# Kanban

„Kan-ban“ ist eine Zusammensetzung der beiden japanischen Worte für Signal und Karte. Diese Signalkarten wurden bei Toyota im Zuge der Umstrukturierung ihres Produktionsprozesses in den 40er Jahren eingesetzt. Dort dienten sie zur Aufforderung an ein später im Produktionsverlauf arbeitendes Team, etwas zu produzieren. Ohne diese Aufforderung fertigte keine Station etwas an, so dass die Menge der überschüssigen Teile gering blieb. Solch ein System wird auch Pull-System genannt. Grund für diese Art von Produktion war der Wunsch nach einer kontinuierlichen Produktionsmenge bei Toyota. So wurde Kanban die Basis für den Prozess der kontinuierlichen Verbesserung, japanisch Kaizen, den Toyota anstrebte.

Im Bereich der Agilen Software-Entwicklung wurde das Pull-System kombiniert mit der Theory of Constraints und dem Lean Manufacturing und formte ein neues Verständnis des Begriffs Kanban. Die Theory of Constraints wurde maßgeblich von Eli Goldratt entwickelt und besagt grob, dass die langsamste Stelle, der sogenannte Engpass oder Bottleneck der Produktion, bestimmt, wie groß der Durchsatz des Systems ist. Deshalb muss man diese Engpässe erkennen und wenn möglich eventuell beheben. Dies geschieht iterativ, d.h. man erkennt erst einen Bottleneck, behebt diesen und eröffnet somit gleichzeitig den Weg für einen neuen an anderer Stelle. Während demnach die Theory of Constraints ihren Schwerpunkt auf Flaschenhälse legt, konzentriert sich Lean Manufacturing und das darauf aufbauende Lean Development auf die Verbesserung des Flusses oder Flows. Diese Eigenschaften versuchte David J. Anderson, der inoffizielle Begründer von Kanban in der Software-Entwicklung, zusammenzuführen. Im Folgenden wird hauptsächlich seine Vorstellung dieses agilen Software-Entwicklungsprozesses erläutert, die sich in seinem Buch „Successful Evolutionary Change for your Technology Business“ nachlesen lässt.

## Eigenschaften von Kanban

„Kanban [...] is used to refer to the methodology of evolutionary, incremental process improvement [...]and has continued to evolve in the wider Lean software development comunity in the years since.“ (e-book, Kapitel 1 Ende ?)

Diese Definition fasst bereits die wichtigsten Eigenschaften von Kanban in einem Satz zusammen. Im Gegensatz zu vielen anderen agilen Prozessen löst Kanban nicht die vorher existierende Methode in einem Schlag ab. Stattdessen soll der bereits existierende und den Mitarbeitern bekannte Prozess in kleinen Schritten verbessert werden. Die Idee dahinter ist, dass Menschen Gewohnheiten haben und mögen. Ihnen widerstrebt Veränderung. Deshalb ist es besser, diese Anpassung schrittweise durchzuführen, um auf so wenig Widerstand zu stoßen.

Abgesehen von der Idee der kontinuierlichen Verbesserung steht im Mittelpunkt von Kanban ein weiteres Konzept: der Flow oder Fluss. Hierbei ist gemeint, dass Tickets möglichst gleichmäßig durch das System wandern sollen, sie also so wenig wie möglich still stehen und warten müssen. Dies setzt voraus, dass Tasks sich jeweils in ihrer Größe nicht allzu sehr unterscheiden. Da dies nicht immer möglich ist, können Aufgaben Service-Level-Agreements und Item-Types zugeordnet werden. Diese können zur besseren Visualisierung mit verschiedenen Farben gekennzeichnet oder in Swim Lanes, abgegrenzten Zeilen am Board, zusammengefasst werden. So ist der Fluss auch bei verschieden großen Tasks möglich. Das Ziel von Kanban ist es nun, alle weiteren Behinderungen des Flows zu erforschen und möglichst zu beseitigen. Dafür werden die verschiedensten Metriken, Diagramme und Statistiken verwendet, so z.B. Burn-Charts, das Cumulative Flow Diagram oder der Durchsatz.

