AGILE SOFTWARE SECURITY
WHITEPAPER

Xebia

BECOMING AN AGILE SECURITY OFFICER.

“FROM DEVELOPMENT BLOCKER TO ENABLER.”
Dave van Stein

In a world driven by rapid change, a gap has grown over the years between application security and development. It's a gap that became painfully visible with the introduction of Agile and DevOps. Suddenly, exhaustive, information-laden security policies with checklists and penetration tests became serious impediments. The challenge we face now is to bridge this gap.

Different Roles to Play

The key to success is to split the security officer role into more Agile-minded roles with various responsibilities and duties. You'll need to become truly Agile. In this whitepaper, we will dive deeper into the different aspects of these roles and the differences in responsibilities and duties.

Why You Need to Become Agile

With the increasing use of web applications, cloud solutions, and connected “things” of today, security is defined, for the most part, by development. However, the security aspects are usually the responsibility of another department with its processes. Until a few years ago, this artificial separation was not much of a problem; security was the department of “no” and developers didn’t think too much about security.

But with risk assessments at the beginning of a project and penetration tests at the end, the final result was more or less secure. Vulnerabilities that still ended up in production could be handled by incident response teams. Findings from the yearly audit were given a high priority and were quickly fixed before fines were issued. This was not an ideal situation, but it worked well enough for most companies.

Out with the old, in with the new

With the introduction of Agile and DevOps, this status quo started to shift. Suddenly, release cycles became so short that penetration tests or large assessments were a serious impediment. Since development is directly linked to business value, security was bypassed more often than not, to save time and money.

Many developers were not even aware of the problems, and security specialists were not trained in modern software development practices. So, we ended up with two colliding worlds: certainty versus speed. History shows that speed will always win over certainty. Even when you are wrong while being fast, you‘ll have more time to correct. You win with speed. Assembly lines, workflow software, Lean, Agile, DevOps, and all other speed-improving approaches, always win. The solution is obvious: security has to start understanding how the new world works and find new ways to regain its grip on the situation.

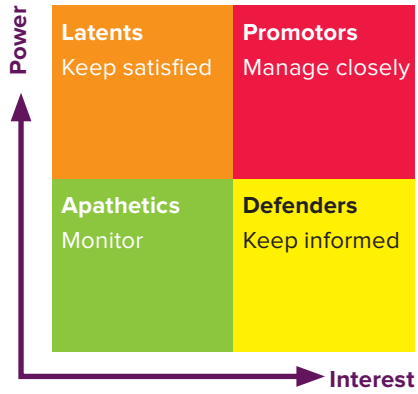
The Security Stakeholder

In the Agile world, the product owner translates business and customer desires into work items and user stories for the teams. The actual desires and requirements are provided by stakeholders, usually representatives of the business and end-users. The product owner needs to consider multiple stakeholders, and one of them should also be the security stakeholder.

Be a promoter

Product owners use a technique called stakeholder mapping, or stakeholder analysis, to identify all the stakes they need to manage. Stakeholder mapping is plotting stakeholders on the power (to influence) and interest (in the product) axis. The outcome helps product owners manage all stakeholders.

Figure 1: Stakeholder mapping



Luckily, security officers are typically in the upper half of this map. (If this is not the case, you need to start working on awareness first!) If you want to become a successful Agile security officer, it is paramount to be a promoter. The most important step for this is a change of mindset. As a security officer, you must switch from a reactive approach towards a proactive one. The only way to achieve this is to start cooperating with the product owners and their teams. Work closely with them, understand their challenges and the decisions they must make, and help them in every way you can to achieve their goals.

You deal in dissatisfiers

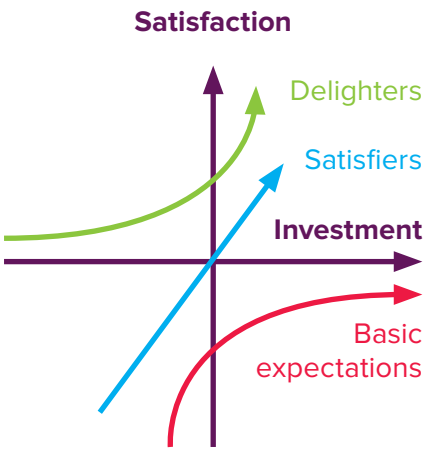
So what type of changes do you bring to the table as a security stakeholder? The KANO model categorizes the different types of changes in three main groups:

- **Delighters:** these are the cool, hip things users didn't even realize they wanted. But once introduced, they can't be missed.
- **Satisfiers:** these are the more traditional changes that improve the product.
- **Basic expectations (or dissatisfiers):** the things that nobody cares about until they're gone.

