CONFORMIQ

Case Study:

Financial Institution Deploys Conformiq 360° Test Automation to Test at the Speed of Agile Development

Background

One of the world's largest financial companies needed to reduce its time to product release. In an industry with an ever-changing set of rules, financial regulations, and microeconomic changes, this company knew that their ability to test complex, new applications quickly and effectively was critical for both current and long-term success. To this end, they decided to adopt an agile development process for new projects, with the expectation that an agile approach with next-generation software testing methodologies would help them deliver better quality products to market faster.

The company's previous testing processes included manually transferring and saving all test scripts for execution, and different projects used different execution frameworks. Test and requirements management tools were consistent throughout the company, but other testing processes varied by project. Because there was concern that manual testing methods would struggle to keep pace with the continuous build cycles and short sprints of Agile, the central testing group explored several testing alternatives, including model based testing (MBT). This group knew that whatever tool and process they adopted needed to have the capability to thoroughly test complex operations, scale for large applications, fit into an agile development process, generate detailed test documentation, and be able to test both back-end operation and user interfaces. After reviewing commercially available tools, the Conformiq 360° Test Automation solution was selected based on their assessment that it best fit their criteria.

Their initial evaluation results, with Conformiq software, are shown in the table below.

Back-end Functions	Manual	Conformiq
Modeling/test design	13 days	1.5 day
Test case creation	23 days	14 days
Test harness interface	N/A	0.5 day
Test cases created	79	33
Requirements coverage (manually created)	100%	>100%

Figure 1: Conformiq results compared with manual methods

Based on these results, the company decided to start a live agile development project for their functional testing using the Conformiq 360° Test Automation approach.

System Under Test

The company's business is packaging real estate mortgages for resale. Accuracy of the calculations in their software application is critical to their profitability. Additionally, it must be released for use on time to prevent missed market opportunities. The project they selected for testing was a loan delivery application. Due to the fact that loan risks constantly vary, the task of calculating the correct pricing for mortgage loans presents an ongoing challenge. Still, it is an absolute necessity that the applications used to calculate loan numbers are accurate, because billions of dollars are involved. To accomplish this, the company needed to test that their application worked correctly versus the specification, and they needed deterministic testing because of the nature of their financial software. The specific application being tested was for BRE (Business Rule Environment), a new uniform interface front-end for applications calling the actual business rule engine. Their BRE has two components: the UI and the computation rules based back-end.

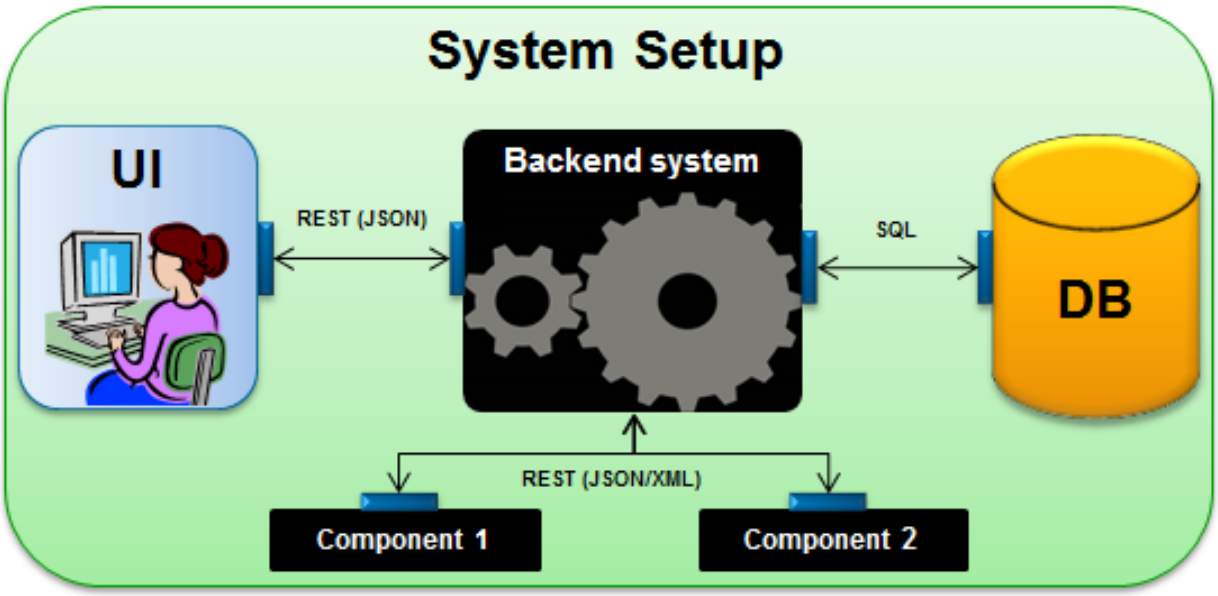

Figure 2: End-to-end system operation for the BRE in the loan delivery application software

Project Overview

This project had an 18-month development timeline for all modules in the entire new system, with an overall team of 50 business analysts, developers and testers. It featured:

Agile

- 2 week sprints, with 4 week development drops
- 15 development drops + final system integration + acceptance test

Test execution:

- HP / QTP for UI + Java and Junit for back-end
- Jenkins for continuous delivery

Project Deployment

Following the Conformiq 360° Test Automation methodology, a model was created for each of the two components: the UI and the backend business rules operation. This modeling included an abstract in-model database to mirror the database of the real system. Reusable components were created at the model level for use across both groups. The modelers were an integral part of the development teams and attended all the development meetings.

