

**MODUL 426**

SOFTWARE MIT AGILEN

METHODEN ENTWICKELN

Reto Glarner

Projektmethode Scrum Teil 1

# Lernziele

› HZ 1.1 Sie kennen eine agile Methode zur SW Entwicklung

› HZ 1.3 Sie kennen Werte und Prinzipien von agilen Methoden

› HZ 1.4 Sie kennen die Abgrenzung von linearen und agilen Methoden

› HZ 2.1 Sie kennen den Ablauf und die Bestandteile eines Releasezyklus’

# Beispiel

Sie gehen in ein teures Restaurant und bestellen das Feinschmeckermenu. Nach kurzer Zeit folgt ein Teller dem anderen mit feinsten Speisen. Wie wurden diese zubereitet? Gibt es mehrere Köche? War es ein Koch, welcher nur für Sie gekocht hat? Gab es für jeden Teller einen anderen Koch? Wurden vorgefertigte Saucen verwendet oder alles frisch zubereitet?

Sie wissen es nicht. Je nachdem mit welcher (Projekt) Methode in der Küche gearbeitet wird, desto besser, schneller, kostengünstiger und zuverlässiger ist das Resultat.

Auch bei der Softwareentwicklung sollten bewährte Methoden verwendet werden!

Unsere Betrachtungen in diesem Modul basieren stets auf folgenden Annahmen:

› Es wird in einem Team mit mindestens 3 Personen gearbeitet

› Unsere zeitlichen Ressourcen sind beschränkt

› Der Kunde weiss was er will (hier unterscheiden sich Theorie und Praxis leider oft)

Wieso ist die Methode so wichtig bzw. was kann alles in einem Projekt schiefgehen?

› Das Projekt dauert länger als geplant

› Das Projekt wird teurer als geplant (wer bezahlt das..)

› Das Projekt wird komplexer oder gar unrealisierbar

Was sind die möglichen Ursachen für ein Scheitern?

