

MoSCoW-Methode

Inhaltsverzeichnis: Definition – Kategorien – Vorteile und Nachteile – Download – Hinweise

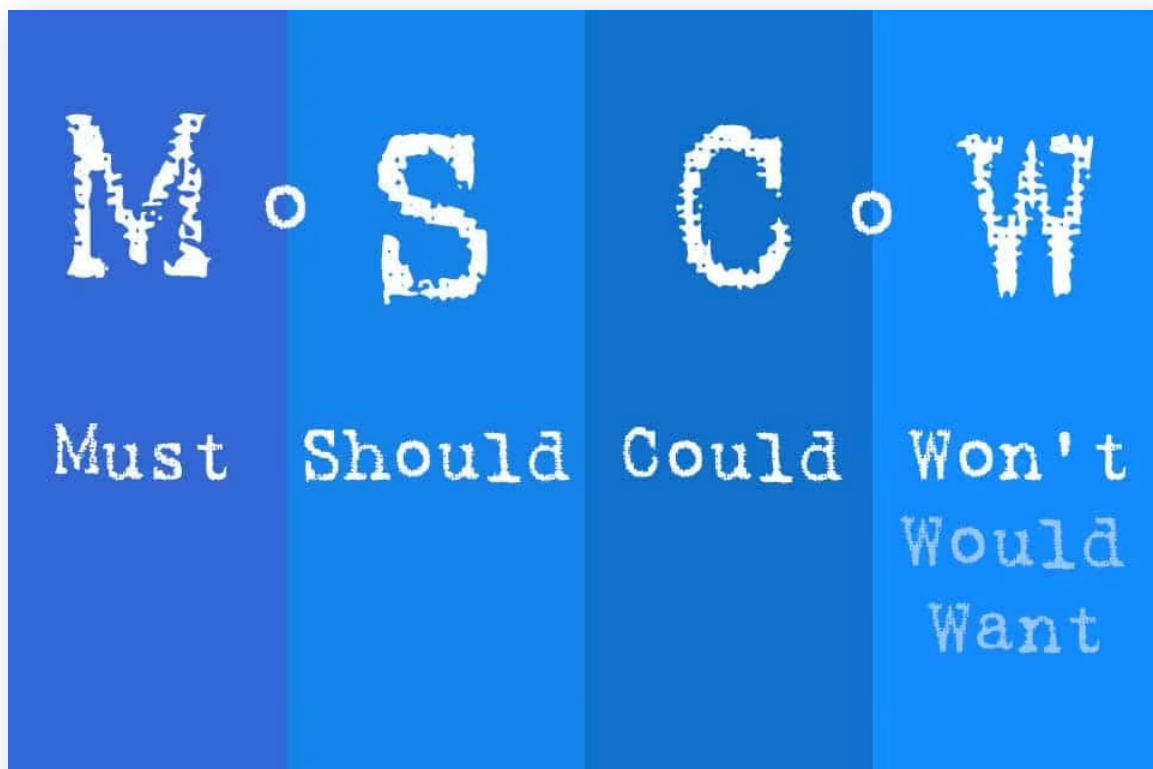
Wissen kompakt: Die MoSCoW-Methode ist ein Verfahren zur Priorisierung von Anforderungen, Zielen oder Aktivitäten, das vier Kategorien (Must, Should, Could und Won't) definiert.

MoSCoW-Methode Definition

Die MoSCoW-Methode ist ein vierstufiges Verfahren der Priorisierung. Meist wird sie zur Kategorisierung von Anforderungen genutzt, grundsätzlich eignet sie sich bspw. auch zur Priorisierung von Zielen, Aktivitäten oder Änderungsanträgen. Als Erfinder der MoSCoW-Methode gilt Dai Clegg, der die Methode 1994 bei Oracle erstmals im Rahmen der sogenannten Dynamic Systems Development Method (DSDM) nutzte. Heute wird MoSCoW – auch als MoSCoW-Prinzip, -Analyse oder -Priorisierung bezeichnet – u.a. in der Business Analyse, im Requirements Engineering, im Projektmanagement und in der Softwareentwicklung verwendet.

Die einzelnen, großgeschriebenen Buchstaben stehen für vier Kategorien:

- **M** = **M**ust (have)
- **S** = **S**hould (have)
- **C** = **C**ould (have)
- **W** = **W**on't (have)



Oftmals wird im Zuge der MoSCoW-Methode von einem Akronym gesprochen; da die kleingeschriebenen Buchstaben lediglich der Lesbarkeit dienen, ist dies nicht ganz korrekt, doch durch die Verwendung der o's lässt sich das Prinzip einfach wie die russische Hauptstadt aussprechen.

