

# Scrum Patterns

von Uta Kapp, Februar 2010

Gute Software schnell und kostengünstig entwickeln, das scheint nur eine technische Aufgabe zu sein. Tatsächlich ist es eine Herausforderung, die sehr viel Organisations- und Sozialkompetenz erfordert. Dies wird von Scrum, einem kleinen, übersichtlichen Projektmanagement-Framework, das sich auf wenigen Seiten beschreiben lässt, berücksichtigt. Seine Größe erhält es durch die Menschen die es mit Leben erfüllen. Damit funktioniert Scrum für jedes Projekt und in jedem Team anders und doch gleich.

Scrum Patterns dienen dazu die Erkenntnisse aus Erfahrungen vieler Scrum Teams als wiederholbare Muster zu definieren, um sie so weiter zu geben. Diese Patterns sind für die praktische Hilfe im Scrum Alltag und haben keinen wissenschaftlichen Anspruch.

## Was ist Scrum?

Scrum ist ein einfaches Projektmanagement-Framework, in das Entwicklungsteams ihre eigenen erprobten Praktiken einbetten.

Der Rahmen sieht einen empirisch, iterativen Prozess vor, bei dem in Zeitabschnitten von je 30 Tagen ein nächster vollständig fertiggestellter Produktstand dem Kunden zur Evaluation vorgeführt wird. Termine sind nicht verschiebbar, sondern die Menge an Features, die in einer Zeitbox vollständig als „fertig“ vermerkt werden können, variiert. Da Zeitabschnitte, sogenannte Sprints, immer gleich lang sind, entwickelt das Team über die Zeit ein Gefühl und Statistiken darüber, wie viele Arbeitseinheiten es in einem Sprint erledigen kann. Damit wird die Planung in einem Team immer verlässlicher.

Scrum verfolgt einen evolutionären Ansatz. Vor jedem Sprint werden neue Prioritäten gesetzt und das Gelernte miteinbezogen. Das schafft Flexibilität in einer sich schnell wandelnden globalen Welt. Es ermöglicht dem Kunden den größten möglichen Wert, für seine Investitionen, zu erhalten.

Sich selbst-organisierende Teams entwickeln die höchste Produktivität, in der die Summe aller Teile mehr ergibt, als jeder Einzelne. Scrum stellt Prozesse und Methoden zur Verfügung, um dies zu ermöglichen.

## Werte und Prinzipien der agilen Softwareentwicklung

Scrum setzt das Konzept der agilen Softwareentwicklung in einen konkreten Vorschlag für das Projektmanagement um. Das menschliche Potenzial jedes Einzelnen wird mit Selbst-Organisation, unbürokratischem Vorgehen, Disziplin und Wertschätzung am intensivsten genutzt. Transparenz und Flexibilität zeichnen den Prozess aus.

Der Wertzuwachs, des Produktes für den Kunden, steht im Mittelpunkt. Dies erfordert einen Lernprozess während der Entwicklung. Neue Erkenntnisse fließen jederzeit ein. Es steht nur fest, was produziert wird und nicht wie. Dies ist ein radikal anderer Ansatz, als wenn das Endprodukt zu Beginn des Projektes vollständig spezifiziert ist. Um das zu erreichen, arbeitet Scrum mit festen Zeiteinheiten von 30 Tagen. Damit muss die Menge an Features pro Zeiteinheit variabel sein. Prioritäten werden klar gesetzt und Termine immer eingehalten. Dieses Vorgehen hat, bei genauer Betrachtung, den Vorteil, dass Termintreue der Entwickler möglich und klare Prioritäten des Kunden gefordert sind.

Da vor jedem Sprint neu begutachtet und geplant wird, fließen neueste Erkenntnisse in den Prozess ein. Es ist also ein empirisches Vorgehen, das der Evolution gleicht. Damit bekommt der Kunde den größten Wert für sein Geld.

## Was sind Scrum Patterns?

Erfahrungen mit Scrum zeigen, dass es viele soziale und organisatorische Muster gibt, die wiederholbar sind. Diese Patterns ermöglichen Scrum Teams die Erfahrungen von Anderen zu nutzen. Das Scrum Framework lässt Raum für jedes Team, es selbst mit Leben zu füllen. Die Scrum-Patterns helfen dabei.

