

# Advanced Software Engineering Agile PM / Scrum



Scrum > Inhalt

Agilität

Prozess

Prinzipien

Rollen

**User Stories** 

Agiles Schätzen



# Scrum > Agilität > Was ist Agilität?

- Agil: flink, gewandt, beweglich
- Vergleiche andere Berufe: Ärzte und Krankenschwestern in einem Krankenhaus → Visite = Daily Scrum?
- Lean Production/Schlanke Produktion → der Buffered Production entgegengesetzt
- Leichtgewichtigkeit → Details wenn sie benötigt werden
- Veränderungen schnell erkennen
- Schnelle Reaktion auf Veränderungen (Anforderungen, Rahmenbedingungen, ...)
- Kontinuierliche Verbesserung
- Ständiges Evaluieren



# Scrum > Agilität > Definition

 Agilität ist die Fähigkeit einer Organisation, flexibel, aktiv, anpassungsfähig und mit Initiative in Zeiten des Wandels und von Unsicherheit zu agieren. Der Schlüssel für Agilität liegt im Wesen der Organisation und nicht so sehr in dem, was sie tut. Agilität entstand als Reaktion auf langsame, bürokratische Organisationen, um veränderten Marktbedingungen zu begegnen.



# Scrum > Agilität > Das Agile Manifest

Individuen und Interaktionen	Gelten mehr als	Prozesse und Tools
Funktionierende Programme	Gelten mehr als	Ausführliche Dokumentation
Die stetige Zusammenarbeit mit dem Kunden	Steht über	Verträgen
Der Mut und die Offenheit für Veränderungen	Steht über dem	Befolgen eines festgelegten Plans

http://www.agilemanifesto.org



Scrum > Prozess

Agilität

Prozess

Prinzipien

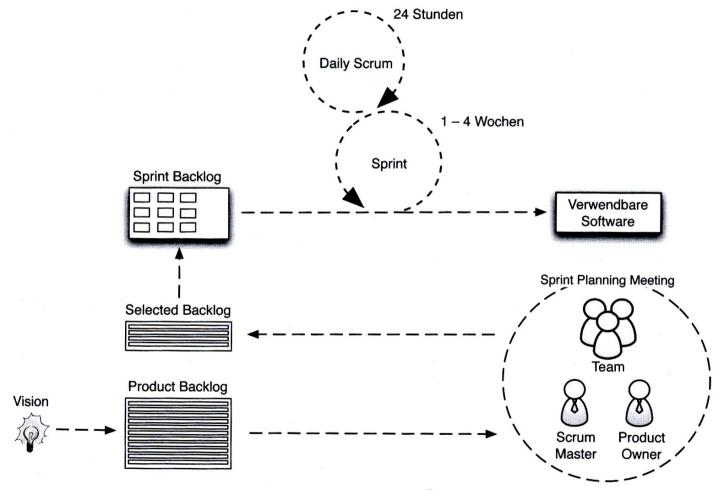
Rollen

**User Stories** 

Agiles Schätzen



## Scrum > Prozess



Quelle: Wirdemann, Ralf: Scrum mit User Stories, Hanser, 2009



## Scrum > Prozess > Scrum Team

- Rollen
  - Team
  - Scrum Master
  - Product Owner
- Teilen sich sämtliche Managementverantwortung
- Kein Chef, kein Projektleiter, keine Vorschrift zur Arbeitsweise
- Verantwortungsteilung, Respekt und Vertrauen bei der Zusammenarbeit



## Scrum > Prozess > Vision

- Idee für das zu entwickelnde Produkt
- Product Owner in Zusammenarbeit mit dem Kunden
  - Wer wird das Produkt aus welchem Grund kaufen?
  - Welche Kundenbedürfnisse bedient das Produkt?
  - Worin unterscheidet sich das Produkt von bereits existierenden Produkten?



