



Hochschule Augsburg University of Applied Sciences

Diplomarbeit Studienrichtung Informatik

Jorge Andrés Cuartas Monroy

Vergleich von Python-Webframeworks
am Beispiel einer Webschnittstelle
für Mikrocontroller

Verfasser der Diplomarbeit:
Jorge Andrés Cuartas Monroy
Rehmstr. 7
866161 Augsburg
Telefon: 0176 21232896
jorge.cuartas@googlemail.com

Fakultät für Informatik
Telefon: +49 821 5586-3450
Fax: +49 821 5586-3499

Hochschule Augsburg
University of Applied Sciences
Baumgartnerstraße 16
D 86161 Augsburg

Telefon +49 821 5586-0
Fax +49 821 5586-3222
<http://www.hs-augsburg.de>
poststelle@hs-augsburg.de

Erstprüfer: Prof. Dr. Hubert Högl

Zweitprüfer:

Abgabe der Arbeit:

Sie dürfen:

- Das Werk bzw. den Inhalt vervielfältigen, verbreiten und öffentlich zugänglich machen
- Abwandlungen und Bearbeitungen des Werkes bzw. Inhaltes anfertigen

Zu den folgenden Bedingungen:

- Namensnennung — Sie müssen den Namen des Autors/Rechteinhabers in der von ihm festgelegten Weise nennen.
- Keine kommerzielle Nutzung — Dieses Werk bzw. dieser Inhalt darf nicht für kommerzielle Zwecke verwendet werden.
- Weitergabe unter gleichen Bedingungen — Wenn Sie das lizenzierte Werk bzw. den lizenzierten Inhalt bearbeiten oder in anderer Weise erkennbar als Grundlage für eigenes Schaffen verwenden, dürfen Sie die daraufhin neu entstandenen Werke bzw. Inhalte nur unter Verwendung von Lizenzbedingungen weitergeben, die mit denen dieses Lizenzvertrages identisch oder vergleichbar sind.

Wobei gilt:

- Verzichtserklärung — Jede der vorgenannten Bedingungen kann aufgehoben werden, sofern Sie die ausdrückliche Einwilligung des Rechteinhabers dazu erhalten.
- Sonstige Rechte — Die Lizenz hat keinerlei Einfluss auf die folgenden Rechte:
 - Die gesetzlichen Schranken des Urheberrechts und sonstigen Befugnisse zur privaten Nutzung
 - Das Urheberpersönlichkeitsrecht des Rechteinhabers
 - Rechte anderer Personen, entweder am Lizenzgegenstand selber oder bezüglich seiner Verwendung, zum Beispiel Persönlichkeitsrechte abgebildeter Personen.
- Hinweis — Im Falle einer Verbreitung müssen Sie anderen alle Lizenzbedingungen mitteilen, die für dieses Werk gelten. Am einfachsten ist es, an entsprechender Stelle einen Link auf diese Seite einzubinden.

Hinweis

Sie können den Lizenztext unter <http://creativecommons.org/licenses/by-nc-sa/3.0/deed.de> nachlesen.

Erklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Augsburg, den Tag. Dezember Jahr

Jorge Andres Cuartas Monroy

Inhaltsverzeichnis

1	Einführung	6
1.1	Beispielanwendung	6
1.2	Durchführung	7
1.2.1	GUI	7
1.2.2	Datenbank	8
1.2.3	Logik	8
2	Webframeworks	9
2.1	Definition	9
2.2	Warum Python	9
2.3	Fullstack vs. Glue	10
2.4	Weitere Unterscheidungsmerkmale	11
2.4.1	Funktionalität	11
2.4.2	Dokumentation	11
2.5	Bekannte Webframeworks im Pythonumfeld	12
2.5.1	Überblick	12
2.5.2	Django	12
2.5.3	Pylons	12
2.5.4	Zope	12
2.5.5	web2py	13
2.6	Webframeworks anderer Programmiersprachen	13
2.6.1	Überblick	13
2.6.2	Zend	13
2.6.3	Ruby on Rails	13
2.6.4	Catalyst	14
2.6.5	Wicket	14
2.7	Arbeiten ohne Framework	14

