

# KANBAN IN DER PRAXIS



Für den Projekterfolg ist die Auswahl der passenden Vorgehensweise, die sich auch noch individuell kombinieren und anpassen lässt, von großer Relevanz. Ein mögliches Vorgehensmodell ist Kanban, das in der Theorie einfach zugänglich ist. Doch wie sieht es mit der Praxis aus?

| von **SEBASTIAN PEUSEN**

## KANBAN, WAS IST DAS EIGENTLICH?

Der Ausdruck „Kanban“ hat seinen Ursprung im Japanischen und setzt sich zusammen aus den Begriffen „kan“ für Signal und „ban“ für Karte. Entwickelt als eine Methodik aus dem Toyota-Produktionssystem<sup>1</sup> ist die dahinterstehende Idee, bei reduzierten Lagerbeständen einen gleichmäßigen Fluss im Produktionsablauf zu gewährleisten. David J. Anderson<sup>2</sup> hat, davon inspiriert, eine Übertragung von Kanban für IT-Projekte entwickelt. Auch hier werden weiterhin nach den Methoden der „Lean Production“ nur dann die Artefakte bereitgestellt, wenn die nächste Instanz diese zur Bearbeitung anfordert. Dieses „Pull-Prinzip“ ist die Basis für alle Kanban-Systeme.

Bei der Anwendung von Kanban baut man auf vier Grundprinzipien:

1. Starte mit dem, was du jetzt machst
2. Verfolge inkrementelle, evolutionäre Veränderung
3. Respektiere initiale Prozesse, Rollen, Verantwortlichkeiten und Job-Titel
4. Fördere Führungs- und Entscheidungskompetenz auf allen Ebenen in der Organisation

Die Operationalisierung beachtet dazu sechs Kernpraktiken:

1. Visualisiere den Arbeitsablauf
2. Limitiere die parallelen Arbeiten durch „WIP“-Limits (Work-in-progress)
3. Manage den Fluss
4. Mache Prozessregeln explizit
5. Implementiere Feedbackmechanismen
6. Führe gemeinschaftliche Verbesserungen durch

## WIE LÄSST SICH KANBAN IN DER PRAXIS UMSETZEN?

Die Umsetzung eines Softwareprojekts ist eine komplexe Aufgabe. Eine vorhandene Systemspezifikation soll über verschiedene Aufgabenbereiche so strukturiert werden, dass mit der technischen Umsetzung begonnen werden kann. Die notwendigen geplanten Arbeiten in den Aufgabenbereichen werden dabei visualisiert, indem sich jede einzelne Aufgabe als eine Karte auf einem sogenannten „Kanban-Board“ durch die Stationen (Status) des Arbeitsablaufs bewegt.

Mithilfe von Softwarewerkzeugen, wie zum Beispiel JIRA<sup>3</sup> oder Swift Kanban<sup>4</sup>, lassen sich die Aufgaben/Karten zusätzlich über weitere Zusammenhänge zueinander in Beziehung setzen. So ist es einfach möglich, Arbeitspakete hierarchisch in Form eines Projektstrukturplans zu gruppieren und Analysen durchzuführen, die mit analogen Kanban-Boards nicht möglich sind.

Unterschiedliche Rollen in einem Projekt arbeiten auf unterschiedlichen Ebenen des Projektstrukturplans in eigenen Arbeitsabläufen. Diese können durch jeweils ein eigenes Kanban-Board visualisiert und verfolgt werden. In den jeweiligen Boards ziehen sich die Teammitglieder mit freier Kapazität

