



UNIVERSITÄT  
LEIPZIG

Wintersemester 2018/19  
Softwaretechnikpraktikum

---

# App zur Inventarisierung von Unternehmenswerten

## Risikoanalyse

---

Gruppe: ak18b

Betreuer: Benjamin Lucas Friedland, Michael Fritz

Gruppenmitglieder: Alexander Zwisler, Leon Kamuf, Leon Rudolph, Maurice Eisenblätter,  
Maximilian Gläfcke, Robin Seidel, Sina Opitz, Steve Woywod

# Inhaltsverzeichnis

1	Unterschiedliche Wissensstände . . . . .	1
2	Ausfall von Teammitgliedern . . . . .	1
3	Änderungen durch Stakeholder am Konzept . . . . .	1
4	Zeitmängel . . . . .	1
5	Fehlerhafte Kommunikation . . . . .	2
6	Technische Probleme . . . . .	2
7	Interne Konflikte . . . . .	2
8	Komplexität von Arbeitspaketen wird unterschätzt . . . . .	2
9	Releases durch Breaking Changes verzögert . . . . .	3
10	Unklare Anforderungen . . . . .	3

## 1 Unterschiedliche Wissensstände

**Problem:** Nicht alle Personen haben den selben Wissensstand.

**Ursachen:** Unterschiedliche Personen durchlaufen unterschiedliche Etappen in ihren Leben (z.B. verschiedene Lehrveranstaltungen, Freizeitaktivitäten etc.), wodurch Wissensdifferenzen zwischen den Gruppenmitgliedern entstehen.

**Folgen:** Erfahrenere Gruppenmitglieder bekommen zu viel Arbeit zugewiesen oder Unerfahrenere wissen nicht, wie sie ihre Aufgaben bearbeiten sollen.

**Lösung:** Ein möglicher Lösungsansatz wäre Codesprints in Gruppen zu organisieren, damit man bei Bedarf schnelle Hilfe von erfahrenen Gruppenmitgliedern erhalten kann. Des Weiteren sollte man Codereviews nach jedem Merge einplanen, um Programmierfehler früh zu erkennen und die Qualität zu garantieren. Es ist ebenfalls besonders wichtig die Rollen innerhalb des Projekts, anhand der Vorkenntnisse des jeweiligen Gruppenmitgliedes, einzuteilen.

## 2 Ausfall von Teammitgliedern

**Problem:** Es kann jederzeit dazu kommen, dass Gruppenmitglieder ausfallen.

**Ursachen:** Durch projekt-externe Gründe (Krankheit, persönliche Gründe, Studienabbruch) oder projekt-interne Gründe (Konflikte innerhalb der Gruppe, Überforderung mit zugeteilten Aufgaben) kann dieses Problem auftreten.

**Folgen:** Wenn Gruppenmitglieder ausfallen, werden deren übernommene Aufgaben zwangsläufig auf die übrigen Gruppenmitglieder verteilt. Dies hat eine erhöhte Belastung der verbleibenden Gruppenmitgliedern zur Folge.

**Lösung:** Es ist wichtig sich genug Pufferzeiten einzuplanen, damit man gegebenenfalls noch umstrukturieren kann und zeitliche Verzögerungen vermeidet. Zudem sollten sich betroffene Gruppenmitglieder zeitnah beim Projektleiter melden, damit der Ausfall so gut wie möglich kompensiert werden kann.

## 3 Änderungen durch Stakeholder am Konzept

**Problem:** Konzeptionelle Änderungen können schnell die Planung durcheinander bringen.

**Ursachen:** Änderungsvorschläge seitens des Kunden oder Umsetzungsversuche funktionieren nicht wie geplant.

**Folgen:** Zeitliche Verzögerung bei der Fertigstellung von Arbeitspaketen und Probleme bei der Adaption von neuen Vorgaben.

**Lösung:** Möglichst zeitnahe Absprachen, im gesamten Team, über diese Änderungen und Änderungsvorschläge.

## 4 Zeitmängel

**Problem:** Zeitmangel bei der Fertigstellung von Arbeitspaketen oder Verzögerung der Release.

**Ursachen:** Diskontinuierliches Arbeiten und schlechtes Zeitmanagement sind die Hauptursache für dieses Problem.

**Folgen:** Wie bereits erwähnt folgen aus diesem Problem verzögerte Abgaben und Unzufriedenheit des/der Kunden.

**Lösung:** Ein gutes Zeitmanagement ist essentiell um dieses Problem zu vermeiden. Sollte es dennoch zu Zeitmängeln kommen, könnte man sich nur auf die wichtigen Requirements konzentrieren und mögliche Features, welche nicht zwangsläufig notwendig sind, erstmal ignorieren.