Unfortunately, as a security stakeholder, you will not have a lot of delighters to bring to the table. In fact, most of your requests will not even be satisfiers. The path you are walking mostly contains the dissatisfiers: the stuff that nobody cares about unless it's not there. On the positive side, there is little discussion about these requirements.

As a security stakeholder, you have to compete with other stakeholders for the most valuable changes. More than ever, it has become important for security officers to translate your requirements into business value. Understanding the types of changes you are requesting is important when creating and assessing user stories.

Figure 2: The KANO model



Help development cut down on security work

The less time teams need to spend on something, the more tasks they can perform and the more efficient they can be. From a security perspective, this means minimizing the impact of policies and requirements. Security officers usually have to deal with

many policies that each need their own proof. However, many of those items do not need to be proven for every software change. Identify which items need to be proven and under which circumstances. This way, the development team only needs to consider the relevant requirements.

Include continuous security in the DoD

Some requirements are always applicable for software development, regardless of the actual change. These items should be placed in the Definition of Done (DoD. In the SCRUM way of working, the DoD is a crucial step, but the actual contents may vary among teams and companies. In essence, the Scrum Alliance covers the DoD pretty well:

> "The Definition of Done (DoD) is a simple list of activities (writing code, coding comments, unit testing, integration testing, release notes, design documents, etc.) that add verifiable/demonstrable value to the product. Focusing on value-added steps allows the team to focus on what must be completed in order to build software while eliminating wasteful activities that only complicate software development efforts."

Figure 3: Requirement positioning

User Story Specific Requirements

Every Sprint Requirements

As a security stakeholder, it is important to understand how the DoD is being used in your team's approach to Agile. Regardless of how it is used, logical items to place there include:

- No open critical and high-risk items identified by security tests.
- All exceptions should be approved by the security stakeholder.
- All applied cryptography should be implemented according to cryptography guidelines.

A common error is placing too many requirements on the DoD. This results in repetitive reviews every sprint, which ultimately leads to ignoring security altogether.

Tell Security Stories

Within Agile, work is usually defined in user stories. These are minimal and defined amounts of work that can typically be finished in one sprint. However, user stories don't magically appear out of thin air. It all begins with so-called epics.

An epic is a high-level wish that describes the desired outcome. An example would be:

> "We want our customers to be able to order products online and have them delivered."

This epic gives an idea of what you want to achieve but leaves options open as to what features and solution can be used. To get from an epic to user stories, you use story mapping. Here you identify the wishes and requirements linked to the epic. Story mapping identifies the major functionalities of the epic and defines the specific user stories. It contains a goal to reach and the applicable requirements for that goal. As a security officer, it is important to become part of this mapping phase as it allows for linking relevant security requirements to the user stories.

In this way, you minimize the work for the teams.

