# Besprechung der Aktivitäten der letzten zwei Tage

Probleme mit Merging-Konflikten

Es ist ein Problem durch das Umbenennen der Packages entstanden, welches sich jetzt gelöst hat.

## Stefan

– Großbuchstaben aus allen möglichen Packpage-Namen entfernt und die Struktur, des Projekts überarbeitet, insbesondere bei Backend und Cache. Fehlendes-Gui-Package ergänzt.

– Problem: [WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!

Abbildung 1: Die Richtige Auswahl in der Textdarstelllung bei Eclipse.

Die Lösung des Problems liegt in der Umstellung der Textcodierung auf UTF-8 in Window/Preferences/General/Workspace  
Dort gibt es im unteren Bereich eine entsprechende Auswahl

Skript zum im-Auge-behalten des Problems liegt im Resourcen-Ordner.

Alle Interfaces und einige implementierende Klassen um noch fehlende Methoden ergänzt.

## David

Test für das Model geschrieben. Weitere Implementierungen eingefügt. Yaml Parser implementiert. Yaml File zum Einlesen geschrieben.

Es gibt noch Unklarheiten bei der Umsetzung des geänderten „Erbauer“-Entwurfs.

## Jan

Main Klasse implementiert als Stub. fehlende Klassen im Backend/Cache angelegt.

## Leon

GUI von MS2 abstrakt implementiert. Die von MS1 werden noch von Leon implementiert. Yaml Files werden von Leon, Linus und David überarbeitet.

## Linus

Yaml Parser implementiert. CSV service implementiert. Yaml service kann Yaml Files einlesen. Png service implementiert. Überflüssige Interfaces gelöscht.

[Interface Lösch-Diskussion]

# Fragen

## Mockito: Kommt Mockito nur bei den Tests zum Einsatz?

Nein. Mockito kann in der Implementierung verwendet werden um gleich MockUps von Klassen zu erstellen. Dieser Teil findet dann in src/main/java statt.

## Yatta: Was ist alles in einem Profil enthalten?

Die Eclipse-Plugins.

## Macht es Sinn Ordner in die Java-Struktur einzufügen?

Nein. Diese werden als Package behandelt.

## Warum sind so viele unbenutzte Imports in manchen Klassen vorhanden?

David hatte am Anfang alle Gui-Klassen in einem einzigen Package und beim Aufsplitten des Packages per Refactoring sind die Imports vermutlich „automatisch“ entstanden, obwohl sie gar nicht benötigt werden.