

The Hidden Benefits of Automated Testing

Ryan J. Melton
Ball Aerospace & Technologies Corp.
Boulder, CO

ABSTRACT

Automated testing provides many benefits including eliminating user input errors, faster execution times, and reduced user oversight during execution. In addition, many other tangible and intangible benefits result from avoiding manual interaction. The added cost and effort involved in running manual tests ensures that tests will only be run the minimum amount of times.. Automated tests can be run repeatedly and can loop continuously, which can wring out timing issues and also lead to better products because of the increased probability of detecting a failure with the increased number of executions. Increased testing time also creates a treasure trove of logged data that can be used to provide very detailed characterization of system behavior. With configurable automated test procedures, regression tests can be executed without having to re-release the test documentation. This paper will explore these benefits provided by writing automated test procedures using the open source Ball Aerospace COSMOS testing system.

KEY WORDS: Test Automation, Benefits, Ball Aerospace COSMOS

INTRODUCTION

Automated testing has changed the landscape of both test quality and quantity within Ball Aerospace. This paper discusses the benefits and downsides that were predicted before moving to ubiquitous use of automated testing as well as several hidden benefits that have been discovered over time.

BENEFITS OF AUTOMATED TESTING

The following benefits of automated testing are often taken as givens:

Automated tests take less time to execute than manual tests – Manual tests can be very slow. Operators have to read the procedure, understand what it says, perform a manual action such as typing in a command or pressing a button, and record the results. All of these delays can be removed by automated testing allowing for tests to complete in an optimal timespan.

Automated tests are less error prone than manual tests – Manual typing and pressing buttons leaves a lot of room for typos, and simple human error. Automated testing removes the possibility of ever typing the wrong thing.

Automated tests can be written to run without any user interaction (or even humans in the room) – Human interaction greatly limits when tests can run and how frequently they can be run. Operators have to be sufficiently trained and enough operators have to be available to maintain 24 hour operations when needed. Automated tests can be run anytime with limited to no human

oversight greatly reducing costs and staffing requirements.

Automated testing scales – Manual testing may work out ok when you are producing one widget at a time, but what happens when you get an order for one hundred? Automated testing can allow for cost effective scaling of your production line where one operator can oversee numerous tests simultaneously.

DOWNSIDES OF AUTOMATED TESTING

With these benefits also come a few fairly given downsides:

Automated tests require automatable hardware – To fully automate a test everything has to be controllable by software. Engineers need to make sure that ground support equipment (GSE) like power supplies can be controlled remotely to support automating tests. This GSE is typically more expensive than manually controlled equivalent hardware.

Automated tests typically take longer to create – Automated tests have to be written in a syntactically correct manner, tested, and debugged before use. Word documents listing manual steps are much more forgiving and often take less time to create.

The skillset required to create automated tests is often different than that to create an equivalent manual test – Automated tests are effectively software. A good test framework can remove a lot of the minutia of software development, but an automated test procedure writer still needs to be able to think linearly, handle basic if/then/else type logic, and write syntactically correct code.

Based on the above, the upfront cost of automated testing is often higher than creating a manual test – Due to the longer development time, and increased skill set demanded, automated tests tend to be more expensive to create.

HIDDEN BENEFITS OF AUTOMATED TESTING

In addition to the given upsides and downsides there are more subtle opportunities of using automated testing that can reap real benefits for your organization.

Automated tests can bring out subtle defects – One of the key benefits of automated testing is that there is little incremental cost to running the tests repeatedly. There are many classes of failures that only occur at a low frequency, sometimes less than 1% of the time. If your test procedure is only executed once, you'll never bring out these types of issues in your system. Automated testing allows for repeated and often overnight testing of your procedures and systems that can bring out the subtle defects in both the system under test and the procedure itself.

