



IBM Rational Software Conference 2009
As Real as It Gets!



IBM Rational Software Conference 2009

Accelerated Test Case Automation Using Rational Functional Tester

Anish Bhanu

Manager, Software, Novellus Systems

Anish.Bhanu@Novellus.com

Sriram Chakravarthi

Sr. Engineer, Software, Novellus Systems

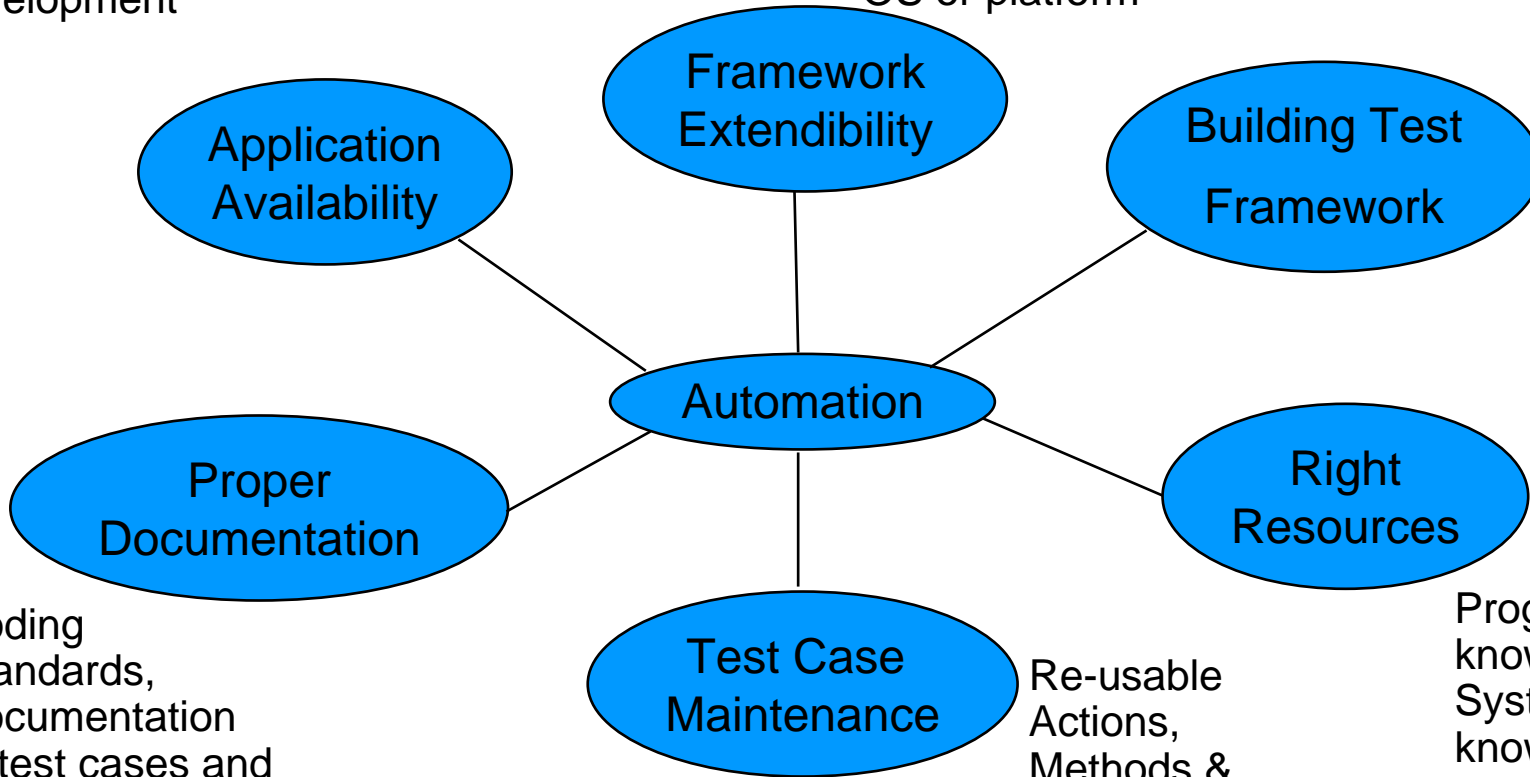
Rational software

Automation Challenges in today's environment

Automation lags development

Framework running on multiple hardware, OS or platform

Test cases with Data Driven or Keyword Driven approach



Coding Standards, Documentation of test cases and modules...

