

Lesson Objectives



- Understanding Framework
- Why Framework
- Types of Framework
- Linear Automation Framework
- Data Driven Automation Framework
- Modular Driven Automation Framework
- Keyword Driven Automation Framework
- Hybrid Automation Framework
- Test Automation Process – Framework Approach
- Test Automation Process – Framework Selection
- Test Automation Process – Tool selection
- Test Automation Process – Automation Process
- Automation Architecture



Understanding "Framework"



- What does "it" mean:
 - A skeletal structure designed to support or enclose something
 - A frame of structure composed of parts fitted and joined together
 - Work done in, on or with a frame or structure
- Definition:
 - A fundamental structure for supporting or enclosing something, especially a skeletal support used as the basis for something being constructed
 - A set of assumptions, concepts, values and practices that constitutes a way of viewing reality

Defining "Framework" – Well, an Attempt !!??



In all, framework is a structured combination of

- Various (testing) assumptions
- Practices
- Concepts

Ultimate goal is to support "Automated Software Testing"

Framework is an Idea

Why Framework



- Outline the Overall Test Structure
 - To Visualize Test Automation “Even before we begin”
- Minimize code development
 - Reduce Maintenance
- Tool Independency
- Repeatable
 - Atleast on homogenic applications
- Portability & Expendability
 - Plug & Play
- Consistency
- Maximize Reusability
- “Ideally” Reduce coding exposure to functional testers

Types of Framework



- Linear Framework (Record and Play back)
- Data driven
- Component/Modular
- Keyword Driven
- Hybrid

Linear Automation Framework (Record & Play Back)



- Script Generation Approach/technique
 - Supported by most of the modern automation tool
 - A micro-like script where each user action is presented
- Adhoc
 - Used for automation of day-to-day activities (Adhoc)
 - Example : if an Engineer need to verify a bug : user creation module, for 1001 user getting data error
- Very limited scope
 - Confined to either Individual tester/functional Requirement
- Shortcomings
 - Hard-coded
 - Error Handling
 - Minimum Reusability
 - High Maintenance (as good as the "creation time" itself)

Data Driven Automation Framework



- “Data” becomes the nucleus
 - Data sits in middle and test cases get repeated with predefined set of Input data and response data coming from AUT
 - To check how the application performs the same operations with multiple sets of data.
- Data Independency
 - Login (user name and password)
- Best suited to tests aimed at larger data sets and less change prone
 - Example :
 - Bank Account creation modules
 - Mail Accounts Creation
 - Forms filling applications

Data Driven Automation Framework



Example :

- `DataTable.AddSheet "LocalSheet"` ' This Step is used to add a empty sheet locally in default excel in the Test
- `DataTable.ImportSheet "C:\QTPTEST\Login.xls", "LoginSheet", "LocalSheet"` 'This step is used to import dat from specific sheet specific datafile into the above created loacal sheet
- `rowCount = DataTable.GetSheet("LocalSheet").GetRowCount` 'This step is used to get the row count of the spcifoied sheet in default excel sheet
- For row = 1 to rowCount
- `DataTable.GetSheet("LocalSheet").SetCurrentRow(row)` 'This step is used to set focus on specified row
- `userName = DataTable.Value("UN", "LocalSheet")` ' This step is used get the value from sheet in the column "UN"
- `pWD = DataTable.Value("PWD", "LocalSheet")` ' This step is used get the value from sheet in the column "PWD"
- `PRINT userName`
- `PRINT pWD`
- `DataTable.Value("Result", "LocalSheet") = "TEST"` 'This Step is used to set a value in specified sheet and in specified column
- Next
- `DataTable.ExportSheet "C:\QTPTEST\Login.xls", "LocalSheet"` 'This step is used to export the local sheet into external excel

Data Driven Automation Framework



- Shortcomings
 - Minimum Reusability
 - Mostly Application/requirement specific scripts
 - Vulnerable to Changes (viz. high maintenance)

