

# Endpräsentation

Team Blau

Endpräsentation

**Team Blau**

# Video

Endpräsentation  
**Team Blau**



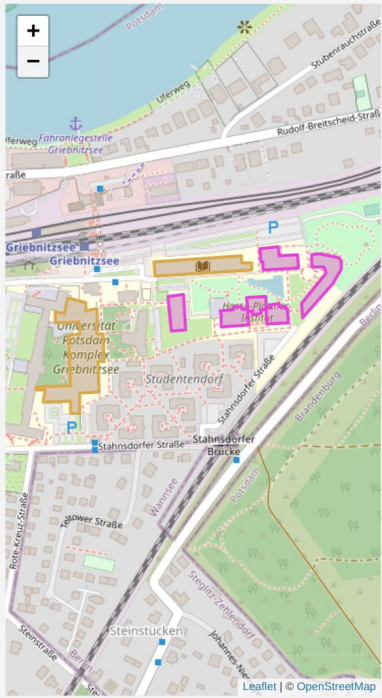
<https://compass-blue.herokuapp.com>

Endpräsentation  
**Team Blau**

# Demo

Endpräsentation  
**Team Blau**

# Roadmap - bis zur Zwischenpräsentation




## Search Results

### Search Result 0

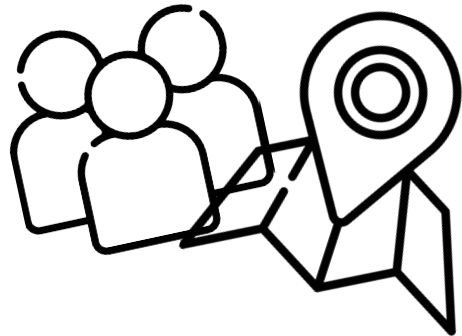
This is a dummy-description meant to be removed. It shows that this text has to be very short. [ remove me at 'app/views/search\_results/index.html.erb' ]

### Search Result 1

This is a dummy-description meant to be removed. It shows that this text has to be very short. [ remove me at 'app/views/search\_results/index.html.erb' ]

### Search Result 2

This is a dummy-description meant to be removed. It shows that this text has to be very short. [ remove me at 'app/views/search\_results/index.html.erb' ]



User, Buildings, Rooms, ...

Karte

Suchleiste

Datenmodell


Endpräsentation  
Team Blau

# Roadmap - bis zur Zwischenpräsentation

Edit User

User Details

Profile picture



Username (your displayed name)

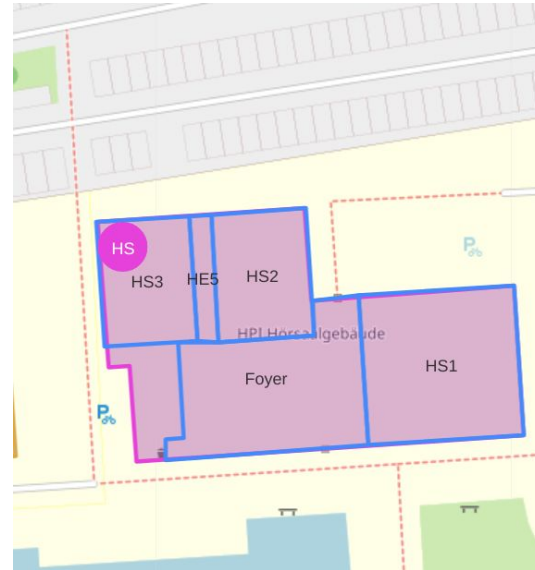
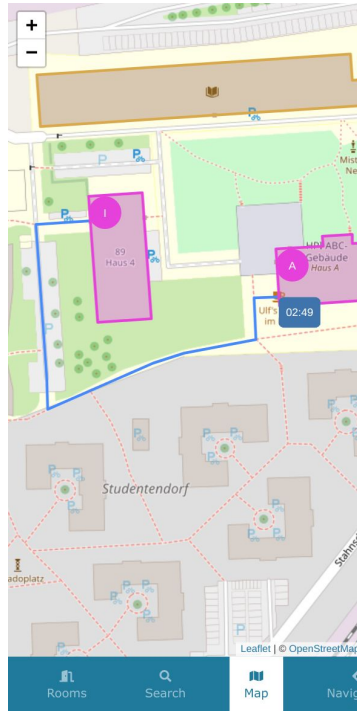
First name

Last name

Email

Phone number

Rooms



Editierbares Profil

Navigation

Indoor Maps

Endpräsentation  
**Team Blau**

# Roadmap

## New Person

Profile picture



First name

Last name

Phone number

Email

Rooms

## New Location

Name

Details

Location photo

File auswählen Keine ausgewählt

Location latitude

Location longitude

Opening times **+** Add opening time

Create Location

[Back](#)