› „Schlechte“ Planung

› „Schlechte“ Umsetzung

› Zu wenig Kontrollmechanismen

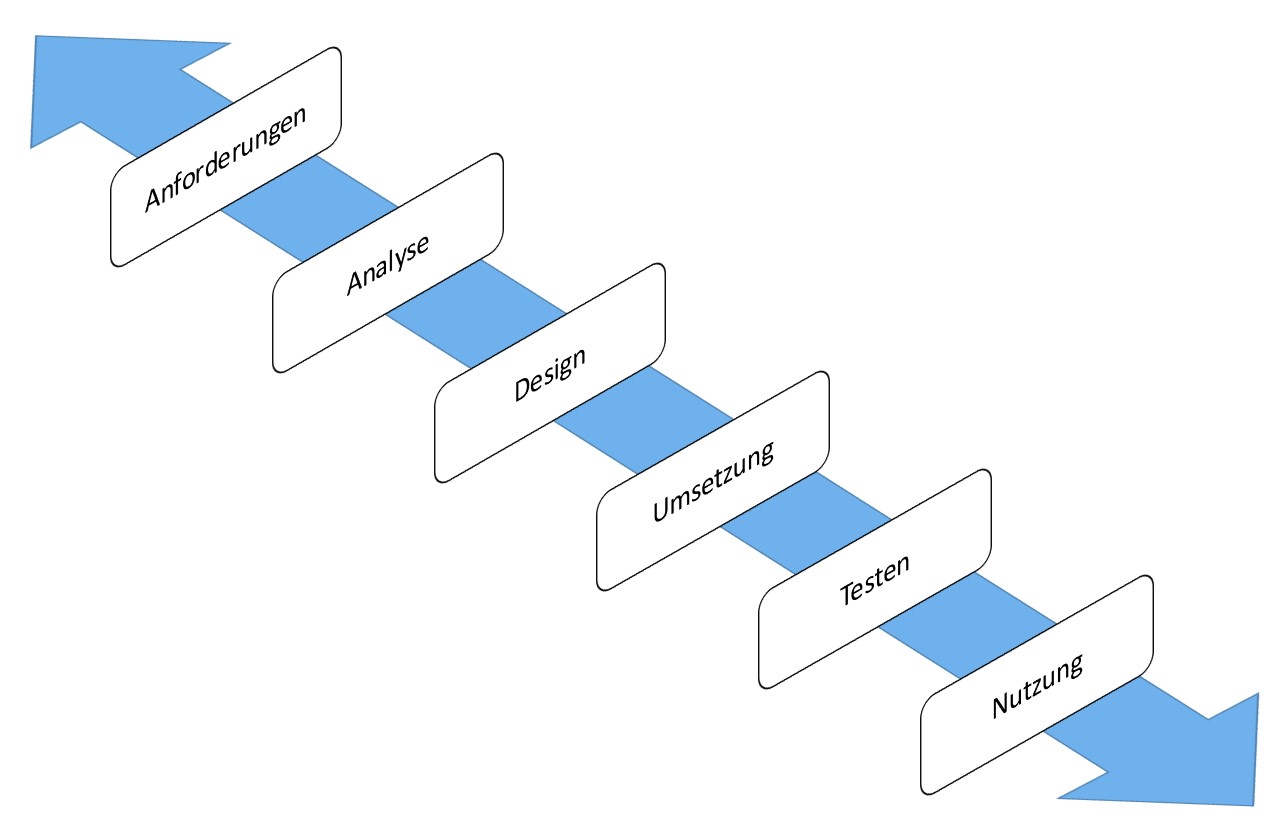
› Personalwechsel

› Streitigkeiten im Team (Kommunikation, Hierarchie, Zuständigkeiten, Transparenz, ...)

# Lineare Methoden

Obwohl heute nur noch wenig eingesetzt, muss das Wasserfallmodell als schlechtes Beispiel einer Projektmethode herhalten. Beim Wasserfallmodell gibt es einen Ablauf, welcher die verschiedenen Phasen vom Entwurf bis zur Nutzung einer Software abbildet.

Ursprünglich gab es nur einen Weg „nach vorne“. Später wurde das Modell angepasst, so dass man auch wieder in die vorherige Phase zurück kann bei Misserfolg. (Daher Pfeile in beiden Richtungen)



Die grösste Problematik bei diesem Modell ist dessen Starrheit:

› Es gibt nur eine Testphase innerhalb der gesamten Projektlaufzeit!

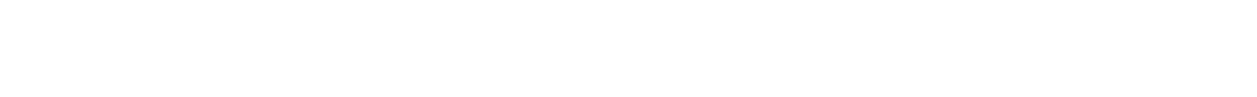
› Anpassungen oder Kundenwünsche während des Projekts bedeuten „gehe zurück zum Anfang“

› Man kann das Resultat erst am Schluss sehen

Daraus ergeben sich erhöhte Risiken, welche für die Informatikwelt bis heute einen Imageschaden bedeuten. (zu spät, zu teuer...)

# Agile Methoden am Beispiel Scrum

Der Grund liegt auf der Hand, wieso sich das Wasserfallmodell nicht bis heute behaupten konnte. Der Drang zur flexibleren und produktorientierteren Softwareentwicklung hat die sogenannten Agilen Methoden entstehen lassen. Es gibt einen ganzen Katalog an agilen Methoden, ja gar ein agiles Manifest mit 12 Prinzipien.



