



# RECHTSFRAGEN DER AGILEN SOFTWARE-ENTWICKLUNG

12. Österreichischer IT-Rechtstag  
Jan Pohle, Rechtsanwalt | Partner

**April 2018**

# Agenda

- Einführung
- Agile Projektmethoden
- Herausforderungen aus rechtlicher Sicht
- Hinweise zur Vertragsgestaltung



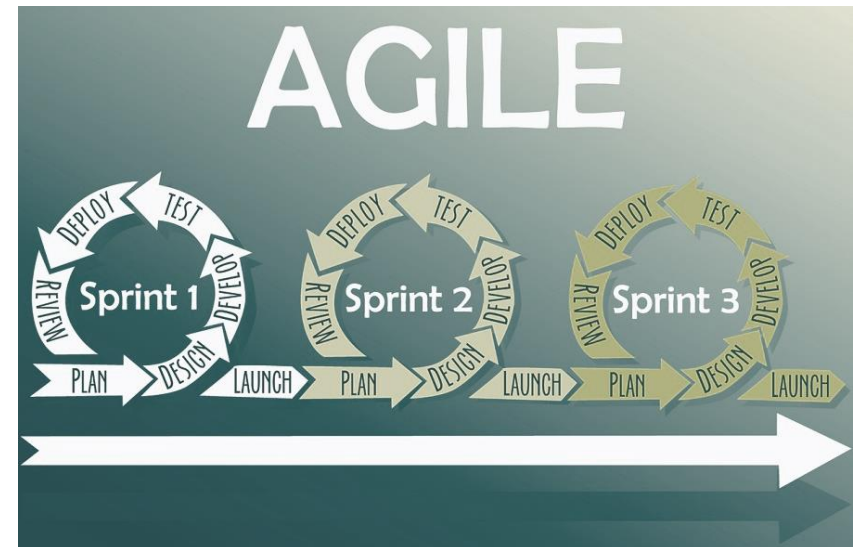
# Einführung



# Was ist Agile Software-Entwicklung?

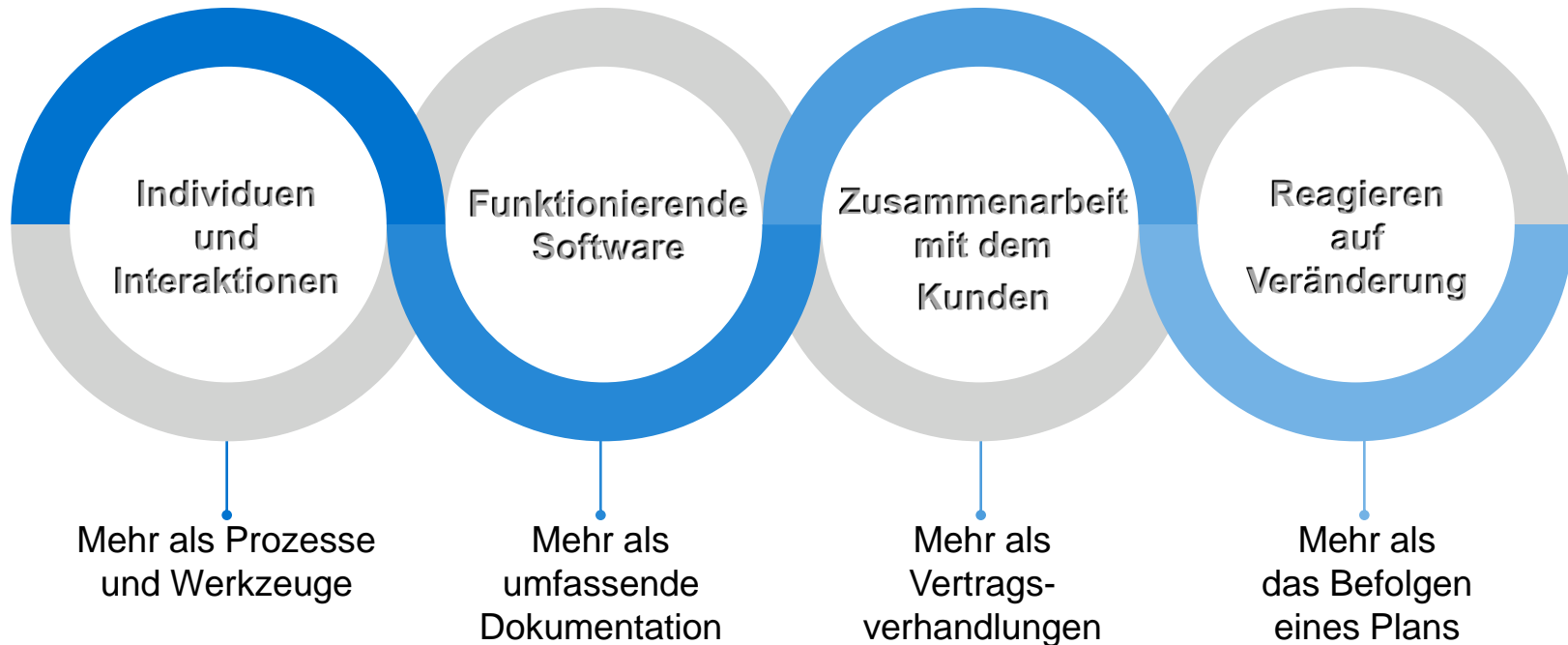
Agile Softwareentwicklung ist:

- ein **Oberbegriff** für eine Gruppe agiler (Projektmanagement-)Methoden im Rahmen der Software-Entwicklung
- jeweils basierend auf einer **iterativen und inkrementellen Vorgehensweise**
- bei denen sich **Anforderungen und Lösungen** durch Selbstorganisation in bereichsübergreifenden Teams **entwickeln**



