

Reflection Report

by *Team Rolf*

Alex Sundbäck

Johannes Mattsson

Kristoffer Ek

Long Nguyen

Marcus Bertilsson

Oscar Hansson

October 2016

Contents

1	Application of scrum	1
1.1	Roles, team work and social contract	1
1.2	Used practises (pair programming, stand-up meetings, etc.) . .	2
1.3	Time distribution (person / role / tasks etc.)	3
1.4	Effort and velocity and task breakdown	5
2	Reflection on the sprint retrospectives	6
3	Documentation of sprint retrospectives	8
4	Reflection on the sprint reviews	10
5	Best practises for using new tools and technologies	11
6	Reflection on the relationship between prototype, process and stakeholder value	12
7	Relation of your process to literature and guest lectures	13
8	Evaluation of D1A and D2	14
9	Burn-down chart	15

1 Application of scrum

1.1 Roles, team work and social contract

During our process of developing *Eventify* we have worked with a flat hierarchy in order to increase the transparency and openness between us and our roles. We have a Scrum master who is not acting as a traditional project manager or team leader but rather as a buffer between the team and any distracting influences. The Scrum master also ensures that the team follows the accepted processes in the Scrum framework. As it stands today, we all agree that working with a flat hierarchy has been profitable since every matter of design, functionality, etc. has been a mutual decision. But even profitable processes have problems, for example it was harder to solve disagreements when we all were equals. To overcome this problem we defaulted to majority opinion or testing both opinions

In our group we also have a Product Owner that represent the product's stakeholders and the voice of the customer. Furthermore the Product Owner is also accountable for ensuring that we, as a team, deliver something with value to our customers. We found that having a Product Owner in a Scrum team is a shrewd way of making sure that each sprint delivers value for our customers. The Product Owner need to have the ability of looking at the process of the application as a customer.

A good chemistry in the group have laid the foundation for a great team work. Everyone has been helpful in the way that if someone gets stuck with something that prevents them from continuing they have been able to ask the team mates and get response related to the problem in question.

The first weekly meeting we sat down and wrote our social contract which is important so that you know that everyone in the group is working and participate at the same values. When looking back at the social contract we discussed and came to a conclusion that *"If you don't know something or need help ask the team."* has been one of the most valuable agreements. Further more we think that *"Notify the team if you can't finish something in time or won't be able to contribute because of various reasons."* has been a valuable agreement amongst each other because of the fact that if someone would get sick or by any other reason wouldn't have the chance to participate the team

mates would have stepped in and helped the person out. It is crucial that the team are aware that situation like this may occur because if you're not the team will lose valuable time and the process will lose continuity.

1.2 Used practises (pair programming, stand-up meetings, etc.)

Pair programming is an agile software development technique in which two programmers work together at one computer. Towards the end, we have come to realise that this is something that have been used quite a lot in the development of this project. A great amount of the time we put in *Even-tify* was spent during collaborative coding sessions. Here, all team members was sitting within a reach and working on their own assigned tasks for the current sprint. In this way, pair programming came naturally and several minor problems as well as a few major tasks was solved with pair programming. Many of them was a product of us sitting together working and the possibility to mention a coding related problem directly to the group. This resulted in a temporary team, of the person needing help and the one with a possible answer, until the problem was solved. Because of this, we were able to maximise the total knowledge of the whole team and members got their tasks done more quickly.

Those of us that, on paper, haven't committed as much code as others have, for example, been involved in major pair programming sessions and therefore these drafts may not perfectly reflect the contributions of each individual.

In the terms of stand-up meeting as being a daily 15-minute meeting at the same place and time, we haven't used this practise in our project. On the other hand, the purpose of a stand-up meeting was fulfilled although this exact method wasn't utilised. Due to the fact that we, as a group, are meeting each other on a daily basis, we often briefed one and other on the same kind of question. *What did you do yesterday?*, *What will you do today?* and *Are there any impediments in your way?* were automatically answered when the group got together.

We all agree on the benefits of having a daily scrum meeting when working in an office environment, but we didn't feel the need nor the possibility

of having these at a dedicated time and place due to our circumstances.

1.3 Time distribution (person / role / tasks etc.)

Participation and contribution is important in this project. It was essential to distribute the work load in a way that we could work simultaneously and deliver customer value at each sprint review.

We assigned different tasks to each person. Our Scrum master, Marcus, did a good job ensuring everyone had something to do and delegating tasks from the backlog to members of the group.

Marcus' main responsibility was to implement the Facebook login API, Firebase API and the Facebook Graph API. The Facebook login API is used to authenticate the user using their own Facebook account and the Facebook Graph API is used to pull data from Facebook's servers. The Firebase API is used to retrieve the user access token remotely from our Firebase database. This token is needed in order to communicate with the Facebook Graph API when the user signs in as guest.

Oscar's main responsibility was to implement Google's map API and location services. Google's API is used to geocode the users coordinates into a specific locality. This locality is shown in the application. Furthermore the LocationUtil class is responsible for calculating the distance to an event and then displayed in the list view.

Kristoffer's main responsibility was the implementation of the translate functionality, detail activity, calendar view. By implementing Microsoft Bing's translate API and several libraries from GitHub Kristoffer managed to create the above in a simple way. Kristoffer also pair-programmed settings with Johannes.

Long's main responsibility was to implement the list view for our main activity class. By creating a custom list we were able to choose which information we wanted to show and make it easy for user to obtain the necessary information. Furthermore he implemented the filtering functionality for the different organisations and saving favourite events locally.

Alex was mainly responsible for the UI of the application and good inputs on matter regarding the design. He was also responsible for contacting various organisations. By having suitable organisations, that were connected with our vision, we were able to have real events and keep our project real instead of just having hard coded events. Alex also wrote tests along side with Johannes. The reflection report is also one of Alex main responsibilities.

Johannes was responsible for writing tests for the application, notifications, and pair-programmed the `SettingsActivity` and most of the code connected to settings with Kristoffer.

Everyone in the group contributed to make this application a reality but as with all group projects problems arise. There was disagreements about what to implement and what not to and the best way to go about doing so. This was all natural things to occur and by discussing our points of views we could expand our initial ideas and improve them together. Bumps along the road are inevitable, for example our last two sprints were not taken as serious as they should have been. This resulted in a lot of stress towards the end of the project as work piled up. As we look back we realise the importance of keeping a structure throughout the project and not disrupt it but rather adapt it along the road. Despite these bumps we managed to deliver on our vision for the project, to create an app that manages to integrate refugees and newly arrived into the Swedish society and culture.

1.4 Effort and velocity and task breakdown

Our velocity for the project was set to 100 per sprint to be distributed to our user stories. We wanted to be careful not to overestimate or underestimate the effort for each task in the backlog. In retrospect we can easily conclude that we had a tendency to underestimate the time required for our tasks. An example of this is the Google maps API, which was harder to implement than first thought. Main reasons for this was lack of previous experience with implementing an API with android. The aftereffect of this was that work began to pile up and resulted in stress among the group members. We resolved this by consulting with each other and after a lot of reforms to the code we managed to get it to work. Initial time estimates rarely reflects the real time that ends up being required to implement a component.

In the beginning of the project it was difficult to break down our tasks into small manageable tasks and it still is. The goal is to always break down the tasks into small vertical tasks. It was a new way of thinking about deliveries and to always have something of value to the customer. We started out with horizontal slices and broke them down into vertical slices. In the case of a feature being too large we created sub-tasks of that specific feature in order to be able to deliver something each sprint cycle.

2 Reflection on the sprint retrospectives

Our sprint retrospectives issued by our Scrum master every Monday was the time where we discussed the just-concluded sprint and determined what we could change in order to make the next sprint more productive. The main questions for discussion during our retrospectives were: what went well during the sprint cycle, what went wrong during the sprint cycle as well as what could we do differently to improve. The sprint retrospective is an important mechanism of the process that allows our team to continuously evolve and improve throughout the life of our project. On our retrospectives everyone had their chance to air their opinions in an open, honest, yet constructive atmosphere. Everyone participated no matter if you were a team member or the Scrum master. We noticed that encouraging the "feedback" loop of the just completed sprint is critical to facilitating a model of continuous improvement. We have learnt that no matter how good a Scrum team is, there is always opportunity to improve.