3	Django	16
3.1	Einführung	16
3.2	Installation	16
3.3	Dokumentation	16
3.4	Datenbanksysteme	16
3.5	Templatesystem	16
3.6	JavaScript	16
3.7	Sicherheitsaspekte	16
3.8	Durchführung der Anwendung	16
4	TurboGears	17
4.1	17
5	Pylons	18
5.1	Einführung	18
6	Zope	19
6.1	Einführung	19
7	Web2Py	20
7.1	Einführung	20
8	Ergebnisse / Übersicht	21
8.1	Bewertungskriterien	21

1 Einführung

1.1 Beispielanwendung

Im Zuge der Diplomarbeit soll neben einem Vergleich von Pythonwebframeworks auch eine Anwendung für des Rechnertechniklabor der HS Augsburg entwickelt werden. Dabei soll der Zugriff auf Informationen und Verwaltung der Mikrocontroller in Form einer Webanwendung möglich sein. Dabei kann der Benutzer die einzelnen Boards reservieren und anschließend per SSH über das Internet mit den Boards arbeiten können. Damit ein reibungsloser Umgang auf den Boards ermöglicht wird soll jeweils einem einzelnen Benutzer gleichzeitig der Zugang auf ein Board gewährt werden. Darüber hinaus wird folgende Funktionalität von der Anwendung aus zugänglich:

- Power on/off
- Reset
- Statusinformationen abrufen
- Reservierung
- Webupload der Root-Filesysteme

Weiterhin bekommt der Benutzer die Möglichkeit Statusinformationen abzurufen, die auf der Weboberfläche angezeigt werden. Darüber hinaus auch ein Webupload von Rootfilesystemen möglich, die auf dem Board mit Hilfe von Skripten auf die Boards automatisch aufgespielt werden. Zu den Arbeiten gehört weiterhin die Installation entsprechender Frameworks und nötige Software für den Reibungslosen Ablauf der Applikation. Die Anwendung wird gegebenenfalls unter verschiedenen Technologien implementiert. Nach der Reservierung der Boards bekommt der Benutzer einen Zugang mit Passwort, damit dieser über SSH auf das Board zugreifen kann. Das Passwort wird nach Reservierung jeweils neu erstellt und nach dem Abmelden aus der Userverwaltung oder nachdem

der User nicht mehr auf dem Board arbeiten will der Zugang gesperrt, bis ein anderer das Board reserviert.

Für jedes der hier vorgestellten Webframeworks wird diese Anwendung entwickelt. Es soll nach dem Durchlesen dieser Arbeit möglich sein, ein Vergleich zu ziehen, welches Framework die beste Unterstützung bietet, um Webprojekte im Pythonumfeld zu erstellen. Normalerweise denkt man heutzutage bei Webentwicklung hauptsächlich an *PHP* als Hauptsprache, für die es unzählige Frameworks und Anwendungen¹ gibt. Auch wird oft *Ruby on Rails* im Zusammenhang gebracht, bei der es eine große Entwicklergemeinde existiert und ebensoviele bekannte Projekte².

So soll aufgezeigt werden, dass auch der Pythonentwickler seine hart erarbeiteten Kenntnisse auch im Webumfeld einsetzen kann und keine neue Programmiersprache dafür lernen muss. Mit Django, Zope, Pylons, web2py gibt es genug Alternativen für die Entwicklungen von Anwendungen in der Python-Welt. Diese werden anhand einer Beispielanwendung vorgestellt und untersucht.

1.2 Durchführung

1.2.1 GUI

Da es sich um eine Webanwendung handelt, kommen für die Interaktion des Benutzers mit der Schnittstelle XHTML³ für die Struktur und den Inhalt, CSS⁴ für die Gestaltung und JavaScript auch in Form eines Frameworks, abhängig davon, was das Webframework anbietet, in Frage. Beispiele für JavaScript-Frameworks⁵ sind z.B. jQuery, Dojo oder Prototype mit denen eine Vereinfachte Interaktion mit dem Benutzer erstellt werden kann. Da diese Frameworks die Möglichkeit bieten AJAX⁶ Funktionalität anzuwenden, kann eine Ressourcen schonende Applikation mit schnellen Reaktionszeiten erstellt werden, da nicht alle Teile der Anwendung geladen werden müssen.

