CORE®

SECURITY IN AGILE PROCEDURES

Daniel Augustin, Managing Director, SSE Secure Systems Engineering
Dr. Waldemar Grudzien, Expert Director, CORE SE

30.10.2018
COREtechmonitor | Blogpost

Key Facts

- Agile development "overlooks" holistic safety requirements
- Security Ambassador combines agile development with a holistic view of security
- Measurable security features become quality features such as product features
- Security-by-Design is not a code-freeze, but has to be questioned anew in every sprint
- Prudent security planning intrinsically prepares for auditing

The transformation of software and infrastructure projects

Software and infrastructure projects are subject to fundamental change. The trend today is towards small, agile teams that work independently of each other and often spatially separated from each other. They develop completed subprograms that communicate with each other via coordinated interfaces and thus form an overall system. The agile approach brings with it fundamental advantages that are particularly useful in complex projects. Small teams develop more productively because they harmonize better with each other, choose their own methods and tools and follow their own rhythm. The separation of organic systems into independent subcomponents also offers noticeable added value in terms of maintainability and assessment capabilities. IT security must not break through this circumstance, but must use it for itself.

In a direct comparison, the evolution of software development precedes that of IT security, which to a large extent still operates in the waterfall model. This circumstance is not unfounded, because a holistic view of the product is required for a security analysis. Security experts group data into protection requirement classes, which are derived from protection goals (e.g. data protection, certifiability, product philosophy) of the information to be protected, and then consider all those components and connection paths that interact with them. Vulnerabilities and the resulting forms of attack result in technical, organizational and personnel measures to reduce the probability of occurrence and the amount of damage. A security assessment takes place at all levels of a software and infrastructure project and therefore seems to contradict agile project structures.

Project costs, residual risk and costs for feasibility increase the later the implementation of suitable measures is started. In our experience, the best results are achieved by adding the "Security Ambassador" function to the agile model.

Security Ambassador as ambassador between the worlds

The harmonization of agile projects with a rather monolithic security accompaniment is an exciting challenge. The goal is nothing less than to link two worlds that could hardly be more different. The uncompromising coexistence between waterfall and agility inevitably leads to frustration. Security experts do not feel taken seriously and developers are patronized and slowed down. Communication gets worse and work on the product as a common goal gets out of focus.

The solution is obvious. In order to create a resonance, security experts must become team members. So-called Security Ambassadors act as adapters between

both worlds. They have security expertise, but apply it only within the boundaries of the one agile team to which they have been assigned. With their knowledge of cross-system security contexts, they provide advice, develop process models and accompany security-relevant user stories. As ambassadors, they communicate with the higher-level security authority and contribute their findings to the overall security assessment. Conversely, security ambassadors can at any time draw on the expertise and input of the security ambassadors from other teams and, if necessary, the security experts of the project owners.

Development focused on features

IT security and data security are necessary prerequisites for the successful digitization of business processes. They create trust with customers, partners, the public and not least with supervisory authorities through effective protection against economic damage and damage to reputation. The necessity of a functioning IT security is beyond question and experience has shown that it is not doubted. Nevertheless, their implementation often hangs in the balance.

One reason for this is to be found in the economic view. IT security primarily serves to avoid financial damage and reputation. It costs money instead of earning it. If one consequently compares investment and return, secure solutions generally perform worse - as long as nothing happens.

With functional requirements - the features - things are different. They are tangible and lead to profits. This is also due to the fact that they are understood by the customer as a quality feature and are used in the decision-making process between otherwise equivalent products. From the customer's perspective, security is a difficult to measure and inflationary advertising promise. This circumstance is also reflected in software development. In the individual phases of a project, the agile teams present their deliverables, i.e. functional interim solutions. Their evaluation by stakeholders is based on tangible markers. These are usually features. There is no reason why IT security, as a non-functional requirement, should be a shadowy existence here. It can also be communicated as tangible progress. Penetration tests with no noticeable findings are only one option here. Even linting tools with security audit rules or freely available online analysis tools can attest solutions and interim solutions to a high level of security. Certified IT security is a measurable quality feature that can be transported to the outside world and monetarized as a product feature.

The prerequisites for this are very simple: both the project management and the management must commit themselves clearly to security and treat it as equivalent to features in requirement specifications and acceptances.

