Institut für Softwaretechnologie
Abteilung Software Engineering

Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Bachelorarbeit

# Dokumentation von Sicherheitsanforderungen beim agilen Requirements Engineering

Marc Voigt

| | |
|---|---|
| **Studiengang:** | Softwaretechnik |
| **Prüfer/in:** | Prof. Dr. Stefan Wagner |
| **Betreuer/in:** | Yang Wang,<br>Dr. Ivan Bogicevic |
| **Beginn am:** | 01. Dezember 2016 |
| **Beendet am:** | 31. Mai 2017 |
| **CR-Nummer:** | D 2.1, K 4.1, K 6.3 |

# Kurzfassung

Das Ziel der vorliegenden Bachelorarbeit ist es, die Umsetzung der Dokumentation von Anforderungen sicherheitskritischer Systeme in agiler Entwicklungsumgebung zu untersuchen. Anhand eines Studienprojekts und der Norm IEC 61508 wird überprüft, ob alle Anforderungen der Norm entsprechen. Es werden Lösungsvorschläge erarbeitet um die Umsetzung vollständig zu realisieren. Diese Lösungen werden am Ende durch zwei Interviewpartner aus der Industrie evaluiert. Ziel dieser Arbeit ist es Vorschläge zur Umsetzung agiler Methoden zu geben.

# Inhaltsverzeichnis

# Tabellenverzeichnis

# 1 Einleitung

In diesem Kapitel wird die Motivation und die Aufgabenstellung dieser Arbeit vorgestellt.

## 1.1 Motivation

Der digitale Wandel ist in vollem Gange. Mit statischem Denken ist es nicht möglich sich in der immer schneller werdenden Welt durchzusetzen. Jeder weiß welche Vorteile eine agile Entwicklung mit sich bringt, aber warum wird sie zum Beispiel in sicherheitskritischen Systemen so selten verwendet? Das Ziel dieser Arbeit ist es, herauszufinden ob sich sicherheitskritische Systeme und die agile Entwicklung vereinbaren lassen. Welche Herausforderungen bestehen und warum scheitert die Umsetzung gegebenenfalls. Kann man auf bisher bewährte Dokumente verzichten und diese ersetzen?

## 1.2 Aufgabenstellung

Die Aufgabe dieser Arbeit ist es mit agilen Methoden ein sicherheitskritisches System zu entwickeln. Die Dokumentation von Sicherheitsanforderungen wird anhand der Norm IEC 61508 überprüft. Da die Norm nicht für agile Entwicklungsmethoden geschrieben ist, ist die Herausforderung die Anforderungen darauf zu transferieren. Alle Anforderungen, die für die Dokumentation relevant sind, müssen untersucht und geprüft werden. Für Sicherheitsanforderungen, die das Projekt Smart Home nicht erfüllen, müssen Lösungsvorschläge gefunden werden, die anschließend umgesetzt werden.

## 1.3 Überblick über die Arbeit

Die folgende Arbeit gliedert sich in die Schwerpunkte Studienprojekt Smart Home, Norm IEC 61508 und die Zusammenführung dieser beiden Teile. Sie ist dabei in folgende Kapitel unterteilt:

- **Kapitel 2 - Verwandte Arbeiten** umfassen die Ergebnisse der Literaturrecherche.
- **Kapitel 3 - Hintergrund** über die Norm und das Projekt Smart Home.

- **Kapitel 4 - Walkthrough** über die Norm.
- **Kapitel 5 - Empfehlungen** für das Studienprojekt Smart Home anhand der Ergebnisse des Walkthroughs.
- **Kapitel 6 - Umsetzung in Smart Home** der Empfehlungen.
- **Kapitel 7 - Validierung Interview** mit zwei Partnern aus der Industrie.
- **Kapitel 8 - Diskussion** und Fazit der Arbeit.
- **Kapitel 9 - Ausblick** auf die Zukunft von sicherheitskritischen Systemen in agiler Entwicklungsumgebung.

# 2 Verwandte Arbeiten

In diesem Kapitel werden fünf Arbeiten vorgestellt, die sehr nah an dieser Arbeit angrenzen.

Die Arbeit „Scrum, documentation and the IEC 61508-3: 2010 software standard“ von Thor Myklebust und weiteren befasst sich genau mit dem Thema dieser Arbeit. Es werden Methoden vorgestellt, wie zum Beispiel SafeScrum um die Norm IEC 61508-3 zu erfüllen. Auch diese Arbeit macht einen Walkthrough durch die Norm, allerdings mit einem anderen Hintergrund. Das Paper befasst sich nur mit Dokumenten und deren Einsparung im agilen Prozess. Vernachlässigt werden agile Methoden zur Kommunikation, was eine erhebliche Schwäche dieser Arbeit aufzeigt.[MSH+14]

Das Dokumentationsmuster „How much is just enough?“ von Rashina Hoda, James Nobel und Stuart Marshall befasst sich mit der mangelnden Dokumentation bei agilen Projekten. Diese Arbeit stellt Muster zur Verfügung um Dokumente agil umzusetzen. Sogenannte Fake Dokumentation wird nur in Absprache erstellt und nur dann wenn es wirklich nötig ist. Durch Time Stamps werden Änderungen in einem Wiki dokumentiert und sind zu jeder Entwicklungszeit nachvollziehbar. Das E-Backup sichert Papier Artefakte elektronisch und macht es möglich jederzeit darauf zurück zu greifen.

Diese Methoden ähneln leider sehr der herkömmlichen Dokumentation. Das agile Konzept wird somit eingebremst und es entsteht eine Art agiler Scheinprozess. Auch hier ist das Problem, dass Kommunikation nicht als Lösung genutzt wird und der Weg weg von den Dokumenten vermieden wird.[HNM10]

Das Paper „Adaptive Software Development for Developing Safety Critical Software“ von Adil A Abdelaziz, Yaseen El-Tahir und Raheeg Osman stellt ein Modell vor um die Softwareentwicklung bei sicherheitskritischen Systemen agiler zu machen. Das Modell gliedert sich in verschiedene Phasen und soll an die agile Methode herankommen. Allerdings wurde die Methode nicht in einer reellen Umgebung getestet und ist somit fraglich. Der agile Teil ist weiterhin sehr eingeschränkt.[ATO15]

Der Buchausschnitt „The Role of Communication in Agile Systems Development“ von Markus Hummel analysiert die Einflüsse von agilen Praktiken auf die Kommunikation in Projekten. Sie kommen zu dem Ergebnis, dass die Team Verteilung, die Teamgröße und der Projektbereich eine große Rolle spielen. Für all diese Probleme erstellen sie Lösungen, die vor allem auf der formellen und informellen Kommunikation basieren. Dieser Artikel zeigt die Rolle der Kommunikation in agilen Projekten deutlich auf, jedoch bezieht sie sich nicht auf sicherheitskritische Systeme. Diese Ergebnisse lassen sich allerdings problemlos darauf anwenden.[HRH08]

Der Artikel „The impact of agile practices on communication in software development“ aus dem Empirical Software Engineering Buch befasst sich noch detaillierter mit Kommunikationsmethoden. Es werden verschiedene Kommunikationsarten vorgestellt, wie zum Beispiel die informelle und formelle Kommunikation. Danach werden zwei Projekte untersucht in denen verschiedene Praktiken angewandt werden. Positive sowie negative Auswirkungen spielen hierbei eine große Rolle. Ziel dieser Studie ist es, die Dokumentation durch Kommunikation zu ersetzen. Diese Studie zeigt, wie wichtig die Kommunikation in agilen Projekten ist und wie viel Dokumentation dadurch gespart werden kann. Dieser Artikel lässt sich sehr gut in diese Arbeit integrieren, da fehlende Dokumentation durch Kommunikation aufgefangen werden kann und auch in sicherheitskritischer Software sinnvoll ist. [PHS+08]

# 3 Hintergrund

In diesem Kapitel werden die Norm und das Studienprojekt vorgestellt.

