Projekthandbuch

Knowledge Base

Dokumentverantwortliche

Max Becker

Mai Chau Nguyen

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Änderungsverzeichnis

Version	Stand	Änderungsbeschreibung
1.0	24.05.2016	Erstellung des Dokuments.
2.0	31.05.2016	Ergänzung des Referenzverzeichnisses
3.0	10.06.2016	Ergänzung der Projektinfrastruktur und Logbuch
4.0	23.06.2016	Einarbeitung Feedback

Referenzverzeichnis

	Name	**Kurzbeschreibung**	**Ort**
[1]	Kundeninformationen	Daten zum Kunden und Projektrahmen	https://github.com/kanbanana/knowledgebase/blob/develop/docs/administration/Kundeninformationen.pdf
[2]	Risikokatalog	Beschreibung, Überwachung und Verantwortlichkeiten der Risiken	https://github.com/kanbanana/knowledgebase/blob/develop/docs/documentation/quality_management/Risikokatalog.pdf
[3]	Deployment-Konfiguration	Stellt das Deployment für die Anwendung dar	https://github.com/kanbanana/knowledgebase/blob/dce6ee7deaed83148a8394e201fb7f570f237529/docs/presentation/jf_13-06_deployment.pdf
[4]	Anforderungsdokument	Beinhaltet die Systemanforderungen vom Kunden, sowohl FA und NFA	https://github.com/kanbanana/knowledgebase/blob/develop/docs/documentation/requirement/Anforderungsdokument.docx
[5]	Architekturdokument mit Schnittstellenbeschreibungen	Stellt detailliert die Schnittstellen zwischen Backend- und Frontend dar	https://github.com/kanbanana/knowledgebase/blob/develop/docs/documentation/architecture/Architecture%20Documentation.pdf
[6]	Dokumentation offene Fehler	Beschreibung der verbliebenen nicht behobenen Fehler im Testreport	https://github.com/kanbanana/knowledgebase/blob/develop/docs/documentation/test/testreport.docx

Management Summary

Das Projektmanagement sieht es als eine zentrale Aufgabe an, die Kommunikation unter den Projektmitgliedern zu fördern. Da den Projektmitgliedern kein eigens für das Projekt reservierter Projektraum zur Verfügung steht, entstehen längere Kommunikationswege zwischen den tendenziell autonom arbeitenden Teams. Eine Herausforderung für das Projektmanagement ist es über die Projektspanne die Kommunikation unter den Teams zu fördern.

Die im Vergleich zu realen Projekten kurze Projektdauer legt ein agiles Entwicklungsmodell nahe. Eine Herausforderung liegt darin, trotz der geringen Erfahrung der Projektmanager ein agiles Modell zu entwerfen und umzusetzen.

Ein agiles Vorgehen erfordert häufiges Kundenfeedback. Der Auftraggeber bzw. der Kunde verhält sich passiv, was für die Projektmitglieder mehr Handlungsspielraum in der Realisierung der Anwendung zukommen lässt. Das bedeutet im Gegenzug auch, dass es schwer festzustellen ist, womit der Kunden zufrieden ist.

Der in diesem Projekt beschriebene Prozess hat das Team erfolgreich unterstützt. Bis Projektwoche 5 wurden alle Muss- und Kann-Anforderungen des Kunden [4] umgesetzt. Ab der Woche 5 folgte ein Feature Freeze. Daraufhin begann eine einwöchige Testphase, welche die Fehler im System deutlich reduzierte. Es wurden 54 neue Bugs gefunden und 50 davon behoben. Die vier verbliebenen Bugs wurden als unkritisch eingestuft und konnten aus zeitlichen Gründen nicht mehr behoben werden, genaueres hierzu in [6].

Bezüglich des Prozesses des Risikomanagements, wurde dieser nicht aktiv von den Teammitgliedern „gelebt“. Da besteht Nachbesserungsbedarf, den der Verbesserungsansatz widerspiegelt.

1 Projektkontext

Das Projekt wird im Rahmen der Veranstaltung MSP an der Hochschule Mannheim in Auftrag des Unternehmen NTT DATA [1] verwirklicht.
Im Unternehmen ist eine sogenannte interne Wissenssammlung namens „Knowledge Base“ im Einsatz, in welchem Artikel, Dokumentationen, Fehlerbeschreibungen, etc. projektweise abgelegt werden und nur Projektmitglieder zugänglich sind (siehe Abb.1).

