

## Woche 4

# Expeditionsplan konsolidieren

## Was ist eine Team Charter?

Stellen sie sich vor, sie sind Reinhold Messner<sup>1</sup> oder Arved Fuchs<sup>2</sup>. Sie planen mit Ihrem Team die nächste große Expedition, beispielsweise die Besteigung eines Berges auf einer unbekanntem Route. Einen solchen Plan nennen wir „Team Charter“. Darin legen Sie die gemeinsame Vision (= Erreichen des Gipfels), die daraus resultierenden Ziele und Kriterien zu deren Erreichung fest. Zusätzlich überlegen Sie sich, welche Ausrüstung und Fähigkeiten sie brauchen. Zuletzt spielen Sie die Expedition durch und überlegen, was ihnen alles zustoßen könnte.

## Ziele in der vierten Woche

Sie erstellen mit ihrem Team eine Team Charter, ggf. nach Rücksprache mit dem Auftraggeber. Ergebnis ist ein Plan, der auch die notwendigen Ressourcen zeigt und bereits erste Risiken betrachtet. Ergebnisse sind

- Produktvision
- Ressourcen-Tabelle, sowie ein gutes Verständnis der Beistellungen, die sie vom Auftraggeber benötigen
- Risikoliste, sowie ein erster Plan, wie sie mit den wichtigsten Projektrisiken umgehen wollen
- Erster Architekturentwurf, damit sie den Bau des Produkts planen können
- Technischer Durchstich, damit sie die technischen Risiken des Architekturentwurfs verstehen und möglichst senken

## 1. Produktvision formulieren

Erstellen Sie im Team eine **Vision** für das System. Diese Vision sollte in einem einzigen Satz das zentrale Ziel des Systems darstellen. Die Vision dient dem Team während des Projekts als ständige Orientierungshilfe und Identifikation. Beispiel (J.F.Kennedy am 25.05.1961):

„I believe that this nation should commit itself to achieving the goal, before the decade is out, of landing a man on the Moon and returning him safely to the earth.“

Sie können beispielsweise zum Formulieren der Vision folgenden Rahmen verwenden:

Für [Kundengruppe / Personas],  
die [Beschreibung des Bedarfs oder der Gelegenheit],  
ist das [Produktname] eine [Produktkategorie],  
die [Hauptvorteil, Grund das Produkt zu kaufen];  
( anders als [Alternative der Wettbewerber]  
kann unser Produkt [Beschreibung des Hauptunterschieds].)

Beispiel (adaptiert aus Gloger: Scrum, Hanser-Verlag):

*Für professionelle Fotografen,  
die schnelle, actionreiche Bilder machen wollen,  
ist die RO-One eine digitale Spiegelreflexkamera,  
die es erlaubt, Bilderserien mit 35 Bildern pro Sekunde mit einer bis dahin nicht erreichten Qualität zu schießen.*

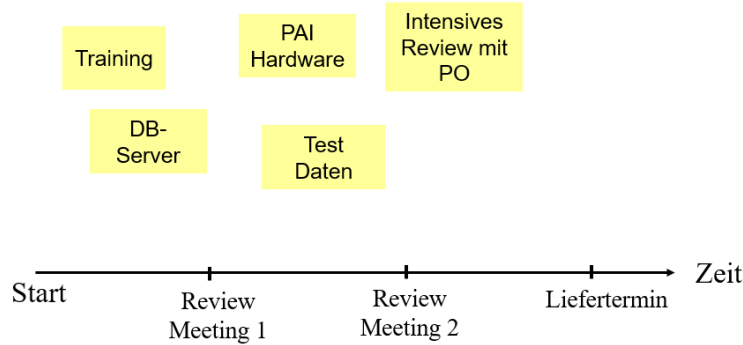
Dokumentieren Sie diese Vision auf einem Flipchart-Bogen (gerne auch grafisch).

## 2. Ressourcen-Tabelle erstellen

Führen sie im Team ein kurzes Brainstorming der Beistellungen, die sie benötigen und der anderen notwendigen Ressourcen, damit Sie ihre Projektziele erreichen können.

