

# **MVVM: Model-View-ViewModel**

**Advanced Softwareengineering - DHBW Stuttgart**

Nico Vogel, Lukas Sopora

31.12.2019

## **Inhaltsverzeichnis**

<b>1</b>	<b>Was ist das Problem das MVVM angeht?</b>	<b>2</b>
<b>2</b>	<b>Beschreibung der Komponenten von MVVM</b>	<b>3</b>
<b>3</b>	<b>Beispiel des MVVM Patterns in C# WPF</b>	<b>4</b>
<b>4</b>	<b>Vergleich zu dem MVC Pattern</b>	<b>5</b>
<b>5</b>	<b>Vergleich zu dem MVP Pattern</b>	<b>6</b>
<b>6</b>	<b>Kritische betrachtung des MVVM Patterns</b>	<b>7</b>

# 1 Was ist das Problem das MVVM angeht?

- ui fix coupled mit logik
- schwierig einzelne komponenten oder ganze ui auszutauschen
- cross platform nicht möglich
- wiederverwendung der logik nicht möglich

## 2 Beschreibung der Komponenten von MVVM

- Komponenten
  - Model
    - \* Entität
    - \* enthält nur daten oder logik für sich selbst
  - ViewModel
    - \* business logik
    - \* verwendet beliebig viele models
    - \* stellt werte bereit für die ui
    - \* läßt werte ein aus der ui
    - \* leitet events weiter an die logik
    - \* kennt Model
  - View
    - \* nur design
    - \* eigenständig
    - \* kennt ViewModel
  - Übersicht komponenten
  - Übersicht einteilung in das Application Layered model
- Kommunikation
  - OneWay Binding: von View zu ViewModel oder von ViewModel zu View
  - TwoWay Binding: Von der View zum ViewModel und gleichzeitig auch andersrum
  - Events: die view reicht ein event an das ViewModel weiter

### **3 Beispiel des MVVM Patterns in C# WPF**

## **4 Vergleich zu dem MVC Pattern**

## **5 Vergleich zu dem MVP Pattern**

## **6 Kritische betrachtung des MVVM Patterns**