# Analyse Kanban Comic

Diese Analyse beschäftigt sich mit einem Comic, der den Kanban-Prozess darstellt und übliche Situationen dieser Projektmanagement Methode beschreibt. <https://blog.crisp.se/2009/06/26/henrikkniberg/1246053060000>

Kanban kommt aus dem japanischen und bedeutet Signalkarte. Dabei wird der Fokus hauptsächlich auf einen gleichmäßigen Arbeitsfortschritt gelegt, bei dem der Fokus auf die Fertigstellung der Aufgaben gerichtet ist und der Arbeitsfluss priorisiert wird. Zusätzlich sollte Ballast minimiert werden, wobei Ballast in Kanban verschiedenste Formen annehmen kann. Charakteristisch gesehen hat jede Form von Ballast den gleichen Effekt: Es hindert, den Kunden schnell zufrieden zu stellen und erhöht den Wert des Produktes nicht.

Kanban besitzt vier Hauptmerkmale. Eines davon ist es, die Anzahl der Aufgaben pro Prozess zu limitieren, um so die Arbeitsdauer zu reduzieren und den Wert des Produktes zu erhöhen. Der Wert ist dabei ein abstraktes Ziel mit nicht verhandelbarem Nutzen. Diese Ziele lassen sich wiederum als Signalkarte darstellen. Alle anderen Merkmale werden im Laufe der Analyse aufgegriffen und erläutert.

Die erste Grafik visualisiert ein Kanban-Board, die alle wichtigen Phasen einer Wertschöpfungskette von einem Softwareentwicklungsprojekt abbildet. An erster Stelle steht der Backlog, der alle zu bearbeitende Aufgaben auflistet. Dieser wird von dem Service-Request Manager verwaltet. Als Nächstes wurde eine Phase aufgezeichnet mit dem Titel „Selected“, welche mit zwei Aufgaben limitiert wurde. Unmittelbar danach befindet sich die Development-Phase. Sie wurde in zwei Spalten unterteilt, darf maximal zwei Aufgaben beinhalten und wird von einem Entwicklungsteam aus vier Personen verwaltet. Am Schluss der Wertschöpfungskette befindet sich die Deployment-Phase, welche von einem Team aus zwei Personen bearbeitet wird. Diese heißen vermutlich Service-Delivery Manager, da sie die Bereitstellung von ausgewählten Elementen für Kunden übernehmen.

Nun startet der Prozess, der durch die nächsten 2 Grafiken illustriert wird: Der Service-Request Manager wählt dabei die zwei wichtigsten Aufgaben A und B aus dem Backlog Pool heraus und signalisiert, dass diese bereit für den Durchlauf sind, indem er sie auf die Phase Selected legt. Das Development-Team teilt sich hier auf und beide Aufgaben werden von jeweils zwei Personen übernommen. Wichtig dabei ist, dass das Team eigenständig organisiert und die Aufgaben von Selected „pullt“. Währenddessen kann der Service-Request Manager bereits zwei weitere Aufgaben C und D zu „Selected“ hinzufügen.

Hier ist bereits eine weitere Haupteigenschaft von Kanban aufgetreten: Kanban ist eine Pull-Methode. Das Team übernimmt neue Aufgaben, sobald sie bereit dafür sind. Es werden dadurch mehrere agile und schlanke Werte realisiert. Zum einen wird Ballast verringert: „Task-Switching“ beschreibt eine Abfallform, bei der sich Entwickler durch einen Wechsel der Aufgabenstellung neu auf ein Themengebiet eindenken muss. Diese Zeit ist signifikant hoch und wird durch das Auswählen einer einzigen Aufgabe verringert. Es verstärkt den Lernprozess und bekräftigt das Teambuilding, da jedes Teammitglied entscheidet, wann es bereit für die gegebene Aufgabe ist. Zusätzlich werden Entscheidungen so spät wie möglich getroffen, was dazu führt, dass mehr Informationen zu einer Aufgabe gesammelt werden können, um den Entscheidungsprozess vereinfachen.

In der nächsten Grafik wird illustriert, wie die erste Aufgabe A erledigt wird und vom Deployment/Service-Delivery Team übernommen wird. Dadurch öffnet sich ein Platz für eine weitere Aufgabe in der Development-Phase, die mit der Aufgabe C gefüllt wird. Nun kommt es in der anschließenden Grafik zum ersten Problem: Das Service-Delivery Team stößt auf eine Schwierigkeit in der Deployment-Phase und kann Aufgabe A nicht abschließen. Gleichzeitig wird jedoch Aufgabe B in der Development Phase ebenfalls fertig. Da die WIP-Limitierung maximal zwei Aufgaben pro Phase vorgibt, dürfen die Teammitglieder aus dem Development-Team keine weitere Aufgabe aufnehmen. Sie haben nun die Wahl, der anderen Hälfte zu helfen, oder den Service-Delivery Managern zu helfen. Währenddessen wird Aufgabe K nun zu „Selected“ hinzugefügt und der Service Request Manager stößt nun ebenfalls an das WIP Limit.

Wie bereits im zweiten Absatz erwähnt, besitzt Kanban eine Limitierung für die Aufgaben, die erfüllt werden dürfen. Dies hat Vorteile als auch Nachteile: Gewöhnlicherweise wird dadurch die Average Completion Rate der Aufgaben erhöht. Gleichzeitig wird der Lerneffekt erhöht, da das Team entlastet wird und Zeit für die Weiterbildung entsteht, Die Aufgaben bleiben so lange in den Positionen, wo sie waren, bis sie gepullt werden und die Little Law Formel wird umgesetzt. In Einzelfällen kann jedoch darauf verzichtet werden, um sich auf den Arbeitsfortschritt zu konzentrieren.

Im Comic entschließen sie sich den Service-Delivery Managern zu helfen. Dies führt zu zusätzlichem Waste wie Task Switching, Defects und Waiting. Es entpuppt sich schnell, wie in den anschließenden Grafiken illustriert wird, dass das Problem in der Aufgabe A von größerer Natur ist und es dadurch zu gröberen Verzögerungen in der gesamten Wertschöpfungskette kommt. Das restliche Development Team schließt die Aufgabe C ab und versichert dem Service Request Manager, dass Aufgabe K gestartet wird, sobald A fertiggestellt ist. Das gesamte Team stellt fest, dass es sich bei A um ein wiederkehrendes Problem handelt, was in einem Bericht erfasst wird, um zukünftige Probleme dieser Art zu vermeiden. Dies fällt unter den letzten 2 wichtigen Hauptmerkmalen von Kanban.

Die letzten zwei Hauptmerkmale beschäftigen sich damit, dass Informationen dargelegt werden müssen und der Ablauf kontinuierlich verbessert wird. Informationen beinhalten die verschiedenen Ablaufphasen der Wertschöpfungskette, Aufgaben in den Phase, die beteiligten Personen und die Limitierung. Der Ablauf wird kontinuierlich durch kleine Einwürfe, wie der erwähnte Testfall, verbessert. In der letzten Grafik wird illustriert, dass ein ähnliches Problem zwar vorherrscht, jedoch bereits kleine inkrementelle Schritte passierten, um den Ablauf zu verbessern. Es wurde das Limit für das Development erhöht und der Testcase für zukünftige Probleme in der Deployment-Phase eingebaut.