Speed vs quality: Ten10 panel discussion highlights

Visit **ten10.com**

Ten10

Our panellists

This report is Ten10's recap from the 'Speed vs Quality: Finding the right balance' panel discussion hosted on 27th June 2024. Ten10 hosted five industry experts to discuss the importance of speed vs quality when implementing change, how definitions of 'speed' and 'quality' can differ, and how to approach stakeholders conversations where balancing speed and quality is essential.

Our panellists were:

Stuart Day, Head of Quality Engineering

Stuart is the Head of Quality Engineering at Capital One. He has over 25 years of experience in the quality, testing and general tech industry, working in multiple different sectors throughout his career such as retail, travel, publishing and financial services. He is also a certified coach, international speaker, blogger and podcast host.

Emma Hargreaves, Ten10 Managing Principal Consultant

Emma has worked in IT for 20 years and in test and quality services for 15 years. Emma specialises in quality and test strategies and supports our teams to deliver our services to many Ten10 clients.

Robbie Falck, Senior QA Lead

Robbie is based in London and has worked in QA for many years. He is currently working with an amazing team at Moneybox, a FinTech start-up helping people build personal wealth and savings.

Mala Benn, Engineering Manager

Mala has worked in Software teams for over two decades, transitioning from developer to various leadership positions. Her passion lies in understanding and leveraging the drivers of success: leadership, motivation, personality, culture, progression, and values. Mala's mission is to continuously learn and empower individuals and teams to achieve their fullest potential.

Vernon Richards, Senior Expert Quality Engineer

Vernon is a Senior Expert Quality Engineer at a health tech company and runs Abode of Quality in his spare time. With a background in testing since 2002, including video games on major platforms, Vernon brings extensive experience to his role.

What does quality mean to you in your everyday role?

Robbie: I've always worked for B2C companies and for the past five or so years I've worked for app-based companies. It's very much similar to what you said - happy customers. But I tend to not think about it with a QA hat on.

I just think about: Would I actually use this product? Would I get information that I need?

I guess specifically to Moneybox, we try to educate people on a better savings and wealth journey so it's really important that we have products that people actually understand. And if they're going to give us our money, we need to know what's going on and make sure that they feel comfortable, and we're the right person to do that.

Emma: I'm a tester but I'm also a consultant working on different projects. Every project is different. Every project's priorities are different. What quality means to one person or one stakeholder is completely different to what it means to another. So I don't really have a definition, but what my role is as a tester is to understand what people's definitions of quality are and help them to understand where the solution is in terms of their benchmark on that quality.

Mala: I would say that quality is a measure of the degree to which customers expectations are met in a service or an application that you're building. There are lots of dimensions to it and there could be expectations on different levels. It could be, for example, the usability of an application or the maintainability, how secure it is, how performant. Those are actually what you define through conversations with your users or your stakeholders or the requesters of the application.

Vernon: There's a chap called Jerry Weinberg who has a definition of quality that I like which is "Quality is value to some person." I really like that definition because it always keeps at the front of my mind 'who are you talking about, exactly, and what do they care about?' When it comes to the workplace, the thing that I like to bring up is, as a business, what problem are you trying to solve and for who? Because that will guide everything that you do within your teams and the business. Whenever I see the divergence from those things, that's when I start to get a little bit more vocal. from little bit more vocal.

Stuart Day: The subjectivity of 'quality'

Quality is a subjective thing. It will mean something different about value to somebody at different times.

That can change as well. One minute, you can be using your phone for one thing, and then you think 'actually, I think this other feature will be really useful. Maybe I need to get a different one.' Your priorities and your value and everything changes. So it is very much subjective. It is trying to understand 'who are your customers?' But also there's layers to that. It's not just the end product that is shipped out, but how you're working, the quality, the communication to enable you to do those things.

It's not just the quality of the the output. It's all the pieces that take you to that point that ultimately help you achieve that.

When we're talking about speed versus quality, what does speed actually mean?

Emma: Every project I've ever worked on, if you spoke to the project managers and the people in charge of the budgets, they want you to do the work faster, more cheaply, and retain the quality (at least, if not have higher quality). So they want everything.

In order to speed up genuinely and reduce costs, you have to get the quality right throughout the process.

That's not just testing thoroughly at the end and making sure you know what the bugs are and where the risk areas are, but it's building quality in so that everybody's life is so much easier throughout the whole process. I think speed and quality, for me, are intrinsically linked because without the quality, everything just takes longer.