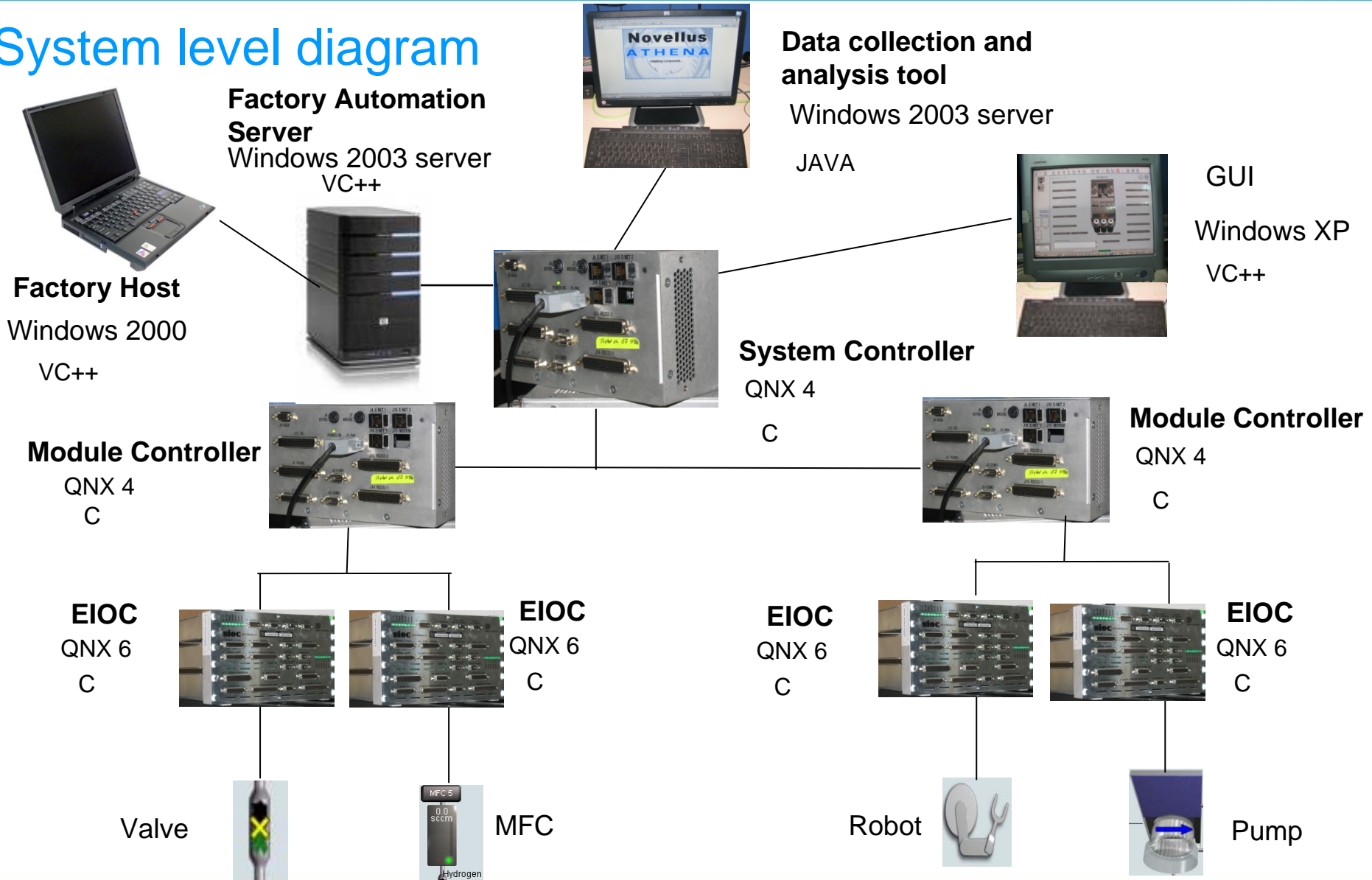
Re-usable Actions, Methods & Modules

Programming knowledge vs System knowledge...

Modified Chinese saying, "May you **automate** in interesting times"



System level diagram



Challenges for testing semi-conductor tool software

- **Around 40 releases a year supporting 14 products**
- **Manual testing of each release takes about 1200 hours of testing for each release**
- **Software reliability is a key differentiator for tool software**
- **System is graphic intensive, event driven and interacts with multiple other systems**
- **Every customer uses the tool in a unique way. Customer needs quick turnaround on features and fixes**



Test Automation Approaches

Record and Playback

Standalone Record & Playback Scripts

Functional Test Decomposition

Modular test functions with data within test script

Data Driven

Test scripts with input & output data outside the test script

Keyword Driven

Test scripts with input & output data outside the test script along with reusable keyword libraries



Novellus Experience with Automation

- **Started automation with Mercury WinRunner in 2003**
- **Started with the Functional decomposition approach and later moved to Data driven approach**
- **Test case development took us 8 times the manual testing time**
- **Maintaining the test cases for major releases took us 2 times the manual testing time**
- **By end of 2008, we had only 15% of test cases automated, which is around 180 hours of manual testing time**
- **Maintaining the script data for multiple releases and product was a challenge**
- **WinRunner was not able to recognize the controls and 25% of Regression test cases could not be automated**



New Automation Approach using RFT

- Simplify scripting
 - Script should mimic what the user does on the screen.
 - Anyone could write the script
- Modular architecture
 - Common framework that supports multiple products
 - Provide reuse of code and test structure
 - Error recovery system for continuous run
- Parameterize test methods and modules
 - Provides mechanism to create new test cases easily and increase coverage
- Documentation
 - Document generation of test methods and modules using JavaDoc
- Use Proxies
 - For custom controls not recognized by RFT
 - Rouge wave Stingray Controls