Editoren für Anlegen von  
Personen und Locations

Layout Überarbeitung

Endpräsentation  
Team Blau


## Ausblick der Zwischenpräsentation

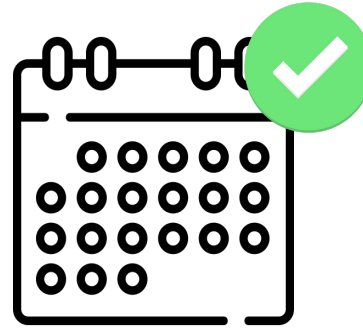


Rechtesystem



Kartendaten

- Indoor Maps Campus I
- POIs
- Live Position Tracking und Sharing 



Raumkalender

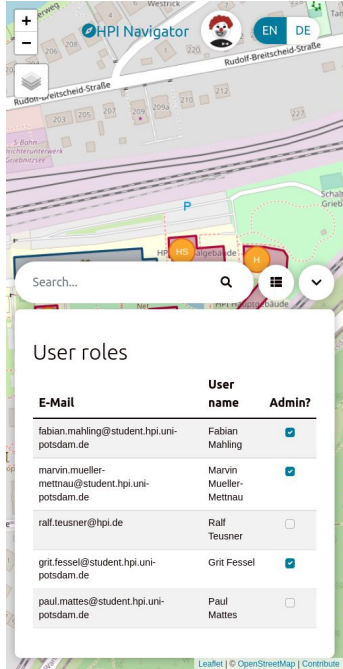


UX-Verbesserungen

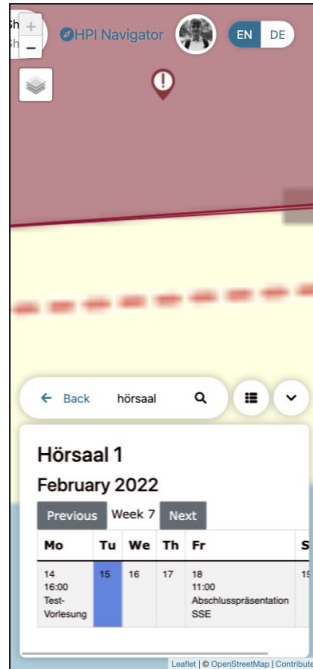
- neue UI
- i18n



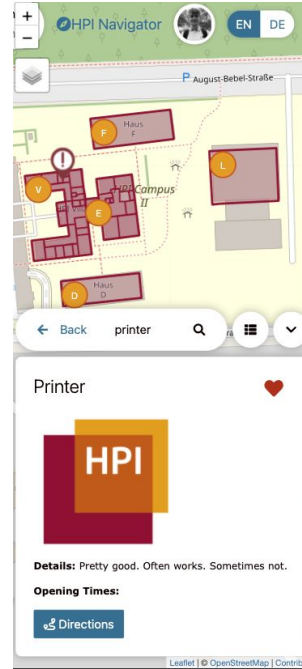
# Roadmap



Rechtssystem



Raumkalender



neue UI & Locations

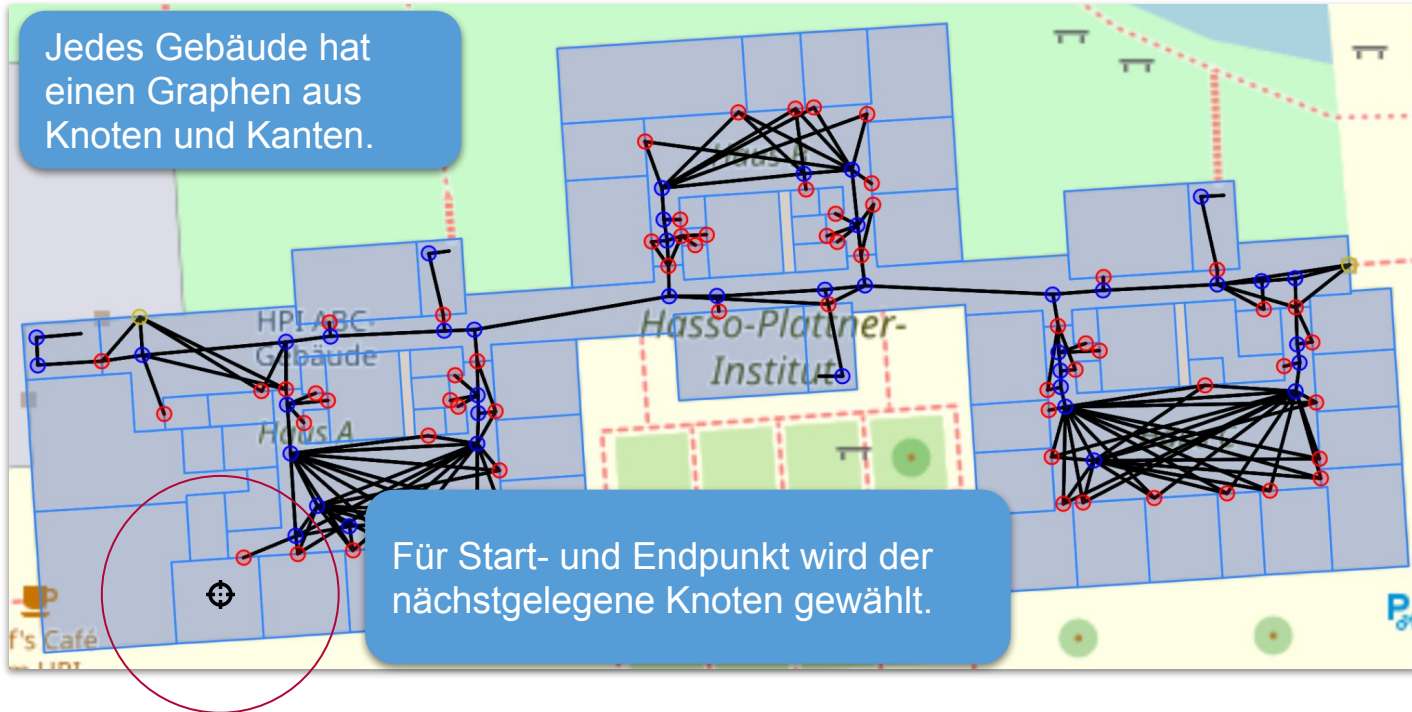


Live Position, Indoor Navigation & Raumpläne

Endpräsentation  
**Team Blau**

# Indoor Navigation - Wie berechnen wir Routen?

Innerhalb von Gebäuden:

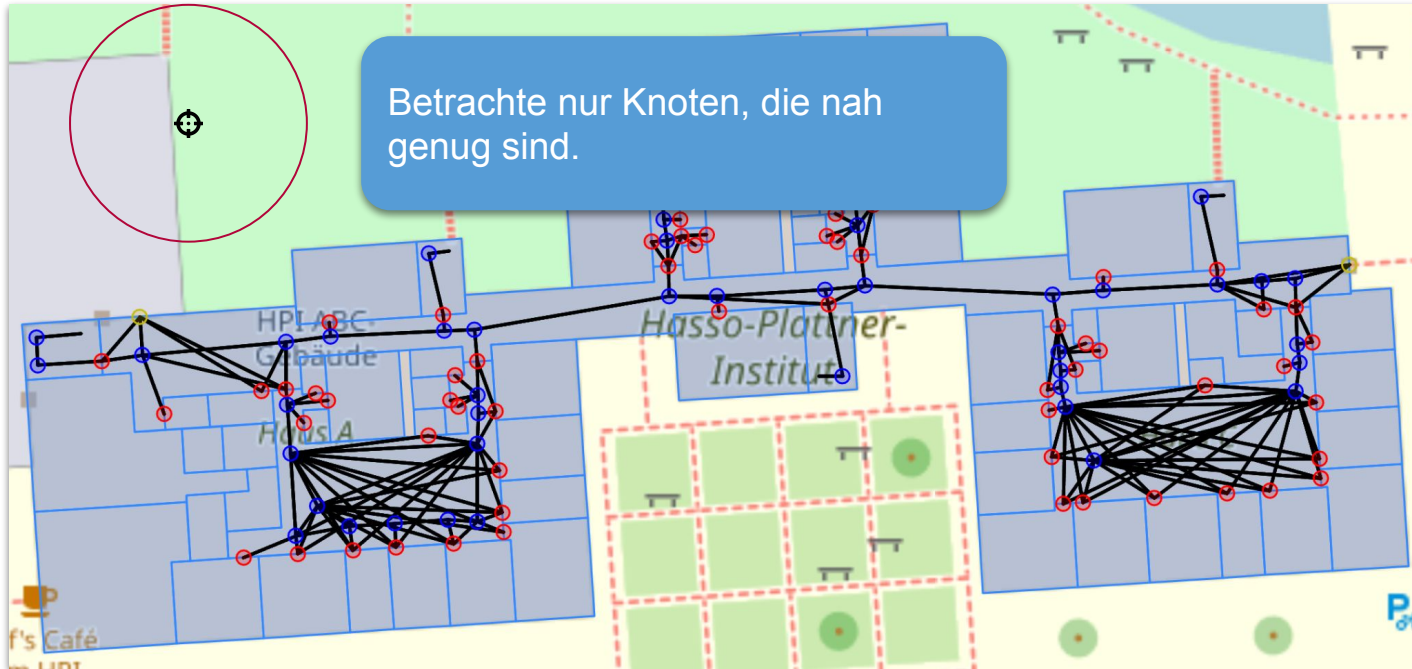


Endpräsentation

Team Blau

# Indoor Navigation - Wie berechnen wir Routen?

Außerhalb von Gebäuden:



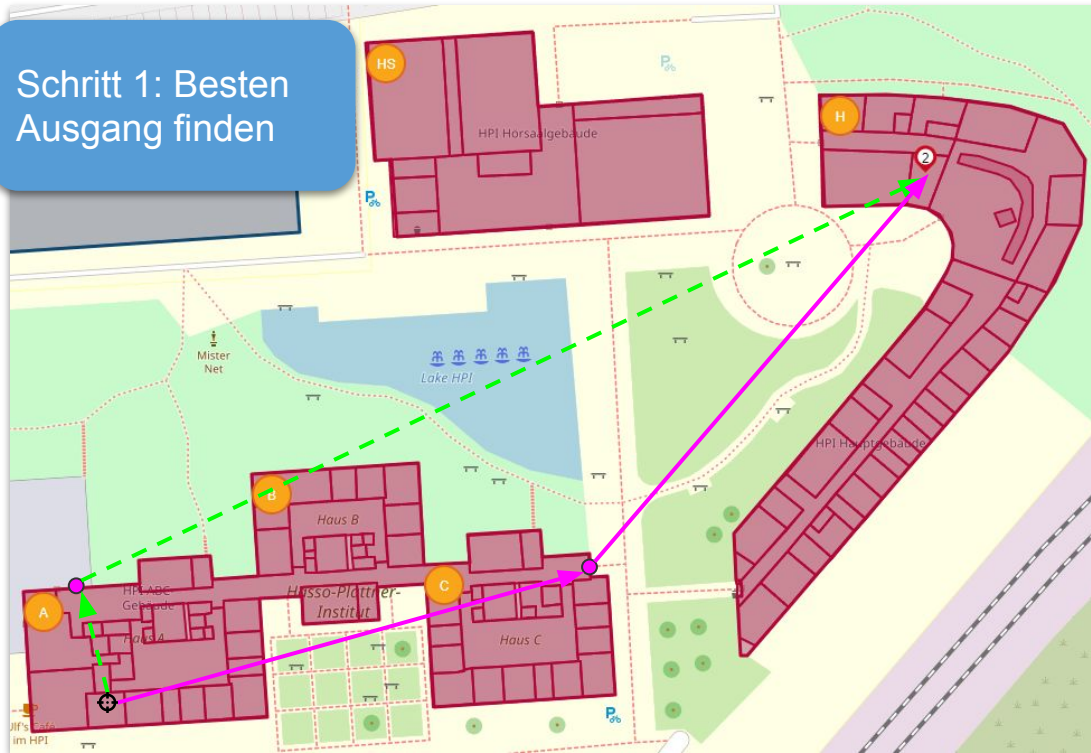
Endpräsentation

Team Blau

# Indoor Navigation - Wie berechnen wir Routen?

Beispiel: A-E.6 zu H-2.58

Schritt 1: Besten Ausgang finden



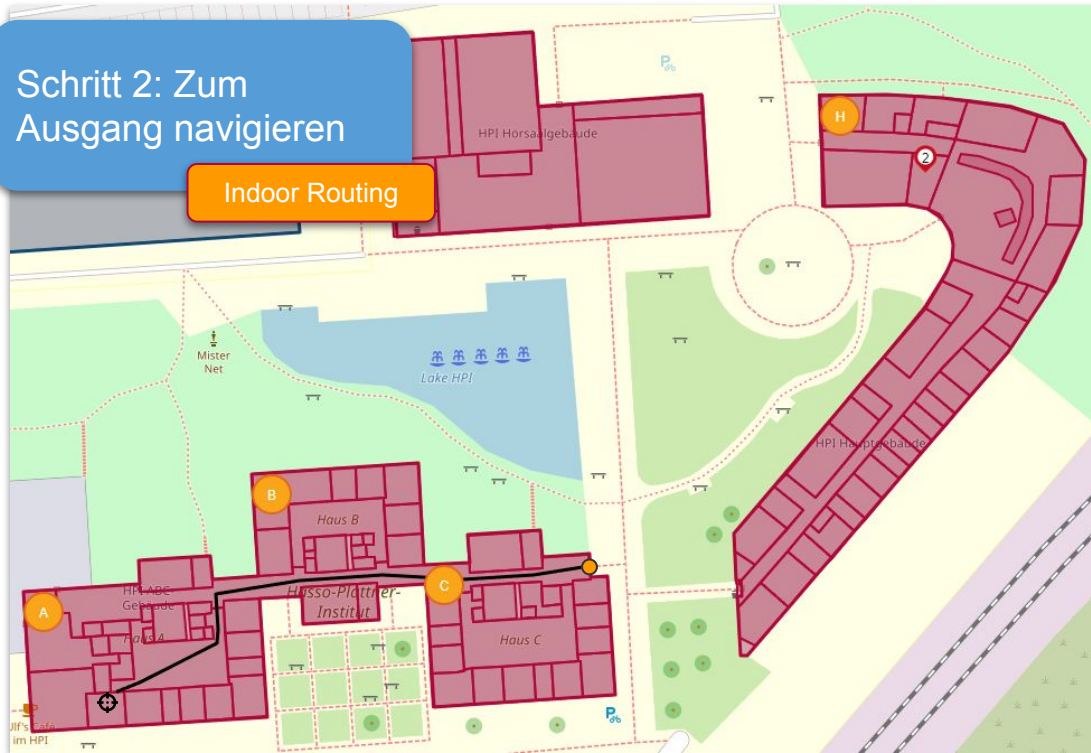
Endpräsentation  
Team Blau

# Indoor Navigation - Wie berechnen wir Routen?

Beispiel: A-E.6 zu H-2.58

Schritt 2: Zum Ausgang navigieren

Indoor Routing



Endpräsentation

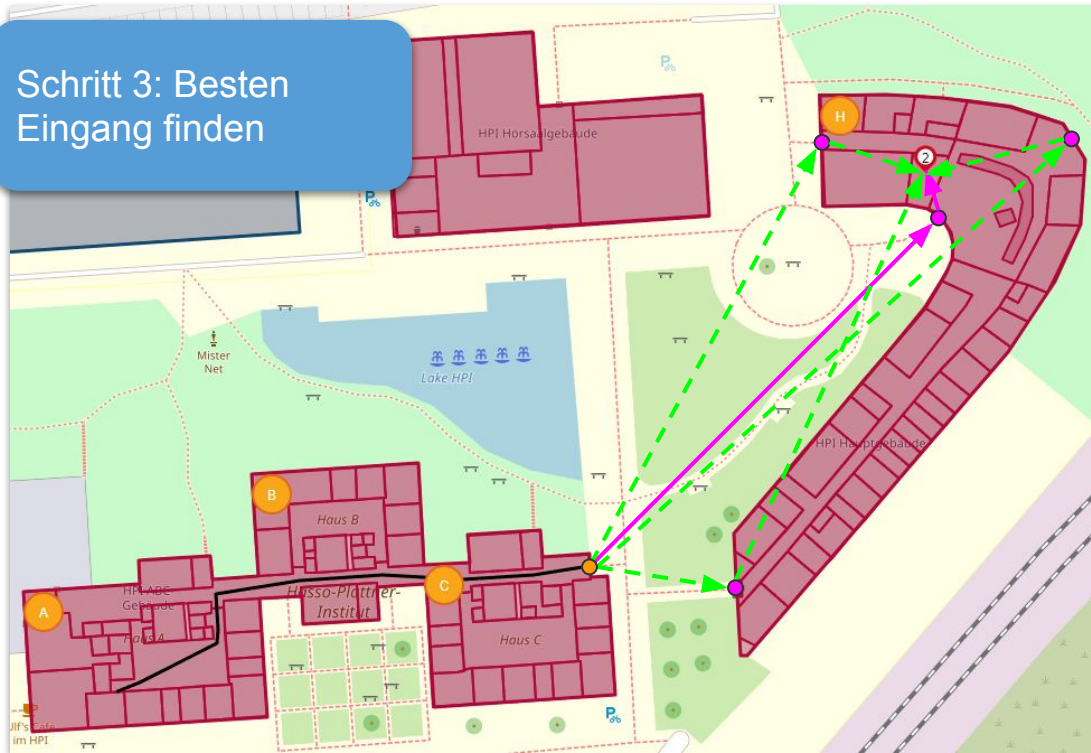
Team Blau



# Indoor Navigation - Wie berechnen wir Routen?

Beispiel: A-E.6 zu H-2.58

Schritt 3: Besten Eingang finden



Endpräsentation

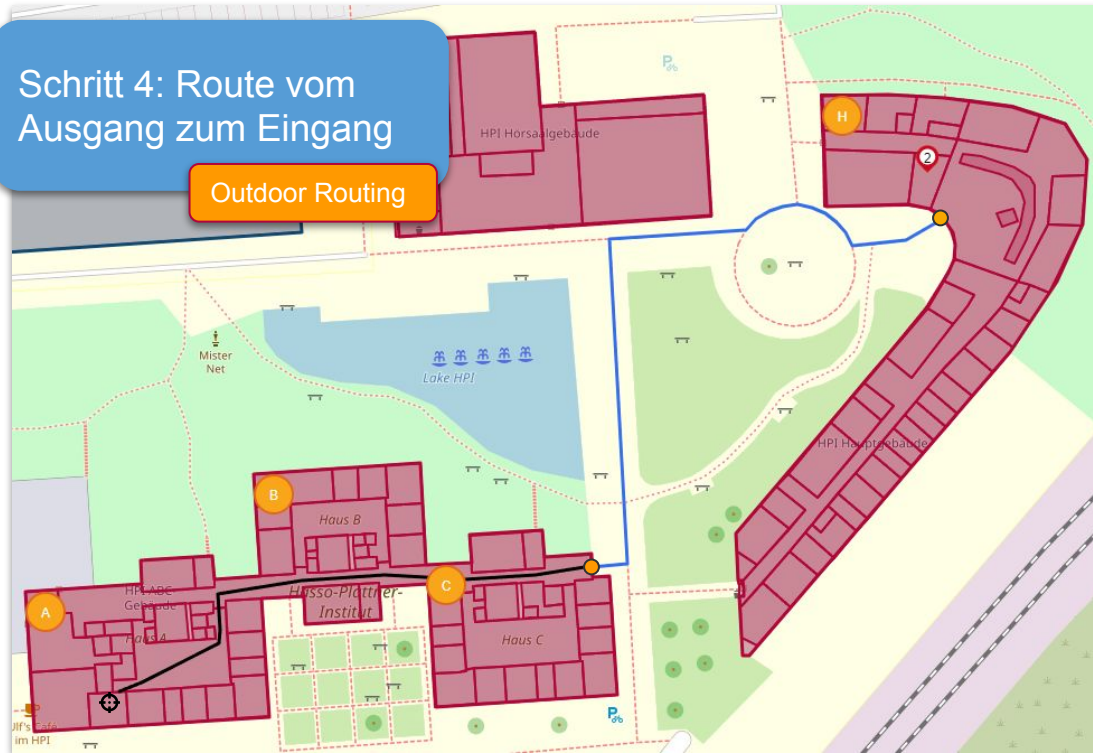
Team Blau

# Indoor Navigation - Wie berechnen wir Routen?

Beispiel: A-E.6 zu H-2.58

Schritt 4: Route vom Ausgang zum Eingang

Outdoor Routing



Endpräsentation

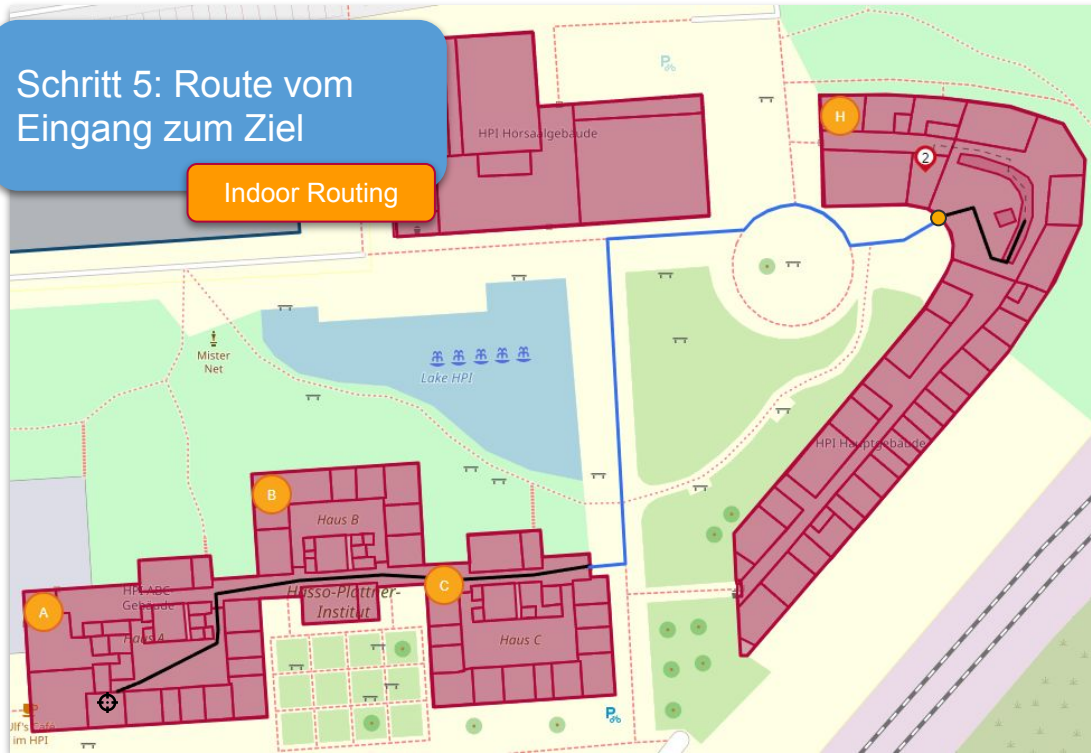
Team Blau

# Indoor Navigation - Wie berechnen wir Routen?

Beispiel: A-E.6 zu H-2.58

Schritt 5: Route vom Eingang zum Ziel

Indoor Routing



Endpräsentation

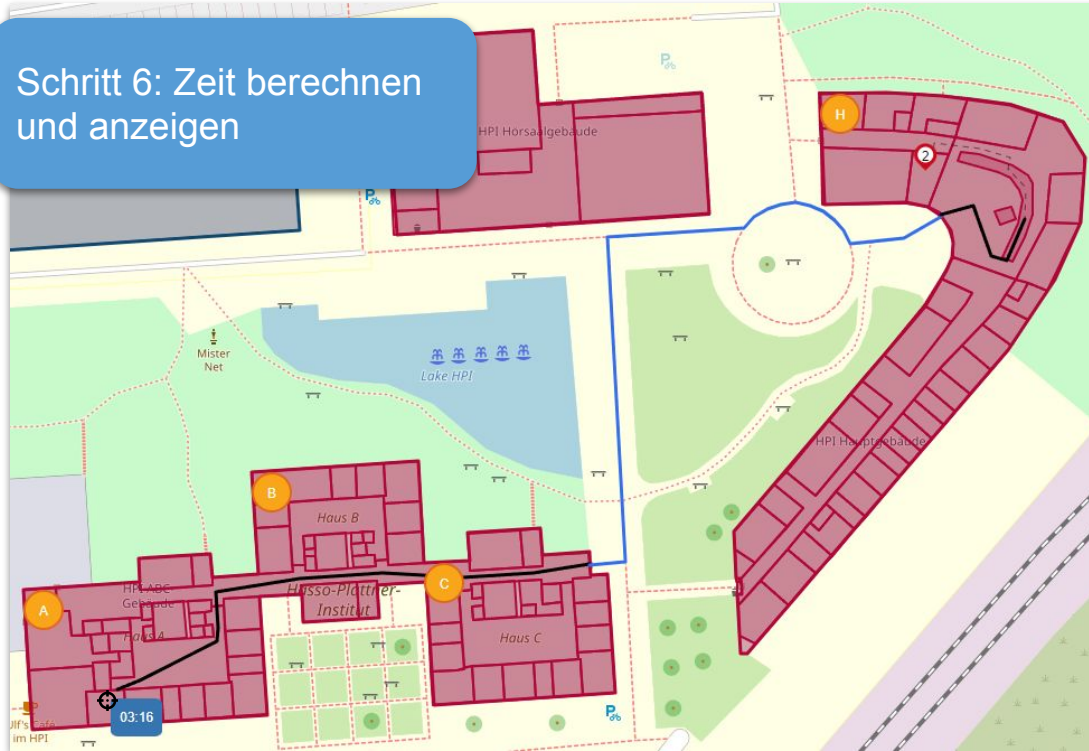
Team Blau



# Indoor Navigation - Wie berechnen wir Routen?

Beispiel: A-E.6 zu H-2.58

