FOM Hochschule für Oekonomie & Management

Hochschulzentrum München

Hausarbeit

im Studiengang IT-Management

im Rahmen der Lehrveranstaltung
IT-Projektmanagement & Software-Engineering

über das Thema

Gestaltung eines Scrum-Frameworks unter Einbeziehung der IT-Sicherheit

von

Felix Schubert

Betreuer : [redacted]
Matrikelnummer : [redacted]
Abgabedatum : [redacted]

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Der Lagebericht „IT-Sicherheit 2021“ des Bundesamts für Sicherheit in der Informationstechnik zeigt, dass die aktuelle Gefährdungslage für Hackerangriffe in Deutschland so hoch ist wie nie zuvor. Pro Monat entstehen mehrere Millionen neue Varianten von Schadprogrammen [Bun21, S. 11]. Cyberangriffe führten bereits bei 86 Prozent der befragten Unternehmen zu Schäden [BS21, S. 6]. Deshalb sollte bereits bei der Entwicklung neuer Software darauf geachtet werden, diese sicher zu programmieren. Da agile Projektmanagement-Methoden immer beliebter werden (vgl. Tabelle 1), ist es zwingend erforderlich, die IT-Sicherheit auch hier fest in den Prozess mit einzubauen.

Tabelle 1: Aufschlüsselung der weltweit praktizierten Softwareentwicklungsmethoden im Jahr 2021 [Git21, S. 5]

Merkmal	Anteil der Befragten
DevOps/DevSecOps	35,9%
Agile/Scrum	31,8%
Kanban	13%
Waterfall	10%
Water/Scrum/Fall	5%
Lean	4,2%

1.1 Problemstellung

Die Tabelle 1 zeigt, dass Scrum neben DevOps/DevSecOps zu den weltweit am häufigsten genutzten Softwareentwicklungsmethoden im Jahr 2021 gehört. Das agile Vorgehensmodell Scrum folgt einem eigenen Manifest (vgl. Kapitel 2.2.1). Einer der Hauptbestandteile der Methode stellt das Backlog dar, das mit Anforderungen an das System gefüllt ist, welche in Iterationen abgearbeitet werden. Ghani et al. [GAJ14, S. 648] sehen durch die iterative Entwicklung von Software verschiedene Limitationen in Bezug auf die IT-Sicherheit. Ein Beispiel ist, dass Teammitglieder nicht über ausreichende Kenntnisse zur Thematik verfügen und in den Entwicklungszyklen kaum Zeit ist, Anforderungen der IT-Sicherheit zu adressieren.

„There are relatively few studies in the literature on secure agile software development models. In Wichers proposal [...] it is argued that secure software development in the agile model needs a quite different approach to that of the waterfall model“ [RJ18, S. 471] nach [Wic08].

1.2 Zielsetzung und Forschungsfrage

Diese Arbeit verfolgt das Ziel, mit Hilfe von Literaturarbeit verschiedene Ansätze für die Implementierung von IT-Security Anforderungen im Scrum-Prozess zu sammeln. Im Anschluss daran sollen diese evaluiert und Gemeinsamkeiten gefunden werden. Mit Hilfe der dadurch gewonnenen Erkenntnisse wird ein Scrum-Framework entwickelt, welches die IT-Sicherheit berücksichtigt. Deshalb beschäftigt sich diese Arbeit mit der folgenden Forschungsfrage:

> Wie gestaltet sich ein Framework, welches die IT-Sicherheit bei einem Scrum Ansatz berücksichtigt?

1.3 Forschungsmethodik

Für die Sammlung der unterschiedlichen Implementierungsarten wird auf die Literaturarbeit zurückgegriffen. Diese orientiert sich nach dem Vorgehensmodell von vom Brocke et al. [Vom+09], welches den Prozess in fünf (v) verschiedene Phasen aufgliedert (vgl. Abbildung 1).

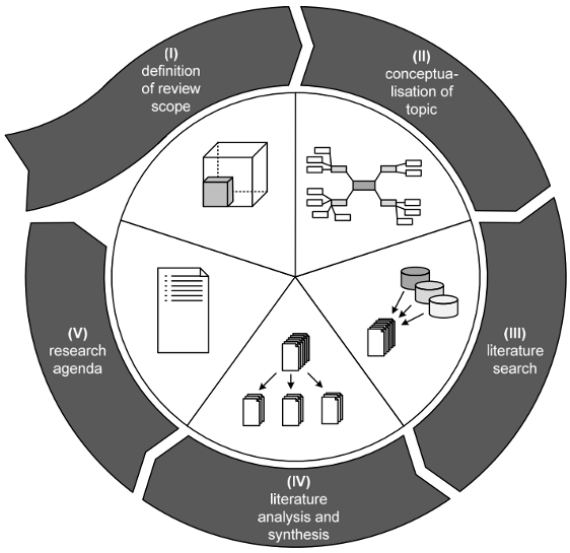

Abbildung 1: Framework für die Literaturrecherche nach vom Brocke et al. [Vom+09, S. 8]

Die Definition des Suchraums (i) erfolgt durch die Eingrenzung des Themas mit Hilfe der Zielsetzung, einer Forschungsfrage (vgl. Kapitel 1.2) sowie der Taxonomie nach Cooper [Coo88]. Die Konzeptualisierung des Themas (ii) wird mit der Schaffung der theoretischen Grundlagen im Kapitel 2.2 vorgenommen. Daraufhin wird die Suche nach passender Literatur (iii), sowie die Literaturanalyse und -synthese (iv) nach den Prinzipen von Webstor

und Watson [WW02] durchgeführt. Auf das Aufstellen einer Forschungsagenda (v) wird in dieser Arbeit verzichtet, da sie nicht das Ziel verfolgt, gefundene Forschungslücken wiederzugeben. Die einzelnen Phasen der Literaturarbeit werden im Kapitel 3.2 vertiefend definiert.

