

# 1 Risikoanalyse

*Manchmal muss man einfach ein Risiko eingehen - und seine Fehler unterwegs korrigieren.  
– Lee Iacocca*

Wesentliche Voraussetzungen für das Gelingen und den Erfolg eines Projektes sind das Identifizieren von Risikofaktoren und das geeignete Festlegen ihrer Gegenmaßnahmen. Dadurch wird die Komplexität der zu lösenden Aufgaben übersichtlicher und das Auftreten von Fehlern, die das Projekt gefährden können, wird reduziert.

Die zehn für unser Projekt wichtigsten Risikofaktoren und entsprechende Lösungsvorschläge werden im Folgenden vorgestellt. (Die einzelnen Punkte der Auflistung sind nicht gewichtet.)

## 1.1 Erstes großes Softwareprojekt, mangelnde Erfahrung und damit verbundene falsche Rollenverteilung

Für die meisten Mitglieder unseres Teams ist dies das erste größere Software-Projekt. Somit fehlt es einigen Teammitgliedern an ausreichender Erfahrung hinsichtlich der Durchführung und Realisierung eines solchen Projekts. Weiterhin gibt es innerhalb des Teams unterschiedliche Fachkenntnisse bezüglich der Anwendung von Prinzipien und Methoden der Softwaretechnik und bei der Verwendung von Werkzeugen, wie zum Beispiel Entwicklungsumgebungen. Somit können Fehleinschätzungen der eigenen bzw. fremden Fähigkeiten entstehen. Dies kann zu einer falschen Rollenzuteilung und zu einer Überforderung eines oder mehrerer Teilnehmer führen.

*Prävention: Eine ausführliche Besprechung des gesamten Projektes ist zu Beginn erforderlich, um spätere Probleme zu minimieren. Erfahrungen werden ausgetauscht und Wünsche für das Projekt angegeben, damit daraus folgend eine passende Rollenverteilung vorgenommen werden kann. Daraufhin arbeitet sich jedes Teammitglied besonders intensiv in seinen Zuständigkeitsbereich ein (eine ausreichende Vorbereitung und das Selbststudium bei Kenntnisrückständen wird vorausgesetzt). Zusätzlich sollte unter den Teammitgliedern eine gegenseitige Hilfestellung gewährleistet sein. Im schlimmsten Fall kann eine Neubesetzung einzelner Rollen erfolgen!*

## 1.2 Kommunikationsprobleme

Ein weiteres Problem könnte die Kommunikation der Teammitglieder untereinander sein. Mangelnde Kommunikation untereinander oder schlechte Erreichbarkeit der einzelnen Teammitglieder kostet Zeit und könnte Terminverschiebungen zur Folge haben. Es könnte zu Missverständnissen und Unstimmigkeiten kommen, wodurch das ganze Projekt in Gefahr geraten kann.

*Prävention: Zum Ersten gibt es regelmäßige Treffen aller Teammitglieder, in denen man die nächsten Schritte der weiteren Entwicklung und größere Probleme klären kann. Zum Zweiten haben wir einen Mailverteiler eingerichtet, mit dessen Hilfe alle Teammitglieder erreicht werden können. Somit können schnell zu beantwortende Fragen gestellt und Terminabsprachen getroffen werden. Und zum Dritten setzen wir Redmine (Projektmanagement-Werkzeug zur Softwareentwicklung) ein, um den Fortschritt unseres Projektes zu planen und zu koordinieren. Auch sollen nach Möglichkeit alle Absprachen schriftlich festhalten werden. All dies sollte die Gefahr eines Scheiterns minimieren.*

## 1.3 Ausfall von Teammitgliedern

Bei jedem durch Teamarbeit durchgeführtem Projekt, in dem jedes Teammitglied spezielle Aufgaben zu erfüllen hat, kann ein Ausfall eines einzelnen oder mehrerer Teammitglieder durch z.B. Krankheit nicht ausgeschlossen werden, wodurch sich der Aufwand für die übrigen Teammitglieder erhöhen würde und es zu zeitlichen Verzögerungen im Projekt kommen könnte.

*Prävention: Ein solcher Ausfall lässt sich natürlich nicht vorhersehen oder vermeiden, jedoch sollten unsere regelmäßigen Treffen und Absprachen dafür sorgen, dass kein Teammitglied isoliert arbeitet und allein über Teile des Projektes Bescheid weiß, damit ein eventueller Ausfall besser kompensiert werden kann. Es ist vorteilhaft, Lösungen innerhalb des Teams zu diskutieren, wobei ein Einzelner dann die Endfassung schreiben kann. Die Teilaufgaben werden nach Möglichkeit zügig bearbeitet. Der Zeitplan sollte stets Freiraum für eventuelle Zwischenfälle enthalten. Auch sollten bei der Rollenverteilung Stellvertreter benannt werden. Des Weiteren ist es von Vorteil, wenn alle Teammitglieder über ein Basiswissen der anderen Zuständigkeitsbereiche verfügen. Alle Aufgaben sollten gleichmäßig verteilt werden. Der Arbeitsfortschritt ist zu dokumentieren, sodass im Ernstfall jedes Teammitglied einfacher zu ersetzen ist.*

## 1.4 Datenverlust und andere technische Probleme

Sämtliche Phasen unseres Projektes werden auf einer rechnergestützten Plattform erarbeitet. Allgemeine Hardware- bzw. Softwareprobleme können zu Daten- oder Informationsverlust führen. Die Behebung von technischen Problemen wie z.B. Serverausfällen, einem schlechten Internetzugang oder dem Ausfall von Hardware benötigt zusätzliche Zeit und verursacht ggf. weitere Kosten.