## 3.1 Norm IEC 61508-3

Die Norm IEC 61508-3 ist eine internationale Norm für die funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer Systeme. Der dritte Teil umfasst die Anforderungen an die Software, bei der uns aber nur der relevante Teil für die Dokumentation interessiert. Das Ziel der Norm ist es, eine Software zu entwickeln, die nach aktuellem Stand der Technik keine unverhältnismäßigen oder unvertretbaren Gefahren für Anwender und Umwelt bedeuten. Als Grundlage wird das Lebenszyklus-Modell verwendet, dass heißt die Software wird von der ersten Planungsstufe bis hin zu seiner Außerbetriebnahme und Entsorgung betrachtet.[Wik17]

Teil 3 der Norm gibt zusätzliche Anforderungen für das Management von sicherheitskritischer Software an und geht dann detailliert auf den Software-Lebenszyklus ein. Alle Anforderungen, die für die Dokumentation relevant sind werden in Kapitel 4 detailliert aufgelistet und analysiert.[IEC10]

Da die Anforderungen alle für das V-Modell zugeschnitten sind, besteht nun die Aufgabe alle Anforderungen in dem Scrum-Prozess umzusetzen und Lösungen zu finden. Bisherige Formen der Dokumentation im Projekt Smart Home sollen überprüft und gegebenenfalls modifiziert werden. Am Ende soll die Frage geklärt sein, ob die Norm IEC 61508 mit der agilen Methode Scrum umsetzbar und erfüllbar ist.

## 3.2 Dokumentation in Smart Home

Die Umsetzung in Smart Home hat sich im Laufe der Zeit verändert. Es wird die Umsetzung ab Sprint 10 analysiert. Es werden ausschließlich die Sicherheitsaspekte im Projekt betrachtet.

### 3.2.1 Safety Backlog

Die Umsetzung des Product Backlog erfolgt in JIRA [atl], einer Projektmanagement Software. Die Sicherheitsanforderungen werden durch Safety Epics von den funktionalen Anforderungen getrennt.

Die Sicherheitsanforderungen werden von dem externen Safety Expert erstellt. Die einzelnen daraus resultierenden Aufgaben erstellt der interne Safety Expert in Absprache mit dem gesamten Team. [WRW17]

### 3.2.2 Sicherheitsanalysen und Dokumentation

Die Sicherheitsanalysen umfassen alle Dokumente in dem Git Repository [Stu16]. Dazu gehören der Safety Plan, der alle Maßnahmen und Durchführungen des gesamten Projekts spezifiziert. Die Umsetzung des Safety Plans ist die Grundlage für die Entwicklung des Systems. Die Sicherheitsziele und Anforderungen sind ein weiterer Teil und werden durch zwei System Theoretic Process Analysis (STPA) ergänzt.

### 3.2.3 Sprint Meetings

Das Team führt folgende Meetings in dem Projekt durch:

- **Pre Planning Meeting**

  Im Pre Planning Meeting werden vorab die Eigenschaften und Abhängigkeiten von Safety User Stories zu den funktionalen User Stories besprochen. Dieses Meeting findet immer unmittelbar vor jedem Sprint Planning Meeting statt.

  Diese Analyse übernehmen der externe Safety Expert und der interne Safety Expert. Die Ergebnisse werden dann im kompletten Entwicklerteam besprochen, so dass jede Sicherheitsanforderung klar und verständlich ist. [WRW17]

- **Sprint Planning Meeting**

  Das Sprint Planning Meeting unterscheidet sich nicht von dem normalen Scrum Prozess. Da die Safety User Stories bereits im Pre-Planning Meeting spezifiziert und diskutiert wurden, werden diese nun für den Sprint Backlog ausgewählt.

  Das Entwicklerteam und der Scrum Master nehmen an diesem Meeting teil. Der externe Safety Expert spielt dabei keine Rolle mehr. [WRW17]

- **Sprint Retrospectiv Meeting**

  Die Sprint Retrospektive ist bei der Entwicklung von sicherheitskritischer Software wichtiger als bei einer normalen Software. Zu der normalen Retrospektive kommt in Smart Home noch die Frage, wie man die Sicherheit optimieren kann, zum Beispiel durch bessere Kommunikation oder neuen Abläufen.

  Die Organisation der Sprint Retrospektive übernimmt der Scrum Master.

### 3.2.4 User Stories

Die User Stories werden mit dem Safety oder Safety-Epic Epic versehen. User Stories mit Sicherheitsanforderungen referenzieren über Verknüpfungen die nötigen funktionalen Anforderungen.

User Stories haben ein konsistentes Format der folgenden Form: „Um <eine Funktion> zu erfüllen, muss das System <etwas erfüllen oder vermeiden>“. Es wurde allerdings keine Priorisierung der User Stories durchgeführt. Die Aufwandsschätzung im jeweiligen Sprint ist unvollständig und nicht konsistent.

Verantwortlich für die sicherheitsbezogenen User Stories ist der externe Safety Expert.

# 4 Walkthrough

In diesem Abschnitt werden die Anforderungen aus IEC 61508-3 auf das Projekt Smart Home angewandt. Es werden alle Anforderungen berücksichtigt, die dokumentiert werden müssen. Die Wertungen bestehen aus „OK“ und „NICHT OK“. Für Wertungen mit „NICHT OK“ werden Lösungsvorschläge angegeben.

Die detaillierte Umsetzung und erstellte Verbesserungsmöglichkeiten werden in Kapitel 5 und 6 ausführlich erläutert. Die folgende Tabelle arbeitet chronologisch alle Anforderungen ab:

| Anforderung | Beschreibung | Wertung | Umsetzung | Probleme | Lösungen |
|---|---|---|---|---|---|
| **Allgemein** | | | | | |
| 7.1.2.8 | Jede Aktivität im Softwarelebenszyklus muss dokumentiert werden | siehe Unterpunkte | siehe Unterpunkte | siehe Unterpunkte | siehe Unterpunkte |
| **Spezifikation** | | | | | |
| 7.2.2.1 | Anforderungen die bereits für das sicherheitsrelevante System spezifiziert sind, müssen nicht mehr wiederholt werden | Es existieren keine bereits spezifizierte Anforderungen | - | - | - |

| | | | | | |
|---|---|---|---|---|---|
| 7.2.2.2 | Anforderungsspezifikation ergibt sich aus den Sicherheitsanforderungen des Systems und der Anforderung des Safety Plannings | NICHT OK | Product Backlog, Sicherheitsanalysen | User Stories geben keinen Bezug auf die Sicherheitsanalysen | Safety Story |
| 7.2.2.3 | Anforderungsspezifikation muss hinreichend detailliert sein, um den Entwurf und die Implementierung der erforderlichen Sicherheitsintegrität zu ermöglichen | NICHT OK | Sprint Backlog | Bezug zu Safety Plan fehlt | Informelle Kommunikation, Safety Story, INVEST |
| 7.2.2.4 | Durchführung einer Ursachenfehleranalyse und eventuell zu treffende Maßnahmen | OK | STPA1 und STPA2 | - | - |

