

## 2.5 Teilen und Tauschen

Kontext	Bei der Vorbereitung einer Entscheidung hat der Fokus bislang auf der ZIELEXPLIKATION (2.1) gelegen sowie auf DEM BLICK ZURÜCK (2.3) und DEM BLICK VON AUSSEN (2.4), die beide vor allem dazu gedient haben, mögliche Risiken zu identifizieren. Vermutlich haben sich auch schon erste Ideen für mögliche Handlungsoptionen ergeben.
Problem	<b>Nur wenn wirklich eine Alternative besteht, ist eine echte Entscheidung möglich. Wie können wir zu unterschiedlichen Handlungsoptionen gelangen?</b>
Hintergrund	Typischerweise gibt es bei Entscheidungen einige mehr oder weniger naheliegende Handlungsoptionen. Dies sind Lösungen, die uns sofort in den Sinn kommen, oder Alternativen, die sich in einer bestimmten Situation immer wieder bieten. Dabei dürfen wir nicht vergessen, dass fast immer auch der Status quo eine Handlungsoption darstellt, die eben darauf hinausläuft, alles beim Alten zu belassen. Das muss im Übrigen nicht für immer sein. Denkbar ist auch, es für den Moment beim Status quo zu belassen und später die Frage nach möglichen Änderungen noch einmal zu stellen. Im Rahmen eines ITERATIVEN ENTSCHEIDUNGSPROZESSES (6.4) ist das ein übliches Vorgehen.
	Klar ist aber, dass eine Entscheidung von echten Alternativen lebt. Wenn wir nur die naheliegenden Handlungsoptionen betrachten, entgehen uns möglicherweise Lösungen, die die gewünschten Kriterien tatsächlich gut erfüllen. Wir sollten uns also nicht zu schnell auf einige wenige Handlungsoptionen einschränken.
	Das Vorhandensein vieler Handlungsoptionen allein macht eine Entscheidung nicht automatisch erfolgversprechend. Eine große Zahl von Optionen führt primär zu einem schwierigeren Entscheidungsprozess. Das ist aber gar nicht unser Wunsch. Wichtig ist vielmehr, dass uns Optionen bekannt sind, die sich hinsichtlich der Umsetzung und Bewertung deutlich unterscheiden.
Lösung	Oft lassen sich neue Handlungsoptionen gewinnen, indem wir eine bekannte Handlungsoption in Teilespekte zerlegen und die typischen Lösungsansätze für diese Teilespekte infrage stellen. Teilespekte können beispielsweise die Komponenten einer Architektur, Aktionen im Rahmen einer Gesamtstrategie oder auch Features innerhalb einer komplexen Anforderung sein. Durch den Austausch einzelner Lösungselemente durch alternative Ideen entstehen neue Optionen.
Details	In der Softwareentwicklung gibt es das Prinzip namens »Teile und herrsche« ( <i>divide and conquer</i> ). Es besagt, dass sich Lösungen finden lassen, indem man eine Gesamtaufgabe in beherrschbare Teilespekte zerlegt und diese einzeln angeht. Ähnlich verhält es sich hier. Das Zer-

legen der Gesamtentscheidung in einzelne Aspekte bietet uns die Möglichkeit, einzelne dieser Aspekte zu variieren und so neue Handlungsoptionen zu generieren:

- Falls noch keine Handlungsoptionen bekannt sind, ist der Status quo, also die existierende Lösung, der Ausgangspunkt. Hier können wir uns fragen, wodurch diese Lösung charakterisiert ist. Aus welchen Bestandteilen ist sie zusammengesetzt? Welche Teilaspekte machen in der Summe die existierende Lösung aus?
- Sobald wir einige Teilaspekte identifiziert haben, können wir uns überlegen, welche möglichen Varianten oder Alternativen für diese Teilaspekte infrage kommen. Durch die Kombination unterschiedlicher Varianten erhalten wir neue Handlungsoptionen, wenngleich wir dabei natürlich berücksichtigen müssen, welche Abhängigkeiten zwischen den verschiedenen Teilaspekten bestehen.
- Sind bereits zu Beginn mehrere Handlungsoptionen bekannt, haben wir darüber hinaus die Möglichkeit, Teilaspekte der unterschiedlichen Optionen miteinander zu kombinieren. Auch so entstehen Mischformen von Lösungen, die nicht immer, aber manchmal sinnvolle, neue Handlungsoptionen ergeben.