Die vier Kategorien der MoSCoW-Methode

Was bedeuten die vier Kategorien genau?

Must (have) – die Kategorie für “Muss”-Anforderungen. Sie gelten als nicht verhandelbar und sollten nicht im Widerspruch zu anderen Muss-Anforderungen stehen. Wendet man den Business Analysis Body of Knowledge – kurz [BABOK](#) – an, sollten sie darüber hinaus auch vollständig, einheitlich, richtig, umsetzbar, änderbar, eindeutig und testbar sein.

Should (have) – die Kategorie für “Soll”-Anforderungen. Idealerweise sollten Anforderungen, die in dieser Kategorie landen, ebenfalls umgesetzt werden. Da die Priorität aber niedriger als bei “Muss”-Anforderungen ist, die zudem immer vorrangig umzusetzen sind, kann es zu einer Implementierung in nachfolgenden [Releases](#) oder Projekten kommen.

Could (have) – die Kategorie für “Kann”-Anforderungen. Sie können umgesetzt werden, nachdem die “Muss”- und “Soll”-Anforderungen umgesetzt wurden. Sie werden daher auch als “nice to have” bezeichnet. Bei zeitlichen Engpässen oder Ressourcenkonflikten werden diese Anforderungen als erstes verschoben bzw. nicht realisiert.

Won't (have) – die Kategorie steht für Anforderungen, die nicht umgesetzt werden. Alternativ wird das **W** auch als **Would** – wäre schön, wenn es demnächst umgesetzt werden könnte – oder **Want** – wird gewünscht, aber in einem anderen Vorhaben, Projekt oder Release – interpretiert. Grundsätzlich wird empfohlen, Anforderungen auch innerhalb dieser Kategorie zu dokumentieren, denn so lässt sich nachvollziehen, ob eine scheinbar neue Anforderung evtl. bereits erfasst wurde. Zusätzlich können dokumentierte Anforderungen in zukünftigen Projekten als Quelle dienen.

In der Praxis landen meist sowohl die “Muss”- als auch die “Soll”-Anforderungen in weiterführenden Dokumenten wie Lastenhefte oder Business Cases. Kann-Anforderungen und nicht umzusetzende Anforderungen verbleiben hingegen meist in internen Backlogs, Listen oder Tools.

Vorteile und Nachteile der MoSCoW-Methode

Die MoSCoW-Methode bietet einige klare Vorteile:

- Es findet eine einfache Kategorisierung von [Anforderungen](#) – idealerweise in Abstimmung mit den Stakeholdern – statt. Die Verwendung der Worte Must, Should, Could und Won't vermittelt jederzeit Klarheit über die Bedeutung der Kategorien.
- Aus der Einordnung der Anforderungen (oder Zielen, Aufgaben, [Änderungsanträgen](#) etc.) ergibt sich eine natürliche Reihenfolge der Umsetzung.
- Das Prinzip hinter der Methode ist sehr leicht verständlich, so dass Stakeholder und Entwickler einfach einen gemeinsamen Nenner finden.

Andererseits hat die MoSCoW-Methode auch einige Nachteile:

- Die Einteilung als solche ist relativ grob und liefert innerhalb der einzelnen Kategorien keinerlei Indikatoren, in welcher Reihenfolge Anforderungen umzusetzen sind. Daher sollten zumindest die "Muss-" und die "Soll"-Anforderungen detaillierter betrachtet werden. Technische Abhängigkeiten, Aufwände, Ressourcen – es gibt einige Aspekte, die bei der weiteren Sortierung innerhalb der Kategorien nützlich sind.
- In der Praxis passiert es immer wieder, dass ein Großteil der Anforderungen als "Muss"-Anforderungen deklariert wird. Die Einordnung von Anforderungen in andere Kategorien führt bestenfalls zu einer späteren Implementierung, im ungünstigsten Fall werden sie gar nicht realisiert. Manche Ratgeber empfehlen daher bspw. maximal 60% der Anforderungen als "Muss"-Anforderungen zu deklarieren; solche Vorschläge greifen allerdings sehr kurz, denn sie ändern nichts an den Sorgen der Anforderungssteller, sie ignorieren jegliche Entwicklungskapazitäten und beantworten natürlich auch nicht die Frage "welche Anforderungen gehören zu den ominösen 60%?".
- Wie bei vielen Methoden handelt es sich um eine Stichtagsbetrachtung. Stakeholder können ihre Meinungen ändern, technische Herausforderungen können auftreten, und neue Anforderungen oder Änderungen hinzukommen – es ist also nicht unüblich, dass Anforderungen innerhalb der Kategorien wandern.

