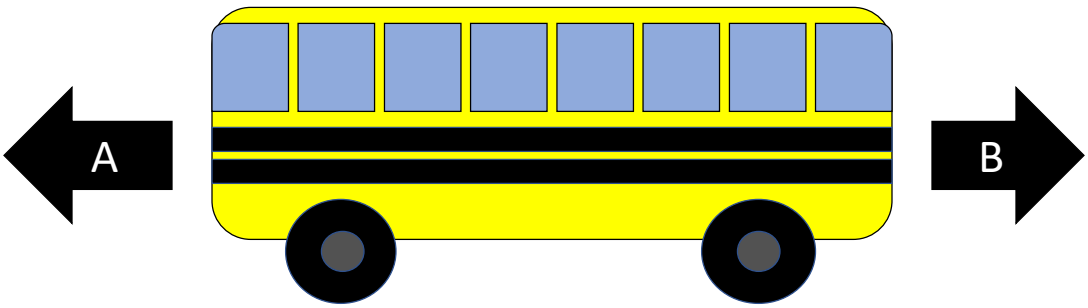
About Bob Crews

- President/Co-founder of Checkpoint Technologies
- Co-Leader Vivit Florida Chapter
- President of TBQAA (Tampa Bay Quality Assurance Association)
- CSTE/CAST Instructor
- 29 years IT experience
- 19 with focus on QA and QC
  - Risk Analysis
  - Test Automation
  - Effective Test Planning & Test Case Design
  - Internet of Things
  - **Agile & DevOps**

Which direction will the bus move once the bus starts moving forward?



Implementing a Reliable Framework Design In a DevOps world

Learning Objectives

- Necessity/value of test automation in DevOps
- Elements of a successful automation framework
- Value of an automation framework
- Practical tips to design and implement an automation framework in DevOps

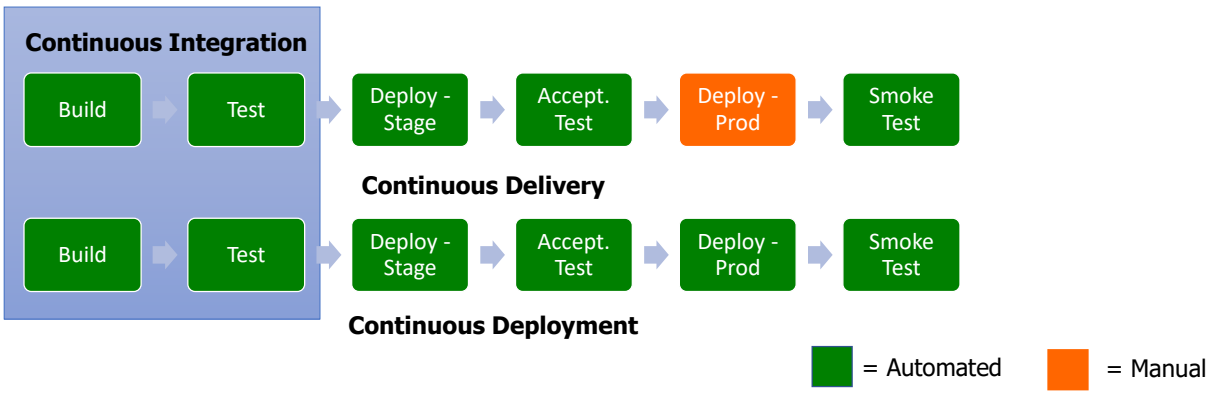
Why DevOps?

Definition, Goals, and Differences

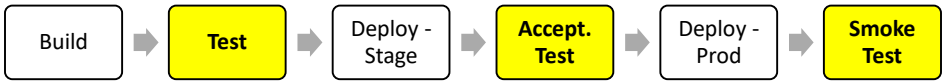
checkpoint TECHNOLOGIES

Implementing a Reliable Framework Design In a DevOps world

Continuous Integration, Continuous Delivery, Continuous Deployment



Let's Focus On Automated Testing



Understand how the scope of testing changes with DevOps.

checkpoint TECHNOLOGIES Software Quality. Assured
Implementing a Reliable Framework Design In a DevOps world

