

Fernstudium Medizinische Informatik an der Beuth Hochschule Berlin  
3. Semester (WS 2009/2010)  
Dr. Felix v. Festenberg  
felix@festenberg.de

### Versionshistorie

Datum	Version	
14.03.2010	0.2.3	<ul style="list-style-type: none"><li>• GoogleMaps Komponente mit looser Kopplung fertig gestellt</li><li>• Komplette Login-Steuerung mit DB-Abgleich fertig</li><li>• Komplette Navigation zu allen geplanten Seiten fertig</li><li>• Stammdaten 9 von 12 per Datagrid im Adminmodus pflegbar</li></ul>
13.01.2010	0.1.3	<ul style="list-style-type: none"><li>• Releaseplanung und -strategie ergänzt</li><li>• Login, tabs im main und googleMaps in Anwendung eingebunden</li><li>• Autom. Source.html entdeckt und eingebaut</li></ul>
10.01.2010	0.0.3	<ul style="list-style-type: none"><li>• Hilfeanwendung technisch fertig (txt, flv, swf)</li><li>• Aufstellung über Features in fertig, Ausbau und fehlt</li><li>• Doku in vorliegende Form gebracht</li></ul>
08.01.2010	0.0.2	<ul style="list-style-type: none"><li>• Keine Änderung der Doku</li><li>• Generelles Layout und Style festgelegt</li></ul>
06.01.2010	0.0.1	<ul style="list-style-type: none"><li>• Doku angelegt,</li><li>• Inhalte zu Usecase</li><li>• Design</li></ul>

## Inhaltsverzeichnis

1	Vorüberlegungen .....	3
2	Textuelle Beschreibung (Usecase).....	3
2.1	Organisation .....	4
2.2	Fortbildungen.....	4
2.3	Projekte.....	5
3	Analyse des Projektfortschritts .....	7
3.1	Fertig und Funktionsfähig.....	7
3.1.1	Projektstruktur und –versionierung .....	7
3.1.2	Neuerungen vom 13.3.2010 .....	10
3.2	Ausbaufähig.....	14
3.2.1	Style und Layout .....	14
3.2.2	Content .....	14
3.3	Was noch nicht fertig ist oder nicht richtig klappt.....	14
3.3.1	Layout Main .....	14
3.3.2	DB .....	14
3.3.3	XML .....	15
3.3.4	Datagrid und Chart .....	15
4	Releaseplanung und -versionierung.....	16
4.1	0xx.....	16
4.2	1xx.....	16
4.3	2xx.....	16
4.4	3xx.....	17
4.5	Visionen .....	17

## **1 Vorüberlegungen**

Im eigenen Arbeitsumfeld (Abteilungsleitung zentrale IT-Klinische Systeme mit 20 Mitarbeitern in einem kommunalen Krankenhauskonzern mit 5 Krankenhäuser davon 4 Akutversorger mit insgesamt 3500 Betten) stellt sich regelmäßig die Frage, wie viele Ressourcen für ein bestimmtes Projekt verbraucht wurden und wie viel tatsächliche Kosten und Zeitaufwände angefallen sind. Aussagen über freie Ressourcen für zusätzliche Projekte können nur grob geschätzt werden.

Es gibt auch keine einfach zu bedienende und leicht anpassbare Übersicht über die Personalkosten und die Wochenstunden je Mitarbeiter außer SAP HR, was nicht gerade benutzerfreundlich ist und nur durch die zuständige Modulbetreuung angepasst werden darf. Zu den besuchten Fortbildungen der Mitarbeiter gibt es gar keine elektronisch erfassten Daten geschweige denn irgendwelche Auswertungen.

Daraus resultieren 3 Bereiche der Anwendung, die miteinander verknüpft sind:

- Organisation
- Projekte
- Fortbildungen

Gängige Projekt- und Personalmanagementsoftware ist entweder nicht ausreichend, zu komplex oder zu teuer.

