

# Moderne GIS-Programmierung mit Python und Quantum GIS

---



## Bachelorarbeit Frühlingssemester 2010

Autor: Simon Keller (s1keller@hsr.ch)

Betreuer: Prof. Stefan Keller (sfkeller@hsr.ch)

HSR Hochschule für Technik Rapperswil  
Oberseestrasse 10  
Postfach 1475  
CH-8640 Rapperswil

## I. Abstract

|   |   |
|---|---|
| <b>Abteilung</b>  | Informatik  |
| <b>Name des Studenten</b>   | Simon Keller  |
| <b>Semester</b>   | FS 2010   |
| <b>Titel der Studienarbeit</b>  | Moderne GIS-Programmierung mit Python und Quantum GIS |
| <b>Examinator</b>   | Prof. Stefan Keller                                   |
| <p>Open Source Software erlangt im Bereich der Geographischen Informationssysteme (GIS) eine immer interessantere Rolle. Das freie Quantum GIS ist ein benutzerfreundliches plattformübergreifendes GIS, welches zu den verbreiteten kommerziellen Geoinformationssystemen wie ArcGIS aufgeschlossen hat.</p> <p>Das Plugin NPLCH soll zeigen wie viel Potential in der Erweiterbarkeit von QGIS mit Python steckt. Es ist ein Prototyp zur Benutzung des Entwurfs der Norm „SIA 424 – Rahmennutzungspläne“. Die Applikation übernimmt die Erstellung eines Projektes, Bearbeitung, topologische Validierung bis hin zum Export nach OEREB. Die gute Integration in QGIS erleichtert die Benutzung der weiteren QGIS-Spezifischen Anwendungen und Optionen. Eine automatische Konfiguration übernimmt dem Benutzer viel Arbeit, die ganzen Einstellungen selbst zu konfigurieren.</p> |   |

## II. Management Summary

### *Ausgangslage*

Python ist eine vielseitige Programmiersprache mit hoher Programmlesbarkeit. Dadurch löst Sie immer mehr die alten Skriptsprachen wie Visual Basic for Application (VBA) ab. So wird Python beispielsweise in OpenOffice und im verbreiteten kommerziellen Geoinformationssystem (GIS) ArcGIS eingesetzt. Quantum GIS (QGIS) ist ein plattformübergreifendes Open Source GIS, das sich mit Python und Qt-Bibliothek zur Programmierung grafischer Benutzeroberflächen (GUI) erweitern lässt. Es wird angenommen, dass in Zukunft immer mehr GIS-Anwender Python als Skript-Sprache wählen.

### *Vorgehen/Technologien*

Zuerst erfolgt der Einstieg in Python und das Ausarbeiten von Kursübungen, zur Einführung in Python, Qt und PyQGIS-Plugins für GIS-Anwender. Danach soll eine Applikation (PyQGIS-Plugin) erstellt werden, welches die „Zonen-/Nutzungsplanung“ in QGIS unterstützt und das Potential der Erweiterbarkeit von QGIS aufzeigt. Das Ganze soll sich auf den Entwurf der SIA Norm 424, „Rahmennutzungspläne“ und dem Datenmodell darin beziehen.

## Ergebnis

Das aufbereitete Datenmodell wird mit den Zonentypendaten in eine PostgreSQL-Datenbank mit der PostGIS-Erweiterung importiert. Diese Erweiterung erlaubt eine effiziente Verwaltung von Geometriedaten. Die Zonendaten werden zusammen mit den Grundlagedaten der Amtlichen Vermessung in QGIS angezeigt. Nach der Erzeugung oder der Bearbeitung der verschiedenen Zonen und Festlegungen kann das Ganze auf Ihre topologische Korrektheit überprüft werden. Ein weiterer Punkt bei der Bearbeitung ist die mehrfache Beschriftung, die von QGIS nicht direkt unterstützt wird. Vor dem Export werden die Daten mit externen Werkzeugen nochmals validiert und können danach nach dem neuen Kataster der öffentlich-rechtlichen Eigentumsbeschränkungen exportiert werden.

Weitere Informationen: <http://dev.ifs.hsr.ch/python4gis>

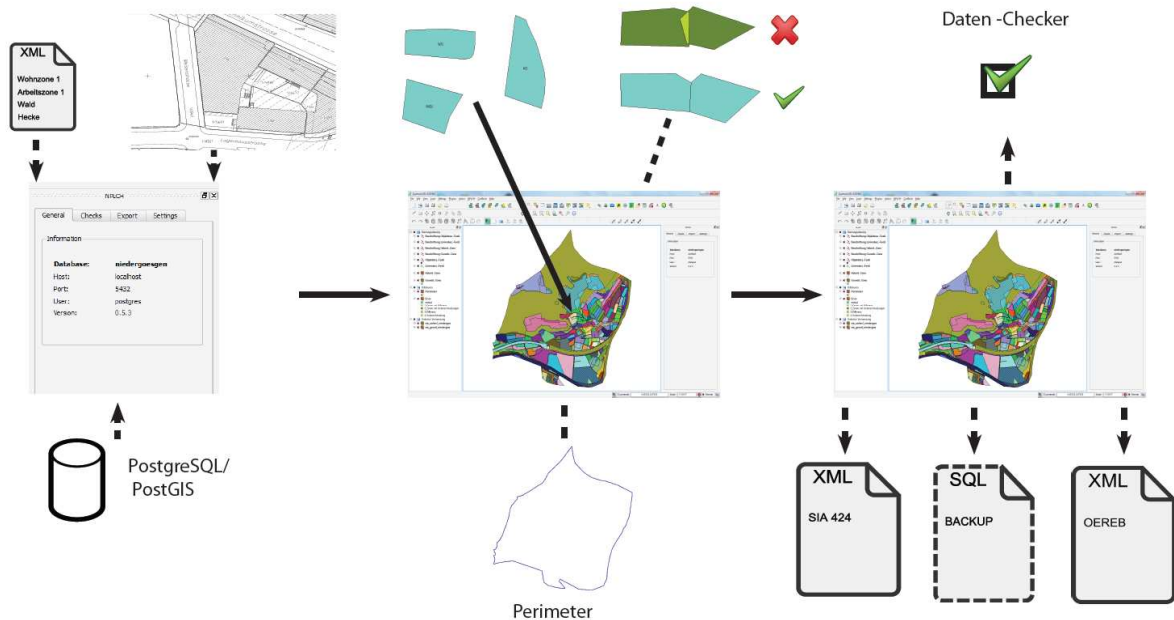


Abbildung 1: Ablauf eines NPLCH-Projektes.

## Inhaltsverzeichnis

|                                   |   |    |
|-----------------------------------|---|----|
| I.                                | Abstract .....  | 2  |
| II.                               | Management Summary.....   | 2  |
| III.                              | Aufgabenstellung.....   | 7  |
| TEIL I: TECHNISCHER BERICHT ..... |   | 9  |
| 1                                 | Einführung .....  | 9  |
| 1.1                               | Überblick .....   | 9  |
| 1.2                               | Problemstellung und Vision .....                                      | 9  |
| 1.3                               | Ziele .....   | 9  |
| 1.4                               | Rahmenbedingungen.....  | 10 |
| 1.5                               | Stand der Technik.....  | 10 |
| 1.6                               | Quantum GIS .....   | 11 |
| 2                                 | Kurs „Einführung in die GIS-Programmierung mit Python und QGIS“ ..... | 12 |
| 2.1                               | Ziel .....  | 12 |
| 2.2                               | Überblick .....   | 12 |
| 2.3                               | Ablauf .....  | 12 |
| 2.4                               | Resultate.....  | 12 |
| 2.5                               | Eigenes kleines Plugin (SimpleReloaderPlugin).....                    | 13 |
| 2.6                               | Tipps und Tricks zu QGIS .....  | 13 |
| 3                                 | Evaluation.....   | 14 |
| 3.1                               | Entwicklungsumgebungen .....  | 14 |
| 3.2                               | PyQGIS-Plugins .....  | 17 |
| 4                                 | Erweiterung der Swissphone-Applikation .....                          | 17 |
| 4.1                               | Ziel .....  | 17 |
| 4.2                               | Vorgehen .....  | 17 |
| 4.3                               | Vergleich von Teilbereichen zu NPLCH.....                             | 18 |
| 4.4                               | Aufbau .....  | 18 |
| 4.5                               | Idee.....   | 19 |
| 4.6                               | Resultat.....   | 20 |
| 5                                 | Resultate.....  | 21 |
| 5.1                               | Bewertung .....   | 21 |
| 5.2                               | Zielerreichung.....   | 21 |
| 5.3                               | Schlussfolgerung und Ausblick .....                                   | 21 |
| 6                                 | Persönlicher Bericht .....  | 22 |



|  |    |
|--|----|
| TEIL II: SW-PROJEKTDOKUMENTATION NPLCH ..... | 23 |
| 7 Anforderungsspezifikation .....            | 23 |
| 7.1 Anforderungen an die Bachelorarbeit..... | 23 |
| 7.2 Allgemeine Beschreibung.....             | 23 |
| 7.3 Use Cases.....                           | 25 |
| 7.4 Spezifische Anforderungen .....          | 30 |
| 8 Analyse .....                              | 31 |
| 9 Design & Implementation .....              | 32 |
| 9.1 Architektonische Darstellung .....       | 32 |
| 9.2 logische Architektur.....                | 33 |
| 9.3 Design Pakete .....                      | 33 |
| 9.4 Datenspeicherung .....                   | 40 |
| 10 Tests.....                                | 42 |
| 10.1 Technologie .....                       | 42 |
| 10.2 Automatische Tests .....                | 42 |
| 10.3 Manuelle Tests – Use Case Tests.....    | 42 |
| 10.4 Nicht berücksichtigte Punkte .....      | 42 |
| 10.5 Testfazit .....                         | 42 |
| 11 Ausblick.....                             | 43 |
| 11.1 Weiterentwicklung .....                 | 43 |
| 12 Projektmanagement.....                    | 44 |
| 12.1 Projektorganisation .....               | 44 |
| 12.2 Management Abläufe.....                 | 44 |
| 12.3 Risiko Management.....                  | 46 |
| 12.4 Arbeitspakete .....                     | 46 |
| 12.5 Infrastruktur .....                     | 46 |
| 12.6 Sitzungs-Protokolle.....                | 47 |
| 12.7 Projektmonitoring .....                 | 48 |
| 13 Softwaredokumentation .....               | 49 |
| 13.1 Voraussetzungen .....                   | 49 |
| 13.2 Installation.....                       | 49 |
| 13.3 Konfiguration.....                      | 53 |
| 13.4 Anwendung .....                         | 53 |
| TEIL III: ANHANG .....                       | 56 |
| 14 Inhalt der CD.....                        | 56 |

|    |   |    |
|----|---|----|
| 15 | Glossar und Abkürzungsverzeichnis ..... | 57 |
| 16 | Literaturverzeichnis.....               | 58 |
| 17 | Kursübungen .....                       | 60 |
| 18 | Sitzungsprotokolle.....                 | 61 |
| 19 | Eigenständigkeitserklärung.....         | 62 |

### III. Aufgabenstellung

Der nachfolgende Text entspricht der original Aufgabenstellung, wie sie von Prof. Stefan Keller formuliert wurde.

#### **Bachelorarbeit FS 2010**

##### **Moderne GIS-Programmierung mit Python und Quantum GIS**

**Autoren/Studenten:** Simon Keller, HSR, Abteilung Informatik

**Verantwortlicher/  
Betreuer:** Prof. Stefan Keller, HSR, Abt. Informatik

**Partner (Firma oder Verwaltung),  
externer Betreuer:** Die „Open Source Community“

**Gegenleser:** Prof. Hansjörg Huser

##### **Einführung**

Python ist eine vielseitige Programmiersprache, die das Potential hat, Visual Basic for Applications (VBA) als Skriptingsprache abzulösen. So wird Python beispielsweise in OpenOffice, in GIMP und im verbreiteten kommerziellen Geoinformationssystem (GIS) ArcGIS eingesetzt. Quantum GIS (kurz: QGIS) ist ein benutzerfreundliches plattform-übergreifendes Open Source GIS, das sich u.a. mit Python erweitern lässt und zwar als Plugin (PyQGIS) oder als stand-alone Applikation - beides mit Hilfe der Qt-Bibliothek zur Programmierung grafischer Benutzeroberflächen (GUI).

Es wird angenommen, dass in Zukunft immer mehr GIS-Anwender Python als Skripting-Sprache wählen. Diesen soll mit einem Einführungskurs der Einstieg in Python vereinfacht werden. Um das Potential der Erweiterbarkeit von QGIS zu zeigen, soll schwerpunktmässig eine Applikation zur Verwaltung eines "Zonen-, bzw. Bodennutzungsplans" entwickelt werden.

##### **Aufgabenstellung**

Zum einen geht es um das teilweise Realisieren von Übungen zu einem Kurs "Einführung in die GIS-Programmierung mit Python und QGIS" für Programmierneinsteiger. Zur Anforderungsanalyse und zum Testen können Assistenten/innen der HSR-Institute ILF und IRAP beigezogen werden. Zum Zweiten soll das aktuelle QGIS-Python-Plugin-Tutorial (englisch, basierend QGIS Version 1) aktualisiert werden.

Im dritten Teil sollen Entwicklungsumgebungen (IDEs) sowie vorhandene PyQGIS-Plugins evaluiert werden. Wenn sinnvoll, soll ein eigenes Plugin realisiert werden, das produktiv eingesetzt werden kann. Schliesslich geht es 4. um eine Entwicklung einer Applikation zur Verwaltung eines "Zonen-, bzw. Bodennutzungsplans" (Fachschale umfassend Datenstruktur mit Erfassungs- und Verwaltungs- und Datenexport-Funktionen). Vorgängig soll eine Funkantennen-Applikation der Fa. Swissphone untersucht werden.

Lieferdokumente: (englisch wo angegeben, sonst deutsch)

- Dokumentierte, produktive Applikation inkl. Projekt-Setup (README, Start Scripts).
- Gesamter compilierbarer Sourcecode und Bytecode mit Continuous Build Scripts und automatisierte Unit Tests (nach Möglichkeit auch GUI).
- Vollständiger technischer Bericht und Software Engineering-Dokumentation der Applikation (bekannt aus SE).
- Ggf. eigenes "QGIS-Python-Plugin" als Plugin-Repository publiziert und auf bekannten Webseiten publiziert.
- Teilweise ausgearbeitete Unterlagen (Übungen) zu einem Python-Kurs.
- "QGIS-Python-Plugin-Tutorial" mit Unterlagen und Übungen (ggf. auf externer, originaler Website).

### **Hinweise**

- Die Arbeitsweise ist iterativ, agil und test-basiert.
- Für die erfolgreich abgeschlossene Arbeit werden 12 ECTS angerechnet. Dies entspricht einer Arbeitsleistung von 360 Stunden.

### **Randbedingungen, Infrastruktur, Termine und Beurteilung**

- Randbedingungen Hardware:
  - Server: Standard-Desktop-PC, virtueller Server
  - Client: Desktop-Browser
- Software
  - Python
  - Vorhandene IDEs
- Termine: gemäss Angaben auf [www.hsr.ch](http://www.hsr.ch).
- Beurteilung: gemäss Angaben auf [www.hsr.ch](http://www.hsr.ch).

Rapperswil, 8. März 2010

Der Betreuer

Der Studierende

Prof. Stefan Keller

Simon Keller

# TEIL I: TECHNISCHER BERICHT

## 1 Einführung

### 1.1 Überblick

Die nachfolgende Tabelle gibt eine Übersicht über den Aufbau dieser Dokumentation:

| Kapitel                                | Beschreibung   |
|--|--|
| TEIL I: Technischer Bericht            | Der erste Teil enthält den ganzen technischen Bericht, welcher eine Gesamtübersicht über die Arbeit gibt. Alle wichtigen Informationen sind darin enthalten, von der Aufgabenstellung bis zu den Resultaten der Arbeit.      |
| TEIL II: SW-Projektdokumentation NPLCH | Dieser Teil richtet sich an Softwareentwickler, welche Einblick in den Aufbau der Applikation gewinnen oder sie erweitern möchten. Nachfolgend werden noch die Bereiche Projektmanagement und Projektmonitoring beschrieben. |
| TEIL III: Anhang                       | Der letzte Teil bildet den Anhang in welchem sich das Glossar und das Literaturverzeichnis befinden. Ebenfalls im Anhang befinden sich die Kursübungen.  |

### 1.2 Problemstellung und Vision

Open Source Software wie Quantum GIS erlangen im Bereich der Geographischen Informationssysteme (GIS) eine ernst zunehmende Konkurrenz zu kommerziellen Geoinformationssystemen wie ArcGIS. Durch die Erweiterbarkeit mit Hilfe von Python und Qt können Sie recht einfach und schnell an die eigenen Bedürfnissen angepasst werden, ohne sich grosse Gedanken zu machen.

### 1.3 Ziele

Die Ziele sind gemäss Aufgabenstellung in vier Bereiche unterteilt.

- 1.) Das Realisieren von Übungen zu einem Kurs "Einführung in die GIS-Programmierung mit Python und QGIS" für Programmierneinsteiger. Zur Anforderungsanalyse und zum Testen können Assistenten/innen der HSR-Institute ILF und IRAP beigezogen werden.
- 2.) Zum Zweiten soll das aktuelle QGIS-Python-Plugin-Tutorial aktualisiert werden.
- 3.) Im dritten Teil sollen Entwicklungsumgebungen (IDEs) sowie vorhandene PyQGIS-Plugins evaluiert werden. Wenn sinnvoll, soll ein eigenes Plugin realisiert werden, das produktiv eingesetzt werden kann.

- 4.) Schliesslich geht es 4. um eine Entwicklung einer Applikation zur Verwaltung eines "Zonen-, bzw. Bodennutzungsplans" (Fachschale umfassend Datenstruktur mit Erfassungs- und Verwaltungs- und Datenexport-Funktionen). Vorgängig soll eine Funkantennen-Applikation der Firma. Swissphone untersucht werden.

## 1.4 Rahmenbedingungen

### 1.4.1 Administrativ

| Wann                 | Was   |
|----------------------|---|
| 11.06.2010           | Abgabe der Kurzbeschreibung der Arbeit an den Betreuer.                                 |
| 16.06.2010           | Abgabe der genehmigten und korrigierten Kurzbeschreibung an das Studiengangsekretariat. |
| 18.06.2010 12.00 Uhr | Abgabe des Berichts an Betreuer und des A0 Plakates im Studiengangsekretariat 6.113.    |

### 1.4.2 Technologien

Folgende Technologien werden eingesetzt:

- Python
- Qt-Bibliothek

### 1.4.3 Entwicklungsumgebung

Als Betriebssystem wird Microsoft Windows verwendet.

