



Universität Stuttgart

Lehrstuhl für ABWL und Wirtschaftsinformatik II (Unternehmenssoftware)

Prof. Dr. Georg Herzworm



Agile Softwareentwicklung und QFD

Dr. Andreas Helferich
Sixten Schockert



highQ



 highQ



Agenda

- Grundlagen agiler Softwareentwicklung
- Agiles Requirements Engineering:
Umgang mit Anforderungen in agiler Entwicklung
- Agiles Software QFD
- Agile Softwareentwicklung im Praxisprojekt
- Fazit

Basics

Agile RE

Agile SQFD

Practice

Conclusion

Agile Softwareentwicklung

- **Gegenbewegung** zu den traditionellen, sequentiellen Prozessmodellen
- Reaktion auf zunehmende **Dynamik der Anforderungsänderungen** → Iterative Entwicklung, inkrementell wachsendes Produkt mit frühzeitiger und regelmäßiger Einholung und Verarbeitung von Feedback der Stakeholder
- Typische Vertreter: **Scrum**, Extreme Programming (XP), Kanban
- Kernideen und Ursprung: **Agiles Manifest** (www.agilemanifesto.org, 2001)

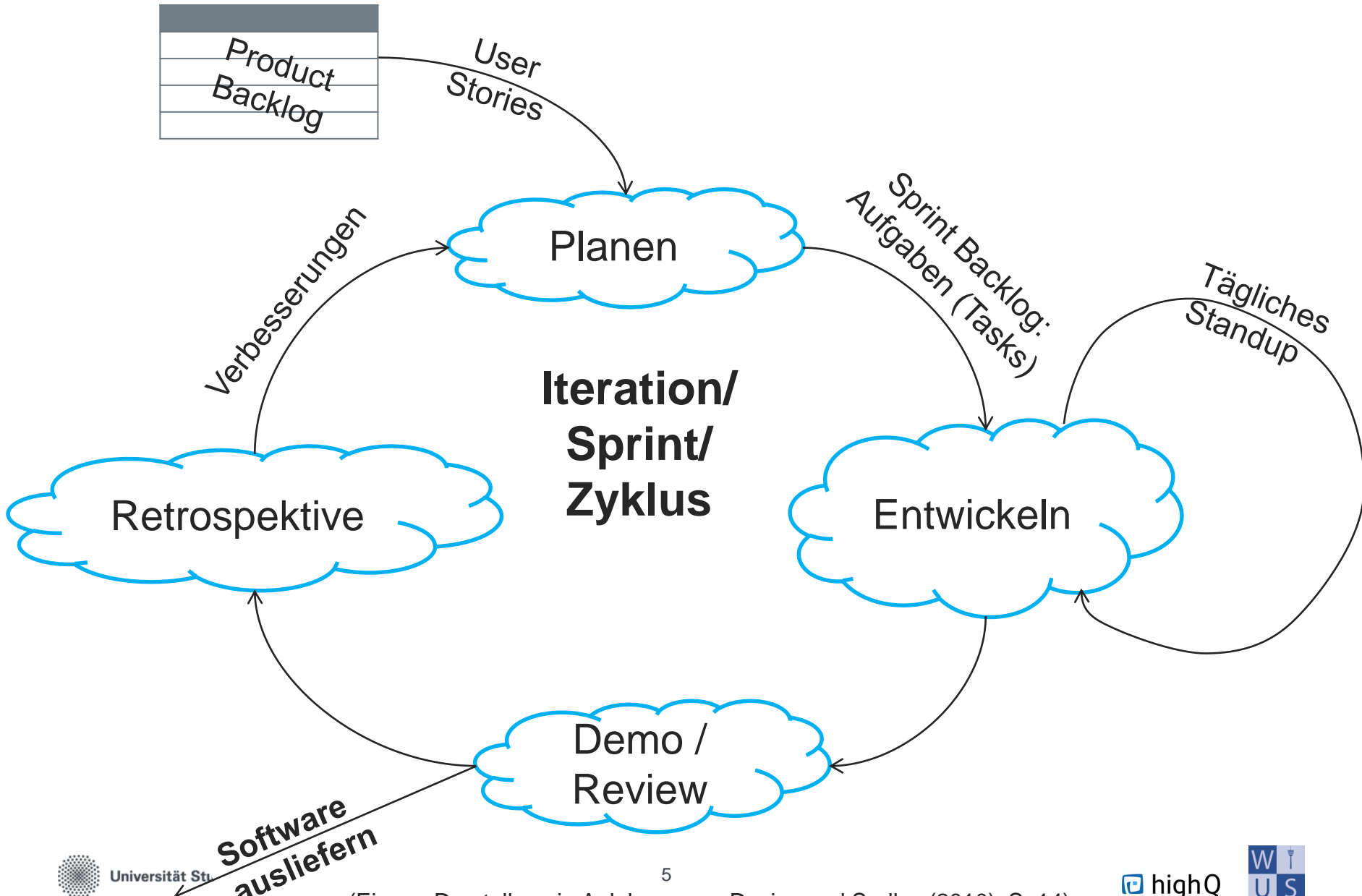
Leitsätze des agilen Manifests		
Höher geschätzte Werte		Weniger geschätzte Werte
Individuen und Interaktionen	sind / ist wichtiger als	Prozesse und Werkzeuge.
Funktionierende Software		umfassende Dokumentation.
Zusammenarbeit mit dem Kunden		Vertragsverhandlung.
Reagieren auf Veränderung		das Befolgen eines Plans.

Quelle: In Anlehnung an Beck et al. (2001).

12 Prinzipien des Agilen Manifests (In Anlehnung an Beck et al. (2001))

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.

(Allgemeines, vereinfachtes) Agiles Entwicklungsmodell

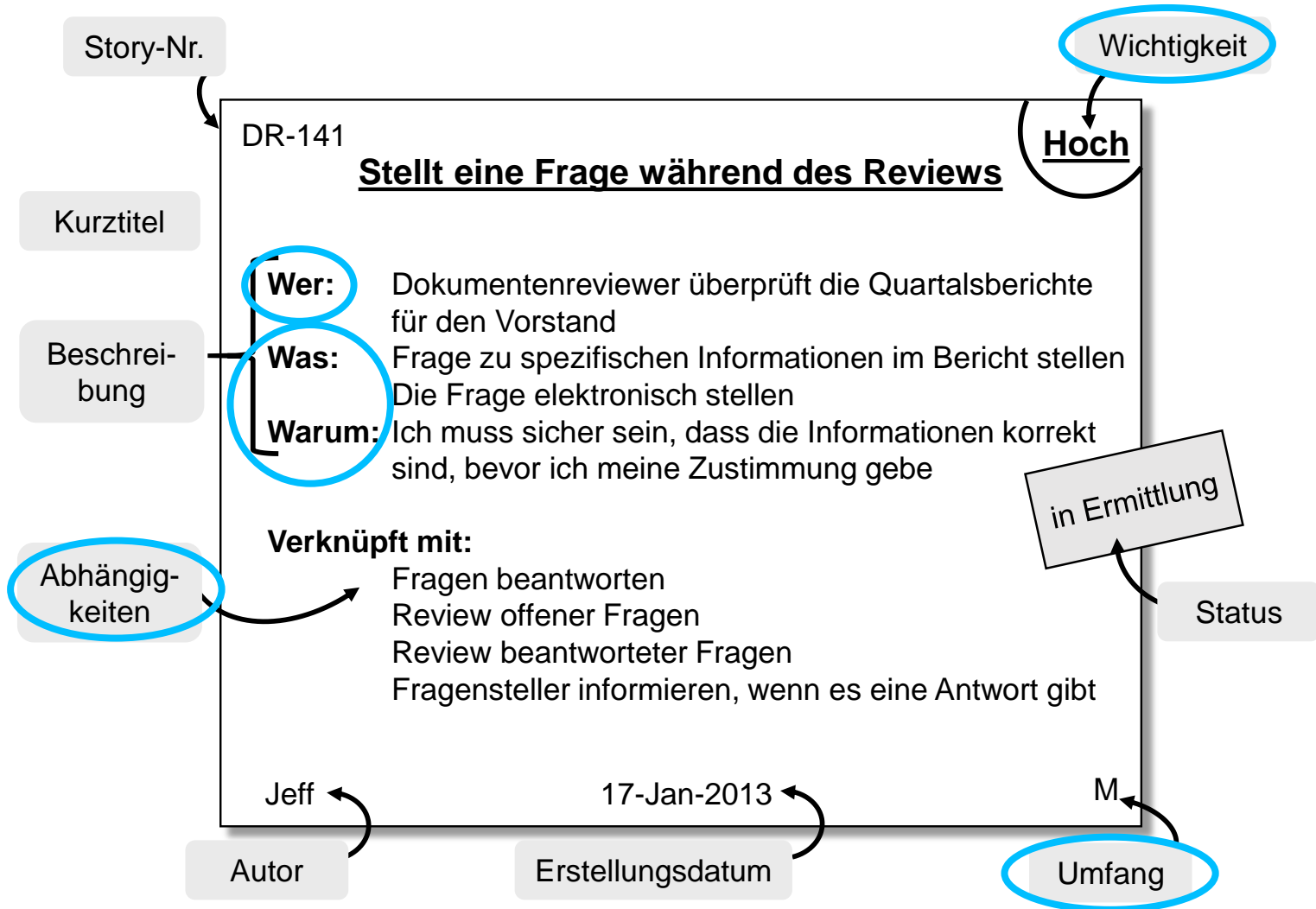


Verbreitete agile Praktiken relevant für die Analyse & Management von Anforderungen

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

Seltener

Beispiel für eine User Story



(In Anlehnung an Patton (2014), S. 114.)

Bewertung agiler Anforderungs-Praxis

→ Ansatzpunkte für Agiles Software QFD

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

Quelle: Schockert (2017)

Agiles Software QFD

Essentielle QFD-Techniken zur Ausrichtung am Business Value

- Verwendung von **(Extended) User Stories** angefangen von ihrer Sammlung in Grooming Meeting I bis zu ihrer Auswahl im Planungsmeeting I
- **Aufspaltung der User Stories in Stakeholderbedürfnisse und Produktanforderungen** (insb. mittels der Fragen der Fragen nach dem „Warum“ bzw. „Wozu“ eine Funktionalität benötigt wird)
- **Priorisierung der Bedürfnisse** durch die Stakeholder mittels Ranggruppenbildung oder Ranking
- **Verknüpfung von Stakeholderbedürfnissen und Produktanforderungen**
 - **Inkrementell wachsende Priorisierungsmatrix**
 - Relationendiagramm im Sinne einer Maximum Value Table (vgl. dazu Mazur (2012), Zultner (2007a))
- **Bewertung von Produktanforderungen** gemäß Abhängigkeiten (einfache Beziehungsmatrix), Aufwand/Umfang der Umsetzung, Risiken und weiterer Kriterien
- **Priority Map zur kompakten Darstellung** der Priorisierung der Stakeholderbedürfnisse, der Priorisierungsmatrix sowie der Bewertung der Produktanforderungen und ihrer Abhängigkeiten untereinander
- **Sprint Map mit Visualisierung** der für das nächste Inkrement ausgewählten Produktanforderungen und zu erfüllenden Stakeholderbedürfnisse

Extended 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)

