

Masterarbeit

Konzipierung eines Verfahrens zur automatisierten Erzeugung von Test-Eingaben für die Qualitätssicherung der Interaktion von Geodiensten und -datenbanken

Dennis Lampert

Matr. 4383358

Erstgutachter: Prof. Dr. Ina Schieferdecker Zweitgutachter: Prof. Dr. Katinka Wolter

Betreuer: Dr. Niels Lohmann

	•	•
1	1	1

Ich versichere, dass ich die vorliegende Arbeit selbständig ver angegebenen Quellen und Hilfsmittel benutzt habe. Alle Stelle aus veröffentlichten Schriften entnommen wurden, sind al Zeichnungen oder Abbildungen sind von mir selbst erstellt wo Quellennachweisen versehen. Diese Arbeit ist in gleicher oder Prüfungsbehörde eingereicht worden.	en, die wörtlich oder sinngemäß ls solche gekennzeichnet. Die orden oder mit entsprechenden
Berlin, den 29. Februar 2016	Dennis Lampert

_				
⊢	rei	α	h	Δ
	ıcı	Ľ۵	v	ᆫ
	_	0		_

Diese Masterarbeit enthält keine internen und vertraulichen Daten der Carmeq GmbH. Diese Arbeit darf Dritten somit ohne ausdrückliche Zustimmung des Unternehmens zugänglich gemacht werden.

Berlin, den 29. Februar 2016

Dr. Niels Lohmann Betreuer Carmeq GmbH Peter Ehlers Business Team Manager

Zusammenfassung

Es gibt eine Vielzahl an Geodiensten wie Google Maps oder Here, welche dem Nutzer die Möglichkeit bieten, nach Adressen und Routen zu suchen. Nach der Eingabe einer Adresse werden dem Anwender ein oder mehrere Ergebnisse präsentiert. Für die Bewertung der Qualität der Dienste können Integrationstests verwendet werden, um einerseits den Datenbestand und andererseits die Erkennung der Eingaben zu testen. Diese Tests sind zudem nützlich um Fehler aufzudecken, wenn sich eine der Komponenten, also die Datenbank oder die Software ändert.

Die Tests können automatisiert durchgeführt werden. Dafür sind Testfälle nötig, die zunächst erzeugt werden müssen. Es wäre daher sinnvoll die Testfallerzeugung ebenfalls zu automatisieren. Dafür wird ein Generator benötigt, welcher unterschiedliche Strategien und Möglichkeiten bietet, um Testfälle für bestimmte Regionen zu erzeugen. Hierbei ist es zudem wichtig verschiedene Formate für die Testfalleingaben zu unterstützen. Solch ein Generator kann anschließend zum Testen von Online-Geodiensten oder beispielsweise Navigationssysteme, sowie deren Eingabesystemen verwendet werden.

In dieser Arbeit wird ein Testfallgenerator konzipiert, welcher es ermöglicht Testfälle automatisiert zu erzeugen. Der Generator wird anschließend in einer Anwendung umgesetzt. Um die Tauglichkeit eines solchen Generators zu untersuchen, werden verschiedene Testszenarien aufgestellt, welche anschließend in einer Fallstudie für unterschiedliche Geodienste untersucht und verifiziert werden.

Inhaltsverzeichnis

1.	Einfi	ührung		1
2.	Hint	ergrun	d	3
	2.1.	Qualit	ätssicherung	3
		2.1.1.	Qualität	3
		2.1.2.	Konstruktive Qualitätssicherung	4
		2.1.3.	Analytische Qualitätssicherung	5
		2.1.4.	Probleme der Qualitätssicherung: Komponenten verschiedener Hersteller	5
	2.2.	Softwa	retesten	6
		2.2.1.	Begriffe	6
		2.2.2.	Testeingaben	7
		2.2.3.	Blackbox-Testentwurfsverfahren	7
		2.2.4.	Systemintegrationstest	8
	2.3.	Geodie	enste	9
	2.4.	Adress	sen	9
		2.4.1.	Adresselemente	9
		2.4.2.	Formate und Modelle	10
	2.5.	-	Tiling	10
	2.6.		nde zwischen Punkten in den Tiles	12
	2.7.	Naviga	ation Data Standard	14
3.	Geo	dienste	und Geodatenbanken	15
	3.1.	Geodie	enste	15
		3.1.1.	Anforderungen	15
		3.1.2.	Vorstellung: Geodienste	16
		3.1.3.	Geocoding	17
		3.1.4.	Softwaretestbegriffe in Bezug auf Geodienste	17
		3.1.5.	Testfälle für Geodienste	18
		3.1.6.	Testen von Geodiensten	20
		3.1.7.	Blackbox-Verfahren bei Geodiensten	21
		3.1.8.	Überdeckungsmaße	23
	3.2.	Geodie	enst-Komponenten	24
		3.2.1.	Geocoding-Software	24
		3.2.2.	Geodatenbanken	24
		3 9 3	Analysa dar Datanbankan	25

x Inhaltsverzeichnis

		3.2.4. Bewertung der Datenbanken	
	3.3.	Eingabesprachen	3
4.	Verf	ahren zur Erzeugung von Testeingaben 31	L
	4.1.	Testgenerator	L
		4.1.1. Überblick	L
		4.1.2. Umsetzung	2
	4.2.	Datenbank Hintergrundwissen	1
	4.3.	Regionen Hintergrundwissen	1
	4.4.	Bildung der Äquivalenzklassen	5
		4.4.1. K-Means-Algorithmus	5
		4.4.2. Überführung der Adressdaten in einen Vektor	3
	4.5.	Testfall-Verteilungsstrategien	3
		4.5.1. Zufällige Verteilung	9
		4.5.2. Geografische Verteilung	9
		4.5.3. Geografische Verteilung mit Einbeziehen der Testfalldichte 44	1
		4.5.4. Zwischenfazit	5
	4.6.	Mehrfach auftretende Straßen	3
	4.7.	Erzeugte Eingaben in unterschiedlichen Diensten	7
5.	Falls	studie 49	9
•		Durchführung der Tests in Geodiensten	
		Testszenarien	
		5.2.1. Szenario 1: Zufällige und geografische Verteilung: Deutschland 51	
		5.2.2. Szenario 2: Zufällige und geografische Verteilung: Alaska	2
		5.2.3. Szenario 3: Verteilung mit Einbeziehen der Testfalldichte: Alaska 53	
		5.2.4. Szenario 4: Äquivalenzklassentest: Berlin	1
		5.2.5. Szenario 5: Mehrfach vorkommende Straßennamen: Berlin 55	5
		5.2.6. Szenario 6: Sprachen: diakritische Zeichen: Prag	5
		5.2.7. Szenario 7: Sprachen: kyrillisch: Moskau	5
		5.2.8. Szenario 8: Sprachen: griechisch: Athen	
		5.2.9. Szenario 9: Zufällige und geografische Verteilung: Australien 50	3
	5.3.	Testauswertung	7
	5.4.	Testergebnisse	7
		5.4.1. Szenario 1: Zufällige und geografische Verteilung: Deutschland 58	3
		5.4.2. Szenario 2: Zufällige und geografische Verteilung: Alaska 60)
		5.4.3. Szenario 3: Verteilung mit Einbeziehen der Testfalldichte: Alaska 63	3
		5.4.4. Szenario 4: Äquivalenzklassentest: Berlin 64	1
		5.4.5. Szenario 5: Mehrfach vorkommende Straßennamen: Berlin 67	7
		5.4.6. Szenario 6: Sprachen: diakritische Zeichen: Prag 68	3
		5.4.7. Szenario 7: Sprachen: kyrillisch: Moskau	
		5.4.8. Szenario 8: Sprachen: Griechisch: Athen)
		5.4.9. Szenario 9: Zufällige und geografische Verteilung: Australien 71	1
	5.5.	Bewertung der Methode und des Werkzeuges	3
		5.5.1. Probleme bei der Testdurchführung	5
	5.6.	Testbewertung der Geodienste	3
	5.7.	Beziehungen zwischen Eingabe und Ausgabe	7

Inhaltsverzeichnis xi

	5.8.	Gütebestimmung	78
6.	Fazit		81
-		Ausblick	82
Lit	eratu	rverzeichnis	83
Α.	Geo	dienstdaten und -resultate	85
	A.1.	In OSM gespeicherte Daten	85
	A.2.	REST API Ergebnisse	86
		A.2.1. Google Maps API Ergebnis	86
		A.2.2. Here API Ergebnis	88
		A.2.3. Bing Maps API Ergebnis	90
		A.2.4. OSM API Ergebnis	92
В.	Test	ergebnisse und Auswertung	95
	B.1.	Szenario 1: Politische und geografische Verteilung: Deutschland	96
	B.2.	Szenario 2: Politische und geografische Verteilung: Alaska	98
	B.3.	Szenario 3: Verteilung mit Einbeziehen der Testfalldichte: Alaska	102
	B.4.	Szenario 4: Äquivalenzklassentest: Berlin	104
	B.5.	Szenario 5: Mehrfach vorkommende Straßennamen: Berlin	105
	B.6.	Szenario 6: Sprachen: diakritische Zeichen: Prag	106
		Szenario 7: Sprachen: kyrillisch: Moskau	107
	B.8.	Szenario 8: Sprachen: Griechisch: Athen	108
	B.9.	Szenario 9: Politische und geografische Verteilung: Australien	109
Αb	bildu	ngsverzeichnis	113
Та	bellei	nverzeichnis	114
Qι	iellco	deverzeichnis	116

KAPITEL 1

Einführung

Geodienste wie Google Maps oder Here ermöglichen Anwendern die Suche nach Adressen, Sonderzielen (POI) und Routen. Die Anbieter der Dienste ermöglichen ihren Kunden Zugriff auf den Service über verschiedene Plattformen, beispielsweise Browseranwendungen oder Applikationen für Smartphones und Tablets. Der Nutzer erwartet nach der Eingabe eines Begriffes oder einer mitunter unvollständigen Adresse ein Ergebnis in Form von Adressdaten, sowie eine Visualisierung der Eingabe auf einer Karte. Jedoch kann es vorkommen, dass der Dienst die Eingabe des Anwenders nicht findet. Dann stellt sich die Frage wie vollständig die Daten des Geodienstes sind und wie gut der Dienst Eingaben interpretieren und finden kann. Sind die Informationen nicht in der Datenbank des Dienstes hinterlegt, können diese nicht gefunden werden. Jedoch kann es auch sein, dass die Eingabe des Anwenders nicht ausreichend genau war und der Dienst deshalb kein Ergebnis findet. Eine Qualitätssicherung der Geodienste und deren Anwendung kann helfen die Qualität des Dienstes zu beurteilen.

Eine Möglichkeit ist der Softwaretest als Teil der analytischen Qualitätssicherung. Der Testvorgang kann durch eine Automatisierung schneller, systematischer sowie effizienter durchgeführt werden und vereinfacht die Qualitätssicherung. Neben der automatisierten Ausführung der Tests ist es zudem sinnvoll die Testerzeugung zu automatisieren. Da real existierende Adressen benötigt werden, welche die Dienste finden sollen, muss eine Adressdatenbank für die Erzeugung verwendet werden. Eine automatisierte Testfallerzeugung ermöglicht es einerseits die Testfälle schneller zu erzeugen. Andererseits liefert ein automatisierter Prozess eine bessere Abdeckung und kann "Extremfälle" abdecken, welche eine manuelle Erzeugung durch eine Person eventuell nicht hervorgebracht hätte und welche sonst nicht aufgefallen wären. Es ist also zu klären ob solch eine automatisierte Erzeugung von Testfällen für Geodienste umsetzbar ist.

Zu diesem Zweck wird in der Arbeit ein Testgenerator für Geodienste konzipiert. Der Generator wird anschließend in einer Anwendung umgesetzt. Damit ist es dann möglich einen Integrationstest zwischen der Geocoding-Software und der dahinter liegenden Datenbank durchzuführen. Der Geodienst und dessen Datenbank wird dabei als Blackbox betrachtet, da keinerlei Information über die Implementierung bekannt sein muss. Der Generator kann zudem dazu genutzt werden weitere Systeme, wie Navigationssysteme und deren Eingabesysteme zu testen. Solch ein Test ist sinnvoll, da die einzelnen Komponenten (Software und Datenbank) unter Umständen von unterschiedlichen Herstellern entwickelt werden.

2 1. Einführung

Die Konzipierung eines solchen Generators umfasst neben dem allgemeinen Aufbau die Unterstützung mehrerer Erzeugungsstrategien für Testfälle. Die Strategien sollen es ermöglichen die Geodienste für unterschiedliche Szenarien zu testen, sodass für alle Regionen geeignete Strategien existieren, um eine sinnvolle Verteilung der Testfälle zu gewährleisten. Die Strategien können beispielsweise verschiedene Ansätze für die geografische Verteilung der Testfälle beschreiben, um eine möglichst hohe Abdeckung zu erhalten. Des Weiteren soll es Möglichkeiten geben unterschiedliche Strukturen in den Testfalleingaben (z.B. besonders lange Eingaben oder Sonderzeichen) zu erkennen und diese gezielt zu testen. Das ermöglicht es die Systeme als Blackbox zu betrachten und lediglich über die Eingabe zu steuern. Aus den Erzeugungsstrategien und Generierungsmöglichkeiten werden anschließend Szenarien erstellt und in einer Fallstudie untersucht. Dazu sollen mehrere Online-Geodienste für die Tests in der Studie verwendet werden. Abschließend werden die getesteten Dienste sowie die Funktionalität des Generators bewertet.

Die Arbeit soll also gezielt unterschiedliche Verteilungen und Erzeugungsmöglichkeiten für einen Testgenerator für Geodienste entwickeln und untersuchen. Bei der Recherche fiel auf, dass in bisherigen Arbeiten solch ein Generator noch nicht behandelt wurde. Somit gibt es zu dieser Thematik auch keine vergleichenden Tests der verschiedenen Dienste.

Der weitere Verlauf der Arbeit ist wie folgt strukturiert: Kapitel 2 gibt einen Überblick über Qualitätssicherung und Softwaretesten. Des Weiteren werden Geodienste und Adressgrundlagen erklärt. Außerdem wird erklärt, wie sich eine Karte mit dem Map-Tiling-Verfahren unterteilen lässt. In Kapitel 3 werden unterschiedliche Geodienste näher erläutert und die Aspekte der Qualitätssicherung und des Softwaretestens bezüglich der Geodienste erklärt. Das Kapitel 4 stellt Konzepte eines Testgenerators vor. Dafür werden unterschiedliche Erzeugungsstrategien für Testfälle vorgestellt und bewertet. Im Kapitel 5 werden die Strategien des umgesetzten Testgenerators verwendet um mehrere Geodienste in verschiedenen Szenarien zu testen. Diese werden verglichen und bewertet. Die Arbeit endet mit einer Zusammenfassung der gewonnenen Erkenntnisse in Kapitel 6.

KAPITEL 2

Hintergrund

Dieses Kapitel gibt einen Überblick über grundlegende Aspekte der Qualitätssicherung und geht dabei insbesondere auf die Elemente ein, welche für die Qualitätssicherung zwischen Geodiensten und -datenbanken relevant sind. Des Weiteren wird das Thema Softwaretesten näher erläutert. Nachfolgend werden weitere Gesichtspunkte bezüglich Geodiensten erklärt.

2.1 Qualitätssicherung

Um eine qualitativ hochwertige Software zu entwickeln, ist es nötig Qualitätssicherung zu betreiben. Ziel der Qualitätssicherung ist es, sämtliche Fehler in einem Produkt vor der Auslieferung an einen Kunden zu finden und zu beheben. Der Qualitätsbegriff soll in diesem Abschnitt erläutert werden. Zudem werden verschiedene Methoden zur Qualitätssicherung beschrieben.

2.1.1 Qualität

In der ISO 9000 wird die Qualität eines Produktes oder einer Leistung als Fähigkeit beschrieben einen Kunden zufrieden zu stellen.

The quality of an organization's products and services is determined by the ability to satisfy customers and the intended and unintended impact on relevant interested parties. [1]

Weitere Arbeiten beschreiben ebenfalls die Kundenzufriedenheit als zentrales Kriterium für die Qualität eines Produktes [2]. Dabei kann es auch sein, dass Spezifikationen nicht erreicht wurden, der Kunde jedoch trotzdem zufrieden ist und somit ein für ihn qualitativ hochwertiges Produkt erhält. Es nützt dem Kunden jedoch nichts, wenn ihm ein Produkt geliefert wird, was sämtliche zuvor festgelegten Spezifikationen umgesetzt hat, ihm aber nicht gefällt [3]. Dieses Problem kann auftreten, wenn sich die Spezifikation im Entwicklungsprozess plötzlich ändern, da sich die zuvor spezifizierten Anforderungen als untauglich herausgestellt haben.

Die Qualität einer Software kann jedoch auch anhand gefundener Fehler nach der Durchführung von Softwaretests gemessen werden. Hierbei gibt es jedoch kein einheitliches Maß, welches verwendet werden kann, um zu bestimmen, welche Fehlerrate zu welcher Qualitätsstufe führt. Die Qualität kann jedoch durch Softwaretesten indirekt erhöht werden, da vor der Auslieferung Fehler gefunden und behoben werden können.

Die Softwarequalität wird anhand wichtiger Qualitätsmerkmale gemessen. Diese sind nach der ISO 9126: Funktionalität, Zuverlässigkeit, Benutzbarkeit, Effizienz, Änderbarkeit, Übertragbarkeit [4]. Diesen Merkmalen unterliegen hierarchische Subkriterien. Am Ende finden sich messbare Attribute, deren Werte mit Hilfe von Metriken ermittelt werden können. Die Merkmale sind in Abbildung 2.1 aufgeführt.

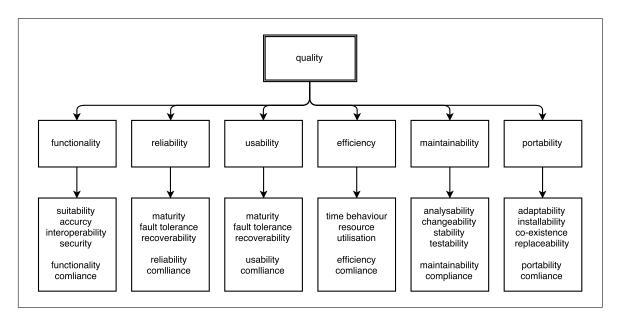


Abbildung 2.1.: Qualitätsmodell nach ISO 9126-1 [4]

2.1.2 Konstruktive Qualitätssicherung

Konstruktive Qualitätssicherung beschreibt den Ansatz, bereits vor der eigentlichen Arbeit und auch bei der Arbeit vorbeugende Maßnahmen zu ergreifen, um die Qualität des späteren Produktes zu erhöhen. Dazu gehört die allgemeine Vorgehensweise in einem Softwareprojekt, welche sorgfältig durchgeführt werden muss. Der Konstruktionsprozess wird so gestaltet, dass Qualitätsmängel seltener werden. Zudem sollen Ursachen für entdeckte Mängel entfernt werden. Die Bestimmung der richtigen Werkzeuge, das Nutzen von Standards und Coding-Guidelines, welche den Programmstil vorgeben und helfen Fehler zu vermeiden, sind einige Maßnahmen für konstruktive Qualitätssicherung. Im Grunde zählt jedoch alles hierzu, was die Entwickler unterstützt und somit eine positive Auswirkung auf das Produkt hat.

Allgemeine Ziele der konstruktiven Qualitätssicherung sind zudem sämtliche Merkmale von Qualität, welche bereits in Abschnitt 2.1.1 genannt wurden.

2.1.3 Analytische Qualitätssicherung

In der analytischen Qualitätssicherung werden prüfende Maßnahmen auf ein Produkt angewendet, um dessen Qualität zu erhöhen. Hierbei wird versucht Fehler und Mängel aufzudecken, nachdem sie entstanden sind. Dabei kann die Überprüfung auf ein Teilprodukt oder das fertige Produkt durchgeführt werden. Im besten Fall wird die Untersuchung der (Teil-)Produkte so früh wie möglich begonnen, um Mängel und Defekte zeitig aufzudecken und um diese zu beheben. Die analytische Qualitätssicherung wird in statische (auch analysierende) und dynamische (auch testende) Verfahren aufgeteilt.

Statisches Verfahren

Zu den statischen oder analysierenden Verfahren gehören vor allem Durchsichten (Reviews) und Inspektionen. Hierbei werden die Testobjekte analysiert. Die Testobjekte können dabei sämtliche Dokumente sein, welche mit dem Softwareentwicklungsprozess in Verbindung stehen.

Bei den Durchsichten werden beispielsweise Dokumente, Anforderungen und Testfälle untersucht. Auf den Quellcode können Inspektionen durchgeführt werden. Ziel ist es Mängel frühzeitig zu entdecken, zusammenzutragen um sie anschließend zu beseitigen. Daraufhin erfolgt eine erneute Prüfung. Es gibt auch automatische Analyseverfahren, welche mit Hilfe von Werkzeugen durchgeführt werden, um formale Dokumente gegen entsprechende Regeln zu überprüfen. Ein Analysewerkzeug kann beispielsweise ein Compiler sein, welcher erkennt, wenn Fehler in der Syntax eines Programmes vorliegen und diese dem Entwickler meldet.

Dynamisches Verfahren

Das dynamische Verfahren der analytischen Qualitätssicherung wird auch dynamisches Testen genannt. Darunter wird das Ausführen eines Testobjektes mit zuvor definierter Eingabe verstanden. Informationen zum Testen von Software befinden sich in Abschnitt 2.2.

2.1.4 Probleme der Qualitätssicherung: Komponenten verschiedener Hersteller

Bei der Qualitätssicherung können jederzeit Probleme und Verzögerungen auftreten. Es kann beispielsweise sein, dass verschiedene Komponenten von unterschiedlichen Herstellern entwickelt und bereit gestellt werden. Da jeder Hersteller seine eigene Entwicklungszeit benötigt, können die Komponenten mitunter erst sehr spät zusammengeführt und somit auch erst spät im Zusammenspiel getestet werden.

Die Komponenten einzelner Hersteller müssen in einem Systemintegrationstest überprüft werden. Liefert einer der Hersteller ein Update für seine Komponente, kann es sein, dass die Interaktion mit den anderen Komponenten nicht mehr korrekt funktioniert, auch wenn der Hersteller seine Komponente im Einzelnen ausreichend überprüft und getestet hat.

2.2 Softwaretesten

Softwaretesten ist ein wichtiger Teil der Qualitätssicherung von Software. Softwaretests sind nötig, um Fehler und Mängel einer Software aufzudecken, welche zu einem fehlerhaften Verhalten der Software führen. Zudem ist das Testen eine Möglichkeit die Qualität einer Software zu messen. Außerdem erhöhen Softwaretests das Vertrauen in die Software, wenn keine oder nur weniger Fehler gefunden wurden [5].

2.2.1 Begriffe

Im folgenden Abschnitt werden grundlegende Begriffe und Vorgehensweisen des Softwaretestens erklärt.

Fehler: Ein Fehler in einer Software bedeutet die Nichterfüllung einer zuvor festgelegten Anforderung. Es handelt sich um eine Abweichung zwischen dem Ist-Verhalten, welches während der Ausführung festgestellt wird, und dem Soll-Verhalten, welches vor der Ausführung spezifiziert wurde [6].

Mangel: Ein Mangel in einer Software bedeutet, dass eine Anforderung nicht angemessen erfüllt wurde [6].

Diese Fehler und Mängel bestehen schon zum Zeitpunkt der Fertigstellung der Software. Sie kommen jedoch erst beim Gebrauch der Software zum Tragen. Der Fehlerbegriff kann jedoch weiter differenziert werden als Error (Fehlerhandlung), Fault (Fehlerzustand) und Failure (Fehlerwirkung) [2] [6].

Error (Fehlerhandlung): Der Error wird durch eine Person hervorgerufen. Dies kann beispielsweise ein Entwickler sein, der durch einen Tippfehler beim Programmieren einen Fault (Fehlerzustand) in der Software verursacht.

Fault (Fehlerzustand): Durch einen Fault kann es sein, dass ein System nicht mehr, oder nur noch fehlerhaft ausgeführt wird. Das ist beispielsweise ein Defekt (innerer Fehlerzustand) in einer Komponente eines Programmes. Der Fault kann einen Failure verursachen.

Failure (Fehlerwirkung): Der Failure ist die Auswirkung eines Faults und der nach außen sichtbare Fehlerzustand, weil hierbei die tatsächlichen Ergebnisse (Ist-Zustand) von den erwarteten Ergebnissen (Soll-Zustand) abweichen.

Beim Softwaretesten wird versucht sämtliche Fehlerzustände zu finden, welche auf die Fehlerhandlungen von Entwicklern zurückgeführt und anschließend behoben werden können. Damit wird versucht die Fehlerwirkungen zu reduzieren, damit die Software den festgelegten Anforderungen entspricht. Wichtig ist zu verstehen, dass Softwaretesten nicht die Abwesenheit, sondern lediglich die Anwesenheit von Fehlern aufzeigt.

Program testing can be used to show the presence of bugs, but never to show their absence.

[7]

Das Ziel vom Testen ist das Ausführen einer Software um Fehler aufzudecken [6].

Testing is the process of executing a program with the intent of finding errors. [8]

2.2. Softwaretesten 7

A test that reveals a problem is a success. A test that did not reveal a problem was a waste of time. [3]

Des Weiteren muss beachtet werden, dass ein vollständiges Testen einer Software in der Regel unmöglich ist [3]. Die Anzahl der Eingaben, welche das Programm akzeptiert, kann selbst bei kleinen Programmen unendlich sein. Aber dazu gibt es noch Eingaben, welche das Programm nicht akzeptieren soll. Diese bilden ebenfalls eine riesige Menge, welche in der Regel nicht alle testbar sind. Somit ist ein vollständiger Test eines Programmes praktisch gesehen nicht möglich.

Es ist zudem zu beachten, dass ein getestetes System noch immer Fehler beinhalten kann. Denn auch wenn sämtliche Testfälle keine Fehler aufgedeckt haben, so kann es dennoch sein, dass Fehlerwirkungen mit anderen, nicht getesteten Testfällen gefunden werden könnten.

2.2.2 Testeingaben

Um Software zu testen, werden Testdaten benötigt. Aus diesen Testdaten lassen sich Testfälle erstellen, welche als Eingabe für die Software verwendet werden, um diese zu testen.

Testdaten: Mit Hilfe der Testdaten lassen sich Testfälle bilden. Sie bilden also die Grundlage eines Tests.

Testfall: Bei einem Testfall handelt es sich um ein Tripel, bestehend aus einer Eingabe, Ausführungsbedingungen und dem erwarteten Ergebnis (Soll-Wert) [9]. Die Testeingaben werden in der Software mit den gegebenen Bedingungen ausgeführt und liefern anschließend ein Ergebnis (Ist-Wert), welche mit den Soll-Werten verglichen werden. Stimmen Soll- und Ist-Werte überein, dann ist der Test erfolgreich durchgelaufen und die Software entspricht den spezifizierten Anforderungen, es liegt demnach kein Fehler für diesen Testfall vor. Dazu kann anschließend ein Report erstellt werden.

Testautomatisierung: Je größer ein Softwareprojekt wird, umso höher wird die Anzahl der Tests, die durchgeführt werden, um die Qualität der Software zu bewerten. Deshalb ist es ab einem bestimmten Grad sinnvoll die Software automatisiert zu testen, da es somit möglich ist mehr Tests durchzuführen. Zudem können automatisierte Tests zuverlässiger sein als manuelles Testen. Bei der Testautomatisierung besteht das Ziel die Tests systematisch durchzuführen, um nachvollziehbare Rückschlüsse über die Qualität der Software zu erlangen. Tests sollten wiederholbar sein, um bei einer optimierten Softwareversion die gleichen Tests ein weiteres Mal durchführen zu können und damit zu zeigen, dass die Qualität des Produktes verbessert wurde. Die Tests müssen also einmal erstellt werden, um sie anschließend beliebig oft zu wiederholen. Unter Umständen kann es auch möglich sein, dass Testfälle angepasst werden müssen, wenn sich beispielsweise Anforderungen an eine Software geändert haben, da es dann sein kann, dass die Software sich gewünscht anders verhält als in einer vorherigen Version.

2.2.3 Blackbox-Testentwurfsverfahren

Das Blackbox-Verfahren wird auch als spezifikationsorientiertes Testentwurfsverfahren bezeichnet und ist Teil der dynamischen Tests der Qualitätssicherung. Die IEEE definiert

eine Blackbox [9] als System oder Komponente dessen Eingaben, Ausgaben und allgemeinen Funktionen bekannt sind, aber deren Inhalte oder Implementierung unbekannt oder irrelevant ist. Das Testobjekt wird also von außen betrachtet. Der Ablauf kann nur über die Eingaben gesteuert werden. Der Ansatz basiert darauf Eingabedaten für die Tests aus den Spezifikationen abzuleiten. Wurden allerdings Anforderungen fehlerhaft spezifiziert, so ist es nicht möglich Fehler aufzudecken, wenn sich die Testfälle an den Spezifikationen orientieren. Es wird dabei versucht möglichst alle Anforderungen zu testen, um eine hohe Anforderungsüberdeckung zu erreichen [2].

Für die Erzeugung der Eingaben wird meist die Äquivalenzklassenbildung benutzt, um repräsentative Testdaten zu erlangen und möglichst viele Testfälle abzudecken.

Äquivalenzklassenbildung

Es wird versucht Eingabewerte für die Testfälle in Äquivalenzklassen zu unterteilen. Dabei sollen sich alle Elemente einer Klasse gleich verhalten. Bei der Eingabe eines Repräsentanten einer Klasse in das Testobjekt, wird also das gleiche Verhalten angenommen, wie für jedes andere Element dieser Klasse [10].

Es werden jedoch nicht nur gültige Äquivalenzklassen aufgestellt, sondern auch ungültige, um beispielsweise fehlerhafte Eingaben zu testen, wo der Test fehlschlägt. Die Schwierigkeit besteht darin geeignete Äquivalenzklassen zu bestimmen. Das Ziel in der Äquivalenzklassenbildung liegt darin den theoretisch unendlichen Testraum durch eine endliche Anzahl von Repräsentanten abzudecken.

Grenzwertanalyse

Die Grenzwertanalyse ist als eine Erweiterung der Äquivalenzklassenbildung zu sehen. Da Fehlerfälle in einem Programm oft im Grenzbereich der Äquivalenzklassen auftreten, ist es sinnvoll diese Grenzwerte genauer zu untersuchen. Daher wird die Grenzwertanalyse benutzt, wo der Tester im Gegensatz zur Äquivalenzklassenbildung nicht ein beliebiges Element der Klasse wählt, sondern Elemente aus den Grenzbereichen der Klassen [10].

2.2.4 Systemintegrationstest

Der Integrationstest ist eine Teststufe im sogenannten V-Modell. Er dient dazu die Interaktion zwischen einzelnen Komponenten, also deren Schnittstellen, zu testen.

Werden jedoch die Schnittstellen zwischen externen Softwaresystemen getestet, so wird auch vom Systemintegrationstest oder "Integrationstest im Großen" gesprochen [6]. Dies ist beispielsweise der Fall, wenn die Komponenten von verschiedenen Herstellern geliefert werden. Hierbei kann es vorkommen, dass beispielsweise Schnittstellenformate nicht zusammenpassen oder Daten bei der Übermittlung fehlen, da Komponenten falsch spezifiziert wurden. Das Ziel eines solchen Systemintegrationstest ist somit Fehler in den Schnittstellen der Komponenten aufzudecken, damit diese behoben werden können.

2.3. Geodienste 9

2.3 Geodienste

Als Geodienste werden Dienste bezeichnet, welche Geodaten (oder auch räumliche Daten) und Applikationen für diese zur Verfügung stellen.

Geodaten sind Informationen mit impliziter oder expliziter Referenz zu einem Standort relativ zur Erde [11]. Diese können Primär- sowie Sekundärdaten (Metadaten) enthalten. Die Sekundärdaten wurden im Gegensatz zu den Primärdaten nicht direkt erhoben, sondern sind aus diesen hervorgegangen.

Die Geodaten sind in einem Geoinformationssystem (GIS) gespeichert. Hier werden raumbezogene Daten erfasst, bearbeitet, organisiert, analysiert und präsentiert. Die Systeme können allgemeine Daten enthalten oder aber auf bestimmte Bereiche spezialisiert sein und dort nähere Informationen zur Verfügung stellen.

Im Onlinebereich sind beispielsweise die Dienste von Google Maps [12], Bing Maps [13], Here [14] und OpenStreetMap [15] bekannt. Diese werden im weiteren Verlauf noch detaillierter vorgestellt.

2.4 Adressen

Für die spätere Testfallerzeugung müssen die Testdaten in ein geeignetes Adressformat gebracht werden. Da es hierbei regionale Unterschiede gibt, ist es wichtig die unterschiedlichen Adressformate der Regionen zu kennen. Dieses kennzeichnet, wie eine Adresse aufgebaut ist und an welcher Stelle die einzelnen Adresselemente (beispielsweise die Hausnummer) stehen.

2.4.1 Adresselemente

Die Elemente einer Adresse können die Folgenden sein:

- Sonderziel (POI) Name
- Hausnummer
- Straße
- Ortsteil oder Distrikt
- Stadt
- Postleitzahl
- Bundesstaat (als Bezeichnung der größten Verwaltungseinheit auf nationaler Ebene)
- Land

Diese Elemente können in ein für das Land spezifisches Adressformat gebracht werden und ergeben somit eine Adresse. Dabei muss nicht jedes dieser Elemente Teil der Adresse sein, da einige der Parameter in den verschiedenen Ländern optional oder gar nicht vorhanden sind. Die Bezeichnungen können ebenfalls unterschiedlich sein. Zudem kann es sein, dass einzelne Elemente unterschiedliche Formate haben. So können Hausnummern numerisch oder auch alphanumerisch sein, zudem können sie Trennzeichen, wie Bindestriche oder Leerzeichen

enthalten. Postleitzahlen haben in verschiedenen Ländern ebenfalls unterschiedliche Format. Teilweise sind sie numerisch, alphanumerisch oder sogar nur aus Buchstaben bestehend. Die Anzahl der Postleitzahltoken kann ebenfalls verschieden sein. In einigen Ländern besteht die Postleitzahl aus zwei Elementen, anstatt nur aus einem. Es gibt hierbei also kein einheitliches Format für diese Elemente. Innerhalb eines Landes sind die Formate für die einzelnen Elemente aber genormt. In Deutschland beispielsweise bestehen sämtliche Postleitzahlen aus fünf zusammenhängenden Zahlen. In Großbritannien hingegen sind Postleitzahlen immer alphanumerisch, zwischen sechs und acht Zeichen lang und durch ein Leerzeichen in zwei Elemente unterteilt. Ähnliche Formate können auch in anderen Ländern auftreten.

2.4.2 Formate und Modelle

In vielen Regionen und Ländern werden Adressen vom spezifischsten Element zum allgemeinsten Element aufgebaut. Das bedeutet, dass der Name eines Adressaten als erstes in der Adresse auftritt und das Land, welches das allgemeinste Element ist, als letztes. Für die Generierung von Testfällen für das Testen von Geodiensten ist das spezifischste Element die Straße inklusive Hausnummer (falls diese vorhanden ist) oder aber der Name eines POI.

Beispielformate [16]:

• Deutschland

Straße [Hausnummer] Postleitzahl Ort [Ortsteil] [Land]

• USA

[Hausnummer] Straße [Ortsteil] Ort Bundesstaat Postleitzahl [Land]

• Großbritannien

[Hausnummer] Straße [Ortsteil] Ort Postleitzahl [Land]

Die Eingabe eines Testfalls sollte also eine Adresse nach solch einem Adressformat beinhalten, welches auch in verkürzter Form vorkommen kann.

2.5 Map-Tiling

Map-Tiling beschreibt einen Vorgang, eine Karte in Rechtecke zu teilen. Diese Methode wird in Online Kartendiensten für die Kartendarstellung von verschiedenen Zoomlevel benutzt. Die Karte wird dabei aus mehreren Tiles zusammengesetzt.

Das Map-Tiling-Verfahren kann genutzt werden, um eine gleichmäßige Verteilung von Testfällen zu erzeugen, indem aus jedem Tile in einer Region ein oder mehrere Testfälle

2.5. Map-Tiling

verwendet werden. Dadurch entsteht eine geografische Gleichverteilung der Tests. Beim globalen Tiling wird eine Art Netz über die Erdoberfläche gelegt, welches diese in disjunkte Rechtecke (Tiles) unterteilt. Die Größe der einzelnen Abschnitte des Netzes wird durch das Level des Tiling bestimmt.

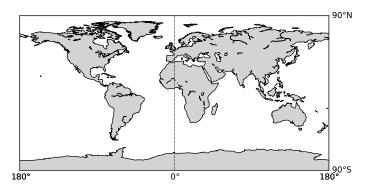


Abbildung 2.2.: Tiling Schema Level 0 - 180° pro Tile (Zylindrische Projektion)

Die Welt wird dabei rekursiv in Hälften geteilt [17]. Auf Level 0 wird sie in zwei Hälften unterteilt, eine Seite ist westlich des Nullmeridian, die andere östlich davon (siehe Abbildung 2.2). Auf dem nächsten Level wird jedes der Tiles in vier weitere Tiles unterteilt, sodass nun insgesamt 8 Tiles entstehen (Abbildung 2.3). In Abbildung 2.4 ist die Darstellung auf Level 2 zu sehen, welche bereits 32 Tiles beinhaltet. Der Vorgang kann beliebig oft weiter geführt werden, bis die gewünschte Tile-Größe erreicht ist. Auf einem Level N umfasst jedes Tile $180,0/2^N$ Grad in jede Richtung, orientiert am WGS 84 Referenzsystem für Koordinaten. Die Anzahl der Tiles beträgt $2^{2\cdot N+1}$.

