

Kapitel 1

Über dieses Buch

1.1 Einleitung

Dieses Buch behandelt das Vorgehensmodell Kanban und seinen Einsatz in Softwareentwicklungsprojekten.

Kanban ist ein Vorgehensmodell der schlanken Softwareentwicklung (*Lean¹ Software Development*). Erstmals der Öffentlichkeit vorgestellt wurde Kanban im Jahre 2007 von David J. Anderson, der seitdem als der Vater von Kanban für die Softwareentwicklung gilt (siehe [1]). Bei Kanban handelt es sich also zum Entstehungszeitpunkt dieses Buches um ein noch junges Vorgehensmodell, über dessen erfolgreichen Einsatz es eine beständig wachsende Anzahl von Berichten gibt.

Dieses Buch kann die Atmosphäre, die durch den Einsatz von Kanban in einem Softwareentwicklungsprojekt

¹ In diesem Buch wird das englische Wort *lean* durchgängig mit *schlank* übersetzt.

entsteht, nur andeutungsweise vermitteln. Diese Atmosphäre ist projektindividuell und von einer Vielzahl von Faktoren, allen voran den beteiligten Personen, abhängig. Nutzen Sie dieses Buch, um gemeinsam mit Ihrem Team Ihre ganz eigene Variation von Kanban zu gestalten!

Die Abbildungen 1.1–1.4 geben jedoch einen ersten Eindruck davon, wie der Einsatz von Kanban in seinen Grundzügen aussieht. Diese vier Abbildungen bilden zusammen einen Comic von Henrik Kniberg, der unter dem Titel *Ein Tag in Kanban-Land* bekannt geworden ist (siehe [2]). Der Comic illustriert die vier Elemente, die Kanban als Vorgehensmodell charakterisieren.

- Arbeit wird genommen, nicht gegeben.
- Mengen werden limitiert.
- Informationen werden veröffentlicht.
- Arbeitsabläufe werden kontinuierlich verbessert.

Der folgende Abschnitt 1.2 beschreibt zunächst die Zielgruppe für dieses Buch, bevor Abschnitt 1.3 seinen Aufbau erläutert. Abschnitt 1.4 schließlich enthält einen wichtigen Gebrauchshinweis, der für alle folgenden Kapitel gilt.

1.2 Zielgruppe

Dieses Buch richtet sich an alle Personen, die sich über den Einsatz von Kanban in einem Softwareentwicklungsprojekt informieren wollen. Dabei ist es nicht auf Personen in einer bestimmten Rolle beschränkt. Es kann darüber hinaus (durch seinen praktischen Anteil) unerfahrenen Anwendern bei der Entscheidung helfen, Kanban in einem Softwareentwicklungsprojekt einzuführen und (durch seinen

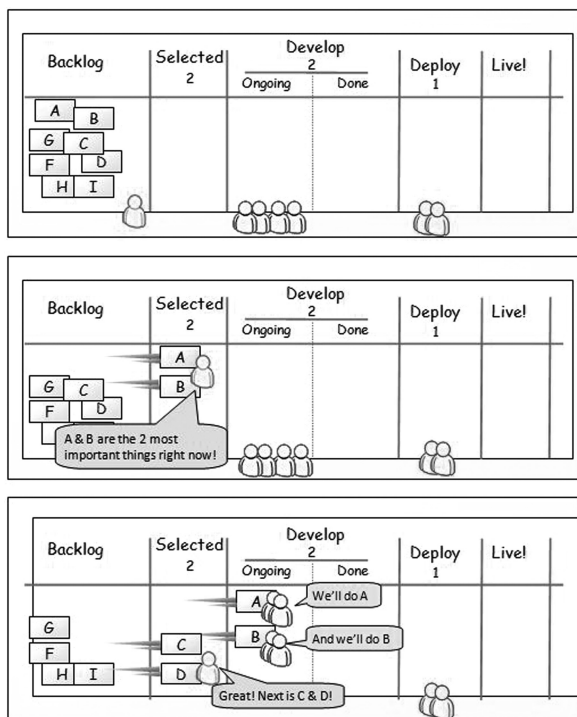


Abb. 1.1 Ein Tag in Kanban-Land, Teil 1: Alle relevanten Phasen der Wertschöpfungskette in einem Softwareentwicklungsprojekt sind (von links nach rechts) visualisiert. Für jede Phase ist ein oberes Limit für die Anzahl von Aufgaben gesetzt, die sich gleichzeitig in dieser Phase befinden dürfen. Aufgaben werden aus einem Vorrat ausgewählt und für den Durchlauf der Wertschöpfungskette zur Verfügung gestellt. Selbstorganisiert und eigenverantwortlich werden die zur Verfügung gestellten Aufgaben angenommen. Die Limitierung der Anzahl von Aufgaben für jede Phase wird dabei beachtet. Ein in seiner Breite limitierter Fluss von Aufgaben durch die Phasen der Wertschöpfungskette entsteht.

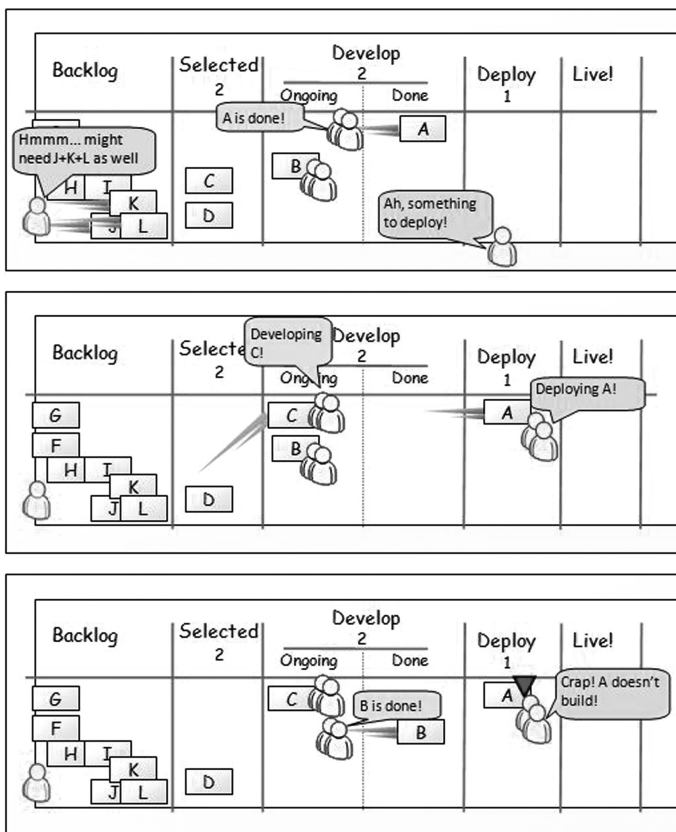


Abb. 1.2 Ein Tag in Kanban-Land, Teil 2: Die erste Aufgabe erreicht die letzte Phase der Wertschöpfungskette. Wann immer eine Aufgabe eine Phase verlässt, entsteht Platz für eine neue Aufgabe. Die an einer Phase beteiligten Personen entscheiden selbstorganisiert und eigenverantwortlich darüber, wann und durch welche der verfügbaren Aufgaben dieser Platz neu besetzt wird. In der letzten Phase der Wertschöpfungskette kommt es nun zu einer Störung im Aufgabenfluss.

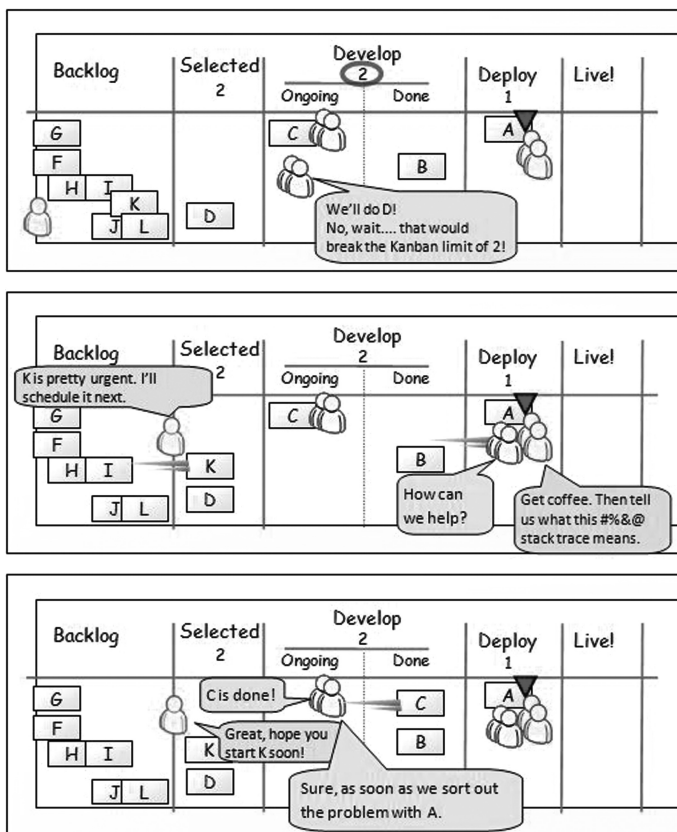


Abb. 1.3 Ein Tag in Kanban-Land, Teil 3: Die Störung im Aufgabenfluss wirkt sich auf vorhergehende Phasen der Wertschöpfungskette aus. Die Limitierung der Anzahl von Aufgaben in jeder Phase provoziert einen Rückstau von Aufgaben. Dieser Rückstau geht von der Phase aus, in der die Störung aufgetreten ist. Personen, die dadurch ihre Arbeit nicht fortsetzen können, bieten ihre Hilfe bei der Beseitigung der Störung an.

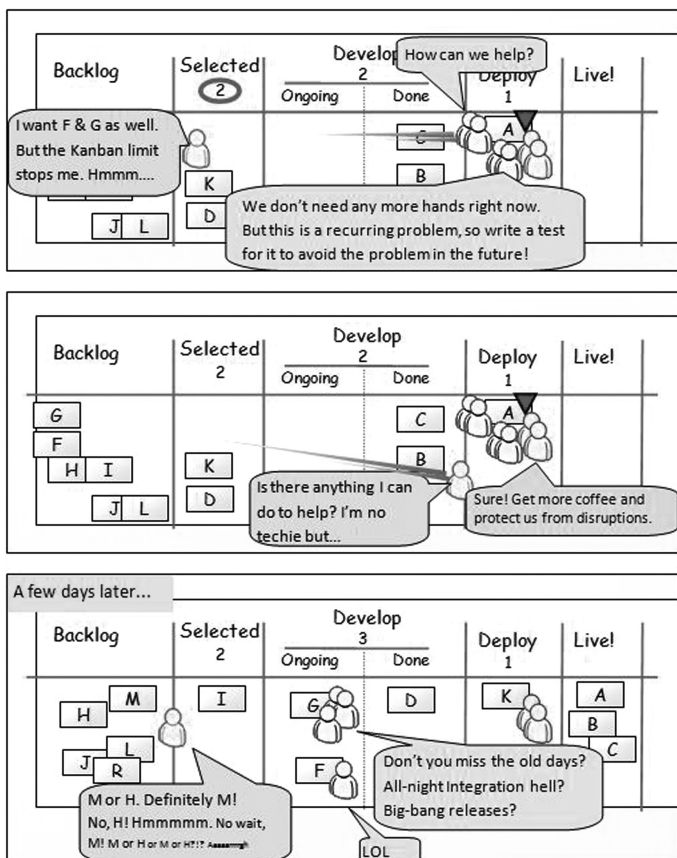


Abb. 1.4 Ein Tag in Kanban-Land, Teil 4: Die Störung im Aufgabenfluss bewirkt, dass der gesamte Aufgabenfluss zum Stillstand kommt. Alle Personen helfen dabei, die Störung zu beseitigen. Zusätzlich ergreifen sie Maßnahmen, die diese Störung in Zukunft verhindern sollen. Nach Beseitigung der Störung stellt sich wieder ein Aufgabenfluss ein. Die Verteilung von Personen auf Phasen sowie die Limitierung der Anzahl von Aufgaben für jede Phase wurden als Konsequenz aus der Störung angepasst.

theoretischen Anteil) erfahrenen Anwendern von Kanban eine neue Sicht auf dieses Vorgehensmodell eröffnen.

Aufgrund dieses Anspruchs beinhaltet dieses Buch zwar eine in sich geschlossene, aber keine vollständige Übersicht über Kanban in der Softwareentwicklung. Es richtet sich nicht an Personen, die sich für umfassende theoretische Grundlagen (dies sind neben Grundlagen auf dem Gebiet der Softwareentwicklung etwa auch Grundlagen auf dem Gebiet der Psychologie oder des Risikomanagements) von Kanban interessieren.

Auf einigen Gebieten außerhalb der Softwareentwicklung ist Kanban teilweise ein seit Jahrzehnten etablierter Begriff. Dieses Buch beschränkt sich (abgesehen von einem Exkurs in Abschnitt 3.4.1) auf die Beschreibung von Kanban in der Softwareentwicklung.

1.3 Aufbau

Dieses Buch beantwortet in den folgenden sechs Kapiteln abstrakte, allgemeine und konkrete Fragen rund um Kanban. Jedes Kapitel beginnt mit einer kurzen Einleitung und endet mit Hinweisen auf weiterführende Literatur.

Was ist ein Vorgehensmodell? Kapitel 2 definiert zunächst die Begriffe *Wert*, *Element*, *Technik* und *Vorgehensmodell*. Dies geschieht unabhängig von Kanban; diese Begriffe sind jedoch zentral auch für die folgenden Kapitel und werden in ihnen immer wieder aufgegriffen.

Auf welchen Werten basiert Kanban? Kapitel 3 erläutert die Grundlagen von Kanban (diese haben ihren Ursprung nicht in der Softwareentwicklung, sondern in der

Automobilindustrie). Aus diesen Grundlagen resultiert ein System von Werten für Kanban in der Softwareentwicklung. Zur Illustration wird es mit einem anderen Wertesystem, dem Wertesystem der agilen Softwareentwicklung, verglichen.

Welche Elemente charakterisieren Kanban? Kapitel 4 erklärt die vier charakteristischen Elemente von Kanban ausführlich. Anhand dieser Elemente wird Kanban sowohl in das Wertesystem der agilen Softwareentwicklung als auch in das Wertesystem der schlanken Softwareentwicklung eingeordnet.

Was für Ansprüche stellt Kanban an typische Phasen und Rollen der Softwareentwicklung? Kapitel 5 beschreibt Kanban aus dem Blickwinkel typischer Phasen und Rollen in einem Softwareentwicklungsprojekt: Requirements Engineering, Entwicklung, Qualitätssicherung und Projektmanagement sowie ergänzend dazu aus der Sicht eines Auftraggebers.

Welche Techniken eignen sich für den Einsatz mit Kanban? Kapitel 6 nennt für jede Phase und Rolle Beispiele von Techniken, die für den Einsatz von Kanban in einem Softwareentwicklungsprojekt geeignet sind. Diese Beispiele werden durch Erfahrungsberichte aus einem realen Softwareentwicklungsprojekt illustriert.

Wie wird Kanban eingeführt? Kapitel 7 behandelt abschließend die Einführung von Kanban in einem Softwareentwicklungsprojekt. Kanban ist in diesem Fall Change Management.

Dieses Buch enthält eine Vielzahl von Beispielen. Jedes Beispiel wird in einem abgesetzten und unterlegten Textkasten gezeigt.

Exemplarisch betrachtet dieses Buch die folgenden Phasen der Softwareentwicklung.

1. Requirements Engineering
2. Entwicklung
3. Qualitätssicherung

Diese Phasen bilden (in dieser Reihenfolge) die Wertschöpfungskette für ein Softwareentwicklungsprojekt aus der Sicht eines Auftragnehmers. Der Begriff *Softwareentwicklung* ist dabei allgemein zu verstehen und umfasst Neuentwicklung, Weiterentwicklung oder auch Wartung einer Anwendung.

Ebenfalls exemplarisch betrachtet werden die folgenden Rollen des Auftragnehmers, die an den Phasen der Wertschöpfungskette beteiligt sind.

- Requirements Engineering
- Entwicklung
- Qualitätssicherung
- Projektmanagement

Drei dieser Rollen korrespondieren (auch in ihrer Bezeichnung) mit jeweils einer Phase der Wertschöpfungskette (im Allgemeinen können jedoch mehrere Rollen an einer Phase beteiligt sein). Die Personen, auf die sich diese Rollen verteilen, bilden das Team des Auftragnehmers und werden in diesem Buch kurz als *Team* bezeichnet. Das Team

arbeitet für einen Auftraggeber, dem in diesem Buch auch die Rolle des Anwenders zugeschrieben wird.

Englische Begriffe und Textpassagen werden in diesem Buch gleichberechtigt zu deutschen verwendet, da zum einen ein Großteil sowohl der Literatur zu Kanban als auch der weiterführenden Literatur englischsprachig ist, zum anderen jede Übersetzung oder Paraphrasierung eine Bedeutungsänderung beinhalten kann.

1.4 Gebrauchshinweis

Jedes Softwareentwicklungsprojekt hat projektindividuelle Rahmenbedingungen. Während grundlegende Werte und Elemente von Kanban übergreifend über Projekte beständig sind, wird sich Kanban jedoch von Projekt zu Projekt in verschiedenen Techniken manifestieren.

Es ist wichtig, diese Freiheit einer projektindividuellen Ausprägung von Kanban als Chance zu begreifen und auch als solche wahrzunehmen. Zur Orientierung mag ein Zitat von David J. Anderson dienen (siehe [1]).

Kanban is giving people permission to think for themselves. It is giving people permission to be different: different from the team across the floor, on the next floor, in the next building, and at a neighboring firm. It is giving people permission to deviate from the textbook.

Auch die Auswahl von Themen und Inhalten in diesem Buch ist offensichtlich subjektiv; sie basiert auf theoretischer und praktischer Beschäftigung mit Kanban in einem konkreten Softwareentwicklungsprojekt. Jede Leserin und jeder Leser ist dazu angehalten, selbst darüber zu entscheiden, welche Themen und Inhalte sich auf ihre oder

seine ganz persönlichen Rahmenbedingungen übertragen oder anwenden lassen.

1.5 Literaturhinweise

David J. Anderson beschreibt in [1] ausführlich die Erfahrungen aus seiner Arbeit mit Kanban. Er geht dabei auch auf theoretische Grundlagen ein und beleuchtet Kanban aus einer Vielzahl von Blickwinkeln. Sein Buch unterscheidet sich von dem vorliegenden dadurch, dass bei David J. Anderson ein Bezug zu einem Wertsystem von Kanban und konkrete Techniken für typische Phasen der Softwareentwicklung weniger im Vordergrund stehen.

Literaturverzeichnis

1. Anderson, D. J.: Kanban – Successful Evolutionary Change for Your Technology Business. Blue Hole Press (2010)
2. Kniberg, H.: One day in Kanban land. (2009)
<http://blog.crisp.se/henrikkniberg/2009/06/26/1246053060000.html>. Stand 11.03.2011



<http://www.springer.com/978-3-642-22594-9>

Kanban für die Softwareentwicklung

Epping, Th.

2011, XVI, 104 S. 24 Abb., Softcover

ISBN: 978-3-642-22594-9