1.4 Aufbau der Arbeit

Im einleitenden Kapitel dieser Arbeit wird zum Thema hingeführt und die aktuelle Problemstellung detailliert erläutert (vgl. Kapitel 1.1). Auf dieser baut die entwickelte Zielsetzung auf, aus welcher die Forschungsfrage abgeleitet wird (vgl. Kapitel 1.2). Die anschließende Erläuterung der Forschungsmethodik zeigt auf, wie diese, mit Hilfe passender wissenschaftlicher Mittel beantwortet wird.

Das Kapitel 2 beschreibt den Prozess der Literaturarbeit. Es erfolgt die Definition des Suchraums und die Konzeptualisierung des Themas. Im Anschluss daran wird der Prozess der Literatursuche nach Webster und Watson [WW02] dargelegt. Abschließend erfolgt eine tabellarische Darstellung der Ergebnisse.

Im Kapitel 3 werden zunächst die Umsetzungskonzepte aus der bestehenden Literatur präsentiert, woraus nachfolgend ein Modell namens VS-Scrum vorgestellt wird. Dabei handelt es sich um ein Framework, welches die IT-Sicherheit bei einem Scrum Ansatz berücksichtigt. Zum Schluss werden im Kapitel 4 die Limitationen dieser Arbeit betrachtet sowie der Mehrwert für die Praxis und Wissenschaft erläutert. Der Ausblick zeigt zuletzt Möglichkeiten auf, wie die Forschung weitergeführt werden kann.

2 Literaturarbeit

Im Literaturteil dieser Arbeit wird im ersten Schritt der Suchraum definiert. Anschließend erfolgt im Rahmen der Konzeptualisierung die Schaffung der theoretischen Grundlagen. Im Nachgang findet die Literatursuche statt. Abschließend wird eine Matrix präsentiert, die aufzeigt, welche Konzepte in den verschiedenen Werken vertreten sind.

2.1 Definition des Suchraums

Zu Beginn erfolgt entsprechend dem Modell der Literaturarbeit von vom Brocke et al. [Vom+09] die Definition des Suchraums. Hierfür wird auf die Taxonomie von Cooper [Coo88] zurückgegriffen. Die unterschiedlichen Arten der Klassifikation zeigt die nachfolgende Tabelle 2 auf.

Tabelle 2: Die Definition des Suchraums in Anlehnung an Cooper [Coo88, S. 109]

Kategorie		Ausprägung			
(1)	Fokus	Forschungs-ergebnisse	Forschungs-methoden	Theorien	Praxis/ An-wendungen
(2)	Ziel	integrierend	kritisierend	herausfordernd	
(3)	Perspektive	neutrale Repräsentation		positionsbeziehend	
(4)	Abdeckung	vollständig	vollständig selektiv	repräsentativ	zentral
(5)	Organisation	historisch	konzeptuell	methodisch	
(6)	Zielgruppe	spezielle Wis-senschaftler	Allgemeine Wissen-schaftler	Praktiker/ Entschei-dungsträger	Allgemeinheit

Der Fokus (1) wird zum einen auf Forschungsergebnisse in der bisherigen Literatur gelegt, um bereits bestehende Modelle zu finden, welche die IT-Sicherheit in einem Scrum Ansatz berücksichtigen. Zum anderen werden Erkenntnisse aus der Praxis gesammelt, um Best-Practices herauszuarbeiten, für die noch kein eigenes Modell entwickelt wurde. Durch die Sammlung und Analyse bestehender Literatur entsteht ein Ziel (2), welches sich nach Cooper [Coo88, S. 108] als integrierend beschreibt. Trotz der Annahme von Cooper, dass eine vollständig neutrale Position des Autors unwahrscheinlich ist, wird die Perspektive (3) der neutralen Repräsentation gewählt.

Die Abdeckung (4) wird als repräsentativ eingestuft, da alles darüber hinaus im Rahmen dieser Arbeit nicht möglich wäre. Weil diese Arbeit das Ziel verfolgt, ein Modell zu entwickeln, wird die Organisation (5) konzeptuell erfolgen. Aus diesem Grund werden Werke mit

gleicher Idee eines Ansatzes zusammen dargestellt. Die Zielgruppe (6) besteht zum einen aus allgemeinen Wissenschaftlern, die mit Hilfe der gewonnen Ergebnisse weiterforschen und zum anderen aus Praktikern/Entscheidungsträgern, welche das ausgewählte Modell experimentell testen und evaluieren können. Im weiteren Verlauf erfolgt die Konzeptualisierung der Thematik.

2.2 Konzeptualisierung des Themas

In diesem Abschnitt der Arbeit werden die theoretischen Grundlagen zu den Themen geschaffen, die relevant für das weitere Verständnis in Bezug auf die Beantwortung der Forschungsfrage sind. Dabei wird zu Beginn das Vorgehensmodell Scrum beschrieben. Im Anschluss daran werden die Kernprinzipien der IT-Sicherheit vorgestellt.

2.2.1 Scrum

„Scrum ist ein leichtgewichtiges Rahmenwerk, welches Menschen, Teams und Organisationen hilft, Wert durch adaptive Lösungen für komplexe Probleme zu generieren“ [SS20, S. 3]. Für den Lösungsprozess nach Scrum wird ein Team aufgestellt, welches sich aus den nachfolgenden Rollen zusammensetzt [SS20, S. 5 ff.]:

Der Scrum Master unterstützt bei der Einhaltung der Scrum-Praktiken und koordiniert zwischen dem Product Owner und dem Entwicklungsteam [SS20, S. 8], [MMB17, S.2].

Der Product Owner ist durch die Festlegung von Zielen und Anforderungen an das Projekt verantwortlich, den Wert des Produktes zu maximieren [SS20, S. 6], [MMB17, S.2].

Die Entwickler tragen die Verantwortung für die Implementierung der Software, wobei es innerhalb des Teams keine strikten Rollen oder Hierarchien gibt [SS20, S. 6], [MMB17, S.2].