Control metrics

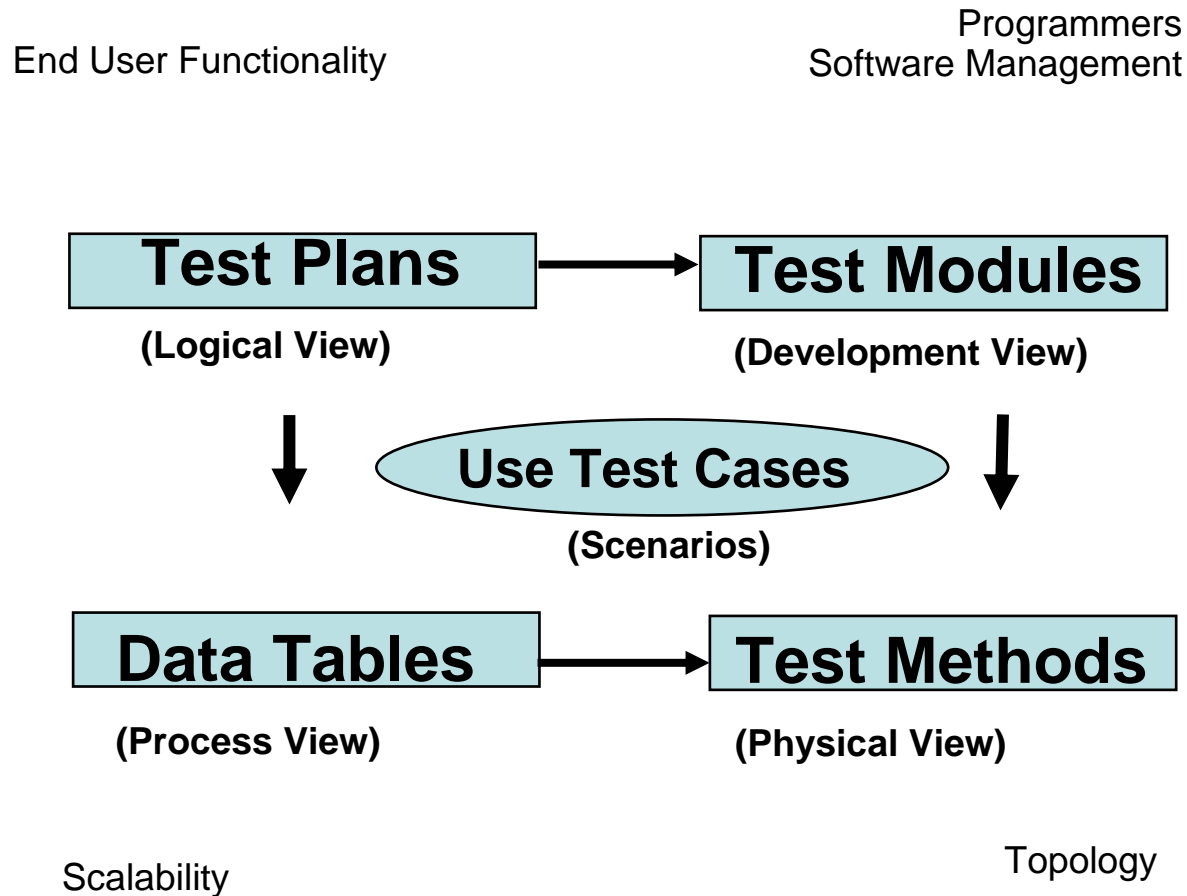
UI Controls	Set	Read	Verify
Pushbutton	✓	NA	NA
Edit Box	✓	✓	✓
Static Text	NA	✓	✓
Combo Box	✓	✓	✓
Radio Button	✓	✓	✓
Check Button	✓	✓	✓
Menu Item	✓	NA	NA
List Box	✓	✓	✓
List Table	✓	✓	✓
Window	✓	NA	✓
Tree List	✓	NA	NA
Stingray Grid	✓	✓	✓
Tab	✓	NA	NA

NA: Not Applicable

- **Basic operation for any UI control**
 - Set, Read and Verify
- **UI control class is derived from RationalTestScript Class and the basic operations are overridden in the derived class**



“4+1” view Architecture



Architectural Blueprints – The “4+1” View Model of Software Architecture
by Philippe Kruchten

