

## Chris Rupp

### Agile Priorisierungsformate

---

Häufig steht man vor der Herausforderung, dass es eine Vielzahl an Anforderungen (z.B. in Form von User-Stories) gibt und diese nicht alle oder alle auf einmal umgesetzt werden können. Folglich benötigt man eine sinnvolle Priorisierung dieser Anforderungen. Eine sinnvolle Priorisierung kann aber nur erfolgen, wenn alle Betroffenen mit einbezogen werden. Somit steht man vor der nächsten Herausforderung mit einer Gruppe von unterschiedlichen Menschen eine gemeinsame Priorität der Anforderungen zu erarbeiten. Damit dies gelingt wurden verschiedene Priorisierungstechniken entwickelt von denen wir Ihnen im Folgenden ein paar vorstellen möchten.

#### Weighted Shortest Job First (WSJF)

---

##### Beschreibung:

---

Methode zur Ermittlung der Anforderungen, die die schnellste Wertgenerierung in Aussicht stellen.

Dabei wird ein Wert ermittelt, um den erwarteten Wert der Anforderung (Cost of Delay – CoD) gegen den erwarteten Aufwand (Job Size) abzuwägen. Die Job Size könnte entweder den Story-Points oder einer anderen mit dem Team abgestimmten Einheit entsprechen. Im agilen Umfeld werden üblicherweise Story-Points für die Job Size verwendet.

Der CoD setzt sich zusammen aus

- dem erwarteten Geschäftswert (Business-Value),
- Zeitkritikalität (Time Criticality) und
- Risiken/Möglichkeiten (Risk Reduction – Opportunity Enablement Value).

Der erwartete Geschäftswert drückt aus, wie sehr der Kunde von der Anforderung profitiert.

Die Zeitkritikalität sagt aus, in welcher Dringlichkeit die Anforderung umgesetzt werden muss (z.B. wegen einer Gesetzesänderung). Risiken/Möglichkeiten beschreiben, ob die Implementierung der Anforderung Projektrisiken reduziert oder sich der Wert des Produkts steigert und ob sich neue Produktfunktionen aus der Implementierung der Anforderung ergeben.

Für die Bewertung der Kriterien stehen verschiedene Möglichkeiten zur Verfügung:

- Bewertung durch Voting Verfahren (z.B. Planning Poker)
- Bewertung durch Metriken (Formeln)

Aus der Division von erwartetem Wert und erwartetem Aufwand der Anforderung ergibt den WSJF-Index. Zu beachten ist, dass die Berechnung mit den gleichen Einheiten stattfinden muss. Wurde der erwartete Aufwand in Story-Points geschätzt, so muss auch der erwartete Wert in Story-Points geschätzt werden, um eine Vergleichbarkeit zu garantieren.

## Beispiel

---

In diesem Beispiel wird festgelegt, dass die Bewertung mit Planning Poker stattfindet. Dabei werden die Story-Points mit der Fibonacci Zahlenfolge bemessen.

User Story:

*Als Kunde möchte ich mit SHS das Hoch- und Herunterfahren meiner Rolläden steuern, damit ich diese nicht mehr manuell betätigen muss.*

In diesem Beispiel wird festgelegt, dass die Bewertung mit Planning Poker stattfindet.

Als erstes wird nun die Job Size geschätzt. Das Team hat die Größe der Story mittels Planning Poker auf beispielsweise 13 Story-Points im Verhältnis zu einer Referenzstory geschätzt.

Danach folgt die Bewertung des CoD. Dazu setzt der Product Owner einen Business Value von 8 Story-Points an weil User-Story für die BenutzerInnen von Wert ist, der Product Owner andere User-Storys profitabler einschätzt. Die Time Criticality schätzt er auf 5 Story-Points, weil Rolläden manuell bedienbar sind, manche Kunden sich diese Funktion jedoch zum Sommer- oder Winteranfang wünschen. Den Risk Reduction – Opportunity Enablement Value wurde auf 2 Story-Points geschätzt, weil er der Meinung ist, dass diese User-Story Projektrisikien nur in kleinem Maße reduzieren oder neue Projektmöglichkeiten schafft. Im nächsten Schritt kann der WSJF-Index berechnet werden. Dieser wird ermittelt, indem der CoD durch die Job Size dividiert wird. Der CoD beträgt  $8+5+2=15$ . Die Job Size beträgt 13. Der WSJF-Index beträgt  $15:13 = 1,15$ .

Zur Priorisierung wird der WSJF-Index miteinander verglichen. Ist der WSJF-Index einer anderen User-Story höher, gilt diese als höher priorisiert.

## Nutzen

---

Diese Methode bietet ein strukturiertes Vorgehen zur Priorisierung von Anforderungen in Bezug auf das Nutzen-Aufwand-Verhältnis. Anforderungen können mithilfe dieses Formats schnell und mit geringem Aufwand priorisiert und eingeordnet werden.