Das Vorgehensmodell Scrum beinhaltet zudem verschiedene Events und Artefakte (vgl. Abbildung 2). Im Folgenden werden einige davon näher beschrieben.[1]

Der Product Backlog (i) umfasst eine Sammlung aller Anforderungen, die zur Verbesserung des Produktes beitragen [SS20, S. 11]. Im nachfolgenden Sprint Backlog (ii) werden

[1] Eine genaue Erläuterung aller Scrum Events und Scrum-Artefakte finden Sie im Werk von Schwaber und Sutherland [SS20].

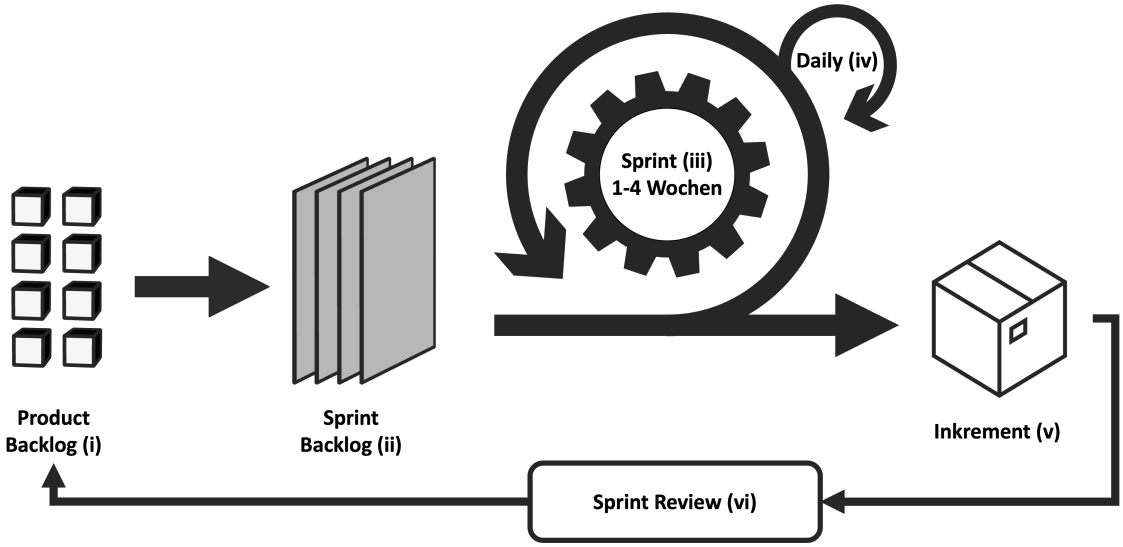

Abbildung 2: Scrum-Modell in Anlehnung an Probst [Pro19, S. 2]

alle Anforderungen gesammelt, die im nächsten Sprint bearbeitet werden sollen [SS20, S. 12]. Anschließend erfolgt im Sprint (iii), die Umsetzung aller Anforderungen aus dem Sprint Backlog innerhalb eines zuvor definierten Zeitraums, ohne Änderungen daran vorzunehmen [SS20, S. 8]. Als Hilfsmittel zum Austausch während eines Sprints wird ein Daily Scrum (iv) durchgeführt. Das Ergebnis des Sprints ist ein funktionierendes Inkrement (v), welches „ein konkreter Schritt in Richtung des Produkt-Ziels“ [SS20, S. 13] darstellt. Das Ziel der Sprint Reviews (vi) ist die Besprechung des bisherigen Vorgehens und Problematiken des vorangegangenen Sprints, um die Qualität der nachfolgenden Iterationen sowie des Gesamtproduktes zu verbessern [SS20, S. 10]. Ein weiterer wichtiger Aspekt der Forschungsthematik dieser Arbeit stellt die IT-Sicherheit dar, welche im nächsten Kapitel grundlegend beschrieben wird.

2.2.2 IT-Sicherheit

„Informationen sind ein wesentlicher Wert für Unternehmen und Behörden und müssen daher angemessen geschützt werden“ [Bun21, S. 1]. Die IT-Sicherheit ist ein Teil der Informationssicherheit und beschreibt die „Methoden und Maßnahmen des Managements, die für eine sichere IT-Umgebung in einem Unternehmen geschaffen werden“ [Lee20, S. 51]. Bei der Informationssicherheit wird im Gegensatz zur IT-Sicherheit der ganzheitliche Prozess von Informationen betrachtet und nicht nur digitale Dokumente [Lee20, S. 51]. Die Maßnahmen der IT-Sicherheit folgen dem Akronym CIA, auf welches nachfolgend eingegangen wird.

Confidentiality = Vertraulichkeit - Wer darf in welcher Art auf Informationen zugreifen.

Integrity = Integrität - Wer darf Informationen verändern/löschen.

Availability = Verfügbarkeit - Wann muss der Zugang zu Informationen gesichert sein.

Das Ziel dieser Prinzipien und somit der IT-Sicherheit ist die technische Absicherung der IT-Infrastruktur. Zusätzlich zu den genannten Grundsätzen existieren weitere zur sicheren Softwareentwicklung.

Das Prinzip der geringsten Privilegien gibt an, dass ein Nutzer nur die Rechte haben soll, die er zum Ausführen einer Tätigkeit benötigt [SS75, S. 1282 f.], [AN16, S. 102]. Die ausfallsichere Standardeinstellung bestimmt, dass der Zugriff standardmäßig nicht gegeben ist. Außerdem werden die Bedingungen festgelegt, unter denen der Zugriff erlaubt ist [AN16, S. 102]. Auf diese beiden Mechanismen setzt beispielsweise auch der Zero-Trust-Ansatz [Ros+20, S. 4 ff.]. Das offene Design sagt aus, dass Sicherheitsmechanismen nicht nur funktionieren dürfen, weil sie unbekannt sind [SS75, S. 1282 f.]. Ein weiteres interessantes Prinzip ist die psychologische Akzeptanz. Dabei sollen Schnittstellen, mit welchen Personen interagieren, leicht zu bedienen sein, damit die Nutzer die Schutzmechanismen routinemäßig und automatisch richtig nutzen [AN16, S. 102]. Saltzer und Schroeder [SS75] stellen in ihrem Werk *The protection of information in computer systems* noch weitere Prinzipien vor. Eine Erläuterung aller Prinzipien ist im Rahmen dieser Arbeit nicht möglich. Im Anschluss an die Konzeptualisierung stellt die Literatursuche den nächsten Schritt dar, welche deshalb als nächstes erläutert wird.