Schritt 6: Zeit berechnen und anzeigen

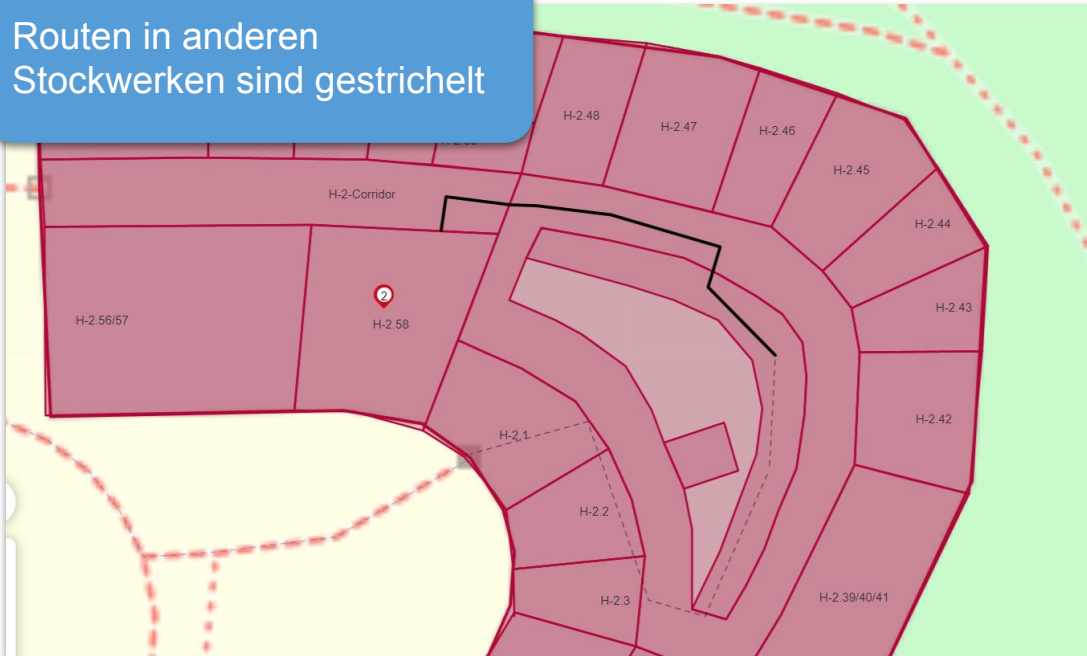


Endpräsentation  
Team Blau

# Indoor Navigation - Wie zeigen wir Routen an?

Beispiel: A-E.6 zu H-2.58

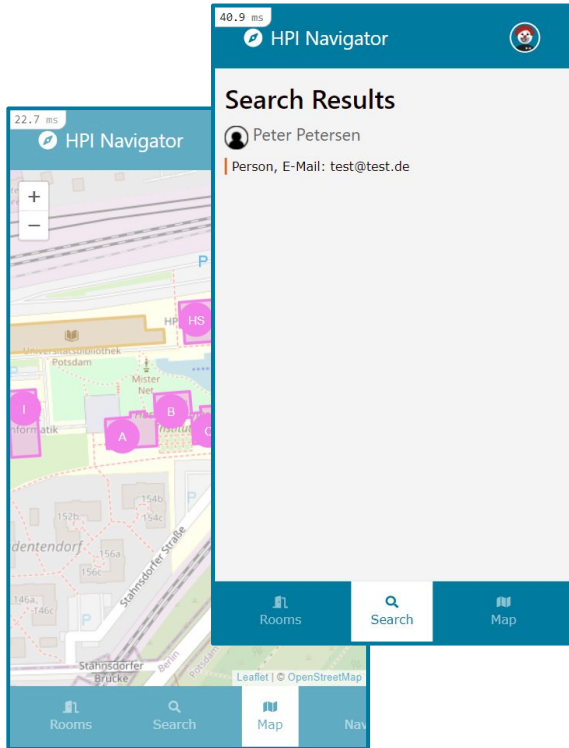
Routen in anderen Stockwerken sind gestrichelt



Endpräsentation

Team Blau

# Single Page Application - Motivation

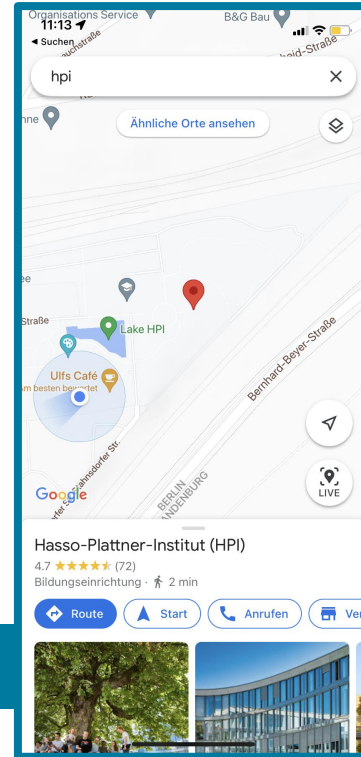


Alter Stand



Ziel

wie Google Maps



Endpräsentation  
Team Blau

# Single Page Application - Anforderungen

- Karte im Zentrum, alles soll sich wie eine Seite anfühlen
- Nutzende sollen sich nicht für Karte **oder** Suche entscheiden müssen
- Suchergebnisse werden direkt auf der Karte markiert
- dynamisch Inhalte in einen ausklappbaren Bereich laden

Show Map on main page, with search bar visible (overlay) #127

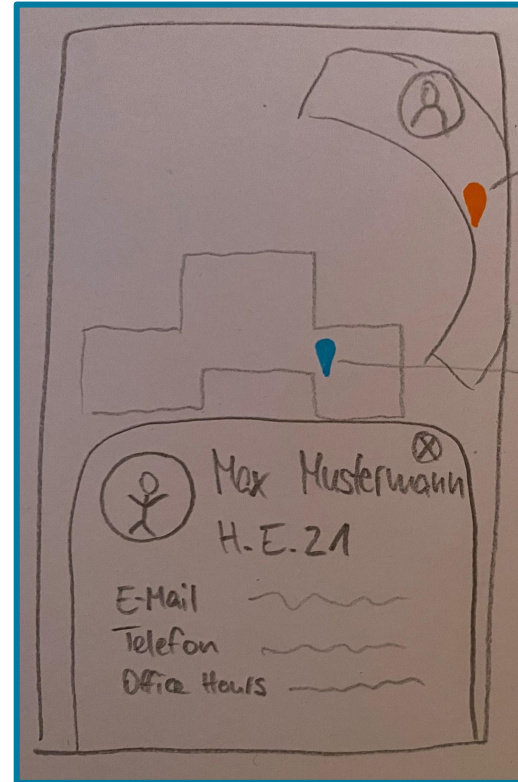
 Closed [hendrix15](#) opened this issue on 5 Jan · 2 comments · Fixed by #153



hendrix15 commented on 5 Jan

Member  

As a user I want to see the map when I open the app. The search bar should be available as well. I want to be able to use these two core features when entering the web page.



