

Charlotte PowerBuilder Conference

Moving at the Speed of Change

May 2015

The “M” Word of Test Automation

Minimizing maintenance of automated tests

Brian Le Suer

Powered by



Sponsored by





Charlotte PowerBuilder Conference

Moving at the Speed of Change

May 2015

- Automation is intended to:
 - Increase efficiency of test organization
 - Increase effectiveness of test effort
 - Reduce costs
- Maintenance
 - Erodes the return on investment gained from test automation
 - Only a small portion of your resources should be spent on maintaining existing automated tests

Maintenance Costs



Charlotte PowerBuilder Conference

Moving at the Speed of Change

May 2015

- Choosing the right tests to automate
- Separating test requirements, object definitions, test functions and test data
- Building small reusable components
- Generating expected results at runtime
- Creating robust object definitions

5 Ways to Minimize Maintenance

Charlotte PowerBuilder Conference

Moving at the Speed of Change

May 2015

- Good candidates
 - Short or simple transactions
 - Universal tests
 - Many data combinations
 - Expected results are stable or easy to generate at runtime
- Poor candidates
 - Long or complex transactions
 - One-offs
 - Unstable application
 - Difficult to predict results

Choosing What To Automate



Charlotte PowerBuilder Conference

Moving at the Speed of Change

May 2015

- Tests that can be executed with minimal navigation
 - Data type validations
 - Data content validations
 - Invalid password
 - record not found
 - Missing required fields

Short or Simple Tests

Charlotte PowerBuilder Conference

Moving at the Speed of Change

May 2015

- Credit Card Application
 - Single page
 - Lots of data validation
 - Field formats
 - Dates
 - Phone numbers
 - Social Security numbers
 - Field lengths
 - Invalid characters
 - Missing required fields
 - Confirmation fields
 - Hundreds of combinations

The screenshot shows a web form for a credit card application, divided into three sections: Application, Confirmation, and Decision. The 'Application' section is active, indicated by a red square. The form includes the following fields and elements:

- Name:** First Name, Middle Name, Last Name, and a dropdown for Suffix.
- General Information:** Street Address, City, State (dropdown), Zip Code, and Apartment (Apt) number.
- Years at Residence:** Years and Months dropdowns.
- Monthly Rent/Mortgage:** Amount in dollars and cents, and a dropdown for Type of Residence.
- Home Phone:** Three input fields for area code, prefix, and number.
- Social Security Number:** Three input fields.
- Date of Birth:** Month, day, and year dropdowns.
- Mother's Maiden Last Name:** Input field.
- Email Address:** Input field and a Confirm Email Address field.
- Agreements:** Two checkboxes: "My computer is capable of printing, and saving my monthly statement and other communications." and "I agree to receive my statement and other communication online only."
- Employment and Finances:** Gross Annual Household Income (input field), Occupation (dropdown), Employer (input field), Work Phone (input field), and Length of Employment (years and months dropdowns).
- Account Types:** Two checkboxes: "Do you have a checking account?" and "Do you have a savings account?", each with Yes/No radio buttons.

Example

Charlotte PowerBuilder Conference

Moving at the Speed of Change

May 2015

- Tests that can be run against each window or page in the application under test
 - Standards
 - Labels must include a colon
 - Dialogboxes must include Closebox
 - Radio group can not have more than 3 choices
 - Z order
 - Min and max button size
 - Field length

Universal Tests

- [-] class VerifyFieldLength : Step
 - ◇ parameter AppObject FieldName
 - ◇ parameter Integer FieldLength
 - [-] Main ()
 - ◇ String TestString = ""
 - ◇
 - ◇ TestString = String.Replicate("a",FieldLength)
 - ◇
 - ◇ FieldName.SetValue(TestString + "x")
 - ◇
 - ◇ FieldName.VerifyValue(TestString)
 - ◇
 - ◇ return



Charlotte PowerBuilder Conference

Moving at the Speed of Change

May 2015

- Path through the application is shared by many tests
- Lots of different combinations of data need to be tested
- Write a reusable data-driven test
 - Separate test steps from test data
 - Relatively little work for a lot of coverage
 - Could pay for themselves in first run

