

Sicherheitsaspekte in der (agilen) Softwareentwicklung

FINN HOEDT*, MAURICE PUTZGER*, and BASTIAN WECKE*, Hochschule für Technik, Wirtschaft und Kultur Leipzig (HTWK Leipzig), Deutschland

(Abstract-Länge ist typischerweise 15-25 Zeilen lang, in der PDF-Darstellung)

1 EINLEITUNG UND MOTIVATION

(Beschreibung von Kontext, Problemen, Anforderungen und Zielen)
(kurze Zusammenfassung der Struktur der Belegarbeit)

Diese Arbeit ist folgendermaßen strukturiert. In Kapitel
Abschließend ...

2 GRUNDLAGEN

(Begriffe und Definitionen)

3 (HAUPTTEIL MIT GGF. MEHREREN SECTIONS)

(der Hauptteil umfasst typischerweise ca. 2/3 bis 3/4 des Texts der Arbeit.)

3.1 Scrum4Safety

Scrum4Safety verfolgt das Ziel, die Innovationsfähigkeit und Effizienz von agilen Methoden mit den strengen Anforderungen an Sicherheit und Konformität zu kombinieren. Dafür führt es spezifische Rollen, Prinzipien und Workflows ein, um die Qualität und Sicherheit der zu entwickelnden Software sicherzustellen. Das Framework erweitert die klassischen Scrum-Rollen um zusätzliche Rollen, die in sicherheitskritischen Projekten notwendig sind. Dabei handelt es sich um Verifier, Validator und Assessor. Diese Rollen sind für den Quality-Assurance-Prozess verantwortlich und überprüfen die Einhaltung der Sicherheitsanforderungen.

Neben den zusätzlichen Rollen führt Scrum4Safety auch neue Konzepte ein, die die Sicherheit der Softwareentwicklung gewährleisten sollen. Dazu gehört das Sprint-Hardening, welches sicherstellen soll, dass am Ende jeder Iteration eine validierte Softwareversion bereitgestellt werden kann. Dafür wird in jeder Iteration darauf geachtet, dass Benutzerdokumentationen und Konformitätsnachweise erstellt werden, die dann für die externe Softwarebewertung verwendet werden können. Ein weiteres Konzept ist die Continuous Compliance. Das beinhaltet, dass kontinuierlich Verifizierungs- und Validierungsaktivitäten durchgeführt werden, sodass die Software jederzeit konform ist. Dadurch sollen kritische Fehler frühzeitig erkannt werden, um sie schnellstmöglich beheben zu können. Das dritte Konzept ist die Living Traceability. Durch die Einführung einer klaren Rückverfolgbarkeit in jedem Prozess bei der Umsetzung von Benutzeranforderungen soll die Zertifizierung der Software durch externe Prüfstellen erleichtert werden. [Andriadi et al. [n. d.]]

Der zentrale Prozess in Scrum4Safety ist der sogenannte Safe-Sprint. Dabei handelt es sich wie beim klassischen Scrum-Prinzip um eine zeitlich begrenzte Iteration, in der ein neues Software-Inkrement entwickelt wird. Im Gegensatz zum klassischen Scrum

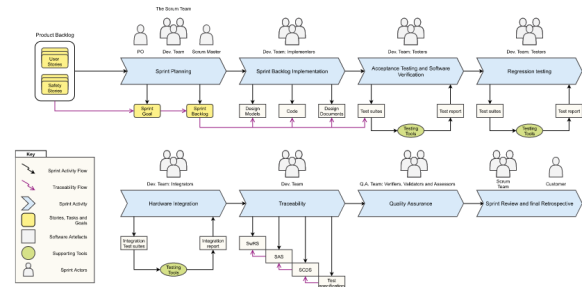


Abb. 1. Scrum4Safety Prozess

soll das entwickelte Inkrement aber durch die eingeführten Konzepte sicherheitstechnisch validiert sein. Dafür setzt Scrum4Safety mit dem Safe-Sprint schon in der Planning-Phase an und integriert Safety-Stories neben den klassischen User-Stories in den Backlog. Dadurch soll gewährleistet sein, dass Sicherheit bereits in der Anforderungsphase berücksichtigt wird. Auch eine frühe Einbindung von Sicherheitsüberprüfungen und Testphasen im Ablauf des Safe-Sprints soll sicherstellen, dass Sicherheitsmaßnahmen frühzeitig in den Entwicklungsprozess integriert werden.

4 DISKUSSION

(Einordnung, Interpretation und Bewertung der Erkenntnisse – (nachvollziehbare, begründbare) Meinungen sind erlaubt)

5 ZUSAMMENFASSUNG UND AUSBLICK

(Überblick über die gesamte Arbeit, Rückführung auf Aussagen aus Kapitel 1 durchführen, offene Punkte als neue Forschungsfragen definieren)

LITERATUR

Kus Andriadi, Haryono Soeparno, Ford Lumban Gaol, and Yulyani Arifin. [n. d.]. The Impact of Shift-Left Testing to Software Quality in Agile Methodology: A Case Study. In *2023 International Conference on Information Management and Technology (ICIM-Tech)* (2023-08). 259–264. <https://doi.org/10.1109/ICIMTech59029.2023.10277919> ISSN: 2837-2778.

A ANHANG 1

* Alle Studierenden trugen zu gleichen Teilen zu dieser Arbeit bei.