

# Recherchebericht

## Inhaltsverzeichnis

<b>1</b>	<b>Begriffe</b>	<b>2</b>
1.1	Service Modeller . . . . .	2
1.2	XML . . . . .	2
1.3	Mapping . . . . .	2
1.4	Framework . . . . .	3
1.5	Laufzeitumgebung . . . . .	3
1.6	Silverlight . . . . .	3
1.7	Plug-In . . . . .	3
1.8	Browser . . . . .	4
1.9	C# . . . . .	4
1.10	.NET . . . . .	4
1.11	GUI / Front-End . . . . .	4
1.12	Back-End . . . . .	4
1.13	Visual Studio . . . . .	5
1.14	Plattform . . . . .	5
1.15	git . . . . .	5
1.16	RIA . . . . .	5
1.17	XAML . . . . .	6
1.18	LINQ . . . . .	6
1.19	WPF . . . . .	6
1.20	Portfolio . . . . .	6
<b>2</b>	<b>Konzepte</b>	<b>6</b>
2.1	Silverlight . . . . .	7
2.1.1	Back-End . . . . .	7
2.1.2	GUI / Front-End . . . . .	7
2.1.3	Datenstruktur . . . . .	7
2.2	Rich Internet Application (RIA) . . . . .	7

2.3	XML-Datenstruktur . . . . .	8
2.4	Framework . . . . .	8
2.5	Programmstand/Vorgabe . . . . .	8
2.6	Git-Repository . . . . .	9
<b>3</b>	<b>Aspekte</b>	<b>9</b>
3.1	Framework-Spezifikationen . . . . .	9
3.2	Mapping . . . . .	9
<b>4</b>	<b>Literaturverzeichnis</b>	<b>9</b>

# 1 Begriffe

## 1.1 Service Modeller

Der Service Modeller ist ein bereits geschriebenes Tool, mit dem ein Unternehmen in der Lage ist, seine Dienstleistungen in einer Art Baumstruktur zu präsentieren. Es können Knoten und Blätter eingefügt und editiert werden, so dass die Unternehmensstruktur in der Baumstruktur der einzelnen Dienstleistungen abgebildet erscheint. So erkennt man auf den ersten Blick über-/ untergeordnete, zusammengehende oder unabhängige Dienstleistungen. Auf der anderen Seite ist der Service Modeller das Interface für Kunden, die sich aus der Angebotspalette ihre individuell passenden Dienstleistungen zusammenstellen (lassen) können. Die Umsetzung des Service Modellers erfolgte in Microsoft Silverlight.

## 1.2 XML

„Extensible Markup Language“ ist eine zur logischen Dokumentauszeichnung geschaffene „Sprache“, die oft als Nachfolger von HTML gehandelt wird. Haupteinsatzgebiete sind Web-Publishing und Web-Services, sowie überhaupt das World Wide Web. XML ist ein einfaches textbasiertes maschinen- sowie menschenlesbares Format für den Austausch strukturierter Informationen. XML verspricht Interoperabilität zwischen Betriebssystemen und kann für Formate auf verschiedenster Hardware, wie z. B. Mobiltelefon, Fernseher, Multimediaeräte usw. verwendet werden.

U. a. die Serialisierbarkeit von XML-Dokumenten erlaubt deren problemlose Übertragung zwischen mehreren Systemen.

## 1.3 Mapping

Unter Mapping versteht man ursprünglich die Kartierung, z.B. von Geodaten. Im Bereich Computergrafik sind diverse Arten von Mapping bekannt, so z. B. Texture Mapping, bei dem beispielsweise ein zweidimensionales Bild auf einen Bereich einer dreidimensionalen Grafik projiziert wird.

In unserem Projekt ist Data Mapping gemeint, womit „Verbinden“ gemeint ist. Zusammenhänge zwischen zwei Datenmodellen können hergestellt und damit z.B. semantische oder strukturelle Verbindungen zwischen Objekten ermöglicht werden. Es wird eine möglichst einfache Realisierung des Mapping-Vorgangs angestrebt, so z.B. Ziehen von Verbindungslinien (u. U. mit verschiedenen Bedeutungen/Farben).

## 1.4 Framework

Ein Framework (engl. Rahmenstruktur) ist ein Gerüst für die Softwaretechnik und objektorientierte Softwareentwicklung. Der Programmierer kann in ihr entwickeln und auf eine Vielzahl von vorgefertigten Funktionen zurückgreifen. Anders als eine Ansammlung von bereits geschriebenen Klassen, liefert ein Framework eine bereits fertige Struktur und Schnittstellen.

