

Tool Selection and Implementation

Paul Gerrard

Systeme Evolutif Limited
email: paulg@evolutif.co.uk
http://www.evolutif.co.uk



© 2000 Systeme Evolutif Ltd

Slide 1

Agenda

- ◆ What Can Test Execution Tools Do For You?
- ◆ Good and Bad Reasons for Buying a Tool
- ◆ What to Look for in a Tool
- ◆ Tool Implementation
- ◆ What is Success?
- ◆ Close.

© 2000 Systeme Evolutif Ltd

Slide 2

What Can Test Execution tools Do For You?

Drawbacks of manual testing

- ◆ Costly
- ◆ Slow
- ◆ Error-prone.

Speed factors

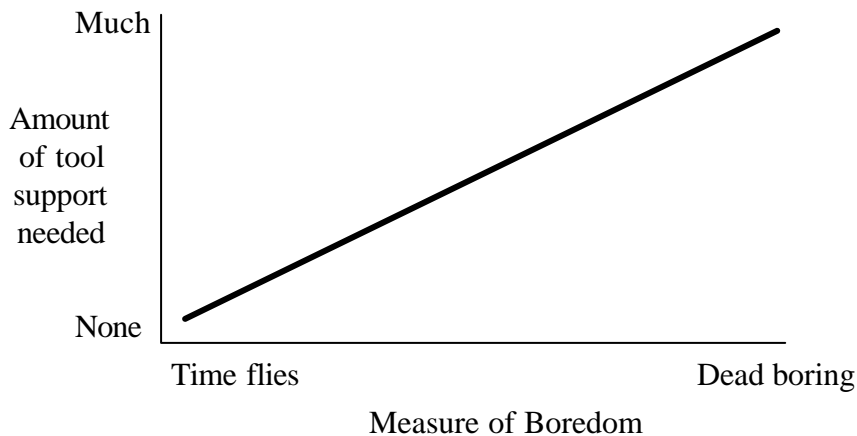
- ◆ Inputs entered at human speeds
- ◆ Testing only conducted during hours people work
- ◆ Outputs checked at human speeds.

Error proneness of manual testing

- ◆ Cloning of test cases
- ◆ Inexact repetition of tests
- ◆ Inaccurate results checking

- ◆ Of faults left after testing the majority were discovered by tests, but not noticed by the testers. (Based on work by Caper Jones and Basili)

Boredom index



And the reality...

- ◆ 90% of organisations have CAST tools (usually test execution)
- ◆ 40-50% CAST tools end up as shelfware
- ◆ <10% have benefited significantly
- ◆ >75% want more CAST tools

- ◆ Everyone knows there is great potential
 - few succeed and achieve real, lasting benefits.

Good and Bad Reasons for Buying a Tool

Bad reasons for buying a tool



- ◆ Test faster!
- ◆ Test more!
- ◆ Save money!
- ◆ Test earlier!
- ◆ Find more bugs!
- ◆ Do regression testing!
- ◆ Get tools, not people!

Test activities

- ◆ planning the testing to be done, both static and dynamic
- ◆ designing the test conditions (logical design)
- ◆ preparing the test input cases (physical and logical design)
- ◆ preparing test data (physical design)
- ◆ preparing the expected results from the requirements specification
- ◆ running the tests
- ◆ examining the mismatches when the expected results do not agree with the actual results
- ◆ isolating bug symptoms so they can be corrected
- ◆ monitoring what tests have been performed
- ◆ evaluating the quality of the testing performed, and extending the tests where required
- ◆ inspection of code, designs, requirements and test cases
- ◆ assessing non-functional aspects of software, such as usability and performance
- ◆ evaluating the quality of the software tested, i.e. release decision
- ◆ rerunning tests after bugs have been corrected
- ◆ updating tests when software is changed.

Where can the tool assist?

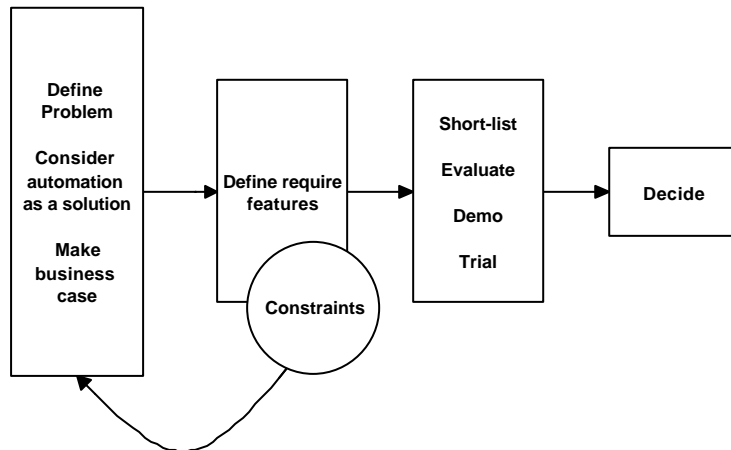
- ◆ planning the testing to be done, both static and dynamic
- ◆ designing the test conditions (logical design)
- ◆ preparing the test input cases (physical and logical design)
- ◆ preparing test data (physical design)
- ◆ preparing the expected results from the requirements specification
- ◆ running the tests
- ◆ examining the mismatches when the expected results do not agree with the actual results
- ◆ isolating bug symptoms so they can be corrected
- ◆ monitoring what tests have been performed
- ◆ evaluating the quality of the testing performed, and extending the tests where required
- ◆ inspection of code, designs, requirements and test cases
- ◆ assessing non-functional aspects of software, such as usability and performance
- ◆ evaluating the quality of the software tested, i.e. release decision
- ◆ rerunning tests after bugs have been corrected
- ◆ updating tests when software is changed.

