Inside Track Research Note

in association with

and

Managing Software Exposure

Time to fully embed security into your application lifecycle

Freeform Dynamics, 2018

Introduction

About this Document

The insights presented in this document are derived from an online study completed in the Summer of 2018, during which views were gathered from 183 respondents via an online survey. Participants were predominantly from North America and the UK, and drawn from a range of industry sectors and company sizes. The work was sponsored by Checkmarx, and conducted in collaboration with *The Register* news site.

In the early years of software development, you would often design it, build it, and only then think about how to secure it.

This was arguably fine in the days of monolithic applications and closed networks, when good perimeter-based protection and effective identity and access management would get you a long way towards minimising the risk. In today's highly connected, API-driven application environments, however, any given software component or service can be invoked and potentially abused in so many different ways. Add to this the increasing pace of change through iterative 'DevOps-style' delivery and ever-faster release cycles, and many understandably assert that security management and assurance nowadays needs to be an ongoing and embedded part of the development and delivery process.

But what are the practicalities of this? Do developers – i.e. those writing the code – need to take more responsibility for software security? If so, what do they need to enable them to step up? And whatever you do to bake security into the development and DevOps cycle, how do you do it without killing productivity, and stifling freedom, flexibility and responsiveness, never mind destroying morale?

Many assert that security needs to become a more embedded part of software delivery.

These were some of the questions we explored recently in an online survey of 183 respondents involved in designing, building, delivering and/or securing applications and services. The study was conducted via *The Register* IT news site and threw up some very interesting results.

The changing shape of software delivery

The backdrop for our discussion is a set of trends that is reshaping the nature of software development and delivery. It's no surprise to see confirmation that software is now integral to most business initiatives, and in response to this, we see IT teams increasingly turning to more flexible, responsive, and automation-enabled delivery approaches such as Agile, DevOps and Continuous Delivery (Figure 1).

How much have the following trends and developments caught up with your organisation in relation to software development and delivery activities?

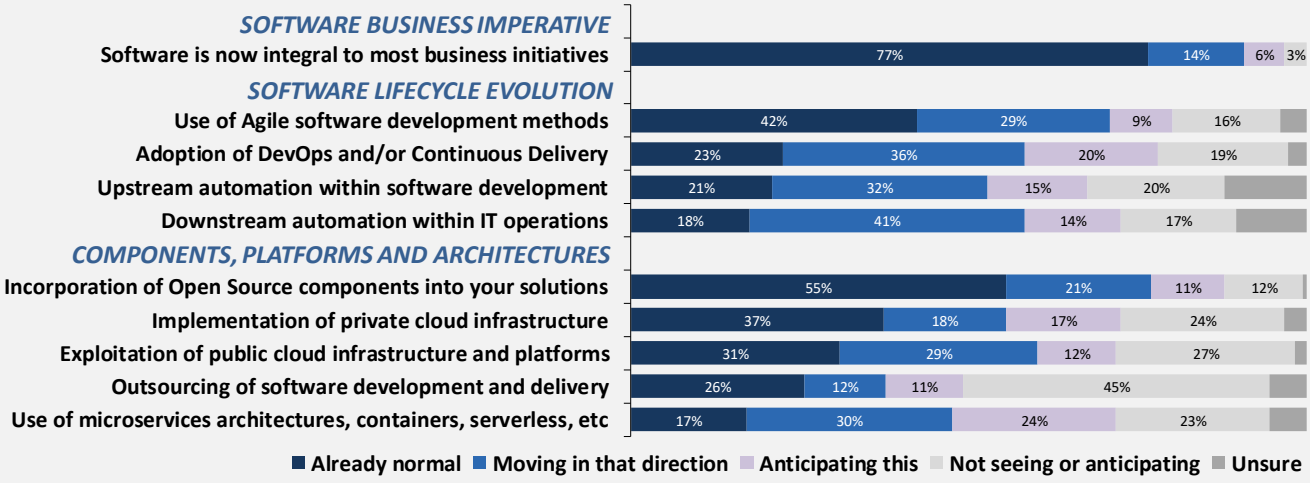

Figure 1 The changing shape of software delivery

Fast, iterative delivery, together with platform and architecture trends, create or accentuate security-related challenges.

