

Florian Thiel

1506 Jonathan Ct
Princeton
NJ 08540

us phone: (609)933 0578

home address:

Johanna-Stegen Str. 8
12167 Berlin
Germany

home phone: +49 30 7690 2427

email: us@noroute.de

MID-PROGRAM REPORT

August 2007 – March 2008

Siemens Corporate Research, Inc.
755 College Road E
Princeton
NJ 08540

Supervisor: Fusheng Wang
fusheng.wang@siemens.com

Florian Thiel

Fusheng Wang

Date

Work experience can be a valuable asset, especially in a foreign country. I will present some stories from my own experiences that will hopefully motivate more people to take internships and make the best out of it. You can have a great time while improving your professional and personal skills. Go for it!

Contents

1	Introduction – Get a job	3
2	Preparations – leaving on a jetplane	4
3	Adaptation – the first days	5
4	Work	6
4.1	What I did – Achievements	6
4.2	What I learned	8
4.2.1	Teams/Management	8
4.2.2	Processes	9
4.2.3	Architecture	9
4.2.4	Technology	10
4.3	The workplace	10
5	Life	12

1 Introduction – Get a job

In August 2006 a fellow student posted an internship offer from Siemens Corporate Research (SCR) on the internal mailing list at my university. They were looking for developers for a position in Data Management. At first I was a little intimidated by the requirements. Siemens was looking for people with proficiency in JEE, several Web frameworks and XML technologies. Most of my knowledge in these areas came from university courses. I decided not to apply at that time, as though working in Princeton, NJ (the primary location of SCR) sounded like a valuable experience.

When I later decided to apply around October I was told all open positions were taken at that time. The project team was expecting additional funding in the middle of 2007 and there would be another chance at that time. I took that as an option but wanted to make sure there was an opportunity for either practical experience or an exchange semester for the winter term of 2007. So I went to the student exchange office and applied to other companies.

I got two offers, one from Google (for the Google Engineering team in Zürich, Switzerland) and one from for Bonn but I waited for Siemens to answer because doing an internship outside Europe seemed more valuable to me.

End of July 2007 I received an email from Fusheng Wang, the lead of the Sciport project at SCR. He told me they were in fact expecting additional funding in the next days and asked I was still interested. I was. Two weeks later I had my definite offer and my starting date: August, 21st. Preparations started.

2 Preparations – leaving on a jetplane

I had about 4 weeks for preparations. Lots of time. How wrong I was. It was no problem to acquire credit cards, suitcases an international driver's license, make and appointment with the visa office (has weeks of waiting time) and so on. But the paperwork took its time. The rather strict regulations for immigration to the US made things more time-consuming and costly.

For the J-1 visa, needed for internships in the US, a certificate of eligibility is needed (certifying that you have been accepted in an approved program and have sufficient financial resources). This so-called DS-2019 is issued by a government-approved organization. In Siemens' case, this organization is CDS.

In the end, there are lots of forms to fill out, mail back and forth and fees to pay. There is one fee for the visa, one additional security fee (since the aftermath of September 11th) and insurance fees. If you want to come to the US on an exchange program, you have to have government-approved insurance. CDS sells one of these insurances. Maybe it's possible to get approval for insurance bought in your home country but it's complicated. I ended up with double insurance.

I ended up waiting for forms most of the time during this process. The final document you need to get into the US for an internship is the so-called DS-2019, documenting the purpose of your travel and the details of your work. It arrived one day prior to my visa office appointment. That was close. It would have taken days to get another appointment and my flight was scheduled just two days after.

The rest of the time between the end of the summer term and my departure went to finishing university projects, a 4-day vacation, reading up on the technologies I would need in my internship and saying goodbye to friends.

In the end, it paid off to get all documentation early and read up on immigration issues on the internet and everything you get from your future employer. It's not hard to go abroad if you use common sense.

3 **Adaptation – the first days**

I arrived at Newark Liberty International Airport in the afternoon of August, 20th. Siemens arranged (and paid for) a room in the Holiday Inn in Princeton for the first 13 nights. I had also arranged airport pickup by Fabian, a fellow intern, a few days before. It took nearly two hours to go through all the immigration procedures and my luggage was not on the plane but I was happy that everything worked out so far and too tired to worry. I left my address at the airport and they promised to get my luggage to my hotel in the night.

The 40 mile drive from the airport to the hotel took about 2 hours because it was rush-hour then. No problem, we had lots to talk about and Fabian told me a lot about the work at Siemens, driving in New Jersey, finding housing and activities in the area. We arrived at the hotel at 7pm (eastern time). I was tired and fell asleep soon.

The next day I checked at the front desk, they actually brought my luggage. Fabian collected me early in the morning to get my car, the most important tool in New Jersey. The night before I tried to find something to eat near the hotel but failed. There is no shop you can reach on foot. The Princeton area is lined up alongside Route 1, a highway, and you're really lost without a car.