- Eclipse 3.5.1 IDE
- Eclipse Plugins
  - PyDev 1.5.7
  - Subclipse 1.6
- Python 2.6

## 1.5 Stand der Technik

Es existieren für QGIS bereits einige Plugins welche mittels Python geschrieben wurden. Diese werden im Kapitel 3.2 evaluiert. Der grösste Teil der Plugins sind einfache Funktionen, welche zu QGIS hinzugefügt werden. Ihr Aufbau ist meist sehr simpel, wenig Beschrieben und nie getestet.

Grundelemente können zur Inspiration dienen, ansonsten sind Sie meist nicht sonderlich brauchbar als „Vorlage“.

## 1.6 Quantum GIS

Quantum GIS ist ein Open Source Geoinformationssystem lizenziert unter der GNU GENERAL PUBLIC License und ein offizielles Projekt der Open Source Geospatial Foundation (OSGeo). Das Projekt wurde im Februar 2002 von Gary Sherman initiiert, worauf im Mai 2002 die Programmierung begann. Am 19. Juli 2002 erschien das erste Release, das zunächst ausschliesslich PostGIS Layers unterstützte und nur teilweise funktionstüchtig war.<sup>1</sup> Die derzeitig aktuelle Version von Quantum GIS ist das Release 1.4.0 mit dem Namen „Enceladus“ vom 10 Januar 2010.<sup>2</sup>

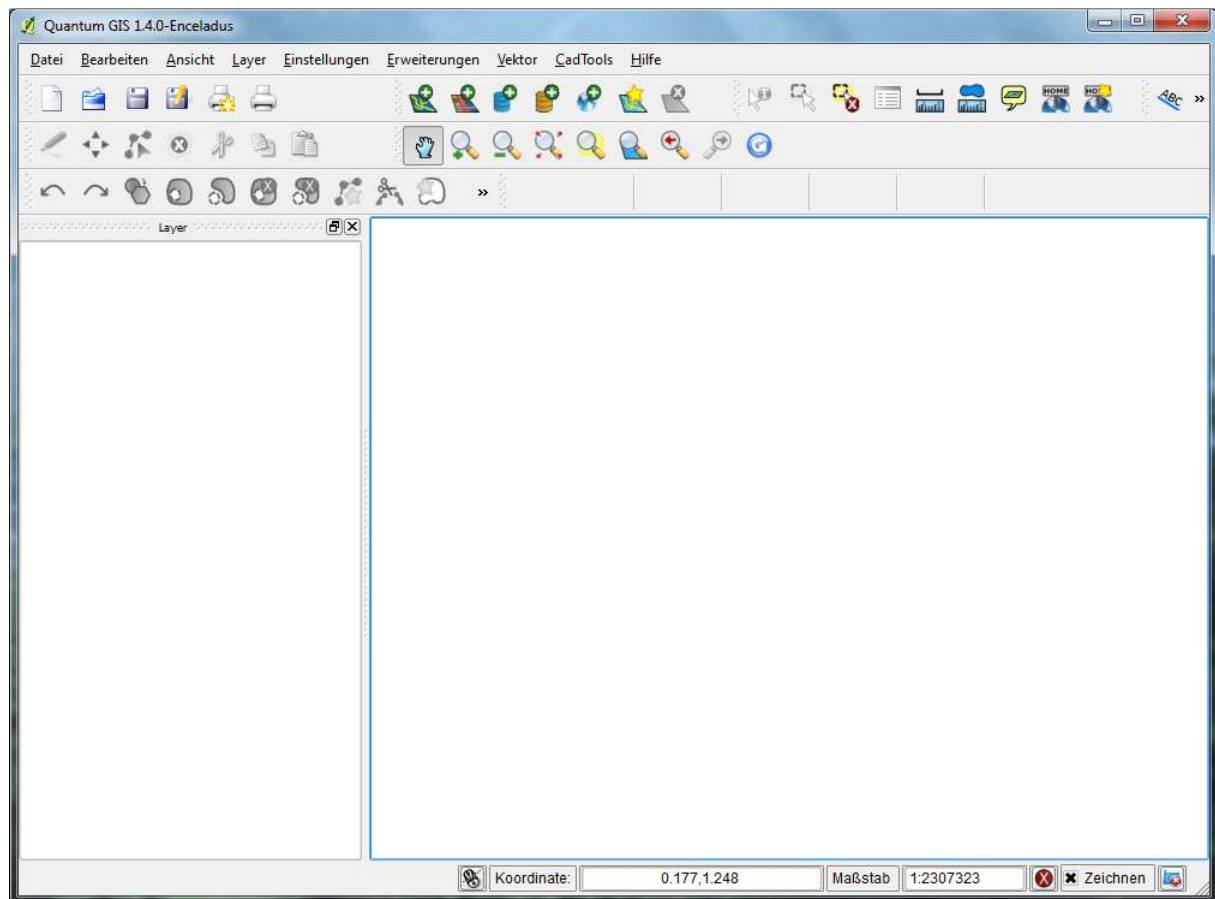


Abbildung 2: QGIS Release 1.4.0 „Enceladus“.

<sup>1</sup> <http://www.qgis.org/de/ueber-qgis.html>

<sup>2</sup> <http://blog.qgis.org>



## 2 Kurs „Einführung in die GIS-Programmierung mit Python und QGIS“

### 2.1 Ziel

Es sollen Übungen zu dem Kurs „Einführung in die GIS Programmierung mit Python und QGIS“ gestaltet werden, welcher als GISpunkt-Seminar gegeben wird.

### 2.2 Überblick

Ein Überblick über den Ablauf:

| Übung | Beschreibung   | Dauer  |
|-------|--|--------|
| 1     | IDE kennenlernen, kleine Übungen zu Ein- und Ausgabe und Variablen.  | 1 h    |
| 2     | Erstellen einfacher Programme unter Benutzung von Modulen (random) und kennenlernen von weiteren Datentypen. | 40 min |
| 3     | Erste Objektorientierte Programme.   | 40 min |
| 4     | Kennenlernen von Qt in Verbindung mit Python.  | 40 min |
| 5     | Verstehen des Aufbaus eines PyQGIS-Plugins sowie das vornehmen kleiner Veränderungen.                        | 40 min |

### 2.3 Ablauf

Nach dem groben Programm des Kurses wurden die Dauer und die Themen der einzelnen Übungen festgelegt. Danach wurde mit den Testbenutzern der Schwierigkeitsgrad der Übungen festgelegt. Anhand dieser wurden die einzelnen Kursübungen erstellt und anschliessend mit den Testbenutzern durchgeführt. Feedbacks der Testbenutzer flossen direkt in die Korrektur sowie in die nächsten Übungen ein.

### 2.4 Resultate

Die Kursübungen mitsamt den Lösungen liegen im Anhang.

Als Kursbuch ist das Buch „Python in a Nutshell“ (Martelli 2006) zu empfehlen. Für die meisten Benutzer würde aber für diese zwei Kurstage ein kleineres Buch zum nachschlagen genügen. Auf die Entwicklungsumgebung wird im folgenden Kapitel 3.1 eingegangen.

## 2.5 Eigenes kleines Plugin (SimpleReloaderPlugin)

### 2.5.1 Beschreibung

Mit Hilfe des QGIS-Python-Plugin-Tutorial sollte ein Plugin entwickelt werden, welches auch in den Kursübungen gebraucht werden kann. Eine kleine aber nützliche Funktion, ist das neu laden der Plugins nach Änderungen, um so einen Neustart von QGIS zu vermeiden. Es entstand ein kleines aber sehr praktisches Plugins welches bei der Entwicklung der NPLCH-Applikation zum Einsatz kam.

### 2.5.2 Anschauungen

Nachfolgend sind einige Screens des Plugins:



Abbildung 3: Icon des SimpleReloaderPlugins.

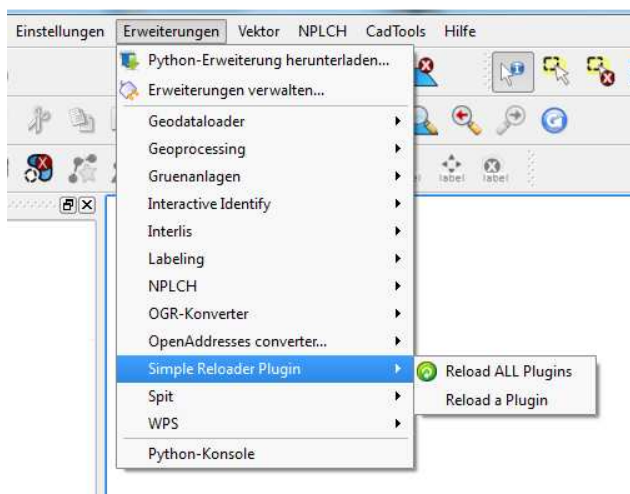


Abbildung 5: SimpleReloaderPlugin Menu.

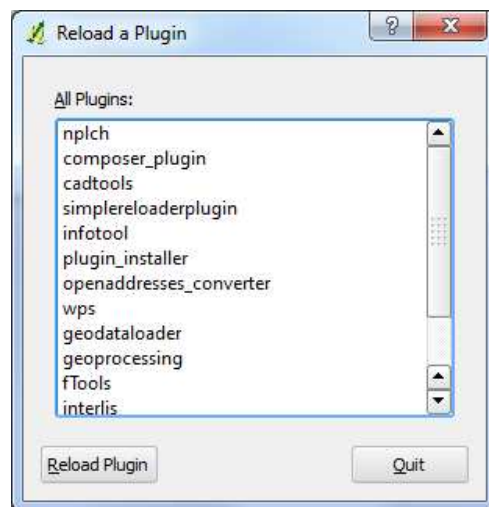


Abbildung 4: Fenster zum neu laden eines Plugins.

## 2.6 Tipps und Tricks zu QGIS

Die während dieser Arbeit gesammelten Informationen sind im Wiki unter [http://gis.hsr.ch/wiki/Quantum\\_GIS\\_-\\_Tipps\\_und\\_Tricks](http://gis.hsr.ch/wiki/Quantum_GIS_-_Tipps_und_Tricks) einsehbar.

## 3 Evaluation

### 3.1 Entwicklungsumgebungen

#### 3.1.1 Ziel

Das Ziel dieser Evaluation besteht darin, eine Python Entwicklungsumgebung zu finden, welche in dieser Arbeit und im Kurs verwendet werden kann. Es musste im Vorhinein nicht zwingend die Gleiche sein.

#### 3.1.2 Evaluation

Folgende Entwicklungsumgebungen wurden getestet:

| Entwicklungsumgebung | Homepage  |
|----------------------|---|
| Dr Python            | <a href="http://drpython.sourceforge.net/">http://drpython.sourceforge.net/</a>   |
| Eclipse/PyDev        | <a href="http://www.eclipse.org/">http://www.eclipse.org/</a> & <a href="http://pydev.org/">http://pydev.org/</a>                       |
| J                    | <a href="http://armedbear-j.sourceforge.net/">http://armedbear-j.sourceforge.net/</a>   |
| jHepWork             | <a href="http://jwork.org/jhepwork/">http://jwork.org/jhepwork/</a>   |
| Jasspa's MicroEmacs  | <a href="http://www.jasspa.com/">http://www.jasspa.com/</a>   |
| Jedit mit JpyDbg     | <a href="http://www.jedit.org/">http://www.jedit.org/</a> & <a href="http://jpydbg.sourceforge.net/">http://jpydbg.sourceforge.net/</a> |
| Komodo Edit          | <a href="http://www.activestate.com/komodo_edit/">http://www.activestate.com/komodo_edit/</a>   |
| Netbeans mit JpyDbg  | <a href="http://netbeans.org/">http://netbeans.org/</a> & <a href="http://jpydbg.sourceforge.net/">http://jpydbg.sourceforge.net/</a>   |
| UliPad               | <a href="http://code.google.com/p/ulipad/">http://code.google.com/p/ulipad/</a>   |
| Pye                  | <a href="http://code.google.com/p/pye-editor/">http://code.google.com/p/pye-editor/</a>   |
| PyPE                 | <a href="http://pype.sourceforge.net/index.shtml">http://pype.sourceforge.net/index.shtml</a>   |
| NotePad++            | <a href="http://notepad-plus.sourceforge.net/">http://notepad-plus.sourceforge.net/</a>   |
| PyScripter           | <a href="http://code.google.com/p/pyscripter/">http://code.google.com/p/pyscripter/</a>   |
| PyK                  | <a href="http://kib2.free.fr/PyK/">http://kib2.free.fr/PyK/</a>   |
| eric4                | <a href="http://eric-ide.python-projects.org/">http://eric-ide.python-projects.org/</a>   |

Table 1: Übersicht der evaluierten Entwicklungsumgebungen

| Name:               | Aktuell | Installation | Aufmachung | Debugging | Intuitiv | Code completion | Code Analysis | Ges: | Einsteiger | Platz | Programmierer | Platz |
|---------------------|---------|--------------|------------|-----------|----------|-----------------|---------------|------|------------|-------|---------------|-------|
| DrPython            | 6       | 5            | 9          | 5         | 8        | 0               | 0             | 33   | 293        | 10    | 202           | 7     |
| Eclipse/PyDev       | 10      | 7            | 9          | 9         | 6        | 10              | 10            | 61   | 505        | 1     | 420           | 1     |
| J                   | 6       | 1            | 0          | 0         | 0        | 0               | 0             | 7    | 58         | 15    | 47            | 15    |
| jHepWork            | 10      | 10           | 6          | 0         | 8        | 0               | 0             | 34   | 320        | 9     | 190           | 10    |
| Jasspa's MicroEmacs | 5       | 8            | 3          | 0         | 5        | 0               | 0             | 21   | 200        | 13    | 115           | 13    |
| Jedit mit JpyDbg    | 9       | 8            | 7          | 9         | 7        | 7               | 9             | 56   | 461        | 4     | 388           | 4     |
| Komodo Edit         | 9       | 9            | 9          | 0         | 7        | 6               | 0             | 40   | 382        | 6     | 218           | 6     |
| Netbeans mit JpyDbg | 9       | 6            | 9          | 9         | 8        | 9               | 9             | 59   | 491        | 3     | 403           | 3     |
| UliPad              | 8       | 9            | 6          | 0         | 7        | 5               | 0             | 35   | 334        | 7     | 191           | 9     |
| Pye                 | 7       | 10           | 4          | 0         | 6        | 0               | 0             | 27   | 256        | 12    | 149           | 12    |
| PyPE                | 9       | 10           | 9          | 0         | 8        | 5               | 0             | 41   | 392        | 5     | 223           | 5     |
| NotePad++           | 10      | 9            | 8          | 0         | 8        | 0               | 0             | 35   | 330        | 8     | 195           | 8     |
| PyScripter          | 7       | 7            | 0          | 0         | 0        | 0               | 0             | 14   | 126        | 14    | 84            | 14    |
| PyK                 | 5       | 10           | 6          | 0         | 7        | 0               | 0             | 28   | 270        | 11    | 150           | 11    |
| eric4               | 10      | 6            | 9          | 9         | 8        | 9               | 9             | 60   | 499        | 2     | 410           | 2     |
| Gewichtung:         |         |              |            |           |          |                 |               |      |            |       |               |       |
| Einsteiger          | 8       | 10           | 10         | 5         | 10       | 10              | 6             |      |            |       |               |       |
| Programmierer       | 7       | 5            | 5          | 10        | 5        | 5               | 10            |      |            |       |               |       |

### 3.1.3 Auswahl

Die in der Tabelle 1 (Kapitel 3.1.2) gelb markierten Einträge wurden alle als gut befunden und werden nun hier der Reihe nach detailliert aufgelistet.

#### *Eclipse*

Diese IDE ist vielseitig einsetzbar und unterstützt mehrere Programmiersprachen. Eclipse ist Open Source und aufgrund seiner offenen Plug-in-basierten Struktur sehr mächtig. Durch Features wie Code completion, Syntax highlighting, Code analysis, Go to definition, Refactoring und Debugging machen ihn für Programmierer zu einem beliebten Hilfsmittel.

#### *Eric4*

Eric4 wurde in PyQt geschrieben und verwendet für die grafische Benutzeroberfläche das Toolkit Qt. Seine Stärken liegen in der tiefen Zusammenarbeit mit Python. Beinhaltet auch Features wie Code completion und Debugging. Negativer Punkt ist das schwierige Installieren.

#### *Netbeans*

Netbeans als Entwicklungsumgebung ist stark mit Eclipse zu vergleichen. Beide befinden sich in etwa auf gleichem Niveau. Zur Unterstützung von Python braucht Netbean ein separates Plugin. Für diesen Zweck gibt es zwei Plugins. Das ältere JpyDbg und nbpython. Diese beinhalten schon einige Features sind aber noch nicht an PyDev dran.

#### *JEdit*

Eine Entwicklungsumgebung geschrieben in Java. Zur Unterstützung von Python wird das JpyDbg Plugin genutzt. Herauszuhebende Features sind interactive editing debugging, code browsing und highlighting.

#### *PyPE*

PyPE ist ein Editor geschrieben in Python. Seine Funktionen sind code folding, snippets, multiple documents, code completion, macros.

#### *Komodo Edit*

Komodo Edit ist eine freie, abgespeckte Version vom Komodo IDE welche auch Python unterstützt. Die gängigen Funktionen wie autocompleate and calltips, multi-language file support, syntax coloring and syntax checking sind dabei. Für manche Programmierer interessant könnten die weiteren Funktionen sein: Vi emulation, Emacs key bindings.

### 3.1.4 Schlussfolgerung

Da sich keine Entwicklungsumgebung hat absetzen können, kamen zur Auswahl immer mehr persönliche Aspekte hinzu. So z.B. mit welchen habe ich schon gearbeitet, welche finde ich intuitiver und weitere ähnliche. Zum Schluss habe ich mich auf Eclipse entschieden, da ich seine Funktionen schätzte und ich schon gute Erfahrungen damit gesammelt habe.

## 3.2 PyQGIS-Plugins

### 3.2.1 Ziel

Das Ziel dieser Evaluation besteht darin, die bestehenden Python-Plugins für QGIS zu testen. Das Ganze soll als Überblick für die Entwicklung dienen. Es sollen interessante Plugins gefunden werden, welche auch als Vorlage für bestimmte Bereiche genommen werden könnten.

### 3.2.2 Vorgehen

Zuerst wurden die zu untersuchenden Kriterien festgelegt:

| Kriterie             | Beschreibung  |
|----------------------|---|
| Minimum QGIS Version | Welches ist die minimale Version von QGIS in welchem das Plugin noch läuft.                     |
| Code Qualität        | Beschreibt die Qualität des Codes, dazu ob Hilfsdateien zur Benutzung und Tests vorhanden sind. |
| Geometrie            | Welche Arten von Geometrien werden vom Plugin unterstützt.                                      |
| Sprache              | Unterstützte Sprachen, welche Plugins sind für mehrere Sprachen programmiert.                   |
| Ausführung           | Bei der Ausführung wird getestet ob die Angaben stimmen und ob Fehler auftreten.                |

### 3.2.3 Resultat

Die gesamte Liste ist im separaten Dokument „qgis\_python\_plugins\_evaluation.xlsx“ ersichtlich.

