



### Bruno Albrecht

studierte bis 2022 Wirtschaftsinformatik am Standort Dresden der Berufsakademie Sachsen. Die vorliegende Zusammenfassung der Bachelorthesis: „Fallstudie zum Entwurf und zur Implementierung einer mobilen Anwendung zur Unterstützung der Scrum-Methode“ wurde im Sommer 2022 eingereicht und erfolgreich verteidigt.

**Kontakt:** bruno.albrecht@gmx.de

# Fallstudie zum Entwurf und zur Implementierung einer mobilen Anwendung zur Unterstützung der Scrum-Methode

*Bruno Albrecht*

*Scrum ist eine agile Projektmanagementmethode, die sich als Standard der agilen Softwareentwicklung durchgesetzt hat.<sup>1</sup> Der bloße Einsatz von Scrum garantiert jedoch noch keine Erfolge. Vielmehr gibt es auch beim Einsatz von Scrum einige Herausforderungen, die bewältigt werden müssen.<sup>2</sup>*

*Eine dieser Herausforderungen entsteht bei der Zusammenarbeit der Teammitglieder. Um diese Zusammenarbeit zu verbessern, gibt es kaum technische Hilfsmittel, weshalb die Entwicklung eines Tools zur Verbesserung dieser sinnvoll ist.*

*Die vorliegende Arbeit untersucht eine mögliche Architektur und Implementierung eines solchen Tools. Zu diesem Zwecke wurde die Forschungsmethode des Design Science gewählt, wobei als Artefakte zuerst eine Architekturbeschreibung und anschließend der Prototyp einer Anwendung erstellt wurden.*

*Dafür wurden, nachdem die Qualitätskriterien definiert waren, das jeweilige Artefakt erstellt und dargestellt und dieses zum Schluss auf Basis der Qualitätskriterien evaluiert. Die Arbeit schließt mit einem Fazit, in dem die Ergebnisse der Arbeit zusammengefasst werden.*

*Scrum is an agile project management method which has been established as the standard in agile software development.<sup>3</sup> The pure usage of scrum, however, does not guarantee success. A number of challenges arise while using scrum that must be overcome.<sup>4</sup>*

*One of these challenges arises during the collaboration of the team members. There are few technical tools that can improve this collaboration, which is why the development of such a tool to improve this collaboration is meaningful.*

*In this thesis a possible architecture and implementation of such a tool were examined. For this purpose, the research method of Design Science was chosen, which involved first an architecture description and subsequently the prototype of an application as artifacts.*

*First the quality requirements were defined so that the artefact, that was developed and later presented, could be evaluated afterwards. The thesis closes with a conclusion that summarizes the results.*

<sup>1</sup> Gloger (2014), S. 11.

<sup>2</sup> vgl. Pichler (2008), S. 2.

<sup>3</sup> Gloger (2014), S. 11.

<sup>4</sup> vgl. Pichler (2008), S. 2.

## Einleitung

Laut Schwaber und Sutherland, den Herausgebern des Scrum-Guides, ist Scrum ein Rahmenwerk, das von Teams und Organisationen genutzt werden kann, um komplexe Probleme zu lösen und so Wert zu generieren.<sup>5</sup> Aus diesem Grund wird Scrum in vielen Softwareentwicklungsprojekten eingesetzt, besonders, wenn diese noch nicht im Detail definierte Anforderungen oder ein dynamisches Umfeld haben.<sup>6</sup>

Die Verwendung von Scrum sollte jedoch kein Selbstzweck sein, denn auch bei der Verwendung von Scrum ergeben sich Herausforderungen, die bewältigt werden müssen, um Scrum effektiv einzusetzen.<sup>7</sup> Es gibt bereits verschiedene Tools, die den Einsatz von Scrum unterstützen sollen, wie z. B. Jira Agile<sup>8</sup> und Rally<sup>9</sup>. Diese sind jedoch eher auf die prozessuale Seite fokussiert, wohingegen es kaum technische Hilfsmittel gibt, welche die Interaktion der Teammitglieder fördern. Die Entwicklung eines solchen Tools ist der Fokus dieser Arbeit.

Im Rahmen dieser Arbeit sollen hierzu vier Forschungsfragen beantwortet werden. Diese sind die Folgenden:

Wie könnte die Architektur für eine mobile Anwendung zur Unterstützung von Scrum aussehen?

Welche Vorteile kann der Einsatz der mobilen Anwendung in Scrum-Projekten bringen?

Wurden die im Rahmen der vorhergehenden Arbeit beschriebenen Qualitätsanforderungen an die mobile Anwendung erfüllt?