Endpräsentation  
Team Blau

# Single Page Application - Asynchrone Kommunikation

**Idee:** `iframes` nutzen

- "inline frames" mit denen Inhalt anderer Seiten angezeigt werden kann

**Pro:**

- einfach umzusetzen, da native Browser-Funktionalität
- Navigation bleibt im Frame ohne die Browser-Seite zu wechseln

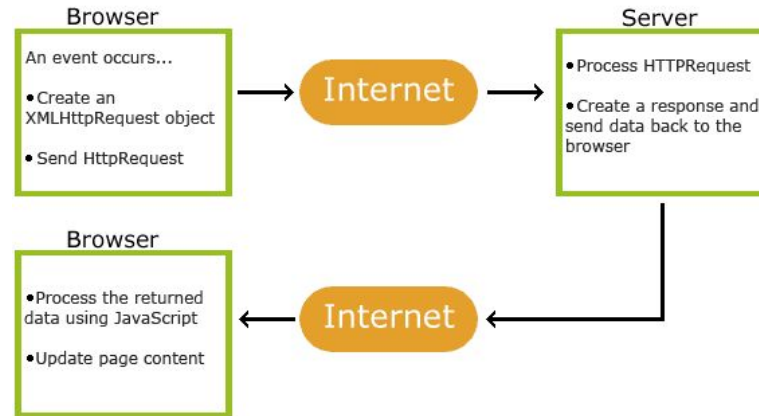
**Kontra:**

- aus Sicherheitsgründen ist Interfacing mit umliegender Website nicht möglich (z.B. per JavaScript) → für Karte ungeeignet

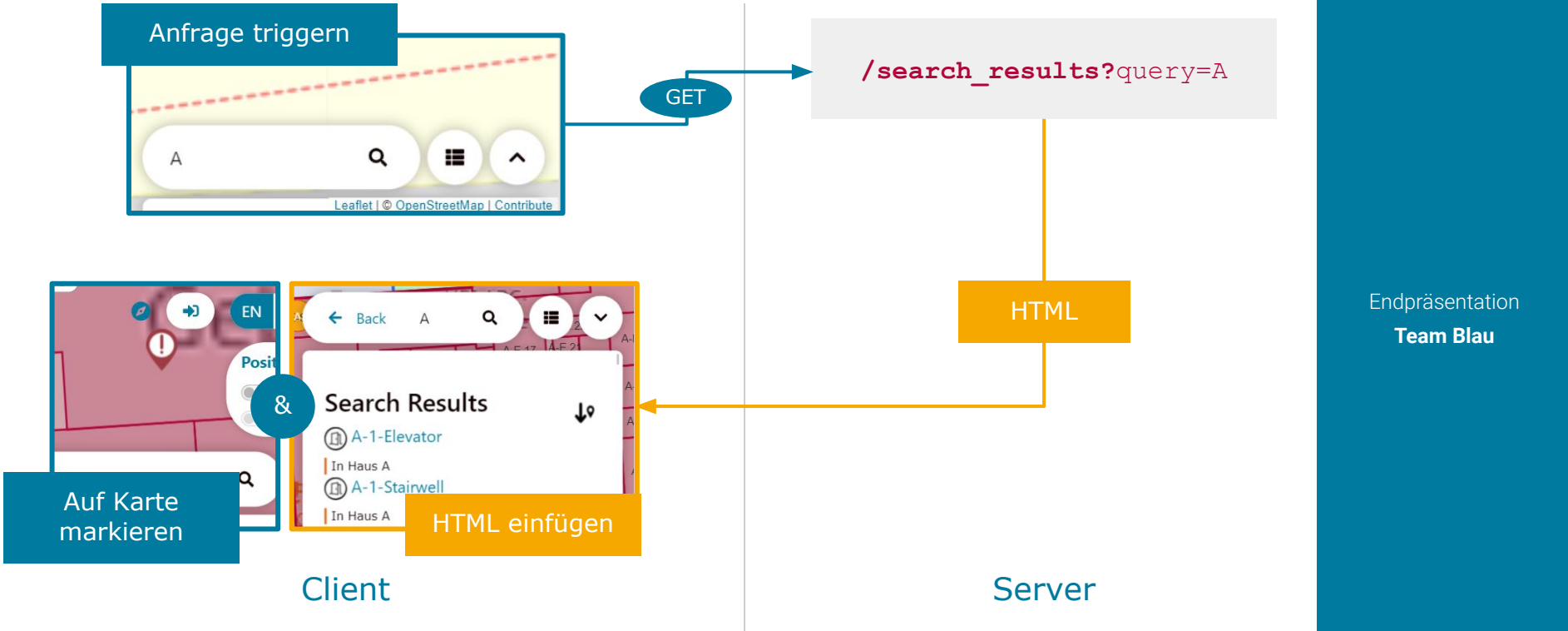
# Single Page Application - Asynchrone Kommunikation

Idee: **AJAX** nutzen

- **A**synchronous **J**avascript **A**nd **X**ML
- Methode, um HTTP-Anfragen an den Server zu stellen und die Antwort per Javascript zu verarbeiten
- genutzt, um existierende Seiten (z.B. show pages) anzufragen und in das browse-outlet einzufügen



# Single Page Application - Asynchrone Kommunikation



# Single Page Application - Asynchrone Kommunikation

```
1 $.ajax({
2   type: "GET",
3   url: target,
4   data: valuesToSubmit,
5   dataType: "html",
6   success: function (text) {
7     const parsed = new DOMParser().parseFromString(text, "text/html");
8     const content = parsed.querySelector("#app-outlet-outer");
9     $("#browse-outlet").html(content.innerHTML);
10    applyLocation();
11    applyAjaxWrap();
12  },
13 });
```

```
1 function applyLocation() {
2   const latLongInfo = content.querySelector("#_latlonginfo");
3   if (latLongInfo) {
4     const latlong = JSON.parse(latLongInfo.innerHTML);
5     showTargetMarker(latlong);
6   }
7 }
```



# Single Page Application - Asynchrone Kommunikation

```
1  const applyAjaxWrap = () => {
2    $("#browse-outlet a").each(function () {
3      const link = $(this);
4      const href = link.attr("href");
5      if (href.match(/(mailto|tel):(.*)/)) {
6        return;
7      }
8      // Remove the original href of the link:
9      link.attr("href", "#");
10     const [baseLink, queryParams] = href.split(/\?(.+)/);
11     link.on("click", (event) => {
12       ajaxCall(event, baseLink, queryParams, true);
13     });
14   });
15 }
```

# Single Page Application - Diskussion AJAX

---

## Technische Herausforderungen

- Browser-History muss manuell gecatcht und verwaltet werden
- Handling von nicht-GET Requests

## Kommunikation

- Grundlegende Änderung → Wissen muss an alle verbreitet werden
  - technisches Wissen
  - Kontextwissen (z.B. andere Show-Page URLs)
- **Lösungsansatz:** Erklärvideo

## PO Retro - Was hat gut geklappt

- konstruktiver und respektvoller Austausch mit den Teams
- tolles Ergebnis
- gemeinsam am Projekt und am Prozess gewachsen

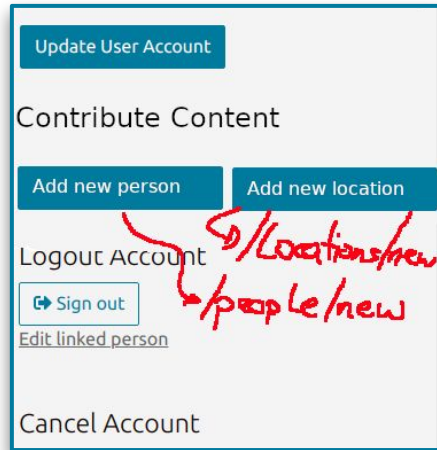


Endpräsentation

**Team Blau**

# PO Retro - Was hat gut geklappt

- **Skizzen/Mockups** zum Festhalten der Vorstellung eines Features



VS.

"As a user I want to be able to create new persons and locations. I want to get to this page from my users/edit page by clicking on a button (e.g. 'Add new Person' & 'Add new location'). The buttons should be placed into their own section with the title 'Contribute Content'. When the user clicks on the Button they should be redirected to the creation pages for persons and locations...."

- höherer Zeitaufwand für PO
- User Stories (wirken) weniger verhandelbar
- + Verständnis über Optik des Features

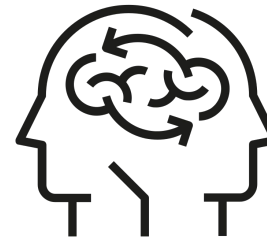
- Interpretation obliegt den Entwicklenden  
→ verschiedene Vorstellungen
- Was muss erwähnt werden, was nicht?

