1 Risikoanalyse

Die Entwicklung von Software ist grundsätzlich ein mit verschiedenen Risiken verbunden, welche vor Beginn der eigentlichen Entwicklungstätigkeit identifiziert und mit geeigneten Mitteln vermieden werde müssen.

1.1 Probleme mit der Aufgabenstellung

1.1.1 Unzureichendes oder fehlendes Verständnis der Aufgabenstellung

Das gesamte Team muss daran interessiert sein, die Aufgabenstellung möglichst gut zu verstehen und sich darüber auszutauschen. Nur so können Missverständnisse an dieser Stelle soweit wie möglich ausgeschlossen werden. Gerade bei einem explorativen Projekt mit flexibleren Anforderungen ist es wichtig, dass jedes Teammitglied sich zu jedem Zeitpunkt voll über die Aufgabe im Klaren ist.

1.2 Konflikte im Team

1.2.1 Mangelnde oder fehlerhafte Kommunikation

Diesem Problem begegnen wir neben Selbstermahnung durch die frühzeitige Etablierung fester Abläufe und Kanäle über die wir Informationen austauschen. Diese werden im Idealfall zur Gewohnheit, sodass erst Störungen von anderer Stelle zu Problemen mit der Kommunikation im Team führen können. Eine gut ausgebaute gruppeninterne Infrastruktur (Mailinglisten, Wikis, etc.) können die gute Kommunikation im Team signifikant unterstützen.

1.2.2 Ausfälle

Ausfälle können sich immer ereignen. Solange sie jedoch nicht gehäuft auftreten ist damit zu rechnen, dass sie zumindest temporär durch das Team

kompensiert werden können. Dies betont insbesondere eine effiziente Zeitplanung wie auch konstant gute Kommunikation während eines Ausfalls und in der sukzessiven Wiedereinarbeitungsphase nach dem Ausfall.

1.2.3 Schlechte Rollenverteilung im Team

Es kann sich herausstellen, dass mit Rollen verknüpfte Aufgaben aus unterschiedlichen Gründen schlecht erledigt werden. Da dies fatale Folgen für das Team und die Erfüllung der Aufgabenstellung insgesamt haben kann, ist es wichtig die Rollen mit großer Sorgfalt zu verteilen, Probleme schnell festzustellen und gegebenenfalls die Rollenverteilung zu reevaluieren.

1.2.4 Überforderung

Durch unterschiedlich ausgebaute Kentnisse oder schlechte Arbeitsverteilung kann es zu Überforderungen kommen, die die Motivation senken und Qualität wie Geschwindigkeit der Arbeit verringern. Um dem entgegen zu wirken, bieten sich regelmäßige Reports zur eigenen Situation an. Kommunikation und gemeinsames Lernen am Problem können wesentlich dazu beitragen, dieses Problem zu eliminieren. Zusätzlich sollte eine Einarbeitung in die verwendeten Technologien und Programmiersprachen bereits zum jetzigen Zeitpunkt geschehen und nicht auf die Implementierungsphase verschoben werden.

1.3 Arbeits- und Zeitmanagment

1.3.1 Deadlines

Nach bereits bestehenden Erfahrungen in der Softwareentwicklung ist zu erwarten, dass es zu Schwierigkeiten bei der fristgerechten Abgabe geforderter Artefakte kommen wird. Um diese unerwünschten Fehlentwicklungen zu umgehen, ist neben funktionierender Arbeitsteilung und Kommunikation ein frühes Angehen des Problems hilfreich um nicht unnötig Zeit zu

verschenken und eventuell auftretende Komplikationen rechtzeitig kompensieren zu können. Gruppeninterne, gestaffelte Deadlines sind ein Mittel um in Notfällen umdisponieren zu können. Falls es Zweifel am prognostizierten Arbeitsaufwand einer Teilaufgabe gibt oder sich dieser nicht adäquat ermitteln lässt, sollte besonders darauf geachtet werden, dass gegebenenfalls zusätzliche Ressourcen zur Erledigung vorhanden sind.

1.3.2 Unzureichend erledigte Arbeitsschritte

Durch unzureichend erledigte Aufgaben werden unter Umständen darauf aufbauende Arbeitsschritte blockiert. Daher muss insbesondere mit den anschließend Beteiligten kommuniziert werden und dass eigene Können und Verständnis der Aufgabenstellung überprüft und ggf. nachgebessert werden. Das Ablegen von frühen Entwürfen in Wikis oder Versionskontrollsystemen erlaubt es den anderen Teammitgliedern den korrekten Ablauf zu verifizieren und schnell einzugreifen.

1.3.3 Veränderliche Anforderungen

Sind die Anforderungen schlecht formuliert oder ändern sich während der eigentlichen Bearbeitung kann dies die Qualität negativ beeinflussen. Hier hilft, neben einem starken Augenmerk auf teaminterner Kommunikation auch auf solche Unklarheiten zu achten und die Arbeit zu unterbrechen bis sichergestellt ist, dass die Problemstellung klar definiert ist. Eine klare Kommunikation mit dem Auftraggeber sollte selbstverständlich sein um eine spätere Änderung der Anforderungen weitestgehend zu vermeiden.

1.4 Technische Probleme

1.4.1 Lange Einarbeitungszeiten

Lange Einarbeitungszeiten beim Umgang mit benutzten Technologien bei Einzelnen oder der ganzen Gruppe können die Arbeit verzögern und unter

anderem das Erreichen von Deadlines schwierig machen oder die Motivation senken. Dadurch, dass alle Gruppenmitglieder bei regelmäßigen Treffen ihre aktuelle Situation schildern und gemeinsam beraten wird, kann dieses Risiko minimiert werden.

1.4.2 Ausfall von Infrastruktur

Technische Probleme können immer auftreten und sind oft schlecht vorauszuahnen. Da wir mit Mercurial über ein verteiltes Versionierungssystem verfügen ist davon auszugehen, dass Projektdaten vielfach vorhanden sind und somit sogar auf einen zentralen Server zeitweise verzichtet werden könnte. Die Auslagerung gruppeninterner Ressourcen auf redundant angebundene Rechnertechnik vermindert die Wahrscheinlichkeit, dass die Gruppenkommunikation zum Erliegen kommt, ausser für den unwahrscheinlichen Fall, dass sowohl universitäre als auch gruppeninterne Infrastruktur ausfällt.

2 Rollenverteilung

Rolle	Person
Projektleiter	Fabian Todt
Technischer Assistent	Rajko Hartmann
Verantwortliche für Recherche	Katharina Hößel
Verantwortlicher für Modellierung	Christoph Blümel
Verantwortlicher für Tests	Benjamin Kiessling
Verantwortlicher für Implementierung	Benjamin Kiessling
Verantwortlicher für Qualitätssicherung und Dokumentation	Jakob Runge