### 3.2.4 Schlussfolgerung

Die meisten Plugins sind durch ihre Einfachheit im Aufbau, keine grosse Hilfe als Inspiration für das eigene NPLCH-Plugin.

Nur in Einem der untersuchten Plugins konnten Test gefunden werden. Und nicht mal dieser konnte nachgestellt werden.

Also konnte auch in diesem Bereich keine Hilfe erwartet werden.

## 4 Erweiterung der Swissphone-Applikation

### 4.1 Ziel

Ein grösseres PyQGIS-Plugin kennen zu lernen, war in diesem Bereich das Hauptziel. Es sollte als Grundlage für die eigene Applikation (NPLCH) dienen. Die Erweiterung ist eine Gegenleistung für das erhaltenen des Codes.

### 4.2 Vorgehen

Nach dem Erhalten des Source Codes konnte mir der Entwickler<sup>3</sup> den ganzen Aufbau erklären. Danach ging es um die Entscheidung welche Erweiterung wir umsetzen wollten. Wir einigten uns auf die Umwandlung der Einheit. [von dBm nach dBμV/m] Danach wurde das Plugin analysiert und anschliessend versuchte ich die beste Lösung für die Erweiterung zu finden und zu implementieren.

---

<sup>3</sup> Die Firma SOURCEPOLE: <http://sourcepole.ch>

### 4.3 Vergleich von Teilbereichen zu NPLCH

| Swissphone                           | NPLCH   |
|--------------------------------------|---|
| Mehrsprachigkeit wird unterstützt    | Mehrsprachig (englisch und deutsch)                     |
| Sqlite-Datenbank                     | PostgreSQL/PostGIS-Datenbank                            |
| Laden von QGIS-Projekten             | dito  |
| Anzeigen von Geometrie aus Datenbank | dito  |
| Anzeigen von Geometrie aus Shapefile | dito  |
| Verwendung von eigenen Styles        | dito  |
| -                                    | Bearbeiten der Geometrie von Shapefiles                 |
| -                                    | Bearbeiten der Geometrie von Datenbank (PostGIS)        |
| -                                    | Einsetzen von Dialogen bei der Erzeugung neuer Elemente |
| Darstellung eines Panels             | dito  |

### 4.4 Aufbau

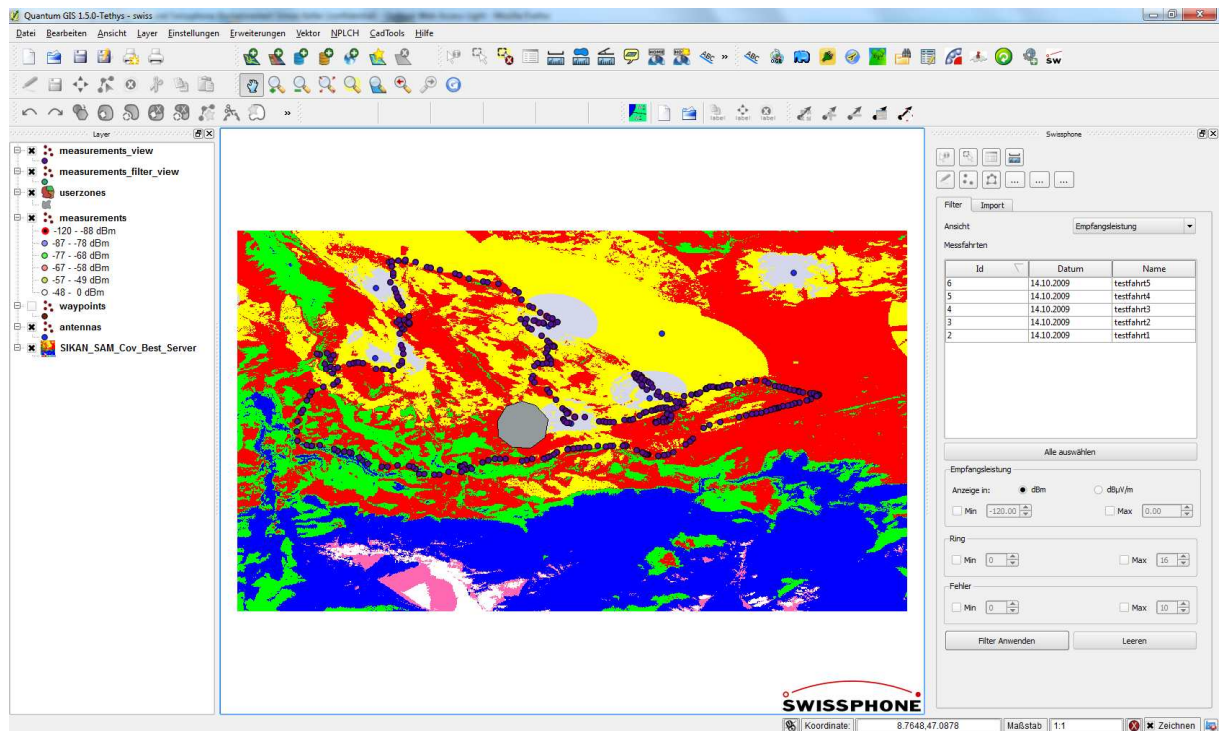


Abbildung 6: Swissphone Beispieldaten in QGIS geöffnet.



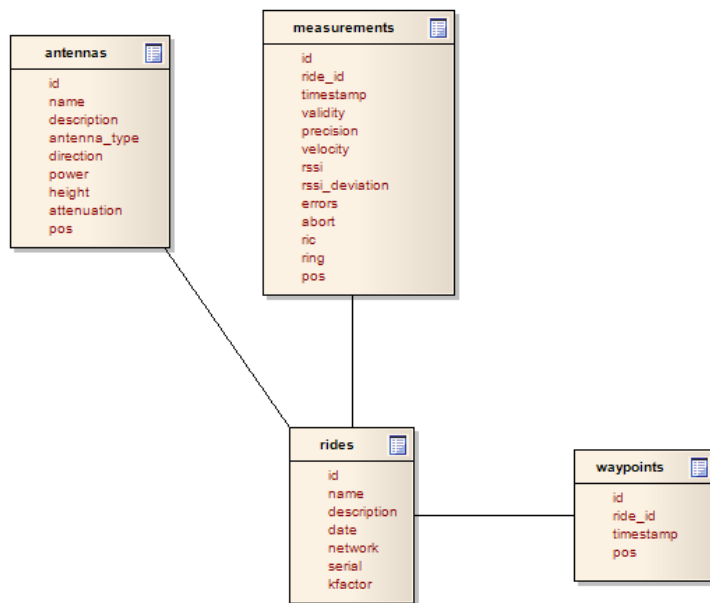


Abbildung 7: Datenbankmodell der Ausgangslage.

## 4.5 Idee

Die Idee ist eine View über die Tabelle „measurements“ mit dem zusätzlichen Feld „rssi\_dbuvm“, welches mit der nachfolgenden Formel umgerechnet wird, zu erstellen.

Formel:

$$x[\text{dB}\mu\text{V}/\text{m}] = y[\text{dBm}] + 77.2 + 20 \cdot \log(f) + k$$

mit  $y$  = dBm-Datenbankwert,  $f$  = Frequenz und  $k$  = Korrekturwert.

### 4.5.1 Umsetzung

Diese saubere und einfache Lösung ist leider im Bezug auf QGIS nicht umzusetzen.

Denn folgende Anforderungen müssen erfüllt sein:

- 1.) Der Typ des neuen Feldes muss ein numerischer Wert sein. Da dieser Wert in QGIS in Kategorien aufgeteilt und so verschiedenfarbig dargestellt wird.

Probleme:

- 1.) View in Spatialite kennen keinen Typ. So wird auch der CAST-Befehl nutzlos.
- 2.) Feld in der View übernimmt den Typ von der unteren Tabelle.

## 4.6 Resultat

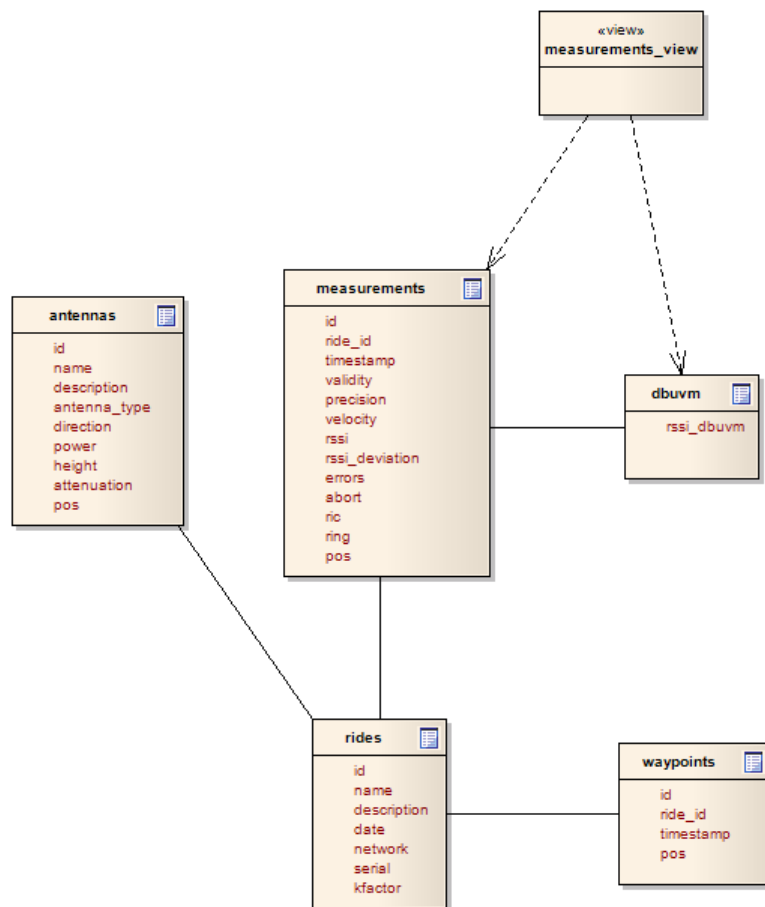


Abbildung 8: Datenbankmodell des Resultates.

Das neue Feld wird über eine neue Tabelle in die View übernommen. Da bei den Tabellen der CAST-Befehl richtig umgesetzt wird, war dies die beste Lösung. Zur Fertigstellung mussten einige kleinere Anpassungen im Code und im User Interface gemacht werden.

## 5 Resultate

### 5.1 Bewertung

Es gibt nun schon einige Plugins zu QGIS, wobei man aber Plugins in diesem Umfang vergeblich sucht. Auch im Bereich Testing wurde ein erster Schritt getan, obwohl noch einige Änderungen in QGIS vollzogen werden müssen, um Unittests vollständig zu unterstützen. Was gewissermassen auch als Neuheit bezeichnet werden kann, sind die QGIS-Einstellungen die direkt aus dem Plugin konfiguriert werden. So können Zonen direkt nach dem Erzeugen von Projekten erstellt werden und die mühsame Konfiguration der Datentypen und Formulareinstellungen entfallen gänzlich.

Es zeigt das hohe Potential von PyQGIS-Plugins auf. Diese Erkenntnis soll möglichst viele Leute dazu motivieren, selbst solche nützliche Plugins zu programmieren.

### 5.2 Zielerreichung

| ID   | Ziel   | Bewertung | Kommentar   |
|------|--|-----------|---|
| UC01 | Projekt erstellen und Daten importieren  | gut       | Gut umgesetzt.                                      |
| UC02 | Daten exportieren  | mässig    | Wenig Daten vorhanden.                              |
| UC03 | Projekt laden  | gut       | Kein Problem.                                       |
| UC04 | Zonen erstellen  | sehr gut  | Sehr schön gelöst mit den Formularen.               |
| UC05 | Topologietests ausführen   | gut       | Gute Prüfung mit Stored Procedure.                  |
| UC06 | Test ob Grundnutzungszonen flächendeckend sind   | gut       | Einfach Lösung                                      |
| UC07 | Weiteres Label hinzufügen  | sehr gut  | Gut an QGIS angepasst.                              |
| NF01 | Die Dokumentation ist klar und verständlich. Es sollte dokumentiert werden, was sinnvoll und notwendig ist, damit die gemachte Arbeit nachvollziehbar ist. | gut       | Ausarbeitung anhand der Dokumentations-Richtlinien. |
| NF02 | Code muss sauber und in englischer Sprache dokumentiert werden.  | gut       | Python eignet sich gut dafür.                       |
| NF03 | Das User Interface soll in englischer und deutscher Sprache angeboten werden.  | gut       | Wird direkt von Qt unterstützt.                     |
| NF04 | Menu und Icons müssen klar erkenntlich sein.   | seht gut  | Schöne integrierte Icons.                           |
| NF05 | Ablauf des Plugins soll intuitiv sein.   | gut       | Einfacher Ablauf.                                   |

### 5.3 Schlussfolgerung und Ausblick

In dieser Arbeit wurde verdeutlicht was alles mit PyQGIS zu machen ist. Aus der NPLCH-Applikation ist ein Prototyp entstanden welchen zur Weiterarbeitung an der SIA Norm 424 beitragen kann. Schon während der Arbeit sind Details hervorgekommen, an welche man noch denken sollte bzw. überarbeiten muss.

Die entstandene Applikation ist im gleichen Stadium wie die SIA Norm 424 und zwar in einem Entwurf bzw. Prototyp. Sie muss mit der Norm weiterentwickelt und angepasst werden. Es besteht auch noch Verbesserungspotential im Bereich User Interface, Handhabung und beim Import/Export der Daten.

Details zu Verbesserungsmöglichkeiten sind im Kapitel 11 „Ausblick“ zu finden.

## 6 Persönlicher Bericht

### 6.1 Arbeitsauswahl

Nach dem bekannt werden der nichtbestanden Arbeit, war meine Motivation auf einem Minimum angelangt. Nach einigen Überlegungen entschied ich mich, die gleiche Arbeit nicht mehr zu wiederholen. Es sollte für mich auch eine Chance auf weitere Erfahrungen sein. Was der misslichen Lage auch nicht zur Besserung verhalf, war die Tatsache, dass die Arbeitsaufteilung des neuen Semesters schon abgeschossen war. Gerne hätte ich eine Arbeit im Bereich „Internet Sicherheit“ geschrieben. Aber das Thema der Arbeit war für mich immer noch ausschlaggebend. So habe ich mich für einen Vorschlag von Herrn Stefan Keller entschieden, da mich das Thema interessierte. Die Erlernung der Programmiersprache Python sah ich als gute Herausforderung.

### 6.2 Durchführung der Arbeit

Die Arbeit wurde sehr gut von Herrn Stefan Keller betreut. Er kannte sich bestens aus, da er auch einen Teil am Entwurf der SIA Norm 424 beigetragen hatte. Die Community der Open Source Produkts QGIS war sehr engagiert. Wann immer ein Problem zum Umgang mit QGIS bestand, konnte eine Lösung mit Ihrer Hilfe gefunden werden. Auch im Bereich Bugs und Entwicklung ging es stets vorwärts. Was den Ablauf der Arbeit ein wenig verzögerte, aber auch spannend und zur selbstinitiative animierte, waren die teils nur grob ausgearbeiteten oder definierten Angaben im Entwurf SIA 424. Viel Zeit brauchte ich auch, um die API von QGIS zu verstehen und die richtigen Objekte, Attribute und Funktionen zu finden.

### 6.3 Mein Fazit

Während der ganzen Arbeit kamen die Vor- und Nachteile einer Einzelarbeit im Vergleich zu einer Gruppenarbeit zum Vorschein. Einfacher war die Aufteilung der Arbeiten und das finden der Termine. Was ich aber vermisste, waren die Tipps und Gedanken einer weiteren Person, wenn ich gerade am Anschlag stand und nicht mehr weiter kam. Auch am Schluss der Bachelorarbeit gab es für eine Einzelarbeit sehr viel zu tun. (Abstract, Poster, Dokumentation, etc.) Im Nachhinein würde ich die Arbeit aber wieder so wie gehabt durchführen. Super war die Integration in die QGIS-Community, welche ich auch am Benutzer-Meeting in Bern kennen lernen durfte. Die Arbeit hat mir grossen Spass gemacht, obwohl ich sehr gefordert wurde. Gefallen hat mir die vielseitige Aufteilung in die verschiedenen Bereiche wie, Kursübungen ausarbeiten, Tutorial anpassen und verschiedene Plugins entwickeln. An der HSR konnte ich am GIS-Lunch meine Applikation vorstellen. Daraus bekam ich ein gutes Feedback, was natürlich ein angenehmer Effekt war.

# TEIL II: SW-PROJEKTDOKUMENTATION NPLCH

## 7 Anforderungsspezifikation

### 7.1 Anforderungen an die Bachelorarbeit

Der folgende Text ist ein Auszug von der Wiki-Seite von Prof. Stefan Keller für die Anforderungen an einer Bachelorarbeit.

*Gerne betreue ich Studien-, Diplom- und Bachelor-Arbeiten, denn dies sind Projektarbeiten, die typischerweise eine Win-Win-Win-Situation für alle Beteiligten darstellen. Die Beteiligten sind Sie als Studierende und Lernende, dann kommen eventuelle externe Firmen oder eine Community und als Drittes sind da noch meine Interessenschwerpunkte, wie (Geo-)Informationssysteme, Information Retrieval, Datenbanken und Java (ev. Skriptingsprachen). Folgendes wird u.a. von Ihnen erwartet:*

- *Wöchentliche Meetings mit:*
  - *"Was wurde gemacht, was ist geplant, wo liegen Schwierigkeiten?"*
  - *Feedback und Hilfestellung meinerseits.*
  - *Plan für Meeting jeweils am Vortag/12h vorher an mich.*
- *Regelmässige/wöchentliche Abgabe von Code/Dokumentation zum Review, idealerweise jeweils 1 Tag vor Statusmeeting, bei umfangreicheren Doks 1/2-1 Woche vorher*
- *Build Prozess (z.B. ANT), automatische Tests (JUnit) und Repository (CVS, SVN)*

*Vorgehensweise: ganz gemäss IFS-Richtlinien - für eine pragmatische aber trotzdem geordnete Vorgehensweise. Grundsätzlich wird gemäss Angaben auf i.hsr.ch bewertet. Dabei gilt, dass für Studentenprojekte manches ein "Muss" ist, dass bei einer ergebnisorientierten Sichtweise im Berufsleben vielleicht nicht nötig wäre. Aber es ist immer notwendig, sinnvoll zu dokumentieren, damit die Arbeit auch von anderen begutachtet werden kann. So sollten zum Beispiel die Anforderungen als Use Cases beschrieben sein. Priorisiert man diese, hat man eine Möglichkeit einen "open scope" Contract zu vereinbaren, wobei ein Mindestmass ein Funktionalität (-> ~4.0) eine wünschenswerte Funktionalität (-> ~5.0) und das Maximum (-> 5.5-6) beschrieben werden können.*

### 7.2 Allgemeine Beschreibung

#### 7.2.1 Einführung

Auszug aus der Aufgabenstellung von Stefan Keller.