Video

Agile Softwareentwicklung in 5 Minuten

-

YouTube.mp4

im Ordner

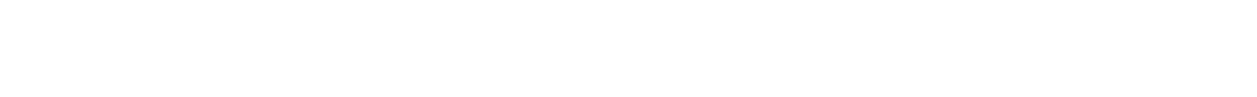
ab426

-

01

Dateien

, Quelle: Youtube



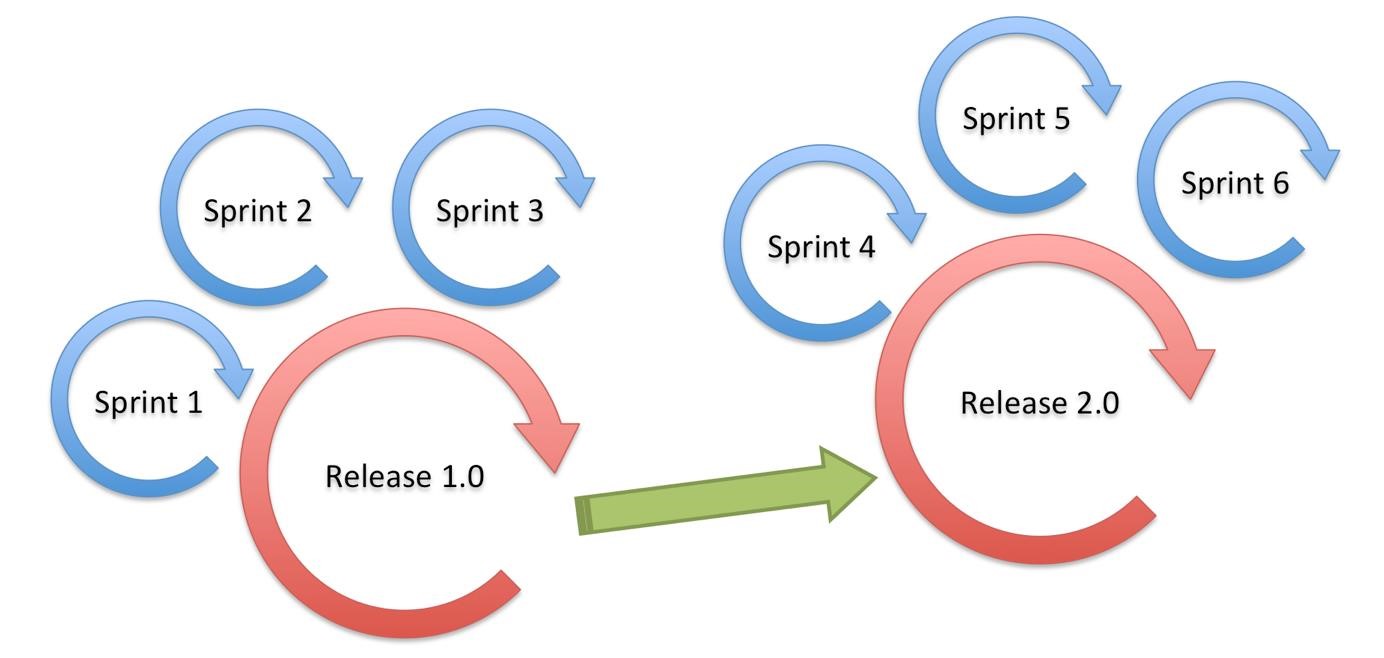
[http://agilemanifesto.org/iso/de](http://agilemanifesto.org/iso/de/)

[/](http://agilemanifesto.org/iso/de/)

# Scrum im Detail

Ein Scrum Projekt beginnt mit den Anforderungen an die auszuliefernde Software. Diese Anforderungen werden als "**User Stories**" festgehalten.

Bei der Erarbeitung der Software werden iterativ sogenannte **Sprints** durchgeführt. Ein Sprint hat eine vordefinierte Länge von ca. 2-8 Wochen. **Nach jedem Sprint ist die Software praktisch lauffähig!** Nach 3-10 Sprints wird eine Major Version erstellt.

Der Aufbau eines Scrum Projekts besteht aus 3 Elementen:

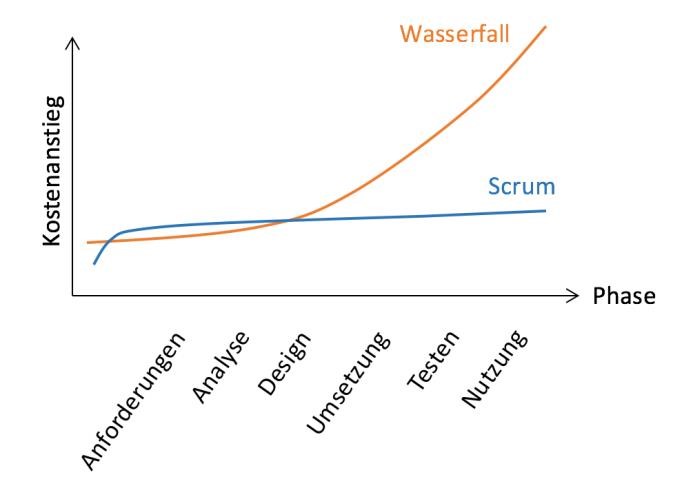
|  |  |
| --- | --- |
| Rollen (Roles) | Es gibt drei innere und drei äussere  Rollen, generell sind alle Personen im Projekt an eine Rolle gebunden und erhalten dementsprechend ihren eigenen Aufgabenbereich. |
| Artefakte (Artifacts) | Darunter fallen alle Dokumente, Grafiken und Auflistungen des Projekts (administrativer Output) |

