



Fakultät für Informatik
der Technischen Universität
München

Natürliche Benutzerschnittstellen im Umfeld verteilter Projektarbeit

Florian Christian Raith

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität
München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender:

Prof. Dr. Georg Groh

Prüfer der Dissertation:

1. Prof. Gudrun J. Klinker, Ph.D.

2. Prof. Dr. Robert Lindermeier

Diese Dissertation wurde am 02.08.2021 bei der Technischen Universität München
eingereicht und durch die Fakultät für Informatik am 04.11.2021 angenommen.

Abstract

In current software development a strong trend towards agile methodologies (e.g. like Scrum, XP) can be observed, as companies want to benefit from more flexible requirements engineering, shorter time to market and better software quality. At the same time, the number of locally distributed software development projects rises as companies want to benefit from location advantages like lower personnel costs and better availability of professionals. These trends are contradicting as agile methodologies make use of concepts to encourage direct interaction and communication amongst project stakeholders, which is much more difficult in the context of distributed development. This contradicting situation also builds on common project management solutions and tools: A basic concept of agile methodologies is the usage of physical media (e.g. flip-charts, partition-walls, etc.) to track development progress and to manage requirements. The physical nature of these tools is important in order to improve transparency, motivation and communication within locally working project teams. In distributed environments, locally working project teams are forced to use digital project management software as distant project teams need to access project data to align joint planning and development. Literature proves negative effects on motivation, transparency and communication within those project teams caused by substituting project management software for physical media.

This research work addresses the mentioned conflict from a computer supported collaborative work (CSCW) perspective and tries to combine the capabilities of modern natural user interfaces with the necessity to integrate digital project management software into distributed project setups. One goal of this research is to understand if natural user interfaces can mitigate negative effects on teamwork caused by the usage of browser based project management software. Moreover, a concept to enhance state-of-the-art project management software by integrating natural user interfaces has been acquired and evaluated. In order to achieve these goals, an interview study with agile experts has been conducted as a first step. The aim was to understand usage of state-of-the-art project management software in distributed projects and to learn about common problems in that area. Based on the acquired knowledge, a concept to integrate a tangible user interface (as representative of natural user interfaces) into conventional, browser-based project management software has been defined and implemented prototypically. Goal of this concept was to combine the physical nature of tangible user interfaces with the ability of project management software to provide relevant project data to distant teams.

The evaluation setup was based on a comparative study with two agile development teams using conventional project management software with a browser-based user interface on the one hand and the acquired prototype providing a tangible user interface on the other hand. We defined measurement models to be able to assess the suitability of both approaches in terms of agile software development. In summary it can be stated, that we found considerable improvements regarding the quality of teamwork caused by the usage of natural user interfaces instead of browser-based user interfaces.

Inhaltsverzeichnis

1	Einleitung	7
1.1	Motivation	7
1.1.1	Von der klassischen Softwareentwicklung zu agilen Methoden . . .	8
1.1.2	Ein konträrer Trend - Agile Methoden in verteilten Projekten . .	8
1.1.3	Digitale Projektmanagement Software als Ersatz für physikalische Medien	9
1.1.4	Chancen moderner Informationstechnologie	10
1.2	Forschungsproblem	10
1.3	Forschungsziele	10
1.4	Struktur der Arbeit	11
2	Forschungsdesign und Methodenauswahl	15
2.1	Design Science - Übergreifender Forschungsansatz	15
2.2	Forschungsmethodik von Vorstudie und Konzeptevaluation	18
2.2.1	Untersuchungsplan der Vorstudie - Die Einzelfallanalyse	19
2.2.2	Untersuchungsverfahren der Vorstudie	20
2.2.3	Untersuchungsplan der Konzeptevaluation - Das qualitative Experiment	23
2.2.4	Untersuchungsverfahren der Konzeptevaluation	25
3	Theoretische Grundlagen und Stand der Wissenschaft	35
3.1	Grundlagen agiler Entwicklung	35
3.1.1	Das agile Manifest - Wertebasis agiler Softwareentwicklung	36
3.1.2	Agile Methoden	37
3.1.3	Agile Arbeitsumgebungen	43
3.1.4	Digitale Projektmanagement-Tools zur Unterstützung agiler Softwareentwicklung	51
3.1.5	Neuartige Ansätze zur Verbesserung agiler Projektmanagement Software (PMS)	51
3.2	Forschungsfeld Computer Supported Cooperative Work (CSCW)	53
3.2.1	Klassifikation von CSCW Anwendungen und Einordnung der Arbeit	54
3.2.2	Eigenschaften Natürlicher Benutzerschnittstellen	55
3.3	Forschungsgegenstand	56
3.4	Konkretisierung des Forschungsproblems	56
3.4.1	Beitrag zur Wissenschaft	57

4	Vorstudie - Einsatz digitaler Projektmanagement-Software in der Praxis	59
4.1	Ziele der Studie	59
4.2	Forschungsplan - Einzelfallanalyse	59
4.2.1	Fragestellung	60
4.2.2	Falldefinition	60
4.2.3	Methodenauswahl	62
4.2.4	Datenaufbereitung	62
4.2.5	Einordnung in den Zusammenhang - Ergebnisse	65
4.2.6	Verbesserungsmöglichkeiten als Schlüsselanforderungen an das Lösungskonzept	67
5	Lösungskonzept - Virtual Task-Board	69
5.1	Konkretisierung der Zieldefinition	69
5.2	Anforderungsanalyse	69
5.2.1	Funktionale Anforderungen	70
5.2.2	Nichtfunktionale Anforderungen	72
5.3	Grundidee - Nahtlose Integration konventioneller PMS mit Hilfe natürlicher Benutzerschnittstellen	73
5.4	Lösungsidee zur automatisierten Erfassung relevanter Projektdaten	75
5.4.1	Erfassen der handschriftlichen Notizen auf den Story-Cards	76
5.4.2	Positionserfassung und Identifikation von Story-Cards auf dem Task-Board	84
5.5	Lösungsidee zur Anbindung konventioneller PMS	89
5.6	Überlegungen zum Systemaufbau	90
5.7	Realisierung eines Prototypen	91
5.7.1	Technische Beschreibung der Komponente "Smartphone-Anwendung"	92
5.7.2	Technische Beschreibung der Anwendung "Marker Tracking"	94
5.7.3	Technische Beschreibung der Komponente "Interpretation und Synchronisation"	96
6	Konzeptevaluation	101
6.1	Forschungsziele und Hypothesen	101
6.2	Studienaufbau	103
6.2.1	Bewertungsmodell	105
6.2.2	Datenerhebung und Protokollierung der Beobachtungsstudie	109
6.2.3	Datenerhebung der schriftlichen Befragung	109
6.2.4	Datenerhebung mit der System Usability Scale	113
6.3	Ergebnisse	113
6.3.1	Ergebnisse der Beobachtungsstudie	113
6.3.2	Ergebnisse der schriftlichen Befragung	121
6.3.3	Ergebnisse der System-Usability-Scale	126
6.3.4	Zusammenführung der Teilergebnisse und Hypothesenbezug	127
6.3.5	Limitationen der Evaluationsstudie	128

7 Zusammenfassung und Ausblick	131
7.1 Stellungnahme zu Forschungszielen und Arbeitsergebnissen	131
7.2 Inhalte der Arbeit	133
7.3 Konferenzbeiträge und Publikationen im Kontext der Arbeit	135
7.4 Ausblick	136
Literaturverzeichnis	139
Abbildungsverzeichnis	145
Tabellenverzeichnis	147
Abkürzungsverzeichnis	149
A Vorstudie	151
A.1 Ausschnitt eines Transkripts	151
A.2 Leitfragen zur Vorstudie	152
A.3 Ablaufdiagramme aller in der Vorstudie untersuchten Projekte	154
B Konzeption	159
B.1 Druckvorlagen Story-Cards	159
B.1.1 Druckvorlage Epic	159
B.1.2 Druckvorlage Story	160
B.1.3 Druckvorlage Task	160
B.2 Konfigurationsparameter der Softwarekomponente "Marker-Tracking" . .	161
B.2.1 Erläuterung der Konfigurationsparameter	161
B.2.2 Beispielkonfiguration der Komponente Marker-Tracking	162
B.2.3 Übertragungsparameter im JSON-Format zur Komponente Interpretation und Synchronisation	162
C Beobachtungsstudie	163
C.1 Vorlage Beobachtungsprotokoll	163
C.2 Indexblatt	163
C.3 Fragenkatalog der System Usability Scale	165

Kapitel 1

Einleitung

1.1 Motivation

Spätestens seit Edsger W. Dijkstra in seiner Dankesrede zum Turing Award² im Jahr 1972 auf die stark angestiegene Komplexität und die damit zusammenhängenden Qualitätsprobleme in der Softwareentwicklung verwies, rückte die Wichtigkeit strukturierter und prozessorientierter Softwareentwicklung in das Bewusstsein der IT-Branche. Dijkstra äußerte sich folgendermaßen zu den Problemen in der Softwareentwicklung und beschrieb die Situation der Branche damit sehr treffend:

"[...]To put it quite bluntly: as long as there were no machines, programming was no problem at all; when we had a few weak computers, programming became a mild problem, and now we have gigantic computers, programming has become an equally gigantic problem. [...]"[24]

Obwohl mit Hilfe unterschiedlichster Prozesse und Entwicklungsmodelle erhebliche Fortschritte und Erfolge in diesem Bereich erzielt werden konnten hat Dijkstra Aussage bis heute uneingeschränkte Gültigkeit, da die Leistungsfähigkeit moderner Computersysteme und gleichzeitig auch die Komplexität heutiger Softwarelösungen weiter ansteigt. Spätestens seit dieser Zeit versucht die Wissenschaft durch die Entwicklung und Anwendung von Prozessen diese Probleme zu adressieren und zu verbessern. Der Begriff des Software Engineerings wurde in diesem Zusammenhang geprägt. Das Institute of Electrical and Electronics Engineers (IEEE) definiert Software Engineering wie folgt:

Definition 1.1. *Software Engineering:* *"The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software." IEEE Standard Glossary of Software Engineering Terminology [32]*

²Der Turing Award ist eine der höchsten Auszeichnungen in der Informatik. Er wird von der Association for Computing (ACM) verliehen und richtet sich an Personen, die sich in besonderer Art und Weise um die Informatik als Wissenschaft verdient gemacht haben.

1.1.1 Von der klassischen Softwareentwicklung zu agilen Methoden

Ein grundlegender Bestandteil des Software Engineerings sind Vorgehensmodelle wie z.B. das Wasserfallmodell [10] oder das V-Modell [36]. Die genannten Vertreter sind sog. plangetriebene oder auch klassische Vorgehensmodelle, die ein rein sequentielles Vorgehen über mehrere Entwicklungsstufen hinweg von der Anforderungsdefinition über die Implementierungsphase bis hin zur Auslieferung und Wartung der Software vorsehen. Plangetriebene Vorgehensmodelle waren in den 90er und 2000er Jahren Stand der Wissenschaft und dominierten die Softwareentwicklung vor allem in größeren Projekten. Der 1995 veröffentlichte Chaos Report der Standish Group [42] zeigte jedoch, dass die Softwarebranche nach wie große Probleme hatte und nur wenige Projekte erfolgreich abgeschlossen wurden. Laut Standish Group wurden im Jahr 1994 ca. 16% der Softwareprojekte innerhalb des geplanten Zeit- und Kostenrahmens abgeschlossen. 53% überschritten diesen Rahmen oder schafften nicht den vorgesehenen Funktionsumfang und 31% wurden vorzeitig abgebrochen. Als Hauptgrund der nicht erfolgreich abgeschlossenen Projekte ermittelte die Standish Group mangelnde Einbeziehung der Benutzer in die Entwicklung. Dieses Problem ist der sequentiellen Struktur der plangetriebenen Vorgehensmodelle geschuldet. Nach der Anforderungsdefinition ist eine weitere Benutzerbeteiligung nicht mehr vorgesehen und missverständliche oder falsche Anforderungen wurden häufig erst zu spät aufgedeckt.

Dieses Problem wurde gegen Ende der 90er Jahre durch eine neue Art der Softwareentwicklung, den sog. agilen Methoden adressiert. Sie setzen die Zufriedenheit des Kunden als oberstes Ziel und stellen Teamarbeit, Transparenz, direkte Kommunikation und Softwarequalität in den Mittelpunkt der Bemühungen. Diese Grundprinzipien wurden 2001 im agilen Manifest [9] zusammengetragen. Der Begriff "agil" steht dabei für eine agile Entwicklung der Anforderungen. Der Auftraggeber wird während der kompletten Projektlaufzeit eng in den Entwicklungsprozess eingebunden und kann auf Basis zahlreicher Zwischenergebnisse beurteilen, ob das entstehende Produkt seinen Anforderungen entspricht, um diese ggf. anzupassen. Artefakte der plangetriebenen Vorgehensmodelle (z.B. Lastenheft, Pflichtenheft) werden durch direkte Gespräche zwischen Projektteam und Auftraggebern ersetzt, um Missverständnissen vorzubeugen und Variabilität bzgl. der Anforderungen zu ermöglichen.

1.1.2 Ein konträrer Trend - Agile Methoden in verteilten Projekten

Seit der Veröffentlichung des agilen Manifests hat der Anteil agil durchgeführter Softwareprojekte stetig zugenommen. Dieser Trend zeigt sich deutlich im "State of Agile" der Firma Version One. Sie veröffentlicht seit 2006 einen jährlichen Bericht, der die Verbreitung agiler Methoden sowie die Erfahrungswerte der Firmen mit agilen Methoden darlegt. Für den aktuellsten Bericht aus dem Jahr 2018 [57] wurden insgesamt knapp 1500 Personen aus gezielt ausgewählten Branchen bzw. Unternehmen befragt. Folgende Zahlen sollen

die gegenwärtige Bedeutung agiler Methoden für die professionelle Softwareentwicklung verdeutlichen:

- 97% der Firmen setzen agile Methoden in ihren Unternehmen ein
- 52% der befragten Firmen setzen agile Methoden in mehr als der Hälfte ihrer Projekte ein
- 71% der Befragten nannten die Möglichkeit mit veränderlichen Anforderungen und Prioritäten umzugehen als wichtigsten Grund für den Einsatz agiler Methoden

Neben einem stark ausgeprägten Trend zu agilen Methoden verdeutlicht der "State of Agile" einen weiteren Trend in der IT-Branche: 79% der Befragten gaben an, einen Teil der agil durchgeführten Projekte mit Hilfe örtlich verteilter Projektteams durchzuführen. Die wichtigsten Gründe für diese Entwicklung sind Kundennähe, Einsparungen bei Personal und Betriebsmitteln sowie Produktivitätssteigerungen [1].

Das Hauptproblem dieser beiden Trends ist, dass sie nur sehr schwer vereinbar sind. Agile Methoden erfordern direkte Gespräche von Angesicht zu Angesicht, um bestmögliche Kommunikation und Transparenz zu schaffen [9]. Weiterhin spielen physikalische Medien (z.B. Stellwände, Flipcharts) eine bedeutende Rolle [56, 38]. Sie dienen neben der Visualisierung relevanter Projektdaten vor allem zur Förderung von Kommunikation, Kreativität sowie Transparenz und sollten fester Bestandteile einer agilen Arbeitsumgebung sein [8]. Durch die örtlich verteilte Projektarbeit können diese grundlegenden Praktiken agiler Methoden nicht angewandt werden. Direkte Gespräche finden deutlich weniger statt, da die Kommunikation auf elektronische Hilfsmittel (z.B. Telefon, Email, Chat-Programme) ausgelagert wird. Außerdem finden physikalische Medien im Rahmen verteilter Projektarbeit häufig kaum noch Anwendung, weil relevante Informationen digital vorliegen müssen, um auch für entfernte Projektmitarbeiter bzw. Projektteams einsehbar zu sein.

1.1.3 Digitale Projektmanagement Software als Ersatz für physikalische Medien

Agile Methoden setzen auf papierbasierte Medien, um Anforderungen zu verwalten, den Entwicklungsfortschritt zu visualisieren, Entwicklungsphasen zu planen und die Aufgabenteilung aufzuzeigen. In verteilten Projekten muss der Austausch dieser Informationen über weite Distanzen gewährleistet sein, weshalb papierbasierte Medien weitestgehend durch digitale Projektmanagement Software (PMS) ersetzt werden. Bekannte Vertreter für PMS sind z.B. "Jira Agile" der Firma Atlassian [40], die Open Source Software "Redmine" [60] oder die Software "YouTrack" der Firma JetBrains [86]. Allen gemeinsam ist, dass es sich um Web-Applikationen mit browserbasierter Benutzeroberfläche handelt. Der Einsatz dieser Tools hat jedoch weitreichende Auswirkungen auf die Gruppenarbeit im Team. Zahlreiche Studien belegen negative Effekte auf die Kommunikation, die Transparenz und den Informationsfluss innerhalb agiler Projektteams [63, 46, 76, 78, 69]. Im Rahmen dieser Arbeit werden digitale Projektmanagementlösungen betrachtet, die zur Unterstützung von agilen Softwareentwicklungsprozessen konzipiert wurden. Dies

beinhaltet beispielsweise Funktionen zur Anforderungsverwaltung, zur Erfassung und Visualisierung des Projektfortschritts, zur Erledigung von Planungsaufgaben und zur nachvollziehbaren Dokumentation der Projektdaten. Kommunikationsanwendungen zur Durchführung von Videokonferenzen, Instant-Messenger (z.B. Zoom, Skype for Business, etc.) fallen nicht in diese Kategorie und werden deshalb nicht betrachtet.

1.1.4 Chancen moderner Informationstechnologie

Eine mögliche Erklärung für die in Abs. 1.1.3 genannten Auswirkungen von PMS auf die Gruppenarbeit ist der Einsatz grafischer Benutzeroberflächen (engl. *Graphical User Interface* (GUI)). Im Vergleich zu physikalischen Medien unterstützen sie kognitive Prozesse weniger gut (Stichwort Embodiment¹) und bringen Einschränkungen bei der gleichzeitigen Benutzung eines Systems durch mehrere Anwender (engl. *Multi User Interaction* (MUI)) mit sich [26]. An diesen und weiteren Schwachpunkten konventioneller PMS soll im Rahmen dieser Arbeit angeknüpft werden (vgl. 1.3). Die moderne Informationstechnologie bietet mittlerweile zahlreiche Möglichkeiten, um Benutzerschnittstellen von Anwenderprogrammen natürlicher und intuitiver zu gestalten. So erlauben sog. *Tangible-User-Interfaces* (TUIs) die Verknüpfung physikalischer Gegenstände mit digitalen Elementen, um kognitive Prozesse bestmöglich zu unterstützen.

Aus wissenschaftlicher Sicht kann dieser Bereich und damit auch diese Arbeit dem Forschungsfeld *Computer Supported Collaborative Work* (CSCW) zugeordnet werden.

1.2 Forschungsproblem

Die einleitenden Abschnitte verdeutlichen, dass die moderne Softwareentwicklung momentan von zwei gegensätzlichen Trends beeinflusst wird. Zum einen setzen sich agile Entwicklungsmethoden gegenüber plangetriebenen Verfahren immer stärker durch. Zum anderen wird ein immer größerer Anteil von Softwareentwicklungsprojekte örtlich verteilt durchgeführt. Dieser Umstand führt dazu, dass elementare Elemente agiler Methoden wie physikalische Medien durch digitale PMS ersetzt werden und die Qualität der Gruppenarbeit bzgl. Kommunikation, Transparenz und Motivation abnimmt. Diese drei Aspekte zählen jedoch zu den Grundwerten agiler Softwareentwicklung [9] und sind essentiell für die erfolgreiche Durchführung entsprechender Projekte.

1.3 Forschungsziele

Wie in Abs. 1.1.4 verdeutlicht wurde, sollen die Möglichkeiten moderner Informationstechnologie zur Optimierung konventioneller PMS untersucht werden. Das übergreifende Ziel der Arbeit (**Z**) ist ein Konzept zur Optimierung konventioneller PMS für agile

¹Die Kognitionswissenschaft versteht unter Embodiment die Annahme, dass Bewusstsein einen Körper und damit eine physikalische Interaktion voraussetzt. [26]

Entwicklungsprojekte auf Basis natürlicher Benutzerschnittstellen, um die Gruppenarbeit hinsichtlich Kommunikation, Transparenz und Motivation innerhalb agiler Teams zu verbessern. Im Wesentlichen soll dieses Ziel durch den Erhalt physikalischer Benutzerinteraktion und die damit einhergehenden kognitiven Prozesse erreicht werden, ohne die Vorteile konventioneller PMS für örtlich verteilte Projektarbeit einzuschränken. Um das übergeordnete Ziel zu erreichen, wurden folgende Zwischenziele definiert:

- **Z1:** Es soll erforscht werden, wie konventionelle PMS im Rahmen verteilter, agiler Entwicklungsprojekte eingesetzt wird, um eine Wissensbasis zur Erarbeitung des Lösungskonzepts bereitzustellen.
 - **Z1.1:** Erfassung der jeweiligen Arbeitsabläufe, die mit Hilfe von PMS in den einzelnen Phasen/Besprechungen des agilen Prozesses ausgeführt werden.
 - **Z1.2:** Erfassung der Vor- und Nachteile konventioneller PMS in agilen Entwicklungsprojekten aus Anwendersicht.
 - **Z1.3:** Identifikation konkreter Verbesserungspotenziale konventioneller PMS zur Optimierung der Gruppenarbeit in agilen Teams durch den Einsatz natürlicher Benutzerschnittstellen.
- **Z2:** Konzeption einer natürlichen Benutzerschnittstelle zur automatisierten Erfassung projektrelevanter Daten im Kontext verteilter, agiler Projekte.
 - **Z2.1:** Konzeptionelle Betrachtung auf Basis fachlicher Anforderungen und unter Berücksichtigung der technischen Realisierbarkeit.
 - **Z2.2:** Technische Konzeption und prototypische Implementierung der erarbeiteten natürlichen Benutzerschnittstelle.
- **Z3:** Evaluation des Gesamtkonzepts (vgl. Z) im Rahmen einer Benutzerstudie, um die Wirksamkeit der in Z1.3 erarbeiteten Verbesserungspotenziale hinsichtlich Kommunikation, Transparenz und Motivation zu eruieren.

1.4 Struktur der Arbeit

Die vorliegende Arbeit besteht (inklusive diesem) aus insgesamt sieben Kapiteln. Nachfolgend werden die Inhalte von Kapitel 2 bis 7 kurz zusammenfasst, um die Struktur der Arbeit zu veranschaulichen:

In **Kapitel 2** werden die im Rahmen der Arbeit angewandten Forschungsmethoden erklärt. Dabei werden verschiedene Varianten und Ausprägungen aufgezeigt, um die getroffene Auswahl geeigneter Methoden für den Leser nachvollziehbar zu machen. Es ist zu beachten, dass es sich zunächst um eine allgemeine Beschreibung der Methoden handelt. Die konkrete Anwendung wird in den Kapiteln zur Vorstudie (Kap. 4) und zur Konzeptevaluation (Kap. 6) beschrieben.

In **Kapitel 3** werden für die Arbeit relevante Forschungsbereiche vorgestellt und der jeweilige Stand der Wissenschaft auf Basis verwandter Arbeiten und wissenschaftlicher Publikationen dargelegt. Das Kapitel beginnt mit einer Einführung in agile Entwicklungsmethoden und Prozesse. Dabei werden alle Prozessschritte und Besprechungen vorgestellt, die allen agilen Methoden (unabhängig von der konkreten Ausprägung) zugrunde liegen. Anschließend wird auf die Besonderheiten verteilter, agiler Projekte eingegangen. Der zweite Teil des Kapitels beschäftigt sich mit CSCW als eigenes Forschungsfeld. Dabei steht die aktuelle Forschung im Bereich natürlicher Benutzerschnittstellen im Fokus. Abschließend wird der Beitrag dieser Arbeit zur Wissenschaft aufgezeigt und von bestehenden Forschungsarbeiten abgegrenzt.

Inhalt von **Kapitel 4** sind Konzeption, Durchführung und Ergebnisse einer Interviewstudie, die zum Aufbau einer Wissensbasis bzgl. PMS in verteilten, agilen Projekten (vgl. Z1 in Abs. 1.3) durchgeführt wurde. Dabei wurden erfahrene Experten aus der Praxis zum Einsatz konventioneller PMS im Rahmen ihrer laufenden Projekte befragt. Ergebnisse der Studie sind Arbeitsabläufe in Form von Ablaufdiagrammen, die den Einsatz von konventioneller PMS in Kombination mit physikalischen Medien zeigen. Des Weiteren werden Vorteile, Nachteile und Verbesserungspotentiale konventioneller PMS ausgearbeitet. Auf diesen Grundlagen werden dann Verbesserungspotenziale konventioneller PMS und Schlüsselanforderungen an das Lösungskonzept definiert.

Anschließend in **Kapitel 5** folgt die konzeptionelle Betrachtung, sowie die prototypische Umsetzung des Lösungsartefakts. Die Grundidee basiert auf der Erweiterung konventioneller PMS mit Hilfe natürlicher Benutzerschnittstellen, um die in Abs. 1.1.3 genannten Beeinträchtigungen von Kommunikation, Motivation und Informationsfluss im Team zu reduzieren. Dabei werden die Ergebnisse der Vorstudie (vgl. Kap. 4) sowie die theoretischen Grundlagen (vgl. Kap. 3) herangezogen, um die Anforderungen an das Lösungskonzept mit Blick auf die festgelegten Forschungsziele zu spezifizieren. Auf Basis dieser Anforderungen werden anschließend die einzelnen Bestandteile des Lösungskonzepts beschrieben.

Die prototypische Umsetzung ist ebenfalls Teil des Kapitels. Ziel der Umsetzung war es, einen benutzbaren Prototypen auf Basis einer natürlichen Benutzerschnittstelle bereitzustellen, der im Rahmen der Konzeptevaluation genutzt werden kann. Hierfür wurde Jira Agile als Vertreter konventioneller PMS entsprechend erweitert. Es sollte gezeigt werden, dass die Vorteile (wie z.B. dezentrale Verfügbarkeit der Daten) konventioneller PMS mit den Vorteilen physikalischer Medien kombiniert werden können. Das Kapitel konzentriert sich auf die technische Umsetzung des Konzepts, weshalb die Gesamtarchitektur des Prototypen sowie der Aufbau der einzelnen Software- und Hardwarekomponenten aufgezeigt wird.

Kapitel 6 beschreibt die Evaluation des Lösungskonzepts. Hierfür wurden zunächst geeignete Hypothesen auf Basis der in Abs. 1.3 erläuterten Forschungsziele ausgearbeitet. Anschließend werden Studienaufbau, die konkrete Anwendung der Erhebungstechniken und ein geeignetes Bewertungsmodell erläutert. Im weiteren Verlauf des Kapitels folgt

eine detaillierte Präsentation der Ergebnisse und eine Stellungnahme zu den eingangs formulierten Hypothesen.

Kapitel 7 enthält eine Zusammenfassung der einzelnen Forschungsziele, der durchgeführten Studien, der Konzeptarbeit sowie eine kurze Zusammenführung der Forschungsergebnisse. Weiterhin werden die im Rahmen dieser Forschungsarbeit entstandenen wissenschaftlichen Veröffentlichungen noch einmal aufgeführt. Abschließend wird ein kurzer Ausblick über weiterführende Forschungsfragen und Ideen auf Basis der erarbeiteten Ergebnisse aufgeführt.

Kapitel 2

Forschungsdesign und Methodenauswahl

Diese Arbeit orientiert sich am lösungsorientierten Forschungsansatz *Design Science Research Process* (DSRP) nach Peffers et al. [62]. Dem DSRP liegt das *Design Science* (DS) Paradigma für Informationssysteme nach Hevner et al. [35] zugrunde. In Abs. 2.1 wird die Grundidee von DS erläutert und dem Forschungsparadigma *Behavioral Science* (BS) [30] gegenüber gestellt. Weiterhin werden die Prozessschritte von DSRP erläutert und den entsprechenden Bereichen dieser Arbeit zugeordnet. Anschließend werden die Forschungsmethoden von Vorstudie und Konzeptevaluation im Detail vorgestellt. Im Rahmen der Vorstudie werden sog. Einzelfallanalysen (vgl. Abs. 2.2.1) mit Hilfe Problemzentrierter Interviews (vgl. Abs. 2.2.2) als zentrale Erhebungsverfahren eingesetzt. Die Konzeptevaluation basiert auf einem Qualitativen Experiment (vgl. Abs. 2.2.3) und nutzt eine Beobachtungsstudie als zentrales Verfahren zur Datenerhebung, welches um eine schriftliche Befragung und um eine Bewertung mittels der sog. *System Usability Scale* (SUS) ergänzt wurde (vgl. 2.2.4). Alle eben aufgezählten Verfahren werden im Rahmen dieses Kapitels allgemein erläutert und in den übergreifenden Kontext sozialwissenschaftlicher Forschungsmethoden eingeordnet. Dieser Schritt ist notwendig, um die Auswahl der entsprechenden Methoden nachvollziehbar darstellen zu können. Die konkrete Anwendung der ausgewählten Methoden wird in separaten Kapiteln zur Vorstudie (vgl. Kap. 4) bzw. zur Konzeptevaluation (vgl. Kap. 6) beschrieben.

2.1 Design Science - Übergreifender Forschungsansatz

DS ist ein lösungsorientiertes Forschungsparadigma, dessen Grundidee die Entwicklung und Evaluation zielgerichteter Lösungen für konkrete Problemstellungen aus der Praxis ist. Es setzt sich damit deutlich vom BS Paradigma ab, dessen Ziel die Entwicklung und Evaluation von Theorien zur Erklärung bestimmter Verhaltensweisen (z.B. von Menschen oder Organisationen) darstellt. Hevner et al. veranschaulicht diese beiden Ansätze indem der als Ziel der BS die Wahrheit in Form einer Theorie und als Ziel der DS den praktischen Nutzen in Form eines innovativen oder neuartigen Artefakts (z.B. Software) angibt.

Peffers et al. [62] hat mit dem DSRP ein Vorgehen zur Anwendung des DS Paradigmas

entwickelt, das im Forschungsumfeld der Informationssysteme mittlerweile breite Anwendung findet. Im Folgenden werden die insgesamt sechs Prozessschritte (vgl. Abb. 2.1) erläutert und mit den jeweiligen Abschnitten der Arbeit in Bezug gesetzt.

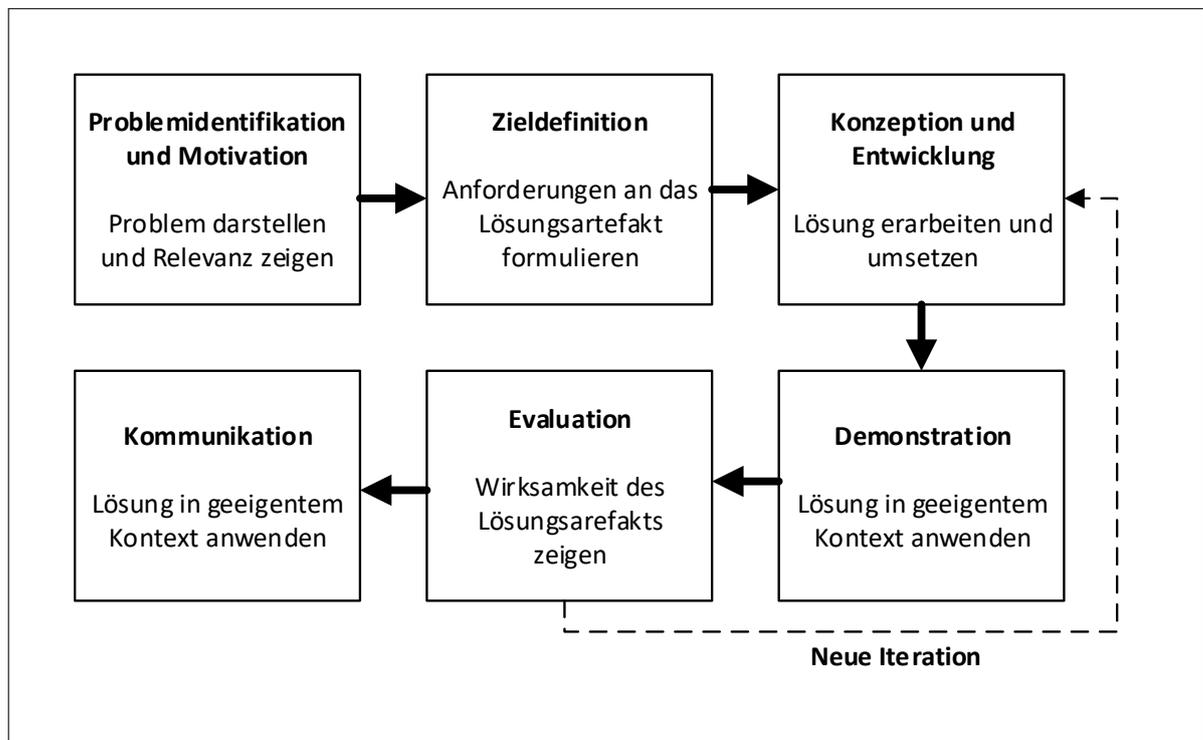


Abbildung 2.1: Design Science Forschungsansatz in Anlehnung an Peffers et al. [62] S.199

Schritt 1 - Problemidentifikation und Motivation: Am Anfang des DSRP steht die Identifikation sowie die genaue Definition des Problems, das gelöst werden soll. Laut Peffers et al. [62] soll zunächst der Mehrwert einer potentiellen Lösung ausgearbeitet werden, um sowohl die Forschenden als auch Interessenten für das Forschungsprojekt zu motivieren. Außerdem ist dieser Schritt notwendig, um die Forschungsergebnisse sowie alle Zwischenschritte nachvollziehen und weiterverwenden zu können. Zusätzlich rät Peffers et al. zur Zerlegung komplexer Problemstellungen in kleine verständlichere Teile gemäß dem Grundsatz "Teile und Herrsche"¹. Neben der reduzierten Komplexität der Teilprobleme erleichtert dieses Vorgehen die Planung des Gesamtprojekts und ermöglicht die Veröffentlichung verwertbarer Zwischenergebnisse bereits vor Abschluss des Forschungsvorhabens.

Im Kontext der Arbeit: Dieser Teil des DSRP wird zunächst durch Kap. 1 aufgegriffen.

¹Der Ansatz "Teile und Herrsche"(lat. divide et impera) sieht es vor, ein Problem rekursiv in Teilprobleme zu zerlegen bis deren Komplexität ausreichend reduziert und damit beherrschbar ist. Aus den Teillösungen kann anschließend eine Gesamtlösung für das übergeordnete Problem konstruiert werden.

Zur Motivation wird das zentrale Problem - die Unvereinbarkeit agiler Prinzipien mit verteilter Softwareentwicklung - dargestellt und gleichzeitig auf die Chancen moderner Lösungsansätze aus dem Forschungsbereich CSCW verwiesen. Die Relevanz des Forschungsproblems wird anschließend in 3.1.3.1 weiter verdeutlicht. Es enthält einen Überblick verwandter Arbeiten, die das genannte Forschungsproblem ebenfalls identifizieren und beschreiben.

Schritt 2 - Zieldefinition: Auf Basis der identifizierten Probleme können nun die konkreten Ziele des Forschungsprojekts erarbeitet werden. Grundsätzlich kann zwischen quantitativen Zielvorgaben (z.B. potentielle Lösung muss besser sein als eine existierende Lösung) oder qualitativen Zielvorgaben (z.B. potentielle Lösung muss ein Problem adressieren, für das bisher noch keine Lösung existiert) unterschieden werden. Grundlage der Zieldefinition sind die Ergebnisse der Problemidentifikation sowie ausreichend Kenntnis über bereits existierende Lösungen.

Im Kontext der Arbeit: Eine erste Definition der Forschungsziele wird auf Basis des zentralen Forschungsproblems in Abs. 1.3 gegeben. In Kap. 4 wird das Forschungsproblem im Rahmen einer Interviewstudie genauer untersucht. Dabei stehen Vorteile und Probleme konventioneller PMS im Fokus. Auf Basis dieser Ergebnisse werden in Kap. 5 die Anforderungen an ein mögliches Lösungsartefakt erarbeitet. Weiterhin stellen die Erkenntnisse der Interviewstudie eine breite Wissensbasis dar, die zur Konzeption eines Lösungsansatzes herangezogen werden.

Schritt 3 - Konzeption und Entwicklung: In dieser Prozessphase entsteht die eigentliche Lösung als konkretes Artefakt auf Basis der definierten Ziele. Hierzu muss Funktionalität und Architektur des Artefakts erarbeitet und umgesetzt werden. Hevner et al. [35] nennt Konstrukte, Modelle, Methoden oder konkrete Anwendungen als typische Beispiele häufig entwickelter Artefakte. Neben der Zieldefinition ist in dieser Phase vor allem umfangreiches Wissen über thematisch relevante Theorien, die zur Anwendung gebracht werden können eine wichtige Voraussetzung.

Im Kontext der Arbeit: Wie bereits erwähnt werden die Anforderungen an des Lösungskonzept zu Beginn von Kap. 4 definiert. Anschließend folgt die Ausarbeitung konzeptioneller und technischer Lösungsansätze in Kap. 5. Dabei werden die in Kap. 3 ausgearbeiteten theoretischen Grundlagen sowie der aktuelle Stand der Forschung berücksichtigt.

Schritt 4 - Demonstration: Die Demonstration ist eine Vorstufe zur Evaluation und soll die generelle Fähigkeit des Artefakts zur Problemlösung zeigen. Peffers et al. führt beispielhaft ein Experiment, eine Simulation, eine Fallstudie oder eine Machbarkeitsstudie als Möglichkeit der Demonstration nach DSRP an. Grundvoraussetzung für diese Prozessphase ist ein tiefes Verständnis über das entwickelte Artefakt und dessen Anwendung im Problemkontext.

Im Kontext der Arbeit: Die Demonstration erfolgte im Rahmen der prototypischen Umsetzung in Kap. 5. Hierfür wurden einzelne Bestandteile des Lösungskonzepts getrennt voneinander getestet und iterativ verbessert, bis sich die Eignung bzgl. der

definierten Konzeptanforderungen abzeichnete. Im Vorfeld der Evaluation wurde das Lösungsartefakt dann im Zusammenhang getestet, um die grundsätzliche Eignung zum Einsatz in der agilen Projektarbeit sicherzustellen.

Schritt 5 - Evaluation: Die Evaluation geht einen Schritt weiter als die Demonstration. In dieser Prozessphase soll ermittelt werden in welchem Maß das Artefakt die gestellten Ziele und Anforderungen erfüllt. Demnach enthält diese Phase einen direkten Vergleich des Artefakts (z.B. Messwerte oder Beobachtungen) mit den in Schritt zwei definierten Zielen und Anforderungen. Voraussetzung der Evaluation ist demnach ausreichende Kenntnis geeigneter Analyse und Messverfahren, die zur Ermittlung relevanter Größen eingesetzt werden. Abhängig von den Forschungszielen können beispielsweise qualitative Vergleiche des Artefakts mit bereits existierenden Lösungen oder quantitative Vergleiche relevanter Kennzahlen mit den Zieldefinitionen eingesetzt werden. Am Ende dieser Phase muss das Forschungsteam entscheiden, ob es mit den Evaluationsergebnissen zufrieden ist oder eine weitere Iteration zur Verbesserung des Artefakts notwendig ist. Wie in Abb. 2.1 dargestellt ist, kann gegebenenfalls ein Rücksprung zu Schritt drei erfolgen, um die bisherigen Erfahrungen in Form von weiteren Verbesserungen des Artefakts einfließen zu lassen. Anschließend müssen die Schritte vier und fünf erneut durchlaufen werden.

Im Kontext der Arbeit: Die Evaluation des Lösungsartefakts wird in Kap. 6 erläutert. Ein grundlegender Schritt ist die Aufstellung geeigneter Hypothesen auf Basis der in Abs. 5 beschriebenen Anforderungen an eine mögliche Lösung. Anschließend wird die konkrete Anwendung der in Abs. 2.2.4 vorgestellten Untersuchungsverfahren gezeigt. Zum Schluss des Kapitels erfolgt die Vorstellung der Ergebnisse sowie die Bewertung der Hypothesen.

Schritt 6 - Kommunikation: Der letzte Schritt umfasst die Veröffentlichung der Forschungsergebnisse im Rahmen von Konferenzen oder wissenschaftlichen Aufsätzen. Peffers et al. empfiehlt die Anlehnung jeder Publikation an die einzelnen Schritte des DSRP. Demnach sollten mindestens die Relevanz des Problems für Wissenschaft und Praxis, das Artefakt, die Anwendung des Artefakts, das Vorgehen bei der Entwicklung sowie die Evaluationsergebnisse Bestandteil der Veröffentlichung sein.

Im Kontext der Arbeit: Im Rahmen der Forschungsarbeit wurden bereits Zwischenergebnisse auf internationalen Fachkonferenzen veröffentlicht (vgl. 7.3). Die Zusammenfassung und Veröffentlichung der gesamten Forschungsarbeit wird mit dieser Dissertationsschrift erreicht.

2.2 Forschungsmethodik von Vorstudie und Konzeptevaluation

In den folgenden Ausführungen zur Forschungsmethodik wird in Anlehnung an Mayring [54] zwischen Untersuchungsplan und Untersuchungsverfahren differenziert. Der Untersuchungsplan umfasst den übergeordneten Ablauf und legt die Rahmenbedingungen jeder Untersuchung fest. Dabei wird unter anderem das Untersuchungsziel, die Auswahl der

Untersuchungsteilnehmer, die Art der eingesetzten Untersuchungsverfahren sowie der zeitliche Rahmen der Untersuchung festgelegt. Ferner wird das Vorgehen zur Beantwortung der eingehenden Hypothesen bzw. Forschungsfragen bestimmt (z.B. Vergleich unterschiedlicher Personen oder Gruppen). Im Gegensatz dazu legt das Untersuchungsverfahren das Vorgehen für Datenerhebung, Datenaufbereitung und Auswertung fest.

2.2.1 Untersuchungsplan der Vorstudie - Die Einzelfallanalyse

Ein Ziel der in Kap. 4 beschriebenen Vorstudie war es, eine breite Wissensbasis zum Einsatz gängiger PMS innerhalb agiler Softwareentwicklung aufzubauen. Hierfür sollten unter anderem die Arbeitsabläufe verschiedener Projektteams sowie die Rolle der PMS im Entwicklungsprozess untersucht werden. Ein weiteres Ziel war es, Vorteile, Nachteile und Verbesserungspotentiale der jeweiligen PMS aus Sicht der Anwender zu ermitteln. Anhand der gegebenen Fragestellung wird deutlich, dass ein explorativer Forschungsansatz zur Anwendung kommen muss. Im Allgemeinen hat ein explorativer Ansatz das Ziel, in weitestgehend unerforschten Bereichen neue Erkenntnisse zu generieren, um eine Wissensbasis zur Formulierung neuer Hypothesen zu schaffen. Im Rahmen der Vorstudie wurden sog. Einzelfallanalysen als Vertreter qualitativer, explorativer Untersuchungspläne eingesetzt. Die Besonderheit der Einzelfallanalyse ist, dass nur einige wenige Untersuchungsgegenstände betrachtet werden. Das ermöglicht eine entsprechend intensive Betrachtung der Eigenarten eines jeden Falls und erlaubt eine intensive Analyse der jeweiligen Besonderheiten. Mayring stellt unter Bezugnahme auf Fuchs [29] und Jüttemann et al. [44] die nachfolgenden fünf Punkte als zentrale Schritte der Einzelfallanalyse dar. Dabei werden Punkt drei und vier durch das jeweils ausgewählte Untersuchungsverfahren festgelegt.

1. **Fragestellung:** Der Zweck einer Einzelfallanalyse muss definiert werden. Auch unter Annahme sehr allgemeiner Forschungsinteressen ist dieser Schritt auszuführen.
2. **Falldefinition:** Dieser Schritt definiert die Voraussetzungen eines für die Fragestellung interessanten Falls. Die Validität der Forschungsergebnisse hängt demnach in hohem Maße von diesen Voraussetzungen ab. Als Orientierungshilfe nennt Mayring beispielhaft Grenzfälle, Idealfälle, besonders häufig auftretende Fälle oder theoretisch interessante Fälle.
3. **Methodenauswahl:** Auf Grundlage der festgelegten Fragestellung und der Falldefinition muss nun ein geeignetes Untersuchungsverfahren ausgewählt werden. Eine Beschreibung der eingesetzten Verfahren wird in Abs. 2.2.2 gegeben.
4. **Aufbereitung:** Die Aufbereitung der Daten ist ebenfalls Teil des Untersuchungsverfahrens. Dazu zählt bereits die Fixierung der Daten (z.B. Videoaufzeichnung, Protokollierung, Tonaufzeichnung) sowie die Nachbearbeitung (z.B. Strukturierung, Codierung, Einfügen von Kommentaren). Die Strukturierung der Daten muss angepasst an die jeweilige Fragestellung durchgeführt werden.

5. **Einordnung in den Zusammenhang:** Im Rahmen einer Untersuchung werden im Regelfall mehrere Fallanalysen durchgeführt. In diesem Schritt müssen die Ergebnisse von besonders gelagerten Fällen mit den Ergebnissen der übrigen Fallanalysen verglichen werden, um deren Gültigkeit abschätzen zu können.

Die eben aufgezählten Schritte der Einzelfallanalyse wurden im Rahmen der Vorstudie angewandt. Eine detaillierte Beschreibung der angewandten Einzelfallanalyse befindet sich in Kap. 4.

2.2.2 Untersuchungsverfahren der Vorstudie

Wie bereits eingangs erwähnt umfasst das Untersuchungsverfahren Methoden zur Erhebung, Aufbereitung und Auswertung von Daten. Im Folgenden wird die Auswahl der eingesetzten Methode kurz erläutert und eine allgemeine Beschreibung zur Anwendung gegeben.

Datenerhebung - Das Problemzentrierte Interview Im Rahmen einer Einzelfallanalyse sind laut Mayring [54] verbale Erhebungsmethoden zu bevorzugen, weil die subjektive Anschauung der untersuchten Personen im Vordergrund steht. Als mögliche Alternativen nennt Mayring das Problemzentrierte Interview und das Narrative Interview. Hauptunterscheidungsmerkmal ist der Grad der Offenheit der Untersuchung. Beim Narrativen Interview wird der Befragte zum freien Erzählen zu einem bestimmten Thema animiert, ohne standardisierte Fragen beantworten zu müssen. Laut Mayring werden durch freie Erzählungen übergreifende Handlungszusammenhänge sichtbar, die bei standardisierten Befragungen im Regelfall nicht offengelegt werden. Der Interviewer greift demnach nur aktiv in das Gespräch ein, wenn der Befragte stark von seiner ursprünglichen Erzählung abweicht bzw. den roten Faden verliert. Ein entscheidender Nachteil der hohen Freiheitsgrade ist, dass die Daten einzelner Interviews nur sehr schwer vergleichbar sind und die Datenaufbereitung sowie Auswertung sehr aufwändig ist. Im Gegensatz dazu basiert das Problemzentrierte Interview auf standardisierten Leitfragen, die dem jeweiligen Interview eine feste Grundstruktur geben. Die einzelnen Leitfragen können wiederum völlig frei beantwortet und um Zwischenfragen seitens des Interviewers ergänzt werden. Der Interviewleitfaden kann theoriegestützt oder auf Basis einer vorangegangenen Problemanalyse erstellt werden. Durch die vorgegebene Grundstruktur wird eine Auswertung stark erleichtert und ein Vergleich einzelner Interviews ermöglicht.

Innerhalb der Vorstudie wurde das Problemzentrierte Interview in Kombination mit einer Tonbandaufzeichnung zur Datenerhebung eingesetzt, weil ein standardisierter Leitfaden die Möglichkeit bietet konkrete Fragen mit starkem Bezug zum agilen Entwicklungsprozess zu stellen. Weiterhin war es notwendig, die konkreten Unterschiede des PMS Einsatzes zu erarbeiten, um auf dieser Basis ein Lösungskonzept (vgl. Kap. 5) zu erarbeiten. Nachfolgend sind die einzelnen Arbeitsschritte des Problemzentrierten Interviews [54] beschrieben:

1. **Problemanalyse:** Am Anfang des Problemzentrierten Interviews steht die Analyse und Formulierung des Problems. Diese Schritt basiert auf der Fragestellung, die bereits beim Erstellen des Untersuchungsplans erarbeitet wurde. Neben einer reinen Problembeschreibung sollten hierbei auch zentrale Interessen und Ziele des geplanten Interviews noch einmal formuliert werden.
2. **Leitfadenkonstruktion:** Auf Basis der Problemanalyse kann anschließend der Interview Leitfaden konstruiert werden. Die Leitfragen sollten alle in der Problemanalyse festgehaltenen Aspekte abdecken. Dabei ist bereits jetzt auf eine Reihenfolge der Fragen zu achten, die einen sinnvollen Gesprächsaufbau ermöglicht, um starke thematische Sprünge während des Interviews auf ein Minimalmaß zu reduzieren und den Gesprächsfluss aufrecht zu erhalten. Mayring schlägt drei Fragetypen zum Aufbau des Leitfadens vor: Zu Beginn sollten sog. Sondierungsfragen eingeplant werden. Dabei handelt es sich um allgemein gehaltene Einstiegsfragen zum Thema. Sie haben das Ziel die subjektive Bedeutung des Themas für den jeweiligen Interviewpartner zu eruieren und dessen Hintergrund bzw. Erfahrungsschatz auf dem jeweiligen Interessengebiet zu ermitteln. Anschließend sollten Leitfragen zu den wesentlichen Aspekten der Studie gestellt werden. Diese können stichpunktartig oder als fertige Formulierungsvorschläge im Leitfaden festgehalten werden. Als letzten Fragetyp führt Mayring die sog. Ad-hoc-Fragen an. Dabei handelt es sich um Detailfragen zu interessanten Aspekten, die unerwartet im Interview auftauchen und für die Fragestellung bedeutsam sind.
3. **Leitfadenerprobung und Interviewerschulung:** Die Leitfadenerprobung wird im Rahmen eines Probeinterviews durchgeführt. Dabei wird im Vorfeld der eigentlichen Untersuchung entweder ein vollständiges Interview oder ein ausgewählter Teil anhand eines Probeinterviews getestet, um den Ablauf zu üben und Schwachstellen oder fehlende Aspekte im Leitfaden aufzudecken. Der Interviewpartner sollte den im Untersuchungsplan festgelegten Kriterien zur Auswahl geeigneter Interviewpartner weitestgehend entsprechen. Bei Bedarf können mehrere Iterationen der Leitfadenerprobung durchgeführt werden, bis man mit dem Ergebnis zufrieden ist.
4. **Aufzeichnung:** Es ist wichtig, dass die Inhalte des Interviews möglichst detailliert festgehalten werden. Eine bewährte Methode ist die Tonbandaufzeichnung, deren Anwendung jedoch das Einverständnis der Befragten Personen voraussetzt. Als Alternative kann während des Gesprächs auch ein Protokoll verfasst werden. Diese Variante ist nur zu empfehlen, wenn eine Tonbandaufnahme nicht möglich ist, da sie hohe Anforderungen an den Interviewer stellt und störend auf den Gesprächsablauf wirken kann.

Datenaufbereitung - Wörtliches Transkript Als Auswertungsgrundlage ist eine textuelle Form der erhobenen Daten unabdingbar. Im Gegensatz zur reinen Tonbandaufnahme können beispielsweise Kommentare ergänzt und Textstellen markiert oder verglichen werden. Die Transformation von Sprache in Textform nennt man Transkription. Mayring unterscheidet hierbei zwischen der wörtlichen und der kommentierten Transkription. Für beide Arten gibt es verschiedene Regelwerke, die beispielsweise den Umgang mit Dialek-

ten, Sprechpausen, Umgangssprache und Wortbetonungen vorgeben. Die Unterscheidung einer kommentierten Transkription von einer wörtlichen Transkription kann anhand der verwendeten Transkriptionsregeln festgemacht werden. Kommentierte Transkriptionen berücksichtigen sprachliche Besonderheiten wie beispielsweise Betonungen, Sprechpausen, Lachen, Sprechgeschwindigkeit, etc. durch festgelegte Notationen im Transkript. Im Allgemeinen ist diese Transkriptionsart nur zu empfehlen, wenn die eben aufgeführten sprachlichen Besonderheiten für die zugrundeliegende Fragestellung von Bedeutung sind, weil der Transkriptionsvorgang sehr aufwändig ist und die Lesbarkeit durch die unterschiedlichen Notationselemente stark reduziert wird. Im Gegensatz dazu wird bei der wörtlichen Transkription zumeist nur eine Übertragung in normales Schriftdeutsch vorgenommen, um bestmögliche Lesbarkeit zu erreichen. Dabei werden Satzbaufehler, Dialekte und Umgangssprache weitestgehend bereinigt und der Fokus auf die inhaltlich-thematische Ebene gelegt.

Unter Berücksichtigung dieser Überlegungen wurde im Rahmen der Vorstudie eine reduzierte Form der kommentierten Transkription verwendet. Diese Variante enthält die Möglichkeit, einzelne Lautäußerungen oder Sprechpausen zu kommentieren. Hierdurch können bedeutsame Elemente wie Betonungen, Sarkasmus, Zustimmung oder Ablehnung festgehalten werden - der Transkriptionsaufwand bleibt dennoch in einem vertretbaren Rahmen. Als Grundlage diente das Transkriptionsregelwerk nach Kuckartz et al. [49], das im Wesentlichen folgende Regeln enthält:

1. Es wird wörtlich transkribiert, also nicht lautsprachlich oder zusammenfassend. Vorhandene Dialekte werden nicht transkribiert.
2. Die Sprache und Interpunktion wird leicht geglättet, d.h. an das Schriftdeutsch angenähert.
3. Alle Angaben, die einen Rückschluss auf eine befragte Person erlauben, werden anonymisiert.
4. Deutliche, längere Pausen werden durch Auslassungspunkte (...) markiert.
5. Besonders betonte Begriffe werden durch Unterstreichungen gekennzeichnet.
6. Zustimmende bzw. bestätigende Lautäußerungen der Interviewer (Mhm, Aha etc.) werden nicht mit transkribiert, sofern sie den Redefluss der befragten Person nicht unterbrechen.
7. Einwürfe der jeweils anderen Person werden in Klammern gesetzt.
8. Lautäußerungen der befragten Person, die eine Aussage unterstützen oder verdeutlichen (etwa lachen oder seufzen) werden in Klammern notiert.
9. Die interviewende Person wird durch ein „I“, die befragte Person durch ein „B“, gefolgt von ihrer Kennnummer gekennzeichnet (etwa „B4:“).
10. Jeder Sprecherwechsel wird durch zweimaliges Drücken der Enter-Taste, also einer Leerzeile zwischen den Sprechern, deutlich gemacht, um die Lesbarkeit zu erhöhen.

Die Transkription wurde mit Hilfe der Software "F4-Transkript"[28] durchgeführt. Sie vereinfacht das Abspielen der Tonbandaufnahmen und das Schreiben der Transkripte

erheblich. Der Ausschnitt eines im Rahmen dieser Arbeit entstandenen Transkripts findet sich zur Veranschaulichung im Anhang (vgl. Abs. A.1).

Datenauswertung - Qualitative Inhaltsanalyse Die Datenauswertung wurde mit Hilfe der sog. Qualitativen Inhaltsanalyse in Anlehnung an Mayring [54] durchgeführt. Dabei handelt es sich um eine systematische Methode zur Auswertung von Texten oder Protokollen. Ziel der Auswertung ist die Reduktion des Datenmaterials auf für die Fragestellung relevante Inhalte. Hierfür greift die Qualitative Inhaltsanalyse auf die Technik der induktiven Kategorisierung zurück, die eine systematische Ableitung von Bewertungskriterien (nachfolgend Kategorien genannt) aus dem bestehenden Datenmaterial vorsieht. Es müssen vor der eigentlichen Auswertung Selektionskriterien für die Kategorienbildung festgelegt werden, welche wiederum von der Fragestellung bzw. den Untersuchungszielen abhängig sind. Anschließend wird das zu untersuchende Datenmaterial zeilenweise nach Textstellen durchsucht, die den Selektionskriterien entsprechen. Sobald eine Stelle gefunden ist, wird entweder eine neue Kategorie angelegt oder sie wird einer passenden, bereits existierenden Kategorie zugeordnet.

Ein kurzer Vorgriff auf die Vorstudie in Kap. 4 veranschaulicht die Anwendung dieser Methode: Ein Ziel der Vorstudie war es, konkrete Verbesserungspotenziale von PMS innerhalb agiler Projekte zu finden. Als Selektionskriterium muss deshalb genau definiert werden, was im Rahmen der Untersuchung unter einem Verbesserungspotential verstanden wird. Jede konkrete Ausprägung eines Verbesserungspotentials wird dann als Kategorie angelegt oder einer bestehenden Kategorie zugeteilt. Sobald ein gewisser Teil der Analyse durchgeführt wurde und keine oder nur wenige neue Kategorien zu erwarten sind, muss eine Überarbeitung des Kategoriensystems durchgeführt werden. Ziel der Überarbeitung ist ein System, das keine inhaltlichen Überschneidungen der Kategorien aufweist und zum Ziel der Untersuchung passt. Bei größeren Veränderungen müssen die bisherigen Ergebnisse noch einmal angepasst werden. Das Ergebnis der Qualitativen Inhaltsanalyse ist dann ein System aus weitestgehend disjunkten Kategorien, denen jeweils mindestens eine Textstelle als konkrete Ausprägung zugewiesen wurde. Das System kann anschließend gemäß dem Untersuchungsziel interpretiert und auf Basis der zugewiesenen Ausprägungen auch quantifiziert werden.

Die qualitative Inhaltsanalyse wurde mit Hilfe der *Computer-Assisted Qualitative Data Analysis* (CAQDAS)-Software "F4Analyse" durchgeführt. Die Software ermöglicht eine übersichtliche Darstellung aller gefundenen Kategorien und der jeweils zugeordneten Textstellen. Auch die anschließende Interpretation der Ergebnisse sowie die angesprochene Quantifizierung wird erheblich vereinfacht.

2.2.3 Untersuchungsplan der Konzeptevaluation - Das qualitative Experiment

Allgemein formuliert bietet sich zur Evaluation eines Lösungsartefakts nach DSRP ein Vergleich der Ausgangssituation (Ist-Zustand) mit der Situation nach einem experimentellen Eingriff (Zielzustand) an. Konkretisiert auf die vorliegende Arbeit

soll demnach der Einfluss gängiger PMS auf einzelne Aspekte agiler Projektarbeit mit dem Einfluss des in Kap. 5 ausgearbeiteten Lösungskonzepts verglichen werden. Aus diesem Grund wurde das qualitative Experiment als Untersuchungsplan ausgewählt. Zur Überprüfung der in Kap. 6 formulierten Hypothesen ist eine Kombination unterschiedlicher Untersuchungsverfahren notwendig. Die qualitative Beobachtung (vgl. 2.2.4.1) stellt dabei das zentrale Verfahren dar und erlaubt vergleichende Aussagen bzgl. Kommunikation, Transparenz und Motivation im Projektteam aus Sicht der Beobachter. Zusätzlich zur qualitativen Beobachtung wurde eine schriftliche Befragung der Untersuchungsteilnehmer mit Hilfe eines standardisierten Fragebogens [79] durchgeführt. Ziel dieses Erhebungsverfahrens war ein direkter Vergleich hinsichtlich gezielt ausgewählter Eigenschaften, die laut Katsma et al. [46] ausschlaggebend für die Qualität von PMS in agilen Projekten sind (vgl. 6.2.1).

