

Reflection week 37 HALO

Smith states that reflection is "assessment of what is in relation to what might or should be and includes feedback designed to reduce the gap" (R. Smith, *Formative Evaluation and the Scholarship of Teaching and Learning*, New Directions for Teaching and Learning, vol. 88, 2001, pp. 51-62) which can be boiled down to describing ...

... the current situation or "what is" (A),

Progress

Fast progress on basic functionality of the website (html, data & databox, search) -> leftover time not used because of inability to hold meetings (schedule conflicts),

Scrum effectiveness

- Low frequency daily update messages
 - Short notice meetings
 - Pair programming - Very scrum, very nice
 - Success! MVP Achieved!
-

... what you want the situation to be or "what might or should be" (B), and

- We should be able to work regardless if we finished all the user stories we set out to complete in the sprint or not. There should always be more user stories to pick from.
 - Daily recap messages should increase in frequency and size
 - 24h meeting notice minimum.
-

... a plan for getting from where you are to where you want to be or "feedback designed to reduce the gap" (A -> B).

- Write more user stories in preparation for the sprint as well as during the sprint if any unexpected elements arise requiring a change in direction (though a large change in direction would constitute a new meeting).
 - Better communication (frequency mostly) in discord.
-

Design decisions and product structure

- how your design decisions (e.g., choice of APIs, architecture patterns, behaviour) support customer value
 - React is easy to use and therefore allows fast time to market which is valuable to the customer. Furthermore, its imposed project structure allows good code visibility (good code allows sustained delivery of new features without project slowdown). Moreover, it allows easy styling with CSS which is valuable to the customer as without it the website would feel ugly to use.
- which technical documentation you use and why (e.g. use cases, interaction diagrams, class diagrams, domain models or component diagrams, text documents)
 - We started sketching a map over what we thought the program should look like, but realized rather quickly that we did not have adequate knowledge of how the programming language worked, so we never finished it. A project map should be revised in a future meeting which will be further discussed in the sprint review meeting.
 - Readme.file for React
 - Readme.file on github, however, this is not currently used.
 - Comments throughout the code explain relevant functions.
- how you use and update your documentation throughout the sprint
 - We made accompanying comments as we wrote the code.
- show you ensure code quality and enforce coding standards
 - We ran the code and made sure it worked as intended
 - We separated code into different React components to maintain a good code and project structure.

Application of Scrum

- the roles you have used within the team and their impact on your work
 - No role specific activity (due mostly to trivial code, secondly because of the lack of need).
- the agile practices you have used and their impact on your work
 - Daily recap messages in the discord channel, though should increase in frequency.
 - Pair programming which worked very effectively at getting everybody up to speed and making progress towards the MVP.

- At demand meetings, though, they should be better notified (longer notice).
- Working with the user stories we have written.
- 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 will have our first sprint review on sunday.
- 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)
 - We have previous experience with git/github which we use for version control.
 - Programming in pairs is another way we used to force a learning together experience which resulted in great results in progress.
 - React is used throughout the project and we developed our expertise on it though a self imposed mandatory tutorial binge before the first sprint. Furthermore we share knowledge continually with each other through for instance pair programming.
 - We continuously learn more about the tools as we are working with the user stories.
- relation to literature and guest lectures (how do your reflections relate to what others have to say?)
 - We have had no guest lectures and haven't used any literature. However, we love youtube tutorials which we use extensively.