

Lastenheft
Nachrichtenkommunikation für das THW

na17b

Inhaltsverzeichnis

| | | |
|----------|---|----------|
| 1 | Ausgangssituation | 2 |
| 2 | Zielsetzung und Produkteinsatz | 2 |
| 3 | Nichtfunktionale Anforderungen | 2 |
| 3.1 | Ausfüllbarkeit wie bei Papierformular | 2 |
| 3.2 | Listenanordnung für Vierfachvordrucke | 2 |
| 3.3 | Archivierung | 2 |
| 3.4 | Kommunikation zwischen Computern | 3 |
| 3.5 | Verschiedene Rollen | 3 |
| 3.6 | Skalierbarkeit | 3 |
| 4 | Funktionale Anforderungen | 3 |
| 4.1 | Visuelle Unterschiede für verschiedene Rollen | 3 |
| 4.2 | Vierfachvordruck renderbar als PDF | 3 |
| 4.3 | Speicherung im RDF-Format | 4 |
| 4.4 | Verschlüsselung | 4 |
| 4.5 | Datensicherheit | 4 |
| 4.6 | Zugriffsbeschränkung der Nutzer | 4 |
| 5 | Qualitätsmatrix | 5 |
| 6 | Lieferumfang und Abnahmekriterien | 5 |
| 7 | Vorprojekt | 6 |
| 8 | Glossar | 6 |
| 8.1 | RDF | 6 |
| 8.2 | SPARQL | 6 |
| 8.3 | Framework | 6 |
| 8.4 | Modultest | 6 |
| 8.5 | Deployment | 7 |
| 8.6 | Model-View-Controller | 7 |
| 8.7 | Java | 7 |
| 8.8 | Markdown | 7 |
| 8.9 | HTML | 7 |
| 8.10 | CSS | 7 |
| 8.11 | Skriptsprache | 7 |
| 8.12 | JavaScript | 8 |
| 8.13 | Continuous Integration | 8 |
| 8.14 | git | 8 |
| 8.15 | QuitStore | 9 |

1 Ausgangssituation

Die Grundlage für die Projekt bildet der sogenannte 4-Fach-Vordruck, ein internes Kommunikationsdokument des technischen Hilfswerks (THW). Es wird in der mobilen Einsatzzentrale des THW, der Fachgruppe Führung und Kommunikation (FGr FK) eingesetzt. Die FGr FK nutzt das Dokument, um ein- und ausgehende Nachrichten abzufassen und so z.B. Einsatzaufträge für Einheiten, eingehende Lagemeldungen oder Materialanforderungen abzuarbeiten. Der 4-Fach-Vordruck ist eine Papier-Vorlage mit dreifachem Durchschlag. Zwei der Durchschläge werden an die zuständigen Personen innerhalb der FGr FK verteilt; der dritte dient der Protokollierung. Im Angesicht heutiger Technologien ist dieses Verfahren nicht mehr zeitgemäß. Mithilfe einer Software wäre es möglich, diesen Prozess digital durchzuführen. Dadurch müssten keine handschriftlichen Dokumente verfasst und verteilt werden. Ein Vorteil wäre der Entfall von Mehraufwand durch schlecht lesbare Handschrift. Außerdem müssten die Zettel nicht mehr durch die Bearbeiter innerhalb der Zentrale verteilt werden, diese müssten somit nicht ständig ihre Posten verlassen. Weiterhin ließe sich die Verwaltung verschiedener Dokumente am jeweiligen Arbeitsplatz übersichtlicher gestalten, etwa durch ein digitales Postfach.

2 Zielsetzung und Produkteinsatz

Ziel des Projekts ist es, eine Anwendung zu schreiben, welche es erlaubt, einen Vierfach-Vordruck abzufassen, anzuzeigen und zu archivieren sowie den damit verbundenen Workflow zu durchlaufen, welcher lediglich durch die Software ergänzt aber nicht ersetzt werden soll. Zur maximalen Kompatibilität soll es sich um eine Browseranwendung handeln, welche ihre Daten in einer versionierten RDF-Datenbank auf Serverseite ablegt. Die Archivierung muss dabei die rechtliche Verbindlichkeit sicherstellen, d.h. nachträgliche Änderungen an den Dokumenten sollen erkennbar sein und verhindert werden. Die Software soll innerhalb der Fachgruppe Führung/Kommunikation eingesetzt werden um den Austausch von ein- und ausgehenden Nachrichten zu beschleunigen und zu vereinfachen.

3 Nichtfunktionale Anforderungen

3.1 Ausfüllbarkeit wie bei Papierformular

Der simulierte Vierfachvordruck sollte in Form und Formulierung dem Papierformular gleichen. Somit soll der gewohnte Arbeitsablauf zuverlässig gewährleistet werden.

3.2 Listenanordnung für Vierfachvordrucke

In dem Fall, dass zeitgleich mehr als ein Vierfachvordruck bei einem Mitarbeiter ankommt, sollte eine Liste zum Abrufen der eingegangenen Dokumente existieren. Somit kann jedes Dokument nacheinander bearbeitet werden, wodurch es auch zu keinem Informationsverlust kommen kann.

3.3 Archivierung

Es ist von immenser Bedeutung für das Technische Hilfswerk, nach einiger Zeit wieder auf die Angaben eines Vierfachvordrucks zugreifen zu können. Momentan wird dies ermöglicht, indem eine Kopie des Zettels im Archiv geordnet abgelegt wird. Da wir das manuelle Ausfüllen eines Vierfachvordrucks mit unserer Applikation abschaffen wollen, müssen wir dafür sorgen, dass die Archivierung trotzdem noch aktiv gehalten werden kann. Das heißt, es ist unbedingt erforderlich, dass die Eingaben, die digital getätigt werden, ausdrückbar sind. Dabei sollte die Darstellungsform der analogen Zettel möglichst beibehalten werden, um die Archivierung einheitlich zu halten. Und weil doppelt immer besser hält, sollen die Eingaben auch noch längerfristig auf

einem Server gespeichert werden, um u.a. einen noch schnelleren Zugriff zu ermöglichen. Dieser Server ist derselbe wie der, über den unser Programm später laufen soll, ist somit also nur lokal zugreifbar.

3.4 Kommunikation zwischen Computern

Zur Weiterleitung des (ausgefüllten) Dokuments von Mitarbeiter zum Mitarbeiter müssen Computer untereinander kommunizieren können. Gegenüber äußeren unbefugten Zugriffen sollten die Computer sowie der Kommunikationsweg geschützt sein.

3.5 Verschiedene Rollen

Zum Zwecke der besseren Übersicht und Nachbildung des Workflows ist die Aufteilung in verschiedene Rollen sinnvoll. Dabei kann gewährleistet werden, dass jeder nur das sieht und machen kann, was für ihn relevant ist und so die Fehleranfälligkeit minimiert wird. Die Auswahl der jeweiligen Rolle erfolgt dabei vor jedem Einsatz über ein Menü. Somit ist nicht ein PC an eine Rolle gebunden und man kann sein Setup frei wählen, das heißt beliebig viele Rollen einer Art hinzufügen.

3.6 Skalierbarkeit

Das System der Vierfachvordrucke bietet theoretisch die Möglichkeit, dass auch mehrere Personen eine Rolle übernehmen. Also das es zum Beispiel zwei oder mehr Funker gibt. Das ist ein wichtiges Konzept, da so im Fall das zu viel auf einmal passiert einfach mehr Leute nach bedarf eingesetzt werden können. Demnach ist es auch wichtig, dass dies mit der Software umsetzbar ist. Das man also einfach weitere Workstations in das System einfügen kann, welche die existierenden Rollen unterstützen. Zu achten ist dabei vor allem auf Konflikte bei der gleichzeitigen Bearbeitung von Vordrucken.

4 Funktionale Anforderungen

4.1 Visuelle Unterschiede für verschiedene Rollen

Die visuelle Representation orientiert sich an der Gestaltung des Vierfachvordrucks. Dabei wird vor Allem die Farbgebung weitergeführt, d.h. jede Ansicht wird thematisch in der Farbe des für sie relevanten Vordruckes gehalten. Jede Rolle bekommt ihre eigene Ansicht. Dies dient vor allem der Übersichtlichkeit und Einfachheit der benutzung, da jede Rolle somit nur das sieht, was sie auch wirklich braucht.

4.2 Vierfachvordruck renderbar als PDF

Um die manuelle Archivierung zu gewährleisten, sollten die digital getätigten Eingaben ausdrückbar sein. Dabei sollte möglichst Einheitlichkeit beibehalten werden, d.h. alle Eingaben müssen so in eine digitale Version des Vierfachvordrucks geschrieben werden, wie es der Verfasser auch tun würde. Die Vorlage sollte dabei möglichst in ein PDF konvertiert werden, um gutes Ausdrucken zu gewährleisten. Das Ausdrucken sollte auch so benutzerfreundlich wie möglich vonstatten gehen, also mit abgespeicherter, aber veränderbarer, Vorkonfiguration welche auf einen Knopfdruck abgeschickt werden kann.

4.3 Speicherung im RDF-Format

Die vom Verfasser im Vierfachvordruck eingegebenen Informationen, sollen im RDF Datenmodell auf einem lokalen Server abgespeichert werden. Demnach werden die Eingaben, z.B. 'Absender: Nathanael Arndt' wie in 'example-1.ttl' vorgegeben in der Form

`<VierfachvordruckID> thw:absender 'Nathanael Arndt'.`

in einer .ttl oder .owl Textdatei abgespeichert werden. Auf diese Art werden alle Daten die gesammelt werden innerhalb einer Datei gespeichert.

Die anderen Stationen, die bisher Kopien des Vierfachvordrucks erhalten haben, müssen nun digital auf die Eintragungen des Verfassers zugreifen können. Also müssen gezielte SPARQL-Anfragen implementiert werden, die die Informationen des jeweiligen Vierfachvordrucks aus der Datei auslesen und an die Ausgabe übergeben.

4.4 Verschlüsselung

Die Verschlüsselung sollte nach einem standardisierten oder offen und frei nachvollziehbaren Verfahren ablaufen, damit juristische Integrität gewährleistet ist. Dieser Teil wird in diesem Fall von QuitStore übernommen, was durch die Überschneidung mit Git dessen Versionierung nutzt. Dadurch wird natürlich auch die cryptografische Eindeutigkeit gewährleistet, da die Git-History durch auf einander aufbauende Hash-Codes gesichert ist. Das heißt ändert man etwas an zum Beispiel einem Commit, werden auch alle IDs der Commits usw. danach geändert.

4.5 Datensicherheit

Die Vordrucke werden nach ihrer Bearbeitung und Durchlaufens des weiteren Prozesses natürlich abgespeichert. Damit eine Archivierung der Vordrucke überhaupt sinnvoll ist, muss sichergestellt werden, dass sie von niemandem nachträglich verändert werden können. Ansonsten verliert das Archiv jegliche juristische Relevanz und die Möglichkeit nach zu vollziehen wo und warum Fehler aufgetreten sind. Somit also zwei wichtige Aspekte, welche essentielle Beweggründe für das Anlegen des Archivs sind.

4.6 Zugriffsbeschränkung der Nutzer

Der Vierfachvordruck hat Abschnitte, die nur von speziellen Mitarbeitern ausgefüllt werden dürfen. Dies sollte auch in der Software beachtet werden. So soll jeder Mitarbeiter nur den zu ihm gehörenden Abschnitt bearbeiten beziehungsweise beschreiben, wodurch Verfälschungen des Dokuments verhindert werden sollen.

5 Qualitätsmatrix

| Qualitätskriterium | Gewichtung |
|---|------------|
| Zuverlässigkeit (Reliability) | Hoch |
| Gebrauchstauglichkeit (Usability) | Hoch |
| Funktionalität (Functional Suitability) | Mittel |
| (IT-)Sicherheit (Security) | Mittel |
| Wartbarkeit (Maintainability) | Mittel |
| Effizienz (Performance Efficiency) | Niedrig |
| Portabilität (Portability) | Niedrig |
| Kompatibilität (Compatibility) | Niedrig |

6 Lieferumfang und Abnahmekriterien

Das fertige Produkt umfasst eine voll funktionale Webseite, welche innerhalb einer mobilen Einsatzzentrale die Arbeit mit dem Vierfachvordruck vollkommen ersetzen kann. Aufgrund der großen Verantwortung des THW ist die Zuverlässigkeit der Software besonders wichtig. Um diese zu gewährleisten muss die Software und deren Module vor ihrer Auslieferung und während der Entwicklung vollumfänglichen automatisierten Tests unterzogen werden. Wofür ein noch zu wählendes Testingframework Anwendung findet, welches vorrausichtlich der Unit-Reihe entstammen wird. Eine weitere Komponente ist das zuverlässige Speichern, Abfragen und Darstellen der ausgefüllten Vordrucke. Hierzu wird ein Datenbanksystem aufgesetzt, welches das RDF-Modell verwendet, SPARQL als Querylanguage unterstützt und die Quit-Store Speicher-Architektur nutzt. Die Darstellung orientiert sich stark am Original des Vierfachvordrucks und erfolgt über eine Webseite. Um die Verlässlichkeit weiter zu steigern wird darüber hinaus das Prinzip der kontinuierlichen Integration angewandt. Somit kann eine gemeinsame, einfach erreichbare und verlässliche Codebasis für das gesamte Team geschaffen werden. Ein weiteres wichtiges Abnahmekriterium ist die hohe Gebrauchstauglichkeit, welche eine wichtige Vorraussetzung für die Akzeptanz der Software ist. Hierbei ist insbesondere die schnelle Zugänglichkeit, die klare Trennung der Rollen und die Möglichkeit des Effizienten und schnellen Arbeitens zu gewährleisten. Aus der geforderten Usability folgt auch eine entsprechende Performanz, welche das unterbrechungsfreie, parallele Arbeiten von bis zu acht Nutzern (vier Sachgebetsbearbeiter, zwei Funker, ein Sichter und ein Stabsleiter) garantiert, wobei als Hardware ein Raspberry Pi vorgesehen ist. Um die Akzeptanz weiter zu steigern, soll die Software in der Lage sein den Nutzern durch ihre Funktionalität weitere Aufgaben abzunehmen oder zumindest zu erleichtern. Dazu seien bspw. das Drucken der rechtlich belastbaren Unterlagen und die verschiedenen Sichten mit ihren Zugriffsbeschränkungen genannt. Im Hinblick auf die mögliche Weiterentwicklung ist auch die Wartbarkeit durch eine ausführliche, wohl strukturierte Dokumentation, sowie einheitlich formatierten und gut leserlichen Code möglichst komplikationsfrei zu gestalten. Die Kompatibilität muss in sofern gegeben sein, dass die Webseite auf unterschiedlichen Geräten gut erkenn- und bedienbar sein soll. Durch die feste Hardware ist die Portabilität weniger wichtig, trotzdem sollte die Möglichkeit bestehen ohne großen Aufwand auf ein anderes System zu migrieren.

7 Vorprojekt

Der Fokus im Vorprojekt liegt auf der Erstellung der Benutzeroberfläche zur Erstellung eines Vierfachvordrucks. Die GUI wird über den Browser aufgerufen und ermöglicht in der Startseite die Auswahl der entsprechenden Rolle über ein Dropdown-Menü. In derselben Übersichtsseite sind die für die ausgewählte Rolle relevanten Informationen ersichtlich. Jede Rolle hat einen eigenen View. Beim Erstellen eines neuen Vordrucks soll das Design des Formulars im Vorprojekt weitestgehend fertig umgesetzt sein. Zu Demonstrationszwecken sollen einige fest voreingestellte Vordrucke zur Verfügung stehen. Neu erstellte Vordrucke bleiben zunächst nur temporär bestehen und gehen beim Verlassen der Seite verloren. Dadurch sollen Testnutzer in der Lage sein, einen ersten Überblick über den Prozess des Verfassens eines Vierfachvordrucks zu erhalten, ohne sich Gedanken über die Integrität der Daten machen zu müssen. Der Workflow liegt hierbei zunächst nicht im Fokus; dieser wird erst im fertigen Projekt ersichtlich sein, da hierzu ein funktionsfähiges Backend benötigt wird.

8 Glossar

8.1 RDF

Kurz für Resource Description Framework, ist ein Datenmodell, das stark von der üblichen Vorstellung von Daten in Tabellen abweicht. Alle Daten werden immer in Triples der Form Subjekt, Prädikat, Objekt dargestellt, z.B. ein Subjekt Leipzig ist durch das Prädikat hatHochschule mit dem Objekt Universität Leipzig verbunden. Durch diese Schreibweise bildet sich bei Verwendung einer großen Menge von Triples eine riesige Baumstruktur auf, in der viele Entitäten durch Relationen miteinander verknüpft sind.

8.2 SPARQL

Kurz für Semantic Protocol and RDF Query Language, ist wie SQL für tabellenbasierte Datenbanken, eine Anfragesprache, mit der man gezielt Informationen aus einem RDF Datenmodell extrahieren kann.

8.3 Framework

Ein Framework stellt dem Softwareentwickler einen Rahmen mit festen Regeln und wiederverwendbare Strukturen zur Verfügung. Insbesondere definiert es den Kontrollfluss der Anwendung und die Schnittstellen der Klassen. Im Gegensatz zur Arbeit mit Klassenbibliotheken werden nicht bereits vorhandene Klassen und Funktionen verknüpft und verwendet um den Ansprüchen des Projekts zu genügen, stattdessen werden selbständig Funktionen und Klassen in einem vorgegebenen Rahmen implementiert, welche dann vom Framework genutzt werden um vordefinierte Aufgaben zu erledigen. Woraus sich auch ergibt das Frameworks stark Anwendungsspezifisch sind.

8.4 Modultest

Modultests sind ein Bestandteil des Softwaretests und beschreiben das automatische Testen von Softwaremodulen auf deren korrekte Funktionalität. Hierfür müssen im Vorfeld Testfälle mit möglichst umfassenden Eingabeparametern und den zu erwartenden Ausgabeparametern definiert werden, mit welchen das Modul - bspw. unter Verwendung eines Testing Frameworks - auf Funktionalität geprüft wird.

8.5 Deployment

Bezeichnet die Visualisierung und Definition von Schritten und der Rollenverteilung in einem Prozess. Die aus Deployment resultierenden Flowcharts stellen eine Reihenfolge von Aktionen/Schritten, sowie die Interaktion zwischen verschiedenen Personen und Gruppen dar. Der große Vorteil dieses Modells liegt bei Hervorhebung von Ineffizienz, Dopplung und unnötigen Abläufen.

8.6 Model-View-Controller

MVC ist ein Muster zur Implementierung von User-Interfaces, welches ein Programm in 3 Teile zerlegt. Dies soll die interne Repräsentation von Informationen und deren Verarbeitung von der Art wie diesem dem Nutzer präsentiert werden trennen. Dies erhöht die Wiederverwendbarkeit von Code und erlaubt die parallele Entwicklung der verschiedenen Teile. Das Model ist hierbei die zentrale Komponente, welche das Problem beschreibt. Es managt direkt die Daten und Logik des Programms. Die View-Komponente besteht aus der Darstellung der Daten z.B. durch Tabellen oder Diagramme und ermöglicht es verschiedene Views für verschiedene Nutzer zu definieren. Der Controller akzeptiert User-Input der dann in die entsprechenden Befehle für Model und View umgewandelt wird. Ursprünglich wurde MVC vor allem für die Entwicklung von Desktop GUIs benutzt, allerdings hat sich das Muster auch mittlerweile für die Entwicklung von Web- und Mobile- Applikationen bewährt.

8.7 Java

Java ist eine allzweck Programmiersprache, welche Klassen- und Objektorientiert ist und sich vor allem durch ihre Plattformunabhängigkeit auszeichnet. Der Quellcode wird hierfür in bytecode kompiliert, der dann auf jeder Java virtual machine unabhängig von System oder Architektur zum laufen gebracht werden kann.

8.8 Markdown

Markdown ist eine einfache Auszeichnungssprache, und lässt sich unter anderem direkt in HTML übersetzen. Die Ausdrücke sind einfach zu lesen, zu schreiben und zu bearbeiten. Verwendet wird Markdown z.B. auf GitHub oder auch für Jekyll

8.9 HTML

HTML steht für Hypertext Markup Language und ist eine Sprachkonvention, mit der digitale Dokumente für die Interpretation durch Web-Browser semantisch strukturiert werden. HTML wurde von Forschern entwickelt, um Informationen digital austauschen zu können, ohne diese mehrmals umformen und anpassen zu müssen. Ein HTML-Dokument besteht aus einem Doctype-Tag (Dokumenttypdeklaration), Head (Meta Informationen) und Body (der Inhalt).

8.10 CSS

CSS steht für Cascading Style Sheets ist eine Gestaltungssprache, über die sich die Darstellung von Informationen steuern lässt. Im Gegensatz zu HTML wird sich dabei nicht auf den Inhalt bezogen, sondern allein auf das Aussehen.

8.11 Skriptsprache

Eine Programmiersprache, die vor allem durch Interpreterausführung und oftmals entfallenden Deklarationszwang auszeichnen.

8.12 JavaScript

JavaScript ist eine Skriptsprache, die dazu dient, Benutzerinteraktionen mit Webseiten auszuwerten und daraufhin Inhalte anzupassen, nachzuladen oder zu erzeugen. Dadurch sollen die Funktionalitäten von HTML und CSS erweitert werden. JavaScript ist eine dynamisch typisierte, objektorientierte, klassenlose Skriptsprache.

8.13 Continuous Integration

Der Begriff Continuous Integration (CI) beschreibt eine Softwareentwicklungspraxis, in der alle beteiligten Entwickler eines Projektes häufig (ein- bis mehrmals am Tag) ihren aktuellen Arbeitsstand (meistens verwirklicht in einem dezidierten Branch) in den Hauptstand (Masterbranch) integrieren. Um dieses Ziel zu erreichen werden meist folgende Tools im Entwicklungsprozess verwendet:

- Version Control System (z.B. Git)
- automatisierte Unit-Tests
- CI-Server (z.B. Github CI)

Die direkten Vorteile der CI sind zum einen der minimale Integrationsaufwand sowie die Fähigkeit, jederzeit eine stabile Version veröffentlichen zu können. Der minimale Integrationsaufwand ergibt sich durch die Häufigkeit, mit der die Entwickler ihren Arbeitsstand integrieren. Durch die Bedingung dass vor jeder Integration alle automatisierten Testfälle ohne Fehlermeldungen durchlaufen müssen ergibt sich ein relativ stabiler Hauptstand, der jederzeit als Grundlage für neue Features dienen kann. Bugs die nicht durch Unit-Tests erkannt wurden und erst im Release auffallen, können dank der kompakten Natur kurzer Integrationszyklen schnell erkannt und behoben werden. Kommunikation ist insofern ein wichtiger Aspekt der CI, als dass jederzeit alle Entwickler den Zustand der Software kennen können sollten. Hierbei hilft der Einsatz eines CI-Servers, der beispielsweise jederzeit wenn Code in den Hauptstand integriert werden soll, den kompletten Build Prozess nachvollzieht, alle Tests durchlaufen lässt und entsprechend alle auftauchenden Fehlermeldungen an das Team weitergibt

8.14 git

Git ist eine freie Software zur verteilten Versionsverwaltung. Die Entwicklung wurde von Linus Torvalds angestoßen. Gegenüber anderen Versionskontrollsystemen unterscheidet es sich allerdings in einigen Aspekten:

- Nicht lineare Entwicklung:
D.h. jeder Entwickler hat sein eigenes Repository und jedes Feature wird in seinem eigenen Branch entwickelt.
- "Branching"(Erstellen neuer Entwicklungszweige) und
- "Merging"(Verschmelzen mind. zweier Entwicklungszweige) sind integrale Bestandteile der Arbeit mit Git. Des Weiteren werden nativ Tools zur Visualisierung der nicht linearen Geschichte bereit gestellt.
- Kein zentraler Server:
Jeder Benutzer hat eine lokale Kopie des gesamten Repository und der Versionsgeschichte. Das bietet zum einen Sicherheit (Datensicherung) und zum anderen ist für die meisten Tätigkeiten so kein Netzwerkzugriff erforderlich. Zwischen den lokalen und entfernten Repositories gibt es zwar keinen technischen Unterschied, aber meist gibt es ein offizielles Repository.

- Kryptographische Sicherheit der Projektgeschichte:
Die Projektgeschichte wird so gespeichert, dass der Hash-Code einzelner Commits immer auf der ganzen Geschichte basiert. Somit ist nachträgliche Manipulation nicht möglich, ohne den Hash-Code automatisch zu ändern.
- Unterstützung vieler Übertragungsprotokolle:
Die Unterstützung diverser Übertragungsprotokolle wie HTTP, HTTPS, FTP, SSH und rsync bietet dem Nutzer große Freiheit seine Daten zu senden wie er es für richtig hält.
- Säubern des Repositories:
Auch wenn man Daten/Aktionen aus den Repositories löscht/zurück- nimmt, so bleiben sie doch erstmal erhalten und können zu jedem Zeitpunkt wiederhergestellt werden, was Fehler beim entfernen von Dateien/Aktionen sehr schwer macht. Erst wenn man sich nach diesem Schritt sicher ist kann man die Daten explizit und für immer löschen.

8.15 QuitStore

Software die Git-Versionierung für sogenannte "Named Graphs" (ein Schlüsselkonzept des Semantic Web das bewirkt, dass eine Menge an RDF Ausdrücken über eine URI identifiziert werden) ermöglicht