Easy Agile Risk Management

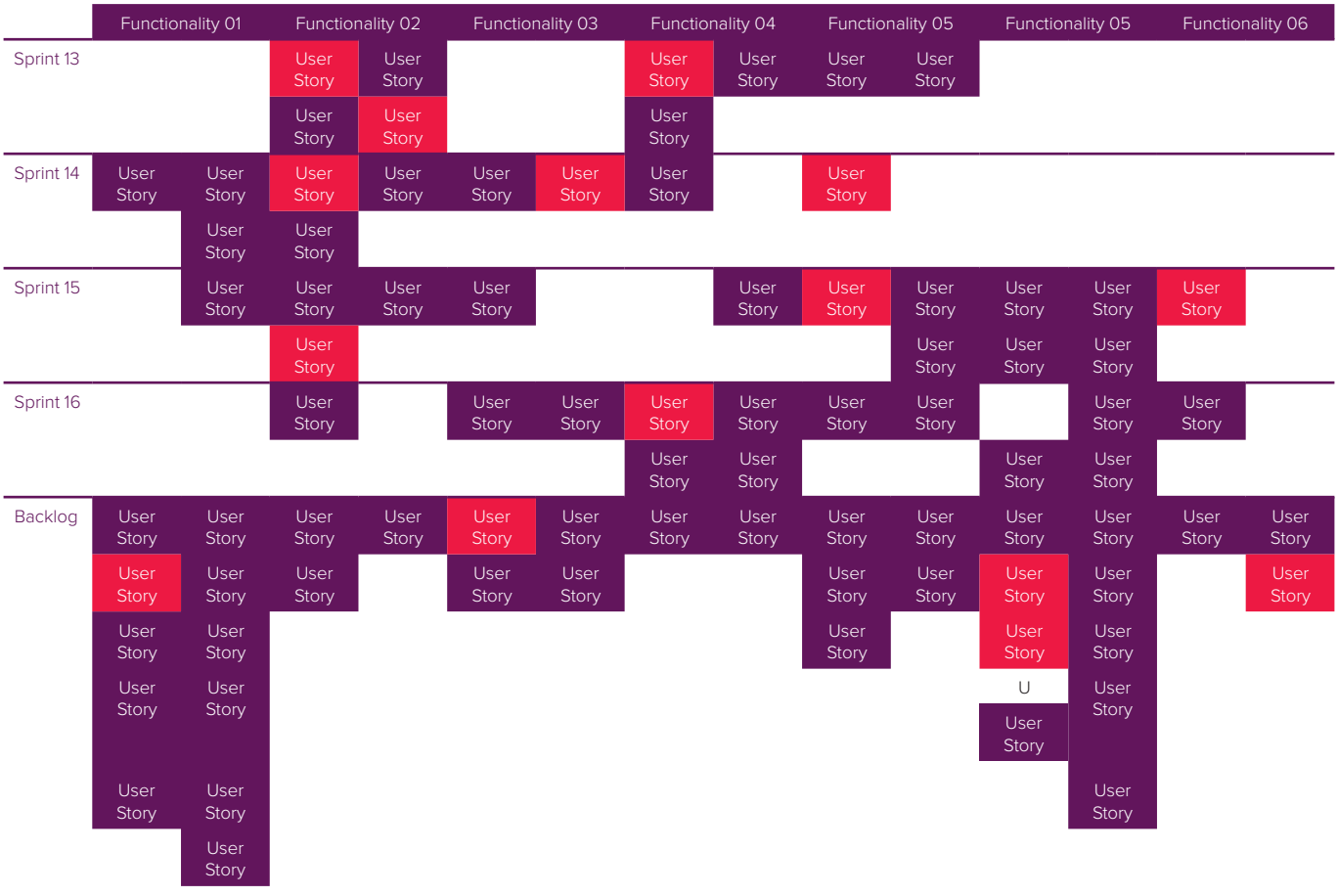
From a security perspective, Agile might introduce a lot of uncertainties and risk. But if the user stories have the right security requirements, this is covered. Three examples:

1. A requirement could be that TLS should protect all communication. The security stakeholder should add this requirement to the relevant user stories. Regardless of which stories are actually selected in the sprint, TLS should be implemented.
2. Another requirement could be that if credit cards are accepted, the system should be compliant with the PCI-DSS standard. As long as this user story for payment is not part of a sprint, the security requirement is not relevant. By specifying this early in the process, a product owner can even decide that accepting credit card payments is not worth the effort. He can opt for another payment method or use a third-party solution.
3. There could also be a requirement that certain data not be allowed to be collected in the online analytics.

As long as the team doesn't implement and enable the tracking, this requirement is not relevant for earlier sprints.

You can identify security and privacy items early only if you are involved as a security stakeholder in the creation of the user stories. This allows you to get these implemented at the right time. By connecting security requirements to the story map, teams can be certain that security won't cause a scope creep.

Figure 4: Risk based story mapping



The Security Expert

As a security expert, you provide active guidance and assistance within the teams. This role requires you to focus on implementing the user story in the right way, discussing the user story itself; the product owner and stakeholders will do that.

You help implement security controls and mechanisms like creating the actual code, advising on encryption protocols, reviewing configurations, executing tests, reviewing cloud solutions, performing risk and privacy assessments or implementing tooling.

