Automated Testing Tools

David Higginbottom
CCRA, E-commerce Testing
August 30, 2001

1

Scope

- Automated Software Quality (ASQ) tools is correct designation
- Types of testing tools (distributed and e-commerce testing)
- Benefits and Risks of each type

2

Types of Testing Tools

- Record/Playback tool
- Load Testing Tools

3

Record/Playback tools

- Most familiar tools in the testing business
  - Mercury Interactive's WinRunner
  - Rational's Robot
  - Segue's SilkTest
  - Compuware's QACenter products

4

Caution!

- Not all tools work in all environments
- No tool is a “one-size-fits-all”
- Know what you are getting – license requirements are important

5

Benefits, Risks and Costs

- Benefits
- Risks
- Common Mistakes
- Setting realistic expectations
- Getting and keeping management commitment

6

Benefits

- Repeatability
- Leverage
- Coverage
- Return on Investment

7

Repeatability

- Tests can be executed more than once
- Tests are executed consistently
- Saves time
- Increases predictability, reproducibility
- Tests must be maintainable (from 1st release to next)

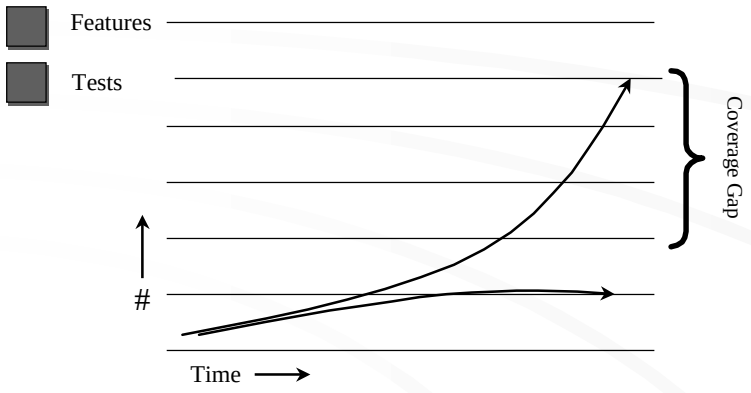
8

Leverage

- Execute high volumes of tests without manual capture
- Generate tests programmatically or from existing data
- Removes constraint of manual resources
- Requires technical design skills

9

Coverage Gap



10

Coverage Gap

- Functionality in application grows over time. Most testing is done in version 1
- Should spend more time in testing successive versions
- Highest risk is in the old functionality. New functionality might not work, but not as bad as old functionality not working anymore.

11

Coverage

- Usually more time spent testing when application is new, yet…
- Features and functions increase over time
- Essence of regression testing: test both old and new

12

Coverage

- Coverage requires maintainability: change old, add new
- Version 1 to Version 2 - 25% changes (1 developer. Should have 4 testers - have to test 100%

13

Cost of Automation

- Invest time for planning and training
- Invest in test environment
  - Hardware
  - Software
  - Data
- Invest in development
  - 10 x manual effort for GUI
  - 5 x manual effort for text

14

Cost of Automation

- Test tools do not reduce the number of testers or reduce testing cycle - if you gain a couple of days - increase testing coverage
- If management expects savings - setting up test team to fail
- Cost of tool - least $ spent to automate - effort to plan/develop will cost more
- Need dedicated/controlled environment - data must be controlled

15

Return on Investment

- Increased revenues
  - Time to market
  - Customer satisfaction
- Reduced costs
  - Less rework
  - Less support
- Quality is a competitive advantage

16

Return on Investment

- Put out a good product - less expense not having to fix and retest it later
- If it goes out with too many bugs - clients will not use it. Loss of revenue
- Quality product is competitive advantage

17

Risks

- Unpredictable results
- Short useful life
- Inexperienced testers
- Temporary testers
- Insufficient time, resources

18

Unpredictable Results

- Real time systems
  - Need simulator (stub out)
- Graphical output
  - Hard to compare images
- Volatile data
  - Cannot predict results (must have known condition - expected results)
- Don’t automate if system is unstable

