# Final Report

Group: Laser Disc
Names: *Louise Tranborg*, Henrik Johansson, *Erik Blomberg*, Sam Noble, Felix Backgård, Carl Bergman, Jacob Svensson
Teachers: Jan-Philipp Steghöfer, Håkan Burden
Date: 25/5-21
Course: DAT257

The reflections for each topic are written on the form below:

*A: the current situation or "what is"*
*B: what you want the situation to be or "what might or should be"*
*A→B: a plan for getting from where you are to where you want to be or "feedback designed to reduce the gap"*

# Customer Value and Scope

**The chosen scope of the application under development including the priority of features and for whom you are creating value.**

A:
The stakeholder was HAST. The team had some meetings before talking to HAST and went into the meeting with HAST thinking a mobile-app was going to be made. HAST came into the first meeting not really having a clear idea, instead they were very open and flexible. HAST and the team brainstormed and came up with many ideas, like a gamification, “plan your week”-app, a reflection quiz, “a message bot” etc. All these projects create value for them and their customers. The team and HAST realized very soon that the project needed to be the minimal product and therefore be more specified and small. HAST also said that many of their customers will use a computer and therefore rather wanted a web-application than a mobile-app. Therefore the plan was changed and a web-application was decided.

Today HAST sends out word-documents to their customers so they can do a self-evaluation before and after the course that they have taken. This is not optimal because it’s done manually, both to answer the questions and also counting the final score. The big scope for the application was therefore decided to be a web-application where the customers can do a self-evaluation easily with their computer at home. HAST prioritised it to work on the web on the computer, but really wanted it to work on a cellphone as well (at least in web browser on the phone).

There were some priority changes during the project. One example was that HAST first said they really wanted to change text on the application by themselves, to create new reflections when needed. The team worked for this and prioritised that in the sprints. Then in one meeting, HAST brought up emails and that collecting emails was the aim for the application. The scope was therefore changed, since the aim appears to be changed. The team and HAST got to the conclusion that email is of greater prioritising than changing reflections. The next sprint, the team let go a little of the changing in data, and worked on the email-implementation instead. The meeting after that it came up that HAST indeed found the changing in data very important, so the team had to reprioritise again. This event is a great example of a miss in communication. The team made it clear that both were difficult to implement and almost only one of them was possible, but HAST thought that the implementation of email did not affect the implementation of changing in data. This was solved with more communication, and the email-implementation was changed to a link to their website instead. The conclusion was a well prioritised product for the stakeholders.

B:
It is important to have a specific scope and make it clear for everyone in the beginning of the project. This means that the project scope will not change drastically during the project. It is also important that everyone has the same expectations so both the team and the stakeholder feel satisfied. For future projects it is wanted that all the prioritized features will be done. This project was almost so, but there was no time for some features HAST really wanted. This is probably because of the relatively small size of this project. More frequent meetings with the stakeholder is also a dream scenario to improve the communication and be able to adjust the scope fast if needed.

A → B:
For future projects it is important to have regular meetings and also to make it clear so all understand the purpose. There can not be too much open communication. More meetings and effective communication will improve the scope and expectations of the project drastically.

**The success criteria for the team in terms of what you want to achieve within the project (this can include the application, but also your learning outcomes, your teamwork, or your effort)**

A:
The finished product follows the success criteria well and the group and stakeholder are satisfied. One big feature, that was implemented, was the possibility for the stakeholder to create new evaluations. The UI also looks good and the product is ready for the customers. One feature HAST wanted as a next step was implementation of the possibility for the user to send the result via email and at the same time give HAST the email. There was no time left for this implementation, and would have required more sprints.

The group members have learned a lot from the project. Every week the members rotated the roles so everyone in the group could try to be product owner, scrum master etc. The approach worked well, and everyone understood how it was to be in different roles. While programming, pair programming was used. The group was divided into smaller teams, about 2-4 people. The size of the team was different and dependent on the degree of difficulty and scope of the user stories. This approach helped to improve teamwork and participation within the group.

A success criteria for the teamwork aspect of the project was to be able to work with Scrum without it becoming a hindrance for the progression of the project. At the beginning of the project, before the group had managed to define clear ways in how the roles and aspects of the teamwork framework should work, the use of Scrum could

often come in the way of the actual development of the product. For example, in the beginning the different roles were difficult to understand and use correctly. This improved during the project until a point was reached where the way of working, Scrum, became less of a hindrance and more of a tool to be able to make the most of the time spent.