Die Basispatterns werden hier in 4 Kategorien aufgeteilt: Meta-Patterns, Scrum-Framework-Patterns, Soziale-Patterns und Tool-Patterns.

### 1. Meta-Patterns

Bei der agilen Softwareentwicklung geht es vor allem um Menschen, deren Potenziale und Beziehungen. Tools und Prozesse stehen im Dienst der Menschen und damit an zweiter Stelle. Diese Werte wurden in dem Manifest für agile Software Entwicklung niedergelegt. (siehe <http://agilemanifesto.org/>) Scrum trägt als Framework für die agile Softwareentwicklung diesen Werten Rechnung.

Bei Scrum steht die Selbst-Organisation im Vordergrund. Dieses Meta-Pattern ist Grundlage für alle anderen Patterns. Damit Selbst-Organisation organisiert werden kann, gilt es das Eisberg-Pattern zu verstehen. Dieses ist die Grundlage aller sozialen Dynamiken.

### 2. Scrum-Framework-Patterns

Diese Patterns definieren das Scrum Framework als Set von Patterns. Diese Muster können auch als Organisations-Patterns bezeichnet werden (siehe Cop05]). Da die Muster in diesem Buch dem Gebrauch im Alltag eines Scrum Projektes dienen sollen, und keinen generischen Anspruch haben, verwenden wir hier gezielt die Terminologie die bei Scrum benutzt wird. Jedes Pattern verweist auf Soziale-Patterns und Tool-Patterns, die es ergänzen.

### 3. Soziale-Patterns

Als Ergänzung der Prozesse in Scrum dienen die sozialen Patterns dazu, Schablonen für den Umgang miteinander bereit zu stellen.

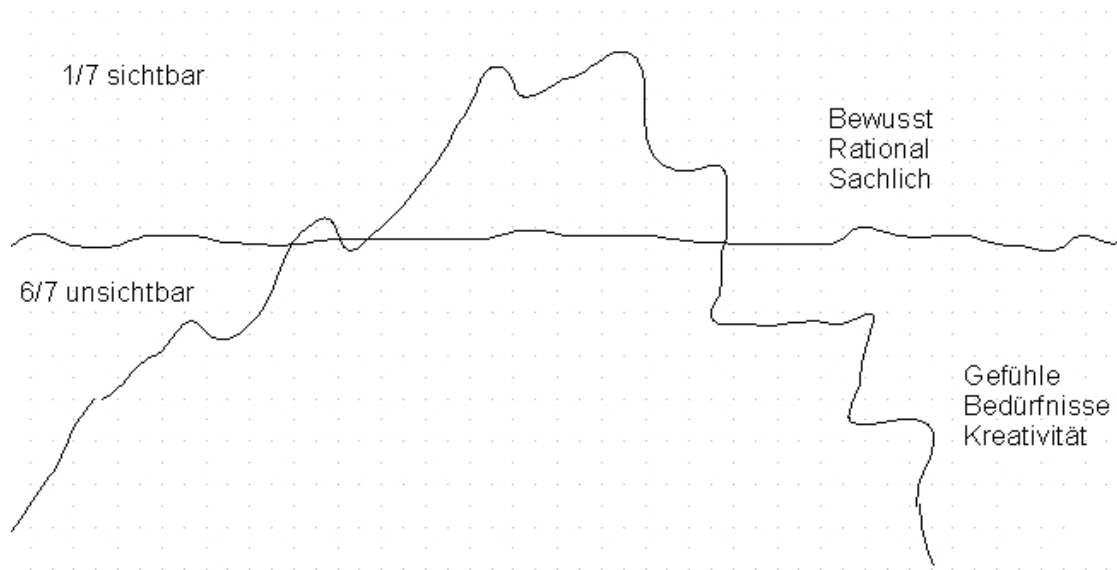
### 4. Tool-Patterns

Wenn das Scrum Team mehrere Sprints erfolgreich durchgeführt hat, oder über mehrere Standorte verteilt arbeitet, kommt der Wunsch nach elektronischer Unterstützung für das Führen des Product Backlog und Sprint Backlog, sowie das automatische generieren von Burndown Charts, auf.

