



**SOFTWARE QUALITY AND
TESTING SUPPLEMENT**

Moving testing on

Where is testing going next?

Building quality earlier in
the development lifecycle

JAVA vs COBOL? We put two
programmers in the ring

Plus 16 pages of software
quality and testing insights



In association with Micro Focus
© 2012 Micro Focus. All rights reserved.

Welcome to Q.T. Supplement



I'd like to welcome you to this special supplement on software quality and testing. As a tester, you are more than familiar with the challenges involved in testing, and that includes mainframe applications. How we go about

addressing these challenges is the focus of this publication. It goes without saying that quality is an important part of the equation. But before we look into the detail, I'd like to give you my own personal take on the subject.

Kevin Brearley, Director of Product Management, Micro Focus

STRESS TEST

Why testing is putting a strain on resources

Companies are under pressure to deliver applications faster than ever whilst maintaining standards of quality. How can they rise to this challenge?

Businesses around the world currently spend more than \$50 billion on application testing and quality assurance, according to Pierre Audoin Consultants (PAC). And the market is set to continue its robust growth. Over the next five years the testing market alone is expected to grow by a compound annual growth rate of 15.4 per cent.

Even though IT budgets remain tight, businesses have no choice but to increase their spending on testing. Think of any large corporation. The chances are that it is under immense pressure to deliver high quality applications in a short space of time. Rapid speed to market counts for everything in today's highly competitive business environment.

Testing is putting intense strain on enterprises' time and internal resources. To ease the burden, companies are increasingly turning to advanced tools and technologies. This trend will accelerate: expect to hear more about crowd sourced testing, Testing-as-a-Service (TaaS) and cloud testing over the coming months.

These services and technologies are perfect for testing applications stored in the cloud or on mobile platforms. But they're of little practical use for testing applications housed on a mainframe. And that is where the majority of most companies' testing continues to be concentrated.

In 2012, we will see more companies moving their testing off the mainframe. Increasingly, businesses will be doing their heavy-lifting testing on other environments such as Windows Server. These platforms will help many businesses resolve the major bottleneck and cost constraints presented by the mainframe.

Businesses that release test processes from the constraints of the mainframe will also be able to complete testing cycles faster. And this will give developers more scope to focus on quality.

2012 could be the year in which testers finally wrestle control of their applications from the mainframe.

THE NEED FOR SPEED: faster time to market with testing

It's not just Tom Cruise who feels the need for speed. Today's companies operate in a constantly shifting business environment. Companies that have the ability to react to commercial opportunities in real time will flourish and prosper. The rest will be left behind.

In many ways, today's market represents a vast Darwinian ecosystem where only the fittest survive. It goes without saying that this battle for market supremacy is having a profound effect on the way in which businesses deliver applications. And the mainframe is right in the middle of the crossfire. Often ridiculed as cumbersome and archaic, the mainframe has set a benchmark in business performance and reliability that has yet to be surpassed. But these workhorses of the IT world require a huge amount of maintenance.

Aside from consuming precious resources, mainframes are often off-limits to testers and developers. Pre-production testing can only be scheduled according to the capacity available on the mainframe. And that capacity is often tied up in running core production systems.

Even when service delivery managers get access to the mainframe they are constrained by its existing capacity. Worse, they are frequently under pressure to reduce the amount of

Millions of Instructions per Second (MIPS) and the time that important testing activities consume.

Given all these constraints, is it any surprise that service delivery managers struggle to meet their deadlines? They need to have the flexibility to speed up application change but are hamstrung by the environment that they are operating in.

Accelerated time to market

So what then, is the alternative? Moving applications off the mainframe and into another test environment is one option. Products such as Micro Focus' Test Server allow users to scale test capacity to meet fluctuations in business demand. The production of business-focused applications can also be sped up because Test Server expands the test environment beyond traditional test teams. It allows business users and developers to fully participate in the testing process.

