Initial Report of Group CONT8

How do we build an ideal CI/CD pipeline deployment solution for software at scale?

a1774020 Chao-Chung Chuang

a1791407 Kwok Chung Kwan

a1806522 Zhuofan Zhang

a1777315 Jinchen Song

a1778478 Runtao Zhuge

a1778829 Zishun Chen

a1839458 Benedict Xu

a1782077 Yunfan Han

a1788058 Yifei Xu

Project Vision:

The vision for this project is to ideate and simulate software deployment techniques used by commercial software at scale. Modern approaches to software development have led to the use of Continuous Integration/Continuous deployment as common practice to bring new and updated software features to market while minimizing costs and risks that come with making changes to the product. Some of the benefits of this include earlier fault detection and resolution, due to the smaller code blocks being deployed constantly, better test reliability, being able to test updated features in isolation, and overall earlier customer feedback, as users discover and interact with newly added features.

Our goal is to create and document a full software development process through the use of CI/CD techniques whilst weighing up the costs and benefits of selecting particular pipeline automation tools and technologies.

Customer Q&A:

During the kickoff meeting, we asked questions to the client regarding the requirements and specifications of the project such as: what tools are allowed for implementation, what were the expectations for the website API, and are there specific tasks that need to be satisfied in the first sprint. In addition to that, we asked questions regarding the user stories, for instance, what is the format of the product and sprint backlog, and how to move a backlog item from one section to another section.

The Kick-off meeting went smoothly in general. It covered important information regarding the assignment such as: what the primary specifications for the sprints were, how we create a user story using GitHub, and what information needs to be provided in the sprint and product backlog. However, before the meeting began, our team did not do much research and preparation in creating user stories by using GitHub. We did not inquire into the details about the user stories because our members were not familiar with them.

In the upcoming meeting, our team will do some preliminary research to further understand the topic. Also, we intend to show deliverables to the client to get advice for improvement.
Here is a list of questions that we would like to ask in the next appointment.

- If we identify a task from DONE that does not satisfy the requirements in the final, should we redo the whole process or move it back to the previous stage for enhancement?
- Is there a time limit for each task in the To-Do/ IN PROGRESS section?

Users:

Developer:

For the developers, this CI/CD project can help the developers constantly review and edit the previous code, detect many small errors, and in addition, can automatically compile the software to build it. It can help the developers to build an automatic server and link the source code on a trigger or timing basis.

Tester:

The CI/CD can allow testers not only to focus on the code and focus on more pressing issues but also to test more scenarios at the same time. It will increase delivery speed, reduce errors, and automatically make the testers do the code reviews, unit tests, integration tests, and system tests.

