

# Webanwendung zur Extraktion von Teildatensätzen aus DBpedia

Christian Ernst, Dominik Strohscheer, Hans Angermann  
Till Nestler, Marvin Hofer, Robert Bielinski, Jonas Rebmann

## Risikoanalyse

### Externe Probleme

#### Ausfall technischer Ressourcen

Wenn technische Ressourcen ausfallen kann es zu schwerwiegendem Datenverlust kommen, der den Fortschritt des Projektes beeinträchtigen kann.

Ein Ausfall externer Ressourcen ist zwar unwahrscheinlich, doch ist es trotzdem ratsam alle Gruppenmitglieder anzuhalten, eine eigene lokale Sicherungskopie des Projektes und der letzten Änderungen zu erstellen. Die Nutzung eines dezentralisierten Versionierungssystems wie git ist hier auch hilfreich.

#### Personalausfall

Es kann jederzeit passieren, dass Mitglieder des Projekts länger krank werden, das Praktikum abbrechen oder aus anderen Gründen ausfallen. Folglich wäre die Rolle im Projekt dann unbesetzt bis eventuell Rollen getauscht werden. Im Extremfall könnte die Kommunikation mit dem Entwickler plötzlich abbrechen.

Diesem Problem kann man durch gute Gruppenkommunikation teilweise entgegenwirken, allerdings lässt sich durch die Kommunikation vor allem der Schaden des Ausfalls begrenzen, da man, wenn man frühzeitig Bescheid weiß, bereits Aufgaben umverteilen kann und man nicht spontan einen Motivationsabfall hat. Um den Schaden, den ein solcher Ausfall verursachen könnte, zu minimieren, sollte auch die Arbeit jedes Mitglieds jederzeit für jeden nachvollziehbar sein. Alle Aufgaben sollten so gut dokumentiert und jeder Code so leserlich sein, dass andere Entwickler ohne Einweisung einspringen können. Außerdem sollten alle Mitglieder des Entwicklerteams wichtige Aufgaben wie die Einhaltung von Deadlines im Blick haben.

### Interne Probleme

#### Unverständlicher Code

Wenn Code unverständlich geschrieben und dazu noch ungenügend kommentiert ist, kann es zu Problemen kommen sollte ein anderes Teammitglied den Code bearbeiten, beziehungsweise wegen Fehlern untersuchen müssen. Auch der Programmierer selbst könnte nach einiger Zeit nicht mehr in der Lage sein, seinen eigenen Code zu verstehen.

Daher sollte jeder in der Gruppe seinen Code von sich aus so übersichtlich wie möglich halten und vor allem immer gut kommentieren. Man kann unter dem Gesichtspunkt der Qualitätssicherung in der Gruppe auch Regeln aufstellen, wie der Code auszusehen hat. Außerdem hilft es, wenn man hin und wieder jemand anderen über seinen Code schauen lässt, sodass dieser Unverständlichkeiten direkt anmerken kann. So sollte jeder in der Lage sein den Code des anderen zu verstehen.

## **Mangelnde Kommunikation**

Die Kommunikation bei der gemeinsamen Projektentwicklung spielt eine sehr wichtige Rolle, da sich die einzelnen Mitglieder über Status, Aufgabenteilung und Planung im Projekt gewiss sein müssen. Findet die Kommunikation zwischen den Mitgliedern nur ausreichend statt kann dies zu Missverständnissen und Problemen in der Projektentwicklung führen.

Die Aufgaben der einzelnen Personen und nächste Ziele im Projekt sollten in Gruppentreffen ausreichend diskutiert werden, sowie für alle Mitglieder auffindbar vermerkt sein. Wichtige Meldung sollten nie an einzelne Personen herangetragen werden sondern immer an das gesamte Team. Auf einzelne Fragen sollte in angemessener Zeit geantwortet werden.

## **Schwindende Motivation**

Die Motivation beim Beginn eines Projekts kann schnell abebben, insbesondere wenn Probleme auftreten, die schwer zu bewältigen scheinen. Manche Entwickler verlieren wahrscheinlich schneller die Motivation als andere.

Diesem Problem lässt sich entgegenwirken indem alle, insbesondere der Projektleiter, sich dieser Gefahr bewusst sind und versuchen, Aufgaben so zu verteilen, dass sich niemand alleine mit zentralen Problemen konfrontiert sieht. Außerdem sollten schwerwiegende Probleme möglichst frühzeitig erkannt und gemeinsam Lösungswege erarbeitet werden. Über Motivation und Arbeitsteilung sollte bei den Treffen gesprochen werden.

## **Ungleiche Aufgabenteilung**

Da die Mitarbeiter des Projekts nie das gleiche Pensum an Zeit, Motivation und Kompetenz haben, wird es zu ungleicher Aufgabenteilung kommen. Diese kann dazu führen, dass Entwickler, die mehr arbeiten, überlastet werden oder Gruppenentscheidungen alleine tragen müssen. Entwickler, die weniger involviert sind, könnten den Anschluss an das Projekt verlieren.

Eine ungleiche Aufgabenteilung ist nicht unbedingt schlecht und kann bis zu einem gewissen Grad auch Vorteile für alle im Team haben, kann aber auch sehr schnell für einzelne zur Belastung werden. Um das zu verhindern sollte bei den Gruppentreffen über die Aufgabenteilung gesprochen werden. Probleme, die einzelnen die Arbeit am Projekt erschweren, sollten angesprochen werden. Falls ein Entwickler den Anschluss an das Projekt verliert sollten die anderen versuchen, ihn auf den Stand der Dinge zu bringen damit er seine Mitarbeit fortsetzen kann.

## **Falsche Zielsetzung**

Bei unzureichender oder fehlerhafter Recherche sowie unzureichender Absprache mit dem Auftraggeber kann es vorkommen, dass Teile des Projekts nicht so spezifiziert werden, wie es dem Wunsch des Auftraggebers beziehungsweise den Eigenschaften des genutzten Unterbaus (DBpedia) entspricht, was zu Problemen in der Programmierung oder bei der Abgabe führen kann.

Um zu vermeiden, dass interne Entwicklungsziele falsch gesetzt werden, sollten alle Entwickler sich ausreichend mit der Materie beschäftigen und an der Recherche teilhaben, sodass sie genug Verständnis haben, um die Gegebenheiten korrekt zu nutzen. Außerdem sollte die Kommunikation mit dem Auftraggeber soweit aufrecht erhalten werden, dass alle Ziele klar definiert sind und eventuelle Fragen in der Richtung beantwortet werden können.

## **Unpassende Rollenverteilung**

Die Rollen wurden in Hinblick auf die individuellen Fähigkeiten und Bedürfnisse aller Teammitglieder verteilt, allerdings bereits beim ersten Treffen des Teams, deshalb könnte sich im Verlauf des Projekts herausstellen, dass Kompetenzen oder Aufgaben anders gelagert sind als erwartet.

Sollte sich herausstellen, dass jemand Schwierigkeiten hat die speziellen Anforderungen einer Rolle zu erfüllen sollten andere Teammitglieder ihn unterstützen um ein eventuelles Scheitern des Projekts zu verhindern. Außerdem wäre es denkbar Rollen zu tauschen falls **alle** Beteiligten damit einverstanden sind.

## **Risiken bei der Fertigstellung**

### **Verfehlung von Fristen**

Die erfolgreiche Durchführung eines Projektes hängt stark davon ab, dass gesetzte Fristen, allen voran die mit dem Auftraggeber vereinbarte Hauptfrist, eingehalten werden. Allerdings ist es oft schwierig, den Umfang und somit auch Zeitbedarf einzelner Projektteile einzuschätzen. Vor allem während des Programmierens können unvorhergesehene Schwierigkeiten auftreten, die die Fertigstellung eines Programmteils hinauszögern. Genauso kann ein Entwickler den Arbeitsaufwand unterschätzen und zu spät mit der Programmierung beginnen (In unserem Falle kommt dazu, dass wir nicht sehr erfahren sind). Beide Fälle können zur Verfehlung von internen und schlimmstenfalls eben auch externen Fristen führen, was erhöhten Stress und möglicherweise noch mehr Fehler durch überhastete Entscheidungen nach sich zieht.

Zur Vermeidung von Fristüberschreitungen ist es empfehlenswert, sich gut auf die Arbeit vorzubereiten und Recherche zu betreiben, um Überraschungen zu minimieren. Des weiteren sollten alle Projektmitglieder sich ihre Arbeitszeit gleichmäßig einteilen und früh anfangen, um nicht von plötzlich anschwellenden Pflichten überwältigt zu werden.

### **Fachliche Unkenntnis**

Keines der Projektmitglieder hat bereits alle Werkzeuge mit denen wir arbeiten werden verwendet, weshalb wir uns alle die entsprechenden Fähigkeiten aneignen werden müssen. Falls das scheitert, könnten die Ergebnisse unserer Arbeit fehlerhaft sein, die Motivation stark abnehmen oder ein erhöhter Arbeitsaufwand entstehen.

Um das zu verhindern sollten sich die Entwickler der bevorstehenden Aufgaben bewusst sein. Die Gruppe sollte über die verwendeten Werkzeuge, deren Erlernen und eventuelle Probleme damit kommunizieren. Insbesondere die Zuständigen für Qualitätssicherung, Tests und Implementierung sollten handwerkliche Fehler ansprechen. Alle Entwickler sollten sich der Bedeutung der genauen Kenntnis der verwendeten Werkzeuge bewusst sein und Hinweise oder Verbesserungsvorschläge zu ihrer Arbeit annehmen.