

Freie Universität Berlin

Bachelorarbeit am Institut für Informatik der Freien Universität Berlin

Upgrade und Neugestaltung des Wiki am Fachbereich Mathematik und Informatik

Bruno Bergmann

Matrikelnummer: 5446789

brub01@inf.fu-berlin.de

Eingereicht bei: Prof. Dr. Lutz Prechelt
Zweitgutachter/in: Prof. Dr. Katharina Baum

Berlin, 24. September 2023

Zusammenfassung

In dieser Arbeit wurde ein Upgradeprozess für das Fachbereichswiki des Fachbereichs Informatik und Mathematik an der Freien Universität Berlin entwickelt. Dabei wurde iterativ vorgegangen und immer wieder getestet und verbessert. Das Ergebnis ist ein schmalere Prozess, der wenig Fehlerpotenzial bietet und wird in dieser Arbeit beschrieben und bewertet. Zudem wurde das Aussehen und die Nutzerführung des Wikis an moderne Standards angepasst. Die Arbeit erläutert und begründet die getroffenen Entscheidungen und führt die Schwierigkeiten einer solchen Aktualisierung eines laufenden Systems vor Augen.

Eidesstattliche Erklärung

Ich versichere hiermit an Eides Statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis angegeben, Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

24. September 2023

Bruno Bergmann

Inhaltsverzeichnis

1	Einleitung	7
1.1	Hintergrund und Motivation	7
1.2	Zielsetzung	7
2	Theoretischer Hintergrund	7
2.1	Allgemeine Einführung in das Wiki-Konzept	8
2.1.1	Konzept von Foswiki	8
2.2	Relevante Konzepte und Technologien	9
2.2.1	Architektur und technische Grundlagen von Foswiki	10
3	Foswiki Upgrade	11
3.1	Analyse des aktuellen Foswiki Systems	11
3.2	Identifizierung des Verbesserungspotentials	15
3.2.1	Analyse der Anforderungen und Erwartungen	15
3.2.2	Entwicklung eines Upgradeprozesses	15
3.3	Durchführung der Migration und des Upgrades	17
3.4	Evaluation der Ergebnisse	20
3.5	Herausforderungen und Lessons Learned	21
4	Foswiki Neugestaltung	23
4.1	Befragung der verschiedenen Parteien	23
4.2	Ziele und Implementierung	24
4.3	Evaluation der Designänderungen	25
5	Fazit und Ausblick	26
5.1	Zusammenfassung der Arbeit	26
5.2	Beantwortung der Zielsetzungen	27
5.3	Ausblick auf zukünftige Trends	27

1 Einleitung

1.1 Hintergrund und Motivation

Das Wiki am Fachbereich *Informatik und Mathematik* der Freien Universität Berlin ist ein zuverlässiger Bestandteil des Wissensmanagements am Fachbereich. Im Jahr 2003 wurde die Wiki-Software TWiki erstmals eingesetzt, zehn Jahre später wurde im Rahmen einer Bachelorarbeit in den Nachfolger Foswiki migriert [27]. Heute, weitere zehn Jahre später, steht die nächste Aktualisierung an. Die zugehörigen Überlegungen und deren Umsetzung werden in dieser Bachelorarbeit beschrieben und bewertet.

Mein persönliches Interesse für kollaborative Plattformen sowie meine Leidenschaft für eine optimale Benutzererfahrung motivierten mich, diese Aufgabe zu übernehmen. Aber auch die sozialen Aspekte rund um eine Wikigemeinschaft weckten mein Interesse für dieses Thema. Als ich mich näher mit dem Foswiki-System beschäftigte, faszinierte mich seine vielfältige Einsatzmöglichkeit als kollaboratives Werkzeug. Insbesondere konnte ich einen Einblick gewinnen, wie die Foswiki-Software aufgebaut ist und wie sie am Fachbereich betrieben wird. Die Kombination aus dem Interesse für die Struktur des Systems und dem Bestreben, die Benutzererfahrung zu optimieren, verleiht meiner Arbeit eine ganzheitliche Perspektive, die sowohl technische als auch menschenzentrierte Herausforderungen anspricht.

1.2 Zielsetzung

Das erste Hauptziel meiner Arbeit ist es, einen Upgradeprozess zu entwickeln, der das Foswiki auf den neuesten Stand bringt und diese Aufgabe auch in Zukunft übernimmt. Das zweite Hauptziel ist es, die Benutzererfahrung durch eine Neugestaltung des Foswikis zu verbessern und die Nutzerinteraktion zu erhöhen. In Bezug darauf gibt es einige Anforderungen von verschiedensten Parteien an der Universität. Aus diesem Grund setzen sich die Hauptziele aus den folgenden Teilzielen zusammen.

- Die Anforderungen der betroffenen Parteien an das System sollen evaluiert und bestmöglich erfüllt werden.
- Das Upgrade soll mit möglichst wenig Schritten durchzuführen sein bzw. wenig Fehlerpotential bieten.
- Die Webseite soll ein mobiles Layout erhalten.
- Das Layout soll sich an dem Corporate Design der Freien Universität Berlin orientieren.
- Die Nutzer sollen angeregt werden, etwas zum Wikiinhalt beizutragen.

2 Theoretischer Hintergrund

In diesem Abschnitt wird der Grundstein für die nachfolgenden Kapitel gelegt. Es wird zunächst knapp das Wiki-Konzept erklärt, um die Anforderungen an ein solches System nachvollziehen zu können. Im nächsten Schritt werden die Besonderheiten des Foswikis herausgearbeitet. Danach werden relevante Konzepte und Technologien erläutert, die im

2. Theoretischer Hintergrund

Verlauf der Arbeit von Bedeutung sind. Abschließend werden die Architektur und die technischen Grundlagen des Foswikis beleuchtet.

2.1 Allgemeine Einführung in das Wiki-Konzept

Als das World-Wide-Web (WWW) von Tim Berners-Lee entwickelt wurde, strebte er ein veränderbares WWW an. Die Nutzer sollten Inhalte nicht nur konsumieren, sondern auch selbst erstellen und bearbeiten [31]. Dieses Konzept scheint heute alltäglich, wenn man einen Blick auf die modernen sozialen Medien wirft. Ward Cunningham entwickelte aus dieser Idee das Wiki-Konzept [31]. Demnach ist ein Wiki eine Applikation im Web, bei der die Nutzer Textinhalte gemeinsam erstellen und bearbeiten können. Dadurch eignet es sich hervorragend zum kollaborativen Arbeiten [13]. Zusätzlich zu den Basisfunktionen bieten die meisten Wikiimplementierungen noch weitere Funktionen. Dazu zählen die Folgenden [31].

Versionskontrolle: Alle Änderungen an einem Inhalt können transparent nachvollzogen und wiederhergestellt werden. Die Änderungen werden nummeriert und Revision genannt.

Verknüpfungsfunktion: Inhalte können beliebig untereinander verknüpft werden.

Volltextsuche: Alle Inhalte können durchsucht werden.

WYSIWYG-Editor: ¹ Mit einem grafischen Texteditor können Inhalte ohne besondere Kenntnisse editiert werden.

Wikis eignen sich hervorragend zum Wissensmanagement und lassen sich effektiv in der Lehre einsetzen. Auch werden Wikis oft als Organisationshilfe eingesetzt [13], so auch am Informatik und Mathematik Fachbereich der Freien Universität Berlin.

2.1.1 Konzept von Foswiki

Die ausschlaggebenden Eigenschaften von Foswiki sind, dass es zum einen kostenlos, Open-Source und sehr ausgereift, zum anderen speziell auf Firmen und Organisationen ausgerichtet ist. Dabei ist die Veröffentlichung des Quellcodes unter einer Open-Source-Lizenz die Grundlage für diese Arbeit. Ohne die Möglichkeit, den Quellcode lesen und verändern zu können, wäre dieser Grad von Anpassung nicht möglich. Auch ist Foswiki erweiterbar durch Plugins, sodass ganze Anwendungen im Browser erstellt werden können [19]. Ein Beispiel hierfür findet sich beim IT-Dienst des Fachbereichs wieder. Dieser stellt ein Formular im Wiki bereit, mit dem Bürocomputer bestellt werden können. Auch zeigen die Infobildschirme am Fachbereich selbst Wikiseiten an und können so einfach aus der Ferne aktualisiert werden. Zudem werden die Wikiseiten im Klartext gespeichert und können dadurch mit Skripten automatisch generiert und auf dem aktuellen Stand gehalten werden. Diese Erweiterbarkeit macht Foswiki zu einer außergewöhnlichen Wikiimplementierung.

Der Inhalt des Foswikis ist in verschiedene Bereiche, sogenannte Webs, aufgeteilt. Die Webs bestehen aus Seiten, sogenannten Topics. Bei der ersten Installation existieren nur die wichtigsten Webs, darunter das MainWeb und das SystemWeb. Das MainWeb speichert einige Topic-Vorlagen für die Webseite und die Benutzer-Topics, „Über-Mich-Seiten“,

¹ "What-you-see-is-what-you-get"

die jeder Nutzer bei der ersten Anmeldung automatisch zugewiesen bekommt. Im SystemWeb sind die systemweiten Konfigurationen gespeichert. Dabei ist wichtig zu erwähnen, dass alle Konfigurationen ebenfalls als Topics gespeichert sind und damit auch grafisch bearbeitet und versioniert werden können.

2.2 Relevante Konzepte und Technologien

Im Folgenden werden die grundlegenden Konzepte und Technologien erläutert, die im weiteren Verlauf der Arbeit benötigt werden. Ziel ist es, ein Grundverständnis für die verwendeten Technologien zu erhalten, um Schwierigkeiten und Entscheidungen nachvollziehen zu können. Die meisten Absätze in diesem Abschnitt werden mit einem Quellenverweis abgeschlossen, der für den gesamten Absatz gilt.

Zunächst wird das Thema Webserver beleuchtet. Ein Webserver ist eine Anwendung, die Webinhalte über das Internet bereitstellt. Der Server versteht die Anfragen eines Webbrowsers und sendet die angeforderten Informationen zurück [40]. Die meist genutzten Webserver sind Apache und NGINX [37]. Beide Optionen haben zusätzlich zu der genannten Funktionalität viele weitere Einsatzzwecke. Im Kontext dieser Arbeit wird diese simple Betrachtung jedoch ausreichen.

Besonders relevant ist auch die Technologie Docker. Docker ist eine Plattform zur Containervirtualisierung von Anwendungen. Sie ermöglicht es, die Anwendung von der zugrundeliegenden Infrastruktur, dem Host, zu isolieren und damit eine architekturunabhängige Produktionsumgebung zu erzeugen. Eine solche Anwendung kann beispielsweise ein Webserver wie Apache oder NGINX sein [11].

Die typische Vorgehensweise zur Erstellung einer Docker-Containeranwendung beginnt mit dem Bau eines Docker-Images. Dazu wird eine sogenannte Dockerfile angelegt, die angibt, welche Schritte dazu ausgeführt werden sollen. Meist wird als Basis das Image eines leichtgewichtigen Betriebssystems verwendet und die nötigen Pakete und Abhängigkeiten für die jeweilige Anwendung werden installiert. Danach werden die nötigen Schritte durchgeführt, um die Anwendung einzurichten. Mit der vollständigen Ausführung des Dockerfile entsteht ein sogenanntes Docker-Image. Es ist der Ausgangspunkt und wird zur Laufzeit zu einem Docker-Container. Beim Start des Containers wird der sogenannte Docker-Entrypoint ausgeführt. Hierbei handelt es sich um ein kurzes Skript, welches die Anwendung startet. Jede Änderung am System wird nach dem Stoppen des Containers verworfen, sodass beim nächsten Start wieder vom Ausgangspunkt des Images begonnen wird. Das ermöglicht eine reproduzierbare Umgebung unabhängig von der Architektur [11].

Meist sollen jedoch einige Dateien erhalten bleiben. Diese können in Docker-Volumes gespeichert werden und sind persistent, auch über die Lebenszeit des Containers hinaus. In einem Volume werden für gewöhnlich die Daten und Konfigurationen der Anwendung gespeichert, nicht aber die Anwendung selbst. Diese Trennung ermöglicht es, die Flexibilität der Anwendung beizubehalten, während gleichzeitig die kritischen Daten sicher und konsistent bleiben. Auch kann so die Anwendung immer wieder aktualisiert werden, indem das Image neu gebaut wird, ohne die Daten des Containers migrieren zu müssen [12].

Da es je nach Anwendungsgröße sehr aufwendig werden kann, das Image zu bauen und die Container mit all ihren Volumes einzeln zu erstellen, stellt Docker die Technologie Docker-Compose bereit. Damit ist es möglich, eine Docker-Compose-Datei zu erstellen, die wie

2. Theoretischer Hintergrund

ein Rezept für die Anwendung funktioniert. In dieser Datei sind alle benötigten Images, Container, Volumes und Netzwerke aufgelistet und können von Docker automatisch mit der Ausführung der Datei erzeugt und gestartet werden [10].

Im Gegensatz zu Docker-Containern steht die Technologie der virtuellen Maschine. Der wichtigste Unterschied zwischen den beiden Technologien ist der Abstraktionsgrad. Eine virtuelle Maschine ist eine Abstraktion der Hardware und simuliert diese für jede virtuelle Maschine einzeln. Jeder virtuellen Maschine werden vor dem Start eine Menge an Prozessorkernen, Arbeitsspeicher, etc. zugewiesen und fest reserviert. Im Gegensatz dazu ist ein Docker-Container eine Abstraktion der Applikation selbst. Der Container existiert als Prozess auf dem Host-System, verhält sich wie ein übliches Programm und verwendet damit nur so viele Ressourcen, wie auch benötigt werden. Dadurch bietet die virtuelle Maschine eine höhere Isolierung und wird meist dafür verwendet, einen vollständigen Computer zu simulieren. Ein Docker-Container bietet weniger Isolierung, dafür mehr Leistung und wird meist verwendet, um nur eine spezifische Funktionalität leichtgewichtig auszuführen [39] [38].

Nun wird das Lightweight Directory Access Protocol (LDAP) betrachtet. Es ist ein weit verbreitetes Protokoll, welches durch die Entwicklungsarbeit der LDAP-Gemeinschaft eine große Erweiterbarkeit mitbringt. Auch wenn dies nicht die einzige Funktionalität ist, wird LDAP in dieser Arbeit vor allem als Authentifizierungs- und Authorisierungsmethode besprochen. Der Fachbereich stellt einen LDAP-Server bereit, der in einer Datenbank die verschiedenen Nutzer und Gruppen des Fachbereiches speichert. Mit einer Serveranfrage werden diese Daten ausgelesen, um beispielsweise eine Liste aller Gruppen in der Datenbank zu erhalten. Auch kann eine Anfrage gestellt werden, ob ein gegebener Nutzernamen und ein Passwort eine valide Kombination sind, um so die Identität eines Nutzers sicherzustellen [33].

Der letzte Baustein ist die wohl bekannteste Software zur verteilten Versionskontrolle: Git. Sie wird auch von der Foswiki-Community verwendet, um Änderungen am Programmcode zu synchronisieren und bereitzustellen. Mit Git werden Quellcode-Änderungen genau aufgezeichnet und es kann zwischen jeder Version beliebig gewechselt werden. Somit können versehentliche Fehler problemlos zurückgerollt werden. Ein Projekt wird Repository genannt und speichert alle Dateien und ihre Änderungshistorie. Die wichtigsten Funktionen von Git für diese Arbeit sind das sogenannte Klonen und Pullen. Beim Klonen wird eine Kopie eines entfernten Repositories heruntergeladen. Dabei werden sowohl die Dateien, als auch die Versionshistorie gespeichert. Ändert sich nun etwas an dem entfernten Repository, können diese Änderungen mit einem sogenannten Pull auf die lokale Kopie übertragen werden, um diese auf den neuesten Stand zu bringen [25].

2.2.1 Architektur und technische Grundlagen von Foswiki

In diesem Abschnitt wird die grundlegende Architektur von Foswiki erörtert. Begonnen wird mit der Installation des Foswikis. Im Anschluss werden spezifischere Konzepte untersucht, die im Verlauf der Arbeit von Bedeutung sind.

Der klassische Weg der Foswiki-Installation ist die sogenannte Paketinstallation. Die Foswiki-Community arbeitet kontinuierlich an dem Projekt, wodurch ständig neue Verbesserungen im Git-Repository [18] bereitgestellt werden. Immer wenn ein Meilenstein erreicht ist, wird ein offizielles Foswiki-Paket mit Versionsnummer veröffentlicht. Dieses

Paket wird Release genannt. Es enthält alle Kernbausteine des Foswikis und einige mitgelieferte Erweiterungen. Das Paket enthält ein vollständig funktionsfähiges Foswiki.

Um die Paketinstallation anzupassen, gibt es das sogenannte Configure-GUI (Graphical User Interface). Dieses lässt sich im Webbrowser aufrufen und es lassen sich alle administrativen Einstellungen an der Plattform über eine grafische Oberfläche setzen. Die Grundlage dafür ist das Configure-CLI (Command Line Interface), das dieselbe Funktionalität über eine Konsole bietet. Das Configure-GUI eignet sich, um komfortabel eine Einstellung zu ändern, ohne den Maschinenraum des Wikis betreten zu müssen. Mit dem Configure-CLI lässt sich das Setzen von Einstellungen auch automatisieren, was vor allem während dem Bauen eines Docker-Images vorteilhaft ist. Alle Einstellungen, die in den beiden Schnittstellen gesetzt werden, sind gespeichert in einer Konfigurationsdatei mit dem Namen `LocalSite.cfg`.

Im Foswiki-Paket werden auch nützliche Administrator-Skripte, sogenannte Tools, mitgeliefert. Dazu gehört unter anderem auch das genannte Configure-CLI, welches ebenfalls ein Tool ist. Wichtig zu erwähnen ist das Tool `tools/bulk_copy`, das eine Datenmigration zwischen zwei vollständigen Foswiki-Instanzen realisiert.

Das Aussehen des Foswikis wird durch einen sogenannten Skin bestimmt. Dieser legt das Aussehen aller Elemente auf der Seite fest. Dazu zählt die Farbe, Schriftart, Schatten, Ränder und viele weitere kosmetische Einstellungen. Ein Skin ist genau betrachtet ein weiteres Plugin für das Foswiki. Sämtliche Konfigurationen des Skins sind im `SystemWeb` als Topic gespeichert. Die Konfiguration kann weiter angepasst werden, indem eine Konfigurationsdatei im `MainWeb` angelegt wird. Diese überschreibt dann die Datei aus dem `SystemWeb` und dient als Standardkonfiguration im Wiki. Bei Bedarf kann sie innerhalb von einzelnen Webs weiter angepasst werden.

Um einen Wikiinhalt zu verfassen, gibt es drei Optionen, die untereinander kompatibel sind. Die einfachste Option, in der ein Topic bearbeitet werden kann, ist der sogenannte Wysiwyg-Editor. Die Abkürzung steht für „What-you-see-is-what-you-get“ und beschreibt den Texteditor, wie man ihn aus gängigen Officeprogrammen kennt. Damit ist diese Möglichkeit einsteigerfreundlich, aber auch limitiert. Die zweite Option besteht darin, das Topic in HTML (Hypertext Markup Language) [26] zu editieren. HTML ist üblicherweise die Sprache, in der Webseiten verfasst werden [36], so auch das Foswiki. Der HTML-Editor ist also von Vorteil, wenn man das Topic auf Quelltextebene verändern möchte. So können einfache HTML-Elemente wie Knöpfe, Links, Tabellen und Banner direkt im Webbrowser eingebaut werden. Zuletzt bleibt noch die sogenannte Wiki Markup Language. Hierbei handelt es sich um eine von Foswiki erschaffene Sprache, die für das Erstellen von Wikiinhalten optimiert wurde. In dieser Sprache lassen sich besonders einfach Topics erstellen und Makros verwenden. Für häufige Wikinutzer ist diese Sprache wahrscheinlich am nützlichsten, weil sie es erlaubt, in einem Markdown [41] ähnlichen Stil schnell Textinhalte zu verfassen. Außerdem ist sie nicht so limitiert wie der Wysiwyg-Editor.

3 Foswiki Upgrade

3.1 Analyse des aktuellen Foswiki Systems

In diesem Abschnitt wird das aktuelle Foswiki-System untersucht und analysiert. Das Ziel der Analyse ist es, ein grundlegendes Verständnis des aktuellen Foswikis-Systems in Bezug

3. Foswiki Upgrade

auf Leistung, Design und Benutzerfreundlichkeit zu erlangen. Durch die Bewertung des Zustands des Foswikis wird die Vorarbeit für das Upgrade geleistet und die Limitationen des Systems werden erkannt. Außerdem ist es das Ziel, das Upgrade so vorzubereiten, dass der Prozess in der Zukunft möglichst wenig Zeit in Anspruch nimmt und komplikationsarm durchzuführen ist.

Zunächst wird das Foswiki-System analysiert, welches grob aus einem Anwendungsserver und einem Webserver besteht. Anschließend wird die zugrundeliegende Umgebung betrachtet, in der das Foswiki ausgeführt wird. Im nächsten Schritt erfolgt die Bewertung des offiziellen Upgradeprozesses des Foswikis und welche Veränderungen daran zu tätigen sind. Zuletzt wird das Upgrade entsprechend der Anforderungen der verschiedenen Parteien am Informatik und Mathematik Fachbereich geplant und durchgeführt.

Foswiki-System

Zunächst wird der Anwendungsserver des Foswikis betrachtet. Das aktuelle Foswiki ist auf dem Stand der Version 2 . 1 . 6 von 2018. Die neueste offizielle Version 2 . 1 . 7 stammt aus dem Jahr 2022 (Stand 27.07.2023). Der Anwendungsserver des Foswikis, der die Logik des Systems implementiert, ist in der Programmiersprache Perl [32] geschrieben. Er kommuniziert mit dem Webserver über das Fast Common Gateway Interface (FCGI) [15]. Dieses Protokoll gewährleistet eine korrekte Weiterleitung von Browser-Anfragen zwischen dem Webserver und dem Anwendungsserver. Der Webserver ist aktuell ein Apache-Webserver und stellt die Wiki-Inhalte für das Ansehen im Webbrowser bereit. Außerdem verwaltet der Apache-Webserver die Authentifizierung der Nutzer. Er kommuniziert mit dem fachbereichsweiten LDAP-Server, wodurch die Nutzer sich mit ihrem Universitätskonto anmelden können.

Die Stärken dieses Aufbaus sind die folgenden:

Leistung und Skalierbarkeit: Der Perl-Anwendungsserver ermöglicht eine hohe Skalierbarkeit und eine gute Leistung des Systems. Perl kann effizient mit einer Vielzahl von Nutzern und Anfragen umgehen.

Erprobte Technologien: Apache und FCGI sind etablierte Technologien in diesem Anwendungsbereich. Durch ihre langjährige Präsenz sind sie stabil und getestet.

Integration von Modulen: Apache unterstützt die Integration von Modulen, um die Funktionalität anzupassen. So ist beispielsweise der Apache-Webserver um die LDAP Authentifizierung erweitert worden.

Die Schwächen des Aufbaus sind:

Lernkurve: Perl ist bekannt für seine eigenwillige Syntax. Dies kann die Einarbeitungszeit deutlich verlängern.

Zuverlässigkeit und Sicherheit: Die Verwendung von FCGI kann zusätzliche Herausforderungen mit sich bringen. Fehler in der Konfiguration können sich auf die Verfügbarkeit und Stabilität des Systems auswirken.

Dieser grundlegende Aufbau des Foswiki-Systems ist also stabil und skalierbar. Für den Einsatzzweck am Fachbereich ist diese Funktionsweise ausreichend und muss nicht verändert werden. Die Nachteile fallen kaum ins Gewicht, da die Lernkurve der Programmiersprache Perl die Entwicklung nicht beeinflusst, wenn an diesem Code nichts verändert

wird. Auch muss FCGI nicht neu konfiguriert werden und es kann sich auf die erprobte Konfiguration des Basisprojektes verlassen werden.

Zugrundeliegende Umgebung

Das Foswiki wird auf einer virtuellen Maschine auf einem Server des Fachbereichs ausgeführt (siehe Abbildung 1). Dieser Aufbau bringt einige Vorteile mit sich. Durch die dedizierte Verwendung einer virtuellen Maschine für das Foswiki gibt es keine Seiteneffekte mit anderen Diensten des Fachbereichs. Das Linux-Betriebssystem der virtuellen Maschine ist typischerweise für Serverdienste geeignet und ermöglicht eine komfortable Wartung des Foswikis über die Kommandozeile. Ein klarer Nachteil wird im Upgradeprozess deutlich. Vor jedem Upgrade sollte eine Sicherungskopie angelegt werden. Es ist auch prinzipiell möglich, den Stand einer virtuellen Maschine zu sichern. Im Rahmen dieser Arbeit hatte ich aber keinen Zugriff auf diese Funktion. Eine reine Dateisicherung kann durch einfaches Kopieren der Daten an eine andere Stelle erfolgen. Diese Lösung ist allerdings weder elegant, noch ausfallsicher. Zum einen besteht eine hohe Wahrscheinlichkeit menschlicher Fehler, wenn beispielsweise Dateien übersehen werden. Denn zu einer Datensicherung des Systems gehören, zusätzlich zu den Foswiki-Dateien, auch noch die Bestandteile des Webservers und von FCGI. Die Anordnung der erforderlichen Dateien kann dabei leicht dazu führen, dass Bestandteile übersehen werden. Zum anderen werden zwar die Daten gesichert, aber nicht der aktuelle Zustand des Systems. Das bedeutet, dass, selbst wenn alle Daten gesichert sind, nicht zum exakt gleichen Zustand zurückgerollt werden kann. Somit können dieselben zugrundeliegenden Daten auf der einen Maschine funktionieren und auf einer anderen nicht.



Abb. 1: Aktuelle Architektur des Fachbereichswikis

Offizieller Upgradeprozess

In diesem Abschnitt wird näher auf die Stärken und Schwächen des offiziellen Upgradeprozesses eingegangen. Die klassische Methode, das Foswiki zu aktualisieren [23], wird im Folgenden erklärt.

Zunächst wird eine ganzheitliche Neuinstallation des Wikis an einem neuen Ort durchgeführt. Dafür wird die offizielle Installationsanleitung [20] von Foswiki verwendet. Danach werden die Daten und Einstellungen in das neue System migriert. Das wesentliche Vorgehen ist hier zusammengefasst.

1. Neuesten Foswiki-Release herunterladen und entpacken:

2. **Apache Webserver konfigurieren:** Foswiki bietet einen Generator für die Konfigurationsdatei an. Auch kann eine selbstgeschriebene Apache-Konfiguration verwendet

3. Foswiki Upgrade

werden.

3. Authentifizierung konfigurieren: Standardmäßig wird das Authentifizieren vom Foswiki-System übernommen. Auch der Apache-Webserver kann den Login übernehmen. So ist es aktuell. Zusätzlich gibt es Erweiterungen, die weitere Login-Manager implementieren.

Durch diese Schritte wurde ein leeres Foswiki in der neuesten offiziellen Releaseversion erstellt. In diese Instanz soll nun migriert werden. Dazu müssen die folgenden Schritte ausgeführt werden.

4. Installation von zusätzlichen Erweiterungen: Alle Erweiterungen, die nicht in der Paket-Installation enthalten sind, müssen erneut von Hand installiert werden.

5. Kopieren der Konfiguration: Die `LocalSite.cfg`-Datei muss vom alten Wiki übertragen werden.

6. Daten kopieren: Das mitgelieferte Skript `tools/bulk_copy.pl` kopiert die Daten aus der alten in die neue Instanz.

7. Inhalt des MainWeb ausgewählt übertragen: Vorlagen und Benutzer-Topics müssen aus dem MainWeb ausgewählt und übertragen werden.

8. Instanz auf Fehler überprüfen: Dafür gibt es keinen allgemeingültigen Weg. Vielmehr muss Erfahrung gesammelt werden, an welchen Stellen das System oft Fehler aufweist, um diese dann zu überprüfen.

9. Alte Installation durch neue ablösen: Nun wird der Apache-Webserver so konfiguriert, dass er auf die neue Installation zeigt und von den Nutzern erreicht werden kann.

Eine große Einschränkung dieser Methode ist, dass sie einen offiziellen Foswiki-Release benötigt. Was zunächst wie eine Nebensächlichkeit klingt, ist nicht ganz unwichtig. Zwischen den Releases liegt regelmäßig ein Jahr, teilweise sind es sogar mehrere Jahre. Und das, obwohl in dem offiziellen Git-Repository [18] kontinuierlich entwickelt wird. Beispielsweise liegen zwischen der Versionen 2.1.7 und dem aktuellen neuesten Stand im Git-Repository [18] 1125 Änderungen (Stand 27.07.2023). Nutzt man also diese Upgrademethode und damit die offiziellen Releases, verzichtet man auf viele Änderungen und damit auch auf Sicherheitsupdates.

Sonst funktioniert die Methode zuverlässig, ist aber aufwendig. Es fällt direkt auf, dass gerade im Upgrade-Teil viele Schritte von Hand durchgeführt werden müssen. Die Installation der Plugins kann je nach Anzahl und Beschaffenheit viel Zeit in Anspruch nehmen. Die meisten zusätzlichen Plugins haben Abhängigkeiten, die nicht automatisch bei der Installation mitinstalliert werden. Beispielsweise benötigt das Fachbereichswiki das `CalendarPlugin` [34]. Dieses Plugin hängt von zwei nicht vorinstallierten Perl-Modulen ab. Sie müssen von Hand auf dem System nachinstalliert werden, wenn man mit einer neuen virtuellen Maschine startet. Dieser Umstand ist ein Beispiel dafür, dass viel spezifisches Wissen über die Beschaffenheit der Plattform und die Abhängigkeiten der Komponenten benötigt wird. Das kann eine große Fehlerquelle beim Upgrade sein, vor allem wenn sich neu in das System einarbeitet wird. Es muss auf viele Feinheiten geachtet werden, die so spezifisch sind, dass sie nicht in einer offiziellen Anleitung stehen können.

Auch die Datenmigration mit dem Skript `tools/bulk_copy.pl` gestaltet sich schwierig. Wie im Upgrade-Guide [23] vermerkt, bringt dieses Skript viele Einschränkungen mit

sich. Zunächst ist das Skript bei der Größe des aktuellen Foswikis außerordentlich langsam. Das liegt daran, dass das Skript abwärtskompatibel sein muss und deswegen nicht einfach die Daten von der Quelle zum Ziel kopiert. Es wird für jede einzelne Datei ein Perl-Prozess gestartet, der über eine Foswiki-Schnittstelle darauf zugreift und jede Revision einzeln durchgeht, gegebenenfalls konvertiert und dann kopiert. Eine Parallelisierung erfolgt nicht. Bei dem aktuellen Fachbereichswiki hat das Durchlaufen des Skripts circa 1,5 Stunden gedauert. Dieser Weg ist offensichtlich nicht optimal. Es ist anzustreben, einen Upgradeprozess zu entwickeln, der keine Migration der Daten erfordert. Auch sollen die Plugins und Konfigurationen erhalten bleiben, damit sie nicht jedes Mal neu erstellt werden müssen.

3.2 Identifizierung des Verbesserungspotentials

In diesem Abschnitt werden Lösungen für die diskutierten Nachteile des letzten Abschnitts erarbeitet. Dabei sollen auch die Anforderungen der Interessensgruppen ermittelt und berücksichtigt werden, um das Upgrade an ihnen auszurichten. Somit kann sichergestellt werden, dass das Upgrade ihre Anforderungen erfüllt. Neben einer simplen Softwareaktualisierung soll das Upgrade auch weitere Verbesserungen mit sich bringen, die das aktuelle Foswiki System nicht vorweisen kann.

3.2.1 Analyse der Anforderungen und Erwartungen

Von dem zuständigen Systemadministrator, Bodo Riediger-Klaus, existieren mehrere Anforderungen an das Wiki-Upgrade und vor allem den Upgradeprozess selbst. Zunächst muss das Foswiki auf den neuesten Stand des Git-Repositorys [18] aktualisiert werden, nicht nur auf den des neuesten offiziellen Releases. Die Methode der Aktualisierung soll geradlinig, ausfallsicher und bei Fehlern einfach zu revidieren sein. Zudem soll das neue Foswiki alle Funktionen des aktuellen Foswikis unterstützen. Zusätzlich muss das Aussehen des Wikis an das Corporate Design der Freien Universität Berlin angepasst werden. Dieser Prozess wird im nächsten Kapitel beschrieben. Aus diesen Anforderungen ergeben sich die Hauptbereiche, in denen nun Lösungen für die Schwächen des aktuellen Wikis erarbeitet werden.

3.2.2 Entwicklung eines Upgradeprozesses

Im Hinblick auf die Schwächen des aktuellen Wikis und die Anforderungen wird im Folgenden ein Upgradeprozess diskutiert. An dem Foswiki-System selbst (Anwendungsserver, Webserver, FCGI) wird nichts geändert, da es seit Jahren aktiv und zuverlässig entwickelt wird. Es ist davon auszugehen, dass die Foswiki-Gemeinschaft auch in Zukunft das System weiterentwickeln wird. Verändert wird aber der grundlegende Aufbau des Systems innerhalb der virtuellen Maschine. Aktuell sind alle Komponenten des Foswiki-Systems nativ auf dem Betriebssystem installiert. Um eine flexible Handhabung des Foswikis zu gewährleisten, wird das Wiki in der neuen Version in einem Docker-Container verpackt (siehe Abb. 2). Dieser Aufbau bringt sowohl Vorteile, als auch Nachteile mit sich. Zu den

3. Foswiki Upgrade

Vorteilen zählen die Folgenden:

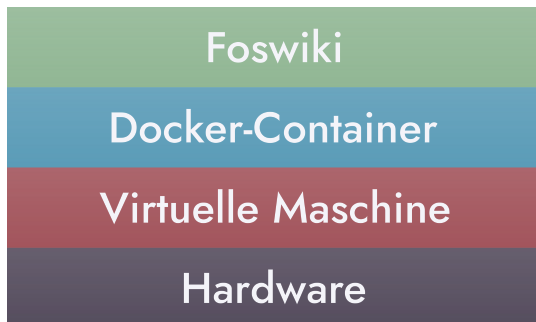


Abb. 2: Architektur des neuen Wikis

Portabilität: Docker ermöglicht eine einfache Portierung zwischen verschiedenen Umgebungen. Der Container kann auf verschiedenen Architekturen ausgeführt werden, ohne dass sich um die Abhängigkeiten und Konfigurationen gekümmert werden muss. Dies macht die Entwicklung der neuen Foswiki-Instanz komfortabel, da diese auf einem beliebigen Rechner erstellt und schlussendlich auf dem Produktivsystem aufgesetzt werden kann.

Konsistente Umgebung: Es wird sichergestellt, dass das Foswiki in einer konsistenten und reproduzierbaren Umgebung läuft. Dies beugt potenziellen Kompatibilitätsproblemen bei einem Systemwechsel oder nach Betriebssystem-Upgrades vor.

Isolierung: Durch eine strikte Isolierung des Containers können Sicherheitsrisiken minimiert werden, da die Anwendung nicht direkt auf dem Host-System läuft.

Versionierung: Mit Docker können verschiedene Versionen des Foswikis in unterschiedlichen Containern betrieben werden. Es kann problemlos zwischen den Versionen gewechselt werden, falls erneut eine ältere Version benötigt wird.

Gerade die Versionierung erleichtert den Upgradeprozess ungemein. Mit Docker können deckungsgleiche Kopien des Systems angelegt werden. Das macht eine Sicherung und Wiederherstellung des Systems komfortabel.

Die Nachteile dieses Aufbaus sind die folgenden:

Ressourcenverbrauch: Docker-Container fügen eine zusätzliche Abstraktionsebene hinzu, die zu einem leicht erhöhten Ressourcenverbrauch führt. Es kann effizienter sein, die Anwendung auf dem Host-System laufen zu lassen.

Komplexität: Die zusätzliche Abstraktionsebene erhöht die Komplexität der Infrastruktur. Gerade, wenn eine Person üblicherweise nicht mit Docker arbeitet, kann dies zu einer erhöhten Einarbeitungszeit und vermeidbaren Fehlern führen.

Abhängigkeit von Docker: Die Nutzung von Docker bindet die Anwendung an diese Technologie. Falls in der Zukunft auf eine andere Lösung zurückgegriffen werden wird, kann dies mit zusätzlichem Aufwand verbunden sein.

Für die Anwendung am Fachbereich sind diese Nachteile eindeutig mit den Vorteilen aufzuwiegen. Der erhöhte Ressourcenverbrauch ist minimal und grundsätzlich kein Problem, da der Fachbereich problemlos genügend Rechenleistung, für eine vergleichbar kleine

Anwendung wie diese, vorweisen kann. Die erhöhte Komplexität ist sicherlich zunächst ein Nachteil. Langfristig ist dennoch das Gegenteil der Fall, da die Wartung des Systems deutlich einfacher ist als die erste Erstellung. Auch die Abhängigkeit von Docker ist nicht schwerwiegend. Falls in der Zukunft wieder eine native Installation in der virtuellen Maschine gewünscht ist, können die Daten aus den Docker-Volumes mit dem `tools/bulk_copy` Skript in die neue Instanz migriert werden. Aus diesen Gründen habe ich mich für eine Docker Umgebung für das neue Foswiki entschieden.

Im nächsten Schritt wird der Upgradeprozess selbst diskutiert. In der Analyse des Foswiki-Systems wurde bereits auf den offiziellen Upgradeprozess [23] eingegangen. Die größten beiden Kritikpunkte sind die Abhängigkeit von den seltenen offiziellen Releases und der mit dem Upgrade verbundene Aufwand. Um diese Nachteile zu kompensieren, wurde sich für einen Weg entschieden, der ein Upgrade mit Rolling-Releases, also der neuesten Version des Foswikis Git-Repositorys [18], ermöglicht. Außerdem ist eine Migration der Daten in diesem Fall nicht nötig, da direkt im produktiven Docker-Container aktualisiert werden kann.

3.3 Durchführung der Migration und des Upgrades

In diesem Abschnitt wird die Migration des aktuellen Foswikis in den Docker-Container erläutert. Diese wird einmalig durchgeführt, um die Docker-Containerumgebung einzurichten. Anschließend wird der Upgradeprozess genauer betrachtet, der in Zukunft für die Aktualisierung des Systems verwendet wird.

Datenmigration

Bei der Erzeugung des Docker-Images für das Foswiki habe ich mich maßgeblich an einem Projekt von Timothy Legge [30] orientiert. Dieses Projekt bietet eine gute Basis, um ein Foswiki für den Fachbereich einzurichten. Es wird bis heute (Stand 4.8.2023) aktiv verbessert und ist gut dokumentiert. Ein großer Unterschied zum aktuellen Fachbereichswiki ist die Verwendung von NGINX statt Apache als Webserver. NGINX ermöglicht im Basispaket allerdings keine Authentifizierung mit LDAP. Der Login wird für das neue Foswiki von dem `LdapContribPlugin` übernommen. Im Folgenden wird das Bauen des Docker-Images betrachtet und auf die Änderungen eingegangen, die getätigt wurden.

Die wichtigsten Schritte beim Bauen des Docker-Image mit Timothy Legges Projekt werden im Folgenden erläutert. Auch werden die benötigten Änderungen für die Verwendung als Fachbereichswiki beschrieben. Um das Docker-Image zu bauen, wird nun die Dockerfile ausgeführt. Sie besteht aus den folgenden Instruktionen:

- 1. Holen des Betriebssystems:** Die neueste Version des Images von Alpine Linux [1] wird heruntergeladen. Dieses Betriebssystem ist eine minimale Linux-Variante, die kaum mitgelieferte Pakete oder Programme enthält. Durch den minimierten Ressourcenverbrauch ist Alpine Linux optimal als Grundlage für ein Docker-Image geeignet.
- 2. Installieren aller nötigen Pakete:** Es werden alle nötigen Pakete mit dem Paketmanager Alpine Package Keeper (APK) [2] installiert. Darunter eine große Anzahl an Perl-Modulen, die für das Foswiki und die Foswiki-Plugins benötigt werden. Dieser Schritt wurde für das Fachbereichswiki um einige Pakete ergänzt, die nötig sind, um spezi-

3. Foswiki Upgrade

elle Plugins, wie das genannte CalendarPlugin, zu unterstützen. Auch das Git-Paket wurde hinzugefügt, da es für das Upgrade benötigt wird.

- 3. Deinstallieren überflüssiger Pakete:** Einige Pakete werden nur ein einziges Mal während der Installation benötigt. Um keinen überflüssigen Speicherplatz zu verbrauchen werden diese nach ihrer Verwendung wieder deinstalliert.
- 4. Holen des offiziellen Foswiki-Releases:** Der neueste Foswiki-Release wird heruntergeladen und entpackt. Zu beachten ist, dass es sich hierbei weiterhin um den offiziellen Release handelt. Er dient als Grundlage. Im nächsten Abschnitt wird auf den neuesten Stand des Git-Repositorys aktualisiert.
- 5. Installation zusätzlicher Plugins:** Es wird eine Auswahl an Plugins installiert, die als Basis für ein funktionierendes Foswiki dienen. An dieser Stelle wird beispielsweise das LdapContribPlugin installiert, welches für den LDAP-Login zuständig ist. Für das Fachbereichswiki wurden einige Plugins hinzugefügt, die bei den Nutzern gefragt sind. Darunter das CalendarPlugin [34] und das LatexModePlugin [21].
- 6. Konfiguration der Plattform:** Die Plattform wird so eingestellt, dass sie schon beim ersten Start funktioniert. Für das Fachbereichswiki wurden hier einige Einstellungen verändert und neue hinzugefügt. Beispielsweise wird die Konfiguration des Ldap-Logins in diesen Schritt vorgenommen, sodass sich Nutzer auf dem Foswiki anmelden können.

Wie zu erkennen ist, braucht der Build-Prozess kaum Veränderung, um an die Anforderungen des Fachbereichswikis angepasst zu werden. Es benötigt für die Verwendung am Fachbereich lediglich einige zusätzliche Pakete, Plugins und Konfigurationen. Mit diesen Anpassungen steht nun ein Docker-Image bereit, womit im Folgenden der Docker-Container gestartet wird. Der Befehl `docker-compose up -d` liest die mitgelieferte Docker-Compose-Datei und startet den Foswiki-Container. Für die Anwendung im Fachbereich wurde an der Docker-Compose-Datei von Timothy Legge geändert, dass die Docker-Volumes an Verzeichnisse im Host-System gebunden werden. Diese Methode heißt Bind-Mounting. Dadurch kann auch vom Host-System die Dateistruktur im Container eingesehen und verändert werden. Außerdem verwendet Timothy Legge in seiner Docker-Compose-Datei sein eigenes Docker-Image aus dem offiziellen Docker-Image-Repository. Dieses Image wurde für das Fachbereichswiki lokal mit den beschriebenen Änderungen gebaut. Zusätzlich zum Foswiki-Container wird durch die Docker-Compose-Datei noch ein weiterer Container gestartet, der die Suchmaschine Solr [3] beinhaltet. Mithilfe des Solr-Containers kann das gesamte Wiki durchsucht werden. Da daran aber keine zusätzlichen Änderungen getätigt wurden, wird Solr nicht weiter beleuchtet.

Nach der Ausführung der Docker-Compose-Datei führt der Container den Docker-Entrypoint aus. Damit ist das Foswiki das erste Mal gestartet und kann im Webbrowser aufgerufen werden. An dieser Stelle müssen noch einige Anpassungen am System vorgenommen werden, damit das Foswiki ordnungsgemäß funktioniert. Da es sich hierbei um Details handelt, die auch in der angefertigten Dokumentation [14] nachgelesen werden können, wird dieser Teil nicht weiter beleuchtet. Im Folgenden wird die Migration der Daten von dem aktuellen in das neue Foswiki erläutert. Dieser Schritt muss ein einziges Mal ausgeführt werden. Danach kann das Foswiki durch ein Upgrade mit Git ohne eine dedizierte Datenmigration aktualisiert werden. Zunächst wird der Zugriff auf das aktuelle

Foswiki für alle Nutzer, außer Administratoren, eingeschränkt. Auf diese Weise kann sichergestellt werden, dass während der Datenmigration keine Änderungen am aktuellen Wiki getätigt werden. Dann kann der Migrationsvorgang beginnen. Dafür wird das Skript `tools/bulk_copy.pl` mit den Dateipfaden der beiden Wikis gestartet. Außerdem können einige Webs der Quelle ausgeschlossen werden, wenn nicht alle Daten migriert werden sollen. Ein Beispielbefehl kann also so aussehen:

```
cd /var/www/<neues_Wiki>/tools
perl bulk_copy.pl --xweb System --latest '*.WebStatistics' /var/
www/<aktuelles_wiki>/bin /var/www/<neues_Wiki>/bin
```

Damit werden die Daten aus dem aktuellen Wiki in das neue übertragen. Dieser Vorgang kann je nach Größe des alten Wikis mehrere Stunden in Anspruch nehmen. Ist er abgeschlossen, sind die Daten des aktuellen Wikis im neuen Wiki verfügbar. Zur Sicherheit kann der Docker-Container auch noch neu gestartet werden - eigentlich sollte dies aber nicht nötig sein.

Mit diesen Schritten ist die Datenmigration abgeschlossen. Das neue Foswiki ist nun auf dem Stand des neuesten offiziellen Releaseses von Foswiki. Im Folgenden wird der Upgradeprozess des Foswikis vom offiziellen Release auf die neueste verfügbare Version des Git-Repositorys erläutert.

Upgrade des Foswikis

Zunächst wird eine interaktive Shell innerhalb des Docker-Containers geöffnet. Dazu kann der folgende Befehl verwendet werden.

```
docker exec -it foswiki-foswiki-1 /bin/bash
```

Damit können Befehle innerhalb des Containers ausgeführt werden. Die neueste Version des offiziellen Git-Repositorys [18] von Foswiki wird nun neben das produktive Foswiki geklont. Um den neuesten Foswiki-Quellcode zu erhalten, müssen die dafür benötigten Dateien gezielt in das produktive Foswiki übertragen werden. Dafür werden die folgenden Verzeichnisse aus dem Git-Repository in das Foswiki kopiert, um die Änderungen anzuwenden.

tools/: Hier liegen vorrangig alle Skripte, die zur Instandhaltung nötig sind. Beispielsweise kann hier das Skript `tools/bulk_copy.pl` gefunden werden, welches im letzten Schritt zur Datenmigration verwendet wurde.

locale/: Hier sind alle Übersetzungen des Foswikis zu finden.

lib/: Hier liegen die Perl-Bibliotheken und der produktive Code der Plattform.

tests/: Hier sind Tests zur Überprüfung des Systems zu finden.

Dabei sollen Dateien, die möglicherweise im Ziel existieren, überschrieben werden. Dateien, die noch nicht existieren, sollen erzeugt werden. Dazu kann das Paket `rsync` mit der Option `-av` verwendet werden.

```
cd /var/www/distro/core
rsync -av tools locale test lib /var/www/foswiki/
```

3. Foswiki Upgrade

Danach sollte der Docker-Container neu gestartet werden, um mögliche Fehler zu entdecken. Hierbei kann nicht spezifiziert werden, welche Fehler entstehen können. Es fällt sicherlich auf, wenn die Seite nicht erreichbar ist. Dennoch gibt es keinen reliablen Weg, um kleinere Fehler zu finden. Es kann von Vorteil sein, eine Liste zusammenzutragen, in der mögliche Fehlerquellen beschrieben sind und diese nach jedem Upgrade zu überprüfen. Häufige Fehlerquellen sind zusätzliche Plugins, E-Mail-Funktionalitäten und der Login.

An diesem Punkt ist die Datenmigration von dem aktuellen auf das neue Foswiki vollständig durchgeführt. Zusätzlich wurde ein einfacher Upgrade-Prozess eingerichtet, der innerhalb des Containers mit zwei Befehlen durchgeführt werden kann. Es ist auch denkbar, das Upgrade automatisch durchführen zu lassen. Dabei ist es ratsam, vor jedem Upgrade, ob von Hand oder automatisch, eine Sicherung des Systems durchzuführen.

Im folgenden Abschnitt werden die durchgeführten Schritte bewertet.

3.4 Evaluation der Ergebnisse

Es ist positiv hervorzuheben, dass der gesamte Vorgang der Migration und des Upgrades geradlinig durchzuführen ist. Die Anzahl der durchzuführenden Befehle ist gering und die Häufigkeit von Fehlern überschaubar. Durch die Verwendung von Docker kann der Großteil des Einrichtungsprozesses automatisiert werden. Es muss sich nicht um den Webserver, die Portfreigabe, die Paketabhängigkeiten und die Architektur des zugrundeliegenden Systems gekümmert werden. Das bedeutet eine große Beschleunigung und ein komfortables Entwicklungserlebnis. Auch ist die Einrichtung zugänglicher für Personen, die noch nicht mit Foswiki gearbeitet haben, da viele Schritte automatisch vorgenommen werden. Dennoch gibt es einige Schwächen, die nicht ausgebessert werden konnten.

Wenn das Docker-Image gebaut wird, werden so gut wie alle wichtigen Einstellungen bereits von der Dockerfile getätigt. Es gibt jedoch einige Einstellungen, die nicht durch ein Skript automatisch gesetzt werden können. Dazu zählen vor allem solche, bei denen viele Zeichen verwendet werden, die von einem Computer missverstanden werden können. Beispielsweise gibt es Einstellungen mit Text und Wahrheitswert oder Zeichenkombinationen, die nicht über die Kommandozeile getätigt werden können. Aus diesem Grund können nicht alle Einstellungen automatisch gesetzt werden und müssen nachträglich über das Configure-GUI eingestellt werden. Diese Aufgabe ist schnell erledigt, wenn dieser Container nur ein einziges Mal erstellt wird. Will man aber die Container-Erstellung automatisieren, kann dieser Mangel nicht übergangen werden.

Ebenfalls ist positiv anzumerken, dass die Datenmigration mit dem langsamen Skript `tools/bulk_copy.pl` nur ein einziges Mal durchgeführt werden muss. Durch den Upgradeprozess innerhalb des Docker-Containers müssen die Daten nicht kopiert oder konvertiert werden, was eine ausschlaggebende Zeitersparnis gegenüber dem offiziellen Upgradeprozess bedeutet. Der Upgradeprozess kann direkt auf dem Produktivsystem erfolgen und ist, wenn fehlerfrei, sehr schnell durchzuführen. Eine Sicherungskopie kann mit Docker leicht erstellt werden und es kann problemlos auf die ältere Version zurückgegriffen werden. Die Installation zusätzlicher Plugins ist in den Build-Prozess des Docker-Images eingebaut worden und muss nicht von Hand vorgenommen werden, was ebenfalls viel Zeit beim Upgrade einspart.

Zudem hat die Verwendung des `LdapContribPlugin` den klaren Vorteil, dass nun nicht nur mehr die Authentifizierung, sondern auch die Authorisierung von LDAP übernommen

wird. Im aktuellen Wiki mussten alle Gruppen im Foswiki per Hand angelegt und verwaltet werden, was oft redundant war, da dieselben Gruppen bereits im LDAP-Server existierten. Im neuen Wiki werden diese Gruppen automatisch vom LDAP-Server geladen und können sofort benutzt werden, was die Pflege der Gruppen deutlich einfacher macht.

Ein schwerwiegender Kritikpunkt ist weiterhin, dass es keinen Weg gibt zu überprüfen, ob das System nach dem Upgrade ordnungsgemäß funktioniert. Zwar existieren Versionsnummern auf der Webseite, diese werden aber von den Foswiki-Entwicklern per Hand geschrieben und zeigen immer die letzte offizielle Releasenummer an. Sie sind also keine verlässliche Quelle, um zu überprüfen, ob wirklich der aktualisierte Code verwendet wird. Es gibt zwar die Möglichkeit, die Konfiguration des Wikis mit einer Kommandozeilenoption zu überprüfen. Dazu kann dieser Befehl innerhalb des Containers verwendet werden.

```
cd /var/www/foswiki
tools/configure -check -verbose
```

Dennoch fehlt ein Kontrollmechanismus, der die fehlerfreie Ausführung des Systems zuverlässig testet. Es bleibt also nur eine Überprüfung per Hand, die sicherlich einige Fehler finden kann. Es kann aber nicht garantiert werden, dass das System fehlerfrei funktioniert. Dabei ist zu betonen, dass dies kein Mangel des neuen Upgradeprozesses ist. Auch mit einem Upgrade nach der offiziellen Anleitung [23] fehlt diese Überprüfung.

3.5 Herausforderungen und Lessons Learned

Die größte Herausforderung in der Entwicklung waren die unerwarteten Komplikationen, die auftreten, wenn an einem laufenden System gearbeitet wird. Denn neben der Entwicklung von neuem Quellcode muss bei so einer Aufgabe auf viele kleine Feinheiten geachtet werden, um eine reibungslose Durchführung gewährleisten zu können. Immer wieder gab es Situationen, in denen der Planungsprozess augenscheinlich abgeschlossen war und mit der Durchführung begonnen wurde, nur um dann während der Durchführung mit neuen Sonderfällen konfrontiert zu werden. Im Rahmen dieser Arbeit habe ich gelernt, wie wichtig Tests bei dieser Art von Entwicklung sind. Egal wie sehr man sich mit der Plattform beschäftigt, es kann nie jeder Grenzfall abgedeckt werden. Also muss immer wieder getestet und verbessert werden, um schlussendlich einen zufriedenstellenden Prozess zu erhalten.

Eine weitere Herausforderung war die fehlende Dokumentation von Foswiki selbst. So haben das Foswiki und die meisten Plugins zwar eine gute Dokumentation für die Nutzer, nicht aber für Systemadministratoren. Zur Funktionsweise im Kern eines Plugins lässt sich selten etwas finden. Das macht es gerade unerfahrenen Entwicklern schwer, die Stellen zu finden, an denen Fehler passieren können. Als Beispiele lassen sich die sechs für den Fachbereich wichtigen Plugins anführen. Sie sind alle von der Foswiki-Community entwickelt. Die Dokumentation für Nutzer ist ausführlich und verständlich geschrieben [4], [7], [16], [21], [28], [34]. Wohingegen keinerlei Information über die Funktionsweise des jeweiligen Plugins preisgegeben wird. Das gilt auch für die auf GitHub zu findenden Repositories der Erweiterungen [5], [8], [17], [22], [29], [35].

Bei Software, die unter einer Open-Source-Lizenz veröffentlicht wird, ist eine ausreichende Dokumentation zu erwarten. Daraus habe ich mitgenommen, wie wichtig die Dokumentation von Software ist. Konkret bedeutet dies, dass für dieses Projekt eine ausführliche

3. Foswiki Upgrade

Dokumentation angefertigt wurde [14]. Diese Dokumentation und jede Änderung am Basisprojekt von Timothy Legge [30] wurden in einem Git-Repository des Fachbereichs gespeichert und ebenfalls mit Git versioniert [14]. An den Änderungen im Repository kann gut abgelesen werden, wie viele Anpassungen an der ersten Idee nötig waren, bis der hier beschriebene Upgradeprozess entstanden ist. Zum Überblick führe ich hier eine Auflistung der Dateien des Repositorys an:

```
.
|-- README.md
|-- docker-entrypoint.sh
|-- foswiki.json
|-- iwatch.xml
|-- nginx.default.conf
|-- data
|   |-- System
|   |   |-- NatSkinUserViewTemplate.txt
|   |   +-- NewLdapUserTemplate.txt
|   +-- Main
|       |-- SiteHtmlHead.txt
|       |-- SiteMenu.txt
|       +-- SiteSideBar.txt
|-- pub
|   +-- System
|       +-- ProjectLogos
|           |-- favicon.ico
|           |-- fu-logo.svg
|           |-- new-favicon.ico
|           +-- new-fu-logo.svg
|-- build.log
|-- doc
|   |-- 1_ErsterStart.md
|   |-- 2_LDAP.md
|   |-- 3_SideBar-NavBar-Styling.md
|   |-- 4_UserTopic.md
|   |-- 5_Logo-Favicon.md
|   |-- 6_Datenmigration.md
|   |-- 7_Upgrade.md
|   |-- 8_E-Mail(optional).md
|   |-- 91_group-change.md
|   |-- 9_Troubleshooting.md
|   |-- group-change-org.txt
|   |-- group-change.sed
|   +-- group-change.txt
|-- https
|   |-- docker-foswiki-bodo.crt -> /etc/ssl/certs/twiki.imp.fu-berlin.de.pem
|   |-- docker-foswiki-bodo.key -> /etc/ssl/private/twiki.imp.fu-berlin.de-key.pem
|   |-- docker-foswiki.crt
|   |-- docker-foswiki.key
|   |-- nginx.default.conf.old.conf
|   +-- nginx.default.conf
|-- Dockerfile
+-- docker-compose.yml
```

Jeder Schritt, der zur Einrichtung und zum Upgrade nötig ist, wird in der Dokumentation erklärt. So kann jederzeit nachvollzogen werden, aus welchen Gründen Entscheidungen getroffen wurden und wie das System funktioniert.

Die letzte Herausforderung sind die ungenauen Logs des Foswikis. Um Fehler zu finden, sollten sich in den Logs Informationen finden lassen. Die Logs des Foswikis werden in die Webserver-Logs geschrieben und sind innerhalb des Containers in der Datei `/var/logs/nginx/foswiki-error.log` einsehbar. Aber Plugins schreiben nicht in diese Datei. Einige Erweiterungen beinhalten eine Debug-Einstellung im Configure-GUI. Dies ist beispielsweise beim `SendEmailPlugin` [4] der Fall. Andere Plugins haben diese Debug-Einstellung nicht, dort muss von Hand im Code ein Wahrheitswert geändert werden, um Log-Informationen zu erhalten. So ist es beim `NewUserPlugin` [9]. Es kann sich auszahlen, in so einem Fall den Code selbst zu analysieren, um diese Details herauszufinden. Dennoch sollte bei der Entwicklung einer Software darauf geachtet werden, dass ausreichend Log-Informationen ausgegeben werden können. Dabei muss aus den Logs ersichtlich sein, an welcher Stelle ein Fehler auftritt, um diesen dann beheben zu können.

4 Foswiki Neugestaltung

In diesem Kapitel wird die Neugestaltung der Benutzeroberfläche vollzogen. Dabei wird zeitlich wieder vor dem Upgrade gestartet und die Anforderungen an das zukünftige Foswiki werden erhoben. Aus diesen Anforderungen entstehen Designrichtlinien, welche dann nach dem Upgrade angewendet werden, um die Benutzeroberfläche anzupassen. Außerdem ergeben sich im Laufe der Anforderungserhebung noch Änderungen an der Benutzerführung, die zuletzt eingerichtet werden.

Die Benutzeroberfläche und -führung sind ein maßgeblicher Faktor für die Nutzungsbereitschaft der Fachbereichsgemeinschaft. Im Prozess der Neugestaltung wurde zusätzlich zur Erfüllung der genannten Anforderungen der Parteien ein Fokus auf die Gestaltung des Wikis gelegt, um die Bereitschaft, an den Inhalten mitzuwirken, zu erhöhen.

4.1 Befragung der verschiedenen Parteien

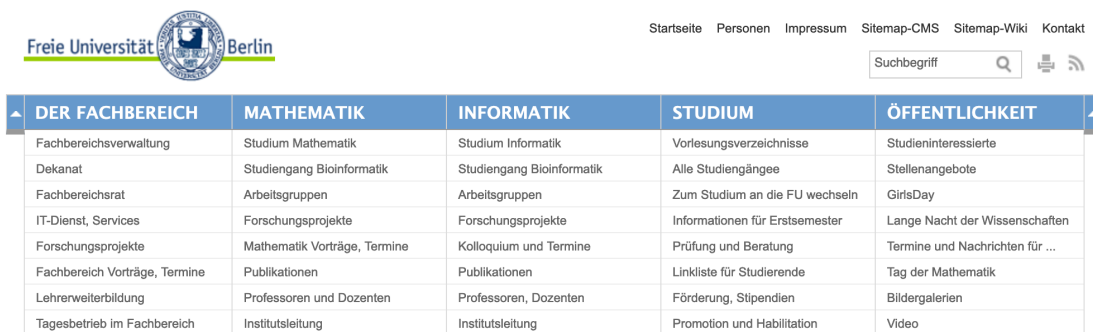
Zur Befragung wurden mehrere Gruppen herangezogen. Zunächst wurde eine kleine Anzahl an Studenten in verschiedenen Semestern freiwillig über das Wiki befragt. Danach gab es freiwillige Gespräche mit Professoren, Mitarbeitern des Prüfungsbüros und der Studienberatung. Die erhobenen Daten wurden selbstverständlich vertraulich und anonym behandelt. Im Folgenden werden die Befragungen ausgewertet und daraus Anforderungen abgeleitet.

Zunächst wurde gerade von den jüngeren Befragten angesprochen, dass das Layout den modernen Anforderungen an das Aussehen einer Webseite nicht gerecht werde. Dazu gehöre im Speziellen das Fehlen eines mobilen Layouts für die Webseite, das das Verwenden des Wikis auf einem Smartphone oder Tablet umständlich gestalten würde. Auch die optische Gestaltung der Webseite wirke nicht mehr zeitgemäß und müsse überarbeitet werden.

Die Mitarbeiter der Studienberatung und des Prüfungsbüros, die regelmäßig mit dem Wiki zu tun haben, merkten an, dass alle Links in der Navigationsleiste des Wikis aus dem Wiki heraus auf andere Webseiten des Fachbereichs führen würden, ohne dies vorher zu

4. Foswiki Neugestaltung

kennzeichnen (siehe Abb. 3). Dieses Phänomen lasse sich auch oft auf den Startseiten der verschiedenen Webs im Wiki finden. Das führe dazu, dass zum einen das Wiki teilweise als überflüssig wahrgenommen werde, zum anderen, dass die Nutzerführung verwirrend sei.



DER FACHBEREICH	MATHEMATIK	INFORMATIK	STUDIUM	ÖFFENTLICHKEIT
Fachbereichsverwaltung	Studium Mathematik	Studium Informatik	Vorlesungsverzeichnisse	Studieninteressierte
Dekanat	Studiengang Bioinformatik	Studiengang Bioinformatik	Alle Studiengänge	Stellenangebote
Fachbereichsrat	Arbeitsgruppen	Arbeitsgruppen	Zum Studium an die FU wechseln	GirlsDay
IT-Dienst, Services	Forschungsprojekte	Forschungsprojekte	Informationen für Erstsemester	Lange Nacht der Wissenschaften
Forschungsprojekte	Mathematik Vorträge, Termine	Kolloquium und Termine	Prüfung und Beratung	Termine und Nachrichten für ...
Fachbereich Vorträge, Termine	Publikationen	Publikationen	Linkliste für Studierende	Tag der Mathematik
Lehrerweiterbildung	Professoren und Dozenten	Professoren, Dozenten	Förderung, Stipendien	Bildergalerien
Tagesbetrieb im Fachbereich	Institutsleitung	Institutsleitung	Promotion und Habilitation	Video

Abb. 3: Navigationsleiste des alten Wikis

Gerade auf Seite der Studenten, inklusive Tutoren und den studentischen Mitarbeitern der Studienberatung, wurde oft angemerkt, dass nicht erkennbar ist, dass die Inhalte durch jeden bearbeitet werden dürfen. Die Wikiseite scheint nicht zu kommunizieren, dass die Bearbeitung durch Studenten erlaubt, sogar erwünscht ist. Das führe dazu, dass die Studenten keine Bearbeitung durchführen, selbst wenn eine Verbesserungsmöglichkeit gefunden wird.

Auch wurde von Professoren, Tutoren und den Mitarbeitern des Studienbüros angegeben, dass ihnen nicht bewusst sei, wozu das Wiki überhaupt verwendet werden soll. Daraus resultiert, dass kaum eine Lehrperson tatsächlich Inhalte auf dem Wiki veröffentlicht bzw. zur Bearbeitung freigibt.

4.2 Ziele und Implementierung

Aus denen in der Befragung abgeleiteten Anforderungen werden in diesem Abschnitt Ziele für die Neugestaltung formuliert. Nach der Formulierung des jeweiligen Ziels werden die erforderlichen Schritte beschrieben, um das Ziel zu erreichen.

Das erste Designziel ist es, das Layout zu verbessern, indem ein moderner Foswiki Skin verwendet wird. Dabei soll sich die Präsentation der Webseite an den anderen Seiten der Universität orientieren und auch dem Corporate Design der Freien Universität entsprechen.

Um dieses Ziel zu erreichen, wurde der Foswiki Skin „NatSkin“ während des Upgrades installiert. Dieser bietet eine solide Grundlage, um ein modernes Design im Stile des Corporate Designs zu entwickeln. Unter anderem bietet der Skin ein mobiles Layout, welches ohne weiteres verwendet werden kann. Bei der ästhetischen Gestaltung ist es besonders hilfreich, dass moderne Webseiten viele weiße Flächen und nur ein oder zwei Primärfarben verwenden. So auch der NatSkin und das Corporate Design der Freien Universität. Aufgrund dieses Umstandes umfasste die Anpassung des Skins CSS-Code [6] Zeilen, die die Primärfarbe, die Schriftart, einige Ränder und Abstände beeinflussen.

Ein weiteres Ziel ist die Verbesserung der Benutzerführung, durch die Reduzierung und Markierung von ausgehenden Links auf dem Wiki. Alle Links aus den Untermenüs der

alten Navigationsleiste wurden auf Wiki Topics umgezogen. Das sorgt dafür, dass die Navigationsleiste aufgeräumter aussieht. Außerdem wurden die Links eindeutig als „externe Links“ markiert, damit diese klar zu erkennen sind (siehe Abbildung 4).

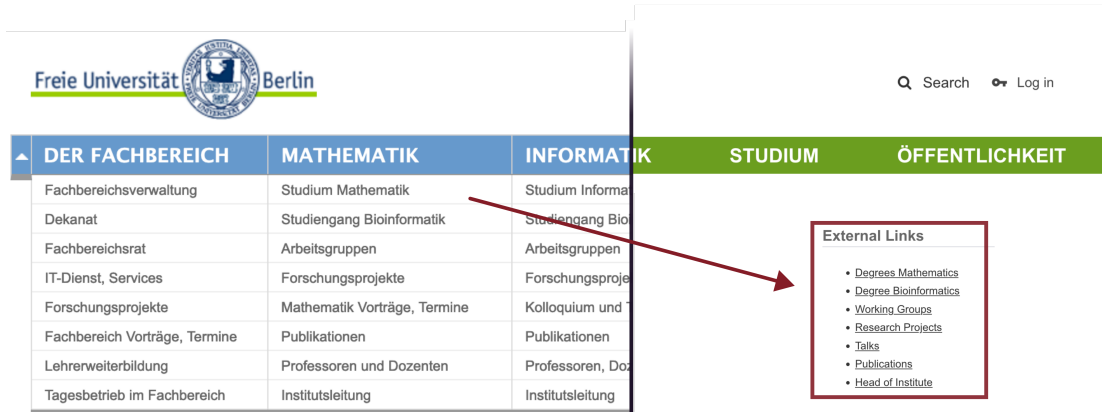


Abb. 4: Vergleich Navigationsleiste

Das letzte Designziel ist es, die Nutzerinteraktion anzuregen, indem klare Hinweise zur Mitarbeit an dem Wiki platziert werden. Dazu wurde ein Banner entwickelt, welches mit einem Foswiki-Makro auf jeder Wikiseite eingebunden werden kann (siehe Abb. 5).

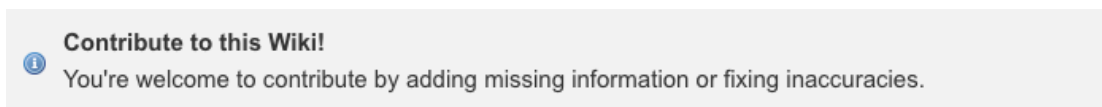


Abb. 5: Banner

Einfluss auf die Fachbereichsgemeinschaft

Studenten werden immer wieder das Wiki besuchen, um Informationen zu erhalten, die relevant für das Studium sind. In Zuge dessen kann die Präsenz eines Banners zu mehr Bearbeitungsbereitschaft führen. Was dadurch aber nicht gelöst wird, ist der Umstand, dass viele Lehrpersonen für das Wiki keine Verwendung sehen. Diese Personengruppe findet sich im Gegensatz zu den Studenten auch relativ wenig auf dem Wiki wieder, was das Fremdeln mit der Plattform begünstigen kann. Mit persönlichen Gesprächen konnte sich diese Einstellung etwas lockern, das erfordert aber viel Öffentlichkeitsarbeit, die im Rahmen dieser Bachelorarbeit nicht geleistet werden konnte.

4.3 Evaluation der Designänderungen

Viele Designziele wurden durch den neuen Skin bereits gelöst. Auch konnten einige Anforderungen durch zusätzlichen CSS-Code erfüllt werden. Dieser Code ist aufgrund der Gegebenheiten der NatSkin-Erweiterung sehr schwer zu lesen, weil viele Elementeigenschaften einzeln überschrieben werden müssen. So gibt es für die Primärfarbe des Skins keine globale Variable, die überschrieben werden kann. Es muss also jede Klasse, die diese Farbe verwendet, einzeln mit einer CSS-Regel überschrieben werden. Davon bekommt

5. Fazit und Ausblick

der Nutzer wenig mit, aber wenn sich wider Erwarten an der Klassenstruktur des NatSkins etwas ändert, kann der CSS-Code unbrauchbar werden. Dies kann einen erhöhten Wartungsaufwand für den Systemadministrator bedeuten.

Einige Umstände konnten nicht gelöst werden. Ich hätte gerne mehr Zeit in die Stärkung der Wikigemeinschaft investiert. Die Gespräche mit einigen Nutzern und das Banner haben eine Grundlage geschaffen, die aber bei Weitem nicht ausreicht. Ich musste viel Zeit in das eigenwillige Upgrade des Foswikis stecken, die mir an dieser Stelle eindeutig gefehlt hat.

Auch das Aussehen des Wikis könnte noch weiter verbessert werden. Beispielsweise wurde der Login-Bildschirm vom Foswiki-Skin nicht verändert und fällt eindeutig aus dem restlichen Layout der Plattform (siehe Abb. 6).

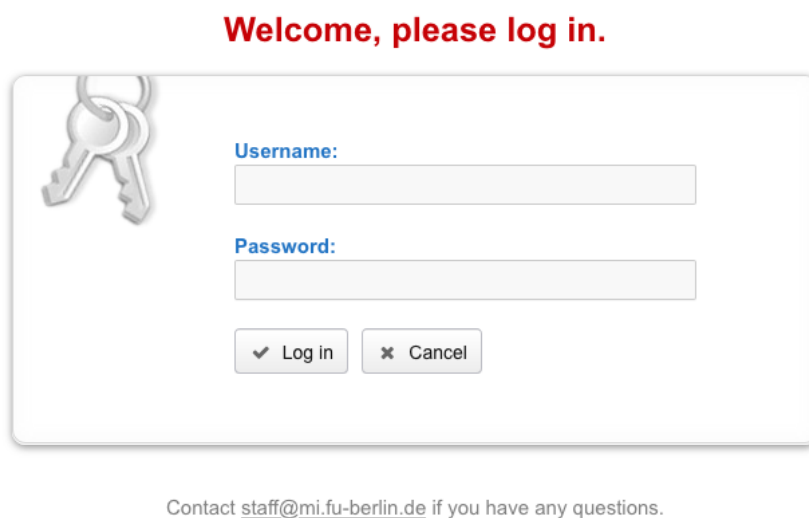


Abb. 6: Login-Bildschirm

5 Fazit und Ausblick

5.1 Zusammenfassung der Arbeit

Im Laufe dieser Arbeit wurde zunächst der Upgradeprozess für das Fachbereichswiki erarbeitet und durchgeführt. Dabei wurde ein individuelles Docker-Image gebaut, aus dem ein Docker-Container entstand, worin das Fachbereichswiki in Zukunft ausgeführt wird. Dann wurde ein auf Git basierender Upgradeprozess erarbeitet, der das Wiki auf dem neuesten Stand des Foswiki-Repositorys hält, ohne, dass auf einen offiziellen Release gewartet werden muss. Basierend auf den Anforderungen verschiedener Parteien am Fachbereich wurde die Benutzeroberfläche und -führung angepasst, um ein besseres Nutzungserlebnis zu bieten und die Nutzungsbereitschaft der Fachbereichsgemeinschaft zu erhöhen.

In diesem Abschnitt möchte ich betonen, dass der beschriebene Upgradeprozess in dieser Arbeit das Ergebnis vieler Iterationen ist. Der Prozess ist keinesfalls von Anfang an so geplant worden, sondern musste unzählige Male angepasst und verbessert werden. Oft, weil erarbeitete Konzepte an der Realität scheiterten, oft aber auch, weil sich die Anforderungen mitten in der Entwicklung geändert haben. So wurde beispielsweise während

der Entwicklung ein neues Corporate Design für den Webauftritt der Freien Universität veröffentlicht, an dem sich meine Designänderungen dann orientierten. Dieses Corporate Design stieß aber an der Universität auf so viel Gegenwind, dass es wieder zurückgenommen wurde. Somit musste ich meine Designänderungen komplett verwerfen und von vorne beginnen. Diese Rückschläge zählen zum Tagesgeschäft eines Systemadministrators und sind keinesfalls ungewöhnlich. Da sie aber so unvorhersehbar sind, kann die Entwicklungszeit leicht unterschätzt werden.

5.2 Beantwortung der Zielsetzungen

Die meisten Hauptziele wurden im Laufe dieser Arbeit erreicht. Der Upgradeprozess ist deutlich einfacher geworden und das Foswiki kann immer auf dem neuesten Stand gehalten werden, ohne auf offizielle Releases warten zu müssen. Falls beim Upgrade etwas schiefgeht, kann zurückgerollt werden, um den Fehler zu beseitigen. Die Neugestaltung hat ebenfalls ihren Zweck erfüllt und es kann nun von einem angenehmeren Nutzererlebnis gesprochen werden, insbesondere auf Mobilgeräten. Damit wurden die genannten Anforderungen der beteiligten Parteien erfüllt. Was die Nutzerinteraktion betrifft, wird sich in Zukunft erst zeigen, ob die Maßnahmen wirken. Durch meine Arbeit am Informatik-Institut sind viele neue Personen auf das Wiki aufmerksam geworden. Wie die Nutzung des Wikis sich in den nächsten Jahren entwickelt, kann aktuell nicht mit Gewissheit prognostiziert werden. Gern hätte ich mehr Zeit in die Neugestaltung und die Gemeinschaft investiert, um die Wikigemeinschaft am Fachbereich zu stärken.

5.3 Ausblick auf zukünftige Trends

Die grundlegende Struktur des Foswikis hat sich in den letzten zehn Jahren kaum verändert. Der erste Beitrag im offiziellen Git-Repository [24] sieht dem aktuellen Aufbau des Foswikis zum Verwechseln ähnlich. Selbstverständlich wurden kontinuierlich Anpassungen von der Foswiki-Gemeinschaft veröffentlicht, diese betreffen aber kaum die Betreiber einer Foswiki-Instanz. Aus diesem Grund kann, wenn auch nicht mit Sicherheit, angenommen werden, dass dies in den nächsten Jahren so bleiben wird. Das bedeutet, dass der in dieser Arbeit entwickelte Upgradeprozess noch einige Zeit in Verwendung bleiben kann.

Ähnlich verhält es sich mit der Neugestaltung des Layouts. Der NatSkin bietet eine gute Grundlage für einen modernen Webauftritt und kann flexibel geändert werden. Wenn sich das Corporate Design in Zukunft ändert, kann das Aussehen durch wenige CSS-Zeilen angepasst werden, ohne zusätzliche Installationen oder Neustarts.

Literatur

- [1] *About Alpine Linux*, Alpine Linux Development Team. Adresse: <https://www.alpinelinux.org/about/> (besucht am 28.07.2023).
- [2] *Alpine Package Keeper*, Alpine Linux Development Team. Adresse: https://wiki.alpinelinux.org/wiki/Alpine_Package_Keeper (besucht am 28.07.2023).
- [3] *Apache Solr*, The Apache Software Foundation. Adresse: <https://solr.apache.org> (besucht am 31.07.2023).

- [4] A. Clemens, *Foswiki SendEmailPlugin*, Arthur Clemens. Adresse: <https://foswiki.org/Extensions/SendEmailPlugin> (besucht am 27.07.2023).
- [5] —, *Foswiki SendEmailPlugin Repository*, Arthur Clemens. Adresse: <https://github.com/foswiki/SendEmailPlugin> (besucht am 02.08.2023).
- [6] *CSS Specification*. Adresse: <https://www.w3.org/TR/css-2022/> (besucht am 25.07.2023).
- [7] M. Daum, *Foswiki LdapContribPlugin*, Foswiki Association e.V. Adresse: <https://foswiki.org/Extensions/LdapContrib> (besucht am 28.07.2023).
- [8] —, *Foswiki LdapContribPlugin Repository*, Foswiki Association e.V. Adresse: <https://github.com/foswiki/LdapContrib> (besucht am 02.08.2023).
- [9] —, *Foswiki NewUserPlugin Repository | Debug Zeile*, Foswiki Association e.V. Adresse: <https://github.com/foswiki/NewUserPlugin/blob/30c6146de445bdedd09fd12f2af71c3ac93d5afc/lib/Foswiki/Plugins/NewUserPlugin.pm#L33C6-L33C6> (besucht am 02.08.2023).
- [10] *Docker Compose*. Adresse: https://docs.docker.com/get-started/08_using_compose/ (besucht am 25.07.2023).
- [11] *Docker Overview*. Adresse: <https://docs.docker.com/get-started/overview/> (besucht am 25.07.2023).
- [12] *Docker Persistent Data*. Adresse: https://docs.docker.com/get-started/05_persisting_data/ (besucht am 25.07.2023).
- [13] A. Ebersbach, M. Glaser und R. Heigl, *WikiTools: Kooperation im Web*. Berlin, Deutschland: Springer, 2005, Kap. 1.
- [14] *Fachbereichswiki Dokumentation*, Nur erreichbar mit Universitätskonto; Neuester Commit-Hash: 2275c966afb332b1ce5ea87faec4d27b8a671c29, FU Informatikinstitut. Adresse: <https://git.imp.fu-berlin.de/it-dienst/foswiki/-/tree/master/doc> (besucht am 02.08.2023).
- [15] *FastCGI Specification*. Adresse: <https://www.mit.edu/~yandros/doc/specs/fcgi-spec.html#S1> (besucht am 25.07.2023).
- [16] *Foswiki BibTeXPlugin*, Foswiki Association e.V. Adresse: <https://foswiki.org/Extensions/BibtexPlugin> (besucht am 25.07.2023).
- [17] *Foswiki BibTeXPlugin Repository*, Foswiki Association e.V. Adresse: <https://github.com/foswiki/BibtexPlugin> (besucht am 02.08.2023).
- [18] *Foswiki Github Repository*, Foswiki Association e.V. Adresse: <https://github.com/foswiki/distro> (besucht am 27.07.2023).
- [19] *Foswiki Homepage*. Adresse: <https://foswiki.org/Home/WebHome>.
- [20] *Foswiki Installation Guide*, Foswiki Association e.V. Adresse: <https://foswiki.org/System/InstallationGuide> (besucht am 25.07.2023).
- [21] *Foswiki LatexModePlugin*, Foswiki Association e.V. Adresse: <https://foswiki.org/Extensions/LatexModePlugin> (besucht am 27.07.2023).
- [22] *Foswiki LatexModePlugin Repository*, Foswiki Association e.V. Adresse: <https://github.com/foswiki/LatexModePlugin> (besucht am 02.08.2023).
- [23] *Foswiki Upgrade Guide*, Foswiki Association e.V. Adresse: <https://foswiki.org/System/UpgradeGuide#Installation> (besucht am 25.07.2023).
- [24] *Foswiki Github Repository (First Commit)*, Foswiki Association e.V. Adresse: <https://github.com/foswiki/distro/tree/9ce54f2fa6f23f937cf93900776143d999f92ece/core> (besucht am 30.08.2023).

- [25] *Git Distributed Workflows*. Adresse: <https://git-scm.com/book/en/v2/Distributed-Git-Distributed-Workflows> (besucht am 25.07.2023).
- [26] *HTML Background*. Adresse: <https://html.spec.whatwg.org/dev/introduction.html#background> (besucht am 25.07.2023).
- [27] S. F. Klutentreter, „Die Migration eines Wiki-Systems von TWiki nach Foswiki,“ Bachelorarb., Freie Universität Berlin, Berlin, Deutschland, 2013.
- [28] K. Larvsen, P. Thoeny und T. Cyrus, *Foswiki ChartPlugin*, Foswiki Association e.V. Adresse: <https://foswiki.org/Extensions/ChartPlugin> (besucht am 25.07.2023).
- [29] —, *Foswiki ChartPlugin Repository*, GitHub, Inc. Adresse: <https://github.com/foswiki/ChartPlugin> (besucht am 02.08.2023).
- [30] T. Legge (@timlegge), *Dockerized Foswiki Github Project*. Adresse: <https://github.com/timlegge/docker-foswiki> (besucht am 28.07.2023).
- [31] M. Notari und B. D. Honegger, *Der Wiki-Weg des Lernens*. Bern, Schweiz: hep, 2013, Kap. 1.
- [32] *Perl Documentation*. Adresse: <https://perl.developpez.com/documentations/en/5.20.0/perlintro.html#What-is-Perl?> (besucht am 25.07.2023).
- [33] J. Sermersheim, *Lightweight Directory Access Protocol (LDAP): The Protocol*, RFC 4511, Juni 2006. DOI: [10.17487/RFC4511](https://doi.org/10.17487/RFC4511). Adresse: <https://www.rfc-editor.org/info/rfc4511>.
- [34] A. Stabini, *Foswiki CalendarPlugin*, Foswiki Association e.V. Adresse: <https://foswiki.org/Extensions/CalendarPlugin> (besucht am 27.07.2023).
- [35] —, *Foswiki CalendarPlugin Repository*, Foswiki Association e.V. Adresse: <https://github.com/foswiki/CalendarPlugin> (besucht am 02.08.2023).
- [36] *Usage statistics of markup languages for websites*, W3Techs. Adresse: https://w3techs.com/technologies/overview/markup_language (besucht am 30.08.2023).
- [37] *Usage statistics of web servers*, W3Techs. Adresse: https://w3techs.com/technologies/overview/web_server (besucht am 30.08.2023).
- [38] *What is a Container*. Adresse: <https://www.docker.com/resources/what-container/> (besucht am 25.07.2023).
- [39] *What is a virtual Machine (VM)*. Adresse: <https://www.redhat.com/en/topics/virtualization/what-is-a-virtual-machine> (besucht am 25.07.2023).
- [40] *What is a web server?* Adresse: https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Web_mechanics/What_is_a_web_server (besucht am 25.07.2023).
- [41] *What is Markdown*. Adresse: <https://daringfireball.net/projects/markdown/> (besucht am 25.07.2023).