Also clear from the above chart are some important trends in relation to platforms, architectures and the origin of code. Most prominent here is the degree to which Open Source Software (OSS) has been embraced by developers, particularly in the form of components that allow quick and easy access to code available from the OSS community. The motivation here is to both speed development and tap into the huge amount of innovation taking place across thousands of Open Source projects.

The trend we see towards more flexible cloud platforms is significant because it allows and encourages more variability between the development, test, staging and production environments. Also, as demands and usage patterns evolve, cloud enables applications or individual components to be easily migrated between platforms over time. Consistent with this flexibility theme is the emerging trend towards microservice architectures, containers and serverless computing.

The software security gap

Fast, iterative delivery, together with platform and architecture trends, create or accentuate security-related challenges. The imperatives are broadly appreciated, but a big gap has emerged between what's needed and what's actually in place (Figure 2).

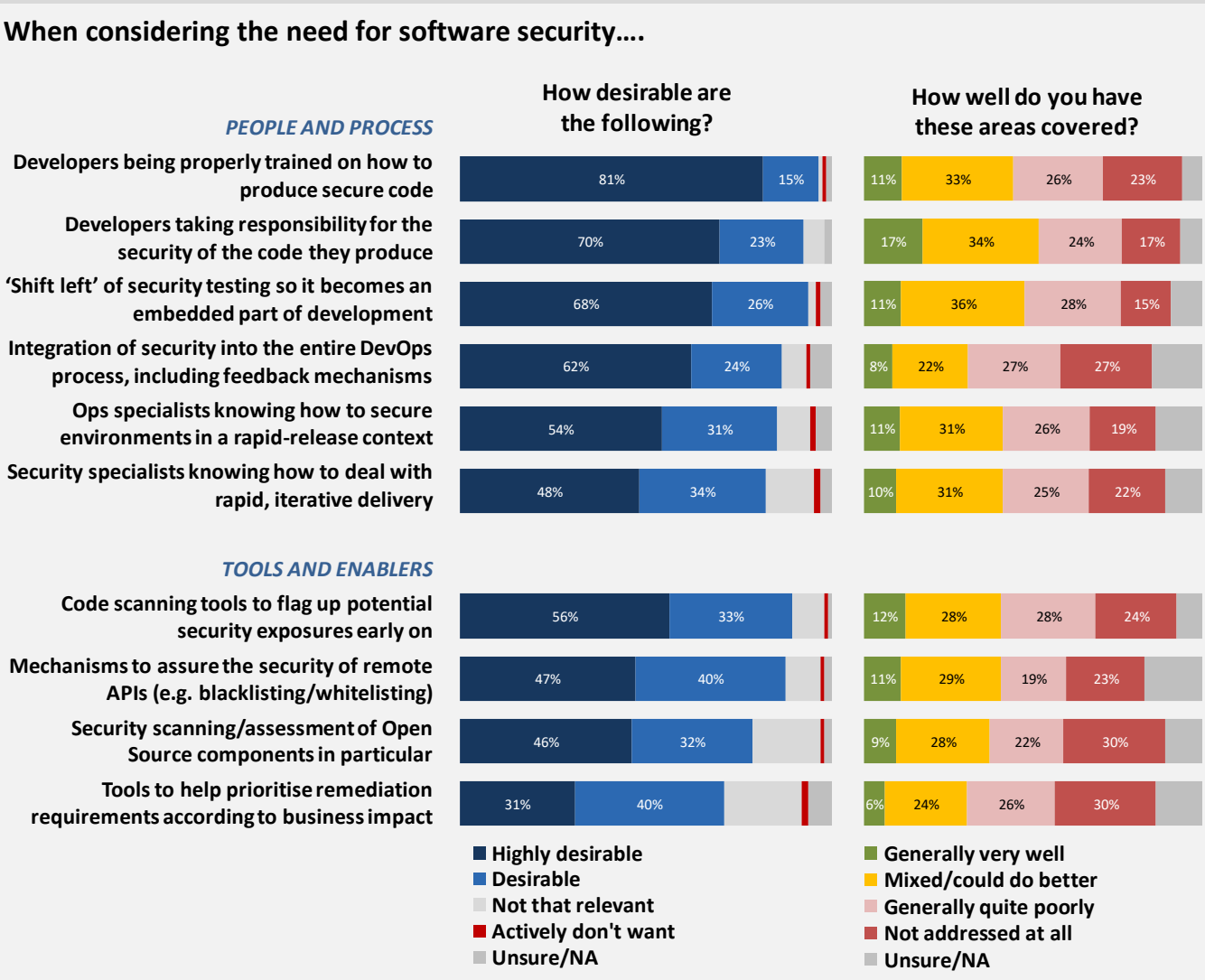

Figure 2 The software security gap

Most of what we see on the above chart is pretty self-explanatory. Zooming out from the detail, though, it's clear that the breadth of the imperatives, with so many items on the list regarded as 'Highly desirable' or 'Desirable', tells us that there are no silver bullets for dealing with today's software security needs. A comprehensive approach is needed to deal with all aspects of the classic people, process and tooling triangle.

It isn't enough for developers to be appropriately 'tooled-up' – they also need to 'step up' and take explicit responsibility for the security implications of the way they work.

On a specific point, the 'shift left' of security testing, along with the need to embed security into the entire lifecycle, shines a spotlight on developers in particular. Ops teams have always carried responsibility for security in many areas, and security specialists and testers, meanwhile, cannot be constantly looking over developers' shoulders to make sure they are doing the right things. With this in mind, it isn't enough for developers to be appropriately 'tooled-up' – they also need to 'step up' and take explicit responsibility for the security implications of the way they work.

Hold that thought while we look at whether any of what we have been discussing actually matters from a performance and outcome perspective.

The development and delivery scorecard

A useful reference for the evolving software security discussion is how well development and delivery teams are performing in relation to key indicators. As with risk management in general, the challenge is always figuring out how best to implement prevention and remediation measures, while at the same time meeting expectations in relation to flexibility, responsiveness and efficiency. Unfortunately, our study suggests that given the gaps we have seen, while some are doing well, others are struggling significantly (Figure 3).

