Assuring Certainty through Effective Regression Testing

Vishvesh Arumugam

TATA CONSULTANCY SERVICES

TATA

Agenda

- Introduction
- The Problem
  - Magnitude
  - Management
  - Regression Test Efficiency
- Solution and Approach
  - Test Suite Maintenance
  - Determining Test Sets
  - Test Sequencing & Scheduling
  - Regression Test Automation
- Question and Answers

Introduction - Regression Testing

What?

- *Regression testing* is *selective retesting of a system or component to verify that modifications have not caused unintended effects and that the system or component still complies with its specified requirements*.

Why?

- Every time code is modified or used in a new environment, regression testing should be used to check the code's integrity.

- Detect unexpected faults

Introduction - Regression Testing

When?

- After each new version of a product is created.
  - Planned Releases
  - Emergency Releases
- Often Late in the Lifecycle
  - Need to pass other tests
  - Last step before delivery

The Problem

Magnitude

- Overall Suite Size grows over time
  - Test Suite Maintenance
  - Determine Regression Test Set

Management

- Test Cycle Time
  - Schedule
  - Sequence
- Budget

Efficiency

- Regression Test Efficiency

The Problem & Solution Approach

Magnitude

- Overall Suite Size grows over time
  - Test Suite Maintenance
  - Determine Regression Test Set