B:
One goal can be to always have some type of learning outcome from each project, for example in teamwork, some new skill or gain experience. For future projects a clear plan for what the group's success criteria is can also be helpful to keep everyone on track and be engaged. In this project no time was set to fully discuss the success criteria, so that is something needed for future projects.

A → B:
For future projects the group can make a list of success criterias in the beginning of the project, that covers what the group wants as a whole but maybe also on a more personal level what each member wants to improve or learn. This does not need to be very formal, but more so everyone gets a time to reflect. Then during the project some time can be used to look back and reevaluate.

**Your user stories in terms of using a standard pattern, acceptance criteria, task breakdown and effort estimation and how this influenced the way you worked and created value.**
A:
The group started with a user story template from a previous course. The template, and each user story, had a unique ID, a name, difficulty level, a description of the story (as a X, I want Y, because Z) and acceptance criteria, where both non-functional and functional criteria were included.

In the beginning of the project the user stories were fully complete and according to the template. However after a while the user stories started to deteriorate as there were many small fixes that were necessary to do, and therefore needed user stories. For example a user story might consist of only an ID and a name and this could be enough as all members of the group knew approximately what was needed.

The user stories were intentionally written to be independent from each other to make it easier to work on simultaneously. During the project user stories were deliberately chosen together so that they would not intersect, this to ease the merges and to separate what the group is working with.

In the beginning the estimate level was problematic to assess, because no one had previous knowledge working with Flutter and Dart. Consequently, this led to early user stories being quite abstract and open-ended. The user stories' estimate levels were graded from a scale of 1 - 10, thus they were distributed accordingly during the progress of the project. For example, a person could take on multiple user stories with lower estimation levels, but a person would be satisfied with solely one user story if it had a high estimation level.

The Scrum board was divided into multiple rows, describing the current state of the user story. This project had: Product Backlog, Sprint Backlog, In Progress, Test Run, Code Review and Done. Starting the project, the group was not executing code reviews thus Trello only had a Test column. But as the project progressed and code quality became more important the group had to split it up into two columns.

B:
Next project the user stories have to be written completely to make it easier to work with. It will always be difficult to assess the estimate level in the beginning but in the perfect project the tools are well known so it will be possible to make an accurate estimation immediately. The template and the structure of the stories in the project felt as if they were good and clear so that will be kept.

One thing to think about is whether the Scrum board size needs to be increased if there are added epics to a future project. As the project was fairly small there was only one epic on the board at all time. Perhaps more columns could be added to more accurately track a user story's current state.

In the next project, the product owner role should take on the responsibility of proof-reading and checking that the user stories are firstly, still relevant and secondly they are clearly defined. This is to avoid open-ended, small user stories that are not clearly defined. Perhaps, creating a different space for jotting down ideas to avoid filling up space in the Scrum board.

In larger projects, someone needs to be responsible for the Scrum process, bringing forth ideas and keeping them up to date.

A → B:
To reach the goal of B from A, the group members need to concentrate and make sure that they follow, and complete, the template when writing new user stories.

To simplify this for the group members it would be important to keep someone in the group more responsible over the scrum board and the user stories. When it is an assigned task to look it over it would be more proper than the current project where it was the wild west on the scrum board.

A suggestion would be to have the product owner keep more control over the scrum board, this would also increase the importance of the role as it was almost neglected during this project.

To prevent short and open ended user stories it would be good to have a place to write down ideas, this however also requires an assigned task to review it so that the ideas get read and not just ignored. The product owner could do this as well.

**Your acceptance tests, such as how they were performed, with whom, and which value they provided for you and the other stakeholders**

A:
In the beginning of the project the group set up criterias for definition of done (DoD) in Trello. The group always checked with the criterias and if all of them were met, the user story was done. This was changed a few weeks later, after a discussion with the TA, the group added code review to the acceptance criterias. The code review was done by another person that did not work on the same user story. The criterias of the code review was functionality, that the code was commented and easy to follow.

The code review was new to all of the team members. In the beginning the group did not know how valuable the code review was, for both the team and for the stakeholder. The acceptance criterias makes it easier for everyone in the group to work with the project and to understand what the code does. This was important when other persons added features to the code that they had not worked with before.

The structure of the code review was not very clear. The group had no time set aside to do this. Due to this it was a little bit confusing for the person whom would perform the code review. The code review went better in the later part of the project, but can still be improved. Just as a good way for code review was implemented, the last sprint was around the corner. The last sprint had many small fixes and not many structural user stories, which made it difficult to do the rewarding code review correctly.