Modular/Component Driven Automation Framework



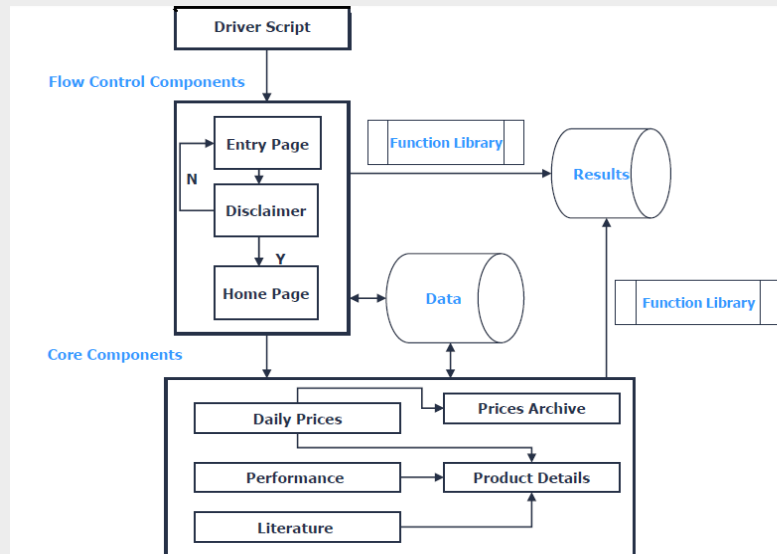
- Based on typical Functional Decomposition technique
- Built on mutually independent scripts/blocks
- Re-usable components identified
- AUT is fragmented into small blocks
 - Functionalities/Requirements
 - Functional flow
 - System components itself
 - Generic Functions/actions on the AUT
- Also termed as "Test Library Architecture Framework"
 - As functions & modules of the AUT are represented in the form of library (common & application specific) file

Modular/Component Driven Framework....



- Shortcomings:
 - High Maintenance
 - Scripts are generated/created on business logic
 - Dependency among scripts

Implementation Model – Component/Modular Driven Framework



Component/Modular Driven Framework



Example – Contracts Management Application TC#1

- Login as Internal User
- Create Contract
- Add external Users
- Add terms
- Get Approvals
- Present Contract to External User
- Logout as Internal User
- Login as Internal User
- Login as External User
- Execute Contract

TC#2

- Login as Internal User
- Create Contract
- Add external Users
- Add terms
- Send Contract for Internal Review
- Get Approvals
- Present Contract to External User
- Logout as Internal User
- Login as External User
- Review Contract
- Execute Contract

Keyword Driven Automation Framework



- Based on Object/Action relationship
- Objects are Universal
 - Applications are built on objects (Browser, Page, Frame, Text box...) and so the "Keyword" framework
 - Maximum re-usability across homogenic applications
- Objects and the actions are represented in a table format
 - Hence also known as "Table based Automation Framework"
- Business logic Kept outside the script
 - Scripting limits to the definition of Object and the associated actions
 - Minimum Maintainability
 - Best suits for "Change-prone applications"

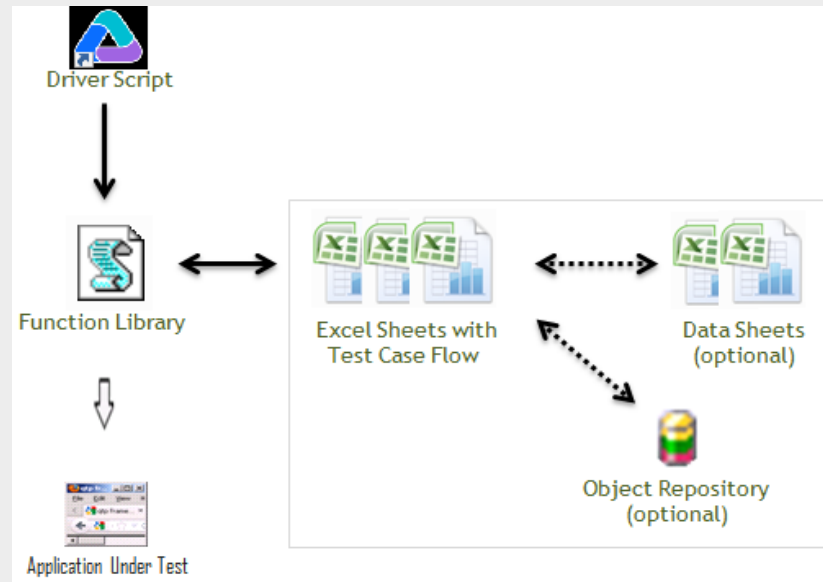
Keyword Driven Automation Framework...



