09.12.2019 // Vorbereitung Jit.si-Konferenz am 09.12.19

https://meet.jit.si/oas

[OAS]

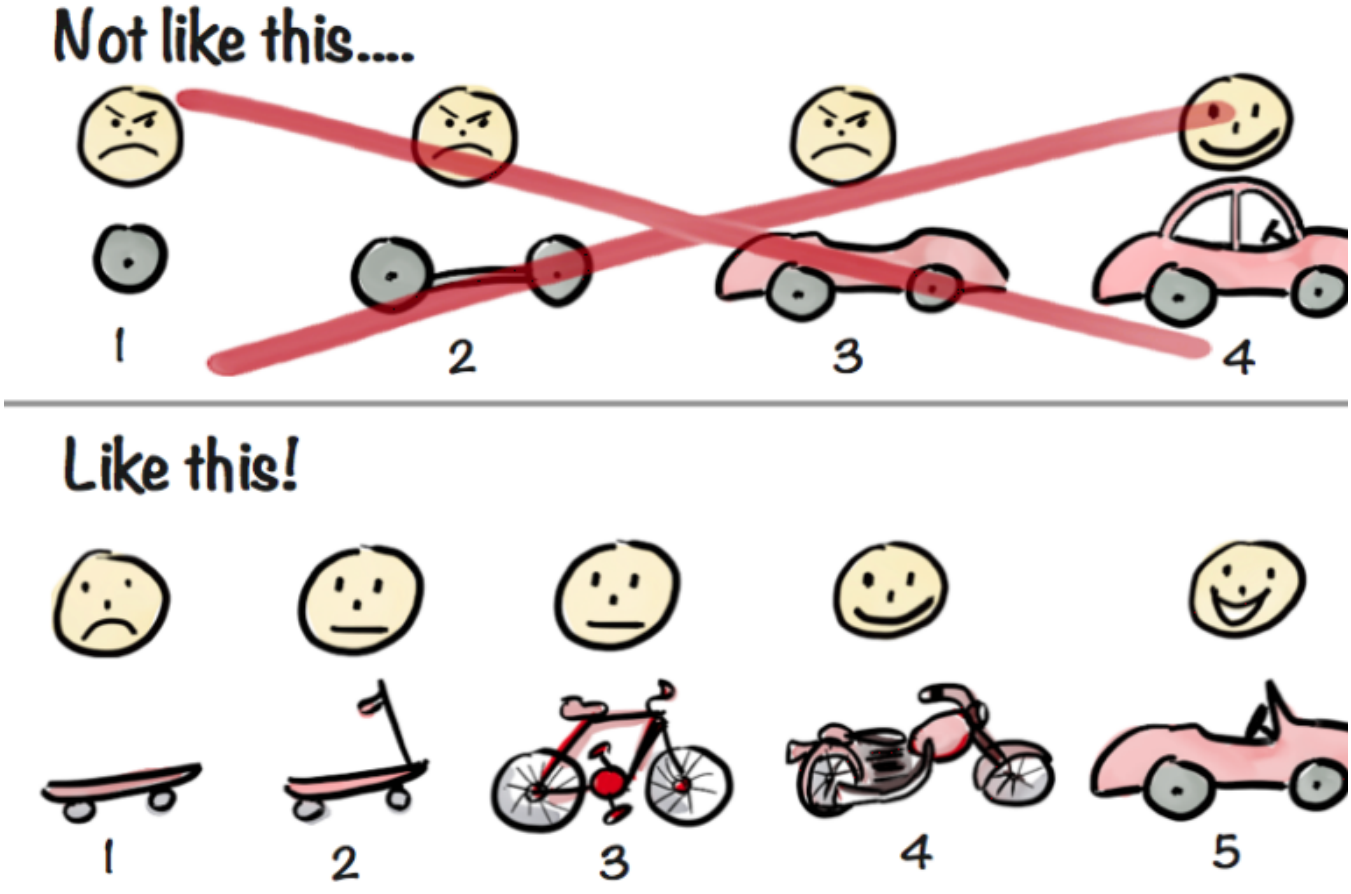
[Roseguarden]

[BF2H]

Agenda

Fragen?

- Wie ändert sich die Deadline / wie organisieren wir uns, damit Fabian ins Team zurück kann ohne seine Deadline zu gefährden? (falls Fabian das möchte.)
- Wollen wir ein MLP/MAP (Minimum Loveable/Awesome Product) definieren? Idee: schnell etwas auf die Beine stellen, um Feedback und neue Entwickler zu bekommen
  -



(siehe: https://de.wikipedia.org/wiki/Minimum_Viable_Product#cite_note-6)

Konsens?

☐ Für MakerSpaces außerhalb DE muss das FrontEnd (&BackEnd?) generell multilingual gedacht werden (i18l).

- ...