Mala: [At] my last role at Sky, we were working on quite a complex code estate, and there was a lot of interlinking and integration into massive back-end services. So it was quite difficult to always get features and releases out to customers. There was always this tension between our stakeholders and engineering as to whether we could get stuff out quicker and develop it while maintaining quality. One thing that we started to get better at is experimentation and releasing smaller pieces of functionality so that we could get it out to customers and be able to test it and gain some insights off the back of it, then decide whether we want to invest time and spend however long we'd need to be able to then productionise our experiments. We found that was a good way of appeasing product and stakeholders while protecting our code base and ensuring that we weren't risking quality from a technical side.

Robbie Falck: The importance of communication

I've worked for start-ups that have moved into scale-ups, and it's been non-profitable startups that have become profitable, and that has a lot of impact on the speed that people want to deliver at.

A good example: Moneybox was, a year ago, not profitable, and we were getting a lot of pressure to increase the speed of things because the speed to market of our products is potentially the difference between becoming profitable and not becoming profitable. Especially the seasonality of things. Tax year end is a massive thing for us, when we make almost all of our money. If we don't have the right products delivered before then, we might have to wait another year.

I've always kind of taken the approach: I don't really mind delivering bugs to production as long as we've thought about them and tried to communicate them and have an idea of 'this the risk area, are you willing to accept it or not?'

For example, we might only test the core scenarios of this feature. The P2s, P3s, and P4s? We're not going to test them. So we're able to say 'you might have some bugs in this area, but we're going to deliver it a lot faster'. That's a very fine line, but the only thing we can really do is communicate that and, at the end of the day, it's up to someone else to make the decision as to whether we go live or not.

How do you start having stakeholder conversations about balancing speed and quality?

Emma: I think essentially it comes down to risk. Protection of production is a massive deal for every organisation.

There's a correlation between the willingness to accept risk and the confidence of being able to fix things if they go wrong.

So (for example) bringing the website down, if you can bring it back up within 10 minutes, that's not catastrophic, especially if you time it at quiet times. You commit mitigate to an extent. A lot of projects that I come across, the projects that they're trying to deliver have been a long time in the making. They've gone through long budgetary approval cycles. Everyone's desperate to get that there, but they kind of know that once this version is live, that's it for a while because the investment is done and they're going to have to wait for more money, more investment, to get the next iteration out. So I think that that drives a lot of nervousness about getting it right.

Mala Benn: Using canary releases

I worked in a team that was focusing on the sales journey, so there was not as much regulation around TV products as they might have been on broadband or mobile. And you could play with those levels of risk. So we did things like canary releases, where we would deliver or launch things to a small percentage of the customer base and be able to then watch and learn and see how things ran.

Robbie: Before we go to the test plan side of things, we put a new step in between which we call a 'balanced approach session'. We basically broke down every QA activity that we do throughout the testing lifecycle. It starts with resource and kind of question of 'who?' But then it's ticket testing, end-to-end, UI automation, API automation.

Instead of seeing it as 'we must do all of these things every time', we change it from 'Activities' to 'Levers'. That means that every activity that we do is a lever that you can pull to change the risk and change the speed.

We work in eight week orbits - it's like an eight week sprint - there's six weeks of development in there and we shape projects to fit into those six weeks, which means we shape our testing to fit into that time too. If people are willing to accept the level of risk that comes with that, if we have to turn all the levers off, for example, then we need to say 'we're doing almost no testing here. You're going to be accepting a lot of risk if we take this approach.' We found that works quite well [and it] makes us have that conversation, think about everything we're doing, and rethink the traditional approaches. We can be flexible in anything, as long as we make the risks clear and you're willing to accept them.

What is the real impact of risk?

Vernon: There are a bunch of different people that risk might apply to:

- There's the people building the product or the service
- There's people consuming that as customers
- There's the business who are trying to monetise that thing and make sure that everyone's okay

So when you start talking about how a particular risk impacts those three different sets of people, there's a constant tension there that you're trying to make sure doesn't skew in one way for too long. But if you can do it intentionally because you've got a clear reason, I think that makes a lot of sense. Really what you're trying to ask is:

What are you worried about the day after we release this thing? What you scared is going to go wrong?

And then they will give you an answer to that. And then you can start talk about 'how can we mitigate that?' And sometimes it is very typical testing activities, but sometimes it isn't. Sometimes it's a canary release - how can we get this in front of a subset of people to get a clue, get some kind of data, and then we can take another decision?

Stuart: Often we'll think about the immediate risk - if this thing breaks, what's the impact to the customer at that moment in time? But then also, there's a ripple effect as well. As an example, working at Capital One, credit cards are our thing. So if somebody's going to pay with their credit card and they're buying a week's shopping and their card doesn't work, it's our fault they can't buy their week's shopping. When you're in a conversation, it might be 'the customer can't purchase'. That's quite simple, but what you don't realise is that person can't buy food for their family for that week. That has a ripple effect. That's just one customer, what if it's two customers? Three customers? Four customers? Then calls go to the call centre.

