

Group project: general feedback

- Dr. Luca Rossi -

Well done to all the groups who have successfully completed the first iteration and are now working on the 2nd one, despite the difficult circumstances you had to work in. This document is meant to provide some general feedback following the first few weeks spent on this project, particularly the story writing workshop of week one. I will release another document outlining general suggestions and recommendations toward the end of this month, sometimes between the end of iteration 2 and 3.

1. Focus on essential features, do not overdesign

Looking through the feedback the TAs gave you in these weeks as well as considering some of the questions raised through the message board, it seems a few of you are still struggling to understand the importance to focus on essential features and avoiding overdesigning the system, particularly during the early software development stages.

Recall that a good software product should first meet the requirements. The high-level requirements are given in the handout. It's up to you to distil these high-level requirements into more detailed stories. An important part in this regard is to pinpoint what you think are the *essential* features of the system, i.e., those features that are necessary for the system to work. Sometimes you may find some of the requirements descriptions in the handout somewhat vague, or at least subject to interpretation. As you've learned, this is perfectly normal and expected in a real-life situation, where requirements need to be analysed and formalised before moving on to the design phase. For example: the system is not required to be multilingual, so in case you wanted to add this feature you need to take into account the fact that this is largely non-essential and very low priority, so it should not be the concern of the first iterations.

When deciding how to interpret these requirements, remember to use Occam's razor¹: the best solution is often the simplest one. In other words, keep things simple and do not

¹ https://en.wikipedia.org/wiki/Occam%27s_razor

overdesign the system. Overdesigning may lead to adding functions that are not necessary, not suitable to the scenario and/or will greatly increase cost on the project. This also has the side effect of distracting you from pursuing the implementation of the essential features. For example: it makes no sense to distinguish between ‘very hungry’ and ‘normal’ users in the system. While the former type of customer is more likely to order add-ons, this does not correspond to any role in the system, where customers should only be distinguished between registered or not registered. In general, use the MoSCoW rule to decide the priority of the features and research existing solutions or ask potential customers to support and justify your decisions.

Finally, while your system needs to communicate to other *external* system, e.g., payment system and email system, you should only focus on your system and not worry about how the external systems work. For example: your implementation of the system doesn’t need to send an actual email to the manager with the weekly reports.

2. User stories priorities, acceptance criteria and iterations

User stories should be assigned priorities so that higher priority stories can be implemented in the early iterations. In addition, each story needs to be given relevant acceptance criteria that can be used to support the understanding of the story. Regarding the granularity of the stories, the rule of thumb is that stories should be small enough to be achievable within an iteration. Finally, please adhere to the standard templated when producing the product backlog: it’s important you fill all the necessary field, as learned in the lectures.

4. Agile teamwork in times of pandemic

Agile teamwork is based on continuous communication, with team members working together and having a common vision of the project and its progress. The current circumstances make it particularly complex to keep a healthy level of communication, however this is easily circumvented by using technologies including (but not limited to) emails, chats, videoconferencing, shared code repositories and documents. You should do your best

to make sure no member of the group is left out in the communication process: failure to do so may hinder the success of the whole project.

5. Other considerations

Other general considerations include: 1) make sure the language of the UI is English, as this is the language used in the country of the client who commissioned the system; 2) remember to make timely updates on the Hub, ideally (but not necessarily) trying to stick to the proposed timeline – this is to help the TAs and the lecturers to give you more prompt feedback, as well as to make sure you stay right on track with the project without having to rush during the last few days; 3) TAs have been assigned to each group to monitor and assist you: make sure you interact with your TA; 4) follow as closely as possible the Hub guidelines when uploading files, journal entries, and other evidences of your group/individual contributions.