- Portable & Extendible
 - Plug & play
 - Limited to the limitations of the platform used to develop the framework
- Keeps the Scripts (Code) away from framework implementers (functional testers)
- Maximum Reusability
- Greater traceability to manual test cases
- Manual Tester / BA can also do automation
- Re Work will be minimal for functional changes
- Fast Automation is possible with less skilled people in team

- Shortcomings
 - Effort to do Initial implementation of the framework is higher
 - Wouldn't be best option for AUTs with large data sets

Implementation Model – Keyword Driven Framework



Keyword Driven Framework - Example



Login

Login Here

User Name :

Password :

Login ▶

Book Tickets Now!

☐ Two Way ☒ One Way

From:

To:

No. Of Passengers: Adult Child

Depart: 09-08-2012 Return: 09-08-2012

Bus Type: SUPER LUXURY

Concession: GENERAL BOOKING

Check Availability ▶

e-Ticket Reservation **Login** ▶

APSRTC Operator/Agent **Login** ▶

Powered By

| BrowserName | PageName | ObjectName | KeyWord | Param1 | Param2 | Param3 | Param4 |
|-------------|----------|-------------------|-------------|-----------------|--------|--------|--------|
| BookTickets | Booking | Login | ClickButton | | | | |
| Login | Login | User Name: | SetText | suneel | | | |
| Login | Login | Password : | SetText | suneel | | | |
| BookTickets | Booking | One Way | SelectRadio | | | | |
| BookTickets | Booking | From: | ListSelect | Chennai | | | |
| BookTickets | Booking | To: | ListSelect | Nellore | | | |
| BookTickets | Booking | Adult | ListSelect | | 2 | | |
| BookTickets | Booking | Child | ListSelect | | 0 | | |
| BookTickets | Booking | Depart | SetText | 09-08-2012 | | | |
| BookTickets | Booking | BusType | ListSelect | EXPRESS | | | |
| BookTickets | Booking | Concession | ListSelect | GENERAL BOOKING | | | |
| BookTickets | Booking | CheckAvailability | ClickButton | | | | |

