Richtlinie zum „Software-Sprint“

Colivery

Schlussbericht

Zuwendungsempfänger:

Projekt Colivery
Homberg, Ponikowski, Maia, Brück, Böck, Auricchio, Luithardt GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS20S23 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Was war Deine Motivation? Welches Problem wolltest Du mit Deinem Projekt lösen? Wie war die geplante Vorgehensweise zur Problemlösung (auch Angabe der wichtigsten Meilensteine)?

Motivation und Herausforderung: Colivery entstand aus der Motivation heraus, die Auswirkungen des COVID-19-Ausbruchs für Risikogruppenzugehörige zu verbessern. Insbesondere allein stehende Personen über 60 Jahre ohne festes soziales Netz stehen in ihrem Alltag vor außerordentlich großen Herausforderungen. Über acht Millionen Bürger:innen in Deutschland besitzen in dieser Altersgruppe keinen ausreichenden Internetzugang. Aus diesem Grunde sind sie zwangsläufig von digital zur Verfügung stehenden Hilfsangeboten sowie digitaler sozialer Teilhabe ausgeschlossen. Colivery hat das Ziel, die Versorgung mit Dingen des alltäglichen Bedarf für in dieser Weise besonders gefährdete Bevölkerungsgruppen sicherzustellen, aber auch langfristig einen Beitrag zur Lösung des generellen, gesellschaftlichen Problems zu leisten, benachteiligte Bevölkerungsgruppen durch aktive Hilfe zu unterstützen.

Unsere geplante Vorgehensweise zur Problemlösung: Colivery löst das Problem der Versorgung mit Dingen des alltäglichen Bedarfs. Mit WebApp und Ki-basierter Hotlinelösung möchten wir langfristig über die Corona-Krise hinaus für eine bessere Versorgung von benachteiligten Gruppen in der Gesellschaft beitragen. Unsere innovative Lösung ermöglicht es, Menschen mit und ohne Internetzugang mit digital organisierten Freiwilligen Helfen zu verbinden. Unser Konzept basiert auf einem bot-betriebenen automatisierten Telefon-Service, über den Hilfesuchende bequem ihre Hilfsanfrage aufgeben können, welche anschließend in einer App für Freiwillige aus der Umgebung angezeigt wird. Engagierte, hilfsbereite Menschen erfahren so per Smartphone von Hilfegesuchen aus ihrem Umfeld und können diese erledigen. So stellen wir eine essentielle Brücke zwischen der

analogen sowie der digitalen Welt her und machen zukunftsweisende Technik für viele Millionen Menschen nutzbar.

Wichtigste Meilensteine:

- Fertigstellung der KI-basierten Hotlinelösung
- Umsetzung der menschlichen Hotline (mittels Cloud-Callcenter)
- Fertigstellung der WebApp
- Implementierung von Push Notifications
- Entwicklung, Test und Integration des Algorithmus zur Routenoptimierung
- Übersetzung in mind. 10 Sprachen
- Stresstest mit 10000 gleichzeitigen Usern
- EU/internationaler Live-Gang in mind. 3 Ländern
- Online spätestens zum 15.05. (live) mit WebApp und Hotline mit ersten lokalen Gemeinde/ NGO-Kooperationen
- Rollout in mind. 10 größeren deutschen Städten zum 01.08.2020

Beitrag des Projektes zu den Zielen der Förderinitiative „Software-Sprint“

Welche Bezüge gibt es zu den Themenfeldern „Civic Tech“ und „Data Literacy“ des Software Sprints oder zu weiteren gesellschaftlich relevanten Zielen bzw. Lösungsansätzen?

Colivery dient der Zivilgesellschaft als Kommunikationsplattform mit Analog-Digital-Brücke, die eine Technik-zentrische Verbesserung des Allgemeinwohls schafft (Civic Tech). Generationenübergreifend und über die Grenzen der Internetkompetenz hinweg.

