

Universität Stuttgart

Lehrstuhl für ABWL und Wirtschaftsinformatik II (Unternehmenssoftware)

Prof. Dr. Georg Herzwurm

Gestaltungs- anforderungen an ein Agiles Requirements Engineering

Sixten Schockert,
GI-Fachgruppentreffen RE,
Kaiserslautern, 23./24.11.17

Agenda

- Kontext & Ziele
- Vorgehen
- Gestaltungsanforderungen gemäß agiler Prinzipien & Werte
- Gestaltungsanforderungen gemäß agiler Entwicklungspraxis/-modelle
- Beispielhafte Anwendung: Agiles Software QFD
- Fazit & Ausblick

Kontext

Vorgehen

Agile Prinzipien

Agile Modelle

Anwendung

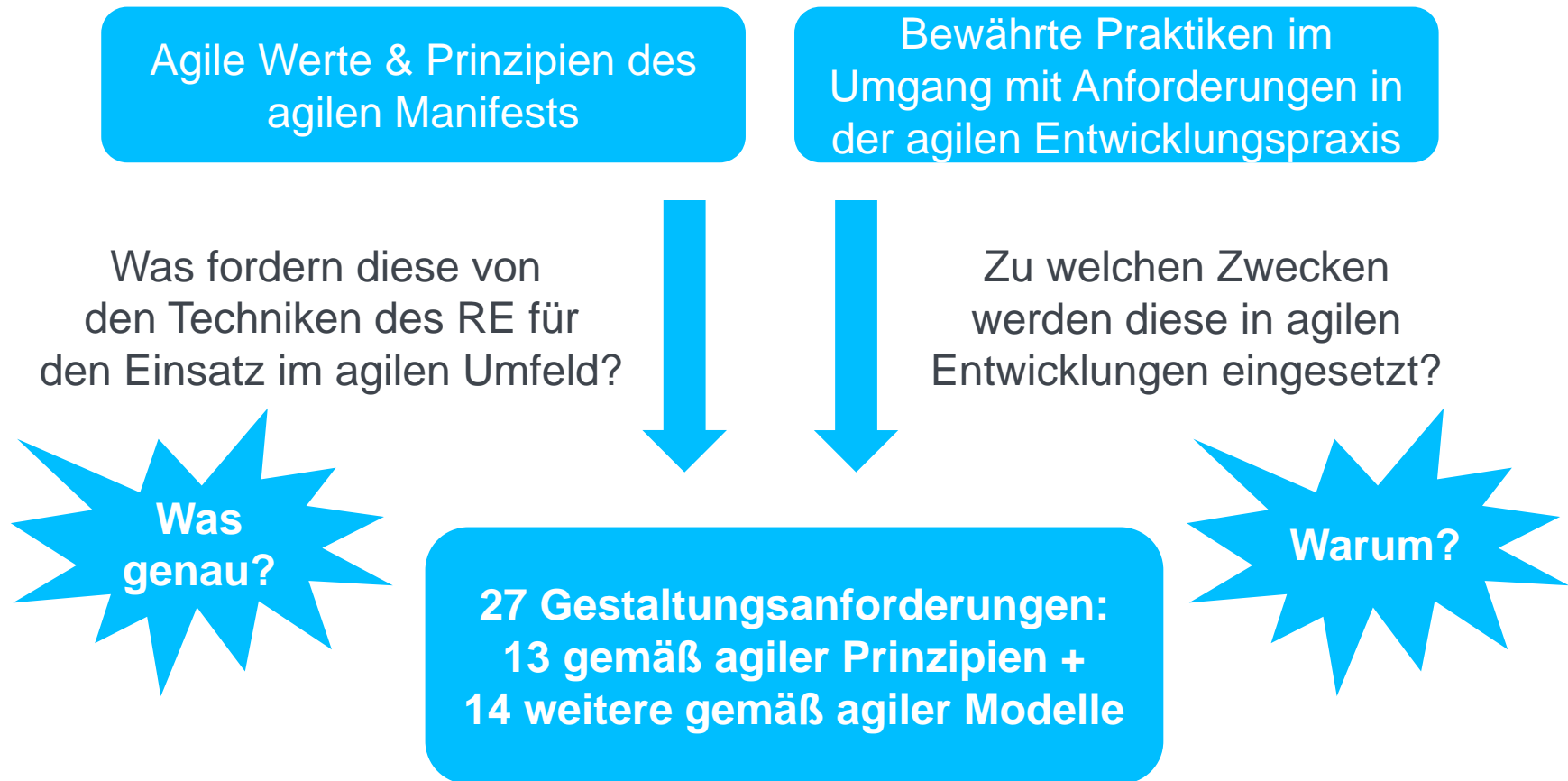
Fazit

Anmerkung: die Quellenangaben zur referenzierten Literatur sowie die detaillierte Darstellung der Inhalte findet sich in Schockert, S. (2017), Agiles Software Quality Function Deployment. Lohmar – Köln, Eul Verlag, Dissertation Universität Stuttgart.

Kontext und Ziele

- Kontext:
Agile Entwicklungsmodelle sind Sammlungen von je nach Situation einzusetzenden Best Practices
- Konkretes Ziel:
Entwicklung, Anpassung und Bewertung einer Methode zum Agilen Software Quality Function Deployment (Agiles SW-QFD)
- ➔ Nötig: Gestaltungsanforderungen an ein Agiles SW-QFD bzw. Agiles RE
- ➔ In welcher Richtung müssen Techniken des RE angepasst werden, um effizient im agilen Umfeld eingesetzt werden zu können?
- ➔ **Allgemeiner:**
Welche Gestaltungsanforderungen müssen Vorschläge für ein Agiles Requirements Engineering erfüllen?

Vorgehen



12 Prinzipien des Agilen Manifests

1	Unsere höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller („valuable“) Software zufrieden zu stellen.
2	Heiße Anforderungsänderungen selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden.
3	Liefere funktionierende Software („working software“) regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne.
4	Fachexperten („business people“) und Entwickler müssen während des Projektes täglich zusammenarbeiten .