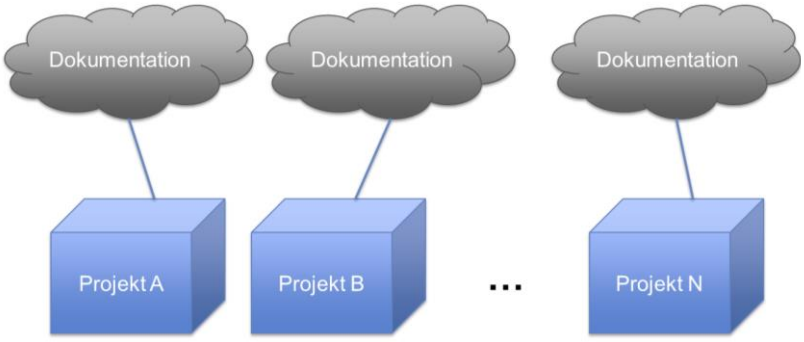

Abbildung 1: IST- Zustand „Knowledge Base“

Einige der Daten sind jedoch auch projektübergreifend interessant. Wie in der Abb. 2 dargestellt, soll aus der aktuell dezentralen Ablage der Daten eine zentrale Ablage entstehen.

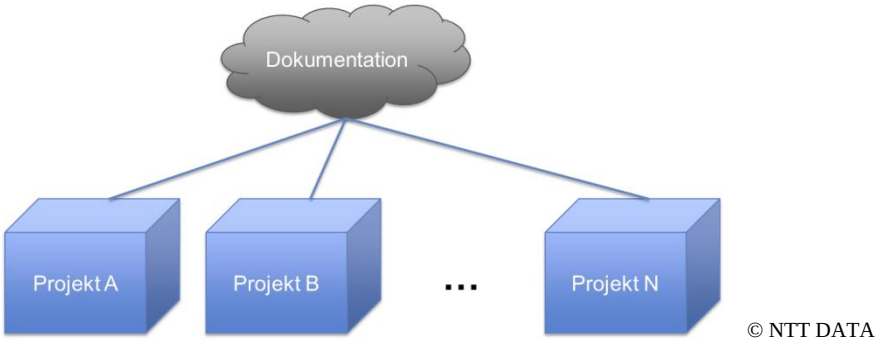

Abbildung 2: SOLL- Zustand „Knowledge Base“

Die Nutzer sollen in der Lage sein, Informationen in der „Knowledge Base“ zu hinterlegen und diese projektübergreifend abzurufen.

2 Entwicklungsmodell

Das Entwicklungsmodell beschreibt den Prozess zur strukturierten Entwicklung eines Produktes. Es beschreibt das Vorgehen aus Managementsicht und ist explizit nicht beschränkt auf die Implementierungsphase, sondern vielmehr die Strukturierung der Aufgabensammlung, -verteilung und –überwachung in einem Prozess. Umfangreiche Aufgaben, auch Epics genannt, werden zunächst von der gesamten Gruppe auf einzelne Teams aufgeteilt (siehe Kapitel 4 Rollenbeschreibung). Teamintern werden diese in kleinere Arbeitspakete aufgeteilt. Arbeiten mehrere Teams an einem Epic, müssen die Ergebnisse bis spätestens zum Sprintende integriert werden. Verantwortlich dafür sind die Teamleiter. Ist die Arbeit des Integrierens sehr komplex, werden Schnittstellenbeauftragte bestimmt. Dies ist beispielsweise für die Integration von Front- und Backend der Fall. Ein Sprint endet damit die integrierten Arbeitsergebnisse dem Kunden zu zeigen und dessen Feedback zu einzuholen.

Der Prozess wurde für dieses Projekt entwickelt und lehnt sich an die Grundideen von Scrum an. Im Folgenden sind die fünf Iterationen dargestellt (siehe Abb. 3), welche jeweils mit einem Meilenstein (M1-M5) abgeschlossen werden. Die Iterationen, auch Sprints genannt, dauern immer sieben Tage.

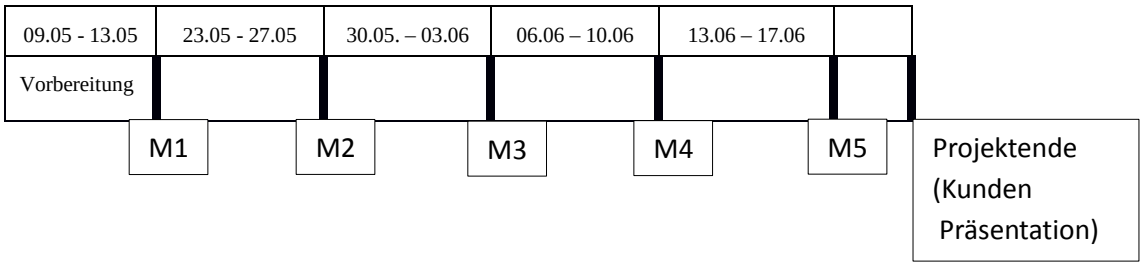

Abbildung 3: Meilensteine des Projektes

Tabelle 1: Datierung der Meilensteine mit Zielvorgabe

Meilenstein	**Zielvorgabe**
M1: 13.05.16	Kommunikations-Tools und PM-Tools eingerichtet; Programmiersprache(n) und grundlegende Technologiewahl getroffen; Wichtigste User-Stories erfasst; Prototyp entwickelt und mit Kunden getestet

(16.05. – 20.05 Pfingstferien)	
M2: 27.05.16	Mit Architekturdokumentation und Implementierung begonnen; Prototyp entwickelt und mit Kunden getestet
M3: 03.06.16	Erste Tests in Testreports dokumentiert; Virtuelle Maschine zum Deployen und Testen der Software erstellt; Schreiben von Installationsanleitung begonnen; Prototyp entwickelt und mit Kunden getestet
M4: 10.06.16	Ende der Implementierungsphase; Architektur finalisiert; Feature Freeze stattgefunden: Angefangene Features in der Gruppe diskutiert und falls die Implementierung noch länger als 2 Tage dauern würde, werden sie verworfen (außer es sind „Muss“-Features, siehe [4]); Prototyp entwickelt und mit Kunden getestet
M5: 17.06.16	Installationsanleitung fertiggestellt; Refactoring durchgeführt; Testen abgeschlossen; Testreporte erstellt; Abnahme mit Kunden durchgeführt

2.1 Vorbereitungswoche (endet mit M1)

In der ersten Projektwoche wird die Basis für die folgenden Projektphasen geschaffen. Jeder Student, außer dem PM-Team, wird einem der drei folgenden Teams zugeteilt:

- **Tooling:** Die Toolauswahl für das Projektmanagement (JIRA), die Kommunikation (Slack) innerhalb der Projektteilnehmer und Dokumenten- und Source-Codeablage (GitHub) erstellen
- **Technik:** Evaluation der Programmiersprache, Entwicklungsumgebung, Frontend- Technologie und Suchengine- Evaluation
- **Anforderungen:** Erstellen aller „high risk“ und „high business value“ User-Stories auf Basis der Workshops im Rahmen der Kick-off Veranstaltung

2.2 Sprintablauf ab Woche 2

2.2.1 Sprint Retrospektive & Planning

- Jeden Montag 9:00 Uhr
- Anwesenheit des gesamten Teams erforderlich
- Moderiert durch das Projektmanagement

Folgend der Ablauf des Treffens:

a) Sprint Retrospektive

- Statusreport jedes Teams durch die Teamleiter:
  Wurden die im letzten Planning geplanten Aufgaben erledigt?
  War es zu viel/ zu wenig Arbeit? Was wurde geschafft?
  Wurden zusätzliche Aufgaben aus dem Backlog in diesem Sprint bearbeitet?
  Gab es Probleme, besondere Vorkommnisse?
- Besprechung des Kundenfeedbacks von letzter Woche
- Wurde das „Sprint-Ziel“ erreicht?
- Feedback zum Entwicklungsmodell einholen

b) Auswahl von User-Stories für diesen Sprint

- Im Kanban- Board befindet sich ein Backlog (siehe Abb. 4), das gefüllt mit priorisierten User Stories ist. Die Aufgabe des Füllens des Backlogs und des Priorisierens liegt beim Anforderungsmanagement-Team. Die gesamten Anforderungen sind ausführlich im Anforderungsdokument zu finden [4].
- Zuteilung durch Projektmanagement in Abstimmung des Teams, welche Stories diesen Sprint erledigt werden
- Angefangen bei den höchst priorisierten User Stories, werden diese aus dem Backlog in „Zur Entwicklung ausgewählt“ im Kanban- Board verschoben bis das Zeitlimit für einen Sprint erreicht ist
- Team entscheidet, wie viele der Stories diesen Sprint erledigt werden können. Hierzu wird gemeinsam der Zeitaufwand der Story grob in Stunden geschätzt.

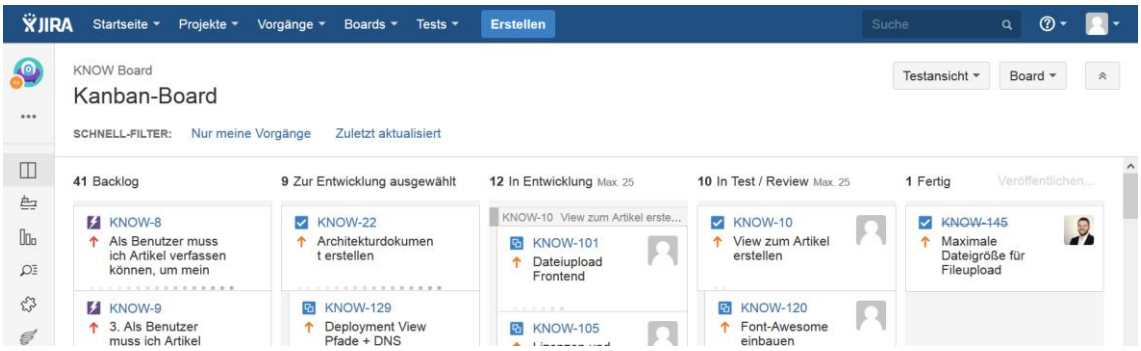

Abbildung 4: Kanban- Board

c) Risiko Management

- Risikomatrix zeigen [2] – Änderungen zur Vorwoche aufzeigen
- Bei Bedarf Aufgaben aus der Risiken ableiten und ins Kanban- Board eintragen

d) Sprint-Ziel und weitere Aufgaben

- Jedes Team wird gefragt, was es diesen Sprint vorhat zu erledigen
- Noch nicht berücksichtigte Aufgaben werden im Kanban- Board für diesen Sprint erstellt für Arbeit, welche nicht durch User-Stories abgebildet wurde, z.B. Projektmanagementbericht erstellen.
- Formulierung eines Sprint-Ziels

Ende des gemeinsamen Treffens

Anmerkung:
Die User Stories sind mit Epics im Kanban- Board gleichzusetzen, welche noch in kleinere Aufgaben und Unteraufgaben aufgeteilt werden müssen. Das erfolgt intern in den Teams am Anfang des Sprints.

2.2.2 Während des Sprints

Während des Sprints hält das Projektmanagement mindestens alle zwei Tage Kontakt zu den Teamleitern. Durch persönliches Nachfragen von Seitens des Projektmanagement-Teams wird überwacht, ob das Sprintziel erreicht wird und ob der Prozess eingehalten wird. Wenn ein Team bereits fertig mit den Aufgaben ist, können weitere Aufgaben aus dem Backlog hinzugenommen werden. Ist ein Team überfordert, wird ein Teil der geplanten Aufgaben für diesen Sprint nicht erledigt. In der kommenden Sprint Retrospektive werden die Gründe dafür aufgeschlüsselt und die Aufgabenverteilung für den folgenden Sprint angepasst. Diese Anpassung moderiert das Projektmanagement.

2.2.3 Sprintende

Jeden Freitag findet ein Kundentermin statt. Es nehmen die Kundenbetreuer und je nach Bedarf die Teamleiter daran teil, die den Schwerpunkt der Präsentation gestalten. Die Aufgabe der Teamleiter ist es bis zum Kundentermin die Arbeit der einzelnen Teams zu integrieren, sodass ein vorzeigbares Ergebnis entsteht. Bevor ein Ergebnis vorzeigbar ist, durchläuft es einen Review/Testing Schritt (siehe Abb. 4). Die Dokumente reviewen die anderen Teammitglieder. Es liegt in der Verantwortung des Testteams das Source-Code Inkrement des Sprints hinreichend zu testen. Das Ergebnis wird dem Kunden präsentiert und sein Feedback in der folgenden Sprint-Retrospektive in der Gruppe besprochen.

3 Projektinfrastruktur

3.1 Aufgabenmanagement

- JIRA mit Kanban- Board
- Tasks sind beschrieben mit
    - Priorität
    - Größe: (Epics – Aufgaben – Unteraufgaben)
    - Bearbeiter
    - Bearbeitungsstand (siehe Abb. 4)

3.2 Dokumenten- und Codeablage

- Remote Git-Repository auf Github
- https://github.com/kanbanana/knowledgebase/
- Arbeiten nach Git-Flow:
    - Jedes Feature ein eigener Branch
    - Feature-Branch trägt den Namen des JIRA-Tasks
    - Bei Fertigstellung des Tasks wird der Feature-Branch in den Develop-Branch gemerged
    - S.a http://danielkummer.github.io/git-flow-cheatsheet/index.de_DE.html
    - Ordnerorganisation ist direkt in Github dokumentiert. Siehe z.B.: https://github.com/kanbanana/knowledgebase/blob/develop/README.md
    - Dokumentenvorlagen sind zu finden unter: https://github.com/kanbanana/knowledgebase/tree/develop/docs/administration/Vorlagen

3.3 Kommunikationstool

- Slack (https://kanbanana.slack.com/)

3.4 Build und Deployment-Infrastruktur

- Deployment auf virtueller Maschine (über Vagrant). Diese startet Docker-Container, s.a. [3]
- Jenkins mit automatischer Testausführung (https://www.danielweidle.de/jenkins/)

- Live-Testsystem für den Kunden:
  - https://danielweidle.de/demo/#/
  - Nutzer: demo
  - PW: demo

3.5 Risikomatrix

- Kategorisierung der identifizierten Risiken (siehe Abb.5)
- Genauere Beschreibung, Überwachung der Risikomatrix und Verantwortlichkeiten der Risiken [2]

4 Rollenbeschreibung

Ab der zweiten Projektwoche werden allen Studenten Rollen zugeteilt. Ein Student kann mehrere Rollen einnehmen. Für jede Rolle wird ein Teamleiter bestimmt. Der Teamleiter ist der direkte Ansprechpartner für das Team dieser Rolle gegenüber den anderen Rollen. Der Teamleiter koordiniert die Arbeit und hat den Überblick, wer aus dem Team gerade an welcher Aufgabe arbeitet und ob das Zeitmanagement innerhalb der Gruppe stimmt. Das Projektmanagement hat dieselben Aufgaben, allerdings auf das gesamte Projekt bezogen.

Bei Meinungsverschiedenheiten würde innerhalb des Teams zunächst der Teamleiter entscheiden. Kann keine Entscheidung getroffen werden, wird das Problem zur Projektleitung eskaliert, wo dann die Entscheidung getroffen wird. Ein Sonderfall ist das Qualitätsmanagement, welches quer zu den anderen Teams steht und alle Rollen, auch die des Projektmanagements überwacht. Es ist allerdings nicht weisungsbefugt. Handlungsanweisungen vom QM werden, falls nicht trivial, in der gesamten Gruppe besprochen und per Mehrheitsentscheid für gültig oder ungültig erklärt.

Die Qualitätsmanager(QM) stehen in der Projekthierarchie (siehe Abb.5) neben dem Projektmanagement. Der Handlungsspielraum der QM beläuft sich die Maßnahmendefinition zur Qualitätssicherung innerhalb des Projektes und die Durchführung der Maßnahmen in Abstimmung der Projektleitung. Da beide Rollen bestrebt sind, den Projekterfolg zu fördern, setzt das eine enge Zusammenarbeit voraus.