| | | | | | |
|---|---|---|---|---|---|
| 7.2.2.5 | 7.2.2.2 auswerten um sicherzustelen, dass die Anforderungen hinreichend spezifiziert werden | NICHT OK | Sprint Planning Meeting | Keine Dokumentation der Meetings vorhanden | Informelle Kommunikation, Snapshots |
| 7.2.2.7 | Safety Anforderungsspezifikation spezifiziert und dokumentiert jede Software bezogene oder relevante Abhängigkeit zwischen Software und Hardware | OK | User Stories, Sicherheitsanforderungsdokumentation | - | - |
| 7.2.2.10 | Spezifikation der Sicherheitsanforderungen muss die erforderlichen Eigenschaften des Produkts ausdrücken | OK | Sicherheitsanforderungsdokumentation | - | - |
| **Validierungsplan** | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| 7.3.2.2 | Software Validierungsplan | NICHT OK | Sprint Planning Meeting | Keine Dokumentation der Meetings, Keine Akzeptanzkriterien für jede User Story definiert | Informelle Kommunikation, Akzeptanzkriterien, Snapshots |
| **Software Entwurf und Umsetzung** | | | | | |
| 7.4.2.5 | Entwurf basiert auf einer Notation, die eindeutig definiert ist | OK | Klassendiagramme, Screenshots, Blockdiagramme, Entwurfsdokument, Story Map | - | - |
| 7.4.2.12 | Wiederverwendbarkeit von Software | NICHT OK | User Stories im Sprint Backlog | User Stories nicht mehr greifbar, wenn sie abgeschlossen sind | Time Stamp, Wiki |
| **Architekturentwurf** | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| 7.4.3.2 | Entwurf der Softwarearchitektur wird detailliert beschrieben | OK | Architekturdiagramm | - | - |
| 7.4.3.3 | Änderungen an der Spezifikation sind nach Durchführung von 7.4.3.2 zu dokumentieren | NICHT OK | Neue User Story | Nicht nachvollziehbar welche User Story geändert wurde | Safety Product Backlog, Time Stamp |
| **Werkzeuge und Programmiersprachen** | | | | | |
| 7.4.4.4 | Alle off-line Werkzeuge müssen eine Spezifikation oder Produktdokumentation haben | NICHT OK | - | - | Informelle Kommunikation |
| 7.4.4.7 | Werkzeugvalidierung dokumentieren | NICHT OK | - | - | Informelle Kommunikation |

| | | | | | |
|---|---|---|---|---|---|
| **Detaillierter Entwurf und Entwicklung** | | | | | |
| 7.4.5.4 | Verfeinerung des Softwareentwurfs in Module | NICHT OK | User Stories im Sprint Backlog | Nicht nach INVEST Prinzip, User Stories geben keinen Bezug auf die Sicherheitsanalysen | Informelle Kommunikation, INVEST, Safety Stories |
| 7.4.5.5 | Festlegung geeigneter Softwareintegrationstests | NICHT OK | Sprint Planning Meeting | Keine Dokumentation vorhanden | Snapshots, Akzeptanzkriterien |
| **Modultests** | | | | | |
| 7.4.7.3 | Modultest Ergebnisse dokumentieren | NICHT OK | - | - | Informelle Kommunikation, STDD |
| **Integrationstests** | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| 7.4.8.1 | Integrationstests müssen spezifiziert werden | NICHT OK | Sprint Planning Meeting | Keine Dokumentation vorhanden | Snapshots, Akzeptanzkriterien |
| 7.4.8.2 | Spezifikation der Integrationstests | NICHT OK | Backlog | Keine Akzeptanzkriterien | Akzeptanzkriterien |
| 7.4.8.4 | Ergebnisse der Integrationstests müssen dokumentiert werden | NICHT OK | Sprint Review Meeting | Keine Dokumentation vorhanden | Informelle Kommunikation,STDD |
| **Hardware und Software Integration** | | | | | |
| 7.5.2.1 | Integrationstests müssen während der Entwurfs- und Entwicklungsphase spezifiziert werden | NICHT OK | Sprint Planning Meeting | Keine Dokumentation vorhanden | Informelle Kommunikation, Snapshots, Referenzen in User Story |
| 7.5.2.2 | Software / Hardware Integrationsspezifikation | OK | Product Backlog | - | - |

| | | | | | |
|---|---|---|---|---|---|
| 7.5.2.7 | Testfälle und Ergebnisse müssen dokumentiert werden | NICHT OK | Sprint Review | Keine Dokumentation vorhanden | Informelle Kommunikation, Snapshots, Protokolle |
| 7.5.2.8 | Testergebnisse und ob Kriterien erfüllt wurden muss dokumentiert werden. Bei Ausfall sind Gründe zu dokumentieren | NICHT OK | Sprint Review | Keine Dokumentation vorhanden | Informelle Kommunikation, Snapshots, Protokolle |
| **System-Sicherheits-Validierung** | | | | | |
| 7.7.2.4 | Validierungsergebnisse müssen dokumentiert sein | NICHT OK | Sprint Review | Keine Dokumentation vorhanden, neue User Stories werden erstellt ohne Bezug auf die geänderte User Story | Snapshots, Protokolle, Time Stamp |

| | | | | | |
|---|---|---|---|---|---|
| 7.7.2.5 | Für jede Sicherheitsfunktion müssen die Ergebnisse der Validierung in genauer Form dokumentiert werden | NICHT OK | Sprint Review | Keine Dokumentation vorhanden, neue User Stories werden erstellt ohne Bezug auf die geänderte User Story | Snapshots, Protokolle, Time Stamp |
| 7.7.2.9 | Ergebnisse der Validierung müssen dokumentiert werden | NICHT OK | Sprint Review | Keine Dokumentation vorhanden, neue User Stories werden erstellt ohne Bezug auf die geänderte User Story | Snapshots, Protokolle, Time Stamp |
| **Software Modifikation** | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| 7.8.2.4 | Analyse über Auswirkungen der vorgeschlagenen Änderungen auf funktionale Sicherheit (7.8.2.3). Ergebnisse müssen dokumentiert werden | NICHT OK | - | Änderungen sind nicht nachvollziehbar, da einfach eine neue User Story angelegt wird | Informelle Kommunikation, Time Stamp, Safety Story |
| 7.8.2.8 | Einzelheiten aller Änderungen sind zu dokumentieren | NICHT OK | - | Änderungen sind nicht nachvollziehbar, da einfach eine neue User Story angelegt wird | Time Stamp, Safety Story |
| 7.8.2.9 | Angaben über die Einzelheiten aller Änderungen sind zu dokumentieren | NICHT OK | - | Änderungen sind nicht nachvollziehbar, da einfach eine neue User Story angelegt wird | Time Stamp, Safety Story |
| **Software Verifikation** | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| 7.9.2.1 | Verifizierung über jede Phase planen und dokumentieren | NICHT OK | Sprint Planning und Review | Keine Dokumentation | Informelle Kommunikation, Protokolle, Snapshots |
| 7.9.2.2 | Software Verifizierungsplan | NICHT OK | Sprint Planning und Review | Keine Dokumentation | Protokolle, Snapshots |
| 7.9.2.4 | Nachweis dokumentieren, ob prüfende Phase abgeschlossen ist | NICHT OK | Sprint Planning und Review | Keine Dokumentation | Informelle Kommunikation, Protokolle, Snapshots |

**Tabelle 4.1:** IEC 61508-3 und Smart Home

# 5 Empfehlungen

In diesem Abschnitt werden alle Optimierungsmöglichkeiten aus Kapitel 4 vorgestellt. Das Ziel ist es, die Anforderungen der Norm zuerst durch informelle Kommunikation und formelle Kommunikation zu lösen. Erst wenn beide Methoden nicht ausreichen, soll auf die Methoden der Abschnitte 5.3 bis 5.9 zurück gegriffen werden. Hierbei ist es wichtig mit den Kunden abzusprechen, welche Dokumentation wirklich benötigt wird und wie weit die Lösungen durch Kommunikation ausreichen. Dies kann sich während dem Projekt ändern und wird individuell definiert.

## 5.1 Formelle Kommunikation

Die formelle Kommunikation ist in SCRUM eine wichtige Kommunikations- und Dokumentationsmethode. Zu formeller Kommunikation zählen vor alle Scrum Meetings. Die Ergebnisse der Meetings können, falls notwendig, durch Snapshots gesichert werden. Hier ist das Verhältnis und die Absprache zwischen Kunden und Entwickler entscheidend, inwiefern die formelle Kommunikation als Dokumentation ausreicht. Allerdings reicht oft die Diskussion aus und ergibt Lösungen und Ergebnisse, zum Beispiel in Form von neuen Safety User Stories. Durch regelmäßige Meetings wird ein Vertrauensverhältnis zwischen den Entwicklern sowie zu den Kunden aufgebaut und gestärkt. Im Folgenden werden die verschiedenen Meetings vorgestellt, die für das sicherheitskritische System wichtig sind.

