



Template

Januar 2023

Einführung und Ziele

Stakeholder

Ein umfassender Überblick über die Stakeholder des Systems ist von entscheidender Bedeutung. Dies bezieht sich auf sämtliche Personen, Rollen oder Organisationen, die entweder die Architektur des Systems kennen sollten oder von dieser überzeugt werden müssen. Zu den Stakeholdern zählen auch jene, die aktiv mit der Architektur oder dem Code arbeiten, beispielsweise indem sie Schnittstellen nutzen. Ebenso gehören Personen dazu, die die Dokumentation der Architektur benötigen, um ihre eigene Arbeit effizient zu gestalten. Darüber hinaus sind Stakeholder involviert, die Entscheidungen über das System und dessen Entwicklung treffen. Die nachfolgende Analyse zeigt alle Stakeholder gebündelt in ihren Gewichtungen und Beziehungen.

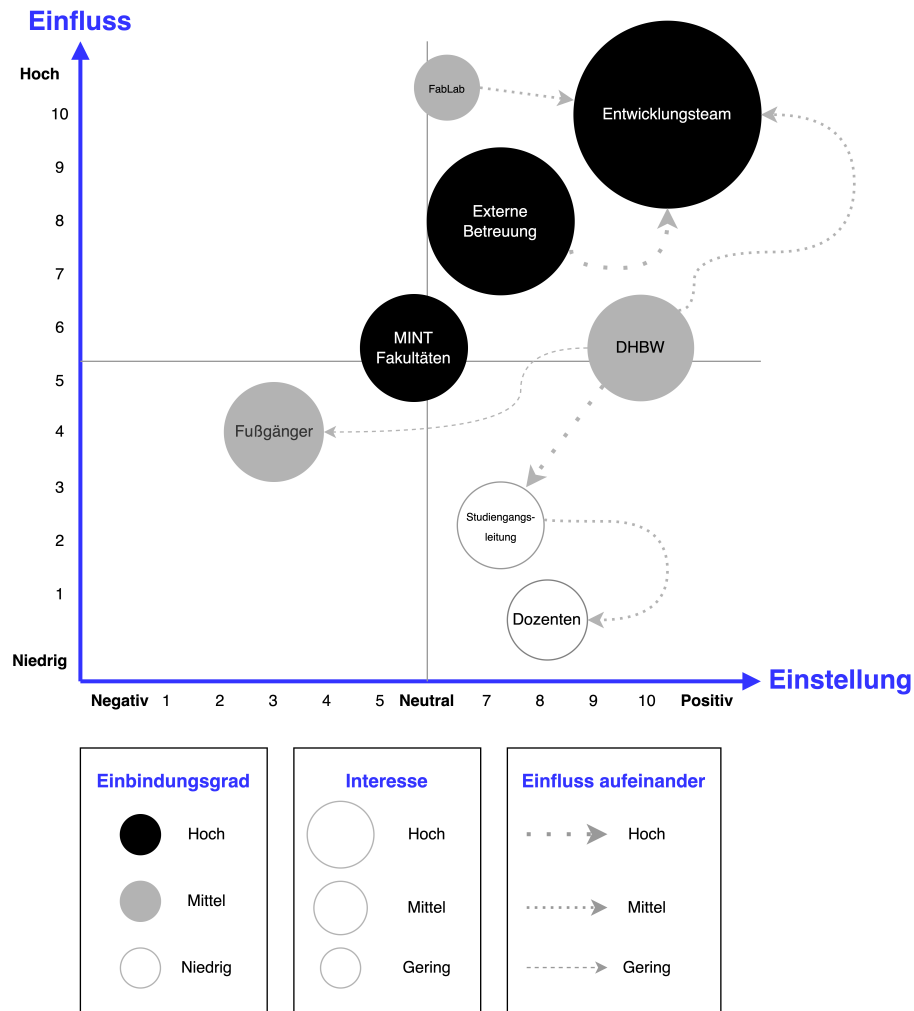


Figure 1: Stakeholder Analyse und Zusammenfassung in den Unterscheidungen: Einbindungsgrad, Interesse und Einflüsse. Das Entwicklungsteam geht deutlich als am stärksten partizipativ gekennzeichneten Stakeholder hervor. Am repräsentivsten zeigen sich die Stakeholder in Form der Fußgänger.

