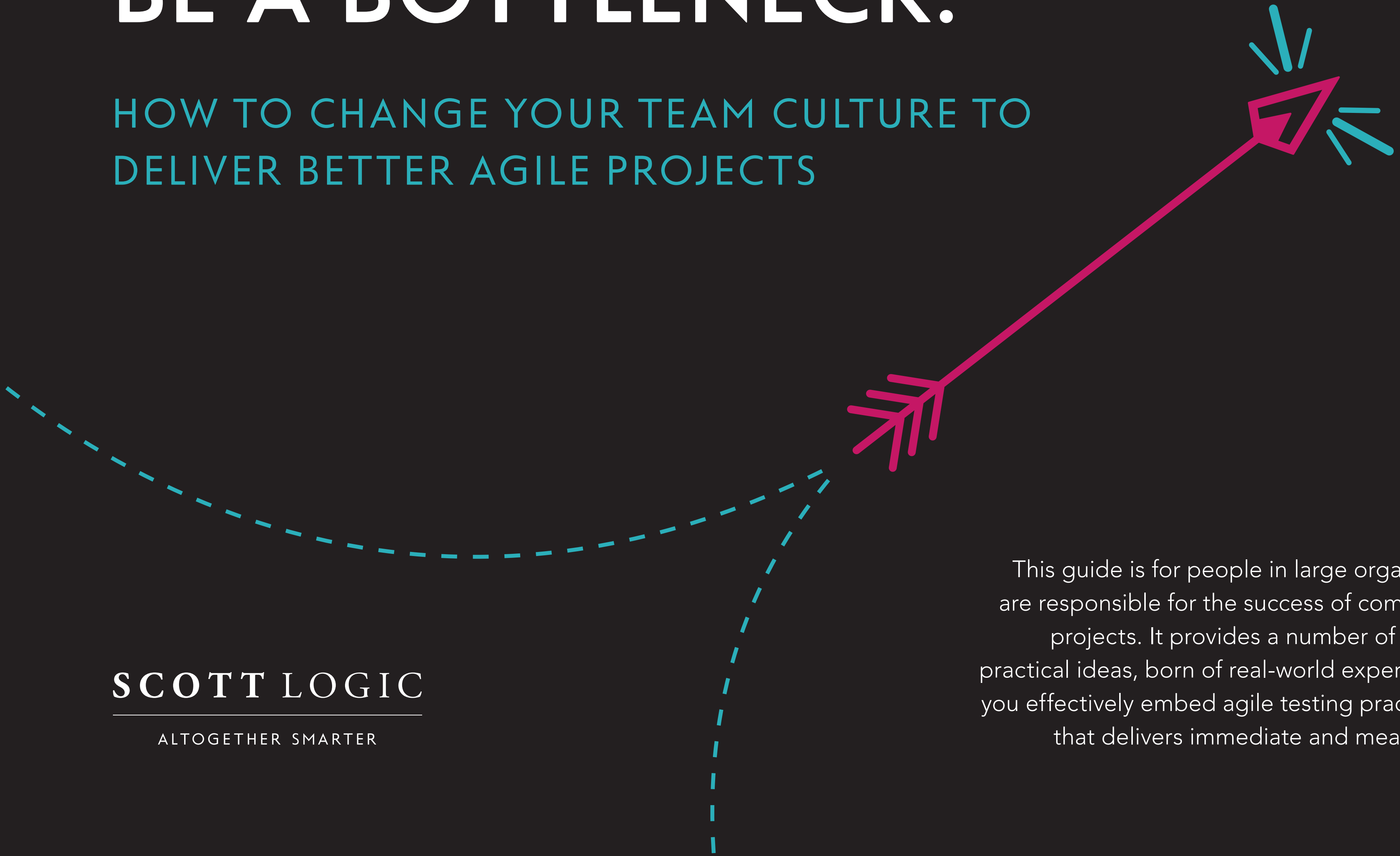
DON'T LET TESTING BE A BOTTLENECK:

HOW TO CHANGE YOUR TEAM CULTURE TO DELIVER BETTER AGILE PROJECTS



This guide is for people in large organisations who are responsible for the success of complex software projects. It provides a number of insightful and practical ideas, born of real-world experience, to help you effectively embed agile testing practices in a way that delivers immediate and measurable value.

SCOTT LOGIC
ALTOGETHER SMARTER

CONTENTS

> "Testing is no longer an **afterthought** or just a **facet** of software development, it's an **essential** part of the **delivery machine** that demands a **proactive approach** and innovative application of both **technique** and **technology**."