Even though we knew how important the "feedback" loop is we could still have been better at it. Sometimes we by any means skipped to really discuss in depth how we can improve our sprints and become more effective. We are not satisfied that we sometimes skipped that really important momentum and if we would do another project we would pay more attention to the matter in question. One way to accomplish this is by sitting down with the team and discuss with each other why it is so important with feedback and come to an agreement with each other that after each sprint we are going to discuss in depth what we did good and what we can improve. But, after all, we learned that the "feedback" loop is really important if a team wants to improve their efficiency, way of working and behaviour to tackle various sorts of problems that may occur throughout the way.

One issue we had to tackle in the beginning of our project was of the fact that some of us did not really take the sprint retrospective as something important. Some members of the group laid almost all their focus on the development of the application and paid no attention on the reflection. This was an issue because some of us knew that reflection and the sprint retrospective is one of the key elements for improved efficiency and for a successful project. We solved this by sitting down and discussing the matter. This was a very important step for our Scrum team because of the fact that after the

discussion we all agreed that the sprint retrospective is a really important element. But, as mentioned earlier we are not fully satisfied with our sprint retrospectives in the way that we could have done so much more of them - so this is something we absolutely will bring with us for future projects.

3 Documentation of sprint retrospectives

Sprint 1

In the beginning of our project we had a hard time getting started. We wasn't really sure how to structure the backlog or divide the tasks between us. But as we started to grasp scrum as well as who should do what, we were able to deliver something that had customer value at the end of the sprint. Things like plan the application in detail and understand what the customer wanted was the main focus here. At this point we didn't have any persona's and in the retrospective we mention this as something we should improve. In the retrospective we also mentioned that we could have started earlier, as the first days wasn't so productive.

Overall, the effort points were well estimated and everyone had something to work on. Members were able to ask each other for help and got fast and useful responses.

Sprint 2

At the end of sprint 2 we connected every part that each of us had been working on. The result was a prototype which ensured us that we were heading the right way and kept in line with the vision. However, many task this sprint were depending on other task to be completed before proceeding. This led to that some members had to wait for other members to finish their task. We made sure that this wasn't going to happen again, as this was inefficient and preventing us from reaching our potential velocity.

Another issue we mentioned during this sprints retrospective was the fact that some task in the backlog were underestimated and took much longer to complete. This was mainly because some members started to work with new technologies and libraries. After this we made some changes to the effort points we had put on some task, as well as pointing out that, when estimating a task that we had no previous experience in, we would slightly overestimate instead.

Sprint 3

Sprint 3 a was very productive sprint, in terms of how much the application developed. Both design and functionality was falling into place. Nevertheless, as productivity became better in the sort term, we started lacking things

in other fields. The new tasks that had been added, as the project went by, were never assigned any effort. As an effect of this, some members in the team weren't able finished their task and others took on new tasks during the sprint. The last one led to some confusion in the group on who is doing what. Another problem that was raised during the retrospective is that a lot of the code committed this sprint were never tested before merged. As it showed later on, this produced many bugs that could have been prevented at an earlier stage. We realised that both effort and testing are pieces that plays a major part in how well a software project goes.

On the other hand, some things were good, we introduced pair-programming into our project and it turned out to be were useful to some. Those who had gotten stuck paired with someone that had more experience in the specific task, and they were able to learn from each other as well as develop the application.

Sprint 4

In the last sprint, time started to catch up to us. As we only had 2 days until the final presentation we all could sens the stress. Members started working on the things they thought were most important at the moment. The backlog and efforts was more or less ignored due to this. Although things got a little bit out of hand in the end, the great communication between the group carried us. Communication is something we now see as a key factor in having a successful scrum team.

After the final presentation the work mainly consisted of quality assurance of the code and it's structure. Without the time pressure we now resumed to work with the help of the backlog and our efforts.

