

Wer wird Millionär Projektmanagementplan

Lei Chen

Marcel Khan

Jendrik Wenke

17. Oktober 2018

Inhaltsverzeichnis

1	Projektorganisation	2
1.1	Rollen & Zuständigkeiten	2
1.2	Projektstrukturplan / Sprintbacklog	3
2	Projektführung	4
2.1	Rahmen- und Projektplan	4
2.1.1	Überblick über den gesamten Projektzeitraum	4
2.1.2	Projektplan für die Initialisierungsphase und den ersten Sprint	5
2.2	Projektkontrolle und Projektsteuerung	5
3	Projektunterstützung	6
3.1	Tools für die Entwicklung	6
3.2	Konfigurationsmanagement	7
3.3	Tools zur Projektkommunikation und zum Dokumentenaustausch	7
4	Qualitäts- und Testplan	7
4.1	Definition of Ready und Definition of Done	7
4.2	Testdesign & Testautomatisierung	8
4.3	Testfälle	8
5	Anhänge	8
5.1	Abkürzungen	8

Rev.	Datum	Autor	Bemerkungen	Status
1	17.01.2018	LC, MK, JW	Erste Version	fertiggestellt



1 Projektorganisation

1.1 Rollen & Zuständigkeiten

Name	Lei Chen, Marcel Khan, Jendrik Wenke
Rolle	Entwickler
Aufgaben	Erstellung des Sprint-Backlog aus dem Product-Backlog, Abarbeiten der Tasks
Verantwortlichkeiten	Selbstorganisation, regelmäßiges Scrum-Meeting,
Befugnisse	Die Entwickler dürfen selbst entscheiden, welche Pakete aus dem Sprintbacklog sie wann in welcher Reihenfolge erledigen, sofern nicht Abhängigkeiten bestehen.
Name	Marcel Khan
Rolle	Product-Owner
Aufgaben	Federführend beim Erstellen und Managen des Product-Backlog, entwickelt die Produktvision, vertritt das Team vor dem Auftraggeber und vermittelt zwischen beiden
Verantwortlichkeiten	Kommunikation der Produktvision ans Team und an den Auftraggeber, steht für Rückfragen des Teams zur Verfügung, verantwortlich für den Erfolg des Projekts
Befugnisse	darf Ergebnisse eines Sprints annehmen oder ablehnen, Nachbesserungen oder Änderungen einfordern sowie Pakete priorisieren
Name	Jendrik Wenke
Rolle	Scrum-Master
Aufgaben	beseitigt Hindernisse, moderiert Scrum-Meetings, hat Aktualität von Scrum-Artefakten im Blick
Verantwortlichkeiten	Verantwortung für Scrum-Prozess und dessen richtige Implementierung
Befugnisse	darf ein Team-Meeting abbrechen oder den Project Owner von einem Team-Meeting ausschließen, wenn nötig um den Teamfrieden zu wahren.