Laurence Pisani,
Head of Testing, Scott Logic

INTRODUCTION

If you're responsible for the success of software development projects, you'll be well aware of the benefits of using an agile methodology. Projects which run within an agile framework typically have faster product releases and better customer satisfaction. Teams work more efficiently, and projects comfortably meet deadlines and budget targets.

That's the theory. And yet, projects don't always work out that well in practice. Why do some agile-run projects still suffer from delays, buggy releases, and frustrated teams?

Could the problem be with the technology, or with the business process? Perhaps. But in our experience, the most common reasons for these kinds of problems are to do with the team's unconscious culture, and their approach to testing and agile.

The good news is that once you're aware of the issues, you'll be able to get your teams and projects back on track. This guide will show you how to spot potential problems before they affect your project, and how to reverse them if they're already having a negative impact.

You'll come away with practical ideas and strategies that will help your projects succeed faster and more reliably.

COMMON PROBLEMS WITH AGILE TESTING

When agile-run projects experience problems such as testing bottlenecks and error-prone releases, it's usually a sign that the team isn't working in a truly agile manner. These projects could actually be taking an approach that employs waterfall techniques within the agile delivery.

The first problem with developing in this way is that it can be difficult to spot. A testing team may look like they're working in an agile way – perhaps they're working in sprints, and using Jira. But if the developers are working in isolation for eight days, and then hand it to test at the very end of a sprint, then the reality is that they're still working in a waterfall framework. Agile should be a continuous flow of outputs from both development and test.

Sometimes testing is not fully part of agile development because project leaders aren't aware of the value that testing can bring to agile processes. And sometimes it's deliberately left out, due to well-intentioned but fundamentally flawed fears that agile testing won't be comprehensive, and that it will slow development instead of speeding it up.

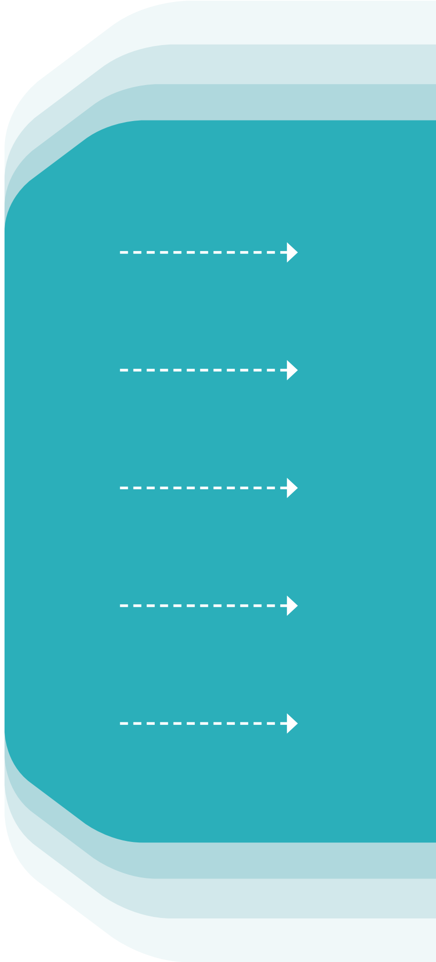
> " Testing is the process of **evaluating** a product by learning about it through **experimentation,** which includes to some degree: **questioning, study, modeling, observation** and **inference.** "
>
> James Bach, one of the initial developers of context-driven testing

AVOIDING THE BOTTLENECK

The fear that agile testing will be incomplete testing seems logical at first glance: how could you possibly test everything if you're working iteratively in short sprints? Agile testing can seem haphazard, but that's not the case.

In a waterfall approach a tester will have a list of requirements, which they use to define test cases. Yet, bugs are regularly found in live systems even after these 'comprehensive' test cases have passed. Little thought is given to how and where applications will fail - the interest is just that it has passed a particular check.

There is a fundamental difference when testing with agility. Testing with agility makes it possible to use modern tooling, gain quick feedback, and execute the same level of checks without relying on heavily documented and time-intensive test cases.



The second fear, that testing will slow development, can be true – in a waterfall approach. Managers who only know testing from waterfall will think of testing as a long process. Testers write a test plan and approach based on the requirements, followed by test suites and cases to meet the coverage demand. It's a very labour intensive process.

