

# Contents

<b>1 Begrüßung</b>	<b>1</b>
<b>2 Letzte Woche</b>	<b>1</b>
<b>3 Rückfragen</b>	<b>1</b>
<b>4 Personenbezogenes</b>	<b>1</b>
<b>5 jobs</b>	<b>2</b>
<b>6 Requirements Sammeln</b>	<b>2</b>
<b>7 Ideen Sammeln, was man machen kann</b>	<b>2</b>
<b>8 Wie</b>	<b>3</b>
<b>9 Zeitplan Skizzieren</b>	<b>3</b>
<b>10 The project plan</b>	<b>3</b>

## 1 Begrüßung

Hallo, bla wie war eure Woche?, wer macht Protokoll? Vorstellen was wir heute machen; Inhaltsverzeichnis stellt das dar.

## 2 Letzte Woche

Jeder trägt kurz vor, was er seit dem letzten Treffen gemacht hat, thematisch zusammengefasst. Google Group erklären, Google Drive erklären. Haben alle Zugriff? Wenn nein nach dem Treffen lösen, sollte es ein Gruppenproblem sein gleich machen.

Wenn möglich kleine Präsentation über die Funktionen von Git und wie man es installiert/benutzt. Agreedo

Bibliotheken erklären, Fabian, Matthias, Matthias

Wichtige Punkte dabei:

- Schnittstellen
- Funktionen der Bibliotheken
- Welche Interfaces werden zur Verfügung gestellt
- 

## 3 Rückfragen

Welche Fragen haben sich beim Bearbeiten der Tasks ergeben? Welche Empfehlungen habt ihr an die anderen?

## 4 Personenbezogenes

Nochmal wer hat wann keine Zeit? Doodle für einen regelmäßigen Termin machen, wöchentliches Treffen wäre optimal, der de Ridder will da afaik dabei sein(?).

Urlaube, Festivals, Geburtstage? Wann hat wer keine Zeit? Ich bin über Southside (Wochenende 21.06.) nicht da, hab eine Klausur.

## 5 jobs

- **Projektmanager** - Leonard - Sorgt dafür, dass alle aufgaben innerhalb des Zeitplanes gemacht wird. Risikoabschätzung. Änderungsmanagement.
- **Versions-(Datei-)Manager** - Rebecca - Sorgt dafür, dass die Daten sicher gespeichert sind und in der richtigen Version vorliegen. Struktur der Daten festlegen. Sorge tragen, dass die Struktur eingehalten wird (nicht alles selbst organisieren, sondern Leute auf Fehler hinweisen und Lösung erarbeiten)
- **Dokumentenmanager** - frei (Matthias Miller?) - Sorgt für einheitliche, lückenlose Dokumentation. **Wichtig, da davon die Note abhängt!!!** (Die Aufgabe ist nicht die Dokumentation alleine zu erstellen, sondern Format festlegen und Leute dazu antreiben die Doku zu erstellen)
- **Spezifikationsmanager** - frei- Experte für Requirements  
Verantwortlich für Requirements und Änderungen  
SDD / SRS prüfen, verantwortlich für die Erstellung dieser  
Sicherstellen, dass die Spezifikationen eingehalten werden.
- **Testmanager** - frei - Experte für Testing.  
Sicherstellen, dass alles mindestens ein mal getestet wurde. Enge Zusammenarbeit mit Spezifikationsmanager. Festlegen wie getestet wird, Tests überprüfen.
- **Software support** 'The expert on development tools. Installs (or helps in- stalling) them and provides knowledge on their proper usage to the other team members.'
- **Library expert** Knows how to use a particular library and its strengths and weaknesses. **SWING, ISGCI, JgraphX**
- Weitere Jobs? Möglich: Experten für Design, Codereview... (ich weiß nicht in wie weit diese Jobs essenziell sind oder ob es Sinn ergibt sie speziell zuzuweisen, da wir alle das wissen haben sollten. Eine zuweisung ergibt insofern sinn, dass dann nicht alle in die Tiefe müssen und man einen Ansprechpartner hat.)

## 6 Requirements Sammeln

**Hier beginnt der Hauptteil des Meetings** Alle sollten welche herausgearbeitet haben, wenn nicht dann gemeinsam erarbeiten, PDF liegt auf Google Drive. Man kann es auch auf der Seite <http://www.informatik.uni-konstanz.de/ag-ernst-de-ridder/lehre/> erreichen, wenn man sich anmeldet. Funktionale und nicht funktionale Anforderungen. Siehe Seng Folien!!

## 7 Ideen Sammeln, was man machen kann

Alle Ideen sind gut und werden ohne Kritik aufgenommen und diskutiert.

Was wären wünschenswerte Features

Zooming, möglichst wenige Überschneidungen von Kanten, Interaktivität (Super- oder Subknoten ein-/ausblenden, Knoten färben...

Was kann das Framework?

Wichtig um zu den Zeitaufwand zu bewerten.

Dabei Machbarkeit im Kopf behalten.

## 8 Wie

Wie können die Spezifikationen und Features umgesetzt werden? Wie viel Zeit wird (geschätzt) gebraucht. Entweder Stunden/ Wochen als Einheit oder loc (Lines of code).

## 9 Zeitplan Skizzieren

Als Prozessmodell schlage ich vor Rapid Prototyping mit dem Wasserfallmodell zu verbinden, Dadurch stellen wir eine gute Dokumentation sicher und minimieren dabei das Risiko. Überlegt welche Tasks man parallel ausführen kann, was nach was kommen muss. Ordnet alles auf einer Timeline an. Möglichst alle 1-2 Wochen einen Milestone einführen. Feinkörnige Planung für die ersten 1-2 Wochen, dann grob. Feste Termine:

- Dienstag 30.4. Zeitplan steht fest
- Dienstag (frühestens) **30.4.** - (spätestens) 7.5. SRS erste Revision ist fertig
- Iterationen mit Mockups um festzustellen, wie das graphische Frontend aussehen soll. Wenn der erste Prptotyp alles erfüllt Requirements prüfen und SDD schreiben.
- Dienstag 14.5. - 21.4. SDD, erste Revision ist fertig
- Dienstag 2.7. Akzeptanztest und Deployment im Erfolgsfall

Klingt das realistisch? Sonst können wir uns auch mehr Zeit lassen. Es hängt auch davon ab wie sicher sich der Ridder ist, was er will.

## 10 The project plan

Nach Section 1.4 in der Aufgabenstellung den Projektplan anfertigen (bitte noch nicht einreichen)