Report and Planning

Willem Vaandrager 4175115 Elgar de Groot 4091108
Johnny Verhoeff 4137175 Hugo Reinbergen 4161173
Koos van der Linden 4133145

May 31, 2013
first draft

Abstract

A list of issues that occurred during the development of the eCoach program has been made. It cover problems we encountered with our own coding and with third party software. This includes the fact that the Avatar needs a separate process on the computer to fit into the GUI of our program and that the reaction time can be randomly slow or fast, highly dependant on the hardware of the computer, but without consistency. Another problem was the IDE Visual Studio Express which offered limited functionality. The development process was meant to be test-driven, but most of our code was GUI based, which made proper test coverage difficult. Another problem was the lack of a proper database, limiting our file upload size and speed. Finally, using GitHub for as subversion system offered it's own set of problems as well.

A good project requires a good team. The teamwork of this team focused on getting things done as fast as possible. No long meetings or drawn out discussions were had to discuss the implementations. Rather the team compromised and divided tasks right away to get to work as fast as possible. The overall experience was considered positive by every team member.

Contents

Chapter 1

Introduction

Every development project needs guidelines and preparation to be successful. These ideas need to be formed before the project itself starts off, which means that the team is required to have an idea of what the development of the product will be like. It might be the case that the team doesn't actually know for certain what is expected of them, so the will have to make an effort to estimate the process to the best of their abilities.

The goal of this document is to lay out the results of our development process. In the past months we have made a large amount of decisions to try and keep our work organized and stable. Now that we are almost done with the process and have to required program working, we will look back on those decisions and evaluate whether they were the right choice to make and learn from possible mistakes we made.

This will be done by first covering major issues we hit on the way to the end. When a team that isn't quite as experienced in the field starts a project like this, they will run into problems that they don't know the solution to. The first chapter will cover the solutions we have found to this kind of problem. Next, we will discuss how the team worked as a whole. A team project is mostly dependant on how the members of the team are capable of working together towards the goal of their project. Reflecting on the that team synergy will offer learning points for future projects. After that, each team member will pose their individual reflections on the project. They will cover what their overall opinion on the assignment was, what they have added to the project and what they have learned from this. Finally, a collection of SCRUM plans is included to show our development process and how we decided to plan things. This gives an overview of what parts of the final implementation took a long time to create and in what order the functionality was implemented.

Chapter 2

Problems and solutions highlights

During the project several problems arose that needed solving. This chapter will highlight some of those problems and their solutions. The problems that will be discussed in this chapter will be the avatar, file uploading to the database, displaying graphs, testing the GUI and using the tools visual studio and github. These problems will be discussed chronologically.

2.1 Avatar

One of the main goals of the project was to use an avatar to stimulate a patient with his/her therapy. There was no need to program this avatar, because an implemented avatar was already given. But this caused other problems because the given implementation reduced the flexibility to use the avatar.

The first problem with the avatar was to get it displayed inside one of the program's windows. To achieve this, the process of the avatar must be started separately and the process and window handle must be retrieved. The problem that arose was that rescaling of the avatar didn't always succeed, neither did linking the avatar window to a window of the program.

The solution that has been found to this problem is not a neat solution. For example, timers are used to time the correct moment for rescaling the avatar because no clear point can be determined to do this. This causes of course that slower computers do not display the avatar correctly. Another difficulty was that you don't want the avatar process to close when the avatar is not displayed anymore, because loading of the avatar takes a long time. The window of the avatar needs to be hidden and displayed again when the avatar is needed again. The solution has been re-factored several times to increase the efficiency of the use of code because multiple GUI elements use the avatar. Still a problem is that the implementation is rather specifically based on the avatar process that is used and cannot be changed easily.

The second problem with the avatar was and still partly is, to let the avatar show emotion and to let him talk. One of the problems is the robustness of the avatar itself and it still is. Quite randomly the avatar does or does not respond to the commands. Sometimes the avatar stops responding and it is not able to respond to multiple consecutive calls. Another problem is that the first time a command is given, it takes the avatar a long time to react, varying from 4 to 20 seconds. No solution for these problems has been found yet.

2.2 Visual studio Express

For the project, we needed to code in the language C#, because one of TA's had created a library with C# for connecting to the avatar, that was written in python. Visual Studio Express 2010 was decided to be the IDE in which the team would code, because it is the easiest IDE for developing in C#. When the coding began, all went well for the first couple of weeks. But then the team faced the limitations of the free version of Visual Studio. We couldn't use a database plug-in, there were no to-do lists and we couldn't use plug-ins in general. The team decided to use the ultimate version of Visual Studio, the downside to this solution was that this is not free but it had an evaluation period, luckily the period would end after the context project had ended.

