Connection

Designing an Android application with a scrum workflow

John Andersson
johnand@student.chalmers.se
Institutionen för Informationsteknik

Elin Björnsson
karelq@student.chalmers.se
Institutionen för Informationsteknik

Johan Blomberg
johblomb@student.chalmers.se
Institutionen för Informationsteknik

Raha Dadgar
raha@student.chalmers.se
Institutionen för Informationsteknik

Gustav Grännsjö
grannsjo@student.chalmers.se
Institutionen för Informationsteknik

Jennifer Linder
linderj@student.chalmers.se
Institutionen för Informationsteknik

DAT255
Software Engineering Project
Chalmers University of Technology

Contents

1 Introduction

The number of refugees in the world has exceeded 60 million in 2015. This is the highest amount in history, according to the UN Refugee Agency[1]. Migrating to a new country involves a lot of difficult challenges, and arguably the largest of them is learning the language.

Software Engineering Project, DAT255, is a course where students learn how to design and develop software, as well as how to manage projects. All the students are given the same challenge: design and develop a mobile solution that improves, supports or contributes to language learning for newly arrived migrants.

The application *Connection* was developed by the group *Welcome*. The application's purpose is to connect Swedish-speaking individuals with asylum seekers. Swedish-speaking individuals and asylum seekers are matched together based on their current occupation or job interests. The focus of the application is to give a means for people to meet in real life. This can be easily achieved through the application's chat where users can set up meetings. The idea is to make it easier for newly arrived migrants to regulate into a foreign society while learning Swedish at the same time.

[1]UNHCR. Global trends. 2016. `https://sverigeforunhcr.se/sites/default/files/media/global-trends-2015.pdf`

2 Getting started

2.1 Application ideas

Before work began on the coding there was a three week period of time for preparation. The group didn't know how much, or what, work the mentors would do so work started on coming up with ideas for the application. The two most promising ideas were one where a 'mentor' and a 'student' would be matched in order to learn Swedish, the other idea was to show important places for refugees on a map. At the first meeting with the mentors they provided five ideas based on the research they had done. One of those ideas was the same as the matching based application, thus that was the choice. In order to differentiate it from others the application had to have some kind of unique twist. The group chose to make the matching based on your profession, allowing asylum seekers to gain knowledge about professions while at the same time forming connections.

2.2 Product vision

As soon as the group had settled on this idea, work began on a product vision in order to solidify the concept. The primary reason for this was that such a document would help ensure that all of the group's members were in agreement in regards to the application's purpose and general capabilities.

Additionally, it would also act as a reference if any member of the group would be unsure of something later on in the development process. This would hopefully help with avoiding scope creep and losing sight of the original plan as the development cycle progressed.

Finally, the third purpose of the document was to help set up a "definition of done" with the application. The group felt that it was necessary to define a concrete point in the app's development where the work could be considered completed, as a kind of finishing line.

At the same time, great care was taken to not let the product vision be too constricting, so as to still allow for some degree of flexibility should the need arise. While it was roughly specified what the application should be able to do, the group refrained from going into detail about what the feature set would look like. The choice was also made to not mention anything regarding the application's design or layout, as the members felt that this was something that would be best decided later, once the application started taking shape.

2.3 Getting started with scrum

As most group members were completely new to the scrum method of working, there was a period at the beginning of the project where the group had to familiarise itself with the scrum process and then lay out a plan for how to structure the work accordingly.

2.3.1 The Lego exercise

The first step in the group's efforts to master scrum consisted of attending a workshop known as "The Lego Exercise". At this session, our group had to undergo a miniature version of a scrum project, divided into three sprints, roughly 15 minutes each. During these, the members would pick out a few user stories, which all revolved around building a certain structure with Lego bricks, and then try to build these structures according to the given specifications within the sprint's time limit. This exercise provided a feel for the basic principles and general workflow of scrum, but most importantly it made the group realise the importance of not "biting off more than it could chew", as the members were unable to complete the tasks they had assigned to themselves in the first sprint and had to spend the entirety of the second sprint to complete them.

2.3.2 Scrum board

As the group also learnt during the Lego exercise, there was a need to keep track of the user stories, how far they had progressed, and which member was handling what. For this purpose, the group opted to use a Trello[2] board.

2.3.3 Planning the sprints

Since a significant part of the course would be dedicated to planning and preparing, only the second half of it would actually be spent working on the implementation of the application. Given this relatively short amount of time, it was decided that each sprint would be one week long. This seemed like it would be just enough time for members to make meaningful changes to the application, and it also made it quite easy to remember when each sprint was supposed to end. As such, the project would consist of four

[2] `https://trello.com/`

sprints, which seemed like a good amount to the group when considering that the Lego exercise had had three.

