



UNIVERSITÄT
LEIPZIG

Wintersemester 2018/19
Softwaretechnikpraktikum

App zur Inventarisierung von Unternehmenswerten

Recherchebericht

Gruppe: ak18b

Betreuer: Benjamin Lucas Friedland, Michael Fritz

Gruppenmitglieder: Alexander Zwisler, Leon Kamuf, Leon Rudolph, Maurice Eisenblätter,
Maximilian Gläfcke, Robin Seidel, Sina Opitz, Steve Woywod

Inhaltsverzeichnis

1	Begriffe	1
1.1	Inventarisierung	1
1.2	Inventar	1
1.3	Abschreibung	1
1.4	Dachgesellschaft	1
1.5	Git	1
1.6	Continuous Integration	2
1.7	HTML	2
1.8	CSS	2
1.9	JavaScript	2
1.10	Progressive Web App	2
1.11	Modultests	2
1.12	Framework	2
1.13	Deployment	3
1.14	Front-End	3
1.15	Back-End	3
2	Aspekte	3
2.1	GitLab	3
2.2	GitLab CI	3
2.3	Gitflow	4
2.4	Slack	4
2.5	Jekyll	4
2.6	Maven	4
3	Konzepte	5
3.1	Anmeldesystem	5
3.2	Formular zur Objekterstellung und -bearbeitung	5
3.3	Liste mit Suchfunktion	5
3.4	Unternehmensverknüpfung	5

1 Begriffe

1.1 Inventarisierung

Unter Inventarisierung versteht man die Bestandsaufnahme von unternehmensspezifischen Objekten in Hinsicht auf spezielle Merkmale selbiger. Sie bildet die Informationsgrundlage für Kosten- und Leistungsrechnungen sowie zur Ermittlung von einem potentiellen Reinvestitionsbedarf.[1]

1.2 Inventar

Als Inventar versteht man die Gesamtheit der zu einem Betrieb/Unternehmen/Haus/etc. gehörenden Einrichtungsgegenstände und Vermögenswerten, des Weiteren werden auch die offenen Schulden berücksichtigt. Zudem muss das Inventar jährlich inventarisiert werden.[2]

1.3 Abschreibung

Im betrieblichen Rechnungswesen bezeichnet man die planmäßige oder auch außerplanmäßige Wertminderung von Unternehmenswerten als Abschreibung.[3]

1.4 Dachgesellschaft

Eine Dachgesellschaft, unter anderem auch als Holdinggesellschaft bezeichnet, beschreibt eine Beteiligungsgesellschaft in Form von Muttergesellschaft und Tochterunternehmen. Als Dachgesellschaft wird in erster Linie ein Organisationsmittel für Firmen bezeichnet. Genutzt wird sie, um Konzerne zu strukturieren und Unternehmen insgesamt straffer aufzustellen.

1.5 Git

Git ist eine Software, dessen Entwicklung von Linus Torvalds angeregt wurde und dient zur verteilten Versionsverwaltung. Gegenüber anderen Versionskontrollsystemen unterscheidet es sich in vielerlei Aspekten.

1. Nicht lineare Entwicklung: Dies bedeutet, dass jeder Entwickler sein eigenes Repository hat und jedes Feature in seinem eigenen Branch entwickelt wird. Wichtige Aspekte für diese Vorgehensweise sind "Branching", was die Erstellung neuer Entwicklungszweige beschreibt und Merging", was die Verschmelzung mindestens zweier dieser Entwicklungszweige definiert. Darüber hinaus sind Tools zur Visualisierung der linearen Historie bereits implementiert.
2. Kein zentraler Server: Jeder Nutzer hat eine lokale Kopie des gesamten Repositories und der entsprechenden Versionshistorie. Dieses Feature beinhaltet zwei grundlegende Aspekte. Zum einen bietet es so eine hohe Datensicherheit und zum anderen benötigt man so für die meisten Tätigkeiten keinen Netzwerkzugriff.
3. Kryptografische Sicherheit der Projekthistorie: Die Hash-Codes einzelner Commits sind so konzipiert, dass sie immer auf der gesamten Historie basieren. Dieses Verfahren schließt eine nachträglich Änderung, ohne den Hash-Code zu ändern, aus.
4. Unterstützung mehrerer Übertragungsprotokolle: Durch die Unterstützung vieler Übertragungsprotokolle wie HTTP, HTTPS, FTP etc., bietet sich dem Nutzer eine große Vielfalt an Möglichkeiten seine Daten zu übermitteln.
5. Säubern des Repositories: Sollte man aus dem Repository Inhalte löschen, so bleiben sie dennoch vorläufig hinterlegt. Zu einem späteren Zeitpunkt kann man dann entscheiden, ob man die Inhalte wiederherstellen möchte oder ob man sie gänzlich löscht. Dies macht es nahezu unmöglich, ungewollt Inhalte dauerhaft zu löschen.