19

Short Useful Life

- Porting to new language, platform
  - Major changes planned
- Targeted for replacement, retirement
  - No changes planned
- Not enough time to recoup investment
  - Benefits don’t outweigh costs

20

Inexperienced Testers

- Lack of application expertise
- Lack of test environment expertise
- Incorrect tests create confusion, cost time
- Tests only as good as tester
- Person who does not know application is good for manual testing - they act as a user

21

Inexperienced Testers

- Do not put inexperienced application tester in role of creating/running scripts
- Manual testing - oddball behavior, random testing - do not try to automate

22

Temporary Testers

- Borrowed resources
- Temporary personnel
- Investment in training, skills not recouped
- No ownership
- Future maintainability in question

23

Temporary Testers

- Need to be trained to use tools, procedures, etc. but will maybe not get return in investment - resource will leave
- Might not be an asset - will not be as dedicated - they are doing you a favour

24

Insufficient Time, Resources

- Automation takes more time in short run
- Automation costs more in short term
- Tools introduce complexity
- Savings come from long term

25

Common Mistakes

- Writing a program to test a program
- Programming around process problems
- Recording manual tests
- Decentralized libraries
- Ex: Developer writing (prog) script to test an installation application - he is just re-writing code. Writing for positive cases only. Not complete. A tester would write for negative cases also.

26

Writing a Program to Test a Program

- Don’t duplicate application program logic in another language
  - Test code will exceed code under test
  - Results are ambiguous
    - Which failed, the test or the application?
- Maintenance is excessive
  - Debugging is time-consuming
  - Distracts from original purpose: to get application ready, not test scripts

27

Duplicating Logic

Application Logic	Test Code Logic
Is Balance>=Trade? If yes, confirm Is Balance + Credit>= Trade? If yes, confirm If no, reject Trade	Is Balance>=Trade? If yes, was Trade confirmed? If no, report error Is Balance + Credit>= Trade? If yes, was Trade confirmed? If no, was Trade rejected? If no, report error

28

Programming around process problems

- If you don't know what to expect it is not a test, it is an experiment
  - Embedded logic creates ambiguity
  - Which failed, the application or the test

  Solve the real issue
  - Unstable data
  - Unknown requirements
  - Incomplete design

29

Need for a framework

- Framework is overall test library architecture
  - Issues, Resolution
  - Standards and Conventions
- Orderly division of responsibilities
  - Efficiency of development
  - Ease of maintenance
  - Plan for management

30

Setting realistic expectations…

- Test automation is strategic
- Tools cannot compensate for poor process
- Use consultants wisely
- Not everything can be automated
- Prioritize efforts based on payback
- Go slowly

31

Getting and keeping management commitment

- You will be measured by expectations, not results
- Commitment of time, money and resources to do it right
- Be conservative – you don't know what you don't know
- Use a pilot – monitor progress, adjust

32

Costs

- Per seat cost is over $5, 000
  - Costs vary from vendor to vendor and with quantity purchased
  - May have to purchase expensive add-ons to work in different environments

33

Questions ?

34

Load Testing tools

- High profile tools
  - Mercury Interactive's LoadRunner
  - Rational Performance Studio
  - Empirix e-test

35

Benefits

- Ability to simulate the activity generated by large numbers of users
- Thoroughly check out the effect of infrastructure changes before they go into production
- Check out the efficiency of program code, web services, etc…

36

Risks

- Need a great deal of technical skill to configure
- Need to set up a production-like test environment
- Testing must be done early enough in the process to assist with technical decision-making

37

Risks

- High costs
- Support from vendor – training ?
- Underestimation of set-up work

38

Costs

- Very expensive
  - 30K to in the millions
  - Depends on numbers of virtual users
  - Depends on which environment you need the tools to work in

39

Other options

- Pay for a “one-time” load test from one of the testing companies
- Bring in a contracting firm which has arrangements with a load test tool company (temporary licenses)

40

Questions???

41