It was decided that each sprint would begin on a Tuesday, so as to coincide with one of the bi-weekly group meetings. Thus, the meeting could be used to discuss and review the previous sprint and plan the upcoming one. The role of scrum master was assigned to Elin as it seemed natural, given that she already was the meeting facilitator and the two roles seemed to go together quite well, given how the primary focus of the meetings would be the sprints.

2.4 Tools and new technologies

This project involved developing an Android application using the scrum method. To do this the group had to learn two primary new tools; Android Studio as an IDE and Trello as a scrum board.

2.4.1 Android Studio

The time before coding started, was spent learning Android and Android Studio. The group went through the first steps of the Android developer tutorial, creating simple activities and switching between them. At first the group were set on creating a map integrating the Google maps api. Time was spent on making a simple application showing something on a map. The rest of what the group needed to know about Android was learnt during the project.

2.4.2 Trello

The scrum board used during this project was Trello. It was used like a whiteboard with post-it stickers, allowing easy movement between different columns. The columns were divided into product backlog, sprint backlog, in progress, testing, done and stories implemented. Aside from letting any group member see and make changes to the board at any place and time, it had the additional benefit of letting the mentors and examiners see how the work progressed.

3 Application of scrum

3.1 Sprints

Sprints are a part of the Scrum methodology. The time for a sprint varies between different kinds of projects. The sprints that were set up for this project were one week long, spanning from Tuesday to Tuesday. In total, four sprints were carried out. The sprint always started with planning the sprint, which meant prioritising, setting time effort, setting up tasks for the user stories and dividing user stories between team members. The planning was documented in a document held for that specific sprint. At the end of the sprint an evaluation was held in the form of a sprint review and a sprint retrospective. These were documented in the same document for that specific sprint, so everything about a sprint would be easily accessible.

The sprint reviews focused on what had been done during the last sprint. Exactly which stories had been completed and which, if any, had tasks left to be done. This was also when the stories on the Trello board got moved to ’implemented’ or back to ’in progress’.

The sprint retrospectives however focused more on how the work sprint progressed rather than what exactly was done during it. The group discussed what had been easy/hard to implement, if anything went wrong, and also if anyone had learnt something useful that the others could partake in.

3.2 User stories

The user stories were set up with the INVEST Criteria in mind, and created from what was learnt in the lesson about how to write user stories. Every user story was written to be a “vertical slice” of the application and also independent from the other user stories.

After the user stories was set up the initial prioritising started. This was made with the stakeholder and the application as a whole in mind. What would be most valuable for the stakeholder, deliver least viable product and help us towards our vision? Before each sprint the accuracy of the prioritising was looked at again, especially if new user stories were added.

3.3 Velocity and estimations

When estimating effort for the user stories, tasks was set up that had to be done in order to fulfil the user story, then everyone estimated the effort. The Fibonacci sequence, up to 21, was used to set points on the user stories. Everyone wrote what their estimations on a piece paper then showed each other at the same time. Everyone motivated their estimations and then together decided which amount of point was accurate for the user story.

3.4 Work practises

Roughly half of the code produced during this project was done with pair programming, where two people sit together and work on a task, usually (but not always) on a single computer. Other stories were worked on by a single person or by two people asynchronously, but quite often two people decided on a place and time, met up and worked on their given user story until it was done.

A Facebook Messenger[3] group was used as the primary form of communication between the group members outside of the physical meetings. In this group anyone could get help with the task they were currently working on or ask about changes to user stories or meeting times.

Standup Meetings were not used during this project in the traditional way, because they were deemed redundant. With such short sprints and frequent communication between the group members, all the members of the team had a good understanding of what the other team members were working on or had problems with. In a way, the Messenger group served as a constant stand up meeting.

3.5 Testing

Because a great part of the application's implementation relied on Android's built in methods, the group felt as though unit testing might be difficult and not necessarily relevant in most cases. Therefore the majority of the focus was put on to user testing.
Unit testing was done once the implementation was complete rather than following a test driven development.

[3]`https://www.messenger.com/`

3.5.1 User testing

User testing was the primary testing method throughout the entire development process. The User Testers could be divided into three different groups: The team itself, Other developers and The target group. Which consists of both Swedish speaking individuals and migrants
Once each user story was implemented it would be tested by the entire team. Feedback would be given and when needed, changes would be made and any bugs found would be fixed.
After each sprint the application would be tested by other developers. This gave the team additional feedback on the application and made any hidden or missed bugs visible.
After the third sprint the application was tested on the actual target groups. Results of the test were documented and taken into account during the fourth and last sprint.

3.5.2 Unit testing

Unit testing was used after the implementation of the application in cases that seemed relevant. This was mostly where logical code was implemented. In addition, some of the specific GUI elements were tested so that they behaved as expected. As an example, making sure that the 'next' button in the wizard was activated only after the user selected a job.

3.6 Meetings

It was decided early on that the group would meet twice per week, and shortly thereafter the days Tuesday and Thursday were settled on. On Tuesdays sprint meetings were held. These consisted of a review and retrospective of last week's sprint as well as planning the upcoming sprint. The meetings on Thursdays were more flexible; the group worked on things that would benefit from having all of them present. For instance, discussing any large issues that affected the whole group or working on documentation and other deliverables.

The meetings started at 15:15 and proceeded until the group unanimously felt they had accomplished what needed to be done. Usually this meant that the meetings ended around 17:00, sometimes later. Small breaks were had about once every hour to clear everyone's mind a bit.

4 Mentors

As part of the course, all groups were assigned students from the *Interaction Design* master's programme to act as mentors and counsellors. From the outset, the group didn't know much more than this, other than the fact that it would meet them during several sessions at campus Lindholmen, the first of which wouldn't take place until a couple of weeks into the course.

4.1 First meeting

Once the group met the mentors face to face and had the chance to speak to them directly, things cleared up quite a bit. The group had three such mentors which were shared with three other groups. One of them, Anette, became the group's *primary mentor*, and the one that the group contacted directly in case of questions. Instead of just taking on a guiding role, as was initially believed, they had in fact come up with application ideas of their own, and would even complement the group's work efforts by assisting with matters outside its area of expertise. Such areas included, for instance, performing field tests and gauging interest among potential customers.

4.2 Lindholmen workshops

From this point, the group mainly communicated with the mentors via e-mail correspondence, and only met with them at campus Lindholmen twice more before the final presentation. At both of these times, they hosted workshops where both parties had the opportunity to work more closely together. At the first of these workshop sessions, the group constructed paper prototypes that mapped out the application's basic progression. This exercise, and the subsequent user testing, gave quite a bit of good feedback, and the group is of the opinion that the insights that were gathered there were instrumental into shaping the user experience that the application provides today. It also turned out to be very important internally, since it gave every member a clear, unified idea of how the application's structure should be laid out.

At the second workshop, which took place a couple of weeks later, the mentors had prepared several people who were willing to test the application. Not only did this give a clearer idea of how intuitive the application was for new users, but it also once again made the group consider the visual structure of the application. The feedback that was obtained at this workshop was the sole focus of the group's fourth and last sprint, which provided a large amount of quality-of-life improvements both in terms of usability and design.

Overall, the group feels that both workshop sessions were invaluable in making the application what it is today. They helped highlight issues which would have either been dismissed as unimportant or maybe even missed completely. The sessions were especially important considering that, given the very limited timeframe available, it would most likely have been too time-consuming for the group to try to set something similar up on its own.

4.3 Target audience testing

While the mentors were quite helpful, there was one thing they were not able to help with: getting to test the application with actual asylum seekers, who were a large part of the application's intended user base. The mentors had met and spoken with several refugees at the beginning of the course, and said at the first meeting that they were quite excited to try the group's applications. This was perfect for the group, since it meant having the perfect test audience without having to sacrifice development time to set up and perform the actual tests.

Sadly, things didn't pan out. The mentors didn't have the time they thought they would, and so, these user tests never actually occurred. For as great as the other test experiences were, the group's members couldn't help but feel that these test sessions could have given a lot of valuable feedback from a unique perspective.

5 Review of Getting started

5.1 Application ideas

As mentioned earlier the expectation was for the group to do everything, from idea to application. This led to quite some time spent at the start coming up with ideas and doing mockups of those. While it wouldn't have been wasted time either way the idea didn't change as the mentors had also come up with a similar idea, allowing work continue directly of what was previously done. This idea wasn't unique requiring the application to differentiate it from the masses somehow. The way this was done was to focus on jobs as it's very hard for an asylum seeker to find a job even if you're highly qualified.

5.2 Product vision

Looking back on the product vision now, and comparing it to the finished application, the group can confidently say that the application has not strayed outside the boundaries that were set up at the beginning. Rather, it offers a concrete realisation of the product vision's guiding principles.

One notable aspect of the application that is not mentioned in the product vision is the focus on matching people based on their occupation. While this was not a part of the initial concept, the group chose to focus the application in this manner later on in the development cycle in order to carve out a more defined niche for it. This is a perfect example of how essential it is to not write a product vision that is all too constricting. As it was now, the group had the room to make this specialisation without feeling like it went against their initial plan.

5.3 Getting started with scrum

5.3.1 The Lego exercise

After gaining half a course's worth of experience in working with scrum, all group members strongly feel that what they learnt from the Lego exercise – don't take on more work than you're sure you can handle – was the most important lesson they learnt when it came to scrum. Despite proclaiming that it had learnt from the workshop, and would try to not make the same mistake again, the group still fell into the trap of taking on user stories that the members were unable to finish in one sprint's time. This became slightly better as the sprints progressed, but the problem never went away entirely. The last sprint was the only one without unfinished user stories lagging behind, but this was simply because the group members had all decided to take on a lighter workload for that week in order to accommodate for any unforeseen issues that would undoubtedly appear.

Although it's easy to blame this on a lack of experience with scrum, there are more factors to consider. For one, the group's unfamiliarity with Android development made it hard to estimate how much time any given user story would take, essentially leaving the members with taking stabs in the dark when they tried to assign points to most of the stories. On top of that, the assigning of points itself was not without its share of complications. Due to how a lot of stories tended to end up three-quarters finished after a sprint, it was difficult to get a grasp of what the group's actual velocity was. Thus, there didn't exist an objective framework to measure the points against, and how much one point was actually worth remained a very subjective issue. As such, the group's opinion is that this problem would not have been remedied by lengthening the sprints.

5.3.2 Scrum board

Using Trello for the group's scrum board turned out to work very well. Having each card be a user story and then breaking it down into smaller tasks with the checklist feature made it easy for everyone to see how far stories had progressed. The fact that Trello made it possible to assign people to cards also made it clear who was working on what story at a glance, and the dynamic nature of the board ensured that it was never a problem for anyone to add anything to a story after it had started if there was a need to do so.

5.4 Review of tools and new technologies

5.4.1 Android Studio

The group was able to grasp the basics of Android development quite easily. The barrier of entry wasn't very high and the guides for getting started were easy to follow. During the course of the project, there were however a multitude of obstacles originating from working with more complicated aspects of Android. This is clearly shown during the first two sprints where the approximated velocity was too high as the group thought some things would be easier than they actually were to implement. During later sprints more story points was assigned to stories containing Android functionality that the group hadn't previously worked on.

Before development of the application started, time was already spent on learning how to use the Google Maps API in an Android application as the group thought that would be implemented in the actual application. This was something very specific that didn't end up in the final revision of the application rendering the time spent learning this wasted in the scope of this project. This time could easily have been focused on something more general that might have made the sprints more efficient instead of focusing on a very specialised feature.

5.4.2 Trello

Trello is, in itself, a very simple and intuitive tool to use allowing the group to get up and running with it quickly. As the sprints progressed it got harder and harder to know which stories belonged to which sprint. As a solution a color coding system was used to distinguish user stories for each sprint. Another problem was talking about individual stories as most of them had similar wording. This issue wasn't addressed during the project but highlights the importance of being able to communicate about what's being worked around. In this case a unique number on every story would have solved the problem.

6 Review of application of scrum

6.1 Sprints

The sprints were a good way to plan the implementation and adjust what to deliver, so the most valuable part for the application and stakeholder was delivered. One week sprints fit the group well. The group didn't try another time span because this one was good and effective, and also gave room for planning when to work. This was needed due to everyone having another course, with lessons on different days. Therefore, a shorter sprint would have made it problematic to actually deliver, which is the goal for a sprint.

The sprint review and sprint retrospective was a good way to review, acknowledge obstacles and learn lessons from previous sprint. By acknowledging previous faults and obstacles, improvements in the next sprint could easier be made. Potential improvements were always discussed, and a decision on how to implement them were made.

The sprint review and retrospective was constructed by the team themselves, by thinking about what to gain from these documentations. This was not made by looking at other example or similar documentation. So the efficient and quality in discovering and addressing problem areas is harder to define. Clearly problems were addressed but was any missed? It would be a good idea to find similar documents and compare, then maybe reconstruct the review and retrospective document from that.

6.2 Prioritizing the user stories

By having a lesson in how to write user stories and what to think about made it easier to thereafter to formulate them. This still took time and a lot of discussion arose how to best formulate each user story. It took time but “vertical sliced” user stories were achieved and only a few was dependent on a different user story to be implemented.

When prioritising user stories the team mostly had to depend on “gut feeling”. This because the team didn't have a stakeholder that the team had direct contact with. Instead the team took several things into account themselves, like what the team thought a stakeholder would prioritise and what was most important for the user experience.

Prioritizing took time but by putting time into it a viable application with the most important features could be delivered, after the four weeks of implementation. After the exhibition and the feedback, it seemed and felt like the prioritizing contributed well in delivering a good viable application.

Instead of everyone thinking as a stakeholder, the group could have appointed a team member to act as a stakeholder. Let that person take the information from the mentors into account, and speak to the group as a stakeholder.

6.3 Velocity and estimations

Estimating effort for user stories was difficult, especially in the beginning. The velocity for the first sprint was achieved as planned but the estimations for the user stories were not accurate. The important thing after this was not to overestimate the team capacity. That was easier said than done. In the second sprint none of the user stories were done at the end of the sprint, although they were close to finished. The team learned from this and made a bigger effort not to take on too much and estimate with the experience from previous tasks. The third sprint worked better, with more experience and better estimations for time effort whole user stories was once again delivered. So estimation of effort become a little bit easier with more experience and after having done a couple of sprints.

6.4 Work practises

The use of pair programming was very appreciated by the team members. The hassle of organising a place and time for the two of you to meet was easily offset by the increase in problem solving ability of being able to bounce ideas off of one another. It also helps to deal with code blindness, being unable to see a bug because your mental model of the code doesn't align with the actually written code. The question is if pair programming actually gives double the working speed or any increase to code quality. If it doesn't, would it not actually be better to work independently on different tasks and thus get more done in the same amount of time? This is a hard question to answer, but at the very least the team felt they had good use of pair programming.

The use of a simple group chat instead of regular daily meetings worked very well for the group. The members had an easy way of getting their voices heard and issues dealt with quickly. However, this method would probably not work very well in a larger group. If there are too many members, it would most likely become disorganized and hard to keep track of everything that's being said. In that case a more organised way of reporting status and issues every day would most likely be optimal.

6.5 Testing

User testing was applied early in the process. As soon as the paper prototype was designed user testing was applied for a better sense of how the design would actually work. This was important because the team wanted to create the easiest most straight forward application possible for the target group. Especially since a great deal of the target group were asylum seekers with diverse technological and education background.

The group received different kinds of feedback depending on the user testing groups. While the team members tested the application, feedback was often technical and related to the implementation. Team testers helped make any bugs noticeable quickly, giving the developers responsible for the user story a quick chance to fix any problematic areas. But team testing wasn't always without problems, sometimes team members would be very invested in an idea or feature which would blind them from seeing the problem areas.

Developer testing was used to receive third party insight of the application.This sort of testing was helpful in finding minor bugs and ambiguous or unnecessary context.

The downside of both team testers and developer testers was that both groups were developers and only a small part of the target group which the application was directed towards. How biased a developer's testing results could be was also somewhat questionable. Therefore the third group of user testers, the actual target group, were the most crucial testers.
The team received great feedback from the target group which was documented and worked on directly afterwards. this was the feedback that mattered most to the team since it came directly from the target group. Interface and design changes were made based on this feedback. Translation errors became noticeable and were also fixed.

The team decided not to follow a test driven development and implement unit tests after the application's implementation. This decision was made mainly because following a test driven development was difficult and would lower the team velocity drastically. Not following a test driven development lead to higher velocity but also made it very easy to forget the need for unit testing in general. This resulted into the group implementing the majority of the unit tests last minute during the last week, which made the team learn the difficult way, that not following a test driven development wasn't a smart choice.

Evidently the team would have also liked to start target group testing sooner in the process and with a larger group but this unfortunately did not happen due to the fact that the team relied on the Lindholmen mentors to provide testers and the they were unable to provide testers to test the application sooner.

Overall, because every piece of feedback received during different testing methods was discussed and acted upon, the team feels satisfied with the testing.

6.6 Meetings

The meeting structure worked well, there was always something useful done or discussed every meeting. Combining sprint review, reflection and planning into a single meeting per week allowed the team to set up a good plan for the next sprint while the results from the last one were still fresh in everyone's mind. This made it easier to gauge the severity of next week's user stories relative to the previous week.

