Cytel
STATISTICAL SOFTWARE & SERVICES

Data-driven Software Testing for Agile Development

Priyadarshan Shinde, Cytel
Ajay Sathe, Cytel

PhUSE 2011 –Brighton

Agenda

- Key Issues in Conventional Automated Testing
- Approach
- Concept
- Test Repository
- Test Repository Walkthrough
- Test Repository Advantages
- Workbench
- Advantages
- Implementation

Key Issues in Conventional Automated Testing

1. Software being *stable* is a mandatory requirement
2. Scripts tend to be hard-coded, making *maintenance* a non-trivial and sometimes complex task
3. Object names tend to be *long and unreadable*. For instance, it may be difficult to see which control has the following name:

```
Sys.Process('notepad').Window('#32770', 'Font', 1).Window('ComboBox',
'', 3).Window('Edit', '', 1)).
```

4. Object name depends on object properties that may *not be persistent*
5. Operating System (OS) *compatibility*
6. Differences in *coding styles* of different testers
7. Tester needs to be fairly *good at programming* with the test script

Approach

SiZ®

East®

Labels

Textboxes

Dropdown List

Buttons

Grid

Plots

ANOVA: One-way

Design Parameters | Anova Parameters | Add to Favorites

Design Name: Design

Type - 1 Error (α):

Power (1-β): *

Sample Size (n):

Effect Size ($\Delta=\sigma_m^2/\sigma^2$):

Number of Groups (R): 2

Common Std. Dev. (σ):

Variance of Means (σ_m^2):

* Value to be Computed: Power

Compute | Clear

Boundary Family: Spending Functions

Lower Efficacy Boundary

Spending Function: Lan-DeMets

Parameter: OF

Lower α: 0.025

Upper Efficacy Boundary

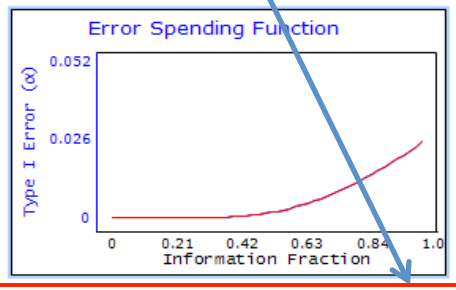
Spending Function: Lan-DeMets

Parameter: OF

Upper α: 0.025

Total Type I Error (α): 0.05

Look #	Info. Fraction	Lower Alpha	Upper Alpha
1	0.5000	0.002	0.002
2	1.0000	0.025	0.025



Compute

Concept

Focus is on Control handling

Action

Value

Control

```
Set txtbox = Sys.Process("Architect").FindChild("Name","Type1Error")
If txtbox.Exists = True Then
    txtbox.Set_Text("0.05")
Else
   Log.Error "Type 1 Error textbox was not found."
End If
```

Test Repository

Test Scenario → List of Controls → Test Cases → Modular Structure

Design Parameters | Look Settings

Error Rates

Trial:	Superiority	**Input Method:**	Difference of Means
Side:	2-Sided		
Type I Error (α):	0.05	**Difference in Means ($\mu_t - \mu_c$):**	0.3
		Std. Deviation (σ):	1
Power (1-β):	0.9		
Sample Size (N):	Compute		
Allocation Ratio (N_t / N_c):	1		

Compute

Test Repository

Test Scenario → List of Controls → Test Cases → Modular Structure

Design Parameters | Look Settings

Error Rates

Trial:	Superiority	**Input Method:**	Difference of Means
Side:	2-Sided		
Type I Error (α):	0.05	**Difference in Means ($\mu_t - \mu_c$):**	0.3
		Std. Deviation (σ):	1
Power (1-β):	0.9		
Sample Size (N):	Compute		
Allocation Ratio (N_t / N_c):	1		

Compute

Test Repository

Test Scenario → List of Controls → Test Cases → Modular Structure

1. Click on Design Parameter tab
2. Trial dropdown list, select “Superiority”
3. Specify Power = 0.9
4. Click on Compute button
5. Verify results

Test Case Step 1				Test Case Step 2			
Control	**Action**	**Value**	**Property Validation**	**Control**	**Action**	**Value**	**Property Validation**
Design Parameter Tab	Click	NA	NA	Trial	Select Item	Superiority	Superiority