What's more, these products allow delivery managers to identify issues considerably earlier in the development cycle, which reduces the need for costly re-working. They also allow testers to accomplish their testing in a shorter time frame.

Another consequence of testing in this fashion is reduced cost. Products like Test Server mitigate the need to make substantial investments in new mainframe MIPS. They actually help reduce mainframe MIPS consumption because more testing is done in the new environment.



We're on a MIPS control mission

It's no secret that mainframes are extremely costly to maintain. Most businesses have a finite number of MIPS and don't want to exceed their quota. Thankfully, there is an alternative – Windows Server. By moving applications onto a Windows environment, Micro Focus customers have reduced their IT costs by as much as 77%. Development throughput can be improved by 40%. It seems the message is getting through to businesses – according to Gartner*, organisations are now routinely transferring workloads worth 2,000 to 3,000 MIPS onto other environments.

THE PRESENT FUTURE: what's around the corner in testing?

Testing applications can be a time consuming and prohibitively expensive exercise. No more so than on the mainframe. Fortunately, there is another way.

The humble mainframe continues to dominate the enterprise space. The 50 largest banks in the world use mainframes to run their mission-critical applications as do 24 of the 27 largest retailers.

The applications that are housed within mainframes need to be updated and enhanced on a continual basis. And since these applications are so critical to the enterprise, every update has to be thoroughly tested prior to release.

An emphasis on cost

In today's cost conscious business environment, IT budgets are under pressure as never before. Your CIO is often tasked with producing good quality applications at much lower cost. Considering that testing is a large component of the overall cost of developing or updating an application, is this really possible?

Traditionally, organisations have addressed this challenge through package replacement and application

re-writing. The former approach forces organisations to do business in the manner dictated by the package's functionality or spend a considerable amount of time and money customising the package. Workflow that is fundamental to the way a business operates can be lost, and costly, complex data modelling and retaining is invariably required. Many package implementations are never fully completed or go over time and budget.

Re-writing applications is an even riskier approach. The majority of re-engineering projects either fail completely or are delivered years later than they should be.

Moving the testing environment off the mainframe

There is an alternative and far more cost-effective way to test mainframe applications – re-hosting mainframe testing. Delivery managers can move the testing environment away from the mainframe and onto lower

cost commodity hardware. From a financial perspective this makes complete sense. Application testing on the mainframe can take up a huge amount of processing power and cost – sometimes as much as 50% of available MIPS. Re-hosting testing on Windows, for example, can halve testing costs and significantly reduce the amount of time needed to deploy a new release or package.

Of course, this all counts for nothing if the completed applications have bugs in them. Testing in a Windows environment mitigates this risk because teams are also able to perform more testing within shorter timeframes at lower cost, which reduces re-work expense, speeds up the completion of testing and generally improves quality.

Does this really work? Yes, it does. Delivery teams today can complete testing phases faster by moving their testing environment onto Windows Server. And test capacity can be scaled up and down as required which is not typically possible in a mainframe environment, meaning organisations



Opening a new window on testing

Delivery teams dread testing applications on the mainframe because they are handcuffed to the mainframe environment and capacity constraints.

Windows frees testers from these restrictions in an instant. All testing takes place on Windows Server rather than the expensive and resource-hungry mainframe. The mainframe testing bottleneck is eliminated and IT application service delivery is dramatically improved. What's more, Java and .NET



programmers developing composite applications can exploit this new testing environment to ensure their application changes meet the same quality standards as their COBOL counterparts, but without having to rely on the expensive and often unavailable mainframe. Now, more than ever, it makes sense to escape the mainframe trap.

Shining a spotlight on Cloud and mobile

We've all heard the buzz about Cloud Computing and mobile, but is it really justified? Over the next four years, according to almost half of all CIOs, the answer is yes. They expect to operate the majority of their applications and infrastructures through Cloud technologies. Will mainframe applications be part of the picture? One has to expect so.