Im Rahmen dieser Projektarbeit soll versucht werden eine möglichst intuitiv zu bedienende Web-Anwendung zu erstellen mit einem erweiterbaren Datenmodell, dass anfangs nur die wirklich benötigten Daten enthält. Da die Qualität der Auswertungen von der Vollständigkeit und Korrektheit der durch die Mitarbeiter eingegebenen Daten abhängt, soll die Anwendung in 3 (ggf. 4) Ausbaustufen erfolgen mit zwischengeschalteten Anwendertests und -feedback. Diese Gliederung findet sich in den folgenden Abschnitten wieder:

- Minimal
- Basis
- Produktiv
- Erweiterung

Für die Projektarbeit wird angestrebt zur Halbzeitabgabe eine funktionsfähige Minimalvariante zu haben, bei der die Anbindung an die Datenbank funktioniert

Wichtig ist ein gutes funktionelles Layout, das optische Layout soll eher schlicht gehalten werden.

## **2 Textuelle Beschreibung (Usecase)**

In der IT-(Unter)Abteilung „klinische Systeme“ eines kommunalen Krankenhauskonzerns, die aus 3 Teams (ishmed, ish und Subsysteme) und insgesamt 20 Mitarbeitern besteht, soll die interne Organisation(a), Fortbildungen(b) und Projekte(c) dokumentiert und ausgewertet werden. Es geht vor allem darum Zeiten und Kosten zu dokumentieren, um einen schnellen Überblick über die tatsächlichen zeitlichen und finanziellen Aufwände von Projekten im

Vergleich zur Planung zu gewinnen. Es sollen Aussagen zu freien personellen Ressourcen bei der Annahme bzw. Ablehnung von neuen Projekten/Aufgaben gemacht werden können.

## **2.1 Organisation**

Mitarbeiter haben als Stammdaten verpflichtend Name, Vorname, email, Einstellungsdatum und Kündigungsdatum, optional kann ein Titel, ein Namenszusatz, eine private Telefonnummer für Notfälle und eine geschäftliche Handynummer hinterlegt werden. Jeder Mitarbeiter sitzt an einem bestimmten Arbeitsplatz, der im Zeitverlauf wechseln kann. Für den einzelnen Arbeitsplatz ist immer eine Beschreibung und die postalische Anschrift (Strasse, PLZ, Ort) hinterlegt, meistens gibt es auch je eine Telefon- und Faxnummer.

Zu jeder Abteilung gibt es einen Kurznamen und eine Beschreibung sowie eine Leitung, die auch gleichzeitig ein Mitarbeiter ist, meistens auch eine Stellvertretung. Jede Abteilung hat genau eine übergeordnete Abteilung (die oberste Stufe hat sich selbst als übergeordnete Abteilung).

Jeder Mitarbeiter kann auch für mehrere Abteilungen gleichzeitig arbeiten und im zeitlichen Verlauf wechseln. Die jeweiligen Wochenstunden und Angaben zum Gehalt (TvÖD-Tabelle ab Stufe 9) sollen hinterlegt sein (Stundensatz immer, da es auch externe Mitarbeiter auf Stundenbasis gibt, optional das Monats- und Jahresgehalt). Kein Mitarbeiter darf der gleichen Abteilung zu jedem beliebigen Stichtag mehr als einmal zugeordnet werden.

Es soll ein Organigramm (ggf. als XML-Tree), eine Telefonliste und eine Notfalltelefonliste angezeigt werden können. In Abhängigkeit von der Rolle des Users (Mitarbeiter, Leitung) sollen die Daten eines einzelnen Mitarbeiter inklusive zeitlichem Verlauf der Abteilungszugehörigkeit angezeigt bzw. geändert werden können.

In einer Übersicht soll ersichtlich sein, wie viele Manntage/Stunden pro Abteilung in einem definierten Zeitraum zur Verfügung stehen und wie hoch dafür die Kosten sind.