In the afternoon I visited Princeton and the Campus. The town is so small, it's easy to see everything in one day.

Monday was my first workday. There was another bunch of forms to fill out, agreements to sign and trainings to take. I got my temporary badge and was introduced to Fusheng, my supervisor and David, my only co-worker at that time. They showed me around and introduced me to other people from my department (Integrated Data Systems, IDS).

Then I was ready to start working. My account was not fully functional and it took some weeks to get access to the source-code repository. We worked around this problem by doing pair-programming at David's computer, a good way for me to get to know the already large codebase (80k LOC).

Siemens only paid for a couple of nights in the hotel so I had to find a room for myself. I tried to find an inexpensive apartment with a co-intern who started working on the the same day as me but it turned out that in most developments an apartment for two was about 1000\$ per month and person, everything included.

In the end, I took over a room by another Siemens intern who was just leaving Siemens at that time. I moved in with Vanessa, an American girl. I thought it to be a good idea to move in with an American, as it would improve my language skills and also give me the opportunity to meet more locals. In a town where a big German company is one of the big employers this is not automatically understood.

Vanessa was studying at a choir college in Princeton. This meant she sang a lot of opera tunes. I really liked it, especially because she was quite good.

4 Work

I used the first month to learn about the aims of SciPort and the codebase. Understanding the aims proved to be difficult during my whole internship because there was no requirements documentation and decisions were mostly passed on as word of mouth.

In the first weeks I watched David, my predecessor, working on the code, asking questions. Soon we moved to a kind of pair-programming where I had to work on the code. I did not know all high-level design intricacies but David watched me and told me when things I was about to do were not in line with the overall design.

I found the code in bad shape. Most of the code was uncommented, sometimes the same name was used for completely different concepts, some names simply made no sense (who would have guessed a “Transformation” is simply a “Document”; and in some circumstances also a “Schmema”). There was no testing code in the 80k LOC project. Over time we also found large amounts of code duplication. David and I decided that refactoring the code would be a good idea and at the same give me the big picture. We didn’t know what we committed to.

David left after a few weeks, Daniel and later Cristobal and Chen came as replacements. I had to train Daniel and did my best but I could not explain all design decisions present in the project as some did not make sense to me (and David) either.

Refactoring the code was a huge undertaking. The code lacked modularization and stuff broke all over the place once you started changing things. Tight coupling existed in many places. Since the XML marshalling/unmarshalling was done in JAXB on the server side and in JavaScript on the client side, the mappings from objects to XML had to be kept in sync independently. There was no tool support for doing this.

I insisted on having unit testing before refactoring but it was hard to do testing when you don’t have clear modules, defined behavior and hard-to-read code. So we could initially only test a small amount of code and had to add tests after refactoring and understanding the code.

4.1 What I did – Achievements

Much of my work was SciPort programming. I’ll spare you the details of my day-to-day work and state only things out of the ordinary. As I had no formal “boss” to give orders I enjoyed implementing some (so I thought) sound Software Engineering practices into SciPort.

standup meeting I thought it would be a good idea to have a daily stand up meeting to synchronize efforts and to see what everybody was doing. It turned out to be a really good idea. Duplicate work was avoided, we could gather estimates for when a component would be finished and problems could be discovered earlier (i.e. when the plans of a team member were not in line with the project goals).

code review in a discussion we created the idea of a short code review required for all commits to the code repository. This also turned out to be very good. We could

fix a lot of problems before they could be committed to the codebase. Sometimes we also discovered flaws in documentation (someone used an interface and the interface author found out that it was not the way he intended the interface to be used and was able to fix the documentation).

unit tests Since the code was completely untested at my time of arrival the beginning I thought it to be good practice to cover as much code as possible by unit tests. I wrote most of the tests in the beginning. There was some reluctance on spending so much time on testing (most tests are longer than the code they are testing) but eventually the team could be convinced and in the end everybody happily wrote tests. In the end we had about 250 tests for about 70 methods.

vision document I was inspired by Scott Berkun's project management book to create a vision documents. Some of the team members lacked motivation and not everybody was sure what we were building anyway. I knew what a vision document was supposed to do but I never wrote one. So I did some research and come up with a document that also found appraisal by my supervisor. The team read it and it clarified some things, although it did not lead to a huge jump in motivation. I hoped it could be of use for future interns and save them a lot of time finding out where SciPort was headed.

requirements document This was also a product of having read Berkun's book. I got Alistair Cockburn's "Writing effective Use-Cases" [Coc01] from the library. I decided that Use-Cases would be a good way to capture requirements (since SciPort development was mostly feedback-driven). Most of the initial use cases were based on how things were done at that time in SciPort since there was no other requirements documentation. I was the only one writing use cases for the project, only Daniel added some in the end. I am not sure if it was too late to start having requirements when the project already had a first release but I hoped it would help future interns save some discussions about how things were supposed to work. These discussions took quite some time during my internship and often discussions were held about things already decided earlier. I hoped, people would look at the requirements in the future.

project planning I wanted for SciPort to resemble a project more than a coding session so I decided to create a project plan. MS Project was installed on all machines and I had just taken a university course on project planning using MS Project so I created a plan. At that time we had a deadline and a list of features to implement. The impressive set of features we wanted to implement fit the timebox we had quite nicely. Unfortunately we grossly underestimated the complexity of most tasks and misjudged dependencies. I updated the project plan a couple of times but in the end it was only used as a approximate reference.

Scrum I read Ken Schwaber's book on Scrum [Sch04] late in my internship and I realized that we had already adopted practices that perfectly fit into Scrum: self-managing

teams, standup-meeting (daily scrum), task estimation and responsibility to write well laid-out and tested code. The things missing to have a real Scrum process (basically iterations and a real backlog) seemed to me to make so much sense in our process. I motivated the team to try shorter iterations (before that, “iterations” took 2 to 3 months and were defined by whoever needed some feature at a certain time). At the time I left the team was planning an iteration that would take a little more than 30 days, was not based on a customer request and had up-front task planning, almost a Sprint backlog.

high-level architecture modelling Once I believed to have an overview of how things worked in SciPort I sketched class diagrams on how the architecture could be cleaned up. We discussed these changes and made a new revision. Everybody was confident the model was a good thing. We were able to implement the model during the time I was at SCR.

taks-management process The issue-tracker was full of tasks which had been in there for a long time. Some were missing descriptions, some were so vague they could not be implemented. I backed up the existing issues and started a new set with a lightweight process. Issues now had three states. “Open”, “Triaged” and “Closed”. The new state (“Triaged”) meant that the issue had been reviewed by someone else than the submitter and that missing information had been added. Only issues that were in “Triaged” state could be worked on. It was too early to say if this really improved issue report quality but I was confident that this would be the case.

In addition, I proposed issues would be self-assigned only and every developer could only have one issue assigned to him at any given time. These measures should increase commitment by developers and decrease an issue’s lifetime. Before that, no one knew which of the many issues assigned to a developer he was actually working on.

4.2 What I learned

4.2.1 Teams/Management

The SciPort project is mostly intern driven. There is no full time developer assigned to the project. This places a lot of responsibility on the interns working on SciPort. There is no senior developer to ask about architectural decisions and not project manager assigning tasks.

I now consider this a good thing for my internship. There could not have been another project where I could have taken that much responsibility and could have been able to try out my ideas for process and technological improvement. At times it was hard to make decisions in areas where I didn’t have full expertise. But in the end it was important that a decision had been made, even if it was not the optimal one. Taking responsibility and driving the project forward was more important to find a perfect solution in the first try.

It may sound mundane, but I really re-learned to appreciate team communication. It shouldn't take you an internship to learn that communication is important for team projects. But it takes a project to realize the deep impact and central role it plays. Everything done in teamwork – and especially in project management – revolves around facilitating communication. Be it having regular meetings and keeping them interesting and on topic, documenting decisions in an easily-accessible way or keeping the project transparent to all stakeholders.

Furthermore, in my position I had to care about the most complicated part of projects: the people involved. Keeping everybody involved and motivated, despite of different personalities or interests, is very challenging. But is it also rewarding. A lot! I can now appreciate the job of project managers more. Sometimes it feels like they're either sitting in their office or walking around, chatting with team members, keeping them away from work. If you look closer you notice that good project managers spend their time in the office keeping trouble away from the team (by talking to other stakeholders or upper management) and they synchronize and motivate the team by walking around and talking to people. It also helps them to extract earnest status information from the team. Creating an atmosphere of trust is an important task for project managers.

Realizing all this reinforced my interest in project management activity and made me want to gather more experience in this domain. I would never have thought I would be allowed to take all this responsibility without years of experience in team projects.

4.2.2 Processes

Reading and hearing a lot about (software) development processes at university and being especially interesting in agile methods I needed a reality check at some time. And I got at least part of it in my internship. Since we (the team) had a lot of autonomy we were able to try out some process improvements (see also 4.1).

The most valuable lesson I learned from that is that lightweight processes which do not force everybody to immediately change their working habits work best. The first thing I introduced was the daily standup meeting and the team learned that it does not take a lot of time and is quite effective at keeping everybody on track. After that, the team was willing to try out the pre-check-in reviews, although they meant more work (you had to find a teammate and explain your changes to him). After a few days, everybody did reviews as if they had been doing it forever.