Besprechungen (Ceremonies, Events) Jede Aktivität hat vordefinierte

Besprechungen, damit wird die Durchgängigkeit des Projekts gewährleistet, ohne dieses aber zu förmlich werden zu lassen. (Ich habe absichtlich den Begriff "Sitzung" nicht gewählt!)

Die 3 Elemente werden hier und auf dem AB02 detailliert behandelt.

Aus Kostensicht, sollte durch die agile Vorgehensweise ein Vorteil gegenüber dem Wasserfallmodel entstehen. Änderungen mitten in der Umsetzungsphase fallen weniger ins Gewicht, weil diese mittels der Sprints einfacher ins laufende Projekt eingeführt werden können. (Prinzip der Agilität)



# Rollen

In Scrum werden sogenannte Rollen im Team verteilt. Es gibt aber im Gegensatz zu anderen Methoden viel weniger davon, nämlich nur 3 Rollen: (innere Rollen)

**Rolle Aufgaben / Verantwortung Tätigkeiten**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Product Owner | Anforderungen erstellen | Erstellt Stories im Backlog | | |
| «Gehört» Produkt und ist verantwortlich |  |  |  |
|  |  |
|  |  |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ScrumMaster | Planung von Projekt | Sprints, Refinements, Retros, Schätzung von Stories | | |
| «Schnittstelle» von PO und Entwicklung |  |  |  |
|  |  |
|  |  |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entwickler | Umsetzung von Projekt | Stories von Sprint umsetzten | | |
|  |  |  |  |
|  |  |
|  |  |
|  |  |

Aufgabe: Füllen Sie die obigen Felder nach eigener Recherche aus und vergleichen Sie die Resultate mit der Lehrperson.

Nicht behandelte Rollen: Kunde, Benutzer, Management (= äussere Rollen)

# Das Scrum Projekt – Produktplanung

Für die Produktplanung gibt es lediglich ein zentrales Artefakt in Scrum, dieses heisst **Product Backlog**. Diese listenförmige Aufzählung ist der *einzige Weg*, wie es eine Funktionalität oder Anforderung schafft, im Produkt berücksichtigt zu werden!

*Wer bestimmt den Inhalt des Product Backlog?*

Zu Beginn kommt sicher der Kunde (oder der interne produktverantwortliche Manager) mit einer Liste von Ideen bzw. Anforderungen. **Diese werden ausschliesslich durch den Product Owner verwaltet!** Input kann von allen Seiten (Product Owner, Entwicklungsteam, User usw.) kommen, Anpassungen an der Liste aber werden nur durch den Product Owner vorgenommen. (Allenfalls nach Absprache mit dem Kunden der die Anforderung eingebracht hat)

Ein Product Backlog kann theoretisch jederzeit erweitert werden. Diese Tatsache unterscheidet agile Projekte von anderen.

*Wie sieht so eine Anforderung aus?*

Die Anforderungen werden als sogenannte **User Stories** erfasst. Diese Stories werden in **kurzer Textform** verfasst. Inputs, welche Funktionen beschreiben, sollten aus folgenden 3 Punkten bestehen:

(1) Wer möchte (2) was tun und (3) warum

In den Product Backlog könnten zudem Bugs oder nicht-funktionale Anforderungen aufgenommen werden. (Performance, Usability, Security, usw.)

Nebst der Beschreibung werden auch (durch das Entwicklerteam) geschätzte **Aufwände** für jedes **PBI** geführt. Derjenige, der einen Eintrag liefert, muss gleichzeitig die

**Wichtigkeit** für das Produkt angeben, diese wird später für die Priorisierung benötigt.

*Wann gilt eine Anforderung als erledigt?*

Zu jeder Anforderung gehört eine Liste von Kriterien, welche definiert wann die Anforderung als korrekt erledigt gilt. (Acceptance Criteria) Nur anhand von dieser Angabe kann später der Entwickler prüfen, ob er seine Arbeit vollständig und korrekt beendet hat. Es liegt in der Verantwortung des Product Owners diese Kriterienliste für jede Anforderung zu prüfen bevor ein Sprint anfängt bzw. die Programmierung beginnt!