Testing Scope Change

With each occurrence it's…

Traditional

Testing less often and validating more revisions

vs

DevOps

Testing more frequently to validate fewer revisions

How? Test automation and…

A Well-Designed Framework

- *Decreases* automated test development time
- Enables tests to be created *sooner*
- *Decreases* maintenance
- *Increases* test automation coverage
- Puts the power of automation in *more hands*

Fundamentals of a Successful Framework

Definition, Goals, Characteristics & Common Elements

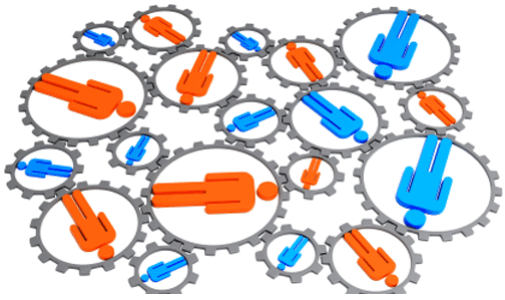
Automation Framework Definition

- A test automation methodology that *separates* the automation code from the data
- Allows for separate development and maintenance of the two assets

checkpoint TECHNOLOGIES Software Quality Assured
Implementing a Reliable Framework Design In a DevOps world

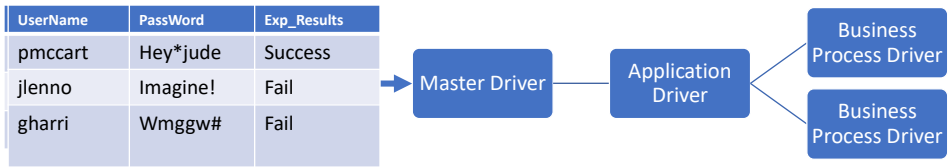
Automation Framework Characteristics

- Modular code
- Scalable
- Error handling
- Reliable
- Data separated from code



Common Elements of Successful Frameworks

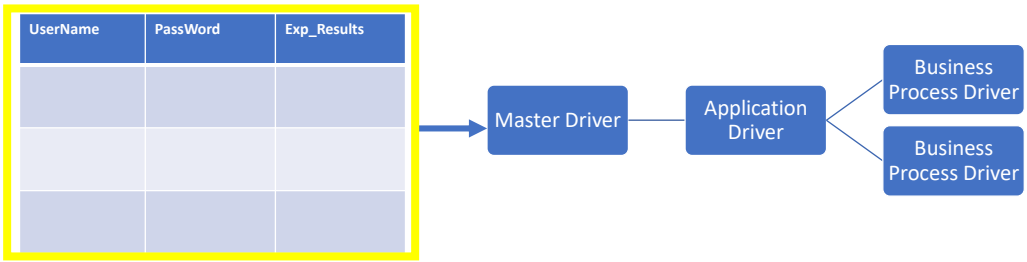
1. Simple front-end
2. Maintainable back-end
   A. Master Driver
   B. Application Driver
   C. Business Process Driver
3. Data



Implementing a Reliable Framework Design In a DevOps world

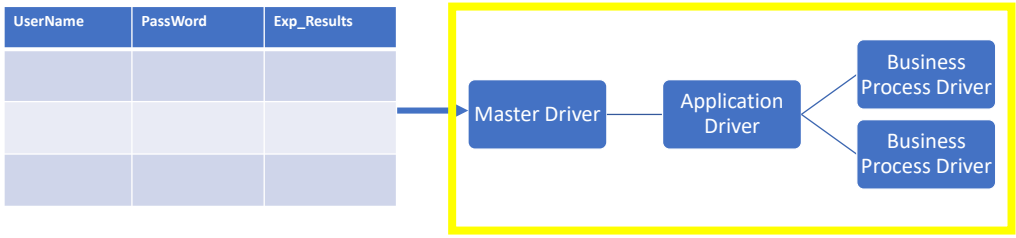
Simple Front-End

➢The "mechanism" for creating the test scripts!