Auf folgender Vorlage^{1.1}, die mittels HTML und CSS erstellt worden ist, basiert die GUI der Anwendung. Es wurde durchgehend darauf geachtet, dass *em* als Maßeinheit

¹magentocommerce.com, atmail.com

²unter anderen twitter.com, shopify.com, github.com

³XHTML: Extensible HyperText Markup Language, <http://www.w3.org/TR/xhtml1/>

⁴Cascading Stylesheets Level 2, <http://www.edition-w3.de/TR/1998/REC-CSS2-19980512/>

⁵JavaScript-Frameworks: <http://webstandard.kulando.de/post/2008/08/21/top-10-aller-javascript-frameworks>

⁶AJAX: Asynchronous Javascript and XML, http://openbook.galileocomputing.de/javascript-ajax/18_ajax_001.htm

verwendet wird, um ein elastisches⁷ Layout sicherzustellen. Der Vorteil liegt hier in der besseren Zugänglichkeit der Seite und der Anpassung des Designs an die Schriftgröße des Browsers, welche der User benutzt. So bleiben Skalierungen proportional und es entstehen weniger negative Seiteneffekte beim Zoomen der Seite.

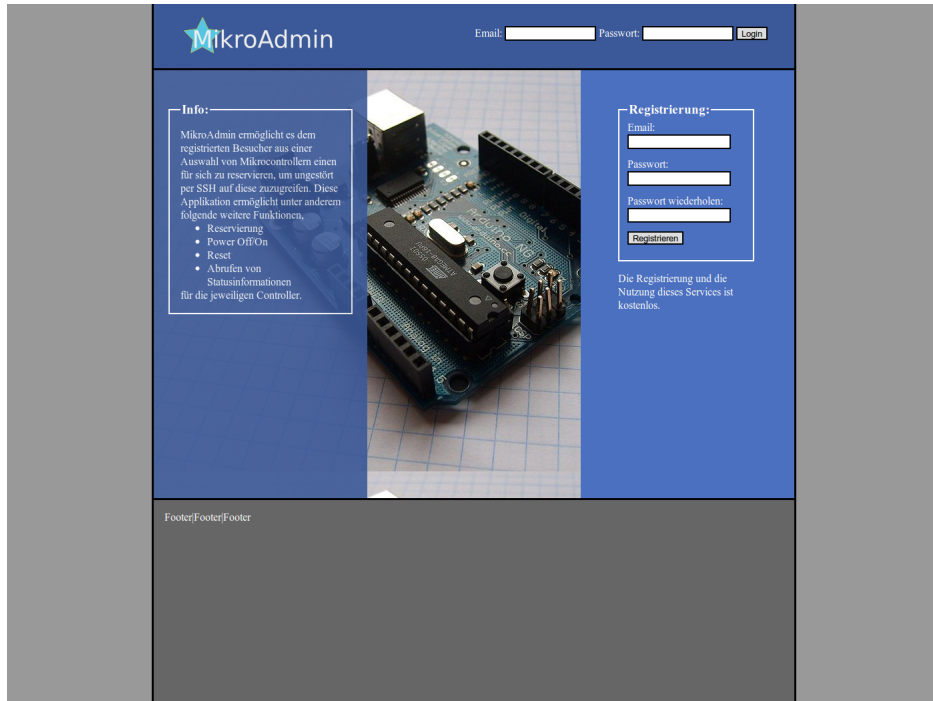


Abbildung 1.1: GUI der Anwendung

Darauf aufbauend, wird das Template in den jeweiligen Frameworks angepasst und weitere Unterseiten von dieser Seite so weit es geht abgeleitet. So wird es einfacher die jeweiligen Möglichkeiten der Template Sprachen der verschiedenen Frameworks zu vergleichen. Der Code des Templates kann unter folgender Adresse eingesehen werden.