As a team member, you must make sure you are there where you're needed. Let people know where you are located, show up at stand-ups and be part of retrospectives of sprints with security features. If you can't be physically present, make sure that quick remote interaction is possible. Be present on communication channels like Slack or Mattermost. At the very least, have a dedicated mailbox for security questions that you give priority to over everything else.

Automate Security

As a security expert, you also focus on continuously reducing the effort it takes to implement security requirements. You should:

- Evaluate safe mechanisms or libraries for the programming languages that are being used.
- Make security checks part of the automated build process.
- Automate the collection of evidence to prove the presence and effectiveness of implemented controls.

Especially in DevOps environments, the ultimate goal should be a security-enabled build pipeline that automatically scans for security-related problems. These pipelines should send alerts for license violations, known vulnerabilities in used software components, and vulnerabilities in the deployed application. Fortunately, many tools can already be combined with most of the popular build automation tools.

Figure 5: Agile Roles

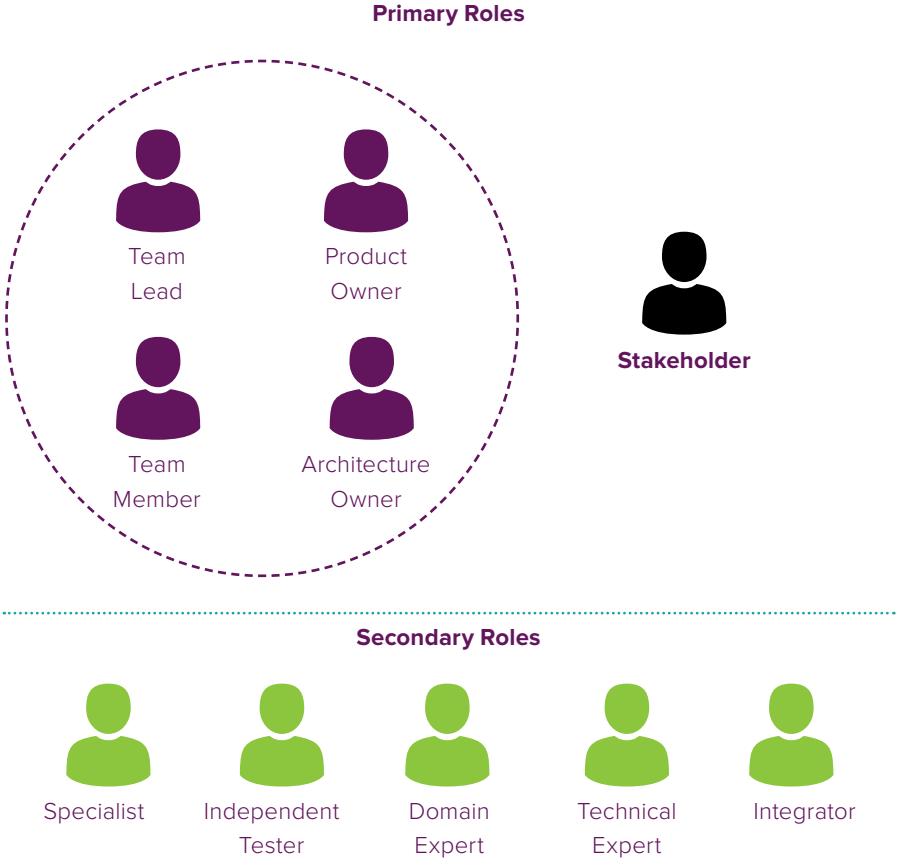
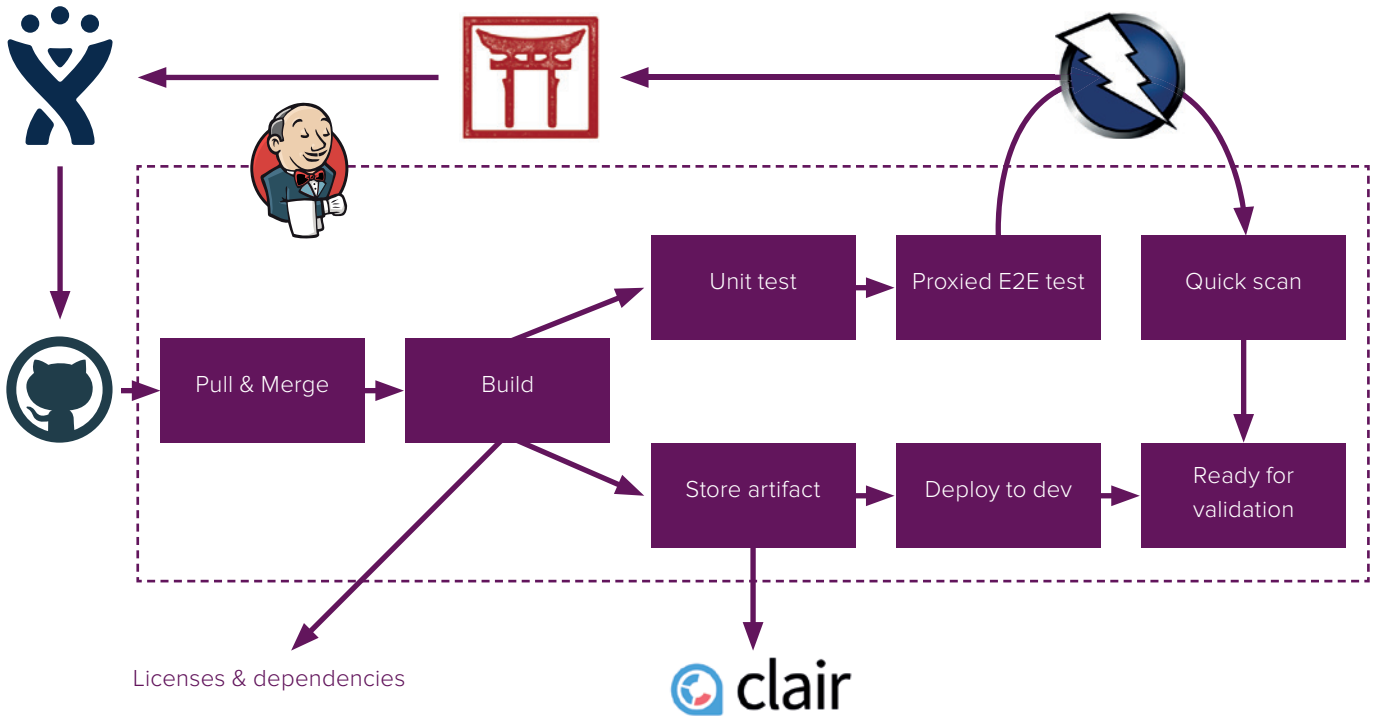