## **2.2 Fortbildungen**

Es gibt eine Liste der möglichen Fortbildungen, die immer einen Titel haben, an einem bestimmten Ort stattfinden, zu einem bestimmten Zeitpunkt beginnen, einen Ansprechpartner mit Kontaktdaten und einen Preis haben. Es muss unterschieden werden können, ob dieser pro Person oder für die ganze Veranstaltung unabhängig von der Teilnehmerzahl gilt. Es soll auch ein Kennzeichen geben, ob es eine interne oder externe Fortbildung ist. Zu einigen Fortbildung gibt es auch eine Beschreibung und einen Link für weitere Informationen.

In einer zweiten Liste stehen die Mitarbeiter und die Anzahl der Stunden, die sie an einer Fortbildung teilnehmen. Es wird unterschieden, ob die Fortbildung geplant ist oder schon stattgefunden hat. Teilweise ist der Besuch einer Fortbildung mit einem Projekt oder einer Abteilung verknüpft. Kein Mitarbeiter darf an derselben Veranstaltung zweimal angemeldet sein.

Es soll eine Liste mit allen eingetragenen Fortbildungen angezeigt werden. Bei Klick auf eine soll die Adresse eine GoogleMap Inlay visualisiert werden sowie eine hinterlegte Textdatei zu den Fortbildungsinhalten geladen und angezeigt werden. In einer weiteren Funktion kann sich ein Mitarbeiter für eine Fortbildung eintragen (idealerweise kann die Anwendung nur mit

personalisiertem Login aufgerufen werden und der angemeldete User wird direkt unveränderbar vorgeschlagen, die Leitung kann jeden User auswählen). Es soll einen eigenen Punkt für die Neuanlage einer Fortbildung geben.

Als Auswertung soll eine Liste mit allen Mitarbeitern ausgegeben werden, die die kumulierten Fortbildungskosten und Dauern über einen vorher einzugebenden Zeitraum ausgibt. Ebenso sollen alle Fortbildungen eines einzelnen Mitarbeiters ausgegeben werden).

### **2.3 Projekte**

Projekte werden in Aufgaben (als kleinste Einheit) unterteilt und können beliebig unter anderen Aufgaben zusammengefasst/gegliedert werden. Die oberste Gliederungsebene hat sich selbst als übergeordnetes Projekt/Aufgabe. Jede Aufgabe hat einen Titel und eine Beschreibung, oft gibt es auch einen Link zur Dokumentation. Für die zeitliche Dokumentation soll immer ein Startdatum, ein geplantes Endedatum (optional das tatsächliche Endedatum) sowie die geplanten internen und externen Stunden und Kosten (optional) erfasst werden. Jede Aufgabe hat einen einheitlichen Status mit den Festwerten „geplant“, „im Zeitplan“, „überfällig“, „abgeschlossen“, „abgebrochen“ und „zurückgestellt“. Jeder Teilaufgabe können beliebig viele Mitarbeiter zugeordnet werden, die jeweils zu einer Abteilung gehören und eine der folgenden Aufgaben haben: „Mitarbeit“, „Leitung“, „Kontrolle“, die ggf. noch erweitert werden.

Des Weiteren soll dokumentiert werden, wie viele Stunden an einem Tag ein in einem Projekt zugeordneter Mitarbeiter dafür verwendet hat, ggf. soll dazu eine Bemerkung erfasst werden können. Es darf pro Tag, Projekt und Mitarbeiter immer nur ein Eintrag existieren.

Analog zu den Fortbildungen soll es eine Übersicht und Detailansicht mit den Funktionen anzeigen/ändern/neu geben. Zusätzlich kommt hier die Zuordnung von Personen hinzu. Eine einfache Übersicht soll dem angemeldeten Benutzer eine Auswahl der ihm zugeordneten Projekte anzeigen, danach eine Tabelle mit den bereits erfassten Zeiten und eine Funktion zur Erfassung neuer Einträge mit vorgeblendetem Tagesdatum.

Als Auswertung soll eine Übersicht aller Projekte mit den bereits verbrauchten Personalressourcen angezeigt werden, eine Kontrollfunktion, ob die eingetragenen Stunden pro Mitarbeiter den wöchentlichen Arbeitsstunden in etwa entsprechen (Urlaub, Krankheit und Feiertage sind wahrscheinlich ein schwer lösbares Problem), ggf. kommen zu einem späteren Zeitpunkt weitere Auswertungen hinzu, die sich aber erst ergeben, wenn genügend Daten eingetragen wurden und die Applikation als Steuerungshilfe von allen Mitarbeitern akzeptiert wurde.



### 3 Analyse des Projektfortschritts

#### 3.1 Fertig und Funktionsfähig

##### 3.1.1 Projektstruktur und –versionierung

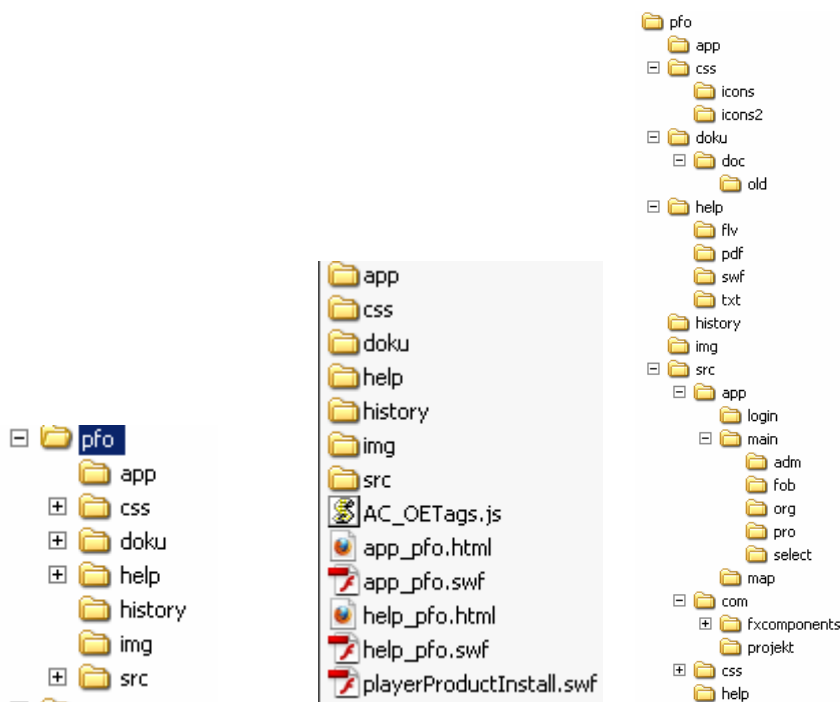
Es wurde eine einheitliche Verzeichnisstruktur und Dateinamenskonvention festgelegt und umgesetzt.

Das Projekt hat das Kürzel pfo(S) (Projekte, Fortbildung, Organisation, das grosse s für Stammdaten im AdminModus sieht einfach nett in Klammern aus und hat keinen tieferen inhaltlichen Sinn).

Struktur der Hauptanwendung mit Komponenten für jede Seite (Unterseiten auf den Registern sind noch nicht als eigene Verzeichnisse angelegt, da noch unklar ist, ob eine solche Granularität überhaupt Sinn macht, gleiches gilt für die die ActionScript-Anbindung. Es wurde bisher konsequent versucht nur mit Includes und ohne Klassen zu arbeiten, es wird sich zeigen, ob die Entscheidung richtig war.