Hybrid Automation Framework

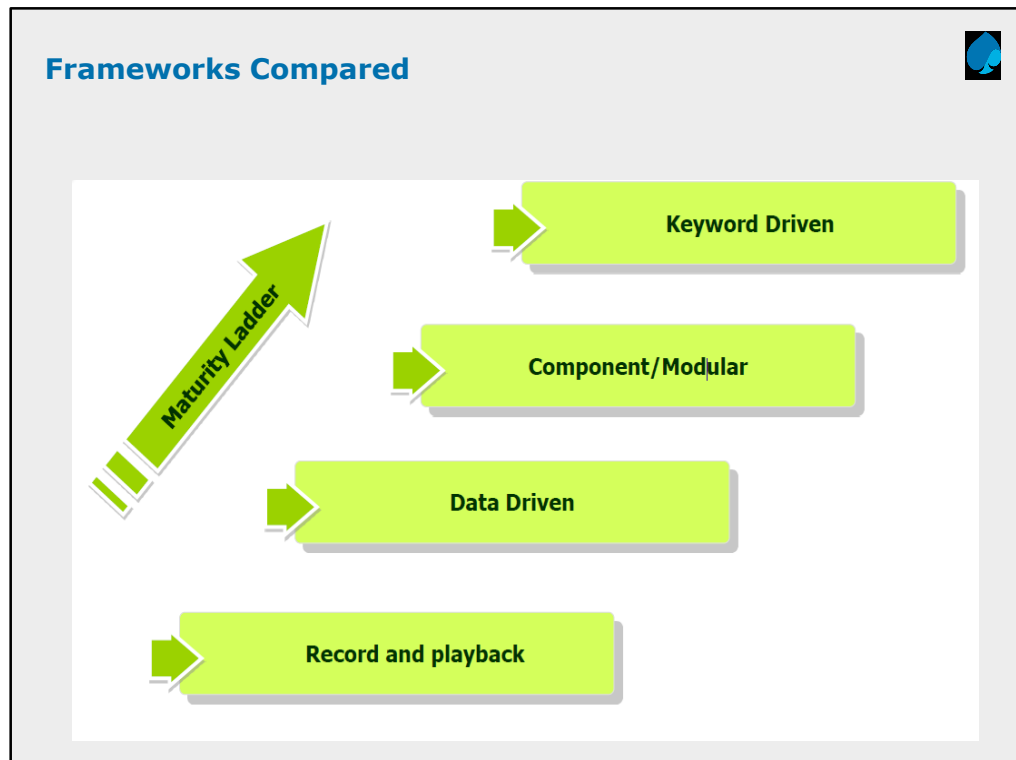


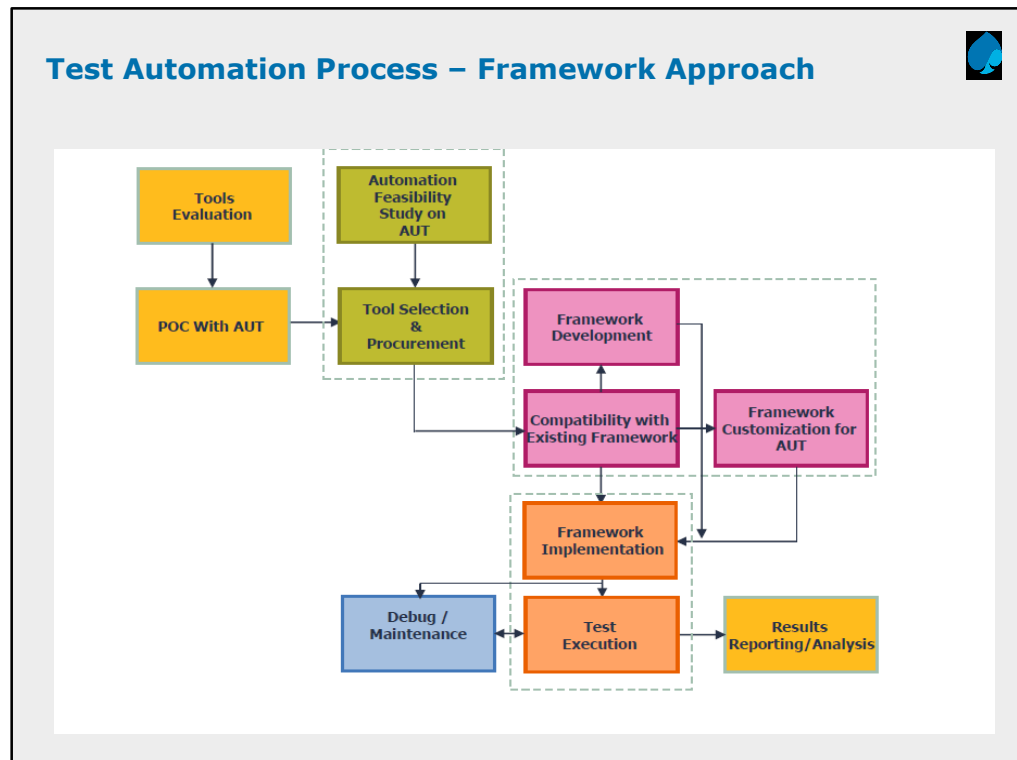
- The most commonly used
- Capitalized on the other techniques
 - Strengths of the other techniques (Record & Playback, Data, Modular, & Keyword) are leveraged
- Most Successful framework
 - Mostly the combination of Keyword + Data driven