Welcher Verbesserungen bedarf die evaluierte Version der mobilen Anwendung?

Um diese Fragen zu beantworten, wird zuerst eine Architekturbeschreibung erstellt und anschließend ein Prototyp einer Anwendung entwickelt, anhand dessen das User Interface (UI) und die Funktionalitäten getestet werden können. Zur Erstellung dieser Artefakte wird die Forschungsmethode des Design Science verwendet, die im Folgenden genauer erläutert wird.

## Forschungsdesign

Im Rahmen der Bachelorarbeit wurde die Forschungsmethode des Design Science gewählt. Entgegen dem Titel dieser Arbeit wurde keine Fallstudie durchgeführt, da es keinen zu untersuchenden Fall gab und sich diese Arbeit auf die Erstellung eines Artefaktes fokussieren sollte und nicht wie bei der Fallstudie üblich auf den theoretischen Erkenntnisgewinn.<sup>10</sup>

Bei der Methode des Design Science werden fünf Phasen durchlaufen:

- die Problembeschreibung, in der das Problem, welches mithilfe des Artefaktes gelöst werden soll, erläutert wird,
- die Definition der Anforderungen, in der Qualitätskriterien für das Artefakt festgelegt werden, anhand derer das Artefakt später evaluiert werden kann,
- die Erstellung des Artefaktes, in der das Artefakt erstellt wird,
- die Darstellung des Artefaktes, in der die Ergebnisse der vorherigen Phase und damit das Artefakt dargestellt werden und
- die Evaluierung des Artefaktes, in der das Artefakt anhand der zuvor festgelegten Qualitätskriterien evaluiert wird.<sup>11</sup>

Da in dieser Arbeit zwei Artefakte erstellt werden sollten, werden die soeben beschriebenen Phasen zweimal durchlaufen. Grund hierfür ist, dass die Architekturbeschreibung als Basis für die Erstellung der Anwendung dienen soll. Der Aufbau dieser Arbeit anhand des Design Science ist in Abbildung 1 dargestellt.

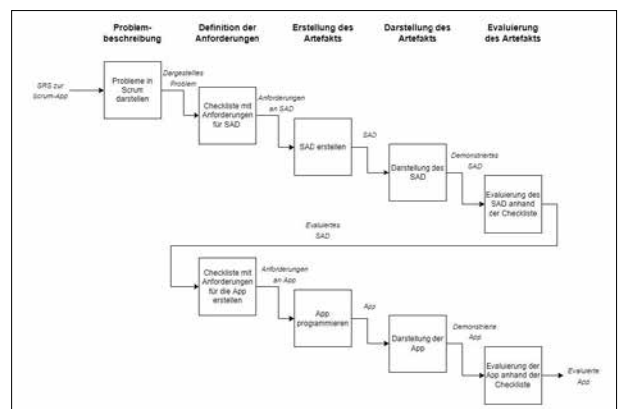


Abbildung 1: Forschungsdesign in Anlehnung an Johannesson et al.<sup>12</sup>

## Erstellung der Artefakte

Die Architekturbeschreibung wurde anhand der von Starke beschriebenen Struktur erstellt und auch der Großteil der Qualitätskriterien für die Architekturbeschreibung leitet sich aus diesem Grund aus den Ausführungen von Starke ab.<sup>13</sup>

In der Architekturbeschreibung wurden hauptsächlich Diagramme der Unified Modeling Language (UML) verwendet. Die verwendeten UML-Diagramme sind:

- ein Komponentendiagramm, das die Teile des Systems und deren Beziehungen untereinander darstellt,
- ein Klassendiagramm, das die Klassen der Objekte darstellt, die zwischen der Anwendung und dem Server ausgetauscht werden und
- Aktivitätsdiagramme, die das Verhalten des Systems zur Laufzeit darstellen.<sup>14</sup>

<sup>5</sup> vgl. Schwaber/Sutherland (2020), S. 3.

<sup>6</sup> vgl. Olfert (2019), S. 253 f.

<sup>7</sup> vgl. Pichler (2008), S. 2.

<sup>8</sup> Jira Agile ist eine von Atlassian entwickelte Software, die Tasks für Scrum darstellen kann. Mehr Informationen sind zu finden unter <https://www.atlassian.com/software/jira/agile> (letzter Zugriff: 05.07.2022).

<sup>9</sup> Rally ist eine von Broadcom entwickelte Software, die Abstimmung bezüglich der Aufgaben zwischen mehreren Teams ermöglicht. Mehr Informationen sind zu finden unter <https://www.broadcom.com/products/software/value-stream-management/rally> (letzter Zugriff 05.07.2022).