Zum einen geht es um das teilweise Realisieren von Übungen zu einem Kurs "Einführung in die GIS-Programmierung mit Python und QGIS" für Programmierneinsteiger. Zur Anforderungsanalyse und zum Testen können Assistenten/innen der HSR-Institute ILF und IRAP beigezogen werden. Zum Zweiten soll das aktuelle QGIS-Python-Plugin-Tutorial (englisch, basierend QGIS Version 1) aktualisiert werden.

Im dritten Teil sollen Entwicklungsumgebungen (IDEs) sowie vorhandene PyQGIS-Plugins evaluiert werden. Wenn sinnvoll, soll ein eigenes Plugin realisiert werden, das produktiv eingesetzt werden kann. Schliesslich geht es 4. um eine Entwicklung einer Applikation zur Verwaltung eines "Zonen-, bzw. Bodennutzungsplans" (Fachschale umfassend Datenstruktur mit Erfassungs- und Verwaltungs- und Datenexport-Funktionen). Vorgängig soll eine Funkantennen-Applikation der Firma Swissphone untersucht werden.

### 7.2.2 Produkt Perspektive

NPLCH ist ein Plugin für QGIS. Dieses Plugin vereinfacht die Nutzungsplanung. Zonen und Festlegungs-Daten können importiert werden, und nach der Bearbeitung auch wieder exportiert werden. QGIS wird durch Funktionen erweitert, welche den Umgang mit den typischen Merkmalen der Nutzungsplanung vereinfacht.

### 7.2.3 Produkt Funktion

- Importieren von Zonen und Festlegungs-Daten
- Exportieren der Daten in der geforderten Darstellung
- Erstellen von Nutzungsplänen
- Bearbeiten von Nutzungsplänen

### 7.2.4 Benutzer Charakteristik

Das Plugin wird von Personen verwendet, welche auf Gemeindeebene, Rahmennutzungspläne erstellen oder bearbeiten. Sie verfügen über Grundkenntnisse im Umgang mit QGIS.

### 7.2.5 Betriebsumgebung

Auf einem beliebigen Windows-System.

### 7.2.6 Einschränkungen

Es gibt Abweichungen beim Plotten von Plänen im Bezug auf den Entwurf des Dokuments SIA 424.

### 7.2.7 Abhängigkeiten

Es wird die Version des QGIS 1.5 vorausgesetzt.

Die Plugins werden mit Python 2.5 entwickelt, daher werden zusätzliche Qt und Python Pakete zur QGIS-Installation vorausgesetzt.

Aus dem osgeo4w-Installer unter Windows) folgende:

Commandline:

- Python

Libraries:

- pyqt4, python-qgis, python-win32, qt4-libs

## 7.3 Use Cases

### 7.3.1 Übersicht

- UC01: Projekt erstellen und Daten importieren
- UC02: Daten exportieren
- UC03: Projekt laden
- UC04: Zonen erstellen
- UC05: Topologietests ausführen
- UC06: Test ob Grundnutzungszonen flächendeckend sind
- UC07: Weiteres Label hinzufügen

### 7.3.2 Aktoren

Der primäre Akteur ist der Benutzer. Er bedient die Software und erstellt oder bearbeitet Rahmennutzungspläne.

### 7.3.3 Use Case Definitionen

#### UC01: Projekt erstellen und Daten importieren

|                           |   |   |
|---------------------------|---|---|
| Use Case ID               | 1   |   |
| Use Case Name             | Projekt erstellen   |   |
| Actors                    | Benutzer  |   |
| Beschreibung              | Ein Benutzer erstellt ein neues Projekt. Dazu verbindet er zum Datenbankserver, gibt einen Datenbanknamen an, wählt die zu Importierenden Zonen- und Festlegungs-Daten anschliessend erzeugt das System die verschiedenen. Tabellen in der Datenbank. |   |
| Vorbedingungen            | <ul style="list-style-type: none"><li>• QGIS ist gestartet.</li><li>• Datenbankserver läuft</li></ul>   |   |
| Nachbedingungen           | <ul style="list-style-type: none"><li>• Das Projekt wurde erfolgreich eröffnet, die Erstellung des Planes beginnt.</li></ul>  |   |
| Standardablauf            | Benutzer  | System  |
|                           | <ol style="list-style-type: none"><li>1. Neues Projekt soll erzeugt werden.</li><li>2. Benutzer öffnet Dialog „Neue Datenbank“ und gibt folgende Parameter ein: Host, Port, User, Passwort, Datenbankname und Zonentypendatei</li></ol>               | <ol style="list-style-type: none"><li>3. System erstellt die Tabellen in der Datenbank.</li><li>4. System importiert die Zonentypendaten</li><li>5. System öffnet das Projekt</li></ol> |
| Alternativer Ablauf       | -   |   |
| Ausnahmen                 | -   |   |
| Häufigkeit des Auftretens | selten (einmal für ein zu bearbeitendes Projekt)  |   |
| Offene Fragen:            | -   |   |



## UC02: Daten exportieren

|                           |  |        |
|---------------------------|--|--------|
| Use Case ID               | 2  |        |
| Use Case Name             | Daten exportieren  |        |
| Actors                    | Benutzer   |        |
| Beschreibung              | Ein Benutzer exportiert nach fertig bearbeitetem bzw. erstelltem Projekt die Daten nach OEREB. Dazu gibt er den Namen der Exportdatei an und drückt auf den entsprechenden Button.   |        |
| Vorbedingungen            | <ul style="list-style-type: none"> <li>UC01: Projekt erstellen und Daten importieren</li> <li>UC04: Zonen erstellen</li> <li>UC05: Topologietests ausführen</li> <li>UC06: Test ob Grundnutzungszonen flächendeckend sind</li> </ul>   |        |
| Nachbedingungen           | <ul style="list-style-type: none"> <li>Die Exportdatei wurde erfolgreich erstellt.</li> </ul>  |        |
| Standardablauf            | Benutzer   | System |
|                           | <ol style="list-style-type: none"> <li>Aktuelles Projekt soll nach OEREB exportiert werden.</li> <li>Benutzer gibt folgende Parameter ein:               <ol style="list-style-type: none"> <li>Benutzer drückt den entsprechenden Button.</li> </ol> </li> <li>System erstellt File mit dem entsprechenden Inhalt.</li> </ol> |        |
| Alternativer Ablauf       | -  |        |
| Ausnahmen                 | -  |        |
| Häufigkeit des Auftretens | selten bis mittel (bei fertigem Projekt)   |        |
| Offene Fragen:            | -  |        |

## UC03: Projekt laden

|                           |  |        |
|---------------------------|--|--------|
| Use Case ID               | 3  |        |
| Use Case Name             | Projekt laden  |        |
| Actors                    | Benutzer   |        |
| Beschreibung              | Der Benutzer lädt ein gespeichertes Projekt. Er öffnet in QGIS das gespeicherte QGIS-Projekt.  |        |
| Vorbedingungen            | <ul style="list-style-type: none"> <li>UC01: Projekt erstellen und Daten importieren</li> <li>Projekt in QGIS gespeichert</li> </ul>                           |        |
| Nachbedingungen           | <ul style="list-style-type: none"> <li>Das Projekt wurde erfolgreich geladen, die Bearbeitung des Planes beginnt.</li> </ul>                                   |        |
| Standardablauf            | Benutzer   | System |
|                           | <ol style="list-style-type: none"> <li>Projekt soll geladen werden.</li> <li>Benutzer öffnet Projekt in QGIS.</li> <li>System lädt das QGIS-Projekt</li> </ol> |        |
| Alternativer Ablauf       | -  |        |
| Ausnahmen                 | -  |        |
| Häufigkeit des Auftretens | häufig (~einmal pro bearbeiten)  |        |
| Offene Fragen:            | -  |        |

## UC04: Zonen erstellen

|                           |   |        |
|---------------------------|---|--------|
| Use Case ID               | 4   |        |
| Use Case Name             | Projekt erstellen   |        |
| Actors                    | Benutzer  |        |
| Beschreibung              | Ein Benutzer erstellt in einem Projekt neue Zonen. Dazu wählt er den zu bearbeiteten Layer aus. Danach erstellt er die Zone mit den QGIS Tools. (Polygon digitalisieren)  |        |
| Vorbedingungen            | <ul style="list-style-type: none"> <li>UC01: Projekt erstellen und Daten importieren</li> <li>Evt. UC03: Projekt laden</li> </ul>   |        |
| Nachbedingungen           | <ul style="list-style-type: none"> <li>Eine oder mehrere Zonen wurden erstellt</li> </ul>   |        |
| Standardablauf            | Benutzer  | System |
|                           | <ol style="list-style-type: none"> <li>1. Layer wählen und in Bearbeitungsmodus schalten.</li> <li>2. Polygon digitalisieren und definieren.</li> <li>3. System erstellt eine neue „Zone“ in der Datenbank</li> </ol> |        |
| Alternativer Ablauf       | -   |        |
| Ausnahmen                 | -   |        |
| Häufigkeit des Auftretens | häufig (mehrmals in einem neuen Projekt)  |        |
| Offene Fragen:            | -   |        |

## UC05: Topologietests ausführen

|                           |   |        |
|---------------------------|---|--------|
| Use Case ID               | 5   |        |
| Use Case Name             | Topologietests ausführen  |        |
| Actors                    | Benutzer  |        |
| Beschreibung              | Ein Benutzer führt nach dem fertigen Erstellen eines Projektes die Topologietests aus. Diese geben zurück ob das Projekt topologiemässig korrekt ist. Wenn nicht werden die Fehler zurückgegeben. |        |
| Vorbedingungen            | <ul style="list-style-type: none"> <li>UC01: Projekt erstellen und Daten importieren</li> <li>UC04: Zonen erstellen</li> </ul>  |        |
| Nachbedingungen           | <ul style="list-style-type: none"> <li>Bei Topologiefehlern werden diese dargestellt.</li> </ul>  |        |
| Standardablauf            | Benutzer  | System |
|                           | <ol style="list-style-type: none"> <li>1. Topologietests ausführen</li> <li>2. System testet die Topologien.</li> <li>3. System erstellt Layer mit den Fehlern</li> </ol>                         |        |
| Alternativer Ablauf       | -   |        |
| Ausnahmen                 | -   |        |
| Häufigkeit des Auftretens | selten bis mittel (bei fertigem Projekt)  |        |
| Offene Fragen:            | -   |        |

## UC06: Test ob Grundnutzungszonen flächendeckend sind

|                           |  |        |
|---------------------------|--|--------|
| Use Case ID               | 6  |        |
| Use Case Name             | Test ob Grundnutzungszonen flächendeckend sind   |        |
| Actors                    | Benutzer   |        |
| Beschreibung              | Ein Benutzer führt nach dem fertigen Erstellen eines Projektes den Test aus, ob die Grundnutzungszonen flächendeckend sind.    |        |
| Vorbedingungen            | <ul style="list-style-type: none"> <li>UC01: Projekt erstellen und Daten importieren</li> <li>UC04: Zonen erstellen</li> </ul> |        |
| Nachbedingungen           | <ul style="list-style-type: none"> <li>Der Benutzer kriegt ein Feedback ob flächendeckend oder nicht.</li> </ul>               |        |
| Standardablauf            | Benutzer   | System |
|                           | 1. Test ausführen. <div>2. System testet die Flächen.</div> <div>3. System gibt Feedback ob flächendeckend.</div>              |        |
| Alternativer Ablauf       | -  |        |
| Ausnahmen                 | -  |        |
| Häufigkeit des Auftretens | selten bis mittel (bei fertigem Projekt)   |        |
| Offene Fragen:            | -  |        |

## UC07: Weiteres Label hinzufügen

|                           |   |        |
|---------------------------|---|--------|
| Use Case ID               | 7   |        |
| Use Case Name             | Weiteres Label hinzufügen   |        |
| Actors                    | Benutzer  |        |
| Beschreibung              | An einer bestehenden Zone wird ein weiteres Label hinzugefügt.  |        |
| Vorbedingungen            | <ul style="list-style-type: none"> <li>UC01: Projekt erstellen und Daten importieren</li> <li>UC04: Zonen erstellen</li> </ul>  |        |
| Nachbedingungen           | <ul style="list-style-type: none"> <li>Weiteres Label zur Zone hinzugefügt.</li> </ul>  |        |
| Standardablauf            | Benutzer  | System |
|                           | 1. Wählt den Beschriftungslayer.<br>2. Makiert das zu kopierende Label.<br>3. Drückt auf das Icon „Kopiere Label“.<br>4. Wählt den neuen Ort. <div>5. System kopiert Label an neuen Ort</div> |        |
| Alternativer Ablauf       | -   |        |
| Ausnahmen                 | -   |        |
| Häufigkeit des Auftretens | häufig (mehrmals in einem neuen Projekt)  |        |
| Offene Fragen:            | -   |        |

### 7.3.4 Use Case Diagramm

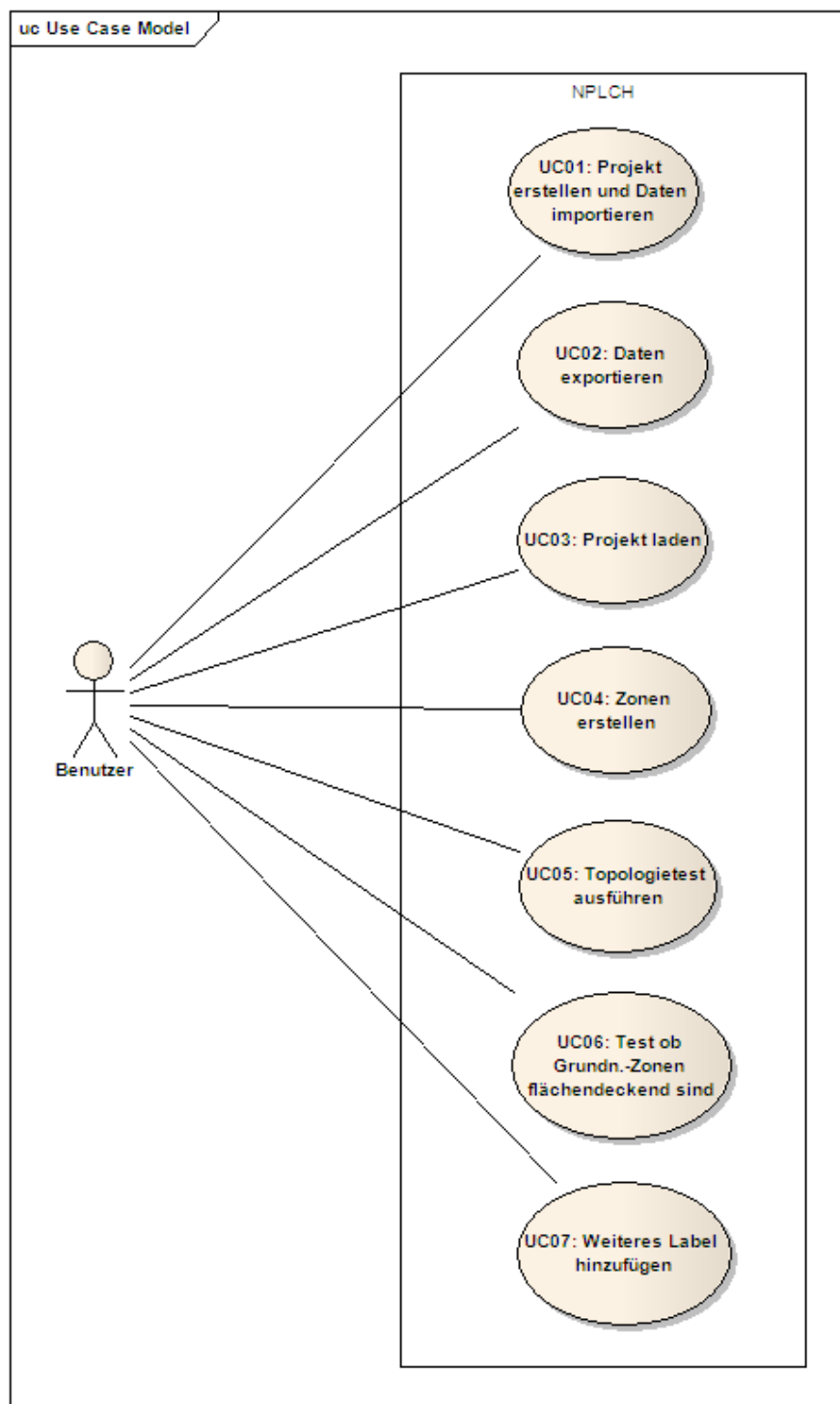


Abbildung 9: Use Case Diagramm mit den einzelnen Use Cases.

## 7.4 Spezifische Anforderungen

### 7.4.1 Funktionale Anforderungen

Die funktionalen Anforderungen entsprechen den definierten und ermittelten Use Cases. Siehe Kapitel 7.3.1 „(Use Case) Überblick“.

### 7.4.2 Nicht-funktionale Anforderungen

| ID   | Ziel   | Kategorie     |
|------|--|---------------|
| NF01 | Die Dokumentation ist klar und verständlich. Es sollte dokumentiert werden, was sinnvoll und notwendig ist, damit die gemachte Arbeit nachvollziehbar ist. | Dokumentation |
| NF02 | Code muss sauber und in englischer Sprache dokumentiert werden.  | Entwicklung   |
| NF03 | Das User Interface soll in englischer und deutscher Sprache angeboten werden.  | Entwicklung   |
| NF04 | Menu und Icons müssen klar erkenntlich sein.   | Entwicklung   |
| NF05 | Ablauf des Plugins soll intuitiv sein.   | Entwicklung   |

### 7.4.3 Schnittstellen

#### *Benutzer*

- Das GUI ist in QGIS integriert.
- Die einzelnen Ebenen sind im QGIS-Panel „Layers“ dargestellt.
- Zur Benutzung des Plugins steht ein Panel und eigene Icons zur Verfügung.

#### *Hardware*

Das Plugin ist durch das QGIS plattformunabhängig. Folglich wird keine weitere Hardware benötigt.