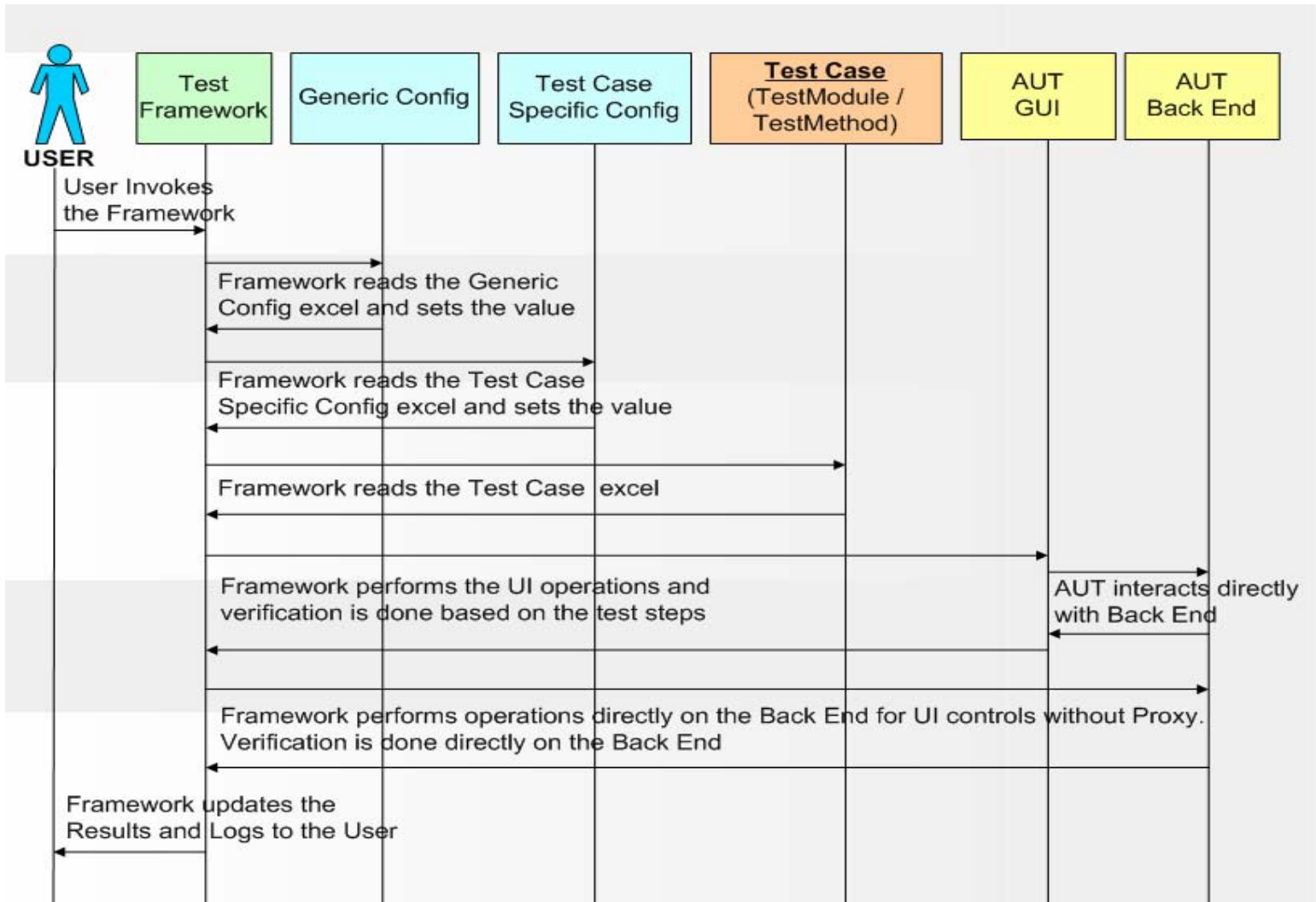


Architecture

- **Test Plan**
 - Represents a collection of test cases that will be executed sequentially.
 - Output is a result sheet indicating whether the selected test cases passed or failed.
- **Test cases**
 - Consists of a collection of Test Modules and Test Methods which are organized in a logical sequence.
 - Output is a result sheet indicating whether the individual Test Modules/Test Methods passed or failed.
- **Test Modules**
 - Readily available entity that performs commonly used operation.
 - Consists of a collection of Test Methods which are organized in a logical sequence.
 - Reused across Test Cases.
- **Test Methods**
 - One class implemented corresponding to each UI control.
 - Operations that have been implemented in the UI control classes in Java.
 - Makes use of RFT APIs and Java libraries



Test Framework Control flow



Test case script

Module/Method	Argument1	Argument2	Argument3	Argument4
#Set-up				
TestModule.LaunchProteusAndLogin	c2	c2		
TestModule.LaunchConfigurationEditor				
TestModule.SetConfig	argument3			
TestModule.LaunchTelnetWindow				
TestModule.Powercycle	stop all;auto			
TestModule.CloseTelnetWindow				
#####Step1 Wafer run and nested error recovery #####				
TestModule.AssignWafers	Port1	1	10	argument5
TestModule.StartWaferRun				
TestModule.WaitForModuleToExecuteStep	argument2	argument6		
TestModule.InduceError	argument2			
TestModule.VerifyModuleInError	argument2			
TestModule.OpenErrorRecovery				
TestModule.VerifyVCComboboxExists	errorrecovery_IDC_ER_MODULES_COMBO	Modules in Error	CONTINUE	
TestModule.SelectModuleForErrorRecovery	argument1			
TestModule.VerifyVCRadioButtonExists	errorrecovery_IDC_DYNAMIC_RADIO1	Resume	CONTINUE	
TestModule.VerifyVCEditBoxExists	errorrecovery_IDC_EDIT_DYNAMIC	-	CONTINUE	
#####Step2 #####				
...				
#Tear-down				
TestModule.ReplaceAllCassettes	hfkB			
TestModule.DeleteRecipe	argument5			
TestModule.ExitProteus				



Error Recovery System

- **Cascading failures are avoided by using an error recovery that brings the system to base state**
- **Generic Teardown is executed if the test cases encounter exceptions**
 - Take the screenshot
 - Log the failure details
 - Close all windows
 - Restart the application
- **Testcase dependent Setup & Teardown is implemented to make each test case independent**



Enhanced Logging

- Logs are captured using excel sheet and text file.
- Every level will have a log with results against each line in an excel sheet.
 - Test Plan Result, TestCase Result & TestModule Result
- A detailed log also will be available in the form of a text file.
- Log level is provided by the user at the test plan selection level.

