UpWind Project Final Report

Project 2 / Spring 2012
Tomi Sarni, Anu Pramila, Juha-Matti Hurnasti and Andrei Vainik

1. Project Description

UpWind project is software development project with a goal build a sailboat navigation software. The software uses accurate sea charts and can be connected to various instruments such as GPS, wind speed and direction meters and so forth. The software will use the information to plot best possible route option for the sailboat depending on weather conditions and location. The project has been running for several years now and has been subject to multiple student projects. There is a UpWind lab for the project located in room FY1052 in University of Oulu.

Currently UpWind is going through an architectural transformation and our team's purpose was to import route calculations logic from the old version to the new architecture. We were to use Scrum method as a development method. The actual work was done during spring 2012 (February to May).

2. A General Review of the Project Process

Our main task was to import "math" part from the old system which consisted of two main components. The task was divided to be executed in four sprints varying from 3-4 weeks in length. After each sprint the plan was re-evaluated based on the progress we had made so far.

The plan of the Spring 2012 project team was to port the remaining Math-plugin to the architecture. This plugin includes functionalities of automatic long term and short term route planning. The old architecture contained these functions and so according to the plan the work was assumed to be mostly porting of the old code to the new architecture style

Beginning the work and writing the first lines of code was difficult but by the end of the second sprint we had written code towards the long term route planning and the goals seemed achievable. However, in the end of the third sprint we struggled with a coding issue and were forced to reevaluate the goals for the last sprint.

The problems we faced were mostly due to the lack of proper documentation. Therefore, a choice was made to focus solely on long term route planning and use the remainder of the time to make a documentation of the project for the future teams. The documentation now includes guides on how to compile the old and new codes, how to use version control system, where to download required software and what difficulties our team faced and how they were

overcome.

In the end of the project, the long term route planning was completed and documentation readied for the future teams so that they could work more efficiently from the beginning. The long term route planning included calculating the route, painting the route to the screen on the map relative to the zooming level and showing the image of the ship in its proper location. The ship answers to GPS coordinates, wind direction and speed. In addition, some coloring of the maps were corrected to better correspond to official nautical maps.

3. Experiences and perceptions about the software

There are a lot of practical difficulties in starting the project which could have been avoided by having a better documentation or person who we contact when we needed technical assistance and would be available at time we were at the UpWind lab. In a way it was a learning experience being forced set up the development environment yourself but in our opinion that time could have been better spent. Also it is somewhat demotivating not being able to actually tackle the work because of the equipment.

As said in previous chapter we were able to complete long distance route calculation but not the short distance route calculation. Our estimation is that, without a technical supervisor to make design decisions regarding the architecture and to assist the team in understanding the software system more comprehensively, we could have done things better. We did have a person in this technical role but he was most of the time reachable only by email and with a delay and the times he was able to come to the lab was when most of us could not.

We did try to have a good conversation about design decision regarding the software, which we were needed to be done but should have not be done by us.

We feel that the software has progressed during our project.

4. Amount of Work

The Project II guidelines depict that each of the project members have 300h to spend for the project. The realized hours have been collected in the following table. The work consisted

Name	Hours done	
Tomi Sarni	295h / 300h	
Juha-Matti Hurnasti	279h / 300h	
Anu Pramila	289h / 300h	
Andrei Vainik	280h / 300h	

5. Experiences about used methods and tools

The project was managed through Agile methods. Scrum meetings were arranged few times a week and the project was divided into four Sprints. After each sprint, a Steering Group Meeting was prepared and held. Having Scrum meetings was unfamiliar to most of the project team members but meetings were quickly adapted. The Scrum meetings helped in dividing the workload and coordinating efforts towards the common goal.

Coding environment was Qt Creator. Everyone of the project team members were at least somewhat familiar with C++ programming language and therefore applying the Qt Creator was not overly difficult. However, some of the special features of Qt left us puzzled. All of these notions and solutions we found were collected in a document for future teams, so that they need not trouble themselves with the same problems.

The version control was handled by Git version control system. In the beginning of the project we were informed that this system was to be used and we were left with an overly extensive and somewhat aversive documentation. None of the team members were familiar with the git and were thus forced to learn how to use it. Through trial and error the system was put into use but many hours were spent in learning. Had there been guidance or a brief documentation, the process of learning and time used could have been cut tremendously.

6. Experiences about the client

In this project The client was represented by prof. Samuli Saukkonen and Victor Arroyo. The original idea for the project was proposed by prof. Saukkonen as he had noticed lack for navigation software for layman sailors. In this sense Samili Saukkonen represented user and at the same time an application domain specialist. Victor Arroyo had written his master thesis earlier on this topic, and had been participating the project ever since. Victor represented the technical specialist.

