Core Banking Enhanced Test Design Case Study

BACKGROUND

Oracle Financial Services Software Ltd. (OFSS) has enhanced its core banking testing services to take advantage of the latest in testing technology. OFSS leverages commercial Model Based Testing (MBT) software from Conformiq to deliver improved functional testing services its customers. Through MBT, Oracle brings technology to manual test design while improving automated test execution. Because of the many mandated changes in the banking industry, having the ability to conform sooner and deliver new customer banking products faster represents a key benefit to Oracle's customers.

OFSS wants to provide the best in breed testing services so it looks to advanced technology to incorporate into its services offerings to deliver ever better testing. This better testing through the MBT process includes improved and known test coverage, reduced defect slip through, test case traceability, full and always current test documentation, faster and sooner testing after software changes to shorten time to release, and the ability to provide more flexible testing services pricing.

SITUATION

As MBT and more specifically, Automated Test Design, has differences from the long accepted manual test design process, Oracle wanted to demonstrate to one of its large banking customers the benefits this technology can deliver when combined with Oracle's unique industry knowledge. A large US bank was selected to showcase the advantages of Oracle's Automated Test Design process. The bank's Exotic Options application was selected as it was mutually determined to be a representative example of the broader bank software as it had complex operations, many rules, single and double barriers, and user interface front office operation together with back office operation as a combined end to end system. Both the exercising and settlement operations were selected. The generated test cases were to be executed using the Rational Robot test framework although Oracle OATS, HP QTP, Selenium, or another framework could have been selected instead.

The issues the bank faced became the goals of the program. These included all the testing improvements previously listed, but the most key customer needs were improved testing coverage, i.e., finding more defects, and faster testing.

GOAL

The project goal was to demonstrate to the bank, using its own software, that this advanced test design process would ensure better software was delivered, test it faster, and allow Oracle to be more aggressive with the bank's on-going requests for year-over-year price reduction. The specific improvement goals were focused on Oracle's ability to:

- Showcase efficiency gain in end-to-end testing
- Showcase faster and easier design change impact

These two testing process speed-ups were the key differentiators for the bank and showcasing them allowed Oracle to differentiate its testing services over other vendors who were using traditional manual test design and execution.

SYSTEM DESCRIPTION

In the Exotic Options application, Barrier Options are Vanilla Put and Call Options with additional barriers. In case of a knock-out, the option expires worthless if the spot ever trades at or beyond the pre-specified barrier. In case of a knock-in option, the option is only activated if the spot ever trades at or beyond the pre-specified barrier. The barrier is valid at all times between inception of the trade and the maturity time of the option. Standard extensions of barrier options are double barrier options, where there is a barrier above and below the current spot. A double knock-in option only becomes a vanilla option if at least one of the two barriers is touched or crossed.

MBT PROCESS

Oracle used the MBT process for automating the design of functional test cases, executable scripts, and the associated reports starts with a model of the way the Exotic Option application is expected to work based on the system specifications. User stories could have been used instead and combined into a system model. This is important as this process will then include the interactions between the use cases, if any, where individual testing may miss defects. From the model, the delivery teams can visualize the system operation and quickly and early in the process see if there are gaps or misunderstandings in the overall operation. As this was a completed and released application, the developers were not involved and finding additional defects was considered unlikely.

Once the model was approved, test cases were automatically generated using one or more of many different test design heuristics, including pairwise, boundary value, transitions, paths or just requirements placed in the model by the SME modeler. Multiple graphical diagrams and reports showed the SME if the coverage is what was expected. If not, changes were made and the model was rerun.

Once the testing was approved, executable test scripts were generated for automated execution, including the correct expected results, so it was easy to determine if the System Under Test (SUT) i.e., the application, passes or not. The full SDLC and how MBT overlays to this process are shown in Figure 1.

