

# Software Engineering HW1 Team 10

By Caleb Klenda, Drew Bender, Nick Kaminski, Lam Nguyen, Donald Wedding

## Task 1: Term Project

a. See SRS PDF

b.

Column1	Power on Raspberry Pi	Maintain Github code	Logical Database	Be able to communicate events	Jenkins pipeline for constant updates	Complete code for the Discord bot
Writing data to Raspberry Pi	x			x	x	
Schedule reminder	x	x	x	x		x
Schedule meeting	x	x	x	x		x
Log data from users			x	x		x
Keep Pi active 99.99% of time	x	x		x	x	
Update software for scheduling	x	x			x	
Restrict software access		x	x			x
Easy restart if corrupted	x	x			x	
Ease of portability	x					
Total:	7	6	4	5	4	4

Priority index:

[1] Power on Raspberry Pi - The team needs to be able to power on the raspberry pi before anything can be hosted or ran.

[2] Maintain Github Code - The team will be using Github for version control. This will allow safe storage of the code, the ability to easily update the code, and to work with the Jenkins pipeline for continuous updates.

[3] Be able to communicate events - This is the step where the discord bot would be sending messages to the users to send reminders and meetings.

[4] Jenkins pipeline - This allows for the bot to be constantly running and able to be updated easily.

[5] Complete code for the Discord bot - This will be when the team has completed the code for the bot and are ready to start running it on the pi.

[6] Logical Database - This is the creation of the database to store the data of the user's so a schedule can be made.

## Task 2: Team Structure

a.

Role on Team	Responsibilities	Skills Needed
Scrum Master	<ul style="list-style-type: none"><li>Facilitate scrum meetings</li><li>Manage Sprint backlog</li><li>Create tickets as work arises</li><li>Participates in code reviews</li><li>Handle other administrative task</li></ul>	<ul style="list-style-type: none"><li>Good communication</li><li>Organization</li><li>Github/Zenhub</li><li>Understanding of processes</li></ul>
System Architect	<ul style="list-style-type: none"><li>Design and document plan for the project</li><li>Create tickets from the design</li><li>Enforce the design in code reviews</li></ul>	<ul style="list-style-type: none"><li>Strong understanding interactions between different parts of project</li></ul>

	<ul style="list-style-type: none"> <li>• Diagnose and fix issues from integration</li> <li>• Participates in code reviews</li> </ul>	<ul style="list-style-type: none"> <li>• Python</li> <li>• Discord API</li> </ul>
DevOps	<ul style="list-style-type: none"> <li>• Implement the Continuous Integration Pipeline through Jenkins</li> <li>• Connect Github to the Pipeline</li> <li>• Address network/connectivity issues</li> <li>• Ensure team understanding of pipeline</li> <li>• Maintain and fix issues with the pipeline</li> <li>• Participates in code reviews</li> </ul>	<ul style="list-style-type: none"> <li>• Grasps CI/CD</li> <li>• Jenkins</li> <li>• Familiarity with Raspberry Pi</li> <li>• Github API</li> </ul>
QA Tester	<ul style="list-style-type: none"> <li>• Understands system requirements</li> <li>• Tests code thoroughly</li> <li>• Participates in code reviews</li> <li>• Reports bugs when they arise, and documents process to reproduce</li> </ul>	<ul style="list-style-type: none"> <li>• Raspberry Pi</li> <li>• Discord API</li> <li>• Python</li> </ul>
Developer	<ul style="list-style-type: none"> <li>• Completes tickets in a timely manner</li> <li>• Works with team to establish priorities</li> <li>• Writes testable, readable code</li> <li>• Participates in code reviews</li> </ul>	<ul style="list-style-type: none"> <li>• Python</li> <li>• Discord API</li> </ul>

b.

Software Process	Implementation
Agile	We will have 2 week sprints which will begin with a Sprint Planning session. During the planning, we will rate tickets and assign them, as well as create any new tickets based on the current status of the project. Throughout the sprint, we will have scrum meetings as needed to discuss progress and roadblocks. The scrum master will lead these meetings and manage the tickets, while the rest will
Test-Driven Development	As part of the design, we will come up with a framework for testing all the code we write. Development will be encouraged to follow the red-green-refactor cycle. The CI system will run these tests on every PR and prevent a merge until all tests are passing

c.

Contingency	Continuation
Network Connectivity on Pi is too unstable to be usable	The Pi is nice because they are available and easy to use. However, if network connectivity is an issue (due to apartment wifi being no so great) we can easily port everything to a virtual server or simply run off an individual's desktop.
Discord API limits functionality	Many tasks with the bot are simple, but others may end up being quite complex. The Discord API may not support certain actions we wish to accomplish. In this case, we would have to write our own implementation through the APIs supported actions.
Continuous Integration fails to function	We are planning to use Jenkins, but there are other open source CI/CD options out there. If Jenkins fails us, we will swap to another.

d.

Metric	Tracking
Team Velocity via Story Points	Rating of tickets will allow the team to build an average velocity over several sprints. This will be represented in a burn down chart. This will allow us to estimate our overall time to complete the project, and track our productivity.
Tickets Completed	Total number of tickets completed will also be used as a measurement of total work completed
Sprint Retrospective	At the end of each sprint, a short retrospective will be held about the current process, with each member listing things we should keep doing, new things to add, and things to stop regarding the current process. This will allow us to adapt as the semester progresses.

### Task 3: Initial Estimation

a.

Role	Members with Role	Responsibilities
Scrum Master	Caleb Klenda	Will assist with all tasks
System Architect	Caleb Klenda	Creation of tasks
Dev Ops	Drew Bender	Creating and maintaining the pipeline.
QA Tester	Drew Bender	Participates in Code reviews and Implements TDD on the raspberry pi.
Developer	Nick Kaminski, Donald Wedding, Lam Nguyen	Develop all python code required for the discord bot.

b.

Task	Description	Time Estimate (hours)	Roles Responsible
Starter Code	We will need some skeleton code to start from for the bot	8	Developer
Discord Developer Account & Bot	We will want a common discord developer account to manage the bot and access tokens since any token shared on the internet will be flagged by Discord and deactivated	2	Developer, System Architect
Raspberry Pi starts bot on boot	The raspberry should start the bot when it turns on. The easiest way to do this is most likely docker, but implementation details are left to the developer.	4	Dev Ops,
Jenkins Job Creation	A job should be created whenever we create a pull request to our master branch on this repo.	6	Dev Ops
Job Restarts Bot on merge to master	When we update our master branch, we want the bot (wherever it lives) to restart with the new code.	4	Dev Ops
Bot Design - Architecture	Create a modular and documented design for the architecture of the code, including.	8	System Architect, Developer
Bot Design -	Create a modular and	8	System Architect,

Features	documented design for the features of the bot.		Developer
Bot Design - Testing Plan	We need a way to test the bot's functionality, preferably in a way that is not just manually sending commands to it in a server.	8	QA tester, Developer

Total Time in Hours: 58

This is all tasks that exist initially, likely more tasks will be added as development progresses.

#### Task 4. Languages, Tools, and Resources Assessment

Identify languages, tools and resources needed for the project.

Language: Python, Git

Tools: Raspberry Pi, Jenkins

Resources: Github, Zenhub, discord.py, Raspberry Pi,

a. Note the merits and shortcomings of tool candidates and the made choice.

Language/Tool/ Resource	Pro	Con
Python	Python is an easier language to use even with its power, and the group has tons of experience with it. It also has an extremely large number of useful libraries for this project, such as discord.py, which is great for creating discord bots easily. It's also extremely portable, able to run on a Raspberry Pi without any issue.	Python is somewhat slower than other languages (such as C) and uses a larger amount of memory.
Git/Github/ Zenhub	Git, Github, and Zenhub come together to allow a larger group of engineers to work together on coding projects without any issues. Github also has great documentation, and Zenhub allows for some useful tools, such as a board for easy access to tasks.	Not everyone on the team has experience with Git/Github/Zenhub.
Raspberry Pi	Small, portable, and powerful, the Raspberry Pi can more than handle discord bots without any issue. The group has experience using it, and the Pi is relatively cheap.	Raspberry Pis are weaker than most laptops or servers.
Jenkins	Allows for an easy pipeline to quickly update the discord bot as it is coded rather than having to manually update the bot. It is an extremely versatile tool, with tons of plugins for easier use and higher compatibility.	Not everyone on the team has experience with Jenkins

b. List the resources you project will need to use and contact CECS IT personnel for availability.

The only resource we need is a Raspberry Pi, but 3 of our members already own one, so we do not need to get one from CECS IT.

Sources for number 4:

<https://techvidvan.com/tutorials/python-advantages-and-disadvantages/>

<https://www.jenkins.io/>