2.3 GUI Testing

One of the demands of the project was that it needed to be coded by the test-driven-development rules. That means that all code should be tested including GUIs. For standard Unit testing the team found that NUnit was quite helpful, but for explicit GUI testing no suitable tool was found that was compatible with Visual Studio Express. Because the team valued automated GUI testing very highly, the team decided it was worth the effort that an own testing suite should be developed. The implementation excessively uses reflection to communicate with the to be tested GUI, for verifying and button clicking etc. The downside of using reflection, it depends on the name of the component that is being tested. So code inspection in required and if the name of the component is changed, the test no longer works. There also the name has to be changed.

2.4 File uploading

When the team decided that the therapist could record messages for his/her patients, so that the avatar could talk to the patients, the problem was encountered that for this files needed to be uploaded to the database. It was needed to upload an audio file to the server and that it could be downloaded and played. The team uses an EWI SQL database server, so only text could be uploaded. So the first sub problem was to convert an audio file into plain text. Then it was discovered that the server has a 1MB upload limit, and the team found that the audio file would be too short. So the next sub problem was to split this string into multiple sections of maximum 1MB and upload it separately. All that was needed now was to download the multiple sections, merge them together and convert them back to the audio file that was recorded be the therapist so that the avatar can speak to the patient. The next problem was to let the avatar actually speak.

2.5 GitHub

GitHub is a tool that is developed to make it easier for a team develop code with each other. But in this case the team had little experience with github, so the use of github caused some problems. Lots of hours have been lost to solving problems with git, solving merge conflicts and trying to understand what's going on. Though the experience with github has increased during the project, still some problems keep coming back. Certain files (for example the project description file) cause merge conflicts each time a merge is done. Another example is that github integration with windows is not fully completed. For example a merging tool is not installed. Because no plugins can be installed with visual studio express (because of the limitations for the free version), github could also not be integrated in the coding IDE.

Chapter 3

Reflection on Teamwork

The posture within our group was very much ’to the point’. The meetings were often brief. The preference seemed to be to quickly divide the tasks and then to go to work. For the points that needed discussion there was often quickly a compromise. There were never any real conflicts actually. This made that the atmosphere was generally very positive, but sometimes a little detached because everyone was doing his own work. The first quarter, where the main focus was on reports and understanding what our program had to do, we had regular meetings practically every project day. This declined during the second quarter, when the actual programming began. The meetings lacked somewhat in structure. We quickly discussed what had to be done and someone wrote some points down, without an agenda. This seemed to work pretty well for us actually. It sometimes made that the planning was not quite clear, although we were always comfortably on time with deadlines. At the beginning of the first quarter we had a weekly course to learn about working in a group. In this course we had to do a little test to find out what kind of team player you are. The result of our test were more or less the same, with ’implementer’ high on all our test results. An ’implementer’ is someone who thinks practically, turns decisions into actions and is efficient and self-disciplined. But it is also someone who wants to start working too soon; to begin planning and working before a concrete plan is made. It seems that this role is reflected in our group dynamic. No long meetings, quickly to work but the work is done good. In hindsight, it could have been a good idea to have appointed someone as co-ordinator for example, so we would have a more diverse group. In the end the project went well, and everyone had a positive collaboration.

Chapter 4

Individual reflections

4.1 Reflection Koos van der Linden

In general, this project was a little disappointing. The main reason for this is that my presumptions and expectations of the project were quite wrong. One of my expectations for example was that we should have to work with virtual reality and 3D modelling. My expectations turned out to be unrealistic during the project, but not until later in the project because the first part of the project was all about documenting. No virtual reality, no 3D, but a lot of trouble with getting the avatar right in the gui. Most of my time in this project has been spent on this problem. And when I solved this problem on my computer I have been busy for another while to get it working on other computers. I also had to refactor the code to make it more usable. Eventually all this code is based very specific on the program of the avatar and because of that not portable, but if the same avatar is used, I have tried to increase portability and usability as much as possible. I also have been busy with creating a sort of gui testing 'framework'. This was in the beginning of the project and I had no C# experience yet. By creating this piece of code I have learned a lot of C#, which I enjoyed, because learning new programming languages and its possibilities broadens your view. One of the most interesting things to learn about C# was reflection, the possibility to inspect and alter already compiled and running code. I am sure this will be useful later for something. I have succeeded to solve these problem and this also one of the reasons that I am content about my own contribution to the project. I have been able to finish my tasks on time with reasonable quality. The peer review also learned me that others were content with my contribution. The team working was good in general. In the beginning I thought that we lacked commitment and structure, but later in the project this improved. We still didn't have a good structure but the group atmosphere was good. With lack of structure I for example, mean that we didn't have fixed moments for meetings. Also the planning was not good. We didn't use planbox in the beginning of the project for example. During this project I have learned a lot, but other things than I expected. This project was longer than the previous on the TU Delft, and made this project one multiple ways different from those previous projects. I also learned a few basic principles of scrum, which seems useful for longer projects and teamwork.