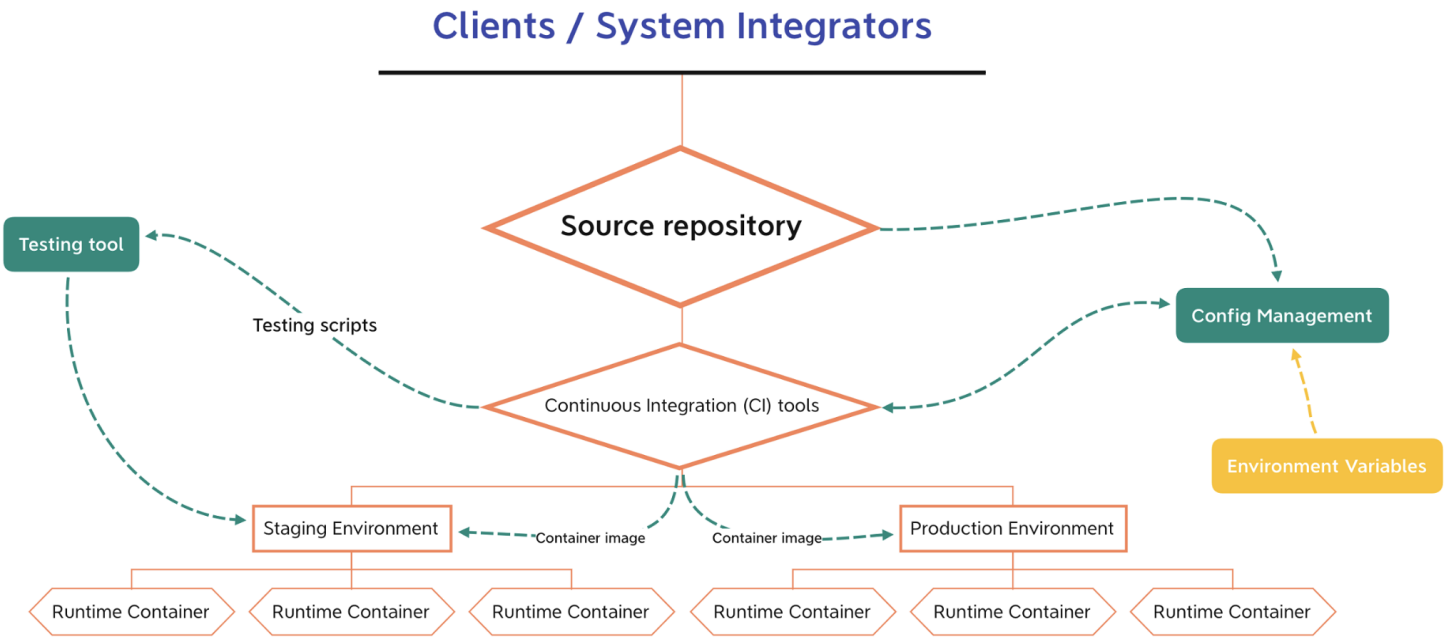
Maintainer:

The tool can help the maintainers automatically publish applications to the production environment, which help the maintainers easier to continuously receive and integrate the user feedback. It allows the output process of software products to be completed in a short period to ensure that the software can be stably and continuously maintained in a state that can be released at any time

Project owner:

The tool can help the project owner contract the product’s process, then give some advice to the developers.

Software Architecture:



The developer modifies code to fulfil the requirements. In many cases, the developer will keep the source code with a timestamp during the development process for backtracking purposes.

The modification in source repositories will trigger the process of the CI server and meanwhile, the server will build a developed artefact. All artefacts will deploy into a repository, for example, the nexus.

The test script is generated by the CI server and the test client will make a test run. Test client makes the test against the environment which contains the most recent artefacts. When the test passes, the CI server will build the rocker image with the current environment and configure the management tools.

Tech Stack and Standards:

Front-end:

- HTML5/CSS/JavaScript
- Vue.js
- SASS/Bootstrap 5/jQuery

Vue.js is implemented as the main framework; SASS, Bootstrap 5, and jQuery are used as supplementary libraries. Html5/CSS/JS would be the main front-end development tool.

Back-end:

- Java
- Spring Boot
- MySQL

The database and APIs are manipulated with Spring Boot, and the programming language is Java SE. MySQL would be the main database for this project.

Communication:

- WeChat
- Slack
- Github

WeChat, Slack, and email would be the main tools for communication, and Github would be the main storage and version control tool.

Coding standards:

- Comment
- JDK version

All the codes should have sufficient comments so that developers can understand what each of the group members has done, JDK coding should be consistent, and the Jenkins version used should be the same

Alternative:

- React.js/Angular.js
- MongoDB/Node.js

React.js and Angular.js would be used for the front-end development if Vue.js cannot fulfil the requirements. MongoDB and Nodejs would be the potential backups for the back-end side.

Group Meetings and Team Member Roles:

Meeting frequency:
The group meeting will be held at least once per week and will take no longer than 1 hour for each meeting. The regular group meeting will be scheduled every Wednesday at 3:00 pm, and the other meeting will be scheduled according to the course schedule this week.

Sprint retrospective meeting:

- **zoom**
- **WeChat voice**

Our next sprint retrospective meeting will be on Friday 20th, 2021 at 3 pm. Due to the impact of the pandemic, all our meetings will be held online via zoom.

Additional feedback channels:

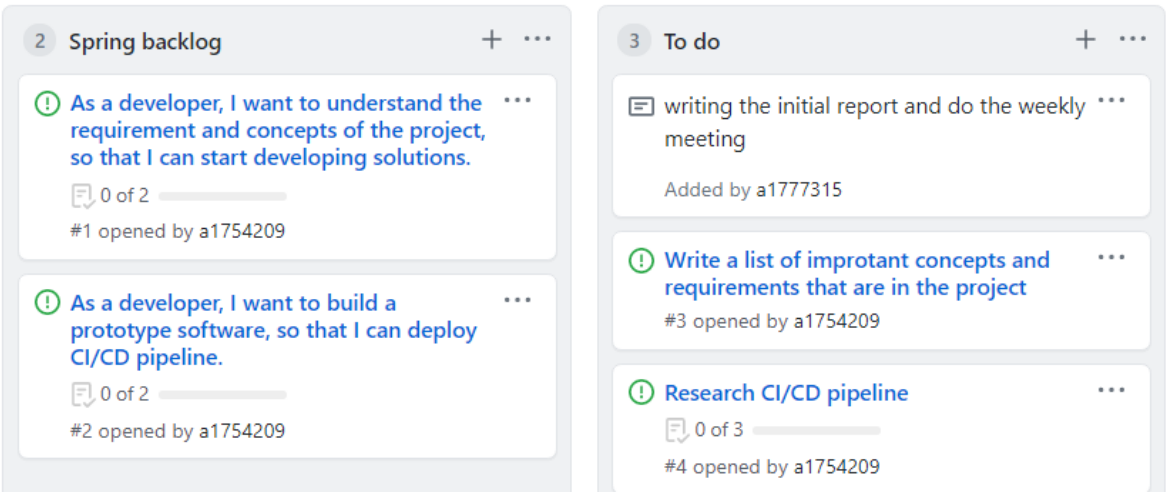
- **Slack**
- **Email**

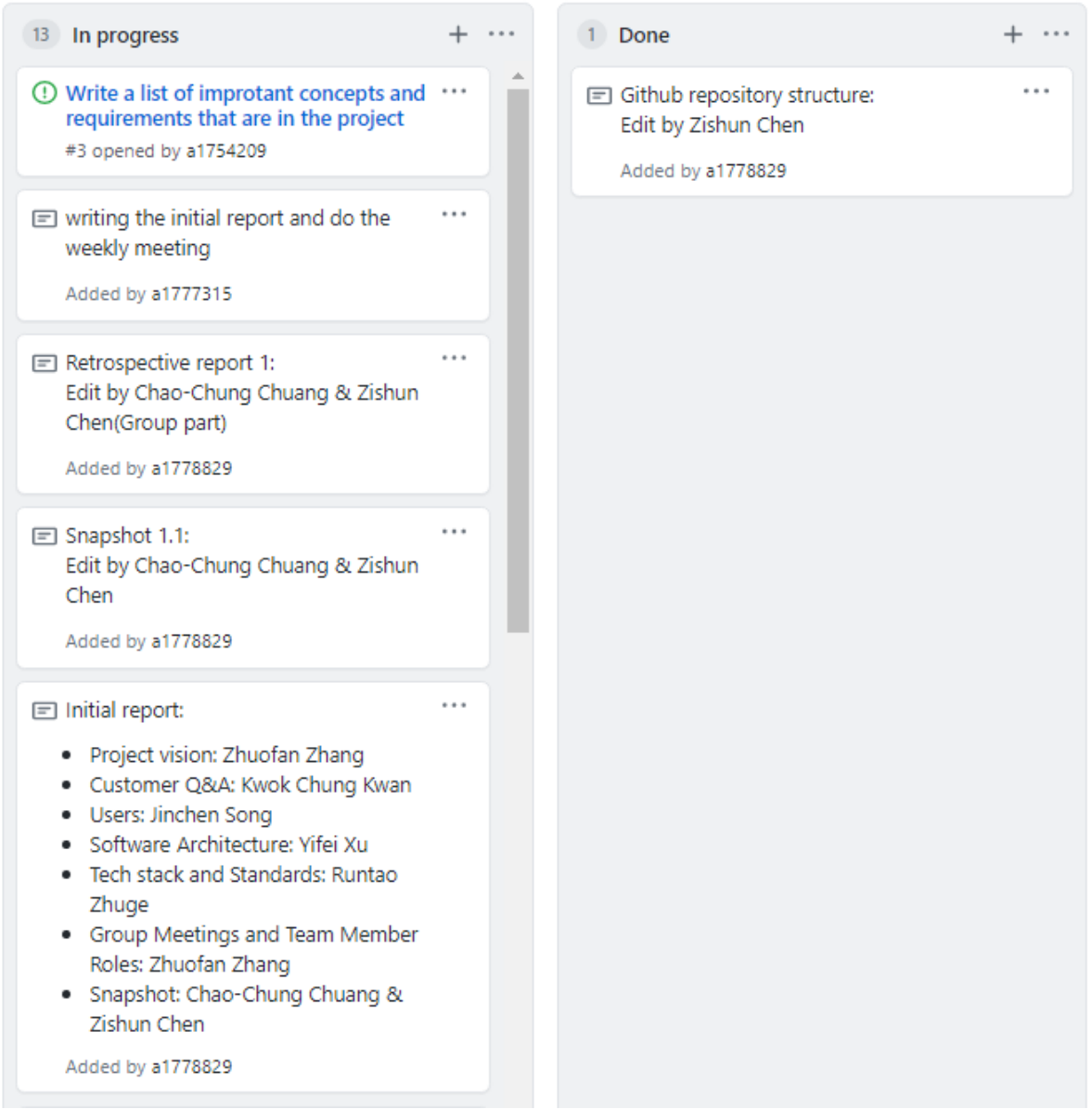
We have established additional feedback channels with customers in Slack and Email so that we can improve and optimize our products.

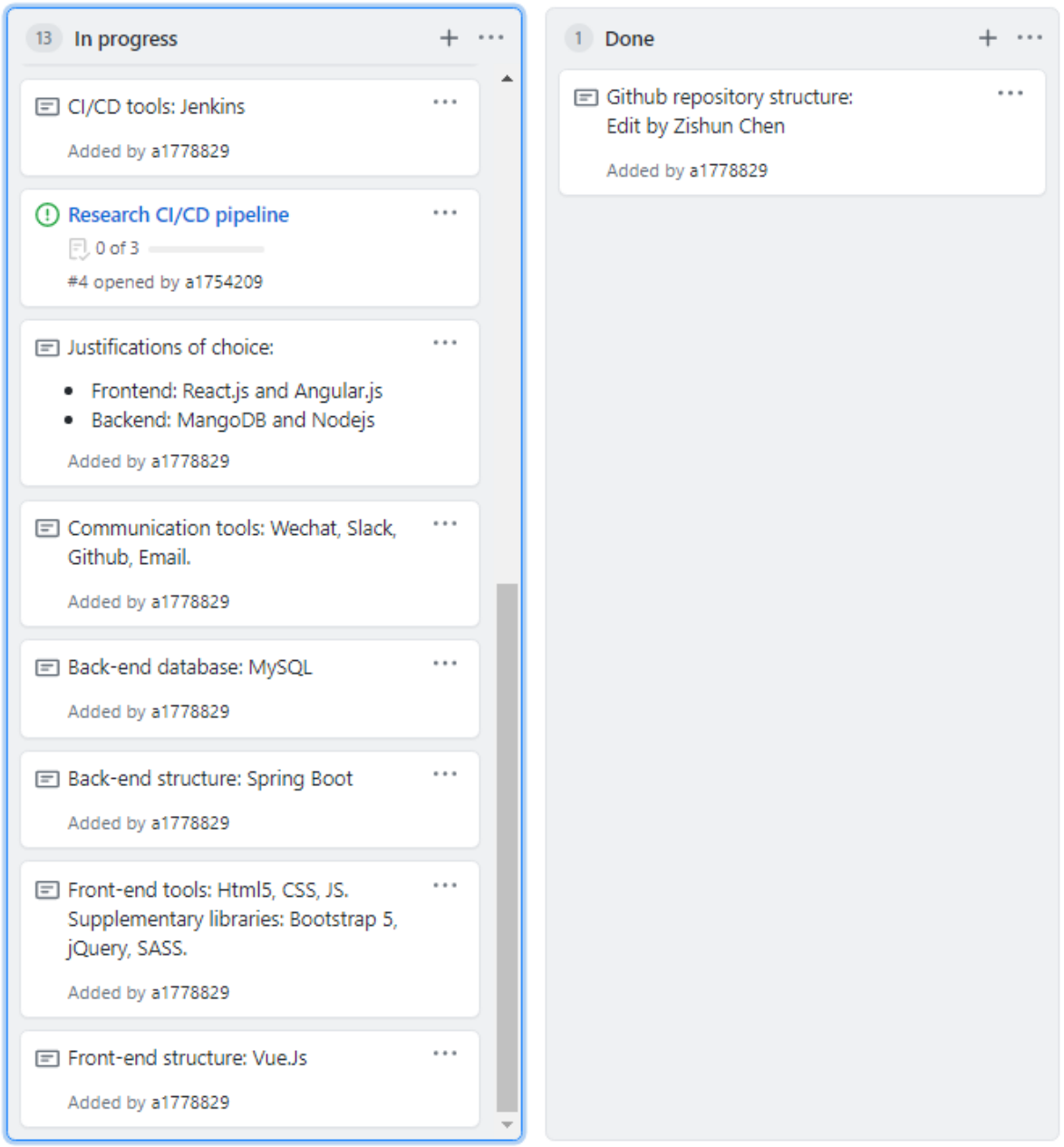
Scrum Master:

- Spring 1: Zishun Chen/Chao-Chung Chuang
- Spring 2: Kwok Chung Kwan/Jinchen Song
- Spring 3: Benedict Xu
- Spring 4: Zhuofan Zhang/Yunfan Han
- Spring 5: Yifei Xu/Runtao Zhuge

Snapshot :




13 In progress
Write a list of improtant concepts and requirements that are in the project
#3 opened by a1754209
writing the initial report and do the weekly meeting
Added by a1777315
Retrospective report 1:
Edit by Chao-Chung Chuang & Zishun Chen(Group part)
Added by a1778829
Snapshot 1.1:
Edit by Chao-Chung Chuang & Zishun Chen
Added by a1778829
Initial report:
Project vision: Zhuofan Zhang
Customer Q&A: Kwok Chung Kwan
Users: Jinchen Song
Software Architecture: Yifei Xu
Tech stack and Standards: Runtao Zhuge
Group Meetings and Team Member Roles: Zhuofan Zhang
Snapshot: Chao-Chung Chuang & Zishun Chen
Added by a1778829
1 Done
Github repository structure:
Edit by Zishun Chen
Added by a1778829


13 In progress
CI/CD tools: Jenkins
Added by a1778829
Research CI/CD pipeline
0 of 3
#4 opened by a1754209
Justifications of choice:
Frontend: React.js and Angular.js
Backend: MangoDB and Nodejs
Added by a1778829
Communication tools: Wechat, Slack, Github, Email.
Added by a1778829
Back-end database: MySQL
Added by a1778829
Back-end structure: Spring Boot
Added by a1778829
Front-end tools: Html5, CSS, JS. Supplementary libraries: Bootstrap 5, jQuery, SASS.
Added by a1778829
Front-end structure: Vue.Js
Added by a1778829
1 Done
Github repository structure: Edit by Zishun Chen
Added by a1778829