## 5 Fehlerhafte Kommunikation

**Problem:** Es werden falsche, beziehungsweise nicht gänzlich korrekte Informationen weitergeleitet.

**Ursachen:** Mitteilungen werden über Mittelsmänner weitergeleitet oder sie werden nicht richtig verstanden.

**Folgen:** Daraus resultieren zum Beispiel unerwünschte Programmfeatures, Codekonflikte beim Mergen oder Nichtbearbeitung von Aufgaben.

**Lösung:** Entgegenwirkend ist die Einrichtung einheitlicher Kommunikationswege (GitLab, WhatsApp, Discord, etc.) und persönlicher Meetings im gesamten Team, um Klarheit in die gestellten Aufgaben zu bringen. Außerdem sollte immer ein ausreichend ausführliches Protokoll des Meetings erstellt werden, um wichtige Punkte nochmal nachlesen zu können.

## 6 Technische Probleme

**Problem:** Technische Probleme können beim Bearbeiten, Herunterladen oder Hochladen von Dateien/Code geschehen.

**Ursachen:** Die Hauptursachen sind nicht einheitliche Tools zum Bearbeiten der Aufgaben oder ein technischer Defekt bei Internetseiten (GitLab).

**Folgen:** Mögliche Folgen sind der Verlust der bisherigen Arbeitsergebnisse oder die Unbrauchbarkeit von Selbigen.

**Lösung:** Man sollte sich so schnell wie möglich auf Standardsoftware bei der Bearbeitung von Aufgaben einigen und sich möglichst zeitnah mit dieser vertraut machen. Für den Fall von technischen Ausfällen muss man sich Backups erstellen, um dem Verlust von Arbeitsabschnitten, oder der ganzen Arbeit, vorzubeugen.

## 7 Interne Konflikte

**Problem:** Es können interne Konflikte bei Meinungsverschiedenheiten im Team entstehen.

**Ursachen:** Unterschiedliche Lösungsvorschläge für die Lösung projekt-interner Probleme oder andere Vorstellungen über das Endergebnis sind Ursachen wodurch diese Konflikte entstehen können.

**Folgen:** Es kommt zu einer schlechten Stimmung im Team und zur Unzufriedenheit von Gruppenmitgliedern.

**Lösung:** Es wird die Regelung getroffen, dass man sich an den Projektleiter wendet, welcher als Richter fungiert und entscheidet, welcher Lösungsansatz gewählt wird.

## 8 Komplexität von Arbeitspaketen wird unterschätzt

**Problem:** Das Problem sind nicht fristgerechte Fertigstellungen von Arbeitspaketen.

**Ursachen:** Indem man sich nicht mit der Komplexität, der Arbeitspakete, vertraut macht, kann es zu diesem Problem führen.

**Folgen:** Dies hat, Abgabe- oder Releaseverzögerungen, zur Folge.

**Lösung:** Es muss eine klare zeitliche Planung von Arbeitspaketen und eine genaue Analyse der Aufgabenstellung erfolgen. Bei offenen Fragen sollte man sich an den Projektleiter, die Betreuer oder die Tutoren so schnell wie möglich wenden.

## 9 Releases durch Breaking Changes verzögert

**Problem:** Zeitliche Verzögerung der geforderten Anforderungen.

**Ursachen:** Mögliche Ursachen sind Breaking Changes, welche commitet werden und nicht so schnell identifizierbar sind.

**Folgen:** Es könnte abermals zu verzögerten Abgaben kommen und die zu investierende Zeit steigt rapide.

**Lösung:** Die Lösung ist Continuous Deployment/Delivery, so dass entweder in einem zeitlich festgelegten Rahmen (Nightly-Builds) oder durch festgesetzte Repository-Events (z.B. Merges) der Build-Prozess angestoßen und deployed wird.

## 10 Unklare Anforderungen

**Problem:** Anforderungen, welche an des Programm gestellt werden, werden nicht komplett verstanden.

**Ursachen:** Ursachen für dieses Problem könnten nicht ausreichende Teamkommunikation zu den Projektvorgaben oder missverstandene Anforderungen an das Projekt sein. Darunter zählen nicht, im vollem Maße, geklärte Projektideen oder das Übersehen von wichtigen Details.

**Folgen:** Es folgt das Fehlen von wichtigen Features im fertigen Programm und die Unzufriedenheit des Kunden.

**Lösung:** Für die Lösung des Problems ist eine ausführliche Planung des Projekts Grundvoraussetzung. Es sollte zudem in den Teammeetings alle Schlüsselfragen geklärt werden. Prototypen erstellen, welche dem Kunden präsentiert werden, können dazu beitragen falsch verstandene Kriterien auszubessern.