[4]

1.6 Continuous Integration

Continuous Integration, kurz 'CI', beschreibt den Prozess des Fortlaufenden Zusammenfügens von Komponenten zu einer einzigen Anwendung. Die gängigsten Aktionen sind das Kompilieren und Verlinken der Komponenten, jedoch sind auch beliebige andere Operationen zur Erzeugung abgeleiteter Informationen durchführbar. Im Normalfall wird dafür kein neues Gesamtsystem gebaut, sondern automatisierte Tests durchgeführt und Softwaremetriken zur Messung der Softwarequalität erstellt. Dieser Vorgang wird durch die Einbindung in das Versionsverwaltungssystem (Git) automatisch realisiert.

1.7 HTML

HTML ist die Abkürzung für die Standardsprache zur Erstellung von Webseiten und Webapplikationen **H**yper-**T**ext-**M**arkup-**L**anguage. Neben der Erstellung von den oben genannten Sachverhalten dient HTML so außerdem der Strukturierung von Webinhalten. Webbrowser erhalten HTML-Dokumente von einem Webserver oder von einem lokalen Speicher und erzeugen daraus multimediale Webseiten.[5]

1.8 CSS

CSS bedeutet ausgeschrieben **C**ascading **S**tyle **S**heet und dient zur Gestaltung für unterschiedliche Dokumentenarten. Hauptsächlich wird es in Verbindung mit den Auszeichnungssprachen HTML und XML verwendet.

1.9 JavaScript

Dies ist ein weiteres essentielles Tool zur Bearbeitung von Webseiten. Es ermöglicht Benutzerinteraktionen auszuwerten, Inhalte zu verändern, nachzuladen oder zu generieren und vermag somit die Erweiterung der Möglichkeiten von HTML und CSS.[6]

1.10 Progressive Web App

Eine Progressive Web App, kurz PWA, ist eine Webseite, welche zahlreiche Merkmale besitzt, die bislang nur nativen Apps vorbehalten waren. Grundsätzlich kann man darunter eine Symbiose aus einer responsiven Webseite und einer App verstehen. Man kann sie, genau wie eine normale Webseite, mit HTML, CSS und JavaScript, konzipieren und realisieren.

1.11 Modultests

Modultests sind ein fester Bestandteil von Softwaretests. Sie beschreiben die Überprüfung auf korrekte Funktionalität von Softwaremodulen. Zur Prüfung müssen im Vorfeld möglichst umfassende Eingabeparameter und deren erwartete Ausgabeparameter festgelegt werden. Mit diesen Werten können dann die entsprechenden Module auf ihre Funktionalität getestet werden.

1.12 Framework

Frameworks bieten ein Grundgerüst für die Programmierung. Sie beinhalten verschiedene Funktionen, die der Entwickler nicht jedes mal neu programmieren muss. Außerdem bieten sie Schnittstellen um Quellen, wie beispielsweise Datenbanken, einfacher anzusprechen. Dadurch liefern Frameworks eine große Zeitersparnis während der Programmierung. Jedoch muss man darauf achten passende Frameworks zu wählen, welche nicht unnötig viele unbrauchbare Funktionen beinhalten, um das Programm nicht künstlich langsamer zu machen.

1.13 Deployment

Dies ist der Fachbegriff für Softwareverteilung. Man bezeichnet damit die halb- oder vollautomatisierten Prozesse zur Installation und Konfiguration von Software auf PCs und Servern. Typischerweise wird dieses Konzept in Unternehmen eingeführt.

1.14 Front-End

Das Front-End wird im Allgemeinen als die Oberfläche bezeichnet, welche der User im Endeffekt sieht beziehungsweise auf welcher er Eingaben tätigen kann. Zum Front-End gehören alle sichtbaren Elemente einer Oberfläche, wie zum Beispiel Bilder, Texte oder das Design.

