



Schlussbericht des Vorhabens

Produktqualität für Agile Softwareentwicklung

Autoren:

Hartmut Schmitt, HK Business Solutions

Dominik Rost, Fraunhofer IESE

Philipp Bender, CAS Software

Richard Wacker, YellowMap





Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS13032 gefördert.



Projekträger:
Deutsches Zentrum für Luft- und Raumfahrt

Schlussbericht des Vorhabens
PQ4Agile – Produktqualität für Agile Softwareentwicklung

Web: www.pq4agile.de

Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

Ansprechpartner:
Hartmut Schmitt
HK Business Solutions GmbH
Mellinweg 20
66280 Sulzbach
Tel.: +49 (0)6897 99904-24
E-Mail: hartmut.schmitt@hk-bs.de



Abstract

Dieses Dokument ist der Schlussbericht gem. NKBF 98 des Projekts „PQ4Agile — Produktqualität für Agile Softwareentwicklung“, BMBF-Förderkennzeichen 01IS13032. Das PQ4Agile-Konsortium bestand aus den Partnern HK Business Solutions GmbH (Konsortialführer, 01IS13032A), CAS Software AG (Projektpartner, 01IS13032B), Fraunhofer-Institut für Experimentelles Software Engineering IESE (Projektpartner, 01IS13032C) und YellowMap AG (Projektpartner, 01IS13032D). Der Bericht stellt unter anderem die Ziele, den Ablauf, die Ergebnisse und die zukünftige Verwertung der Resultate des Projekts vor.

Schlagworte

Agile Entwicklung, Agilität, Scrum, Software Engineering, Softwareentwicklung, Requirements Engineering, Softwarearchitektur, Usability, User Experience, UX, Anwendungssoftware

Aus Gründen der besseren Lesbarkeit wird auf die gleichzeitige Verwendung männlicher und weiblicher Sprachformen verzichtet. Sämtliche Personenbezeichnungen gelten gleichwohl für beiderlei Geschlechter.

Inhalt

1	Aufgabenstellung	5
1.1	Motivation	5
1.2	Wissenschaftliche und technische Arbeitsziele des Vorhabens	6
1.3	Bezug des Vorhabens zu den förderpolitischen Zielen	7
2	Voraussetzungen	8
2.1	Notwendigkeit der Zuwendung	8
2.2	Konsortialpartner und bisherige Arbeiten	8
2.2.1	HK Business Solutions GmbH	8
2.2.2	CAS Software AG	8
2.2.3	Fraunhofer-Institut für Experimentelles Software Engineering IESE	9
2.2.4	YellowMap AG	9
3	Planung und Ablauf des Vorhabens	10
3.1	Lösungsidee	10
3.2	Projektplan	11
3.2.1	Laufzeit, Arbeitspakete und Meilensteinplanung	11
3.2.2	Risikomanagement und Abbruchkriterien	12
3.3	Ablauf	12
4	Wissenschaftlicher und technischer Stand	14
4.1	Agile Entwicklungsmethoden	14
4.1.1	Qualitätserreichung in agilen Entwicklungsmethoden	14
4.1.2	Scaled Agile Framework	14
4.1.3	Agile Methoden und Software-Engineering	15
4.2	Reifegradmodelle	15
4.2.1	Capability Maturity Model Integration	15
4.2.2	Agile Maturity Model Integration	16
4.3	Verwandte Projekte	16
4.4	Verwendete Fachliteratur	16
4.5	Benutzte Informations- und Dokumentationsdienste	25
5	Verwendung der Zuwendung und Projektergebnisse	26
5.1	Arbeitspaket 1: Grundlagen	26
5.1.1	Ziele des Arbeitspakets	26
5.1.2	Verwendung der Zuwendung	26
5.1.3	Ergebnisse	28
5.2	Arbeitspaket 2: Softwareengineering Best Practices	30
5.2.1	Ziele des Arbeitspakets	30
5.2.2	Verwendung der Zuwendung	31
5.2.3	Ergebnisse	32
5.3	Arbeitspaket 3: Integration Best Practices und Entwicklungsprozess	34
5.3.1	Ziele des Arbeitspakets	34
5.3.2	Verwendung der Zuwendung	34
5.3.3	Ergebnisse	35
5.4	Arbeitspaket 4: Evaluation	36
5.4.1	Ziele des Arbeitspakets	36
5.4.2	Verwendung der Zuwendung	36
5.4.3	Ergebnisse	37

5.5	Arbeitspaket 5: Methodenkompetenz	37
5.5.1	Ziele des Arbeitspakets	38
5.5.2	Verwendung der Zuwendung	38
5.5.3	Ergebnisse	39
5.6	Arbeitspaket 6: Projektkoordination	40
5.6.1	Ziele des Arbeitspakets	40
5.6.2	Verwendung der Zuwendung	40
6	Wichtigste Positionen des zahlenmäßigen Nachweises	41
6.1	HK Business Solutions	41
6.2	CAS Software	41
6.3	Fraunhofer IESE	41
6.4	YellowMap	41
7	Notwendigkeit und Angemessenheit der geleisteten Arbeit	42
8	Voraussichtlicher Nutzen und Verwertbarkeit der Ergebnisse	43
8.1	HK Business Solutions	43
8.2	CAS Software	43
8.3	Fraunhofer IESE	44
8.4	YellowMap	45
8.5	Methodentransfer	45
9	Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen	47
10	Erfolge und geplante Veröffentlichungen	48
11	Anhang: Projektposter	49
12	Abbildungsverzeichnis	50
13	Dokumentinformation	51

1 Aufgabenstellung

Insbesondere bei kleinen und mittleren Unternehmen (KMU) der Softwarebranche haben agile Entwicklungsmethoden eine weite Verbreitung erreicht. Die Betrachtung von Qualität ist bei diesen Methoden oft auf funktionale Korrektheit fokussiert, während Hilfestellungen zur zielgerichteten Erreichung von Produktqualitäten fehlen. Dadurch leidet häufig bereits in frühen Entwicklungsphasen die Qualität der Softwareprodukte und es sind aufwändige Nacharbeiten notwendig. Dies wiederum hat unmittelbar negativen Einfluss auf die Absatzchancen der Produkte und auf die Marktposition der Unternehmen.

Daher bestand die Aufgabenstellung des Forschungsvorhabens PQ4Agile darin, agile Entwicklungsprozesse so durch etablierte Aktivitäten des Softwareengineerings anzureichern, dass hierdurch eine ganzheitliche Qualitätsbetrachtung und eine vorhersagbar hohe Produktqualität in agilen Softwareentwicklungsprojekten möglich werden. Vorgesehen war, Softwareengineering-Methoden und zusammenhängende Aktivitäten aus den Bereichen Requirements Engineering, Softwarearchitektur und User Experience Engineering in Einzelteile zu zerlegen, die separat ausführbar sind, und diese Aktivitäten anschließend an die Erfordernisse agiler Entwicklungsvorgehen anzupassen, so dass die gewonnenen Best Practices Softwareentwicklern für einen möglichst einfachen Einsatz in agilen Projekten zur Verfügung stehen.

Durch diese Vorgehensweise sollte insbesondere in softwareentwickelnden KMU die Entstehung eines eigenständigen und ohne ausführliche Prozessbeschreibung gelebten, agilen Entwicklungsprozesses gefördert werden, der eine strukturierte Qualitätsbetrachtung ermöglicht. Mit PQ4Agile sollte also kein neues Vorgehensmodell geschaffen werden, sondern neue, innovative Wege des Übergangs aufgezeigt werden, die von Softwareunternehmen beschritten werden können, gleich ob sie bisher einen agilen oder einen eher qualitätsgetriebenen Entwicklungsansatz verfolgten.

Softwareentwickelnde Unternehmen sollten die angestrebten Projektergebnisse direkt in ihren Entwicklungsprozess einfließen lassen können, damit sie schneller und effizienter als bisher und unter Berücksichtigung kundenspezifischer Anforderungen hochqualitative Lösungen entwickeln können. Der Wissenschaft und Industrie sollten die Projektergebnisse in Form von Publikationen und Vorträgen vorgestellt werden. Über ein Projektportal und Social-Media-Kanäle sollte zudem die interessierte Öffentlichkeit regelmäßig über das Projekt und seine aktuellen Ergebnisse informiert werden.

1.1 Motivation

Agile Entwicklungsmethoden (vgl. 4.1) versprechen eine Reihe bestechender Vorteile, wie schnelles und stetiges Ausliefern nutzbarer (Teil-)Funktionalitäten durch kurze Entwicklungszyklen, Kundenzufriedenheit durch enge Interaktion der Stakeholder und kontinuierliche Anpassung der Anforderungen oder Zufriedenheit der Entwickler durch eigenverantwortliches und selbstbestimmtes Handeln. Agile Entwicklungsvorgehen haben aus diesen Gründen schon eine weite Verbreitung erreicht, insbesondere bei KMU. Agile Entwicklungsmethoden bieten primär Unterstützung für Projektmanagement (beispielsweise durch Sprints, Backlog, etc.) oder für die Implementierungstätigkeit selbst (beispielsweise durch Pair Programming, Test-Driven-Development, etc.). Die Betrachtung von Qualität ist jedoch meist auf funktionale Korrektheit fokussiert. Hilfestellung zur zielgerichteten Erreichung von nichtfunktionalen Produktqualitäten wie Performanz, Skalierbarkeit, Benutzbarkeit oder Wartbarkeit fehlt oft.

Diese mangelnde Unterstützung für Produktqualitäten führt häufig dazu, dass erst mit Abschluss eines Inkrements oder noch später festgestellt wird, dass Antwortzeiten nicht erreicht werden, Änderungen schwieriger sind als gedacht oder die Benutzbarkeit unzumutbar ist. Die Antwort agiler Entwicklungsvorgehen, um ein System von solchen Problemen zu befreien, sind Refactorings. Die Praxis zeigt jedoch, dass solche Nacharbeiten, speziell für nichtfunktionale Eigenschaften, oft große Teile des Systems betreffen und mit äußerst hohen Aufwänden und damit Kosten verbunden sind. Außerdem müssen solche Arbeiten oft „versteckt“ werden, da sie keinen funktionalen Mehrwert für den Kunden liefern. Deshalb werden in der Praxis weitreichende Refactorings oft verschoben und schließlich ausgelassen. Mangelnde Qualität schon in frühen Phasen der Entwicklung ist die Folge. Damit steigen später die Wartungskosten, oftmals so stark, dass eine Neuentwicklung notwendig wird. Die mangelnde Betrachtung von Produktqualitäten bedeutet also erhöhte Kosten, Aufwände und Wahrscheinlichkeit für das Scheitern von Entwicklungsprojekten. Sie hat damit einen direkten Einfluss auf Kundenbeziehungen, Absatzchancen und Marktposition des KMU. Insbesondere im zunehmend internationalisierten Wettbewerb ist für deutsche Firmen jedoch Qualität ein wichtiges Alleinstellungsmerkmal gegenüber der Produktion in Ländern mit niedrigerem Lohnniveau (Offshore-Entwicklung).

Trotzdem gibt es viele erfolgreiche agile Entwicklungsprojekte: Mangelnde Unterstützung der Methode wird dabei durch die Erfahrung der Entwickler kompensiert. Es gibt einige typische Projekt- und Sys-

temeigenschaften, die den Einsatz agiler Methoden unterstützen: die Ähnlichkeit des zu entwickelnden Systems gegenüber früheren Systemen, der Einsatz weniger und kleiner Teams, eine begrenzte Komplexität der Systemumgebung, eine geringe Systemgröße oder keine Notwendigkeit für eine langlaufende Systemwartung. In solchen Fällen können Produktqualitäten vornehmlich durch die Fähigkeiten und Erfahrung der Entwickler erreicht werden. Agile Methoden werden aber auch für viele andere Projekte eingesetzt, die nicht die zuvor genannten Charakteristika aufweisen. Da in solchen Projekten beobachtbar die Qualität leidet, sollte in dem PQ4Agile-Projekt eine systematische Unterstützung entwickelt werden, um den breiteren erfolgreichen Einsatz agiler Methoden zu ermöglichen.

Die angestrebte Integration von Softwareengineering-Aktivitäten und agilen Entwicklungsprozessen birgt augenscheinlich einen immanenten Widerspruch. Eines der grundlegenden Ziele von agilen Entwicklungsprozessen ist die hohe Geschwindigkeit bei der Auslieferung von Arbeitsergebnissen und die damit verbundene schnelle Reaktion auf Änderungen im Entwicklungsprojekt. Es stellte sich also die Frage, ob dieser Vorteil durch die zusätzlichen Planungsaufwände durch Softwareengineering-Aktivitäten nicht ausgehöhlt wird. Jedoch erwartete sich das Konsortium mit Hilfe des beschriebenen Ansatzes das Gegenteil: Auch in agilen Softwareentwicklungsprozessen ist es das Ziel, ein hochqualitatives Produkt zu erstellen, das heißt, ein Produkt, das alle geforderten Qualitätseigenschaften erfüllt. Solche Qualitäten können nur erreicht werden, indem Designentscheidungen getroffen werden, die zu einer adäquaten Systemarchitektur führen. Ohne entsprechende Planung, wie es bei rein agilen Vorgehen oft der Fall ist, besteht nur eine geringe Wahrscheinlichkeit, das Produkt so zu entwickeln, dass die Qualitätsziele direkt erreicht werden. Das heißt, in späteren Sprints muss häufig Aufwand und Zeit in eine entsprechende Überarbeitung und ein Refactoring des Codes investiert werden.

Bei einer Integration von Softwareengineering-Aktivitäten und einer damit einhergehenden Planung für Qualitätseigenschaften erhöht sich jedoch die Wahrscheinlichkeit zur Erreichung der Qualitätsziele bei gleichzeitiger Reduktion der Wahrscheinlichkeit für aufwändige Nacharbeiten. Werden die Softwareengineering-Aktivitäten im Vergleich zu konventionellen Vorgehen noch leichtgewichtig gehalten – so die Hypothese des Konsortiums –, lässt sich auf die Projektlaufzeit gesehen eine Reduktion des Gesamtaufwandes erzielen. Hierfür müsste es gelingen, die Softwareengineering-Best-Practices als abgeschlossene, integrierbare Bausteine zu entwickeln, damit eine stetige Lieferung von Projektinkrementen beibehalten werden kann.

1.2 Wissenschaftliche und technische Arbeitsziele des Vorhabens

Ausgehend von der oben beschriebenen Aufgabenstellung wurden folgende Arbeitsziele definiert, die von dem Konsortium – bestehend aus drei Softwareherstellern und einem Forschungspartner – im Rahmen des Forschungsvorhabens erreicht werden sollten.

- Beschreibung der Softwareengineering-Best-Practices: Es sollten Best-Practice-Bausteine der Softwareengineering-Disziplinen Requirements Engineering, Softwarearchitektur und User Experience Engineering entwickelt werden, die den agilen Prinzipien entsprechen und in agilen Entwicklungsvorgehen anwendbar sind. Für die einzelnen Best-Practices sollten Aktivitätsbeschreibungen angefertigt werden, die die Anpassungen der SE-Aktivitäten beschreiben und die die Durchführung unterstützen.
- Integrationsbeschreibung: Die Integrationsbeschreibung sollte auf Basis der Best-Practice-Beschreibungen die Integration der einzelnen, angepassten Aktivitäten untereinander erläutern sowie die Integration in bestehende Entwicklungsprozesse beschreiben. Dabei sollte insbesondere auf Abhängigkeiten zwischen Rollen, Aktivitäten und Artefakten eingegangen werden.
- Erhebung empirische Daten: Die entwickelten Best-Practices und deren Anwendung im Entwicklungsprozess sollten im Industriekontext evaluiert werden, um eine empirisch fundierte Aussage über die Effekte auf die Qualitäten der resultierenden Produkte treffen zu können. Außerdem sollte das Verhältnis von Kosten und Nutzen in Pilotprojekten der beteiligten KMU quantifizierbar gemacht werden und es sollten Verbesserungsmöglichkeiten identifiziert werden.
- Erstellung eines Schulungskonzepts: Im Rahmen des PQ4Agile-Vorhabens sollte ein Schulungskonzept für die Durchführung von agilen Entwicklungsprojekten nach der PQ4Agile-Methode erarbeitet werden. Dieses sollte als Grundlage für Seminare und Vorlesungen sowie für Beratungsprojekte und als Referenz in der Softwareentwicklung dienen.
- Webseite und Social Media: Es sollte eine Website aufgebaut werden, die als Informationsplattform für die interessierte Fachöffentlichkeit dienen kann. Auf der Website sollten kontinuierlich allgemeine Informationen zum Forschungsprojekt und insbesondere sämtliche Projektergebnisse veröffentlicht werden, damit diese allen interessierten KMU zum Einsatz zur Ver-

fügung stehen. Um den Bekanntheitsgrad des Projektes zu erhöhen, sollten entsprechende Informationen über Social-Media-Kanäle verbreitet werden.

- **Werkzeugunterstützung:** Als Werkzeugunterstützung für die einzelnen Softwareengineering-Aktivitäten sollten Anpassungen erstellt werden, beispielsweise in Form von Plugins für existierende Applikationen. Zur Unterstützung bei der Ausführung der SE-Aktivitäten und der Zusammenarbeit im Entwicklungsprozess war beispielsweise eine Anpassung der CAS On-Premise Lösung CAS genesisWorld geplant.

Querschnittlich dazu sollten im Forschungsvorhaben PQ4Agile Konzepte zur konkreten Ausgestaltung bei der Ausführung der Methoden-Bausteine erarbeitet werden, die die beteiligten Personen, die Durchführungsmodi sowie die verwendeten Artefakte und Materialien beinhalten. Darunter fallen beispielsweise die Durchführung von Workshops, Gruppendiskussionen am Whiteboard oder die Verwendung von computergestützten (u. U. angepassten) SE-Werkzeugen.

1.3 Bezug des Vorhabens zu den förderpolitischen Zielen

Das Bundesministerium für Bildung und Forschung (BMBF) verfolgt mit der Fördermaßnahme „KMU-Innovationsoffensive IKT“ das Ziel, dass KMU die Potenziale der IKT stärker nutzen, sich im Markt für IKT etablieren und wettbewerbsfähiger werden. Insbesondere sollen KMU unterstützt werden, die auf dem Gebiet der IKT tätig sind bzw. ihr Geschäftsfeld durch den Einsatz von IKT erweitern und stärken wollen.

Mit Hilfe der Resultate des Forschungsvorhabens PQ4Agile sollten KMU der Softwarebranche in die Lage versetzt werden, im Rahmen agiler Entwicklungsvorgehen hochqualitative Produkte zu entwickeln. Da Produktqualitäten wie User Experience, Performanz oder Sicherheit entscheidende Faktoren für den Produkterfolg sind, trägt dieser Ansatz zur Steigerung der Absatzchancen von Produkten deutscher KMU bei und ermöglicht diesen eine bessere Positionierung am Markt. Die Schaffung eines Alleinstellungsmerkmals durch die Qualität der Produkte dieser Unternehmen ist insbesondere im internationalen Wettbewerb von Bedeutung, da zahlreiche Firmen zu einer Offshore-Entwicklung übergehen, das heißt, Ihre Produkte in Ländern mit geringerem Lohnniveau entwickeln lassen.

Die in PQ4Agile entwickelte Methode sollte außerdem grundlegend dazu beitragen, agile Entwicklungsprojekte erfolgreich, das heißt unter gegebenen zeitlichen und finanziellen Beschränkungen durchzuführen. Dies sollte unter verschiedenen Projektfaktoren möglich werden, auch in Situationen, in denen agile Vorgehen bisher nicht optimal geeignet sind. Auch hier war das Ziel, mit PQ4Agile maßgeblich zum Erfolg der KMU beizutragen.

Darüber hinaus erwartete sich das Konsortium eine Effizienzsteigerung und Kostenreduktion bei der Projektdurchführung und in den Wartungsphasen der resultierenden Produkte. Unter ganzheitlicher Betrachtung von Produktqualität ist die Gefahr für eine späte Identifikation von Qualitätsproblemen geringer und mit einer frühen Bearbeitung ist eine günstigere Erreichung der Qualitätsziele möglich. Die gebotenen Vorgehensunterstützungen für die beteiligten Softwareingenieure sollten in diesem Zusammenhang zusätzliche Effizienzsteigerungen für KMU ermöglichen.

2 Voraussetzungen

2.1 Notwendigkeit der Zuwendung

Eine geförderte Zusammenarbeit zwischen KMU und Forschungseinrichtungen wie im PQ4Agile-Projekt ist eine geeignete Möglichkeit, um KMU eine nachhaltige Marktposition zu sichern und so den Produktionsstandort Deutschland zu stärken. Für kleine und mittelgroße Unternehmen ist eine erfolgreiche Positionierung im Bereich der Informations- und Kommunikationstechnologien eine große Herausforderung. Die Standardprodukte großer und global agierender Unternehmen stehen potentiellen Kunden kostengünstig zur Verfügung und erhöhen damit den Zeit- und Konkurrenzdruck auf KMU. Dies und die Begrenzung an personellen und finanziellen Ressourcen machen die Durchführung von risikoreichen und experimentellen Entwicklungsprojekten in solchen Unternehmen oft unmöglich, bilden aber die notwendige Grundlage für Innovationen und damit für die Schaffung von Alleinstellungsmerkmalen. Diese Einschränkung gilt insbesondere für Verfahrensforschung wo es einen fundamentalen Einfluss auf die Produktqualität gibt, der aber nur indirekt zu messen ist.

Im konkreten Fall des PQ4Agile-Projekts war eine Zusammenarbeit der beteiligten KMU mit einer Forschungsorganisation notwendig, da für die Integration von Softwareengineering-Methoden in agile Entwicklungsvorgehen ein fundiertes Methodenwissen zum Stand der Forschung notwendig war. Dieses Wissen konnte nur ein entsprechender Forschungspartner beitragen, da KMU in der Regel nicht über die erforderlichen wissenschaftlichen Kenntnisse verfügen. Insbesondere das Fraunhofer IESE als anwendungsorientierte Forschungsorganisation konnte hier einen maßgeblichen Beitrag liefern.

Eine Zusammenarbeit mit KMU ist für das Fraunhofer IESE wiederum notwendig, weil die zu untersuchenden Forschungsfragen einen starken Anwendungsfokus besitzen; das heißt, die Methoden müssen auf Basis von realen Entwicklungsprojekten entwickelt werden und anhand solcher evaluiert werden. Als Forschungsorganisation hat das Fraunhofer IESE üblicherweise keinen Zugriff auf solche Projekte, da es selbst keine Produktentwicklung betreibt und außerhalb von Kooperationen die Entwicklung in Unternehmen nicht einsehen kann.

2.2 Konsortialpartner und bisherige Arbeiten

Die folgenden Kapitel beschreiben die fachlichen Kompetenzen sowie relevante Vorarbeiten der einzelnen Konsortialpartner.

2.2.1 HK Business Solutions GmbH

Als Experte für Software- und Hardwarelösungen unterstützt die HK Business Solutions GmbH (kurz: HKBS, www.hk-bs.de) kleine und mittelständische Unternehmen bei der Optimierung ihrer Investitionen in strategische Geschäfts- und Technologieinitiativen. HKBS stellt diesen Kunden ein innovatives und umfassendes Softwareangebot bereit, bei dem passende Hardware und strukturierte Netzwerktechnik den Rahmen für eine perfekte Office-Lösung bilden. Kompetente Beratung, Planung, Installation und optimale Sicherheitsvorkehrungen gehören ebenso zum Standard wie die Zusammenarbeit mit renommierten Partnerunternehmen.

HKBS verfügt über 10 Jahre Erfahrung bei der Entwicklung firmenspezifischer Softwarelösungen für Kunden aus unterschiedlichsten Branchen. Neben der Softwareentwicklung gehört insbesondere die Einführung von ERP-, PPS- und CRM-Software bei kleinen und mittelständischen Unternehmen zum Kerngeschäft von HKBS. Die Kundenprojekte der HKBS besitzen durch ihre Laufzeit und ihren Umfang optimale Voraussetzungen, um neu entwickelte Methoden und prototypische Umsetzungen zu evaluieren; durch den Einsatz agiler Entwicklungsmethoden eigneten sie sich insbesondere für das PQ4Agile-Vorhaben.

Für die inhaltliche Arbeit brachte HKBS neben dem Know-how bei der Softwareentwicklung insbesondere Kompetenzen in den Bereichen Requirements Engineering, Usability/User Experience Engineering und intuitive Softwarebenutzung sowie bei der Durchführung von Evaluationen mit Endbenutzern in das Projekt ein. Hartmut Schmitt, Projektleiter des Gesamtvorhabens PQ4Agile sowie des Teilvorhabens von HKBS, ist seit 2006 in Forschungsprojekten auf dem Gebiet Mensch-Computer-Interaktion tätig, u. a. als Projektkoordinator in den Verbundvorhaben „FUN – Fun of Use für Geschäftsanwendungen“, „FUN-NI – Fun of Use with Natural Interactions“ und „IBIS - Gestaltung intuitiver Benutzung mit Image Schemata“, auf deren Arbeiten und Ergebnissen in PQ4Agile aufgebaut werden konnte.