Insgesamt ist Kanban jedoch sehr frei und anpassbar in seinen Techniken und Prinzipien. Dies ist auch gar nicht anders möglich, da man auf dem vorhandenen Prozess aufbaut, und dieser von Team zu Team unterschiedlich ist. So gut wie alle im Folgenden vorgestellten Methoden haben sich zwar in vielen Teams bewährt, aber außer der Beschränkung des WIP gibt es keine Pflichtvorgaben. Sie sind nur Best Practices. Selbst das Kanban-Board ist nur ein Vorschlag zur Visualisierung.

Ebenso freiwillig sind Iterationen in Kanban; sie werden jedoch fast von allen Quellen trotzdem empfohlen. Allerdings ist es möglich, verschiedene Taktfrequenzen (englisch „cadences“) bei Planung, Release oder Priorisierung zu setzen. So kann ein Meeting zur Priorisierung jede Woche stattfinden, aber eine Lieferung nur aller vier Wochen.

Kanban selbst schreibt weiterhin auch keine Rollen vor, so dass diese aus dem vorher existierenden Prozess beibehalten werden können, und auch die Priorisierung des Backlogs ist freiwillig. Dies ist aus Kundensicht aber eher störend, da der Kunde selten seine Aufgaben nach dem First-In-First-Out-Prinzip abgeordnet haben will.

Auf Grund dieser doch sehr starken Modifizierbarkeit von Kanban, wird die Methode noch immer kontrovers diskutiert. Allerdings ist weniger umstritten, dass Teams, die diese Methode intensiv nutzen, messbar verbesserte Produktivität, Qualität, Kunden- und auch Mitarbeiterzufriedenheit sowie kürzere Lieferzeiten vorweisen können.

## Kanban einführen

Die bereits erwähnte kleinschrittige Einführung von Kanban unterliegt wie alles andere ebenso keiner festen Regel oder Reihenfolge. Trotzdem haben sich einige Schritte bewährt.

Der erste und einfachste Schritt ist die Visualisierung des Workflows. Dies wird für gewöhnlich mit Hilfe eines Boards realisiert wie es auch viele andere agile Softwareprozesse verwenden. Jeder Schritt im Workflow erhält mindestens eine Spalte am Board. Allerdings hat ein Kanban-Board auch einige Besonderheiten. Hierzu zählen die Begrenzung des Work-In-Progress aus Schritt 2 und die mögliche Anordnung der Aufgaben in Swim Lanes. Das Ziel dieser Visualisierung ist einerseits das Bewusstmachen und Verinnerlichen des Arbeitsprozesses für alle Mitarbeiter. Andererseits werden so bereits relativ zeitig die bereits angesprochenen Bottlenecks sichtbar. Jede Spalte innerhalb eines Kanban-Boards kann als Work-Queue verstanden werden. Das bedeutet, alle Tickets bewegen sich nach und nach durch die Spalten bis zum Ende des Boards. Wenn sich in einer Queue nun immer mehr Tickets ansammeln, ist die Station unmittelbar danach der Engpass, denn sie kann die Tickets nicht schnell genug abarbeiten und nachziehen.

Schritt 2 kann bereits schwieriger zu implementieren sein. Mit der Begrenzung des Work in Progress wird paralleles Arbeiten verringert. Am Board wird dies durch Begrenzungen der Spalten ausgedrückt. Jede Spalte erhält eine maximale Anzahl an Tickets. Diese Zahl steht für gewöhnlich über der Spalte am Board und darf nur in wenigen Ausnahmefällen überschritten werden. Damit findet weniger Multi-Tasking statt, was wiederum bedeutet, man beendet eine Aufgabe bevor man eine neue beginnt. Damit bewegen sich die Tickets automatisch schneller zum Ende des Boards statt in einigen Spalten lange zu verweilen, d.h. der Durchsatz wird verbessert. Daraus wiederum resultieren schnellere Releases an den Kunden, der eher Feedback zum gelieferten Produkt geben kann. Gleichzeitig ist ein Entwickler gezwungen bei Problemen nicht einfach eine andere Aufgabe zu suchen, sondern stattdessen sofort an der Lösung des Problems zu arbeiten, damit sie den Workflow nicht zu lange blockiert. Damit ist Kooperation und Hilfe untereinander gefragt.

