

Agile Programmierung - Theorie II

SCRUM

Arne Brenneisen

Universität Hamburg

Fakultät für Mathematik, Informatik und Naturwissenschaften

Seminar Softwareentwicklung in der Wissenschaft

Betreuer: Christian Hovy

23. Juni 2014

Agenda

1. Agiles Manifest
2. Warum ist Scrum so erfolgreich?
3. Was ist Scrum?
4. Wie funktioniert Scrum?
5. Grenzen und Nachteile
6. Zusammenfassung

1. Agiles Manifest

1. Agiles Manifest

Individuen und Interaktionen sind wichtiger als Prozesse und Werkzeuge

Funktionierende Software ist wichtiger als umfangreiche Dokumentation

Zusammenarbeit mit Projektbetroffenen ist wichtiger als Vertragsverhandlungen

Reaktion auf Änderungen ist wichtiger als Verfolgung eines festgelegten Plans

2. Warum ist Scrum so erfolgreich?

2.1. Dynamisches Umfeld als Indikator

2.2. Komplexität in Projekten

2.3. Vorteile durch Scrum

2.1. Dynamisches Umfeld als Indikator

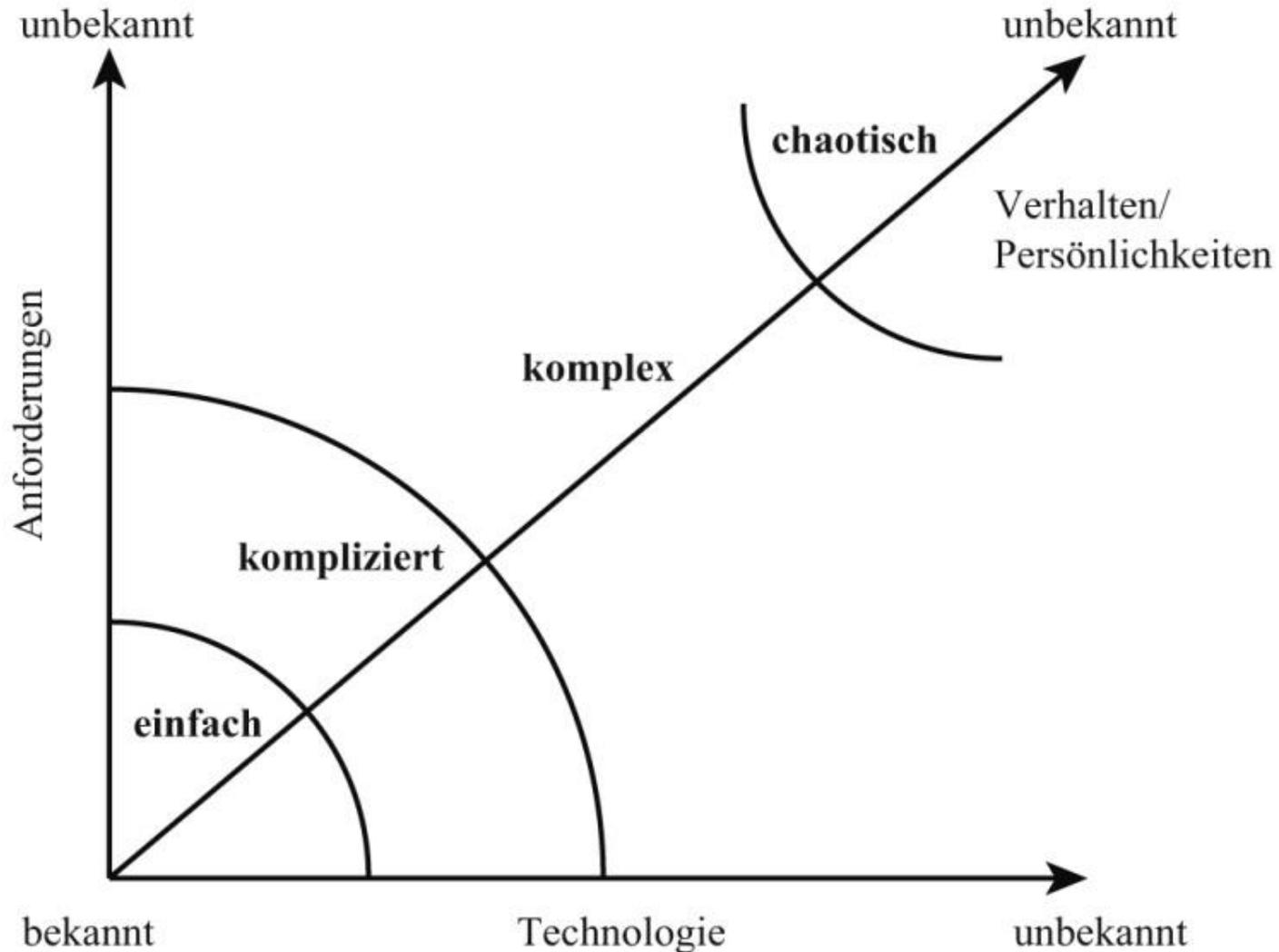
Rahmenbedingungen können sich ändern

Anforderungen sind oft instabil oder unvollständig

Überraschungen sind mittlerweile der Normalfall

- Entwicklungen in der Umwelt
- Entwicklungen innerhalb des Projekts
- Annahmen, die sich als unzutreffend erweisen