2.2.2 CAS Software AG

Die CAS Software AG (kurz: CAS) ist deutscher Marktführer im Bereich XRM (Extended Relationship Management) und Informationsmanagement-Komplettlösungen für den Mittelstand. Im Bereich Standard-Software bietet die CAS praxisorientierte Lösungen für das operative Kundenmanagement und

eine effektive, unternehmensweite Zusammenarbeit. Zu den Hauptprodukten zählen die CRM-Groupware genesisWorld und die SaaS-Lösung CAS PIA, die insbesondere KMU einen schnellen Einstieg in erfolgreiches CRM erlaubt. Mit CAS OPEN steht eine PaaS-Lösung zur Verfügung, die als Entwicklungsplattform für spezifische SaaS-Branchenlösungen verwendet werden kann. Für seine innovative Produktpalette und sein Engagement im Mittelstand hat das Unternehmen mehrere Auszeichnungen erhalten.

Die CAS Software AG hat langjährige Erfahrung in der Durchführung und Koordination von nationalen (BMBF/BMWi) und europäischen Forschungsprojekten und der Umsetzung von Forschungsergebnissen in Kundenvorteile. Relevante Vorarbeiten der CAS zu den Inhalten des Projektes entstammen neben der langjährigen Erfahrung mit der Entwicklung und Vermarktung von Applikationen für CRM- und Informationsmanagement auch einer Reihe von Fördervorhaben, wie CRUISe, MODIFRAME, SumoDacs oder M3V.

2.2.3 Fraunhofer-Institut für Experimentelles Software Engineering IESE

Das Fraunhofer-Institut für Experimentelles Software Engineering IESE wurde 1996 als erste Einrichtung der Fraunhofer Gesellschaft für Angewandte Forschung in Rheinland-Pfalz gegründet. Es hat sich unter der Leitung von Prof. Dr. Dr. h.c. Dieter Rombach in kurzer Zeit zu einem international führenden Kompetenzzentrum für Softwareengineering entwickelt. Es betreibt angewandte Forschung und Beratung der Industrie in Softwareengineering-Methoden. Die Wissenschaftler für Requirements Engineering, User Experience Engineering und Softwarearchitektur leisten als international bekannte Mitglieder der Forschungsgemeinschaft stetig signifikante Beiträge zum wissenschaftlichen Fortschritt.

Das IESE führt regelmäßig Forschungsprojekte verschiedener Größe und in verschiedenen Programmen durch, sowohl national (beispielsweise EMERGENT (BMBF)) als auch international (z. B. CESAR (Artemis Joint Undertaking (JU))). Speziell zur Stärkung von KMU hat das IESE bereits in Projekte mit Fokus auf Agilität (Reqman (BMBF SE2006)) und auf Softwareproduktqualitäten (FUN-NI, IBIS (BMBF KMU-innovativ)) beigetragen.

2.2.4 YellowMap AG

Die YellowMap AG bietet das umfangreichste deutschsprachige Online-Branchenbuch. Durch Kooperationen mit führenden Portalen und Anbietern haben die Suchdienste des Branchenbuchs kontinuierlich neue Nutzer gewonnen. Ein weiteres Geschäftsfeld von YellowMap sind standortbezogene Dienste (Location Based Services, LBS) für mobile Anwender. YellowMap betreibt LBS-Dienste für Endkunden bei verschiedenen Telekommunikationsunternehmen und nutzt dazu die Erfahrung in der Bereitstellung von Umfeld-Informationen in Form von interessanten Punkten (POIs) und in der Erzeugung der Suchergebnisse eines vom Nutzer vorzugebenden geografischen Kontextes, z. B. einem Umkreis um den aktuellen Standpunkt oder einen Bereich entlang einer Route.

Bei der Durchführung von Forschungsprojekten verfügt die YellowMap AG über langjährige Erfahrungen. Durchgeführte Projekte waren zum Beispiel: Green Mobility, di.me (FP7 STREP), E-Sponder (FP7 IP).

3 Planung und Ablauf des Vorhabens

3.1 Lösungsidee

Kernidee des PQ4Agile-Projekts war es, Softwareengineering-Best-Practices zu schaffen, die Softwareentwickler bei einer strukturierten Erreichung von Qualitätsanforderungen unterstützen. Diese Best Practices sollten – entsprechend den Eigenschaften agiler Entwicklungsaktivitäten – schnell, effizient und dynamisch ausführbar sein und von den Entwicklern während der Umsetzung selbstbestimmt und nahtlos in den Entwicklungsprozess integriert werden können.

Hierfür sollten Softwareengineering-Methoden und zusammenhängende Aktivitäten aus den Bereichen Requirements Engineering, Softwarearchitektur und User Experience Engineering zunächst in Einzelteile zerlegt werden, die separat ausführbar sind. Anschließend sollten diese Aktivitäten an die Gegebenheiten agiler Entwicklungsvorgehen angepasst werden, so dass sie Softwareentwicklern als Best Practices für einen möglichst einfachen Einsatz in agilen Projekten zur Verfügung stehen. Abbildung 1 illustriert das Konzept dieser Lösungsidee:

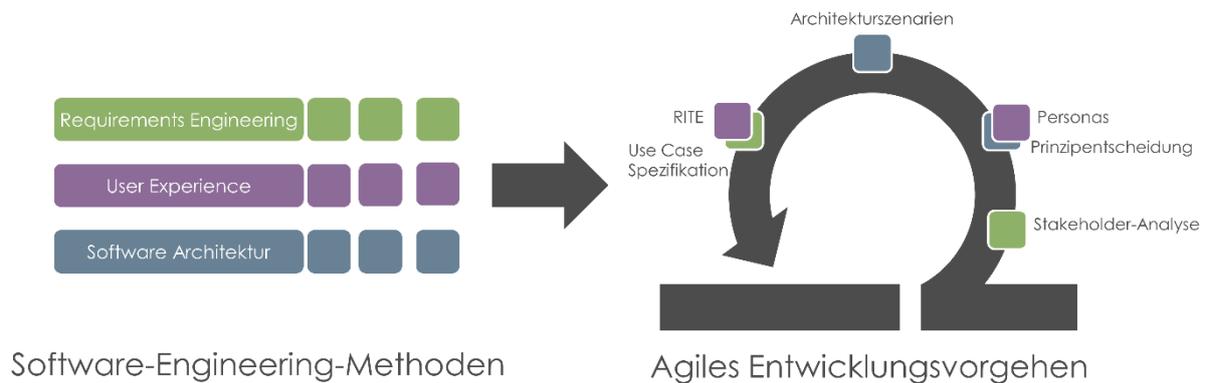


Abbildung 1: Lösungsidee des PQ4Agile-Vorhabens

Im Vorfeld des Projekts hat das Konsortium sieben Prinzipien definiert, die als Leitlinien bei der Anpassung bzw. Entwicklung der Softwareengineering-Best-Practices dienen sollten. Diese Prinzipien basieren auf den Werten und den daraus abgeleiteten Prinzipien des agilen Manifests, wurden aber auf die Idee der Software Engineering-Best Practices übertragen und entsprechend umformuliert. Zusätzlich leiten sich einige Anforderungen aus dem gegebenen Integrationsszenario zwischen agilen Vorgehen und Software-Engineering-Aktivitäten ab, wie beispielsweise „Integrierbarkeit“. Diese Anforderungen wurden ebenfalls als Prinzipien formuliert. Die vom Konsortium formulierten „Prinzipien für agile Softwareengineering-Best-Practices“ sind im Einzelnen:

- Minimalismus/Simplizität: Eine Softwareengineering-Aktivität soll nur so umfangreich ausgeführt und deren Ergebnisse so umfangreich dokumentiert werden, wie durch die gegebene Projektsituation notwendig ist, damit eine effiziente Weiterbearbeitung und Wartung möglich wird.
- Personennähe: Die Verfügbarkeit und enge Zusammenarbeit mit Stakeholdern des Kunden und von Teammitgliedern untereinander soll durch Softwareengineering-Aktivitäten genutzt werden können.
- Artefaktnähe: Softwareengineering-Artefakte sollen so produziert und verfügbar gemacht werden, dass sie möglichst effizient von Nachfolgetätigkeiten verwendet werden können und möglichst nahe am Zielprodukt sind.
- Timing: Softwareengineering-Aktivitäten sollen immer genau (bzw. spätestens) dann ausgeführt werden, wenn es für den Projektfortschritt am nützlichsten und effizientesten ist.
- Skalierbarkeit: Bei einer Zunahme von Projektfaktoren wie Stakeholdern, Systemkomplexität oder Teamgröße müssen Softwareengineering-Aktivitäten weiterhin so ausführbar sein, dass sie nicht den Grundprinzipien von agilen Vorgehen widersprechen, und dennoch genügend Unterstützung zur Erreichung von Qualitätsattributen bieten.
- Änderungsaffinität: Häufige Änderungen werden von agilen Vorgehen als essentieller Teil der Projektdurchführung gesehen. Softwareengineering-Aktivitäten sollen Änderungen ebenso begreifen und Änderungen, die die Bereiche Requirements Engineering, Softwarearchitektur

und User Experience Engineering betreffen, effizient ermöglichen und Stakeholder dabei unterstützen.

- Integrierbarkeit: Die Softwareengineering Aktivitäten müssen so beschaffen sein, dass sie eine möglichst nahtlose Integration in bestehende agile Entwicklungsprozesse erlauben.

Im Zusammenhang mit der Definition dieser Prinzipien stellten sich verschiedene Forschungsfragen, die in Abbildung 2 dargestellt sind. Diese Forschungsfragen wurden zunächst nur übergreifend für die einzelnen Prinzipien formuliert. Innerhalb des ersten Arbeitspakets von PQ4Agile, das die methodischen Grundlagen für das Projekt schuf, wurden die Forschungsfragen für die einzelnen Softwareengineering-Disziplinen präzisiert, priorisiert und es wurden die wichtigsten Fragen zur weiteren Bearbeitung ausgewählt.

	Requirements Engineering	User Experience	Software Architektur
Minimalismus/Simplizität	Wie viel Aufwand muss investiert werden und welcher Detailgrad ist ausreichend?		
Personennähe	Wie können relevante Stakeholder identifiziert und Ergebnisse effizient kommuniziert werden?		
Artefaktnähe	Wie können Ergebnisse für eine effiziente Bearbeitung in nachfolgenden Aktivitäten dokumentiert werden?		
Timing	Wann ist der optimale und der späteste Zeitpunkt zu dem die Software-Engineering-Aktivitäten ausgeführt werden können?		
Skalierbarkeit	Wie kann eine zunehmende Anzahl von Entitäten verwaltet und Konflikte aufgedeckt werden?		
Änderungsaffinität	Wie können die Kosten und Auswirkungen von Änderungen bestimmt werden?		
Integrierbarkeit	Wie müssen Software-Engineering-Aktivitäten beschaffen sein, um sie in bestehende agile Prozesse integrieren zu können?		

Abbildung 2: Forschungsfragen des PQ4Agile-Vorhabens

3.2 Projektplan

3.2.1 Laufzeit, Arbeitspakete und Meilensteinplanung

Das Projekt war auf eine Laufzeit von 24 Monaten ausgelegt (mit einem ursprünglich geplanten Förderbeginn zum 01.11.2013). PQ4Agile bestand aus fünf Arbeitspaketen zur Erarbeitung von inhaltlichen Ergebnissen („Grundlagen“, „Softwareengineering Best Practices“, „Integration Best Practices und Entwicklungsprozess“, „Evaluation“ und „Methodenkompetenz“) und einem Arbeitspaket „Projektmanagement“ (vgl. Kapitel 5).

Der Projektplan des PQ4Agile-Vorhabens sah die folgenden wichtigen Meilensteine vor:

- Meilenstein 1 (nach 6 Monaten): Die Entwicklungsprozessanalyse ist abgeschlossen; Forschungsfragen, Qualitätsmodell und Evaluationsmodell sind entwickelt (vgl. AP 1). Das Projektportal und die Social-Media-Kanäle sind aufgebaut (vgl. AP 5).
- Meilenstein 2 (nach 12 Monaten): Das Beschreibungskonzept für Softwareengineering-Best-Practices ist erstellt (AP 2); die Best-Practices sind untereinander integriert (vgl. AP 3). Die initiale Prozess- und Produktevaluation ist abgeschlossen (vgl. AP 4).
- Meilenstein 3 (nach 18 Monaten): Die Best-Practices sind vollständig beschrieben; die Untersuchung der Werkzeugunterstützung für Softwareengineering-Best-Practices ist abgeschlossen (vgl. AP 2).
- Projektabschluss (nach 24 Monaten): Die Integration der Best-Practices und der Werkzeuge in agile Entwicklungsprozesse ist abgeschlossen (vgl. AP 3); die vergleichende Analyse der Prozesse und Produkte ist abgeschlossen (vgl. AP 4). Sämtliche Projektergebnisse liegen vollständig aufbereitet vor (vgl. AP 5).

3.2.2 Risikomanagement und Abbruchkriterien

Mögliche Projektrisiken – inhaltlich-technische, personelle sowie wirtschaftliche Risiken – wurden von den Partnern im Vorfeld des Projekts untersucht und bewertet. Durch die Zusammensetzung des Konsortiums und die Ausgestaltung des Projektplans wurde versucht, alle identifizierten Risiken gezielt zu minimieren. Da innerhalb eines Forschungsvorhabens nicht sämtliche Risiken von vornherein ausgeschlossen werden können, wurden für PQ4Agile folgende Abbruchkriterien definiert, die zu frühen Zeitpunkten im Projekt überprüfbar waren:

- Es kann kein tragfähiges Qualitätsmodell für den Bereich nichtfunktionaler Produktqualität in agiler Softwareentwicklung entwickelt werden. (überprüfbar mit Meilenstein 1)
- Es kann kein anwendbares Evaluationskonzept (geeignete Metriken bzw. Evaluationswerkzeuge) zur Messung der Produkt- und Prozessqualität erstellt werden. (überprüfbar mit Meilenstein 1)
- Es kann kein geeignetes Beschreibungskonzept für die Softwareengineering-Best-Practices entwickelt werden. (überprüfbar mit Meilenstein 2)
- Fortschritte in der wissenschaftlichen oder technischen Entwicklung außerhalb des Konsortiums machen die Ergebnisse des Vorhabens PQ4Agile hinfällig.

3.3 Ablauf

Unmittelbar vor Projektstart kam es zum Ausfall eines Zuwendungsempfängers, der in seiner Rolle als Konsortialführer und Projektleiter durch HK Business Solutions ersetzt wurde. Hierdurch entstand bei der Aufnahme der inhaltlichen Arbeiten im Projekt eine ca. dreimonatige Verzögerung. Abbildung 3 zeigt den entsprechend angepassten Projektplan mit der Verteilung der Arbeitspakete und der Meilensteine auf die 24 Monate.

Die dreimonatige Verzögerung sowie eine geplante kostenneutrale Verlängerung des Projekts wurden nach Rücksprache mit dem Projektträger in der weiteren Arbeitsplanung des Projekts berücksichtigt. Bezüglich der sonstigen Arbeits- und Kostenplanung ergaben sich durch den Wechsel des Zuwendungsempfängers keine Änderungen gegenüber der ursprünglichen Planung des Vorhabens. Insbesondere entsprachen sämtliche im Berichtszeitraum durchgeführten Aktivitäten und die erzielten Ergebnisse den Ausführungen aus dem Antrag.

4 Wissenschaftlicher und technischer Stand

Die folgenden Kapitel stellen den Stand der Wissenschaft und Technik in den Bereichen agile Entwicklungsmethoden und Reifegradmodelle dar und beschreiben verwandte Projekte außerhalb des Konsortiums. Ferner werden die verwendete Fachliteratur sowie benutzte Informations- und Dokumentationsdienste genannt.

4.1 Agile Entwicklungsmethoden

Agile Entwicklungsansätze haben einen grundlegenden Wandel in der Entwicklung von Softwaresystem herbeigeführt und gewinnen seit der Veröffentlichung des „Agile Manifesto“ im Jahre 2001 zunehmend an Bedeutung. Agile Entwicklungsmethoden übertragen Kernprinzipien aus der „Lean Production“-Idee in die Softwareentwicklung. Mit der Umstellung auf agile Entwicklung sollen also hochqualitative Softwareprodukte in einer schnellen, transparenten und kostengünstigen Weise erstellt werden.

Die agile Entwicklungsmethode, die heute größte Verbreitung findet, ist Scrum. Die drei Hauptprinzipien von Scrum sind Transparenz, Überprüfung und Anpassung. Scrum definiert zur Strukturierung der Softwareentwicklung ein Vorgehensmodell mit drei Stakeholderrollen (Product Owner, Entwicklungsteam und Scrum Master), vier Ereignissen (Sprint, Sprint Planning, Daily Scrum und Sprint Review) und drei Artefakten (Product Backlog, Sprint Backlog und Burndown Chart). Ziel von Scrum ist es, eine effiziente Entwicklung von komplexen Softwareprodukten zu ermöglichen.

Scrum wird heute oft als Projektrahmenwerk verwendet und durch zusätzliche Techniken und Vorgaben erweitert, die die Projektdurchführung, aber vornehmlich auch die Produktrealisierung betreffen. Dazu gehören beispielsweise Planning Poker, Pair Programming oder Test-Driven Development. Die Arten der zusätzlich verwendeten Techniken unterscheiden sich nach Unternehmen und Projekt.

4.1.1 Qualitätserreichung in agilen Entwicklungsmethoden

In der Literatur finden sich zahlreiche Arbeiten zum Thema der Betrachtung und Erreichung von Qualität in agiler Entwicklung. Dies umfasst beispielsweise Studien, Methoden zur Erreichung von Qualität sowie systematische Literaturrecherchen. Insgesamt lässt sich jedoch erkennen, dass im aktuellen Stand der Wissenschaft bei den Begriffen Qualität und Qualitätssicherung deutlich auf die Sicherstellung von funktionaler Korrektheit und auf Codequalität fokussiert wird und dass dies entsprechend mit implementierungsnahen Techniken sicherzustellen versucht wird. Eine ganzheitliche Betrachtung von Qualität, die auch Qualitätsattribute wie Wartbarkeit oder Verfügbarkeit einschließt, ist in der Regel nicht vorhanden.

4.1.2 Scaled Agile Framework

Das Scaled Architecture Framework (SAFe) ist ein Ansatz, um die Prinzipien der agilen Softwareentwicklung auf die Eigenschaften großer Unternehmen und deren Projekte auszuweiten. Dabei sollen insbesondere eine höhere Zahl von Entwicklern und Entwicklungsteams, parallele Projekte und komplexe Projektszenarien mit globaler Verteilung berücksichtigt werden. Um diesen Herausforderungen gerecht zu werden, definiert das SAFe – aufbauend auf agilen Entwicklungsvorgehen wie Scrum – zusätzliche Rollen, Teams, Aktivitäten und Artefakte. Zur Skalierung werden auf Basis agiler Praktiken in der Ebene „Team“ zwei weitere Ebenen „Program“ (Fokus auf ein Produkt) und „Portfolio“ (Fokus auf gesamtes Unternehmen) definiert. Zentrales Artefakt des Ansatzes ist eine navigierbare Grafik, die auf der Webseite www.scaledagileframework.com öffentlich zugänglich ist.

Das SAFe berücksichtigt explizit nichtfunktionale Anforderungen, also Produktqualitäten, wie sie auch im Fokus des PQ4Agile-Projekts standen. Sie werden als Constraints an Backlog-Elemente angehängt, um diese näher zu spezifizieren. Basierend darauf beschreibt das Framework Ansätze zur Analyse, Implementierung und zum Testen von nichtfunktionalen Anforderungen.

Außerdem geht das Scaled Agile Framework im Speziellen auf die Software-Engineering-Disziplinen Software-Architektur und User Experience ein. Auf der obersten Ebene „Portfolio“ des Frameworks werden „Architectural Epics“ definiert, die das ganze Unternehmen und seine Produkte betreffen und die sich außerdem über eine längere Zeitspanne erstrecken. Sie sind also von strategischer Bedeutung. Solche Epics werden nach ihrer Priorität bearbeitet und werden entweder umgesetzt oder verworfen. Auf der „Program“-Ebene wird die Rolle des Systemarchitekten eingeführt, der mit Entwicklungsteams arbeitet und hauptsächlich für das Systemdesign, also das Treffen von Designentscheidungen verantwortlich ist. Damit kommt ihm auch eine besondere Verantwortung für die Erfüllung nichtfunktionaler Anforderungen zu. Als Artefakt auf der „Program“-Ebene wird das „Architectural Feature“ definiert. „Architectural Features“ werden von Architekten festgelegt und von den Entwicklungsteams umgesetzt. Es handelt sich dabei um technische Dienste, mit deren Hilfe potentiell gleichzeitig mehrere Endnutzerfunktionalitäten umgesetzt werden können. Da sie aber keinen direkten Endnut-

zermehrwert bringen, müssen sie als zusätzliche Entität definiert werden. Zusätzlich zur Architektur wird auch die User Experience berücksichtigt. Für die User Experience werden auf der „Program“-Ebene folgende Praktiken definiert: (1) User Experience Design als eine der Entwicklung vorausgehende Aktivität, (2) Fokussierung von Design (nicht Implementierung), (3) Nutzung von Iterationen und funktionierendem Code zur Eliminierung von Unsicherheit und Risiken und (4) Nutzung des HIP (Hardening | Innovation | Planning)-Sprints zur Einarbeitung später Änderungen.

Das Scaled Agile Framework erkennt die Wichtigkeit von Produktqualität in agilen Entwicklungsprozessen an. Dies lässt sich an der expliziten Berücksichtigung von nichtfunktionalen Anforderungen und Softwareengineering-Disziplinen erkennen. Für die Herausforderungen, denen PQ4Agile begegnen wollte, wird dennoch keine adäquate Lösung geboten. Bei der Ausgestaltung der Softwareengineering-Praktiken bleibt das SAFe zu unkonkret und überlässt diese dem jeweiligen Unternehmen oder der verantwortlichen Rolle. Der Anspruch des Projekts PQ4Agile war es jedoch, konkrete und direkt nutzbare Hilfestellung für die Projektbeteiligten zu bieten. Zusätzlich dazu wird das SAFe dafür kritisiert, dass es versucht einen neuen Entwicklungsprozess zu etablieren und dabei wieder schwergewichtige Praktiken unter anderem Namen einführen zu wollen, die bei agilen Vorgehen absichtlich abgeschafft wurden. PQ4Agile schloss diesen Weg explizit aus.

4.1.3 Agile Methoden und Software-Engineering

Neben einer Gesamtbetrachtung von Qualität in agilen Methoden sind bereits einige Arbeiten zur Verzahnung von einzelnen Softwareengineering-Methoden mit agilen Entwicklungsprozessen erschienen. Diese umfassen grundlegende Diskussionen, Literaturrecherchen, einzelne Methodenbeschreibungen, Erfahrungsberichte einer methodischen Integration und empirische Studien. Sie sind sich überwiegend einig in der positiven Einschätzung, dass Softwareengineering-Ansätze und agile Methoden grundsätzlich vereinbar sind und dass in einer Integration von Softwareengineering-Methoden und agiler Entwicklung Potentiale für eine Verbesserung der Produktqualität vorhanden sind. Es ließ sich zum Zeitpunkt des Projektstarts von PQ4Agile jedoch feststellen, dass die seinerzeit gebotene Unterstützung für die ganzheitliche Betrachtung von Qualität in agiler Entwicklung noch unzureichend war. Die Erfahrungsberichte thematisierten meist spezialisierte Lösungen, die die spezifischen Ansätze einer konkreten Situation in einem konkreten Kontext beschreiben, jedoch nicht ohne weiteres zu verallgemeinern sind. Die vorhandenen Einzelmethode wiesen nicht den benötigten Detailgrad und damit auch keine ausreichende Unterstützung auf, um ganze Entwicklungsprojekte erfolgreich zu bearbeiten. Ein umfassender Ansatz, wie er im PQ4Agile-Vorhaben geplant war, ließ sich überhaupt nicht finden.

4.2 Reifegradmodelle

Reifegradmodelle bieten die Möglichkeit zur Bewertung und Verbesserung von Entwicklungsprozessen. Dazu definieren sie ebenfalls Best Practices, deren Vorhandensein in Unternehmen bei einer Bewertung überprüft wird. Im Folgenden wird näher auf die Reifegradmodelle Capability Maturity Model Integration (CMMI) sowie Agile Maturity Model Integration (AMMI) eingegangen.

4.2.1 Capability Maturity Model Integration

Capability Maturity Model Integration (CMMI) ist eine Familie von Referenzmodellen für die Bewertung und Verbesserung von Prozessen zur Entwicklung von Produkten, zur Erbringung von Diensten und für den Einkauf von Produkten.

Jedes dieser Referenzmodelle definiert zunächst „Process Areas“, also Prozessgebiete, zur Strukturierung von Praktiken. Beispiele für solche Prozessgebiete sind „Projektplanung“, „Produktintegration“, „Messung und Analyse“ oder „Konfigurationsmanagement“. Jedes dieser Prozessgebiete gehört zu einer der drei Kategorien „Projektmanagement“, „Support“ oder „Prozessmanagement“. Innerhalb der Prozessgebiete werden bewährte Praktiken, also Best Practices, definiert. Ein Unternehmen kann in jedem Prozessgebiet einen Fähigkeitsgrad erreichen, der den Umfang der Etablierung und Durchführung der entsprechenden Praktiken repräsentiert. Die Fähigkeitsgrade sind „0 – Incomplete“, „1 – Performed“, „2 – Managed“ und „3 – Defined“. Prozessgebiet-übergreifend werden Reifegrade definiert, die aus den Fähigkeitsgraden aggregiert werden und eine Gesamtaussage über die Prozesse des Unternehmens treffen. Die Reifegrade sind „1 – Initial“, „2 – Managed“, „3 – Defined“, „4 – Quantitatively Managed“ und „5 – Optimizing“. Unternehmen sollen dadurch hauptsächlich in die Lage versetzt werden, eigene Prozesse zu verbessern. Im Rahmen von Bewertungen kann der Reifegrad von Unternehmen bestimmt werden, der dann häufig auch zur Außendarstellung verwendet wird.