Bottlenecks are created when the development and testing are not done in parallel, and then at the end of a lengthy test period the development team are presented with a long list of problems to resolve. This process is repeated and repeated as code changes introduce more, new and different bugs, leading to increased costs and missed delivery dates.

But agile testing takes the opposite approach. Testing with agility produces frequent feedback based on shared models and information, and constant questioning as development is ongoing.

Even simple changes to the development cycle, such as testing on development branches, allow testers to give frequent feedback, which developers can immediately incorporate into their own work to produce effective results.

CHANGING THE CULTURE

Creating an agile culture is one of the most important and least discussed areas of modern software development.

THE AGILE MANIFESTO SUMMARISES AGILE SOFTWARE DEVELOPMENT IN FOUR STATEMENTS:

- ✓ individuals and interactions over processes and tools
- ✓ working software over comprehensive documentation
- ✓ customer collaboration over contract negotiation
- ✓ responding to change over following a plan

As a guiding set of principles they are excellent, but they lead transforming teams to concentrate on the mechanics of working in an agile framework, rather than establishing the necessary culture.

For agile to work, it has to be embraced by everyone in the project – from developers and testers to business analysts and product owners. This can be challenging, especially in hierarchies where management roles have clearly marked areas of responsibility, and must be persuaded to relinquish some control.

5 WAYS TO EMBED TESTING IN AN AGILE CULTURE

1 START WORK IN SPRINT ZERO

The traditional view of testing is that it follows on from development: it's a final quality check before the code is further developed, or released. But this approach is precisely why so many projects suffer from bottlenecks at the quality assurance stage. In a truly agile environment, testers must be active from the very beginning of the project. Testers should be involved in everything from the requirements gathering process and defining acceptance criteria, to testing design elements.

Bringing testing in early gives your project a high level of quality and confidence from the start. If you're working in two-week sprints, you'll immediately see the benefits of starting testing in the first week, instead of in the middle of week two. Bugs will be driven out early in the cycle, smoothing delivery towards the end of the sprint.

2 SIT WITH THE DEVELOPERS

Integrating testers within the development team is one of the simplest ways to embed an agile culture in a team. Being physically close to one another makes communication easier, and it's a simple change that can make a big difference to how well a team works together. That's not to say that teams can't work remotely – just that single testers or testing teams shouldn't work remotely from developers.

3 PARTICIPATE IN CEREMONIES

Testers need to be fully participating members of the agile software team. If the team is running Scrum, for example, then the testers should have input in stand-ups, sprint planning, and backlog grooming. Sizing stories well is an essential part of running an agile project, and testers' experience can help make the process more accurate.
A change which doesn't involve much work for a developer might require a lot of testing, for example, or a large development change may only require a small amount of work for the tester if there is already automation set up for that area.

4 MINIMISE DOCUMENTATION

Avoiding over-documentation is a key element of working in an agile way. Sharing knowledge with minimal documentation speeds up development and creates a culture of action, where people feel more confident in their own abilities and accountable to each other. Testing is a learning journey, and while we want to avoid over-documenting we do need to feel confident that we can share the right knowledge with new testers who may work on the project in the future.

5 MAKE TESTING EVERYONE'S RESPONSIBILITY

Skilful and resourceful testers are an essential part of a software project team, but in agile development testing is not just the responsibility of the testers. A good tester will share their knowledge, and teach developers how to test effectively. If a developer has considered testing as they create code, they'll be asking the right questions of themselves. Developers with good knowledge of testing can find problems much faster, and they also facilitate the process for more sophisticated tests later in the process.