Ebenso hat Colivery eine sehr hohe Datenkompetenz - die Fähigkeit, mit Daten kompetent umzugehen. Verschiedene Einzelkompetenzen sind vorhanden, um Daten zu erfassen, anzupassen, zu verändern, zu interpretieren und zu präsentieren. Diese Grundlage ist ein wichtiger Skill der Digitalisierung und hilft der Gesellschaft eine Analog-Digital-Brücke zu schaffen.

Ausführliche Darstellung der Ergebnisse

Welche konkreten Ergebnisse hast Du erzielt? Konnten alle Meilensteine erreicht werden? Welche zusätzlichen Erkenntnisse hast Du aus der Projektarbeit gewonnen, auch im Hinblick auf die Begleitung durch die Open Knowledge Foundation?

Konkrete Ergebnisse

Kooperation mit dem Projekt „Machbarschaft“

Während der Durchführung des Projekts hat sich bereites relativ früh herausgestellt, dass die Konzepte wie auch die Herangehensweise an die vor uns liegenden Herausforderungen dem ebenfalls geförderten Projekt „Machbarschaft“ sehr gleichen. Im Interesse des Projekterfolgs haben wir uns

entschlossen unsere Kräfte zu bündeln und die Entwicklung gemeinsam voranzutreiben. In diesem Rahmen wurde das von Colivery entwickelte Backend-System sowie der Algorithmus zur Routenoptimierung in die Systemlandschaft von Machbarschaft integriert und mit der App von Machbarschaft kombiniert. Ferner wurden die Projektmanagement-Tools welche ursprünglich von Colivery eingerichtet wurden sowie die interne Kommunikationsplattform mit Machbarschaft geteilt und nach Abschluss der Projektphase auch komplett an Machbarschaft übergeben. Die Kooperation besteht fort.

Colivery WebApp als PWA

Colivery hat ferner eine eigene App als WebApp entwickelt. Diese wurde mittels HTML, CSS und TypeScript entwickelt. Durch diese Technologiewahl ist die WebApp auf allen Plattformen - Android, iOS etc. plattformübergreifend lauffähig und kann selbst ohne App Store-Installation direkt ausgeführt werden. Diese Funktionsweise verringert die technische Hürde für Hilfesuchende als auch für Helfende, da die App über einen einzigen Link im Webbrowser erreichbar ist und anschließend zum Home-Screen hinzugefügt werden kann. Das Design der App wurde in Zusammenarbeit mit UX/UI-Designern im Team überarbeitet und optimiert, sowie an das Designkonzept der Gesamtlösung angepasst. Während der Implementierung wurde viel Wert auf eine hohe Code-Qualität gelegt. Der Programmcode ist somit für zukünftige Weiterentwicklung, Forschungszwecke und Projektfremde Softwareentwickler sehr gut zugänglich und kann leicht für zukünftige, unbekannte Verwendungszwecke erschlossen werden. Diese Leistung wird auch dadurch belegt, dass im Rahmen unserer Kooperation mit dem ebenfalls geförderten Projekt „Machbarschaft“ der gesamte Programmcode für die Backend-Lösung (Datenbank/Programmlogik) übernommen wurde und in die Systemlandschaft von Machbarschaft als Kernsystem zur Datenverarbeitung integriert wurde. In der Folge wurde die App und das Admin-Dashboard von Machbarschaft an das ursprünglich von Colivery entwickelte System angebunden und wird fortan von Machbarschaft betrieben. Teile der Programmcode-Schnittstellen konnten von Machbarschaft ebenfalls aus der Colivery WebApp übernommen werden.

Backend

Der wichtigste Bestandteil, das Kernsystem, einer datenverarbeitenden Lösung sind die Datenbank sowie die gewählten/entwickelten Algorithmen zur Verarbeitung eingehender uns ausgehender Daten. Im Falle von Colivery und Machbarschaft handelte es sich um ein Verteilungsproblem das eine skalierbare, hochfunktionale Datenbanklösung in Verbindung mit einem Algorithmus, der Angebot und Nachfrage regional-selektiv miteinander in Verbindung setzt, erforderte. Ausschließlich eine Lösung, die in der Lage ist, Hilfsgesuche und Hilfsangebote regional abzugleichen („Geo-Matchmaking“) und auf intelligente Art und Weise Schlüsse daraus zu ziehen („Welche Helfende Person kann/sollte welcher hilfsbedürftigen Person in ihrer Umgebung im gewählten Zeitraum helfen?“). Im Detail ergab sich aus dieser einfach erscheinenden Fragestellung einiger Forschungs- und Entwicklungsbedarf.

Die gesamte Backend-Lösung (Datenbanklösung, Geohash-Algorithmus und Programmierschnittstelle) wurde mit der Open Source-Programmiersprache „Kolin“ entwickelt, welche auf der Grundlage der Programmiersprache Java basiert (JVM). Die Programmiersprache ist somit Java-kompatibel. Durch diese Technologiewahl ist die von Colivery entwickelte Lösung außerordentlich plattformunabhängig und kosteneffizient einsetzbar.

Innovativer, effizienter Matchmaking-Algorithmus

Im Rahmen der Suche nach dem besten Algorithmen für unseren Anwendungsfall haben wir uns für das Matchmaking letztlich für einen „Geohash"-basierten Ansatz entschieden. Auf Basis existierender Konzepte (u.A. veröffentlicht von Google Research) entwarfen wir einen Geohashing basierten Algorithmus um die Frage („Welcher Helfende ist regional für eine Hilfsanfrage verfügbar?") zu beantworten. Beim Geohashing wird der Planet als ein Ellipsoid-Modell angenommen - anschließend in Längen- und Breitengrade aufgeteilt, sodass ein Raster entsteht. Teilt man das Raster mathematisch immer weiter, entsteht dadurch, geometrisch gesehen ein Mesh („Raster") aus vielen, kleinzelligen „Kacheln". Bei jedem Teilungsvorgang (innerhalb einer „Kachel" höherer Ordnung) entsteht eine weitere Dimension. Eine Kachel niedrigster Dimension kann beispielsweise eine Flächengröße von 10qm darstellen. Eine Kachel höchster Dimension, beispielsweise die Flächengröße einer Intersection eines Längen- und Breitengrades. Ordnet man nun jeder mathematisch definierten Kachel eine eindeutige Kennzeichnung zu, entsteht ein s.g. Geohash. Die „Kacheln" sind dadurch benannt und können mit deutlich weniger Komplexität und Aufwand verarbeitet werden, als wenn mit jeder Hilfsanfrage mit GPS-Punktkoordinaten gerechnet werden müsste.

Sobald Hilfsgesuche an einer bestimmten GPS-Position eingestellt werden, muss mit dem implementierten Geohash-Algorithmus lediglich der Geohash (die Kachel) errechnet werden, in dem sich diese GPS-Position anhand ihrer Koordinaten befindet. Anschließend wird ermittelt, welche potentiell helfenden Personen sich momentan ebenfalls in ebendiesem Geohash (selbe Kachel) befinden und regional effizient helfen könnten. Kommt der Algorithmus zu einer Übereinstimmung, kann diese Antwort mit der App/WebApp geteilt und Helfende Personen direkt informiert werden.

Skalierbare Datenbanklösung

Im Anschluss an die Implementierung des Algorithmus, wurde eine Datenbanklösung entwickelt, welche die Speicherung und Abfrage aller Hilfegesuche und Hilfsangebote ermöglicht. Dabei wurde zunächst auf eine Cloud-basierte Lösung gesetzt (Google Firebase). Bei der Umsetzung der Firebase-Lösung kamen wir zu der Erkenntnis, dass Google Firebase jedoch keine Möglichkeiten zur Volltextsuche bietet. Diese Limitation machte das effiziente Suchen in den Freitext-Angaben der Hilfsgesuche nur umständlich möglich, weswegen wir in der Folge eine alternative Datenbanktechnologie, die Open Source-Datenbanklösung PostgreSQL wählten. Für den Betrieb der Lösungen haben wir uns ebenfalls für ein Hosting innerhalb der Google Cloud entschieden.

Programmierschnittstelle (API)

Im Rahmen der Entwicklung einer zugänglichen Systemlösung haben wir uns entschieden, eine Programmierschnittstelle zu implementieren, die eine mittels OpenAPI-Spezifikation dokumentiert ist. Auf Basis dieser Dokumentation können automatisch Adapter generiert werden, welche in der Lage sind, die Programmierschnittstelle anzusprechen. Diese Lösung entspricht modernsten Standards in der Softwareentwicklung und ermöglicht eine einfache Anbindung des Kernsystems mit s.g. Clients - beispielsweise der Colivery WebApp oder auch der Machbarschaft App. Folgende Schnittstellen wurden implementiert:

- **Benutzerkonto erstellen**
    - Hilfesuchende Person
        - Vorname
        - Nachname
        - Straße
        - Hausnummer
        - Postleitzahl
        - Ort
        - E-Mail
        - Geohash (errechnet durch Algorithmus)
        - Telefonnummer
        - Quelle der Registrierung (z.B. Registrierung in Machbarschaft App oder Colivery WebApp oder durch Support-Hilfsraft im Machbarschaft Admin Panel)
    - Helfende Person
        - Analog zu „Hilfesuchende Person“
- **Benutzerkonto aktualisieren**
    - Personenbezogene Daten anpassen (s.o.)
- **Benutzerkonto aktualisieren (partiell)**
    - Passwort ändern
- **Benutzerkonto: Daten abrufen**
    - Ausschließlich möglich für autorisiertes Supportpersonal oder den Eigentümer des Benutzerkontos oder nach Einwilligung des Eigentümers auch für die helfende Person
- **Hilfsangebot aufgeben**
    - Name
    - Telefonnummer
    - Quelle des Eintrags (s.o.)
- **Hilfsangebot aktualisieren**
    - Name
    - Telefonnummer

- **Hilfsgesuch aufgeben**
    - Freitext des Hilfsgesuchs (eigegeben in App, Admin-Dashboard (Supportpersonal) oder aufgenommen von KI-Hotline)
    - Status des Hilfsgesuchs (offen, in Bearbeitung, abgeschlossen)
    - Referenz zu helfender Person
- **Hilfsgesuch aktualisieren**
    - Freitext aktualisieren
    - Status aktualisieren
- **Nach Hilfsgesuchen suchen**
    - Regionale Suchabfrage mit Radiusangabe in km
    - Gibt Liste an Hilfsgesuchen zurück, sodass eine helfende Person in der Colivery WebApp / Machbarschaft App darauf reagieren kann
- **Hilfe anbieten**
    - Helfende Person bietet Hilfe an
- **Hilfsangebot akzeptieren**
    - Hilfe suchende Person wird informiert und akzeptiert das Hilfsangebot (oder nicht)

Der Programmcode zur Programmierschnittstelle, der von Colivery implementiert wurde, wurde im Rahmen der Kooperation mit Machbarschaft an Machbarschaft übertragen.

KI-basierte Hotline und menschliche Hotline

Im Rahmen der Durchführung des Projekts wurden mehrere Wege geprüft, wie eine KI-basierte Hotline und menschliche Hotline effizient und kostengünstig umgesetzt werden kann. In diesem Rahmen wurden Skripte in der Programmsprache JavaScript entwickelt die eine automatische Transformation der Sprachaufnahmen eines intelligenten, geführten Telefon-Dialogsystems in das Datenmodell des Backends ermöglichen.

Ferner wurde eine automatische API-Integration mit dem Backend-System entwickelt. Das vollautomatische Telefonsystem wurde bis in den Teststand entwickelt und optimiert. Die entstandene Lösung basiert auf Voximplant mit Integration von Google Dialogflow. Das integrierte Cloud-Telefonsystem ermöglicht ebenfalls das Aufschalten von menschlichen Fallcenter-Mitarbeiter:innen, die für Hilfesuchende manuell Hilfsgesuche mittels der WebApp einstellen können.

Nach technisch-organisatorischem Vergleich der Lösungen von Colivery und Machbarschaft haben wir uns entschieden, unsere Forschungsergebnisse in die Gesamtlösung von Machbarschaft einfließen zu lassen und mit der bereits weiterentwickelten und optimierten Telefonie-Lösung von Machbarschaft fortzufahren.

Testen

Im Rahmen umfangreicher Anwendertests, welche von den Entwicklern intern durchgeführt wurden, konnten wir eine Gesamtstabilität der von uns entwickelten Systeme erreichen. Ferner haben wir dedizierte Testsysteme bereitgestellt, auf denen wir größere Gruppen (> 100 Teilnehmer in echten Anwendertests) eingeladen haben, unsere Systeme (Colivery WebApp, Datenbank, Algorithmus und Programmierschnittstelle) zu testen. Das von uns erfasste Feedback wurde priorisiert und floss anschließend wieder in den Entwicklungsprozess ein. Darüber hinaus wurde das UX/UI-Design hinsichtlich im Test erkannter Unzulänglichkeiten überarbeitet und optimiert.

Im Laufe der Entwicklung haben wir unsere Systeme fortlaufend weiterentwickelt und getestet, sodass sie eine Qualitätsstufe erreichen konnten, die für einen produktiven Einsatz genügte.

Im Rahmen der Tests mit Anwender*innen haben wir festgestellt, dass unser ursprünglicher Ansatz, jede Bedarfsposition (Artikel zum Einkauf) einzeln zu Erfassen letztlich zu großer Komplexität für die Hilfesuchenden führt und die Hürde zur tatsächlichen Verwendung der WebApp deutlich erhöht. Ferner stellten wir in Systemtests fest, dass es, insbesondere unter dem Einfluss von Dialektsprache äußerst schwierig war, das KI-basierten Hotlinesystem insofern zu stabilisieren, dass es die tatsächlichen Hilfsgesuche detailgetreu erfassen würde. Mit der Entscheidung, die Flexibilität des Systems zu erhöhen, indem wir lediglich einen Freitext als Hilfsgesuch verwenden, konnten beide Problembereiche zufriedenstellend gelöst werden.

Der kooperative Austausch mit dem Projekt Machbarschaft hat auch in diesem Bereich zu fruchtbringenden Erkenntnissen geführt - beispielsweise bezüglich des Aspekts der „menschlichen“ Hotline (nicht KI-basiert) und den vom Projekt Machbarschaft gewonnenen Testergebnissen mit dieser Lösung.

Design: Wie Strategisches Design zum Software Sprit betragen hat.

Für uns bedeutete Strategisches Design zunächst einmal, zu hinterfragen, wie, d.h. mit welchen Prozessen, Komponenten und Herangehensweisen wir eine Komplettlösung erarbeiten können, die für den Endanwender einerseits einfach zu bedienen, gleichsam jedoch auch funktional sein kann.

In diesem Spannungsfeld identifizierten wir zwei wichtige Arbeitsbereiche für die Softwareentwicklung:

- Frontend
    - WebApp-Entwicklung mittels HTML, CSS, TypeScript mit Anbindung an die Programmierschnittstelle (Backend)
- Backend
    - Datenbank, Geohash-Algorithmus und Programmierschnittstelle
- Hotline KI-Lösung
    - Voximplant / Google DialogFlow mit Anbindung an die Programmierschnittstelle (Backend)

Innerhalb der Arbeitsbereiche wurden unterschiedliche, und auf die jeweiligen Arbeitsfelder zugeschnittene Prozesse erarbeitet.

Frontend

Bei der Entwicklung der Colivery WebApp lag der Fokus des strategischen Designs maßgeblich bei der Frage, wie wir eine WebApp für die Endanwender entwicklen und designen können, welche es möglichst einfach macht, Hilfe anzubieten. Unsere Grundannahme war es, dass sich vor allem Menschen der jüngeren Altersgruppen über die App dazu bereit erklären würden, Hilfe zu leisten, währenddessen Hilfe suchende Personen eher telefonisch an die KI-Hotline wenden würden. Aus diesem Grund haben wir gemeinsam mit den UX/UI-Designern im Team eine moderne Anwendungsoberfläche designed und implementiert, die den heutigen Ansprüchen an eine gute Usability gerecht werden würde.

