Final Report

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

May 31, 2013
first version

Contents

Chapter 1

Introduction

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

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.