Figure 6: Security Automation



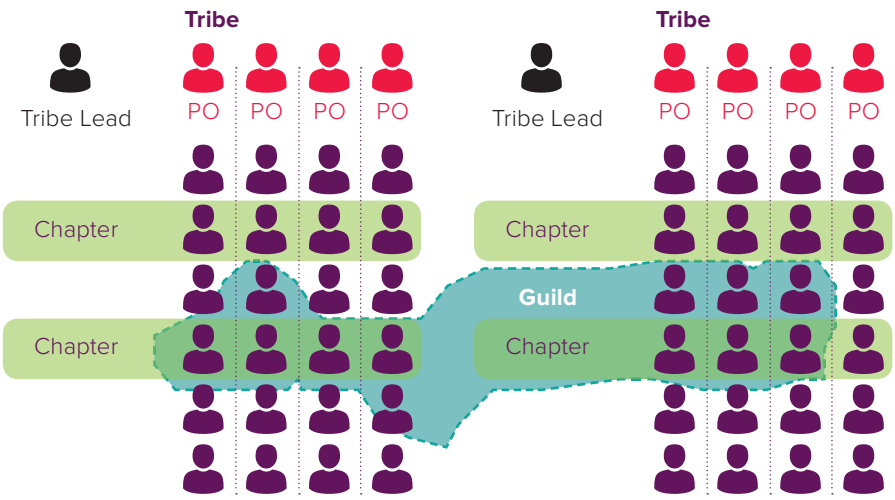
The Security Evangelist: Raising Champions

A last crucial role of Agile security officers is the security evangelist. As a security evangelist, you not only act as a domain expert across all teams but you also coach teams to become more security aware.

One way is to form a special-interest or focus group, or a guild like in the Spotify model. This group acts as an "organic and wide-reaching community that shares knowledge, tools, code, and practices." The security guild's mission is to spread knowledge on best practices, regulations, tooling, and intelligence; and to discuss security and privacy related topics, and to increase awareness and skills across the teams.

