Graph Repo Visualizations

The assignment consists of 9 sprints: one initial sprint in which you gather the requirements, one managed completely by SIG and 7 fully managed by students. We will use Github for all our communication and interactions. You are requested to create a Github project and invite us with the address *a.serban@cs.ru.nl* and handle *lbergmans*.

Assignment Details

You will choose a role between Software Manager or Software Architect and design a system that allows to visualize and extract useful information from software repositories. You are requested to come up with innovative ways to present and filter the data. SIG will provide support whenever you get stuck, but will not provide hard requirements except for the sprint we manage. We will dedicate a full sprint for gathering these requirements and selecting the best. Afterwards, we start developing in an Agile fashion with one sprint/week. If time allows, we dedicate one final sprint to refactoring and polishing the project.

Methodology

Github will be our main workhorse. Although you are not yet familiar with git, it is an essential tool both for the course and for software engineering. SIG will provide study materials so you can get up to date in no time. There are two ways of looking at Github: (1) from a developer's perspective, who commits code and solves bugs or issues and (2) from a managerial perspective, by tracking the progress and providing a higher level interface to the organizational part of the project.

For the first part we will use Gitflow [1], which is a methodology for collaboration with git that supports large scale teams. Briefly, we will have a main branch that contains all the changes that are reviewed and approved. Whenever we make a new change or implement a new feature, we open a *feature* branch for our changes. Once we complete the work, we create a *pull request* to the main branch to integrate the code. On a pull request, we will configure BetterCodeHub to measure the quality of our change and we will also do manual code reviews. After the code is reviewed, it is approved and will be merged in the main branch.

For the second part, we will use Github Project Boards [2], which is a way of managing tasks and tracking progress in a sprint. For each sprint we will configure a Kanban board where we select and split the tasks we will do in the sprint. Once a task is implemented, we can move it from *todo* to *in review* or *completed.* Please browse this repository of materials about kanban [3]. Although it seems there is a lot to learn, everything is very simple and easy to use in Github.

Final thoughts

It seems the project is self-managed: you are requested to organize and deliver the project with little intervention from outside. However, SIG will offer support and coaching whenever is needed. Feel free to contact us for any questions.

Since time is short, we advise to start with the current technology stack and focus on developing and delivering some features. It will allow us to focus more on software engineering aspects and less on selecting or researching new technologies. However, if you would like to learn a particular technology and use it in this project, we are not against it. Just remember we will have to deliver some functional features until the end of the semester ☺.

Please browse through the references and let us know your thoughts.

References

[1] https://datasift.github.io/gitflow/IntroducingGitFlow.html
[2] https://help.github.com/en/articles/about-project-boards
[3] https://www.atlassian.com/agile/kanban