Mainframe applications are already being tested in environments like Windows Server. There is no reason why an environment shouldn't allow users to test mainframe, cloud and mobile applications at the same time.

Increasingly, applications that run in the Cloud and on mobile devices utilize Java. But the back end processing continues to be done using COBOL. Being able to test both languages in the same environment will provide companies with significant economies of scale and allow them to deliver applications even faster than before.

And by providing a unified environment for executing these applications, organisations are on the first step towards being able to re-host the whole production system on those lower cost platforms, where it makes sense to do so. It may sound like a pipe dream, but it will be a reality in the near future.

can plan to test for peak loads and critical application batch runs with greater confidence and flexibility than before. Added to which, further Micro Focus technology can then be used to establish a fully automated process for testing these applications both under Windows (for low-cost system testing) or back on the mainframe (for final pre-production testing).

Moving the testing of core applications off the mainframe doesn't just make sense from a cost perspective. It can also help companies reduce risk and stay one step ahead of the competition.

QUALITY AND TEST - are they on the same page?

You may think that they are only partly related, but quality practices and testing are inextricably linked. And their interaction starts at the very beginning of the development cycle.

For many organisations testing is something of an afterthought. Typically, an IT team will develop the applications that the business requires and only test them once they have been built. There are considerable flaws to this strategy. Substantial development resources can be allocated to applications that are peripheral to the company's core business. Testers can wind up testing applications that are generally ignored by the business in preference to ones that are regularly used.

Appropriate quality and testing practices need to be built into the beginning of the development cycle – not at the end. And that process starts with the applications a company owns.

Managing business processes

Most large enterprises use application portfolios to control their core business processes. However, over time, these portfolios can become more complex, harder to manage and less aligned to the needs of the business.

So how can a company regain control over its business processes? In the first place, it needs to perform an audit, a quality check if you like, of its existing applications. This can be achieved through Application Portfolio Management (APM). A key step in any IT modernisation programme, APM allows IT directors to evaluate the value of business applications from a cost, value and risk perspective.

But what happens if this exercise itself is poorly implemented? Then it is next to useless. Proper procedures have to be built into any APM operation right at its inception. Establishing benchmarks for data sourcing is a key starting point. Unless the right data is extracted from the organisation in the appropriate way, it will be virtually impossible for directors to determine which applications to retire and retain.

Establishing key performance indicators

Take technical metrics. Every APM operation involves the collection of technical metrics related to the quality of enterprise applications. But these metrics need to be refreshed

on an ongoing basis, something many organisations fail to do. An APM solution should also assess a wide spread of application environments. After all, the typical application portfolio contains everything from COBOL to Visual Basic and Java.

Once raw technical data has been extracted it needs to be analysed properly. This is only possible if businesses put in place appropriate key performance indicators. Again, many companies fail to put sufficient effort into this activity.

The right metrics also need to be obtained from stakeholders. Given that a company's stakeholders are typically the main users of applications, it is surprising

how rarely they are asked for their opinions. APM solutions should include a surveying mechanism that allows users to readily deploy questionnaires specific to their organisation.

It is well worth building quality procedures into an APM solution. Development teams today are under huge pressure to deliver. They also have fewer resources available to them. Ensuring that these resources are allocated to the right applications is critical. This is particularly the case when it comes to testing. At a time when 80% of IT budgets are dedicated to 'lights-on' activities, CIOs can ill afford to waste money testing redundant applications.

Failure to test can trigger disaster

You've established what applications to retain, but what happens if you don't actually test them? Will you really be courting disaster?

Enterprises are notoriously secretive about system crashes, but stories still manage to trickle into the media. There's the retailer that lost £15m on one of the busiest days of the year due to a system crash. Then there's the online ticketing system that collapsed leaving baseball fans out of pocket. And less we forget, there's also the three day network outage that affected millions of BlackBerry subscribers in October 2011.

These are just a few examples that illustrate the vital importance of linking quality and testing.



