Implementing test automation in an Independent Test Unit

© Lloyd Roden

1 Introduction and Background

1.1 The Company

Peterborough Software develops, markets and supports computerised human resource management, financial and distribution systems, and offers a service ranging from bureau, consultancy, education and training to installation assistance and ongoing support. Its solutions operate on most major hardware environments whether open systems, mainframe, mini or PC. It now has a global distribution of products in over 40 countries worldwide and employs more than 500 staff, 40% of whom are dedicated to research and development and customer support.

The company has the largest human resource customer base in the UK, consisting of 1,600 organisations, which includes 73 of 'The Times' Top 100 companies. In addition, Peterborough Software has successful operations in France, Holland, Australia, Singapore, New Zealand and Hong Kong.

1.2 The Case Study and the Test Tool

We have been using test execution tools at Peterborough Software for the past eight years. This particular case study is as a result of the work we have done since 1996 in evaluating and using "QARun" – Compuware's 32bit test execution tool, on our new product "PSenterprise".

This case study unfolds into six distinct areas:

- Evaluation
- Implementation
- Deployment
- Usage
- Problems
- Benefits

Each section describes the processes we went through and key decisions made. I have also included a paragraph in each, entitled "key issues to note", which summarises the salient points.

I am responsible for the Independent Test Unit (ITU) at Peterborough Software which has been in operation since December 1996. This is where this case study begins.

2 The evaluation process

2.1 Our requirements

Our new product, Psenterprise, uses the latest architecture in software engineering, namely 32Bit, ODBC, 3-tier, client/server technology. Having been involved in software testing for over ten years I had already gained appreciation for the benefits of using test automation.

To compliment our new product we needed a testing tool that interfaced well, but at the time of evaluation it seemed that very few products were available. We did not, however, consult the CAST report, which in hindsight would have assisted in this process. "Visual Test" and "QARun" were the two tools we evaluated.

The Development department had already purchased and was using “Visual Test”. This meant that there was immense pressure for us to conform and inherit this tool within the ITU.

I decided that a requirements list was needed to help with the evaluation process. The list is shown in Table 1.

Requirement	**Reason**
Easy to use	This was top on the list. The tool had to be easy to use for the team who were good testers but not programmers so it was therefore vital that the tool did not require too much technical expertise. Also having a tool that was easy to use meant that the testers would enjoy using it and the benefits would be seen sooner, rather than later.
Reliable	Close second was reliability. We had previously experienced a tool that proved unreliable and it soon became ‘shelfware’.
Good features embedded in the product	By ‘good features’ I mean anything that would help the testers perform their job effectively and efficiently. A good point to make here is that vendors are usually very quick to boast of their ‘prize’ features - make a list of these when you are looking at all the tools so that you can compare and evaluate which would be most useful to you.
A comprehensive scripting language	Yes, the tool had to be easy to use, but I was aware that this requirement would be superseded by the need for a comprehensive scripting language as soon as the testers became competent.
Regular maintenance/ upgrades	I needed to know how often upgrades would be received as this showed the level of commitment to enhancing the product as well as fixing issues that would be found.

Table 1 Requirements for tool evaluation

2.2 Relationship with the tool vendor

We also wanted to build relationships with the tool vendor and the requirements shown in Table 2 were important to us at Peterborough Software:

Requirement	**Reason**
The tool had to be developed in line with our own product	It was imperative that the tool was compatible with our own software now, but more importantly – in the future. Developing scripts and tests using the tool is a large investment and we needed to be convinced that the investment was ‘long-term’.
Good customer support	This requirement is a little like having insurance. We were going to find problems with using the tool, but we needed to be reassured that customer support would handle our issues with efficiency and urgency.
Tool training	Comprehensive training was also a requirement – training which was flexible, but also in stages (beginners, intermediate and advanced).

Table 2 Vendor requirements

We invited Compuware to demonstrate QARun on our own software. This was a challenge to them but must be a pre-requisite in the evaluation process. A rule of thumb is that one should not become easily impressed with demonstrations given by tool vendors. Remember that they have mastered the demonstration and this will work every time! The acid test is to see the product demonstrated on your own software.

We then asked Compuware for an evaluation copy of QARun, so that we could compare both tools at our leisure. Once we had decided on the two tools - the actual evaluation process took very little time - probably less than one week.