Samuli Saukkonen offered support whenever we requested. Communication was carried out by email, or by visiting the UpWind laboratory whenever possible.

Victor Arroyo was in contact with project group nearly on daily basis. He offered technical support to any technical related problems that group came across. Most of the communication was done over internet chat, because Victor works daily and has a very busy schedule. Most of the visits by Victor were done on his personal time. Sometimes we would have needed more support and internet chat wasn't sufficient mean of communication to discuss highly technical and complex problems, but unfortunately we had to cope with the situation. The high need for face-to-face support was no doubt result of insufficient documentation.

7. Summary of experiences

Anu: The project taught us a lot although I'm not sure if it was about working in a project. There were no big deadlines to hit, nor there were any project funding applications to be sent. University as a workplace gives you some freedom about when you are working and maybe even about what you are supposed to do. However, it comes with a cost of writing applications, publications and tedious bureaucracy. During this project we wrote couple of documents and spent rest of the time coding. I would be thrilled if in my Ph.D. I could spend all my time just working towards my Ph.D instead of shifting papers. In that sense, this project work was not much like a project work (at a university) generally is.

We did have a nice team and we were able to work well together. There were some confusion now and then about what each one of us was doing at the moment and even some overlaps, but they were resolved quickly enough. I feel that a lot of time of the project were spent on studying work that someone else could have explained in five minutes in the beginning of the work. For example, had there been someone from the previous project team to help us begin would have helped tremendously.

Tomi: I felt at times that the whole UpWind project was put in place to teach and simulate how projects work in real IT firms. The tools and methods that we were supposed to use were similar to those being used in real firms. By real firm i mean companies that actually function under financial pressures. Putting four people that do not know each other beforehand and drop them in middle of a 6-7 year legacy software without any real support and give them a set of tasks with deadlines to do resembled a sort of worst case scenario. The learning curve to get into the project felt extremely steep. I think this sort of setting gave strength to follow through the project but it also took away motivation to do anything extra. I made a decision just to do the hours i was expected with best of my abilities and thats it. I am certain that with some different type of a project, i would have engaged the project with more enthusiasm. All in all working on project that might never see the daylight (no

financial pressure) but still try to simulate the difficulty of a poorly handled disaster project ultimately made completion of the project like chewing through bitter chalk.

Beside the negatives i felt that the chosen tools and development environment and having a place to work were the best parts of the project. So even though i had some troubles motivating myself at times to work 110%, i did appreciate the fact that the time i am putting into this project is not time wasted. All in all, i believe this project gave me much better set of skills and readiness to work in a real project in a real IT firm. In comparison to many of the other available projects in the Project 2 - course.

Juha-Matti: First of all, the biggest thing this course taught me was the meaning and necessity of documentation and knowledge management in projects. Since previous project groups had not put so much effort on these issues, it caused us a lot of extra effort to even understand what to do and how to do it. But then again, situations like these probably happen in the real world as well since switching jobs today is quite common practice. Also, working with legacy code is common today since not very often organizations implement a completely new information systems without any legacy systems. So therefore very valuable lessons have been learned on this course. But due to the reasons mentioned, I was not able to learn those things I sought to learn on this course in the first place. My goal in the beginning of this course was to learn more about C++ and especially about Qt, but not much learning happened on those areas. To be honest, this course worked the other way around: at the moment I have no motivation or interest towards learning to use Qt at all. And that probably wasn't the aim of this course.

Andrei: In the beginning I was very enthusiastic about the project, because it sounded like a real world application that had actual meaning, in contrast to usual exercise works done at TOL. The project idea sounded quite technical and as I learned to know the development environment it, sounded even more technical, which was great.

In the beginning I think all of us had problems getting familiar with the project and development environment. The software itself was around 6 years old, and had been developed by numerous project groups. This usually leads to not so practical solutions, when more than one chefs work in the same kitchen. Our job was to isolate certain component of the software from old version and encapsulate it in new version. At some points this task seemed like a nightmare, taking into account that it took me about one week just to compile the project. Also we had to learn to use new tools, which wasn’t an

easy task either. The amount of information to handle was quite large. But in my opinion this was the good side about the project, because this is what programming is all about.

During the project I learned new technologies, such as: QT and GIT and gained some experience about working in a team. Our team consisted of four people, with different programming backgrounds and we got along quite well. The client was also quite nice. The only “bad” side was the lack of documentation from previous groups and lack of immediate support.
I would recommend this project to future Project II participants.