



Ensure Cinderella comes to the ball

Testing is the Cinderella task of software maintenance and development, writes Peter Goldfinch – GFG Group’s General Manager for South East Asia. In this issue of the monthly Goldfinch Report, Peter homes in on the importance of testing before implementation of new EFT systems. Peter’s credentials have been earned over 23 years experience in electronic payments, including pioneering work on ATMs and EFTPOS, and the introduction of credit and debit cards in Russia.

EFT systems are increasingly becoming more complex. There are more bilateral and payment networks to connect, and the number of device types keeps expanding. Originally there was one channel, ATMs and then came POS, followed by phone banking, Internet banking payments and now mobile banking plus payments.

Any change made that is not tested properly can bring down all these services – and with them the business.

The other critical aspect is that many of these switches are older platforms that have been stretched to meet the business demands over an expanded period. Many of these switches represent a business risk as at any time they have the potential to implode. This may happen when the business load pushes beyond a critical breaking point,

“TESTING IS AN EXACTING SCIENCE REQUIRING A COMMITMENT TO DETAIL. DEVELOPING A TEST STRATEGY AND PERFORMING THE ANALYSIS TO DEVELOP ALL THE SCRIPTS FOR EACH TEST CASE REQUIRES AN IN-DEPTH UNDERSTANDING OF THE SYSTEM AND HOW IT IS DEPLOYED IN THE REAL WORLD.”

when customers and merchants are most dependent on the service. Alternatively that small innocuous program change that is made without being fully tested may cause the switch to collapse.

In recent years there have been a number of examples of EFT systems failing – if only for a short time – or simply not being able to handle the traffic volume resulting in numerous incomplete transactions over the peak (Christmas) period. When this happens retailers must resort to manual transactions, or can only accept cash. Commerce slows if not stops.

Recognising the exposure these switches present, financial institutions are looking to upgrade their switches. In varying degrees this task is proving difficult, with some significant failures along the way.

Risk reduction strategies

There are a number of strategies that can be deployed to reduce the risk.

A considerable investment has been made in disaster recovery sites. The issue with these DR sites is they are a mirror image of the primary site. A software deficiency in the primary site will also be in the secondary. So if the primary fails and if the condition is repeated the secondary site will fail.

DR should not be discounted as worthless because a DR site provides protection against hardware failure. The point is it will not provide protection necessary against software failure, if the software is common across both platforms.

Disaster recovery strategies will help to mitigate the impact but they are not substitutes for doing the fundamentals. The fundamentals relate to developing and implementing a well-designed solution and high quality software.

There are a number of tools available in the market to assist developers to write well-structured and maintainable software. The focus of this document is on the proofing.

Proofing is all about testing. As these switches have grown in their complexity the testing effort has become more challenging. Rigorous testing processes need to be put in place.

Traditionally testing has relied on the need to use people to perform the testing. This is often a task given to the less experienced staff. Many organisations set up testing units with excellent generic skills, but rarely with the in-depth knowledge to develop comprehensive test scripts or with the ability to fully interpret the results.

Testing is required to take place at a number of levels. Different organisations will go through various testing processes, often passing through different groups that span the IT and business divisions of the organisation, each with a specific responsibility. The following is a typical four-step process:

1. **Unit testing is the task of the developer who has written the code. This is a task most developers struggle to perform effectively. Normally this is less about a reluctance to test but more about not having an adequate testing environment made available.**

To deliver a testing tool that can be used from the desktop is the requirement.

2. **System testing is again a task normally performed by the development team. It is the developers' assurance testing phase that ensures changes (often from a number of developers), have been made correctly and the system still functions correctly.**

This is the quality assurance phase of the development effort and needs to be comprehensive. A professional development team will not want to deliver low quality work.

This testing by definition must be end-to-end and comprehensive. It must be able to be repeated and test results compared. This task is best automated.

“CREATING CERTAIN CONDITIONS ON DEVICES LIKE AN ATM CAN BE A CHALLENGE. WHO WANTS TO DESTROY SENSORS IN A MACHINE SO A DEVICE FAILS TO PERFORM FUNCTIONS SUCH AS NOTE DISPENSING?”

“IN A TESTING PROGRAM THE FIRST SERIES OF TESTS MAY BE THOROUGH, BUT SUBSEQUENT TEST CYCLES REPEATED AFTER ERRORS HAVE BEEN CORRECTED BECOME LESS RIGOROUS. ‘IT WORKED LAST TIME AND THIS FIX DID NOT TOUCH THAT PART OF THE SYSTEM SO WE WILL SKIP THAT TEST.’ IN PRODUCTION IT PROVES OTHERWISE.”