1.2.2 Datenbank

1.2.3 Logik

⁷<http://greattalk.ch/2008/02/09/die-unglaublichen-em-elastischen-layouts-mit-css/>

2 Webframeworks

2.1 Definition

Webframeworks sind im Grunde eine Sammlung von Funktionalität, die es dem Webentwickler erlaubt Anwendungen zu schreiben. Die Funktionalität besteht aus verschiedenen Libraries, die den Zugriff auf Datenbanken, die Erstellung von Templates für die Ausgabe in HTML oder die Erstellung von Klassen und Methoden für die Geschäftslogik.

Webanwendungen, die sich auf Datenbanken stützen können mit Webframeworks relativ schnell erstellt werden, dabei helfen OR-Mapper¹, die aus Klassenkonstrukte SQL² für eine ausgewählte Datenbank erstellen. Die Mapper bieten darüberhinaus eine eigene Sprache, um auf die Klassen bzw. Informationen in der Datenbank zuzugreifen. Der Vorteil liegt darin, dass der Code unabhängig von einer Datenbank erstellt werden kann. Die Geschäftslogik kann abhängig vom Framework in einer beliebigen Programmiersprache geschrieben werden, meistens Java, PHP oder Python. Um die Ausgabe zu generieren bringen die Frameworks eine Templatesprache mit. Diese ermöglicht die einfache Erstellung von HTML aufgrund der Mischung von HTML und Templatecode. Dieser Code dient überwiegend zur Darstellung von Information, die aus der Geschäftslogik stammt.

2.2 Warum Python

Die Programmiersprache Python ist eine sehr flexible Programmiersprache. Sie kann in verschiedenen Gebieten eingesetzt werden. Unter anderem bei der Verarbeitung von XML, zur Verbindung von Datenbanken, in der Netzwerk Programmierung, im Webbereich auch die Erstellung von Programmen mit grafischer Oberfläche (wxPython, PyQt) ist möglich. In dieser Arbeit wird vor allem der Webbereich genauer analysiert und verschiedene Webframeworks am Beispiel einer Anwendung verglichen.

Python ist nicht an ein bestimmtes Programmierparadigma gebunden und je nach

¹Objekt Relationale Mapper

²Structured Query Language

Aufgabe kann ein Paradigma, wie z.B. Objektorientierung, Funktionaler Programmierung oder ein aspektorientierter Ansatz gewählt werden.

Im Jahre 1991 wurde die Sprache veröffentlicht. Sie wurde im *Centrum voor Wiskunde en Informatica* in Amsterdam entwickelt. Dabei gilt *Guido van Rossum* als Erfinder der Sprache. Python wurde unter anderem von C, LISP und Smalltalk beeinflusst. Wobei Python selbst wiederum Ruby, Boo oder Groovy beeinflusst hat.

Ja, Python ist gewachsen und hat sich zu einer ausgereiften Sprache samt eigenem Ökosystem entwickelt.[1]

Im Vergleich zu z.B. PHP[3] gibt es zwar auch viele Python-Webframeworks[4], diese sind aber weniger bekannt, mit der Ausnahme von z.B. Django oder TurboGears. Es werden diese zwei beim Vergleich eingesetzt, zusätzlich auch neuere oder weniger bekannte Frameworks im Pythonökosystem.

2.3 Fullstack vs. Glue

Frameworks werden hauptsächlich unterschieden zwischen Fullstack- und Glueframeworks.

Ein Fullstackframework bringt alle möglichen Komponenten mit, um eine Webanwendung schreiben zu können. Unter anderem sind das ein OR-Mapper für die Datenbank, eine eigene Templatesprache oder ein URL-Dispatcher³ für lesbare Links. Der Vorteil eines solchen Frameworks ist, dass eine klare Linie vorgegeben wird, um eine Anwendung schreiben zu können. Dies führt in der Regel zu schnellen Ergebnissen. Vertreter dieser Art sind z.B. Django.

