



The Design and Implementation of a Flexible, Reusable, and Maintainable Automation Framework

HUNG Q. NGUYEN
LogiGear® Corporation

- Prepare you to build a successful and reusable automation architecture
- Share the keyword approach to creating an automation framework
- Share lessons learned in implementing a flexible architecture

- The early days
- Developing an automation framework
- The table-driven approach
- The keyword-driven approach

- Collect acceptance/regression test cases to be automated
- Record and script test cases
- Improve reusability
 - Parameterize hard-coded values
 - Separate data from code by moving variables to INCLUDE files
 - Create utility functions to be shared
- Train test specialists to run scripts

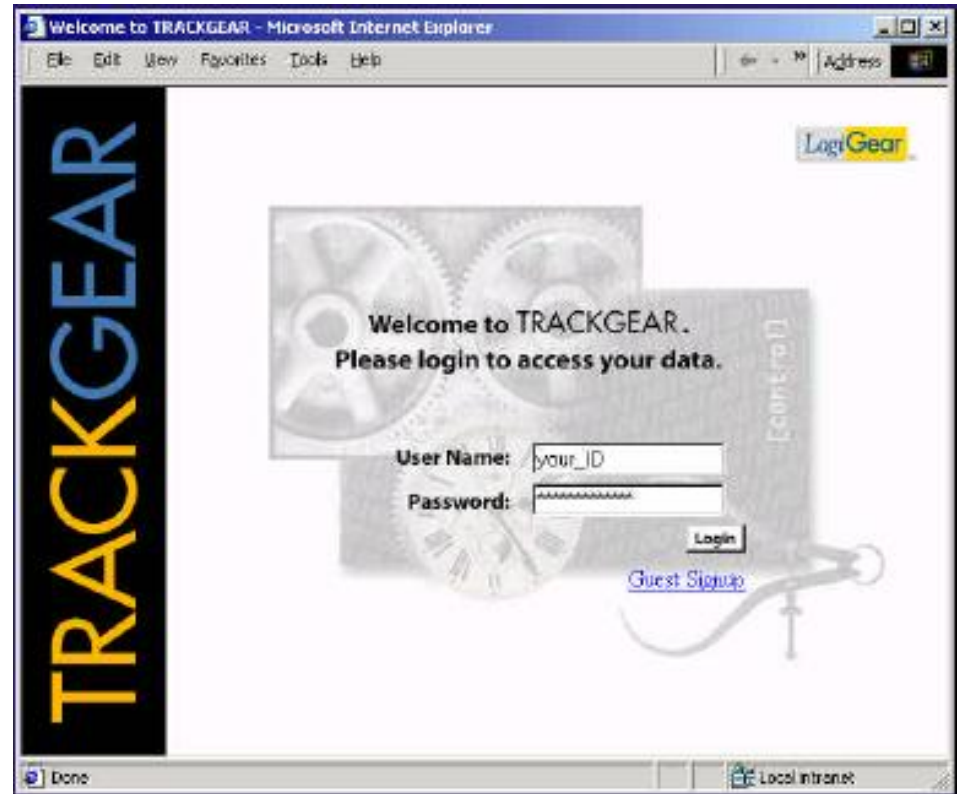
- Work with test specialists to understand their testing needs
- Go beyond acceptance/regression tests--
Analyzing user-scenario test cases
- Recognize the difference between task driven and object-driven test cases

- **Object-Driven**

- Click User Name text box
- Enter your_ID
- Click Password text box
- Enter your_password
- Click Login button

