



UNIVERSITÄT
LEIPZIG

Softwaretechnikpraktikum

Risikoanalyse

Gruppe: nw19a

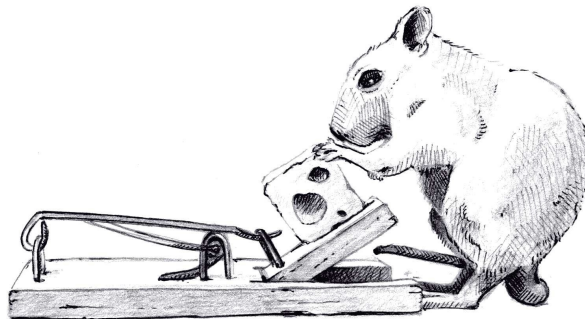
Mitglieder: Thomas Pause, Sabine Lorus, Arik Korte,
Martin George, Josephine Annabelle Lange,
Esther Prause, Anh Kiet Nguyen, Bärbel Hanle

Verantwortlich: Esther Prause

Betreuer: Dr. Nicolas Wieseke

Tutor: Martin Frühauf

Abgabedatum: 04.11.2019



Stand: 4. November 2019

1 Einleitung

Im Zuge der Planung und Abarbeitung eines Softwareentwicklungsprojektes kann eine Risikoanalyse das Entwicklerteam dabei unterstützen, menschlichen, planerischen und technischen Risiken vorzubeugen oder auftretende Probleme mit geeigneten Gegenmaßnahmen zu behandeln, so dass sie das Gesamtziel nicht gefährden. Für eine konsequente Einbeziehung der erarbeiteten Präventivmaßnahmen und die eng getaktete Überwachung der Risiken wählen wir SCRUM-Methoden.

2 Präventivmaßnahmen

Im Sinne des **agilen Arbeitens** differenzieren wir Teilaufgaben und finales Projektziel laufend aus. Die Arbeitstreffen werden mitprotokolliert; alle Beteiligten haben Zugang zur **Dokumentation**. Indem wir sämtliche internen und externen Diskussionen und Entscheidungen schriftlich festhalten, schaffen wir Planungssicherheit und sorgen dafür, dass unsere Arbeitsschritte stets mit den Vorstellungen des Auftraggebers in Einklang stehen.

In der Rollenvergabe achten wir auf eine faire Verteilung von Arbeitspensum und Verantwortung. Um den besonderen Druck, der auf der Projektleitung lastet, abzubauen, arbeiten wir mit einer **Doppelspitze**. So können sich die Kompetenzen zweier Menschen ergänzen und die restlichen Teammitglieder haben zwei Ansprechpartner in der Projektleitung. Themen und Ablauf von Teammeetings werden von einem im Rotationssystem besetzten **Moderatorenpaar** vorbereitet. Dadurch macht jeder einmal die Erfahrung, ein Treffen zu leiten.

Ein bez. der Projektleitung externer **Mediator** hilft, menschliche Konflikte zu lösen und individuelle Stressoren frühzeitig zu erkennen. Neben dem Kultivieren einer konstruktiven, wertungsfreien Diskussionskultur soll mit dieser Rolle verhindert werden, dass sich Teammitglieder isoliert oder abgehängt fühlen. Generell legen wir Wert auf psychosoziale Faktoren in der Teambildung und eine von psychologischer Sicherheit geprägte **Gruppenatmosphäre**, die das Team bestärkt und motiviert. Beginnend bei unserer umfangreichen Vorstellungsrunde üben wir fortlaufend, persönliche **Grenzen** zu **kommunizieren** und zu respektieren.

Für den Erfolg der Umsetzungsphase sehen wir eine gründliche **Recherche** von Tools und Frameworks vor. Darauf aufbauend verteilen wir Lernaufträge, möglichst unter Berücksichtigung persönlicher Präferenzen. Um den technischen **Kenntnisstand** der Teammitglieder anzuheben, organisieren wir als Ergänzung zum unerlässlichen individuellen Aufbereiten gemeinsames Lernen und gruppenweises Durcharbeiten von Tutorials. Mit **Pairprogramming** gleichen wir fehlende Programmiererfahrungen aus. Programmierpartner mit unterschiedlichem Wissensstand profitieren nicht nur fachlich voneinander, sondern lernen sich auch persönlich besser kennen, was das Team als Ganzes stärkt.

Zeitlich planen wir das Projekt nicht zu eng. Um Überlastungen vorzubeugen wollen wir es jedem Teammitglied ermöglichen, eine **Auszeit** im Sinne eines Urlaubs vom Projekt zu nehmen. Außerdem bauen wir **Pufferzeiten** ein, um Reaktionsspielraum für unvorhergesehene Probleme und Verzögerungen zu haben.

Zur Qualitätssicherung führen wir regelmäßig **Codereviews** durch. Dabei sucht man nach Fehlern/Mängeln im Code oder versucht die Qualität des Codes zu verbessern. Mit geeigneten **Testing-Frameworks** und **CI** sorgen wir für stabile Releases und Sicherheit in der Entwicklung.

3 Risikofaktoren und Gegenmaßnahmen

3.1 Konflikte im Team (EW: 1-2, S: 2-3)

Die Anforderungen und der Stress, welche während eines solchen Projektes leicht aufkommen, führen unter Umständen schnell zu Unstimmigkeiten und Konflikten unter den Teammitgliedern. Die damit

einhergehenden Auseinandersetzungen können viel Zeit und Motivation für alle Beteiligten kosten.

Gegenmaßnahmen: Klärende Gespräche mit der Mediatorin; falls nötig, werden Konfliktparteien aufgabentechnisch getrennt. **Präventivmaßnahmen:** Mediator, Gruppenatmosphäre, Grenzen kommunizieren, Auszeiten.

3.2 Missverständnisse bez. des Projektziels (EW: 1, S: 2-3)

Wenn nicht eindeutig geklärt ist, welche Ziele im Rahmen des Projekts umgesetzt werden sollen, kann es passieren, dass unnötig Energie in die falschen Dinge gesteckt wird und andererseits notwendige Teile des Projektes unerfüllt bleiben. Missverständnisse innerhalb des Teams können dazu führen, dass Teile des Teams in unterschiedliche Richtungen arbeiten.

Gegenmaßnahmen: Wenn schon viel Arbeit in eine „falsche“ Richtung geflossen ist, könnten wir mit dem Betreuer überlegen, ob sich das Projektziel anpassen lässt. **Präventivmaßnahmen:** agiles Vorgehen, Dokumentation, gut durchdachtes Lastenheft.

3.3 Ausfall eines Teammitglieds (EW: 3, S: 1-3)

Krankheiten, familiäre Gründe etc. können dazu führen, dass es zu personellen Ausfällen kommt. Im Extremfall fallen Teammitglieder sogar dauerhaft aus. Solche plötzlichen Ausfälle bedeuten eine Mehrbelastung für den Rest des Teams und gefährden den Zeitplan.

Gegenmaßnahmen: Vorhersehbare Ausfälle sollten rechtzeitig angekündigt werden, damit betroffene Aufgaben umverteilt werden können. Beim dauerhaften Ausfall strukturiert die Projektleitung um und passt die Planung an. **Präventivmaßnahmen:** Puffer, Doppelspitze, Auszeiten.

3.4 Zeitliche Belastung und Stress (EW: 3, S: 1-3)

Durch zusätzliche Belastungen wie Studium, Nebenjob, Familie kann es dazu kommen, dass nicht alle Teammitglieder ausreichend Zeit haben, sich den Aufgaben des Projekts zu widmen. Durch Planungsfehler kann es passieren, dass Aufgaben unter Zeitdruck fertig gestellt werden müssen, worunter dann die Qualität leidet.

Gegenmaßnahmen: Wenn absehbar ist, dass eine übernommene Aufgabe nicht rechtzeitig fertig gestellt werden kann, sollten sich die Verantwortlichen an die Projektleitung wenden, um gemeinsam nach Lösungen zu suchen, beispielsweise Unterstützung durch andere Teammitglieder, „abspecken“ der Ansprüche an die entsprechende Abgabe und im Extremfall die Beantragung der Verlängerung der Abgabefrist. **Präventivmaßnahmen:** Puffer, Auszeiten, Grenzen kommunizieren, Codereviews.

3.5 Arbeitsaufwand falsch eingeschätzt (EW: 3, S: 3)

Das Arbeiten an einem Software-Projekt stellt für die allermeisten Mitglieder des Teams ein Novum dar. Möglicherweise werden der Aufwand für das Gesamtprojekt und die benötigte Zeit für die Fertigstellung einzelner Komponenten im Vorfeld falsch bewertet. Dadurch entstehende Probleme bei der Einhaltung des Zeitplans würden die rechtzeitige Fertigstellung des Projekts gefährden.

Gegenmaßnahmen: Zeichnet sich ab, dass Ziele in der verfügbaren Zeit nicht erreicht werden können, müssen Prioritäten neu gesetzt und notfalls auch geplante Features gestrichen werden. **Präventivmaßnahmen:** Puffer, Recherche, Grenzen kommunizieren, Doppelspitze.

3.6 Fehleinschätzung persönlicher Fähigkeiten und Fertigkeiten (EW: 2, S: 2)

Selbsteinschätzung ist prinzipiell keine ganz leichte Aufgabe. Für das Projekt ist eine Unterschätzung eher hilfreich, da ungeahnte Ressourcen freigesetzt werden können. Eine Überschätzung jedoch kann

sehr destruktiv wirken, da u.a. entsprechend der individuellen Fähigkeiten und Fertigkeiten die Aufgaben verteilt werden.

Gegenmaßnahmen: *Regelmäßige Treffen, bei denen man ansprechen kann und soll, wenn man einer Aufgabe nicht gewachsen ist. Die Projektleitung verteilt im Ernstfall die Aufgaben anders.*

Präventivmaßnahmen: *Gruppenatmosphäre, Grenzen kommunizieren, Mediator.*

3.7 Mangelndes Verständnis der Arbeitsmittel (EW: 3, S: 2-3)

Ein unzureichendes Verständnis verwendeter Arbeitsmittel führt für den Einzelnen und das gesamte Team zu erheblichen zeitlichen Verlusten im Workflow, zu einem erhöhten Stresslevel und schlimmstenfalls zum Misserfolg des gesamten Projekts. Zudem birgt es die Gefahr, möglichst optimale Lösungen für Teilprobleme nicht zu finden oder Code unnötig komplex, umständlich, ressourcenintensiv und/oder fehleranfällig zu erstellen.

Gegenmaßnahmen: *Den zu erstellende Recherchebericht ergänzen und Puffer nutzen, um ein verfehltes Teilziel noch termingerecht zu realisieren. Ggf. sind dafür Teilaufgaben neu zu verteilen. Das präzise Formulieren von Issues und Teilzielen ist unablässig. Bisher umsetzungsrelevante optionale Features sollten neu bewertet werden.* **Präventivmaßnahmen:** *Puffer, Recherche, Dokumentation, Pairprogramming, Codereviews, Kenntnisstand anpassen, Testing.*

3.8 Falsche Frameworks eingesetzt (EW: 1, S: 3)

Die Entscheidung, welche Frameworks verwendet werden sollen, fällt relativ früh im Projektverlauf und kann nicht zurückgenommen werden, ohne große Teile bereits verrichteter Arbeit zu verlieren. Wird zu spät bemerkt, dass die gewählten Tools sich für eine Implementierung im Sinne des Projektziels nicht eignen, ist das gesamte Projekt gefährdet.

Gegenmaßnahmen: *Recherche ausweiten oder Zurückgreifen auf bereits diskutierte Alternativen. Klare Vorstellung entwickeln, welche Eigenschaften in Frage kommende Frameworks haben müssen, um das Projekt erfolgreich abzuschließen. Problemanalyse in Evaluation mit aufnehmen.*

Präventivmaßnahmen: *Recherche, Kenntnisstand anpassen.*

3.9 Mangelndes Hintergrundwissen (EW: 1, S: 2)

Eine wichtige Ressource für dieses Projekt ist das Verständnis mathematischer und physikalischer Zusammenhänge. Es kann passieren, dass in der Umsetzung ein Punkt erreicht wird, an dem vorhandenes Fachwissen nicht mehr ausreicht und die Aneignung von neuem Wissen das Projekt verzögert.

Gegenmaßnahmen: *Die Arbeit an bestimmten Komponenten (etwa User-Interface) wird vorgezogen, sodass kein Leerlauf entsteht, während sich ein Teil des Teams mit der Aufarbeitung der Wissenslücken beschäftigt.* **Präventivmaßnahmen:** *Recherche, Kenntnisstand anpassen.*

3.10 Wissensgefälle (EW: 3, S: 2)

Aufgrund heterogener Vorerfahrungen der Teammitglieder bestehen zu Projektbeginn Unterschiede im Verständnis der verwendeten Technologien und Methoden. Dies kann zu erhöhten Arbeitsbelastungen, Stress, Konflikten, Verzögerungen und Qualitätsverlust führen.

Gegenmaßnahmen: *Kenntnisstand anpassen.* **Präventivmaßnahmen:** *Kontinuierliches Überprüfen und Anpassen der Aufgabenverteilung, Recherche, Pairprogramming.*

EW := Eintrittswahrscheinlichkeit; S := Schaden; (Skala von 1 für gering bis 3 für hoch)