Transfer of security knowledge into agile systems

Looking at the OWASP list of the most common vulnerabilities and most successful attacks, stable conditions have been discovered for years. The most common attacks such as Injection, Bruteforce and the usual vulnerabilities such as misconfigurations, missing vulnerability/ patch management, inadequate monitoring or inadequate access controls have been persistently at the top of the charts since 2010.

Identification and evaluation tools exist for most of the vulnerabilities. They are used for penetration tests. This allows developers to close gaps that have been uncovered without the help of security experts. However, observations in large-scale projects have shown that once closed security gaps often reopen in later project phases.

More agile security support offers a remedy here. Linting tools within continuous integration (CI) are used to transfer knowledge about common vulnerabilities. Known vulnerabilities and solution notes can thus be called up independently and without the intervention of security experts and corrections implemented.

The frequency of required penetration tests as well as the effort of internal audits are reduced, which leads to a significant acceleration of the development process.

If the most common threats are covered automatically, then the individual and technology, regulatory or industry-specific IT security challenges can be addressed. These drivers constantly create new attack vectors and new vulnerabilities due to the complexity of an ecosystem that is broken down into its constituent parts and split across multiple players - a simple example: online payment systems and third-party market entry due to PSD II.

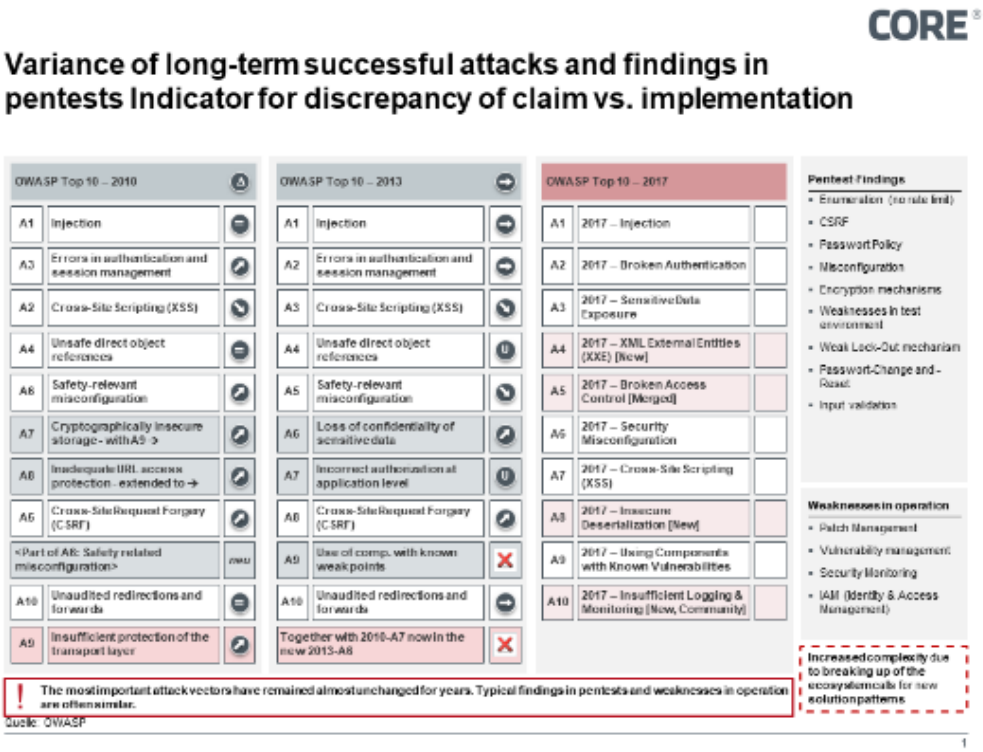

Figure 1: Successful attacks, pentest findings and weaknesses in operations show stable behavior over many years.

Buzzword Security-by-Design

The increase in complexity caused by the breaking up of the ecosystem Security-by-Design is the motto of everyone, not least because it has codified regulation and supervision. However, from the start there is often not much left of it. Due to increasing requirements, system design and architectural design tend to be a "ball of mud" in which everything is still somehow implemented in order to meet the next sprint, the next release. Security issues, documentation and code review are often neglected. Security as a non-functional topic is often only involved temporarily and additionally only later in the advanced project, then a design should be "quickly made secure", because security "only generates effort and earns no money". Security is perceived as a blocker of feature development and therefore often less prioritized.