2.3 The Business Case

Having decided on QARun as our preferred test execution tool, I had to present a strong business case to convince senior management to spend more for our preferred tool. It is worth noting at this point that there is sometimes flexibility when it comes to price negotiation with the tool vendors!

The business case itself took into account factors such as benefits to us as a department: ease of use, important key features in the product and helpdesk support. It also took into account benefits to us as a company: greater productivity within the team, tool being used rather than becoming 'shelfware' and past business with Compuware.

2.4 Key issues to note in the evaluation process:

- Produce a requirement list and prioritise which is most important
- Inform each of the vendors that there are others in the evaluation process
- Invite the vendor to demonstrate their tool on your own software.
- Don’t spend too long evaluating the tools
- Once you have decided on the preferred tool, produce a good business case and present to Senior Management.

3 The implementation phase

3.1 First step: three copies

Having presented the business case and convinced Senior Management that this was the right way we should go I needed to ensure that the tool was successfully implemented.

We had agreed to purchase three copies of QARun initially. This was an important first step – to minimise the initial costs and maximise initial benefits. It was important for us not to be over ambitious in the early stages (to walk before we sprinted!). This philosophy is one I would advocate if you want to see the project succeed.

3.2 Training

Training in the tool is important and the timing is vital. My team consisted of three people, all of whom would be using QARun. To get maximum benefit from the course, we had implemented the tool four weeks earlier before the course was due to take place. This gave us sufficient time to try things out and jot down any questions we had. We also asked Compuware to run the course in-house and on our software.

The problem I have found with ‘standard courses’ is that you learn about how to use the tool on a demo application. The enthusiasm wanes when you try applying what you have learnt back in the office. The bespoke in-house course may cost a little more but, in my opinion, it is worth every penny!

3.3 More than one tool in use

At this stage we had implemented three copies of QARun within the ITU. Development/Product Assurance were still using Visual Test. Whilst it is not necessarily a bad thing for the company to use more than one test tool, it is, in my opinion, not advantageous as we 'water down' the knowledge base. I wanted us to be 'singing off the same hymn sheet' for the benefit of PSenterprise and ultimately for the benefit of the company.

My goal was to adopt QARun as the company test tool. This was achieved by showing the potential of the tool to the rest of the project team.

3.4 Key issues to note in the implementation phase:

- Start small to maximise benefits and minimise initial costs. Think about purchasing 2-3 copies of the tool initially.
- Think about how you are going to train your staff in the use of the tool. Timing of this is essential – ensure that the staff will be using the tool as soon as the training has finished, otherwise they will soon forget what they have learnt.
- Try to have the training performed on your own software, this way it will be more relevant.
- Once implemented – show the tool to others in the project team. It is important that the whole 'project team' buy into the tool.

4 The Deployment of the tool

4.1 Standards and naming conventions

It was important for us to deploy QARun correctly. We had made mistakes in the past in releasing the tools to too many people, too soon. We needed to put together sensible standards and naming conventions that could be easily understood, but more importantly proven to be usable and flexible.

Having only purchased three copies of the tool initially, it was easy to contain the deployment of this tool. Two of us worked on the naming conventions and standards and a draft version was implemented within the team in a matter of a few days. Ownership of these standards and naming conventions remained with one person in the team and that person became the recognised expert in the use of QARun within the company. Some examples of the standards and naming conventions we adopted are shown in Table 3.

Topic	Standards & Naming Conventions
QARun Database	1. Central QARun database to be used at all times and this is to be stored on the common server for ease of maintenance and backup 2. The database must be compacted once a week by the ITU 3. On receiving a major release of PSenterprise, both QARun and associated

