

---

# Entwurfsbeschreibung

---

Dokumentverantwortlicher: Martin Böhm

8. April 2014

In dieser Entwurfsbeschreibung werden die wesentlichen Struktur- und Entwurfsprinzipien für Vor- und - wo es sinnvoll ist - auch schon für das Gesamtprojekt dargelegt. Zudem werden die wesentlichen GUI-Elemente des Projekts skizziert.

<b>1</b>	<b>Allgemeines</b>	<b>2</b>
<b>2</b>	<b>Produktübersicht</b>	<b>2</b>
<b>3</b>	<b>Grundsätzliche Struktur- und Entwurfsprinzipien</b>	<b>5</b>
<b>4</b>	<b>Struktur- und Entwurfsprinzipien einzelner Pakete</b>	<b>5</b>
<b>5</b>	<b>Testkonzept</b>	<b>7</b>
<b>6</b>	<b>Glossar</b>	<b>7</b>

Funktion	Zugehörige Sicht	Abb.
(Initial-)Konfiguration der Dienstleistungen	Vollständige Baumansicht	1
Betrachten der kompatiblen Anbieter	Vollständige Ergebnisliste	2
Detailansicht eines Anbieters	Verkürzte Ergebnisliste und Detailansicht	-
Rekonfiguration	Baumansicht mit verkürzter Ergebnisliste	3
Ändern und Hinzufügen von Portfolien	Formular für Dateiauswahl	-

Tabelle 1: Zuordnung der Funktionalitäten zu den sie darstellenden Benutzersichten, vgl. auch die zugehörigen Abbildungen.

## 1 ALLGEMEINES

Ziel des Projekts ist die Erstellung einer Plattform, mit deren Hilfe es möglich sein soll, komplexe Dienstleistungen anbieterübergreifend zu konfigurieren und zu vergleichen. Anhand von Gemeinsamkeiten verschiedener Dienstleistungsangebote wird zunächst von der Partnergruppe SMD ein Standardportfolio erstellt, auf dessen Grundlage ein Mapping von individuellen Anbieterportfolien auf dieses Standardportfolio stattfinden kann. Dieses Mapping gewährleistet die Vergleichbarkeit der einzelnen Angebote nach bestimmten Kriterien. Dieser Vergleich ist Kern unseres Projekts. Der Kunde soll dabei eine übersichtliche, intuitiv zu bedienende Nutzeroberfläche vorfinden. Es werden weiterhin verschiedene Vergleichsmöglichkeiten geboten, sowie eine anschauliche Ergebnisseite erstellt, auf der sich der Kunde für den passendsten Anbieter entscheiden kann. Da diese Vorgehensweise und somit auch unsere Anwendung sich nicht auf einen bestimmten Dienstleistungssektor beschränkt, betrachten wir alle grundlegenden Funktionalitäten domänenübergreifend.

## 2 PRODUKTÜBERSICHT

Zur Konfiguration der Dienstleistungen und nachfolgenden Auswahl der passenden Anbieter wird dem Benutzer eine Reihe verschiedener Benutzersichten geboten, die die zur jeweiligen Funktion zugehörigen GUI-Elemente darstellen. Diese Zuordnung ist Tabelle 1 zu entnehmen. Die einzelnen Sichten sind prototypisch in den Abb. 1 bis 3 zu sehen bzw. werden im Folgenden erläutert.

**Detailansicht eines Anbieters:** Wird ein Anbieter ausgewählt, so werden detaillierte Informationen zu diesem angezeigt und Möglichkeiten der Kontaktaufnahme angeboten. Die Ergebnisliste zum schnellen Wechsel zwischen Anbietern wird derweil komprimiert dargestellt.

**Formular für Portfolio:** Das Hochladen von Portfolien wird durch einen üblichen Dateiauswahlvorgang gestartet.

In Abb. 4 sind die Zusammenhänge zwischen den verschiedenen Benutzersichten dargestellt.