Hybrid Automation Framework



| SuiteName | TestCaseId | StepNumber | StepDescription | ExecutionFlag | ActionKeyWord | InputParameter |
|----------------------|--------------|------------|---|---------------|------------------|--|
| ICECreditClearing_FS | TC_DCH_FS_01 | 1 | TC #1.3.3. Triclear Exception - MS Recall as EB-CX.NA.IC.HVOL | Yes | CREATEFTE | ExcelWorksheet=TE_TradeCreation.xls UniqueCollName=Comments UniqCollValue=502 |
| ICECreditClearing_FS | TC_DCH_FS_01 | 2 | Get field values required to match trade in Stix | Yes | GETFIELDDATA | PTradeRef=C1.PTradeRef ProductTypeCode=ClearedIndex FieldList=Template.Reference |
| ICECreditClearing_FS | TC_DCH_FS_01 | 3 | Search for the C1 trade in Stix | Yes | SEARCH | TradeId=C1.PTradeRef |
| ICECreditClearing_FS | TC_DCH_FS_01 | 4 | Create a new trade in Stix | Yes | CREATE | MSBuyProtection=C1.MSbuysSells IndexIssue=C1.ReferenceEntity Template=C1.Template |
| ICECreditClearing_FS | TC_DCH_FS_01 | 5 | Match trade in Stix | Yes | VERIFYMATCH | TradeId=C1.PTradeRef |
| ICECreditClearing_FS | TC_DCH_FS_01 | 6 | Verify that the Allocation status of trade in Stix is "Unallocated" | Yes | CHECKTRADESTATUS | TradeRef=C1.PTradeRef TradeStatus=Unallocated |
| ICECreditClearing_FS | TC_DCH_FS_01 | 7 | Verify TradeFlow Status in C1. It should be in "New-Pending Sales" | Yes | TRADESTATUS | PTradeRef=C1.PTradeRef ProductTypeCode=ClearedIndex strTradeFlowState=Live New |
| ICECreditClearing_FS | TC_DCH_FS_01 | 8 | Search trade using Trade Ref | Yes | SEARCH | PTradeRef=C1.PTradeRef TradeOperation=Clearing SearchBy=Trade IDs FlowType=EB |
| ICECreditClearing_FS | TC_DCH_FS_01 | 9 | Retrieve Platform ref for the trade | Yes | RETRIEVEDATA | PTradeRef=C1.PTradeRef FieldList=EntityType.Event Type=Notional PlatformRef |
| ICECreditClearing_FS | TC_DCH_FS_01 | 10 | Verify Trade is RECEIVE the ALLEGED from STIX in Triclear | Yes | VERIFYMESSAGE | PTradeRef=C1.PTradeRef ExchangeType=INTERNAL_ALLEGE ConversationEntity=STIX ExpectedStatus=TRUE |
| ICECreditClearing_FS | TC_DCH_FS_01 | 11 | Verify Trade is sent to ICE from Triclear -> Transaction History | Yes | VERIFYMESSAGE | PTradeRef=C1.PTradeRef ExchangeType=TRADE_SUBMISSION ConversationEntity=ICELN |
| ICECreditClearing_FS | TC_DCH_FS_01 | 12 | Verify Acknowledgement from ICE in Triclear -> Transaction History | Yes | VERIFYMESSAGE | PTradeRef=C1.PTradeRef ExchangeType=ACKNOWLEDGEMENT ConversationEntity=ICELN |
| ICECreditClearing_FS | TC_DCH_FS_01 | 13 | Search trade in ICE (HF1) under Transactions tab | Yes | HF1SEARCH | RefId=TRICLEAR.PlatformRef FlowType=EB TabSelection=TRANSACTIONS |
| ICECreditClearing_FS | TC_DCH_FS_01 | 14 | Affirm trade in ICE | Yes | AFFIRM | FlowType=ClearingEB NumberOfFunds=C1 Fundsmc_hft_1es NotionalAmount=991 FC |
