

AUTOMATED TEST DESIGN HELPED A MAJOR INSURANCE COMPANY ACHIEVE OPTIMIZED AND HIGHER SOFTWARE QUALITY

### **ABOUT THE CLIENT**

The client is a major insurance company in the United States that provides direct-response home and car insurance. The client was looking for a better way to achieve known test quality while lowering their costs.



#### Automate the Automation

# CONFORMI



### PROJECT BACKGROUND

The client company was exploring new tools and techniques to improve its existing manual test case and manual script design process for test automation on both waterfall and agile development projects. Their TCoE (Testing Center of Excellence) was tasked with determining if there were commercial tools that would deliver the ability to transform the company's existing testing process into a more formal and consistent testing process to improve or at least quantify known test coverage as well as reduce time to product release. Their quality need was very critical as errors missed in their testing coverage could mean significant financial losses.

After exploring the space of test design tools, Conformiq Creator<sup>™</sup> was selected for deployment. Their selection of two different interfaces to test mirrored their actual testing needs. They tested for both optimized data combinations as well as optimized requirements test coverage through the entire application flow from GUI to mainframe.

### **TESTING CHALLENGES**

The client's major testing challenges were the typical issues resulting from using manual test design and script creation for automated test execution – test case design takes longer; application design changes delay test case creation resulting in delayed product release; regression tests deliver spotty coverage and take longer to write executable scripts, together causing most tests to be manually executed even though they have automated execution tools in-house.

## THE APPROACH

The selected projects were staffed by test automation specialist with a test analyst/Subject Matter Expert (SME) available as needed to explain the application's operation. A Conformiq support specialist helped with best practice modeling recommendations and answered tool use questions. They worked with the SME in complex areas of the application that required detailed knowledge of the way the application worked. Models of the system operation were developed iteratively as the team increased its understanding of the application as explained by the SME.

The approach was intended to mimic their deployment "run-rate" efficiency use of Conformiq and the MBT process once their start up period ended in the expected 1-2 months. Using Conformiq Creator, the TCoE team created a model of the application from scratch using the tool's structure and activity diagrams. These diagrams represented the user interface operations (button clicks, data fill forms with data ranges, etc.), plus the flow of the application, respectively, at an abstract level.

This proved to be easier than initially expected due to Creator's ready to use library elements that graphically represented the actual application operation, not the test flow of the application. This was an important detail they hadn't previously considered. Truly modeling the application from the user stories and/or requirements instead of the test flow meant modeling was fast and models were easily reusable.

Once the models were done the test cases were automatically generated, along with the test steps and expected test validations at these points. The generation of tests for data coverage with pairwise optimization was matched against optimized test requirements coverage generation and both were compared to manual design efforts for completeness and quality. Tests were uploaded to HPE (now Micro Focus) ALM.

In a full end-to-end test process flow, the next step would be linkage with and automated execution by the selected test execution tool or tools directly from the Creator generated executable test scripts and test execution validation results.



### **BUSINESS OBJECTIVES**

- Automated test case generation
  - o Optimized test cases
  - o 100% known requirements coverage
  - o Readable test step format for manual testers and stakeholder review
  - o Documentation of test cases and application coverage
  - o Requirements traceability to every test case
  - o Integration with HPE ALM and UFT
- Easy to modify and maintain process
- Minimal tester involvement in the automated test design process
- Repeatable process to replace current ad-hoc methods
- Faster and more accurate than manual test design
- Use cases included the following characteristics for the two projects:
  - o Complex detailed process driven API
  - o Mainframe IBM 3270 user interface

## THE SOLUTION

Conformiq Creator, part of the Conformiq 360° Test Automation solution, enables an end-toend automation process that starts with the creation of a model which is a graphical description of the application being developed using the requirements and user stories. Directly from the model, all functional test cases, documentation, executable scripts and validations are automatically generated in one automated step without user involvement except to select the test design algorithms to be used.

The Conformiq Creator modeling software was used to capture the application's operation and system flow from requirements. Inconsistencies between requirements and the model were automatically flagged, so when these graphical models were reviewed, logical process errors were easily and quickly identified, often before application coding was completed for a sprint. Data was included for the automatically generated positive and negative test cases.

Once modeled, the Conformiq test generation engine automatically generated an optimal test suite based on the user-selected test design algorithms in order to achieve 100% coverage of requirements. These results were analyzed by reviewing the generated test step reports, message sequence charts, test-torequirements traceability matrix, and the graphical model coverage diagram. The Conformiq engine automatically split the model across all available computation processors, which sped up the process of test generation for real world model size and complexity. With design changes, the model was extended as needed for additional features which generates an impact analysis of changes to test cases specifying which were unchanged, which were new, and which were then obsolete for execution. Thus the regression test pack was also automatically optimized.



Figure 1: Three Step Process for Test Design

## THE SOLUTION

The three step process shown above shows modeling, generated test case review and test scripts for automated execution, including the expected correct test execution results (test oracle).

The process of incrementally creating models and adding to existing models to test as development progresses supports the incremental creation process in agile development and allows for continuous integration, which is ideal for use with tools like Jenkins. This was not a focus area during this project, but it is an additional benefit from Conformiq's 360° Test Automation approach.

The project's goal was to create check points to validate the automated test generation work. Previous manual efforts to complete the tasks in the goals list were matched against the results automatically generated by Conformiq. The main goal of this work, known test quality with lower testing costs, were easily demonstrated by Conformiq's fully automatic generation of the stimuli with the correct expected test execution results, along with complete documentation of the test coverage for each test case.

### **CLIENT BENEFITS**

With Conformiq Creator, the client was able to fully documented test coverage plus witnessed an 82% cost reduction. The process consisted of modeling and then automatically generating test cases for the initial operation of the application and then making changes to the application and automatically regenerating test cases showing the impact of the change (which test cases were unchanged, which were new, and which became obsolete).

Category	Pairwise Method Optimized Coverage M		Manual Process	
Total test cases	125	59	125	
% coverage	95%	89%	95%	
Complexity	1497	814	NA	
Generation time	31 min	20 min	NA	
Modeling time	44 hours		NA	
TOTAL TIME	54.5 hours	54.3 hours	250 hours	

#### 1a – Quick Quote process – initial model

Figure 2: API Test Case Design Time Comparison

#### 1b - Quick Quote process - application change by adding a new state

Category	Pairwise Method Optimized Coverage		Manual Process	
Total test cases	125 + 13 = 138	125 + 13 = 138 NA		
% coverage		NA	NA	
Complexity		NA	NA	
Generation time	35 min	NA	NA	
Modeling time	3 min		NA	
TOTAL TIME	38 min	NA	NA	

Figure 3: API Test Case Update Time Comparison

## CLIENT BENEFITS

#### 2a – Cancel Coverage user interface – initial model

Category	Pairwise Method Optimized Covera		Manual Process	
Total test cases	60 30		60	
% coverage	100% 100%		100%	
Complexity	100 100		NA	
Generation time	6 sec	4 sec	NA	
Modeling time	3 hours		NA	
TOTAL TIME	3 hrs 6 sec	3 hours 4 sec	60 hours	
UFT Script	2 min + 30 min	NA	8 hours	
<b>Generation Time</b>	mapping			

Figure 4: UI Test Case Design Time Comparison

2b – Cancel Coverage user interface – application change if certain state is selected then prompt for additional information

Category	Pairwise Method	Optimized Coverage	Manual Process	
Total test cases	60 + 60 NA		NA	
% coverage	95% NA		NA	
Complexity	100 NA		NA	
Generation time	6 sec	NA	NA	
Modeling time	5 min		NA	
TOTAL TIME	5 min 6 sec		NA	
UFT Script Generation Time	NA	NA	NA	

Figure 5: UI Test Case Update Time Comparison

As can be seen from the above results, the model creation was proven to be substantially faster when existing test assets can be reused. In this case, the existing models were changed to match new design changes and updated test cases and the impact analysis were automatically re-generated.

#### SAVINGS ANALYSIS

Below is the summary of the aggregated time savings achieving an 82% overall test design time reduction. It is important to note that modeling an original process (Quick Quote example) is a onetime effort while changes made to every project occur many times every year over the life of the product. On an annual savings basis, the time savings for each change (Cancel Coverage example) could be looked on as a monthly (or even more often depending on revision frequency) change savings. Thus the 95% savings achieved would be at least 12X more important and then a weighted average would bring the projected savings to be almost 94% instead of 82%.

Use case name	Manual pairwise	Conformiq pairwise	Savings	% Savings
Quick Quote (Process)	250 hours	54 hours	196 hours	78%
Cancel Coverage (User Interface)	60 hours test cases + 8 hours UFT scripting = 68 hours	3 hours + 30 min = 3.5 hours	64.5 hours	95%
TOTAL	318 hours	57.5 hours	260.5 hours	82%

Figure 6: Overall Comparison of Test Time Savings

## KNOWN TEST COVERAGE FOR IMPROVED QUALITY

The second part of the insurance company's test improvement goals was to improve the quality of their application software by knowing what was and wasn't covered by their testing process. With manual testing, it was difficult to know if all paths and data combinations were tested. This was especially true for the negative paths through an application. It was exceedingly difficult to know if complex data dependent process changes and loopback flows were fully tested. Just determining the expected test validations (expected correct results from automated execution) was an added difficulty for testers, often leading to just testing happy path scenarios.

Conformiq Creator also solved this need by generating graphical coverage reports documenting the exact path each test case took through the model. Note that when the model is reviewed by the development team early in the process, it becomes the visual representation of the actual application at a high level of abstraction. Thus it accurately represents the application (flow and data) itself and thus the test coverage does reflect actual SUT (System Under Test) test coverage.

### SUMMARY

The key project benefits demonstrated by the insurance company from using Conformiq's MBT-based tool process were:

- Reverse engineer existing test assets reusability
- Visualize functional flow of the SUT find issues first
- Computer generated test cases/scripts/validations faster testing
- Optimized minimum test cases faster execution
- Progression testing during sprints reduced time to market
- Known test coverage and traceability greater confidence
- Design change impact optimized regression suite
- Integration with SDLC tools end-to-end automation
- Formalize testing process consistent and repeatable testing

In delivering the above testing process benefits, the biggest benefit to this company was cutting their testing time and costs almost in half. To aid their transition, Conformiq provides training in the use of its software as well as best practices to ensure good use. These types of best practices learning include modeling for reuse, best modeling practices, and staffing plans for the reduced personnel and shortened testing time.

WWW.CONFORMIQ.COM

@2019 ALL RIGHTS RESERVED