An dieser Stelle kommt auch unsere Intuition ins Spiel. Sie erlaubt es uns, sinnvolle neue Handlungsoptionen als solche zu erkennen und unsinnige Pfade nicht weiter zu verfolgen.

## 1. Methodenauswahl

*Beispiele*

Ein neu zusammengestelltes Team übernimmt die Verantwortung für die Entwicklung einer Applikation. Neben einigen organisatorischen Dingen stellt sich für das Team auch die Frage nach der Methode, nach der es vorgehen möchte. Grundsätzlich besteht der Wunsch nach einem agilen Vorgehen. Schnell wird von mehreren Teammitgliedern Scrum ins Spiel gebracht.

Eine Entwicklungsmethode umfasst viele Facetten, die potenziell variiert werden können. Dies gilt auch für Scrum, zumal es sich dabei nicht um einen Prozess, sondern um ein Prozessframework handelt, das bestimmte Dinge bewusst offenlässt. Zum Beispiel können die Länge der Sprints oder der Ablauf der Retrospektive variiert werden, wodurch im Prinzip neue Handlungsoptionen entstehen, was die Auswahl der Methode angeht.

Man kann auch weitergehen und Teilaspekte von Scrum durch Elemente ersetzen, die aus anderen Methoden bekannt sind. Hier ist allerdings Vorsicht geboten, weil so nicht automatisch sinnvolle Handlungsoptionen entstehen. Ersetzt man beispielsweise die Rolle des Scrum Masters – die ein wesentlicher Aspekt von Scrum ist – durch die Rolle eines klassischen Projektleiters, erhält man vielleicht theoretisch

eine weitere Handlungsoption, aber keine, die in irgendeiner Form sinnvoll wäre. Bei dem resultierenden Methodenmix passen einfach die Teilaufgaben nicht mehr zusammen.

Trotzdem ist es nicht von vornherein ausgeschlossen, auch Scrum selbst zu variieren. So ist es denkbar, das vorgesehene Taskboard – ebenfalls ein wesentlicher Aspekt von Scrum – durch ein anderes zu ersetzen, beispielsweise durch ein Board, wie es in Kanban vorgesehen ist. Das Kanban-Taskboard unterscheidet sich vom Scrum-Taskboard vor allem dadurch, dass es flexibler in der Verwendung der Spalten für die einzelnen Arbeitsschritte ist. Durch diese Variation entsteht ein Methodenmix, der sicher nicht in jeder Situation geeignet ist, der aber eine weitere Option darstellt, die grundsätzlich infrage kommen kann.<sup>3</sup>

## 2. Softwarearchitektur

Für eine neue Applikation soll eine Architektur definiert werden. Ein Vorschlag liegt bereits auf dem Tisch. Es handelt sich dabei um eine klassische Schichtenarchitektur.

Alternativen dazu lassen sich finden, indem die einzelnen Bestandteile der Architektur infrage gestellt werden. Was passiert, wenn die Datenbank ausgetauscht wird, zum Beispiel eine SQL-Datenbank durch eine Non-SQL-Datenbank? Oder die Middleware? Oder die Präsentationsschicht? Dabei ist natürlich zu bedenken, dass es zwischen den einzelnen Komponenten Abhängigkeiten geben kann. Durch die unterschiedlichen Kombinationsmöglichkeiten entstehen aber in jedem Fall weitere, und zwar teilweise signifikant unterschiedliche Vorschläge dafür, wie die Architektur gestaltet werden kann.

Damit ist es aber noch nicht getan. Eine Architektur umfasst nicht nur eine Reihe von Systemkomponenten, sondern besagt auch, wie diese miteinander in Verbindung stehen und interagieren. Auch der Aufbau der Architektur lässt sich variieren. So gelangt man möglicherweise von einer Schichtenarchitektur zu einer vertikalen Architektur, die wiederum eine weitere Alternative darstellt.

## 3. Projektplanung

Für ein neues Projekt soll eine grobe initiale Planung aufgestellt werden. Ziel ist es, eine Applikation für einen Kunden zu entwickeln. Ein Team wird zusammengestellt und aus den Anforderungen werden die

---

3. Tatsächlich sind Scrum und Kanban keine konkurrierenden Verfahren. Während es sich bei Scrum um ein Prozessframework handelt [Pichler 2008; Roock/Wolf 2015], ist Kanban eine Methode zur Visualisierung des Workflows mit dem Ziel, Engpässe zu identifizieren [Anderson 2011; Burrows 2015]. Entsprechend ist es nicht überraschend, dass eine Kombination aus Scrum und Kanban möglich ist.

ersten groben Arbeitspakete geschnürt. Eine erste Schätzung ergibt, dass mit der vorliegenden Planung der zunächst anvisierte Termin kaum zu halten ist. Es steht die Frage im Raum, ob und wie die Planung verändert werden kann.

Blicken wir zunächst auf die typischen Bestandteile einer groben Projektplanung: Team, Funktionalität, Qualität und Deadline. Es ist weitgehender Konsens, dass man bei den Qualitätsansprüchen keine Abstriche machen sollte, weswegen Variationen bezüglich der Qualitätsforderungen außen vor bleiben. Es bleiben die drei Klassiker: ein größeres Team, ein geringerer Funktionsumfang und eine spätere Deadline. Selbstverständlich kann man so zu alternativen Planungen gelangen. Es bestehen aber noch mehr Möglichkeiten der Variation. Zum Beispiel kann man nicht nur die Teamgröße, sondern auch die Zusammensetzung des Teams variieren. Lässt sich ein Team mit erfahreneren Entwicklern zusammenstellen? Besteht die Möglichkeit, auch Kundenmitarbeiter mit ins Team aufzunehmen? Auch in puncto Funktionalität und Termin lassen sich weitere Variationen finden. Ist ein inkrementelles Vorgehen denkbar, bei dem die Software schrittweise ausgeliefert wird? All dies sind Fragen, die zu Handlungsalternativen geradezu einladen.

Die Beispiele zeigen, dass sich durch die Variation bekannter Handlungsoptionen nicht automatisch neue, sinnvolle Handlungsoptionen ergeben. Sie zeigen aber auch, dass die Variation einzelner Aspekte einer Handlungsoptionen Ideen für weitere Handlungsoptionen liefern kann. Letztlich führt dies zu einer Bereicherung der eigentlichen Entscheidung, die natürlich davon lebt, dass unterschiedliche Optionen zur Verfügung stehen.

*Diskussion*

## 2.6 Entscheidungsmatrix

Wir haben eine Entscheidung so weit vorbereitet, dass wir im Rahmen einer ZIELEXPLIKATION (2.1) Kriterien festgelegt haben. DER BLICK ZURÜCK (2.3) und DER BLICK VON AUSSEN (2.4) haben sichergestellt, dass wir eine gute Vorstellung von den möglichen Handlungsoptionen entwickeln konnten und uns deren Vor- und Nachteile sowie der damit verbundenen Risiken weitgehend bewusst sind.

*Kontext*

Wie gelingt es uns, die Handlungsoptionen, ihre Vor- und Nachteile sowie die Rahmenbedingungen und Risiken übersichtlich darzustellen?

*Problem*

Eine Entscheidung besteht aus mehreren Komponenten: vor allem Kriterien, Risiken und Handlungsoptionen. Alle diese Komponenten müssen betrachtet werden, um eine gute Entscheidung zu treffen. Dies geht am besten, wenn die Informationen in übersichtlicher Form vorliegen.

*Hintergrund*