	data must be copied and archived as version x.x.
Testdata Files	1. Test datafiles must be created in Microsoft Excel so that they can be easily maintained and can incorporate any comments for documenting purposes. - one xls file for each driver script - each xls file will have multiple sheets to group the testdata logically 2. XLS files to be stored on the central server within folder called ‘itu\data\xls’ 3. Save each sheet in the xls file as a .csv file which is used to drive the QARun scripts and store these on the central server within folder called ‘itu\data\csv’ - the name of the xls sheet and its csv file should be the same - delete out the comment lines 4. Copy the relevant csv files to ‘c:\program files\compuware\data’ when you need to use them.
Naming conventions	1. Scripts names : “ABCCCCCCCCCC” where A = T for test script, D for driver script B = Y for Payroll R for Personnel, A for AMS, CCCCCCCCCC = meaningful name to describe the script e.g. ‘TYEleDef’ for Payroll element definition. 2. Checks must have a meaningful name which describes the check, e.g. ‘PD 48 GP1-0001 Payslip History Elements’ which checks the Elements tab on Payslip History window for employee GP1-0001’s period 48 payslip. 3. Events must have a meaningful name which describes the event e.g. ‘Formula Hypertext window exists’

Table 3 Standards and naming conventions

4.2 The tool expert and champion

In my opinion, it is essential to have someone within the company who is the recognised expert in the use of the testing tool - "a champion". In my own team, this person has not only helped others, but has also ensured standards and naming conventions are adhered to. I am also of the opinion that 'best practice' must be adopted and I was encouraged, having attended the "Effective Test Automation" course run by Mark Fewster, that we were heading in the right direction.

4.3 The word began to spread

It wasn't long before other testers within the company wanted to see how we had used QARun and what benefits we had gained. Visual test users were becoming increasingly frustrated and I was soon asked to demonstrate the power and versatility of QARun. The outcome is that Peterborough Software has adopted QARun as its standard test execution tool.

Having already established the foundations in standards, naming conventions and generic scripts and by presenting key sessions on how we had used the tool, it was then easy to 'roll out' QARun to the rest of the Company. We have to date deployed 19 copies of QARun and it is currently being used with determination and enthusiasm on all our product range.

The use of an Access database as a central repository for scripts, object map entries and checks provides a 'portable' testing tool which encourages the sharing of ideas, scripts and knowledge.

4.4 Key issues to note in the deployment of the tool:

- Deploy the tool when you are ready. Releasing the tool prematurely will lead to uncontrolled scripting and bad practices being adopted. Once this happens it is then difficult to regain control.
- Set aside time at the beginning in order to produce naming conventions and standards for the tools use.
- Assign a key person responsible for the implementation and policing of these standards.

5 How QARun has been used

The philosophy I adopt in all walks of life is based upon the K.I.S.S principle – Keep It Simple Stupid. This philosophy has never let me down and in testing terms, this translates to 'Breadth First', 'Depth Last'.

5.1 Breadth Tests

These tests are designed to test the breadth of our application. At this stage we are not concerned with detailed tests, but want a series of tests which provide maximum impact in the shortest possible time. These tests are usually very easy to automate. Examples of these tests are shown in Table 4.

Script	Description	Purpose
Dialogue Scripts	Script will open all menu dialogues within a given application and will perform a basic 'text' check before closing. Time to run – approximately 5 minutes per application. No data input.	Tests all menu dialogues for each release. Diagnoses any changes to the screens. Used on different configurations (i.e. Win95, Win98, WinNT)

Solo Script	Script enters minimum data required to process a new starter, run payroll and Absence processes and performs a number of key checks. Time to run – approximately 20 minutes per application. Minimum of data input.	Test of 'key' processes within the applications. Checks employee based screens Used on different configurations (i.e. Win95, Win98, WinNT) and on different databases (SQL, Oracle, AS400)

Table 4 Example breadth tests

I can't emphasise the importance of these breadth tests enough. They are relatively easy and quick to produce and they provide you with sufficient confidence to continue with the depth tests and other manual tests.

Just by way of an example, our product has a number of applications namely: payroll, personnel, absence management and recruitment. Each of these has its own dialogue regression script. These took, on average, half a day to produce and would run in 5 minutes. Having 4 PCs available, I can be assured that all application windows would open and I would know whether any changes had been made to the actual windows. This you will appreciate is an excellent first test for any new release.

The SOLO script would put the minimum of data into the system to be able to run the batch processes connected to the application and would check all employee based screens. More work was involved in producing these scripts due to the data entry aspect. On average, a Solo script would take about 3 days to produce and it would execute in 20 mins.

These Solo scripts added to the dialogue scripts, when run, provide me with enough confidence in the system to move to the next stage of regression benchmarks - namely the depth tests.

