Praktikumsbericht

über das Praktikum

bei

**valantic FSA**

vom

**10.01.2022**

bis zum

**31.12.2024**

von

**Janik Teune**

**Matrikelnummer: 233966**

Inhaltsverzeichnis

[1 Einleitung 2](#_Toc184752003)

[1.1 Motivation 2](#_Toc184752004)

[1.2 Tätigkeitsübersicht 2](#_Toc184752005)

[2 Hauptteil 2](#_Toc184752006)

[2.1 Unternehmensprofil 2](#_Toc184752007)

[2.1.1 Eckdaten 2](#_Toc184752008)

[2.2 Aufgabenbereiche 3](#_Toc184752009)

[2.3 Speichern von Log und Konfigurationsdateien 4](#_Toc184752010)

[2.4 Der „Rangeslider“ 4](#_Toc184752011)

[2.5 Git-Migration der „User Guides“ 4](#_Toc184752012)

[2.6 AppInvoker Simulator 5](#_Toc184752013)

[2.7 Arbeitsbedingungen 6](#_Toc184752014)

[3 Fazit 6](#_Toc184752015)

[3.1 Vergleich der Werksstudententätigkeit zum Berufspraktikum 6](#_Toc184752016)

[3.1.1 Bearbeitung einer komplexen Aufgabenstellung mit Arbeitsverfahren, -mitteln und -prozessen des jeweiligen Fachgebietes 7](#_Toc184752017)

[3.1.2 Bekanntmachung mit organisatorischen und sozialen Verhältnisse der Praxis 7](#_Toc184752018)

[3.1.3 Fördern der Fähigkeit zur Teamarbeit 7](#_Toc184752019)

[3.1.4 Erwerb von Kenntnissen über Betriebsorganisation, Solzialstruktur, Sicherheits- und Wirtschaftlichkeitsaspekte 7](#_Toc184752020)

[3.2 8](#_Toc184752021)

[4 Literaturverzeichnis 9](#_Toc184752022)

# Einleitung

## Motivation

Das Schreiben von Programmen hat mich schon lange interessiert, doch während der ersten Semester konnte ich an der Universität nur wenig praktische Erfahrung sammeln. Die Module „Einführung in die Informatik“ und „Algorithmen und Datenstrukturen“ konnten grundlegende Konzepte des Programmierens anhand der Programmiersprache Java vermitteln. Die dort vorgestellten Probleme und Übungsaufgaben konnten allerdings in wenigen Zeilen Code gelöst werden. Das Arbeiten an einem größeren Projekt und in einem Team habe ich im Studium nur wenig kennengelernt.

Außerdem wurden eher grundlegende Strukturen der Programmiersprache Java vermittelt. Moderne und neue Ideen habe ich kaum kennengelernt.

Von meinem Arbeitsplatz bei valantic FSA erhoffte ich mir moderne Konzepte beim Programmieren kennenzulernen, sowie die Arbeit an einem großen Projekt zu erfahren.

Das Unternehmen habe ich durch das Sponsoring des Programmierwettbewerbs kennengelernt, welcher im Rahmen des Moduls „Algorithmen und Datenstrukturen“ stattfand. Das freundliche Auftreten der Mitarbeitenden sowie die Empfehlung eines Kommilitonen überzeugten mich schließlich, mich bei valantic FSA zu bewerben. Nach einem erfolgreichen Bewerbungsgespräch konnte ich am 10.01.2022 meinen ersten Arbeitstag und meine sechsmonatige Probezeit in der Abteilung iQbonds beginnen.

## Tätigkeitsübersicht

Zu meinen Aufgaben gehört:

* Spezifikation des Verhaltens neuer Anforderungen
* Implementation dieser Spezifikationen
* Fixen von Bugs
* Code Reviews
* Migration alter Projekte
  + Von SVN zu Git
  + Von Ant zu Gradle
  + Von älteren Java Versionen zu Java 21

# Hauptteil

## Unternehmensprofil

### Eckdaten

Die valantic GmbH ist ein Unternehmen, welches Beratungsdienstleistungen und Software in verschiedenen Branchen anbietet. Auf der Webseite sind unter anderem die Baubranche, Chemische und Pharmazeutische Industrie, das Finanzwesen, die Konsumgüterindustrie und die Kulturwirtschaft vermerkt. (1; 2)

valantic Financial Services Automation (kurz valantic FSA) ist ein Privatunternehmen, das zur valantic GmbH gehört und Finanzdienstleistungen anbietet. Es beschäftigt über 300 Mitarbeitende in 9 Büros in Europa und Nordamerika. Das Büro in Magdeburg befindet sich auf dem Werder in der Mittelstraße 10. Der Hauptsitz des Unternehmens befindet sich in Frankfurt am Main. (3; 4)

valantic FSA beschäftigt sich mit der Automatisierung von Prozessen in der Finanzindustrie. Die angebotene Software richtet sich an Banken. Das Ziel der Firma ist es, die Werteströme ihrer Kunden zu digitalisieren und weiterzuentwickeln. Damit will valantic FSA die Effektivität und Agilität verbessern sowie das Verständnis dafür steigern. (4; 5)

