

Agility in reality



Online bank Egg has adopted an agile development methodology. Technology leader **Adrian Roberts** explains what this means for testing and testers

Egg is the world's largest pure online bank, with 3.5m customers and 2,500 employees.

Egg has a huge appetite for change. Its business success depends on the ability to implement technical change quickly and reliably enough to gain first mover advantage in the financial services market. This change is delivered directly to the customer through the bank's "shop window" – www.egg.com. There are no branches. If the web site is down, the bank might as well be closed for business.

The great Egg race

An average of 50 code changes are made to the live site each week, whilst maintaining 98% availability on a 24x7 basis, including through peak periods where up to 15,000 people an hour will log on to look after their money. Many changes are implemented during the night, when traffic is at its lowest.

These are not simple web content changes. Around half will involve some modification to the core bank IT infrastructure, such as the servers that support the web site or call centre. The remainder comprises modifications to the software that delivers functionality on the web site or to the operational business, through a complex, multi-tier application architecture.

The dilemma

That's the context for implementing change at Egg. So how would you take it on?

Conventional wisdom says that the rate of change should be slowed in order to improve the chance of keeping web site availability high. At Egg this is not an option as technical change is the very fuel that fires delivery of value to the business.

Equally, one might assume that corners must be cut to deliver change at such volume. But again, quality is the cornerstone of Egg's requirement for high live service availability, so the corner-cutting scissors are well and truly back in the drawer.

The Egg approach - EQDS

Egg describes its approach to technical change: *enabling quality delivery at speed*. It has to happen quickly, and it has to be quality: "customer strength quality". That phrase

resonates throughout Egg, and refers to the unforgiving market in which the Egg operates. If customers cannot trust us with their money, they will simply bank elsewhere. Egg sees the quality of the customer experience delivered as paramount to its relationship with customers and the way they interact with the bank.

The move to agile

At the centre of Egg's change community is a 40-strong system testing team, together with regression and stress test functions. During early 2003, the community experimented with agile software development as a means of delivering maximum value to the business in the shortest possible time.

Although a number of agile projects showed encouraging early results, the business remained unconvinced at the time that this approach was "ready for the masses". The "accepted" approach across the organisation was broadly waterfall, with the development life-cycle broken into clearly defined stages and everyone in their comfort zone: from a tester's point of view, V-Model Heaven.

Only it wasn't really working, if anyone actually admitted it. Spending months on analysis and design, followed by several more months of development, before a product was thrown over the wall to a testing team that ripped it to pieces was clearly no way to deliver business value in any organisation, least of all one like Egg.

The opportunity

And fortunately Egg did recognise this, and was prepared to take on the challenge of doing something about it. In Spring 2003, the heart of the technical change community was relocated from Dudley 60 miles north to Derby, in order that it could work physically alongside and co-operate more closely with business staff on change projects.

The move provided the catalyst for a massive organisational change within Egg Technology, out of which came a true partnership with its business customer. Encouraged by the early successes of the pilot projects, agile was established as the development methodology of choice. The new physical

proximity would mean that many aspects of agile development that had previously been constrained could now be put to full effect. We would be with the customer, we would develop quality solutions together. An exciting prospect indeed.

Exciting enough to enable Egg CIO Tom Ilube to capture the imaginations of senior business leaders within Egg, and to get them on board. For this sort of change to work on a grand scale, you need massive permissions from your business customer, and Tom's stand behind agile was a key factor in securing such permission.

Early reservations

That's not to say that the business – or indeed people within Egg Technology itself – had no reservations about the move to agile. Quite the opposite. Some people believed agile to be a passing fad, and that "normal business" would resume shortly. Others were concerned about the lack of formal specifications, not being sure what we would actually deliver.

For many of us, it was simply a matter of being out of the comfort zone and taking a leap into the unknown. Nobody really knew whether this was going to come off. Not many of us actually knew anything at all about agile methodologies, we just liked the sound of the idea and thought it made sense. We all had the books, but the reality of moving to agile was more like an exotic holiday discussion: "We're going agile." "Oh yes, we thought of going there. Where is it? What's it like? What sort of clothes do you have to take?"

Lack of supporting evidence

It wasn't like there was a wealth of reference cases to support the scale of the change we were making either. It was more a leap of faith, taken in the belief that we could and would pull this off, and that there would be a real breakthrough in the way we worked with the business.

What case studies did exist extolled the virtues of using agile for smaller projects. But Egg Technology was betting the farm on it.



DELIVER ONLY SUPERIOR CUSTOMER EXPERIENCES.

**Want to ensure
a superior
customer experience?**

You can with TestView—the test solution that delivers a quality process for Web application testing. Only TestView provides functional testing, load testing, test management, and seamless JavaScript sharing between products in a single suite. Plus, TestView lets you perform tests on any mix of platforms, and it comes with a comprehensive range of analytics.

Try TestView today—you'll be impressed by its ease-of-use, openness, affordability, and great value!



Attend our Web conference series, “5 Innovative Ways to Implement a Quality Process for Web Application Testing” and get a **FREE T-shirt**.