### 5.1.1 Sprint Planning

Im Planning Meeting werden die Stories ausgewählt, die für die Erreichung des Sprintziels erforderlich sind. Alle ausgewählten Stories sind hinreichend spezifiziert, dass heißt die technische Herausforderung ist klar, die Akzeptanzkriterien wurden festgelegt, der Aufwand wurde geschätzt und die Priorität bestimmt. Am Ende des Meetings sollten die Anforderungen und Ziele klar definiert sein. [PHS+08]

Im Sprint Planning Meeting nimmt das gesamte Projektteam, der Scrum Master und der Safety Expert teil.

Die Sicherheitsanforderungen sollten in diesem Meeting allerdings schon vorher diskutiert worden sein (siehe 5.1.3).

### 5.1.2 Daily Scrum Meeting

Im Daily Scrum Meeting sollen Probleme diskutiert und gelöst werden. Die Berichterstattung sollte nur ein kleiner Teil des Meetings umfassen. Das Ziel ist es, die Probleme frühzeitig aufzudecken, um spätere Probleme zu vermeiden. [PHS+08] Das Daily Scrum Meeting in Sicherheitskritischen Softwareprojekten muss sich mit folgenden vier Fragen beschäftigen [MSH16]:

- Was habe ich gestern fertiggestellt?
- Was ist für heute geplant?
- Welche Probleme gibt es?
- Gibt es Sicherheitseinflüsse?

Das Daily Scrum Meeting muss nicht täglich stattfinden und ist von Projekt zu Projekt zu definieren. Die Ergebnisse können zusätzlich auf einem Whiteboard festgehalten werden und durch Snapshots gesichert werden. [Agi16]

Am Daily Scrum Meeting nimmt das Projekt Team, der Scrum Master und gegebenenfalls der Safety Expert teil.

### 5.1.3 Safety Scrum Meeting

Da die Sicherheitsaspekte nicht ausreichend in den Daily Scrum Meetings besprochen werden können, sollte einmal in der Woche ein zusätzliches Safety Scrum Meeting durchgeführt werden. Hier werden ausschließlich sicherheitskritische Funktionen und Anforderungen besprochen und diskutiert. Die Ergebnisse sind dann die Grundlage für alle weiteren Meetings. [MSH16]

Der Safety Expert und das Projekt Team sowie der Scrum Master müssen hier teilnehmen. Eine abgeschwächte Möglichkeit ist es, einen internen Safety Expert zu bestimmen, der das Safety Scrum Meeting mit dem externen Safety Expert führt.

### 5.1.4 Sprint Review

Im Sprint Review werden die letzten Arbeiten des vergangenen Sprints vorgestellt. Alle noch nicht fertiggestellten Stories werden diskutiert und für den nächsten Sprint angepasst. Die Herausforderung bei komplizierten und großen Projekten ist es, die vielen Anforderungen in das kurze Meeting unterzubringen. [Agi15] Deshalb ist es sehr wichtig Anforderungen zu priorisieren. [PHS+08] Nach dem Sprint Review entstehen meist Diskussionen und die direkte Kommunikation wird gefördert. [HRH08]

Im Sprint Planning Meeting nimmt das gesamte Projektteam, der Scrum Master, der Product Owner und der Safety Expert teil.

### 5.1.5 Sprint Retrospective

Die Sprint Retrospektive bietet die Möglichkeiten, Scrum Prozesse und Praktiken zu diskutieren. Hier ist es möglich die Entwicklung individuell anzupassen und auf Veränderungen zu reagieren. Auch Kommunikationsprobleme, die durch Scrum Praktiken entstehen werden hier diskutiert. [PHS+08]

Im Sprint Retrospective Meeting nimmt das gesamte Projektteam, der Scrum Master, der Product Owner und der Safety Expert teil.

## 5.2 Informelle Kommunikation

Die informelle Kommunikation ist in der agilen Entwicklung von großer Bedeutung. Es ist allerdings abhängig von den Kunden, in wie weit diese Methoden akzeptiert werden. Verlangen die Kunden zusätzliche Dokumentation und geben sich nicht mit der informellen Kommunikation zufrieden, bleibt den Entwicklern nichts anderes übrig als auf Abschnitte 5.3, 5.4, 5.6 und 5.8 zurück zu greifen. Daher ist es wichtig eine Vertrauensbasis zwischen den Entwicklern und den Kunden aufzubauen, um möglichst viel Dokumentation einzusparen.

Folgende agile Praktiken werden verwendet um die informelle Kommunikation zu verbessern:

- **Open Office Space**

  Alle Entwickler arbeiten in einem offenen Arbeitsraum, der gegebenenfalls durch einzelne Trennwände unterteilt wird. Dadurch ist es jederzeit möglich Diskussionen zu führen. Es können allerdings auch Probleme bei dieser Praktik auftretetn. Meetings innerhalb kleiner Gruppen können andere unbeteiligte Entwickler ablenken und die Konzentration mindern. Dennoch ist der Profit aus dieser Umsetzung höher zu bewerten. [PHS+08]

- **Pair Programming**

  Pair Programming wird vor allem verwendet um Code-Reviews durchzuführen. [PHS+08]

- **Kontinuierliche Integration**

  Durch kontinuierliche Integration wird der aktuelle Stand für Tester regelmäßig verfügbar gestellt. [PHS+08]

Die Praktiken bieten einen effizienteren Informationsaustausch und verbessern die Zusammenarbeit im Team. [PHS+08] Bei der informellen Kommunikation wird über synchrone und asynchrone Kanäle kommuniziert. Diese werden in diesem Abschnitt erklärt und differenziert.

### 5.2.1 Synchrone Kommunikation

Die schnellste und effizienteste Lösung ist es, durch synchrone Kommunikation zu einer Lösung zu gelangen. Dies ist vor allem durch Telefon, Videokonferenzen oder von Angesicht zu Angesicht umzusetzen. [HRH08]

Vor allem bei Entwurfsentscheidungen führt diese Art der Kommunikation schnell zum Ziel und verringert den Dokumentationsaufwand. [PHS+08] Falls keine direkte Kommunikation möglich ist, muss auf asynchrone Kommunikation zurück gegriffen werden. [KA07]

### 5.2.2 Asynchrone Kommunikation

Die asynchrone Kommunikation wird hauptsächlich verwendet, wenn es nicht möglich ist direkte Kommunikation durch zu führen. Bei der Verbreitung von Informationen ist es zudem oft sinnvoller asynchrone Kanäle zu verwenden. Dies muss im Projekt von Fall zu Fall unterschieden werden.

Zu den asynchronen Kanälen gehören E-Mail Listen, Ticketing Systeme oder Code Repositories.

Im Projekt Smart Home wird JIRA als Verwaltungstool verwendet. Über JIRA kann über User Stories kommuniziert werden, indem der Status verändert wird. Diese Kommunikation allein reicht aber nicht aus und es werden weitere Informationen benötigt, die am besten über synchrone Kommunikation ausgetauscht werden. [CMD16]

## 5.3 Safety Product Backlog

Bei sicherheitskritischer Software ist es wichtig die safety Anforderungen von den funktionalen Anforderungen zu trennen. Funktionale Anforderungen verändern sich regelmäßig, hingegen bleiben safety Anforderungen meist stabil und lassen sich sogar auf andere Projekte anwenden. Es wird daher empfohlen zwei Product Backlogs anzulegen [MSH16]. Einen funktionalen Product Backlog der alle funktionalen Anforderungen beinhaltet und einen Safety Product Backlog, der nur die safety Anforderungen enthält. Alternativ zu einem zweiten separaten Backlog lassen sich die safety Anforderungen auch durch Tags abgrenzen. Im Projekt SmartHome wird dies durch Safety Epics realisiert.