#### *Software*

- Python
- Qt-Bibliothek

#### *Datenbank*

PostgreSQL Datenbank mit der PostGIS-Erweiterung

### 7.4.4 Lizenzanforderungen

Es werden keine Lizenzen für allfällige Klassenbibliotheken, Programmiersprachen oder Konzepte benötigt.

Die Plugins stehen nach der Veröffentlichung unter der *GNU Lesser General Public License*.

## 8 Analyse

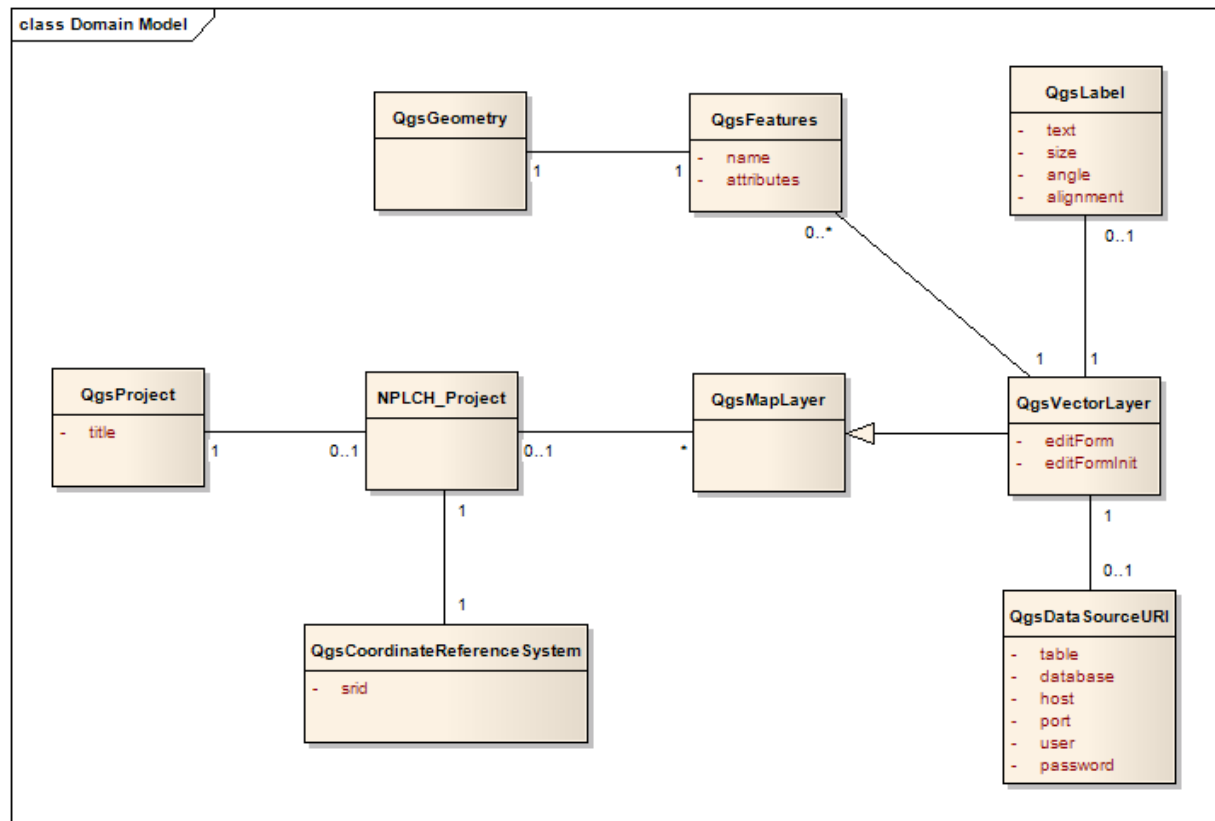


Abbildung 10: Klassendiagramm der Problem Domain.

## 9 Design & Implementation

### 9.1 Architektonische Darstellung

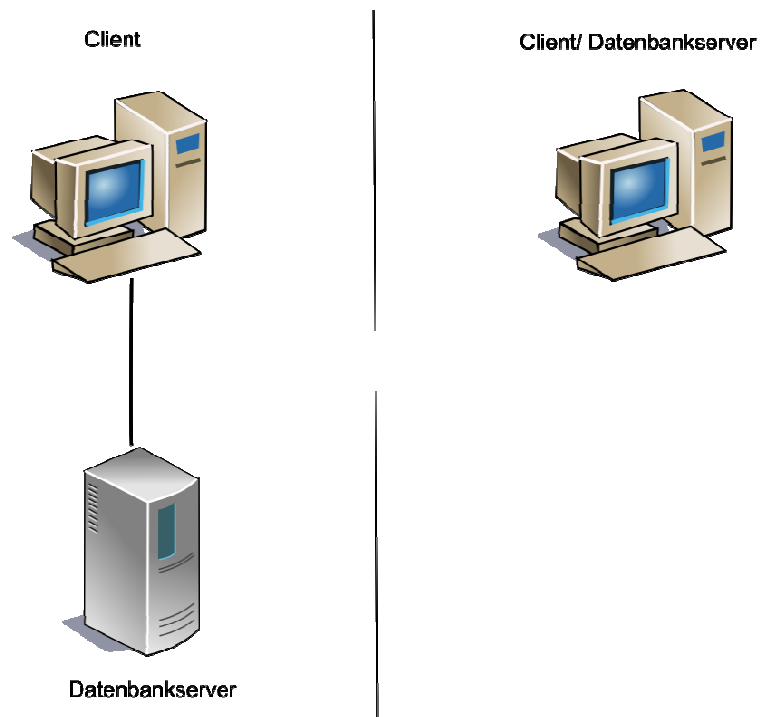


Abbildung 11: Physikalische Architektur.

Die NPLCH-Applikation beruht auf einem Client-Server-System. Der Client und der Datenbankserver können auf dem gleichen Rechner laufen oder auf Verschiedenen welche via Netzwerk miteinander verbunden sind. Der Client liest und bearbeitet die Daten in der Datenbank durch abschieken von SQL-Befehlen. Jeden abgeschickten SQL-Befehl sendet der Client als Anforderung an den Server diesen dort auszuführen. Das Ergebnis liefert der Server als Antwort zurück.



## 9.2 logische Architektur

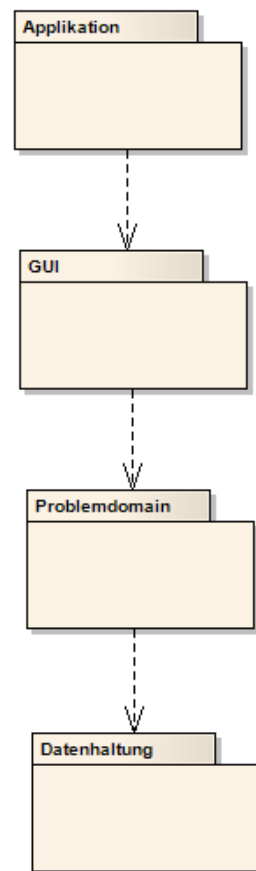


Abbildung 12: Schichten Architektur

## 9.3 Design Pakete

Detaillierte Informationen sind auf der CD unter nplch/doc zu finden.

---

### Package Applikation

---

#### *Beschreibung des Package*

Diese Ebene ist für die Einbindung in QGIS verantwortlich. Die Hauptfunktionen sind von QGIS vorgegeben.

## Diagramme

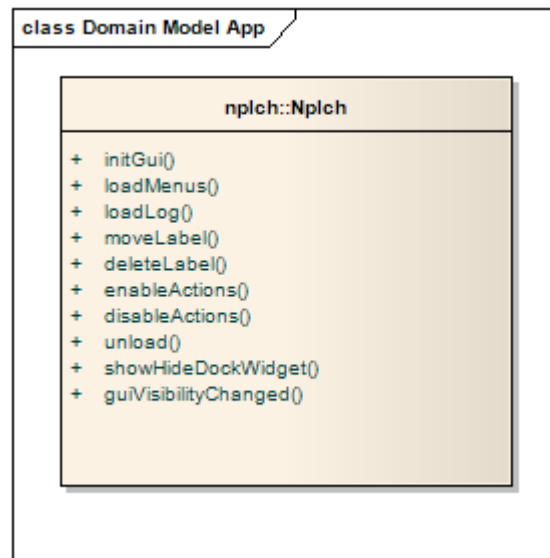


Abbildung 13: Applikation Domain Modell.

## Schnittstellen

### QGIS <-> PLUGIN

| Schnittstelle | Beschreibung                                       |
|---------------|--|
| initGUI()     | Wird ausgeführt, wenn das Plugin geladen wird.     |
| unload()      | Wird ausgeführt, wenn das Plugin geschlossen wird. |

## Klassen und Operationen

| NPLCH  |
|--|
| <b>Initialisiert das Plugin.</b>   |
| <b>Beschreibung</b>  |
| Diese Klasse ist für die Verbindung zwischen dem Plugin und QGIS verantwortlich. Es initialisiert die Menus und Icons. Dazu startet Sie eine Logger für Debugausgaben. |

## Package Gui

### Beschreibung des Package

Dieses Package ist für die Darstellung der Daten zuständig. Das heisst, die gesamte grafische Darstellung läuft über dieses Package.

### Diagramme

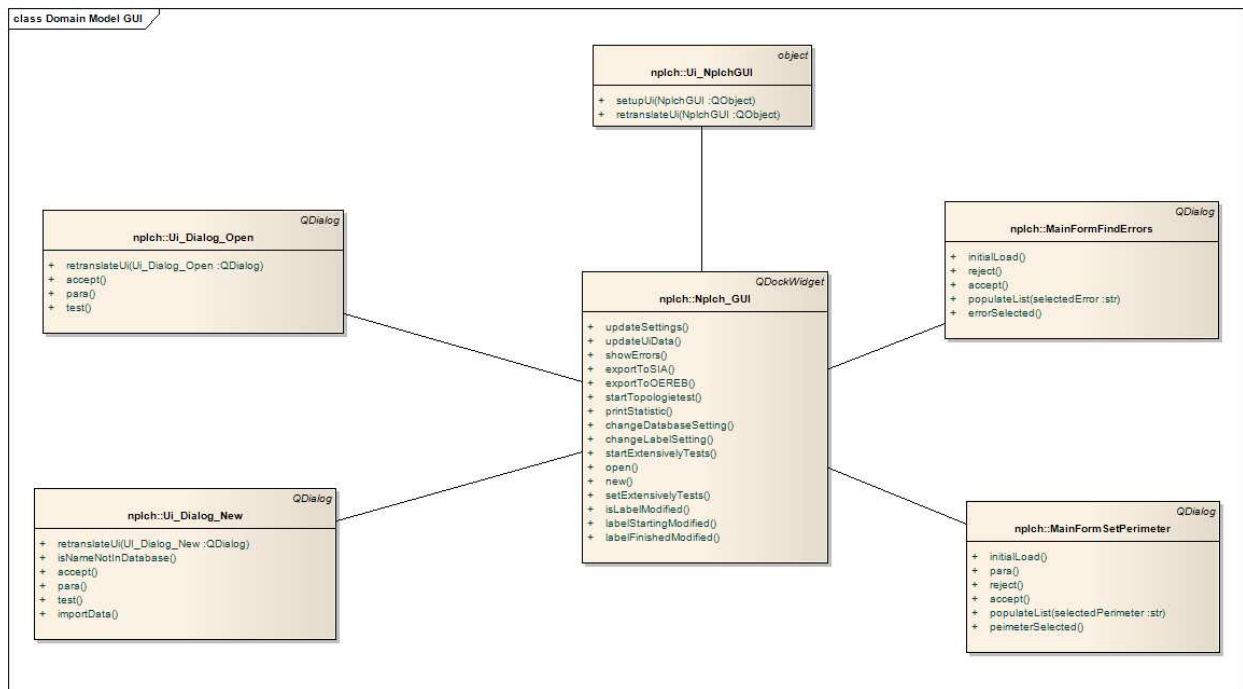


Abbildung 14: GUI Domain Modell.

### Schnittstellen

Die wichtigste Schnittstelle, welche das GUI darstellt, ist die Schnittstelle zwischen Benutzer und System. Durch das grafische User-Interface werden sämtliche Benutzereingaben an die Problem Domain weitergeleitet und bearbeitet.

### Klassen und Operationen

#### Ui\_Dialog\_New

**Dialog zum erstellen neuer Projekte.**

##### Beschreibung

Dieser Dialog übergibt den Namen des neuen Projektes zurück damit die neue Datenbank erstellt werden kann.

#### Ui\_Dialog\_Open

**Dialog zum öffnen eines Projekte.**

##### Beschreibung

Dieser Dialog öffnet ein Projekt von der Datenbank.

|   |
|---|
| <b>Ui_NplchGUI</b>  |
| <b>NPLCH-DockWidget</b>   |
| <b>Beschreibung</b>   |
| Diese Klasse ist für das Hauptfenster der NPLCH-Applikation verantwortlich. |

|  |
|--|
| <b>MainFormFindError</b>                       |
| <b>QGIS Fenster</b>                            |
| <b>Beschreibung</b>                            |
| Eine Liste um zu einem Fehler darin zu zoomen. |

|   |
|---|
| <b>MainFormSetPerimeter</b>             |
| <b>QGIS Fenster</b>                     |
| <b>Beschreibung</b>                     |
| Eine Tabelle zum setzen des Perimeters. |

|  |
|--|
| <b>Nplch_GUI</b>                         |
| <b>Hauptklasse</b>                       |
| <b>Beschreibung</b>                      |
| Verwaltet alle weiteren GUI Aktivitäten. |

## Package Problem Domain

### Beschreibung des Package

Die Problem Domain kapselt die Programmlogik, welche das Programm ausmacht. Sie verwaltet die QGIS-Objekte. Des Weiteren befinden sich hier auch die zur Programmlogik gehörenden Funktionen.

### Diagramme

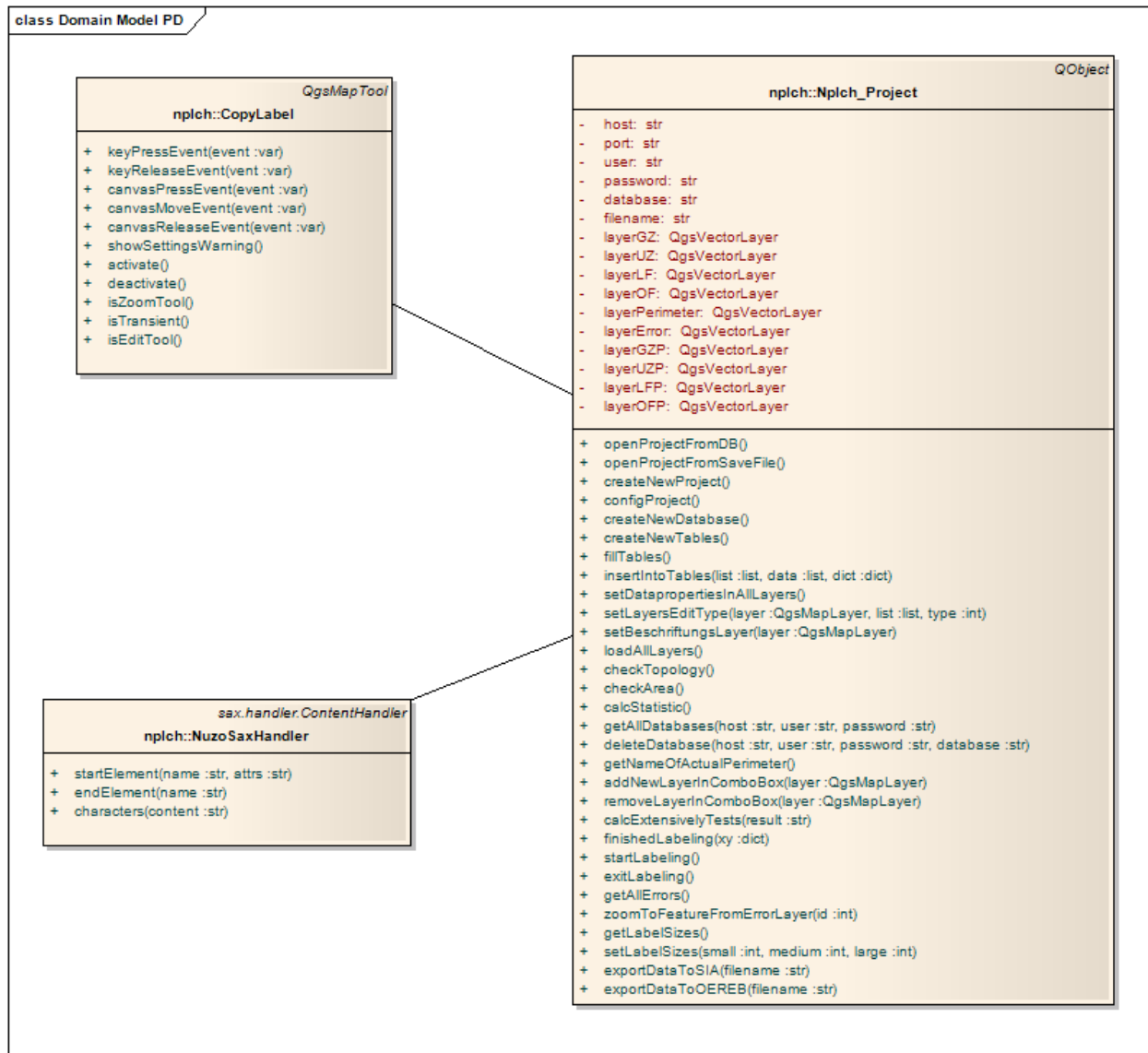


Abbildung 15: Problem Domain Modell

### Schnittstellen

Als Schnittstelle zur Problem Domain stellt sich die Klasse „NplchProject“ zur Verfügung. Das heisst die meisten Operationen welche die Problem Domain betreffen laufen über Sie ab.

---

## *Klassen und Operationen*

### **CopyLabel**

**Wählt Punkt für das neue Label aus.**

#### **Beschreibung**

Diese Klasse ist ein QGISMapTool und dient zur Definition des neuen Einsetzpunktes für das Label.

### **NuzoSaxHandler**

**XML-Handler zur einlesen der Zonentypdaten.**

#### **Beschreibung**

Klassischer Sax-Handler für XML.

### **Nplch\_Poject**

**Verwaltet ein NPLCH-Projekt.**

#### **Beschreibung**

Hauptklasse in der Problem Domain, verwaltet das ganze Projekt.

## Package Datenhaltung

### Beschreibung des Package

Das Package Datenhaltung beinhaltet die Schnittstelle zur Datenbank.