| ICECreditClearing_FS | TC_DCH_FS_01 | 15 | Search trade in ICE (HF1) under Transactions tab | Yes | HF1SEARCH | RefId=TRICLEAR.PlatformRef FlowType=EB TabSelection=TRANSACTIONS |
| ICECreditClearing_FS | TC_DCH_FS_01 | 16 | Verify Buyer approval in ICE HF1 Application | Yes | ICEVERIFY | WindowName=ICELink_HF1 StatusVerify=PROTECTIONBUYER strExpectedStatus=True |
| ICECreditClearing_FS | TC_DCH_FS_01 | 17 | Verify Seller approval in ICE HF1 Application | Yes | ICEVERIFY | WindowName=ICELink_HF1 StatusVerify=PROTECTIONSELLER strExpectedStatus=True |
| ICECreditClearing_FS | TC_DCH_FS_01 | 18 | Search trade using Trade Ref | Yes | SEARCH | PTradeRef=C1.PTradeRef TradeOperation=Clearing SearchBy=Trade IDs FlowType=EB |
| ICECreditClearing_FS | TC_DCH_FS_01 | 19 | Verify Triclear receive Allge from ICE | Yes | VERIFYMESSAGE | PTradeRef=C1.PTradeRef ExchangeType=EXTERNAL_ALLEGE ExchangeStatus=ALLEGED Cc |
| ICECreditClearing_FS | TC_DCH_FS_01 | 20 | Verify Triclear send Booking Request to STIX | Yes | VERIFYMESSAGE | PTradeRef=C1.PTradeRef ExchangeType=BOOKING_REQUEST ConversationEntity=STIX ExpectedStatus=TRUE |
| ICECreditClearing_FS | TC_DCH_FS_01 | 21 | Verify Allocation is approved in STIX | Yes | CHECKTRADESTATUS | TradeId=C1.PTradeRef TradeStatus=Allocated (1) |
| ICECreditClearing_FS | TC_DCH_FS_01 | 22 | Verify Allocation is created in C1 under Trade Groups tab and Blog | Yes | TRADESTATUSONLY | PTradeRef=C1.PTradeRef ProductTypeCode=ClearedIndex strTradeFlowState=Live New |
| ICECreditClearing_FS | TC_DCH_FS_01 | 23 | Search trade using ICE Transaction ID for Retrieving CB Trade | Yes | SEARCH | PTradeRef=TRI.CLEAR.PlatformRef TradeOperation=Clearing SearchBy=Platform Refs FlowType=EB |
| ICECreditClearing_FS | TC_DCH_FS_01 | 24 | Verify Triclear receive Status update for Cleared from ICE | Yes | VERIFYMESSAGE | PTradeRef=TRI.CLEAR.PlatformRef ExchangeType=EXTERNAL_ALLEGE ExchangeStatus=ALL |
| ICECreditClearing_FS | TC_DCH_FS_01 | 25 | Verify Triclear sends Status update for Cleared to C1 | Yes | VERIFYMESSAGE | PTradeRef=TRI.CLEAR.PlatformRef ExchangeType=BOOKING_REQUEST ConversationEntity=STIX ExpectedStatus=TRUE |
| ICECreditClearing_FS | TC_DCH_FS_01 | 26 | Search the trade in ICE as DL1 | Yes | DL1SEARCH | RefId=TRICLEAR.PlatformRef FlowType=FS TabSelection=ICETransactionId |
| ICECreditClearing_FS | TC_DCH_FS_01 | 27 | Recall trade as EB from ICE (DL1) and Verify Buyer Approval as Rec | Yes | ICEVERIFY | WindowName=ICELink_DL1 StatusVerify=RECALL strExpectedStatus=True |
| ICECreditClearing_FS | TC_DCH_FS_01 | 28 | Search CB trade in C1 under Trade search tool using External ID | Yes | SEARCH | ExternalId=TRI.CLEAR.PlatformRef TradeName=strix |
| ICECreditClearing_FS | TC_DCH_FS_01 | 29 | Verify Trade in C1 for pending approval | Yes | TRADESTATUSONLY | PTradeRef=C1.CBTradeRef_0 ProductTypeCode=ClearedIndex strTradeFlowState=Non-L |