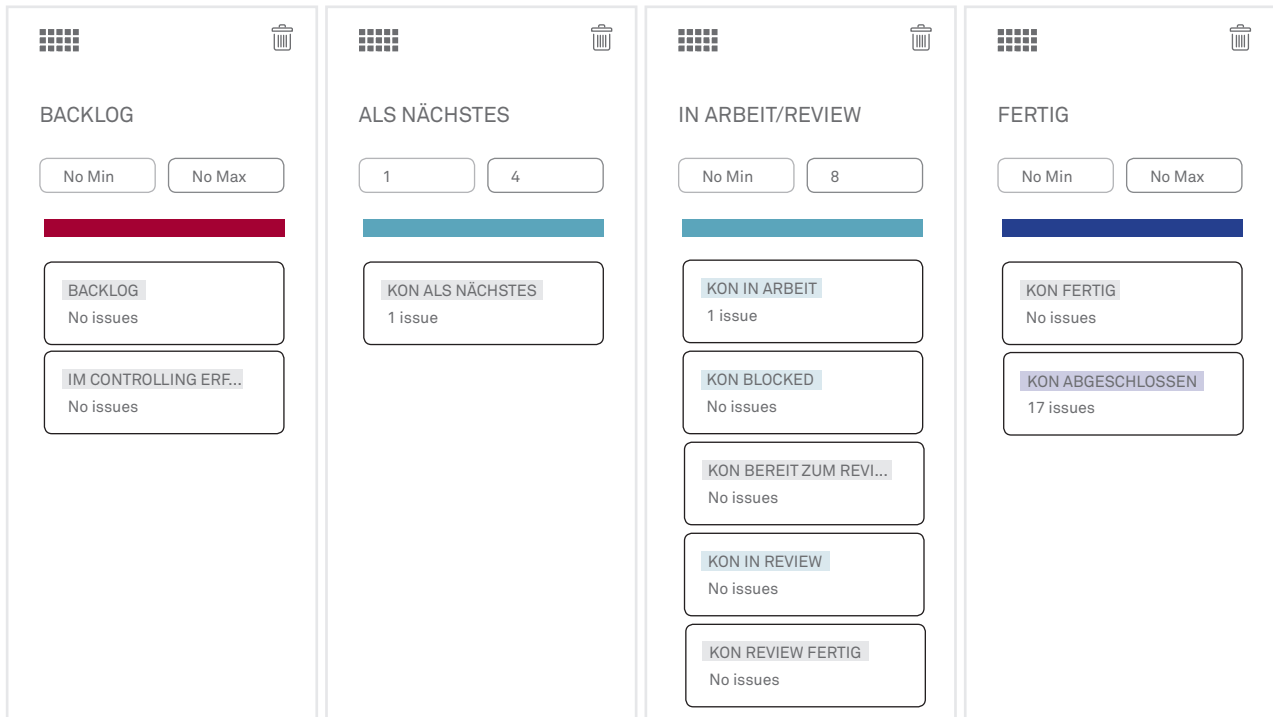


Abbildung 1: Vereinfachte Ansicht eines Jira-Kanban-Boards mit Beispielspalten und möglichen Status für Konstruktionsaufgaben

ein Arbeitspaket in Form einer Karte nach dem erwähnten „Pull-Prinzip“ zur Bearbeitung.

Die Systemspezifikation wird in mehrere Umsetzungskomponenten aufgeteilt. Jede dieser Komponenten setzt sich wieder aus mehreren Arbeitspaketen in Form von Konstruktions- und Designkomponenten zusammen. Das Ergebnis dieser Arbeitspakete sind dann detaillierte Entwicklungsaufgaben für die Realisierung.

Mit Softwarewerkzeugen können auch komplexe Arbeitsabläufe über verteilte, individuelle Kanban-Boards mit wenigen Spalten dargestellt werden und eine passende Übersichtlichkeit gewährleisten. Die Spalten stellen dabei den Lebenszyklus einer Aufgabe dar. Die Karte durchwandert die Spalten von links nach rechts. Jeder Spalte werden dabei die Status des Arbeitsablaufs zugeordnet (siehe Abbildung 1). Bei sehr vielen Karten auf einem einzelnen Kanban-Board bieten die oben genannten Softwarewerkzeuge neben der Möglichkeit, „Schwimmbahnen“ für die Karten auf dem Board zu definieren, zusätzlich die Möglichkeit, Filter anhand von weiteren Attributen der Aufgaben zu setzen.