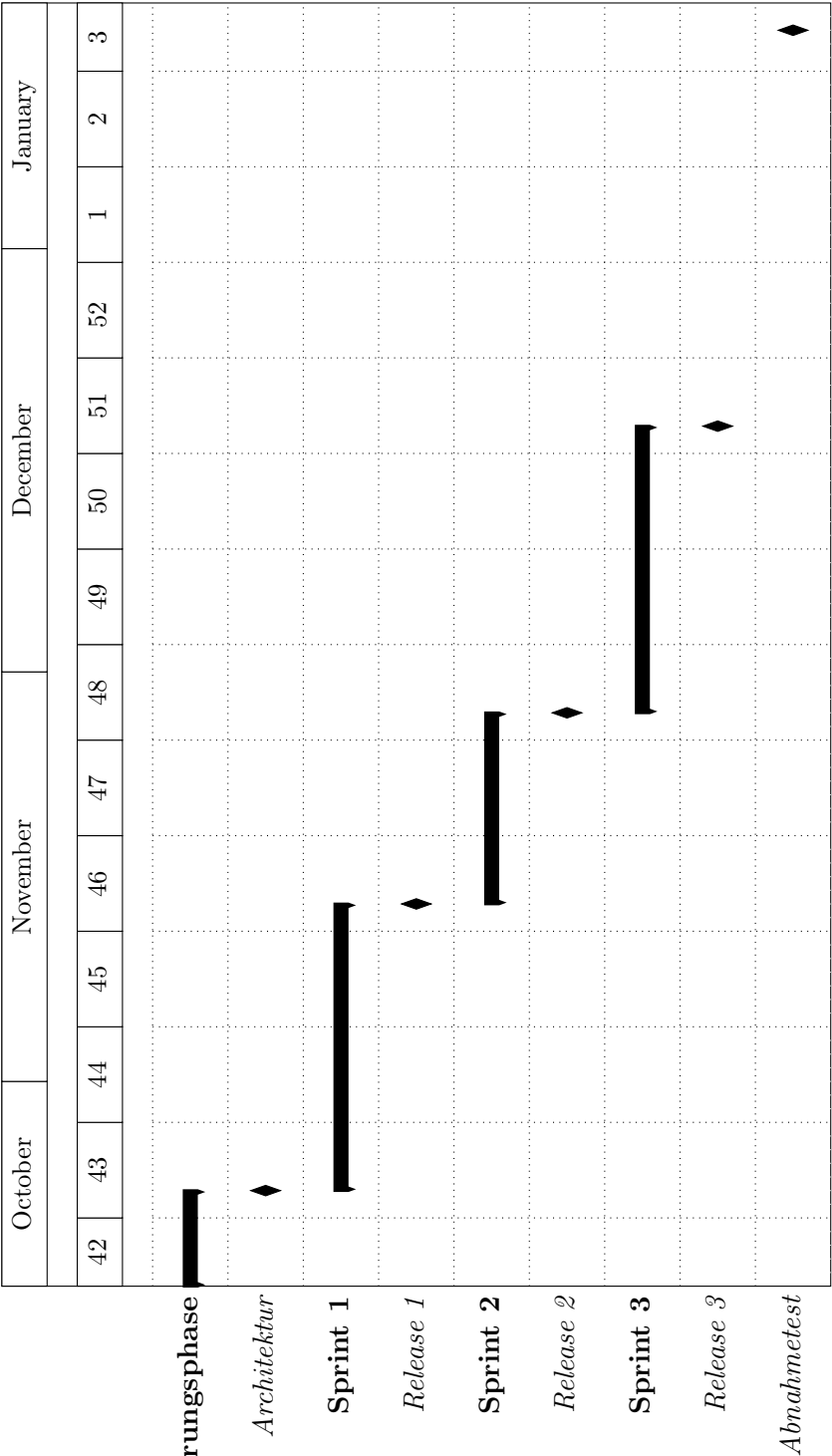


1.2 Projektstrukturplan / Sprintbacklog

Für eine genauere Zeitplanung dient das Gantt-Diagramm in Kapitel Überblick über den gesamten Projektzeitraum.

- Verstehen des Datenformats: Damit mit den Datensätzen gearbeitet werden kann, muss jeder die Struktur und die Konzepte dahinter (Entity, Property, Claim etc.) verstehen.
- Generierung von Wissensfragen: Als erster Schritt sollen Wissensfragen zu Themen generiert werden, die einfach aus den Daten herzuleiten sind.
 - Sinnvolle Antworten
 - Ein Fragentyp pro Person