## Scrum > Prozess > Product Backlog

- Ausgehend von der Vision werden Anforderungen erstellt
- Das Product Backlog ist eine Liste aller Anforderungen
- Typischerweise sind die Anforderungen als User Stories definiert
- User Stories sind priorisiert und nach ihrer Größe geschätzt



# Scrum > Prozess > Sprint Planning Meeting

- Vorbereitung des anstehenden Sprint
  - Team, Product Owner und Scrum Master
- Sprint Planning 1
  - Sprint-Ziel festlegen
  - Product Owner stellt die User Stories im Backlog nach Priorität vor
  - Team entscheidet, wie viele Stories im Sprint erledigt werden können
     → Selected Backlog
- Sprint Planning 2
  - Software Design, Zerlegung der Stories in Einzeltasks → Sprint Backlog
  - Das Sprint Backlog visualisiert den Fortschritt des Projekts innerhalb des Sprints
  - Das Team aktualisiert das Sprint Backlog täglich



# Scrum > Prozess > Sprints

- Entwicklungsphasen fester Länge
  - Typischerweise min. 1 bis max. 4 Wochen
- Iterationen in denen das Team an der Umsetzung der Aufgaben im Sprint Backlog arbeitet
  - Phasen der Ruhe und Ordung (keine ungeplante Arbeit)
  - Scrum Master sorgt für die Einhaltung der Regeln
  - Team arbeitet selbstorganisiert
  - Team synchronisiert sich im *Daily Scrum*
- Ergebnis des Sprint ist *verwendbare Software* 
  - Potentiell lieferbare Software
  - Stand des Projekts bezogen auf den Gesamtplan



## Scrum > Prozess > Sprints

- "normale" Sprints
  - Liefern "nur" verwendbare Software
  - Software könnte ausgeliefert werden, muss aber nicht
- Release Sprints
  - Beinhalten zudem die konkrete Lieferung → Deployment
  - Qualität des Sprint-Ergebnis ist die selbst
  - Jedoch werden in der Regel mehr Personen benötigt
    - Administratoren müssen Systeme bereitstellen, Ausfallzeiten planen und kommunizieren
    - Handbücher müssen geschrieben werden
    - Werbeaktionen müssen geplant werden

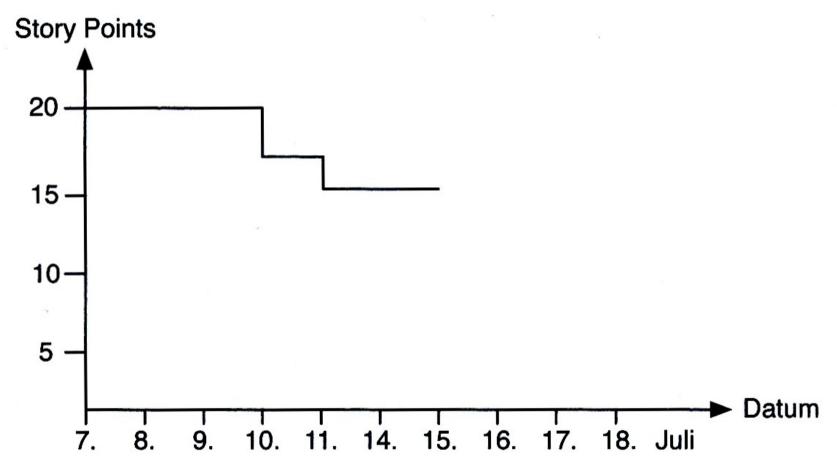


## Scrum > Prozess > Daily Scrums

- Tägliches Standup-Meeting des Teams zu fester Zeit und Dauer
  - Typischerweise morgens
- Synchronisierung des Teams über die Aufgaben im Sprint Backlog
  - Was wurde am Vortag erreicht?
  - Was ist für den laufenden Tag geplant?
  - Gibt es aktuelle Probleme?
- Stand des aktuellen Sprint
  - Scrum Master aktualisiert das sog. *Burndown Chart*
- Feedbackschleifen
  - Transparenz und Sichtbarkeit → Korrekturmöglichkeiten



## Scrum > Prozess > Burndown Chart



Quelle: Wirdemann, Ralf: Scrum mit User Stories, Hanser, 2009



## Scrum > Prozess > Sprint Review

- Öffentliche Präsentation im Sprint umgesetzter User Stories
- Team, Product Owner, Scrum Master, Kunde, zukünftige Anwender, Geschäftsführer, Abteilungsleiter, Entwickler anderer Projekte
- Diskussion des Sprint-Ergebnis
- Vorab Absprache zwischen Scrum Team, wer welche User Storie präsentiert, aber keiner darf sich mehr als 1 Stunde auf das Meeting vorbereiten
- Power Point ist verboten, es wird ausschließlich laufende Software präsentiert
- Entscheidung, ob die Ergebnisse den Erwartungen entsprechen
- Welche Richtung wird weiterhin eingeschlagen? → *Kurs Korrekturen*



