Contents

• Why test automation objectives are important
• Commonly-held objectives
  – and why they are not good
• Good objectives for automation
• Testers and Automators
Why are objectives important for automation?

• automation costs money and takes time
  – even with free tools!

• why define objectives for automation
  – set expectations
    • unrealistic expectations cause many automation
      initiatives to fail
  – so everyone understands where we want to go
  – to know if you are going in the right direction

• testing and automation have different objectives

Contents

• Why test automation objectives are important
  Commonly-held objectives
    – and why they are not good

• Good objectives for automation
• Testers and Automators
Are these good objectives for automation?

– run more tests
– give testers a new skill / enhance their image
– run tests tedious and error-prone if run manually
– gain confidence in the system
– reduce the number of defects found by users

Some commonly-held objectives

• run regression tests overnight and weekends
• reduce testing staff
• find more bugs
• reduce elapsed time for testing
• automate x% of the testing
Run regression tests overnight & weekends

• what tests?
  – are they actually worth running at all?
  – easy to look impressive while giving little or no benefit
• if useful tests, then making use of under-used resource is a good idea

Reduce testing staff

• popular with managers!
• need more skills
  – tester skills not the same as automator skills
  – may need more people, not less
• tools don’t replace testers, they support them
  – tools don’t decide what a good test is!
• staffing & management issue, not automation objective
Find more bugs

• automation of regression tests
  – re-runs tests that have already passed
  – is unlikely to find bugs
• what is most effective at finding bugs?
  – testers
  – tests
  – exploratory testing
  – testing new code
• good objective for testing, not for automation

Reduce elapsed test time by \( x\% \)

• automation team goal for test execution?
• elapsed test time depends on
  – quality of the software
  – number & nature of faults found
  – environment availability
  – effectiveness of the tests
  – whether manual or automated
• which of these is under the control of the automators / automation team?
  - none!
  - well, not a lot
Impact on elapsed test time

Mature automation

- edit tests (maintenance)
- set-up
- execute
- analyse failures
- clear-up

Manual testing

Same tests automated

More mature automation
Automate x% of the tests

- are your existing tests worth automating?
  - if testing is in chaos, automating gives you faster chaos
- which tests to automate (first)?
- what % of manual tests should be automated?
  - “100%” sounds impressive but may not be wise
- what else can be automated
  - automation can do things not possible or practical in manual testing!

Manual vs automated

- manual tests
- automated tests
- tests not worth automating
- manual tests automated (% manual)
- tests not possible to run manually (esp verification)
Contents

• Why test automation objectives are important
• Commonly-held objectives
  – and why they are not good

Good objectives for automation
• Testers and Automators

Examples of good objectives

• find more regression bugs
  – measured by increase in DDP of regression tests by 10% over 6 mo.
• run most important tests using spare resource
  – top 10% of usefulness rating, run out of hours
• reduce elapsed time of tool-supported activities
  – measured for maintenance & failure analysis time
• improve automation support for testers
  – testers rate usefulness of automation support, how often utilities/automation features are used
Characteristics of good automation objectives

- measurable
  - EMTE (Equivalent Manual Test Effort)
  - test cycles run, coverage (e.g. features tested)
- different from test objectives
- should support testing activities
- realistic and achievable
- short and long term
- regularly re-visited and revised

EMTE – what is it?

- Equivalent Manual Test Effort
  - given a set of automated tests,
  - how much effort would it take
    - IF those tests were run manually
- note
  - you would not actually run these tests manually
  - EMTE = what you could have tested manually
    - and what you did test automatically
  - used to show test automation benefit
## EMTE – how does it work?

<table>
<thead>
<tr>
<th>a manual test</th>
<th>Manual testing</th>
<th>Automate the manual testing?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>the manual test now automated</td>
</tr>
</tbody>
</table>

- doesn’t make sense – can run them more
- only time to run the tests 1.5 times

## EMTE – how does it work? (2)

<table>
<thead>
<tr>
<th>Automated testing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

**EMTE**
EMTE example

- **example**
  - automated tests take 2 hours
  - if those same tests were run manually, 4 days

- **frequency**
  - automated tests run every day for 2 weeks
    (including once at the weekend), 11 times

- **calculation**
  - EMTE =

---

Uses of EMTE

- **decide whether or not to automate a set of tests**
  - EMTE should be > effort to automate them

- **automation efficiency**
  - example: a set of tests with EMTE = 50 hours
    - machine time to run automated tests = 2 hours
      - machine efficiency ratio is 25:1
      - if machines run the tests twice as fast (in 1 hr), then 50:1
    - tester time needed for a set of automated tests = 15 mins
      - human efficiency ratio is 200:1
      - i.e. using one hour of a tester’s time, can run tests that would have taken 200 hours to execute manually
Costs of tools

- effort to decide which tool to get
- effort to investigate how it works, how to use it
- provide machines to run the tests on
- effort to automate the tests (well)
- once tests are automated
  - testers: effort to select & start tests, analyse failures
  - automators: effort to debug, maintain tests
- plus purchase / license costs if not free tools

Sample ‘starter kit’ for metrics for test automation (and testing)

- some measure of benefit
  - e.g. EMTE
- average time to automate a test (or set of related tests)
- total effort spent on maintaining automated tests (expressed as an average per test)
- also measure testing, e.g. Defect Detection Percentage (DDP) – test effectiveness
  - more info on DDP on my web site & blog
Contents

• Why test automation objectives are important
• Commonly-held objectives
  – and why they are not good
• Good objectives for automation

Testers and Automators

Automation in iterative development

sprint/release (reqs, design, dev, test)  (if not TDD)

manual testing of this release (testers)

regression testing (automators automate the best tests)

run automated tests (testers)
Where are we going?

- **driver:** tester
- **engine:** test tool
- **car:** test infrastructure
- **passengers:** test cases
- **mechanic:** test automator

Responsibilities

**Testers**
- test the software
  - design tests
  - select tests for automation
    - requires planning / negotiation
- execute automated tests
  - should not need detailed technical expertise
- analyse failed automated tests
  - report bugs found by tests
  - problems with the tests may need help from the automation team

**Automators**
- automate tests (given to them)
- support automated testing
  - allow testers to execute tests
  - help testers debug failed tests
  - provide additional tools (home-grown)
- predict
  - maintenance effort for software changes
  - cost of automating new tests
- improve the automation
Summary: Key Points

- Test automation objectives are important
- Many commonly-held objectives are not good
  - often confused with objectives for testing
- Good objectives for automation are
  - measurable, realistic and achievable
- Testers and Automators: different roles

Shameless commercial plug - & a request*

*Wanted: your automation experience stories (for a new book)

I do a one-day seminar on Test Automation (the Smart Way)

www.DorothyGraham.co.uk  info@dorothygraham.co.uk