→ Agile Welt: Hauptform der Kommunikation & Dokumentation von Anforderungen

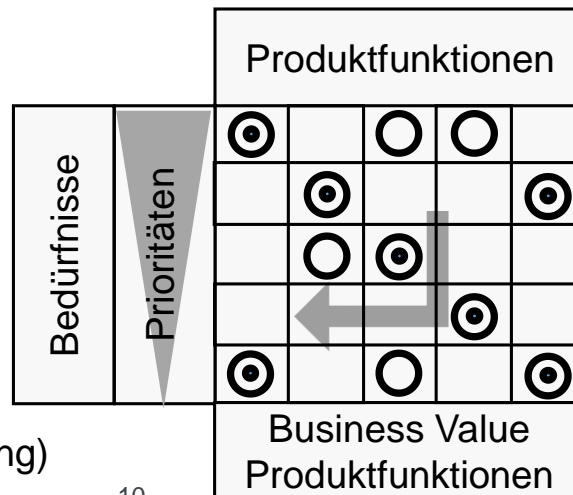
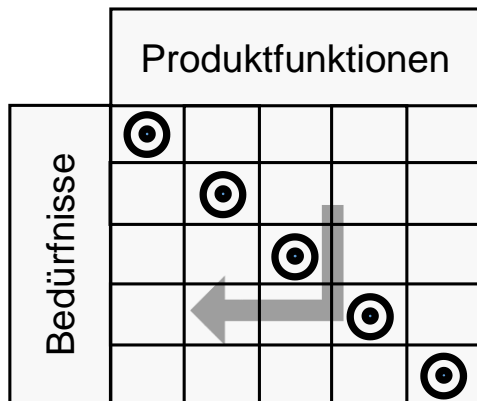
→ QFD Welt: „Als (Stakeholder) möchte ich (Produktfunktion), um (Bedürfnis) zu befriedigen.“ bzw. „Als (WER) möchte ich (WAS), um (WARUM) zu befriedigen.“

