Test Optimization using Combinatorial Test Design

Real-world experience in deployment of combinatorial testing at scale

Saritha Route
Global Business Services, IBM
IBM India Pvt. Ltd, Bangalore, India

Abstract:

- Clients today want more for less and the IBM test mantra of "Test Less, Test Right" helps address this ask by placing Combinatorial Test Design (CTD) at the heart of the solution.
- This document presents two case studies of CTD implementation in client engagements and focuses on the approach, process and challenges addressed to scale up the implementation and make CTD a mainstream activity. The IBM Focus tool was used in both cases to implement Combinatorial Test Design for optimization of tests and for reducing test effort while increasing test coverage.

Introduction

"Test Less and Test Right" has been a mantra within the IBM test practice for a few years now. The heart of this formula has been implementation of techniques and practices that help optimize test effort. We have been propagating test design optimization as a core transformation lever for our clients.

Test design is one of the most critical phases in the test life cycle as it determines the quantum of testing that will be performed during execution, the quantum and type of data needed for testing. It also determines what tests will finally be automated for future execution. Combinatorial Test as a design technique becomes invaluable, as the outcome is an optimized test suite - which translates to the minimum number of tests for a requisite test coverage.

The logic of the solution appeals to clients and teams alike, but adoption usually requires a larger transformative approach; in an environment where testing prowess is typically measured by the quantum of tests being executed. This approach is not only new but also considered disruptive. This paper shares the results and journey from two examples of large scale CTD implementation in Test Design. The implementations are across two insurance sector clients in North America.

Practical lessons from large scale implementations of CTD in test design

Our first example is of a large scale implementation of CTD that commenced in 2013-2014 for a large insurance house and it required proving the concept in a commercial context. In the second, more recent large scale implementation that commenced in 2016; we established the process upfront to embed CTD as the mainstream test design approach. The first client required a greater focus on test cost reduction and hence effort reduction; while in the latter, the impetus was test coverage with a lowered test execution effort.

A. CTD to reduce test effort and cost
In the first client example, the test program spanned 19 lines of business and the team comprised over 500 testers. The aim of implementing CTD in this program was to reduce test cases and thereby reduce test effort and cost. It was, therefore, critical to evaluate the existing regression tests beds and build a repository of optimized test suites that could be used repeatedly with a smaller test team. We needed to identify the right set of applications and the right tests upfront for optimization to ensure early benefits, by assessing:

- Size of the existing regression suite
- Frequency of usage of the tests
- Quantum of changes expected in per software release
- Types/ nature of changes
- Nature of the application testing (front end, user interface functional, end to end interactions, tests high on variability and interactions etc.)
- The number of releases in the coming year(s) in which the optimized test suite would be used (for ensuing greater returns on design investment)
- Early return on investment of the CTD modeling effort

In the first year alone we developed over 75 CTD models covering existing regression test beds of different applications. Across these models we analyzed 4516 existing test cases; deconstructed them into comprehensive trace files that listed all paths and all variables impacted. The base test case count increased to 4859 test cases; to make up for omissions, obliteration and for assumptions built into the original tests. Usually these aspects get addressed during test execution leading to increased execution time, as it means that design decisions are deferred to execution. We then leveraged the IBM Focus tool to read the parameters and points of variation across tests and provide the optimal combinations required and to down-select tests that would provide the same coverage.

IBM Functional Coverage Unified Solution (IBM FOCUS) is an advanced test planning tool. IBM FOCUS uses Combinatorial Test Design (CTD) to generate an efficient test plan that provides consistent coverage across the test space at a known depth, while significantly reducing the required resources. IBM FOCUS is independent of the application's domain, and can be applied at different levels of testing. IBM FOCUS can also read existing tests, analyze their functional coverage, select a subset of the tests that maintains the same coverage, and generate new tests to close the coverage gaps.

The modelling process yielded an average reduction of 32% on the absolute test count and a corresponding 27% reduction of the original base test pack. As the average test execution effort was 3 hours per test case, the overall effort reduction translated to nearly 4000 hours or two person years' effort savings for one execution cycle. Being regression tests that are run multiple times a year based on the program-based release schedules, the cumulative savings delivered was manifold.