## 1.5 Laufzeitumgebung

Die Laufzeitumgebung ist der Vermittler zwischen einem nicht direkt ausführbaren Anwendungsprogramm und dem Betriebssystem. Ferner beschreibt der Begriff das Umfeld eines ablaufenden Programms, falls dieses direkt mit einem anderen Programm oder Betriebssystem kommuniziert. Hierzu lädt die Laufzeitumgebung unabhängige Programme, welche eine eigene Plattform bilden und vermittelt zwischen Anwendungsprogramm und Zielsystem. Microsoft entwickelte zur Kommunikation zwischen Softwarekomponenten das Component Object Model (COM). Da hierdurch eine zusätzliche einheitliche Programmierung nötig war, wurde der einheitliche Code jedoch unnötig komplex. Deshalb wurde das COM mit Einführung des .NET Frameworks zur Common Language Runtime weiterentwickelt. Diese Laufzeitumgebung prüft zum Beispiel jeden Methodenaufruf oder jedes Objekt auf eine einheitliche Version und erzeugt dadurch einen prozessorunabhängigen Zwischencode.

## 1.6 Silverlight

Silverlight ist eine von Microsoft entwickelte Internetbrowser-Erweiterung, um das „Rich Internet Application“-Format anzuzeigen. Es basiert auf einer Version des .NET-Frameworks. Der wichtigste Grundsatz in Silverlight ist die Plattform-Unabhängigkeit. Aktuelle Konkurrenten des bereits zurückgehenden Silverlight-„Hypes“ sind Adobe Flash und JavaFX.

## 1.7 Plug-In

Plug-Ins sind Softwaremodule, die eine bestehende Anwendung auch zur Laufzeit erweitern und neue Funktionalitäten hinzufügen.

## 1.8 Browser

Ein Browser ist eine clientseitige Software, die das Frontend einer Webanwendung, meist einer Webseite, ausgibt und die gesamte damit verbundene Abwicklung übernimmt. Diese Abwicklungen beziehen sich hauptsächlich auf das Senden und Empfangen von Daten über Netzwerkprotokolle, typischerweise HTTP.

## 1.9 C#

C# ist eine objektorientierte Programmiersprache, die für die Mechanismen hinter der Darstellung sorgt. Sie wurde von Microsoft speziell für .Net entwickelt. Beeinflusst wurde C# von Programmiersprachen wie C++, Java und Delphi. Die Sprache an sich ähnelt sehr Java, war jedoch zu Beginn nicht plattformunabhängig. Durch das Open-Source-Projekt Mono hat sich dies jedoch geändert durch die Entwicklung einer plattformunabhängigen .Net Version inklusive C#-Compiler.

## 1.10 .NET

Microsoft .NET ist eine Kombination aus Laufzeitumgebung und Framework. .Net unterstützt zudem noch eine Vielzahl von Programmiersprachen. Durch .Net ist es möglich mit verschiedenen Programmiersprachen an einem Projekt zu arbeiten. .Net übersetzt die Hochsprachen (zB. C#) in eine Zwischensprache (CIL = Common Intermediate Language). Bei dieser „Zwischensprache“ handelt es sich um eine Universalsprache, welche von der in .Net integrierten Laufzeitumgebung in Maschinensprache übersetzt wird.

## 1.11 GUI / Front-End

Als GUI oder auch Front-End bezeichnet man die graphische Benutzeroberfläche eines Programms oder einer Applikation. Der Benutzer verwendet sie meistens für Eingaben (Buttons, Textfelder, Scrollbars, etc.) und erhält auch entsprechende Ausgaben. Demnach ist die GUI der Teil einer Software, in welcher der Benutzer mit dem Programm interagiert. Dies ist per Tastatur oder Maus, sowie mit diversen spezialisierten weiteren Eingabegeräten möglich. Um eine möglichst umfassende Plattformunabhängigkeit zu erreichen, wird das Front-End von der Programmlogik getrennt und je nach Plattformbedingungen gestaltet. So ist der Entwicklungsaufwand nur im Bereich der Darstellung einzuschätzen und wesentlich geringer als bei komplett neuer Implementierung des ganzen Programmes.