Register at www.radview.com/superior



The Smart Choice in Web Application Testing

© RadView Software, Limited. All rights reserved.

On the road

On the test conference circuit at the time, I found that hardly anyone was talking agile. What discussion existed again seemed to focus on small scale projects. Fortunately, that has changed over the last couple of years, with agile gurus such as Kent Beck speaking regularly.

There is a commonly held view that agile is suitable for certain (small and low-risk) organisations and projects but cannot work in high-integrity, environments such as banks". hearing this expressed at conferences made me and my colleagues all the more determined to make agile a success at Egg.

Where testers fit in

The adoption of agile at Egg has resulted in a fundamental change in role for the people working in system testing.

- the "Tester" role as it existed has been replaced by an enhanced role of "QA Engineer"
- QAEs are an integral part of a multi-skilled development team of 10-12 people, led by a project manager and an Engineering Lead
- they line directly to the Engineering Lead in the development team, and no longer to a Head of Testing
- they are accountable for performing a quality assurance role on the test-driven development work put together by the developers, rather than being seen as the "person in the team that does the testing". This test-driven development will be described in detail in the next issue of *Professional Tester*
- Wherever possible, only a single QAE exists within the team, *because* this makes it unrealistic to depend upon the QAE to do all the testing work. The situation requires the development team to co-operate fully with the QAE and to take on responsibility for quality assurance themselves if they are to support the Engineering Lead in their collective accountability for delivering quality code
- QAEs take accountability for ensuring that quality is seen as the responsibility of every member within the team. They are effectively "quality evangelists"
- QAEs work closely with "The Truth" (a business customer, fully empowered to make decisions on the functionality that must be implemented) to develop user stories and the acceptance tests to let the developers know when they are "done".

Line management madness

Many objections have been raised (by people who haven't tried it) to this way of working. Here are three of the most common, with my responses based on real experience.

"The QA Engineer reporting directly to the Engineering Lead rather than an "independent" testing manager won't work because the Engineering Lead and Project Manager will exert pressure on the QAE to compromise quality in favour of deadlines"

No. Engineering Leads have ultimate accountability for the quality of the code leaving their team, and they depend on the QAE to support them – even if at times that comes into direct conflict with the Project Manager's accountability to meet deadlines.

Surely, over time the QA Engineer will just "go native"?

In my opinion, using the term "going native" reinforces the "us and them" attitude that characterises the worst of organisational relationships between developers and testers. What we are trying to achieve here is the full integration of quality assurance within the team. That can only happen if the QAE *does* go native. If they haven't gone native, ask why not.

If the QAEs don't actually do all of the testing themselves, how can you trust the developers to be able to test their own code?

Anyone who asks this question has failed fundamentally to understand the role of the QAE – to extend the ownership of quality assurance to every member of that team, whoever they are, whatever role they perform.

Test infect me

At Egg we use the term *test-infection* to describe this extension of ownership and realisation that responsibility for quality assurance belongs to all. Sure, the QA Engineer is going to need some support to get the whole team "test-infected", and at Egg this has been provided in several ways:

- QAEs' mentoring of other members of the team around quality assurance activities
- formal training in testing techniques for developers within the teams
- formal training in use of automated test tools for developers within the teams
- tactical deployment of QA "coaches" within the development team to support the QAE in extending the ownership of quality assurance to all team members.

Trust in me

Many people seem to worry not only that developers won't be capable of testing their code, but that they can't be trusted not to fail to do it, or do it properly, on purpose! Again, this demonstrates an "us and them" mentality, or even a fear over demarcation. I believe that if the culture is nurtured correctly, the accountabilities are clear, and the support is there, this is not a problem.

Let's be honest. Where is the most effective place to identify and eliminate a defect? I would say in the developer's mind, before they write a line of code. If they feel accountability for the quality of the code they write, and they are testing it for themselves, there is a clear incentive for them to write quality code from the outset.

The mindset myth

There is a widespread theory that some quality in a tester's mindset makes them better qualified to test software than the developers that write the code. I am not referring to the "second pair of eyes" argument (which is destroyed by agile techniques such as pair programming), but the view that there is just something unique about the way testers see the world. I believe that the success of the agile approach to development at Egg Technology is clear evidence that no such special something exists.

Find out more about agile

If you are not familiar with agile development methodologies, you may find the following websites useful. No "pure" agile development methodology is used at Egg, but many of the core agile ideas such as pair programming, continuous integration, and stand-up meetings are.

agilemanifesto.org
www.exoftware.com
www.agilealliance.com
www.threeriversinstitute.org
www.martinfowler.com

Do you agree with Adrian's views on the changing relationship between developers and testers? Questions (which we will ask Adrian to answer) and opinions, positive or negative, are welcomed, as always, at editor@professionaltester.com.

In the next issue: Adrian looks at how agile changes the actual transition to live for technical changes at Egg, and explains some examples of how use of an agile approach has delivered real business value more effectively, including the story of the very successful recently-completed Egg Money Manager project.

PT