

## Anforderungserhebung

### Übersicht:

- IBuchungsbefugt
  - buchen(Skischule sh)
  - stornieren()
    - 24h vor Kurs möglich
- Teilnehmer : implements IBuchungsbefugt
  - m\_alter
  - m\_schuhgr
  - m\_hotel
- Hotel : implements IBuchungsbefugt
  - Teilnehmer m\_teilnehmer[]
- Reiseveranstalter : implements IBuchungsbefugt
  - Teilnehmer m\_teilnehmer[]
- Kurs
  - Lehrer m\_Lehrer
  - enum status{ OKAY, AUSFALL, ZUSAMMENGELEGT }
  - Kurs zusammengelegt;
  - lehrerBuchen(Lehrer l)
  - lehrerZahlen()
- Skischule
  - Teilnehmer teilnehmer[]
  - Kurs kurse[]
  - Lehrer lehrer[]
  - skiAusrüstungLeihen()
  - skirennen[]
- Skirennen
  - Gemeindeverwaltung m\_verwaltung
  - private bool rennstreckeFreigeben()
  - private bool versorgungBereitstellen()
  - private bool bergwachtInformieren()
  - public bool vorbereiten()
- skilehrer sollen immer zur verfügung stehen
- kurse sollen bestmöglich ausgelastet sein

- skischuhe in gewünschter gröÙe müssen kurzfristig beschaffen werden können
- unterschiedliche Preise für Dauer und Altersgruppe

### **Funktionale Anforderungen**

- Teilnehmer sollen Kurse buchen können
- Hotels sollen Kurse buchen können
- Reiseveranstalter sollen Kurse buchen können
- Skischule soll rennen organisieren können
- Skilehrer sollen für Kurse gebucht werden können

### **Qualitätsanforderungen**

- Reaktionszeit pro Aktion sollte unter 5s liegen
- System darf maximal eine Stunde in der Woche ausfallen
- darf nie falsche Buchungen durchführen. Lieber scheitern lassen

### **Technische Anforderungen**

- muss auf systemen vor ort laufen
- auch auf kundensystem bedienbar (cross platform)

### **Benutzungsschnittstellen-Anforderungen**

- buchung mit max 2 klicks findbar
- nur eine Aktion vom Nutzer um zu buchen
- zuständigkeit eindeutig

### **Prozessanforderungen**

- Kurse sollen automatisch storniert und zusammengelegt werden falls:
  - lehrer ausfällt / nicht verfügbar
  - Kurs nicht optimal belegt
- Teilnehmer sollen frühzeitig informiert werden wenn Ausrüstung nicht verfügbar und es soll ein alternativer Kurs angeboten werden.

### **rechtlich-vertragliche Anforderungen**

- Teilnehmer haftungsausschluss vor buchung bestätigen lassen
- Versicherung anbieten / auskunft geben
- gemeindeverwaltung muss informiert werden wenn ein rennen geplant ist
  - auf bestätigung warten
- Bergwacht muss über rennen informiert werden

### **sonstige Anforderungen**

- automatische umbuchung von Teilnehmern um Kursauslastung und Ausrüstungsausnutzung zu optimieren.
- über web interface erreichbar

### **Anforderungen bewerten**

**Gerd Geldsack, Einkauf**

- 10% schneller - optionales Ziel, mittlere Priorität. Jedoch vermutlich "einfach" umzusetzen und Gewinn versprechend.
- Laser stärker fokussieren / 30m dicke Bleche Bohrung - höhere Priorität da Funktionalität fehlt die der Kunde braucht.

**Hans Müller, Arbeiter am Band**

- Nach Sicherheitsausschaltung funktioniert das Gerät zeitweise nicht - primäres Ziel, höhere Priorität, Sicherheit und Funktionalität sollten primäre Ziele sein, das hört sich nach einem Bug an der die Sicherheit der Arbeiter gefährden könnte und ist daher ein primäres Ziel, obwohl nicht das gewinnbringeste.

**Sabine Sachte, Arbeiterin am Band:**

- allgemeine Benutzerfreundlichkeit.
  - Schrift anpassbar machen - mittlere Priorität
  - übersichtlicheres menu - mittlere Priorität
  - "bestätigung mit zusammenfassung" - könnte Maßnahme gegen Fehler sein. geht evtl. auf Kosten der Produktivität. - optional

**Brigitte Achtsam, Qualitätsmanagerin**

- weniger Abweichung - primäres Ziel, sonst evtl. Verdrängung aus dem Markt durch japanische Konkurrenz

**Volker Vorwurf, Sicherheitsingenieur**

- Werkzeugwechsler hat zu viel Spiel zwischen 2 Scheiben: primäres Ziel, da evtl. verantwortlich für die Abweichung und ein Sicherheitsrisiko für Arbeiter.

**Benutzerschnittstelle Fahrkartenautomat**

[IMAGE: Wireframe.png]