

Kort Reloaded – A Gamified App for Collecting OpenStreetMap Data

Bachelorarbeit

Abteilung Informatik

HSR Hochschule für Technik Rapperswil

Frühjahrssemester 2016

Autoren:	Marino Melchiori Dominic Mülhaupt
Betreuer:	Prof. Stefan Keller
Projektpartner:	Jürg Hunziker Stefan Oderbolz Liip AG
Experte:	
Datum:	

Impressum

Autoren:	Marino Melchiori (mmelchio@hsr.ch) Dominic Mülhaupt (dmuelhau@hsr.ch)
Dokument erstellt:	10.03.2016
Letzte Aktualisierung:	

Dieses Dokument wurde mit L^AT_EX erstellt.

Management Summary

Ausgangslage

Das *OpenStreetMap*-Projekt beinhaltet eine sehr grosse Menge an Daten, welche frei zugänglich ist. Für die Pflege dieser Daten ist es daher naheliegend, auf unterstützende Software zurückzugreifen. Zu diesem Zweck gibt es eine Reihe von Applikationen, welche sich grob in zwei Kategorien einteilen lassen: Editoren und Tools zur Qualitätssicherung.

Mit den Editoren lässt sich direkt oder indirekt die *OpenStreetMap*-Karte verändern und ergänzen. Die Qualitätssicherungstools haben sich zum Ziel gesetzt, fehlende oder falsche Daten aufzuspüren. Diese werden dann entweder automatisch korrigiert oder übersichtlich dargestellt, um eine manuelle Korrektur zu ermöglichen. Ein solcher Editor ist zum Beispiel die *Kort*¹-App, welche in Form einer Bachelorarbeit von Jürg Hunziker und Stefan Oderbolz im Herbstsemester 2012/13 realisiert wurde.

Kort ist eine gamified cross-platform [Web-App](#) KORT für die Verbesserung von *OpenStreetMap*-Daten.

Einige Tools wie *KeepRight*² oder *Osmose*³ berechnen aus den Karten-Rohdaten die vorhandenen Fehler. Dazu werden einige Heuristiken verwendet oder einfache Plausibilitätsprüfungen durchgeführt. Typische Fehler aus diesen Quellen sind [POIs](#) ohne Namen oder Strassen ohne definierte Geschwindigkeitslimiten.

Ergebnisse

Frontend

Ausblick

¹<http://play.kort.ch/>

²<http://keepright.ipax.at/>

³<http://osmose.openstreetmap.fr/map/>

Inhaltsverzeichnis

I. Technischer Bericht	1
1. Einführung	2
1.1. Problemstellung, Vision	2
1.2. Ziele	2
1.3. Rahmenbedingungen, Umfeld, Definitionen, Abgrenzungen	2
1.4. Vorgehen, Aufbau der Arbeit	2
2. Stand der Technik	3
2.1. Bestehende Lösungsansätze und Normen	3
2.2. Kurzbeschreibung und Charakterisierung	3
2.3. Defizite	3
3. Bewertung	4
3.1. Kriterien	4
3.2. Schlussfolgerungen	4
4. Umsetzungskonzept	5
5. Resultate	6
5.1. Zielerreichung	6
5.2. Ausblick	6
5.3. Persönliche Berichte	6
5.4. Dank	6
II. Projektdokumentation	7
6. Vision	8
7. Anforderungsspezifikation	9
7.1. Anforderungen an die Arbeit	9
7.2. Use Cases	9
7.3. System-Sequenzdiagramme	9
7.4. Weitere Funktionen	9
7.5. Nicht-funktionale Anforderungen	9
7.6. Detailspezifikation	9

8. Analyse	10
8.1. Domain Modell, Klassendiagramme	10
8.2. Objektkatalog	10
9. Design	11
9.1. Architektur	11
9.2. Objektkatalog	11
9.3. Package- und Klassendiagramme	11
9.4. Sequenzdiagramm, UI Design	11
10. Implementation	12
11. Testing	13
12. Resultate und Weiterentwicklung	14
12.1. Resultate	14
12.2. Möglichkeiten der Weiterentwicklung	14
12.2.1. Vorgehen	14
13. Software-Dokumentation	15
13.1. Installation	15
13.2. Tutorial	15
13.3. Referenzhandbuch	15
III. Projektmanagement	16
14. Projektmanagement	17
14.1. Prototypen, Releases, Meilensteine	17
14.2. Team, Rollen, Verantwortlichkeiten	17
14.3. Aufwandschätzung, Zeitplan, Projektplan	17
14.4. Risiken	17
14.5. Prozessmodell	17
15. Projektmonitoring	18
15.1. Soll-Ist-Zeitvergleich	18
15.2. Code-Statistik	18
IV. Anhänge	19
Abbildungsverzeichnis	21
Tabellenverzeichnis	22

Teil I.

Technischer Bericht

1. Einführung

1.1. Problemstellung, Vision

1.2. Ziele

1.3. Rahmenbedingungen, Umfeld, Definitionen, Abgrenzungen

1.4. Vorgehen, Aufbau der Arbeit

2. Stand der Technik

2.1. Bestehende Lösungsansätze und Normen

2.2. Kurzbeschreibung und Charakterisierung

2.3. Defizite

3. Bewertung

3.1. Kriterien

3.2. Schlussfolgerungen

4. Umsetzungskonzept

5. Resultate

5.1. Zielerreichung

5.2. Ausblick

5.3. Persönliche Berichte

5.4. Dank

Teil II.

Projektdokumentation

6. Vision

7. Anforderungsspezifikation

7.1. Anforderungen an die Arbeit

Anforderungsspezifikation

7.2. Use Cases

7.3. System-Sequenzdiagramme

7.4. Weitere Funktionen

7.5. Nicht-funktionale Anforderungen

7.6. Detailspezifikation

8. Analyse

8.1. Domain Modell, Klassendiagramme

8.2. Objektkatalog

9. Design

9.1. Architektur

9.2. Objektkatalog

9.3. Package- und Klassendiagramme

9.4. Sequenzdiagramm, UI Design

10. Implementation

11. Testing

12. Resultate und Weiterentwicklung

12.1. Resultate

12.2. Möglichkeiten der Weiterentwicklung

12.2.1. Vorgehen

13. Software-Dokumentation

13.1. Installation

13.2. Tutorial

13.3. Referenzhandbuch

Teil III.

Projektmanagement

14. Projektmanagement

14.1. Prototypen, Releases, Meilensteine

14.2. Team, Rollen, Verantwortlichkeiten

14.3. Aufwandschätzung, Zeitplan, Projektplan

14.4. Risiken

14.5. Prozessmodell

15. Projektmonitoring

15.1. Soll-Ist-Zeitvergleich

15.2. Code-Statistik

Teil IV.

Anhänge

Abbildungsverzeichnis

Tabellenverzeichnis