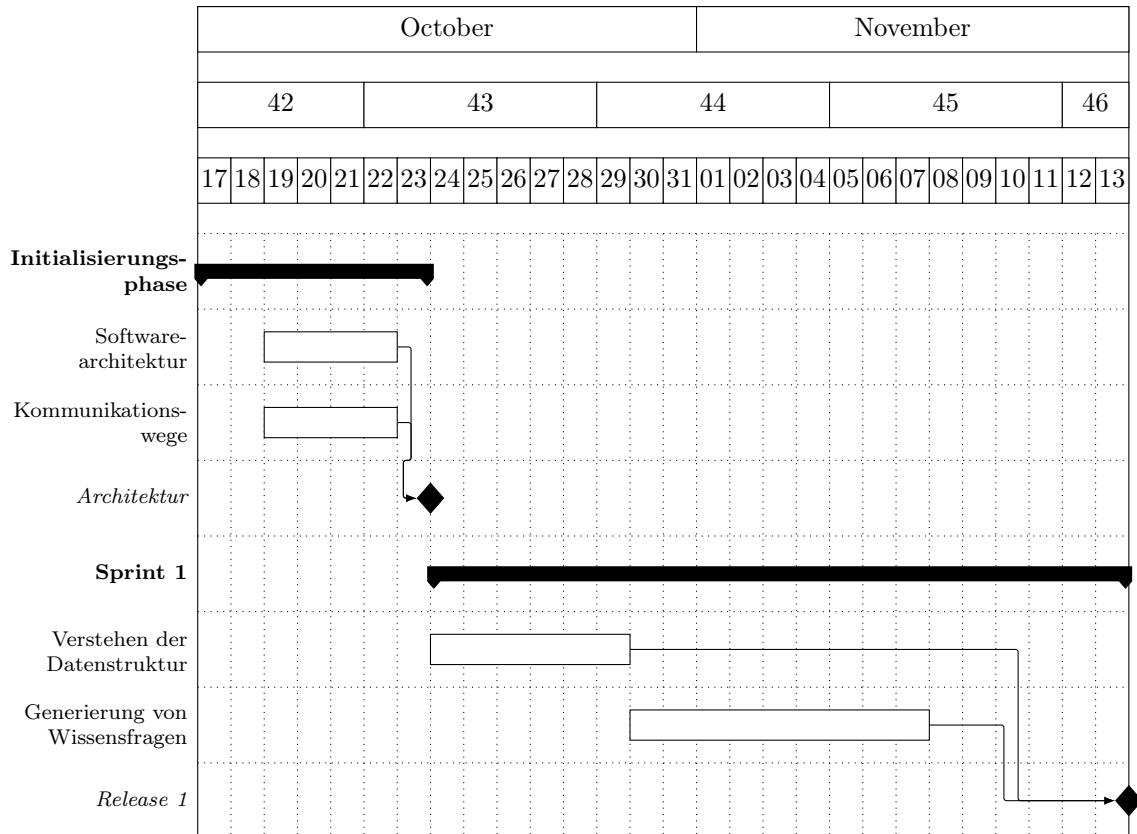
2 Projektführung
2.1 Rahmen- und Projektplan
2.1.1 Überblick über den gesamten Projektzeitraum





2.1.2 Projektplan für die Initialisierungsphase und den ersten Sprint

Viele Dinge, die für die Initialisierungsphase vorgesehen sind, haben wir schon erledigt. Dazu gehören die Technologiefestlegung und die Infrastruktur, welche in Kapitel Projektunterstützung beschrieben werden.



2.2 Projektkontrolle und Projektsteuerung

Kosten- / Zeit-Controlling: Jedes Teammitglied erfasst seine Zeitaufwände für jedes Issue und trägt sie in eine gemeinsame Textdatei ein. Bei den regelmäßigen Treffen werden diese zusammengetragen und ausgewertet. Bei Abweichungen vom geplanten Aufwand werden die Ursachen im Team besprochen und eine Lösung gesucht. Damit wird ständig im Auge behalten, ob ein Mitglied sich eventuell bei der Aufgabe über- bzw. unterschätzt hat und wir können schnell nachregulieren.

Leistungsumfang-Controlling: Zu Beginn jedes Sprints werden die Aufgabenpakete im Team besprochen und gemeinsam ein Leistungsumfang besprochen. Sollte sich im Lauf des Sprints herausstellen, dass einzelne Pakete doch umfangreicher sind als gedacht, werden diese in dem nächsten Team-Meeting besprochen und eventuell in zwei Pakete geteilt und unter den Teammitgliedern delegiert.