<sup>1</sup> <https://www.reinhold-messner.de/>

<sup>2</sup> <https://www.arved-fuchs.de/>



**Abbildung 1: Brainstorming der benötigten Beistellungen und anderer Ressourcen**

Erstellen Sie eine Tabelle der von ihnen benötigten Ressourcen mit fünf Spalten: Beschreibung der Ressource (wir brauchen) und wie kann diese beschafft werden? Was hat das Team schon? sowie Beschreibung des Bedarfs (es fehlt), spätester Termin und verantwortliche Person aus dem Team oder vom Auftraggeber.

OK	Wir brauchen ...	Wir haben ...	Es fehlt ...	Wer?	Wann?
<input type="checkbox"/>	5 <u>Okkulus Quest Brillen</u>	3	2	<u>Be</u>	
<input checked="" type="checkbox"/>	Amazon Zugang für <u>Srv.</u>				
<input checked="" type="checkbox"/>	<u>Unity</u> - Kenntnisse	x	Schulung	Magerl	30.03.
<input type="checkbox"/>	3 leistungsstarke Rechner	x	3	Kucich	04.04.
<input type="checkbox"/>	CI-Server	x	Skript	Horst	10.04.

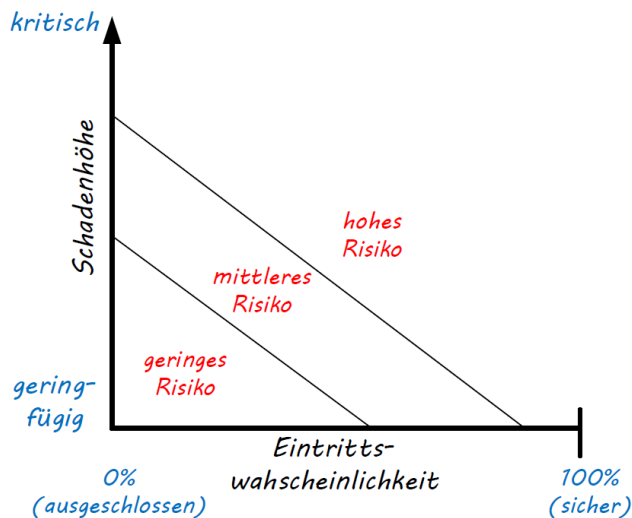
Zu den Ressourcen, die ihr Auftraggeber liefern sollte, sog. Beistellungen, gehören Ansprechpartner (inklusive Verfügbarkeit!), Lizenzen, Server-Infrastruktur, Passwörter und Dokumente. Die vierte Spalte stellt dar, wer für die Beschaffung der Ressource verantwortlich ist, ein Teammitglied oder der Auftraggeber. Die fünfte Spalte zeigt den spätesten Termin, zu dem Sie diese Ressource haben müssen.

**Besprechen sie bitte die Beistellungen mit ihrem Auftraggeber. Deren Lieferung müssen sie engmaschig überwachen. Beistellungen kommen häufig zu spät und in schlechter Qualität.**

**Überlegen Sie bitte auch, welche Fähigkeiten Sie für das Projekt brauchen**, z.B. Unity-Kenntnisse, Kotlin-Programmierung oder 3D-Modellierung. Auch diese Fähigkeiten tragen Sie bitte in die Tabelle ein. Bei fehlenden Fähigkeiten tragen Sie bitte eine Schulung oder einen Online-Kurs als benötigte Ressource mit ein.

### 3. Brainstorming der Projektrisiken

Zeichnen Sie auf einem Whiteboard ein Koordinatensystem mit Schadenshöhe auf der Y-Achse und der Eintrittswahrscheinlichkeit auf der X-Achse. Nun brainstormen Sie im Team mögliche negative Ereignisse (Risiko) die ihrem Projekt zustoßen können. Typischerweise bedrohen solche Risiken den Liefertermin, das Budget, die Qualität oder den Funktionsumfang sowie die Produktivität des Teams.