Es ist wichtig, dass nachvollziehbar ist welche Sicherheitsanforderung sich auf welche funktionale Anforderung auswirkt. Das heißt es müssen alle Beziehungen zwischen den beiden Backlogs herausgearbeitet werden. Die Umsetzung erfolgt über Referenzen von der funktionalen Anforderung auf die jeweilige Sicherheitsanforderung. Zusätzlich ist es möglich, falls sinnvoll, eine Erklärung hinzuzufügen, die kurz darstellt warum die Beziehung besteht. [MSH16]

Da nach jedem Sprint der Product Backlog aktualisiert werden kann, ist es wichtig, dass diese Referenzen existieren. Jede Änderung muss durch die Sicherheitsanforderungen geprüft werden. [MSH+14]

Der Safety Product Backlog muss von einem safety Experten erstellt werden. Der Product Owner besitzt meist nicht das nötige Wissen um diesen zu erstellen. Die Herausforderung besteht darin, die Abstimmung zwischen Sicherheitsexperten, Product Owner und dem Team optimal umzusetzen. Es empfiehlt sich, einen internen Sicherheitsexperten für jedes Team zu benennen, der regelmäßigen Kontakt zu dem externen Sicherheitsexperten aufnimmt. Diese zusätzliche Kommunikation wird im Kapitel 5.1 genauer beschrieben.

## 5.4 Safety Story

Die Safety User Story ist eine Spezialisierung der normalen User Story im Scrum Prozess. Die Safety User Story besitzt zwei Quellen, die im folgenden genauer erklärt werden:

- **Standard Anforderung:**

  Die Safety Story wird direkt oder indirekt mit der funktionalen User Story verknüpft. Die Form ergibt sich aus der Sicherheitsanforderung selbst und der Kriterien, die zum erfüllen oder vermeiden sind. “ Um <eine Sicherheitsanforderung> zu erfüllen, muss das System <etwas erfüllen oder vermeiden>“. Dieses Muster beinhaltet alle wichtigen Informationen in einem Satz und erfüllt die INVEST Kriterien.

- **System Gefahrenanalyse**

  Die Safety User Story muss sich auf die Analyse beziehen und diese referenzieren, die dafür gesorgt hat, dass sie besteht.

  Form: “Um <die Funktion> sicher zu halten, muss das System <etwas erreichen oder vermeiden>’".

Die Safety User Story wird durch den Safety Experten erstellt. Jede Safety User Story muss allerdings mit dem gesamten Team und dem Product Owner diskutiert werden, bevor sie abgeschlossen wird. Dadurch entsteht ein besseres Sicherheitsgefühl für alle Teilnehmer und die Kommunikation wird gefördert.

## 5.5 INVEST

Das INVEST Prinzip ist eine Empfehlung für die Umsetzung von User Stories. Scrum lässt die Art und Weise, User Stories zu schreiben größtenteils offen, dadurch gibt es keine einheitliche und strukturierte Umsetzung. Das INVEST Prinzip lässt sich problemlos auf Safety User Stories übertragen. Jede Safety User Story muss daher folgenden Kriterien erfüllen [wib14]:

- **Independence**

  Die Safety User Story sollte eigenständig sein und keine Abhängigkeiten zu anderen Safety User Stories beinhalten. Abhängigkeiten zu Funktionalen User Stories sind allerdings von Nöten.

- **Negotiable**

  Safety User Stories können jederzeit verändert werden bis sie Teil eines Sprints werden. Alle Änderungen und Folgen für funktionale User Stories müssen berücksichtigt werden.

- **Valuable**

  Eine Safety User Story muss für den Kunden und externen Sicherheitsexperten wertvoll sein.

- **Estimable**

  Der Umfang einer Safety User Story muss abschätzbar sein.

- **Small**

  Die Safety User Story muss so klein wie möglich gehalten werden.

- **Testable**

  Die Safety User Story muss alle nötigen Informationen liefern um sie später testen zu können.

Die Umsetzung des INVEST-Prinzips muss vom Scrum Master kontrolliert werden.

## 5.6 Time Stamp und Wiki

Im Scrum Prozess werden regelmäßig Anforderungen verändert und angepasst. Sicherheitsanforderungen ändern sich nicht so häufig wie funktionale Anforderungen, dennoch müssen sie immer wieder angepasst werden. Durch Änderungen an safety Anforderungen werden alte User Stories verworfen und neue erstellt. Dies lässt sich später im Projekt nicht mehr konstruieren und ist daher nicht nachvollziehbar. Da Änderungen in Safety User Stories weitreichende Folgen haben können müssen diese nachvollziehbar sein. Deshalb sollte jede Änderung einer Sicherheitsanforderung mit einem Time Stamp versehen und in einem Wiki abgelegt werden.

Der Time Stamp sollte das Änderungsdatum, den Bearbeiter und einen Verweis auf die alte User Story haben.

Alle Änderungen sind vom Safety Experten in Absprache mit dem Product Owner durchzuführen. [HNM10]

## 5.7 Safety Test Driven Development

Das Safety Test Driven Development (STDD) ist eine modifizierte Version des Test Driven Development. In dieser Spezialisierung ist es wichtig sobald es möglich ist einen Test zu erstellen. Danach lässt man die Tests laufen, die nach ausgewählter Kriterien relevant sind, und stellt fest dass die neuen Tests fehlschlagen. Nun setzt man die nötigen Veränderungen um und testet erneut, aber nun erfolgreich. Die Tests sollten bestenfalls automatisiert durchgeführt werden. Hierfür gibt es eine Reihe von Frameworks , auf die hier nicht genauer eingegangen wird. [MSH16]

Gut geschriebene Unit Tests können als Testdokumentation verwendet werden und sind meist ausreichend. Hierbei reicht es aus die Ergebnisse zu exportieren, das durch verschiedene Automatisierungstools möglich ist.

Verantwortlich für die Tests ist das Entwicklungsteam, das auf Grundlage der User Stories und deren Akzeptanzkriterien die Tests entwickelt.

## 5.8 Snapshots

Zusätzlich zur Kommunikation in den Meetings können die Ergebnisse über Snapshots festgehalten und dokumentiert werden. Es empfiehlt sich, die Ergebnisse der Planung und Diskussion eines jeden Meetings auf einem Whiteboard festzuhalten. Am Ende einer jeden Sitzung können diese Ergebnisse mit einem Snapshot gesichert werden. Zusätzlich ist es wichtig, die Snapshots mit Datum und Teilnehmerliste zu archivieren. Es liegt allerdings an den festgelegten Projektanforderungen, in wie weit Snapshots als Dokumentation anerkannt werden. [MSH+14] Snapshots bieten sich zu dem bei Entwurfsmodellen und Diskussionen an und werden dort in nahezu jedem Projekt anerkannt.

Snapshots sollte der Scrum Master anfertigen und dem Team zur Archivierung aushändigen.

## 5.9 Akzeptanzkriterien

Akzeptanzkriterien sind ein Teil der Safety User Stories. Akzeptanzkriterien für Sicherheitsfunktionen wird vom externen Safety Expert festgelegt und in den User Stories festgehalten.

Akzeptanzkriterien sind die Grundlage für spätere Tests, denn sie bilden die Vorlage. Sie werden in folgende drei Bereiche unterteilt. Als erstes wird eine fixe Vorbedingung definiert. Danach folgt die Beschreibung der durchzuführenden Aktionen. Der dritte Bereich ist das erwartete Ergebnis des Tests.

