

Nr.	Status	Risikoanalyse	
		Ursache	Ereignis
		Was verursacht das Risiko?	Was kann passieren?
1	2	3	4
1.1	Monitoring	Mangelnde Disziplin	Deadlines nicht eingehalten
1.2	Monitoring	Fehlende Erfahrung	Falsche Planung, falsches Coden, falsches Testen
1.3	Monitoring	Diverse Gründe	Teilnehmer/Projektleiter fällt weg
1.4	Monitoring	Teilnehmerkonflikte	Streitigkeiten, keine Einigungen möglich, kein Teamwork vorhanden
2	Risiken für die Planungs-/Spezifikationsphase		
2.1	Verhindert	Wünsche des Auftraggebers falsch verstanden	Planung/ Ausführung eines Projekts, dass den tatsächlichen Anforderungen nicht entspricht
2.2	Monitoring	Spezifikationsänderung seitens des Auftraggebers	Bisher erreichtes entspricht den neuen Anforderungen nicht
2.3	Monitoring	Mangelnde Recherche, unerfahrenes Team	Ineffiziente Algorithmen wurden entwickelt
3	Risiken für die Codierungsphase		
3.1	Monitoring	Mangelnde Disziplin/ Erfahrung	Code Convention nicht eingehalten

3.2	Monitoring	Mangelnde Disziplin/ Erfahrung	Konzept missachtet
3.3	Monitoring	Fehlende Dokumentation	Code für außenstehende kaum verständlich
4	Risiken für die Testphase		

Monitoring	Zu wenig Testdaten	Fehler übersehen
Monitoring	falsche Testdaten	Fehler übersehen oder worst case Programm läuft mit richtigen Daten nicht wie vorhergesehen

5	Risiken für die Abschlussphase		
---	--------------------------------	--	--

Monitoring	Schlechte Implementation, Schlechtes Konzept	Zu hohe Systemanforderun- gen
Verhindert	Kein Java/Neo4J beim End- benutzer vorhanden	Programm läuft nicht in der Angedachten Umgebung
Verhindert	Nicht auf Zielgruppe der Be- nutzer angepasstes Produkt	Geringe Akzeptanz des Pro- dukts beim Endnutzer

Auswirkung	Bewertung	Maßnahmen	
		Vorbeugung	Überwachung
Was sind die unmittelbaren Auswirkungen?		Maßnahmen zur Senkung der Eintrittswahrscheinlichkeit oder Auswirkung	Wer überwacht das Risiko und wie?
5	6	7	8
Verschiebung des Zeitplanes der Meilensteine	Normal	Motivation der Mitarbeiter	Das Team überwacht jeden durch Arbeitsberichte
Zusätzlicher Aufwand wegen Fehlerbewegung, zusätzliche Recherche, zusätzliches Testen, Projekt gescheitert	Normal	Zusätzliche Schulung/Recherche einplanen, Gespräche mit dem Projektleiter	Der Projektleiter und das Team, halten Meetings ab um Planung und Wissenstand zu koordinieren.
Arbeitspensum wird nicht geschafft	Kritisch	keine Möglichkeit vorhanden	Jedes Teammitglied beobachtet die Gesundheit, die Studienleistung etc. jedes einzelnen
Unangenehmes Arbeitsklima, Verzögerungen des Projekts, mangelnde Motivation	Normal	Gesellschaftsspiele, Kneipentour, soziale Kontakte und Interaktionen fördern, bewusst kommunizieren	Jeder einzelne achtet auf eine angenehme Atmosphäre und angemessene Kommunikation
Zusätzlicher Aufwand, Projektergebnis womöglich nutzlos	Kritisch	Rege Kommunikation mit dem Auftraggeber	Projektleiter sowie jeder Teilnehmer hinterfragen stets genau die Wünsche des Auftraggebers um Missverständnisse auszuschließen
Zusätzlicher Aufwand, Projektergebnis womöglich nutzlos	Hoch	Rege Kommunikation mit dem Auftraggeber	Versuchen zu verstehen, warum Auftraggeber was wünscht, um womöglich Konflikte in Anforderungen zu erkennen und Änderungen vorwegzunehmen
Fertiges Produkt läuft zu langsam oder gar nicht unter Last	Normal	Sofern möglich mit etablierte Algorithmen arbeiten, Stetig auf Effizienz achten	Jedes Teammitglied achtet insbesondere bei dem Zugeordneten Aufgabenbereich auf Effizienz
Code für außenstehende schwer verständlich	Normal	Von Anfang an auf Konventionen achten, nach Fertigstellen einer Komponente anderes Teammitglied „Probelesen“ lassen	In erster Linie der jeweilige Autor der Komponente, ggf. „Probelesen“ veranlassen