😊 Enthält Informationen über Grund eines Lösungswunsches → Bedürfnis

😞 Nur 1:1-Beziehungen zwischen Bedürfnissen und Lösungen

User Stories als Matrixdiagramm

Einfache Priorisierungsmatrix

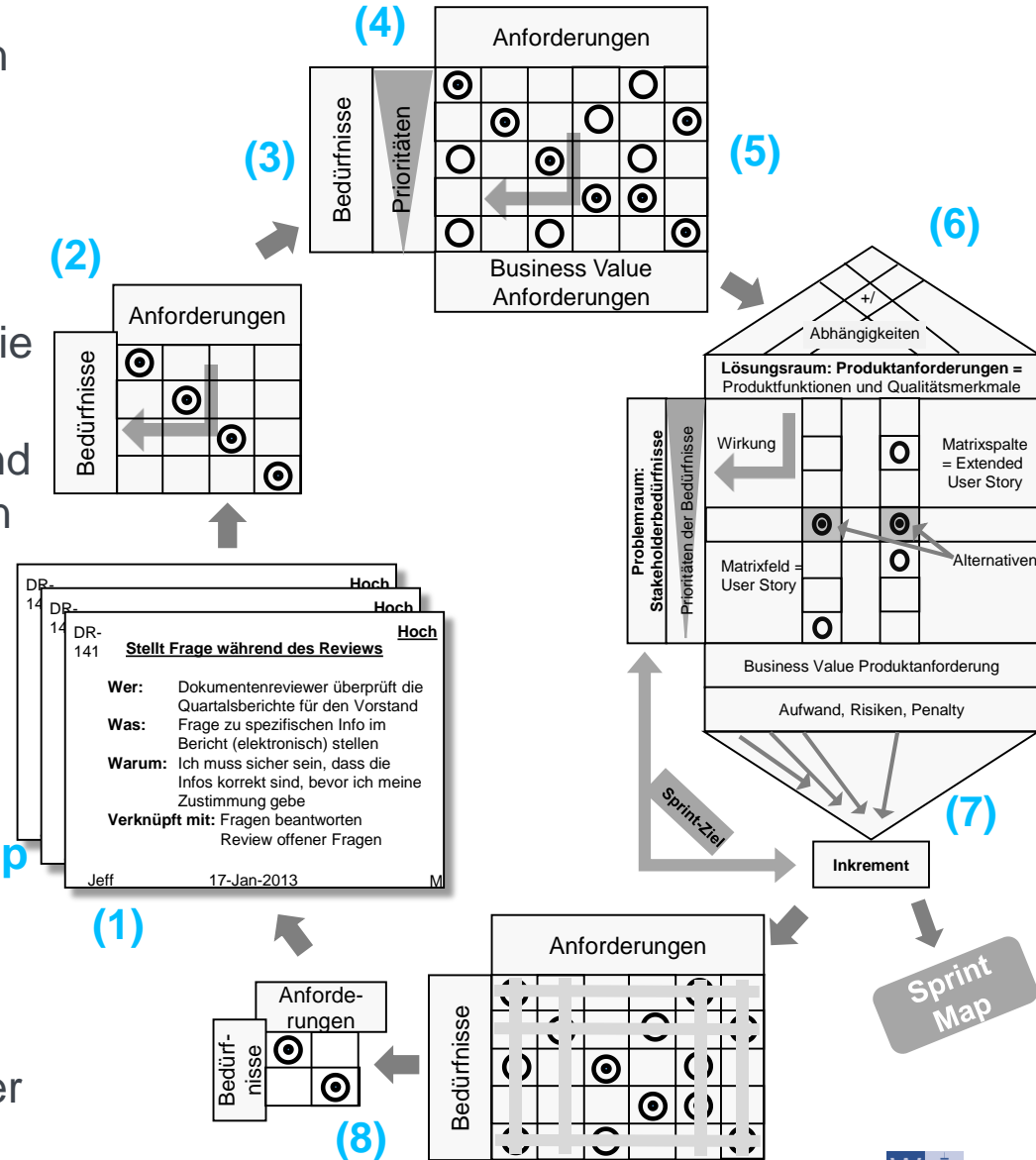


Spalten = Extended User Stories

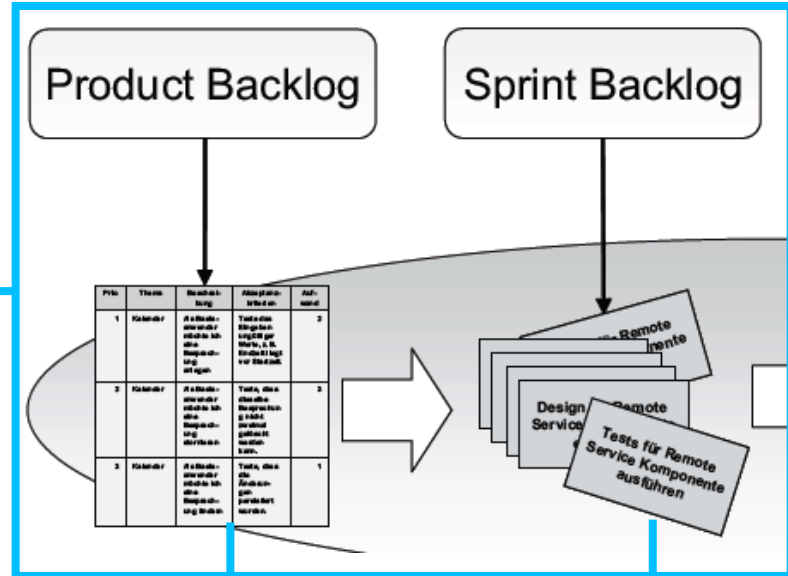
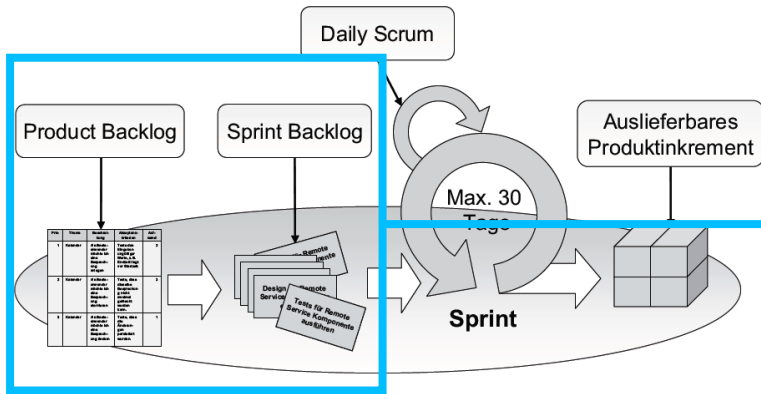
(⊙ Starke Wirkung)
(○ Mittlere Wirkung)

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



Product Backlog → Sprint Backlog

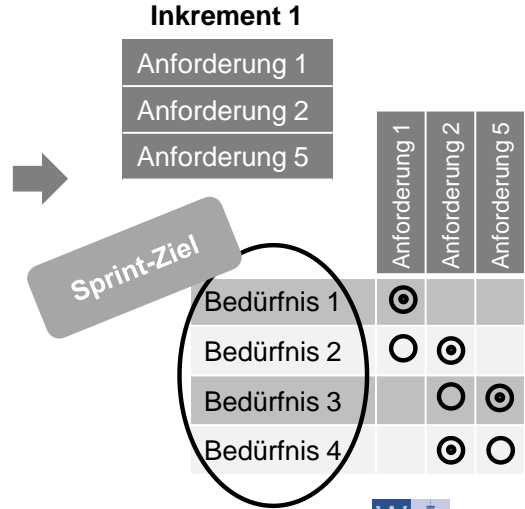


Priority Map

Sprint Map

- Anforderungen im **Product Backlog (Priority Map)** können jederzeit geändert, erweitert und verfeinert werden
 → **Pflege Product Backlog (Grooming Meetings)**
- Übernahme einer Gruppe von Anforderungen aus dem Product Backlog in den **Sprint Backlog (Sprint Map)**, der für den Sprint unverändert bleibt
 → **Planungsmeeting I & II**

	Anforderung 1	Anforderung 2	Anforderung 3	Anforderung 4	Anforderung 5	Anforderung 6	Anforderung 7	Anforderung 8
Bedürfnis 1	⊙		⊙	⊙				
Bedürfnis 2	○	⊙		⊙			○	
Bedürfnis 3		○	○		⊙	⊙		
Bedürfnis 4		⊙			○		⊙	
Bedürfnis 5				○				⊙
Bedürfnis 6						○		⊙