Number of Test Cases Selected:	23	
PASS	TestCase.INXT_P1508_TC1_01	InovaxTSamsung
PASS	TestCase.INXT_P1508_TC2_01	InovaxTSamsung
PASS	TestCase.INXT_P1508_TC2_06	InovaxTSamsung
PASS	Variable.setLogLevel	TestCase.INXT_P1508_TC2_07
PASS	TestCase.INXT_P1508_TC2_08	InovaxTSamsung

Number of Test Plans Selected:	1	
PASS	2009-07-27 19:07:48 :	Test Module: LaunchProteusAndLogin
PASS	2009-07-27 19:07:49 :	Proteus memory usage before public boolean Scripts.Utility.startApplication(java.lang.String)
PASS	2009-07-27 19:07:49 :	Utility startApplication
PASS	2009-07-27 19:07:49 :	Argument: proteus
PASS	2009-07-27 19:07:49 :	Entering startApplication() function
PASS	2009-07-27 19:07:49 :	Application proteus Started
PASS	2009-07-27 19:07:59 :	Application started successfully
PASS	2009-07-27 19:07:59 :	RESULT =true
PASS	2009-07-27 19:07:59 :	Proteus memory usage after public boolean Scripts.Utility.startApplication(java.lang.String)
FAIL	2009-07-27 19:08:00 :	Proteus memory usage before public boolean Scripts.Window.waitForWindow(java.lang.String, java
FAIL	2009-07-27 19:08:00 :	Window waitForWindow
FAIL	2009-07-27 19:08:00 :	Argument: Login
Number of	2009-07-27 19:08:00 :	Argument: 60
Number of	2009-07-27 19:08:00 :	Entered waitForWindow() method
Number of	2009-07-27 19:08:08 :	Window Login was found after 7 seconds.
	2009-07-27 19:08:08 :	RESULT =true
	2009-07-27 19:08:08 :	Proteus memory usage after public boolean Scripts.Window.waitForWindow(java.lang.String, java
	2009-07-27 19:08:08 :	Proteus memory usage before public boolean Scripts.Window.activate(java.lang.String) 60296KB
	2009-07-27 19:08:08 :	Window activate
	2009-07-27 19:08:08 :	Argument: Login
	2009-07-27 19:08:15 :	Window Activated
	2009-07-27 19:08:15 :	Exiting function activate()

Number of Test Cases FAILED:	0	
------------------------------	---	--



Documentation using Java Doc for Modules and Methods

Class Summary

Backend	Description: Cla
CESetting	Description: Cla
Conditional	Description: Cla
FileOperation	Description: Cla
Installation	
ListBox	Description: Cla
ListTable	Description: Cla
MenuBar	Description: Cla
Navigate	Description: Cla
Recipe	Description: Cla
RightClick	Description: Cla
Tab	Description: Cla
TelnetWindow	Description: Cla
TreeList	Description: Cla
Utility	Description: Cla
Variable	Description: Cla
VCButton	Description: Cla

Scripts

Class Backend

java.lang.Object
 ↳ com.rational.test.ft.script.DatapoolScriptSupport

Method Summary

boolean	<p>getValue(java.lang.String machineIP, java.lang.String portNum, java.lang.String kbName, java.lang.String objectName, java.lang.String slotName, java.lang.String localRemote) Sends the required GETVAL message to the backend task</p> <p>Usage: <code>getValue("10.0.58.179","7000","sched","SCHD","stat","+");</code></p> <p>Default Arguments: NONE</p>
boolean	<p>setValue(java.lang.String machineIP, java.lang.String portNum, java.lang.String kbName, java.lang.String objectName, java.lang.String slotName, java.lang.String localRemote, java.lang.String argVal, java.lang.String argType) Sends the required SETVAL message to the backend task</p> <p>Usage: <code>setValue("10.0.58.179","7000","sched","SCHD","stat","+","Idle","4");</code></p> <p>Default Arguments: NONE</p>
boolean	<p>verifySlotValueInFile(java.lang.String machineIP, java.lang.String portNum, java.lang.String kbName, java.lang.String objectName, java.lang.String slotName, java.lang.String localRemote, java.lang.String fileName) Verifies if required slot value is present in the given QNX file</p> <p>Usage: <code>verifySlotValueInFile("10.0.58.179","7000","sched","SCHD","stat","+","/c2/log/sched/sched.log");</code></p> <p>Default Arguments: NONE</p>



Benefits with New Automation Approach

Project - Nested Error Recovery on INOVAxT	
Approach	Effort (ManDays)
Automation using old Framework	169
Automation using new Framework	63
Effort Savings	62%