When two worlds collide

The difficulties with testing enterprise applications are compounded by the fact that many core systems span language and even platform boundaries. For example, many major mainframe applications, running COBOL, are presented to end-users by a front-end Java application interface. So, what's the problem here. They're both well known languages albeit with different histories. One, COBOL, is deemed to be rooted in the past, while the other, JAVA, is currently stealing all the headlines.

Ian Barrow is currently a programmer who works with JAVA, while Charlie Grant works in COBOL in a financial company which has recently re-hosted to a Windows environment. Derek Britton of Micro Focus referees an intriguing interview, as our two contestants slug it out – discussing the merits, or otherwise, of both languages.



Derek: We are here to discuss what's different, good, or complementary about your two different languages. So who would like to lead with a question?

Ian: I would start by saying the big difference that exists in our two working environments is agility... the fact that I use rich, productive tools. JAVA is a very visual and intuitive test environment... that's got to be something you envy, Charlie.

Charlie: Not at all, I already have what you have. I'm probably working on the same IDE as you? My preference is Eclipse but I also use Visual Studio. It just happens that the Apps I'm working with are COBOL-based. They were moved off the mainframe with very little disruption, and so now my interface is totally visual, with all the bells and whistles you have.

Ian: So, do you have Intellisense?

Charlie: Yep. Intellisense, auto-completion, background parse, red

squiggles, the full IDE. For COBOL, it's the same as any other new language. And anyway, COBOL was the original write-once run-anywhere language and had multi-platform support long before Java was born.

Ian: Well, JAVA has grown up it's definitely now the programming language of the future.

Charlie: Of course Java is superb for building front ends.

Ian: Yes, and the code runs anywhere, it runs on tons of platforms, and it's high-quality code. We're talking about the difference between object-oriented versus the old-fashioned procedural techniques of COBOL. Who really does that anymore?

Charlie: Well that's a lot of claims. I said JAVA is ideal for front ends but that doesn't mean COBOL isn't better at running the back end. It always has been. And as for the quality of code, well, we've been churning out high-

quality enterprise systems for many more years than you guys have.

Ian: It's still a bloated, verbose language though, admit that.

Derek: ...that is an argument that is often levelled at COBOL, Charlie.

Charlie: Sure, many people think it's a verbose language. That's probably because we have billions of lines of code out there. But it's only as wordy as you need, and more to the point it is readable. And also, as it's syntax driven, you can do a one-liner in a COBOL program like DISPLAY "Hello World" and it compiles and runs... interestingly Ian you could be up and running with COBOL in a couple of hours flat: because you can read it, understand it, and use the same IDE to code it, there's nothing stopping you.

Ian: Well it might be easy to pick up but, look, you basically can't conceive and implement Apps quickly using a small team with COBOL. JAVA does

that and that can only be good for our business users.

Charlie: What's really important for the business is the value of the business logic and data that already exists, getting at that, and using it in different ways. Who would choose JAVA to try to build an entire new banking application for example?

Derek: Ian sorry to interrupt, I want to move you on to another important area, and that is testing for mobile and cloud, anything to be said here?

Ian: Well obviously JAVA will be on mobile and in cloud. It's a mainstay language there for development and test.

Charlie: You're right there, Ian. Java is designed for that and that's why COBOL sticks to the back end business logic. You can now leverage COBOL systems wherever they might be running by interfacing with new technologies like web services, for example. So your back-end core business system can be

accessed by a web portal, a mobile app, or whatever.

Ian: You make that sound so easy, yet it can't be.

Charlie: OK, you're thinking 'how do I call COBOL from Java when a mainframe App has all of these weird data types I can't use from Java'? Well, we can talk language-to-language. You can create interfaces to COBOL programs that Java developers can use without worrying about what they are calling.

Derek: That's an interesting proposition. Just to give you both a chance to finish this off, I'd like to ask what the future holds employment-wise for developers such as yourselves.