Studieren Sie das Beispiel - PBI im Anhang!

*Was bedeutet die Reihenfolge der Product Backlog Liste?*

Allen PBI werden Prioritäten zugeordnet. Die wichtigste Anforderung steht zuoberst auf der Liste usw.

Wie werden die Prioritäten definiert? (5 Minuten Einzelarbeit, versuchen Sie ohne Internet zu arbeiten)

Stories, die gemacht werden müssen um andere Stories auszuführen können.

# Aufgabe 1 30 Minuten

Ein sauberer Product Backlog ist das A und O für das weitere Vorgehen, deshalb ist an dieser Stelle ein wenig Übung notwendig. Die Aufgabe wird in Gruppen von 3-4 Personen erledigt.

**Schreiben Sie einen Produkt Backlog für eine professionelle Videokonferenz Software.**

Stichworte:

› Kontaktliste

› Geschlossene „Räume“

› Ton- und Bildübertragung

› 1:1 sowie 1:n Sitzungen

› Dokumentaustausch

› Unterschiedliche Bandbreiten pro Teilnehmer

› Unterbrüche in der Verbindung

› Plattformunabhängigkeit

› Verschlüsselung

› ...

Sie müssen nicht alle Stichworte in Ihr Produkt einfliessen lassen. Keine Angst, Sie müssen es auch nichts davon umsetzen später. 😉

Der Umfang sollte mindestens 8 User Stories mit allen notwendigen Angaben enthalten. (Titel, Akzeptanz, Aufwandschätzung)

Die Liste muss priorisiert werden, die Prioritäten müssen *erklärbar* festgehalten werden. (Siehe 4 Punkte von oben)

Kümmern Sie sich nicht zu viel um die Darstellung! Am besten verwenden Sie eine

Tabelle zur Darstellung. Wir werden die „Scrum-Werkzeuge“ später auswählen und anwenden.

Die Resultate werden am Ende im Plenum kurz besprochen, notieren Sie dazu alle

Namen der Teilnehmer im Dokument und speichern Sie dieses auf den Klassenshare!

# Aufgabe 2 5 Minuten

Markieren Sie die bereits behandelten Themen auf dem Modulleitfaden Seite 2 in der Grafik!

# Aufgabe 3 (freiwillig) 5 Minuten

Schlagen Sie die anderen agilen Methoden, welche im Video vorkommen nach und vergleichen diese mit Scrum.

Anhang: Beispiel Product Backlog

# Item eines ERP Systems

|  |  |
| --- | --- |
| User Story | Der Sachbearbeiter kann einem Wiederverkäufer Rabatt gewähren, weil dieser grosse Mengen einkauft |
| Feature | Auftragsverwaltung |
| Wirtschaftlicher Nutzen | Mittel |
| Risiken bei der  Umsetzung | klein, Mehrwertsteuer muss korrekt bleiben! |
| Abhängigkeit zu anderen PBI | Auftragspositionen müssen funktionieren |
| Termin für die Fertigstellung | Bis Ende Geschäftsjahr |
| Aufwandschätzung | 40-50 Stunden |
| Akzeptanzkriterien | 1. Einer Position kann kein Rabatt gewährt werden, dies ist das   Standardverhalten (0%) dies wird auf der Rechnung nicht angezeigt.   1. Einer Position kann ein manueller Rabatt gewährt werden. Der Rabatt ist eine ganze Zahl zwischen 1-100 (in Prozent) 2. Einem Kunden kann ein fixer Rabatt gewährt werden, welcher automatisch auf alle Artikel angewandt wird. 3. Einer Kombination Kunde-Artikel kann ein Mengenrabatt gewährt werden. (Bsp: 0-9 Stk = 5%, 10-19 Stk = 10%, ab 20 Stk 15%) 4. Einer Rechnung kann ein Gesamtrabatt zugewiesen werden, welcher sich auf den Gesamtbetrag der Rechnung bezieht. |