Success with test running tools

- ◆ Have realistic expectations - no silver bullet
- ◆ Commitment (management and testers)
- ◆ Implementation project:
 - plan, mobilise, needs, select, train, pilot, review, roll-out....
- ◆ Tactical use of tools (not blanket use)
- ◆ Tools are for life, not just Christmas
- ◆ **PROCESS, THEN TOOLS = BENEFITS**

What to Look for in a Tool

Overview of the selection process



© 2000 Systeme Evolutif Ltd

Slide 15

Is a tool the right solution?

- ◆ Tools are not the only way:
 - code inspections are effective at fault finding
 - better documentation and test management can reduce the problem of omitting or repeating tests
 - better impact analysis reduces the tests to be run
- ◆ Tools are ‘sexy’, easy to buy and fun!
- ◆ Process improvement is hard
 - people, organisation, and resistance to change can be daunting and hard to overcome.

© 2000 Systeme Evolutif Ltd

Slide 16

Tool selection considerations

- ◆ Just what do you want to automate?
 - automated running or automated thinking?
 - regression testing a mature product?
 - to find bugs during development?
- ◆ Are your people interested in using tools?
- ◆ What skills are available to use the tool?
 - users can't use technical tools
 - automated scripting needs programmer skills.

Tool selection considerations (2)

- ◆ What technical environment(s) will the tool be used in?
- ◆ Are you organised enough to use tools?
 - who will design the tests?
 - who will write the automated scripts?
- ◆ Implement as a one-off or are tools part of an infrastructure project?
- ◆ If the wrong tool is selected the benefits will not be achieved.

Tool selection and evaluation team

- ◆ Give someone responsibility for managing the selection and evaluation process
- ◆ A single individual authorised to
 - investigate what tools are available
 - prepare a shortlist
- ◆ Before you start, you need to know:
 - what type of tool is needed
 - who might use it
 - factors for tool to qualify for the shortlist.

Evaluating the shortlist

- ◆ Involve representatives from
 - groups planning to use the tool
 - different job functions who will use it
- ◆ If you trial the tools
 - usability an important consideration to non-technical users so involve technical support staff
 - non-technical users need this support.
- ◆ The selection and evaluation team may become the implementation team.

How much help should the tool be?

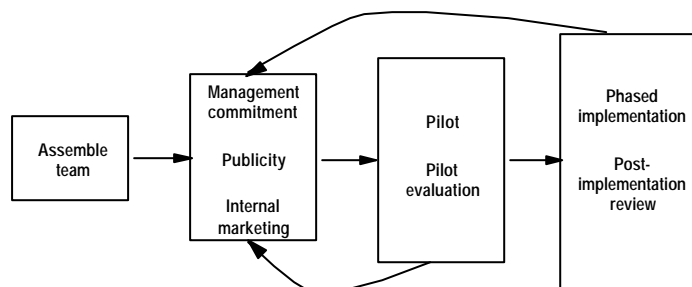
- ◆ How will we know a tool is effective?
 - do the testers ‘feel better’?
- ◆ Need measurable criteria for success
 - if length of time taken to run tests manually is the problem, how much quicker should the tests be run using a tool?
- ◆ Setting measurable criteria is not so difficult
 - setting reasonable expectations is the problem.

Measurable success criteria example

- ◆ Manual execution of tests currently takes 4 man-weeks
 - in the first 3 months of using the tool, 50 per cent of these tests should be automated, with the whole test suite run in 2–2½ man-weeks
 - next year at this time we aim to have 80 per cent of the tests automated, with the equivalent test suite being run in 5 man-days.

Tool Implementation

Tool implementation process



Keys to success

◆ Selling the concept

- commitment to testing a pre-requisite
- tools can save time/money but only if time is currently being spent on the task to be automated

◆ Selecting the right tool

- tool should fit the test process or you will have to refine/develop test process at the same time
- define the stages of testing the tool supports
- not all testing can be automated!