Die Übergänge zwischen den Status werden durch Übergänge (Transitionen) in einer Arbeitsablaufmodellierung abgebildet. Mit

diesen Statusübergängen werden Regeln für erlaubte Statuswechsel verbunden. Die zu einem Prozessschritt gehörenden Arbeitsvorgänge sind erst dann abgeschlossen, wenn die abgestimmten und dokumentierten „Definitions-of-done“ (DoD) erfüllt sind. Die DoD sind ein Katalog von Beschreibungen, wann ein Artefakt alle Anforderungen zum Abschluss eines Prozessschritts erfüllt hat. Nur in diesem Fall ist der Übergang in den nächsten Status zulässig. Mit den Softwarewerkzeugen kann man dafür notwendige Prüfungen automatisieren. Auf diese Weise wird eine hohe und gleichbleibende Qualität der Ergebnisse erreicht (siehe Abbildung 2).

### WAS IST BEI ERSTELLUNG DER ARBEITSABLÄUFE ZU BEACHTEN?

Bei der Festlegung der Arbeitsabläufe sollten die Status der Arbeiten in einem ausreichenden Detailgrad definiert werden, um:

- Statusfortschritte sichtbar zu machen,
- die Zusammenarbeit der Teammitglieder zu visualisieren; es sollten Übergabepunkte zwischen den Statuswechseln eingeplant werden,
- eine Flussoptimierung der Arbeiten durch gleichmäßige Verteilung der Karten über die unterschiedlichen Status zu erzielen,

- Optimierungsmöglichkeiten innerhalb der Arbeitsabläufe zu erkennen und eventuell notwendige Ergänzungen um weitere Status zu identifizieren.

Die Struktur der Aufgaben, die in Form der Karten in einem Arbeitsablauf abgearbeitet werden, verdient dabei ebenfalls Beachtung. Damit zur Optimierung der Arbeitsabläufe in Kanban Kennzahlen herangezogen werden können, muss eine grundsätzliche Vergleichbarkeit der Aufgaben in Größe und Komplexität gegeben sein. Die Herausforderung besteht darin, für eine ursprünglich heterogene Menge von Aufgaben einen passenden Zuschnitt zu finden. Zur Planung der Arbeiten und Kapazitäten hat sich für eine Karte ein Arbeitspaket in einer Größe als vorteilhaft herausgestellt, die ein Teammitglied in einer Woche beenden kann. So kann das Bereitstellen neuer Aufgaben über wöchentliche Planungsmeetings verteilt und der Aufwand zur Pflege der einzelnen Karten reduziert werden. Dies ist in der Praxis natürlich nicht immer möglich, aber man sollte untersuchen, ob sich:

- kleinere Aufgaben (bis 20 Stunden Aufwand) zusammenfassen und
- größerer Aufgaben (ab 60 Stunden Aufwand) aufteilen lassen.

Zusätzlich sollten die möglichen Abhängigkeiten der Aufgaben untereinander identifiziert werden, bevor sie in einer sinnvollen Reihenfolge zur Bearbeitung freigegeben werden. Unter diesen Rahmenbedingungen erhält man auch bei ungleicher Aufwandsgröße der Aufgaben eine gute Vergleichbarkeit durch Betrachtung des Verhältnisses:

*Laufzeit der Aufgabe durch den Arbeitsablauf*

*Geplanter Aufwand der Aufgabe*

als Kennzahl für die Größe einer Aufgabe.

### UMGANG MIT BLOCKADEN

Blockaden führen zu Verzögerungen in der Umsetzung. Hierbei gilt es, zwischen Arbeitspaketen, die auf den nächsten Bearbeitungsschritt warten, und solchen, die in sich selbst blockiert sind, zu unterscheiden. Durch das „Pull-Prinzip“ in Kanban gibt es immer Karten, die in einem Übergabestatus warten, bis sie zur Weiterbearbeitung aufgenommen werden. Im Gegensatz dazu sind blockierte Karten aktuell eigentlich in Bearbeitung bei einem Teammitglied, können aber durch ungeplante Umstände nicht weiterbearbeitet werden. Mögliche Ursachen sind Abhängigkeiten zwischen Aufgaben, zu Teammitgliedern oder zu Dritten. Außerdem können fehlende Informationen und unscharfe Anforderungen dazu führen, dass Klärungen notwendig sind und die Arbeit in dieser Zeit nicht fortgeführt werden kann.

Um in der Zukunft Blockaden zu vermeiden, gilt es, die Ursachen zu identifizieren. Die Gründe werden auf den Karten dokumentiert und gesammelt. Mit Softwarewerkzeugen bieten sich viel mehr Möglichkeiten, die Blockaden zu erfassen und zu beschreiben. Zum Beispiel können die Ursachen von Blockaden anhand von Attributen gruppiert werden, wodurch die weiterführenden Analysen und Auswertungen erleichtert werden. Regelmäßig werden die Auswirkungen von Blockaden in Retros-

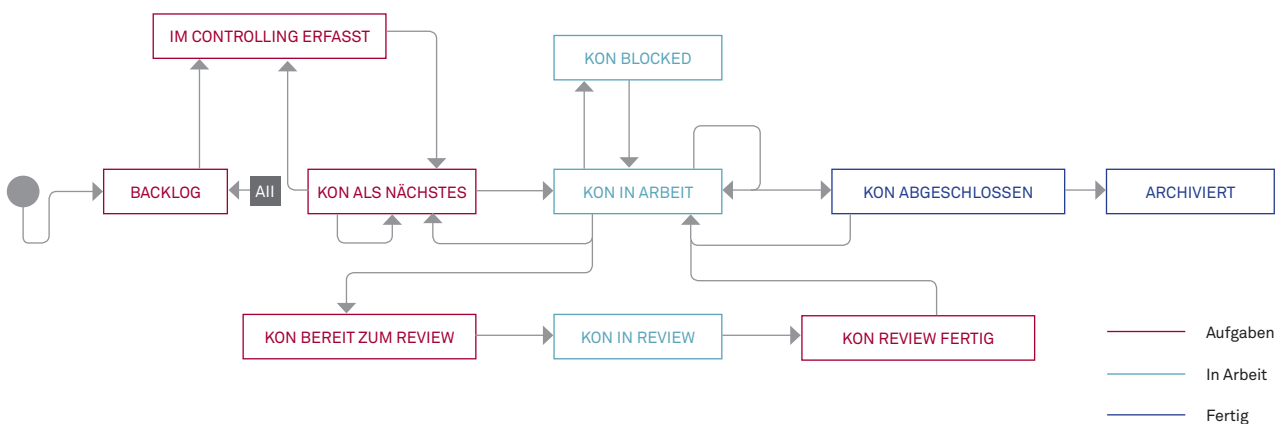


Abbildung 2: Beispiel-Arbeitsablauf für Konstruktionsaufgaben

pektiven analysiert und bewertet. Durch Einteilen der Ursachen in Kategorien lassen sich dann für wiederkehrende Blockaden präventive Maßnahmen entwickeln. Unter den Beteiligten wird dann gemeinschaftlich entschieden, welche dieser Maßnahmen durchgeführt werden soll. Alle Erkenntnisse werden parallel im Risikomanagement widergespiegelt. Auf diese Weise wird das Gesamtsystem sukzessive optimiert.

## UMGANG MIT FLASCHENHÄLSEN

Als Flaschenhalse werden die Status bezeichnet, in denen sich mehr Karten zur weiteren Bearbeitung ansammeln, als in gleicher Zeit von den Folgeprozessen abgearbeitet werden. Es bildet sich ein klassischer Stau. Dies kann zum Beispiel in der IT bei Review- und anderen Qualitätssicherungsprozessen oder bei Integrationsprozessen der Fall sein. Durch die Visualisierung der Arbeit in Form von Karten auf einem Kanban-Board wird schnell sichtbar, wo sich ein Stau bildet und man im aktuellen Arbeitsablauf einen Flaschenhals hat. Flaschenhälse führen zu Verzögerungen in der Umsetzung, weil die Durchlaufdauer einer Karte durch die Wartezeiten vor dem Flaschenhals dann merklich länger wird als der geplante Aufwand.