2.2. Komplexität in Projekten



[Abb. 1](#)

2.3. Vorteile durch Scrum

Höhere Kundenorientierung durch Vorgehen nach Prioritäten

Mehr Transparenz schafft Sicherheit und Vertrauen

Mehr Flexibilität und Überraschungsrobustheit

Besseres Teamwork und höhere Mitarbeitermotivation

Arbeitslast verteilt sich besser

Fokus auf der Qualität des Produkts

Insgesamt höhere Nachhaltigkeit

3. Was ist Scrum?

3. Was ist Scrum?

Agile Projektmanagementmethode

Wurde in den frühen 1990er Jahren entwickelt

Ist der De-facto-Standard in der agilen Softwareentwicklung

Folgt der Idee des Lean Managements

Ablauf des Projekts steht im Vordergrund

Vorgehensmodell mit *Rollen*, *Meetings* und *Artefakten*

Entwicklungszyklen von wenigen Wochen (*Sprints*)

Charakteristika

Selbst-organisierende *Teams*

Produkt wird in Abschnitten von monatlichen *Sprints* entwickelt

Alle Anforderungen sind im *Product Backlog* festgehalten

Mehrere Arten von *Meetings* mit bestimmten Zwecken

Iteratives Vorgehen

Wenige Regeln um ein agiles Umfeld zu schaffen

4. Wie funktioniert Scrum?

4.1. Sprints

4.2. Prozess

4.3. Rahmen

4.1. Sprints

Analog zu den Iterationen des “Extreme Programming”

