

NTT Data



hochschule mannheim



kanbanana

Projektmanagementbericht

Knowledge Base

Dokumentverantwortliche

Max Becker

Mai Chau Nguyen

Inhaltsverzeichnis

Inhaltsverzeichnis	2
Abbildungsverzeichnis.....	3
Tabellenverzeichnis	3
Änderungsverzeichnis	4
Referenzverzeichnis	5
Vorwort.....	6
1 Projektkontext	7
2 Entwicklungsmodell	8
2.1 Vorbereitungswoche (endet mit M1).....	9
2.2 Sprintablauf ab Woche 2	9
2.2.1 Sprint Retrospektive & Planning	9
2.2.2 Kunden-Feedback.....	11
3 Projektinfrastruktur	12
3.1 Aufgabenmanagement	12
3.2 Dokumenten- und Codeablage	12
3.3 Kommunikationstool	12
3.4 Build und Deployment-Infrastruktur	12
4 Rollenbeschreibung	14
5 Logbuch	17

Abbildungsverzeichnis

Abbildung 1: IST- Zustand "Knowledge Base"	7
Abbildung 2: SOLL- Zustand "Knowledge Base"	7
Abbildung 3: Meilensteine des Projektes	8
Abbildung 4: Kanban-Board.....	10

Tabellenverzeichnis

Tabelle 1: Datierung der Meilensteine mit Zielvorgabe	8
Tabelle 2: Rollenbeschreibung	14

Änderungsverzeichnis

Version	Stand	Änderungsbeschreibung
1.0	24.05.2016	Erstellung des Dokuments.
2.0	08.06.2016	Ergänzung des Referenzverzeichnisses
3.0	10.06.2016	Ergänzung der Projektinfrastruktur und Logbuch

Referenzverzeichnis

ID	Name	Kurzbeschreibung	Ort
[1]	Kundeninformationen	Daten zum Kunden und Projekt-rahmen	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/Qualitätsmanagement/Risikokatalog.docx
[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/master/docs/documentation/REQ/Anforderungsdokument.docx
[5]	Architekturdokument mit Schnittstellenbeschreibungen	Stellt detailliert die Schnittstellen zwischen Backend- und Frontend dar	https://github.com/kanbanana/knowledgebase/tree/develop/docs/documentation/architecture

Vorwort

Der Projektmanagementbericht dokumentiert die entscheidenden Schritte des Projektmanagements. Das Ziel ist es, den Professoren eine Übersicht über die getroffenen Entscheidungen und Maßnahmen zu informieren.

Besondere Herausforderungen beim diesem Projekt an das Projektmanagement finden sich zu einem in der Tatsache, dass den Projektmitgliedern kein eigens für das Projekt reservierter Projektraum zur Verfügung steht. Das führt zu einer verminderten Kommunikation unterhalb der Projektteams aufgrund von den längeren Kommunikationswegen und der tendenziell autonomen Arbeitsweise der Teams.

Die im Vergleich zu realen Projekten kurze Projektdauer legt ein agiles Entwicklungsmodell nahe. Die Herausforderung liegt darin, trotz der geringen Erfahrung der Projektmanager ein solches Modell durchzuführen und die Projektplanung so zu gestalten, dass mit den vorhandenen Ressourcen ein Projekterfolg erreicht wird.

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 die tatsächliche Kundenzufriedenheit nicht offensichtlich ist und es durch Lenkung des Kunden zu erreichen gilt.

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).

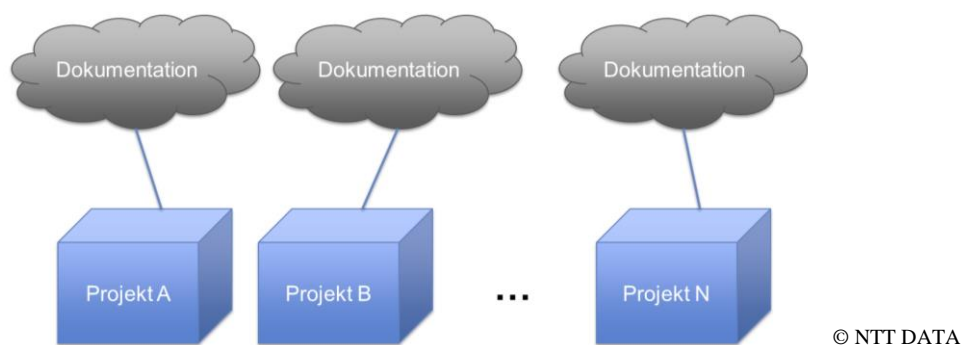


Abbildung 1: IST- Zustand "Knowledge Base"

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.

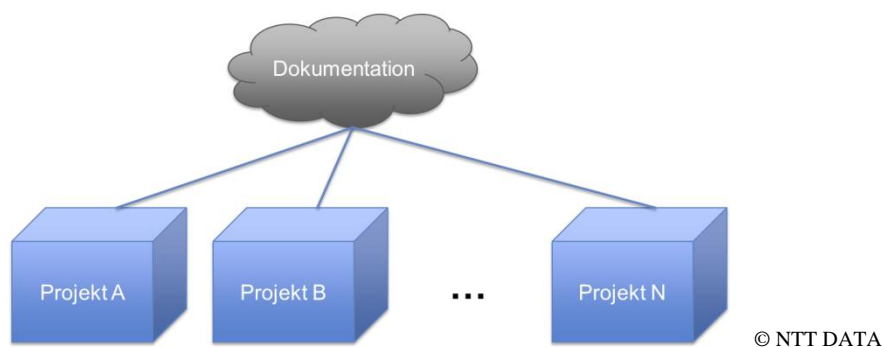


Abbildung 2: SOLL- Zustand "Knowledge Base"

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. Unter den Aufgaben fallen die Arbeitspakete und Maßnahmen, die für die erfolgreiche Projektdurchführung anfallen. Der Prozess wurde für dieses Projekt entwickelt und lehnt sich an die Grundideen von Scrum an. Folgend die 5 Iterationen, welche jeweils mit einem Meilenstein (M1-M5) abgeschlossen werden. Die Iterationen, hier Sprints genannt, dauern immer 7 Tage.

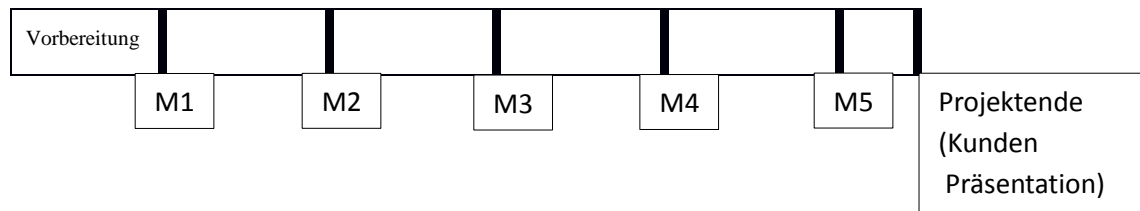


Abbildung 3: Meilensteine des Projektes

Tabelle 1: Datierung der Meilensteine mit Zielvorgabe

Meilenstein	Zielvorgabe
M1: 13.05.16	Kommunikations-Tools und PM-Tools eingerichtet; Programmiersprache und grundlegende Technologiewahl getroffen; Wichtigste User-Stories erfasst
(16.05. - 20.05 Pfingstferien)	
M2: 27.05.16	Mit Architekturdokumentation und Implementierung begonnen
M3: 03.06.16	Erste Tests in Testreports dokumentiert; Virtuelle Maschine zum Deployen und Testen der Software erstellt; Schreiben von Installationsanleitung begonnen
M4: 10.06.	Ende der Implementierungsphase; Architektur finalisiert;

	Featurefreeze: Angefangene Features werden 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])
M5: 17.06.16	Installationsanleitung fertiggestellt; Refactoring durchgeführt; Testen abgeschlossen; Testberichte erstellt

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 Sourcecode-ablage(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:

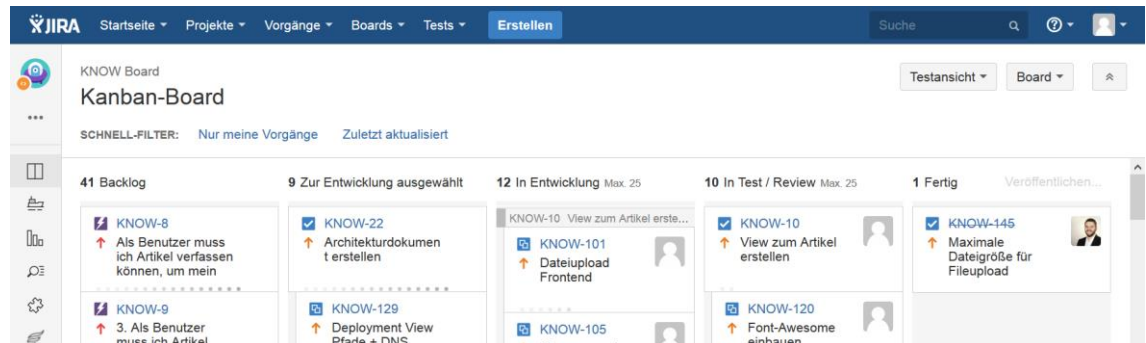
a) Sprint Retrospektive

- 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



- Abbildung 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 Kanbanboard 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.

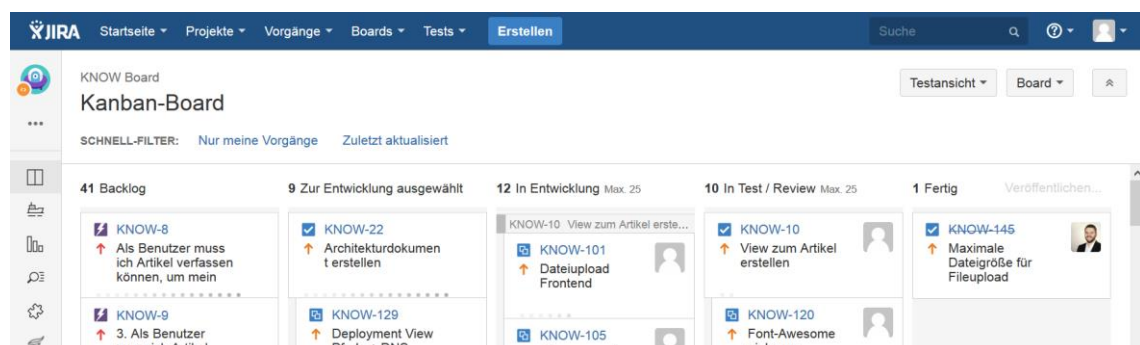


Abbildung 4: Kanban-Board

c) Risiko Management

- Risiko-Matrix zeigen[2] – Änderungen zur Vorwoche aufzeigen
- Bei Bedarf Aufgaben aus der Risiken ableiten und ins Kanbanboard eintragen

d) Sprint-Ziel und weitere Aufgaben

- Jedes Team wird gefragt, was es diesen Sprint vorhat zu erledigen
- Noch nicht berücksichtigte Aufgaben werden im Kanbanboard 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 Kunden-Feedback

- Jeden Freitag findet ein Kundentermin statt
- Ziel: am Ende jeder Iteration vorzeigbare Ergebnisse dem Kunden präsentieren und sein Feedback einholen

3 Projektinfrastruktur

3.1 Aufgabenmanagement

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

3.2 Dokumenten- und Codeablage

- Remote Git-Repository auf Github
- <https://github.com/kanbanana/knowledgebase/>
- Arbeiten nach Git-Flow:
 - o Jedes Feature ein eigener Branch
 - o Feature-Branch trägt den Namen des JIRA-Tasks
 - o Bei Fertigstellung des Tasks wird der Feature-Branch in den Develop-Branch gemerged
 - o S.a http://danielkummer.github.io/git-flow-cheatsheet/index.de_DE.html
- Ordnerorganisation in Github:
 - o Develop – SourceCode
 - o Docs – Fließtext Dokumentation
 - administration – Vom PM-erstellte Dokumente
 - documentation – Alle weiteren Dokumente
 - presentation - Offizielle Vorträge
 - o Provision – Deployment Files

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:
 - o <https://danielweidle.de/demo/#/>
 - o Nutzer: demo
 - o PW: demo

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 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 anstatt zu behindern, setzt das eine enge Zusammenarbeit voraus.

Tabelle 2: Rollenbeschreibung

Rolle	Beschreibung
Projektmanager	<ul style="list-style-type: none">- 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)

Kundenbetreuer	<ul style="list-style-type: none">- Kundenkontakt herstellen- Ansprechpartner für den Kunden
Architekt	<ul style="list-style-type: none">- Architekturentscheidungen treffen- Architekturdokumentation erstellen
Entwickler	<ul style="list-style-type: none">- Frontend entwickeln- Backend entwickeln
Schnittstellenbeauftragter	<ul style="list-style-type: none">- 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	<ul style="list-style-type: none">- Mock ups erstellen- Kunden-Feedback zu Mock-ups einholen und dieses umsetzen
Qualitätsmanager	<ul style="list-style-type: none">- Qualitätshandbuch erstellen- Qualitätsmaßnahmen definieren und Handlungsanweisungen geben- Testplan erstellen
Tester	<ul style="list-style-type: none">- Tooling-Landschaft für das Testen erstellen- Tests durchführen:<ul style="list-style-type: none">-> Integrationstests-> Unit-Tests-> GUI-Tests- Erfüllung der Abnahmetests des Kunden- Testreports schreiben
Requirement Engineer	<ul style="list-style-type: none">- Anforderungsanalyse durchführen- User Stories finden und schreiben- Anforderungskontrolle durchführen- Anforderungsdokument erstellen und pflegen
Builder/ Configuration Manager / Deployer	<ul style="list-style-type: none">- Installationsanleitung erstellen- Entwicklungsumgebung erstellen (VM)

	<ul style="list-style-type: none">- Tool Chain festlegen- Automatisierte Build- und Testumgebung aufsetzen und warten
--	----------------------------------------------------------------------------------------------------------------------------------------------

5 Logbuch

Datum der Änderung	Änderungsbeschreibung
24.05.16	Erstellung des Dokuments.
31.05.16	<p>Entwicklungsmodell:</p> <p>Statt der Bewertung von Aufgaben mit Punkten, werden diese jetzt in Zeit bewertet. In der Kürze des Projektes ist die Zeitschätzung intuitiver.</p> <p>Die wichtigsten User-Stories werden nicht mehr während des gemeinsamen Meetings in Aufgaben und Unteraufgaben aufgespalten. 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.</p> <p>Einführung des Featurefreezes in M4</p> <p>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.</p>
10.06.16	<p>Hinzufügen von Projektstrukturplan, Kanban-Board-Beschreibung und Erstellung dieses Logbuchs.</p> <p>Einfügen der Rolle „Schnittstellenbeauftragter“.</p> <p>Entwicklungsmodell:</p> <p>Umbenennung des Montags-Meetings von Sprint Planning & Review in Sprint Planning & Retrospektive (beschreibt besser was wir tun)</p>