Ein weiterer, in den Hypothesen enthaltener Aspekt, ist ein Vergleich der Gebrauchstauglichkeit gem. DIN EN ISO 9241 [32]. Zur Messung der Gebrauchstauglichkeit ist die SUS [12] ein anerkanntes sowie leicht anwendbares Instrument und wurde deshalb ergänzend eingesetzt (vgl. 2.2.4.3). Die nachfolgenden Absätze enthalten eine allgemeine Beschreibung des Qualitativen Experiments sowie der eingesetzten Untersuchungsverfahren. Die Konkrete Anwendung wird in Kap. 6 zur Evaluationsstudie detailliert beschrieben.

Mayring [54] nennt das Experiment als einzige Möglichkeit Kausalzusammenhänge aufzudecken. Als Kausalzusammenhang kann eine Beziehung zwischen Ursache und Wirkung verstanden werden, d.h. eine gegebene Vorbedingung (z.B. Eigenschaft eines Produkts) hat nachweislich Auswirkungen auf eine nachgelagerte Variable (z.B. Art und Weise der Benutzung eines Produkts). Um solch einen Zusammenhang aufzudecken muss zwangsläufig ein Vergleich des Untersuchungsgegenstandes vor und nach dem sog. experimentellen Eingriff stattfinden. Unter Bezugnahme auf Leining et al. [47] nennt Mayring sechs verschiedene Techniken, um den experimentellen Eingriff durchzuführen. Im Rahmen dieser Arbeit wurde die Technik der Substitution verwendet. Dabei werden bestimmte Teile des Gegenstands durch neue Teile ersetzt und beobachtet, ob entsprechende Auswirkungen erkennbar sind. Es ist besonders darauf zu achten, dass alle anderen Aspekte des Untersuchungsgegenstands sowie die Rahmenbedingungen des Experiments vor und nach der Substitution vergleichbar sind, um sicherzugehen, dass die Auswirkungen möglichst ausschließlich auf den experimentellen Eingriff zurückzuführen sind. Im Rahmen der Evaluationsstudie (vgl. Kap. 6) wurde ein Vergleich zweier studentischer Projektteams durchgeführt. Dabei arbeitete das erste Team mit der PMS Jira [40] und repräsentierte damit die Ausgangssituation vor dem experimentellen Eingriff. Das zweite Team arbeitete mit der prototypischen Umsetzung des Lösungskonzepts bei sonst gleichen Rahmenbedingungen. In diesem Zusammenhang wird häufig von einer Kontrollgruppe (erstes Team) und einer Experimentalgruppe (zweites Team) gesprochen. Während des Experiments werden nach identischen Untersuchungsverfahren relevante Daten der Kontroll- und Experimentalgruppe erfasst, die anschließend verglichen werden.

Zusammenfassend lässt sich das qualitative Experiment in folgende Teilschritte aufteilen:

1. **Erste Versuchsdurchführung und Beschreibung der Ergebnisse:** Dieser Schritt besteht aus der Durchführung der geplanten Untersuchung mit der Kontrollgruppe. Dabei werden die vorher festgelegten Untersuchungsverfahren -bestehend aus Datenerhebung, Datenaufbereitung und Datenauswertung- vollständig angewandt.
2. **Experimenteller Eingriff:** Wie bereits beschrieben muss in diesem Schritt die Anpassung des Untersuchungsgegenstands erfolgen. Dabei muss darauf geachtet werden, dass möglichst nur die gewünschte Variable verändert wird und alle anderen Bedingungen identisch bleiben.
3. **Zweite Versuchsdurchführung und Beschreibung der Ergebnisse:** Jetzt kann die zweite Untersuchung mit angepasstem Untersuchungsgegenstand mit Hilfe der Experimentalgruppe erfolgen. Die Untersuchungsverfahren müssen identisch zur ersten Versuchsdurchführung angewendet werden.
4. **Schlussfolgerungen:** Die zu überprüfenden Hypothesen können auf Basis der Ergebnisse beider Versuchsdurchführungen überprüft werden.

2.2.4 Untersuchungsverfahren der Konzeptevaluation

Wie bereits beschrieben wurden im Rahmen der Konzeptevaluation die folgenden drei Untersuchungsverfahren eingesetzt:

- Beobachtung mit schriftlicher Protokollierung
- Schriftliche Befragung mit standardisiertem Fragebogen
- Die System Usability Scale (SUS)

In den nachfolgenden Absätzen werden die drei Untersuchungsverfahren in der angegebenen Reihenfolge behandelt. Dabei steht die Art der Datenerhebung und der Datenauswertung im Fokus.

2.2.4.1 Beobachtung mit schriftlicher Protokollierung

Ein typischer Effekt, der im Rahmen von wissenschaftlichen Befragungen nachgewiesen wurde ist, dass Menschen in realen Situationen häufig anders handeln als bei der Beantwortung einer Forschungsfrage angegeben. So kann die Art und Weise wie ein Interviewer Fragen stellt oder z.B. das Geschlecht des Interviewers erheblichen Einfluss auf die Antworten haben [34]. Diesem Effekt kann mittels einer Beobachtungsstudie entgegengewirkt werden, da das tatsächliche und somit validere Verhalten in einer bestimmten Situation beobachtet und protokolliert werden kann [13].

Generell bietet das Feld der Beobachtungsstudien aus methodischer Sicht ein breites Spektrum an Variationsmöglichkeiten bzgl. Beobachtungsart, Beobachtungssituation und Protokollierung, weshalb an dieser Stelle eine kurze Einordnung der durchgeführten Studie folgt:

Beobachtungsart: Abb. 2.2 zeigt drei Merkmale, die bei der Studienplanung bzgl. der Beobachtungsart beachtet werden müssen. Zunächst wird zwischen internen und externen Beobachtern unterschieden. Intern bedeutet, dass der Beobachter Teil des Forschungsteams ist und somit über breites Hintergrundwissen zum Forschungsinteresse verfügt. Aus forschungsökonomischer Sicht hat das den Vorteil, dass keine externen Beobachter hinzugezogen werden müssen und irrelevante Beobachtungen bereits während der Protokollierung außen vor gelassen werden können. Der Nachteil einer internen Beobachtung ist, dass es durch Vorannahmen des Beobachters über den Forschungsgegenstand zu hypothesengeleiteter Wahrnehmung und somit zur Verfälschung der Ergebnisse kommen kann. In der vorliegenden Arbeit wurde aufgrund beschränkter Ressourcen die interne Beobachtung gewählt. Der Beobachter selbst war nicht Teil des Projektteams und verhielt sich möglichst passiv, um eine Beeinflussung des Arbeitsprozesses gering zu halten. Mayring [54] bezeichnet diese Art der Beobachtung als nicht teilnehmend. Das dritte Merkmal in Abb. 2.2 differenziert zwischen Selbstbeobachtung bzw. Fremdbeobachtung und gibt an, ob man sich selbst oder andere Personen beobachtet. Die Selbstbeobachtung wird nur in Ausnahmefällen angewandt wenn beispielsweise mentale Prozesse im Fokus stehen und nicht mit ausreichend extern erfassbaren Reaktionen gerechnet werden kann. Die vorliegende Studie wurde demnach mit Hilfe der Fremdbeobachtung durchgeführt.

Beobachtungssituation: Wie in Abb. 2.2 dargestellt ist, lässt sich die Beobachtungssituation ebenfalls anhand dreier Merkmale klassifizieren. Zunächst wird unterschieden, ob die Beobachtung verdeckt (für den Beobachteten nicht sichtbar) oder offen (für den Beobachteten sichtbar) durchgeführt wird. Hierbei gilt es abzuwägen, inwieweit die offene Beobachtung das Forschungsergebnis beeinflusst bzw. vertretbar ist. In unserem Fall wurde eine Zwischenlösung eingesetzt: Wir haben uns aufgrund der einfacheren Realisierbarkeit für eine offene Beobachtung entschieden, aber die genauen Interessenschwerpunkte nicht bekannt gegeben. Ein bewusstes oder unbewusstes Verhalten zur Beeinflussung des Forschungsergebnisses durch die Mitglieder des Projektteams in eine bestimmte Richtung konnte somit weitestgehend vermieden werden. Weiterhin wird unterschieden, ob die Beobachtung als Feldversuch oder Laborversuch durchgeführt wird. Von einem Feldversuch spricht man, wenn das Verhalten dort beobachtet wird, wo es natürlicherweise auftritt, um eine möglichst hohe Authentizität des beobachteten Verhaltens zu erreichen. Im vorliegenden Fall handelt es sich um einen Laborversuch, da das Umfeld bzw. die Räumlichkeiten den Bedürfnissen der Studie angepasst wurden. Dazu zählt z.B. die Bereitstellung der jeweils eingesetzten PMS. Da es sich bei Virtual Task-Board um einen eigens entwickelten Prototypen handelt, schließt sich ein Feldversuch aus. Das dritte Merkmal zur Klassifizierung der Beobachtungssituation unterscheidet eine Studiendurchführung mit oder ohne Stimulus. Rechnet man mit eher geringem Auftreten des interessanten Verhaltens besteht die Möglichkeit dieses durch gezielte Einflussnahme von außen zu provozieren. In diesem Fall spricht man von einer Beobachtung mit Stimulus. In der vorliegenden Studie wurde auf diese Art der Einflussnahme weitestgehend verzichtet. Lediglich gelegentliche Hinweise zum Entwicklungsprozess wurden eingestreut, um die Situation in beiden Projektteams

vergleichbar zu gestalten.

Abb. 2.2 fasst die Klassifizierung der Beobachtungsstudie noch einmal zusammen. Die grau hinterlegten Merkmalsausprägungen entsprechen der durchgeführten Studie.

Protokollierung: Bei der Protokollierung unterscheidet man zwischen strukturierter und unstrukturierter Vorgehensweise. Die strukturierte Protokollierung setzt breites Vorwissen über den Untersuchungsgegenstand voraus und basiert auf vorbereiteten Protokollschemata, die vom Beobachter vervollständigt werden. Eine vollkommen unstrukturierte Protokollierung lässt dem Beobachter hier alle Freiheiten. Bei der vorliegenden Studie sprechen wir von einer strukturierten Protokollierung, da ein bestehendes Bewertungsmodell nach Katsma et al. [46] für PMS eingesetzt wurde, das bereits verschiedene Bewertungskriterien vorgibt. Details zum Bewertungsmodell sind nicht Teil dieses Kapitels, sondern werden in Abs. 6.2.2 zur Evaluationsstudie erläutert. Das zweite Merkmal zur Klassifizierung der Protokollierung gibt an, ob die Beobachtungen direkt, also zum Zeitpunkt des Geschehens, notiert und ggf. interpretiert werden oder ob dieser Schritt im Nachgang erfolgt (z.B. Auswertung aufgezeichneter Daten). Die Protokollierung erfolgte bei der vorliegenden Studie direkt, da keine Beeinflussung des Studienablaufs zu erwarten war. Ein ähnlicher Punkt ist die Unterscheidung zwischen unvermittelter und vermittelter Protokollierung. Bei unvermittelter Protokollierung, wie im vorliegenden Fall, wird die Studiensituation direkt zum Zeitpunkt des Geschehens beobachtet und protokolliert. Von vermittelter Protokollierung spricht man, wenn die Beobachtung auf Basis von Video- oder Audioaufzeichnungen durchgeführt wird. Im Regelfall ist die Validität von unvermittelter Protokollierung höher, weil der Beobachter einen authentischeren Eindruck der Situation erhält. Das letzte Klassifikationsmerkmal der Protokollierung differenziert zwischen manueller und automatisierter Protokollierung. Die vorliegende Studie wurde ausschließlich manuell protokolliert, da die relevanten Handlungen durch einen menschlichen Beobachter erkannt und interpretiert werden mussten. Eine Automatisierung wäre nicht möglich gewesen.

Beobachtungs- art	intern	extern
	teilnehmend	nicht teilnehmend
	Selbstbeobachtung	Fremdbeobachtung
Beobachtungs- situation	offen	verdeckt
	Feld	Labor
	ohne Stimulus	mit Stimulus
Protokollierung	strukturiert	nicht strukturiert
	direkt	indirekt
	unvermittelt	vermittelt
	manuell	automatisiert

Abbildung 2.2: Klassifizierung der Beobachtungsverfahren nach Brosius [13] S.199.

Zur Auswertung der Beobachtungsprotokolle wurde das bereits beschriebene Verfahren der Qualitativen Inhaltsanalyse (vgl. Abs. 2.2.2) eingesetzt. Im Vergleich zur Vorstudie liegen jedoch keine kommentierten Transkripte vor, sondern strukturierte Beobachtungsprotokolle. Aus diesem Grund ergeben sich einige nennenswerte Unterschiede: Das Bewertungsmodell nach Katsma et al. gibt eine feste Struktur von Kriterien vor, welche bereits Bestandteil der Protokollvorlage waren. Die protokollierten Beobachtungen sind also mindestens einem Kriterium zugeordnet und geben bereits ein erstes Kategoriensystem vor. Die eigentliche Inhaltsanalyse findet also innerhalb dieser Kategorien statt und führt zur Bildung von Unterkategorien, denen wiederum verschiedene Textstellen zugeordnet wurden. Auch hier soll ein kurzer Vorgriff auf die Evaluationsstudie (vgl. Kap. 6) zur Veranschaulichung herangezogen werden: Das Bewertungsmodell nach Katsma et al. [46] beinhaltet beispielsweise das Bewertungskriterium "Motivation". Verschiedene Beobachtungen wurden diesem Kriterium bereits vom Protokollanten zugeordnet. Im Rahmen der Inhaltsanalyse werden diese Beobachtungen dann detailliert betrachtet und Unterkategorien gebildet. So könnte beispielsweise eine Unterkategorie lauten: "Geringe

Motivation äußert sich durch geringe Beteiligung an der Diskussion" oder "Hohe Motivation - alle Teammitglieder arbeiten aktiv mit". Diesen Unterkategorien werden dann alle entsprechenden Textstellen zugeordnet. Nachdem ein Großteil der Protokolle ausgewertet wurden, müssen die Unterkategorien überarbeitet werden, um inhaltlich möglichst disjunkte Unterkategorien zu erhalten. Das Vorgehen unterscheidet sich hierbei nicht von der in Abs. 2.2.2 beschriebenen Verfahrensweise. Anschließend können die Kategoriensysteme der Kontrollgruppe und der Experimentalgruppe verglichen und entsprechend der Fragestellung interpretiert werden.

2.2.4.2 Schriftliche Befragung

Die qualitative Beobachtung hat den Nachteil, dass die subjektive Betrachtung des Beobachters die Ergebnisse beeinflussen kann. Dieses Problem verstärkt sich im Regelfall wenn der Beobachter Teil des Forschungsteams ist, da die Beobachtung unbewusst einseitig (z.B. zum Vorteil des eigenen Lösungsansatzes) durchgeführt wird. Um diesem Problem entgegenzuwirken, wurde eine von der Beobachtung unabhängige Befragung der Versuchsteilnehmer durchgeführt. Diese Perspektive erlaubt es, Abweichungen bzw. Übereinstimmungen zu den Ergebnissen der Beobachtungsstudie festzustellen. Um eine Vergleichbarkeit beider Erhebungen durchführen zu können, wurde auch die schriftliche Befragung auf Basis des Bewertungsmodells für PMS nach Katsma et al. [46] konzeptioniert.

Im Rahmen dieser Arbeit wurden ein klassisches Rangordnungsverfahren sowie die sog. Q-Sort-Technik (QST) [80] zur Durchführung einer schriftlichen Befragung herangezogen. Bei beiden Varianten werden vordefinierte Aussagen bzgl. eines Untersuchungsgegenstands (z.B. "Die Nutzung des Produkts macht Spaß") einem Wert auf einer definierten Ratingskala zugeordnet. Die in Kap. 6 beschriebene Evaluationsstudie wurde mit Hilfe einer fünfstufigen Ratingskala nach Likert [52] von "trifft überhaupt nicht zu" bis "trifft voll und ganz zu" durchgeführt (vgl. Grafik 2.3). Das Hauptunterscheidungsmerkmal beider Varianten ist, dass es sich bei der klassischen Erhebungstechnik um ein Rangordnungsverfahren mit freier Zuordnung der Aussagen zu einem beliebigen Wert auf der Ratingskala handelt und die QST eine Zuordnung verlangt, die sich an der Normalverteilung orientiert.

Der genaue Ablauf sowie die Auswirkungen auf das zu erwartenden Untersuchungsergebnis und dessen Interpretation wird im Folgenden genauer erläutert:

Die erzwungene Normalverteilung im Kontext der QST bedeutet, dass jedem Wert auf der Ratingskala eine fest vorgegebene Anzahl an Aussagen zugeordnet werden muss. Diese Anzahl orientiert sich dabei an der Normalverteilung, weshalb der neutralen Position in der Mitte die meisten Aussagen zugeordnet werden müssen. Zu den Rändern der Skala hin nimmt die erlaubte Anzahl an Aussagen dann entsprechend ab (vgl. 2.3). Im abgebildeten Beispiel dürfen den Positionen -2 und 2 jeweils drei Aussagen und den Positionen -1 und 1 jeweils vier Aussagen zugeordnet werden. Häufig wird die QST mit Hilfe von Karteikarten durchgeführt, auf denen jeweils eine zu bewertende Aussage steht. Die Karten werden direkt unter einer abgebildeten Skala angeordnet. Der Befragte

wird entsprechend informiert, wie viele Aussagen einem Skalenwert zugewiesen werden müssen. Er hat dann die Möglichkeit einzelne Karten so oft umzusortieren, bis das Ergebnis seinem Meinungsbild bestmöglich entspricht. Grafik 2.3 zeigt einen fertigen Q-Sort aus insgesamt 20 Aussagen. Die roten Rechtecke stellen Karten mit jeweils einer zu bewertenden Aussage dar.

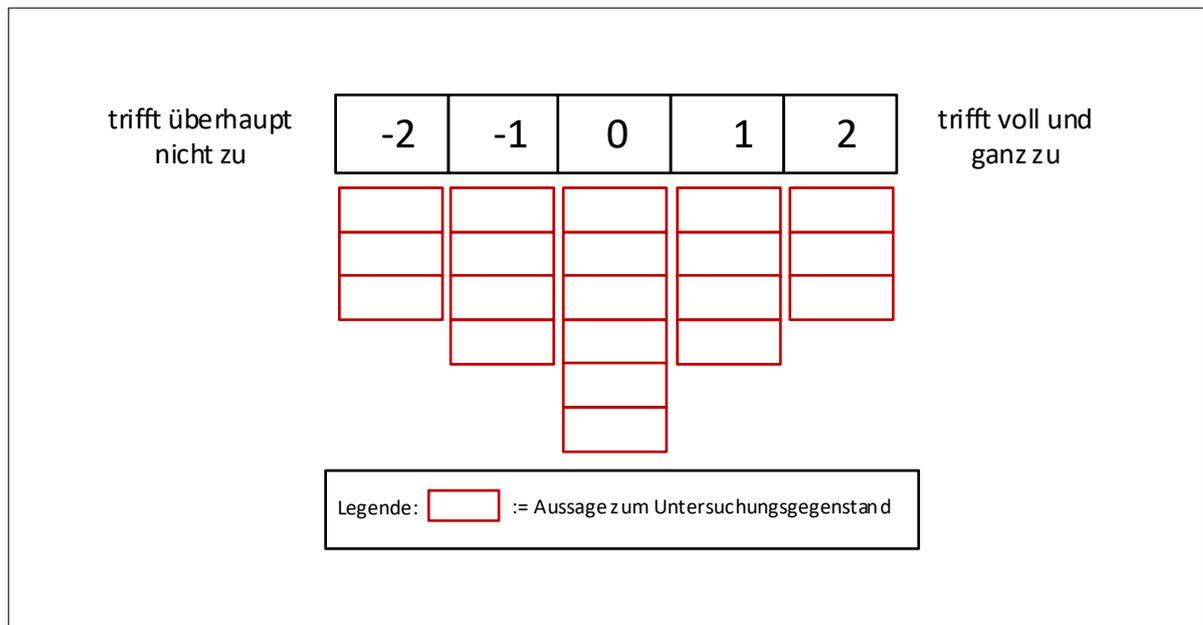


Abbildung 2.3: Beispiel einer Ratingskala nach QST

Diese Art der Datenerhebung bringt im Vergleich klassischen Variante einige Besonderheiten mit sich:

- **Relationale Bewertung der Aussagen:** Im Gegensatz zur klassischen Datenerhebung mit Fragebögen werden die Aussagen zum Forschungsgegenstand bei der QST nicht unabhängig voneinander betrachtet, sondern miteinander in Beziehung gesetzt. Das hat den Vorteil, dass sich befragte Personen länger mit den einzelnen Aussagen und ihrem Meinungsbild beschäftigen müssen und somit validere Ergebnisse zu erwarten sind. Laut Brown [14] festigt sich die Meinung vieler befragter Personen sogar erst während der Durchführung der QST.
- **Gesamtbewertung des Qualitativen Experiments:** Wendet man die klassische Variante an, kann im Rahmen des Qualitativen Experiments eine Gesamtaussage bzgl. des Untersuchungsgegenstands vor und nach dem experimentellen Eingriff getroffen werden. So würde beispielsweise durch Aufsummierung der Einzelergebnisse ein Gesamtbewertung zustande kommen, die je nach Forschungsfrage eine übergreifende Aussage (wie z.B. "Der Untersuchungsgegenstand eignet sich nach dem experimentellen Eingriff besser/schlechter zur Unterstützung agiler Projektarbeit") treffen lassen.

Im Fall der Evaluationsstudie (vgl. Kap. 6) wird ein Bewertungsmodell (vgl. Abs. 6.2.1) verwendet, dessen Bewertungskriterien gleich gewichtet sind. Durch die erzwungene Normalverteilung der QST ergibt die Aufsummierung immer Null, wodurch eine Gesamtaussage dieser Art nicht mehr möglich ist. Interessant ist hierbei jedoch die Bewertung einzelner Bewertungskriterien. Es lässt sich ablesen, welche Eigenschaften des Untersuchungsgegenstands sich verändert haben. Diese Eigenschaften der klassischen Variante bzw. der QST müssen bei der Ergebnisinterpretation entsprechend berücksichtigt werden. Weitere Details zu diesem Thema werden in Abs. 6.3.2 zusammen mit den Studienergebnissen erläutert.

- **Flexibilität und Korrekturmöglichkeiten während der Datenerhebung:** Vor allem durch die Verwendung von Karteikarten entsteht große Flexibilität bei der Einordnung der einzelnen Aussagen in die vorgegebene Verteilung. Die befragte Person kann ihr Urteil beliebig oft überdenken und abändern, bis ein zufriedenstellendes Ergebnis erreicht ist.
- **Vorbeugung unerwünschter Antworttendenzen:** Menschen weisen innerhalb von Meinungsumfragen häufig verstärkte Tendenzen in eine Richtung - also z.B. durchgängig positive Antworten - auf. Die erzwungene, normalverteilte Bewertung der Aussagen lässt diese Antworttendenzen nicht zu.

Ein entscheidender Faktor für die Validität der Studienergebnisse beider Varianten ist die Festlegung und Formulierung der zu bewertenden Aussagen. Als Ausgangslage dienen geeignete Modelle aus unterschiedlichen Variablen, mit deren Hilfe der Untersuchungsgegenstand bzw. relevante Eigenschaften des Untersuchungsgegenstands beschrieben werden können. Generell unterscheidet man hierbei zwischen latenten und manifesten Variablen, die nach Weiber et al. [84] folgendermaßen zu verstehen sind:

Manifeste Variablen sind direkt beobachtbar und ihre konkreten Ausprägungen können mit geeigneten Messinstrumenten direkt erfasst werden. Im Gegensatz dazu entziehen sich latente Variablen der direkten Beobachtbarkeit und setzen ein geeignetes Messmodell zur Erfassung ihrer Ausprägungen voraus.

Im Fall der Evaluationsstudie (vgl. Abs. 6) dient das Bewertungsmodell für PMS nach Katsma et al. [46] als Konstrukt zur Beschreibung des Untersuchungsgegenstandes und damit als Basis zur Festlegung der zu bewertenden Aussagen. Es besteht ausschließlich aus latenten Variablen (z.B. Kommunikation, Motivation, etc.), weshalb geeignete Messmodelle konstruiert werden müssen. Dabei ist zu beachten, dass nach Diamantopoulos et al. [22] zwischen formativen und reflektiven Messmodellen (vgl. Abb. 2.4) unterschieden werden muss. Aus diesem Grund erfolgt an dieser Stelle eine kurze Gegenüberstellung beider Ansätze.

Vergleich reflektiver und formativer Messmodelle Allgemein betrachtet bildet ein Messmodell die Beziehung zwischen latenten Variablen (dargestellt in Abb. 2.4 durch η) und messbarer Größen - den sog. Indikatoren (dargestellt in Abb. 2.4 durch x) ab. Im Fall der schriftlichen Befragung entsprechen die Indikatoren den einzelnen Aussagen,

die vom Untersuchungsteilnehmer bewertet werden. Entsprechend wird im Rahmen der Evaluationsstudie jeder Variablen aus dem gegebenen Bewertungsmodell nach Katsma et al. [46] eine bestimmte Anzahl bewertbarer Aussagen zugewiesen, deren Bewertungsergebnisse anschließend zu einem Gesamtmesswert pro latenter Variable verrechnet werden. Das Vorgehen hierbei ist von der Art des gewählten Messmodells (formativ oder reflektiv) abhängig. Der Unterschied beider Varianten liegt in erster Linie in der Richtung des Kausalzusammenhangs zwischen latenten Variablen und Indikatoren.

Reflektive Modelle sehen die Indikatoren als Folge der latenten Variablen. [2]. Wie die Pfeilrichtung in Abb. 2.4 verdeutlicht, reflektieren die Indikatoren die latente Variable und machen sie somit messbar. Da die Indikatoren von derselben Variable abhängen spiegelt sich eine Veränderung der latenten Variable bei einem gut gewählten Modell in allen Indikatoren wieder. Eine hohe Korrelation zwischen den einzelnen Indikatoren (dargestellt in Abb. 2.4 durch r) ist demnach ein Indikator für eine hohe Modellgüte und Grundlage gängiger Verfahren zur Validitätsprüfung (z.B. Cronbachs Alpha [19]) reflektiver Modelle. Eine perfekte Messung der latenten Variable durch die eingesetzten Indikatoren ist praktisch unmöglich, weshalb jeder Indikator um einen Messfehler (dargestellt in Abb. 2.4 durch E) ergänzt wird. Die entsprechende Gleichung des reflektiven Messmodells ist ebenfalls in Abb. 2.4 dargestellt.

Im Vergleich dazu geht ein formatives Messmodell von einem gegensätzlichen Kausalzusammenhang zwischen latenter Variable und den Indikatoren aus [2]. Demnach werden die Indikatoren als Ursache der latenten Variable interpretiert. Mathematisch wird die latente Variable deshalb durch eine Linearkombination aus den einzelnen Indikatoren (dargestellt in Abb. 2.4 durch x) und den zugehörigen Gewichten (dargestellt in Abb. 2.4 durch γ) dargestellt (vgl. Gleichung in Abb. 2.4). Entsprechend gilt die Überlegung, dass die Veränderung des Messwerts eines Indikators zwangsläufig Auswirkungen auf den Wert der latenten Variable hat. Im Gegensatz zum reflektiven Ansatz sollten die Indikatoren zur Messung einer latenten Variable beim formativen Ansatz möglichst unabhängig voneinander sein [84]. Es soll vermieden werden, dass Indikatoren mit weitestgehend gleichem Informationsgehalt in das Messmodell einer latenten Variable einfließen. Aus diesem Grund sollte die Güteprüfung eines formativen Messmodells die Kollinearität zwischen den einzelnen Indikatoren einer latenten Variable berücksichtigen. Hohe Kollinearität eines Indikators zu einem oder mehreren anderen Indikatoren sagt aus, dass sich der Informationsgehalt jenes Indikators weitestgehend durch die anderen Indikatoren beschreiben lässt. Ein gängiges Maß zur Prüfung der Kollinearität zwischen dem zu prüfenden Indikator und den übrigen Indikatoren im Messmodell ist der sog. *Variance Inflation Factor* (VIF). Zur Berechnung wird eine multiple Regression durchgeführt, wobei der zu prüfende Indikator die abhängige Größe und die übrigen Indikatoren die unabhängigen Größen darstellen. Als Ergebnis erhält man unter anderem das Bestimmtheitsmaß (R^2) der Regression. Der VIF berechnet sich dann wie folgt:

Formel 2.1. Variance Inflation Factor:

$$VIF_n = \frac{1}{1 - R_n^2}$$

VIF_n : Variance Inflation Factor der n -ten unabhängigen Variablen im formativen Messmodell

R_n^2 : Adjustiertes Bestimmtheitsmaß der n -ten unabhängigen Variablen im formativen Messmodell

Laut Diamantopoulos [23] sollte bei einem VIF größer fünf eine genau inhaltliche Prüfung und gegebenenfalls eine Anpassung des Messmodells durchgeführt werden.

Im Rahmen der Evaluationsstudie (vgl. Kap. 6) wurden formative Messmodelle verwendet, weil aus inhaltlichen Überlegungen heraus keine Kollinearität zwischen den formulierten Aussagen zu erwarten war. Es war durchaus denkbar, dass sich einzelne Indikatoren derselben latenten Variable in gegensätzliche Richtungen entwickeln können. Einzelheiten zur Entwicklung der Messmodelle sind in Kap. 6 zur Evaluationsstudie zu finden.

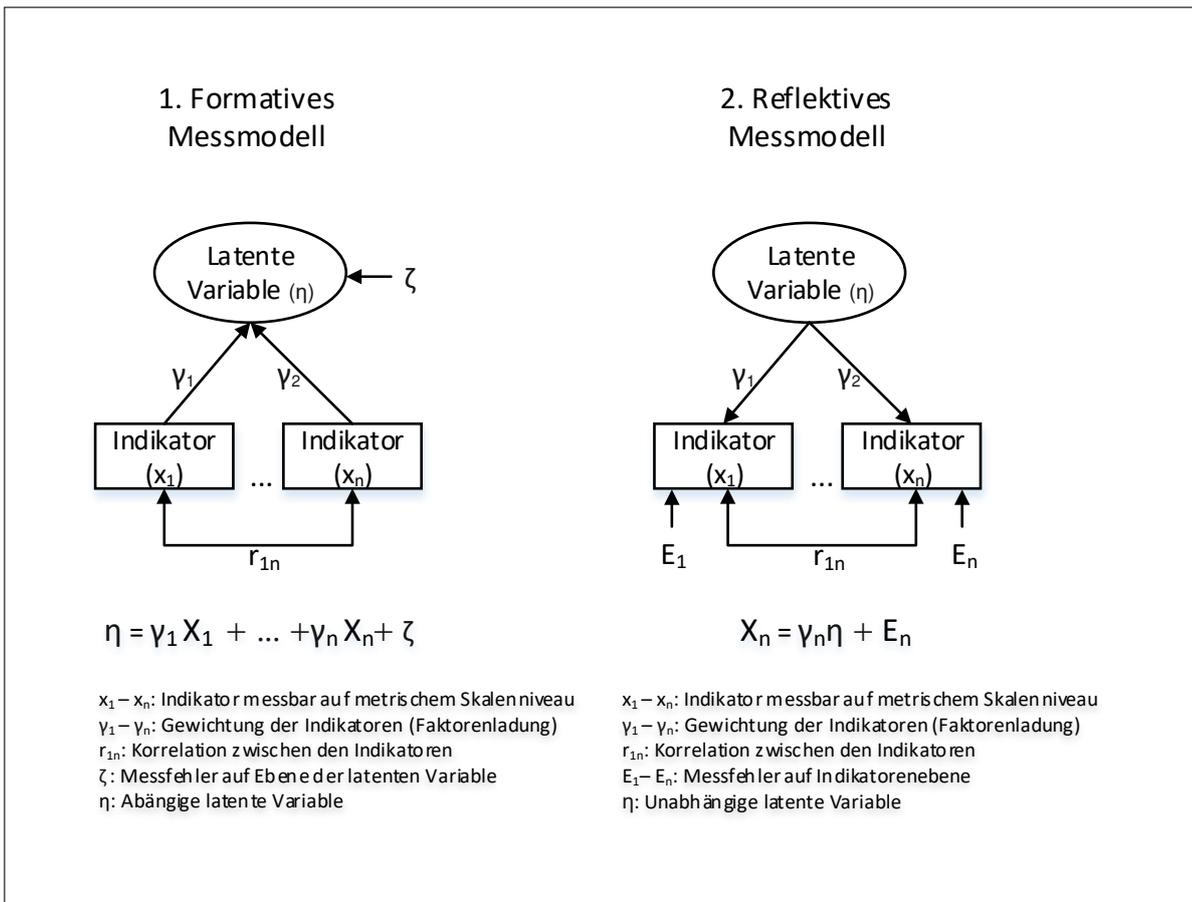


Abbildung 2.4: Vergleich formatives und reflektives Messmodell

2.2.4.3 Die System Usability Scale

Neben den Auswirkungen auf die Gruppenarbeit, die mit dem Bewertungsmodell nach Katsma et al. [46] erfasst werden spielt auch die Gebrauchstauglichkeit gem. DIN EN ISO 9241 [25] (vgl. auch Def.: 6.1) eine wichtige Rolle bei der Evaluation von PMS. Zur Beurteilung der subjektiv wahrgenommenen Gebrauchstauglichkeit ist die System Usability Scale [12] ein bewährtes und leicht anzuwendendes Instrument. Sie besteht aus einem Katalog von insgesamt zehn Aussagen zum untersuchten System, die anhand einer Likert-Skala [52] von jedem Studienteilnehmer bewertet werden (vgl. Anhang C.3). Als Ergebnis der Auswertung erhält man den sog. SUS-Score - einen prozentualen Wert, der einen direkten Vergleich verschiedener Systeme ermöglicht. Zur Interpretation des SUS-Scores veröffentlichte Bangor et al. [7] verschiedene Möglichkeiten (vgl. Abb. 2.5). Zum einen definierte er den sog. Acceptable Range (dt. Akzeptanzbereich) [6], der eine erste Orientierungshilfe gibt, ob die Gebrauchstauglichkeit aus Benutzersicht akzeptabel ist oder nicht (vgl. Abb. 2.5 oben) und zum anderen eine auf dem amerikanischen Schulnotensystem basierende Einstufung (vgl. Abb. 2.5 mittig) [6]. Bemerkenswert dabei ist, dass bereits ab einem SUS-Score kleiner 60 Prozent die schlechteste Note bzw. der nicht akzeptable Bereich beginnt. Dementsprechend eng ist die Aufteilung im Bereich von 60 bis 100 Prozent. Eine weitere Möglichkeit veröffentlichte Bangor et al. [7] ein Jahr später mit den sog. Adjective ratings (dt. Adjektive Bewertung). Dabei erarbeitete er sechs gebräuchliche Adjektive (vgl. Abb. 2.5 unten), die den Grad der Gebrauchstauglichkeit ausdrücken sollen.

Die detaillierte Anwendung der SUS wird in 6.3.3 im Rahmen der Evaluationsstudie beschrieben.

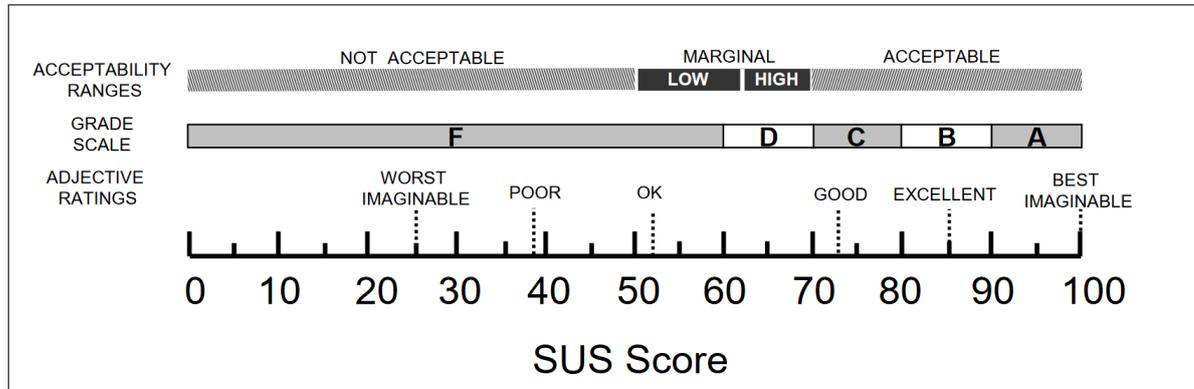


Abbildung 2.5: Interpretationsmöglichkeiten der System Usability Scale
(Quelle: Bangor et al. [7])

Kapitel 3

Theoretische Grundlagen und Stand der Wissenschaft

Dieses Kapitel besteht aus insgesamt drei Teilen. Im ersten Teil (vgl. 3.1) werden die Grundlagen agiler Softwareentwicklung aufgezeigt. Dabei stehen die Grundwerte agiler Softwareentwicklung (vgl. 3.1.1), agile Methoden (vgl. 3.1.2.1 und 3.1.2.2), sowie die Bedeutung physikalischer Medien in agilen Arbeitsumgebungen (vgl. 3.1.3) im Fokus. Als Grundlage für die Konzeptentwicklung (vgl. Kap. 5) werden die wichtigsten Prozessschritte agiler Methoden nach einem allgemeingültigem Modell von Mike Griffiths [31] vorgestellt und anhand von Scrum - der am häufigsten eingesetzten agilen Methode - veranschaulicht. Weiterhin werden die Besonderheiten örtlich verteilter Projekte und die damit einhergehenden Anforderungen an agile Methoden beschrieben. Dazu zählt auch der Einsatz konventioneller PMS, welche eine verteilte Entwicklung ermöglicht sowie die damit zusammenhängenden Auswirkungen auf die Gruppenarbeit in agilen Projekten. Im zweiten Teil des Kapitels (vgl. 3.2) wird das Forschungsgebiet CSCW vorgestellt. Nach einem allgemeinen Überblick des Forschungsgebiets stehen relevante Forschungsarbeiten bzgl. des Einsatzes natürlicher Benutzerschnittstellen im thematischen Umfeld dieser Arbeit im Fokus. Der dritte Teil des Kapitels beschreibt den Kern dieser Forschungsarbeit. Hierfür werden zunächst die Forschungsziele (vgl. Abs. 1.3) näher ausgeführt. Anschließend werden verwandte Forschungsarbeiten betrachtet, um eine Abgrenzung dieser Arbeit auszuarbeiten und die Beiträge zur Wissenschaft zu klären.

3.1 Grundlagen agiler Entwicklung

Der erste Chaos-Report der Standish-Group im Jahr 1994 zeigte erstmals konkret, dass die Konzepte der plangetriebenen Softwareentwicklung nicht ausreichten, um die immer komplexeren und umfangreicheren Anforderungen an Softwaresysteme umzusetzen. Unter anderem zeigte der Chaos-Report die häufigsten Problemursachen plangetriebener Vorgehensmodelle aus Sicht der befragten Personen auf (vgl. Tab. 3.1). An dieser Stelle soll vor allem verdeutlicht werden, welche ursächlichen Probleme zur Entwicklung der agilen Methoden geführt haben. Aus diesem Grund werden explizit noch einmal die Ergebnisse des ersten Chaos-Reports von 1994 aufgelistet.

Tabelle 3.1: Häufigste Problemursachen von Softwareprojekten im Jahr 1994
(Quelle: Chaos-Report 1994 [42] der Standish-Group)

Project Challenged Factors	% of responses
1. Lack of User Input	12,8
2. Incomplete Requirements	12,3
3. Changing Requirements and Specifications	11,8
4. Lack of Executive Support	7,5
5. Technology Incompetence	7,0
6. Lack of Resources	6,4
7. Unrealistic Expectations	5,9
8. Unclear Objectives	5,3
9. Unrealistic Timeframes	4,3
10. New Technology	3,7
11. Other	23,0

Die Verfasser des agilen Manifests um Kent Beck [9] berücksichtigten die Ergebnisse des Chaos-Reports, weshalb agile Methoden Lösungsansätze für viele der in Tab. 3.1 gelisteten Probleme bieten.

Im Rahmen der nächsten Absätze werden die Grundlagen agiler Methoden erläutert und in Bezug zu den entsprechenden Problemursachen gesetzt. Auf eine detaillierte Erklärung der relevanten Problemursachen wird deshalb an dieser Stelle verzichtet und auf die nachfolgenden Absätze verwiesen.

3.1.1 Das agile Manifest - Wertebasis agiler Softwareentwicklung

Das agile Manifest wurde 2001 in Utah in den USA von insgesamt 17 Personen ausgearbeitet und unterzeichnet. Ziel war es, eine schlankere und flexiblere Alternative zur klassischen bzw. plangetriebenen Softwareentwicklung zu bieten, um die zentralen Probleme der Softwarekrise zu lösen. Das agile Manifest enthält keine Prozess- oder Vorgehensbeschreibungen, sondern stellt eine Wertebasis dar, die als Grundlage für zukünftige Entwicklungen im Bereich des Software Engineerings dienen soll. Das agile Manifest besteht aus insgesamt vier Grundwerten, die durch weitere 12 Prinzipien konkretisiert wurden.

Die vier Grundwerte des agilen Manifests [9] lauten wie folgt:

1. *"Individuen und Interaktionen mehr als Prozesse und Werkzeuge"*
2. *"Funktionierende Software mehr als umfassende Dokumentation"*
3. *"Zusammenarbeit mit dem Kunden mehr als Vertragsverhandlung"*
4. *"Reagieren auf Veränderung mehr als das Befolgen eines Plans"*

Individuen und Interaktionen mehr als Prozesse und Werkzeuge: Dieser Grundsatz stellt die Bedürfnisse aller beteiligten Personen und die Anforderungen an eine gut funktionierende Zusammenarbeit in den Fokus. Konkret bedeutet das, dass vor allem der

Einsatz von Prozessen und Werkzeugen (wie z.B. PMS) mit Bedacht beschlossen werden sollte. Ziel muss es immer sein, die Zusammenarbeit der einzelnen Akteure zu fördern. Die Einführung von Prozessen und Werkzeugen zum Selbstzweck (z.B. zur Erfüllung von formalen Vorgaben) lehnt die agile Entwicklung demnach ab.

Funktionierende Software mehr als umfassende Dokumentation: Im Wesentlichen ist die Argumentation hier ähnlich wie beim vorherigen Grundsatz. Die Dokumentationsarbeit sollte auf das Notwendige beschränkt werden und einen Mehrwert bieten. Sobald Dokumentation nur der Erfüllung obligatorischer Prozessvorgaben dient, sollte darauf verzichtet werden. Generell sollten Projektressourcen vorrangig in die Programmierung anstatt in Dokumentationsarbeit fließen.

Zusammenarbeit mit dem Kunden mehr als Vertragsverhandlung: Hinter diesem Grundsatz steckt eine besonders wichtige Aussage. Agile Entwicklung hat das Ziel, dass sowohl Auftraggeber als auch Auftragnehmer an einem Strang ziehen, um ein möglichst gutes Ergebnis zu erreichen. Es soll vermieden werden, dass das Pochen auf Vertragsvorgaben die Entwicklung behindert und Ressourcen verbraucht.

Reagieren auf Veränderung mehr als das Befolgen eines Plans: In der plangetriebenen Softwareentwicklung stellt die Anpassung einer Anforderung in einer späten Projektphase meist ein aufwändiges und teures Unterfangen dar. Aus diesem Grund werden Änderungsanfragen häufig abgelehnt oder zurückgehalten. Da in der agilen Entwicklung ein zufriedenstellendes Endergebnis jedoch oberste Priorität hat, sollten Änderungswünsche auch zu einem späten Projektzeitpunkt noch willkommen sein, auch wenn der Projektplan abgeändert werden muss.

Die bereits erwähnten 12 Prinzipien konkretisieren diese vier Grundsätze und geben erste Anhaltspunkte für die praktische Anwendung. Eine detaillierte Ausführung der Prinzipien ist jedoch nicht Bestandteil dieser Arbeit und kann unter <http://agilemanifesto.org> nachgelesen werden.

3.1.2 Agile Methoden

Seit Veröffentlichung des Agilen Manifests haben sich unterschiedliche agile Methoden entwickelt, deren gemeinsames Ziel die Umsetzung der agilen Grundwerte in der praktischen Softwareentwicklung ist. Die Firma Version One veröffentlicht jährlich den Annual State Of Agile [58], einen Bericht, der die wichtigsten Kennzahlen zur Verbreitung und Anwendung agiler Methoden in der praktischen Softwareentwicklung enthält. Anbei ein kurzer Auszug der am häufigsten eingesetzten agilen Methoden und ihrer Verbreitung in der Praxis:

Tabelle 3.2: Verbreitung agiler Methoden in 2020
(Quelle: Annual State of Agile 2020 [58])

Agile Methode	Anteil in %
Scrum	58
ScrumBan	10
Hybrid (Kombination mehrerer Methoden)	9
Scrum/XP hybrid	8
Kanban	7
Iterative Development	4
Sonstige	3
Lean Startup	1
Extreme Programming (XP)	1

Die Auflistung verdeutlicht, dass eine Vielzahl agiler Methoden Anwendung findet. Die Methode Scrum wird jedoch mit Abstand am häufigsten eingesetzt, weshalb sie in der weiteren Arbeit stellvertretend für agile Methoden verwendet wird. Bezüglich der Forschungsziele spielt die konkrete agile Methode nur eine untergeordnete Rolle, weil die zentralen Bestandteile und Prozessschritte ähnlich sind und die gleichen Prinzipien (vgl. Agiles Manifest [9]) zum Einsatz kommen.

3.1.2.1 Der agile Prozess

Agile Prozesse weisen drei wichtige Merkmale auf, die auf den agilen Grundsätzen und Prinzipien (vgl. Agiles Manifest [9]) beruhen. Sie lauten wie folgt:

- **Transparenz:** Dieses Merkmal steht für Transparenz zwischen allen Stakeholdern. Sowohl innerhalb des Entwicklungsteams als auch zwischen Auftraggeber und Auftragnehmer sollte absolut offen mit Informationen umgegangen werden. Ein agiler Prozess sollte also Werkzeuge bieten, um Informationen einfach und effizient zu verteilen.
- **Inspektion:** In der agilen Entwicklung sollen regelmäßige Abgleiche des Ist-Zustands mit den Projektzielen stattfinden, um festzustellen, ob man noch effektiv arbeitet. Weiterhin wird auch der Prozess an sich betrachtet, um Verbesserungspotentiale zu identifizieren.
- **Adaption:** Identifizierte Verbesserungspotentiale sollte möglichst bald in den Entwicklungsprozess einfließen. Agile Prozesse beinhalten Mechanismen, um dieses Vorgehen zu erleichtern.

Um diese drei Merkmale möglichst gut zu erfüllen, haben sich in der agilen Entwicklung iterative Vorgehensweisen etabliert. Das bedeutet, dass alle klassischen Phasen der Produktentwicklung (z.B. Spezifikation, Entwicklung, Test, Auslieferung usw.) mehrfach durchlaufen werden.

Im folgenden Absatz werden die zentralen Prozessschritte agiler Methoden allgemeingültig

erläutert und anhand von Scrum veranschaulicht. Die folgenden Ausführungen sind an das generische Phasenmodell für agile Methoden von Mike Griffiths [31] sowie an den offiziellen Scrum Guide von Ken Schwaber [75] angelehnt. Ein erster Überblick kann Abb. 3.1 entnommen werden. Die Grafik zeigt den Scrum-Prozess und die nach Griffiths allgemeingültig festgelegten Phasen eines agilen Prozesses. Eine exakte Zuordnung einzelner Scrum-Prozessschritte und Artefakte zu den allgemeingültigen Phasen nach Griffiths ist nicht immer möglich, da die einzelnen Prozessschritte nach Scrum eng verzahnt sind und teilweise überlappen. Die Grundidee agiler Prozesse kann durch die Zuordnung von Scrum zum Phasenmodell jedoch gut veranschaulicht werden.

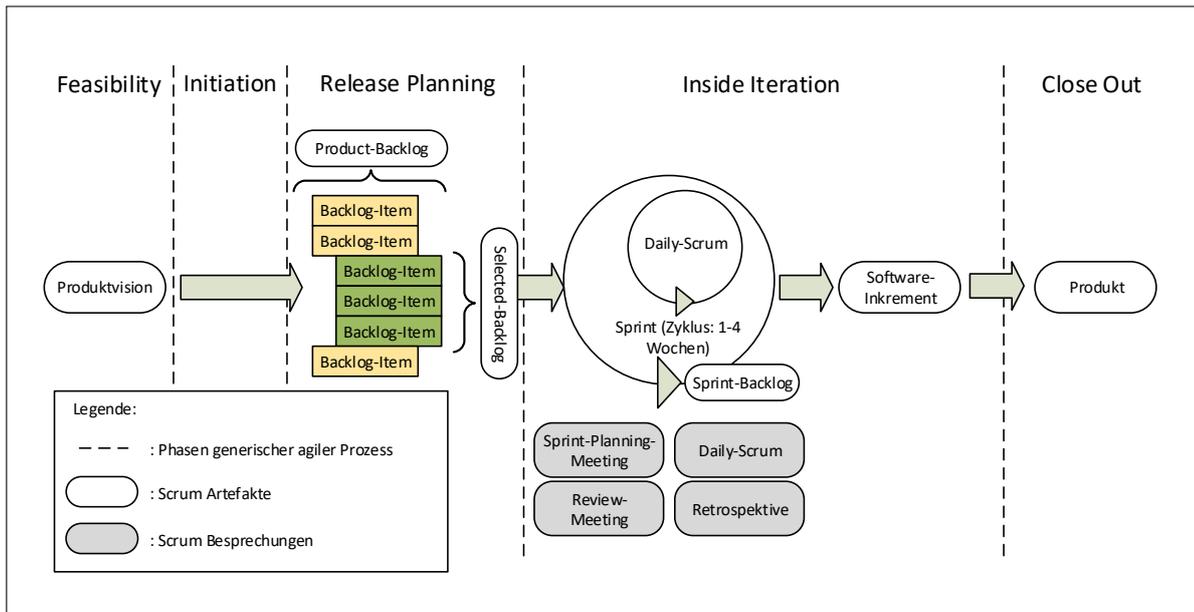


Abbildung 3.1: Der Scrum Prozess nach Schwaber et al. [75] mit den Phasen des generischen Prozessmodells nach Griffiths [31].

1. Feasibility: Allgemein betrachtet dient diese Prozessphase zur Ausarbeitung einer Produktvision sowie zu einer ersten betriebswirtschaftlichen Betrachtung des Projekts [31]. Genaue Vorgaben zum Vorgehen macht der Scrum Prozess nach Schwaber et al. [75] hier nicht. Die Produktvision ist das wichtigste Artefakt und das Ergebnis dieser Prozessphase. Sie sollte die grundlegende Produktidee transportieren und ein übergeordnetes Projektziel beschreiben [64]. Gleichzeitig sollte die formulierte Produktidee genug Freiheiten für die spätere Weiterentwicklung im Rahmen des fortschreitenden Projekts lassen [64]. Gemäß den agilen Grundsätzen ist es auch bei der Produktvision essentiell, dass sie innerhalb des Projekts kommuniziert wird, um ein gemeinsames Verständnis der Produktidee zu erreichen.

2. Initiation: Diese Prozessphase beinhaltet vorbereitende Aufgaben wie z.B. die Zusammenstellung eines Projektteams und die Festlegung der Rahmenbedingungen des Projekts (z.B. zeitlicher Rahmen, Kostenrahmen, Festlegung von Rollen und Verantwortlichkeiten). Diese Aufgaben werden durch den Scrum-Prozess nicht explizit abgedeckt, sind jedoch

fester Bestandteil in jedem Projekt. Ein weiterer wichtiger Schritt der Initiation-Phase ist die Vorbereitung des Product-Backlogs. Beim Product-Backlog handelt es sich um die Gesamtheit aller noch umzusetzenden Anforderungen im Projekt, die an einer zentrale Stelle gesammelt und priorisiert werden. Das Product-Backlog dient als einzige Quelle für Anforderungen (egal ob Funktionen, Verbesserungen oder Fehlerbehebungen) und entwickelt sich während des Projekts ständig weiter. Die initiale Erstellung des Product-Backlogs ist Teil der Initiation-Phase und wird im Regelfall im Rahmen eines Anforderungsworkshops durchgeführt. Eine erste Aufwandsabschätzung ist ebenfalls Teil des Anforderungsworkshops.

3. Release-Planning: Ein Release ist eine Softwareversion, die an den Kunden ausgeliefert wird. Ziel des Release-Plannings ist die zeitliche und funktionale Ausplanung dieser Softwareversionen (vgl. Wirdemann [85]). Es muss festgelegt werden, welche Funktionsumfänge in den einzelnen Versionen enthalten sind und wann die Software ausgeliefert wird. Eine detaillierte Planung, die weit in die Zukunft reicht ist aus Sicht der agilen Softwareentwicklung nur bedingt sinnvoll, da auch hier der agile Grundsatz "Reagieren auf Veränderung ist mehr als das Befolgen eines Plans" beachtet werden muss. Trotz vorausschauender Release-Planung sollten sinnvolle Änderungen jederzeit möglich sein. Je weiter man in die Zukunft plant, umso unwahrscheinlicher ist eine korrekte Einhaltung des Release-Plans. Der Scrum-Prozess nach Schwaber et al. sieht eine Release-Planung nicht direkt vor, weil er die Annahme trifft, dass jedes Softwareinkrement (vgl. Phase 4 - Inside Iteration) eine ausreichende Qualität besitzt, um an den Kunden ausgeliefert werden zu können. Scrum legt besonderen Wert auf eine genaue Planung der einzelnen Softwareinkremente. Hierfür werden Teile des Product-Backlogs ausgewählt und anschließend in die nächste Iteration eingeplant (vgl. Phase 4 - Inside Iteration). Diese Vorauswahl wird in Scrum häufig als Selected-Backlog (in Abb. 3.1 grün gekennzeichnet) bezeichnet.

4. Inside Iteration: Wie eingangs bereits erwähnt ist die iterative Vorgehensweise ein Basiselement agiler Prozesse. Iteratives Vorgehen bedeutet, dass alle Phasen eines plangetriebenen Entwicklungsprozesses für Teilfunktionalitäten der zu entwickelnden Software immer wieder durchlaufen werden. In Abb. 3.1 wird dieses Prinzip durch den großen kreisförmigen Pfeil gekennzeichnet. Der entscheidende Vorteil einer iterativen Vorgehensweise ist, dass zahlreiche Softwareinkremente entstehen, die dem Auftraggeber und auch den potentiellen Benutzern gezeigt werden können, um entsprechendes Feedback zu erhalten. Im Rahmen von Scrum wird eine Iteration als Sprint bezeichnet. Jeder Sprint beginnt mit dem Sprint-Planning-Meeting, dessen Ziel die genaue Festlegung des Sprint-Backlogs ist. Als Sprint-Backlog werden die Anforderungen bezeichnet, die im Rahmen des aktuellen Sprints umgesetzt werden sollen. Zu diesem Zweck wird das Selected-Backlog (vgl. Phase 3 - Release Planning) herangezogen, um die enthaltenen Anforderungen erneut bzgl. des Aufwands zu bewerten und letzte Unklarheiten zu klären. Anschließend wird eine Teilmenge des Selected-Backlogs als Sprint-Backlog festgelegt. Dieses Schritt muss einvernehmlich zwischen Auftraggeber und Auftragnehmer (inkl. Entwicklern) erfolgen. Anschließend startet die Entwicklungsphase, welche je nach Iterationslänge zwischen ein und vier Wochen dauern kann. Während der Entwicklung gibt es das sog. Daily-Scrum (in Abb. 3.1 durch den kleinen kreisförmigen Pfeil dargestellt).

Dabei handelt es sich um eine tägliche, 10-minütige Besprechung, welche genutzt wird, um einen Statusaustausch innerhalb des Entwicklerteams durchzuführen. Jeder Entwickler gibt einen kurzen Überblick seiner aktuellen Arbeit und nennt ggf. Probleme, die seine Arbeit behindern. Am Ende des Sprints findet das Review-Meeting statt. Es dient im Rahmen des Scrum-Prozesses, um die Arbeitsergebnisse des aktuellen Sprints (das Softwareinkrement) dem Auftraggeber vorzustellen. Dieser Schritt schafft Transparenz zwischen allen Beteiligten und ist ein wichtiger Mechanismus, der die eingangs genannten Merkmale Inspection und Adaption auf inhaltlicher Ebene realisiert. Ein weiteres Werkzeug zur Umsetzung dieser beiden Merkmale ist die Retrospektive. Dabei geht es jedoch nicht um inhaltliche Aspekte der Software, sondern um prozessrelevante Themen. Die Retrospektive ist eine Besprechung, die ohne Beteiligung des Auftraggebers stattfindet. Das Ziel ist die Identifikation von Verbesserungspotentialen rund um die Umsetzung des agilen Prozesses. Dabei können z.B. Probleme mit Arbeitsabläufen, mit eingesetzten Entwicklungswerkzeugen aber auch soziale Aspekte diskutiert werden. Wichtig hierbei ist, dass konkrete Verbesserungsmaßnahmen erarbeitet werden, die in einem der nächsten Sprints umgesetzt werden. Dieses Prinzip ermöglicht eine stetige Verbesserung der Entwicklung und steigert die Produktivität.

5. Close Out: Diese Projektphase wird nur aus Gründen der Vollständigkeit erwähnt. Hierzu macht weder das Phasenmodell nach Griffiths [31] noch der Scrum-Prozess nach Schwaber et al. [75] genaue Angaben. Im Regelfall kommen hier projektspezifische Vorgaben zum Tragen wie z.B. die Überführung der Software in den produktiven Betrieb oder Wartungsaufgaben. Im Rahmen der vorliegenden Arbeit hat die Close Out Phase keine besondere Relevanz.

3.1.2.2 Das Anforderungsmanagement mit User-Stories

Im Kontext agiler Methoden sind sog. User-Stories ein häufig eingesetztes Mittel, um Anforderungen auszuarbeiten und zu planen. Im Gegensatz zu klassischen Anforderungsdokumenten (z.B. Lastenheft, Pflichtenheft) beinhalten sie jedoch keine detaillierte Spezifikation sondern geben lediglich eine knappe Beschreibung der Funktion aus Sicht eines Benutzers, sowie eine Reihe von Akzeptanzkriterien [17]. Diese Herangehensweise passt sehr gut zu den agilen Grundsätzen, weil Anforderungen nicht gleich zu Beginn eines Projektes ausgearbeitet und dokumentiert werden müssen. Vielmehr gilt hier die Annahme, dass Anforderungen sich im Laufe der Entwicklung verändern und erst kurz vor der Implementierung detailliert festgelegt werden können. Dieses Konzept adressiert die häufigste Fehlerursache plangetriebener Vorgehensweisen - die mangelnde Einbindung des Benutzers (vgl. Tab. 3.1). Sowohl Benutzer als auch Auftraggeber werden während der gesamten Projektphase immer wieder eingebunden, um Anforderungsdetails zu klären und User-Stories zu formulieren. Weiterhin wird aktiv die Kommunikation zwischen den einzelnen Stakeholdern gefördert. Missverständnissen wie sie beim Einsatz klassischer Anforderungsdokumente häufig entstehen wird also vorgebeugt.

Um die Formulierung von User-Stories zu vereinfachen, wird häufig ein Template nach Cohn [17] verwendet, das sich folgendermaßen zusammensetzt:

"Als (Rolle) möchte ich (Funktion), um (Mehrwert/Nutzen)."

Ein Beispiel einer typischen User-Story könnte also folgendermaßen lauten:

Als Bankkunde möchte ich meine Kontoauszüge als PDF Datei herunterladen, um sie auf meiner Festplatte archivieren zu können.

Um weitere Details einzubringen eignen sich die bereits erwähnten Akzeptanzkriterien sehr gut. Sie sind obligatorischer Bestandteil jeder User-Story und dienen neben ihrem eigentlichen Zweck - der Durchführung von Akzeptanztests - dem Entwickler als Grundlage für ein besseres Verständnis der Anforderungen. Für das eben aufgeführte Beispiel könnte man folgende Akzeptanzkriterien formulieren:

- **Die Kontoauszüge der letzten vier Wochen lassen sich herunterladen und speichern.**
- **Ich kann die PDF Datei des Kontoauszugs öffnen und ausdrucken.**
- **In der geöffneten Datei sind alle Kontoumsätze des jeweiligen Zeitraums gut lesbar aufgelistet.**

Um das Konzept der User-Stories zu verstehen, ist es notwendig das Ebenenmodell in Anlehnung an Leffingwell [51] zu kennen (vgl. Abb. 3.2). Wie aus Abb. 3.1 hervorgeht starten agile Prozesse mit der Produktvision. Sie beschreibt auf oberster Ebene das grundlegende Ziel des Projekts sowie die wichtigsten Anforderungen an die zu entwickelnde Software. In einem ersten Schritt werden sog. Epics abgeleitet und als User-Story formuliert. Als Epics bezeichnet man User-Stories der obersten Ebene (vgl. Abb. 3.2). Sie sind durch einen hohen funktionalen Umfang und einen entsprechend niedrigen Detailgrad gekennzeichnet. Dieser erste Schritt hat den Vorteil, dass sich der funktionale Umfang der gesamten Software zunächst gut überblicken lässt, eine einfache Priorisierung möglich ist und man sich noch nicht mit den Anforderungsdetails beschäftigen muss. Der Nachteil von Epics ist, dass sie aufgrund des geringen Detailgrads weder zur Umsetzung noch zu einer genauen Aufwandsabschätzung geeignet sind.

Aus diesem Grund werden in einem nächsten Schritt sog. Stories abgeleitet. Dabei handelt es sich um User-Stories mit einem geringeren funktionalen Umfang aber einem entsprechend höheren Detailgrad. Hierbei spricht man auch von einer höheren Granularität. An dieser Stelle muss außerdem darauf hingewiesen werden, dass sich die Begriffe Epic bzw. Story auf das Ebenenmodell beziehen und der Ausdruck User-Story als Überbegriff für beide Arten verwendet wird. Die Idee hinter der schrittweisen Aufteilung von Epics ist, dass dieser Schritt zunächst nur für höher priorisierte Epics durchgeführt wird. Dabei gilt wiederum der Grundsatz, dass Details erst wenige Wochen vor der Umsetzung ausgearbeitet werden, da sich die genauen Anforderungen jederzeit noch ändern könnten. Zur Aufteilung von User-Stories (bzw. zur Erhöhung der Granularität) gibt es praxisnahe Verfahren von Wirdemann [85] und Lawrence [50], die diese Aufgabe erleichtern sollen. Laut Leffingwell [51] besitzt eine Story die richtige Granularität, wenn sie innerhalb einer

Iteration umgesetzt werden kann.

Auf der untersten Ebene im Modell befinden sich die sog. Tasks. Im Unterschied zu User-Stories beschreiben sie keine abgeschlossene Anforderung aus Kundensicht, sondern lediglich eine konkrete Aufgabe, die von einem Entwickler durchgeführt werden muss. Im Unterschied zu User-Stories werden die Tasks von den Entwicklern im Rahmen des Sprint-Plannings (vgl. 3.1.2.1) erstellt. Sie besitzen demnach die höchste Granularität.

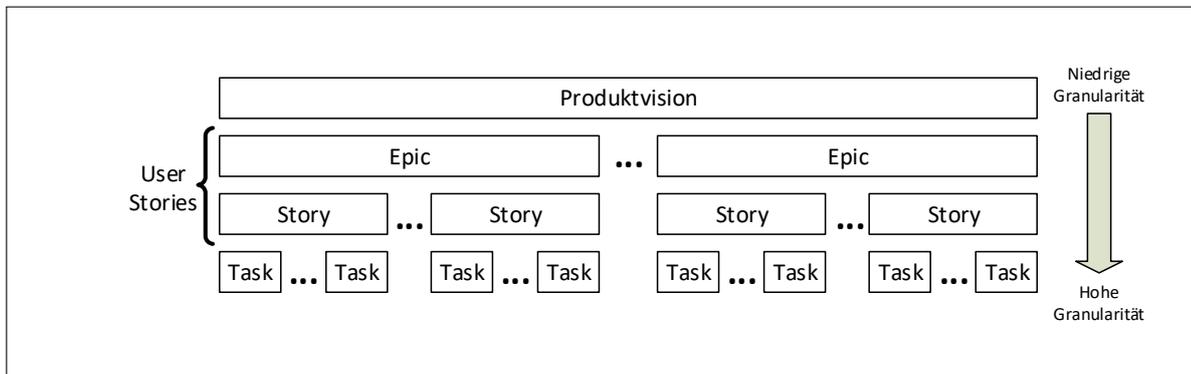


Abbildung 3.2: Ebenenmodell zur Granularität von User-Stories nach Leffingwell [51].

3.1.3 Agile Arbeitsumgebungen

Die Forderung nach Transparenz und direkten Kommunikationswegen (von Angesicht zu Angesicht) ist ein wichtiger Grundsatz agiler Softwareentwicklung (vgl. Agiles Manifest [9]). Die Arbeitsumgebungen agiler Projektteams versuchen dieser Forderung durch die Umsetzung des *Open Workspace Principle* (OWP) nachzukommen. Laut Martin et al. [53] umfasst diese Prinzip im Wesentlichen folgende Eigenschaften eines Arbeitsbereichs:

- **Offen:** Dieser Aspekt bedeutet, dass agile Projektteams bevorzugt in einem einzelnen Arbeitsraum untergebracht werden sollten, um spontane Gespräche und Situationsbewusstsein bei den Teammitgliedern zu fördern. Laut Martin et al. erreicht man das am besten, indem alle Teammitglieder in Hörweite zueinander arbeiten.
- **Informativ:** Dieser Punkt bedeutet, dass alle relevanten Informationen zum Projekt direkt zugänglich und für alle einsehbar sein sollen. In agilen Arbeitsbereichen wird das im Regelfall durch physikalische Medien (meist papierbasiert) realisiert, indem zahlreiche Poster, Flip-Charts, Karteikarten, usw. an Stellwänden oder direkt an der Wand aufgehängt und für die tägliche Arbeit verwendet werden. Dieser Aspekt trägt wesentlich zur Transparenz im Team bei, weil Statusupdates sofort wahrgenommen und für alle ersichtlich sind.
- **Privat:** Das OWP gesteht jedem Teammitglied auch ein Stück Privatsphäre zu. Das erreicht man z.B. durch Einzelschreibtische, die so positioniert werden, dass

die Bildschirme nicht direkt von anderen Arbeitsplätzen aus einsehbar sind. Trotz der Forderung nach Transparenz sollten sich die Teammitglieder nicht ständig beobachtet fühlen.

Eine häufig zitierte Studie der Universität von Michigan von Teasley et al. [82] zeigt deutlich die positiven Effekte des OWP auf die Produktivität eines Projektteams in der Softwareentwicklung. Hierfür wurden insgesamt sechs Teams mit je sechs bis acht Mitgliedern über einen Zeitraum von vier Monaten untersucht. Die Arbeitsumgebungen waren entsprechend dem OWP konzipiert. Während der Studie wurden unter anderem die Produktivität, die Präferenz der Teammitglieder für Arbeitsumgebungen nach OWP bzw. klassisch aufgeteilten Büros sowie die Zufriedenheit des Auftraggebers bzw. der Benutzer mit dem Endprodukt gemessen. Zur Messung der Produktivität wurden Function Points (vgl. Jones et al. [43]) herangezogen und mit dem bisherigen Firmendurchschnitt des Auftraggebers verglichen. Hier lies sich eine Verdoppelung der Produktivität feststellen. Die Zufriedenheit des Auftraggebers bzw. der Benutzer mit dem Produkt blieb mit durchschnittlich vier von fünf Punkten auf einem gleich hohen Niveau wie der Firmendurchschnitt. Nach dem Projekt bevorzugte die Mehrheit der Entwickler die Arbeit in einem Arbeitsumfeld nach dem OWP.

Eine weitere Arbeit von de Melo Oliveira et al. [55] untersucht im Rahmen einer explorativen Interviewstudie acht agile Projekte, um die wichtigsten Erfolgsfaktoren im Umgang mit agilen Arbeitsumgebungen zu ermitteln und zu beschreiben. Hierfür wurden alle 48 Mitglieder der acht untersuchten Teams befragt, um die wichtigsten Probleme und Lösungsansätze im Kontext agiler Arbeitsumgebungen zu verstehen. Auf Basis der Ergebnisse und weiter Literaturrecherche formulierte de Melo Oliveira et al. folgende sieben Heuristiken zur Optimalen Einrichtung und Nutzung agiler Arbeitsumgebungen. Der darin verwendete Begriff *Informative Workspace* (IW) wurde von Kent Beck geprägt [8] und wird im Kontext agiler Methoden zur Bezeichnung von Arbeitsbereichen verwendet, die das OWP umsetzen.

- H.1:** Keep the most important frames/boards near to daily meeting's place.
- H.2:** Share IW related work between all team members.
- H.3:** Focus your IW on current specific needs.
- H.4:** Display information so as to be easily assimilated by the human brain.
- H.5:** Make your daily IW tasks made by a pair of team's members.
- H.6:** Provide an easy-maintainable and practical IW.
- H.7:** Remove lower value frames/boards.

Zwei Jahre später veröffentlichte de Melo Oliveira et al. eine weitere Arbeit [56], die sich mit der praktischen Umsetzung der Heuristiken beschäftigt. Hierfür wurden Personen mit einschlägiger Erfahrung in der agilen Softwareentwicklung befragt (z.B. Agile Coaches, Scrum-Master). Im Fokus stand warum, wann und wie sie die einzelnen Heuristiken in einem agilen Projekt umsetzen würden. Die Ergebnisse können als Orientierung

und Leitfaden für die Konzeption und Umsetzung agiler Arbeitsumgebungen eingesetzt werden.

3.1.3.1 Bedeutung physikalischer Medien für die Projektarbeit

Aufgrund der großen Bedeutung physikalischer Medien für diese Forschungsarbeit zeigt dieser Absatz verschiedene Studien auf, die bereits deren Relevanz in der agilen Softwareentwicklung untersucht haben. Zunächst muss jedoch definiert werden, was im Rahmen dieser Arbeit unter dem Begriff physikalische Medien verstanden wird:

Definition 3.1. *Physikalische Medien:* *Unter physikalischen Medien werden im Kontext dieser Arbeit analoge (nicht elektronische) Medien verstanden, die das direkte Speichern, Bearbeiten und Visualisieren von Informationen erlauben.*

Im Rahmen der Arbeit zählen vor allem Dinge wie Plakate, Stellwände, Flip-Charts und Whiteboards zu den typischen Vertretern.

Die erste relevante Forschungsarbeit von Paredes et al. [61] ist eine Literaturrecherche bzgl. Visualisierungstechniken von Softwareartefakten in agilen Projekten. Dabei geht es besonders um die Frage, welche Visualisierungstechniken zu Transparenz und Situationsbewusstsein in agilen Teams beiträgt. Paredes et al. ordnet die recherchierten Publikationen vier verschiedenen Kategorien zu, die typischerweise zu den Aufgaben eines Softwareentwicklers zählen: Softwaredesign, Softwareentwicklung, Kommunikation und Statusvisualisierung. Ein wichtiger Aspekt physikalischer Medien zur Unterstützung im Bereich Softwaredesign sind Flexibilität und Freiheitsgrade bei der Benutzung. Der Einsatz von Softwaretools (z.B. bei der Erstellung von UML-Diagrammen) wird hier eher als Einschränkung statt als Hilfe empfunden. In der Kategorie Softwareentwicklung werden physikalische Medien weniger stark eingesetzt. Hier kommen elektronische Visualisierungen von z.B. Build-Status, Testabdeckung, Quellcode-Metriken zum Einsatz. Dabei handelt es sich um automatisch generierte Kennzahlen, die häufig aktualisiert und verändert werden. Aus diesem Grund sind physikalische Medien nur bedingt geeignet. Anders verhält es sich im Bereich Kommunikation. Hier spielen physikalische Medien in Form von Stellwänden und Plakaten eine große Rolle. Laut Paredes et al. werden unterschiedlichste Darstellungen eingesetzt, um wichtige Informationen im Arbeitsbereich zu kommunizieren. Die Recherchen von Paredes et al. zeigen ebenfalls auf, dass elektronische Hilfsmittel zur Darstellung statischer Projektinformationen (z.B. Termine, Deadline, Projektpläne, etc.) vergleichsweise wenig zum Einsatz kommen. Der am stärksten durch physikalische Medien geprägte Aufgabenbereich ist die Visualisierung des Projektstatus. In der agilen Softwareentwicklung wird der Arbeitsfortschritt meist mit Hilfe papierbasierter Medien dargestellt. Laut Paredes et al. wirkt sich die Visualisierung des Arbeitsfortschritts entscheidend auf die Produktivität eines Projektteams aus. Interessant ist weiterhin, dass der Einsatz elektronischer Medien im Vergleich zu physikalischen Medien weniger gut zur Produktivität im Projekt beiträgt,

weil Transparenz und direkte Kommunikation nicht so stark gefördert werden.

Eine weitere Publikation, die für den Einsatz physikalischer Medien in agilen Projekten spricht wurde von Tom Perry [63] im Jahr 2008 veröffentlicht. Dabei handelt es sich um Praxisberichte agiler Teams, die ein physikalisches Task-Board (vgl. Abs. 3.1.3.3) durch PMS ersetzen, um den darauf abgebildeten Projektstatus Auftraggebern und örtlich verteilten Entwicklungsteams zugänglich zu machen. Perry beobachtete dabei, dass die Beteiligung der Teammitglieder an Diskussionen während der zahlreichen Besprechungen stark zurückging. Der Fokus war in erster Linie auf den Moderator und den Beamer gerichtet anstatt auf die fachliche Diskussion. Weiterhin ließ die Aufmerksamkeit während des Daily-Scrum stark nach. Auch die Transparenz im Team ließ vor allem während der Entwicklungsphasen stark nach, weil Statusupdates in der PMS häufig gar nicht oder erst stark zeitverzögert wahrgenommen wurden.

Auch Sharp et al. [78, 76] beschäftigte sich im Rahmen zweier qualitativer Studien mit der Bedeutung physikalischer Medien für die agile Projektarbeit. Die erste Studie [78] nutzte den sog. *Distributed Cognition of Teamwork* (DiCoT) Ansatz nach Blanford et al. [11]. Dabei handelt es sich um eine systematische Vorgehensweise zur Analyse und Beurteilung von Gruppenarbeit auf Basis der Distributed Cognition Theorie nach Hollan et al. [37]. Distributed Cognition betrachtet die Gruppenarbeit als ein großes kognitives System bestehend aus Personen, Arbeitsumgebung und Artefakten. Der DiCoT Ansatz unterstützt die Analyse des Informationsflusses, die Verwendung von Artefakten und die Interaktion der einzelnen Personen bei der Teamarbeit. Sharp et al. untersuchte insgesamt vier agile Projektteams, die das physikalische Task-Board durch PMS ersetzen. Sie stellte negative Effekte in den Bereichen Situationsbewusstsein, Transparenz und Motivation fest.

In einer weiterführenden Arbeit [76] ging Sharp et al. detaillierter auf die Bedeutung eines physikalischen Task-Boards für die agile Gruppenarbeit ein. Sie stellte fest, dass aus inhaltlicher und fachlicher Sicht die Vorteile von PMS gegenüber physikalischen Task-Boards überwiegen. Hierzu zählt sie vor allem die dezentrale Verfügbarkeit, automatisierte Möglichkeiten zur Archivierung von Daten und die Generierung von Projektkennzahlen. Wichtiger erachtet Sharp et al. jedoch die soziale Komponente eines Task-Boards. Dazu zählt sie Aspekte wie Motivation, Kreativität und die Förderung direkter Gespräche in der Gruppenarbeit. Diese Eigenschaften erfüllt nach Sharp et al. ein physikalisches Task-Board besser als PMS. Diese Erkenntnis sollte laut Sharp et al. bei der geplanten Einführung von PMS unbedingt berücksichtigt werden.

Eine weitere Arbeit von Katsma et al. [46] vergleicht die Eignung von PMS im Vergleich zu physikalischen Task-Boards im Kontext global verteilter Softwareentwicklung. Hierfür wurde zunächst eine Literaturrecherche zu den Vor- und Nachteilen physikalischer Task-Boards bzw. PMS durchgeführt. Neben der zentralen Verfügbarkeit projektrelevanter Daten nennt Katsma et al. die folgenden Vorteile des PMS-Einsatzes als Ergebnis seiner Recherche: (1) Daten können automatisiert archiviert und durchsucht werden. Dieser Punkt ist wichtig, um vergangene Sachverhalte aufklären und nachvollziehen zu können.

- (2) Außerdem können Projektkennzahlen automatisch generiert und dargestellt werden.
- (3) Ein weiterer Vorteil nach Katsma et al. ist die automatische Erfassung der Arbeitszeit an bestimmten Themen. Diese Information kann genutzt werden, um Projektplanung und Aufwandsabschätzung zu optimieren. Als Schwachpunkt von PMS nennt Katsma et al. ebenfalls die negativen Auswirkungen auf Kommunikation und Transparenz innerhalb agiler Projektteams.

Der zweite Teil der Arbeit befasst sich mit einem Bewertungsmodell für PMS. Das Bewertungsmodell wird im Rahmen dieser Arbeit für die Evaluationsstudie herangezogen (vgl. Kap. 6). Eine detaillierte Beschreibung des Bewertungsmodells befindet sich in Abs. 6.2.1, weshalb es an dieser Stelle nicht näher erläutert werden soll.

Nachdem nun die Bedeutung physikalischer Medien auf Basis verschiedener Publikationen herausgearbeitet wurde, soll in den nachfolgenden Absätzen das Product-Backlog-Chart sowie das Task-Board vorgestellt werden. Beides sind wichtige Vertreter physikalischer Medien in agilen Entwicklungsprojekten.

Diese Forschungsarbeit von Richter et al. [71] beschäftigte sich ebenfalls mit den Konflikten, die aus dem Einsatz agiler Methoden innerhalb großer, verteilter Softwareprojekte entstehen können. Dabei wurden insbesondere auch Problemstellungen betrachtet, die innerhalb traditionell organisierter Unternehmen entstehen können. Im Unterschied zu dieser Dissertation wurde der Fokus jedoch nicht auf eine Problembetrachtung aus CSCW Perspektive beschränkt. Studienziel der explorativen Interviewstudie war es, konkrete Problemfelder, die aus den genannten Konfliktbereichen entstehen aufzuzeigen, konkret zu benennen und mögliche Lösungsansätze zu diskutieren. Eine Relation zu dieser Arbeit entsteht, weil auch im Rahmen dieser Studie der Konflikt zwischen agilen Arbeitsumgebungen, dem damit verbundenen Einsatz physikalischer Medien und der Notwendigkeit des PMS Einsatzes zur ortsunabhängigen Bereitstellung projektrelevanter Daten adressiert wurde. Es wurde deutlich, dass der Einsatz physikalischer Medien innerhalb agiler Teams präferiert wird. Gleichzeitig wurde aber auch klar, dass dies immer in Kombination mit konventioneller PMS geschieht und die Projektteams der Studienteilnehmer durch redundante und deshalb häufig inkonsistenten Daten (zwischen physikalischer Arbeitsumgebung und PMS) Nachteile in Kauf nehmen müssen. Dieses Ergebnis bekräftigt die Relevanz der vorliegenden Dissertation.

3.1.3.2 Das Product-Backlog-Chart

Das Product-Backlog-Chart ist ein sehr häufig eingesetztes Mittel, um die aus der Produktvision erarbeiteten Epics und Stories übersichtlich zu visualisieren. Alle Epics werden in der ersten Zeile an einer Stellwand angebracht (vgl. Abb. 3.3). Darunter werden spaltenweise die zugehörigen User-Stories angeordnet. Häufig werden die User-Stories dabei der Priorität nach (von oben nach unten abnehmend) angeordnet.

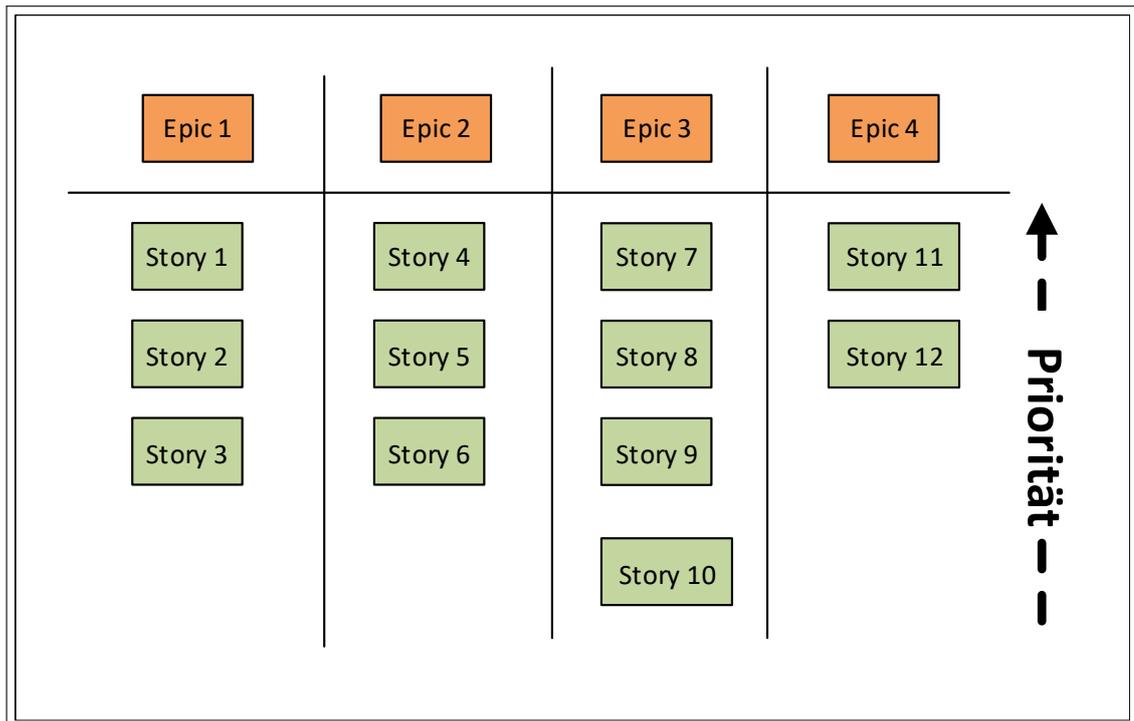


Abbildung 3.3: Beispiel zur Darstellung des Product-Backlogs auf einer Stellwand

3.1.3.3 Das Task-Board

Das Task-Board ist das bekannteste und zugleich wichtigste Werkzeug zur Unterstützung agiler Softwareentwicklung [31]. Ursprünglich wurde es als Teil der Projektmanagementmethode Kanban konzipiert, die bereits in den 1950ern von Taiichi Ōno [59] entwickelt wurde. Ein wichtiges Prinzip der Kanban-Methode ist die Visualisierung aller Arbeitspakete, an denen gerade gearbeitet wird, um die momentane Arbeitslast des Projektteams kontrollieren zu können. Die Anzahl der in Arbeit befindlichen Arbeitspakete wird dabei limitiert. Damit möchte man erreichen, dass sich die Projektmitarbeiter auf einige wenige Aufgaben konzentrieren, um diese möglichst effizient abschließen zu können.

Das Task-Board sorgt in diesem Zusammenhang für die notwendige Transparenz, indem es den aktuellen Bearbeitungszustand anfallender Arbeitspakete übersichtlich darstellt. Die Grundidee des Task-Boards basiert auf einer großen Stellwand, die im gemeinschaftlichen Arbeitsbereich des Projektteams aufgestellt wird. Wie in Abb. 3.4 dargestellt, werden die möglichen Bearbeitungszustände eines Arbeitspakets durch einzelne Spalten repräsentiert. Im dargestellten Beispiel wird eine minimalistische Version mit nur drei möglichen Zuständen dargestellt. Die Bearbeitungszustände können aber entsprechend den individuellen Erfordernissen eines Projektes angepasst werden. Die Arbeitspakete werden durch einfache Karteikarten oder Haftnotizen am Task-Board angebracht und gemäß Arbeitszustand in die jeweiligen Spalten eingeordnet. Während eines laufenden Projekts sind die Bearbeiter einer Aufgabe für die Aktualisierung des Bearbeitungszustands verantwortlich.

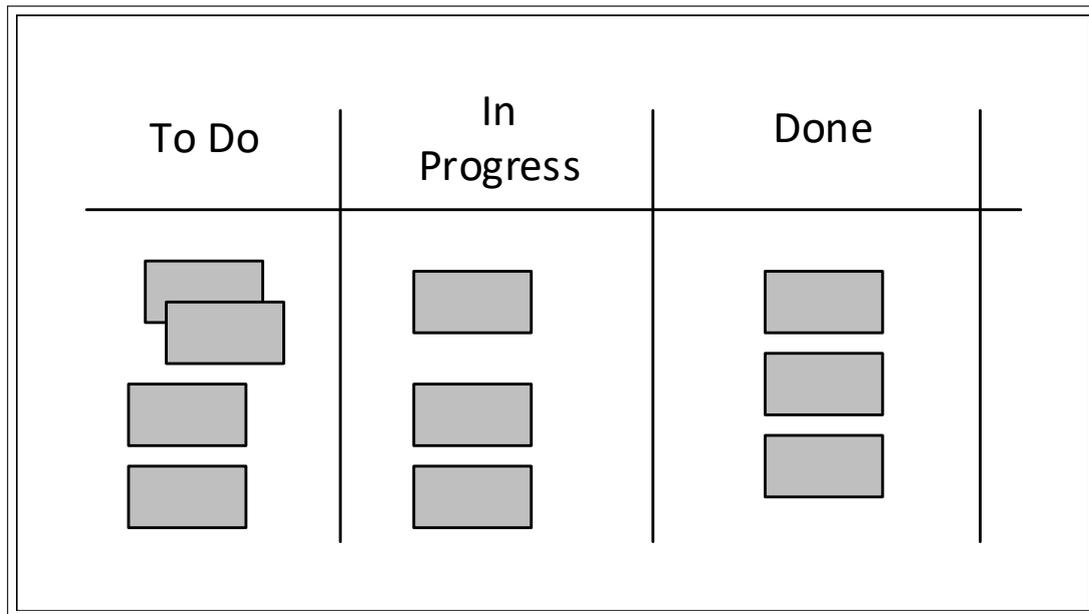


Abbildung 3.4: Beispiel eines einfachen Task-Boards.

Für die Verwendung in Scrum sind einige Anpassungen an der minimalistischen Version des in Abb. 3.4 gezeigten Task-Boards notwendig (vgl. Abb. 3.5). Im Rahmen von Scrum repräsentiert das Task-Board den Inhalt eines einzelnen Sprints. Zur Verwaltung der Anforderungen mit Hilfe der in Abs. 3.1.2.2 vorgestellten User-Stories wird das Task-Board folgendermaßen angepasst:

Zunächst wird eine Spalte zur Aufnahme der einzelnen User-Stories ergänzt. Diese Spalte repräsentiert dementsprechend des Sprint-Backlog. Außerdem wird pro User-Story eine Zeile hinzugefügt. Hierdurch wird die Zuordnung einzelner Tasks zu einer User-Story ermöglicht. Während des Sprint-Planning-Meetings werden alle User-Stories im Sprint-Backlog herangezogen, um einzelne Arbeitspakete abzuleiten. Diese werden dann auf einer Karteikarte kurz notiert und in der Spalte "To Do" neben der zugehörigen User-Story aufgehängt. Sobald alle Arbeitspakete den Endzustand "Done" erreicht haben gilt eine User-Story als abgeschlossen. Der Sprint-Ziel ist erreicht, sobald alle User-Stories vollständig abgeschlossen wurden.

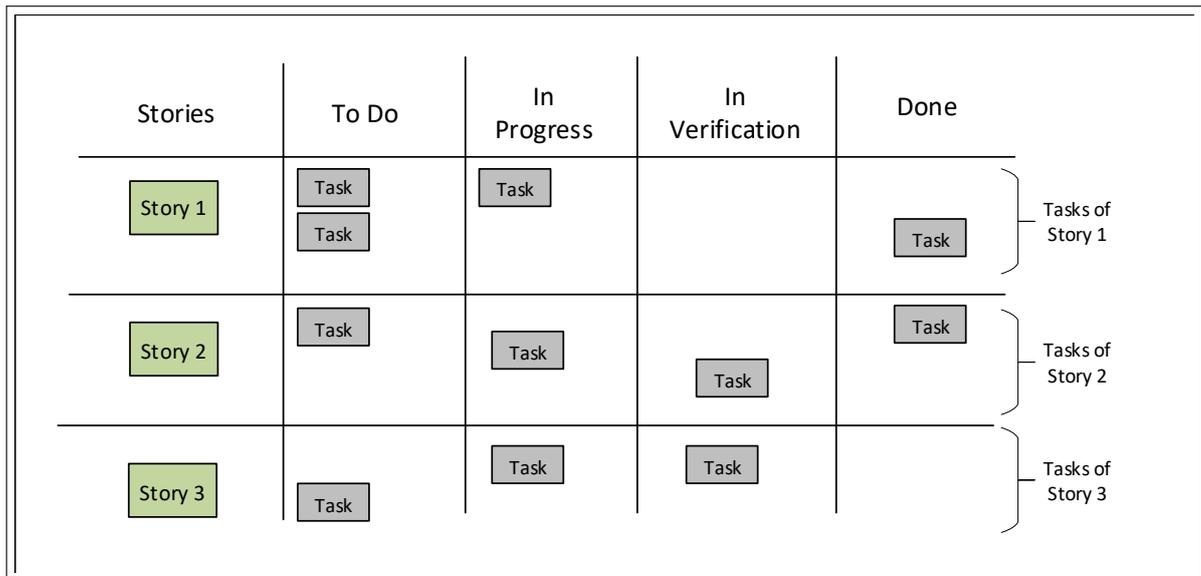


Abbildung 3.5: Beispiel eines einfachen Task-Boards für die Verwendung in Scrum.

Das Task-Board erfüllt neben der Visualisierung der Arbeitszustände weitere wichtige Aufgaben bzw. Eigenschaften, die es zu einem unverzichtbaren Hilfsmittel innerhalb agiler Methoden machen:

- **Transparenz nach außen:** Eine wichtige Eigenschaft des physikalischen Task-Boards ist, dass es für jedermann zugänglich und leicht verständlich ist. Außenstehende Personen (z.B. Manager, Auftraggeber, etc.), die den Raum betreten können sich innerhalb kurzer Zeit einen Überblick des aktuellen Arbeitsfortschritts verschaffen.
- **Transparenz nach innen:** Statusänderungen am physikalischen Task-Board werden im Regelfall von allen anwesenden Personen sofort wahrgenommen, da sie nicht vom Arbeitsrechner aus durchgeführt werden können. Das sorgt für Transparenz während der täglichen Arbeit. Es ist sofort ersichtlich, wenn z.B. einzelne Arbeitspakete begonnen oder abgeschlossen werden.
- **Zentraler Treffpunkt:** Das Task-Board dient als zentraler Treffpunkt bei Besprechungen. So findet z.B. das Daily-Scrum vor dem Task-Board statt, um den Projektzustand diskutieren und aktualisieren zu können.
- **Erhöhung der Motivation:** Das Task-Board dient als Motivationsquelle der Teammitglieder. Erfolgreich abgeschlossene Aufgaben sowie der Projektfortschritt werden durch die Darstellung am Task-Board vom gesamten Team wahrgenommen. Das wirkt häufig stark motivierend auf das gesamte Team. Die motivierende Wirkung physikalischer Task-Boards wurde bereits durch Sharp et al. [78] und Perry [63] untersucht und bestätigt.

3.1.4 Digitale Projektmanagement-Tools zur Unterstützung agiler Softwareentwicklung

Die Studie State of agile aus dem Jahr 2020 [58] zeigt deutlich, dass der Einsatz von PMS mittlerweile stark verbreitet ist. Insgesamt gaben 65% der knapp 1200 befragten Personen an, dass sie PMS in agilen Softwareprojekten einsetzen. Zusätzliche 13% planen den zukünftigen Einsatz von PMS.

Ein mögliche Erklärung für den stark verbreiteten Einsatz ist die Zunahme von verteilt durchgeführten Softwareprojekten und die daraus resultierende Notwendigkeit dezentral zugreifbarer Projektdaten (vgl. auch Abs. 1.1.2). Physikalische Medien (wie z.B. das Task-Board) als Bestandteil agiler Arbeitsumgebungen können die Forderung nach dezentraler Verfügbarkeit nicht erfüllen, weshalb auf PMS zurückgegriffen wird.

Im State of Agile 2020 [58] wurden beispielsweise Jira Agile [40], Microsoft Azure DevOps [4] sowie Agile Management von Version One [18] als häufig genutzte Vertreter agiler PMS ermittelt. Jira Agile wird von 67% der befragten Personen am häufigsten eingesetzt, weshalb es im Rahmen dieser Arbeit stellvertretend für agile PMS als Basis für weitere Untersuchungen herangezogen wird.

3.1.5 Neuartige Ansätze zur Verbesserung agiler PMS

Zusätzlich zu den in 3.1.4 genannten Beispielen kommerzieller PMS gibt es eine Reihe von Forschungsansätzen, die sich ebenfalls -wie auch diese Arbeit- mit den Nachteilen und entsprechenden Optimierungspotentialen von PMS im Rahmen agiler Softwareentwicklung beschäftigen. In den nachfolgenden Absätzen werden ausgewählte Arbeiten vorgestellt, denen ähnliche Forschungsziele wie dieser Arbeit zugrunde liegen.

Die erste interessante Forschungsarbeit von Kropp et al. [48] wurde 2017 veröffentlicht. Es wurde eine qualitative Interviewstudie durchgeführt, um zu verstehen weshalb, in der Praxis häufig PMS und physikalische Task-Boards parallel eingesetzt werden. Die am häufigsten genannten Gründe sind Größe und Übersichtlichkeit eines physikalischen Task-Boards, leichte Zugänglichkeit, ständige Verfügbarkeit und volle Flexibilität bzgl. der darstellbaren Inhalte. Aus diesen Ergebnissen wurden Anforderungen an ein neues Lösungskonzept definiert. Anschließend wurde ein Konzept auf Basis großer, multitouch-fähiger Displays erarbeitet und prototypisch umgesetzt. Im Wesentlichen handelt es sich dabei um ein neues User-Interface für die PMS Jira Agile. Durch diese Lösungsidee sollen die wichtigsten Nachteile konventioneller PMS (e.g. komplizierte Benutzung, eingeschränkte Übersichtlichkeit, fehlende MUI, eingeschränkte Flexibilität bei der Darstellung von Inhalten) verbessert werden. Mit Hilfe des entwickelten Prototyps wurde anschließend eine Benutzerstudie zur Evaluation des Lösungskonzepts durchgeführt. Hierfür wurden insgesamt 11 Probanden mit Vorerfahrungen in der agilen Softwareentwicklung ausgewählt, die typische Aufgaben und Besprechungen eines agilen Prozesses mit Hilfe des Prototypen durchführen mussten und anschließend befragt wurden. Insgesamt ließen sich Verbesserungen in nahezu allen untersuchten Bereichen feststellen. Es wurden jedoch auch

Einschränkungen bzgl. Lesbarkeit, Flexibilität bzgl. darstellbarer Informationen gefunden.

Die zweite Arbeit aus dem Jahr 2014 von Rubart [72] hat sich die automatisierte Erhebung unterschiedlicher *Key-Performance-Indicators* (KPIs) innerhalb agiler Entwicklungsprojekte als Ziel gesetzt. Mit Hilfe dieser Kennzahlen sollen Analysen zur Prozessverbesserung ermöglicht werden, um z.B. Aufwandsabschätzungen und Planungsaufgaben zu erleichtern. Die automatisierte Erhebung derartiger Kennzahlen setzt den Einsatz von PMS voraus. Ein weiteres Ziel Rubarts war es, trotz Erhebung der Kennzahlen die Teamarbeit bestmöglich zu unterstützen. Aus diesem Grund wurde nicht auf konventionelle PMS auf Basis browserbasierter Technologien zurückgegriffen. Stattdessen kam ein multitouch-basiertes Lösungskonzept zum Einsatz. Ähnlich wie bei Kropp et al. [48] wurde eine Task-Board Applikation entwickelt, die auf einem Multitouch-Tisch bedient werden kann. Dabei wurden die wichtigsten Anwendungsfälle wie z.B. das Release-Planning und die Entwicklungsphase (vgl. auch Inside Iteration in Abb. 3.1) berücksichtigt. Im Rahmen einer Benutzerstudie wurde eine erste Evaluation durchgeführt. Dabei stand jedoch in erster Linie die Gebrauchstauglichkeit des User-Interfaces bzgl. einzelner Anwendungsfälle im Fokus. Aspekte der Gruppenarbeit wie z.B. Kommunikation, Motivation und Transparenz wurden nicht bewertet.

Ein weitere Arbeit von Rubart [73] aus dem Jahr 2015 beschreibt eine Lösungsidee zur Integration von physikalischen (papierbasierten) Karteikarten in das eben beschriebene Task-Board Konzept. Hierbei sollten die Karteikarten mittels Kamera erfasst und entsprechend digital dargestellt werden. Die Schwierigkeiten dieser Lösung sind laut Rubart in der technischen Realisierung (wie z.B. der Handschrifterkennung) zu erwarten. Leider gibt es bis heute (2021) kein technischen Lösungsvorschläge noch weitere Veröffentlichungen zu diesem Ansatz.

Eine von Raith et al. [70] durchgeführte Forschungsarbeit aus dem Jahr 2013 verfolgte einen ähnlichen Ansatz wie Rubart. Hierbei wurde ebenfalls die automatisierte Generierung von Kennzahlen in agilen Entwicklungsprozessen betrachtet. Wie auch bei Rubart musste hierfür PMS eingesetzt werden. Um die Teamarbeit möglichst gut zu unterstützen, setzte Raith et al. dabei ebenfalls auf ein multitouch-basiertes Lösungskonzept. Als Anwendungsbeispiel wurde das sog. Planning Poker nach Cohn [16] ausgewählt. Dabei handelt es sich um eine weit verbreitete Methode zur Aufwandsschätzung in agilen Projekten, die in gemeinschaftlicher Teamarbeit durchgeführt wird. Jeder Teilnehmer schätzt den Aufwand einer User-Story (vgl. 3.1.2.2) mit Hilfe von Spielkarten. Als Maßeinheit für den Aufwand kommen sog. Story-Points zum Einsatz. Es handelt sich dabei um eine relative Einheit, die den Aufwand einzelner User-Stories in Relation zueinander beschreibt (z.B.: Eine User-Story, die mit 3 Story-Points bewertet wurde ist dreimal so aufwändig wie eine User-Story, die mit nur einem Story-Point bewertet wurde.) Das dabei entwickelte Lösungskonzept basiert auf einer Multitouch-Anwendung, die den gesamten Planning-Poker-Prozess abbildet. Dabei kam ein Multitouch-Tisch, ein Tablet, sowie ein Smartphone für jeden Teilnehmer als Ein- bzw. Ausgabegerät zum Einsatz. Eine erster Proof of Concept wurde im Rahmen einer Benutzerstudie mit insgesamt 21 Studenten durchgeführt. Beim Untersuchungsplan handelte es sich um ein Qualitatives

Experiment. Dabei setzte eine Gruppe einen Prototypen entsprechend dem entwickelten Lösungskonzept ein. Eine Kontrollgruppe führte die Aufwandsabschätzungen mit Hilfe physikalischer Medien (Story-Cards und Planning-Pokerkarten) durch. Es konnte gezeigt werden, dass die automatisierte Erfassung von Kennzahlen grundsätzlich möglich ist und sich Multitouch-Geräte zur Durchführung von Gruppenarbeit eignen. Es ließen sich gegenüber der physikalischen Variante jedoch auch Nachteile feststellen. Ein negativer Aspekt war die fehlende Flexibilität bei der Bearbeitung der jeweiligen Anforderungen. Spontane Notizen oder Anmerkungen konnten beispielsweise nicht ergänzt werden. Auch die Anpassung des Sprint- bzw. Product-Backlogs war im Rahmen der prototypischen Umsetzung nicht möglich. Als wesentlich störender wurde jedoch die eingeschränkte Mobilität empfunden. Ein spontaner Ortswechsel war beispielsweise nicht möglich. Das gesamte Team war an den Standort des Multitouch-Tisches gebunden. Aufgrund der recht langen Studiendauer wurde das als sehr störend wahrgenommen.

Eine weitere Lösung bietet Esbensen et al. [27] in einer 2015 veröffentlichten Arbeit an. Im Rahmen einer Beobachtungsstudie zweier verteilt arbeitender Teams stellte er fest, dass die eingeschränkte Verfügbarkeit adäquater Kommunikations- und Projektmanagementlösungen die Hauptursachen für Missverständnisse und Fehlplanungen in verteilten Entwicklungsprojekten sind. Als Lösung bietet er ein Konzept zur Kombination von Technologien zur Durchführung von Videokonferenzen und konventioneller PMS an. Dabei kommt ein großes Touch-Display zum Einsatz, das ein Videobild des entfernten Entwicklungsteams wiedergibt und damit als eine Art Fenster zum Offshore-Team dient. Gleichzeitig werden darauf die Inhalte eines gemeinsamen Task-Boards angezeigt. Mittels einfacher Gestensteuerung kann die Position und damit der Status der auf dem Task-Board dargestellten User-Stories verändert werden. Eine erste Evaluation des Konzepts wurde mit Hilfe eines Prototypen im Rahmen eines Evaluationsworkshops durchgeführt. Dabei wurden zwei typische Szenarien der agilen Softwareentwicklung betrachtet (1. Updaten des Task-Boards nach Erledigung einer Aufgabe, 2. Durchführung eines Daily-Scrum-Meetings). Anschließend wurde eine Auswertung mittels Technology Acceptance Model [21] durchgeführt, die durchaus positive Ergebnisse erzielte. Eine Evaluation bzgl. der ursprünglichen Probleme in den Bereichen Kommunikation und Gruppenarbeit wurde seither jedoch nicht veröffentlicht.

3.2 Forschungsfeld Computer Supported Cooperative Work (CSCW)

Um das Kernproblem dieser Arbeit (vgl. 1.2) zu lösen, bietet das Forschungsfeld CSCW vielversprechende Ansätze. Der Fokus dieses Forschungsfeldes liegt auf der Kombination und dem zielorientierten Einsatz unterschiedlichster IT- und Kommunikationstechnologie, um Entscheidungs- und Arbeitsprozesse zu unterstützen [83]. Hierbei liegen moderne Technologien (z.B. Augmented Reality, Virtual Reality) im Fokus der Forschung, da sie weitreichende Möglichkeiten zur Umsetzung natürlicher Benutzerschnittstellen be-

reithalten. Hasenkamp et al. [33] sieht die folgenden drei Teilbereiche im Zentrum der CSCW-Forschung:

- Entwicklung eines Verständnisses von Zusammenarbeit und Koordination innerhalb arbeitsteiliger Prozesse
- Entwicklung von Konzepten und Werkzeugen für die Unterstützung arbeitsteiliger Prozesse
- Bewertung / Evaluation dieser Konzepte und Werkzeuge

Die drei genannten Punkte verdeutlichen noch einmal die Zuordnung dieser Arbeit zum Forschungsfeld CSCW und die Eignung des DSRP als grundlegenden Forschungsansatz.

3.2.1 Klassifikation von CSCW Anwendungen und Einordnung der Arbeit

Um die zahlreichen Anwendungsfelder der CSCW Forschung zu strukturieren, wird am häufigsten die Klassifikationsmatrix nach Johansen [41] verwendet. Er unterscheidet CSCW-Anwendungen nach den beiden Dimensionen Zeit und Ort. Abb. 3.6 zeigt die rudimentäre Aufteilung beider Dimensionen von Johansen in gleiche Zeit (synchron) / ungleiche Zeit (asynchron) bzw. gleicher Ort / ungleicher Ort. Demnach ergeben sich vier Klassen zur Zuordnung unterschiedlichster CSCW-Anwendungen:

Face-to-Face interactions: Alle Mitglieder einer Gruppe befinden sich zur gleichen Zeit am gleichen Ort und können direkt miteinander kommunizieren. Sie kann z.B. durch den Einsatz von Präsentationssoftware oder Entscheidungsunterstützungssoftware profitieren. Hierbei sind Systeme mit MUI Unterstützung zu bevorzugen (z.B. Multitouch-Tische), um möglichst alle Gruppenmitglieder aktiv in die Gruppenarbeit einzubinden.

Continuous Tasks: CSCW Anwendungen dieser Kategorie werden ebenfalls von lokal arbeitenden Gruppenmitglieder eingesetzt. Jedoch steht hier nicht der unmittelbare Einsatz in einer konkreten Besprechung im Fokus, sondern der kontinuierliche Einsatz über einen längeren Zeitraum. Typische Beispiele sind Gruppenkalender und Projektmanagementsysteme.

Remote interactions: Diese Anwendungen werden von örtlich verteilten Gruppenmitgliedern eingesetzt, die zur gleichen Zeit Arbeiten. Typische Beispiele sind Videokonferenzsysteme, Instant-Messenger und Anwendungen zum Screen-Sharing.

Communication and Coordination: Zu dieser Kategorie zählen CSCW Anwendungen, die zu unterschiedlichen Zeiten und von verschiedenen Orten aus genutzt werden. Klassische Anwendungen dieser Kategorie sind Emailsysteme, Wissensdatenbanken (z.B. Wikis), Versionsverwaltungen und konventionelle PMS (z.B. Jira Agile).

In der Praxis eingesetzte CSCW Anwendungen weisen häufig ein breites Funktionsspektrum auf und müssen somit mehreren Kategorien zugeordnet werden. Das in dieser Arbeit betrachtete Referenzsystem für PMS (Jira Agile [40]) zur Unterstützung agiler

Projektarbeit ist in erster Linie der Kategorie "Communication and Coordination" zuzuweisen, da es eine browserbasierte Benutzeroberfläche aufweist, die unabhängig von Ort und Zeit bedient werden kann. Diese Eigenschaft ist notwendig, um den Einsatz von PMS innerhalb von Projekten mit örtlich und zeitlich verteilt arbeitenden Teams zu ermöglichen. Gleichzeitig muss jedoch auch die Perspektive eines einzelnen Teams vor Ort betrachtet werden. Zahlreiche Arbeitssituationen (z.B. Besprechungen) erfordern den Einsatz der PMS in typischen Face-to-Face Situationen (vgl. Fig. 3.6). Diese werden durch die Benutzerschnittstellen konventioneller PMS aus unserer Sicht nicht optimal unterstützt. Eine Verbesserung in diesem Bereich steht somit im Fokus dieser Arbeit (vgl. 1.3).

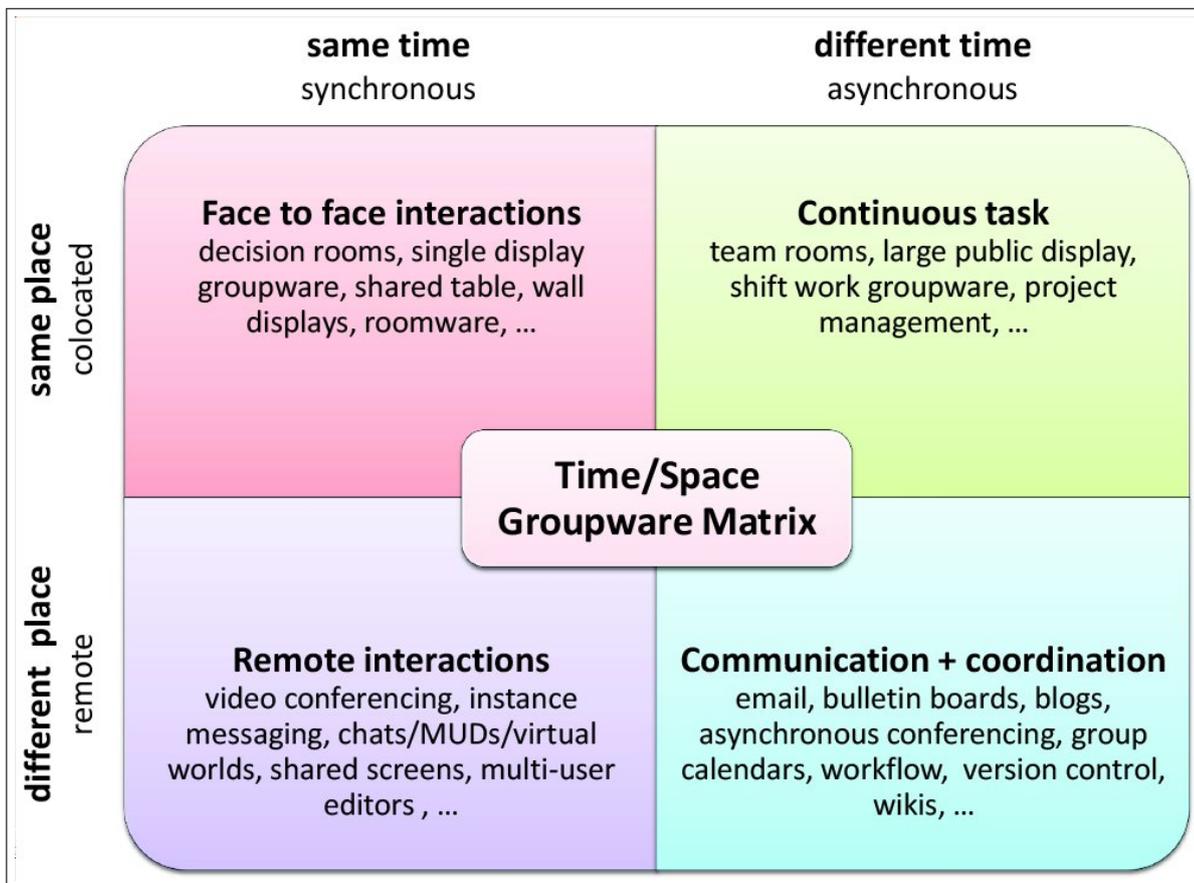


Abbildung 3.6: CSCW-Klassifizierung nach Johansen [41].

(Quelle: <https://upload.wikimedia.org/wikipedia/commons/2/28/Cscwmatrix.jpg>)

3.2.2 Eigenschaften Natürlicher Benutzerschnittstellen

Dieser Abschnitt geht näher auf die Eigenschaften von TUIs als Vertreter natürlicher Benutzerschnittstellen ein. Ziel dieses Abschnitts ist es, die zugrundeliegende Idee hinter dem Konzept *Tangible-User-Interface* (TUI) klar darzustellen.

Eine natürliche Fähigkeit des Menschen ist die Wahrnehmung und Veränderung seiner

physikalischen Umgebung. Diese Art der Interaktion erfolgt häufig dementsprechend intuitiv und mühelos. Diesen Vorteil können *Graphical User Interfaces* (GUIs) nur eingeschränkt nutzen, indem Metaphern eingesetzt und am Bildschirm dargestellt werden. Ein bekanntes Beispiel sind Ordner, die zur Ablage und Strukturierung von Dokumenten eingesetzt werden können. Ein bewährtes Konzept, das der Anwender bereits aus seiner physikalischen Welt kennt und somit intuitiv versteht. Die Einschränkung von GUIs entsteht durch die Bedienung dieser Elemente durch Maus, Tastatur und Bildschirm. Sie verhindern eine intuitive Manipulation der dargestellten Daten und sind der Grund, dass GUIs nicht zu den natürlichen Benutzerschnittstellen zählen [39]. Einen Schritt weiter sind Benutzerschnittstellen auf Basis multitouch-fähiger Bildschirme. Sie nutzen das gleiche Prinzip der metaphorischen Darstellung von Daten und erlauben gleichzeitig eine direktere Interaktion durch den Benutzer mittels Fingergesten. Trotzdem bleibt der Bildschirm als Barriere, weshalb eine natürliche Interaktion mit Objekten nicht möglich ist. Das Konzept der TUIs versucht diese letzte Barriere zu umgehen, um digitalen Inhalten eine physikalische Form zu geben und damit die Möglichkeit zur direkten Interaktion anzubieten. Laut Ishii [39] werden Entscheidungsprozesse, Gruppenarbeit und Lernaufgaben besser unterstützt, weil der Endanwender seine über die Evolution erlernten Fähigkeiten zur Interaktion mit seiner physikalischen Umgebung nutzen kann.

3.3 Forschungsgegenstand

Dieser Abschnitt fasst noch einmal kurz die wichtigsten Aspekte der eben dargelegten Informationen zusammen und konkretisiert das Grundproblem, der schwierigen Vereinbarkeit verteilter Softwareentwicklung mit agilen Grundprinzipien. Anschließend werden die wichtigsten Beiträge zur Wissenschaft dieser Arbeit kurz vorgestellt.

3.4 Konkretisierung des Forschungsproblems

Neben der Einführung in agile Methoden und Techniken hat dieses Kapitel die große Bedeutung physikalischer Medien für Gruppenarbeitsprozesse in der agilen Softwareentwicklung aufgezeigt (vgl. 3.1.3.1). Gleichzeitig zeigen Erhebungen wie z.B. "State of agile" [58] deutlich, dass die große Mehrheit agiler Projekte nicht mehr ohne den Einsatz digitaler PMS auskommt (vgl. 3.1.4). Grund für den häufigen Einsatz von PMS ist in erster Linie die Notwendigkeit ortsunabhängiger Informationsbereitstellung in örtlich verteilten Projekten (vgl. 3.1.4). Konventionelle PMS nutzt nahezu ausschließlich browserbasierte Benutzeroberflächen zur Interaktion. Der Einsatz dieser Werkzeuge steht jedoch dem OWP (vgl. 3.1.3.1) und damit auch den Forderungen des Agilen Manifests [9] nach offener, transparenter Kommunikation entgegen. Nachteilige Effekte auf Kommunikation, Transparenz und Kommunikation innerhalb agiler Entwicklungsprojekte wurden bereits in Studien belegt (vgl. 3.1.3.1). Dieses offensichtliche Problem ist Kern dieser Arbeit. Das Forschungsfeld CSCW bietet vor allem durch den Einsatz von TUI das Potenzial, eine Brücke zwischen der Forderung nach ortsunabhängiger Verfügbarkeit projektrelevanter

Daten und den Forderungen des agilen Manifests nach Arbeitsumgebungen auf Basis physikalischer Medien zu schlagen. Diese Thematik ist Kernbestandteil der Arbeit.

3.4.1 Beitrag zur Wissenschaft

Im Wesentlichen lassen sich die Beiträge zu Forschung und Wissenschaft auf Basis der in 1.3 formulierten Forschungsziele in folgende Bereiche unterteilen:

1. **Ergebnisse zu Z1.1:** Im Rahmen der Vorstudie wurden Arbeitsabläufe innerhalb agiler Projekte mit örtlich verteilt arbeitenden Projektteams untersucht. Die Ergebnisse beinhalten Ablaufdiagramme (vgl. A.3), die den Einsatz von Artefakten und unterstützenden Medien aufzeigen. Zusätzlich wurden detaillierte Beschreibungen auf Basis der einzelnen Prozessschritte und Besprechungen der jeweils eingesetzten agilen Verfahren erarbeitet. Eine ausführliche Beschreibung wurde bereits im Rahmen der Arbeit "How Project-management-tools are used in Agile Practice: Benefits, Drawbacks and Potentials" [69] publiziert.
2. **Ergebnisse zu Z1.2:** Weiterhin wurden im Rahmen der Vorstudie Vorteile, Nachteile und Verbesserungspotenziale konventioneller PMS aus Expertensicht erarbeitet (vgl. Tab. 4.3 und Raith et al. [69]). Diese Auflistung kann als Grundlage für weitere Forschungsarbeiten in diesem Bereich herangezogen werden.
3. **Ergebnisse zu Z1.3:** Konkrete Verbesserungsmöglichkeiten und darauf basierende Schlüsselanforderungen an das Lösungskonzept werden in 4.2.6 auf Basis der in der Vorstudie erarbeiteten Ergebnisse aufgezeigt.
4. **Ergebnisse zu Z2 und Z3:** Es wurde ein Lösungskonzept auf Basis von TUI erarbeitet, das die Vorteile physikalischer Medien mit den Vorteilen konventioneller PMS vereint. Im Gegensatz zu den in 3.1.3.1 und 3.1.5 erläuterten Forschungsarbeiten setzt das erarbeitete Konzept nicht auf den Einsatz multitouch-basierter Benutzerschnittstellen, sondern nahezu ausschließlich auf den Einsatz physikalischer Medien zur nahtlosen Integration einer PMS in agile Entwicklungsprojekte. Dieser Ansatz wurde bisher wissenschaftlich nicht behandelt und ein konkretes Lösungskonzept unter Berücksichtigung der technischen Machbarkeit fehlt. Im Rahmen der Evaluationsstudie konnten Verbesserungen im Vergleich zu konventioneller PMS bzgl. Kommunikation, Transparenz und Motivation innerhalb agiler Projektteams festgestellt werden. Zusätzlich konnte eine Verbesserung der Gebrauchstauglichkeit nach DIN EN ISO 9241 [32] festgestellt werden.

Kapitel 4

Vorstudie - Einsatz digitaler Projektmanagement-Software in der Praxis

Dieses Kapitel beschreibt Durchführung und Ergebnisse der Vorstudie. Ziel der Studie war es, ein besseres Verständnis bzgl. des PMS Einsatzes in verteilten, agilen Softwareprojekten zu erreichen. Die Vorstudie lässt sich aus Sicht des DSRP der Zieldefinition (Schritt 2) zuordnen, weil die gewonnen Ergebnisse die Basis der Anforderungsdefinition (vgl. 5.2.1 und 5.2.2) darstellen.

Eine detaillierte Beschreibung der Studie wurde bereits in der Arbeit "How Project-management-tools are used in Agile Practice: Benefits, Drawbacks and Potentials" von Raith et al. [69] veröffentlicht.

4.1 Ziele der Studie

Das übergeordnete Forschungsziel dieser Arbeit ist die Optimierung konventioneller PMS zur Verbesserung der Gruppenarbeit in verteilten, agilen Projekten (vgl. Z in 1.3). Im Rahmen der Vorstudie soll nun erforscht werden, wie konventionelle PMS in der Praxis eingesetzt wird (vgl. Z1.1 in 1.3) und welche Aspekte als Vorteile bzw. Nachteile aus Sicht der Anwender beurteilt werden. Ein weiterer Aspekt ist die Untersuchung der Arbeitsabläufe der einzelnen Projektteams mit Fokus auf dem Einsatz der PMS (vgl. Z1.2 in 1.3). Die Ergebnisse beider Zwischenziele dienen als Basis, um konkrete Verbesserungspotenziale hinsichtlich konventioneller PMS zu definieren (vgl. Z1.3 in 1.3), die wiederum in die Anforderungsdefinition des Lösungskonzepts einfließen werden (vgl. Z2 in 1.3).

4.2 Forschungsplan - Einzelfallanalyse

In 2.2.1 wurde bereits die Einzelfallanalyse als zentraler Untersuchungsplan der Vorstudie festgelegt und beschrieben. Die einzelnen Schritte werden hier nochmals aufgegriffen und anhand konkreter Ergebnisse und Daten dargestellt.

4.2.1 Fragestellung

Um die definierten Studienziele zu erreichen, wurden zunächst folgende Forschungsfragen allgemein formuliert:

- **F1:** Wie wird PMS eingesetzt, um verteilte, agile Softwareentwicklungsprojekte bestmöglich zu unterstützen?
- **F2:** Welche Eigenschaften und Funktionen von PMS werden im Rahmen verteilter, agiler Softwareentwicklungsprojekte als Vorteil, Nachteil bzw. Verbesserungspotenzial gesehen und was ist der Grund dafür?

Grundlage für die Beantwortung von F2 ist ein klares Verständnis, welche Eigenschaften und Funktionen als Vorteil, Nachteil oder Verbesserungspotenzial im Sinne der Fragestellung verstanden werden. Anbei die Definition:

Vorteil im Sinne von F2: Ein Vorteil ist eine Eigenschaft oder ein Effekt, der sich positiv auf die Erreichung der Projektziele auswirkt und auf die Nutzung von PMS anstelle physikalischer Medien zurückzuführen ist.

Nachteile im Sinne von F2: Ein Nachteil ist eine Eigenschaft oder ein Effekt, der sich negativ auf die Erreichung der Projektziele auswirkt und auf die Nutzung von PMS anstelle physikalischer Medien zurückzuführen ist.

Verbesserungspotenzial im Sinne von F2: Unter Verbesserungspotenzialen werden im Rahmen der Interviewstudie Ideen zur Verbesserung konventioneller PMS verstanden, die sich aus Sicht des Probanden positiv auf die Erreichung der Projektziele auswirken.

Um die Fragestellungen zu konkretisieren wurde im Rahmen des durchgeführten problemzentrierten Interviews ein Leitfragenkatalog ausgearbeitet (vgl. 4.2.3).

4.2.2 Falldefinition

Die Falldefinition (vgl. 2.2.1) ist entscheidend für die spätere Validität der Forschungsergebnisse. Hier wird definiert welche Fälle und damit auch welche Personen als Studienteilnehmer in Frage kommen. Besonders wichtig für aussagekräftige Ergebnisse war hierbei eine entsprechende Praxiserfahrung im Rahmen agiler Softwareentwicklungsprozesse sowie die Aktualität der Erfahrungswerte. Aus diesen Gründen wurden folgende Kriterien an die Studienteilnehmer bzw. deren Projekte definiert:

Anforderungen an den Probanden:

- Mind. 3-jährige Erfahrung in agilen Softwareentwicklungsprojekten
- Beteiligung an mindestens einem agilen Softwareentwicklungsprojekt zum Zeitpunkt des Interviews
- Mind. 1-jährige Erfahrung in örtlich verteilten Softwareentwicklungsprojekten

Anforderungen an das aktuelle Projekt des Probanden:

- Agiles Softwareentwicklungsprojekt nach Scrum
- Einsatz verteilter, agiler Teams und/oder verteilter Teammitglieder
- Einsatz einer konventionellen PMS (z.B. Jira Agile)

Bzgl. der Projektbeschaffenheit wurde darauf geachtet, möglichst unterschiedliche Projektgrößen (Anzahl der beteiligten agilen Teams) abzudecken. Tabelle 4.1 gibt einen entsprechenden Überblick. Alter, Geschlecht und Projekterfahrung der Studienteilnehmer zeigt Tabelle 4.2. Die Anwendung des problemzentrierten Interviews (vgl. 2.2.2) im Rahmen einer Einzelfallanalyse erlaubt es uns, die Anzahl der Probanden auf eine geringe Menge (in unserem Fall fünf Probanden mit je einem aktuellen Projekt) zu beschränken und gleichzeitig den Durchführungsaufwand in einem machbaren Rahmen zu halten. Ziel der Einzelfallanalyse ist gerade die detaillierte Untersuchung einiger weniger Fälle (vgl. 2.2.1).

Tabelle 4.1: Struktur der untersuchten Projekte

Projekt	Agile Teams	Standorte
P1	2 Scrum-Teams	Köln, Berlin
P2	3 Scrum-Teams	München, Mumbai
P3	10 Scrum-Teams	München, Nürnberg, Bangkok
P4	14 Scrum-Teams, 4 Kanban-Teams	München (innerhalb eines Gebäudekomplexes)
P5	1 Scrum-Team + Stakeholder	Berlin (Stakeholder verteilt in Deutschland)

Tabelle 4.2: Demographische Daten und Projekterfahrung der Studienteilnehmer

Projekt	Angaben zur Person	Erfahrung in der agilen Softwareentwicklung	Rollen in agilen Projekten
P1	männlich, 34	5 Jahre	Entwickler Scrum-Master Agile-Coach
P2	weiblich, 36	6,5 Jahre	Product-Owner Scrum-Master Agile-Coach
P3	männlich, 43	8 Jahre	Product-Owner Agile-Coach
P4	männlich, 47	10 Jahre	Product-Owner Agile-Coach
P5	weiblich, 41	12 Jahre	Scrum-Master Agile-Coach

4.2.3 Methodenauswahl

Die Entscheidung für das Problemzentrierte Interview wurde bereits in 2.2.2 ausführlich erläutert und begründet. Basis des Interviews ist ein Leitfragenkatalog, der entsprechend der Forschungsfragen ausgearbeitet wurde. Im Folgenden werden die einzelnen Fragekategorien sowie deren Zweck kurz erläutert. Der vollständige Fragenkatalog befindet sich zudem im Anhang unter A.2. Im Folgenden werden die drei im Interview eingesetzten Fragekategorien kurz erläutert:

1. **Generelle Fragen zum Probanden und seinem aktuellen Softwareentwicklungsprojekt:** Diese Kategorie hatte das Ziel, die Vorerfahrung des Probanden mit Softwareentwicklungsprojekten kennenzulernen. Besonders die einzelnen Rollen des Probanden im Projekt waren hierbei wichtig. Außerdem wurde der Projektaufbau (Anzahl beteiligter Teams, regionale Verteilung der Teams und Teammitglieder, Art der Software) genauer hinterfragt.
2. **Fragen zum Entwicklungsprozess mit Fokus auf Medieneinsatz und Artefakten:** Diese Fragekategorie hatte das Ziel, den Entwicklungsprozess im aktuellen Projekt des Probanden genau zu verstehen. Dabei wurden alle Prozessschritte genau beleuchtet und insbesondere der Einsatz von PMS sowie Artefakten (z.B. Dokumente, Anforderungsbeschreibungen, Task-Boards etc.) betrachtet.
3. **Detailfragen zum Einsatz von PMS, Medien und Entwicklungswerkzeugen:** Diese Fragekategorie zielte auf weitere Details zur Arbeit mit Anforderungen, spezifischen Funktionen der eingesetzten PMS und agilen Arbeitsumgebungen ab.

Während des kompletten Interviews war neben sachlichen Schilderungen auch die persönliche Meinung des Probanden willkommen, um Vor- und Nachteile der jeweils eingesetzten Prozesse und Werkzeuge aus dessen persönlicher Sicht kennenzulernen.

4.2.4 Datenaufbereitung

Die Transkription der aufgezeichneten Interviews wurde auf Basis des Transkriptionsregelwerks nach Kuckartz et al. [49] (vgl. 2.2.2) durchgeführt. Anschließend wurden die Transkripte wie in 2.2.2 beschrieben mit Hilfe der Qualitativen Inhaltsanalyse ausgewertet. Im Fokus standen dabei die beiden in 4.2.1 formulierten Fragestellungen. Um den Einsatz von PMS in Kombination mit physikalischen Medien in den jeweiligen Projekten besser beurteilen zu können, wurden zunächst Workflow-Diagramme zu jedem vorliegenden Transkript angefertigt (vgl. Beispiel in Abb. A.3). Ein vollständige Auflistung der Diagramme befindet sich im Anhang unter A.3. Ziel dieser Diagramme war eine übersichtliche Darstellung der eingesetzten physikalischen und digitalen Medien im jeweiligen Entwicklungsprozess, um einen Vergleich der untersuchten Projekte zu erleichtern. Abb. A.3 zeigt beispielhaft das Workflow-Diagramm zu Projekt P3. Auf der linken Seite (blau markiert) befinden sich die einzelnen Prozessschritte und Besprechungen. Das Ablaufdiagramm rechts zeigt den Weg der Softwareanforderungen im Prozess auf. Jeder Schritt enthält dabei eine Information zum eingesetzten Medium. Blau markierte Prozessschritte

entsprechend dabei physikalischen (papierbasierten) Medien. Orange dargestellte Prozessschritte wurden auf Basis digitaler Medien (wie z.B. PMS) durchgeführt. Mit Hilfe dieser Darstellungen konnten Medienbrüche wie z.B. eine manuelle Transformation relevanter Daten von physikalischen zu digitalen Medien (oder auch umgekehrt) leicht ermittelt werden. Im dargestellten Beispiel (Abb. A.3) ist ein Medienbruch beim Übergang der Anforderungen vom Product-Backlog zum Sprint-Backlog zu erkennen. Zusätzlich werden durch diese Art der Darstellung Redundanzen der Projektdaten zwischen physikalischen und digitalen Medien aufgedeckt. Im Beispiel (Abb. A.3) werden Tasks redundant in Jira Agile und an einem physikalischen Task-Board dargestellt.

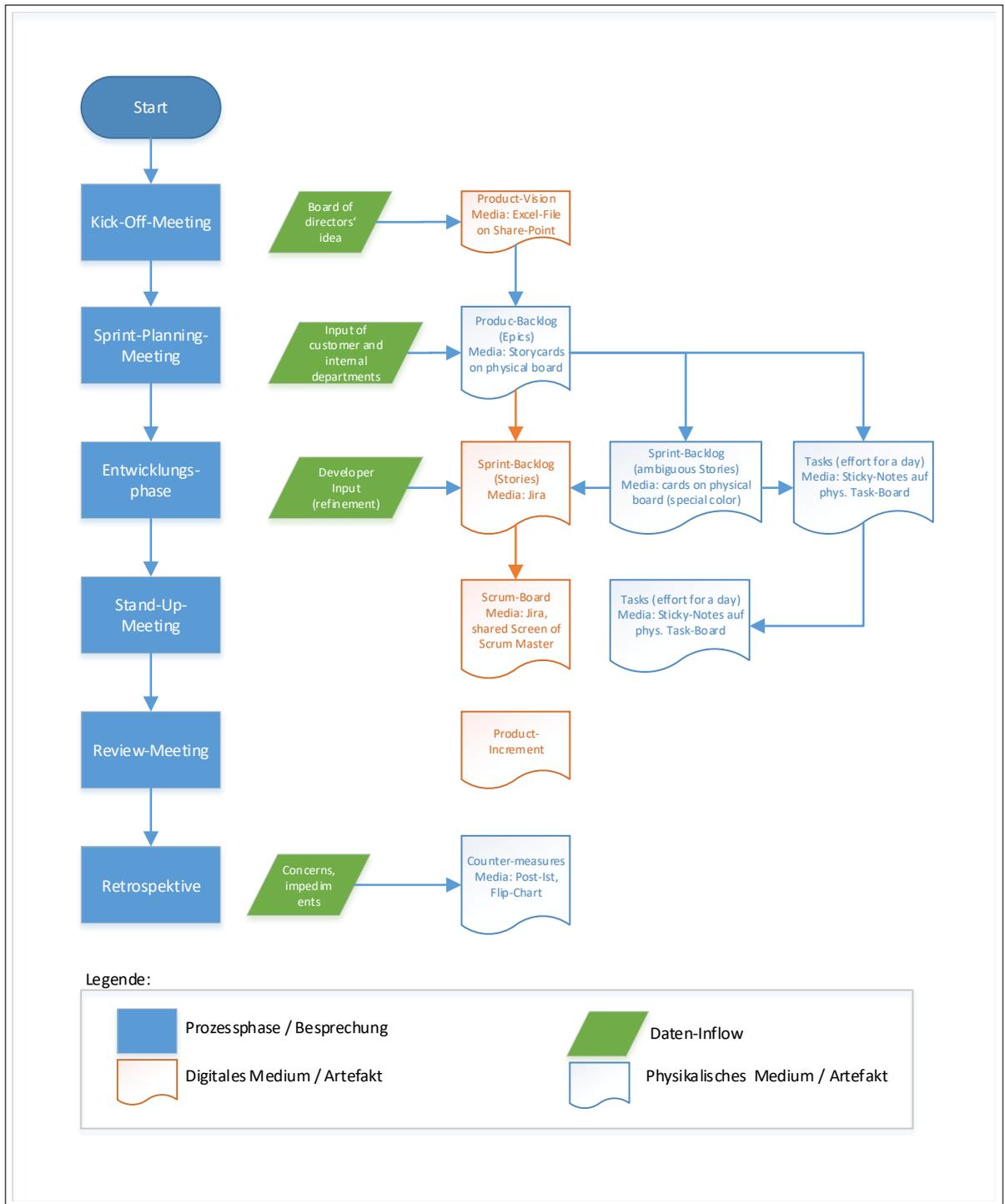


Abbildung 4.1: Beispieldiagramm zum Einsatz digitaler und physischer Medien im Entwicklungsprozess von Projekt P3

Bzgl. Forschungsfrage F2 wurde nach dem in 2.2.3 beschriebenen Verfahren der induktiven Kategorisierung vorgegangen. Auf eine weitere Darstellung der Datenaufbereitung kann deshalb an dieser Stelle verzichtet werden.

4.2.5 Einordnung in den Zusammenhang - Ergebnisse

In diesem Absatz werden die Erkenntnisse der Vorstudie zusammengetragen. Die hier dargestellten Ergebnisse wurden ebenfalls in der entsprechenden Arbeit von Raith et al. [69] veröffentlicht.

Ergebnisse zu F1: Wie wird PMS eingesetzt, um verteilte, agile Softwareentwicklungsprojekte bestmöglich zu unterstützen?

Konventionelle PMS wird in allen untersuchten Projekten verwendet, um den aktuellen Projektfortschritt (Gesamtstatus sowie Status innerhalb der Entwicklungsphasen) im verteilten Projekt zu kommunizieren. Zusätzlich ergänzen 4 von 5 untersuchten Projekten die eingesetzte PMS mit physikalischen Medien um Informationsfluss, Transparenz und Kommunikation in Gruppenarbeitsprozessen zu verbessern. Die grundlegenden Schwierigkeiten sind in allen untersuchten Projekten ähnlich: Redundanzen, Inkonsistenzen und Mehraufwand lassen sich nicht vermeiden, sobald physikalische Medien und PMS parallel eingesetzt werden. Folgende Erkenntnisse müssen als Grundlage für ein mögliches Lösungskonzept gelten und sollten die Basis für jede konzeptionelle Entscheidung sein:

1. Brüche zwischen physikalischen und digitalen Medien sollten auf ein Minimum reduziert werden.
2. Physikalische Medien sollten gerade in Gruppenprozessen bevorzugt eingesetzt werden.
3. Redundante Inhalte zwischen physikalischen und digitalen Medien sollten vermieden werden.
4. Arbeitsaufwände zur Mediennutzung in den Projekten sollten so gering wie möglich sein.

Ergebnisse zu F2: Welche Eigenschaften und Funktionen von PMS werden im Rahmen verteilter, agiler Softwareentwicklungsprojekte als Vorteil, Nachteil bzw. Verbesserungspotential gesehen und was ist der Grund dafür?

In Tab. 4.3 sind alle aus den Transkripten extrahierten Kategorien zu Vorteilen, Nachteilen und Verbesserungspotenzialen bzgl. konventioneller PMS gelistet. Einen weiteren Anhaltspunkt bzgl. der Praxisrelevanz einer einzelnen Kategorie geben die beiden Kennzahlen *Number of Appearance* (NoA) und *Number of Interviewees* (NoI). Die Kennzahlen wurden im Rahmen der Studienauswertung folgendermaßen definiert:

Definition 4.1. *Number of Appearance:* Diese Kennzahl gibt an wie oft eine Kategorie innerhalb der Interviews adressiert wurde. Die Kennzahl wurde nur erhöht, wenn derselbe Aspekt in einem anderen, klar differenzierbaren Kontext erwähnt wurde.

Definition 4.2. *Number of Interviewees:* Diese Kennzahl gibt an, wie viele Studienteilnehmer diese Kategorie im Rahmen der Interviews angesprochen haben. Sie kann also maximal der Anzahl der Studienteilnehmer entsprechen.

Tabelle 4.3: Ergebnisse zu F2: Vorteile, Nachteile und Verbesserungspotenziale konventioneller PMS im Kontext verteilter, agiler Projekte

Bezeichner	NoA	NoI	Nachteile von PMS-Einsatz
N.1	8	5	Eingeschränkte Flexibilität und Limitationen beim Einsatz der PMS
N.2	6	4	Einstiegsschwierigkeiten bei der Benutzung der PMS
N.3	5	4	Reduzierte Kommunikation und Interaktion in Gruppenarbeitsprozessen
N.4	4	3	Probleme und erhöhter Arbeitsaufwand durch die (teils manuelle) Synchronisation von Daten zwischen physikalischen und digitalen Medien
N.5	3	2	Einsatz physikalischer Medien (z.B. White-Boards, Flip-Charts, etc.) zur Förderung von Kommunikation und Kreativität ist stark reduziert
N.6	3	2	Ein zentraler Treffpunkt der Teammitglieder fehlt
N.7	3	2	Ineffiziente Besprechungskultur
N.8	2	2	Einschränkungen bei gleichzeitiger Benutzung durch mehrere Anwender (MUI)
N.9	2	2	Reduzierte Übersichtlichkeit / Klarheit bei der Darstellung von Projektdaten
N.10	2	1	Während der Besprechungen werden weniger persönliche (handschriftliche) Notizen angefertigt, wodurch Informationen verloren gehen
Bezeichner	NoA	NoI	Vorteil von PMS-Einsatz
V.1	10	5	Weiterführende Dokumente / Unterlagen können direkt mit der Anforderung digital abgelegt werden
V.2	4	4	Automatisierte Archivierung relevanter Projektdaten
V.3	3	3	Digitale Filter- und Suchfunktionen innerhalb der Anforderungen
V.4	3	2	Automatische Generierung und Visualisierung wichtiger Projektkennzahlen
V.5	3	2	Fernzugriff und Screen-Sharing erleichtern die Arbeit in verteilten Projekten
Bezeichner	NoA	NoI	Verbesserungspotenziale konventioneller PMS
P.1	5	2	Zusätzliche Funktionen zur Unterstützung von Controlling-Aufgaben (z.B. Budgetplanung, Zeitplanung)
P.2	1	1	Detaillierte Visualisierung des Projektfortschritts
P.3	1	1	Unterstützung klassischer und agiler Entwicklungsmodelle

4.2.6 Verbesserungsmöglichkeiten als Schlüsselanforderungen an das Lösungskonzept

Nachfolgend sollen auf Basis der Vorstudienresultate Verbesserungsmöglichkeiten aufgezeigt werden, die in erster Linie durch den Einsatz natürlicher Benutzerschnittstellen erreicht werden können. Dabei spielen die Ergebnisse der Vorstudie eine wichtige Rolle: Die im Rahmen der Studie erarbeiteten Vorteile (Tab. 4.3) konventioneller PMS sollen erhalten bleiben, während die Nachteile (Tab. 4.3) abgemildert werden. Die ebenfalls in Tab. (Tab. 4.3) gelisteten Verbesserungspotenziale beziehen sich auf spezifische Funktionserweiterungen innerhalb konventioneller PMS, die aus Sicht der Studienteilnehmer sinnvoll sind. Sie wurden aus Gründen der Vollständigkeit aufgelistet, sind aber nicht im Fokus dieser Arbeit (vgl. Forschungsziele 1.3) und werden nicht direkt adressiert. Neben den in Tab. 4.3 gezeigten Aspekten sollen in erster Linie die im übergreifenden Ziel (Z) (vgl. 1.3) definierten Aspekte Motivation, Transparenz und Kommunikation als Schlüsselfaktoren guter Teamarbeit im Fokus stehen.

Folgende Verbesserungsmöglichkeiten sehen wir durch den Einsatz natürlicher Benutzerschnittstellen im Umfeld konventioneller PMS als Ersatz bzw. Ergänzung zu den üblicherweise browserbasierten Benutzerschnittstellen:

- **Reduktion der Bildschirmarbeitszeit in Besprechungen und verstärkter Einsatz physikalischer Medien:** Die Reduktion der Bildschirmarbeit während der Regelbesprechungen im agilen Prozess birgt unserer Meinung nach das größte Verbesserungspotenzial bzgl. der in Tab. 4.3 gelisteten Nachteile und der Schlüsselfaktoren Motivation, Kommunikation und Transparenz in einem Projektteam. Diese Annahme beruht in erster Linie auf den Recherchen zur Bedeutung physikalischer Medien für agile Arbeitsbereiche (vgl. 3.1.3 ff.) und auf den Grundlagen des OWP.
- **Mehr Transparenz im Sinne des OWP (vgl. 3.1.3) im Arbeitsbereich:** Dieser Punkt adressiert die transparente Darstellung des agilen Arbeitsprozesses im unmittelbaren Arbeitsumfeld agiler Entwicklungsteams. Durch die Integration natürlicher Benutzerschnittstellen in konventionelle PMS können elementare Bestandteile agiler Arbeitsumgebungen (z.B. Task-Board, Product-Backlog-Board) auch dann zum Einsatz kommen, wenn die dezentrale Verfügbarkeit der darauf dargestellten Daten notwendig ist (also immer dann mehrere Entwicklungsteams an unterschiedlichen Standorten zusammen arbeiten). Die größten Vorteile des OWPs (z.B. Förderung direkter Kommunikation/Interaktion, Transparenz und Motivation im Entwicklungsteam) sollen wieder besser unterstützt werden.
- **Vermeidung redundanter Projektdaten in PMS und auf physikalischen Medien:** Die Workflow-Analysen (vgl. F1 der Vorstudie) haben deutlich gezeigt, dass redundante Datenhaltung (z.B. PMS und Task-Board) Probleme wie Inkonsistenzen zwischen den Systemen und erhöhten Aufwand (Pflege der Projektdaten an mehreren Stellen) mit sich bringen. Durch die Integration natürlicher Benutzerschnittstellen könnten diese Probleme adressiert werden, weil die parallele Verwendung von PMS und physikalischen Medien vermieden wird.

- **Erhalt der Standardfunktionen konventioneller PMS:** Eine weitere Schlüsselanforderung an das Lösungskonzept ist der Erhalt der Standardfunktionen konventioneller PMS. Neben der ortsunabhängigen Verfügbarkeit sind Funktionen wie automatische Archivierung von Projektdaten, Generierung von Kennzahlen sowie Filter und Suchfunktionen der Grund für den stark verbreiteten Einsatz von PMS. Mit Hilfe eines geeigneten Lösungskonzepts sollen diese Funktionen weiterhin nutzbar bleiben.

Kapitel 5

Lösungskonzept - Virtual Task-Board

Dieses Kapitel beinhaltet sowohl die Zieldefinition (Schritt 2 im DSRP) als auch die Konzeption eines entsprechenden Lösungsartefakts (Schritt 3 im DSRP). In 1.3 wurde bereits das grundlegende Ziel der Konzeptionsphase beschrieben. Zu Beginn dieses Kapitels in 5.1 wird es noch einmal konkretisiert. Der erste Schritt zur Zielerreichung ist die anschließende Anforderungsanalyse in 5.2. Sie definiert die wichtigsten Aspekte zur Zielerreichung auf funktionaler und nichtfunktionaler Ebene. Anschließend folgt die Vorstellung des Lösungskonzepts Virtual Task-Board. Hierbei stehen sowohl fachliche Überlegungen auf Basis der dokumentierten Anforderungen als auch technische Machbarkeitsüberlegungen im Fokus. Die Realisierung eines Prototypen als Evaluationsgrundlage wird anschließend in 5.7 ff. erläutert.

5.1 Konkretisierung der Zieldefinition

Es soll eine natürliche Benutzerschnittstelle zur automatisierten Erfassung projektrelevanter Daten für den Einsatz in agilen Projekten konzipiert werden, um die Gruppenarbeit innerhalb agiler Teams bzgl. Kommunikation, Transparenz und Motivation bestmöglich zu unterstützen. Dabei sollen die in 4.2.5 vorgestellten Erkenntnisse der Vorstudie als grundlegende Gestaltungsprinzipien konsequent angewandt werden. Weiterhin sollen die Vorteile konventioneller PMS (vgl. Tab. 4.3) erhalten bleiben und Verbesserungen bzgl. der eruierten Nachteile erreicht werden.

5.2 Anforderungsanalyse

Die Anforderungsanalyse muss aus zwei unterschiedlichen Perspektiven durchgeführt werden. Zum einen muss der Blickwinkel eines lokalen Entwicklungsteams untersucht werden. Dabei nehmen wir im Rahmen dieser Arbeit ein physikalisches (papierbasiertes) Task-Board als die beste State-of-the-Art Lösung an und versuchen dessen Eigenschaften bzgl. Datenmodell, Gebrauchstauglichkeit und Funktionsumfang möglichst nahe zu kommen. Das physikalische Task-Board gilt im Umfeld agiler Methoden als aktueller Stand der Wissenschaft bzgl. der Kriterien Kommunikation, Transparenz und Motivation (vgl. [61, 78, 76, 77, 46, 63] und 3.1.3.1).

Zum anderen muss die Anforderungsanalyse aus der Perspektive eines entfernt

arbeitenden Entwicklungsteams durchgeführt werden. Dabei stehen vor allem Daten und Projektinformationen im Fokus, die für ein verteilt arbeitendes Team relevant sind, um die eigene Projektplanung entsprechend durchführen zu können.

Anmerkung: In den nachfolgenden Abschnitten bezieht sich der Terminus "Projektanforderungen" auf die Gesamtheit von Epic, Stories und Tasks in einem Softwareentwicklungsprojekt. Dieses Vorgehen soll die Unterscheidung zu den Anforderungen an das im Rahmen dieser Arbeit entwickelte Lösungskonzept erleichtern.

5.2.1 Funktionale Anforderungen

Für den ersten Schritt der Anforderungsanalyse wurde ein Datenmodell erstellt (vgl. Fig. 5.1), das die grundlegenden Entitäten eines Task-Boards sowie deren Beziehungen zueinander darstellt. Eine einheitliche Definition eines Task-Boards findet sich in der Literatur nicht, da es immer an die individuellen Erfordernisse eines Projekts bzw. Entwicklerteams angepasst wird. Die grundlegende Funktionsweise und damit auch das grundlegende Datenmodell sind jedoch immer gleich. Das in Fig. 5.1 gezeigte Modell beruht auf dem Vorschlag eines Task-Boards nach Dalton [20]. Es wurde die sog. Chen-Notation [15] zur Darstellung von Entitäten und deren Beziehungen zueinander genutzt. Im Wesentlichen basiert das gezeigte Modell auf dem in 3.1.2.2 vorgestellten Ebenenmodell für User Stories nach Leffingwell [51]. Entsprechend gibt es im Sprint-Backlog sowie im Product-Backlog beliebig viele Epics (dargestellt durch die 1 zu n Beziehung). Das Epic beinhaltet wiederum beliebig viele Stories, die anhand von Tasks umgesetzt werden. Die jeweiligen Attribute der User-Stories (oval dargestellt) entsprechen den Anforderungen an eine gute User-Story nach Mike Cohn [17]. Der Bearbeitungszustand entspricht den am Task-Board dargestellten Spalten (vgl. 3.1.3.3). Hier werden in der Regel die Ausprägungen "ToDo", "In Progress", "In Verification" und "Done" verwendet [20].

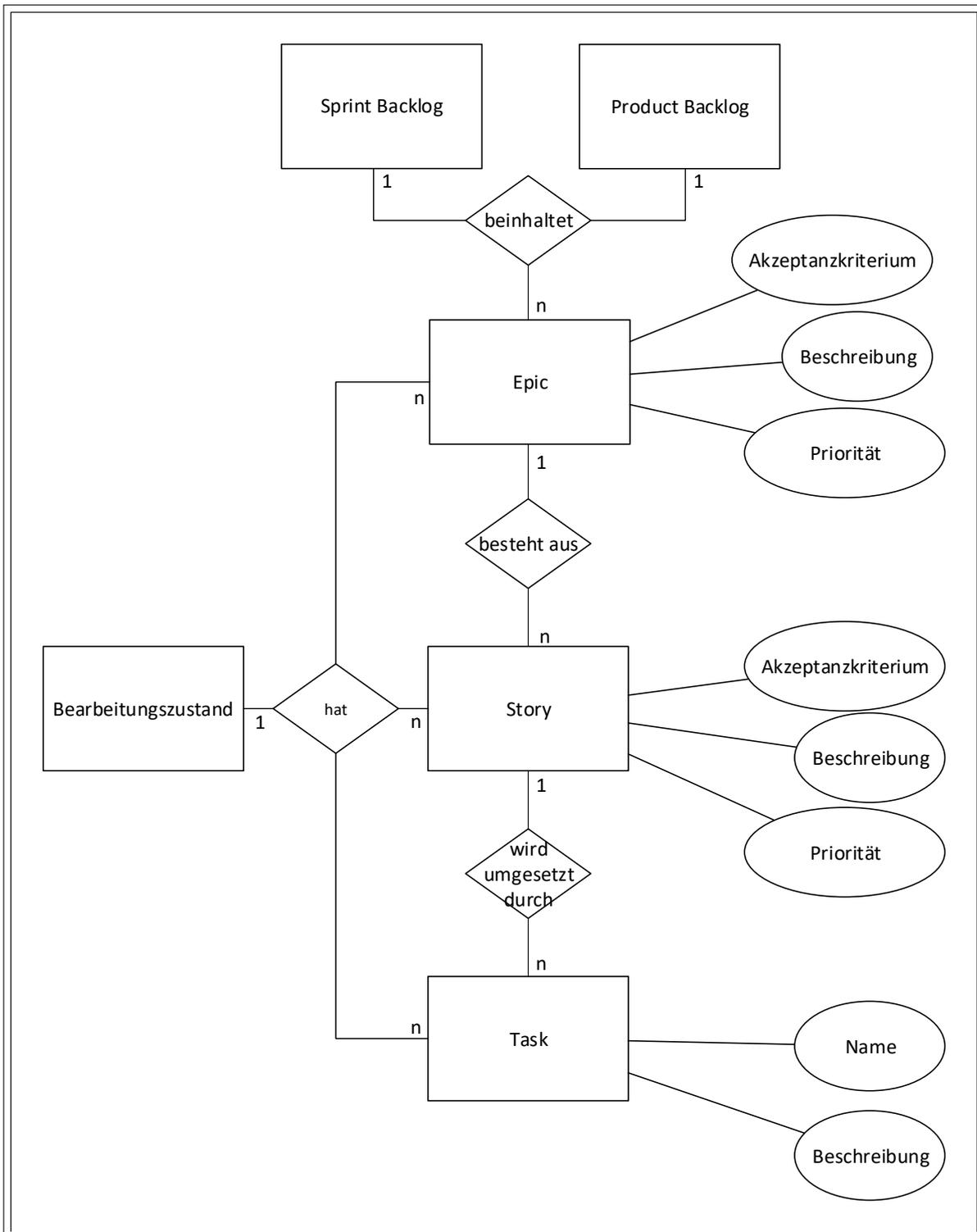


Abbildung 5.1: Grundlegendes Datenmodell eines Task-Boards

Entsprechend dem in 1.3 formulierten Ziel Z2 ergeben sich daraus folgende funktionale Anforderungen:

Datenerfassung

- A.1.:** Erfassung aller am Task-Board dargestellten Projektanforderungen inklusive der im Datenmodell definierten Attribute.
- A.2.:** Unterscheidung der Ausprägungen aller Projektanforderungen (z.B. Epic, Story, Tasks).
- A.3.:** Erfassung der im Datenmodell dargestellten Relationen zwischen den Projektanforderungen.
- A.4.:** Zuordnung der erfassten Projektanforderungen zu Sprint- bzw. Product-Backlog.

Ein wichtiger Aspekt des Lösungskonzepts ist die ortsunabhängige Bereitstellung der erfassten Daten, um die Zusammenarbeit örtlich verteilter Entwicklungsteams zu unterstützen. Die Grundidee hierfür basiert auf der Verwendung einer konventionellen PMS zur automatisierten Spiegelung der erfassten Daten. Hierbei ist zu beachten, dass das entfernte Entwicklungsteam nur lesend auf die Daten zugreifen soll. Es muss sichergestellt werden, dass die in der PMS dargestellten Daten zu jederzeit den Zustand auf dem physikalischen Task-Board widerspiegeln, um Inkonsistenzen auszuschließen. Des Weiteren müssen die Ergebnisse der Vorstudie (vgl. 4) berücksichtigt werden. Im Rahmen der Studie wurden Vorteile, Nachteile und Verbesserungspotentiale bei der Verwendung konventioneller PMS erarbeitet. Eine detaillierte Auflistung ist in Tab. 4.3 gegeben. Entsprechend ergeben sich folgende Anforderungen bzgl. der Bereitstellung und Verarbeitung der erfassten Daten:

Bereitstellung und Verarbeitung der erfassten Projektdaten

- B.1.:** Unidirektionale Übertragung der erfassten Daten in eine geeignete PMS.
- B.2.:** Vermeidung von Inkonsistenzen zwischen den Daten am physikalischen Task-Board und der PMS.
- B.3.:** Verknüpfung bzw. Ablage digitaler Dokumente mit einer Projektanforderung.
- B.4.:** Automatisierte Archivierung relevanter Projektdaten (entsprechend dem Datenmodell).
- B.5.:** Filter und Suchoptionen innerhalb der Projektanforderungen.
- B.6.:** Generierung und Visualisierung projektrelevanter Kennzahlen.
- B.7.:** Ortsunabhängiger Zugriff auf die erfassten Daten.
- B.8.:** Gleichzeitige Bearbeitung durch mehrere Benutzer (MUI) muss möglich sein.

5.2.2 Nichtfunktionale Anforderungen

Entsprechend dem in 1.2 beschriebenen Forschungsproblem liegt der Fokus des Lösungskonzepts auf der Verbesserung der Gruppenarbeit innerhalb der jeweiligen

Entwicklungsteams. Ansatzpunkt sind die in 3.1.3.1 und Raith et al. [69] aufgezeigten Probleme, die auf den Einsatz von PMS im Rahmen verteilter agiler Projekte zurückzuführen sind. Für die Anforderungsanalyse werden deshalb nichtfunktionale Anforderungen aus zwei Quellen herangezogen. Erstens finden sich in der Literatur verschiedene Arbeiten mit Heuristiken und Praxisvorschlägen für die Gestaltung und Konzeption agiler Arbeitsumgebungen wie z.B. Perry [63] und del melo Oliveira et al [55, 56]. Zweitens müssen auch hier die Ergebnisse der Vorstudie (vgl. Kap. 4) mit in die Anforderungsanalyse einfließen. Entsprechend ergeben sich folgende Anforderungen:

- C.1.:** Intuitive Gestaltung der Benutzerschnittstelle, um Einarbeitungszeiten gering zu halten und die tägliche Arbeit zu erleichtern.
- C.2.:** Reduktion der dargestellten Inhalte auf das Wesentliche (Orientierung am Datenmodell).
- C.3.:** Geringer Kosten und Wartungsaufwand für die Anwender.
- C.4.:** Hohe Flexibilität, um die individuelle Anpassbarkeit an die Bedürfnisse unterschiedlicher Entwicklungsteams zu ermöglichen.
- C.5.:** Reduktion von Medienbrüchen und redundanten Arbeitsabläufen.
- C.6.:** Interaktion mit digitalen Benutzerschnittstellen (z.B. PC mit Maus und Tastatur) soll stark reduziert werden.

Besonders bei den nichtfunktionalen Anforderungen ist eine eindeutige Definition nicht immer möglich. Weder aus der genannten Literatur noch aus der Vorstudie lassen sich hierzu genaue Aussagen (z.B. Metriken, Kennzahlen) ableiten. Eine explizite Auflistung erscheint dennoch sinnvoll, da sie bei der Konzeption unbedingt berücksichtigt und umgesetzt werden müssen.

5.3 Grundidee - Nahtlose Integration konventioneller PMS mit Hilfe natürlicher Benutzerschnittstellen

Entsprechend unserer in 5.2 getroffenen Annahme ist ein physikalisches Task-Board aus der Perspektive eines lokal arbeitenden, agilen Projektteams die zu bevorzugende Lösung zur Unterstützung des agilen Anforderungsmanagements, da es die Gruppenarbeit hinsichtlich Kommunikation, Transparenz und Motivation sehr gut unterstützt. Deshalb soll ein physikalisches Task-Board als grundlegendes TUI zur digitalen Erfassung und direkten Bearbeitung aller in Fig. 5.1 dargestellten Daten dienen (vgl. Z2 in 1.3). Entscheidend hierbei ist, dass sich die Benutzung der angedachten Lösung möglichst wenig von der Benutzung eines konventionellen, physikalischen Task-Boards unterscheidet, um den Arbeitsprozess nicht zu beeinflussen. Das Lösungskonzept muss demnach die Kernprobleme der automatisierten Datenerfassung (vgl. Anforderungen A.1. - A.4.), die ortsunabhängige Verfügbarkeit und Darstellung relevanter Daten und Kennzahlen (vgl.

Anforderungen B.1. - B.8.) unter Einhaltung der nichtfunktionalen Anforderungen (vgl. C.1. - C.5.) lösen. Der grundlegende Lösungsansatz ist nachfolgend in Fig. 5.2 dargestellt:

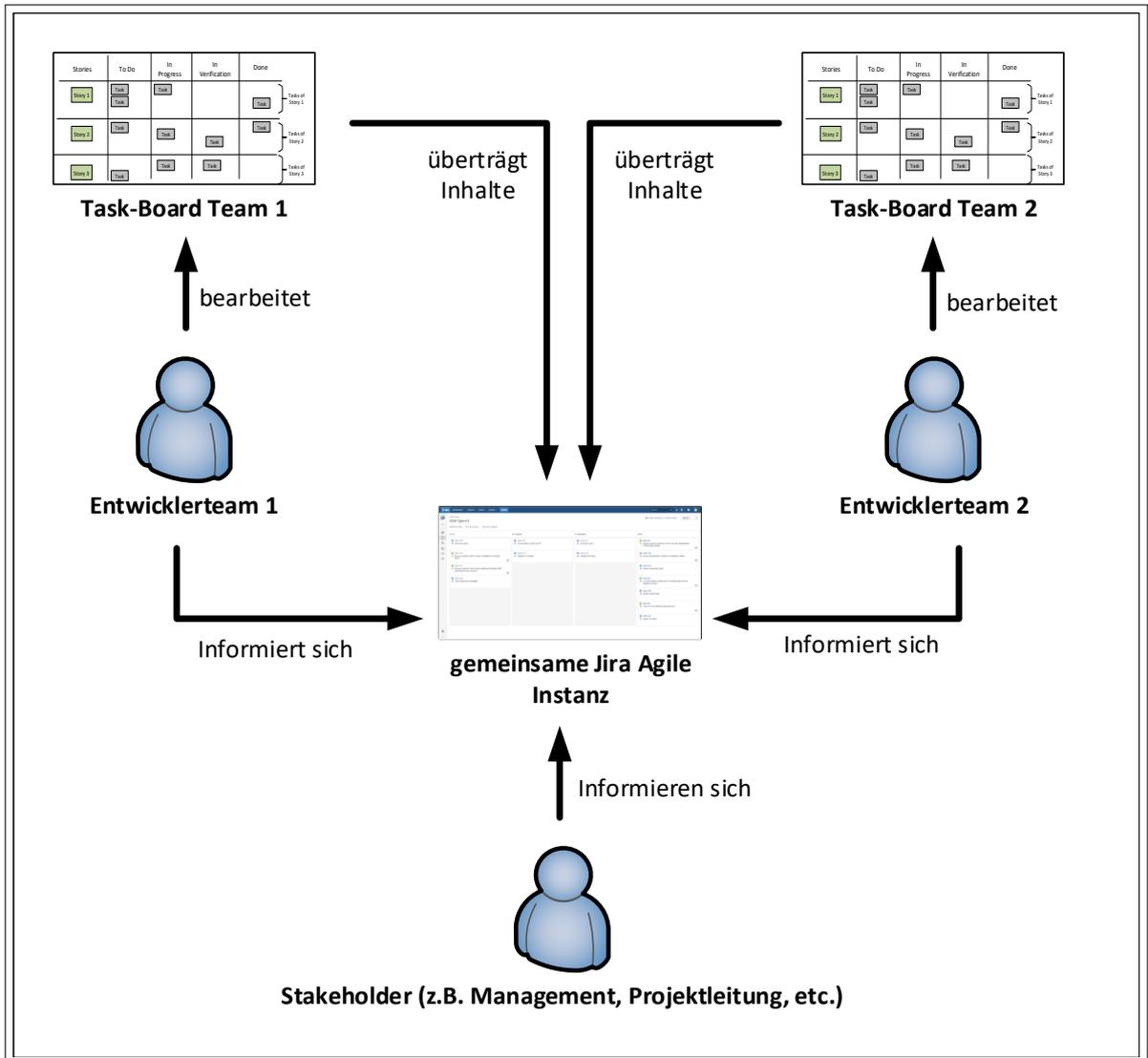


Abbildung 5.2: Grundlegender Aufbau des Lösungskonzepts

Die Grafik zeigt beispielhaft die Beteiligung zweier agiler Teams am Arbeitsprozess. Es können jedoch beliebig viele Entwicklerteams in den Arbeitsablauf eingebunden werden. Jedes Entwicklerteam arbeitet an einem eigenen zentralen Task-Board, das die Inhalte der aktuellen Iteration sowie den momentanen Arbeitsfortschritt zeigt (vgl. 3.1.3.3). Die Inhalte auf dem Task-Board werden automatisiert erfasst und in eine konventionelle PMS (in diesem Fall Jira Agile) übertragen. Wichtig ist hierbei der Aspekt der unidirektionalen Datenübertragung vom physikalischen Task-Board zur PMS. Durch diese Vorgehensweise bleibt das physikalische Task-Board die zentrale Benutzerschnittstelle und Inkonsistenzen zwischen physikalischem Task-Board und PMS werden automatisch vermieden. Die

Inhalte des physikalischen Task-Boards überschreiben die Inhalte der PMS, wodurch immer ein konsistenter Zustand der Daten sichergestellt ist. Diese Grundidee trifft unter der Annahme zu, dass jedes Team ausschließlich an seinen eigenen Inhalten und Anforderungen arbeitet. Inhalte eines anderen Teams können und sollen explizit nicht verändert werden, sondern dienen als Informationsquelle. Diese Annahme muss nicht als Einschränkung verstanden werden, sondern unterstützt den Grundgedanken, dass ausschließlich die physikalische Benutzerschnittstelle als Eingabemedium genutzt wird. Auf diese Art ist sichergestellt, dass die Transparenz innerhalb des Projektteams gegeben ist. Die grafische Benutzeroberfläche der PMS wird demnach nur lesend genutzt. Sie kann verwendet werden, um relevante Informationen der anderen Entwicklerteams einzusehen und in der eigenen Projektplanung zu berücksichtigen. Weiterhin können alle Funktionsumfänge der PMS zur Generierung und Darstellung wichtiger Projektkennzahlen genutzt werden. Die browserbasierte Benutzeroberfläche erlaubt einen ortsunabhängigen Zugriff mittels PC und Mobilgerät durch alle Mitglieder der Entwicklerteams sowie weitere interessierte Stakeholder (z.B. Management, Projektleitung, etc.).

5.4 Lösungsidee zur automatisierten Erfassung relevanter Projektdaten

Die Anforderungen A.1. bis A.4. fordern die automatisierte Erfassung aller im Datenmodell (vgl. Abb. 5.1) dargestellten Entitäten und Relationen. Um diese Daten vollständig erheben zu können müssen nur wenige grundlegende Eckdaten bzgl. des Task-Boards bzw. des Product-Backlogs und der darauf dargestellten Inhalte korrekt interpretiert werden. Folgende Eckdaten sind notwendig:

1. Handschriftliche Inhalte jeder Story-Card
2. Identifikation einer Story-Card am Task-Board
3. Die exakte Position einer Story-Card auf dem Task-Board bzw. am Product-Backlog
4. Die Aufteilung des Task-Boards bzw. des Product-Backlogs (Bedeutung der einzelnen Zeilen und Spalten)
5. Unterscheidung der Granularität einer User-Story nach dem Ebenenmodell (Epic, Story, Task) in Anlehnung an Leffingwell [51]

In den folgenden Absätzen werden entsprechende Lösungsmöglichkeiten zur Erfassung dieser Daten aufgezeigt. Außerdem wird auf die korrekte Interpretation der Daten auf Basis des geforderten Datenmodells eingegangen. Neben der automatisierten Erfassung von Daten sind auch eine Reihe statisch hinterlegter Konfigurationen (z.B. zur Beschreibung projektspezifischer Eigenschaften des Task-Boards) notwendig, die ebenfalls in den folgenden Absätzen beschrieben werden.

5.4.1 Erfassen der handschriftlichen Notizen auf den Story-Cards

Eine Grundvoraussetzung zur Umsetzung der übergreifenden Lösungsidee ist die Erfassung der handschriftlich festgehaltenen Inhalte auf den Story-Cards. Grundlage hierfür ist die Umwandlung der handschriftlichen Texte in digital lesbare bzw. bearbeitbare Texte. Aus wissenschaftlicher Sicht wird das Thema Handschrifterkennung (engl. *Handwriting Recognition* (HWR)) nach zwei Ansätzen unterschieden [65]. Der erste Ansatz -die Offline Erkennung- basiert auf pixelbasierten Bildformaten als Eingangsgröße, wie man sie z.B. durch das Einscannen oder Abfotografieren handschriftlicher Texte erhält. Der zweite Ansatz -die Online Erkennung- basiert auf zweidimensionalen Koordinaten, die mit Hilfe eines elektronischen Eingabegeräts (z.B. Computermaus, Touch-Display, etc.) erfasst werden. Häufig wird dieses Eingabeformat auch als "Digital Ink"(dt. digitale Tinte) bezeichnet. Grundlegend setzt es sich aus sog. "Strokes" (dt. Striche) zusammen. Ein Stroke ist die entsprechende Trajektorie des Schriftzuges vom Zeitpunkt des Aufsetzens bis zum ersten Abheben des Stiftes. Unter der Trajektorie versteht man in diesem Zusammenhang die zweidimensionalen Koordinaten in chronologischer Reihenfolge des jeweiligen Schriftzuges. Das wichtigste Qualitätsmerkmal aller HWR Ansätze ist eine hohe Erkennungsrate. Laut Priya et al. [67] ist eine Erkennungsrate größer 97 Prozent notwendig, um Akzeptanz bei den Benutzern zu erreichen. Der Online-Ansatz kann aufgrund des präziseren Eingangssignals wesentlich bessere Erkennungsraten aufweisen [67, 65], weshalb wir uns im Rahmen dieser Arbeit für diesen Ansatz entschieden haben. Eines der zu lösenden Kernprobleme ist demnach die Integration der Online-Handschrifterkennung in die tägliche Arbeit mit dem Task-Board. Entsprechend der formulierten nichtfunktionalen Anforderungen C.5. und C.6. (vgl. 5.2.2) soll die Arbeit mit digitalen Eingabegeräten weitestgehend vermieden werden. Weiterhin ist ein Ziel des Lösungskonzepts, dass die Arbeit mit dem Virtual Task-Board Konzept sich möglichst wenig von der Arbeit mit einem konventionellen, papierbasierten Task-Board unterscheidet. Da die Online-Handschrifterkennung aufgrund der benötigten Eingabedaten ein elektronisches Eingabegerät erfordert, sind wir im Rahmen erster Recherchen schnell auf sog. Smart-Pens oder auch Digitalstifte gestoßen. Dabei handelt es sich grundlegend um normale Stifte (z.B. mit herkömmlicher Kugelschreibermine), die zusätzlich um geeignete Sensorik zur Erfassung von Handschrift in Form von zweidimensionalen Koordinaten erweitert wurden. Der Markt hält dabei zwei unterschiedliche Ansätze bereit:

Digitalstifte mit Spezialpapier: Diese Stifte besitzen im Regelfall eine kleine Infrarotkamera, die an der Unterseite des Stiftes sitzt und in Richtung Papier ausgerichtet ist. Auf dem zwingend benötigtem Spezialpapier befinden sich kleine Markierungen (meist kleine Punkte), die der Stift zur Positionierung nutzt. Der große Nachteil hierbei ist, dass nur das vom entsprechenden Hersteller angebotene Spezialpapier verwendet werden kann. Neben relativ hohen Preisen gibt es meist nur wenige vorgegebene Formate in Form kleinerer Notizbücher (z.B. DIN A5 Ringbücher).

Digitalstifte mit Empfänger: Diese Art der Digitalstifte kann auf ganz normalem Papier verwendet werden, da die Positionserkennung nicht vom Papier, sondern von einem externen Empfänger abhängig ist. Der Empfänger ist im Regelfall mit Infrarot- und

5.4 Lösungsidee zur automatisierten Erfassung relevanter Projektdaten

Ultraschallsensorik ausgestattet und ermöglicht eine Positionserkennung des Stiftes relativ zur Position des Empfängers.

Auswahl eines geeigneten Digitalstiftes: Zur Anwendung im Virtual Task-Board Konzept haben wir uns für einen Digitalstift mit Empfänger entschieden. Entscheidend war hierbei die Unabhängigkeit von herstellereigenem Spezialpapier und die damit verbundenen Einschränkungen bzgl. Papierformaten und Farben. Zusätzlich können weitere Kosten und Aufwände für die Beschaffung von Spezialpapier vermieden werden. Als Vertreter digitaler Stifte mit Empfänger haben wir das Modell Smart-Pen 2 der Firma Equil gewählt. Das Modell bringt einen kleinen kabellosen Empfänger mit, der an einer Seite des Papiers bzw. der Karteikarte befestigt werden kann (vgl. Abb. 5.3). Weiterhin stellt der Hersteller freie Programmierschnittstellen (engl. *Application Programming Interface* (API)) zur Verfügung, die zur technischen Anbindung zwingend erforderlich sind. Abb. 5.3 zeigt den Equil Smart-Pen 2 inklusive dem zugehörigen Empfänger.



Abbildung 5.3: Digitalstift Smart-Pen 2 von Equil mit Empfänger
(Quelle: <https://img.purch.com>)

Auswahl einer geeigneten Online-Handschrifterkennung: Als besonders geeignet hat sich die Online-Handschrifterkennung der Firma MyScript herauskristallisiert. Neben einer sehr hohen Erkennungsrate bringt das Produkt weitere Funktionen mit sich, die für die Bearbeitung von User-Stories besonders wichtig sind. So können z.B. Zeichnungen, Skizzen und mathematische Formeln von normalem Text unterschieden werden. Zudem ermöglicht die Handschrifterkennung das Löschen von einzelnen Textstellen durch einfaches Durchstreichen mit dem Digitalstift. Diese Eigenschaften waren für die Entscheidung

ausschlaggebend, weil es durchaus üblich und sinnvoll ist, einzelne Anforderungen mittels Zeichnungen oder Formeln auf der Story-Card zu veranschaulichen.

5.4.1.1 Integration des Digitalstiftes und der Handschrifterkennung in den Arbeitsprozess

Ziel dieses Abschnitts ist ein Teilkonzept zur Integration des Digitalstiftes sowie der Handschrifterkennung in die normale Arbeit mit dem Task-Board. Alle technischen Eingriffe, die zur Datenerfassung notwendig sind sollen möglichst nahtlos in den Arbeitsablauf integriert werden, um intuitives und benutzerfreundliches Arbeiten zu ermöglichen. Mit Hilfe des Digitalstiftes werden die folgenden Anwendungsfälle abgebildet:

- Hinzufügen neuer Story-Cards zum Product-Backlog und Sprint-Backlog
- Editieren bereits existierende Story-Cards im Product-Backlog und Sprint-Backlog

Das Teilkonzept besteht einerseits aus einer speziellen Vorlage für Story-Cards, welche die Anwendung des Digitalstiftes erleichtert und die Erfassung der im Datenmodell geforderten Angaben ermöglicht. Andererseits beinhaltet das Konzept eine Smartphone-Applikation, die zur Weiterverarbeitung der erfassten Daten und als einfache Benutzerschnittstelle eingesetzt wird. In den folgenden Absätzen werden die einzelnen Bestandteile des Konzepts kurz vorgestellt.

Konzeption einer geeigneten Vorlage für Story-Cards: Zunächst muss ein geeignetes Format der Story-Cards konzipiert werden, das den reibungslosen Einsatz des Stiftes sowie das Anbringen des Empfängers erlaubt. Es ist notwendig, dass der Empfänger immer an der gleichen Stelle der Story-Card angebracht wird, da die Positionierung des Stiftes stets relativ zum Empfänger berechnet wird. Maßgeblich bei der Ermittlung einer geeigneten Stelle war die maximale Reichweite des Empfängers sowie der unterbrechungsfreie Einsatz des Stiftes. Bei unseren Versuchen fiel auf, dass bei einer Positionierung am oberen Kartenrand häufig die eigene Hand zu Verbindungsunterbrechungen führt. Ist der Empfänger zu nah an am Stift führt das ebenfalls zu kurzen Unterbrechungen. Nach einigen Testläufen mit unterschiedlichen Personen hat sich der linke Kartenrand als die am besten geeignetste Stelle herauskristallisiert. Es muss jedoch beachtet werden, dass die Benutzung des Digitalstiftes durch Linkshänder somit nur eingeschränkt möglich ist. Eine Lösungsmöglichkeit wäre die optionale Anbringung des Sensors am rechten Rand mit einer softwareseitigen Spiegelung der Koordinaten. Da es sich hierbei jedoch um ein sehr gerätespezifisches Problem handelt, wird diese Lösungsmöglichkeit im Rahmen dieser Arbeit nicht weiter betrachtet und die Einschränkung in Kauf genommen. Abb. 5.4 zeigt den beispielhaften Aufbau einer Story-Card.

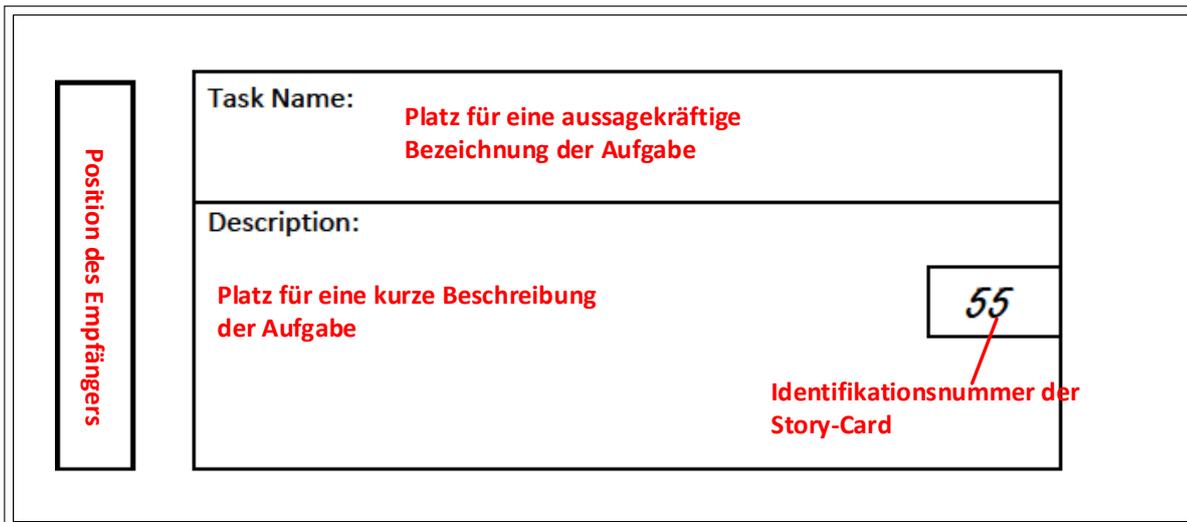


Abbildung 5.4: Vorlage einer Story-Card

Links befindet sich eine Markierung zum Anbringen des Empfängers. Er kann wahlweise am Papier festgeklemmt oder einfach darauf abgelegt werden. Weiterhin gibt es verschiedene vordefinierte Bereiche, die zur Erfassung einzelner Attribute benötigt werden. Im dargestellten Beispiel handelt es sich um eine Vorlage vom Typ "Task" (vgl. Datenmodell in Abb. 5.1). Entsprechend gibt es vordefinierte Bereiche für die Attribute "Name" und "Beschreibung". Durch die Festlegung der einzelnen Bereiche kann der digital erfasste Inhalt später korrekt interpretiert und den richtigen Attributen zugeordnet werden.

Ein weitere technische Anforderung der automatisierten Datenerfassung ist die korrekte Identifikation einer Story-Card vor deren Bearbeitung. Im System muss bekannt sein, welche Story-Card gerade bearbeitet wird. Zu diesem Zweck enthält jede Vorlage eine eindeutige Identifikationsnummer, die ebenfalls innerhalb eines vordefinierten Bereichs auf der Story-Card platziert ist. Diese Identifikationsnummer ist für jede einzelne Story-Card fest vorgegeben und muss bereits beim Druck der Vorlage enthalten sein. Bevor mit der Bearbeitung einer Story-Card begonnen werden kann muss diese Identifikationsnummer lediglich einmal mit dem Digitalstift nachgezeichnet werden. Auf diese Weise ist im System bekannt, dass die Bearbeitung einer neuen Story-Card begonnen hat und die erfassten Daten können dem korrekten Datensatz zugeordnet werden. Der beschriebene Mechanismus stellt aus Anwendersicht einen Mehraufwand gegenüber der Arbeit mit einem konventionellem Task-Board dar. Trotzdem ist dieser Schritt aus technischer Sicht notwendig und kann nicht ausgespart werden.

Nach dem gleichen Prinzip wurden auch Vorlagen für Story-Cards vom Typ "Story" und "Epic" mit den entsprechend notwendigen Attributen (siehe Datenmodell in Abb. 5.1) konzipiert. Ein Beispiel befindet sich jeweils im Anhang dieser Arbeit unter B.1.

Konzeption einer Smartphone-Anwendung: Ein weiterer wichtiger Schritt zur Integration des Digitalstiftes in den Arbeitsprozess ist die technische Anbindung an ein Gerät,

das die Weiterverarbeitung der erfassten Daten übernimmt und gleichzeitig eine einfache grafische Benutzerschnittstelle bereitstellt. Die Benutzerschnittstelle soll ausschließlich als Feedback-Schnittstelle Richtung Anwender dienen, um über die korrekte Funktion des Systems bzw. über Anwendungsfehler sowie technische Probleme zu informieren. Eine direkte Aktion mit der grafischen Benutzerschnittstelle soll vermieden werden, da die Interaktion während des Arbeitsprozesses ausschließlich über das Task-Board bzw. den Digitalstift erfolgen soll (vgl. nichtfunktionale Anforderung C.6.). Ein geeignetes Gerät ist z.B. ein Smartphone, da es ortsunabhängig eingesetzt werden kann und über alle technischen Anforderungen wie z.B. eine Bluetooth-Schnittstelle zur Kommunikation mit dem Digitalstift, ein Touch-Display sowie eine Netzwerkschnittstelle verfügt.

Abb. 5.5 zeigt ganzheitlich die Integration der Smartphone-Anwendung in das Gesamtkonzept. Die einzelnen Aufgaben bzw. Funktionen der Smartphone-Anwendung werden in den anschließenden Absätzen beschrieben.

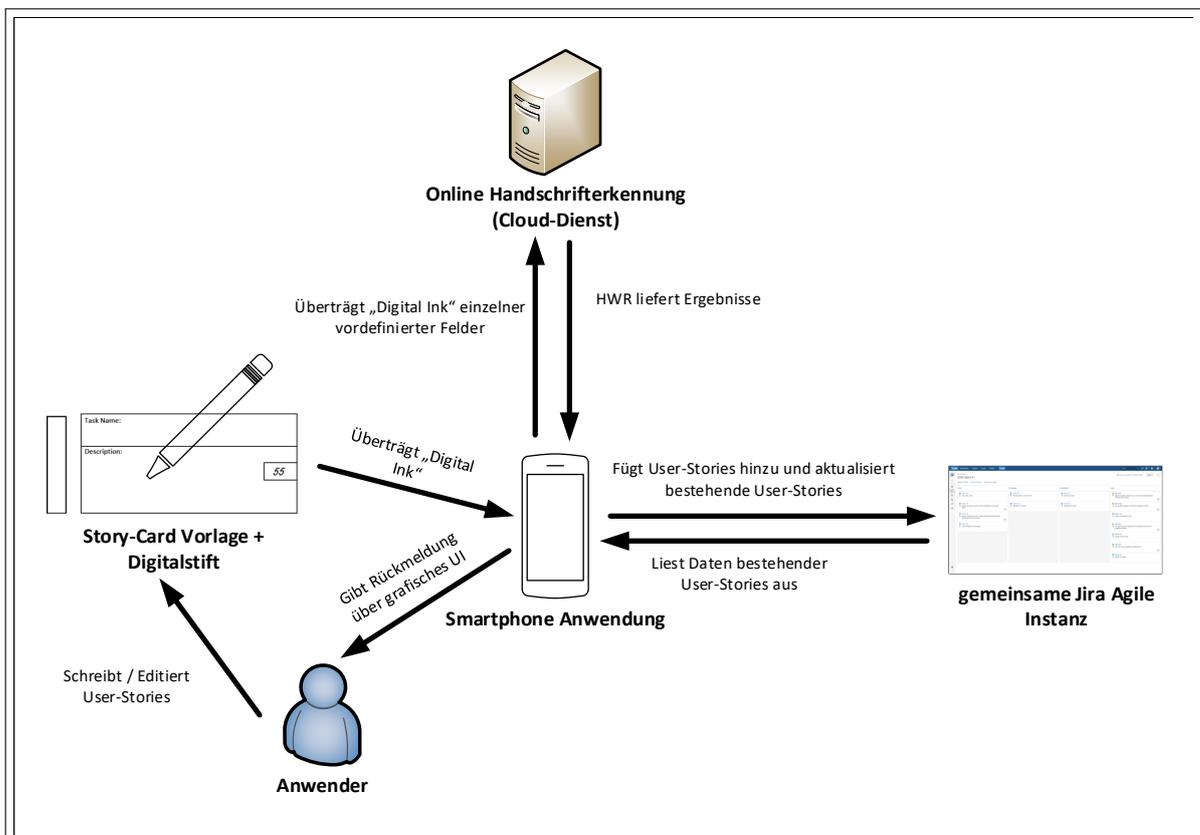


Abbildung 5.5: Einbindung Smartphone-Anwendung in das Gesamtkonzept

Technische Anbindung des Digitalstiftes via Bluetooth: Das Virtual Task-Board Konzept sieht die direkte Verarbeitung der handschriftlichen Eingaben beim Erstellen bzw. Editieren einer Story-Card vor, um verzögerungsfreies Arbeiten zu ermöglichen. Zu diesem Zweck wird eine dauerhafte Datenverbindung des Digitalstiftes mit der Smartphone-Anwendung benötigt. Das ausgewählte Modell Smart-Pen 2 von Equil (vgl. 5.4.1) erlaubt das Auslesen dieser Daten bereits während des Schreibvorgangs

via Bluetooth-Schnittstelle. Aufgabe der Smartphone-Anwendung ist ein zuverlässiger Verbindungsaufbau sowie das Zwischenspeichern der Eingabedaten während des Schreibvorgangs.

Aufbereitung der Digital Ink: Die wichtigste Aufgabe der Smartphone-Anwendung hierbei ist die Zuteilung der Handschriftdaten zu den vordefinierten Feldern auf der Story-Card. Wie bereits eingangs beschrieben besitzen alle Vorlagen fest definierte Bereiche, deren handschriftlicher Inhalt bestimmten Feldern im Datenmodell zugeordnet werden muss. Diese Bereiche müssen innerhalb der Smartphone-Anwendung bekannt sein (z.B. durch Konfiguration). Damit ist die Anwendung in der Lage, die eingehenden Daten des Digitalstiftes entsprechend der Felder aufzuteilen, um sie separat von der Handschrifterkennung auswerten zu lassen.

Anbindung der Online-Handschrifterkennung und Auswertung der Ergebnisse: Wie in Abb. 5.5 dargestellt handelt es sich bei der ausgewählten Handschrifterkennung um einen Onlinedienst, der mit Hilfe einer entsprechenden WEB-API angesprochen werden kann. Die Smartphone-Anwendung ist für die Anbindung des Dienstes sowie die korrekte Verarbeitung der zurückgelieferten Ergebnisse verantwortlich. Hierzu zählt z.B. die Zuordnung der erkannten Texte zu den einzelnen Feldern und Attributen im Datenmodell als Voraussetzung für die Synchronisation mit der angebundenen PMS.

Identifikation der Story-Card: Die Identifikation einer Story-Card wird wie bereits beschrieben innerhalb der Smartphone Anwendung durchgeführt. Alle auf den Vorlagen enthaltenen Identifikationsnummern sind als Konfiguration hinterlegt und fest mit einem der drei Typen (Epic, Story, Task) verknüpft. Entsprechend kann festgestellt werden, welcher Kartentyp gerade in Bearbeitung ist. Da die erfolgreiche Identifikation zwingende Voraussetzung für die Synchronisation mit der PMS ist, muss im Fehlerfall entsprechend Feedback an den Benutzer gegeben werden, um den Identifikationsvorgang ggf. wiederholen zu können. Weitere Details hierzu werden unter dem Punkt "Feedback an den Benutzer" erläutert.

Datenabgleich der Story-Cards mit der angebundener PMS: Die Smartphone Anwendung übernimmt selbstständig das Einfügen neuer User-Stories sowie das Aktualisieren existierender User-Stories in der PMS. Direkt nach der Identifikation einer Story-Card überprüft die Applikation, ob bereits eine entsprechende User-Story in der PMS existiert. Ist das der Fall werden alle dort hinterlegten Daten geladen und können direkt durch den Benutzer bearbeitet werden. Hierfür ist es notwendig, dass auch die Digital-Ink Daten in der PMS hinterlegt und bei Bearbeitung in die Smartphone-Anwendung geladen werden. Durch dieses Vorgehen ist eine vollständige Bearbeitung der gesamten handschriftlichen Inhalte zu jeder Zeit möglich. So können auch bereits existierende Textteile abgeändert werden (z.B. Durchstreichen bestimmter Textstellen, Ergänzen von Kommentaren, etc.). Da trotz der hohen Erkennungsrate der verwendeten Online-Handschrifterkennung Fehler zu erwarten sind, wird zusätzlich zu den digital erfassten Daten eine Pixelgrafik der Story-Card generiert und in der PMS

abgelegt. Diese kann im Bedarfsfall vom Anwender betrachtet werden, um falsch interpretierte Textstellen überprüfen zu können. Abb. 5.6 zeigt die Ansicht einer User-Story in der PMS Jira Agile, die mittels Digitalstift und Smartphone-Anwendung erstellt wurde.

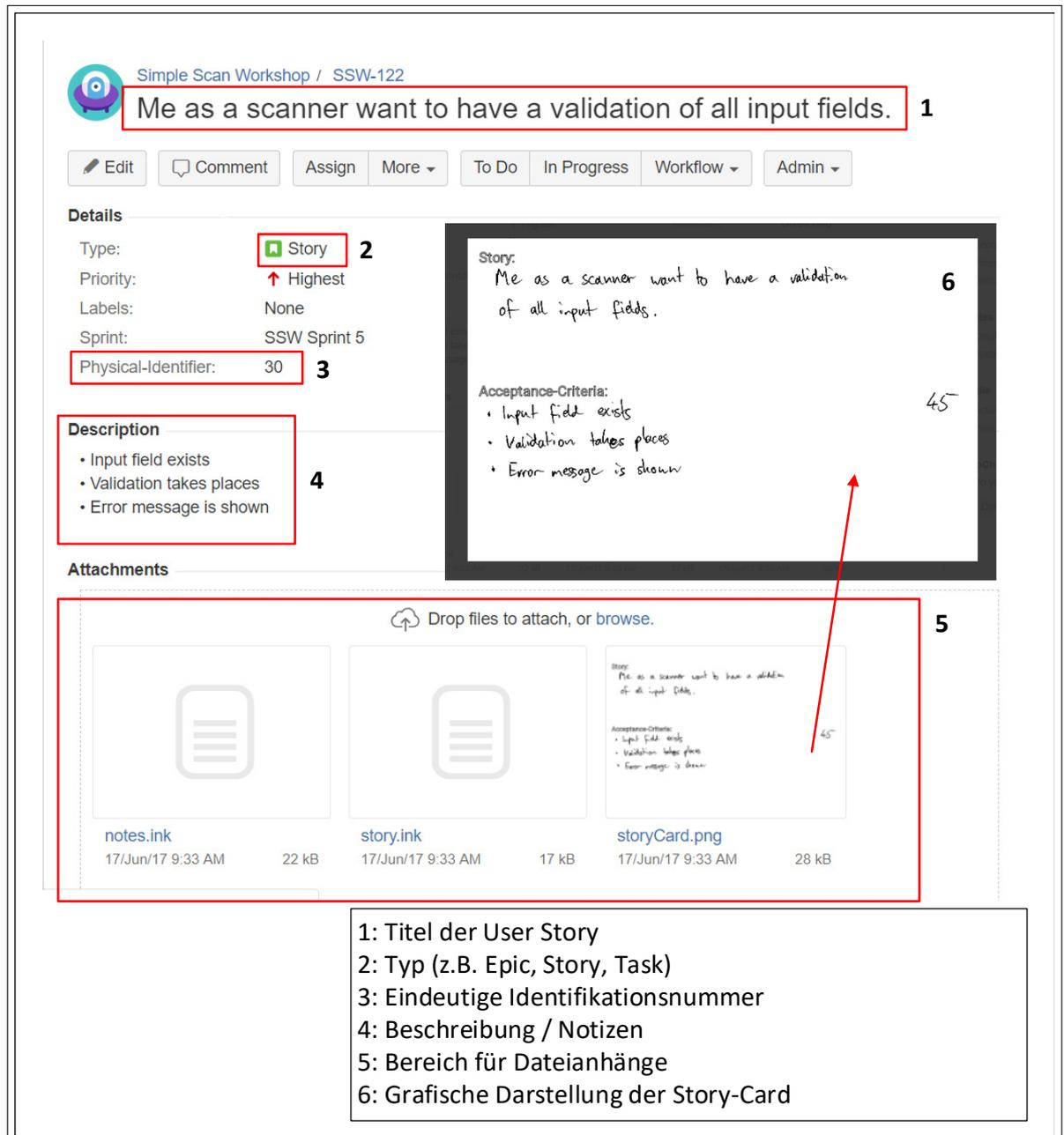


Abbildung 5.6: Ansicht einer User-Story in Jira-Agile

Feedback an den Benutzer: In Abb. 5.7 sind alle wesentlichen Elemente der grafischen Benutzerschnittstelle der Smartphone-Anwendung dargestellt. Die Benutzerschnittstelle dient dabei ausschließlich als Feedback-Schnittstelle, um den Kunden über die korrekte

Funktion des Systems und ggf. über Fehlfunktionen zu informieren.

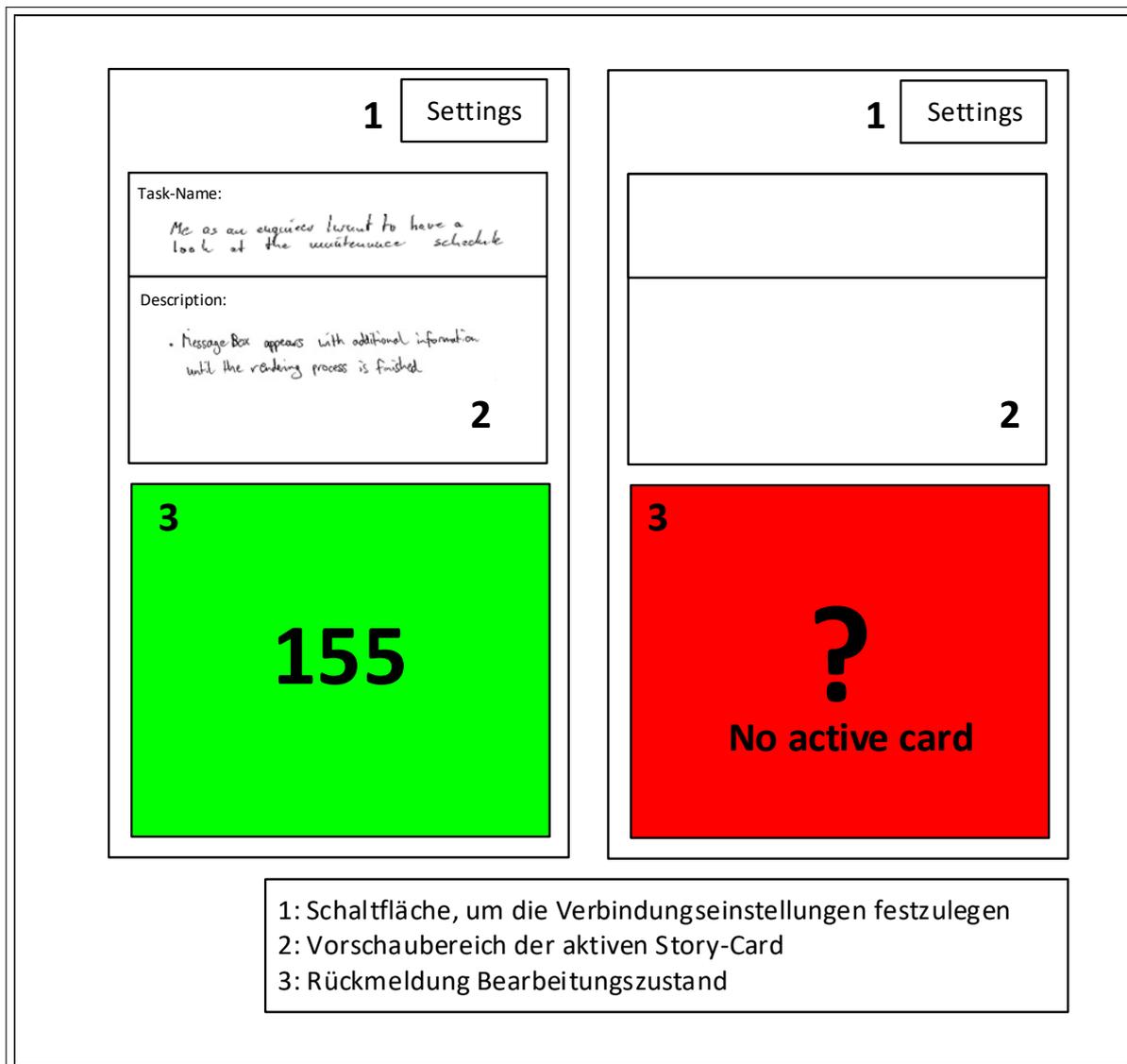


Abbildung 5.7: Konzeptvorlage zur Benutzerschnittstelle der Smartphone-Anwendung

Die Benutzerschnittstelle beinhaltet zwei wichtige Bereiche: Die obere Hälfte des Bildschirms zeigt den Inhalt der Story-Card, die gerade bearbeitet wird. Die Inhalte werden dabei parallel zum Schreibvorgang aktualisiert, um dem Anwender bei Bedarf zu zeigen, dass das System korrekt funktioniert. Die untere Hälfte zeigt dem Anwender welche Story-Card gerade bearbeitet wird. Nach korrekter Identifikation (durch das Nachzeichnen der Identifikationsnummer auf der Story-Card) wird der Hintergrund grün gefärbt und die Identifikationsnummer wird angezeigt (links dargestellt in Abb. 5.7) Der Benutzer kann nun mit der Bearbeitung der Inhalte beginnen. Wenn keine Story-Card identifiziert wurde bzw. aktiv ist wird der Hintergrund rot gefärbt (rechts dargestellt in Abb. 5.7).

Treten während der Identifikation Fehler auf (z.B. Identifikationsnummer wird nicht erkannt) wird der Hintergrund ebenfalls rot eingefärbt und zusätzlich eine aussagekräftige Fehlermeldung eingeblendet.

Auf die Anzeige des durch die HWR erkannten Textes zur Kontrolle durch den Anwender wurde bewusst verzichtet. Im Fokus der Projektarbeit stehen die Daten auf den Karteikarten und die ständige Kontrolle durch den Benutzer wäre ein zusätzlicher Schritt und damit ein Störfaktor im Arbeitsablauf. Stattdessen wurde die Kontrollmöglichkeit in die angebundene PMS ausgelagert. Mit Hilfe der dort abgelegten grafischen Darstellung aller erfassten Story-Cards (vgl. Abb. 5.6) kann der digital erfasste Text im Bedarfsfall mit der handschriftlichen Darstellung abgeglichen werden.

5.4.2 Positionserfassung und Identifikation von Story-Cards auf dem Task-Board

Der vorangehende Abschnitt beschreibt ein Konzept zur inhaltlichen Erfassung der User-Stories entsprechend den Anforderungen A.1 und A.2 mit Hilfe des Digitalstiftes, der geeigneten Story-Card Vorlagen und der vorgestellten Smartphone-Anwendung. In diesem Abschnitt steht nun die Erfassung des Task-Boards sowie des Product-Backlogs im Mittelpunkt. Bezugnehmend auf das Datenmodell (vgl. Abb. 5.1) stehen nun die Bearbeitungszustände, die Relationen zwischen den einzelnen Entitäten sowie die Prioritäten der einzelnen User-Stories im Fokus. Grundvoraussetzung der Datenerfassung ist die automatisierte Identifikation aller Story-Cards auf dem Task-Board sowie die Positionserfassung jeder einzelnen Karte. Alle weiteren Daten können durch geeignete Interpretation der einzelnen Positionen errechnet werden.

Zur technischen Realisierung wurde im Rahmen dieser Arbeit auf eine kamera-basierte Positionserkennung mit Hilfe optischer Markierungen (nachfolgend Marker genannt) auf den Story-Cards gesetzt.

5.4.2.1 Einführung Marker-Tracking

Die Technologie Marker-Tracking wird im Bereich *Augmented Reality* (AR) (dt. Erweiterte Realität) bereits seit Jahren erfolgreich eingesetzt [66, 87, 45]. Je nach verwendetem System werden im Regelfall quadratische Markierungen eingesetzt, die z.B. auf Papier gedruckt und anschließend auf physikalischen Gegenständen angebracht werden. Der relevante Bereich wird mit Hilfe einer Kamera erfasst. Die zugehörige Software identifiziert auf Basis des Videostreams vorhandene Marker und deren Position bzw. Ausrichtung im dreidimensionalen Raum. In typischen AR-Anwendungen werden dann virtuelle Objekte relativ zu den erfassten Markerpositionen in das Videobild eingefügt, um den Anwendern den Eindruck einer digitale erweiterten Realität zu vermitteln.

Für das Lösungskonzept Virtual Task-Board soll diese Art der Positionserkennung ebenfalls eingesetzt werden. Hierbei spielt die Ausrichtung eines Markers im dreidimensionalen Raum eine untergeordnete Rolle, da lediglich die Position der Story-Cards auf dem zweidimensionalen Task-Board benötigt wird. Allerdings ist die eindeutige Identifizierung eines Markers und damit der Story-Card eine zusätzliche Anforderung zur Umsetzung des

Virtual Task-Board Konzepts. Gängige AR-Frameworks bieten hierfür die Möglichkeit generische Informationen (z.B. eine Identifikationsnummer) im Marker zu kodieren. Abb. 5.8 zeigt vier Beispiele entsprechender Marker sowie die zugehörige Identifikationsnummer:

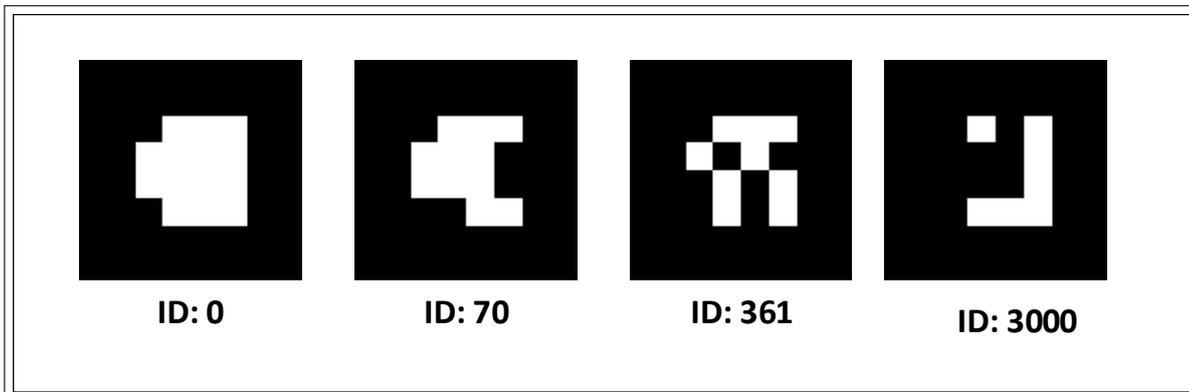


Abbildung 5.8: Beispiele für AR-Marker mit kodierter Identifikationsnummer

Die dargestellten Marker sind Beispiele des AR-Frameworks ARToolkit [3]. Innerhalb des quadratischen Rahmens wird bei diesem Markertyp eine 4x4 Matrix aus Binärfeldern (schwarz/weiß) zur Kodierung aller notwendiger Informationen wie z.B. der Identifikationsnummer verwendet. Mit diesem Marker-Typ lassen sich über 8000 unterschiedliche Identifikationsnummern darstellen. Da ein entsprechender AR-Marker auf jeder Story-Card abgedruckt ist, wäre auch die in 5.4.1.1 beschriebene Identifikation der Story-Card durch einfaches Abfotografieren mit der Smartphone-Anwendung technisch möglich. Auf diese Option wurde jedoch verzichtet, da die Interaktion mit der Smartphone-Anwendung auf ein Minimalmaß beschränkt werden soll (vgl. Anforderung C.6 in 5.2.2). Durch das Nachzeichnen mittels Digital-Stift kann dieser Schritt umgangen werden.

5.4.2.2 Einsatz von AR-Markern im Virtual Task-Board Konzept

In diesem Absatz wird der Einsatz der AR-Marker im Rahmen des Virtual Task-Board Konzepts näher erläutert. Die Abb. 5.9 und 5.10 zeigen die Anwendung der AR-Marker beispielhaft am Product-Backlog-Chart und am Task-Board. Das Konzept erlaubt eine freie Anpassung der Task-Board Konfiguration an die individuellen Anforderungen der jeweiligen Projektsituation. So können jederzeit Bearbeitungszustände und weitere Zeilen ergänzt bzw. angepasst werden. Zusätzlich zeigt Abb. 5.12 erneut die Vorlage einer Story-Card (am Beispiel "Task"). Die Vorlage wurde um einen Marker mit der entsprechenden Identifikationsnummer ergänzt. Auf diese Art kann die Zuordnung einer bestimmten Story-Card zu den einzelnen Spalten bzw. Zeilen am Task-Board realisiert werden.

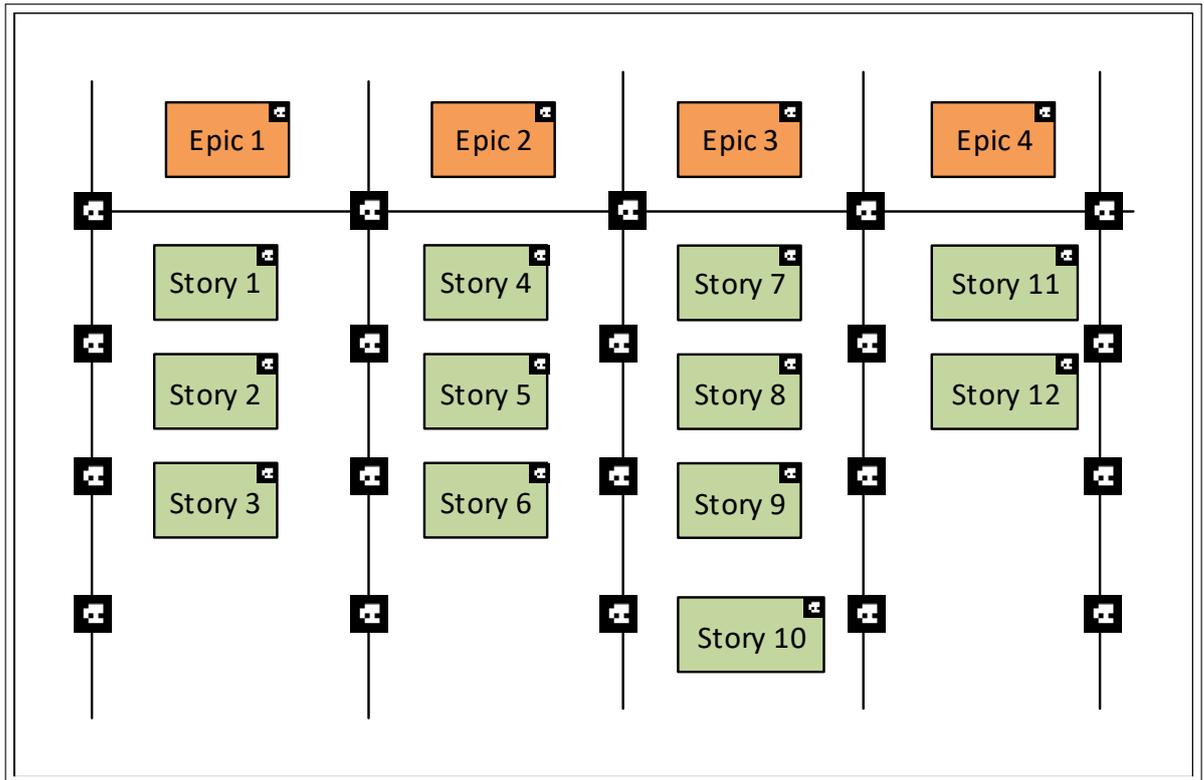


Abbildung 5.9: Einsatz der AR-Marker am Product-Backlog-Chart

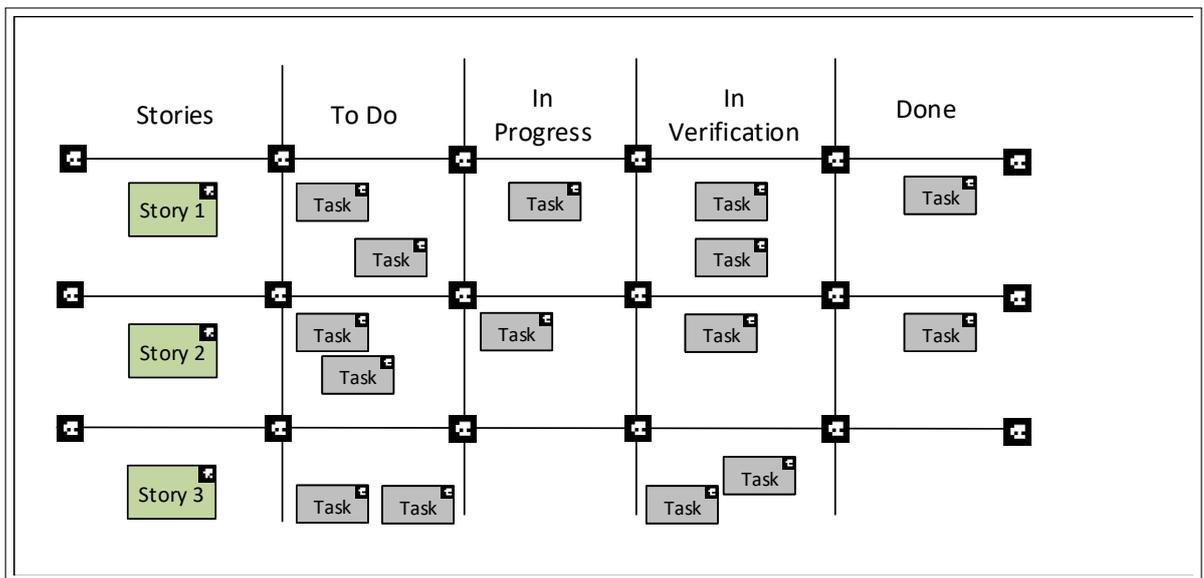


Abbildung 5.10: Einsatz der AR-Marker am Task-Board

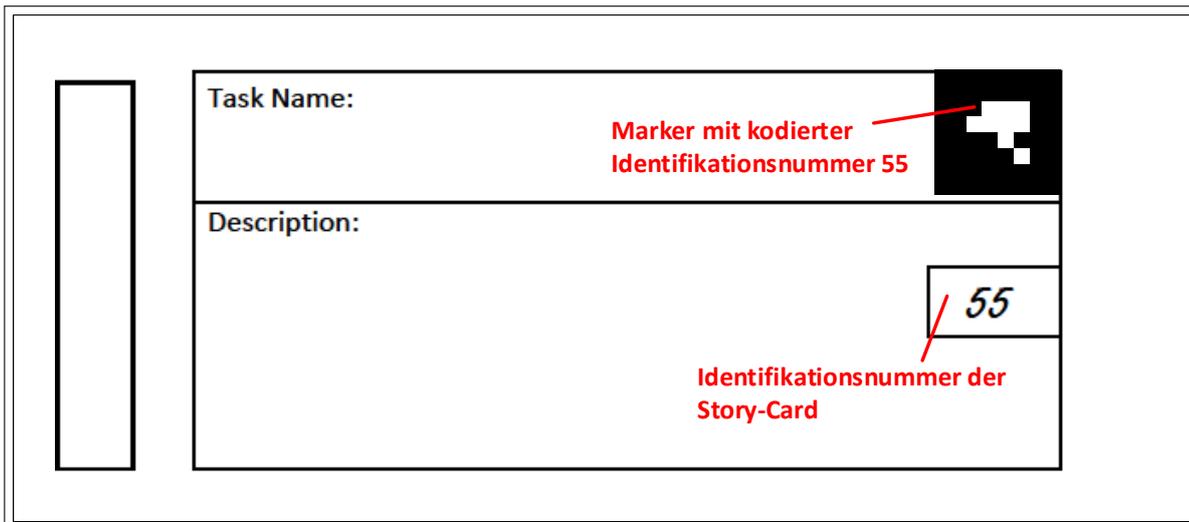


Abbildung 5.11: Vorlage einer Story-Card mit Identifikationsnummer und Marker

In den folgenden Absätzen werden die Konzeptdetails zur Erfassung relevanter Daten gem. Datenmodell (vgl. Abb. 5.1) sowie die korrekte Interpretation der Daten erläutert.

Erfassung und Identifikation aller Story-Cards am Task-Board und am Product-Backlog-Chart: Grundvoraussetzung der Datenerfassung ist die korrekte Identifikation aller Story-Cards am Task-Board (vgl. 3.1.3.3) und am Product-Backlog-Chart (vgl. 3.1.3.2). Hierfür muss jeweils eine hochauflösende Kamera für Product-Backlog und Task-Board eingesetzt werden. Die Datenerfassung ist zeitlich unkritisch, weshalb eine geringe Aufnahmefrequenz (z.B. alle 30 Sekunden ein Bild) ausreichend ist. Mittels einer geeigneten Marker-Tracking-Software (in vorliegendem Fall AR-Toolkit [3]) werden alle in den Kamerabildern enthaltenen Marker identifiziert und lokalisiert, um den Inhalt jeder Zeile/Spalte am Task-Board festzustellen. Dabei ist zu beachten, dass jederzeit einzelne Marker durch Personen verdeckt sein können. Entsprechende Kameraaufnahmen müssen demnach erkannt und verworfen werden, da die erfassten Daten sonst unvollständig sein könnten. Zu diesem Zweck wird ein einfaches Konzept eingesetzt: Alle am Task-Board dargestellten Zeilen und Spalten sind durch entsprechende Marker gekennzeichnet. Die Identifikationsnummern dieser Marker sind statisch durch Konfiguration im System hinterlegt. Nach jeder neuen Bildaufnahme wird nun überprüft, ob diese Marker vollständig sichtbar sind. Andernfalls wird davon ausgegangen, dass gerade Personen am Task-Board arbeiten und die Inhalte teilweise verdeckt sind. Als Folge wird das entsprechende Bild nicht weiter ausgewertet und verworfen. Abb. 5.10 zeigt deutlich, dass sich aus den vorkonfigurierten Markern ein enges Netz ergibt, das es nahezu unmöglich macht eine Story-Card zu verdecken, ohne dass der eben beschriebene Mechanismus greift. Am Product-Backlog-Chart kann dieses Konzept analog umgesetzt werden.

Abb. 5.12 zeigt das Prinzip noch einmal auf. Die Komponente "Marker-Tracking" beinhaltet die eben beschriebenen Funktionen. Es ist entsprechend ein eigene Instanz

der Komponente für die Erfassung des Task-Boards und des Product-Backlogs notwendig. Die Konfiguration der statischen Marker zur Markierung der Zeilen bzw. Spalten ist hinterlegt, um ungültige Aufnahmen mit verdeckten Story-Cards (z.B. verursacht durch Personen) filtern zu können. Die so erfassten AR-Marker (Position und Identifikationsnummer) werden anschließend an die Komponente "Interpretation und Synchronisation" weitergegeben. Eine detaillierte Beschreibung zur technischen Umsetzung folgt in 5.7.2.

Interpretation des Product-Backlog-Charts und des Task-Boards: Durch das eben aufgezeigte Konzept ist bekannt, welche Story-Cards sich in den jeweiligen Spalten und Zeilen des Product-Backlog-Charts bzw. des Task-Boards befinden. Ebenso kann aufgrund der Identifikationsnummer festgestellt werden, zu welchem Typ des Ebenenmodells (vgl. 3.2) jede Story-Card gehört. Diese Daten müssen nun entsprechend dem zugrundeliegenden Datenmodell (vgl. Abb. 5.1) interpretiert werden, um auch die zugehörigen Relationen zu ermitteln. Im Fall des Product-Backlogs ist das relativ einfach möglich: Die Gesamtheit aller auf dem Product-Backlog-Chart angebrachten Story-Cards vom Typ "Epic" definieren den Gesamtumfang des Product-Backlogs. Die in derselben Spalte angebrachten Story-Cards vom Typ "Story" können dann jeweils dem richtigen Epic zugeordnet werden. Die Attribute "Akzeptanzkriterium" und "Beschreibung" wurden bereits durch den Einsatz der Smartphone Anwendung erfasst und können mittels der Identifikationsnummer entsprechend zugeordnet werden.

Die Inhalte am Task-Board können in ähnlicher Weise interpretiert werden. Die Gesamtheit aller am Task-Board dargestellten Inhalte repräsentieren das aktuelle Sprint-Backlog. Die Relation zwischen Story und Task kann zeilenweise festgelegt werden. Der Bearbeitungszustand aller Tasks wird durch die Zugehörigkeit zu den entsprechenden Spalten am Task-Board festgelegt. Die Bearbeitungszustände von Stories und Epics sind jeweils durch die Bearbeitungszustände der zugehörigen Tasks definiert.

Die in Abb. 5.12 dargestellte Komponente "Interpretation und Synchronisation" enthält die beschriebenen Funktionen. Sie übernimmt die Interpretation nach dem eben erläuterten Konzept und aktualisiert die Daten in der angebundenen PMS entsprechend. Details zur technischen Umsetzung folgen in 5.7.3.

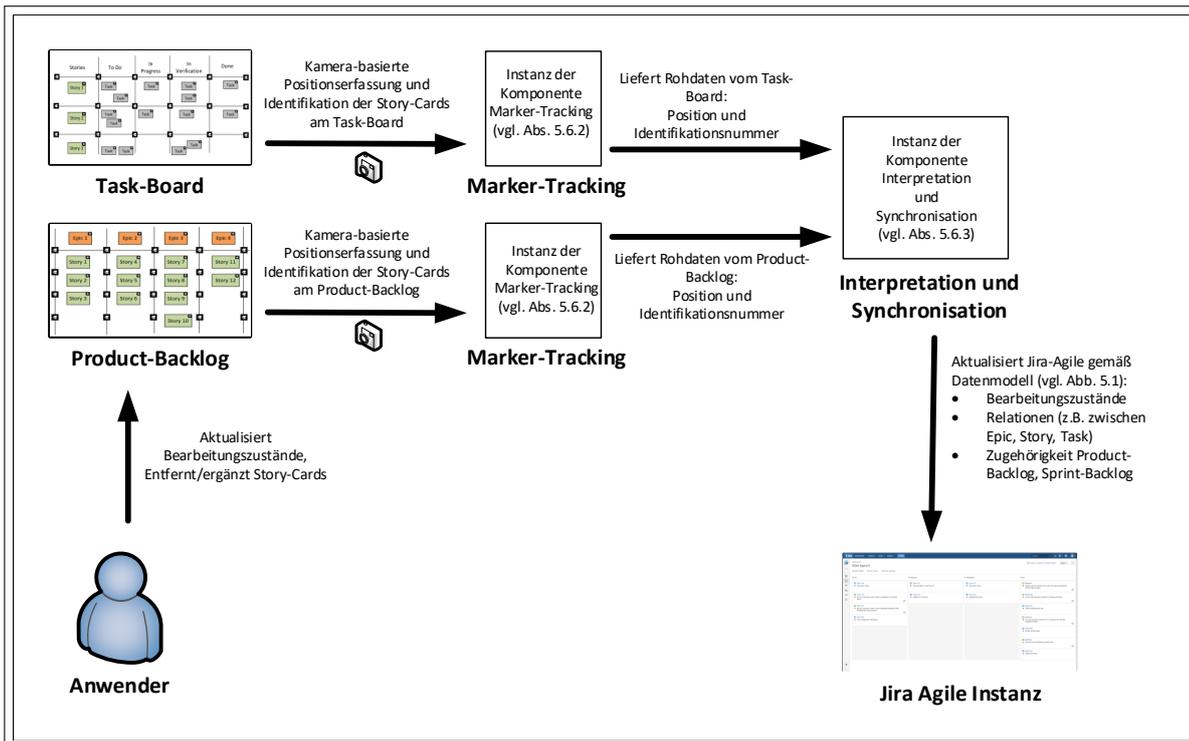


Abbildung 5.12: Konzeptübersicht zur Erfassung, Interpretation und Synchronisation der Projektdaten auf Product-Backlog und Task-Board

5.5 Lösungsidee zur Anbindung konventioneller PMS

Die ortsunabhängige Bereitstellung relevanter Projektdaten ist eines der Hauptziele des Virtual Task-Board Konzepts (vgl. Anforderungen B.1 - B.7). Dieses Ziel lässt sich durch die Übertragung der relevanten Daten des Task-Boards und des Product-Backlog-Charts in eine konventionelle PMS erfüllen. Hierbei ist es ausreichend, dass die PMS nur lesend-also als Informationsquelle- verwendet wird. Dieser Ansatz unterstützt gleichzeitig die nichtfunktionalen Anforderungen C.1, C.2, C.5, und C.6, die im Wesentlichen alle einen möglichst geringen Einsatz digitaler Benutzerschnittstellen fordern. Deshalb gilt im Rahmen des Virtual Task-Board Konzepts der Grundsatz, dass die Synchronisierung der Daten nur unidirektional in Richtung PMS stattfindet. Entsprechend werden die auf den physikalischen Medien (Task-Board, Product-Backlog-Chart) dargestellten Daten stets als korrekt angenommen und in der PMS abgebildet.

In den Absätzen 5.4.1 zur Erfassung handschriftlicher Inhalte und 5.4.2.2 zur Erfassung der auf dem Task-Board bzw. auf dem Product-Backlog-Chart dargestellten Inhalte wurde bereits deutlich, dass es je Projektteam zwei unterschiedliche Instanzen gibt (Smartphone-Anwendung, Server-Anwendung zur Synchronisation Datenerfassung auf Task-Board und Product-Backlog-Chart), die Informationen zur PMS senden. Beide Instanzen arbeiten auf Basis derselben eindeutigen Identifikationsnummern, welche für die Verknüpfung der

jeweiligen Inhalte notwendig ist. Die Eindeutigkeit der Identifikationsnummern muss durch die Druckvorlagen (vgl. 5.12) sichergestellt werden.

5.6 Überlegungen zum Systemaufbau

Die individuellen Anforderungen an ein Task-Board können je nach Projektteam stark unterschiedlich ausfallen, da agile Methoden als Rahmenwerk zu verstehen sind und viele Prozessdetails an die individuellen Erfordernisse der jeweiligen Projektsituation angepasst werden müssen. Das Virtual Task-Board Konzept muss deshalb ausreichend flexibel an die individuellen Prozessanforderungen anpassbar sein. Folgende Überlegungen und Arbeitsschritte müssen für einen individuell angepassten Aufbau erledigt werden:

1. **Grundgerüst Task-Board:** Zunächst muss festgelegt werden, wie viele Spalten (entsprechend der Bearbeitungszustände) und Zeilen (entsprechend der maximalen Anzahl an User-Stories in einem Sprint) erforderlich sind, um die Gesamtzahl an benötigten Markern zur Darstellung des Task-Boards zu ermitteln. Das in Abb. 5.10 gezeigte Beispiel (3 Zeilen, 5 Spalten) benötigt entsprechend 18 Marker zur Abbildung des Grundgerüsts. Anschließend müssen die Marker gedruckt und an einer entsprechend großen Fläche (z.B. Zimmerwand) gem. dem in Abb. 5.10 gezeigten Prinzip angebracht werden (z.B. mit doppelseitigem Klebeband). Die Spalten können dann beispielsweise mit normalen Karteikarten beliebig beschriftet werden.
2. **Überlegungen zum Kamerasetup:** Die verwendete Kamera sollte rein aus praktischen Gründen einen festen Installationsstandort erhalten, an dem sie für die gesamte Projektlaufzeit verbleiben kann (z.B. Deckenhalterung). Dabei ist darauf zu achten, dass die Kamera alle AR-Marker erfasst, die zur Markierung der Zeilen und Spalten verwendet wurden. Eine separate Kalibrierung von Kamera und Software ist nicht notwendig. Während der Versuchsphase hat sich gezeigt, dass jeder Marker durch mind. 65 x 65 Pixel vom Kamerasensor erfasst werden muss, um eine reibungslose Erkennung zu gewährleisten. Zur genauen Auswahl, Positionierung und Einstellung der Kamera soll an dieser Stelle keine Aussage gemacht werden, da diese Thematik nicht im Fokus der Arbeit steht. Dennoch können die Angaben aus der in Kap. 6 beschriebenen Evaluationsstudie eine erste Orientierung bieten. Im Rahmen der Studie kam das Kameramodell X-T2 von Fujifilm mit einer Auflösung von 20.25 Megapixeln (6000px x 3376px) zum Einsatz. Die Marker zur Festlegung des Task-Boards hatten eine Seitenlänge von 3,5cm. Auf Basis der genannten Auflösung und Markergröße wäre theoretisch die Erfassung eines Task-Boards mit ca. 3.5m Breite und 2m Höhe denkbar. Diese Überlegungen sind allerdings eher theoretischer Natur und hängen stark von der jeweils eingesetzten Kamera ab. Abhängig von Position und Brennweite der Kamera muss der Bildausschnitt optimal gewählt werden können, weshalb in der Praxis von einer etwas geringeren Größe des Task-Boards ausgegangen werden sollte. Zur Unterstützung größerer Task-Boards ist der Einsatz mehrerer Kameras zu emp-

fehlen. Hierbei muss lediglich darauf geachtet werden, dass im Überlappungsbereich zweier Kameras mindestens ein AR-Marker enthalten ist, der zur Synchronisation der Koordinatensysteme verwendet werden kann. Der in 5.7 beschriebene Prototyp wurde zur Unterstützung mehrerer Kameras bereits vorbereitet. Die Funktion wurde zur Durchführung der Studie jedoch nicht benötigt und nicht im Detail getestet.

3. **Konfiguration der Virtual Task-Board Software:** Die Virtual Task-Board Software benötigt eine Reihe an Konfigurationsparametern, die einmalig für ein Projekt hinterlegt werden müssen. Alle AR-Marker, die zum Aufbau des Grundgerüsts (Zeilen und Spalten) verwendet wurden müssen hinterlegt werden. Hierzu zählt auch die Definition der verwendeten Bearbeitungszustände sowie deren Zuweisung zu den einzelnen Spalten. Weiterhin müssen einige wenige Konfigurationsparameter bzgl. der Verbindung zur PMS hinterlegt werden (z.B. IP-Adresse, Port, Projektname, etc.). Weitere Details zu diesem Punkt werden in Kap. 5.7 erläutert.
4. **Vorbereitung der Story-Cards:** Ein Vorteil der Virtual Task-Board Lösung ist, dass die benötigten Story-Cards selbst gedruckt werden können. Hierfür wurde bereits im Rahmen dieser Arbeit ein entsprechendes Werkzeug entwickelt, das die Erstellung der Druckvorlagen gem. Abb. 5.4 entsprechend der individuellen Mengenanforderungen übernimmt.

Dieser Abschnitt hatte das Ziel, einen Überblick der erforderlichen Vorbereitungsmaßnahmen zur Erstellung eines individuellen Projektsetups unter Einsatz der Virtual Task-Board Lösung zu geben. Die genannten Aufgaben müssen aktuell händisch vorgenommen werden. Für den produktiven Einsatz des Systems müsste entsprechend ein softwaregestützter Prozess zur Durchführung dieser Aufgaben angeboten werden, um die einzelnen Schritte verständlich und effizient zu gestalten. Hierzu könnte beispielsweise die Konfiguration des Task-Boards, die Generierung der erforderlichen Konfigurationsdateien sowie die Ausleitung fertiger Druckvorlagen zur Erstellung des Grundgerüsts am Task-Board und der Story-Cards zählen. Dieser Aspekt steht jedoch nicht im Fokus der in 1.3 definierten Forschungsziele und wird deshalb nicht weiter betrachtet.

5.7 Realisierung eines Prototypen

Dieser Abschnitt beschreibt die Umsetzung eines Prototypen, der als Evaluationsgrundlage (vgl. Kap. 6) dienen soll. Bzgl. Reifegrad und Funktionsumfang wird ein sog. Pilotsystem nach Balzert et al. [5] umgesetzt. Ein Pilotsystem bezeichnet Prototypen mit sehr hohem Reifegrad und Funktionsumfang. Es erlaubt bereits die Einbindung der Endanwender und bildet die Kernfunktionen des geplanten Produkts ab. Für den vorliegenden Fall sollen deshalb folgende Komponenten und Funktionen umgesetzt werden:

1. Komponente "Smartphone-Anwendung" zur Realisierung der in 5.4.1 beschriebenen Konzepte. Im Einzelnen handelt es sich um folgende Funktionsumfänge:

- Erfassung handschriftlicher Inhalte mittels Digitalstift und Online-Handschrifterkennung
 - Identifizierung der Story-Card mittels Digitalstift
 - Erfassung vordefinierter Felder auf der Story-Card (gem. Datenmodell (vgl. Abb. 5.1))
 - Umsetzung der grafischen Benutzerschnittstelle (gem. Abb. 5.7)
 - Erstellen und Bearbeiten von User-Stories durch den Anwender
 - Datenübertragung zur PMS
2. Komponente "Marker-Tracking" zur Realisierung der in 5.4.2 beschriebenen Konzepte. Im Einzelnen handelt es sich um folgende Funktionsumfänge:
- Identifikation aller am Task-Board vorhandenen AR-Marker
 - Positionserfassung aller am Task-Board vorhandenen AR-Marker
 - Erkennung verdeckter Inhalte am Task-Board
3. Komponente "Dateninterpretation und Synchronisation" zur Realisierung der in 5.4.2.2 beschriebenen Konzepte. Im Einzelnen handelt es sich um folgende Funktionsumfänge:
- Erfassung aller am Task-Board dargestellten Spalten und Zeilen
 - Erfassung und Identifikation aller am Task-Board vorhandenen Story-Cards
 - Erfassung der Bearbeitungszustände aller User-Stories und Tasks
 - Erfassung des aktuellen Sprint-Backlogs
 - Synchronisation aller erfassten Daten in die PMS Jira Agile
4. Sonstige Hilfsmittel:
- Programm zur Generierung von Druckvorlagen für die jeweiligen Story-Cards

Auf die Erfassung des Product-Backlogs wurde im Rahmen der prototypischen Implementierung und damit auch in der Evaluationsstudie (vgl. Kap. 6) verzichtet. Für den Nachweis der technischen Machbarkeit ist eine Umsetzung nicht notwendig, da die Lösung analog zur Erfassung des Task-Boards funktioniert. Im Rahmen der Evaluationsstudie wurde ein Product-Backlog-Board eingesetzt - jedoch nicht digital erfasst.

5.7.1 Technische Beschreibung der Komponente "Smartphone-Anwendung"

Die Smartphone Anwendung wurde auf Basis von Android 7.0 (API Level 24) aufgebaut. Zusätzlich wurden folgende Bibliotheken eingebunden:

Equil Android SDK (Version 2015.5): Diese Bibliothek wird von der Firma Equil bereitgestellt und enthält die notwendigen Funktionen, um die Rohdaten via Bluetooth aus dem Digitalstift auszulesen.

Volley (Version 1.0.0): Hierbei handelt es sich um einen gängigen HTTP-Client für Android, der speziell den Umgang mit asynchronen HTTP-Requests erleichtert.

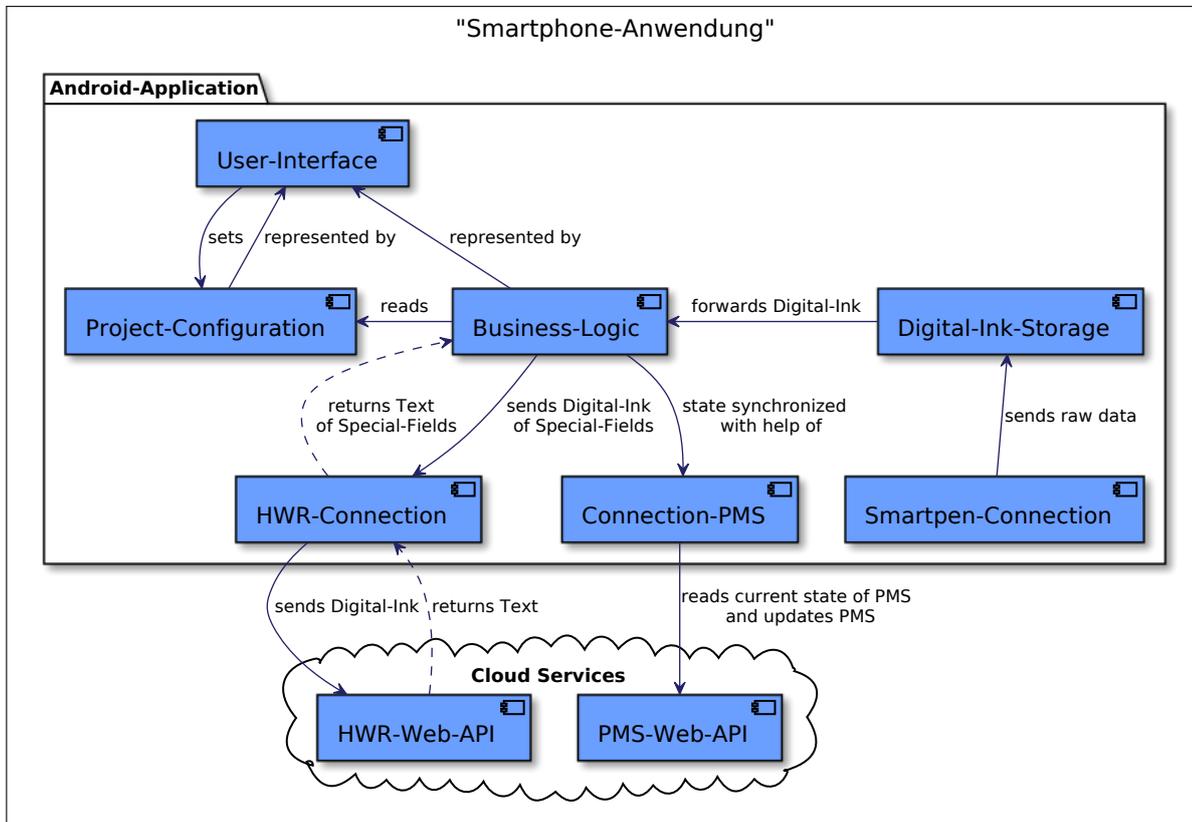


Abbildung 5.13: Vereinfachtes Komponentendiagramm der Smartphone-Anwendung.

Abb. 5.13 zeigt eine stark vereinfachte Komponentenübersicht der Smartphone-Anwendung. Die Komponente *User-Interface* implementiert die in 5.7 konzipierte Benutzerschnittstelle der Anwendung. Zusätzlich ermöglicht sie die Einstellung der wichtigsten Projektparameter, wie z.B. die URL der PMS und die Auswahl des zu bearbeitenden Projekts. Abb. 5.14 zeigt Screenshots der prototypisch implementierten Smartphone-Anwendung. Die Komponente *Project-Configuration* dient lediglich zur Verwaltung bzw. persistenten Speicherung der konfigurierten Projektparameter.

Zentrale Bedeutung kommt der Komponente *Business-Logic* zu. Wichtigste Aufgabe dieser Komponente ist die Auswertung der auf den Story-Cards vordefinierten Felder. Die Grundidee hierfür sind feste Bereiche auf den Story-Cards, um eine entsprechende Interpretation der handschriftlichen Inhalte gem. Datenmodell zu ermöglichen (vgl. 5.1). Die *Business-Logic* erhält die Digital-Ink Daten der entsprechenden Felder vom *Ink-Storage* und gibt sie separat an die Komponente *HWR-Connection* weiter. Diese sendet

die Digital-Ink Daten entsprechend zur angebundenen Online-Handschrifterkennung (*HWR-Web-API*). Ergebnis ist maschinell lesbarer Text aller vordefinierter Felder. Diese Daten werden in zyklischen Intervallen Richtung PMS gesendet, um die dort vorhandenen Informationen zu aktualisieren. Die Komponente *Connection-PMS* stellt hierfür die entsprechenden Schnittstellen auf Basis der von der PMS angebotenen REST-Schnittstelle (*PMS-Web-API*) bereit.

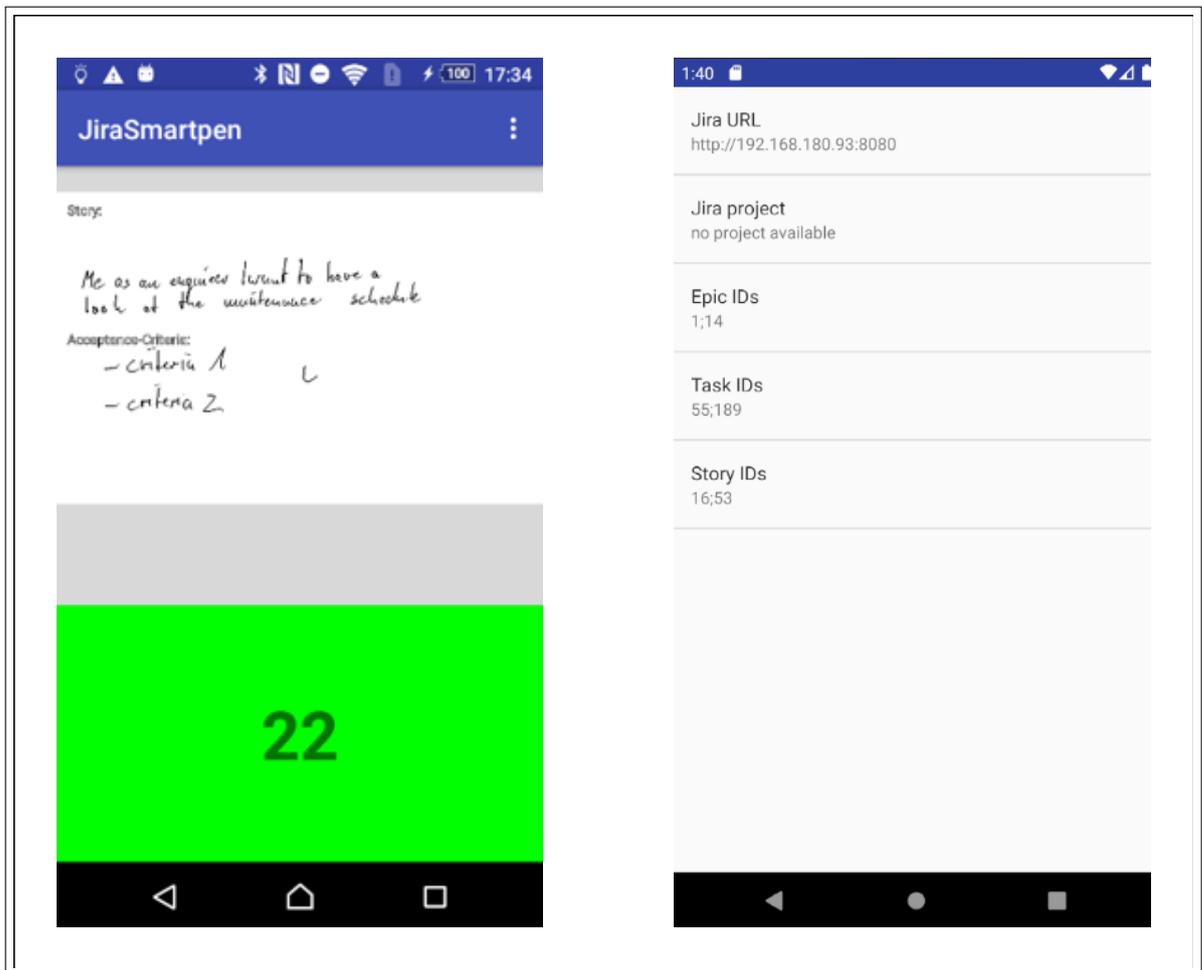


Abbildung 5.14: Screenshots der Smartphone-Anwendung
Links: Benutzerschnittstelle gem. Konzeptvorlage (vgl. Abb. 5.7), Rechts:
Einstellungsmenü

5.7.2 Technische Beschreibung der Anwendung "Marker Tracking"

Die Softwareanwendung "Marker Tracking" liefert die Rohdaten (Identifikationsnummer und Position) aller am Task-Board eingesetzten AR-Marker. Jede im Gesamtaufbau eingesetzte Kamera benötigt eine aktive Instanz, um die Anwendung "Dateninterpretation und Synchronisation" (vgl. 5.7.3) mit Daten zu versorgen. Da im Rahmen der prototypischen Implementierung nur das Task-Board automatisiert erfasst wurde, war der

Einsatz einer Kamera ausreichend. Die Implementierung ist aber bereits in der Lage auch weitere Kameras zu unterstützen. Technische Grundlage der Anwendung ist Unity3D (Version 2019.2). Dabei handelt es sich um eine Laufzeit- und Entwicklungsumgebung, die vorwiegend zur Programmierung von Spielen und AR-Anwendungen eingesetzt wird. Zusätzlich wurden folgende Erweiterungen bzw. Bibliotheken in Unity3D eingebunden:

ARToolkit5-Unity: (Version 5.3.2) Diese Bibliothek stellt Funktionen zur Erfassung von AR-Markern in Unity3D bereit.

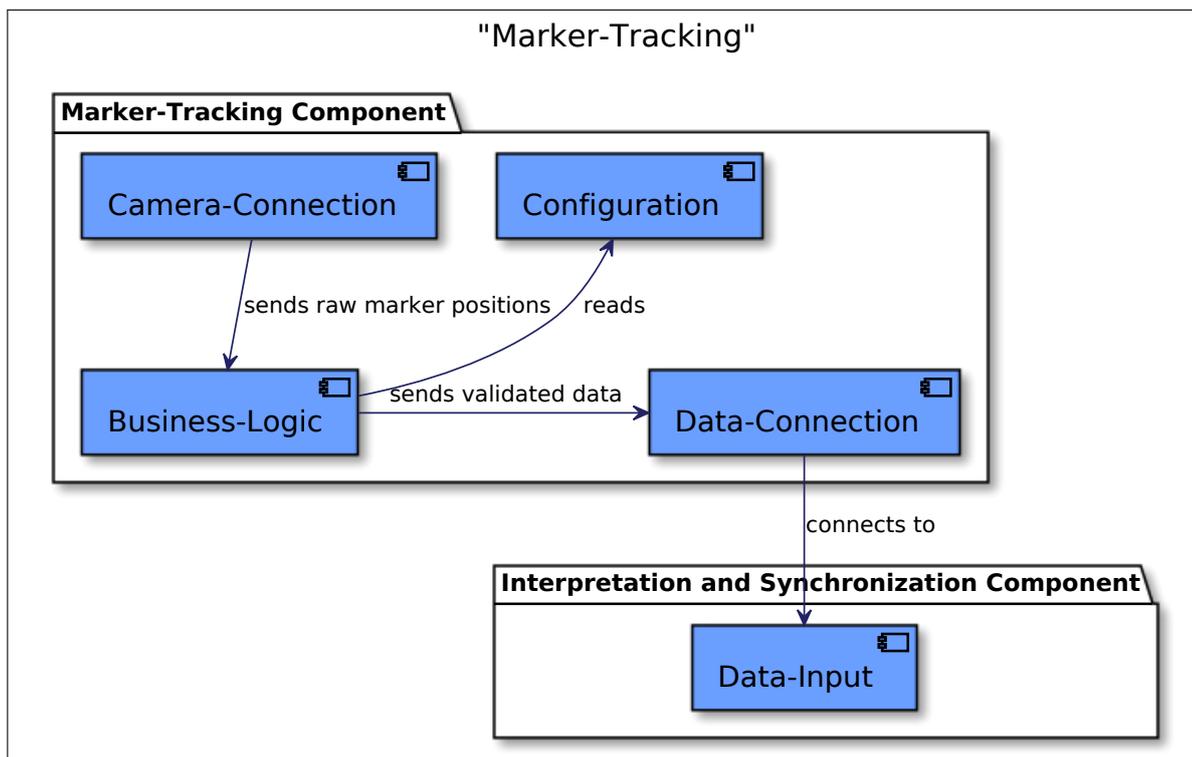


Abbildung 5.15: Vereinfachtes Komponentendiagramm der Komponente "Marker Tracking".

Abb. 5.15 zeigt den internen Aufbau der Anwendung "Marker Tracking". Dabei dient die Komponente *Camera-Connection* zur Übertragung der aktuellen Aufnahmen von der Kamera zur Applikation. Im Versuchsaufbau wurde das Modell XT-2 von Fuji verwendet. Ausgelöst wurde die Kamera mit Hilfe einer integrierten Zeitauslösefunktion (alle 2 Minuten). Die Bilder wurden mittels WIFI automatisiert abgerufen und in einem lokalem Verzeichnis bereitgestellt. Das Modul *Business Logic* ist für die Verarbeitung der dort abgelegten Bilder verantwortlich. Grundlage der Bildverarbeitung ist die Erweiterung AR-Toolkit (vgl. 5.4.2.2). Sie liefert die Positionen und Identifikationsnummern aller im jeweiligen Bild enthaltenen Marker zurück. Diese Daten werden anschließend auf Vollständigkeit geprüft, um unvollständige Bildaufnahmen (z.B. durch Personen verdeckte Marker) auszusortieren. Das Konzept hierfür wurde bereits in 5.4.2.2 beschrieben

und ist ebenfalls in der Komponente *Business-Logic* umgesetzt. Die Komponente *Configuration* hält lediglich statische Parameter vor, die zur Anpassung der Anwendung an die jeweilige Projektsituation notwendig sind. Alle Parameter können einfach mit Hilfe einer JSON-Datei angepasst werden. Eine detaillierte Beschreibung der einzelnen Konfigurationsparameter findet sich im Anhang unter B.2.1. Zudem befindet sich die Beispielkonfiguration der Evaluationsstudie (vgl. Kap. 6) unter B.2.2. Die Komponente *Data-Connection* ist lediglich für den Verbindungsaufbau zur Anwendung "Interpretation und Synchronisation" notwendig. Zur Datenübertragung werden die relevanten Informationen (z.B. Position, Identifikationsnummer, etc.) mittels JSON-Format serialisiert und via Socket-Verbindung übertragen. Abb. B.5 zeigt beispielhaft das verwendete Übertragungsformat.



Abbildung 5.16: Screenshot der IDE Unity3D mit der Applikation "Marker Tracking". Die Oberfläche zeigt ein prototypisches Task-Board mit den erkannten Identifikationsnummern der vorhandenen AR-Marker.

5.7.3 Technische Beschreibung der Komponente "Interpretation und Synchronisation"

Die Anwendung "Interpretation und Synchronisation" wurde als einfache Konsolenanwendung auf Basis des .NET-Frameworks (Version 4.6.1) umgesetzt. Wie zu Anfang des Kapitels (vgl. 5.7) beschrieben, zählt die korrekte Interpretation der am Task-Board erfassten Rohdaten zu den Kernaufgaben dieser Komponente. Weiterhin werden die interpretierten Projektdaten zur angebundenen PMS (in diesem Fall Jira Agile) übertragen. Folgende Bibliotheken wurden zur Realisierung der Anwendung verwendet:

Atlassian.SDK (Version 9.2.4): Diese Bibliothek vereinfacht den Zugriff auf die Jira Agile Web API stark. Sie bietet eine Programmierschnittstelle auf Basis von *Language Integrated Query* (LINQ). Dabei handelt es sich um ein .NET -Konzept, das vergleichbar mit *Structured Query Language* (SQL) ist und einfachen Zugriff auf Daten mit Hilfe einer standardisierten Sprache erlaubt.

CommandLineParser (Version 1.9.71): Diese Bibliothek erleichtert die Verwendung eines *Command-Line Interface* (CLI). Insbesondere das Einlesen der Kommandozeilenparameter wird vereinfacht.

RestSharp (Version: 105.2.3): Diese Erweiterung erleichtert die Verwendung von *Representational State Transfer* (REST) Schnittstellen.

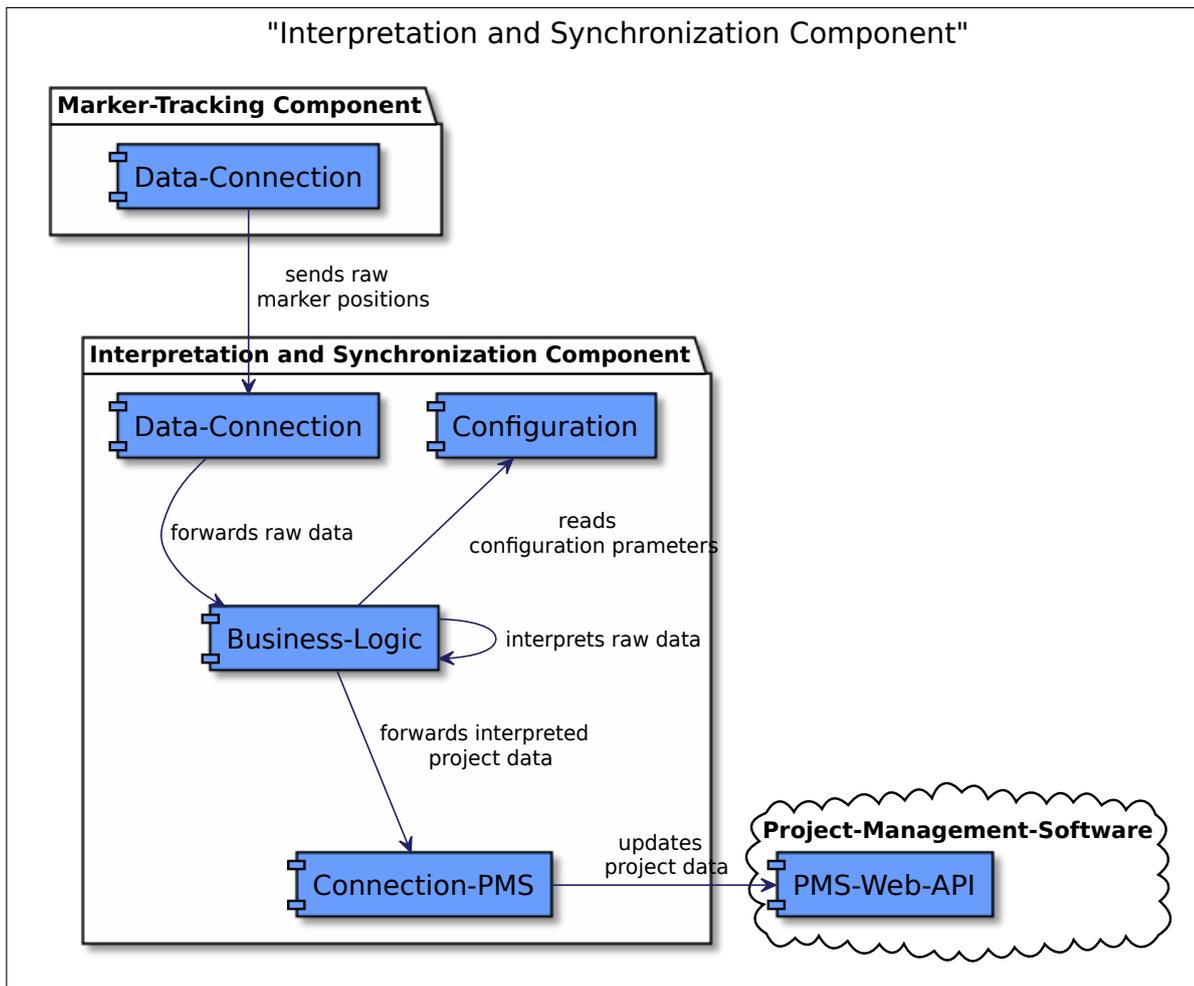


Abbildung 5.17: Vereinfachtes Komponentendiagramm der Komponente "Interpretation und Synchronisation".

Abb. 5.17 zeigen den schematischen Aufbau der Anwendung "Interpretation und Synchronisation". Die Rohdaten (im Wesentlichen Identifikationsnummer und Marker (vgl.

Abb. B.5)) werden durch die Komponente *Data-Connection* empfangen und zur weiteren Verarbeitung an die *Business-Logic* gegeben. Hierbei kann eine beliebige Anzahl an Instanzen der Anwendung "Marker-Tracking" eingesetzt werden. Die Anzahl hängt hierbei vom jeweils eingesetzten Projektaufbau und der Anzahl verwendeter Kameras ab (vgl. 5.7.2). Weiterhin muss bekannt sein, welche AR-Marker zur Markierung von Spalten bzw. Zeilen auf dem Task-Board eingesetzt werden und welche Bedeutung die jeweiligen Bereiche auf dem Task-Board haben. Hierfür können in einer entsprechenden Konfigurationsdatei die jeweiligen Bereiche auf dem Task-Board definiert werden. Abb. 5.18 zeigt eine entsprechende Konfiguration. Dabei wird beispielhaft festgelegt, welche AR-Marker die Spalte für den Bearbeitungszustand "In Verification" begrenzen. Eine analoge Definition muss einmalig für alle Bearbeitungszustände in einer entsprechenden Konfigurationsdatei hinterlegt werden. Das Auslesen dieser Konfiguration übernimmt das Modul *Configuration*.

```
{
  "MarkerIDs": [2981, 2979, 2976, 2973, 2967, 2970, 2966, 2972, 2969, 2975, 2978, 2965],
  "Type": "IN_VERIFICATION"
}
```

Abbildung 5.18: JSON-Format zur Definition einer Spalte auf einem Task-Board

Im Modul *Business-Logic* findet dann die eigentliche Interpretation und zugleich auch die Abbildung der Rohdaten auf das Datenmodell (vgl. Abb. 5.1) statt. Die Interpretation ist von den jeweils im Projekt definierten Datenmodellen bzw. Arbeitsabläufen abhängig. Im Rahmen der prototypischen Implementierung wurde die nachfolgend beschriebene Interpretationslogik umgesetzt:

- Interpretation der Eltern-Kind-Beziehungen: Die Eltern-Kind-Beziehung (hier: eine Story (Eltern) besteht aus n Tasks (Kind) gem. Abb. 5.1) wird am physikalischen Task-Board durch jeweils eine Zeile repräsentiert (vgl. 3.1.3.3). Die Interpretationskomponente kennt den Typ der jeweiligen Story-Card (Epic, Story, Task (vgl. 3.1.2.2)) sowie die Position am Task-Board. Mit Hilfe dieser Informationen können alle Eltern-Kind-Beziehungen der am Task-Board dargestellten User-Stories erkannt und in die PMS übertragen werden.
- Zuordnung einer Story zum aktiven Sprint: Alle User-Stories auf dem Task-Board und die zugeordneten Tasks werden dem aktiven Sprint in der PMS zugeordnet. Das Sprint-Backlog wird somit automatisiert in der PMS übernommen.
- Aktualisierung der Bearbeitungszustände: Die Bearbeitungszustände sind sehr einfach über die jeweilige Spalte am Task-Board interpretierbar. Sie werden ebenfalls automatisiert erkannt und in die PMS übernommen.

Zusammen mit den Daten der Komponente "Smartphone-Anwendung" sind die im Datenmodell (vgl. Abb. 5.1) geforderten Informationen vollständig. Sollten weitere Attribute, Entitäten oder Bearbeitungszustände aufgrund spezieller Projektanforderungen

5.7 Realisierung eines Prototypen

erforderlich sein, können diese nach dem gleichen Prinzip ergänzt werden. Für die in dieser Arbeit enthaltene Konzeptevaluation ist der eben vorgestellte Funktionsumfang jedoch ausreichend.

Kapitel 6

Konzeptevaluation

Das folgende Kapitel dient der Evaluation des in Kap. 5 erarbeiteten Konzepts Virtual Task-Board und entspricht damit dem fünften Schritt im DSRP (vgl. Abs. 2.1).

Im Fokus der Untersuchung steht der Einfluss des prototypisch implementierten Lösungskonzepts Virtual Task-Board (vgl. 5.7) auf die Gruppenarbeit lokaler Projektteams. Als Untersuchungsplan wurde ein Qualitatives Experiment mit Hilfe zweier studentischer Softwareentwicklungsteams durchgeführt. Dabei kamen drei Erhebungstechniken zum Einsatz:

Kern der Datenerhebung war eine qualitative Beobachtung (vgl. Abs. 2.2.4.1), die sich weitestgehend an dem von Katsma et al. publizierten Bewertungsmodell für agile Projektmanagement-Software [46] orientierte. Ergänzend dazu wurde eine schriftliche Befragung mit Hilfe klassischer Erhebungstechniken und der sog. QST (vgl. Abs. 2.2.4) eingesetzt, die ebenfalls auf dem Bewertungsmodell nach Katsma et al. basierte. Während die Ergebnisse der Beobachtung eine externe Perspektive auf die Teilnehmergruppen ermöglicht, spiegeln die Resultate der schriftlichen Befragung die interne Perspektive der Studienteilnehmer wieder. Als dritter Baustein wurde die sog. System Usability Scale nach Brooke [12] eingesetzt. Hierbei handelt es sich um eine bewährte Methode zur Messung der Gebrauchstauglichkeit (gem. DIN EN ISO 9241-11) (engl. Usability), die einen direkten Vergleich mehrerer Systeme zulässt. Eine allgemeine Beschreibung der genannten Forschungsmethoden wurde bereits in Kap. 2 gegeben.

6.1 Forschungsziele und Hypothesen

Die prototypische Implementierung des Virtual Task-Board Konzepts basiert weitestgehend auf bewährten Frameworks und Geräten (vgl. Kap. 5.7). Trotzdem werden zu einem gewissen Teil neu entwickelte Softwarekomponenten benötigt, die das Zusammenspiel der eingesetzten Soft- und Hardware im Sinne des Virtual Task-Board Konzepts ermöglichen. Weiterhin fehlen praktische Erfahrungen mit dem Virtual Task-Board Konzept im Rahmen agiler Projektarbeit, weshalb neben der eigentlichen Evaluation auch das folgende, übergreifende Studienziel formuliert wurde:

Ergänzendes Studienziel: Ein ergänzendes Ziel der Studie ist es, konzeptionelle und technische Schwächen sowie Optimierungsmöglichkeiten aufzudecken, um sie für die Weiterentwicklung des Virtual Task-Board Konzepts auf technischer und konzeptioneller

Ebene bereitzustellen.

Hauptfokus der Studie: Der Fokus liegt auf der Gruppenarbeit innerhalb agiler Entwicklungsteams. Wie bereits in 3.1.3.1 dargelegt wurde, hat der Einsatz konventioneller PMS gegenüber der Verwendung physikalischer Projektmanagementlösungen Nachteile in den Bereichen Kommunikation im Team, Motivation und Transparenz. Von Interesse ist demnach, ob und in welchem Ausmaß sich diese negativen Tendenzen auch beim Einsatz des TUI zeigen bzw. inwieweit sich Verbesserungen gegenüber der browserbasierten Benutzerschnittstelle feststellen lassen. Um diesen Aspekt genauer zu untersuchen soll folgende Hypothese überprüft werden:

Hypothese 1 (H1): Im Vergleich zu konventionellen, browserbasierten Benutzerschnittstellen agiler Projektmanagement-Software lassen sich beim Einsatz von Virtual Task-Board Verbesserungen in den Bereichen Kommunikation, Transparenz und Motivation innerhalb lokaler, agiler Projektteams feststellen.

Neben den bereits formulierten Zielen ist es ebenso wichtig, dass die Gebrauchstauglichkeit gem. DIN EN ISO 9241 durch den Einsatz von Virtual Task-Board gegenüber der konventionellen Jira Agile Benutzerschnittstelle nicht negativ beeinträchtigt bzw. idealerweise sogar verbessert wird. Die DIN EN ISO 9241 definiert Gebrauchstauglichkeit folgendermaßen:

Definition 6.1. Gebrauchstauglichkeit: *"Das Ausmaß in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen."*[32]

Für weitere Definitionen zu Effizienz, Effektivität bzw. Zufriedenstellung sei an dieser Stelle direkt auf die entsprechende Norm DIN EN ISO 9241 verwiesen. Um den Aspekt der Gebrauchstauglichkeit im Rahmen dieser Studie zu berücksichtigen soll folgende Hypothese überprüft werden:

Hypothese H2 (H2): Der Einsatz von Virtual Task-Board innerhalb agiler Projektteams wirkt sich im Vergleich zu konventionellen, browserbasierten Benutzerschnittstellen agiler Projektmanagement-Software nicht negativ auf die Gebrauchstauglichkeit aus.

Die nachfolgend aufgeführten Aspekte wurden im Rahmen der Evaluation nicht berücksichtigt, da sie zur Erreichung der in 1.3 definierten Ziele nicht zwingend erforderlich sind. Dennoch macht eine begründete Auflistung an dieser Stelle Sinn, um die Abgrenzung zu weiterführenden Forschungsfragen zu verdeutlichen.

1. **Betrachtung der Teamarbeit zwischen örtlich verteilten Projektteams:** Die Zusammenarbeit örtlich verteilter Projektteams benötigt zweifelsohne die ortsunabhängige Verfügbarkeit relevanter Projektdaten. In der Praxis ist hierfür der Einsatz von PMS eine gängige Lösungsmöglichkeit. Die dadurch auftretenden Probleme innerhalb der lokal arbeitenden Projektteams stehen im Fokus der Forschung

(vgl. 3.1.3.1) und dieser Arbeit. Der Einfluss von PMS auf die Zusammenarbeit zwischen verteilten Projektteams wird als separates Thema betrachtet und ist nicht Kernbestandteil dieser Arbeit. Entsprechend wird auch der Einfluss des Virtual Task-Board Konzepts auf die Zusammenarbeit zwischen den verteilten Teams nicht im Detail betrachtet.

2. **Betrachtung der Qualität der erfassten Projektdaten:** Die Qualität der automatisch erfassten Daten (z.B. vom Task-Board bzw. Product-Backlog) ist zu großen Teilen von der Qualität der Handschrifterkennung abhängig. Im Rahmen der Arbeit wurde deshalb der State-of-the-Art im Bereich Handschrifterkennung (also die Online-Handschrifterkennung) verwendet. Eine tiefgehende Untersuchung der verfügbaren Methoden zur Handschrifterkennung ist nicht Teil dieser Arbeit. An dieser Stelle wurde auf bereits vorliegende Forschungsarbeiten in diesem separaten Bereich zurückgegriffen (vgl. 5.4.1).
3. **Vergleich des Lösungskonzepts mit einem physikalischem Task-Board:** Das hier beschriebene qualitative Experiment beinhaltet den Vergleich des Lösungskonzepts mit einer konventionellen PMS, weshalb keine vergleichenden Aussagen mit Blick auf ein rein physikalisches Task-Board getroffen werden können. Ergänzend betrachtet hätte der zusätzliche Vergleich mit einem rein physikalischen Task-Boards den Mehrwert, dass die Schwachpunkte bzgl. der Qualität der Gruppenarbeit des Lösungskonzepts klar benannt werden könnten. Ziel der Arbeit ist jedoch ein Lösungskonzept zur Unterstützung der verteilten Projektarbeit, weshalb im ersten Schritt ein Vergleich mit konventioneller PMS zielführender ist. In einem weiteren Schritt (nicht mehr Bestandteil dieser Arbeit) wäre jedoch ein Vergleich mit dem physikalischen Task-Board aus den genannten Gründen sinnvoll.

6.2 Studienaufbau

Alle Hypothesen beinhalten jeweils einen Vergleich zwischen dem in Abs. 5 vorgestellten Konzept Virtual Task-Board und einer konventionellen, browserbasierten Benutzerschnittstelle. Entsprechend wurde die Beobachtungsstudie als Qualitatives Experiment (vgl. 2.2.3) konzipiert.

Die Grundidee zur Studie beinhaltet zwei Projektteams mit vergleichbarer Vorerfahrung, die unter möglichst identischen Rahmenbedingungen jeweils eine der beiden Projektmanagement-Lösungen einsetzen. Um diese Voraussetzungen erfüllen zu können, wurden die nachfolgend beschriebenen Rahmenbedingungen geschaffen.

Allgemeine Rahmenbedingungen und Zusammenstellung der Projektteams:

Im Vorfeld der Studie wurden zwei studentische Teams der Hochschule München gebildet, die im Rahmen einer einsemestrigen Lehrveranstaltung ein Softwareentwicklungsprojekt mit Hilfe von Scrum durchführen sollten. Durch Vorgespräche wurde das Vorwissen der Studenten ermittelt, um die einzelnen Scrum Rollen (Product-Owner, Scrum-Master, Entwickler) vergleichbar zu besetzen und die Teams geeignet aufzuteilen. Die genaue Zusammensetzung der beiden Teams sowie Angaben zu Alter, Geschlecht und Vorerfahrung

mit agiler Softwareentwicklung kann den Tabellen 6.1 sowie 6.2 entnommen werden. Während des laufenden Semesters fanden wöchentliche Präsenztermine statt, die zur Wissensvermittlung bzgl. agiler Prozesse, zur Präsentation von Zwischenergebnissen und zur Klärung von Anforderungsdetails genutzt wurden. Während dieser Termine wurde weder der entwickelte Prototyp noch eine rein PMS basierte Lösung eingesetzt. Ziel der Lehrveranstaltung war unter anderem die Wissensvermittlung agiler Entwicklungsmethoden, weshalb größtenteils physikalische Medien (Task-Boards, Product-Backlog-Charts) eingesetzt wurden. Eine Einführung in die PMS Jira Agile war ebenfalls Bestandteil der Lehrveranstaltung. Das qualitative Experiment, also die eigentliche Studie wurde am Ende des Semesters durchgeführt, weil nach dieser Zeit die praktische Vorerfahrung aller Untersuchungsteilnehmer ein vergleichbares Niveau erreicht hatte. Hierfür wurde zusammen mit den Product-Ownern ein begrenzter fachlicher Umfang festgelegt, der innerhalb der Studiendauer umsetzbar war. Die detaillierte Sprint-Planung wurde dann gemäß Scrum-Prozess während der Studie von den beiden Projektteams durchgeführt. Der Studienzeitraum wurde auf zwei Tage pro Team (jeweils von 09:00 bis 17:00 Uhr) festgelegt. Pro Tag wurde jeweils ein kompletter Sprint mit allen Phasen und Besprechungen des Scrum-Prozesses durchgeführt. Der Zeitplan hierfür wurde jeweils vom Forschungsteam ausgearbeitet und mit den Scrum-Mastern abgestimmt. Das in 6.1 aufgezeigte Team "Virtual Task-Board" setzte während des Studienzeitraums den entwickelten Prototypen ein. Das Team "Jira Agile" entsprechend die PMS (vgl. 6.2).

Tabelle 6.1: Demographische Daten Team "Virtual Task-Board"

Team "Virtual Task-Board"			
Rolle	Alter	Geschlecht	Vorerfahrung in agilen Projekten
Scrum Master	27	m	2 Jahre
Product Owner	24	m	2,5 Jahre
Entwickler	25	m	1 Jahr
Entwickler	26	m	1,5 Jahre
Entwickler	25	m	2 Jahre

Tabelle 6.2: Demographische Daten Team "Jira Agile"

Team "Jira Agile"			
Rolle	Alter	Geschlecht	Vorerfahrung in agilen Projekten
Scrum Master	24	w	1,5 Jahre
Product Owner	26	m	3 Jahre
Entwickler	23	w	0,5 Jahre
Entwickler	24	m	0,5 Jahre
Entwickler	26	m	1 Jahr
Entwickler	25	m	1,5 Jahre
Entwickler	27	m	2,5 Jahre

Vergabe der Entwicklungsthemen: Die Themen beider Teams wurden aus technischer Sicht vergleichbar gewählt. Es handelte sich bei beiden Projektgruppen um eine Client-Server Anwendung, die sich jeweils aus einer Webapplikation mit browserbasierter Benutzerschnittstelle und Datenbankbindung sowie einer Applikation für Mobilgeräte zusammensetzte. Die fachliche Projektidee sowie Anforderungsdetails wurden im Rahmen der übergreifenden Lehrveranstaltung festgelegt. Da sie für die Studienergebnisse nur von untergeordneter Rolle sind werden die Themen hier nur kurz erläutert:

Thema Nr. 1: Fachliches Projektziel war eine Software zur dreidimensionalen Vermessung von Räumlichkeiten mit Hilfe eines Mobilgeräts (z.B. Smartphone oder Tablet). Zur Erfassung der Raumstruktur wurde die Google Tango Technologie [81] eingesetzt. Ein weiteres Ziel war die Bereitstellung der erfassten 3D-Modelle mit Hilfe einer Online-Plattform. Der Upload sollte dabei direkt vom Mobilgerät aus ermöglicht werden.

Thema Nr. 2: Das fachliche Projektziel des zweiten Teams war eine Onlineplattform, die potentiellen Nutzern eine Suche nach attraktiven Mittagsangeboten sowie eine Vorbestellung der Mahlzeiten ermöglichen sollte. Ein wesentlicher Bestandteil war neben der entsprechenden Webapplikation eine browserbasierte Benutzerschnittstelle für Restaurant- und Imbissbesitzer, um Essensangebote einstellen und pflegen zu können. Weiterhin war eine Applikation für Mobilgeräte zur Suche und Bestellung von Essensangeboten im fachlichen Umfang der Anwendung enthalten.

6.2.1 Bewertungsmodell

Im Folgenden wird ein Bewertungsmodell vorgestellt, das den bestehenden Ansatz von Katsma et al. [46] aufgreift und damit als Auswertungsgrundlage für die Beobachtungsstudie und für die in Abs. 6.3.2 beschriebene schriftliche Befragung geeignet ist. Das Bewertungsmodell besteht aus Kriterien, die mit Hilfe von Beobachtungsstudien in vier Unternehmen sowie mit insgesamt 15 Experteninterviews ermittelt wurden. Das Modell wurde speziell für die Bewertung von Projektmanagement-Lösungen innerhalb agil arbeitender Projektteams erarbeitet und kann deshalb für beide Lösungen (Jira Agile und Virtual Task-Board) angewendet werden, um eine Vergleichbarkeit zu ermöglichen. Es setzt sich aus folgenden Kriterien zusammen:

- **(K1) Zugänglichkeit (engl. accessibility):** Agile Task-Boards sollten im Arbeitsbereich des Teams zu jeder Zeit gut sichtbar und lesbar sein, um einen schnellen Überblick des aktuellen Arbeitsstands zu ermöglichen. Der Entwicklungsprozess sollte dabei verständlich abgebildet und schnell zu erfassen sein. Interaktionen mit dem Task-Board sollten jederzeit möglich und für alle Teammitglieder sichtbar sein, um möglichst hohe Transparenz zu schaffen.

Hinweis bzgl. Zugänglichkeit und der Hypothese H1: Dieses Kriterium entspricht inhaltlich dem in H1 angenommenen Verständnis von Transparenz im agilen Projektteam. Der Begriff Zugänglichkeit wurde jedoch beibehalten, da er von

Katsma et al. [46] entsprechend definiert wurde.

- **(K2) Flexibilität (engl. flexibility):** Dieses Kriterium bezieht sich in erster Linie auf die individuelle Gestaltbarkeit der dargestellten Inhalte. Es sollte ohne größeren Aufwand möglich sein kleine Abweichungen vom Entwicklungsprozess oder Zusatzinformationen abzubilden bzw. optisch hervorzuheben. Wird der Entwicklungsprozess angepasst, um den Vorgaben der Projektmanagement-Lösung bzw. dem Task-Board zu entsprechen, ist das ein Anzeichen für geringere Flexibilität.
- **(K3) Motivation (engl. motivation):** Laut Katsma et al. resultiert eine motivierende Wirkung in erster Linie aus transparenter Interaktion mit dem Task-Board. Verändert man den Status auf dem Task-Board kann das zu unmittelbaren Reaktionen aus dem Projektteam führen (z.B. Lob) und somit die Motivation steigern.
- **(K4) Haptische Rückmeldung (engl. haptic feedback):** Unter haptischer Rückmeldung versteht Katsma et al. eine positive und motivierende Wirkung, die im Zusammenhang mit der Arbeit am Task-Board entsteht. In erster Linie knüpft er diese Wirkung an die direkte Interaktion mit physikalischen Medien. Wenn man beispielsweise eine Karte am Task-Board zu den erledigten Aufgaben hängt geht das mit einem positiven und motivierenden Gefühl einher. Durch positive Rückmeldung anderer Kollegen kann dieser Faktor noch verstärkt werden.
- **(K5) Integration: (engl. integration)** Unter diesem Kriterium ist die Fähigkeit der eingesetzten Projektmanagement-Lösung bzw. des Task-Boards zu verstehen, in bestehende Prozesse und Arbeitsabläufe integrierbar zu sein. Konkret nennt Katsma et al. hier z.B. die Fähigkeit des Datenaustauschs mit anderen Softwaresystemen und die Möglichkeit zur Weiterverarbeitung der Daten, um beispielsweise Statusdiagramme (z.B. ein Burn-Down-Chart) zu generieren.
- **(K6) Archivierung (engl. archiving):** Katsma et al. adressiert mit diesem Punkt die Möglichkeit auf Daten vergangener Sprints oder Projekte zuzugreifen, um beispielsweise aus vergangenen Fehlern zu lernen und den Entwicklungsprozess entsprechend zu optimieren.
- **(K7) Mehraufwand (engl. overhead):** Dieses Kriterium zieht mehrere Faktoren in Betracht. Laut Katsma et al. ist der erste Faktor die erforderliche Einarbeitungszeit, die das Projektteam bzw. neu hinzukommende Mitarbeiter benötigen, um den Umgang mit der Projektmanagement-Lösung bzw. mit dem Task-Board zu erlernen. Ein weiterer Aspekt, sind die notwendigen Schritte vor der erstmaligen Nutzung. Dazu zählt beispielsweise das Anlegen von Accounts oder das Beantragen von Zugriffsrechten. Am wichtigsten ist laut Katsma et al. jedoch der

Aufwand, der während des Projekts zur Nutzung des Task-Boards notwendig ist. Im Falle einer browserbasierten Lösung wäre das z.B. das Öffnen des Webbrowsers oder der Login in das System.

- **(K8) Verfügbarkeit (engl. availability):** Unter hoher Verfügbarkeit versteht Katsma et al. die ständige Einsatzbereitschaft der Projektmanagement-Lösung. Je weniger Beeinträchtigungen der Projektarbeit durch fehlerhaftes Verhalten der Projektmanagement-Lösung oder durch Systemausfälle auftreten, desto besser kann die Verfügbarkeit bewertet werden.
- **(K9) Distanz (engl. distance):** Katsma et al bezieht sich hier auf die geografische Distanz und spricht mit diesem Kriterium die Zugänglichkeit der Projektdaten von geographisch entfernten Orten an. Damit ist beispielsweise der Zugriff auf Projektdaten durch ein örtlich entfernt arbeitendes Projektteam gemeint. Zu bewerten ist demnach die Eignung der Zugriffsmöglichkeit für agil arbeitende Teams sowie der damit verbundene Aufwand.
- **(K10) Kommunikation (engl. communication):** Unter diesem Kriterium wird die Fähigkeit verstanden persönliche Gespräche im Team anzuregen und für Transparenz im Projektteam zu sorgen. Katsma et al. führt die Wichtigkeit eines Task-Boards als zentralen Treffpunkt für Gespräche als Beispiel an.
- **(K11) Kosten (engl. costs):** Dieser Aspekt meint sowohl die Anschaffungskosten als auch die Betriebskosten (z.B. Hardwareressourcen, Lizenzkosten) der betrachteten PMS.

Kriterienauswahl: In diesem Schritt wurde das Bewertungsmodell an die Studienziele, die Hypothesen und die jeweils eingesetzte Erhebungsmethode angepasst, indem eine entsprechende Teilmenge der Kriterien ausgewählt wurde. Dieser Schritt ist notwendig, da sich die ausgewählten Erhebungsmethoden nicht für alle Kriterien im Bewertungsmodell eignen. Die Auswahl der Kriterien wurde folgendermaßen vorgenommen:

Tabelle 6.3: Kriterienauswahl Evaluationsstudie

Bewertungskriterium	Beobachtung	Schriftl. Befragung
(BK1) Zugänglichkeit	X	X
(BK2) Flexibilität	X	X
(BK3) Motivation	X	X
(BK4) Haptische Rückmeldung	X	
(BK5) Integration	X	
(BK6) Archivierung		
(BK7) Mehraufwand	X	X
(BK8) Verfügbarkeit		
(BK9) Distanz		
(BK10) Kommunikation	X	X
(BK11) Kosten		

Tabelle 6.3 zeigt, dass einzelne Bewertungskriterien weder durch die Beobachtungsstudie noch durch die schriftliche Befragung abgedeckt wurden. An dieser Stelle möchten wir kurz auf die Gründe dafür eingehen:

Archivierung: Der Fokus der Studie liegt auf dem Vergleich zwischen der PMS Jira und dem Virtual Task-Board Konzept. Wie in Kap. 5 beschrieben setzt das Virtual Task-Board Konzept jedoch auf Jira auf. Alle erfassten Daten werden mit Jira synchronisiert und können dort über die browserbasierte Benutzerschnittstelle weiterverarbeitet werden. Die Archivierungsmöglichkeiten der Jira Standardlösung können also gleichermaßen verwendet werden, weshalb ein Vergleich an dieser Stelle nicht durchgeführt wird.

Verfügbarkeit: Die Verfügbarkeit soll im Rahmen der Studie nicht beurteilt werden, weil sie zur Beantwortung der Hypothesen nur eine sehr untergeordnete Rolle spielt. Zudem wurde das Virtual Task-Board Konzept nur prototypisch implementiert, wodurch ein Vergleich mit der kommerziellen PMS Jira Agile nicht aussagekräftig wäre.

Distanz: Nach obiger Definition ist die Eignung der zu bewertenden PMS bzgl. ihrer Zugänglichkeit aus der Distanz zu betrachten. Der Zugriff auf die Projektdaten aus der Distanz erfolgt bei beiden Lösungen über das GUI der PMS (in diesem Fall Jira Agile). Entsprechend ist die Eignung des Zugriffs sowie der Aufwand zur Nutzung des Zugriffs identisch. Vor diesem Hintergrund ist ein Vergleich beider Lösungen nicht zielführend.

Kosten: Für beide PMS Lösungen wird eine Jira Standardinstallation sowie eine entsprechende Lizenz benötigt, weshalb an dieser Stelle die gleichen Kosten anfallen. Zusätzliche Kosten für den Betrieb mit der Virtual Task-Board Lösung können derzeit noch nicht eingeschätzt werden. Weiterhin ist dieses Bewertungskriterium zur Überprüfung der Hypothesen sowie zur Erreichung der Forschungsziele nicht relevant.

6.2.2 Datenerhebung und Protokollierung der Beobachtungsstudie

Die Beobachtungsprotokolle sind der erste wichtige Baustein, um zu einer Aussage bzgl. Hypothese 1 (H1) zu gelangen. Beobachtung und Protokollierung wurden durch ein Mitglied des Forschungsteams durchgeführt, das nicht Teil der Projektteams war. Aus diesem Grund spricht man hier von einer internen, nicht teilnehmenden Fremdbeobachtung (vgl. Abs. 2.2.4.1). Ein Eingriff durch den Beobachter war nur erlaubt, um die Einhaltung des Zeitplans weitestgehend sicherzustellen. Während der Beobachtung wurde zu jeder Prozessphase bzw. Besprechung ein eigenes Protokoll angefertigt, das jeweils aus folgenden drei Teilen besteht:

Im ersten Teil sollte der Beobachter die Rahmenbedingungen der jeweiligen Prozessphase beschreiben. Dazu zählen alle eingehenden Daten (z.B. Ergebnisse und Artefakte aus vorangegangenen Prozessphasen) sowie die aktiv eingesetzten Medien und Hilfsmittel. Zusätzlich sollte kurz beschrieben werden, wo (z.B. Raum) und in welcher Form (z.B. sitzend oder stehend) das Meeting stattfindet. Start- und Endzeitpunkt wurden ebenfalls notiert.

Im zweiten Teil sollte der Ablauf der jeweiligen Prozessphase beschrieben werden. Hier war insbesondere wichtig, wie die Teilnehmer vorgehen und wie die einzelnen Medien bzw. Artefakte eingesetzt werden. Außerdem sollte kurz protokolliert werden, wie die einzelnen Rollen im Projektteam agieren und welche Aufgaben sie wahrnehmen.

Der dritte Teil des Protokolls bezieht sich nun direkt auf die in Tab. 6.3 gezeigten Kriterien. Hierbei sollten Auffälligkeiten notiert werden, die als Indikator für mindestens eines der Bewertungskriterien gewertet werden können. Um diese Aufgabe zu erleichtern wurde ein Indexblatt angefertigt (vgl. Anh. C.2), das Leitfragen zu jedem Kriterium bereitstellt. Diese Leitfragen wurden im Vorfeld der Studie auf Basis der Ausführungen von Katsma et al. erarbeitet. Im Protokoll wurde zu jeder der beobachteten Auffälligkeiten die zugehörigen Kriterien, die aktuelle Uhrzeit, eine Beschreibung sowie optional eine erster Interpretationsansatz angegeben.

Um die Protokollierung zu vereinfachen, wurde dem Protokollanten eine entsprechende Vorlage bereitgestellt, die der eben aufgeführten Protokollstruktur entsprach (vgl. Anh. C.1).

6.2.3 Datenerhebung der schriftlichen Befragung

Wie bereits in Abs. 2.2.4.2 beschrieben wurde die schriftliche Befragung mittels eines klassischen Rangordnungsverfahrens und zusätzlich mit Hilfe der QST [80] durchgeführt. Allgemeine Gründe für den Einsatz beider Verfahren sowie der Einfluss der jeweiligen Erhebungsmethode auf die zu erwartenden Ergebnisse wurden ebenfalls in 2.2.4.2 erläutert. Im Rahmen dieser Arbeit ist das klassische Rangordnungsverfahren notwendig, um eine quantitative Vergleichbarkeit beider Lösungen zu ermöglichen. Neben den Daten aus der Beobachtungsstudie ist die schriftliche Befragung der zweite wichtige Baustein, um eine Aussage bzgl. Hypothese 1 (H1) treffen zu können. Die Befragung auf Basis des klassischen Rangordnungsverfahrens wäre bereits geeignet, um eine vergleichende Aussage

bzgl. der Qualität der Gruppenarbeit (vgl. Forschungsfrage Z3 in 4.1) zu treffen. Der Einsatz der QST ist lediglich ergänzend zu betrachten und nicht zwingend notwendig, um eine Aussage bzgl. der definierten Hypothesen zu treffen. Diese Erhebungsmethode wurde dennoch ausgewählt, um die Einflüsse der beiden Lösungskonzepte auf einzelne Aspekte der Gruppenarbeit (dargestellt durch das Bewertungsmodell (vgl. 6.2.1) und die zugehörigen Indikatoren (vgl. Tab. 6.4)) im Detail zu betrachten. Die QST zwingt den Probanden eine Abstufung bzgl. der Bewertung einzelner Indikatoren einzuhalten und kann demnach ein differenzierteres Ergebnis liefern.

In beiden Fällen kam eine fünfstufige Ratingskala nach Likert [52] von "trifft überhaupt nicht zu" bis "trifft voll und ganz zu" zum Einsatz. Um Datenerhebung und Auswertung zu vereinfachen wurde auf eine digitale Lösung [74] zurückgegriffen, die eine einfache Durchführung der Befragung nach klassischer Vorgehensweise sowie mit Hilfe der QST erlaubt. Alle Studienteilnehmer erhielten vorab eine kurze Einweisung dazu. Abb. 6.1 zeigt eine Bildschirmaufnahme der eingesetzten Softwarelösung. Die zu bewertenden Aussagen werden dabei zeilenweise in zufälliger Reihenfolge angezeigt. Rechts kann per Mausklick die jeweilige Bewertung abgegeben werden. Im Fall der QST zeigt die Software an, ob die getroffene Auswahl bereits der geforderten Normalverteilung entspricht, indem der Indikator der jeweiligen Bewertungsstufe zugeordnet wird. In Abb. 6.1 ist bereits eine abgeschlossene Bewertung entsprechend der vorgegebenen Normalverteilung dargestellt.

+2		-2	-1	0	+1	+2
13.	Der Einsatz des Projektmanagement-Tools während der Meetings hat meiner Meinung nach nicht gestört.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
17.	Die Zeit, die wir in den Einsatz des Projektmanagement-Tools investiert haben hat sich gelohnt.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
15.	Wir haben während der Meetings zielführende Diskussionen geführt.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
+1		-2	-1	0	+1	+2
3.	Die anfallende Arbeit zur Aktualisierung der Daten im Projektmanagement-Tool war gleichmäßig auf alle Teammitglieder verteilt.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
12.	Ich habe immer sofort daran gedacht, die Daten im Projektmanagement-Tool zu aktualisieren, wenn sich etwas am Projektfortschritt verändert hat.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
9.	Nachdem ich eine Aufgabe fertiggestellt habe bekam ich immer unmittelbares Feedback / Reaktionen von meinen Teamkollegen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
2.	Ich habe stets einen Überblick, wer gerade an einem bestimmten Thema arbeitet.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
0		-2	-1	0	+1	+2
5.	Das eingesetzte Projektmanagement-Tool hat genau zur eingesetzten Entwicklungsmethode gepasst.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
20.	Ich habe Verstanden wie ich das Projektmanagement-Tool sinnvoll bedienen und benutzen kann.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
4.	Ich kann die im Projektmanagement-Tool verwalteten Daten in allen Situationen (Meetings und Phasen) gut überblicken.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
14.	Ich habe Status-Updates anderer Teammitglieder im Projektmanagement-Tool bemerkt und daraufhin das Gespräch mit diesen Personen gesucht.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
19.	Ich hatte jederzeit sofortigen Zugang zum Projektmanagement-Tool wenn ich Daten aktualisieren oder einsehen wollte.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
18.	Die Kurzeinführung in das Projektmanagement-Tool war absolut ausreichend.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
-1		-2	-1	0	+1	+2
16.	Ich hatte Fragen zu Informationen im Projektmanagement-Tool und habe mich daraufhin direkt an die zuständige(n) Person(en) gewandt.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8.	Ich denke, dass man das Projektmanagement-Tool auch für andere agile Methoden problemlos einsetzen kann.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.	Ich bin jederzeit über den Arbeitszustand einzelner Aufgabenpakete informiert und kann sofort darauf reagieren.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10.	Ich habe das Projektmanagement-Tool zu jeder Zeit im Projektverlauf gerne eingesetzt.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
-2		-2	-1	0	+1	+2
6.	Das Projektmanagement-Tool bietet genug Freiheiten, um den Entwicklungsprozess an die individuellen Bedürfnisse des Entwicklungsteams anzupassen.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7.	Einzelne Abweichungen vom Entwicklungsprozess konnten mühelos im Projektmanagement-Tool abgebildet werden.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11.	Es gibt mir ein gutes Gefühl, wenn ich einzelne Aufgabenpakete im Projektmanagement-Tool als "erledigt" markieren kann.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Abbildung 6.1: Browserbasierte Weboberfläche der Q-Sort Befragung.
(Quelle: Webanwendung von Schmolck [74])

Wie bereits in Abs. 2.2.4.2 erläutert, handelt es sich bei allen herangezogenen Bewertungskriterien der schriftlichen Befragung (vgl. Tabelle 6.3) um latente -also nicht direkt

messbare- Variablen. Aus diesem Grund wurde für jedes Kriterium ein Messmodell in Form von jeweils vier zu bewertenden Aussagen (nf. Indikatoren genannt) formuliert. Die Anzahl der Indikatoren pro Bewertungskriterium ergibt sich aus zwei Gründen:

Erstens verlangt die Datenerhebung nach QST 20 Indikatoren, um eine normalverteilte Datenerhebung über die fünf ausgewählten Bewertungskriterien zu ermöglichen. Zweitens muss aufgrund der formativen Messmodelle (vgl. Abs. 2.2.4.2) die Anzahl der Indikatoren pro Bewertungskriterium identisch sein, um eine direkte Vergleichbarkeit der Kriterien zu erreichen. Tabelle 6.4 enthält eine Übersicht der verwendeten Bewertungskriterien und deren Indikatoren.

Tabelle 6.4: Bewertungskriterien und Indikatoren der schriftlichen Befragung.

(BK 1) Zugänglichkeit	
(I 1.1)	"Ich bin jederzeit über den Arbeitszustand einzelner Arbeitspakete informiert und kann sofort darauf reagieren."
(I 1.2)	"Ich habe stets einen Überblick, wer gerade an einem bestimmten Thema arbeitet."
(I 1.3)	"Die anfallende Arbeit zur Aktualisierung der Daten im Projektmanagement-Tool war gleichmäßig auf alle Teammitglieder verteilt."
(I 1.4)	"Ich kann die im Projektmanagement-Tool verwalteten Daten in allen Situationen (Meetings und Phasen) gut überblicken."
(BK 2) Flexibilität	
(I 2.1)	"Das eingesetzte Projektmanagement-Tool hat genau zur eingesetzten Entwicklungsmethode gepasst."
(I 2.2)	"Das Projektmanagement-Tool bietet genug Freiheiten, um den Entwicklungsprozess an die individuellen Bedürfnisse des Entwicklungsteams anzupassen."
(I 2.3)	"Einzelne Abweichungen vom Entwicklungsprozess konnten mühelos im Projektmanagement-Tool abgebildet werden."
(I 2.4)	"Ich denke, dass man das Projektmanagement-Tool auch für andere agile Methoden problemlos einsetzen kann."
(BK 3) Motivation	
(I 3.1)	"Nachdem ich eine Aufgabe fertiggestellt habe bekam ich immer unmittelbares Feedback / Reaktionen von meinen Teamkollegen."
(I 3.2)	"Ich habe das Projektmanagement-Tool zu jeder Zeit im Projektverlauf gerne eingesetzt.";
(I 3.3)	"Es gibt mir ein gutes Gefühl, wenn ich einzelne Aufgabenpakete im Projektmanagement-Tool als erledigt markieren kann."
(I 3.4)	"Ich habe immer sofort daran gedacht, die Daten im Projektmanagement-Tool zu aktualisieren, wenn sich etwas am Projektfortschritt verändert hat."
(BK 7) Mehraufwand	
(I 4.1)	"Die Zeit, die wir in den Einsatz des Projektmanagement-Tools investiert haben hat sich gelohnt."
(I 4.2)	"Die Kurzeinführung in das Projektmanagement-Tool war absolut ausreichend."
(I 4.3)	"Ich hatte jederzeit sofortigen Zugang zum Projektmanagement-Tool wenn ich Daten aktualisieren oder einsehen wollte."
(I 4.4)	"Ich habe verstanden, wie ich das Projektmanagement-Tool sinnvoll bedienen und benutzen kann."
(BK 10) Kommunikation	
(I 5.1)	"Der Einsatz des Projektmanagement-Tools während der Meetings hat meiner Meinung nach nicht gestört."
(I 5.2)	"Ich habe Status-Updates anderer Teammitglieder im Projektmanagement-Tool bemerkt und daraufhin das Gespräch mit diesen Personen gesucht."
(I 5.3)	"Wir haben während der Meetings zielführende Diskussionen geführt."
(I 5.4)	"Ich hatte Fragen zu Informationen im Projektmanagement-Tool und habe mich daraufhin direkt an die zuständige(n) Person(en) gewandt."

6.2.4 Datenerhebung mit der System Usability Scale

Die Datenerhebung wurde gem. der in Abs. 2.2.4.3 beschriebenen Vorgehensweise durchgeführt. Sie ist notwendig, um eine Aussage bzgl. Hypothese 2 (H2) zur Gebrauchstauglichkeit treffen zu können. Hierfür erhielt jeder Studienteilnehmer am Nachmittag des zweiten Studientages den standardisierten Fragebogen zur SUS (vgl. C.3) und konnte diesen ohne Zeitvorgabe ausfüllen.

6.3 Ergebnisse

Nachfolgend werden zunächst die Teilergebnisse der einzelnen Erhebungsverfahren aufgeführt und erläutert. Anschließend in Abs. 6.3.4 erfolgt eine Zusammenführung der Teilergebnisse und eine Bezugnahme auf die in Abs. 6.1 formulierten Hypothesen und Ziele.

6.3.1 Ergebnisse der Beobachtungsstudie

Tabelle 6.5 zeigt eine erste Übersicht der Ergebnisse aus der Beobachtungsstudie. Dabei werden die mittels induktiver Kategorisierung (vgl. Abs. 2.2.2) ausgearbeiteten Kategorien (KAT) den einzelnen Bewertungskriterien (BK) zugeordnet. Die Spalte "Bewertung" gibt dabei jeweils einen ersten Vergleich zwischen dem Lösungskonzept Virtual Task-Board und der konventionellen Lösung Jira Agile. Eine mit "+" bewertete Kategorie sagt demnach aus, dass das Lösungskonzept Virtual Task-Board bzgl. der entsprechenden Kategorie besser als das Vergleichskonzept Jira Agile abgeschnitten hat. Analog muss eine Bewertung mit "-" bzw. "=" als schlechteres bzw. vergleichbares Abschneiden des Lösungskonzepts Virtual Task-Board interpretiert werden. Die Ergebnisse der überordneten Bewertungskriterien ergeben sich aus den Einzelbewertungen der jeweiligen Kategorien. Überwiegt die Anzahl der positiv bewerteten Kategorien ergibt sich ein "+" für das jeweilige Bewertungskriterium, andernfalls entsprechend ein "-" oder bei Gleichstand ein "=". Die gleich bewerteten Kategorien werden bei dieser Art der Aggregation nicht berücksichtigt, da auf Basis der vorliegenden Ergebnisse keine eindeutige Aussage getroffen werden kann.

Tabelle 6.5: Ergebnisse der qualitativen Beobachtung

Kategorien	Bewertung
(BK 1) Zugänglichkeit	+
(KAT 1.1) Transparente Interaktion	+
(KAT 1.2) Übersichtlichkeit und Gesamtüberblick	+
(KAT 1.3) Ständige Präsenz	+
(KAT 1.4) Lesbarkeit	-
(BK 2) Flexibilität	+
(KAT 2.1) Beeinflussung des Arbeitsprozesses	+
(KAT 2.2) Freiheitsgrade bei der Darstellung	=
(BK 3) Motivation	+
(KAT 3.1) Visualisierung des Arbeitsfortschritts	+
(KAT 3.2) Embodiment	+
(KAT 3.3) Medienwechsel	+
(BK 4) Haptische Rückmeldung	+
(KAT 4.1) Einsatz physikalischer Medien	+
(KAT 4.2) Multi-User-Interaktion	=
(BK 5) Integration	-
(KAT 5.1) Verfügbarkeit und Export von Kennzahlen	=
(KAT 5.2) Integration zusätzlicher Systeme	-
(BK 7) Mehraufwand	=
(KAT 7.1) Vorbereitungszeit von Besprechungen und Prozessphasen	+
(KAT 7.2) Spontane Interaktion	=
(KAT 7.3) Bedienkomplexität	=
(KAT 7.4) Effizienz	-
(BK 10) Kommunikation	+
(KAT 10.1) Transparenz und spontane Gespräche	+
(KAT 10.2) Embodiment steigert Aufmerksamkeit	+
(KAT 10.3) Multi-User-Interaktion regt zu Gesprächen an	+

Im Folgenden werden Lösungskonzept (Virtual Task-Board) und Vergleichskonzept (Jira Agile, als Vertreter konventioneller PMS) auf Basis aller, im Rahmen der Qualitativen Inhaltsanalyse erarbeiteten, Kategorien (vgl. Abs. 6.2.1) ausführlicher gegenübergestellt. Anschließend werden die beobachteten technischen und konzeptionellen Schwächen von Virtual Task-Board sowie entsprechende Optimierungsmöglichkeiten diskutiert.

(BK 1) Zugänglichkeit

(KAT 1.1) Transparente Interaktion:

Mit Blick auf die Zugänglichkeit hat sich diese Kategorie als einer der wichtigsten Vorteile des Virtual Task-Board Konzepts herausgestellt. Sobald Statusupdates am Task-Board durchgeführt oder Informationen bearbeitet wurden, war das für alle anwesenden Teammitglieder sichtbar. Während der Studie ließen sich Reaktion wie z.B. spontane

Nachfragen zum Thema (z.B. "Bist du schon fertig?" oder "Kann ich das dann testen?") und unmittelbares Feedback (z.B. Lob) von anderen Teammitgliedern beobachten. Bei der konventionellen Lösung mit Jira Agile ließen sich hierbei starke Unterschiede beobachten. Statusupdates wurden teilweise erst verzögert wahrgenommen und Reaktionen blieben aus. Bemerkenswert ist, dass dieser Umstand vom Projektteam adressiert wurde, indem Email-Benachrichtigungen bei Statusupdates in Jira Agile aktiviert wurden. Es lies sich aber beobachten, dass dieser Mechanismus nur teilweise eine Verbesserung gebracht hat. Aufgrund einer recht hohen Anzahl an Email-Benachrichtigungen wurden diese teilweise nicht beachtet bzw. erst nach einiger Zeit bemerkt.

(KAT 1.2) Übersichtlichkeit und Gesamtüberblick:

Im Rahmen der Evaluationsstudie gab es verschiedene Beobachtungen, die eine Aussage bzgl. der Übersichtlichkeit beider Varianten zulassen. Das Team Virtual Task-Board versammelte sich regelmäßig vor dem physikalischen Task-Board, um sich den aktuellen Arbeitsstand anzusehen oder z.B. die Vorauswahl der User-Stories für das Sprint-Planning zu treffen. Dabei konnten keine nennenswerten Schwierigkeiten beobachtet werden.

Im Vergleich dazu ließ sich bei Team Jira Agile gelegentlich beobachten, dass eine User-Story nicht gefunden wurde und es deshalb zu minutenlangen Unterbrechungen einer Besprechung kam. Eine besondere Herausforderung war es beispielsweise, einen Überblick über die geschätzten Aufwände der einzelnen User-Stories zu bekommen. Eine gleichzeitige Darstellung aller relevanten User-Stories war aufgrund der beschränkten Projektorauflösung nicht möglich.

(KAT 1.3) Ständige Präsenz:

Ein weitere Kategorie mit wesentlichem Einfluss auf die Zugänglichkeit ist die Präsenz der Statusinformationen. Hier stellt sich das Virtual Task-Board als gut geeignet heraus, da alle notwendigen Informationen aufgrund der physikalischen Präsenz permanent zur Verfügung stehen. Im Rahmen der Beobachtungsstudie gab es immer wieder Situationen, in denen Teammitglieder spontan zum Task-Board gingen, um Einzelheiten zu diskutieren oder zu planen.

Diese Arbeitsweise war mit der konventionellen Lösung nicht zu beobachten. Hier war vermutlich die Einstiegshürde bzw. der Vorbereitungsaufwand zu groß (Projektor starten und Jira Agile aufrufen).

(KAT 1.4) Lesbarkeit:

Der für das Virtual Task-Board eingesetzte Smart-Pen hatte eine normale Kugelschreibermine, die sich teilweise als zu dünn und auf größere Distanz als unleserlich herausstellte. Seitens der konventionellen Lösung mit Jira Agile gab es diese Probleme nicht. Hier gab es lediglich aufgrund der Projektorhelligkeit kleine Einschränkungen bzgl. der Lesbarkeit.

(BK 2) Flexibilität:

(KAT 2.1) Beeinflussung des Arbeitsprozesses:

Ein wichtiges Kriterium ausreichender Flexibilität nach Katsma et al. [46] ist, dass

sich die PMS den individuellen Bedürfnissen der Projektteams anpassen lässt. Im Fall des Virtual Task-Board Konzepts waren keine konkreten Auffälligkeiten diesbezüglich zu beobachten. Das kann durchaus als positives Zeichen gewertet werden, weil entsprechend auch keine störenden Einschränkungen festgestellt wurden.

Auch im anderen Team hat sich der zugrundeliegende Entwicklungsprozess (Scrum) gut umsetzen lassen. Ein paar kleine Einschränkungen konnten jedoch beobachtet werden. Jira Agile bietet zum Anlegen und Bearbeiten einer User-Story eine vorkonfigurierte Eingabemaske an. Dabei konnte festgestellt werden, dass das Projektteam starr nach der Reihenfolge der darin enthaltenen Eingabemöglichkeiten gearbeitet hat. So wurden beispielsweise die Priorisierung und die Aufwandsabschätzung einzelner User-Stories zeitlich nach vorne gezogen, um die Werte direkt eingeben zu können.

(KAT 2.2) Freiheitsgrade bei der Darstellung von Inhalten:

Zur Flexibilität einer PMS zählt auch die Möglichkeit, Inhalte bedarfsgerecht visualisieren zu können. Durch die physikalische Verfügbarkeit der Statusinformationen im Virtual Task-Board Konzept bieten sich verschiedene Möglichkeiten, die im Rahmen der Studie auch beobachtet werden konnten. Beispielsweise wurden die Informationen am Task-Board gelegentlich gruppiert. Hierfür wurden die entsprechenden Karteikarten einfach übereinander gehängt, um ihre Zusammengehörigkeit darzustellen. Ein weiteres Beispiel war die farbige Markierung einzelner Karten, um die Verantwortlichkeiten (wer erledigt welche Aufgabe) darzustellen. Eine Einschränkung des Virtual Task-Board Konzepts ist jedoch, dass diese Darstellungen nicht automatisch erfasst und digital zur Verfügung gestellt werden können.

(BK 3) Motivation:

(KAT 3.1) Visualisierung des Arbeitsfortschritts sorgt für Motivation:

Ein gut funktionierendes Mittel zur Motivationssteigerung im Projektteam ist die Visualisierung des Arbeitsfortschritts. Hier hat das Virtual Task-Board Konzept den Vorteil, dass diese Information ständig und für das ganze Team sichtbar vorliegt. Kommentare wie "Wir kommen richtig gut voran" oder "Ich glaube wir schaffen das Sprint-Ziel diesmal" belegten recht deutlich die hohe Motivation in diesem Team.

Die konventionelle Lösung mit Jira Agile hat den Nachteil, dass der Arbeitsfortschritt nicht ständig präsent ist, weil der Projektor zur Darstellung der Arbeitsfortschritte nur während der Besprechungen verwendet wurde. Kommentare wie eben beschrieben konnten in diesem Team nicht beobachtet werden.

(KAT 3.2) Embodiment - Motivation durch Körpereinsatz:

Die einfache Tatsache, dass man beim Einsatz von Virtual Task-Board aufstehen und zum Task-Board gehen muss, um eine Statusänderung durchzuführen bringt sichtliche Motivationssteigerungen im Team, die sich hier ebenfalls durch entsprechende Äußerungen der Teammitglieder erkennen ließen. Zusätzlich sorgt diese Art der Interaktion mit dem Task-Board für Transparenz und unmittelbares Feedback (z.B. Lob,

Anerkennung) aus dem Projektteam.

Dieser Effekt konnte beim Einsatz der Jira Agile Lösung wesentlich weniger häufig beobachtet werden, da die Statusupdates erst mit einiger Verzögerung von den anderen Teammitgliedern wahrgenommen wurden.

(KAT 3.3) Medienwechsel bringt Motivation:

Ein Großteil der Softwareentwicklung findet am Bildschirmarbeitsplatz statt. Im Rahmen der Beobachtungsstudie hat sich deutlich gezeigt, dass die Arbeit an einem physikalischen Task-Board eine Abwechslung darstellt, die sich positiv auf die Motivation auswirkt. Das zeigte sich letztendlich auch dadurch, dass die Arbeit mit der Virtual Task-Board Lösung immer direkt erledigt wurde.

Beim Einsatz der konventionellen Lösung mit Jira Agile zeigten sich hierbei deutliche Unterschiede. Die Statusupdates wurden häufig nicht durchgeführt und die Arbeit mit der digitalen Task-Board Variante wurde vor allem gegen Ende der Studie eher als zusätzlicher Aufwand und nicht als Unterstützung empfunden.

(BK 4) Haptische Rückmeldung:

(KAT 4.1) Einsatz physikalischer Medien: Im Rahmen der Studie ließ sich beobachten, dass im Fall des Virtual Task-Board Teams aktiv Gebrauch von weiteren physikalischen Medien (z.B. Whiteboard, Flip-Chart, handschriftliche Notizen) gemacht wurde. Während dieser Arbeitsphasen ließen sich engagierte Diskussionen zu fachlichen Themen und Problemstellungen beobachten, die durchaus als positives Zeichen bzgl. der Teamarbeit zu verstehen sind.

Im Fall der konventionellen Variante mit Jira Agile konnte diese Beobachtung nicht gemacht werden. Hier konzentrierte sich das Team auf die direkte Arbeit mit Jira Agile. Anforderungsdetails und Notizen wurden direkt in die PMS eingetragen. Entsprechend fanden fachliche Diskussionen wesentlich weniger statt.

(KAT 4.2) Multi-User-Interaktion: Die Arbeit am Virtual Task-Board unterstützt gemeinsame Diskussionen zu einem Thema sehr gut, da alle Informationen übersichtlich dargestellt sind. Bei der gleichzeitigen Bearbeitung einzelner Anforderungen durch mehrere Benutzer ließen sich jedoch Einschränkungen feststellen, da nur ein Eingabegerät (Smart-Pen) zur Verfügung stand. Außerdem hat es teilweise gestört, dass die Karteikarten nicht direkt am Task-Board bearbeitet werden konnten.

Jira Agile brachte ebenfalls Einschränkungen bzgl. Multi-User Bedienung mit sich. Für die gleichzeitige Bearbeitung der Inhalte ist eine Darstellung über den Projektor notwendig. Hierdurch kann nur eine Person aktiv Inhalte einfügen und bearbeiten. Gerade während der Besprechungen ließ sich beobachten, dass diese Person nicht an den fachlichen Diskussionen teilnahm, sondern lediglich mit der Bedienung der PMS beschäftigt war.

(BK 5) Integration:

(KAT 5.1) Verfügbarkeit und Export projektspezifischer Kennzahlen:

Beide Projektteams haben während der Review-Meetings von dieser Möglichkeit

Gebrauch gemacht. Zur Veranschaulichung des Arbeitsfortschritts wurden verschiedene Kennzahlen (z.B. zur Erstellung von Burn-Down-Charts) exportiert oder direkt über die Jira Agile Bedienoberfläche verwendet. Hierbei ließen sich keine nennenswerten Unterschiede zwischen dem Virtual Task-Board Konzept und der konventionellen Benutzung von Jira Agile feststellen, da beide Varianten die gleichen Möglichkeiten bieten.

(KAT 5.2) Integration zusätzlicher Systeme:

Digitale PMS bietet üblicherweise eine Möglichkeit um Drittsysteme (z.B. *Source Code Management* (SCM), Verlinkung digitaler Dokumente) einzubinden. Im Rahmen der Beobachtungsstudie fiel auf, dass diese Möglichkeit beim Einsatz des Virtual Task-Board Konzepts nicht genutzt wurde. Ein mögliche Erklärung ist, dass die natürliche Benutzerschnittstelle diese Möglichkeit nicht bietet und die Nutzung der browserbasierten Oberfläche notwendig wäre. Im Gegensatz dazu nutzte das zweite Team mit Jira Agile diese Möglichkeit zur Integration der verwendeten SCM Lösung.

(BK 7) Mehraufwand:

(KAT 7.1) Vorbereitungszeit von Besprechungen und Prozessphasen:

Das Virtual Task-Board Konzept bietet den Vorteil, dass die Anforderungs- und Statusinformationen zum Projekt jederzeit verfügbar und sichtbar sind. Im Rahmen der Beobachtungsstudie konnte deshalb die Vorbereitungszeit (z.B. für das Standup-Meeting) eingespart werden. Im Vergleich dazu musste bei der konventionellen Lösung mit Jira Agile zunächst der Projektor vorbereitet und der Raum abgedunkelt werden. Dieser Mehraufwand führte gelegentlich dazu, dass auf den Einsatz der PMS verzichtet wurde und notwendige Informationen während der Besprechung nicht bereitstanden.

(KAT 7.2) Spontanes Hinzufügen und Bearbeiten von Inhalten:

Mit Blick auf das Virtual Task-Board Konzept muss dieser Aspekt differenziert betrachtet werden. Das Erfassen von Informationen und das Ändern von Bearbeitungszuständen konnte spontan und ohne Vorbereitung erfolgen, weil hierfür direkt mit den Karteikarten am physikalischen Task-Board gearbeitet wurde. Zum Ergänzen bzw. Abändern von Informationen auf den Karteikarten ist jedoch die entsprechende Smartphone-Anwendung sowie der Smart-Pen nötig, wodurch ein gewisser Mehraufwand entstand (z.B. Anschalten und Verbinden des Smart-Pens). Im Vergleich dazu konnte die Bearbeitung von Projektinformationen und Status gut mit Hilfe der konventionellen Jira Agile Lösung in die Entwicklungsphasen integriert werden, weil die Mitglieder des Projektteams ohnehin an den Arbeitsrechnern tätig waren und somit schnellen Zugriff hatten. Während Besprechungen oder Arbeitsphasen, die nicht an den Arbeitsrechnern stattfanden, war ein spontanes Zugreifen bzw. Bearbeiten der Daten jedoch störend, da ein Arbeitsrechner bzw. teilweise auch ein Projektor benötigt wurde.

(KAT 7.3) Bedienkomplexität:

Die Bedienung des Virtual Task-Board Konzepts wurde vom Projektteam problemlos verstanden und angewendet. Nach einer kurzen Erklärung traten hierbei keinen nennenswerten Probleme auf. Die Bedienung von Jira Agile hingegen führte gelegentlich

zu Wartezeiten während der Besprechungen, weil verschiedene Eingabe- und Einstellmöglichkeiten nicht gefunden wurden. Nach längerer Nutzungsdauer traten Probleme dieser Art jedoch nur noch selten auf. Insgesamt lässt sich festhalten, dass das Virtual Task-Board Konzept eine geringere Einstiegshürde für Erstanwender darstellte.

(KAT 7.4) Effizienz:

Beim Thema Effizienz zeigten sich leichte Vorteile bei der Verwendung von Jira Agile. Gelegentlich ist es notwendig Tasks mit ähnlichen Inhalten am Task-Board anzulegen, weil sie obligatorischer Bestandteil zur Umsetzung jeder User-Story sind. Das Team "Jira Agile" legte zu jeder User-Story eine obligatorische Task zur Implementierung und Ausführung von Unit-Tests an. Diese Art von Tasks konnte mittels digitaler PMS schnell kopiert und angelegt werden. Das Konzept Virtual Task-Board erlaubt diese Vorgehensweise nicht - hier mussten händisch Karten geschrieben werden.

(BK 10) Kommunikation:

(KAT 10.1) Transparenz und spontane Gespräche:

Bezüglich dieser Kategorie unterscheidet sich die Virtual Task-Board Lösung stark von der konventionellen Lösung mit Jira Agile. Es gab zahlreiche Situationen, die zeigten, dass der Einsatz der physikalischen Benutzerschnittstelle für wesentlich mehr Reaktionen und Feedback aus dem Projektteam sorgte. Viele Statusänderungen am Task-Board führten zu Diskussionen, wodurch z.B. unklare Anforderungen aufgedeckt und Missverständnisse geklärt wurden. Diese Reaktionen ließen sich beim Einsatz der konventionellen Lösung nicht beobachten. Hier wurden Statusänderungen häufig erst spät bemerkt und konkrete Reaktionen blieben aus.

(KAT 10.2) Embodiment sorgt länger für Konzentration und Aufmerksamkeit während der Diskussionen:

Während der Arbeit mit dem Virtual Task-Board stehen die Teammitglieder immer wieder auf, um Statusupdates durchzuführen oder einzelne Story-Cards zu bearbeiten. Gerade gegen Ende der Studie fiel auf, dass sich das positiv auf Konzentration und Aufmerksamkeit der Projektgruppe auswirkte. Beim Einsatz der konventionellen Lösung blieben die Teammitglieder auch während der Besprechungen meist an ihrem Arbeitsrechner sitzen und einzelne Teammitglieder beteiligten sich kaum noch an der Diskussion.

(KAT 10.3) Multiuser-Interaktion wird erleichtert und regt zu Gesprächen an:

Die physikalische Benutzerschnittstelle des Virtual Task-Boards ermöglicht gleichzeitiges Arbeiten mehrerer Teammitglieder am Task-Board (z.B. Statusaktualisierungen, Klärung von Anforderungsdetails, etc.). Im Rahmen der Studie wurde diese Möglichkeit recht häufig in Anspruch genommen, wodurch fachliche Diskussionen entstanden. Die konventionelle Lösung mit Jira Agile erlaubt ebenfalls die gleichzeitige Bearbeitung einzelner Inhalte am Task-Board. Der Unterschied dabei ist jedoch, dass jeder Nutzer

an seinem Arbeitsrechner sitzt, wodurch Einschränkungen bei Transparenz und Kommunikation beobachtbar waren. Gleichzeitiges Editieren einzelner Anforderungen war im Rahmen der Studie nicht möglich, da nur ein Smart-Pen zur Verfügung stand. Mit Hilfe mehrerer Smart-Pens wäre das aber problemlos realisierbar. Dennoch fließt dieser Aspekt aufgrund der genannten Einschränkung nicht in die Gesamtbewertung mit ein.

Beobachtete Konzeptschwächen und Optimierungsmöglichkeiten:

An dieser Stelle werden die Beobachtungen bzgl. des ergänzenden Studienziels (vgl. 6.1) beschrieben. Es werden sowohl konzeptionelle als auch technische Schwächen bzw. Optimierungsmöglichkeiten des Virtual Task-Board Konzepts aufgelistet:

1. **Identifikation der Benutzer fehlt:** Das momentane Konzept beinhaltet keine Möglichkeit einzelne Interaktionen (z.B. Hinzufügen / Bearbeiten von Story-Cards) bestimmten Benutzern zuzuordnen. Entsprechend kann diese Information auch nicht in Jira Agile hinterlegt werden. Im lokalen Team spielt das im Regelfall keine große Rolle, da die Verantwortlichkeiten und Aufgaben bekannt sind. In einem entfernt arbeitenden Team ist die Identifikation und Zuordnung von Benutzern jedoch eine wichtige Information, um im Bedarfsfall den richtigen Ansprechpartner zu kennen. Um die Transparenz auch teamübergreifend zu gewährleisten muss das Lösungskonzept an dieser Stelle ergänzt werden.
2. **Kartenregistrierung vor Bearbeitung:** Bevor der Inhalt einer Karte am Task-Board bearbeitet werden kann ist eine Registrierung durch das Nachzeichnen einer eindeutigen Identifikationsnummer mittels Smart-Pen erforderlich. Nur so kann die Verknüpfung zwischen physikalischer Karte und dem zugehörigen Datensatz in Jira Agile hergestellt werden. Es konnte jedoch beobachtet werden, dass dieser Schritt teilweise vergessen wurde. Die Konsequenz ist, dass der Inhalt nicht digital erfasst werden kann und es zu entsprechenden Abweichungen zwischen Datensatz und den Informationen auf der physikalischen Karte kommt. Eine unkomplizierte Lösungsmöglichkeit wäre z.B. ein akustisches Signal, wenn die Registrierung vergessen wurde. Generell wäre jedoch ein Konzept wünschenswert, das ohne die Registrierung der Karte durch den Benutzer auskommt.
3. **Durchstreichen von Texten:** Die im Prototypen eingesetzte HWR unterstützt das Streichen von Texten. Leider kam es im Rahmen der Studie an dieser Stelle häufiger zu Fehlinterpretationen. Im Rahmen der Beobachtungsstudie wurde diese Einschränkung als besonders störend empfunden. Eine entsprechend zuverlässig funktionierende Lösung ist aus unserer Sicht zwingend erforderlich.
4. **Bearbeiten der Karteikarten direkt am Task-Board:** Eine weitere technische Einschränkung bringt der eingesetzte Smart-Pen mit sich. Damit er funktioniert ist es erforderlich den zugehörigen Empfänger auf der dafür vorgesehen

Stelle der Story-Card zu platzieren. Aus diesem Grund ist eine direkte Bearbeitung der Inhalte am Task-Board nicht möglich. Die Karte muss zur Bearbeitung auf einer ebenen Fläche abgelegt werden. Das stört den Arbeitsfluss. Im Rahmen der Studie wurde das Problem durch einen Tisch gelöst, der direkt neben das Task-Board gestellt wurde. Dauerhaft wäre aber eine Smart-Pen Lösung, die den Einsatz eines separaten Empfängers nicht benötigt sinnvoll.

5. **Verfügbarkeit und Einsatzbereitschaft des Smart-Pens:** Der eingesetzte Smart-Pen schaltete sich nach längerem Nichtgebrauch selbstständig ab. Für den spontanen Einsatz ist das leider sehr störend, da das Herstellen die Bluetooth-Verbindung nach erneutem einschalten etwas Zeit in Anspruch nimmt. Auch dieser Aspekt stört den Arbeitsfluss und sollte zukünftig vermieden werden. Zudem war ein einzelner Smart-Pen trotz der geringen Gruppengröße zu wenig. In manchen Situationen wäre ein weiteres Gerät notwendig gewesen, um die Verfügbarkeit zu jeder Zeit gewährleisten zu können.

6.3.2 Ergebnisse der schriftlichen Befragung

Die Ergebnisse der schriftlichen Befragung müssen nach den beiden Erhebungsmethoden (klassischer Fragebogen vs. QST) differenziert werden. Ein erster Überblick über die Ergebnisse der klassischen Erhebungsmethode kann Abb. 6.3 entnommen werden. In den Abbildungen 6.4 sowie 6.8 sind die Ergebnisse der Datenerhebung mit Hilfe der QST dargestellt.

Bevor die Ergebnisse im einzelnen vorgestellt und diskutiert werden (vgl. Abs. 6.3.2.3), wird das Vorgehen bei der Auswertung (vgl. Abs. 6.3.2.1) sowie die Prüfung der Modellgüte mit Hilfe des VIFs (vgl. Abs. 6.3.2.2) beschrieben.

6.3.2.1 Auswertung der schriftlichen Befragung

Die Bewertung der einzelnen Indikatoren im Messmodell entspricht dem arithmetischen Mittel der abgegebenen Einzelbewertungen und berechnet sich folgendermaßen:

Formel 6.1. Berechnung der Gesamtbewertung eines Indikators:

$$I = \frac{1}{n} \cdot \sum_{k=1}^n E_k$$

I: Gesamtbewertung eines Indikators

n: Anzahl Befragungsteilnehmer

E_k: Einzelmesswert bzgl. eines Indikators der k-ten befragten Person

Die Gesamtbewertungen der einzelnen Bewertungskriterien errechnet sich entsprechend der in Abb. 2.4 angegebenen Formel für formative Messmodelle, durch die Linearkombination der einzelnen Indikatoren mit ihren Gewichten plus den Fehlerterm. Im Rahmen der

Evaluationsstudie wird jedoch von einer Gleichgewichtung aller Indikatoren ausgegangen. Zudem wird auf den Fehlerterm verzichtet, weil diesbezüglich keine Aussage getroffen werden kann. In einem letzten Schritt wird eine Skalierung in den Wertebereich der verwendeten Ratingskala durchgeführt. Da alle Bewertungskriterien mit Hilfe von vier Indikatoren gemessen wurden ergibt sich folgende Formel:

Formel 6.2. Berechnung der Gesamtbewertung eines Bewertungskriteriums:

$$BK = \frac{1}{4} \cdot \sum_{k=1}^4 I_k$$

BK: Gesamtbewertung eines Bewertungskriteriums

I_k: k-ter Indikator eines Bewertungskriteriums

6.3.2.2 Güteprüfung der verwendeten Messmodelle

Die Güteprüfung der verwendeten Messmodelle wurde wie in Abs. 2.2.4.2 beschrieben mit Hilfe des VIF durchgeführt. Für beide Erhebungstechniken (klassischer Fragebogen und QST) wurden die VIFs aller Indikatoren berechnet.¹ In beiden Fällen war die Güte der eingesetzten Messmodelle laut Diamantopoulos et al. [23] ausreichend (VIF < 5). Die Forderung nach geringer Multikollinearität zwischen den einzelnen Indikatoren der Messmodelle ist demnach erfüllt, weshalb eine Anpassung nicht notwendig ist.

Die nachfolgenden Tabellen 6.6 und 6.7 listen die VIFs aller Indikatoren vollständig auf:

Tabelle 6.6: Übersicht der VIFs mit der QST

Indikator	VIF	Indikator	VIF
I 1.1	1,12	I 3.3	0,93
I 1.2	1,1	I 3.4	0,85
I 1.3	1,51	I 4.1	0,76
I 1.4	1,06	I 4.2	0,86
I 2.1	1,12	I 4.3	0,90
I 2.2	1,18	I 4.4	0,79
I 2.3	0,84	I 5.1	3,10
I 2.4	0,96	I 5.2	1,80
I 3.1	0,83	I 5.3	1,82
I 3.2	0,85	I 5.4	0,88

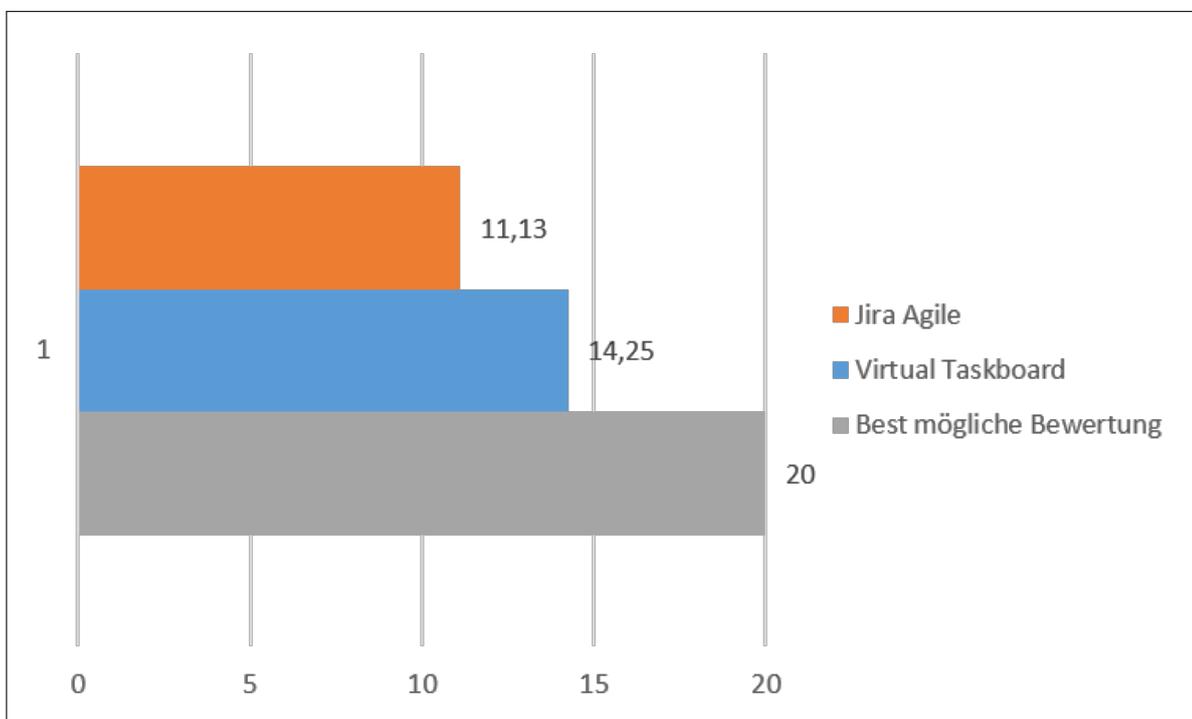
¹Die Berechnungen wurden mit Hilfe der Analyse-Funktion "Regression" in Microsoft Excel durchgeführt.

Tabelle 6.7: Übersicht der VIFs mit der klassischen Erhebungstechnik

Indikator	VIF	Indikator	VIF
I 1.1	0,83	I 3.3	1,08
I 1.2	1,18	I 3.4	0,88
I 1.3	1,12	I 4.1	1,26
I 1.4	1,37	I 4.2	1,23
I 2.1	1,43	I 4.3	1,60
I 2.2	1,26	I 4.4	1,56
I 2.3	1,20	I 5.1	0,85
I 2.4	1,76	I 5.2	0,80
I 3.1	1,77	I 5.3	0,82
I 3.2	1,19	I 5.4	0,96

6.3.2.3 Beschreibung der Ergebnisse

Die Ergebnisse der schriftlichen Befragungen werden in den Abb. 6.2 bis 6.4 mit Hilfe von Säulendiagrammen bzw. Balkendiagrammen gegenübergestellt. Die Ergebnisse des Lösungskonzepts Virtual Task-Board sind durchgehend blau und die Ergebnisse der konventionelle PMS Jira Agile orange dargestellt. Der ursprüngliche Wertebereich aller Bewertungskriterien (von -2 bis 2) wurde in den positiven Bereich (von 0 bis vier) verschoben, weil dem Vorzeichen keine weitere Bedeutung mehr zukommt und die nachfolgenden Darstellungen entsprechend vereinfacht werden.

**Abbildung 6.2:** Gesamtergebnis der schriftl. Befragung mit klassischer Erhebung.

Die klassische Erhebung erlaubt einen direkten Gesamtvergleich des Lösungskonzepts Virtual Task-Board mit der konventionellen PMS Jira Agile. Abb. 6.2 zeigt die aufsummierten Einzelergebnisse aller untersuchten Bewertungskriterien, wodurch sich ein Wertebereich von 0 (theoretisch schlechtestes Ergebnis) bis 20 (theoretisch bestes Gesamtergebnis) ergibt. Das Lösungskonzept Virtual Task-Board wurde mit einem Ergebnis von 14,25 also besser bewertet als die konventionelle Lösung mit 11,13.

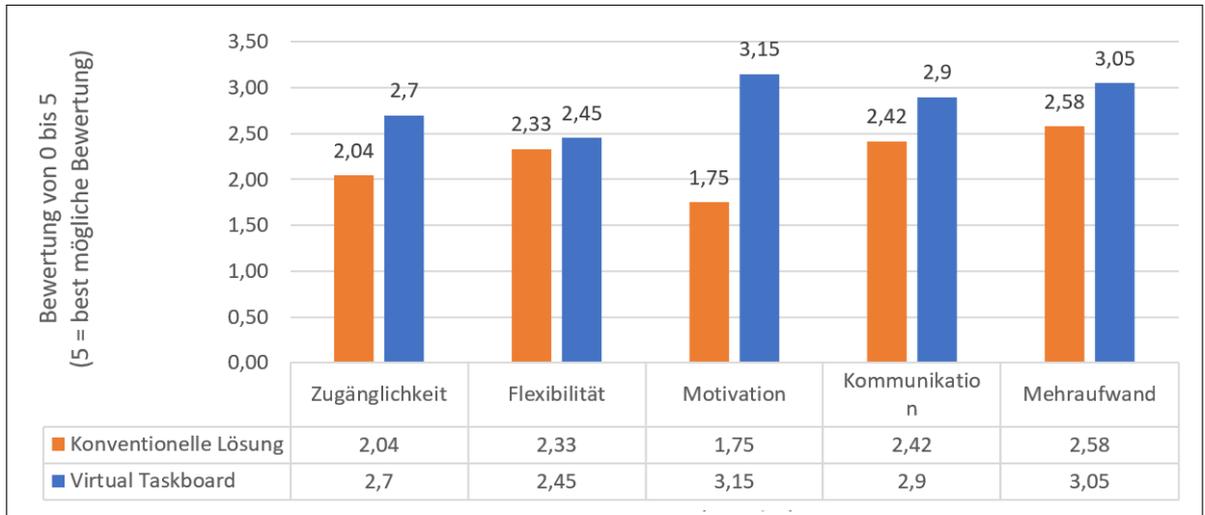


Abbildung 6.3: Einzelergebnisse der schriftl. Befragung mit klassischer Erhebung.

Abb. 6.3 zeigt nun die Einzelergebnisse der betrachteten Bewertungskriterien. Für das Lösungskonzept Virtual Task-Board konnte in allen Bereichen eine Verbesserung festgestellt werden. Besonders hervorzuheben ist die deutliche Steigerung der gemessenen Motivation im Projektteam.

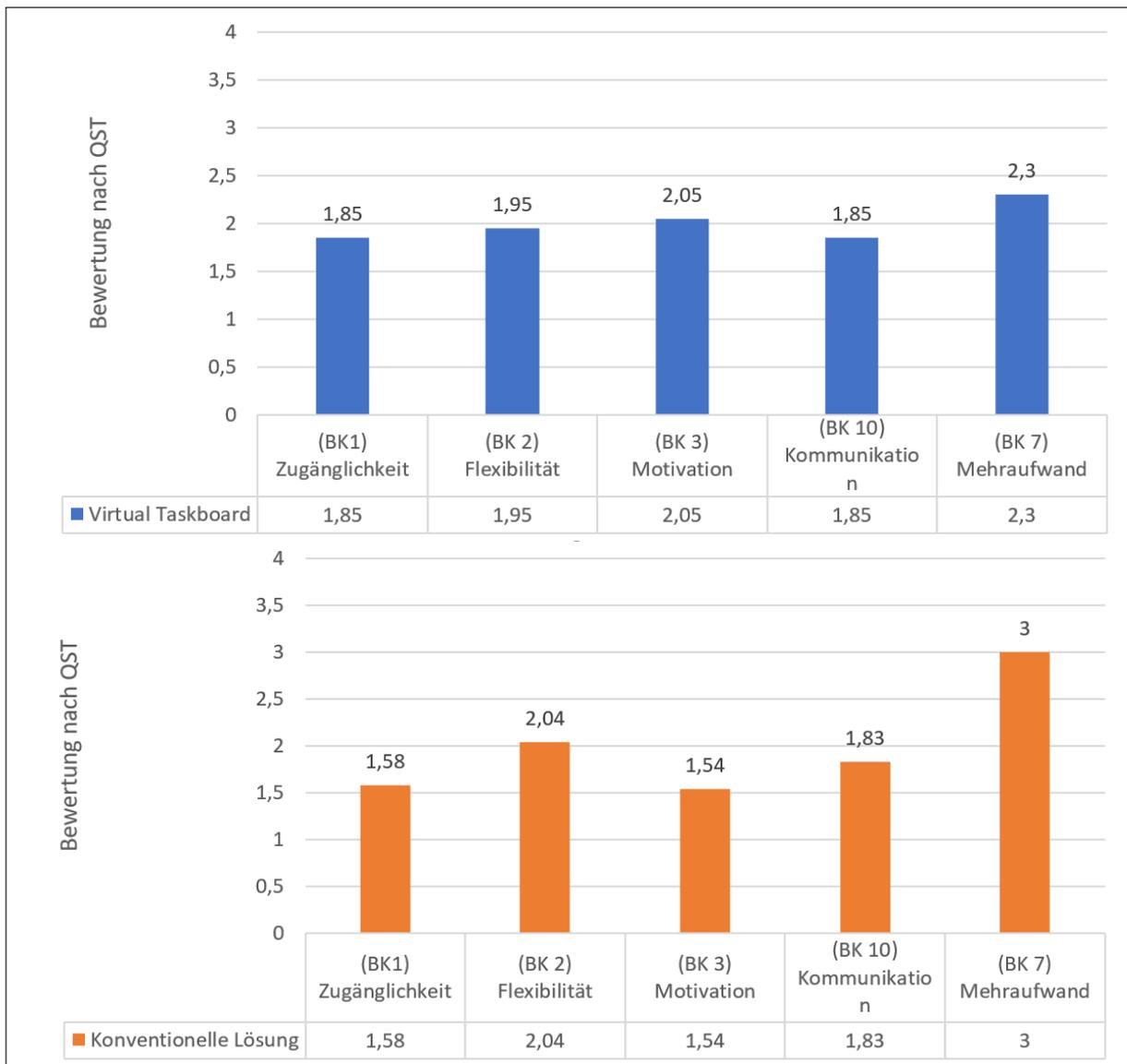


Abbildung 6.4: Ergebnisse der schriftl. Befragung mit Hilfe der QST.

Tabelle 6.8: Streumaße zur schriftl. Befragung mit Hilfe der QST.

	Standardabweichung σ	Streubreite ω
<i>Virtual Task-Board</i>	0,167	0,45
<i>Jira Agile</i>	0,532	1,46

Die Besonderheit der QST mit erzwungener Normalverteilung gegenüber der klassischen Datenerhebung zeigt sich in Abb. 6.4. Eine absolute Gesamtbewertung beider Konzepte wie in Abb. 6.2 ist nun nicht mehr möglich, da die Summe aller Einzelergebnisse in beiden Fällen identisch wäre. Es zeigt sich jedoch, welche Aspekte aus Sicht der Studienteilnehmer positiver bzw. negativer wahrgenommen wurden. Die Ergebnisse der einzelnen

Bewertungskriterien können nur relativ zueinander interpretiert werden. Die Befragungsergebnisse der konventionellen Lösung zeigen Stärken in den Bereichen Mehraufwand (BK 7) und Flexibilität (BK 2). In Relation dazu schneidet die konventionelle Lösung in den Bereichen Zugänglichkeit (BK 1), Motivation (BK 3) und Kommunikation (BK 10) weniger gut ab. Dieses Ergebnis bestätigt tendenziell die Erkenntnisse der in 3.1.3.1 gelisteten Forschungsarbeiten, dass die Schwächen konventioneller PMS vor allem in den Bereichen Kommunikation, Motivation und Zugänglichkeit in Erscheinung treten. Im Vergleich dazu zeigen die Ergebnisse der Virtual Task-Board Lösung auf Basis der QST ein ausgeglicheneres Bild. Die Streuung ist wesentlich geringer weshalb zunächst keine Aussage bzgl. auffallender Stärken bzw. Schwächen möglich ist. Erst durch die Einbeziehung der in Abb. 6.2 gezeigten Ergebnisse auf Basis der Befragung mit klassischem Rangordnungsverfahren ist die Annahme zulässig, dass durch den Einsatz des Virtual Task-Board Konzepts Verbesserungen bzgl. der genannten Schwachpunkte (Kommunikation, Motivation, Zugänglichkeit) erzielt werden konnten. Besonders nennenswert ist zudem die deutlich bessere Bewertung der Motivation (BK 3) im Falle der Virtual Task-Board Lösung.

6.3.3 Ergebnisse der System-Usability-Scale

Die Ergebnisauswertung der System-Usability-Scale [12] (vgl. Abs. 2.2.4.3) basiert ausschließlich auf den ausgefüllten Fragebogen der Studienteilnehmer (vgl. C.3) und ist denkbar einfach. Es werden alle Bewertungen aufsummiert und anschließend mit der Konstante 2,5 multipliziert. Man erhält einen Wert zwischen 0 und 100 der als Prozentsatz interpretiert werden kann (0%: schlechteste vorstellbare Anwendung, 100% beste vorstellbare Anwendung). Abb. 6.5 zeigt die Ergebnisse:

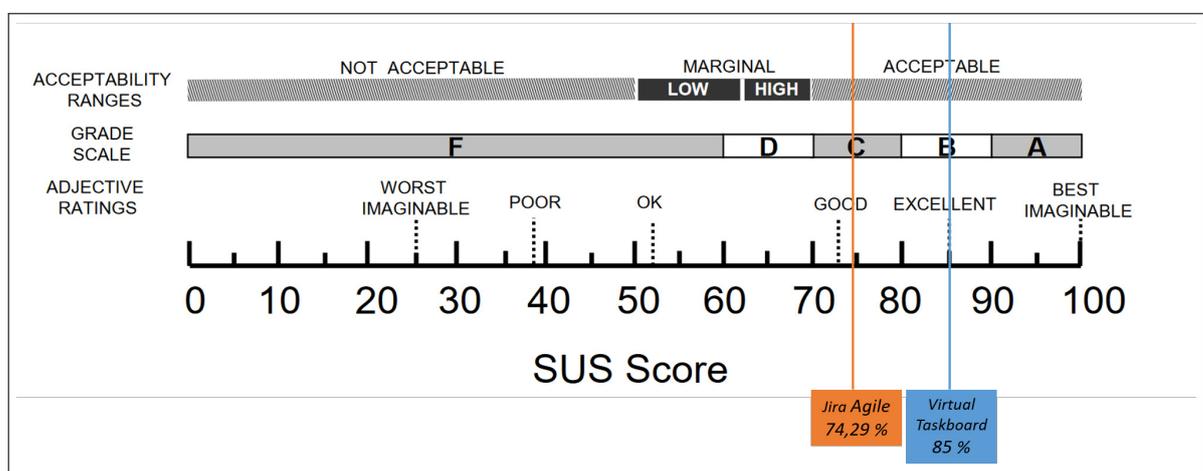


Abbildung 6.5: Bewertung der Gebrauchstauglichkeit mit Hilfe der SUS (Quelle: Bangor et al. [7])

Beide Ansätze bewegen sich laut Bangor et al. [6] im akzeptablen Bereich, der ab 60% beginnt. Innerhalb dieses Bereichs ist jedoch eine deutlich Verbesserung der gemessenen

Gebrauchstauglichkeit durch den Einsatz der Virtual-Task-Board Lösung sichtbar. Zieht man die mittels Schulnotensystem (vgl. Abs. 2.2.4.3) nach Bangor et al. [7] heran bedeutet das Ergebnis eine Verbesserung um eine Notenstufe von "C" nach "B".

6.3.4 Zusammenführung der Teilergebnisse und Hypothesenbezug

Die eben vorgestellten Einzelergebnisse werden nun im Zusammenhang betrachtet und in Bezug zu den in Abs. 6.1 formulierten Hypothesen gebracht.

Bezugnahme zur Hypothese H1: *Im Vergleich zu konventionellen, browserbasierten Benutzerschnittstellen agiler Projektmanagement-Software lassen sich beim Einsatz von Virtual Task-Board Verbesserungen in den Bereichen Kommunikation, Transparenz und Motivation innerhalb lokaler, agiler Projektteams feststellen.*

Betrachtet man die in Tab. 6.5 gelisteten Ergebnisse der Beobachtungsstudie lassen sich Verbesserungen bei fünf von sieben Bewertungskriterien feststellen. Bezugnehmend auf Hypothese H1 stehen vor allem die Kriterien Kommunikation (BK 10), Motivation (BK 3) und Zugänglichkeit (BK 1) im Fokus. In 6.2.1 wurde bereits darauf hingewiesen, dass das Bewertungskriterium Zugänglichkeit synonym für Transparenz steht und die Bezeichnung lediglich beibehalten wurde, weil sie von Katsma et al. [46] entsprechend definiert und veröffentlicht wurde. Mit Ausnahme der Lesbarkeit (Kategorie innerhalb des Bewertungskriteriums Zugänglichkeit) wurden alle Kategorien dieser drei Bewertungskriterien im Fall des Virtual Task-Board Konzepts besser als im Fall der konventionellen Lösung mit Jira Agile bewertet. Es kann also festgehalten werden, dass Hypothese H1 durch die vorliegenden Ergebnisse der Beobachtungsstudie unterstützt wird. Die restlichen Bewertungskriterien werden nicht direkt von Hypothese H1 adressiert. Dennoch sind sie als Bestandteil des Bewertungsmodells nach Katsma et al. [46] aussagekräftig bzgl. der Eignung einer Projektmanagementlösung für die agile Projektarbeit. Entsprechend kann festgehalten werden, dass sich bzgl. Flexibilität (BK 2), Haptische Rückmeldung (BK 4) positive Effekte durch den Einsatz der Virtual Task-Board erkennen ließen. Nachteile ergaben sich innerhalb des Bewertungskriteriums Integration (BK 5) bzgl. der Integration zusätzlicher Systeme. Es lies sich feststellen, dass die Möglichkeit zur Integration von Drittsystemen (wie z.B. SCM) kaum genutzt wurde. Die Erklärung hierfür ist vermutlich, dass diese Funktion durch die natürliche Benutzerschnittstelle der Virtual Task-Board Lösung nicht unterstützt wird und ein Umweg über die browserbasierte Schnittstelle notwendig wäre. Eine Erweiterung des Virtual Task-Board Konzepts an dieser Stelle sollte für zukünftige Weiterentwicklungen also unbedingt diskutiert werden. Bzgl. des Kriteriums Mehraufwand (BK 7) konnte keine Gesamttendenz in eine positive bzw. negative Richtung festgestellt werden.

Die Erkenntnisse der Beobachtungsstudie können auch durch die schriftliche Befragung mittels klassischer Erhebung (vgl. Abb. 6.3) bestätigt werden. Das Ergebnis ist besonders wichtig, da es die direkte Einschätzung der Studienteilnehmer widerspiegelt. Für alle gemessenen Bewertungskriterien ergab die Befragung eine Verbesserung gegenüber der

konventionellen Lösung. Besonders im Bereich der Motivation (BK 3) konnten starke Verbesserungen (von 1,75 bei Jira Agile zu 3,15 bei Virtual Task-Board) festgestellt werden. Die Hypothese (H1) wird demnach durch die Ergebnisse der schriftlichen Befragung nach klassischem Rangordnungsverfahren ebenfalls unterstützt.

Die schriftliche Befragung mittels QST ist ergänzend zu betrachten, da sie einen direkten Vergleich zwischen dem Lösungskonzept Virtual Task-Board und der konventionellen PMS nur bedingt zulässt (vgl. 2.2.4.2). Folgt man der Ergebnisinterpretation in 6.3.2.3 finden sich auch hier Hinweise, dass das Lösungskonzept Virtual Task-Board gerade bzgl. der Schwachpunkte konventioneller PMS (Zugänglichkeit (BK 1), Motivation (BK 3) und Kommunikation (BK 10) zu Verbesserungen geführt hat.

Bezugnahme zur Hypothese H2: *Der Einsatz von Virtual Task-Board innerhalb agiler Projektteams wirkt sich im Vergleich zu konventionellen, browserbasierten Benutzerschnittstellen agiler Projektmanagement-Software nicht negativ auf die Gebrauchstauglichkeit aus.*

Die Gebrauchstauglichkeit ist ein wichtiges Qualitätskriterium jeder Software, die Interaktionen mit menschlichen Benutzern vorsieht. Das Bewertungsmodell in Abs. 6.2.1 adressiert die Gebrauchstauglichkeit gem. DIN EN ISO 9241 allenfalls indirekt, da der Einfluss auf die einzelnen Bewertungskriterien nicht genau bekannt ist. Aus diesen Gründen wurde die SUS eingesetzt, um eine Aussage bzgl. der Gebrauchstauglichkeit zu erhalten. Laut Brooke [12] ist zu beachten, dass die Ergebnisse der SUS keine exakte Messung sondern lediglich eine erste Indikation der Gebrauchstauglichkeit erlauben. Zu Überprüfung von Hypothese H2 wurde eine Befragung bzgl. der Gesamtkonzepte Virtual Task-Board und Jira Agile durchgeführt. Hierbei wurde das Virtual Task-Board Konzept mit 85% deutlich besser bewertet als die konventionelle PMS mit 74,29%. Gem. Brooke [12] ist das ein Hinweis, dass die Gebrauchstauglichkeit des Virtual Task-Board Konzepts verbessert werden konnte. Entsprechend wird die Hypothese H2 durch die Ergebnisse der Befragung nach SUS gestützt.

6.3.5 Limitationen der Evaluationsstudie

Die Evaluationsstudie unterliegt verschiedenen Limitationen. Die wichtigsten Limitationen werden nachfolgend kurz vorgestellt, um die Validität der Studienergebnisse entsprechend einordnen zu können. Insgesamt sind vier offensichtliche Einschränkungen bzgl. der eingebrachten Hypothesen (H1, H2) zu nennen:

Beobachter als Teil des Forschungsteams: Trotz genauer Beobachtungsrichtlinien ist es nicht möglich, eine vollständig objektive Beobachtung und Protokollierung durchzuführen. Dieser Aspekt wird durch die Tatsache verstärkt, dass der Beobachter Teil des Forschungsteams war. Es besteht die reale Gefahr, dass eine gewisse Erwartungshaltung bzgl. auftretender Ereignisse und Auswirkungen der verglichenen Konzepte die Sichtweise des Beobachters beeinflusst haben. So fallen erwartete Aspekte

(z.B. gesteigerte Motivation im Team) ggf. stärker auf. Eine entsprechende Beeinflussung der Forschungsergebnisse kann demnach nicht vollständig ausgeschlossen werden.

Vergleichbarkeit der Projektteams: Trotz der vorbereitenden Maßnahmen während des Semesters kann nicht vollkommen sichergestellt werden, dass die Vorkenntnisse von Experimental- und Kontrollgruppe identisch sind. Außerdem kann eine gewisse Verfälschung der Ergebnisse durch nicht betrachtete Faktoren eine Rolle spielen. So ist es z.B. möglich, dass eine Gruppe aufgrund sozialer Aspekte besser harmonisiert hat und deshalb die Motivation im Team höher war. Weiterhin können die unterschiedlichen Projektthemen Einfluss auf das Arbeitsverhalten in der Gruppe genommen haben. Ein höhere Validität der Ergebnisse wäre durch eine wiederholte Durchführung des qualitativen Experiments mit weiteren Versuchsgruppen erreichbar.

Versuchsdurchführung unter Laborbedingungen: Die Arbeitssituation während der Evaluationsstudie war künstlich arrangiert. Ein wesentlicher Unterschied war z.B. die Sprint-Länge von lediglich zwei Tagen pro Team. Weiterhin handelte es sich um studentische Teams, die losgelöst vom industriellen Umfeld und den damit zusammenhängenden Verantwortlichkeiten bzw. Abhängigkeiten arbeiten konnten. Es kann also nicht ausgeschlossen werden, dass der Einsatz der Virtual Task-Board Lösung im professionellen Umfeld zu anderen Ergebnissen geführt hätte.

Kapitel 7

Zusammenfassung und Ausblick

7.1 Stellungnahme zu Forschungszielen und Arbeitsergebnissen

Dieser Absatz enthält eine kurze Stellungnahme bzgl. der in 1.3 aufgeführten Forschungsziele und den jeweiligen Ergebnissen dieser Arbeit.

Stellungnahme bzgl. Z1: *Es soll erforscht werden, wie konventionelle PMS im Rahmen verteilter, agiler Entwicklungsprojekte eingesetzt wird, um eine Wissensbasis zur Erarbeitung des Lösungskonzepts bereitzustellen.*

Z1 stellt die Grundlage der Anforderungsdefinition für das in **Z** geforderte Lösungskonzept dar. Im Wesentlichen wurde dieses Ziel definiert, um eine ausreichende Wissensbasis bzgl. des PMS Einsatzes innerhalb agiler Softwareentwicklung bereitzustellen.

Ein wichtiger Beitrag zur Zielerreichung ist zunächst die in Kap. 5 beschriebene Vorstudie, innerhalb der mit Hilfe einer Einzelfallanalyse (vgl. 2.2) der Einsatz von PMS in insgesamt fünf Fällen im industriellen Umfeld untersucht wurde. Die Erreichung von Z1 ist an insgesamt drei untergeordnete Ziele geknüpft. Zur Untersuchung von **Z1.1** wurden die mit der PMS in Zusammenhang stehenden Arbeitsabläufe und Prozesse der fünf betrachteten Projekte erfasst und dokumentiert. Auf Basis der Arbeitsabläufe und den damit zusammenhängenden Problemstellen (z.B. Inkonsistenzen und Redundanzen bzgl. der erfassten Projektdaten, Mehraufwand zur Pflege beider Ansätze, etc.) konnten vier wichtige Grundsätze (vgl. 4.2.5) zur Erarbeitung des Lösungskonzepts definiert werden. Ein weiterer Aspekt der Vorstudie war die Erhebung von Vor- und Nachteilen des PMS Einsatzes aus Sicht der befragten Personen (vgl. **Z1.2**). Dieser Schritt war notwendig, um konkrete Bedürfnisse und Probleme aus Sicht der Anwender im Lösungskonzept berücksichtigen zu können und somit ein praxisorientiertes Konzept zu erhalten. Auf Basis der in Z1.1 und Z1.2 gewonnenen Ergebnisse konnten anschließend Schlüsselfaktoren bzw. grundlegende Verbesserungspotenziale benannt werden (vgl. 4.2.6), die innerhalb des Lösungskonzepts berücksichtigt wurden (vgl. **Z1.3**).

Stellungnahme bzgl. Z2: *Konzeption einer natürlichen Benutzerschnittstelle zur automatisierten Erfassung projektrelevanter Daten im Kontext verteilter, agiler Projekte.*

Zur Erreichung von Z2 wurden zunächst das untergeordnete Ziel **Z2.1** in Kap. 5 behandelt, das auf Basis einer vorausgehenden Anforderungsdefinition einen konkreten Vorschlag zur Erfassung projektrelevanter Daten in einer konventionellen PMS mittels natürlicher Benutzerschnittstellen enthält. Das Konzept beinhaltet einen Lösungsvorschlag zur automatisierten Erfassung der Daten auf einem physikalischen, papierbasierten Task-Board sowie Product-Backlog-Chart. Die Grundidee hierbei ist, dass das jeweilige physikalische Board (Task-Board und Product-Backlog-Chart) zur natürlichen Schnittstelle wird, der Anwender in gewohnter Weise damit arbeiten kann und die Notwendigkeit der Datenerfassung geringstmöglichen Einfluss auf die Projektarbeit nimmt. Diese Grundidee konnte zu großen Teilen realisiert werden. In bestimmten Bereichen mussten jedoch auch Abstriche hingenommen werden (z.B. bei der Erfassung handschriftlicher Daten mittels Smart-Pen (vgl. 5.4.1), da ein minimaler Einsatz digitaler Geräte (Smartphone, Smart-Pen, etc.) nicht vollständig vermieden werden konnte. Das untergeordnete Ziel **Z2.2** konnte durch die Umsetzung eines funktionalen Prototypen erreicht werden. Die technischen Details zur Implementierung der notwendigen Komponenten "Smartphone-Anwendung", "Marker-Tracking" und "Dateninterpretation und Synchronisation" sind in 5.7 entsprechend dokumentiert. Die Arbeitsergebnisse zu Z2 bilden die Grundlage der Konzeptevaluation.

Stellungnahme bzgl. Z3: *Evaluation des Gesamtkonzepts (vgl. Z) im Rahmen einer Benutzerstudie, um die Wirksamkeit der in Z1.3 erarbeiteten Verbesserungsmöglichkeiten hinsichtlich Kommunikation, Transparenz und Motivation zu eruieren.*

Ziel Z3 dient der Evaluation des Zielerreichungsgrades bzgl. dem übergreifenden Ziel Z. Hierbei wurde ein Qualitatives Experiment mit Hilfe des realisierten Prototypen durchgeführt (vgl. Kap. 6). Das Experiment vergleicht die Auswirkungen auf die Projektarbeit der neu integrierten, natürlichen Benutzerschnittstelle mit der browserbasierten Schnittstelle konventioneller PMS. Der Vergleich wurde auf Basis eines Bewertungsmodells nach Katsma et al. [46] (vgl. 6.2.1) sowie mit Hilfe eines standardisierten Verfahrens zur Erfassung der Gebrauchstauglichkeit nach DIN EN ISO 9241 durchgeführt. Es wurde jeweils eine Hypothese H1 bzgl. der in Z1 definierten Merkmale (Motivation, Kommunikation und Transparenz) sowie eine Hypothese H2 zur Gebrauchstauglichkeit definiert (vgl. 6.1). Die Ergebnisse der Beobachtungsstudie sowie der schriftlichen Befragung liefern deutliche Hinweise, dass Motivation, Kommunikation und Transparenz im Projektteam mit Hilfe der Virtual Task-Board Lösung verbessert werden konnten und unterstützen damit die Hypothese H1. Die mit Hilfe der SUS erhobenen Ergebnisse deuten ebenfalls auf Verbesserungen bzgl. der Gebrauchstauglichkeit hin und unterstützen somit Hypothese H2.

Stellungnahme bzgl. Z: *Konzept zur Optimierung konventioneller PMS für agile Entwicklungsprojekte auf Basis natürlicher Benutzerschnittstellen, um die Gruppenarbeit hinsichtlich Kommunikation, Transparenz und Motivation innerhalb agiler Teams zu verbessern.*

Basierend auf den untergeordneten Zielen Z1 bis Z3 kann festgehalten werden, dass eine Optimierung konventioneller PMS entsprechend der in Z definierten Merkmale erreicht werden konnte. Dieses Ergebnis verdeutlicht die Potenziale natürlicher Benutzerschnittstellen zur Unterstützung von Gruppenarbeitsprozessen und zeigt, dass die Forschungsziele der Arbeit zu großen Teilen erreicht werden konnten.

7.2 Inhalte der Arbeit

Diese Arbeit setzt sich mit einem Teilproblem zweier widersprüchlicher Trends in der Softwareentwicklung auseinander: Zum einen lässt sich in der Praxis ein starker Trend zu agilen Entwicklungsmethoden feststellen, welche starken Fokus auf direkte Interaktion und Kommunikation zwischen allen Stakeholdern legen. Zum anderen finden immer mehr Softwareentwicklungsprojekte in örtlich verteiltem Kontext statt, wodurch persönlicher Austausch und Absprachen enorm erschwert werden (vgl. 1.2). Dieser Widerspruch setzt sich auch beim Einsatz von Methoden und Werkzeugen zur Unterstützung des Projektmanagements fort. Agile Methoden fordern den Einsatz physikalischer Medien und die Reduktion digitaler PMS auf ein Minimum. Örtlich oder gar global verteilte Projekte benötigen digitale PMS jedoch zwingend, um relevante Projektinformationen auszutauschen. Dieser Konflikt ist Kernbestandteil der Arbeit und wird aus dem Blickwinkel der CSCW-Forschung betrachtet. Definiertes Ziel ist die Optimierung konventioneller PMS mit Hilfe natürlicher Benutzerschnittstellen (Stichwort: Tangible User Interfaces), um die genannten Gegensätze agiler Softwareentwicklung und örtlich verteilter Softwareentwicklung im Bereich PMS abzumildern (vgl. 1.3).

Hierfür wurden zunächst relevante Grundlagen der agilen Softwareentwicklung sowie der aktuelle Stand der Forschung (vgl. Kap. 3) aufgearbeitet und dargestellt. Ebenso wurde der Forschungsbereich CSCW vorgestellt, um die Arbeit entsprechend einzuordnen (vgl. 3.2). Als Grundlage der Anforderungsdefinition und als breite Wissensbasis wurde eine Vorstudie auf Basis eines qualitativen Interviews durchgeführt. Ziel der Vorstudie war ein genaues Verständnis, wie konventionelle PMS in verteilten Softwareentwicklungsprojekten in den Arbeitsprozess eingebunden wird und welche Aspekte aus Sicht der Anwender positiv bzw. negativ zu bewerten sind (vgl. Kap. 4). Hierfür wurden im Rahmen einer Einzelfallanalyse Interviews mit fünf Experten für verteilte, agile Softwareentwicklung durchgeführt und deren aktuelle Projekte untersucht. Auf Basis der zugehörigen Transkripte wurden die Arbeitsabläufe der einzelnen Projekte dokumentiert und ausgewertet. Auffällig hierbei war, dass in vier von fünf Projekten der parallele Einsatz physikalischer Medien und digitaler PMS Probleme verursachte. Informationen werden häufig redundant auf physikalischen Medien und in digitaler PMS verwaltet und bearbeitet. Dies führt zu Inkonsistenzen innerhalb der Projektdaten und erhöhtem Aktualisierungsaufwand bei den Entwicklungsteams. Ein weiteres Problem stellen Medienbrüche zwischen digitaler PMS und physikalischen Medien dar. Sie verursachen erheblichen Aufwand zur manuellen Übertragung relevanter Projektdaten (z.B. Übertragung der Inhalte vom physikalischen Task-Board in die PMS). Neben zusätzlichem Arbeitsaufwand für die Projektteams führen diese manuellen Vorgänge zu geringerer Qualität der Daten bzw. zu Datenverlusten

(z.B. werden Informationen umformuliert, abgekürzt, etc.). Aus den Erkenntnissen der Vorstudie wurden anschließend grundlegende Prinzipien formuliert, die in der nachfolgenden Konzeptionsphase als Entscheidungsgrundlage dienten (vgl. 4.2.5). Weiterhin wurden die in Tab. 4.3 dargestellten Vor- und Nachteile konventioneller PMS während der Anforderungsdefinition für das Lösungskonzept berücksichtigt.

In Kap. 5 wurden die Kernanforderungen an ein Lösungskonzept aus funktionaler und nichtfunktionaler Sicht erarbeitet. Die Grundidee des Lösungskonzepts ist der Einsatz eines konventionellen, papierbasierten Task-Boards als Tangible-User-Interface für konventionelle PMS. Das bedeutet, dass alle darauf dargestellten Daten automatisiert erfasst werden müssen, ohne den gewohnten Arbeitsablauf am Task-Board zu unterbrechen bzw. negativ zu beeinflussen. Anschließend muss eine entsprechende Synchronisation der Daten in eine konventionelle PMS erfolgen. Auf diese Weise können die Grundforderungen nach dezentral verfügbaren Projektdaten erfüllt werden und der bewährte Arbeitsfluss innerhalb der agilen Projektteams kann beibehalten werden. Während der Konzeptionsphase wurde zunächst ein Datenmodell der relevanten Projektdaten erarbeitet, um eine definierte Basis für die weiteren Schritte bereitzustellen (vgl. Abb. 5.1). Anschließend wurden Konzepte zur Erfassung handschriftlicher Inhalte auf Basis gängiger Technologien (z.B. Einsatz von Digitalstiften, Online-Handschrifterkennung) erarbeitet (vgl. 5.4.1.1). Weiterhin wurden ein Konzept zur Zustandserfassung aller Anforderungen auf dem Task-Board erstellt (vgl. 5.4.2.2). Die Grundidee hierfür war der Einsatz einer kamerabasierten Positionserkennung auf Basis von optischen Markierungen (AR-Markern) auf dem Task-Board. Im nächsten Schritt wurde eine Architektur zur Synchronisation der erfassten Daten in konventionelle PMS erarbeitet (vgl. 5.5). Hauptbestandteil des Lösungskonzepts ist die Kombination der genannten Teilkonzepte, um eine nahtlose Integration in den agilen Softwareentwicklungsprozess zu ermöglichen. Als Machbarkeitsnachweis und als Basis für eine anschließende Konzeptevaluation wurde ein voll funktionaler Prototyp (auch Pilotsystem [5] genannt) umgesetzt (vgl. 5.7).

Die Konzeptevaluation konzentriert sich auf den Vergleich konventioneller PMS mit dem im Rahmen der Arbeit vorgestellten Lösungskonzept und deren Einfluss auf die Gruppenarbeit innerhalb agiler Projektteams. Hierzu wurden zunächst geeignete Hypothesen formuliert, die in erster Linie Aussagen zu Motivation, Transparenz und Kommunikation in der Gruppenarbeit treffen (vgl. H1 in 6.1). Weiterhin steht auch die Gebrauchstauglichkeit nach DIN EN ISO 9241 im Fokus der Evaluation (vgl. H2 in 6.1). Grundaufbau der Evaluationsstudie war ein Qualitatives Experiment auf Basis zweier studentischer Projektteams. Die Kontrollgruppe arbeitete dabei mit Hilfe konventioneller PMS und die Experimentalgruppe mit dem prototypisch implementierten Lösungskonzept. Als Erhebungsverfahren kamen schriftliche Befragungen sowie qualitative Beobachtungen zum Einsatz. Beide Erhebungsverfahren wurden auf Basis eines literaturgestützten Evaluationsmodells für PMS durchgeführt (vgl. 6.2.1). Latente Größen wurde mittels eines separat konstruierten Messmodells erhoben. Insgesamt konnten durch den Einsatz des Prototypen Verbesserungen bzgl. der Kernaspekte Kommunikation, Transparenz und Motivation innerhalb der Projektteams im Vergleich zur Kontrollgruppe nachgewiesen werden. Weiterhin wurde im Rahmen der Evaluationsstudie eine Verbesserung der Gebrauchstauglichkeit festgestellt. Insgesamt lässt sich also festhalten, dass mit gängigen

Techniken der CSCW-Forschung und dem Einsatz von TUIs eine Verbesserung der Projektarbeit in regional verteilten Projekten erreichbar ist.

