

## Bugs

Given where we are at with development, I don't believe we've really found any bugs by this point, at least in the service layer. So, I'm not sure if I can contribute much to this section at the moment.

Even if there were bugs, at the time of writing, the components of the backend are still being written, and we can't begin testing until it is finished.

## Dev Ops

<https://www.atlassian.com/devops>

<https://about.gitlab.com/topics/devops/>

<https://aws.amazon.com/devops/what-is-devops/>

<https://www.redhat.com/en/topics/devops>

## Stages

Dev Ops utilizes an endless loop of stages, each flowing into each other over the course of development. Here they are in order:

### *Discover*

Simply find ideas and have them ready to present.

### *Plan*

With the ideas gathered, plan out the software that you want to build.

### *Build*

Utilize software such as Git to collaborate and build out what was planned.

### *Test*

After software is built, load into a test environment, such as Docker container, so it can be seen how well it works.

### *Deploy*

Now that the software is determined to work, put it out through an Amazon Web Server, or another type of computer, to serve your customers.

### *Operate*

Handle day-to-day dealings, utilizing the software provided by the hardware of choice.

### *Observe*

Identify issues customers may have with the software and notify the team of potential changes.

### *Continuous Feedback*

Make reports and determine what should be done to improve later releases.

Aside from the build, test, and deploy phases, everything can use typical meeting tools like Teams or Zoom to collaborate. For the build phase, a tool such as git—a version control platform—would be the best choice, as it allows you to upload code for others to view and edit, as well as approve and merge changes in a way that will be sustainable for continual functionality.

## **Meeting Minutes**

Last week was slow for meetings, but this second week had a fair bit happen, with us determining who was responsible for what section, determining the models we will be using, and meeting regularly for assistance actually coding all the components.

# Reflection

**Overall Status:** On Schedule

**Summary of Project Tasking:**

I attended all meetings with team mates and the class. I worked on the Service layer of the Project and Task sections of the code, which is the actual logic of the actual transferral of information.

<https://github.com/byui-cse397/AcornNursery/issues/51>

**Number of Hours:**

At least 15 hours total—this week was more involved in actually coding and planning things.

**Accomplishments:**

Number of Story Points: 1

Number of Stories: 2

Personal Velocity: Good

Project Velocity: Good

Earned Value: 2?

**Performance:** The pace was slow at first, but it sped up quickly in the second week.

**Challenges:**

I would say the biggest problem this sprint was communication. Not only was communication between the teams delayed, I had trouble properly talking with my lead at times, given his accent and a lack of shared terminology we use. I can still manage, but it is a concern I have, since the issues were brought to the surface even more than before.

**Plans:** As before, continuing development of the backend. Not much else to say here.