

Risikoanalyse

Thema: Semantic Chess

Gruppe: swp13-sc

Verantwortlich: S. Hildebrandt, L. Kohlmeyer

Letzte Änderung: 15.12.2012

Einleitung:

Da mit jedem größeren Softwareprojekt einige Risiken verbunden sind, die die korrekte Ausführung und einen zufriedenstellenden Abschluss des Projektes bedrohen, ist es unabdingbar im Vorfeld des Projektes diese Risiken aufzulisten, zu analysieren und geeignete Lösungsstrategien sowie entsprechende Gegenmaßnahmen zu entwickeln.

Im Folgenden werden die 10 wichtigsten Risiken aufgelistet und erläutert, anschließend wird ein auf diese Risiken zugeschnittener Maßnahmenplan vorgestellt.

Identifizierte Risiken:

- unzureichende Absprache bzw. kommunikative Mängel im Team (Team-intern), oder mit den Betreuern (Team-extern);
- Zerschneiden des Teams durch interne Streitigkeiten (Team-intern);
- Fehleinschätzungen im Zeitmanagement (Team-intern);
- sinkende Motivation, bedingt durch quantitativ oder qualitativ zu hohe Anforderungen an einzelne oder mehrere Teammitglieder (Team-intern);
- durch Krankheit oder technisches Versagen bedingte Probleme (Team-intern);
- Mängel oder gar Fehlinformationen bei der Recherche (Team-intern);
- mangelnde bzw. fehlerhafte Modellierung (Team-intern und -extern);
- mangelhafte Dokumentation (Team-intern);
- zu enge Kalkulation der Testphase (Team-intern);
- Kompatibilitäts- und Implementierungsprobleme (Team-intern und -extern);

Maßnahmenplan

1. Unzureichende Absprache bzw. kommunikative Mängel im Team oder mit den Betreuern

Problem: Hierbei kann ein unzureichendes Verständnis der jeweiligen Aufgabe eines einzelnen Teammitglieds sowie allgemeine Fehler im Gesamtablauf der Projektplanung und Ausführung die Folge sein, was zu schwerwiegenden Verzögerungen bzw. einem kompletten Scheitern des Projektes führen kann.

Maßnahmen: Um eine hinreichende Absicherung gegenüber diesem Problem zu gewährleisten, ist es absolut essenziell, dass sich die Teammitglieder untereinander und mit jeweiligen Betreuern in regelmäßigen Abständen treffen, um auftretende Probleme zu besprechen und Unstimmigkeiten im Ablauf zu beheben. Des Weiteren sind unabhängig davon Kommunikationskanäle zu schaffen und zu pflegen, um zu gewährleisten, dass zu jeder Zeit Absprachen zu auftretenden Problemen gehalten werden können. Bei diesen Kommunikationskanälen wurde sich auf E-Mail-Verteiler, Facebook und Skype geeinigt – allesamt leicht nutzbare Tools, die zu unterschiedlichen Kommunikationszwecken herangezogen werden können. Zur Terminkoordination wird das Online-Tool Doodle beansprucht.

2. Zerbrechen des Teams durch interne Streitigkeiten

Problem: Das Auftreten von Antipathien bzw. Streitigkeiten privater sowie fachlicher Natur ist bei einem Team, welches sich gerade erst kennenlernt und zum ersten Mal an einem größeren Software-Projekt arbeitet, ein nicht zu vernachlässigender Faktor. Dies kann im Extremfall zum Zerbrechen des Teams und damit zum Scheitern des Projektes führen. Auch Gruppenbildung und die Isolation einzelner Teammitglieder kann hier die Folge sein.

Maßnahmen: Wie in den Maßnahmen zu *Punkt 1* erwähnt, sind hierbei regelmäßige Treffen in jedem Falle förderlich. Diese sollten allerdings nach Möglichkeit nicht nur im universitären Rahmen abgehalten werden, um auch das menschliche Miteinander der Teammitglieder zu fördern. Der respekt- und empathievollen Umgang der Teammitglieder ist eine wichtige Voraussetzung zur Prävention dieses Risikos.