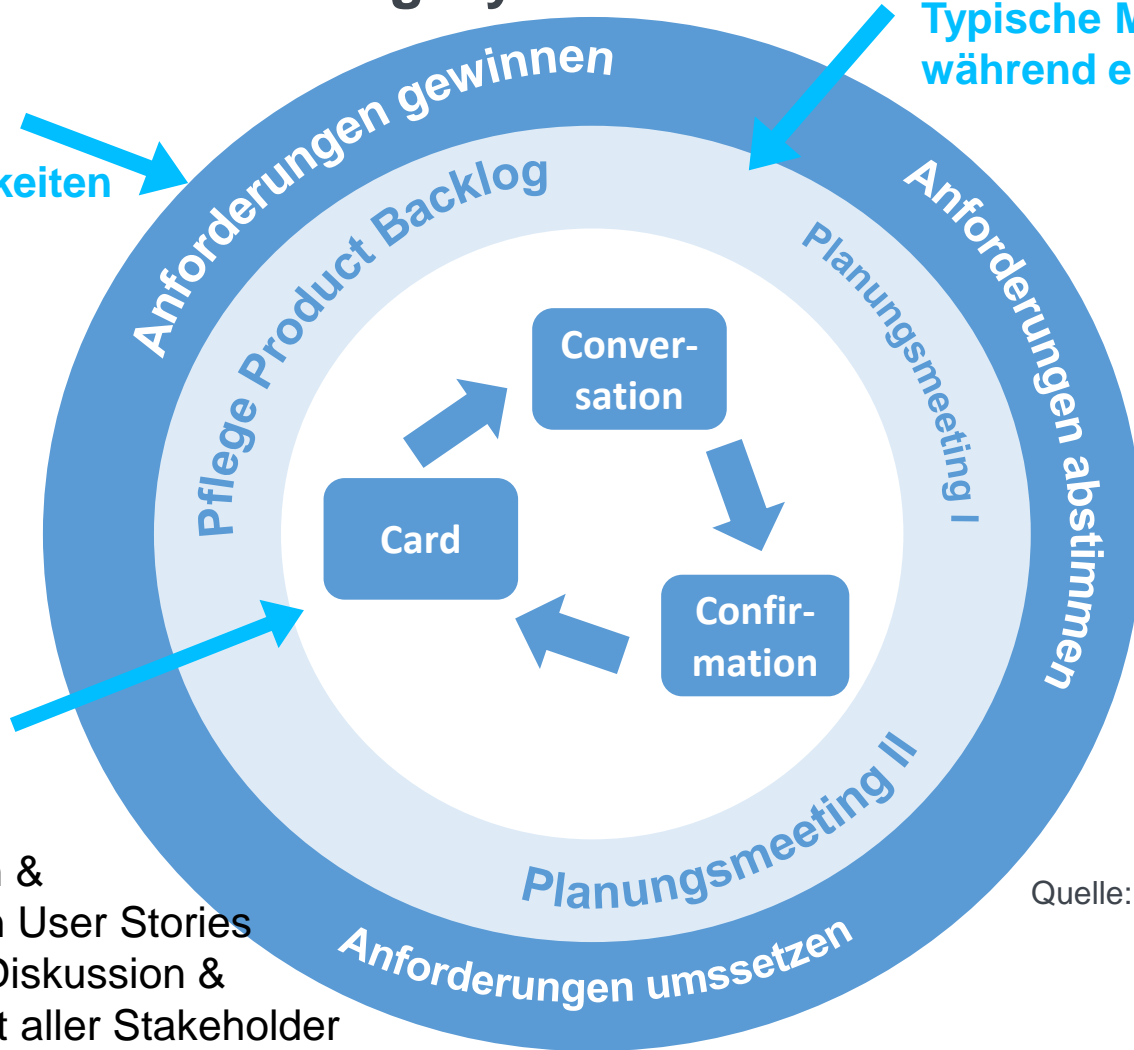
Schematische Einbettung von Software QFD in den agilen Entwicklungszyklus

Äußerer Kreis – QFD- Haupttätigkeiten

Mittlerer Kreis – Typische Meetings während eines Sprints

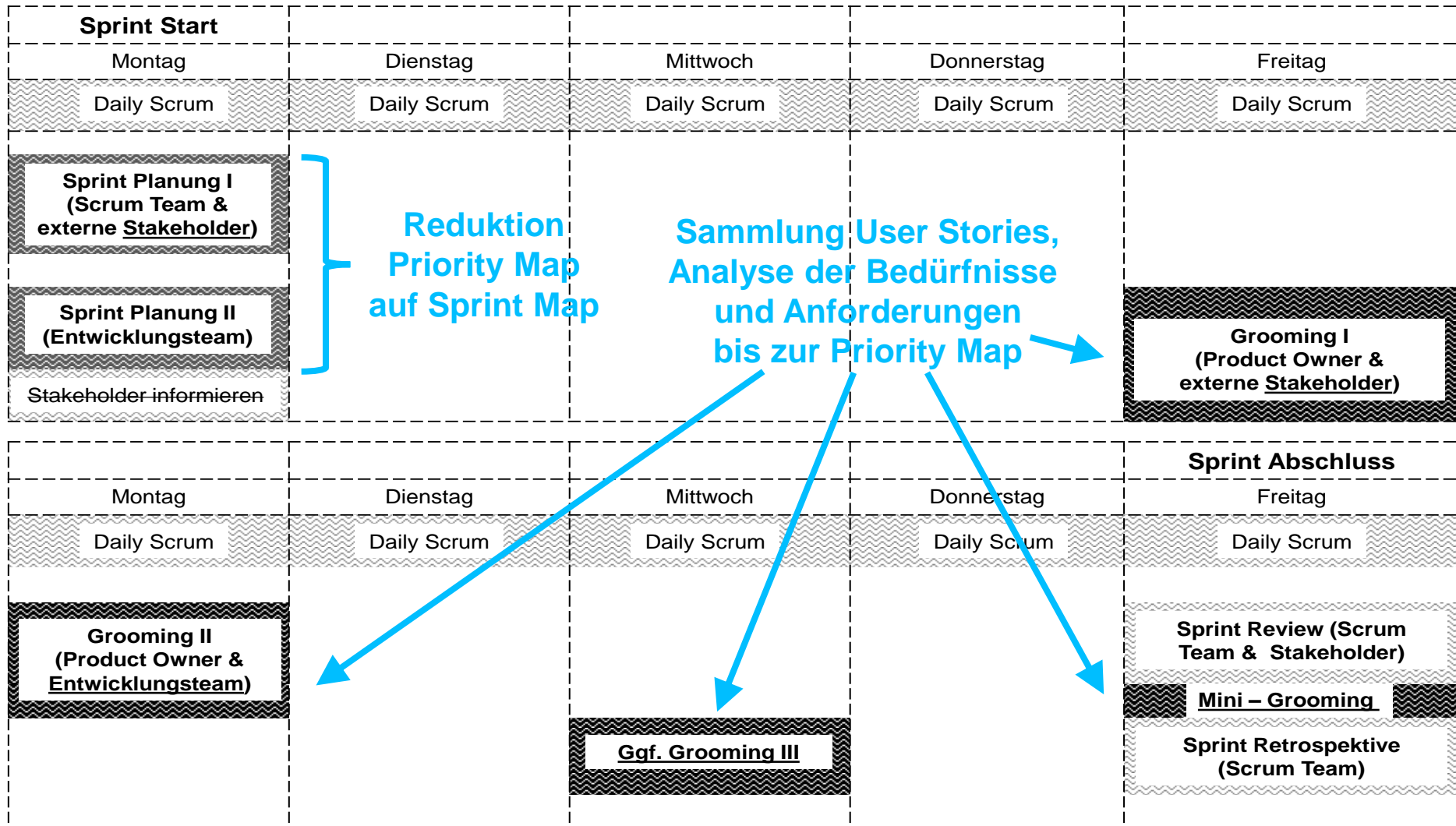
Innerer Kreis – User Stories - Aktivitäten:

- “Card” = Erkennen & Beschreiben von User Stories
- “Conversation” = Diskussion & Zusammenarbeit aller Stakeholder
- “Confirmation” = Detaillierung der User Stories zu Sprint Aufgaben



Quelle: Schockert (2017)

14-Tage Sprint mit Agilem Software QFD



Quelle: Schockert (2017), in Anlehnung an Foegen et al. (2013), S. 90-91

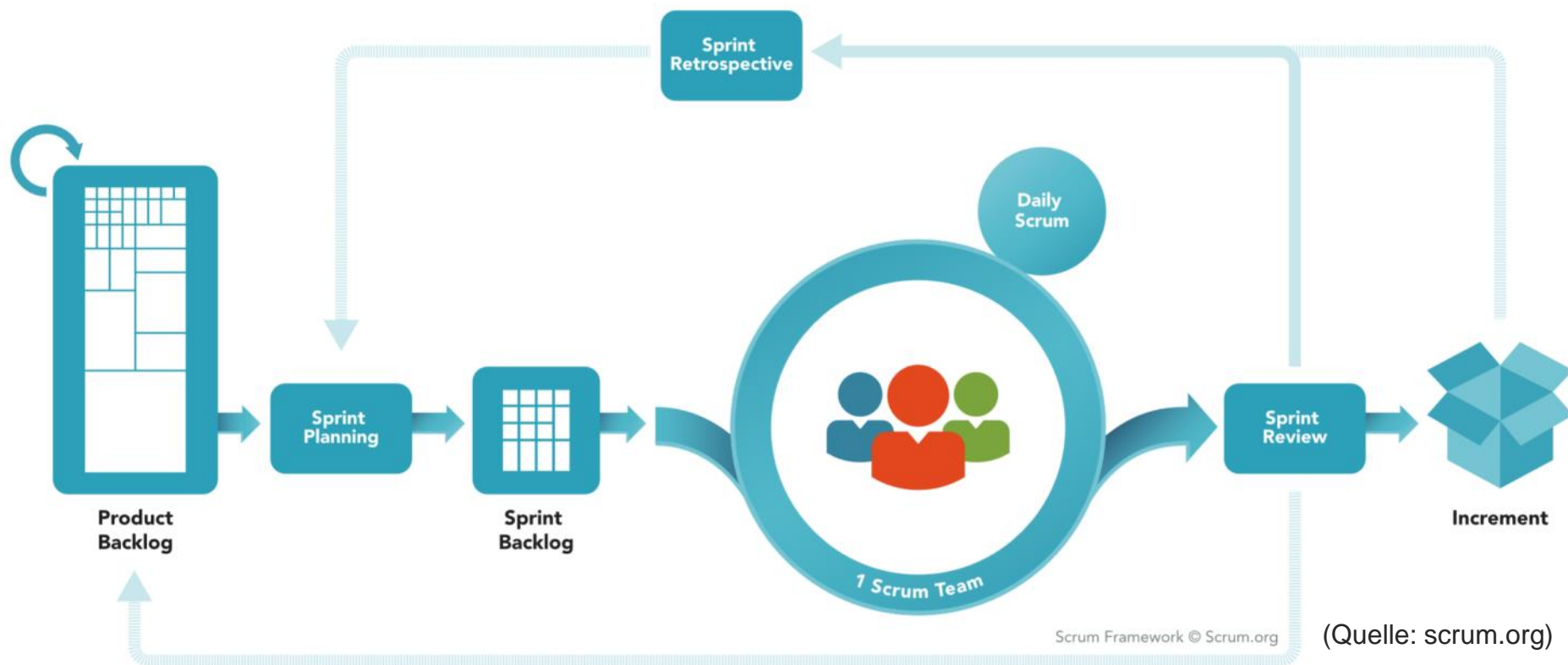
Vergleich Agiles Software QFD vs. Agile RE-Praxis 😊

➔ Verbesserungen durch Agiles Software QFD

Aber auch Vorteile Agile!

