**Ziel 1 „MVC Architektur Überarbeitung“**

**Nach inzwischen 3 Jahren erfolgreichen Einsatzes der neuen Web Technologie „MVC“ benötigen wir eine Überarbeitung der MVC Architektur.**

**Hintergrund ist, dass unsere MVC Architektur stets anspruchsvollsten Kundenanforderungen z. B. nach flexiblen Web Designs in unterschiedlichster Optik + HTML Seitenaufbau gerecht werden muss.  
Hierbei ist die größte Herausforderung, alle neuen Web Portale mit ihren unterschiedlichen Designs stets auf nur einer einzigen Code Basis bereitstellen zu können.  
Denn nur mit einer einzigen Code Basis sind geringste Wartungsaufwände sowie effiziente Funktionserweiterungen mit maximaler Produktivität überhaupt möglich.  
  
Zielanforderung im Detail:  
  
Die neuen MVC Web Design Themes erweitern auf die Funktionen „Hierarchie + Vererbung“**Beispiel, konkreter Bedarfsfall aus der Praxis:   
Neben dem Standard Metronic Design Theme des DAD setzen wir seit Kurzem das neue „Kroschke On“ Design Theme ein.  
Das „On“ Theme soll nun die Basis für das Erscheinungsbild künftiger Web Portale stellen, wie z. B. das „Kroschke COSTAR“ Portal.  
Hierzu benötigen wir zwingend die Möglichkeit einer hierarchischen Ableitung aus bestehenden Themes, also quasi ein Feature, bestehende Funktionen + Design-Features vom Basis Theme (Kroschke On) zu erben und nur die Abweichungen im Ziel Portal (Co Star) zu „überschreiben“ bzw. neu zu implementieren.

Durch die Bereitstellung dieser hierarchischen Vererbungsarchitektur können wir künftig die Design Basis für neue Web-Portale frei wählen und nur die nötigen Abweichungen bezüglich Optik und Seitenaufbau mit geringstem Aufwand sowie ohne Code Redundanz implementieren.  
  
  
**Geschätzter Aufwand für dieses Ziel:**8 Manntage (64 h)  
  
  
**Vorgeschlagene Zielgewichtung:**50%

**Ziel 2 „Pflege MVC Komponenten + HTML Templates“**

**Nach mehreren Jahren erfolgreichen Einsatzes der neuen Web Technologie „MVC“ benötigen wir eine Überarbeitung einiger MVC Komponenten + HTML Templates.**

**Hintergrund ist, dass die Architektur historisch schnell gewachsen ist, dabei trotzdem den ständig neuen Web Anforderungen gerecht werden muss(te), im historischen Verlauf aber nicht immer Zeit + Budget vorhanden waren (und sind), alle neuen Komponenten so sauber in die Architektur zu integrieren, dass ein maximal schlanker Programmier Code – verbunden mit geringstem Einarbeitungsaufwand insbesondere für neue / externe Entwickler – erzielt werden kann.  
  
Zielanforderung im Detail:  
  
Kapselung bestimmter Web Komponenten und Bausteine (z. B. Multi-Select, etc)**Jedes Mal wenn die betroffenen Komponenten zum Einsatz kommen, entsteht unnötig redundanter HTML + CSS Code (partiell auch Script Logik), der die Wartung bestehender Einsatzfälle sowie den erneuten Einsatz dieser Komponenten unnötig verkompliziert.  
Ziel ist es, für die betroffenen Komponenten wiederverwendbare Programmierbausteine zu entwickeln, die idealerweise künftig mit nur einer einzigen Codezeile zum Einsatz kämen.  
 **Vereinfachung unserer hauseigenen Grid Komponente**Die Grid Komponente ist nicht nur der definitiv zentrale + mächtigste Baustein innerhalb unserer Web Komponentensammlung, er kommt auch am häufigsten zum Einsatz, z. B. in unseren Standard Web-Reports und in unseren Assistenten gesteuerten Anwendungen („wizard driven applications“).  
Der ursprüngliche Quellcode unserer Grid-Komponente ist Open Source und wurde ab dem Stand 2013 von uns in eigener Regie weiter entwickelt.

Die Anforderungen seitens unserer Kunden an unsere Grid Komponente sind dabei immens.  
Fähigkeiten wie kundenbezogene Spaltennamen + Lokalisierungen, ein Wysiwyg Übersetzungs-Feature für (Kunden-)Administratoren, sowie eine maximal verzögerungsfreie Darstellung, schnelle serverseitige Filterung und auch das Caching von großen Datenmengen - auch bei langsamen Internetverbindungen - sind nur eine Auswahl der inzwischen verfügbaren Features.

Die Features „Caching“ + „Filterung“ bringen allein im .NET Backend eine Ansammlung von ca. 2 Bildschirmseiten Code pro Grid Einsatz mit sich.  
Auch auf den Front End Seiten haben wir es unnötiger Weise mit einer gewissen Anzahl wiederkehrender Script Zeilen zu tun.  
  
Hier besteht nun der dringende Handlungsbedarf, den betroffenen Code in einem Standardbaustein zu kapseln.   
Denn der Code ist war nicht per se redundant, seine Lesbarkeit und somit seine Wartbarkeit leiden aber darunter in nicht unerheblichen Maße, dass es insbesondere neuen oder externen Entwickler die Einarbeitung stark erschwert.  
  
  
**Geschätzter Aufwand für dieses Ziel:**7 Manntage (56 h)  
  
  
**Vorgeschlagene Zielgewichtung:**50%