Schritt 3 ist der Optimierungsschrit und betrifft die Kontrolle des Flusses, indem verschiedene Größen gemessen werden. Dazu zählen der Durchsatz, Warteschlangenlängen oder der Flow selbst. Ziel dabei ist es, die Planung zu erleichtern und sich immer mehr an eine möglichst korrekte Zielvorgabe für den Kunden anzunähern.

In Schritt 4 sollten die selbstaufgestellten Regeln des Teams schriftlich festgehalten werden. Dazu können Dinge wie eine Definition of Done gehören oder auch wie die Wahl des nächsten Tickets vonstatten gehen soll. Damit wird Unsicherheit über den Prozessablauf bei den Teammitgliedern vorgebeugt.

Schließlich sollen im letzten Schritt Modelle verwendet werden, „um Chancen für kollaborative Verbesserungen zu erkennen“ (Quelle Zitat: <http://de.wikipedia.org/wiki/Kanban_%28Softwareentwicklung%29>). Diese Modelle können aus den verschiedensten Bereichen übernommen werden, so z.B. aus der Theorie of Constraints, und sind ebenfalls wieder frei wählbar. Auch Eigenentwicklungen oder Modifikationen sid erlaubt.

Die eben vorgestellte Einführung wird oft auch kürzer in nur drei Schritten zusammengefasst (Abbildung 1): Visualisieren, Messen und Optimieren.

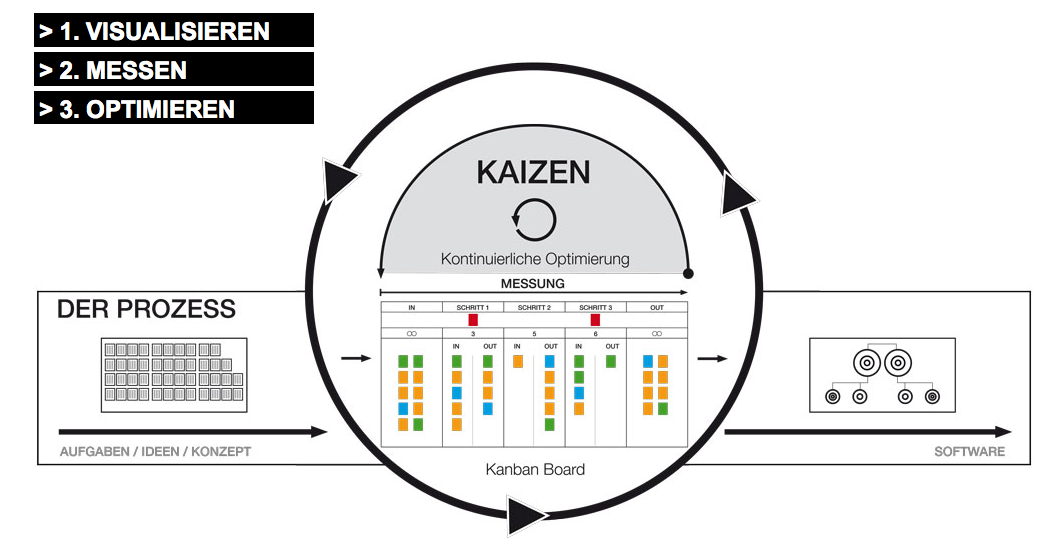


Abbildung 1: http://kanban-plakat.de/

## Kommunikation und Koordination

Je nach Umsetzung gibt es verschiedene Meetings, die den Arbeitsprozess mit Kanban vereinfachen. Dazu gehört zu allererst das Daily, ein tägliches Standup-Meeting, bei dem der Projektfortschritt am Board betrachtet wird. Dabei soll nach Anderson jedoch im Gegensatz zu anderen agilen Prozessen nicht erläutert werden, wer was tut und tun wird. Dies sollte aus dem Board schnell ersichtlich sein. Stattdessen wird lediglich überprüft, ob Tickets existieren, die nicht weiter kommen, oder ob Bottlenecks auffallen, die behoben werden müssten und können. Als mögliche Idee schlägt er vor, Tickets, die sich einen Tag nicht bewegt haben, zu markieren, beispielsweise mit einem Punkt für jeden Tag. Damit fallen sie im nächsten Daily sofort wieder auf.