Schlecht erweiterbarer Code, hohe Kopplung geringe Bindung, für Auftraggeber womöglich nur bedingt nützlich	Normal	Vor und nach Implementati- on einer Komponente mit Konzept abgleichen	In erster Linie der jeweilige Autor der Komponente, ggf. „Probelesen“ veranlassen
Endprodukt für Auftraggeber nur bedingt nützlich	Normal	Von vorn herein mit JavaDoc eine Doku im Quellcode er- zeugen	In erster Linie der jeweilige Autor der Komponente, ggf. „Probelesen“ veranlassen

Abstürze oder Performance- probleme im Fertigen Pro- dukt	Hoch	Bewährte Vorgänge zu Soft- waretesting befolgen, Test- datenanalyse	Je Komponente ein Verant- wortlicher für Tests, Testda- tenanalyse
Abstürze oder Performance- probleme im Fertigen Pro- dukt	Hoch	Testdatenanalyse bereits vor Tatsächlichen Tests	Je Komponente ein Verant- wortlicher für Tests, Testda- tenanalyse

Womöglich teure Nachrü- stung beim Auftraggeber not- wendig	Hoch	Beim Testen auf Performan- ce achten und falls nötig Code überarbeiten	Jeweiliger Testverantwortli- cher für eine Komponente achtet nicht nur auf Abstürze etc sondern auch auf Lauf- zeit
Womöglich teure Nachrü- stung beim Auftraggeber not- wendig	Hoch	Vorab Systemanforderungen klären	Dieses Risiko sollte in enger Zusammenarbeit mit dem Auftraggeber überwacht werden
Keine Verbesserung der Arbeitsbedingungen beim Auftraggeber -> Projekt war Ressourcenverschwendung	Normal	Genügendes Requirements- Engineering, falls Endnutzer nicht gleich Auftraggeber auch mit diesen und anderen Stakeholdern sprechen	Kommunikationsbeauftragter stellt bereits Vorab Prototy- pen und Konzepte auch den Endnutzern vor

Reaktion	
Sofortmaßnahmen nachdem das Risiko eingetreten ist	
9	10
Restlichen Zeitplan überdenken, Disziplinarmaßnahmen überlegen, Arbeitstempo erhöhen	
Tiefergehende Recherche, Berichtigung der Fehler, Anpassung des Zeitplans	
Überstunden schieben, beim Prüfungsamt Bescheid sagen, Projektziele überprüfen	
Streitschlichtungsmaßnahmen anwenden, Mediator konsultieren, Projektleiter informieren	
Falls nicht zu spät Umlanzen, schauen was man erhalten kann und was man ändern muss, Überlegen ob Änderung sinnvoll oder Neuanfang besser	
Falls nicht zu spät Umlanzen, schauen was man erhalten kann und was man ändern muss, Überlegen ob Änderung sinnvoll oder Neuanfang besser	
Verwendete Algorithmen analysieren, Schwächen identifizieren, betroffene Teile ersetzen	
Refactoring des Quelltextes und anschließende Kontrolle durch anderes Teammitglied	

Refactoring des Quelltextes
und anschließende Kontrolle
durch anderes Teammitglied

Refactoring des Quelltextes
und anschließende Kontrolle
durch anderes Teammitglied

Weitere Testdaten generie-
ren oder ggf. vom Auftrag-
geber anfragen
Testdaten überdenken,
neue, nützliche Tests Konzi-
pieren, falls nötig Tests mit
neuen Daten wiederholen

Nachreichen von Fixes und
Patches falls möglich

Auftraggeber nachträglich
detaillierte Systemanforde-
rungen zukommen lassen
um ggf. Nachrüsten zu er-
möglichen

Nachreichen von Fixes und
Patches

Bewertung
Normal
Hoch
Kritisch

Bewertung
Normale Risiken haben Auswirkungen, die das Projekt aus eigener Kraft kompensieren Kann. Gegenmaßnahmen (falls erforderlich) sind bereits umgesetzt und für das Restrisiko sind ausreichende Zeit und Geldreserven allokiert. Die Auswirkungen auf die Projektziele sind gering.
Hohe Risiken haben erhebliche Auswirkungen auf die Projektziele. Obwohl die Projektdurchführung weiterhin möglich ist, braucht das Projekt beim Eintritt dieses Risikos Hilfe von außen.
Kritische Risiken verhindern faktisch den Projekterfolg. Weder das Projekt noch die Sponsoren können die Auswirkungen kompensieren und ein Abbruch des Projekts ist sehr wahrscheinlich.