5	Errichte Projekte rund um motivierte Individuen . Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen („trust them to get the job done“).
6	Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteams zu übermitteln, ist im Gespräch von Angesicht zu Angesicht („face-to-face conversation“).
7	Funktionierende Software ist das wichtigste Fortschrittsmaß („measure of progress“).
8	Agile Prozesse fördern nachhaltige („sustainable“) Entwicklung. Die Auftraggeber („sponsors“), Entwickler und Benutzer sollten ein gleichmäßiges Tempo („constant pace“) auf unbegrenzte Zeit halten können.
9	Ständiges Augenmerk („continuous attention“) auf technische Exzellenz und gutes Design fördert Agilität.
10	Einfachheit („simplicity“) – die Kunst, die Menge nicht getaner Arbeit zu maximieren – ist essenziell.
11	Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams .
12	In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann und passt („tunes and adjust“) sein Verhalten entsprechend an.

Anforderungen an Agiles RE gemäß agiler Werte/Prinzipien

Gestaltungsanforderungen an Agiles RE (A1-13)		Prinzipien/ Werte
A1	Ausrichtung am Business Value der Anforderungen	1, 2, 3, 7, 10
A2	Vorgehen in kurzen, regelmäßigen Iterationen unterstützen	1, 3, 8
A3	RE-Artefakte können inkrementell wachsen und detailliert werden	1
A4	Feedback zu RE-Artefakten einholen	2, Feedback
A5	Änderungen der RE-Artefakte schnell und einfach (wenig aufwendig) durchführen	2, Feedback
A6	Zusammenarbeit von Fachseite (Experte, Kunde, User) und Entwicklern unterstützen	4, 12
A7	Nachhaltig motivierendes Vorgehen	5
A8	Glaubwürdiges Vorgehen und Ergebnis liefern	5
A9	Direkte persönliche Kommunikation unterstützen	6, Communication
A10	Qualitätsmerkmale und Design-Einschränkungen berücksichtigen	9, Openness
A11	Fokussierung auf die wesentlichen Arbeiten	10, Focus, Simplicity
A12	Selbstorganisation im Team erleichtern	11
A13	Unterstützung der Verpflichtung (Commitment) auf gemeinsame Ziele	Commitment