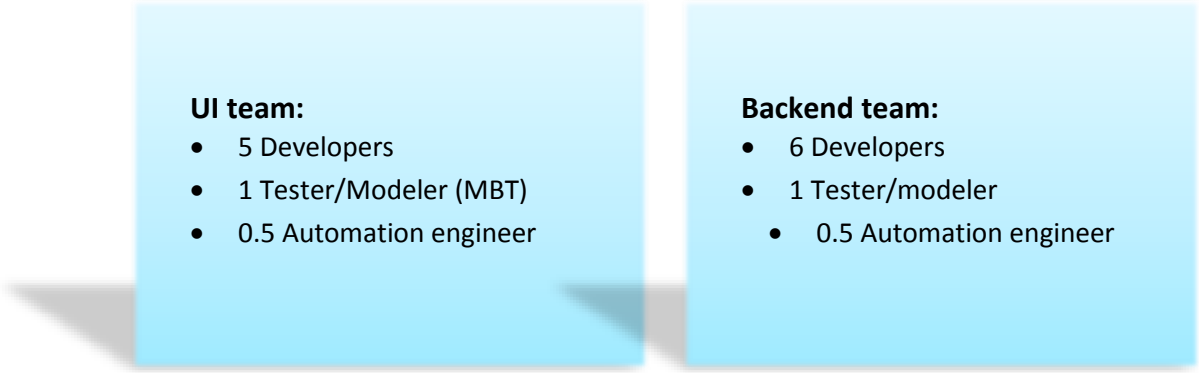

Figure 3: UI and Back-end Teams

The developers spent the first week of each sprint deciding what new functionality to include, and fixing previously found defects. The second week was spent developing code, during which time the modelers extended the existing model and generated new test cases for automated execution of the code with the new enhancements. The test execution harness and its function library were updated as required during this sprint, so it was ready for immediate automated test execution at each incremental code drop. The company also extended MBT testing through Conformiq to include connecting with, and verifying, the correctness of the data in the database within this application.

Figure 4: Agile sprint schedule using MBT

Conformiq 360° Test Automation

With the **Conformiq 360° Test Automation** approach, which includes MBT, the user models the expected system operation based on the requirements, and in terms of system flow and business logic for each action. Since the Conformiq solution is based on **functional system models**, two models were created: a UI model and a back-end model, and the interfaces were added to these models. This way, Conformiq generated **silo tests** and also combined the two individual models together, testing the entire system to capture **complex end-to-end tests** with **data variations**.

Because modeling was done during development, it enforced a thorough understanding of ”what was to be developed.” This visualization uncovered design issues before test generation and execution occurred. Conformiq’s thorough testing process uncovered issues that, without this approach, would have gone unnoticed until much later in the development cycle.

With Conformiq, test case generation is fully automatic. It is important to understand the need to model the expected operation of the application, because the tool will generate and optimize for the minimum number of necessary test cases, as well as generating the expected test execution results (the test oracle.)
Because the actual data was company confidential, the modelers included variables and placeholders in the model, allowing the company to insert the actual data during test execution. This made the model more reusable than it would have been with just predefining the data or hardcoding it into the model.

When the design was changed or additional features were added, the model (“Test IP”) was changed to reflect those changes. Conformiq automatically generated the new test cases, marked previously current test cases that were no longer valid, and indicated which test cases remained valid but were unchanged. Therefore, tests that were no longer useful were never included in the regression suite, which reduced the size of the suite by up to 40%, for large applications with numerous changes.

The documentation that was generated described the test cases. It was automatically created with names and step-by-step test case descriptions, test coverage (including a report of what was not covered), graphical test steps, a test case to requirements traceability matrix, and graphical traceability of what every test case covered, for the user created model. It is critical for users to know why every test case was generated so that they can trust that the coverage and the model are both good and will help debug errors.

Quality Center (QC) and Quick Test Pro (QTP) from Hewlett Packard, now combined into HP/UFT, were used for UI testing. Those scripts were generated as VB (Visual Basic) scripts for direct automated execution in HP/QTP, either through batch or serial execution. For back-end code testing, the test scripts were generated in Java for execution in their custom harness; a combined test execution framework could have been used, if desired.