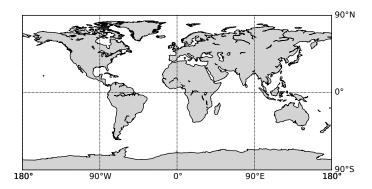


Abbildung 2.3.: Tiling Schema Level 1 - 90° pro Tile (Zylindrische Projektion)

Ein wichtiger Aspekt ist, dass die Tiles entgegen der Form auf den Abbildungen 2.2, 2.3 und 2.4 in Wirklichkeit keine Quadrate darstellen. Auf den Abbildungen wurde eine zylindrische Projekt für die Darstellung verwendet, in der die Achsen verzerrt wurden. Das Tiling-Schema basiert auf dem World Geodetic System 1984 (WGS 84) [17] mit den bekannten Gradmaßen. Daher haben die Tiles eine trapezartige Form, deren Größe sich mit zunehmender Entfernung vom Äquator verringert. Dies wird in der orthografischen Azimutalprojektion (Abbildung 2.5) verdeutlicht. Hier werden ebenfalls die Tiles auf dem Level 2 dargestellt. Dabei sind die

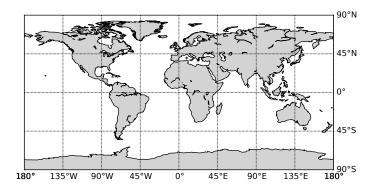


Abbildung 2.4.: Tiling Schema Level 2 - 45° pro Tile (Zylindrische Projektion)

Tiles am Äquator eindeutig größer, als die Tiles, welche weiter vom Äquator entfernt sind. Die Kantenlängen des Tiles sind bezüglich des WGS 84 Referenzsystems gleich groß, aber unterschiedlich groß hinsichtlich der Distanz in Metern.

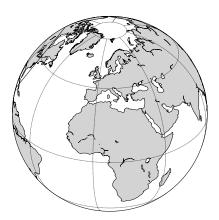


Abbildung 2.5.: Tiling Schema Level 2 - 45° pro Tile (Orthografische Azimutalprojektion)

2.6 Abstände zwischen Punkten in den Tiles

Das Tiling-Schema kann genutzt werden, um Testfälle in bestimmten Tiles zu erzeugen und eine gleichmäßig geografische Verteilung zu erreichen. Ein interessanter Aspekt ist dabei die maximale Entfernung zweier Punkte in benachbarten Tiles, um später Aussagen über die Verteilung tätigen zu können. Der Abstand ist abhängig von der effektiven Tile-Größe, welche vom Äquator in Richtung der Pole kleiner werden. Die Tile-Größe hängt von der Abweitung ab, welche den Abstand zwischen zwei Punkten auf einem Breitenkreisbogen beschreibt. Dieser Abstand ist ebenfalls am Äquator am größten und nimmt in Richtung der Pole ab.

Die Größe der Tiles kann bezüglich des WGS84 Referenzsystems mit der Formel $\frac{180,0^{\circ}}{2^{N}}$ berechnet werden. Nachfolgend werden Tiles auf Level 9 betrachtet. Ein Tile umfasst auf diesem Level in jede Richtung also $\frac{180,0^{\circ}}{2^{9}}=0,3515625^{\circ}$.

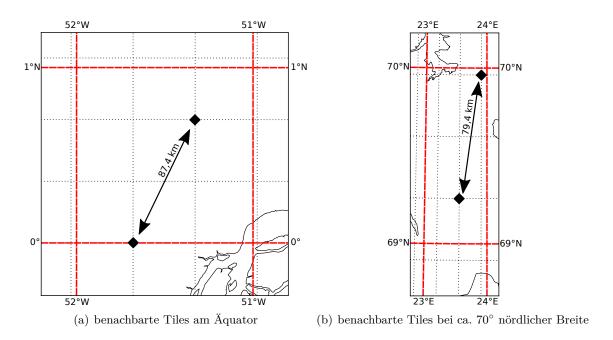


Abbildung 2.6.: maximale Abstände von zwei Punkten in benachbarten Tiles in unterschiedlichen Breiten

Um den maximalen Abstand zwischen zwei Punkten zu ermitteln, wird die Formel zur Berechnung von Orthodromen (kürzeste Verbindung zweier Punkte auf einer Kugeloberfläche) verwendet. Dabei wird angenommen, dass es sich um zwei benachbarte Tiles handelt, welche übereinander liegen. Der Punkt A mit der Breite ϕ_A und der Länge λ_A liegt dabei an der süd-westlichen Ecke des unteren Tiles, Punkt B mit der Breite ϕ_B und der Länge λ_B an der nord-östlichen Ecke des oberen Tiles (Abbildung 2.6a und 2.6b). ζ ist der zentrale Winkel zwischen den beiden Punkten. Für die Berechnung des Abstandes wird dieser mit dem Erdradius r multipliziert.

$$L = \zeta \cdot r$$

$$r = 6371 \text{ km}$$

$$\zeta = \arccos(\sin(\phi_A) \cdot \sin(\phi_B) + \cos(\phi_A) \cdot \cos(\phi_B) \cdot \cos(\lambda_B - \lambda_A))$$

Diese Formel liefert je nach Breitengrad unterschiedliche Ergebnisse, welche am Äquator am größten sind und zum Nordpol hin geringer werden, da sich die Abstände zwischen den Meridianen verringern. Die Transversale Mercator-Projektion in den Abbildungen 2.6a und 2.6b zeigt den Unterschied, wodurch die Abweitung der Meridiane besonders deutlich wird. Während die Tiles am Äquator noch eine nahezu quadratische Form haben, werden sie in höheren Breiten schmaler. Das liegt daran, dass sich der Abstand (in Kilometern) zwischen den Meridianen mit höherer Breite verringert, während er bezüglich des WGS 84 Referenzsystems gleich bleibt. Bei 70° Breite ist der Abstand zwischen zwei Meridianen beispielsweise ungefähr 38 km (Tilebreite circa 13 km) während er am Äquator circa 111 km (Tilebreite circa 39 km) beträgt.

Mit diesem Wissen kann der maximale Abstand zwischen zwei Punkten in benachbarten Tiles

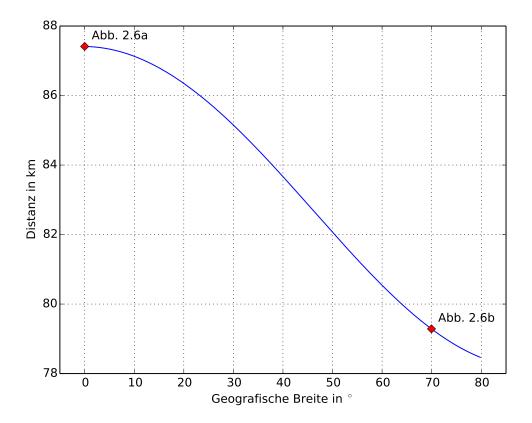


Abbildung 2.7.: Maximalen Abstände zwischen Testfällen benachbarter Tiles

errechnet werden. Abbildung 2.7 zeigt die Verteilung der errechneten maximalen Abstände zwischen den Breiten 0° und 80° . Der Abstand ist dabei wie bereits beschrieben am Äquator am größten und nimmt mit zunehmender Breite ab.

Der minimale Abstand zwischen Testfällen aus benachbarten Tiles hingegen tendiert immer gegen 0, wenn diese direkt aneinander grenzen.

2.7 Navigation Data Standard

Der Navigation Data Standard (NDS) e.V. hat das Ziel Navigationsdaten zu standardisieren. Der Verein ist ein Zusammenschluss von Firmen, welche Automobilhersteller, sowie Unternehmen aus dem Automotive-Bereich und Kartendatenerstellung sind. Der Standard beschreibt ein Kartenformat für Navigationsgeräte. Sämtliche Mitglieder des Vereins erhalten die Lizenzen für den Standard kostenlos [18]. Der Standard hat zum Ziel ein Datenbankformat zu entwickeln, welches einen vereinfachten Austausch und vereinfachte Updates von Navigations- und Kartendaten ermöglichen soll. Der NDS-Standard benutzt zum Speichern der Kartendaten das SQLite-Datenbankformat [19].

KAPITEL 3

Geodienste und Geodatenbanken

Nachdem bereits im Abschnitt 2.3 Geodienste im Allgemeinen beschrieben wurden, stellt dieses Kapitel einige Dienste vor, welche im weiteren Verlauf der Arbeit getestet werden sollen. Zudem werden Aspekte der Qualitätssicherung und des Softwaretestens auf den Fall der Geodienste bezogen und beschrieben. Des Weiteren werden die Komponenten der Geodienste näher erläutert und bewertet.

3.1 Geodienste

In diesem Abschnitt werden Anforderungen von Geodiensten, konkrete Dienste und Aspekte der Qualitätssicherung und des Softwaretestens bezüglich der Geodienste beschrieben.

3.1.1 Anforderungen

Ein typischer Anwendungsfall in einem Geodienst, ist ein Nutzer, welcher nach einer Adresse sucht. Der Anwender möchte, dass diese vom Geodienst gefunden und visualisiert wird. Dabei kann der Geodienst eine Webanwendung sein, eine Anwendung auf einem Mobiltelefon oder beispielsweise die Eingabe für ein Navigationssystem.

Die Anforderungen verlangen also, dass dem Anwender ein Ergebnis nach der Eingabe einer Adresse präsentiert wird. Diese muss dabei nicht unbedingt vollständig sein. Der Dienst soll die Adresse also auch finden, wenn Elemente (z.B. das Land) fehlen, da auch weniger Elemente für eine genaue Bestimmung ausreichen können. Des Weiteren soll der Dienst die Position des Ergebnisses auf einer Karte visualisieren, um für den Anwender deutlich zu machen, wo sich das Ziel befindet. Dabei darf es keine Unterschied machen, welches Adressformat die Eingabe hat, ob diese aus speziellen Zeichen besteht oder es eine besonders lange Eingabe ist, da es sich für den Anwender immer um eine Adresse handelt.

Der Dienst soll in der Lage sein weltweit nach Ergebnissen zu suchen und diese zu finden und darzustellen. Zudem soll es möglich sein, die Eingabe in mehreren Sprachen vorzunehmen, da beispielsweise Städte in unterschiedlichen Sprachen unterschiedliche Bezeichnungen (z.B. DE - München, EN - Munich) haben. Damit verbunden ist die Anforderung, dass der Dienst

verschiedene Alphabete (z.B. kyrillisch) unterstützt, um somit sämtliche Transliterationen, also die Übertragungen von Wörtern in ein anderes Alphabet (z.B. DE - Moskau, RU - Mockβά) zu finden. Denn ein Kunde aus Deutschland sucht beispielsweise mit hoher Wahrscheinlichkeit Adressen im Ausland mit den deutschen Bezeichnungen anstatt der Bezeichnung in der Landessprache, besonders wenn diese ein anderes Alphabet benutzt.

3.1.2 Vorstellung: Geodienste

Der folgende Abschnitt stellt kurz einige Geodienste vor, welche im späteren Verlauf der Arbeit in einer Fallstudie untersucht werden sollen. In Anhang A.2 befinden sich Beispiel-API-Resultate für die Anfrage "Takustraße 9 Berlin".

Google Maps

Google Maps ist der Online-Kartendienst von Google Inc. Der Dienst wurde im Februar 2005 gestartet und enthält noch weitere Anwendungen, unter anderem die Darstellung von Satellitenbildern oder Navigation für Auto inklusive Echtzeitverkehrsinformationen, Fußgänger, Radfahrer und öffentliche Verkehrsmittel. Der Dienst ist für unterschiedliche Plattformen, darunter auch mobile Geräte verfügbar. Leider macht der Dienst keinerlei Aussagen zur Abdeckung seiner Kartendaten. Für die Verwendung der Google Maps REST-API sind für einen Nutzer des kostenlosen API-Schlüssels 2.500 Transaktionen pro Tag möglich. Der API kann über den Parameter language mitgeteilt werden, dass Ergebnisse in einer bestimmten Sprache zurückgegeben werden sollen.

Bing Maps

Bing Maps ist der Online-Kartendienst von Microsoft. Neben Satellitenbilder enthält der Dienst ebenfalls die Möglichkeit der Routenberechnung und Navigation. Ein interessanter Aspekt ist, dass Microsoft Aussagen über die Abdeckung der Kartendaten und Geopositionen verschiedener Regionen und Länder tätigt [20]. Bing Maps lässt 125.000 Transaktionen für einen Nutzer des Basisschlüssels zu. Die Bing Maps API akzeptiert einen Parameter für sogenannte Culture Codes. Dieser Code bestimmt unter anderem die Sprache der Ergebnisse. Allerdings nimmt der Culture Code auch Einfluss auf die Suchregion. Beispielsweise wird durch das setzen des Codes auf de für Deutschland nicht nur die Sprache auf Deutsch gesetzt, sondern auch die Suchregion auf Deutschland fokussiert. Das könnte später Einfluss auf die Suche in anderen Regionen mit einem deutschen Culture Code haben.

Here

Here ist ein Online-Kartendienst des Herstellers Nokia. Der Dienst bietet ebenfalls Anwendungen wie die Satellitenbilddarstellung und Navigation. Here benutzt die Daten des aufgekauften Unternehmens Navteq. Die Abdeckung der Kartendaten und Geopositionen wird ebenfalls durch den Anbieter gelistet [21]. Bei der Verwendung des kostenlosen Plans sind für den Anwender 100.000 Transaktionen pro Monat möglich. Der API von Here kann ein

3.1. Geodienste

zusätzlicher Parameter language übergeben werden, um die Rückgabesprache der Resultate festzulegen.

OpenStreetMap

OpenStreetMap (OSM) ist ein freies Projekt, welches Kartendaten für jeden zur Verfügung stellt. Auf der Website steht eine Visualisierung inklusive Suchmaske zur Verfügung. Sämtliche Kartendaten und Geopositionen werden dabei von einer Community zusammengetragen. Sie werden vor allem in dicht besiedelten Gebieten oft aktualisiert und bieten hier hohe Genauigkeiten, welche laut eigenen Aussagen teilweise sogar besser sind, als die Daten kommerzieller Anbieter. Die Geocoding-Anfragen für OSM werden mit der Geocoding-Software Nominatim aufgelöst. Diese können beispielsweise über den Service von MapQuest [22] genutzt werden, da dieser höhere Kapazitäten besitzt. Die verwendeten Daten, sowie die Geocoding-Software sind jedoch die gleichen wie bei OpenStreetMap. Hier werden maximal 15.000 Transaktionen pro Monat zugelassen. Die API von Nominatim akzeptiert einen Parameter accept-language, welcher die Rückgabesprache der Resultate festlegt.

3.1.3 Geocoding

Geocoding beschreibt den Vorgang mittels einem mitunter unvollständigen Eingabe von raumbezogenen Daten einen Geodatensatz zu erhalten. Die Eingabe für das Geocoding kann beispielsweise eine (nicht vollständige) Adresse sein. Das Ergebnis enthält die Position in Koordinaten, sowie die komplette textuelle Angabe zum Ort (z.B. die Adresse). Der umgekehrte Vorgang wird als Reverse Geocoding bezeichnet. Hierbei werden mittels Koordinaten textuelle Angaben zu einem Ort geliefert (beispielsweise eine Adresse). Der Testgenerator soll Testfälle für das Geocoding mittels Adresseingaben erzeugen.

3.1.4 Softwaretestbegriffe in Bezug auf Geodienste

Im Folgenden werden die in Abschnitt 2.2.1 genannten Begriffe des Softwaretestens beispielhaft für den Fall der Geodienste erklärt.

Error (Fehlerhandlung): Ein Entwickler vergisst eine Methode zu entwickeln, welche überprüft, ob ein Eingabetoken eine Hausnummer sein kann.

Fault (Fehlerzustand): Nachdem ein Anwender eine Adresse bestehend aus Straße, Hausnummer, Postleitzahl und Ort eingegeben hat, sucht die Geocoding-Software nach der Adresse, ignoriert dabei jedoch die eingegebene Hausnummer.

Failure (Fehlerwirkung): Der Anwender kann das gesuchte Haus nicht finden, sondern lediglich die Straße, in welcher das Haus steht. Der Ist-Zustand weicht somit vom Soll-Zustand ab, es liegt ein Fehler vor.

Testdaten: Die Testdaten bestehen bei Geodiensten aus den einzelnen Adresselementen. Mögliche Adresselemente sind: POI-Name, Hausnummer, Straße, Postleitzahl, Ort, Ortsteil, Bundesland, Land.

Testfall: Ein Testfall für einen Geodienst ist die Zusammenstellung einer mitunter nicht vollständigen Adresse aus den Testdaten in einem bestimmten (Adress-)Format. Des Wei-

teren gehört zum Testfall das erwartete Ergebnis, welches die Software liefern soll. Die Soll-Daten sind damit in diesem speziellen Fall die Testdaten. Zudem können bestimmte Ausführungsbedingungen im Testfall vorhanden sein, beispielsweise die Eingabesprache.

3.1.5 Testfälle für Geodienste

Die Testfälle für Geodienste müssen von den Anforderungen abgeleitet werden, welche bereits in Abschnitt 3.1.1 vorgestellt wurden. Dabei stellt sich die Frage, wie Eingaben für die Testfälle schließlich aufgebaut werden sollen. Es muss überlegt werden, welches Format gewählt wird, ob einzelne Adresselemente nicht in der Adresse auftauchen, ob Elemente gekürzt, ob einzelne Zeichen wie Separatoren zwischen den Elementen gesetzt oder ob Sonderzeichen in den einzelnen Elementen weggelassen werden. Dieser Fall kann auftreten, da der Nutzer unter Umständen seine Eingabe in den Geodienst abkürzt, in der Hoffnung trotzdem noch das richtige Ergebnis zu finden. Das heißt, dass für einen Testdatensatz verschiedene Eingaben und somit unterschiedliche Testfälle erzeugt werden können.

Beispieltestdaten und -testfälle

Nachfolgend wird anhand der Beispieltestdaten in Tabelle 3.1 gezeigt, wie die Testfälle erzeugt werden. Es werden beispielhafte Eingaben und dazugehörige erwartete Ergebnisse vorgestellt.

Adresselement	Beispiel 1	Beispiel 2
POI Name	Rathaus Charlottenburg	_
Hausnummer	100	9
$\mathbf{Straße}$	Otto-Suhr-Allee	Takustraße
Postleitzahl	10585	14195
\mathbf{Ort}	Berlin	Berlin
$\mathbf{Ortsteil}$	Charlottenburg	Dahlem
Staat	Berlin	Berlin
Land	Deutschland	Deutschland

Tabelle 3.1.: Beispiel Testdaten

Der Nutzer hat unterschiedlichste Möglichkeiten die zu suchende Adresse einzugeben. Im Folgenden sind einige Beispieleingaben eines Benutzers inklusive der zugehörigen erwarteten Ergebnisse aufgelistet.

• Testfall: Eingabe einer kompletten Adresse inklusive POI Name

Eingabe: Rathaus Charlottenburg, Otto-Suhr-Allee 100, 10585 Berlin Charlottenburg, Deutschland

Erwartetes Ergebnis: POI Name: Rathaus Charlottenburg, Straße: Otto-Suhr-Allee, Hausnummer: 100, Postleitzahl: 10585, Stadt: Berlin, Ortsteil: Charlottenburg, Bundesland: Berlin, Land: Deutschland

3.1. Geodienste

• Testfall: Eingabe einer kompletten Adresse

Eingabe: Otto-Suhr-Allee 100, 10585 Berlin Charlottenburg, Deutschland

Erwartetes Ergebnis: Straße: Otto-Suhr-Allee, Hausnummer: 100, Postleitzahl: 10585, Stadt: Berlin, Ortsteil: Charlottenburg, Bundesland: Berlin, Land: Deutschland

• Testfall: Eingabe Straße, Postleitzahl, Ort und Land

Eingabe: Otto-Suhr-Allee 100, 10585 Berlin, Deutschland

Erwartetes Ergebnis: Straße: Otto-Suhr-Allee, Hausnummer: 100, Postleitzahl: 10585, Stadt: Berlin, Bundesland: Berlin, Land: Deutschland

• Testfall: Eingabe von Straße, Postleitzahl und Ort

Eingabe: Otto-Suhr-Allee 100, 10585 Berlin

Erwartetes Ergebnis: Straße: Otto-Suhr-Allee, Hausnummer: 100, Postleitzahl: 10585, Stadt: Berlin, Bundesland: Berlin, Land: Deutschland

• Testfall: Eingabe von Straße und Ort

Eingabe: Otto-Suhr-Allee 100, Berlin

Erwartetes Ergebnis: Straße: Otto-Suhr-Allee, Hausnummer: 100, Postleitzahl: 10585, Stadt: Berlin, Bundesland: Berlin, Land: Deutschland

• Testfall: Eingabe von Straße und Postleitzahl

Eingabe: Otto-Suhr-Allee 100, 10585

Erwartetes Ergebnis: Straße: Otto-Suhr-Allee, Hausnummer: 100, Postleitzahl: 10585, Stadt: Berlin, Bundesland: Berlin, Land: Deutschland

• Testfall: Eingabe von Straße und Ortsteil

Eingabe: Otto-Suhr-Allee 100, Charlottenburg

Erwartetes Ergebnis: Straße: Otto-Suhr-Allee, Hausnummer: 100, Postleitzahl: 10585, Stadt: Berlin, Ortsteil: Charlottenburg, Bundesland: Berlin, Land: Deutschland

• Testfall: Eingabe der Adresse ohne Trennzeichen (Kommata entfernt)

Eingabe: Otto-Suhr-Allee 100 10585 Berlin

Erwartetes Ergebnis: Straße: Otto-Suhr-Allee, Hausnummer: 100, Postleitzahl: 10585, Stadt: Berlin, Bundesland: Berlin, Land: Deutschland

• Testfall: Eingabe einer modifizierten Adresse (Bindestriche entfernt)

Eingabe: Otto Suhr Allee 100, 10585 Berlin

Erwartetes Ergebnis: Straße: Otto-Suhr-Allee, Hausnummer: 100, Postleitzahl: 10585, Stadt: Berlin, Bundesland: Berlin, Land: Deutschland

• Testfall: Eingabe von gekürzten Elementen

Eingabe: Takustr. 9, 14195 Berlin

Erwartetes Ergebnis: Straße: Takustraße, Hausnummer: 9, Postleitzahl: 14195,

Stadt: Berlin, Bundesland: Berlin, Land: Deutschland

Es gibt also viele mögliche Eingaben für einen bestimmten Testdatensatz. Somit lassen sich für einen Satz mehrere Testfälle mit unterschiedlichen Eingaben, aber teilweise gleichen erwarteten Ausgaben erzeugen.

Es kann einerseits getestet werden, ob die Eingabe in einem landesspezifischen Adressformat wichtig ist, oder ob das Format vernachlässigt werden kann. Vielleicht verwendet der Nutzer eine Reihenfolge, die nicht dem Adressformat entspricht, woraufhin er natürlich trotzdem das Ergebnis angezeigt bekommen möchte.

Somit kann am Ende bewertet werden, wie detailliert eine Eingabe sein muss, um gefunden zu werden, also ob einzelne Elemente entfernt oder vertauscht werden können. Vielleicht enthält das Resultat bei zu ungenauer Eingabe mehrere Ergebnisse, da die Eingabe nicht eindeutig zugeordnet werden konnte.

Zudem kann betrachtet werden, wie genau die Übereinstimmung der Eingabe mit den Daten in der Datenbank für eine erfolgreiche Suche sein muss. Vielleicht schlagen auch Tests mit abgekürzten Eingaben oder Eingaben einzelner Elemente mit entfernten Sonderzeichen fehl, da die Eingaben komplett übereinstimmen müssen. Vielleicht reicht hingegen auch ein Präfix eines Elementes aus und der Geodienst kann das Suffix von sich aus ermitteln.

Des Weiteren kann geprüft werden, ob die Software damit besser zurecht kommt, wenn die Adress-Elemente der Eingaben durch Separatoren wie Kommas getrennt werden, um somit bereits zusammenhängende Elemente zu markieren (wie zum Beispiel Hausnummer und Straße). Somit könnte die Software zusammenhängende Elemente leichter erkennen. Vielleicht verwirft die Software aber auch sämtliche Zeichen zwischen den Elementen und betrachtet nur diese.

3.1.6 Testen von Geodiensten

Geodienste wie die von Google Maps oder Here sind kommerzielle Produkte, welche keine öffentliche Einsicht in die Software und die dazugehörigen Datenbanken zulassen.

Für die Qualitätsuntersuchung dieser Dienste, um sie beispielsweise untereinander oder mit öffentlich zugänglichen Dienste wie OpenStreetMap zu vergleichen, können Softwaretests sinnvoll sein. Da die kommerziellen Produkte ihre Navigations- und Kartendaten in der Regel von Anbietern für Fahrzeugnavigation erhalten, sollten diese eine hohe Qualität liefern und daher als Referenzbestand und auch zum Testen gut geeignet sein [23].

Die Tests geben zudem Aufschluss über die Abdeckung der einzelnen Dienste. Beispielsweise kann es sein, dass ein Dienst besser in großen Städten arbeitet oder aber in spärlich besiedelten Gebieten bessere oder genauere Ergebnisse finden kann. Vielleicht kann ein Dienst auch besonders gut mit speziellen Sonderfällen umgehen, wie Kreuzungen, POIs oder Anfragen mit Sonderzeichen. Es kann zudem untersucht werden, wie gut ein Dienst in den unterschiedlichen Sprachen und Alphabeten arbeiten kann, also ob es ihm möglich ist, Transliterationen von Adressen oder Orten zu finden.

Ein anderer Aspekt, wieso das Testen von Geodiensten nützlich sein kann, ist die Integration zwischen der eigentliche Software, welche beispielsweise die Suchmaske bereitstellt, und der zugrunde liegenden Datenbank zu untersuchen. Dabei handelt es sich dann um einen

3.1. Geodienste

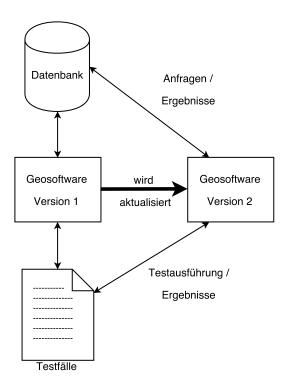


Abbildung 3.1.: Systemintegration: Update der Software

Systemintegrationstest zwischen diesen beiden Komponenten.

Systemintegrationstest bei Geodiensten

Ein Systemintegrationstest zwischen einer Geocoding-Software und ihrer Datenbank soll Fehler aufdecken, wenn sich eine der Komponenten ändert.

Der erste Fall betrifft ein Update der Datenbank. Zuvor wurde dort ein Fehler festgestellt, welcher mit der neuen Version behoben wurde. Die Software hingegen wurde nicht aktualisiert. Für die Qualitätssicherung kann nun erneut ein Integrationstest dieser beiden Komponenten durchgeführt werden.

Im anderen Fall wurde die Software aufgrund eines Fehler aktualisiert. Die Datenbank hingegen ist noch die alte Version. Nun muss die neue Software mit der alten Datenbank getestet werden. Die Testfälle, welche bereits vorher vorhanden waren, können weiterhin für den Test verwendet werden. Dieser Vorgang wird in Abbildung 3.1 dargestellt.

3.1.7 Blackbox-Verfahren bei Geodiensten

Im Abschnitt 2.2.3 wurde das Blackbox-Verfahren für Softwaretesten bereits vorgestellt. Der folgende Abschnitt beschreibt, wieso diese Methode zum Testen von Geodiensten benutzt werden kann. Zudem wird erklärt, welche Vorgehensweise hilft, Testfälle für das Blackbox-Verfahren für die Geodienste zu generieren.

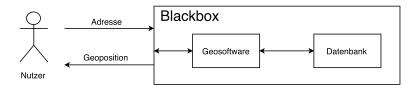


Abbildung 3.2.: Geodienst als Blackbox aus Nutzersicht

Für ein besseres Verständnis wird in Abbildung 3.2 der Geodienst aus der Nutzersicht dargestellt. Der Anwender sieht den Geodienst als Blackbox. Er gibt eine Adresse in eine Suchmaske ein und erhält Geoinformationen sowie eine Geoposition als Resultat, welche ihm visualisiert wird. Wie das Ergebnis intern ermittelt wird, erfährt der Anwender nicht.

Für die Durchführung der Qualitätssicherung gibt es die konstruktiven und analytischen Verfahren. Da die Produkte bereits existieren und von außen kein Einfluss auf die Entwicklung dieser besteht, ist eine konstruktive Qualitätssicherung nicht für jedermann möglich. Um analytische Qualitätssicherung zu betreiben, existieren die statischen und dynamischen Verfahren. Da die statischen Verfahren jedoch versuchen Code und Dokumente durch Durchsichten und Inspektionen zu untersuchen, sind diese ebenfalls nicht aus einer externen Sicht möglich. Es bleibt jedoch noch das dynamische Verfahren, welches darauf zielt die Software zu testen. Da jedoch kein Einblick in die Programmstruktur besteht, kann die Software nicht als Whitebox betrachtet werden, sondern lediglich als Blackbox. Da die Geodienste bestimmte Anforderungen an ihre Software haben, können somit auch Testfälle abgeleitet werden oder aber es werden zufällig gewählte Eingaben verwendet [24]. Somit ist es möglich, die Software als Blackbox zu testen. Die Eingaben sind beispielsweise Adressen (eventuell in gekürzter Form) oder Sonderziel (POI) Namen, die Ausgaben sind die genaue Lage inklusive der Adresse, sowie eine Visualisierung auf einer Karte. Für die Durchführung des Blackbox-Testverfahrens kann das Testobjekt, in diesem Fall also die Geocoding-Software, lediglich über die Eingabe gesteuert werden.

Die Adressen oder POI-Namen, welche als Eingaben für die Tests dienen, müssen also erzeugt werden. Wie bereits in Abschnitt 2.2.3 beschrieben wurde, kann die Äquivalenzklassenbildung benutzt werden, um Klassen zu erstellen, welche bestimmte Elemente enthalten, die alle das gleiche Verhalten der Software sowie die dazugehörige Ausgabe produzieren, wenn sie als Eingabe für das Testobjekt verwendet werden. Hierbei stellt sich nun wieder die Schwierigkeit, geeignete Äquivalenzklassen zu finden, welche eine sinnvolle Testabdeckung ermöglichen. Zudem ist es nötig reale Adressen aus den Äquivalenzklassen als Repräsentanten zu wählen, welche der Dienst auch wirklich finden kann. Hierbei muss eine Adressdatenbank hinzugezogen werden, denn sonst ist es nur mit geringer Wahrscheinlichkeit möglich, reale Adressen zu erzeugen. Wenn die Äquivalenzklasse beispielsweise vorgibt, welches Format eine Straße haben kann, sich in der Klasse jedoch nur Straßen befinden, die in einem bestimmten Ort nicht existieren, nützt es nichts diese zu testen, da bereits davon ausgegangen werden kann, dass die Software diese Adresse nicht finden kann. Es müssen also die real existierenden Adressen aus der Datenbank benutzt und in die Äquivalenzklassen eingeteilt werden.

Mit den Äquivalenzklassen ist ein systematisches Ausprobieren möglich. Sie ermöglichen es Testfälle sinnvoll auszuwählen, um eine möglichst hohe Testabdeckung zu erhalten ohne

3.1. Geodienste

dabei sämtliche Eingaben zu testen (was in der Regel nicht möglich ist). Interessant dabei ist, wie gleiche Testdaten genutzt werden können, um Testfälle mit unterschiedlicher Eingabe aber gleichem erwarteten Ergebnis zu konstruieren. Dazu werden gleiche Adressdaten in der Eingabe auf unterschiedliche Weise dargestellt. Der Abschnitt 2.4 geht genauer darauf ein, wie Testdaten zu mehreren Testfällen verarbeitet werden können.

3.1.8 Überdeckungsmaße

Wann wurde genug getestet? Da das Blackbox-Verfahren zum Testen von Geodiensten und ihren Datenbanken genutzt werden soll, wird eine hohe Überdeckung erreicht, wenn alle Anforderungen an den Dienst mit Hilfe der Tests nachgewiesen werden können.

Da ein Geodienst weltweite Adressen erkennt, muss überlegt werden, ob die Anforderungen für sämtliche Länder überdeckt werden müssen oder lediglich für einzelne Staaten. Es stellt sich die Frage, wie die Testfälle dann verteilt werden müssen, um eine hohes Überdeckungsmaß zu erzielen. Politische Grenzen sollten hierbei genutzt werden, um die Überdeckung in den einzelnen Ländern zu gewährleisten. Dabei kommt der Aspekt unterschiedlicher Sprachen und Alphabete ins Spiel.

Neben der politischen Verteilung kann die geografische Verteilung betrachtet werden. Um die Überdeckung zu gewährleisten ist dieser Vorgang unter Umständen sinnvoller als ein rein politischer Ansatz. Denn so können auch niedrig bevölkerte Regionen (z.B. Sibirien) untersucht und abgedeckt werden, wenn die Verteilung geografisch gleichmäßig vorgenommen wird. Damit lassen sich wesentlich mehr Aussagen über die Überdeckung tätigen. Die geografische Verteilung hat zudem den Vorteil, dass auch bewertet werden kann, wie groß die Abdeckung der Datenbank des Geodienstes ist. Wird beispielsweise eine Stadt überhaupt nicht gefunden, welche geografisch weit von anderen Orten abgegrenzt ist, so kann die Annahme getätigt werden, dass sie überhaupt nicht in der Datenbank vorhanden ist.

Des Weiteren kann eine Kombination aus politischer und geografischer Verteilung genutzt werden. Hierbei können politische Zonen, wie Länder, um eine geografische Komponenten erweitert werden, damit wenig und dicht besiedelte Gebiete überdeckt werden. Oder es werden mehrere politische Zonen zusammen geschlossen und somit eine logische Ebene erstellt, auf der getestet wird.

Am Ende sollte zudem geklärt werden, wie sinnvoll eine Randomisierung der Erzeugung von Testfällen ist. Dabei kann versucht werden echt-zufällige oder pseudo-zufällige Testfälle zu erzeugen. Der Nachteil in den zufällig erzeugten Tests besteht jedoch darin, dass sich nur schwer Aussagen über die Abdeckung machen lassen [24]. So ergibt es unter Umständen wenig Sinn, wahllos Testfälle für ein politisches Gebiet zu erzeugen, wenn durch den Zufall aufgrund dicht besiedelter Gebiete im Prinzip nur ein Bruchteil der kompletten Region abgedeckt wird und somit gering besiedelte Gebiete stark vernachlässigt werden.

Ein geeignetes Überdeckungsmaß ist also nicht so leicht zu finden. Es sollte demnach wohl überlegt sein, wie die Verteilung der Tests vorgenommen wird und welche Aussagen solche Verteilungen für die Überdeckung bedeuten.

3.2 Geodienst-Komponenten

Mit einem Systemintegrationstest kann die Interaktion verschiedener Komponenten getestet werden. Dies sind die Anwendung, inklusive Suchmaske und Kartendarstellung, also die Geocoding-Software und die Datenbank. Das Frontend sendet die Suchanfragen an die Geocoding-Software, welche wiederum Anfragen an die Datenbank sendet und die Ergebnisse entgegen nimmt. Die Datenbank enthält sämtliche Geoinformationen, darunter die für die Testerzeugung wichtigen Adressen.

3.2.1 Geocoding-Software

Das Frontend enthält eine Eingabemaske, in welcher der Benutzer Anfragen eingeben kann. Das sind beispielsweise eine Adresse oder ein POI, welche von einer Geocoding-Software verarbeitet werden. Diese erzeugt eine Datenbankanfrage und schickt sie an die Datenbank. Nachdem die Datenbank ein Ergebnis findet, wird es vom Backend zurück an die Geocoding-Software gesendet, welche das Resultat dem Nutzer im Frontend in Form von Text und einer Karte präsentiert.

Eine wichtige Aufgabe vom Geocoding ist dabei zu interpretieren, was der Nutzer mit seiner Eingabe meinen könnte, damit es entsprechende Anfragen für die Datenbank absetzen kann. Dabei können beispielsweise die Eingabesprache (welche über den Browser zu erkennen ist) und die Position des Nutzers (die Wahrscheinlichkeit ist hoch, dass der Nutzer ein Ergebnis in seiner Umgebung oder seinem Land sucht) mit in die Interpretation der Eingabe einfließen.

Die Schwierigkeit für die Geocoding-Software besteht darin die korrekte Interpretation zu finden, um zu ermitteln, was der Nutzer mit seiner unter Umständen mehrdeutigen Eingabe meinen könnte. Die Software ermöglicht dem Kunden bei vielen Diensten mit wenig Eingabe viele Informationen zu finden. Dies kann zum Beispiel eine Präfixsuche sein, welche oft verwendet wird. Diese lässt allerdings auch mehr Interpretationen zu, als eine Suche, die nur auf komplette Token sucht. Ein Beispiel hierfür ist die Eingabe "Berlin Bahnhof". Der Nutzer könnte hiermit das Sonderziel Bahnhof in der Stadt "Berlin" meinen. Dabei stellt sich aber die Fragen, welcher Bahnhof gemeint ist. Vielleicht ist der Ausdruck "Berlin" aber auch lediglich ein Präfix für den Ort "Berlingen", in dem der Nutzer den Bahnhof finden möchte. Eine weitere Möglichkeit ist, dass der Anwender auf der Suche nach der "Bahnhofstraße" in "Berlin" ist und das Wort "Bahnhof" hierbei als Präfix fungiert.

3.2.2 Geodatenbanken

Die Geodatenbank, welche im Hintergrund des Geodienstes arbeitet, stellt die Geodaten bereit. Diese umfassen einerseits sämtliche Adressinformationen. Dazu gehören die Adresselemente, welche in Abschnitt 2.4.1 beschrieben werden. Zusätzlich dazu müssen Informationen über die Positionen all dieser Adressen abgelegt sein, um sie visualisieren zu können. Andererseits sind in den Datenbanken geografische Raster- und Vektordaten gespeichert. Diese werden für die Darstellung der Karte genutzt. Zudem können anhand dieser Daten Routen erstellt werden.

Die Struktur einer Geodatenbank kann komplett unterschiedlich ausfallen. Diese können

beispielsweise relational gespeichert sein. Es ist jedoch auch möglich Graphdatenbanken zu verwenden, welche zum Speichern von Raster- und Vektordaten vorteilhaft sein können. Oder aber der Geodienst nutzt Kombinationen aus diesen Datenbanken, um seine Daten zu speichern.

Struktur, Aufbau und die Art und Weise wie Daten dort gespeichert werden, sind dem Geodienst überlassen.

3.2.3 Analyse der Datenbanken

Für die Testerzeugung ist eine Geodatenbank nötig, denn ohne diese können keine realen Testfälle gebildet werden. Hier sind die Adressen in einer bestimmten Art und Weise abgelegt. Die Struktur ist dabei vom Anbieter der Datenbank abhängig. Für die Testfallerzeugung müssen zunächst sämtliche Testdaten erlangt werden. Also wird versucht alle Elemente, wie in Abschnitt 2.4.1 beschrieben zu extrahieren. Je nach Datenbank und ihrer Struktur kann dies unterschiedlich komplex sein.

Im Folgenden werden die OpenStreetMap-Datenbank und Datenbanken des Navigation Data Standard (NDS) erläutert und soweit möglich analysiert, wie gut die Verwendung dieser Datenbank zur Erzeugung von Testdaten und Testfällen ist.

OpenStreetMap-Datenbank

OpenStreetMap (OSM) verwendet eine PostgreSQL-Datenbank [25]. Die Daten sind in einem relationalen Format gespeichert. Die Datenbank speichert dabei jedoch nicht die Adressen relational ab. Die Adressen sind nur implizit in den Elementen gespeichert, welche aus Knoten, Wegen und Relationen bestehen [26]. Diese Elemente werden mit Attributen versehen [27], welche unter anderem die Position (bei Knoten), die ID des Elementes, den Bearbeiter und den Modifikationszeitstempel enthalten. Zusätzlich enthält es Informationen zur Historie und zu Veränderungen des Elementes. Wenn es sich bei diesen Elementen um eine Adresse handelt, dann ist diese ebenfalls mit Tags als Kinderknoten in den jeweiligen Knoten, Wegen oder Relationen gespeichert. Dabei befinden sich in der Regel sämtliche Informationen dazu direkt in diesem Element. Die Adresselemente sind also nicht in einer hierarchischen Struktur gespeichert. Somit ist ein direkter Zugriff auf Adressen relativ umständlich. Die komplette Datenbank wird regelmäßig von OSM exportiert und bereitgestellt. Das Ausgabeformat der Daten ist ein XML Format. Es gibt Dumps für unterschiedliche Regionen, sodass Anwender auf Teildatensätzen arbeiten können.

10 </node>

Listing 3.1: OSM-Export: Takustraße 9

In Anhang A.1 ist der XML Eintrag für die Takustraße 9 in Berlin Dahlem zu finden. Es handelt sich dabei um einen OSM-Knoten, welcher unter anderem Attribute wie ID, Version und Position enthält. Zudem enthält er Kinderknoten, welche die Tags sind und deren Attribute die Adressinformation enthält. Dabei sind Einträge für Stadt, Land, Hausnummer, Postleitzahl, Straßenname und Stadtbezirk enthalten. Bei OpenStreetMap sind sämtliche Adressen in diesem oder einem ähnlichen Format gespeichert. Eine Adresse kann dabei auch in einem Weg oder einer Relation abgelegt sein. Jedoch wird hier auch deutlich, dass die Adressinformationen nicht hierarchisch gespeichert sind. Das erschwert den Zugriff und somit auch die Extraktion der Adressinformationen. Zudem ist die Zuordnung zu einem Staat nicht trivial, da diese Information in vielen Fällen nicht mit im Knoten gespeichert ist. Der Vorteil der OpenStreetMap-Daten ist, dass sie für jeden zugänglich sind und eine relativ hohe Aktualisierungsfrequenz besitzen (zumindest in stark besiedelten Gebieten).

Navigation Data Standard

Beim Navigation Data Standard (NDS) werden Kartendaten im SQLite-Datenbankformat, also einem relationalen Format, gespeichert. Innerhalb der Datenbank sind die Informationen in sogenannte Update Regions unterteilt [28]. Dabei werden einer Update Region ein oder mehrere Länder zugeteilt. Diese können dann separat aktualisiert werden. Die Update Regions sind wiederum in einzelne Komponenten unterteilt. Diese enthalten Building Blocks, welche für ein bestimmtes Feature vorgesehen sind. Die Daten der einzelnen Building Blocks sind über ein vorgegebenes Format miteinander verknüpft. Eine Übersicht der Building Blocks ist in Abbildung 3.3 dargestellt.

Für die Adresssuche und Testerzeugung mit einer NDS-Datenbank sind lediglich die Building Blocks Full-Text Search und SQLite Index (SLI) aus der Darstellung 3.3 interessant, da hier Adressinformationen abgelegt sind. Für eine Unterstützung von POIs wäre zusätzlich der Points of Interests Building Block interessant.

Der genaue Aufbau und die Struktur der NDS-Datenbanken kann zwischen den Herstellern jedoch ebenfalls differieren. Aber ein allgemeines Grundwissen über das Format legt nahe, dass diese Datenbanken vor allem wegen der hierarchischen Struktur für die Testfallerzeugung gut geeignet sind. Die Geodaten in den NDS-Datenbanken werden von Herstellern wie Here und TomTom geliefert.

3.2.4 Bewertung der Datenbanken

Um zu entscheiden, welche Datenbanken sich für eine automatisierte Erzeugung von Testfällen eignen, müssen einige Kriterien bewertet werden. Einerseits sollte der Aufbau und die Struktur zur Entscheidung beitragen, um möglichst effektiv Testdaten aus den Datenbanken zu extrahieren. Andererseits muss die Datenqualität und die Vollständigkeit der Datenbanken betrachtet werden. Fehlende oder inkorrekte Einträge erzeugen keine beziehungsweise falsche Testdaten. Diese bringen dann keinen Nutzen für die Testfallerzeugung.

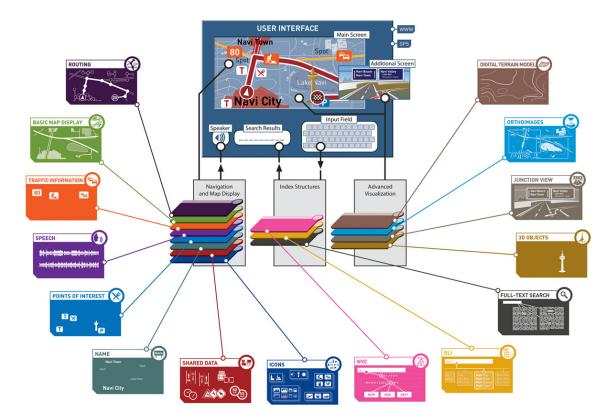


Abbildung 3.3.: Übersicht: NDS Building Blocks (entnommen aus [28])

Die Struktur der OSM-Datenbank ist hinsichtlich dem Auslesen von Adressdaten relativ ungeeignet. Wie bereits zuvor erwähnt, unterliegen die Adressdaten keiner hierarchischen Struktur, was es erschwert, diese gezielt zu erfassen, um die Testdaten zu erlangen. NDS Datenbanken besitzen eine feste Struktur, auch wenn sie sich bei einzelnen Herstellern unterscheiden kann. Eine feste Grundstruktur ist immer vorhanden, womit sie sich somit eher für die Testdatenerzeugung eignet. Für die Betrachtung des Anwendungsfalls sämtliche Straßen einer Stadt zu ermitteln, ist der Aufwand dafür mit der OSM-Datenbank aufgrund der uneinheitlichen Struktur sehr hoch, während der Vorgang mit der NDS-Datenbank wesentlich einfacher ist.

Ein weiterer wichtiger Punkt für die Bewertung ist die Qualität der Daten. Die Informationen in OpenStreetMap werden auch als VGI (volunteered geographic information) [29] (zu deutsch "freiwillig erhobene geografische Daten") bezeichnet. Sie können also von jeder Person eingetragen und geändert werden. Jedoch kann es dadurch passieren, dass Daten fehlerhaft eingetragen werden, auch wenn dies nicht beabsichtigt geschieht. Fehlerhafte Daten führen jedoch somit auch zu inkorrekten Testdaten und somit unter Umständen zu einer geringeren Qualität.

Die Vollständigkeit der Daten in den Datenbanken stellt ein weiteres wichtiges Kriterium für die Wahl der Datenbank dar. Es muss abgewägt werden, ob die Vollständigkeit einer Datenbank, welche durch eine große Community gefüllt wird, ausreichend gegenüber einer Datenbank eines kommerziellen Unternehmens ist. Die Abdeckung und Vollständigkeit bei OpenStreetMap wurde schon in vielen Arbeiten thematisiert [23] [30] [31] [32]. Beispielsweise

wurden in den Arbeiten von Voss [23] [30] das deutsche Straßennetz von OpenStreetMap der Jahre 2009 und 20111 mit dem Netz des Anbieters von kommerziellen Kartendaten Navteq (mittlerweile Here) vom Jahr 2008, welcher als Referenzbestand diente, untersucht. Dort wurde festgestellt, dass die Vollständigkeit der Daten in stärker besiedelten Gebieten wesentlich höher ist, als in schwächer besiedelten Gebieten [23]. Zur Vollständigkeit gehören hierbei unter anderem die Bezeichnungen der Straßen und Metadaten, wie Höchstgeschwindigkeiten oder Straßenart. Zudem hat sich die Abdeckung der Navteq Daten durch OSM zwischen den Jahren 2009 und 2011 stark verbessert. Die relative Vollständigkeit stieg deutschlandweit von 58% im Jahr 2009 auf 77% im Jahr 2011 [30]. Allgemein wurde festgestellt, dass zumindest in Städten mit über 100.000 Einwohnern die Qualität der Daten bezüglich der Vollständigkeit scheinbar ausreichend ist. Gering besiedelte Regionen jedoch scheinen aufgrund ihrer relativ starken Unvollständigkeit der Daten eine geringere Qualität zu besitzen. Es gibt natürlich noch andere Qualitätsmaße, welche in den Arbeiten untersucht wurde, wie die Positionsgenauigkeit der Daten. Diese sind allerdings für die Testfallerzeugung weniger interessant. Das Problem der Unvollständigkeit hingegen ist hier stärker zu beachten. Denn wenn Straßen überhaupt nicht vorhanden sind, oder aber vorhandene Straßen keinerlei Attribute wie den Straßennamen besitzen, dann können daraus auch keine Adressen abgeleitet werden. Somit könnten diese Gebiete nicht getestet werden, was für eine geografische Abdeckung schlecht wäre.

Kommerzielle Datenbanken, wie eine NDS-Datenbank, sollten sich hingegen sehr gut für die Testfallerzeugung eignen [23]. Der entscheidende Vorteil liegt vor allem in der Datenvollständigkeit, welche bei einem kommerziellen Produkt angenommen werden kann. Da die Hersteller überall Kunden haben, müssen sie dafür sorgen, dass Ihre Datenbanken eine hohe Abdeckung liefern. Hier liegt also gegenüber OpenStreetMap der Vorteil in der höheren Datenvollständigkeit. Auch die Datenqualität sollte in solch einem Produkt einen hohen Stellenwert haben, da es mehrere Qualitätsstufen durchlaufen sollte. Des Weiteren hat die NDS-Datenbank durch ihre relationale Datenbankstruktur mit eindeutigen Hierarchien für Adressen einen klaren Vorteil gegenüber OSM.

Natürlich ist nicht zu einhundert Prozent sichergestellt, dass ein kommerzielles Produkt einem nicht-kommerziellen Produkt überlegen ist. Sicherlich ist der Fakt, dass es sich um ein kommerzielles Produkt handelt, nicht ausreichend, um zu argumentieren, dass es somit eine höhere Datenqualität besitzt. Gerade durch die häufigen Aktualisierungsfrequenzen in offenen Projekten wie OSM, kann die Qualität der Daten schnell steigen und somit unter Umständen eine höhere Datenqualität als ein kommerzielles Produkt erreichen [28] (auch wenn dies nur für einen Teil der Daten gilt).

3.3 Eingabesprachen

Benutzer eines Geodienstes können aus unterschiedlichen Ländern kommen. Das bedeutet auch, dass sie verschiedene Sprachen beherrschen, was die Eingabe beeinflusst. Der Dienst sollte jedoch jede Eingabesprache akzeptieren, um dem Nutzer die Eingabe so komfortabel wie möglich zu gestalten. Denn es kann nicht davon ausgegangen werden, dass jeder Nutzer die Bezeichnungen von Orten und Straßen in einer anderen Sprache beherrscht. Zudem ist dem Anwender unter Umständen nicht bewusst, welche diakritischen Zeichen für eine Eingabe in der ursprünglichen Sprache verwendet werden müssten.

Orte können zum Beispiel in unterschiedlichen Sprachen andere Bezeichnungen haben (Beispiel: DE - München, EN - Munich). Das gleiche Problem kann bei Straßennamen auftreten. Deshalb müssen Adresselemente in der Datenbank in mehreren Sprachen hinterlegt sein, wenn diese gefunden werden sollen. Aber auch die Software muss erkennen, dass es sich um das selbe Element handelt, wenn es in unterschiedlichen Sprachen eingegeben wurde. Somit ist es sinnvoll Testfälle mit Eingaben in mehrere Sprachen zu erstellen.

Während die Bezeichnungen in einer anderen Sprache eventuell ein geringes Problem darstellen, wenn sie das gleiche Alphabet verwenden, so kann es sich mit einer Sprache, welches ein anderes Alphabet verwendet, wesentlich schwieriger gestalten. Da der Nutzer jedoch in vielen Fällen lediglich das Alphabet seiner Muttersprache beherrscht und er unter Umständen gar nicht die Möglichkeit besitzt, ein anderes Alphabet einzugeben, muss die Software Transliterationen erkennen. Während der Name einer Stadt in der deutschen Transliteration das lateinische Alphabet benutzt, ist dies beispielsweise in der russischen Transliteration das kyrillische Alphabet (z.B. DE - Moskau, RU - Mockbá). Auch hier sollte die Software erkennen, dass es sich um das gleiche Element handelt, auch wenn die Eingaben unterschiedlich und in verschiedenen Sprachen sind.

KAPITEL 4

Verfahren zur Erzeugung von Testeingaben

Das folgende Kapitel stellt allgemeine Konzepte eines Testgenerators für die Erzeugung von Testfällen für Geodienste sowie seinen Aufbau vor. Neben der Bildung von Äquivalenzklassen für Testdaten werden unterschiedliche Verteilungen für die zu erzeugenden Testfälle diskutiert und bewertet. Hinweis: Für die Testfallerzeugung in dieser Arbeit wurde eine NDS-Datenbank verwendet.

4.1 Testgenerator

Der Abschnitt erklärt den allgemeinen Aufbau eines Testgenerators zum Testen von Geodiensten und auf welche Art und Weise er umgesetzt werden soll.

4.1.1 Überblick

Für die Konzipierung und Entwicklung eines Testgenerators für die Erzeugung von Testeingaben für Geodienste müssen mehrere Punkte betrachtet werden. Wichtig hierbei ist die Verteilung der Testfälle um eine ausreichende Überdeckung zu schaffen. Zudem muss konfigurierbar sein, wie die Testfälle aufgebaut werden. Dazu gehört neben dem Adressformat die Auswahl der Komponenten einer Adresse, welche für den Test verwendet werden sollen. Beispielsweise sollte konfigurierbar sein, ob die Testeingabe aus sämtlichen Komponenten der Adresse besteht oder lediglich aus den relevantesten Komponenten (z.B. Straße und Stadt).

Der Aufbau des in der Arbeit umgesetzten Testgenerators ist in Abbildung 4.1 dargestellt. Dabei stellt der grau hinterlegte Bereich die Komponenten des Testgenerator dar. Der Generator interagiert mit einer Adressdatenbank, welche verwendet wird, um real existierende Adressen zu erzeugen. Der Anwender steuert den Generator, indem er ihm Eingaben liefert. Dies sind Test-Region, Strategie, Testdatenanzahl und Eingabesprache. Das Programm erzeugt die entsprechenden Datenbankanfragen, welche durch die vorherige Konfiguration gebildet werden. Anschließend gibt das Programm eine Adressliste aus, die sämtliche Adresskomponenten enthält, sofern diese in der Datenbank vorhanden waren.

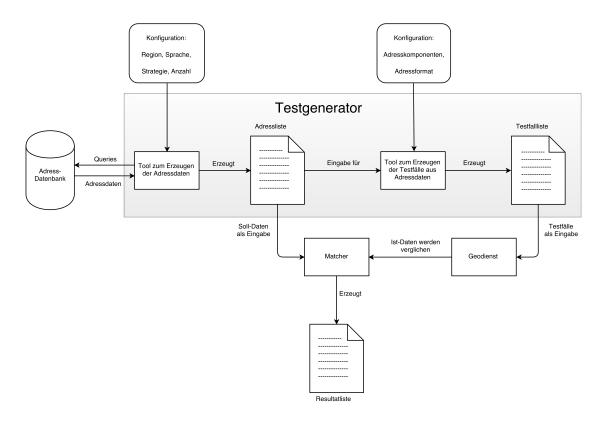


Abbildung 4.1.: Testgenerator Überblick

Aus dieser Liste muss nun ein weiteres Tool die Testfälle erzeugen. Dieses muss zuvor ebenfalls konfiguriert werden. Die Konfigurationseingaben bestehen aus dem (Adress-)Format, sowie die Auswahl der Komponenten, welche später im Testfall auftreten sollen. Die Ausgabe der Anwendung ist eine Testfallliste, welche neben den Testfällen die erwarteten Ausgaben, also die Soll-Daten, und eventuelle Ausführungsbedingungen (z.B. Eingabesprache) beinhaltet.

Die vom Generator erzeugten Testfälle können nun in einem anderen Prozess im Geodienst getestet und unter Umständen vergleichen werden (beispielsweise mit anderen Geodiensten).

4.1.2 Umsetzung

Beim Generator handelt es sich um einen datenbankbasierten Testdatengenerator. Die Datenbank ist zwingend notwendig, um real existierende Adressen zu erhalten und somit echte Testeingaben zu erzeugen. Zufällig generierte Adressdaten, die jedoch in der Realität nicht existieren sind sinnlos, da diese von den Geodiensten nicht gefunden werden können (es sei denn, sie würden zufällig eine gewisse Ähnlichkeit mit real existierenden Adressen aufweisen). Die vom Generator erzeugten Testdaten sind gleichzeitig die Soll-Daten für den späteren Vergleich. Aus diesen Testdaten muss der Generator für den Integrationstest zwischen Geocoding-Software und Geodatenbank nun Testeingaben generieren. Diese bilden zusammen mit den Soll-Daten und eventuellen Ausführungsbedingungen die Testfälle.

Der erste Schritt ist die Konfiguration des Generators durch den Tester. Er legt dabei

4.1. Testgenerator 33

fest, in welcher Region, beziehungsweise in welchem Gebiet die Testfälle erzeugt werden sollen. Beispielsweise kann sich dabei auf den nordamerikanischen Kontinent oder Europa beschränkt werden. Außerdem kann genauer spezifiziert werden, ob der gesamte Kontinent, ein Land, Bundesstaaten oder nur Städte für die Erzeugung betrachtet werden sollen. Eine weitere Konfigurationseingabe ist die Erzeugungsstrategie, welche für die Testfallerzeugung genutzt werden soll (siehe Abschnitt 4.5). Dabei wird festgelegt, ob die Verteilung zufällig in einer bestimmten politischen Region stattfinden soll. Der Tester kann sich jedoch auch für eine gleichverteilte geografische Erzeugung entscheiden, um sämtliche Gebiete in einer Region abzudecken. Vielleicht möchte der Tester jedoch auch eine komplette Stadt testen, dann steht ihm die Möglichkeit bereit, sämtliche Straßen eines Ortes mit dem Generator zu erzeugen, wobei Hausnummern in diesem Fall vernachlässigt werden. Da der Umfang dieser Tests jedoch enorm groß werden kann, besteht die Möglichkeit sich mit Hilfe der Äquivalenzklassenbildung Gruppen ähnlicher Eingaben zu erzeugen, welcher aber in der Erwartung die gleichen Ergebnisse liefern, um nur eine Teilmenge zu testen. Eine weitere Strategie erzeugt in einem Ort Adressen, in denen gewisse Straßennamen mehrmals vorkommen, damit getestet werden kann, ob auch alle unterschiedlichen Straßen (beispielsweise aus verschiedenen Stadtbezirken) gefunden werden können. Der Generator besitzt weitere Erzeugungsstrategien, um beispielsweise geordnete oder randomisierte Testfälle in den unterschiedlichen politischen Verwaltungseinheiten (z.B Kontinent, Land, Bundesstaat, Stadt) oder einer geografischen Region (z.B. Umkreis um einen Punkt) zu erzeugen. Nachdem die Strategie für die Erzeugung festgelegt wurde, kann der Tester den Generator für eine bestimmte Sprache beschränken. Damit kann beispielsweise ausgewählt werden, Testfälle nur in englischer Sprache zu erzeugen (z.B. Munich anstatt München). Zu guter Letzt kann konfiguriert werden, wie viele Testdaten für die jeweilige Erzeugungsstrategie gewünscht sind.

Nach der Konfiguration werden die Datenbankanfragen erzeugt, die Testdaten extrahiert und eine Adressliste erstellt. Diese enthält sämtliche Adresselemente einzeln aufgeschlüsselt (Beispiel in Tabelle 3.1). Dabei müssen nicht alle Elemente in der Tabelle sein (z.B. wenn zu einer Straße keine Hausnummern existieren). Diese Adressliste ist nun neben einer weiteren Konfiguration durch den Tester die Eingabe für ein weiteres Tool, welches aus der Adressliste eine Testfallliste erzeugt. Die Konfiguration enthält dabei einerseits das Adressformat, in welchem die Adresselemente zu einem kompletten String zusammengefügt werden sollen, also die Anordnung der Elemente (siehe Abschnitt 2.4.2). Andererseits gibt der Tester an, welche Adresselemente überhaupt verwendet werden sollen. Er kann sich beispielsweise dafür entscheiden, sämtliche Elemente zu nutzen, um die Adresse so präzise und ausführlich wie möglich zu machen. Er kann jedoch auch nur die nötigsten Elemente (z.B. Straße und Stadt) verwenden, um zu schauen, ob der Dienst auch mit einer Eingabe mit wenig Adresselementen zurecht kommt. Daraus resultieren unterschiedliche Testfälle mit teilweise unterschiedlichen erwarteten Ergebnissen. Diese Strings stellen später die Testfalleingaben dar. Unter Umständen kann der Tester zusätzlich festlegen, ob Elemente einer Adresse modifiziert werden sollen, indem beispielsweise Zeichen ersetzt werden (z.B. "Otto-Suhr-Alle" wird zu "Otto Suhr Alle") oder aber Teile eines Elementes weggelassen werden (z.B. entfernen des Token "Straße"), um zu erkennen, ob die Software veränderte Eingaben und Präfixe erkennt. Die Soll-Daten für die Testfälle werden direkt aus den Adressdaten der bereits vorhandenen Adressliste gewonnen. Zu guter Letzt können Ausführungsbedingungen festgelegt werden. Das kann beispielsweise die zuvor konfigurierte Eingabesprache sein, welche jedoch nicht bei allen Diensten Voraussetzung sein muss.

Nach der Erzeugung werden die Testfälle in den jeweiligen Geodiensten getestet. Dies kann durch unterschiedliche Tools geschehen. Anschließend können die Resultate mit den Soll-Daten abgeglichen werden, um festzustellen, ob die Tests erfolgreich waren. Zudem können die Ergebnisse unterschiedlicher Dienste verglichen werden.

4.2 Datenbank Hintergrundwissen

Für die Testfallerzeugung wird eine geeignete Datenbank benötigt. Diese muss real existierende Adressen enthalten, aus welchen die Testdaten gebildet werden können. Um auf die Datenbank zuzugreifen, wird das Wissen über das Datenbankschema und die Inhalte benötigt. In OpenStreetMap (OSM) sind die Adressdaten, wenn sie denn existieren je nach Datenimport unterschiedlich gespeichert. In der Regel entspricht jedoch eine Zeile in den Tabellen einer Adresse. Dort müssen dann lediglich die entsprechenden Spalten extrahiert werden. In anderen Datenbanken sind die Adressdaten vielleicht auf andere Art und Weise abgelegt. Ein hierarchischer Ansatz ist denkbar.

Des Weiteren muss geklärt werden, ob Sonderfälle wie Kreuzungen extra gespeichert sind, falls diese einen für die Testfälle interessieren. Eventuell gibt es Tabellen, welche diese Informationen speichern. Unter Umständen müssen diese Daten aber auch mittels Geocoding bestimmt werden, wenn sie nicht extra abgelegt sind.

Zuletzt bleibt die Frage, ob auch wirklich alle wichtige Informationen in der Datenbank abgelegt sind, oder ob es Dinge gibt, die fehlen. So ist beispielsweise bei OSM oft der Bundesstaat nicht in der Adressinformation enthalten.

Wenn bekannt ist, wie sämtliche notwendigen Informationen abgelegt sind, kann begonnen werden die Datenbankzugriffe für die entsprechenden Erzeugungsstrategien zu entwerfen.

4.3 Regionen Hintergrundwissen

Auch wenn einige Anwendungsfälle für den Nutzer normale Eingaben darstellen, können diese für den Geodienst Spezialfälle sein. Das kann beispielsweise durch unterschiedliche Adressformate, aber auch Unterschiede in den Formaten der einzelnen Adresselemente geschehen.

Vor allem spezielle Postleitzahlen können hierbei gleiche Formate haben wie Hausnummern, sodass die Software des Geodienstes unterscheiden muss, um welches Elemente es sich dabei handelt. Postleitzahlen können in den unterschiedlichen Ländern dabei die verschiedensten Formate haben. Meist sind sie lediglich ein numerischer String mit vier bis sechs Zeichen. Allerdings kann der String auch alphanumerisch und durch Leerzeichen oder Bindestriche getrennt sein [33]. Im Vereinigten Königreich bestehen die Postleitzahlen beispielsweise immer aus zwei alphanumerischen Token, welche zudem unterschiedliche Längen haben können. Ähnlich sehen die Postleitzahlen in Kanada aus. In einigen Ländern, wie Polen und Portugal sind die Zahlen durch Bindestriche getrennt, in Griechenland und der Tschechischen Republik hingegen durch ein Leerzeichen. Hier kann die Unterscheidung zwischen Postleitzahltoken und Hausnummerntoken für die Software des Geodienstes eine schwere Aufgabe werden.

Hausnummern können in den verschiedenen Ländern ebenfalls unterschiedliche Formate

haben (z.B. numerisch oder alphanumerisch) und besitzen teilweise keine allgemeine Struktur. Deshalb kann auch hier die Erkennung der Hausnummern ein Spezialfall für die Software werden.

Für die Software ist die Erkennung dieser Element eventuell einfacher, wenn die zu suchende Adresse im landesüblichen Adressformat eingegeben wird. So kann beispielsweise in den USA angenommen werden, dass der Nutzer die Hausnummer vor dem Straßennamen eingibt, sodass diese nicht mit der Postleitzahl verwechselt wird, wenn die Hausnummer ebenfalls fünf Zeichen lang ist. Deshalb ist es sinnvoll verschiedene Formate der unterschiedlichen Länder zu testen.

4.4 Bildung der Äquivalenzklassen

Mit Hilfe von Äquivalenzklassen können Eingaben anhand ihrer Struktur unterteilt werden. Es wird bei Tests von Elementen aus derselben Äquivalenzklasse das gleiche Verhalten erwartet. Für das Testen von Adressen in Geodiensten können mit Hilfe der Äquivalenzklassen beispielsweise unterschiedliche Formate von Straßenbezeichnungen in einer Stadt getestet werden. Damit wird sichergestellt, dass sämtliche Formate und Strukturen korrekt erkannt werden. Dieser Vorgang eignet sich beispielsweise, um Straßennamen, welche Zahlen oder Sonderzeichen beinhalten, zu testen.

Für die Bildung der Äquivalenzklassen wird ein Vorgang benötigt, welcher gewisse Testfälle findet, die ein ähnliches Verhalten aufweisen. Es gibt eine Reihe Algorithmen, welche entweder dem überwachten Lernen (supervised learning) oder zum unüberwachten Lernen (unsupervised learning) [34] zugeordnet werden.

Supervised learning Methoden müssen mit Beispieldaten trainiert werden, um Vorhersagen tätigen zu können. Zu diesen Methoden gehören unter anderem neuronale Netzwerke oder Entscheidungsbäume.

Unsupervides learning Methoden hingegen benötigen kein Training. Dazu gehören Clustering-Algorithmen, welche die Daten in disjunkte Gruppen (Cluster) unterteilen.

Da für die Äquivalenzklassenbildung keinerlei Trainingsdaten vorhanden sind, ist es sinnvoll einen Clustering-Algorithmus zu verwenden. Somit können die Testfalldaten in disjunkte Cluster geordnet werden, welche im späteren Verlauf als Äquivalenzklassen dienen. Aus diesen Klassen können dann ein oder mehrere Repräsentanten als Testfall gewählt werden. Ein relativ simpler Ansatz ist der K-Means-Algorithmus zum Clustern von Daten.

4.4.1 K-Means-Algorithmus

Beim K-Means-Clustering werden anfangs k zufällig gewählte Zentroide in den Daten verteilt. Anschließend wird jeder Datenpunkt dem nächsten Zentroiden zugeordnet. Darauf werden die Zentroiden versetzt, sodass sie in der Mitte ihrer zugeordneten Datenpunkte stehen. Dann wiederholt sich der Vorgang, es werden also wieder alle Datenpunkte dem nächsten Zentroiden zugeordnet, welcher nochmals in die Mitte aller Punkte versetzt wird. Dieser Vorgang wird so lange wiederholt, bis keine Änderung mehr auftritt oder eine festgelegte Anzahl an Iterationen erreicht ist.

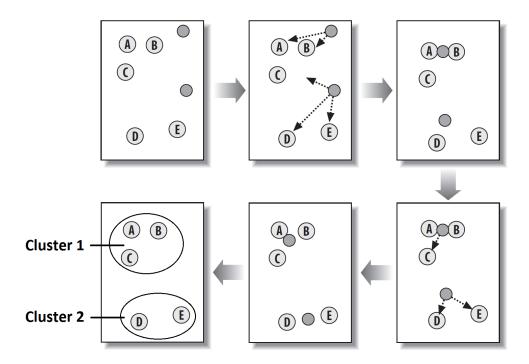


Abbildung 4.2.: K-Means Algorithmus [34]

Der wichtige Aspekt bei diesem Algorithmus ist die Erzeugung eines numerischen Vektors, welcher dem K-Means-Algorithmus als Eingabe dient. Für das Clustern von Adressdaten müssen diese also in einen geeigneten Vektor überführt werden, welcher die Daten möglichst genau spezifiziert und deren charakteristischen Eigenschaften widerspiegelt. Die Aufgabe hierbei besteht darin, eine Funktion zu finden, welche die Adressdaten in einen numerischen Vektor überführen kann, dass diese sinnvoll aufgeteilt werden können und die Adressdaten später korrekt in die entsprechenden Äquivalenzklassen überführt werden.

4.4.2 Überführung der Adressdaten in einen Vektor

Für die Überführung der Adressdaten in einen Vektor, welcher als Eingabe für den K-Means-Algorithmus dient, ist es nötig zu wissen, welche Faktoren ausschlaggebend für die Adressen sind. Dazu muss die genaue Struktur betrachtet werden. Werden beispielsweise lediglich die Straßennamen begutachtet, fallen Unterschiede zwischen den einzelnen Bezeichnungen bezüglich ihrer Struktur und verwendeter Zeichen auf.

Im deutschen Sprachraum können beispielsweise folgende Besonderheiten auftreten:

- Verwendung "normaler" Buchstaben aus dem ASCII Bereich (lateinisches Alphabet)
- Verwendung von Buchstaben (z.B. Umlaute) aus dem nicht-ASCII Bereich (z.B. UTF-8 kodierte Unicode Zeichen wie "ä", "ö", "ü", "ß", "B")
- Verwendung von Punkten und/oder Bindestrichen (z.B. "Dr.-Markus-Straße" oder "Otto-Suhr-Allee")
- Andere Sonderzeichen (z.B. "'" oder "/")

- Straßennamen, welche aus mehreren Token bestehen (z.B. "Alte Potsdamer Straße")
- Straßen welche Zahlen enthalten (z.B. "Straße 339" oder "Straße des 17. Juni")
- Besonders lange Straßennamen (z.B. "Platz der Vereinten Nationen")
- Besonders kurze Straßennamen (z.B. "B5")

Diese Besonderheiten müssen also in einer Überführungsfunktion erkannt und in einer Zahl repräsentiert werden. Damit können die numerischen Vektoren für alle Straßennamen errechnet werden, welche der K-Means-Algorithmus anschließend zum Clustern verwenden kann.

Überführungsfunktion

Die Überführungsfunktion wird benötigt, um aus einem Wort einen Wert oder einen Vektor zu bilden, der die charakteristischen Eigenschaften des Wortes wiedergibt. Einige dieser Eigenschaften wurden bereits im oberen Abschnitt beschrieben. Diese sind allerdings sprachund landesabhängig. Für die Überführung müssen die Besonderheiten erkannt und gezählt werden. Anschließend werden sie in einem Vektor festgehalten. Dieser dient dann als Eingabe für den K-Means-Algorithmus, welcher damit eine Clusterisierung durchführt. Der Vektor enthält Einträge für die Anzahl der ASCII Zeichen und der UTF-8 kodierten 2- und 3-Byte Zeichen sowie die Anzahl der Bindestriche, Punkte, Leerzeichen, sonstigen Sonderzeichen und Zahlen. Für eine geeignete Verteilung ist es sinnvoll die einzelnen Besonderheiten unterschiedlich zu gewichten. Dazu wird die Anzahl jeder Eigenschaft zusätzlich mit einem Wert 2^i mit $i \in 0...7$ multipliziert.

Ein Vektor beinhaltet 8 Werte und ist wie folgt aufgebaut:

[ASCII-Zeichen, UTF-8 2 Byte Zeichen, UTF-8 3 Byte Zeichen, Punkte, Bindestriche, Leerzeichen, Zahlen, Sonderzeichen]

Beispielvektoren für verschiedene Szenarien:

• Akazienallee (ASCII-Zeichen)

[12, 0, 0, 0, 0, 0, 0, 0]

Beispiele aus diesem Cluster: Adlergestell, Birkenstasse, Fliederallee

• Takustraße (ASCII-Zeichen + UTF-8 2 Byte Zeichen)

[9, 2, 0, 0, 0, 0, 0, 0]

Beispiele aus diesem Cluster: Lärchenweg, Parkstraße, Ringstraße

ullet Otto-Suhr-Allee (ASCII-Zeichen + Bindestrich)

[15, 0, 0, 0, 32, 0, 0, 0]

Beispiele aus diesem Cluster: Carl-Herz-Ufer, Otto-Hahn-Platz, Zwerg-Nase-Weg

• Alte Potsdamer Straße (ASCII-Zeichen + UTF-8 2 Byte Zeichen + Leerzeichen)

[20, 2, 0, 0, 0, 64, 0, 0]

Beispiele aus diesem Cluster: An den Siedlergärten, Straße zum Müggelturm, Zur Neuruppiner Straße

• Straße 339 (ASCII-Zeichen + Zahl + Bindestrich)

[9, 2, 0, 0, 0, 32, 192, 0]

Beispiele aus diesem Cluster: Fußweg 469, Straße 100

• Platz der Vereinten Nationen (ASCII-Zeichen + Leerzeichen)

[28, 0, 0, 0, 0, 96, 0, 0]

Beispiele aus diesem Cluster: Avenue Charles de Gaulle, Platz vor dem Brandenburger Tor

• B5 (ASCII-Zeichen + Zahl)

[2,0,0,0,0,0,64,0]

Beispiele aus diesem Cluster: B1, B2

Diese Beispiele zeigen, dass die unterschiedliche Struktur der Straßennamen zu verschiedensten Vektoren führt. Damit kann der K-Means-Algorithmus eine effektive Clusterisierung vornehmen und die Eingaben anhand ihrer Struktur in unterschiedliche Cluster einordnen.

Diese Methode kann zum Beispiel genutzt werden, um sämtliche Straßenformate einer Region (z.B. einer Stadt) zu testen. Für die Bewertung sprachlicher Besonderheiten in Adressen kann beispielsweise die größte Stadt im Land (in vielen Fällen gleichzeitig die Hauptstadt) verwendet werden. Dafür werden sämtliche Straßen in dieser Region vom Testdatengenerator erzeugt. Anschließend wird auf die erzeugten Daten die Äquivalenzklassenbildung ausgeführt. Somit werden gleiche Formate in die Klassen eingeordnet. Am Ende kann aus jeder Klasse ein Element entnommen werden, welches sämtliche Elemente seiner Klasse repräsentiert, um daraus einen Testfall zu erzeugen. Dieser kann anschließend getestet werden. Es wird erwartet, dass alle anderen Klassenelemente bei einem Test das gleiche Verhalten haben. Somit kann nach dem Test die Klasse und somit auch deren Elemente bewertet werden. Der Vorteil hierbei ist, dass besondere Fälle, welche nicht oft vorkommen, trotzdem getestet und somit abgedeckt werden.

Für eine Beispielclusterberechnung sämtlicher Berliner Straßen mit der vorgestellten Kostenfunktion, ergeben sich im Schnitt 110 Cluster. Die ursprünglichen Daten umfassen dabei nahezu 14.900 Einträge. Viele der Straßen Berlins sind sich also so ähnlich, dass es sinnvoll ist, nur ein Elemente aus jedem Cluster zu testen, um damit den Testumfang zu reduzieren.

Unter Umständen kann es jedoch sinnvoll sein, die Überführungsfunktion für andere Regionen anzupassen.

4.5 Testfall-Verteilungsstrategien

Wie lassen sich Testfälle sinnvoll verteilen? Um später Aussagen über die Abdeckung der Tests zu machen, ist es wichtig, sich Gedanken um die Verteilung der Testfälle zu machen. Was nützt es, eine komplette Stadt zu testen, welche viele Testfälle enthält, wenn dann aber keinerlei Aussagen über den Rest des Landes, in welchem die Stadt liegt, getroffen werden kann. Ist es nützlich Testfälle lediglich für ein Land zu erstellen? Was kann dann mit diesen Ergebnissen für eine größere Region wie einen Kontinent gesagt werden? Kann ein kompletter Kontinent getestet werden oder ist dies aufgrund der Größe unmöglich?

Diese Testfallverteilung und dazu gehörigen Fragestellungen sind Thema des folgenden Abschnittes.

Die Karten aus den folgenden Abschnitten wurden aus Adressdaten und deren Geoinformationen erstellt, welche zuvor mit dem umgesetzten Testgenerator erzeugt wurden.

4.5.1 Zufällige Verteilung

Für diese Verteilungsstrategie werden die Testfälle zufällig in einer bestimmten politischen Region erzeugt. Solch eine Region kann ein kompletter Kontinent sein, oder aber ein Land, ein Bundesland oder eine Stadt. In diesen Regionen können die Testfälle dann zufällig verteilt werden. Je nach Größe und Bevölkerungsdichte der Regionen kann diese Verteilung besser oder schlechter ausfallen. Gibt es in einem Land beispielsweise einen flächenmäßig großen Bereich, der gering besiedelt und einen flächenmäßig kleinen Bereich, der sehr dicht besiedelt ist, dann kann die Verteilung zugunsten des dicht besiedelten Bereichs ausfallen.

Eine weitere Möglichkeit ist beispielsweise die Verteilung nach Gemeinden in einem Land. So kann für jede Gemeinde ein Testfall erzeugt werden, um jede Stadt abzudecken. Das kann jedoch je nach Größe des Landes einen hohen Testaufwand bedeuten. Zudem würden Gemeinden mit wenigen hundert Einwohnern einer Großstadt mit mehreren Millionen Einwohnern gleichgestellt werden. Um dem entgegenzuwirken könnte die Bevölkerungsanzahl mit einbezogen werden, sodass eine höhere Bevölkerungszahl mehr Testfälle für eine Gemeinde bedeutet. Hierbei würde der Testaufwand jedoch noch weiter steigen.

Wichtig ist der Vergleich der Tests in politischen Regionen. Wenn für unterschiedliche Regionen die gleiche Anzahl an Testfällen erzeugt werden, macht es dann Sinn die Ergebnisse zu vergleichen, wenn eine Region sehr klein ist, die andere aber sehr groß (z.B. das Bundesland Bremen im Gegensatz zum Bundesland Bayern)? Können für die Untersuchung auf Ebene der Bundesländer die Ergebnisse für das flächenmäßig kleinste und größte Bundesland verglichen werden, wenn in beiden Gebieten zum Beispiel 1000 Testfälle erzeugt wurden? Während im flächenmäßig kleinsten Gebiet dabei ein großer Anteil der Adressen getestet wurde, so ist im größten Gebiet wahrscheinlich nur ein Bruchteil getestet worden.

Im umgesetzten Testgenerator können Testfälle zufällig in politischen Regionen erzeugt werden. Diese Region kann die komplette Karte sein, aber auch Länder, Bundesländer oder Städte.

4.5.2 Geografische Verteilung

Neben einer rein zufälligen Verteilung, kann eine geografische Verteilung genutzt werden. Dazu sollten die Testfälle sinnvoll auf der Karte verteilt werden, um eine möglichst hohe Abdeckung zu erreichen. Die Tests können komplett zufällig auf einer Karte verteilt werden. Dann ist jedoch die Möglichkeit sehr hoch, dass bestimmte Regionen gar nicht getestet werden. Das kann beispielsweise auftreten, wenn es in einer Region (z.B. ein Land) sehr wenige stark bevölkerte und viele schwach bevölkerte Gebiete gibt. Dann ist die Wahrscheinlichkeit sehr hoch, dass lediglich die dicht besiedelten Gebiete von den Tests abgedeckt werden, die anderen Gebiete hingegen überhaupt nicht.

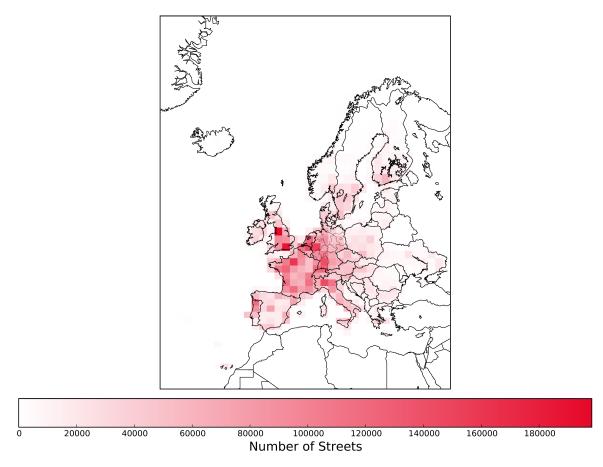
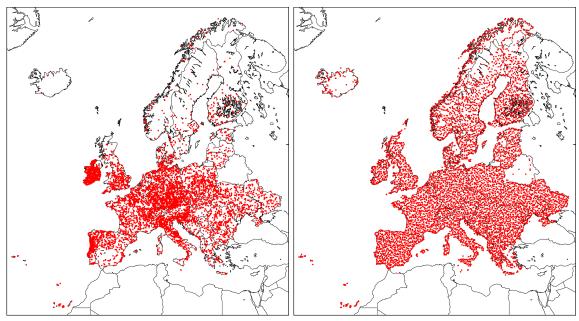


Abbildung 4.3.: Heatmap - Straßenverteilung Europa

Um klar zu machen, wie sinnvoll die gleichmäßige Verteilung ist, können Heatmaps betrachtet werden, welche die Verteilung von Straßen auf einer Karte darstellen. In Abbildung 4.3 ist die Straßenverteilung in Europa dargestellt. Je intensiver die rötliche Färbung ist, umso mehr Straßen existieren in diesem Gebiet. Die blassen, weißen Gebiete zeigen an, dass hier nur wenig Straßen vorhanden sind. Beim Betrachten der Karte wird deutlich, dass die Straßendichte in Zentraleuropa am höchsten ist. In den nördlichen Staaten hingegen nimmt die Anzahl der Straßen ab. Auch die östlichen Länder Europas haben eine geringere Straßendichte. Würde eine zufällige Verteilung gewählt, so wären die meisten Testfälle mit hoher Wahrscheinlichkeit in den zentralen Regionen Europas zu finden und nördliche sowie östliche Gebiete würden vernachlässigt.

Der umgesetzte Testgenerator versucht dem entgegenzuwirken, indem die Karte in Tiles gegliedert wird (siehe Abschnitt 2.5). Somit ist es möglich gezielt aus jedem Tile eine zuvor festgelegt Anzahl von Testfällen zu erzeugen, da jedem Testfall die Zugehörigkeit zu einem Tile anhand seiner Koordinaten eindeutig zugeordnet werden kann. Die Auswahl der Fälle in den einzelnen Tiles geschieht dabei zufällig. Die Verteilung kann mit höherem Tile Level und somit geringerer Tile-Größe erweitert werden. Der Generator kann die geografisch gleichverteilten Testfälle für eine komplette Karte, ein Land oder einen Bundesstaat erzeugen.

In Abbildung 4.4 wird die zufällige und die geografische Verteilung in Europa in den Tiles



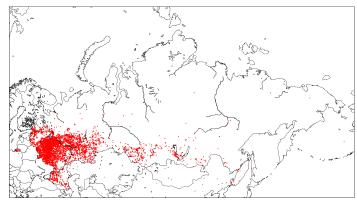
- (a) Zufällige Verteilung in Europa (6.808 Testfälle) (b) Gleichmäßige Verteilung in den 6.808 Tiles in Europa

Abbildung 4.4.: Unterschiedliche Verteilungen von Testfällen in Europa

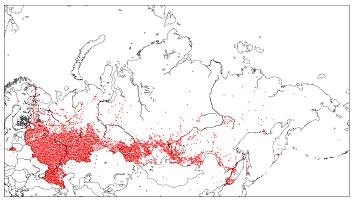
(Level 9) dargestellt. Dafür wurden Testfälle mit dem umgesetzten Generator erzeugt und visualisiert. Ein Punkt auf der Karte stellt dabei einen Testfall dar. Hinweis: Weißrussland enthält so wenig Testfälle, da die verwendete Datenbank hier nur sehr spärlich gefüllt ist.

Da es in Europa 6.808 Tiles gibt, in denen Testfälle existieren, wurde für die zufällige Verteilung die gleiche Anzahl verwendet. Wie erwartet sind bei der zufälligen Verteilung (Abbildung 4.4a) die meisten Testfälle in Zentraleuropa zu finden. In Skandinavien sind nur wenige Testfälle vorhanden, in Island sind es sogar nur 3 Testfälle. Im östlichen Bereich Europas sind entgegen der ersten Annahmen relativ viele Testfälle zu finden. Auffällig ist die hohe Anzahl an Tests in Irland. Bei Betrachtung der geografisch gleichmäßigen Verteilung (Abbildung 4.4b) fällt sofort auf, dass hier sämtliche bevölkerten Gebiete abgedeckt sind. Es gibt noch immer weiße Flecken auf der Karte, welche aus dem Umstand resultieren, dass diese Gebiete nicht besiedelt sind und dort somit keine Testfälle erzeugt werden können.

Ähnliche Beobachtungen können in einer Beispieltestfallerzeugung in Russland gemacht werden (Abbildung 4.5). Da weite Teile des Landes nicht oder nur sehr gering bevölkert sind (z.B. Sibirien), ist bei einer rein zufälligen Verteilung der Testfälle die Wahrscheinlichkeit sehr hoch, dass in diesen Regionen keine oder nur sehr vereinzelt Testfälle erzeugt werden. Die zufällige Verteilung (Abbildung 4.5a) zeigt, dass vor allem im Westen und Südwesten Russlands die meisten Testfälle erzeugt wurden, während in Sibirien nur sehr wenig Fälle auftreten. In den südlichen Grenzen Russlands hingegen wurden ebenfalls Test erzeugt. Bei der gleichmäßigen Verteilung in den Tiles (Abbildung 4.5b) sind die Tests deutlich ausgeglichener. Hier überdecken sich die Tests nicht gegenseitig wie bei der zufälligen Verteilung. Außerdem gibt es wesentlich mehr Fälle in Sibirien, an der Grenze zu Finnland



(a) Zufällige Verteilung in Russland (5.419 Testfälle)



(b) Gleichmäßige Verteilung in den 5.419 Tiles in Russland

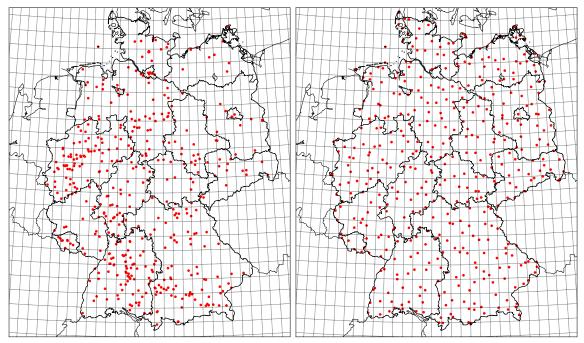
Abbildung 4.5.: Unterschiedliche Verteilungen von Testfällen in Russland

und an den südlichen Grenzen.

Der Vorteil in der Methodik liegt darin, dass selbst sehr gering besiedelte Gebiete getestet werden können. Wenn es beispielsweise innerhalb einer $1.000\,\mathrm{km^2}$ großen Fläche lediglich einen Ort mit fünf Straßen gibt, so ist es durch diese Methode möglich eine dieser Straßen als Testfall zu erhalten. Die Wahrscheinlichkeit eine der Straßen mit einer Zufallsmethode zu erhalten, ist hingegen sehr gering.

Auch für eine Erzeugung durch den Testgenerator in Deutschland wird deutlich, dass bei der zufälligen Verteilung (Abbildung 4.6a) einige Gebiete sehr wenig getestet werden, andere hingegen wesentlich mehr. Beispielsweise sind in Rheinland-Pfalz, Mecklenburg-Vorpommern und Brandenburg viele Gebiete ohne Testfall, während in Nordrhein-Westfalen, Baden-Württemberg und Bayern sehr viele Testfälle vorhanden sind. Insgesamt sind nur 234 der 440 Tiles abgedeckt. In der geografisch gleichmäßigen Verteilung (Abbildung 4.6b) sind hingegen alle 440 Tiles durch einen Testfall abgedeckt. In dieser Verteilungsmethode werden somit keine Regionen bevorzugt, wenn sie eine hohe Straßendichte besitzen und somit viele mögliche Testfälle enthalten. Gleich große Regionen enthalten ungefähr die gleiche Anzahl an Testfällen, größere Regionen enthalten mehr, kleinere Regionen enthalten weniger.

Um mehr geografisch gleichverteilte Testfälle zu erhalten, muss lediglich die Anzahl der Tiles erhöht werden. Dafür wird ein größeres Tile-Level gewählt, womit sich die Tile-Größe



(a) Zufällige Verteilung von 440 Testfällen in Deutsch-(b) Geografisch gleichmäßige Verteilung in den 440 land (234 Tiles durch Testsfall abgedeckt)

Tiles (Level 9) in Deutschland

Abbildung 4.6.: Zufällige und geografisch gleichmäßgige Verteilung in Deutschland

verringert. Mit jeder Tile-Level Erhöhung steigt die Tile-Anzahl um das Vierfache. In Deutschland wären somit anstatt 440 Tiles auf Level 9 ungefähr 1760 Tiles auf Level 10 vorhanden. Die Bezeichnung ungefähr resultiert daraus, dass nicht jedes der neuen Tiles auf Level 10 einen Teil von Deutschland überdecken muss. Lag beispielsweise ein Level 9 Tile an der Grenze zu Deutschland und es enthielt nur zu einem Viertel deutsches Staatsgebiet, so werden die daraus resultieren vier kleineren Level 10 Tiles nicht alle im deutschen Staatsgebiet liegen.

Diese Methode ist besonders gut geeignet, wenn die Straßenverteilung (und somit auch die Testfallverteilung) regional sehr unterschiedlich für ein Gebiet ausfällt. Die Strategie kann vor allem dabei helfen wenig besiedelte Gebiete trotzdem ausreichend zu testen und sorgt dafür, dass diese nicht von den stark besiedelten Regionen dominiert werden.

Die geografische Verteilung deckt sämtliche Tiles der zu testenden Region ab. Auf dem Tile Level 9 ist die maximale Entfernung zwischen zwei Punkten, also zwei Testfällen, ungefähr 87.41 km (siehe Abschnitt 2.6). Dieser Abstand schrumpft mit steigendem Breitengrad. Mit Erhöhung des Tile Levels kann die Anzahl der zu erzeugenden Testfälle noch gesteigert werden, wobei hier auch der maximale Abstand zwischen den Fällen verringert wird. Jedoch kann es sein, dass aufgrund der zufälligen Auswahl der Testfälle in den Tiles große Städte noch immer untergehen und somit nicht getestet werden.

Die geografische Verteilung eignet sich besonders gut um die Abdeckung der Datenbanken zu studieren. Schlagen Testfälle in geringer besiedelten Gebieten fehl, so ist die Wahrscheinlichkeit hoch, dass die Adresse nicht in der Datenbank des zu untersuchenden Geodienstes vorhanden ist. Des Weiteres kann hiermit überprüft werden, ob sämtliche Gebiete mit der gleichen Genauigkeit in der Datenbank erfasst sind oder ob einzelne Adresselemente wie Hausnummern für einige Gebiete fehlen.

Die vorgestellte geografische Verteilung hat jedoch nicht nur Vorteile: Die Testfälle sind mit dieser Methode zwar sehr gleich verteilt, allerdings betrachtet sie nicht die Anzahl der Straßen. In der Darstellung einer geografischen Verteilung in den Tiles in Deutschland (Abbildung 4.6b) wird deutlich, welche Auswirkungen das haben kann. In der Hauptstadt Berlin ist beispielsweise nur ein Testfall zu finden. Die Stadt wird von vier Tiles abgedeckt, in zwei Tiles sogar zu einem sehr großen Anteil. Jedoch haben wir nur in einem dieser Tiles einen Testfall, da in den anderen drei Tiles auch Teile vom Bundesland Brandenburg liegen. Es wäre jedoch wünschenswert mehr Testfälle in der Hauptstadt zu haben, da das Straßennetz Berlins wesentlich dichter ist, als das Brandenburger Straßennetz. Somit existieren hier viele Adressen, die getestet werden könnten. Wird die Anzahl der Straßen verglichen (Berlin: 14.873, Brandenburg: 53.416), so gibt es in Brandenburg 3,5 mal so viele Straßen wie in Berlin, welche sich aber auf eine circa 33 mal größere Fläche aufteilen (Berlin: circa 891 km². Brandenburg: circa 29.654 km²). Es wäre also sinnvoll die Anzahl der Straßen in einem Tile in die Testfallerzeugung mit einzubeziehen. Deutlich wird dieser Nachteil ebenfalls bei der Betrachtung der Stadtstaaten Hamburg und Bremen. Bei der geografischen Verteilung gibt es im Beispiel aus Abbildung 4.6b keine Testfälle in diesen Städten, da Fälle aus der Umgebung gewählt wurden.

4.5.3 Geografische Verteilung mit Einbeziehen der Testfalldichte

Da bei der geografischen Verteilung teilweise wichtige Testfälle untergehen, ist es sinnvoll zusätzliche Informationen in die Erzeugung mit einfließen zu lassen, um die Verteilung besser zu gestalten. Diese Information kann beispielsweise die Bevölkerungsdichte sein. Die Strukturierung anhand der Bevölkerungsdichte ist jedoch schwierig, da sie keine Tilegrenzen kennt. Zudem ist diese Information wahrscheinlich nicht in einer Adressdatenbank vorhanden. Eine andere Möglichkeit ist die Verwendung der Testfalldichte. Dazu muss eine Zuordnung von Adressen zu den Tiles bestehen. Da jeder Straße eine Geokoordinate zugeordnet ist, kann sie eindeutig in ein Tile eingeordnet werden. Mit dieser Information kann die Dichte der Straßen pro Tile und somit auch die Testfalldichte pro Tile ermittelt werden. Wenn nun die Testfalldichte in einem Tile sehr hoch ist, können in diesem mehr Testfälle erzeugt werden, als in einem Tile mit geringerer Dichte. Damit wird sichergestellt, dass sämtliche Regionen abgedeckt sind, auch wenn diese nur eine geringe Dichte aufweisen.

Der umgesetzte Generator bezieht für diese Strategie die Testfalldichte sämtlicher Tiles eines Gebietes mit ein. Diese Gebiete kann die komplette Karte, ein Land oder ein Bundesstaat sein. Dafür wird anfangs der Wert n festgelegt, welcher die maximale Anzahl an Testfällen pro Tile beschreibt. Es wird ermittelt, wie viele Testfälle in jedem Tile existieren. Diese Werte werden anschließend normiert, sodass jedem Tile ein Wert zwischen 1 und n zugeordnet werden kann. Dieser Wert ist die Anzahl der zu erzeugenden Testfälle pro Tile. Eine Beispieltestfallerzeugung durch den Testgenerator für die geografische Verteilung mit Einbeziehung der Testfalldichte (Dichteverteilung) in Deutschland ist in Abbildung 4.7 dargestellt. Hierbei wurde die maximale Anzahl der zu erzeugenden Testfälle pro Tile auf 10 gesetzt. Insgesamt erzeugt die Strategie 1162 Testfälle in Deutschland.

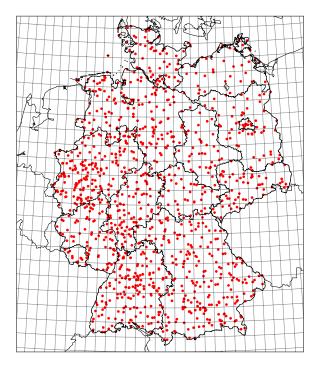


Abbildung 4.7.: Geografisch gleichmäßige Verteilung inklusive Berücksichtigung der Testfalldichte in den 440 Tiles (Level 9) in Deutschland (mit eingezeichneten Tiles, 1162 Testfälle)

Die Abbildung zeigt deutlich, dass in der geografischen Verteilung in jedem Tile mindestens ein Testfall erzeugt wurde. Besitzt ein Tile eine höhere Testfalldichte, so werden hier mehr Tests erzeugt, maximal jedoch 10. Die Visualisierung zeigt, dass weniger bevölkerte Regionen wie Mecklenburg-Vorpommern oder Teile Brandenburgs und Sachsen-Anhalts meist nur einen Testfall pro Tile besitzen. In stärker bevölkerten Bundesländern wie Nordrhein-Westfalen werden mit der Dichteverteilung mehr Testfälle erzeugt. Die Erzeugungsstrategie generiert in den Stadtstaaten Berlin, Hamburg und Bremen ebenfalls mehr Testfälle, sodass diese besser abgedeckt werden als bei der geografischen Verteilung.

Die Kombination aus geografischer Erzeugung und Berücksichtigung der Testfalldichte in den Tiles stellt somit einen guten Kompromiss dar, um Gebiete strukturiert zu testen. Um die Anzahl der Testfälle zu steigern, kann hierbei einerseits die maximale Anzahl der Testfälle pro Tile erhöht werden und andererseits das Tile Level vergrößert werden, womit die geografische Verteilung weiter erhöht wird.

4.5.4 Zwischenfazit

Letztendlich können die unterschiedlichen Verteilungsstrategien für Testfälle verglichen werden. Eine zufällige Verteilung von Testfällen in einer politischen Region ist in der Regel wenig sinnvoll. Hierbei kann es vorkommen, dass bestimmte Bereiche (z.B. gering bevölkerte Regionen) gar nicht getestet werden und andere Gebiete hingegen sehr viele Fälle enthalten. Zudem ist die Aufteilung teilweise nicht sehr sinnvoll, wenn in einer sehr großen politischen Region die gleiche Anzahl an Testfällen erzeugt wird wie in einer sehr kleinen politischen Region. Um dem entgegen zu wirken und in allen Regionen nahezu gleichverteilte Testfälle

zu erzeugen, kann die geografische Verteilungsstrategie genutzt werden. Dabei wird die zu testende Region in Tiles untergliedert, in welchen die gleiche Anzahl an Testfällen erzeugt werden. Somit ist sichergestellt, dass jeder Abschnitt der Region durch mindestens einen Testfall abgedeckt ist. Für eine höhere Dichte kann die Testfallanzahl pro Tile oder das Tile Level erhöht werden. Diese Methode ist vor allem in Gebieten nützlich, in denen es extrem stark, aber auch extrem schwach bevölkerte Gebiete gibt (z.B. Russland). Der Nachteil hierbei ist allerdings, dass dichter besiedelte Gebiete unter Umständen zu wenig getestet werden. Deshalb ist das Einbeziehen der Testfalldichte in den Tiles eine gute Möglichkeit diesen Nachteil auszugleichen. Für Tiles in denen mehr Testfälle möglich sind, werden demnach auch mehr erzeugt. Somit werden sämtliche Regionen abgedeckt, wobei in stärker besiedelten Gebieten mit höherer Testfalldichte mehr Fälle erzeugt werden.

Der Generator kann die Testfälle momentan für die komplette Karte, ein Land oder einen Bundesstaat erzeugen. Eine mögliche Erweiterung der geografischen Verteilung und der Dichteverteilung ist die Erzeugung der Testfälle in den Tiles komplett unabhängig von politischen Grenzen zu machen. Beispielsweise könnte ein Rechteck über eine Karte aufgespannt werden, in dessen Tiles die Testfälle erzeugt werden sollen. Damit würden politische Grenzen komplett ignoriert.

4.6 Mehrfach auftretende Straßen

Ein weiterer Untersuchungsaspekt sind Straßen, welche mehrfach in einer Stadt vorkommen. Das kann vor allem in größeren Städten auftreten. Die Geodienste sollten also in der Lage sein sämtliche mehrfach vorkommenden Straßen zu finden. Für diesen Aspekt können zwei Fälle unterschieden werden.

Einerseits kann es sein, dass Straßen in unterschiedlichen Stadtbezirken auftreten und unterschiedliche Postleitzahlen haben. Dabei kann beispielsweise eine der Straßen im nördlichsten Bereich der Stadt liegen, während die andere Straße im südlichsten Bereich zu finden ist. Es handelt sich also um verschiedene Objekte. Dieser Fall kann zum Beispiel auftreten, wenn mehrere Städte zusammengelegt, die Straßennamen aber beibehalten wurden. Oder aber die Straßen haben ihre Bezeichnung durch eine Besonderheit in der Umgebung erhalten (z.B. "Bahnhofstraße", "Parkstraße").

Andererseits sind Straßen betroffen, welche physikalisch zwar nur einmal existieren, aber durch mehrere Bezirke und Postleitzahlen laufen, da sie sehr lang ist. Somit existieren sie scheinbar mehrmals, da sie mit unterschiedlichen Postleitzahlen auffindbar sind. Das betrifft beispielsweise Bundesstraßen (welche jedoch abschnittsweise eine weitere Bezeichnung haben) oder innerstädtische Autobahnen. Aber auch normale Straßen können solche Eigenschaften besitzen und durch ihre Länge in mehreren Bezirke verlaufen (z.B. "Kurfürstendamm Berlin").

Für diesen Anwendungsfall soll der Testgenerator Testfälle für mehrfach vorhandene Straßen in einer Stadt erzeugen. Vor allem der Fall physikalisch unterschiedlicher Straßen ist dabei interessant, da der Geodienst alle diese finden sollte. Für die Erkennung dieser Straßen durch die Geodienste spielt unter Umständen das Eingabeformat eine wichtige Rolle, weshalb hierbei mehrere Formate untersucht werden sollten. Die Eingabe für den Testfall kann beispielsweise nur aus Straßenname und Stadtname bestehen. Eventuell ist jedoch auch der

Stadtbezirk oder die Postleitzahl für eine eindeutige Zuordnung wichtig und muss an die Eingabe angehängt werden.

4.7 Erzeugte Eingaben in unterschiedlichen Diensten

Wurde die passende Erzeugungsstrategie ausgewählt und damit die Testfälle erzeugt, so können diese in den Geodiensten getestet werden. Vor der Ausführung der Tests gibt es noch einige Dinge zu beachten.

Die Testfälle wurden mit einer bestimmten Datenbank (in dieser Arbeit eine NDS-Datenbank) erzeugt. Wenn diese Datenbank in einem Dienst verwendet wird, der getestet werden soll, dann ist sichergestellt, dass die gesuchten Adressen auch wirklich vorhanden sind. Wenn der Dienst jedoch eine andere Datenbank verwendet, ist das nicht sichergestellt. Somit stellt sich die Frage, wie festgestellt werden kann, ob eine Adresse zwar in der Datenbank vorhanden ist aber nicht von der Geocoding-Software gefunden werden kann oder ob sie nicht in der Datenbank vorhanden ist.

Wenn die Adresse nicht in der Datenbank eingetragen ist, dann kann sie logischerweise nicht gefunden werden. Die beste Möglichkeit zu überprüfen, ob der Eintrag in der Datenbank fehlt, ist die Adresse so komplett wie möglich im Dienst zu testen. Dabei sollten sämtliche vorhandene Token im richtigen Adressformat separiert durch Trennzeichen wie Kommata in der Eingabe vorhanden sein. Zusätzlich sollten keine Abkürzungen verwendet werden und die Token unverändert sein (z.B. keine Bindestriche mit Leerzeichen ersetzen). Wenn der Dienst diese Eingabe nicht findet, dann ist davon auszugehen, dass die verwendete Datenbank die Adresse nicht enthält. Deshalb ist es unter Umständen sinnvoll, dass jeder Testfall neben einer eventuell kürzeren und veränderten Eingabe, die einem Benutzer Use Case entspricht, zusätzlich die komplette oder zumindest eine sehr detaillierte Eingabe enthält, um bei einem Test, der kein Resultat liefert zu überprüfen, ob die Adresse womöglich nicht in der Datenbank vorhanden ist.

Wird die Adresse nicht gefunden, obwohl sie in der Datenbank vorhanden ist, kann das unterschiedliche Gründe haben. Einerseits kann es sein, dass die Eingabe nicht detailliert genug beziehungsweise nicht eindeutig war. Dann wäre es unter Umständen nötig, dass mehr Token für die Eingabe verwendet werden sollten. Jedoch wäre die kurze Eingabe ein Anwendungsfall für einen Benutzer, der erwartet, dass er auch bei einer mitunter nicht vollständigen Eingabe ein Resultat erhält, welches im besten Fall sein gesuchtes Ergebnis ist. Andererseits kann es auch sein, dass die Eingabe gekürzte oder veränderte Token enthält. Dies entspricht ebenfalls einem Anwendungsfall, da ein Benutzer beispielsweise Straßennamen abkürzt (z.B. "Takustr" anstatt "Takustraße").

Es sollte jedoch davon ausgegangen werden können, dass die Entwickler solche Anwendungsfälle beachtet haben und somit trotzdem Ergebnisse finden können, obwohl Eingaben nur aus den wesentlichsten Adresselementen bestehen oder aber die Elemente verändert wurden. Es muss also letztendlich abgewägt werden, ob die Integration zwischen Software und Datenbank fehlgeschlagen ist oder die Daten nicht in der Datenbank vorhanden sind.

KAPITEL 5

Fallstudie

In diesem Kapitel werden die vorgestellten Methoden verwendet um unterschiedliche Geodienste in einer Fallstudie zu testen. Die Testfälle wurden mit dem in der Arbeit umgesetzten Testgenerator erzeugt. Nach den Tests werden die Ergebnisse präsentiert und bewertet.

5.1 Durchführung der Tests in Geodiensten

Für die Testdurchführung werden anfangs mit dem Testgenerator Testfälle für unterschiedliche Szenarien erstellt. Zu den Testfalleingaben erzeugt der Generator die entsprechenden Soll-Daten. Für die Durchführung der Tests werden die Online-Geodienste von Google Maps, Here, Bing Maps und OpenStreetMap verwendet, welche bereits in Abschnitt 3.1.2 vorgestellt wurden. Die Dienste werden über eine REST-API angesprochen. Dabei wird die Testfalleingabe übergeben, welche für das Geocoding benutzt werden soll. Ein weiterer wichtiger Parameter ist ein API-Schlüssel, welcher jedem Benutzer zugewiesen wird und die Zugriffe auf die API loggt. Optional kann die Rückgabesprache für die Adresse angegeben werden, damit der Name des Landes in der gewünschten Sprache zurück gegeben wird (z.B. "Deutschland" anstatt "Germany").

Die Rückgabe der Dienste ist eine Liste, welche bei mehrdeutigen Eingaben mehrere Ergebnisse enthalten kann. Diese beinhalten neben den Koordinaten des gewünschten Ergebnisses die Adresse mit den Adresselementen, wobei nicht bei allen Diensten alle Elemente enthalten sein müssen. Da die Rückgabeformate der Dienste unterschiedlich sind, werden diese mit einem Formatter-Tool in ein einheitliches Format gebracht, um später die Auswertung zu vereinfachen. Hierbei kann es auftreten, dass einige Ergebnisstrings nicht die gleiche Syntax wie in den Soll-Daten haben. Semantisch ist das Ergebnis jedoch identisch. Aus diesem Grund muss an dieser Stelle eventuell eine manuelle Anpassung vorgenommen werden. So kann es beispielsweise auftreten, dass die Rückgabe "USA", "Vereinigte Staate" oder "Vereinigte Staaten von Amerika" enthält. In den Soll-Daten existiert jedoch nur einer dieser Strings. Das ist allerdings nur durch eine manuelle Inspektion möglich, weshalb in diesem und in ähnlichen Fällen eine Anpassung vorgenommen wird. Abschließend können die Soll-Daten mit den Resultaten im einheitlichen Format in einem Abgleich-Tool verglichen werden und

5. Fallstudie

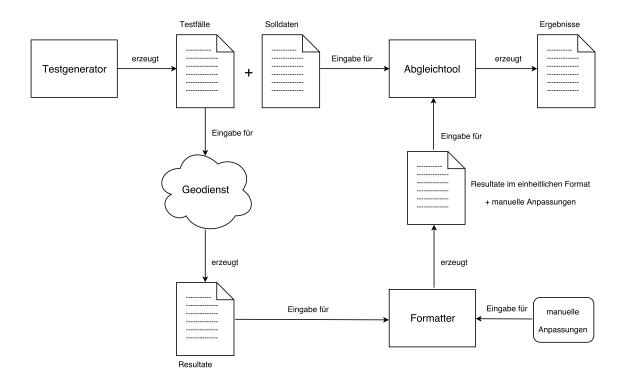


Abbildung 5.1.: Prozess: Testdurchführung und Testauswertung

die Ergebnisse erstellt werden. Der Vorgang ist in Abbildung 5.1 dargestellt.

Die Durchführung der Tests in den Geodiensten soll nach Möglichkeit unterschiedliche Aspekte und Strategien behandeln. Es soll dabei nach Repräsentanten aus Äquivalenzklassen gesucht werden, um die Erkennung der unterschiedlichen Strukturen der Elemente zu testen. Des Weiteren sollen die verschiedenen Verteilungen getestet werden, um zu klären, ob eine Verteilung nach politischen oder geografischen Aspekten sinnvoll ist. Zusätzlich soll bewertet werden, ob die geografische Verteilung mit Einbeziehung der Testfalldichte sinnvoll ist oder nicht. Ein weiterer Aspekt ist die Untersuchung von Straßen, deren Name mehrfach in einer Stadt vorkommt. Zu guter Letzt werden noch verschiedene Eingabealphabete getestet. Bei einigen Szenarien werden zusätzlich unterschiedliche Eingabeformate verwendet, um zu prüfen, ob genauer spezifizierte Adressen besser gefunden werden können. Die einzelnen Szenarien werden im nächsten Abschnitt vorgestellt.

5.2 Testszenarien

Die Testszenarien stellen sich aus der Wahl einer Testerzeugungsstrategie und der Wahl einer Testregion zusammen. Es ist nicht sinnvoll jede Testerzeugungsstrategie in jeder Region zu nutzen. Einige Strategien ergeben eventuell nur in bestimmten Regionen Sinn.

Die Testerzeugung mit Hilfe der Äquivalenzklassen ist sinnvoll wenn überprüft werden soll, ob die Geocoding-Software sämtliche syntaktischen Strukturen und Formate erkennt. Da diese in den unterschiedlichen Ländern verschieden sind, ist es beispielsweise sinnvoll,

5.2. Testszenarien 51

die größte Stadt in einem Land (in vielen Fällen die Hauptstadt) zu wählen und dort die Äquivalenzklassen der Struktur der Testfälle zu bilden und zu testen. Somit ist eine hohe Abdeckung der Erkennung für Formate und Strukturen für das ganze Land sehr wahrscheinlich.

Die Wahl der geografischen Erzeugungsstrategie ist beispielsweise sinnvoll, wenn eine Region über einerseits stark bevölkerte Gebiete verfügt und andererseits auch über schwach bevölkerte Gebiete. Ist die Bevölkerungsdichte und somit die Testfalldichte in der gesamten Region relativ ähnlich, ergibt die Untersuchung dieser Verteilung hier recht wenig Sinn. Die Strategie ist jedoch auch sinnvoll, wenn das zu untersuchende Gebiet eine große geografische Ausdehnung hat (beispielsweise in Form von Inseln). Die geografische Erzeugungsstrategie kann hier dann beispielsweise mit der politisch zufälligen Strategie verglichen werden. Für die Einbeziehung der Testfalldichte eignet sich auch eine Region, in der die Bevölkerungsdichte und somit auch die Testfalldichte sehr unterschiedlich ist.

Sollen mehrfach auftretende Straßennahmen in Städten getestet werden, sind ebenfalls große Städte, wie die Hauptstadt, in einem Land geeignet. Zu kleine Orte haben mit hoher Wahrscheinlichkeit keine mehrfach vergebenen Straßennamen. Zudem ist die Wahrscheinlichkeit hoch, dass in Hauptstädten die Straßennamen nicht nur zweimal vorkommen, sondern noch öfter. Somit kann getestet werden, wie gute die Geodienste diese Adressen finden.

Ein weiterer Aspekt, den es zu testen gilt, ist die Erkennung verschiedener Eingabesprachen. Dafür ist es sinnvoll in Ländern zu suchen, welche nicht das lateinische Alphabet verwenden, sondern beispielsweise Alphabete mit vielen diakritischen Zeichen, das kyrillische Alphabet oder das griechische Alphabet. Hierfür kann jedoch eine beliebige Strategie zur Testerzeugung verwendet werden.

Für die Durchführung der Tests wurden folgende Versionen der Geodienste verwendet:

Bing Maps: Geocoding API v1Google Maps: Geocoding API v3

• Here: Geocoding API 6.2.99

• OSM: Nominatim Geocoding API Version 1

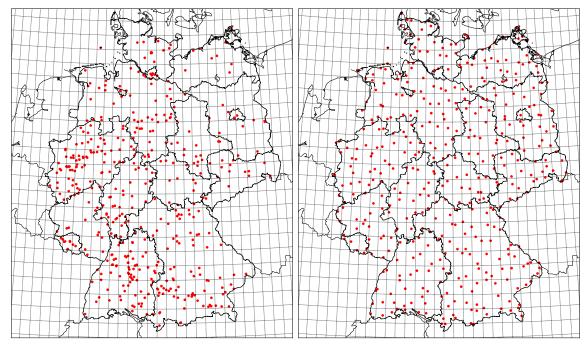
Hinweis: In den folgenden Abschnitten wird die Bezeichnung Staat beziehungsweise Bundesstaat als größte Verwaltungseinheit auf nationale Ebene betrachtet. Die korrekte Bezeichnung kann in den getesteten Regionen jedoch anders lauten (z.B. Russland: Förderationskreis).

5.2.1 Szenario 1: Zufällige und geografische Verteilung: Deutschland

Im ersten Szenario werden die zufällige und die geografisch gleichverteilte Testerzeugungsstrategie aus Abschnitt 4.5 in Deutschland getestet. Die geografische Verteilung ermittelt für Deutschland 440 Tiles und erzeugt in jedem dieser einen Testfall. Für die zufällige Verteilung werden daher ebenfalls 440 Testfälle in Deutschland erzeugt. Die Visualisierungen hierfür sind in Abbildung 5.2 dargestellt.

Die Rückgabesprache der Resultate wird auf deutsch gestellt. In diesem Szenario haben die Testfälle folgende Strukturen:

5. Fallstudie



(a) Zufällige Verteilung von 440 Testfällen in Deutsch-(b) Geografisch gleichmäßige Verteilung in den 440 land Tiles (Level 9) in Deutschland

Abbildung 5.2.: Zufällige und geografisch gleichmäßgige Verteilung in Deutschland

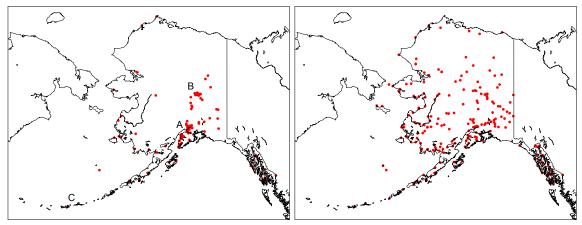
- Straße Hausnummer Stadt
- Straße Hausnummer Stadt Postleitzahl

5.2.2 Szenario 2: Zufällige und geografische Verteilung: Alaska

Das zweite Szenario behandelt die zufällige und geografische Verteilung aus Abschnitt 4.5 im US-amerikanischen Bundesstaat Alaska. Hintergrund dieses Szenarios ist die dünne Besiedlung Alaskas. Zudem gehören mehrere Inseln zu Alaska (z.B. die Aleuten), auf denen es meist nur sehr kleine Orte oder Forschungsstationen gibt. Jedoch ist nicht sichergestellt, dass diese auch von den Geodiensten gefunden werden. Der Großteil der Bevölkerung lebt in der Stadt Anchorage. Inklusive Metropolregion befinden sich hier ungefähr die Hälfte der Einwohner des Bundesstaates (circa 360.000). Die nächst größere Stadt hat hingegen nur circa 31.000 Einwohner. Damit werden bei einer zufälligen Verteilung ein Großteil der Testfälle in der Region um die Stadt Anchorage erzeugt, während die dünn besiedelten Gebiete und kleineren Städte vernachlässigt werden. Hier hilft also die geografische Verteilung auch die anderen Gebiete zu testen.

Abbildung 5.3 zeigt den Vergleich der Verteilungen in Alaska. Hier gibt es 246 Level-9-Tiles, in denen Testfälle vorhanden sind. In den restlichen Tiles des Landes gibt es keine Testfälle. In der zufälligen Verteilung liegen vor allem viele Testfälle nahe der Städte Anchorage (A) oder Fairbanks (B), beziehungsweise in deren Metropolregion. Die geografisch gleichverteilte Strategie erzeugt hingegen weiter gestreute Testfälle im Bundesstaat. Im Norden, Nordwesten

5.2. Testszenarien 53



(a) Zufällige Verteilung von 246 Testfällen in Alaska (b) Geografisch gleichmäßige Verteilung in den 246 Tiles (Level 9) in Alaska

Abbildung 5.3.: Zufällige und geografisch gleichmäßige Verteilung in Alaska

und im Zentrum des Staates sind wesentlich mehr Fälle zu finden. Auch auf den Aleuten (C - Inselkette im Südwesten) sind mehr Tests zu finden. Auffällig ist jedoch, dass auch in der geografisch gleichverteilten Strategie viele Testfälle nahe beieinander liegen. Das liegt an der in Abschnitt 2.6 beschriebenen Abweitung, welche nahe der Pole geringer wird. Damit verringert sich auch die Größe der Tiles wesentlich und die Testfälle liegen näher zusammen. Zusammenfassend kann die geografische Verteilung gegenüber der zufälligen Verteilung punkten, da auch die sehr schwach besiedelten Gebiete getestet werden können.

Die Rückgabesprache der Resultate wird auf deutsch gestellt. In diesem Szenario haben die Testfälle folgende Strukturen:

- Hausnummer Straße Stadt
- Hausnummer Straße Stadt Postleitzahl
- Hausnummer Straße Stadt Bundesstaat
- Hausnummer Straße Stadt Bundesstaat Postleitzahl

5.2.3 Szenario 3: Verteilung mit Einbeziehen der Testfalldichte: Alaska

Wie die zufällige Verteilung aus dem zweiten Szenario gezeigt hat, sind weite Teile Alaskas nur sehr dünn besiedelt. Durch die geografisch gleichmäßige Verteilung lassen sich diese Regionen jedoch testen. Allerdings werden unter Umständen einige Gebiete dabei zu wenig getestet (z.B. Regionen um die Städte Anchorage und Fairbanks), da in jedem Tile nur eine festgelegte Anzahl an Tests erzeugt wird (in Szenario 2: ein Testfall pro Tile). Somit kann es passieren, dass Metropolregionen eventuell nicht ausreichen getestet werden. Wenn beispielsweise ein dicht besiedeltes Gebiet in genau einem Tile liegt, wird hierfür nur ein Testfall erzeugt. Durch die Betrachtung der Testfalldichte in Kombination mit einer geografisch gleichmäßigen Verteilung kann die Testfallanzahl in diesen Gebieten gesteigert werden ohne, dass dünn besiedelte Regionen vernachlässigt werden. Für das dritte Szenario wird dieses Verfahren

5. Fallstudie

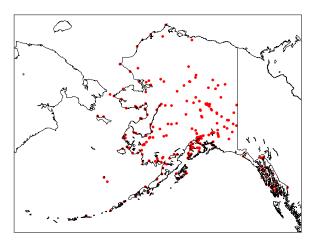


Abbildung 5.4.: Dichteverteilung in Alaska (288 Testfälle)

verwendet, wobei die maximale Testfallanzahl in einem Tile auf 10 Testfälle begrenzt wird.

In Abbildung 5.4 wird das Verfahren für Alaska visualisiert. Hier wird deutlich, dass ebenso wie in Abbildung 5.3b eine Verteilung über den kompletten Bundesstaat geschieht, jedoch die Metropolregionen stärker getestet werden. Die Anzahl der Testfälle ist gegenüber der geografischen Verteilung von 246 auf 288 gestiegen.

Die Rückgabesprache der Resultate wird auf deutsch gestellt. In diesem Szenario haben die Testfälle folgende Strukturen:

- Hausnummer Straße Stadt
- Hausnummer Straße Stadt Postleitzahl
- Hausnummer Straße Stadt Bundesstaat
- Hausnummer Straße Stadt Bundesstaat Postleitzahl

5.2.4 Szenario 4: Äquivalenzklassentest: Berlin

Dieses Szenario testet die Bildung der Äquivalenzklassen mit Hilfe des K-Means-Algorithmus und dem in Abschnitt 4.4 vorgestellten Ansatz. Für die Bildung der Klassen werden sämtliche Straßen der Stadt Berlin erzeugt. Anschließend wird der Algorithmus angewendet und die Testdaten werden anhand ihrer Struktur geclustert. Nun werden Repräsentanten gewählt und daraus die Testfälle erzeugt. Der Generator erzeugt mit Hilfe der Äquivalenzklassen 110 Klassen. Somit werden deren Repräsentanten getestet.

Die Rückgabesprache der Resultate wird auf deutsch gestellt. In diesem Szenario haben die Testfälle folgende Strukturen:

- Straße Hausnummer Stadt
- Straße Hausnummer Stadt Postleitzahl

5.2. Testszenarien 55

5.2.5 Szenario 5: Mehrfach vorkommende Straßennamen: Berlin

Das fünfte Szenario behandelt das mehrfache Vorkommen von Straßennamen in Berlin, wie es bereits in Abschnitt 4.6 vorgestellt wurde. Die getesteten Straßen kommen also unter gleichem Namen in unterschiedlichen Bezirken und Postleitzahlgebieten vor (z.B. "Bahnhofstraße Berlin"), stellen aber keine physikalische Einheit dar (z.B. im Gegensatz zu "Kurfürstendamm Berlin"). Es wird dabei getestet, ob die Geodienste sämtliche Straßen vorschlagen, auch wenn lediglich deren Name und der Stadtname eingegeben wird. In einem weiteren Testlauf werden die Eingaben durch Anfügen der Postleitzahl weiter detailliert. Somit kann sichergestellt werden, dass die erwartete Adresse zurückgegeben wird, wenn der Geodienst sie findet.

Der Generator findet in Berlin 1182 Straßennamen, welche mehrmals vorkommen. Für das Szenario wurde jede zehnte Straße für die Testfälle verwendet, sodass am Ende 119 mehrfach vorkommende Straßen übrig blieben. Am Ende entstehen somit 252 Testfälle.

Die Rückgabesprache der Resultate wird auf deutsch gestellt. In diesem Szenario haben die Testfälle folgende Strukturen:

- Straße Hausnummer Stadt
- Straße Hausnummer Stadt Postleitzahl

5.2.6 Szenario 6: Sprachen: diakritische Zeichen: Prag

Szenario sechs testet die Eingabe von Adressen, welche viele diakritische Zeichen beinhalten. Für diesen Zweck wird die tschechische Sprache getestet. Dafür werden die Äquivalenzklassen in Prag mit dem K-Means-Ansatz erstellt. Durch die Auswahl der Repräsentanten aus den Clustern entstanden 158 Testfälle.

Die Rückgabesprache der Resultate wird auf tschechisch gestellt. In diesem Szenario haben die Testfälle folgende Struktur:

• Straße Hausnummer Stadt

5.2.7 Szenario 7: Sprachen: kyrillisch: Moskau

Das siebte Szenario soll die kyrillische Sprache testen. Dafür werden Testfälle in Moskau erzeugt. Auch hier werden die Äquivalenzklassen für die Hauptstadt gebildet. Durch die Auswahl der Repräsentanten ergeben sich 62 Testfälle.

Die Rückgabesprache der Resultate wird auf russisch gestellt. In diesem Szenario haben die Testfälle folgende Struktur:

• Straße Hausnummer Stadt

5.2.8 Szenario 8: Sprachen: griechisch: Athen

Im achten Szenario wird die griechische Sprache getestet. Hierfür werden die Testfälle in Griechenlands Hauptstadt Athen erzeugt. Auch hierbei werden die Äquivalenzklassen gebildet und somit 43 Testfälle erzeugt.

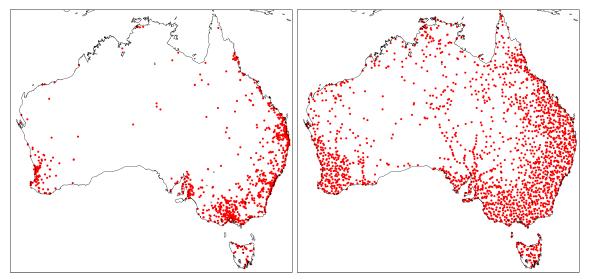
5. Fallstudie

Die Rückgabesprache der Resultate wird auf griechisch gestellt. In diesem Szenario haben die Testfälle folgende Struktur:

• Straße Hausnummer Stadt

5.2.9 Szenario 9: Zufällige und geografische Verteilung: Australien

Das neunte Szenario behandelt die zufällige und geografische Verteilung in Australien. In Abbildung 5.5 sind die beiden Verteilungen dargestellt.



(a) Zufällige Verteilung von 2038 Testfällen in Austra-(b) Geografisch gleichmäßige Verteilung in den 2038 lien Tiles (Level 9) in Australien

Abbildung 5.5.: Zufällige und geografisch gleichmäßige Verteilung in Australien

Da das Land an der West-, Ost- und Südostküste am stärksten besiedelt ist, werden bei einer zufälligen Testfallerzeugung im Land hier die meisten Fälle erzeugt. Regionen im australischen Outback hingegen werden durch die geringe Besiedlung vernachlässigt. Durch die geografische Verteilung werden hier jedoch ebenfalls Testfälle erzeugt, falls welche vorhanden sind. Hier ist gut zu sehen, dass die zufällige Verteilung fast nur Testfälle an der Ost-, Südost- und Westküste erzeugt, das Outback jedoch nahezu komplett vernachlässigt. Die geografische Verteilung hingegen erzeugt wesentlich mehr Fälle in den gering besiedelten Regionen.

Die geografische Verteilung erzeugt für Australien 2038 Testfälle, sodass der gleiche Wert für die Zufällige Verteilung verwendet wird. Die Rückgabesprache der Resultate wird auf deutsch gestellt. In diesem Szenario haben die Testfälle folgende Strukturen:

- Hausnummer Straße Stadt
- Hausnummer Straße Stadt Postleitzahl
- Hausnummer Straße Stadt Postleitzahl Land

5.3 Testauswertung

Für die Auswertung der Tests werden die API-Ergebnisse der Geodienste in ein einheitliches Format übertragen. Hierbei werden zudem für den Abgleich uninteressante Informationen entfernt (z.B. Geopositionen und Metainformationen). Da die einzelnen Resultatelemente bei den Diensten teilweise syntaktisch unterschiedlich, aber semantisch identisch sind, müssen diese unter Umständen angepasst werden. Einige Dienste liefern mehrere Strings in unterschiedlicher Syntax zurück, sodass das passende Format gewählt werden kann (beispielsweise ISO-Code und komplette Schreibweise). Somit muss ein einheitliches Format verwendet werden, um später beim Abgleich mit den Soll-Daten keine Fehler zu produzieren, die eigentlich keine sind.

Wurden die Daten in ein einheitliches Format übertragen, so kann der Abgleich der Testdaten mit den Soll-Daten durchgeführt werden. Dabei können drei Fälle auftreten. Ersten kann es sein, dass eine Testeingabe kein Resultat produziert. Dann wurde kein Ergebnis gefunden und der Test wird als fehlgeschlagen markiert. Im zweiten Fall wurde genau ein Resultat gefunden. Hier werden sämtliche Elemente mit den Soll-Daten abgeglichen. Stimmen dabei nicht alle Elemente überein, ist der Test fehlgeschlagen. Der dritte Fall beschreibt die Rückgabe von mehreren Resultaten. Jedes Resultat wird einzeln mit den Soll-Daten verglichen. Stimmen bei einem Ergebnis alle Elemente überein, war der Test erfolgreich. Stimmen sie jedoch bei keinem Ergebnis überein, ist der Test fehlgeschlagen.

Beim Auswerten werden sämtliche Elemente der Adresse abgeglichen. Dazu gehören Land, Bundesstaat, Stadt, Stadtbezirk, Postleitzahl, Straße und Hausnummer. Da bei einigen Diensten jedoch teilweise der Stadtbezirk als Stadt zurückgegeben wird, werden bei der Überprüfung Stadt und Stadtbezirk gleich gestellt. Somit kann es hier nicht zu Fehlern kommen, welche eigentlich keine sind.

5.4 Testergebnisse

Der folgende Abschnitt enthält die Testergebnisse sowie die Auswertung für die Szenarien aus Abschnitt 5.2. Neben den im Abschnitt dargestellten Resultaten, befinden sich noch weitere im Anhang B. Dort sind unter anderem die Gründe aufgelistet, weshalb die Tests fehlgeschlagen sind.

In den Abbildungen wurden folgende Abkürzungen für die Testkonfigurationen verwendet:

- ZV Zufällige Verteilung
- GV Gleichmäßige Verteilung
- DV Dichteverteilung
- H Hausnummer
- S Straße
- O Ort
- P Postleitzahl
- B Bundesstaat

5. Fallstudie

• L - Land

Beispiel: Die Abkürzung GV-HSOBP steht für Gleichmäßige Verteilung – Hausnummer Straße Ort Bundesstaat Postleitzahl.

5.4.1 Szenario 1: Zufällige und geografische Verteilung: Deutschland

Die Tabellen 5.1 und 5.2 zeigen die Ergebnisse des ersten Szenarios. In Abbildung 5.6 sind die Ergebnisse zudem in einem Sterndiagramm dargestellt. Die Tests wurden mit den Eingabeformaten Straße Hausnummer Ort (SHO) und Straße Hausnummer Ort Postleitzahl (SHOP) durchgeführt. Im Anhang B.1 befinden sich detaillierte Informationen zu den Resultaten.

	Zufälli	Zufällige Verteilung (ZV)			Gleichverteilung (GV)			
GIS	Erfolg	Fehler	Fehlerrate		Erfolg	Fehler	Fehlerrate	
Bing Maps	409	31	$7{,}05\%$		400	40	$9{,}09\%$	
Google Maps	375	65	$14{,}77\%$		364	76	$17{,}27\%$	
Here	420	20	$4{,}55\%$		422	18	$4{,}09\%$	
OSM	171	269	$61{,}14\%$		105	335	$76{,}14\%$	

Tabelle 5.1.: Szenario 1: Zufällige und gleichmäßige Verteilung: Straße Hausnr. Ort (SHO)

	Zufälli	Zufällige Verteilung (ZV)			Gleichverteilung (GV)		
GIS	Erfolg	Fehler	Fehlerrate	Erfolg	Fehler	Fehlerrate	
Bing Maps	410	30	6,82 %	401	39	8,86 %	
Google Maps	337	63	$14{,}32\%$	365	75	$17{,}05\%$	
Here	422	18	$4{,}09\%$	422	18	$4{,}09\%$	
OSM	169	271	$61{,}59\%$	105	335	$76{,}14\%$	

Tabelle 5.2.: Szenario 1: Zufällige und gleichmäßige Verteilung: Straße Hausnr. Ort Postleitzahl (SHOP)

Der Dienst Here findet den Großteil der Adressen. Nachfolgend kommt Bing Maps, anschließend Google Maps. Bei OSM schlagen die meisten Tests fehl, was auf fehlende Daten zurückgeführt werden könnte.

Der Unterschied zwischen einer gleichmäßigen und einer zufälligen Verteilung ist gering. Für Bing Maps und Google Maps ist die Zufallsverteilung leicht besser, für Here hingegen ist die Gleichverteilung etwas besser. Bei OSM hingegen schlagen deutlich mehr Tests bei der Gleichverteilung fehl. Das könnte daran liegen, dass hierbei geringer besiedelte Regionen stärker mit einbezogen werden als bei der zufälligen Verteilung. Diese Gebiete sind scheinbar teilweise nicht in der OSM Datenbank vorhanden.

Wird für das Eingabeformat die Postleitzahl hinzugenommen, so fällt der Unterschied zum

5.4. Testergebnisse 59



Abbildung 5.6.: Szenario 1: Übersicht - Bestandene Tests (100 %abzüglich Fehlerrate)

5. Fallstudie

vorherigen Eingabeformat sehr gering aus. Das Element scheint in Deutschland aufgrund der eindeutigen Ortsbenennungen relativ belanglos zu sein. Bei der Betrachtung der Online Geodienste schneidet Here am besten ab, gefolgt von Bing Maps, Google Maps und schließlich OSM.

Die Netzdiagramme in Abbildung 5.6 belegen die Aussagen. Die Graphen bei Bing Maps, Google Maps und Here haben eine nahezu quadratische Form, da die Ergebnisse der einzelnen Dienste sehr eng beieinander liegen. Bei OSM hingegen ist der Graph eher trapezförmig, da die Werte für die Gleichverteilung schlechter ausfallen als für die zufällige Verteilung.

5.4.2 Szenario 2: Zufällige und geografische Verteilung: Alaska

Die Tabellen 5.3 bis 5.6, sowie Abbildung 5.7 stellen die Ergebnisse für das zweite Szenario dar. Hierbei wurden die Tests mit den Eingabeformaten Hausnummer Straße Ort (HSO), Hausnummer Straße Ort Postleitzahl (HSOP), Hausnummer Straße Ort Bundesstaat (HSOB) und Hausnummer Straße Ort Bundesstaat Postleitzahl (HSOBP) durchgeführt. Es handelt sich um das lokale Adressformat. Der Anhang B.2 enthält weitere Daten zur Auswertung.

	Zufälli	Zufällige Verteilung (ZV)			Gleichverteilung (GV)			
GIS	Erfolg	Fehler	Fehlerrate		Erfolg	Fehler	Fehlerrate	
Bing Maps	94	152	$61{,}79\%$		14	232	$94{,}31\%$	
Google Maps	140	106	$43{,}09\%$		90	156	$63{,}41\%$	
Here	144	102	$41{,}46\%$		98	148	$60{,}16\%$	
OSM	74	172	$69{,}92\%$		25	221	$89{,}84\%$	

Tabelle 5.3.: Szenario 2: Zufällige und gleichmäßige Verteilung: Hausnr. Straße Ort (HSO)

	Zufällige Verteilung (ZV)			Gleichverteilung (GV)		
GIS	Erfolg	Fehler	Fehlerrate	Erfolg	Fehler	Fehlerrate
Bing Maps	130	116	$47,\!15\%$	88	158	$64,\!23\%$
Google Maps	144	102	$41{,}46\%$	95	151	$61{,}38\%$
Here	159	87	$35{,}37\%$	103	143	$58{,}13\%$
OSM	67	179	$72{,}76\%$	24	222	$90{,}24\%$

Tabelle 5.4.: Szenario 2: Zufällige und gleichmäßige Verteilung: Hausnr. Straße Ort Postleitzahl (HSOP)

Bei der zufälligen Verteilung sind die Unterschiede zwischen den Ergebnissen der verschiedenen Eingabeformate relativ gering. Während die Fehlerquote bei Bing Maps für das Format Hausnummer Straße Ort (HSO) bei circa 62 % liegt, liegt sie bei den anderen Formaten zwischen 45 % und 47 %. Google Maps liegt bei allen Formaten zwischen 43 % und 40 %. Der Dienst Here liegt bei circa 41 % wenn die Postleitzahl nicht Teil der Eingabe ist, ansonsten

	Zufällige Verteilung (ZV)			Gleichverteilung (GV)		
GIS	Erfolg	Fehler	Fehlerrate	Erfolg	Fehler	Fehlerrate
Bing Maps	134	112	$45{,}53\%$	94	154	$61{,}79\%$
Google Maps	139	107	$43{,}50\%$	92	154	$62{,}60\%$
Here	144	102	$41{,}46\%$	98	148	$60{,}16\%$
OSM	74	172	$69{,}92\%$	24	222	$90{,}24\%$

Tabelle 5.5.: Szenario 2: Zufällige und gleichmäßige Verteilung: Hausnr. Straße Ort Bundesstaat (HSOB)

	Zufällige Verteilung (ZV)			Gleichverteilung (GV)		
GIS	Erfolg	Fehler	Fehlerrate	Erfolg	Fehler	Fehlerrate
Bing Maps	135	111	$45{,}12\%$	96	150	$60{,}98\%$
Google Maps	146	100	$40{,}65\%$	95	151	$61{,}38\%$
Here	161	85	$34{,}55\%$	104	142	$57{,}72\%$
OSM	66	180	$73{,}17\%$	23	223	$90,\!65\%$

Tabelle 5.6.: Szenario 2: Zufällige und gleichmäßige Verteilung: Hausnr. Straße Ort Bundesstaat Postleitzahl (HSOBP)

bei circa $35\,\%$. Die Fehlerquote von OSM ist ohne Eingabe der Postleitzahl bei Testfällen bei circa $69\,\%$ während sie mit Eingabe der Postleitzahl auf circa $73\,\%$ steigt.

Bei der Betrachtung der Unterschiede der Ergebnisse zwischen den Eingabeformaten für das Szenario für gleichverteilte Testfälle fällt auf, dass außer bei Bing Maps kaum Abweichungen auftreten. OSM hat bei sämtlichen Eingabeformaten lediglich bei circa 10% der Testfälle das erwartete Ergebnis gefunden. Bing Maps, Google Maps und Here hingegen haben alle eine ähnliche Fehlerwahrscheinlichkeit von ungefähr 60% (abgesehen von Bing Maps beim Eingabeformat $Hausnummer\ Straße\ Stadt\ (HSO)$). Genauere Eingaben (beispielsweise mit Postleitzahl oder Bundesstaat) führen lediglich zu einer geringen Verbesserung.

Der Vergleich zwischen der zufälligen und der gleichmäßigen Verteilung zeigt eindeutig, dass die geografisch gleichmäßige Verteilung schlechter abschneidet. Die Tabellen in Anhang B.2 zeigen die Fehlergründe der einzelnen Tests.

Bei OSM werden für einen Großteil der Ergebnisse keine Resultate geliefert. Das liegt wahrscheinlich daran, dass die Daten sich nicht in der OSM Datenbank befinden. Bei der zufälligen Verteilung schlagen aber auch einige Tests beim Abgleichen mit den Soll-Daten fehl. Dies liegt vor allem an fehlenden Hausnummern, da diese wahrscheinlich ebenfalls in der Datenbank fehlen. Teilweise tragen aber auch Postleitzahlen dazu bei, dass der Abgleich fehlschlägt. Wenn diese mit zur Eingabe des Testfalls gehören, wird oft kein Resultat geliefert. Sind sie hingegen nicht Teil der Eingabe, so schlägt der Soll-Daten-Abgleich wegen nicht von OSM gelieferter Postleitzahlen fehl. Bei der gleichmäßigen Verteilung hingegen schlägt der Abgleich bei OSM oft wegen Postleitzahl, Straße, Hausnummer oder mehreren dieser Elementen fehl. Wird jedoch wieder die Postleitzahl zur Eingabe mit hinzugezogen, so wird

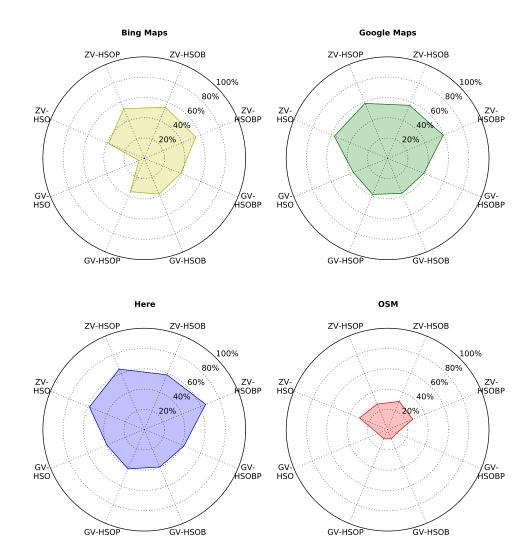


Abbildung 5.7.: Szenario 2: Übersicht - Bestandene Tests (100 % abzüglich Fehlerrate)

5.4. Testergebnisse 63

öfter kein Resultat geliefert als bei Eingaben ohne Postleitzahl.

Bei den Ergebnissen von Google Maps, Bing Maps und Here existieren bei dem ungenauesten Eingabeformat teilweise ebenfalls keine Ergebnisse, was sich jedoch reduziert, wenn Postleitzahl, Bundesstaat oder beides hinzugefügt wird. Bei der zufälligen Verteilung sind vor allem Postleitzahl, Straße, Hausnummer oder mehrere Elemente für fehlgeschlagene Abgleiche mit den Soll-Daten verantwortlich. Vor allem bei Bing Maps sind oft mehrere Elemente nicht korrekt. Bei der gleichmäßigen Verteilung ist zudem teilweise das Stadtelement nicht korrekt. Insgesamt erzielt Here die besten Ergebnisse bei den Online-Geodiensten, gefolgt von Google Maps, Bing Maps und abschließend OSM, was auch im Netzdiagramm 5.7 deutlich wird.

5.4.3 Szenario 3: Verteilung mit Einbeziehen der Testfalldichte: Alaska

In den Tabellen 5.7 bis 5.10 sind die Ergebnisse für das dritte Szenario erfasst. Die Resultate wurden um die Fehlerraten der Zufallsverteilung (ZV) und der Gleichverteilung (GV) aus dem zweiten Szenario zum Vergleichen erweitert. Zusätzlich werden die Resultate in Abbildung 5.8 visualisiert. In diese, Szenario wurden ebenfalls die Eingabeformate Hausnummer Straße Ort (HSO), Hausnummer Straße Ort Postleitzahl (HSOP), Hausnummer Straße Ort Bundesstaat (HSOB) und Hausnummer Straße Ort Bundesstaat Postleitzahl (HSOBP) gewählt. Im Anhang B.3 befinden sich weitere Resultate.

GIS	Erfolg	Fehler	Fehlerrate DV	Fehlerrate ZV	Fehlerrate GV
Bing Maps	58	230	$79{,}86\%$	$61{,}79\%$	94,31%
Google Maps	137	151	$52{,}43\%$	$43{,}09\%$	$63{,}41\%$
Here	160	128	$44{,}44\%$	$41{,}46\%$	$60{,}16\%$
OSM	50	238	$82{,}64\%$	$69{,}92\%$	$89{,}84\%$

Tabelle 5.7.: Szenario 3: Dichteverteilung: Hausnr. Straße Ort (HSO)

GIS	Erfolg	Fehler	Fehlerrate DV	Fehlerrate ZV	Fehlerrate GV
Bing Maps	140	148	$51{,}39\%$	$47{,}15\%$	64,23%
Google Maps	138	150	$52{,}08\%$	$41{,}46\%$	$61{,}38\%$
Here	183	105	$36{,}46\%$	$35{,}37\%$	$58{,}13\%$
OSM	48	240	$83{,}33\%$	$72{,}76\%$	$90{,}24\%$

Tabelle 5.8.: Szenario 3: Dichteverteilung: Hausnr. Straße Ort Postleitzahl (HSOP)

Beim Vergleich der Ergebnisse mit den Resultaten des zweiten Szenarios wird deutlich, dass sich die Fehlerraten der Dichteverteilung zwischen den Raten der Zufallsverteilung und der Gleichverteilung befinden. Dieser Umstand resultiert aus der Dichteverteilung, welche eine Kombination der anderen beiden Verteilungen darstellt. Des Weiteren ist auffällig, dass die Verwendung unterschiedlicher Eingabeformate einen ähnlichen Effekt wie im zweiten

GIS	Erfolg	Fehler	Fehlerrate DV	Fehlerrate ZV	Fehlerrate GV
Bing Maps	139	149	$51{,}74\%$	$45{,}53\%$	$61{,}79\%$
Google Maps	135	153	$53{,}12\%$	$43{,}50\%$	$62{,}60\%$
Here	162	126	$43{,}75\%$	$41{,}46\%$	$60{,}16\%$
OSM	50	238	$82,\!64\%$	$69{,}92\%$	$90{,}24\%$

Tabelle 5.9.: Szenario 3: Dichteverteilung: Hausnr. Straße Ort Bundesstaat (HSOB)

GIS	Erfolg	Fehler	Fehlerrate DV	Fehlerrate ZV	Fehlerrate GV
Bing Maps	145	143	$49{,}65\%$	$45{,}12\%$	$60{,}98\%$
Google Maps	139	149	$51{,}74\%$	$40{,}65\%$	$61{,}38\%$
Here	183	105	$36{,}46\%$	$34{,}55\%$	$57{,}72\%$
OSM	48	240	$83{,}33\%$	$73{,}17\%$	$90,\!65\%$

Tabelle 5.10.: Szenario 3: Dichteverteilung: Hausnr. Straße Ort Bundesstaat Postleitzahl (HSOBP)

Szenario hat. OSM hat eine hohe Fehlerrate und zeigt nahezu keine Veränderung bei der Verwendung unterschiedlicher Eingabeformate, was auch in Abbildung 5.8 durch die nahezu quadratische Form ersichtlich wird. Bing Maps findet für das Format Hausnummer Straße Ort (HSO) lediglich ein Fünftel aller Resultate. Bei Hinzunahme der Postleitzahl, des Bundesstaates oder beider Elemente sind ungefähr die Hälfte aller Tests erfolgreich. Bei Google Maps treten ähnlich wie bei OSM kaum Veränderungen für unterschiedliche Eingabeformate auf, jedoch liegt hier die Fehlerrate bei etwas über 50 %, weshalb auch hier eine nahezu quadratische Form in der Abbildung 5.8 ersichtlich wird. Beim Dienst Here nimmt die Erfolgsrate bei Hinzunahme der Postleitzahl zu, sodass circa 63 % aller Tests erfolgreich sind. Ohne Postleitzahl liegt die Erfolgsrate bei circa 55% für das Format Hausnummer Straße Ort (HSO) beziehungsweise bei etwa 56 % für das Format Hausnummer Straße Ort Bundesstaat (HSOB). Für den Dienst ist die Darstellung in Abbildung 5.8 somit trapezförmig. Grundlegend schneidet Here bei den Online-Geodiensten am besten ab. Darauf folgen die Dienste Bing Maps und Google Maps mit einer ähnlichen Fehlerrate. Lediglich bei beim Eingabeformat Hausnummer Straße Ort (HSO) schneidet Bing Maps deutlich schlechter ab als Google Maps. OSM hat mit Abstand die schlechteste Fehlerrate.

5.4.4 Szenario 4: Äquivalenzklassentest: Berlin

In Tabelle 5.11 und 5.12 sind die Ergebnisse für Szenario vier dargestellt. Die Abbildung 5.9 visualisiert die Ergebnisse zusätzlich. Der Anhang B.4 enthält weitere Resultate.

Tabelle 5.11 listet die Resultate der Äquivalenzklassentests in Berlin für das Eingabeformat Straße Ort (SO) auf. In Tabelle 5.12 hingegen befinden sich die Ergebnisse für das Format Straße Ort Postleitzahl (SOP). Zum Testen wurden die Repräsentanten aus den Äquivalenzklassen verwendet.

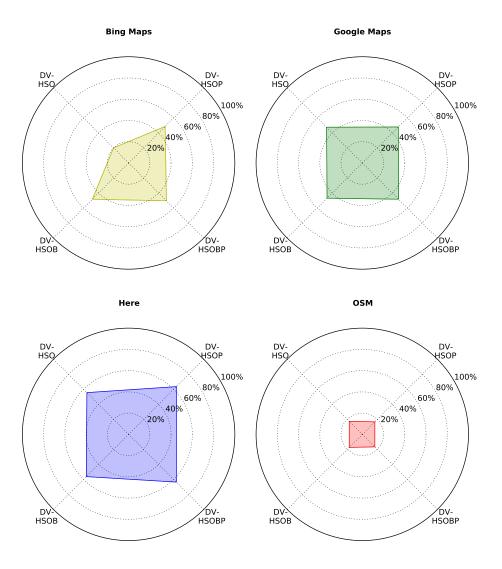


Abbildung 5.8.: Szenario 3: Übersicht - Bestandene Tests (100 % abzüglich Fehlerrate)

GIS	Erfolg	Fehler	Fehlerrate
Bing Maps	52	58	$52{,}73\%$
Google Maps	56	54	$49{,}09\%$
Here	83	27	$24{,}55\%$
OSM	69	41	$37{,}27\%$

Tabelle 5.11.: Szenario 4: Straße Ort (SO)

G	IS	Erfolg	Fehler	Fehlerrate
Bing Ma	ps	78	32	$29{,}09\%$
Google Ma	ps	70	40	$36{,}36\%$
${ m He}$	ere	102	8	$7{,}27\%$
OS	M	69	41	$37{,}27\%$
Не	ere	102	8	$7,\!27\%$

Tabelle 5.12.: Szenario 4: Straße Ort Postleitzahl (SOP)

Die Fehlerrate verringert sich bei Hinzunahme der Postleitzahl bei allen Diensten außer OSM. Ein Grund hierfür ist, dass für einige Eingaben keine Postleitzahl zurückgegeben wird (z.B. wenn eine Straße durch mehrere Postleitzahlbereiche verläuft), wenn diese nicht Teil der Eingabe ist. Teilweise wird die Postleitzahl inkonsequenterweise auch bei Straßen nicht zurückgegeben, wenn die Straße nur für eine Postleitzahl vorhanden ist. Ist die erwartete Postleitzahl jedoch Teil der Eingabe, dann wird sie auch korrekt zurück gegeben. Da der Test jedoch immer eine Postleitzahl erwartet, schlagen einige Tests fehl.

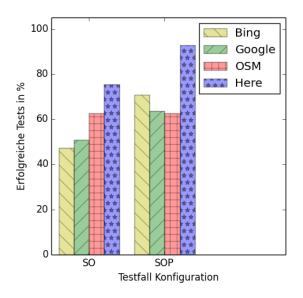


Abbildung 5.9.: Szenario 4: Übersicht - Bestandene Tests (100 % abzüglich Fehlerrate)

Die Ergebnisse von Here sind mit und ohne Eingabe der Postleitzahl am Besten. Die OSM-Ergebnisse sind zu etwa zwei Drittel korrekt, wobei die zusätzliche Eingabe der Postleitzahl keine Änderung erzeugt. Hier wird immer die entsprechende Postleitzahl mit zurückgegeben. Google Maps und Bing Maps finden für ungefähr die Hälfte aller Tests das korrekte Ergebnis wenn die Postleitzahl nicht Teil der Eingabe ist. Wird sie mit eingegeben hat Bing Maps mit einer Fehlerrate von 29 % einen Vorteil gegenüber Google Maps mit 36 %. Die Ergebnisse hängen stark vom Eingabeformat ab. Here hat immer die geringste Fehlerrate. Für das Format $Stra\betae\ Ort\ (SO)$ liegt OSM vor Google Maps und Bing Maps. Bei der Hinzunahme der Postleitzahl hingegen wird das Bild gedreht, sodass die Reihenfolge dann Bing Maps, Google Maps und schließlich OSM ist.

5.4. Testergebnisse 67

5.4.5 Szenario 5: Mehrfach vorkommende Straßennamen: Berlin

Tabelle 5.13 und 5.14 zeigt die Ergebnisse für das fünfte Szenario. Abbildung 5.10 visualisiert die Ergebnisse. In Anhang B.5 befinden sich weitere Resultate.

GIS	Erfolg	Fehler	Fehlerrate
Bing Maps	72	180	$71{,}43\%$
Google Maps	55	197	$78{,}17\%$
Here	219	33	$13{,}10\%$
OSM	164	88	$34{,}92\%$

Tabelle 5.13.: Szenario 5: Straße Ort (SO)

GIS	Erfolg	Fehler	Fehlerrate
Bing Maps	210	42	$16{,}67\%$
Google Maps	150	102	$40{,}48\%$
Here	250	2	$0{,}79\%$
OSM	173	79	$31{,}35\%$

Tabelle 5.14.: Szenario 5: Straße Ort Postleitzahl (SOP)

Tabelle 5.13 zeigt die Resultate für eine Auswahl mehrfacher Straßen in Berlin, bei denen der Testfall das Format Straße Ort (SO) hat. In Tabelle 5.14 befinden sich die Ergebnisse für die Testfälle mit dem Format Straße Ort Postleitzahl (SOP).

Der Unterschied zwischen der Eingaben mit und ohne Postleitzahl ist bei allen Diensten deutlich zu sehen. Here findet circa 87 % aller Adresse korrekt, auch wenn die Postleitzahl fehlt. Mit Postleitzahl findet der Dienste nahezu alle Ergebnisse. Ein sehr deutlicher Unterschied ist bei Bing Maps zu sehen. Der Dienst findet weniger als ein Drittel aller Adressen wenn die Postleitzahl nicht eingegeben wird, jedoch sind circa 83 % aller Tests erfolgreich, wenn die Postleitzahl angefügt wird. Google Maps hat die meisten Probleme die korrekte Adresse zu finden, wenn die Postleitzahl fehlt. Hier laufen nur rund ein Fünftel aller Tests durch. Mit Hinzunahme der Postleitzahl sind es drei Fünftel aller Tests. Die Erfolgsquote bei OSM bleibt bei beiden Eingabeformate bei ungefähr zwei Drittel, wobei sie mit Postleitzahl ein wenig höher ist.

Das Szenario macht deutlich, dass viele Dienste die korrekte Adresse erst mit Eingabe der richtigen Postleitzahl finden, da sie nicht sämtliche Ergebnisse für alle Postleitzahlen präsentieren. Dabei fallen die erfolgreichen Resultate je nach Geodienst unterschiedlich stark aus. Bei Google Maps und Bing Maps sind die Trefferwahrscheinlichkeiten sehr stark von der Postleitzahl abhängig, während die anderen Dienste sich nur gering verbessern. Imposant ist das Ergebnis des Dienstes Here, welche auch ohne Eingabe der Postleitzahlen bereits besser abschneidet als Google Maps, Bing Maps und OSM. Mit Eingabe der Postleitzahlen besteht Here fast alle Tests. Allgemein hat Here bei den Online-Geodiensten wieder die geringste Fehlerrate, Google Maps hingegen die größte. OSM ist für die Konfiguration Straße Ort (SO)

wesentlich besser als Bing Maps, für das Format Straße Ort Postleitzahl (SOP) hingegen ist Bing Maps etwas besser.

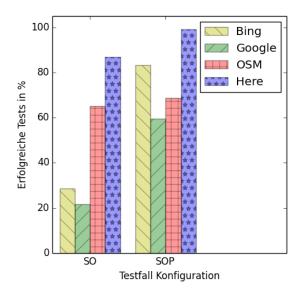


Abbildung 5.10.: Szenario 5: Übersicht - Bestandene Tests (100 % abzüglich Fehlerrate)

5.4.6 Szenario 6: Sprachen: diakritische Zeichen: Prag

Tabelle 5.15 zeigt die Ergebnisse für das sechste Szenario. Zusätzlich werden die Ergebnisse in Abbildung 5.11 dargestellt. Der Anhang B.6 beinhaltet weitere Auswertungsdaten.

GIS	Erfolg	Fehler	Fehlerrate
Bing Maps	141	15	$9{,}62\%$
Google Maps	112	44	$28{,}21\%$
Here	137	19	$12{,}18\%$
OSM	102	54	$34{,}62\%$

Tabelle 5.15.: Szenario 6: Tschechisch: Straße Ort (SO)

Der Test für diakritische Zeichen in Prag hat bei allen Diensten eine Erfolgsrate von über 65 %. . Bing Maps hat eine sehr hohe Erfolgsrate von 90 %, dicht gefolgt vom Dienst Here mit fast 88 %. Google Maps schneidet etwas schlechter ab und besitzt eine Erfolgsrate von knapp 72 %. OSM schneidet am schlechtesten ab und findet zu circa zwei Dritteln aller Anfragen das korrekte Ergebnis. Allerdings gibt es bei OSM im Gegensatz zu den anderen Diensten bei einigen Anfragen kein Resultat. Der Rest der Fehler resultiert aus Abgleichfehlern mit den Soll-Daten. Google Maps findet für einige Resultate nicht den korrekten Staat. Ansonsten liegt der Abgleichfehler oft an einer falschen Postleitzahl oder

5.4. Testergebnisse 69

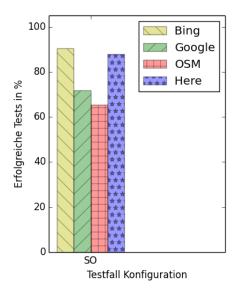


Abbildung 5.11.: Szenario 6: Übersicht - Bestandene Tests (100 % abzüglich Fehlerrate)

an einer Kombination aus falschen Elementen (meist Postleitzahl und Straße). Teilweise ist lediglich eine falsche Straßenbezeichnung für den Fehler verantwortlich.

5.4.7 Szenario 7: Sprachen: kyrillisch: Moskau

Tabelle 5.16 zeigt die Ergebnisse für das siebente Szenario. Zusätzlich werden die Ergebnisse in Abbildung 5.12 dargestellt. Der Anhang B.7 beinhaltet weitere Auswertungsdaten.

_				
	GIS	Erfolg	Fehler	Fehlerrate
	Bing Maps	47	15	$24{,}19\%$
(Google Maps	0	62	$100,\!00\%$
	Here	36	26	$41{,}94\%$
	OSM	0	62	$100{,}00\%$

Tabelle 5.16.: Szenario 7: Russisch: Straße Ort (SO)

Für das siebente Szenario mit der Eingabesprache russisch liegen die Ergebnisse sehr weit auseinander. Bing Maps besteht circa drei Viertel aller Tests. Here hat eine Erfolgsrate von circa 58 %. Bing Maps findet für einen Test kein Resultat, ansonsten sind sämtliche Fehler der beiden Dienste auf Abgleichfehler zurückzuführen. Bei Bing Maps werden diese durch Postleitzahl-Fehler oder mehrfach fehlerhafte Elemente erzeugt, bei Here sind zusätzlich noch fehlerhafte Straßen dafür verantwortlich. Auffällig ist, dass für Google Maps und OSM kein einziger erfolgreicher Testfall existiert. Bei Google Maps bestehen alle fehlerhaften Testfälle aus Abgleichfehlern, welche meist aufgrund eines falschen Staates fehlschlagen. Bei OSM gibt es hingegen teilweise kein Ergebnis. Weiterhin fällt auf, dass bei beiden Diensten teilweise der Förderationskreis (Russisches Verwaltungseinheit ähnlich zu Staat) ein Fehlerelement

ist oder sich mehrfach fehlerhafte Elemente in den Resultaten befinden. Beide Dienste geben als Förderationskreis Moskau zurück, welcher jedoch Zentralrussland ist. Hinweis: Die verwendete Datenbank zur Erzeugung der Testfälle liefert für das Element Bundesstaat immer die größte Verwaltungseinheit auf nationaler Ebene. Diese ist im Beispiel Russland der Förderationskreis. Google Maps und OSM hingegen liefern eine kleinere Verwaltungseinheit (Förderationssubjekt) zurück, weshalb diese Fehler auftreten.

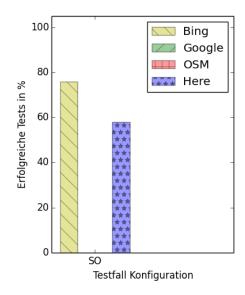


Abbildung 5.12.: Szenario 7: Übersicht - Bestandene Tests (100 % abzüglich Fehlerrate)

5.4.8 Szenario 8: Sprachen: Griechisch: Athen

Tabelle 5.17 zeigt die Ergebnisse für das achte Szenario. Zusätzlich werden die Ergebnisse in Abbildung 5.13 dargestellt. Der Anhang B.8 beinhaltet weitere Auswertungsdaten.

GIS	Erfolg	Fehler	Fehlerrate
Bing Maps	8	35	81,40 %
Google Maps	0	43	$100{,}00\%$
Here	6	37	$86{,}05\%$
OSM	0	43	$100{,}00\%$

Tabelle 5.17.: Auswertung: Szenario 8: Griechisch: Straße Ort (SO)

Das achte Szenario für die griechische Sprache in Athen weist eine sehr hohe Fehlerquote bei den Online-Geodiensten auf. Bing Maps liefert für nahezu 19 % der Testfälle das richtige Ergebnis, Here für knapp 14 % der Testfälle. Teilweise gibt es für die Dienste jedoch kein Resultat. Die Abgleichfehler werden durch fehlerhafte Postleitzahlen oder mehrfach falsche Adresselemente (meist Kombination Postleitzahl und Straße) verursacht. Für Google Maps und OSM existieren, wie bereits beim vorherigen Szenario, keine erfolgreichen Testfälle. Bei

Google Maps existieren lediglich Abgleichfehler, bei OSM wurde zudem bei der Mehrzahl der Tests kein Resultat gefunden. Bei Google Maps existieren die Abgleichfehler aufgrund des fehlerhaften Staates (keine Rückgabe anstatt Sollwert Attika) oder einer Straße, überwiegend jedoch aufgrund mehrfach falscher Elemente (meist Kombination Postleitzahl und Straße). Bei OSM bestehen die Fehler wegen falscher Straßenbezeichnungen oder mehrfach falschen Elementen. Die Aussagekraft dieses Tests sollte jedoch aufgrund der hohen Fehlerquoten aller Online-Geodienste mit Bedacht betrachtet werden. Eventuell muss der Abgleichvorgang für andere Eingabealphabete angepasst werden.

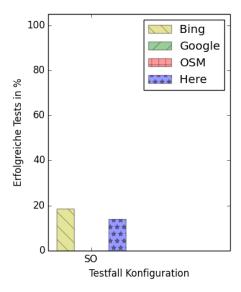


Abbildung 5.13.: Szenario 8: Übersicht - Bestandene Tests (100 % abzüglich Fehlerrate)

5.4.9 Szenario 9: Zufällige und geografische Verteilung: Australien

In Abbildung 5.14 wird die Auswertung des neunten Szenarios visualisiert. In den Tabellen 5.18 bis 5.20 befinden sich die dazu gehörigen Werte. Der Anhang B.9 enthält weitere Daten sowie Fehlergründe.

	Zufällige Verteilung (ZV)			Gleio	Gleichverteilung (GV)			
GIS	Erfolg	Fehler	Fehlerrate	Erfolg	Fehler	Fehlerrate		
Bing Maps	923	1115	54,71 %	57	1981	97,20 %		
Google Maps	1413	625	$30{,}67\%$	748	1290	$63{,}30\%$		
Here	1765	273	$13{,}40\%$	1670	368	$18{,}06\%$		
OSM	45	1993	$97{,}79\%$	39	1999	$98,\!09\%$		

Tabelle 5.18.: Szenario 9: Zufällige und gleichmäßige Verteilung: Hausnr. Straße Ort (HSO)

	Zufälli	Zufällige Verteilung (ZV)			Gleichverteilung (GV)			
GIS	Erfolg	Fehler	Fehlerrate		Erfolg	Fehler	Fehlerrate	
Bing Maps	1242	796	$39{,}06\%$		499	1539	$75{,}52\%$	
Google Maps	1472	566	$27{,}77\%$		777	1261	$61{,}87\%$	
Here	1835	203	$9{,}96\%$		1707	331	$16{,}24\%$	
OSM	42	1996	$97{,}94\%$		40	1998	$98{,}04\%$	

Tabelle 5.19.: Szenario 9: Zufällige und gleichmäßige Verteilung: Hausnr. Straße Ort Postleitzahl (HSOP)

	Zufälli	Zufällige Verteilung (ZV)			Gleichverteilung (GV)		
GIS	Erfolg	Fehler	Fehlerrate		Erfolg	Fehler	Fehlerrate
Bing Maps	1662	376	$18,\!45\%$		1562	476	$23,\!36\%$
Google Maps	1343	695	$34{,}10\%$		728	1310	$64{,}28\%$
Here	1835	203	$9{,}96\%$		1705	333	$16{,}34\%$
OSM	42	1996	$97{,}94\%$		39	1999	$98{,}09\%$

Tabelle 5.20.: Szenario 9: Zufällige und gleichmäßige Verteilung: Hausnr. Straße Ort Postleitzahl Land (HSOPL)

Tabelle 5.18 listet die Ergebnisse für das Eingabeformat Hausnummer Straβe Ort (HSO) in Australien auf. Die zufällige Verteilung ist hier gegenüber der Gleichverteilung im Vorteil. Während bei Bing Maps fast die Hälfte aller Tests in der zufälligen Verteilung erfolgreich sind, sind es bei Google Maps fast 70 % und bei Here sogar über 86 %. Die erfolgreichen Tests in der Gleichverteilung hingegen verlaufen bei Bing Maps sehr schlecht und liegen bei unter 3 %. Google Maps besteht ungefähr ein Drittel der Tests, Here schafft nahezu 82 %. Wie die Tabellen in Anhang B.9 zeigen, findet Bing Maps in vielen Fällen gar kein Resultat. Wenn jedoch ein Ergebnis gefunden wird, schlägt der Soll-Daten-Abgleich aufgrund mehrere Elemente fehl, die nicht übereinstimmen. Google findet in einigen Fällen ebenfalls kein Resultat, die meisten Fehler werden jedoch durch fehlende Übereinstimmung mit den Soll-Daten erzeugt. Here findet zu fast allen Eingaben ein Ergebnis. Wenn hier ein Test fehlschlägt, dann geschieht dies oft aufgrund einer falschen Postleitzahl gegenüber den Soll-Daten.

In Tabelle 5.19 befinden sich die Ergebnisse für das Format Hausnummer Straße Ort Postleitzahl (HSOP). Bei Bing Maps haben sich die Ergebnisse gegenüber dem ersten Eingabeformat bei beiden Verteilungen deutlich verbessert. Es werden vor allem mehr Ergebnisse zurück gegeben, aber auch die Abgleichfehler haben sich bei beiden Verteilungen reduziert. Bei Google Maps und Here gibt es ebenfalls Verbesserungen, diese fallen aber geringer aus (circa 2 % bis 3 %). Here findet bei der zufälligen Verteilung zu nahezu jeder Eingabe ein Ergebnis, bei der Gleichverteilung findet der Dienst bei jeder Eingabe ein Ergebnis. Auch Google Maps findet bei beiden Verteilungen mehr Resultate zu den Testeingaben. Allerdings schlägt hier öfter der Soll-Daten-Abgleich fehl.

Tabelle 5.20 zeigt schließlich die Ergebnisse für das Format Hausnummer Straße Ort Postleitzahl Land (HSOPL). Für den Dienst Bing Maps wurden durch das Anfügen des Landes in die Testeingabe die Ergebnisse ein weiteres Mal in beiden Verteilungen verbessert. Durch das vollständige Format kann Bing Maps die Ergebnisse zu rund vier Fünftel finden. Zudem wird nun zu fast jeder Eingabe ein Resultat geliefert. Die Ergebnisse für Google Maps haben sich im Gegensatz zu Format Hausnummer Straße Ort Postleitzahl ein wenig verschlechtert. Bei der zufälligen Verteilung sind rund zwei Drittel aller Resultate korrekt, während bei der Gleichverteilung rund ein Drittel aller Ergebnisse korrekt sind. Die Ergebnisse für Here haben sich bei der zufälligen Verteilung nicht geändert, bei der gleichmäßigen Verteilung gab es zwei Treffer weniger.

Auffällig in allen Tests ist das schlechte Abschneiden von OSM. Die Fehlerrate liegt bei allen Tests und Eingabeformaten bei ungefähr 98 %. Bei der Betrachtung der Fehlergründe in Anhang B.9 fällt auf, dass ein genaueres Eingabeformat zu mehr Fehlern führt, die aufgrund eines nicht vorhandenen Resultates beruhen. Für das Format Hausnummer Straße Ort werden in der zufälligen Verteilung mehr Fehler durch den Soll-Daten-Abgleich erzeugt. Diese treten aufgrund falscher Postleitzahlen, Hausnummern oder mehreren Abgleichfehlern auf. Bei der Gleichverteilung treten bereits mehr Fehler aufgrund fehlender Ergebnisse auf. Wird das Format genauer (Hausnummer Straße Ort Postleitzahl) werden mehr Fehler aufgrund fehlender Ergebnisse erzeugt. Bei der zufälligen Verteilung treten jedoch noch einige Abgleichfehler vor allem aufgrund falscher oder fehlender Hausnummern auf. Die Tests schlagen bei der Gleichverteilung bei diesem Eingabeformat wesentlich öfter aufgrund fehlender Resultate fehl. Für das dritte Eingabeformat Hausnummer Straße Ort Postleitzahl Land zeigen die Tests ein ähnliches Verhalten wie beim vorherigen Format.

Die Dienste verhalten sich bezüglich der genaueren Eingabeformate unterschiedlich. Bing Maps und Here profitieren von den zusätzlichen angefügten Elementen, welche die Adresse weiter spezifizieren. Der große Anstieg der erfolgreichen Tests bei Bing Maps lässt sich sicherlich darauf zurückführen, dass als der "Culture Code" auf Deutsch gestellt wurde und dabei nicht nur die Rückgabesprache festgelegt wird, sondern auch das Land, in welchem primär gesucht werden soll. Google Maps profitiert zwar von den zusätzlich angefügten Postleitzahlen, aber die Hinzunahme des Landes in die Testeingabe hat die Fehlerrate wieder erhöht. OSM findet nur sehr wenige Ergebnisse, wahrscheinlich fehlen hierbei die Daten. Da einige Tests aufgrund der fehlenden oder inkorrekten Hausnummer gegenüber den Soll-Daten fehlschlugen, ist anzunehmen, dass OSM besser abschneidet, wenn Hausnummern ignoriert werden würden. Das könnte daran liegen, dass die Straßen in der Datenbank enthalten sind, diese aber (noch) nicht nummeriert wurden.

Für dieses Szenario schneidet Here bei den Online Diensten wieder am besten ab. Google Maps schneidet bei weniger Eingabe am zweitbesten ab, gefolgt von Bing Maps. Mit zunehmender Genauigkeit in der Eingabe jedoch verringert sich die Fehlerrate bei Bing Maps, sodass der Dienst nun besser abschneidet als Google Maps. OSM besitzt aufgrund fehlender Daten die höchste Fehlerrate.

5.5 Bewertung der Methode und des Werkzeuges

Durch die Verwendung eines Testgenerators lassen sich mit Hilfe von unterschiedlichen Strategien Testfälle für bestimmte Szenarien erzeugen. Der in der Arbeit umgesetzte Generator kann dabei beispielsweise verschiedene geografische Verteilungen erzeugen, sodass Regionen

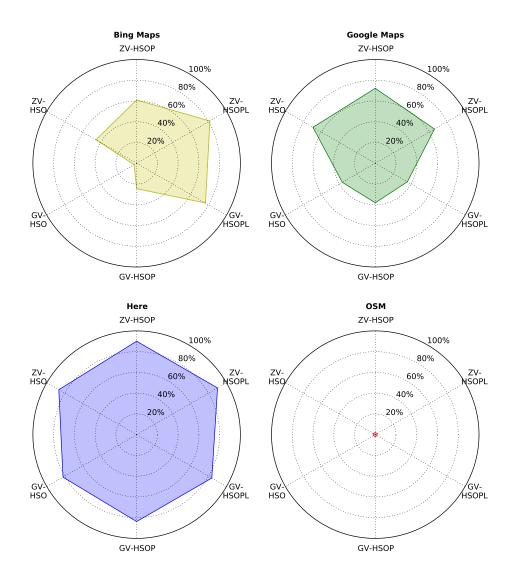


Abbildung 5.14.: Szenario 9: Übersicht - Bestandene Tests (100 %abzüglich Fehlerrate)

mit einer zufälligen Verteilung, einer gleichmäßigen Verteilung oder einer gleichmäßigen Verteilung mit Einbeziehen der Testfalldichte getestet werden können. Damit lässt sich vor allem die Datenvollständigkeit untersuchen. Des Weiteren können für eine Region Äquivalenzklassen aus den Adressdaten gebildet werden, um aus diesen Klassen lediglich ein Element zu testen, damit so der Testumfang reduziert wird. Außerdem können die Testfälle mit dem Generator in unterschiedlichen Sprachen und Alphabeten erzeugt werden. Das ermöglicht die Untersuchung der Geocoding-Software auf die Erkennung der Eingaben. Neben der Testfalleingabe, welche in ihrer Form durch den Anwender konfigurierbar ist, generiert das Tool die Soll-Daten, um später die Ergebnisse abzugleichen. Der Generator ermöglicht es somit unterschiedliche Geodienste für verschiedene Anwendungsfälle und Szenarien zu testen.

Das Testen von Online-Geodiensten kann beispielsweise über die REST-API der Dienste erfolgen. Dafür wird die konkrete Testfalleingabe des Generators an den Dienst gesendet, welcher anschließend ein Ergebnis liefert. Da unterschiedliche Dienste verwendet werden, müssen die Resultate in ein einheitliches Format gebracht werden, um sie einfacher mit den Soll-Daten abzugleichen. Eventuell müssen hierbei einige Elemente der Resultate zusätzlich vereinheitlicht werden, wenn bei verschiedenen Diensten unterschiedliche Bezeichnungen mit gleicher semantischer Bedeutung geliefert werden (z.B. "USA", "Vereinigte Staaten von Amerika", "Vereinigte Staaten). Im nächsten Schritt werden die in ein einheitliches Format übertragenen Daten mit den Soll-Daten abgeglichen. Dabei können Testfälle fehlschlagen, wenn es einerseits kein Resultat gibt und andererseits die Ist-Daten nicht mit den Soll-Daten übereinstimmen. Da für eine Eingabe mehrere Resultate zurückgegeben werden können, müssen sämtliche Ergebnisse verglichen werden. Stimmen beim Abgleich einzelne oder mehrere Elemente nicht überein, so wird der Testfall als fehlgeschlagen markiert. Der Fehlergrund kann anschließend in der Auswertung nachvollzogen werden.

Die Erzeugung der Testdaten, sowie das anschließende Testen bietet die Möglichkeit unterschiedliche Geodienste zu vergleichen. Dies kann einerseits hinsichtlich ihrer Datenvollständigkeit geschehen und andererseits hinsichtlich der Erkennung von Testeingaben.

Wichtig für das Erzeugen der Testfälle ist hierbei die verwendete Adressdatenbank. Da die Datenbanken von Geodiensten in der Regel nicht öffentlich zugänglich sind, muss hierfür eine Lösung gefunden werden. Eine Möglichkeit ist die Verwendung der öffentlich zugänglichen Datenbank von OSM. Da die Struktur der Daten jedoch teilweise nicht konsequent umgesetzt ist und vor allem weil die Datenvollständigkeit für eine weit reichende Testabdeckung zu gering ist, wurde für die durchgeführten Tests eine kommerzielle NDS-Datenbank verwendet. Der Vorteil liegt in der hohen Datenqualität der Datenbank, welche jedoch nicht garantiert ist, auch wenn es sich um ein kommerzielles Produkt handelt.

5.5.1 Probleme bei der Testdurchführung

Bei der Testdurchführung traten einige Probleme auf, welche bei der Auswertung beachtet und behandelt werden müssen.

Beim Abgleichen der Ist-Daten mit den Soll-Daten trat oft der Fall auf, dass Postleitzahlen Grund für den fehlgeschlagenen Testfall sind. Bei der Untersuchung der Testfälle hat sich schließlich herausgestellt, dass die Datenbank teilweise zu viele Postleitzahlen zu einer Adresse zuordnet. Neben der richtigen Postleitzahl kann beispielsweise die Postleitzahl eines Nachbarortes für eine Straße zugeordnet sein. Der Testgenerator wählt dabei zufällig eines

dieser Ergebnisse aus, weshalb er neben einem Ergebnis mit der korrekten Postleitzahl auch ein Ergebnis mit der inkorrekten Postleitzahl wählen kann.

Die Geodienste unterscheiden sich zudem in ihren Rückgaben. Deutlich wird dies beim Testen von Moskau im siebenten Szenario (Abschnitt 5.2.7 und 5.4.7). Während Bing Maps und Here die größte Verwaltungseinheit auf nationaler Ebene zurückgeben (Förderationskreis) geben Google Maps und OSM eine kleinere Verwaltungseinheit (Förderationssubjekt) zurück.

Ein anderes Problem besteht darin, dass die Geocoding Software von Bing Maps keine Resultate für den Stadtbezirk zurück gibt. Im Gegensatz zu den anderen Diensten, welche ein Element sublocality, district oder suburb haben, ist dieses Element bei Bing Maps nicht vorhanden. Es kann also für den Vergleich mit den Soll-Daten nicht verwendet werden. Jedoch ist bei der Auswertung aufgefallen, dass der Dienst den Bezirk in vielen Fällen als Ort angibt, dafür aber der eigentliche Ortsname nicht im Resultat vorhanden ist. Somit fehlt diese Information in vielen Fällen.

Ein weiterer Unterschied von Bing Maps gegenüber den anderen Diensten ist die Rückgabe der Straße. Hier ist die Hausnummer und die Straße in einem Element enthalten. Die anderen Dienste geben beide Elemente einzeln zurück. Da die Position der Hausnummer in den einzelnen Ländern unterschiedlich ist (z.B. Deutschland: nach dem Straßennamen, USA: vor dem Straßennamen), können beide Elemente nur mit dem Hintergrundwissen über die Position der Hausnummer getrennt werden. Die Rückgabe der Elemente als einzelne Token wäre hier sinnvoller und würden einen einfacheren Zugriff ermöglichen.

5.6 Testbewertung der Geodienste

Nach der Durchführung sämtlicher Tests in den vorgestellten Szenarien können die getesteten Geodienste untereinander verglichen werden. Zu diesem Zweck wurden die Fehlerraten aller Tests gemittelt und in Tabelle 5.21 festgehalten.

GIS	Fehlerrate	Erfolgsrate
Bing Maps	$46{,}17\%$	$53,\!83\%$
Google Maps	$48{,}43\%$	$51{,}57\%$
Here	$28{,}85\%$	$71{,}15\%$
OSM	$76{,}45\%$	$23{,}55\%$

Tabelle 5.21.: Fehler- und Erfolgsraten über alle Tests

Die Tabelle zeigt eindeutig, dass Here von den getesteten Diensten mit über 71 % am besten abschneidet. Der Dienst schneidet in fast allen Tests besser ab als die Dienste von Bing Maps, Google Maps und OSM. Auch bei wenig Informationen in der Eingabe (z.B. Hausnummer Straße Ort) findet Here in einigen Fällen mehr Resultate als die anderen Dienste mit einem spezifischeren Eingabeformat. Die Differenz zwischen den erfolgreichen Tests für unterschiedlich spezifische Eingabeformate ist bei Here relativ gering. Die richtigen Ergebnisse werden also auch oft mit einem Eingabeformat gefunden, welches wenig Adresselemente

beinhaltet. Die Dienste Bing Maps und Google Maps schneiden ähnlich ab und finden über die Hälfte aller Ergebnisse korrekt. Es gibt zwischen den beiden Diensten jedoch Unterschiede, welche vom jeweiligen Szenario abhängen. Beispielsweise profitiert der Dienst Bing Maps in einigen Tests sehr deutlich von einer genaueren Eingabe. Teilweise wird für Formate mit wenigen Adresselementen kaum etwas gefunden, mit mehr Eingabeelementen hingegen steigt die Erfolgsrate stark an. Dieser Umstand ist vor allem im neunten Szenario gut sichtbar. Bei Google Maps steigt die Erfolgsrate mit spezifischeren Eingabeformate weniger stark an, teilweise bleibt sie sogar gleich. Für die Erkennung von mehrfach auftretenden Straßen in einer Stadt ist die Hinzunahme der Postleitzahl für beide Dienste jedoch wichtig um die korrekten Adressen zu erhalten. Zudem schneidet Bing Maps besser bei der Nutzung anderer Sprachen ab. Dieser Umstand resultiert jedoch teilweise aus der Rückgabe für den Bundesstaat, welche bei Google Maps nicht immer korrekt ist oder es sich teilweise um eine leere Eingabe handelt. Die Tests für OSM sind im Mittel lediglich zu knapp ein Viertel erfolgreich. Da hier jedoch oft kein Resultat zurückgegeben wurde, kann darauf geschlossen werden, dass die Daten wahrscheinlich nicht in der OSM-Datenbank vorhanden sind. Sind die Daten jedoch vorhanden und wurden für die Testeingaben Ergebnisse gefunden, dann fällt auf, dass ein spezifischeres Eingabeformat meist nicht dazu beiträgt mehr Ergebnisse zu finden als bei einem Eingabeformat mit weniger Adresselementen. Bei den Szenarien vier und fünf schneidet der Dienst relativ gut ab und weist ähnliche Erfolgsraten auf, wie die Dienste Bing Maps und Google Maps. Das liegt wahrscheinlich an der Auswahl der Testfälle in Berlin. Da es sich hierbei um ein dicht besiedeltes Gebiet handelt, ist die Datenqualität hier wahrscheinlich höher als in anderen (ländlichen) Gebieten.

5.7 Beziehungen zwischen Eingabe und Ausgabe

Bei der Auswertung der Tests muss beachtet werden, dass die Ausgaben teilweise in ihrer Syntax von der Eingabe und den Soll-Daten abweichen. Wenn dies nach festen Regeln geschieht, lassen sich die Abweichungen leicht behandeln. Allerdings können die Daten auch unregelmäßig abweichen, sodass beim Abgleichen der Ergebnisse mit den Soll-Daten Fehler aufgrund der unterschiedlichen Syntax auftreten, obwohl die Werte semantisch gleich sind.

Eine dieser syntaktischen Abweichungen ist die Rückgabe von abgekürzten Namen, welche beispielsweise das Land betreffen. Bei Here und Google Maps wird zusätzlich der komplette Name geliefert. Bing Maps hingegen liefert nur eine Form. Ein Beispiel hierfür sind die "Vereinigten Staaten von Amerika". Diese werden teilweise auch als "Vereinigte Staaten" oder lediglich als "USA" vom Geodienst bezeichnet. Hierfür lässt sich jedoch leicht ein Mapping dieser unterschiedlichen Rückgabenamen erstellen. Die Werte des Mappings können dann mit den Soll-Daten verglichen werden.

Ein Problem kann in der Schreibweise von einzelnen Elementen existieren. Beispielsweise werden einige Straßen anstatt mit einem "ß" mit "ss" zurückgegeben. Für den Fall der deutschen Sprache ist das Problem leicht zu beheben, jedoch kann es in anderen Sprachen ähnliche Spezialfälle geben, welche zuerst erfasst werden müssen, damit eine Lösung für das Problem erstellt werden kann. Wenn hierfür feste Regeln existieren, ist die Behandlung des Problem relativ simpel. Wenn solche Unterschiede in den Schreibweisen jedoch unregelmäßig auftreten, lassen sich die Probleme unter Umständen schwer behandeln.

Zudem kam es vor, dass einige Straßen Präfixe oder Suffixe verwenden, welche so nicht in

den Soll-Daten vorhanden sind. Beispielsweise wurde die erwartete Straße "Lamkes Road" bei einigen Diensten als "Lamke Rd" zurück gegeben. Für diesen speziellen Fall stellt eine Lösung natürlich kein Problem dar, jedoch kann der Umstand für verschiedenste Präfixe und Suffixe auftreten (z.B. "Boulevard" und "Blvd", "First" und "1st", "North" und "N"). Ohne das Vorwissen, welche Abkürzungen in den unterschiedlichen Testregionen existieren, können hierbei immer wieder Fehler auftreten, welche eigentlich keine sind, da der Inhalt semantisch äquivalent ist.

Weiterhin ist zu beachten, dass Elemente in den verschiedenen Diensten unter unterschiedlichen Namen existieren können. So kann es Straßen geben, welche aus mehreren Elementen bestehen, jedoch bei einigen Diensten nicht alle dieser Elemente der Straße abgespeichert sind (z.B. "Miller St" und "Miller"). Ebenso kann dies für die Bezeichnung von Orten oder Bezirken vorkommen. Ein Beispiel hierfür ist die Bezeichnung des Ortes "Navy Town" auf der Insel Attu Island in Alaska. Die Bezeichnung für den Ort unterscheidet sich bei den Geodiensten, sowie bei der Datenbank, welche für die Testerzeugung genutzt wurde. Dieser Umstand kann vor allem in entlegenen Gebieten auftreten und sollte daher nicht missachtet werden. Es ist jedoch schwer hierfür eine Lösung zu finden, da diese Fälle unregelmäßig auftreten. Während das Problem fehlender Token in einem Element relativ einfach zu handhaben ist, stellt die unterschiedliche Benennung ein schwerwiegendes Problem dar, welches unter Umständen nur manuell behoben werden kann.

Ein weiterer Umstand ist die unter Umständen unterschiedliche Zeichenverwendung bei Adresselementen. So kann nicht ausgeschlossen werden, dass beispielsweise eine Straße welche Bindestriche enthält in einer Geodatenbank mit Leerzeichen eingetragen ist, oder dass Leerzeichen zu Bindestrichen geändert wurden.

Die erläuterten Umstände sollten bei der Auswertung beachtet werden, da sie ansonsten zu fehlerbehafteten Ergebnissen führen können.

5.8 Gütebestimmung

In den vorgestellten und getesteten Szenarien wurden die Testerzeugungsstrategien des umgesetzten Testgenerators benutzt, um verschiedene Bereiche abzudecken. Dabei wurden unterschiedliche Regionen und Eingabesprachen getestet, um ein hohes Überdeckungsmaß zu erreichen. Jedoch sind bei den durchgeführten Tests nicht sämtliche Anforderungen an die Geodienste abgedeckt worden. Da die Anforderungen an die Dienste jedoch relativ trivial sind, wäre eine hohe Abdeckung dieser wünschenswert gewesen. Die Anforderungen beschreiben, dass ein Nutzer nach der Eingabe einer mitunter nicht vollständigen Adresse ein Resultat erhalten möchte. Da einige Tests jedoch kein Ergebnis hervor brachten, wurden diese Anforderungen nur teilweise abgedeckt. Dieser Umstand kann aus fehlenden Daten in der Datenbank oder schlechter Erkennung der eingegebenen Adresse resultieren. Andere Tests hingegen erzeugten ein Resultat, welches jedoch nicht mit den Soll-Daten übereinstimmte, sodass nicht die korrekte Adresse gefunden wurde oder Elemente der Adresse nicht korrekt oder gar nicht gesetzt waren. Auch eine explizite Angabe aller wichtigen Adresselemente sorgte hierbei teilweise für Fehler.

Die geografische Verteilung hat gegenüber der zufälligen Verteilung in fast allen Fällen schlechter abgeschnitten. Die Dichteverteilung lag im Vergleich prozentual zwischen den

79

anderen Verteilungen. Daraus lässt sich der Rückschluss ziehen, dass in viele Regionen die dicht besiedelten Gebiete besser abgedeckt sind als gering besiedelte Gebiete und somit in diesen Regionen eine höhere Güte besteht. Die gering besiedelten Gebiete sind dabei entweder ungenauer erfasst oder teilweise überhaupt nicht vorhanden. Eine reine randomisierte Testerzeugungsstrategie ist also weniger sinnvoll, wenn es darum geht eine Aussage über die Abdeckung sämtlicher Gebiete der zu untersuchenden Region zu tätigen.

Des Weiteren konnte festgestellt werden, dass eine detailliertere Eingabe meist bessere Ergebnisse und somit eine höhere Güte liefert. Teilweise ist die Güte bei einigen Szenarien und Geodiensten jedoch auch mit einer weniger detaillierten Eingabe höher.

KAPITEL 6

Fazit

In dieser Arbeit wurde das Konzept eines Testgenerators beschrieben, welcher automatisiert Testfälle erzeugt, um Geodienste zu testen. Es wurden Begriffe und Konzepte des Softwaretestens auf den Fall der Geodienste bezogen. Zudem wurde untersucht, welche Datenbank am besten geeignet ist, um reale Adressen zu generieren. Anschließend wurde die grundlegende Konzipierung eines Testgenerators vorgestellt und wie dieser arbeitet. Weiterhin wurde erklärt, wie es möglich ist, Äquivalenzklassen für Adressen zu bilden und welche verschiedenen Strategien es gibt, um Testfälle auf unterschiedliche Art und Weise zu verteilen. Es handelt sich dabei um eine zufällige, eine gleichmäßige und eine testfalldichtebasierende Verteilung der Testfälle, welche auf unterschiedliche politische Regionen angewendet werden kann. Zudem ermöglicht der Generator die Erzeugung von Testfällen in mehreren Sprachen. Des Weiteren kann der Generator die Testfälle unterschiedlich spezifisch bezüglich des Eingabeformates generieren. Die Strategien wurden anschließend in einem Testgenerator umgesetzt. Nachfolgend wurde der Generator verwendet, um die unterschiedlichen Strategien zur Testfallerzeugung in einer Fallstudie zu untersuchen und zu bewerten. Dabei wurden mehrere Online-Geodienste getestet und deren Ergebnisse ausgewertet und verglichen.

Der Generator eignet sich aufgrund seiner verschiedenen Erzeugungsstrategien gut dazu in den Geodiensten unterschiedliche Regionen zu testen. Dabei kann einerseits intensiv die korrekte Erkennung der Adresselement durch die Geocoding-Software untersucht werden, sowie die fehlerfreie Rückgabe des gewünschten Ergebnisses. Um eine große Menge syntaktisch unterschiedlicher Eingaben zu testen kann zum Beispiel die Äquivalenzklassenbildung verwendet werden, da hier ähnliche Eingabeformate in gleiche Klassen gegliedert werden. Anschließend muss lediglich ein Repräsentant aus der Klasse getestet werden. Des Weiteren kann die Menge der Rückgabe-Resultate untersucht werden, indem in einem Ort mehrfach auftretende Straßen mit unterschiedlichen Eingaben gesucht werden. Zudem ist es möglich den Datenbestand in der Datenbank mit Hilfe der unterschiedlichen Verteilungen zu testen. Die geografisch gleichmäßige Verteilung erzeugt dafür auch Testfälle in entlegenen Regionen, welche mit einer zufälligen Verteilung wahrscheinlich nicht erzeugt werden würden und unter Umständen nicht in der Datenbank des Geodienstes enthalten sind. Die vorgestellten und umgesetzten Konzepte des Testgenerators ermöglichen somit eine umfangreiche Testmöglichkeit für Geodienste, welche sich lediglich durch die erzeugten Eingaben und Eingabebedingungen, wie die Rückgabesprache, steuern lässt und somit einen Blackboxtest

82 6. Fazit

der Dienste ermöglicht.

Die Fallstudie hat gezeigt, dass die mit dem Generator erzeugten Testfälle sehr gut für unterschiedliche Testszenarien geeignet sind. Beim Abgleichen der Resultate mit den Soll-Daten sind jedoch einige Probleme aufgetreten, welche teilweise behoben werden konnten. Allerdings ist hierbei aufgefallen, dass es auch viele nicht triviale Probleme gibt, welche noch zu lösen sind, da sie Auswirkungen auf die finalen Ergebnisse der Tests haben.

Die Tests in der Fallstudie haben dennoch grundlegende Erkenntnisse über die vier Online-Geodienste Bing Maps, Google Maps, Here und OSM offen gelegt. Bei den Online-Diensten schneidet Here mit einer Erfolgsrate von über 71 % am besten ab. Anschließend folgen die Dienste Bing Maps und Google Maps mit nahe beieinander liegenden Resultaten von circa 53 % respektive 51 %. Abgeschlagen liegt der Dienst des nicht-kommerziellen Projektes OpenStreetMap mit 23 %. Die geringe Erfolgsrate liegt hierbei jedoch oft an fehlenden Daten in der Datenbank.

Der Testgenerator ist also sehr gut für die Erzeugung von Testfällen zum Testen von Geodiensten geeignet und kann auch verwendet werden, um weitere Geodienste als die vorgestellten zu testen. Es wird lediglich eine passende Adressdatenbank für die Erzeugung benötigt. Das Testen der Geodienste, beziehungsweise das Abgleichen der Resultate mit den Soll-Daten ist hingegen weniger trivial. Die API-Ergebnisse der Geodienste lassen sich relativ einfach in ein einheitliches Format übertragen. Jedoch müssen dabei einige Sonderfälle betrachtet werden, welche aufgrund unterschiedlicher Rückgabewerte der einzelnen Dienste resultieren.

6.1 Ausblick

Auch wenn der Testgenerator es ermöglicht Testfälle für Geodienste zu generieren, kann er trotzdem noch durch einige Features erweitert werden. Momentan erzeugt der Generator lediglich "normale" Adressen. Eine Erweiterung zur Generierung von Sonderzielen (POIs) wie Restaurants oder Tankstellen wäre denkbar. Diese sind in der verwendeten Datenbank enthalten und werden von den Geodiensten wie Google Maps und Here unterstützt.

Zudem könnten Methoden verwendet werden, um die Eingabe der Testfälle zu manipulieren um Fehleingaben oder Abkürzungen von Nutzern zu simulieren. Beispielsweise könnten fehlende oder überflüssige Buchstaben Tippfehler vortäuschen. Des Weiteren können Eingaben so verändert werden, dass sie Abkürzungen oder Abänderungen der Nutzer simulieren (z.B. "Takustr" anstatt "Takustraße" oder "Otto Suhr Allee" anstatt "Otto-Suhr-Allee").

Eine weitere Möglichkeit den Generator zu ergänzen ist die Untersuchung der korrekten Position gefundener Adressen. So kann mit jedem generierten Testfall die Geoposition der Adresse zu den Soll-Daten hinzugefügt werden. Beim späteren Testen wird die Position aus den Soll-Daten mit der Geoposition der Ist-Daten vom Geodienst verglichen. Hierbei muss unter Umständen die Übereinstimmung in einem gewissen Radius überprüft werden, da die Position nicht auf den Meter genau sein muss, wenn die Daten aus verschiedenen Quellen kommen.

Literaturverzeichnis

- [1] ISO. Quality management systems Fundamentals and vocabulary. ISO 9000:2015, International Organization for Standardization, Geneva, Switzerland, 2015.
- [2] Alexander Schatten, Stefan Biffl, Markus Demolsky, Erik Gostischa-Franta, Thomas Östreicher, and Dietmar Winkler. Best Practice Software-Engineering: Eine praxiserprobte Zusammenstellung von komponentenorientierten Konzepten, Methoden und Werkzeugen. Springer-Verlag, 2010.
- [3] Cem Kaner, Jack Falk, and Hung Quoc Nguyen. Testing Computer Software Second Edition. Dreamtech Press, 2000.
- [4] ISO. Information technology Software product quality. ISO 9126-1:2001, International Organization for Standardization, Geneva, Switzerland, 2001.
- [5] Torsten Cleff. Basiswissen Testen von Software: Vorbereitung zum Certified Tester (Foundation Level) nach ISTQB-Standard. W3l GmbH, 2010.
- [6] Andreas Spillner and Tilo Linz. Basiswissen Softwaretest. dpunkt, 2012.
- [7] Edsger Wybe Dijkstra. Notes on Structured Programming, 1970.
- [8] Glenford J Myers, Corey Sandler, and Tom Badgett. *The Art of Software Testing*. John Wiley & Sons, 2011.
- [9] Approved September. IEEE Standard Glossary of Software Engineering Terminology. Office, 121990(1):1, 1990.
- [10] Ilene Burnstein. Practical Software Testing A Process-Oriented Approach. Springer Science & Business Media, 2003.
- [11] ISO. Geographic information Data quality. ISO 19157:2013, International Organization for Standardization, Geneva, Switzerland, 2013.
- [12] Google Maps. https://www.google.de/maps. Zugriff: 22.10.2015.
- [13] Bing Maps. https://www.bing.com/maps/. Zugriff: 22.10.2015.
- [14] Here. https://www.here.com/. Zugriff: 22.10.2015.
- [15] OpenStreetMap. http://www.openstreetmap.org. Zugriff: 22.10.2015.
- [16] UPU Universal Postal Union. http://www.upu.int. Zugriff: 12.11.2015.
- [17] HERE Platform Data Extension API Developer's Guide Key Concepts. https://developer.here.com/platform-extensions/documentation/platform-data/topics/key-concepts.html. Zugriff: 27.10.2015.

84 Literaturverzeichnis

[18] TM Müller. Navigation Data Standard (NDS): Bald Industriestandard. *Automobil-Elektronik*, 6:30, 2010.

- [19] SQLite. https://www.sqlite.org/. Zugriff: 29.10.2015.
- [20] Bing Maps Coverage. https://msdn.microsoft.com/en-us/library/dd435699. aspx. Zugriff: 10.11.2015.
- [21] HERE Coverage. https://developer.here.com/coverage-info. Zugriff: 10.11.2015.
- [22] MapQuest Geocoding. http://open.mapquestapi.com/nominatim/. Zugriff: 26.11.2015.
- [23] Ina Ludwig, Angi Voss, and Maike Krause-Traudes. Ein Vergleich der Straßennetze von OpenStreetMap und NAVTEQ. Angewandte Geoinformatik 2010, 2010.
- [24] Ove Armbrust, Michael Ochs, and Björn Snoek. Stand der Forschung von Software-Tests und deren Automatisierung, 2004.
- [25] PostgreSQL. http://www.postgresql.org/. Zugriff: 29.10.2015.
- [26] OpenStreetMap Elements. http://wiki.openstreetmap.org/wiki/Elements. Zugriff: 29.10.2015.
- [27] Mordechai Haklay and Patrick Weber. Openstreetmap: User-Generated Street Maps. *Pervasive Computing, IEEE*, 7(4):12–18, 2008.
- [28] Ralph Behrens, Thomas Kleine-Besten, Werner Pöchmüller, and Andreas Engelsberg. Digitale Karten im Navigation Data Standard Format. In *Handbuch Fahrerassistenzsysteme*, pages 513–523. Springer, 2015.
- [29] MF Goodchild. Citizens as Voluntary Sensors: Spatial Data Infrastructure in the World of Web 2.0. International Journal of Spatial Data Infrastructures Research, 2:24–32, 2007.
- [30] Angi Voss, Maike Krause-Traudes, and Ahmet Ocakli. Verbesserungen des Straßennetzes von OpenStreetMap zwischen 2009 und 2011. *VDV-Magazin*, 6, 2011.
- [31] Mordechai Haklay et al. How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets. *Environment and planning*. B, Planning & design, 37(4):682, 2010.
- [32] Ina Ludwig. Abbildung von Straßendaten für Qualitätsuntersuchungen Ein Vergleich von OpenStreetMap mit Navteq. PhD thesis, Diplomarbeit, 2010.
- [33] DHL Postleitzahlen-Formate Weltweit. https://www.dhl.de/content/dam/dhlde/downloads/express/dhl-express-formate-postleitzahlen-062014.pdf. Zugriff: 24.11.2015.
- [34] Toby Segaran. Programming Collective Intelligence: Building Smart Web 2.0 Applications. O'Reilly Media, Inc., 2007.

ANHANG A

Geodienstdaten und -resultate

A.1 In OSM gespeicherte Daten

Der Eintrag für die Takustraße 9, 14195 Berlin in OpenStreetMap XML.

Listing A.1: Takustraße 9 in OSM

A.2 REST API Ergebnisse

Dieser Abschnitt enthält die REST API Resultate für die Anfrage Takustraße 9 Berlin.

A.2.1 Google Maps API Ergebnis

```
1
      "results" : [
2
3
           "address_components" : [
4
5
               "long_name" : "9"
6
               "short name" : "9",
               "types" : [ "street_number" ]
8
             },
10
               "long_name" : "Takustraße",
11
               "short_name" : "Takustraße",
12
               "types" : [ "route" ]
13
14
15
               "long_name" : "Bezirk Steglitz-Zehlendorf",
16
               "short_name" : "Bezirk Steglitz-Zehlendorf"
17
               "types" : [ "sublocality_level_1", "sublocality", "
18
                   political"
19
20
               "long_name" : "Berlin",
"short_name" : "Berlin",
"types" : [ "locality", "political" ]
21
22
23
24
25
               "long_name" : "Berlin",
               "short_name" : "Berlin",
               "types": [ "administrative area level 1", "political" ]
28
29
30
               "long_name" : "Deutschland",
               "short_name" : "DE",
32
               "types" : [ "country", "political" ]
33
34
               "long\_name" : "14195"
36
               "short_name" : "14195"
37
               "types" : [ "postal_code" ]
38
40
          "formatted_address" : "Takustraße 9, 14195 Berlin, Deutschland"
41
           "geometry" : {}
42
             "location"
43
```

```
"lat" : 52.455859,
"lng" : 13.2972068
44
45
46
              "location_type" : "ROOFTOP",
47
              "viewport" : {
48
                 "northeast": {
49
                   "lat" : 52.4572079802915,
50
                   "lng" : 13.2985557802915
51
52
                  southwest" : {
53
                   "lat" : 52.4545100197085,
54
                   "lng" : 13.2958578197085
55
56
              }
57
           },
"place_id": "ChIJWR1UiG9aqEcRkX1o6gfXsPc",
"types": [ "street_address" ]
58
59
60
61
62
       "status" : "OK"
63
64
```

Listing A.2: Takustraße 9 - Google Maps API Result

A.2.2 Here API Ergebnis

```
1
      "Response": {
2
        "MetaInfo": {
3
          "Timestamp": "2015-11-26T13:53:45.804+0000"
4
       } ,
"View": [
5
6
7
              _type": "SearchResultsViewType",
8
            "ViewId": 0,
9
            "Result": [
10
11
                 "Relevance": 1,
12
                 "MatchLevel": "houseNumber",\\
13
                 "MatchQuality": {
14
                   "City": 1,
                   "Street": [
16
                     1
17
                   "HouseNumber": 1
19
                 },
"MatchType": "pointAddress",
20
21
                 "Location": {
22
                   "LocationId": "NT_McwwRf-M9IjdPAUSludlmB_5A",
                   "LocationType": "address",
24
                   "DisplayPosition": {
25
                     "Latitude": 52.4558899,
26
                     "Longitude": 13.29721
                   },
"NavigationPosition": [
28
29
30
                        "Latitude": 52.45585,
31
                        "Longitude": 13.29738
32
                     }
33
                   "MapView": {
35
                     "TopLeft": {
36
                        "Latitude": 52.4570141,
37
                       "Longitude": 13.2953652
                     },
39
                     BottomRight": {
40
                        "Latitude": 52.4547657,
41
                       "Longitude": 13.2990548
                     }
43
44
                   "Address": {
45
                     "Label": "Takustraße 9, 14195 Berlin, Deutschland",
46
                     "Country": "DEU",
47
                     "State": "Berlin"
48
                     "County": "Berlin",
49
                     "City": "Berlin",
50
                     "District": "Dahlem",
51
```

```
"Street": "Takustraße",
52
                          "HouseNumber": "9",
"PostalCode": "14195",
53
54
                           "AdditionalData": [
55
56
                                "value": "Deutschland", 
"key": "CountryName"
57
58
                             \Big\}\;,\\ \Big\{
59
60
                                "value": "Berlin",
61
                                "key": "StateName"
62
63
64
                                "value": "Berlin",
65
                                "key": "CountyName"
66
         } } }
67
68
69
70
71
72
73
74
       }
75
76
```

Listing A.3: Takustraße 9 - Here API Result

A.2.3 Bing Maps API Ergebnis

```
1
     "authenticationResultCode": "ValidCredentials",
2
     "brandLogoUri": "http://dev.virtualearth.net/Branding/
3
         logo_powered_by.png",
     "copyright": "Copyright © 2015 Microsoft and its suppliers. All
4
         rights reserved. This API cannot be accessed and the content and
          any results may not be used, reproduced or transmitted in any
         manner without express written permission from Microsoft
         Corporation.",
     "resourceSets": [
5
6
          "estimatedTotal": 1,
          "resources":
8
            {
9
                 type": "Location: http://schemas.microsoft.com/search/
10
                 local/ws/rest/v1",
              "bbox ":
11
                52.452020282429324,
12
                13.288743943236014,
13
                52.45974571757068,
14
                13.305646056763987
15
16
              "name": "Takustraße 9, 14195 Berlin",
17
              "point": {
18
                "type": "Point".
19
                "coordinates": [
20
                  52.455883,
                  13.297195
22
23
^{24}
              'address": {
25
                "addressLine": "Takustraße 9",
26
                "adminDistrict": "BE",
27
                "adminDistrict2": "Berlin",
                "countryRegion": "Deutschland",
29
                "formattedAddress": "Takustraße 9, 14195 Berlin",
30
                "locality": "Berlin",
31
                "postalCode": "14195"
33
              "confidence": "High",
34
              "entityType": "Address",
35
              "geocodePoints": [
37
                  "type": "Point",
38
                  "coordinates": [
39
                    52.455883,
40
                    13.297195
41
42
                  "calculationMethod": "Rooftop",
43
                  "usageTypes": [
44
                     "Display"
45
```

```
46
47
48
                    "type": "Point",
49
                    "coordinates": [
50
                      52.455804,
51
                      13.2972830000000002\\
52
                    ],
"calculationMethod": "Interpolation",
53
54
                    "usageTypes": [
55
                       "Display",
56
                      "Route"
57
58
                  }
59
60
               "matchCodes": [
61
                    Good
62
63
64
65
66
67
      "statusCode": 200,
68
      "statusDescription": "OK",
69
      " traceId " : " dc06f7b59f9c4fba8d740988b0926bdd | DB40231539
70
          |\,02.00.118.2700\,|\,DB4SCH010061262\,,\;\;DB4SCH010061754\,"
71
```

Listing A.4: Takustraße 9 - Bing Maps API Result

A.2.4 OSM API Ergebnis

```
1
2
        "place_id": "11081385",
3
        "licence": "Data @ OpenStreetMap contributors, ODbL 1.0. http://
4
           www.openstreetmap.org/copyright",
        "osm_type": "node",
5
        "osm_id": "1049544291",
6
        "boundingbox":
          " 52.4558043 " ,
8
          " 52.4558043",
9
          " 13.2970082 "
10
          " 13.2970082 "
11
12
        "lat": "52.4558043",
13
        "lon": "13.2970082",
14
        "display name": "Caféte, 9, Takustraße, Dahlem, Steglitz-
15
           Zehlendorf, Berlin, 14195, Germany",
        "class": "amenity",
16
        "type": "cafe",
17
        "importance": 0.401,
18
        icon": "http://open.mapquestapi.com/nominatim/v1/images/mapicons"
19
           /food_cafe.p.20.png",
        "address": {
20
          "cafe": "Caféte".
21
          "house_number": "9"
22
          "road": "Takustraße",
          "suburb": "Dahlem"
24
          " \mathtt{city\_district} " : " \mathtt{Steglitz} –Zehlendorf " ,
25
          "state": "Berlin".
26
          "postcode": "14195".
27
          "country": "Germany",
28
          "country code": "de"
29
       }
30
     },
31
32
        " place_id " : " 151240157 " ,
33
        "licence": "Data @ OpenStreetMap contributors, ODbL 1.0. http://
34
           www.openstreetmap.org/copyright",
        "osm_type": "relation",
35
        "osm_id": "450685",
36
        "boundingbox": [
37
          "52.45\overline{5}6244",
          " 52.4562927 " ,
39
          "13.2964925"
40
          "13.2977531"
41
        "lat": "52.455955",
43
        "lon": "13.2974508565487",
44
        "display_name": "Computer Science, 9, Takustraße, Dahlem,
45
           Steglitz-Zehlendorf, Berlin, 14195, Germany",
        "class": "place",
46
```

```
"type": "house",
47
        "importance": 0.401,
48
        "address": {
49
          "house": "Computer Science",
50
          "house_number": "9"
51
          "road": "Takustraße",
52
          "suburb": "Dahlem"
53
          "city_district": "Steglitz-Zehlendorf",
54
          "state": "Berlin"
55
          " postcode " : "14195 "
56
          "country": "Germany"
57
          "country_code": "de"
58
        }
59
     },
60
61
        "place_id": "2597461242",
62
        "licence": "Data @ OpenStreetMap contributors, ODbL 1.0. http://
63
           www.openstreetmap.org/copyright",
        "osm_type": "node"
64
        "osm_id": "3029544258",
65
        "boundingbox": [
66
          " 52.4558836 " ,
67
          "\,52.4558836\,"
          " 13.2971957 "
69
          "13.2971957"
70
71
        "lat": "52.4558836",
72
        "lon": "13.2971957".
73
        "display_name": "9, Takustraße, Dahlem, Steglitz-Zehlendorf,
74
           Berlin, 14195, Germany",
        "class": "place",
"type": "house",
75
76
        "importance": 0.401,
77
        "address": {
78
          "house_number": "9"
79
          "road": "Takustraße",
80
          "suburb": "Dahlem"
81
          "city_district": "Steglitz-Zehlendorf",
82
          "state": "Berlin"
83
          " postcode " : "14195 "
84
          "country": "Germany"
85
          "country_code": "de"
86
        }
87
     },
88
89
        "place_id": "11080329",
        "licence": "Data © OpenStreetMap contributors, ODbL 1.0. http://
91
           www.openstreetmap.org/copyright",
        "osm_type": "node",
92
        "osm_id": "1049539115",
93
        "boundingbox": [
94
          " 52.4560765 " ,
95
          "52.4560765",
96
```

```
"13.2974781",
97
            " 13.2974781 "
98
         ],
"lat": "52.4560765",
"12.2974781",
99
100
101
          "display_name": "spline, 9, Takustraße, Dahlem, Steglitz-
102
              Zehlendorf, Berlin, 14195, Germany",
          "class": "leisure",
"type": "hackerspace",
103
104
          "importance": 0.401,
105
          "address": {
106
             "address29": "spline",
107
            "house_number": "9",
108
            "road": "Takustraße",
109
            "suburb": "Dahlem",
110
            "city_district": "Steglitz-Zehlendorf", "state": "Berlin",
111
112
            "postcode": "14195",
113
            "country": "Germany",
"country_code": "de"
114
115
          }
116
117
118
```

Listing A.5: Takustraße 9 - OSM API Result

ANHANG B

Testergebnisse und Auswertung

Dieser Abschnitt enthält zusätzliche Tabellen, welche die Szenarien aus Abschnitt 5.4 auswerten.

Die Tabellen des folgenden Abschnitts enthalten weitere Information zu sämtlichen durchgeführten Tests. In den Auswertungstabellen (z.B. Tabelle B.1) ist eine Übersicht für die einzelnen Geodienste zu finden, welche aufzeigt, wie viele Tests erfolgreich und wie viele Tests fehlerhaft verliefen. Zudem zeigen sie auf, ob der Fehler verursacht wurde, da für den Testfall kein Ergebnis geliefert wurde (Spalte Kein Resultat) oder ob der Fehler beim Abgleichen mit den Soll-Daten aufgetreten ist (Spalte Abgleichfehler). Zudem ist die Fehlerrate für den Test ersichtlich.

Auf jede Auswertungstabelle folgt eine Fehlergrundtabelle (z.B. Tabelle B.2), die beschreibt, wieso der Abgleich mit den Soll-Daten fehlschlug. Ein Test kann entweder wegen einem einzelnen Adresselement fehlschlagen (z.B. Postleitzahl, Straße <u>oder</u> Hausnummer) oder aber aus mehreren Gründen (z.B. Postleitzahl <u>und</u> Straße). Die Tabelle listet auf, wie hoch der Fehleranteil der einzelnen Adresselemente oder aber mehrerer Adresselemente in Kombination ist (Spalte *Mehrf.*).

Ein Testfall kann mehrere unterschiedliche Fehlergründe haben. Die Verteilungen der Fehlergründe müssen daher nicht proportional zu den Fehlern sein. Wenn für einen Testfall beispielsweise fünf Ergebnisse geliefert wurden, jedoch alle fünf Ergebnisse nicht den Soll-Daten entsprechen, fließen diese fehlerhaften Ergebnisse einzeln in die Fehlergrundtabelle ein. Somit kann zum Beispiel ein Fehler für den Ort, ein Fehler für die Postleitzahl, ein Fehler für die Straße und zwei Fehler aufgrund mehrfache falscher Elemente eingetragen werden, obwohl es sich dabei lediglich um einen Testfall handelt.

B.1 Szenario 1: Politische und geografische Verteilung: Deutschland

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate
Bing Maps	409	31	3	28	$7{,}05\%$
Google Maps	375	65	12	53	$14{,}77\%$
Here	420	20	1	19	$4{,}55\%$
OSM	171	269	40	229	$61{,}14\%$

Tabelle B.1.: Auswertung: Szenario 1: Zufällige Verteilung: Straße Hausnr. Ort

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00 %	0,00 %	$7,\!89\%$	0,00 %	$18{,}42\%$	$21{,}05\%$	$18{,}42\%$	$\overline{34,21\%}$
Google Maps	$0{,}00\%$	0,00%	$2{,}70\%$	$0,\!00\%$	$14{,}86\%$	$22{,}97\%$	$13{,}51\%$	$45{,}95\%$
Here	$0{,}00\%$	$0,\!00\%$	$0,\!00\%$	$0{,}00\%$	$47{,}37\%$	$5{,}26\%$	$47{,}37\%$	$0,\!00\%$
OSM	$0,\!00\%$	$0,\!00\%$	$0,\!00\%$	$0{,}00\%$	$3{,}39\%$	$1{,}41\%$	$66{,}67\%$	$28{,}53\%$

Tabelle B.2.: Fehlergründe: Szenario 1: Zufällige Verteilung: Straße Hausnr. Ort

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate
Bing Maps	400	40	2	38	$9{,}09\%$
Google Maps	364	76	5	71	$17{,}27\%$
Here	422	18	1	17	$4{,}09\%$
OSM	105	335	45	290	$76{,}14\%$

Tabelle B.3.: Auswertung: Szenario 1: Gleichmäßige Verteilung: Straße Hausnr. Ort

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00 %	0,00 %	$18,\!64\%$	0,00 %	$13,\!56\%$	$10,\!17\%$	$20,\!34\%$	$\overline{37,29}\%$
Google Maps	$0{,}00\%$	0,00%	$10{,}00\%$	$0,\!00\%$	$15{,}00\%$	$20,\!00\%$	$16{,}25\%$	$38{,}75\%$
Here	$0{,}00\%$	$0,\!00\%$	0,00%	$0,\!00\%$	$54{,}55\%$	$4{,}55\%$	$40{,}91\%$	$0,\!00\%$
OSM	$0{,}00\%$	$0,\!00\%$	$0{,}00\%$	$0,\!00\%$	$1{,}50\%$	$3{,}21\%$	$61{,}54\%$	$33{,}76\%$

Tabelle B.4.: Fehlergründe: Szenario 1: Gleichmäßige Verteilung: Straße Hausnr. Ort

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate
Bing Maps	410	30	1	29	$6,\!82\%$
Google Maps	377	63	0	63	$14{,}32\%$
Here	422	18	0	18	$4{,}09\%$
OSM	169	271	55	216	$61{,}59\%$

Tabelle B.5.: Auswertung: Szenario 1: Zufällige Verteilung: Straße Hausnr. Ort PLZ

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00%	0,00%	$7{,}50\%$	$0{,}00\%$	$15{,}00\%$	$22{,}50\%$	$22{,}50\%$	$32,\!50\%$
Google Maps	0,00%	$1{,}37\%$	$2{,}74\%$	$0,\!00\%$	$6,\!85\%$	$27{,}40\%$	$15{,}07\%$	$46{,}58\%$
Here	$0{,}00\%$	$0,\!00\%$	$0,\!00\%$	0,00%	$38{,}89\%$	$5{,}56\%$	$50{,}00\%$	$5{,}56\%$
OSM	$0{,}00\%$	$0{,}00\%$	$0{,}00\%$	$0{,}00\%$	$1{,}43\%$	$1{,}79\%$	$85{,}00\%$	$11{,}79\%$

Tabelle B.6.: Fehlergründe: Szenario 1: Zufällige Verteilung: Straße Hausnr. Ort PLZ

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate
Bing Maps	401	39	1	38	$8,\!86\%$
Google Maps	365	75	2	73	$17{,}05\%$
Here	422	18	0	18	$4{,}09\%$
OSM	105	335	89	246	$76{,}14\%$

Tabelle B.7.: Auswertung: Szenario 1: Gleichmäßige Verteilung: Straße Hausnr. Ort PLZ

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00 %	0,00 %	$21{,}82\%$	0,00 %	$10{,}91\%$	$12{,}73\%$	$21{,}82\%$	32,73%
Google Maps	$0{,}00\%$	$0,\!00\%$	$12{,}50\%$	$0,\!00\%$	$7{,}50\%$	$21{,}25\%$	$20{,}00\%$	$38{,}75\%$
Here	$0{,}00\%$	$0,\!00\%$	$4{,}35\%$	$0,\!00\%$	$52{,}17\%$	$4{,}35\%$	$39{,}13\%$	$0,\!00\%$
OSM	$0,\!00\%$	$0,\!00\%$	$0{,}00\%$	$0{,}00\%$	$0{,}58\%$	$3{,}80\%$	$87{,}43\%$	$8{,}19\%$

Tabelle B.8.: Fehlergründe: Szenario 1: Gleichmäßige Verteilung: Straße Hausnr. Ort PLZ

B.2 Szenario 2: Politische und geografische Verteilung: Alaska

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate
Bing Maps	94	152	77	75	$61{,}79\%$
Google Maps	140	106	8	98	$43{,}09\%$
Here	144	102	29	73	$41{,}46\%$
OSM	74	172	112	60	$69{,}92\%$

Tabelle B.9.: Auswertung: Szenario 2: Zufällige Verteilung: Straße Hausnr. Ort

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00 %	0,00 %	0,00 %	0,00 %	$6,\!47\%$	$9{,}35\%$	$4{,}32\%$	79,86%
Google Maps	$0{,}00\%$	0,00%	$1{,}96\%$	$0,\!00\%$	$24{,}51\%$	$26{,}47\%$	$8{,}82\%$	$38{,}24\%$
Here	$0{,}00\%$	$0,\!00\%$	$2{,}11\%$	$0,\!00\%$	$33{,}68\%$	$24{,}21\%$	$11{,}58\%$	$28{,}42\%$
OSM	$0,\!00\%$	$0,\!00\%$	$0,\!00\%$	$0{,}00\%$	$31{,}53\%$	$3{,}60\%$	$18{,}02\%$	$46{,}85\%$

Tabelle B.10.: Fehlergründe: Szenario 2: Zufällige Verteilung: Straße Hausnr. Ort

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate
Bing Maps	14	232	130	102	$94,\!31\%$
Google Maps	90	156	20	136	$63{,}41\%$
Here	98	148	62	86	$60{,}16\%$
OSM	25	221	128	93	$89{,}84\%$

Tabelle B.11.: Auswertung: Szenario 2: Gleichmäßige Verteilung: Straße Hausnr. Ort

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00 %	0,00 %	0,00 %	0,00 %	0,45 %	2,23%	0,89 %	96,43%
Google Maps	$0{,}00\%$	0,00%	$2{,}72\%$	$0,\!00\%$	$26{,}53\%$	$12{,}24\%$	$6{,}12\%$	$52,\!38\%$
Here	$0{,}00\%$	$0,\!00\%$	$8{,}70\%$	$0{,}00\%$	$20,\!65\%$	$22{,}83\%$	$14{,}13\%$	$33{,}70\%$
OSM	$0,\!00\%$	$0{,}70\%$	$0,\!00\%$	$0,\!00\%$	$37{,}06\%$	$11{,}19\%$	$2{,}80\%$	$48,\!25\%$

Tabelle B.12.: Fehlergründe: Szenario 2: Gleichmäßige Verteilung: Straße Hausnr. Ort

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate	
Bing Maps	135	111	1	110	$45{,}12\%$	
Google Maps	146	100	0	100	$40{,}65\%$	
Here	161	85	4	81	$34{,}55\%$	
OSM	66	180	156	24	$73{,}17\%$	

Tabelle B.13.: Auswertung: Szenario 2: Zufällige Verteilung: Straße Hausnr. Ort PLZ

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00%	0,00%	$0,\!00\%$	$0,\!00\%$	$11{,}02\%$	$22{,}83\%$	$11,\!81\%$	54,33%
Google Maps	$0{,}00\%$	0,00%	$1{,}94\%$	$0,\!00\%$	$18{,}45\%$	$35{,}92\%$	$10{,}68\%$	$33{,}01\%$
Here	$0{,}00\%$	$0,\!00\%$	$2{,}22\%$	0,00%	$27{,}78\%$	$40,\!00\%$	$16{,}67\%$	$13{,}33\%$
OSM	$0{,}00\%$	$0{,}00\%$	$0{,}00\%$	$0{,}00\%$	$0{,}00\%$	$10{,}34\%$	$72{,}41\%$	$17{,}24\%$

Tabelle B.14.: Fehlergründe: Szenario 2: Zufällige Verteilung: Straße Hausnr. Ort PLZ

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate
Bing Maps	88	158	20	138	$64{,}23\%$
Google Maps	95	151	1	150	$61{,}38\%$
Here	103	143	13	130	$58{,}13\%$
OSM	24	222	193	29	$90{,}24\%$

Tabelle B.15.: Auswertung: Szenario 2: Gleichmäßige Verteilung: Straße Hausnr. Ort PLZ

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00 %	0,00 %	$4{,}43\%$	0,00 %	$19{,}62\%$	$25{,}32\%$	$7{,}59\%$	43,04%
Google Maps	0,00%	$0,\!00\%$	$2{,}86\%$	$0,\!00\%$	$17{,}14\%$	$26{,}29\%$	$5{,}71\%$	$48{,}00\%$
Here	$0,\!00\%$	$0,\!00\%$	$15{,}33\%$	$0,\!00\%$	$13{,}87\%$	$45{,}99\%$	$10{,}95\%$	$13{,}87\%$
OSM	$0{,}00\%$	$2{,}63\%$	$0{,}00\%$	$0{,}00\%$	$34{,}21\%$	$36{,}84\%$	$10{,}53\%$	$15{,}79\%$

Tabelle B.16.: Fehlergründe: Szenario 2: Gleichmäßige Verteilung: Straße Hausnr. Ort PLZ

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate
Bing Maps	134	112	0	112	45,53%
Google Maps	139	107	5	102	$43{,}50\%$
Here	144	102	18	84	$41{,}46\%$
OSM	74	172	117	55	$69{,}92\%$

Tabelle B.17.: Auswertung: Szenario 2: Zufällige Verteilung: Straße Hausnr. Ort Bundesstaat

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	$0{,}00\%$	$0,\!00\%$	$0,\!00\%$	$0{,}00\%$	$32{,}26\%$	$14,\!84\%$	$7{,}10\%$	$45,\!81\%$
Google Maps	$0{,}00\%$	0,00%	$1{,}92\%$	$0,\!00\%$	$25{,}00\%$	$26{,}92\%$	$8{,}65\%$	$37{,}50\%$
Here	$0{,}00\%$	$0,\!00\%$	$1{,}90\%$	$0,\!00\%$	$30{,}48\%$	$27{,}62\%$	$11{,}43\%$	$28{,}57\%$
OSM	$0{,}00\%$	$0{,}00\%$	$0{,}00\%$	$0{,}00\%$	$34{,}65\%$	$2{,}97\%$	$19{,}80\%$	$42{,}57\%$

Tabelle B.18.: Fehlergründe: Szenario 2: Zufällige Verteilung: Straße Hausnr. Ort Bundesstaat

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate
Bing Maps	94	152	7	145	$61{,}79\%$
Google Maps	92	154	2	152	$62{,}60\%$
Here	98	148	20	128	$60{,}16\%$
OSM	24	222	144	78	$90{,}24\%$

Tabelle B.19.: Auswertung: Szenario 2: Gleichmäßige Verteilung: Straße Hausnr. Ort Bundesstaat

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00%	0,00 %	$5{,}26\%$	0,00%	$36{,}14\%$	$6{,}32\%$	$4{,}21\%$	48,07%
Google Maps	$0{,}00\%$	0,00%	$2{,}63\%$	$0,\!00\%$	$23{,}68\%$	$13{,}82\%$	$5{,}92\%$	$53{,}95\%$
Here	$0,\!00\%$	$0,\!00\%$	$6{,}47\%$	$0,\!00\%$	$13{,}67\%$	$41{,}73\%$	$10{,}07\%$	$28{,}06\%$
OSM	$0{,}00\%$	$0,\!00\%$	$0,\!00\%$	$0{,}00\%$	$50{,}96\%$	$14{,}42\%$	$3{,}85\%$	$30{,}77\%$

Tabelle B.20.: Fehlergründe: Szenario 2: Gleichmäßige Verteilung: Straße Hausnr. Ort Bundesstaat

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate	
Bing Maps	135	111	1	110	$45{,}12\%$	
Google Maps	146	100	0	100	$40{,}65\%$	
Here	161	85	4	81	$34{,}55\%$	
OSM	66	180	156	24	$73{,}17\%$	

Tabelle B.21.: Auswertung: Szenario 2: Zufällige Verteilung: Straße Hausnr. Ort Bundesstaat PLZ

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00%	$0,\!00\%$	$0,\!00\%$	$0,\!00\%$	$11{,}97\%$	$24{,}79\%$	$12{,}82\%$	$50,\!43\%$
Google Maps	$0{,}00\%$	0,00%	$1{,}82\%$	$0,\!00\%$	$18{,}18\%$	$36{,}36\%$	$10{,}00\%$	$33{,}64\%$
Here	$0{,}00\%$	$0,\!00\%$	$2{,}30\%$	0,00%	$21{,}84\%$	$42{,}53\%$	$17{,}24\%$	$16{,}09\%$
OSM	$0{,}00\%$	$0{,}00\%$	$0{,}00\%$	$0{,}00\%$	$0{,}00\%$	$10{,}71\%$	$71{,}43\%$	$17{,}86\%$

Tabelle B.22.: Fehlergründe: Szenario 2: Zufällige Verteilung: Straße Hausnr. Ort Bundesstaat PLZ

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate	
Bing Maps	96	150	7	143	$60{,}98\%$	
Google Maps	95	151	1	150	$61{,}38\%$	
Here	104	142	3	139	$57{,}72\%$	
OSM	23	223	195	28	$90{,}65\%$	

Tabelle B.23.: Auswertung: Szenario 2: Gleichmäßige Verteilung: Straße Hausn
r. Ort Bundesstaat PLZ

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00%	0,00%	$9{,}68\%$	$0,\!00\%$	$20,\!00\%$	$21{,}94\%$	$7{,}74\%$	40,65%
Google Maps	$0{,}00\%$	$0,\!00\%$	$3{,}05\%$	$0,\!00\%$	$18{,}29\%$	$31{,}71\%$	$6{,}10\%$	$40,\!85\%$
Here	$0{,}00\%$	$0,\!00\%$	$14{,}19\%$	$0,\!00\%$	$14{,}86\%$	$45{,}27\%$	$10{,}14\%$	$15{,}54\%$
OSM	$0,\!00\%$	$0,\!00\%$	$0{,}00\%$	$0,\!00\%$	$35{,}14\%$	$37{,}84\%$	$10,\!81\%$	$16{,}22\%$

Tabelle B.24.: Fehlergründe: Szenario 2: Gleichmäßige Verteilung: Straße Hausnr. Ort Bundesstaat PLZ

B.3 Szenario 3: Verteilung mit Einbeziehen der Testfalldichte: Alaska

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate
Bing Maps	58	230	110	120	$79{,}86\%$
Google Maps	137	151	17	134	$52{,}43\%$
Here	160	128	63	65	$44{,}44\%$
OSM	50	238	165	73	$82{,}64\%$

Tabelle B.25.: Auswertung: Szenario 3: Dichteverteilung: Straße Hausnr. Ort

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00 %	0,00 %	0,00 %	0,00 %	$0,\!37\%$	3,00%	$0,\!37\%$	96,25%
Google Maps	$0{,}00\%$	0,00%	$5{,}00\%$	$0,\!00\%$	$24{,}29\%$	$17{,}86\%$	$7{,}86\%$	$45{,}00\%$
Here	$0{,}00\%$	$0{,}00\%$	$11{,}43\%$	$0,\!00\%$	$18{,}57\%$	$30{,}00\%$	$7{,}14\%$	$32{,}86\%$
OSM	$0{,}00\%$	$0,\!00\%$	$0{,}00\%$	$0,\!00\%$	$42{,}98\%$	$1{,}75\%$	$5{,}26\%$	$50{,}00\%$

Tabelle B.26.: Fehlergründe: Szenario 3: Dichteverteilung: Straße Hausnr. Ort

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate
Bing Maps	140	148	33	115	$51{,}39\%$
Google Maps	138	150	0	150	$52{,}08\%$
Here	183	105	18	87	$36{,}46\%$
OSM	48	240	213	27	$83{,}33\%$

Tabelle B.27.: Auswertung: Szenario 3: Dichteverteilung: Straße Hausnr. Ort PLZ

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00 %	0,00 %	2,03 %	0,00 %	12,84 %	$26,\!35\%$	2,70 %	$\overline{56,08}\%$
Google Maps	$0{,}00\%$	0,00%	$4{,}73\%$	$0,\!00\%$	$17{,}75\%$	$28{,}40\%$	$7{,}69\%$	$41{,}42\%$
Here	$0{,}00\%$	$0,\!00\%$	$10{,}23\%$	$0,\!00\%$	$13{,}64\%$	$52{,}27\%$	$7{,}95\%$	$15{,}91\%$
OSM	$0,\!00\%$	$0{,}00\%$	$0{,}00\%$	$0,\!00\%$	$36{,}11\%$	$2{,}78\%$	$22{,}22\%$	38,89%

Tabelle B.28.: Fehlergründe: Szenario 3: Dichteverteilung: Straße Hausnr. Ort PLZ

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate
Bing Maps	139	149	31	118	51,74%
Google Maps	135	153	7	146	$53{,}12\%$
Here	162	126	35	91	$43{,}75\%$
OSM	50	238	170	68	$82{,}64\%$

Tabelle B.29.: Auswertung: Szenario 3: Dichteverteilung: Straße Hausnr. Ort Bundesstaat

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00%	$0,\!00\%$	$2{,}35\%$	$0,\!00\%$	$23{,}53\%$	$9{,}41\%$	$2{,}35\%$	$62,\!35\%$
Google Maps	$0{,}00\%$	0,00%	$5{,}48\%$	$0{,}00\%$	$23{,}29\%$	$19{,}18\%$	$7{,}53\%$	$44{,}52\%$
Here	$0{,}00\%$	$0,\!00\%$	$8{,}42\%$	0,00%	$13{,}68\%$	$37{,}89\%$	$5{,}26\%$	$34{,}74\%$
OSM	$0{,}00\%$	$0{,}00\%$	$0,\!00\%$	$0{,}00\%$	$49{,}48\%$	$1{,}03\%$	$6{,}19\%$	$43{,}30\%$

Tabelle B.30.: Fehlergründe: Szenario 3: Dichteverteilung: Straße Hausnr. Ort Bundesstaat

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate	
Bing Maps	145	143	24	119	$49{,}65\%$	
Google Maps	139	149	0	149	$51{,}74\%$	
Here	183	105	6	99	$36{,}46\%$	
OSM	48	240	216	24	$83{,}33\%$	

Tabelle B.31.: Auswertung: Szenario 3: Dichteverteilung: Straße Hausnr. Ort Bundesstaat PLZ

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00%	0,00 %	$3,\!20\%$	0,00 %	$16,\!80\%$	$28,\!00\%$	$3{,}20\%$	48,80 %
Google Maps	0,00%	$0,\!00\%$	$5{,}03\%$	$0,\!00\%$	$18,\!87\%$	$32{,}70\%$	$8{,}18\%$	$35{,}22\%$
Here	$0,\!00\%$	$0,\!00\%$	$8{,}91\%$	0,00%	$13{,}86\%$	$54{,}46\%$	$6{,}93\%$	$15{,}84\%$
OSM	$0{,}00\%$	$0,\!00\%$	$0,\!00\%$	$0{,}00\%$	$41{,}94\%$	$3{,}23\%$	$25{,}81\%$	$29{,}03\%$

Tabelle B.32.: Fehlergründe: Szenario 3: Dichteverteilung: Straße Hausnr. Ort Bundesstaat PLZ

B.4 Szenario 4: Äquivalenzklassentest: Berlin

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate
Bing Maps	52	58	0	58	$52{,}73\%$
Google Maps	56	54	0	54	$49{,}09\%$
Here	83	27	1	26	$24{,}55\%$
OSM	69	41	13	28	$37{,}27\%$

Tabelle B.33.: Auswertung: Szenario 4: Straße Ort

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00 %	0,00 %	0,00 %	0,00 %	$45,\!83\%$	6,94%	0,00 %	47,22%
Google Maps	$0{,}00\%$	0,00%	$0,\!00\%$	$0,\!00\%$	$13{,}76\%$	$7{,}34\%$	$0{,}00\%$	$78{,}90\%$
Here	$0{,}00\%$	$0,\!00\%$	$0,\!00\%$	$0,\!00\%$	$76,\!67\%$	$3{,}33\%$	0,00%	$20,\!00\%$
OSM	$0,\!00\%$	$0,\!00\%$	$0,\!00\%$	$0{,}00\%$	$12{,}04\%$	$17{,}59\%$	$0{,}00\%$	$70{,}37\%$

Tabelle B.34.: Fehlergründe: Szenario 4: Straße Ort

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate
Bing Maps	78	32	0	32	$29{,}09\%$
Google Maps	70	40	0	40	$36{,}36\%$
Here	102	8	0	8	$7{,}27\%$
OSM	69	41	19	22	$37{,}27\%$

Tabelle B.35.: Auswertung: Szenario 4: Straße Ort PLZ

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00 %	0,00 %	0,00 %	0,00 %	$19,\!15\%$	$36,\!17\%$	0,00 %	$\overline{44,68\%}$
Google Maps	$0{,}00\%$	$1{,}45\%$	0,00%	$0,\!00\%$	$2{,}90\%$	$36{,}23\%$	$0{,}00\%$	$59{,}42\%$
Here	$0{,}00\%$	$0,\!00\%$	$0,\!00\%$	$0,\!00\%$	$0,\!00\%$	$75{,}00\%$	0,00%	$25{,}00\%$
OSM	$0,\!00\%$	$0,\!00\%$	$0,\!00\%$	$0,\!00\%$	$22{,}58\%$	$38{,}71\%$	$0,\!00\%$	38,71%

Tabelle B.36.: Fehlergründe: Szenario 4: Straße Ort PLZ

B.5 Szenario 5: Mehrfach vorkommende Straßennamen: Berlin

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate
Bing Maps	72	180	0	180	$71{,}43\%$
Google Maps	55	197	0	197	$78{,}17\%$
Here	219	33	0	33	$13{,}10\%$
OSM	164	88	2	86	$34{,}92\%$

Tabelle B.37.: Auswertung: Szenario 5: Straße Ort

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00%	0,00%	0,00%	0,00%	$80,\!36\%$	$2{,}14\%$	0,00%	17,50%
Google Maps	$0{,}00\%$	$0,\!00\%$	$0,\!00\%$	$0,\!00\%$	$82{,}61\%$	$0,\!00\%$	0,00%	$17{,}39\%$
Here	$0{,}00\%$	$0,\!00\%$	$0,\!00\%$	0,00%	$100,\!00\%$	$0,\!00\%$	$0,\!00\%$	0,00%
OSM	$0{,}00\%$	$0{,}00\%$	$0,\!00\%$	$0{,}00\%$	$63{,}61\%$	$3{,}58\%$	$0{,}00\%$	$32{,}81\%$

Tabelle B.38.: Fehlergründe: Szenario 5: Straße Ort

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate
Bing Maps	210	42	0	42	$16{,}67\%$
Google Maps	150	102	0	102	$40{,}48\%$
Here	250	2	0	2	$0{,}79\%$
OSM	173	79	2	77	$31{,}35\%$

Tabelle B.39.: Auswertung: Szenario 5: Straße Ort PLZ

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00 %	0,00 %	0,00 %	0,00 %	32,00 %	54,00 %	0,00 %	$\overline{14,00}\%$
Google Maps	$0{,}00\%$	$0{,}97\%$	$0,\!00\%$	$0,\!00\%$	$66{,}02\%$	$6{,}80\%$	$0,\!00\%$	$26{,}21\%$
Here	$0{,}00\%$	$0,\!00\%$	$0,\!00\%$	0,00%	$100,\!00\%$	$0{,}00\%$	$0,\!00\%$	0,00%
OSM	$0{,}00\%$	$0,\!00\%$	$0,\!00\%$	$0,\!00\%$	$62,\!08\%$	$6{,}35\%$	$0,\!00\%$	31,57%

Tabelle B.40.: Fehlergründe: Szenario 5: Straße Ort PLZ

B.6 Szenario 6: Sprachen: diakritische Zeichen: Prag

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate
Bing Maps	141	15	0	15	$9{,}62\%$
Google Maps	112	44	0	44	$28{,}21\%$
Here	137	19	0	19	$12{,}18\%$
OSM	102	54	15	39	$34{,}62\%$

Tabelle B.41.: Auswertung: Szenario 6: Tschechisch: Straße Ort

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00%	$0,\!00\%$	0,00%	0,00%	$0,\!00\%$	$9{,}52\%$	$0{,}00\%$	$90,\!48\%$
Google Maps	$0{,}00\%$	$17{,}24\%$	0,00%	$0,\!00\%$	$25{,}86\%$	$10{,}34\%$	$0,\!00\%$	$46{,}55\%$
Here	$0{,}00\%$	$0{,}00\%$	0,00%	$0,\!00\%$	$47{,}83\%$	$17{,}39\%$	$0,\!00\%$	$34{,}78\%$
OSM	$0{,}00\%$	$0{,}00\%$	$0{,}00\%$	$0,\!00\%$	$28,\!68\%$	$5{,}43\%$	$0{,}00\%$	$65{,}89\%$

Tabelle B.42.: Fehlergründe: Szenario 6: Tschechisch: Straße Ort

B.7 Szenario 7: Sprachen: kyrillisch: Moskau

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate
Bing Maps	47	15	1	14	$24,\!19\%$
Google Maps	0	62	0	62	$100,\!00\%$
Here	36	26	0	26	$41{,}94\%$
OSM	0	62	26	36	$100,\!00\%$

Tabelle B.43.: Auswertung: Szenario 7: Russisch: Straße Ort

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00%	0,00%	0,00%	0,00%	$60,\!00\%$	0,00%	$0,\!00\%$	40,00%
Google Maps	$0{,}00\%$	$27{,}42\%$	0,00%	$0,\!00\%$	$0,\!00\%$	$0{,}00\%$	$0,\!00\%$	$72{,}58\%$
Here	$0{,}00\%$	0,00%	0,00%	$0,\!00\%$	$74{,}29\%$	$11{,}43\%$	$0,\!00\%$	$14{,}29\%$
OSM	$0{,}00\%$	$5{,}93\%$	$0{,}00\%$	$0{,}00\%$	$0{,}00\%$	$0{,}00\%$	$0{,}00\%$	$94{,}07\%$

Tabelle B.44.: Fehlergründe: Szenario 7: Russisch: Straße Ort

B.8 Szenario 8: Sprachen: Griechisch: Athen

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate
Bing Maps	8	35	14	21	$81,\!40\%$
Google Maps	0	43	0	43	$100,\!00\%$
Here	6	37	11	26	$86{,}05\%$
OSM	0	43	31	12	$100,\!00\%$

Tabelle B.45.: Auswertung: Szenario 8: Griechisch: Straße Ort

	GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bin	g Maps	0,00%	$0,\!00\%$	0,00%	0,00%	$16{,}67\%$	0,00%	0,00%	83,33%
Goog	le Maps	$0{,}00\%$	$11{,}63\%$	$0,\!00\%$	$0,\!00\%$	$0,\!00\%$	$2{,}33\%$	0,00%	$86,\!05\%$
	Here	$0{,}00\%$	$0{,}00\%$	0,00%	$0,\!00\%$	$28{,}57\%$	$0,\!00\%$	$0{,}00\%$	$71{,}43\%$
	OSM	$0{,}00\%$	$0{,}00\%$	$0{,}00\%$	$0{,}00\%$	$3{,}28\%$	$0,\!00\%$	$0{,}00\%$	$96{,}72\%$

Tabelle B.46.: Fehlergründe: Szenario 8: Griechisch: Straße Ort

B.9 Szenario 9: Politische und geografische Verteilung: Australien

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate	
Bing Maps	923	1115	578	537	$54{,}71\%$	
Google Maps	1413	625	145	480	$30{,}67\%$	
Here	1765	273	29	244	$13{,}40\%$	
OSM	45	1993	857	1136	$97{,}79\%$	

Tabelle B.47.: Auswertung: Szenario 9: Zufällige Verteilung: Hausnr. Straße Ort

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00%	$0,\!00\%$	$4{,}24\%$	$0,\!00\%$	$8{,}31\%$	0,09%	$0,\!87\%$	86,49%
Google Maps	$0{,}00\%$	$0,\!00\%$	$6{,}48\%$	$0,\!00\%$	$20{,}28\%$	$6{,}20\%$	$10{,}28\%$	$56{,}76\%$
Here	$0{,}00\%$	$0,\!00\%$	$0,\!00\%$	0,00%	$72{,}94\%$	$2{,}35\%$	$11{,}47\%$	$13{,}24\%$
OSM	$0{,}00\%$	$0,\!04\%$	$0,\!00\%$	$0{,}00\%$	$9{,}02\%$	$0{,}09\%$	$24{,}47\%$	$66{,}37\%$

Tabelle B.48.: Fehlergründe: Szenario 9: Zufällige Verteilung: Hausnr. Straße Ort

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate
Bing Maps	57	1981	1428	553	$97{,}20\%$
Google Maps	748	1290	590	700	$63{,}30\%$
Here	1670	368	35	333	$18{,}06\%$
OSM	39	1999	1519	480	$98{,}09\%$

Tabelle B.49.: Auswertung: Szenario 9: Gleichmäßige Verteilung: Hausnr. Straße Ort

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00%	$0,\!00\%$	$0,\!30\%$	$0{,}00\%$	$0{,}22\%$	$0,\!07\%$	$0{,}22\%$	99,18%
Google Maps	$0{,}00\%$	$0{,}10\%$	$0,\!68\%$	0,00%	$11{,}95\%$	$13{,}61\%$	$13{,}12\%$	$60{,}54\%$
Here	$0{,}00\%$	$0{,}00\%$	$0{,}00\%$	0,00%	$65{,}43\%$	$4{,}57\%$	$16{,}00\%$	$14{,}00\%$
OSM	$0{,}00\%$	$0{,}11\%$	$0,\!00\%$	$0{,}00\%$	$18{,}35\%$	$0{,}55\%$	$8{,}46\%$	$72{,}53\%$

Tabelle B.50.: Fehlergründe: Szenario 9: Gleichmäßige Verteilung: Hausnr. Straße Ort

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate
Bing Maps	1242	796	436	360	$39{,}06\%$
Google Maps	1343	695	0	695	$34{,}10\%$
Here	1835	203	3	200	$9{,}96\%$
OSM	42	1996	1287	709	$97{,}94\%$

Tabelle B.51.: Auswertung: Szenario 9: Zufällige Verteilung: Hausnr. Straße Ort PLZ

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00%	0,00%	$10{,}02\%$	$0,\!00\%$	$12{,}14\%$	$1{,}16\%$	$7{,}90\%$	68,79%
Google Maps	$0{,}00\%$	0,00%	$6{,}27\%$	$0,\!00\%$	$10{,}74\%$	$21{,}10\%$	$26{,}60\%$	$35{,}29\%$
Here	$0{,}00\%$	$0,\!00\%$	$0{,}49\%$	$0{,}00\%$	$56{,}31\%$	$6{,}31\%$	$24{,}76\%$	$12{,}14\%$
OSM	$0{,}00\%$	$0{,}10\%$	$0{,}00\%$	$0,\!00\%$	$18{,}40\%$	$0{,}20\%$	$55{,}11\%$	$26{,}19\%$

Tabelle B.52.: Fehlergründe: Szenario 9: Gleichmäßige Verteilung: Hausnr. Straße Ort PLZ

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate
Bing Maps	499	1539	1239	300	$75{,}52\%$
Google Maps	728	1310	30	1280	$64{,}28\%$
Here	1707	331	0	331	$16{,}24\%$
OSM	40	1998	1806	192	$98{,}04\%$

Tabelle B.53.: Auswertung: Szenario 9: Gleichmäßige Verteilung: Hausnr. Straße Ort PLZ

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00%	0,00 %	0,95%	0,00 %	$6{,}17\%$	1,58%	$7{,}91\%$	83,39%
Google Maps	$0{,}00\%$	$0{,}07\%$	$0,\!51\%$	$0,\!00\%$	$8{,}08\%$	$49{,}09\%$	$12{,}16\%$	$30{,}08\%$
Here	$0{,}00\%$	$0,\!00\%$	$1{,}75\%$	$0,\!00\%$	$62,\!87\%$	$8{,}19\%$	$19{,}30\%$	$7{,}89\%$
OSM	$0{,}00\%$	$0{,}32\%$	$0,\!00\%$	$0{,}00\%$	$40{,}97\%$	$0{,}97\%$	$18{,}71\%$	$39{,}03\%$

Tabelle B.54.: Fehlergründe: Szenario 9: Gleichmäßige Verteilung: Hausnr. Straße Ort PLZ

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate	
Bing Maps	1662	376	6	370	$18{,}45\%$	
Google Maps	1472	566	65	501	$27{,}77\%$	
Here	1835	203	2	201	$9{,}96\%$	
OSM	42	1996	1299	697	$97{,}94\%$	

Tabelle B.55.: Auswertung: Szenario 9: Zufällige Verteilung: Hausnr. Straße Ort PLZ Land

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00%	$0,\!00\%$	$12{,}82\%$	$0,\!00\%$	$22{,}38\%$	$4{,}66\%$	$13{,}75\%$	46,39%
Google Maps	$0{,}00\%$	0,00%	$7{,}82\%$	$0,\!00\%$	$12{,}99\%$	$19{,}56\%$	$15{,}96\%$	$43{,}66\%$
Here	$0{,}00\%$	$0,\!00\%$	$0,\!48\%$	$0,\!00\%$	$55{,}98\%$	$6{,}22\%$	$24{,}40\%$	$12{,}92\%$
OSM	$0{,}00\%$	$0{,}10\%$	$0{,}00\%$	$0,\!00\%$	$18{,}38\%$	$0{,}21\%$	$55{,}54\%$	$25{,}77\%$

Tabelle B.56.: Fehlergründe: Szenario 9: Zufällige Verteilung: Hausnr. Straße Ort PLZ Land

GIS	Erfolg	Fehler	Kein Resultat	Abgleichfehler	Fehlerrate	
Bing Maps	1562	476	40	436	$23{,}36\%$	
Google Maps	777	1261	216	1045	$61,\!87\%$	
Here	1705	333	0	333	$16{,}34\%$	
OSM	39	1999	1845	154	$98{,}09\%$	

Tabelle B.57.: Auswertung: Szenario 9: Gleichmäßige Verteilung: Hausnr. Straße Ort PLZ Land

GIS	Land	Staat	Ort	Bezirk	PLZ	Straße	Hausnr.	Mehrf.
Bing Maps	0,00%	0,00 %	$3{,}52\%$	0,00 %	$33{,}54\%$	$12{,}22\%$	$25{,}88\%$	24,84%
Google Maps	$0{,}00\%$	$0,\!08\%$	$0,\!66\%$	$0,\!00\%$	$7{,}20\%$	$39{,}87\%$	$12{,}24\%$	$39{,}95\%$
Here	$0{,}00\%$	$0,\!00\%$	$1{,}75\%$	0,00%	$62,\!68\%$	$8{,}16\%$	$19{,}24\%$	$8{,}16\%$
OSM	$0{,}00\%$	$0{,}53\%$	$0,\!00\%$	$0{,}00\%$	$53{,}72\%$	$0{,}00\%$	$30{,}85\%$	$14{,}89\%$

Tabelle B.58.: Fehlergründe: Szenario 9: Gleichmäßige Verteilung: Hausnr. Straße Ort PLZ Land

Abbildungsverzeichnis

2.1.	Qualitätsmodell nach ISO 9126-1 [4]	4
2.2.	Tiling Schema Level 0 - 180° pro Tile (Zylindrische Projektion)	11
2.3.	Tiling Schema Level 1 - 90° pro Tile (Zylindrische Projektion)	11
2.4.	Tiling Schema Level 2 - 45° pro Tile (Zylindrische Projektion)	12
2.5.	Tiling Schema Level 2 - 45° pro Tile (Orthografische Azimutalprojektion) .	12
2.6.	maximale Abstände von zwei Punkten in benachbarten Tiles in unterschiedlichen Breiten	13
2.7.	Maximalen Abstände zwischen Testfällen benachbarter Tiles	14
3.1.	Systemintegration: Update der Software	21
3.2.	Geodienst als Blackbox aus Nutzersicht	22
3.3.	Übersicht: NDS Building Blocks (entnommen aus [28])	27
4.1.	Testgenerator Überblick	32
4.2.	K-Means Algorithmus [34]	36
4.3.	Heatmap - Straßenverteilung Europa	40
4.4.	Unterschiedliche Verteilungen von Testfällen in Europa	41
4.5.	Unterschiedliche Verteilungen von Testfällen in Russland	42
4.6.	Zufällige und geografisch gleichmäßgige Verteilung in Deutschland	43
4.7.	Geografisch gleichmäßige Verteilung inklusive Berücksichtigung der Testfall-	
	dichte in den 440 Tiles (Level 9) in Deutschland (mit eingezeichneten Tiles,	
	1162 Testfälle)	45
5.1.	Prozess: Testdurchführung und Testauswertung	50
5.2.	Zufällige und geografisch gleichmäßgige Verteilung in Deutschland	52
5.3.	Zufällige und geografisch gleichmäßige Verteilung in Alaska	53
5.4.	Dichteverteilung in Alaska (288 Testfälle)	54
5.5.	Zufällige und geografisch gleichmäßige Verteilung in Australien	56
5.6.	Szenario 1: Übersicht - Bestandene Tests (100 % abzüglich Fehlerrate) $$	59
5.7.	Szenario 2: Übersicht - Bestandene Tests (100 % abzüglich Fehlerrate) $\ . \ . \ .$	62
5.8.	Szenario 3: Übersicht - Bestandene Tests (100 % abzüglich Fehlerrate) $$	65
5.9.	Szenario 4: Übersicht - Bestandene Tests (100 % abzüglich Fehlerrate)	66
5.10.	Szenario 5: Übersicht - Bestandene Tests (100 % abzüglich Fehlerrate)	68
	Szenario 6: Übersicht - Bestandene Tests (100 % abzüglich Fehlerrate) $\ . \ . \ .$	69
	Szenario 7: Übersicht - Bestandene Tests (100 % abzüglich Fehlerrate) $\ . \ . \ .$	70
5.13.	Szenario 8: Übersicht - Bestandene Tests (100 % abzüglich Fehlerrate)	71

Tabellenverzeichnis

5.14. Szenario 9: Übersicht - Bestandene Tests (100 % abzüglich Fehlerrate) . . . $\,$ 74

Tabellenverzeichnis

3.1.	Beispiel Testdaten	18
5.1. 5.2.	Szenario 1: Zufällige und gleichmäßige Verteilung: Straße Hausnr. Ort (SHO) Szenario 1: Zufällige und gleichmäßige Verteilung: Straße Hausnr. Ort Postleitzahl (SHOP)	58 58
5.3.	Szenario 2: Zufällige und gleichmäßige Verteilung: Hausnr. Straße Ort (HSO)	60
5.4.	Szenario 2: Zufällige und gleichmäßige Verteilung: Hausnr. Straße Ort Postleitzahl (HSOP)	60
5.5.	Szenario 2: Zufällige und gleichmäßige Verteilung: Hausnr. Straße Ort Bun-	
F C	desstaat (HSOB)	61
5.6.	Szenario 2: Zufällige und gleichmäßige Verteilung: Hausnr. Straße Ort Bunderstand Berthilder II (HCORD)	61
F 7	desstaat Postleitzahl (HSOBP)	
5.7.	Szenario 3: Dichteverteilung: Hausnr. Straße Ort (HSO)	63 63
5.8.	Szenario 3: Dichteverteilung: Hausnr. Straße Ort Postleitzahl (HSOP)	
5.9.	Szenario 3: Dichteverteilung: Hausnr. Straße Ort Bundesstaat (HSOB) Szenario 3: Dichteverteilung: Hausnr. Straße Ort Bundesstaat Postleitzahl	64
5.10.		64
5 11	(HSOBP)	65
	Szenario 4: Straße Ort (SO)	66
	Szenario 5: Straße Ort (SO)	67
5.14	Szenario 5: Straße Ort Postleitzahl (SOP)	67
	Szenario 6: Tschechisch: Straße Ort (SO)	68
	Szenario 7: Russisch: Straße Ort (SO)	69
	Auswertung: Szenario 8: Griechisch: Straße Ort (SO)	70
	Szenario 9: Zufällige und gleichmäßige Verteilung: Hausnr. Straße Ort (HSO)	71
	Szenario 9: Zufällige und gleichmäßige Verteilung: Hausnr. Straße Ort Post-	• -
	leitzahl (HSOP)	72
5.20.	Szenario 9: Zufällige und gleichmäßige Verteilung: Hausnr. Straße Ort Post-	
	leitzahl Land (HSOPL)	72
5.21.	Fehler- und Erfolgsraten über alle Tests	76
B.1.	Auswertung: Szenario 1: Zufällige Verteilung: Straße Hausnr. Ort	96
B.2.	Fehlergründe: Szenario 1: Zufällige Verteilung: Straße Hausnr. Ort	96
B.3.	Auswertung: Szenario 1: Gleichmäßige Verteilung: Straße Hausnr. Ort	96
B.4.	Fehlergründe: Szenario 1: Gleichmäßige Verteilung: Straße Hausnr. Ort	96

Tabellenverzeichnis 115

B.5. Auswertung: Szenario 1: Zufällige Verteilung: Straße Hausnr. Ort PLZ	97
B.6. Fehlergründe: Szenario 1: Zufällige Verteilung: Straße Hausnr. Ort PLZ	97
B.7. Auswertung: Szenario 1: Gleichmäßige Verteilung: Straße Hausnr. Ort PLZ	97
B.8. Fehlergründe: Szenario 1: Gleichmäßige Verteilung: Straße Hausnr. Ort PLZ	97
B.9. Auswertung: Szenario 2: Zufällige Verteilung: Straße Hausnr. Ort	98
B.10. Fehlergründe: Szenario 2: Zufällige Verteilung: Straße Hausnr. Ort	98
B.11. Auswertung: Szenario 2: Gleichmäßige Verteilung: Straße Hausnr. Ort	98
B.12. Fehlergründe: Szenario 2: Gleichmäßige Verteilung: Straße Hausnr. Ort	98
B.13. Auswertung: Szenario 2: Zufällige Verteilung: Straße Hausnr. Ort PLZ	99
B.14. Fehlergründe: Szenario 2: Zufällige Verteilung: Straße Hausnr. Ort PLZ	99
B.15. Auswertung: Szenario 2: Gleichmäßige Verteilung: Straße Hausnr. Ort PLZ	99
B.16. Fehlergründe: Szenario 2: Gleichmäßige Verteilung: Straße Hausnr. Ort PLZ	99
B.17. Auswertung: Szenario 2: Zufällige Verteilung: Straße Hausnr. Ort Bundesstaa	t 100
B.18. Fehlergründe: Szenario 2: Zufällige Verteilung: Straße Hausnr. Ort Bundesstaa	
B.19. Auswertung: Szenario 2: Gleichmäßige Verteilung: Straße Hausnr. Ort Bun-	
desstaat	100
B.20. Fehlergründe: Szenario 2: Gleichmäßige Verteilung: Straße Hausnr. Ort Bun-	
desstaat	100
B.21. Auswertung: Szenario 2: Zufällige Verteilung: Straße Hausnr. Ort Bundesstaat	
PLZ	101
B.22. Fehlergründe: Szenario 2: Zufällige Verteilung: Straße Hausnr. Ort Bundes-	
staat PLZ	101
B.23. Auswertung: Szenario 2: Gleichmäßige Verteilung: Straße Hausnr. Ort Bun-	
desstaat PLZ	101
B.24. Fehlergründe: Szenario 2: Gleichmäßige Verteilung: Straße Hausnr. Ort Bun-	
desstaat PLZ	101
B.25. Auswertung: Szenario 3: Dichteverteilung: Straße Hausnr. Ort	102
B.26. Fehlergründe: Szenario 3: Dichteverteilung: Straße Hausnr. Ort	102
B.27. Auswertung: Szenario 3: Dichteverteilung: Straße Hausnr. Ort PLZ	102
B.28. Fehlergründe: Szenario 3: Dichteverteilung: Straße Hausnr. Ort PLZ	102
B.29. Auswertung: Szenario 3: Dichteverteilung: Straße Hausnr. Ort Bundesstaat	103
B.30. Fehlergründe: Szenario 3: Dichteverteilung: Straße Hausnr. Ort Bundesstaat	103
B.31. Auswertung: Szenario 3: Dichteverteilung: Straße Hausnr. Ort Bundesstaat PL	Z103
B.32.Fehlergründe: Szenario 3: Dichteverteilung: Straße Hausnr. Ort Bundesstaat	
PLZ	103
B.33. Auswertung: Szenario 4: Straße Ort	104
B.34.Fehlergründe: Szenario 4: Straße Ort	104
B.35. Auswertung: Szenario 4: Straße Ort PLZ	104
B.36.Fehlergründe: Szenario 4: Straße Ort PLZ	104
B.37. Auswertung: Szenario 5: Straße Ort	105
B.38.Fehlergründe: Szenario 5: Straße Ort	105
B.39. Auswertung: Szenario 5: Straße Ort PLZ	105
B.40.Fehlergründe: Szenario 5: Straße Ort PLZ	105
B.41. Auswertung: Szenario 6: Tschechisch: Straße Ort	106
B.42.Fehlergründe: Szenario 6: Tschechisch: Straße Ort	106
B.43. Auswertung: Szenario 7: Russisch: Straße Ort	107
B.44. Fehlergründe: Szenario 7: Russisch: Straße Ort	107

B.45. Auswertung: Szenario 8: Griechisch: Straße Ort	108
B.46. Fehlergründe: Szenario 8: Griechisch: Straße Ort	108
B.47. Auswertung: Szenario 9: Zufällige Verteilung: Hausnr. Straße Ort	109
B.48. Fehlergründe: Szenario 9: Zufällige Verteilung: Hausnr. Straße Ort	109
${\bf B.49. Auswertung:}$ Szenario 9: Gleichmäßige Verteilung: Hausn r. Straße Ort $$	109
${\bf B}.50.{\bf Fehlergründe}:$ Szenario 9: Gleichmäßige Verteilung: Hausn r. Straße Ort $.$	109
${\rm B.51.Auswertung:}$ Szenario 9: Zufällige Verteilung: Hausn r. Straße Ort PLZ $\ .\ .\ .$	110
B.52.Fehlergründe: Szenario 9: Gleichmäßige Verteilung: Hausnr. Straße Ort PLZ	110
B.53. Auswertung: Szenario 9: Gleichmäßige Verteilung: Hausnr. Straße Ort PLZ	110
B.54. Fehlergründe: Szenario 9: Gleichmäßige Verteilung: Hausnr. Straße Ort PLZ	110
B.55. Auswertung: Szenario 9: Zufällige Verteilung: Hausnr. Straße Ort PLZ Land	111
B.56. Fehlergründe: Szenario 9: Zufällige Verteilung: Hausnr. Straße Ort PLZ Land	111
B.57. Auswertung: Szenario 9: Gleichmäßige Verteilung: Hausnr. Straße Ort PLZ	
Land	111
B.58.Fehlergründe: Szenario 9: Gleichmäßige Verteilung: Hausnr. Straße Ort PLZ	
Land	111

Quellcodeverzeichnis

3.1.	OSM-Export: Takustraße 9	25
A.1.	Takustraße 9 in OSM	85
A.2.	Takustraße 9 - Google Maps API Result	86
A.3.	Takustraße 9 - Here API Result	88
A.4.	Takustraße 9 - Bing Maps API Result	90
A.5.	Takustraße 9 - OSM API Result	92