Auf der anderen Seite gibt es Frameworks, die aus einzelnen unabhängigen Komponenten bestehen. Dabei sind diese oft lose gekoppelt und können leicht ausgetauscht werden. So können für diese Frameworks immer die neuesten Komponenten benutzt werden, die von Gluecode⁴ zusammengehalten werden. Es ist aber ein größerer Aufwand nötig, um diese losen Teile zusammenzubringen. Das bekannteste Framework dieser Art in Python ist TurboGears.

³<http://www.w3.org/Provider/Style/URI>

⁴http://en.wikipedia.org/wiki/Glue_code

2.4 Weitere Unterscheidungsmerkmale

2.4.1 Funktionalität

Wenn ein neues Produkt das Interesse geweckt hat, liegt es meistens an der Funktionalität, die es anbietet. Für eine Webapplikation ist es wichtig, dass auf eine Art und Weise der Client auf die Dateien mittels Browser zugreifen kann. Als Schnittstelle bietet sich im Pythonumfeld WSGI⁵ an, damit der Webserver mit der Webapplikation kommunizieren kann, die wiederum HTML an den Client liefert. Für einfache Anwendungen reicht es aus HTML-Code mit Hilfe von Python zu generieren und diesem dem User zur Verfügung zu stellen. Soll die Anwendung aber komplexere Ausmaße haben und zum Beispiel eine Userverwaltung mit LogIn Möglichkeit bieten ist es ratsam ein Framework auszuwählen, welches diese Funktionalität mit einfachen Handgriffen bietet. Auch die Kommunikation zu Datenbanken ist ein wichtiger Bestandteil, da viele Webapplikationen Klassen bereitstellen, mittels derer Objekte generiert werden, die miteinander interagieren. Daraus entstehen meistens Informationen die persistent gehalten werden müssen. Beispiele sind Webshops, Blogs/Wikis oder soziale Netzwerke. Bei der Entwicklung sollte auch berücksichtigt werden, dass Webdesigner leicht im Entwicklungsprozess integriert werden können. Dabei sollte wichtig sein, dass CSS und JavaScript (oder JavaScript-Frameworks), die als Standard bei der Gestaltung gelten eingesetzt werden können. Dabei spielen Templatesprachen für die Erstellung des HTML-Codes eine große Rolle, denn viele Webframeworks bieten Templatesprachen, die sich an Pythoncode anlehnen aber nur geringe logische Funktionalität bieten und im Grunde nur für die Darstellung der einzelnen Seiten konzipiert sind.

2.4.2 Dokumentation

Viele Projekte und im Speziellen hier Webframeworks werden hauptsächlich wegen ihrer Funktionalität ausgewählt. Ein weiterer Grund kann auch ihre Popularität sein auf Grund eines Hypes. Ein sehr wichtiger Aspekt, um überhaupt mit einem Framework oder Programmiersprache anzufangen, ist die Lernhürde die manchmal hoch sein kann. Deswegen ist es wichtig und sollte eine Priorität der Entwickler sein die Frameworks sehr gut zu dokumentieren und Anfängertutorials zu anbieten.

⁵<http://wsgi.org/wsgi/>

2.5 Bekannte Webframeworks im Pythonumfeld

2.5.1 Überblick

Zwar werden Webapplikationen sehr oft mit Hilfe von PHP oder Java geschrieben und mit Frameworks, die sich schon länger etabliert haben. Doch unter Python ist Webentwicklung leicht möglich und für Pythonentwickler optimal, da keine neue Sprache dafür gelernt werden muss. Die folgend beschriebenen Frameworks⁶ sind nur eine kleine Auswahl und sind sicher die bekanntesten. Da eine große Auswahl herrscht, findet sich für jede Projektgröße das passende Framework.

2.5.2 Django

Django ist ein High-Level Python Webframework, das eine schnelle Entwicklung fördert und ein sauberes und pragmatisches Design verfolgt.