I have also found these breadth tests (solos & dialogues) useful for testing other aspects which may affect our application, such as; machine upgrades, operating system upgrades (NT version 4 to NT version 5) and database upgrades (SQL version 6.5 to SQL version 7.0). Bugs can then be found quickly and efficiently.

By way of examples as to the usefulness of these scripts:

a) It was our payroll solo script which found a problem when we upgraded from SQL version 6.5 to SQL version 7.0

b) It was the dialogue scripts which found discrepancies between Windows95 and Windows98 clients.

5.2 Depth Tests

These tests are designed to ‘drill down’ into the system, testing various features and attributes of the system. We at Peterborough Software call these regression benchmarks ‘scenario tests’. Some examples are shown in Table 5.

Script	Description	Purpose
Scenario Scripts	These scripts test a particular scenario. The size of this scenario is variable as is the time it takes to run. Some of these scenarios are fully automated and some have manual intervention Time to run – varies between 1 hour and 2 days. Lots of data input	1. Tests various test cases within each application. 2. Builds further confidence in the system. 3. Used on different databases SQL, Oracle, AS400

Table 5 Example depth tests

The time taken in initially setting up the scenario tests was in fact longer than anticipated. Past experience had showed that scripting usually took four times longer than manual tests and I had always used this ratio (4:1) to determine whether it was worthwhile in automating the tests.

This time the ratio initially was probably closer to 10:1. The reason for this is that we took the decision to produce scripts that were 'data driven' rather than coding the data into the scripts. The result is that initial set up time takes longer but the benefits are as follows:

- Scripts are easier to maintain
- Scripts can be copied and used as templates for similar style windows
- More tests can be run without changing the scripts. This can be achieved by changing the data.
- Whilst initial set up times are longer the net effect is one of 'long-term gain'.

A point for managers reading this – please allow your testers sufficient time to build ‘generic’, ‘low-maintenance’, ‘data-driven’ scripts. The long term benefits far outweigh the short term costs.

5.3 Other Tests

Having spent the time developing these scripts, we decided to make more use of what we had and build further benchmarks relatively easily. By re-using scripts, but increasing the volume of data we were able to build a ‘performance benchmark’ and by looping the solo script we were able to build a ‘robustness benchmark’. These are shown in Table 6.

Script	Description	Purpose
Performance Scripts	Performance scripts have been developed from existing scripts, but the amount of data has increased substantially. Copying data in excel is very easy, therefore increasing the employee data from 1 to 10,000 means that the Solo script can become a volume script in an instant. Time to run – varies depending on the amount of data	1. Ability to generate large volumes to test database performance 2. Used on different databases SQL V's Oracle V's AS400 for comparison
Robustness Scripts	Looping the part of the solo script means that we can perform the same test over long periods of time to check the robustness of the system. Also, by using QARun's clock checks we can time transactions on various specification PCs and can determine the degradation. This is vital for us so that we can advise customers what response times to expect on the different size PCs (diagram 1 is the result of such a test).	1. Check for memory leaks 2. Check the difference in response times on different specification PCs

Table 6 Example depth tests

Figure 1 shows example results of robustness testing. The tests ran for a period of 8 hours and were repeated on a P60 and P166 machine both with 32mb RAM. The top line shows the Pentium P60, and the lower line shows the Pentium 166.

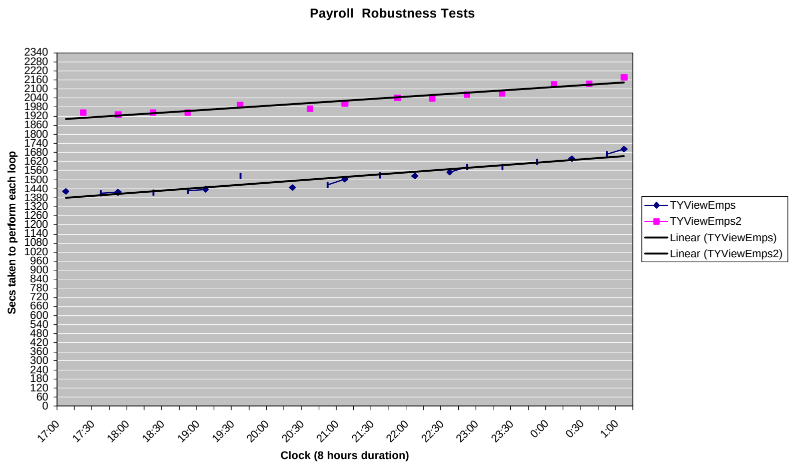

Figure 1 Results of robustness tests for the Payroll system.