4.2 Reflection Elgar de Groot

One of the first things I realised about this project is that it's not just about writing quality software, it is about learning to work in a group on some software. There was a lot of focus on the lifetime of a software project. When I look at just the software we had to write, I am not really exited about it. A lot of times when I thought I knew what the purpose of the program was, I was proven to be wrong. This was a little frustrating for me at first, because I had now idea what we were meant to write. I got over it and accepted it for what it was, and that seemed to work. It went better with time and had more and more fun with it. Overall, from this perspective, it was ok. But when I look at it as a chance to learn to work in a group on a piece of software that someone else requests with vague language, it is a great learning chance. There is a fat chance that this kind of situation will occur when working in the industry. You have to write some software that you don't really find interesting. You have to work in a group of random people and have to make it work. You don't have a detailed description of what the program has to do. So from this perspective I think it is a great asset to have done this project.

4.3 Reflection Johnny Verhoeff

When the project began, we first had a lot of seminars about all the different contexts. And I said to myself that I wanted to do the context with the avatar because I thought that was the coolest. While it was the coolest, it wasn't that easy to embed the avatar in de C# GUI. I started to implement it but couldn't do it alone so another team member helped me with that. In this project we had to work with scrum, which for me was the first time I had to do that. I found that this planning worked better then traditional planning, we always had a working system which was nice. We had to write a couple of documents at the beginning of the project before we even had written some code. The documents had to be updated during the project so we always had an up-to-date version of all documents. But one problem with this was that we didn't really know what we were going to build and what our limitations were. So I wrote about all sorts of cool stuff that we could do such as a local copy at the patient's computer for when they are not connected to the Internet. That would automatically sync with the main database. This was proven to be too hard to implement and we didn't have the time do to it, so we scratched that idea. This project was the first time that I programmed in C# and the first time I worked with Visual Studio. In the past I had coded in Java and I found that C# was very similar to Java, so that wasn't a major issue. Visual Studio was a nice IDE to develop in, but it's a shame that the free version is very limited, so halfway into the project we had to switch to the evaluation version of Visual Studio Ultimate. We needed to do this because else we couldn't connect to the database. We could not find a good GUI testing tool for the free version of Visual Studio so one of the team members created one with bits he found on StackOverflow. I used this GUI test to write a GUI test for one of the forms. One downside was that it uses reflection so it depended heavily on the names of the objects in the form. If one of the names in the form would change then the test failed and had to be manually fixed.

All things considered, the working environment was great. The teamwork was great, often we would joke around and we could all get along. I would gladly do another project with this team.

4.4 Reflection Hugo Reinbergen

When this project just started I had high expectations of what we were supposed to build. Using virtual reality to support the treatment of patient with social anxiety sounds like a great field to develop software for. Sadly it turned out there wasn't much virtual reality in place. The project was actually about creating an interface that displayed an avatar that was already created for us. So actually it was a project about creating a GUI and handling an external process in that GUI. This isn't exactly a bad thing, but it's not what I expected to be making.

Anyway, in the end I'm looking back at the project with content. Though it wasn't exactly what I expected, I can say that I have learned a lot from it. During this project I've focused mainly on the back-end of the program. More concretely, I've created a XML to Object parser, set up a database and created the functionality to access that database smoothly from within the program. With this, I've mainly improved my overall skills in programming in C# and understanding relational database mapping.

The project was set up to be Test Driven Scrum. This meant we had to work in sprints of 2 weeks where we made small improvements every iteration. This helped me understanding the value of proper planning and guessing the time required to implement certain features. This will be of great help in future projects since distributing your available time increases your efficiency which in turn makes you more valuable for your cost.