A lot got done during the Thursday meetings, arguably a bit too much. The meetings sometimes dragged on closer to 18 than 17, and perhaps at that point it would be better to split the meeting into two separate instances to make sure people weren't getting mentally exhausted. Even with breaks, one can only do so much in a day before they get a bit drained.

7 Teamwork

As with any group project, teamwork is essential. In order to coordinate a project of this magnitude between six members, a healthy and cooperative team structure is needed.

7.1 Roles

It was decided at the first group meeting that specific group roles would be necessary to keep the project on track. The group decided on two main roles: A scrum master and a group secretary.

The scrum master would be responsible for facilitating group meetings. They would also make sure user stories and tasks were divided between teammates so that each member would have assigned tasks each sprint. The scrum master would also be the main contact person for the Lindholmen team.

The secretary would be responsible for keeping protocol meaning the secretary would be in charge of documenting all the important discussions and decisions of meetings in a clear way that is understandable for every member of the group.

Elin was assigned the role of scrum master and Raha was assigned the role of secretary.

7.2 Social contract

At the group's first official meeting a Social Contract was written to set some ground rules. During the meeting topics such as individual and group ambition level, workload distribution and problem management were also discussed and written down. This was important to do together to avoid any future misunderstanding or confusion. The social contract was also helpful in uniting the team towards a clearer goal.

7.3 Review of teamwork

The group worked well together as a team. Everyone was excited to work on the project and had the same mindset which was to develop and deliver the best possible product together.

Group members were very outspoken and there were no hesitation in voicing opinions and concerns during the development. This only worked to the group's benefit because different matters and issues would be brought up and discussed, allowing the team to come to a decision as a whole.
The team had very strong communication skills. Teammates would receive feedback on parts they had worked on and there was no uncertainty in asking for help. Help was also given without trouble once asked for.
Different user stories would be assigned to pairs with the exception of a couple of individual user stories. The pair would then decide on whether to divide the tasks or to work on the tasks together at the same time.
Assigning smaller groups of two people to each user story benefited the teammates in a way that no one felt alone while working on the tasks and always had the opportunity to ask for help directly.

During the first sprints the members worked on different aspects of the program, the longer the project progressed, teammates concentrated on more specific aspects that were either more attractive for the member or aspects that they were more familiar with. Some concentrated more on the database, some concentrated on design while others concentrated on language and testing.

8 Evaluation of deliverables

8.1 D1A – Lego Exercise

From the Lego exercise, there were three major insights that the group felt would have a significant impact on future work with the Scrum model.
The first one was:
Take time to plan and communicate properly prior to engaging the task at hand.

During sprint meetings, the group really took the time to talk about the user stories that were going to be implemented. A lot of time was also spent on discussing what tasks needed to be done for each specific user story. During the Lego exercise the team realised that everyone had different ideas of how to build the models, which created time-consuming discussions at a point where the team needed to build, not argue. To avoid this happening again, the group really made sure that everyone was on the same page concerning what needed to be done in each sprint and how to do it.

The second insight was:
Don't overestimate your velocity, and don't expect that you'll be working at maximum efficiency all the time.

Even though a lot of time was spent on planning and communicating before each sprint, the group discovered more things that needed to be added to the user stories during the first two weeks. The main reason for this was because none of the members of the group had any experience of working with Android development before, which led to unexpected surprises when it came to implementing user stories.

The team tried its best to set the story points for each user story, but some values had to change for a couple of the user stories the first two weeks after the sprint was done since they took more time to finish than expected. The group also had to change the points for the user stories where tasks were added to user stories that were not thought of from the start.

When the team had learnt Android development better, there was no longer a need to add additional tasks or change the points for the user stories. If more time had been spent on learning Android development before implementing user stories this could have been avoided. Experience is key when deciding story points.

During the Lego exercise, the group learnt that it helps to be prepared for handling a diminished working pace. Working in full force should not be treated as the default. This was implemented into the planning by checking if any group member had something very time consuming to do in any other course each week. If that was the case, that specific person took some smaller user stories for that sprint.

The third insight was:
Quality over quantity.

After the first sprint, the team quickly realised that the main focus had been on completing as many tasks as possible, which resulted into poor quality Lego models. The lesson learnt was: it does not matter how much you produce if what you create lacks quality. This was something that the group kept in mind through the whole project by not adding too many user stories per sprint and by testing what had been implemented before moving any user story to *Done* in Trello.

8.2 D2 – Half Time Review

In the Half Time Review it was written that *"in our experience there are both advantages and disadvantages of using Scrum. It takes a long time to plan and evaluate each sprint, but it's also very rewarding to have a good project structure."* Now that the project is over, the benefits of using Scrum has become obvious. Using Scrum has been greatly positive, Scrum is definitely something all group members would use and strongly recommend in future projects.

The Half Time Review written helped the group realise what had been worked on in the project and what needed to be improved. Writing the review was also a good way of communicating and reflecting on the project's progress. The Half Time Review gave some important insights to implement from the beginning into future projects.

9 The team’s way of working in relation to companies

By comparing *Welcome’s* way of working with that of the different companies that the group got a glimpse of from the guest lectures in the course, a larger understanding of how *Welcome* stand in relation to the ’real world’ can be gained. The group got a look at the work style of three different companies: Spotify, Volvo and The Techno Creatives. All three of them seem to have abandoned the waterfall model and use different styles of agile development, so at least in that aspect the group and the companies align. But even within agile development there are many nuances.

Take Volvo as an example, they work in large teams and have certain deadlines they have to meet where they combine their software development with the hardware development of the company. *Welcome* might fit in well there in some aspects, but maybe not in other. The group is a relatively small team and don’t have experience with working within a team where you can’t keep everyone 100% up to date with everything that’s happening in the project. But on the other hand, *Welcome* has worked with user stories, scrum and sprints which are all things they utilise in their development. The group is also experienced with having to communicate with a different part of a company that’s not directly involved with the software development of the project, in this case the Lindholmen Group.

While Volvo’s software development isn’t as hierarchical as other parts of the company, it is still more so than *Welcome* is, with team leaders, project leaders, a bunch of people who are ’above’ you in the company structure. This contrasts how the group worked during this course where they had no traditional authoritative team leader, and instead delegated assignments amongst themselves.

Spotify on the other hand uses smaller groups for their projects, and instead connects all these smaller groups into a larger whole while letting single employees be part of several project groups simultaneously. The group members have not been part of several different project groups at the same time before, but they are however accustomed to the smaller team structure. Spotify also does a lot of user testing and data collection to guide their development. The group members have conducted and reflected upon user tests before, but they haven’t done any large scale user data collection or learnt how to interpret that kind of data. Spotify also gave the impression that they had a more flat company structure in spite of them being a pretty large company. As said above, *Welcome* also worked in quite a flat structure, which might make fitting into their company culture easier.

The group did not get a good feel for how The Techno Creatives work from day to day. What could be gleaned from their presentation was that they worked agile and in smaller teams, but more than that was hard to pick up and compare.

To summarise, *Welcome's* experiences with agile development will most likely serve the team well going into a future career in software development, although they need to be prepared to adapt to different styles of working from what we're used to. Both in terms of team size, sprint lengths etc, but also the general culture of the company.

10 Prototype and stakeholder value relations

10.1 Application completeness

It was very important for the team to design and develop a prototype that would meet the stakeholders' value.
The application's completeness was one of the first matters discussed. A simple definition of done was written down to clarify when the application was complete. The most important factors for the group while defining done was the application's functionality and stability. It was agreed as a team, that once the application had the main functionality desired which was to connect Swedish speaking individuals with migrants and was stable, the goal to create a complete prototype had been achieved.

10.2 Graphical User Interface

The graphical user interface was another important factor which was worked on throughout the development process. The goal was to create an easy to use GUI that would instruct both migrants and Swedish speaking individuals users through the application without giving excess information. Therefore the wizard design pattern was chosen and implemented, as it seemed to be perfect for this case. Because a great deal of the application's target group were foreign non Swedish speaking users with different backgrounds, finding communication measures was a challenge. It was important that the users would have a fine understanding of what the application was able to do and what it was not. Dialogues and pop up messages with easy, explanatory text were created to solve the problem.

The team decided early on to provide the prototype in four different languages, these languages where chosen based on the languages of the countries with the most recent refugees in Sweden, according to the Swedish Migration Agency[4]. As at least one group member had experienced the false and incoherent translations of Google translate and similar applications firsthand, it was decided that human translators should be used to translate the application for a more efficient usable prototype. This took more time than estimated but was vital since it was a key

[4] Swedish Migration Agency.
`www.migrationsverket.se/English/About-the-Migration-Agency/Facts-and-statistics-/Statistics.html`

communication measure with our target group.

10.3 Relevance to vision

While designing the prototype the vision was kept in mind in all times, this helped greatly during the entire process from prioritising user stories and introducing new features to designing the interface and text formulation. The team's vision was to create an application that would focus on real life meet ups and help asylum seekers learn Swedish. During the process, great focus was put on to how to achieve this. Exactly how could this application connect people in a way that would be motivating enough to eventually lead into real life meet-ups, and how could the meetings be more interesting and more helpful for the migrants to learn Swedish and integrate into society. These were questions discussed as soon as the vision was written and concentrating on the work aspect was created as an answer to these questions. Since the group believed meeting someone in the same work field as oneself would provide the motivation and interest to both actually meet up in real life and encourage a foreign to learn the new language. Meeting with people in the same work field would also be more attractive to Swedish speaking individuals as it would play the role of common grounds as opposed to meeting a random asylum seeker.

11 Future development

The biggest hurdle to overcome with this type of application is gaining a large userbase. It requires both a solid backend capable of handling a heavy load while also making the application interesting enough to retain users. For a “big launch” to happen the application would need to be spread to at least one of the two demographics (“students” and “mentors”) so the other would have a seamless introduction to using it. The hosting of the backend would also need an upgrade.

An application is never truly complete, as there are always new features to be added. The feature with the most value to the user is matching based on current residence. This would allow the application to be used by anyone, anywhere while finding a match close enough to actually meet. Another one is integrating a map, that would show certain good places to meet, to allow users to plan meetings easily. Displaying a brief bit of information about your match, interests, age, gender for example, was requested by user that tested the application late in the development cycle but would currently be the highest prioritised feature.

12 Distribution of working time

The following table shows the amount of hours that each member has allotted to the different parts of the course.

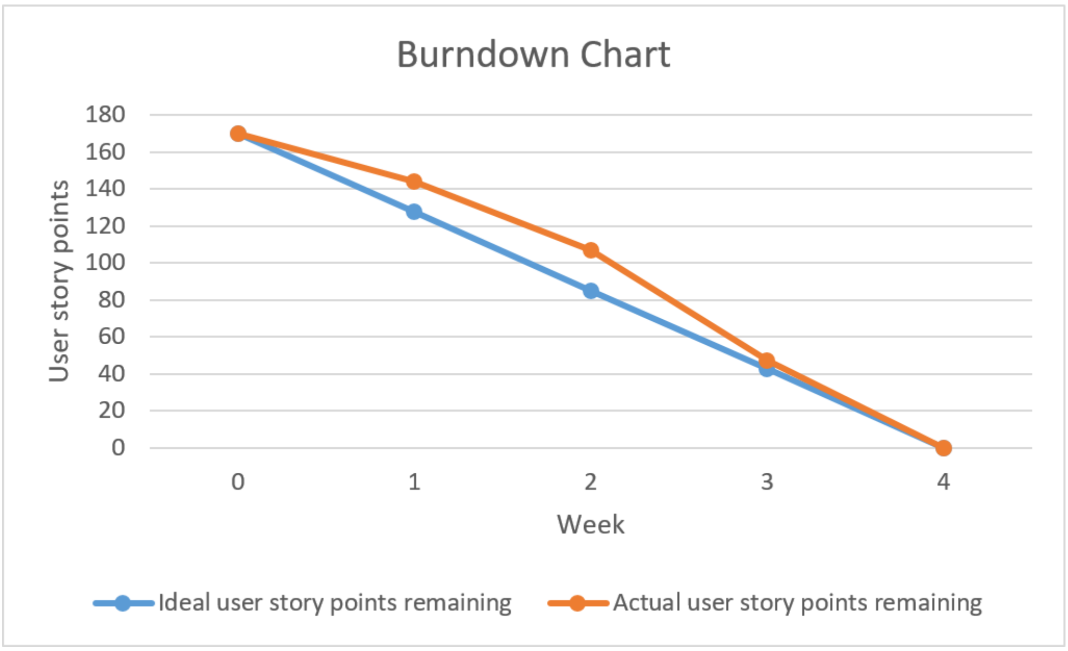
Table 1: Time table

	Lectures	Meetings	Implementation	Workshops	Report	Total
Raha	12	38	43	19	20	132
Jennifer	14	38	58	16	12	138
Gustav	18	36	27	19	17	117
Johan	16	38	28	19	13	114
John	4	36	75	15	15	145
Elin	12	38	54	16	20	140

Appendix

Burndown chart

In this chart only the implemented user stories are taken into account, not the future user stories.


Burndown Chart
User story points
Week
Ideal user story points remaining
Actual user story points remaining