Ian: That's easy, I think the JAVA developer has never been more in demand.

Charlie: And that's actually true of COBOL too – no Ian it is! – really.

There's a lot of major enterprise software projects out there. And I think programmers and testers who have COBOL and new programming language skills such as C# or Java are well placed to take a lead role in that.

Ian: So who knows we might even up working in the same team!

Charlie: Really, it's not so crazy as you think. I know I'm collaborating more now with other programming teams, and using testing technology that bridges COBOL and Java environments, there are a lot of new composite applications out there.

Derek: Well let's shake hands on that thought please, gents. Sounds to me like there is space for both languages to flourish, and start to cooperate more, both for developers and testers. Thank you for giving your time to this debate.

THE MISSING LINK: addressing the skills gap

Old fashioned but indispensable – COBOL will dominate computing for many years to come. But with a shortage of experienced programmers on the market, how can businesses get the support they need?

Without COBOL the world would literally grind to a halt. COBOL manages every aspect of our daily existence. Whenever you use an ATM, stop at a traffic light or shop online, you're likely to be interacting with a system that runs on COBOL.

The statistics speak for themselves. There are currently 220 billion lines of COBOL code in existence, a figure that equates to almost 80% of the world's actively used code. Three quarters of the world's business data is processed in COBOL, while 90% of all financial transactions are handled by the language. Every year, COBOL systems are responsible for processing 80% of point-of-sale transactions and connecting 500 million mobile phone users.

Given the central role that COBOL plays in our daily lives, it seems scarcely believable that the language is barely studied any more.

A fifty year old language

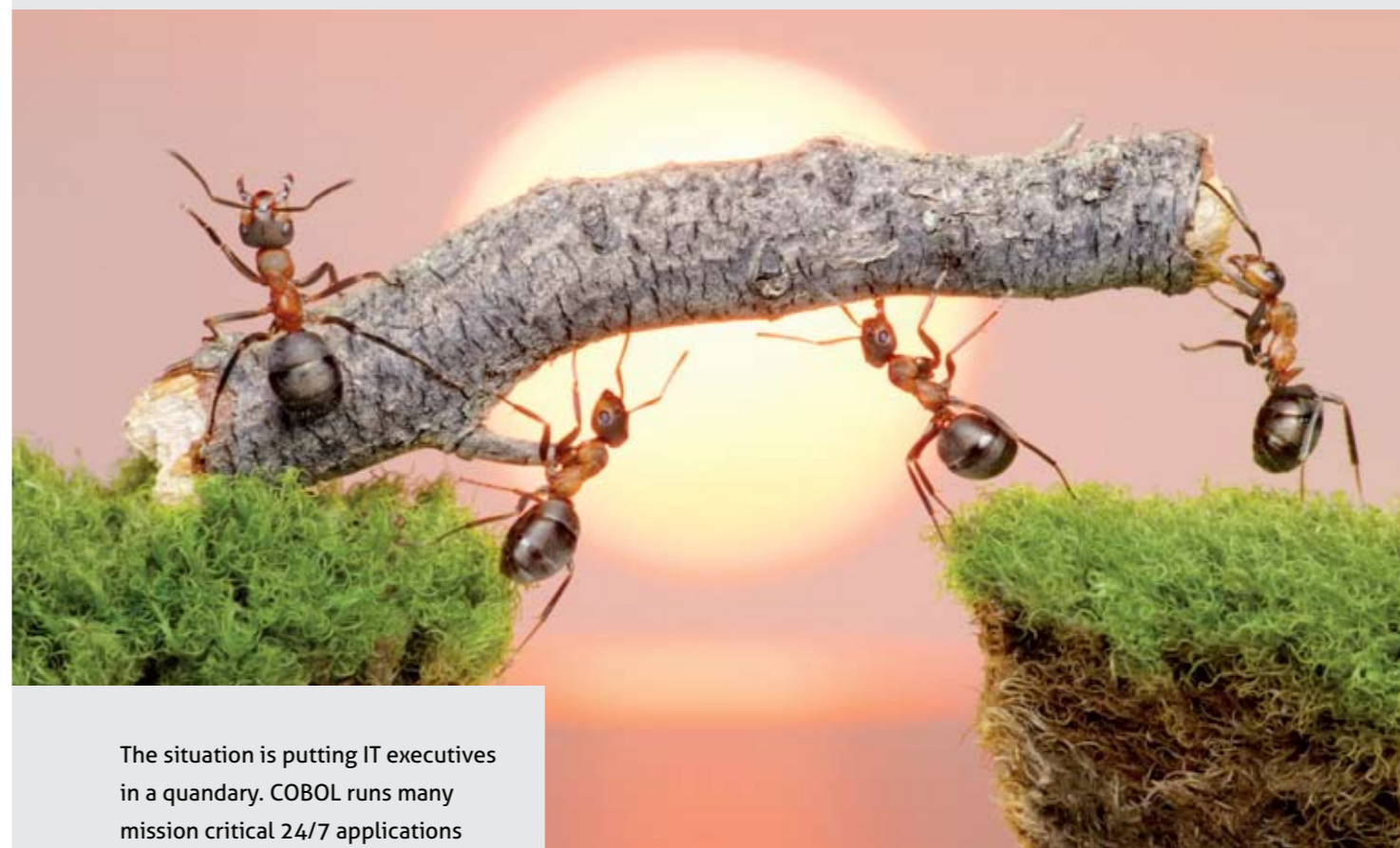
Antiquated, archaic and outmoded – those are some of the more pleasant expressions that have been used to describe COBOL. Others have been less kind. Computer scientist Edsger Dijkstra said: "The use of COBOL cripples the mind; its teaching should, therefore, be regarded as a criminal offense."