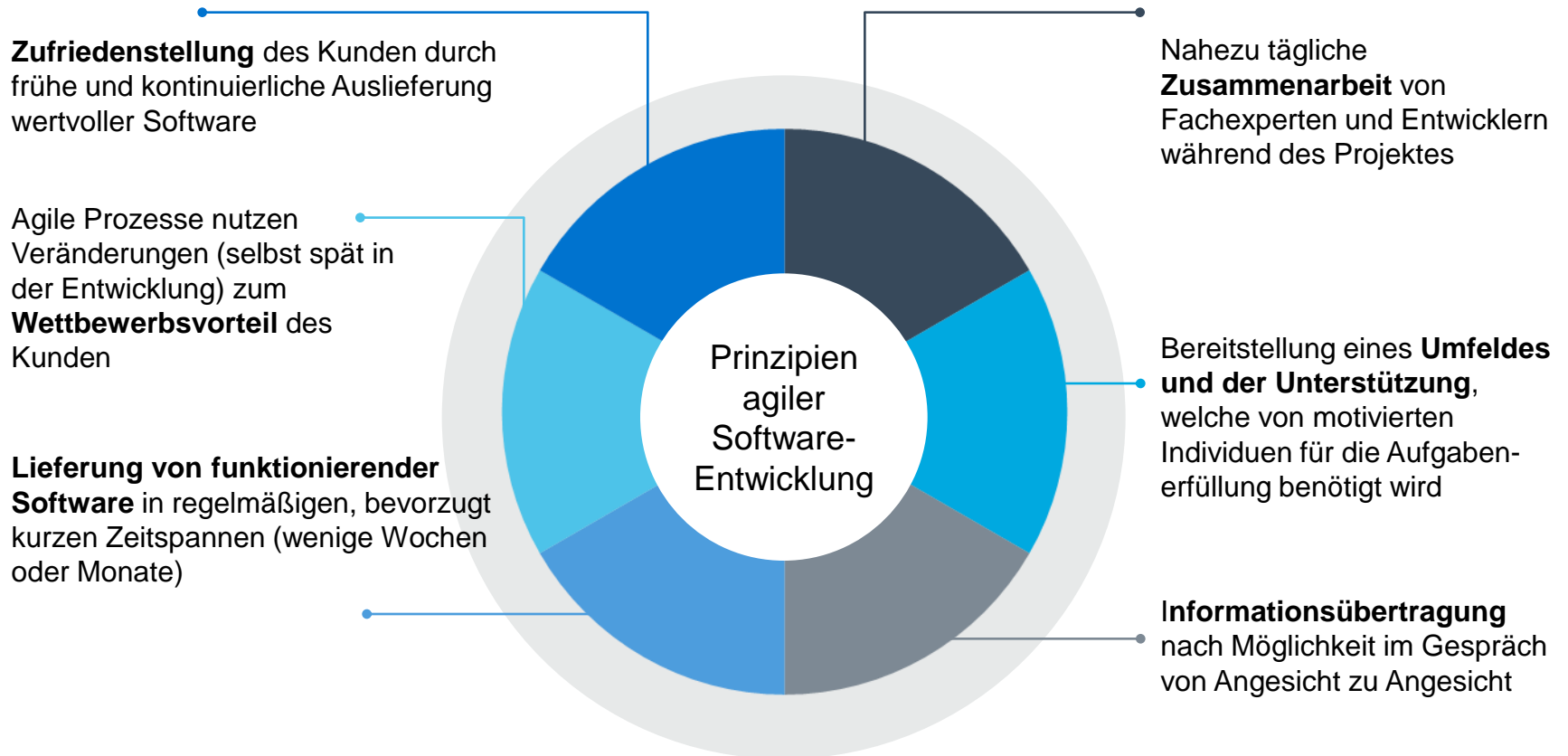
# Manifest für Agile Software-Entwicklung

Wir erschließen bessere Wege, Software zu entwickeln, indem wir es selbst tun und anderen dabei helfen. Durch diese Tätigkeit haben wir diese Werte zu schätzen gelernt:



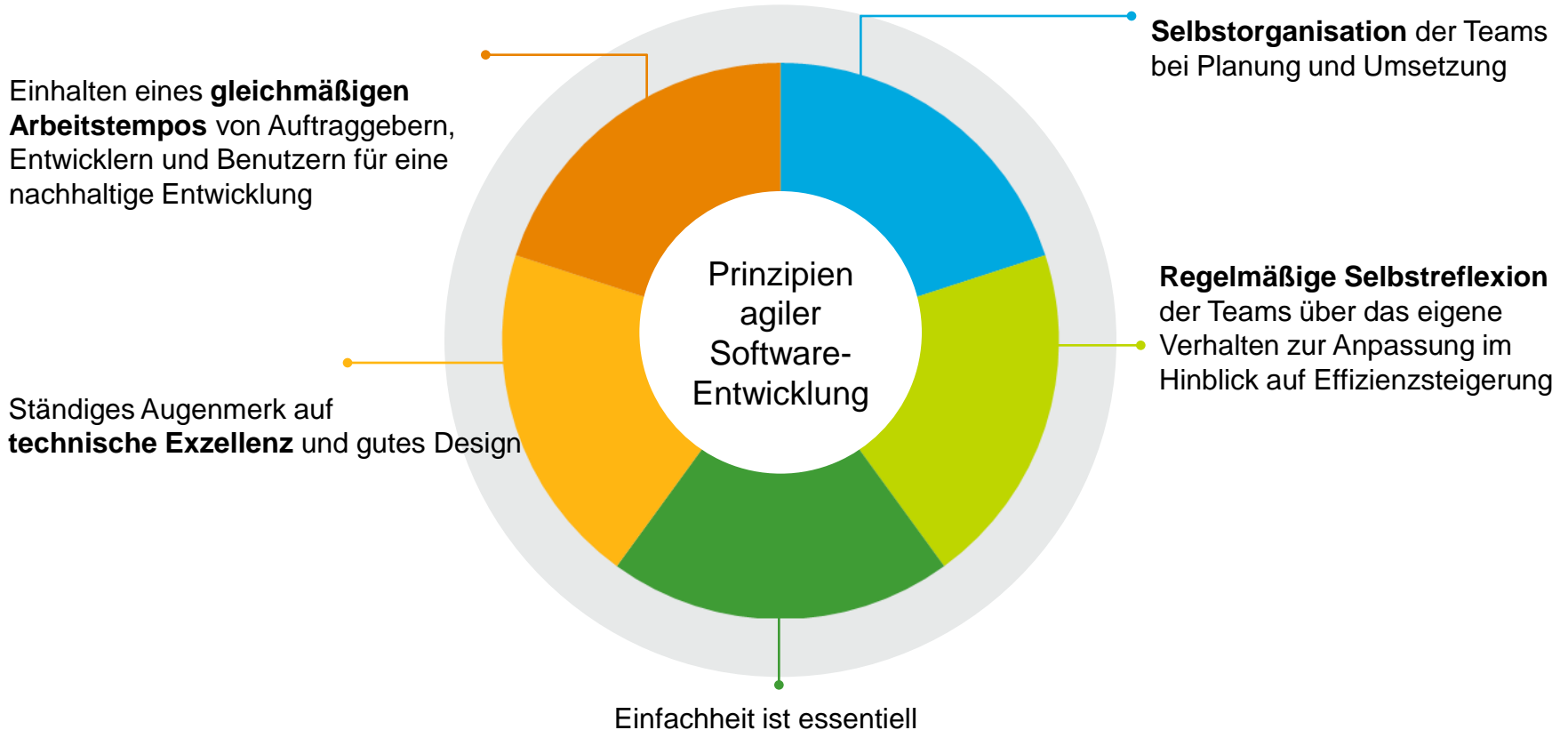
Das heißt, obwohl wir die Werte auf der rechten (unteren) Seite wichtig finden, schätzen wir die Werte auf der linken (oberen) Seite höher ein

# Prinzipien Agiler Software-Entwicklung



Als wichtigster Fortschrittsmaßstab gilt die Funktionsfähigkeit der Software

# Prinzipien Agiler Software-Entwicklung



Der Übergang zwischen Prinzipien und Methoden ist fließend.



# Charakteristika und Treiber

## Charakteristika

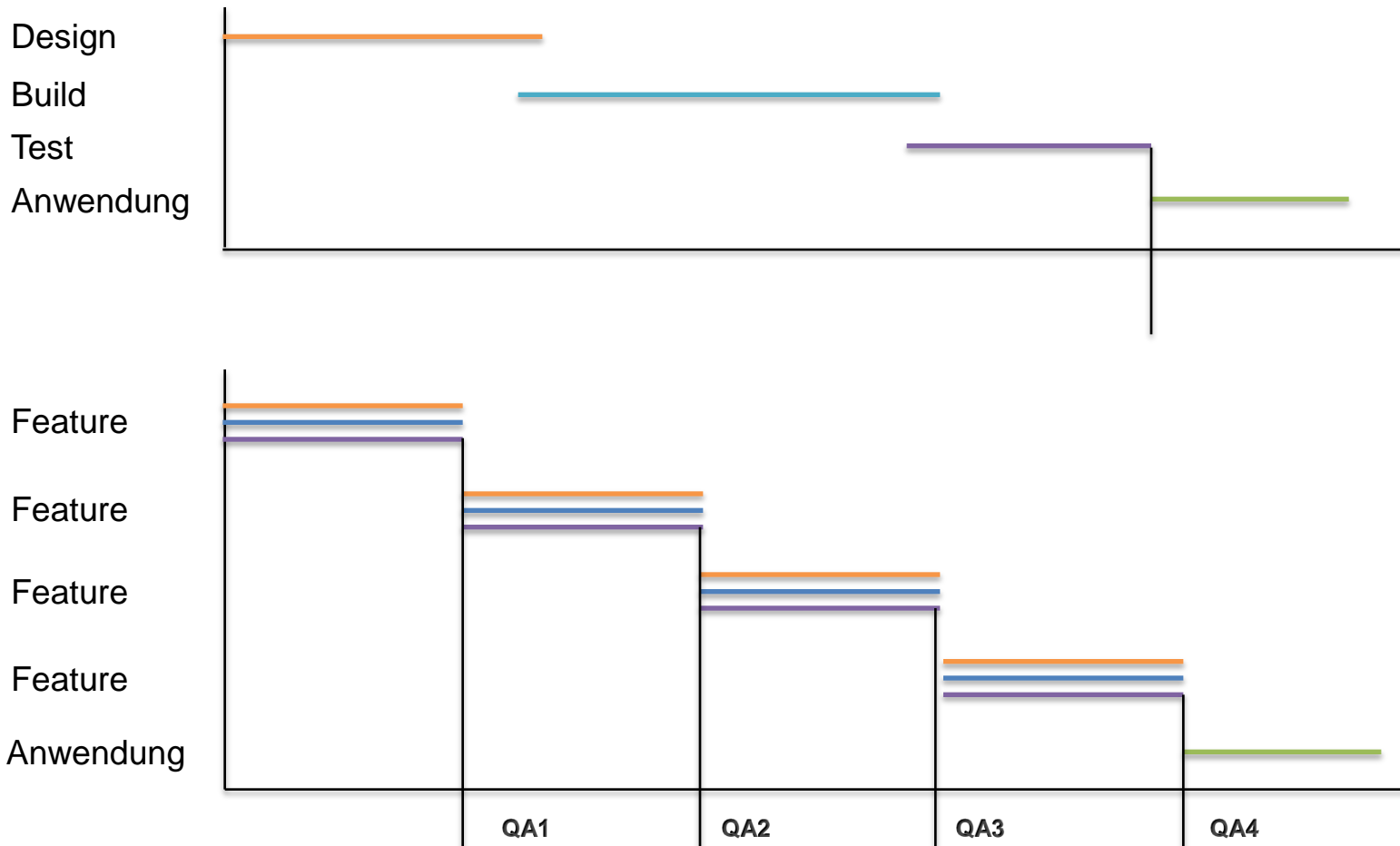
- Allgemeine Beschreibung der Produkt "Vision"
- Zusammenfassung der gewünschten Softwarefunktionalitäten in (dynamischen) User Stories
- Iterativer Softwareentwicklungsprozess
- Zeitnahe Zyklen zwecks Analyse der Anforderungen, des Designs, der Implementierung und des Testing
- Andauernde Einbeziehung des Kunden in den Softwareentwicklungsprozess
- Zeitnahe Code Reviews
- Begrenzte Projektdokumentation durch den Service Provider

