

Risikoanalyse

swp13-olat

Die Durchführung unseres Softwareprojektes bringt viele Risiken mit sich. Um gegen die wichtigsten gerüstet zu sein, ist es von großer Wichtigkeit, diese schon im Vorfeld zu kennen und ihnen mit entsprechenden Maßnahmen zu begegnen. Im Folgenden sind die zehn größten Risiken für unser Projekt mit entsprechenden Gegenmaßnahmen und Lösungsvorschlägen aufgelistet.

Zeitmanagement / Organisation

Mangelhafte Organisation und Zeitplanung führen zu Stress und Fehlern, was schließlich zu einem Qualitätsverlust der Software führen kann.

Lösung: Wichtig ist hier eine rege Kommunikation zwischen den Teammitgliedern; besonders ist dabei die Kommunikation mit Projektleiter hervorzuheben: Dieser muss sich für seine Mitglieder Zeit nehmen und auf ihre Probleme eingehen.

Einarbeitung in das Projekt

Da OpenOLAT ein komplexes System mit umfangreicher Dokumentation ist, ist viel Zeit notwendig um sich in das Projekt einzuarbeiten. Dies kann zu Motivationsschwund und Problemen in der Umsetzung führen.

Lösung: Wichtig ist hierbei, dass die Mitglieder der Gruppe sich gegenseitig unterstützen. Die technisch versierten sollten dabei die technisch weniger erfahrenen gezielt unterstützen.

Motivationsschwund

Motivationsverlust führt schnell zu Ungenauigkeiten und der Vernachlässigung von zugewiesenen Aufgaben, was im schlimmsten Fall zum Ausstieg einzelner Gruppenmitglieder führen kann.

Lösung: Eine aktive Kommunikation in der Gruppe ist sehr wichtig. Probleme müssen offen angesprochen und Unstimmigkeiten geklärt werden. Im Notfall sollten Aufgaben rechtzeitig umverteilt werden.

Hardwareausfall / Datenverlust

Projektdateien können durch Hardwarefehler oder durch ungeschickte Handhabung dauerhaft verloren gehen. Dies kann im schlimmsten Fall zum Abbruch des Projektes führen.

Lösung: Es wird das verteilte Versionsverwaltungssystem git eingesetzt. Zusätzlich werden regelmäßig Backups des Projektserver angelegt.

Mangelnde Erfahrung

Für viele Mitglieder ist es das erste größere Software-Projekt mit dieser Dimension.

Lösung: Wichtig ist, dass die erfahrenen Mitglieder Hilfestellung leisten und auf Fehler aufmerksam machen.

Schwer verständlicher Code / schlechte Kommentare

Unverständlicher Code und schlechte Kommentarqualität führen zu schlechter Les- und Wiederverwend- bzw. Erweiterbarkeit. Dies verlängert das Projekt unnötig.

Lösung: Die ständige Kontrolle des Quellcodes ist hierbei essentiell.

Ausfallende Projektmitglieder

Einzelne Mitglieder könnten aufgrund von Krankheit oder persönlichen Gründen ausscheiden. Dies führt dazu, dass die Organisationsstrukturen durcheinander geraten und Aufgaben unerledigt bleiben

Lösung: Es werden Stellvertreter für jeden Bereich ernannt.

Ungünstige Rollenverteilung

Da das Team bei diesem Projekt zum ersten Mal zusammenarbeitet und jeder einzelne kaum Erfahrungen mit solchen Projekten hat, kann die initiale Rollenverteilung ungünstig sein.

Lösung: Sollte sich die Rollenverteilung als ungünstig herausstellen, müssen Rollen umverteilt werden.

Missverständnisse bei den Anforderungen / mit dem Auftraggeber

Bei der Extraktion der Anforderungen aus dem bereits vorliegenden Anforderungsdokument des Auftraggebers könnten relevante Informationen vergessen oder falsch interpretiert werden.

Lösung: Essenziell ist die aktive Kommunikation mit dem Auftraggeber. Dessen Feedback sollte sehr ernst genommen werden.

Schwierigkeiten mit den verwendeten Werkzeugen

Viele Mitglieder des Teams haben noch nie mit einem Versionsverwaltungssystem gearbeitet oder sich mit Webtechnologien beschäftigt.

Lösung: Hier setzen wir auf die Unterstützung einzelner Mitglieder mit Erfahrungen aus anderen Projekten oder Praktika.

Gruppeneinteilung

Projektleiter: *Björn Henrik Reder*

Technische Assistenz und Recherche: *Klemens Schölnhorn*

Modellierung: *Boie Thomsen*

Implementierung: *Oliver Swoboda*

Tests: *Sebastian Thamm*

Dokumentation: *Robert Seidler*

Qualitätssicherung: *Melanie Mendikowsk*