*Prävention: Die Teammitglieder sollten die Intervalle der Veröffentlichung ihrer Daten so weit wie möglich verkürzen. Der Einsatz eines Versionsverwaltungssystems ist von Vorteil. Der technische Assistent hat die Aufgabe, regelmäßig ein Backup aller Daten zu erstellen, die von den Teammitgliedern zentral auf dem zur Verfügung gestellten Rechner der Universität gespeichert werden. Zusätzlich kann die Gefahr eines Datenverlust durch das Anlegen von Sicherheitskopien auf den heimischen Systemen aller Teammitglieder minimiert werden.*

### **1.5 Nicht eindeutige oder missverständene Kundenanforderungen**

Während der Planungsphase muss man ständig den Kontakt zum Auftraggeber halten, um seine Produkt-Spezifikationen zu erfahren. Dabei können leicht Missverständnisse auftreten. Das liegt vor allem daran, dass der Softwareentwickler hauptsächlich im Lösungsbereich denkt, wohingegen der Kunde, der im Allgemeinen nicht das genügende Wissen über die Softwareentwicklung mitbringt, im Problembereich denkt. Das kann dazu führen, dass man zwar über das Gleiche spricht, aber etwas Unterschiedliches meint. Schwierigkeiten bei der Kommunikation mit dem Auftraggeber oder Missverständnisse bei der Interpretation der Projektanforderungen können einen zeitlichen Mehraufwand, erhöhte Kosten und eine falsche Implementierung des Projektes zur Folge haben.

*Prävention: Um diesem entgegen zu wirken, nutzt man Lastenheft, Pflichtenheft und Glossar, um eine einheitliche Terminologie zu gewährleisten und möglichst genau zu spezifizieren, wie genau etwas später mit dem Softwareprodukt umgesetzt werden soll. Gleichzeitig erfolgt eine regelmäßige Konsultation mit dem Auftraggeber, um falsche Entwicklungen so früh wie möglich festzustellen.*

### **1.6 Zeitaufwand / Abgabefristen**

Über das gesamte Praktikum hindurch ist die Zeit zur Aufgabenerfüllung sehr eng bemessen. Aufgaben werden unterschätzt, und somit wird die Bearbeitung nicht zeitgerecht angegangen. Zugleich könnte es mit den Zeitressourcen aufgrund anderer Vorlesungen, Übungen, etc. knapp werden. Somit passiert es schnell, dass man eine geforderte Aufgabe nicht termingerecht erfüllen kann. Die folgenden Aufgaben stehen bereits an, während ältere Probleme noch nicht gelöst wurden. Die Projektbearbeitung gerät in Zeitmangel, und kann daher nicht mit der gewünschten Qualität fortgeführt werden. Aus den bereits genannten und auch anderen Gründen besteht allgemein das Risiko, dass vorgegebene Fristen nicht eingehalten werden.

*Prävention: Dieser Zeitnot muss man schon von Anfang an entgegen wirken. Dies realisiert man am Besten durch rechtzeitiges Auseinandersetzen mit den Aufgabenstellungen, einer genauen Planung der Zeiteinteilung und natürlich durch die gleichmäßige Verteilung der Aufgaben auf die einzelnen Teammitglieder. Natürlich erfordern diese Maßnahmen die absolute Zuverlässigkeit der gesamten Teammitglieder. Mögliche zeitliche Engpässe sollten rechtzeitig zur Sprache gebracht werden, damit Gegenmaßnahmen ergriffen werden können. Es wird eine ToDo-Liste erstellt, in denen Aufgaben frühzeitig erfasst und zugeteilt werden. Zuständigkeiten werden klar formuliert*

*und Fortschrittsberichte (Rückmeldungen) sind von jedem einzelnen Mitglied zu erstellen. So ist der Projektleiter bzw. der Verantwortliche stets über den Entwicklungsstand informiert und kann Rückstände früh erkennen, um ihnen entgegen zu wirken.*

### **1.7 Programmierprobleme (Fehler, mangelnde Dokumentation, etc.)**

Wird nicht zeitgleich der Codeerstellung dokumentiert, gehen wichtige Informationen verloren. Warum welche Lösungsansätze anderen vorgezogen wurden ist nicht mehr ersichtlich. Nachträgliches Dokumentieren erfordert erneute Einarbeitung in den Code und verlangt damit einen erhöhten Aufwand. Zudem ist es möglich, dass sich Fehler in den Quellcode einschleichen und damit ein ordnungsgemäßes Funktionieren der Software verhindern.

*Prävention: Jedes Mitglied ist angehalten seinen Code ausreichend zu dokumentieren. Anderen Mitgliedern fällt das Verstehen fremden Codes so deutlich leichter und sie können somit von bisherigen Lösungen profitieren. Gleichzeitig wird Pair-Programming angewandt, um die Software-Qualität zu steigern. Die Qualitätssicherung prüft dies regelmäßig und weist zeitnah auf Missstände hin. Weiterhin ist es wichtig ein gutes Testkonzept zu erarbeiten. D.h. es sollte versucht werden alle möglichen auftretenden Situation durch ordentliche Unit-Tests bereits kurz nach Beginn der Implementierungsphase zu simulieren, um so Fehler in der Software frühzeitig zu erkennen und dafür entsprechende Lösungen zu finden.*

### **1.8 Desinteresse und mangelnde Motivation**

Es könnte während der Arbeit am Softwaretechnikpraktikum dazu kommen, dass einige Mitglieder nicht die erforderliche Motivation aufbringen können, um sich aktiv am Projekt zu beteiligen. Dies kann zu Ineffizienz im Projekt führen. Des weiteren könnte es passieren, dass manche Mitglieder des Teams in dem Projekt eine unlösbare Aufgabe sehen und deshalb aufgeben. Dies stört nicht nur die Arbeitsatmosphäre als solche, sondern erschwert auch eine rege Kommunikation untereinander.

*Prävention: Das Team arbeitet an einem Projekt. Jeder einzelne sollte zu einem angenehmen Arbeitsklima in der Gruppe beitragen. Das gute Gelingen des Projektes ist das Primärziel und ist allen Mitgliedern zu allen Zeitpunkten klar. Spaß macht, was man kann. Also werden große Aufgabe in kleine, überschaubare Teilaufgaben geteilt, so das jedes Mitglied Aufgaben löst, die es gut bewältigen kann. Jeder ist mit Allem etwas vertraut, damit Aufgaben von Ausfallenden übernommen werden können.*

### **1.9 Komplexität des Projekts**

Der Umfang sowie die Komplexität wird falsch eingeschätzt. Anfangs leicht aussehende Aufgaben können sich als deutlich komplizierter und aufwändiger herausstellen. Wenn plötzlich neue nicht bedachte Probleme auftreten, so kann der Aufwand stark ansteigen. Dadurch kommt es zu Verzögerungen in der Umsetzung des Projekts und es entsteht die

Gefahr der Nichteinhaltung von Fristen. Dies erhöht den Zeitdruck und kann Probleme mit dem Auftraggeber auslösen.

*Prävention: Eine gute Aufwandsabschätzung erkennt solche Probleme schon im Voraus. Frühes Anfangen mit der Aufgabe legt diese Probleme rechtzeitig offen und lässt Handlungsfreiheit zum bearbeiten. Zugleich erfolgt eine regelmäßige Kontrolle und Protokollierung des Zeitaufwands. Die Arbeit von Heute nicht auf Morgen verschieben!*

### 1.10 Unvollständige Risikoanalyse

Trotz intensiver Diskussion sind Risiken falsch eingestuft worden, oder fehlen komplett. Auch sind aus mangelnder Auseinandersetzung mit weiteren möglichen Risiken, Maßnahmen zum Entgegenwirken nicht vorhanden oder werden erst zu spät getroffen. Risiken werden somit unkontrollierbar, und stellen eine Gefährdung des Projektes dar.

*Prävention: Die Risikoanalyse wird im Verlauf der Arbeit mehrmals erneut aufgestellt. Alte Risiken werden angepasst, und neu aufgetauchte Probleme in die Liste aufgenommen. Somit wird angestrebt, zu jedem „Worst-Case-Szenario“ die erforderlichen Gegenmaßnahmen zu kennen, und zu beherrschen.*

## 2 Rollenverteilung

Im Treffen zur Risikoanalyse zeigten sich bereits einige Stärken und Schwächen der Teammitglieder. Auch wurden dabei Erfahrungen und Wünsche bezüglich möglicher Aufgaben während des Praktikums ausgetauscht. Somit konnten wir die erforderlichen Rollen/Verantwortlichkeiten innerhalb unseres Projektes treffend zuweisen.

---

Rolle	Name
Projektleiterin	Anne Vogler
Technischer Assistent	Philipp Möller
Verantwortlicher für Recherche	Sascha Haseloff
Verantwortlicher für Modellierung	David Lamparter
Verantwortlicher für Tests	Martin Huschenbett
Verantwortlicher für Implementierung	David Bieberstein
Verantwortlicher für Qualitätssicherung und Dokumentation	Thomas Lehmann

---

## 3 Homepage der Gruppe

<http://pcai042.informatik.uni-leipzig.de/~swp09-5/>