I was not able to implement a full Scrum process before leaving but it looked like SciPort was getting there. I fear that in a more long-established team or with more complex processes (e.g. full-fledged XP) the introduction of new methods would not have been so easy but the “incremental” approach to change seemed to work out.

4.2.3 Architecture

I was unhappy with the state of SciPort at the time of my arrival. There did not seem to be an architectural central theme, no architectural patterns were in use and a lot

of responsibilities were spread out over multiple packages. The system did not have modules at that time.

The natural architecture for SciPort seemed to be a Model-View-Controller architecture since in the end, it was a web application. We had a lot of discussions about the correct use of the Model-View-Controller pattern (the controller must not contain any business logic and the separation between view and controller is somewhat arbitrary) and I learned a lot about the intricacies and limitations of this pattern. It turned out to be harder than I thought to correctly apply the pattern and I noticed that some applications that claim to use MVC actually don't.

4.2.4 Technology

I expected technology to be the area where I would learn most during my internship. It turned to be the other way around. Since I spent so much time on architecture, processes and refactoring, there was not much left for learning new technologies.

Nevertheless, I did my fair share of coding. I learned a lot about JAXB, the Java/XML mapper and the central role of Java Beans for many of Java's technologies. I wrote JavaScript code to build dynamic web pages for SciPort using AJAX. I got to know the popular JavaScript libraries Prototype [Teaa], Behaviour [Nol] and Scriptaculous [Teab]. FireBug [Incb] turned out to be a good tool to know for JavaScript debugging.

For static code analysis I found JDepend [Inca] useful to get a quick overview of dependencies between Java packages and cycles. I already knew about findBugs [?] and PMD (FIXME: ref) and used them. I did not know that PMD could be easily integrated into a Maven report. This turned out to be useful to keep a close eye on the quality of the code and helped minimize code policy violations.

4.3 The workplace

The SCR building is on Princeton's "Forrestal Center", an area home to 175 companies, from Fortune 500 to small startups [PFC]. About 400 people work at SCR enriching the place their more than 100 different nationalities.

Access to the building is strictly guarded and controlled by the use of Smartcards. Nevertheless, the atmosphere is mostly relaxed. The atmosphere is not exactly big-business-like since most of the employees work in research.

The workplace offers many opportunities for learning. Colleagues from different research areas (from computer vision to medical research) are usually willing to talk to you about their work. There are public talks about diverse topics, often by job candidates. Employees visit conferences and then offer to talk about it.

SCR has a private library, filled with books about computer science, medical science, optics, etc. For every research area present at SCR there is a decent stack of literatures. They even have magazines, from IEEE monthlies to German "Der Spiegel" (there are a lot of German employees at SCR). You can keep the books as long as nobody else requests them. I built a nice SciPort library in a short time.

4 Work

The Multi-Purpose-Room is the place to have lunch (if you brought your own), a coffee (coffee's free but doesn't live up to the standard of Starbucks and co.) or just have a little chit-chat (mostly work-related) with colleagues.

Last but not least, SCR even features a small gym and a ping-pong room in the basement to get your mind free for the next round of brainstorming.

All taken together, the SCR building is a wonderful workplace. I didn't know working in cubicles could be so agreeable.

5 Life

Besides the vast distances you have to go to get anything done, life in New Jersey is very agreeable. People are friendly, the weather is decent (up to the 80s till November and 60s in January). But life is not cheap by German standards. Rent was more than 500\$ and food is more expensive than in Germany. Especially fruit, vegetables and bread (if your definition of bread includes the attribute firm, at least).

It becomes much cheaper if you adopt American eating habits. Microwaveable food is cheap and so is pasta and all kinds of soda. Eating out is cheaper than in Germany but people do it much more often. I went out for lunch most of the time (adapting to the habit of most Siemens workers) which became a little costly over time.

Fuel is dirt cheap by German standards, usually half the price. The problem is that you need to take the car virtually every time when you leave the house and cars are much more thirsty (usually 12+ liters/100km).

All taken together, my intern wage didn't cover all my expenses. This was OK because I saved some money for my internship and I did not exactly live a life of ascetism.

References

- [Coc01] Alistair Cockburn. *Writing Effective Use Cases*. Addison-Wesley, 2001.
- [Inca] Clarkwaren Inc. Jdepend. web.
- [Incb] Parakey Inc. Firebug. web.
- [Nol] Ben Nolan. Behaviour. web.
- [PFC] Princeton forrestal center, about us. Web.
<http://princetonforrestalcenter.com/about-us.html>.
- [Sch04] Ken Schwaber. *Agile Project Management with Scrum*. Microsoft Press, 2004.
- [Teaa] Prototype Core Team. Prototype javascript framework. web.
- [Teab] Scriptaculous Development Team. Script.acoul.us. web.