Beispiel Prinzip 5:

Errichte Projekte rund um **motivierte Individuen**. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und **vertraue darauf, dass sie die Aufgabe erledigen** („trust them to get the job done“)

- Motivation
 - ➔ RE-Techniken sollten „Spaß“ machen, als dass das Team motiviert ist und bleibt, diese auch anzuwenden
 - ➔ A6: Nachhaltig motivierendes Vorgehen
- Vertrauen in die Leistung
 - ➔ Keine direkte Anforderungen an RE-Technik
 - ➔ Anwendung einer RE-Technik kann z. B. durch ein gut nachvollziehbares oder reproduzierbares Vorgehen „vertrauenswürdig“ sein, als dass den Ergebnissen „vertraut“ wird, sie also nicht in Frage gestellt werden.
 - ➔ A7: Glaubwürdiges Vorgehen und Ergebnis liefern

Agile Praktiken, aber...

...RE-Praktiken?

...agiles RE?

- Product Management**
- Blacklog grooming
 - Personas
 - Story mapping
 - Story splitting
 - User stories
 - 3C's
 - INVEST
 - Incremental development

- Design**
- CRC cards
 - Quick design session
 - Rules of simplicity
 - Refactoring
 - Simple design
 - Ubiquitous Language

- Testing**
- Role-Feature
 - Given-When-Then
 - Behaviour-Driven Development (BDD)
 - Acceptance Test Driven Development (ATDD)
 - Acceptance tests
 - Mock objects
 - Test-Driven Development (TDD)
 - Unit Tests
 - Exploratory Testing
 - Usability Testing

- Teams**
- Team
 - Facilitation
 - Heartbeat retrospective
 - Team room
 - Sustainable Pace
 - Niko-niko
 - Scrum of Scrums
 - Project charters
 - Pair Programming

- Scrum**
- Iterative Development
 - Timebox
 - Iterations
 - Daily meeting
 - Three Questions
 - Burndown chart
 - Task board
 - Definition of Done
 - Definition of Ready
 - Point estimates
 - Relative estimation
 - Planning poker
 - Backlog
 - Backlog grooming

- Lean**
- Lead time
 - Kanban board
 - Definition of Done

- Development & Operations**
- Continuous Deployment
 - Continuous Integration
 - Automated build
 - Version control

- Extreme Programming**
- Sustainable Pace
 - Pair Programming
 - Sign up
 - Daily meeting
 - Iterations
 - Velocity
 - Frequent Releases
 - User stories
 - Collective Ownership
 - Continuous Integration
 - Simple design
 - Refactoring
 - Test-Driven Development

Eigene Darstellung in Anlehnung an Agile Alliance (2016c).

Vgl. zu Kurzerläuterungen der Praktiken Agile Alliance (2016b).

- Fundamentals**
- Team
 - Incremental Development
 - Iterative Development
 - Version control

Verbreitete agile Praktiken relevant für das RE

- Verwendung von **User Stories** unterschiedlicher Detaillierungsgrade zur Diskussion im Team und mit den Stakeholdern als wesentliche Elemente des priorisierten **Product Backlogs** (\approx priorisierte Liste von Anforderungen)
- **Personas** zur Repräsentation der Stakeholder(-gruppen) bzw. aufgabenorientierter Benutzerrollen
- **Story Mapping** zur visuellen Anordnung der User Stories unterschiedlicher Detaillierungsgrade im Backlog und Zuordnung zu Outcomes als dem Business Value eines Inkrements
- **Bewertung der User Stories** insb. hinsichtlich Aufwand bzw. Umfang mittels Story Points, Planning Poker o.ä. sowie weiterer Bewertungen wie Risiko oder Abhängigkeiten
- **Priorisierung der User Stories** mittels Ranggruppenbildung (z. B. MoSCoW-Rules) oder Ranking
- **Sprint Backlog** mit den für ein Inkrement ausgewählten Backlog-Einträgen sowie Sprint-Aufgaben einschließlich Personenzuordnungen und Schätzungen
- **Kano-Analyse** der Produkthanforderungen insb. zur Beurteilung wie einzelne Funktionalitäten im Produkt wahrgenommen werden