Django wurde designt, um zwei Herausforderungen zu meistern: Intensiver Deadlines eines Newsrooms und den strengen Anforderungen der erfahrenen Webentwickler, die das Framework geschrieben haben, gerecht zu werden. Django ermöglicht es performante und elegante Anwendungen, schnell zu entwickeln. [6]

2.5.3 Pylons

Pylons verbindet Ideen aus der Welt von Ruby, Python und Perl. Dabei ist es ein sehr flexibles Python-Webframework und verhalf dem WSGI-Standard zum Durchbruch. Das Ziel von Pylons ist es, die Webentwicklung schnell, flexibel und einfach zu gestalten. [5]

2.5.4 Zope

Die Zope Softwarebibliothek ermöglicht die komponentenbasierte Entwicklung von Web-Anwendungen in der objektorientierten Programmiersprache Python. In der komponentenbasierten Programmierung werden komplexe Anwendungen mit Hilfe wiederverwendbarer Komponenten erstellt. Eine Komponente stellt dabei die Implementierung einer bestimmten genau spezifizierten Funktionalität dar. [7]

⁶<http://i.imgur.com/SwhUS.jpg>

2.5.5 web2py

web2py wurde inspiriert von Ruby on Rails und, wie Rails, liegt der Focus in schneller Webentwicklung und folgt dem Model View Controller Muster. web2py unterscheidet sich von Rails, weil es auf Python basiert. Weil es eine umfassende Web basierte Administrationsoberfläche bereitstellt, darüber hinaus beinhaltet es Bibliotheken, um mit mehr Protokollen zurechtzukommen und ist auf der Google App Engine lauffähig.[8]

2.6 Webframeworks anderer Programmiersprachen

2.6.1 Überblick

Zur Erstellung von Webapplikationen gibt es viele Möglichkeiten dies zu bewerkstelligen. Der einfachste Ansatz ist, eine Programmiersprache zu lernen, mit deren Hilfe HTML-Code erzeugt und dem Client zur Verfügung gestellt wird.

Der erste Kontakt wird sicherlich mit PHP gemacht. Leicht zu erlernen und mit einer Vielzahl von Funktionen. Darüber hinaus existieren eine Menge Frameworks, die den Entwickler bei der Arbeit unterstützen. Unter anderen Zend, CakePHP oder Symphony.

Andere Programmiersprachen, wie Ruby, Java oder Perl⁷ ermöglichen ebenfalls Programme fürs Web zu schreiben und einem breiten Publikum zur Verfügung zu stellen. Darunter bekannte wie Ruby on Rails, Spring oder Jifty. Zwar liegt der Focus auf Python, dennoch ist es wichtig andere Möglichkeiten der Webentwicklung für einen besseren Überblick kurz vorzustellen.

2.6.2 Zend

Das Zend Framework erweitert die Kunst und den Geist von PHP, dabei basiert es auf Einfachheit, beste praktische Vorgehensweise, freundliche Lizenzierung für Firmen und einer gründlich getesteten Codebasis. Dabei liegt der Focus des Zend Frameworks auf sicherere und moderne Web 2.0 Anwendungen und Webservices. [9]

2.6.3 Ruby on Rails

Ruby on Rails ist eine OpenSource Webframework für die Zufriedenstellung der Webentwickler und kontinuierliche Unterstützung der Produktivität. Damit lässt es sich wun-

⁷http://www.socialtext.net/perl5/index.cgi?web_frameworks

derbaren code schreiben wobei das Design Paradigma Convention over Configuration⁸ angewandt wird.[10]

„Ruby on Rails is a breakthrough in lowering the barriers of entry to programming. Powerful web applications that formerly might have taken weeks or months to develop can be produced in a matter of days.“⁹

2.6.4 Catalyst

Das Catalyst ist ein in Perl geschriebenes MVC¹⁰ Webframework. Dieses Framework kann auf verschiedenen Plattformen installiert werden und ist für schnelle Webentwicklung geeignet.[11]

Bei der Entwicklung wird bedacht auch KISS¹¹ gelegt, was dazu führt, dass Catalyst skalierbar und robust bleibt. Was sich bei den daraus entwickelten Applikationen bemerkbar macht.[12]

