# Entscheidung Vorgehensmodell

## Einleitung

Bevor man mit dem eigentlichem Projekt beginnt ist es wichtig ein geeignetes Vorgehensmodell für unsere Softwareentwicklung zu wählen. Ein passendes Vorgehensmodell zu finden ist wichtig, da es die Softwareentwicklung übersichtlicher gestaltet und die Komplexität unseres Problems beherrschbarer zu machen.

## Klassische und agile Vorgehensmodelle

Im Allgemeinen unterscheidet man klassische und agile Vorgehensmodelle. Klassische Vorgehensmodelle konzentrieren sich auf den formalisierten Prozess, während agile Vorgehensmodelle die eigentliche Entwicklung und die Kommunikation zwischen Entwickler und Kunde im Mittelpunkt sehen. Ein typischer Vertreter des klassischen Vorgehensmodells ist das Wasserfallmodell, agile Vorgehensmodelle werden meistens durch Scrum oder Kanban vertreten.

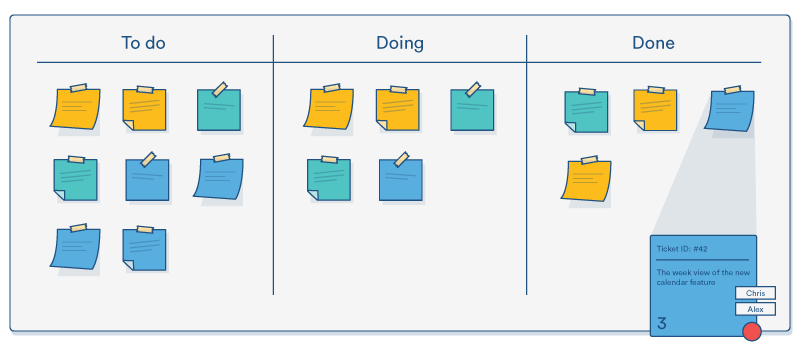
Da in unserem Entwicklungsprozess die Kommunikation zwischen Entwickler und Kunde im Mittelpunkt steht entscheiden wir uns ganz klar für ein **agiles Vorgehensmodell**.

## Scrum oder Kanban

Nach verschiedenen Vergleichen von agilen Vorgehensmodellen sind Scrum und Kanban besonders in unseren Fokus gerückt, da auch bereits die allgemeinen Vorgehensschritte der Modelle bekannt sind.  
Da unser Team nur aus 4 Personen besteht wäre Scrum zu aufwendig umzusetzen, da Scrum erst ab einer Teamgröße von +5 \*Personen wirklich Sinn macht. Bei Kanban hingegen gibt es keine wirklichen Beschränkungen wie groß das Team sein soll. Neben der Teamgröße ist ein weiterer Entscheidungsfaktor, dass Scrum vor allem für sehr komplexe Software verwendet wird. Unser Programm ist nicht hochkomplex und darum eignet sich Kanban wiederum besser.  
Aufgrund der oben genannten Gründe, haben wir uns für das Vorgehensmodell **Kanban** entschieden.

## Wie funktioniert Kanban?

[Agile Softwareentwicklungsteams](https://de.atlassian.com/software/jira/agile-project-management) ordnen die Anzahl der Work-in-Progress-Aufgaben der Kapazität des Teams zu. So kann das Team flexibler planen, Aufgaben schneller erledigen, eine klare Fokussierung erreichen und für Transparenz im gesamten Entwicklungszyklus sorgen.



Kanban ist eine der beliebtesten Methoden bei der Softwareentwicklung, die heute von agilen Teams übernommen wird. Kanban bietet verschiedene zusätzliche Vorteile bezüglich Aufgabenplanung und Durchsatz bei Teams jeder Größe.  
Ein Kanban-Team konzentriert sich nur auf die Arbeit, die aktiv ausgeführt wird. Sobald das Team ein Aufgabenelement abgeschlossen hat, wird das nächste Aufgabenelement aus dem obersten Bereich des [Backlogs](https://de.atlassian.com/agile/backlogs) entnommen. Der [Product Owner](https://de.atlassian.com/agile/product-management) kann Prioritäten für Aufgaben im Backlog beliebig neu festlegen, ohne das Team zu unterbrechen, da sich alle Änderungen, die das aktuelle Aufgabenelement nicht betreffen, nicht auf das Team auswirken. Solange der Product Owner dafür sorgt, dass die wichtigsten Aufgabenelemente oben im Backlog stehen, kann das Entwicklerteam dem Unternehmen den größtmöglichen Wert bereitstellen. Die für [Scrum](https://de.atlassian.com/agile/scrum) typischen Iterationen mit fester Länge sind hier also nicht erforderlich.

## Kanban in unserem Team

Da wir fast jede Stunde den Raum wechseln müssen, ist es wichtig, dass wir kein klassisches Kanban-Board besitzen, sondern Kanban über ein Webprojekt angewendet wird. Nach einer Recherche sind wir auf die Seite easyproject.com gestoßen. Diese Webseite ermöglichte es uns Kanban Tasks digital zu erstellen. Die Benutzer konnten einzeln unter der Seite <http://projektfu.easyproject.com/> erstellt werde und so ist es relativ einfach gewesen, den Verlauf des Projektes zu verfolgen.   
Es gab nur einen kleinen Nachteil bei der Software, da die eingetragenen Stunden nach ungeklärten Gründen immer zu hoch waren. Da es jedoch nicht zwingend relevant war, die Stunden zu verfolgen ist dieses Tool äußert hilfreich bei der Umsetzung des Projektes gewesen.