Tool-Patterns stellen nützliche Schablonen bereit, die bei der Anwendung dieser Tools nützlich sind.

# Das Eisberg-Meta-Pattern

## Hintergrund



Dieses Meta-Pattern ist ein vereinfachtes Schema, das ursprünglich auf das Eisberg-Prinzip zurück geht, das von dem Psychoanalytiker Freud zuerst definiert wurde. Das Eisberg-Meta-Pattern verdeutlicht schematisch, dass es wie bei einem Eisberg zwei Ebenen der menschlichen Kommunikation und Interaktion gibt. Die rational, sachliche Ebene ist offensichtlich für alle. Die nicht rationale Ebene, die durch Gefühle, Wahrnehmungen und Bedürfnisse erlebt wird, macht den weit größeren Anteil der Kommunikation aus. Dieser Teil wird durch den schwimmenden Eisberg symbolisiert, der sich unter der Oberfläche befindet.

Auf das Eisberg-Meta-Pattern verweisen wir immer, wenn es um die beiden Ebenen menschlicher Beziehungen geht.

In Scrum wird ständig verhandelt. Es geht um Features, Zeiten, fertig oder nicht. Um erfolgreich zu verhandeln ist es notwendig die sachliche und die menschliche Seite getrennt voneinander zu behandeln.

## Pattern

Dieses Muster besagt, dass wir unsere Kreativität und unser menschliches Potential dann am besten nutzen, wenn wir die Beziehungsebene zulassen und sie uns bewusst sein lassen. Damit nutzen wir das Potenzial, dass aus der Verbundenheit miteinander entsteht.

In Scrum wird ständig verhandelt. Es geht um Features, Zeiten, fertig oder nicht. Um erfolgreich zu verhandeln ist es notwendig die sachliche und die menschliche Seite getrennt zu behandeln.

Auf das Eisberg-Meta-Pattern verweisen wir immer, wenn es um die beiden Ebenen menschlicher Beziehungen geht.

# Das Selbstorganisations-Meta-Pattern

## Hintergrund

Jedes Soziale System hat eine Ordnung. Die offizielle Ordnung in Form von dem Organigramm des Unternehmens sagt nur bedingt etwas über die reale Ordnung in einem System aus. Wenn man es schafft, sich diese Ordnung zunutze zu machen, dann erhält man als Ergebnis die größte mögliche Performanz des Systems, weil sehr wenig Zeit und Kraft für Machtspiele aufgebracht werden. Außerdem kann jeder sein volles Potential dem System zur Verfügung stellen.

## Meta-Pattern

Soziale Systeme können von außen organisiert und mit Kontrolle und Kommando geführt werden, oder von den an System teilnehmenden Personen selbst gestaltet werden. Ein selbstorganisierendes System ähnelt einem lebenden Organismus, wie zum Beispiel ein Riff.

Selbstorganisation wird hier beschrieben, weil es einer bewussten Entscheidung eines Teams bedarf, dieses als Meta-Pattern einzusetzen. Es ist ein wesentliches Pattern für Scrum.

## Anti-Pattern

- Kontrolle und Kommando: Hier gibt ein Projektleiter oder Vorgesetzter dem Team Anweisungen, wie Aufgaben erledigt werden sollen und bewertet das Ergebnis.

## **Scrum-But-Meta-Pattern**

### **Pattern**

Syntax eines Scrum-But (Scrum-But) (Begründung) (Umgehung)

Beispiel:

Wir nutzen Scrum, aber  
(Die täglichen Scrum Meetings sind zu viel Overhead)  
(weil die Teammitglieder keine Zeit haben)  
(also führen wir sie nur 1x pro Woche aus, außer wenn wir sie öfter brauchen)

### **Hintergrund**

Das Scrum-But dient dazu das Problem zu vertuschen, statt es aufzulösen.

# Scrum-Framework-Patterns

## Daily-Scrum-Pattern

Dieses Muster wird auch als Stand-Up Meeting Pattern [Cop05] bezeichnet.