# Scrum > Prozess > Sprint Retrospektive

- Das Team beendet den Sprint mit der Retrospektive
  - Diskussion über den eigentlichen Entwicklungsprozess und die Zusammenarbeit des Teams während des Sprint
- Ziel: Kontinuierliche Verbesserung
  - Des Entwicklungsprozesses
  - Der Produktivität des Teams
- Ergebnis: Liste konkreter Verbesserungsmaßnahmen
  - Werden gemeinsam vom Scrum Master und dem Team umgesetzt



Scrum > Prinzipien

Agilität

Prozess

Prinzipien

Rollen

**User Stories** 

Agiles Schätzen



## Scrum > Prinzipien > Transparenz

- Aspekte des Prozesses oder dessen Ergebnisse sichtbar machen
  - Sprint Backlog → Zu Jeder Zeit Stand und Fortschritt im aktuellen Sprint
  - Daily Scrum → Täglicher Einblick in die Arbeit des Teams
  - Sprint Review → Ergebnis des Sprint öffentlich sichtbar machen
- Vor allem aber auch Probleme und Hindernisse erkennen
  - Enthalten Potential für Verbesserungen
  - Steigerung der Produktivität



# Scrum > Prinzipien > Beobachten und Anpassen

- Prozesse, Zusammenarbeit des Teams und produzierte Ergebnisse werden ständig beobachtet
- Unmittelbare Reaktion bei Abweichungen außerhalb des gesetzten Limit
  - Regelmäßige Codereviews, Unit-Tests, Kontinuierliche Integration über einen Buildserver, Software-Metriken
     → bspw. Fehlschlagenden Test unmittelbar korrigieren
  - Sprint Review → bspw. kann der Kunde sofort reagieren, falls die User Experience deutlich hinter den Erwartungen zurückbleiben
  - Product Owner kenn das Geschäftsmodell des Kunden gleicht den Projektverlauf mit den Zielen und der Vision ab (Geschäftswert) → Kurskorrektur über Priorisierung der User Stories im Product Backlog



## Scrum > Prinzipien > Timeboxing

- Managementtechnik: Zeitbudget einer Tätigkeit
  - Das Ende einer Timebox ist fix und kann unter keinen Umständen verschoben werden
  - Der Inhalt (zu erledigende Arbeit) einer Timebox ist variabel
- Timeboxing in Scrum
  - Sprints haben feste Länge (1 bis 4 Wochen)
  - Daily Scrums dürfen nicht länger als bspw. 15min sein
  - Sprint Planning Meetings sind auf 1 Tag begrenzt
  - User Stories mit Forschungscharakter sind timeboxed



# Scrum > Prinzipien > Timeboxing

# Variabler Umfang?

- Was passiert, wenn Aufgaben nicht innerhalb der Timebox erledigt werden?
- Es wird dennoch abgebrochen, bspw. muss das Design einer User Story, das im Sprint Planning 2 nicht fertig wurde im Sprint nachgeholt werden, sobald die Story in die Entwicklung geht

# Timeboxing

- Schafft kurzfristige Ziele
- ist entscheidend für das Abschließen von Aufgaben
- Wird vom Scrum Master implementiert, gecoacht und vorgelebt
- Der Scrum Master hat die Timeboxes immer im Blick und greift ggf. steuernd ein



## Scrum > Prinzipien > Dinge abschließen

- Wann eine Story als abgeschlossen gilt definiert jedes Team für sich mit seiner "Definition of Done"
- Der Kunde kann jederzeit verlangen eine als abgeschlossene Story kommunizierte Story in Produktion zu nehmen
- Verboten: "die Story ist zwar fertig, aber…" → nicht abgeschlossen
- Nicht abgeschlossene User Stories
  - Haben für den Kunden einen Mehrwert von Null!
  - Kommen zurück ins Backlog



