Keep Automation Agile

Hans Buwalda

1

1

Who is your speaker

LogiGear Corporation

- Software testing company, around since 1994
  - US, Japan, Vietnam
- Testing and test automation services:
  - consultancy, training
  - test development and automation services
  - "test integrated" development services
- Products:
  - TestArchitect™, TestArchitect for Visual Studio™
  - integrating test development with test management and automation
  - based on modularized keyword-driven testing
- LogiGear Magazine:
  - themed issues, non-commercial

www.logigear.com
www.testarchitect.com

Team

- Hans Buwalda -- CTO
- Van Pham -- Support Manager
- Thanh Le -- Automation Engineer, Project Manager

hans @ logigear.com
vanp @ logigear.com
thanh.le @ logigear.com
www.happytester.com

2

Often hard to keep up for QA

LogiGear

SCRUM PROJECT

| Sprint | Sprint | Sprint | S |

PO ————→ PO ————→ PO ————→ PO —

Dev ————→ Dev ————→ Dev ————→ Dev —

QA ————→ QA ————→ QA ————→
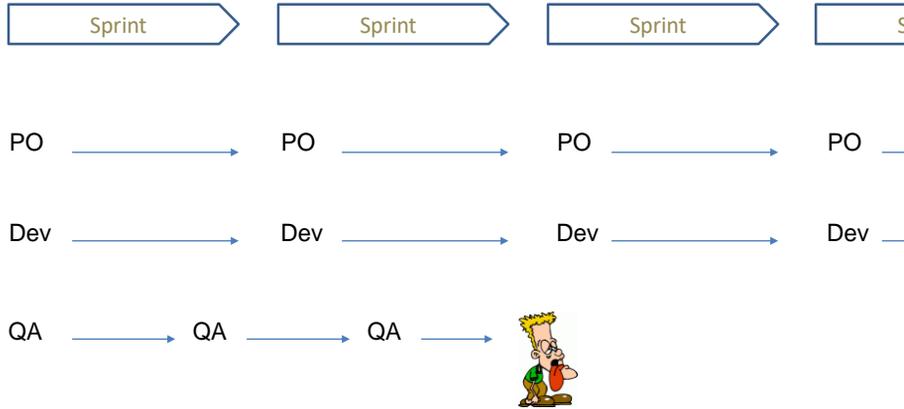
3

Some test kinds and their scalability (simplified)

LogiGear

	Relation to code	Quality / depth	Automation	Scalability
Unit Testing	Close relationship with the code	Singular test scope, but deep into the code	Fully automated by nature	Scalable, grows with the code, easy to repeat
Functional Testing	Usually does not have a one-on-one relation with code	Quality and scope depends on test design	In particular UI based automation can be a challenge	Often a bottle-neck in scalability
Exploratory Testing	Human driven, not seeking a relation with code	Usually deep and thorough, good at finding problems	May or may not be automated afterwards	Not meant to be repeatable. Rather do a new session

4

Some key items in agile and scalable (automated) testing

LogiGear

- Agreement, commitment, cooperation
- Expertise: domain, testing, automation, technical, ...
- Organization and design of tests
- Testability of the application under test
- Project and production focus
  – make it a project to have good tests and automation
  – take into account it has to perform in the pipeline
- Technology, tooling (but don't drown in it)
- Infrastructure, capacity
  – includes VM's, containers, cloud
- Make good tests, that find issues . . .

5

Don't just automate manual testing

LogiGear

6
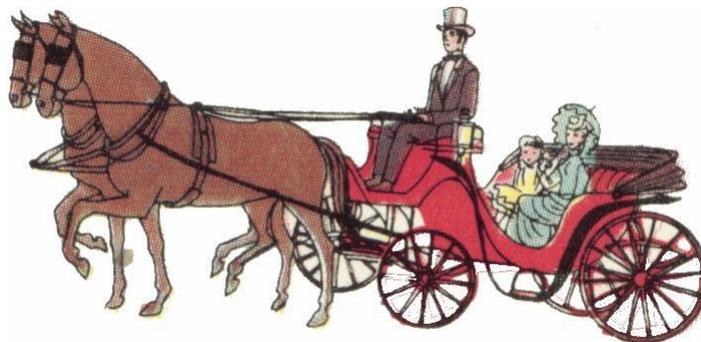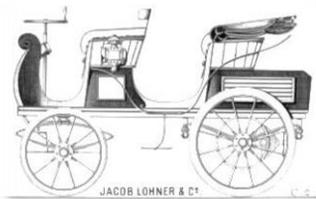
Don't just automate manual testing   *Logi*Gear

7

Don't just automate manual testing   *Logi*Gear

Good automated testing is not the same as automating good manual testing. . .

8

Why test design can matter?
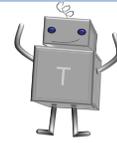
1 - Quality
will be better

- There are coding bugs: mishandling of expected situations
- And "jungle" bugs: mishandling of unexpected situations
- A good design can help identify more situations

9

Issues are not always obvious...
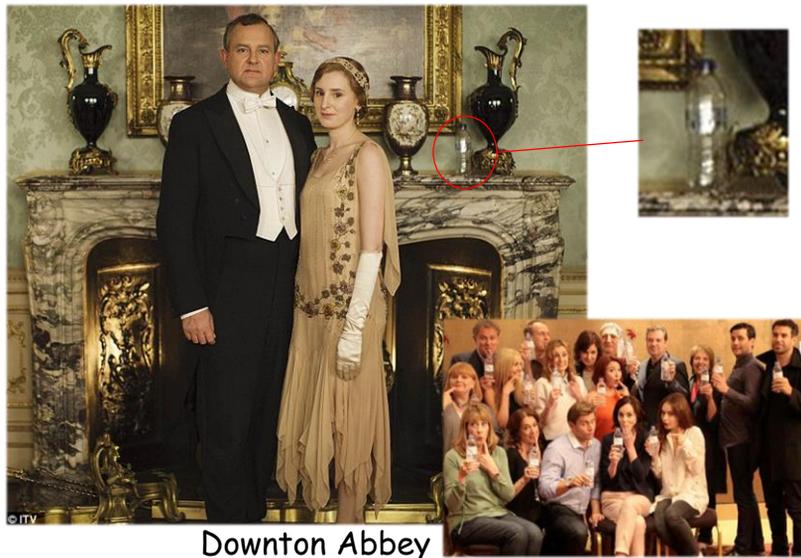
Downton Abbey

10

Surprising mistake

11

Why test design can matter?

1 - Quality
will be better

2 - Automation
will be easier

???
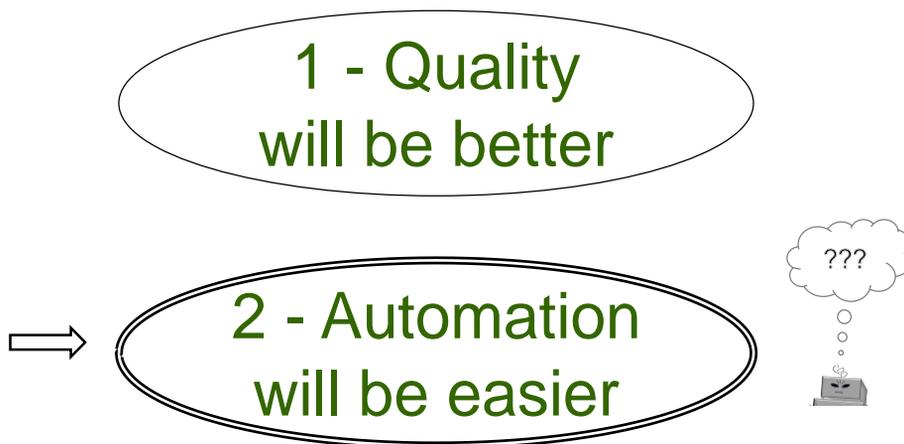
12

## Example of tests that are difficult to automate

Description	Expected
Open http://www.bigstore.com	The "BIG Store" main page is displayed, with a "sign in" link
Click on "Sign In", upper right corner	A sign in dialog shows, the "Sign in" button is disabled
Click on the "Search" button	5 watches of Acme Corporation are displayed
Double click on "Acme Super Watch 2"	The details page of the Acme Super Watch 2 is displayed
Verify the picture of the watch	The picture should show a black Acme Super Watch 2
Select "red" in the "Color" dropdown list	The picture now shows a red Acme Super Watch 2
Type 2 in the "Order quantity" textbox	The price in the right shows "$79.00 + Free Shipping"
Click on "Add to cart"	The status panel shows "Acme Super Watch 2" added
Click on "Check out"	The Cart Check Out open, with the 2 Acme Super Watches
Click on "Add to cart"	The status panel shows "Acme Super Watch 2" added
Click on "Check out"	The Cart Check Out open, with the 2 Acme Super Watches
etc	etc

13

Action Based Testing

- Modularized keyword-driven method for automated testing

- Particularly suited for large and complex projects

- High degree of agility, even when used in traditional projects

- High level of automation, 95% or more

- Puts the focus on test design, less on technology

14

Actions

fragment from a test with actions

	acc nr	first	last
open account	123123	John	Doe

	acc nr	amount
deposit	123123	10.11
deposit	123123	20.22

	acc nr	expected
check balance	123123	30.33

4 actions, each with an action keyword and arguments

read from top to bottom
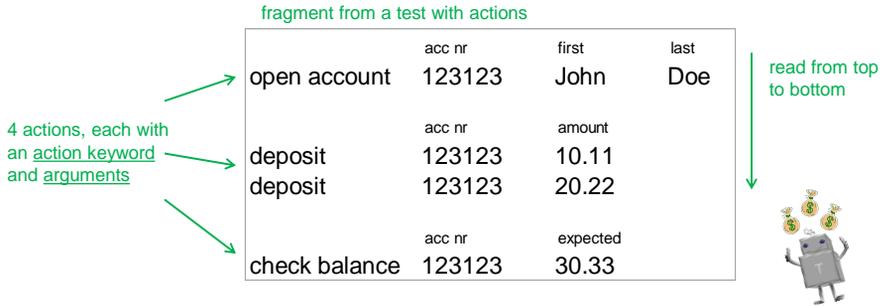
- The test developer creates tests using actions with keywords and arguments
- The automation task focuses on automating the keywords, each keyword is automated only once
- This format is easy to convert from and to BDD, but generally more manageable than BDD

15

"Test Module"

TEST MODULE	Car Rental Reservation			
OBJECTIVES				
test objective	TO 01	Invoice total is calculated correctly		
test objective	TO 02	Number of currently available cars is updated after a rental reservation		
test objective	TO 03	An error message is displayed when user tries to rent unavailable car		
INITIAL	Setting up			
config				
start application				
login				
TEST CASE	TC 01	Check invoice total for a single reservation		
test objective	TO 01	Invoice total is calculated correctly		
	first name	last name	car	rental option
rent car	John	Doe	Compact/Toyota Prius	Infant Seat ($10.0/day)
	last name	total charge		
check total payment	Doe	140.40		
TEST CASE	TC 02	Check invoice total for multiple reservations		
TEST CASE	TC 03	Check number of currently available cars is updated		
FINAL	Cleaning up			
	current window			
back to home	car selection			
	window			
close application	welcome			

Objectives

Initial - setup

Test cases

Final - cleanup

16

Action Based Testing

LogiGear

| High Level Test Design - Test Development Plan | ← define the "chapters" |

```
Test Module 1        Test Module 2              Test Module N
  Objectives           Objectives         ...     Objectives      ← create the "chapters"
  Test Cases           Test Cases                 Test Cases
```

Actions ← create the "words"

AUTOMATION ← make the words work

interaction test

	window	control	value	
enter	log in	user name	jdoe	
enter	log in	password	car guy	
	window	control	property	expected
check property	log in	ok button	enabled	true

business test

	user	password		
log in	jdoe	car guy		
	first	last	brand	model
enter rental	Mary	Renter	Ford	Escape
	last	total		
check bill	Renter	140.42		

17

My view on test automation . . .

LogiGear

Successful automation is not

as much a technical challenge,

as it is a test design challenge.

18

Paradox

*Logi*Gear

Test design is important for automation

but . . .

many testers are not engineers
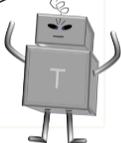
19

What's the trick...

*Logi*Gear

20

What's the trick...

- Have or acquire facilities to store and organize your content

- Select your stuff

- Decide where to put what
  - assign and label the shelves

- Put it there

- If the organization is not sufficient anymore, add to it or change it

21

"Pre-structure"

(1) Early in the project: identify test modules

QA, Product Owners, Dev, Stakeholders, Domain Experts, . . .

Price Calculations

Login Tests

Address Entry

label the (empty) shelves
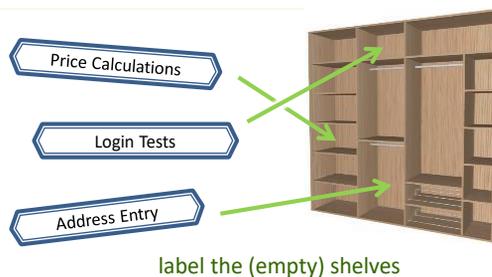
(2) During sprints: develop the tests into the modules

use actions and checks that fit the scope of the module

avoid unnecessary details, hide them in the actions

adjust the pre-structure whenever needed, the team owns it

22

Example Top Level Structure

```
Project
 ├── Business Objects
 │    ├── "Cars"
 │    │    ├── Business, life cycles
 │    │    └── Interaction
 │    └── "Customers"
 ├── Business Flows
 ├── architectures, technologies
 ├── Functions and Features
 ├── Data (handling, quality, ETL, ...)
 ├── Security, authorization, admin
 ├── Graphics, multi-media, charts, ...
 ├── Customizing, extensibility
 ├── Interoperability
 └── Concurrency, race conditions, ...
```

create, update, delete/close
copy, move
categorize, enumerate, identify
convert, serialize, export/import, ...

UI, dialogs, forms, pages
input (validation, defaulting, dependencies)
flows (primary paths, alternate paths)
keyboard shortcuts, keyboard controls, ...

processes, transactions, end-to-end, day in the life,
combinations of flows, ...

calculations, analyses, PDF output, ...

23

Example: E-commerce site

- Articles
- Categories
- Customers
- Promotions
- Orders
- Invoices
- Payments
- Credit cards
- Staff members
- Countries
- . . .

24

What tests could look like

LogiGear

Business tests for business object "Promotions":

create a promotion, from 12/1 to 12/25, 11% discount on all inventory
create a promotion, from 12/20 to 12/31, 20% discount on tablets

set the date to 12/23

check the discounted price on an iPad Mini 4

Interaction tests for business object "Promotions":

click on the "New Promotion" button in the main screen

select "state specific" as the type of promotion

in the list of states check if all US states are listed

25

What tests could look like

LogiGear

Business test

	name	start	finish	percentage	category
create promotion	christmas	12/1/2016	12/25/2016	11	all
create promotion	tablets	12/20/2016	1/1/2017	20	tablets
	date				
time travel	12/23/2016				
	article	price			
check nett price	iPad Mini 4	284.79			

Interaction test

	window	control	
click	main	new promotion	
	window	type	
select	promotion	state	
	window	list	value
check list item exist	promotion	state	American Samoa

26

Testability, why it matters

LogiGear

- Testability is a major contributor to:
  - testing success, in particular completeness
  - automation success

- Developers count:
  - "small step for a dev, giant leap for QA"
  - next are the product owners
  - team owns quality and testability

- Main topics for testability:
  - design of the application
  - clarity of functionality, domain, user stories, etc
  - interfacing, in particular UI
  - service and API access
  - emulation

- Consider testability as a "must have" requirement
  - first question in a development project: "how do we test this?"

Small step for Dev
Giant leap for QA

27

## Identifying controls
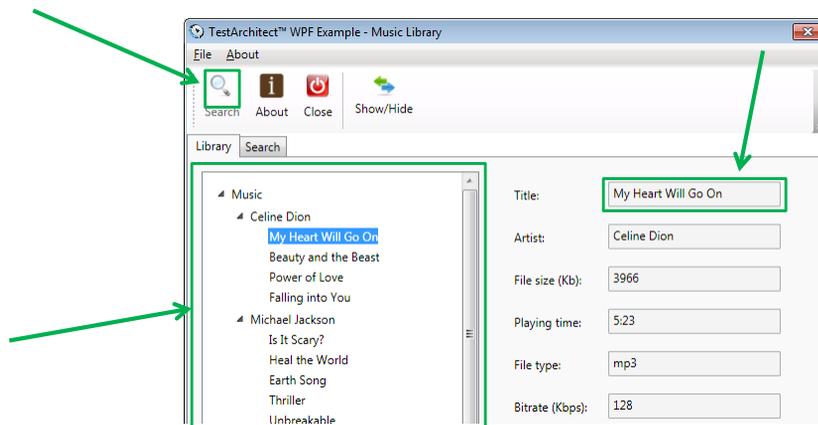
LogiGear

- Identify windows and controls, and assign names to them
- These names encapsulate the properties that the tool can use to identify the windows and controls when executing the tests

28

Mapping an interface

LogiGear

name	class	label		
title	text	Title:	→	Title: My Heart Will Go On
artist	text	Artist:	→	Artist: Celine Dion
file size	text	File size (Kb):	→	File size (Kb): 3966

name	class	position		
playing time	text	textbox 4	→	Playing time: 5:23
file type	text	textbox 5	→	File type: mp3
bitrate	text	textbox 6	→	Bitrate (Kbps): 128
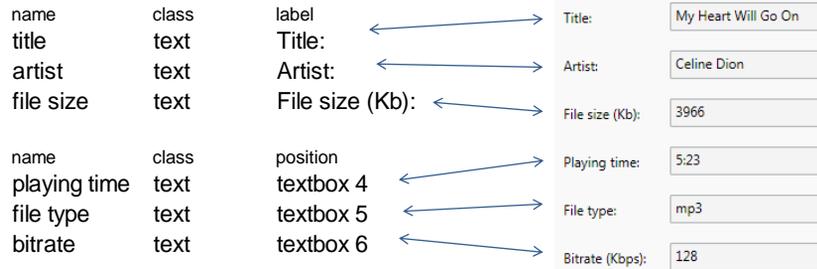
- An interface mapping (common in test tools) will map windows and controls to names
- When the interface of an application changes, you only have to update this in one place
- The interface mapping is a key step in your automation success, allocate time to design it well, in particular naming and choosing identifying properties

29

Hidden interface properties

LogiGear

- Use properties a human user can't see, but a test tool can
  - less need for "spy" tools (which take a lot of time)
  - less sensitive to changes in the system under test
  - not sensitive to languages and localizations

- Examples:
  - "id" or other attributes for HTML elements (also simplifies xpath expressions)
  - "name" field for Java controls
  - "AccessibleName" or "Automation ID" properties in .Net controls (see below)
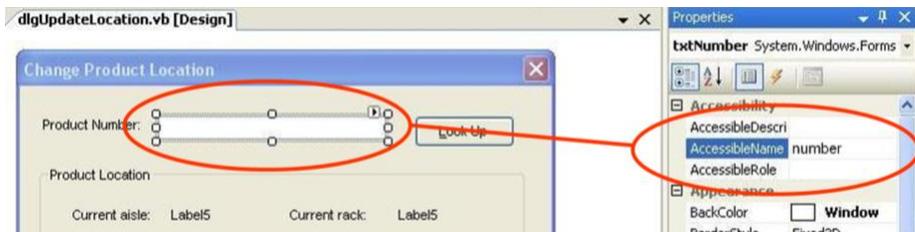
30

Active Timing

- Passive timing
  - wait a set amount of time
  - in large scale testing, try to avoid passive timing altogether:
    - if wait too short, test will be interrupted
    - if wait too long, time is wasted

- Active timing
  - wait for a measurable event
  - usually the wait is up to a, generous, maximum time
  - common example: wait for a table to fill or a window to appear

- Even if not obvious, find something to wait for...

- Involve developers if needed
  - relatively easy in an agile team, but also in traditional projects, give this priority

- If using a waiting loop
  - make sure to use a "sleep" function in each cycle that frees up the processor (giving the AUT time to respond)
  - wait for an end time, rather then a set amount of cycles

- Good news: tools and frameworks have good wait features built in

31

White-box access

- Input data
  - ability to emulate data values

- Output data
  - what is the underlying data being displayed in chart etc

- States, objects
  - like in a game, "where are the monsters"

- Random generators
  - can I switch them off, or set the seed?

- Emulation, mock-up
  - service replies
  - GPS and time travel

- . . .

32

Same sprint automation?

LogiGear

- Commitment
- Agreement on the approach
- Tools and technologies in place
  - and well understood in the team
- Testable code
  - well structured, clear purpose
  - generous automation access
- Know what and how to test
  - test modules with clear scopes
  - domain expertise (in or near the team)
  - testing techniques
- Have a strategy for data and configurations
- Get help if needed, stay in sync

33

ABT in Agile

LogiGear

Agile life cycle

| product owner | team | prod owner & team |

| Product Backlog | ⇨ | Sprint | ⇨ | Products |

Test Module Definition (optional)	Test Module Development	Test re-use
	Interface Definition	Automation re-use
	Action Automation	
	Test Execution	

Test development

User stories Documentation Domain understanding	→	Acceptance Criteria PO Questions Situations Relations	→	Main Level Test Modules
			→	Interaction Test Modules
			→	Cross over Test Modules
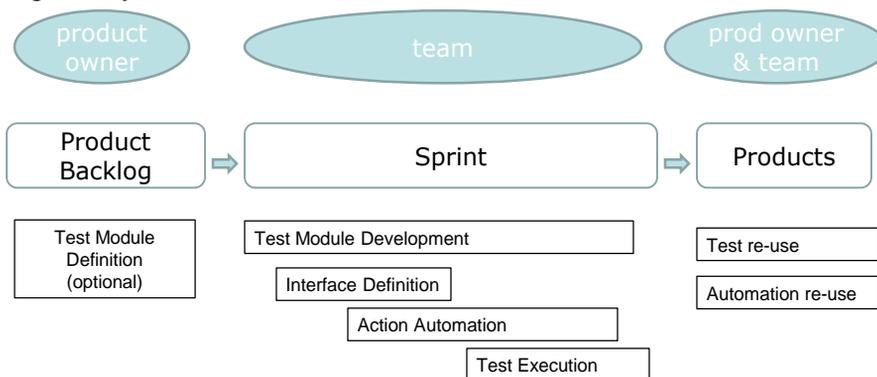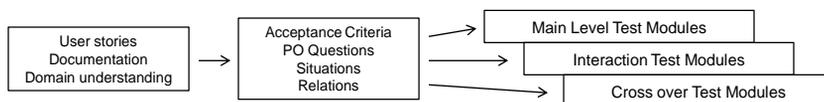
34

Sit-down meeting?

- An end-of-day counterpart to the early-morning "stand up" meetings

- Short and friendly, have a cup of tea

- Not about progress and impediments, but about practices and experiences with them (like "what actions did you use?")

- The team can invite an experienced person as moderator

- A few meetings may suffice

35

Leveraging tests and automation

- Soap Opera Testing

- Day in the Office testing

- Lead-Deputy testing

- Action based service emulation

- Continuous variation

- Random regression

- Transparent cross-platform testing

36

Summary

LogiGear

- Automation is a key requirement in agile and CI/CD

- Key success factors are test design, testability and team work

- Action Based Testing helps with a modular keyword driven approach

- Same-sprint automation is achievable in most cases, but needs commitment and an eye for a good process and technology

37