Anforderungskonzept

Die Anforderungen an die Umsetzung des Blickbox-Projekts werden in die Kategorien funktionale, nicht funktionale und hypothetische Anforderungen unterteilt. Diese Differenzierung ermöglicht eine umfassende Abdeckung aller Aufgabenstellungen im Zusammenhang mit der Umsetzung des Blickbox-Projekts.

Funktionale Anforderungen (FA)

Funktionale Anforderungen beschreiben spezifisch die konkreten Zwecke, die das zu entwickelnde Produkt erfüllen soll.

Nicht funktionale Anforderungen (NFA)

Im Gegensatz zu funktionalen Anforderungen sind nicht-funktionale Anforderungen eher allgemein gehalten und betreffen die gesamte Architektur und das Design des Produkts. Sie können auf verschiedene Projekte angewendet werden.

Hypothetische Anforderungen (HFA)

Hypothetische Anforderungen werden aufgrund von unsicheren Ergebnissen und vorherigen Abhängigkeiten definiert. Ihr Zweck besteht darin, mögliche Entscheidungen und Eventualitäten abzudecken, die eintreten können oder auch nicht.

<p>Datenerfassung und -übertragung (FA1)</p> <p>Beschreibung: Das IOT-System muss in der Lage sein, kontinuierlich Sensordaten zu erfassen. Diese Daten sollen im Raspberry PI gespeichert und verarbeitet werden. Die Datenübertragung auf den Server erfolgt Event-gesteuert und drahtlos. Die Daten werden nicht in Echtzeit, sondern in festgelegten Intervallen übertragen und festgehalten.</p>
<p>Bereitstellung eines Dashboard (FA2)</p> <p>Beschreibung: Die Daten der Datenbank sollen mit Grafana grafisch dargestellt werden.</p>
<p>Interaktivität zu Dritten (FA3)</p> <p>Beschreibung: Der Ausstellungscontainer soll attraktiver werden und Fußgänger sollen mit ihm interagieren können. Das soll in Form eines Displays oder Ton und Licht geschehen.</p>
<p>Präventive Lösung für Datenverlust (FA4)</p> <p>Beschreibung: Ein default-Programm auf dem Raspberry Pi sichert, eine ungestörte Funktionalität der Ausstellungsbox. Die Daten puffern wir auf dem Pi, damit die Datenbank nur zur Darstellung in Grafana existiert.</p>
<p>Bereitstellung einer geeigneten Umgebung für die Hardware (FA3)</p> <p>Beschreibung: Die Hardware soll sich im Ausstellungscontainer an einem trockenen, sicheren Platz befinden.</p>
<p>Containerisierung (FA6)</p> <p>Beschreibung: Die Anwendungen sollen in Containern gekapselt werden.</p>
<p>Erweiterbarkeit und Wartung (NF1)</p> <p>Beschreibung: Die Architektur und Dokumentation soll möglichst zugänglich gehalten werden.</p> <p>Erweiterungen und Verbesserungen der Features sollen durch die Dokumentation und einen vereinfachten Hardwareaufbau erleichtert werden.</p>
<p>Sicherheit (NF2)</p> <p>Beschreibung: Die Verbindung zum Server soll verschlüsselt sein.</p>

Entscheidungen zur Architektur

Evaluation zur Wahl des Frontend-Frameworks des Clients

Der Client zur Anzeige und Aufbereitung der Sensordaten wird unter dem Arbeitstitel *Valentin* (Komposita aus *Value* und *Notification*) geführt. Vor der

Initialisierung von *Valentin* muss ein geeignetes Framework gewählt werden. Die nachfolgende Nutzwertanalyse dient dafür als Entscheidungsgrundlage.

	Gewichtung	React		Angular		Vue	
Popularität ⁽¹⁾	2	80	160	56	112	49	98
Awareness ⁽¹⁾	3	100	300	100	300	99	297
Vorkenntnisse	5	70	350	5	25	25	125
Client Rendering Geschwindigkeit ⁽²⁾	5	67	335	68	340	67	335
<i>Summe</i>	15	317	1145	229	777	240	855

(1: <https://www.sitepoint.com/most-popular-frontend-frameworks-compared/>)

(2: <https://blog.smile.eu/en/digital/what-frontend-framework-fastest>)

Figure 2: Nutzwertanalyse zur Entscheidung eines geeigneten Frameworks gemessen an der Popularität, der Entwickler Awareness (Welches Framework enthält welchen Support), den Vorkenntnissen des Entwicklungsteams und der Rendergeschwindigkeit auf den Systemen der Clients.

Die Auswahl zum Framework innerhalb des *Valentin* Client Aufbaus viel auf *React*.