Many Data Combinations

Charlotte PowerBuilder Conference

Moving at the Speed of Change

May 2015

- POS Sale Transaction
 - Definite path through the application
 - Easy to encapsulate GUI steps
 - Transactions executed with
 - Different tenders
 - Different items
 - In different states (taxes)
 - With different discounts

▣	POS Regression
▣	Sales
▣	Regular Sale
◇	Cash
◇	Personal Check
◇	Travel Check
◇	Gift Card
◇	Merch Voucher
▣	Credit
◇	MasterCard
◇	Visa
◇	Discover
▣	Discounted Sale
◇	Transaction Percent Off1
◇	Transaction Dollar Off
◇	Disaster Discount
◇	Design/Pref Business
◇	Technique Class
◇	Gift Reg Completion
◇	District Improve Fee

Example

Charlotte PowerBuilder Conference

Moving at the Speed of Change

May 2015

- Test requirements/objectives
 - What needs to be tested
- Tests
 - Ordered list of steps for satisfying test requirements
- Application object definitions
 - Insulating tests from changes in the GUI
- Test steps, units, functions
 - How application is traversed, manipulated and verified
- Test data
 - Inputs to and expected results of tests

Separation of Test Components



Charlotte PowerBuilder Conference

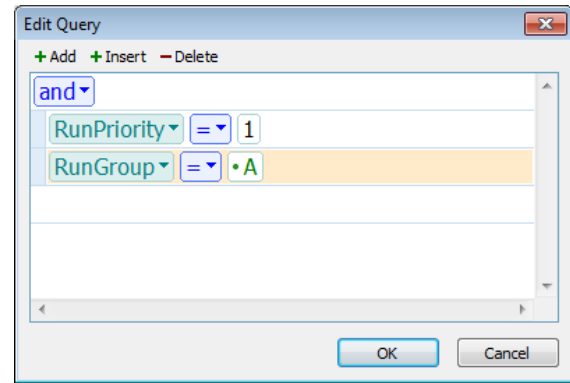
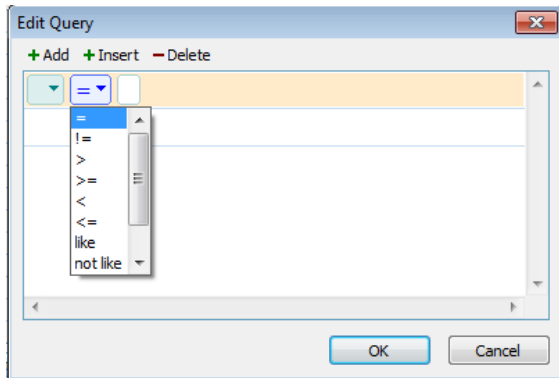
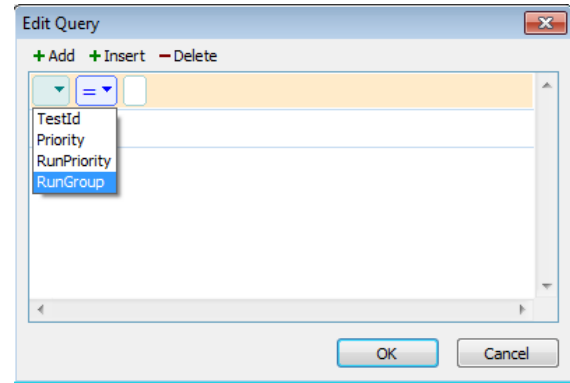
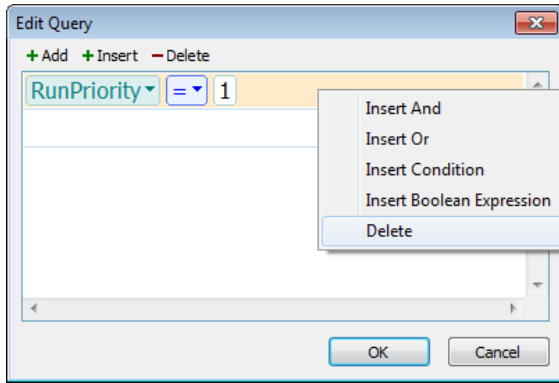
Moving at the Speed of Change