Erweiterte Bestandteile agiler Modelle mit Fokus auf RE

(unter Verwendung von Scrum-Terminologie)



Rollen	Meetings	Artefakte
Product Owner	Sprint Planning I: Anforderungen für das Inkrement festlegen	Product Backlog Sprint Backlog
Scrum Master	Sprint Planning II: Design und Planung der Umsetzung	Inkrement Sprint Goal, Outcome
Entwicklungsteam	Daily Scrum	Sprint-Aufgaben im Taskboard
Manager der Entwicklungsorganisation	Sprint Review	Produkt Backlog Einträge, i. d. R. User Stories inkl. ihrer Bewertungen
Kunden bzw. Käufer	Product Backlog Pflege – Grooming (Refinement)	Akzeptanztests der Backlog Einträge
User	Sprint Retrospektive	Ggf. Story Maps, Personas

Anforderungen an Agiles RE gemäß agiler Entwicklungsmodelle (1)

RE-Artefakt: (Einträge im) Product Backlog

Gestaltungsanforderungen an das Agile RE (A14-19)

- | | |
|------------|---|
| A14 | Eindeutige Ordnung (Priorisierung) des Product Backlog gemäß Business Value (konkretisiert A1) |
| A15 | Bewertung der Product Backlog Einträge hinsichtlich Umfang (Aufwand der Umsetzung) |
| A16 | Bewertung des Product Backlog Einträge hinsichtlich weiterer Kriterien wie z. B. Risiko, Kano-Kategorisierung, Verzögerungskosten (cost of delay) |
| A17 | Abhängigkeiten zwischen Product Backlog Einträgen abbilden |
| A18 | Anpassung der Priorisierung und Bewertungen des Product Backlog in jeder Iteration (konkretisiert A5) |
| A19 | Unterstützung der Auswahl der Product Backlog Einträge für das nächste Inkrement (den nächsten Sprint) |

Quelle: Schockert (2017)

Anforderungen an Agiles RE gemäß agiler Entwicklungsmodelle (2)

RE-Artefakte: User Stories (insb. A20-A23), Personas (A22), Story Maps (A19, A24-A26), Sprint Backlog (A19, A27)

Gestaltungsanforderungen an das Agile RE (A20-27)

A20	Problem- und Lösungsraum d. h. Stakeholderbedürfnisse und Produktanforderungen berücksichtigen
A21	Verbindung von Problem- und Lösungsraum d. h. Verbindung von Stakeholderbedürfnissen und Produktanforderungen
A22	Berücksichtigung unterschiedlicher Stakeholderperspektiven bei Anforderungen
A23	Diskussion über Anforderungen anregen
A24	Unterstützung der Ableitung des Sprint-Ziels
A25	Unterstützung bei der Ableitung von Akzeptanztests für Product Backlog Einträge
A26	Umsetzen der Anforderungen aus dem Product Backlog in Sprint-Aufgaben
A27	Kontrolle der Umsetzung der Sprint-Aufgaben

Beispiel User Stories

Von einem Stakeholder
geforderte Funktionalitäten

Lösung

Bedürfnis

„Als <Rolle> möchte ich <Funktion>, um <Nutzen/Vorteil> zu erreichen“

(Cohn (2004), S. 81, vgl. auch Leffingwell (2011), S. 342)

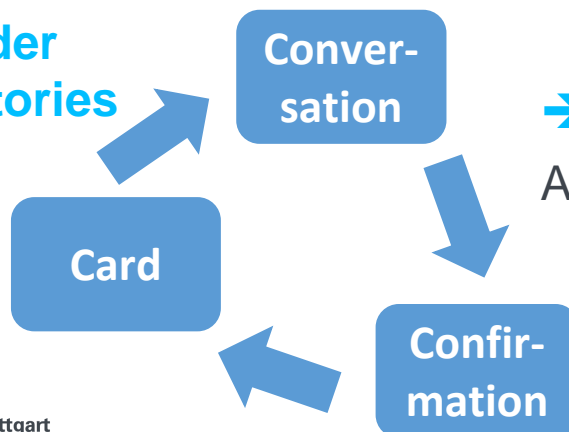
= „Als (Stakeholder) möchte ich (Produktfunktion), um (Bedürfnis) zu befriedigen.“

= Oder abstrakter: „Als (WER) möchte ich (WAS), um (WARUM) zu befriedigen.“

→ Problem- und Lösungsraum d. h. Bedürfnisse und Anforderungen berücksichtigen (A20) und verbinden (A21)

→ A22: Berücksichtigung unterschiedlicher Stakeholderperspektiven

**3Cs der
User Stories**

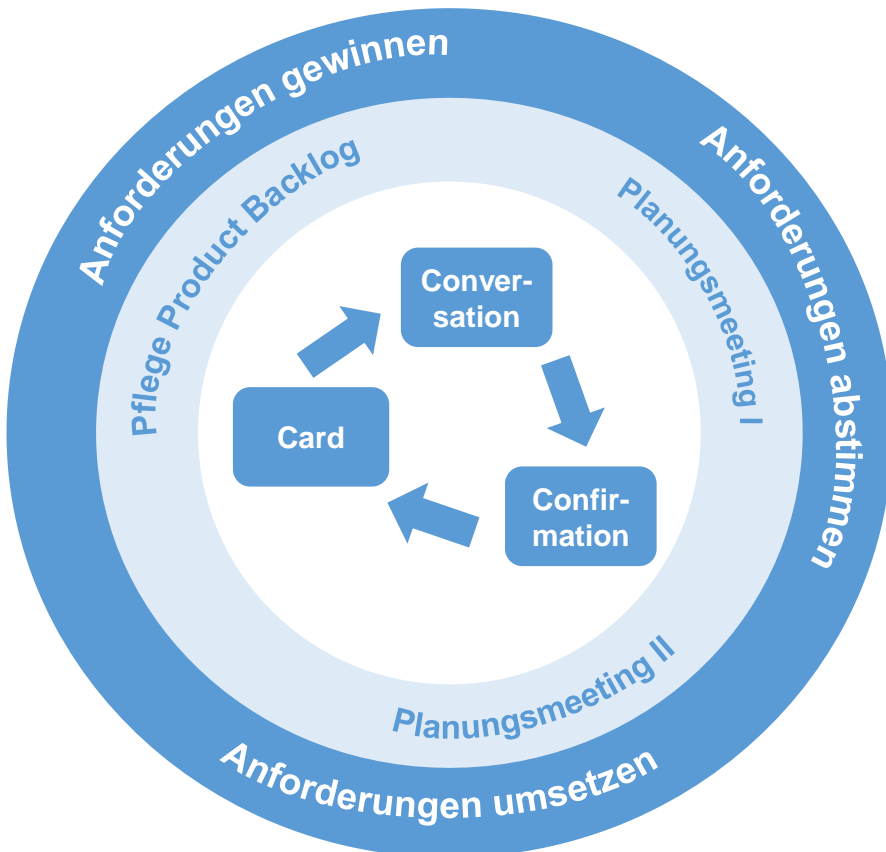


→ A23: Diskussion über Anforderungen anregen

Anwendungsbeispiel: Agiles Software QFD

Einbettung von Software QFD in agilen Iterationszyklus