2.3 Literatursuche

Zunächst erfolgt die Identifizierung relevanter Literatur nach Webster und Watson [WW02, S. xvi]. Hierfür werden die Literaturdatenbanken Google Scholar und Ebsco Discovery Service[2] genutzt. Anhand der gefundenen Literatur werden mit Hilfe der Vorwärts- und Rückwärtssuche weitere Werke identifiziert. Die dadurch gesammelte Literatur wird im nachfolgenden Kapitel vorgestellt.

2.4 Ergebnisse der Literaturarbeit

Die nachfolgende Tabelle 3 zeigt die als relevant identifizierte Literatur.

[2] Dort sind Inhalte von Business Source Premier, EconLit, APA PsycArticles, PSYNDEX, Medline, CINAHL, Engineering Source, GreenFILE, der IEEE Xplore digital Library, der ACM Digital Library und des Springer-Links enthalten.

Tabelle 3: Ergebnisse der Literatursuche

Nr.	Titel	Jahr	Methode	Security Ansatz
1	Security backlog in scrum security practices [AGI11]	2011	Literaturarbeit	Es gibt zusätzlich zu einem normalen Backlog einen Security Backlog, welcher von einem Security Master gemanagt wird. Dieser prüft die Aufgaben aus dem normalen Backlog und fügt Security Tasks hinzu, wenn diese benötigt werden.
2	Secure Scrum: Development of Secure Software with Scrum [PH15]	2015	Feldexperiment	Es wird der Ansatz von Secure Scrum genutzt, welcher aus den Komponenten Identifizierung, Implementierung, Verifizierung und Definition of Done besteht. User-Storys werden bei Bedarf mit einem Security-Tag versehen. Des Weiteren wird eine Security User-Story dazu formuliert.
3	A Security Testing Framework for Scrum based Projects [RM16]	2016	Modellentwicklung auf der Basis von Literaturarbeit	Es gibt zwei Stufen von Security Tests. Die erste Stufe überprüft das Programm und erfolgt im direkten Anschluss an den Sprint. Die zweite Stufe wird nach der Fertigstellung des Inkrements durchgeführt und soll durch einen Pentest die Schwachstellen im Netzwerk aufdecken.

Fortsetzung auf nächster Seite

Tabelle 3 – *Fortsetzung von vorheriger Seite*

Nr.	Titel	Jahr	Methode	Security Ansatz
4	Integrating Software Security into Agile-Scrum Method [GAJ14]	2014	Modellweiterentwicklung auf der Basis von Literaturarbeit und Experimenten	Der Scrum-Prozess wird durch einen Security Backlog ergänzt und die Rolle des Security Masters wird etabliert. Dieser prüft die Aufgaben aus dem normalen Backlog und fügt Security Tasks hinzu, wenn diese benötigt werden.
5	Penetration Testing in Agile Software Development Projects [TK15]	2015	Modellentwicklung auf der Basis von Literaturarbeit	Es wird auf einen eigenen Security Backlog verzichtet. Die Security Anforderungen werden stattdessen in den Product Backlog mit aufgenommen. Während des Sprints werden automatische Penetrationstest durchgeführt. Im Anschluss an den Sprint erfolgen manuelle Penetrationstests.
6	Case Study of security development in an agile environment: Building identity management for a government agency [RHL16]	2016	Fallstudie	Es gibt eigene Entwickler, deren Fokus auf der IT-Sicherheit liegt. Sie bringen Anforderungen an die IT-Sicherheit in den Backlog mit ein. Während dem Sprint finden Audits und Pentestings statt. Einen wichtigen Teil stellt die Dokumentation aller Sicherheitstätigkeiten im Nachgang dar.

Fortsetzung auf nächster Seite

Tabelle 3 – *Fortsetzung von vorheriger Seite*

Nr.	Titel	Jahr	Methode	Security Ansatz
7	Breaking the Waterfall Mindset of the Security Industry and OWASP Conferences Chair The OWASP Foundation [Wic08]	2008	—	Zu Beginn werden die einzelnen Stakeholder identifiziert. Anschließend werden die verschiedenen User-Storys hinsichtlich ihres Einflusses auf die IT-Schutzziele geprüft. Abschließend formulieren die Stakeholder Security-User-Storys, womit eine Kompromittierung der Schutzziele verhindert werden kann. Darüber hinaus gibt es Sprints, in welchen nur Thematiken der IT-Sicherheit überprüft werden.
8	US-Scrum: A Methodology for Developing Software with Enhanced Correctness, Usability and Security [Raf+15]	2015	Modellentwicklung auf der Basis von Literaturarbeit	Es gibt einen gemeinsamen Backlog, in dem alle Anforderungen konsolidiert werden. Im Anschluss daran erfolgt die Aufteilung in Merkmale bezüglich Sicherheit, Funktionalität und Benutzerfreundlichkeit. Während des eigentlichen Sprints finden zwei kleine Security Sprints statt.
9	Towards a Secure SCRUM Process for Agile Web Application Development [MMB17]	2017	Modellentwicklung auf der Basis von Literaturarbeit	Im Vorfeld der Veröffentlichung gibt es einen Security Sprint. In diesem werden Security-User-Storys formuliert und umgesetzt, Penetrationstests durchgeführt und eine Risikoanalye erstellt. Daraufhin erfolgt die Validierung mit Hilfe von Code-Reviews und paarweisen Penetrationtstests.