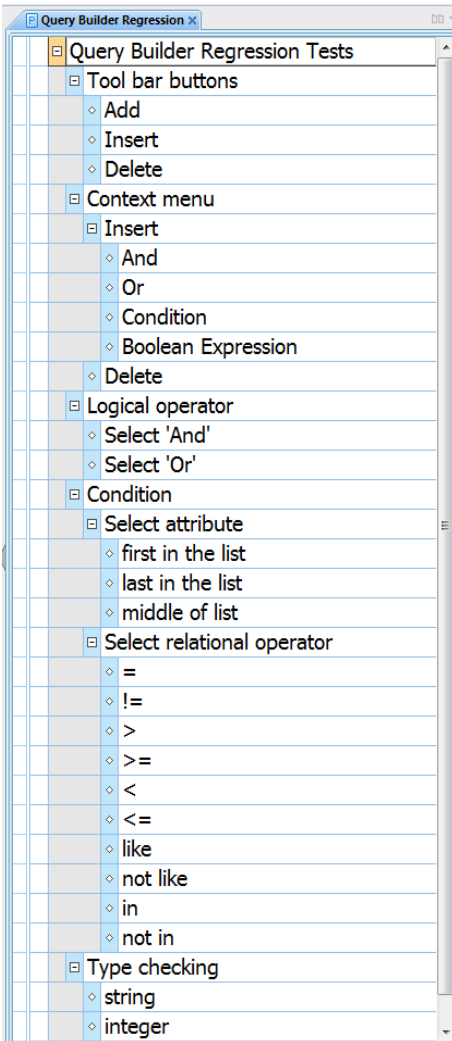
May 2015

- Describes what needs to be tested
 - Does not describe test steps
 - Does not describe how tests will be executed
 - Does not describe the data that will be used to execute test cases
- Graphical representation
 - Outline notation
 - Easy to understand coverage at a glance

Automation Ready Test Plans

12





Charlotte PowerBuilder Conference

Moving at the Speed of Change

May 2015

- Create tests that call reusable units
 - Automated tests should not contain GUI-specific actions
 - Automated tests should call units that encapsulate GUI-specific steps so that they can be reused by other tests
 - An Automated test should order the steps that need to be completed to satisfy the requirements of the test

Test Architecture

step	EnterItem	EnterItem1												
<table border="1"> <tr> <td colspan="4">Item</td> </tr> <tr> <td>Department</td> <td>SKU</td> <td>Price</td> <td>TaxIndicator</td> </tr> <tr> <td>•</td> <td>•</td> <td></td> <td>▪</td> </tr> </table>			Item				Department	SKU	Price	TaxIndicator	•	•		▪
Item														
Department	SKU	Price	TaxIndicator											
•	•		▪											
step	EnterItem	EnterItem2												
<table border="1"> <tr> <td colspan="4">Item</td> </tr> <tr> <td>Department</td> <td>SKU</td> <td>Price</td> <td>TaxIndicator</td> </tr> <tr> <td>•</td> <td>•</td> <td></td> <td>▪</td> </tr> </table>			Item				Department	SKU	Price	TaxIndicator	•	•		▪
Item														
Department	SKU	Price	TaxIndicator											
•	•		▪											
step	TotalTransaction													
step	CaptureVirtualReceipt													
step	TenderWithCheck													
<table border="1"> <tr> <td colspan="4">Tender</td> </tr> <tr> <td>TenderType</td> <td>Amount</td> <td>CardNumber</td> <td>CardKeyed</td> </tr> <tr> <td>▪</td> <td>•</td> <td>•</td> <td>•</td> </tr> </table>			Tender				TenderType	Amount	CardNumber	CardKeyed	▪	•	•	•
Tender														
TenderType	Amount	CardNumber	CardKeyed											
▪	•	•	•											
step	GenerateExpectedResults													
<table border="1"> <tr> <td>TranType</td> </tr> <tr> <td>▪</td> </tr> </table>			TranType	▪										
TranType														
▪														
step	CaptureElectronicJournal													
<table border="1"> <tr> <td>TranType</td> </tr> <tr> <td>▪</td> </tr> </table>			TranType	▪										
TranType														
▪														
step	VerifyTransaction													
<table border="1"> <tr> <td>CurrentTransactionType</td> </tr> <tr> <td>▪</td> </tr> </table>			CurrentTransactionType	▪										
CurrentTransactionType														
▪														

