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 Risikoanalyse

1.1 Unterschiedliche Wissensstände

Das Problem besteht darin, dass Personen unterschiedliche Vorkenntnisse aufweisen. Diese Tatsache ist darauf zurückzuführen, dass nicht jeder die selben Aktivitäten in seinem Leben durchlaufen ist und somit Wissensdifferenzen bestehen. Um diesem Fakt vorzubeugen, könnte man Codesprints in Gruppen organisieren, damit man bei Bedarf schnelle Hilfe von Erfahreneren Gruppenmitgliedern erhalten kann. Des Weiteren kann man Codereviews in festen Zeitabständen planen, um Programmierfehler früh zu erkennen und die Qualität zu garantieren.

1.2 Ausfall von Teammitgliedern

Da es immer sein kann, dass Gruppenmitglieder Ausfallen (bedingt durch Krankheit, persönlichen Gründen oder aber durch Studienabbruch), muss man sich Gedanken für diese Fälle machen. Mögliche Gegenmaßnamen sind genügend Pufferzeiten, um zeitliche Verzögerungen zu vermeiden. Außerdem ist es überaus wichtig, dass sich betroffene Mitglieder unverzüglich beim Projektleiter melden, damit eine potenzielle Umstrukturierung der Planung stattfinden kann.

1.3 Änderungen durch Stakeholder am Konzept

Konzeptionelle Änderungen können schnell außer Kontrolle geraten, jedoch kann man dem relativ einfach vorbeugen. Essenziell ist die sofortige Absprache im gesamten Team über diese Änderungen und zeitnahe Absprache über die Änderungsvorschläge. Zudem sollte man sich Iterationen für Arbeitspakete einplanen.

1.4 Zeitmängel

Zeitliche Probleme können schnell durch nicht kontinuierliches Arbeiten und ein schlechtes Zeitmanagement hervorgerufen werden. Der logische Schluss daraus ist Selbstreflexion und Ändern dieses Verhaltens, um dieses Problem zu vermeiden. Sollte es dennoch zu Zeitmängel kommen könnte man sich nur auf die essenziellen Requirements konzentrieren und mögliche Features, welche nicht zwangsläufig notwendig sind, fürs Erste ignorieren.

1.5 Fehlerhafte Kommunikation

Misskommunikationen können schnell geschehen. Besonders, wenn wichtige Mitteilungen über einen Mittelsmann weitergeleitet werden. Um dieser Problematik entgegenzuwirken ist es wichtig einheitliche Kommunikationswege einzurichten, zum Beispiel GitLab, WhatsApp oder Discord. Zudem ist ein wöchentliches persönliches Meeting empfehlenswert.

1.6 Technische Probleme

Technische Probleme ist ein Themenfeld, welches von der Wahl der zu benutzenden Software bis zum Ausfall von wichtigen Tools, wie zum Beispiel GitLab, reicht. Folglich ist es wichtig, sich am Anfang auf Standardsoftware zu einigen und sich mit dieser vertraut zu machen. Für technische Ausfälle sollten Backups erstellt werden, damit kein Teil, oder gar die ganze Arbeit, verloren geht.

1.7 Interne Konflikte

Durch unterschiedliche Vorstellungen über die Lösung eines Problems kann es zu internen Konflikten im Team kommen. Um dies weitmöglichst zu umgehen wird festgelegt, dass man sich an den Projektleiter wendet, welcher als Richter fungiert und entscheidet, welcher Lösungsansatz gewählt wird.

1.8 Komplexität von Arbeitspaketen wird unterschätzt

Durch Überschätzung einzelner Gruppenmitglieder kann es zu zeitlichen Verzögerungen kommen, wie zum Beispiel durch nicht Fertigstellung eines Arbeitsabschnittes in der vorgegebenen Zeit. In dem man sich jedes Arbeitspaket zeitlich klar plant und die Aufgabenstellung analysiert, kann dem entgegengewirkt werden.

1.9 Releases durch Breaking Changes verzögert

Der Release kann nicht fristgerecht veröffentlicht werden, falls Breaking Changes commited werden, welche nicht schnell genug identifiziert werden können. Die Lösung ist Continuous Deployment/Delivery, so dass auf einem festgelegten zeitlichen Rahmen (täglich, wöchentlich, ...) die Release-Builds auf dem Master-Branch gepusht werden, um die potentiellen Fehlerquellen so minimal wie möglich zu halten.

1.10 Unklare Anforderungen

Missverstandene Anforderungen, welche an das Projekt gestellt werden, können zu starken Abweichungen gegenüber den ursprünglichen Vorstellungen des Auftragsgebers führen. Diese Abweichungen können, zum Beispiel durch nicht, im vollem Maße, geklärten Projektideen oder durch das Übersehen von wichtigen Details, entstehen. Um dem vorzubeugen ist eine ausführliche Planung des Projekts Grundvoraussetzung, sowie das stellen aller Schlüsselfragen. Außerdem sollte man auch Prototypen erstellen, welche dem Kunden präsentiert werden, sodass falsch verstandene Kriterien noch ausgebessert werden können.