2.6.5 Wicket

Mit zweckmäßiger Aufteilung von Darstellung und Logik, ein POJO¹² Datenmodell und erfrischendem Verzicht von XML, macht Apache Wicket das Entwickeln von Webapplikationen wieder einfach und angenehm. Es wird mit unnötigem Code, komplexem Debugging und brüchigem Code gebrochen, um wiederverwendbare Komponenten, die nur auf Java und HTML aufbauen zu erstellen.[13]

2.7 Arbeiten ohne Framework

Die Erstellung einer Anwendung ohne ein entsprechendes Framework ist natürlich möglich doch oft sehr mühsam. Der Entwickler hat alle Freiheiten eine Anwendung zu schreiben, ihm wird aber mit einem guten Framework vieles erleichtert. Webframeworks bringen Unterstützung in Aspekten, wie z.B. Sicherheit, Benutzerverwaltung und Applikationsadministration. Diese müssen, falls benötigt von Grund auf immer wieder neu

⁸Convention over Configuration: Dabei ist das Design Paradigma gemeint, bei dem der Entwickler Regeln folgen muss um z.B. dass Primärschlüssel in der Datenbank Integer sein müssen oder Klassennamen und Tabellennamen gleich lauten müssen

⁹Tim O'Reilly, Founder of O'Reilly Media: <http://rubyonrails.org/quotes>

¹⁰MVC: Model View Controller Design Muster <http://www.oracle.com/technetwork/java/mvc-140477.html>

¹¹KISS: Keep it simple stupid <http://techcrunch.com/2009/04/28/keep-it-simple-stupid/>

¹²POJO: Plain Old Java Object <http://www.martinfowler.com/bliki/POJO.html>

geschriben werden und lenken somit von der eigentlichen Entwicklung der Kernapplikation ab.

Die meisten Frameworks arbeiten nach dem Prinzip „Don’t repeat your self“ oder „KISS“ keep it simple stupid. Somit werden immer wiederkehrende Aufgaben, die mit komplexen Projekten verbunden sind bereits vorgefertigt geliefert und der Entwickler kann sich sofort der eigentlichen Kernaufgabe widmen ohne sich um nötigen Boilerplate kümmern zu müssen.

3 Django

3.1 Einführung

3.2 Installation

3.3 Dokumentation

3.4 Datenbanksysteme

3.5 Templatesystem

3.6 JavaScript

3.7 Sicherheitsaspekte

3.8 Durchführung der Anwendung

4 TurboGears

4.1

5 Pylons

5.1 Einführung

6 Zope

6.1 Einführung

7 Web2Py

7.1 Einführung

8 Ergebnisse / Übersicht

8.1 Bewertungskriterien

- Dokumentation
- Installation
- Templatesystem / Darstellung
- Unterstützung von Javascript
- Unterstützung von CSS
- Datenbankbindung
- Handling
- Lizenz

Literaturverzeichnis

- [1] Farid Hajji, Das Python Praxisbuch - Der große Profi-Leitfaden für Programmierer, 2008
- [2] Python Beschreibung, <http://www.python.org/about/>, 28.07.2010
- [3] phpwact.org, http://www.phpwact.org/php/mvc_frameworks, 28.07.2010
- [4] Python Webframeworks Übersicht , <http://wiki.python.org/moin/WebFrameworks>, 28.07.2010
- [5] Pylons, <http://pylonshq.com/>, 29.07.2010
- [6] Django, <http://www.djangoproject.com/>, 29.07.2010
- [7] zope, <http://www.zope.de/uber-zope>, 29.07.2010
- [8] web2py, <http://www.web2py.com/examples/default/what>, 29.07.2010
- [9] zend, <http://framework.zend.com/about/overview>, 30.07.2010
- [10] Ruby on Rails, <http://rubyonrails.org/>, 30.07.2010
- [11] perl.net.au, <http://perl.net.au/wiki/Catalyst>, 30.07.2010
- [12] Catalystframework, catalystframework.org/, 30.07.2010
- [13] Wicket Webframework, catalystframework.org/, 30.07.2010