

Risikomanagement

Eintrittswahrscheinlichkeit
1: unwahrscheinlich
2: möglich
3: sehr wahrscheinlich

Auswirkungen:
1: geringe Auswirkungen
2: mittlere Auswirkungen
3: große Auswirkungen

Nummer	Kategorie	Beschreibung	Risikovermeidung	Risikobehandlung	Eintrittswahrscheinlichkeit								Auswirkungen								Bewertung							
					Sprint 1	Sprint 2.1	Sprint 2.2	Sprint 3.1	Sprint 3.2	Sprint 3.3	Sprint 4.1	Sprint 4.2	Sprint 1	Sprint 2.1	Sprint 2.2	Sprint 3.1	Sprint 3.2	Sprint 3.3	Sprint 4.1	Sprint 4.2	Sprint 1	Sprint 2	Sprint 2.2	Sprint 3.1	Sprint 3.2	Sprint 3.3	Sprint 4.1	Sprint 4.2
4	Zeitplan	Terminüberschreitung	Das Wichtigste zuerst entwickeln, Puffer einplanen	Features weglassen	2	2	3	3	3				3	3	3	3	3				6	6	9	9	9			
10	Personal	Team arbeitet aneinander vorbei	Regelmäßige Meetings	Regelmäßige Meetings	2	3	3	3	3				2	3	3	3	3				4	9	9	9	9			
3	Technisch	Inkompatibilität Tools, Merge-Probleme	Kompatibilität vor Verwendung prüfen	Alternative Software, Ausweichstrategie	2	3	3	3	2				3	3	3	3	3				6	9	9	9	6			
5	Technisch	Datenverlust	Daten redundant speichern, Backups	Wenn möglich Backups zurückspielen	2	2	2	3	2				3	3	3	3	3				6	6	6	9	6			
8	Personal	Teammitglied fällt kurzfristig aus	Wissen teilen	Aufgaben neu verteilen	3	3	3	3	3				2	2	2	2	2				6	6	6	6	6			
1	Technisch	Unerfahrenheit in Technologien	Seminare, gegenseitiger Austausch	In Technologien einarbeiten, Seminare	3	2	2	2	2				3	2	2	2	2				9	4	4	4	4			
6	Technisch	Technologie besitzt erwartetes Features nicht	Prototyp	Andere Technologie verwenden	2	3	3	3	2				3	2	2	2	2				6	6	6	6	4			
7	Kunde	Änderung der Anforderungen	Agile Vorgehensweise	Änderung der Planung	3	3	3	3	2				2	2	2	2	2				6	6	6	6	4			
16	Technisch	Schlechte Qualität	Testgetriebene Entwicklung, Reviews	Reviews, Tests, Fehler beheben	1	1	2	2	2				2	2	2	2	2				2	2	4	4	4			
17	Zeitplan	Zu spätes Testen	Testgetriebene Entwicklung	Zeitplan anpassen, ggf. Features weglassen	1	1	2	2	2				2	2	2	2	2				2	2	4	4	4			
2	Anforderungen	Spielkonzept fehlerhaft	Spielkonzept auf Papier austesten	Spielkonzept anpassen, Workaround	2	1	1	1	1				3	3	3	3	3				6	3	3	3	3			
12	Kunde	Auftraggeber kooperiert nicht	Auftraggeber in Entwicklung einbeziehen	Kooperation einfordern	1	1	1	1	1				3	3	3	3	3				3	3	3	3	3			
13	Technisch	Apple lehnt App ab	Kriterien kennen und einhalten https://developer.apple.com/	App anpassen	1	1	1	1	1				3	3	3	3	3				3	3	3	3	3			
14	Personal	Teammitglied fällt komplett aus	Wissen teilen, keine "Gurus", Dokumentation	Aufgaben neu verteilen	1	1	1	1	1				3	3	3	3	3				3	3	3	3	3			
9	Anforderungen	Aufgabenstellung ist widersprüchlich	Aufgabenstellung eindeutig formulieren	Aufgabenstellung mit Auftraggeber präzisieren	2	2	2	2	1				2	2	2	2	2				4	4	4	4	2			
15	Personal	Kompetenzkonflikte im Team	Aufgaben klar abgrenzen	Kommunikation der beteiligten Teammitglieder	3	3	3	2	2				1	1	1	1	1				3	3	3	2	2			
18	Kunde	Zu hohe Erwartungen	Umsetzbarkeit prüfen	Anforderungen ggf. ablehnen	2	2	2	2	2				1	1	1	1	1				2	2	2	2	2			
19	Personal	Wissen wird nicht geteilt	Wissen teilen, z.B. Dokumentation in Wiki	Seminare	2	2	2	2	1				1	2	2	2	2				2	4	4	4	2			
11	Anforderungen	Design nicht umsetzbar	Prototyp	Design anpassen	1	1	1	1	-	-	-	-	3	3	3	3	-	-	-	-	3	3	3	3	-	-	-	-

Bemerkungen:

In Sprint 1 ist **Risiko Nr. 3** eingetreten: Es gibt kein kostenloses Test Coverage Tool für MonoDevelop unter Mac OS. Deshalb muss die Testabdeckung von einem Teammitglied unter Windows ermittelt werden.

In Sprint 1 ist **Risiko Nr. 6** eingetreten: Entgegen der Erwartungen arbeiten Unity und das Apple Game Center nicht zusammen. Mit der in Unity bereits vorhandenen Networking-Funktion konnte jedoch ein guter Ersatz gefunden werden.

In Sprint 1 ist **Risiko Nr. 10** eingetreten: Es wurden Grafiken mit zwei unterschiedlichen Styles entworfen (Cyber vs. Ritter). Das Team hat entschieden, zunächst nur den Cyber-Style zu verwenden.

In Sprint 1 wurde erkannt, dass die Eintrittswahrscheinlichkeit von **Risiko Nr. 1** nicht so hoch ist, wie angenommen. Durch die Tutorials und die Seminare konnte sich das Team gut in Unity einarbeiten.

In Sprint 2.1 konnten nicht alle Features umgesetzt werden, die geplant waren. (**Risiko Nr. 4**)

In Sprint 2.1 wurde festgestellt, dass es Codeteile gibt, die nicht sinnvoll durch Unittests getestet werden können. Stattdessen sollen nun Integrationstests durchgeführt werden. (**Risiko Nr. 16 und 17**)

In Sprint 2.2 gabe es ein Problem mit dem Mergen der einzelnen Branches in den master-Branch. Da Unity die Designer-Dateien in einem nicht nachvollziehbaren Format abspeichert, war das Mergen zunächst nicht möglich. Die Änderungen wurden daher Schritt-für-Schritt von Hand in den master-Branch integriert. Bis eine bessere Lösung gefunden ist, dürfen Änderungen nur sequentiell durchgeführt werden, um Merge-Konflikte zu vermeiden. (**Risiko Nr. 3 und 5**)

In Sprint 2.2 konnten nicht alle geplanten Issues fertiggestellt werden. (**Risiko Nr. 4**)

In Sprint 3.1 wurde das Merge-Problem gelöst. Mithilfe von zwei Tools ist es jetzt möglich, die Designer-Dateien von Unity sinnvoll zu mergen. (**Risiko Nr. 3 und 5**)