- Gerne wird die Einteilung anhand der Wünsche und Vorstellungen von Stakeholdern empfohlen. Hier gibt es jedoch andere Konzepte wie bspw. das [Kano-Modell](#) der Kundenzufriedenheit, die Anforderungen differenzierter betrachten. Unternehmen sollten daher nachdenken, verschiedene Methoden und Prinzipien im Zuge einer Analyse und Priorisierung miteinander zu kombinieren.

Priorisierung Whitepaper

Alles Wichtige über

Priorisierung

auf einen Blick.



Wollen Sie das Priorisierung Whitepaper kostenlos
downloaden?

Alles Wichtige über Priorisierung auf einen Blick.

- Definition und Funktionsweise

- Methoden wie bspw. Eisenhower Matrix, Pugh Matrix oder MoSCow-Methode
- Vor- und Nachteile von Priorisierungen
- Herausforderungen und Tipps

Wissen auf 18 Seiten zum Mitnehmen.

[Zum Download des Whitepapers](#)

Hinweise:

Haben Sie Lust auf einen neuen [Lieblings-Newsletter](#)?

Die Inhalte auf dieser Seite dürfen Sie gerne teilen oder verlinken.

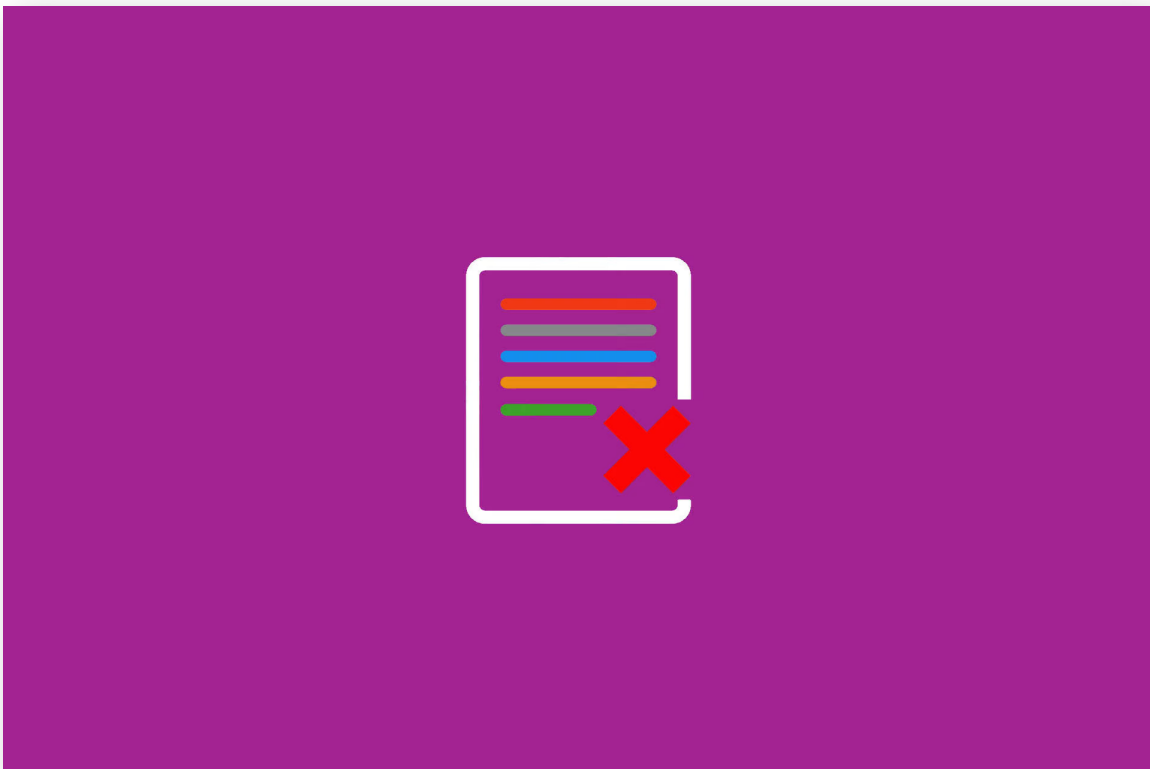
Es gibt eine alternative Interpretation der kleingeschriebenen o's: Werden sie als "oder" bzw. "or" verstanden, entstehen zwei sich gegenüberstehende Kategorien, also **Must or Should** sowie **Could or Would** bzw. **Won't**.