The ultimate goal should be to identify potential security champions. These are Agile team members that have shown a good understanding of a specific area of security. They can act as the security voice for a specific product, team or technology. These security champions can represent and offload the security domain expert and do the initial triage to determine whether the security expert gets involved.

Figure 7: The Spotify model



Summary

We started this whitepaper by stating that over the years, a gap has been created between application security and development. It's a gap we created that became painfully visible with the introduction of Agile and DevOps. We presented the way the role of a security officer has to change to bridge this gap. The key to success is to split the security officer role into more Agile-minded roles with different responsibilities and duties:

– The security stakeholder: defining the "what?" and "what not?"
– The security expert: helping with the "how?"
– The evangelist: raising the bar

Integrating and aligning security with the Agile way of working is the only way we can bridge this gap. In this way, we can eventually make security an enabler instead of a blocker.

Customer Study Case

How KPN Bridged the Gap between Security and Innovation

KPN is the largest telecom operator in the Netherlands. Their landscape consisted of multiple back-ends and front-ends, each containing different information about the same customers.

This fragmentation made it difficult to implement changes in the systems due to multiple dependencies. With Xebia's consultation, KPN decided to create a single front end before the multiple backends. Using an Agile approach, KPN gained the flexibility to connect one system at a time and implement continuous improvements to the system.

Initially, this caused friction with the security department. The KPN security policy consists of hundreds of rules that require validation before every production change. In an Agile environment, this validation would require an inordinate amount of time, especially with twenty Scrum teams deploying every other week.

Quicker validation means faster deployment

First, Xebia and KPN analyzed the complete policy and identified the circumstances most likely to trigger a rule. Next, a questionnaire was created, that highlighted relevant controls depending on the answers.

With a little more than twelve questions, the questionnaire acts as a smart filter. It helps identify the relevant controls that require validation. It also makes responsibilities more transparent, allowing KPN to apply risk profiles with clear boundaries.

This approach resulted in a better understanding of security within the teams. It made it easier to implement changes and reduced the time needed for validation. Development and business can confidently apply changes without concerns regarding the security impact.

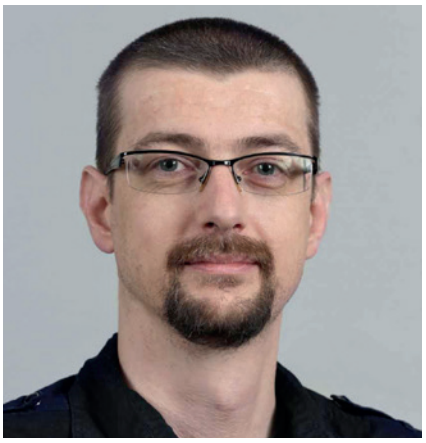
About us

Xebia security offers security services to top companies in the Netherlands that accelerate and innovate by using modern software development methods. We support your road to success by safeguarding your software development processes. We are developers who understand your security needs.

We focus on the top companies that use Agile, DevOps, CI or CD to develop cutting-edge software and applications. We help integrate security into the company culture and modern software development processes. A change in mindset is often required. Besides a “will this work?” perspective, asking “what could possibly go wrong?” is essential to mitigate security risks.

Want to become an Agile security officer?

We offer coaching, public and in-company training in secure development practices. For security officers, we provide all the templates, tools and methodologies you need to do your job in an Agile way.
Our experts understand your challenges and are happy to support you.



Dave van Stein

Dave van Stein is Principal Consultant Security at Xebia and a SecDevOps enthusiast. Over the years, Dave has performed numerous security assessments for customers and has helped clients implement security into the software development life cycle. Dave has several publications and presentations to his name and has certifications for ISEB/ISTQB, C|EH, and GWAPT. Dave currently focuses on helping customers apply the secure and privacy by design approach in Agile and DevOps environments.

Xebia explores and creates new frontiers. Always one step ahead of their customers' needs, Xebia turns new technology trends into advantages. As mainstream frontrunners, they create new solutions and build the future, together with their clients. As passionate technologists and pioneering craftsman, they provide the cutting-edge tools, training and consulting services that make businesses work better, smarter and faster.

For more information, please visit xebia.com