Nach der Vorstellung der Forschungsergebnisse erfolgt die Organisation (vgl. Kapitel 2.1). Dafür werden den herausgearbeiteten Konzepten Buchstaben zugeordnet.

A Neben dem normalen Product Backlog wird ein Security Backlog etabliert, in welchem alle Security-User-Storys gesammelt werden [AGI11, S. 416].

B Für die IT-Security relevante User-Storys werden mit einem S-Tag versehen. Dabei handelt es sich um eine detaillierte Beschreibung eines Sicherheitsproblems in Form einer User-Story, die vom Scrum-Team genutzt wird, um die Problematik besser zu verstehen [PH15, S. 3].

C Für die Verantwortung des Security Backlogs (A) wird die Rolle des Security Masters etabliert. Dieser ist verantwortlich für die Identifizierung von User-Storys, die ein Sicherheitsrisiko darstellen sowie die Dokumentation und Durchführung von Penetrationstests [AGI11, S. 416].

D Es findet mindestens eine Form von Penetrationstest statt. Diese können sowohl manuell also auch automatisch durchgeführt werden.

E Zusätzlich zu den normalen Sprints werden separate, sogenannte Security Sprints, durchgeführt, in welchen lediglich Anforderungen der IT-Security bearbeitet werden.

Die untenstehende Tabelle 4 zeigt die Organisation der Literatur.

Tabelle 4: Organisation der Literatur

Konzepte	Werk								
	1	2	3	4	5	6	7	8	9
A	X			X					
B		X		X	X		X	X	
C	X			X		O			
D			X	X	X	X		X	X
E							X	X	X

X = Konezpt kommt im Werk vor.
O = Ähnliches Konzept kommt im Werk vor.

Nachdem die bisherigen Forschungsergebnisse strukturiert wurden, soll im nächsten Kapitel ein VS-Scrum-Framework entwickelt werden.

3 Entwicklung von VS-Scrum

Im weiteren Verlauf werden zunächst die verschiedenen Konzepte aus der Literaturorganisation vorgestellt. Daraus wird im Anschluss daran ein ganzheitliches Framework entwickelt, welches die Forschungsfrage beantwortet.

3.1 Umsetzungskonzepte

Die zuvor abgebildete Tabelle 4 zeigt sowohl die Gemeinsamkeiten als auch Unterschiede der einzelnen Konzepte auf. Es wird ersichtlich, dass ein Security Backlog nur dann etabliert wird, wenn es auch einen Security Master gibt, welcher diesen managen kann (vgl. Tabelle 4). Im Modell von Rindell et al. wird nicht von einem Security Master gesprochen, allerdings gibt es eine ähnliche Funktion, den Security Developer [RHL16, S. 559].

Mehr als die Hälfte der Security Ansätze bauen anstatt des Security Backlogs auf das Konzept B (vgl. Tabelle 4). Dabei wird der Schadenswert einer User-Story geschätzt, der entsteht, wenn die verarbeiteten Daten, die in einer Story verarbeitet, kompromittiert werden [PH15, S. 3]. „User-Stories werden [anschließend] mittels sog. S-Marks gekennzeichnet welche wiederum auf ein S-Tag verweist - ein S-Tag beschreibt das entsprechende Security Problem als auch mögliche Lösungsansätze (z.B. Verweise auf Best Practices)“ [Rie17, S. 2].

Ein wichtiger Bestandteil der verschiedenen Ansätze ist bei mehr als 65 Prozent die Durchführung von Penetrationstests (vgl. Tabelle 4). Die vorgestellten Artikel 3, 5, 8 und 9 setzten dabei auf einen hybriden Ansatz; das heißt es werden sowohl automatisierte als auch manuelle Penetrationstests durchgeführt. In den Werken 4 und 6 wird lediglich von Penetrationstests gesprochen, jedoch werden diese nicht näher definiert.

Während die meisten Werke auf eine Umsetzung von Security Anforderungen während des normalen Sprints setzen, bauen die Artikel 7, 8 und 9 auf einen eigenen Security-Sprint. Rafi et al. empfehlen in ihrem vorgeschlagenen Modell zwei kleinere Security-Sprints während des funktionellen Sprints [Raf+15, S. 381]. Meier et al. setzen hingegen auf einen Security Sprint bereits vor der Durchführung des funktionellen Sprints [MMB17, S. 5]. Mit Hilfe der gewonnen Erkenntnisse wird im nachfolgenden Kapitel 3.2 ein Framework entwickelt, womit die Forschungsfrage beantwortet wird.

3.2 Modellvorschlag

Im Kapitel 2.4 werden die verschiedenen Konzepte der Implementierung von IT-Sicherheit in den Scrum-Ansatz vorgestellt. Ziel dieser Arbeit ist es, ein Framework vorzustellen, welches diese konsolidiert (vgl. Kapitel 1.2). Hierfür wird nachfolgend das Framework VS-Scrum erläutert.

Für die Koordination der einzelnen Security Anforderungen wird die Rolle des Security-Masters etabliert (vgl. Konzept C). Dieser hat im VS-Scrum Modell drei verschiedene Tätigkeiten (vgl. Abbildung 3) [AGI11, S. 416]:

1. Verantwortung über den Security Backlog.
2. Prüfung hinsichtlich der Notwendigkeit von S-Tags für User-Storys aus dem Product Backlog
3. Koordination des Security Sprints

Im Security Backlog werden die Risiken dokumentiert, die entweder in den Security Sprints oder im Netzwerk-Penetrationstest aufgetreten sind (vgl. Konzept A). Dem Security Master obliegt die Entscheidung, welche User-Storys er aus dem Security Backlog in das Sprint Planning mit aufnimmt. Bei User-Storys aus dem Product Backlog hat der Security Master die Aufgabe, zu prüfen, ob diese einen Security-Tag benötigen (vgl. Konzept B).

Das Ergebnis des Sprint Plannings ist das Artefakt Sprint Backlog. In diesem sind alle User-Storys enthalten, die im nachfolgenden Sprint umgesetzt werden sollen. Der Sprint kann je nach Bedarf eine Dauer zwischen einer und vier Wochen haben. Zum Austausch während des Sprints erfolgt ein tägliches Meeting namens Daily (vgl. Kapitel 2.2.1).

Zum Ende eines jeden Sprints erfolgt ein sogenannter Security Sprint (vgl. Abbildung 3 und Konzept E). Ziel dieses Security Sprints ist es, die User-Storys aus dem Security Backlog und jene mit S-Tag umzusetzen. Bereits während der Umsetzungsphase erfolgen automatisierte Penetrationstests[3] (vgl. Konzept D). Im Anschluss an den Security Sprint erfolgt ein Netzwerk-Penetrationstest, bei dem neben der Anwendung beispielsweise die Firewall getestet wird [RM16, S. 15]. Hierbei gefundene Risiken werden wieder zurück in den Security Backlog gegeben. Diese Konsolidierung aller Ansätze bietet das höchstmögliche Ausmaß an IT-Security bei der Anwendungsentwicklung. Die nachfolgende Abbildung 3 veranschaulicht abschließend das VS-Scrum-Framework.

[3] Eine Möglichkeit der Automatisierung wäre beispielsweise mit Hilfe der Pipeline Tools von OWASP möglich: https://owasp.org/www-project-appsec-pipeline/pipeline-tools

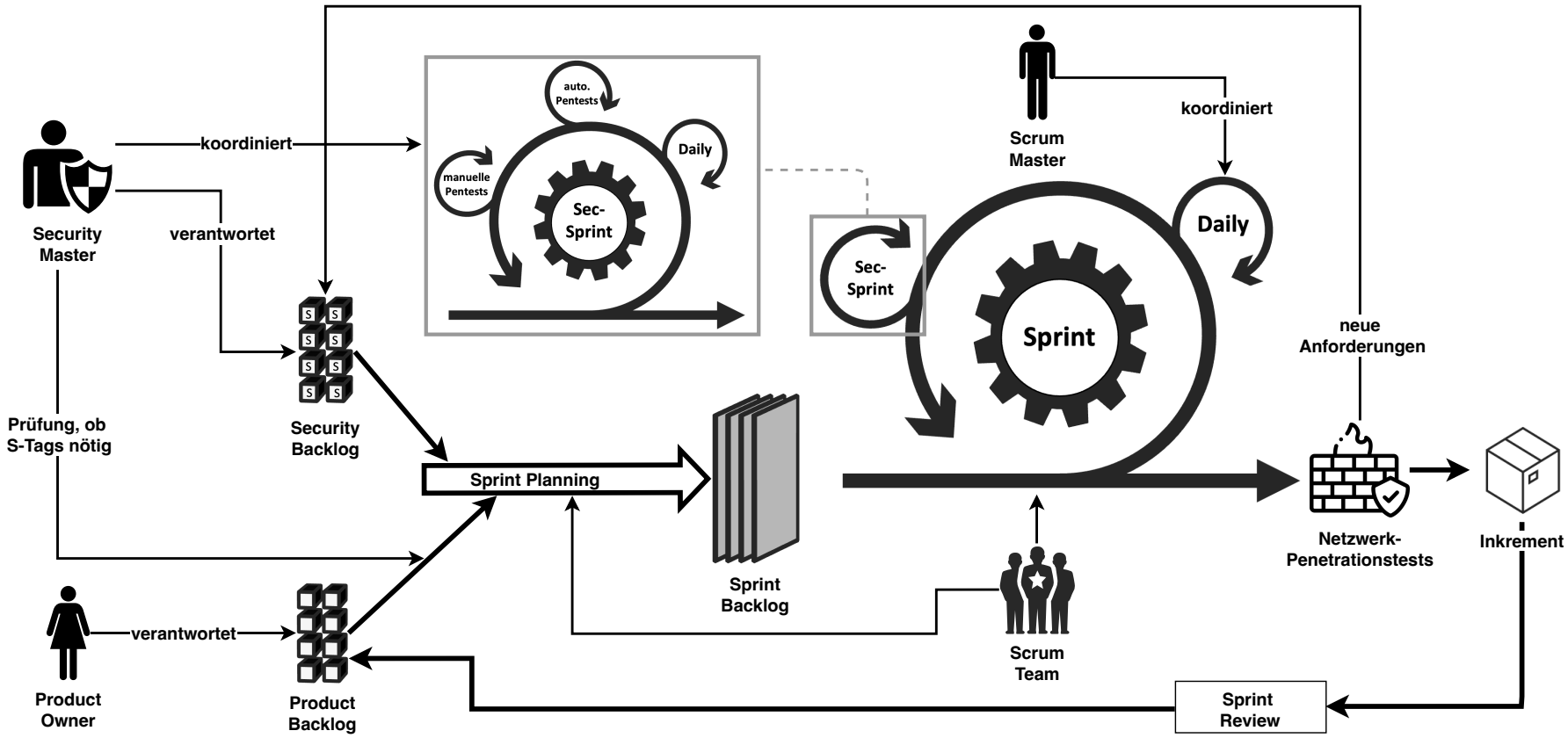

Abbildung 3: VS-Scrum-Framework

Im abschließenden Kapitel 4 wird aus den erläuterten Ergebnissen ein Fazit gezogen sowie ein Ausblick auf zukünftige Forschungsmöglichkeiten gegeben.

