Security Assurance im agilen Umfeld

In diesem White Paper wird die Problematik der Security Assurance im agilen Umfeld thematisiert. Die Sicherheitsexperten der mVISE AG haben sich mit den Konflikten dieser Disziplin auseinandergesetzt und entsprechende Handlungsempfehlungen formuliert.

Zusammenfasung

In den traditionellen Verfahrensmodellen (vgl. Wasserfallmodell oder V-Modell) stellt die Security Assurance – beim Übergang in die nächste Entwicklungsphase – die Sicherstellung der erstellten Artefakte anhand der Ergebnisse und der angelegten Dokumentation dar.

Im Gegensatz zu diesen starren Verfahrensmodellen bietet das agile Umfeld eine weitaus größere Flexibilität in einer jeden Phase, welche nur schwer abzubilden ist. Aus dieser Feststellung entspringt unsere Motivation, die Problemstellung zu beleuchten und eine mögliche Handlungsempfehlung auszusprechen. Dazu werden neue Ansätze, basierend auf wissenschaftlicher Publikation, sowie erste Thesen zur Implementierung vorgestellt.

Untersucht wird, inwieweit es möglich ist, Security Assurance in der agilen Softwareentwicklung zu integrieren. Dabei sollen erste Thesen zur Implementierung einer Security Assurance für ein allgemeines, agiles Softwareprojekt (z.B. nach dem Verfahrensmodell *Scrum*) aufgestellt werden.

Die Herausforderung

Wenn die Praktiken und Techniken von Security Assurance im Kontext von agilen Methoden angewendet werden, treten einige grundlegende Herausforderungen auf. Diese gehen zum einen aus dem Vorgehen dritter Sicherheitsexperten und zum anderen aus der Natur agiler Methoden hervor. Entwicklungsprojekte, die der Security Assurance unterliegen, sind auf drei wesentliche Aspekte angewiesen[1].

Auf die ZUVERLÄSSIGKEIT von Drittanbieterbewertungen, die ABHÄNGIGKEIT von der Bewertung von Drittanbietern und das VERTRAUEN auf Drittanbieter-Tests.

Einer der identifizierten Konflikte lässt sich darauf zurückführen, dass sich ein Verantwortlicher – im Widerspruch zur agilen Methodik – auf die schriftlich festgehaltene Dokumentation verlassen muss. Eine Schlüsselaufgabe agiler Security Assurance sollte es sein, frühzeitig die Identifikation von Designtypen und Codeänderungen vorzunehmen, da diese die typischen Sicherheitsprobleme darstellen.

[1] Vgl. (Beznosov & Kruchten 2005, p.2)

Dieses Paper wurde in enger Kooperation zwischen der Hochschule Niederrhein in Mönchengladbach und der mVISE AG im Rahmen einer wissenschaftlichen Arbeit erstellt.

Hochschule Niederrhein
University of Applied Sciences

Im Folgenden werden vier Ansätze in Bezug auf die beschriebene Problematik vorgestellt.

KOMPROMISSE EINGEHEN

Unter der Berücksichtigung, dass in einem agilen Entwicklungsumfeld die größten Fragen so früh wie möglich beantwortet werden müssen, schlagen wir vor, die unzuvereinbarenden Assurance Methoden[2] mindestens zweimal im Entwicklungslebenszyklus anzuwenden: Einmal nach der ersten „*User Story*" in einem Projekt und einmal kurz vor der Fertigstellung eines Moduls (Summe mehrerer zusammengehöriger „*User Stories*"). Der letztgenannte Anwendungspunkt ist eindeutig notwendig, um die Gewährleistung der Sicherheit im Endprodukt zu erhalten.

Tipp 1:
Sicherheitsexperten früh in das agile Projekt integrieren

Ersteres ermöglicht ein frühes Vertrauen in die Sicherheitseigenschaften und reduziert mögliche Redundanzen gegen Ende des Projekts. Zeit und Ressourcen erlauben zusätzliche Anwendungen der potentiell unzuvereinbarenden Methoden.