Figure 1 shows that there is a steady/linear degradation over the 8 hour period, which does indicate a slight memory leak. The other observation which interested me was that there appeared to be an improvement in response times of up to 30% when using a P166 machine when compared to a P60 machine.

The above information was achieved relatively easily using existing scripts with clock checks built in.

5.4 Key issues to note in the use of the tool:

- Start with simple scripts that cover the whole system quickly – Breadth tests
- Build upon the Breadth tests to produce your Depth tests.
- Use test datafiles to drive your scripts. This makes the scripts easy to maintain and more versatile.
- Think about how existing scripts can be used to run tests such as ‘performance’ and ‘robustness’

6 Problems we have experienced

So far I have painted a very positive picture of how we have evaluated, implemented and used QARun within Peterborough software. And it is a success story. But it hasn’t been free from frustrations and headaches. This section looks at some of the problems we have encountered during the past two years – this section will keep your feet firmly on the ground!

6.1 Running of scripts

Most of us have the perfect picture in our minds, that we can run a script overnight with very few problems. We start the script, put our coats on and expect the script to have finished perfectly the next morning. While in theory this is possible, my experience has shown that in reality it is unlikely. I would estimate that less than 40% run without some form of manual intervention.

There are a number of reasons for this, but the point I would make is don't become overly optimistic and persevere in trying to make the scripts more robust.

6.2 Maintenance of scripts & data

Don't be fooled by the tool vendors, there is a large overhead in maintaining your scripts and test data. Plan this into your project schedules. The problem here is when you don't plan for this activity. Obviously the amount of maintenance is dependent on the nature of the changes. However this is not linear – a small change (such a new ocx file) can have a large effect on your scripts.

By way of an example of this non-linear effect, our own developers changed from using VB4 (Visual Basic version 4) to VB5. This gave the developers more flexibility in developing the software, but gave us testers a 'headache' because we had to change the controls within our testing tool so that our scripts could run against the new system.

A note must be made here for both testers and developers - we must work closer together, to understand how the system is being developed and how we are proposing to use the testing tool. This attitude promotes a good working relationship.

6.3 Maintenance of the tool

When purchasing the tool, two options were available: Buy the tool with or without the maintenance agreement. I would always advocate taking the maintenance agreement option as you will encounter problems with the tool that will hopefully be fixed in subsequent releases. Also the tool vendors will be enhancing the tool for your benefit.

However be aware that upgrading the tool takes time and needs to be planned. It might also mean that new problems arise which were not expected. We ourselves had problems with one such upgrade to the database which did cost us time in resolving the issues with Compuware.

6.4 Bugs in the tool

At times we have found problems with QARun. Time has been spent determining whether the problem is with our software or the tool. Most of the bugs found have indeed been fixed by Compuware in subsequent releases, which is a further endorsement of taking out the maintenance agreement. However there have been times when problems found could not be recreated – these have been the most frustrating, as they have been very real problems to us.

6.5 Configuration Management of Scripts & Data

The more people involved in scripting and running of the tests the more likelihood there is of problems occurring in version control. Even with strict change control, people will want to get their tests run as quickly as possible and corners will want to be cut!

I have five people in my team and we share a central QARun database. The benefits of this certainly outweigh the disadvantages. However there have been times when changes made by one person have caused scripts not to run. Change control is not as good as it should be but is manageable with only five people. I could envisage serious problems if there were more people involved. It is worth pointing out at this stage that QARun does have good version control built into the product which makes it a little easier to manage.

6.6 Don't Automate too soon

There is a distinct danger to try to automate tests before the software is ready. We have fallen prey to this. The result has been to code around 'buggy' software. It is far better to wait until the software is robust and reliable as this then leads to beneficial and positive scripting.