3. On completion of systems testing, User Acceptance Testing usually follows. In many organisations systems testing and user testing is rolled into the one phase. Users generally focus on the business functions and not necessarily the robustness.

The User Testing for major changes or new development ideally should be a separately managed and performed task, with the software developers being in a support role but removed from any hands on involvement.

4. A final testing stage may be performed that is often referred to as sociability. This is performed by a non-specialist systems group responsible for the migration of software into the production environment. This group’s main aim is to ensure a change to one system will not adversely impact any other system and threaten the whole production environment.

The second step, Systems Testing or a separate phase should be introduced for the critical task of measuring technical quality. A new development may not meet the user’s requirements and although this is obviously important it is the quality of the change in terms of its operational stability that is equally important.

In addition; for a change it is important to verify that all existing functionality not requested to be changed continues to be supported as before.

In summary there are three high level testing goals:

1. Does the change deliver the new user requirement
2. All the functions not changed operate as before
3. Is the system operationally stable

The Devil is in the details

Testing is an exacting science requiring a commitment to detail. Developing a test strategy and performing the analysis to develop all the scripts for each test case requires an in-depth understanding of the system and how it is deployed in the real world.

There is a need to understand the devices attached to the network and the range of potential faults they can generate.

Creating the test plan and scripts, inclusive of the expected results, is where the real effort is spent. This can only be done by the business and technical staff who have an in depth understanding of the application and business processes.

The traditional approach to testing is to treat it as a manual task. The issues with this approach are:

1. Because testing must be performed independently to the developers, this results in a ‘generic’ test where knowledge of the testing requirement is limited. Also testing staff are often junior and inexperienced.
2. In a testing program the first series of tests may be thorough, but subsequent test cycles repeated after errors have been corrected become less rigorous. “It worked last time and this fix did not touch that part of the system so we will skip that test.” In production it proves otherwise.
3. In an EFT environment for example, creating and maintaining a test environment is a time consuming task. Setting up accounts, creating the plastic cards, and ensuring the various devices supported in production are also present in the test environment must be managed carefully. After a series of tests everything must be reset back as it was to enable the test to be repeated.
4. Creating certain conditions on devices like an ATM can be a challenge. Who wants to destroy sensors in a machine so a device fails to perform functions such as note dispensing?

About Peter

Peter Goldfinch, GFG Group's General Manager South East Asia, is a respected analyst and commentator on global trends in payment technology.

One of the original founders and shareholders in GFG Group, Peter has a background of more than 23 years in the information technology industry, most of which has been involved with consulting and systems development for banking and finance customers in 25 countries.

He has particular expertise and experience in payment systems, including mobile payment systems. His career highlights include pioneering work on the first ATM and EFTPOS networks.

In the mid-1990s, he played a key role in the introduction of credit and debit cards into the Russian market, working with GFG's customer SBRF.

About GFG

GFG Group is a highly-specialised payment solutions company, providing its clients with products, advice, and systems integration and outsourcing services. Accredited by the World Bank, the company has established a global presence over the last decade – delivering leadership payment solutions to more than 50 customers in over 40 countries.

A key element in GFG Group's success is its focus on development and investment in five high-demand payment solution areas:

- Card Management
- Mobile Payments
- Payment Tools
- Managed Services

The company's core research and development team is based in Auckland with consultants and technical staff located in the international offices to provide front line 24 x 7 support for customers in multiple geographies.

GFG Group Ltd
Level 10
Qantas House
191 Queen Street
Auckland
P O Box 5825 Wellesley Street
Auckland

Telephone: +64 9 966 7041
Facsimile: +64 9 966 7070
E-mail: info@gfg-group.com

These weaknesses can all be managed by the use of a testing tool.

1. A testing tool can be delivered to the developers so they can test as they code. They can be given the tool to allow them to set up specific test scripts to test specific conditions from their desktop.
2. A testing environment can be established for comprehensive end-to-end system testing. As changes are made to the system, the test scripts can be amended and increased to ensure the testing facility is current, reflecting the changes to the production environment.
3. Testing can be performed effectively and efficiently. Resetting the test environment and rerunning tests can take minutes rather than days. Repeating a comprehensive test because a last minute coding error is found does not mean the go live date will slip.
4. Load testing the system prior to a major release of software can be scheduled and performed within a manageable timeframe.
5. When the operational environment is upgraded stress testing can be performed so the capacity planners have a current view of what volumes, (sustainable TPS) the switch platform can comfortably handle.

The objective is to take the people out of the repetitive testing activity. Use highly skilled and knowledgeable staff to develop and maintain the test scripts to ensure the testing tools are effective.

The tools will increase the testing accuracy by reducing the human error, by reducing the time to test and therefore cost, delivering a high quality switching system and therefore reducing the risk of failure.