge; he joined the specialist testing consultancy SIM as a trainee test analyst. Having graduated from the SIM Testing Academy he rapidly rose to the role of test consultant and now leads SIM testing projects at client sites. Most of Joe's testing has been for a number of the UK's leading insurers.

Joe strives to provide clients with plain, easily understood tests and test results that enable developers to resolve defects and managers to understand and therefore control the development cycle. Joe also believes in re-using and delivering re-usable test assets that will continue to add value to the life of the software.

Joe specializes in Performance and Load testing and sees e-commerce as the most significant testing arena for the next few years.