Feste Iterationslängen und fester Endtermin

Typische Sprintdauer beträgt 2 - 4 Wochen

Besserer Rhythmus durch konstante Dauer

Entwurf, Implementierungen und Test

Team darf während des Sprints nicht gestört werden

Ziele werden in Sprint-Planungssitzung definiert

Sprint Backlog enthält alle Anforderungen

4.2. Prozess

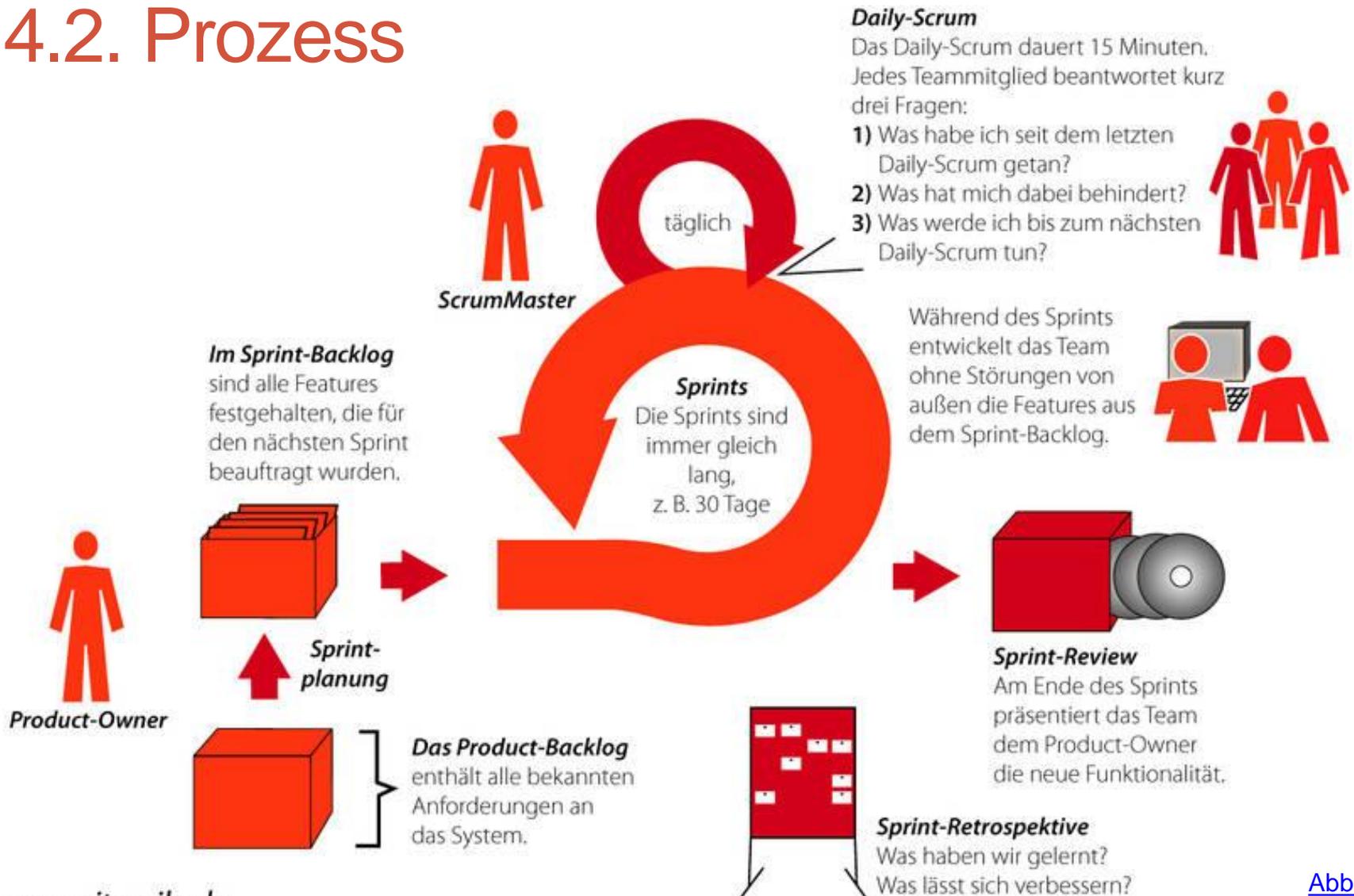


Abb. 2

4.3. Rahmen

Rollen

- Product Owner
- Team
- ScrumMaster

Meetings

- Sprint-Planung
- Daily Scrum
- Sprint-Review
- Sprint-Retrospektive

Artefakte

- Product Backlog
- Sprint Backlog
- Releaseplan / Burndown-Diagramm

4.3.1. Product Owner

Produktverantwortlicher

Übernimmt die Sicht des Kunden

Steuert die Software-Entwicklung

Erfasst die Anforderungen und priorisiert Produkt-Features

Beschreibt diese in Form des *Product Backlogs*

Bestimmt Auslieferungsdatum und Inhalt

Ist verantwortlich für das finanzielle Ergebnis des Projekts

Passt Features und Prioritäten nach Bedarf für jeden Sprint an

Akzeptiert oder weist Arbeitsergebnisse zurück

4.3.2. Team

Entwickelt das Produkt

Keine Teamleitung

Zusammensetzung ist entscheidend

Typischerweise 5-9 Personen

Funktionsübergreifend

Selbstorganisierend