Robbie: I work for a parking company and if you break people's apps and they can't pay for parking it's amazing how angry people will be. We all would [be]. I can't get out of my car to go do the things that I got here for because I can't pay for it, and if I leave I'm going to get a fine. And the fines are ridiculously expensive. So any app that you build or website, you've just got to continuously think about what the person is actually doing in their day-to-day life and the impacts on that.

Stuart Day: The 'why', 'who', and 'worry'

You're only as quick as your slowest thing, whatever that might be, and your quality is only as good as your lowest quality thing.

There isn't a silver bullet to it, either. You've got to have those conversations. You've got to understand your customers. You've got to understand your technology. What are your limitations? There's no point in just saying we need to go quicker. Slamming your fist on a desk to say 'we need to go quicker'. It's not going to make you go quicker. If anything, it's probably going to make you go slower.

Breaking it down to its simplistic form:

- **Why?** Understanding why we're doing this in the first place.
- **Who's the customer?** Who are we doing this for? What problems are we trying to solve?
- **What are we worried about?** What are we concerned about? What's the impact?

Can speed and risk coexist? And if so, what needs to be in place for that to happen?

Emma: I think the lever analogy is the one that ticked the boxes. You can have anything you want in terms of mitigating risk and doing things quickly. Whether you can have everything you want is another matter, and that's where the levers and the dials come in as to where you focus your time and effort.

Robbie: I don't know where it comes from, but the phrase:

Slow is smooth and smooth is fast.

Like a pit crew. When a car comes into a pit stop, if you look at any one member in a pit crew doing the change, they're not moving frantically or fast, they're moving with purpose. They're doing what they need to do. It's not hectic but the the overall effect, the process and the system together, is incredibly fast. I think a lot of it comes down to discipline and efficiency. A lot of the time, people who want you to go faster - they want to cut corners to go faster and that slows you down because you've got this group of disorganised people going as fast as they can but they don't know where they're going. They don't know what they're doing, so you don't deliver anything.

Vernon Richards: Make their alignment explicit

They are intrinsically linked. One enables the other and I think what underlies it is not necessarily the technology but it's getting alignment. Because not only is quality subjective, but speed is subjective as well. The challenge is to make it explicit. So what do we mean by quality? What do we mean by speed?

How do you convince your stakeholders that going slower, from the outside is actually going to let you go faster?

Stuart: I really like that [pit crew] example in terms of because one thing we didn't talk about was team maturity and giving your teams that time to gel. I think it's about the data that surrounds things. How do you convince people? Data is a great example of doing that. Experimentation - can we just do something small? And often we don't give ourselves time to learn anything, because we just want to go quicker. I always try and approach it in a way of: what can we try that's small? It doesn't have a huge impact on everything else that's going on. Maybe we take a couple of people and we just try something. From the data you get from that, you can kind of go 'okay, if we were to expand this, maybe we'd see a bigger return'.

Emma: To quickly add on the data, I think it's important to benchmark where you are now so when you try doing something different, you can show what actually worked. Working out what data to capture to be meaningful and to show an improvement, that that can be a bit of a bit of a challenge, but being able to make demonstrable progress.

Emma Hargreaves: Understand stakeholder priorities

I think the other thing is understanding, again, what your stakeholder finds important. Is that doing something more cheaply? Is it having a higher velocity? Is it having fewer defects leak into production? What actually causes them pain? What are they worried about? Speak to that, because if you're if they're worried about the money and you're talking about defect leakage, they can't join the dots to put the two things together.

If you're close to a deadline and you find an unexpected issue created by a third party, what should be your priority?

Vernon: It's a classic 'it depends'. I think it depends on the things that we've talked about. It's the risk. What are the consequences of this issue that you have found? Let's get clear on that. Let's figure out what the impacts of that situation are. Then it's a question of what levers can we pull? It's like incident management at that point. You're trying to figure out, what is the nature of the problem and what are the consequences of that thing? When we get clarity on that, then we can make a decision on what to do.

If you can keep a cool head and treat it like an incident, and figure out what is going on, try and get some knowledge and explanation around it that will help you create a list of options. And then, in as calm a way as possible, figure out which is the best option to take and be able to explain it.

Mala: Communication is really important in situations like that. Being able to communicate and over-communicate with your stakeholders to tell them what your plans are, what you're going to do, what you're investigating. And to be able to buy time and get people to back off the developers, so the developers can focus on the thing that's really important.