## PO Retro - Was könnten wir noch verbessern



- **Akzeptanzkriterien** klarer formulieren
  - häufig **kleine Missverständnisse** (zum Glück keine schwerwiegenden)
  - Akzeptanzkriterien waren z.T. nicht aussagekräftig genug
  - **Skizzen** schienen zu helfen

- **Wissensaustausch** fehlte zum Ende
  - "**Silos**": SPA, Backend, Rechtesystem, Kalender, Navigation & Karte



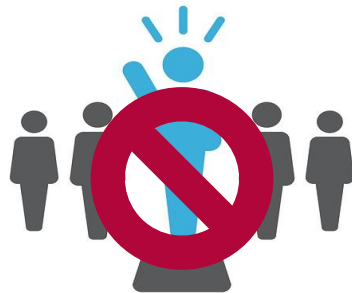
## PO Retro - Was könnten wir noch verbessern

- **Design** wurde schwieriger
  - erst hauptsächlich von einem Team umgesetzt → einheitlich
  - spätestens in Kanban verschiedene Einflüsse
  - Skizzen zu ungenau? Wollen wir wirklich mit Mockups alles "diktieren"?

gemeinsame Arbeitszeit	"freie" Arbeitszeit
<ul style="list-style-type: none"> <li>+ einfachere Planung, Kommunikation, Erreichbarkeit</li> <li>+ Sustainable Development</li> <li>- Austausch mit anderen Teams komplizierter</li> </ul>	<ul style="list-style-type: none"> <li>- (gefühl) weniger Kommunikation zwischen Dev und PO</li> <li>+ für Studierende leichter zu realisieren</li> </ul>

## PO Retro - Was uns als POs aufgefallen ist

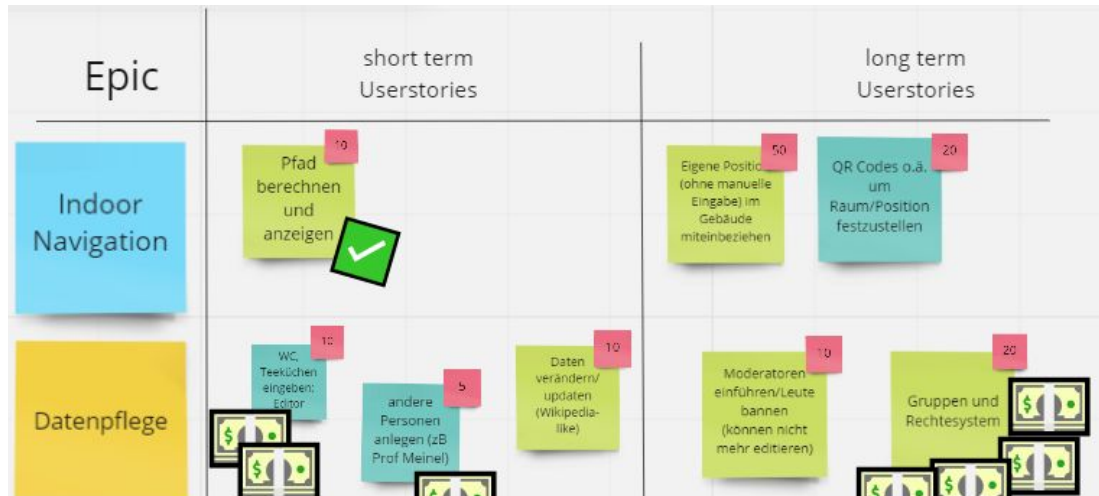
- wichtig, **nicht als "Boss"** aufzutreten
  - immer daran denken, jedem/r eine Wahl zu lassen, was er/sie lieber machen möchte
  - kaum Hierarchie da wir alle Studierende sind
- **kein technisches Wissen** zu haben ist spannend
  - nicht "den Helden spielen", falls verplant / falls etwas nicht klappt
  - Absprachen und Vertrauen wichtig - hat auch gut geklappt



# PO Retro - Kundenkommunikation

**Problem:** Wir möchten vom Kunden wissen, welche Features und User Stories Priorität haben

**Idee:** "Monopoly Spiel" - Der Kunde hat ein bestimmten Budget und kann es für die Features ausgeben, die ihm am wichtigsten sind





# PO Retro - Kundenkommunikation

## Vorteile:

- Kunde ist gezwungen zu selektieren (es kann nicht alles gewählt werden)
- detaillierte Diskussion über mögliche Features
- Kernfeatures und größter Kundennutzen können ermittelt werden

## Gefahren:

- Features müssen klar definiert sein
- Schätzung der Kosten eines Features ist im voraus sehr schwierig
- Dem Kunden muss deutlich gemacht werden, dass es sich nur um eine Priorisierung handelt und nicht um ein Versprechen

**Was wir beim nächsten Mal anders machen würden:**  
weniger Budget → Kunde muss stärker priorisieren

## PO Retro - Scrum vs Kanban

Scrum	Kanban
<ul style="list-style-type: none"> <li>+ Informationsaustausch</li> <li>+ <b>leichtere "Steuerung"</b> der zu wählenden Stories durch Priorisierung</li> </ul>	<ul style="list-style-type: none"> <li>+ keine festen Meetings</li> <li>+ geringerer Zeitaufwand</li> </ul>
<ul style="list-style-type: none"> <li>- Schätzen eher nervig, Resultat identisch</li> <li>- mehr Meetings</li> <li>- mehr Zeitaufwand</li> </ul>	<ul style="list-style-type: none"> <li>- <b>kein Sprint (-ziel)</b> <ul style="list-style-type: none"> <li>- weniger Motivation</li> <li>- keine gemeinsame Vision</li> </ul> </li> <li>- weniger Struktur, kein Platz für "offizielle" Infos</li> <li>- "uninteressante" User Stories bleiben liegen</li> </ul>

# PO Retro - Beispiel i18n

## Wichtigste Kundenwünsche



Karte des HPI



Suche nach Personen, Orten und Räumen



Navigation & Accessibility



“Wir machen erstmal nur Englisch”



- Intuitiv nutzbar
- Auf Deutsch und Englisch verfügbar

PO-Präsentation zum Projektstart

Translation of old code #123

Closed anonym-HPI opened this issue on 4 Jan · 0 comments



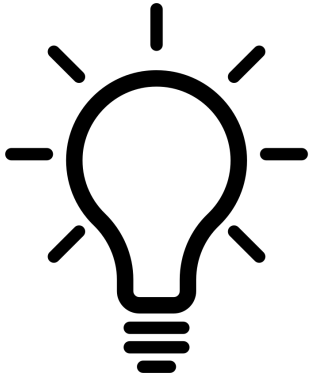
speydril moved this from Selected \_ Limit: 3 to Develop – Ongoing \_ Limit: 3 in Team CMJD – Kanban 2 days ago

2 days ago = 15. Februar

Endpräsentation  
Team Blau

## PO Retro - Beispiel i18n

- Fleißarbeit: niemand hat wirklich Lust auf i18n
- in Kanban werden immer andere Tickets bearbeitet



### **Lösungsidee:**

- i18n in Definition of Done aufnehmen
- bei jedem Feature für PO Approval beachten

# PO Retro - Was passiert, wenn man gegen den Kanban-Flow kämpft

- für das Video wurde ein laufendes Ticket unterbrochen
- an Devs in anderem Team übergeben  
→ Ticket bewegt sich entgegen Kanban-Flow

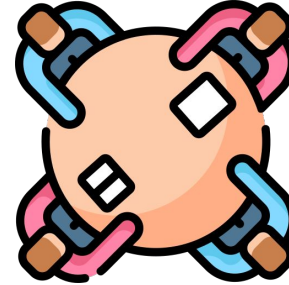
## Folgen:

- Verwirrung
- lange Bearbeitungszeit
- unklare Zuständigkeiten



## SM Retro - Was haben wir gelernt?

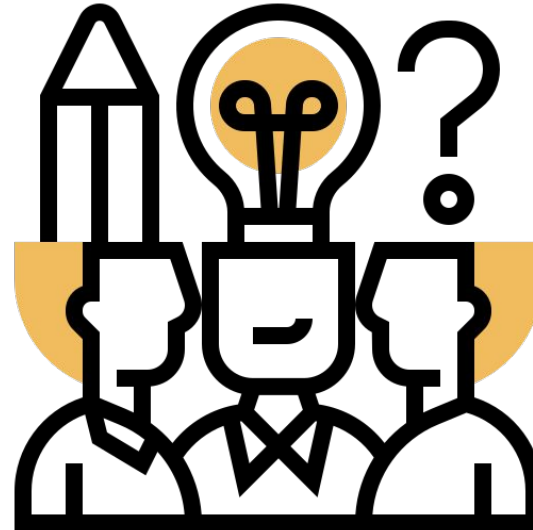
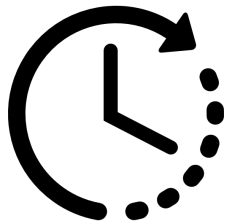
- **Meeting Plan und Struktur**
  - extrem wichtig (mehr als erwartet)
  - sonst extremer Kommunikationsoverhead
  
- **Ergebnisse**
  - müssen ebenfalls protokolliert werden  
→ Scrum Diary ist sinnvoll



## SM Retro - Was war cool?

- **Zusammenarbeit im Scrum-Team**
  - gegenseitiges Unterstützen + Kommunikation

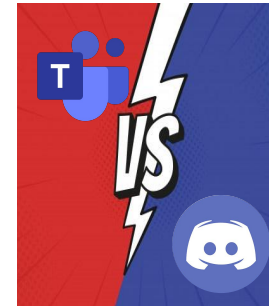
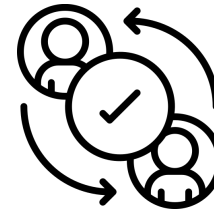
- **Meetings mit Agenda und festen Zeitrahmen**



## SM Retro - Was hätten wir uns gewünscht?



- **am Anfang noch etwas mehr Anleitung/ Heranführen an die SM-Rolle**
- **Scrum und Kanban die gleiche Zeit/Chance geben**
- **mehr Interaktion im Meeting**
- **früherer Umstieg auf Discord und bessere Abstimmung zwischen den Teams**

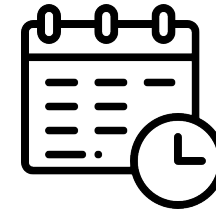




## SM Retro - Scrum Artefakte

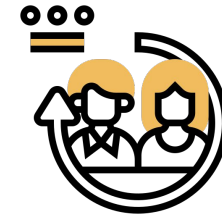
Welches war am herausforderndsten?