### Pattern

Der „Daily Scrum“ ist ein Stand-Up Meeting von 15 Minuten, das täglich abgehalten wird.

Ablauf: Das Team trifft sich in einem abgetrennten, ruhigen Raum. Jeder beantwortet folgende Fragen:

- Was habe ich seit dem letzten Daily Scrum erreicht?
- Was plane ich bis zum nächsten Daily Scrum?
- Welche Hindernisse gibt es?

### Hintergrund

Wenn wir in einem Team zusammen arbeiten, dann sind wir mehr als eine Sammlung von Programmierern. Wir haben eine gemeinsame, geteilte Vision. Da können wir viel von Sportteams lernen. Ein gutes Beispiel ist hier der Fußball oder Rugby. In der Pause stimmt sich das Team neu aufeinander ab. Einerseits geht es darum, dass jedes Team-Mitglied die gleiche Information erhält. Andererseits geht es darum, sich neu aufeinander einzustimmen. Dadurch entfaltet ein Team sein gemeinsames Potenzial besser. Es ist mehr als nur ein logisch-rational erklärbarer Vorgang. Hier kommt das Eisberg Meta-Pattern ins Spiel. Ein Austausch von Angesicht zu Angesicht schließt die Kommunikation über Gefühle, Mimik und Gestik mit ein. Wir können es ein aufeinander einschwingen nennen. Hier geht es darum einander ganz wahrzunehmen. Dies schließt unsere subjektive Wahrnehmung und Intuition ein.

Unsere Gefühle können wir gezielt ins Spiel bringen, wenn wir das Feedback-Pattern nutzen.

### Anti-Patterns

- Die Besprechung dauert länger als 15 Minuten.
- In der Besprechung werden Lösungen für Probleme erarbeitet.
- Nicht-Team Mitglieder melden sich zu Wort.

## Sprint-Pattern

Dieses Muster wird auch als Programming Episodes Pattern [Cop05] bezeichnet.

### Pattern

Ein Sprint ist eine festgelegte Zeitbox in der programmiert wird. Es ist eine Iteration an dessen Ende ein lauffähiges Produkt steht. Die Zeitbox hat für jeden Sprint die gleiche Anzahl an Tagen. Der Ablauf eines Sprints ist als Rahmen festgelegt. Meistens sind dies 30 Tage. Dies ist der Kern von Scrum.

Am Anfang von jedem Sprint steht die Sprint Planung. (siehe Sprint-Planungs-Pattern) Am Ende des Sprint stehen das Review und die Retrospektive (siehe Review-Pattern und Retrospektive Pattern). Täglich findet ein Stand-U222p Meeting statt (siehe Daily Scrum Meeting).

### Anti Patterns

- neue Features während dem Sprint hinzufügen
- Zeitbox verändern
- Scrum But
- „Fertig“ missachten

### Hintergrund

Feste Zeiteinheiten haben den Vorteil, dass Terminverschiebungen nicht stattfinden können. Wenn in einem Sprint nicht alle gewünschten Features erledigt wurden, dann ist die Velocity des Teams langsamer als angenommen. Im nächsten Sprint kann das bei der Planung berücksichtigt werden. Dann werden weniger Features eingeplant.

Die Diskussion könnte also sein

- Wie können wir die Velocity erhöhen?
  - Motivations-Pattern
  - Konflikt-Pattern
  - Trainings-Pattern
- Wie können wir mehr Ressourcen einsetzen?
  - mehr Personal?
  - mehr Tools?
- Wie können wir die Qualifikation erhöhen?
  - Welche Ausbildung bringt Effizienz?
- Wie können wir Prioritäten strenger setzen?
  - Welche Teilfeatures sind wirklich wichtig?

## **Review-Pattern**

### **Pattern**

Am Ende von jedem Sprint findet ein Review Meeting statt. Ziel dieses Meetings ist es möglichst viel aus einer lauffähigen, vorführbaren Version zu lernen. Das Review Meeting solle eine feste Zeiteinheit von ca. 4 Stunden nicht überschreiten.