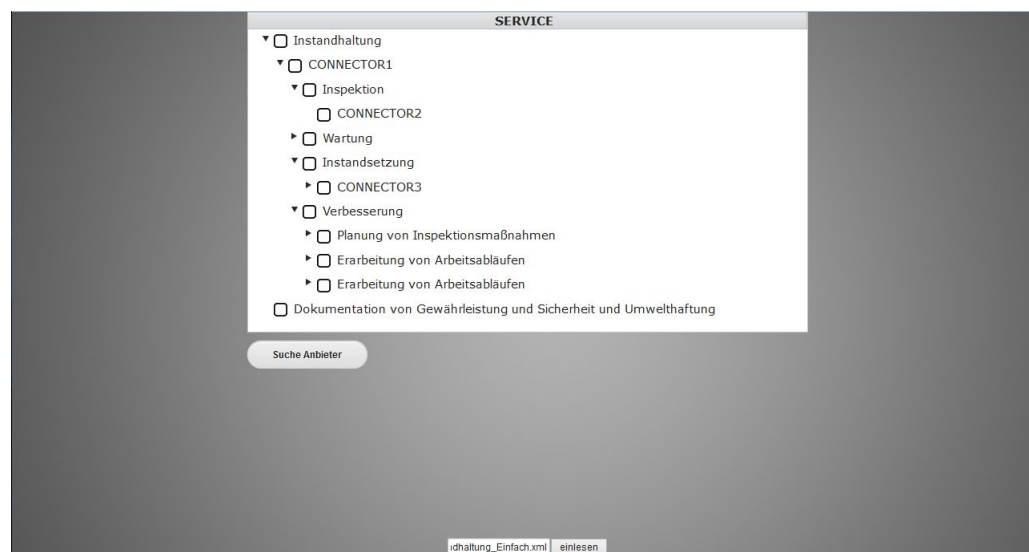


Abbildung 1: Ansicht zur Konfiguration der Dienstleistungen. Die Hierarchie wird durch Expandieren der entsprechenden Knoten aufgezeigt; Markieren der entsprechenden Checkboxen markiert die Auswahl der jeweiligen Dienstleistung. Für das Vorprojekt ist das Hochladen von Portfolien provisorisch realisiert.

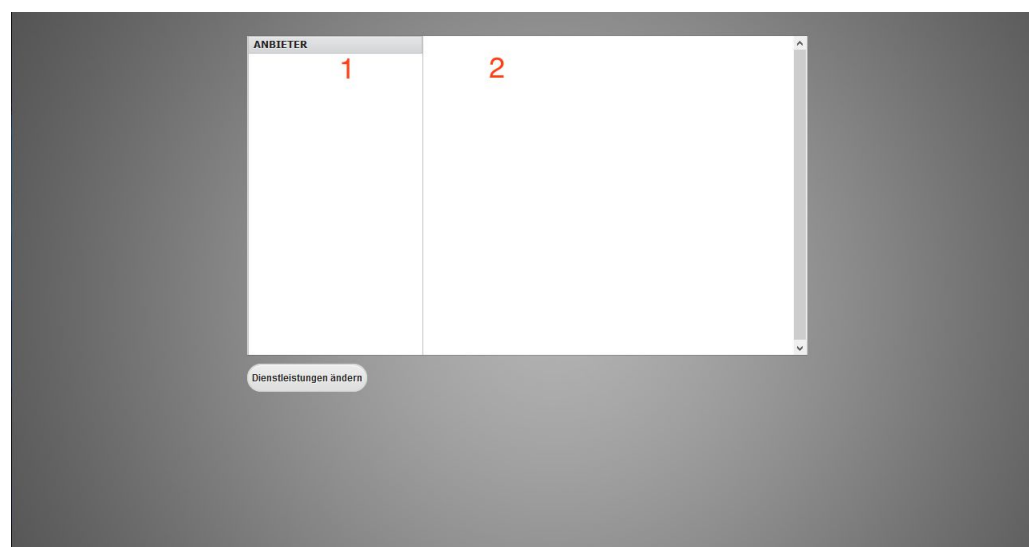


Abbildung 2: Nach Konfiguration der Dienstleistungen wird eine detaillierte Liste kompatibler Anbieter angezeigt (1). Neben dem Namen sollen dabei auch wesentliche Details wie der Preis angezeigt werden (2). Die Liste ist nach den jeweiligen Spalten sortierbar.