EXAMPLE OF BETTER COMMUNICATION FROM WORKING TOGETHER

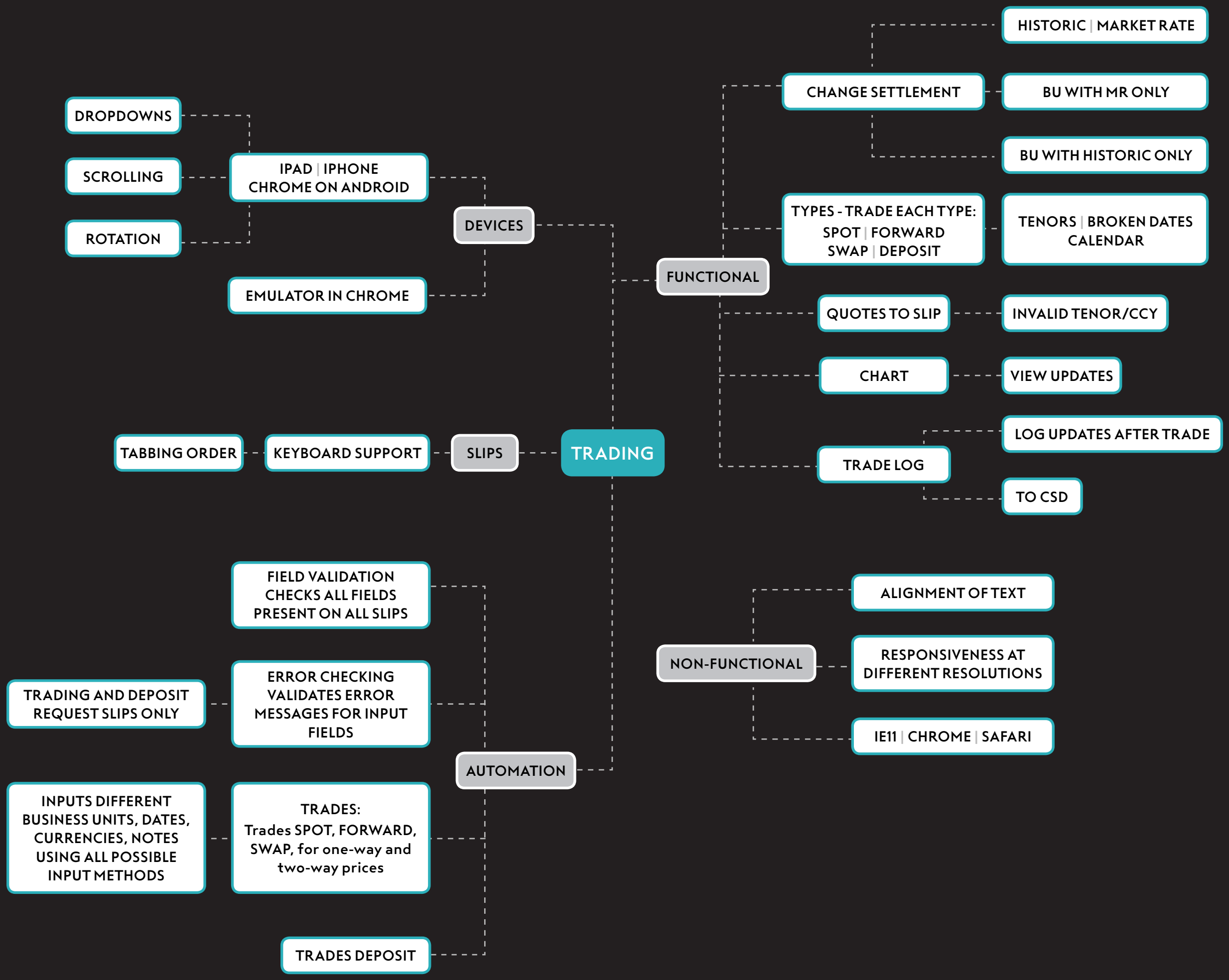
In a team with separate testing and development teams, a developer might pick an item from the backlog, complete the task, identify it as 'done', and notify the tester. But in a team that integrates testers and developers, the developer will be able to have conversations with the tester as they complete the task.

The tester will be able to flag possible problems early, or alternatively instill confidence that the functionality works well on that particular branch. They will have the opportunity to do additional checks as it's integrated with more code, which means that by the time the task is identified as 'done' by the developer it's far closer to its final, fully functional form.

HELPING CLIENTS EMBRACE MINIMAL DOCUMENTATION

At Scott Logic we've found mindmaps to be an excellent aide memoire, and extremely useful alternatives to writing out full test cases. Recording exploratory journeys can provide useful evidence and the ability to replay this execution can offer the added scrutiny needed to spot hidden problems. Where possible, replacing manual test cases with automation will reduce the need for manual testing for regression. Testers can also ensure good unit and integration tests are used and new ones added when code is changed and bugs are fixed. This will help prevent future regression.

Accepting software with minimal documentation can be challenging for some clients, particularly those working in sectors such as finance which heavily rely on reports. At Scott Logic we've found that bringing clients into the agile process is the best way to help them realise that they don't need or want highly detailed documentation. By being involved in the project from the beginning, our clients see the efficacy of our testing, they understand how and why we've tested, and they have the confidence to be able to replicate similar tests if they do any more development in the future.



