

## Übungsgruppe: Qianli Wang und Nazar Sopiha

### **dAufgabe 2-1:**

- a) 1. **Problemstellungen:** die Frage beantworten: "Mit welchen Phänomenen muss die Softwaretechnik fertig werden?"
2. **Lösungsansätze:** definieren einen Lösungsweg zu den gestellten Problemen. Unter anderem: Architektur, Erweiterungsmöglichkeiten, Prinzipien, Technologien
3. **Organisation des Arbeitsprozesses:** Das Team soll die gestellten Aufgaben effizient lösen, dafür braucht es gutes Hardware, essen, Teambuilding, Konflikte auflösen u.s.w....
4. (idealerweise) **Erstellung des fehlerfreien Produkts(Softwaretest):** Während und nach der Entwicklung der Produkte soll man die genügend testen und Bugs töten.
5. **Qualitätsplanung:** Identifizierung der für Projekt relevanten Qualitätskriterien.
6. **Flexibilität:** Während des Entwurfs des Softwares soll das Produkt immer aktuelle Anforderungen erfüllen, auch die Dokumentation soll rechtzeitig aktualisiert werden und falls nötig geändert werden. (z.B. Agile software development mit Sprints)
7. **Wartung/Pflege:** Etwas wie Support Line, Updates, Weiterentwicklung

(Quelle: <https://de.wikipedia.org/wiki/Softwaretechnik>)

- b) **Anforderungen:** Problemstellungen, Lösungsansätze  
**Entwurf/Implementierung:** Lösungsansätze, Flexibilität, Wartung/Pflege  
**Qualitätssicherung:** Erstellung des fehlerfreien Produkts, Qualitätsplanung, Wartung/Pflege  
**Management:** Organisation des Arbeitsprozesses, Flexibilität

### Aufgabe 2-2:

#### **Projektname: LOST & FOUND**

**Motivation(Stammung der Idee):** Letztes mal, als ich das Flugzeug genommen habe, habe ich leider mein Jacker am Bord vergessen. Danach habe ich nur ein SMS von Aircraft bekommt mit einer ID Nummer von meinem Gegenstand. Ich soll dann selbst wieder nach Flughafen fahren, um es abzuholen, was ich sehr unpraktisch finde. Deswegen haben wir uns dafür entschieden, ein App dafür zu entwickeln, was auch in vielen Bereichen angewendet werden kann, wie z.B. bei BVG und Tegel.

**Anwendungsdomäne:** Man kann Lost&Found überall, wo die Dingen vergessen werden können, benutzen. Als ersten Kundenbereich nehmen wir Flughäfen.

**Arbeitsprozess:** Lost&Found soll helfen, vergessene Sachen wiederum zu finden und dann Sachen, die nach bestimmter Zeit nicht abgeholt werden, verteilen.

**Funktionen:** Alle Dinge, die vergessen werden, werden in eine Datenbank hinzugefügt mit möglichst vielen Details, wie zum Beispiel **Marke, Farbe, Größe, ggf.**

**Gegenstände(Kleinigkeiten) in Taschen** (z.B. in der Jacke oder im Rucksack), **Foto, ...**  
Man sollte erstmal in App melden und möglicherweise viel Informationen über sein verlorene Gegenstände anbieten, damit wir besser sie mit der Informationen in der Datenbank vergleichen können.

Dann können wir ins App eine Meldung hinzufügen: z.B. Wartezimmer, 2. Reihe, links - Jacke vergessen. Der Benutzer, der diese Jacke vergessen hat (oder so denkt), soll möglichst viele Details vom Ding bereitstellen (event. auch Foto mit diesem Ding). Im Fall von Flughafen, soll der auch das Ticket vorlegen, um nachzuweisen, dass er an diesem (+- 1) Tag im Wartezimmer sein könnte. Also nach bestimmten Schritten zur Sicherung stellen wir fest, ob wir das verlorene Ding in unserem Datenbank haben. (Benutzer, die sich ganz oft und nicht erfolgreich melden, werden geprüft und eventuell blockiert)

Nach 30 Tagen werden nicht abgeholte Dinge für andere Benutzer freigestellt. Die können fürs Minimalpreis verkauft werden oder einfach so gespendet werden (wird noch überlegt, weil es wahrscheinlich mit dem Recht zu tun ist). Benutzer können "freie" Dinge reservieren und später selbst abholen. Oder das Ganze kann sich an (z.B.) Waisenhaus richten.

**Verhältnis:** Die Benutzer sollen kein Geld fürs App bezahlen. Es ist als nicht profitabel für Kunden gedacht.

**Kunde:** App wird vom z.B. Flughafen, BVG oder eventuell Regierung unterstützt.

**Zielgruppe:** Leute, die im öffentlichen Ort irgendwas vergessen.

**Wissensstand:** Benutzer - Minimal, alles wird nativ. Mitarbeiter (die die Datenbank erweitern) auch minimal, denn alle technischen Hintergründe (wie Ding hinzufügen, Ding verteilen) werden durch User Interface realisiert/versteckt.

**Probleme:** Identifizierung des Produkts, Nachweis der Zugehörigkeit