Robust Back-End

- Code that reads the data executes the test and handles everything possible



Implementing a Reliable Framework Design In a DevOps world

Back-End Drivers

- Master Driver Script
  - The primary "traffic cop"! Calls the appropriate Application Driver
- Application Driver
  - The secondary "traffic cop". Calls the automated business process related to the name of the current sheet
- Business Processes Driver
  - Executes the granular, functional business process and reports status

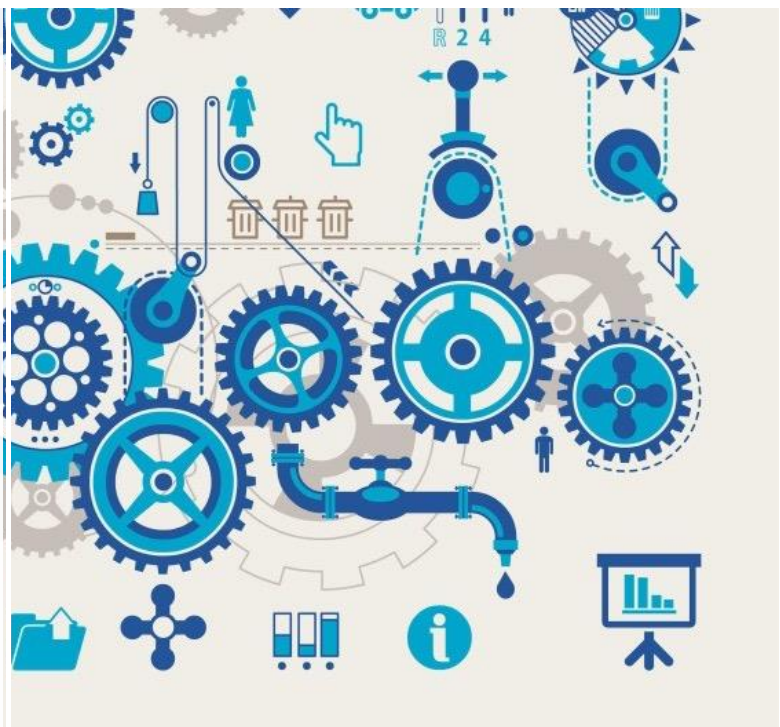
Data

- The fuel for the framework!
- In the form of input and/or keywords
- Must allow for multiple data iterations

UserName	PassWord	Exp_Results
pmccart	Hey*jude	Success
jlenno	Imagine!	Fail
gharri	Wmggw#	Fail

Automation Framework

Description, Design & Tips

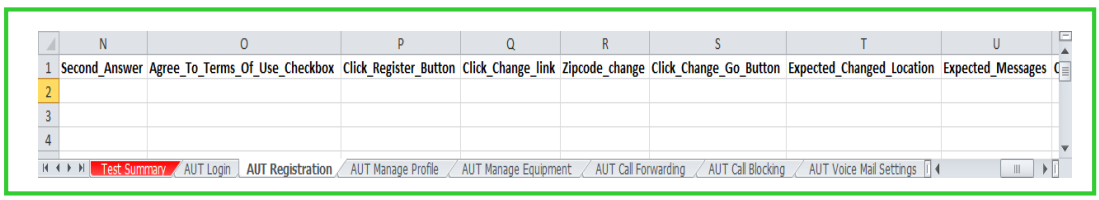


Key Framework Features

- Key-word & data-driven based on the AUT business processes (*Key-words*) and parameters (*Data*)
- Parameter Types:
  - Input – Data entered into edit fields, drop-down lists, etc.
  - Verification – Expected values compared to actual results
  - Decision/Action – Creates an action against an object
  - Output parameter – Place holders for data captured from the AUT during execution
- Low-level common functions
- Reporting

Front-End Description

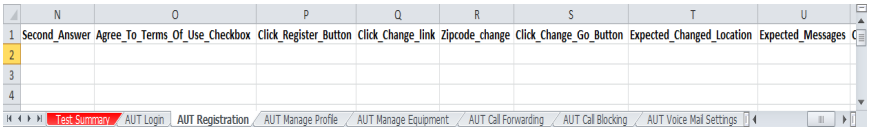
- Front-end utilizes Excel for script design
  - Each column represents a test step
  - Each record (row) a test iteration
  - Each tab represents a Business Process
  - Each BP tab contains all necessary parameters to create positive and negative tests
  - Output is captured to Output Parameters and can be used in subsequent iterations