3. Fehleinschätzungen im Zeitmanagement

Problem: Die fehlerhafte Einschätzung und Veranschlagung von Zeitfenstern verschiedener Abläufe bzw. Projektphasen kann zur Überschlagung der Teamressourcen führen; im schlimmsten Fall zur Nichteinhaltung von wichtigen Terminen.

Maßnahmen: Hier ist es wichtig Pufferzeiträume zu schaffen. Diese sollen zum einen zum frühzeitigen Fertigstellen einfacher Aufgaben führen, was die Zeitfenster schwierigerer Aufgaben vergrößert, zum anderen ein frühzeitiges Fertigstellen komplexerer Aufgaben in Betracht ziehen, so dass im Zweifelsfall die noch nötigen Zeitressourcen für eine termingerechte Abgabe zur Verfügung stehen. Des Weiteren steht gerade hier der Teamleiter in der besonderen Verantwortung Projektabläufe durch Absprache mit einzelnen Mitgliedern oder mit dem gesamten Team bestmöglich zu koordinieren.

4. Sinkende Motivation, bedingt durch quantitativ oder qualitativ zu hohe Anforderungen an einzelne oder mehrere Teammitglieder

Problem: Die zu starke Arbeitsbelastung einzelner Teammitglieder sowie die mögliche fachliche Überforderung dieser, kann unter Umständen dazu führen, dass Teammitglieder die Lust am Projekt verlieren, nur noch halbherzig arbeiten und im schlimmsten Fall sogar das Team verlassen. Dies kann sich stark negativ auf die Weiterarbeit des Teams auswirken.

Maßnahmen: Um diesem Risiko vorzubeugen, ist es wichtig eine sinnvolle und faire Aufteilung der einzelnen Projektaufgaben bzw. des Aufgabensumms vorzunehmen. Außerdem beugt eine rechtzeitige entsprechende Rückmeldung an den Verantwortlichen des jeweiligen Arbeitsschrittes sowie an den Projektleiter und eine erfolgende entsprechende entlastende Reaktionen, diesem Risiko vor.

5. Durch Krankheit oder technisches Versagen bedingte Probleme

Problem: Dieses Risiko ist allgegenwärtig und es ist nur schwierig zu minimieren. Die daraus resultierenden Probleme können leicht zu behebbende, aber auch schwerwiegende Folgen haben. Sie alle lassen sich unter den Begrifflichkeiten Zeitverzögerung und erhöhter Arbeitsaufwand zusammenfassen. Besonders in der Test- und Implementierungsphase können technische Ausfälle zu starken Problemen führen.

Maßnahmen: Um Risiken bei krankheitsbedingten Ausfällen zu vermeiden, sollte man sich schon im Vorfeld über etwaige Vertretungen Gedanken machen und diese festlegen. Des Weiteren sollten auch dem Projektleiter alle Zugangsdaten für die einzelnen Projektwerkzeuge bekannt sein. Bei unbeheblichen technischen Problemen sollte schnellstmöglich ein Betreuer kontaktiert werden.

6. Mängel oder gar Fehlinformationen bei der Recherche

Problem: Sollte das Team mangelhaft oder gar fehl informiert sein, so hat das sehr nachhaltige Folgen, die sich durch das ganze Projekt ziehen können. Das Verständnis des Teams für das Projekt wird geschwächt und kann schon bei der Modellierung zu nachhaltigen Fehlern führen.

Maßnahmen: Eine sehr ausführliche und verantwortlich ausgeführte Recherche ist hier der erste Schritt, um dieses Risiko zu minimieren. Außerdem sollten auf die einzelnen Teammitglieder zugeschnittene Recherche-Aufgaben und ein anschließendes gemeinsames Review-Gespräch im Team stattfinden.