- **Planning**
  - Schätzen gerade bei begrenztem Domänenwissen herausfordernd
  - User Stories mit Abhängigkeiten schwerer zu verteilen
  - teilweise wurde das Sprint Ziel nicht genügend eingebunden



Was ist das Lieblings Scrum Artefakt?

- **Retro**
  - schärft Verständnis für andere Teammitglieder
  - spannende Einblicke in andere Perspektiven
  - bei einigen Teams gerade die konkreten Action Points sehr hilfreich



# SM Retro - Herausforderungen

## Sicht der SM Rolle

- anfangs teilweise Skepsis gegenüber Scrum
- **Balance** zwischen Agenda und jede/n ausreden lassen
  - Fokus auf das Wesentliche



## Vereinigung der SM und Dev Rolle

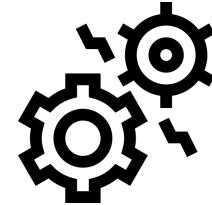
- Zeitmanagement!
- Team "alleine" lassen und unsichtbare Aufgaben übernehmen
- SM darf keine erhabene Rolle einnehmen  
(keine Hierarchie im Team)



## SM Retro - Wie war die SM-Rolle?

### fordernd

- **besonders: Geteilte Dev-SM-Rolle**
  - nicht sichtbare Arbeit
  - Pair Programming
  - SM als Ansprechpartner für Alles
    - zeitliches Vereinen mit weiteren SM-Aufgaben herausfordernd



### → Lösungsansatz

Abgabe an anderes Team nach Absprache

### fördernd

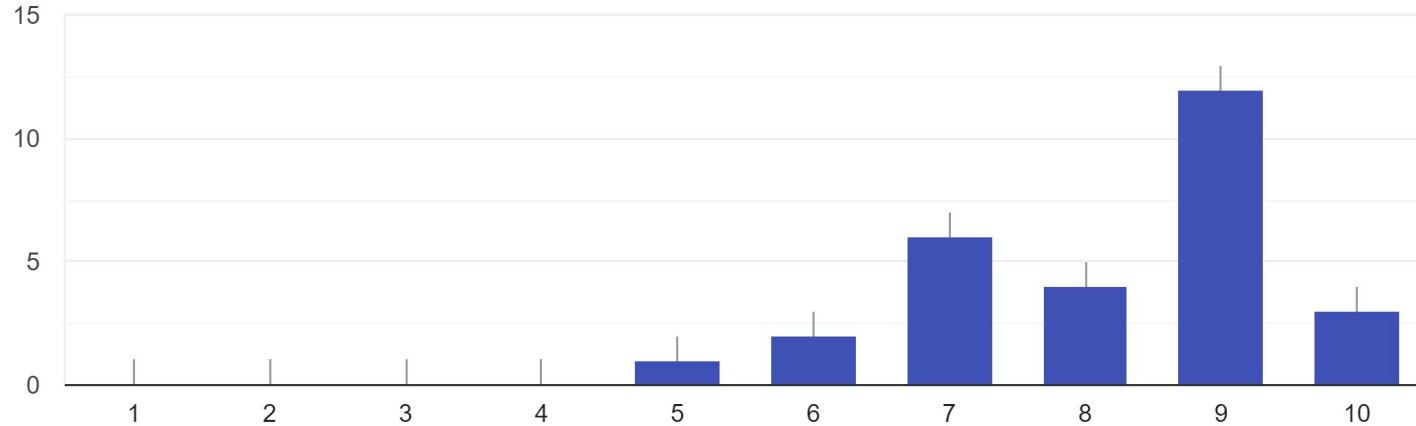
- insgesamt spannende Erfahrung
- Spaß daran wenn Meetings + Prozesse sich verbessern



# Umfrage - Team Blau

Wie zufrieden bist du mit dem Ergebnis des Projekts?

28 Antworten

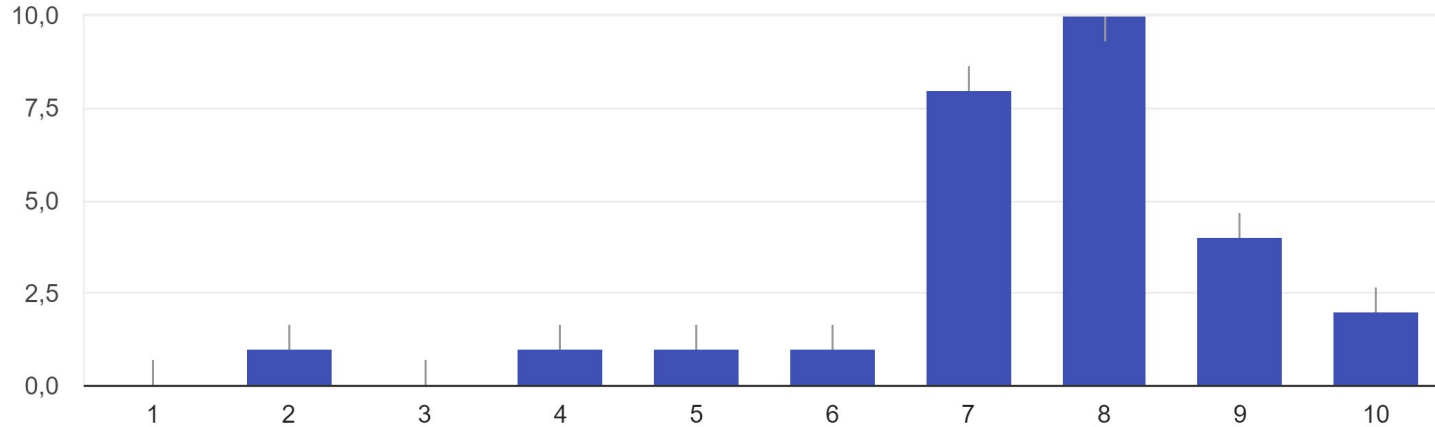


Endpräsentation  
Team Blau

# Umfrage - Team Blau

Wie zufrieden bist du mit dem Prozess?

28 Antworten

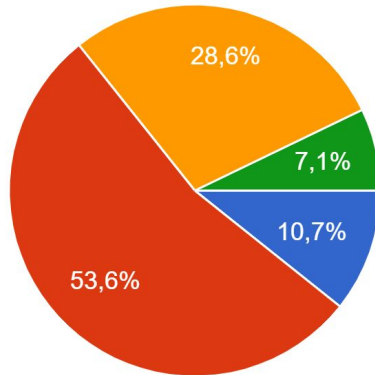


Endpräsentation  
**Team Blau**

# Umfrage - Team Blau

Welche Projektphase hat dir am besten gefallen?

28 Antworten



- Auf jeden Fall Scrum
- Eher Scrum
- Eher Kanban
- Auf jeden Fall Kanban

Endpräsentation

**Team Blau**

# Umfrage - Was hast du von der Projektarbeit mitgenommen?

- Wie die SM- und PO-Rollen in der Zusammenarbeit funktionieren
- Wie **effiziente Meetings** aussehen
- Projektarbeit im großen Team

**Integrate early** wichtiger als man denkt

**Ruby ist cool**

Gute **Kommunikation** ist sehr wichtig

Entwicklung eines Produkts in **größerem Rahmen**

bei klarer **Aufgabenverteilung** läuft ein Projekt mit so vielen Leuten

Wie viel **Kommunikationsoverhead** entstehen kann und wie feste Strukturen diesen zwar immer noch nicht beseitigen, aber zumindest **beherrschbar machen**

Wie kompliziert es ist, viele Programmierer:innen zu syncen, sodass immer alle alles wichtige wissen aber nicht zu viel **zeit für Meetings** drauf geht

**Kommunikation** ist das Wichtigste. Außerdem muss der **Prozess** für alle klar sein und sollte falls nötig auch angepasst werden.

# Umfrage - Was hast du von der Projektarbeit mitgenommen?

**Kommunikation** zwischen und in Teams ist essentiell

**Schätzen ist schwierig** in so einem großen Projekt (gerade bei Kanban) ist präzise Formulierung der Issues super wichtig wie sehr man in der **Kommunikation** untereinander aufpassen sollte, dass nicht sowas wie "(Team) XY hat schon wieder..." nicht zur Gewohnheit wird  
**häufige Retros** sind sinnvoll (ohne in ein "alles läuft immer nur hervorragend"-Muster zu kommen)

Eine **Abneigung** gegen **Web-Development**

Arbeiten in großen Teams kann funktionieren, wenn die **Kommunikation** stimmt.

**Agenda** in **Meetings** ist ein must

Es ist echt wichtig, eine gute **Teamdynamik** untereinander zu entwickeln. So kann man regelmäßig in den Retros den Prozess improven, ohne die "Blame" Frage oder andere persönliche Streitigkeiten erst aufkommen zu lassen.



# Umfrage - Was hast du von der Projektarbeit mitgenommen?

Man muss auch mit unfähigen Leuten/Teams zusammenarbeiten.

**Product Owner** muss ein **grauehafter Job** sein, den ich hoffentlich niemals übernehmen muss.

Erfahrung für Arbeit an größerem Projekt.