## Scrum > Prinzipien > Maximierung von Geschäftswert

- Durch Reduktion der Planungsphase auf das Notwendigste
  - Product Backlog mit priorisierten und nach Größe geschätzten User Stories für 1 bis 3 Sprints
- Durch Priorisierung
  - Stories mit hoher Priorität (hohem Geschäftswert oder auch hohem Risiko) zuerst
  - Auch Tasks einer Story sind priorisiert
- Der Kunde sollt so früh wie möglich mit verwendbarer Software Geld verdienen
  - Nicht erst nach Abschluss des kompletten Projekts
- Der Kunde soll Risiken (insbesondere die Umsetzbarkeit) so früh wie möglich erkennen
  - Das Projekt scheitert bereits in den ersten Sprints, nicht erst nach bspw. einem Jahr



# Scrum > Prinzipien > Teams scheitern nicht

- Akzeptieren, dass Fehler passieren und daraus lernen
  - Bspw. Durch Reduktion der angenommenen Entwicklungsgeschwindigkeit im nächsten Sprint
- Schwierige Situationen erkennen und ermitteln, was unter den Umständen noch erledigt werden kann
  - Bspw. Nach Mitteln und Wegen suchen, wie das Sprint-Ziel noch erreicht werden kann, wenn die Aufgaben doch komplexer sind, als ursprünglich angenommen.



# Scrum > Prinzipien > Ergebnisorientierung

- Nur das Ergebnis zählt
- Umsetzung der Produkt-Vision
  - Die durch Sprints realisiert wird
- Jeder Sprint hat sein eigenes Ergebnis
  - Im Sprint-Ziel festgehalten
- Ergebnis eines Arbeitstages
  - Im Daily Scrum festgelegt



Scrum > Rollen

Agilität

Prozess

Prinzipien

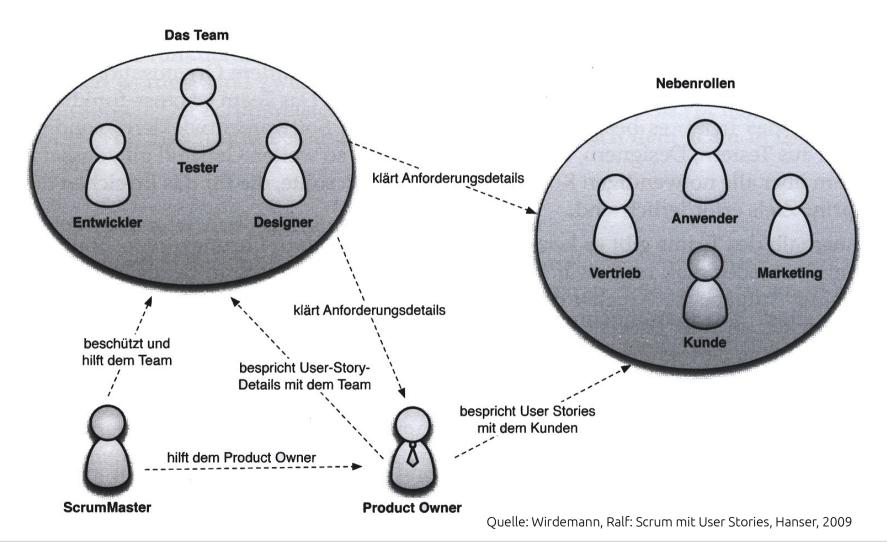
Rollen

**User Stories** 

Agiles Schätzen



## Scrum > Rollen





## Scrum > Rollen

- Rollen sind keine Positionen innerhalb eines Linienmanagements
- 3 Hauptrollen (Scrum Team)
  - Team, Scrum Master, Product Owner
  - Projektverantwortlich
  - Rolle bildet eine Menge von Verantwortlichkeiten
  - Rollen sind disjunkt
    - Ausgewogene Kräfteverteilung
    - Wie könnte ein Scrum Master sein Team im Sprint vor zusätzlichen Anforderungen des Product Owners beschützen, wenn Scrum Master und Product Owner dieselbe Person sind?

#### • Nebenrolle

- Kunde, Vertrieb, Anwender, Marketing
- Nicht beteiligt an der Entwicklung, können diese aber beeinflussen



## Scrum > Rollen > Das Team

- Besteht aus 5 bis 9 plus/minus 2 Personen
  - Mehr Personen erfordert bspw. das Aufteilen in 2 Teams
- Ist cross-funktional besetzt
  - Softwareentwickler, Tester, Designer, Dokumentatoren, ...
  - Das Team sollte über alle Kompetenzen verfügen um das Sprint-Ziel zu erreichen
- Keine Hierarchien
  - Keine Führungspositionen
  - Jeder führt, wenn er der Experte einer bestimmten Aufgabe ist



## Scrum > Rollen > Das Team

- Entwickelt die Software
- Ist für das Erreichen jedes Sprint-Ziels verantwortlich
  - Volle Autorität alle erforderlichen Maßnahmen zu ergreifen, um das Ziel zu erreichen (diese Kompetenz geht demnach weit über traditionell gemanagte Projekte hinaus)
    - Bspw. Berater, Tools, Bücher, Trainings einfordern
- Arbeitet selbstorganisert
  - Hat sich freiwillig auf das Ziel committet → Verantwortungsgefühl
    - Keine Vorschriften, wie lange das Team arbeitet
    - Das Team entscheidet selbst, ob es ein Wochenende durcharbeiten muss, um das Ziel zu erreichen, oder ob am Freitag bereits um 13:00 Uhr Feierabend ist



## Scrum > Rollen > Scrum Master

- 2 Hauptverantwortlichkeiten
  - Einhaltung von Scrum und den zugrunde liegenden Regeln und Prinzipien
  - Für optimale Arbeitsbedingungen für das Team sorgen
  - → Verantwortlich für den Prozess
- Dienstleistender Anführer und Problembeseitiger
  - Servant Leader: nimmt keine Verantwortungen ab, sondern achtet darauf, dass Verantwortungen angenommen werden, indem er auf die Einhaltung von Prinzipien achtet
  - Probleme und Hindernisse (Impediments) erkennen und beseitigen
  - Impediment Backlog



## Scrum > Rollen > Scrum Master

- Impediments
  - Es gibt keinen Product Owner.
  - Das Team braucht mehr Tester.
  - Im Raum ist es viel zu laut.
  - Wir brauchen bessere Pair-Programming-Arbeitsplätze.
  - Der Product Owner hat keine Entscheidungskompetenz.
  - Persönliche Konflikte zwischen Teammitgliedern.
  - ...
- Müssen Scrum Master programmieren können?



## Scrum > Rollen > Product Owner

- Verantwortlich f
   ür das Ergebnis
- Repräsentiert den Kunden und alle Stakeholder
  - Wirtschaftlicher Erfolg
  - Anforderung entgegennehmen, im Product Backlog als User Stories sammeln und priorisieren
  - Besitzt die volle Autorität des Kunden
  - Kennt das Geschäftsmodell des Kunden
  - Zusammen mit dem Kunden der Visionär des Projekts
- Begleitet das Team während des Sprint
  - Klären von Implementierungsdetails
- Bestimmt, wann eine Story fertig ist



## Scrum > Rollen > Nebenrolle Kunde/Stakeholder

- Anwender
  - Absprache mit dem Team → Anwendungsdomäne des Teams besser verstehen (der Product Owner ist kein Anwender)
- Geschäftsführung, Marketing, Vertrieb, Support, ...
- Beeinflussen das Projekt, stellen Anforderungen
  - Jedoch an den Product Owner, nicht an das Team
  - Bei Implementierungsdetails Absprache mit dem Team