## 1.12 Back-End

Das Back-End ist das komplette Modell hinter der graphischen Darstellung und daher für den User nicht zugänglich. In Web-Applications ist es der Code, der serverseitig auf dem Webserver läuft und die Logik der Applikation enthält, wie z. B.: Datenspeicherung, Verifizierung von Benutzereingaben, etc. Die Aufteilung in Front-End und Back-End einer Anwendung ist nicht immer sinnvoll, da eine klare Definition und Abgrenzung

schwierig ist. Eine Gebrauchsdefinition kann dabei sein, dass das Front-End die Schnittstelle zum Benutzer darstellt (Buttons, Textfelder, etc.), das Back-End als Programmlogik näher zum System umfasst.

### 1.13 Visual Studio

Visual Studio ist eine Entwicklungsumgebung von Microsoft. Ab 2002 unterstützt Visual Studio das Microsoft .Net Framework, welches bis heute primär ist. Die .NET Anwendungen können in Basic, C++, C#, JavaScript und F# sowie weiterer kleiner Programmiersprachen implementiert werden.

Konzeptuell bildet Visual Studio hierzu Projektmappen, welche auf bereitgestellten Vorlagen basieren und sowohl verschiedene Oberflächentechniken als auch die Integration von Code in andere Microsoft-Anwendungen wie Office oder SQL-Server ermöglichen.

Insgesamt handelt es sich bei der von uns verwendeten Professional Version 2013 um eine mächtige Tool-sammlung zum Entwickeln von komplexen Anwendungen.

### 1.14 Plattform

Unter einer Computerplattform versteht man meist ein Betriebssystem (etwa Windows, Ubuntu etc.) aber auch die Systemarchitektur der Hardware. Wegen der spezifischen Eigenheiten der Betriebssysteme bzw. der Maschinen, auf denen sie ausgeführt werden, ist ein Programm, welches für ein bestimmtes Betriebssystem auf einem bestimmten Hardwaretyp entwickelt wurde, meist nicht auf einem anderen Betriebssystem/Hardwaretyp ausführbar. Um diese (Plattform-)Abhängigkeit zu umgehen, greifen Entwickler gerne auf Laufzeitumgebungen zurück, die es für möglichst viele Betriebssysteme gibt oder entwickeln ihr Programm so, dass es direkt im Internetbrowser ausgeführt wird.

### 1.15 git

git ist ein Programm zum Anlegen und Verwalten sogenannter Repositories. Repositories können Aufbewahrungsort für jede Art von Textdokument sein, häufig sind es jedoch Programmcode-Teile. Wird ein Dokument zu einem Repository hinzugefügt, so kann mit git jede Veränderung daran festgehalten werden. Da man häufig in einer Gruppe an Dokumenten arbeiten will, ist es üblich, dass sich jedes Gruppenmitglied eine vollständige Kopie des zentralen Repository anlegt und an diesen Daten arbeitet. Die Änderungen, die die Gruppe für sinnvoll hält, werden dann in das zentrale Repository übernommen. Diese Arbeitsweise wird auch Branching (engl. Verzweigen) genannt.

### 1.16 RIA

steht für Rich Internet Application. Unter einer Internet Application oder auch Web-App versteht man ein Programm, welches direkt im Internetbrowser ausgeführt wird, ohne dass dazu das Programm selbst auf dem eigenen Rechner installiert werden muss. Der Zusatz „Rich“ sagt aus, dass es sich um eine Internet

Application handelt, die ein besonders großes Funktions- und Interaktionsangebot bereitstellt, so etwa das Bedienen per Tastenkürzel, den Dateiimport durch Ziehen der Datei in das Browserfenster, Unterstützung von Animationen, Audio- und Videodaten und vielem mehr. Dadurch reichen RIAs an den Funktionsumfang von Programmen heran, die direkt auf dem eigenen Rechner installiert sind.

### 1.17 XAML

steht für Extensible Application Markup Language und ist eine von Microsoft entwickelte abstrakte Sprache für die Gestaltung von Bedienoberflächen von Programmen und basiert auf XML. XAML wird in Programmen für Windows und Silverlight verwendet.

### 1.18 LINQ

LINQ (Language Integrated Query) ist ein von Microsoft entwickeltes Verfahren, um den Zugriff auf Daten zu organisieren. Ein Programm bezieht seine Daten aus verschiedenen internen und externen Quellen. Abhängig davon, wie die externen Quellen aussehen, müssen unterschiedliche Methoden angewendet werden (Datenbanken, XML, einfache Dateien). LINQ versucht eine Zugriffsmethode für verschiedene Datei- und Speichertypen zur Verfügung zu stellen.