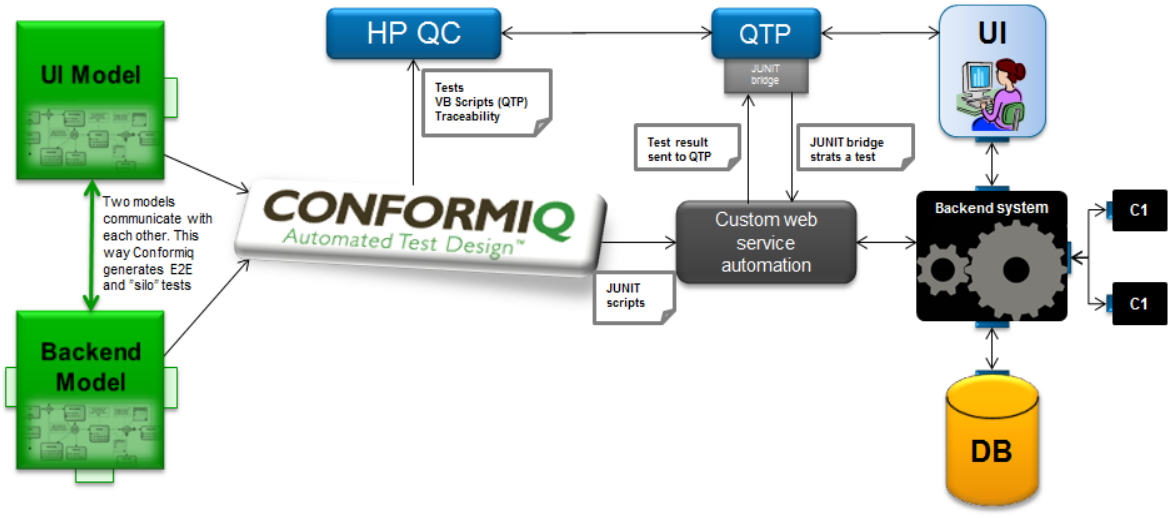

Figure 5: Graphical Overview Diagram of the testing process

CONFORMIQ

Project Results Using Conformiq 360° Test Automation

The average Conformiq test results achieved over the entire project are listed below.

Project Results	Using MBT
Average number of test cases generated/sprint	10 new + 16 modified
Percent test coverage per sprint	90%
Defects found per sprint	9+
Percent test case automation per sprint	80%
Requirements coverage	100%

Figure 6: Conformiq results compared with manual methods

The overall results achieved were:

- **With Conformiq 360° Test Automation, automated in-sprint progression testing was maintained throughout the project, while the project using manual test design did not.**
- **Testing coverage was better than that achieved by manual test design methods.**
- **The use of Conformiq, and the rigor and completeness this MBT approach delivers, even during the short sprint times, uncovered many issues that would have been unnoticed, possibly until deployment.**

The company previously had a separate but similar development project using manual functional test design methods. Within a few months after that project began, the test scripts and their weekly changes became too complex and difficult to create manually, let alone execute, within the short sprint times. Testing during this project fell well behind development and the company had to greatly reduce the scope and completeness of its functional testing during each sprint to only a few selected use cases.

In contrast, the Conformiq in-sprint "progression" testing based on MBT eliminated much of the need for system integration testing after the final code drop, since all functional testing for the application was completed during development. End-to-end system testing was still needed, but the models created during development were used to minimize this work. This is a good use of MBT since abstractions can make end-to-end modeling easier, capturing the full system operation as black-box test cases.

In summary, without the use of Conformiq 360° Test Automation, the project would have not been able to test at the speed of development. The company also would not have been able to test with the confidence gained from knowing what was covered and what was not covered, by the generated test cases for this financially critical application. Conformiq's unique 360° Test Automation technology enables the next generation of testing for complex testing environments, successfully transitioning customers from classic manual testing methodologies to next-generation test automation, for agile projects.

CONFORMIQ

Conformiq is transforming software testing with Conformiq 360° Test Automation™, providing the most sophisticated and comprehensive automated test design solution in the industry. The unique Conformiq 360° Test Automation technology enables the next generation of testing: transforming, streamlining and automating even the most complex system-level testing environments. Conformiq 360° Test Automation improves efficiency with a 40% faster test case development cycle; enables delivery of higher quality code with 50% more defects found; increases manageability with 50% better collaboration: and reduces costs with a 400% return on investment. Conformiq serves enterprise IT, communications and embedded software markets worldwide. Privately-held Conformiq is headquartered in San Jose, California, with a worldwide delivery and support organization including offices in Finland, Germany, Sweden, and India.

www.conformiq.com **sales@conformiq.com**

USA
4030 Moorpark Ave
San Jose, CA 95117
Tel: +1 408 898 2140
Fax: +1 408 725 8405

FINLAND
Westendintie 1
02160 Espoo
Tel: +358 10 286 6300
Fax: +358 10 286 6309

SWEDEN
Stureplan 4C
SE-11435 Stockholm
Tel: +46 852 500 222
Fax: +358 10 286 6309

GERMANY
Maximilianstrasse 35
80539 Munich
Tel: +49 89 89 659 275
Fax: +358 10 286 6309

INDIA
29 M.G. Road Ste 504
Bangalore 560 001
Tel: +91 80 4155 0994