## Aufgabenbereiche

Ich arbeite in der Abteilung an verschiedenen Projekten, selten helfe ich auch noch in anderen Abteilungen aus.

Einen großen Teil meiner Arbeitszeit verbringe ich am „Diagnosetool“, welches vor allem von der Test-Abteilung verwendet wird. Die Mitarbeitenden können das Tool nutzen, um verschiedene Performance-Counter aus Log-Dateien auszulesen und sich diese tabellarisch und graphisch anzeigen zu lassen.

Häufig setze ich auch neue Anforderungen der Kunden um, zum Beispiel im iQbonds -client. Während ich als einziger Entwickler am Diagnosetool arbeite, handelt es sich beim iQbonds-client um ein großes Projekt, an dem seit vielen Jahren von vielen verschiedenen Mitarbeitenden gearbeitet wird. Gerade hier ist es notwendig, das gewünschte Verhalten eindeutig zu spezifizieren. Die Spezifikation wird vor Beginn der Implementation abgeschlossen und kann dann aus verschiedenen Perspektiven verifiziert werden. Unser Dokumentations-Team überprüft das Dokument auf sprachlicher Ebene, ein weiterer Entwickler verifiziert die technischen Aspekte und das Projektmanagement überprüft, ob das spezifizierte mit dem gewünschten Verhalten übereinstimmt. Während implementiert wird, kann jemand aus der Test-Abteilung anhand der Spezifikation diverse Testfälle ableiten, um das Verhalten des Programms zu verifizieren. Durch diese Vorgehensweise werden viele Bugs früh erkannt und das Verhalten ist eindeutig dokumentiert.

Nach dem Abschluss einer Implementation oder dem Fix eines Bugs werden die Änderungen in Gitlab reviewt. Das wird meist von meinem Ansprechpartner oder einem anderen Werksstudenten gemacht. Auch ich reviewe regelmäßig den Code von anderen Mitarbeitenden.

## Speichern von Log und Konfigurationsdateien

Im iQbonds-Client hatte der Kunde die Möglichkeit, die Konfiguration seiner Client-Instanz in ein ZIP-Archiv zu schreiben und dieses zu speichern. Damit kann zum Beispiel der Zustand dieser Instanz besser nachvollzogen und das Nachstellen eines Bugs erleichtert werden.

Diese Funktion habe ich erweitert. Mit der java-swing-API habe ich einen Dialog gebaut, wo der Kunde mithilfe von Kontrollkästchen auswählen kann, welche Dateien in ein ZIP-Archiv hinzugefügt werden.

Die Anforderung habe ich zunächst spezifiziert, sodass die Testabteilung diverse Blackbox-Tests vorbereiten und das spezifizierte Verhalten vom Projektmanagement bestätigt werden kann und danach implementiert.

Der Kunde hat nun die Optionen

* Die Konfigurationsdateien des Clients zu speichern
* Die Log-Dateien des Clients zu speichern
* Ältere archivierte Log-Dateien zu speichern
* Die Konfigurationsdateien des iQbonds-Servers anzufragen und diese ebenfalls zu speichern

Das Speichern der Dateien des Clients war einfacher, da alle Daten zum Speichern der Dateien bereits vorliegen. Zur Speicherung der Serverkonfiguration muss allerdings eine Nachricht an den Servergesendet werden. Danach muss auf die Antwortnachricht des Servers gewartet, diese Nachricht ausgewertet und transformiert werden und schließlich im XML-Format der Konfigurationsdateien gespeichert werden.

