CarCompare - Ahch-To

Adam Tafvelin, Christoffer Persson, Ebubekir Simsek and Tor Falkenberg Gunberg

Göteborgs Universitet & Chalmers Tekniska Högskola

2022-10-28

Customer Value and Scope

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

When choosing the scope of the project we kept in mind the resources, time and previous knowledge we had available. We felt that we were quite conservative when creating the MVP but when looking back this was invaluable in creating a common goal while leaving room for both development difficulties and additional features.

Features were prioritized in order of: MVP features, dependencies, stakeholder requested features and lastly additional features. Of course this structure was quite malleable since a deliverable of lower priority might fit into the sprint more easily than an MVP feature because of unimplemented dependencies or approximate effort requirements.

Features chosen for the project were above all targeted towards the customers using our product. Since we did not have a stakeholder during the first few weeks we envisioned that their primary priority was to keep customers engaged by having a useful product. When we managed to secure an external stakeholder our priority shifted somewhat and their concerns became our primary priority. Providing value for the stakeholder was still under the context of satisfying customers however if the development team and the stakeholder held different opinions on features the team would defer to the stakeholder or try to convince them to approve.(A)

Our way of working with scope and priorities for both features and who we create value for have been working well throughout the project. We feel that this is a feasible and good way to work in future projects and something that we should strive toward. (B)

This manner of working requires a good understanding and thoughtfulness when creating the scope, user stories and priorities. The INVEST criteria for user stories would act as a good benchmark to keep dependencies in check and the value of features in mind. The scope and MVP should be drafted to be as conservative as possible however this might prove to be a lot more difficult given the pressures of highly invested parties. (A->B)

- **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)**

Given what we felt was a conservative MVP we of course wanted to implement additional features to really flesh out the website however completion of the MVP was initially the project's primary concern. Development success during each sprint was measured in the satisfaction of the stakeholder and the completion of the user stories according to the DoD and acceptance criteria in the sprint backlog. Finding accurate approximations of each user story was the most important aspect in accomplishing this goal since team members would structure their schedule around these. Issues arose during the implementation of some features which made us fail in this respect during a couple of sprints, however given that issues will arise at some point during all projects we feel that the goal was largely successful.

Regarding teamwork our goal was for each member to contribute to discussions, product development and knowledge sharing. Our way of having planned multiple meetings with active discussions each sprint is probably the primary factor in us having been able to accomplish this goal. However we also feel that in a larger team this goal would be harder to accomplish and require more structure in order for no one member to passively take a backseat during discussions.(A)

The success criteria we used have been working well during the project to further our knowledge, vision and concretely evaluate our work and should be considered for use in successive projects. (B)

To be able to use these for coming projects we would have to share the goals pertaining to each criteria with the new team and get them on board to follow and strive to succeed in these regards. Considering the teamwork aspect, if there were to be a larger team in this new project a more structured meeting plan and code review session would be useful in order for points and opinions to avoid being lost while discussing freely. Free discussion has however been very beneficial to us and an alternative manner to conduct these would be to split the larger team and have discussions in smaller groups then coagulate their results into a larger whole. (A->B)

- **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**

We thought in the terms of the INVEST criteria for each user story created. This lent itself well to thoroughly analyzing each user story to gain as much value as possible regarding the feature. While we could not always avoid clashing with certain parts of the criteria, like dependencies, we were able to notice and bring attention to them and plan accordingly in order to decrease bottlenecks and increase value. After PO wrote user stories and their acceptance criteria they were discussed and finalized with the team in order to iron out future problems and increase the understanding of the context. When a user story was discussed and finalized it was given an initial effort approximation and once again when considered for the sprint backlog. This helped us limiting the amount of work we took on during each sprint to hopefully be able to finish with all items in the sprint backlog each sprint.(A)

We found this way of creating and using user stories to be helpful in clarifying the intricacies of tasks and solving issues before they arise and as such we felt this method is something we should keep in mind for future projects.(B)

Using and explaining the methods for creating, discussing and and working with user stories should be enough to replicate these results with future teams. (A->B)

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

Acceptance tests were performed during the sprint review where all team members and PO were in attendance. They consisted of a live demo of the acceptance criteria pertaining to the specified user story and a code review. To be accepted by the PO it was also necessary for the user story to be implemented into the main branch of the project without breaking anything. The value provided was that the team and stakeholder could ensure that previously agreed upon criteria and already implemented user stories functioned as a complete product sprint to sprint. This provided comfort to all parties that real progress was being made and that a product existed to be improved upon.(A)

We would like to expand upon the testing phase in coming projects. While accepted user stories never produced bugs during the course of the project a better practice would be to create actual test cases and document these experiments to have evidence of our confidence in the implemented work to show stakeholders instead of only an interactive demonstration.(B)

If we were to extend the sprints to more than one week we would probably have more time to create and perform more extensive tests for each user story. While this may impact perceived efficiency regarding the output of user stories, larger projects would require this sort of commitment to guarantee confidence and decrease risks associated with each deliverable.(A->B)

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

We have been using three different kinds of KPIs from the first week to the last week. Those are the Burnup chart, our weekly team reflection and finally the reflection we get from our stakeholders. Those were really useful as we could always analyze and reflect how our work process is going through different perspectives. If something went wrong, then we could always point it out before losing a lot of time trying to locate it. On the other hand, if everything was going well then we could in the same way reflect over it and still work in the same way as we have done in the upcoming sprints. Additionally, until week 4 we performed the team retrospective reflection orally, after week 4 we started writing it in order to have it visually documented(A).

Given that everything has been working well, we haven’t talked about restructuring our KPIs or adding something else for the next project(B).

As mentioned above, one thing that we could improve is the team retrospective reflections. We could start writing the team retrospective reflections from the beginning of the project instead of week 4 (A->B).

Social Contract and Effort

- **Your social contract, Links to an external site. (Links to an external site.), 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) There is a survey (Links to an external site.) You can use it for evaluating how the team is perceiving the process and if it is used by several teams it will also help you to assess if your team is following a general pattern or not.**

In this course, already from the beginning we made a reliable social contract that everyone agreed on. Where we included our meeting days and times, how we should communicate and how we should act towards each other in these meetings was only a couple points that were covered. For example, when a member does not have time to attend the meetings we created compromises beforehand to ensure that no bottlenecks would occur because of their absence. It took a little bit of time to get a social contract that everyone was satisfied with, but a thoughtful social contract had a really good impact on our work process and the result as well(A).

To be honest, regarding our social contract, we as a group do not want to do any changes for the next project because we believe that our social contract was a good one that covered important and good points(B).

We don’t want to change anything on how we worked regarding the social contract for another project, we would make it in the same way. However still leave room for additional points to be added as necessary (A -> B).

- **The time you have spent on the course and how it relates to what you delivered (so keep track of your hours so you can describe the current situation)**

From the start we wanted to be done one week before the course planned us to be done so we had more time to study for our other exams. This in conjunction with our belief that the product was to be handed in a week prior to the actual deadline made us plan to end development a sprint too soon. This did affect the amount of weeks we spent on the project, but we do not believe it affected the final product or much on the amount of time we spent on the project as we worked harder each sprint. We structured each sprint by choosing prioritized user stories for each developer. In the case that we wanted to work together on a user story, we made sure that the pair would get more work in total that sprint so it would be fair. (A)

An ideal situation for the next project would be that we were able to make use of all the sprints and allocate the last sprint to only debugging and testing. The dynamic of who did what was well executed in the current project however, so we believe it would be great to do something similar for the next one.(B)

To achieve this, we would start the project and plan within a time frame so we would not be affected by the exams that are inevitable so we can make use of the last sprint properly. We believe that keeping the same dynamic regarding the allocation of work would also be of great benefit.(A->B)

Design decisions and product structure

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

We had decided from the start to make use of the ASP.NET MVC to have a proper structure of our project. Though this does not directly affect the value for the customer, it is much easier to maintain. The API we planned to use turned out to be a dataset, this did lead to the fact that we cannot update our data without paying for a new set of data. For a real product, this could be managed by updating the data once every year or so to keep the costs down for the customer. For the choices regarding what the customer wants on the actual product, we placed an MVP to develop against, and we added additional features if we had time. During the sprints we also expanded the amount of user stories if we saw that we were able to, this to increase the value to the customer.(A)

We believe that our choices were effective and for the next project, we would do something similar to what we did here. However, regarding the dataset, we would make sure that we have a proper API so we can update our data properly. We would also make use of an external product owner to make sure that we don't have any bias towards our product that would possibly deviate from the proper customer values.(B)

Doing more research regarding the availability of what we are able to use would be of great benefit. As stated, using an external product owner would also be better. All in all though, we believe that we had a solid structure and design decisions, and not much would be changed.(A->B)

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

At the start of the project we agreed that all team members would comment their written code each week in a reasonably detailed and clear manner. This tactic was discussed with our TA and teacher who said that this would be sufficient. In combination with the open discussion and explanation about the weekly results every Friday meeting, this led to a good overall understanding for the whole team. Since we were only four people in our group, this was quite an easy and rewarding process. One of the greatest advantages has been the timesaving aspect. Creating a larger set of documentation within this group would have cost both time and energy, and probably would not have given much value in return(A).

If future projects were done in a similarly small group where every team member has roughly the same prior knowledge, we would like to use a similar approach. However, if the team was larger, or consisted of people from different areas of expertise, a more robust solution might be necessary(B).

A larger explaining document, putting the different classes in relation to each other, combined with class diagrams clearly explaining the expected functionality for each part of the application, would be a good way of doing this(A->B).

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

As a group, we had an agreement about making documentation in the code we have committed each week. That means that everyone had a responsibility to comment on their own work before committing it to github. This has helped us a lot during our work process thus everyone in a way was active documenting and describing. Basically, that made it easy to follow each other's process as well(A).

For the next project, we could implement a sort of documentation outside of the code to get an easier understanding of each other’s part of code. Especially if there are more members in our group and the project is bigger. Of course, as a developer it would take more time to document in this way. However we believe that this would have a good impact on our work, otherwise it could be hard to understand by just seeing plain code without any description(B).

To achieve this for our next project, we could use a tool named Document360 or just a google document in order to make our documentation available for all the group members and update during development time(A->B).

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

As stated in a previous point, we made use of ASP.NET MVC for the structure of our project in regards to the code. This allows us to be professional in the quality of our development and the fact that everything is written in the right file etc. In regards to the quality of our code, we had a checkup meeting each Wednesday where we demonstrated what we had done and if we had any questions. This along with the demonstration of the finished user story at the end of the week where the product owner would accept or decline what we had delivered, with additional input from the other developers if something seemed off.(A)

For the next project we believe that the somewhat heavy, but very useful MVC pattern would be beneficial and if a project is made regarding anything graphical, this would be used again. Apart from that, since everything else had worked very well together, we will most likely do something similar if not the exact same next time.(B)

This level of efficiency would be possible only if we were in the same group again. And it was quite lucky that we were able to have three meetings each week as making a common plan in schedules is very difficult. We would try to make sure we are in the same team again.(A->B)

Application of Scrum

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

Within the first week, we decided to vary all kinds of roles so everyone in the group had an opportunity to feel how to act in such a role. We used a tool named “Spin the wheel” and allocated everyone's role for 8 weeks ahead in order to be as equal as possible. Of course, we still believe that this method affected us in a positive way as a group and our weekly deliveries as well. On the other hand, we did not rotate out the internal project owner, this was due to the fact that we believed that it was more sustainable to have the same project owner during the work process with an already established connection to our external stakeholder. We managed it very well and it did not create some biased situations within the group as well due to the fact that we always had good communication within the team(A).

For the next project we believe that we would work in the same way as we have done during this process. One thing that could be more useful for us, is that we could use an external project owner instead of an active developer one. That would give us an opportunity to have an extra eye outside the group. On the other hand, that could complicate some parts as it would be more difficult to communicate with the project owner as well as managing meetings(B).

In order to find an external project owner for the next project, we as a group could always ask and contact close friends and family members. Or perhaps, do an actual project where we try to find a real client that is interested in our project (A->B).

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

Although working towards a deadline is not new for any of us, the concept of shorter deadlines with smaller tasks, i.e. sprints and user stories, has been very useful and generally made the task seem less daunting and relieved stress.

We decided to have three meetings each week. Monday naturally being the startup meeting, making sure everyone knows what they should do in the coming sprint. Wednesday meetings have usually been a bit shorter and have been used to communicate about "how we are doing" as well as helping each other in case of unforeseen problems. And of course, Friday meetings are used to finish up the sprint, merging our efforts, and evaluating the sprint. This has felt natural and comforting during the process.

The KPIs we used are discussed above, and proved to be great tools. Feedback from the stakeholder has been somewhat useful, although often mostly reflecting bugs and changes we already had thoughts about changing/fixing. Both team reflections and individual reflections have been tedious at times, but being 'forced' to reflect over the past sprint has brought some real insights as well(A).

We are very happy about the KPI's and our meeting structure. For future projects the only real change we can think of is to put more time into creating user stories and transforming them to more precise tasks, and maybe also to have an actual external PO(B).

Some more discussion and planning for the user stories and tasks at the start of the project and at the start of each sprint, could make them even better aids during the actual working process. Also assigning an external person as PO (and instructing them to be strict) might be worth a real try(A->B).

- **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?)**

We chose one of our team members as an 'acting PO' at the start of the project. The sprint review is a collaboration with the team members and the PO. Since the PO is also developing user stories their approval of that one may be subject to bias, though all of us helped out to prevent this from becoming a problem. The PO and the team is shown what has been made during the week from each team member and if this is deemed to satisfy the acceptance criteria, definition of done and passes the peer review of the code it will be accepted into a done state to be shown to the stakeholder. Some weeks the addition of one completed user story made it clear that another one had to be added to the product backlog to have the product fulfill MVP. At other times it became obvious what user stories we had to prioritize for the next sprint, to have the development continue in a logical and smooth manner. An example of this would be when the basic list functionality was finalized. We then realized that we had to prioritize at least part of the design of the list, to be able to expand and test the further expansion of filtering and information handling(A).

As a whole this process has worked very well. As mentioned already in this report, it would probably be a good idea to have an external PO next time around(B) & (A->B).

- **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)**

For the IDE, we made use of the heavyweight Visual Studio 2019, this was due to the fact that it supports the ASP.NET MVC structure, which greatly benefited us when making the structure of the project. One member of the team was already familiar with this IDE and set everything up and provided an instruction manual for installation. We all learnt by using it and if we were unsure about something, we asked other team members, or Google. For the scrum board, we used Trello, this was very intuitive and helpful when creating user stories, like it was created for that very purpose. Git and github were used for the version control. It was a little difficult at first to merge as we had not done the structure for merging properly, but we learned a way to manually merge, which did take more time, but it did get the job done.(A)

For the next project we may still use Visual Studio 2019 or any other version of it as it did greatly benefit us. However, the use of IDE is not the most important thing. We also wish to have better structure with Git and github to ease the merging process.(B)

We will make sure that we continue to keep up communication with the other members during projects and if any questions arise, someone may be able to help. In order to maintain the structure, we may need to do some research on how to properly set everything up regarding Git. (A->B)

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

One of the largest issues that was presented in the guest lecture was the halt of continued support by the ones holding the open source engine they were building their project upon.

This reflects upon the importance of knowing your key partners and in some ways minimizing their influence over your project. You do not wish for a huge project being reliant on a single point of failure. What we have observed as a point of failure for us is the server which the website is running on. Much of the code used in the product uses the server to compute and then return data to the client. While we never thought this was an issue before we have seen a slight decrease in performance as the product grew bigger. After consideration we found that this was strange since our website is quite small and so we found that the slow nature of the website was due to us never having thought of the difference between debug and release modes of the website. Switching to release mode with optimizations enabled our website was faster than when initially started developing and our problems were solved. This was however an issue that we could not have foreseen without knowledge or research of website development before beginning development and shows, as inferred from the guest lecture, how important either underlying knowledge or a research phase would have been before entering the development phase.

The information in the lectures and workshop has shown us in what ways Agile and Scrum can lead to a better working experience and result(A).

In our project the largest issue that has come from the development is based on the data provided to us by our database provider. This issue could probably be minimized by researching exactly the type of data that would be provided and which techniques could be used to implement it into the project beforehand(B).

In the future when relying on partners or data we should keep in mind to research them beforehand(A->B).