Der wesentliche Nachteil, der diesem Kompromiss einhergeht ist jedoch, dass dies zu einer umfangreicheren Dokumentationsarbeit führen wird, was dem agilen Verfahrensmodell widerspricht. Um dem entgegenzuwirken ist es ratsam, einen Sicherheitsexperten früh in den Entwicklungsprozess mit einzubeziehen. Dabei entsteht wiederum die Schwierigkeit eine Beeinflussung der beiden Rollen zu verhindern. Eine Möglichkeit sieht vor, dass der Sicherheitsexperte eine erziehende Rolle einnimmt und Sensibilisierungsmaßnahmen in Bezug auf Sicherheitsfragen bei dem Entwicklerteam vornimmt. Ein einzelner Sicherheitsexperte könnte so mehrere Entwickler bei der Umsetzung von Sicherheitsanforderungen unterstützen.

INKREMENTELL SICHERHEITSANFORDERUNGEN UMSETZEN

Das Wort Architektur wird in agilen Entwicklungsprojekten oft vermieden. Das liegt an der traditionellen Assoziation mit dem Top-Down-Design. Eine Architektur muss aber nicht als vollständiges Konzept abgedeckt werden. Eine iterative Sicherheitsarchitektur entwickelt sich mit dem System und beinhaltet nur die Funktionen, welche notwendig für die aktuelle Iteration oder Auslieferung sind. Sie ist eine Vorstellungshilfe für Entwickler, die die wichtigsten Sicherheitsaspekte des bestehenden Designs einkapselt.

Wenn Sicherheit iterativ entwickelt wird, dann ist es essentiell notwendig Sicherheitskriterien anzuwenden, die es den Entwicklern ermöglichen gute von schlechten Praktiken zu unterscheiden. Die grundsätzlichen Kriterien für Sicherheitsarchitekturen richten sich nach den konventionellen und allgemein anerkannten Prinzipien der Softwaresicherheit. Lose Kopplung zwischen Systemfunktionen, um die Anzahl an Funktionsaufrufen zwischen Modulen zu reduzieren, sowie die Vermeidung von funktionalen Duplikaten, sind etablierte Designmethoden und von entscheidender Bedeutung für die Einführung von Sicherheitsaspekten. Ebenso sind folgende Prinzipien für Sicherheitsarchitekturen relevant[3]:

- ÜBERPRÜFEN statt Vertrauen
- AUSFÜHREN aller Systemkomponenten mit den niedrigsten Rollenberechtigungen
- REDUZIEREN der Codekomplexität

Tipp 2:
Iterative Sicherheitsarchitektur verwenden!

Eine Top-Down-Architektur mit inkrementeller Implementierung kann eine Verschwendung von Designzeit sein, wird aber nur in hohen Kosten resultieren, wenn es zu einer fehlentschiedenen Lösung kommt. Auf der anderen Seite ist die inkrementelle Architektur eindeutig wirksam; sie bietet Flexibilität, Qualitätssicherung und minimiert Up-Front-Kosten.

SICHERHEITSANFORDERUNGEN ANHAND USER STORIES IDENTIFIZIEREN

Ein weiterer Ansatz, die im Konflikt stehenden Anforderungen in Einklang zu bringen ist es, das Konzept von „*Abuser Stories*" einzuführen. „*Abuser Stories*" identifizieren, wie Angreifer ein System missbrauchen und die Vermögenswerte der Stakeholder gefährden würden. Sie geben die Sicherheitsanforderungen des Systems an. Ähnlich wie „*User Stories*", sind sie kurz und informell.

[2] Vgl. (Beznosov & Kruchten 2005, p.2)
[3] Vgl. (Chivers et al. 2005, p.3)

„*Abuser Stories*“ erhalten, nach Einstufung und Bewertung der Bedrohung, die sie für die Vermögenswerte der Kunden darstellen, eine Punktzahl für ihr Ranking. Das Ranking muss berücksichtigen, wie der Schaden angerichtet werden kann und wie hoch die Wahrscheinlichkeit eines erfolgreichen Angriffs ist. Ihr Ranking sollte mit dem von „*User Stories*“ übereinstimmen.

Tipp 3:
„*Abuser Stories*“ verwenden, um das Risiko zu bewerten

In anderen Worten sollte eine „*Abuser Story*“ dasselbe Ranking haben, wie die dazugehörige „*User Story*“, wenn zu erwarten ist, dass die „*Abuser Story*“ den Ertrag der „*User Story*“ tilgt. Eine Reihe von „*Abuser Stories*“ ist effektiv gesehen die Grundstruktur eines Angriffsmodells.