Charlotte PowerBuilder Conference

Moving at the Speed of Change

May 2015

- Navigational steps
 - Invoke
 - Close
- Input steps
 - Page specific
 - Universal
- Verification steps
 - Page specific
 - Universal

Small Reusable Components

17

set	InformationRequest.FirstName	• Value
set	InformationRequest.LastName	• Value
set	InformationRequest.CompanyName	• Value
set	InformationRequest.Address1	• Value
set	InformationRequest.Address2	• Value
set	InformationRequest.City	• Value
set	InformationRequest.State	• Value
set	InformationRequest.ZipCode	• Value
set	InformationRequest.PhoneNumber	• Value
set	InformationRequest.Email	• Value
set	InformationRequest.ConfirmEmail	• Value
set	InformationRequest.FirstTimeContact	• Value
	InformationRequest.Submit	Click

- ▣ class DemoStep : Step
 - ▣ Main()
 - ◆ InfoRequest.FirstName.SetValue ("")
 - ◆ InfoRequest.LastName.SetValue ("")
 - ◆ InfoRequest.CompanyName.SetValue ("")
 - ◆ InfoRequest.FirstName2.SetValue ("")
 - ◆ InfoRequest.Address2.SetValue ("")
 - ◆ InfoRequest.City.SetValue ("")
 - ◆ InfoRequest.State.SetValue ("")
 - ◆ InfoRequest.ZipCode.SetValue ("")
 - ◆ InfoRequest.PhoneNumber.SetValue ("")
 - ◆ InfoRequest.Email.SetValue ("")
 - ◆ InfoRequest.ConfirmEmail.SetValue ("")
 - ◆ InfoRequest.Submit.Click ()



Charlotte PowerBuilder Conference

Moving at the Speed of Change

May 2015

- Capture from application
 - At runtime during the test when the data is not dictated by test case requirements
- Capture from application data sources
 - Look up data in application database if available (SKUs, Prices)
- Store in format where data can be easily created and maintained

Test Data

Charlotte PowerBuilder Conference

Moving at the Speed of Change

May 2015

- Universal Verification Approach
 - The Expected Result is
 - generated based on inputs
 - is used as the baseline
 - A reusable verification function compares data sets that are captured from the AUT at runtime
 - For each 'l-value' in the baseline, the function compares all the 'r-values'

Small Reusable Components

20



Charlotte PowerBuilder Conference

Moving at the Speed of Change

May 2015

- Retail web site order entry
 - Products are selected randomly or specifically
 - Product attributes such as SKU, Description, Qty and Price are captured as part of the test
 - Same product attributes can be captured from the shopping cart, review order and submitted order pages
 - All attributes are written to a data store such as an .ini file

Example

Predicted Results:

[RegularSale1]
Item3_Style=528534
Item3_Price=29.99
Item3_Taxable=N
Item3_Dept=40
TaxState=MA
TaxRate=5.00
SubTotal=29.99
Total=31.49

Review Order Capture

[RegularSale1]
Item3_Style=528534
Item3_Price=29.99
Item3_Taxable=N
Item3_Dept=40
TaxState=MA
TaxRate=5.00
SubTotal=29.99
Total=31.49

Shopping Cart Capture:

[RegularSale1]
Item3_Style=528534
Item3_Price=29.99
Item3_Taxable=N
Item3_Dept=40
TaxState=MA
TaxRate=5.00
SubTotal=29.99
Total=31.49

Charlotte PowerBuilder Conference

Moving at the Speed of Change

May 2015

- Create Object Definitions/GUI Map rather than recording dynamic objects
- Use stable attributes
 - Field labels
 - Internal Window identifiers
- Do not use attributes that may change at runtime
 - Absolute location or RECT
- Dynamically define objects when it makes sense

Robust Object Definitions

23

Home

Store Pick List ×

Store Pick 9723002106 ×



Quantity Pick

Finalize Pick

Back

Order number: 9723002106 Order Type: In-Store Pickup

Due in: **Overdue**

IPAD WI-FI 16GB BLACK-USA (285142)