In diesem Meeting wird das lauffähige Produkt von Mitgliedern des Teams dem ProductOwner und den Kunden bzw. Benutzern demonstriert. Eine lauffähige Softwareversion besagt mehr als 1000 Worte. Die Erfahrung aus diesem Produkt fließt dann wieder in den nächsten Sprint ein.

### **Anti-Pattern**

- Zeitüberschreitung
- Konflikte (siehe Verhandlungs-Pattern)

### **Hintergrund**

Bei einem evolutionären, iterativen Ansatz für die Softwareentwicklung lernen wir aus jeder lauffähigen Version, also aus jeder Iteration. Zum Lernen gehört immer die Bereitschaft Emotionen, Intuition und Fehler zuzulassen (siehe Eisberg-Pattern).

Damit kommen Konflikte und Meinungsverschiedenheiten an die Oberfläche (siehe Verhandlungs-Pattern). Nur bei Diskussionen „mit heißem Herzen“ ist das Interesse wirklich groß genug, um ernsthaft zu lernen. Hier kann es leicht zu Auseinandersetzungen kommen. Hier ist es die Aufgabe des ScrumMasters zu schlichten und die menschliche Ebene von der Sachebene zu trennen.



## **Retrospektive-Pattern**

### **Pattern**

Am Ende von jedem Sprint, nach dem Review, findet ein Retrospektive-Meeting statt. Hier wird untersucht, was gut lief und was verbesserungsfähig ist. Nach dem Eisberg-Pattern ist es hier, um wirklich zu lernen, notwendig Gefühle, Emotionen und Intuition zuzulassen. Bewertungen und Bestrafungen sollten möglichst gering gehalten werden. Dies öffnet neue Räume für die Kreativität.  
Siehe auch Rückschau-Pattern

### **Anti-Pattern**

- Angst vor Fehlern
- Vertuschen: Wir haben uns alle Lieb!
- Probleme unter den Tisch kehren.

### **Hintergrund**

Wenn wir uns auf das Eisberg-Pattern und das Selbstorganisations-Pattern einlassen, dann sind wir bereit auch die Beziehungsebene einzubringen und Gefühl, Kreativität und Intuition als Qualitäten für das Team zu entdecken. Dies ermöglicht es auch den unbewussten Teil der Persönlichkeiten der Teammitglieder zu nutzen. Dies ist das größte Potenzial eines Teams, das lernen erst möglich macht. Hier wird Offenheit und Vertrauen miteinander offenbar.

## **Fertig-Pattern**

### **Pattern**

Jedes Team stellt für sein Projekt Kriterien auf, was „Fertig“ für ein Feature bedeutet.

### **Hintergrund**

User Stories aus dem Product Backlog gelten als „Fertig“ (Done), wenn die festgelegten Kriterien erfüllt sind. Wenn sie nicht erfüllt sind, dann kommt die User Story wieder in den Product Backlog und muss für den nächsten Sprint neu selektiert werden.

### **Anti-Pattern**

- Features die nicht „Fertig“ sind, gehen in ein Review.

# Soziale-Patterns

## Stimmungsbarometer-Pattern

Ziel: Stimmungslage im Team finden.

Frage: Wie fühlst du dich im Team?

*Vorbereitung:* Es werden Karten vorbereitet, auf denen verschiedene Gefühle stehen.

(Freude, Zufriedenheit, Enthusiasmus, Besorgnis, Wut, Enttäuschung, Verunsicherung)

*Aufgabe:* Jedes Teammitglied sucht sich die Karte aus, die seine augenblickliche Gefühlslage in Bezug auf die gemeinsame Arbeit im Team widerspiegelt. Dazu schaut er/sie jede Gefühlskarte an und prüft, ob diese Stimmungslage der Realität nahe kommt.

Die gezogene Karte wird verdeckt auf einen Stapel gelegt.

*Auswertung:* Der Moderator nimmt die Karten verdeckt auf und mischt sie, bevor er sie offen auf dem Tisch ausbreitet.

## **Feedback-Pattern**

### **Pattern**

Jeder stellt seine Gefühlslage in zwei bis drei Sätzen dar.

Frage: Wie fühlst du dich?