Test Repository

Test Scenario → List of Controls → Test Cases → Modular Structure

Detail Scenario Flow					
Scenario ID	Modules	Control Actions & Test Data			Comment
		On Sheet Name	Execute From row	Upto row	
1	Menu bar	Menu Bar	10	10	Insert Design - Difference of Means for Independent Data
	Design Parameter	Design Parameter	15	37	Test Cases for Single Field Validation type of Testing
2	Menu bar	Menu Bar	10	10	Insert Design - Difference of Means for Independent Data
	Design Parameter	Design Parameter	40	65	Test cases for Combination of Fields Validation Testing
3	Menu bar	Menu Bar	10	10	Insert Design - Difference of Means for Independent Data
	Look Settings	Look Settings	15	20	Test Cases for Single Field Validation type of Testing
4	Menu bar	Menu Bar	10	10	Insert Design - Difference of Means for Independent Data
	Look Settings	Look Settings	25	30	Test Cases for Combination of Fields Validation Testing
5	Menu bar	Menu Bar	10	10	Insert Design - Difference of Means for Independent Data
	Design Parameter	Design Parameter	70	83	Test Cases for Dialog Navigation Testing
	Look Settings	Look Settings	35	50	

Test Repository Walkthrough

AutoScriptFramework_Exl2003 (version 1) [Compatibility Mode] - Microsoft Excel

Home | Insert | Page Layout | Formulas | Data | Review | View | Load Test | Team

C51

Test Script Framework East 6 — Cytel

Process
Executed on

Scenario
1
2
3
4
5
6
7
8

Test Scenarios List

Scenario ID	Description	Detail Scenario Flow: From Row	Detail Scenario Flow: To row
	Difference of Means for Independent Data Testing		
1	Single field validation of "Design Parameter" Tab	18	19
2	Combination of Field validation of "Design Parameter" tab	21	22
3	Single field validation of "Look Settings" Tab	24	25
4	Combination of Field validation of "Look Settings" tab	27	28
7	Sanity Test Scenarios	40	44	No
8	Regression Test Scenarios	46	50	No

	Difference of Means for Crossover Data Testing				
17	Single field validation of "Design Parameter" Tab	88	89	Yes	Pass
18	Combination of Field validation of "Design Parameter" tab	91	92	Yes	Pass
19	Single field validation of "Look Settings" Tab	94	95	Yes	Pass
20	Combination of Field validation of "Look Settings" tab	97	98	Yes	Pass
21	Dialog Navigation Testing	100	103	Yes	Pass
22	Application Flow Testing	105	109	Yes	Pass
23	Sanity Test Scenarios	111	115	Yes	Pass
24	Regression Test Scenarios	117	121	Yes	Pass

SOP | Main Sheet | Property Matrix | Message List | Single Field Validation | Combination of Field Validation | Menu Bar | Design | Look Sett

Ready

Detail Scenario Flow

Scenario ID	Modules	Control Actions & Test Data: On Sheet Name	Execute From row	Upto row	Comment
	Difference of Means for Independent Data Testing				
1	Menu bar	Menu Bar	11	11	Insert Design - Difference of Means for Independent Data
	Design Parameter	Design Parameter	15	37	Test Cases for Single Field Validation type of Testing
2	Menu bar	Menu Bar	11	11	Insert Design - Difference of Means for Independent Data
	Design Parameter	Design Parameter	40	65	Test cases for Combination of Fields Validation Testing
3	Menu bar	Menu Bar	11	11	
	Look Settings	Look Settings	15	20	
4	Menu bar	Menu Bar	11	11	
	Look Settings	Look Settings	25	30	
4	Menu bar	Menu Bar	17	17	Insert Design - Difference of Means for Crossover Data Testing
	Look Settings	Look Settings	25	30	Test Cases for Combination of Fields Validation Testing
5	Menu bar	Menu Bar	17	17	Insert Design - Difference of Means for Crossover Data Testing
	Design Parameter	Design Parameter	70	83	
	Look Settings	Look Settings	35	50	Test Cases for Dialog Navigation Testing
6	Menu bar	Menu Bar	17	17	Insert Design - Difference of Means for Crossover Data Testing
	Design Parameter	Design Parameter	70	83	
	Look Settings	Look Settings	35	50	
	Output Preview	Output Preview	15	24	Test Cases for Application Flow Testing
	Library	Library	17	25	
7	Menu bar	Menu Bar	17	17	Insert Design - Difference of Means for Crossover Data Testing
	Design Parameter	Design Parameter	70	75	
	Look Settings	Look Settings	35	40	
	Output Preview	Output Preview	15	20	Test Cases for Sanity Testing
	Library	Library	17	15	

Test Repository Walkthrough

Detail Scenario Flow				
Scenario ID	Modules	Control Actions & Test Data		
		On Sheet Name	Execute From row	Upto row
	Difference of Means for Independent Data Testing			
1	Menu bar	Menu Bar	11	11
	Design Parameter	Design Parameter	15	37

Single Field Validation | Combination of Field Validation | Menu Bar | Design Parameters | Look Settings | Simulation Settings | Keep In | Global Options

9	Comments	Control	Action	Value	PropMatrix	Control	Action	Value	PropMatrix
10		Test case for invoking : - Difference of Means for Independent Data Testing							
11		Menubar	ShortKey			Difference of	ShortKey		
12									

15	TabDesign	ClickTab	Number of Looks	SetValue	1	Side Type	ClickItem	1-Sided	Alpha	Keys	0.36
16	TabDesign	ClickTab	Number of Looks	SetValue	1	Side Type	ClickItem	1-Sided	Alpha	Keys	0.07
17	TabDesign	ClickTab	Number of Looks	SetValue	1	Side Type	ClickItem	1-Sided	Alpha	Keys	0.26
18	TabDesign	ClickTab	Number of Looks	SetValue	1	Side Type	ClickItem	1-Sided	Alpha	Keys	0.42
19	TabDesign	ClickTab	Number of Looks	SetValue	1	Side Type	ClickItem	1-Sided	Alpha	Keys	0.02
20	TabDesign	ClickTab	Number of Looks	SetValue	1	Side Type	ClickItem	1-Sided	Alpha	Keys	0.07
21	TabDesign	ClickTab	Number of Looks	SetValue	1	Side Type	ClickItem	1-Sided	Alpha	Keys	0.26

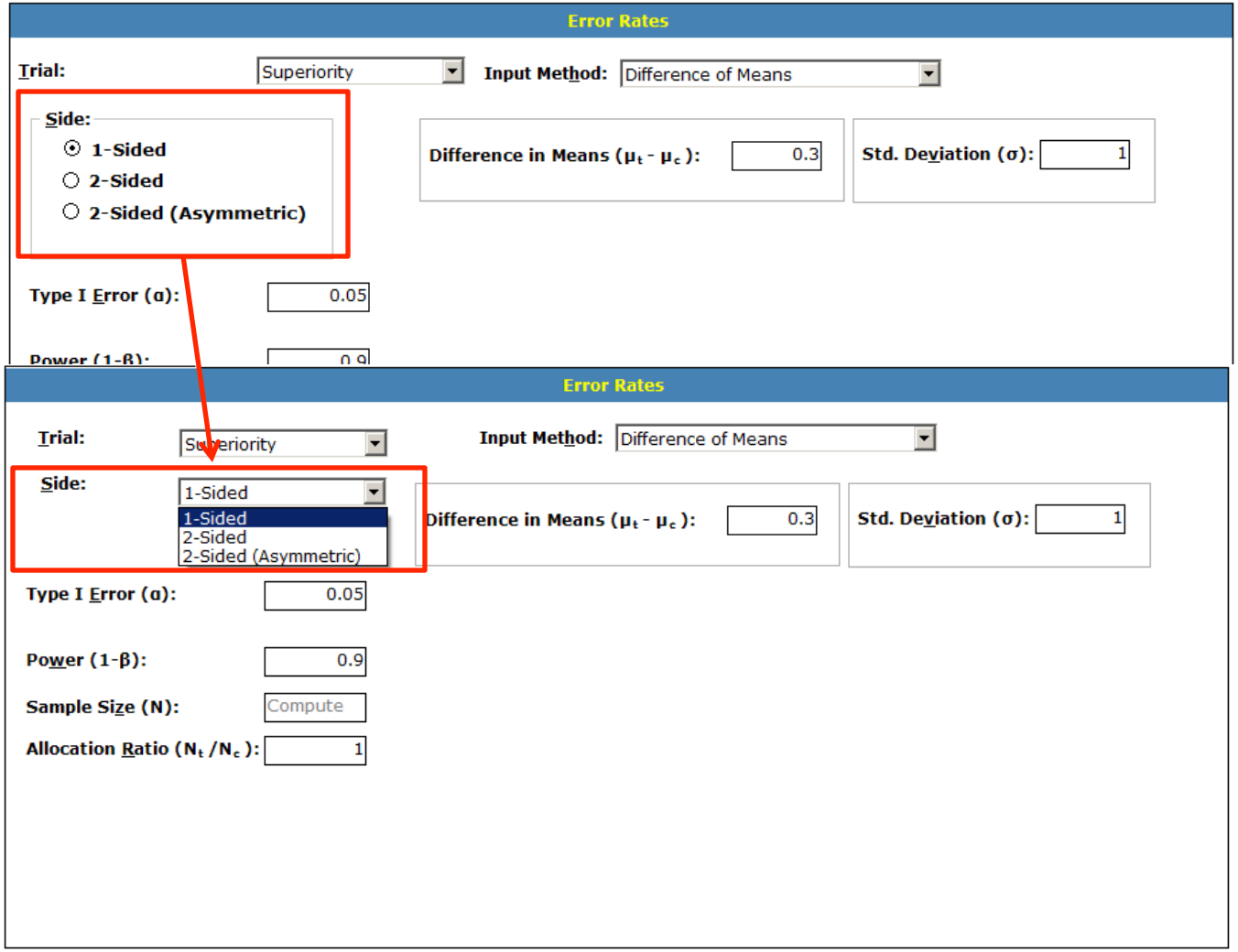
Test Repository Advantages

- Create test cases earlier
- Reusability of test cases within the software
- Reusability of test cases across different software
- Easy to handle the change requests

Test Repository Advantages

Requirement

Change Request



Test Repository Advantages

List of Controls

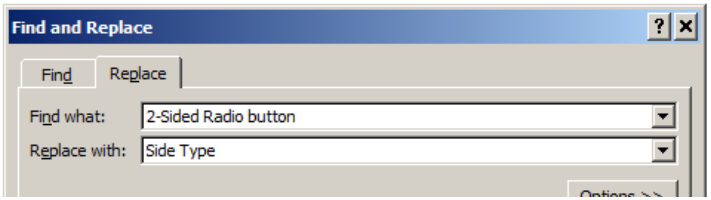
Control	Control Type	Name
1-Sided Radio button	RadioButton	rdbOneSided
2-Sided Radio button	RadioButton	rdbTwoSided
2-Sided (Asymmetric) Radio button	RadioButton	rdbTwoSidedAysm

Control	Control Type	Name
Side Type	Combo Box	cmdSideType

Test Repository Advantages

Test Cases

Control	Action	Value	PropMatrix
1-Sided Radio button	Click		
2-Sided Radio button	Click		
2-Sided (Asymmetric) Radio button	Click		



Control	Action	Value	PropMatrix
Side Type	Select Item	2-Sided	

East6	Design Parameters	B39	2-Sided Radio button
East6	Design Parameters	I46	2-Sided Radio button
East6	Design Parameters	O47	2-Sided Radio button
East6	Design Parameters	J49	2-Sided Radio button
East6	Design Parameters	I50	2-Sided Radio button
East6	Design Parameters	M51	2-Sided Radio button
East6	Design Parameters	I53	2-Sided Radio button
East6	Design Parameters	J53	2-Sided Radio button
East6	Design Parameters	Q55	2-Sided Radio button
East6	Design Parameters	I57	2-Sided Radio button
East6	Design Parameters	I68	2-Sided Radio button
East6	Design Parameters	Q68	2-Sided Radio button

22 cell(s) found

Workbench

Setting a Process

Test Repository

```
Set P = Sys.Process(AUTProcess)
```

Setting a Window Object

Test Repository

```
Set WndObject = Sys.Process(P).FindChild("Name",ControlName, 100)

If WndObject.Exists = True then
        WndObject.Activate
        Log.Message("Specified Window is exist and activated")
                :
                :
                :
Else
        Log.Error("Specified Window was not found")
End If
```

Workbench

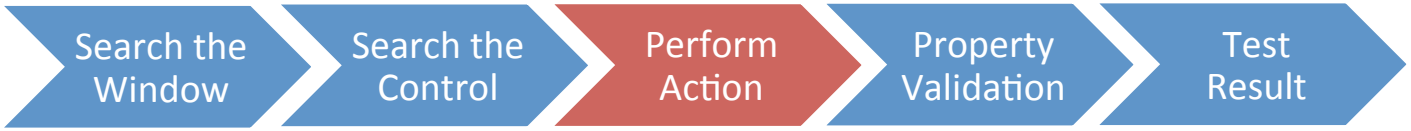
Finding a Control within the Window

```
WndObject.Refresh
WndObject.Activate

Set Control  = WndObject.FindChild("Name", ControlName , 100)
If Control.Exists = True Then
Control.Refresh
Control.Activate
            :
            :
```