Picked Quantity: of 3 Shortage Quantity 3 Shortage Reason: Storage Location

IPAD WI-FI 32GB BLACK-USA (285143)

Picked Quantity: of 2 Shortage Quantity 2 Shortage Reason: Storage Location



Quantity Pick

Order

[Return To Home](#)

Order # 9726033037

Order Date 05/26/2014 03:30:37

Recipient Name DE ROSSI

Email Subhasish.Halder@staples.com

Phone 5553337410

OrderLines

[Finalize Pick](#) [Pick Slip Reprint](#)

Line	SKU Description	SKU	Ordered Qty	Picked Qty	Shortage Qty	Shortage Reason	Storage Location
2	BICREG ROUND STICREG BALLPOINT PENS MEDIUM POINT RED DOZEN	123455	1.00	<input type="text"/>		<input type="text"/>	<input type="text"/>
1	AVERY WRITEON BIG TAB DIVIDERS STAB SET WHITE TABS	135848	1.00	<input type="text"/>		<input type="text"/>	<input type="text"/>

- [-] WebBlock
 - [-] WebBlock
 - [-] WebBlock
 - ◇ WebText [@Text=="IPAD WI-FI 16GB BLACK-USA (285142)"]
 - [-] WebBlock
 - [-] WebBlock
 - [-] WebLabel [@Text=="Picked Quantity:"]
 - ◇ WebTextField [@Label=="Picked Quantity:"]
 - ◇ WebText [@Text=="of 3"]
 - [-] WebBlock
 - [-] WebLabel [@Text=="Shortage Quantity"]
 - ◇ WebTextField [@Label=="Shortage Quantity"]
 - [-] WebBlock
 - [-] WebLabel [@Text=="Shortage Reason:"]
 - [-] WebBlock
 - [-] WebTextField
 - ◇ WebTextField [@Label=="Shortage Reason:"]
 - [-] WebBlock
 - [-] WebLabel [@Text=="Storage Location"]
 - ◇ WebTextField [@Label=="Storage Location"]
 - [-] WebBlock

- [-] WebTable
 - [-] Row [@RowIndex==1]
 - [-] Row [@RowIndex==2]
 - [-] Cell [@CellIndex==1] / WebText [@Text=="2 "]
 - [-] Cell [@CellIndex==2] / WebText [@Text=="BICREG ROUND STICREG B/"]
 - [-] Cell [@CellIndex==3] / WebText [@Text=="123455 "]
 - [-] Cell [@CellIndex==4] / WebText [@Text=="1.00 "]
 - [-] Cell [@CellIndex==5] / WebTextField [@Name=="xml:/Order/OrderLines"]
 - [-] Cell [@CellIndex==6] / WebLabel
 - [-] Cell [@CellIndex==7]
 - [-] WebPopupList [@Name=="xml:/Order/OrderLines/OrderLine_1/@Extr"]
 - [-] Cell [@CellIndex==8]
 - ◇ WebTextField [@Name=="xml:/Order/OrderLines/OrderLine_1/@Extr"]
 - [-] Row [@RowIndex==4]

- [-] ✓ WebElem OrderLineTable
 - ◇ ✓ WebText PickedQuantity
 - [-] 2 class Row
 - ◇ selector [string SelectByDescription]
 - ◇ 2 WebText Description
 - ◇ 2 WebText PickedQuantityLabel
 - ◇ 2 WebTextField PickedQuantity
 - ◇ 2 WebText ShortageQuantityLabel
 - ◇ 2 WebTextField ShortageQuantity
 - ◇ 2 WebTextField ShortageReason
 - ◇ 2 WebText ShortageReasonLabel
 - ◇ 2 WebText StorageLocationLabel
 - ◇ 2 WebTextField StorageLocation



Charlotte PowerBuilder Conference

Moving at the Speed of Change

May 2015

- Choosing the right tests to automate
- Separating test requirements, object definitions, test functions and test data
- Building small reusable components
- Generating expected results at runtime
- Creating robust object definitions

5 Ways to Minimize Maintenance

28



Charlotte PowerBuilder Conference

Moving at the Speed of Change

May 2015

Thank you!
Questions?