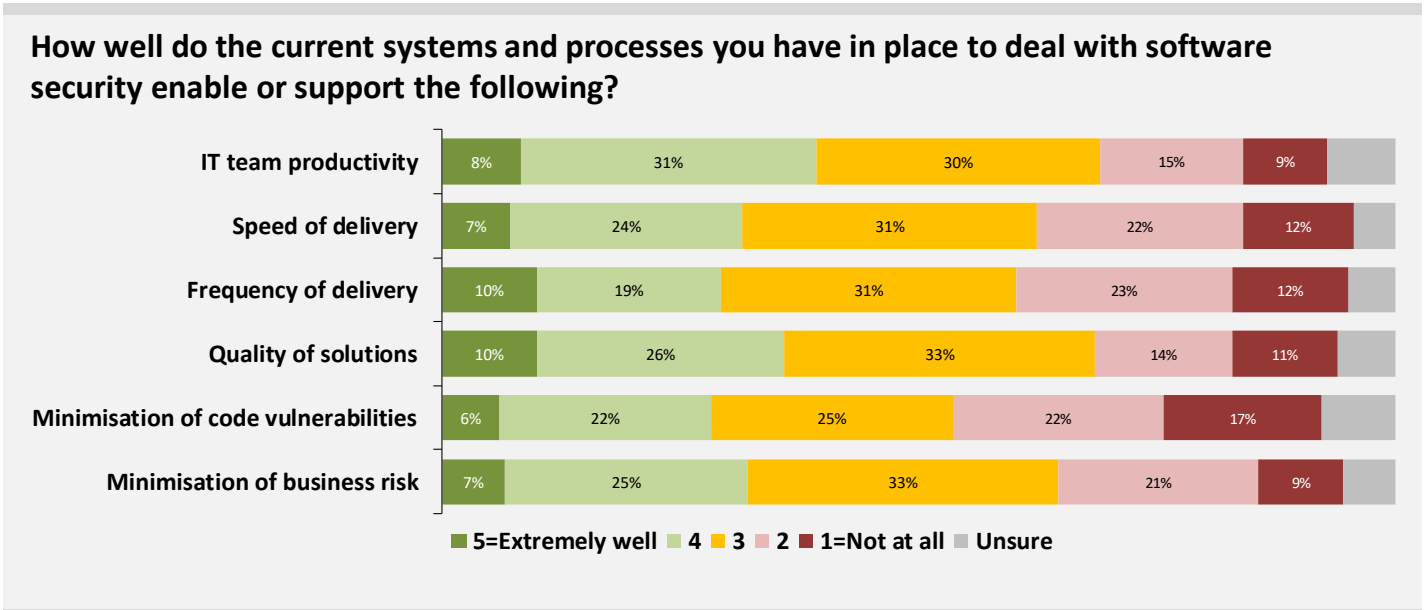

Figure 3 The development and delivery scorecard

To explore the relationship between 'the gap' and outcomes further, we calculated the average score across the above indicators for each study respondent, then divided the sample into two groups. Given the subjective nature of the 1-5 scale used, this is not an exact science, but when we compared 'Top performers' (average score 20 or higher, 37% of respondents) with 'Others' (the remainder), we saw some interesting differences (Figure 4).