Die Akzeptanzkriterien werden bei funktionalen Anforderungen vom Product Owner definiert. Bei sicherheitskritischen Anforderungen ist der Safety Expert verantwortlich. [adm16]

# 6 Umsetzung in Smart Home

Im folgenden Kapitel werden die  Empfehlungen auf das Projekt Smart Home angewandt.

## 6.1 Formelle Kommunikation

Zusätzlich zu den Standard Meetings wird das Projekt Smart Home um die folgenden Meetings ergänzt:

- **Weekly Safety Meeting**

  Das Weekly Safety Meeting ist abgeleitet von dem Daily Scrum Meeting. Allerdings werden in diesem Meeting ausschließlich Sicherheitsaspekte besprochen und diskutiert. Dadurch entsteht eine Sicherheitskultur im Rahmen des gesamten Projektteams. [WRW17]

- **Pre Planning Meeting**

  Das Pre Planning Meeting wird in Kapitel 3.2.3 beschrieben.

## 6.2 Informelle Kommunikation

Die informelle Kommunikation wird bereits durch mehrere Methoden umgesetzt und muss nur noch ergänzt werden. Es existiert ein gemeinsamer Arbeitsraum, in dem sich die Projektteilnehmer zweimal wöchentlich 8 Stunden treffen. Somit ist der ständige und direkte Austausch gewährleistet und einfache Probleme werden schnell gelöst. Zusätzlich wird mit Pair Programming gearbeitet, so dass immer einer anwesend ist der Bescheid weiß. Das Pair Programming könnte noch für Code Reviews sinnvoll eingesetzt werden.

In Smart Home könnte man durch kontinuierliche Integration die Arbteit des Testers erleichtern. Der aktuelle Stand wird regelmäßig verfügbar gestellt, so dass die Tester während der Entwicklung Fehler entdecken und Probleme frühzeitig erkennen können.

## 6.3 Safety Product Backlog

Der Safety Product Backlog kann im Projekt Smart Home nicht komplett vom Product Backlog getrennt werden, da JIRA dies nicht unterstützt. Deshalb werden Sicherheitsanforderungen mit Safety Epics gekennzeichnet und so eindeutig abgegrenzt.

## 6.4 Safety Story

Den bisherigen Safety Stories wird die Gefahrenanalyse, die dafür gesorgt hat, dass die Anforderung entsteht, hinzugefügt. Jede Safety User Story muss nach dem INVEST Prinzip erstellt werden. Diese konsistente Umsetzung ist bisher nicht vorhanden.

User Stories müssen zudem priorisiert und aktuell gehalten werden.

Im Folgenden werden zwei Beschreibungen der neuen Safety Story vorgestellt:

- **Bisherige Safety Story:**

  1. „Um den Roboter zurück auf den schwarzen Linienpfad zu bringen, muss der Roboter den Weg suchen können."

  2. „Um den Roboter auf der schwarzen Linie zu halten, muss der Roboter diese erkennen."

- **Neue Safety Story:**

  1. „ Um den Roboter zurück auf den schwarzen Linienpfad zu bringen, muss der Roboter den Weg suchen können. Um die Sicherheitsziele 1 und 2 zu erfüllen, muss der Roboter die Fahrt stoppen, sobald er die Linie verlässt."

  2. „Um den Roboter auf der schwarzen Linie zu halten, muss der Roboter diese erkennen. Um die Funktionen H 2.4 und H 2.5 sicherzustellen, muss der Roboter der schwarzen Linie folgen."

## 6.5 INVEST

Das INVEST Prinzip wird durch die Safety Story erfüllt und muss durchgängig umgesetzt werden.

## 6.6 Time Stamp und Wiki

Sobald Safety Stories geändert oder verworfen werden, soll ein Verweis auf die neu entstandene Safety Story gesetzt und mit einem Time Stamp versehen werden. Dies ermöglicht es später genau nachzuvollziehen warum eine Änderung vollzogen wurde. Alte User Stories sollen dann in einem Wiki abgelegt werden.

Eine weitere Möglichkeit diese beiden Methoden einzusetzen ist es, Meetings oder Gespräche zu dokumentieren und mit einem Time Stamp versehen abzulegen. Diese Methode wird nur dann eingesetzt, wenn alle obigen Ansätze für den Kunden nicht ausreichen. Denn sie erzeugen für den agilen Ansatz unnötig viel Dokumentation. Dieser Teil ist für das Projekt Smart Home nicht relevant, da der Kunde keine Dokumentation der Gespräche benötigt.

## 6.7 Safety Test Driven Development

Im Projekt Smart Home werden Unit Tests geschrieben, allerdings nicht nach dieser Methode. Durch das Safety Test Driven Development könnten Tests früher ausgeführt und Fehler schneller erkannt werden. Voraussetzung hierfür ist die kontinuierliche Integration.

## 6.8 Snapshots

Snapshots werden vor allem für Entwurfszwecke verwendet. Auch hier ist es wichtig, nur Snapshots anzufertigen wenn sie wirklich nötig sind oder der Kunde es fordert. Es können einzelne Entwurfsmuster, Skizzen von Meetings oder Abläufe gesichert werden. Im Projekt Smart Home wurden ausschließlich Entwurfsentscheidungen festgehalten.

## 6.9 Akzeptanzkriterien

Sie sind Teil der Safety Story und werden vom externen Safety Expert festgelegt.

# 7 Validierung (Interview)

In diesem Kapitel werden die Ergebnisse aus Kapitel 5 genutzt, um ein Interview für zwei Industriepartner zu erstellen. Dadurch werden die Empfehlungen überprüft und eventuelle Mängel diskutiert.

## 7.1 Vorgehensweise

Die Fallstudie basiert auf den Ergebnissen dieser Arbeit. Im Laufe des Studienprojekts Smart Home werden die Sicherheitsanforderungen analysiert und das Erfüllen der Norm IEC 61508 überprüft. Die Empfehlungen für die Umsetzung der Anforderungen aus der Norm werden als Grundlage für diese Fallstudie verwendet. Die Empfehlungen sind während der Bachelorthesis vom 1.12.2016 bis zum 31.03.2017 entstanden.

In einem Interview mit zwei Personen aus der Industrie werden die Empfehlungen vorgestellt und die verschiedenen Meinungen eingeholt. Die eine Person arbeitet als Softwareentwickler bei einem mittelständischen Automobilzulieferer und befindet sich aktuell in einem agilen Projekt. Die andere Person arbeitet als IT-Consultant bei einem IT-Dienstleister und beschäftigt sich mit Car to Car Kommunikation. In diesem Bereich wird sicherheitskritische Software agil entwickelt.

Das Interview gliedert sich in fünf Abschnitte, die wie folgt erläutert werden:

- **Formelle Kommunikation**

  Zusätzlich zu den üblichen Scrum Meetings wird ein Safety Scrum Meeting eingeführt, da die Sicherheitsaspekte in den Daily Meetings nicht ausreichend besprochen werden können. Hier werden ausschließlich sicherheitskritische Funktionen und Anforderungen besprochen und diskutiert. Anhand dieser Erklärung wird der Interviewpartner auf Nutzen, Nachteile und Kundensicht befragt.

- **Informelle Kommunikation**

  Um die informelle Kommunikation zu verbessern kann ein offener Arbeitsraum angelegt werden. Der Open Office Space wird eingerichtet um jederzeit Diskussionen zu ermöglichen, die offene Probleme und Fragen direkt lösen. Eine weitere Praktik ist es, Pair Programming bei Code - Reviews einzusetzen. Anhand diesen Empfehlungen wird nach Störfaktoren, Teamarbeit und Einsparung von Dokumentation befragt.

- **Safety Backlog**

  Es werden zwei Product Backlogs erstellt. Einen funktionalen und einen Safety Backlog. Die Trennung kann auch durch Tags im Product Backlog geschehen. Die Safety User Stories bestehen aus zwei Quellen. Zum einen aus der Standard Anforderung und zum anderen aus der Gefahrenanalyse, die dafür gesorgt hat, dass sie besteht. Der Interviewpartner wird nach einer Einschätzung dieser Praktik befragt.