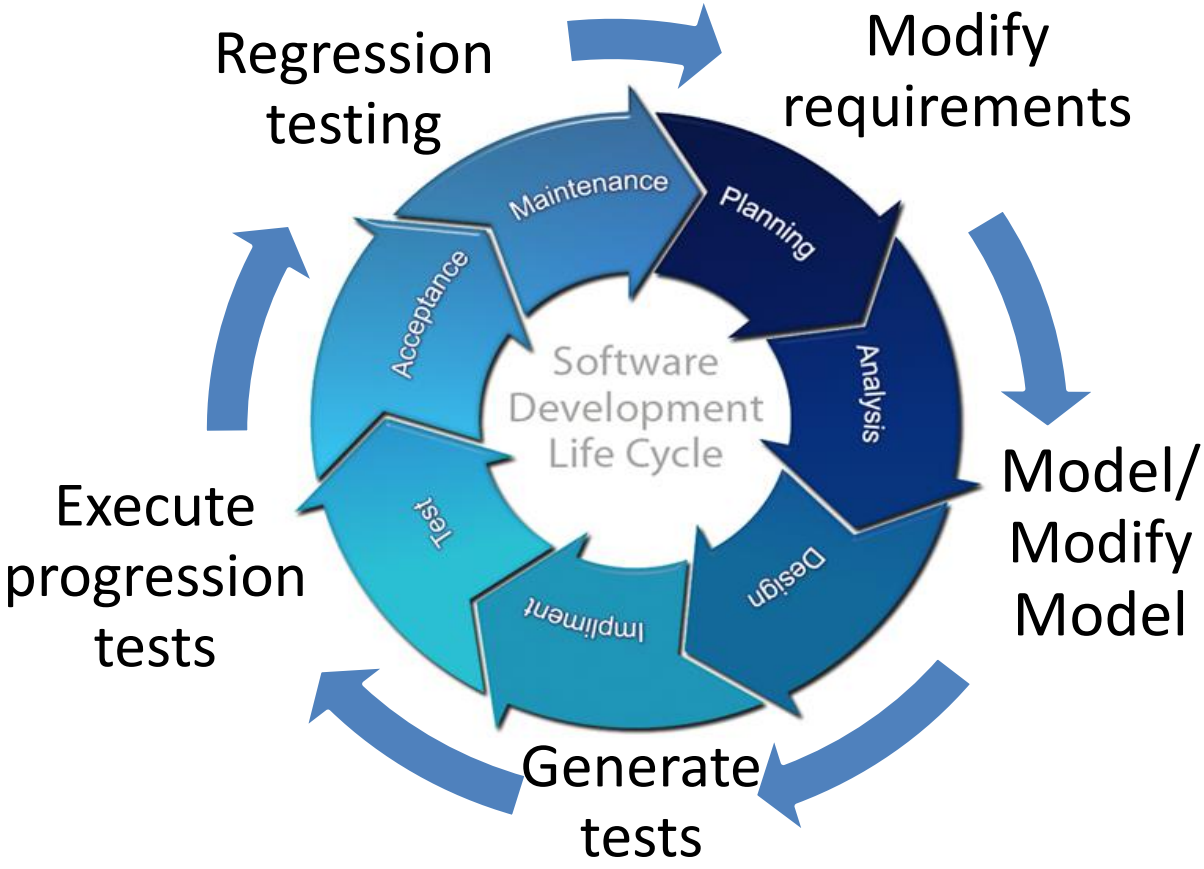

Figure 1 – Model Based Testing in SDLC process

The actual flow of Oracle's Automated Test Design process is captures in screen shots in Figure 2. It is a three step process, 1) Model, 2) Validate the model is good by reviewing the generated test cases and determine if the test coverage is appropriate for the desired coverage. If not, either the model is changed or different/additional test design heuristics are selected and used for the updated test cases. If they are as desired, then executable test scripts with the correct expected results or the manual test steps in a user determined format are "exported" into a file outside of Conformiq to be independently used in either automated or manual test execution, respectively. Scripts can be saved, but the key is to save the models because new scripts can be quickly generated whenever needed. This eliminates the need to manually manage scripts. The models also represent customer IP that can be saved and reused. This IP allows the customer to control its own testing knowledge and not give it to the testing vendor. Oracle promotes the ability for its customers to "own" their own testing assets.