- **Saving of 60% on efforts with the new approach.**
- **Test cases can be scripted without the application fully developed.**
- **Test case scripts are portable across operating systems.**
- **Maintenance of the test case scripts is negligible.**

7% coverage achieved using the new framework



Proxy Development

- **Proxy developed for Novellus Application**

- **Problem Statement:**

- ▶ Novellus software use a third party grid application (Rogue Wave Stingray).
- ▶ RFT was not able to recognize the GRID Control and recorded script was referring to co-ordinates.
- ▶ Our requirement was to recognize the cells and their row and column indices

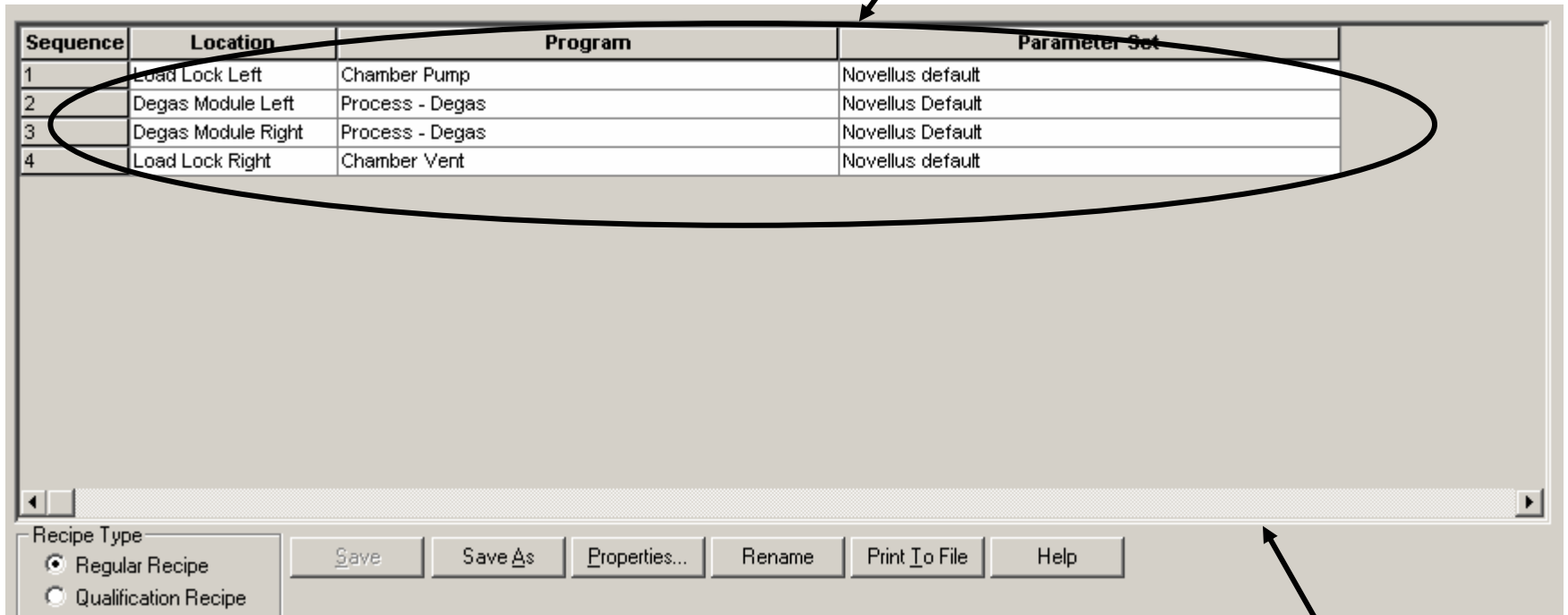
- **Solution:**

- ▶ GRID Control API are exposed by the third party
- ▶ Helper DLL was required for getting the window handle for the Grid and in turn calling Grid APIs
- ▶ We developed a Proxy DLL for the GRID Control by extending the Win.Generic proxy class.
- ▶ When the RFT recognizes the GRID control, Proxy DLL creates the Grid Test Object on the RFT



Stingray grid control

Stingray grid cells



The screenshot displays the Stingray grid control interface. At the top, a table lists four grid cells. Below the table is a control panel with a 'Recipe Type' section and several buttons.

Sequence	Location	Program	Parameter Set
1	Load Lock Left	Chamber Pump	Novellus default
2	Degas Module Left	Process - Degas	Novellus Default
3	Degas Module Right	Process - Degas	Novellus Default
4	Load Lock Right	Chamber Vent	Novellus default

