



INSTITUT FÜR INFORMATIK
DER LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Präsentation

Juristisches IT-Projektmanagement im Wintersemester 2017/2018

Vergütungsaspekte bei agilen IT-Projekten

Robert Keiselt

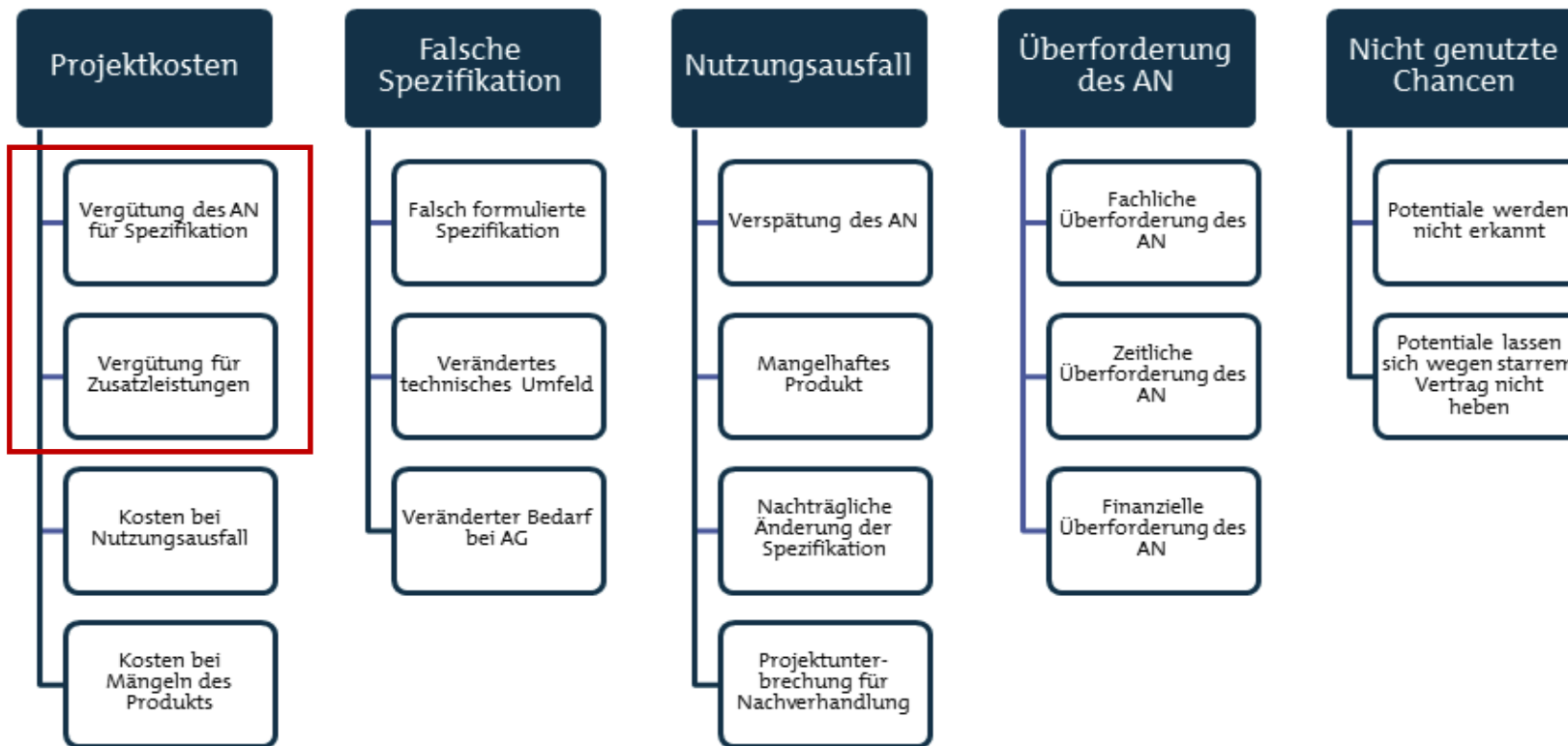
09. Januar 2018

Gliederung

- Einleitende Gedanken
- Entwicklungsmethoden
- Mögliche Vertragsarten
- Vergütungsmodelle
- Problemstellungen
- Zusammenfassung

