Symphony helped a fast-growing Sweden-based ERP vendor to reduce testing cycle significantly and improve test coverage to uncover inherent defects in its ERP application by custom developing a smart, data-driven UI automation library.

The Client

The client is one of the leading, Sweden-based ERP vendors focused on providing small and midsize companies with efficient ERP solutions to develop and support their business processes. For more than 15 years, they have been providing ERP systems to manufacturing, trade, service-oriented companies, and have 4000+ clients across 40 countries.

The Challenges

The client profile for the ERP system includes small to midsize and large companies with lots of diversified features, customizations and screens. Every new build release had to undergo tedious manual testing cycle against the growing demand for defect-free, quick releases to market.

The customer sought to cut down the product testing cycle and improve test coverage considering the extensive customizations of the product. Hence, they decided to outsource software testing to an experienced product-engineering partner to help them in their QA initiative.

The Solution

Symphony's solution to the client included designing a layered architecture to build test infrastructure for functional test automation. The main aim was to build functional test cases quickly, allow import of test case inputs and cover complex test scenarios with ease. Integration with Cruise Control enabled fully automated functional test setup.

Using White Library (a wrapper over Microsoft's UI Automation library and Windows messages) as the UI Automation platform, Symphony's Test engineers developed an infrastructure library to

Software Testing

Functional Test Automation

Platform

Microsoft

Application

ERP

Engagement Key Points

- Project duration – 3 Months
- Ramp Up Time from SOW sign off – 15 Days
- Team composition – Project Manager, Test automation developer
- Peak Team Size – 5
- Engagement model: Time and Material

Technology and Tools Used

- .NET
- NUnit
- Project White
- Microsoft's UI Automation library and Windows messages

Major Features

- A maintainable and extendable UI Automation Library
- Infrastructure library for automating the Rich Client UI
- Easy simulation of simple to complex test scenarios
- Extensive logging and reporting options -

automate ERP Rich Client. This supported rich client applications, such as Win32, WinForm, WPF and SWT (java) and supported some custom Win32 Controls like Grid Views.

The infrastructure library consisting of various layers (such as UI library, Utility, Test Cases) allowed writing easily extendable and maintainable test cases for new entities. UI Library represented various windows, which construct the Rich Client's UI providing the way to access each control on any program window like text boxes, combo boxes, check boxes, buttons grids tab controls etc. The Utility Layer has two sub layers named as Data Accessor and UI Helper. Data Accessor helps to import test cases - gets the data from the MS Excel files - to create a particular entity using the entity model. The UI Helper uses this entity to map entity data with the UI controls on various program windows using the services of UI Library.

Every functional test case helps to simulate a particular user flow. The UI Library and Utility layers provide great level of abstraction thereby making it very easy to simulate any simple or complex test scenario. AutoCode Generator using CodeGen tool provided all infrastructure code that is required to extend the library for other program windows and also provide basic test cases (create, update, delete, search) for the same program window which are ready to run.

The UI Automation library is completely data driven. Test cases use Data Access layer to get the list of entities in the corresponding MS Excel file. Then each entity in the list uses the services of UI Helper to map this data to controls on the Company control window. Each test case integrates with NUnit framework and is supported by Cruise Control.

Extensive logging and reporting was available through a portal. This allowed in-depth test result analysis from multiple automation machines to provide greater insight to QA team and faster action by developers.

The Benefits

Symphony's Software Testing and Test Automation skills helped bring in the following benefits for the client:

- Automation Test suite helped speed daily test cycle time significantly
- Automated functional test setup helped to execute functional test case on-demand or after each new build thereby reducing testing cycle
- Improved width and depth of test coverage (by adopting the layered test infrastructure) helped uncover inherent defects in the applications