



PROJEKTPLAN

SCRUMIVERSE



“The whole universe of scrum projects”

*Lasse Jacobs
Joshua Ward
Kevin Jolitz*

*Toni Serfling
Kevin Wessler*





INHALTSVERZEICHNIS

1	<u>PROJEKTBESCHREIBUNG SCRUM MANAGER</u>	3
2	<u>BEGRÜNDUNG DER PROJEKTWAHL</u>	3
3	<u>IST-ANALYSE „VORGEHENSMODELL SCRUM“</u>	4
3.1	ERKLÄRUNG „VORGEHENSMODELL SCRUM“	4
3.2	DARSTELLUNG DES ARBEITSPROZESSES	4
3.3	UMSETZUNGEN DES ARBEITSPROZESSES	5
3.4	PROBLEME DER UMSETZUNGEN	5
3.5	FAZIT	6
4	<u>SOLL-KONZEPT: VORGEHENSMODELL</u>	6
4.1	GENERELLES	6
4.2	PROJEKTSTRUKTURPLAN	7
4.3	ZEITPLANUNG	7
5	<u>SOLL-KONZEPT: LÖSUNGSANSATZ</u>	8
5.1	GENERELLES	8
5.2	SYSTEMARCHITEKTUR	8
5.3	DATENMODELL	9
6	<u>DURCHFÜHRUNGSBESCHREIBUNG</u>	9
6.1	PLANÄNDERUNGEN	9
6.2	ARCHITEKTURÄNDERUNGEN	9
6.3	GRUPPENDYNAMIK	10
7	<u>DURCHFÜHRUNGSÜBERSICHT</u>	11
7.1	KEVIN JOLITZ	11
7.1.1	SPRINT 1: DOKUMENTATION UND PROTOTYP	11
7.1.2	SPRINT 2: KERN IMPLEMENTATION	11
7.1.3	SPRINT 3: FINALISIERUNG DER WEBSITE	11
7.1.4	SPRINT 4: PRÄSENTATION & DOKUMENTATION	11
7.2	KEVIN WESSELER	11
7.2.1	SPRINT 1: DOKUMENTATION UND PROTOTYP	11
7.2.2	SPRINT 2: KERN IMPLEMENTATION	11
7.2.3	SPRINT 3: FINALISIERUNG DER WEBSITE	11
7.2.4	SPRINT 4: PRÄSENTATION & DOKUMENTATION	11
7.3	LASSE JACOBS	11
7.3.1	SPRINT 1: DOKUMENTATION UND PROTOTYP	11
7.3.2	SPRINT 2: KERN IMPLEMENTATION	11
7.3.3	SPRINT 3: FINALISIERUNG DER WEBSITE	12
7.3.4	SPRINT 4: PRÄSENTATION & DOKUMENTATION	12
7.4	TONI SERFLING	12
7.4.1	SPRINT 1: DOKUMENTATION UND PROTOTYP	12
7.4.2	SPRINT 2: KERN IMPLEMENTATION	12
7.4.3	SPRINT 3: FINALISIERUNG DER WEBSITE	12
7.4.4	SPRINT 4: PRÄSENTATION & DOKUMENTATION	12
7.5	JOSHUA WARD	12
7.5.1	SPRINT 1: DOKUMENTATION UND PROTOTYP	12
7.5.2	SPRINT 2: KERN IMPLEMENTATION	12
7.5.3	SPRINT 3: FINALISIERUNG DER WEBSITE	12
7.5.4	SPRINT 4: PRÄSENTATION & DOKUMENTATION	13

8	FAZIT	13
8.1	EINZELFAZITS	13
8.1.1	LASSE JACOBS	13
8.1.2	KEVIN JOLITZ	13
8.1.3	TONI SERFLING	13
8.1.4	JOSHUA WARD	14
8.1.5	KEVIN WESSELER	14
8.2	GESAMTFAZIT	14
9	ANHANG	15
9.1	PROJEKTSTRUKTURPLAN	15
9.2	GANTT DIAGRAMME	17
9.3	DATENMODELL	21
9.4	SYSTEMARCHITEKTUR	24
9.5	PAPIERPROTOTYP	25

1 PROJEKTBSCHREIBUNG SCRUM MANAGER

Ziel dieses Projektes ist die Entwicklung einer „Scrum-Projektmanagement-Software“. Die Website-Anwendung soll die onlinebasierte Zusammenarbeit von Projektteams mit dem Vorgehensmodell „Scrum“ ermöglichen und fördern.

Durch den Auftraggeber sind folgende Technologien vorgegeben:

- **Entwicklung**
 - Spring MVC, Java
 - JSP / JSTL
 - JavaScript
- **Planung und Realisierung**
 - Webplattform [Scrumdesk](#)
 - Versionsverwaltung mit Git mit einem Repository zur Verfügung gestellt von [GitHub](#)

Um unser Projektziel zu erreichen hat sich das Projektteam auf folgenden Funktionsumfang geeinigt:

- Ein umfangreiches Accountmanagement, welches Rechteverwaltung innerhalb von Projekten ermöglicht.
- Funktionen zur Unterstützung des Projektmanagements
 - Detailgetreue Abbildung des Scrum-Modells
- Eine angemessene grafische Aufbereitung, welches jederzeit eine Auswertung des aktuellen Projektstandes ermöglicht.
 - Dashboard mit Ereignisreport
- Echtzeit Informationsverteilung durch E-Mail notifier und einem Nachrichtensystem zwischen Projektteilnehmern.

Folgende Elemente werden eventuell realisiert:

- Rating System
- Kapazitätenplanung der Benutzer durch den Projektleiter.
- Anbindung an externe Versionsverwaltung Software wie beispielsweise Github oder andere Git Repositories.

Der zeitliche Rahmen für die Umsetzung beginnt am 04.01.2016 0:00 und endet am 25.04.2016 um 23:59 Uhr, mit der Abgabe der folgenden Elemente:

- Kommentierter und archivierter (zip) Quellcode der Webplattform
- Projektdokumentation
- Projektpräsentation

2 BEGRÜNDUNG DER PROJEKTWAHL

Moderne Softwareentwicklung wird immer häufiger auf Basis von agilen Vorgehensmodellen durchgeführt. Für diese agilen Ansätze existieren auch schon verschiedene Softwareprojekte, welche die Arbeit unterstützen. Jedoch kosten diese viel Geld oder weisen einige Mängel in der Bedienung oder Rechteverwaltung auf (Siehe Ist-Analyse).

Durch eigene gute Erfahrungen im Team mit dem agilen Ansatz „Scrum“ besteht im Team ein hohes Interesse, eine gute Plattform für private oder eventuell sogar berufliche Projekte zu planen und umzusetzen.

3 IST-ANALYSE „VORGEHENSMODELL SCRUM“

Im Folgenden wird das Vorgehensmodell Scrum näher betrachtet. Nach einer kurzen Erklärung wird die Darstellung des Arbeitsprozesses und dessen Umsetzung beleuchtet, sowie die Probleme der bereits bestehenden Umsetzungen. Abschließend wird daraus ein Fazit gezogen, in dem auf Verbesserungsmöglichkeiten eingegangen wird.

3.1 ERKLÄRUNG „VORGEHENSMODELL SCRUM“

Scrum ist ein Vorgehensmodell im Bereich Projektmanagement, welches ursprünglich in der Softwareentwicklung entwickelt und dort zuerst eingesetzt wurde. Es ist ein agiler Ansatz der auf iterativer Entwicklung aufbaut und den Kunden (bei Scrum Product Owner) aktiv in die Entwicklung mit einbezieht.

3.2 DARSTELLUNG DES ARBEITSPROZESSES

Scrum besteht aus einem zyklischen Aufbau um Softwareentwicklung zu unterstützen, sogenannten Sprints.

Ein Sprint ist ein zeitlicher Rahmen (meist 2-4 Wochen) in welchem eine neue Iteration des Produkts entwickelt wird.

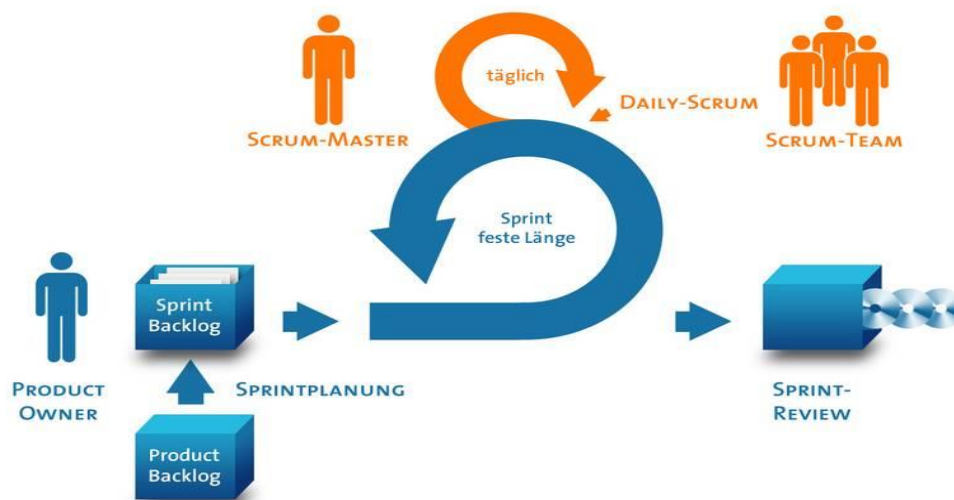


Abbildung 1: Scrum-Arbeitsprozess
<https://www.3m5.de/scrum/>

Während des Sprints gibt es tägliche Meetings (Daily Scrum), bei denen der Scrum Master und das Team ihren Fortschritt, Probleme und Erfolge kurz erläutern. Am Ende eines Sprints wird dem Product Owner diese Iteration vorgestellt und mit diesem der nächste Sprint geplant. Damit wird der Kunde, wie oben bereits erwähnt, aktiv in den Entwicklungsprozess einbezogen. Dies erlaubt eine im Vergleich zu nicht-agilen Vorgehensmodellen sehr flexible Entwicklung, in der schnell auf Änderungswünsche reagiert werden kann.

Außerdem wird am Ende jedes Sprints eine Sprint-Retrospektive durchgeführt, in der alle Beteiligten des Projekts Feedback zum vergangenen Sprint geben können.

Dokumentiert wird Scrum mithilfe von zwei zentralen Dokumenten: Der Produkt-Backlog (ähnlich dem Lasten-/Pflichtenheft aus traditionellen Vorgehensmodellen) bestimmt die Spezifikationen des fertigen Produkts, ist jedoch jederzeit anpassbar und zu verbessern. Der Sprint-Backlog nimmt einzelne Teile dieses Product-Backlogs und erstellt Arbeitspakete für den nächsten Sprint, welche

wiederrum in kleinere Tasks aufgeteilt werden können, welche zum Beispiel in Form von User Stories formuliert werden können.

Eine User Story ist ein Anwendungsfall des Produkts, quasi eine Antwort auf die Frage „Was möchte der User mit dem Produkt machen können?“. Ebenfalls besitzt Scrum drei Schlüsselrollen im Entwicklungsprozess, zunächst den bereits erwähnten Product Owner, welcher der Auftraggeber des Produkts ist, Backlog-Items priorisiert und für den wirtschaftlichen Erfolg ebendieses Produktes zuständig ist.

Der Scrum Master ist der Leiter des Scrums und dafür verantwortlich dass der Scrum gelingt. Er arbeitet mit dem Entwicklungsteam zusammen, welches für die Lieferung der im Product-/ Sprint Backlog aufgelisteten Features zuständig ist, gehört aber meist nicht dazu.

3.3 UMSETZUNGEN DES ARBEITSPROZESSES

In heutigen Arbeitsumgebungen existieren viele Ansätze Scrum als Projektplanungstool einzusetzen. Dabei lässt sich zwischen dem „Offline Ansatz“ und dem „Online-Ansatz“ unterscheiden. Der Offline Ansatz besteht in seiner Urform aus Karteikarten, welche die User Stories repräsentieren. Diese werden entsprechend auf einem Whiteboard oder einer der Bürowände angeordnet, je nach Zuständigkeit und aktiven Sprint. Zumeist wird dieser Ansatz auch durch digitale Dokumente gestützt, ist aber eher in der eben beschriebenen Papierform zu finden, um dynamisch Dinge verschieben zu können und die Kommunikation und Zusammenarbeit innerhalb eines Büros zu fördern.