QFD-Techniken zur Ausrichtung am Business Value

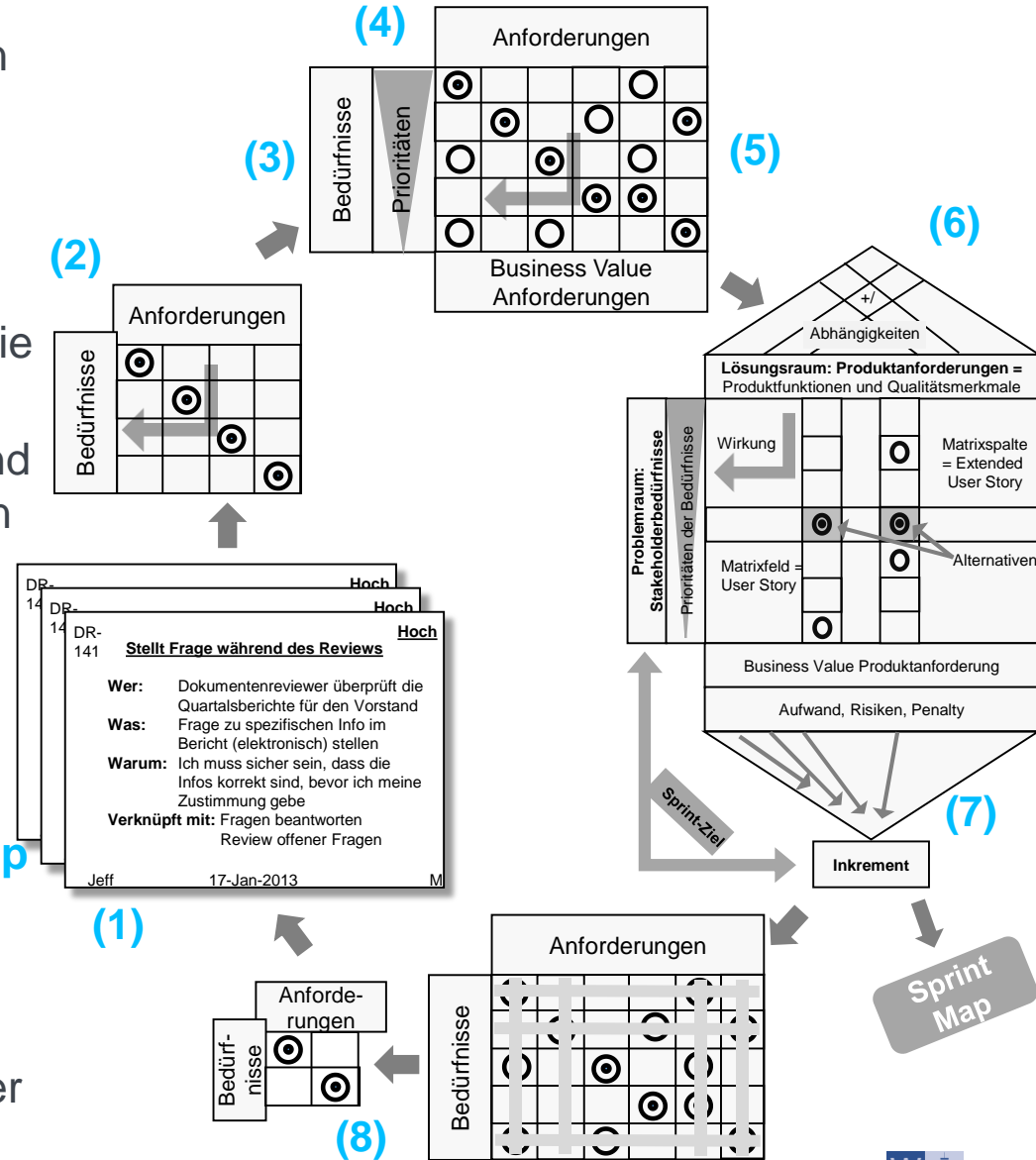


- Verwendung von **(Erweiterten) User Stories**
- **Aufspaltung der User Stories in Bedürfnisse und Produktanforderungen**
- **Priorisierung der Bedürfnisse** durch die Stakeholder
- **Verknüpfung von Bedürfnissen und Produktanforderungen in**
 - inkrementell wachsende Priorisierungsmatrix
 - Relationendiagramm
- **Bewertung von Produktanforderungen**
- **Priority Map** zur kompakten Darstellung der Priorisierung und Bewertungen
- **Sprint Map** für das nächste Inkrement als Extrakt der Priority Map

Quelle: Schockert (2017)

Iterationszyklus des Agilen Software QFD

- (1) **Sammlung von User Stories** mit den Stakeholdern
- (2) **Spaltung** der User Stories in Bedürfnisse und Anforderungen (entsprechend in einer Diagonalmatrix)
- (3) **Bewertung der Bedürfnisse** durch die Stakeholder
- (4) Aufdecken **alternativer Lösungen** und Synergien der Erfüllung von Bedürfnissen
- (5) **Verbindung** der Bedürfnisse und Anforderungen in der **inkrementell wachsenden Priorisierungsmatrix**
- (6) Weitergehende Bewertung der Anforderungen und Darstellung der Extended User Stories in der **Priority Map**
- (7) Auswahl der User Stories für das nächste Inkrement zur Erreichung des Sprint-Ziels in der **Sprint Map**
- (8) **Reduzierung der Matrix** auf Basis der realisierten und voll erfüllten Bedürfnisse



Anwendungsbeispiel: Bewertung anhand der Gestaltungsforderungen an Agiles RE

Agile Software QFD	Gestaltungsforderungen an den Umgang mit Anforderungen im agilen Umfeld	Agile RE practice	Scrum & UX
+ / ++	A1: Ausrichtung am Business Value der Anforderungen	O	+
++	A6: Zusammenarbeit von Fachseite (Experte, Kunde, User) und Entwicklern unterstützen	O / +	+
O / +	A7: Nachhaltig motivierendes Vorgehen	+	+
+ / ++	A8: Glaubwürdiges Vorgehen und Ergebnis liefern	O / +	+
+ / ++	A10: Qualitätsmerkmale und Design-Einschränkungen berücksichtigen	O / +	O / +
O / +	A12: Selbstorganisation im Team erleichtern	+ / ++	+
++	A14: Eindeutige Ordnung (Priorisierung) des Product Backlog gemäß Business Value (konkretisiert A1)	O	O
+	A17: Abhängigkeiten zwischen Product Backlog Einträgen abbilden	O	O
++	A20: Problem- und Lösungsraum d. h. Stakeholderbedürfnisse und Produkthanforderungen berücksichtigen	O	+
++	A21: Verbindung von Problem- und Lösungsraum d. h. Verbindung von Stakeholderbedürfnissen und Produkthanforderungen	O	O
O	A26: Umsetzen der Anforderungen aus dem Product Backlog in Sprint-Aufgaben	O / +	O / +
O	A27: Kontrolle der Umsetzung der Sprint-Aufgaben	+	+

Ausblick & Fazit

- Anwendung der Gestaltungsanforderungen:
Vorschläge für ein agiles RE entwickeln, dagegen zu evaluieren und vergleichend bewerten
- Fortlaufende Anpassung (Evolution) der Gestaltungsanforderungen mit der Etablierung neuer Vorschläge zum agilen RE
- Empirische Evaluierung und Validierung
- Techniken anderer Domänen wie Innovationstechniken etc. mit den Gestaltungsanforderungen an agiles RE konfrontieren
- Übertragbarkeit auf Nicht-Software-Domänen?
- Etablierung von einfachen, handhabbaren, flexiblen Handlungsanweisungen für Agiles RE („generative rules“)?

Anmerkung: die Quellenangaben zur referenzierten Literatur sowie die detaillierte Darstellung der Inhalte findet sich in Schockert, S. (2017), Agiles Software Quality Function Deployment. Lohmar – Köln, Eul Verlag, Dissertation Universität Stuttgart.



Universität Stuttgart

Vielen Dank!



Sixten Schockert

E-Mail schockert@wius.bwi.uni-stuttgart.de

Telefon +49 (0) 711 685-82387

Fax +49 (0) 711 685-82388

Universität Stuttgart

Abteilung VIII, Lehrstuhl für ABWL und Wirtschaftsinformatik II
(Unternehmenssoftware), Prof. Dr. Georg Herzwurm

Keplerstraße 17

70174 Stuttgart

www.wius.bwi.uni-stuttgart.de