## Treiber

- Kaufmännische Erwägungen
- Technische Überlegungen



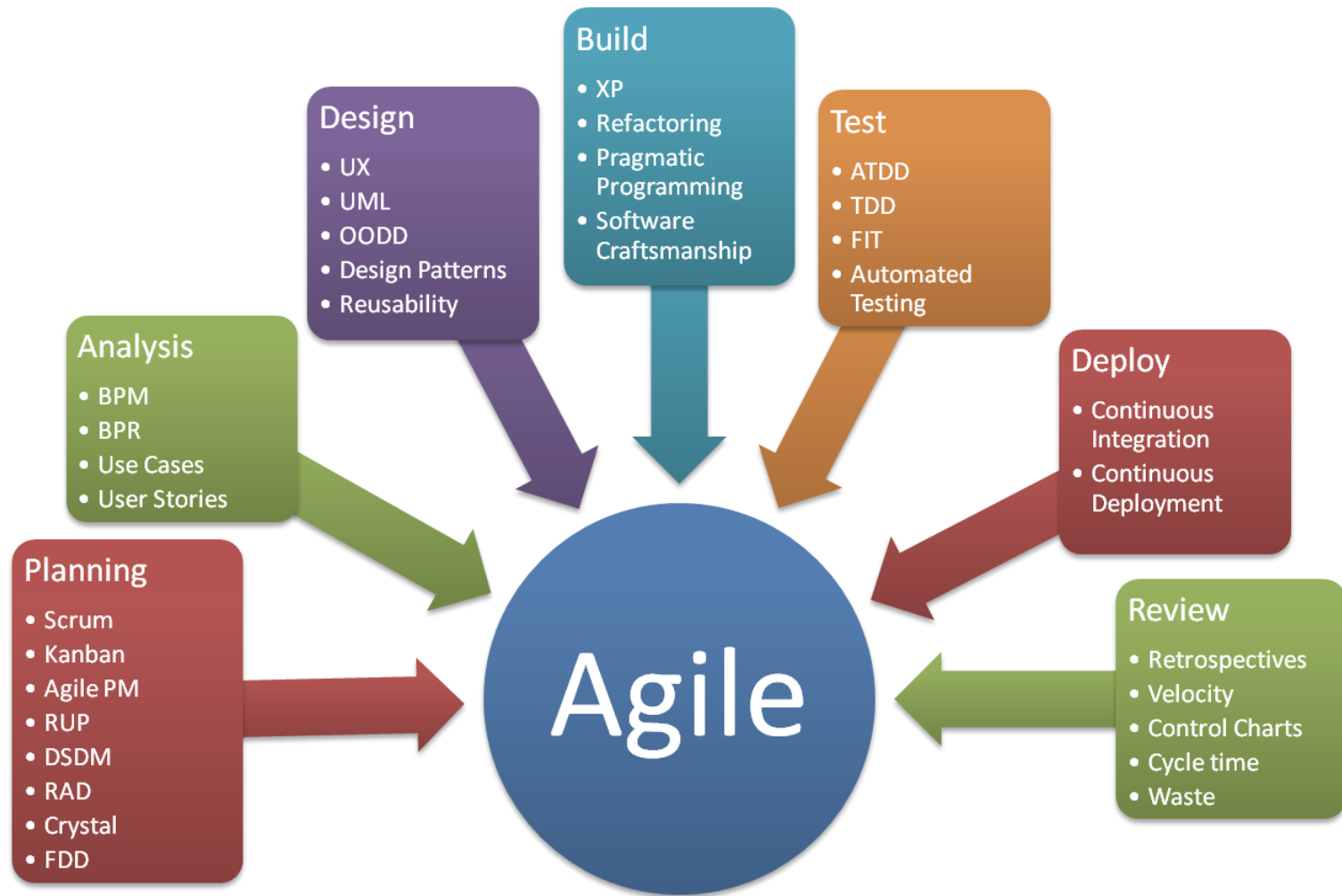
# Unterschiede zwischen Wasserfall & Agiler Entwicklungsmethodik



# Agile Projektmethoden

# Agile Entwicklungsmethoden

## Überblick und Elemente



### Release

Lorem ipsum dolor sit amet, consetetur adip iscing elitr.



100100  
001101  
101001



### Presentation

Lorem ipsum dolor sit amet, consetetur adip iscing elitr.

### Development

Lorem ipsum dolor sit amet, consetetur adip iscing elitr.



### Burndown Chart

Lorem ipsum dolor sit amet, consetetur adip iscing elitr.



### Architecture

Lorem ipsum dolor sit amet, consetetur adip iscing elitr.



### Debugging

Lorem ipsum dolor sit amet, consetetur adip iscing elitr.



### Sprint Time

Lorem ipsum dolor sit amet, consetetur adip iscing elitr.



### Sprint Review

Lorem ipsum dolor sit amet, consetetur adip iscing elitr.



### Brainstorm

Lorem ipsum dolor sit amet, consetetur adip iscing elitr.



### Team

Lorem ipsum dolor sit amet, consetetur adip iscing elitr.



### Sprint

Lorem ipsum dolor sit amet, consetetur adip iscing elitr.



### Scrum Master

Lorem ipsum dolor sit amet, consetetur adip iscing elitr.



### Documentation

Lorem ipsum dolor sit amet, consetetur adip iscing elitr.



# Scrum-Grundlagen

---

## Scrum ist

- die meistgenutzte agile (Projektmanagement-)Methode (85%)
  - ein Rahmenwerk, innerhalb dessen Menschen komplexe adaptive Aufgabenstellungen angehen können, und durch das sie in die Lage versetzt werden, produktiv und kreativ Produkte mit höchstmöglichem Wert auszuliefern
- 

---

## Scrum ist

- leichtgewichtig
  - einfach zu verstehen
  - schwierig zu meistern
- 

---

## Scrum ist

- empirisch
  - iterativ
  - inkrementell
-

# Scrum-Grundlagen

## Die Regeln

Scrum-Regeln beschreiben



### Erfahrungssatz:

Infolge großer Komplexität von Projekten sind wesentliche Anforderungen und Lösungsansätze zu Beginn unklar

# Grundlagen zu Scrum

## Rollen innerhalb von Scrum

### Verantwortlicher Entscheider

- für Eigenschaften und den wirtschaftlichen Erfolg des Projekts
- für Management des Product Backlog

### Coach des Entwicklungsteams

- Verantwortlich für das Gelingen von Scrum
- Einführung und Kontrolle der Scrum-Regeln, Behebung von (kommunikations-) Störungen und Hindernissen

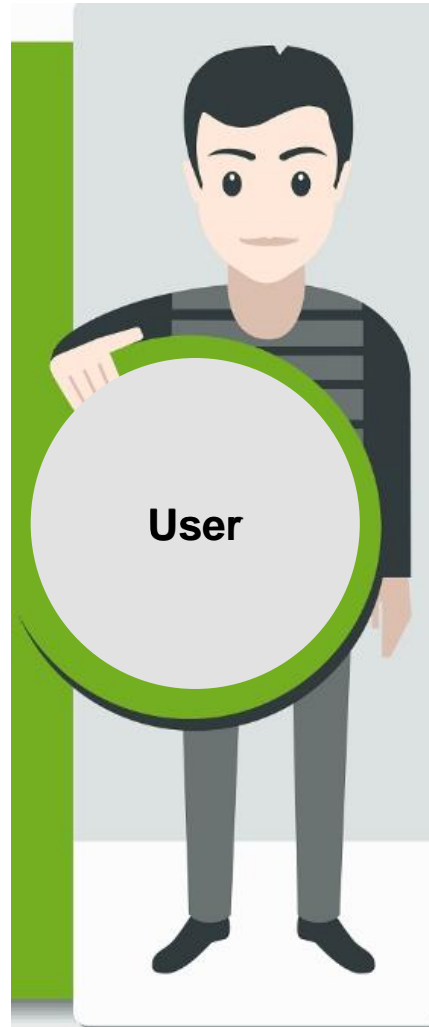
Delivery-Verantwortung hinsichtlich des Produktinkrements in der vom Project Owner gewünschten Reihenfolge - selbstorganisierend und interdisziplinär





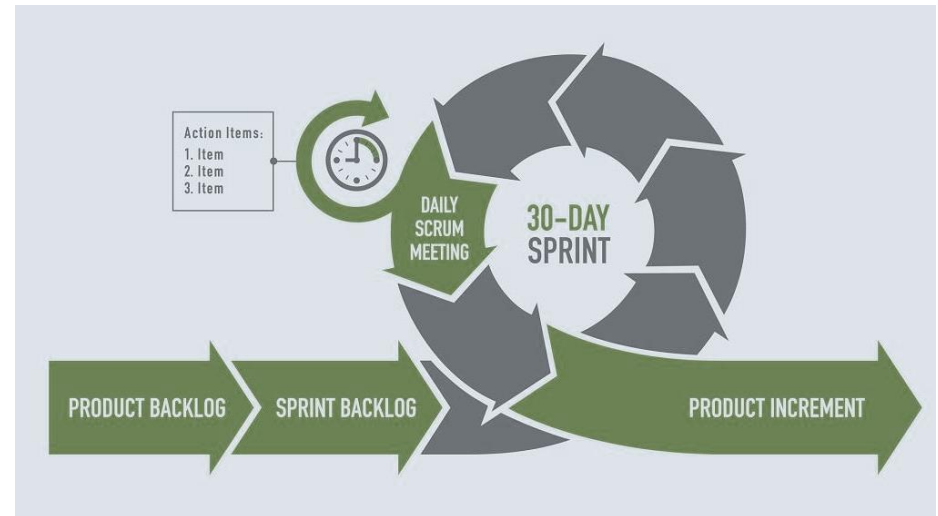
# Grundlagen zu Scrum

## Rollen außerhalb von Scrum



# Grundlagen zu Scrum Sprint

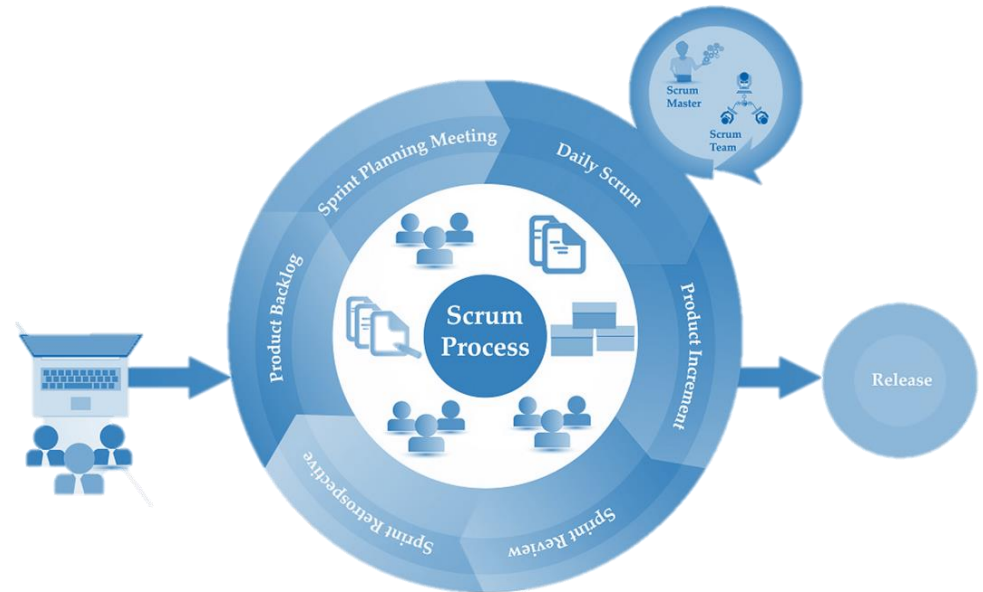
- Sprint = Arbeitsabschnitt: Herstellung eines (funktions-) fertigen (releasefähigen) Produktinkrement ("Done")
- Beginn mit Sprint Planning
- Ende mit Sprint Review und Sprint Retrospektive
- Dauer 4 (bis 6) Wochen – oder bis das Ziel erreicht ist
- Jeder Sprint hat einen Plan und ein (Entwicklungs-)Ziel
- Keine Änderungen im Sprint, die das Sprintziel gefährden
- Änderungen sind zw. Project Owner und Sprint-Team auszuhandeln



# Grundlagen zu Scrum

## Ereignisse (Aktivitäten) im Sprint

- **Sprint Planning:** Max. 4-stündige Festlegung des **Sprint-Ziels**, d.h. des
  - Was (Arbeitsinhalte – Product-Backlog) durch Product Owner und Entwicklungsteam und des
  - Wie der Erstellung des Produktinkrements durch Entwicklungsteam im Sprint
- **Daily Scrum:** Tägliches ca. 15 minütiges Treffen des Entwicklungsteams zum Info-Austausch (Status, Hindernisse, Lösungen usw.) – nicht erledigte Fragen gehen an den Scrum Master
- **Sprint Review:** Überprüfung des Ergebnisses des Sprint auf Zielerreichung unter Beteiligung des Entwicklungsteams, Product Owner, Stakeholdern d. Kunden
- **Sprint Retrospective:** Überprüfung der Arbeitsweise des Entwicklungsteams zu Ende des Sprints zwecks Steigerung von Effizienz und Effektivität
- **Product-Backlog Refinement:** Fortwährende Verfeinerung des Product Backlog durch Product Owner und Entwicklungsteam (einschließlich Schätzungen)



# Grundlagen zu Scrum

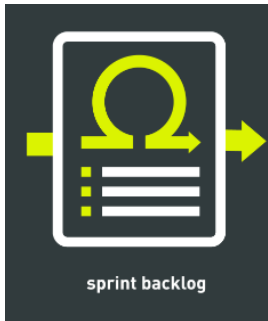
## Artefakte und Techniken



**Product-Backlog:** Dynamische und stets fortentwickelte Auflistung der Anforderungen an des Produkt verantwortlich gepflegt durch Product Owner (Features, Funktionalitäten, Verbesserungen, Fehlerbehebungen)

**User Stories:** Technik zur Beschreibung der (fachlichen) Anforderungen aus Sicht des End-Users zwecks Formulierung der Product Backlog Einträge

**Release Backlog:** Zuordnung der User Stories zu einem bestimmten Release und Priorisierung der einzelnen Releases



**Sprint Backlog:** Menge der für den Sprint ausgewählten Product-Backlog Einträge sowie dynamischer, stets fortentwickelter Plan zur Lieferung des fertigen Produktinkrements ("Done") und Erreichung des Sprint-Ziels

**Taskboard:** Technik zur Visualisierung des Sprint Backlogs



**(Product) Increment:** Summe aller innerhalb eines aktuellen abgearbeiteten Product Backlog Einträge und dem Resultat der Inkremente aller früheren Sprints

# Grundlagen zu Scrum

## Definition of Done

- Definition der Fertigstellungskriterien
- Bezogen auf die Arbeitsergebnisse einer Iteration/eines Sprints (gem. Product-Backlog bzw. Sprint-Backlog)
- Kann funktionale Fertigstellungs-kriterien enthalten (Testfälle, Kriterien für Benutzerakzeptanz)
- Regelinhalt auch Integrationstests, Release Notes, Fortschreibung der Dokumentation



# Scrum – User Stories

## (Fachliche) Anforderungsbeschreibung



- Als [spez. Nutzer]
- will ich [Funktion o. Eigenschaft]
- damit [Nutzen]

*Als Brand Manager, möchte ich die Möglichkeit haben, den Twitter-Verlauf des Unternehmens unmittelbar in MS-Outlook zu sehen, damit ich rechtzeitig negative Publicity entdecken kann.*

# Scrum – Release Backlog

1 Tag

4 Stunden

5 Tage

2 Stunden

8 Tage

3 Tage

4 Stunden

3 Tage

4 Stunden

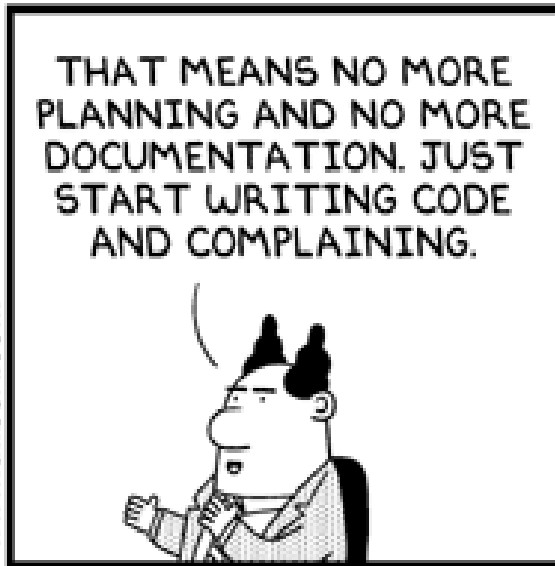
**Geschätzter Arbeitsaufwand = 22 Tage**

# Herausforderungen aus rechtlicher Sicht





www.dilbert.com  
scottadams@aol.com



© 2007 Scott Adams, Inc./Dist. by UFS, Inc.