## Scrum > User Stories

Agilität

Prozess

Prinzipien

Rollen

**User Stories** 

Agiles Schätzen

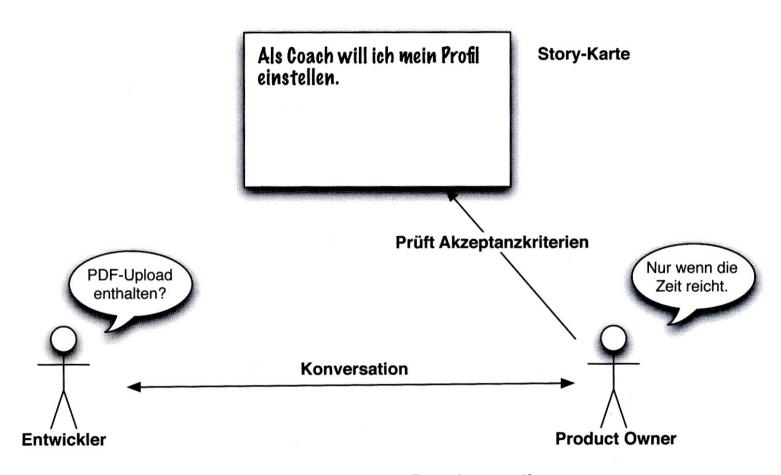


#### Scrum > User Stories > Was sind User Stories?

- Sehr vage und unscharf formulierte Anforderungen aus Benutzersicht
- User Stories akzeptieren, dass Benutzer und Kunden erst relativ spät wissen, was sie wollen. → In der Regel wenn sie erste Teile der Software gesehen haben.
- Eine User Story besitzt einen konkreten und sichtbaren Mehrwert für den Kunden
  - Gut: "Als Benutzer will ich mein Profil einstellen."
  - Schlecht: "Die Applikation soll in Java implementiert sein." → kein Geschäftswert für den Kunden



#### Scrum > User Stories > Bestandteile



Quelle: Wirdemann, Ralf: Scrum mit User Stories, Hanser, 2009



#### Scrum > User Stories > Bestandteile

- Story Karte
  - Drückt das Ziel des Benutzers möglichst klar aus
  - Platzhalter für die spätere Kommunikation
- Konversation zwischen Team und Product Owner
  - Story wird konkret, sobald sie für den anstehenden Sprint eingeplant wird
  - Festlegen von Details
    - Bsp: "Als Benutzer kann ich mein Profil einstellen"
      - PDF Upload implementieren
      - Welche Felder sind Pflicht?
      - Müssen Profile vom Admin freigegeben werden?
- Akzeptanzkriterien
  - "Definition of Done" wann ist die Story fertig? …Rückseite der Story Karte
  - Für Tester und Product Owner wichtig für Akzeptanztests



#### Scrum > User Stories > Muster

### Ausführlich:

- Als <Benutzerrolle> will ich <das Ziel>, so dass <Grund für das Ziel>

#### • Kurzform:

Als <Benutzerrolle> will ich <das Ziel>

# Beispiele:

- Als Kunde will ich nach Anzeigen suchen, um Kontakt mit dem dem Anbieter aufzunehmen
- Als Anbieter will ich Anzeigen einstellen, so dass ich ich von Kunden gefunden werde
- Als ehemaliger Kunde will ich Anbieter bewerten, so dass andere potentielle Kunden eine bessere Entscheidungsgrundlage haben



# Scrum > User Stories > Eigenschaften guter User Stories

-1	Independent	User Stories sollen unabhängig voneinander sein.
Ν	Negotiable	User Stories sollen verhandelbar sein.
V	Valuable	User Stories sollen einen Wert für den Kunden haben.
Ε	Estimatable	User Stories sollen schätzbar sein.
S	Small	User Stories sollen klein sein.
Т	Testable	User Stories sollen testbar sein.

Quelle: Cohn, Mike: User Stories Applied, Addison-Wesley, 2004



# Scrum > Agiles Schätzen

Agilität

Prozess

Prinzipien

Rollen

**User Stories** 

Agiles Schätzen



## Scrum > Agiles Schätzen > Relative Größe statt Dauer

- Tatsächlicher Aufwand liegt über oder unter der geschätzten Dauer, aber nie bei 100% Genauigkeit
  - Möglichst nahe an die 100% herankommen
  - Die Genauigkeit der Schätzung nimmt in der Regel nur marginal mit dem investierten Aufwand zu
- Story A dauert 2 Tage, Story B dauert 3 Tage
  - Die Dauer wird wahrscheinlich nicht eintreffen
  - Dass Story B länger dauert als Story A wird sehr wahrscheinlich
- Größenklassen statt Dauer schätzen
  - "S", "M" und "L"
  - Mittelgroße Story bildet eine Referenzstory
  - Alle weiteren Stories werden nur noch in die passende Größenklasse einsortiert



### Scrum > Agiles Schätzen > Story Points

- User Stories werden in Story Points geschätzt
  - 4 Punkte Story doppelt so groß wie eine 2 Punkte Story
  - Nur relative Dauer, keine konkrete Entwicklungsdauer
- Kriterien f
  ür die Gr
  öße
  - Komplexität
  - Risiken
  - Kenntnis der Basistechnologie
  - Externe Abhängigkeiten
  - Anzahl der Formulare, Anzahl zusätzlicher Datenbanktabellen
  - ...
- Schätzen ist wenig formal und setzt auf die Erfahrung der Entwickler
- Dauer wird aus Erfahrung abgeleitet (ständiges Evaluieren)



