Written by:
Paul Holland

ATM Switched Services Verification
Alcatel Canada Inc
600 March Road,
Kanata, ON, K2K 2E6

Phone: (613) 784-6858

Having been asked to write an experience report paper on Realism in Testing, I was trying to decide which recent verification situation I would document. I gathered together a number of senior verification people, and we discussed potential topics. Despite having many good ideas, none of them seemed to fit the requirements well enough, or I was not adequately familiar with the details, to create a good paper. As a group we decided that it would be better to highlight the similarities of all of the projects that we were discussing. So we came up with the topic for this paper: "Common Themes in Realistic Testing".

Common Themes in Realistic Testing

There are many types of testing that are used when testing any product. There is unit testing, feature testing, integration testing, and system testing to name a few. As you progress through these phases of testing it becomes more important to perform your verification in a realistic manner. That is testing in the way that is as close as possible to the manner in which the product will be used by the end users. To effectively perform realistic testing, it is important to consider many different factors. Some of the factors involve decisions within our control like how much investment to make, while others are more out of our control like is there test equipment available that can help us.

To perform verification in a realistic context, here are a number of factors that must be considered:

- High capital costs vs. perceived benefit;
- Limitation of test equipment;
- Customer profiling;
- Randomness vs. repeatability and defect replication; and,
- Scheduling and sharing of resources.

These factors can potentially compromise the goals and outcome of the verification process; however, with an effective use of teamwork and innovation, it is possible to succeed in the overall goal of verifying the system in a realistic manner.

High Capital Costs

To be able to perform realistic testing in many environments it is necessary to have not only a large amount of equipment under test, but also test equipment, a large amount of lab space, and all of the power and cooling to properly operate this required gear. It is necessary for management to decide how much money to spend on realistic test beds and balance that expenditure against the perceived benefit. In some industries a test bed could cost millions of dollars, but there is no doubt that the expense is required (a tilt table or an engine dynamometer in vehicle testing for example). In telecommunications, it can be just as expensive, but much more difficult to rationalize the expense. In these instances it is necessary to use creative methods to substitute the purchase of a complete system.

> When it came time to develop a large switch for testing our largest switch, the cost of the system was difficult to justify, especially with the recent market conditions. At Alcatel, we have a 15 shelf system that has a high production cost. We had the obvious requirement of testing a full system, but the cost of a complete system was prohibitive. We explored many different options, including purchasing new equipment and/or borrowing from existing lab setups. These two options were dismissed due to their financial impact and their impact on the ability of others to perform necessary testing. Our final solution was to have our manufacturing facility agree to allow us to use equipment they had produced but not yet shipped. We created a mini-lab in their warehouse and setup our large system with a rolling inventory. In periods of high demand, like quarter-end, the setup had to be reduced in size and scope. For the most part this solution has allowed us to perform realistic full system testing with minimal capital cost expenditure on units under test. Essentially we replaced our fairly basic pre-shipment product verification with a much more in depth full product system test.

Another method of reducing capital costs comes from the effective sharing of resources between different groups. Historically, it was possible for individual groups to adequately equip themselves for all of their needs. In today's reality that is not fiscally possible. At the Ottawa campus of Alcatel, we have implemented a few cost saving measures that have helped reduce our overall capital expenditures without limiting our ability to test effectively:

- Lab interconnection is facilitated by patch panels. Test equipment is centralized. Networks of nodes and test equipment can be assembled as required;
- Optical switches are used in some lab areas to facilitate reconfiguration and sharing of test equipment;
- We negotiated a site license for test equipment application software;
- Efficient use of what we have: Periodic lab sweeps conducted to bring all gear up to the latest revision;
- Implemented a more efficient repair process (this one still needs a bit more work).

These methods are not perfect as we have to reduce the total amount of testing performed because we are sharing our equipment. We have also had to place additional demands on our personnel, asking them to work in shifts to make a better use of the equipment. In the long run, however, despite having very constraining limitations placed upon our capital budget, we have managed to continue to perform realistic testing by implementing innovative solutions to share our limited resources.

Limitation of Test Equipment

The limitation of test equipment is probably one of the most difficult problems to overcome to achieve realistic testing (equipment not available, too expensive, or inadequate).

Despite the increase in complexity of test equipment, Their development is always lagging behind the leading developers. This makes sense, as it is impractical for test equipment manufacturers to develop testers for every new, unproven technology that is attempted. The test equipment manufacturers must wait until it appears that a technology is likely to succeed or become popular before it is feasible to develop new gear. When test equipment is initially available it is very expensive, and we know there will be a limited useful life of the product. The solution that we have employed occasionally over the past few years is to create our own test environments and test gear when we have had no other options.

> When Alcatel (Newbridge at the time) first developed Frame Relay Svc ability, there were no testers available (and none on the immediate horizon). We decided that we had to develop our own home-grown testers. We took some of our T1/E1 interfaces on SPARC workstations, and modified their drivers to support FR Svcs. This was a large and expensive undertaking, but it was deemed necessary at the time. We still use this particular test environment over four years after its creation.

The same home-grown method of test equipment development may need to be employed if the cost of the 3rd party test equipment is prohibitive.