Hier finden Sie ein englischsprachiges Video zu [Agile in Practice: Prioritisation using MoSCoW](#).

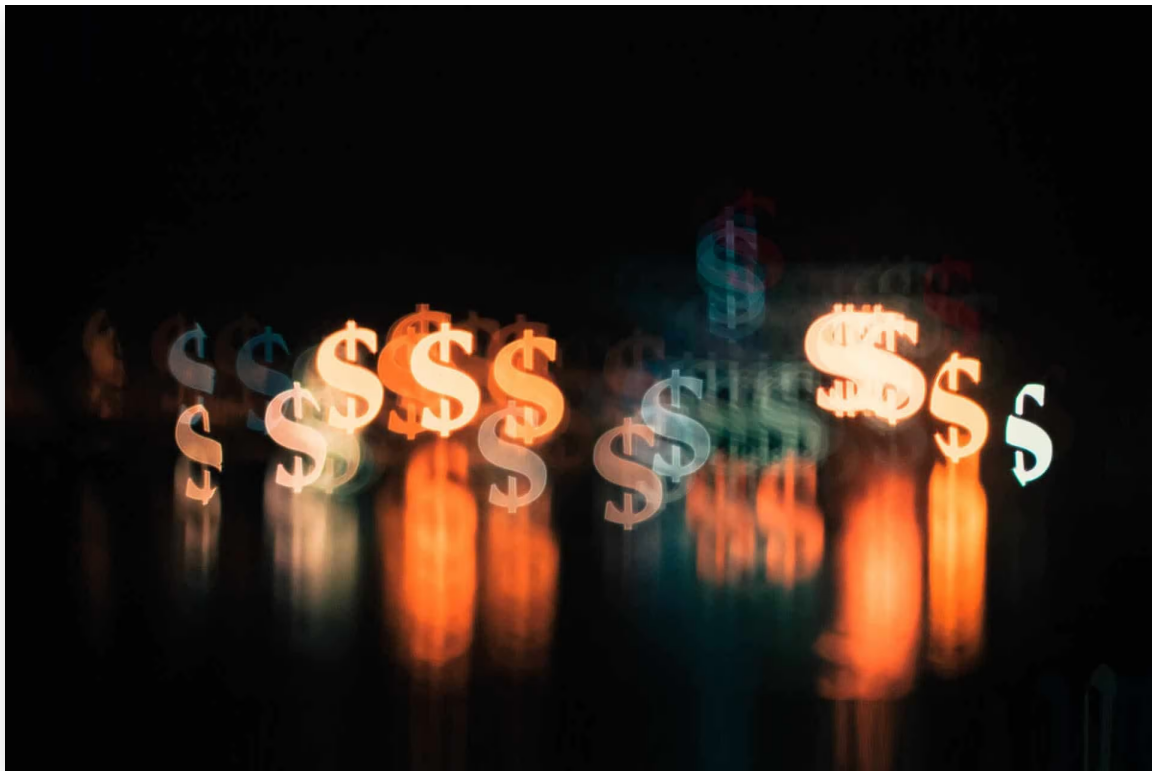
Und Hier finden Sie ergänzende Informationen aus unserem t2informatik Blog:



Best Practices bei Anforderungsworkshops



Anforderungen eliminieren



Auf der Jagd nach Kennzahlen

Was macht eigentlich t2informatik? [Ein Klick hier und Sie wissen es.](#)

Kompetenzen

Softwareentwicklung

Über uns

Fragen und Antworten

Kommunikation

Sprechen Sie mit uns

Feedback

Ihr Weg zu uns

Kooperation

Referenzen

Partnerschaften

Jobs

Inhalte

Blog

Wissen kompakt

Downloads

Newsletter

© t2informatik GmbH, Bülowstraße 66, Aufgang D3, 1. OG, 10783 Berlin, E-Mail:
internet@t2informatik.de