7.3 Konferenzbeiträge und Publikationen im Kontext der Arbeit

Nachfolgend werden die im Zusammenhang mit dieser Arbeit veröffentlichten Konferenzbeiträge kurz zusammengefasst und der Bezug zur vorliegenden Forschungsarbeit erläutert. Insgesamt wurden die folgenden vier Beiträge veröffentlicht:

Raith, Richter, Lindermeier, Klinker - „Identification of Inaccurate Effort Estimates in Agile Software Development“. In: 20th Asia-Pacific Software Engineering Conference 2013 [70]

Diese Veröffentlichung aus dem Jahr 2013 wurde bereits in 3.1.5 detailliert erläutert, weshalb an dieser Stelle nur noch kurz darauf eingegangen wird. Die Arbeit beschäftigte sich mit State-of-the-Art Methoden zur Abschätzung des Entwicklungsaufwands innerhalb agiler Methoden. Kern der Arbeit ist ein Konzept, das die Identifikation ungenauer Aufwandsabschätzungen und damit die effektivere und genauere Planung agiler Softwareprojekte zum Ziel hat. Das Konzept basiert auf der automatisierten Erhebung und Auswertung spezifischer Kennzahlen, die während des Planning-Poker Prozesses¹ erhoben werden. An dieser Stelle besteht auch die Relation zu dieser Arbeit, da auch hier eine physikalische Arbeitsumgebung entsprechend modifiziert werden musste, um digitale Daten erheben zu können. Hierfür wurde ein entsprechendes Konzept zur Durchführung des Planning-Poker Prozesses auf Basis einer Multitouch-Anwendung konzipiert und prototypisch implementiert. Insgesamt zeigte sich im Rahmen einer vergleichenden Studie, dass der Einsatz digitaler Medien innerhalb agiler Prozesse durchaus funktionieren kann, um Daten digital zu erfassen. Jedoch zeigten sich auch Nachteile wie z.B. fehlende Flexibilität (z.B. bei Abweichungen vom vorgegebenen Standardprozess). Die Erkenntnisse mit Bezug zu dieser Arbeit waren unter anderem, dass selbst aufwändige grafische Benutzeroberflächen (in diesem Fall eine Multitouch-Anwendung) immer noch Nachteile gegenüber der ausschließlichen Verwendung physikalischer Medien mit sich bringen. Als Konsequenz haben wir im Rahmen der vorliegenden Arbeit auf ein Lösungskonzept mit Fokus auf natürlichen Benutzerschnittstellen (in diesem Fall TUI) gesetzt.

¹Die Methode Planning-Poker ist eine weit verbreitete Technik zur Aufwandsabschätzung von Anforderungen. Der Prozess bezieht alle Entwickler eines agilen Teams mit ein und wird mit Hilfe von physikalischen Karten durchgeführt (deshalb die Namensgebung). Die Methode setzt im Gegensatz zu anderen Schätzmethode nicht auf absolute Schätzwerte, sondern auf eine relative Schätzung. Details dieser Methode sind beispielsweise in "Agile Estimating and Planning" von Mike Cohn [16] beschrieben.

Raith, Lindermeier - "Media Supported Workspaces in Agile Software Development: Doctoral Symposium Paper" In: 38th Computer Software and Applications Conference 2014 [68]

Im Rahmen dieser Veröffentlichung wurden zum ersten Mal die grundlegenden Forschungsfragen dieser Dissertation auf Basis aktueller Literatur begründet und öffentlich zur Diskussion gestellt. Dabei wurde zunächst auf den generellen Konflikt zwischen agilen Methoden und verteilter Softwareentwicklung aufmerksam gemacht. Dieser Konflikt wurde anschließend auf den Teilbereich der agilen Arbeitsumgebungen (vgl. 3.1.3) heruntergebrochen und diskutiert. Am Beispiel eines Task-Boards wurde dann die Idee zur Integration natürlicher Benutzerschnittstellen in agile Arbeitsumgebungen vorgestellt. Als weiter Bestandteil dieser Veröffentlichung wurden Evaluationsmöglichkeiten auf Basis existierender Ansätze (z.B. Distributed Cognition [37]) aus dem Forschungsbereich CSCW betrachtet.

Richter, Raith, Weber - "Problems in Agile Global Software Engineering Projects especially within Traditionally Organised Corporations:[An exploratory semi-structured interview study]" In: 9th International Conference on Computer Science and Software Engineering 2016 [71]

Diese Forschungsarbeit beschäftigt sich ebenfalls mit den Konflikten und Problemen, die sich durch den Einsatz agiler Methoden in verteilten Softwareentwicklungsprojekten ergeben. Im Unterschied zu dieser Arbeit wurde der Konflikt jedoch allgemeiner betrachtet und nicht auf die CSCW Perspektive beschränkt. Im Rahmen dieser explorativen Interviewstudie wurde jedoch auch der Kernkonflikt dieser Arbeit, die Unvereinbarkeit agiler Arbeitsumgebungen mit der Forderung nach dezentral verfügbaren Daten, adressiert. Eine detaillierte Darstellung des Bezugs zu dieser Arbeit findet sich bereits in 3.1.

Raith, Richter, Lindermeier - "How Project-management-tools are used in Agile Practice: Benefits, Drawbacks and Potentials" In: 21st International Database Engineering and Applications Symposium 2017 [69]

Diese Arbeit umfasst die im Rahmen dieser Dissertation durchgeführte Einzelfallanalyse (vgl. Vorstudie in Kap. 4). Dabei wurden Forschungsziele, Forschungsmethodik und Ergebnisse im Detail veröffentlicht. Die Inhalte der Arbeit wurden bereits in Kap. 4 im Detail vorgestellt, weshalb an dieser Stelle auf eine erneute Darstellung verzichtet wird.

7.4 Ausblick

Die Ergebnisse der Arbeit zeigen, dass mit dem Einsatz moderner Benutzerschnittstellen Verbesserungen von Gruppenarbeitsprozessen, die auf Softwareunterstützung angewiesen sind, erreicht werden können. Eine weitere Vertiefung der durchgeführten Forschungsarbeit erscheint demnach durchaus sinnvoll. Ein erster Anknüpfungspunkt sind die in 6.3.5 aufgeführten Limitationen. Eine maßgebliche Einschränkung der Evaluationsstudie war die Durchführung unter Laborbedingungen, also in einer künstlich geschaffenen

Arbeitssituation mit studentischen Teams. Eine entsprechende Ausweitung der Studie im industriellen Kontext könnte weitere Aspekte (positiv wie negativ) ans Tageslicht bringen und die Validität der Ergebnisse verbessern. Gleichzeitig könnte dabei ein externes Beobachterteam eingesetzt werden, um auch an diesem Punkt die Objektivität der Ergebnisse zu erhöhen.

Natürlich gibt es neben den vorgeschlagenen Verbesserungspotenzialen zur durchgeführten Evaluationsstudie auch Erweiterungsmöglichkeiten des vorgestellten Lösungskonzepts, die im Folgenden kurz vorgestellt werden: So wurde beispielsweise die Sichtweise eines Projektteams auf die erfassten Projektdaten eines anderen Projektteams kaum betrachtet. Im Rahmen der Studie wurde angenommen, dass die Darstellungsmöglichkeiten konventioneller PMS hierfür ausreichend ist. An dieser Stelle könnte im Rahmen weiterer Forschungsarbeiten an neuen Konzepten gearbeitet werden. Konkret könnte beispielsweise der Entwicklungsfortschritt entfernter Projektteams an den physikalischen Task-Boards dargestellt werden. Auch hierfür wäre der Einsatz moderner Informationstechnologie denkbar (z.B. Einblendung wichtiger Informationen mittels Projektor, etc.). Auch hier muss das Ziel der reduzierte Einsatz graphischer Benutzeroberflächen in Besprechungen und Gruppenarbeitsszenarien sein. Dem gleichen Grundgedanken folgend, könnte z.B. auch der Einsatz des Digitalstiftes und der zugehörigen Smartphone-Anwendung ersetzt werden. Hierfür müssten die Möglichkeiten von pixelbasierter Handschrifterkennung untersucht werden. Eine Verbesserung in diesem Bereich könnte die Anwender komplett unabhängig vom Einsatz digitaler Hilfsmittel während der Arbeit mit dem Task-Board machen, weil die alleinige Erfassung mittels Kamera ausreichend wäre.

Weitere Verbesserungsmöglichkeiten ergeben sich außerdem, wenn man den Blickwinkel etwas verschiebt und überlegt, welche Funktionen konventioneller PMS in die Welt physikalischer Medien übertragen werden können. Betrachtet man die im Rahmen der Vorstudie erarbeiteten Vorteile konventioneller PMS, so finden sich z.B. Funktionen wie die automatische Filterung von Anforderungen oder eine Möglichkeit zur Verlinkung digitaler Dokumente darunter. Zukünftig wäre auch eine Umsetzung dieser Funktionen auf Basis natürlicher Benutzerschnittstellen denkbar.

Literaturverzeichnis

- [1] Pär J. Ågerfalk, Brian Fitzgerald, Helena Holmström Olsson und Eoin Ó Conchúir. „Benefits of Global Software Development: The Known and Unknown“. In: *Making Globally Distributed Software Development a Success Story*. Springer Berlin Heidelberg, 2008, S. 1–9.
- [2] Sönke Albers. *Methodik der empirischen Forschung*. 3., überarb. und erw. Aufl. Gabler, 2009.
- [3] *ARToolKit - GitHub Repository*. URL: <https://github.com/artoolkit> (besucht am 24.06.2021).
- [4] *Azure DevOps Services - Microsoft Azure*. URL: <https://azure.microsoft.com/de-de/services/devops/> (besucht am 24.06.2021).
- [5] Helmut Balzert und Christof Ebert. *Softwaremanagement*. 2. Aufl. Lehrbuch der Softwaretechnik. Heidelberg: Spektrum, Akad. Verl, 2008. 721 S.
- [6] Aaron Bangor, Philip T. Kortum und James T. Miller. „An Empirical Evaluation of the System Usability Scale“. In: *International Journal of Human-Computer Interaction* 24(6) (2008), S. 574–594.
- [7] Aaron Bangor, Philip Kortum und James Miller. „Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale“. In: *J. Usability Studies* 4(3) (Mai 2009), S. 114–123.
- [8] Kent Beck. *Extreme programming explained: embrace change*. Unter Mitarb. von Cynthia Andres. 2. Aufl. Boston, MA: Addison-Wesley, 2005. 189 S.
- [9] Kent Beck. *Manifesto for Agile Software Development*. 2001. URL: <http://agilemanifesto.org/> (besucht am 24.06.2021).
- [10] Herbert D. Benington. „Production of Large Computer Programs“. In: *IEEE Annals of the History of Computing* 5(4) (Okt. 1983), S. 350–361.
- [11] Ann Blandford und Dominic Furniss. „DiCoT: A Methodology for Applying Distributed Cognition to the Design of Teamworking Systems“. In: *Interactive Systems. Design, Specification, and Verification*. 3941. Springer Berlin Heidelberg, 1. Jan. 2006, S. 26–38.
- [12] John Brooke. „SUS: A quick and dirty usability scale“. In: *Usability Eval. Ind.* 189 (30. Nov. 1995).
- [13] Hans-Bernd Brosius, Alexander Haas und Friederike Koschel. *Methoden der empirischen Kommunikationsforschung*. Wiesbaden: VS Verlag für Sozialwissenschaften, 2016.
- [14] Steven Brown. „A Primer on Q Methodology“. In: *Operant Subjectivity* 16 (1. Jan. 1993).
- [15] Peter Pin-Shan Chen. „The Entity Relationship Model — Toward a Unified View of Data“. In: *Pioneers and Their Contributions to Software Engineering*. Hrsg. von

- Manfred Broy und Ernst Denert. Berlin, Heidelberg: Springer Berlin Heidelberg, 1976, S. 205–234.
- [16] Mike Cohn. *Agile Estimating and Planning*. 1. Aufl. Pearson Education, 1. Nov. 2005.
- [17] Mike Cohn. *User stories applied: for agile software development*. Addison-Wesley signature series. Boston: Addison-Wesley, 2004. 268 S.
- [18] *CollabNet Products - CollabNet VersionOne*. URL: <https://www.collab.net/products> (besucht am 24.06.2021).
- [19] Lee J. Cronbach. „Coefficient alpha and the internal structure of tests“. In: *Psychometrika* 16(3) (Sep. 1951), S. 297–334.
- [20] Jeff Dalton. „Scrum Wall/Scrum Board“. In: *Great Big Agile: An OS for Agile Leaders*. Hrsg. von Jeff Dalton. Berkeley, CA: Apress, 2019, S. 229–230.
- [21] Fred D. Davis. „Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology“. In: *MIS Quarterly* 13(3) (Sep. 1989), S. 319.
- [22] Adamantios Diamantopoulos. „Viewpoint – Export performance measurement: reflective versus formative indicators“. In: *International Marketing Review* 16(6) (Dez. 1999), S. 444–457.
- [23] Adamantios Diamantopoulos und Petra Riefler. „Formative Indikatoren: Einige Anmerkungen zu ihrer Art, Validität und Multikollinearität“. In: *Zeitschrift für Betriebswirtschaft* 78(11) (1. Nov. 2008), S. 1183–1196.
- [24] Edsger W. Dijkstra. „The Humble Programmer“. In: *Commun. ACM* 15(10) (Okt. 1972), S. 859–866.
- [25] *DIN EN ISO 9241-11:2018-11, Ergonomie der Mensch-System-Interaktion - Teil 11: Gebrauchstauglichkeit: Begriffe und Konzepte (ISO 9241-11:2018); Deutsche Fassung EN ISO 9241-11:2018*. Beuth Verlag GmbH.
- [26] Paul Dourish. *Where the action is: the foundations of embodied interaction*. 1. MIT Press paperback ed. A Bradford book. OCLC: 254468221. Cambridge, Mass.: MIT Press, 2004. 233 S.
- [27] Morten Esbensen, Paolo Tell, Jacob B. Cholewa, Mathias K. Pedersen und Jakob Bardram. „The dBoard: A Digital Scrum Board for Distributed Software Development“. In: *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces*. ITS '15. New York, NY, USA: ACM, 2015, S. 161–170.
- [28] *f4transkript - Audiotranskription.de*. URL: <https://www.audiotranskription.de/f4> (besucht am 24.06.2021).
- [29] Werner Fuchs-Heinritz. *Biographische Forschung*. Wiesbaden: VS Verlag für Sozialwissenschaften, 2009.
- [30] Göran Goldkuhl. „Separation or unity? Behavioral science vs. design science“. In: *AIS SIGPRAG Pre-ICIS Workshop 2016 "Practice-based Design and Innovation of Digital Artifacts"*. International Conference on information Systems (ICIS). Dublin, 10. Dez. 2016.
- [31] Mike Griffiths. *Pmi-acp exam prep*. 2nd edition. Minnetonka, MN: RMC Publications, 2015.
- [32] *Guidance on usability (ISO 9241-11:1998); German version EN ISO 9241-11:1998*. Standard. German Institute for Standardization, 1998.

- [33] Ulrich Hasenkamp, Hrsg. *Computer supported cooperative work: CSCW ; Informationssysteme für dezentralisierte Unternehmensstrukturen*. 1. Aufl. OCLC: 75389703. Bonn Paris: Addison-Wesley, 1994. 259 S.
- [34] Sigrid Haunberger. „Das standardisierte Interview als soziale Interaktion: Intervieweffekte in der Umfrageforschung“. In: *ZA-Information/Zentralarchiv für Empirische Sozialforschung* (58) (2006), S. 23–46.
- [35] Alan R. Hevner, Salvatore T. March und Sudha Ram. „Design Science in Information Systems Research“. In: *MIS Quarterly* 28(1) (2004), S. 75–105.
- [36] Reinhard Höhn, Stephan Höppner und Andreas Rausch, Hrsg. *Das V-Modell XT: Anwendungen, Werkzeuge, Standards*. eXamen.press. OCLC: 244035190. Berlin: Springer, 2008. 612 S.
- [37] James Hollan, Edwin Hutchins und David Kirsh. „Distributed Cognition: Toward a New Foundation for Human-computer Interaction Research“. In: *ACM Trans. Comput.-Hum. Interact.* 7(2) (Juni 2000), S. 174–196.
- [38] J. Hunt, T. Hume und D. Lozdan. „On Rabbits, Space and Cards: Moving Towards an Informative Workspace“. In: *Agile Conference (AGILE), 2007*. Agile Conference (AGILE), 2007. Aug. 2007, S. 212–217.
- [39] Hiroshi Ishii. „The tangible user interface and its evolution“. In: *Communications of the ACM* 51(6) (1. Juni 2008), S. 32.
- [40] *JIRA Agile - Atlassian*. URL: <https://www.atlassian.com/de/software/jira/agile> (besucht am 24.06.2021).
- [41] Robert Johansen. *Groupware: computer support for business teams*. Series in communication technology and society. New York : London: Free Press ; Collier Macmillan, 1988. 205 S.
- [42] Jim Johnson, Hans Mulder und Standish Group. *The Standish Group Report 1994*. 31. Jan. 1994.
- [43] Capers Jones. *Applied software measurement: assuring productivity and quality*. Software engineering series. OCLC: 23584676. New York: McGraw-Hill, 1991. 493 S.
- [44] Gerd Jüttemann und Hans Thomae, Hrsg. *Biographie und Psychologie*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1987.
- [45] H. Kato und M. Billinghurst. „Marker tracking and HMD calibration for a video-based augmented reality conferencing system“. In: *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*. Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99). Okt. 1999, S. 85–94.
- [46] Christiaan Katsma, Chintan Amrit, Jos van Hillegersberg und Klaas Sikkel. „Can Agile Software Tools Bring the Benefits of a Task Board to Globally Distributed Teams?“ In: *Advances in Global Sourcing. Models, Governance, and Relationships*. Lecture Notes in Business Information Processing 163. Springer Berlin Heidelberg, 1. Jan. 2013, S. 163–179.
- [47] G. Kleining. „Das qualitative Experiment“. In: *Kölner Zeitschrift für Soziologie und Sozialpsychologie* (38) ().
- [48] Martin Kropp, Craig Anslow, Magdalena Mateescu, Roger Burkhard, Dario Vischi und Carmen Zahn. „Enhancing Agile Team Collaboration Through the Use of Large

- Digital Multi-touch Cardwalls“. In: *Agile Processes in Software Engineering and Extreme Programming*. Hrsg. von Hubert Baumeister, Horst Lichter und Matthias Riebisch. Springer International Publishing, 2017, S. 119–134.
- [49] Udo Kuckartz, Hrsg. *Qualitative Evaluation: der Einstieg in die Praxis*. 2., aktualisierte Aufl. Wiesbaden: VS, Verl. für Sozialwiss, 2008. 119 S.
- [50] Richard Lawrence. *How To Split A User Story*. The Humanizing Work Guide to Splitting User Stories. URL: <http://www.richardlawrence.info/splitting-user-stories> (besucht am 24.06.2021).
- [51] Dean Leffingwell. *Agile software requirements: lean requirements practices for teams, programs, and the enterprise*. The Agile software development series. Upper Saddle River, NJ: Addison-Wesley, 2011. 518 S.
- [52] R. Likert. „A technique for the measurement of attitudes.“ In: *Archives of Psychology* 22 140 (1932), S. 55–55.
- [53] Robert C. Martin und Micah Martin. *Agile principles, patterns, and practices in C#*. Robert C. Martin series. Upper Saddle River, NJ: Prentice Hall, 2007. 732 S.
- [54] Philipp Mayring. *Einführung in die qualitative Sozialforschung: eine Anleitung zu qualitativem Denken*. 6., überarbeitete Auflage. Pädagogik. Weinheim Basel: Beltz, 2016. 170 S.
- [55] R. de Melo Oliveira und A. Goldman. „How to Build an Informative Workspace? An Experience Using Data Collection and Feedback“. In: *Agile Conference (AGILE), 2011*. Aug. 2011, S. 143–146.
- [56] Renan de Melo Oliveira, Alfredo Goldman und Claudia O. Melo. „Designing and Managing Agile Informative Workspaces: Discovering and Exploring Patterns“. In: *2013 46th Hawaii International Conference on System Sciences (HICSS)*. Jan. 2013, S. 4790–4799.
- [57] Version One. *Annual State of Agile Survey 2006*. State of Agile Survey. 2016. URL: <http://stateofagile.versionone.com/> (besucht am 24.06.2021).
- [58] Version One. *Annual State of Agile Survey 2020*. State of Agile Survey. 2020. URL: <http://stateofagile.versionone.com/> (besucht am 24.06.2021).
- [59] Taiichi Ōno. *Das Toyota-Produktionssystem*. Frankfurt/Main: Campus-Verl, 2005. 160 S.
- [60] *Overview - Redmine*. URL: <http://www.redmine.org/> (besucht am 24.06.2021).
- [61] J. Paredes, C. Anslow und F. Maurer. „Information Visualization for Agile Software Development“. In: *2014 Second IEEE Working Conference on Software Visualization*. 2014 Second IEEE Working Conference on Software Visualization. Sep. 2014, S. 157–166.
- [62] Ken Peffers, Tuure Tuunanen, Charles E. Gengler, Matti Rossi, Wendy Hui, Ville Virtanen und Johanna Bragge. „The design science research process: a model for producing and presenting information systems research“. In: *Proceedings of the first international conference on design science research in information systems and technology (DESRIST 2006)*. sn, 2006, S. 83–106.
- [63] T. Perry. „Drifting Toward Invisibility: The Transition to the Electronic Task Board“. In: *Agile, 2008. AGILE '08. Conference*. Agile, 2008. AGILE '08. Conference. Aug. 2008, S. 496–500.

- [64] Roman Pichler. *Agile product management with Scrum: creating products that customers love*. The Addison-Wesley signature series. Upper Saddle River, NJ: Addison-Wesley, 2010. 133 S.
- [65] R. Plamondon und S. N. Srihari. „Online and off-line handwriting recognition: a comprehensive survey“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(1) (Jan. 2000), S. 63–84.
- [66] I. Poupyrev, D. S. Tan, M. Billinghamurst, H. Kato, H. Regenbrecht und N. Tetsutani. „Developing a generic augmented-reality interface“. In: *Computer* 35(3) (März 2002), S. 44–50.
- [67] A. Priya, S. Mishra, S. Raj, S. Mandal und S. Datta. „Online and offline character recognition: A survey“. In: *2016 International Conference on Communication and Signal Processing (ICCSP)*. Apr. 2016, S. 0967–0970.
- [68] Florian Raith und Robert Lindermeier. „Media Supported Workspaces in Agile Software Development: Doctoral Symposium Paper“. In: *Computer Software and Applications Conference (COMPSAC), 2014 IEEE 38th Annual*. Computer Software and Applications Conference (COMPSAC), 2014 IEEE 38th Annual. Juli 2014, S. 630–633.
- [69] Florian Raith, Ingo Richter und Robert Lindermeier. „How Project-management-tools are used in Agile Practice: Benefits, Drawbacks and Potentials“. In: *Proceedings of the 21st International Database Engineering & Applications Symposium*. IDEAS 2017. Bristol United Kingdom: Association for Computing Machinery, 12. Juli 2017, S. 30–39.
- [70] Florian Raith, Ingo Richter, Robert Lindermeier und Gudrun Klinker. „Identification of Inaccurate Effort Estimates in Agile Software Development“. In: *Software Engineering Conference (APSEC, 2013 20th Asia-Pacific)*. Dez. 2013, S. 67–72.
- [71] Ingo Richter, Florian Raith und Michael Weber. „Problems in Agile Global Software Engineering Projects Especially Within Traditionally Organised Corporations: [An Exploratory Semi-structured Interview Study]“. In: *Proceedings of the Ninth International C* Conference on Computer Science & Software Engineering*. C3S2E '16. New York, NY, USA: ACM, 2016, S. 33–43.
- [72] Jessica Rubart. „A Cooperative Multitouch Scrum Task Board for Synchronous Face-to-Face Collaboration“. In: *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*. ITS '14. New York, NY, USA: ACM, 2014, S. 387–392.
- [73] Jessica Rubart. „Augmented Notes: Supporting Paper-based Notes on Multitouch Tabletops“. In: *SIGWEB Newsl.* (Summer 2015 Aug. 2015), 2:1–2:4.
- [74] Peter Schmolck. *WebQ Q-Sorting over the Net*. URL: <http://schmolck.userweb.mwn.de/qmethod/webq/> (besucht am 24.06.2021).
- [75] Ken Schwaber und Jeff Sutherland. *Der Scrum Guide*. 2020. URL: www.scrumguides.org (besucht am 24.06.2021).
- [76] Helen Sharp und Hugh Robinson. „Collaboration and co-ordination in mature eXtreme programming teams“. In: *International Journal of Human-Computer Studies* 66(7) (Juli 2008), S. 506–518.

- [77] Helen Sharp, Hugh Robinson und Marian Petre. „The Role of Physical Artefacts in Agile Software Development: Two Complementary Perspectives“. In: *Interact. Comput.* 21(1) (Jan. 2009), S. 108–116.
- [78] Helen Sharp, Hugh Robinson, Judith Segal und Dominic Furniss. „The Role of Story Cards and the Wall in XP Teams: A Distributed Cognition Perspective“. In: *Proceedings of the Conference on AGILE 2006*. Washington, DC, USA: IEEE Computer Society, 2006, S. 65–75.
- [79] W. Stephenson. *The Study of Behavior: Q-technique and Its Methodology*. University of Chicago Press, 1953.
- [80] William Stephenson und Herbert Solomon. „The Study of Behavior: Q-Technique and Its Methodology.“ In: *Journal of the American Statistical Association* 50(272) (Dez. 1955), S. 1415.
- [81] *Tango*. Google Tango Plattform. URL: https://get.google.com/intl/de_ALL/tango/ (besucht am 24.06.2021).
- [82] Stephanie Teasley, Lisa Covi, M. S. Krishnan und Judith S. Olson. „How does radical collocation help a team succeed?“ In: ACM Press, 2000, S. 339–346.
- [83] Stephanie Teufel, Hrsg. *Computerunterstützung für die Gruppenarbeit*. Bonn: Addison-Wesley, 1995. 278 S.
- [84] Rolf Weiber und Daniel Mühlhaus. *Strukturgleichungsmodellierung: eine anwendungsorientierte Einführung in die Kausalanalyse mit Hilfe von AMOS, SmartPLS und SPSS*. 2., erw. und korrigierte Aufl. Springer-Lehrbuch. Berlin Heidelberg: Springer Gabler, 2014. 387 S.
- [85] Ralf Wirdemann und Johannes Mainusch. *Scrum mit User Stories*. 3., erweiterte Auflage. München: Hanser, 2017. 271 S.
- [86] *YouTrack: Issue Tracking and Project Management Tool for Developers*. JetBrains. URL: <https://www.jetbrains.com/youtrack/> (besucht am 24.06.2021).
- [87] X. Zhang, N. Navab und S.- Liou. „E-commerce direct marketing using augmented reality“. In: *ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia*. 2000 IEEE International Conference on Multimedia and Expo. Bd. 1. Juli 2000, 88–91 vol.1.

Abbildungsverzeichnis

2.1	Design Science Forschungsansatz	16
2.2	Klassifizierung der Beobachtungsverfahren	28
2.3	Beispiel einer Ratingskala nach QST	30
2.4	Vergleich formatives und reflektives Messmodell	33
2.5	Interpretationsmöglichkeiten der System Usability Scale	34
3.1	Der Scrum Prozess nach Schwaber et al.	39
3.2	Ebenenmodell zur Granularität von User-Stories	43
3.3	Beispiel zur Darstellung des Product-Backlogs auf einer Stellwand	48
3.4	Beispiel eines einfachen Task-Boards.	49
3.5	Beispiel eines einfachen Task-Boards für die Verwendung in Scrum.	50
3.6	CSCW-Klassifizierung nach Johansen [41].	55
4.1	Beispieldiagramm zum Einsatz digitaler und physikalischer Medien	64
5.1	Grundlegendes Datenmodell eines Task-Boards	71
5.2	Grundlegender Aufbau des Lösungskonzepts	74
5.3	Digitalstift Smart-Pen 2 von Equil mit Empfänger	77
5.4	Vorlage einer Story-Card	79
5.5	Einbindung Smartphone-Anwendung in das Gesamtkonzept	80
5.6	Ansicht einer User-Story in Jira-Agile	82
5.7	Konzeptvorlage zur Benutzerschnittstelle der Smartphone-Anwendung	83
5.8	Beispiele für AR-Marker mit kodierter Identifikationsnummer	85
5.9	Einsatz der AR-Marker am Product-Backlog-Chart	86
5.10	Einsatz der AR-Marker am Task-Board	86
5.11	Vorlage einer Story-Card mit Identifikationsnummer und Marker	87
5.12	Konzeptübersicht zur Erfassung, Interpretation und Synchronisation der Projektdateien auf Product-Backlog und Task-Board	89
5.13	Vereinfachtes Komponentendiagramm der Smartphone-Anwendung.	93
5.14	Screenshots der Smartphone-Anwendung	94
5.15	Vereinfachtes Komponentendiagramm der Komponente "Marker Tracking".	95
5.16	Screenshot der IDE Unity3D mit der Applikation "Marker Tracking".	96
5.17	Vereinfachtes Komponentendiagramm der Komponente "Interpretation und Synchronisation".	97
5.18	JSON-Format zur Definition einer Spalte auf einem Task-Board	98
6.1	Browserbasierte Weboberfläche der Q-Sort Befragung.	110
6.2	Gesamtergebnis der schriftl. Befragung mit klassischer Erhebung.	123

6.3	Einzelergebnisse der schriftl. Befragung mit klassischer Erhebung.	124
6.4	Ergebnisse der schriftl. Befragung mit Hilfe der QST.	125
6.5	Bewertung der Gebrauchstauglichkeit mit Hilfe der SUS	126
A.1	Projekt P1: Ablaufdiagramm zum Medieneinsatz	154
A.2	Projekt P2: Ablaufdiagramm zum Medieneinsatz	155
A.3	Projekt P3: Ablaufdiagramm zum Medieneinsatz	156
A.4	Projekt P4: Ablaufdiagramm zum Medieneinsatz	157
A.5	Projekt P5: Ablaufdiagramm zum Medieneinsatz	158
B.1	Vorlage einer Story-Card vom Typ Epic	159
B.2	Vorlage einer Story-Card vom Typ Story	160
B.3	Vorlage einer Story-Card vom Typ Task	160
B.4	Beispielkonfiguration der "Marker-Tracking Komponente"	162
B.5	JSON-Format zur Übertragung der relevanten Marker-Daten an die Komponente "Interpretation und Synchronisation"	162

Tabellenverzeichnis

3.1	Häufigste Problemursachen von Softwareprojekten im Jahr 1994	36
3.2	Verbreitung agiler Methoden in 2020	38
4.1	Struktur der untersuchten Projekte	61
4.2	Demographische Daten und Projekterfahrung der Studienteilnehmer . . .	61
4.3	Ergebnisse zu F2: Vorteile, Nachteile und Verbesserungspotenziale kon- ventioneller PMS im Kontext verteilter, agiler Projekte	66
6.1	Demographische Daten Team "Virtual Task-Board"	104
6.2	Demographische Daten Team "Jira Agile"	104
6.3	Kriterienauswahl Evaluationsstudie	108
6.4	Bewertungskriterien und Indikatoren der schriftlichen Befragung.	112
6.5	Ergebnisse der qualitativen Beobachtung	114
6.6	Übersicht der VIFs mit der QST	122
6.7	Übersicht der VIFs mit der klassischen Erhebungstechnik	123
6.8	Streuemaße zur schriftl. Befragung mit Hilfe der QST.	125
A.1	Leitfragen der Vorstudie	152

Abkürzungsverzeichnis

API	<i>Application Programming Interface</i>	77
AR	<i>Augmented Reality</i>	84
BS	<i>Behavioral Science</i>	15
CAQDAS	<i>Computer-Assisted Qualitative Data Analysis</i>	23
CLI	<i>Command-Line Interface</i>	97
CSCW	<i>Computer Supported Collaborative Work</i>	10
DiCoT	<i>Distributed Cognition of Teamwork</i>	46
DS	<i>Design Science</i>	15
DSRP	<i>Design Science Research Process</i>	15
GUI	<i>Graphical User Interface</i>	10
GUIs	<i>Graphical User Interfaces</i>	56
HWR	<i>Handwriting Recognition</i>	76
HWR	<i>Handwriting Recognition</i>	76
IW	<i>Informative Workspace</i>	44
KPIs	<i>Key-Perfromance-Indicators</i>	52
LINQ	<i>Language Integrated Query</i>	97
MUI	<i>Multi User Interaction</i>	10
NoA	<i>Number of Appearance</i>	65
NoI	<i>Number of Interviewees</i>	65
OWP	<i>Open Workspace Principle</i>	43
PMS	<i>Projektmanagement Software</i>	3
QST	<i>Q-Sort-Technik</i>	29
REST	<i>Representational State Transfer</i>	97
SCM	<i>Source Code Management</i>	118
SQL	<i>Structured Query Language</i>	97
SUS	<i>System Usability Scale</i>	15
TUI	<i>Tangible-User-Interface</i>	55
TUIs	<i>Tangible-User-Interfaces</i>	10
VIF	<i>Variance Inflation Factor</i>	32

Anhang A

Vorstudie

A.1 Ausschnitt eines Transkripts

Nachfolgend befindet sich ein kurzer Ausschnitt eines Transkripts der Interviewstudie (vgl. Kap. 4). Der Ausschnitt dient lediglich zur Veranschaulichung der Transkriptionstechnik und wurde willkürlich ausgewählt.

I: *Interessant. Gut meine nächste Frage ist eigentlich schon beantwortet. Das wäre jetzt gewesen, welche Rollen Sie in dieser Zeit besetzt haben. Aber ja... 00:04:03-8*

B: *Eigentlich alle Rollen. Von Entwickler, Scrum-Master auch Product-Owner schon gemacht, seit (...) ja wann habe ich angefangen (...) seit 2012 / 2013 bin ich sehr stark mit agile Coaching beschäftigt. 00:04:19-4*

I: *Die nächste Frage wäre jetzt spezifisch auf die Projekte, die sie aktuell betreuen bzw. begleiten. Da wäre jetzt interessant, um welche Art von Produkt/Dienstleistung handelt usw. handelt es sich und wie viele Mitarbeiter... 00:04:34-3*

B: *Also im Moment ist es ein Direktversicherer und es sind ungefähr 200 Mitarbeiter. 00:04:40-3*

I: *Ungefähr 200 Mitarbeiter. Und welche Vorgehensweise (...) es ist Softwareentwicklung oder? (B: ja Softwareentwicklung, genau). Sie arbeiten nach Scrum oder Kanban? 00:04:51-1*

A.2 Leitfragen zur Vorstudie

Tabelle A.1: Leitfragen der Vorstudie

1. General questions about the interviewee:
 - a) How long are you already working in software development projects? How long thereof are you using agile methodologies?
 - b) What roles (within the development process) did you take in the past and what are your current roles and responsibilities?
 - c) Please describe the software projects you are working on at the moment. Take into account: kind of software, number of teams and team-members, regional allocation, duration of project
2. Questions about usage of artifacts and media in the respective phases or meetings of development process:
 - a) What kind of digital media or tools do you use to support the development process?
 - b) Please describe the development process in detail? Take into account every meeting or phase with focus on participating roles, used artifacts, digital or analog media and work environment.
3. Detailed questions about tool and media usage:
 - a) What kind of media do you mainly use beyond the meetings? (e.g. knowledge-base, transfer of artifacts, communication, etc.)
 - b) How do you organize short-term needs for coordination?
 - c) What happens to the collected data (e.g. requirements, key-figures regarding the process) after a iteration? Are they archived and reused in a certain manner?
 - d) Is the collected data (e.g. in a project-management-tool) cross referenced to other tools or artifacts? (e.g. revision control, technical documents, etc.)
 - e) Did you notice any further considerable tools or means used by the team members facilitating their daily work?
 - f) Can you give me an overall description of your team's workspace taking into account room arrangement and available media?
 - g) What should an ideal agile workspace look like in your opinion? What aspects or elements are important?

A.3 Ablaufdiagramme aller in der Vorstudie untersuchten Projekte

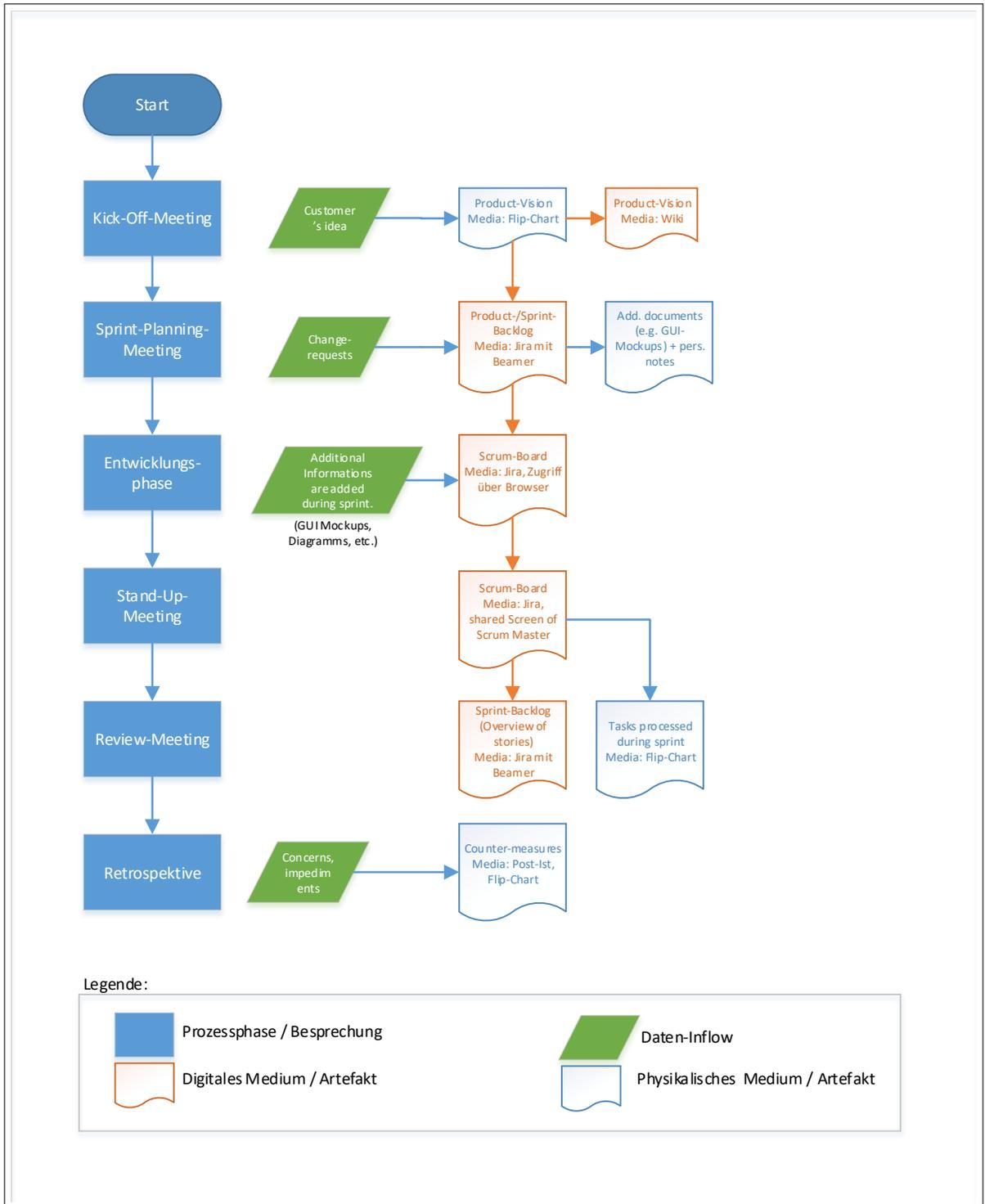


Abbildung A.1: Projekt P1: Ablaufdiagramm zum Medieneinsatz

A.3 Ablaufdiagramme aller in der Vorstudie untersuchten Projekte

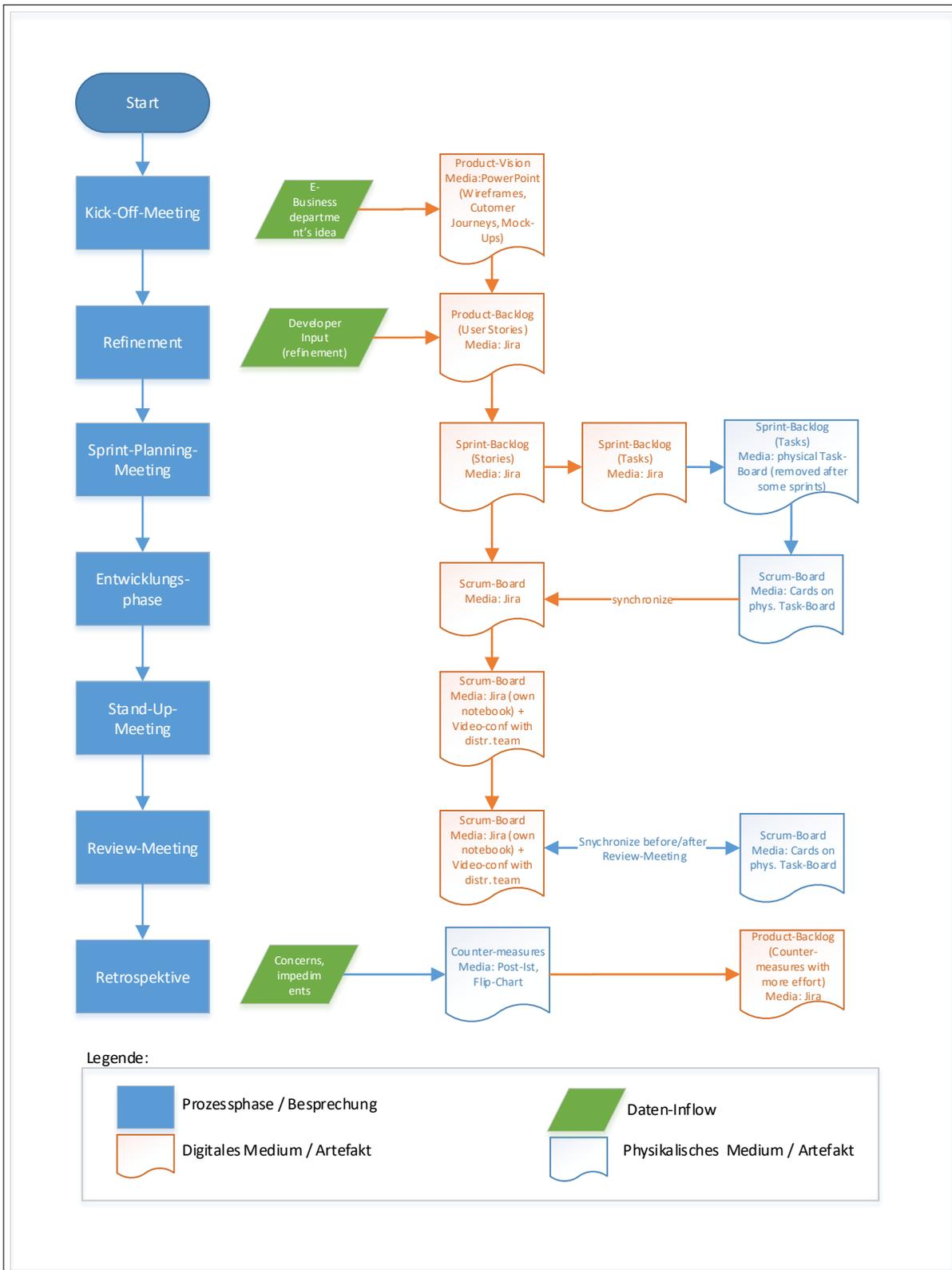


Abbildung A.2: Projekt P2: Ablaufdiagramm zum Medieneinsatz

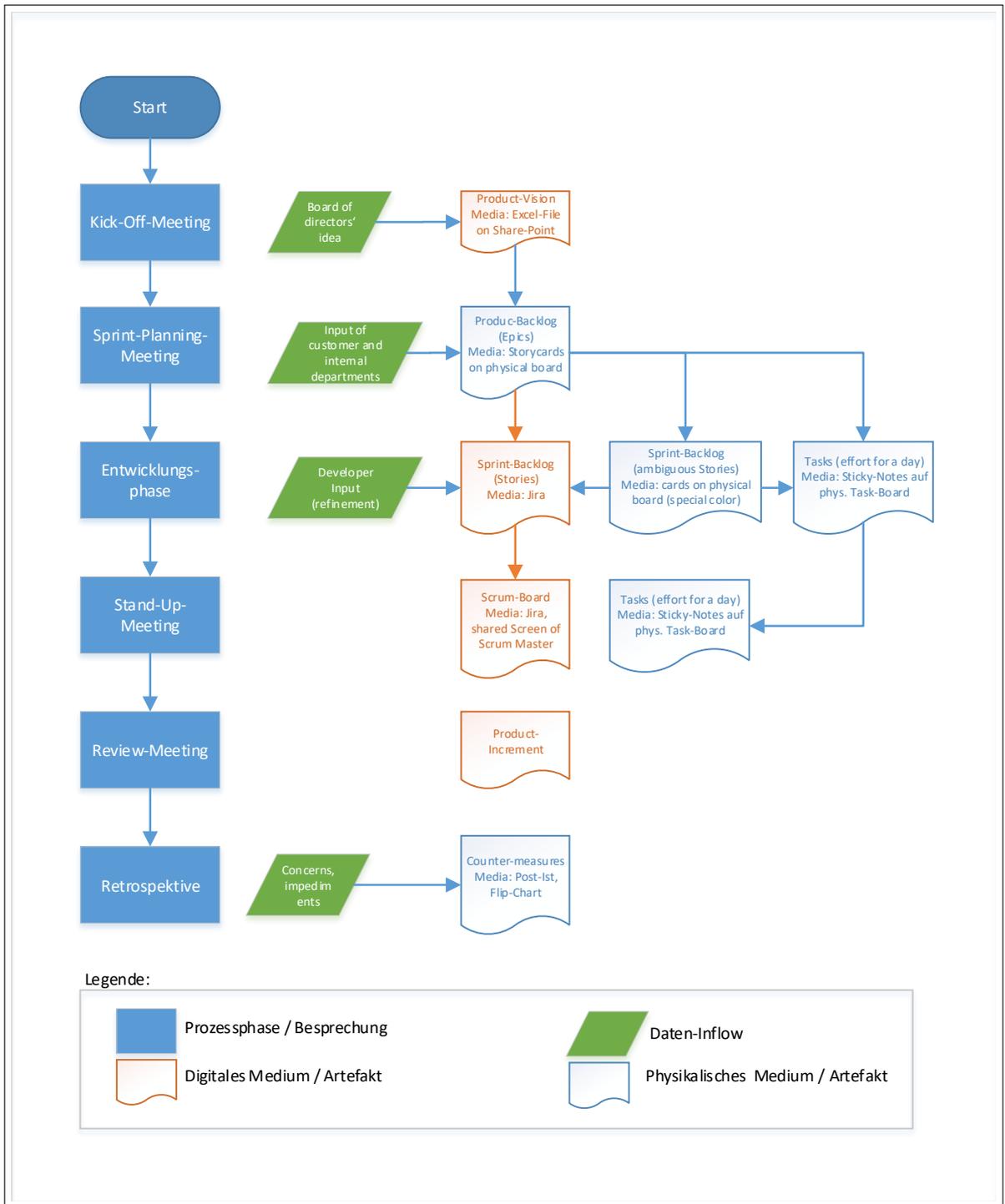


Abbildung A.3: Projekt P3: Ablaufdiagramm zum Medieneinsatz

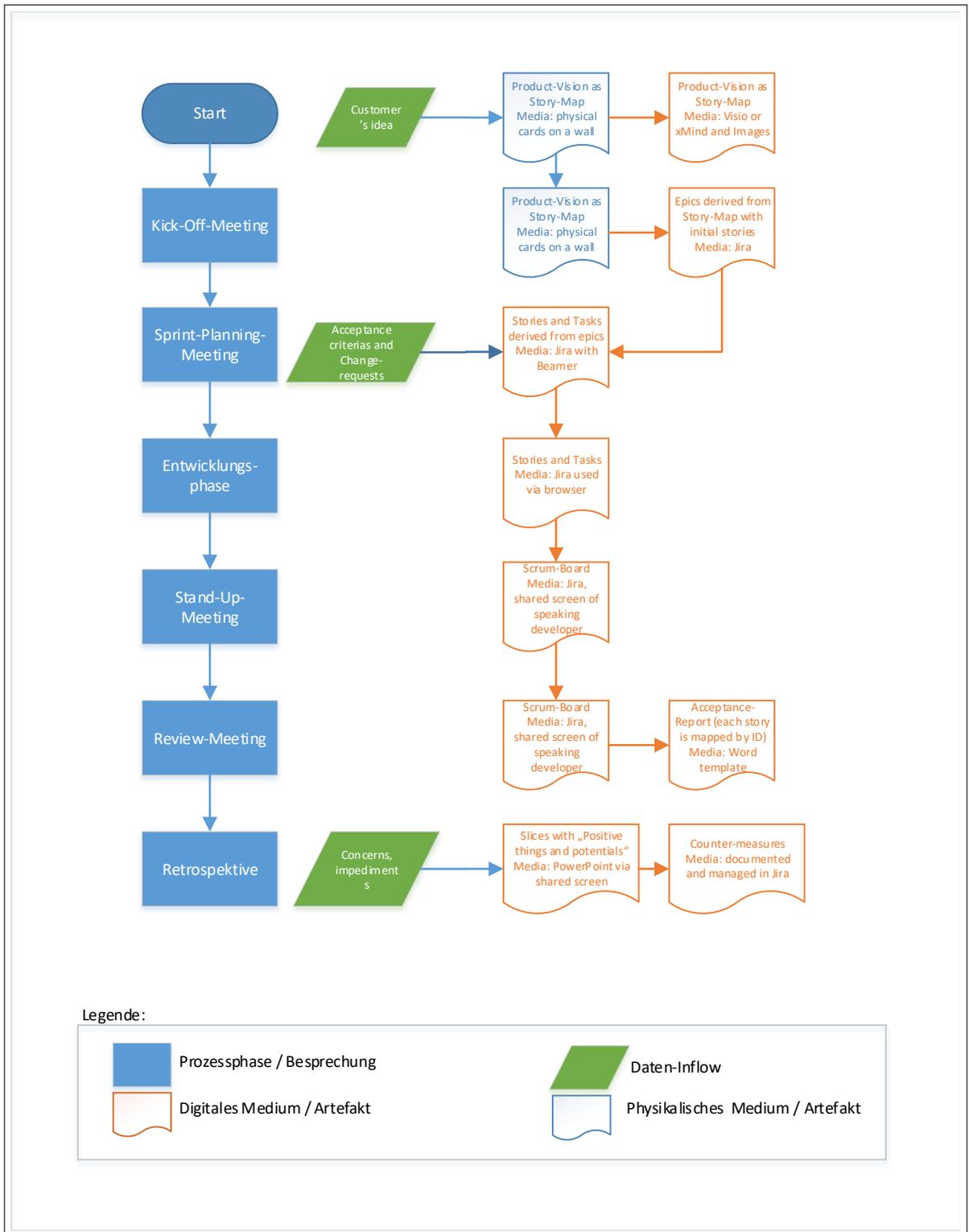


Abbildung A.5: Projekt P5: Ablaufdiagramm zum Medieneinsatz

Anhang B

Konzeption

B.1 Druckvorlagen Story-Cards

B.1.1 Druckvorlage Epic

The image shows a template for an Epic story card. It consists of a large rectangular frame. On the left side, there is a vertical rectangle containing a large, bold letter 'E'. The main area of the card is divided into two horizontal sections. The top section is labeled 'Epic:' and contains a large, solid black square with a white, pixelated icon of a person's head and shoulders. The bottom section is labeled 'Acceptance-Criteria:' and is mostly empty. In the bottom right corner of this section, there is a small rectangular box containing the number '1'.

Abbildung B.1: Vorlage einer Story-Card vom Typ Epic

B.1.2 Druckvorlage Story

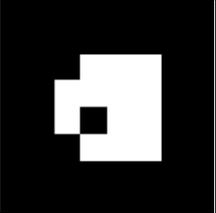
S <div style="border: 1px solid black; height: 100px; width: 60px;"></div>	Story:	
	Acceptance-Criteria:	

Abbildung B.2: Vorlage einer Story-Card vom Typ Story

B.1.3 Druckvorlage Task

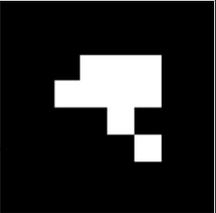
<div style="border: 1px solid black; height: 100px; width: 60px;"></div>	Task Name:	
	Description:	

Abbildung B.3: Vorlage einer Story-Card vom Typ Task

B.2 Konfigurationsparameter der Softwarekomponente "Marker-Tracking"

B.2.1 Erläuterung der Konfigurationsparameter

Konfigurationsparameter	Beschreibung
clientTag	Der hier konfigurierte Wert ist notwendig, um mehrere Instanzen der Anwendung "Marker Tracking" unterschieden zu können. Entsprechend muss sich dieser Wert bei allen eingesetzten Instanzen unterscheiden.
firstGridMarker	Legt den Marker mit der niedrigsten Identifikationsnummer fest, der zur Definition von Spalten und Zeilen auf dem Task-Board eingesetzt wurde.
lastGridMarker	Legt den Marker mit der höchsten Identifikationsnummer fest, der zur Definition von Spalten und Zeilen auf dem Task-Board eingesetzt wurde.
firstCardMarker	Legt den Marker mit der niedrigsten Identifikationsnummer fest, der auf den Druckvorlagen verwendet wird.
lastCardMarker	Legt den Marker mit der höchsten Identifikationsnummer fest, der auf den Druckvorlagen verwendet wird.
ip	IP-Adresse für die Verbindung zur Applikation "Dateninterpretation und Synchronisation"
port	Port für die Verbindung zur Applikation "Dateninterpretation und Synchronisation"
syncRate	Intervall zum Senden der Daten an die Applikation "Dateninterpretation und Synchronisation"
widthOfGridMarker	Breite der Marker, die zur Definition von Spalten und Zeilen auf dem Task-Board verwendet wurden in Meter.
widthOfCardMarker	Breite der Marker auf den Druckvorlagen in Meter.
recognitionRate	Intervall, das definiert nach welcher Zeit die Anwendung nach neuen Bildern sucht, um diese entsprechend auszuwerten (in Sekunden).

B.2.2 Beispielkonfiguration der Komponente Marker-Tracking

```
{
  "clientTag": "taskboard",
  "firstGridMarker": 2964,
  "lastGridMarker": 3000,
  "firstCardMarker": 1,
  "lastCardMarker": 199,
  "ip": "127.0.0.1",
  "port": 12345,
  "syncRate": 30,
  "widthOfGridMarker": 0.035,
  "widthOfCardMarker": 0.035,
  "threshold": 82,
  "recognitionRate": 10
}
```

Abbildung B.4: Beispielkonfiguration der "Marker-Tracking Komponente"

B.2.3 Übertragungsparameter im JSON-Format zur Komponente Interpretation und Synchronisation

```
{
  "Markers": [{
    "ClientTag": "taskboard",
    "Id": 57,
    "PosX": 539.8145751953125,
    "PosY": 228.94496154785157,
    "Visible": true
  }, {
    "ClientTag": "taskboard",
    "Id": 60,
    "PosX": 327.7639465332031,
    "PosY": 228.1000518798828,
    "Visible": true
  }
  ]
}
```

Abbildung B.5: JSON-Format zur Übertragung der relevanten Marker-Daten an die Komponente "Interpretation und Synchronisation"

Anhang C

Beobachtungsstudie

C.1 Vorlage Beobachtungsprotokoll

Beobachtungsprotokoll: Studie „Virtual Task-Board“

Team: <TEAMNAME>

Prozessphase: <NAME DER PROZESSPHASE>

Starzeitpunkt: <UHRZEIT>

Endzeitpunkt: <UHRZEIT>

Ausgangssituation: Welche Ergebnisse, Artefakte sind für die aktuelle Prozessphase relevant und liegen dem Team zu Beginn der Phase vor? Welche Medien werden dafür benutzt?

Ablaufbeschreibung: Wie gehen die Teilnehmer in der aktuellen Prozessphase vor (Prozessschritte, Mediennutzung, Rollen)?

Auffälligkeiten: Gibt es Verhaltensweisen oder Auffälligkeiten, die direkten oder indirekten Bezug auf die Forschungsfragen haben könnten? Orientierung am Indexblatt!

Angaben: (Relevante Indizes + Zeitpunkt + Beschreibung inkl. beteiligter Rollen + Interpretation)

(<KRITERIEN> | <UHRZEIT>)

Beschreibung:

Interpretation:

C.2 Indexblatt

(Ac) Accessibility:

- Wie gut kann der Entwicklungsprozess abgebildet bzw. dargestellt werden?
- Werden Prozessbeteiligte auf Änderungen zeitnah aufmerksam?
- Können Informationen auch von externen Besuchern schnell erfasst werden?

- Teilt sich die Arbeit mit dem Tool gleichmäßig auf alle Prozessbeteiligten auf?
- Sind die Informationen schnell erfassbar und übersichtlich dargestellt? Werden die Informationen von den Studienteilnehmern verstanden?

(F) Flexibility:

- Wurde genau nach Prozessvorgaben gearbeitet oder wurden eigene Anpassungen vorgenommen, welche sich auch bei der Toolnutzung widerspiegelten?
- Wurden Informationen im Tool festgehalten, die vom vorgegebenen Standardumfang abweichen?
- Welche durch die Toolnutzung gegebenen Einschränkungen der Projektarbeit lassen sich beobachten? Gibt es Handlungsweisen, die auf Einschränkungen in der Toolnutzung zurückzuführen sind?

(M) Motivation:

- Geben Kollegen unmittelbares Feedback nach dem Erledigen einer Aufgabe? (Kritik/Lob o. Ä.)
- Beurteilung der Motivation der ganzen Gruppe aus Sicht des Beobachters (evtl. Motivation in bestimmten Situationen/Prozessabschnitten beurteilen)
- Lassen sich bei Toolnutzung (z.B. Durchführung von Statusupdates) bestimmte Verhaltensweisen beobachten? (z.B. Aufgabe wird als lästig/störend empfunden v.s. Aufgabe wird gerne durchgeführt)

(H) Haptic Quality:

- In welchem Maß arbeiten die Versuchsteilnehmer mit dem digitalen Board? Wird zusätzlich auf physikalische Medien zurückgegriffen?
- Wie werden die digitalen Story-Cards eingesetzt? Wird aktiv damit gearbeitet oder werden sie nur mitgezogen bzw. die Arbeit damit sogar als störend empfunden?

(I) Integration:

- Werden die vorhandenen Möglichkeiten von Jira zur Integration von anderen Systemen genutzt?
- Welche konkreten Beispiele für die Integration von anderen Systemen lassen sich beobachten?
- Gibt es Schwierigkeiten bei der Nutzung der genannten Möglichkeiten?

(Av) Availability:

- Technische Schwierigkeiten sollen hier notiert und dokumentiert werden?
- Ausfallzeit, kurze Beschreibung der Situation, Konsequenzen für die Projektarbeit
- Wie gehen die Teilnehmer damit um?

(O) Overhead:

- Gibt es Aspekte, die von den Teilnehmern häufig falsch gemacht werden? (Fehlbedienung)
- Können alle Teilnehmer gleichermaßen mit den betrachteten Tools umgehen?
- Gibt es häufige Nachfragen zu bestimmten Aspekten während der Projektphase?

(C) Communication:

- Finden angeregte Diskussionen statt an denen das ganze Team beteiligt ist?
- Gibt es Personen, die größtenteils mit der Toolnutzung beschäftigt sind?
- Gibt es längere Phasen innerhalb der Meetings, die nur der Toolnutzung gewidmet sind?
- Entstehen Gespräche nachdem Statusupdates am Task-Board durchgeführt wurden?
- Finden Diskussionen am Task-Board statt? Wird es als zentrale Treffpunkt genutzt?

C.3 Fragenkatalog der System Usability Scale

1. Ich denke, dass ich das System gerne häufig benutzen würde.
2. Ich fand das System unnötig komplex.
3. Ich fand das System einfach zu benutzen.
4. Ich glaube, ich würde die Hilfe einer technisch versierten Person benötigen, um das System benutzen zu können.
5. Ich fand, die verschiedenen Funktionen in diesem System waren gut integriert.
6. Ich denke, das System enthielt zu viele Inkonsistenzen.
7. Ich kann mir vorstellen, dass die meisten Menschen den Umgang mit diesem System sehr schnell lernen.
8. Ich fand das System sehr umständlich zu nutzen.
9. Ich fühlte mich bei der Benutzung des Systems sehr sicher.
10. Ich musste eine Menge lernen, bevor ich anfangen konnte das System zu verwenden.