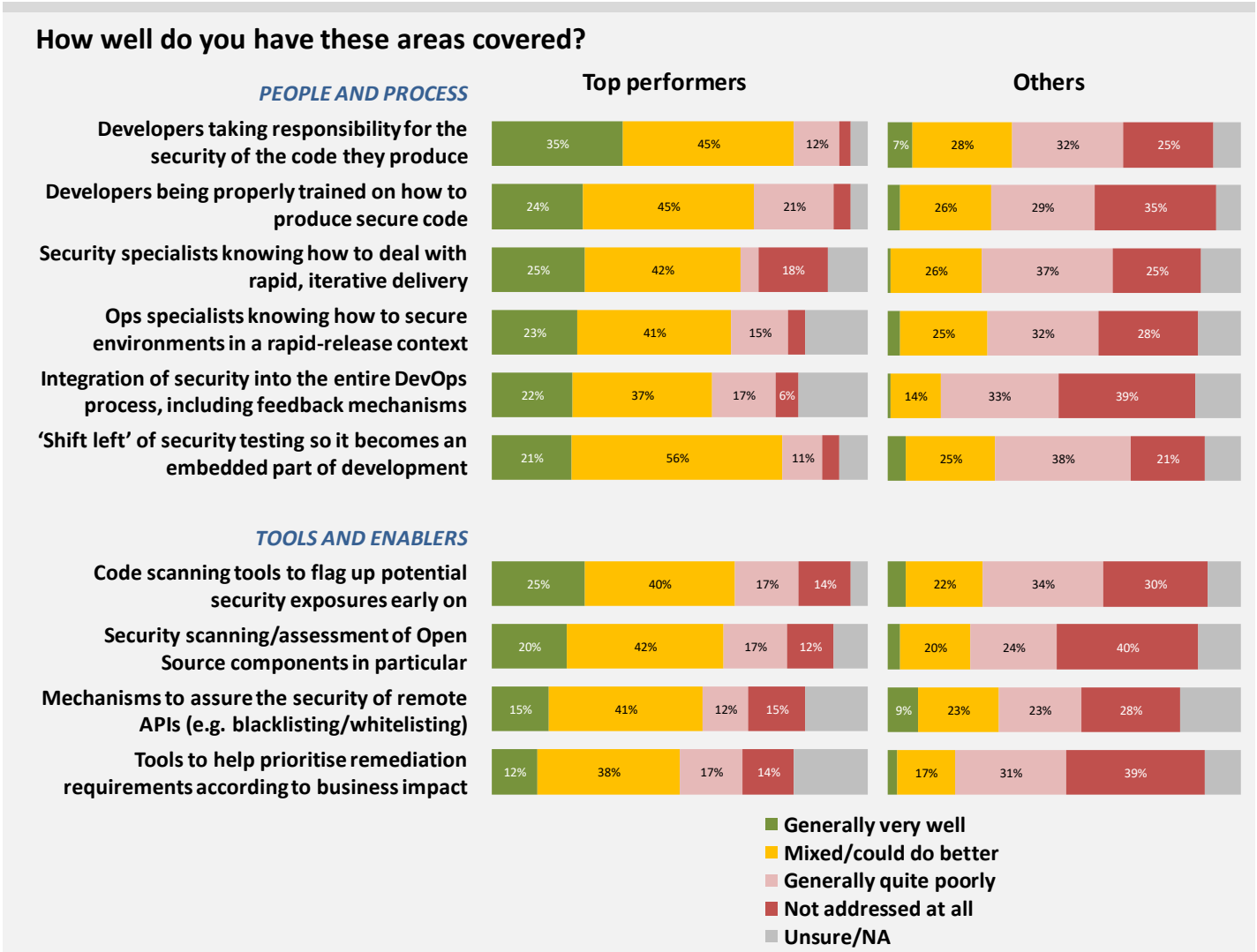

Figure 4 Relationship between capability and performance

The first and most obvious observation is that those in the higher performing group, while often still having work to do, have generally made a lot more progress in dealing with software security imperatives.

The strength of your environment from a people, process and tooling perspective, impacts how well you are likely to be able to keep up with evolving demands.

To really appreciate the magnitude of the difference when looking at the above chart, don't just focus on the green towards the left, but also the amount of red on the right-hand side. This tells us that many in the 'Other' (lower performing) group, have yet to even begin to address some of the key requirements.

The picture overall confirms the relationship between the strength of your environment from a people, process and tooling perspective, and how well you are likely to be able to keep up with evolving demands. This is clearly something you could probably have guessed, but it always focuses the mind when you see these kinds of correlations coming through in survey data.

And coming back to the issue we parked earlier on, it is notable that the most prominent difference between the two groups is in relation to getting developers to take more responsibility for the security of their work. Only then will they be able to play their part in driving improvement.

Making things better

As you begin or continue to strengthen the way you tackle application security needs in the development and delivery cycle, there is overwhelming agreement that it's important to break down traditional barriers between disciplines and get everyone pulling together in a coordinated manner. The suggestion is that this must begin with a rethink of the way application security is approached, and the kind of processes and tools required to make it a more embedded part of the software lifecycle (Figure 5).