The peculiarities of modern software development pose further challenges that complicate the appropriate integration of security aspects. Nowadays, projects often consist of globally distributed development teams. They are geographically dispersed across countries and time zones, and their members speak different languages. Not only do the developers not have a common lingua franca, but the individual domain members also speak different "languages" such as business, marketing, legal, security, etc. Both language phenomena lead to communication problems, which ultimately lead to "misunderstood", intrinsically insecure code. The spatial, temporal and linguistic distribution as well as the time corset of the agile procedure require closely timed

agreements which must be kept tight in a disciplined manner. If a crack in the timetable threatens, the non-functional requirements fall by the wayside first. The goal must be to change the operationalization of development by linking codes and security.

Security-by-Design starts with the right questions

The security of a system is not a code-freeze, but an agile process in which a holistic view of system security is achieved by continuously questioning the 5 essential system variables:

- **Model**: Which external interfaces are provided? Is a dedicated or cloud-based infrastructure used? Is it a distributed or a centralized solution? How is the domain tailored to Microservices? Is the preference for eventsourcing or databases? The system model significantly determines the possible
- **Attacks/Vulnerabilities**: How do I protect inputs into my system? What influence do I have on the security of the tools, libraries, and services I use? How do I prevent enumeration, brute force or fingerprinting attacks due to the unintentional disclosure of system internals? Is there a threat of internal perpetrators or priviledge escalation attacks? The identified possible attacks and vulnerabilities require a precisely tailored
- **Measures**: Which technical, organizational or personnel measures Which technical, organizational or personnel measures lead to a reduction in the probability of attacks occurring (e.g. input validation, vulnerability and patch management or hardening of components, network zones)? Are there additional possibilities to reduce the amount of damage caused by attacks (e.g. security monitoring, incident management, hardware security modules)? How can my organization be protected from internal perpetrators (e.g. IAM, Segregation of Duties, security training)? Do I have access management and segregation of duties in core processes? The degrees of freedom of these measures are decisively determined by the used
- **Technologies:** New technologies must be mastered, because poorly implemented technologies are dangerous. This applies to the entire tool chain, the protocols used such as OAUTH 2.0 and OpenID Connect and, of course, the correct parameter setting in libraries and frameworks. Hover above everything
- **Regulation/supervision:** Which regulations and supervisory regimes determine how my system model and the selected technologies and measures? Cloud or non-cloud? Only within the EU? US Homeland Security Act? Which personal data is processed where? Can my system delete individual data fields? Also in backups and archives?

Questions over questions that require concentrated and focused experts in secure agile development from the outset.

Lessons learned

To improve the integration of features and security, there are several lessons learned

in projects of various sizes. These experiences are actually obvious simple measures, but in practice not self-implementable. Security in agile development is a team effort - so simple, so difficult. In the best case, all development teams should work on the project at the same time and in the same place. The same language offers a big advantage. In the second best case, the different domain teams work distributed on the project. Team domain 1 at location 1, team domain 2 at location 2, etc. The decisive factor is that the Security Ambassador belongs to his team. The ambassador must be an equal member of the team.

The close support of the developers through the newly installed function of the Security Ambassador per Business Domain has proven to be very effective, following the example of Extreme Programming. The Security Ambassador acts as an advisor to the developer, solving problems as well and not just making demands and waiting for the task to be completed. This process model means daily hard work and rewards with secure systems. For longer running projects, a team rotation of the Security Ambassador can be considered after a few months in order to prevent a "cronyism effect" with lenient effects.