- Multiple AUTs can be incorporated by adding tabs associated with other AUTs

Front-End Description (con't)

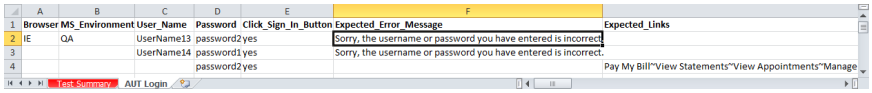
- A Master Excel Template contains all the business process for a given application.



	N	O	P	Q	R	S	T	U
1	Second_Answer	Agree_To_Terms_Of_Use_Checkbox	Click_Register_Button	Click_Change_link	Zipcode_change	Click_Change_Go_Button	Expected_Changed_Location	Expected_Messages
2								
3								
4								

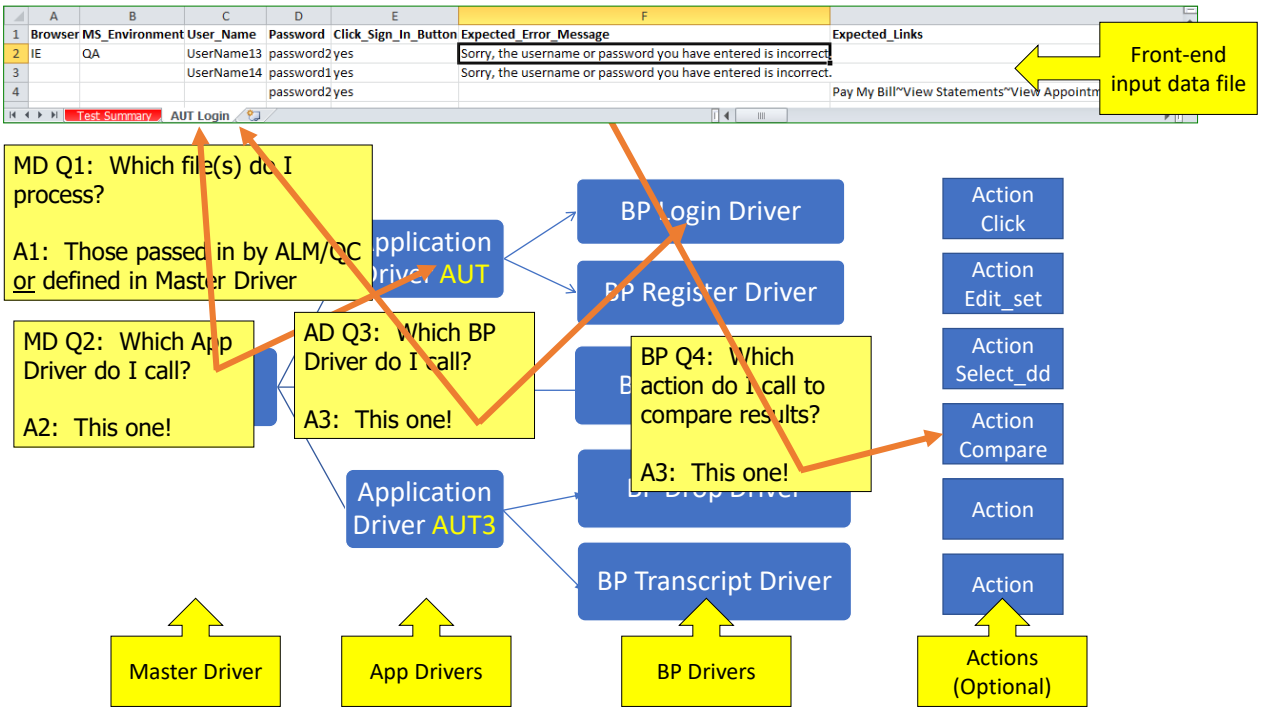
Test Summary | AUT Login | AUT Registration | AUT Manage Profile | AUT Manage Equipment | AUT Call Forwarding | AUT Call Blocking | AUT Voice Mail Settings

- To create a new test, simply save the Master Template as another file, remove the tabs not required for the test and reorder in the order they will execute, then apply the **data**.