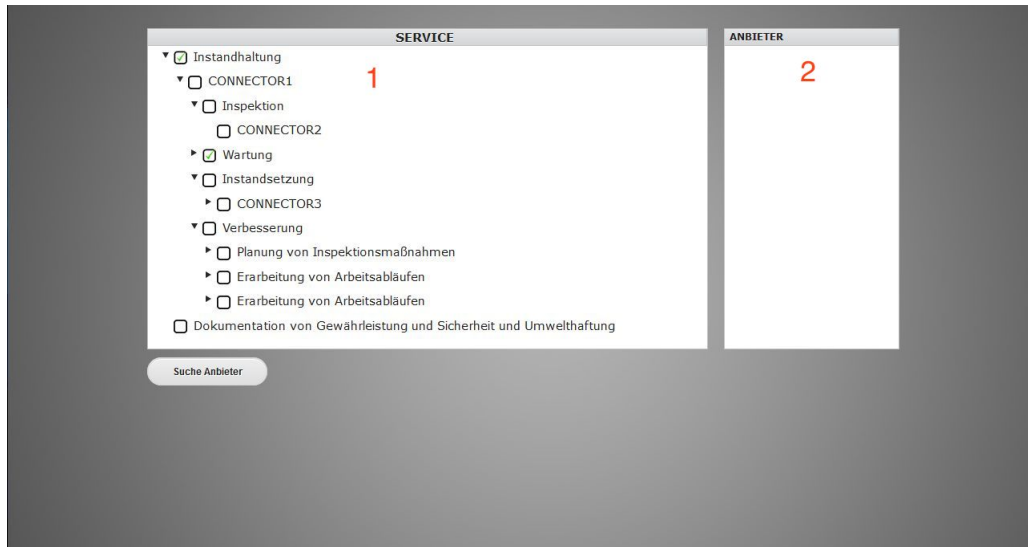


Abbildung 3: Liefert die Konfiguration kein zufriedenstellendes Ergebnis, kann der Benutzer eine Rekonfiguration vornehmen (1), während der die bisher gefundenen Anbieter weiter komprimiert angezeigt bleiben (2).