Why has COBOL plummeted so much in popularity? Part of the reason is generational. COBOL came into existence on 28 May 1959 – more than fifty years ago. It pre-dates the microprocessor by a whole decade and was running the bulk of the world's business systems before the likes of Microsoft, Apple and Oracle had even been established. For a generation brought up on Java and .NET, COBOL is decidedly unsexy. Most graduates find the idea of operating in a green screen mainframe environment deeply

unappealing. There is also widespread perception that it takes 100 lines of code in COBOL to write something that can be done in only 20 lines of another language – something the advocates of COBOL hotly dispute.

The disparaging posts added to a recent article on COBOL give a flavour of how the language is viewed. One respondent wrote: "COBOL is a Vietnam-era computing language so outdated that many college students don't even bother to learn it anymore." Another wrote: "The sooner the dinosaur is extinct... the better."

While many students balk at the idea of studying COBOL, many highly experienced programmers are retiring. In 2004, Gartner estimated that there were two million COBOL programmers worldwide, with the number declining by 5% annually. This trend is bound to accelerate over the coming years as more baby boomers reach retirement age.



The situation is putting IT executives in a quandary. COBOL runs many mission critical 24/7 applications that are too costly to replace. At the same time, there is a shortage of COBOL programmers on the market to maintain these systems. In 2007 Micro Focus conducted a survey of CIOs. Almost three quarters said they were struggling to recruit trained COBOL professionals.

An alternative approach

Making COBOL more marketable and encouraging universities to incorporate the language into their curricula may solve this issue eventually. But this is of cold comfort to companies that need to solve their IT maintenance issues now.

Companies have responded to the skills shortage in COBOL in a number of ways. Some businesses are outsourcing

COBOL work, while others are bringing in contract COBOL programmers. Neither of these approaches is cost-effective or sustainable over the long term.

To plug the skills gap, more IT graduates need to be encouraged to embrace COBOL. Micro Focus is leading the charge by working with hundreds of schools and universities in a number of countries to boost the number of graduates with COBOL skills. At the same time, efforts are being made to make the language more user-friendly – COBOL can be used within a Windows (and .NET) or Java environment.