Insgesamt wird immer zu einem definierten Zeitpunkt eine Kopie des Arbeitsordners (pfo) gemacht und eine fortlaufende Nummer vergeben werden. Diese werden dann in den Webespace [www.projekt.festenberg.de](http://www.projekt.festenberg.de) transportiert und auf einer Startseite ist eine Tabelle mit jeweils einem Link zur aktuellsten Version der Anwendung und der der Hilfe, sowie zu dieser Dokumentation und sämtlichen.

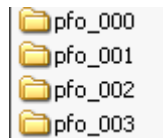
(genaueres zur Versionierung siehe Kapitel 4 Releaseplanung und –versionierung)



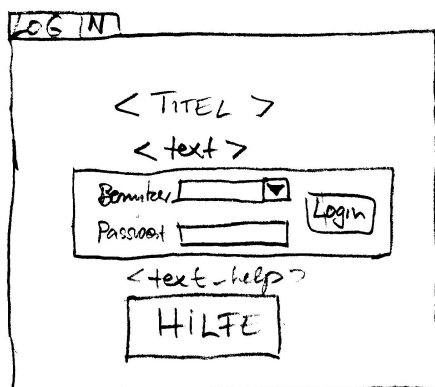
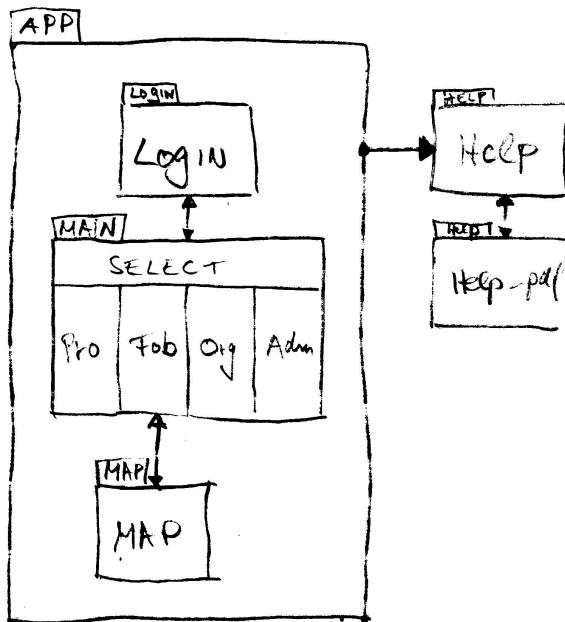
(history, AC\_OETags und player... sind von Flex generiert, um die Browser Vor/Zurück-Steuerung anzubinden)

Ansonsten ergibt sich der Aufbau der Verzeichnisstruktur völlig zwanglos und leicht nachvollziehbar, da durchgängig einheitlich umgesetzt aus dem Layout der Anwendungen und bedarf keiner weiteren Erläuterungen, da er sich bis jetzt als praxistauglich erwiesen hat.

Es werden je nach gemachter Fortschritte komplett lauffähige Versionen als Kopie des gesamten Ordners pfo als pfo\_<fortlaufende 3-stellige Zahl> abgelegt und upgeloadet auf usb, um in der lokalen Umgebung mit eigenem Apache und MySQL zu testen, um danach auf den Webserver (projekt.festenberg.de) abgelegt zu werden.