### 1.19 WPF

Die Windows Presentation Foundation (WPF) ist ein Grafik- (auch GUI)-Framework, welches Teil des großen .NET-Frameworks aus dem Hause Microsoft ist. WPF versucht die durch XAML definierten Oberfläche mit Grafiken, audiovisuellen Inhalten und Typographien performant zu „präsentieren“.

### 1.20 Portfolio

Ein Portfolio ist in erster Linie eine Sammlung von Objekten gleichen Typs. Unter Portfolio eines Unternehmens, z. B. Dienstleistungs- oder Produktportfolio versteht man die Zusammenstellung aller angebotenen z. B. Dienstleistungen oder Produkte.

## 2 Konzepte

Der Service-Modeller ist ein C#-basiertes Dienstleistungstool in einer per Microsoft Silverlight implementierten Webinterface-Umgebung. Das Projekt befasst sich damit, dieses Tool um Mapping-Features von individuellen Offerten auf Standard-Portfolios zu erweitern. Damit ist das grundlegende technische Fundament der Implementierung gegeben.

## 2.1 Silverlight

### 2.1.1 Back-End

Microsoft Silverlight ist eine angepasste cross-platform Variante des .Net-Framework 3.5, um Rich Internet Applications (RIA) schreiben und anwenden zu können. Die Silverlight-Technologie ermöglicht über Webinterfaces eine Funktionalitätsvielfalt bereitzustellen, die ansonsten nur mit clientseitig installierten Anwendungen in jeweils separatem Bezug auf die Plattform realisierbar ist. So kann eine redundante plattformabhängige Implementierung vermieden werden und mit einer Entwicklung die Nutzbarkeit auf verschiedenen Plattformen ermöglicht werden. Um Silverlight-Anwendungen benutzen zu können, ist userseitig ein Crossover-Webbrowser (Firefox, Safari, Chrome, IE, Opera, etc.) Plug-In zu installieren. Die Anwendungsvielfalt umfasst dabei interprozesskommunikative Datenströme, Multimedia (Sounds, Grafiken, Videos) und Animationen. Entwicklern werden dabei Schnittstellen für Command-Line-Interfaces und Entwicklungsumgebungen geboten.

### 2.1.2 GUI / Front-End

Silverlight beinhaltet in seiner Präsentationsschicht ein Grafiksystem, ähnlich dem WPF (Windows Presentation Foundation), indem Layout-Elemente und Programmlogik durch XAML (Extensible Application Markup Language) basierend auf XML getrennt bearbeitet und dargestellt werden können. Dies hat einige positive Auswirkungen auf die Leistungs- und Darstellungsperformance der Anwendungen. Aus diesem Grund ist Silverlight auch unter dem Namen Windows Presentation Foundation Everywhere (WPF/E) bekannt.

### 2.1.3 Datenstruktur

Bedingt durch diese Trennung ist eine Bearbeitung von XML-(Extensible Markup Language) Dateien über Silverlight-Anwendungen gut zu implementieren. So gibt es als Alternative zu dem in die Jahre gekommenen Document Object Modell (DOM) Verfahren eine eigene Schnittstellenarchitektur, nämlich System.XML.Linq.

## 2.2 Rich Internet Application (RIA)

RIAs sind Internet-Anwendungen, die sich in ihrem Funktionsumfang Desktopanwendungen annähern. Typischerweise sind sie für Webbrowsererweiterungen (wie das Silverlight-Plugin) entwickelt. Der Vorteil ist, dass nur das Browser-Plug-in installiert und nicht ein jeweils eigenständiges Programm installiert werden muss. Die Internet-Anwendungen selbst bieten dann eine hohe Reichweite von Interaktionspotentialen. So können Features der Desktop-Anwendungen übernommen werden. Gemeint sind hier Möglichkeiten wie das Benutzen von Drag&Drop, GUI-Virtualisierungen oder Zugriff auf das Daten- und Dateisystem sowie zu einigen Systemeinstellungen. Rich Internet Application stellen so die Verbindungsbrücke von rich-Anwendungen, die hohe User-Interaktivität und -Komfort garantieren, mit reach-Anwendungen, eine hohe Erreichbarkeit und

plattformübergreifende usability, dar. Dieses Konzept benötigt eine architektonische Technologie, so z.B. Silverlight. Die Überführung in Desktop-Anwendungen ist bei gegebener permanenter Internetanbindung ohne Einschränkungen möglich.