- **Sicherheitsexperten**

  Es wird ein interner und ein externer Sicherheitsexperte eingesetzt. Der externe Sicherheitsexperte hat direkten Kontakt zum Kunden und erstellt den Safety Backlog. Um die Kommunikation mit dem Team zu verbessern wird ein interner Sicherheitsexperte eingesetzt, der ständigen Kontakt mit dem externen Sicherheitsexperten hat. Ersparen Experten zusätzliche Dokumentation und wird somit die Qualität gesteigert?

- **Allgemein**

  Abschließend wird der Interviewpartner nach seiner abschließenden Meinung befragt. Er soll eine Einschätzung darüber geben, ob es durch diese Methoden möglich ist, sicherheitskritische Systeme in agilen Projekten umzusetzen. Des Weiteren wird die Bereitschaft des Kunden, agile Methoden einzusetzen, durch Erfahrungen aus früheren Projekten hinterfragt.

Die Antworten aus beiden Interviews werden zusammengetragen und im folgenden Abschnitt analysiert und diskutiert.

## 7.2 Ergebnisse

Die Ergebnisse beziehen sich auf die zwei Interviews mit den Industriepartnern. Sie werden anhand der fünf Abschnitte analysiert und diskutiert. In den folgenden Abschnitten wird der Softwareentwickler als Partner A und der IT-Consultant als Partner B bezeichnet.

- **Formelle Kommunikation**

  Partner B hat bereits Erfahrungen mit diesem zusätzlichen Meeting und sieht nur Vorteile darin. „Es reicht nicht aus Sicherheitsanforderungen im Daily Scrum Meeting zu besprechen. Es ist dringend notwendig diese vorher im Detail zu klären, da die Zeit im Daily Scrum Meeting begrenzt ist.“ Laut Partner A reichen die Ergebnisse des Meetings aber nicht immer aus um den Kunden zufrieden zu stellen. „Der Kunde erwartet manchmal zusätzliche Protokolle oder Dokumentation.“ Partner B hingegen sieht die formelle Kommunikation als Ersatz für die Dokumentation an. „ Die Ergebnisse formeller Kommunikation machen zusätzliche Dokumentation überflüssig. Jeder Kunde, der den agilen Prozess annimmt und das werden immer mehr, sieht früher oder später die Vorteile dieser Methode. Die Qualität der Software und des Projekts wird so deutlich verbessert.“

Die Vorteile, die durch die formelle Kommunikation entstehen spiegeln sich auch in der Industrie wider. Dennoch ist es immer noch von den Kunden abhängig, in wie weit diese Methode umgesetzt werden kann. Moderne Unternehmen gehen aber mit der Zeit und sind immer mehr bereit, diesen Weg zu gehen.

- **Informelle Kommunikation**

  Mit einem offenen Arbeitsraum haben bisher beide Interviewpartner ihre Erfahrungen gemacht. Partner A sieht große Chancen in dieser Methode, allerdings auch viele Gefahren. „ Durch einen offenen Arbeitsraum wird die Bindung im Team deutlich verbessert. Auch kleinere Probleme können so schnell und effektiv gelöst werden. Allerdings ist das Arbeiten, während andere Projektteilnehmer Meetings abhalten, oft nicht möglich. Der daraus entstehende Nachteil ist leider zu groß. Da der Ansatz aber gut ist, muss man versuchen einen Mittelweg dieser Methode zu finden.“ Partner B hat bisher nur schlechte Erfahrungen gemacht und die Methode wurde in den jeweiligen Projekten wieder abgebrochen. „Die Störfaktoren durch andere Mitarbeiter hat das Arbeiten in kleinen Teams oder alleine nicht effizient umsetzen lassen.“

  Diese Methode muss für zukünftige Projekte angepasst und neu analysiert werden. Die Störfaktoren eines offenen Arbeitsraumes müssen beseitigt werden, so dass die Vorteile erhalten bleiben. Es ist vielleicht sinnvoll größere Meetings in einen anderen Raum zu verlegen, um andere Projektmitarbeiter nicht zu stören.

- **Safety Backlog**

  Die Trennung des Backlogs ist laut Partner B eine Grundvoraussetzung. „Ohne eine Unterteilung des Backlogs ist es nicht möglich ein sicherheitskritisches System zu entwickeln. Die Gefahrenanalyse in die User Story mit einzubinden sehe ich als sinnvoll, obwohl ich aus Erfahrung diese Information nie mehr benötigt habe.“ Partner A sieht keinen entscheidenden Vorteil darin, den Backlog aufzusplitten. „Eine Trennung in zwei Backlogs ist meiner Meinung nach überflüssig, denn die Trennung kann man auch so sichtbar machen.“

  Bei diesem Thema gehen die Meinungen sehr auseinander. Da Partner A bisher keine sicherheitskritischen Systeme entwickelt hat, gewichte ich die Aussage von Partner B stärker. Die Aussage von Partner B lässt sich aber auf die Tatsache, dass eine Trennung auch durch Epics erfolgen kann, relativieren.

- **Sicherheitsexperten**

  Der interne Sicherheitsexperte ist für beide Interviewpartner noch neu. Aus Erfahrungen früherer Projekte sieht Partner B entscheidende Vorteile, die damals die Umsetzung behindert haben. „In einem früheren Projekt war der externe Sicherheitsexperte nur schwer zu greifen und selten vor Ort. Durch einen zusätzlichen internen Sicherheitsexperten hätte man Unklarheiten schneller und effizienter lösen können. Es war üblich, dass der Sicherheitsexperte gar nicht alle Anforderungen im Detail mit den Entwicklern

absprechen konnte, da die Zeit begrenzt war. Die Qualität des Produkts hätte meiner Meinung nach mit einem internen Sicherheitsexperten deutlich gesteigert werden können." Partner A möchte sich zu dieser Frage nicht äußern.

Auch in der Industrie ist es sinnvoll einen internen Safety Expert einzusetzen, auch wenn es noch keine praktischen Erfahrungen der beiden Interviewpartner gibt.

- **Allgemein**

  Die abschließenden Meinungen sind sehr ähnlich. Partner A sieht einzig und allein den Kunden als Bremse, sicherheitskritische Systeme agil zu entwickeln. „Wenn es darum geht Systeme agil zu entwickeln spielt der Kunde die entscheidende Rolle. Da ist es nicht relevant, ob es sich zusätzlich um sicherheitskritische Systeme handelt. Wenn der Kunde die agilen Methoden akzeptiert und auf gewisse Dokumente verzichtet, dann lässt sich jedes System agil umsetzen. Allerdings habe ich in der Vergangenheit oft die Erfahrung gemacht, dass der Kunde dazu nicht bereit ist." Partner B hat bisher bessere Erfahrungen gemacht und sieht dies anders. „Sicherheitskritische Systeme lassen sich perfekt mit agilen Methoden umsetzen. Gerade bei solchen Systemen ist es wichtig den schnellen Wandel zu berücksichtigen und eine rege Kommunikation zu führen. Das Denken in Monaten und Jahren führt in der heutigen Zeit nicht mehr zum gewünschten Ziel. Das Umdenken findet aktuell zum Glück auch bei den Kunden statt. So ist es möglich die agilen Methoden gewinnbringend einzusetzen. In Zukunft führt kein Weg an der agilen Entwicklung vorbei!"

  Es wird deutlich, dass auch die Industriepartner den agilen Weg als sinnvoll einschätzen. Sicherheitskritische Systeme können so ohne Nachteile entwickelt werden.

Abschließend lässt sich sagen, dass die gefundenen Methoden die agile Entwicklung unterstützen und bis auf kleinere Anpassungen effizient umgesetzt werden können.

