

Chapter 6

Software Testing
ISTQB / ISEB Foundation Exam Practice

1 Principles	2 Lifecycle	3 Static testing
4 Dynamic test techniques	5 Management	6 Tools

Tool support for testing (CAST)

ISTQB / ISEB Foundation Exam Practice

1	2	3
4	5	6

Tool support

Contents

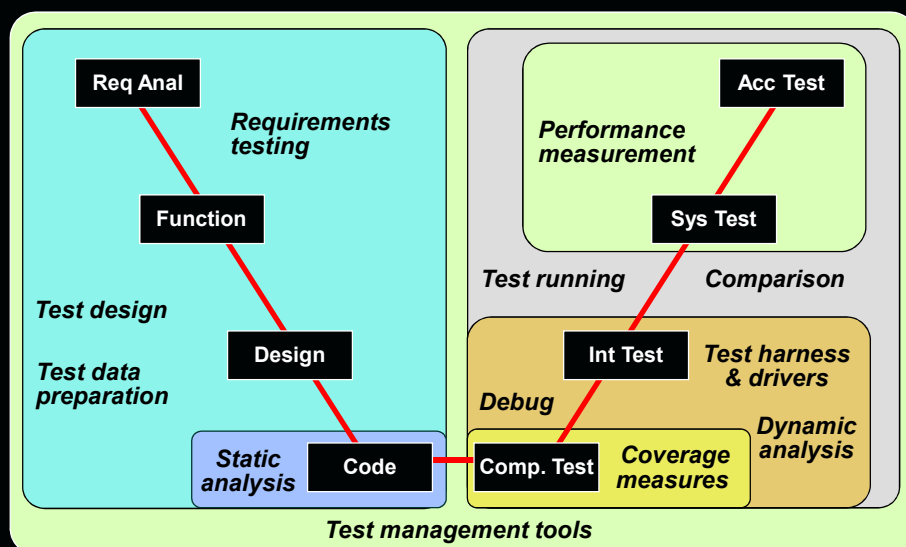
Types of CAST tool

- Why capture/replay is not test automation
- Automating and testing are separate skills
- Best practice

Testing tool classification

Requirements testing tools
 Static analysis tools
 Test design tools
 Test data preparation tools
 Test running tools - character-based, GUI
 Comparison tools
 Test harnesses and drivers
 Performance test tools
 Dynamic analysis tools
 Debugging tools
 Test management tools
 Coverage measurement

Where tools fit



Requirements testing tools

- **Automated support for verification and validation of requirements models**
 - consistency checking
 - animation

Tool information available from:

Ovum Evaluates Software Testing Tools (subscription service)

CAST Report, 1999

World Wide Web

Static analysis tools

- **Provide information about the quality of software**
- **Code is examined, not executed**
- **Objective measures**
 - cyclomatic complexity
 - others: nesting levels, size

Test design tools

- **Generate test inputs**
 - from a formal specification or CASE repository
 - from code (e.g. code not covered yet)

Test data preparation tools

- **Data manipulation**
 - selected from existing databases or files
 - created according to some rules
 - edited from other sources

Test running tools 1

- Interface to the software being tested
 - Run tests as though run by a human tester
 - Test scripts in a programmable language
 - Data, test inputs and expected results held in test repositories
 - Most often used to automate regression testing
-

Test running tools 2

- **Character-based**
 - simulates user interaction from dumb terminals
 - capture keystrokes and screen responses
- **GUI (Graphical User Interface)**
 - simulates user interaction for WIMP applications (Windows, Icons, Mouse, Pointer)
 - capture mouse movement, button clicks, and keyboard inputs
 - capture screens, bitmaps, characters, object states

Comparison tools

- **Detect differences between actual test results and expected results**
 - screens, characters, bitmaps
 - masking and filtering
 - **Test running tools normally include comparison capability**
 - **Stand-alone comparison tools for files or databases**
-

Test harnesses and drivers

- **Used to exercise software which does not have a user interface (yet)**
- **Used to run groups of automated tests or comparisons**
- **Often custom-build**
- **Simulators (where testing in real environment would be too costly or dangerous)**