4 Reflection on the sprint reviews

At the end of each sprint all team members gathered to show what they have accomplished. The participants during our sprint reviews included the Scrum team, the Scrum master and the Product Owner. Usually the management and other related people participate but that was not the case during this project. What is important for us and are included in the Scrum framework is the fact that during these sprint reviews you are required to deliver a potentially shippable product increment. This means that at each sprint review we have produced code that is tested, functional and a usable piece of software and this is why we thought it was so helpful to have a Product Owner. The reason for the importance of delivering something with functionality and that is usable is because you need to show some customer value. Ideally, if we completed every product backlog item brought into the sprint we would have some added. We have learnt that if you make sure that you have customer value at each sprint you will always make improvements on the applications functionality and design - and that is what really matters for the customer.

We are really happy and satisfied with the result of what we had to show off on each sprint review. Sometimes we had not finished all product backlog items but we still made sure that we always had some customer value and some functionality that is usable by the customer on each sprint.

Now that we have a better understanding of how to work in an agile work environment we can use this for upcoming projects to better estimate and plan ahead. This can be achieved by not delegating too heavy and many backlog items to one person, as well as understanding the importance of adapting the velocity and estimates for the user stories as the project progress.

5 Best practises for using new tools and technologies

As with everything new there is a learning curve and this curve might be steeper for some. In the beginning of this project a few of the group members had previous experience with coding for Android, which was a good thing for the remainder of the group as this opened up the possibility to share experiences and teach each other. We believe that this is one of the most effective ways of learning, to have a first-hand source to learn from and together try to implement these ideas.

Using new technologies and tools can be hard. Sometimes the best way is to use solutions that already exists, instead of creating it allover again and reinvent the wheel. In the case of our application we have decided to use both parts of Google's maps API and Facebook's API in order to create our seamless flow of events. This makes it easier for us as we are not responsible for the creation of events. Without the use of external APIs we would not have been able to produce an application of this extent in the time-frame that we had. It allows for us as developers to directly implement something of value which ties in with our tasks or features. Using tools that are tested and developed by the community or trusted companies is in most cases the optimal thing to do, unless a very specific solution is required for a specific project.

6 Reflection on the relationship between prototype, process and stakeholder value

In the beginning, we have a hard time making decisions regarding different functionality. After the second exercise at Lindholmen, our supervisors asked us to create persona's which became a huge help in identifying and analysing our user's needs. By understanding our user's needs, our goal has become easier to visualise and work towards. For example our first persona had basic English knowledge, was from the Middle East with prior knowledge about smart phones. So we had the freedom to use basic English in our menus and a pretty standard layout design for our application. Since we did not want to exclude users without basic English and smart phones knowledge the second persona was created. The highest priority with this persona was to use icons more and later offer full translation of event descriptions in a range of languages. This allowed users like our second persona to navigate in our app without too much of a problem and access vital information.

From the beginning our entire process focused a lot on stakeholder value in each task for each sprint. We wanted the user to experience a difference in the application each time we completed a task. A big challenge in this project was the fact that we didn't have any real stakeholder. We had to act as our own and was always trying to place our self in the shoes of a imagined product owner.

7 Relation of your process to literature and guest lectures

During this project we have laid minimal effort and time on literature relating to technology like Git and Android. The reason behind this is because we felt that we, as a team, had the fundamental knowledge and experience of Git and Android in order to make a prototype with very good result. If someone would get stuck along the way due to lack of knowledge, the person in question would first asks its team mates for help. If, however, no one in the team has that specific knowledge in order to solve the issue, then the person would go and find appropriate literature that would enhance the knowledge that is needed to solve the problem.

What we did spend effort and time on is literature regarding the knowledge about the Scrum framework. To get the knowledge we needed we used an article by Gojko Adzic called *Splitting user stories – the hamburger method*, an article by Jeff Patton where he talks about *The New User Story Backlog is a Map* as well as articles talking about Scrum and its framework in general published by Mountain Goat Software. We found that all these articles were very helpful for us in order to understand Scrum and its values.

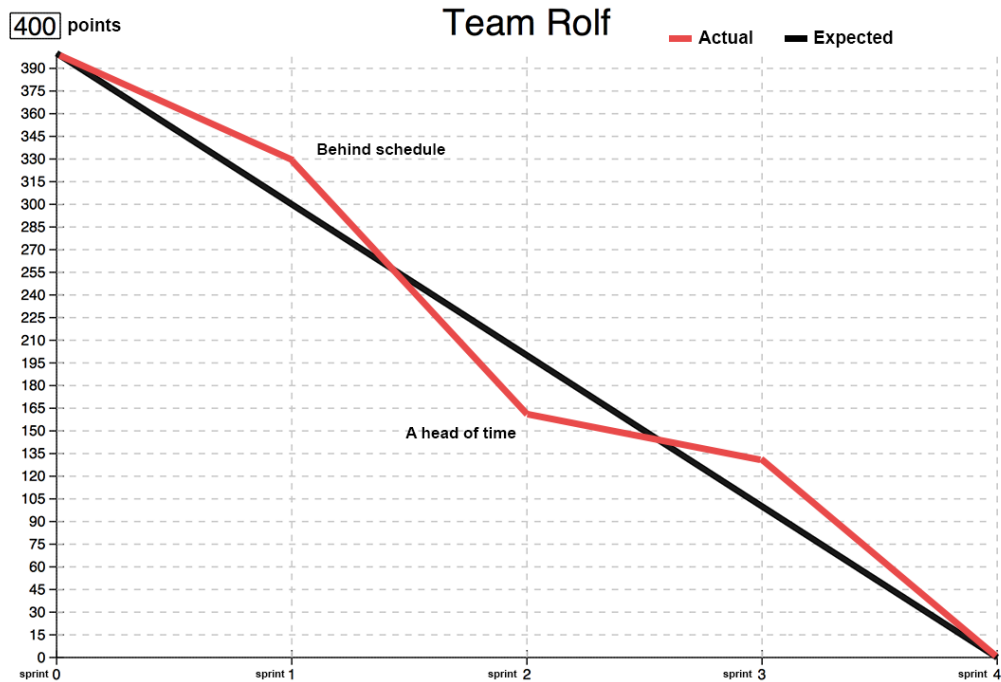
We attended all of the guest lectures and we found them very interesting but we felt that they were not connected to the course as we hoped that they would be. If you ask us, it would be more appropriate if the speaker on the guest lecture talked about working in an agile project and so forth. Anyhow, they were interesting and shared some valuable knowledge but we felt that they were not related to the core of this course.

8 Evaluation of D1A and D2

The D1A, also known as the "Lego exercise", was according to us one of the key moment in the course in order to understand the fundamental values of Scrum. It was the perfect exercise to understand Scrum in the way that you were allowed to take the framework and apply it to a smaller more playfully project and learn the core values of Scrum during the process. The first sprint during the D1A was, if you ask us, chaotic because we did not pay so much attention on our effort, velocity and task breakdown which made it difficult for us to estimate what we as a team were able to accomplish under that eight minutes. Further more we did not pay any attention to customer value during the first sprint which resulted in that we had nothing to present for our customer. We did not have any dialogue at all with the customer which do not really make sense looking back at it now. When the first sprint was over we gathered and reflected on it and discussed some improvements that we could bring with us to the following sprint. We rated all user stories, came up with a velocity, had a plan for how we should encounter to talk with our customer, added items to the product backlog and we also made sure that at the end of the eight minute sprint we would have something with value to show to our customer during the sprint review. At the second sprint retrospective we came up with even more improvements in order to make our process more efficient and in the last sprint we had improved so much compared to the first; we acted as a team and had a clear plan over how we should accomplish as much as possible in a short period of time. This is they key of reflection - going from chaos to order just by reflection and team work.

The Half-time review were an opportunity to take a break and reflect over the past weeks. We found this momentum helpful since it gave us a push to work even harder. When we wrote the review we realised that we were forced to reflect over matters that we would not normally evaluate. In the report we discussed and reviewed different topics like our backlog, our work load as well as the value for the stakeholder.

9 Burn-down chart



As mentioned in the section "Documentation of the sprint retrospectives" we had a hard time getting started. During this period of time we also dedicated most of our time to the course we all studied in parallel to this. Because of this, we went slightly behind schedule. In Sprint 2 we got back on track and was now a bit ahead of time. In sprint 3 we added several new tasks to our backlog. This made us fall behind the schedule again. Towards the end we caught up and had completed all of our tasks in time for the final presentation.