<sup>10</sup> vgl. Yin (2014), S. 16 f.

<sup>11</sup> vgl. Johannesson/Perjons (2014), S. 77.

<sup>12</sup> vgl. Johannesson/Perjons (2014), S. 77.

Außerdem wurden mehrere Architecture Decision Records (ADR) erstellt, in denen die wichtigsten Architekturentscheidungen festgehalten wurden. Hierfür wurde das Framework nach Fischer verwendet.<sup>15</sup> Für die mobile Anwendung wurden besonders die Qualitätskriterien verwendet, die bereits im Rahmen der Anforderungsspezifikation festgelegt wurden. Hierzu gehören die Kriterien der Norm ISO/IEC 25010<sup>16</sup>, die Maßgaben des Clean Code<sup>17</sup> und weitere Design Constraints, die sich auf das Design der UI beziehen, wie z. B. das „Paddeln, Schwimmen & Tauchen“-Prinzip<sup>18</sup>.

Zur Implementierung der mobilen Anwendung wurde die Entwicklungsumgebung Android Studio<sup>19</sup> und die Programmiersprache Java verwendet. Außerdem wurde der Code mithilfe von JavaDocs kommentiert, um die Wartbarkeit zu verbessern. Der Prototyp der Anwendung fokussiert sich vor allem auf ein funktionelles UI und die Umsetzung der Funktionalitäten, da dies eine anschließende Evaluierung der Funktionalitäten und damit des Nutzens der Anwendung ermöglichte.

## Ergebnisse

Die Ergebnisse der Architekturbeschreibung wurden in Form einer Software Architecture Description festgehalten, deren wichtigste In-

halte im Folgenden dargestellt werden. Der Systemkontext besteht aus dem Server, auf dem die Anwendung Daten speichert, und den Nutzern, die in die Gruppen Scrum Master, Product Owner und Developer eingeteilt werden. Diese Einteilung wurde getroffen, da jede der Nutzergruppen anders mit der Anwendung interagiert, so ist z. B. die einzige Eingabe eines Product Owners Feedback, während der Scrum Master die Team Charta eingibt.

Die Komponenten, die im Komponentendiagramm dargestellt wurden, sind:

- der Controller, der für die Kommunikation mit dem Server zuständig ist,
- das Model, in dem die Daten der Anwendung zur Laufzeit gespeichert werden und
- der View, der für das UI zuständig ist, mit den Unterkomponenten
- Navigation,
- Fragmente Schätzen, Feedback und Team Charta sowie
- die Viewkomponente.

Durch das Klassendiagramm wurden die Datenobjekte dargestellt. Diese ist in Abbildung 2 zu sehen.

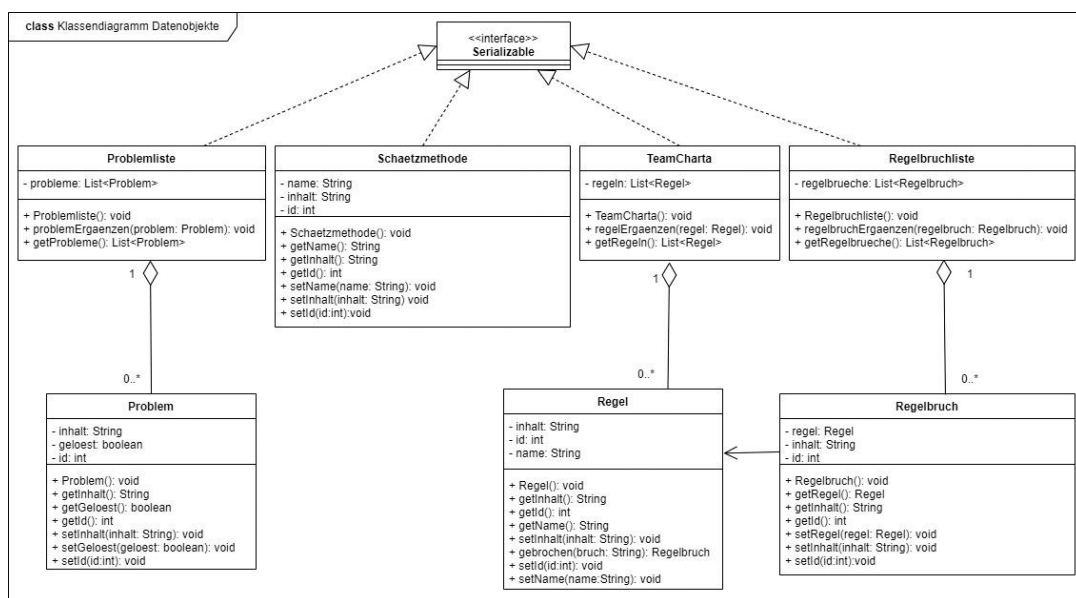


Abbildung 2: Klassendiagramm der Datenobjekte

13 vgl. Starke (2020), S. 144 ff.