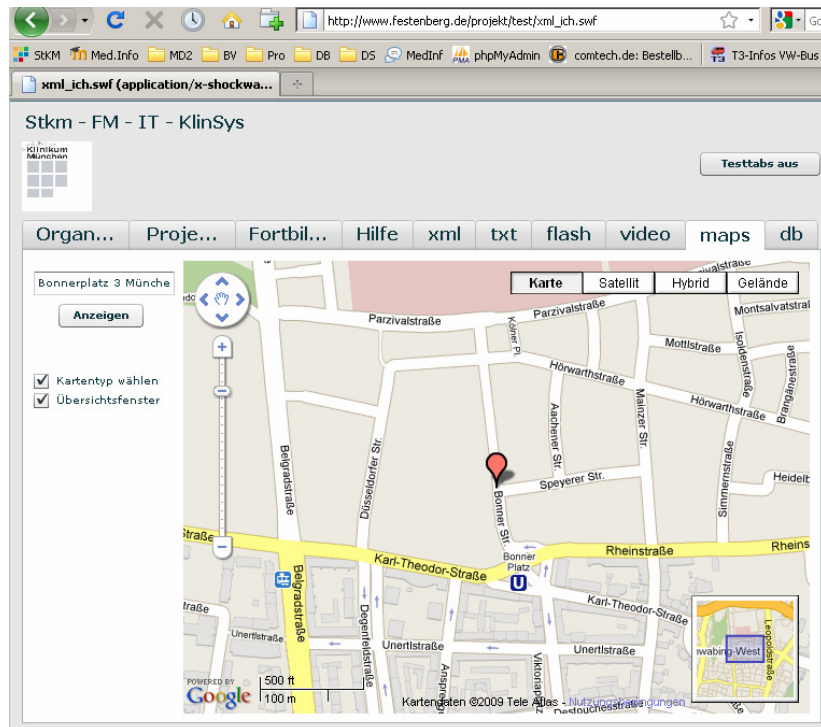
Folgender Layout-Entwurf, der in vielen Iterationsstufen auf dem Papier erstellt wurde, liegt der gesamten Anwendung zugrunde (so auch die Verzeichnisstruktur)





Damit sind von den 10 geforderten Punkten, die Punkte 1-4 (Stack, state, txt, flv) erfüllt. Die Anzeige der Videos wurde mit der Freeware-Komponente FxVideo erzeugt, die flash Animationen wurden mit der Freeware Wink erstellt und bearbeitet. Für die Videos habe ich trotz intensiver Recherche noch kein zu mir passendes Tool gefunden.

Die GoogleMaps-Integration ist in der Testumgebung fertig und als Komponente aufgebaut, die in die Fortbildungsseite bei Klick auf bestimmten Ort einer Fortbildung erscheinen soll, hier ist sämtliche Technik (wie die swc-Integration, komplett selbst gemacht. Es ist also noch die Anforderung 9a (Geotagging) und 8 (Komponente) erfüllt



Sprung auf 0.1.3 ist vollbracht: (13.1.2010 2:00 )

In der Anwendung sind die 3 Rahmenseiten inklusive Verlinkung erstellt. Login und Map ist vorläufig fertig. Kommen noch ein paar Verfeinerungen, die jetzt aber erstmal nicht wichtig sind. Der automatische Aufruf vom richtigen Kartenausschnitt funktioniert noch nicht, da das bei looser Kopplung (wie hier) irgendwie anders gehen muss.

Ebenfalls entdeckt habe ich den automatische Codeexport nebst html-Aufbereitung und download-Paket (sogar als rechte Mausfeature in der Anwendung selbst) und natürlich für diese Version in die Projektübersichtsseite eingebaut.

### 3.1.2 GoogleMaps Komponente (Neuerungen vom 13.3.2010)

Die GoogleMaps Komponente mit looser Kopplung wurde verfeinert und tatsächlich loose gekoppelt gemacht mit dem der Übergabe der wichtigen Parameter und einer externen Aufrufmethode. Es wurde auch für die Übersichtlichkeit und bessere Wiederverwendbarkeit MXML und Actionscript in einer einzigen Datei vereint und diese in das neue Unterverzeichnis

geotag überführt (der Verzeichnisname „map“ hat Schwierigkeiten gemacht wegen gleichnamiger ID)

Kurzbeschreibung (aus Quelltext)

Diese loose koppelbare Komponente liefert eine komplette Einbindung von GoogleMaps.

Attribute(properties)

/key GoogleMapsKey (beantragen für jede URL unter [code.google.com/intl/de-DE/apis/maps/signup.html](http://code.google.com/intl/de-DE/apis/maps/signup.html))

/ort String für den ersten Aufruf (wird auf der Karte angezeigt)

Methoden:

showInMap(String):void zeigt übergebenen String an

ACHTUNG:

Wenn die Komponente initial nicht angezeigt wird, kann es zu Fehler kommen.