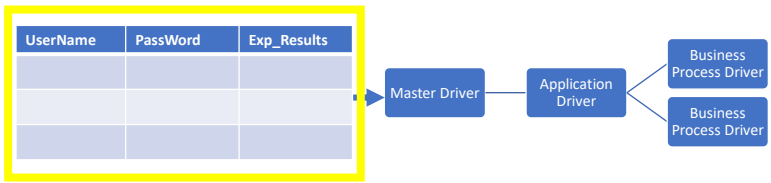
	A	B	C	D	E	F	
1	Browser	MS_Environment	User_Name	Password	Click_Sign_In_Button	Expected_Error_Message	Expected_Links
2	IE	QA	UserName13	password2	yes	Sorry, the username or password you have entered is incorrect.	
3			UserName14	password1	yes	Sorry, the username or password you have entered is incorrect.	
4				password2	yes		Pay My Bill~View Statements~View Appointments~Manage

Test Summary | AUT Login



Front-End & Data Design

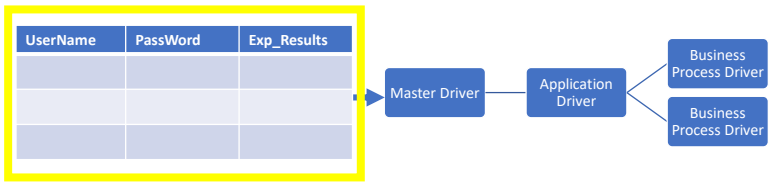
- Design front-end prior to back-end
  - Who will be tasked with using the framework to design the automated tests?
  - In what format will the tests be designed?
  - What tool will manage the data? (Excel, DB, XML, etc.)
  - Utilize Input, Output, Decision and Verification parameters
  - What is the syntax/rules governing the data input?



checkpoint TECHNOLOGIES Software Quality Assured
Implementing a Reliable Framework Design In a DevOps world

Front-End Design Tips

- Tips & Tricks
  - Know your users
  - Use common tool for data input
  - Document and communicate standard
  - Incorporate simple keywords



Back-End Description - Drivers

- **Master driver script** controls the flow of the script based on the data.
  - Calls **Application driver scripts** which in turn calls **Business Process drivers**

```
'Evaluate if the current sheet (action). If data exists in the current iteration the sub-driver will be called
If ExecuteAction(LocalActionExcelObject,objActionWorkSheet,Environment("IterationRow")) Then
    'Call the application specific parent action based on the current sheet name prefix
    Select Case Ucase(sheetNamePrefix)
        Case "MS"
            RunAction "My Services Driver [My Services Driver Script]", oneIteration
        Case "ATIM"
            RunAction "ATIM Driver Script Action [ATIM Driver Script]", oneIteration
        Case "CRM"
            RunAction "CRM Driver Script Action [CRM Driver Script]", oneIteration
        Case Else
            Reporter.ReportEvent micFail,"Process " & sheetName & " is not yet accounted…
    End Select
End If 'End If ExecuteAction(LocalActionExc
```

