

Freie Universität Berlin, Fachbereich Mathematik und Informatik
Institut für Informatik, Studiengang Bachelor Informatik

Praktikumsbericht

von

Max Peter Wisniewski

4370074

max.wisniewski@fu-berlin.de

2. Juli 2012

bei scheideggerasen, Gründervilla Altensteinstraße 44b

von 01.01.2011 bis 31.3.2012

Betreuer: Janosch Asen, asen@scheideggerasen.com

Inhaltsverzeichnis

1	Praktikumsstelle	3
2	Aufgaben und Tätigkeiten	4
2.1	Tätigkeitsumfeld	4
2.2	Aufgabe und Ziele	5
2.3	Tätigkeiten und Arbeitsergebnisse	6
2.3.1	Einarbeitung	6
2.3.2	Refactoring und Strukturdesign	6
2.3.3	Umsetzung und Erweiterung	7
2.3.4	Nachwirkung	8
2.3.5	Ergebnis	8
3	Einsichten und Fazit	9
3.1	Technik	9
3.1.1	GWT	9
3.1.2	Git	9
3.1.3	Whiteboards	9
3.2	Methodik	9
3.3	Sonstiges	10
3.4	Fazit	11

1 Praktikumsstelle

Mein Berufspraktikum habe ich bei der Firma scheideggerasen absolviert. Scheideggerasen ist eine Agentur für ästhetische Kommunikation, die neben Werbung und Kommunikationsberatung auch an Projekten aus dem Kultur- und medienwissenschaftlichen Bereich arbeitet. Das momentane Projekt von scheideggerasen ist FarFromHomepage (FFHP), an dem auch ich gearbeitet habe, und das zur eigenen Gründung führen soll. Am Projekt FFHP arbeiteten als ich dort war, sechs fest mitarbeitende Personen, die ergänzt wurden durch freie Mitarbeiter, Praktikanten und Kooperationsvereinbarungen, die teilweise von der Gründerinitiative der FU - Berlin vermittelt wurden. Die Anzahl der Praktikanten schwankte über meine Anwesenheit, aber im Durchschnitt haben etwa 18 Personen an dem Projekt gearbeitet. Davon waren zu Beginn vier Programmierer. Die Branche des Unternehmens kann man der IT-, dem Kommunikationsmedien- oder auch dem Kulturbereich zuordnen.

Als ich dort arbeitete war das Unternehmen in der Vorgründungsphase und wurde durch verschiedene Förderungen von Seiten der FU - Berlin unterstützt. Ziel war es das Programm soweit zu entwickeln, dass man eine Beta-Phase zu testen erreichen konnte. Diese Phase wurde ein halbes Jahr nach Abschluss meines Praktikums mit freundlicher Hilfe aller ehemaligen Praktikanten erreicht. Bei dem von FFHP entwickelten Programm Guido handelt es sich um eine Art *Schnittprogramm* für das Internet. Man kann als *Autor* es allen Internetmedien eine *Tour* zusammenstellen und dann auf der Internetseite von FFHP präsentieren. Ein Betrachter dieser Tour kann sie sich einfach ansehen oder in den gezeigten Medien weiter surfen, da die ganze Tour interaktiv mit dem WorldWideWeb arbeitet.

Da es sich um ein kleines Unternehmen handelt, ist die Leitungsstruktur sehr flach. Gegründet wurde das Unternehmen vom Medien-, Theaterwissenschaftler und Philosophen Manuel Scheidegger und dem Medienwissenschaftler und Medienkünstler Janosch Asen, deren Nachnamen das Unternehmen auch seinen Namen verdankt. Mein Betreuer war der zweit genannte Gründer Janosch Asen.

Ich war als Praktikant bei der Entwicklung und der Programmierung des Frontends von Guido tätig. Da es zum damaligen Zeitpunkt noch recht wenig Programmierer und sehr vielfältige Aufgaben gab, kann man nicht direkt einen Aufgabenbereich herausstellen, für den ich Tätig war. Und bei einer Größe von vier Programmierern gibt es auch keine unterschiedlichen Abteilungen. Dadurch war das Arbeitsklima dennoch sehr angenehm. Sind Probleme an einer Stelle aufgetreten, konnte man die anderen leicht hinzuziehen, da zu diesem Zeitpunkt Kenntnisse über das gesamte System noch nicht schwer zu bekommen war und das Projekt auch noch nicht groß war. Da unser Chef und Betreuer immer bei uns war und obwohl er von Programmierung keine Ahnung hat, konnte er uns immer konkret neue und Sinnvolle Aufgaben geben. Neben der Koordination der Programmierer ist Janosch Asen für die Bürokratie, die Öffentlichkeitsarbeit und die Konzept- und Ideenentwicklung in der Firma zuständig.