**Kommunikation** in einem großen Team ist zwar schwierig, aber mit einer entsprechenden **Organisation** wie vor allem in Scrum und einer Plattform die von allen genutzt wird war es erstaunlicherweise **leichter als erwartet**.

Die hohe Bedeutung von **Kommunikation**

Das es **cool** ist, in **größeren Gruppen** zu arbeiten, sofern man gut mit diesen **kommunizieren** und sich auf sie verlassen kann

Es ist **schwieriger** in einem **großen Team** zu arbeiten als gedacht. Viele Aufgaben dauern länger als man denkt. Die **Softwarearchitektur** wird meist **wild**.

# Umfrage - Was ist das interessanteste, das du gelernt hast?

## Praktische Anwendung von **Scrum und Kanban**

Das **Scrum und Kanban** zwar ganz gut erprobte Methodensets sind, aber man sich als **Team** immer damit auseinandersetzen muss, welche Elemente von beiden Sets am besten zum eigenen Team passen.

wie wichtig eine gemeinsame **Vision** und ein gemeinsames **Ziel** sind

## **Ruby on Rails**

## Umgang mit **Javascript & Cookies**

**ajax**

**Ruby on Rails** kann "interessant" sein.

**Ruby On Rails** - ist aber auch das **absurdeste**, das ich gelernt habe.

mit vielen Leuten in **geordneten Strukturen** ist in kurzer Zeit viel erreichbar

Der Prozess der Entwicklung von software im **großen Team** generell

**Kanban-Prozess** ausprobiert, **Ruby-On-Rails**

# Umfrage - Was ist das interessanteste, das du gelernt hast?

Die konkrete Aufteilung/Struktur der **Scrum-Meetings**.

Der Unterschied zwischen **Kanban und Scrum** klingt in der Theorie nicht so groß, ist in der Praxis dann aber doch merklich **anders** vom gesamten **Workflow und der Organisation** her.

## GitHub Workflows

Wie schwierig es ist nicht nur im kleinen Team sondern vor allem auch zwischen den Kleinteams zu **kommunizieren**. Wie teamübergreifende **Dependencies** den Workflow verlangsamen.

Frontend Tests mit **Capybara** und **Selenium**, auch wenn Selenium viele Schmerzen bereitet

Gute und klare **Koordination** und **Kommunikation** ist sehr wichtig.

# Umfrage - Was wünschst du dir für zukünftige Projekte?

## mehr **Kommunikation**

Teilweise bessere **Kommunikation** zwischen Teams

mehr **Wissenstransfer** zwischen den Teams (& mehr Zeit)

## Mehr **Kommunikation**

Bessere **Kommunikation**, mehr Zeit um daran zu arbeiten.

Gleichere **Arbeitsverteilung** (schwer umsetzbar)

Ein selbst anpassbaren **Scrumbanstil**. Gute SMs und POs -> erleichtern das Leben

Weniger Kanban, mehr Scrum. **Kommunikation** zwischen den einzelnen Teams/Devs sollte von Beginn an besser sein. Wichtige **Informationen und Designentscheidungen** sollten **zentral gesammelt** und für alle zugänglich sein. Besonders Designentscheidungen einzelner Teams oder der PO's war Teilweise nicht für alle Teams ganz nachvollziehbar.

Weniger Menschen (aka **kleinere Teams**)

Dass ich mit Leuten zusammenarbeiten kann, die **genauso zuverlässig** sind, wie meine **Teamkollegen** es in diesem Semester waren.

# Umfrage - Was wünschst du dir für zukünftige Projekte?

Ein **anderes Framework**, als Ruby on Rails bzw. gar nicht erst Web-Development

**Mehr Einarbeitung** in ein Framework bevor das eigentliche Projekt startet

**CI für JS Formatting** (von Anfang an)

Man sollte einen **Tech-Stack** benutzen, den mehr Leute schon **kennen**, da dann der Fokus noch mehr auf den Prozess gelegt werden könnte.

eine **andere Programmiersprache**

**respektvollen Umgang** untereinander (hat bei uns nicht gefehlt, aber sollte mMn essentielle Grundlage von allem sein)

**Spaß**

**spannende technische Herausforderungen**

Dass man nicht nur ein **Projekt** erarbeitet, was dann, ich sags mal vorsichtig, **in der Tonne** landet. Und selbst wenn das mal irgendwann genutzt werden sollte, wer entscheidet dann ob es die Lösung von Team Blau oder Rot wird? Generell halte ich es **nicht** für **sinnvoll**, **zwei competing Teams dieselbe Softwarelösung** bauen zu lassen, wenn doch eh klar ist, dass davon maximal eine Lösung überhaupt mal benutzt wird...

# Umfrage - Was würdest du zukünftigen SSE-Studierenden empfehlen?

Issues zwischen Teams "**vertikal**" anstatt "**horizontal**" aufteilen. **Dependencies** zwischen Teams so gut es geht **minimieren**.

Von Anfang an **am Prozess feilen**.

Bei Fragen und Unklarheiten nicht allzu lange warten sondern schnell **auf andere Teams/Devs zugehen**. Große Änderungen entsprechend **Kommunizieren**.

Direkt **Discord** statt teams nutzen

**Belegt es während ihr auch im BP seid**, man reflektiert doch noch einmal anders, wenn man die Prozesse in SSE mit denen im BP vergleichen kann.

Mehr **Kommunikation**

Sich **einzulassen** auf den **Prozess**.

# Umfrage - Was würdest du zukünftigen SSE-Studierenden empfehlen?

**probiert** mal die **PO Rolle** ;)

Wenn ihr den Code beim reviewen **nicht versteht**, schreibt das ran!

mit den anderen Teams **REDEN**, das Projekt als Chance sehen, versuchen **hilfreiche Reviews** zu machen

Immer schön **aktiv mitarbeiten**

Sei **kein Product Owner!** :)

**nicht Scrum-Master** werden, da dieser doppelt so viel Arbeit hat

mit erschrecktem Blicke Richtung Team Rot, es **entspannt(er)** und mit **viel Zusammenarbeit** anzugehen.

# Umfrage - Wie ist dein allgemeines Fazit?

---

Überraschend **positiv**. Die **Kommunikation** zwischen den kleineren Teams war besonders Anfangs etwas holprig aber ich fand, dass es im Ganzen **sehr gut zusammen gekommen** ist.

Es war eine sehr **hilfreiche** und **neue Erfahrung**, die teilweise jedoch auch **frustrierend** war was eventuell daran lag, dass manche der Praktiken **nicht sinnvoll in einer Lehrveranstaltung** angewandt werden konnten.

Eigentlich ganz **lustig**, aber es ist schwierig **Zeit in der Gruppe** zu finden, wenn alle natürlich nicht Vollzeit am Projekt arbeiten. Schon frustrierend so

Cooler Projekt. Ich bin froh in **Team Blau** gelandet zu sein. **Zusammenarbeit** hat **Spaß** gemacht



# Umfrage - Wie ist dein allgemeines Fazit?

<3

sehr **cooles Projekt**, das schwierigste waren Absprachen und lange Review-Zyklen

Es hat alles deutlich **besser geklappt**, als ich es mir je vorgestellt hätte.

**Team Blau > Team Rot**

**lehrreiche Veranstaltung**

bei einem langfristigeren Projekt wäre mehr **Dokumentation sinnvoll** gewesen

Es war ein **gutes Projekt** und hat mir echt viel **Freude** bereitet. Es war nicht stressig und man hat viel mitgenommen.

Endpräsentation  
**Team Blau**

# Zusammenfassung



- spannendes Projekt
- viel gelernt, Spaß und auch Mal Frust gehabt
- gelernt, wie wichtig Kommunikation und Planung/Struktur ist
- Pro- und Contra von Scrum und Kanban probieren, wenn auch zu wenig Zeit für Kanban
- Projekt was ins blaue gehen kann hat Möglichkeit des ausprobierens, aber auch Risiko von fehlender Motivation
- desto größer ein Projekt, desto leichter ist es Bugs zu übersehen (nicht alles decken Tests akkurat ab)
- Blue + Red ? -> Blue vs. Red



## Mögliche Fortführung

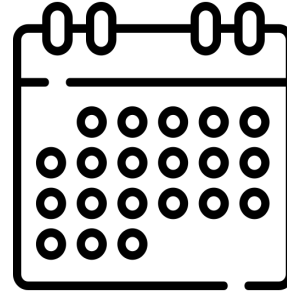


- ???



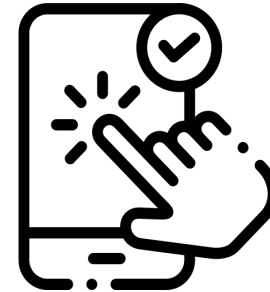
### Kartendaten

- Indoor Maps  
Campus III
- POIs



### Raumkalender

- aut.  
importieren  
ermöglichen



### UX-Verbesserungen

- mehr mobile  
first
- Designverein-  
heitlichung

# Bildquellen

---

<https://vignette.wikia.nocookie.net/monopoly/images/4/44/Bay-Monopoly-2.jpg/revision/latest?cb=20180808001004>

[https://upload.wikimedia.org/wikipedia/commons/thumb/6/69/How to use icon.svg/1200px-How to use icon.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/6/69/How_to_use_icon.svg/1200px-How_to_use_icon.svg.png)

<https://wallsheaven.de/fototapeten/3d-m%C3%A4nnchen-am-runten-tisch-blaue-zahnr%C3%A4der-B48904675>

<https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSHFwcYIwbpolKmv6S22xi5QYqjXq-VHCcG0q&usqp=CAU>