Alle Themen, die für das normale Daily zu detailliert oder nicht ganz passend sind, können im After Meeting gleich im Anschluss besprochen werden. Hier finden sich eher kleine Gruppen von zwei bis drei Personen zusammen, die ein gemeinsames Problem lösen möchten oder andere Informationen austauschen, die nicht für alle am Daily Beteiligten wichtig sind.

Die Queue Replenishment Meetings dienen zur Priorisierung der Input-Queue, also der Spalte am Board, in der die abzuarbeitenden Tickets zu finden sind. Die Priorisierung wird vom Product Owner und eventuellen anderen Stakeholdern durchgeführt. Das Entwicklungsteam ist für gwöhnlich nicht involviert. Es ist möglich diese Meetings regelmäßig einzutakten oder eher spontan (on-demand) abzuhalten, wenn das Team die Priorisierung von allein zufriedenstellend erledigt. Die Dauer zwischen den regelmäßigen Meetings beeinflusst die benötigte Größe der Queue und sollte eher kurz sein, z.B. eine Woche.

Ähnliches gilt für Release Planning Meetings. Auch sie können regelmäßig stattfinden, was regelmäßige Releases zur Folge hat. Damit ist keine weitere Koordination der Teilnehmer notwendig. Werden sie unregelmäßig durchgeführt, muss der Kunde mit unregelmäßigen Releases einverstanden sein. Sie erfordern sehr viel mehr Planung, um allen, die möchten, die Teilnahme zu ermöglichen.

Während der Triage bzw. Backlog-Triage wird für jedes Item im Backlog geprüft, ob es gelöscht werden kann oder noch relevant ist. Für gewöhnlich nehmen auch hier eher Stakeholder teil, die nicht an der technischen Umsetzung beteiligt sind. Eine Alternative zur Triage ist die Möglichkeit, regelmäßig alle Items im Backlog zu löschen, die ein bestimmtes Alter überschritten haben. Sinn von beiden Varianten ist der selbe: den Backlog nicht zu sehr anwachsen zu lassen, so dass das Queue Replenishing Meeting zur Priorisierung schneller vonstatten gehen kann.

Schließlich hat auch Kanban ein Meeting, in dem es um die Verbesserung des Prozesses selbst geht - den Operations Review. Die Durchführung kann prinzipiell unregelmäßig sein, sollte jedoch möglichst nicht nur das Entwicklungsteam selbst enthalten, sondern auch Teilnehmer aus allen Bereichen, mit denen das Team zusammenarbeitet. Somit wird sichergestellt, dass auch an den Schnittstellen des Teams nach außen Probleme auffallen und verbessert werden können.

Abgesehen von all diesen freiwilligen Meetings, sollte das Team hauptsächlich mit Hilfe des Kanban-Boards oder einem Tool zum elektronischen Tracking kommunizieren. Bei ersterem können Probleme in Form von blockierenden oder blockierten Tickets sowie Bottlenecks im Ablauf schnell erkannt und gemeinsam behoben werden. Das Gegenstück eines Bottlenecks, eine Station, die nicht ausgelastet ist, bietet ebenfalls Raum für Verbesserung, eventuell indem das WIP-Limit erhöht wird oder indem die Zahl der Teammitglieder für diese Station im Workflow reduziert wird. Das elektronische Tracking meint für gewöhnlich einen Issue-Tracker, der zusätzlich zur Auflistung aller Tickets mit Details meist ein virtuelles Board enthält. Best Practice bei Kanban ist das Verwenden von beidem zusammen.

## Fazit

Kanban unterscheidet sich von anderen agilen Prozessen, da es einen bestehenden Prozess verbessert, statt ihn zu ersetzen. Es bringt ebenso eine Verbesserung der Selbstorganisationsfähigkeiten der beteiligten Personen und gute Visualisierungsmöglichkeiten des Workflows und damit mehr Transparenz und Verständnis. Schließlich verlangt Kanban mehr Kooperation und Fokussierung, um die Vorhersagbarkeit von Terminen für den Kunden zu verbessern. Dies führt zu höherer Kundenzufriedenheit und Mitarbeitermotivation.