The code review and definition of done created value for HAST due to better quality of code. The code is well commented, files and methods are of decent size, and the code is future-proofed. HAST can now with ease work further with the project without bumping into big and expensive problems.

B:
For future projects, a clearer structure for acceptance tests and more area-specific frameworks such as a code review is needed. As well as making sure that these acceptance tests actually take place when they are meant to.

A → B:
To reach these goals for future projects, the structures for DoD and code review should preferably be defined early in the project. Which is in contrast to this project where the code review was first defined quite late in the project. Something that can be done in order to achieve this goal is to schedule a specific time when the code reviews for that week's sprint should be done. As well as making sure that there is someone in the group that has the responsibility of ensuring that the code reviews happen and follow the guidelines set up by the group.

**The three KPIs you use for monitoring your progress and how you use them to improve your process**

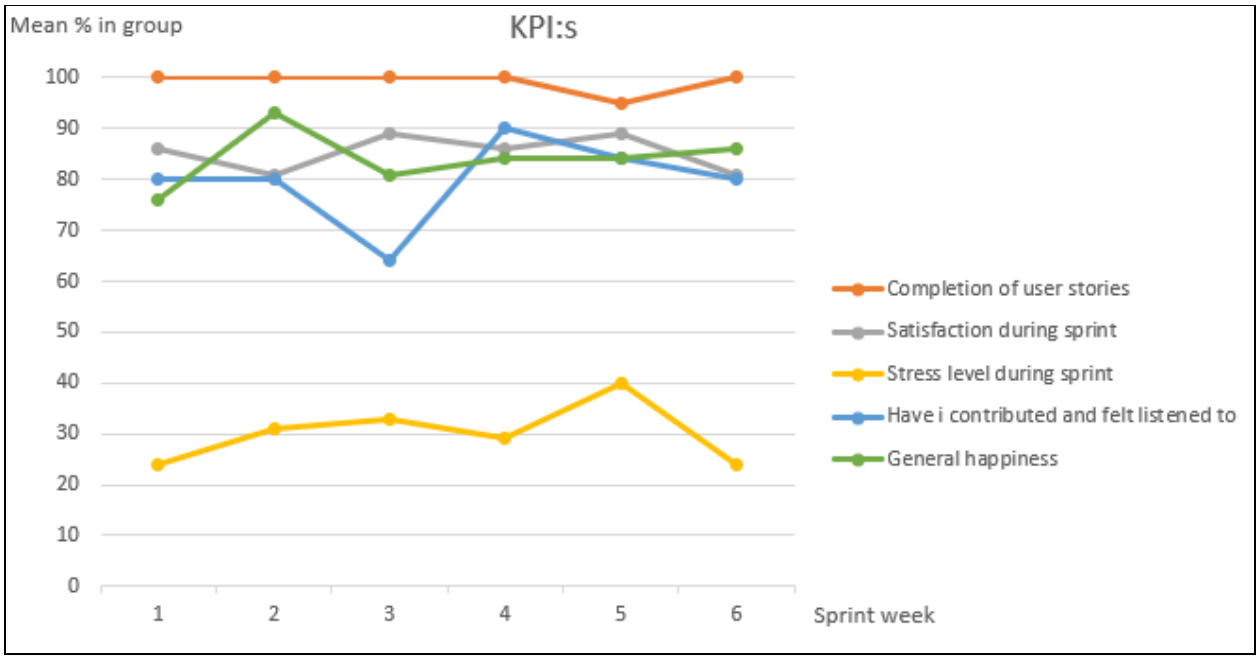

Figure 1: Overview of the KPIs for each sprint

A:
Each week we assed 5 different KPI, where each group member gave a score between 1 - 10, and then a pooled percentage was given. The KPI:s were quite personal and focused a lot on team work and team dynamic. The KPI:s were the following:

- Did we complete all our planned user stories in time?
- How satisfied are we with this week's work?
- What has my stress level been this week?
- Have I contributed and been listened to in this sprint?
- How happy have I felt in this sprint?

Although the gathering process was simple and quick, it gave a really good look of how the week had gone and to determine what needed to change during the next sprint. An example of this was when the contribution KPI dipped. Why? That week compared to previous weeks larger groups had been formed, that led to many feeling like they contributed less. To combat this, the following week smaller groups were formed, which in turn increased the contribution levels.

Overall the KPIs were good for this project, they have given the group value and changes have been made when a KPI has seen a shift in a bad direction.

B:
A KPI the group would like to add to the next project is one where the time is kept for each member of the group. Having this written down would create value for the other KPIs that were previously used. For example the stress KPI would have severely increased in importance if it was known how much time the group members spent.

The same KPIs were used all throughout the project, for future projects they should be changed as deemed necessary. This is because there may be different things that the group wants to know according to how the project goes or how the KPIs have looked previously.

Something a bit more aesthetic that would be nice to have is some graphs plotted over the KPIs which would give them more impact and ease understanding of how the project progresses.

A → B:
For improving the success of the project it would be crucial to schedule more time for reflection each sprint. Having more time to go over the KPIs and reflect on them would give more accurate results on the KPIs and would also force discussions over which KPIs that are necessary. From this it comes naturally that the KPIs that are deemed unnecessary are removed and new ones may be added.

Creating graphs for the KPIs would be easy if they were collected with some sort of tool, for example an online form, that could automatically compile the results into something viewable and easily understandable.

# Social Contract and Effort

**Your social contract, i.e., the rules that define how you work together as a team, how it influenced your work, and how it evolved during the project (this means, of course, you should create one in the first week and continuously update it when the need arrives)**

A:
The group created a social contract the first week. The contract covers a lot of topics and was the ground to create a common ground for all participants. No conflicts occurred during the project, no more than the conflicts during code merging. Therefore, the social contract worked well during the project and no changes were needed. Some very good decisions were made in the contract, for example the requirement to have cameras on for the formal meetings. The ability to see each other during the meetings greatly improved the communication and the mood within the group.

In the contract, the importance of keeping track of time was also talked about. The group had rather strict rules to get to the meeting on time and communicate in decent time if one is going to be late. This improved the effectiveness of the team and also helped with the spirit of the team. The social contract also covered every group member's ambition level. This helped the group to be on the same level and know what to expect from each other.

In conclusion, the social contract helped the team during the entire project. The communication was favored and also the group dynamic as a whole. The time spent to write the social contract in the beginning of the project paid off and saved the group tons of time later on.

B:
It is important to have a social contract and discuss this in a very early stage of the project. It facilitates that everyone knows the rules and goals.

A → B:
For future projects, take the time to sit down with everyone and set up a social contract, as was done for this project.

**The time you have spent on the course and how it relates to what you delivered.**

A:
The time spent on the course has been appropriate. This is something that can be seen on the KPI:s that are connected to stress, where consequently throughout the project the experienced stress level has been quite low.

The time spent has correlated well with what the team has delivered, where there has been steady increments to the final project throughout.

The group can’t see anything that would greatly change this. It’s always possible to do things a little better, keeping meetings more focused, or having clearer user stories, but the group thinks that it will come with time, and can not see an action that would greatly change this as of today.

B:
Keep this up for future projects.

A → B:
Keep having honest discussions about what should be prioritized. Also be present and active during meetings to maximise effectiveness, which was done well in this project.

# Design decisions and product structure

**How your design decisions (e.g., choice of APIs, architecture patterns, behaviour) support customer value.**

A:
Early on in the project the plan was to create a phone application but this was soon changed to a web app because this fitted the customers needs better. A design decision regarding data storage was made which resulted in using JSON files for the reflections that the stakeholder wanted to present. This was not the obvious choice but has worked very well. For the overall code design the application implements state management with an MVC-like pattern which has proven to be efficient. Also, the app has the ability to handle more than one type of quiz at a time as a consequence of the JSON usage for data storage.

Another feature that was requested from the stakeholders a bit late was the ability to export the data to the customers. It was not requested to send the data to the stakeholder, only the customer. However as it was requested late there was not enough time to implement it. Once again communication has to be worked on so that it is made sure that all the requested features reach the development team in time.

Some things that the group wanted to do was to integrate a so-called embed to collect email addresses. This was implemented, but not in a way that the stakeholders wanted so it had to be left out.

B:
For the next project the communication needs to be very clear between stakeholders and the project team so that all work creates value.

A → B:
As seen in A the primary problem with developing according to customer values was the communication between the development team and the stakeholders. So in the next project this has to be improved dramatically. Planning more before meetings with the stakeholders would be key as well as specifying someone to have more contact with the stakeholders, perhaps the product owner.

**Which technical documentation you use and why (e.g. use cases, interaction diagrams, class diagrams, domain models or component diagrams, text documents).**

A:

At the start of the project visual tools had to be utilized to give a clear picture of how the program flow would commence. To achieve this, Figma was used. As the group embarked on the programming journey and as the lines increased additional documentation had to be introduced. Forward facing public methods had to be documented and an overview at the top of each file should be added.

As the application grew in complexity and state management was established, more comprehensive documentation was developed. As the journey soon came to an end, the treasure had to be handed over to its rightful owners, but they needed help to navigate the tricky waters in the form of the application. Therefore, they would need a nautical chart in the shape of a README file.

B:
In future projects it would be preferred to start from the beginning with creating class diagrams to get a better overview of the entire application. Another suggestion would be to start the technical documentation earlier and continuously improve and add new information.

A → B:
The important thing is to create documents aimed towards creating class diagrams and technical documentation from the get go. All group members must be aware of their existence and why it's necessary to maintain them. Don't aim for perfection from the start, this is a live document that will be improved over time, the important thing is to get everything down.

**How you use and update your documentation throughout the sprints**

A:
At the start of the project the group was updating the Figma page according to how the application was thought to be. However, once the stakeholders agreed to a proposal it was not updated further. This despite the actual application having changed afterwards.

The text files that cover the major design decisions of the project were updated once; when the design decisions took place. Fortunately there has not been much deviation from the decisions that took place at that time but the document would still need an update and a rewrite in case someone outside the development team wants to read it.

The documentation in the code has been going smoothly between the sprints, the group members have updated the code as they write or go through it.

B:
In the next project the documentation should start immediately so it is possible to fill in when you make new decisions that may change the design of the application. Further on, the visual models should also be updated accordingly. Updating the visual models and displaying them for the stakeholders will lead to an increased understanding from their side and therefore the development team will be able to work better according to their needs. Design models should also be done continuously so they are easy to follow.

A→B:
To increase the documentation there would need to be additions to the DoD so that when a user story is done the documentation will be updated as well. Something that also would help is to have smaller groups make design decisions and then present them for the entire group. When they are thinking about these decisions they could at the same time start with the documentation to prepare for presenting their idea to the development team. This is because making decisions in the entire project group may make it so that not everyone is able to voice their opinion and ideas. It also takes away valuable time from the meetings.

**How you ensure code quality and enforce coding standards**

A:
In the current project there existed a definition of done (DoD) which covered partially what was necessary for good code. Later in the process there was also created a document for code reviews, which specified what was required from the code for it to be added to the main branch.

There have been continuous changes to the DoD according to how the development has been, the code review document was created late so it has been kept in its original form.

Unfortunately, as the code review idea came in late to the project, only a few code reviews took place.

B:
For the next project the amount of code reviews done should be increased so that all the user stories become reviewed according to the standards set. In the previous project these were informally done over the main communication channel but they should preferably be done in a way that makes them easy to refer to and see what has been reviewed.

A→B:
To reach the goal, there could be a role added whose task is to review the code before it gets merged into main, or it could be attached to each user story to review some other stories code. To make it easier to follow the code review there should be a list of what to look for within each user story.

# Application of Scrum

**The roles you have used within the team and their impact on your work.**

A:
Throughout the project the established roles have rotated within the group. These have been one Coordinator (or Scrum Master) who is mainly in charge of leading the meetings, one secretary who is in charge of keeping notes of any decisions made or topics discussed. In accordance with Scrum the group also had a Product Owner who demos the product to the stakeholder and is meant to take an extra look at the user stories. After about two weeks into the project the group received a tip from HAST that it can be useful to have a role that they like to call “BSSARE” who is in charge of keeping the meetings focused on the agenda (stop the bullshit) and to keep track of the time. The team found this very useful and decided to have two “BSSARE” roles for each sprint.

The introduction of BSSARE made a big difference to the effectiveness of the group meetings. With regards to the timeframe of this project and the size of the group, we believe that the group was close to a perfect execution of this project. However, as we have learned from HAST, meetings can always become more effective and there are some things with the product that we did not have time to implement.

B:
For future projects it would be helpful to become even better at keeping meetings more on point.

A → B:
One reason for straying away on irrelevant topics in meetings can be due to exhaustion and fatigue. Something we can do in the future to avoid this is to plan the meetings more precisely beforehand and include breaks so that every participant feels energised and can have an impact during the meetings.

It is essential to early on in a project define the roles clearly and along the way keep evaluating them and adding/taking away responsibilities. Decisions regarding this are not final, and can be updated as you go. Therefore it is important to come up with definitions of roles early, and then every week updating them. This leads to continuous improvement and an overall better project.

**The agile practices you have used and their impact on your work**

A:

A Scrum board was used in the project for the user stories. The agile practices have been enabled mostly through personal initiatives, for example when user stories have been poorly defined and unclear the people responsible to solve the user story have made an interpretation of what the customer wants/needs (sometimes with consultation from other group members and/or HAST) and taken action after that. Sometimes when people have had spare time they have gone ahead and solved some foreseen issues or corrected design flaws etc in order to be agile and make the MVP more valuable for HAST.

B:
Ideally the team should not have to depend on personal initiatives to complete the project and the user stories should be well defined from the start of each sprint to avoid misinterpretation.

A→B:
Keep practicing at the craftsmanship of writing user stories and becoming better at appreciating the required effort to complete a user story will get better with time and experience spent with scrum and the programming language. Another way to solve the problem of doing “extra” work is to have longer sprints with more intermediate meetings where the group can define more user stories that can be completed within the time frame without individual group members having to do this by themselves and guess what the next step might be.

**The sprint review and how it relates to your scope and customer value (Did you have a PO, if yes, who?, if no, how did you carry out the review? Did the review result in a re-prioritisation of user stories? How did the reviews relate to your DoD? Did the feedback change your way of working?)**

A:
Everyone has been PO for at least one sprint. The team had some difficulties actually nailing down the responsibilities for the PO, but after a few weeks the role became responsible for demonstrating the code at the weekly meeting with the stakeholder. The review however was mainly carried out by the Scrum master for the week. It consisted of a meeting every Friday where the team wrote the team reflection, merged everything into the main branch and also evaluated the KPI:s. The spring review led to a few user stories that the whole team agreed on both in regard to prioritization and difficulty.

The Definition of Done (DoD) evolved through the project, but it was not always carried out in the prescribed way. One way of making sure this happens to a larger extent could be to have checkboxes in the user stories on Trello for each DoD check.

It has been difficult to prioritize the user stories for two main reasons. Firstly it has sometimes been unclear what has been a priority from HAST because they may have been uncertain or due to poor communication. Secondly it has sometimes been difficult to understand what can be done within the given timeframe and with the team members limited experience with Flutter.

B:
The group  wants to become even better at prioritizing the user stories. To really understand customer value is difficult, and getting there is not only about asking your customer what they want, it is about seeing past that and identifying the actual value. The group also wants to be able to estimate what kind of user stories that can be solved within a given timeframe.

A→B:
Reducing the gap in this particular regard is challenging. One way of doing it is to increase communication with stakeholders, but it is also about helping the stakeholder really understand what they want. This could perhaps be done with a workshop or through customer surveys. Closer contact/relations with HAST could allow for more feedback regarding user story priority and thereby make it clear what to prioritize. More experience with Scrum and Flutter would most likely mean that the team becomes better at understanding how to limit project scope due to flutter capabilities and the timeframe.

**Best practices for learning and using new tools and technologies (IDEs, version control, scrum boards etc.; do not only describe which tools you used but focus on how you developed the expertise to use them)**

A:
The group has managed to gain new knowledge in a very satisfying way. The application has been created in flutter with dart, a coding language that has a lot of relevant documentation. The way that the team has worked in order to learn and spread knowledge in the group is through posting links in the relevant channels in Discord and then writing a summary if needed. Usually summaries haven't been needed since the documentation is really solid, but the times when it has been needed, the person who finds the source makes a comment like “you can find info about X on page Y, please read and we can implement it tomorrow”.

B:

This has worked extremely well and the group believes that there are no obvious improvements needed at this time and within this scope.

A→B:
As it has worked well in the current project the team will keep helping each other in similar ways as described in A.

**Relation to literature and guest lectures (how do your reflections relate to what others have to say?)**

A:
The group's experience is in large part reflected in the litterature and lectures. The challenges heard beforehand from lectures and literature were for example trying to make user stories independent, even in size for easy integration and really identifying customer value were things that were experienced. This is a bit tricky but also something that comes with time.

B:
The optimal state for the relation to literature and lectures would be that the team really can absorb what is being communicated to solve or completely avoid the difficulties of Scrum.

A→B:
The plan for getting from A to B would be to keep listening to individuals who are experienced in Scrum and take their advice. However the group also recognizes that learning Scrum takes time and there will be mistakes, so doing more Scrum is also an action to bring the group closer to point B.