Keys to success (2)

◆ Implementation

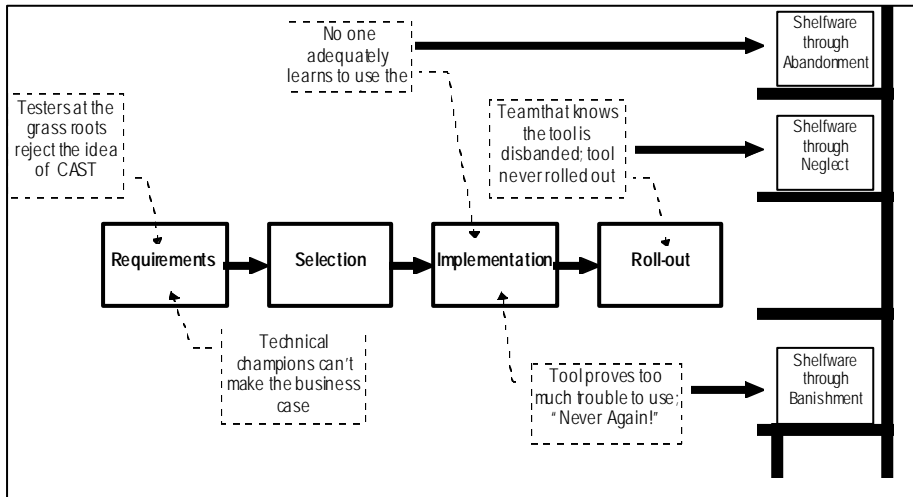
- CAST tools no different from any software
- process, training, documentation
- pilot to gain quick-wins and gain support

◆ Roll-out the things that work

- learn from pilot what works, what doesn't
- move skilled resources with the tool
- measure success and publicise

◆ One-off successes are difficult to roll-out.

Three routes to “shelfware”



© 2000 Systeme Evolutif Ltd

Slide 27

Pilot project

- ◆ Try out the tool on a small pilot project first
 - risk of problems encountered much lower
 - helps you to iron out process problems
- ◆ Business case for the pilot
 - objectives for the pilot, e.g.
 - lessons to be learned
 - implementation concerns
 - benefits to be gained.

© 2000 Systeme Evolutif Ltd

Slide 28

Evaluation of pilot

- ◆ Compare results with the business case
- ◆ If objectives met, lessons learned will help the next project gain more benefits
- ◆ If objectives not met
 - either the tool is not suitable
 - tool not yet being used in a suitable way
 - decision: abandon the tool, re-state realistic objectives, or change the approach to gain success next time.

Planned phased installation

- ◆ Publicise the success of the pilot
- ◆ Plan, conduct training, prepare in-house manuals
- ◆ Nominate a change management team to act as internal consultants
- ◆ Main risks to successful roll-out
 - failure to follow through with training
 - over-ambition
 - under investment.

What is Success?

Damn those faults!

- ◆ As usual, testing paradoxes...
- ◆ A successful test detects a fault
...but stops our automated test working
- ◆ A manual tester can cope easily
 - stop, log incident, do another test...
- ◆ The ~~tester~~ programmer says he can write us a
general purpose error handling routine
- ◆ Ah, that gets us going again.

False sense of security

- ◆ It works! Unattended run of 73 scripts!
- ◆ But at what cost? - more than we thought
- ◆ With what compromises?
 - “we took out all the test checks...”
- ◆ It’s a test Jim, but not as we know it
- ◆ It works! But what is ‘it’?
- ◆ It works! But what is ‘works’?
- ◆ What DOES this test prove?

The next software release

- ◆ So, we have test scripts that run reliably
- ◆ We’ve found a few bugs too
- ◆ New release is quite different and all our tests fail dramatically
- ◆ It takes several days to get them working again
- ◆ Now, lets get testing!

Regression testing

- ◆ The next software release arrives before we got the test scripts ‘working’ again
- ◆ Project Manager: “what do you mean, you haven’t started yet?”

- ◆ Need to plan for script maintenance
- ◆ Need to script for maintainability.

Close

Do automated tests find bugs?

- ◆ If the system we are automating tests for has a bug, when is the bug found?
- ◆ During recording, of course!
- ◆ Does the script we record 2nd time round find a new bug - not very often
- ◆ It is the process of automating test scripts that finds the bugs
- ◆ Do we need the tool to do this?

Conclusion

- ◆ What do you really, really want a tool for?
 - tools don't necessarily find bugs, but they are more fun than manual testing (sometimes)
 - automated tests provide confidence for rapid development - if the testers can keep up
 - essential for certain types of test - but do you know which?
- ◆ Be careful to separate test design from script development - two skills, both are required.

Papers

Systeme Evolutif web site:
www.evolutif.co.uk

Testing GUI Applications
a strategy for successful GUI test design and test automation

Selecting and Evaluating CAST Tools