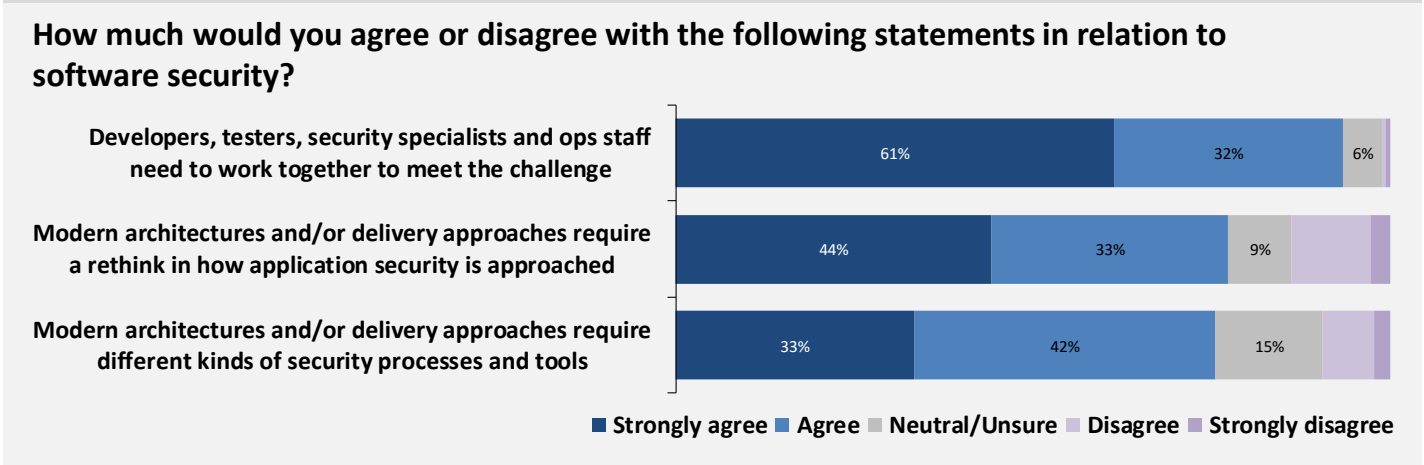

Figure 5 Some important practicalities

As you go through this kind of transformation in your organisation, however, it's important to be prepared for the kind of pitfalls you are likely to run into. Part of this involves tackling the potentially thorny issue of ownership and responsibility, especially against the backdrop of historical trust issues between teams and disciplines. There is then the challenge of getting senior management to understand the imperatives and to allocate the necessary resources and funding required to address them (Figure 6).

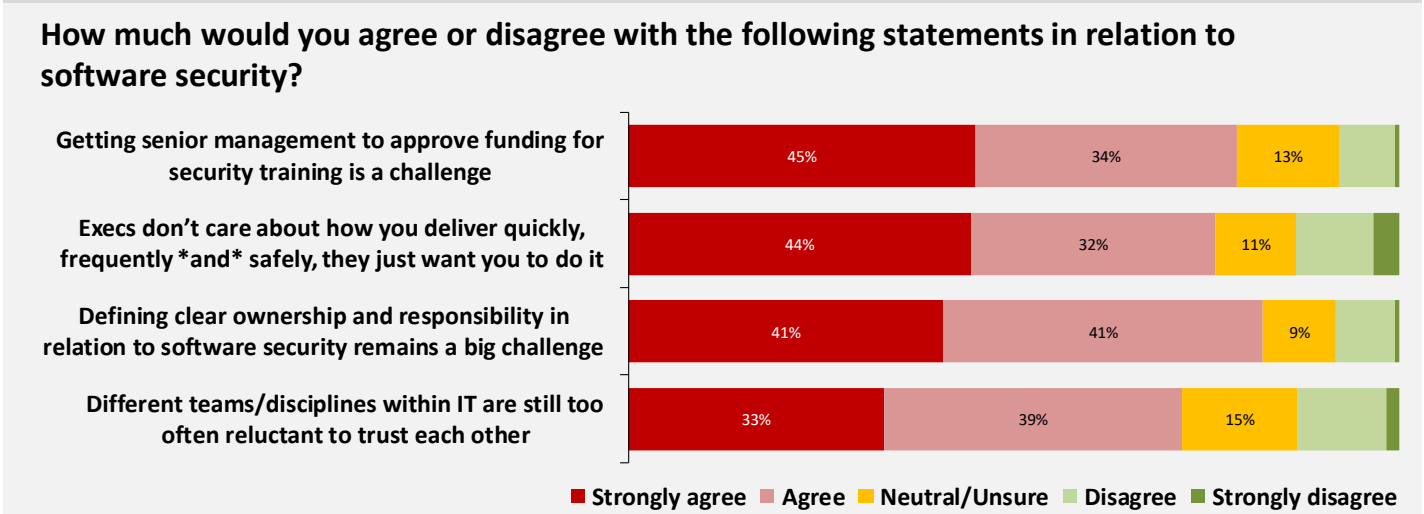

Figure 6 Be ready for the challenges

When it comes to securing or strengthening senior management air cover, the tactics you employ will depend on your situation. There is a general assumption in the

industry nowadays that software security is now a boardroom level discussion, but a significant percentage of those participating in our survey said this wasn't the case, or were at least unsure about it (Figure 7).