Test Automation Process – Framework Selection

- Application Behavior
- Expected ROI
- Expertise in Team
- Re Work Probability



Test Automation Process – Tool Selection

- Application Behavior
- Budget
- Expected ROI
- Resources
- Technology
- Tool Support
- Resources in market

Test Automation Process – Automation Process



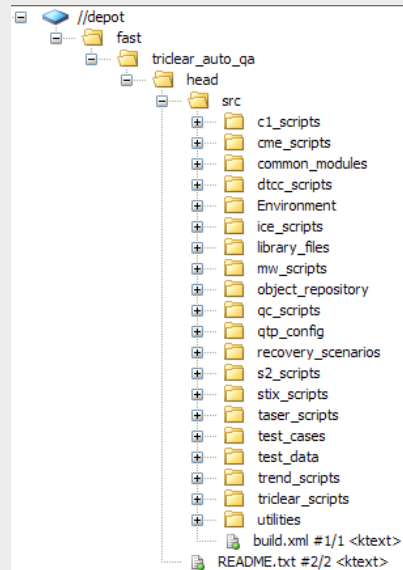
- POC
 - Feasibility
 - Approach
 - TCs for Automation
 - Tentative Estimations
 - Risks
- Automation Plan
- Pre Review
- Object Repository
- Test Data Preparation
- Re Usable Components
- Test Case wise automation
- Debugging and Execution
- Post Review
- Sign-Off / Release notes

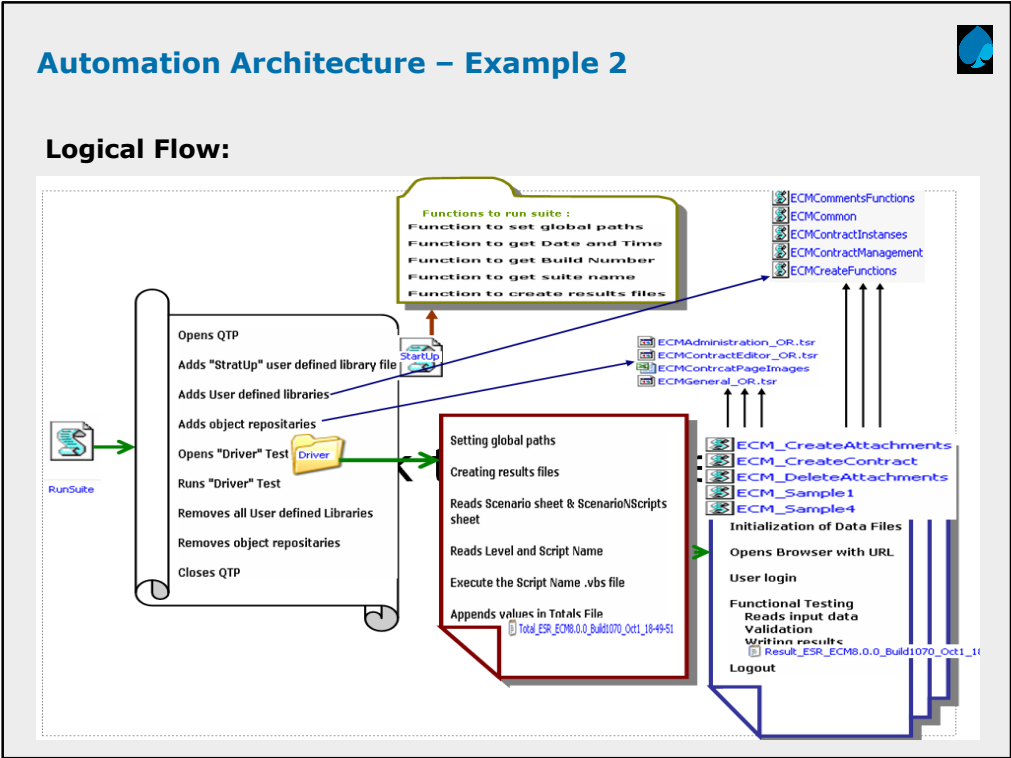
Automation Architecture



- What is being tested?
- How is the test set up?
- Where are the inputs coming from?
- What is being checked?
- Where are the expected results?
- How do you know pass or fail?
- Where I can have reusable components?

Automation Architecture – Example 1





Summary



In this lesson, you have learnt:

- What is Framework
A fundamental structure for supporting or enclosing something, especially a skeletal support used as the basis for something being constructed
- Different types of Frameworks
 - Linear Framework
 - Modular Framework
 - Data Driven Framework
 - Keyword Driven Framework
 - Hybrid Framework
- Test Automation Process
- Automation Architecture