*Vorbereitung:* Es werden Karten vorbereitet, auf denen verschiedene Gefühle stehen.  
(Freude, Zufriedenheit, Enthusiasmus, Besorgnis, Wut, Enttäuschung, Verunsicherung)

*Aufgabe:* Jeder im Team zieht eine Karte, die seine Stimmungslage gut darstellt. Reih um sagt jeder ein paar Sätze zu dieser Stimmungslage und was die Ursache dafür ist.

### **Anti-Pattern**

- Bewertungen der Aussagen

### **Hintergrund**

Der emotionale Zustand einer Person ist ein guter Indikator für die Position einer Person im Team. Die Sachlage sollte getrennt behandelt werden. Dieses Pattern ist eine gute Möglichkeit sich das Eisberg-Meta-Pattern zunutze zu machen und die Beziehungsebenen in einem Team zu ergründen.

## **Elevator-Pitch-Pattern**

### **Pattern**

Jeder stellt sich, der Reihe nach, vor und beschreibt seine Position im Team.

### **Ant-Pattern**

- Bewertung
- Unterbrechung der Person

### **Hintergrund**

Wenn neue Teammitglieder ins Team dazu stoßen oder außen stehende am Team interessiert sind, dann kommt dieses Pattern zum Zug. Dies hilft vor allem bei der Teambildung, weil jeder seinen Platz im Team neu bestätigt. Das hilft dem Teammitglied das sich vorstellt, aber auch dem gesamten Team.

Damit immer nur eine Person das Wort hat, kann ein Symbol weitergereicht werden. Die Person, die das Symbol in der Hand hält, hat einzig das Wort.

## Planungspoker-Pattern

### Pattern

Damit die Schätzung eines jeden Teammitglieds das gleiche Gewicht erhält, wird ein Planungspoker durchgeführt. Jede Story wird hierzu einzeln betrachtet, eventuell diskutiert und jeder legt eine Karte, mit seiner persönlichen Größeneinschätzung, in Story Punkten, auf den Tisch. Schätzungs-Meetings finden regelmäßig statt und dauern ca. 90 Minuten.

*Vorbereitung:* Jeder schreibt ein Set Karten mit den Zahlen 1 2 3 5 8 13 21  
Eine mittelgroße Story als Referenz mit der Größe 5 wird ausgesucht.

### Anti-Pattern

- Stories sind zu groß zum schätzen

## **Committment-Pattern**

### **Pattern**

Feststellen, ob jeder hinter einer gemeinsam gefällten Entscheidung steht.

*Frage:* Welches Gefühl spiegelt das Bauchgefühl, dass du zu einer getroffenen Entscheidung hast, am besten wieder?

*Vorbereitung:* Es werden Karten vorbereitet, auf denen verschiedene Gefühle stehen.

(Freude, Zufriedenheit, Enthusiasmus, Besorgnis, Wut, Enttäuschung, Verunsicherung)

*Aufgabe:* Jedes Teammitglied sucht sich die Karte aus, die sein „Bauchgefühl“ in Bezug auf die getroffene Entscheidung wieder spiegelt.

Die gezogene Karte wird verdeckt auf einen Stapel gelegt.

*Auswertung:* Der Moderator nimmt die Karten verdeckt auf und mischt sie, bevor er sie offen auf dem Tisch ausbreitet.

## **Feedback-Pattern**

### **Pattern**

Frage: Wie lief die Zusammenarbeit im letzten Zeitabschnitt ?

Aufgabe: Jeder sucht ein positives und ein negatives Gefühl aus, dass im betrachteten Zeitraum auftrat und das Ereignis, das dieses Gefühl ausgelöst hat.

Auswertung: Jeder beschreibt den Anderen zuerst die positive Situation und dann die negative.

### **Anti-Pattern**

- Bewertung einer Antwort aus dem Team

### **Hintergrund**

Wenn alle Teammitglieder empathisch zuhören, dann können sie über die Art der Zusammenarbeit im Team reflektieren. Allein das Zuhören bewirkt bereits eine Bewusstseins-Veränderung im Team.