### Diagramme

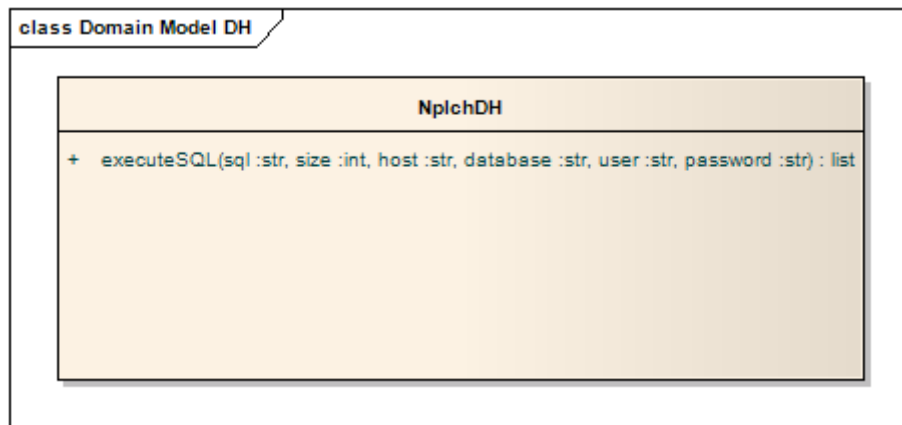


Abbildung 16: Datenhaltung Domain Modell.

### Schnittstellen

Die Methode „executeSQL“ beschreibt die Schnittstelle zur PostgreSQL-Datenbank.

### Klassen und Operationen

| NplchDH  |
|--|
| <b>Stellt die Schnittstelle zur Datenbank zur Verfügung.</b>             |
| <b>Beschreibung</b>  |
| Führt den SQL-Befehl auf der Datenbank aus und gibt das Resultat zurück. |

### 9.3.1 Zusammenspiel der Packages

In jedem Packet kann nur immer die Hauptklasse auf die Hauptklasse des Packages tiefer zugreifen.

Siehe „zusammenspiel.pdf“

## 9.4 Datenspeicherung

Zur Speicherung der Daten wird eine PostgreSQL-Datenbank mit der PostGIS-Erweiterung eingesetzt.

Nachfolgend eine Übersicht der Datenbank:

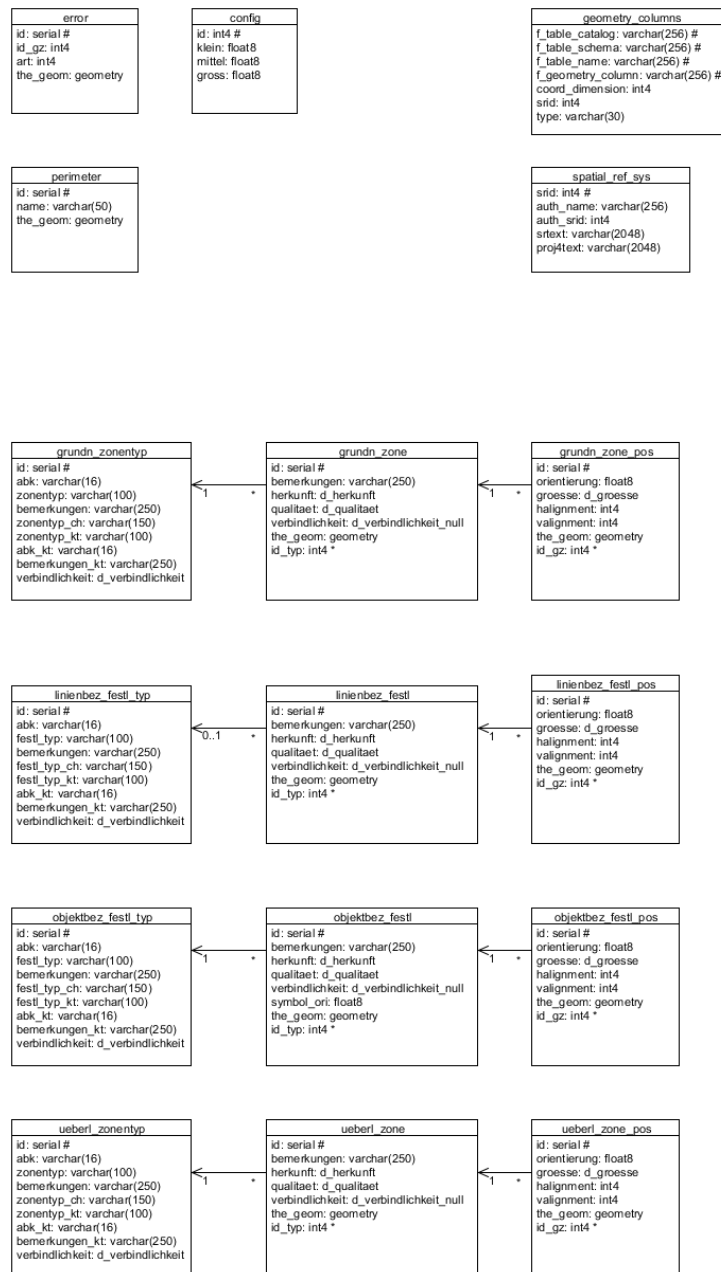


Abbildung 17: Datenbankschema



### 9.4.1 Import in QGIS

Damit die ursprüngliche Form beibehalten werden konnte wurden zu jedem Typ (Grundnutzungszone, Überlagernde Zone, etc.) zwei Views erstellt.

Diese Views verbinden die einzelnen Tabelle und zeigen die Zonen und die Beschriftungen an.

Einige Hilfsfunktionen übernehmen die Umwandlung der Attribute von Interlis nach QGIS.

Zum Beispiels die Positionierung des Anfasspunktes von 15 auf 9 Punkte.

### 9.4.2 Stored Procedure

Zum Topologietest wurde eine Stored Procedure geschrieben. Diese kann leicht erweitert werden.

Direkte Manipulationen können an der SQL-Datei (nplch/data/sqlTopologieTests.sql) vorgenommen werden.

Dafür ist der Aufbau des Layers Error von Nöten. Die wichtigen Attribute:

| Attribut | Beschreibung  |
|----------|---|
| art      | Legt die Art des Fehlers fest. 1= Zonen mit Differenz, 2 = Zonen mit Überschneidungen, 3 = Differenz und 4 = Überschneidung |
| id_gz    | Wenn es Grundn.-Zonen sind, gibt dies die Referenz an.  |

## 10 Tests

### 10.1 Technologie

Die Unit Tests wurden in PyUnit geschrieben. Dies ist eine Version von JUnit (Java) in Python geschrieben.

Zum Testen der Graphischen Benutzeroberfläche wird SIKULI<sup>4</sup> eingesetzt. Dies ist eine Software, welche nach bestimmten Bildschirmausschnitten sucht und diese identifizieren kann.

### 10.2 Automatische Tests

Durch einen Fehler in der Zusammenarbeit zwischen QGIS und Python unter Windows, konnten die QGIS Libraries(qgis.core und qgis.gui) nicht importiert bzw. gefunden werden. So konnten nur die Methoden getestet werden, welche keine der besagten Libraries verwenden. Die Entwickler von QGIS sollten auf die neueste Version von Python updaten und so diesen Bug bereinigen.<sup>5</sup>

### 10.3 Manuelle Tests – Use Case Tests

Nach jedem abgeschlossen Use Case wird dieser auf seine Funktion getestet. Die bisherigen Tests werden immer wiederholt.

### 10.4 Nicht berücksichtigte Punkte

Wie schon erwähnt konnten viele Unit Test nicht durchgeführt werden.

### 10.5 Testfazit

Leider konnte nicht wie gewünscht getestet werden. Doch im Gegensatz zu den anderen Plugins ist dies schon mal ein Anfang. Bei dem Update auf eine neuere Pythonversion sollte dies wenn möglich noch korrigiert werden. Doch mit den GUI-Tests sind die wichtigen Bereiche abgedeckt.

---

<sup>4</sup> Informationen unter <http://sikuli.csail.mit.edu>

<sup>5</sup> Weitere Informationen unter <http://www.qgis.org/community/mailling-lists.html>

## **11 Ausblick**

### **11.1 Weiterentwicklung**

#### **11.1.1 Unterstützung weiterer Betriebssysteme**

Im Moment ist die ganze Applikation nur unter Windows getestet und lauffähig. Die zusätzliche Unterstützung von Linux und Mac sollte keine grossen Probleme mit sich bringen und wäre eine kleinere Arbeit.

#### **11.1.2 Ausarbeiten von qualitativ hohen Beispieldaten.**

Das importieren und exportieren konnte bis jetzt nur mit ganz simplen Testdaten kontrolliert werden. Um alle Sonderfälle abzudecken und zu prüfen, wären bessere oder einfach qualitativ höhere Beispieldaten nötig.

#### **11.1.3 XML-Handler und Writer**

Durch zu ungenauen Definitionen und zu wenig Zeit, konnte der Xml-Handler und Writer erst sehr primitiv hinzugefügt werden. Dies wäre natürlich ein wichtiger Punkt bei der Weiterentwicklung.

#### **11.1.4 Anpassungen an GUI**

Durch grössere Usability-Tests könnte das GUI noch auf die intuitiven Abhandlungen solcher Projekte zugeschnitten werden. Bei einem definierten Vorgang könnte noch einiges angepasst werden.

#### **11.1.5 Verbessern der Möglichkeiten zum Erzeugen von neuen Zonen**

Mit Hilfe der Integration von andern Plugins und den Möglichkeiten von QGIS, kann eine einfachere und weniger anfällige Erzeugung und Bearbeitung neuer Zonen erreicht werden.

## 12 Projektmanagement

### 12.1 Projektorganisation

Diese Bachelorarbeit ist eine Einzelarbeit.

Es steht dem Projekt ein Dozent als Betreuer zur Seite.

#### 12.1.1 Organisationsstruktur

**Projektmitglied:**

- SiK / Simon Keller ([s1keller@hsr.ch](mailto:s1keller@hsr.ch))

**Betreuer:**

- SfK / Prof. Steffan Keller ([sfkeller@hsr.ch](mailto:sfkeller@hsr.ch))

**Gegenleser:**

- HH / Prof. Hansjörg Huser ([hhuser@hsr.ch](mailto:hhuser@hsr.ch))

#### 12.1.2 Externe Schnittstelle

Als externe Schnittstelle dienen der Betreuer des Projektes, sowie die Open-Source Community von QGIS.

## 12.2 Management Abläufe

#### 12.2.1 Aufwand

- Der Zeitaufwand beträgt zwischen 390-440 Stunden pro Projektmitglied. (25 h/Woche)
- Das Projekt startet am 22.02.2010 und endet am 18.06.2010. (17 Wochen)
- Die Arbeitszeit pro Woche kann variieren, am Ende des Projekts sollte der gesamte Zeitaufwand jedoch in dem besagten Rahmen liegen.

#### 12.2.2 Zeitplan

Siehe separates Dokument „Projektplan.xlsx“.

#### 12.2.3 Meilensteine

| Meilenstein | Teil  | Beschreibung              | Arbeitsergebnisse  | Datum            |
|-------------|-------|---------------------------|--|------------------|
| MS1         | Allg. | Projektplan Review        | Im Projektplan ist die grundlegende Planung für die Iterationen erfasst. | 12.03.10<br>SW03 |
| MS2         | 1     | Evaluation IDE/PyQGIS     | Überprüfung der vorhandenen Entwicklungsumgebungen und PyQGIS-Plugins    | 19.03.10<br>SW04 |
| MS3         | 3 & 4 | Anforderungen und Analyse | Anforderungen und Analyse der Fachapplikation sind abgeschlossen.        | 02.04.10<br>SW06 |
| MS4         | 4     | Prototyp Alpha            | Die Designphase ist abgeschlossen. Es existiert eine einfache grafische  | 16.04.10         |

|     |        |  |  |                  |
|-----|--------|--|--|------------------|
|     |        | Version  | Oberfläche. Die grundlegenden Funktionen sind implementiert.   | SW08             |
| MS5 | 3. & 4 | Release Candidate 1 / Swissphone Erweiterung Final Version | Das NPLCH-Plugin ist in einer Version verfügbar welches öffentlich gemacht werden kann. Die Swissphone Erweiterung wurde fertiggestellt. | 07.05.10<br>SW11 |
| MS6 | 2      | Einführung in die GIS-Programmierung mit Python und QGIS   | Ausgearbeitete Übungen zu dem Kurs „Einführung in die GIS-Programmierung mit Python und QGIS“  | 14.05.10<br>SW12 |
| MS7 | 4      | Final Version  | Das Plugin wurde fertiggestellt inkl. Dokumentation(Korrektur, Reviews und Ausdruck)   | 28.05.10<br>SW14 |
| MS8 | Allg.  | Abgabe   | Schlussabgabe des Projekts in elektronischer und gedruckter Form.  | 9.06.10<br>SW16  |

#### Meilensteine

#### 12.2.4 Phasen-/Iterationsplanung

| Iteration            | Beschreibung   | Ende | Dauer in Wochen |
|----------------------|--|------|-----------------|
| Inception 1          | Dokumentvorlagen, Projektplan (Zeitplan, Risiko Management, Codierungsrichtlinien, Qualitätsmerkmale) , Konfigurationsverwaltung   | SW03 | 3               |
| Elaboration 1        | Plugin-Prototyp, Evaluation Plugins und IDE  | SW04 | 1               |
| Elaboration 2        | Design Model, Logische Architektur der Packages, 100% Use Cases im fully dressed Format, Externen Design, Internes Design des UI incl. Prototyp einer einfachen grafischen Oberfläche, | SW06 | 2               |
| Construction 1       | Prototyp des Plugins, Prototyp des UI  | SW08 | 2               |
| Construction 2       | Implementation weiterer Funktionen, Ausbau der GUI sowie Usability Tests   | SW11 | 3               |
| Construction 3       | Erweiterungen an der grafischen Oberfläche(aus den Usability Tests), Bugfixing, Verbesserungen von den User Tests, (Implementierung zusätzlicher Features)                             | SW14 | 3               |
| Transition & Reserve | Fertigstellung der Dokumentation, Vorbereitung der Abgabe  | SW16 | 2               |

Tabelle Iterationen

## 12.3 Risiko Management

Siehe separates Dokument „Risiko\_Kosten.pdf“.

## 12.4 Arbeitspakete

Siehe separates Dokument „Projektplan.xlsx“.

## 12.5 Infrastruktur

### 12.5.1 Räumlichkeiten

Ein Arbeitsplatz ist an der HSR im Raum 1.258 für dieses Projekt reserviert. Alle Arbeiten werden mobil, zu Hause oder an diesem Arbeitsplatz durchgeführt.

### 12.5.2 Hardware

#### 12.5.2.1 Notebook, Arbeitsplatzcomputer, Heimcomputer

Die Arbeiten werden auf dem Notebook präsentiert. Die meiste Arbeit wird am Arbeitsplatzcomputer an der HSR oder am Privatrechner zu Hause erledigt.

#### 12.5.2.2 SVN / Trac

Für das zentrale Versionsmanagement des Codes und der Dokumente steht ein Server der HSR mit installiertem SVN / Trac zur Verfügung.

### 12.5.3 SoftwareQualitätsmassnahmen

| Applikation            | Version | Beschreibung   |
|------------------------|---------|--|
| QGIS                   | 1.5     | Geoinformationssystem-Software                                     |
| Eclipse                | 3.5.1   | Entwicklungsumgebung   |
| Pydev für Eclipse      | 1.5.5   | Python Unterstützung für Eclipse                                   |
| PostgreSQL             | 8.4     | Objektrelationales<br>Datenbankmanagementsystem                    |
| PostGIS                | 1.5.1   | Erweitert PostgreSQL um geografische Objekte<br>und Funktionen     |
| Microsoft Office Paket | 2007    | Textverarbeitung, Tabellenkalkulation und<br>Präsentationssoftware |
| GIMP                   | 2.6     | Bildbearbeitungs Programm  |
| Enterprise Architect   | 7       | UML tool   |
| Inkscape               | 0.47    | Software zur Bearbeitung und Erstellung von<br>Vektorgrafiken      |

#### 12.5.4 Coding Guidelines

- Die verwendeten Code Richtlinien basieren auf den Standardeinstellungen des Code Formatters von Eclipse bzw. Pydev. Sowie dem PEP 8—Style Guide for Python Code. [[www.python.org/dev/peps/pep-008](http://www.python.org/dev/peps/pep-008)]
- Der Code sollte mit den anderen Plugins übereinstimmen.
- Die Funktionsnamen müssen aussagekräftig gewählt werden. Die Namen sollten möglichst ohne Kommentare auf den Zweck der Methode deuten.
- Der Inhalt komplexer Funktionen wird mit Hilfe von Docstrings beschrieben.
- Der Code innerhalb einer Methode wird sauber strukturiert.
- Es sollen die in den Software Engineering gelernten Praktiken und Standards angewandt werden. (Patterns, Code Richtlinien)

#### 12.5.5 Code Reviews/Inspections

Während dem Projekt werden einzelne Systemelemente und Dokumente durch den Betreuer reviewt und an den wöchentlichen Meetings besprochen.

#### 12.5.6 Versionsmanagement

Alle Daten des Systems werden im Projekt-SVN verwaltet. Änderungen können somit jederzeit nachvollzogen oder rückgängig gemacht werden.

#### 12.5.7 Tests

Während der Entwicklung werden die einzelnen Programmteile unter Verwendung von PyUnit-Tests einer Prüfung unterzogen. Des Weiteren wird in den Code Reviews nach Fehlern gesucht damit diese behoben werden können.

#### 12.5.8 Automatisierung

Der Buildprozess wird mit Hilfe von Apache Ant automatisiert.

Er erstellt die Plugins aus den Sourcen.

### 12.6 Sitzungs-Protokolle

Es wurde zu jeder Sitzung mit SfK ein Sitzungsprotokoll erstellt. Dies war nach spätestens zwei Tagen nach der Sitzung für die Teilnehmenden auf dem Trac abrufbar.

## 12.7 Projektmonitoring

### 12.7.1 Soll-IST-Zeit-Vergleich

Anschliessend eine Tabelle mit den Ist- und Soll-Zeiten. Die genauen Angaben zu den einzelnen Paketen können dem Excel Projektplan entnommen werden.

| Überpakete          | Soll  | Ist   | Differenz |
|---------------------|-------|-------|-----------|
| Projekt Management  | 12.0  | 13.5  | -1.5      |
| Requirements        | 9.0   | 11.0  | -2.0      |
| Analyse             | 42.0  | 45.5  | -3.5      |
| Design Analyse      | 14.0  | 7.5   | 6.5       |
| Implementation      | 205.0 | 279.5 | -74.5     |
| Qualitätsmassnahmen | 49.0  | 47.5  | 1.5       |
| Dokumentation       | 68.0  | 84.5  | -16.5     |
| Sitzungen           | 51.0  | 57.0  | -6        |

### 12.7.2 Codestatistik

| Gesamt            |       |
|-------------------|-------|
| Plugins           | 1     |
| Packages          | 5     |
| Klassen           | 24    |
| Source code lines | 5'372 |
| Total lines       | 5'963 |

| Packages       | Total lines | Source code lines | Comment lines |
|----------------|-------------|-------------------|---------------|
| Applikation    | 1501        | 1324              | 129           |
| GUI            | 304         | 245               | 48            |
| Problem Domain | 59          | 51                | 7             |
| Datenhaltung   | 1196        | 1071              | 117           |
| Forms          | 2814        | 2673              | 188           |
|                |             |                   |               |
| Total          | 5874        | 5346              | 489           |



## 13 Softwaredokumentation

Diese Anleitung beschreibt die Installation, Konfiguration und die Anwendung der NPLCH-Applikation.

