**Meetingprotokoll:**

|  |  |
| --- | --- |
| Beginn: | 10.04.17 19:00 |
| Ende: | 10.04.17 21:00 |
| Ort: | TeamSpeak |
| Anwesende: | Martin Klampfer, Benjamin Schaden, Alexander Schneider, Calvin Claus, Christopher Glantschnig |
| Abwesende: | / |
| Getroffenen Entscheidungen: | Beispiel für generelle Aufteilung in Redmine:  Feature „Authentifizierung“  Sub-Feature: Accountsperre nach falscher Passworteingabe  besteht aus Tasks  .) z.B: wenn ausgesperrt, darf sich nicht einloggen  .) weiterer Task für 5x falsch eingeben --> sperren  .) Tests...usw  Wichtige Punkte, mit denen beginnen wir:   * Ticket * Veranstaltung * Kunden   Datenbankmodell-Entwurf erstellen  Schaden: erstellt Grundstruktur  Erweiterte User-Stories - diese haben wir festgelegt, die wir implementieren müssen:   * Zahlungsabwicklung <- Schneider und Schaden interessieren sich dafür * Benutzerverwaltung   Jetzt ca. ein Sprint bis zum MR 1  da wird voraussichtlich gemacht:  .) Kunde anlegen  .) Kunde bearbeiten (beides mit UI - nicht vergessen auf auslagern der Texte im UI)  .) Kunden-Demo-Demo-Daten erstellen  .) Demo-Daten für Veranstaltungen erstellen  Tasks anlegen: (Grobstruktur)  ER Diagramm erstellen  Service erstellen  UI erstellen  Kunde anlegen Tests  Kunde bearbeiten Tests   * alles Codebezogene (JavaDOC, Logger, Git Commit Messages) auf Englisch * alles in Redmine auf Deutsch * was wird geloggt: Ticketreservierungen, einloggen oder so, eher Service Sachen   zu Github - Branching:   * User Stories sind Features - 1 Person ist immer zuständig für ein Feature (Ansprechpartner dazu)   die zuständige Person erstellt sich selbst die (Sub) Tasks in Redmine dafür  jedes Feature in Redmine wird zu einem Branch - also z.B: Kunden anlegen, Kunde bearbeiten werden zu Branches  Unterpunkte von User Stories sind Features, alles darunter Tasks |
| Ergebnis des Meetings oder des Reviews: |  |