AGREE/DISAGREE?

Software security is now a boardroom level discussion

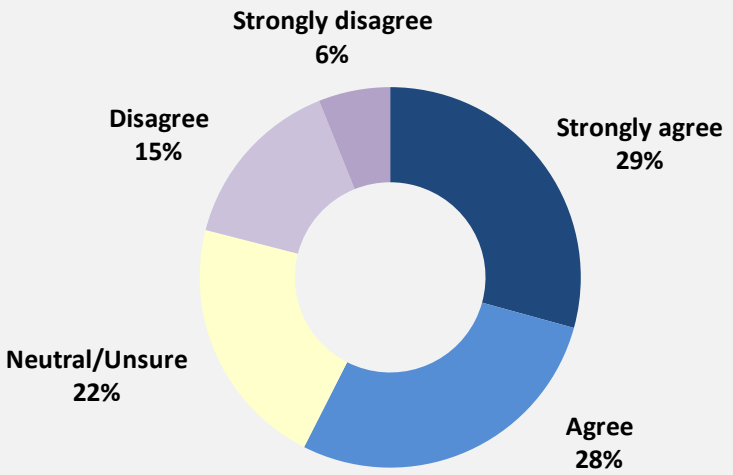

Figure 7 Executive attention

If software security is a boardroom level discussion in your organisation, and executives understand its importance in the context of the business, then you're in a good position. If, however, senior managers are not paying attention or are taking it seriously but failing to appreciate that application security is fundamentally a business risk, then the immediate objective has to be to educate and motivate them. Easier said than done, but without this IT ends up carrying the responsibility, but without the means or authority to do a good job. This is an untenable position.

Final thoughts

We have spoken a lot about the people and organisational aspects of dealing with today's application security needs, but it's important to reiterate that you need to move forward on all fronts together. It's no good, for example, simply telling developers that they are now responsible for security-related risks if you don't give them the knowledge and the tools to step up and manage them effectively.

You have to think of software security first and foremost as a matter of business risk. Everything then becomes a lot easier.

The good news is that modern tools often have best practices and 'intelligence' baked into them that can really help developers embrace security imperatives without becoming overburdened. Whether it's scanning tools to highlight vulnerable segments of code or risky Open Source components, or tools to help prioritise remediation activity according to business impact, the technology available increasingly deals with efficiency as well as risk requirements. And from an automation perspective, enhanced security management tooling will often fit seamlessly into familiar integrated development and continuous integration environments, which further eases their adoption and helps promote changes in work behaviour.

But to reiterate, probably the most important principle to do with software security is you have to think of it first and foremost as a matter of business risk. Conversations about investments, resourcing and even the inevitable trade-offs that need to be considered then become a whole lot easier.

About Freeform Dynamics

Freeform Dynamics is an IT industry analyst firm. Through our research and insights, we aim to help busy IT and business professionals get up to speed on the latest technology developments, and make better-informed investment decisions.

For more information, and access to our library of free research, please visit www.freeformdynamics.com or follow us on Twitter @FreeformCentral.

About The Register

The Register (www.theregister.co.uk) started life as a daily news operation on the web in May 1998. On the first day, 300 readers visited; today over 10 million unique readers visited the site every month. The Register's blend of breaking news, strong personalities, and its accessible online execution, has made it one of the most popular authorities on the IT industry.

With an international team of journalists and columnists, The Register reports on the IT industry from the inside out – covering everything from enterprise software to chip developments.

About Checkmarx

Checkmarx delivers a perfect platform for DevOps and CI environments by redefining security's role in the SDLC while operating at the speed of DevOps. The fast feedback loop makes security testing of new or edited code fragments quick with speedy remediation by developers.

This significantly reduces costs and eliminates the problem of having to deal with many security vulnerabilities close to release. Ultimately, by enabling developers to test their own code for security issues thus allowing them to get instant results and remediate the issues on the spot, everyone wins.

For more information, please see www.checkmarx.com.

Terms of Use