

Risikoanalyse

Semantic Chess

10.12.2013

Interne Risiken

1. Unerwartetes Ausscheiden eines Teammitglieds

Aufgrund plötzlicher Erkrankung oder dem unerwarteten Scheitern bei der Endklausur, welche eine Voraussetzung für die Teilnahme an dem Software-Technik-Praktikum darstellt, kann ein Teammitglied vorzeitig aus dem Teamprojekt ausscheiden.

Lösung:

Durch Kommentierung des Quellcodes, der Rechercheergebnisse und Führung von kleineren Protokollen sollte jeder in der Lage sein sich ohne weitere Umstände kurzfristig in das Teilprojekt einzuarbeiten. Zusätzlich festigen regelmäßige Meetings mit den einzelnen Teammitgliedern (auch ohne Betreuer) den gemeinsamen Wissenstand.

2. Zeit läuft davon

Durch ungenaue Zeiteinteilung der einzelnen Projektschritte oder unerwartet schwierige Teilaufgaben, welche viel Zeit in Anspruch nehmen, kann es im Gesamtprojekt zu Zeitverzögerungen kommen, welche die Fertigstellung des Projektes im zeitlich abgesprochenen Rahmen nahezu unmöglich machen.

Lösung:

Durch Einsetzen vieler kurz aufeinander folgender Deadlines soll ein Aufschub der Teilaufgaben vermieden werden. Zu diesem Zweck müssen Teilaufgaben konkret definiert werden.

3. Kommunikationsprobleme/Organisatorisches/Ernst

Mangelhafte Kommunikation zwischen den Teammitgliedern sowie mangelndes Organisatorisches Geschick und Ernst bezüglich der Meetings führen zur Ausbremsung des gesamten Projekts.

Lösung:

Es ist wichtig Probleme jeglicher Art sofort anzusprechen und so mögliche Hilfe zu erhalten. Das Festsetzen eines obligatorischen wöchentlichen Termins ist unabdingbar für die strikte Organisation. Weiterhin entgegnet der eingerichtete Mail-Verteiler dieser Problematik.

4. Aufgabenverteilung

Durch falsche Verteilung oder mangelnde Einhaltung der Aufgaben und Rollen können Komplikationen auftreten. Hierzu gehören zum Beispiel eine ungewollte Rolle oder Aufgabe innerhalb des Projekts, das Verlassen des Zuständigkeitsbereiches, Einmischen in andere Aufgaben oder die ungleiche Verteilung der Arbeit.

Lösung:

Es gilt sich klar zu machen dass die Rollen der „Verantwortlichen“ nur organisatorisch sind, sodass jeder alles machen muss, allerdings angeleitet und organisiert durch den Verantwortliche für den jeweiligen Bereich. Wichtig ist zudem eine gute Modularisierung, sodass sich die einzelnen Mitglieder auf eine Aufgabe konzentrieren können. Voraussetzung ist außerdem eine möglichst ausgeglichene Aufgabenverteilung.

5. Verschiedene Standpunkte des Wissens

Ein mögliches Problem könnte der unterschiedliche Wissens stand der Teammitglieder, durch die Spezialisierung auf einzelne Teilgebiete oder Vorwissen werden. Hierdurch können Missverständnisse sowie Probleme bei der Aufgabenverteilung auftreten.

Lösung:

Unabdingbar für das gegenseitige Verständnis ist selbstverständlich eine gute Dokumentation, sodass fremder Code leicht verständlich ist. Zudem sollten Mitglieder die Hilfe benötigen berücksichtigt und an die Hand genommen werden.

6. Ziellosigkeit

Durch die Spezialisierung auf eine modulare Teilaufgabe kann es passieren, dass beim Programmieren oder Modellieren am eigentlichen Ziel vorbei gearbeitet wird.

Lösung:

Um die falsche Interpretation zu vermeiden, müssen Aufgaben und Zielstellungen konkret formuliert werden und immer in den Kontext des gesamten Projekts eingeordnet werden. Zudem werden regelmäßige Absprachen stattfinden.

7. Blocker als Problem

Ein großes Problem könnten unauffindbare Bugs und Blockaden in Modellierung oder Programmierung werden.

Lösung:

Um solche „Blocker“ zu beheben sollte die Gruppe ihre Priorität verlagern, sich auf das Problem konzentrieren und gemeinsam eine Lösung suchen.

8. Umfangreiche Änderung bestehenden Codes

Schwer wird es wenn ein Bug oder eine nötige Verbesserung so tief im Code verankert ist, dass eine umfangreiche Änderung des bestehenden Codes unumgänglich ist. Aus eigenen Erfahrungen ist dies äußerst unangenehm.

Lösung:

Der Lösungsansatz ist hier erneut eine gute Modularisierung um nachträgliche Änderungen zu erleichtern. Außerdem sind regelmäßige Tests nötig um mögliche Fehler zu finden und zu beheben.

Externe Risiken

1. Technische Probleme

Hardwareprobleme oder potentieller Datenverlust können während des Projektes auftreten.

Lösung:

Durch häufige Backups und dicht gestaffelte Commits können die Auswirkungen eines Ausfalls minimiert werden. Bei Hardwareproblemen kann auf die Uni-PCs zurückgegriffen werden.

2. Stress/Überbelastung

Nicht zu vernachlässigen ist die Gefahr der Überbelastung und Stressreaktion aufgrund verdichteter, schwieriger Aufgaben oder äußerer Stressfaktoren.

Lösung:

Um Stress und dessen Auswirkungen entgegen zu treten ist es von Nöten Probleme direkt mit der Gruppe zu besprechen, Hilfe zu verlangen und sich selber nicht zu stark zu belasten. Bei der Zeiteinteilung, sollte für ausreichend Puffer sowie Pausen gesorgt werden.