# 8 Diskussion

Nahezu kein sicherheitskritisches Systeme wird bisher agil entwickelt (Stand Dezember 2016). Innerhalb eines halben Jahres hat sich diese Situation schon verändert und es trauen sich mehr Firmen sicherheitskritische Systeme agil zu entwickeln. Da die Zukunft in der agilen Entwicklung liegt hat sich diese Bachelorthesis mit dieser Umsetzung auseinander gesetzt.

Die Grundlage für die Erfüllung sicherheitskritischer Anforderungen stellt die Norm IEC 61508. Da die Norm bislang nur für das V-Modell existiert, stellt die Umsetzung im agilen Projekt eine Herausforderung dar. Doch ist diese Herausforderung kein Grund, die Umsetzung zu verweigern.

Das Studienprojekt Smart Home und die Empfehlungen aus Kapitel 5 zeigen nämlich sehr gut, dass sich sicherheitskritische Systeme auch agil umsetzen lassen. Die Herausforderung liegt viel mehr in den Menschen, die dieses Projekt umsetzen. Jeder Stakeholder muss sich darauf einlassen, dass agil entwickelt wird. Es wird viel Dokumentation gespart und durch häufige Kommunikation ersetzt. Da dies für viele immer noch neu ist, bewähren sich Kunden oft auf altbekanntes. Vor allem bei einem so heiklen Gebiet wie der funktionalen Sicherheit, sind wenige bereit, dieses Risiko einzugehen. Durch die vorgestellten Methoden ist diese Angst allerdings unbegründet. Zu Beginn könnte man sich immer noch mit Methoden, wie zum Beispiel Time Stamp und Wiki oder Snapshots absichern. Dies wäre der erste Schritt in Richtung agiler Entwicklung von sicherheitskritischer Software. Durch anwenden der agilen Methoden wird man zunehmend sicherer und verzichtet in naher Zukunft auf diese Absicherung.

Zusammenfassend lässt sich sagen, dass sicherheitskritische Software und die Norm IEC 61508 mit der agilen Entwicklung vereinbar sind.

# 9 Ausblick

Die Anforderungen an Systeme mit funktionaler Sicherheit werden in Zukunft noch stärker ansteigen, so dass herkömmliche Herangehensweisen nicht mehr den Maßstäben von morgen standhalten werden. Die Komplexität der Systeme wird weiter steigen und so ist es nötig innovative Wege in der Software Entwicklung zu verwenden. Die Umsetzung durch agile Entwicklungsmethoden wird sich in Zukunft durchsetzen, um den Anforderungen der Zukunft gerecht zu werden.

Die Welt ist momentan im digitalen Wandel. Die Prozesse werden kurzweiliger, individueller und anpassungsfähiger. Ein sicherheitskritisches Projekt das in naher Zukunft über eine herkömmliche Methode entwickelt wird, kann bei seiner Fertigstellung schon wieder veraltet sein. Deshalb ist es wichtig sich auf den agilen Prozess einzulassen und mit der Zeit zu gehen. Das es möglich ist Normen mit dieser Entwicklungsmethode zu erfüllen hat das Projekt Smart Home gezeigt. Die Unternehmen müssen die Zeit investieren und ein Umdenken, meist auch ein Kulturwandel in der Organisation durchführen.

# Literaturverzeichnis


[adm16] admin. *Akzeptanztests*. 2016. URL: https://www.scrum.de/einfach-mit-akzeptanztests-loslegen/ (zitiert auf S. 35).

[Agi15] S. Agile. *Team Demo Abstract*. 2015. URL: http://www.scaledagileframework.com/team-demo/ (zitiert auf S. 30).

[Agi16] S. Agile. *Iteration Planning Abstract*. 2016. URL: http://www.scaledagileframework.com/iteration-planning/ (zitiert auf S. 30).

[atl] atlassian. *Smart Home*. JIRA. URL: http://brest.informatik.uni-stuttgart.de:8080/secure/RapidBoard.jspa?rapidView=8&projectKey=ST (zitiert auf S. 14).

[ATO15] A. A. Abdelaziz, Y. El-Tahir, R. Osman. *Adaptive Software Development for Developing Safety Critical Software*. 2015. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7381425 (zitiert auf S. 11).

[CMD16] D. S. Cruzes, N. B. Moe, T. Dybå. *Communication between Developers and Testers in Distributed Continuous Agile Testing*. 2016. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7577420 (zitiert auf S. 32).

[HNM10] R. HODA, J. NOBLE, S. MARSHALL, Hrsg. *How Much is Just Enough? Some Documentation Patterns on Agile Projects*. 2010 (zitiert auf S. 11, 34).

[HRH08] M. Hummel, D. C. Rosenkranz, P. D. R. Holten. *The Role of Communication in Agile Systems Development*. 2008. URL: http://download.springer.com/static/pdf/887/art%3A10.1007%2Fs12599-013-0282-4.pdf?originUrl=http://link.springer.com/article/10.1007/s12599-013-0282-4&token2=exp=1491211527~acl=/static/pdf/887/art%253A10.1007%252Fs12599-013-0282-4.pdf?originUrl=http%3A%2F%2Flink.springer.com%2Farticle%2F10.1007%2Fs12599-013-0282-4*~hmac=ae371a191b14d390d00362001abc1bbe10e1c99283e24292f5061b56efdba3a5 (zitiert auf S. 11, 30, 32).

[IEC10] IEC. *Functional safety of electrical/electronic/programmable electronic safety-related systems Part 3: Software requirements*. 2010 (zitiert auf S. 13).

[KA07] M. Korkala, P. Abrahamsson. *Communication in Distributed Agile Development: A Case Study*. 2007. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4301081 (zitiert auf S. 32).

[MSH+14] T. Myklebust, T. Stålhane, G. K. Hanssen, T. Wien, B. Haugset. *Scrum, documentation and the IEC 61508-3:2010 software standard*. 2014. URL: http://meetingsandconferences.com/psam12/proceedings/paper/paper_31_1.pdf (zitiert auf S. 11, 33, 35).

[MSH16] T. Myklebust, T. Stålhane, G. K. Hanssen. *Use of Agile Practices when developing Safety-Critical Software*. 2016. URL: http://safescrum.no/wp-content/uploads/2016/08/Use-of-Agile-Practices-when-developing-SCSW-V_.pdf (zitiert auf S. 30, 32, 35).

[PHS+08] M. Pikkarainen, J. Haikara, O. Salo, P. Abrahamsson, J. Still. *The impact of agile practices on communication in software development*. 2008. URL: http://download.springer.com/static/pdf/781/art%3A10.1007%2Fs10664-008-9065-9.pdf?originUrl=http://link.springer.com/article/10.1007/s10664-008-9065-9&token2=exp=1491211426~acl=/static/pdf/781/art%253A10.1007%252Fs10664-008-9065-9.pdf?originUrl=http%3A%2F%2Flink.springer.com%2Farticle%2F10.1007%2Fs10664-008-9065-9*~hmac=26b4018ba8dbfa9bfe795626ff1468b7a695760076ef0e53ad8673a5d12b1874 (zitiert auf S. 12, 29–32).

[Stu16] StuPro. *Smart Home Main*. 2016. URL: https://gilbert.informatik.uni-stuttgart.de/dashboard/projects (zitiert auf S. 14).

[wib14] wibas. *User Story*. 2014. URL: https://www.wibas.com/scrum/user-story/de (zitiert auf S. 33).

[Wik17] Wikipedia. *IEC 61508*. 2017. URL: https://de.wikipedia.org/wiki/IEC_61508 (zitiert auf S. 13).

[WRW17] Y. Wang, J. Ramadani, S. Wagner. *An Exploratory Study of Applying a Scrum Development Process for Safety-Critical Systems*. 2017. URL: https://arxiv.org/pdf/1703.05375.pdf (zitiert auf S. 14, 37).

Alle URLs wurden zuletzt am 30. 05. 2017 geprüft.

**Erklärung**

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

---

Ort, Datum, Unterschrift