Einleitung

RISIKEN BEI SOFTWAREENTWICKLUNGSPROJEKTEN

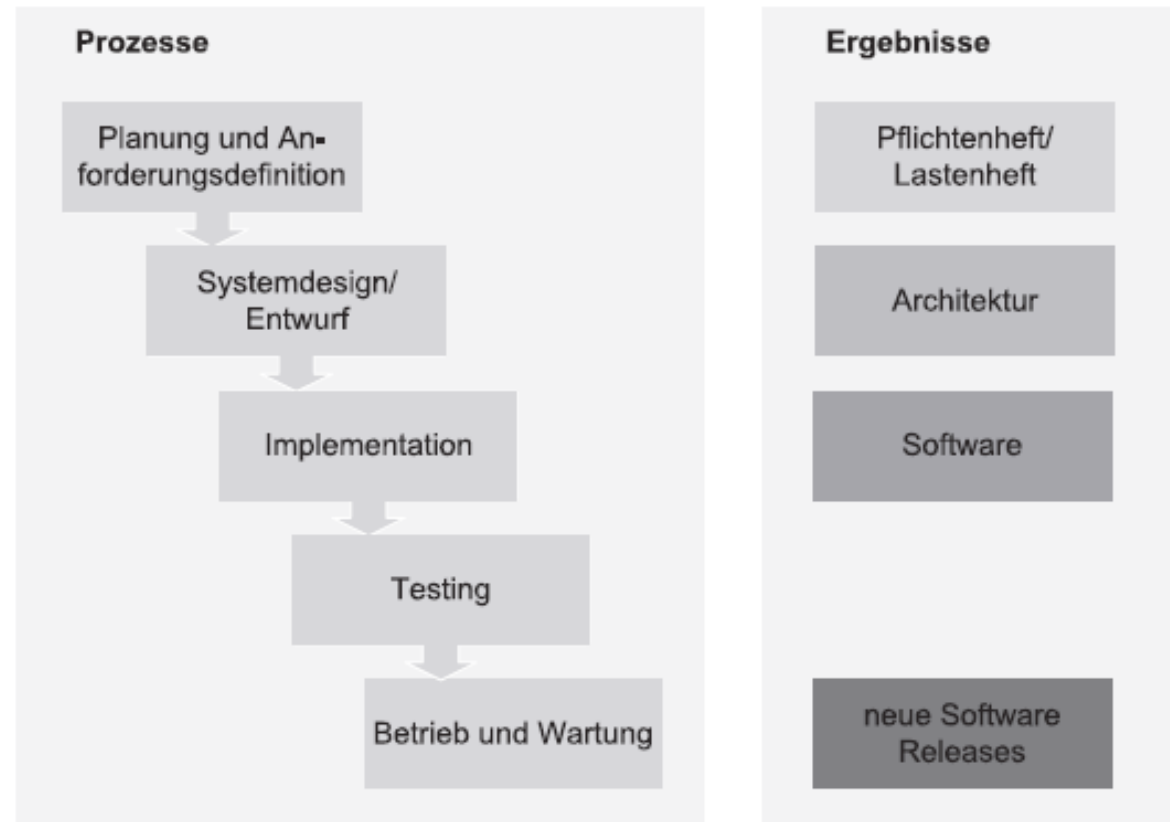


Entwicklungsmethoden

Klassische Methoden

- Wasserfallmodell
- Spiralmodell
- V-Modell XT
- HERMES
- ITIL RM
- etc.

Phasenmodelle



Agile Vorgehensweisen

- Extreme Programming (XP)
- Feature Driven Development (FDD)
- Dynamic Systems Development Method (DSDM)
- Crystal Methods
- Kanban/Lean SD
- SCRUM
- etc.

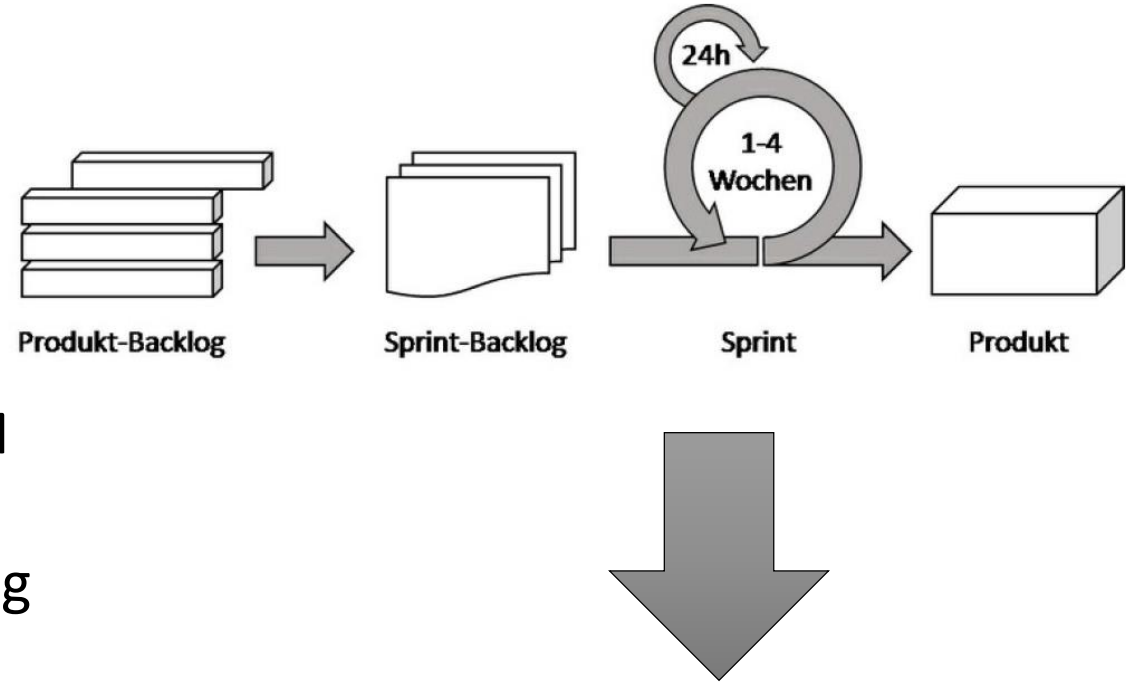
Vier Leitsätze und 12 Prinzipien des Agilen Manifests:

1. Individuen und Interaktionen sind wichtiger als Prozesse und Werkzeuge
2. Funktionierende Software hat Vorrang vor umfassender Dokumentation
3. Zusammenarbeit mit den Kunden ist wichtiger als Vertragsverhandlungen
4. Reagieren auf Veränderung geht vor Befolgen eines Plans.

1	Höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufrieden zu stellen.
2	Anforderungsänderungen sind selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden.
3	Funktionierende Software soll regelmäßig innerhalb weniger Wochen oder Monate unter Bevorzugung der kürzeren Zeitspanne geliefert werden.
4	Fachexperten 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.
6	Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteams zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.
7	Funktionierende Software ist das wichtigste Fortschrittsmaß.
8	Agile Prozesse fördern nachhaltige Entwicklung. Die Auftraggeber, Entwickler und Benutzer sollten ein gleichmäßiges Tempo auf unbegrenzte Zeit einhalten können.
9	Ständiges Augenmerk auf technische Exzellenz und gutes Design fördert Agilität.
10	Einfachheit – die Kunst, die Menge nicht getaner Arbeit zu maximieren – ist essenziell.
11	Selbstorganisierte Teams erstellen die besten Anforderungen, Architekturen und Entwürfe.
12	Das Team reflektiert regelmäßig wie es effektiver werden kann und passt sein Verhalten an.