© Scott Adams, Inc./Dist. by UFS, Inc.

# Vertragstypologische Einordnung und Gestaltungsoptionen

- Relevanz:
  - AGB-Kontrolle
  - Ges. Gewährleistungsregime
- Werkvertrag
  - Regelmäßig Erfolg geschuldet (User Stories/Epics)
  - Sprint Reviews = Teilabnahmen
- Dienstvertrag
  - Keine anfängliche Definition des Werkes (Product-Backlog ist dynamisch)
  - Vergütung nach Time & Material
- Typengemischter Vertrag – Dienstvertrag mit werkvertraglichen Elementen
- Rahmenvertrag mit dienst- und werkvertraglichen Einzelverträgen
- Gesellschaftsvertrag (GbR – JV)



# Vertragstypologische Einordnung und Gestaltungsoptionen

- LG Wiesbaden – Werkvertrag
  - Konzeptzionshoheit beim AG
  - Ausführungsverantwortung des AN
  - Softwarerealisation als Erfolg geschuldet
  - Vergütung T&M nicht werkvertragsschädlich
- OLG Frankfurt – Fragestellung nicht entscheidungserheblich
- Vertragspraxis
  - Werk- o. Dienstvertrag sinnvoll o. geboten?
  - Konkrete Ausgestaltung maßgeblich



# Vorprojekt

- Scrum = unmittelbarer Projektstart
- Vorprojekt
  - Definition der (fachlichen) Anforderungen des Auftraggebers als Grundlage
    - als Grundlage für User Stories und/oder Epics
    - als Grundlage für Definition des vertraglichen Leistungsgegenstands
    - als Anknüpfung für Vergütungsregelung
  - Mitwirkung des Auftraggebers
  - Festlegung der IT-Infrastruktur
  - Systemintegration - Schnittstellen
  - Festlegung der Test- und Produktionsumgebung
  - Datenmigration
  - Dokumentationsumfang



# Herausforderungen aus rechtlicher Sicht

- Kundenzufriedenheit vor Vertragsverhandlung – Projekt "ohne" Vertrag?
- Beschreibung von Leistung und Gegenleistung - Flexibilität im Projekt vs. Planbarkeit von Ergebnis und Ausgaben
  - Verzicht auf initiales Spezifikationsprofil (Lasten-/Pflichtenheft) - Erarbeitung und Aktualisierung des Anforderungsprofils im Projektlauf
  - (Agiler) Festpreis vs. Time & Material
- (Entwickler- und Anwender-) Dokumentation der Software als Leistungsbestandteil
- (Laufende) Zusammenarbeit der Parteien
  - Definition/Festlegung von Rollen
  - Regeln der Zusammenarbeit
  - Mitwirkung des Kunden (!)
  - Abgrenzung zur Arbeitnehmerüberlassung



# Herausforderungen aus rechtlicher Sicht

- Zeitplanung und Festlegung von Lieferzyklen
- Abnahmekonzept
  - Gesamtabnahme
  - Teilabnahmen
  - Freigaben
- Gegenstand von Abnahme und Gewährleistung
  - Produkt Backlog
  - Lauf von Gewährleistungsfristen
- Rechteeinräumung
  - Quelcodezugriff
  - OSS-Komponenten
  - Sonderproblem: Gemischte Teams
- Management von Dissens - Eskalation



# Fokus: Leistungsbeschreibung

- Agile Softwareentwicklung basiert auf einem iterativen und inkrementellen Entwicklungsprozess, der der andauernde Zusammenarbeit der Beteiligten zur Erreichung des Projektziels bedarf
- Leistungsbeschreibungen beinhalten daher Prozess-beschreibungen innerhalb einzelner Projektzyklen
  - Product Vision
  - Agile Team
  - Product Backlog
  - Sprints
  - Development Sprints
  - Development Sprint Testing and Definition of Done
  - Release Planning
  - Deployment Sprints



# Fokus: Preisvereinbarung

## Aufwandsabhängige Vergütung (Time & Material)

- Problem: Kostenkontrolle
- Lösungsansätze
  - Gesamtbudget – Budgetobergrenze
  - Bonus-/Malusregelungen

## Agiler Festpreis

- Problem: Agilität d. Projekts (Scope, Timing, Ressourcen)
- Agiles Spezifikationsprojekt (Projektantragsphase, Spezifikationsphase, Angebotsphase, Realisierungsphase)
- MVP - Minimum Viable Product
  - Produkt mit Mindestanforderung (Sprint Zero)
    - Erweiterungen in gesonderten Sprints
- Problem: Annahmen des AG