## Scrum > Agiles Schätzen > Größenordnungen

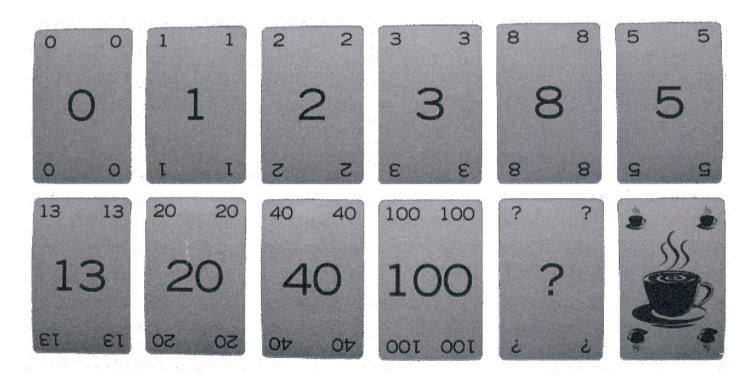
- Größenordnungen
  - Abstand Bäcker und Supermarkt zur Wohnung
    - Abstand Bäcker = 2x Supermarkt
  - Abstand Stadtteile
    - Karlruhe Weststadt zu Durchlach = 2x zur Oststadt
  - Zwischen unterschiedlichen Größenordnungen???
    - Abstand Bäcker und Durlach zur Wohnung? 58 oder 59x? → nicht wirklich sinnvoll
- Innerhalb gleichen Größenordnungen bleiben



#### Scrum > Agiles Schätzen > Punktesequenz

- Nicht linear, sondern exponentiell
  - Fibonacci-ähnliche Folge hat sich bewährt:
    - 1, 2, 3, 5, 8, 13, 20, 40, 100
    - Schätzung wird unsicherer, je größer und ungenauer eine Story definiert ist
    - 20:21 → 5%
    - 1:2 → 100%
    - Ab 20 Punkte gilt die Story als Epic → muss zerlegt werden in einzelne User Stories
- Bewährte Regel:
  - User Stories kurz vor der Übernahme in das Sprint Backlog → untere Skala 1,
     2, 3, 5, 8, 13
  - Epics auf der Skala 20, 40, 100





Quelle: Wirdemann, Ralf: Scrum mit User Stories, Hanser, 2009



- Scrum Team (Scrum Master, Product Owner, Team)
- Ablauf
  - Jedes Team Mitglied bekommt einen Satz Punktekarten
  - Scrum Master moderiert
    - Achtet auf die Einhaltung der Regeln
    - Timeboxing (Bsp. Sanduhr)
  - Scrum Owner stellt die User Story vor und erläutert sie kurz
  - Team stellt Fragen und diskutiert die Story
  - Jedes Team Mitglied entscheidet sich für eine Karte und legt sie verdeckt auf den Tisch
  - Hat jedes Team Mitglied seine Karte auf den Tisch gelegt, werden die Karten gleichzeitig aufgedeckt



- Ziel: das Team einigt sich auf einen gemeinsamen Wert für die Story
- Min und Max liegen oft weit auseinander
  - Das ist normal, denn es besteht unterschiedliches (Experten-)Wissen
  - Die Mitglieder mit niedrigster und höchster Schätzung begründen diese
  - Danach wird die Story ein zweites Mal geschätzt
  - Bis sich ein gemeinsamer Wert deutlich abzeichnet



- Mit geringem Aufwand eine möglichst realistische Schätzung
  - Es wird akzeptiert dass die Schätzung eine Schätzung ist und nicht perfekt sein muss
  - Es wird lediglich die Größe und nicht die Dauer geschätzt
  - Die Dauer leitet sich aus Erfahrung/Evaluierung aus den Story Points ab (Story Points pro Sprint)
  - Wissensverteilung
    - Alle Team Mitglieder von Anfang an dabei
    - Jeder kann an der Story arbeiten
  - Expertenwissen nutzen
    - Unterschiedliche Aspekte, Seiteneffekte und technische Implikationen werde frühst möglich aufgedeckt
  - Planungspoker erzwingt einen Dialog
  - Planungspoker macht Spaß → Als Faktor für das Erzeugen guter Arbeitsergebnisse