„*Abuser Stories*“ weichen vom traditionellen agilen Requirements Engineering in einem solchen Ausmaß ab, dass sie nicht exklusiv von den Stakeholdern, sondern gemeinsam mit einem Sicherheitsexperten geschrieben werden, weil das ausgeprägte Fachgebiet der Experten sensibler für gewisse Arten von Bedrohungen ist, als das der nicht technischversierten Autoren. Im Folgenden werden die notwendigen Schritte gelistet, wie der Product/Sprint-Backlog zu verfeinern bzw. anzupassen ist[4]:

Schritt 1 - USER STORIES FORMULIEREN
Schritt 2 - ABUSER STORIES FORMULIEREN
Schritt 3 - WIDERLEGUNGEN PRÜFEN
Schritt 4 - USER STORIES IMPLEMENTIEREN
Schritt 5 - AKZEPTANZTESTS DURCHFÜHREN
Schritt 6 - ABUSER STORY IMPLEMENTIEREN

GRAD DER AGILITÄT VON SICHERHEITSANFORDERUNGEN ERMITTELN

Nicht-funktionale Anforderungen – wie Anforderungen an die Sicherheit – können einem System nicht, wie funktionale Anforderungen, hinzugeführt werden. Das Entwicklerteam sollte den Sicherheitsaspekten während des Softwarelebenszyklus große Aufmerksamkeit geben. Um einer Verringerung der Agilität eines agilen Vorgehensmodells entgegenzuwirken, wird eine aus mehreren Schritten bestehende Methode vorgestellt.
Die Klassifizierung von Sicherheitsaktivitäten wird dabei helfen, jede Aktivität und ihren Ausführungskontext besser zu verstehen und die damit verbundenen Aktivitäten in der geeigneten Kategorie[5] zu sehen.

Der Agilitätsgrad stellt die Kompatibilität einer Aktivität und der agilen Methodik dar und wird auf Grundlage von Agilitätsmerkmalen wie SIMPLIZITÄT, AUSFÜHRUNGSGESCHWINDIGKEIT und PERSONENORIENTIERUNG berechnet.

Es werden die Hauptmerkmale[6] der agilen Methode betrachtet und einer Punktzahl zwischen 0 und 5 zugewiesen, was zu einer einspaltigen Matrix namens

- Agilitäts Grad Vektor -

(ADVect) führt. Eine große Zahl deutet auf eine hohe Kompatibilität mit diesem Merkmal hin – also grünes Licht für die Implementierung – und eine niedrige Zahl deutet auf einen Konflikt hin.

Tipp 4:
„*Agile Readiness*“ bestimmen, um Sicherheitsanforderungen umzusetzen

Neben den Sicherheitsaktivitäten sollte dieser Wert für extrahierte agile Aktivitäten berechnet werden. Es können Agilitätsmerkmale dieser beiden Arten von Aktivitäten verglichen und auch den Agilitätsgrad einer kombinierten Aktivität eingeschätzt werden.

Die Integration von zwei Aktivitäten mit dem Agilitätsgrad von a und b ergibt eine Aktivität mit einem Minimum (a, b) als ihren Grad. Es ist unmöglich, eine Sicherheitsaktivität mit allen agilen Aktivitäten zu integrieren.

Mit dieser Vorgehensweise wird gezeigt, dass der Grad der Agilität von Sicherheitsanforderungen ein wesentliches Merkmal bietet um ein agiles Entwicklungsprojekt mit Security Assurance Methoden zu kombinieren.

[4] Vgl. (Peeters 2004, pp.2–3); [5] Vgl. (Keramati & Mirian-Hosseinabadi 2008, p.3); [6] Vgl. (Keramati & Mirian-Hosseinabadi 2008, p.4)

Profitieren Sie von qualifizierten Lösungen

Die mVISE AG berät, unterstützt und setzt um! Bei Herausforderungen in Ihrer Projektmanagement-Organisation helfen wir Ihnen, diese zu meistern. Unser Schwerpunkt liegt dabei auf der Einführung von agilen Projektmethoden wie Scrum in bestehende Organisationen und der optimalen Umsetzung der anzuwendenden Tools, wie etwa des Product Backlogs. So bietet die mVISE ihren Kunden auch die nötige Hilfestellung bei der agilen Transformation im MANAGEMENT, der ENTWICKLUNG und der QUALITÄTSSICHERUNG an. Darüber hinaus kann die mVISE auch das Projektmanagement und weitere Funktionen in Projektteams übernehmen und Ihre Projekte zu einem erfolgreichen Abschluss führen.
Unsere IT-Security-Experten haben in den letzten Jahren in der Security Assurance umfassende Erfahrungen und Erkenntnisse aus vielen verschiedenen IT-Umfeldern sammeln können.

Ausblick

Mit den hier in diesem Paper aufgeführten Problematiken zeigen wir, dass Security Assurance in agilen Entwicklungsprojekten nicht ohne Weiteres zu vereinen ist. Mit den Handlungsempfehlungen werden Ansätze geliefert, wie eigene Scrum-Prozessabläufe optimiert werden können. Dabei sind letztendlich Kompromisse einzugehen, um eine Vereinbarung zu erzielen. Ohne Kompromisse ist eine Vereinbarung nach unserem Ermessen nicht möglich.
Für die Zukunft empfehlen wir die genannten Handlungsempfehlungen an einem konkreten Praxisbeispiel anzuwenden und Informationen zu sammeln, um die Bedeutsamkeit und Relevanz der Handlungsempfehlungen zu messen.

Über die Autoren

Nabil Arzouni und Steven Grimberg studieren im Masterstudiengang der Wirtschaftsinformatik an der Hochschule Niederrhein. Sie sind Ansprechpartner im Fachbereich Wirtschaftswissenschaften und unterstützend bei der Konzeption und Realisierung des neuen Kompetenzzentrums für IT-Sicherheit als wissenschaftliche Hilfskräfte mit. Zurzeit begleiten sie zahlreiche Projekte im Themenfeld IT-Sicherheit, während sie mit verschiedenen Unternehmen, wie der mVISE AG, zusammenarbeiten und unterschiedliche informationstechnische Problemfelder beleuchten.

Bernhard Borsch ist Manager im IT-Security Consulting bei der mVISE AG. Zu seinen persönlichen Themenfeldern gehört neben den klassischen Themen, wie Enterprise- & Cloud-Security, PKI und Kryptographie, auch Zukunftsthemen der IT Security, wie Blockchain und Deception Technology. Professionelle und fachkompetente Beratung sind sein Schlüssel zum Erfolg.

Referenzen

Beznosov, K. & Kruchten, P., 2005.
Towards agile security assurance. Proceedings of the 2004 workshop on New security paradigms - NSPW '04, p.47. Available at: http://portal.acm.org/citation.cfm?doid=1065907.1066034.

Chivers, H., Paige, R.F. & Ge, X., 2005.
Agile Security Using an Incremental Security Architecture. Extreme Programming and Agile Processes in Software Engineering Lecture Notes in Computer Science, pp.57–65.

Keramati, H. & Mirian-Hosseinabadi, S.H., 2008.
Integrating software development security activities with agile methodologies. AICCSA 08 - 6th IEEE/ACS International Conference on Computer Systems and Applications, pp.749–754.

Peeters, J., 2004.
Agile Security Requirements Engineering. Independent, p.4.

mVISE

Wir unterstützen mittelständische und große Unternehmen aller Branchen dabei, von der digitalen Revolution zu profitieren. Die besondere Kombination aus firmeneigenen Software-Lösungen mit ausgewählten Experten-Teams in den relevanten und aktuellen IT-Themengebieten schafft nachhaltige Wettbewerbsvorteile für unsere Kunden.

Unsere Experten bestimmen, gestalten, kreieren und steuern IT-Infrastrukturen und Software-Lösungen für Datenintegrations- und Enterprise-Data-Management-Projekte, mit dem Ziel, die aktuellen Geschäftsmodelle unserer Kunden zukunftssicher zu machen und gleichzeitig neue Geschäftsmodelle zu identifizieren.

Sprechen Sie uns an – gerne stellen wir Ihnen unser Angebot
in einem persönlichen Gespräch näher vor.
service@mvise.de | www.mvise.de

mVISE AG
Wahler Straße 2
40472 Düsseldorf
Fon: +49 211 78 17 80 – 0