CMMI definiert zwar ein Rahmenwerk und Praktiken, lässt jedoch offen, wie diese Praktiken ausgestaltet sind. Dies kann einerseits als Vorteil gesehen werden, weil es damit unabhängig von konkreten, etablierten Prozessmodellen ist, sich also auch gut mit agilen Entwicklungsvorgehen integrieren lässt. Andererseits bietet es damit aber auch keine Unterstützung bei der Ausgestaltung von Aktivitä-

ten. Diese Ausgestaltung und damit eine konkrete Hilfestellung stand jedoch gerade im Fokus des PQ4Agile-Projektes. Es sollten Entwicklern konkrete Praktiken zur Verfügung gestellt werden, um sie in der Entwicklung zu unterstützen.

4.2.2 Agile Maturity Model Integration

Agile Maturity Model Integration (AMMI) überträgt einige Prinzipien des CMMI auf agile Entwicklungsvorgehen. Es handelt sich dabei um ein Reifegradmodell für agile Softwareentwicklungsmethoden. Analog zu CMMI definiert es fünf Reifegradstufen, sogenannte „Agility Levels“, um Vorgehen in Unternehmen zu bewerten. Die Stufen sind in aufsteigender Reihenfolge: „Level 1 – Iterative & Incremental“, „Level 2 – 3-Dim. Practices“, „Level 3 – 2-Dim. Practices“, „Level 4 – 1-Dim. Practices“ und „Level 5 – Adapting Practices“. Dabei beziehen sich die Dimensionen in den Stufen 2 bis 4 auf Prozess, Produkt und Ressourcen. Eine Stufe wird erreicht, wenn eine „ausreichende“ Anzahl agiler Praktiken eingesetzt wird, um die Qualität der entsprechenden Anzahl von Dimensionen sicherzustellen.

Ebenfalls analog zu CMMI bleibt AMMI unspezifisch bezüglich der Ausgestaltung der einzelnen Praktiken, definiert keine konkreten Best Practices und macht keine weiteren Aussagen zum Begriff „ausreichend“ bzw. darüber, wie dieses Kriterium geprüft werden soll. Es bietet somit keinen Lösungsansatz für die im PQ4Agile-Projekt angegangenen Herausforderungen.

4.3 Verwandte Projekte

Dem Konsortium waren zum gegebenen Zeitpunkt keine Forschungsprojekte eines nationalen oder internationalen Programms bekannt, die das gleiche oder ein ähnliches Forschungsziel verfolgten wie PQ4Agile.

Im KMU-innovativ-Programm gibt es das Projekt Modagile Mobile (<http://www.modagile-mobile.de>), welches explizit agile Entwicklung als zentrales Thema verfolgt. Das Ziel dieses Projektes war die modellgetriebene Unterstützung der Entwicklung von Anwendungen für mobile Endgeräte. Eine dedizierte Betrachtung von Produktqualitäten in vergleichbarer Weise wie im PQ4Agile-Vorhaben stand dabei nicht im Fokus.

Das Projekt Quamoco (<http://quamoco.in.tum.de>) war ein BMBF-gefördertes Projekt, welches von einem Konsortium aus Forschungseinrichtungen und Unternehmen aus Deutschland bearbeitet wurde. Das Projekt hatte die Schaffung eines in der Praxis anwendbaren Software-Qualitätsstandards für Deutschland zum Ziel, mit dem „Software-Qualität objektiv definiert, überprüft und dauerhaft sichergestellt werden kann“. Die Motivation des Projektes basierte darauf, dass es im Softwarebereich keine allgemeingültigen Normen oder Vorschriften für Qualität gibt, so wie dies in anderen Domänen der Fall ist. Vorhandene Qualitätsrahmenwerke wie ISO 9126 waren jedoch zu allgemein, um unmittelbar in Projekten eingesetzt werden zu können. Daher hat das Konsortium ein Qualitätsmodell erarbeitet, bei dem Praxistauglichkeit und Abdeckung von möglichst vielen Softwareentwicklungsbereichen von besonderer Bedeutung waren. Damit bot Quamoco keinen Lösungsansatz für die genannten Herausforderungen, da es keine methodische Hilfestellung für Entwickler in agilen Vorgehen anbot. PQ4Agile konnte jedoch von den Ergebnissen des Quamoco-Projektes profitieren, da zur Erarbeitung der Softwareengineering-Best-Practices eine eindeutige Definition des Qualitätsbegriffes benötigt wurde. Das Quamoco-Qualitätsmodell bot hierfür eine der wesentlichen Grundlagen.

4.4 Verwendete Fachliteratur

Agile Alliance (2001): Manifesto for Agile Software Development. Verfügbar unter: agilemanifesto.org [26.06.2014]

Ian F. Alexander (2004): A Better Fit - Characterising the Stakeholders. In: Janis Grundspenkis, Marite Kirikova (Hrsg.): CAiSE 2004 Workshops (2), S. 215-223. Riga Technical University, Riga

Scott W. Ambler (2007): Introduction to Agile Usability: User Experience (UX) Activities on Agile Development Projects. Verfügbar unter: <http://www.agilemodeling.com/essays/agileUsability.htm> [24.06.2014]

Anforderungsfabrik (2014): Anforderungsvermittlung. Positionspapier, August 2014. Verfügbar unter: http://www.anforderungsfabrik.de/images/download/Anforderungsfabrik_Positionspapier_Anforderungsvermittlung_Aug14.pdf [22.09.2014]

Nils Arndt (2013): Der agile Architekt. In: Software und Support Media GmbH (Hrsg.): Entwickler Magazin 2013 (4) - Real Application Testing, S. 13-17. Software & Support Verlag, Frankfurt am Main

Helmut Balzert (1996): Lehrbuch der Software-Technik - Software-Entwicklung. Spektrum Akademischer Verlag, Heidelberg

- Helmut Balzert (2011): Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb. 3. Auflage. Spektrum Akademischer Verlag, Heidelberg
- Mario R. Barbacci, Robert J. Ellison, Anthony J. Lattanze, Judith A. Stafford, Charles B. Weinstock, William G. Wood (2003): Quality Attribute Workshops (QAWs), Third Edition. In: Software Engineering Institute (Hrsg.): Technical Report CMU/SEI-2003-TR-016. Carnegie Mellon University, Pittsburgh
- Torsten Bartel & Gesine Quint (2015): usability.de - Leistungen. Verfügbar unter: <http://www.usability.de/leistungen.html> [24.09.2015]
- Len Bass, Paul Clements, Rick Kazman (2012): Software Architecture in Practice, 3. Auflage. Addison-Wesley, Upper Saddle River
- Jörg Becker, Michael Rosemann, Reinhard Schütte (1995): Grundsätze ordnungsmäßiger Modellierung. In: Peter Mertens (Hrsg.): WIRTSCHAFTSINFORMATIK 37 (5), S. 435-445. Gabler Verlag, Wiesbaden
- Alex Beppe (2015): Story Maps – Landkarten für das Backlog. Verfügbar unter: <http://www.it-agile.de/schulung/scrum-zertifizierung/schulung-zum-certified-product-owner-cspo/story-maps> [25.09.2015]
- Martin Beschnitt (2010): Usability-Test - 16 Methoden zur Messung der Usability. Verfügbar unter: <http://www.onlinemarketing-praxis.de/web-usability/usability-test-16-methoden-zur-messung-der-usability> [22.09.2014]
- Martin Beschnitt (2010): Welche Usability-Methoden gehören heutzutage in die Toolbox eines (Web-) Usability-Professionals? Verfügbar unter: <http://www.usabilityblog.de/2010/03/welche-usability-methoden-gehoren-heutzutage-in-die-toolbox-eines-web-usability-professionals/> [22.09.2014]
- Hugh Beyer, Karen Holtzblatt, Lisa Baker (2004): An Agile User-Centered Method: Rapid Contextual Design. In: Carmen Zannier, Hakan Erdogmus, Lowell Lindstrom (Hrsg.): Extreme Programming and Agile Methods - XP/Agile Universe 2004, S. 50-59. Springer-Verlag, Berlin
- Kenneth Boness & Rachel Harrison (2007): Goal Sketching: Towards Agile Requirements Engineering. In: IEEE (Hrsg.): 2nd Internationall Conference on Software Engineering Advances ICSEA 2007, S. 71. IEEE Computer Society, Los Alamitos
- Grady Booch (2010): An Architectural Oxymoron. In: IEEE (Hrsg.): IEEE Software 27 (5), S. 96. IEEE Computer Society, Piscataway
- Nanette Brown, Robert Nord, Ipek Ozkaya (2010): Enabling Agility Through Architecture. In: Kasey Thompson (Hrsg.): CrossTalk 23 (6), S. 12-17. USAF Software Technology Support Center, Hill Air Force Base
- Bundesministerium des Innern & Bundesverwaltungsamt (2015): Handbuch für Organisationsuntersuchungen und Personalbedarfsermittlung. Verfügbar unter: http://www.orghandbuch.de/OHB/DE/ohb_pdf.pdf?__blob=publicationFile [25.09.2015]
- Lan Cao & Balasubramaniam Ramesh (2008): Agile Requirements Engineering Practices: An Empirical Study. IEEE (Hrsg.): IEEE Software 25 (1), S. 60-67. IEEE Computer Society, Piscataway
- CAS Software: Webbasiertes CRM System - sicher & professionell | CAS PIA. Verfügbar unter: <http://www.cas-pia.de> [28.09.2015]
- CESAR AVL team (2013): CESAR - Cost-Efficient methods and processes for SAFety Relevant embedded systems. Verfügbar unter: <http://www.cesarproject.eu/> [28.09.2015]
- Pooja Chinnapattan (2013): A guide to conducting Contextual Inquiry user research. Verfügbar unter: <http://www.webcredible.com/blog-reports/web-usability/contextual-inquiry.shtml> [03.12.2014]
- Mary B. Chrissis, Mike Konrad, Sandy Shrum (2006): CMMI - Richtlinien für Prozess-Integration und Produkt-Verbesserung. Verfügbar unter: <http://www.sei.cmu.edu/library/assets/cmmi-dev-v12-g.pdf> [27.06.2014]
- Paul Clements, Felix Bachmann, Len Bass, David Garlan, James Ivers, Reed Little, Paulo Merson, Robert Nord, Judith Stafford (2010): Documenting Software Architectures: Views and Beyond. Addison-Wesley, Boston
- Paul Clements, Rick Kazman, Mark Klein (2001): Evaluating Software Architectures: Methods and Case Studies. Addison-Wesley, Boston
- CMMI Product Team (2011): CMMI für Entwicklung, Version 1.3. Carnegie Mellon University, Pittsburgh
- Mike Cohn (2004): User Stories Applied for Agile Software Development. Addison-Wesley, Boston

- Thorsten Cziharz, Peter Hruschka, Stefan Queins, Thorsten Weyer (2014): Handbuch der Anforderungsmodellierung nach IREB Standard. Verfügbar unter:
http://www.ireb.org/fileadmin/IREB/Handbuecher/IREB_CPRE_Handbuch_Requirements_Modeling_Advanced_Level_v1.0-3.pdf [27.06.2014]
- Jeff Dalton: Agile CMMI. Verfügbar unter:
<https://www.upstartsystems.com/Portals/1/Agile%20CMMI%20-%20DC%20SPIN.pdf> [24.09.2015]
- Deutsche Akkreditierungsstelle (2010): Leitfaden Usability, Version 1.3. Verfügbar unter:
<http://www.dakks.de/content/leitfaden-usability> [29.09.2014]
- Deutsches Institut für Normung e. V. (2005): Qualitätsmanagementsysteme - Grundlagen und Begriffe (ISO 9000:2005); Dreisprachige Fassung EN ISO 9000:2005
- Deutsches Institut für Normung e. V. (2008): Ergonomie der Mensch-System-Interaktion - Teil 110: Grundsätze der Dialoggestaltung (ISO 9241-110:2006); Deutsche Fassung EN ISO 9241-110:2006
- Deutsches Institut für Normung e. V. (2008): Qualitätsmanagementsysteme - Anforderungen (ISO 9001:2008); Dreisprachige Fassung EN ISO 9001:2008
- Deutsches Institut für Normung e. V. (2012): DIN EN 15224: Dienstleistungen in der Gesundheitsversorgung - Qualitätsmanagementsysteme - Anforderungen nach EN ISO 9001:2008; Deutsche Fassung EN 15224:2012
- Deutsches Institut für Normung e. V. (2013): Software-Engineering - Qualitätskriterien und Bewertung von Softwareprodukten (SQuaRE) - Leitfaden für SQuaRE (ISO/IEC 25000:2005)
- Avedis Donabedian (1966): Evaluating the Quality of Medical Care. In: Ernest M. Gruenberg (Hrsg.): The Milbank Memorial Fund Quarterly 44 (3), S. 166–203. Milbank Memorial Fund, New York
- Jörg Dörr (2007): RE-Wissen.de, das Portal für Anforderungsmanagement. Verfügbar unter:
<http://www.re-wissen.de> [12.12.2014]
- Jörg Dörr (2011): Elicitation of a Complete Set of Non-Functional Requirements. In: Dieter Rombach, Peter Liggesmeyer, Frank Bomarius (Hrsg.): PhD Theses in Experimental Software Engineering 34. Fraunhofer Verlag, Stuttgart
- Jörg Dörr & Hendrik Härter (2010): Serie Software Engineering für Embedded Systeme, Teil 3: Nichtfunktionale Anforderungen systematisch erheben. Verfügbar unter:
<http://www.elektronikpraxis.vogel.de/themen/embeddedsoftwareengineering/analyseentwurf/articles/282892/> [24.09.2015]
- Jörg Dörr, Daniel Kerkow, Tom Koenig, Thomas Olsson (2006): Qualität in Software & Systemen - Ein praxiserprobter Ansatz zur Erhebung und Spezifikation von Nichtfunktionalen Anforderungen – und was kommt jetzt? In: Gesellschaft für Informatik (Hrsg.): Softwaretechnik-Trends 26 (1), S. 3-4. Gesellschaft für Informatik, Bonn
- Jörg Dörr, Tom Koenig, Thomas Olsson, Sebastian Adam (2006): Das ReqMan Prozessrahmenwerk. IESE-Report, 141.06/D. Fraunhofer IESE, Kaiserslautern
- Hugh Dubberly (2008): How do you design? A Compendium of Models. Dubberly Design Office, San Francisco
- Zoya Durdik, Benjamin Klatt, Heiko Koziulek, Klaus Krogmann, Johannes Stammel, Roland Weiss (2012): Sustainability Guidelines for Long-Living Software Systems. In: Paolo Tonella, Massimiliano Di Penta, Jonathan I. Maletic (Hrsg.): Proceedings of the 28th IEEE International Conference on Software Maintenance (ICSM 2012), S. 517-526. IEEE Computer Society, New York
- Christof Ebert (2009): Reifegradmodelle - Routenplaner für die Softwareentwicklung. In: Jens Coldewey (Hrsg.): OBJEKTspektrum 2009 (5), S. 73-79. SIGS DATACOM, Troisdorf
- Christof Ebert, Patrick Keil (2010): Reifegradmodelle: Denkfallen und Denkanstöße. In: Jens Coldewey (Hrsg.): OBJEKTspektrum 2010 (5), S. 47-53. SIGS DATACOM, Troisdorf
- John Eklund, Ciaran Levingston (2008): Usability in Agile development. Verfügbar unter:
<http://www.uxresearch.com.au/wp-content/uploads/2008/10/agile.pdf> [24.06.2014]
- Urs Enzler (2013): Agilität und Architektur: Aus dem Alltag eines agilen Softwarearchitekten. In: Thorsten Janning (Hrsg.): OBJEKTspektrum 2013 (5), S. 42-44. SIGS DATACOM, Troisdorf
- Davide Falessi, Giovanni Cantone, Salvatore A. Sarcia, Giuseppe Calavaro, Paolo Subiaco, Cristiana D'Amore (2010): Peaceful Coexistence: Agile Developer Perspectives on Software Architecture. In: IEEE (Hrsg.): IEEE Software 27 (2), S. 23–25. IEEE Computer Society, Piscataway
- Herbert Falk (2013): Die Norm ISO 9001. Verfügbar unter: <http://www.iso9001.qmb.info> [14.05.2014]

- Manfred Ferken, Peter Hruschka, Stefan Queins, Thorsten Weyer (2011): IREB Certified Professional for Requirements Engineering - Requirements Modeling, Advanced Level - Lehrplan Version 1.0. Verfügbar unter: https://www.ireb.org/content/downloads/11-syllabus-cpre-advanced-level-requirements-modeling/ireb_cpre_lehrplan_requirements_modeling_advanced_level_v1.pdf [24.09.2015]
- Jennifer Ferreira, Helen Sharp, Hugh Robinson (2012): Agile Development and User Experience Design Integration as an Ongoing Achievement in Practice. In: Randall Bilof (2012): Proceedings 2012 Agile Conference - Agile 2012, S. 11–20. IEEE Computer Society, Los Alamitos
- Eelke Folmer & Jan Bosch (2003): Usability Patterns in Software Architecture. In: Constantine Stephanidis (Hrsg.): Proceedings of the Tenth International Conference on Human-Computer Interaction, S. 93–97. Lawrence Erlbaum Associates, New Jersey
- Foraker Labs (2015): Usability Methods | Usability First. Verfügbar unter: <http://www.usabilityfirst.com/usability-methods/> [24.09.2015]
- Martin Fritzsche, Patrick Keil (2007): Agilität und Prozessreife: Erfüllbarkeit der CMMI-Prozessgebiete durch agile Methoden am Beispiel von XP. In: Wolf-Gideon Bleek, Jörg Raasch, Heinz Züllighoven (Hrsg.): Software Engineering 2007, S. 95-106. Köllen Verlag, Berlin
- FUN-NI-Projektconsortium (2013): FUN-NI - Fun of Use with Natural Interactions. Verfügbar unter: <http://fun-ni.org/> [28.09.2015]
- Gerry Gaffney (2015): Usability resources. Verfügbar unter: <http://infodesign.com.au/usabilityresources/> [25.09.2015]
- Hillel Glazer, Jeff Dalton, David Anderson, Mike Konrad, Sandy Shrum (2008): CMMI or Agile: Why Not Embrace Both! Carnegie Mellon University, Pittsburgh
- Martin Glinz (2014): A Glossary of Requirements Engineering Terminology. Verfügbar unter: http://www.certified-re.de/fileadmin/IREB/Download/Homepage%20Downloads/IREB_CPRES_Glossary_16.pdf [24.09.2015]
- Wolfgang Göbl, André Köhler (2013): Agilität und Requirements-Engineering: Ergänzung oder Widerspruch? In: Thorsten Janning (Hrsg.): OBJEKTSpektrum 2013 (4), S. 80-83. SIGS DATACOM, Troisdorf
- Rolf Götz, Heiko Scharnweber, SOPHIST GmbH (2000): IVENA: Integriertes Vorgehen zur Erhebung nichtfunktionaler Anforderungen. Verfügbar unter: <http://www.pst.informatik.uni-muenchen.de/lehre/WS0102/architektur/VL1/lvena.pdf> [24.09.2015]
- Robert B. Grady (1992): Practical Software Metrics for Project Management and Process Improvement. Prentice Hall, Englewood Cliffs
- Andrea Grass & Marcus Winteroll (2015): Schöner scheitern: Die beliebtesten Missverständnisse im agilen RE. Verfügbar unter: <http://www.oose.de/abendvortrag/schoener-scheitern-die-beliebtesten-missverstaendnisse-im-agilen-re/> [24.09.2015]
- Rainer Grau, Kim Lauenroth (2014): Requirements Engineering and Agile Development - collaborative, just enough, just in time, sustainable. Verfügbar unter: <http://www.ireb.org/fileadmin/IREB/Download/Homepage%20Downloads/REandAgile.pdf> [26.06.2014]
- Marc Hassenzahl (2003): The Thing and I: Understanding the Relationship Between User and Product. In: Mark A. Blythe, Andrew F. Monk, Kees Overbeeke, Peter C. Wright (Hrsg.): Funology: From Usability to Enjoyment, S. 31–42. Kluwer Academic Publishers, Dordrecht
- Børge Haugset & Tor Stålhane (2012): Automated Acceptance Testing as an Agile Requirements Engineering Practice. In: Ralph H. Sprague, Jr. (Hrsg.): Proceedings of the 45th Annual Hawaii International Conference on System Sciences, S. 5289-5298. IEEE Computer Society, Piscataway
- Ben Heuwing (2012): Usability-Methoden. Verfügbar unter: <http://usability-toolkit.de/usability-methoden/> [22.09.2014]
- Paul Hodgetts (2005): Experiences Integrating Sophisticated User Experience Design Practices into Agile Processes. In: Stephanie Kawada (Hrsg.): Proceedings Agile 2005, S. 235–242. IEEE Computer Society, Los Alamitos
- Eva-Maria Holt, Dominique Winter, Jörg Thomaschewski (2012): Von der Idee zum Prototypen - Werkzeuge für die agile Welt. In: Henning Brau, Andreas Lehmann, Kostanija Petrovic, Matthias C. Schroeder (Hrsg.): Usability Professionals 2012. German UPA, Stuttgart
- Karen Holtzblatt, Jessamyn Burns Wendell, Shelley Wood (2004): Rapid Contextual Design: A How-to Guide to Key Techniques for User-Centered Design. Morgan Kaufmann, San Francisco

- Colin Hood, Rupert Wiebel (2005): Optimieren von Requirements Management & Engineering. Springer-Verlag, Berlin
- Peter Hruschka & Gernot Starke (2012): Das arc42 Template. Verfügbar unter: <http://confluence.arc42.org/display/templateDE/Das+arc42+Template> [24.09.2015]
- Peter Hruschka, Chris Rupp, Gernot Starke (2004): Agility kompakt - Tipps für erfolgreiche Systementwicklung. Spektrum Akademischer Verlag, Heidelberg
- Jonathan Hung & Colin Clark (2010): UX Walkthrough Protocols and Checklists. Verfügbar unter: <https://wiki.fluidproject.org/display/fluid/UX+Walkthrough+Protocols+and+Checklists> [25.09.2015]
- Ming Huo, June Verner, Liming Zhu, Muhammad A. Babar (2004): Software Quality and Agile Methods. In: Proceedings of the 28th Annual International Computer Software and Applications Conference (COMPSAC 2004), S. 520–525. IEEE Computer Society, Piscataway
- IBIS-Projektconsortium (2014): IBIS - Gestaltung intuitiver Benutzung mit Image Schemata. Verfügbar unter: <http://www.ibis-projekt.de>. [28.09.2015]
- Institute of Electrical and Electronics Engineers (1998): 830-1998 - IEEE Recommended Practice for Software Requirements Specifications
- International Organization for Standardization / IEC – International Electrotechnical Commission (1991): Software engineering - Product quality
- International Organization for Standardization / IEC – International Electrotechnical Commission (2001): Software engineering - Product quality - Part 1: Quality model
- International Requirements Engineering Board (2012): IREB Certified Professional for Requirements Engineering - Elicitation and Consolidation, Advanced Level - Lehrplan Version 1.0. Verfügbar unter: <http://www.ireb.org/lehrplaene/advanced-level.html> [09.01.2015]
- International Requirements Engineering Board (2012): IREB Certified Professional for Requirements Engineering - Foundation Level - Lehrplan Version 2.1. Verfügbar unter: <http://www.ireb.org/lehrplaene/foundation-level.html> [01.10.2014]
- Minna Isomursu, Andrey Sirotkin, Petri Voltti, Markku Halonen (2012): User Experience Design Goes Agile in Lean Transformation - A Case Study. In: Randall Bilof (2012): Proceedings 2012 Agile Conference - Agile 2012, S. 1–10. IEEE Computer Society, Los Alamitos
- Stephan Jacobs (2014): Reifegradmodelle. Verfügbar unter: <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/is-management/Systementwicklung/reifegradmodelle><http://www.ireb.org/lehrplaene/foundation-level.html> [25.09.2015]
- Jens Jacobsen (2003): Usability-Review Nutzen und Grenzen. Verfügbar unter: <http://www.contentmanager.de/cms/usability-review-nutzen-und-grenzen/> [25.09.2015]
- Jens Jacobsen (2013): Customer Journey, User Experience, Maps und der umfassende Blick auf den Nutzer. Verfügbar unter: <http://www.usabilityblog.de/2013/09/customer-journey-user-experience-maps-und-der-umfassende-blick-auf-den-nutzer/> [25.09.2015]
- André Janus (2012): Qualitätsbasierte Bewertung Agiler Entwicklungsmethoden mit dem AMMI. In: Gesellschaft für Informatik (Hrsg.): Softwaretechnik-Trends 32 (2). Köllen Verlag, Berlin.
- André Janus, Reiner R. Dumke, Andreas Schmietendorf, Jens Jäger (2012): The 3C Approach for Agile Quality Assurance. In: Giulio Concas, Gerardo Canfora, Ewan D. Tempero, Hongyu Zhang (Hrsg.): 2012 3rd International Workshop on Emerging Trends in Software Metrics (WETSoM), S. 9-13. IEEE, Piscataway
- Sanghoon Jeon, Myungjin Han, EunSeok Lee, Keun Lee (2011): Quality Attribute Driven Agile Development. In: Roger Y. Lee (Hrsg.): 9th ACIS International Conference on Software Engineering Research, Management and Applications (SERA 2011), s. 203-210. IEEE Computer Society, Los Alamitos
- Stefan Jesse: Tutorium (2008): Erhebung von Produktanforderungen durch den Requirements Engineer. In: Georg Herzwurm, Martin Mikusz (Hrsg.): Industrialisierung des Software-Managements - Proceedings. Gesellschaft für Informatik, Bonn.
- David Kane (2003): Finding a Place for Discount Usability Engineering in Agile Development: Throwing Down the Gauntlet. In: IEEE (Hrsg.): ADC '03 - Proceedings of the Agile Development Conference 2003. IEEE Computer Society, Washington
- Ralf Kneuper (2011): Was ist eigentlich Prozessqualität? Verfügbar unter: <http://www.user.tu-berlin.de/komm/CD/paper/090712.pdf> [26.06.2014]