Zusätzlich habe ich im Zuge dieser Anforderung den FileChooser-Dialog der java-swing-API erweitert. Die Erweiterung validiert den eingegebenen Dateinamen und kann eine Auswahl an Fehlermeldungen anzeigen. Sollte der Dateiname invalide sein, so wird der Bestätigungsknopf deaktiviert. (6)

## Der „Rangeslider“

Wenn das Diagnosetool Log-Dateien importiert, kann man einen Start- und Endzeitpunkt für die Auswertung der Counter-Daten auswählen. Für diese Auswahl habe ich einen „Rangeslider“ implementiert. Der Rangeslider ist funktioniert genauso wie der JSlider aus der java-swing-API. Der Unterschied ist, dass es 2 Knöpfe auf dem Slider gibt. Die Knöpfe können nicht aneinander vorbeigezogen werden. So kann der linke Knopf zur Auswahl der Startzeit und der rechte Knopf zur Auswahl der Endzeit verwendet werden. Die Herausforderung bei dieser Aufgabe lag bei der Darstellung des zweiten Knopfs. (7)

## Git-Migration der „User Guides“

Das Dokumentationsteam hat für jedes Produkt nochmal ein eigenes Projekt „User Guide“. In diesen Projekten arbeitet das Dokumentationsteam mit einem externen Tool, um den Kunden in verschiedenen Varianten (PDF, HTML5, …) einen umfassenden Guide zu dem entsprechenden Produkt in Deutsch und Englisch zur Verfügung zu stellen.

Zur Versionsverwaltung der User Guide Projekte wurde Subversion (kurz: SVN) verwendet.

Nun war es meine Aufgabe jedes einzelne dieser Projekte von SVN zu Git zu migrieren. Dafür wurde mir ein Tool „svntogit“ zur Verfügung gestellt, welches mir viel Arbeit abnehmen konnte.

Der grobe Ablauf für die Migration eines Projektes ist der folgende:

1. Svntogit ausführen
   1. Den alten Projektpfad angeben
   2. Den neuen Projektpfad angeben
   3. Die Ausführung dauert mehrere Stunden
   4. Das Ergebnis ist ein Git-Repository mit dem Namen „repo“
   5. Alle Commits, Branches und Tags aus dem alten SVN-Repository wurden übernommen
2. Das neue Git-Repository dem Projekt entsprechend umbenennen
3. Die .gitignore und .gitattributes Dateien erstellen
4. Das Projekt auf Gitlab anlegen und das lokale Repository hochladen
5. Die Pipeline aufsetzen
6. Auf dem alten SVN-Repository markieren, dass dieses Projekt migriert wurde

Da es sehr viele User Guide Projekte gibt, habe ich die Projekte nicht alle händisch migriert. Stattdessen habe ich ein Batch-Skript geschrieben, was anhand einiger Parameter die Schritte der Reihe nach für mich durchführt.

Bei der Aufgabe habe ich mich mit Windows-Batch-Skripten, Subversions und Gitlabs Kommandozeileninterface und mit der Git-Pipeline auseinandergesetzt. Außerdem habe ich das Dokumentationsteam im Umgang mit Git und Gitlab geschult.

## AppInvoker Simulator

Eine andere Anforderung, die ich für unsere Testabteilung umgesetzt habe, ist der AppInvoker Simulator. Der AppInvoker ist eine Komponente, welche definierte Befehle, die vom Kunden im Client konfiguriert werden können, an die firmenexternen Systeme Bloomberg oder Reuters Eikon schicken kann. Der Kunde kann jeden konfigurierten Befehl in einem Menü auswählen. Entsprechend wird der Befehl über den AppInvoker entweder an Bloomberg oder an Eikon gesendet.

Wenn ein Testentwickler das Verhalten der Software nach Antwort von Bloomberg oder Eikon testen möchte, war es nötig eine Verbindung vom AppInvoker dorthin herzustellen und auf die Antwortnachricht zu warten. Mit dem Simulator soll den Testern hier Zeit und Arbeit erspart werden.

Dafür verbindet sich der AppInvoker Simulator über eine separate Netzwerkverbindung zum iQbonds-Server und meldet sich als AppInvoker an. Im GUI des Simulators kann ausgewählt werden, ob eine Verbindung zu Eikon oder zu Bloomberg simuliert werden soll. Der Server denkt nun es gäbe über den AppInvoker eine Verbindung zu einem der beiden Systeme und es kann ein Nachrichtenaustausch zwischen Simulator und iQbonds-Server stattfinden.

Alle Nachrichten, die an Eikon oder Bloomberg gesendet werden sollen, werden im GUI des AppInvoker Simulators in einer Liste angezeigt. Ein Tester kann nun jedes Element dieser einzeln auswählen und eine Antwort auswählen, die zurück an den Server geschickt werden soll. Außerdem werden alle beim Simulator ein- und ausgehenden Nachrichten geloggt und können in einem extra Fenster angezeigt werden.

So kann auf einfache Art und Weise das Verhalten getestet werden, ohne von firmenexternen Systemen abhängig zu sein.