TRADING

DEVICES
IPAD | IPHONE CHROME ON ANDROID
DROPDOWNS
SCROLLING
ROTATION
EMULATOR IN CHROME

FUNCTIONAL
CHANGE SETTLEMENT
HISTORIC | MARKET RATE
BU WITH MR ONLY
BU WITH HISTORIC ONLY
TYPES - TRADE EACH TYPE: SPOT | FORWARD SWAP | DEPOSIT
TENORS | BROKEN DATES CALENDAR
QUOTES TO SLIP
INVALID TENOR/CCY
CHART
VIEW UPDATES
TRADE LOG
LOG UPDATES AFTER TRADE
TO CSD

SLIPS
KEYBOARD SUPPORT
TABBING ORDER

AUTOMATION
FIELD VALIDATION CHECKS ALL FIELDS PRESENT ON ALL SLIPS
ERROR CHECKING VALIDATES ERROR MESSAGES FOR INPUT FIELDS
TRADING AND DEPOSIT REQUEST SLIPS ONLY
TRADES: Trades SPOT, FORWARD, SWAP, for one-way and two-way prices
INPUTS DIFFERENT BUSINESS UNITS, DATES, CURRENCIES, NOTES USING ALL POSSIBLE INPUT METHODS
TRADES DEPOSIT

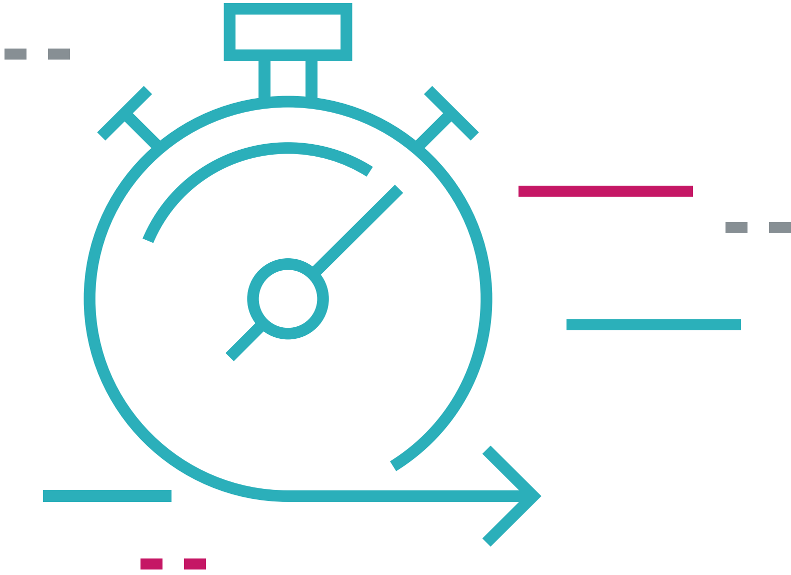
NON-FUNCTIONAL
ALIGNMENT OF TEXT
RESPONSIVENESS AT DIFFERENT RESOLUTIONS
IE11 | CHROME | SAFARI