Technologiefragen (hinzugefügt von mdrobisch)

Entscheidungsfindung

- gut geklärter Entscheidungsprozess ist sehr wichtig (vorher)
- besonders bei technischen Fragestellungen ist Kompromissfindung meist schwierig und führt zu schlechteren Entscheidungen (persönliche Erfahrung)
- offtmals (persönliche Erfahrung) sind Diskussionen über Werkzeuge schlussendlich nicht wichtig, da mit verschiedenen Werkzeugen gleiche Ziele erreicht werden können
- wichtig ist der Wiederstand einzelner Personen gegenüber getroffenen Entscheidungen besonders auf lange Sicht im Auge zu behalten
- Entscheidungskriterien sollten zusammen erstellt werden, (persönliche Erfahrung: führt zu langen teilweise dogmatischen Diskussion -> macht keinen Spaß)
- danach abstimmen (in kleiner Gruppe) oder von einem Leiter (meist der am meisten Energie aufbringt)

Hauptprogrammiersprache im Backend

- Können/Wollen wir das gerade Entscheiden?
- Rust
- Python

Vorschlag Entscheidungskriterien zur Auswahl der Programmiersprache:

Rangfolge entspricht Priorität (1. Sehr Wichtig ... n+1. nicht wichtig)

1. Zu Verfügung stehende Manpower (im Moment), d.h. kurzfristig zu erwartender Geschwindigkeit der Entwicklung

2. Zu erwartende Manpower/Entwickler, d.h. langfristige Entwicklung / Bestand des Projektes / Geschwindigkeit der Entwicklung
3. Niederschwelligkeit für Einsteiger und nicht-Entwickler (Doku, Entwicklungsumgebungen)
4. Zur Verfügung stehende relevante Bibliotheken
5. Bestehende Projekte auf denen aufgebaut werden kann d.h. starten wir am Anfang oder in der Mitte und gibt es Hilfestellung beim einrichten Programmieren
6. Technische Feature der Sprache (ggf. mit Mindestschwälle, Turing-Vollständig)

Bitte weitere ergänzen oder Prio ändern

Schnittstellen

- Können/Wollen wir das gerade Entscheiden?
- P2P (Point-to-Point, MQTT, ZMQ, etc.) über z.B. Broker-Infrastruktur
- HTTPS

Vorschlag Entscheidungskriterien zur Auswahl der Hauptschnittstelle:

Rangfolge entspricht Priorität (1. Sehr Wichtig ... n+1. nicht wichtig)

1. Erfahrung mit der Schnittstelle (im Moment)
2. Interne Integrierbarkeit (nach innen, intern zwischen Komponenten des System bzw. bestehende Systeme, z.B. Bibliotheken, Bestandslösungen)
3. Client/Hardware Integrierbarkeit (möglichkeit/aufwand Mikrcontroller und Clients)
4. Externe Integrierbarkeit (nach außen, für externe/andere Systeme)
5. Niederschweligkeit für Einsteiger (Doku, Verbreitung, Infrastruktur)
6. Technische Feature der Schnittstelle

Bitte weitere ergänzen oder Prio ändern

Frontend

- besteht hier schon Konsens?

Angebundene Geräte (Clients/Mikrocontroller)

- wollen wir das überhaupt mit in die Entwicklungsziele aufnehmen?

Fragen zu Rust

- @Gruppe/Devs: Wer hat bereits Erfahrung damit? (gleiche Frage für Python)
- @Gruppe/Devs: Wer könnte sich vorstellen sich darin einzuarbeiten? (gleiche Frage für Python)
- @Gruppe/Devs: Wer kann sich vorstellen Hauptentwickler zu sein (Thematik: Basar-Stil vs Lead-Developers)
- Sind technische Aspekte der Programmiersprache für unsere Entscheidungin der Gruppe überhaupt wichtig? (Exceptions, Typsicherheit, Lineartät, Multitthreading)
- Wie stellt sich ein Kosten/Nutzen-Verhältnis konkret dar? Welche Prios sind dort angenommen?
- Müssten wir bei 0 Anfangen oder gibt es schon einen Prototypen?
- welche Bibliotheken würden wir für konkret nutzen?
- ist ein Plugin-System in Rust schnell umsetzbar? (auch möglich ohne andere Programmier/Skriptsprache)

Fragen zum P2P-Ansatz

- Welche Probleme löst P2P?
- Hat schon jemand Erfahrung mit MQTT, ZMQ, etc. an konkreten produktiven Projekten
- Hat schon jemand Erfahrung mit Websockify in Frontends? (ist das exotisch oder standard)?
- Wollen wir Sicherheitsthemen/Verschlüsselung auf Programmierebene wirklich selber machen? (besonders auf Arduino, ESP und Co), bzw. können wir das überhaupt?
- Wie binden wir mit P2P externe Services/Komponenten (Odoo, Groupware, Hardware) an? Über ein selbst programmiertes Gateway/Protokollwandler?

Version #3
Erstellt: 2024-10-13 01:04:55 CEST von Mario Voigt (Stadtfabrikanten e.V.)
Zuletzt aktualisiert: 2025-02-25 21:22:13 CET von Mario Voigt (Stadtfabrikanten e.V.)