SCRUM

- Aktivitäten, Artefakte und Rollen
- Product Backlog, Sprint Backlog, Sprint und Product Increment
- User Story, Story Points, Daily SCRUM und Definition of Ready bzw. Done
- Transparenz, Überprüfung, Anpassung
- Vision, schnelle Entwicklung ohne umfangreiche Lasten-/Pflichtenhefte
- Erfolg zeigt sich u.a. in der Grafik:



Sprint = Mini-Werkvertrag ?

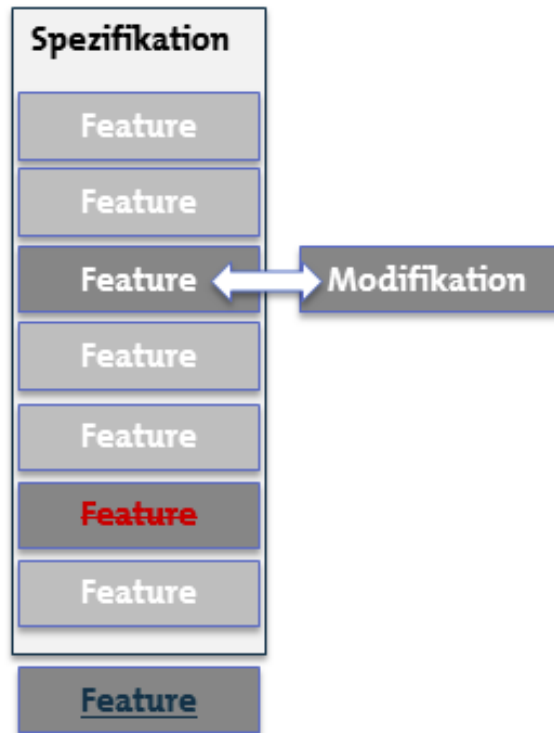
Methode	Erfolgreich	Teilweise erfolgreich	Gescheitert
Agil	39%	52%	9%
Wasserfall	11%	60%	29%

Vertragsarten

- Kaufvertrag
- Werkvertrag
- Dienstvertrag
- Rahmenvertrag
- IT-Projektvertrag
- ggfs. Sonderfälle
- etc.

VERTRAGSMODELLE

Klassischer Werkvertrag



Vorteile:

- Kostensicherheit bei gleichbleibender Spezifikation
- Verantwortung des Auftragnehmers bei Mängeln
- Vorhersehbarkeit für den Auftragnehmer

Nachteile:

- Änderungen erfordern Nachverhandlungen
- Änderungen sind mit unvorhersehbaren Kosten verbunden
- Bedingter Anreiz für den AN über Verbesserungen im Sinne des AG nachzudenken

Geeignet für:

Einfache Projekte in unveränderlicher Umgebung

VERTRAGSMODELLE

Dienstvertrag

Dienstvertrag
Sprint 1
Sprint 2
Sprint 3
Sprint 4
Sprint 5
Sprint 6



Vorteile:

- Volle Flexibilität im Hinblick auf Vertragsgegenstand
- Wirtschaftliche Sicherheit für Auftragnehmer

Nachteile:

- Zeit und Kosten unvorhersehbar
- Ggf. Zusatzkosten für Mängelbeseitigung

Geeignet für:

Kleinere Projekte und Einzelkomponenten im Rahmen einer eingespielten Zusammenarbeit

VERTRAGSMODELLE

Rahmenvertrag mit einzelnen Werkverträgen

Rahmenvertrag

Werkvertrag Sprint 1

Werkvertrag Sprint 2

Werkvertrag Sprint 3

Werkvertrag Sprint 4

Werkvertrag Sprint 5

Werkvertrag Sprint 6

Werkvertrag Sprint 7



Vorteile:

- Hohe Flexibilität
- Teilabnahmen
- Mängelgewährleistung
- Kostensicherheit bei gleichbleibender Spezifikation möglich

Nachteile:

- Verhandlungsaufwand vor jedem Sprint
- Verbleibende Kostenunsicherheit bei Änderungen im Scope

Geeignet für:

Komplizierte Projekte ohne Zeitdruck

VERTRAGSMODELLE

Agiler Festpreis

Werkvertrag
Sprint 1: x Story Points
Sprint 2: x Story Points
Sprint 3: x Story Points
Sprint 4: x Story Points
Sprint 5: x Story Points
Sprint 6: x Story Points



Vorteile:

- Hohe Flexibilität
- Haftung des Auftragnehmers für Mängel
- Prozesse für Scope Änderungen im Vertrag angelegt

Nachteile:

- Testphase erforderlich
- Zeit und Kosten erst nach Testphase planbar
- Referenz-Userstories geben nur bedingte Sicherheit

Geeignet für:

- Eingespielte Teams aus AN und AG
- Hausinterne Softwareentwicklung

Vergütungsmodelle

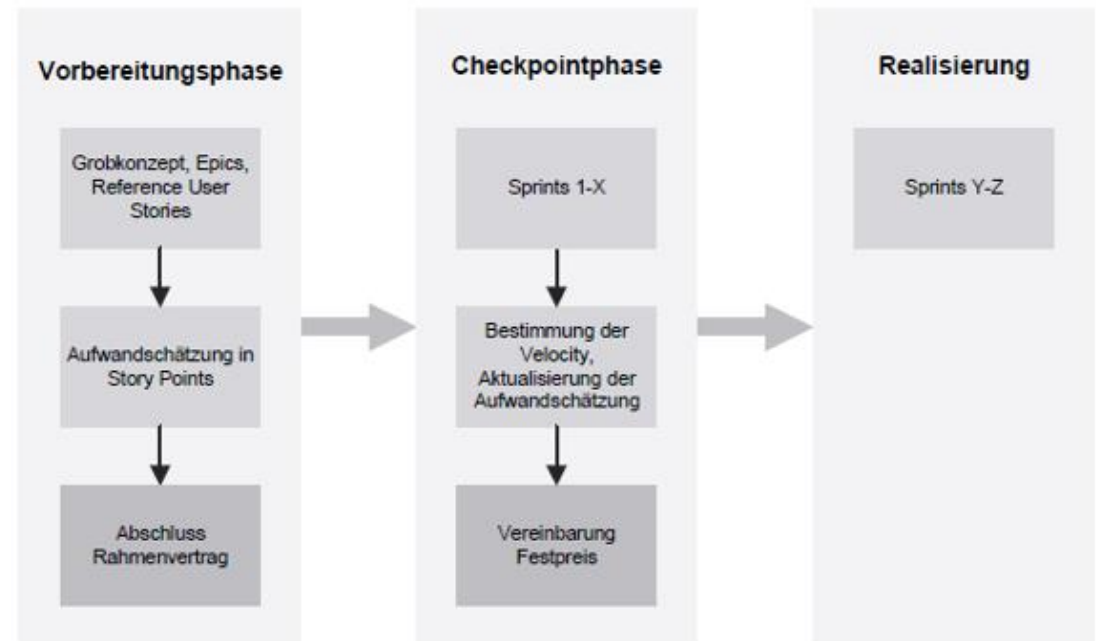
- Budgetlimit/Festpreis
- Fixed Preis, Fixed Scope
- Vergütung nach Aufwand
- Fixed Profit als Alternative
- Phasenbasierte Beauftragung
- Bonus- und Strafklauseln
- ggfs. Besonderheiten
- etc.