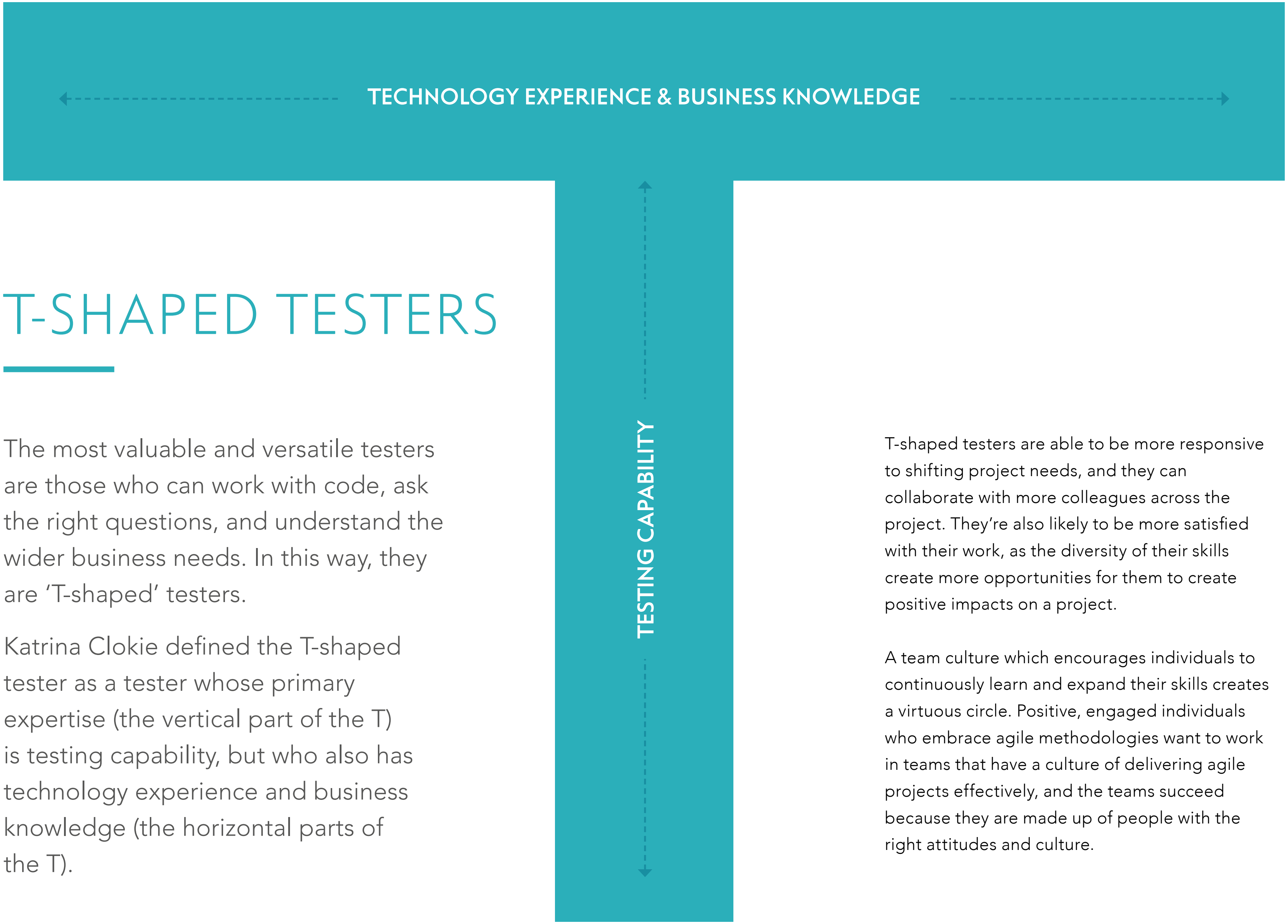
MAINTAINING THE RIGHT CULTURE

Once your project team has adopted the agile mindset, it's important to put steps in place to maintain it. Working within an agile culture means working to a model of continuous improvement, where the team is improving in every sprint. Performance indicators need to be fully transparent, so that everyone is aware of how progress is being monitored, and how success is measured.

Continuous improvement applies as much to an individual's skills as it does to project performance. For testers, this can involve developing their technical skills. Traditionally, and especially in a waterfall framework, testers tended to be very user-focused, with little technical knowledge beyond writing SQL queries to interrogate a database. The tester's main activity would have been to manually identify requirements and write test cases, which would take a long time.

Working in agile delivery, testers need a much greater level of technical knowledge. They can typically read and write the project language. Testers need to understand the code in order to work closely with developers and have meaningful input at all stages of the project. Testing is far more than technical skill, however; it's not enough to be a developer who is just automating checks. The true value of a tester is in asking the right questions of the software and of the team.





T-SHAPED TESTERS

The most valuable and versatile testers are those who can work with code, ask the right questions, and understand the wider business needs. In this way, they are 'T-shaped' testers.

Katrina Clokie defined the T-shaped tester as a tester whose primary expertise (the vertical part of the T) is testing capability, but who also has technology experience and business knowledge (the horizontal parts of the T).

T-shaped testers are able to be more responsive to shifting project needs, and they can collaborate with more colleagues across the project. They're also likely to be more satisfied with their work, as the diversity of their skills create more opportunities for them to create positive impacts on a project.

A team culture which encourages individuals to continuously learn and expand their skills creates a virtuous circle. Positive, engaged individuals who embrace agile methodologies want to work in teams that have a culture of delivering agile projects effectively, and the teams succeed because they are made up of people with the right attitudes and culture.

HAS THIS GUIDE HELPED YOU THINK ABOUT TESTING IN A NEW WAY? LET US KNOW HOW YOU GET ON WITH TRANSFORMING YOUR TEAM AND PROJECTS!

If you need any more advice or support around software testing, feel free to get in touch:

Laurence Pisani

+44 (0) 333 101 0020 | lpisani@scottlogic.com