BSD

# 2. Iteration

### Abgabetermin

17. Dezember 2015

### Aufgabenstellung

#### WICHTIG: Events

Für alle Schichten, werden in diesem Sprint die Events implementiert.

##### Generell

Idee: Profilseiten   
Clan und User haben infoseite mit custom text und wappen etc.

##### Client

Kommandos geben, Benachrichtigungen für auflösungen anzeigen („Du wurdest angegriffen. Hier Kampfergebnis anschauen.“). Anzeige von Truppen auf dem Weg. Anzeige der Truppen in deinem Dorf (inklusive Unterstützungen (extra anzeigen))

Auf Webservice umbauen. Unterteile werden sein: to XML, Web Service Calls, Objekte werden sehr wahrscheinlich überarbeitet werden müssen. Mit Web-Service-Person absprechen. Wichtig ist, dass sich der Client an den Webservice orientiert und nicht umgekehrt.

##### Server (Engine)

Event-Auflösungs-Aufrufe der Datenbank von abgelaufenen Events auslösen.

##### Datenbank

Event-Auflösungs-Aufrufe!

Boards brauchen eine Zugehörigkeit. Berechtigung Sachen.

Das bedeutet: Benutzer brauchen Eigenschaften wie z.B. Admin.

Gut wäre auch ein Clan Leader / Clan Gründer. Dadurch können wir extra werte wie z.B. Beschreibung, Wappen etc. in den Clan einbauen.

Dafür müssen die Clan beitreten sachen bis zum Client runter funktionieren. Danach bauen wir das erst ein. (Dann sehen wir nämlich wo es am vernünftigsten ist. Es kommt eigentlich sicher in den WebService.)

(Überlegung) Zu board eingeladene Nutzer!

##### Admin-GUI

Events einbauen. Mehr automatisierte Knöpfe (aka Prozeduren-Buttons)

##### Webservice

Bauen.

GET als erstes machen. Get alle Dörfer: Hier darf der Nutzer nur die Informationen bekommen, die er auch haben darf. (also nicht alles, sondern nur Besitzer, Ort, und so) auch sollen diese Objekte nicht dieselben sein wie in der DB. Stattdessen die benötigten (oder möglicherweise benötigten) Daten. Der Clan soll beispielsweise bei einem Dorf mitgegeben werden.

Boards sollen nur den Clan Mitgliedern zugänglich machbar sein. So im Sinne von: Mein Benutzer hat 10 sichtbare Boards: 4 von seinem Clan, 3 Öffentliche, 3 von Entwicklern (Update discuss oder so zeug.)

Session wird hier gemerkt werden. (Siehe beispiel Programm SessionHandler). Die Nutzer die er da hat darf er dann mit dem bedienen, was sie dürfen.

Der WS muss auch alle Befehle (DB-Prozeduren und mehr) maskieren bzw. selbst anbieten:

* Clan beitreten
* Clan verlassen
* Truppen schicken
* Board erstellen
* Gebäude ausbauen
* Geschwindigkeit des Ressourcenupdating meines Dorfes. (Praktisch für Client zum mitrechen wie Viele Ressourcen er hat, Entlastung der Leitung, weil man nicht ständig fragen muss wie viel man hat, hilfreich, wenn Ressource update timer langsam)

### Aufteilung

##### Notsch

* Engine
  + Event resolve
  + DB komm
* Admin-Tool
  + Events & Prozeduren

##### Millonig

* DB
  + Event auflösungen
  + Reformen wie oben genannt (Clan etc) Rechte etc
    - Clan beschr. Hinzufügen etc.
    - User Clan rechtelevel (ADMIN, owner, leader, member)!
      * In allen Prozeduren abprüfen

##### Jäger H.

* Webservice
  + Front end
    - WS für Client anbieten
  + Back end
    - Zugriff auf DB
  + Userhandling
    - Ressourcen Verwaltung am Client
* Client C#
  + Forum GUI
    - Boards
    - Kommentare
  + Überarbeitung der Karten-GUI

#### Zeiringer

* Client C#
  + DB-Connect Klasse zu WS-Connect Klasse umbauen
  + Events
  + Deko
  + Grafik
  + Remember password
  + Clan GUI
  + Generell GUI-Design überarbeiten