

POST-MORTEM REPORT

Group 2 – Virtual Gamepad – DAT225 - 2013

INNEHÅLL

1. Which processes and practices did you use in your project?	2
2. Approximately, how much time was spent (in total and by each group member) on the steps/activities involved as well as for the project as a whole?	3
3. For each of the techniques and practices used (such as standup meetings, pair programming, TDD, etc.) in your project you should answer all the questions:	3
3.1 What was the advantage of this technique based on your experience in this assignment?	3
3.2 What was the disadvantage of this technique based on your experience in this assignment? ..	4
3.3 How efficient was the technique given the time it took to use?	4
3.4 In which situations would you use this technique in a future project?	5
3.5 In which situations would you not use this technique in a future project?	5
3.6 If you had the practice/technique in a part of the project and not the entire project, how was using it compared to not using it?	5
4. What worked well in how you worked in this project?	6
5. What did not work well in how you worked in this project?	6
6. Reflections over non process specific decisions?	6
7. How did you work together as a group in the project? What worked and not in your interaction(s)?	8
8. What would you do differently in a future but similar project?	9

1. WHICH PROCESSES AND PRACTICES DID YOU USE IN YOUR PROJECT?

When the project first started the whole team sat down and divided the project into three parts with two members in each part. The project was divided into communication, GUI and app architecture. Patrik Wållgren was voted the project manager and three lists of main user stories was written on a whiteboard which Patrik prioritized and every team was given their main tasks.

We chose to work in a form of eXtreme programming (see first section of Q3) because we like the idea of working in pairs but still working in the same room and we also liked the idea of getting useful code and great quality from the beginning.

One thing that we got a lot of benefit from was that 95% of our time spent we spent on the project we spent in the same room, normally around the same table. This made us able to keep up a verbal backlog which we used to very quickly solve small bugs and keep track of how everybody was doing. This also made us able to very quickly solve problems that occurred along the way, that we either could not solve ourselves or was not a part of the user stories that person was working on. For example, if a new button was needed (through Patrik) for connection the GUI developers could quickly push one that was immediately inserted by the app developers and the connection team could continue working.

2. APPROXIMATELY, HOW MUCH TIME WAS SPENT (IN TOTAL AND BY EACH GROUP MEMBER) ON THE STEPS/ACTIVITIES INVOLVED AS WELL AS FOR THE PROJECT AS A WHOLE?

We have aimed at spending roughly half our study time towards this project, implying roughly 20 hours per week each. We have thus in total spent roughly 800-850 hours working on the project including planning, research, development, bug fixing and report writing.

Martin Ljungdahl ~ 145 hours

Victor Olausson ~ 145 hours

Linus Lindgren ~ 150 hours (including 50 hours of Bluetooth researching)

Patrik Wållgren ~ 150 hours

Isak Eriksson ~ 150 hours (including 50 hours of Bluetooth researching)

Willam Dahlberg ~ 100 hours

3. FOR EACH OF THE TECHNIQUES AND PRACTICES USED (SUCH AS STANDUP MEETINGS, PAIR PROGRAMMING, TDD, ETC.) IN YOUR PROJECT YOU SHOULD ANSWER ALL THE QUESTIONS:

Because we did not use only one technique but instead mixed Agile methods (Scrum and eXP) into a method that worked very well for us the answers below will be general for the mixed method and not one answer for each.

3.1 WHAT WAS THE ADVANTAGE OF THIS TECHNIQUE BASED ON YOUR EXPERIENCE IN THIS ASSIGNMENT?

A great advantage of dividing the work into three pieces was that everybody got a chance to specialize within their part of work. This combined with our version of eXtreme

Programming, with two members on each team, gave us a lot of high quality material in a short time and made it possible to solve bugs quickly.

Another advantage of working a lot in the same room at the same time was that if something needed to be changed or added, that group simply gave Patrik a verbal report. Patrik then prioritized the request and filed the report to the responsible person in the relevant team. This was normally done in a couple of seconds and if Patrik found it important and gave it a high priority it was normally solved within 15 minutes.

A great advantage of using the Agile way (which Tomas Luvö recommended to us) of a release every week was that it was very clear how well the project was going and also our friends that followed us could check the release and come with some input.

3.2 WHAT WAS THE DISADVANTAGE OF THIS TECHNIQUE BASED ON YOUR EXPERIENCE IN THIS ASSIGNMENT?

A disadvantage of splitting the work into three pieces is that the group members have low knowledge about the other parts that they are not involved in. This means that they do have a high knowledge in their own part of field but if we would lose members we might lose invaluable knowledge about parts of our project.

We think a verbal backlog is a good idea if you are working close together in smaller projects because of the fast response but it is very hard to trace backwards. You have no record of what you did last week and if we would take in new members it would be hard for them to see what was recently done.

3.3 HOW EFFICIENT WAS THE TECHNIQUE GIVEN THE TIME IT TOOK TO USE?

For such a small project we found it very efficient and it worked very well for us. One example was when the plans of emulating a joystick by using Bluetooth HID did not work out. During the time when the connection team was working on a solution, the rest of the team could still work on their assignment (which is one of the great things with Scrum). So when the connection problem was solved the GUI and the code could just be implemented on top of the client/server connection.

If we would be in a bigger project over a longer time it would not be so efficient.

This method of working might not have been so efficient with different group members, as we were quite a tight group that really enjoyed working together which made it more fun and therefore also raised the level of efficiency.

3.4 IN WHICH SITUATIONS WOULD YOU USE THIS TECHNIQUE IN A FUTURE PROJECT?

We think this is a great technique for small groups in small projects, probably working best for group projects in school or sub-projects at companies.

Even if we did not follow Scrum and XP by the book we would recommend both on both bigger and same size projects. Scrum comes with the advantage of being easily overviewed and has a great way to inspect and adjust.

XP will also be great if it works for the team with the advantage of producing high quality code and continuous integration.

3.5 IN WHICH SITUATIONS WOULD YOU NOT USE THIS TECHNIQUE IN A FUTURE PROJECT?

If you would be working on a bigger project, towards a real customer or a project where the product owner is not a part of the team we would probably not recommend our modelmix.

If a customer or the project owner would like to see the backlog or if it is difficult to gather everyone at the same time in the same room, then again this method is not recommended.

3.6 IF YOU HAD THE PRACTICE/TECHNIQUE IN A PART OF THE PROJECT AND NOT THE ENTIRE PROJECT, HOW WAS USING IT COMPARED TO NOT USING IT?

We decided to go with the same technique the whole project due to great success and a lack of time to experiment with other methods.

4. WHAT WORKED WELL IN HOW YOU WORKED IN THIS PROJECT?

Splitting up in pairs, each working with a specific part of the project, worked well for us. There were three major parts, GUI, application structure and bluetooth communication. When you are in small teams you can focus on just one of those areas, you could research more and develop an expertise, thus giving the whole project a better overall quality.

5. WHAT DID NOT WORK WELL IN HOW YOU WORKED IN THIS PROJECT?

Apart from what we wrote above we also gave Asana a chance because we wanted to have a written backlog and avoid having to process all bugs through Patrik. Asana was recommended to us by our supervisor and we think it is a great way of keeping a backlog but it was too complicated to learn in the time that we had available.

6. REFLECTIONS OVER NON PROCESS SPECIFIC DECISIONS?

We originally planned to implement our app to work as a HID device, this in order to have a perfectly working joystick and to have a standard device that works with all operating systems without any specific drivers needed.

To make the Android device act like a HID device, we had to add the HID service in the SDP protocol in the bluetooth stack. This was possible on rooted devices with a tool called **sdptool** in Android versions earlier than 4.2 because Bluez was used as the bluetooth stack. However, after version 4.2 the bluetooth stack were changed to Broadcom's own stack 'BlueDroid' and it seems to be almost impossible to edit the SDP protocol after this changed. Since we did not want to support only old devices we continued researching for another solution.

After a lot of researching we hadn't found a single project that had solved this problem. So we decided that this solution was impossible in the time we had available.

It turned out that in Android version 4.3, API for GATT (Generic Attribute Profile) was added. This makes it possible to use Bluetooth LE (Low Energy) and use the specified GATT

profiles, in this case HOGP (HID over GATT Profile, <https://developer.bluetooth.org/TechnologyOverview/Pages/HOGP.aspx>). We attempted to implement a HID Over Gatt Profile (see the attached source code in *HIDoverGattProfile.java*) and the computer partly recognized the HID device.

However, it was almost impossible for us to see what we did wrong and what worked due to several reasons. We did not have access to any other working Low Energy device which made it hard to know if we had configured the computer correctly for LE devices.

In the end we had a partly working profile that our Linux computer was able to discover some of our services in. We now realized that the time required to make this work was too great. This solution would also require Android 4.3 or later and a Bluetooth 4.0 and LE compatible computer. We decided to wipe all GATT code and use RFCOMM instead. This forced us to design our own protocol which specifications can be found in *Protocol.java* in the VirtualGamepadLib project.

The GUI team made the decision to make all the layout in Adobe Photoshop. We chose Photoshop based on three different reasons. First was that both the GUI team members were familiar with photoshop from before and saw it as a great opportunity to learn more in the design field. Second was that photoshop works with layers which makes it very useful when you are designing buttons and are calculating on sizes. Third was that Android Developers.com recommended Photoshop when working with Android design, where one reason among many was that it is fairly easy to resize for different resolutions when done editing.

We also chose to look at the designs of the Nintendo Entertainment System, Nintendo GameCube and Sony Playstation gamepad when we made the graphical design. The reason for this was that these three are among (according to Game Magazine) the most popular gamepads of all time and they have all celebrated for their design. We also wanted our users to be familiar with the design since we think that makes it easier to attract new users. A third criteria was that we wanted to have different gamepads so they could be used for different kind of games (e.g. joystick for car games, few buttons design for early 80 & 90 games etc).

Testing the application has been difficult, other than actually using it. When writing unit tests, you do not want to depend on anything other than what you actually test. Since pretty much everything depends on the bluetooth, you would have to mock that connection in some way, or test the functionality directly. Since a Button depends on its Context in the

application, and the MockContext that exists throws an exception in each method, it was very difficult to write unit tests. Instead, we've focused on using the application a lot, and found bugs that way. Since our application is quite simple and small, it is a simple matter to test everything you can see. Then we could just test the "hidden" functionality using JUnit, such as insertion of the 'escape sequence'-byte at correct places in the bytestream to the server.

7. HOW DID YOU WORK TOGETHER AS A GROUP IN THE PROJECT? WHAT WORKED AND NOT IN YOUR INTERACTION(S)?

Our initial thought was to work around the MVC-model (model, view, controller), thus allowing us to easily split the workload between group members. Since there were three segments, we split into groups of two and started working on different parts: the bluetooth server, the layouts and the code that made the screen respond properly. After having started on our individual sections, we felt it was natural to continue working on our areas, since we realized that we could improve the overall quality of our project by having different areas of expertise.

Working this way allowed us to delegate our work easier, giving us specific roles to play in making the app. It is however a slightly risky method; since we were only two assigned to each section, it could have become arduous work if one of us became unable to participate at important times.

8. WHAT WOULD YOU DO DIFFERENTLY IN A FUTURE BUT SIMILAR PROJECT?

To, after every release, set realistic and clear goals for the week ahead, planned as much as possible in advance. While we managed to keep up with most weekly goals, we occasionally felt our app could have been improved and some features were made on the go as we progressed. If more details about the app had been planned beforehand we wouldn't have needed to come up with ideas and changes to the product as much as we had while actually making the app, and could instead have focused on making the current code more stable and effective. Do keep in mind that some things that we decided in the beginning of the project was impossible to implement.

If we were given more time, we would probably also have invested more time in roughly understanding the other two sections on the project, as not to segregate the workload too much.