It will take time to change the wider IT community's perception of COBOL but the fight has begun, in earnest.

The future of COBOL

COBOL continues to evolve and flourish in new ways that make it more accessible, relevant and valuable than ever. Recently, Micro Focus and Microsoft announced a strategic relationship to help businesses modernise COBOL applications by porting them onto the Windows platform. It is already possible to work with COBOL in a Java environment.

If you're tied to the mainframe, you may still be using COBOL in a green screen environment. But using the latest technology available, now this language is fully accessible to a completely new demographic.

Testing you can bank on

“By facilitating higher quality scalable testing and detecting errors earlier in the software development lifecycle we hope to be able to deliver changes faster and deliver more robust software through to the User Acceptance Testing phase and subsequently production.”

A leading bank put the spotlight on the entire delivery process to find ways to accelerate innovation and delivery.

Testing was a major cost element of the process – and therefore came under scrutiny. A senior IT manager remarked: “We needed to find an innovative solution that would facilitate higher quality, scalable testing.”

The bank has adopted the efficient application service delivery approach offered by Micro Focus through its Test Server product. Exploiting this cost-effective mainframe testing solution, the spokesperson commented, “We will be able to deliver changes faster and deliver more robust software through to the User Acceptance Test phase and subsequent production.”

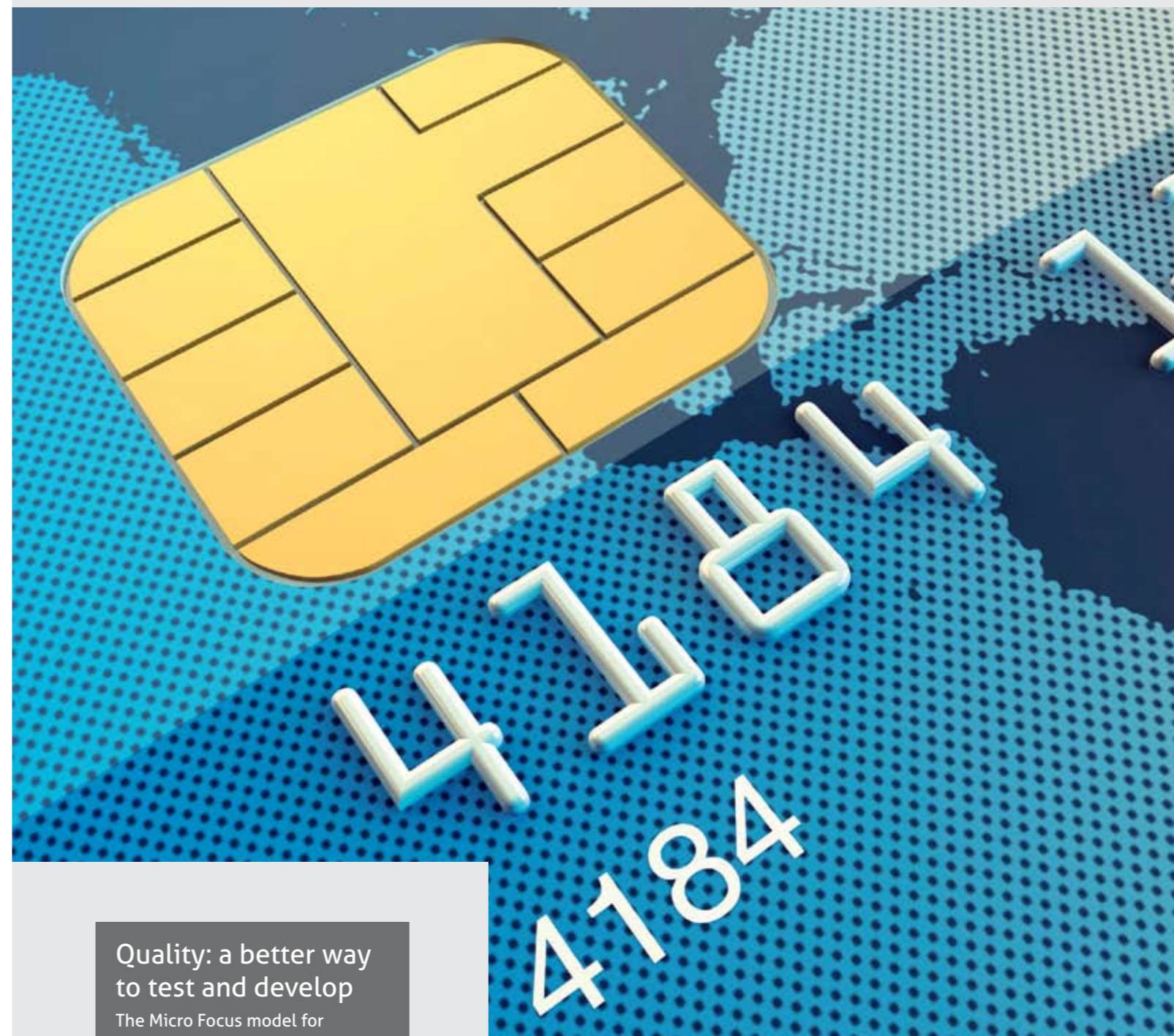
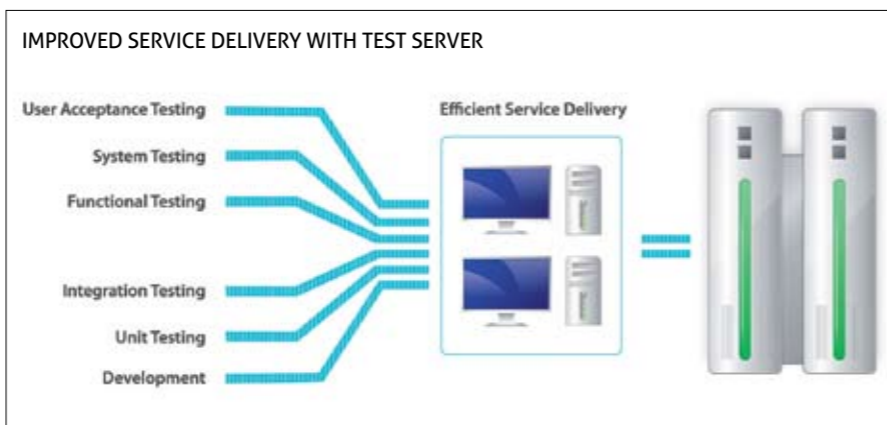
Remarking on the same solution, the industry journal Professional Tester commented: “Integration, system and even acceptance testing could be done with no need for a real mainframe!”

Challenge

- Test activity constrained by mainframe resources
- Quality of applications is a key consideration
- Mainframe environment is very complex


Benefits

- Shorter development lifecycle
- Clear separation of unit and integration test cycles
- Faster delivery of changes with higher quality



Quality: a better way to test and develop
 The Micro Focus model for testing offers ground-breaking quality, flexibility and cost-efficiency. What was once a capacity bottleneck is removed – and time-to-delivery can be boosted, delivery quality raised, and the development process can be unified.

Need to transform the mainframe test environment? The sponsors of Q.T. Supplement are Micro Focus, a company with 30 years' experience of modernising and maximising the value of core applications in both mainframe and distributed environments.

Contact us by email at mainframe.testing@microfocus.com to learn more, or visit www.microfocus.com



TEST Magazine, c/o 31 Media Ltd, Unit 8a, Nice Business Park,
Sylvan Grove, London, SE15 1PD
Telephone: +44 (0) 870 863 6930 Email: info@31media.co.uk



Micro Focus, The Lawn, 22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
Telephone: +44 (0) 1635 565200 Email: info@microfocus.com