> We had a requirement for many generators of switched services calls. This caused us a problem because the testers were either too expensive or not powerful enough to be functional. We modified our Call Processing card to be able to generate calls at a rate that exceeded our ability to process calls. This allowed us to use a modified version of the SUT to test the customer load of the SUT. In other instances, to a limited degree, we also use our hardened products to test against newer products.

These practices do not lend themselves to realistic testing and must be used in moderation by the development teams. It would be very dangerous to use only home-grown test equipment exclusively, as they may hide some interaction problems or timing issues.

There is also a limitation associated with one piece of test equipment trying to simulate many inputs. Although they can generate complex data streams over multiple channels, all the data is still coming from one (or a small number of) CPU versus the real scenario of many CPUs each generating a completely independent data stream. This can allow timing issues to be missed in lab testing. The only way to work around this is to purchase more test equipment (expensive) or use home-grown (potentially dangerous) solutions.

Customer Profiling

To accurately test anything in a realistic manner, you must fully understand how the equipment/software will be used in a customer environment. For some industries/products this is quite easy because you have a fixed user interface or some basic knowledge on how the products will be used (a GPS or Gameboy for example). For other industries it can be very difficult to understand how the equipment/software will be used. In telecommunications a lot of the people in R&D do not know how the equipment is used by our customers. We understand how the equipment works in the lab, but we are not aware of the complete list of features or how those features are used or interact at a given customer site.

> Our product support team started customer profiling a while ago, and that information has been used by our Hardening and Independent Quality Test teams (the final two stages of testing before shipping to our customers). One of these two teams uses customer databases for their upgrade testing and exercises the features in the exact same way that a particular customer will in their networks. The other group looks at a group of customers that will be adopting a new release, and designs test plans based on common themes and feature interactions that are apparent across these environments. These two teams were previously using input from Product Management to develop test plans. This was less effective at performing realistic testing.

In an ideal world we would be able to verify each feature in a perfectly simulated customer environment. This would obviously not be practical, but by effectively using customer profiling, it is possible to increase the realism of the testing performed without much extra effort or costs. We have also found that our customers are very willing to participate in these ventures as they find fewer defects when they take the products into their labs. We have had customers come regularly and present their operational views and issues with the R&D team to help us understand their environments better.

Randomness vs. Repeatability and Defect Replication

One of the characteristics inherent in realistic testing is trying to generate randomness in the input to the SUT. This will usually reveal some excellent timing bugs in the SUT, but with the added drawback of having found a problem that you can not reproduce. There must be a balance that strives to create essentially random tests but in such a manner that they can be recreated. A part of the solution to help achieve this balance is to fully document every step that is taken along the way including the timing and specific order of all steps – even if they appear unrelated. In addition, the system under test must be designed for easy debugging, and provide for stimulus logging within subsystems. The system does a lot without any external stimulus (stats summaries, connection audits, PNNI DB synchronization, etc.) that all play a part in the behaviour.

Ideally, it should be possible to identify & fix the defect without being able to reliably reproduce the stimuli which caused it.

Scheduling and Sharing of Resources

In many instances when a complicated or large setup is required to perform realistic testing there will be issues with the scheduling and sharing of resources. It is imperative that individuals and different teams work together to achieve the desired test coverage. Once again, we must call upon imaginative ideas to help maximize the overall effectiveness of limited resources. Scheduling methods (like a web site, or a controlled spreadsheet), frequent status meeting and/or updates, risk management, and cooperation of different groups are all mandatory for the testing to be completed.

> My team verifies ATM switched services on our large ATM/IP/MPLS switch. We have one large setup of 6 nodes that we must perform a variety of functions during each testing cycle. There are many verification tasks that share this resource including: manual test case execution, automation execution, core regression, and an overall system test. In addition, we must share this setup with the design community to debug complex problems that can not be reproduced elsewhere. All of these items must be planned for and properly scheduled for every phase of testing. In peak usage times, we have a weekly meeting where we update the progress of each group and reassess our priorities in setting the schedule for the up coming week. A color coded spreadsheet that shows the plan for the setup is updated at these meetings and a current hard copy displayed in a common area. This clear communication is invaluable in achieving the most effective use of this expensive resource.

Whenever resources must be shared, there is a time cost to the schedule because each group only has access to the equipment for a specific period. There is also a time penalty for reconfiguring the setup and test equipment with each switch of testers. Because of these time constraints, it is imperative that testing priorities are set, planned for, and followed. If there are any slips to the plan for any reason (unexpected blocked tests, defects that prove more difficult to solve, etc.) then the coverage must suffer or the completion date must slip. These are risks that must be considered and evaluated by management at the beginning of each test cycle.

Conclusion

Although there are many factors that must be considered to adequately perform realistic testing, it is possible to achieve a reasonable amount of coverage and acceptable results by utilizing innovative methods of developing test equipment and more importantly the sharing of equipment. The trade off of a lower capital budget is either increased time required or a reduced amount of coverage. These can both be minimized by employing creative new methods and embracing the challenge. A strong sense of teamwork and a common set of priorities for the entire verification team are crucial in the ability to effectively share resources and achieve the required test coverage.