Further, on every model change, Conformiq automatically notes which test cases are unchanged, which are no longer valid and can be dropped, and which are new. Thus the regression suite is always optimized for every design change throughout the Software Development Life Cycle (SDLC).

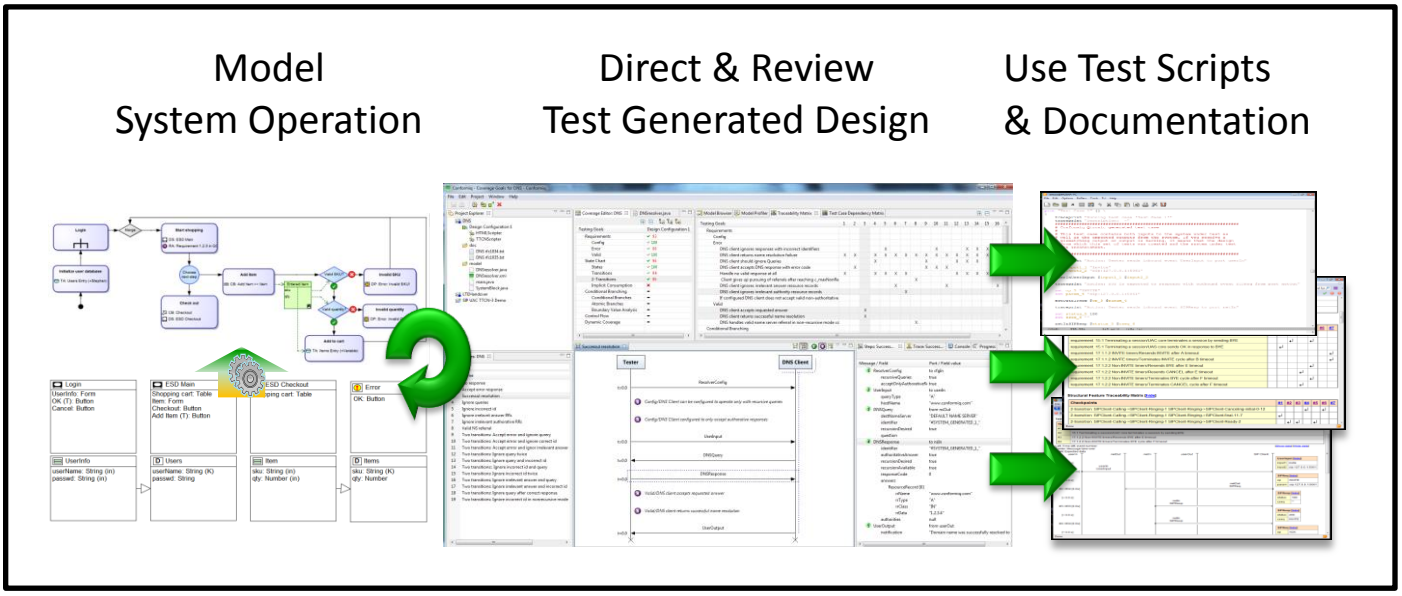

Figure 2 – Automated Test Design is a three step process

MBT PROJECT ACTIVITIES

The scope of the Automated Test Design work on this project included the following steps:

- Create the model for the Exotic Option application
  - Single Barrier
  - Double Barrier
- Create the backend scripter in SQA form for Robot execution
- Execute at least one end-to-end test case (Login scenario)
- Autogenerate test documentation (in HTML, Word, and Excel)

Note that the initial plan was to execute all the generated test scripts. However, the Robot test function libraries were not available during the period allotted to this program, so only one test case for the Login scenario was executed as an example. The customer understood the situation and recognized that if one test case script could be automatically generated and automatically executed, then all generated test cases could be executed.