## Arbeitsbedingungen

Bis auf die ersten paar Tage habe ich aufgrund der Corona-Pandemie im Home-Office gearbeitet. An meinem Arbeitsplatz stand ein Rechner im Büro und ich habe mich über ein Programm mit meinem privaten Gerät auf diesen Rechner eingewählt. Nachdem sich die Lage entspannt hatte, wurde mir freigestellt, weiterhin im Home-Office zu arbeiten oder an meinen Arbeitsplatz im Büro zu wechseln, und ich habe bis auf wenige Ausnahmen meinen Arbeitsplatz im Büro verwendet. Seit meinem Umzug im April arbeite ich fast ausschließlich im Home-Office.

# Fazit

Insgesamt schätze ich meine Werksstudentenstelle positiv ein. Der Anspruch und die Aufgaben sind abwechslungsreich und ich kann breitgefächert neue Erfahrungen bei der Arbeit als Softwareentwickler sammeln.

Neue Erkenntnisse konnte ich nicht nur über die Programmiersprache Java erhalten, sondern auch über Build-Umgebungen wie Ant oder Gradle.

Im Umgang mit Git habe ich gelernt, Pipelines in Gitlab aufzusetzen und mit komplexeren Projektstrukturen umzugehen. Dazu gehört zum Beispiel das Arbeiten an verschiedenen Versionssträngen und deren Synchronisation durch Cherry Picks.

Auf der anderen Seite muss ich viel mit Legacy code arbeiten, der an vielen Stellen undokumentiert ist. Das kommt gerade im Diagnosetool häufig vor, da dieses Projekt fast nur von Werksstudenten betreut und dadurch von vielen unterschiedlichen Entwicklern mit teilweise wenig Erfahrung implementiert wurde.

Abschließend denke ich, dass mir die Arbeit als Werksstudent bei valantic FSA ein gutes Bild davon gibt, wie ein Vollzeitjob als Softwareentwickler aussehen kann.

## Vergleich der Werksstudententätigkeit zum Berufspraktikum

In diesem Kapitel soll diskutiert werden, ob eine Werksstudententätigkeit vergleichbar ist zu einem Berufspraktikum. Dazu möchte ich zuerst auf die Praktikumsordnung eingehen.

Praktikumsordnung vom 07.02.2011 **§1 Zweck des Praktikums**:

„Das Praktikum hat das Ziel, die Studierenden […] bei der Bearbeitung einer komplexen Aufgabenstellung mit Arbeitsverfahren, -mitteln und -prozessen des jeweiligen Fachgebietes sowie mit organisatorischen und sozialen Verhältnissen der Praxis bekanntzumachen. Das Praktikum soll die Fähigkeit zur Teamarbeit fördern. Neben der fachspezifischen Tätigkeit soll die Praktikantin bzw. der Praktikant auch um den Erwerb von Kenntnissen über Betriebsorganisation, Sozialstrukturen, Sicherheits- und Wirtschaftlichkeitsaspekte bemüht sein.“ (8)

Die Ziele eines Berufspraktikums sind also die Folgenden.

### Bearbeitung einer komplexen Aufgabenstellung mit Arbeitsverfahren, -mitteln und -prozessen des jeweiligen Fachgebietes

Zur Implementierung des AppInvoker Simulators habe ich ein graphisches Userinterface und eine Netzwerkverbindung zum iQbonds Server implementiert. Dazu habe ich ein System zur Verarbeitung der Nachrichten und der entsprechenden Darstellung auf dem Userinterface geschrieben.

Zur Bewältigung dieser Aufgabe habe ich mich mit dem Nachrichtenprotokoll zwischen Server und AppInvoker und dem Bauen eines GUI mit Java-Swing auseinandergesetzt.

### Bekanntmachung mit organisatorischen und sozialen Verhältnisse der Praxis

Organisatorische Verhältnisse habe ich vor allem in Meetings kennengelernt. Zum Beispiel in der „Wochenplanung“. Dieses Meeting findet jeden Montag um 13 Uhr statt. Es wird zusammen mit dem für das Produkt zuständigen Projektmanager abgeklärt, wer in der anstehenden Woche welche Aufgaben zu bearbeiten hat und welche Priorität die Aufgaben bekommen.

Soziale Verhältnisse habe ich als Werksstudent wahrscheinlich deutlich besser kennengelernt. So habe ich Verhältnisse beim Mittagessen kennengelernt oder auch beim Tischtennis spielen im Büro, um den Kopf freizukriegen. Dazu kommen noch unregelmäßige und regelmäßige Treffen der Abteilung zum Beispiel bei den Magdeburger Developer Days oder auf dem Weihnachtsmarkt.