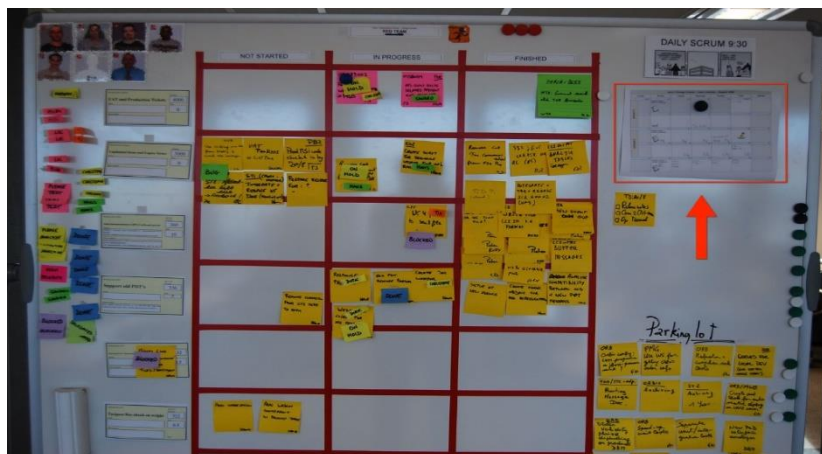


Abbildung 2 Bsp.: Offline-Scrumboard

http://www.xqa.com.ar/visualmanagement/wp-content/gallery/general-pictures/board_with_calendar.jpg

Die andere Art der Scrum Projektplanung sind digitale Webplattformen, welche durch ihren Aufbau und Workflow versuchen die Grundprinzipien von Scrum umzusetzen und den haptischen und agilen Ansatz der Papier- und Stiftvariante zu imitieren.

Zusätzlich bieten sie weitere Funktionen, wie z.B. die Möglichkeit mehrere Projekte gleichzeitig zu verwalten, Ressourcen einzuteilen oder wichtige grafische Elemente von Scrum (z.B. BurnDown-Chart) automatisch generieren zu lassen.

Auch die Kommunikation wird versucht zu übertragen und bei größeren Projektgruppen zu vereinfachen. Dazu existieren verschiedene Kommentarfunktionen oder Nachrichtensysteme auf den Plattformen.

3.4 PROBLEME DER UMSETZUNGEN

Trotz der vielen unterstützenden Funktionen der Webanwendungen gibt es viele Probleme, die wir durch die Analyse von Konkurrenzprodukten gefunden haben (für detaillierte Informationen siehe 2_Konkurrenz_Analyse.pdf). Allen voran gibt es bei

vielen der Plattformen eine große Einstiegshürde durch preisliche Barrieren, die sich meist anhand der Elemente, der Teilnehmeranzahl oder nach einem festen Abo Modell berechnen. Sollte man sich dann für eine der Plattformen entschieden haben, gibt es dann meist größere Mängel was die Übersicht über Funktionen und Informationen angeht. Intuitiv sind dadurch die Oberflächen selten und der Scrum Arbeitsprozess wird entweder abgeändert oder schlichtweg mangelhaft umgesetzt.

Dazu kommt dann ein meist starres oder schlichtweg kaputtes Rechtemanagement. Beispielsweise auf der Plattform ScrumDesk, an denen Benutzer User Stories und Tasks selbständig erstellen, löschen und ändern können. Ändern kann man dies nicht, Teilnehmer können alles oder nichts.

Wenn die Rechte nicht schon zu Chaos führen, könnte dies durch die teilweise sehr langsame Synchronisierung kommen. Auf vielen der Plattformen ist es unklar, wann und wie eine Synchronisierung erfolgt und zudem dauert es häufig zu lange.

Der letzte Punkt betrifft die Datensicherheit und die Zusammenarbeit mit anderen Systemen wie Git oder SVN zur Versionskontrolle. Datensicherheit ist meist nicht gegeben, da keines der von uns analysierten Tools eine Export Funktion bereitstellte. Dadurch muss man sich auf die Sicherungen des Anbieters verlassen. Ebenso ist die Anbindung an andere Systeme nur selten möglich und wenn, nur in den kostenpflichtigen Anwendungen.

Viele dieser Nachteile hat die klassische offline Variante nicht. Dafür ist ihr gravierendster Nachteil die kaum koordinierbare Arbeit mit vielen Teilnehmern. Wenn die Leute nicht im gemeinsamen Großraumbüro sitzen wird es enorm schwierig zu kommunizieren und aktuelle Fotos oder Office Dokumente mit dem aktuellen Sprint umher zu schicken und zwischen den Teilnehmern zu synchronisieren. Dazu kann es zu einem Tunnelblick bei der Papiervariante kommen, da man häufig den Fokus auf einen Sprint legt und so das große ganze eventuell im Papierchaos aus den Augen verliert.

3.5 FAZIT

Aus diesen von uns erarbeiteten Mängeln lassen sich folgende Kernprobleme ableiten, die wir verbessern und überarbeiten wollen:

- Kostenloses Geschäftsmodell anbieten
- Funktionen und Informationen übersichtlich darstellen, auch für mehrere Projekte.
- Saubere und intuitive Umsetzung des Scrum Arbeitsprozesses
 - Versuchen den klassischen/haptischen Ansatz in die digitale Welt zu übertragen
- Konfigurierbares und detailliertes Rechtemanagement
- Den Benutzern vielfältige Kommunikations- und Kommentarfunktionen bieten
- Versuchen die Synchronisationszeit zu verringern.

4 SOLL-KONZEPT: VORGEHENSMODELL

4.1 GENERELLES

In der Gruppe haben wir uns dazu entschieden das agile Vorgehensmodell Scrum zu verwenden. Jedoch viele Elemente von Scrum sind nicht vollständig benutzbar aufgrund der Projektvorgaben, daher benutzen wir eine abgewandelte Form. Zunächst haben wir keinen Product Owner und Scrum Master, sondern einen traditionellen Projektleiter. Die User Stories (Arbeitspakete) werden gemeinsam in der Gruppe besprochen und erstellt. Übertragen werden diese auf die Website „Scrumdesk“, welche viele Funktionen und Erleichterungen für die Projektplanung bereitstellt. Eine

zusätzliche Abweichung betrifft die Abhängigkeit zwischen den User Stories, welche durchaus gegeben ist bei unserem Projekt.

Eine weitere Abweichung betrifft die Flexibilität. Durch die externen Rahmenbedingungen müssen Sprints in unserem Projekt verschiedene Längen haben und die Abgabe von User Stories ist streng zeitlich vorgegeben und ist weitaus weniger flexibel als im Standard Scrum. Die Kommunikationsinstrumente von Scrum verwenden wir gar nicht oder nur in leicht abgewandelter Form. Aus dem Daily Scrum haben wir ein weekly Scrum gemacht, welches wir am Ende der Woche in Textform niederschreiben. Aus diesem Dokument und der allgemeinen Zeitverschreibung wird dann ein Ablauf und Fazit für das Projekt ermittelt.

Auf Sprint Reviews verzichten wir, genauso wie die Vorgabe dass Sprints feste Iterationen des Produkts ergeben müssen. In den Sprints ergeben sich nicht immer Prototypen der Software, sondern fest definierte Ergebnisse des Gesamtprojektes (Projektdokumentation, Prototyp, Abgabefassung, Präsentation).

Regelungen, die Toolsbaseline und die Tagging Richtlinien für unsere Commits auf der Seite GitHub, sind in einem externen Dokument festgehalten, dem sogenannten „Teilnehmer Handbuch“ (**siehe xyz**).

4.2 PROJEKTSTRUKTURPLAN

In unserem Projektstrukturplan haben wir uns auf die Epics, welche wir in „Scrumdesk“ festgelegt haben, beschränkt. Diese beschreiben die Features, welche wir in unserem Projekt umsetzen wollen. Der Grund für die Beschränkung wäre ein Verlust der Übersicht, durch die hohe Anzahl von User Stories.

Im Projektstrukturplan werden wir eine Aufteilung nach den verschiedenen Sprintphasen vornehmen. Diese Phasen werden in der Umsetzung gestaffelt (**siehe AStruktur 2 - Sprint Übersicht**). So werden im ersten Sprint die Projektplanung und die Planung eines Prototypens vorgenommen (**siehe AStruktur 3 - Sprint 1**). In der zweiten Sprintphase wird die Implementierung der Kernfunktionen der Software vorgenommen (**siehe AStruktur 4 - Sprint 2**). Die dritte Sprintphase wird zur Finalisierung und Erweiterung der Funktionen/des Funktionsumfanges genutzt (**siehe AStruktur 5 - Sprint 3**). Im dem Sprint wird die Entwicklung abgeschlossen. Der letzte Sprint wird sich mit der Erstellung der Präsentation und Dokumentation beschäftigen (**siehe AStruktur 6 - Sprint 4**). Mit diesem Sprint wird das Projekt „Scrumiverse“ beendet.

4.3 ZEITPLANUNG

Die im Projektstrukturplan erwähnten Sprintphasen haben verschiedene Prioritäten und Längen. Da die Planung eine sehr hohe Priorität in unserem Projekt hat, wurde der erste Sprint mit 5 Wochen etwas länger angesetzt (**siehe Gantt Diagramme: AGantt1 – Sprint 1**).

Für den zweiten Sprint haben wir wie im klassischen Scrum haben hierfür 4 Wochen bzw. 1 Monat vorgesehen (**siehe Gantt Diagramme: AGantt2 – Sprint 2**).

Im dritten Sprint haben wir für die ersten drei Wochen für die Finalisierung der Webseite geplant, in dem wir nach dem zweiten Sprint noch fehlende Features implementieren. Der Meilenstein des dritten Sprint ist erst nach 4 Wochen erreicht, sodass wir noch 1 Woche Zeit haben um fehlerhafte Teile der Software zu beheben, noch unfertige User-Stories zu bearbeiten oder weitere Features hinzuzufügen (**siehe Gantt Diagramme: AGantt3 – Sprint3**).

Nachdem wir im zweiten und dritten Sprint die Implementation abgeschlossen haben, werden wir im vierten Sprint die erste Woche nutzen um die erwähnte Präsentation zu

erstellen und die Dokumentation aus Sprint 1 zu finalisieren. Ebenso wie in Sprint 3, ist der Meilenstein ein paar Tage nach Ende des geplanten Sprints angesetzt, um Verzögerungen oder Unerledigtes zu kompensieren (siehe **Gantt Diagramme: AGantt4 – Sprint 4**).

Nachdem die Sprints erstellt wurden, hat sich das Team zusammengesetzt, User-Stories gemeinsam erarbeitet und anschließend auf Sprint 2 und 3 aufgeteilt. Im Zuge dessen waren die Teammitglieder, nachdem das Team in Backend und Frontend eingeteilt wurde, selbstständig dafür verantwortlich sich User-Stories zuzuweisen und in dieser Tasks zu erstellen. Dabei wurde auch auf die Zeiteinteilung eines Teammitglieds während der Sprints geachtet, da zu den Task auch die Zeit geschätzt werden musste.

5 SOLL-KONZEPT: LÖSUNGSANSATZ

5.1 GENERELLES

Unter der Betrachtung der Konkurrenz verfolgen wir den Ansatz, ein mit von Scrum gestützte Online-Projektplanung zu erstellen. Dabei möchten wir eine ähnliche Plattform schaffen. Funktionen und Umsetzungen, welche wir für umständlich oder nicht intuitiv halten, wollen wir verbessern. Um mit den bereits existierenden Plattformen zu konkurrieren werden wir auch Funktionen adaptieren, welche wir für unsere Plattform als sinnvoll erachten.

Das Vorgehensmodell von Scrum soll besser und originalgetreu umgesetzt werden. Hierfür wollen wir die haptische Umsetzung in eine Online-Version bündeln und zentral erreichbar machen. Auf Konkurrenz-Plattformen ist uns schnell aufgefallen, dass ein Rechtemanagement nur schlecht oder gar nicht vorhanden ist. Dies wollen wir für unsere Benutzer sinnvoller umzusetzen.

Ein weiterer Aspekt der originalgetreuen Umsetzung betrifft das Weglassen einiger zusätzlicher Funktionen, wie das Verwalten von Releases und Kanban. Aufgrund der zeitlichen Knappheit werden auch wenige Kernfunktionen nicht umgesetzt, wie z.B. eine Retrospektive.

5.2 SYSTEMARCHITEKTUR

Die Architektur der Website „Scrumiverse“ ist in Frontend und Backend aufgeteilt. Die Basis des Backends bildet Java, für das Controlling kommt das SpringMVC Framework, für die Persistenz das Hibernate ORM-Framework zum Einsatz. SpringMVC arbeitet mit einem Dispatcher Servlet, welches Serveranfragen an einen passenden Controller weiterleitet. Dieser mappt ein Model welches dann an die View weitergegeben wird. Die Systemarchitektur ist durch SpringMVC streng vorgegeben. Außerdem wird durch das MVC pattern das Backend klar von dem Frontend getrennt.

Hibernate ist ein O/R-Framework, d.h. es verbindet das objektorientierte Modell von Java mit dem relationalen Modell von klassischen Datenbanken. Hibernate erspart das Schreiben eigener Datenbankzugriffe und somit auch wertvolle Zeit.

In der Entwicklung des Frontends werden hauptsächlich JQuery, JSTL, JavaScript, Bootstrap, Ajax und Highchart verwendet, auf welche hier jedoch nicht weiter eingegangen wird. Durch JSTL wird viel Glue Code in der View Ebene gespart. Zur Vereinfachung und Standardisierung der Benutzeroberfläche verwenden wir Bootstrap. Des Weiteren liefert Bootstrap auch neue Kontrollelemente. Highchart erleichtert die Darstellung von Diagrammen.