The initial models developed by testers certified in the CTD techniques and on the IBM Focus tool helped establish the concept and provide an early advantage. However, the design effort increased as there was a dependence on application subject matter experts (SMEs) and on testers well-versed in the testing process, functionality and the test cases. Further the test optimization was carried out as a process parallel to the core test service, not always aligned to the release cycles. The inclusion of the optimized test suites in release testing was also delayed as the adoption of the reduced test pack was not instant as expected. Clients who were familiar with the old test beds needed convincing that a smaller set of tests would provide the same benefits as the pre-existing tests and that fears of defect leakage due to lack of coverage was unfounded. We showcased the benefits of CTD – with pre and post-CTD test cases analysis to show that the redundant tests were not essential and to prove that the down-selected tests would provide the same coverage. The coverage reports from the IBM Focus tool helped in visually demonstrating the coverage impact.

Another deployment challenge was that the optimized test beds were not being used in the next regression cycle as a matter of course. Since the models were hosted outside the test management tool, the optimized test cases were not updated into application regression pack. A process was established to update the existing tests in the repository mark them as retained, obsolete, modified or new; and testers were trained to propagate tests that were optimized.

The results of these actions along-with an early CTD model developed for the business users to optimize their User Acceptance Testing and data set-up for a large sweeping business change helped bring CTD into main stream test design. We trained over 100 testers in the team on CTD techniques and the IBM Focus tool. We also modified the approach to include CTD in functional test design to create the optimized set of tests upfront ensuring that optimized tests were propagated into down-stream regression testing.

B. CTD to replace risk based testing needs
In the recent implementation of CTD for another Insurance client, we leveraged the processes established in the earlier example and created a framework of transformation upfront to include CTD in the functional test design process to focus on test coverage with optimization. We conducted a pilot to establish the benefits and seek early buy-in from stakeholders and mentored the team to be self-sufficient in CTD modelling. The test process was modified so all new program requirements where converted to test cases using CTD by default and core design practice.

A key result was that the team was able to create more test cases per day than planned, as the throughput improved due to the CTD modelling technique. The planned test design productivity was 8 test cases per day per tester through the program. With CTD the throughput increased from 4 to 6 test cases per day in the inception phase to over 12 test cases per day in steady state; thereby delivering increased productivity in test design and enabling effort savings in test design and in test execution. With just 8 weeks of learning, the team was able to hit steady state of test design and increased design throughput by 9%. They were able to ensure 100% coverage with 18% less test cases than was estimated.

The client was initially keen on a risk based testing approach as the execution time windows were short. The client required assurance that key processes would be covered. With CTD as a core practice, we were able to ensure that not only was a reduced test pack developed, it was developed faster and it covered all flows and required less effort and time for execution. We also eliminated the effort of determining the right tests for risk based coverage and removed dependence on client SMEs for risk validation and redirected that effort to executing all test cases within the same execution window. The coverage reports from the Focus tool were used to show the client that the right business flows were covered, and that they were covered by 100%.

Conclusion

Implementing CT design techniques in our test process has provided the dual advantage of an optimized pack with assured coverage. Further large scale deployments are possible and yield strong results. Approaches might vary based on the size of the client engagement, scope of work and nature of engagement. Overall acceptance of the technique in mainstream test design has increased over the years and can be used for test reduction and optimal test generation. The key to ensuring success at scale is having an approach that is socialized early with all stakeholders and enabling the team early on the technique.

References

[1] Krishnan, R and Krishna, S Murali and Nandhan, P Siva, "Combinatorial testing: learnings from our experience," ACM SIGSOFT Software Engineering Notes, vol. 32, no. 3, pp. 100-108, 2007.
[2] Rogstad, Erik and Briand, Lionel, "Cost effective strategies for the regression testing of database applications: Case study and lessons learned," Journal of Systems and Software, vol. 113, pp 257-274, 2016

IBM Corporation
Global Business Services
Route 100
Somers, NY 10589

Produced in the United States of America
February 2018

Please Recycle

20013420-USEN-00