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

Agiles Entwicklungsmodell nach Scrum



Rollen in Scrum:

- **Scrum Master:** Überprüft die Einhaltung der Scrum Regeln, löst methodische Probleme, entwickelt, fördert und formt das Entwicklungsteam
- **Product Owner:** Repräsentiert die Stakeholder, führt die Anforderungspriorisierung durch, ist verantwortlich für Business Value
- **Entwicklungsteam:** führt Projektaufgaben aus und liefert die Produktfunktionalitäten

Agiles Vorgehen mit Scrum und User Experience (UX) – Phase im Vorlauf (1/4)

UX Approach

Gain insights from stakeholders and usability users in Stuttgart

Top Finding Overview

Hanc

Oktober

User --> Persona transformation



#1 People

The car is even providing comfort. Travel behaviour, attitudes and life situations of the user research participants were analysed and transferred into six personas that represent key characteristics.

Validate with us

September



#2 I can't

Public transport extra costs are a problem.



#3 Same

Many commuters take the usual way.



#4 Person

Daily chores of managing time of transportation.

#1 Achim
Young family man

#2 Ingrid
Careful organizer

#3 Susanne
Busy LOHAS mum

#4 Lukas
Young pragmatic individualist

#5 Selin
Day-dreamer

#6 Pete
Part-of-the-problem

Agiles Vorgehen mit Scrum und User Experience (UX) – Phase im Vorlauf (2/4)

Co-Creation Workshop August 17th

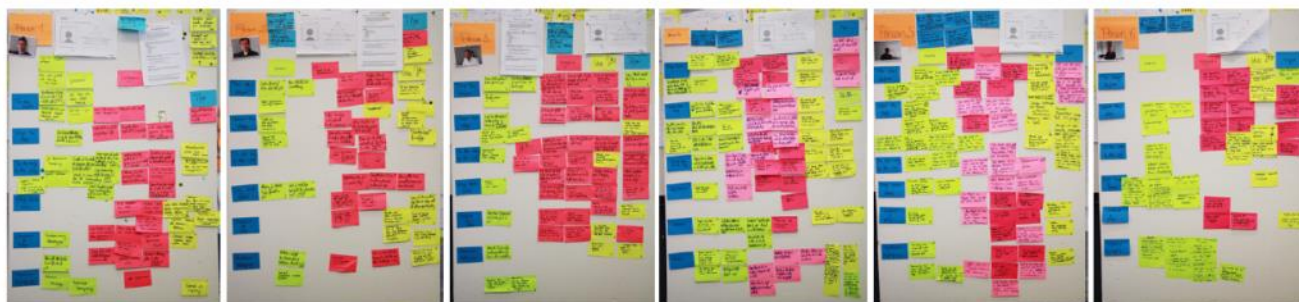
Design Sprint 1

Screen Examples (1/4)



User Test 1

Testing the 'Happy Stuttgart' concept with six users that were interviewed for 90 minutes each. All participants commute within the Stuttgart region. Their feedback was collected and compiled into a conclusion.



Agiles Vorgehen mit Scrum und User Experience (UX) – Phase im Vorlauf (3/4)

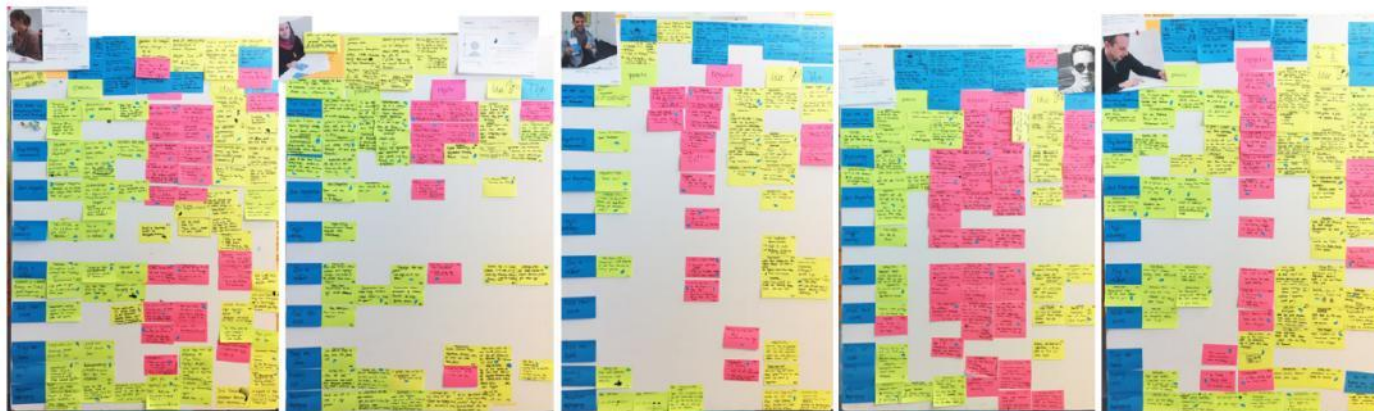
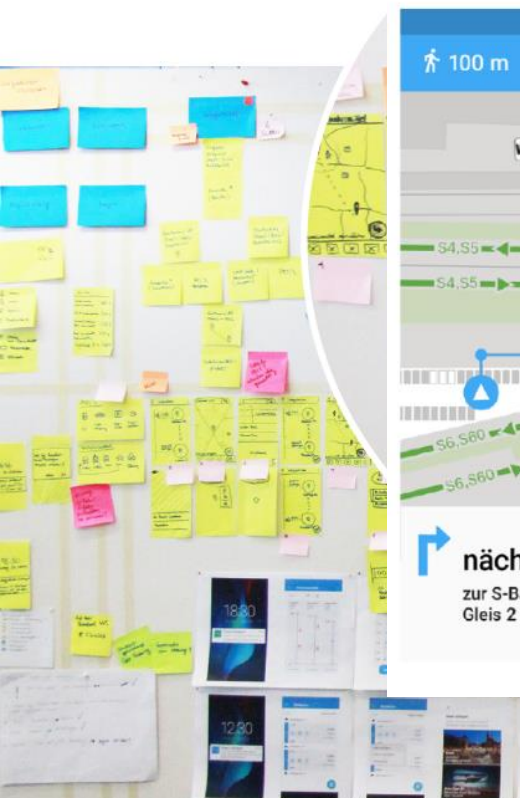
Design Sprint 2

Screen Examples (6/7)

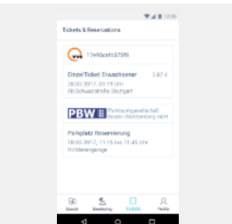


User Test 2


User testing with five participants that commute from or into Stuttgart by car; further characteristics are high smartphone usage (also for mobility), own a car, working, age 27-32.