6.7 Key issues to note from problems we experienced:

- Don't be too optimistic in running scripts overnight with no problems – this is a nice theory, but reality dictates otherwise.
- Recognise and plan for the maintenance of your scripts. This is no easy task and could result in the test tool failing because of poor management. Remember that small changes could have substantial consequences in your scripts.
- Don't automate too soon
- It is advisable to take out the maintenance agreement, but be aware that the upgrade of the tool could take some time and could present unexpected problems.

7 The benefits achieved in two years

This is a success story of how we have evaluated, implemented, deployed and used QARun at Peterborough Software - despite some of the problems we have encountered along the way.

I guess the success should be measured by how Peterborough Software have responded to this tool. At the start we were given permission to purchase three copies of this tool for sole use within the Independent Test Team. To date, QARun is now the company standard in test execution tools – Peterborough Software have 19 copies around the Company and it is being used on all our products.

In this section, I will outline the benefits that we have achieved.

7.1 More with less

The number one benefit has been the amount of tests that we can now run in the timescales given. The regression tests we now have would take us approximately four man weeks to run manually (that is if we were to key and check everything perfectly). During our last release, these same tests using QARun took two of us (using 5 PCs) only two days, so four man-days instead of four man-weeks, a five-fold improvement.

7.2 Improved testing

The test execution tool has given us more time. More time to develop better tests, more time to run ad-hoc manual tests and more time to think about what to test. All this has led to improved testing of the Psenterprise product which has ultimately benefited our customers.

We can also be assured that input and checks (using QARun) are accurate, which ultimately builds confidence.

7.3 Machine usage

Our PCs are usually running scripts unattended overnight, which is making the best use of the resources we have. By having 2 PCs per person, productivity increases as we are able to develop scripts on one whilst running regression scripts on the other.

7.4 Lack of regression in our product

The acid test for us, being in the software industry, is that our customers are satisfied with the quality of our products. Customers should also be confident when new product releases/upgrades are shipped to them. Automating our tests has meant that there is very little chance of product regression. The measurable benefit for us is fewer issues being raised by the customer.

7.5 Performance tests

I have already mentioned the ability to run performance tests relatively easily. Building a 10,000 employee database can be run over a weekend. Performing this manually would take considerably longer and would probably end up with a lot of unhappy people!

7.6 Customer impact

I have been privileged within Peterborough Software to be able to present the work of the Independent Test Unit to our customers. The presentation has included the work that we have achieved with QARun. Without exception, all customers have been impressed with the work undertaken using the tool. The customer impact has been encouraging to say the least.

7.7 Testers being acknowledged in a skill

QARun brings with it sophisticated scripting language (as do many of the test execution tools). Testers are encouraged to improve in this programming skill, making scripts structured, maintainable, easy to read and efficient. This skill set is, in my opinion, as important as the programmer next door building the system that is being tested.

7.8 Better morale within the team

"Running a manual test once is exciting, running a manual test twice is a necessity, running a manual test more than twice is boring" – quote from Lloyd Roden's testing handbook! We have to wake up to the fact that regression testing is boring if it is to be done manually. I have found that by introducing a test automation tool, the testers will get on and do the work they do best – creative testing. Someone once said that "human beings were never created to repeat something more than once", I'll say that again.

8 Conclusion

It has been a privilege working for a company like Peterborough Software who take testing seriously. We have achieved a great deal in the last two years using QARun, but we must not rest on our laurels. There is so much more that can be achieved - with thought, training and certainly in working with other companies with experience in test automation.

Lloyd Roden

Lloyd has been involved in the software industry since 1980, studying Computer Science at Leicester University. He joined Pearl Assurance as a programmer in 1983 and worked there for five years, before becoming a Senior Independent Test Analyst for Royal Life. Three years later he joined Peterborough Software where he became project manager for the product assurance department. He also set up and managed the independent test unit for over 2 years. During his 8 years at Peterborough Software he worked through key issues in test management such as; testing to pre-defined deadlines, managing a test team, successfully implementing and using test automation tools and building quality into the testing process. He joined Grove Consultants in April 1999.

Lloyd was chairman of the QARun User Group for three years, and is a lively and enthusiastic speaker at conferences and seminars. He has spoken at EuroStar, TCS, SQE and Unicom conferences. At Grove Consultants, he provides consultancy and training in all aspects of testing, specialising in test management, people issues and test automation.

Lloyd is married to Chris, and when software testing, consultancy and training isn't taking his time and energy – his two children are!