Ein besonderes Augenmerk lag in diesem Bereich auf die initiale Registrierung des Benutzerkontos, da bei diesem Schritt Anwender schnell ein Gefühl der Überforderung aufkommen kann. Wir haben daher mit Weitblick eine schnelle und einfache Registrierung umgesetzt. Direkt nach der Registrierung gelangt der Anwender in eine Ansicht der offnen Hilfsgesuche und kann mit nur zwei Klicks bzw. Taps ein Hilfsangebot abgeben. Die Einfachheit dieser Lösung stellte einen besonders positiven Punkt im Nutzerfeedback dar.

Ein weiterer wichtiger Aspekt des Designs war die strategisch gewählte Farbgebung, Schriften und Kontraste, die unter anderem sicherstellen sollten, dass die Nutzung der WebApp auch für Menschen mit körperlichen Einschränkungen keine überdurchschnittliche Herausforderung darstellen würde.

Backend

Hinsichtlich des Backends wurde im Team vor allem über den strategischen Einsatz der Lösung als zentrales Kernsystem beraten. Ein Ergebnis dieses Beratungsprozesses war der ausschließliche Einsatz von Open Source-Technologien bei der Entwicklung unserer Systemlösung. Ferner wurde großer Wert auf eine für Softwareentwickler leicht zugängliche und schlüssige Programmcode-Struktur gelegt.

Das Design der Programmierschnittstelle erfolgte unter Einhaltung aller OpenAPI-Standards und ermöglichte ferner die reibungslose Anbindung des Systems als Kernsystem in Systemlandschaften, die aus mehreren Anwendungen bestehen. Der strategische Schwerpunkt lag hierbei vor allem auf dem Aspekt der Wiederverwendung. So konnte das Backend in der Systemlandschaft bei Colivery eingesetzt werden - jedoch auch zur Weiterentwicklung in anderen Systemlandschaften, wie der von Machbarschaft eingesetzt werden. Dies unterstreicht den strategischen Ansatz der Entwicklung und die Flexibilität der von Colivery entwickelten Lösung.

Über die genannten Punkte hinaus gewährleistet die gewählte Architektur der Programmierschnittstelle auch eine Zukunftsorientierung: Änderungen an der Schnittstelle und den Datenmodellen werden durch den OpenAPI-Standard automatisch reflektiert, sind offensichtlich und können von Entwicklerwerkzeugen zur (Re-)Generierung von Adapter-Programmcode verwendet werden. Die Vorteile dieser Architektur und die strategische Ausrichtung der Systemarchitektur in diese Richtung hatten maßgeblichen Einfluss auf den funktionalen Erfolg des Projekts.

Erreichbarkeit der Meilensteine:

- Fertigstellung der KI-basierten Hotlinelösung: unter Einsatz von Google DialogFlow erreicht; jedoch invalidiert und zu Gunsten der Lösung von Machbarschaft verworfen
- Umsetzung der menschlichen Hotline (mit Cloud-Callcenter): unter Einsatz von Voximplant *erreicht*; jedoch ebenfalls verworfen zu Gunsten der Lösung von Machbarschaft
- Fertigstellung der WebApp: *erreicht*
- Implementierung von Push Notifications: *nicht erreicht*; die Entwicklung wurde im Zeitraum gestoppt um die verfügbaren Entwicklungsressourcen auf unsere Backendlösung zu konzentrieren
- Entwicklung, Test und Integration des Algorithmus zur Routenoptimierung: *teilweise erreicht*; der Geohash-Algorithmus erlaubt bereits die Eingrenzung auf regionales Matchmaking. Diese Lösung ist hinreichend für die Kernanforderung. Eine zusätzliche Routenoptimierung wurde evaluiert, jedoch haben wir uns entschieden, die zur Verfügung stehenden Entwicklungsressourcen zunächst auf die Gesamtfunktionalität und Stabilität des Backends zu richten.
- Übersetzung in mind. 10 Sprachen: *teilweise erreicht*; die Übersetzung bezieht sich auf 6 Sprachen: Deutsch, Englisch, Hebräisch, Indisch, Norwegisch, Koreanisch
- Stresstest mit 10000 gleichzeitigen Usern: *teilweise erreicht*; die WebApp wurde einem Test mit mehreren hundert Usern in München unterzogen. Backend (Datenbank, Programmierschnittstelle und Algorithmus) wurden ebenfalls mit mehreren hundert Usern gleichzeitig getestet. Die Lösung ist jedoch theoretisch linear skalierbar und sollte mit entsprechendem Cloud-Budget erfahrungsgemäß problemlos im Bereich 10 - 20000 gleichzeitige User skalierbar sein.
- EU/internationaler Live-Gang in mind. 3 Ländern: *teilweise erreicht*; bis zur Phase des Live-Gangs haben wir bereits mit Machbarschaft kooperiert. Der Live-Gang sollte demnach unter dem Namen Machbarschaft stattfinden um Kräfte und Kanäle zu bündeln.
- Online spätestens zum 15.05. (live) mit WebApp und Hotline mit ersten lokalen Gemeinde/ NGO-Kooperationen: *nicht erreicht*; einerseits haben wir den Entwicklungs- und Testaufwand unterschätzt - andererseits zeigte es sich, dass eine Kooperation mit Gemeinden/NGO’s deutlich mehr koordinativen Aufwand erfordert, als ursprünglich gedacht. Wir haben unter anderen mit einer Studentenorganisation zur Nachbarschaftshilfe aus Münster gesprochen, sind letztlich jedoch durch die Kooperation mit Machbarschaft zunächst in eine Testphase in Passau eingetreten (Colivery Backend).

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Welcher Nutzen ergibt sich für die Zielgruppe aus den Ergebnissen Deines Projekts? Welche weitergehenden Effekte ergeben sich aus der Open-Source-Stellung der Ergebnisse? Gibt es Ideen für die Weiterentwicklung Deiner Lösung und Pläne zu deren Umsetzung?

Hat die Arbeit in dem Projekt Dich in Deiner persönlichen, fachlichen Weiterentwicklung unterstützt?

Colivery hat innerhalb des Projektzeitraums eine komplett funktionsfähige Lösung für die angestrebte Idee implementiert, die darüber hinaus auch flexibel und offen genug designed war, um im Projekt Machbarschaft die zentrale Rolle als Kernsystem im Backend (Datenbank, Programmierschnittstelle, Algorithmus) zu spielen.

Auf persönlicher Ebene konnten alle Beteiligten im Team deutlich von der Teilnahme profitieren. Einer internen Umfrage zufolge wurde insbesondere die folgenden Kriterien besonders positiv aufgenommen:

- Einarbeitung in neue Technologien (Kotlin, TypeScript)
- Einarbeitung in neue Cloud-Service Technologien (Google Cloud, Firebase, Google DialogFlow, Voximplant)
- Sammeln von Erfahrungen durch die Arbeit im Open Source Team, insbesondere durch den 100% Remote-Anteil (Online-Kollaboration)
- Sammeln von Erfahrungen mit agilen Arbeitsprozessen. Die von Colivery erarbeiteten Arbeitsprozesse und eingerichteten Tools (Slack, Atlassian Jira, Atlassian Confluence) wurden im Anschluss auch 1:1 von Machbarschaft übernommen
- Die Stimmung im Team

Effekte der Open-Source-Stellung

Innerhalb des Software Sprint konnten wir als Team wachsen und durch die Kooperation mit Machbarschaft schließlich auch die Potentiale von Open Source-Arbeit vollständig realisieren: Ein großer Teil der von Colivery im Software Sprint entwickelten Software lebt im Projekt Machbarschaft fort.

Ideen für die Weiterentwicklung

Das Projekt Colivery überlässt die Weiterentwicklung dem Projekt Machbarschaft - durch die Fusion der Teams im Anschluss an den Software Sprint sind Teammitglieder von Colivery dem Projekt Machbarschaft beigetreten und entwickeln die ursprüngliche Idee in diesem neuen, breiteren organisatorischen Rahmen weiter.

Nutzen für die Zielgruppe

Da die von Colivery entwickelte Software im Rahmen des Projekts Machbarschaft weiterlebt, erschließt sich der Nutzen exakt aus dem Projekterfolg des Projekts Machbarschaft.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Gab es Arbeiten bzw. Lösungsansätze, die nicht weiter verfolgt wurden? Was waren die Hintergründe, und wie bist Du alternativ vorgegangen?

Entwicklung der WebApp

Die vom Projekt Colivery entwickelte WebApp funktionierte sehr gut im Rahmen der Colivery Systemlandschaft, wurde im Zuge der Kooperation mit Machbarschaft jedoch nicht übernommen. Aus diesem Grund befindet sich der Programmcode der WebApp nicht im direkten Einsatz.

Entwicklung der KI-Hotline-Lösung

Im Rahmen des Software Sprints wurde von uns eine auf Google DialogFlow und Voximplant basierende KI-Hotline-Lösung entwickelt. Diese zeigte zuletzt auch eine Funktionsfähigkeit, hatte jedoch in Anwendertests zu dem Effekt geführt, dass Hilfesuchende Personen nur schwer Vertrauen in den Service aufbauen konnten.

Letztlich haben wir uns im Rahmen der Kooperation mit dem Projekt Machbarschaft für den Einsatz der von Machbarschaft entwickelten Hotline-Lösung entschieden, sodass die von Colivery entwickelte Lösung nicht mehr im Einsatz ist.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Wo können sich Interessenten detailliert über Deine Projektergebnisse informieren (z.B. Webseite, GitHub, Veröffentlichungen)?

GitHub-Profil und dortige Repositories:
https://github.com/Colivery

Von Colivery entwickelte Backend-Lösung:
https://github.com/machbarschaft/service-api-ng
(nach Kooperation mit Machbarschaft, zu deren Repository übertragen)

Webapp: Im Programmcode vorliegend:
https://github.com/Colivery/pwa

KI-Hotline-Lösung (Voximplant/Google DialogFlow): Im Programmcode vorliegend:
https://github.com/Colivery/callcenter-voximplant-integration/tree/master/application/simplecallforwarding.colivery.n4.voximplant.com/scenarios

Demo des Backends mit Datenbank, Programmierschnittstelle und Algorithmus:
https://machbarschaft.jetzt

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Gab es im Projektverlauf Ereignisse, die eine Anpassung der Planung erforderlich machten – z.B. Mehr- oder Minderaufwand bei der Bearbeitung von Teilaufgaben?

Während der Umsetzung des Projekts kam es zu kaum vorhersehbaren Planungsänderungen bei den Mitgliedern des Projekts. Dies führte einerseits dazu, dass mehr Mitglieder der GbR beitraten, als letztlich substantiell zum Projekt beitrugen - andererseits führte es zu Ausgleichsreaktionen, sodass

andere Projektmitglieder Mehrarbeit leisteten. Wiederum andere Projektmitglieder zeigten kein Interesse an einer Erfassung im Rahmen des Software Sprint, leisteten jedoch substantielle Arbeit im Rahmen des Gesamtprojekts.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Gab es Entwicklungen anderer Personen oder Institutionen, die Einfluss auf Deine Arbeiten und die Zielsetzung hatten? Wenn ja, worin bestand dieser und wie bist Du damit umgegangen?

Im Rahmen des Engagements des Prototype Funds war es dem Team Colivery möglich, Kontakt zu verschiedenen anderen Projekten im selben Themenfeld herzustellen. In diesem Rahmen stießen einige neue Teammitglieder zum Team Colivery hinzu.

Im Anschluss an den Software Sprint fusionierte das Team Colivery vollständig mit dem Projekt Machbarschaft sodass das Projekt Colivery als eigenständiges Projekt nicht mehr fortbestand. An der ursprünglichen Zielsetzung des Projektes änderte sich hierbei jedoch grundsätzlich nichts.