## Erfinder

---

Donald G. Reinertsen hat sich innerhalb seiner Forschung zur schlanken Produktentwicklung (Lean Product Development) u.a. damit befasst, wie man Produktfunktionalitäten priorisiert. Dabei hat er WSJF entwickelt.

## Quellen

---

- <https://www.projektmagazin.de/methoden/wsjf-weighted-shortest-job-first>
- <https://digitaleneuordnung.de/blog/wsjf-methode/>
- <https://www.scaledagileframework.com/wsjf/>
- <https://www.productplan.com/glossary/weighted-shortest-job-first/>

## MoSCoW

---

### Beschreibung

---

In dieser Methode werden alle User-Stories als wichtig angesehen, aber entsprechend ihrem Geschäftswert in vier Kategorien eingeteilt:

**M – Must have:**

Essentielle Anforderungen, ohne die das Produkt nicht funktionieren würde.

**S – Should have:**

Wichtige Anforderungen, die umgesetzt werden sollten, sofern sie die „Muss“-Anforderungen nicht gefährden.

**Co – Could have:**

Wünschenswerte Anforderungen, die erfüllt werden, falls nach Erfüllung der oberen Kategorien noch Ressourcen zur Verfügung stehen. Sie sind „nice-to-have“.

**W – Won't have:**

Anforderungen, die wenige Chancen auf eine Umsetzung haben. Sie werden evtl. in späteren Releases berücksichtigt.

Gewöhnlich wird die MoSCoW Methode in Meetings, mit einer Gruppe von Stakeholdern, angewendet. Dort bietet es sich an ein Zeitlimit zu setzen. Ein Grund dafür ist, dass der Fokus auf Ergebnisse geschärft wird. In dem Artikel *Time boxing planning: Buffered Moscow rules* von Eduarda Miranda, der an dem Institute of Software Research der Carnegie Mellon University angefertigt wurde, wird ausführlich erläutert, weshalb sich Zeitlimits positiv auf die MoSCoW Methode auswirken und wie man Zeitlimits bei der Ausführung der MoSCoW Methode setzt. Wichtig ist es aus der Priorisierung keine Wissenschaft zu machen, sondern möglichst zügig vorzugehen.

### Beispiel

---

Bei diesem Beispiel werden 3 User-Stories mit einer Timebox von 1,5 Minuten priorisiert.

User-Story 1:

*Als BenutzerIn möchte ich mit dem SHS das Hoch- und Herunterfahren meiner Rolläden steuern, damit ich diese nicht mehr manuell betätigen muss.*

User-Story 2:

*Als BenutzerIn möchte ich, dass das SHS Annäherungen von Personen an meine Haustür erkennt, damit ich mich darauf einstellen kann, wer vor der Haustür steht.*

User-Story 3:

*Als BenutzerIn möchte ich, dass das SHS während der Entriegelung meinen aktuellen Nummer-eins-Hit spielt, damit ich mich noch mehr darüber freue, nach Hause zu kommen.*

Die Stakeholder haben sich für folgende Priorisierung entschieden:

**M – Must have**

User-Story 2

**S – Should have**

User-Story 1

**C – Could have**

User-Story 3

In diesem Beispiel wurde der Kategorie W - Won't have keine User-Story zugeordnet.

### Nutzen:

---

MoSCoW bietet eine einfache, klare und leicht verständliche Kategorisierung der User-Stories, die den geschäftlichen Wert der User-Stories mit einbezieht.

### Erfinder:

---

Softwareexperte Dai Clegg entwickelte diese Methode als er bei Oracle arbeitete. Sie wurde erstmals hauptsächlich bei der dynamischen Systementwicklungsmethode (Dynamic Systems Development Method – DSDM) benutzt. Heute wird in Zusammenhang mit Timeboxing in der agilen Softwareentwicklung gebraucht.

### Quellen:

---

- <https://www.productplan.com/glossary/moscow-prioritization/>
- [https://www.researchgate.net/publication/220630837\\_Time\\_boxing\\_planning\\_buffered\\_moscow\\_rules](https://www.researchgate.net/publication/220630837_Time_boxing_planning_buffered_moscow_rules)
- <https://www.hotpmo.com/management-models/moscow-kano-prioritize>
- <https://t2informatik.de/wissen-kompakt/moscow-methode/>

Copyright © 2019 by SOPHIST GmbH

Publikation urheberrechtlich geschützt. Alle Rechte, auch die der Übersetzung, des Nachdruckens und der Vervielfältigung oder Teilen daraus, vorbehalten. Kein Teil der Publikation darf in irgendeiner Form, egal welches Verfahren, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet werden, vervielfältigt oder verbreitet werden.

Dies gilt auch für Zwecke der Unterrichtsgestaltung. Eine schriftliche Genehmigung ist einzuholen. Die Rechte Dritter bleiben unberührt.