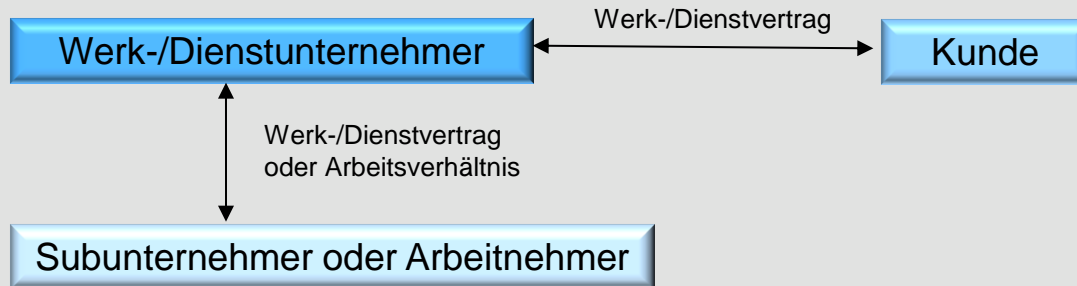
## Pay per Sprint (Mini Gewerke)



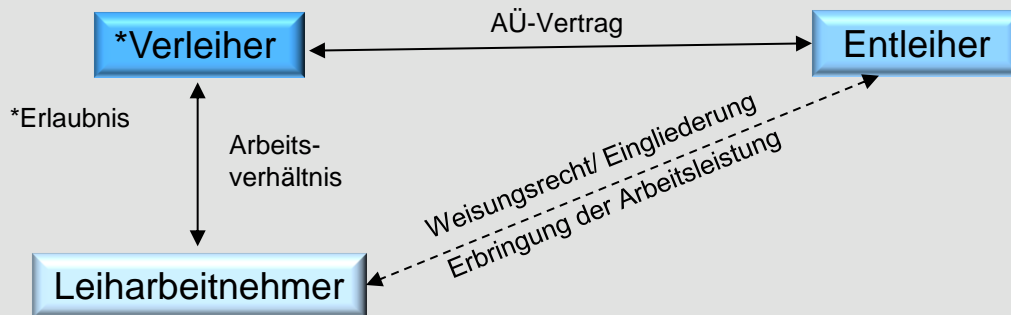


# Problem: Arbeitnehmerüberlassung

## Werk- und Dienstverträge



## Erlaubnispflichtige Arbeitnehmerüberlassung



- Tatsächliche Durchführung entscheidend, § 12 Abs. 1 S. 2 AÜG n.F.
- "Definition" des **Arbeitsvertrages** in § 611 a BGB
- Abgrenzungshilfe" bzgl. Werk- und Dienstverträge

# Arbeitnehmerüberlassung

- *Arbeitnehmer werden zur Arbeitsleistung überlassen,*
  - *wenn sie in die Arbeitsorganisation des Entleihers eingegliedert sind und*
  - *seinen Weisungen unterliegen (§ 1 Abs. 1 S. 2 AÜG)*
- Bei Agilen Software Projekten sollen u.U. gerade fremdes und eigenes Personal eng im Betrieb des Kunden zusammenarbeiten!
- Abgrenzung zum Dienst- oder Werkvertrag fließend
- Auswirkungen der Änderung AÜG
  - Wegfall der Vorratsüberlassungserlaubnis
  - Kennzeichnungspflicht (§ 1 Abs. 1 S. 5 AÜG)
  - Konkretisierungspflicht (§ 1 Abs. 1 S. 6 AÜG)
  - Überlassungshöchstdauer von 18 Monaten (§ 1 Abs. 1b AÜG)



# Hinweise zur Vertragsgestaltung

# Herausforderungen "agiler" Vertragsgestaltung

- Wahrung der rechten Balance zwischen Flexibilität und Sicherheit (u.a. "Minimum Viable Product")
- Festlegung des Leistungsinhalts
  - Zentrale Kundenanforderungen (User Stories)
  - Dokumentationspflichten
- Prozessbeschreibung innerhalb Projektzyklen (u.a. Inhalte und Laufzeiten von Sprints)
- Festlegung von Rollen im Projekt
  - Allokierung der Projektverantwortung
  - Ausformulierung der Rollen (Aufgaben und Befugnisse)
  - Rechte und Pflichten innerhalb von Teams



# Herausforderungen "agiler" Vertragsgestaltung

- Regeln für die Zusammenarbeit
  - DailyScrum-Teilnahme als Hauptpflicht des AG?
- Mitwirkungspflichten des Kunden
  - Was wird wann in welchem Umfang erwartet?
  - Festlegung benötigter Skills
- Regelungen zur Rechtezuordnung
- Preisvereinbarung
- Abnahme und Abnahmetests
  - Gegenstand von Abnahmen
  - Gesamt-, Teilabnahmen und Freigaben
  - Grundsätzliches zu Abnahmetest
- Abbruchszenarien
  - Konsequenzen für geleistete Zahlungen
  - (Weiter-)Nutzung von Teilleistungen (Deliverables)
  - Schadensersatz





# (Neuere) Literatur

- Schwaber, Ken und Sutherland, Jeff, Der Scrum Guide  
<https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-German.pdf>
- Pieper, Fritz-Ulli und Rock, Stefan, Agile Verträge, 1. Auflage, 2017
- Sarre, Frank, Kritische Schnittstellen zwischen der Projektmethodik SCRUM und der juristischen Vertragsgestaltung, Computer und Recht 2018, 198 ff.
- Kühn, Philipp M. und Ehlenz, Nikolaus, Agile Werkverträge mit Scrum – Konkrete Vertragsgestaltungs- und Formulierungsvorschläge, Computer und Recht 2018, 139 ff.
- Welkenbach, Christian, Scrum auf dem Prüfstand der Rechtsprechung – Lehren für die Vertragsgestaltung, Computer und Recht 2017, 639 ff.
- Ernst, Stefan, Agile Softwareprogrammierung, Computer und Recht 2017, 285 ff.
- Schneider, Jochen, Vergütungsanspruch bei agilem Softwareprojekt, Anmerkung zu OLG Frankfurt v. 17.08.2017, 5 U 152/16, IT-Rechtsberater 2017, 231
- Pohle, Jan, Nutzungsrechte bei agiler Softwareprogrammierung, Festschrift für Axel-Artur Wandtke zum 70. Geburtstag, 2013, S. 557 ff.