Budgetlimit

- Lauffähiges Produkt nach jedem Zyklus
- Werkvertrag mit Festpreis
- Einsatzart:
 - Agile Softwareentwicklung
- Budget:
 - Festgesetzt
 - Entwicklung bis Budget aufgebraucht
- Endprodukt:
 - Nicht unbedingt fest definiert
 - Ggfs. weitere Beauftragung

Agiler Festpreis

- Für die gesamten Anforderungen eines Projekts wird ein verbindlicher Gesamtpreis vereinbart
- Vorteile:
 - Auftraggeber hat Budgetsicherheit
 - Auftragnehmer hat Gewinnaussicht
- Nachteile:
 - Spezifikation vor Auftragsvergabe
 - Änderungen als Ergänzungsverträge
 - Unsicherheit wg. versteckter Mängel
 - Überhöhte Sicherheitspreisaufschläge



Fixed Price, Fixed Scope

- Anforderungspaket nach Festpreis
- Kaufvertrag oder Werkvertrag
- Einsatzart:
 - Agile Softwareentwicklung
 - Traditionelle Softwareentwicklung
- Budget oder Ziele:
 - Festgesetzt
- Endprodukt:
 - Festgesetzt

Vergütung nach Aufwand

- Alternative zu fixem Budget
- Werkvertrag oder Dienstvertrag
- Einsatzart:
 - Agile Softwareentwicklung
 - Ggf. traditionelle Softwareentwicklung
- Budget:
 - Nicht definiert
- Endprodukt:
 - Kein Überblick über Endumfang

Fixed Profit als Alternative

- Nach Budget nur noch kostendeckende Stundensätze
- Werkvertrag, Teilung des Vergütungsrisikos
- Einsatzart:
 - Agile Softwareentwicklung
 - Traditionelle Softwareentwicklung
- Budget:
 - Abschätzbar
 - Kosten hängen vom Aufwand ab
 - Gewinn des Auftragnehmers bleibt fix
- Endprodukt:
 - (theoretisch) festgesetzt

Phasenbasierte Beauftragung

- Lauffähiges Produkt zum Ende jeder Phase
- Risikoreduktion für Kunde auf jwg. Phase
- Kontrolle, Flexibilität durch Auftraggeber
- Einsatzart:
 - Agile Softwareentwicklung
- Budget:
 - Fest je Phase
- Endprodukt:
 - Nicht festgesetzt

Bonus- und Strafklauseln

- Einsatzart:
 - Agile Softwareentwicklung
 - Traditionelle Softwareentwicklung
- Besonderheit:
 - Kein eigenes Vergütungsmodell
 - Strafen für verspätete Abgabe
 - Bonus für vorzeitige Fertigstellung
- Problem:
 - Änderungswünsche durch Auftragnehmer schwierig

Probleme

- Fachbereichskompetenz
- Budgetüberschreitung
- Terminüberschreitung
- Mängelansprüche
- Aufwandschätzung
- ggfs. sonstige Probleme
- etc.

Empfehlung

- Probleme bei Einordnung als reiner Werkvertrag
- Rahmenvertrag
 - Einmalige Einigung bzgl. Zusammenarbeit
 - Dienstvertraglich gestaltete Regelungen
 - Werkvertragliche Teilprojektelemente
 - Sollbruchstellen und Exitstrategie
- Dienstvertrag
 - mit eindeutig definierten Prozessen
 - ggfs. Sollbruchstellen und Exitstrategie
 - bessere Steuerungs- und Kontrollmöglichkeiten
 - Einsparmaßnahmen durch schnelle Problemerkennung

Zusammenfassung

- Überblick über
 - Entwicklungsmethoden
 - mögliche Vertragsarten
 - Vergütungsmodelle
- Schwierigkeiten
 - Problemstellungen
 - Aufwandsschätzung
- Empfehlungen

Geschafft!



Fragen ?