The Exotic Options model covered the business flow of the different options creation process:

- Log-Out
- Selecting the option creation link from the left navigation menu

- Entering/Selecting the different fields on FX deal capture screen
- Entering/Selecting the different fields on FX-OTC tab screen
- Verifying the delta values

The model that was developed had the following scope and details:

- The model had one state machine for the UI behavior
- Different views of the UI were mapped into the states of the state machine
- Transitions define the user interactions that allow moving from states == UI views from one to another
- Code on the transitions of the model call functions in the textual part of the model
- These functions displayed what was shown on the UI as a response to user inputs
- Re-Usable component for the Login functionality
- Covered valid and invalid login scenarios

Note that since this case study was completed, Oracle is using Conformiq's latest modeling tool that requires no programming. This new method has demonstrated even greater test design efficiency.

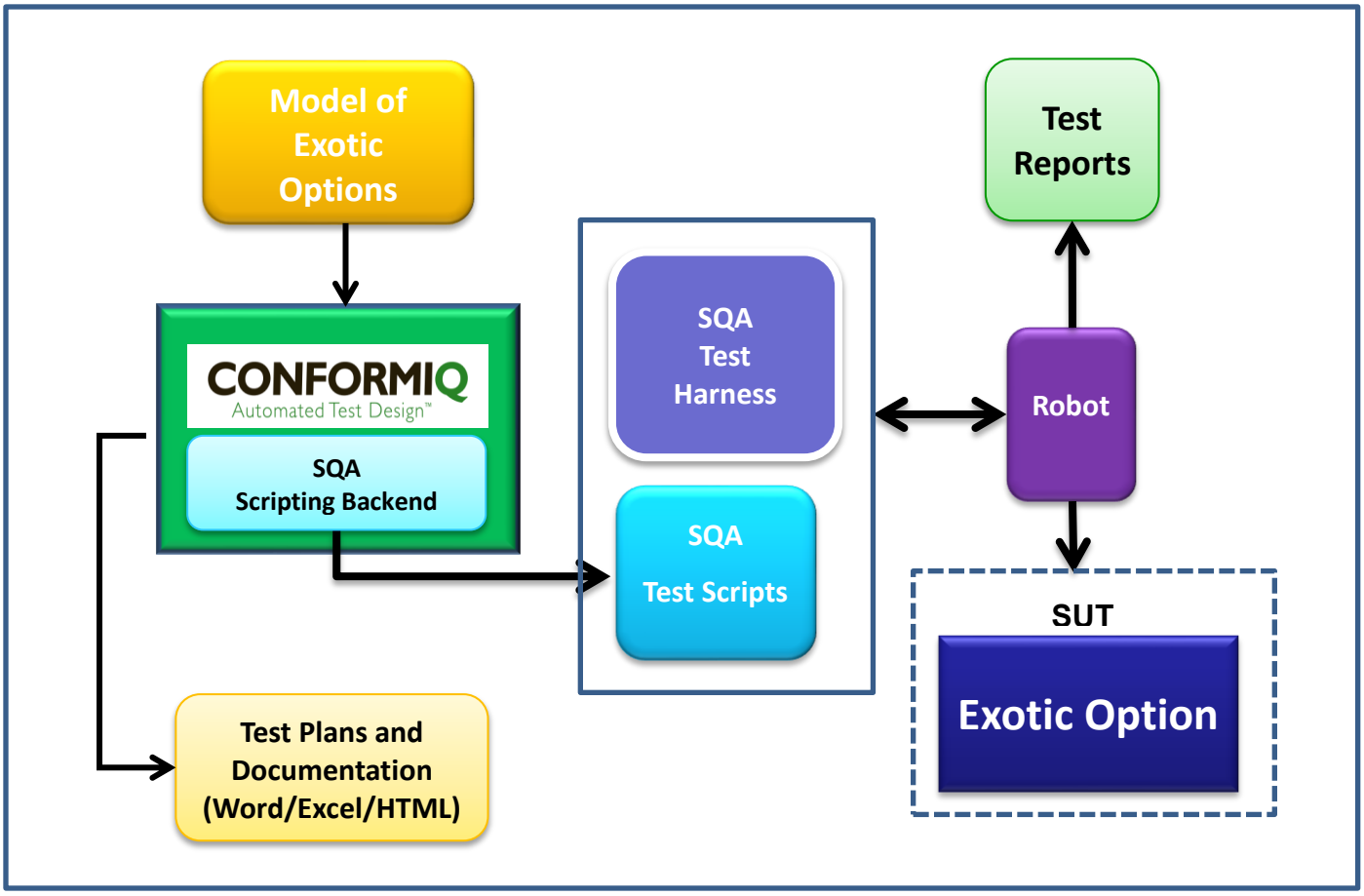

Figure 3 – Test System Architecture

Figure 3 shows the overall Exotic Options testing system architecture. The model flows to test generation. Once the test scripts were generated, there is no need to manually write scripts so there is no test case backlog. Automated test execution or manual execution, if desired, can begin. This allows progression testing during the development process by SME modelers in the development team.

OBSERVATIONS

In addition to the targeted results, several learnings valid on this program as well as on future programs were observed. These included:

- Test case dependencies and requirement traceability were represented well by the functional models
- Users would essentially need to have some amount of coding knowledge/experience (JAVA/QML) [After this program finished, Conformiq released a new modeling tool, Conformiq Creator™, which requires no programming and is targeted for use by SMEs.]
- Users with technical/automation experience may have a shorter learning curve.
- Efforts would be required to map the scripts generated by the tool with the project libraries before starting the actual test execution. This step needs to be planned so staffing occurs during development rather than waiting until after code drop.
- For Greenfield initiatives, the approach and tool would contribute to delivery and QA efficiency by virtue of visualizing the flow of the application in the model.

EFFORT

The generated test suite consisted of Login /Logout, verification of the filed validations and option creations. The detailed time for each effort is specified in the table below.

MBT Activities	Time Efforts
Modeling	40 Hours
Backend/Harness work	5 Hours