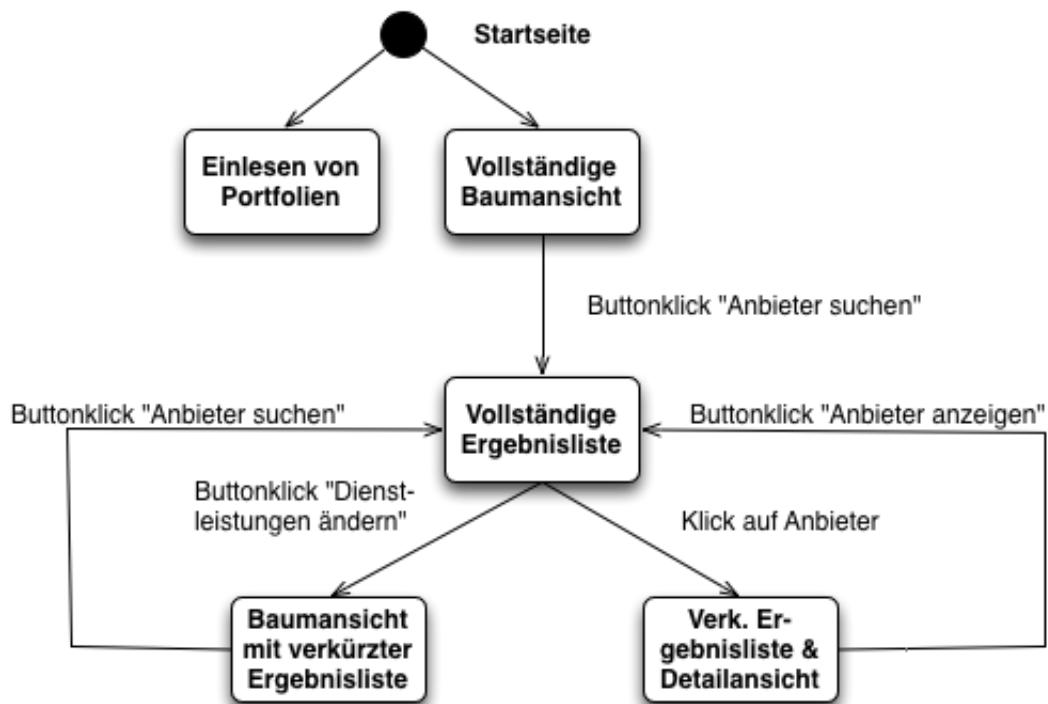


Abbildung 4: Übergänge zwischen den verschiedenen Benutzersichten. Die Rückkehr zur Startseite ist dabei jederzeit möglich.

### 3 GRUNDSÄTZLICHE STRUKTUR- UND ENTWURFSPRINZIPIEN

Abbildung 5 zeigt den bisherigen Entwicklungsstand der Java-Klassen und ihrer Beziehungen zueinander.

- Um der domänenübergreifenden Modellierung Rechnung zu tragen, sind die wesentlichen Objekte (Anbieter, Dienstleistungen, Portfolien) zunächst über Interfaces modelliert, sodass bei Bedarf später leicht noch andere Typen von Anbietern/Dienstleistungen entworfen werden können, die nur die entsprechenden abstrakten Methoden zu implementieren brauchen.
- Während im Vorprojekt aus naheliegenden Gründen das gesamte Benutzerinterface in der Klasse View implementiert wurde, wird für das Gesamtprojekt dem MVC-Paradigma entsprechend eine saubere Trennung zwischen dem eigentlichen Modell sowie View und Controller strikt gefordert.
- Das Benutzerinterface für die Webanwendung wird mittels Vaadin realisiert. Davon sind die Klassen "View" sowie "Controller" betroffen; das Modell muss selbstverständlich unabhängig davon funktionieren.

### 4 STRUKTUR- UND ENTWURFSPRINZIPIEN EINZELNER PAKETE

Da es bei diesem Projekt im Wesentlichen um die Darstellung der Benutzerschnittstelle geht, nehmen die "View"- und "Controller"-Klassen naturgemäß eine zentrale Rolle ein. Daher werden diese im Folgenden genauer erläutert<sup>1</sup>.

#### 4.1 DIE KLASSE VIEW

Wie bereits erwähnt, werden diese Klassen für die zu erstellende Webanwendung mittels des Java Frameworks Vaadin realisiert. Beim Design der View ist, wie schon in Abschnitt 2 skizziert, auf ein intuitives und übersichtliches Design zu achten. Die Ansicht ist dabei im Rahmen dieses Projekts für Desktop-Systeme zu optimieren.

Die Klasse "View" realisiert dabei die grafische Darstellung der in Abschnitt 2 erwähnten Benutzersichten durch Ausführen der entsprechenden Methoden:

- Die Methode "populateTree" bevölkert den Baum<sup>2</sup> der Dienstleistungen, der durch Einlesen der entsprechenden Portfolien implizit<sup>3</sup> gegeben ist. Dabei sind nur die Blätter tatsächliche Dienstleistungen; innere Knoten stellen lediglich Kategorien dar<sup>4</sup>.

---

<sup>1</sup>Die Informationen für das Gesamtprojekt folgen noch.

<sup>2</sup>Der Theoretiker denke sich eine fiktive Wurzel in der grafischen Darstellung.

<sup>3</sup>Man beachte, dass jede Instanz der Klasse SService als Member-Variablen Elternteil und Kinder (als Liste) kennt. Aus diesen Hierarchieinformationen wird der Baum aufgebaut.

<sup>4</sup>Im Vorprojekt ist dies noch nicht berücksichtigt, sodass auch "Kategorien" ausgewählt werden können, ohne dass dies eine sinnvolle Konsequenz nach sich zieht.

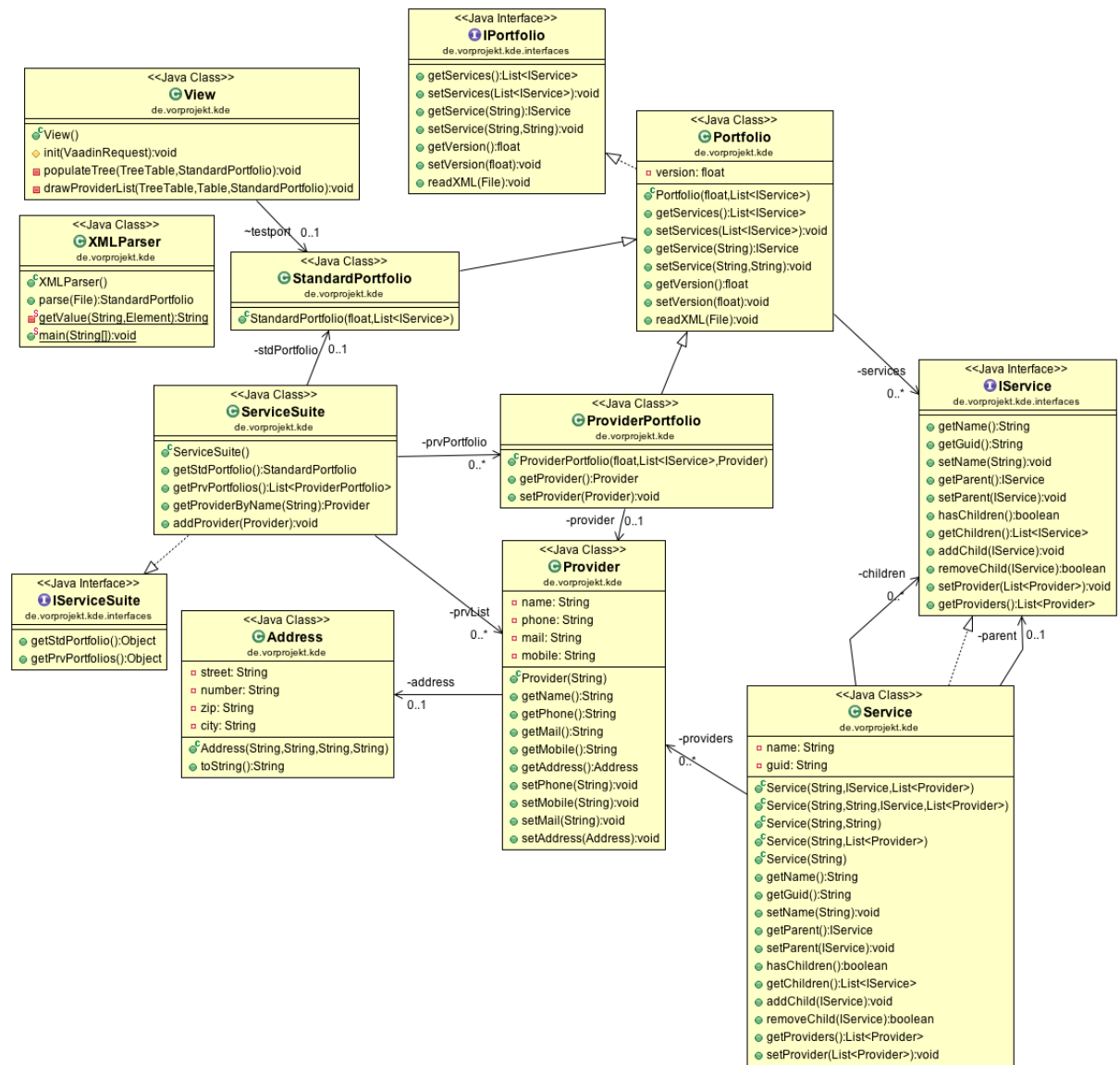


Abbildung 5: Darstellung der Java-Klassen und ihrer Beziehungen zueinander mittels UML.

- Die Methode “drawProviderList” baut - ausgehend von den tatsächlich ausgewählten Dienstleistungen - eine Ergebnisliste kompatibler Anbieter auf, die auch wesentliche Zusatzinformationen wie den geforderten Preis enthält und sortierbar sein wird. Logische Abhängigkeiten und Konsistenzbedingungen in Anbieterportfolien sind im Vorprojekt allerdings noch nicht realisiert.

## 4.2 PARSER

Die Parser-Klasse liest die XML-Dateien ein. Dabei müssen die Hierarchien und logische Abhängigkeiten für das Gesamtprojekt verlustfrei in die entsprechenden Java-Datenstrukturen überführt werden, was im Vorprojekt noch nicht realisiert ist.

## 5 TESTKONZEPT

Das Testkonzept ist im Qualitätssicherungskonzept detailliert beschrieben. Dort, wo automatisierte Tests einzelner Komponenten möglich sind - also im Wesentlichen bei der dem Modell zugeordneten internen Programmlogik - werden diese mittels JUnit implementiert. Die Integration- und Systemtests, die speziell auf die intuitive Bedienbarkeit der GUI und deren korrekte und ästhetische Darstellung in verschiedenen Browsern zielen, müssen dagegen manuell ausgeführt werden. Gefundene Fehler werden in Rallydev nach Art und Dringlichkeit kategorisiert und entsprechend zur Behebung an einen Programmierer delegiert. Dem Paradigma von gitflow folgend, werden die einzelnen User Stories weitgehend unabhängig voneinander und parallel in separaten Feature Branches implementiert und erst bei Fertigstellung der Kernfunktionalität - und unter der Voraussetzung, dass das Gesamtsystem ausführbar ist - in einen develop-Branch gemerged, der demzufolge stets einen mit Blick auf den derzeitigen Entwicklungsstand feature-konsistenten Entwicklungsstand enthält. Auf diese Weise ist die Erstellung wöchentlicher (und lauffähiger) Releasebundles zunehmender Funktionalität problemlos möglich, da die Arbeit an neuen Features nicht Gefahr läuft, vorübergehend neue, die Stabilität gefährdende Fehler einzuführen. Dieser Ablauf ist in Abb. 6 schematisch dargestellt.

## 6 GLOSSAR

### 6.1 KLASSIFIKATIONSSCHEMA

Klassifikationen werden zur planmäßigen Sammlung von abstrakten Klassen verwendet um eine Abgrenzung und Ordnung zu erzeugen. Es entsteht eine hierarchische Unterteilung in Ebenen.

Auf der entstehenden Plattform soll im ersten Schritt ein standardisiertes Portfolio, auf Basis branchenspezifischer Dienstleistungen modelliert werden. Es soll ein allgemeingültiges Modell geschaffen werden, anhand dessen ein Mapping individueller Anbieterportfolien stattfinden kann. Das Portfolio baut auf einer branchenspezifischen Datenbasis (Dienstleistungsklassifikation) auf, die eine einheitliche Terminologie und Strukturierung der Dienstleistungen bietet. Die Klassen werden dabei als Dienstleistungskomponenten in eine hierarchische Baumstruktur überführt, in welcher der Detaillierungsgrad nach unten hin zunimmt, bis die

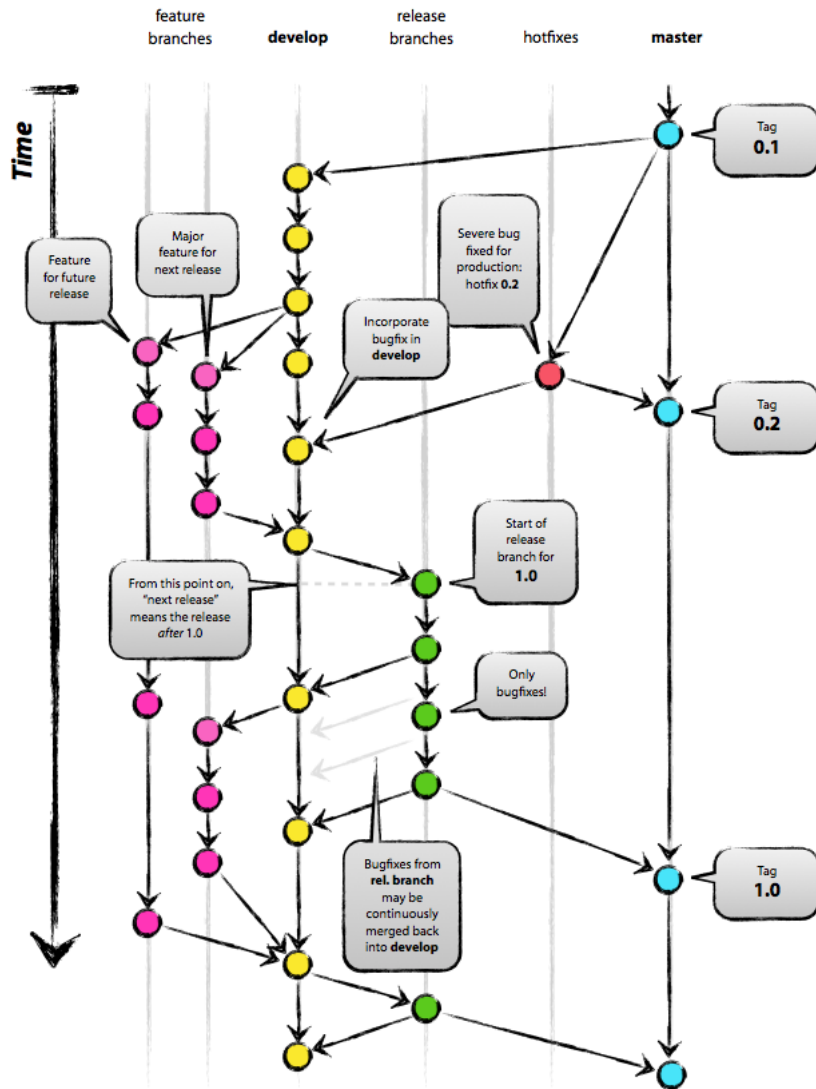


Abbildung 6: Schematische Darstellung des benutzten gitflow. Bildquelle: <http://nvie.com/posts/a-successful-git-branching-model/>, zuletzt abgerufen: 7.4.2014.



jeweilige Zusammensetzung der Dienstleistung hinreichend genau beschrieben ist.

#### 6.1.1 DIENSTLEISTUNGSKLASSIFIKATION

Klassifikation einer Dienstleistung anhand branchenspezifischer Daten.

#### 6.2 REFERENZMODELL

Ein Referenzmodell ist ein allgemeines Modell für eine Klasse von Sachverhalten mit folgenden Eigenschaften:

Auf Basis des allgemeinen Modells können spezielle Modelle geplant werden. Es kann als Vergleichsobjekt herangezogen werden. Das heißt, es ermöglicht Vergleiche mit anderen Modellen, die die gleichen Sachverhalte beschreiben.

Das Referenzmodell stellt somit ein Modellmuster bzw. Entwurfsmuster dar, das als idealtypisches Modell für die Klasse der zu modellierenden Sachverhalte betrachtet werden kann.

In unserem konkreten Fall dient das Plattformportfolio als Referenzmodell für die Anbieterportfolien.

#### 6.3 MAPPING

Verknüpfung der individuellen Daten der Anbieter (Anbieterportfolio) mit dem Referenzmodell (Plattformportfolio) zur Erzeugung der angestrebten Vereinheitlichung.

#### 6.4 KOMPLEXE DIENSTLEISTUNG

Komplexe Dienstleistungen bestehen aus einer Vielzahl unterschiedlicher Einzelleistungen, die zu einem Gesamtpaket kombiniert werden. Je mehr einzelne Leistungen erforderlich sind, um eine Dienstleistung zu erbringen, desto komplexer wird diese Dienstleistung.

Für das Projekt ist ein hoher Komponentisierungsgrad der Dienstleistungen erforderlich, also die Zerlegung der Dienstleistung in möglichst viele und kleine Leistungsbestandteile.

#### 6.5 BENUTZERGRUPPEN

Im wesentlichen gibt es zwei wichtige Benutzergruppen, welche auf die Plattform zugreifen werden. Diese werden im folgenden weiter erläutert.

##### 6.5.1 ANBIETER

Die erste große Gruppe sind die Anbieter, welche ihre Dienstleistungen an den Kunden bringen wollen. Sie stellen ihr individuelles Anbieterportfolio zusammen und mappen dieses mit dem standardisierten Plattformportfolio. Dieses Mapping erfolgt mit dem Tool der Partnergruppe, welches unter Konzepten kurz erläutert wird.

### 6.5.2 KUNDEN

Diese Gruppe kann über das zu erstellende Portal Dienstleistungen anbieterübergreifend vergleichen und nach ihren Bedürfnissen zusammenstellen.

## 6.6 PORTFOLIEN

Der Begriff Portfolio, bezeichnet eine Sammlung von Objekten eines bestimmten Typs. Im übertragenen Sinne kann es auch eine Sammlung von hilfreichen Methoden, Verfahren oder Handlungsoptionen bedeuten. In unserem Fall bieten die verschiedenen Portfolios einen Abriss der angebotenen Dienstleistungen der verschiedenen Anbieter, das Anbieterportfolio, bzw. das Standardportfolio der Plattform, auch Plattformportfolio genannt.

### 6.6.1 PLATTFORMPORTFOLIO

= Standardportfolio. Das Plattformportfolio bietet eine standardisierte Sammlung an Dienstleistungen. Es dient der Vereinheitlichung und der Vergleichbarkeit der vom Dienstleister angebotenen komplexen Dienstleistungen. Die Standardisierung wird durch die Unterteilung der komplexen Dienstleistung in kleinere, nicht weiter unterteilbare Dienstleistungskomponenten gewährleistet.

Die Dienstleistungen, welche vom Anbieter angeboten werden, werden auf dieses Plattformportfolio gemappt, wobei die Konfiguration im Servicemodeller vorgenommen wird. Dabei ist sicherzustellen, dass Mapping, Standard- und jeweiliges Anbieterportfolio zueinander kompatibel sind. Das Plattformportfolio wird ebenfalls im Servicemodeller konfiguriert.

### 6.6.2 ANBIETERPORTFOLIO

Das Anbieterportfolio stellt die individuelle Angebotspalette der einzelnen Dienstleister dar. Um eine Standardisierung und Vergleichbarkeit zu erreichen wird dieses stets auf das Plattformportfolio gemappt. Die Kunden können dann aus den verschiedenen gemappten Anbieterportfolios ihre bevorzugten Dienstleistungen auswählen und untereinander vergleichen.

## 6.7 HIERARCHISCHE BAUMSTRUKTUR

Eine Baumstruktur besteht aus klar definierten Teilen. Unter der Wurzel versteht man den Startpunkt der Struktur. Oft wird hierfür die englische Bezeichnung root verwendet. Von der Wurzel ausgehend, verzweigt sich die Struktur dann bis hin zu den untersten Elementen (Endpunkte). Jeder Baum kann aus mehreren Ebenen bestehen. Diese enthält jeweils untergeordnete Elemente, also weitere Verzweigungen oder auch Endpunkte.

## 6.8 DOMÄNE

Als Domäne oder Fachgebiet bezeichnet man sachbezogene Themenbereiche zur inhaltlichen Spezialisierung. Die Zuordnung muss nicht eindeutig sein, da themenübergreifende Arbeiten auch verschiedenen Domänen zugeordnet werden können. Bei unserem Projekt liegen die Schwerpunkte auf den Domänen „Erneuerbare Energien“ und „Dienstleistungsmanagement“.

## 6.9 WEBOBERFLÄCHE

Ist die visuelle Darstellung und Ausgestaltung eines Website-Dokumentes. Der Nutzer und Betrachter dieses Dokumentes kann mit diesem agieren, wobei auf einen hohen Grad an Intuitivität und Benutzerfreundlichkeit zu achten ist, damit der Nutzer wenig bis gar keine Zeit zum Eingewöhnen braucht und die Daten so aufbereitet sieht, dass er sie sofort für seine Zwecke gebrauchen kann.