Literaturverzeichnis

- [1] <http://www.amendos.de/fachartikel-whitepaper-download/klassisches-versus-agiles-it-projektmanagement-die-wahl-der-richtigen-vorgehensweise.html> abgerufen am 20.11.2017
- [2] Siehe dazu auch den Überblick bei Reichert Jörg, Vertragsfreiheit und agile Softwareentwicklung: vertragstypologische Einordnung agiler Softwareentwicklungsverträge, Saarbrücken 2013, S. 15ff;
- [3] Royce Winston W., Managing the Development of Large Software Systems, in: Proceedings of IEEE Western Electronic Show and Convention, Los Angeles 1970, S. 1–9.
- [4] Boehm Barry, Guidelines for Verifying and Validation Software Requirements and Design Specifications, in: Samet Jonathan (Hrsg.), Proceedings of Euro IFIP 79, Amsterdam 1979, S. 711–719, zit. Boehm, Guidelines for Verifying and Validation, S. 711ff.
- [5] Siehe zur Einordnung von Phasenmodellen in den Beschaffungsablauf auch Schreiber Josef, Beschaffung von Informatikmitteln: Submissionsverfahren, Pflichtenheft, Evaluation, 5. A., Bern 2015, S. 20ff.
- [6] Informatiksteuerungsorgan des Bundes ISB (Hrsg.), HERMES 5.1 Referenzhandbuch, 2.A., Bern 2015, <http://www.isb.admin.ch/themen/methoden/01661/01662/index.html?lang=de>, S. 5ff.
- [7] Hoppen Peter/Victor Frank, ITIL – Die IT Infrastructure Library – Möglichkeiten, Nutzen und Anwendungsfälle in IT-Verträgen, CR 2008, S. 199–204, S. 201ff.
- [8] <https://planit.legal/blog/de/agile-projekte-agile-vertraege/> abgerufen am 20.11.2017
- [9] Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., et al. Manifesto for agile software development, 2001. URL: <http://agilemanifesto.org> abgerufen am 08.01.2017
- [10] Pichler, R., Scrum: agiles Projektmanagement erfolgreich einsetzen, dpunkt. Verlag, 2013.

- [11] Highsmith, Jim, Agile Software Development Ecosystems. Addison-Wesley, 2002.
- [12] Thewalt, S. Der Softwareerstellungsvertrag nach der Schuldrechtsreform: Rechtsnatur, Leistungsbestimmung und Mängelhaftung. Rhombos-Verlag, 2004.
- [13] http://www.evb-it.de/pages/frame_a.html abgerufen am 05.01.2018
- [14] Witte, A. Agiles Programmieren und § 651 BGB. ITRB 2010, 44.
- [15] Sarre, F., Juristisches IT-Projektmanagement. München, Ludwig-Maximilians-Universität, 2014.
- [16] Wirtschaftslexikon, Gabler: Werkvertrag, 2016: <http://wirtschaftslexikon.gabler.de/Archiv/3791/werkvertrag-v7.html> abgerufen am 23.12.2017
- [17] Koch, F., Agile Softwareentwicklung - Dokumentation, Qualitätssicherung und Kundenmitwirkung. ITRB 2010, 114.
- [18] Sprau: Palandt - Bürgerliches Gesetzbuch, 2009.
- [19] Söbbing, T., Agile Projekte in der IT-rechtlichen Praxis. Der IT-Rechtsberater, 2014, Verlag Dr. Otto Schmidt, 214-219.
- [20] Opelt, A., Gloger, B., Pfarl, W., und Mittermayr, R. Der agile Festpreis: Leitfaden für wirklich erfolgreiche IT-Projekt-Verträge. Carl Hanser Verlag GmbH Co KG, 2014.
- [21] Schwaber, K., Sutherland, J., The Scrum Guide, Version 1.2, 2013, online verfügbar unter <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf#zoom=100> (deutsche Übersetzung: <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-DE.pdf>), S. 3ff;
- [22] <https://www.heise.de/developer/artikel/Standard-fuer-automatisierte-Function-Point-Analyse-2067044.html> abgerufen am 22.11.2017
- [23] <http://blog-it-recht.de/2016/08/11/vertragsgestaltung-bei-agiler-softwareerstellung-teil/> abgerufen am 21.11.2017