Agiles Vorgehen mit Scrum und User Experience (UX) – Phase im Vorlauf (4/4)




Tickets & Reservations - Ove
116 kB 23.08.2017 15:06



Tickets & Reservations - PT Ticket
388 kB 23.08.2017 15:06



VVS_Android_EinzelTagesTicket
285 kB 08.08.2017 10:56



VVS_Android_EinzelTicket_3Z
328 kB 08.08.2017 10:56

Details Issue Links

Type: is related by [MOVEBW-653 Implement Buy-Ticket - \(Fares, multiple Tickets, Error Handling by HighQ, multiple tickets ...\)](#) NEW

Priority

Affects Activity

Comments Work Log History Activity Transitions

Activity

Comments

Comment 1: a.helferich@highq.de added a comment - 08.08.2017 11:07 - Restricted to jira-users
Die Angaben dazu, welche Daten schon vom TicketService geliefert werden bzw. noch implementiert werden müssen, sind noch mit Vorsicht zu genießen 😊 Hier sollten Michael Osswald und Markus Esenwein wohl noch einmal drüberschauen.

Comment 2: [Redacted] added a comment - 10.08.2017 10:24 - Restricted to jira-users - edited
In my opinion mainly in the information about the following points are missing:
*Zones (provided by TicketService or faresService)
*Name and last name, birthday (tdc where to enter the information)
*Conditions (provided by faresService)
*Creation date (transaction date)
This points will be provided to the APP with ICMSB-2848

Comment 3: [Redacted] added a comment - 18.08.2017 11:29 - Restricted to jira-users
Nach Absprache mit a.helferich@highq.de: Vorerst Kontrollgrafik als Platzhalter (Breite: 332 px, Höhe 322 px)

Comment 4: a.helferich@highq.de added a comment - 01.09.2017 17:12 - Restricted to jira-users
Feedback von VVS: Upcount notwendig, soll nach Ablauf der ersten 120 Sekunden langsam ein- und ausfaden (kein Wechsel auf Anzeige der Minuten bis Ticket abläuft, wie ich zuerst verstanden hatte). Siehe Film aus anderer App als neuen Anhang

Comment 5: a.helferich@highq.de added a comment - 01.09.2017 17:14 - Restricted to jira-users
Upcount-Verhalten nach Ablauf der ersten 120 Sekunden - aus App "Rheinbahn"

2 of 21

Export

6.07.17 17:15
7 16:08
7 15:19

Bewertung von Scrum & UX-Phase mit „Agile RE“-Brille

- Agiles Vorgehen lebt von der **engen Einbindung des Kunden** und der Anpassung der Planung anhand von dessen **Feedback**
 - Wer ist der Kunde?
 - Kann er mir kurzfristig ein klares Feedback geben?
- **Umfangreiche UX-Phase** sehr nahe am Ziel eines QFD-Workshops
 - Interdisziplinäres Team
 - Gemeinsames Verständnis schaffen
 - Leichtgewichtige Dokumentation
- **Keine Trennung in Anforderungen und Lösungen**, mit Screens sehr schnell Demonstration von (möglichen) Lösungen
- Gefahr, sich im **Wald der User Stories** zu verirren → User Mapping-Workshops als Hilfe, aber herausfordernd und aufwendig
- **Hauptsächliche Herausforderungen** unabhängig von RE
 - Großes, räumlich und über Unternehmen verteiltes Projektteam
 - Inhaltlich komplexes, innovatives Projekt
 - Wechsel von Teammitgliedern und Teammitglieder, die nicht Vollzeit auf dem Projekt arbeiten

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	+	+

Fazit & Ausblick

- Es gibt nicht DAS agile Vorgehen & es gibt nicht DAS QFD-Vorgehen
→ **Es gibt nicht DAS agile (Software) QFD Vorgehen!**
- Agiles Software QFD entspricht **Evolution von (Continuous) QFD** und Integration mit agiler d. h. iterativer & inkrementeller Entwicklung
- **UX verbessert Agiles RE** um Upfront-Aktivitäten und ist nah an QFD
- Aber **Bedarfe für Forschung und Praxis:**
 - Wie viel Kundenintegration ist sinnvoll?
 - Was ist mit weiteren Methoden wie Story Mapping, Design Thinking, Buglione-Trudel-Matrix, Twin Peaks Ansatz etc.?
 - Wie adaptieren für Nicht-Software-Domänen?
 - Was mit „mehr“ Mathematik und präziseren Berechnungen?
 - Wie integrieren bzw. kombinieren mit ISO 16355?
 - **Und, und, und...**

Arbeitskreis?!!



Universität Stuttgart

Vielen Dank!



Sixten Schockert

E-Mail schockert@wius.bwi.uni-stuttgart.de, schockert@qfd-id.de

Telefon +49 (0) 711 685-82387

Fax +49 (0) 711 685-82388



Dr. Andreas Helferich

E-Mail a.helferich@highQ.de

Telefon +49 (0) 152 54 96 2006

Fax +49 (0) 761 70 60 44