Da Bootstrap und Highchart teilweise auf JavaScript basiert, werden dessen Elemente auf älteren Browsern nicht Unterstützt.

In der Persistenzschicht existieren Interfaces für User (UserDAO), Rollen (RoleDAO), Projekte (ProjektDAO), Sprints (SprintDAO) und User Stories (UserStoryDAO). Zusätzlich dazu gibt es Taskobjekte.

Ähnlich zu den DAOs in der Persistenzschicht arbeitet das Backend durch SpringMVC auch mit passenden Controllern: User-Controller, Rollen-Controller, Projekt-Controller, Sprint-Controller und User Story-Controller.

Für eine graphische Abbildung der Systemarchitektur: siehe **Systemarchitektur, ASystem 1**

5.3 DATENMODELL

Bei der Erstellung des Datenmodells haben wir uns stark an dem Basis-Vorgehensmodell von Scrum orientiert (siehe **Datenmodell: Business Domain**). In Anbetracht unserer Umsetzung haben wir dann noch zusätzlich Website relevante Datenmodelle abgebildet (siehe **Datenmodell: Message und Ereignis Modelle** und **Datenmodell: Benutzerdaten**).

6 DURCHFÜHRUNGSBESCHREIBUNG

6.1 PLANÄNDERUNGEN

Größtenteils konnte die von uns geplante Durchführung fast genauso durchgeführt werden, wie in den vorherigen Kapiteln beschrieben. Änderungen betreffen zwei Bereiche. Zunächst den Einsatz externer Technologien, um den Upload von Bildern für die Profile und Projekte zu ermöglichen. Hierzu wurden zusätzliche Bibliotheken von Apache benötigt, welche während des laufenden Projektes eingebunden und benutzt werden mussten (weitere Beschreibungen siehe Architekturänderungen).

Der zweite Bereich betrifft die Planung des Projektes, was aber für ein Scrum Projekt normal ist. So mussten User Stories neu Priorisiert werden und Sprint Längen angepasst werden. Vor allem der Zeitgewinn gegen Ende des Projektes ermöglichte die Verlängerung unserer Polishing Phase.

Diese Anpassungen waren relativ leicht durch die Unterstützung mit unserer Planungs-Plattform Scrumdesk, welches von uns effizient und insgesamt Projektunterstützend verwendet wurde. Die Verschreibung der Zeiten war manchmal etwas verzögert, aber wurde insgesamt sorgfältig gepflegt.

Der Zeitrahmen und die von uns geschätzte Dauer und Aufteilung der einzelnen User Stories war insgesamt zu gering. Durch Probleme mit unseren gewählten GitHub Client und die mangelnde Erfahrung mit Hibernate führte zu Verzögerungen und letztlich zum Streichen von Features, welche in der Architekturbeschreibung näher beschrieben werden.

6.2 ARCHITEKTURÄNDERUNGEN

Im Laufe der Arbeit hat sich herauskristallisiert, dass unsere initial Architektur an vielen Stellen noch zu Grob und Ungenau war. Viele Schnittstellen im Datenmodell mussten neu definiert / dynamisch ermittelt werden. Vor allem durch die Arbeit mit Hibernate mussten manche Beziehungen zwischen den Datenmodell Klassen überarbeitet werden. Hierbei seien vor allem die komplexen Beziehungen zwischen einer Rolle, Projekt und Benutzer zu erwähnen, welche sehr schwer in Hibernate umzusetzen waren. Letztendlich führten wir eine neue Klasse ein, ProjectUser. Diese Klasse war die Verbindung zwischen Benutzer und Rolle, welche in Projekt dann verwaltet wurde. Damit konnte die vorhandene Dreiecksbeziehung aufgelöst werden und daraus eine einfache 1 zu n Beziehung zwischen Projekt und der neuen Entität gemacht werden, welche durch Hibernate besser persistiert werden konnte.



Ebenso ergaben sich durch die Anforderungen des Rechtemanagements Änderungen bei den Controllern. Da jede Funktion des Controllers im Grunde die Rechte prüfen musste, nämlich auf den aktuellen Benutzer und/oder Projekt in der Session, wurde ein MetaController eingeführt von dem die anderen Controller erben. Dieser lieferte die Benötigten Getter und Prüfungen, ohne doppelten Code in den einzelnen Controllern. Natürlich brachten diese Änderungen durchaus einige Risiken mit, welche die Gruppe aber sehr gut durch ausgebiegige Gespräche umschiffen hat.

Der von uns ausgearbeitete Papierprototyp und damit die Vision des Tools (siehe Papierprototyp) erwies sich als äußerst nützlich und konnte zu vielen Teilen fast so umgesetzt werden. Änderungen betrafen den Stil, welcher leicht angepasst werden musste und das Zusammenlegen von manchen Seiten, was vor allem die Projekteinstellungen betraf und die Benutzerfreundlichkeit dieser erhöhte.

Leider mussten dennoch einige Funktionen gestrichen werden, unter anderem die Möglichkeit von Touch Interaktionen mit der Oberfläche.

Gravierender war das Streichen von zwei wichtigen Kernfunktionen. Es war geplant die Registrierung und Notifications über ein E-Mail System laufen zu lassen, was sich aber als zu umfangreich im Rahmen der Projektzeit herausstellte. Ebenso wie das Filtern von einzelnen Planungselementen und ein weiterer Chart, welcher erledigte UserStories nach verschiedenen Kategorien aufzeigen sollte. Was aber stattdessen im Rahmen der Zeit umgesetzt werden konnte, war der Upload von Benutzer- und Projektbildern, zur besseren Unterscheidung von Benutzern und Projekten.

6.3 GRUPPENDYNAMIK

Insgesamt waren das Arbeitsklima und der Umgang miteinander sehr positiv. Das Vorgehensmodell Scrum bietet viele Stolperfallen für Gruppen, welche wir aber fast alle erfolgreich umgangen haben. Bei notwendigen Diskussionen beteiligte sich das Team die meiste Zeit rege an diesen und Ergebnisse konnten rasch erzielt werden. Dabei entstanden im Laufe der Zeit auch weitaus bessere Lösungen, als die von uns zunächst geplanten. Es gab aber durchaus Phasen, an denen die Gruppe doch eher unkonzentriert war und es schwer war, alle Teammitglieder auf die Aufgabe zurück zu bringen. Dennoch war der Einsatz von Scrum für uns also insgesamt ein Gewinn, der zu einem besseren Produkt geführt hat als ursprünglich geplant.

Die von uns in der Projektplanungsphase vereinbarten Absprachen wurden größtenteils eingehalten, bis auf ein paar Ausnahmen. So waren manchmal die Zeitverschiebungen etwas verzögert und nicht immer aktuell und manche vom Team und Projektleiter gesetzte Prioritäten wurden eigenmächtig geändert. Dies war jedoch zum Glück nur einmal der Fall und gefährdete das Gelingen dieses Projektes nicht. Schwerwiegender war da schon der mangelhafte Umgang mit unserem GitHub Client. Trotz der Absprache, dass regelmäßig committed / gepusht werden sollte, hielten sich manche nicht daran. So kam es teilweise zu Überschreibungen oder Merges, die vor allem in Frontend Bereich unnötig Mehrarbeit auslösten.

Passend dazu war das Problem, dass manche mit aller Gewalt ihren Teil auch alleine fertigstellen wollten. Durch die Teamzusammenstellung, welche aus durchaus erfahrenen Entwicklern besteht, versuchte jeder seine gestellte Aufgabe möglichst alleine zu bewältigen. Dadurch haben manche Features Verzögerungen erlitten, welche durchaus hätten früher fertig sein können, wenn die Leute früher eingelenkt und um Hilfe gebeten hätten.

7 DURCHFÜHRUNGSÜBERSICHT

7.1 KEVIN JOLITZ

7.1.1 SPRINT 1: DOKUMENTATION UND PROTOTYP

Im ersten Sprint hat sich Kevin Jolitz um den Projektantrag gekümmert und hat die Ist-Analyse aus den Ergebnissen von anderen Projektmitgliedern zusammengetragen. Zusammen mit der Gruppe ermittelte er User Stories für das Projekt. Zuletzt lag der Fokus auf dem Vorgehensmodell und der Architektur.

7.1.2 SPRINT 2: KERN IMPLEMENTATION

Im zweiten Sprint hat sich Kevin zunächst mit dem Accountsystem und dem Projekt im Backend beschäftigt. Damit einher ging die Rollenverwaltung, ebenfalls im Backend. Zwischendurch unterstützte er Kevin Wessler im Frontend bei der Zuweisung von User Stories in der Sprintübersicht. Am Ende des Sprints implementierte er die Verwaltung von Tasks.

7.1.3 SPRINT 3: FINALISIERUNG DER WEBSITE

Der Fokus des dritten Sprints lag für Kevin auf dem Rechtemanagement sowohl im Frontend als auch im Backend. Zudem implementierte er noch den Upload von Bildern, unterstützte die Taskdetailansicht und überarbeitete das Datenmodell Sprint. Gegen Ende des Sprints hat Kevin noch Javadoc verfasst.

7.1.4 SPRINT 4: PRÄSENTATION & DOKUMENTATION

Im letzten Sprint arbeitete Kevin an der Beschreibung der Projektdurchführung und verfasste sein persönliches Fazit.

7.2 KEVIN WESSELER

7.2.1 SPRINT 1: DOKUMENTATION UND PROTOTYP

Im ersten Sprint arbeitete Kevin Wessler in Zusammenarbeit mit Lasse an der Analyse von Konkurrenzprodukten. Zusammen mit der Gruppe ermittelte er User Stories für das Projekt und zuletzt erstellte er mit Joshua einen Papierprototyp, der ein grobes Bild des Frontends lieferte.

7.2.2 SPRINT 2: KERN IMPLEMENTATION

Im zweiten Sprint stellte Kevin zunächst für Kevin Jolitz die Login- und Registrationsoberfläche zur Verfügung. Im weiteren Verlauf des Sprints schrieb er die ersten Oberflächen für die Produkt-, Sprint-, Task- und UserStory-Übersicht.

7.2.3 SPRINT 3: FINALISIERUNG DER WEBSITE

Im nächsten Sprint fokussierte sich Kevin vor allem auf die grafischen Elemente der Projekt- und Kategorie Verwaltung. Darauf folgend verknüpfte er das Frontend für die Sprintdetailansicht mit dem Backend. Zuletzt implementierte er eine Post-It-Ansicht für das Backlog und schrieb Codekommentare.

7.2.4 SPRINT 4: PRÄSENTATION & DOKUMENTATION

Im letzten Sprint arbeitete Kevin an der Gestaltung der Projektpräsentation mithilfe von Joshua und verfasste sein persönliches Fazit.

7.3 LASSE JACOBS

7.3.1 SPRINT 1: DOKUMENTATION UND PROTOTYP

Im ersten Sprint arbeitete Lasse Jacobs in Zusammenarbeit mit Kevin W. an der Analyse von Konkurrenzprodukten. Zusammen mit der Gruppe ermittelte er User Stories für das Projekt und zuletzt erstellte er mit Kevin Jolitz die Architektur des Systems.

7.3.2 SPRINT 2: KERN IMPLEMENTATION

Im zweiten Sprint begann Lasse mit dem Datenmodell der Userstory und des PlanElements. Im weiteren Verlauf des Sprints entwickelte er das Backend für das

Backlog und verknüpfte das Frontend mit dem Backend für den Backlog und die UserStoryDetailansicht. Am Ende überarbeitete er Listen, Arrays und Sets im Backend und sorgte für eine Sortierung dieser in der Datenbank.

7.3.3 SPRINT 3: FINALISIERUNG DER WEBSITE

Zu Beginn des nächsten Sprints, überarbeitete Lasse die Userstorydetailansicht nach Joshuas design. Zunächst implementierte er dann das Backend des Notificationsystems, worauf er die Taskdetailansicht verknüpfte. Bis zum Ende des Sprints arbeitete er an einer zentralen und einheitlichen Gestaltung für die Detailansichten und die History. Ebenso verfasste er Codekommentare.

7.3.4 SPRINT 4: PRÄSENTATION & DOKUMENTATION

Im letzten Sprint arbeitete Lasse an den Arbeitsberichten der Projektmitglieder und verfasste sein persönliches Fazit.

7.4 TONI SERFLING

7.4.1 SPRINT 1: DOKUMENTATION UND PROTOTYP

Im ersten Sprint hat sich Toni Serfling unter anderem mit dem Vorgehensmodell Scrum beschäftigt und dieses analysiert. Zusammen mit der Gruppe ermittelte er User Stories für das Projekt. Zuletzt lag der Fokus für Toni auf der Architektur des Systems.

7.4.2 SPRINT 2: KERN IMPLEMENTATION

Im zweiten Sprint war Toni zunächst mit dem Aufsetzen des Projekts beschäftigt. Seine Hauptarbeitsbereiche in diesem Sprint waren das Scrumiverse Projekt- und Sprint Backend, speziell in der Erstellung dieser Scrum-Elemente. Am Ende des Sprints begann er mit der Implementation des Reporting-Systems mithilfe der Highcharts-API, welche jedoch zunächst nur ein Protoyp war und erst im nächsten Sprint voll realisiert wurde.