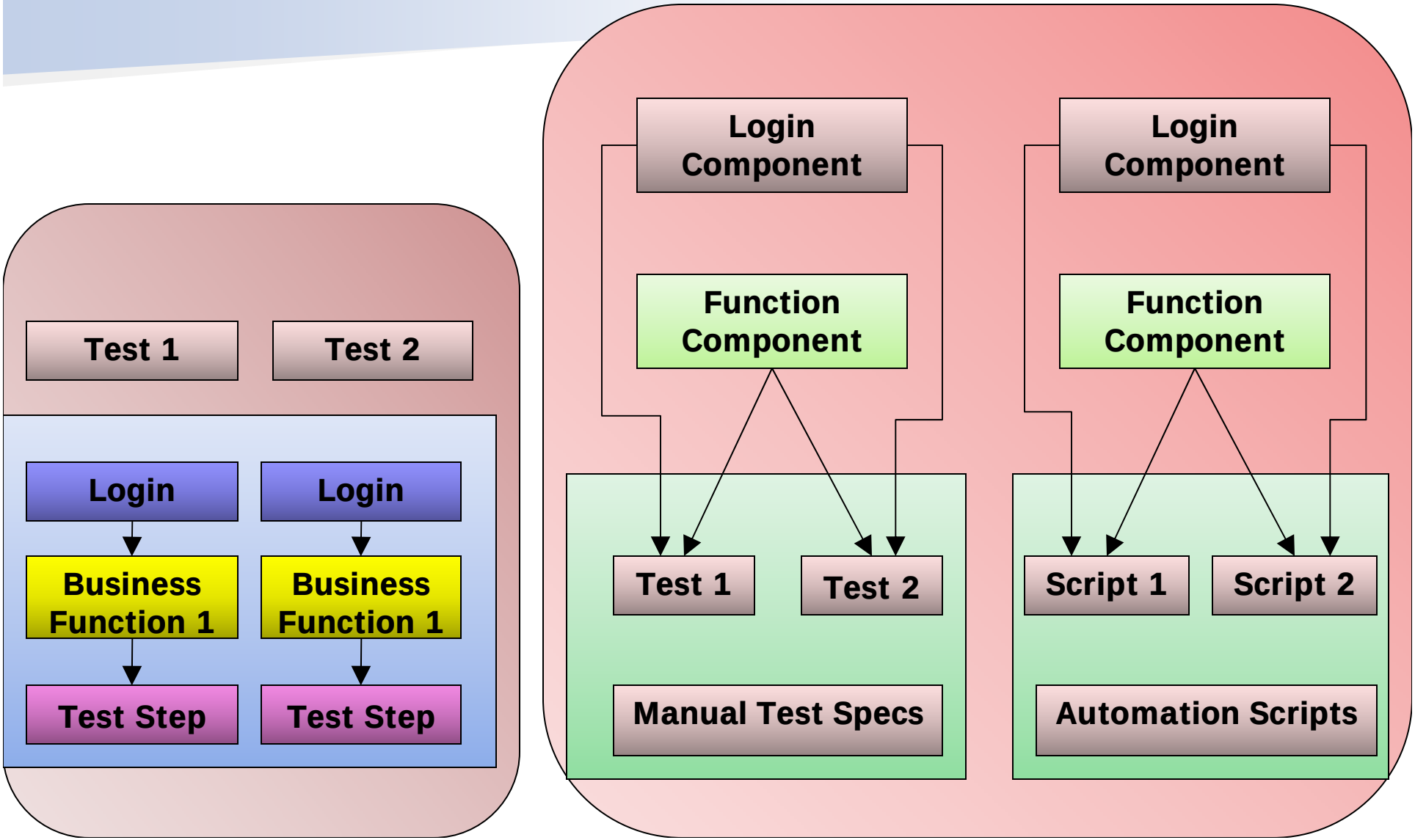
Solution

- Test Suite Maintenance
  - Component Based Architecture
- Determine Regression Test Set
  - Strategy 1 – Regression Impact Matrix
  - Strategy 2 – Core Regression Set
  - Strategy 3 – Smoke Testing

Test Suite Maintenance : Component based Architecture

- Creation of Reusable Test Case and Script Components
- Invoking them from the Test Case and Automation Scripts

Test Suite Maintenance : Component based Architecture



Component Based Architecture - Benefits

- Reduce the effort to maintain Test Components to keep up with changes to Functionality.
- Pre-Built Test Components can be used to build New Test Cases, there by minimizing the Pre-Validation Effort.
- Accelerates the implementation of upgrades and new installations while significantly reducing testing cycle.
- Collaborative Approach Between Component based Framework for Test Design and Automation Framework.

Enforces Modularization	**Reduces Maintenance**

Regression Set Strategy

- Strategy 1 – Regression Impact Matrix
  - Set consists of test suites for areas directly impacted by new Feature or a code change
  - Small subset of tests covering basic functions of all the features
- Strategy 2 – Core Regression Set
  - Executed irrespective of the number and type of bug fixes
  - Set consists of test suites that cover customer sensitive and critical functions
- Strategy 3 – Sanity Set
  - Sanity testing prior to deployment of software release to customers

Strategy1 : Regression Impact Matrix

New Functions or Code Changes \ Existing Features	Func 1	Func 2	Func 3	Func 4
New Function 1		D		R
Change Function 2	R		D	
New Function 3	D			
Change Function 4		D		D

- Create Regression impact matrix with existing features on Row 1 and the New or change Functions in Column 1
- **D** – Direct Impact and **R** – Related Impact
- Design and Development team working on this new or change function prepares the matrix
- D - Prioritized first and R - Prioritized low

Strategy 2 : Core Regression Set

Criticality of the Function

- Critical Function Survey is done among the User group
- Identified based on Historical statistics of Hits, Issues
- Inputs from the Infrastructure and support team

Defect Prone Areas

- Defect Statistics report in previous testing cycles
- Discussion with the Application Support team
- Defect Reports from Production environment

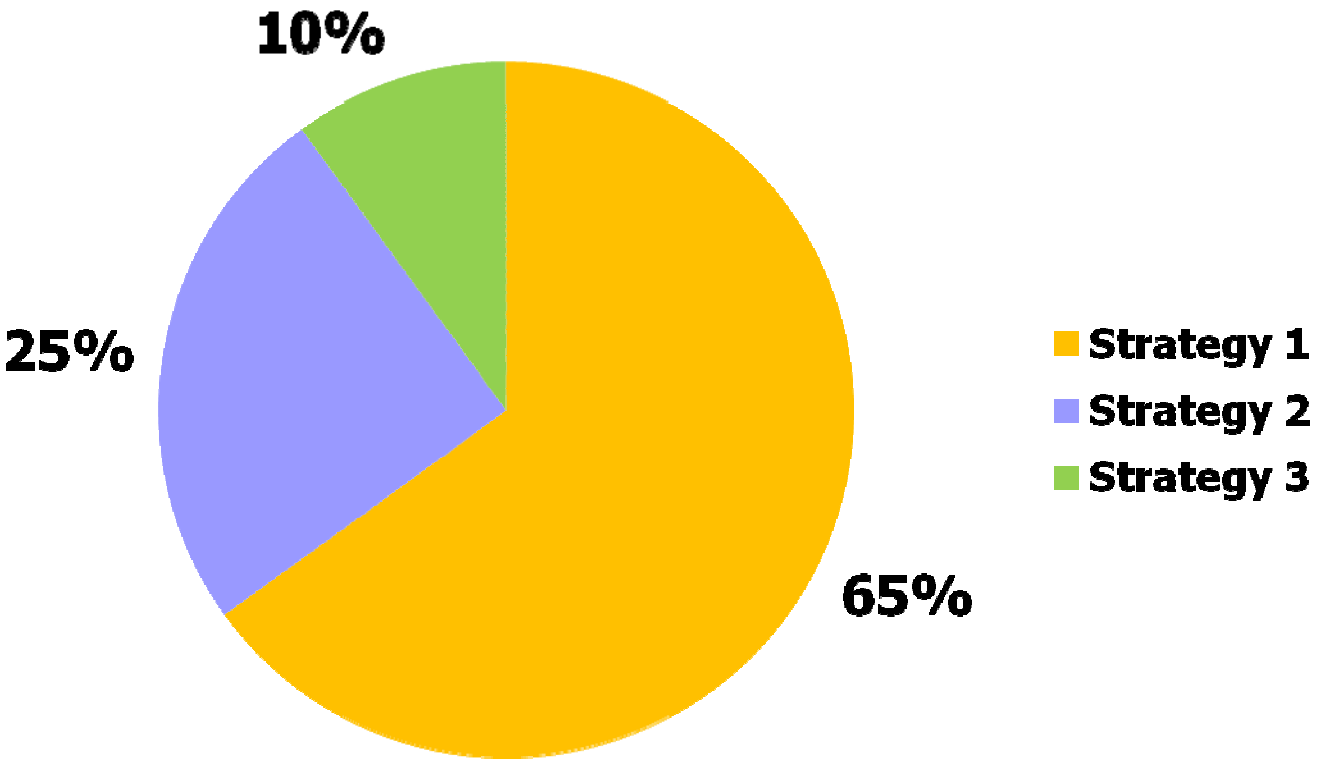
Frequent Code Change

- Frequent Code change areas are identified based on the reports from Configuration Management team
- Discussion with the Development team

Functional and Technical Complexity

- Functional complexity are identified and ranked based on the Functional documents.
- Discussion with Business and Technical team

Regression Set Components



The Problem & Solution Approach

Management

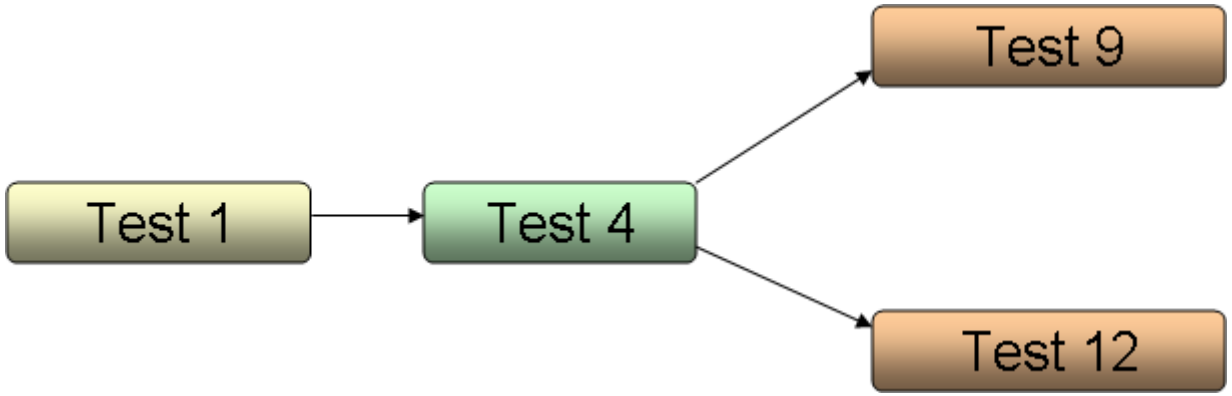
- Test Cycle Time
  - Sequence
  - Schedule
- Budget

Solution

- Sequence
  - Test Case Array
- Schedule
  - Run Plan
- Regression Automation

Test Case Array

- Identifying and ordering based on the logical relationship between the test cases
- Test sequence attribute in Test cases indicate the previous Test spec to be executed



Run Plan

- Validate High Risk Items First
- Create a Run Plan based on Prioritization and sequencing
- Scheduling the Automation Execution

Day	Date	Test Case Number	Responsible Tester	Status
1	2/8/2009	Test Case 1	Tester 1	Completed
		Test Case 4	Tester 1	Completed
		Test Case 9	Tester 1	Completed
		Test Case 12	Tester 1	Completed

Regression Test Automation

- Rapid validation of software changes
- Enables to achieve detailed product testing with significant reduction in test cycle time.
- automated testing incorporated into product lifecycle can generate sustainable time and money savings.
- Automated testing increases the significance and accuracy of testing
- Automation eliminates many of the mundane functions associated with regression testing.

Questions

Thank you

Email : vishvesh.arumugam@tcs.com