- **Task-Driven**

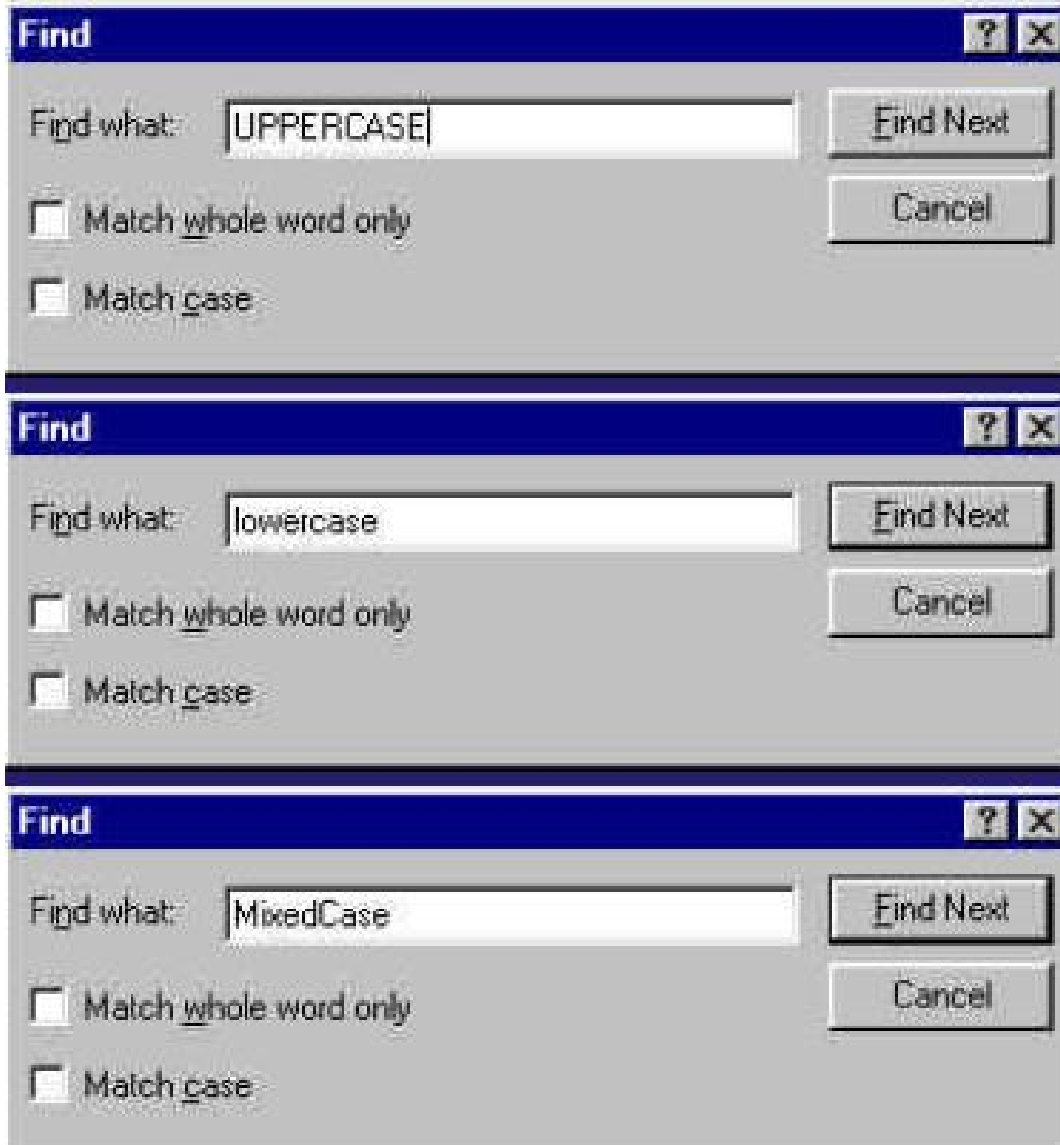
- Log in using
 - User Name = your_ID
 - Password = your_password



- Pre-separating data and code
 - Start by defining functions to be written
 - Variablize data and keep variables in INCLUDE files
- Pair up a test specialist and an automation engineer to improve communication and to ensure that the framework design and implementation meet the test objectives
- Train test specialists to run test scripts

- Take advantage of tester's familiarity with test case creation using tables and matrices
- Accommodate localization projects
- Recognize the importance of patterns in test cases
- Enable testers to catalog test cases with Excel spreadsheets
- Enable testers to specify expected results in spreadsheets

A Table-Driven Example



A Table-Driven Example

- for (i=1; i<= iLastDataSet; i++)
 - Open the dialog box.
 - Use the data in **DataSet[i]** (The first set is 1 and the last set is 12) to set the values of Match Case, Match Whole Word and Find What controls.
 - Click Find Next.
 - Verify the results.

	PROPERTY											
CONTROL	1	2	3	4	5	6	7	8	9	10	11	12
Match case	OFF	OFF	ON	ON	OFF	OFF	ON	ON	OFF	OFF	ON	ON
Match whole	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON
Find What	"UPPERCASE"	"UPPERCASE"	"UPPERCASE"	"UPPERCASE"	"lowercase"	"lowercase"	"lowercase"	"lowercase"	"MixedCase"	"MixedCase"	"MixedCase"	"MixedCase"

- Business issues
- People and process issues
- Technology issues

- Need to expand our service offerings and share success through our test automation expertise
- Need to have a methodology for quick deployment of test automation
- Need to build a transferable architecture
- Need a better approach to test automation job costing

- Need to deliver an automation program that is practical, explainable, and trainable
- Need to be more cost effective through reusability across projects
- Need to make technology a viable business solution
- Need a tangible approach to deciding between manual testing and automated testing

- Need to standardize test methodology--
Enabling testers and automation engineers to collaborate
- Enable testers to better specify their needs and automation engineers to better serve those needs
- Need to integrate test automation as part of the process of software testing

- Need testers to focus on test case design, and automation engineers to focus on driver script writing
- Make data more visible and understandable from the human perspective
- Need to incorporate test case design techniques with Excel, which test specialists are already familiar

- Need to build an architecture that's tool independent as well as application independent
- Need to improve the ability to share code across projects and tools
- Need to separate control of task variables, input variables, and code
- Need to integrate action keyword into the existing data-driven model

- Want to focus the development and maintenance of test scripts on the navigation of the application under test
- Need to take advantage of Excel features to automate test case and test data creation
- Need to incorporate test case design techniques using Excel, any database, XML, or other viable data service solutions

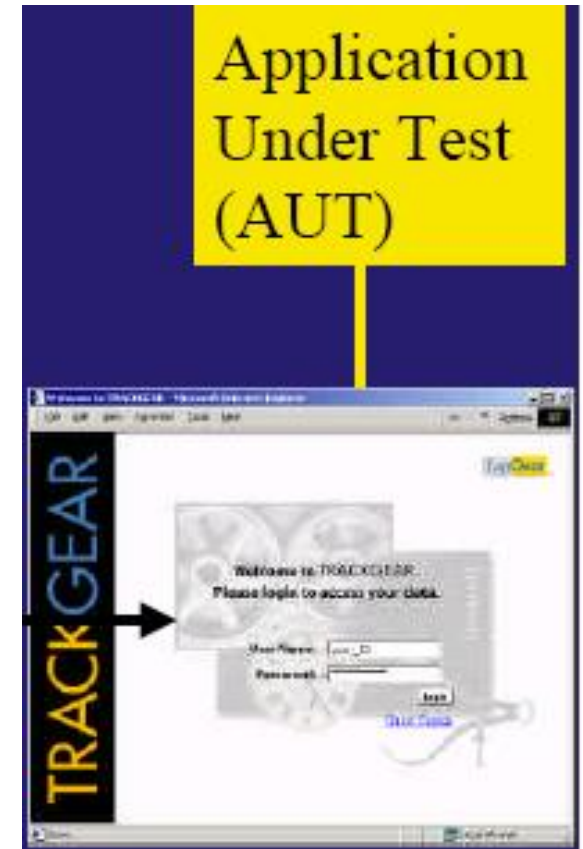
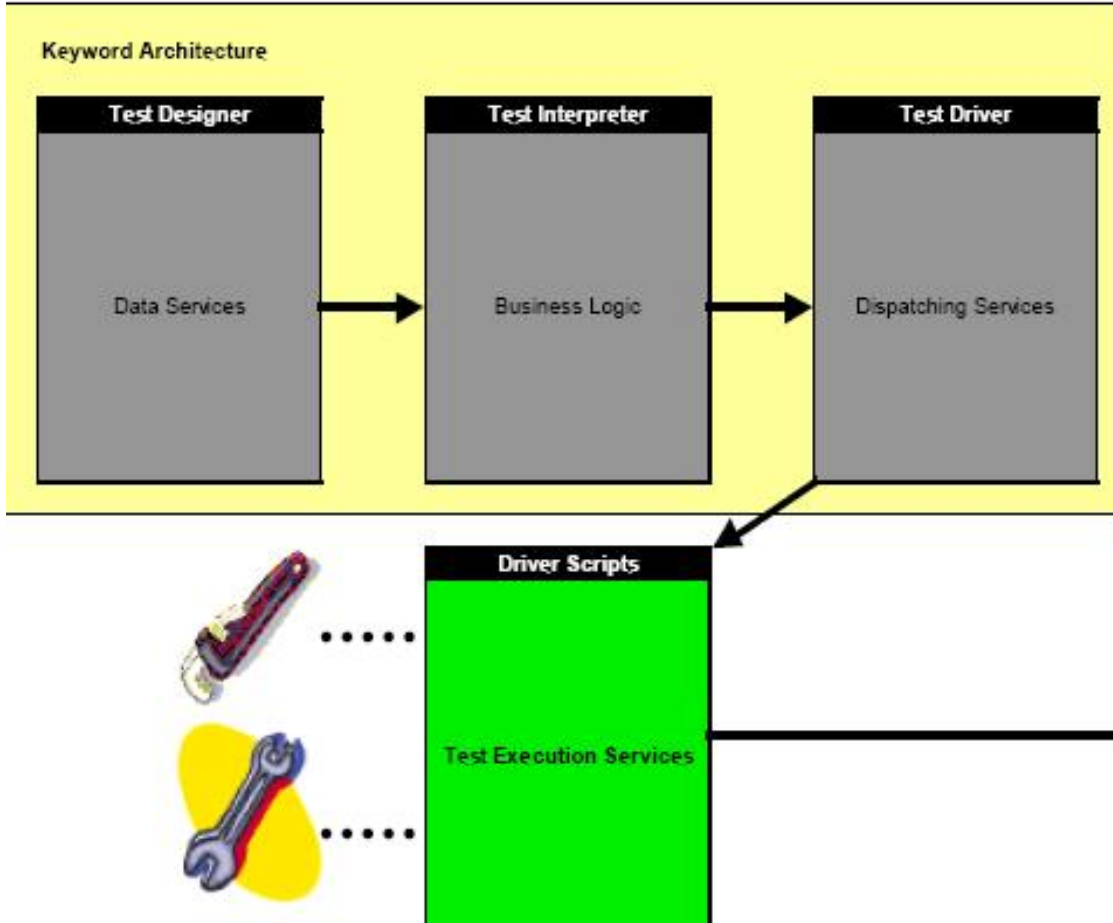
Integrated Testing Solutions =
[Test Specialist's Domain Expertise] +
[Manual Testing] +
[Automated Testing: Reusable Framework &
Application Specific Scripts]

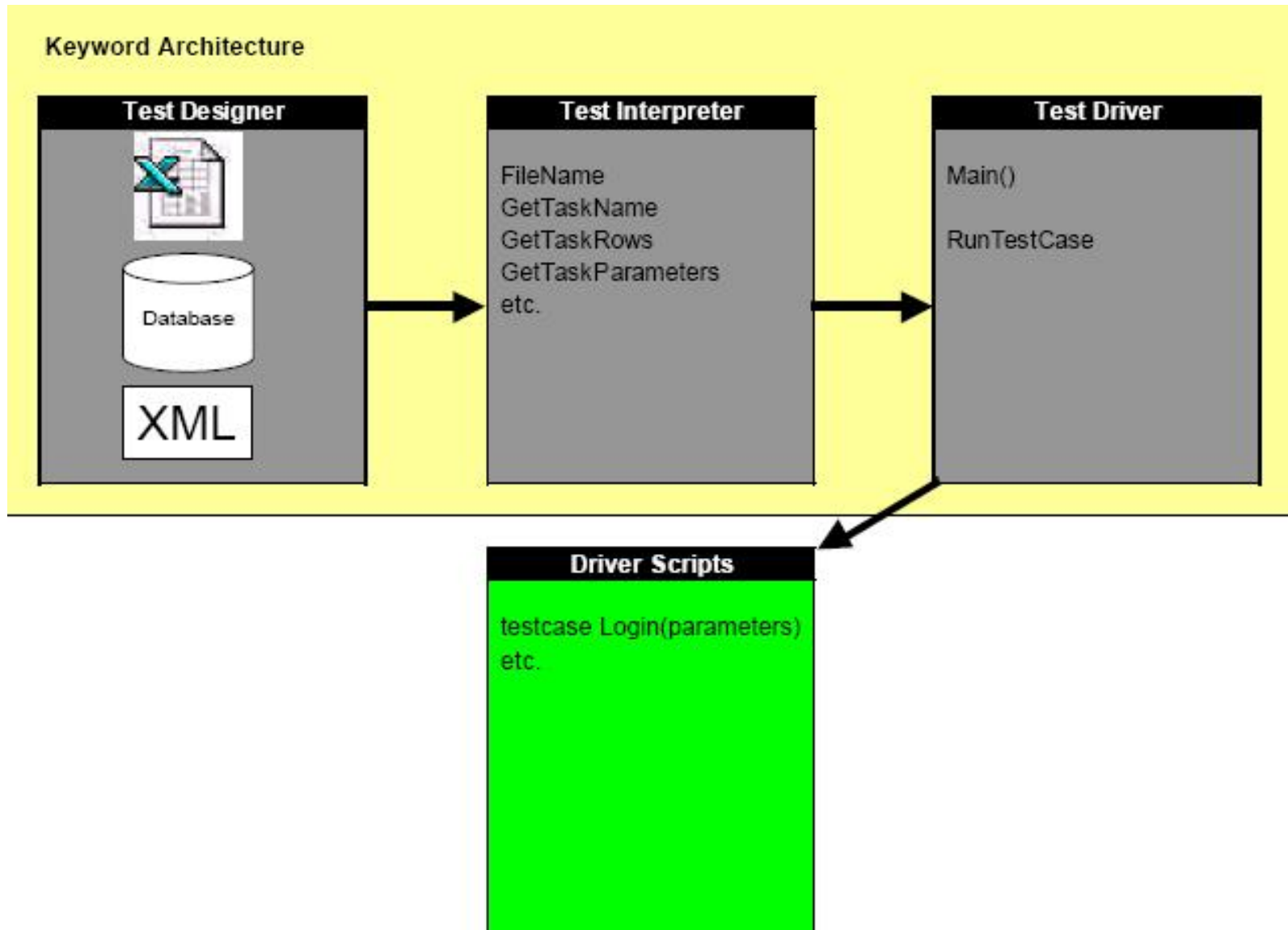
- Research possible solutions and evaluate options
- Develop requirements
- Develop the architecture
- Build the framework
- Test the framework
- Develop documentation
- Deploy the framework on a real project
- Measure performance and refine the design

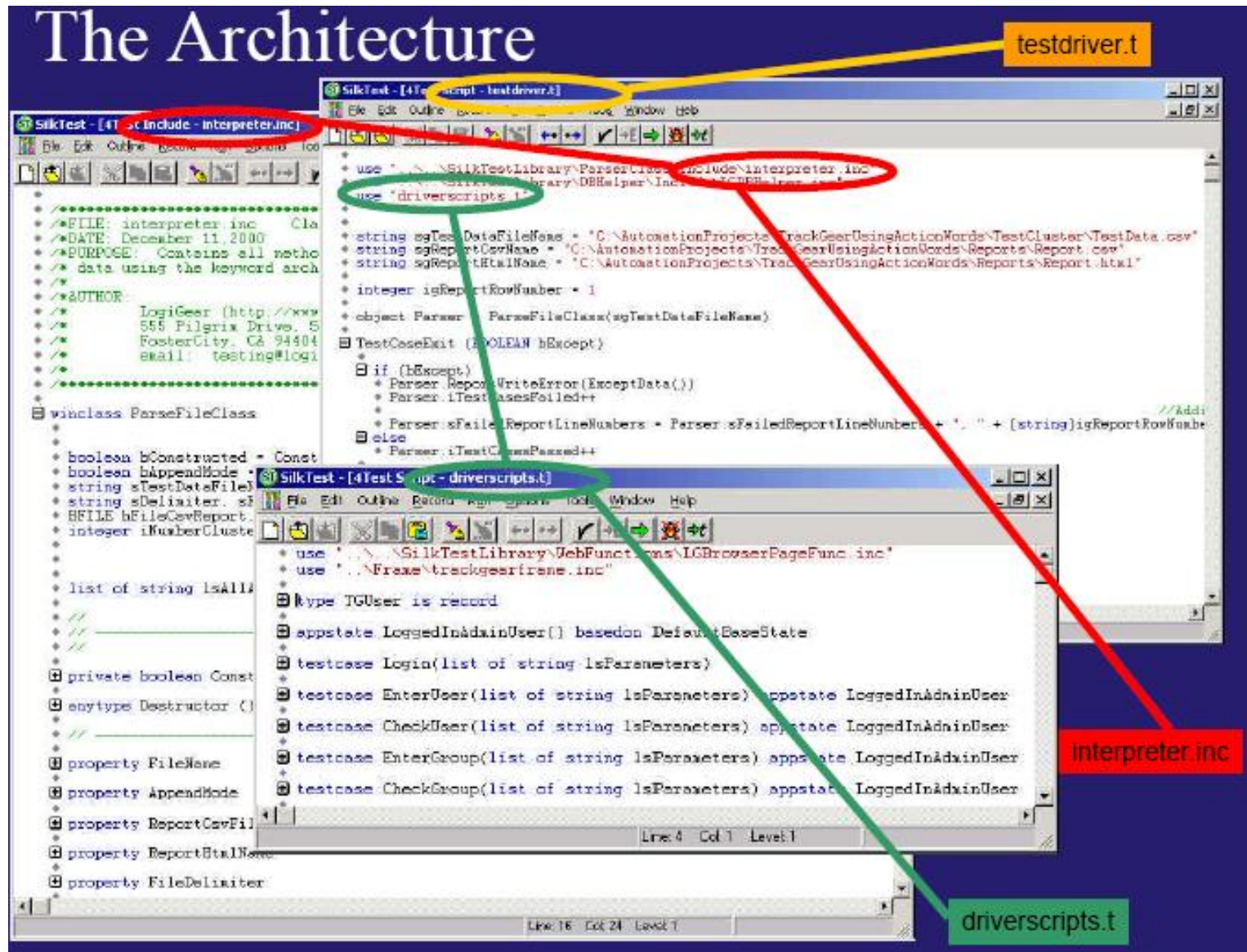
- Learn from past experience
- Discuss possibilities with software developers
- Talk to friends
- Read books
 - Recommend “*Software Test Automation*” by Graham and Fewster, 1999, Addison-Wesley
- Use the Internet
 - Recommend www.QACity.com, the Automated Testing page

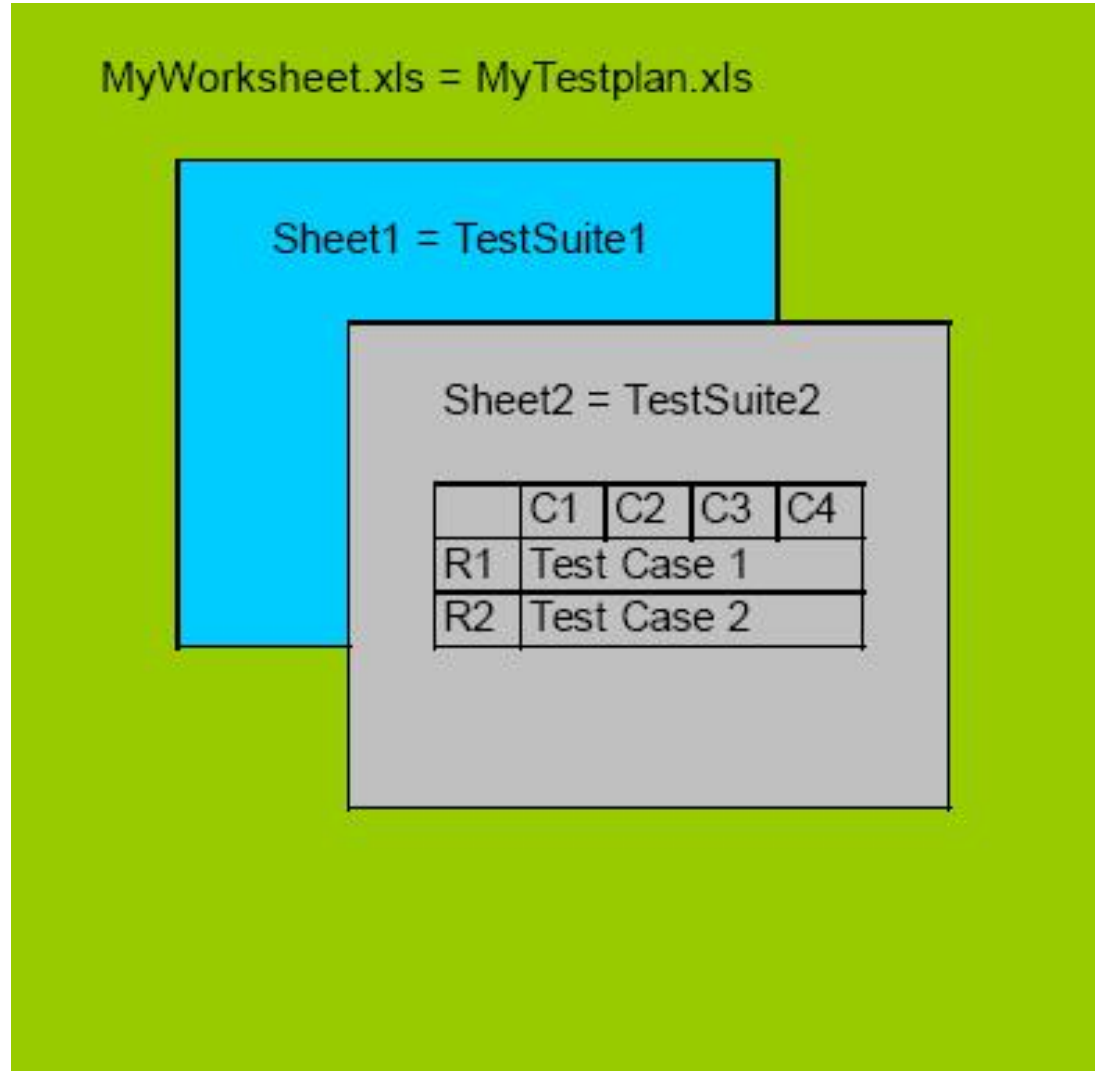
- Clearly state the business, people/process, and technology objectives
- Set expectations through well defined deliverables (e.g., requirement and design documents, code modules, whitepapers, training materials, etc.)
- Clearly define ways to measure success (e.g. quality of the design and code, budget, schedule, customer approval upon deployment, etc.)

Application Independent
Tool/Application Dependent









Test Plan

Test Suite

Microsoft Excel - TestPlan

	A	B	C	D	E	F	G	H	I	J
1	TestSuite		Administration Acceptance test							
2	Version		1							
3	Author		Joe Tester							
4										
5	section		1. Log in as Admin							
6	Task		User Id	Password	GroupMember					
7	LogIn		myID	myPassword	myID in Admin					
8										
9	section		2. Adding users							
10	Task		Username	First Name	Last Name	Password	Email	Phone	Pager	
11	EnterUser		joeQA	Joe	Tester	joeQA	joe_tester@qacity.com	(660) 565.1212 ext.100	(NONE)	
12	EnterUser		joeDev	Joe	Tester	joeDev	joe_tester@qacity.com	(660) 565.1212 ext.101	(NONE)	
13	EnterUser		joeProj	Joe	Tester	joeProj	joe_tester@qacity.com	(660) 565.1212 ext.102	(NONE)	
14	EnterUser		joeAdmin	Joe	Tester	joeAdmin	joe_tester@qacity.com	(660) 565.1212 ext.103	(NONE)	
15										
16	section		3. Adding groups							
17	Task		Group							
18	EnterGroup		Marketing							
19	EnterGroup		Engineering							
20	EnterGroup		Sales							
21										
22	section		4. Checking Users and Groups							
23	Task		Username	First Name	Last Name	Password	Email	Phone	Pager	
24	CheckUser		joeQA	Joe	Tester	joeQA	joe_tester@qacity.com	(660) 565.1212 ext.100	(NONE)	
25	CheckUser		joeDev	Joe	Tester	joeDev	joe_tester@qacity.com	(660) 565.1212 ext.101	(NONE)	
26	CheckUser		joeProj	Joe	Tester	joeProj	joe_tester@qacity.com	(660) 565.1212 ext.102	(NONE)	
27	CheckUser		joeAdmin	Joe	Tester	joeAdmin	joe_tester@qacity.com	(660) 565.1212 ext.103	(NONE)	
28										
29	Task		Group							
30	CheckGroup		Marketing							
31	CheckGroup		Engineering							
32	CheckGroup		Sales							
33										