Recipe Type

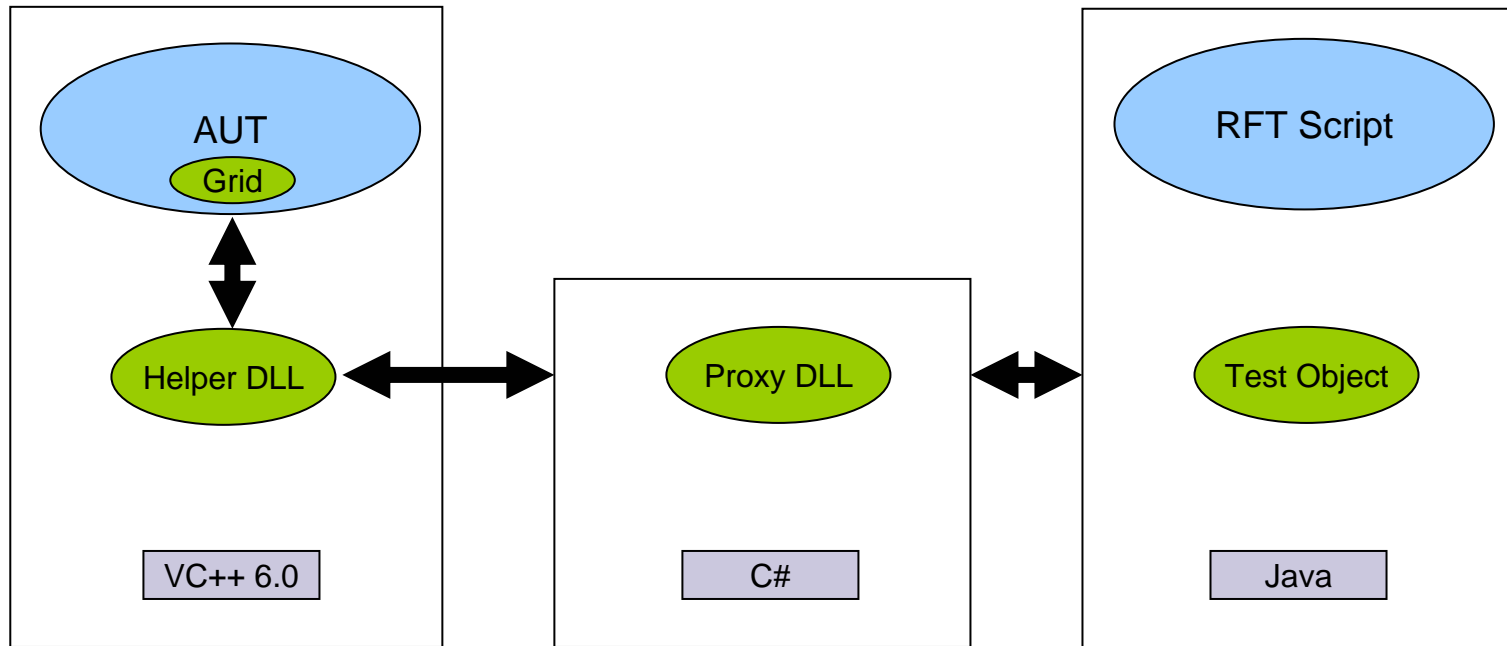
- Regular Recipe
- Qualification Recipe

Buttons: Save, Save As, Properties..., Rename, Print To File, Help

Stingray grid window

Proxy Development

- Proxy developed for Novellus Application



Summary

- **Test cases can be written faster to support quick turnaround.**
- **Everyone can write test cases, does not require programming skills.**
- **Framework is used to quickly write test scenarios to re-create customer issues**
- **Enables re-use of test methods and modules since the design is modular**
- **Java Doc helps in easy scripting**
- **Robust design and error recovery system**
- **Proxy SDK provides freedom to develop proxies for custom controls**





Thank You

© Copyright IBM Corporation 2009. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, Rational, the Rational logo, Telelogic, the Telelogic logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.



Helper DLL code snippet

```
include "stdafx.h"
#include "NovDll.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
/*
Returns the row count of Stingray grid
*/
int GetGridRowCount(HWND hWnd)
{
    CString rowCount;
    CGXGridWnd *gridObject = NULL;

    gridObject = (CGXGridWnd *)CWnd::FromHandle(hWnd);

    return gridObject->GetRowCount();
}
/*
Returns the column count of Stingray grid
*/
int GetGridColCount(HWND hWnd)
{
    CString rowCount;
    CGXGridWnd *gridObject = NULL;

    gridObject = (CGXGridWnd *)CWnd::FromHandle(hWnd);

    return gridObject->GetColCount();
}
```



Proxy DLL code snippet

```

using System;
using System.Runtime.InteropServices;
using Rational.Test.Ft.Domain.Win;
using Rational.Test.Ft.Object;
using Rational.Test.Ft.Script;
using Rational.Test.Ft.Object.Interfaces;
using Rational.Test.Ft.Object.Manager;
using Rational.Test.Ft.Object.Map;
using StingrayControls;

namespace Rational.Test.Ft.Domain.Win.Stingray
{
    public class StingrayGridProxy :GenericProxy
    {
        public static int x;
        public static int y ;
        public StingrayGridProxy ( Rational.Test.Ft.Domain.Win.WinTestDomainImplementation domain
Domain.Win.IWinControl theTestObject )
            :base(domain,channel,theTestObject)
        {
        }
        // import DLLs
        [DllImport("NovDll.dll", EntryPoint = "GetGridRowCount", CharSet = CharSet.Ansi, CallingCo:
static public extern int GetGridRowCount(System.IntPtr hWnd);

        [DllImport("NovDll.dll", EntryPoint = "GetHitTest", CharSet = CharSet.Ansi, CallingConvent
static public extern int HitTest(System.IntPtr hWnd,int x,int y,ref int ncRow, ref int ncC
        // for playback
        [DllImport("NovDll.dll", EntryPoint = "GetRectFromRowCol", CharSet = CharSet.Ansi, Calling
static public extern void GetRectFromRowCol(System.IntPtr hWnd,int row,int col,ref System.:

```



Proxy DLL code snippet contd...

```
public void Click(Subitem subitem)
{
    if(subitem is Cell)
    {
        Cell cell = (Cell)subitem;
        Row row = cell.GetRow();
        Column col = cell.GetColumn();
        x = row.GetIndex().GetIndex();
        y = col.GetIndex().GetIndex();
        System.Drawing.Point screenPt = ((IWinControl)theTestObject).PointToScreen(new System.Drawing.Point(0,0));
        System.Drawing.Point point = new System.Drawing.Point(row.GetIndex().GetIndex(),col.GetIndex().GetIndex());

        GetRectFromRowCol(((IWinControl)theTestObject).HWND,row.GetIndex().GetIndex(),col.GetIndex().GetIndex(),ref point);

        base.Click(point);
    }
    else
    {
        base.Click (subitem);
    }
}
```


Test Object

The screenshot shows a window titled "Private Test Object Map for Script FinalVersion_With_GetSet". The window has a menu bar with "File", "Edit", "Find", "Test Object", "Preferences", "Applications", "Display", and "Help". Below the menu bar is a toolbar with various icons. The main area displays a tree view of test objects:

- Win: Window: _RecipeEditorRecipePPID1: Afx:400000:b:10011:6:d0adb
 - Win: Window: AfxMDIFrame42: AfxMDIFrame42
 - Win: Window: _32770: #32770
 - Win: Window: Slwre_Grid: GXWND (highlighted)

Below the tree view is a table with two tabs: "Recognition" and "Administrative". The "Recognition" tab is selected, and the table contains the following data:

Property	Value
Descriptive Name (#name)	Slwre_Grid
Map ID (read only: #id)	G.gL2FmJ8VAP6:1mggF2:LlnKIws:8WW
Proxy Class Name (#proxy)	.Win.Stingray.StingrayGridProxy
Role (#role)	Window
Test Domain (#domain)	Win
Test Object Class Name (#testobject)	StingrayTestObject

RFT script

```
import resources.StngrayTestHelper;
import SDK.Stingray.StingrayTestObject;

import com.rational.test.ft.*;
import com.rational.test.ft.object.interfaces.*;
import com.rational.test.ft.object.interfaces.SAP.*;
import com.rational.test.ft.object.interfaces.WPF.*;
import com.rational.test.ft.object.interfaces.dojo.*;
import com.rational.test.ft.object.interfaces.siebel.*;
import com.rational.test.ft.object.interfaces.flex.*;
import com.rational.test.ft.script.*;
import com.rational.test.ft.value.*;
import com.rational.test.ft.vp.*;

/**
 * Description   : Functional Test Script
 * @author ChakraS
 */
public class StngrayTest extends StngrayTestHelper
{
    |
    public void testMain(Object[] args)
    {
        RootTestObject root = getRootTestObject();
        TestObject[] to = root.find(atDescendant(".class", "GXWND"));

        StingrayTestObject grid = new StingrayTestObject(to[0]);
        grid.getCellValue(1, 0);
        grid.setCellvalue(1, 0, "PVD1");
    }
}
```



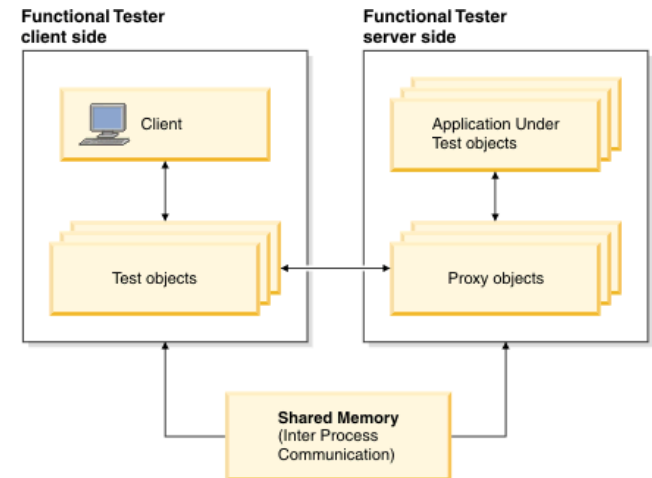
Proxy Development

- **Why Proxy is required?**

- Every test object within RFT framework has a corresponding proxy
- Proxies need to be developed for controls that are not recognized by RFT
- Proxies can also be used to expose hidden properties of controls even for controls recognized by RFT

- **What is a Proxy?**

- Proxies are interface between the UI controls and RFT
- Proxy objects can interface with the UI controls using native APIs supported by the UI control
- Test objects interact with proxy objects



Proxy Development

- **How is a Proxy developed**

- Proxy Development requires the Proxy SDK provided by RFT.
- Proxies can be developed either in Java or C#.
- Decide upon the control for which we need to develop the proxy and the functionalities required out of that.
- Generic proxy class is extended and mapped to the actual class corresponding to control object