Backend Scripter Creation for Robot (* This is one time effort)	12 Hours (***Note:** Backend Scripter creation in SQA for Rational Robot is a **one time activity**, which includes all learning of a new tool / language that the modelling SME's were not familiar with initially.**)**
Test execution	1 Minute/Test Case
Project management	74 Hours

Even though this program had a modest scope, it demonstrated to the bank that the OFSS MBT approach delivers key benefits of improved productivity and shortening of the testing process. Because the Exotic Options software was mature, new additional defects were not found but the development team did see that they would be able to improve their work in the future which should deliver additional quality and time improvements.

The Conformiq Automated Test Design software was determined to be scalable for full scope application testing by virtue of its ability to automatically split the model and deterministically use multiple computer cores, thus reducing the test generation time roughly linearly with the core numbers. This provided additional confidence in the Oracle approach.

RESULTS

This MBT program delivered the following results against already good customer efficiency:

- Generated test cases: 32 test cases contained 597 test steps for automated execution
- Number of automation Scripts: 4 for manual versus 32 independent sub flows with Conformiq
- Total requirements covered: 24
- **Test design effort was 1 hour using Conformiq versus 72 hours manually**
- **Script readiness time was reduced by 8 hours compared to the previous manual process**
- **Script generation time was 1 hour compared to earlier manual process which took 240 hours**

OFSS proved to a skeptical but interested customer that the impact of using Automated Test Design in a Model Based Testing process delivers a tremendous benefit which spreads across the entire product delivery team. Remodeling with future design changes will be faster than the initial modeling effort and will provide even greater customer testing efficiency improvements and faster go to market times.

SUMMARY

Oracle Financial Services Software can effectively use Model Based Testing to deliver:

1. **Reduced testing time** ✓
2. **Improved quality** ✓
3. **Model as a document** ✓
4. **Reusable model architecture** ✓
5. **Automated test execution** ✓

Oracle Financial Services Software Ltd. is a subsidiary of Oracle Corporation. OFSS is an IT solution provider to the banking industry, having more than 900 customers in over 145 countries. OFSS delivers its industry leading FLEXCUBE software and testing services through offices worldwide.

Oracle Financial Services Software Limited
Oracle Park off Western Express Highway, Goregaon East, Mumbai, Maharashtra 400 063 India
www.oracle.com/financialservices