Auf die Praktikumsstelle wurde ich erstmals im Sommer 2010 aufmerksam, als die Stellenausschreibung auf dem Mailverteiler von Spline herum geschickt wurde. Damals

wurde aber zunächst nur ein Programmierer gesucht und mein Freund Alexander Steen hat die Stelle bekommen. Ein halbes Jahr später wurden wieder neue Leute gesucht und Alex hat mich zum einen geworben und zum anderen Empfohlen. So habe ich vor Weihnachten 2010 ein Vorstellungsgespräch gehabt und konnte gegen Beginn des neuen Jahres (2011) dort anfangen zu arbeiten. Vor diesem Zeitpunkt habe ich weder dort noch irgendwo anders jemals gearbeitet.

Zu Beginn war mein Praktikum in der Studienzeit und ich hab 1-2 Tage die Woche für etwa 12h gearbeitet. Sobald das Semester vorüber war habe ich auf eine 40h Woche erhöht. Die Arbeitszeiten waren frei wählbar ich bin dennoch meist morgens gegen 10 Uhr ins Büro gekommen. Da ich keinen Schlüssel zum Gebäude hatte, war ich immer darauf angewiesen, dass schon eine Person vor mir im Büro war.

Das Praktikum an sich war nicht vergütet, ich wurde allerdings für eine Teilzeitstelle über das darauf folgende Semester bezahlt.

Das mehrwöchige Berufspraktikum war für mich sehr interessant, da ich zu diesem Zeitpunkt noch nie in einem größeren Projekt gearbeitet habe und es mir auch nicht recht vorstellen konnte. Da es sich um ein recht junges Unternehmen handelte, hat man einen tiefen Einblick in die komplette Struktur und Bürokratie eines solchen Unternehmens erhalten. Vom Programmiertechnischen Standpunkt auch, habe ich auch viel gelernt, da noch kurzer Zeit, als ich angefangen hatte zu arbeiten, der komplette Code refactored wurde.

So konnte ich daran teilhaben, die komplette Struktur eines Projektes mit zu diskutieren, zu entwerfen und mit zu bestimmen. Dabei konnte man vieles vom Programm-entwurf was man in Softwaretechnik gelernt hatte umsetzen. Der Entwurf an sich, war von außen betrachtet, ein bisschen zu viel für so ein Projekt. Die Modularisierung und Trennung der einzelnen Teile war recht lose gehalten. Nach einem halben Jahr allerdings konnte man beobachten, dass, obwohl das große Konzept nicht durchbrochen wurde, in Einzelfällen ein solcher Zusammenhalt bestand, dass es nicht zu trennen war und vor allem auch nicht mehr Wartbar. Die Geschwindigkeit, mit der wir den Code an eine nicht mehr wartbare Stelle gebracht haben, hat uns alle (zumindest hoffe ich das) ein bisschen Disziplin gelehrt.

2 Aufgaben und Tätigkeiten

2.1 Tätigkeitsumfeld

Wie schon vorher erwähnt in *Tätigkeiten und Arbeitsergebnisse* noch einmal genauer erläutert wird. Wurde kurz nach dem Beginn meines Praktikums der komplette Code weggeschmissen und das System als ganzen neu Entworfen.

An diesem Teil der Arbeit hat sich das komplette Programmiererteam inklusive meines Betreuers und Programmierkoordinators Janosch Asen beteiligt. Hier fand ein Großteil der Arbeit an Whiteboard und Diskussionsrunden statt. Alles in allem eine gute Zusammenarbeit, die den Zusammenhalt innerhalb des Teams auch gefördert hat. Dieses refactoring war nötig, da bis zu diesem Zeitpunkt eher alles als Prototyp entworfen

wurde. Vieles konnte schon getestet werden, aber alles zusammengekommen wurde die Struktur grausam und nicht skalierbar.

Nachdem der Großteil des Planes stand, musste zunächst die Struktur umgesetzt werden. Dies fand größtenteils im Side-by-Side programming statt, da auf diese Weise ein umfassendes Wissen über das Gesamtsystem geschaffen wurde und wir bei einer schnell wechselnden Belegschaft (es waren fast alle Programmierer nur Praktikanten) kein Expertenwissen schaffen wollten. Zum anderen konnten so schnell Probleme gelöst werden, wenn schnell viele Ideen zu diesem Problem kommen, ohne dass jeder neu in die Problematik eingearbeitet werden musste.

Als auch dieser Schritt getan war, konnte überwiegend an einzelnen kleinen Projekten gearbeitet werden. Diese waren größtenteils unabhängig von einander und in einzelnen Tickets und Milestones angeordnet. Trotzdem war es immer noch möglich mit Problemen umzugehen und sich helfen zu lassen oder anderen zu helfen, aber es wurde nicht mehr so eng zusammen gearbeitet, wie in der groben Programmstruktur. Der einzige, mit dem man immer eng zusammen gearbeitet hat, war Herr Asen, da dieser bis zum jetzigen Zeitpunkt unser Kunde war und uns immer schnell Rückmeldung geben konnte, ob das Getaute so gemeint war oder schon richtig umgesetzt war.

2.2 Aufgabe und Ziele

Im Projekt sollte ich das Programmiererteam unterstützen und die Entwicklung des selbigen voran bringen. Da das Unternehmen wenig Geld zur Verfügung hatte, gerade um Vollzeit Programmierer einzustellen, sollte ich in der Zeit meines Praktikums einen regulären Programmierer darstellen. Da keiner der Anwesenden Erfahrung mit großen Projekten hatte, war die Devise Learning-By-Doing. Im Grunde war es meine Aufgabe, die von Herrn Asen gestellten Aufgaben in Zusammenarbeit mit den anderen Programmieren zu erledigen. Sobald ich einige Erfahrung gesammelt hatte. Sollte ich auch zunehmend selbstständig an meinen Aufgaben arbeiten.

Da zu Beginn noch keine Aufgaben gesteckt waren, wurde an keiner Stelle festgesteckt, was ich im Verlauf erreichen sollte. Das vorrangige Ziel, war es nach dem Refactoring wieder eine Lauffähige Version zu erhalten. Dies hat auch die meiste Zeit des Praktikums in Anspruch genommen.

Meine eigenen Ziele sahen nicht ganz so Ziel gerichtet aus. Da ich bis zum damaligen Zeitpunkt noch nie ein Arbeitsverhältnis eingenommen habe, hatte ich keine Ahnung, wie ich auf die Forderungen, die auf mich zukommen würden, reagieren würde. Ich wollte also erst einmal sehen, ob ich unter Arbeitsanforderungen produktiv arbeiten könne.

Als nächstes musste ich Verifizieren, ob meine übliche Herangehensweise in einem Unternehmen angenommen wird. Ich habe die Eigenschaft immer lange über ein Problem nachzudenken, bevor ich auch nur anfangs etwas zu tun. Dies hätte sich als negativ auswirken können, wenn schnelle Ergebnisse verlangt sind. Was mich auf jedenfall noch interessiert hatte, war es das Gelernte aus Softwaretechnik in der Praxis zu erle-

ben und umzusetzen. Die Themen der Vorlesung sind sehr praxisorientiert und da ich zu diesem Zeitpunkt noch nie gearbeitet hatte, konnte ich viele Themen der Vorlesung nicht nachvollziehen. Wenn man aber alles einmal erlebt hat, macht dies die Vorlesung viel leichter verständlich. Es bietet sich unter Umständen ein, erst sein Praktikum zu machen und danach Softwaretechnik zu hören. Zu guter Letzt wollte ich erfahren, wie ich in größeren Gruppen arbeite, da ich von Natur aus eher eine stille Person bin.

2.3 Tätigkeiten und Arbeitsergebnisse

Zu Beginn des Praktikums habe ich eine Geheimhaltungserklärung unterschrieben, dass ich nichts genaues über das Projekt erklären darf. Da zum Zeitpunkt zu dem ich den Bericht nun schreibe, die Beta-Version schon öffentlich ist, kann ich konkreter werden, als ich es zum Ende meines Praktikums schon gekonnte hätte, aber ins Detail kann ich immer noch nicht gehen.

2.3.1 Einarbeitung

In den ersten zwei Wochen, in denen ich noch neben der Uni gearbeitet habe, habe ich mich zunächst in das schon vorhandene Programm eingelesen. Bei Guido handelt es sich im groben um eine Webseite und ich habe bis zum damaligen Zeitpunkt nur selten mit HTML und PHP und eigentlich noch nie mit Javascript gearbeitet. Zum Programmieren haben wir jedoch GWT¹ verwendet. GWT ist eine Sammlung von Bibliotheken und eine Sammlung von Precompilern und Compilern, die aus Java Code Javascript generieren können. Da ich durch die Uni Java Programmieren konnte, musste ich mich zunächst nur in den ungefähren Aufbau eines HTML Dokuments einlesen und wie man in Javascript an Probleme heran geht. Später habe ich mich genauer damit beschäftigt um einige Methode, zu denen es von GWT keine Interfaces gab in nativem Javascriptcode zu schreiben und in Java einzubinden.

Bei der Einarbeitung gab es keine Probleme, aber meinen Zielen hat es nicht unbedingt geholfen. Dass ich mich in neue Situationen einlesen kann wusste ich vorher schon.

2.3.2 Refactoring und Strukturdesign

Wie schon mehrfach erwähnt wurde nach wenigen Wochen der bisherige Prototyp des Programms weggeworfen und wir haben uns daran gemacht das komplette Programm neu zu entwerfen. Da wir an viele Stellen gesehen hatte, wo es Probleme geben kann und die Anforderungen langsam eine klarere Struktur bekamen, konnten wir in etwa abschätzen, was wir brauchen würden.

Der Plan war, ein reaktives System umzusetzen, dass in großen Teilen indirekt über einen Eventbus kommuniziert. So kann ein Objekt ganz allgemein sagen, dass gerade

¹GWT (Google Web Toolkit) ist ein Crosscompiler, der Javacode mit speziellen Bibliotheken nach Javascript compilieren kann.

ein Ereignis eingetroffen ist (z.B. ein neues Item soll zum aktuellen Zeitpunkt angezeigt werden) und alle Teile des Systems, die es interessiert, können diese Nachricht erhalten. Der Teil, der durch Mausklick oder anderen Trigger, was passieren muss, braucht so keine Ahnung zu haben, wen er alles ansprechen muss. Bei etwaigen Erweiterungen, muss dieser Teil des Codes nicht mehr verändert werden, sondern nur noch die neue Komponenten darauf eingestellt werden, die Nachricht zu empfangen. Der andere wichtige Teil, die wir im System umgesetzt haben, ist MVP (Model View Presenter), der analog zum in Softwaretechnik gezeigten MVC (Model View Controller) funktioniert. Später haben wir aufgrund der Komplexität den Presenter doch nochmal aufgeteilt, sodass wir sowohl Presenter als auch Controller hatten.

Was meine Ziele betrifft war dieser Teil des Praktikums ein großer Erfolg. Meine Natur eher länger über ein Problem nachzudenken, war an dieser Stelle von Vorteil, da im Entwurf an vieles Gedacht wurde, was erst ein halbes Jahr später erst in Angriff genommen wurde. Ein kleiner Nachteil war, dass wir im Team alle dazu geneigt waren zu viel nachzudenken und das System über zu designen. Es täte dem Team gut, wenn eine durchsetzungsfähige Person da wäre, die einspringt, wenn genug entworfen wurde. Sonst ist man mit dieser Phase nie fertig. Meine Schüchternheit war auch kein Problem. Die ganze Atmosphäre war sehr familiär und man konnte gut mit den anderen diskutieren.

Ein kleiner Misserfolg ist allerdings, dass ich im Nachhinein, nicht jeder ganz an den Entwurf gehalten hat. Das ist, solange es Verbesserungen sind, kein Problem, allerdings wurde der Entwurf zu Gunsten von Entscheidungen abgeändert, die wir eigentlich verhindern wollten und die das System an manchen Stellen schon nach ein paar Monaten schlecht Wartbar gemacht haben.

2.3.3 Umsetzung und Erweiterung

Nachdem der Entwurf soweit fertig war, haben wir das Grundgerüst fertig gestellt. Wie zuvor beschrieben, haben wir dabei noch eng zusammen gearbeitet. Gerade weil an vielen Stellen noch nicht fertig geklärt war, wie wir es Umsetzen wollen. Wir haben im Programmentwurf noch keine große Erfahrung gehabt, so dass wir nicht an alles denken konnten.

Nachdem das Eventsystem stand und wir für MVP alle Interfaces und schon mehrere abstrakte Klassen fertig hatten, konnten wir Anfangen aus dem Prototypen vorhandene Funktionalität nach zubauen. An dieser Stelle habe ich erst einmal einfache Aufgaben übernommen, da ich mich mit den Probleme, die schon einmal aufgetreten sind, noch nicht aus kannte.

Als das System soweit stand, konnte dazu übergegangen werden neue und schon einmal vorhandene Features einzubauen. Mein Tätigkeitsfeld waren unter anderem das einbinden der Youtube-API. Dieses API wurde bis zum jetzigen Zeitpunkt bestimmt acht mal implementiert von unserer Seite und funktioniert immer noch nicht einwandfrei. Als nächstes musste zu jeder Tour eine zeitliche Anordnung realisiert werden, über die ich mir Gedanken machen durfte. Und neben vielen Kleinigkeiten durfte ich

zuletzt noch eine Verwaltung schreiben, wie z-Layer (in HTML) an unsere dynamisch angeordneten Items vergeben werden, ohne dass man das komplette Dokument neu aufbauen muss, da man auf einem Layer keinen Platz mehr hat.

An dieser Stelle habe ich gelernt, wie ich damit umgehen kann, wenn ich vorher lange über ein Problem nachdenke, ohne anfangs zu programmieren. Ich hab alles, über das ich nachgedacht habe gut dokumentiert und in textueller Form im Projekt hinterlegt. Das hat erstens den Fortschritt belegt und zweitens schon in mehreren Bereichen dem nachfolgenden Programmierer gezeigt, wo Probleme auftreten können. Im Falle der Timeline habe ich gar nicht angefangen etwas umzusetzen, aber habe alle Fehler und Probleme dokumentiert, die allgemein oder in der aktuellen Struktur auftreten können. Dies hat bei der Umsetzung von der nächsten Person geholfen, als diese die ersten fünf Fehler gemacht hatte, die ich aufgeschrieben habe.

2.3.4 Nachwirkung

Dies kann ich rückblickend auf die Hinarbeit der Beta-Phase ein halbes Jahr später sagen, da es sich im Praktikum bereits angebahnt hat. Da aufgrund der knappen Finanzlage ein Zeitdruck hinter allem stand und wir getrieben wurden eine präsentationsfähige Software aufzustellen, mussten wir an vielen Stellen Kompromisse eingehen. Gegen Ende hieß es immer, dass die Beta nächste Woche herauskommen sollte. Dies hat dazu geführt, dass an vielen Stellen nur Bugfixes eingefügt wurden und das Problem an sich nicht mehr gelöst werden konnte. Was nach dem Beta release dazu führte, dass wir nicht mehr wissen, wo nun gerade Fehler im Code sind, die wir noch korrigieren müssen und das ganze System äußerst labil läuft.

Dies ist ein großer Fehler von unserer Programmiererseite gewesen. Erstens hätte man die Bugfixes besser dokumentieren sollen und zweitens die Deadline einmal weit hinausschieben sollen und nicht viele Male ein bisschen.

2.3.5 Ergebnis

Insgesamt war das Praktikum erfolgreich. Wir haben für die Beta-Phase einen Grundstein gelegt, der innerhalb des Praktikums schon funktionierte. Ich habe vieles gelernt, wie man in größeren Gruppen arbeitet und wie man die Anforderungen von einer Person bestimmen kann, die keine Ahnung von Informatik hat. Ich habe viele Fehler gemacht und hoffentlich aus allen etwas gelernt. Darüber hinaus hat das Praktikum mir Spaß gemacht und ich helfe dem Unternehmen immer mal wieder, wenn etwas getan werden muss.

3 Einsichten und Fazit

Dieser Abschnitt beschreibt, welche wichtigsten Einsichten (Aha-Erlebnisse) ich aus dem Praktikum mitgenommen habe.

3.1 Technik

3.1.1 GWT

Das Google Web Toolkit war für mich in jedem Fall wunderbar zu benutzen, da ich von Javascript keine Ahnung hatte und auf diese Weise erst einmal langsam heran geführt werden und dabei schon trotzdem effizient arbeiten konnte. Eine Falle in die man am Anfang aber schnell läuft ist, dass durch die Javascript Umsetzung im Hintergrund manche Operationen und Datentypen nicht so effizient sind, wie man glaubt. Hash-Maps sind in Javascript nach versuchen auf jeden Fall nicht so schnell gewesen, wie wir erfahren mussten. Auf anderen Datentypen sind manche Operationen nicht portiert worden. Zum Beispiel haben wir auf TreeMaps kein floorKey und ceilKey gehabt, die aber dringend gebraucht wurden. Nach dem rein lesen und den ersten Wochen Verzweiflung konnte man aber recht gut damit umgehen.

3.1.2 Git

Wir haben mit dem Versionsverwaltungssystem Git gearbeitet. Git an sich ist sehr mächtig, da wir aber keine Erfahrung damit hatten, ist es zu Beginn öfters passiert, dass man seine kompletten Fortschritte vernichtet hat.

Ein kleiner Crashkurs sollte dem Abhilfe schaffen, wir hatten aber leider keinen.

3.1.3 Whiteboards

Whiteboards sind wunderbar, wenn man über einen Entwurf diskutieren will. Das einzige Problem ist es ohne einen Schriftführer alles zu dokumentieren. Und den hatten wir aufgrund der knappen Belegschaft nicht, da alle mit diskutiert haben. Aber ein Whiteboard, Flipchart oder eine Tafel sollte in keiner Besprechung fehlen. Software-gestützte Präsentationen wirkten dagegen meist fehl am Platz bei den Besprechungen.

3.2 Methodik

Wenn man einen Entwurf gemacht hat, dann muss man sich nicht sklavisch daran halten. Aber wenn man entdeckt, dass es in einer speziellen Situation keinen Sinn macht nach dem Entwurf vorzugehen, dann sollte man nicht eigenmächtig beschließen ihn abzuändern und schon gar nicht das ganze undokumentiert zu tun. Wenn man an einen solchen Punkt kommt, sollte man mit den anderen darüber reden und zusammen entscheiden, ob eine Änderung nötig ist oder wie diese auszusehen hat. Bei uns ist es an einer Stelle vorgekommen, dass eine Person eigenmächtig Direktzugriff durch 3 Abstraktionsebenen gewährt hat, weil es an der Stelle gut besser ging. Wir haben gesehen, dass die Abstraktion an dieser Stelle zu schwer fällig war, aber wir haben den

Eingriff erst bemerkt, als wir durch Zufall über einen Fehler an dieser Stelle gestolpert sind. Durch die Veränderung haben wir diesen Fehler auch sehr schlecht gefunden.

Wenn man eine Beschreibung bekommen hat, die nicht vollständig spezifiziert ist, sollte man auf jeden Fall nochmal nachfragen. Dies tritt gerade auf, wenn der andere kein Informatiker ist. Man sollte sich an dieser Stelle die Lücken nicht selber zu denken, da am Ende etwas heraus kommen kann, was überhaupt nicht gewünscht ist. Da wir einen engen Kontakt mit unserem Kunden, der immer im selben Raum saß, konnten wir als Lösung immer einen schnellen Prototypen bauen und weiter entwickeln, in die Richtung, die er meinte. Sollte er aber einmal nicht so perfekt zur Verfügung stehen, sollte man sich alle Fragen vorher überlegen oder man läuft Gefahr alles mehrfach neu schreiben zu müssen.

Ruhe bewahren ist das oberste Motto einer Besprechung oder in einer Diskussion. Ich selber neige dazu, wenn ich nicht verstanden werden, etwas aggressiv in meiner Sprechweise zu werden. Dies hat aber noch nicht geholfen, dass es der andere besser versteht. Dies viel aber nur auf, da unsere Besprechungen immer allgemein sehr ruhig waren. Als Training war es sehr gut Tutor zu sein. Man darf die selben Fragen ziemlich oft erklären und bleibt darüber recht ruhig, auch wenn man es zehn mal erklären darf.

3.3 Sonstiges

Wir waren zu Beginn eine recht kleine Gruppe, in der sich die meisten Probleme von selbst gelöst haben. Sobald wir aber etwas mehr wurden, stellte sich raus, dass wir zu wenig Disziplin und Dokumentation aller unserer Arbeitsergebnisse, Aufgabenverteilung und Probleme hatten. In größere Unternehmen hat sich an dieser Stelle schon ein Workflow eingestellt, den wir noch finden mussten. Gerade regelmäßige Besprechungen sind bei einer Gruppengröße von mehr als drei Mitgliedern unabdingbar, da sonst niemand einen Überblick hat.

Es gibt manche Personen, die reagieren zwar, darauf, wenn man ihnen sagt, dass sie etwas falsch tun oder es anders besser geht, aber trotzdem weiter versuchen ihre Idee durchzusetzen. Auf diese Leute sollte man ein Auge haben. Es ist nicht schlecht zum entwerfen eines Prototyps, wie wir feststellen konnten, aber im Endergebnis ist ein bisschen mehr Disziplin nötig. Also sollte man darauf achten, dass diese Personen sich an die Richtlinien halten.

Ich selbst habe mich unterschätzt, wenn es darum geht, große Aufgaben zu bewältigen. Ich bin, wie nicht nur bei der Umsetzung von großer Software, von einer großen Aufgabe überwältigt, so dass ich nicht weiß, wo ich beginnen soll, oder wie ich das ganze überhaupt schaffen kann. Doch wenn man sich immer nur auf kleine Teilstücke konzentriert, kann man es doch auch große Aufgaben zügig und gut schaffen.

3.4 Fazit

Das Praktikum hat mir an einigen Stellen gezeigt, wie ich mein Wissen in die Praxis umsetzen kann. Das war schön einmal zu sehen, aber ich kann es mir nicht vorstellen später einmal als Programmierer zu arbeiten. Diese Aufgabe ist mir doch zu eintönig und Öde. Im Praktikum habe ich allerdings gelernt, mehr Lösungen von anderen zu lesen, die das selbe Problem, wie ich hatten. Im Informatik Studium wird man sollten dazu geführt, sich Lösungen von anderen durchzulesen um sein Problem zu lösen. Dort muss man oft seine Probleme von Grund auf selber lösen und darf jeden Fehler einmal selber begehen. Im Grunde genommen gut, da man aus seinen Fehlern lernen kann, aber es ist unvernünftig schon vorhandenes Wissen zu verwerfen um alles noch einmal neu zu machen.

Das Studium bringt mir an sich aber mehr Wissen. Im Praktikum war es oft immer wieder das selbe das man tun musste. Im Studium gibt es mehr Abwechslung und vielfältigere Probleme. Es macht mir insgesamt mehr Spaß.

Mein Studienverhalten werde ich nach meinen Erfahrungen im Praktikum eher nicht ändern. Vieles von dem was ich erfahren habe war ein Erfolg oder eine Bestätigung. Die Richtung meines Praktikums, das heißt insbesondere Webprogrammierung, ist nicht mein Interessengebiet, es hat mich eher bestätigt, nicht weiter in diese Richtung zu gehen.

Allen anderen, die ein Praktikum suchen, kann ich es auf jeden Fall empfehlen, in ein kleines Unternehmen zu gehen. In großen Unternehmen kann man unter Umständen ein feste Struktur lernen, wie man an große Projekte heran geht. Man wird aber nicht lernen, warum dies so gemacht wird, sondern wird nur dazu gedrängt es so zu machen. In einem kleinen Unternehmen ist man sehr nah an der Entscheidungsfindung und man wird vielleicht sogar daran beteiligt. Man darf einmal jeden Fehler begehen, den man begehen kann. Gerade zum lernen ist dies ein Vorteil. In kleinen Unternehmen hat man auch eine größere Auswahl woran man arbeiten möchte. Bei großen Unternehmen wird man meist ohne Wahl vor eine Aufgabe gesetzt.

Deshalb kann ich jedem nur ein Praktikum bei einem Startup empfehlen. Wir haben unter anderem auch schon viele Mitstudenten an FFHP vermittelt und den meisten hat das Praktikum Spaß gemacht.