# Design a simplified process or framework for implementing DevOps throughout the company

To Implement DevOps for the first time in a company will definitely be a trying mission. Instead of implementing it as a whole, it would be ideal to progress stage by stage. It’s a continuous improvement process. With a team of a rookie and a veteran, whole journey will be a roller-coaster ride.

As a First step, I would ask Jalen to identify a team to start the process with. Once the team is identified, the next step will be to understand how the team works and identify the bottlenecks. Ren will be able to easily identify and document those with the amount of experience he possesses. The next step will be to schedule daily meetings with the team to discuss how to facelift. It would be wise and productive to introduce Agile as a first step of DevOps. Because Agile and DevOps working together yields more benefits.

Initially we will have to breakdown the daily tasks to be done into smaller chunks and document it as a user stories and tasks. For this we can introduce new tools like Jira or Azure DevOps to the team where they will look for their daily work. Developer or testers will not work on any other tasks except the daily tasks which will be a smaller job. Ren can act as a scrum master for the time being and will schedule daily meetings with the team to review the daily tasks completed. We will have Sprint Planning, Backlog Refinement and other meetings introduced on sprint basis. Gradually, by showing the effort and iterations graph, we can increase the amount of work to be completed by the developers. The quality and quantity of the work are guaranteed to rise as developers gradually adapt to the new ways of working. This will in turn decrease the time to market. Also, the microservices architecture should be introduced.

In parallel, we will have to introduce the CI/CD to the team. Jalen can do some research on the new and effective tools in the market which can be introduced to the team. Because, Once the developers complete the coding, we should be able to test the code in a testing environment without any delay. For fast paced deployment of the code to servers, we need deployment tools like Jenkins, Octopus or Azure DevOps Pipelines. When the CI/CD is implemented, the moment a developer check-in his code, a build will be triggered and the code will be deployed to the testing environment.

Once the CI/CD pipeline is almost stable, we can introduce extra features in the build pipeline. We can have build validation, Unit testing in the build pipeline with VSTest or JUnit, keep a criterion of 85% code coverage, Static code scans with fortify or checkmarx, Code quality check with likes of sonarqube. When we get a report of SAST scans, we can ask the developers to fix the code vulnerability with some SLAs. Also, we can improve on quality of the code by running through the code quality report. With these features added in the build pipeline, the coding quality of the developers will gradually increase.

We can now extend the deployment pipeline to deploy the code till production environment and can also have the automated regression testing happening right after the deployment. We can use tools like ReadyAPI, Postman and Selenium for this.

While these steps are getting stable, we can have the monitoring in place for the production environment. We can use tools like elastic search, Splunk, Dynatrace for monitoring. The alerts will be worked on by the developers based on the errors. Developers analyse the errors to find if it’s a bug in the code or any environmental issues.

Please refer the Figure 1 to find the DevOps Process Roadmap.

It is a certainty that the team will be much more efficient when agile process is followed diligently. However, Continuous improvement at every stage is the key in creating an efficient team.

# Design a process or framework for modernizing existing code

Investing in application modernization can improve customer and employee experiences, ensure that applications continue to be responsive to changing business needs, and ensure that organizations stay ahead of the competition. I would prefer to have the existing code rewritten in Python. The information can only be retrieved from Windows servers by PowerShell scripts at the moment. To meet future expansion goals, systems might get moved to Linux or the cloud. So, Python works better than PowerShell. The existing script should be written in a way to identify the OS where the script is getting executed and it should provide the event logs based on the OS. Create a user story to understand what the script or program actually does. We have to list out the dependencies and modules of the PowerShell script. Separate out the modules in separate scripts and call through an API, Implement microservices architecture . Do a study of the dependencies and see if the same dependencies are required for the new Python script. Each module can be moved to different user stories and can be handled by different programmers. Also, the study of the dependencies can be moved to a user story. Program should be moved to a version control system so that any mistakes can be easily compared with earlier versions. Also, multiple developers can work on the same code if its in a version control system like GIT.

# Review the code provided to demonstrate how your modernization processes would work

# Conclusion

# Appendices

Figure 1: DevOps Process RoadMap