### 13.1 Voraussetzungen

Folgende Programme werden benötigt:

| Programm                  | mind. Version |
|---------------------------|---------------|
| Quantum QGIS <sup>6</sup> | 1.5           |
| Python                    | 2.6           |
| PostgreSQL                | 8.4           |
| PostGIS                   | 1.5           |

### 13.2 Installation

Starten Sie Quantum GIS. Zum Beispiel wie nachfolgend via Icon auf dem Desktop. Achten Sie auf die Version 1.5.



Abbildung 18: Icon von QGIS 1.5.0.

Wenn das Programm gestartet ist, geht es über das Menu „Erweiterungen“, „Python-Erweiterung herunterladen...“ weiter.

<sup>6</sup> Informationen zur Installation unter [www.qgis.org/de/herunterladen.html](http://www.qgis.org/de/herunterladen.html) oder [http://gis.hsr.ch/wiki/Quantum\\_GIS](http://gis.hsr.ch/wiki/Quantum_GIS)

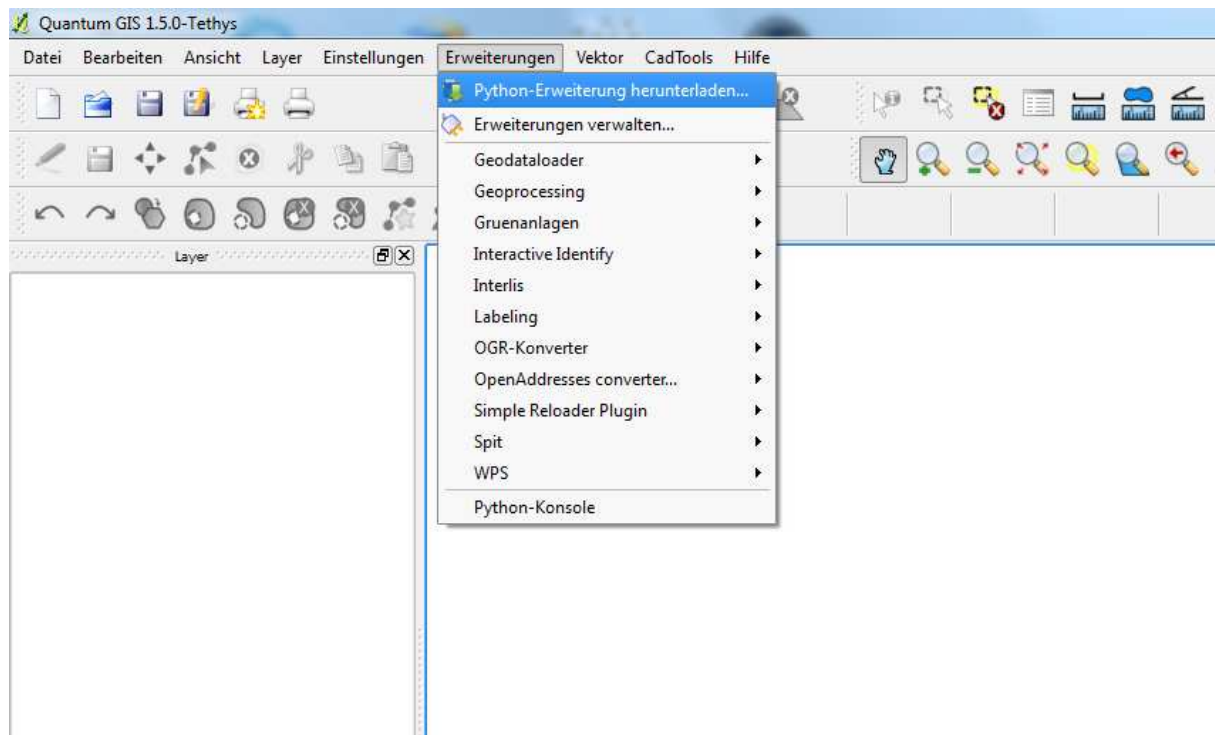


Abbildung 19: QGIS im Menu "Erweiterungen".

Es öffnet sich folgendes Fenster:

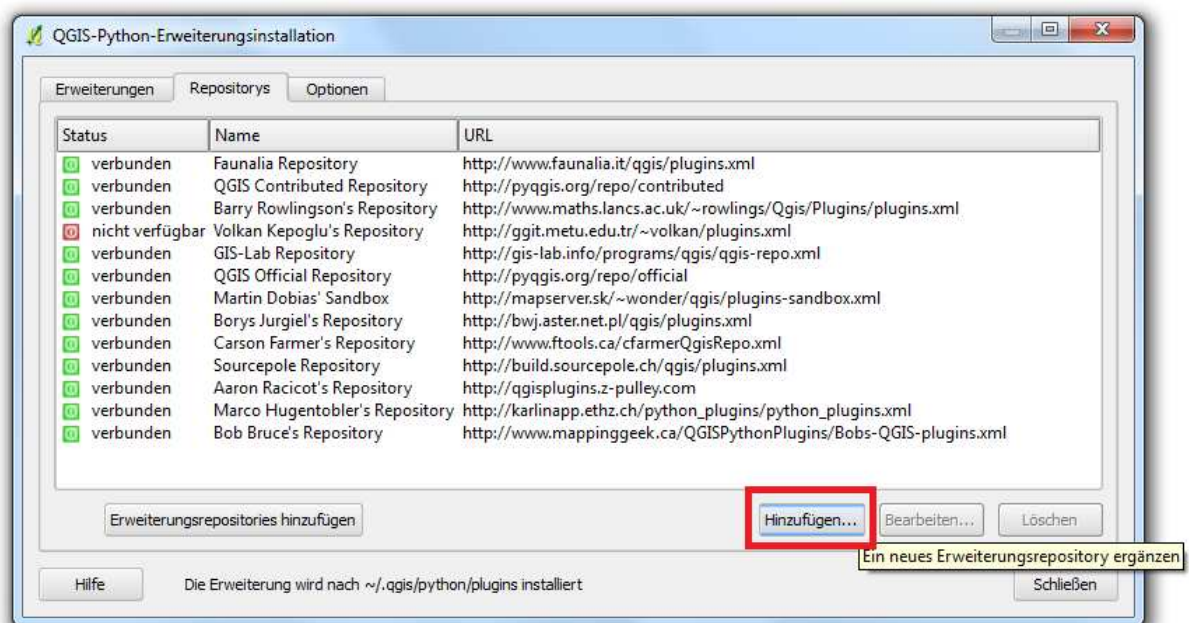


Abbildung 20: Hinzufügen eines Repositories.

Drücken Sie auf den Button „Hinzufügen“ um die Repositories zu erweitern.

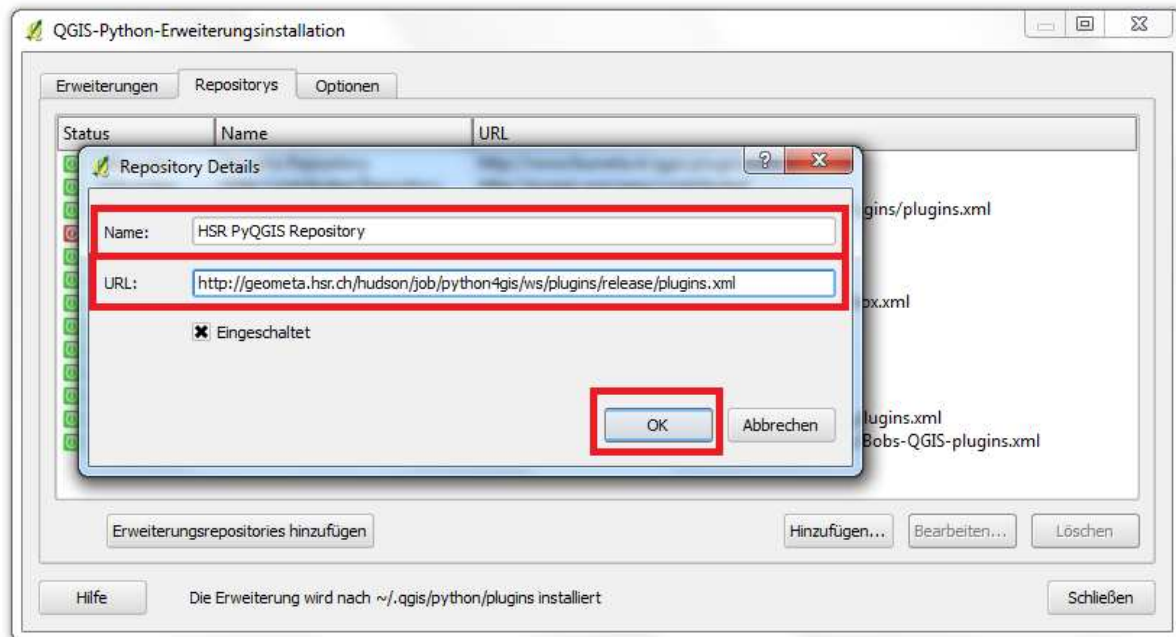


Abbildung 21: Detaileingabe des Repository.

Folgende Angaben sind einzugeben:

|       |   |
|-------|---|
| Name: | HSR PyQGIS Repository   |
| URL:  | <a href="http://geometa.hsr.ch/hudson/job/python4gis/ws/plugins/release/plugins.xml">http://geometa.hsr.ch/hudson/job/python4gis/ws/plugins/release/plugins.xml</a> |

Mit „OK“ werden die Daten hinzugefügt.

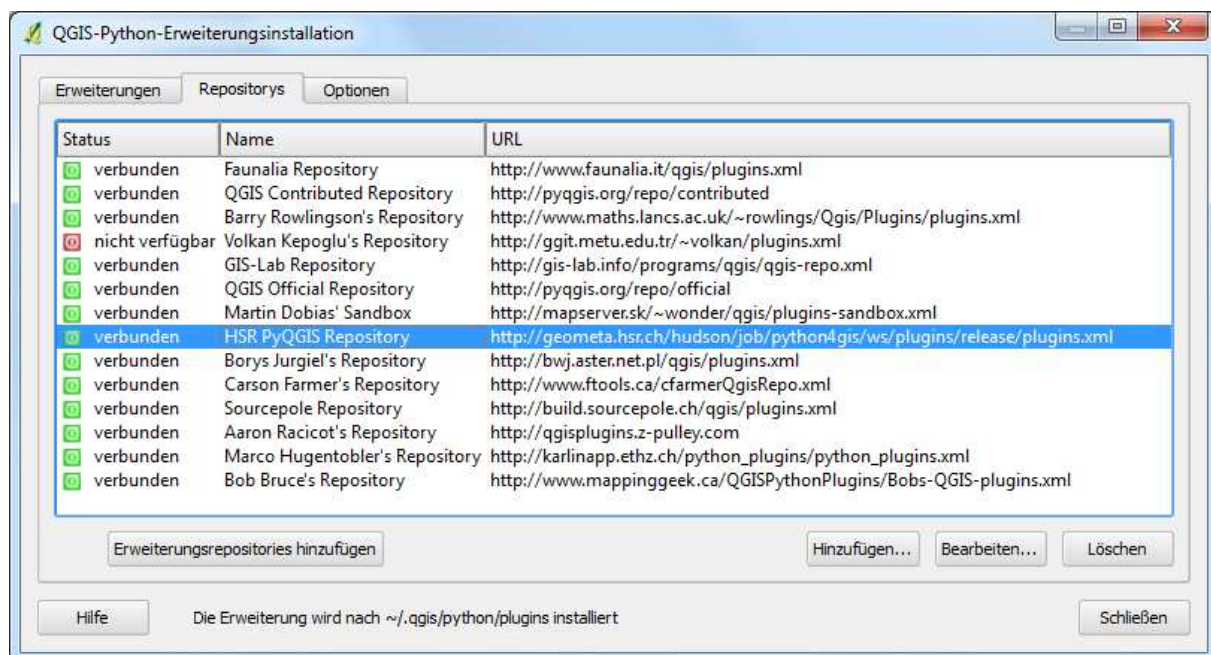


Abbildung 22: Neues Repository wurde eingetragen.

Im Tab „Erweiterungen“ kann nun das NPLCH-Plugin installiert werden.

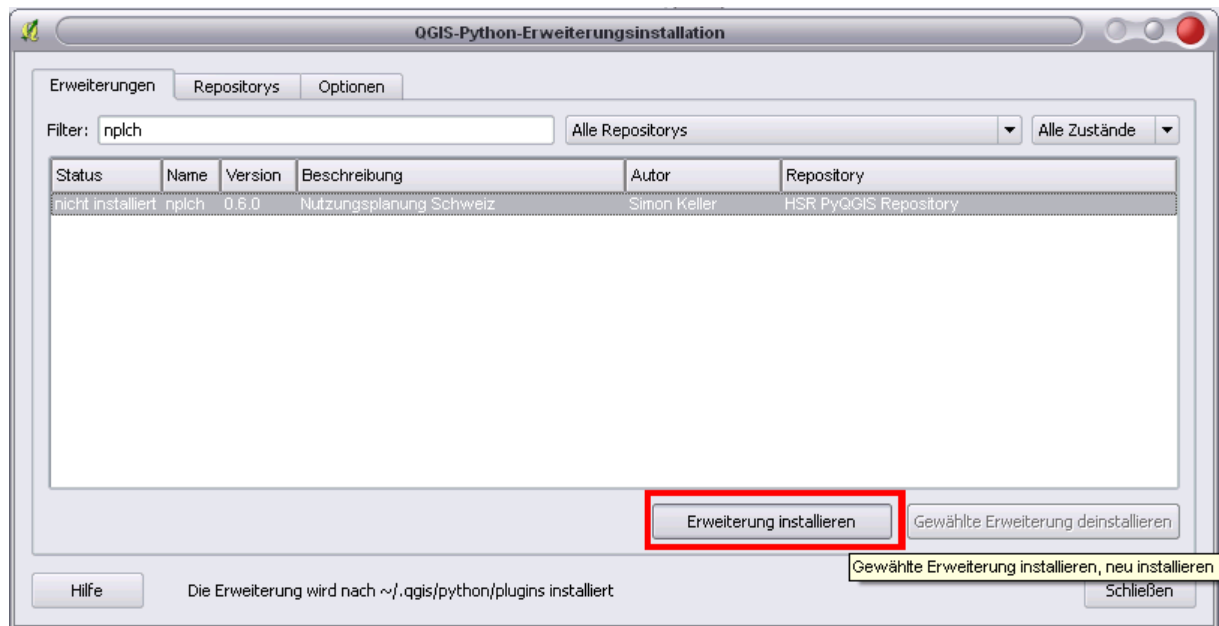


Abbildung 23: NPLCH-Erweiterung installieren.

Hier sehen wir das Ergebnis:

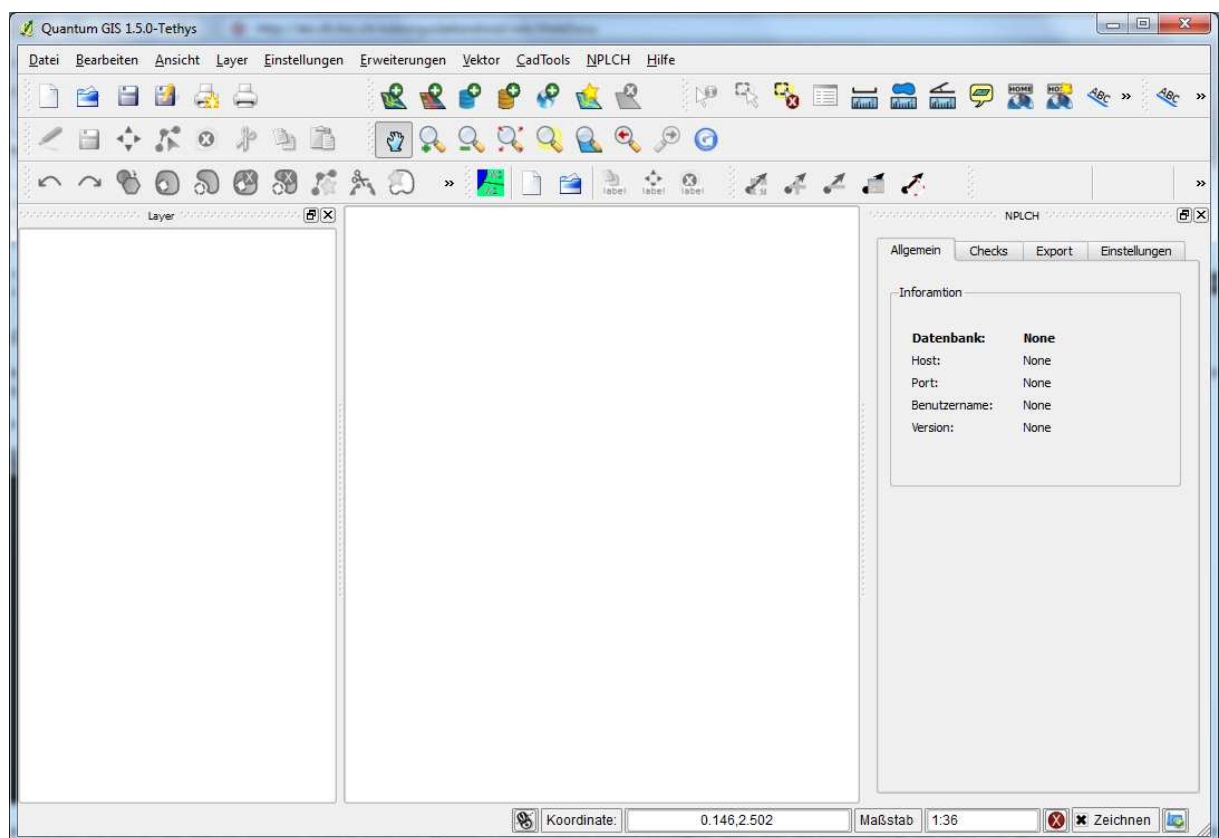


Abbildung 24: QGIS 1.5.0 mit installiertem NPLCH-Plugin.