Tabelle 2: Rollenbeschreibung

Rolle	**Beschreibung**
Projektmanager	- Entwicklungsmodell erstellen - Korrekte Durchführung des Entwicklungsmodells gewährleisten (vglb. Scrum-Master) - Projektkontrolle durchführen - als Ansprechpartner für Gruppen agieren - Entscheidungsträger - Zeitmanagement - Risikomanagement (mit QM) - Juristische Fragen (Lizenz, Kundenvertrag)

Teamleiter	- Jede folgend genannte Rolle bestimmt einen Teamleiter - Vertritt das eigene Team vor dem PM - Verantwortet die Integration der Ergebnisse des eigenen Teams mit den Ergebnissen der anderen Teams - Einhaltung des Entwicklungs-Prozesses innerhalb des Teams - Trägt die technische Verantwortung für seinen Teilbereich
Kundenbetreuer	- Kundenkontakt herstellen - Ansprechpartner für den Kunden
Architekt	- Architekturentscheidungen treffen - Architekturdokumentation erstellen
Entwickler	- Fronend entwickeln - Backend entwickeln
Schnittstellenbeauftragter	- Je eine Person aus Front- und Backend - Spezifikation und Dokumentation der Schnittstellen erstellen und warten. Zum Dokument, siehe [5] - Frühzeitiges Integrieren von Front- und Backend voran treiben
UI/UX Designer	- GUI Mock ups erstellen - Kunden-Feedback zu Mock-ups einholen und dieses umsetzten
Qualitätsmanager	- Qualitätshandbuch erstellen - Qualitätsmaßnahmen definieren und Handlungsanweisungen geben - Testplan erstellen
Tester	- Tooling-Landschaft für das Testen erstellen - Tests durchführen: -> Integrationstests -> Unit-Tests -> GUI-Tests - Testreports schreiben

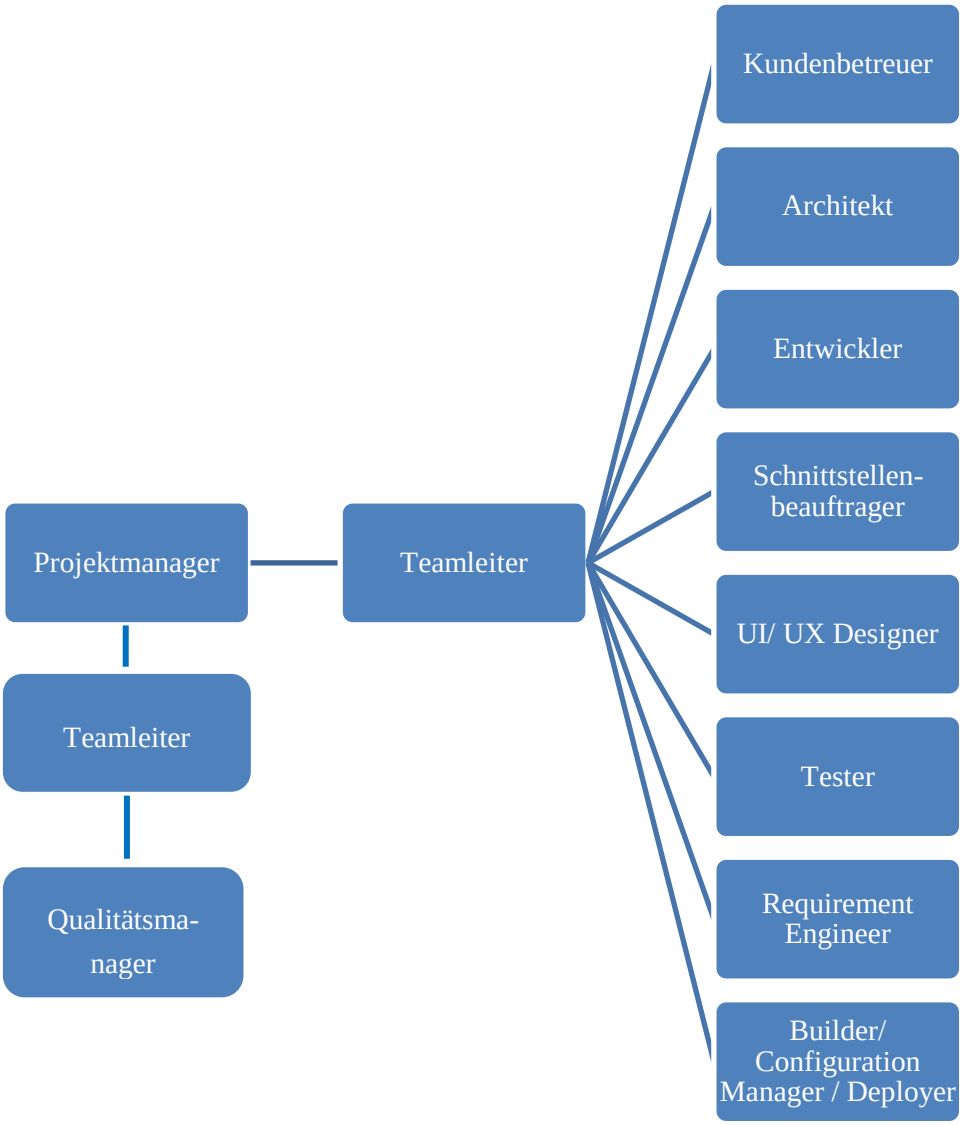

Abbildung 5: Projekthierarchie

5 Risikomanagement

5.1 Prozessbeschreibung

Die initiale Risikoerfassung wurde in der zweiten Iteration in einem Workshop erarbeitet. Jedes einzelne eindeutige identifizierte Risiko obliegt dezentral der Verantwortlichkeit des jeweiligen zugeordneten Teamleiters und übergeordnet dem Projektmanagement. Diese haben dafür Sorge zu tragen während einer Iteration neue Risiken zu erfassen und nach einer Iteration die vorhandenen Risiken zu aktualisieren. Auf Basis des Risikokatalogs werden in der Sprint Planung Aufgaben abgeleitet und umgesetzt.

5.2 Risikomatrix

Bedingt durch die Bewertung der Eintrittswahrscheinlichkeit und Schadenshöhe, ordnet sich jedes Risiko einer der drei Kategorien an. In der Risikomatrix(siehe Abb. 6) ist das farblich gekennzeichnet:

- Grün: kein Handlungsbedarf
- Orange: steht unter Beobachtung
- Rot: akuter Handlungsbedarf

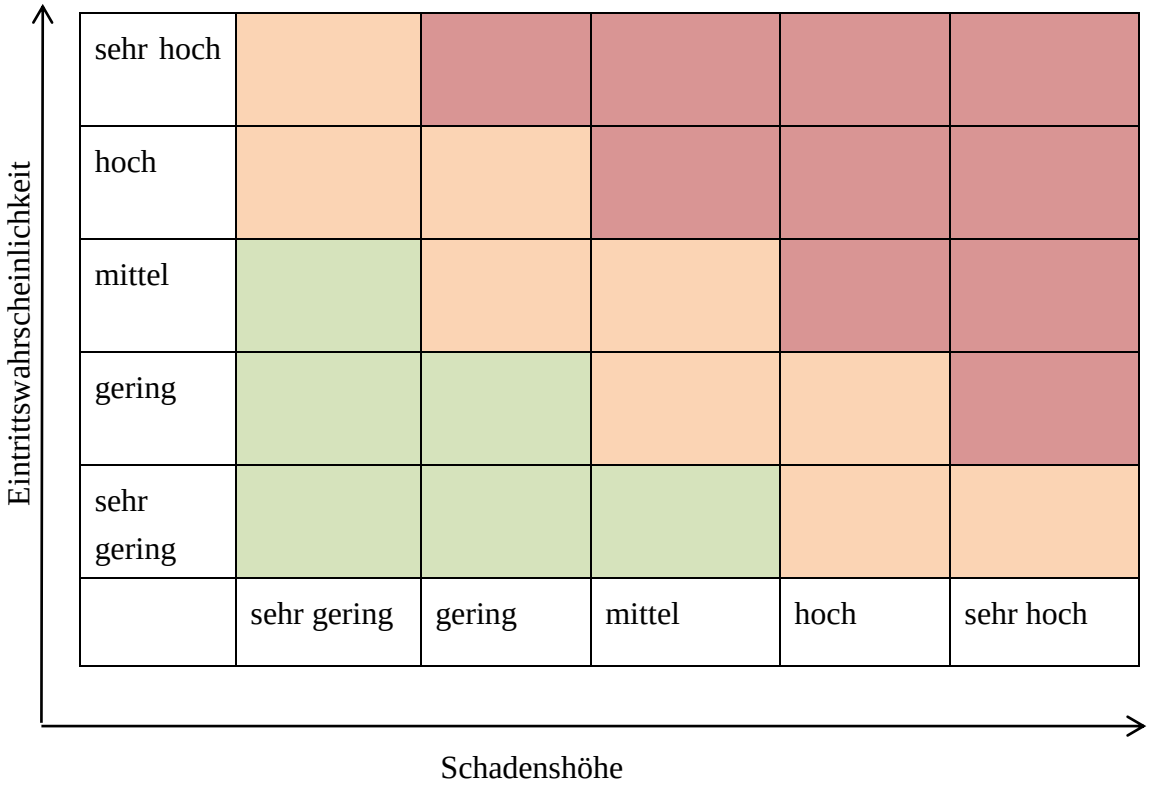

Abbildung 6: Risikomatrix

5.3 Verbesserungsansatz

Der Einsatz eines Dokumentes als Hauptgegenstand des Risikomanagements und für die Kommunikation der Risiken mit den Teamleitern war in diesem Projekt nicht praktikabel. Da dieses nicht präsent im Bewusstsein der Teamleiter war und nicht aktuell gehalten wurde. Falls ein Dokument angelegt wird, welches die Risiken dokumentiert, soll dieses nur zur Dokumentationszwecken und zur Beschreibung von Risikodetails existieren. Darunter fallen die präventiven und korrektiven Maßnahmen für das jeweilige Risiko und dessen Bedeutung für das Projekt.

Bei einem zur Verfügung stehenden Projektraum kann stattdessen ein Board zur Visualisierung einer reduzierten Risikomatrix dienen. Dabei werden nur die Risiken betrachtet, die unter Beobachtung stehen oder aus denen akuter Handlungsbedarf entsteht. Die Risiken werden in dem Board in Stichpunkten beschrieben und beinhalten auch mögliche Folgehandlungen, welche aus den Risiken abzuleiten sind. Die Risikomatrix wird beim Sprint Planning besprochen und aktualisiert.

6 Logbuch

Datum der Änderung,Version	Änderungsbeschreibung
24.05.16, 1.0	Erstellung des Dokuments.
31.05.16, 2.0	Entwicklungsmodell: Statt der Bewertung von Aufgaben mit Punkten, werden diese jetzt in Zeit bewertet. In der Kürze des Projektes ist die Zeitschätzung intuitiver. Die wichtigsten User-Stories werden nicht mehr während des gemeinsamen Meetings in Aufgaben und Unteraufgaben aufgespaltet. Stattdessen werden den Teams größere Arbeitsblöcke zugeteilt und diese teilen im Laufe des Sprints diese in kleinere Aufgaben aufspalten. Grund: Das Meeting dauert sonst für alle zu lange. Einführung des Feature Freezes in M4 Die wichtigsten User-Stories werden nicht durch das Team während des Meetings priorisiert. Stattdessen liegt es in der Verantwortung des Anforderungsteams die Anforderungen in Form von User-Stories priorisiert in das Backlog des Kanban- Boards zu schreiben.
10.06.16, 3.0	Hinzufügen von Projektstrukturplan, Kanban- Board-Beschreibung und Erstellung dieses Logbuchs. Einfügen der Rolle „Schnittstellenbeauftragter“. Entwicklungsmodell: Umbenennung des Montags-Meetings von Sprint Planning & Review in Sprint Planning & Retrospektive (beschreibt besser was wir tun)
23.06.16, 4.0	Ergänzung Management Summary um Betrachtung über Erfolg des Entwicklungsmodelles Entwicklungsmodell: Integrationsbeschreibung innerhalb des Teams Anpassung der Meilensteinbeschreibungen Tätigkeitsbeschreibungen während des Sprints und am Sprintende, um die Kommunikation zwischen Projektmanagement und den in-

	ternen Teams und die Integration nachvollziehen zu können Projektinfrastruktur: Ergänzung der Risikomatrix als Infrastrukturelement Rollenbeschreibung: Definition der Aufgaben der Rolle des Teamleiters Prozessbeschreibung des Risikomanagements hinzugefügt, da sich der Prozess nicht mehr verändert und Teil des Projekthandbuchs ist anstatt des Risikokatalogs Ergänzung des Verbesserungsansatzes des Risikomanagements