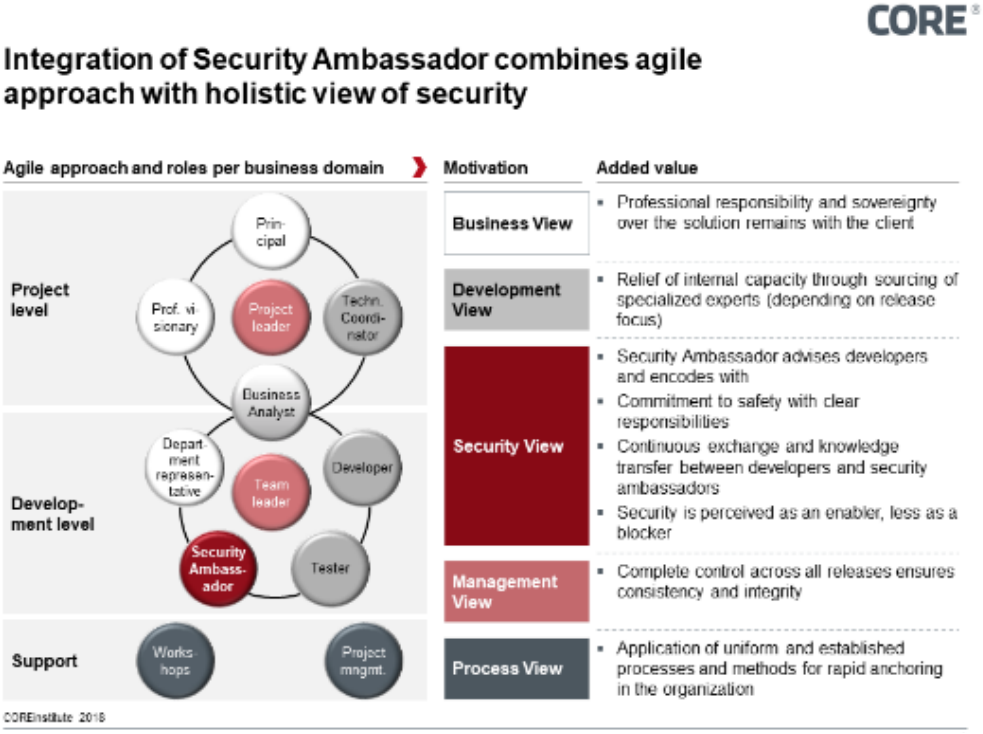

Figure 2: Security Ambassador combines advantages of the agile development model with the precision required for security requirements of the waterfall model

By integrating code analysis tools and sensor technology developed in-house into the deployment chain, developers are at best able to identify security-critical code elements during their development work and heal them in cooperation with the Security Ambassador.

If security-by-design is taken into account in the project right from the start, a shift of responsibility for security into architecture and product design takes place. In the design phase, assets and protection goals are derived from (abstract) requirements of the product philosophy (user experience, data protection, certifiability, ...) and by dividing large tasks into several small tasks (segregations of concerns), the system's ability to assess is prepared "in passing". Measures "from the scratch" are planned as part of the architecture (e.g. Encryption service) and enable a flexible design of the security for future changes. A continuous exchange and knowledge transfer takes place between developers and security ambassadors, security is perceived more as an enabler (features, usability, ...) and less as a blocker.

In the implementation a few institutions and responsible persons have to be involved in the project management, which increase the acceptance of security as a further and equal design goal of the team by their own positive action with a clear allocation of responsibilities and process flows. The security ambassadors are integrated in their domains into decision-making processes. This anchors a commitment to security with clear responsibilities in the project.

Résumé

For decades, attempts have been made to reconcile security and features. From our point of view, one step along this path is the integration of the Security Ambassador function into the agile development teams. Starting with Sprint zero, the ambassador works on an equal footing with the functional requirements at the product goal and has overall external responsibility for product safety. This addition to the agile model combines an agile approach with a holistic view of security based on the experience of our projects in recent years. By consistently treating the security requirements as equal targets, the security of a system can be assessed and therefore certified; a characteristic of supervision that is now required. Overall, IT security becomes a measurable quality feature such as functional product features that can be transported to the outside world and monetarized as a product feature.

Sources

The Open Web Application Security Project (OWASP): https://www.owasp.org/index.php/Main_Page

Dr. Waldemar Grudzien as Expert Director, deals with current regulatory requirements and their technical implementation. As PhD electrical engineer, he was responsible for retail banking and banking technologies in a national banking association.

Mail: waldemar.grudzien@core.se

COREinstitute
Am Sandwerder 21-23
14109 Berlin | Germany
https://institute.core.se
Phone: +49 30 26344 020
office@coreinstitute.org

COREtransform GmbH
Am Sandwerder 21-23
14109 Berlin | Germany
https://www.core.se
Phone: +49 30 26344 020
office@coretransform.de

COREtransform GmbH
Limmatquai 1
8001 Zürich | Helvetia
https://www.core.se
Phone: +41 442 610 143
office@coretransform.ch

COREtransform Ltd.
One Canada Square
London E14 5DY | Great Britain
https://www.core.se
Phone: +44 203 319 0356
office@coretransform.co.uk

COREtransform MEA LLC
DIFC – 105, Currency House, Tower 1
P.O. Box 506656 Dubai I UAE
https://www.core.se
office@coretransform.ae