MVVM

# Forklaring

Forkortelse af Model – View – ViewModel. Dette designmønster (og framework) bruges til at øge testability af GUI-tunge applikationer:

*”MVVM was designed to make use of data binding functions in WPF to better facilitate the separation of view layer development from the rest of the pattern, by removing virtually all GUI code (code-behind) from the view layer.”*

Et eksempel kan findes [her](http://www.markwithall.com/programming/2013/03/01/worlds-simplest-csharp-wpf-mvvm-example.html)

## Model

Business-logic, reference til data access layer.

## View

User-interfacet

## ViewModel

?

## Binder

XAML.

# Fordele

* Udviklet specifikt til WPF af Microsoft
* Øget testability
* Skiller UX-arbejde fra backend arbejde, altså splitter GUI-designeren fra software-udvikleren.

**Et meget godt argument**, jeg synes der især er relevant for vores projekt faktisk:  
*”Even when a single developer works on the entire code base, a proper separation of the view from the model is more productive as the user interface typically changes frequently and late in the development cycle based on end-user feedback.”*

# Ulemper

* Lidt svært at forstå når man ikke har nogen praktisk erfaring med det
* Føles kompliceret i forhold til den simple 3-lags-model jeg personligt er vant til.
* Meget refaktoreringsarbejde ift. vi skal aflevere om en måneds tid.

# The verdict

Jeg synes vi skal skifte til det. Det bliver besværligt at teste vores GUI uden MVVM, og det er en god idé at skifte inden vi laver nye usercontrols.

Godt nok er der en del refaktoreringsarbejde, men det er vi jo blevet gode til efterhånden ;)