7. Mangelnde bzw. Fehlerhafte Modellierung

Problem: Fehler in der Modellierung ziehen eine Menge von Problemen nach sich. Im Allgemeinen führen sie zur fehlerhaften Ausführung des Projektes oder einem von der Aufgabenstellung abweichendem Ergebnis. Im schlimmsten Fall führen sie dazu, dass die Implementierung nur schwer, sehr zeitaufwändig oder gar nicht zu realisieren ist.

Maßnahmen: Wie bei der Recherche ist eine sinnvolle Arbeitsteilung in dieser Phase das A und O. Die Stärken der Modellierungsfähigkeiten der einzelnen Teammitglieder sollte hier sehr berücksichtigt werden. Eine sinnvolle Modellierung setzt unter anderem eine gute Recherche voraus, die das Wissen dafür liefert, welche Konzepte mit welchem Werkzeug am besten realisierbar sind. Ein weiterer bedeutungsvoller Punkt dieser Phase ist die Einigung auf eine Namenskonvention (bzgl. Variablen, Klassen), um Unstimmigkeiten zu verhindern.

8. Mangelhafte Dokumentation

Problem: Änderungen am Code sowie die Wartung der Software sind ohne ausreichende Dokumentation im Nachhinein nur schwer oder gar nicht realisierbar. Außerdem führt die mangelnde Dokumentation während der gesamten Projektlaufzeit zu Problemen bei der Nachvollziehbarkeit der einzelnen Arbeitsschritte der verschiedenen Projekt-Phasen.

Maßnahmen: Zur Sicherstellung einer ausführlichen Dokumentation ist es ratsam formlose stichpunktartige Protokolle zu den einzelnen Teamsitzungen anzufertigen, um einzelne Arbeitsschritte besser nachvollziehen zu können. In der Implementierungsphase sollte der Code ausreichend kommentiert werden.

9. zu enge Kalkulation der Testphase

Problem: Eine zu spät einsetzende Testphase bewirkt, dass sich bisher unbemerkte Fehler in der Software auch erst sehr spät bemerkbar machen, so dass nur wenig Zeit bleibt, diese Fehlerquellen unschädlich zu machen, was zu starkem Terminverzug in der Endphase des Projektes führen kann und einen vermeidbaren Stressfaktor birgt.

Maßnahmen: Um dies zu verhindern, sollte mit der Testphase so früh wie möglich begonnen werden, also unmittelbar nachdem erste Software-Module fertig gestellt wurden. Dabei sollte eine Prüfung auf modulinterne Funktionalitäten erfolgen, aber auch auf das Zusammenspiel bestimmter Module. Sofern möglich sollten bereits die Prototypen früherer Projektphasen getestet werden.

10. Kompatibilitäts- und Implementierungsprobleme

Problem: Da es sich bei dem veranschlagten Softwareprojekt um eine Web-Anwendung handelt können nicht nur Probleme bei den Server- und Clientanwendungen auftreten, es können auch Probleme bei der Darstellung der Software in unterschiedlichen Browsern auftreten. Weiterhin kann es durch fehlerhaft definierte oder fehlerhaft umgesetzte Schnittstellen zwischen Semantic Web Datenbanken und dem Abfragen von Daten aus diesen zu Kompatibilitätsproblemen kommen. Andere Kompatibilitätsprobleme können bei zu stark abweichender Namensgebung auftreten.

Maßnahmen: Probleme dieser Art können nur durch eine sehr ausführliche Testphase, guter Modellierung und mit einer guten Dokumentation minimiert werden. Gegebenenfalls sollte spezifisches Fachwissen der Betreuer herangezogen werden. Vor dem Hintergrund der Kompatibilität sollte auch eine entsprechende Programmiersprache, wie zum Beispiel Java, gewählt werden. Wie in *Punkt 7* erwähnt, sollte sich auf eine einheitliche Namensgebung geeinigt werden.