1.15 Back-End

Im Gegensatz zum Front-End, bekommt der User das Back-End nicht zu sehen. Im Back-End werden jegliche Funktionen, sowie die Gestaltung des Front-Ends, definiert. Das heißt alle Inhalte des späteren Front-Ends, sowohl als auch eventuelle Datenbanken, werden im Back-End eingebunden.

2 Aspekte

2.1 GitLab

GitLab ist eine Webanwendung zur Versionsverwaltung für Softwareprojekte auf Git-Basis. Es bietet mehrere nützliche Features für das Programmieren in Teams, wie Features für Management und Bug-Tracking etc., außerdem beinhaltet es ein System für kontinuierliche Integration. Einige der wichtigsten Management-Tools sind das Issue-Board, Issues an sich, mit welchen man spezifischen Personen Aufgaben zuteilen kann, Milestones als Zusammenfassung von Issues und Time-Tracking. Die Entwicklung ist ebenfalls sehr angenehm durch Features, wie die visualisierte Repräsentation von Branches, Merge Requests etc. Des Weiteren lässt sich der Arbeitsablauf klar definieren, zum Beispiel kann die Push-Funktion auf den Master-Branch gesperrt werden. Somit verringert man die Fehlerquote und jeder kann sich auf seine eigentlichen Aufgaben konzentrieren, ohne sich mit dem Projektmanagement auseinander zu setzen.

2.2 GitLab CI

GitLab CI ist ein bereits implementiertes, open-source Feature im GitLab. Dadurch, dass dieses Feature bereits von Beginn an zur Verfügung steht, bietet es sich als CI-Service für das gesamte Projekt an.

GitLab-CI unterstützt unter anderem folgendes:

- Parallele Build-Prozesse
- einfache Konfigurierbarkeit über `.gitlab-ci.yml`
- Echtzeit Logging

Des Weiteren bietet GitLab-Ci folgende Vorteile:

- Frühzeitige Erkennung von Fehlern
- Integrationsprobleme werden laufend entdeckt
- Verfügbarkeit eines Systems für Demo, Test oder Vertriebszwecke

2.3 Gitflow

Gitflow ist ein konzeptioneller Arbeitsablauf mit Git. Es wird dabei die Historie gut strukturiert dargestellt und ist damit selbst für Außenstehende gut ersichtlich. Das Konzept basiert darauf, zwei Hauptverzweigungen zu erzeugen. Eine dieser Verzweigungen ist der Master-Branch, die andere der Develop-Branch, dazu kommen noch kleinere Feature- und Release-Branches. Auf den Master-Branch werden nur fertige Release aus dem Release-Branch commitet. Der Develop-Branch dient zur Entwicklung der einzelnen Release, dafür werden die einzelnen Feature-Branches gemerget. Feature-Branches sind kleine Zweige, welche für jedes einzelne zu erstellende Feature erstellt werden. Man eröffnet einen Release-Branch immer wenn ein Release ansteht und im Develop-Branch genügend Features implementiert sind. Im Release-Branch werden keine Features mehr hinzugefügt, sondern nur Bug-Fixes vorgenommen oder Dokumentationen angefügt. Dieses Vorgehen ist essentiell für ein strukturiertes und sicheres Arbeiten.

2.4 Slack

Slack ist ein webbasierter Instant-Messaging Dienst und dient zur teaminternen Kommunikation. Dieser Messenger erlaubt das Chatten mit Einzelpersonen oder Gruppen sowie das Austauschen von Dokumenten. Dieses Feature allein macht ihn nicht besonders, da andere Messenger, wie Facebook, WhatsApp oder Discord, das Selbe können. Was Slack jedoch besonders macht ist die hohe Konfigurierbarkeit mittels externer Apps (zum Beispiel Polly), wodurch man ihn den Anforderungen der Gruppe anpassen kann. Weiterhin ist es möglich, dass man Eventnachrichten von externen Diensten (zum Beispiel GitLab) direkt an die Slack-Gruppe senden kann. Dies erlaubt Slack als zentrales Informations-Hub zu fungieren und für jedes Gruppenmitglied eine schnelle Übersichtsmöglichkeit zu schaffen. Zusammengefasst kann man sagen, dass Slack ein mächtiges Kommunikationswerkzeug ist, vor allem in Bezug auf Arbeiten in Teams und sich hervorragend für einen organisierten und strukturierten Projektablauf eignet.

2.5 Jekyll

Jekyll ist ein in der Skriptsprache 'Ruby' geschriebenes Programm, welches statische HTML-Webseiten, mit Hilfe einfacher Textdateien (zum Beispiel Markdown), generiert. Jekyll findet vor allem Anwendung bei Blogs oder kleineren Webseiten. Die Vorteile sind:

- einfache Installation und Verwendbarkeit
- Sicherheit
- weite Verbreitung, zum Beispiel bei GitHub-Pages
- benötigt keine weiteren Dienste, wie beispielsweise einen Datenbankserver
- leichte Erweiterbarkeit und Wartung

Diese Vorteile zeigen, dass Jekyll für kleine bis mittelgroße Projekte sehr gut geeignet ist.

2.6 Maven

Maven ist ein Build-Management Tool, mit dem die Standardisierung von Java Programmen realisiert werden soll. Es wird beispielsweise definiert wie `.java` Dateien zu `.class` Dateien kompiliert und anschließend zu `.jar` gepackt werden, welche pre-/post-processing Schritte durchgeführt werden oder wie der CLASSPATH geregelt ist. Das angestrebte Ziel von Maven ist die größtmögliche Automatisierung von Schritten im Build-Lifecycle. Der Build-Lifecycle ist der Weg von der Validierung des Codes bis hin zum Deployment. Es müssen hauptsächlich nur die vom konventionellen Build-Prozess abweichenden Ideen festgehalten, beziehungsweise definiert werden. Des Weiteren versucht Maven unter sich abgeschlossen zu bleiben, das heißt, das automatische

Downloaden von Updates der Bibliotheken und Maven-Plugins. Somit werden keine weiteren Tools oder Scripts benötigt.

3 Konzepte

3.1 Anmeldesystem

Ein Anmeldesystem verwaltet den Zugriff auf die Objekte. So wird sichergestellt, dass Objekte nur von autorisierten Personen erstellt, bearbeitet und gelöscht werden können (Rechteverwaltung). Zudem lassen sich so Mitarbeiter verschiedener Firmen unterscheiden und Funktionen zur Verwaltung können bereitgestellt werden. So können verschiedene Benutzergruppen erstellt werden, die unterschiedliche Rechte besitzen (Personengruppe X kann bspw. alle Objekte sehen und ändern, Personengruppe Y nur alle Objekte sehen, Personengruppe Z nur bestimmte Objekte sehen, etc.).

3.2 Formular zur Objekterstellung und -bearbeitung

Um eine optimale Konfigurierbarkeit zu gewährleisten, wird es für jedes Objekt, das inventarisiert werden soll, ein simples Formular geben. Dieses besteht aus Grundvoraussetzungen (Pflichtfelder) und individuell anpassbaren Feldern. Mithilfe der individuellen Feldern können neue Werte, die das Objekt näher beschreiben, hinzugefügt werden und somit kann jedes Unternehmen alle möglichen Objekte ohne Probleme inventarisieren. Auch das Bearbeiten oder nachträgliches Hinzufügen von Attributen der Objekte wird über ein simples Formular gehandhabt.

3.3 Liste mit Suchfunktion

Objekte der Inventarisierung werden übersichtlich in einer Liste dargestellt, die von bestimmten Personengruppen verwaltet werden kann. Es soll zudem ein Suchfeld geben, mit dem nach bestimmten Objekten innerhalb der Liste gesucht werden kann.

3.4 Unternehmensverknüpfung

Mutter- und Tochterunternehmen sind hierarchisch miteinander verbunden, sodass die Tochterunternehmen Objekte erben. Trotzdem können untergeordnete Unternehmen unabhängig voneinander agieren.

Literaturverzeichnis

- [1] Inventarisierung. <https://de.wikipedia.org/wiki/Inventarisierung>. Besucht: 10.11.2018.
- [2] Inventar. <https://www.duden.de/rechtschreibung/Inventar>. Besucht: 10.11.2018.
- [3] Abschreibung. <https://wirtschaftslexikon.gabler.de/definition/abschreibung-31208>. Besucht: 10.11.2018.
- [4] Git. <https://en.wikipedia.org/wiki/Git>. Besucht: 10.11.2018.
- [5] Html. <https://en.wikipedia.org/wiki/HTML>. besucht: 10.11.2018.
- [6] Javascript. <https://de.wikipedia.org/wiki/JavaScript>. Besucht: 10.11.2018.