4 Fazit und Ausblick

Das Ziel dieser Arbeit ist es, ein Framework zu entwickeln, welches die IT-Sicherheit bei einem Scrum Ansatz berücksichtigt. Dieses Kapitel beginnt mit einer Zusammenfassung der betrachteten Thematiken, welche darüber hinaus kritisch betrachtet werden. Daraufhin wird der Mehrwert für die Praxis und Wissenschaft vorgestellt. Zum Schluss erfolgt ein Ausblick über die mögliche Fortführung der Forschung.

4.1 Zusammenfassung

Die Angriffe auf die IT-Infrastruktur von Unternehmen nimmt stetig zu (vgl. Kapitel 1). Aus diesem Grund ist es wichtig, bereits bei der Entwicklung von Software auf die Anforderungen der IT-Sicherheit zu achten. Als Vorgehensmodell für die Entwicklung von Software gewinnt das Scrum-Framework an immer größerem Interesse. Durch das iterative Vorgehen entstehen allerdings Limitationen in Bezug auf die IT-Sicherheit (vgl. Kapitel 1.1). Deshalb setzt sich diese Arbeit als Ziel, ein Framework zu entwickeln, welches die verschiedenen Anforderungen der IT-Sicherheit in einem Scrum Ansatz berücksichtigt.

Die anschließende Literaturarbeit wird nach dem Modell von vom Brocke et al. [Vom+09] durchgeführt. Dafür erfolgt im ersten Schritt die Definition des Suchraums mit Hilfe der Taxonomie von Cooper [Coo88]. Der Fokus der Literatur liegt dabei auf den Forschungsergebnissen und Praxis/Anwendungen. Das Ziel ist integrierend und als Perspektive wird die neutrale Repräsentation gewählt. Die Abdeckung erfolgt repräsentativ, die Organisation konzeptuell und die Zielgruppe umfasst allgemeine Wissenschaftler sowie Praktiker.

Die Konzeptualisierung zeigt, das Vorgehensmodell Scrum arbeitet die Anforderungen an ein System iterativ ab. Dabei unterstützt die Etablierung von verschiedenen Rollen wie beispielsweise dem Scrum Master, der für die Einhaltung der Scrum-Praktiken verantwortlich ist. Die IT-Sicherheit selbst verfolgt die Schutzziele Vertraulichkeit, Integrität und Verfügbarkeit.

Die Literatursuche wird nach den Prinzipien von Webster und Watson [WW02]. durchgeführt. Im ersten Schritt werden die führenden Fachzeitschriften durchsucht, um im Anschluss daran eine Vorwärts- und Rückwärtssuche bei der gegebenen Literatur durchzuführen. Anschließend wird geprüft, ob die Modelle in der Literatur ähnliche Konzepte verfolgen.

Das Ergebnis sind fünf Konzepte, die als Grundlage für die Entwicklung des Frameworks genutzt werden. Die Abbildung 3 zeigt das VS-Scrum-Framework, welches die IT-Sicherheit in einem Scrum Ansatz berücksichtigt.

4.2 Kritische Betrachtung

Durch die repräsentative Abdeckung wurde nicht die Gesamtheit der Literatur betrachtet, die sich mit der Thematik IT-Sicherheit im Scrum Framework auseindersetzt. Des Weiteren wären Experteninterviews mit Praktikern interessant gewesen, um die Differenzen zwischen der Literatur und der Praxis aufzudecken. Der letzte kritische Punkt stellt die Tatsache dar, dass es sich beim entwickelten Modell um ein rein theoretisches Modell handelt, welches in noch keiner Form validiert wurde.

4.3 Mehrwert für Praxis und Wissenschaft

Der Mehrwert für die Wissenschaft entsteht durch die Konsolidierung der bereits durchgeführten Forschung zur Thematik. Anhand der tabellarischen Auflistung der Ergebnisse kann eine weiterführende Forschung betrieben und die Tabelle entsprechend ergänzt werden.

Der Mehrwert für die Praxis entsteht durch die Erstellung einer Übersicht bereits vorhandener Modelle sowie die Entwicklung des VS-Scrum Frameworks. Des Weiteren werden verschiedene Möglichkeiten aufgezeigt, wie die IT-Sicherheit im agilen Vorgehensmodell Scrum berücksichtigt werden kann.

4.4 Ausblick

In künftigen Forschungsarbeiten könnte zum einen die repräsentative Abdeckung erweitert werden, um zusätzliche Konzepte zu finden. Zum anderen sollte das vorgestellte VS-Scrum Framework durch beispielsweise Fallstudien validiert werden. Ein weiterer Mehrwert könnte durch die Befragung von Experten und Praktikern entstehen, die bereits eigene Erfahrungen mit der sicheren, agilen Softwareentwicklung haben. Es ist wichtig, dass Unternehmen auch bei der Nutzung von agilen Methoden auf die IT-Sicherheit achten, denn

> „[i]t takes 20 years to build a reputation and few minutes of cyber-incident to ruin it.“
> – Stephane Nappo (Global CISO of the year 2018)

Literaturverzeichnis

[AN16] S. Hassan Adelyar und Alex Norta. „Towards a Secure Agile Software Development Process“. In: *2016 10th International Conference on the Quality of Information and Communications Technology (QUATIC)*. IEEE, 11. Sep. 2016, S. 101–106. ISBN: 978-1-5090-3581-6. DOI: 10.1109/QUATIC.2016.028. URL: http://ieeexplore.ieee.org/document/7814525/ (besucht am 13. 01. 2022).

[AGI11] Zulkarnain Azham, Imran Ghani und Norafida Ithnin. „Security Backlog in Scrum Security Practices“. In: *2011 5th Malaysian Conference in Software Engineering, MySEC 2011*. 2011, S. 414–417. ISBN: 978-1-4577-1531-0. DOI: 10.1109/MySEC.2011.6140708.

[BS21] Achim Berg und Sinan Selen. *Wirtschaftsschutz 2021*. Berlin: Bitkom e.V., 5. Aug. 2021, S. 19.

[Bun21] Bundesamt für Sicherheit in der Informationstechnik (BSI). *Die Lage Der IT-Sicherheit in Deutschland 2021*. Bonn: Bundesamt für Sicherheit in der Informationstechnik (BSI), 2021, S. 100.

[Coo88] Harris M. Cooper. „Organizing Knowledge Syntheses: A Taxonomy of Literature Reviews“. In: *Knowledge in Society* 1.1 (März 1988), S. 104–126. ISSN: 0897-1986. DOI: 10.1007/BF03177550. URL: http://link.springer.com/10.1007/BF03177550 (besucht am 23. 02. 2021).

[GAJ14] Imran Ghani, Zulkarnain Azham und Seung Ryul Jeong. „Integrating Software Security into Agile-Scrum Method“. In: *KSII Transactions on Internet and Information Systems (TIIS)* 8.2 (2014), S. 646–663. ISSN: 1976-7277. DOI: 10.3837/TIIS.2014.02.0019. URL: http://dx.doi.org/10.3837/tiis.2014.02.0019 (besucht am 12. 01. 2022).

[Git21] GitLab. *Software Development Methods Worldwide 2021*. GitLab, Mai 2021.

[Lee20] Daniel Christian Leeser. *Digitalisierung in KMU Kompakt*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2020. 202 S. ISBN: 978-3-662-59737-8. DOI: 10.1007/978-3-662-59738-5. URL: http://link.springer.com/10.1007/978-3-662-59738-5 (besucht am 27. 01. 2022).

[MMB17] Patrik Maier, Zhendong Ma und Roderick Bloem. „Towards a Secure SCRUM Process for Agile Web Application Development“. In: *Proceedings of the 12th International Conference on Availability, Reliability and Security* 8 (2017). DOI: 10.1145/3098954.

[PH15] Christoph Pohl und Hans-Joachim Hof. „Secure Scrum: Development of Secure Software with Scrum“. In: *CoRR* abs/1507.0 (2015). arXiv: 1507.02992. URL: http://arxiv.org/abs/1507.02992.

[Raf+15] Usman Rafi u. a. „US-Scrum: A Methodology for Developing Software with Enhanced Correctness, Usability and Security“. In: *International Journal of Scientific & Engineering Research* 6.9 (2015), S. 377–383. ISSN: 2229-5518. URL: http://www.ijser.org (besucht am 13. 01. 2022).

[RM16] Nagy Ramadan und Ihab Mohamed. „A Security Testing Framework for Scrum Based Projects“. In: *International Journal of Computer Applications* 138.7 (17. März 2016), S. 12–17. DOI: 10.5120/ijca2016908928.

[RHL16] Kalle Rindell, Sami Hyrynsalmi und Ville Leppänen. „Case Study of Security Development in an Agile Environment: Building Identity Management for a Government Agency“. In: *Proceedings - 2016 11th International Conference on Availability, Reliability and Security, ARES 2016*. Institute of Electrical and Electronics Engineers Inc., 14. Dez. 2016, S. 556–563. ISBN: 978-1-5090-0990-9. DOI: 10.1109/ARES.2016.45.

[Ros+20] Scott Rose u. a. *Zero Trust Architecture*. Gaithersburg, MD: National Institute of Standards and Technology, 11. Aug. 2020, S. 49. DOI: 10.6028/NIST.SP.800-207. URL: https://doi.org/10.6028/NIST.SP.800-207 (besucht am 06. 02. 2022).

[RJ18] Hanne Rygge und Audun Jøsang. „Threat Poker: Solving Security and Privacy Threats in Agile Software Development“. In: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Bd. 11252 LNCS. Springer, Cham, 28. Nov. 2018, S. 468–483. ISBN: 978-3-030-03637-9. DOI: 10.1007/978-3-030-03638-6_29. URL: http://link.springer.com/10.1007/978-3-030-03638-6_29 (besucht am 13. 01. 2022).

[SS75] J.H. Saltzer und M.D. Schroeder. „The Protection of Information in Computer Systems“. In: *Proceedings of the IEEE* 63.9 (1975), S. 1278–1308. ISSN: 0018-9219. DOI: 10.1109/PROC.1975.9939. URL: http://ieeexplore.ieee.org/document/1451869/ (besucht am 06. 02. 2022).

[SS20] Ken Schwaber und Jeff Sutherland. „Der Scrum Guide - Der Gültige Leitfaden Für Scrum: Die Spielregeln“. In: *Scrumguides.Org* (November 2020), S. 22.

[TK15] Martin Tomanek und Tomas Klima. „Penetration Testing in Agile Software Development Projects“. In: *International Journal on Cryptography and Information Security* 5.1 (2015), S. 01–07. ISSN: 1839-8626. DOI: 10.5121/ijcis.2015.5101.

[Vom+09] Jan Vom Brocke u. a. „Reconstructing the Giant: On the Importance of Rigour in Documenting the Literature Search Process“. In: *17th European Conference on Information Systems, ECIS 2009*. 2009.

[WW02] Jane Webster und Richard T Watson. „Analyzing the Past to Prepare for the Future: Writing a Literature Review.“ In: *MIS Quarterly* 26.2 (2002), S. xiii–xxiii.

[Wic08] Dave Wichers. *Breaking the Waterfall Mindset of the Security Industry*. Open Web Application Security Project, 2008, S. 19.

Internetquellen

[Pro19] Dominik Probst. *Scrum Projektmanagement: Agil Trotz Fester Struktur*. Axonic Informationssysteme GmbH. 28. Juni 2019. URL: https://zenkit.com/de/blog/uebersicht-scrum-projektmanagement/ (besucht am 11. 02. 2022).

[Rie17] Silvio Riener. *Secure Agile Development: Secure Scrum vs. Security Backlog*. Orange Cyberdefense Germany GmbH. 24. Jan. 2017. URL: https://www.cubespotter.de/cubespotter/secure-agile-development-secure-scrum-vs-security-backlog/ (besucht am 06. 02. 2022).