Maßnahmen zur Verminderung der Auswirkungen von Flaschenhälsen zielen darauf ab, den Fluss an der Stelle im Arbeitsablauf wieder zu verbessern. Dies geschieht einerseits dadurch, dass mehr Kapazität geschaffen wird, um Karten an dieser Stelle zu bearbeiten, und andererseits dadurch, dass der Zufluss in diesen Status limitiert wird.

Eine Erhöhung der Kapazitäten im System ist prinzipiell durch eine Vergrößerung des Teams möglich. Oft ist es jedoch so, dass das kurzfristig nicht möglich ist oder spezielles Wissen gebraucht wird, eventuell auch nur für sehr kurze Zeit. In diesem Fall ist es nicht sinnvoll, auf eine Vergrößerung des Teams zu setzen, wenn man die entsprechenden Kosten mit den Kosten für die Verzögerung vergleicht. Auch mit einem festen Team kann mehr Kapazität geschaffen werden, wenn die Einsatzgebiete der Teammitglieder flexibler disponiert werden können. Dafür ist es wichtig, die Wissensverteilung im Team zu

fördern, so dass bei der Bildung eines Staus an anderer Stelle freie Ressourcen bei der Abarbeitung der Karten unterstützen können. Diese Diversifizierung der Fähigkeiten trägt zur weiteren Entwicklung der Teammitglieder bei, stärkt zusätzlich das Verantwortungsgefühl für das Gesamtprojekt und verbessert damit letztlich die Qualität der Arbeit. Dadurch wird ebenfalls einer Überlastung von Schlüsselwissensträgern an kritischen Stellen vorgebeugt, was ansonsten einen entsprechenden Flaschenhals noch weiter verschlechtern würde.

Eine Verringerung des Arbeitszuflusses an einen Flaschenhals kann über die WIP-Limits gesteuert werden. Die Menge an paralleler Arbeit wird dabei limitiert. Nicht nur anhand der natürlichen Kapazitäten des Teams für einen Aufgabenbereich, sondern zusätzlich durch einen Wert für die Regelung der Systemkapazität. Zu hohe WIP-Limits im System können zu wartenden Aufgaben vor einem Flaschenhals führen und damit zu den genannten Verzögerungen. Zu niedrige WIP-Limits können an anderen Stellen im Arbeitsablauf zu unterbeschäftigten Teammitgliedern führen und damit zu einer schlechten Produktivität. Das kontinuierliche Analysieren des Durchlaufs und in Folge das korrekte Festlegen und laufende Anpassen der WIP-Limits für die individuellen Arbeitsabläufe ist eine Herausforderung und gleichzeitig Kernelement für Optimierungen bei Kanban.

## FAZIT

Mit Kanban hat man die Möglichkeit, bestehende Arbeitsabläufe erst zu übernehmen, dann methodisch zu analysieren und anschließend zu optimieren. Mit der Unterstützung von Softwarewerkzeugen können auch komplexe Zusammenhänge abgebildet werden, die Pinnwände, Whiteboards und Haftnotizen nicht darstellen können. In einer Zeit, in der durch die Herausforderungen der Digitalisierung bisherige Arbeitsweisen infrage gestellt werden, bietet Kanban praktikable Methoden zur evolutionären Verbesserung. ●



1 Taiichi Ohno: Das Toyota-Produktionssystem. Campus, Frankfurt am Main 1993, ISBN 3-593-37801-9.

2 David J. Anderson: Agile Management for Software Engineering - Applying the Theory of Constraints for Business Results. Prentice Hall, 2004, ISBN 0-13-142460-2.

3 <https://www.atlassian.com/de/software/jira> (abgerufen am 18.02.2020).

4 <https://www.digite.com/swiftkanban/> (abgerufen am 18.02.2020).