- Ralf Kneuper (2012): Gokyo Ri - Messung und Bewertung der Qualität von Entwicklungsprozessen. Verfügbar unter: http://www.kneuper.de/Publikationen/Vortraege/Prozessqualitaet_WI-VM_2012_Folien.pdf [28.5.2014]
- Ralf Kneuper (2012): Überblick CMMI for Development. Verfügbar unter: <http://www.kneuper.de/Cmmi/cmmi-ueberblick.html> [25.3.2014]
- Ralf Kneuper (2013): Messung und Bewertung von Prozessqualität mit Gokyo Ri. Verfügbar unter: <http://www.kneuper.de/GokyoRi> [14.5.2014]
- KompUEterchen4KMU (2014): Steckbrief-Sammlung: Usability Engineering-Methoden. Verfügbar unter: http://www.kmu-usability.de/downloads/Steckbrief-Sammlung_UE.pdf [26.06.2014]
- Ayelt Komus (2012): Internationale Studie: Status Quo Agile 2014. Ergebnisbericht (Langfassung für Studieninteressenten). Hochschule Koblenz, Koblenz
- Ayelt Komus (2012): Studie: Status Quo Agile Verbreitung und Nutzen agiler Methoden. Ergebnisbericht (Langfassung). Hochschule Koblenz, Koblenz
- KONTECXT GmbH (2015): Checklisten, Anforderungen an die technische Dokumentation. Verfügbar unter: <http://www.technische-dokumentation.de/checklisten/checklisten.html> [25.09.2015]
- Clara Lange (2010): Softwarequalitätsmodelle. Verfügbar unter: <https://www.matthes.in.tum.de/file/6ikz2i550193/sebis-Public-Website/-/Proseminar/Lange-Qualitaetsmodelle-Ausarbeitung.pdf> [14.04.2014]
- Dean Leffingwell, Alex Yakyma, Drew Jemilo, Richard Knaster, Al Shalloway, Inbar Oren: Scaled Agile Framework. Verfügbar unter: <http://scaledagileframework.com/> [25.09.2015]
- Diana Löffler, Anne Heß, Andreas Maier, Hartmut Schmitt (2012): Die IBIS-Methode. Verfügbar unter: <http://www.ibis-projekt.de/> [25.09.2015]
- Stefan Luckhaus (2014): Software-Produktivitäts-Blog. Verfügbar unter: <http://www.de.software-productivity.com/> [25.09.2015]
- Alexander Mädche, Dieter Wallach (2014): Agile und Nutzerzentrierte Softwareentwicklung. In: Franz Büllingen, Peter Stamm (Hrsg.): WISSENSCHAFT TRIFFT PRAXIS (1), S. 24-30. Begleitforschung Mittelstand-Digital, Bad Honnef
- Thomas Mangold (2013): Warum du unbedingt mit Checklisten arbeiten solltest. Verfügbar unter: <http://selbst-management.biz/warum-du-unbedingt-mit-checklisten-arbeiten-solltest/> [25.09.2015]
- Jim A. McCall, Paul K. Richards, Gene F. Walters (1977): Factors in Software Quality. US Rome Air Development Center Reports I-III. U.S. Department of Commerce, Washington
- Paul McInerney, Frank Maurer (2005): UCD in Agile Projects: Dream Team or Odd Couple? In: Jonathan Arnowitz, Elizabeth Dykstra-Erickson (Hrsg.): interactions 12 (6), S. 19-23. ACM, New York
- Michael C. Medlock, Dennis Wixon, Mick McGee, Dan Welsh (2005): The Rapid Iterative Test and Evaluation Method: Better Products in Less Time. In: Randolph G. Bias and Deborah J. Mayhew (Hrsg.): Cost-Justifying Usability: An Update for an Internet Age, S. 489-517. Morgan Kaufmann, San Francisco
- Thomas Memmel, Henning Brau, Dirk Zimmermann (2007): Agile nutzerzentrierte Softwareentwicklung mit leichtgewichtigen Usability Methoden - Mythos oder strategischer Erfolgsfaktor? In: Kerstin Röse, Henning Brau (Hrsg.): Usability Professionals 2007, S. 223-227. Fraunhofer IRB, Stuttgart
- Rolf Molich, Jakob Nielsen (1990): Improving a Human-Computer Dialogue. In: Edgar H. Sibley (Hrsg.): Communications of the ACM 33 (3), S. 338-348. ACM, New York
- Kerstin Nagel, Martina Hafner (2009): Echte Wiederverwendung von Anforderungen. Verfügbar unter: <http://www.elektronikpraxis.vogel.de/themen/embeddedsoftwareengineering/planung/articles/238765/index3.html> [12.12.2014]
- Jakob Nielsen (1993): Usability Engineering. Morgan Kaufmann, San Francisco
- Jakob Nielsen (1994): Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier. Verfügbar unter: <http://www.nngroup.com/articles/guerrilla-hci/> [29.09.2014]
- Jakob Nielsen (1995): 10 Usability Heuristics for User Interface Design. Verfügbar unter: <http://www.nngroup.com/articles/ten-usability-heuristics/> [24.09.2015]
- Jakob Nielsen (1995): How to Conduct a Heuristic Evaluation. Verfügbar unter: <http://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/> [24.09.2015]
- Jakob Nielsen (1995): Summary of Usability Inspection Methods. Verfügbar unter: <http://www.nngroup.com/articles/summary-of-usability-inspection-methods/> [29.09.2014]

- Jakob Nielsen (1995): Technology Transfer of Heuristic Evaluation and Usability Inspection. Verfügbar unter: <http://www.nngroup.com/articles/technology-transfer-of-heuristic-evaluation/> [24.09.2015]
- Jakob Nielsen (1997): Discount Usability for the Web. Verfügbar unter: <http://www.nngroup.com/articles/web-discount-usability/> [29.09.2014]
- Jakob Nielsen (2009): Discount Usability: 20 Years. Verfügbar unter: <http://www.nngroup.com/articles/discount-usability-20-years/> [29.09.2014]
- Robert L. Nord & James E. Tomayko (2006): Software Architecture-Centric Methods and Agile Development. In: IEEE (Hrsg.): IEEE Software 23 (2), S. 47-53. IEEE Computer Society, Piscataway
- Ken Orr (2004): Agile Requirements: Opportunity or Oxymoron? In: Suzanne Robertson (Hrsg.): IEEE Software 21 (3), S. 71-73. IEEE Computer Society, Piscataway
- Christian Patschke (2014): Irrtum Nr. 9: Requirements Engineering in Projekten kostet nichts. Verfügbar unter: <http://blog.hood-group.com/blog/2014/04/29/irrtum-nr-9-requirements-engineering-in-projekten-kostet-nichts/> [25.04.2015]
- Horst Peterjohann (2015): Requirements Engineering: Eine Einführung (RE-Basispräsentation). Verfügbar unter: http://www.peterjohann-consulting.de/_pdf/peco-re-einfuehrung.pdf [25.04.2015]
- Roman Pichler (2009): The Product Vision. Verfügbar unter: <https://www.scrumalliance.org/community/articles/2009/january/the-product-vision> [25.09.2015]
- Roman Pichler (2012): A Product Canvas for Agile Product Management, Lean UX, Lean Startup. Verfügbar unter: <http://www.romanpichler.com/blog/the-product-canvas/> [23.04.2015]
- Roman Pichler (2012): Agiles Produktmanagement mit Scrum. Addison-Wesley Verlag, München
- Roman Pichler (2012): The Product Canvas. Verfügbar unter: <http://www.romanpichler.com/blog/the-product-canvas/> [25.09.2015]
- Roman Pichler (2013): Introducing the Product Canvas. Verfügbar unter: <http://jaxenter.com/introducing-the-product-canvas-105480.html> [23.04.2015]
- Roman Pichler (2013): Roman Pichler's Product Canvas Tutorial Version 1.0. Verfügbar unter: <http://www.romanpichler.com/blog/product-canvas-tutorial/> [23.04.2015]
- Roman Pichler (2014): 8 Tips for Creating A Compelling Product Vision. Verfügbar unter: <http://www.romanpichler.com/blog/tips-for-writing-compelling-product-vision/> [24.09.2015]
- Platinum Edge (2012): Agile Artifacts: The Product Vision Statement. Verfügbar unter: <https://platinumedge.com/blog/agile-artifacts-product-vision-statement> [24.09.2015]
- Hans-Jürgen Plewan, Benjamin Poensgen (2011): Produktive Softwareentwicklung: Bewertung und Verbesserung von Produktivität und Qualität in der Praxis. dpunkt Verlag, Heidelberg
- Hans-Jürgen Plewan, Benjamin Poensgen (2012): Produktive Softwareentwicklung: Acht Faktoren für mehr Produktivität und Qualität. Thorsten Janning (Hrsg.): OBJEKTSpektrum 2012 (2), S. 12-17. SIGS DATACOM, Troisdorf
- Klaus Pohl, Chris Rupp (2010): Basiswissen Requirements Engineering. 2. Auflage. dpunkt Verlag, Heidelberg
- Lutz Reder, Christoph Theis (2013): Professionelles Requirements Engineering in Scrum-Projekten. In: Software und Support Media GmbH (Hrsg.): Entwickler Magazin Ausgabe 2013 (6), S. 11-18. Software & Support Verlag, Frankfurt am Main
- James Robertson, Suzanne Robertson (2014): Volere Requirements Specification Template - Extracts and Samples from the Template. Verfügbar unter: <http://www.volere.co.uk/template.htm> [06.10.2014]
- Holger Röder (2012): Usability Patterns – Eine Technik zur Spezifikation funktionaler Usability-Merkmale. Cuvillier Verlag, Göttingen
- Holger Röder (2012): Usability Patterns: Katalog bewährter funktionaler Lösungsmuster für interaktive Software-Systeme. Verfügbar unter: <http://www.usabilitypatterns.info/catalog/catalog.html> [25.09.2015]
- Curt Ronninger (2012): Systemanalyse. Verfügbar unter: <http://www.crgraph.de/Systemanalyse.pdf> [02.10.2014]
- Jim Ross (2012): Why Are Contextual Inquiries So Difficult? Verfügbar unter: <http://www.uxmatters.com/mt/archives/2012/06/why-are-contextual-inquiries-so-difficult.php> [03.12.2014]

- Dominik Rost, Matthias Naab, Crescencio Lima, Christina von Flach Garcia Chavez (2013): Architecture Documentation for Developers: A Survey. IESE-Report, 028.13/E. Fraunhofer IESE, Kaiserslautern
- Nick Rozanski, Edin Woods (2005): Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives. Addison-Wesley, Boston
- Christiane Rudlof (2006): Handbuch Software-Ergonomie - Usability Engineering, 2. Auflage. Unfallkasse Post und Telekom, Tübingen
- Chris Rupp, die SOPHISTen (2009): Requirements-Engineering und -Management: Professionelle, iterative Anforderungsanalyse für die Praxis. 5. Auflage. Hanser Verlag, München
- Chris Rupp, die SOPHISTen (2014): Requirements-Engineering und -Management: Aus der Praxis von klassisch bis agil. 6. Auflage. Hanser Verlag, München
- Jeff Sauro (2010): 6 things you didn't know about Heuristic Evaluations. Verfügbar unter: <http://www.measuringu.com/blog/he.php> [24.09.2015]
- Jeff Sauro (2011): What's the difference between a Heuristic Evaluation and a Cognitive Walkthrough? Verfügbar unter: <http://www.measuringu.com/blog/he-cw.php> [24.09.2015]
- Andreas Schlenkhoff (2012): Duden Ratgeber - Technische Dokumentation. Beschreibende und anleitende Texte erstellen. Bibliographisches Institut, Mannheim
- Eva-Maria Schön (2014): Menschzentriertes Vorgehensmodell für einen agilen Produktentwicklungsprozess. Masterarbeit. Hochschule Emden/Leer, Emden
- Christian Schubert, Michael Kläs, Jonathan Streit, Andreas Göb, Klaus Lochmann, Stefan Wagner, Reinhold Plösch, Christian Körner (2012): Schlussbericht des Verbundprojektes QuaMoCo - Software Qualität: Flexible Modellierung und integriertes Controlling. Capgemini Deutschland, Berlin
- Peter Schuh (2005): Integrating Agile Development in the Real World. Charles River Media, Hingham
- Ken Schwaber, Jeff Sutherland (2001): Scrum Guide - Der gültige Leitfaden für Scrum: Die Spielregeln. Verfügbar unter: <https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum%20Guide%20-%20DE.pdf> [26.06.2014]
- Benjamin Seidler (2015): Product Canvas: Werkzeugkasten und Tools für den Agilen Requirements Engineer. Verfügbar unter: <https://www.andrena.de/event/objektforum-stuttgart-september2015> [24.09.2015]
- Kivanc Semen & Ralf Kneuper (2011): Kombination von agiler Softwareentwicklung und CMMI in kleinen Umgebungen. Verfügbar unter: <http://www.kneuper.de/Publikationen/Vortraege/see2011.pdf> [25.09.2015]
- Angshu M. Sen & K. Hemachandran (2010): Elicitation of Goals in Requirements Engineering Using Agile Methods. In: Sheikh Iqbal Ahamed, Doo-Hwan Bae, Sungdeok Cha, Carl K. Chang, Rajesh Subramanyan, Eric Wong, Hen-I Yang (Hrsg.): Proceedings 34th Annual IEEE International Computer Software and Applications Conference COMPSAC 2010 Volume II: Workshops, S. 263–268, IEEE Computer Society, Piscataway
- Panagiotis Sfetsos & Ioannis Stamelos (2010): Empirical Studies on Quality in Agile Practices: A Systematic Literature Review. In: Fernando Brito e Abreu, João Pascoal de Faria, Ricardo J. Machado (Hrsg.): Proceedings Seventh International Conference on the Quality of Information and Communications Technology, S. 44-53. IEEE Computer Society, Washington
- James Shore (2010): The Art of Agile Development: Vision. Verfügbar unter: <http://www.jamesshore.com/Agile-Book/vision.html> [24.09.2015]
- Tiago Silva da Silva, Angela Martin, Frank Maurer, Milene Silveira (2011): User-Centered Design and Agile Methods: A Systematic Review. In: Lisa O'Conner (Hrsg.): Agile 2011 Conference, S. 77-86. CPS Conference Publishing Services, Los Alamitos
- Simply usable (2015): Leitfaden Kontextanalyse. Verfügbar unter: http://www.simply-usable.de/wp-content/uploads/2015/06/M-LKA_Leitfaden_Kontextanalyse.pdf [24.09.2015]
- Mona Singh (2008): U-SCRUM: An Agile Methodology for Promoting Usability. In: Grigori Melnik, Philippe Kruchten, Mary Poppendieck (Hrsg.): Proceedings Agile 2008 Conference, S. 555-560. IEEE Computer Society, Los Alamitos
- Software-Cluster Koordinierungsstelle (2015): EMERGENT - Grundlagen Emergenter Software. Verfügbar unter: <http://www.software-cluster.org/de/forschung/projekte/verbundprojekte/emergent> [28.09.2015]

- Osama Sohaib, Khalid Khan (2010): Integrating Usability Engineering and Agile Software Development: A Literature Review. In: Jinkuan Wang, Bin Wang (Hrsg.): 2010 International Conference on Computer Design and Applications 2, S. 32-38. IEEE Computer Society, Washington
- Osama Sohaib, Khalid Khan (2011): Incorporating Discount Usability in Extreme Programming. In: Haeng-kon Kim (Hrsg.): International Journal of Software Engineering and Its Applications 5 (1), S. 51-62. Science & Engineering Research Support Center, Daegu
- Die SOPHISTen (2015): Die kleine RE-Fibel. 2. Auflage. SOPHIST GmbH, Nürnberg
- Jörg Spiegelhoff (2012): Produktivität in Softwareentwicklungsprojekten. Verfügbar unter: <https://blog.codecentric.de/2012/02/produktivitat-in-softwareentwicklungsprojekten/> [25.09.2015]
- Gernot Starke, Peter Hruschka (2009): Software-Architektur kompakt - angemessen und zielorientiert. Spektrum Akademischer Verlag, Heidelberg.
- Gernot Starke, Peter Hruschka (2014): arc42 - Ressourcen für Softwarearchitekten. Verfügbar unter: <http://www.arc42.de/> [20.10.2014]
- Ruth Stock-Homburg (2010): Personalmanagement: Theorien - Instrumente - Konzepte. 2. Auflage. Gabler Verlag, Wiesbaden
- Maren Stypka (2009): Untersuchungen zur Strukturierten Dokumentation mit DITA im Bereich der Schienenfahrzeugtechnik. Diplomarbeit. Hochschule Merseburg, Merseburg
- Desirée Sy (2007): Adapting Usability Investigations for Agile User-Centered Design. In: Avi Parush (Hrsg.): Journal of Usability Studies 2 (3), S. 112-132. Usability Professionals Association, Bloomington
- Manfred Thüring (2014): UseTree: Wissen - Usability aus Berlin. Verfügbar unter: <http://www.usetree.de/category/wissen/?filter=48> [03.12.2014]
- Jenifer Tidwell (2010): Designing Interfaces. 2. Auflage. O'Reilly, Sebastopol
- Craig Tomlin (2010): How to Conduct a Usability Review. Verfügbar unter: <http://www.usefulusability.com/how-to-conduct-a-usability-review/> [24.09.2015]
- Neil Turner (2011): A guide to carrying out usability reviews. Verfügbar unter: <http://www.uxforthemasses.com/usability-reviews/> [18.12.2014]
- U.S. Department of Health and Human Services (2014): Methods | Usability.gov. Verfügbar unter: <http://www.usability.gov/how-to-and-tools/methods/index.html> [22.09.2014]
- usability-toolkit.de (2010): Usability-Methoden. Verfügbar unter: <http://usability-toolkit.de/usability-methoden/> [24.09.2015]
- UsabilityNet (2006): Methods table. Verfügbar unter: <http://www.usabilitynet.org/tools/methods.htm> [01.10.2014]
- User Experience Stack Exchange (2012): How to find out usability problems myself? Verfügbar unter: <http://ux.stackexchange.com/questions/16130/how-to-find-out-usability-problems-myself> [25.09.2015]
- The User Experience Professionals' Association (2012): Usability Body of Knowledge. Verfügbar unter: <http://www.usabilitybok.org/> [03.12.2014]
- Martijn van Welie (2008): A Pattern Library for Interaction Design. Verfügbar unter: <http://www.welie.com> [14.01.2015]
- Stefan Wagner, Klaus Lochmann, Lars Heinemann, Michael Kläs, Adam Trendowicz, Reinhold Plösch, Andreas Seidl, Andreas Goeb, Jonathan Streit (2012): The quamoco product quality modelling and assessment approach. In: Martin Glinz, Gail Murphy, Mauro Pezzè (Hrsg.): Proceedings of the ICSE 2012. IEEE Press, Piscataway.
- Ernest Wallmüller (2002): Qualitätsmodelle im Software Engineering: Boden unter den Füßen. In: Galledia (Hrsg.): MQ - Management und Qualität 2002 (9). Galledia Verlag, Berneck
- Oliver Wenk (2011): Mit Checklisten Zeit sparen und Nerven schonen – Wie funktioniert das? Verfügbar unter: <http://www.gruenderhelden.de/mit-checklisten-zeit-sparen-und-nerven-schonen-wie-funktioniert-das/> [25.09.2015]
- Birger Wernerfelt (1984): A Resource-based View of the Firm, Strategic Management Journal 1984 (5), S.171-180. John Wiley & Sons, New York
- Dominique Winter, Eva-Maria Schön, Jan Uhlenbrok, Jörg Thomaschewski (2013): User Experience in Kanban - Die UX-Karte ausspielen. In: Henning Brau, Andreas Lehmann, Kostanija Petrovic, Matthias C. Schroeder (Hrsg.): Usability Professionals 2013, S. 220-224. German UPA, Stuttgart



Marcus Winteroll (2010): Agiles Requirements-Engineering. Verfügbar unter:
http://www.sigs.de/download/season_10/Mo_2-1_Winteroll.pdf [24.09.2015]

Roger Wolfer (2010): Produktvision und Requirements Engineering. Verfügbar unter:
https://files.ifi.uzh.ch/req/amadeus/teaching/courses/re_seminar_spm_fs10/Produktvision.pdf
[24.09.2015]

Daniel Ziegler (2014): Usability inside - Maßnahmen und Methoden. Verfügbar unter:
<http://www.usability-inside.net/wissen/massnahmen-und-methoden/> [25.09.2015]

4.5 Benutzte Informations- und Dokumentationsdienste

Von den Partnern wurden im Rahmen des Projekts diverse Informations- und Dokumentationsdienste genutzt, um Informationen zu gewinnen, aber auch um Neuigkeiten aus dem PQ4Agile-Projekt zu verbreiten. Neben der freien Enzyklopädie Wikipedia (www.wikipedia.de) und diversen Nachrichtenportalen für Softwareentwickler sind hier insbesondere die Newsletterdienste des BITMi (www.bitmi.de), des Bitkom (www.bitkom.org), des Software Clusters (www.software-cluster.org), des CyberForums (www.cyberforum.de) und der German UPA (www.germanupa.de) zu nennen.

5 Verwendung der Zuwendung und Projektergebnisse

Das PQ4Agile-Vorhaben umfasste sechs Arbeitspakete. In den folgenden Kapiteln sind die Ziele dieser Arbeitspakete, die jeweilige Verwendung der Zuwendung sowie die erzielten Ergebnisse näher erläutert. Die Laufzeitangaben bei den Arbeitspaketen entsprechen dem zu Projektbeginn angepassten Zeitplan (vgl. Kapitel 3.3).

5.1 Arbeitspaket 1: Grundlagen

Laufzeit: Februar bis Juli 2014

Lead: Fraunhofer IESE

5.1.1 Ziele des Arbeitspakets

Das Arbeitspaket 1 diente zur Schaffung der Grundlagen für die weiteren Arbeiten im PQ4Agile-Vorhaben. In diesem Arbeitspaket sollten daher agile Entwicklungsprozesse analysiert werden, konkrete Herausforderungen bei der Integration von agilen Vorgehen und Softwareengineering-Methoden identifiziert werden und außerdem Projektkontextfaktoren ermittelt werden, die einen Einfluss auf die Qualität der erstellten Produkte haben.

5.1.2 Verwendung der Zuwendung

Arbeitspaket 1 umfasste die folgenden Aktivitäten:

AP 1.1: Entwicklungsprozessanalyse

In AP 1.1 wurden die Entwicklungsprozesse der beteiligten KMU CAS, HKBS und YellowMap erhoben und analysiert. Hierfür wurden vom IESE zunächst Erhebungs- und Dokumentationswerkzeuge (Fragenkatalog, Interviewleitfaden) erstellt und mit den anderen Partnern abgestimmt; auf dieser Grundlage hat HKBS ein Word-Template für die Prozessdokumentation erarbeitet. Von den KMU wurde eine fragebogenbasierte Erhebung des Ist-Entwicklungsprozesses (Methoden, Tools usw.) durchgeführt, deren Ergebnisse vom IESE konsolidiert wurden. Gegenstand der Erhebung waren die durchgeführten Aktivitäten, die beteiligten Rollen, die durchgeführten Rituale bzw. Meetings und die eingesetzten Werkzeuge. Anschließend wurden vom IESE Workshops bei den KMU ausgerichtet, an denen jeweils zwei wissenschaftliche Mitarbeiter des IESE und mehrere relevante Stakeholder des KMU teilnahmen. Die Ergebnisse der Prozessanalyse wurden vom jeweiligen KMU aufbereitet (textuelle Beschreibung und Diagramm) und vom IESE begutachtet. Unterstützend haben die Partner etablierte agile Entwicklungsprozesse (insbesondere Scrum) sowie CMMI und AMMI analysiert.

Auf Basis dieser Prozessdokumentationen haben die Partner in einem Workshop einen Referenzprozess definiert, der die Gemeinsamkeiten der verschiedenen Prozesse konsolidiert und gleichzeitig auf die spezifischen Umsetzungen bei den Partnern verweist. Hierdurch wurde sichergestellt, dass bei der späteren Entwicklung der Best Practices eine hohe Verallgemeinerbarkeit gegeben war, jedoch bei vorhandener Verbindung zu den Spezifika der einzelnen Unternehmen. Bei der Untersuchung zeigte sich, dass die erhobenen Rollen, Rituale bzw. Meetings und Werkzeuge eine zu hohe Spezifität aufweisen, um eine Verallgemeinerbarkeit zu erreichen. Der Referenzprozess umfasst daher ausschließlich Aktivitäten. Diese Aktivitäten sind in folgende Bereiche unterteilt: „Anforderungen“, „Planung und Design“, „Evaluation“, „Realisierung“, „Kontrolle“ und „Projektplanung und -Steuerung“. Um zu vermeiden, dass der Referenzprozess essentielle Aktivitäten unberücksichtigt lässt, die aus bestimmten Gründen bei den Unternehmen nicht angewendet werden, wurde dieser gegen Scrum, CMMI und verschiedene Prozessbeschreibungen aus der Softwareengineering-Literatur validiert und entsprechend ergänzt.

AP 1.2: Forschungsfragen und Qualitätsmodell

In AP 1.2 definierten die Partner die konkreten Forschungsfragen für PQ4Agile, die bei der späteren Entwicklung der Best Practices helfen sollten. Hierzu wurden die in Abbildung 2 dargestellten Forschungsfragen (z. B. „Wie können Ergebnisse für eine effiziente Bearbeitung in nachfolgenden Aktivitäten dokumentiert werden?“) für die Bereiche Requirements Engineering, Softwarearchitektur und User Experience von den Partnern zunächst ausdetailliert. Die so gewonnenen Leitfragen wurden in einem Workshop zusammengeführt, gefiltert und gemeinsam bewertet.

Hierbei wurde zwischen unterschiedlichen Detailgraden für die weitere Bearbeitung unterschieden:

- **Niedrig:** Es gibt bestehende Aktivitäten, für die es keinen oder kaum Anpassungsbedarf gibt und die daher nur selektiert werden müssen. Es sind eine geringere Detailtiefe und weniger Aufwand für die Lösungserarbeitung notwendig.

- Medium: Es gibt bestehende Aktivitäten, für die es Anpassungsbedarf gibt, allerdings mit bestehenden Vorarbeiten. Aktivitäten müssen selektiert und geringe bis mittlere Anpassungen durchgeführt werden. Es sind eine mittlere Detailtiefe und mittlerer Aufwand für die Ergebniserarbeitung notwendig.
- Hoch: Es sind kaum Vorarbeiten vorhanden, neue Best-Practices müssen grundlegend erarbeitet werden. Es sind eine hohe Detailtiefe und im Vergleich hoher Aufwand für die Ergebniserarbeitung notwendig.

Auf diese Weise wurde sichergestellt, dass von allen möglichen Forschungsfragen in PQ4Agile gezielt die relevantesten Fragestellungen weiterbearbeitet werden.

Weiterhin haben das IESE und HKBS in AP 1.2 verschiedene Qualitätsmodelle analysiert (insbesondere hinsichtlich ihrer Verwendbarkeit für PQ4Agile) und hierauf basierend ein integriertes Qualitätsmodell für die Bereiche Produkt-, Prozess- und Strukturqualität erstellt. Hiermit sollte die Basis für ein einheitliches Qualitätsverständnis und damit die Grundlage für alle weiteren Aktivitäten im Projekt geschaffen werden. Für jeden der drei Teilbereiche diente ein bestehendes Qualitätsmodell als Basis, die ISO 25010 für den Bereich Produktqualität, das von Ralf Kneuper entwickelte Modell Gokyo Ri für den Bereich Prozessqualität und die ISO 9001 für den Bereich Strukturqualität. Die verwendeten Qualitätsmodelle definieren die Qualitätsattribute der jeweiligen Bereiche auf zwei Hierarchiestufen als Qualitätsmerkmale und Qualitätsteilmerkmale. Diese können wiederum in Qualitätsmerkmalsgruppen zusammengefasst werden. Für das PQ4Agile-Qualitätsmodell wurden die (Teil-)Merkmale aus den zugrundeliegenden Qualitätsmodellen übernommen, bei Bedarf angepasst und um weitere (Teil-)Merkmale ergänzt. Verknüpfungspunkte und Überlappungen zwischen einzelnen Teilbereichen wurden explizit gemacht. Das resultierende Modell deckte somit alle relevanten Bereiche in zusammenhängender Weise ab. Abbildung 4 zeigt eine Visualisierung der Struktur des Qualitätsmodells:

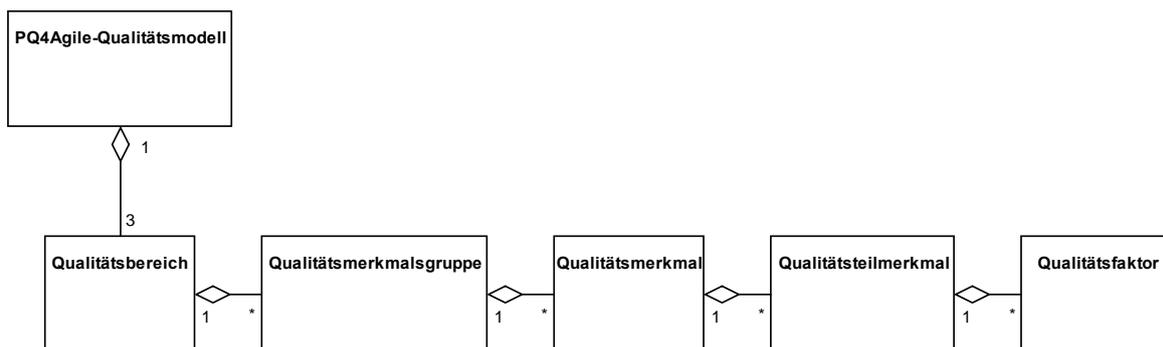


Abbildung 4: Struktur des PQ4Agile-Qualitätsmodells

Das Qualitätsmodell wurde vom IESE und HKBS ausgearbeitet und dokumentiert. Die (Zwischen-) Ergebnisse wurden mit CAS und YellowMap in mehreren Workshops abgestimmt. Als integriertes Modell bildete das PQ4Agile-Qualitätsmodell die Basis für das grundlegende Verständnis der verschiedenen Qualitätsbereiche. Für den weiteren Projektverlauf in PQ4Agile war jedoch nur eine Untermenge der beschriebenen Bereiche, Merkmale und Teilmerkmale von Relevanz.

AP 1.3: Evaluationskonzept

In AP 1.3 erarbeiteten die Partner unter Leitung des IESE ein Evaluationskonzept für agile Entwicklungsprozesse und die mit ihnen erstellten Produkte; dieses bildete die Grundlage für die Erhebungen zur Produkt- und Prozessqualität in AP 4. Die grundlegenden Konzepte zur Produkt- und Prozessevaluation wurden vom IESE eingebracht; im Bereich Prozessqualität hat HKBS eine Analyse existierender Bewertungsverfahren und -tools (z. B. SPiCE-Lite Assessment Tool) durchgeführt.

Für die Evaluationsdurchführung wurde vom IESE ein excelbasiertes Erhebungstemplate erstellt. Die Ausarbeitung der Fragebögen erfolgte durch das IESE (Bereich Produktqualität) und HKBS (Bereich Prozessqualität). Anschließend wurde das Evaluationskonzept mit allen Partnern abgestimmt. Um sicherzustellen, dass die Werkzeuge und Metriken für die Evaluationen in Arbeitspaket 4 geeignet sind, führten HKBS und IESE im eigenen Kontext jeweils Probedurchläufe durch.

5.1.3 Ergebnisse

Der in AP 1.1 erarbeitete **agile Referenzprozess** bildete eine der Grundlagen für die in AP 2.2 erarbeiteten Best Practices. Als Anknüpfungspunkte für die Best Practices dienten die 22 Aktivitäten des Referenzprozesses. Abbildung 5 illustriert den PQ4Agile-Referenzprozess.

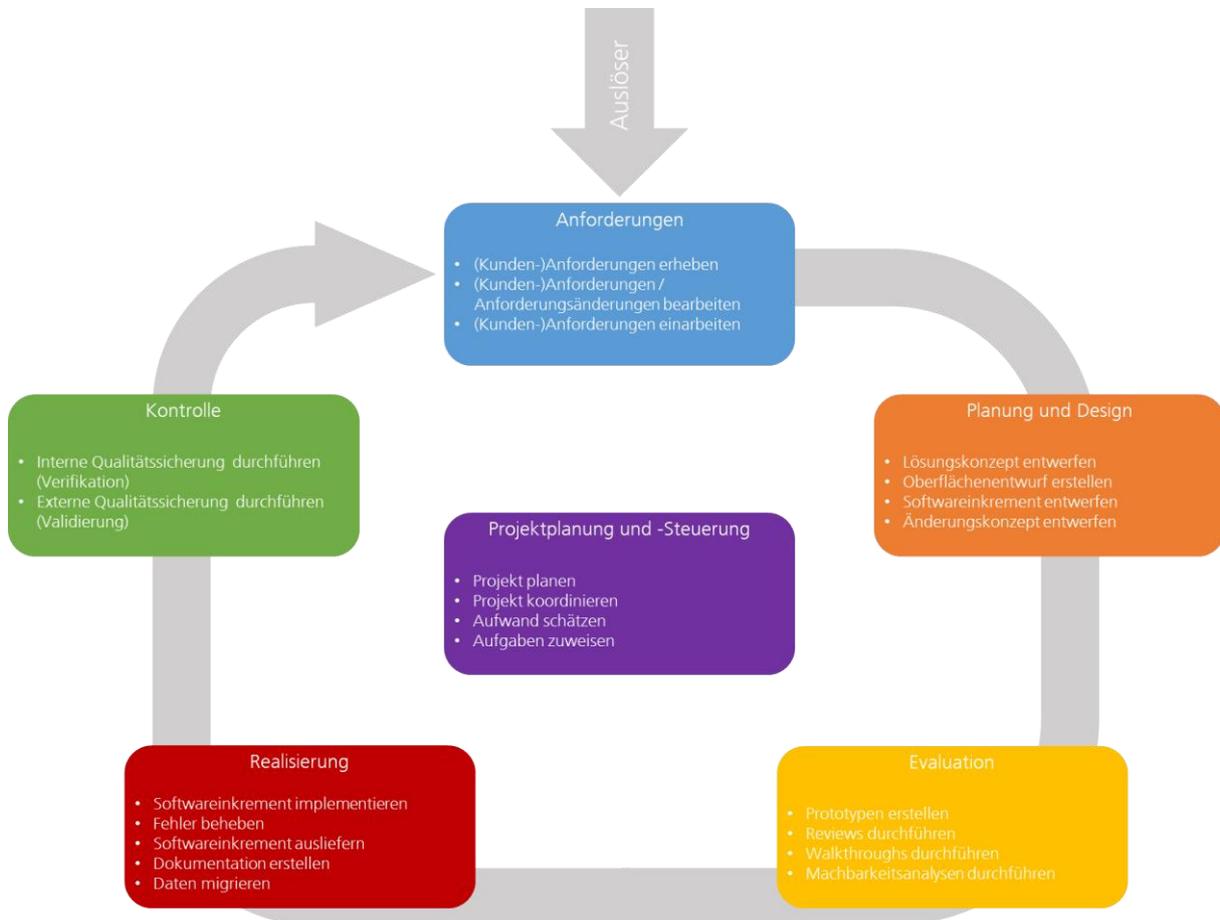


Abbildung 5: Agiler Referenzprozess des PQ4Agile-Vorhabens

Mit Ausnahme des Bereichs Projektplanung und -Steuerung werden die beschriebenen Aktivitäten während des Entwicklungsprozesses iterativ ausgeführt. Ein Neudurchlauf wird durch das Auftreten eines bestimmten Auslösers initiiert. Dies können beispielsweise das Auftreten neuer oder geänderter Kundenanforderungen, Umpriorisierungen, Änderungen von Projektfaktoren oder auch einfach der Start eines neuen Sprints sein. Diese wiederholte Ausführung ist in Abbildung 5 mit den grauen Pfeilen illustriert.

Auch der in AP 1.2 entwickelte **Katalog disziplinspezifischer Forschungs- und Leitfragen** lieferte eine Grundlage für die spätere Ausarbeitung der Best Practices. Der Fragenkatalog diente dazu, bestehende Software-Engineering-Aktivitäten zu selektieren, die sich grundlegend für die Anwendung in agilen Entwicklungsvorgehen eignen, und die Anpassungsbedarfe bei den Software-Engineering-Aktivitäten zu identifizieren bzw. die damit verbundenen voraussichtlichen Aufwände zu bestimmen. Abbildung 6 zeigt einige Beispiele aus dem entwickelten Fragenkatalog:

3.1 Minimalismus

Disziplin	Leitfrage	Detailgrad
Architektur	In welchem Umfang und Detailgrad muss ein System mindestens beschrieben werden?	Medium
RE/UX	Wie können von abstrakten Qualitätsattributen Anforderungen für das System und die User Experience abgeleitet werden?	Hoch
RE/UX	Können Heuristiken für den Detailgrad und den Umfang der Anforderungserfassung definiert werden? Wie kann die User Experience des Systems mit wenig Aufwand im Entwicklungsprozess getestet werden?	Niedrig

3.2 Personennähe

Disziplin	Leitfrage	Detailgrad
Architektur	Wie können Designentscheidungen effizient und effektiv ins Team transportiert werden?	Hoch
Architektur	Welche Maßnahmen begünstigen, dass Designentscheidungen vom gesamten Team getragen werden?	Medium
Architektur	Wie kann eine effiziente Entscheidungsfindung (im Team) ermöglicht werden?	Niedrig
RE/UX	Wie kann eine integrierte Konzeption von User Experience und Architektur aussehen und welche Abhängigkeiten gibt es dazwischen?	Hoch
RE/UX	Wie können relevante Stakeholder und deren Concerns im Projekt identifiziert, berücksichtigt, balanciert und erfüllt werden?	Niedrig

Abbildung 6: Forschungs- und Leitfragenkatalog (Auszug)

Das in AP 1.2 erarbeitete **Qualitätsmodell** umfasst mit den drei Qualitätsbereichen Prozessqualität, Softwarequalität und Strukturqualität alle Aspekte von Qualität, die aus Sicht des PQ4Agile-Konsortiums im Softwareentwicklungsprozess relevant sind. Die Qualität der entwickelten Produkte ist dabei von zentraler Bedeutung, kann aber nicht isoliert betrachtet werden. Zusätzlich zur Produktqualität umfasst das Qualitätsmodell daher auch die Qualitäten von Entwicklungsprozessen, z. B. um die durch Veränderungen am Entwicklungsprozess verursachten Auswirkungen auf die Prozessqualitäten nachvollziehbar machen zu können. Darüber hinaus haben auch strukturelle Aspekte und deren Qualität, wie die vorhandene technische Infrastruktur oder die verfügbaren personellen Ressourcen, einen Einfluss auf die resultierende Produktqualität. Auch diese wurden folglich in dem Qualitätsmodell berücksichtigt.

In einem konkreten Anwendungskontext können die Bereiche des PQ4Agile-Qualitätsmodells nicht unabhängig voneinander betrachtet werden. Vielmehr sind sie eng miteinander verbunden und bedingen sich gegenseitig. Abbildung 7 illustriert diese Zusammenhänge:

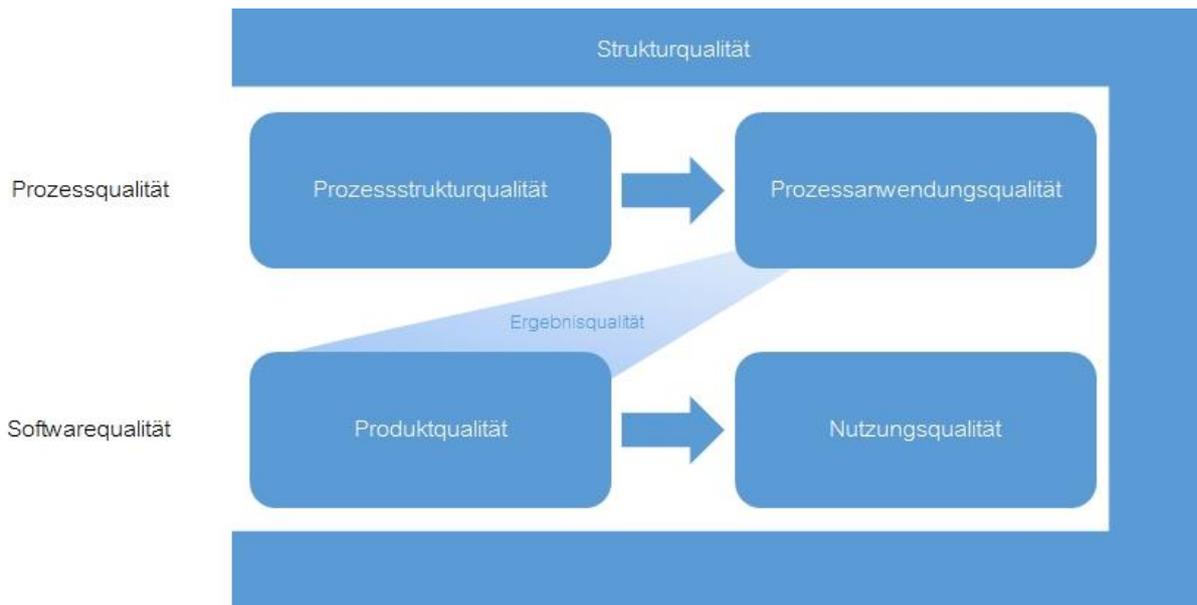


Abbildung 7: Integration der Bereiche des PQ4Agile-Qualitätsmodells

Bei der Prozessqualität kann zwischen Strukturqualität und Anwendungsqualität unterschieden werden. Dabei umfasst die Strukturqualität Attribute, die sich auf den Entwicklungsprozess als solchen beziehen, insbesondere auch auf den Prozess als geplantes Artefakt. Während der Anwendung bzw. Ausführung eines solchen Prozesses sind die Attribute der Anwendungsqualität von Relevanz. Dies bedeutet, dass die Qualitäten des angewendeten Prozesses maßgeblich von den Strukturqualitäten bestimmt sind, aber auch von anderen Faktoren beeinflusst werden können. Während der Ausführung eines Entwicklungsprozesses werden verschiedene Ergebnisse wie beispielsweise Pflichtenhefte, Architekturdokumentationen, Projektsteuerungsartefakte usw. produziert. Diese Ergebnisse des Entwicklungsprozesses besitzen jeweils eine spezifische Qualität. Diese Qualität bezieht sich auf das Attribut Ergebnisqualität der Prozessanwendungsqualität. Handelt es sich hierbei um das Softwareprodukt selbst, so entspricht die Ergebnisqualität der Produktqualität, die im Qualitätsbereich Softwarequalität dargestellt ist. Darin wird analog zum Bereich Prozess zwischen der Qualität des Softwareproduktes als solchem und der Qualität bei der Nutzung des Produkts unterschieden. Die Qualität der strukturellen Gegebenheiten bildet den dritten Bereich des Qualitätsmodells. Ein Entwicklungsprozess wird immer innerhalb einer gegebenen Infrastruktur, mit bestimmten personellen Ressourcen und in einer gegebenen Arbeitsumgebung ausgeführt. Die Strukturqualität kann zudem die Nutzungsqualität eines fertiggestellten (Software-)Produkts beeinflussen, z.B. durch die Qualität von Dienstleistungen, die während des Betriebs einer Software erbracht werden. Die Qualität der gegebenen Struktur beeinflusst also direkt oder indirekt alle zuvor beschriebenen Qualitätsbereiche. Aus diesem Grund ist die Strukturqualität in Abbildung 7 als Rahmen dargestellt, in den die anderen Qualitätsbereiche eingebettet sind.

Das in AP 1.3 erstellte **PQ4Agile-Evaluationskonzept** mitsamt den entsprechenden Werkzeugen kann zur Bewertung der Produkt- bzw. Prozessqualität in beliebigen agilen Entwicklungsvorgehen verwendet werden. Das Evaluationskonzept beruht auf Evaluationsaussagen, die in mehreren Fragebögen für interne Stakeholder (Teamleiter, Entwickler, Kundenansprechpartner) und für die externe Zielgruppe (insbesondere Endnutzer) zusammengestellt sind und die im Rahmen einer Befragung bzw. Datenerhebung bei Softwareherstellern und deren Kunden eingesetzt werden können.

5.2 Arbeitspaket 2: Softwareengineering Best Practices

Laufzeit: August 2014 bis Juli 2015

Lead: Fraunhofer IESE

5.2.1 Ziele des Arbeitspakets

Ziel des Arbeitspakets 2 war es, agile Prozesse durch Softwareengineering-Aktivitäten so zu erweitern, dass eine ganzheitliche Betrachtung von Produktqualitäten möglich wird. Um dies zu erreichen, sollten Softwareengineering-Best-Practices entwickelt werden, die den in Kapitel 3.1 beschriebenen Prinzipien (Minimalismus/Simplizität, Personennähe, Artefaktnähe, Timing, Änderungsaffinität, Skalierbarkeit und Integrierbarkeit) entsprechen und die von Entwicklern selbstbestimmt während der Umsetzung eingesetzt werden können. In Arbeitspaket 2 sollte außerdem die Bedeutung dieser Prin-

zipien für die einzelnen Disziplinen (Requirements Engineering, Usability und Softwarearchitektur) herausgearbeitet werden.

5.2.2 Verwendung der Zuwendung

Arbeitspaket 2 umfasste die folgenden Aktivitäten:

AP 2.1: Beschreibungskonzept Best Practices

In AP 2.1 untersuchten die Partner, wie die Best Practices in geeigneter Form gestaltet und dokumentiert werden können. Hierzu bestimmten die Partner, auf welcher Beschreibungsebene die Best Practices idealerweise dokumentiert werden und wie sie am besten „zugeschnitten“ werden. Im Zuge dessen fand eine Sichtung von Best-Practice-Kandidaten statt (basierend auf einer umfangreichen Literatur- und Internetrecherche nach bekannten Praktiken), welche anschließend nach den Aktivitäten des PQ4Agile-Referenzprozesses geclustert wurden. Zudem generierten die Partner eigene Ideen (basierend auf ihren Praxiserfahrungen) und dokumentierten Erfahrungen und neue Ideen der Community. Unter Leitung des IESE entwickelte das Konsortium ein Beschreibungskonzept (inkl. weiterer Vorgehensweise und Auswahlkriterien), das von HKBS dokumentiert wurde.

Basierend auf einer Recherche existierender Templates wurde vom IESE und HKBS ein generisches Template zur Dokumentation der Best-Practices entwickelt. Dieses Template wurde mit CAS und YellowMap abgestimmt und iterativ angepasst. Hierfür wurden vergleichbare Templates für Methodenbeschreibungen recherchiert, mehrere Vorschläge der Partner zusammengeführt und es wurde überprüft, dass das Template zur Dokumentation von Best Practices aus allen betrachteten Disziplinen geeignet ist.

AP 2.2: Entwicklung und Dokumentation Best Practices

In AP 2.2 erarbeiteten und dokumentierten die Partner gemeinsam die Softwareengineering-Best-Practices. Basierend auf Literaturrecherchen, eigenen Erfahrungen und den Ergebnissen eines Workshops, der auf der Usability Professionals 2014 ausgerichtet wurde, haben das IESE und HKBS zunächst über 300 Best-Practice-Kandidaten gesammelt. Anschließend haben die Partner in einem gemeinsamen Workshop diejenigen Best Practices ausgewählt, die weiter ausgearbeitet werden sollten. Ergänzend hat das IESE Best Practices aus dem Bereich Testen gesammelt und dokumentiert.

In Frage kamen nur Best Practices, die den in Kapitel beschriebenen „Prinzipien für agile Softwareengineering-Best-Practices“ entsprachen. Diese mussten zudem für die Zielgruppe des PQ4Agile-Projekts geeignet sein, also für Mitglieder eines agilen Entwicklungsteams (Generalisten ohne tiefergehendes Expertenwissen in Bereichen RE/UX/Architektur). Bei der Auswahl der Best Practices wurden Praktiken bevorzugt, die sich auszeichneten durch Effizienz, Leichtgewichtigkeit, einfache Erlernbarkeit und frühe Anwendung im Entwicklungsprozess. Die ausgewählten Praktiken sollten außerdem nach Möglichkeit zu einer direkten Erreichung von mehr Produktqualität beitragen. Konstruktive Praktiken wurden analytischen Praktiken vorgezogen.

Die Ausarbeitung der Best Practices wurde unter den Partnern aufgeteilt. Einzelne Methoden aus der Literatur wurden bei Bedarf in Aktivitätsblöcke zerlegt oder für die Anwendung in agilen Entwicklungsprojekten angepasst. Die Erstellung bzw. Dokumentation anhand des in AP 2.1 erstellten Beschreibungstemplates erfolgte ebenso wie die anschließende Veröffentlichung sukzessive. Das IESE führte ein Review sämtlicher erstellten Best-Practice-Beschreibungen durch.

AP 2.3: Werkzeugunterstützung Best Practices

In AP 2.3 untersuchten CAS, HKBS und YellowMap, wie eine geeignete Werkzeugunterstützung bei der Anwendung der Softwareengineering-Best-Practices erfolgen kann. Hierzu wurde analysiert, welche existierenden Werkzeuge für den Einsatz in Frage kommen, welche Bedarfe noch nicht abgedeckt sind und welche Werkzeuge erweitert bzw. neu entwickelt werden können. Bei dieser Untersuchung wurden CAS, HKBS und YellowMap vom Fraunhofer IESE beratend unterstützt.

Bei einer Integration der Best Practices in den Entwicklungsprozess fließen die Ergebnisse einzelner Best Practices bzw. Prozessschritte als Eingabe für nachfolgende Best Practices bzw. Prozessschritte ein. Somit bietet sich auch eine integrierte Werkzeugunterstützung an, bei der die Artefakte innerhalb des Prozesses nicht von einem Werkzeug ins nächste übertragen werden müssen, sondern innerhalb eines Werkzeugs von einem Mitarbeiter zum nächsten weitergereicht werden. Bei der Integration der im Projekt beschriebenen Best Practices in die agilen Entwicklungsprozesse hinterfragten die Partner daher jeweils ihre aktuell eingesetzte Werkzeuginfrastruktur.

Zusätzlich zu den Werkzeugen, die die Partner bereits vor dem PQ4Agile-Projekt zur Unterstützung bestimmter Aktivitäten einsetzten, untersuchten die Partner verschiedene Werkzeuge aus den Bereichen Requirements Engineering, Usability Engineering, Softwarearchitektur und Qualitätssicherung. Die Partner entwickelten Konzepte und Ideen, wie die Best Practices unterstützt werden können, und

tauschten diese aus. Hierbei sollte vermieden werden, dass ein Anwender an ein bestimmtes Produkt eines bestimmten Herstellers gebunden ist; vielmehr sollte er ein eventuell bereits vorhandenes Produkt auswählen können, das die gewünschte Funktionalität bietet und das bei Bedarf gegebenenfalls angepasst oder erweitert werden kann.

5.2.3 Ergebnisse

Für die Dokumentation der in PQ4Agile entwickelten Best Practices wurde in AP 2.1 ein **generisches Beschreibungstemplate** erstellt. Dieses Template enthält folgende Felder:

- Disziplin (RE, UX, Architektur, Testen, unterstützende Praktiken)
- Titel: Bezeichnung der Best Practice (in der Regel Substantiv + Verb)
- Bereich des Referenzprozesses: Einordnung in den Referenzprozess
- Aktivität des Referenzprozesses: Einordnung in den Referenzprozess
- Ziele: Was soll mit Einsatz der Best Practice erreicht werden?
- Motivation/Problemstellung: Was spricht für den Einsatz der Best Practice? Was läuft ohne diese Best Practice potentiell schlechter?
- Kurzbeschreibung: Zusammenfassung der Best Practice zur Einschätzung ihrer Eignung
- Input: benötigte Ergebnisse, Informationen, Artefakte, etc.
- Output: erzeugte Ergebnisse, Informationen, Artefakte, etc.
- Rahmenbedingungen: Ausführender, Vorkenntnisse/Erfahrungen, weitere Teilnehmer, Werkzeuge, Hilfsmittel, Ort/Umgebung, voraussichtliche Dauer
- Vorgehensweise: Vorbereitung, Durchführung, Nachbereitung/Auswertung
- Gütekriterien/Empfehlungen: Welche Faktoren zeigen die erfolgreiche Ausführung der Best Practice an?
- Risiken: Welche Faktoren können das Ergebnis der Best Practice negativ beeinflussen?
- Einordnung in den agilen Referenzprozess: mögliche Vorgänger, mögliche Nachfolger, mögliche Alternativen, verwandte Praktiken
- Einordnung in das PQ4Agile-Qualitätsmodell: Welche (Teil-)Merkmale stehen im Fokus?
- Schlagworte: Verschlagwortung der Best Practice
- Weiterführende Informationen: Informationen im Internet, Literatur

Zusätzlich gibt es drei jeweils dreistufige Skalen, die dem Nutzer eine schnelle Einschätzung erlauben, ob die Best Practice hinsichtlich Aufwand, Schwierigkeit und „Agilitätsfaktor“ für seinen konkreten Anwendungskontext geeignet ist. Durch die Feldauswahl wurde sichergestellt, dass ein Mapping der Best Practices mit dem PQ4Agile-Referenzprozess (Bereich und Aktivität, Vorgänger/Alternativen/Nachfolger) sowie mit dem PQ4Agile-Qualitätsmodell möglich ist. Für die Zuordnung zu den Softwareengineering-Disziplinen wurde ein Farbleitsystem erstellt, das dem Nutzer eine schnelle Orientierung ermöglicht (RE: grün, UX: lila, Architektur: graublau, andere: dunkelgrau).

Das in AP 2.2 erarbeitete **Best-Practice-Kompodium** umfasst insgesamt 42 Best Practices (Stand: April 2016). Dies sind Praktiken aus den Bereichen Requirements Engineering, Softwarearchitektur, Usability/UX und Testen sowie einige unterstützende Praktiken. Die Best-Practice-Beschreibungen enthalten alle grundlegenden Informationen zu den auszuführenden Tätigkeiten, den involvierten Rollen, den zu erstellenden Artefakten sowie zu Möglichkeiten der Auswirkungsanalyse von Entscheidungen und der Einbeziehung von Projektkontextfaktoren. Die Best Practices decken sämtliche Bereiche des PQ4Agile-Referenzprozesses ab (vgl. Abbildung 8) und bieten somit eine umfassende Unterstützung zur Erreichung von mehr Produkt- und Prozessqualität.

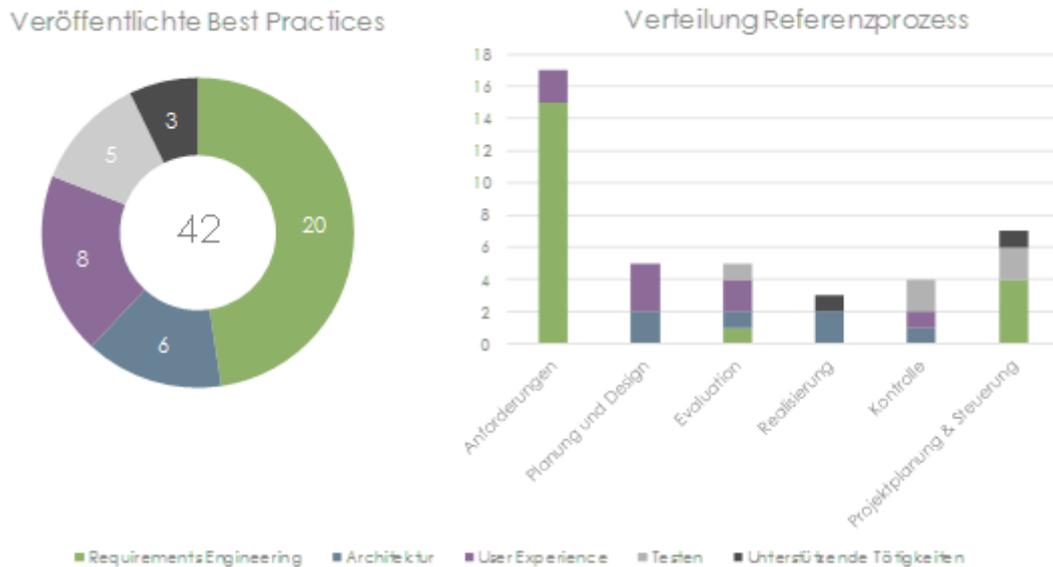


Abbildung 8: Verteilung der Best Practices auf den Referenzprozess

Alle Best Practices stehen der Entwicklergemeinde kostenlos zur Verfügung. Sie wurden sowohl auf der Projektwebsite www.pq4agile.de veröffentlicht als auch in dem Semantic Mediawiki wiki.pq4agile.de, das von YellowMap aufgebaut wurde (s. Abbildung 9).



Abbildung 9: Semantic Mediawiki des PQ4Agile-Vorhabens

Die in AP 2.3 identifizierten, geeigneten **Werkzeuge für die Umsetzung der Best Practices** sowie der Bedarf nach neuen Werkzeugen wurden von den Partnern in einem zentralen Dokument erfasst. Dieses Deliverable enthält eine Matrix, in der pro Partner der aktuelle bzw. geplante Tool-Einsatz für die einzelnen Best Practices erfasst ist.

Liegt eine Auswahl an Best Practices vor und sollen diese innerhalb des Prozesses zum Einsatz kommen, so kann eine geeignete Werkzeugunterstützung gewählt werden. Abhängig von den ausgewählten Best Practices und von der geplanten konkreten Ausgestaltung im jeweiligen Unternehmenskontext kann diese Auswahl unterschiedlich ausfallen. Best Practices können auf sehr unterschiedliche Art und Weise umgesetzt werden. Zum einen sind die Best Practices bereits in ihrer Ausgestal-

tung sehr vielfältig, zum anderen sind die Prozesse individuell auf die Bedürfnisse des jeweiligen Unternehmens zugeschnitten. Es kann daher nicht eine universelle Werkzeugempfehlung zur Unterstützung aller oder bestimmter Best Practices geben. Vielmehr bietet es sich an, bereits vorhandene Werkzeuge innerhalb des Prozesses auf ihre Verwendbarkeit hin zu überprüfen bzw. zu erweitern, falls Funktionalität fehlen sollte. Je nach gewählter Best Practices kann eine Werkzeugunterstützung sinnvoll sein. Bei einigen Best Practices stellten sich einfache Hilfsmittel wie Papier und Stift als Mittel der Wahl heraus. Andere Best Practices konnten wiederum durch rein organisatorische Maßnahmen umgesetzt werden, indem etwa Meetings eingeführt wurden, in denen die entsprechenden Themen besprochen wurden.

5.3 Arbeitspaket 3: Integration Best Practices und Entwicklungsprozess

Laufzeit: November 2014 bis Oktober 2015

Lead: CAS

5.3.1 Ziele des Arbeitspakets

Während in Arbeitspaket 2 die Softwareengineering-Best-Practices als Einzelbauteile erarbeitet werden sollten, zielte Arbeitspaket 3 auf eine Integration der Arbeitsergebnisse ab. Angestrebt wurde hierbei sowohl eine Integration der einzelnen Softwareengineering-Bausteine untereinander als auch eine Integration der Softwareengineering-Aktivitäten in existierende agile Prozesse.

5.3.2 Verwendung der Zuwendung

Arbeitspaket 3 umfasste die folgenden Aktivitäten:

AP 3.1: Best Practices-Integration

In AP 3.1 erarbeiteten die Partner ein Konzept, wie die Softwareengineering-Best-Practices untereinander integriert werden können, und stimmten dieses iterativ ab. Ziel des Konzepts war es, geeignete Schnittstellen zwischen den Best Practices herauszuarbeiten (also z. B. entsprechende Übergabezeitpunkte und -artefakte zu definieren) und Abhängigkeiten sowie potentielle Konflikte zwischen den Aktivitäten zu analysieren. Auch wenn die Best Practices nach Möglichkeit so gestaltet wurden, dass es sich um Einzelbausteine handelt, können diese jedoch nicht vollständig unabhängig voneinander betrachtet werden. Best Practices können sich positiv oder negativ beeinflussen, eine Best Practice kann ein logischer Vorgänger oder Nachfolger einer anderen Best Practice sein oder Best Practices können Alternativen zueinander darstellen.

Zur Erarbeitung des Konzeptes wurden zunächst Fragestellungen identifiziert, die die Integration der Best Practices untereinander betreffen (Welcher Input wird von der Best Practice benötigt? Welcher Output wird generiert? Gibt es komplementäre Aktivitäten? Gibt es konkurrierende Aktivitäten? usw.). Diese Fragestellungen wurden im weiteren Projektverlauf pro Best Practice untersucht. Diese Untersuchung fand jeweils zeitgleich mit der Entwicklung und Dokumentation der Best Practices statt. Alle ermittelten Integrationsaspekte und Informationen wurden in strukturierter Form erfasst.

AP 3.2: Best Practices-Prozess-Integration

In AP 3.2 erfolgte die Integration der Softwareengineering-Best Practices in die Entwicklungsprozesse bei CAS, HKBS und YellowMap. Diese Prozessintegration nahmen die Anwendungspartner im jeweiligen selbst Unternehmen vor; sie wurden hierbei vom IESE unterstützt. Die Softwareentwicklungsprozesse bei CAS, HKBS und YellowMap werden unterschiedlich gelebt, zudem kommen bei der Prozessunterstützung unterschiedliche Werkzeuge zum Einsatz. Jedoch hat sich herausgestellt, dass die Vorgehensweise bei der Einführung der Best Practices bei allen Partnern ähnlich ablief. Zu Beginn stand eine Einführungsveranstaltung, deren Ziel war es, die Projektidee von PQ4Agile vorzustellen und den Entwicklern den generellen Aufbau der Best-Practice-Beschreibungen zu erläutern. Da die Best Practices von Generalisten ohne tiefere Kenntnisse in den relevanten Softwareengineering-Disziplinen angewendet werden können sollten, wurden die Best Practices hierbei bewusst nicht einzeln im Detail vorgestellt. Somit ergab sich eine Bewertung zum gewählten Format der Best Practices-Beschreibungen, insbesondere, ob dieses verständlich ist und somit als Ausgangspunkt für eine Integration geeignet ist.

Anschließend erfolgte die eigenständige Auswahl der zu verwendenden Best Practices in den einzelnen Teams und die Integration der Best Practices mit entsprechender Werkzeugunterstützung in die Prozesse. Den Entwicklern bei HKBS stand während dieser Phase täglich ein zweistündiger Timeslot zur Verfügung, um Best Practices zu testen und anzuwenden. Bei CAS und YellowMap wurden insbesondere Synergien mit dem kontinuierlichen Verbesserungsprozess (Customer-Centricity-Leitbild) genutzt. Bei YellowMap fanden zudem Änderungen an der Prozessgestaltung und dem systematischen Methodeneinsatz statt.

Die Auswahl der Best Practices, welche in den Prozess integriert werden sollten, erfolgte bei CAS, HKBS und YellowMap in regelmäßigen Terminen, die gekoppelt an die (Sprint-)Zyklen des jeweiligen agilen Entwicklungsprozesses stattfanden. An einem solchen Termin wurden auf Basis der Best-Practice-Sammlung einzelne Best Practices ausgewählt bzw. es wurde bewertet, ob diese für den Entwicklungsprozess und das jeweilige Entwicklungsprojekt geeignet sind. Hier dienten zum einen die beschriebenen Ziele und die Kurzbeschreibung der Best Practices, zum anderen die Bewertungen in den drei Kategorien Schnelligkeit, Einfachheit und Agilität als Basis für eine erste Einordnung; eine wichtige Rolle spielte auch der Prozessschritt, der unterstützt werden sollte. Innerhalb der regelmäßigen Termine wurden außerdem erste Bewertungen der Best Practices vorgenommen. War eine Best Practice über einen längeren Zeitraum im Einsatz, so konnte von den Entwicklern auch ihr längerfristiger Nutzen im Vergleich zu den erforderlichen Aufwänden bewertet werden.

Die entwickelten Best Practices standen den Entwicklern zum einen als Dokumentensammlung auf der Projektwebsite zum Download zur Verfügung, zum anderen wurden sie im Laufe des Projekts in ein neu entstandenes Wiki auf Basis der Mediawiki-Erweiterung Semantic Mediawiki integriert. Im Vergleich zu einer reinen Website bietet ein Wiki die Möglichkeit, Wissen semantisch zu strukturieren, indem einfache Textseiten mit semantischen Attributierungen und Relationen angereichert und über Templates vorstrukturiert werden. Auf diese Weise konnte die logische Struktur (wie beispielsweise die Beziehungen zwischen Best Practices und korrespondierenden Merkmalen des Qualitätsmodells), die Beziehung einer Best Practice zu einem bestimmten Schritt des Referenzprozesses oder die Zugehörigkeit zu einer Disziplin (z. B. Requirements Engineering) im Wiki formal abgebildet und so auch für die Suche genutzt werden. Es gibt also beispielsweise die Möglichkeit, solche Best Practices zu identifizieren, die in einem bestimmten Prozessschritt ein bestimmtes Qualitätskriterium verbessern und dabei maximal einen mittleren Aufwand erfordern. Über spezielle Eingabeformulare wird zudem sichergestellt, dass sich neue Best Practices automatisch in das semantische Modell einfügen. Das Ausgabeformat orientiert sich an dem Texttemplate für die Best Practices. Wie ein herkömmliches Wiki verfügt auch Semantic Mediawiki über die Möglichkeiten der Verschlagwortung und Volltextsuche.

Darüber hinaus bietet ein Wiki auch die Möglichkeit, zu den Best Practices jeweils Kommentare zu hinterlegen und somit eine Diskussion über die Erfahrungen zu den Best Practices anzuregen. Hier können Probleme und positive Erfahrungen beim Einsatz der Best Practices erfasst und diskutiert werden. Aufgrund dieser Erfahrungsberichte können die Best Practices ergänzt oder geändert werden. Wird das Wiki offen gestaltet, können auch neue Best Practices vorgeschlagen bzw. erfasst werden und von den Anwendern des Wikis diskutiert werden.

AP 3.3: Werkzeug-Integration

In AP 3.3 integrierten CAS, HKBS und YellowMap geeignete Werkzeuge in ihre eigenen, teilweise umgestalteten Entwicklungsprozesse; hierbei wurden sie vom IESE beraten. Grundlage der Arbeiten waren die in AP 2.3 identifizierten Werkzeuge, die bei Bedarf angepasst wurden. Bei der Umsetzung der Werkzeugunterstützung für die Best Practices hat es sich als sinnvoll herausgestellt, nicht bestimmte Werkzeuge vorzugeben, sondern die Umsetzung anhand der angebotenen Funktionalität vorzunehmen. Dies hat den Vorteil, dass bereits vorhandene Systeme weiterverwendet werden können und dass kein komplett neues System eingeführt und gegebenenfalls parallel zum alten System betrieben werden muss. Zur Verbesserung der Werkzeugunterstützung für die einzelnen Softwareengineering-Aktivitäten kommen aber auch Erweiterungen oder Neuentwicklungen in Betracht, wie sie von den Partnern im Rahmen des Projekts vorgenommen wurden.

5.3.3 Ergebnisse

Die Vorgehensweise bei der Integration der Software-Engineering-Best-Practices untereinander bzw. bei deren Integration in diverse agile Entwicklungsprozesse wurde in Form mehrerer Referenzdokumente beschrieben:

Das in AP 3.1 erarbeitete **Konzept zur Integration der Best Practices untereinander** wurde von HKBS und dem IESE in einem Deliverable dokumentiert. Dieses Konzept sieht vor, dass bei der Entwicklung der Best Practices pro Best Practice alle Punkte, die für die Integration der Best Practices relevant sind, in dafür vorgesehenen Feldern des Best-Practice-Beschreibungstemplates dokumentiert werden. Dies umfasst in erster Linie die Vorgänger- bzw. Nachfolgerbeziehungen zwischen einzelnen Best Practices, die Felder „Input“ und „Output“ (der Output der Best Practice A dient als Input für Best Practice B usw.) und Risiken, die sich z. B. aus der synchronen Anwendung mehrerer Best Practices ergeben können.

CAS erstellte basierend auf den Erfahrungen der Partner in AP 3.2 ein Deliverable, in dem die **Methode zur Integration der Best Practices in agile Prozesse** beschrieben ist. Hier sind neben der Vorgehensweise bei der Auswahl und Einführung der Best Practices pro Partner mehrere Beispiele für

verwendete Best Practices (bzw. die bei der Integration gemachten Erfahrungen) im Detail dargestellt. Insgesamt stellte sich heraus, dass die grundlegende Vorgehensweise bei der Integration der Best Practices in den jeweiligen Entwicklungsprozess bei allen Partnern sehr ähnlich verlaufen ist. Diese bestand im Wesentlichen aus der Auswahl der passenden Best Practices in den Entwicklungsteams, der Integration in den Prozess und der anschließenden Auswahl geeigneter Werkzeuge.

Auf Grundlage der Erfahrungen, die CAS, HKBS und YellowMap in AP 3.3 gesammelt haben, wurde in diesem Deliverable auch die **Methode zur Werkzeug-Integration** beschrieben. Von allen Partnern wurden im Rahmen von PQ4Agile **Eigenentwicklungen bzw. Erweiterungen bereits eingesetzter Tools** vorgenommen. Seitens des IESE wurde etwa ein Werkzeug weiterentwickelt, das zur Unterstützung der Best Practice „Produktphilosophie erstellen“ eingesetzt werden kann. Von CAS wurden JIRA und CAS genesisWorld entsprechend erweitert bzw. angepasst. HKBS implementierte ein web-basiertes Tool, mit dem bestimmte Kennzahlen des Entwicklungsprozesses (z. B. Personalaufwände für bestimmte Aktivitäten) erhoben werden können. Von YellowMap wurde ein strukturiertes Informationsportal aufgebaut und in Form eines Semantic Mediawiki bereitgestellt. Hierfür wurde die Gliederung der Best-Practices in eine semantische Struktur überführt und die Beschreibungen der Gütekriterien aus dem Qualitätsmodell als Referenz eingefügt. Die Struktur des Wikis erlaubt den direkten Zugriff auf Best Practices nach verschiedenen einschränkenden Kriterien, beispielsweise nach dem Bereich des Referenzprozesses, der Softwareengineering-Disziplin oder einem bestimmten Gütekriterium (gemäß PQ4Agile-Qualitätsmodell). Zudem ist eine Filterung nach den Kriterien Aufwand, Schwierigkeit und Agilitätsfaktor möglich.

5.4 Arbeitspaket 4: Evaluation

Laufzeit: August 2014 bis Januar 2016

Lead: YellowMap

5.4.1 Ziele des Arbeitspakets

Das Arbeitspaket 4 befasste sich mit der Evaluation der entwickelten PQ4Agile-Methode durch die Partner; hierzu sollten während des Projektes kontinuierlich Daten erhoben werden. Die Grundlage für alle durchgeführten Evaluationen sollte das Evaluationskonzept bilden, das in AP 1.3 erarbeitet wurde.

5.4.2 Verwendung der Zuwendung

Arbeitspaket 4 umfasste die folgenden Aktivitäten:

AP 4.1: Initiale Prozess- und Produktevaluation

In AP 4.1 erfassten die Anwendungspartner auf Basis der in AP 1.3 entwickelten Evaluationsfragen und -metriken den Ist-Zustand ihrer Prozesse und Produkte. Als Vorarbeit hierfür führten das IESE und HKBS mehrere Reviews der in AP 1.3 entwickelten Fragebögen durch. CAS, HKBS und YellowMap planten anschließend die initiale Evaluation für das eigene Teilprojekt und trafen entsprechende Vorbereitungen. Vom IESE wurde ein excelbasiertes Auswertungsdokument erstellt, mit dem einfache, automatisierte Ergebnisberechnungen durchgeführt werden können. HKBS entwickelte eine datenbankbasierte Webanwendung zur Erhebung relevanter Kennzahlen im eigenen Unternehmenskontext. Anhand des hiermit gesammelten Datenmaterials wurde bei HKBS überprüft, inwieweit die erhobenen Kennzahlen mit den subjektiven Wahrnehmungen der Entwickler übereinstimmen.

Insgesamt nahmen an der initialen Datenerhebung 51 Teilnehmer teil (CAS: 11 Teilnehmer, HKBS: 24 Teilnehmer, YellowMap: 16 Teilnehmer). Bei der Durchführung der Erhebungen und der Erfassung der Daten wurden die Partner vom IESE unterstützt. Das Ergebnis dieser sechsmonatigen Phase bildete die Baseline für die beiden anschließenden Evaluationen.

AP 4.2: Vergleichende Analyse revidierter Prozesse

In AP 4.2 erweiterten CAS, HKBS und YellowMap ihre agilen Entwicklungsprozesse mit den erarbeiteten Softwareengineering-Best-Practices und wendeten diese in (Pilot-)Entwicklungsprojekten an. Über einen Zeitraum von sechs Monaten führten CAS, HKBS und YellowMap regelmäßig unternehmensinterne Workshops mit denjenigen Softwareentwicklern durch, die die Best Practices einsetzten. Als Richtlinien für die Durchführung der Workshops wurden vom IESE Leitfragen zur Erhebung von qualitativem Datenmaterial formuliert, beispielsweise:

- Welche PQ4Agile Best Practices setzen Sie neu innerhalb Ihres Software-Entwicklungsprozesses ein?
- Wie haben Sie die neu verwendeten Best Practices im Detail ausgewählt?
- Welche konkreten Ziele verfolgen Sie mit dem Einsatz der Best Practices?

- Welche Qualitätsmerkmale oder -teilmerkmale des PQ4Agile-Qualitätsmodells möchten Sie durch den Einsatz der Best Practice positiv beeinflussen?

Die Erhebungen wurden von den Anwendungspartnern selbständig und in Eigenregie durchgeführt. Sämtliche Antworten der Softwareentwickler wurden notiert, auch wenn diese zunächst unwichtig oder irrelevant erschienen. Die Durchführung der Workshops über einen Zeitraum von 6 Monaten ließ eine etwaige Veränderung bei der Akzeptanz, der Auswahl, der Integration und dem Einsatz der Best Practices zu.

Abschließend führten CAS, HKBS und YellowMap eine retrospektive Befragung der Softwareentwickler durch, in der die Auswahl, die Integration und der Einsatz der Best Practices nochmals bewertet wurden – frei von den Eindrücken neuer Best Practices und ohne den zeitlichen Druck einer Sprintplanung. Die quantitative Evaluation der Auswirkungen des Best-Practice-Einsatzes auf die Prozessqualität erfolgte durch eine zweite Erhebung mittels der in AP 4.1 verwendeten Fragebögen.

AP 4.3: Vergleichende Analyse Produktqualität

Auch die in AP 4.3 durchgeführte vergleichende Analyse der Produktqualität erfolgte anhand der in AP 4.1 verwendeten Fragebögen. Sofern dies möglich war, erfolgte die Datenerhebung mit denselben Nutzern wie die initiale Produktevaluation. Dies erlaubte es, bei der quantitativen Analyse Rückschlüsse auf wahrgenommene Veränderungen der Qualität zu ziehen, die mit hoher Wahrscheinlichkeit auf die Verwendung der Best Practices zurückzuführen sind.

5.4.3 Ergebnisse

Sämtliche durchgeführten Evaluationen und die hierbei erhobenen empirischen Daten wurden von den Partnern in Form eines **Evaluationsberichts** dokumentiert. In diesem Evaluationsbericht konnten die Partner aufzeigen, dass einzelne Qualitätsmerkmale durch den Einsatz der Best Practices verbessert wurden und dass diese Verbesserungen – beispielsweise ein effizienteres Arbeiten – auch von den Softwareentwicklern wahrgenommen wurden. Die Anwendung der Best Practices wurde aus Sicht der Softwareentwickler für nachhaltig sehr nützlich gehalten. Zudem konnte ein unbewusster Einsatz einzelner Best Practices beobachtet werden, der für die Qualität der Best-Practice-Beschreibungen spricht, insbesondere für deren Verständlichkeit und Anwendbarkeit. Auch die Integration der Best Practices in den Software-Entwicklungsprozess wurde als „nahtlos“ beschrieben. Weiterhin entwickelten die Software-Entwickler durch die Beschäftigung mit den Best Practices einen stärkeren Qualitätsgedanken. So wurden Optimierungspotentiale des bestehenden Software-Entwicklungsprozesses identifiziert und es werden Lösungen für auftretende Probleme erarbeitet.

Eine Varianzanalyse über alle Projektpartner hinweg ergab, dass insbesondere die Prozessanwendungsqualität gesteigert werden konnte. Eine weitere Varianzanalyse gruppiert nach den Projektpartnern ergab, dass einer der Partner die Prozessanwendungsqualität, ein anderer Partner die Produktqualität und der dritte Partner sogar alle betrachteten Qualitätskategorien statistisch signifikant steigern konnte. Außerdem konnten neun Qualitätsmerkmale positiv beeinflusst werden:

- Produktivität,
- Wiederverwendbarkeit,
- Nutzungsflexibilität,
- Interoperabilität,
- Ressourcenverbrauch,
- Zeitverhalten,
- Installierbarkeit,
- Analysierbarkeit und
- Wiederherstellbarkeit.

Hinsichtlich der Eindrücke, die die Produkte von CAS, HKBS und YellowMap bei den Kunden bzw. den Nutzern hervorrufen, war festzustellen, dass nach der Anwendung der Best Practices negative Attribute weniger häufig, weniger stark oder überhaupt nicht mehr wahrgenommen wurden.

5.5 Arbeitspaket 5: Methodenkompetenz

Laufzeit: Februar 2014 bis Januar 2016

Lead: HKBS

5.5.1 Ziele des Arbeitspakets

Das Arbeitspaket 5 befasste sich mit dem Transfer der erarbeiteten Forschungsergebnisse zur Nutzung durch die Öffentlichkeit und die beteiligten Projektpartner.

5.5.2 Verwendung der Zuwendung

Arbeitspaket 5 umfasste die folgenden Aktivitäten:

AP 5.1: Aufbau Projektportal / Social-Media-Kanäle

In AP 5.1 sorgte HKBS für den technischen und inhaltlichen Aufbau der PQ4Agile-Projektwebsite. Das IESE unterstützte den Aufbau mit grafischen Arbeiten und erstellte eine englische Sprachversion der Seiteninhalte. Das Projekt bzw. die Website wurden über Social-Media-Kanäle bekannt gemacht. CAS kümmerte sich um die Einrichtung entsprechender Gruppen bei Xing und LinkedIn. Zur Bekanntheitssteigerung ergriffen die Partner außerdem Online-PR- und -Marketing-Maßnahmen, z. B. wurden mehrere (Online-)Pressemittelungen herausgegeben.

Im Laufe des Projekts sorgte HKBS für Betrieb, Wartung und statistische Auswertung der Website und veröffentlichte kontinuierlich Neuigkeiten aus dem Projekt, insbesondere zu den erzielten Projektergebnissen. Die Website enthält sämtliche veröffentlichten Ergebnisse des Projekts, so dass sich interessierte Entwickler(firmen) und Wissenschaftler gezielt über diese Ergebnisse informieren können. Die Website steht auch nach Projektabschluss weiter als zentrale Wissensbasis für den Transfer der PQ4Agile-Ergebnisse zu Wissenschaft und Wirtschaft zur Verfügung.

AP 5.2: Transferaktivitäten

In AP 5.2 hat das IESE ein PQ4Agile-Logo, ein entsprechendes Farbschema, Dokumenten- und Präsentationstemplates sowie ein Projektposter erstellt. Zu Beginn des Projekts wurde von HKBS erfolgreich das Prädikat „Projekt im Software-Cluster“ für PQ4Agile beantragt.

Im Laufe des Projekts haben HKBS und das IESE an zahlreichen Calls for papers für Konferenzen, Tagungen und Workshops teilgenommen. Dies waren: Agile Bodensee 2014, Berlin DoSE 2014, BITKOM Software Summit 2015, BITKOM AK Quality Management 2015, ECSA 2015, Mensch und Computer / Usability Professionals 2014, Mensch und Computer / Usability Professionals 2015, Ready2Use 2015, REConf 2015, Software Engineering 2015 (Workshop InUSE), Tools4AgileTeams 2014, UIG-Frühjahrstagung 2015, ux congress 2015, UX-Day Mannheim 2015, World Usability Day 2015 Frankfurt. Viele dieser Teilnahmen waren erfolgreich und mündeten in (wissenschaftliche) Veröffentlichungen, die in Kapitel 10 aufgeführt sind. Weitere Veranstaltungen ohne Veröffentlichung waren:

- BITKOM Software Summit (23.9.2014), Vortrag: „Produktqualität in agilen Entwicklungsvorgehen“
- REConf 2015 (16.3.2015), Vortrag: „Agilität & Qualität: Welchen Beitrag können Best Practices aus dem Requirements Engineering leisten?“
- UIG-Frühjahrstagung (15.4.2015), Vortrag: „Agile Softwareentwicklung und Usability“
- UX Day Mannheim (29.10.2015), Workshop: „UX verstehen für agile Entwickler“
- Bitkom – AK Quality Management (4.11.2015), Vortrag: „Mit kleinen Bausteinen zu mehr Qualität: Software Engineering Best Practices für Agile Entwicklung“

HKBS, das IESE und YellowMap stellten das Projekt auf der Fachtagung KMU-innovativ 2014 in Berlin vor. CAS präsentierte das Projekt 2015 im Rahmen des CeBIT-Messeauftritts und im Forschungsbericht des Unternehmens. Vom gesamten Konsortium wurde ein englischsprachiger Fachbeitrag für die Zeitschrift „i-com – Journal of Interactive Media“ verfasst. Von HKBS und dem IESE wurden außerdem zwei Artikel für das Entwickler Magazin verfasst.

AP 5.3: Finalisierung der Projektergebnisse

In AP 5.3 wurden von den Partnern sämtliche Transferprodukte konsolidiert. Dies umfasste die während der Projektlaufzeit erstellten Referenzdokumente und das erstellte Schulungskonzept. Als Grundlage für das Schulungskonzept bzw. die erstellten Schulungsmaterialien dienten die Materialien des Workshops „UX verstehen für agile Entwickler“, den das IESE und HKBS auf dem UX-Day Mannheim (29.10.2015) ausgerichtet hatten. Die Materialien dieses Workshops wurden im Nachgang der Veranstaltung überarbeitet und ergänzt.

5.5.3 Ergebnisse

Abbildung 10 zeigt die zweisprachige **PQ4Agile-Projektwebsite** (www.pq4agile.de). In den Rubriken „Best Practices“ und „Service“ stehen alle veröffentlichten Projektergebnisse kostenlos und diskriminierungsfrei zum Download zur Verfügung.

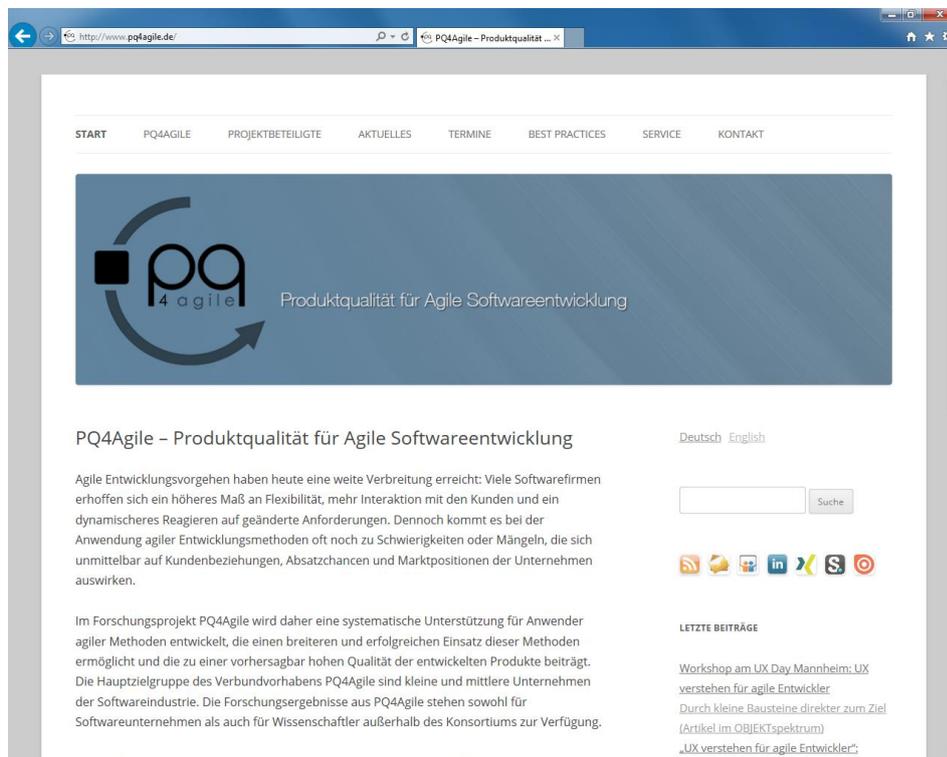


Abbildung 10: PQ4Agile-Projektwebsite

Die in AP 5.3 erstellten **Schulungsmaterialien** umfassen Präsentationsfolien zu insgesamt 10 Lernzielen und Unterlagen für zwei praktische Übungen (vgl. Abbildung 11). Anhand des Schulungskonzepts können interessierte Entwickler oder Firmen vertiefend in der Anwendung des Best-Practice-Kompodiums geschult werden.

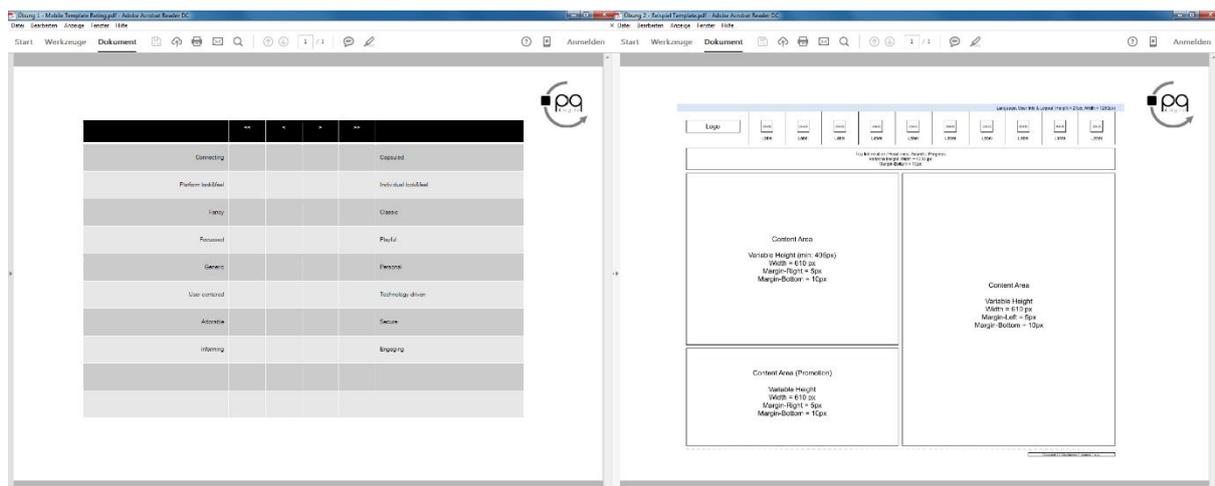


Abbildung 11: Schulungsmaterialien des PQ4Agile-Vorhabens (Beispiele)

Das in Abschnitt 5.3.2 beschriebene **Semantic Mediawiki** (vgl. Abbildung 12) bietet als semantische Wissensplattform die Möglichkeit, die im Projekt entstandenen Ergebnisse so zu erweitern und zu ergänzen, dass sich Erweiterungen in die formale Struktur einfügen.

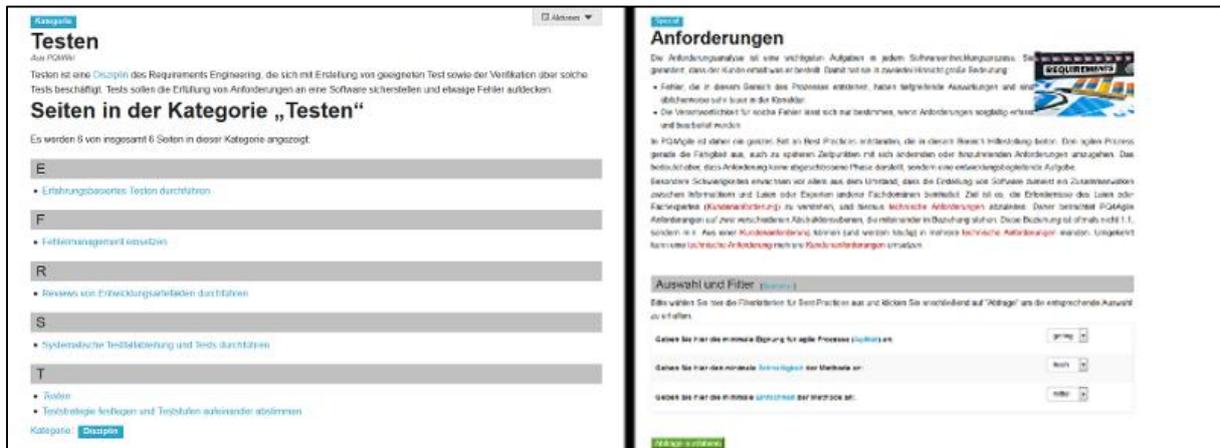


Abbildung 12: PQ4Agile-Wiki – Semantische Suche

5.6 Arbeitspaket 6: Projektkoordination

Laufzeit: Februar 2014 bis Januar 2016

Lead: HKBS

5.6.1 Ziele des Arbeitspakets

Dieses Arbeitspaket umfasste die organisatorischen Tätigkeiten aller Partner, die der Koordination des Verbundprojektes dienen. Als Gesamtprojektverantwortlicher sollte HKBS die Konsortialleitungsaufgaben übernehmen und die Arbeiten der Projektpartner koordinieren.

5.6.2 Verwendung der Zuwendung

Zu Projektbeginn wurden von allen Projektpartnern interne organisatorische Maßnahmen getroffen, die die Projektinitialisierung und den Projektablauf betreffen (Ressourcenplanung, Reporting, Dokumentation usw.). Es wurden die benötigten Infrastrukturen für das Projekt geschaffen und die hierfür erforderlichen Anschaffungen gemacht. Als zentrale Kommunikationsinfrastruktur, insbesondere für den Dokumentenaustausch der Partner, diente ein SVN-Server mit gesicherten persönlichen Zugängen, der vom IESE eingerichtet, administriert und betrieben wurde.

Im weiteren Projektverlauf koordinierte HKBS als Konsortialführer und Projektleiter alle im Projekt zu erbringenden Arbeiten. Projektmanagementaufgaben innerhalb der einzelnen Arbeitspakete lagen außerdem beim Projektpartner, der die Leitung des jeweiligen Arbeitspakets innehatte. Zu Beginn der einzelnen Arbeitspakete wurde der Projektplan jeweils detaillierter ausgearbeitet und es wurden die konkreten Arbeitsergebnisse festgelegt.

Durch regelmäßige Konsortialtreffen aller Projektpartner wurden der Projektfortschritt und die Projektausrichtung kontrolliert, dokumentiert und im Rahmen der Zielsetzungen bedarfsweise angepasst. Darüber hinaus fanden kontinuierlich Arbeitstreffen und Telefonkonferenzen statt, um die laufenden und anstehenden Arbeiten abzustimmen und um den Informationsaustausch zwischen den Partnern zu gewährleisten.

6 Wichtigste Positionen des zahlenmäßigen Nachweises

6.1 HK Business Solutions

Während der Projektdurchführung sind bei HK Business Solutions Personalkosten und Reisekosten entstanden, über die im Verwendungsnachweis im Detail berichtet wurde. Ein geringer Fördermittelbeitrag für Reisekosten wurde von der HK Business Solutions nach Rücksprache mit dem Projektträger für Personalkosten eingesetzt. Der in der Gesamtvorhabenbeschreibung skizzierte Kostenrahmen wurde im Wesentlichen eingehalten. Durch die intensive Nutzung der Best Practices und die Aufbereitung sämtlicher Projektergebnisse in der letzten Projektphase kam es zu einer leichten Überschreitung der veranschlagten Gesamtaufwände.

6.2 CAS Software

Die Aufwandsplanung ist seitens CAS eingehalten worden (ca. 24,5 Personenmonate Gesamtaufwand). Allerdings wurde das Projekt von Mitarbeitern mit höherem Gehalt als das ursprünglich geplante durchgeführt, so dass die Personalkosten die Plankosten um ca. 16% überschreiten. Ein kleines Gegengewicht dazu waren die Reisekosten: aufgrund der räumlichen Nähe der Projektteilnehmer sind weniger Reisekosten angefallen als geplant. Die zusätzlich entstandenen Aufwände sind aus Projektsicht kostenneutral, d.h. sie sind bereits von CAS getragen worden.

6.3 Fraunhofer IESE

Der Aufgaben- und der Zeitplan wurden von Fraunhofer IESE eingehalten. Auch die Kostenplanung wurde von Fraunhofer IESE weitgehend eingehalten, es kam lediglich zu einer leichten Überschreitung der geplanten Gesamtaufwände. Die Überschreitung wurde aus Eigenmitteln des Institutes ausgeglichen. Die Gesamtkosten verteilen sich auf 97,6 Prozent Personalkosten und 2,4 Prozent Reisekosten.

6.4 YellowMap

Seitens der YellowMap AG liegen die genannten Kostenpositionen bis auf Verschiebungen durch die Verlagerung / Verlängerung der Projektlaufzeit und kleinere (kostenneutrale) Verschiebungen zwischen Arbeitspaketen nahezu bei den geplanten Projektkosten. Für die genauen Zahlen wird auf den Verwendungsnachweis verwiesen.

7 Notwendigkeit und Angemessenheit der geleisteten Arbeit

Kleine und mittelständische Hersteller stehen am Softwaremarkt schlagkräftigen Konkurrenten gegenüber, die global agieren und die meist über eigene Forschungsabteilungen verfügen. Um gegen diese Konkurrenz langfristig bestehen zu können, ist es für KMU zwingend notwendig, auf möglichst effiziente Weise qualitativ hochwertige Produkte zu entwickeln. Insbesondere kleine Unternehmen verfügen jedoch nicht über das Expertenwissen und die notwendige Infrastruktur zur Durchführung anspruchsvoller Forschungsprojekte, die zur Entwicklung entsprechender Prozess- und Produktinnovationen notwendig sind.

Das im PQ4Agile-Vorhaben untersuchte Thema der Optimierung agiler Softwareentwicklung ist Teil eines jungen, innovativen und wissenschaftlich anspruchsvollen Forschungsgebiets. Die methodischen Vorarbeiten des Projekts, die Entwicklung der Best Practices, deren Anwendung in Pilotprojekten sowie die Durchführung großangelegter Studien im Rahmen des Projekts erfordern zudem Kenntnisse in diversen Disziplinen des Softwareengineerings. Innovationsorientierte KMU wie HKBS, CAS und YellowMap sind dank ihrer finanziellen und personellen Ausstattung zwar in der Lage, weitreichende Vorleistungen zu erbringen, sie können die erforderlichen Forschungsaufwände jedoch nicht alleine tragen. Die Durchführung des Projekts war daher nur in einer geförderten Zusammenarbeit mit einem Forschungspartner realisierbar, der über die entsprechende Expertise verfügt.

Auch von dem beteiligten Forschungspartner hätte der Forschungsaufwand, der im Rahmen des PQ4Agile-Vorhabens geleistet werden musste, nicht aus Eigenmitteln bestritten werden können. Die praktische Anwendung der Best Practices und die Evaluation der Prozess- und Produktqualität sind an einen konkreten Anwendungskontext gebunden. Um eine einfache Übertragbarkeit der Projektergebnisse auf unterschiedliche Unternehmen und insbesondere eine Integrierbarkeit der Best Practices in beliebige agile Prozesse zu gewährleisten, war deren Anwendung unter verschiedenen Rahmenbedingungen (z. B. hinsichtlich organisatorischem Umfeld und verwendeter Technologien) notwendig. Dies erforderte eine Zusammenarbeit mit mehreren geeigneten KMU-Partnern aus der Softwarebranche.

Daraus ergibt sich zusammenfassend, dass die Projektergebnisse nur in der vorliegenden Konstellation – also in einem Projektkonsortium bestehend aus einem Forschungspartner und mehreren KMU-Partnern aus der Softwareindustrie – und dank der sich hieraus ergebenden Synergien erreicht werden konnten. Die nachhaltige Verbreitung der Ergebnisse, die die kostenlose Nutzung durch andere KMU anregen sollte und dadurch zu einer Stärkung des deutschen Markts insgesamt beitragen sollte, machte ebenfalls eine Zuwendung durch öffentliche Mittel notwendig.

8 Voraussichtlicher Nutzen und Verwertbarkeit der Ergebnisse

8.1 HK Business Solutions

HKBS konnte die in PQ4Agile entwickelten Best Practices bereits zur Projektlaufzeit in einer Vielzahl von Pilotprojekten anwenden, sowohl bei der Weiterentwicklung bzw. Anpassung von Standardprodukten als auch bei der Entwicklung kundenspezifischer Individuallösungen. HKBS hatte die Möglichkeit, geeignete Tools für die Unterstützung der Aktivitäten zu untersuchen, diese bei Bedarf anzupassen und gegebenenfalls weiterzuentwickeln. Im Zuge des PQ4Agile-Projekts konnte bei HKBS ein Prozess etabliert werden, bei dem insbesondere Best Practices aus den Bereichen Requirements Engineering und Usability/UX (z. B. „Usability Patterns verwenden“) mit Erfolg eingesetzt werden. Die PQ4Agile-Methode und der Einsatz der entwickelten Best Practices tragen dadurch bereits heute unmittelbar zu einer Qualitätssteigerung der von HKBS hergestellten Softwarelösungen bei. Dies wurde durch die Ergebnisse der Evaluation bestätigt, wurde aber auch durch die beteiligten Entwickler so eingeschätzt und wahrgenommen.

Der zur Projektlaufzeit erreichte Stand, also die Erprobung und Anwendung ausgewählter Best Practices in Pilotprojekten, soll daher auf andere Bereiche des Entwicklungsprozesses ausgeweitet werden und der Best-Practice-Einsatz soll weiter intensiviert werden. Zusätzliche Best Practices, z. B. aus dem Bereich Testen, sollen sukzessive in den heute gelebten Entwicklungsprozess integriert werden. Diese Integration wird schrittweise innerhalb des nächsten Halbjahres bzw. Jahres erfolgen. Da die Best Practices leicht in den Entwicklungsprozess bei HKBS integriert werden können und da die Entwickler[teams] die Best Practices in ihrem Bereich bzw. im jeweiligen Projektkontext selbstbestimmt einsetzen können, ist auch dauerhaft von einer hohen Akzeptanz der PQ4Agile-Methode im Unternehmen auszugehen.

HKBS geht davon aus, dass das Projekt PQ4Agile hierdurch einen nachhaltigen positiven Effekt auf den Markterfolg der Produkte und auf die Kundenzufriedenheit und Kundenbindung haben wird. Außerdem wird durch die agile PQ4Agile-Vorgehensweise die Entwicklung qualitativ hochwertiger Lösungen beschleunigt, was es erlaubt, Entwicklungsprojekte zügiger abzuschließen und schneller auf die Erfordernisse des Marktes zu reagieren. Durch die Verwendung der Best Practices, die zur PQ4Agile-Projektlaufzeit bereits evaluiert wurden, wird zudem das Entwicklungsrisiko innovativer Lösungen gesenkt.

All diese Faktoren verschaffen HKBS als kleinem Anbieter mit begrenzten Ressourcen einen Know-how-Vorsprung und einen erheblichen Vorteil am Softwaremarkt – sowohl gegenüber vergleichbaren deutschen Mitbewerbern als auch gegenüber größeren, meist ausländischen Herstellern von Standardsoftware-Produkten. Durch die eingehende Analyse des Softwareentwicklungsprozesses sowie vorgelagerter Vertriebsaktivitäten war es HKBS zudem möglich, Schwachstellen in den Unternehmensprozessen aufzudecken und diese bereits teilweise zu beheben (durch den Einsatz von PQ4Agile-Best-Practices, aber auch durch andere Maßnahmen). Hinzu kommt, dass HKBS durch die erfolgreiche Durchführung des PQ4Agile-Vorhabens seine Reputation als innovationsorientiertes Unternehmen stärken konnte und die Erfolgswahrscheinlichkeit bei der Gewinnung neuer Forschungsvorhaben erhöhen konnte.

8.2 CAS Software

Agile Softwareentwicklung wird bei der CAS seit ca. 5 Jahren eingesetzt, um schneller, transparenter, kollaborativer und kundenorientierter sowohl standardisierte als auch individualisierte Softwarelösungen bereitzustellen. Mit den Ergebnissen aus PQ4Agile wird die CAS ihren „Smart Development Process“ verbessern, um Qualitätsinkonsistenzen der aus agilen Entwicklungsarbeiten entstandenen Softwarelösungen abzuschaffen. Diese entstehen derzeit noch häufig durch unterschiedliche Erfahrungen, Know-how und Empfindungen der Entwickler, da agile Softwareentwicklung stark individuell abhängig ist. Durch die Best Practices aus PQ4Agile erwartet CAS verbesserte Werkzeuge und Prozesse für kontinuierlich hohe Qualität bei agilen Softwareentwicklungsprojekten. Die Qualitätskonsistenz ist aus der Perspektive der CAS besonders wichtig für Kundenanpassungen durch CAS-Partner sowie für die zukünftige offene „Platform as a Service“-Strategie mit CAS OPEN.

Durch die Breite der von CAS angebotenen Lösungen sowie die hohe Anzahl verschiedener Stakeholder bei Entwicklungen muss eine konsistent hohe Produktqualität garantiert werden, um maximale Kundenzufriedenheit zu gewährleisten. Dies führt aus der wirtschaftlichen Sicht zu einer signifikanten Verbesserung der Effizienz, da weniger nachgebessert werden muss, und gleichzeitig bedeutet eine konsistent hohe Qualität eine Verbesserung der Reputation des Unternehmens.

Aus der wissenschaftlichen und technischen Perspektive beabsichtigt CAS eine langfristige Integration der PQ4Agile-Best-Practices und -Tools in allen Smart Companies und Units (Abteilungen) der CAS,

so dass die Methoden und Werkzeuge kontinuierlich verbessert werden. Zusätzlich dazu plant CAS die Nutzung der gewonnenen Erfahrungen und die Verbreitung der Ergebnisse im Rahmen der engen Zusammenarbeit mit anderen KMU und Forschungseinrichtungen. Dadurch können Synergien identifiziert, Feedbacks angeregt und Verbesserungen vorgenommen werden. Darüber hinaus können empirische Daten aus der Evaluation veröffentlicht werden, um das Bewusstsein deutscher KMU für Produktqualität und Best Practices in agiler Softwareentwicklung zu steigern. CAS plant außerdem das Vorantreiben der PQ4Agile-Methoden in unterschiedlichen Fachgremien (z. B. BITMi) und unter den Partnern der CAS.

8.3 Fraunhofer IESE

Auch wenn das IESE keine eigene Produktentwicklung betreibt, werden jedoch auch regelmäßig große prototypische Entwicklungsprojekte mit Kunden umgesetzt. Die erarbeiteten Best Practices helfen damit auch dem IESE, Entwicklungsprojekte so durchzuführen, dass von Anfang an ein klarer Fokus auf die Erreichung von Produktqualitäten gelegt wird. Dies wird einen positiven Einfluss auf die Ergebnisse und damit auf die Beziehungen zu Kunden haben.

Neben dem direkten Einsatz im Entwicklungsprozess werden die Projektergebnisse dem Fraunhofer IESE helfen, sein Beratungsportfolio insbesondere für KMU zu erweitern und damit zusätzliche und besser angepasste Dienstleistungen anzubieten. Konkret werden die Requirements Engineering-Methode „SATISFY“, die Softwarearchitektur-Methode „ACES“ (Architecture-centric Engineering Solutions) und die User Experience-Methode „UXelerate“ um spezifische Aspekte agiler Entwicklungsmethoden für kleine und mittelgroße Unternehmen erweitert. Insbesondere weil KMU einen nennenswerten Anteil der Industriekunden des IESE bilden und agile Entwicklungsprozesse bei diesen eine wachsende Verbreitung finden, werden auch Beratungsdienstleistungen zu diesem Thema stetig wichtiger und eine wachsende Anzahl von Unternehmen wird davon profitieren können. Dies entspricht auch der Idee des mit dem Fraunhofer-Modell angestrebten Wissenstransfers zwischen Forschung und Industrie. Hier ist bereits ein kurzfristiger Mehrwert durch die Kenntnisse der Projektteilnehmer, aber auch ein mittelfristiger Mehrwert durch explizite Integrationsarbeiten zu erwarten.

Darüber hinaus bietet das IESE Seminare für Industriekunden an (beispielsweise zum Thema Softwarearchitektur), welche um entsprechende Aspekte erweitert werden können, um damit besser auf die Bedürfnisse von Unternehmen eingehen zu können und damit eine größere Anzahl von Unternehmen dafür zu gewinnen. Hier ist ebenfalls bereits kurz- bis mittelfristig von einem Mehrwert auszugehen.

Auf Basis der Methode und den empirischen Ergebnissen wurden Publikationen und Vorträge erarbeitet, um die Projektergebnisse der Fachöffentlichkeit zu präsentieren. Damit wurde die wissenschaftliche Konkurrenzsituation des IESE gestärkt und künftige Veröffentlichungen zu diesen Themen können den Effekt noch verstärken. Damit wird die die Forschung und Entwicklung in den Bereichen Requirements Engineering, Software-Architektur und User Experience sowie von agiler Softwareentwicklung gefördert und die wissenschaftliche Konkurrenzsituation des IESE gestärkt. Empirische Daten aus Entwicklungsprojekten der Industrie tragen zur praktischen Relevanz der Ergebnisse bei und helfen dem Anspruch des IESE als anwendungsorientiertes Forschungsinstitut gerecht zu werden. Andererseits werden die Ergebnisse aber auch anderen interessierten Organisationen zugänglich gemacht und können für deren Weiterentwicklung genutzt werden. Da bereits zahlreiche Veröffentlichungen erfolgt sind, ist hier von einem unmittelbaren Effekt auszugehen.

Darüber hinaus ist ein Einfluss auf die Lehre zu erwarten. Werden heute agile Entwicklungsmethoden und Softwareengineering-Disziplinen noch als separate Themen dargestellt, kann zukünftig eine zunehmend integrierte Sicht auf diese Themen vermittelt werden. Da das IESE bereits einige Lehrveranstaltungen an Hochschulen durchführt und zahlreiche Kontakte zu Hochschulen besitzt, kann eine solche Sicht effektiv in die Lehre transportiert werden. Das erarbeitete Schulungskonzept bildet dafür die Grundlage. Dies ist voraussichtlich mittelfristig zu erreichen.

Aus Sicht des IESE sind einige Möglichkeiten für Anschlussaktivitäten nach dem Projektabschluss vorstellbar. Ein Beispiel ist das Fortschreiben der grundsätzlichen Ideen für verwandte Ansätze wie DevOps. Auch hier ist eine Unterstützung durch Best Practices aus dem Bereich Software Engineering vorstellbar. Darüber hinaus bietet der Einsatz in Softwareökosystemen einige Herausforderungen für weitere Forschungsaktivitäten des IESE. Softwareökosysteme zeichnen sich durch eine partizipative Entwicklungssituation aus, bei der ein Unternehmen ein offenes Softwareprodukt erstellt, welches durch Ergänzungen von anderen Unternehmen erweitert werden kann und damit Mehrwertdienste für den Kunden oder Nutzer bietet. Solche Entwicklungssituationen haben eine erhöhte Komplexität durch zusätzliche Abhängigkeiten, deren Einfluss auf die Methode erforscht werden kann. Außerdem ist auch die Untersuchung einer Rückführung der angepassten Softwareengineering-Aktivitäten auf konventionelle Entwicklungsvorgehen eine mögliche Anschlussaktivität.

8.4 YellowMap

PQ4Agile sorgt bezüglich der Qualität in agiler Softwareentwicklung einerseits für einen direkten Wissenstransfer aus der Wissenschaft in die Entwicklungsprozesse von KMU und schafft darüber hinaus die Grundlage für einen intensiven Austausch über bewährte Methoden und Werkzeuge von KMU untereinander. Aus Erfahrung von YellowMap zeigt sich, dass gerade in der agilen Softwareentwicklung oft ein Verfahren genutzt wird, dessen Ausgestaltung in erster Linie vom erlernten Wissen der verantwortlichen Entwickler und deren Erfahrung determiniert wird. Durch den Austausch wird der Erfahrungshorizont stark erweitert und zudem eine Methodik entwickelt, sowohl eigene als auch neue Verfahren, Best Practices und Werkzeuge hinsichtlich der Qualitätsauswirkungen zu bewerten und zu vergleichen. An die Stelle einer mehr oder minder pragmatischen Entscheidung für eine Methode tritt somit eine systematische und fundierte Bewertung und bewusste Auswahl. YellowMap erwartet durch die Vermeidung qualitätsverursachter Nacharbeiten eine erhebliche Effizienz- und Qualitätssteigerung in der Entwicklung. Durch die systematische Betrachtung der Qualität im Entwicklungsprozess können somit bessere Endprodukte in gleicher Zeit realisiert werden und neue Kunden gewonnen sowie die Kundenzufriedenheit gesteigert werden.

Durch den intensiven Austausch über die im Projektrahmen erhobenen und evaluierten Verfahrensschritte, Best Practices und Werkzeuge entsteht eine breite Wissensbasis für die betrachteten Disziplinen insgesamt. Hinzu tritt das wissenschaftliche Fundament, welches sich im Wesentlichen aus dem erarbeiteten Qualitätsmodell, dem Evaluationskonzept sowie dem Beschreibungskonzept für Best Practices zusammensetzt. YellowMap geht davon aus, dass damit eine Grundlage für ein umfassendes Qualitätsmonitoring in der agilen Softwareentwicklung entsteht, das auch nichtfunktionale Aspekte umfasst. Daraus ergibt sich für YellowMap die Basis für einen kontinuierlichen Evolutionsprozess, der darin besteht, auch künftig neu auftretende Verfahren, Best Practices und Werkzeuge hinsichtlich ihrer Auswirkungen auf die Produkt- und Entwicklungsqualität zu bewerten. Das trägt maßgeblich zu einer stetigen methodischen Weiterentwicklung in der agilen Softwareentwicklung bei YellowMap bei.

Die Ergebnisse aus PQ4Agile stellen einen beträchtlichen Mehrwert für die agile Softwareentwicklung allgemein dar, der sich sowohl in von YellowMap und anderen durchgeführten Industrieprojekten als auch in allen softwarebezogenen Forschungsaktivitäten niederschlagen wird. Es ist zu erwarten, dass die Entwicklung von Softwareartefakten in der Forschung wie auch in Kundenprojekten von einer frühzeitigen Berücksichtigung nichtfunktionaler Qualitätsanforderungen stark profitiert. Eine interessante Weiterentwicklung liegt sowohl aus Forschungs- als auch aus wirtschaftlicher Sicht in Spezialisierungen des in PQ4Agile zunächst generisch verfolgten Ansatzes auf bestimmte Klassen von Produkten und Projekten. So ist aus Sicht von YellowMap zu erwarten, dass die Relevanz eines bestimmten Qualitätskriteriums und der Beitrag, den ein bestimmtes Verfahren zu dessen Verbesserung leistet, auch von Projekt- und Produkteigenschaften sowie sonstigen Rahmenbedingungen abhängt. PQ4Agile kann in diesem Sinne auch als ein generischer Überbau für spezifische Verfeinerungen gesehen werden, die Gegenstand künftiger Forschungsaktivitäten, als auch KMU-interner qualitätssteigernder Verfahrensevolution sein können.

8.5 Methodentransfer

Bereits im Vorfeld des Projekts wurde ein Verbreitungskonzept entwickelt, das den nachhaltigen wissenschaftlich-technischen Erfolg des Projekts sicherstellen und eine breite Zielgruppe ansprechen sollte: neben softwareproduzierenden Unternehmen außerhalb des Konsortiums waren dies z. B. potentielle Kunden der PQ4Agile-Partner sowie die wissenschaftliche Forschung und Lehre.

Zur Projektlaufzeit wurden die erreichten Forschungsergebnisse einer umfassenden gesamtgesellschaftlichen Verwertung zugeführt. Der Fachöffentlichkeit wurden die Projektergebnisse in Form von Publikationen, Vorträgen und Workshops vorgestellt. Darüber hinaus wurde die allgemeine Öffentlichkeit über die Projektwebsite und mit Hilfe von Social-Media-Kanälen, insbesondere einer eigens eingerichteten Xing-Gruppe, regelmäßig über das Projekt im Allgemeinen und über aktuelle Projektergebnisse informiert. Außerdem wurden die zahlreichen Kontakte aus Forschungsprojekten und aus Partnernetzwerken bzw. Branchenverbänden wie z. B. Software Cluster oder Bitkom genutzt, um die Ergebnisse zu verbreiten. Durch die Veröffentlichung der PQ4Agile-Projektergebnisse stehen diese nicht nur den beteiligten KMU-Partnern zur Verfügung, sondern sie sind auch für andere Softwarehersteller frei zugänglich. Insbesondere die Best-Practice-Sammlung steht damit auch für KMU außerhalb des Konsortiums kostenlos und diskriminierungsfrei zur Verfügung und kann von diesen für die eigene Entwicklung verwendet werden. Alle entsprechenden Aktivitäten, die zur Projektlaufzeit durchgeführt wurden, sind in Kapitel 5.5.2 ausführlich dargelegt.

Die Aktivitäten zur Verbreitung des Projekts werden auch nach Projektabschluss weiter fortgeführt. Beispielsweise können die erarbeiteten Ergebnisse anhand des entwickelten Schulungskonzepts wei-



terhin im Rahmen von Workshops, Seminaren oder Vorlesungen verbreitet werden. Auch die Beratungsdienstleistungen der Partner, die auf Basis des erworbenen Know-hows optimiert werden können, führen dazu, dass eine zunehmende Anzahl von Unternehmen von den Projektergebnissen profitieren kann.

9 Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen

Vom Konsortium sind zur Projektlaufzeit regelmäßig Informationsrecherchen zu den untersuchten Themengebieten durchgeführt worden, insbesondere zur Integration agiler Entwicklungsvorgehen und etablierter oder neuartiger Methoden aus den Bereichen Requirements Engineering, Softwarearchitektur und User Experience. Die Diskussion dieser Themen in der wissenschaftlichen Community wurde von den Konsortialpartnern mitverfolgt bzw. mitgeführt (z. B. im Rahmen wissenschaftlicher Konferenzen) und sie war von großer Bedeutung für das Vorhaben.

Dies war allerdings nicht im Sinne einer Gefährdung zu sehen, insbesondere sind dem Konsortium keine Ergebnisse von dritter Seite bekannt geworden, die die Durchführung der Arbeiten im PQ4Agile-Vorhaben hätten behindern oder in Frage stellen können. Vielmehr stellte die Diskussion eine gewinnbringende Bereicherung des Projekts dar, denn zum einen diente sie als wertvolle Quelle für neue Forschungsansätze und -ideen, zum anderen belegte sie die wissenschaftliche Relevanz der Arbeiten und den Bedarf nach erprobten, praxistauglichen Ergebnissen.

10 Erfolgte und geplante Veröffentlichungen

Neben der Veröffentlichung der Projektergebnisse in diesem Schlussbericht erfolgte eine Veröffentlichung von Teilergebnissen bereits während der Projektlaufzeit in folgenden Publikationen:

- CAS Software: Agil ist nicht genug, auf die Qualität kommt es an. In: CAS Software (Hrsg.): Forschungsbericht 2015, S. 26-27. CAS Software, Karlsruhe
- Steffen Hess, Dominik Rost, Hartmut Schmitt (2014): UX4Agile – Integration von Best Practices des Usability- und User-Experience-Engineering in agile Entwicklungsprozesse. In: German UPA (Hrsg.): Tagungsband Usability Professionals 2014
- Dominik Rost, Balthasar Weitzel, Matthias Naab, Torsten Lenhart: Durch kleine Bausteine direkter zum Ziel: Architektur-Best-Practices für agile Entwicklung. In: SIGS DATACOM GmbH (Hrsg.): OBJEKTSpektrum Ausgabe Agility/2015. SIGS DATACOM GmbH, Troisdorf
- Dominik Rost, Balthasar Weitzel, Matthias Naab, Torsten Lenhart, Hartmut Schmitt: Distilling Best Practices for Agile Development from Architecture Methodology: Experiences from Industrial Application. In: Danny Weyns, Raffaella Mirandola, Ivica Crnkovic (Hrsg.): Software Architecture: 9th European Conference, ECSA 2015, S. 259-267. Springer, Cham
- Hartmut Schmitt, Dominik Pascal Magin, Steffen Hess, Dominik Rost: Doppelt hält besser – Mit UUX-Best-Practices agile Projekte zum Erfolg führen. In: Holger Fischer, Anja Endmann, Malte Krökel (Hrsg.): Mensch und Computer 2015 – Usability Professionals, S. 95-105. De Gruyter, Berlin
- Hartmut Schmitt, Dominik Magin, Andreas Maier, Richard Wacker, Josh Wang: Usability Integration in Agile Development Processes: A Practice-Oriented Best Practice Approach. In: Jürgen Ziegler (Hrsg.): i-com – Journal of Interactive Media, Band 14, Heft 2, S. 161-168. De Gruyter, Berlin
- Hartmut Schmitt, Dominik Pascal Magin, Dominik Rost: PQ4Agile – Steigerung der Produktqualität in agilen Projekten. In: Martin Pielot, Sarah Diefenbach, Niels Henze (Hrsg.): Mensch und Computer 2015 – Tagungsband, S. 363-366. De Gruyter, Berlin
- Hartmut Schmitt, Dominik Rost (2015): Optimierte Agilität – Qualitätssteigerung durch Best Practices. In: Software und Support Media GmbH (Hrsg.): Entwickler Magazin Ausgabe 2015 (2), S. 82-86. Software & Support, Frankfurt am Main
- Hartmut Schmitt, Dominik Rost, Andreas Maier (2016): Messbar mehr Qualität: Optimierte Agilität durch Best Practices. In: Software und Support Media GmbH (Hrsg.): Entwickler Magazin Ausgabe 2016 (3), S. 28-35. Software & Support, Frankfurt am Main

Weiterhin besteht die Möglichkeit die Ergebnisse des PQ4Agile-Vorhabens auf der Projektwebsite (www.pq4agile.de) und im PQ4Agile-Wiki (wiki.pq4agile.de) einzusehen.

11 Anhang: Projektposter

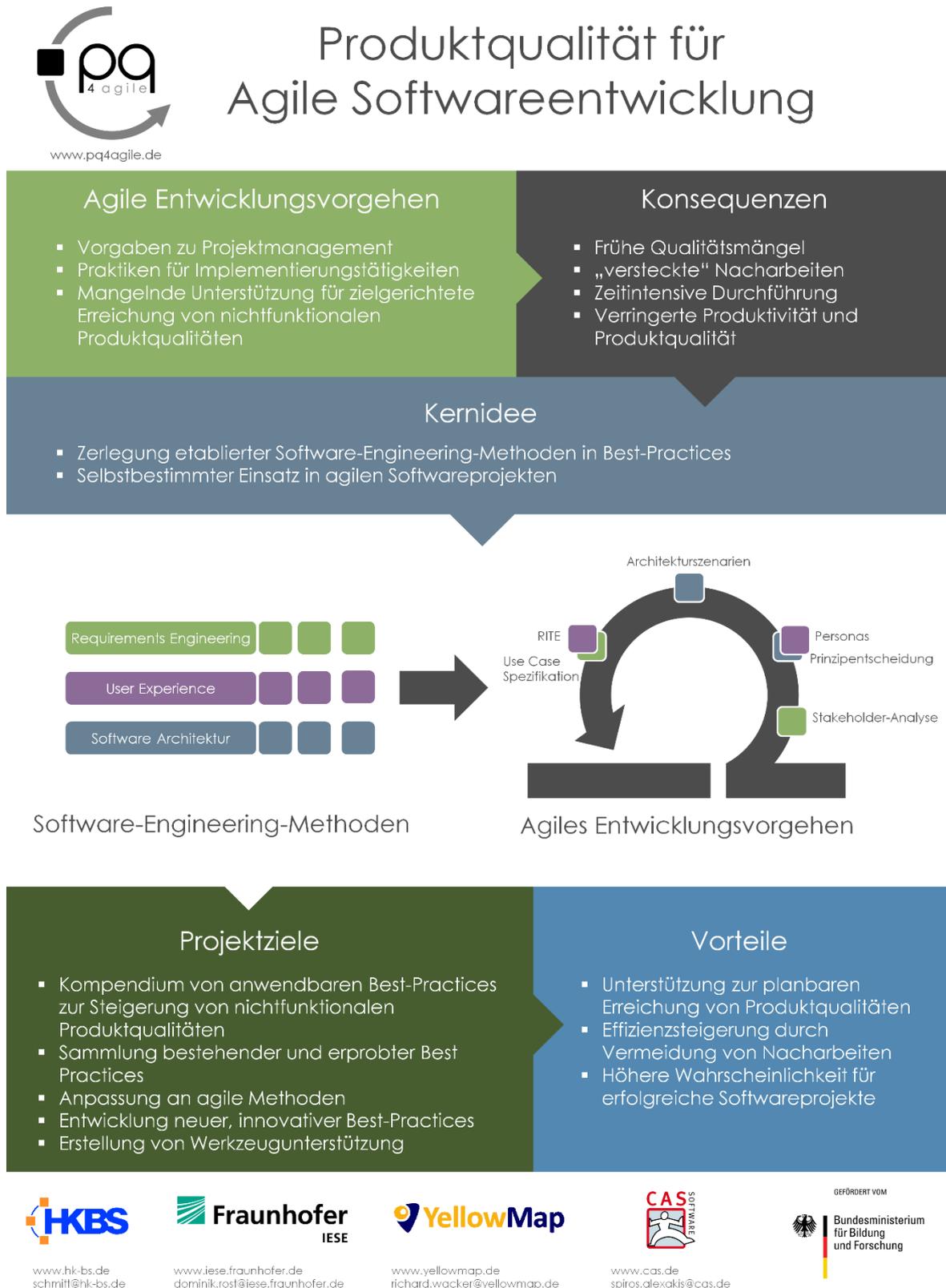


Abbildung 13: Projektposter des PQ4Agile-Vorhabens

12 Abbildungsverzeichnis

Abbildung 1: Lösungsidee des PQ4Agile-Vorhabens	10
Abbildung 2: Forschungsfragen des PQ4Agile-Vorhabens	11
Abbildung 3: Projektplan des PQ4Agile-Vorhabens	13
Abbildung 4: Struktur des PQ4Agile-Qualitätsmodells	27
Abbildung 5: Agiler Referenzprozess des PQ4Agile-Vorhabens	28
Abbildung 6: Forschungs- und Leitfragenkatalog (Auszug)	29
Abbildung 7: Integration der Bereiche des PQ4Agile-Qualitätsmodells	30
Abbildung 8: Verteilung der Best Practices auf den Referenzprozess	33
Abbildung 9: Semantic Mediawiki des PQ4Agile-Vorhabens	33
Abbildung 10: PQ4Agile-Projektwebsite	39
Abbildung 11: Schulungsmaterialien des PQ4Agile-Vorhabens (Beispiele)	39
Abbildung 12: PQ4Agile-Wiki – Semantische Suche	40
Abbildung 13: Projektposter des PQ4Agile-Vorhabens	49



13 Dokumentinformation

Copyright © PQ4Agile-Konsortium, 2015-2016

Alle Rechte vorbehalten. Diese Veröffentlichung darf für kommerzielle Zwecke ohne vorherige schriftliche Erlaubnis des Herausgebers in keiner Weise, auch nicht auszugsweise, insbesondere elektronisch oder mechanisch, als Fotokopie oder als Aufnahme oder sonst wie vervielfältigt, gespeichert oder übertragen werden. Eine schriftliche Genehmigung ist nicht erforderlich für die Vervielfältigung oder Verteilung der Veröffentlichung von bzw. an Personen zu privaten Zwecken.

Titel: Schlussbericht des Vorhabens „Produktqualität für Agile Softwareentwicklung“

Datum: 23.05.2016

Bericht: Schlussbericht gem. NKBF 98 des Projekts „PQ4Agile — Produktqualität für Agile Softwareentwicklung“, BMBF-Förderkennzeichen 01IS13032

Status: final

Klassifikation: öffentlich