## 2.3 XML-Datenstruktur

Das Silverlight-Projekt basiert auf einer XML-Datenstruktur. Diese erweiterbare Auszeichnungssprache ist eine Form der Darstellung von hierarchisch strukturierten Daten als Textdatei. Somit ist eine plattform- und implementationsunabhängige Bearbeitung von verschiedenen Systemen auf einheitlichen Datenbestand möglich. Über die Funktionsbereitstellungen von System.XML.Linq ist die Schnittstelle von RIA zu XML vorhanden und nutzbar. LINQ ist dabei eine Lösung von Microsoft, wie objektorientierte Bearbeitung mit Datenstrukturen verbunden werden können. Es ist verwendbar für SQL, XML, verschiedenen Web-Services usw., s. o..

## 2.4 Framework

Das Framework bietet verschiedene Angebote für die Auswahl der Framework Language. Über das Framework Developer Tool Visual Studio 2013 ist eine Implementierung in verschiedenen Programmiersprachen möglich. Zur Auswahl stehen dabei C#, Visual Basic, JScript, IronRuby, IronPython, sowie VBx (Visual Basic Extension). Als Serverumgebungen bieten sich Microsoft-Produkte wie der Microsoft Server 2013 an. Da der Service-Modeller bisher auf einer solchen Umgebung läuft, wird in Anpassung an die weiteren Erfordernisse auf diese Realisationsumgebung gesetzt.

## 2.5 Programmstand/Vorgabe

Der Service Modeller stellt ein Referenzmodell für Dienstleistungen im Bereich erneuerbarer Energien dar (Forschungsprojekt). Der bisherige Projektstand soll erweitert werden, so dass Dienstleistungskomponenten auf standardisierte Referenzportfolios zugeordnet werden können. Der Service Modeller arbeitet in einer Komponentendarstellung. Es können in der Dienstleistungserstellung Komponenten erstellt werden, die, definiert durch Konnektoren, in Verbindung zu anderen Konnektoren stehen. Dadurch ergibt sich eine Baumstruktur, in welcher die Dienstleistungsportfolios des Unternehmens visualisiert werden können. Zudem ist eine Bewertung der Komponenten mit Kennzahlen (Key Performance Indicators = KPI) möglich. Mit der fertigen Dienstleistungsstruktur können dann Kunden im Configurator ihre individuellen Anforderungen zusammenstellen.

Entwickelt werden muss dazu eine grafische Benutzeroberfläche des Service Modellers, die diese Features zur Verfügung stellt. Bestimmungen und Bedingungen, die der Service Modeller für das Speichern der Zuordnungen müssen dabei eingehalten werden.



Zudem ist eine Export-Funktionalität einzubinden, die es erlaubt, die Zuordnung in Form einer XML-Datei auszugeben und somit für eine weitere Bearbeitung ohne Service Modeller zur Verfügung zu stellen.

Dabei werden die Dienstleistungen standardisiert in die Standard-Portfolios in Form dieser XML-Dateien gespeichert bzw. verarbeitet. Durch die LINQ-Schnittstelle sollen diese Dateien in den Service Modeller geladen werden können. Separat soll in dem gleichen Darstellungsbereich der Software ein weiteres Fenster für das Einbinden von Individual-Dienstleistungs-Portfolios implementiert sein. Die nun ge„mapp“ten Dienstleistungen sollen dann als XML-Datei exportiert werden können.

## 2.6 Git-Repository

Um das Projekt effizient durchführen zu können, soll ein git-Repository verwendet werden, um mit Hilfe von „Branching“ an verschiedenen Stellen gleichzeitig programmieren zu können und so die Arbeit besser aufzuteilen oder um z.B. keine Datei in mehreren Versionen im Umlauf zu haben. Es lassen sich auch Dateiversionen zurücksetzen oder von anderen Gruppenmitgliedern kontrollieren. Dies kann je nach Komplexität oder Anforderungen der entsprechenden Programmteile nötig werden.