Workaround: creationPolicy="all" im Viewstack, Tabnavigator etc. setzen

### 3.1.3 Anbindung MySQL-DB

Die unter 2. beschriebene Datenbank wurde implementiert, mit Inhalten gefüllt und auf [projekt.festenberg.de](http://projekt.festenberg.de) importiert.

Es wurde eine php-Datei erstellt (pfo\_select.php), über die mit GET-Parametern eine SQL-Abfrage als String zurückgibt (Felder mit „ „ getrennt, Zeilen mit „;“). Diese kann wahlweise auch die Spaltennamen zurückgeben.

Der Zugriff auf die Datenbank wurde in ein include (mysqlconn.php) ausgelagert, die die tatsächlichen Zugangsdaten enthält.

GET-Parameter:

- tab: Tabellename, mehrere mit Komma getrennt ohne Leerzeichen (keine Angabe → Rückgabe „NULL“)
- feld: zurückgegebene Spalten in der angegebenen Reihenfolge mit Komma getrennt, keine Leerzeichen (default ist \*)
- where: komplette Bedingungen
- oder: Spaltenname für Sortierreihenfolge
- option:
  - ohne Angabe → Spaltennamen als 1. Datensatz, dann Daten
  - =only → nur Spaltennamen
  - =off → nur Daten



TO BE CONTINUED HERE (hopefully as a success-story 😊)

## 3.2 Ausbaufähig

### 3.2.1 Style und Layout

Das grundsätzliche Layout ist komplett selbst erstellt, allerdings angeregt durch lange Touren im www. Grundsätzlich sollen später noch die Umstellung des Styles komplett in eine css erfolgen und alle Elementpositionen ausschließlich relativ zueinander mit mindest Pixel erfolgen. Dies ist im Rahmen des Projektes wahrscheinlich zu aufwändig und die Zeit sollte primär für die Funktionalität und Benutzerfreundlichkeit der Anwendung verwendet werden.

Als Minimalvariante könnte eine Subanwendung (evtl als lockere Kopplung) erstellt werden mit der die Farben für Außenrahmen/buttonDown, Kopf-/Fussleiste/RegisterFarbe, Hintergrund, Schrift Kopf/Fuss/Register, Schrift Inhalt, Rahmenfarbe, Steuerelement farbe für Hintergrund und Schrift,,... die Liste könnte beliebig verlängert werden, am pragmatischsten ist es wahrscheinlich ein Relativschema mit nur 2-3 Eingabefarben anzulegen.

### 3.2.2 Content

Es fehlt:

- Inhalt-DB  
(am 1.3.2010 eingerichtet und auf [projekt.festenberg.de](http://projekt.festenberg.de) verfügbar gemacht)
- Inhalt Hilfe-Video
- Inhalt Hilfe-Flash
- Inhalt Hilfe-Text
- Inhalt Hilfe-pdf

Der DB-Inhalt ist eher ein Muss und steht im Kontext mit den Umsetzungen der DB-Anbindung in den zweiten 50% des Projektes.

Für alle anderen Anbindungen (Textfile, Video mit progressive Load, Flash) wurden jeweils 4 Demodateien angelegt, um technisch zu zeigen, das es geht. Es wäre wünschenswert, wenn bis Projektende für jede Datei sinnvolle Inhalt erstellt werde. Besser kurz, aber demonstrativ, was die Technik leisten kann. Als Projekteinzekämpfer muss ich hier eventuell Abstriche machen.

## 3.3 Was noch nicht fertig ist oder nicht richtig klappt

### 3.3.1 Layout Main

Hier gibt es bisher nur das RahmenLayout und für das Inhalte-Tabregister gibt es nur folgende handschriftliche Aufzeichnungen, die evtl. nicht mehr bis Montag abend bis ins kleinste Detail funktionsfähig vorliegen. Sind aber mit der DB-Anbindung gekoppelt und der allernächste Schritt.

### 3.3.2 DB

Zuerst wurde mit tomcat experimentiert und das lief lokal vom usb nur schlecht als recht. Auf meinem eigenen Webspace läuft ausserdem kein tomcat sondern nur php. Die bis dato in die flex-hibernate-tomcat-apache gesteckte Zeit war bis auf den Erkenntnisgewinn und

Erfahrungen, die ich vielleicht später mal gebrauchen kann, nicht relevant für die Projektarbeit, also umsonst. So ist zur Halbzeitabgabe diebezüglich nichts konkretes vorzuweisen, ausser dass die mySQL-DB schon steht (der Ehrlichkeit halber muss ich sagen, dass mich da die ESA für DB erheblich inspiriert hat ☺).

### **3.3.3 XML**

Auch die Versuche mit XML waren nur unzureichend, es soll dann bis Projektende über Daten aus der DB und mx:Tree umgesetzt werden in der Rubrik „Auswertung“

### **3.3.4 Datagrid und Chart**

Dies hängt eng mit der Datengewinnung zusammen, die wiederum an der DB hängt und da es da noch nichts gibt, ist hier auch noch nichts. Bis Projektende soll auch hier im Bereich „Auswertungen“ ein Set Anzeigevarianten zur Verfügung stehen analog der Umsetzung bei der Hilfe-Applikation.

## 4 Releaseplanung und -versionierung

Für die Programmversionen wird eine dreistellige Versionsnummer eingeführt. Die erste Stelle bezeichnet den Gesamtstand, die 2.Stelle die Anwendung und die 3. die Hilfe. Bei einem Versionsprung an der ersten Stelle, werden 2. und 3. wieder auf 0 gesetzt.

### 4.1 **0xx**

Es liegt keine vollständige Version vor mit der produktiv gearbeitet werden könnte. Jeder auf [www.projekt.festenberg.de](http://www.projekt.festenberg.de) publizierte Stand ist trotzdem technisch lauffähig.

0x4:

- XML-Ansicht für Inhalte über Treekomponente mit Darstellung des Contents zwischen einzelnen Tags

01x:

- Google-maps, login fertig und mit Tabsteuerung im Main verknüpft
- Layout für Main-Seite mit Tabregistern und Menü links im Tab
- Select-Komponente als Rohentwurf

02x:

- DB-Anbindung
- „Stammdaten“ fertig für Datenpflege

### 4.2 **1xx**

- Hilfe voll lauffähig, noch kein inhaltlich sinnvoller Kontent hinterlegt
  - Anwendung voll lauffähig mit allen geplanten Seiten inklusive DB-Anbindung
  - Jeweils nur 1 Basisvariante pro Seite (z.B. bei grafischer Datenaufbereitung der Auswertung)
  - DB-Inhalte sind stimmig und Betatest-tauglich, aber nicht vollständig
  - Benutzersteuerung ist hartcodiert (nur als Anwender und Admin)
- Ein Test mit 1-2 keyusern ist möglich

### 4.3 **2xx**

- Hilfeinhalte sind hinterlegt und erklären tatsächlich die Anwendung (lieber wenig, dafür aber ollständig)
- 2 oder mehr Anzeigevarianten für grafische Datenaufbereitung
- DB-Inhalte (insbesondere Stammdaten) sind vollständig
- Benutzersteuerung über DB (nur eigene Daten können geändert werden)

→ breiter Anwendungstest in der Abteilung ist möglich

#### **4.4 3xx**

- Layout vom User einstellbar
- Differenziertes Berechtigungskonzept auch für reine Anzeige
- Layout und Style über css und ausschließlich relative Positionierung der Anzeigeelemente

#### **4.5 Visionen**

- Schnittstelle zum SAP-HR
- ??