**Abbildung 2: Beispiel für ein Schema**

Denkbare **Risiken** für ihr Projekt sind:

- Technische Infrastruktur leistet nicht das, was Sie brauchen oder ist zu komplex (z.B. Reifegrad von .NET-Entity-Framework oder den dazu notwendigen Tools nicht ausreichend).
- Wichtiger Stakeholder springt ab oder ist nicht mehr verfügbar.
- Probleme in Ihrem Team (z.B., weil nicht jeder im Team gleich viel arbeitet)
- Hardware oder Software-Infrastruktur ist nicht rechtzeitig vorhanden, etwa der Raspberry Pi, ein Moodle-Server oder ein Haus zum Testen?

Wenn ihr System beim Nutzer nicht oder falsch funktioniert, entsteht in der Regel ein Schaden. Mögliche Schäden, die ihr Produkt anrichten kann, werden als **Produkttrisiken** bezeichnet. Diese reichen vom einfachen Komfortverlust bis hin zu gravierenden Schäden für Menschen und die Umwelt. Diese Risiken betrachten wir später.

**Maßnahmen überlegen:** Greifen sie im Team die hohen und eventuell die mittleren Risiken heraus und überlegen sie sich nach dem Brainstorming, wie sie diese Risiken verkleinern können. Verkleinern bedeutet, dass sie die Eintrittswahrscheinlichkeit senken oder die mögliche Schadenhöhe verringern. Eine wichtige Maßnahme ist beispielsweise der Bau des Technischen Durchstichs unten. Weitere Maßnahmen sind:

- Regelmäßige Besprechungen mit dem Auftraggeber, dort z.B. Verfolgung der Beistellungen
- Häufiges Liefern funktionsfähiger Inkremente
- Intensive Qualitätssicherung durch Testgetriebene Entwicklung, Code Reviews in Merge-Requests, eine laufende Build-Pipeline, ...

**Maßnahmen beschließen:** Erstellen sie für die geplanten Maßnahmen entsprechende Tickets in GitLab und weisen sie diese gleich den entsprechenden Teammitgliedern zu.

## 4. Erster Architekturentwurf

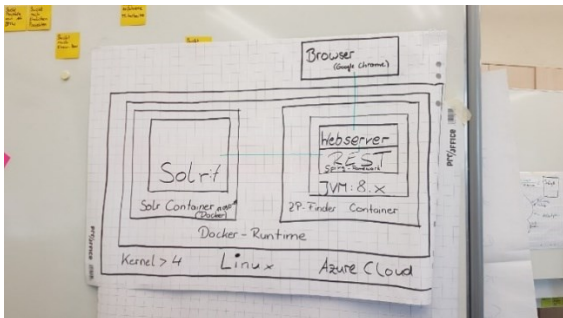
In der letzten Woche haben sie grundlegende Anforderungen an das zu bauende Produkt / System erfahren. Sie haben erste Qualitätsanforderungen und Rahmenbedingungen vom Auftraggeber erfragt. Eventuell liegen erste Skizzen der grafischen Oberfläche (wenn es eine gibt) oder ein erstes Verständnis des Datenmodells vor.

Auf dieser Grundlage können sie einen ersten Entwurf der Architektur machen, eventuell sind bereits große Teile davon durch die Rahmenbedingungen vorgegeben, da sich der Auftraggeber bestimmte Frameworks, Datenbanktechnologien oder Infrastruktur wünscht.

Um drei Themen müssen sie sich an dieser Stelle kümmern:

1. Verteilung ihres Produkts / Systems auf dem Trägersystem und damit eine erste Entscheidung der deploybaren Einheiten. Sie entscheiden sich damit für einen Architekturstil (Microservice, Monolith, Service-Basiert, Event-Driven ...)
2. Wenn sie in größerem Umfang Daten speichern, Entscheidung für eine Datenbank-Technologie (Relational, Graph-Datenbank, Document-Store, Zeitreihen-Datenbank, ....)
3. Auswahl von Frameworks für die einzelnen deploybaren Einheiten, beispielsweise entscheiden sie sich für Quarkus und Java am Backend und für React und JavaScript am Frontend.

**Verteilung:** Dokumentieren Sie die Verteilung ihrer deploybaren Einheiten auf die verschiedenen Rechner des Trägersystems über ein informelles Schaubild oder über ein **UML-Verteilungsdiagramm**. Diese Architektur wird auch Verteilungsarchitektur genannt. Sie zeigt, was auf dem Smartphone, dem Desktop, der AR-Brille oder in der Cloud läuft und welche Betriebssysteme und Netzwerkprotokolle verwendet werden. Als Dokumentation können sie ein Foto von einer Handzeichnung verwenden. Mit **PlantUML** können sie die Architektur auch als Text in ihrem Wiki beschreiben.



**Abbildung 3: Informelles Schaubild der Verteilungsarchitektur**

**Datenbank-Technologie:** Ihre Entscheidung für eine oder mehrere Datenbank-Technologien findet sich im Verteilungsdiagramm wieder, da dort auch die entsprechenden Datenbanksysteme eingezeichnet werden. Um diese Entscheidungen begründet zu treffen, analysieren sie bitte die Anforderungen, die daraus resultierenden Zugriffsmuster und das Datenmodell. Eventuell macht ihr Auftraggeber hierzu Vorgaben. Ein relationales Datenbankmanagement-System funktioniert meistens.

**Frameworks:** Entscheiden sie im Team die wichtigsten Frameworks. Halten sie sich an die Vorgaben des Auftraggebers, eventuell hat dieser schon bestimmte Frameworks im Einsatz. Wenn es keine Vorgaben gibt, verwenden sie die Frameworks, mit denen sie bereits Erfahrungen haben. Für unbekannte Frameworks analysieren sie bitte, ob diese noch gepflegt werden und hinreichend weit verbreitet sind. Ein Blick auf Stackoverflow, github oder openhub lohnt sich. Dokumentieren sie die ausgewählten Frameworks inklusive der jeweiligen Lizenz in ihrem Wiki.

## 5. Technischen Durchstich (Spike, Tracer Bullet) erstellen

Erstellen Sie –wenn nicht schon geschehen- einen technischen Durchstich mit ihrer Zieltechnologie! Ein Durchstich versucht, alle wichtigen technischen Elemente ihrer Software / Hardware so früh wie möglich zu integrieren. Dieser erste technische Durchstich dient zunächst nur dazu, dass Sie sich mit der Technologie vertraut machen und die Entwicklungsumgebung bei sich auf der Maschine installieren. Als Durchstich genügt zunächst ein besseres „HelloWorld“

- Erste zwei REST-Services mit Symphony, Flask, Express oder mit SpringBoot?
- Erster Button auf einem Android Telefon, erste Buttons in ihrer React / Angular Applikation
- Erster detektierter Mitarbeiter, der sich mit seinem Smartphone in die Nähe ihres WLAN Knotens bewegt

**Ihre Architektur muss noch nicht fertig sein, sie sollten aber die wichtigsten Frameworks entschieden haben. Notfalls bauen sie mit den aktuell heißesten Kandidaten den Durchstich. Wenn sich die Auswahl später ändert, müssen sie den Durchstich entsprechend anpassen.**

## 6. Build Pipeline erstellen / absichern

Wenn es ihre Technologie erlaubt, sorgen sie für eine laufende Build-Pipeline durch die ihr technischer Durchstich bereits durchläuft. Sie brauchen für jede deploybare Einheit eine eigene Pipeline bzw. entsprechende Jobs in einer gemeinsamen Pipeline. GitLab bietet hierzu sehr gute Unterstützung an (.gitlab-ci.yml). Für Projekte, in denen PWA, Server oder einfache Web-Clients entstehen, sollte die Pipeline kein Problem darstellen.