### Fördern der Fähigkeit zur Teamarbeit

Im Team habe ich nicht nur mit anderen Software Developern (zum Beispiel in Code Reviews), sondern zusätzlich mit Leuten aus der Testabteilung, dem Dokumentationsteam und verschiedenen Projektmanagern. Zum Beispiel habe ich wie in 2.5 beschrieben das Dokumentationsteam im Umgang mit Git und Gitlab gelehrt.

### Erwerb von Kenntnissen über Betriebsorganisation, Solzialstruktur, Sicherheits- und Wirtschaftlichkeitsaspekte

Die Betriebsorganisation und Sozialstruktur habe ich bei valantic FSA gleich mehrmals kennengelernt, denn seit meiner Einstellung im Januar 2022 erfahre ich gerade die zweite Umstrukturierung der Abteilungen. Dabei ist die zweite Umstrukturierung zu großen Teilen invers zu der ersten Umstrukturierung, das heißt es werden einige Änderungen der ersten Umstrukturierung rückgängig gemacht. Der größte Unterschied dieser Umstrukturierungen ist die Aufteilung der Abteilungen in Produkte und Tätigkeiten. Bis zum Ende dieses Jahres sind viele Leute in einer Abteilung, die zwar die gleiche Tätigkeit haben (zum Beispiel Software Developer), aber an verschiedenen Projekten arbeiten. Nach der Umstrukturierung werden stattdessen Leute mit Unterschiedlichen Tätigkeitsfeldern in einer Abteilung sein, die am gleichen Produkt arbeiten. Zum Beispiel alle Personen, die am Produkt iQbonds arbeiten.

Sicherheitsaspekte habe ich durch diverse Schulungen bezüglich IT-Sicherheit und Datenschutz sowie Arbeitsschutz kennengelernt.

Über Wirtschaftlichkeitsaspekte wurde im Unternehmen vor allem viel während und kurz nach der Covid-19 Infektion gesprochen. Da Die Arbeit während der Pandemie nur vom Homeoffice aus möglich war, wurde einige Zeit danach ein Desk-Sharing-System eingerichtet. Viele Mitarbeitende haben jetzt Schließfächer und Laptops, die sie an eine Docking-Station an einem Arbeitsplatz anschließen, der von verschiedenen Mitarbeitenden genutzt werden kann. Durch diese Maßnahmen konnten einige Arbeitsplätze entfernt und die Bürofläche deutlich verringert werden.

## Anrechnung als Praktikum

Da der Werksstudentenjob den in der Praktikumsordnung beschriebenen Zweck erfüllt und die vorgegebene Mindestdauer des Praktikums von 12 Wochen weit überschritten wird (Siehe Praktikumsordung §2 – Einteilung und Dauer des Praktikums (8)), kann die Werksstudentenstelle als Praktikum angerechnet werden.

# Literaturverzeichnis

1. **valantic. *valantic.* [Online] 29. 06 2024. https://www.valantic.com/en/.**

**2. Wikipedia valantic. *Wikipedia.* [Online] 29. 06 2024. https://de.wikipedia.org/wiki/Valantic.**

**3. LinkedIn valantic FSA. *LinkedIn.* [Online] 29. 06 2024. https://de.linkedin.com/company/valantic-fsa.**

**4. About Us: valantic FSA. *valantic FSA.* [Online] 01. 07 2024. https://www.valantic.com/fsa/about-us/.**

**5. Wikipedia Finanztechnologie. *Wikipedia.* [Online] 29. 06 2024. https://de.wikipedia.org/wiki/Finanztechnologie.**

**6. Oracle Docs - JFileChooser. *Oracle Java Platform.* [Online] 16. 11 2024. https://docs.oracle.com/javase/8/docs/api/javax/swing/JFileChooser.html.**

**7. Oracle Docs - JSlider. *Oracle Java Platform.* [Online] 16. 11 2024. https://docs.oracle.com/javase/8/docs/api/javax/swing/JSlider.html.**

**8. FIN, OVGU - Studiendokumente Praktikumsordnung. *FIN, OVGU.* [Online] 01. 09 2010. https://www.verwaltungshandbuch.ovgu.de/H%C3%B6B+Teil+I/1\_07+Praktikumsordnungen-media\_id-2010-p-52.html.**

Hiermit erkläre ich, dass ich diesen Praktikumsbericht selbstständig und ohne fremde Hilfe verfasst und keine anderen Hilfsmittel als angegeben verwendet habe.

Insbesondere versichere ich, dass ich alle wörtlichen und sinngemäßen Übernahmen aus anderen Werken als solche kenntlich gemacht habe.

Ort:

Datum: Unterschrift: