

# TELEGRAM CHAT-BOT

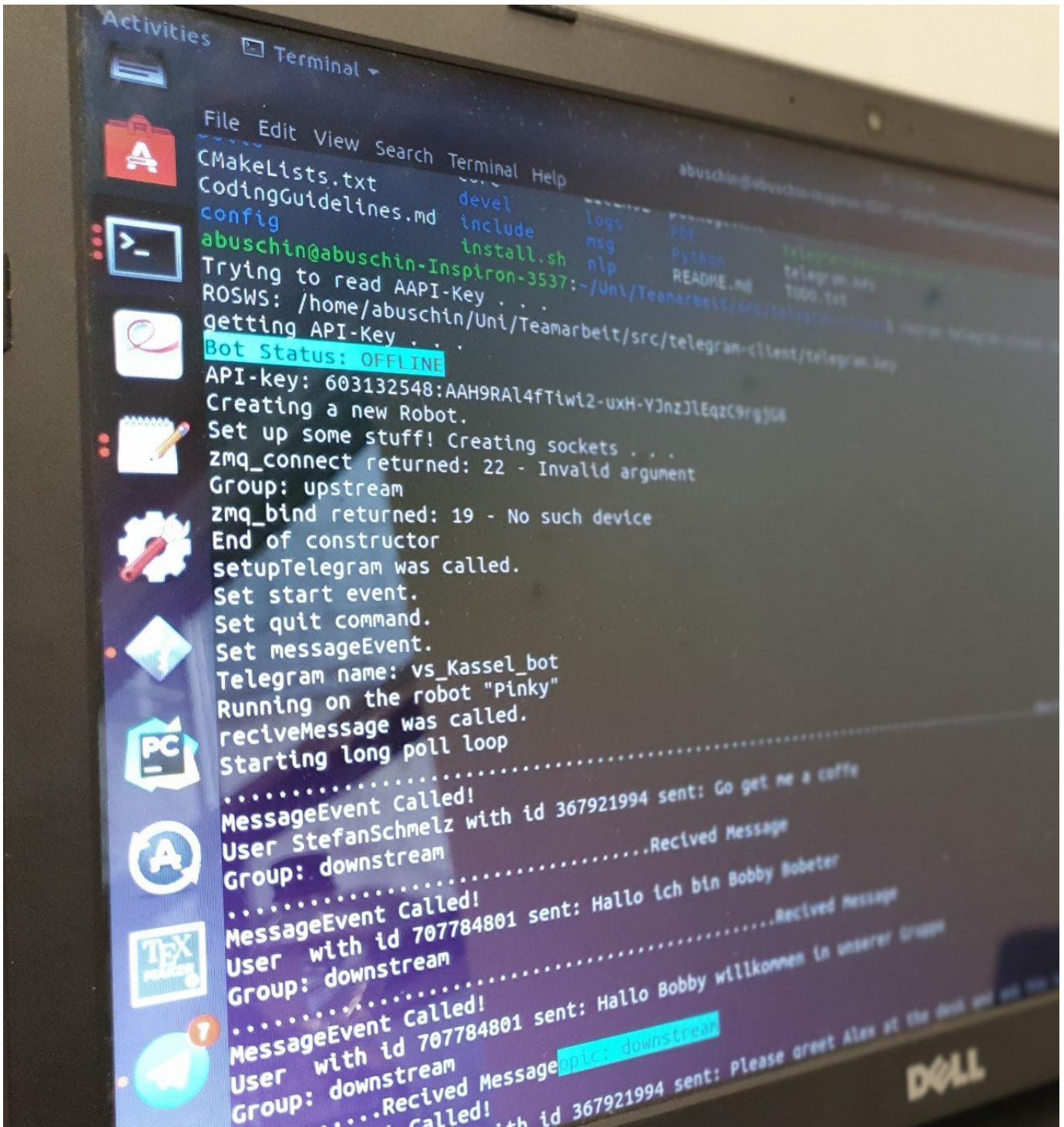
Eine Zusammenarbeit von:

Stefan Schmelz

Artjom Buschin

Rohhullah Walizada

Amir Hossein Ziaian Galehdar



```
Activities Terminal
File Edit View Search Terminal Help
CMakeLists.txt
CodingGuidelines.md
config
abuschin@abuschin-Inspiron-3537:~/Uni/Teamarbeit/src/telegram-client/telegram-key
Trying to read AAPI-Key
ROSWS: /home/abuschin/Uni/Teamarbeit/src/telegram-client/telegram-key
getting API-Key
Bot Status: OFFLINE
API-key: 603132548:AAH9RA14ftiwl2-uxH-YJnzJLEqzC9rgjG
Creating a new Robot.
Set up some stuff! Creating sockets ...
zmq_connect returned: 22 - Invalid argument
Group: upstream
zmq_bind returned: 19 - No such device
End of constructor
setupTelegram was called.
Set start event.
Set quit command.
Set messageEvent.
Telegram name: vs_Kassel_bot
Running on the robot "Pinky"
recvMessage was called.
Starting long poll loop
.....
MessageEvent Called!
User StefansSchmelz with id 367921994 sent: Go get me a coffee
Group: downstream
.....
MessageEvent Called!
User with id 707784801 sent: Hallo ich bin Bobby Bobeter
Group: downstream
.....
MessageEvent Called!
User with id 707784801 sent: Hallo Bobby willkommen in unserer Gruppe
Group: downstream
.....
MessageEvent Called!
User with id 367921994 sent: Please greet Alex at the desk and tell him...
```

Modul: Teamarbeit



# **I. Inhaltsverzeichnis**

## **II. Technische Überlegungen:4**

## **III. Komponenten des Projekts:5**

## **IV. Definition Teamarbeit:7**

- . *Warum ist Teamarbeit wichtig für Informatiker?8*

## **V. Persönliche Rolleneinschätzungen:9**

- . *Stefan Schmelz:9*
- . *Artjom Buschin:10*
- . *Rohhullah Walizada:11*
- . *Amir Hossein Ziaiyan Galehdar:12*

## **VI. Konflikte und Bewältigungsstrategien13**

- . *Wie entstehen Konflikte in einem Team?13*
- . *Bewältigungsstrategien13*

## **VII. Ablauf unserer Teamarbeit:14**

## **VIII. Fazit:15**

## II. Technische Überlegungen:

Unsere Aufgabe war es, unter Verwendung von Capnzero (Einem Projekt von Stephan Opfer um ROS als Middleware durch Capnproto und Zeromq FG-intern zu ersetzen) einen Chat-Bot zu entwickeln, welcher auf natürliche Sprache reagieren kann. Dies sollte dazu benutzt werden um dem Roboter mehr oder weniger einfache Befehle wie „Hol mir einen Kaffee“, oder „Finde meine Tasse“ zu geben.

Die erste Frage, die sich uns stellte war, ob wir eine eigene Chat-Anwendung entwickeln sollten, oder einen bestehenden Messenger zu benutzen. Anschließend haben wir festgestellt, dass wir am besten einen bestehenden Messenger benutzen, da wir uns dann mehr auf die Textverarbeitung konzentrieren können.

Die Entscheidung welchen Messenger wir benutzen möchten fiel auf Telegram, denn die API, die Telegram zur Verfügung stellt ist kostenlos, sehr umfangreich und gut dokumentiert.

Wir haben als erstes angefangen den Teil des Projekts zu implementieren, der mit Telegram kommuniziert. Einer unserer Gedanken war, unser Projekt möglichst modular zu halten, damit es uns leichter fällt, in der Zukunft einzelne Teile des Projekts umzubauen oder von außen einen bestimmten Teil des Projekts anzusprechen.

Ein Telegram-Bot benötigt eine API-Key, welchen man über einen Bot in einem Automatisieren Verfahren beantragen kann. Der Bot, bei dem man einen API-Key beantragen kann nennt sich „Bot Father“ und das beantragen läuft folgendermaßen ab:

- 1.) Man schreibt den Bot an, in dem man einen Link auf einem Gerät mit Telegram öffnet.
- 2.) Damit der Bot weiß, dass du mit ihm kommunizieren möchtest, muss man den Befehl: „/start“ senden.
- 3.) Der Bot schickt einem eine Nachricht, in der er erklärt was er kann und wie er benutzt werden möchte. Man kann die Befehle in der Nachricht anklicken, in das Textfeld tippen oder aus einer Liste auswählen. Wie genau das funktioniert hängt vom verwendeten Gerät ab. Um einen neuen Bot zu erstellen, muss man den Befehl „/newbot“ senden.
- 4.) Der Bot fragt nach dem Namen, der im Client angezeigt werden soll. Hier antwortet man mit dem Namen, den der eigene Bot tragen soll.
- 5.) Der Bot fragt nach einem Benutzernamen für den eigenen Bot. Dieser Benutzername muss eindeutig sein, wobei der Bot-Father einem mitteilt, wenn ein Name bereits vergeben ist. Der Benutzername für einen Bot muss immer auf „bot“ oder „Bot“ enden. Auch hier soll man nur den Namen schicken.
- 6.) Die Erstellung ist abgeschlossen und der Bot-Father teilt den API-Key und die URL, unter welcher man den Bot kontaktieren kann mit. Die URL folgt in den meisten Fällen dem Muster: „t.me/<Benutzername>“

### III. Komponenten des Projekts:

Wir haben temporäre Namen für die einzelnen Programme in unserem Projekt vergeben, damit es einfacher ist darüber zu reden und man damit einen übersichtlicheren Überblick über das Programm bekommt.

Nach der Einrichtung kann man, wenn gewünscht, noch ein paar Einstellungen vornehmen, die vom Anwendungsfall abhängen und in der Dokumentation der API ausführlich erklärt und dargestellt sind.

Mit Hilfe einer inoffiziellen Bibliothek für die Programmiersprache C++ haben wir die TgBridge geschrieben. Diese soll Nachrichten von dem Messenger Telegram entgegennehmen und Informationen aus der Telegram Nachricht in eine Datenstruktur verpacken, die wir zum internen Weitertransport und Verarbeitung benutzen wollten. Nach dem das funktionierte, haben wir die TgBridge erweitert, so dass man auch eine ausgefüllte Datenstruktur zur TgBridge schicken kann, die dann über Telegram verschickt wird. Ursprünglich wollten wir eigentlich nur senden und empfangen, haben aber festgestellt, dass man über einen API-Key nur eine Instanz gleichzeitig verwenden kann. Startet man eine zweite Instanz der TgBridge über denselben Key, bekommt man zuerst einfach keine Nachrichten mehr und kann auch keine mehr senden. (Dies könnte sich in der Zwischenzeit geändert haben, denn wir haben es nicht weiter getestet.)

Die Nachrichtenobjekte sollen von dem NLP-Modul namens „msgProc“ verarbeitet und in einen Parse-Tree zerlegt werden. Wir haben keine C++ Bibliothek zur Verarbeitung von Sprachen gefunden, die uns gefallen hat, also haben wir uns bei den vielen Sprachen umgesehen, die von Capnproto und Zeromq unterstützt werden und sind durch unsere Suche bei einem Python-Modul namens „spacy“ fündig geworden. Unser durchlaufene Suchprozess zur Findung des NLP-Moduls kann den folgenden Abschnitten entnommen werden.

Um die natürlich sprachlichen Anfragen und Texte der Messenger User zu verarbeiten, benötigten wir eine „Natural Language Processing“-Bibliothek, mit der wir die benötigten Informationen extrahieren und verarbeiten könnten. Eine sehr große Auswahl an Bibliotheken findet sich besonders in der Programmiersprache Python. Der erste vielversprechende Fund war die „Flair“-Bibliothek. Diese war mit einer großen Sprachenvielfalt, in der unter anderem auch Deutsch wiederzufinden war und mit einer guten „Named Entity Recognition“ ausgestattet.

Das Themengebiet „Named Entity Recognition“ beschäftigt sich damit wichtige, in Namen enthaltende Wörter aus einem Text zu extrahieren. Nach der Implementierung eines Testcodes ließen wir einige Anfragen von der Bibliothek verarbeiten. Unter anderem erhielten wir bei den Anfragen „Holst du mir bitte einen Kaffee.“ und „Wie viel Kaffee hast du heute schon geholt?“ dieselbe Ausgabe „Kaffee“. Da merkten wir schnell, dass wir allein mit „Named Entity Recognition“ nicht weit kommen würden. Wir konnten damit nicht das Verb beziehungsweise das „Ziel“ der Anfrage extrahieren und würden somit nicht wissen was der Roboter genau machen sollte. Da in der „Flair“-Bibliothek solche Funktionen nicht implementiert waren, mussten wir auf diese verzichten und eine neue und passendere Bibliothek suchen.

Unter den anderen bekannten NLP-Bibliotheken wie „Natural Language Toolkit“ (NLTK) fanden wir auch „spaCy“. „spaCy“ bietet die Möglichkeit der Verwendung eines schnellen und guten „Dependency Parser“. So konnte man mit Hilfe dieser Bibliothek sich zu jeder deutschen Anfrage

einen „Parse Tree“ erstellen lassen. Mit kleinen Navigationsschritten durch den Tree erhielten wir bereits das wichtigste Verb mit allen dazugehörigen Abhängigkeiten.

Das nächste bevorstehende Problem war es eine Kommunikation zwischen dem NLP-Modul und der TgBridge, die in C++ umgesetzt wurde, und der in Python umgesetzten Sprachverarbeitung herzustellen. Dazu verwendeten wir „Cap'n Proto“ und „ZeroMQ“. So wird die vom Telegramclient empfangene Nachricht des Users zunächst mit unserer Nachrichtenklasse mit „Cap'n Proto“ in Bytes umgewandelt, die dann mithilfe von „ZeroMQ“ an unser Sprachverarbeitungsprogramm geschickt werden. Das Sprachverarbeitungsprogramm empfängt die versendeten Bytes mit „ZeroMQ“ und dekodiert diese dann wieder mithilfe von „Cap'n Proto“, sodass aus der dekodierten Nachrichtenklasse der Nachrichtentext und alle anderen wichtigen Informationen rausgelesen und verarbeitet werden können. Die Kommunikation in die entgegengesetzte Richtung erfolgt analog.

Es sollte auch eine Nutzerdatenbank geben, die aber bisher nicht implementiert wurde. Sie sollte zu Beginn auch auf Message-Objekte lauschen, die von der TgBridge empfangen werden und alle Metadaten automatisch aufnehmen, die sie aus den Metadaten der Nachricht lesen kann. Dieses Vorgehen wollten wir benutzen, damit man den Inhalt der Datenbank zurücksetzen kann, ohne dass der Nutzer etwas davon merkt.

Jetzt halten wir es für besser, dass die TgBridge eine spezielle Nachricht aussendet, wenn sie einen „/start“-Befehl empfängt, denn diesen muss jeder neue Benutzer auf jeden Fall senden. Allerdings ist es dann nicht mehr ohne weiteres möglich Veränderungen der Metadaten zu bemerken und die Datenbank automatisch zu aktualisieren. Wir sind noch nicht sicher, ob das überhaupt notwendig ist da sich zumindest laut der API-Dokumentation die User-ID nicht ändern kann und wir diese zur Identifizierung benutzen.

Wir waren am Überlegen, ob es sinnvoll sein könnte, ein weiteres Modul zu bauen, welches sich um die Verwaltung der Konfiguration kümmert und das andere Modul benachrichtigt, wenn sich die Konfiguration und die Einstellung ändert. Allerdings haben wir dazu noch nichts implementiert und wissen noch nicht, wie viele Werte der Konfiguration und der Einstellung sich überschneiden. Wahrscheinlich ist ein solches Modul erst sinnvoll, wenn nur größere Teile der Roboter-Software betrachtet werden, da sich diese erst bei größeren Projekten die Einstellungen signifikant doppeln. In dem Fall kann ein Manager aber Probleme mit der Konfiguration deutlich reduzieren.

#### **IV. Definition Teamarbeit:**

Als Team wird ein Zusammenschluss von mehreren Personen bezeichnet, die ein bestimmtes Ziel bzw. eine bestimmte Aufgabe im Blick haben und dieses gemeinsam durch ihre persönlichen Fähigkeiten und Fertigkeiten Lösen wollen.

Der Begriff „Team“ hat aber in der Praxis sehr viele verschiedene Bedeutungen und Assoziationen, mit denen dieser in Verbindung gebracht werden kann. Wegen diesen Assoziationen werden diesem Begriff sechs verschiedene Dimensionen zugeordnet. Es existieren diesbezüglich die Erlebnis-Dimension, die Aufgaben-Dimension, die Image-Dimension, die Krisen-Dimension, die Prozess-Dimension und die Ergebnis-Dimension. Diese werden nun hier nochmal verdeutlicht.

- Die Erlebnis-Dimension zeichnet sich dadurch aus, dass der Aspekt der gefühlsmäßigen Verbundenheit im Vordergrund steht. Darüber hinaus empfinden sich alle Teammitglieder als Gemeinschaft Gleichgesinnter.
- Die Krisen-Dimension wird durch sehr schnelle Teamfindung und sehr guter Funktionalität dieser Teams in einer schwierigen Zeit gekennzeichnet. Allerdings ist der Erfolg meistens nicht von langer Dauer und nur auf die Krisenzeit beschränkt.
- Die Ergebnis-Dimension zeichnet sich durch Begeisterung des ganzen Teams, aufgrund einer faszinierenden gemeinsamen Aufgabe bzw. Arbeit aus. Dies hat zur Folge, dass meistens der Erfolg im Vordergrund steht und Einzelne nicht mehr berücksichtigt werden.
- Die Prozess-Dimension wird dadurch gekennzeichnet, dass die bereichs- bzw. abteilungsübergreifende Teamarbeit im Mittelpunkt steht (die Sache steht im Vordergrund). Diese Dimension hat den Vorteil, dass eine dauerhafte gute Zusammenarbeit zwischen Abteilungen gewährleistet wird, aufgrund von Kommunikations- und Schnittstellenproblemen Bewältigung.
- Die Aufgaben-Dimension zeichnet sich durch Spezialwissen/Fachwissen der einzelnen Mitglieder aus, durch eine fachliche/sachliche Aufgabenstellung und gemeinsamen Herausforderungen wird eine Ebene von Zusammenhalt geschaffen.

## **.Warum ist Teamarbeit wichtig für Informatiker?**

In der Informatik ist Teamarbeit essenziell. Ohne Fähigkeiten und Kompetenzen, die für eine Arbeit zusammen mit anderen Personen im Team erforderlich sind, kann sogar ein fähiger Informatiker nicht lange in der Arbeitswelt konkurrenzfähig bleiben. Teamarbeit bzw. Fähigkeiten dazu werden in der heutigen Arbeitswelt als Grundvoraussetzungen angesehen. Die meisten größeren Projekte bzw. Aufgaben die ein Unternehmen seinen Angestellten, insbesondere den Informatiker stellt, können nicht von einer Person gelöst werden bzw. sind viel zu zeitaufwendig für eine einzelne Person und kostspielig für das Unternehmen. Darüber hinaus hat eine einzelne Person nicht das gleiche Wissen, wie eine andere Person. Durch eine Arbeit im Team ergänzen sich diese Personen mit ihrem Wissen gegenseitig und können dann die Aufgaben bzw. das Projekt gemeinsam lösen bzw. bewältigen. Aus diesen Gründen ist Teamarbeit für Informatiker von großer Bedeutung.



## **V. Persönliche Rolleneinschätzungen:**

### **.Stefan Schmelz:**

Zum Kickoff-Seminar habe ich mich mehr als Konzeptionierter oder Macher eingeschätzt, da ich bei der Schilderung eines Problems meistens nach wenigen Minuten der Überlegung schon eine Idee habe, wie es zu lösen sein könnte. Diese Ideen sind meistens mit mehr oder weniger großen Denkfehlern behaftet, aber meistens fallen sie von ganz alleine relativ schnell auf.

Ich arbeite normalerweise mehr oder weniger iterativ. Damit meine ich, dass die erste Version meines Codes meistens wenig aufgeräumt und eher kryptisch ist, doch meistens verbessert sich die Qualität innerhalb weniger Durchläufe drastisch und ich verbessere meinen Code nach und nach bis zur Perfektion.

Bisher habe ich einige kleinere Projekte alleine absolviert und bei Carpe Noctem einen Einblick in die Arbeit mit einer größeren Code-basis in einem Projektteam bekommen.

In der Praktischen Arbeit mit Rohhullah, Artjom und Amir hat sich aber relativ schnell gezeigt, dass die anderen Teammitglieder nicht wussten, wo sie anfangen sollten. Deshalb hätte ich eigentlich schon viel früher die Rolle als Koordinator einnehmen müssen, doch das habe ich in der Situation nicht erkannt, da mir die Organisation von mehr als mir selbst schwerfällt.

Ich habe meine Rolle als Conceptualizer und Umsetzer ernster genommen, da ich schon auf der Heimfahrt vom Kick-Off Seminar angefangen habe, mich in die vermeidlich notwendige Materie einzulesen und nach passenden und richtigen Bibliotheken zu suchen.

Für mich kristallisierte sich im Ablauf der Zeit mehr heraus, dass von meinen Teamkameraden nicht zu erwarten war, dass sie eine Position als Anführer beziehen würden, da ihnen an fachlichen Kenntnissen mangelte. Von Carpe Noctem war ich es gewohnt, dass man bei den Treffen nicht unbedingt ein festes Ende hatte, weswegen ich auch gegenüber weniger projektorientierten Gesprächen eher gelassen war. Trotzdem hat der leichte Antriebs von Artjom mir sehr geholfen produktiv zu sein.

Wenn ich über die ganze Laufzeit des Projektes zurückblicke, bin ich der Meinung, dass ich am besten schon früh die Rolle eines Teamleiters einnehmen konnte, aber meiner Ansicht nach habe ich über die Laufzeit verteilt, nicht nur die Rolle eines Generators, sondern auch des Umsetzers und eines Teamleiters versucht auszufüllen und auszuüben.

Das Projekt hat mir viele neue nützliche Erfahrungen beschert und hat mir auch gezeigt, dass man am besten gewisse Dinge bereits zu Beginn festlegt.

## **.Artjom Buschin:**

Zu Beginn des Kick-Off Meetings stand ich vor der interessanten Aufgabe, meine Rolle im Team einzuschätzen. Dies ist mir und der Reaktion meiner Kommilitonen zufolge einigen anderen auch nicht gerade leichtgefallen. Das Rollenmodell bietet vier markante Entscheidungsmöglichkeiten, die folgende Frage aufwarfen:

Bin ich ein motivierender Initiator, ein Konzeptionist, ein Optimierer oder bin ich ein Umsetzer?

Die Rolle des Optimierers konnte ich nach einigen Denkminuten schon ausschließen. Ein funktionierender und ergebnisreicher Programmcode gibt mir schon eine große und sichere Zufriedenheit, sodass ich mich auch bei unaufgeräumtem Code und fehlenden Kommentaren, einer ruhigen Nacht erfreuen kann. Bis zu diesem Zeitpunkt hatte ich erst an nur einem größeren Projekt mitgearbeitet. In diesem erfüllte ich die Rolle des „Conceptualizer“. Aufgrund meines fachlichen Überblicks konnte ich damals sehr gut eine Lösung für das bestehende Problem entwerfen und diese in kleinere Aufgaben unterteilen. Hinsichtlich dessen entschied ich mich bei dieser Entscheidung auch für die Rolle des „Conceptualizer“.

Doch schon nach den ersten beiden Wochen merkte ich, dass ich dieser Rolle nicht gerecht werden konnte. Mir fehlte das fachliche Wissen, das benötigt war, um ein erstes Konzept für unser Projekt zu entwerfen. Die Wissensaneignung fiel mir in einigen Themengebieten schwer, sodass ich mir die Rolle nicht aufzwingen konnte, auch wenn ich es bedauerlicherweise öfters versuchte.

In Anbetracht der gesamten Projektlaufzeit ist mir aufgefallen, dass ich meist derjenige war, der sicherstellen wollte, dass die Gruppentreffen ordentlich und pünktlich stattfanden. Während den Gruppentreffen zeigte ich mich eher ungeduldig. Themenabweichende Gespräche brachten mich etwas aus der Ruhe. Denn Ich wollte produktiv sein und gerne schon die ersten Probleme lösen.

Im Nachhinein sehe ich meine Rolle in diesem Projekt als „Implementer“ aber in gewisser Weise auch als „Generator“. So war ich zum einen sehr erpicht darauf Aufgaben und daraus entstandene Probleme schnellstmöglich zu lösen, konnte andererseits durch meine Ungeduld meine Teamkollegen und letzten Endes auch mich motivieren.

Mit diesem Projekt konnte ich einige nützliche und hilfreiche Erfahrungen sammeln. Diese zeigten mir, dass die individuelle Rolle in einem Team von dem allgemeinen Projekt aber auch stark vom Team an sich abhängt, sodass ich bei dem nächsten anstehenden Projekt mit einem anderen Team meine Rolle wieder erstmal finden muss.

## **.Rohhullah Walizada:**

Zu Beginn dieser Veranstaltung besaß ich persönlich nicht viel Wissen und Erfahrung über die verschiedenen Rollen, die eine Person in einem Team einnehmen kann und was diese Rollen für die jeweilige Person selbst und insbesondere für das Team bedeutet.

Bei dem Kickoff-Meeting wurden die Grundlagen der Gruppen- und Teamarbeit und die verschiedenen Rollen eines Teams erklärt und dargestellt. Während der Vorstellung der Rollen, sollten wir uns selbst eine Rolle zuordnen, die unserer Meinung nach zu unserer Person passt. Ich kann für mich selbst sagen, dass es mir schwer gefallen ist mich selbst einzuschätzen und die passende Rolle auszusuchen, aber anhand der Reaktionen der anderen Teilnehmer konnte man sehen, dass nicht nur ich dieses Problem hatte.

Es wurden zwei Rollenmodelle gezeigt. Das erste Modell wurde nur in wesentlich vier Rollen aufgeteilt. Es gab den Optimizer, den Implementer, den Generator und den Conceptualizer. Das andere Modell zeichnete sich dadurch aus, dass es mehr Rollenbereiche und detaillierte Beschreibung dieser Rollen aufwies.

Ich hatte mich aufgrund der Erfahrungen, die ich in Gruppen bzw. Teams in der Schule gesammelt hatte, tendenziell als Umsetzer (Implementer) aus dem ersten Modell und als Spezialist und Macher aus dem anderen Modell gesehen.

Aber schnell wurde mir aufgrund meines Mangels an Fachwissen und fachlichen Kenntnissen bezüglich unseres Projektes klar, dass ich mich persönlich falsch eingeschätzt hatte. In meinem Team waren Personen, die viel mehr Wissen hatten und so zum Spezialisten wurden, somit war diese Rolle für mich nicht geeignet, so ging es auch mit der Rolle des Umsetzers und einigen anderen Rollen weiter. Es hatte einige Zeit gedauert bis ich die passende Rolle und den Platz im Team gefunden hatte.

Letztendlich hatte ich die richtige Rolle für mich und konnte meinen Teil zum Projekt beitragen. Ich wurde zum Koordinator und Teamarbeiter. Ich versuchte stets eine angenehme, freundlichen und lockere Atmosphäre zu schaffen um eine gute Kommunikation zwischen den Teammitglieder zu gewährleisten. Darüber hinaus versuchte ich, jedes einzelnen Themengebiet so gut wie möglich zu verstehen, in dem ich die jeweilige Person, die dafür zuständig war, aufwies nochmal zu erklären bzw. es verständlicher zu machen.

Infolge dieser Veranstaltung habe ich sehr viel gelernt und auch selbst gemerkt, dass es meistens nicht so läuft wie man es sich wünscht und die Selbsteinschätzung nicht immer die gegebene Situation widerspiegelt.

Nun kann ich mit Zuversicht behaupten, dass die nächste Teamarbeit anders ablaufen würde und ich Fehler, die ich hier gemacht habe in der Zukunft nicht mehr zu wiederholen versuche.

## **.Amir Hossein Ziaiyan Galehdar:**

Ich habe im Laufe meines Lebens in vielen Gruppen und Teams mitgewirkt. Sowohl in der Schule in den Projektwochen, wo wir in Gruppen bestimmte Projekte abarbeiten mussten, als auch an meiner Arbeitsstelle wo wir in einem Team an einem umfangreichen Software-Tool arbeiten müssen, habe ich verschiedene Rollen in den Gruppen spielen dürfen. Sowohl als Macher und Umsetzer als auch als Koordinator und Teamarbeiter, habe ich in diesen Gruppen fungieren können.

Während des Kick-Off Meetings, als uns die Aufgaben für unsere Gruppenarbeit bekannt gegeben worden, musste ich feststellen, dass man in einer Gruppe nicht immer die selbe Rolle übernehmen kann, die man früher in anderen Arbeitsgruppe hatte. Ich stellte fest, dass in mir diesbezüglich das fachliche Wissen und die nötigen Kenntnisse für unsere Aufgaben komplett fehlte und ich vor einer großen neuen Herausforderung stand.

Natürlich war mir sofort klar, dass in unserer Gruppe Stefan Schmelz die hauptsächliche Rolle des Machers und des Umsetzers am stärksten übernehmen könnte, da mir seine fachliche Kenntnis im Bereich C-Programmierung bekannt war. Als weiterer Umsetzer sah ich auch den Artjom Buschin, da er unter anderem als Tutor in Einführung in die Programmierung gearbeitet hat und meiner Meinung nach viele Kenntnisse und Fähigkeiten im Bereich Programmierung mitbrachte. Rouhhullah Walizada sah ich als Teamarbeiter und Macher, da ich bereits mit ihm in anderen Studienmodule in einer Gruppe zusammengearbeitet habe.

Ich habe mich in dieser Gruppe als Koordinator und Beobachter gesehen, weil diese Eigenschaften fast immer in meiner Person zu finden sind und ich durch das Beobachten eines Ablaufes schnell einen Überblick bekomme und danach die Gruppe stets gut koordinieren kann.

Obwohl ich gerne auch als Macher und Umsetzer lieber tätig sein würde, wusste ich, dass mir dies aufgrund des fehlenden Fachwissens in dieser Zeit nicht gelinge, weswegen ich das praktische fachliche Teil denjenigen überlassen habe, die sich meiner Meinung nach und auch während des Ablaufes des Projektes, als fachlich kompetent erwiesen, also Stefan Schmelz und Artjom Buschin.

Allesamt war dies eine großartige und sehr hilfreiche Erfahrung für mich um einfach zu sehen wie man in verschiedenen Gruppen und Teams verschiedenen Rollen übernehmen kann. Es ist nicht möglich immer die selbe Rolle zu haben, weil die Aufgabenbereiche in jeder Arbeitsgruppe verschieden sind und man immer wieder vor neuen Herausforderungen steht.

## **VI. Konflikte und Bewältigungsstrategien**

### **.Wie entstehen Konflikte in einem Team?**

Meist entstehen Konflikte, wenn zwei oder mehr Parteien gleichzeitig derartig planen oder handeln, sodass sie im Endeffekt einander blockieren und so an einander stoßen. Weiterhin können Konflikte eine *objektive* oder eine *subjektive* Seite haben.

Bei *objektiven Konflikten* geht es beispielsweise darum, dass gewisse Teammitglieder beim anstehenden Projekt verschiedene Zielvorstellungen und unterschiedliche Ansichten sowohl zeitlich als auch ergebnistechnisch haben könnten. Je länger dieser Konflikt bestehen bleibt, desto fataler fallen auch die Folgen für das Projekt aus. Ein weiterer Konflikt könnte auch bei der Umsetzungsvorstellung entstehen. Allein die Frage, ob man für die Variablen eines Codes die „ungarische Notation“ oder die „CamelCase-Notation“ verwendet, kann ein Team in angespannten Situationen teilen. Der häufigste Konflikt entsteht meist jedoch bei der Frage nach den genauen Arbeitsabläufen und Aufgabenbereichen. Die Häufigkeit und die Inhalte der Teamtreffen sind Festlegungen, bei denen man oft nicht alle Teammitglieder komplett zufrieden stellen kann und einige Teammitglieder Kompromissbereitschaft an den Tag legen müssen.

*Subjektive Konflikte* sind meist um einiges schwerwiegender als objektive Konflikte. Aufgrund von Konkurrenz, Eifersucht, Meinungsunterschieden oder persönlichen Antipathien kann es zu einem größeren Streit zwischen Teammitgliedern führen. Das große Risiko dabei ist, dass andere Teammitglieder keine Schlichtung ersuchen, sondern sich innerhalb des Streits positionieren und es dadurch zu einer Gruppenbildung innerhalb des Teams führen kann. Diese Konflikte lassen sich nicht so leicht bewältigen und erfordern eine große Aussprache im Team.

### **.Bewältigungsstrategien**

Eine Vereinbarung von Werten und Regeln innerhalb des Teams auf die sich alle Teammitglieder einigen können und versuchen werden diese im zeitlichen Rahmen des Projekts einzuhalten, können einige Konflikte von vornerein verhindern beziehungsweise mildern. Ein weiterer wichtiger Punkt ist die Aussprache im Team. Alle Teammitglieder sollten die Pflicht haben Konflikte anzusprechen. Nur wenn Konflikte angesprochen werden, können diese auch angesprochen und bestenfalls wieder gelöst werden. Besonders wichtig ist es, dass es bei der Konfliktlösung nicht darum geht, Schuldige zu finden, sondern Lösungen zu entwickeln. Dabei ist es irrelevant, ob es sich dabei um subjektive oder objektive Konflikte handelt.

Konflikte müssen allerdings nicht immer nur eine schlechte Seite haben. Objektive Konflikte bezüglich der Umsetzung einer gewissen Anforderung können durch verschiedene persönliche Sichtweisen zu besseren Ergebnissen führen und somit auch die gemeinsame Kreativität verstärken.

## **VII. Ablauf unserer Teamarbeit:**

Nach dem Kick-Off Meeting, als die Gruppen bestimmt waren standen wir vor einer großen Herausforderung und zwar die Aufgabenverteilung, Selbsteinschätzung der Gruppenmitglieder und die Funktion und Position jedes einzelnen Gruppenmitgliedes.

Jedoch stellten wir sehr schnell fest, dass Stefan, der einiges an Erfahrungen und Kenntnissen im Bereich C-Programmierung mitbrachte, der eigentliche Umsetzer in der Gruppe war.

Wir einigten uns darauf jeden Mittwoch durch ein Gruppentreffen die Aufgaben in der Gruppe zu besprechen und jedem seine Funktion und Position zu übergeben. Schwerer als gedacht kamen immer wieder Probleme und Unstimmigkeiten zum Vorschein, da sich die Gruppenmitglieder noch nicht sehr gut gegenseitig kannten.

Für Stefan war der praktische Teil selbstverständlich und während er sein Vorhaben in die Tat umsetzte und uns diese erklärte, stellten wir fest, dass wir diesbezüglich große Mängel an fachlichen Kenntnissen haben.

Natürlich bemühte sich jeder einzelne von uns seinen Teil zu diesem Projekt beizutragen, trotzdem war es, unserem Glauben nach, einfach ein Hauch von Schamgefühl, dass jeden von uns daran hinderte zu geben zu wollen beziehungsweise zu können, dass ihm fachlich an Wissen und Kenntnissen fehlte.

Nach einigen Treffen stellten wir fest, dass wir uns offen und ehrlich einander gegenüber öffnen sollen um uns gegenseitig besser kennen zu lernen und einschätzen zu können.

Danach war es uns in der Gruppe einfacher einzuschätzen wer welchen Aufgabenbereich und welche Funktionen in dieser Gruppe übernehmen kann und wir entschieden die Aufgabenbereiche in kleinere Teilaufgaben zu unterteilen, damit auch diejenigen von uns etwas tun können, die nicht ausreichende fachliche Kenntnisse mitbrachten.

Unsere Umsetzer und Macher Artjom und Stefan, kümmerten sich mehr um das Umsetzen der Ideen in unserem Programm und Rohhullah und Amir beschäftigten sich mit den kleineren Teilaufgaben und sammelten Quellen und Informationen über das Vorhaben und die Umsetzungsmöglichkeiten verschiedener Ideen für das Gruppenprojekt.

## **VIII. Fazit:**

Alles in einem war das für diese Gruppe eine sehr große, aber auch sehr hilfreiche Herausforderung. Hilfreich weil jeder von uns dadurch Kenntnisse erlangte, wie man sich in einer Gruppe als Team und als Teammitglied Verhalten und zeigen soll und wie man Strategien entwickeln kann um Konflikte zu bewältigen und zu lösen und die Gruppe besser zu koordinieren.

Aus Mangel an Fachwissen einer einzelnen Person, werden für große Programmier- und Softwareprojekte auf eine Gruppe gebildeter und fähiger Kräfte gesetzt, statt den gesamten Aufgabenbereich einer einzigen Person zu überlassen.

Die Tüchtigkeit zur Team- und Gruppenarbeit ist die Grundannahme in dem heutigen dynamischen Arbeitsumfeld und ist von großer Bedeutung und Wichtigkeit, da die Firmen und Unternehmen, aus Gründen bezüglich Kosten und Zeitaufwand, auf mehrere gebildete Fachkräfte angewiesen sind, die strukturiert und organisiert in einem Team gut zusammen arbeiten und fungieren können.

Im Laufe der Projekt- und Gruppenarbeit entwickelt man automatisch Verhaltenszüge, die einem die Zusammenarbeit und die Beschaffung einer organisatorischen Struktur für die Aufgabenziele einfacher ermöglichen.

Besonders für uns Informatiker ist das Wissen über verschiedene Aspekte und Seiten einer Teamarbeit sehr wichtig, da wir zur Erstellung und zum Aufbau künftiger Software-Projekte auf mehrere Fachkräfte angewiesen sind und mit diesen in einer Gruppe als Team gut fungieren müssen. Denn Ohne die Kompetenzen, Fähigkeiten und Vorstellungskraft anderer Fachkräfte, kann sich ein Informatiker in der herausforderungsreichen Arbeitswelt nicht durchschlagen.

Aus diesen Gründen ist Teamarbeit für Informatiker eine der größten Voraussetzungen und eine bedeutende Angelegenheit.