Vollzeitmitgliedschaft der Mitglieder

Mitgliedschaft kann sich nur zwischen Sprints verändern

4.3.3. ScrumMaster

Prozessverantwortlicher

Coacht das Team und etabliert den Prozess

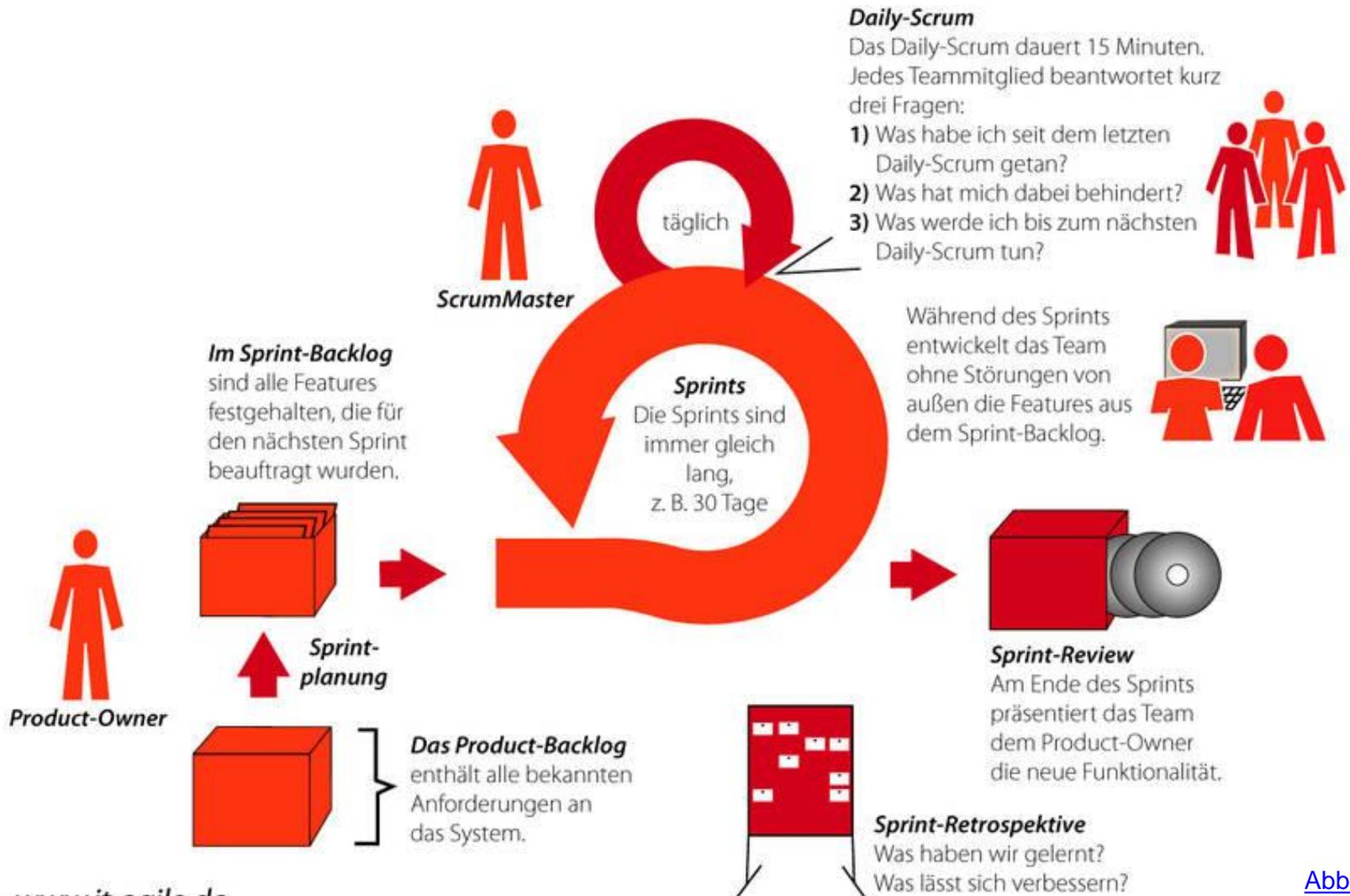
Repräsentiert das Management gegenüber dem Projekt

Verantwortlich für die Einhaltung der Regeln

Beseitigt Hindernisse und schützt das Team vor Störungen

Stellt sicher, dass das Team produktiv ist

Unterstützt die Zusammenarbeit zwischen allen Rollen



4.3.4. Sprint-Planung

Product Backlog wird analysiert und ausgewertet

Product Owner legt Sprint Ziel fest

Entscheiden, wie man das Sprint Ziel erreichen kann

Sprint Backlog wird aus *Product Backlog* und *Releaseplan* gemäß dem Ziel erstellt

Zeitlicher Aufwand für *Sprint Backlog* muss geschätzt werden

4.3.5. Daily Scrum

Teambesprechung

Täglich 15 Minuten lang, selber Ort, selbe Zeit

Ziel ist es die Arbeit des Teams zu koordinieren und Probleme anzusprechen

Sprint Backlog wird aktualisiert

Gäste sind erlaubt, haben aber kein Rederecht

Jedes Team-Mitglied muss drei Fragen beantworten

Fragen an Team-Mitglieder

Was habe ich seit dem letzten Daily Scrum für das Projekt getan?

Was will ich bis zum nächsten Daily Scrum für das Projekt tun?

Was hindert mich daran, so effektiv wie möglich zu arbeiten?

4.3.6. Sprint-Review

Team präsentiert das entstandene Software-Inkrement

Maximal 4 Stunden

Product Owner akzeptiert nur vollständige Resultate

Live-Demo der neuen Features auf einem Integrationsrechner

ScrumMaster moderiert die Sitzung

Das ganze *Team* nimmt teil

Gut ist es wenn Endanwender bei der Sitzung dabei sind

4.3.7. Sprint-Retrospektive

Was hat gut und was hat nicht so gut funktioniert?

Typischerweise 15–30 Minuten lang, maximal 3 Stunden

Nach jedem *Sprint-Review*

Das ganze Team nimmt teil

ScrumMaster notiert vorgeschlagene Veränderungen

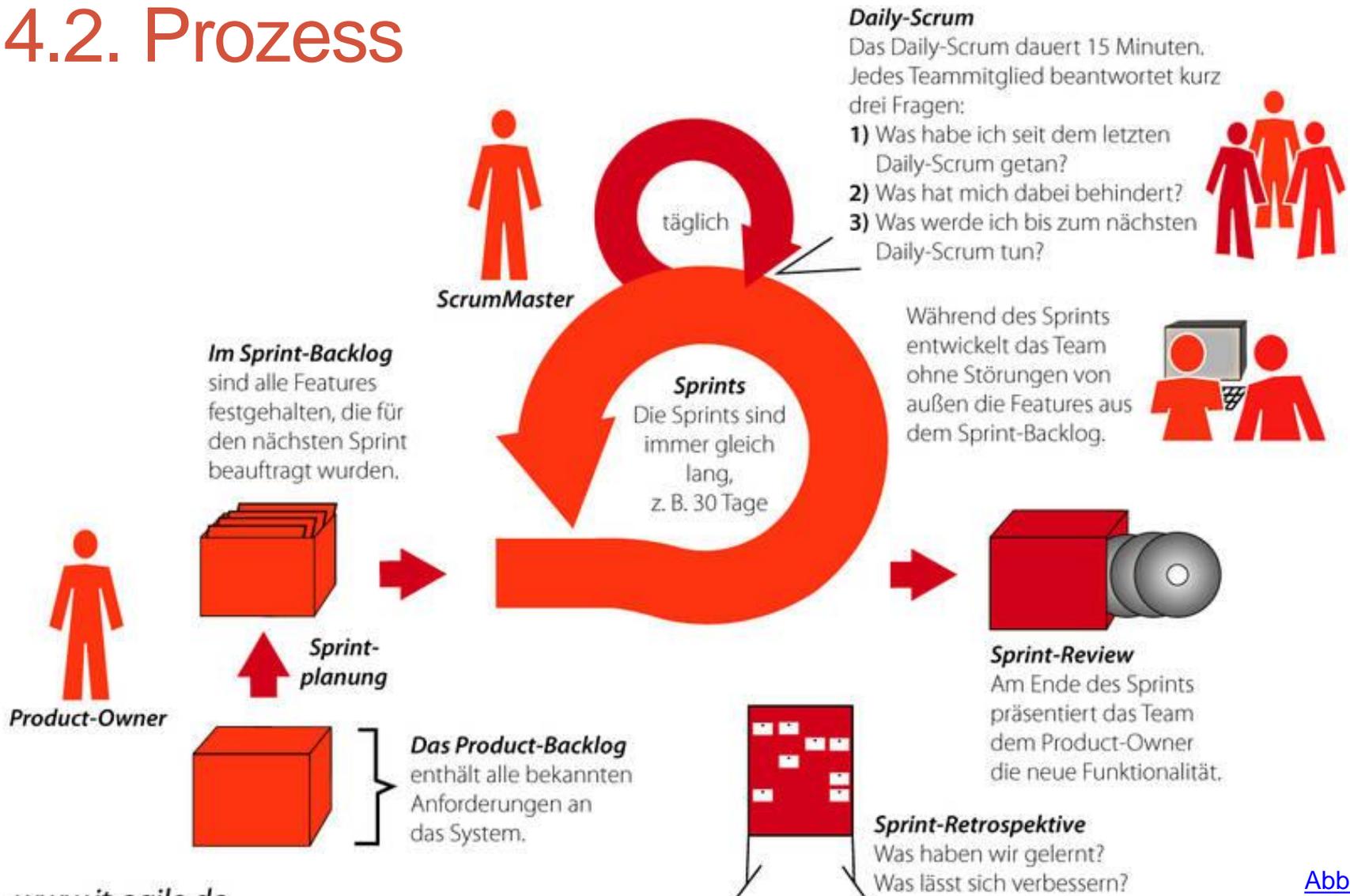
Team diskutiert Veränderungen

Veränderungen werden gemeinsam beschlossen

Ziel ist Optimierung des Prozesses

Grundlage für den nächsten *Sprint*

4.2. Prozess



4.3.8. Product Backlog

Zentrales Dokument zum Erfassen und Verwalten der Anforderungen

Gesammelt werden die zu liefernden Funktionalitäten

Product-Owner priorisiert die Funktionalitäten

Alle Anforderungen werden zu Beginn jedes Sprints neu priorisiert

Product Backlog muss für das gesamte Team deutlich sichtbar sein

4.3.9. Sprint Backlog

Liste der Aktivitäten eines Sprints

Anforderungen für Sprint Ziel werden identifiziert

Aufwand muss geschätzt werden, 4 - 16 Stunden

Team-Mitglieder wählen Tasks selber aus

Die geschätzte restliche Arbeit wird täglich aktualisiert

Jedes Team-Mitglied kann Tasks hinzufügen, löschen oder ändern

Oft in Form von User Stories an einer Stellwand

4.3.10. Burndown-Diagramm

Releaseplan bei größeren Projekten

Abfolge der Sprints wird geplant

Gibt Auskunft über den zeitlichen Rahmen

Überwachung der Planung mit Hilfe des Burndown-Diagramm

Stellt pro Sprint den insgesamt verbleibenden Aufwand dar

5. Grenzen und Nachteile

5. Grenzen und Nachteile

Gewonnene Erkenntnisse müssen auch verwertet werden

Problematisch, wenn ein Projekt engen Zeit- oder Budgetvorgaben unterliegt

Hinderliche Einflüsse bei der Teamzusammensetzung

Hoher Aufwand für Tests

Juristische Erwägungen

Kann zu Konflikten führen

Bedeutung der wenigen Scrum-Elemente ist im Detail nicht immer offensichtlich

Verleitet oft besonders Anfänger zu kleinen, gravierenden Fehlentscheidungen

6. Zusammenfassung

6. Zusammenfassung

Agiles Vorgehensmodell

Höhere Transparenz und Sicherheit

Große Flexibilität und Überraschungsrobustheit

Überschaubares Rahmenwerk

Wenige Rollen, Artefakte und Meetings

Fordert von den Beteiligten Verantwortung zu übernehmen

Rolle des ScrumMasters sehr wichtig

Quellen

- Dominik Maximini, Scrum - Einführung in der Unternehmenspraxis, 2013
- Eckhart Hanser, Agile Prozesse: Von XP über Scrum bis MAP, 2010, Springer
- <http://www.contentmanager.de/cms/portalmanagement/scrum-ist-nicht-planlos/>
- <http://www.computerwoche.de/a/warum-scrum-so-erfolgreich-ist,1231935>
- <https://github.com/fh-koeln/wba1-2012/wiki/Agiles-Projektmanagement-Dokumentation>
- <https://www.fernuni-hagen.de/imperia/md/content/ps/masterarbeit-harwardt.pdf>
- <http://tigertechtalk.wordpress.com/2013/02/04/grenzen-und-moglichkeiten-von-scrum/>
- <http://blog.seibert-media.net/blog/2010/12/03/welche-vorteile-bietet-mir-als-kunde-ein-scrum-projekt/>
- <http://www.it-agile.de/wissen/vorteile-agiler-methoden/>
- http://www.gpm-ipma.de/fileadmin/user_upload/Know-How/studien/PM_2_14_S40.pdf