Qualitäts-Controlling: Der Project Owner macht einige Tage vor dem Ende der Sprints eine vorläufige Abnahme und prüft, ob die geforderten Ziele in den einzelnen Paketen erreicht wurden. Desweiteren schaut er sich einmal die Woche die Änderungen an und bespricht sie mit dem Team. So soll gewährleistet werden, dass alle Mitglieder die Änderungen verstehen und eventuell andere Möglichkeiten zumindest aufgezeigt und besprochen werden. Im Idealfall kann man so noch die ein oder andere Optimierungsmöglichkeiten finden.

3 Projektunterstützung

3.1 Tools für die Entwicklung

- IntelliJ IDEA
 - zur Entwicklung
 - einfach und effizient
- Java
 - Programmiersprache
 - Wir können alle in Java programmieren
 - Hadoop basiert auf Java
- Linux
 - Betriebssystem
 - Wir alle benutzen Linux
 - Linux ist besser geeignet für die Entwicklung von Hadoop
- Maven
 - zur Entwicklung
 - bietet dem Entwickler eine einfache Einbindung von Librarys
- WikiData Toolkit
 - Datenverarbeitungs-Tool
 - WikiData Toolkit macht es für Entwickler einfach, die Daten von WikiData aus Dumps zu generieren und in ihren eigenen Anwendungen zu nutzen
- Hadoop
 - Datenverarbeitungs-Framework
 - Der Hauptvorteil bei der Verwendung von Hadoop ist, dass es ideal geeignet ist, um große Datenmengen zu analysieren
 - Open Source
- LaTeX



- Dokumentenvorbereitungssystem
- einfache Möglichkeit zur Generierung von Inhalts- und Literaturverzeichnis
- sehr gute Unterstützung von mathematischen Formeln
- gut mit git zur Versionsverwaltung nutzbar

3.2 Konfigurationsmanagement

- Wikidata-Dump vom 01.10.2018
 - Datenquelle (`wikidata-20181001-all.json.bz2`)
 - Wikidata empfiehlt Dumps als Javascript Object Notation (JSON).
 - Neueste Version bei Projektstart
 - Ergebnisse sind reproduzierbar, wenn alle die gleichen Daten benutzen
 - Viele Daten ändern sich nicht, sodass eine Aktualisierung nicht notwendig ist

3.3 Tools zur Projektkommunikation und zum Dokumentenaustausch

- Git
 - Versionskontrollwerkzeug
 - Es ist unglaublich schnell, äußerst effizient.
 - dezentralisiert
- GitLab
 - web-basierter Git-Repository-Manager
 - Wir benutzen Gitlab, um unser SCRUM zu ermöglichen. Issues ist wie eine product backlog. Wir können auch direkt darauf kommunizieren.

4 Qualitäts- und Testplan

4.1 Definition of Ready und Definition of Done

- **Definition of Ready:**
 - Items dürfen nicht von anderen Product Backlog Items (PBIs) abhängen, die nicht im Sprint-Backlog sind, haben
 - Die Aufgabe muss von allen verstanden und akzeptiert worden sein
 - Der Arbeitsaufwand muss geschätzt worden sein
- **Definition of Done:**
 - Alle automatischen und manuellen Tests wurden bestanden
 - Der Code wurde kommentiert und eine Dokumentation erstellt
 - Der Product-Owner würde die Story dem Kunden vorstellen wollen

4.2 Testdesign & Testautomatisierung

Wir müssen nach jeder Neuerung testen, ob die Ergebnisse sinnvoll sind. Dies ist von Hand zu machen, da eine Testautomatisierung in dem Fall fast schwieriger zu programmieren wäre als die eigentliche Software. Aus diesem Grund brauchen wir vorerst keine Werkzeuge zur Testautomatisierung. Wenn später einiges an Code geschrieben wurde, können Unit-Test benutzt werden, um einzelne Methoden zu testen.

4.3 Testfälle

5 Anhänge

5.1 Abkürzungen

JSON Javascript Object Notation. 7

PBI Product Backlog Item. 7