# Veit Meeting wg. VDMA 04.05.2015:

* Frequenzumrichter anklickbar ohne absturz
* Vom Veit wird 90% Zuverlässigkeit erwartet
* Gui auf 4K (entweder Auflösung runter oder Schriftgröße etc. Hoch)
* Abstecken -> erneuter Scan -> Gerät nicht mehr da ohne Programm zu schließen (vllt mit „Process-Kill VS weg)
* Eigenschaten die als Geräte erkannt werden evtl. durch hart gecodete Liste rausschmeissen
* WPF mit Blend oder auch nicht
* ModelView Controler –Methodik für GUI Programmierung mittelfristig anschauen
* Logiccode nicht in xaml.cs schreiben! bei MVVM seperate Ordner mit View, View-Model und Model (BL)
* In xaml DataBindings nutzen um GUI reagieren zu lassen
* Xaml.cs : DataContext = Vmodel (Verweis auf Datei um Variablen für DataBindings nutzen zu können)
* Bindings müssen public Propertys sein
  + Public int Zahl = 2 (FALSCH: Feld!)
  + Public int Zahl {get;set;}
* Listen in ObservableColletions um dynamisch Databindings vorzunehmen und so evtl. ButtonListen zuerzeugen
* Verschiedene Devices wie Frequenzumrichter und IOLink über INTERFACES auf GUI ansprechen
* Geräte-Klassen:
  + 1. Primitive Geräte (detektieren über A/D-Karte versuchen) (GUI abfrage nach Primitiven Geräten)
  + 2. IO-Link
  + 3. Feld-Bus (Frequenzumrichter)
  + (Ethernet)
  + (Next-Generation)
* Allgemeinste Stufe unserer Software: Suche nach Comm.Geräten über Schnittstelle
* Schnittstellenaufruf eines bestimmten Gerätes -> spezifischer SPS-Treiber wird geladen und sucht nach Geräten
* Ab Nutzung der Beckhoff-API ist die Software ein Universeller Beckhoffspezifischer Treiber (universell wg. Universeller Geräteerkennung)