Performance testing tools

- **Load generation**
 - drive application via user interface or test harness
 - simulates realistic load on the system & logs the number of transactions
 - **Transaction measurement**
 - response times for selected transactions via user interface
 - **Reports based on logs, graphs of load versus response times**
-

Dynamic analysis tools

- **Provide run-time information on software (while tests are run)**
 - allocation, use and de-allocation of resources, e.g. memory leaks
 - flag unassigned pointers or pointer arithmetic faults

Debugging tools

- **Used by programmers when investigating, fixing and testing faults**
- **Used to reproduce faults and examine program execution in detail**
 - single-stepping
 - breakpoints or watchpoints at any statement
 - examine contents of variables and other data

Test management tools

- **Management of testware: test plans, specifications, results**
- **Project management of the test process, e.g. estimation, schedule tests, log results**
- **Incident management tools (may include workflow facilities to track allocation, correction and retesting)**
- **Traceability (of tests to requirements, designs)**

Coverage measurement tools

- Objective measure of what parts of the software structure was executed by tests
- Code is instrumented in a static analysis pass
- Tests are run through the instrumented code
- Tool reports what has and has not been covered by those tests, line by line and summary statistics
- Different types of coverage: statement, branch, condition, LCSAJ, et al

1	2	3
4	5	6

Tool support

ISTQB / ISEB Foundation Exam Practice

Content

Types of CAST tool

Why capture/replay is not test automation

Automating and testing are separate skills

Best practice

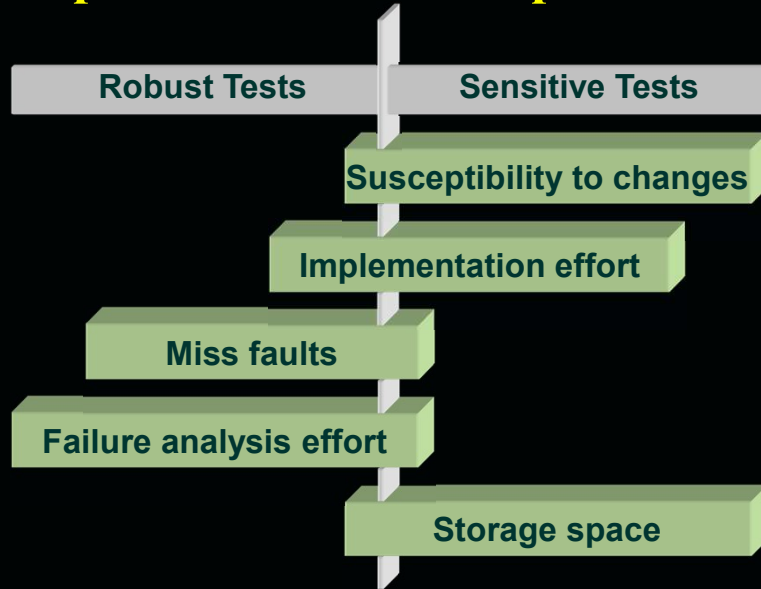
Advantages of recording manual tests

- **documents what the tester actually did**
 - useful for capturing ad hoc tests (e.g. end users)
 - may enable software failures to be reproduced
 - **produces a detailed “script”**
 - records actual inputs
 - can be used by a technical person to implement a more maintainable automated test
 - **ideal for one-off tasks**
 - such as long or complicated data entry
-

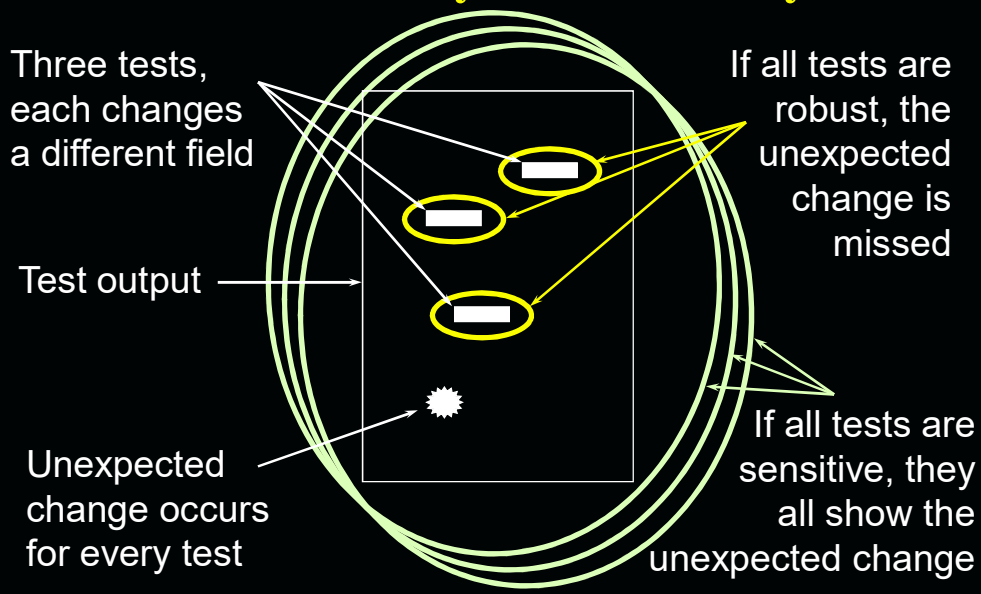
Captured test scripts

- **will not be very understandable**
 - it is a programming language after all!
 - during maintenance will need to know more than can ever be ‘automatically commented’
- **will not be resilient to many software changes**
 - a simple interface change can impact many scripts
- **do not include verification**
 - may be easy to add a few simple screen based comparisons

Compare seldom vs. compare often



Too much sensitivity = redundancy



Automated verification

- **there are many choices to be made**
 - dynamic / post execution, compare lots / compare little, resilience to change / bug finding effective
- **scripts can soon become very complex**
 - more susceptible to change, harder to maintain
- **there is a lot of work involved**
 - speed and accuracy of tool use is very important
- **usually there is more verification that can (and perhaps should) be done**
 - automation can lead to better testing (not guaranteed!)

1	2	3
4	5	6

Tool support

ISTQB / ISEB Foundation Exam Practice

Content

Types of CAST tool

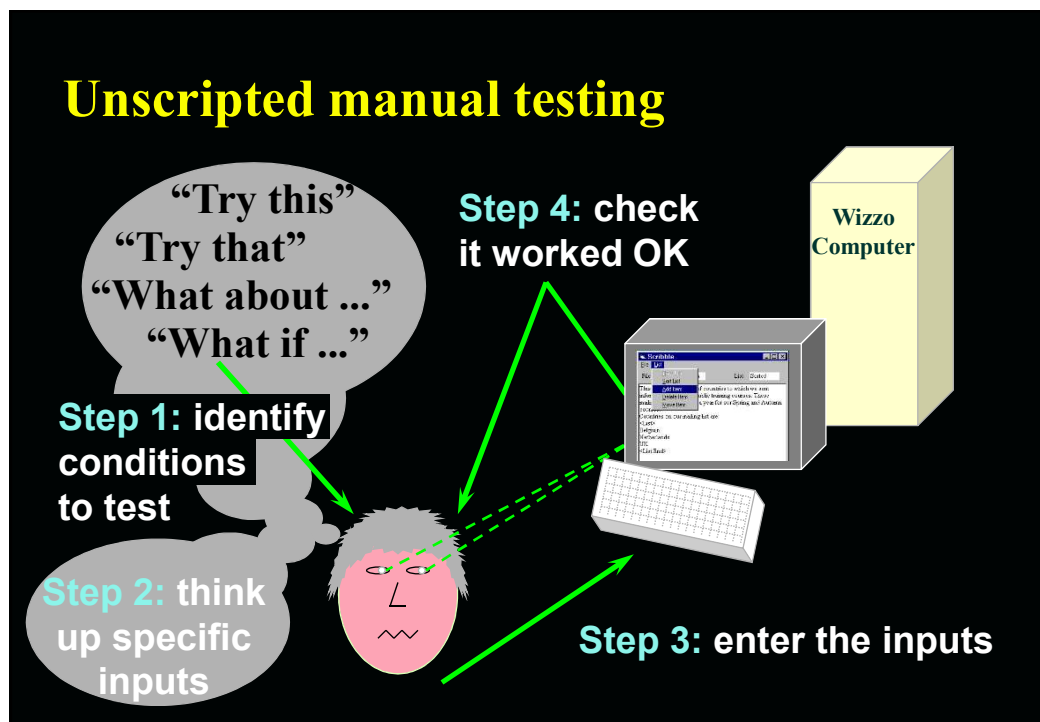
Why capture/replay is not test automation

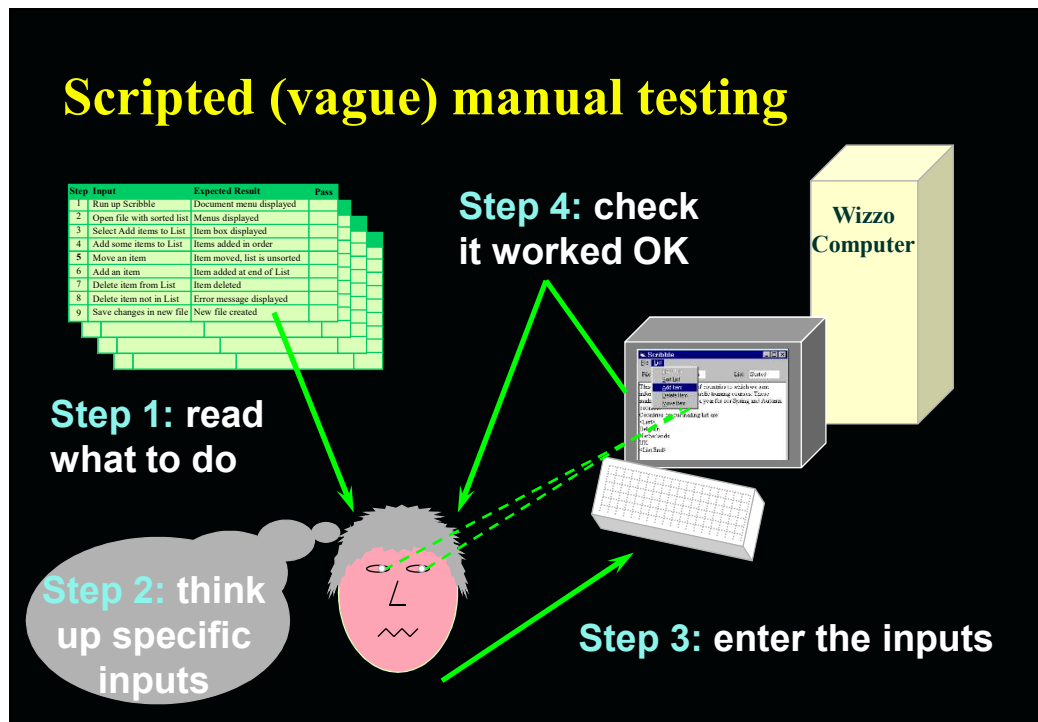
Automating and testing are separate skills

Best practice

Effort to automate

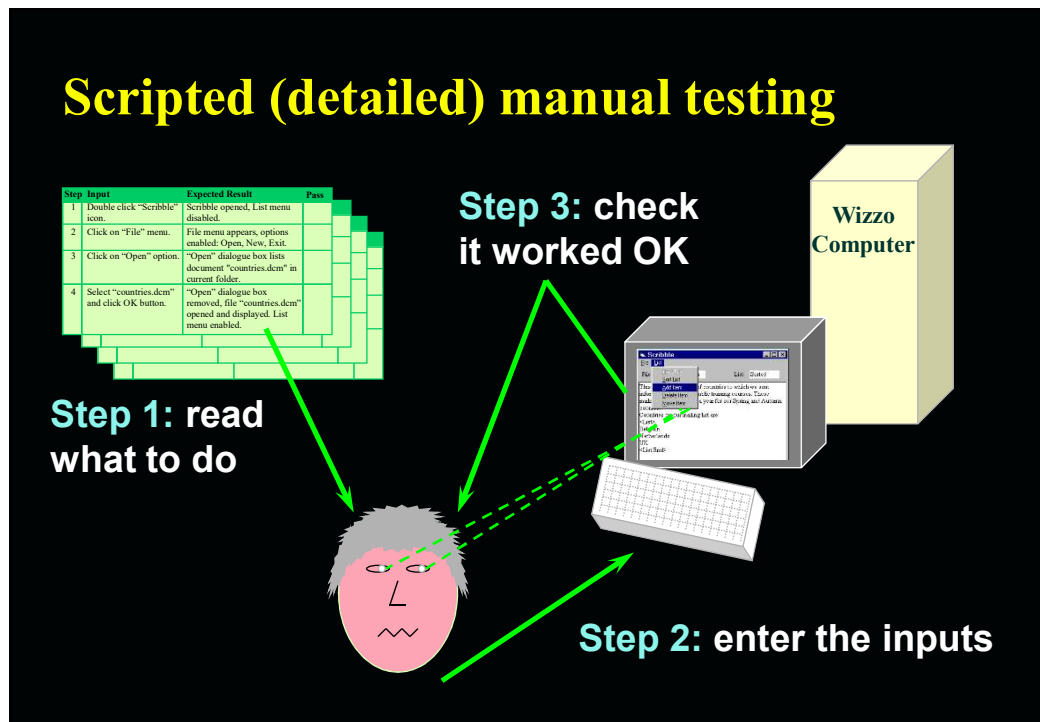
- The effort required to automate any one test varies greatly
 - typically between 2 and 10 times the manual test effort
- and depends on:
 - tool, skills, environment and software under test
 - existing manual test process which may be:
 - unscripted manual testing
 - scripted (vague) manual testing
 - scripted (detailed) manual testing





A vague manual test script

Step	Input	Expected Result	Pass
1	Run up Scribble	Document menu displayed	
2	Open file with sorted list	Menus displayed	
3	Select Add items to List	Item box displayed	
4	Add two items to List	Items added in order	
5	Move an item	Item moved, list is unsorted	
6	Add an item	Item added at end of List	
7	Delete item from List	Item deleted	
8	Delete item not in List	Error message displayed	
9	Save changes in new file	New file created	



1	2	3
4	5	6

Tool support

Content

Types of CAST tool

Why capture/replay is not test automation

Automating and testing are separate skills

Best practice

ISTQB / ISEB Foundation Exam Practice

Don't automate too much long term

- **as the test suite grows ever larger, so do the maintenance costs**
 - maintenance effort is cumulative, benefits are not
- **the test suite takes on a life of its own**
 - testers depart, others arrive, test suite grows larger
nobody knows exactly what they all do ... dare not throw away tests in case they're important
- **inappropriate tests are automated**
 - automation becomes an end in itself

Maintain control

Best
practice

- **keep pruning**
 - remove dead-wood: redundant, superceded, duplicated, worn-out
 - challenge new additions (what's the benefit?)
- **measure costs & benefits**
 - maintenance costs
 - time or effort saved, faults found?

Invest

Best
practice

- **commit and maintain resources**
 - “champion” to promote automation
 - technical support
 - consultancy/advice
- **scripting**
 - develop and maintain library
 - data driven approach, lots of re-use

Tests to automate

Best
practice

- **run many times**
 - regression tests
 - mundane
- **expensive to perform manually**
 - time consuming and necessary
 - multi-user tests, endurance/reliability tests
- **difficult to perform manually**
 - timing critical
 - complex / intricate

Automate

Tests not to automate

Best
practice

- **not run often**
 - if no need (rather than expensive to run manually)
 - one off tests (unless several iterations likely and build cost can be minimised)
- **not important**
 - will not find serious problems
- **usability tests**
 - do the colours look nice?
- **some aspects of multi-media applications**

Automate

1	2	3
4	5	6

Tool support

ISTQB / ISEB Foundation Exam Practice

Summary: Key Points

There are many different types of tool support for testing, covering all areas of the life cycle.

Automation requires planning and up-front effort

Identify and adopt best practice