14 vgl. Kecher u.a. (2018), S. 225 f.

15 vgl. Fischer (2020), S. 2.

16 vgl. ISO/IEC (2011), S. 6 ff.; Die ISO/IEC 25010 ist eine Norm, die sich mit der Qualität von Softwareprodukten beschäftigt.

17 vgl. Martin (2009), S. 45 ff.

18 vgl. Scholz u.a. (2014), S. 24 f.

19 Android Studio ist eine von Google bereitgestellte Entwicklungsumgebung für Android-Anwendungen, in der die Programmiersprachen Java und Kotlin verwendet werden können. Mehr Informationen zu finden unter: <https://developer.android.com/studio?hl=de> (letzter Zugriff: 07.07.2022)

Durch ADRs wurden die folgenden wichtigen Entwurfsentscheidungen festgehalten:

- die Client-Server-Architektur,
- das Model-View-Controller-Muster sowie
- Kommunikationsprotokoll.

Der Prototyp der Anwendung bildet die geforderten Funktionalitäten ab und diese wurden von Nutzern während der Tests als positiv wahrgenommen. Die Umsetzung dieser wurde jedoch teilweise kritisiert, da es z. B. keine Möglichkeit gibt die Schätzmethode und Regeln direkt zu bearbeiten. Mit dieser Einschätzung wurden zwar Verbesserungen an der genauen Umsetzung bei der Implementierung gefordert, das Konzept der Anwendung und die Funktionalitäten wurden jedoch als positiv bewertet.

#### Fazit

In dieser Arbeit wurden eine Architekturbeschreibung und der Prototyp für eine Anwendung erstellt. Hierbei wurde das Design des Design Science verwendet und die Forschungsfragen beantwortet. Die Antworten auf die Forschungsfragen werden im Folgenden dargestellt.

Die wichtigsten Erkenntnisse der Architekturbeschreibung sind:

- dass eine Client-Server-Architektur gewählt wird und
- dass die Anwendung anhand des Model-View-Controller-Prinzips aufgebaut ist.

Die wichtigsten Vorteile der Anwendung sind die offenere Fehlerkultur durch die Anonymität und der einfachere Zugang zu allen wichtigen Daten wie Team Charta und Schätzmethode.

Die einzige Qualitätsanforderung, die vom Prototyp nicht erfüllt wurde, war die der Zuverlässigkeit, da es bei den Tests zu einem Absturz kam, wenn eine Regel gelöscht werden sollte, obwohl noch keine Regel angelegt war.

Der Verbesserungsbedarf der Anwendung bezieht sich vor allem auf die Darstellung im UI und die Möglichkeit die Schätzmethode und Regeln zu bearbeiten.

#### Literatur

Fischer, Oliver (2020): Gut dokumentiert: Architecture Decision Record, in: heise online.

Gloger, Boris (2014): Wie schätzt man in agilen Projekten. oder wieso Scrum-Projekte erfolgreicher sind, München.

ISO/IEC (2011): Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models, ISO/IEC 25010:2011, Genf.

Johannesson, Paul/Perjons, Erik (2014): An Introduction to Design Science, Heidelberg.

Kecher, Christoph/Salvanos, Alexander/Hoffmann-Elbern Ralf (2018): UML 2.5 Das umfassende Handbuch, 6. Aufl., Bonn.

Martin, Robert C. (2009): Clean Code. Refactoring, Patterns, Testen und Techniken für sauberen Code, Heidelberg/München/Landsberg/Frechen/Hamburg.

Olfert, Klaus (2019): Projektmanagement, 11. Aufl., Herne.

Pichler, Roman (2008): Scrum. Agiles Projektmanagement erfolgreich einsetzen, Heidelberg.

Scholz, Heike u.a. (2014): Android-Apps entwickeln. Konzeption, Programmierung und Vermarktung, Haar bei München.

Schwaber, Ken/Sutherland, Jeff (2020): Der Scrum Guide, URL: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-German.pdf>, Stand: 30. Juli 2021.

Starke, G. (2020): Effektive Software-Architekturen, 9. Aufl.

Yin, Robert K. (2014): Case Study Research. Design and Methods, 5. Aufl., California.

#### Abkürzungsverzeichnis

ADR	Architecture Decision Record
UML	Unified Modeling Language
UI	User-Interface