## 3 Aspekte

### 3.1 Framework-Spezifikationen

Da das Projekt in die Kategorie des Weiterentwicklung anzusiedeln ist, erscheint eine weitere Nutzung der bisherigen Umgebung und Sprache sinnvoll. So lassen sich Schnittstellen-Probleme vermeiden. Die Logik des Programmes wird deswegen in der Programmiersprache C# in Visual Studio 2013 entwickelt werden.

### 3.2 Mapping

Mit Rich Internet Application Features soll der User nun Verbindungen (Mapping) der Einzelangebote auf das Standard-Portfolio binden. Konkret ist daran gedacht, dies in der Form umzusetzen, dass grafische Verbindungslinien von der individuellen Angebotszusammenstellung auf ein Standardportfolio gezogen werden. Als Alternative soll ein Verbindungs-Interface zur Verfügung gestellt werden, in dem das Objekt der Offerte sowie das Objekt des Standardportfolio eingetragen werden und somit die Zuordnung stattfindet und gespeichert werden kann.

## 4 Literaturverzeichnis

- [1] 1-2014. URL: <http://www.itwissen.info/definition/lexikon/Browser-browser.html>.
- [2] 1-2014. URL: <http://wirtschaftslexikon.gabler.de/Definition/plugin.html>.

- 
- [3] Olaf Bergmann. *XML. Grundlagen : Einführung in die logische Dokumentauszeichnung*. Hrsg. von Heinz Wittenbrink. Bd. Bd. 1. Berlin: SPC Teia Lehrbuch Verl., 2005, 154 S. ISBN: 393553955X.
  - [4] KoProServ. *Service Modeller: 5-Punkte-Kurzanleitung*. 2013. URL: <http://koproserv.uni-leipzig.de/wp-content/uploads/2012/12/Service-Modeller-Kurzbeschreibung.pdf>.
  - [5] Robert Lair. *Beginning Silverlight 5 in C#*. Apress, 2012.
  - [6] Matthew MacDonald. *Pro Silverlight 5 in C#*. Apress, 2012.
  - [7] Johannes Nowak/Daniel Moses. *Programmieren unter .net. C# Edition*. ISBN: 3772372244.
  - [8] Microsoft Developer Network. *Die .NET Common Language Runtime*. 2013. URL: <http://msdn.microsoft.com/de-de/library/bb979570.aspx>.
  - [9] IT Vision. *Visual Studio als Entwicklungsumgebung*. 2013. URL: [http://www.it-visions.de/glossar/alle/2422/Visual\\_Studio\\_als\\_Entwicklungsumgebung\\_f%C3%BCr\\_.NET.aspx](http://www.it-visions.de/glossar/alle/2422/Visual_Studio_als_Entwicklungsumgebung_f%C3%BCr_.NET.aspx).
  - [10] Wikipedia. *C-Sharp*. 2013. URL: <http://de.wikipedia.org/wiki/C-Sharp>.
  - [11] Wikipedia. *Framework*. 2013. URL: <http://de.wikipedia.org/wiki/Framework>.
  - [12] Wikipedia. *Laufzeitumgebung*. 2013. URL: <http://de.wikipedia.org/wiki/Laufzeitumgebung>.
  - [13] Wikipedia. *Mapping*. 2013. URL: <http://de.wikipedia.org/wiki/Mapping>.
  - [14] Wikipedia. *.Net*. 2013. URL: <http://de.wikipedia.org/wiki/.NET>.
  - [15] Wikipedia. *Portfolio*. 2013. URL: [http://de.wikipedia.org/wiki/Portfolio#Vertrieb.2C\\_Marketing.2C\\_Produktionsmanagement](http://de.wikipedia.org/wiki/Portfolio#Vertrieb.2C_Marketing.2C_Produktionsmanagement).
  - [16] Wikipedia. *Silverlight*. 2013. URL: <http://de.wikipedia.org/wiki/Silverlight>.
  - [17] Wikipedia. *Visual Studio*. 2013. URL: [http://de.wikipedia.org/wiki/Visual\\_Studio](http://de.wikipedia.org/wiki/Visual_Studio).
  - [18] Wikipedia. *WPF*. 2013. URL: [http://de.wikipedia.org/wiki/Windows\\_Presentation\\_Foundation](http://de.wikipedia.org/wiki/Windows_Presentation_Foundation).
  - [19] Wikipedia. *Windows Presentation Foundation*. 2013. URL: [http://de.wikipedia.org/wiki/Windows\\_Presentation\\_Foundation](http://de.wikipedia.org/wiki/Windows_Presentation_Foundation).