Microsoft Excel - TestPlan

The Architecture: The Report

```

Microsoft Word - Report.doc
File Edit View Insert Format Tools Table Window Help
-----
TEST AUTOMATION REPORT
TestSuite: Administration Acceptance test
Version: 1
Author: Joe Tester
File: C:\Automation\Projects\TrackGearUsage\Report\Words\TestCluster\TestPlan.doc

Script name: Test-Automation Driver Script for TrackGear/Data
Script version: 1.0
Script release date: January 2001
Run Date and time: 03/03/01,12:12:29
-----
SECTION 1. log in as Admin
1(7): ,login,,myID,,my@domain,,myID in Admin,,,

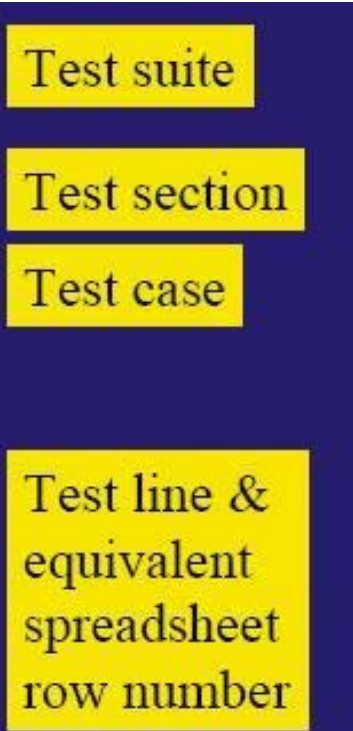
SECTION 2. Adding users
2(11): ,InterUser,,joeQA,,Joe,Tester,,joeQA,,joe@qacuity.com,,(650) 555.1212.wab.100,(NONE)
3(12): ,InterUser,,joeDev,,Joe,Tester,,joeDev,,joe@qacuity.com,,(650) 555.1212.ext.101,(NONE)
4(13): ,InterUser,,joeProj,,Joe,Tester,,joeProj,,joe@qacuity.com,,(650) 555.1212.wab.102,(NONE)
5(14): ,InterUser,,joeAdmin,,Joe,Tester,,joeAdmin,,joe@qacuity.com,,(650) 555.1212.ext.103,(NONE)

SECTION 3. Adding groups
6(18): ,InterGroup,,Marketing,,,,,
7(19): ,InterGroup,,Engineering,,,,,
8(20): ,InterGroup,,Sales,,,,,

SECTION 4. Checking Users and Groups
9(24):
10(25): ,CheckUser,,joeDev,,Joe,Tester,,joeDev,,joe@qacuity.com,,(650) 555.1212.wab.101,(NONE)
11(26): ,CheckUser,,joeProj,,Joe,Tester,,joeProj,,joe@qacuity.com,,(650) 555.1212.ext.102,(NONE)
12(27): ,CheckUser,,joeAdmin,,Joe,Tester,,joeAdmin,,joe@qacuity.com,,(650) 555.1212.wab.103,(NONE)
13(30): ,CheckGroup,,Marketing,,,,,
14(31): ,CheckGroup,,Engineering,,,,,
15(32): ,CheckGroup,,Sales,,,,,

-----
end of test suite: Administration Acceptance test
finished at: 03/03/01,12:25:28
time used: 0:13:15
number of cluster lines: 32
number of test lines: 15
number of testcases passed: 15
number of testcases failed: 0
percentage passed: 100%
failed at report lines: none
-----
Page 1 Sec 1 1/1 At 9:57 In 46 Col 95

```



- Prototype the components
- Implement the Test Designer
- Implement the Test Interpreter
- Implement AUT specific Test Drivers
- Add the reporting function to the Test Interpreter
- Test, fix bugs, and write documentation

- Clear requirements help focus the team on the important issues.
- Leaving “Fill in the blank” sections in requirements is manageable.
- Spending time on designing and prototyping helps flush out design issues; making it more scaleable, and helping write more maintainable code.
- If the project is overly complex and the schedule is aggressive, you may need to scale back. Don't forget to communicate changes in your plan.

- The necessary information is available! We need a way to find and analyze relevant information more quickly and effectively.
- Thoroughly research your options. Choose your designs wisely by taking business issues, people and process issues, and technology issues into consideration.
- Keep in mind that your solution might be used by one group, and maintained by another group.

- Your effort is a serious development project. Treat it as such: The key to success is good planning, scheduling and budgeting.
- Get feedback! How else can you learn?
- It won't be perfect! It's acceptable to learn from mistakes and refine the design as you go. Iteration and hard work make perfection.
- The keyword approach *works!*

Special thanks to Hans Buwalda for sharing his experience and vision on the action-word approach to creating test automation framework.

About Hung Q. Nguyen

Hung Q. Nguyen is Founder, President and CEO of LogiGear Corporation, Silicon Valley software testing company whose mission is to help software development organizations deliver the highest quality products possible while juggling limited resources and schedule constraints. LogiGear offers many value-added services including application testing, automated testing and web load/performance testing for e-business and consumer applications. Nguyen's company produces and markets TRACKGEAR™, a web-based defect tracking system. LogiGear also specializes in Web application, handheld communication device and consumer electronic product testing, and offers the software development community a comprehensive "Practical Software Testing Training Series." In the past two decades, Nguyen has held leadership roles in business development, engineering, quality assurance, testing, product development, and information technology. Nguyen is the author of Testing Applications on the Web (Wiley) and co-author of the best-selling book, Testing Computer Software (Wiley). He also develops and teaches software testing courses for UC Berkeley and UC Santa Cruz Extension, and for LogiGear. He holds a Bachelor of Science in Quality Assurance from Cogswell Polytechnical College, and is an ASQ-Certified Quality Engineer and active senior member of American Society for Quality.

LogiGear® Corporation is a full service software qualityengineering firm that provides testing expertise and resources to software development organizations. Some of our value-added services include application testing, automated testing, and web load/performance testing for e-business and consumer applications. LogiGear specializes in Web application, hand-held communication device, and consumer electronic product testing. LogiGear also produces and markets TRACKGEAR™, a Webbased defect-tracking solution, and offers QA Training through the Practical Software Testing Training Series.

www.logigear.com