

IoT-Pflanze

Das Team

- Mohammad Ali Alnaser (@ali-alnaser1)
- Jonas Wolf (@jonaswolf)
- Mahmoud Almahdi (@Mahdi256)

Umsetzung Abgabe 1

- Gruppe finden
- Festlegung eines ersten Meetings
 - Ziel: eine Projektidee finden
 - Brainstorming
- Zweites Meeting
 - Organisatorisches/Projektablauf
 - Aufgabenverteilung usw.
 - Projektmanager auswählen (Ali)
 - GitHub Repository anlegen (Jonas)
- Drittes Meeting
 - Product-Vision-Board erstellen
 - Metriken
 - Qualitäts-Matrix
 - Risiken erkennen & bewerten
- Nach dem dritten Meeting
 - Feinarbeit an den Ergebnissen des Dritten Meetings
 - Aufbereitung
- Viertes Meeting
 - Kurze Besprechung, ob alle Aufgaben erledigt wurden
 - Pushen
 - Aufteilen der Präsentation

Umsetzung Abgabe 2

- Vor der Umsetzung

- Einholen bzw. Abwarten des Feedbacks
- Feedback im Team besprechen

- Umsetzung der Abgabe 2

- Feedback auf Abgabe 1 anwenden (Iterieren)
- Abgabe 2 vorbereiten, insbesondere:
 - Eigener Branch für Abgabe 2 (Branch: Abgabe2)
 - Ordnerstruktur installieren
 - Dateien für jede Aufgabe anlegen
 - (Dateien in PowerPoint einbinden)
 - Auch Aufgaben aus Abgabe 1 beachten!
- Vorbereitung der Powerpoint
 - Kapitelübersicht
 - Aufteilen der Präsentation
- Aufteilen der Aufgaben
- Aufarbeiten des Feedbacks
- Neue Aufgaben
- in PPT einfügen

--> Alles in die angelegten leeren Dateien

Product Vision Board

Vision: Pflanzenpflege vereinfachen und effizienter gestalten und dadurch natürliche Ressourcen schonen.

Target Group:

Haushalte
Büros
Gewächshäuser

Needs:

Vertrocknen vermeiden
Effizienz und Schonung
der Umwelt.

Product:

Intelligentes
Bewässerungssystem.
Klein, einfach zu
installieren, keine
Erfahrung benötigt und
Günstig.

Business Goal:

Eine Innovation auf den
Markt bringen.
Engagement im
Umweltschutz
unterstreichen
(gesellschaftlicher Trend)

Competetors:

u. A. Gutyán
Automatisches
Bewässerungssystem,
allerdings ohne
Datenanalyse und
Computer-Kontrolle.

Revenue Streams:

Verkauf von IoT Bundles
inklusive Soft- und
Hardware
Partnerschaften mit
bspw. Pflanzenzüchtern
(Wartungsverträge über
Soft- und Hardware).

Cost Factors:

Entwicklungskosten
(Programmierung)
Hardware (Raspberry Pi,
Sensoren, Relay usw.)
Wartungs-, Erweiterungs-
und
Fehlerbehebungskosten

Channels:

Ebay, Amazon, Messen ,
eigene Webseite,
Vertriebspartner und
Direktvertrieb.



Metriken

Business-Metriken

- **Umsatzerlöse (sales revenue):** Anzahl der verkauften Produkte, Aufgeschlüsselt nach Vertriebsplattform.
- **Kundentreue und Bindung:** Marktforschung, App-Nutzungsstatistiken und Umfragen.
- **Kosten für Kundengewinnung:** Marketings-Kosten.
- Bei unklaren Daten weitere Metriken zur Hilfe ziehen.

Preis-Metriken

- Verkauf pro Stück → Einzelverkauf und Einzelpreise (Einzelhandel)
- Wartungsverträge nach Vertragsdauer und Status des Kunden
 - Silber, Gold und Platinstatus, kostenlose Wartung und Reparatur etc.
- Software einzeln verkaufen ohne Hardware.
 - Nutzungsrechte zur Software, Hardware kann bei anderem Händler gekauft werden
- Großmengenverkauf
 - Unternehmen, Gärtnereien, Pflanzenzuchtbetriebe etc.
 - Baumärkte (Weitervertrieb)

Nutzungs-Metriken

- Statistiken zur Nutzung der App.
- Welche Features werden am meisten benutzt.
- Wie oft wird die App am Tag/Woche geöffnet
- Menge des gesparten Wassers erfassen und zur Werbung nutzen.

Operational Metriken

- Speicherplatzverbrauch der Software auf dem Arduino
 - Optimierung möglich?
- Ab wie vielen Nutzern oder verkauften Geräten muss hochskaliert werden
 - Dazu: Prüfen der Last pro Nutzer und hochrechnen
 - Prüfen der Last pro Arduino

Qualitätsstandard Metriken

- Anzahl der Fehlermeldungen der Kunden erfassen
- Wettbewerb ausschreiben
 - Wer es schafft unser Produkt zu hacken, erhält einen Preis von uns

Qualität

	Functional Stability	Performance	Compatibility	Usability	Reliability	Security	Maintainability	Portability
Functional Stability	/							
Performance	<	/						
Compatibility	^	^	/					
Usability	/	^	<	/				
Reliability	<	<	<	^	/			
Security	<	<	<	^	<	/		
Maintainability	^	<	<	^	^	^	/	
Portability	^	/	/	^	^	^	^	/

Risiken

- Konkurrenzprodukte
 - Andere Smarte Bewässerungssysteme
(Ohne Anbindung ans Internet oder App, aber ähnliche Funktionen & Zielgruppe)
 - Kann selbst gebaut werden oder durch kleine Anwendung kann Code schnell OpenSource verfügbar sein
 - Aufwendige Entwicklung im Vergleich zur Marge



Bewertung und Strategie zur Mitigation

- Vorteile einer App-Gesteuerten Anwendung gegenüber einer nicht-App-Gesteuerten Anwendung müssen dem Kunden vermittelt werden
- Code selbst ggfs. OpenSource anbieten und nur das Bundle und Support vertreiben (abwägen)

sehr hoch

hoch

mittel

gering

Sehr gering

				Bestandsgefährdendes IT-Risiko
			Sehr Hohes IT-Risiko	
		Hohes IT-Risiko		
	Geringes IT-Risiko			
Sehr Geringes IT-Risiko				

sehr
Unwahrscheinlich

eher
unwahrscheinlich

eher
wahrscheinlich

wahrscheinlich

sehr
wahrscheinlich



Requirements Engineering



Stakeholder

Intern

Was:

- Interne Prozesse
- Aufbauorganisation

Wer:

- Projektteam
- Projektleiter

Extern

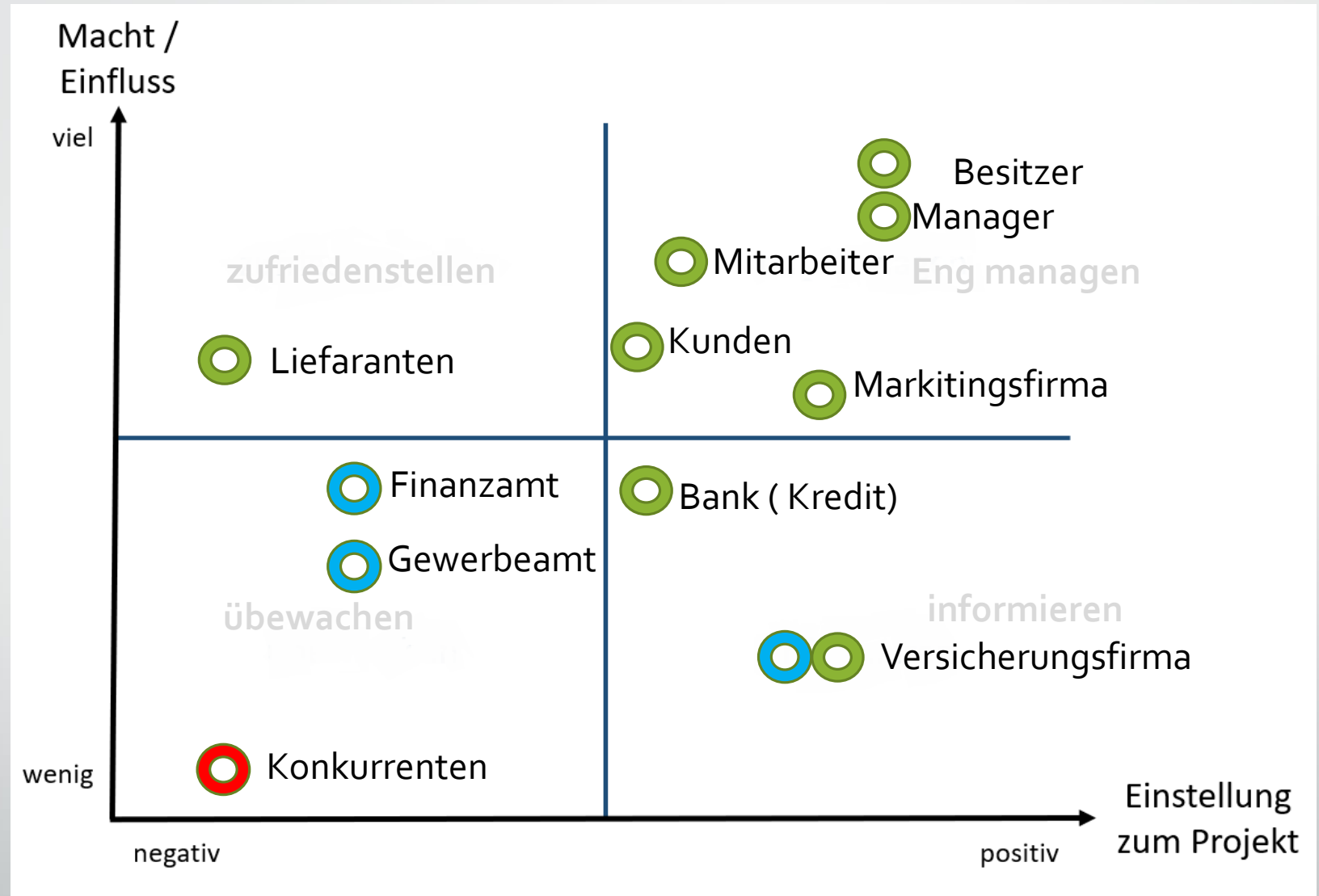
Was:

- Gesetze
- Patent
- Vertragliche Rechtlinie

Wer:

- Kundengruppen
- Behörden
- Lieferanten
- Verbände

- Neutral
- Gegner
- Unterstützer





Business Requirements

- Die App wird für Andriod-Geräte entwickelt .
- Die App soll einfach zubedienen , und erweiterbar.

Business Rules

- Bilanzierungsregeln: 40 % Eigenen Kapital – 60 % Kredit.
- Die Firma ist in OHG Form.
- Die App wird nebenberuflich in der Freizeit erstellt

Constraints

- Die APP wird mit Java entwickelt.
- Die APP läuft auf eignen Server, und besteht die Möglichkeit Offline Vision.
- Als Mikrokontroller wird Arduino benutzt.

User Classes

- **Nutzergruppen:**
 - Leute, die sich für neue Technologien interessieren und sich sehr gut mit Technik auskennen und wollen auf dem neuesten Stand der Technik bleiben (meistens Jugendliche).
 - Ältere Leute, die durch Technik ihr Leben einfacher machen wollen, die meistens unerfahren sind und sich kaum mit Technik auskennen.
 - Business Leute, die Interesse an neuen Technologien haben und sie innerhalb des Geschäftsbereichs einsetzen wollen.



Personas

Persona 1: Eva Iotstudentin

Hintergrund zur Person:

- Eva ist eine Informatik-Studentin
- Sie studiert und arbeitet nebenbei als Werkstudentin bei einem Softwarehersteller.

Demographie:

- Weiblich
- 21 Jahre alt
- Vorort von Aachen
- Wohnung, sie lebt in 4er WG

Identifikatoren:

- Eva geht für ihr Leben gern tanzen und singen.
- Sie achtet auf ihr Äußeres und geht gerne Markenklamotten shoppen.
- Eva interessiert sich für neue Technologien insbesondere IoT und möchte immer auf dem aktuellen Zustande der Technik bleiben. Sie hat z.B. Sprachassistentin bei ihr im Zimmer (Alexa von Amazon) und einen Staubsaugroboter.
- Eva interessiert sich für Gartenbau. Sie hat auch eine Terrasse, wo die einpflanzen kann.

Erwartungen, Ziele & Emotionen:

- Eva möchte total modern leben und die aktuellsten Technologien einsetzen um ihre Zeit zu sparen und cool gegenüber ihren Freunden zu wirken.
- Sie möchte sich ein System anschaffen, das das Gießen der Pflanzen automatisiert. Sie möchte das System über Internet steuern.

Herausforderungen:

- Eva ist Studentin und arbeitet nebenbei Minijob. Sie bekommt Bafög. Daher hat sie nicht so viel Geld.
- Eva hat auch anderen Mitbewohner, die in der WG wohnen, die das System auch bedienen können sollen. Insbesondere Annika, Sozialwissenschaften Studentin, die sich gar nicht mit Technik auskennt.
- Im Rahmen einer WG-Party tritt (betrunken) jemand auf das System.

Ideale Lösung:

- Wir können Eva die Sicherheit geben, dass unsere Produkte von guter Qualität sind.
- Unsere Produktentwickler haben lange am Gehäuse des Produktes gearbeitet, damit es stabil und sicher bleibt.

Häufige Einwände:

- Preis, Preis, Preis. Sie möchte nicht verhungern, weil sie so viel Geld ausgegeben hat.
- Sie möchte an ihrem Feierabend nicht 20 Fachhändler besuchen, um ihre Wahl zu treffen.



Persona 2: Familie Iotrenter

Hintergrund zur Person:

- Ursula und Wolfgang Iotrentner sind Rentner. Sie sind verheiratet seit 40 Jahren und haben 5 verheiratete Kinder und 9 Enkelkinder.
- Sie haben einen Husky-Hund
- Die Familie ist wohlhabend.

Demographie:

- Sie sind beide 82 Jahre alt
- Vorort von Frankfurt
- Haus, Wohnfläche von 180 qm, wohnen alleine

Identifikatoren:

- Familie Iotrentner fährt sehr gerne in Urlaub, meistens langfristig. Sie wollen alles nachholen, was sie wegen Zeitmangel früher nicht gemacht haben.
- Sie achten auf ihr Äußeres und kaufen immer teure Sachen.
- Die Familie interessiert sich für alles, was ihr Leben erleichtert und achten nicht auf die Kosten.
- F. Iotrentner „gärtner“ gerne.
- Durch das hohe Alter können sie nicht alleine die ganze Gartenarbeit erledigen, deswegen haben sie den Herrn Gärtnermann eingestellt, allerdings nach 2 Monaten entlassen, weil er ins Haus eingebrochen hat, während sie im Urlaub waren.

Erwartungen, Ziele & Emotionen:

- F. Iotrentner möchte sich, so weit wie möglich ihr Leben erleichtern.
- Sie möchte sich ein System anschaffen, das das Gießen der Pflanzen automatisiert.
- Da sie normalerweise länger als 1 Monat im Urlaub bleiben, möchten dass ihre Pflanzen automatisiert gegossen werden, ohne jemanden für die Aufgabe einzustellen.

Herausforderungen:

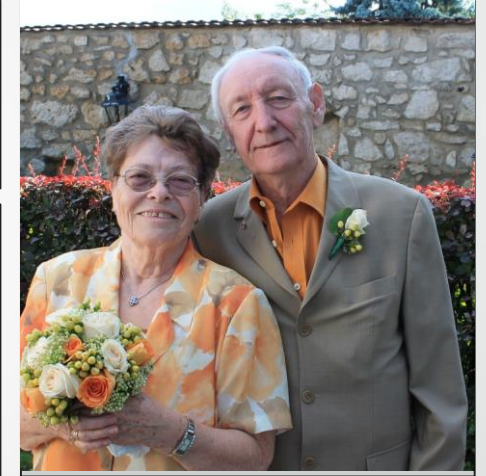
- Am Wochenende kommen die 9 Enkelkinder die Großeltern besuchen, und spielen im Garten, die Enkelkinder sind sehr neugierig und wollen alles anfassen.
- Die Familie hat einen Hund der ziemlich groß ist und auch neugierig, der liebt es sich zwischen Bäumen zu verstecken.
- Frau und Herr Iotrentner kennen sich kaum mit neuen Technologien aus.

Ideale Lösung:

- Wir können der Familie Iotrenter sichern, dass wir ein einfach bedienbares Smart-Bewässerungs-System verkaufen.
- Unsere Produktentwickler haben lange am Gehäuse des Produktes gearbeitet, damit es stabil und sicher bleibt.
- Das Produkt ist für Kinder und Hunde sicher, also kein Stromschlag ist möglich.

Häufige Einwände:

- Nutzerfreundlichkeit.



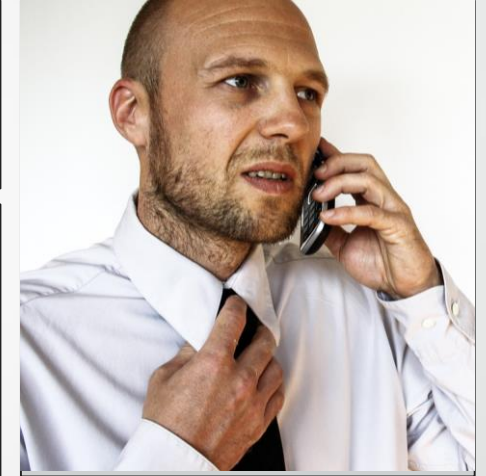
Persona 3: Max Iotmanager

Hintergrund zur Person:

- Max arbeitet bei einem großen Unternehmen.
- Er ist Manager einer großen Abteilung.

Demographie:

- Er ist 45 Jahre alt
- Vorort von Frankfurt



Identifikatoren:

- Da viele wichtige Kunden zu ihm zum Büro kommen, achtet er auf Schönheit und Dekoration des Büros.
- Max liebt das Grüne, deswegen hat er einige Pflanzen bei sich im Büro.
- Er ist meistens beschäftigt und hat viel zu tun.
- Max hat großes Interesse an IoT und Technik.
- Er hat die Zeitschrift Computer-Woche abonniert.

Erwartungen, Ziele & Emotionen:

- Max möchte sich den Zeit- und Arbeitsaufwand des Gießens der Pflanze sparen.
- Er möchte sich ein System anschaffen, das das Gießen der Pflanzen automatisiert und offline bedienen und Luftfeuchtigkeit sowie Lufttemperatur gezeigt bekommen.
- Er möchte seine Kunden begeistern und sich als innovativ zeigen.
- Da er keine Zeit zum Einkaufen hat, möchte er über Internet das Produkt kaufen.

Herausforderungen:

- Max hat viel zu tun und will sich um nichts kümmern außer seiner Arbeit.
- Das Aussehen des Büros ist ihm ganz wichtig.
- Er will nicht, dass Jemand in sein Büro in seiner Abwesenheit reingeht, um die Pflanzen zu gießen.
- Da er keine Zeit hat, möchte er nicht zum 10 Händlern gehen um das Produkt zu kaufen.

Ideale Lösung:

- Wir können Eva die Sicherheit geben, dass unsere Produkte von guter Qualität sind.
- Unser Produkt kann Max auch Online kaufen.
- Unser Produkt kann auch die Temperatur zeigen.
- Unser Wartungsteam wird den Herrn nach festgelegten Terminen besuchen um das System kostenlos zu checken, außerdem kann Max unser Support 24/7 telefonisch erreichen.

Häufige Einwände:

- Preis, weil er die Kosten durch das Unternehmen nicht erstattet bekommen wird.
- Wartungsaufwand und Lebensdauer.



Use Cases

Das System über Internet steuern

- Als Eva möchte ich das System über das Internet steuern und einsehen, ob das System an ist oder aus, da ich viel unterwegs bin.

Die Pflanzen automatisiert gießen

- Als Familie Iotrentner möchten wir die Pflanzen automatisiert gegossen werden, während wir im Urlaub sind, so dass wir niemand Fremden in unser Haus/Garten hereingehen lassen.

offline Bedienung und Lufttemperatur und Luftfeuchtigkeit zeigen

- Als Max möchte ich das System offline laufen lassen, so dass ich mich um den ganzen Prozess des Gießens nicht kümmern muss.
- Somit spare ich mir Mühe und Zeit.
- Ich möchte auch die Temperatur des Raums gezeigt bekommen.

User Story Epic

- Über das Internet stößt Karl, der Geschäftsmann ist, auf eine Werbung von unserem Produkt.
- Er ist davon begeistert geworden, das Produkt in seinem Büro anzuschaffen.
- Somit kann er seine Geschäftspartner, die zu ihm ins Büro kommen, davon auch begeistern und zeigen dass er innovative ist.
- Karl stellt das System ein, offline oder online zu steuern.
- Er kann die Lufttemperatur und Luftfeuchtigkeit im Raum einsehen.

External Requirements

- Im Grunde keine Externen Anforderungen
 - Software auf dem Server soll bereits beim Bau darauf ausgelegt sein ggfs. später auf AWS, Google, Azure oder ähnlichem zu funktionieren
 - Schnelles Skalieren ist bei einem möglichen Erfolg in der Cloud einfacher als mit eigenen Servern

Functional/Technical Requirements

- Siehe auch External Requirements
- Zur einfacheren "Produktion" soll ein Arduino verwendet werden und keine eigene Platine
 - Kosten ließen sich ggfs. durch noch geringere Entwicklung senken
 - Arduino kann mehr als Version 0 benötigt, ist aber für zukünftige Versionen sinnvoll

Non-Functional Requirement

- Siehe auch Produktqualitätsmatrix
- App sollte schnell starten
 - unter 2 Sekunden: Ideal
 - 2 - 4 Sekunden: ab und zu in Ordnung
 - über 4 Sekunden: nicht zumutbar, der Nutzer wird die App vermutlich nicht mehr starten und fast keine Funktionen/Mehrwerte in den Produkt gegenüber einer Gießkanne sehen
- Datenverarbeitung und Abfrage
 - Online/Cloud
 - Sollte 99,98 % Verfügbar sein
 - Geplante Wartungen maximal 2 mal im Jahr für 1 Stunde (idealerweise Nachts)
 - Ggfs. Verfügbarkeitsgarantie
 - Daten
 - Müssen für den Nutzer lesbar und verständlich sein (Hier sollten vorher mögliche Nutzer befragt werden)
 - Datenschutz

Releaseplanung

- Nachfolgend werden die verschiedenen Releases dargestellt, die Module sollen dabei einzeln nach Aufwand bewertet werden

Version 0

- Hardwareanforderungen
 - Arduino mit Sensoren (S)
 - Messen der Bodenfeuchtigkeit
 - Messen der Temperatur
 - ggfs. Messen der Luftfeuchtigkeit
 - Bluetooth (S)
- Softwareanforderungen
 - App: Oben beschriebene Daten in der App einsehen
 - Bluetoothverbindung (M)
 - App kann auf Daten des Arduino zugreifen und lesen (S)
 - Arduino
 - Anhand der gemessenen Daten die Wasserversorgung steuern (M)
 - Freigeben/Senden der aktuellen Daten (S)

Zusammenspiel HW und SW

Bluetooth Verbindung

Version 1

- Hardwareanforderungen
 - [...]
 - WLAN
- Softwareanforderungen
 - [...]
 - Anbindung an Datenbank (M)
 - Anzeigen von Vergangenheitswerten (S)
 - [...]
- Arduino
 - [...]
 - Schreiben der aktuellen Daten in eine Datenbank (S)
- Server
 - Datenbankstruktur (S)
 - Nutzer können auf Arduino zugreifen (nur Lesen, kein Account benötigt) (M)

Zusammenspiel HW und SW

WLAN/Internet Verbindung

Version 2

- Hardwareanforderungen
 - [...]
- Softwareanforderungen
 - [...]
 - App: Ändern der Wassermenge des Arduinos (L)
 - Muss Zwangsläufig Nutzerrechte beinhalten (M)
 - Arduino
 - [...]
 - Wasserzufuhr durch App anpassbar (M)
 - [...]
- Server
 - Datenbankstruktur → Erweitern um Nutzer (M)
 - Nutzer können auf Arduino zugreifen → Nutzersystem (M)

Zusammenspiel HW und SW

[...]

Version 3

- Hardwareanforderungen
 - [...]
- Softwareanforderungen
 - [...]
 - App: Analysefunktion durch Server bereitgestellt
- Server
 - [...]
 - Analysefunktion
 - Nutzer können Daten Analysieren und so Wasserzufuhr optimieren (M)
- Zusammenspiel HW und SW
 - [...]

Weiteres Vorgehen

- Hardware organisieren
- Coden
- Testing
- Aufsetzen