### 13.3 Konfiguration

Alle nötigen Einstellungen können im NPLCH-DockWidget konfiguriert werden.

Nachfolgend eine Übersicht:

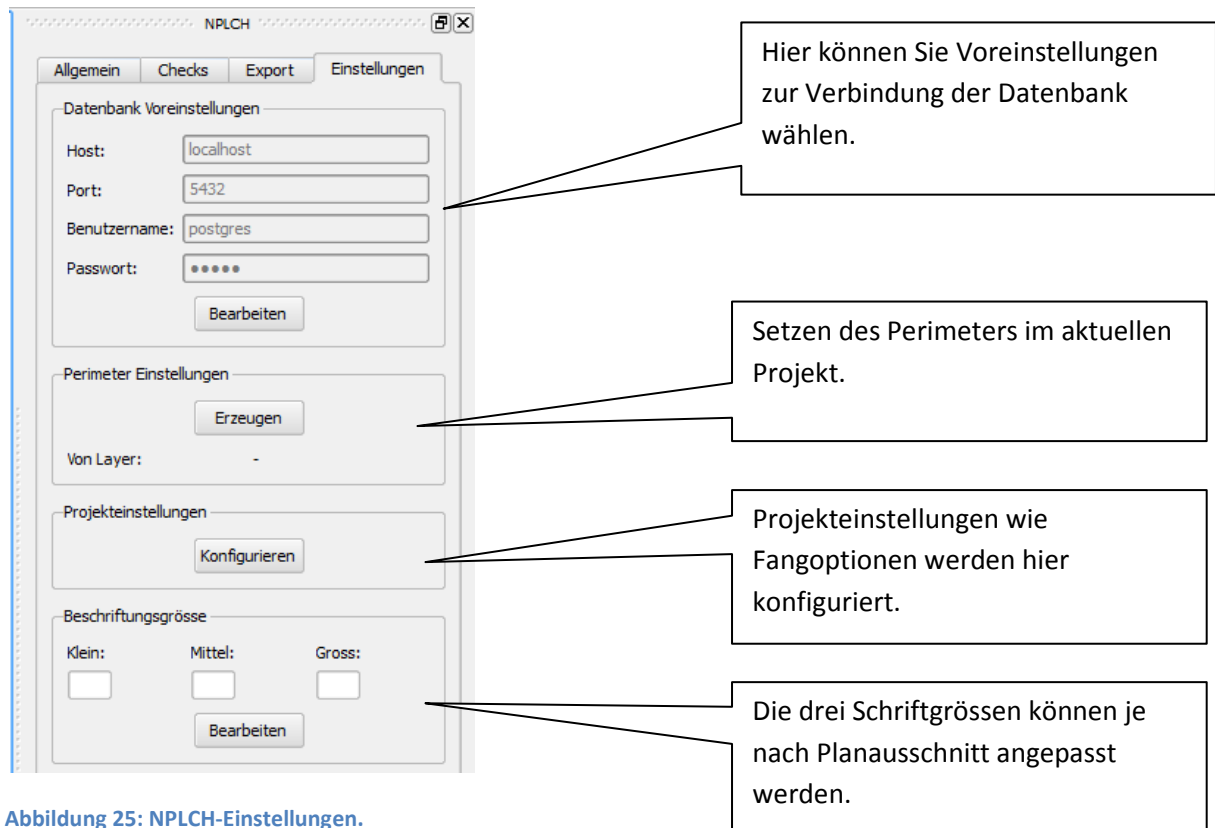


Abbildung 25: NPLCH-Einstellungen.

### 13.4 Anwendung

Nachfolgend werden die wichtigen Abläufe und Funktionen der NPLCH-Applikation erklärt.

#### *Erstellen eines neuen Projektes*

Um ein neues Projekt zu erstellen, drücken Sie auf das entsprechende Icon in QGIS.

Danach öffnet sich folgender Dialog:

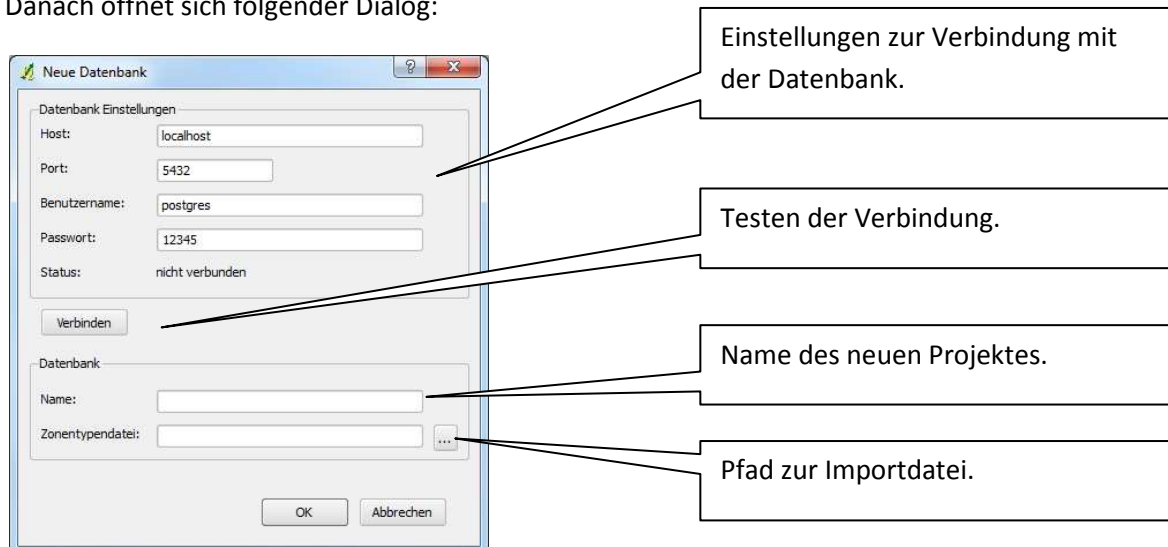


Abbildung 26: Dialog "Neue Datenbank" erstellen.

## Öffnen bestehender Projekte

Um ein bestehendes Projekt zu laden, gibt es zwei Varianten.

Das Projekt wurde in QGIS gespeichert.

- In diesem Fall kann es auch wieder ganz normal in QGIS geladen werden.

Das Projekt wurde nur in der Datenbank gespeichert.

- In diesem Fall kann es über das Icon „Datenbank öffnen“ geladen werden.

Wird das Icon gedrückt, öffnet sich folgender Dialog:

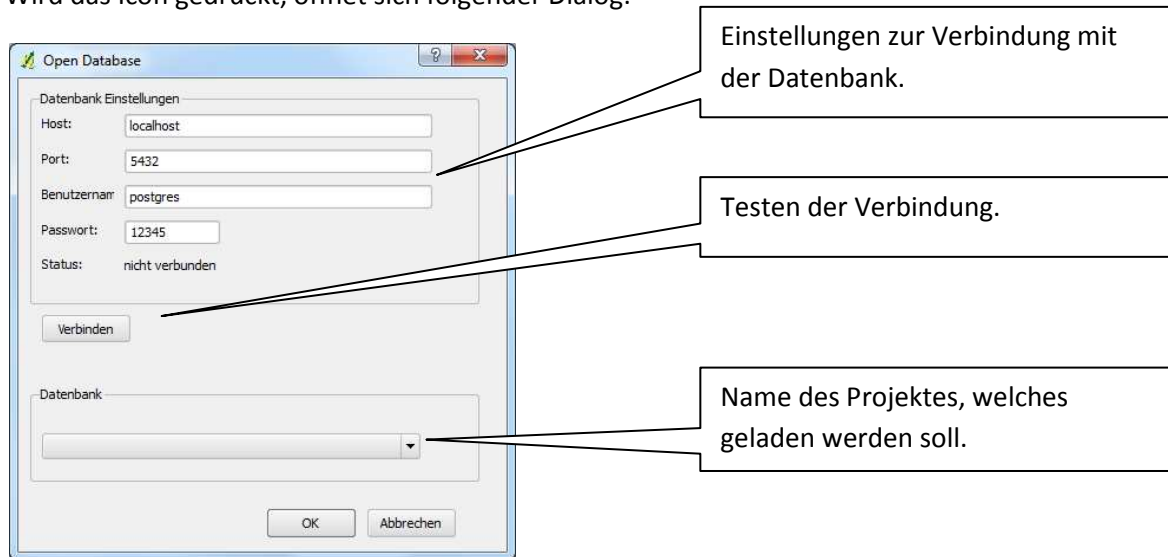


Abbildung 27: Dialog "Datenbank öffnen".

## Erstellung neuer Zonen

Nach dem erstellen neuer Zonen, folgt eine Oberfläche zur Eingabe der Parameter.

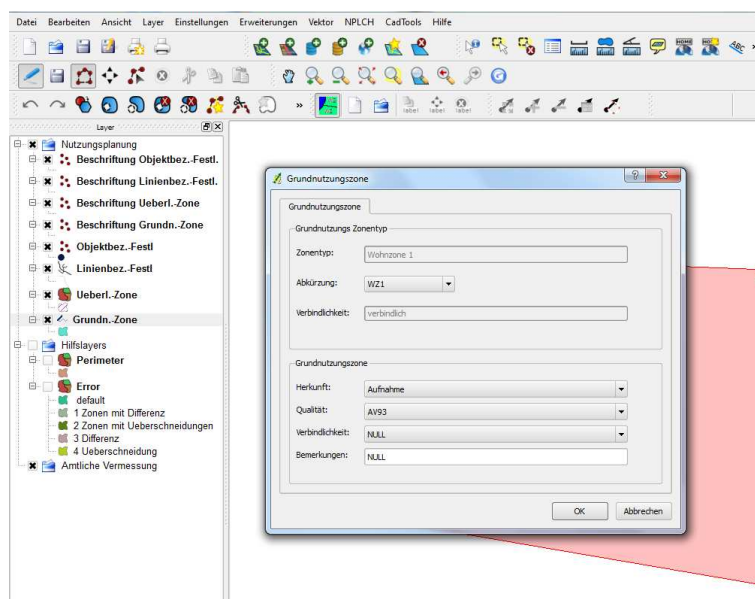


Abbildung 28: Eingabeformulare.



## Bearbeiten der Beschriftungen

Wenn ein Beschriftungs-Layer im „bearbeiten“ Modus ist, werden die Punkte sichtbar und man kann Sie markieren. Danach können die markierten Beschriftungen via den Icons bearbeitet werden.

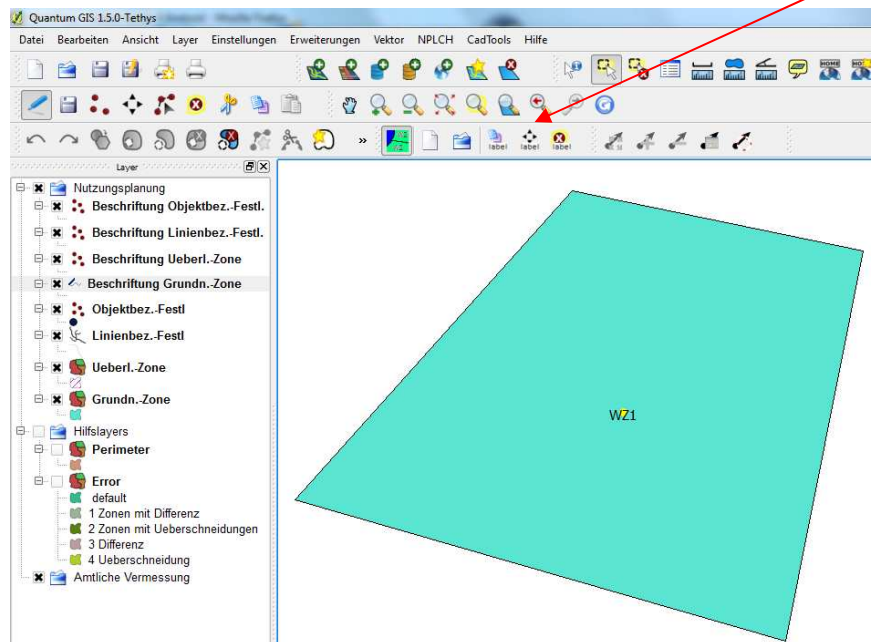


Abbildung 29: Icons zum bearbeiten der Beschriftung.

Über die Aktion „Objektformular bearbeiten“ werden die Einstellungen der Beschriftung gesetzt bzw. geändert.



Abbildung 30: Objektformular bei den Beschriftungen.

## TEIL III: ANHANG

### 14 Inhalt der CD

Der Inhalt der CD gliedert sich wie folgt:

| Verzeichnis            | Inhalt   |
|------------------------|--|
| + bachelorarbeit       | README.txt<br>Abstract.pdf<br>Aufgabenstellung.pdf<br>Bachelorarbeit.pdf |
| + doc_src              | Dokumente und Abbildungen im Originalformat                              |
| + dokumente            | Projektdokumente und Sitzungsprotokolle                                  |
| + kurs                 |  |
| + übungen              | Alle Übungen und Lösungen zum Kurs                                       |
| + nplch                | README.txt, Plugin   |
| + doc                  | API Dokumentation  |
| + sourcen              | Alle Source Dateien  |
| + daten                | Verschiedene Daten zu NPLCH  |
| + simplereloaderplugin | README.txt, Plugin   |
| + sourcen              | Alle Source Dateien  |
| + swissphone           | README.txt, Plugin   |
| + sourcen              | Alle Source Dateien  |



## 15 Glossar und Abkürzungsverzeichnis

| Ausdruck                   | Definition und Erklärung  |
|----------------------------|---|
| <b>QGIS</b>                | Quantum GIS; <a href="http://www.qgis.org">http://www.qgis.org</a>  |
| <b>GIS</b>                 | Geoinformationssysteme  |
| <b>NPLCH</b>               | Nutzungsplanung Schweiz   |
| <b>GIMP</b>                | <i>GNU Image Manipulation Program</i> , freies Bildbearbeitungsprogramm   |
| <b>IDE</b>                 | Entwicklungsumgebung  |
| <b>GUI</b>                 | <b>G</b> raphical <b>U</b> ser <b>I</b> nterface, grafische Benutzeroberfläche  |
| <b>Py</b>                  | Steht für die Programmiersprache Python   |
| <b>Pydev</b>               | Eine Python Entwicklungsumgebung für Eclipse  |
| <b>HSR</b>                 | Hochschule für Technik Rapperswil; <a href="http://www.hsr.ch">http://www.hsr.ch</a>  |
| <b>i18n</b>                | Internationalisierung, Anpassung des Programmes an andere Sprachen und Kulturen.  |
| <b>Open Source Projekt</b> | Quelloffenes Software Projekt. Der Quellcode kann eingesehen und verändert werden.  |
| <b>OEREB</b>               | Öffentlich-rechtlichen Eigentumsbeschränkungen  |
| <b>PostgreSQL</b>          | Datenbankdialekt; Ein objektrelationales Datenbanksystem.   |
| <b>RUP</b>                 | <i>Rational Unified Process</i> ; Vorgehensmodell für die Entwicklung von Software.   |
| <b>SIA</b>                 | Schweizerischer Ingenieur- und Architektenverein  |
| <b>SQL</b>                 | <i>Structured Query Language</i> ; SQL ist eine deklarative Datenbanksprache für relationale Datenbanken.                                     |
| <b>Wiki</b>                | Ein Wiki ist eine Sammlung von Internetseiten, die von den Benutzern nicht nur gelesen, sondern auch in Echtzeit online geändert werden kann. |
| <b>Trac</b>                | Trac ist ein freies, webbasiertes Projektmanagement-Werkzeug; <a href="http://trac.edgewall.org/">http://trac.edgewall.org/</a>               |

## 16 Literaturverzeichnis

### *Publikationen*

Griffiths, David, und Barry Paul. *Head First Programming: A Learner's Guide to Programming Using the Python Language*. O'Reilly Media, 2009.

Martelli, Alex. *Python in a Nutshell*. O'Reilly Media, 2006.

Maruch, Stef, und Aahz Maruch. *Python For Dummies*. WILEY, 2006.

Rempt, Boudewijn. *Gui Programming with PYTHON: Using the Qt Toolkit*. OpenDocs Llc, 2002.

Sherman, Gary. *Desktop GIS: Mapping the Planet with Open Source Tools*. Pragmatic Bookshelf, 2008.

Summerfield, Mark. *Rapid GUI Programming with Python and Qt*. Prentic Hall Press, 2007.

## *Internetquellen*

### OSGeo.org

<http://www.osgeo.org> vom 02.03.2010

### Python

<http://www.python.org> vom 22.02.2010

<http://docs.python.org> vom 05.04.2010

<http://code.google.com/intl/de-DE/edu/languages/google-python-class> vom 15.03.2010

<http://openbook.galileocomputing.de/python> vom 15.03.2010

<http://www.python.org/dev/peps/pep-008> vom 12.04.2010

### Qt

<http://doc.qt.nokia.com/4.6> vom 24.05.2010

### Quantum GIS

<http://www.qgis.org> vom 02.03.2010

<http://blog.qgis.org> vom 02.03.2010

<http://www.qgis.org/de/herunterladen.html> vom 02.03.2010

<http://doc.qgis.org/head/classes.html> vom 10.05.2010

<http://www.qgis.org/de/ueber-qgis.html> vom 07.06.2010

<http://www.qgis.org/de/ueber-qgis/features.html> vom 10.05.2010

[http://www.qgis.org/wiki/Developers\\_corner](http://www.qgis.org/wiki/Developers_corner) vom 10.05.2010

<http://mapserver.sk/~wonder/qgis/html/> vom 05.05.2010

[http://gis.hsr.ch/wiki/Quantum\\_GIS](http://gis.hsr.ch/wiki/Quantum_GIS) vom 12.04.2010

<http://www.qgis.org/community/mailling-lists.html>

### Spatialite

<http://www.gaia-gis.it/spatialite/docs.html> vom 16.04.2010

### SIKULI

<http://sikuli.csail.mit.edu> vom 14.06.2010

### Rahmennutzungsplan

<http://gis.hsr.ch/wiki/Rahmennutzungsplanung> vom 12.04.2010

## 17 Kursübungen

## 18 Sitzungsprotokolle

Die Sitzungsprotokolle sind auf dem Trac einsehbar. <http://dev.ifs.hsr.ch/python4gis/wiki/Meetings>

## 19 Eigenständigkeitserklärung

Ich erkläre hiermit,

- dass ich die vorliegende Arbeit selber und ohne fremde Hilfe durchgeführt habe, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit dem Betreuer schriftlich vereinbart wurde,
- dass ich sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitier regeln korrekt angegeben habe.

Ort, Datum:

Name, Unterschrift: