

## Hindustan Unilever Limited Testing Solution

### Business Case

CMC has implemented Automation Testing Solution to HUL using SILK TEST 2006 Release 2 on RS Unify Application, which is a billing package that completely automates all operations of a Redistribution Stockiest (RS) of HUL. Unify is a mini-ERP solution, that resides in RS offices and communicates with a central server (called Central Unify) to receive all HUL information pertaining to his operations – such as product, price masters, Schemes etc. The Unify product is a Client –Server based application developed using the technology DELPHI and MSSQL2000. SILK TEST was the only tool which was supporting RSUNIFY Application, rest all the tools such as QTP, Rational Testing Tool, and Win Runner etc. failed.

CMC successfully implemented the testing solution to Hindustan Unilever Limited by demonstrating the Proof of Concept using the tool.

### Background

Hindustan Unilever Limited is the top FMCG Company in India. Redistribution Stockiest (RS) of HUL makes use of this RS Unify application for there day to day transactions process and reporting operations. HUL has the strategy of implementing new business process in the market for the growth of the company. Hence HUL implements these business strategy into IT solutions which leads to a release strategy every 3 months once to the field. The information/data received from RS Unify application plays a vital role for statistical analysis of HUL business in the market.

Hence accuracy and quality of data is expected from the application, which in turn triggers for bug free release of the product.

RSUNIFY application comprises:

- 250+ User Interface Screens (GUI) with Delphi Front End.
- 70 Master Tables, 170+ Transaction Tables, 150 PBAY Tables which act as an interface table between the systems RSUnify and Central Unify for push-pull of data.
- Core Batch Processing modules around 8- 10.

### Challenges

In all the cases below, release may or may not always be stable.

- The product went through customisation for varying requirements from business and the users.
- Large amount of manual regression testing was used to ensure the product stability for the users.
- Continuous enhancements to the product grew in sizes which lead to regression testing as a challenge.
- Business and the users expected quick turn around time in receiving a stable IT solution that would suffice their specific business process.

Quality delivery of the product to the field was a prime challenge to the customer. Hence to overcome this challenge the need of automation for regression testing became a key factor.

To overcome the above challenges, CMC identified the tool which supports HUL application – RSUNIFY, demonstrated and implemented the features of the tool and proved how the tool can be full-fledged used during regression testing.

## CMC's Solution and Implementation

### Borland Silk Test Release 2008 Overview

Silk Test offers a complete solution for the automated testing of Web and Client/Server application. With silk test we can prove

- ✓ Functional, operability and regression testing.
- ✓ Test planning and management using Test Plan editor.
- ✓ Database validation using DBTester.

### General Application Testing Features

- ✓ Organization of resources within a project.
- ✓ Guidance via workflow bars.
- ✓ Automatic completion within editors.
- ✓ Easy recording of tests.
- ✓ True object recognition.
- ✓ Unattended testing with built-in recovery system.
- ✓ Fully integrated test plan creation and maintenance.
- ✓ Database testing capabilities.
- ✓ Full distributed testing.
- ✓ Browser and platform independence.
- ✓ Technology Independence.
- ✓ Multiple testing options using browser extensions.

### Tool Architecture



### Work Flow Approach

- ✓ Initiated with training to understand the product.
- ✓ Understood the domain by discussions and clarifications sought from the client and functional team.
- ✓ Documented our understanding, converted them into requirements and placed them in Excel (Functional Test Plan).
- ✓ Agreed upon modules that are critical and needs to be automated in phase-1
- ✓ Identified resources (team lead/engineers) with skills in the tool and who would fit working on this specific domain.
- ✓ Developed test cases based on functional documented (In the process arrived at high coverage test cases covering negative testing and multiple work flows)
- ✓ Verification of the content of the test cases was carried out by reviews (peer and client) to identify the gaps.
- ✓ Once the functional test plan and test cases got signed off from the customer, automation process would get initiated.

### Technical Approach

- ✓ Evaluated multiple tools that would suite the product automation.
- ✓ Provided a Proof of Concept to the client and suggested an industry standard tool to procure.
- ✓ Identified critical modules and respective test cases to be automated on priority in concurrence with the client.

- ✓ Designed an automation test plan and framework. Discussed the same with the client.
- ✓ Used the “Functional Decomposition Approach” for automation.
- ✓ Delivered an automation test plan detailing out the scope, approach, effort, schedules/milestones and deliverables.

#### **Automation Framework**

- ✓ Preparation of test plan and framework.
- ✓ Identified all common functions.
- ✓ Separated “Object Names” from the scripts to ensure adaptability to UI changes and product enhancements/modifications.
- ✓ Identified and organized test data (in the form that is scalable for enhancement).
- ✓ Used Excel spreadsheets for data preparation.
- ✓ Auto executing the test cases by marking and running the test plan.
- ✓ Created the Infrastructure and configuration required for executing automated scripts.
- ✓ Test results stating Pass/Fail report used to be generated.

#### **Deliverables**

- ✓ Automation test plan
- ✓ Automation framework/design documents
- ✓ Automated test execution summary report
- ✓ Defect summary reports
- ✓ Progress monitoring/status reports
- ✓ Automated test coverage reports

#### **Benefits to the Customer**

- ✓ Reduction in manual testing effort (No of man days saved per release).
- ✓ Reduction in time by marking the test cases for auto runs during night hours.
- ✓ Zero bug release.
- ✓ Identification of cascading effects due to bug fixes /enhancements.
- ✓ Reduction in dependency on manual test cases and testing