## **Tools-Patterns**

### **Product-Backlog-Tracker**

#### **Pattern**

In einer elektronischen Liste, die auch Tracker genannt wird, wird für jede User Story eines Scrum Projektes ein Eintrag (auch WorkItem) angelegt. Zusätzlich gibt es Felder für die Priorität, den Bearbeitungsstatus und die Größe der User Stories.

### **Task-Board**

#### **Pattern**

Auf einer Tafel werden, manuell, z.B. auf Kärtchen, alle Tasks geschrieben.

### **Release-Tracker**

#### **Pattern**

In dem Release Tracker werden alle Sprints registriert und zu welchem Release sie gehören. Hier wird auch das Sprintziel vermerkt und in welchem Zeitraum der Sprint läuft. Für jeden Sprint wird eine geschätzte Velocity angegeben, mit der ein Zeitaufwand pro Tasks errechnet wird.

### **Sprint-Backlog-Tracker**

#### **Pattern**

In einer elektronischen Liste, die auch Tracker genannt wird, wird für jede User Story des Product Backlogs ein oder mehrere Tasks als Eintrag (auch WorkItem) angelegt. Hier wird der Task beschrieben und in welchem Sprint er bearbeitet wird.. Zusätzlich gibt es Felder für die Priorität, den Bearbeitungsstatus und die Größe der User Stories. Idealerweise wird der Zeitaufwand

### **Technical-Backlog-Tracker**

#### **Pattern**

In einer elektronischen Liste, die auch Tracker genannt wird, wird für jede technische Anforderung eines Scrum Projektes ein Eintrag (auch WorkItem) angelegt. Zusätzlich gibt es Felder für die Priorität, den Bearbeitungsstatus und die Größe der User Stories.

# Impediments-Tracker

## Pattern

In einer elektronischen Liste, die auch Tracker genannt wird, wird für jeden bekannten Fehler ein Eintrag (Work Item) angelegt. Weitere Felder sind der Bearbeitungsstatus und der geschätzte Zeitaufwand in Storypunkten (siehe Planungspoker-Pattern). Impediments können hier nicht nur Bugs sein, sondern alle Hindernisse und Blockaden die das Projekt behindern.

In diesem Tracker wird vermerkt, wann ein Fehler behoben wurde, wann er getestet und in welcher Softwareversion er freigegeben wurde.

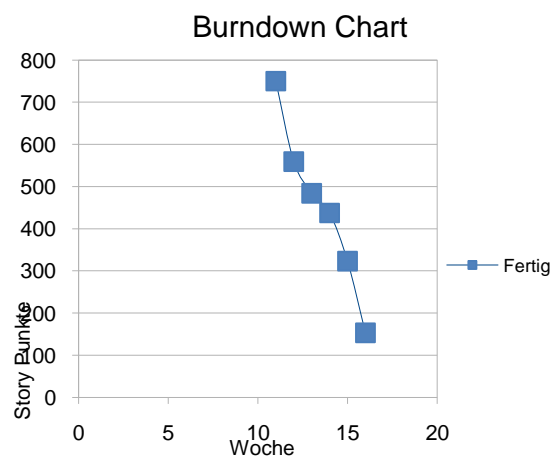
## Anti-Pattern

Neue Features werden als Impediment deklariert und erscheinen in dieser Liste.

# Sprint-Burndown-Chart-Pattern

## Pattern

In einer grafischen Darstellung wird der Verlauf eines Sprints dargestellt. Dieses kann automatisch aus einem Sprint-Tracker und Release-Tracker errechnet werden.



# Release-Burndown-Chart-Pattern

## Pattern

In einer grafischen Darstellung wird der Verlauf aller Sprints einer Release dargestellt. Dieses kann automatisch aus einem Sprint-Tracker und Release-Tracker errechnet werden. Diese Metrik ist wie in dem Burndown-Chart-Pattern, aber errechnet sich über alle Sprints eines Release.

## Literatur

[Cop05] James O. Coplien, Neil B. Harrison, Organizational Patterns of Agile Software Development, Pearson Prentice Hall 2005

[Schw04] Ken Schwaber, Agile Project Management with Scrum, Microsoft Press 2004