**Ziele:**

* bessere Planung
* Produktionsoptimierung
* (nahezu) Echtzeit Auskunft über Ladungsträger

**Risiken / Probleme:**

* Bilder sind wegen Austausch offen zugänglich (Datenschutz) (evtl vor und nach schicht beginnt, (ggf. gesichts/kopf erkennung und entsprechend zensieren))
* QR verschmutzt / nicht lesbar

**Offene Fragen:**

* Kameratyp (Auflösung) – Raumgröße im Verhältnis zur Kamera für genauere Bestimmung der Position(en) → Wie viele Kameras? → möglichst wenige Kameras (4k für 80 €)
* Kriegen wir die 3D Modelle der Ladungsträger oder selber erstellen   
  → item.engineering
* Wie werden die Kameras positioniert (Sind sie geneigt?) → von uns frei definierbar
* Müssen wir die QR-Codes selber generieren? → Ja, bzw. von item.engineering

* Benutzerverwaltung? → Rechtesystem? (Wer kann Ladungsträger anlegen/verwalten) → kein Rechtesystem
* Schnittstelle von Aufnahmen (Bildern) zu Software → Shared Ordner über Netzwerk, oder Standort spezifische Angaben

→ EvoBus passt sich uns an! Wie wir das machen möchten egal - Infrastruktur wird darauf aufgebaut

* Wird das Bild für die Aktualisierung der Standorte geliefert (alle x Minuten im Shared Ordner), oder lösen wir mit unserer Software die Kamera selbst aus?

→ EvoBus passt sich uns an! Wie wir das machen möchten egal - Infrastruktur wird darauf aufgebaut

* Wer stellt die Kamerainfrastruktur zur Verfügung?
  + Wer löst das Bild aus?
  + Wer kümmert sich um Speicherort?
  + Wer stellt Kamera zur Verfügung?
  + Verkabelung

→ Infrastruktur komplett frei von  uns definierbar → lokalen Ordner auf selbem Rechner

* Wie viele Ladungsträger befinden sich im Bereich einer Station gleichzeitig?

* <10
* 2 - 5 hinbekommen

**Besprechung 21.10.2020**

item.engineering:

* Herr Baer leitet uns einen Ansprechpartner weiter, sodass wir möglichst reale Ladungsträger (ggf. inkl. QR-Codes) erhalten - dauert wohl 2-4 Wochen
* <https://item.engineering/DEde/tools/engineeringtool>

Lt. Herr Baer: in 2 Stufen arbeiten

* **Kamerapositionierung** ist uns überlassen, aber Herr Baer würde sie mittig im Raum an die Decke platzieren

Stufe1:

* möglichst **wenige Kameras**
* evtl. zu Beginn sogar mit einer **Handycam** beginnen

Stufe2:

* Räume (Stationen entsprechen realer Länge)
  + eine Station hat z.B. **10x20m** und ist mind. **10 m** **hoch**
* lieber mehrere günstige Kameras nutzen, als eine teure

Sonstiges:

* **QR-Codes selbst** generieren, bzw. von **item.engineering** nutzen
* Datenschutz:
  + Kopferkennung und automatisches Zensieren der Gesichter (optional)
  + Aktualisierung außerhalb der Arbeitszeiten (widerspricht der “nahezu” Echtzeit)

**To-Do:**

- Scrum-Team Positionen/Aufteilung festlegen:

* Ansprechpartner für Evobus wählen und Kontaktdaten an Herrn Baer/Herrn Gaisbauer weiterleiten
* Scrum Master: Viktor
* Product Owner: Felix, Basti
* Development Team: Florian, Niko, Fabi, Moritz

- Lunde zu Github Repository einladen

- Projektnamen bestimmen: CarrierTracking

**Besprechung 04.11.2020**

Rechteverwaltung:

* kein Rechtesystem notwendig
* funktionierendes system definitiv wichtiger dafür SW erweiterbar halten
* keine direkte Anforderung von Kunden

Schnittstelle Kamera/SW:

* Kamera alle 5 Min ein Bild → Standorte werden ausgewertet
* Lokal auf einem Gerät oder Netzwerk
  + aktuell nichts vorhanden → wir können etwas vorschlagen
  + wir legen Lokal an und speichern bilder darauf

* Kamera
  + Auflösung von 4K sollte Voraussetzung sein
  + Szenario unserer HW entsprechend setzen
  + Deckenhöhe “selber setzen”
  + Anfrage an Herrn Lunde stellen: Kamera raussuchen und er kümmert sich um Finanzierung

* Ladungsträger
  + Wie viele gleichzeitig in einer Station?
    - nicht über 10 (2-5 klingt realistisch, wenn wir das hinbekommen “schon klasse”)
    - QR Code Anpassungen richten sich nach uns, wenn wir vorgeben wir brauchen eine bestimmte größe, versuchen diese die umzusetzen
    - Bsp von Ladungsträgern?