Automated tests can very precisely show where a test can be made more efficient – Good automated test frameworks timestamp everything. While it's useful knowing how long an overall test procedure takes to run, it can be even more valuable seeing which parts of the test procedure

take the longest to complete. This can provide a laser focus as to where the test procedure can be optimized or even what aspects of the system under test could benefit from performance improvements.

Automated tests can create a treasure trove of system characterization data – An enormous amount of telemetry data is generally created with each execution of a test procedure. The ease of running automated tests can produce a lot of extra data available to characterize the performance of your system. What is the average temperature of the focal plane, how much current does the system normally draw when moving the filter wheel? These kinds of questions can be answered with increased accuracy given an increased amount of data.

Automated tests can be run in parallel – While a human operator can only do one thing at a time, computers do multiple things at once. With well architected automated tests that verify different subsystems in isolation, multiple subsystems can potentially be tested at once. This is more difficult than it seems, but is a real possibility when doing automated testing that is not available with manual testing. As mentioned above, automated testing also works very well in a scaling application where the same test is simultaneously run across multiple units under test.

Automated tests can create automated test reports and post-processing test products – Automated tests can automate more than the test itself. They can also automate things that are typically done manually after the test is complete. In this case creating an automated test report that indicates the PASS/FAIL result of each test case is trivial for a good automated test framework. Additionally, the automated test can start any post-processing necessary after data has been recorded.

Automated tests can allow for regression tests without going through any release process – A configurable automated test system can allow the user to select and run a subset of a given test procedure without having to release any new documentation or to specifically release a stripped down version of the original test procedure. This can allow for quick regression tests without slow review cycles, therefore saving both time and money. For these regressions tests, the automated test report can be used to document exactly what was run and provide a record of the test.

USING BALL AEROSPACE COSMOS TO REALIZE THESE BENEFITS

Ball Aerospace COSMOS is a freely available open source testing system that provides a suite of tools that allow for the efficient creation of automated tests. In particular, a tool called Test Runner can enable a lot of the benefits of automated testing with very little effort from test implementers.

Test Runner automatically creates a test report at the end of each execution. This report clearly lists the pass/fail status of each test case. The automated test report also lists the start time and end time of each test case making it easy to determine which sections of the test procedure would benefit most from optimization.

Test Runner provides a test selection dialog that allows for executing any arbitrary set of test

cases. This feature makes it easy to select and run just the relevant tests necessary to regression test a portion of your overall system.

Test Runner provides a loop testing feature that allows for the repeated execution of an entire test procedure all the way down to a specific test case. It can be configured to stop if anything fails (recommended if it is necessary to preserve a failure's configuration) or to continue and simply log the failure (useful for discovering as many possible failures as possible within a given time frame).

In addition to the functionality provided by Test Runner, the rest of the COSMOS system also provides a wealth of tools for post-test analysis to help with maximizing the value of the data collected during any test and for troubleshooting anomalies.

SUMMARY

When viewed together, the combination of the clear and the hidden benefits of automated testing outweigh the negatives. The benefits discussed in this paper lead to better products and better tests. Making use of Ball Aerospace's COSMOS test system can allow anyone testing embedded systems to quickly realize these benefits.

REFERENCES

Zallar, Kerry, “Practical Experience in Automated Testing”, Retrieved from <http://www.methodsandtools.com/archive/archive.php?id=33>, August 14, 2015

Bach, James, “Test Automation Snake Oil”, Retrieved from http://www.satisfice.com/articles/test_automation_snake_oil.pdf, August 14, 2015

BIOGRAPHIES

Ryan Melton has been working at Ball Aerospace in Boulder Colorado and helping to develop, integrate, and test aerospace products for the past 14 years. Notable programs on which Ryan has worked include Kepler, GPM, and CALIPSO. Ryan is very active in the open source community with the recent release of Ball Aerospace COSMOS as well as for many years working with the Ruby bindings to the Qt GUI framework, qtbindings. He holds a Bachelor’s of Science in Computer Engineering from Purdue University and an MBA from Regis University. Please address any questions on this paper to rmelton@ball.com.