Test Repository

Workbench



GUI Control Handling

```
If Control.Exists = True Then
   Select Case Action
      Case "Click"
            Control.Click

      Case "SelectItem"
            Control.ClickItem(ControlValue)
                  :
                  :
   End Select
Else
    Log.Error("Control does not exist:" & ControlName)
End If
```

Workbench

Search the Window → Search the Control → Perform Action → **Property Validation** → Test Result

Before Performing Action

Control	Property	Expected Value
Side		
Lower Label		
Upper Label		
Lower Type 1		
Upper Type 1		
Lower Type 1		
Upper Type 1		
Type 1 Error	visibleonscreen	FALSE

Side: 2-Sided

Type I Error (α): 0.05

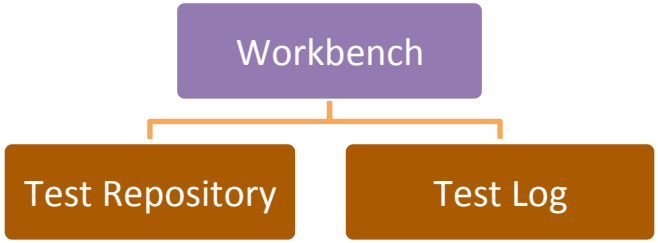
After Performing Action
Property validation

Side: 2-Sided (Asymmetric)

Type I Error (α): Lower 0.025 Upper 0.025

```
    Actu
If  Actu
    Log.                                ssed")
Else
    Log.Error("Property Validation failed")
End If
```

Workbench



Log Tree

Process Name	East
Executed on Build	01-Apr-11

Test Scenarios List

Scenario ID	Description	Detail Scenario Flow		Execute	Pass/ Fail
		From Row	To row		
1	Single field validation of "Design Parameter" Tab	17	18	Yes	Pass
2	Combination of Field validation of "Design Parameter" tab	20	21	No	
3	Single field validation of "Look Settings" Tab	23	24	Yes	Fail
4	Combination of Field validation of "Look Settings" tab	26	27	No	
5	Dialog Navigation Testing	29	31	No	
6	Application Flow Testing	33	58	No	

Workbench Advantages

- Portable Architecture Between Tools
- Minimized Maintenance
- No Tool Expertise Needed to Create Test Cases
- Drastic Reduction in Testing Time and Effort

Develop Test Cases, Scripts Earlier

Control	Control Type	TestComplete	Visual Studio 2010
Design Parameter Tab	Tab	[illegible]	[illegible]
Trial	Combo Box	WinFormsObject("cmbTrial")	cmbTrial
Input Method	Combo Box	WinFormsObject("cmbInputMethod")	cmbInputMethod
Side	Combo Box	WinFormsObject("cmbSideType")	cmbSideType
Type I Error	Textbox	WinFormsObject("txtType1Error")	txtType1Error
Power	Textbox	WinFormsObject("txtPower")	txtPower
Sample Size	Textbox	WinFormsObject("txtSampleSize")	txtSampleSize
Allocation Ratio	Textbox	WinFormsObject("txtAllocationRatio")	txtAllocationRatio
Difference in Means	Textbox	WinFormsObject("txtDiffinMeans")	txtDiffinMeans
Std Deviation	Textbox	WinFormsObject("txtStdDev")	txtStdDev
Compute Power	RadioButton	WinFormsObject("rbPower")	rbPower
Compute Sample Size	RadioButton	WinFormsObject("rbSampleSize")	rbSampleSize
Compute	Button	WinFormsObject("btnCompute")	btnCompute

Implementation

- *SiZ®*
- *Compass®*
- *Architect*
- *East®*

TestComplete™

Microsoft® Visual Studio®

Questions

Contact
Priyadarshan Shinde | priyadarshan.s@cytel.com | +91 20 6604 0106
Ajay Sathe | ajay.sathe@cytel.com | +91 20 6604 0119

Cytel Statistical Software & Services Pvt. Ltd., 8th Floor, Siddharth Towers,
Off Karve Road, Kothrud, Pune 411029, India