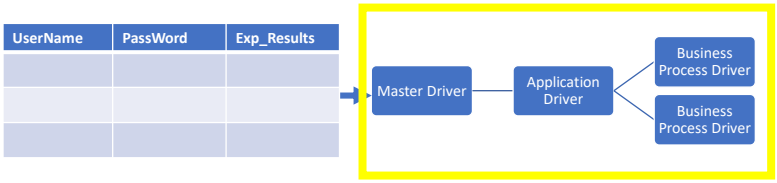
Back-End Description - Actions

- Code in the form of actions
- Mostly devoid of logic
- Contain steps referencing the object with the related data

```
Browser("Networks").Page("Networks").WebElement("Register for a My Services").VerifyExist "Register Page"

Browser("Networks").Page("Networks").WebEdit("Email Address").Set Parameter("Email_Address")
Browser("Networks").Page("Networks").WebEdit("Re-Enter Email Address").Set Parameter("ReEnter_Email_Address")
Browser("Networks").Page("Networks").WebEdit("Account Number").Set Parameter("Account_Number")
Browser("Networks").Page("Networks").WebEdit("Customer Code").Set Parameter("Customer_Code")
Browser("Networks").Page("Networks").WebEdit("Username").Set Parameter("Username")
Browser("Networks").Page("Networks").WebEdit("Choose Password").Set Parameter("Choose_Password")
Browser("Networks").Page("Networks").WebEdit("Question 1").Set Parameter("First_Security_Question")
Browser("Networks").Page("Networks").WebEdit("Your Answer 1").Set Parameter("First_Answer")
Browser("Networks").Page("Networks").WebEdit("Your Answer 2").Set Parameter("Second_Answer")
Browser("Networks").Page("Networks").WebCheckBox("Iagree").Set Parameter("Agree_To_Terms_Checkbox")
Browser("Networks").Page("Networks").Link("Register").Click Parameter("Click_Register_Button")

Browser("Bright House Networks").Page("Bright House Networks").VerifyMessages Parameter("Expected_Messages")

Browser("Bright House Networks").Page("Bright House Networks").ClickLinkOnPage Parameter("Link_To_Click")
```

Back-End Design

- Build the back-end to support the front-end
  - How does the data get fed into the framework?
  - How does runtime data get captured and reused?
  - How are multiple iterations handled?
  - What type and level of reporting is created – tool generated and/or custom? (screenshots)
  - How are unexpected events handled?



checkpoint TECHNOLOGIES Software Quality Assured
Implementing a Reliable Framework Design In a DevOps world

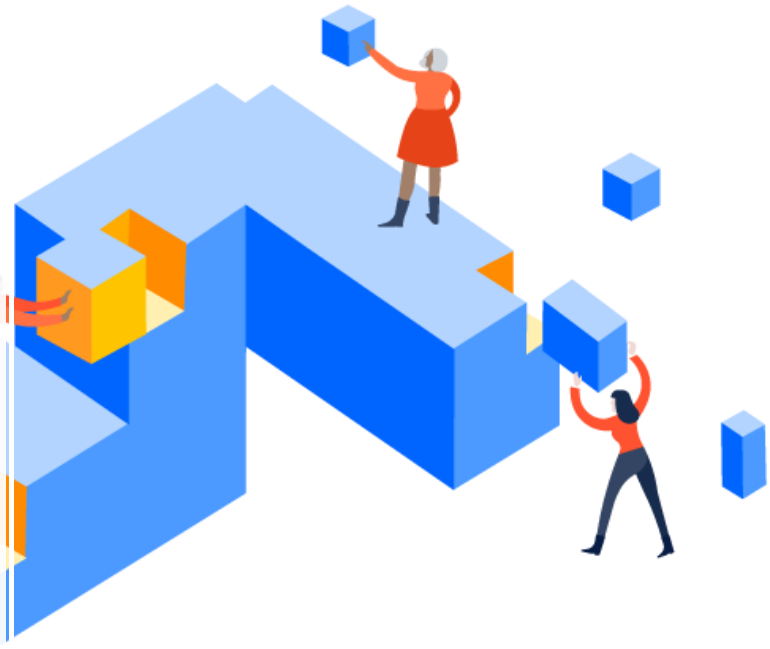
Back-End Design Tips

- The code must be:
  - Modular in design
  - Well documented/commented
  - Reporting needs to be detailed, clear & concise
  - Simple enough so that a master's degree in software development is not required for maintenance and support
- Multiple applications and platforms need to be taken into consideration
- Know the scope and expectations of automation within your organization

TIP

Applying Your Framework

For DevOps or Continuous Integration



Next steps

- Initially start with getting to CI and Continuous Delivery
- Start with automation of unit tests
- Automate your deployments to stage environment ASAP
- Configure CI system to launch all tests based upon appropriate event (build, deployment, or time)
- Releasing SW on daily basis? Time to look at Continuous Deployment

checkpoint TECHNOLOGIES Software Quality. Assured
Implementing a Reliable Framework Design In a DevOps world

Thank you!

Bob Crews
Checkpoint Technologies, Inc.
Email: bcrews@checkpointech.com