



Introduction

This project was conducted with a major Australian manufacturing organisation comprised at the time of 5 relatively independent divisions.

The project and involved the implementation of SAP 4 and APO Legacy Regression Testing.

TestPro's Role:

- Build an automated suite of tests to enable regression testing of the legacy SAP implementations.
- Assist in volume testing of a division's implementation of APO.

Project Outcome:

We significantly reduced the amount of time it took the client to regression test 70+ transactions (reduced from weeks to hours)

We simulated the load in a number of SAP environments, thus providing a clear view of performance bottlenecks over a 24 hour processing period.

Project Details:

The approach had 3 major streams of work:

- Functional Regression Testing
- HP Mercury toolset implementation
- Performance Testing

The focus was initially on Functional testing, in particular the regression testing of the legacy environment.

Performance testing followed the Functional testing since all functions should pass before being performance or stress tested in any way.

Regression Testing Approach Used:

There are several essential components to Functional testing:

- Business requirements
- Documentation of business process
- Generation of Test Cases
- Execution of Test Cases (either individually or as scenarios)
- Analyzing/Reporting of Execution

End to end testing relies on the generation of the test case from a documented business process and the involvement of the BPO or SME during the translation of the business process to a logical series of SAP transactions and the test case.

In this way ownership of the business process is maintained with the business. Execution and analysis of the test cases are controlled

Existing Client Test Processes

The client IS team had previously determined that there was little current business process documentation. Most of the processes resided with individuals and their own knowledge of their jobs.

The concentrations of overall transactional scenarios resided with client SAP functional consultants.

Generally, internal ownership of testing was held by the business. It had been sufficient for minor, ongoing changes to SAP, but it did not follow a number of standard testing processes and hence would not conform to the general testing rule of repeatability.

In the case of the significant SAP upgrade that TestPro was involved with, the implementation risk was higher due to code changes to transactions used across the divisions.



It was therefore decided a more formal testing approach should be adopted, but with a degree of flexibility due to the lack of formal documentation of business processes and limited time.

Functional Testing

With the client objectives and restrictions in mind the overall approach of the Functional testing was to:

1. Determine and limit what the critical transactions are to be tested
2. Determine what Scenarios (business Processes) incorporate the transactions to be tested
3. Determine possible corner case and negative testing to be performed manually
4. Document the Scenarios to be executed
5. Automate and parameterize scenarios
6. Execute on environment representing “current environment”
7. Analyze for issues and risk exposure
8. Re script to address for changes due code deployment
9. Execute on system representing “environment prior to activation of code”
10. Analyze for issues and risk exposure
11. Execute on environment representing “post activation of code”
12. Analyze for issues and risk exposure Performance Testing

The objectives of the testing included:

- Identifying the order and frequency of activities performed in a normal day during peak periods
- Identifying the transactions that might break the system, e.g. reports, batch processes, month end, etc.

- Identifying user load. In order to get a realistic load, user groups were identified and allocated to Scenarios to simulate their normal working patterns.
- Groups were given different loadings to represent the balance of the different user activities.

The acceptance criteria for the implementation took into account, response times, error rates, user loads etc derived from existing SAP statistics.

The test scenarios included:

- Best performance of BAU transactions (workload < 10%),
- Best performance of APO transactions (workload < 10%),
- BAU performance at 50% and 100% workloads,
- APO performance at 50% and 100% workloads,
- Mixed BAU and APO workloads at 50% and 100% workloads,
- Mixed BAU and APO workloads at 50% and 100% workloads with additional batch jobs executing.

Performance Test Scenarios

1. Business As Usual (BAU) background load at 50% load and 100% expected load.
2. Additional processing from the upgrade.
3. BAU and additional processing from the upgrade combined.

Testing used a method of progressively adding load in steps in order to validate and measure each step. This process also allowed the various interested parties to make changes to improve performance or rectify areas of major concern.



Test Results

From testing we found a number of areas where changes were made, with varying degrees of impact on performance.

- It was found that parallel processing in the APO sales to requisition process was unable to be supported and changes were reverted to single processing.
- Changes were made to Dialog processes to remove a bottleneck in BAU causing performance degradation (50% performance increase).
- Two servers had a different parameter set which caused the requisition process to hang.

Testing ran a series of scenarios, in particular in BAU, testing matched the current production load with the simulated load. This provided the required impact for measurement of the impact of the upgrade.

General BAU performance was acceptable, being under 2 seconds for standard SAP screen changes.