7.4.3 SPRINT 3: FINALISIERUNG DER WEBSITE

Wie bereits oben erwähnt lag der Fokus im dritten Sprint vor allem auf dem Reporting-System, welches durch einen Sprint Burndown-Chart realisiert wurde. Außerdem war er noch mit der Implementation des Backends für das Messaging System beschäftigt.

7.4.4 SPRINT 4: PRÄSENTATION & DOKUMENTATION

Im letzten Sprint arbeitete Toni an der Sammlung und Zusammenfassung der Projektfazits jedes Projektmitglieds.

7.5 JOSHUA WARD

7.5.1 SPRINT 1: DOKUMENTATION UND PROTOTYP

Im ersten Sprint arbeitete Joshua Ward an der Analyse des Scrumvorgangsmodells. Zusammen mit der Gruppe ermittelte er User Stories für das Projekt und zuletzt erstellte er mit Kevin W. einen Papierprototyp, der ein grobes Bild des Frontends lieferte.

7.5.2 SPRINT 2: KERN IMPLEMENTATION

Zu Beginn des zweiten Sprints, organisierte Joshua die Indexseite und die Navigation des Systems. Im weiteren Verlauf entwickelte er eine erste Version des Frontends für die Detailansicht eines Tasks und stellte eine Oberfläche für die Verwaltung von Projekten zur Verfügung.

7.5.3 SPRINT 3: FINALISIERUNG DER WEBSITE

Im dritten Sprint arbeitete Joshua an dem Dashboard und stellte ein finales Design für die Detailansichten zur Verfügung. Im restlichen Verlauf des Sprints entwickelte er das Frontend für Notifications und Messages und verknüpfte dieses auch mit dem Backend. Desweiteren verfasste er Codekommentare.

7.5.4 SPRINT 4: PRÄSENTATION & DOKUMENTATION

Im letzten Sprint arbeitete Joshua an der Gestaltung der Projektpräsentation mithilfe von Joshua und verfasste sein persönliches Fazit.

8 FAZIT

8.1 EINZELFAZITS

8.1.1 LASSE JACOBS

Ich habe wieder einmal viel gelernt im Laufe des Projektes. Einerseits den Umgang mit dem Spring-MVC Modell, JSTL, CSS, Hibernate und dem Vorgehensmodell Scrum. Andererseits die Projektarbeit allgemein, da alle Arbeitsschritte genau in Scrumdesk festgehalten werden mussten. Ebenso haben wir auch ein Tool für unser Repository verwendet: Github Desktop. Dieses „Programm“ ist am negativsten während der Projektarbeit aufgefallen. Ein schlechtes Commit/Sync-System, ein Mergingsystem bei dem immer die Angst besteht, Arbeit von Projektmitgliedern zu löschen sind nur einige Beispiele. Wenn das Repository das schlechteste am Projekt war, hat das aber auch was Gutes, denn das heißt, dass die restliche Projektarbeit gar nicht so schlimm gewesen sein kann. Sie war sogar ziemlich gut: Durch unser ehemaliges Projekt hatten wir uns schon ein wenig eingestimmt und haben dieses Mal besser zusammengearbeitet. Dies war auch nötig, da häufig mehrere an einer Funktion gearbeitet haben, sodass eigene Arbeit auf die von anderen angewiesen war, damit diese fertig gestellt werden konnte. Dennoch gab es ein wenig Probleme, wie z.B. die Priorisierung der Aufgaben oder das Zurückhalten bei der Arbeit, um nicht Userstories von anderen zu bearbeiten. Insgesamt konnte ich durch Arbeiten im Backend und im Frontend einen sehr guten Überblick über alle Benutzten Technologien sammeln. Ich freue mich, dass wir somit ein weiteres Projekt erfolgreich abschließen können.

8.1.2 KEVIN JOLITZ

Im Vergleich zum letzten Projekt im Fach Anwendung war Scrumiverse eine weitaus größere Herausforderung. Keiner von uns hatte wirklich Erfahrung im Bereich der Webentwicklung (geschweige denn vom Framework SpringMVC oder anderen Teilen dieses Projektes). Dadurch gestaltete sich die Planung, vor allem das Gestalten der Architektur und des Datenmodells, als schwierig. Dennoch haben wir als Gruppe diese Herausforderung gut gemeistert, jeder von uns hat viel gelernt und ein meiner Meinung nach sehr gutes Produkt abgeliefert. Ich für meinen Teil konnte meine Erfahrungen unter anderem vom Planen und umsetzen von Webanwendungen ausbauen.

Was die Gruppendynamik und Kommunikation betrifft, bin ich etwas weniger zufrieden. Teilweise wurden Absprachen nicht gehalten, Dinge zu spät hochgeladen und es entstanden meiner Meinung nach unnötige Probleme. Auch in Diskussionen war es anstrengend den Fokus bei den eigentlichen Themen zu halten, was auch irgendwie mein Gefühl stärkt, dass die Gruppe weitaus weniger motiviert war als beim letzten Projekt. Für zukünftige Projekte sollte ich als Projektleiter an vielen Stellen einfach strikter und den Leuten weniger Freiraum lassen. Ein gesundes Mittelmaß zu finden zwischen Freiraum und strikten Vorgaben ist schwierig und wird noch einiges an Erfahrung meinerseits benötigen.

Zusammenfassend hat mir aber auch dieses Projekt eine Menge Spaß gemacht und mir neue Perspektiven in diesem Bereich aufgezeigt. Ich könnte mir durchaus vorstellen, später im Webbereich zu arbeiten und mit entsprechenden Frameworks eindrucksvolle Webservices oder Webseiten zu entwickeln.

8.1.3 TONI SERFLING

Trotz des meiner Meinung nach gelungenen Ablaufs konnte ich mich über die gesamte Projektphase hinweg wenig für das Projekt begeistern. Dies lag jedoch wenig an der

Arbeit am Projekt an sich sondern eher am Thema und den verwendeten Technologien. Ich habe mich schon vor dem Start des Projekts nicht für Webtechnologien interessiert und SpringMVC und Hibernate speziell haben mir nicht dabei geholfen Begeisterung und Interesse zu wecken. Für mich war dies zu abstrakt und uninteressant und führte letztendlich dazu, dass ich mich nie wirklich für die Aufgabe motivieren konnte. Ich habe dadurch nur genau das gemacht, was von mir gefordert war und musste mir oft Hilfe einholen. Der Einblick in die Webentwicklung war trotz allem insofern für mich hilfreich, indem es mir bestätigt hat, dass mir dieser Bereich der Anwendungsentwicklung nicht liegt. Die Arbeit in der Gruppe bewerte ich als gut, jedoch verbesserungswürdig. Kommunikation war zwar gegeben, jedoch durch diverse Faktoren gehemmt, z.B. Alleingänge ohne Absprache. Unterschiedliche Wissensstände in den Themen Datenbanken und Webentwicklung führten öfters zu Missverständnissen und Missinterpretation von Fragen. Unsere straffere Organisation und klare Aufgabenverteilung war jedoch definitiv hilfreich und positiv zu bewerten. Insgesamt bewerte ich das Mittelstufenprojekt als durchschnittliche Erfahrung, welche durch die Themenwahl und verwendete Technologien negativ und durch das Teamwork positiv beeinflusst wurde.

8.1.4 JOSHUA WARD

Durch ein vorheriges Projekt hatte ich schon etwas Vorwissen im Bereich der Webtechnologie SpringMVC. Im Laufe dieses Projektes habe ich trotzdem noch sehr viel über Frontend-Technologien (CSS, JavaScript, jQuery), Scrum, SpringMVC, JSTL, Git und Hibernate gelernt. Mit der Projektplanung und dem Ablauf des Projektes bin ich voll und ganz zufrieden. Natürlich gab es einige Probleme, jedoch sind diese meiner Meinung nach aufgrund von zu hoch gesteckten Zielen und fehlendem Wissen entstanden. Dieses fehlende Wissen über Webtechnologien hat auch zur falschen Einschätzung des Zeitaufwandes geführt. Im Großen und Ganzen hat mir die Arbeit an diesem Projekt viel Spaß gemacht und mein Interesse an Webtechnologien geweckt.

8.1.5 KEVIN WESSELER

Wenn ich das gesamte Projekt noch einmal Revue passieren lassen, dann blicke ich trotz einiger Probleme im Commit-Management und sehr hoch gesteckte Ziele auf ein gelungenes Projekt.

Wir haben in diesem Projekt viel mehr den Fokus auf die Planung des Projektes mit der Projektgruppe gelegt. Dadurch konnte ich noch mehr in dieser Richtung lernen und fühle mich für spätere Projekte sicherer. Somit kann ich mich positiv äußern, da ich die Aufteilung der einzelnen Sprints sinnig fand und sollte es zu Verschiebungen von Umsetzungen kommen, konnten diese gut abgefangen werden. Da mein Schwerpunkt in diesem Projekt auf dem Frontend lag, habe ich mit CSS und JSTL viele neue Kenntnisse sammeln können und alte Erinnerungen wiederauffrischen. Trotz dem Schwerpunkt Frontend konnte ich auch, durch Kommunikation mit dem Backend und den Wöchentlichen Meetings, stückweit in diesem Bereich Erfahrungen sammeln und bei Ausfällen auch Aufgaben übernehmen.

8.2 GESAMTFAZIT

Insgesamt ist das Projekt zufriedenstellend verlaufen. Zwar konnten nicht alle geplanten Features umgesetzt werden, diese waren größtenteils jedoch nur Quality-of-Life Improvements und verhindern nicht die Nutzung des Produkts. Es wurden wertvolle Erfahrungen im Bereich Webentwicklung und –design und der Arbeit in der Gruppe gewonnen. Besonders im Bereich Gruppenkommunikation wurde deutliches Verbesserungspotential gesehen, es wurde generell nicht genug kommuniziert.

Gesondert zu erwähnen ist die Arbeit mit dem Github-Desktop Client, mit welchem die gesamte Gruppe unzufrieden war und welcher zukünftig nicht mehr verwendet werden wird.

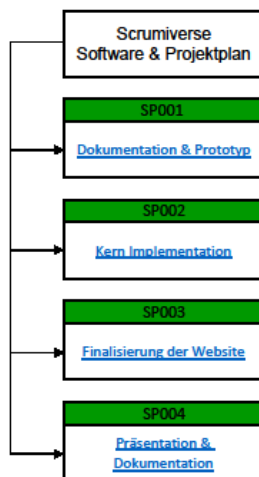
9 ANHANG

9.1 PROJEKTSTRUKTURPLAN

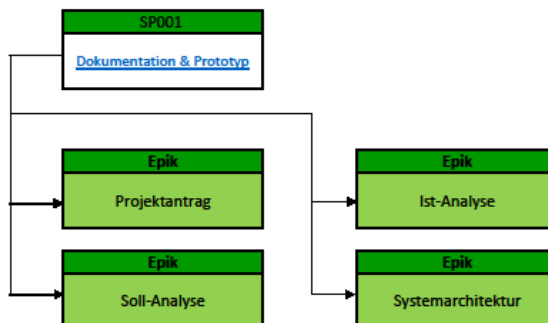
Erklärung



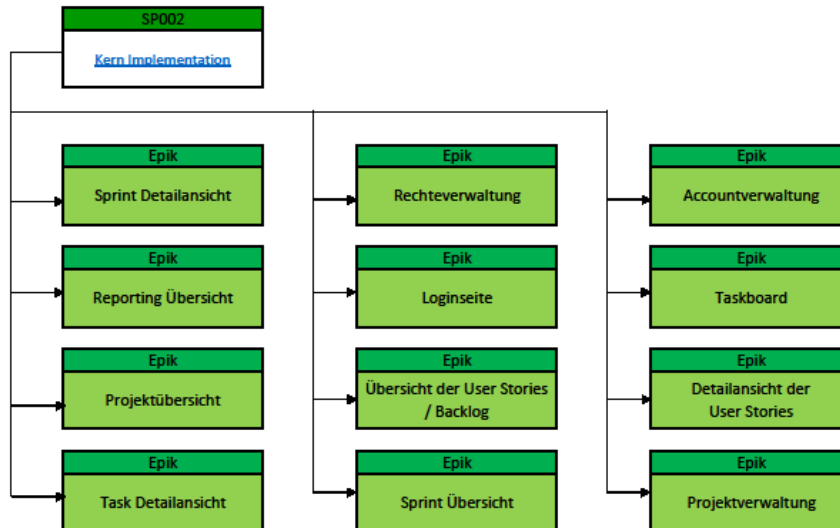
AStruktur 1 – Erklärung



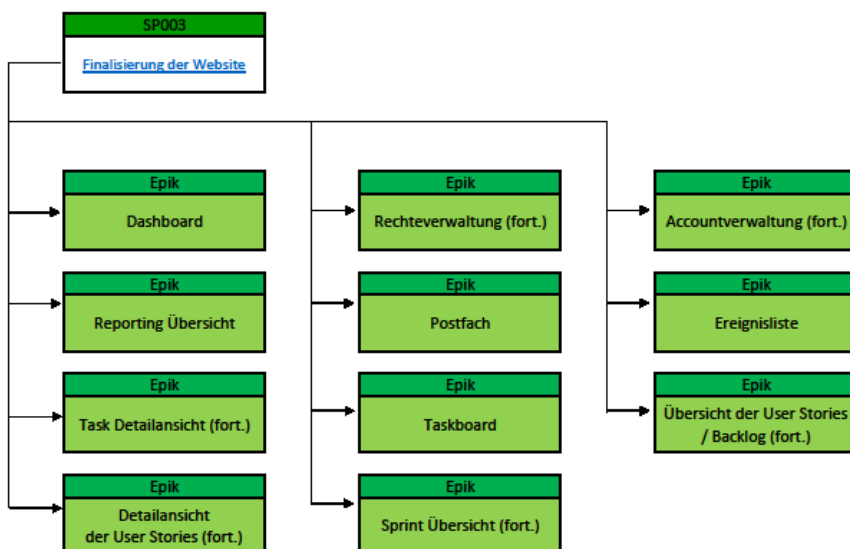
AStruktur 2 - Sprint Übersicht



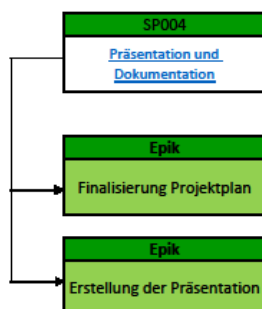
AStruktur 3 - Sprint 1



AStruktur 4 - Sprint 2



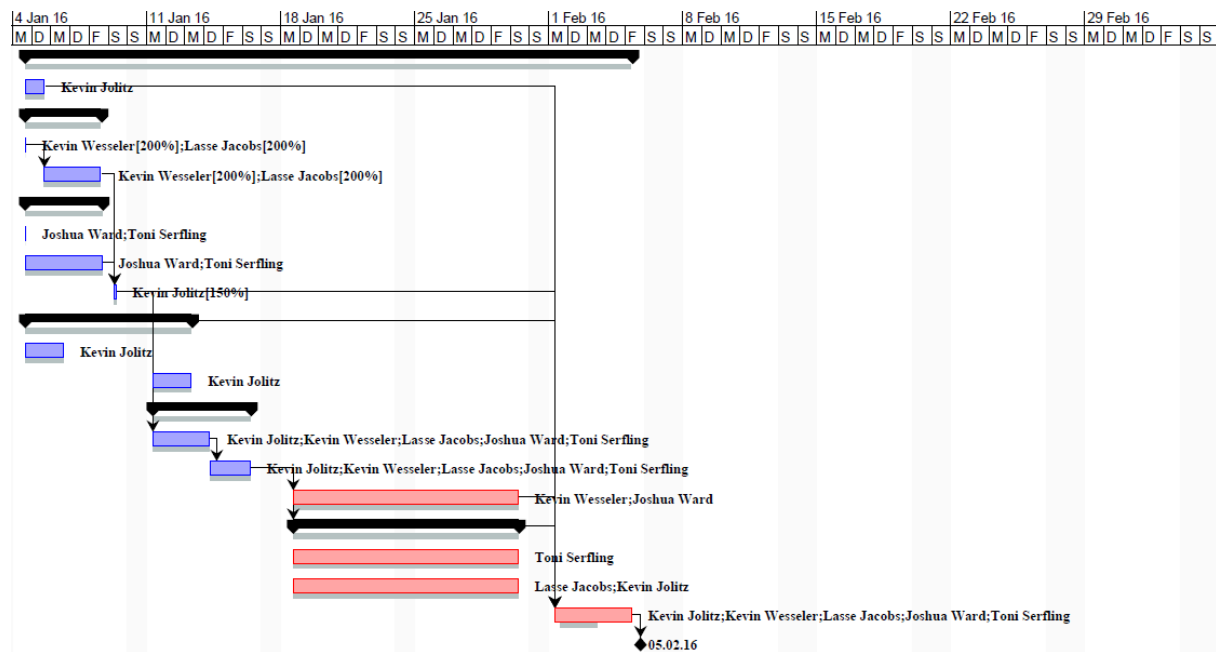
AStruktur 5 - Sprint 3



AStruktur 6 - Sprint 4

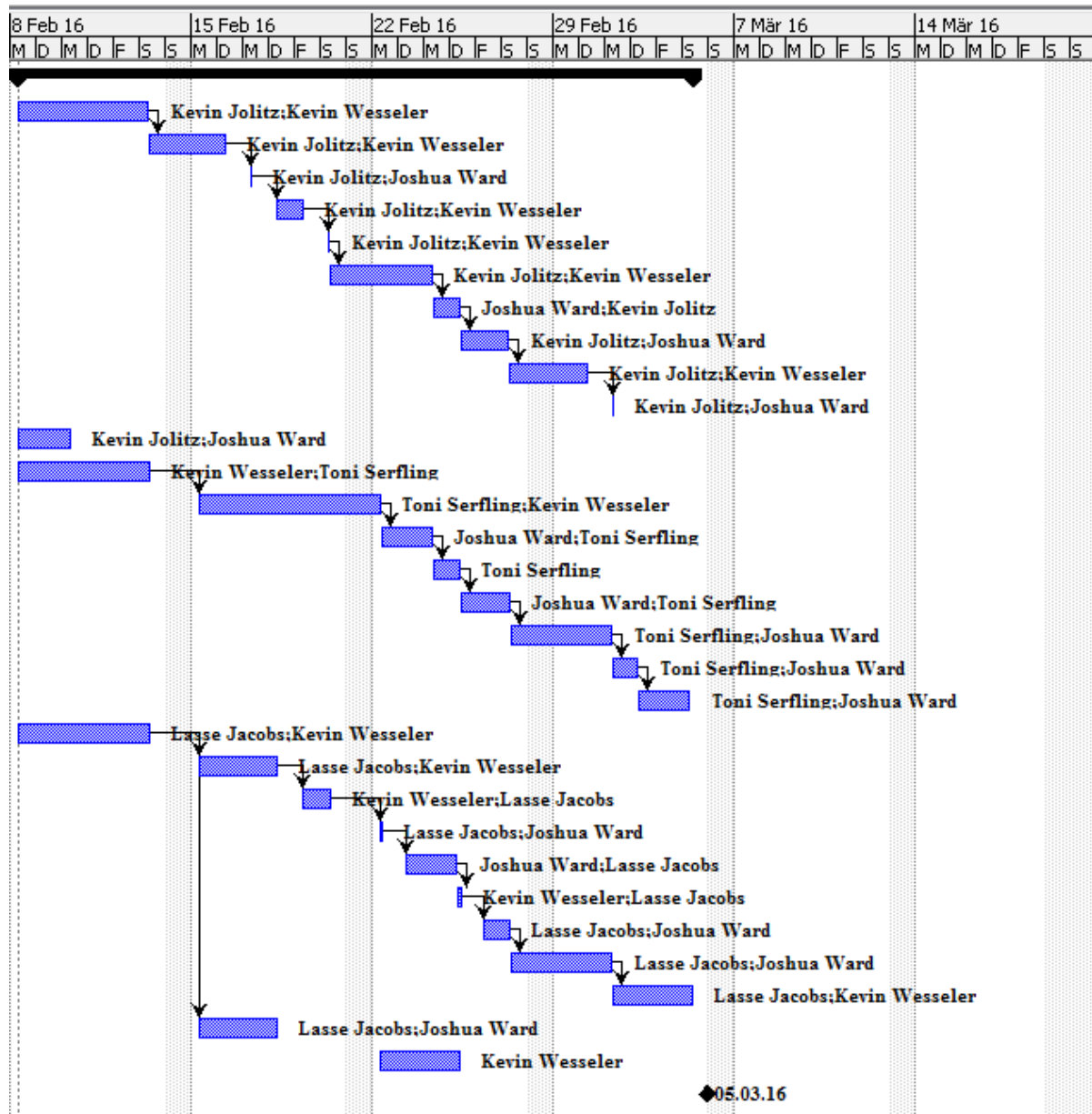
9.2 GANTT DIAGRAMME

	📅	Name	Dauer	Aufwand	Start	Ende
1		SP001 Papierprototyp & Projektplan		249 Stunden	04.01.16 16:00	05.02.16 09:30
2		US001 Projektantrag ausformulieren	2 Tage	2 Stunden	04.01.16 16:00	05.01.16 17:00
3		US002 Analyse von Konkurrenzprodukten	5 Tage	18 Stunden	04.01.16 16:00	08.01.16 16:30
4		T001 Ermitteln der Konkurrenzsoftware	2 Tage	4 Stunden	04.01.16 16:00	04.01.16 17:00
5		T002 Stärken und Schwächenanalyse der Konkurrenz	3 Tage	14 Stunden	05.01.16 16:00	08.01.16 16:30
6		US003 Analyse des Scrumvorgangsmodells	5 Tage	12 Stunden	04.01.16 16:00	08.01.16 17:00
7		T003 Interviews führen	1 Tag	2 Stunden	04.01.16 16:00	04.01.16 17:00
8		T004 Analyse und Darstellung des Workflows	4 Tage	10 Stunden	04.01.16 16:00	08.01.16 17:00
9		US004 Schreiben des Kapitels "Ist Analyse"	1 Tag	4 Stunden	09.01.16 08:00	09.01.16 10:40
10		US005 Formulieren des gewählten Vorgangsmodells	6 Tage	8 Stunden	04.01.16 16:00	13.01.16 09:00
11		T005 Erfassen der Toolsbaseline	3 Tage	3 Stunden	04.01.16 16:00	06.01.16 17:00
12		T006 Formatvereinbarungen ausarbeiten / vereinbaren	2 Tage	5 Stunden	11.01.16 08:00	13.01.16 09:00
13		US006 Programmelevante User Stories ermitteln	6 Tage	75 Stunden	11.01.16 08:00	16.01.16 10:30
14		T007 Ermitteln der User Stories	3 Tage	35 Stunden	11.01.16 08:00	14.01.16 08:30
15		T008 Spezifizieren der User Stories	3 Tage	40 Stunden	14.01.16 08:30	16.01.16 10:30
16		US007 Papierprototyp erstellen	12 Tage	32 Stunden	18.01.16 16:00	30.01.16 11:00
17		US008 Architektur erstellen	12 Tage	48 Stunden	18.01.16 16:00	30.01.16 11:00
18		T009 Erstellen des Datenmodells	12 Tage	16 Stunden	18.01.16 16:00	30.01.16 11:00
19		T010 Erstellen des Systementwurfs	12 Tage	32 Stunden	18.01.16 16:00	30.01.16 11:00
20		US009 Zusammenführen der Ergebnisse ggf. Korrektur	6 Tage	50 Stunden	01.02.16 08:00	05.02.16 09:30
21		Abgabe Projektplan		0 Stunden	05.02.16 18:30	05.02.16 09:30



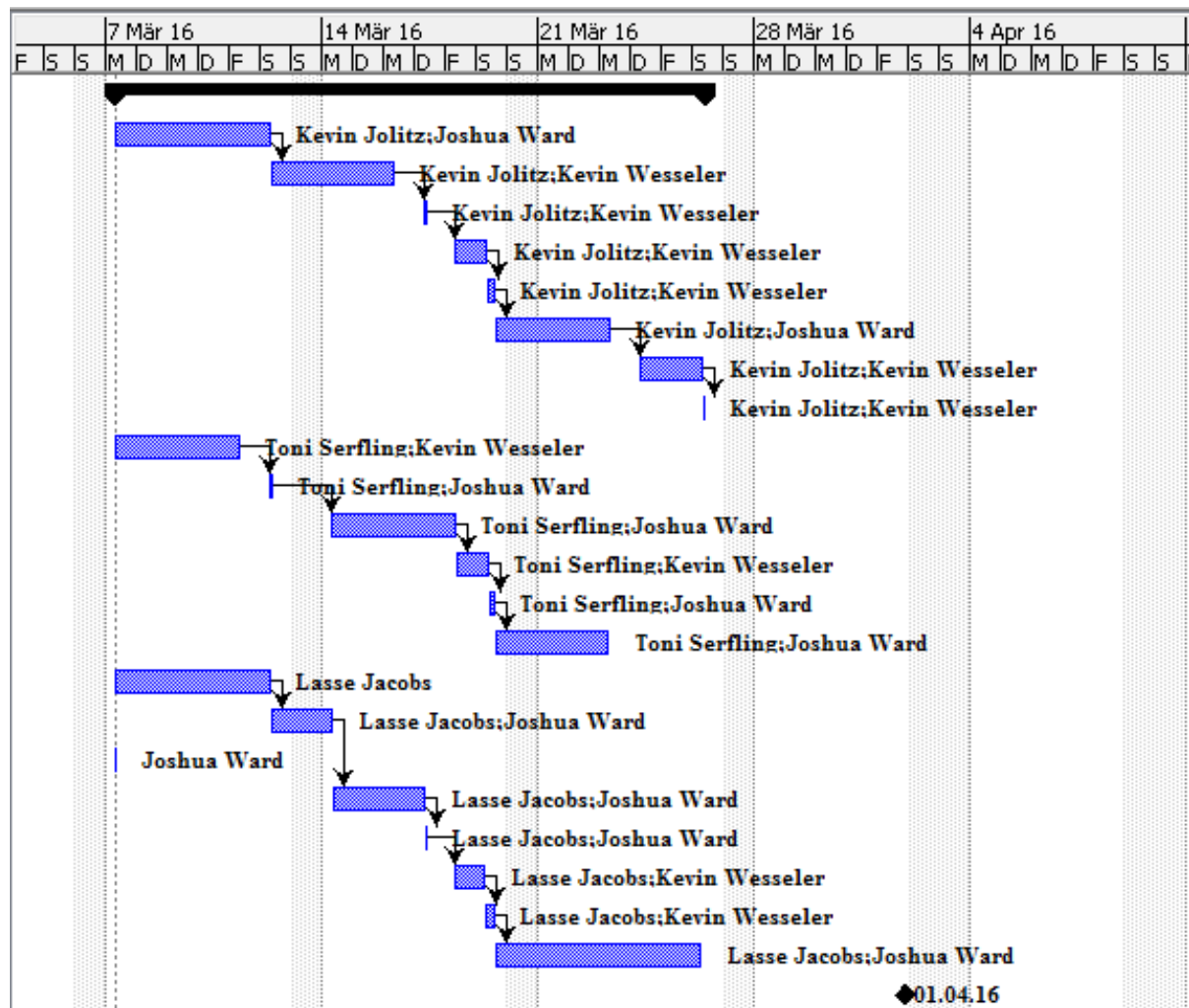
AGantt 1 - Sprint 1

Name	Dauer	Aufwand	Start	Ende
SP002 Kern Implementation		164,5 Stun...	08.02.16 08:00	05.03.16 10:30
US010 Der Benutzer will ein Account erstellen können.	5 Tage	10 Stunden	08.02.16 08:00	13.02.16 10:00
US014 Der Benutzer möchte sich einloggen können.	3 Tage	3,5 Stunden	13.02.16 10:00	16.02.16 09:00
US018 Der Projektadmin will Rollen verteilen können	1 Tag	2 Stunden	17.02.16 08:00	17.02.16 09:00
US025 Der Product Owner will eine Übersicht über alle erstellten Sprints	2 Tage	3 Stunden	18.02.16 08:00	19.02.16 09:00
US028 Der Product Owner will alle zu einem Sprint hinzugefügten User Stories anzeigen lassen.	1 Tag	3,5 Stunden	20.02.16 08:00	20.02.16 10:00
US029 Der Projektteilnehmer will eine neue Task erstellen können.	5 Tage	6,5 Stunden	20.02.16 10:00	24.02.16 09:00
US030 Der Projektteilnehmer möchte detaillierte Information zum Task in einem Popup sehen	2 Tage	4 Stunden	24.02.16 09:00	25.02.16 10:30
US031 Die Projektteilnehmer wollen Tasks verwalten können (RUD)	3 Tage	7 Stunden	25.02.16 10:30	27.02.16 08:30
US032 Der Projektteilnehmer will eine Übersicht seiner Tasks zu einer User Story (Task Board)	5 Tage	8,5 Stunden	27.02.16 08:30	01.03.16 09:00
US033 Der Projektteilnehmer will in der Detailansicht des Tasks diesen in verschiedene Phasen (TO DO, II	1 Tag	2 Stunden	02.03.16 08:00	02.03.16 09:00
US040 Der Projektteilnehmer will einer Task TAGS zu teilen können	3 Tage	4 Stunden	08.02.16 08:00	10.02.16 09:00
US011 Der Benutzer will ein Scrum-Projekt erstellen können	5 Tage	8,5 Stunden	08.02.16 08:00	13.02.16 11:00
US013 Der Product Owner will Sprints erstellen können.	8 Tage	9,5 Stunden	15.02.16 08:00	22.02.16 09:00
US016 Der Projektadmin will andere Benutzer einladen können	3 Tage	7 Stunden	22.02.16 09:00	24.02.16 09:00
US017 Der Projektadmin will vordefinierte Rollen benutzen können.	2 Tage	3 Stunden	24.02.16 09:00	25.02.16 10:30
US019 Der Benutzer will ein Projekt auswählen können und anschließend auf das Dashboard weitergeleit	3 Tage	6 Stunden	25.02.16 10:30	27.02.16 09:30
US034 Der Projektadmin will andere Benutzer ausladen können.	4 Tage	5 Stunden	27.02.16 09:30	02.03.16 08:30
US035 Der Projektadmin will das Projekt umbenennen können.	2 Tage	2 Stunden	02.03.16 08:30	03.03.16 08:30
US036 Der Projektadmin will ein Scrum-Projekt löschen können	3 Tage	3 Stunden	03.03.16 08:30	05.03.16 08:30
US012 Der Product Owner will eine User Story erstellen können.	6 Tage	8,5 Stunden	08.02.16 08:00	13.02.16 11:00
US015 Der Benutzer will eine Übersicht über alle seine zugehörigen Projekte.	4 Tage	6 Stunden	15.02.16 08:00	18.02.16 09:00
US020 Der Projektteilnehmer will eine tabellarische Übersicht über alle User Stories	2 Tage	6 Stunden	19.02.16 08:00	20.02.16 11:00
US021 Der Product Owner will User Stories verwalten können (RUD)	1 Tag	4 Stunden	22.02.16 08:00	22.02.16 11:00
US022 Der Projektteilnehmer möchte detaillierte Information zur User Story in einem Popup sehen	3 Tage	4 Stunden	23.02.16 08:00	25.02.16 08:30
US023 Der Projektteilnehmer will eine Post-It Übersicht über alle User Stories.	1 Tag	5 Stunden	25.02.16 08:30	25.02.16 12:30
US024 Der Product Owner will in der Detailansicht User Stories Sprints zuweisen können.	2 Tage	4 Stunden	26.02.16 08:00	27.02.16 09:30
US026 Der Product Owner will Sprints verwalten können (RUD)	6 Tage	6 Stunden	27.02.16 09:30	02.03.16 08:30
US037 Der Projektteilnehmer möchte ein Burn down/up chart über die verschiedenen Sprints haben.	4 Tage	8 Stunden	02.03.16 08:30	05.03.16 10:30
US039 Der Projektteilnehmer will eine History über alle Änderungen einer User Story	4 Tage	7 Stunden	15.02.16 08:00	18.02.16 09:00
US027 Der Product Owner will eine ausklappbare, tabellarische User Story Übersicht um diese den Sprint	4 Tage	8 Stunden	22.02.16 08:00	25.02.16 10:30
Sprint Ende		0 Stunden	05.03.16 23:59	05.03.16 11:00



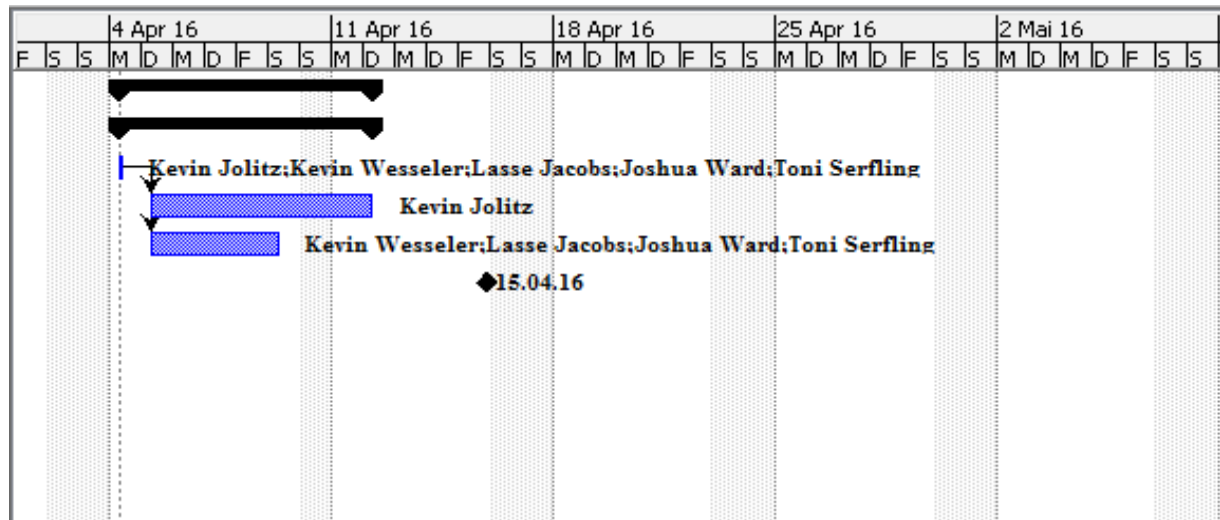
AGantt 2 - Sprint 2

Name	Dauer	Aufwand	Start	Ende
SP003 Finalisierung der Website		129,25 Stund...	07.03.16 08:00	26.03.16 11:00
US038 Der Projektteilnehmer will eine History über alle Changes einer Task	6 Tage	10 Stunden	07.03.16 08:00	12.03.16 09:00
US042 Der Product Owner will Kategorien erstellen und editieren können (Farbe)	5 Tage	12 Stunden	12.03.16 09:00	16.03.16 09:30
US043 Der Product Owner will User Stories in eine (vordefinierte) Kategorie einordnen können.	1 Tag	4,5 Stunden	17.03.16 08:00	17.03.16 11:00
US050 Der Projektadmin will Rollen definieren können.	2 Tage	6,5 Stunden	18.03.16 08:00	19.03.16 10:00
US049 Der Projektadmin will Rechte an selbstdefinierte Rollen verteilen können	1 Tag	6,5 Stunden	19.03.16 10:00	19.03.16 16:00
US057 Der Projektteilnehmer will seine Zeitverschreibung über den Task loggen können. (In welcher Zeit welche Änderung)	5 Tage	4,75 Stunden	19.03.16 16:00	23.03.16 09:00
US059 Die Projektteilnehmer wollen eine Übersicht über die einzelnen Zeiten (Geplant, Verbraucht, Verbleibend) aller Teilnehmer einer Task	3 Tage	4,75 Stunden	24.03.16 08:00	26.03.16 10:00
US060 Die Projektteilnehmer wollen in den Tasks deren zeitlichen Aufwand schätzen können.	1 Tag	1,75 Stunden	26.03.16 10:00	26.03.16 11:00
US041 Der Benutzer möchte Accounteinstellungen vornehmen können	5 Tage	9 Stunden	07.03.16 08:00	11.03.16 09:00
US044 Der Benutzer möchte einstellen können, ob er über Ereignisse per E-Mail informiert wird.	1 Tag	4 Stunden	12.03.16 08:00	12.03.16 11:00
US051 Der Projektteilnehmer möchte eine Nachricht schreiben, lesen und löschen können	5 Tage	13 Stunden	14.03.16 08:00	18.03.16 09:00
US054 Der Projektteilnehmer möchte innerhalb der detaillierten User Story Ansicht dazugehörige Tasks sehen (Auflistung)	2 Tage	3,75 Stunden	18.03.16 09:00	19.03.16 10:30
US055 Der Projektteilnehmer will User Stories filtern können.	1 Tag	4,75 Stunden	19.03.16 10:30	19.03.16 15:30
US058 Der Projektteilnehmer will Tasks filtern können.	5 Tage	4,75 Stunden	19.03.16 15:30	23.03.16 08:30
US045 Der Projektteilnehmer möchte alternativ Ereignisbenachrichtigungen per E-Mail erhalten.	6 Tage	6 Stunden	07.03.16 08:00	12.03.16 09:00
US046 Der Projektteilnehmer möchte über Änderungen (Changes und Kommentare) an seiner User Story / Task informiert werden	3 Tage	5 Stunden	12.03.16 09:00	14.03.16 10:00
US047 Der Scrummaster möchte über Änderung der Aufwandsschätzung einer Task des aktuellen Sprint informiert werden	1 Tag	1 Stunde	07.03.16 08:00	07.03.16 09:00
US048 Der Scrummaster möchte über Änderung einer User Story / Task des aktuellen Sprint informiert werden	4 Tage	6,75 Stunden	14.03.16 10:00	17.03.16 10:00
US052 Der Projektteilnehmer möchte Ereignisse als gelesen markieren können.	1 Tag	2 Stunden	17.03.16 10:00	17.03.16 11:00
US053 Der Projektteilnehmer möchte in einer visuellen Darstellung die Sprints in eine Gegenüberstellung von geplanter / tatsächlicher Zeit	2 Tage	2,75 Stunden	18.03.16 08:00	19.03.16 08:30
US056 Der Projektteilnehmer möchte per Drag n Drop einen Task in verschiedene Phasen einteilen können.	1 Tag	6,75 Stunden	19.03.16 08:30	19.03.16 15:30
US061 Der Projektteilnehmer will nach Projektauswahl ein Dashboard mit Elementen aus anderen Boards	8 Tage	9 Stunden	19.03.16 15:30	26.03.16 08:30
Sprint Ende		0 Stunden	01.04.16 22:59	01.04.16 09:00

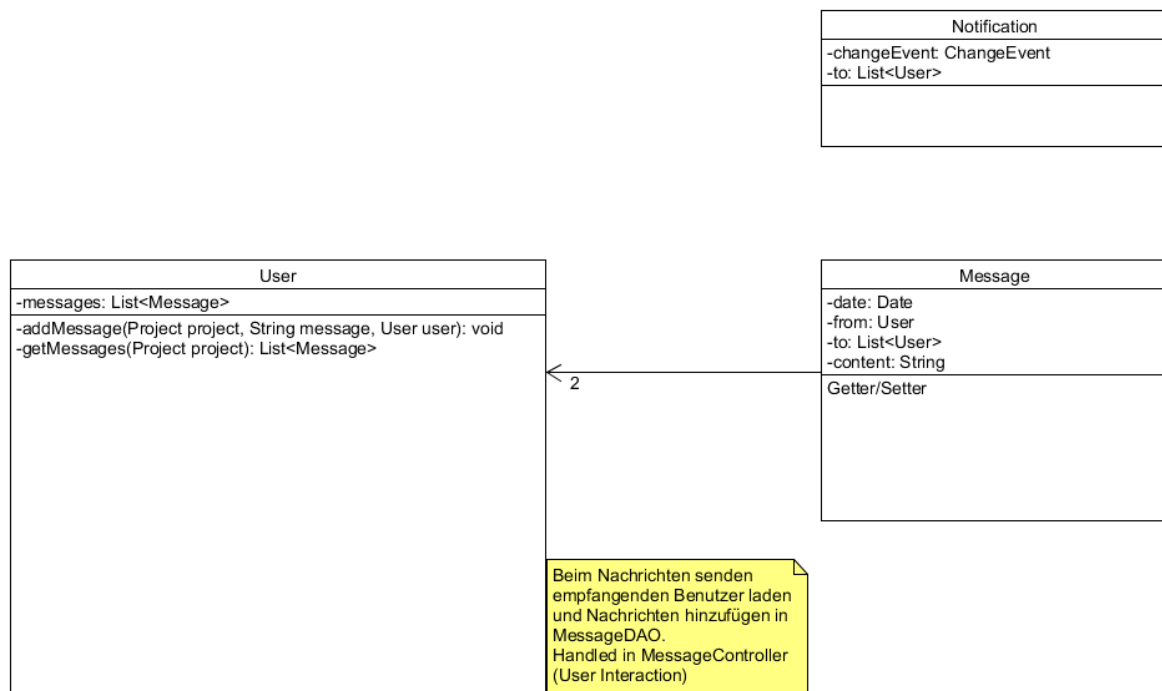


AGantt 3 - Sprint 3

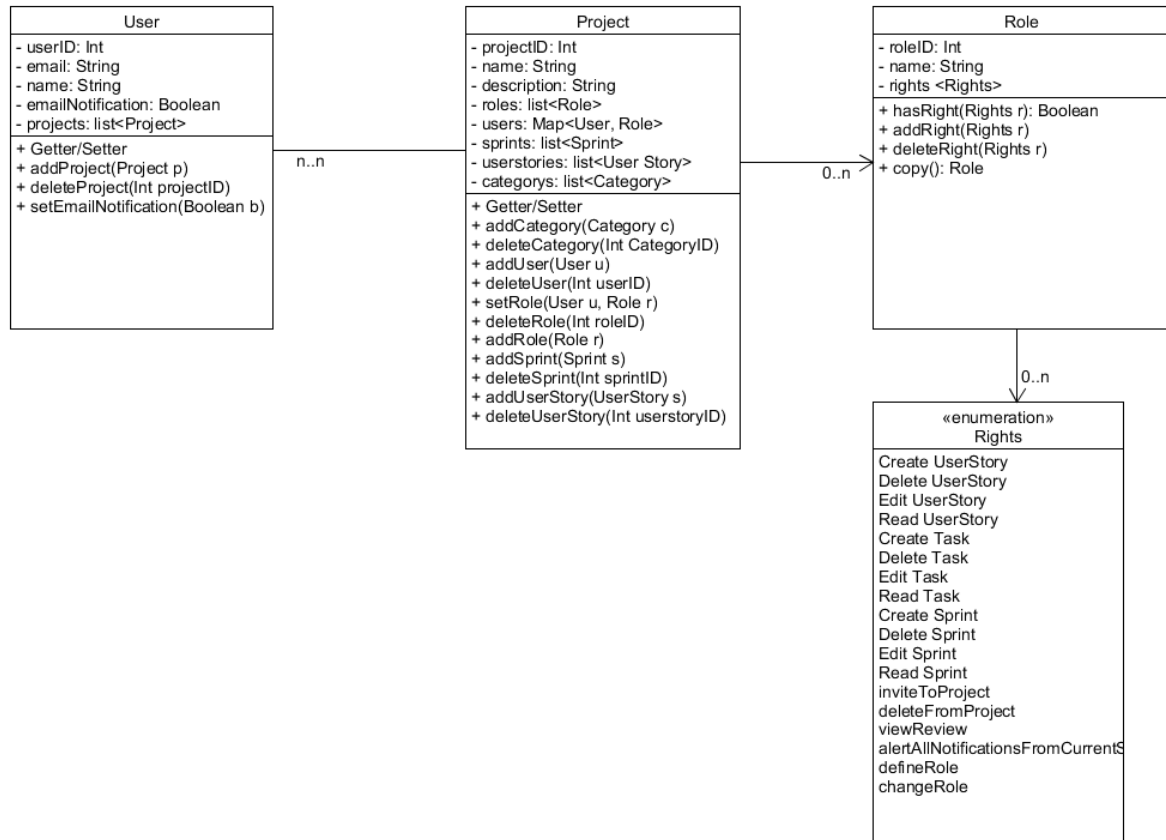
Name	Dauer	Aufwand	Start	Ende
SP004 Präsentation & Dokumentation		68 Stunden	04.04.16 08:00	12.04.16 08:30
US062 Projektdokumentation fertigstellen		28 Stunden	04.04.16 08:00	12.04.16 08:30
Teambesprechung Projektfazit		15 Stunden	04.04.16 08:00	04.04.16 11:00
Projektdokumentation finalisieren		13 Stunden	05.04.16 08:00	12.04.16 08:30
US063 Präsentation erstellen		40 Stunden	05.04.16 08:00	09.04.16 09:30
Sprint Ende		0 Stunden	15.04.16 22:59	15.04.16 09:00



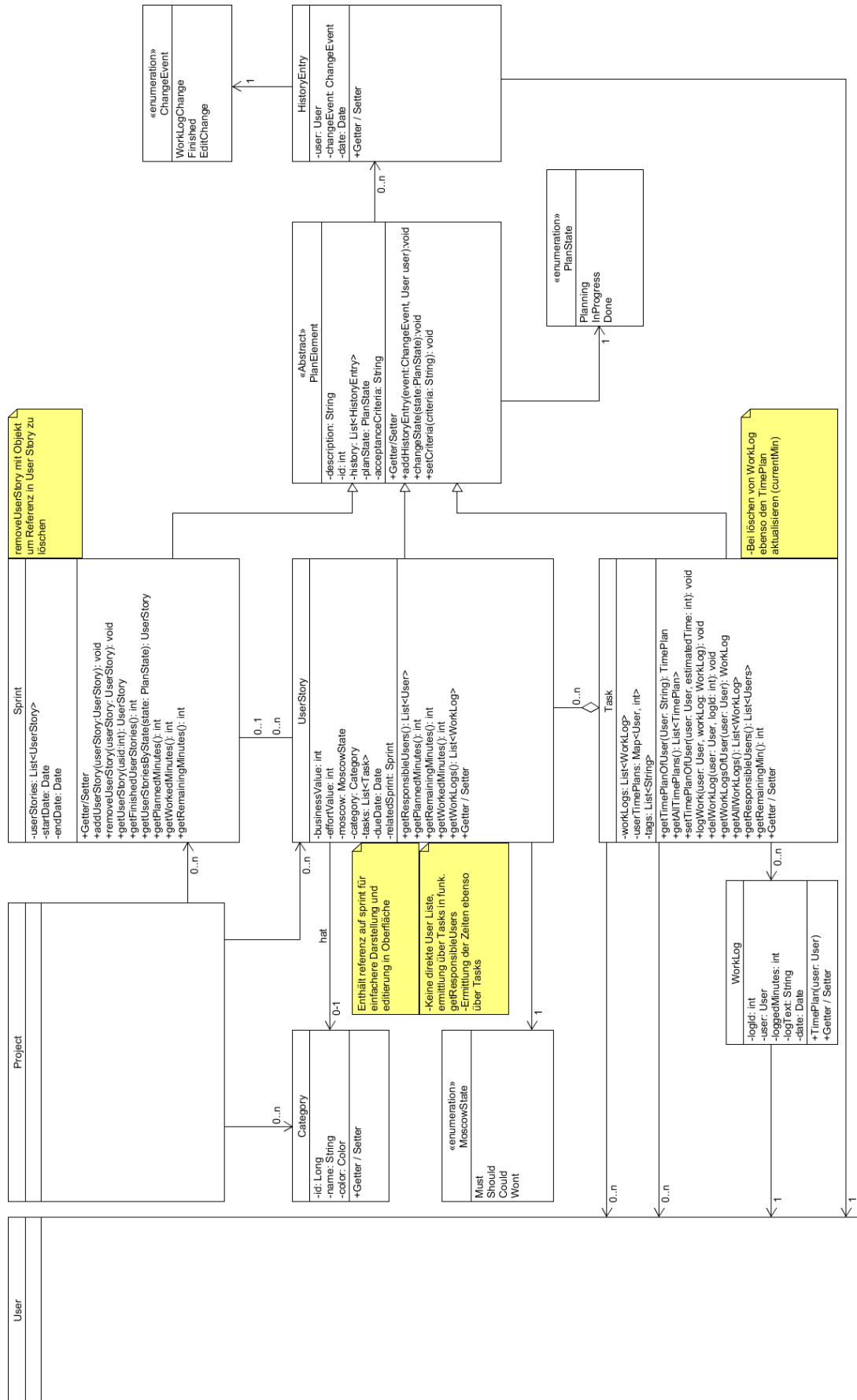
9.3 DATENMODELL



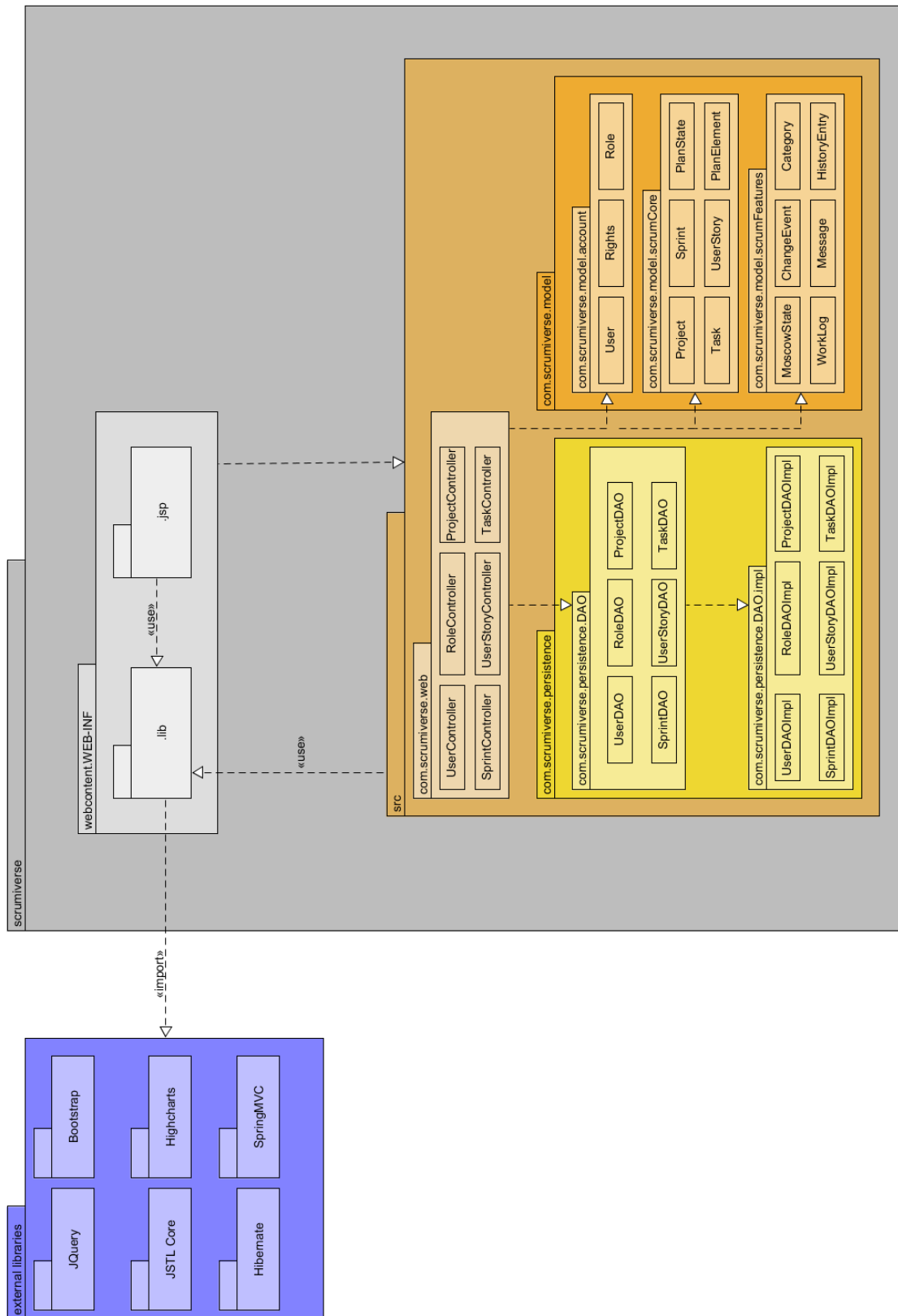
AData 1 - Message und Ereignis Modelle



AData 2 - Benutzerdaten



9.4 SYSTEMARCHITEKTUR



ASystem 1 - Overview

9.5 PAPIERPROTOTYP

ID ## - USERSTORY - HEADER	TIME	MEMBERS	CATEGORY SPRINT	TIME STATS MOSCOW VALUE RISK	EFFORT	
ID 001 - USERSTORY	TIME	6+	CATEGORY SPRINT	## / ## / ## M 100 40	40	
ID 001 - USERSTORY	TIME	6+	CATEGORY SPRINT	## / ## / ## M 100 40	40	⚙
ID 001 - USERSTORY	TIME	6+	CATEGORY SPRINT	## / ## / ## M 100 40	40	>

ID 001 - USERSTORY	TIME	6+	CATEGORY SPRINT	## / ## / ## M 100 40	40	⚙
ID 001 - USERSTORY	TIME	6+	CATEGORY SPRINT	## / ## / ## M 100 40	40	>

SAND UHR BEST ICON EVER - ZEIGT DUE DATE ALS SAND

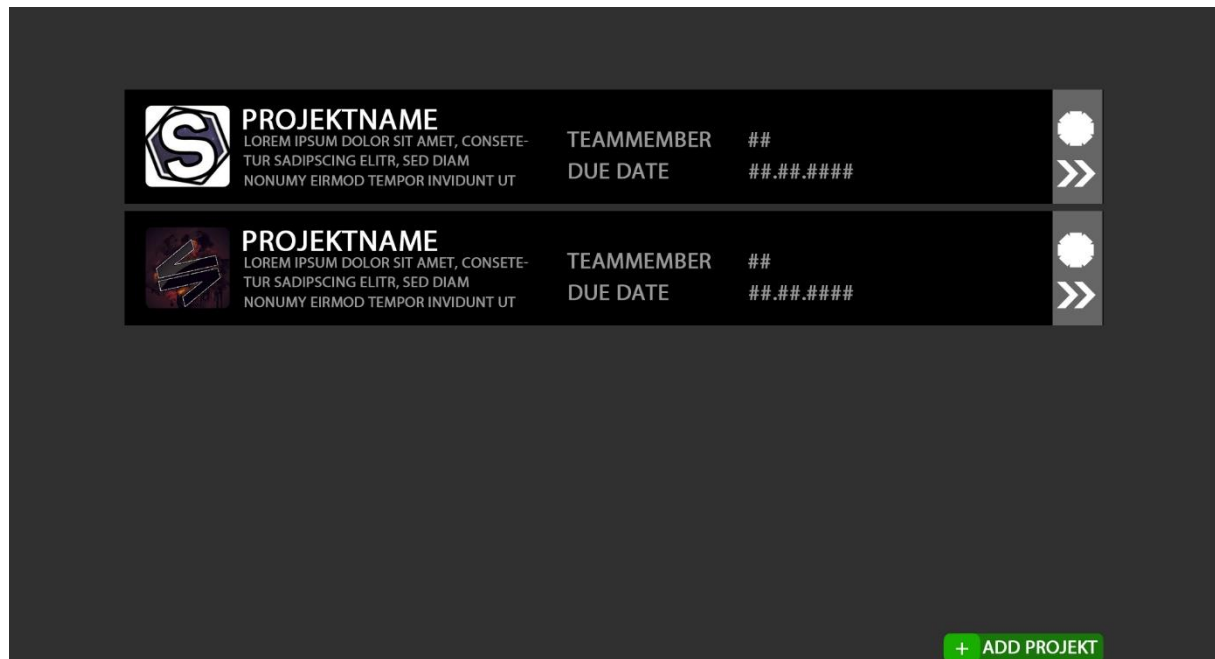
SAND UHR BEST ICON EVER - ZEIGT DUE DATE ALS SAND

+ ADD STORY ↗

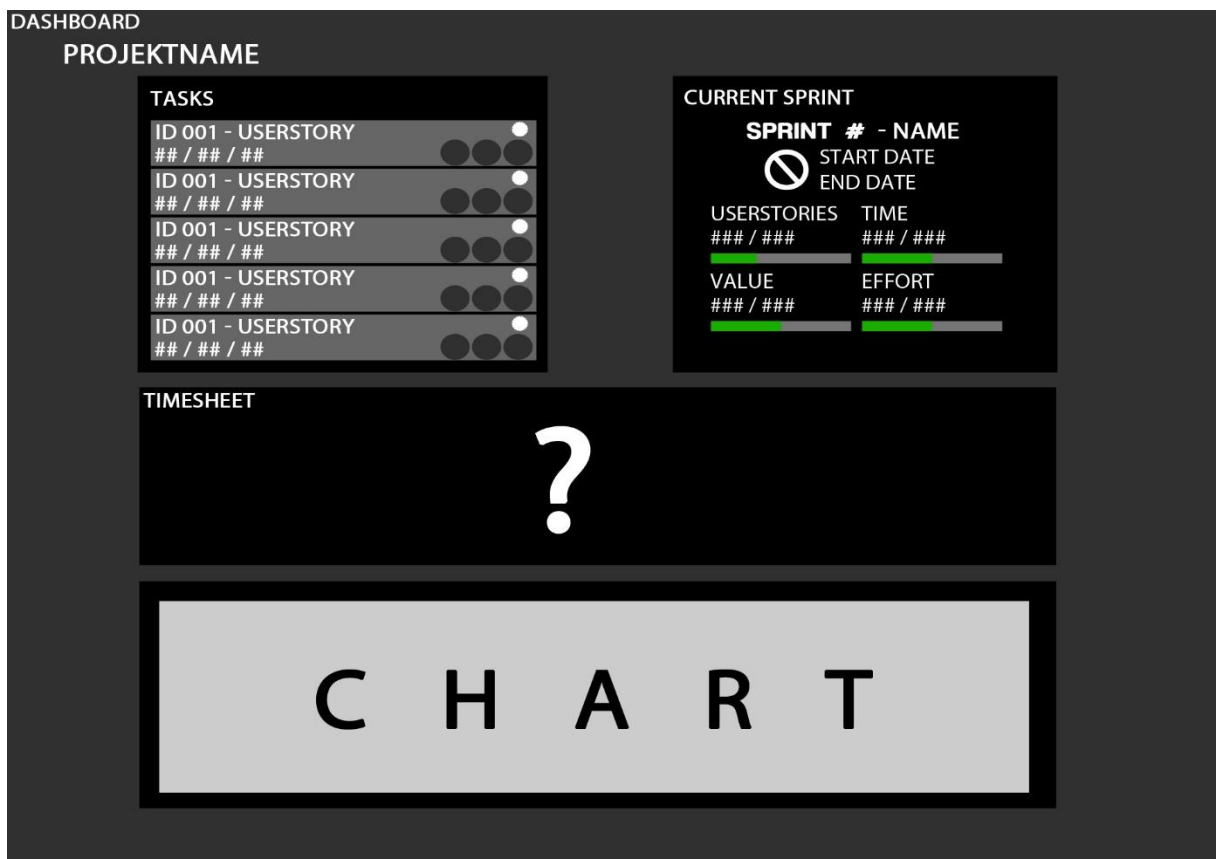
AProto 1 - Userstory Übersicht



AProto 2 - Menüführung



AProto 3 - Projektwahl



AProto 4 - User Stories

RECHTEVERWALTUNG

HEADING

MEMBERROLE
MEMBERROLE
MEMBERROLE

RECHTENNAME

RECHT ☐

RECHT ☐

RECHT ☐

RECHT ☐

RECHT ☐

RECHT ☐

SAVE

+ ADD ROLE

AProto 5 - Rechteverwaltung

PROJEKTNAME

LOREM IPSUM DOLOR SIT AMET, CONSETETUR SADIPSCING ELITR, SED DIAM NONUMY
EIRMOD TEMPOR INVIDUNT UT LABORE ET DOLORE MAGNA ALIQUYAM ERAT, SED DIAM
VOLUPTUA. AT VERO EOS ET ACCUSAM ET

KLICK = FORMULAR

BENUTZERNAME

EVTL DATA: ## ☐

ROLLE

BENUTZERNAME

EVTL DATA: ## ☐

ROLLE

BENUTZERNAME

EVTL DATA: ## ☐

ROLLE

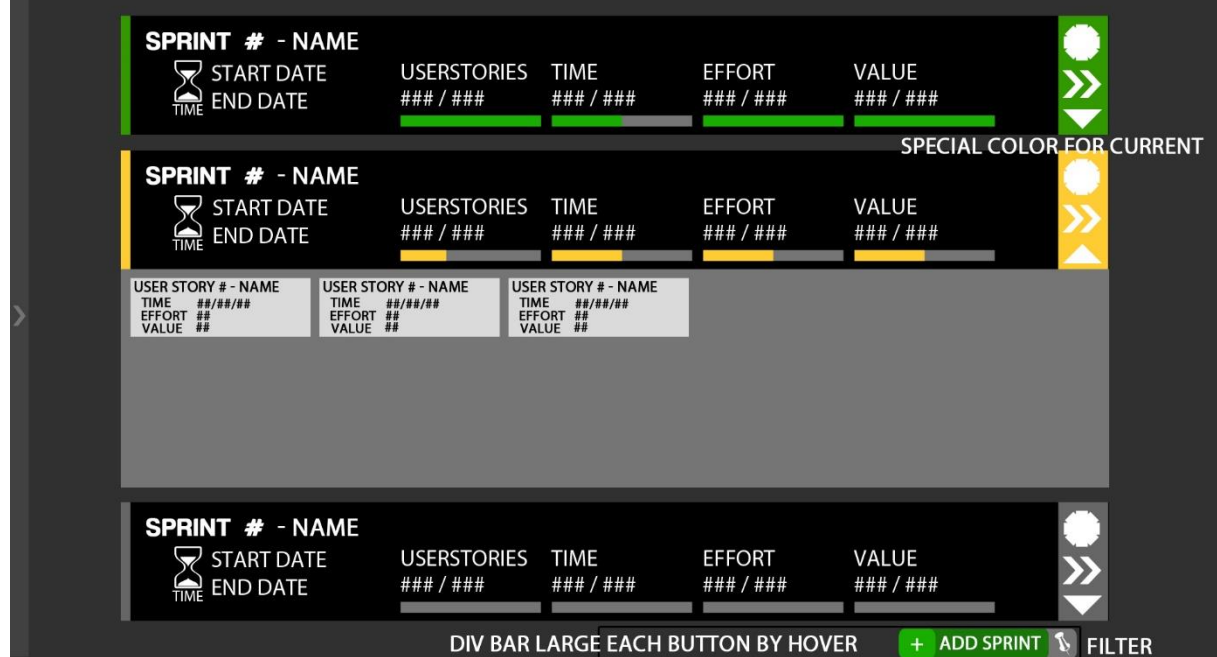
SAVE

+ ADD MEMBER

AProto 6 - Projektverwaltung

27

SPRINT ÜBERSICHT



AProto 7 - Sprint Übersicht



AProto 8 - Sprint Übersicht 2

SPRINT ÜBERICHT

SPRINT # - NAME

SPRINTNAME

DATA: ### ← FORM ZUM EDITIEREN
ALLE VALUES IN FORMS / DROPDOWNS

SAVE CANCEL

DIV BAR LARGE EACH BUTTON BY HOVER

+ ADD SPRINT B FILTER

AProto 9 - Editierungsformulare

	TODO	IN PROGRESS	DONE
ID 001 - USERSTORY	ID 002 - TASK ## / ## / ##	ID 001 - TASK ## / ## / ##	ID 001 - TASK ## / ## / ##
	ID 003 - TASK ## / ## / ##		

M 100 40
TIME ## / ## / ## 40 6+

+ ADD TASK

AProto 10 - Task Übersicht