Grundsystem + Infrastruktur erstmal zum Laufen bringen und Empfehlungen geben, bevor wir uns in Kleinigkeiten verlieren → Wenn es produktiv wird spielen sowieso noch mehr Faktoren eine Rolle

Feedback zum Product Backlog:

* 2D Plan nicht von Item Engineering
* Import von verschiedenen Layouts sollte nicht im Fokus stehen sonder nicht auf Raumplanung → erstes Bild wird Layout der Station (=> macht keinen Sinn, zum zeichnen braucht man den ganzen Raum, die Kamera deckt ja nur einen Bereich ab)
* Alternative: Wir bekommen 2D Tool und zeichnen dieses einmalig manuell und später kann dieser von einem Tool kommen → “Problem” umgehen
* Feedback: Viel Gedanken, wenn nur ein Bruchteil der User Stories funktioniert schon sehr gut → wird wohl nicht alles umsetzbar sein
* User Stories sehr gut gemacht

Feedback Lunde:

Product Backlog ist das beste was heute präsentiert wurde

Vision ist sehr klar geworden

Experimente sehr gut gewählt, Wo sind Herausforderungen, was brauch ich? Bib Auswahl, alle Fragen halt

Aufgabenbereiche gut gewählt

Nächstes mal erst Produkt präsentieren, Überblick geben und dann die Rückfragen stellen

Alle sehr gut vorgestellt

Feedback Gaisbauer:

User Stories wirklich gut, Als Kunde gibt es nichts besseres (s.o.)

Pragmatischer an manche Sachen rangehen. Erst Grundgerüst und dann Feinheiten (Blackboxen nutzen) → zu viele Anforderungen, bewusst Abstriche zu machen um Kernfunktion priorisieren

Feedback Kaiser:

Produkt Backlog sehr gut

Zu viele Details

Priorisierung gut gewählt und diese auch so abarbeiten

**Besprechung 18.11.2020**

Vorstellung der neuen Funktionen

Vorstellung der nächsten Schritte

Frage: Ob das auch real vorgestellt.

1. alles vor Ort anschauen
2. Bausatz für Modelle zur Verfügung stellen, rechtzeitig Bescheid geben
   1. erstmal mit den Boxen soweit Produkt fertigstellen
   2. danach dann Bausatz für Ladungsträger
   3. wenn dann spätestens zum nächsten Sprint
      1. Anfang Dez → sonst extrem eng
      2. AP für itemengeneering in Ulm

Offset Definieren, wo der QR sich auf dem Ladungsträger befindet

Auch mit Attrappen arbeiten und dann mit realen Ladungsträger (Drehung)

Einzelteile sehr gut, wie wird das im Endprodukt aussehen?

Projekt öffnen und Menü (UI), am anfang keine  Stationen

neue Stationen anlegen

Programmablauf definieren

In 2 Wochen ist Meilenstein. Wie spielt alles zusammen? In 2 Wochen “Prototyp” abliefern.

Menüführung, alles zusammenfügen, Stationen erstellen und dann mit Foto(ob selbst aufgenommen oder simuliert) → Proof of Concept

Alles fortführen wäre Ideallösung für 03.12.20

Meeting Gestaltung: Demos zeigen ist sehr gut, In Zukunft als Überlegung bevor man in die Demos einsteigt. Kurze Folien mit Einleitung, worum gehts im projekt allgemein dann das Sprint Ziel der aktuellen Woche und dann Demos.

Erst allgemein vorstellen und dann in die Fachbereiche

Dann Endfolie das und das haben wir erreicht und das ist unsere Ziel für nächst Woche. Feedback von Kunden einholen und dort restlichen Fragen klären

Produktname geben und Marketingaspekte

Starten mit was war die Aufgabenstellung, zu schnell in die Details

Meilenstein!

**Ideen:**

*Idee - wie von Grp. vor uns vorgetragen:*

*Stationen wie in bspw. Excel jeweils in eigenen Tabs verwalten, sowie ein Tab mit einem Eingabefeld, sodass man direkt zu z.B. Station 17 springen kann.*