4.5 Reflection Willem Vaandrager

The project was a lot different than I expected it would be. In the first weeks of the project we received a lot of lectures in which we were given a background about social phobic patients and connecting with a person with virtual reality. Because of this I assumed the project would be about using an avatar to talk to patients and implement artificial intelligence in order to help the patient at home. This wasn't the case and we would build a program that should help the therapist communicate better with the patient outside sessions. We should only build a framework in which the therapist can give instructions to the program and we shouldn't get into knowing how to communicate with the patient with the avatar.

During the project I most of my time went in the therapist side of the program and connecting the two sides with the database. In the beginning we were told not to focus too much on the therapist sides, but since the project turned out to let the therapist give all the options of setting up a treatment plan, increasing the usability of the program for the therapist became a higher priority.

Although the project wasn't what I expected in the first place I did learn a lot. I had never programmed in C# before, but I got used it quite fast and my experience could really be helpful later. Our planning started out good, but in next couple weeks we split the workload into more separate parts and we planned less and focused mainly on our own parts. Because of this the planning was not always clear. We improved on this later on and this showed me the value of good planning, which would be useful for projects in the future.

Chapter 5

Lightweight SCRUM Plans

SCRUM Plan 22/3 - 5/4

In this iteration the following backlog items are covered: - As a therapist I want to gain information about the patient via an adaptable questionnaire And a part of: - As a therapist I want to monitor the patient's progress

We have created the following tasks to complete these backlog items:

1. Layout XML
   Estimated time: 4 hours
   Assigned to: Elgar

2. Patient: Layout GUI
   Estimated time: 3 hours
   Assigned to: Koos

3. Patient: Loading XML files in GUI
   Estimated time: 6 hours
   Assigned to: Hugo

4. Patient: Saving results in XML
   Estimated time: 5 hours
   Assigned to: Johnny

5. Therapist: Text to XML converter
   Estimated time: 6 hours
   Assigned to: Willem

6. Therapist: Layout GUI
   Estimated time: 3 hours
   Assigned to: Koos

7. Therapist: Viewing questionnaire results
   Estimated time: 4 hours
   Assigned to: Koos

8. Patient: Avatar in GUI
   Estimated time: 4 hours
   Assigned to: Elgar

9. Patient: Reaction avatar to questions
   Estimated time: 4 hours
   Assigned to: Hugo

These tasks are optional. These tasks will improve the therapists GUI, but are not necessary for the program and will only be performed if we have spare time.

Optional:

1. Therapist: List editor
2. Therapist: Saving XML forms
3. Therapist: Question editor
4. Therapist: Question selector box
5. Therapist: Previous question list
6. Therapist: Question tree
7. Therapist: Linking emotions to answers
8. Therapist: Drag/drop questions

SCRUM Plan 3/5 - 31/5

This iteration will last 5 weeks. In this time a lot will be done and the program should be close to finished.

Tasks:

1. Set up database
   Estimated hours: 4 hours
   Actual hours: 4 hours
   Assigned to: Hugo
2. Creating a stable connection between the program and database
   Estimated hours: 30 hours
   Actual hours: 40 hours
   Assigned to: Hugo, Willem
3. GUI testing questionnaire form
   Estimated hours: 5 hours
   Actual hours: 4 hours
   Assigned to: Koos
4. Creating framework for GUI testing
   Estimated hours: 12 hours
   Actual hours: 10 hours
   Assigned to: Koos
5. Unit tests
   Estimated hours: 4 hours
   Actual hours: 6 hours
   Assigned to: Koos
6. GUI testing
   Estimated hours: 10 hours
   Actual hours: 8 hours
   Assigned to: Johnny
7. Avatar in GUI
   Estimated hours: 4 hours
   Actual hours: 50 hours
   Assigned to: Koos, Johnny en Elgar
8. Talk functionality of the avatar
   Estimated hours: 4 hours
   Actual hours: 4 hours
   Assigned to: Willem

9. Storing and retrieving audio files from the server
   Estimated hours: 4 hours
   Actual hours: 6 hours
   Assigned to: Willem

10. Creating GUI for therapist
    Estimated hours: 16 hours
    Actual hours: 20 hours
    Assigned to: Willem

11. Show progress of patient in graphs
    Estimated hours: 18 hours
    Actual hours: 24 hours
    Assigned to: Elgar

Problems: The biggest problem we had during this iteration was running the avatar in the GUI. We expected some integration would have been implemented and this could be easily combined with the program. It turned out to create a lot of problems and it was hard to combine the two. In the future we will assign more time to tasks we dont know what to expect. This will give us more time to understand the problem and give us more freedom if we encounter difficulties during the process.