

Test Automation and Keyword-driven testing

Brian Nielsen,

`bnielsen@cs.aau.dk`



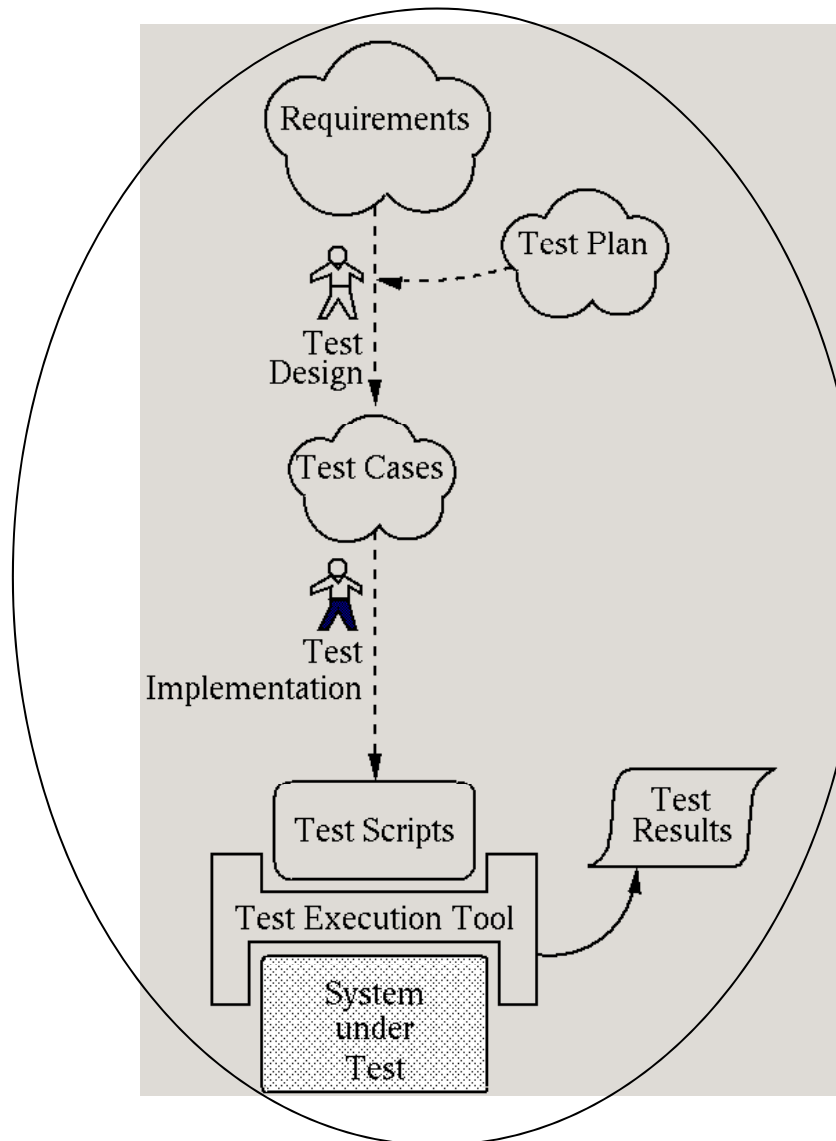
BRICS

Basic Research
in Computer Science



CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

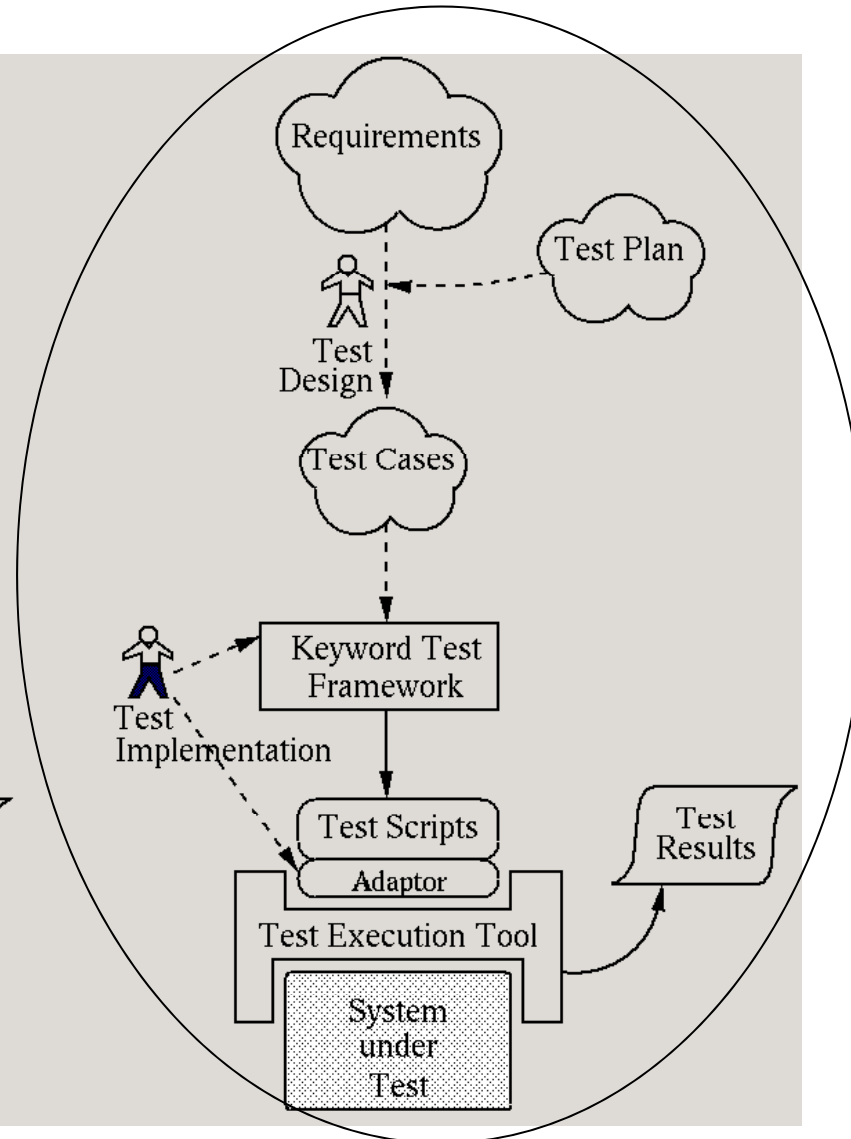
3. Script-Based Testing



- +/- test impl. = programming
- + automatic execution
- + auto regression testing
- fragile tests break easily?
(depends on abstraction)
- ad-hoc coverage
- no coverage measurement

4. Keyword-Driven Testing

- + abstract tests
- + automatic execution
- + auto regression testing
- robust tests
- ad-hoc coverage
- no coverage measurement



Script-based Testing

Test inputs and observations (verdict assignment) *programmed* in scripts in dedicated or general-purpose languages.

- + Repeatable (for regression testing)
- + Data driven testing: Fixed scripts, each parameterized with different data
- Mixes test harness and logical test cases
- Must be updated when specs (or impl) change
- Very lengthy
- Good programmers may write well-structured reusable test code but *“it is just test code”*

Keyword Driven Testing

- Script based automation where test case design is *separated* from automation
 - Focus on actions user/environment can do on objects in SUT (at different abstraction levels)
 - "Actions" appear in scripts,
 - "Action-code" implements the action
- aka "Action-word" testing (or, table driven testing), < 1994
- Black-box, subsystem, accept tests
- Automated execution using a "framework".

Keyword Driven Testing

- + Concise, flexible, maintainable,
- + Read-/writeable by non-programmers
- ÷ Expressiveness of a scripting language
 - Control structures and complex computations
 - (Branching in test cases, complex data, matching of expected results)

Keyword Driven Test

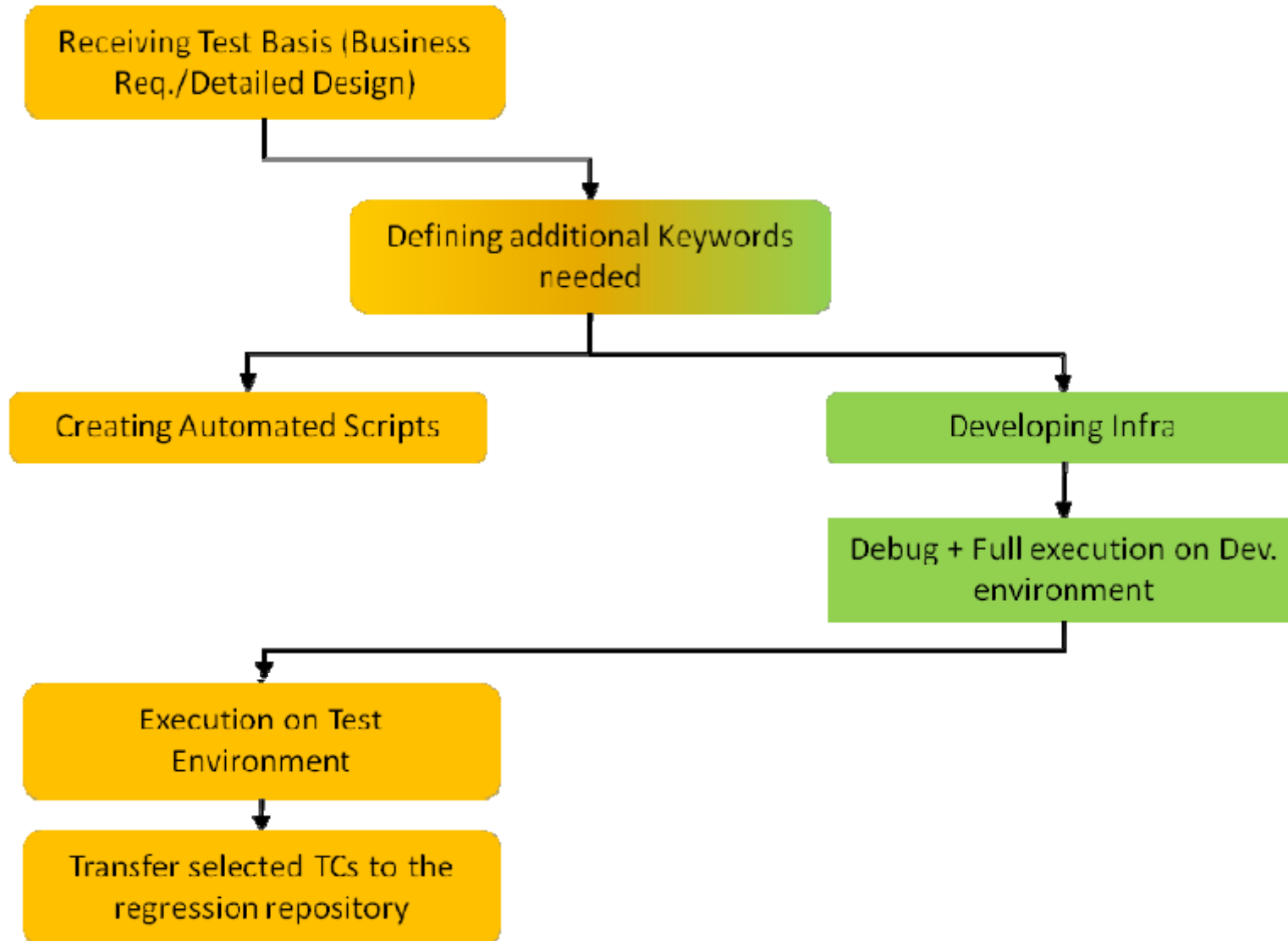
Eg. Sequence

Object	Keyword	Value
LoginDialog	Login	"badUserId", "GoodPass"
	VerifyLogin	"Login failed"
LoginDialog	Login	"goodUserId", "badPass"
	VerifyLogin	"Login failed"
LoginDialog	Login	"goodUserId", "GoodPass"
	VerifyLogin	"welcome"
WelcomeDialog	changeLogin	"newUserName", "newpwd"
	confirm	"goodUserId", "newpwd"
	confirmChange	"Change Failed"

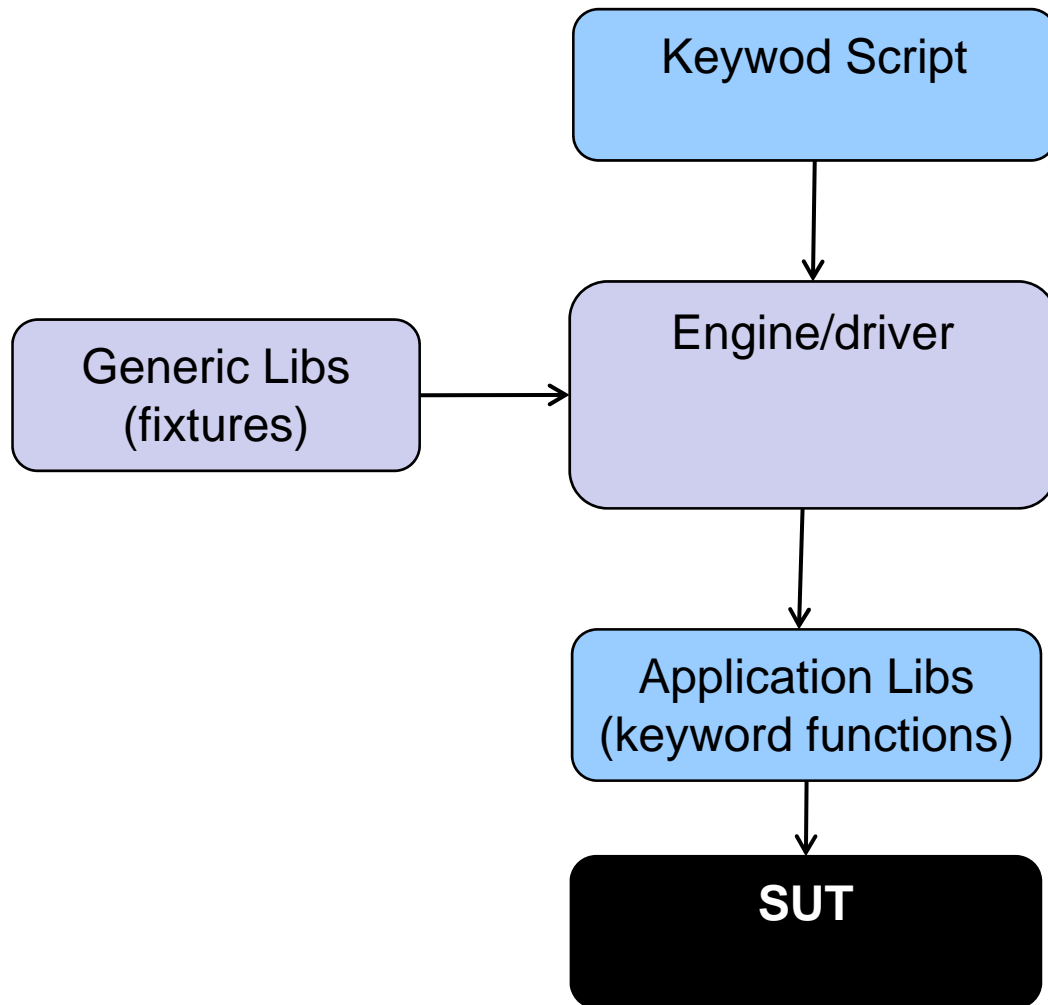
KWD Process

Test Expert

Automation Expert



Automation Fixture



```
Open(Scripts)
While(!EOF){
  Keyword, data=readFile();
  Case(keyword){
    Login: widget->submit(data);

    verifyLogin: ...
  }
Close()
```

}

Tools

- SAFS, open source
- EMOS Framework, open source
- FIT/FitNesse, open source
- Certify, Worksoft
- Unified TestPro, SDT
- TestFrame, LogicaCMG
- TestArchitect, LogiGear
- Tools with built-in keyword support:
 - TestQuest Pro,
 - QuickTest Professional

Example

<http://localhost:8080/FitBookExamples.Chapter4TestingActions.Fig6TestChatServer>

The screenshot shows a Mozilla Firefox browser window displaying a test runner interface for 'FitBookExamples.Chapter4TestingActions.Fig6TestChatServer'. The interface includes a sidebar with navigation options like 'Test', 'Edit', 'Properties', 'Refactor', 'Where Used', 'Search', 'Files', 'Versions', 'Recent Changes', 'User Guide', and 'Test History'. The main area shows a table of test actions and their results.

fit.ActionFixture	
start	ChatServerActions
enter	user anna
press	connect
enter	room lotr
press	new room
press	enter room
enter	user luke
press	connect
press	enter room
check	occupant count 2

At the bottom of the browser window, there is a search bar and a status bar showing the date '26-09-2010'.

```
14 public class ChatServer2 extends fit.Fixture { //COPY:ALL
15     public static ChatRoom CHAT; //COPY:ALL
16     private String userName = ""; //COPY:ALL
17     private String roomName = ""; //COPY:ALL
18     private boolean roomCreatedOK = false;
19     private boolean roomEnteredOK = false;
20     //COPY:ALL
21     public ChatServer2() { //COPY:ALL
22         CHAT = new ChatRoom(); //COPY:ALL
23     } //COPY:ALL
24     public void user(String userName) { //COPY:ALL
25         this.userName = userName; //COPY:ALL
26     } //COPY:ALL
27     public void connect() { //COPY:ALL
28         CHAT.connectUser(userName); //COPY:ALL
29     } //COPY:ALL
30     public void room(String roomName) { //COPY:ALL
31         this.roomName = roomName; //COPY:ALL
32     } //COPY:ALL
33     public void newRoom() { //COPY:ALL
34         roomCreatedOK = CHAT.userCreatesRoom(userName, roomName);
35     } //COPY:ALL
36     public void enterRoom() { //COPY:ALL
37         roomEnteredOK = CHAT.userEntersRoom(userName, roomName);
```

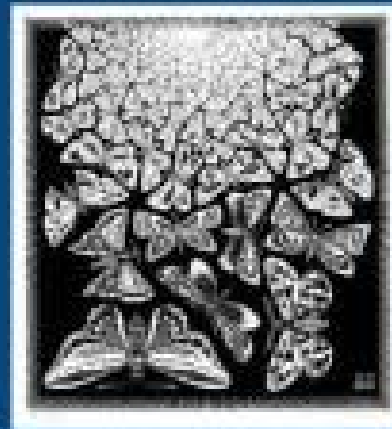
Robert C. Martin Series



Fit for Developing Software

Framework for Integrated Tests

- The definitive guide to the Fit testing framework
- Use Fit to test business rules and ensure the business value of software
- Dramatically improve communication between software consumers and creators
- Master Fitbase, the Web-based environment for creating, managing, and running Fit tables



Rick Mugridge
Ward Cunningham

Foreword by Brian Marick

<http://fit.c2.com/wiki.cgi?FitBook>

<http://my.safaribooksonline.com/0321269349>

Comparison

Criteria		Manual	Script	KWD	Model-Based
Cost	Development	L	M	M	L
	Maintenance	H?	H		L
	Tools/equip	L	M	M	H
Time	Time to first test	L	M	M	M-H
	Re-execution	H	L	L	L
	Maintenance	L	H	H	L
Learn-ability	Skills	L	H	M*	H*
	Maturity	L	M	M-H	H
	Readability	L	L	M	H*
	Interest / Challenging	L	M	M	H
Effective-ness	Bug detection	H	L	L	H
	Tracability	L	L	M	H
	Coverage	L	L	M	H

References

- http://en.wikipedia.org/wiki/Keyword-driven_testing
- <http://www.cs.waikato.ac.nz/~marku/mbt/>
- <http://www.onestoptesting.com/automation-framework/keyword-driven-testing/>
- <http://www.softwaretestinghelp.com/wp-content/qa/uploads/2010/01/keyword-driven-testing.pdf>
- http://www.musala.com/press/PR_2008_03_17/pr/Keyword-driven%20Testing.pdf
- *Mark Fewster and Dorothy Graham, Software Test Automation, 1999 (Chapter 22)*
- *M. Utting Practical Model-based testing*