**ATLANTIC TECHNOLOGICAL UNIVERSITY**

ASSIGNMENT COVER SHEET

To Be Completed By The Student

Lecturer’s Name: RUTH LENNON

Assessment Title: DevOps Implementation & Code Modernization in an Legacy System

Submission Date: 25th November 2022

Student’s Name: ARAVIND GIREESH KUMAR Id. Number: L00170985

Course / Stage: Masters in DevOps

Subject/Module: IaC for DevOps Pipelines

Word Limit: Actual Word Count:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

I confirm that the work submitted has been produced solely through my own efforts.

Student’s signature: Aravind Gireesh Kumar Date: 25/11/2022

|  |
| --- |
| **Note**: **PENALTIES**   * The total marks available for an assessment will be reduced by 15% for work submitted up to one week late. The total marks available are reduced by 30% for work up to two weeks late. * Assessment work received more than two weeks late, without prior approval by the lecturer will receive a mark of zero. * Marks awarded will be reduced by 10 % if submitted work is greater than 10% above or below the assigned word limit. * A further hard or electronic copy of your submitted work may be requested, and therefore you must keep a copy on disc. * Incidents of alleged plagiarism and cheating are dealt with in accordance with the Institute’s Assessment Regulations   **Plagiarism:** Presenting the ideas, words of someone else without proper acknowledgement. Refer to the Institutes’ procedures and guidelines for the assessment of learners. |

## DevOps Implementation Framework

Implementing a DevOps is not a easy task as it will take a lot of time to introduce the whole DevOps methodology into system. It can only be achieved only by gradually adopting different aspects of the agile methods of DevOps into the existing system by utilizing the experience as well as the fresh mind set of Ren and Jalen.

Chart

Description automatically generated

### Step 1: Requirement gathering and introduction to Agile

By the term requirement gathering I meant about identifying the right business that has to be prioritized on the basis of criteria such as low cost, fast implementation and the profit that brings to the organisation in a short period of time. For this analysis I will set up a meeting with the stakeholders to understand the different businesses involved and identify the critical task for my firm. As an introduction to agile methodology, we will set up a project tracking tools such as Jira for the daily scrum calls, scrum boards and the project road maps. Now the identified critical task will be broken down into smaller pieces and the smaller tasks are shared among the team members, which will be discussed on a daily basis in the form of scrum call. Gradually by showing the achieved tasks and the progress of the team in graphical way will boost the confidence in the team as for most of the members this is a new way of working.

### Step 2: Infrastructure Restructuring

Parallelly, Ren can interact with the Infra personals in our organisation and get the information regarding the infrastructure properties of the legacy system and then we have to setup the workspace in any of the Cloud platforms such as Azure DevOps or AWS, because of the minimised human error and ensured reliability and scalability of the environment resource, and are also cost effective compared to the traditional infrastructure setup. Now for implementation of the continuous integration and build automation, I prefer to introduce each CI components one by one instead of a leap into CI. As a first step, I would introduce the team with a centralized or distributed version controlling tools (SVN/Git). Before that with the help of Ren we can understand if there are any existing version controlling enabled in the current system and if so then we have to identify whether it support with the new technologies of CI tools. Also we can get the help of Jalen to identify a good version controlling tools as he will be aware of the latest technologies. Similarly the build management tools like Maven or .NET will be placed. Once these two necessary tools are in place, CI/build automation tools like Jenkins or GitHub Actions can be set up to automatically built the code. Once the CI is done, a test suite has to be created for testing the integration and the functional quality of the code. Once the test is successful, the developed code will be pushed to the Dev environment. It can be done by automating the deployment process and standardizing it. It is crucial to reuse the same deployment script across environments and application types by parameterizing it. The deployment will be validated by the Developers and will be send to the testing environment for it to be validated by the testers.

### Step 3: User Acceptance Testing

After the successful working of the integrated code in FSB2, the updated code has to be moved into the UAT environment for the business testing. UAT is almost similarly configured to the production environment. For this deployment the approval from a PO is required. Once the code is deployed with the PO’s approval, it will be tested by the business user. After the successful confirmation from User, they can be moved to production.

### Step 4: Performance Monitoring

The monitoring automation will also be adopted into the legacy system to analyze the business expectation at the end-user, performance bottle-neck etc.. Additionally, a warning mechanism will also be set up in case of any exception. So that such issues can be addressed as quickly as feasible by the development team. We can use the tools like Splunk and AppDynamics for the end-to-end monitoring and to measure the performance metrices.

Diagram

Description automatically generated

## Process for Modernizing Legacy Code and PowerShell Code Review

It is really challenging to work with code that we didn't build, and I have no prior experience with the legacy coding languages that are used in my company. Particularly PowerShell, as we need to start repairing the networking functionality in that given piece of code. One option would be to start from scratch, but before making that choice, I would conduct some research to ensure that it is compatible with the legacy code.

I will start by going over the previous documentation on the functionality and origins of the code. This is my first attempt at modernizing the code since, without the necessary knowledge or comprehension of the code's functioning, I risk making mistakes that could result in inappropriate behaviour. With Ren's cooperation, I will now work with a developer who is more knowledgeable about the functionality or scripting language so that I can better comprehend the code.  This practice will go much more quickly because we can readily understand the code from someone who is familiar with the codebase. Once I have a basic understanding of the code, though, I would like run a test on it to assist me understand how it actually functions and to identify any potential issues in the PowerShell script. In my situation, it has previously been stated that the code is broken and my primary job is to fix it.

If the code is not broken, I would start refactoring or cleaning it up. Otherwise, if the script fails the test, we must either rewrite the code using the most recent stack, such as Bash, PowerShell Core, etc., with the business' approval. Or locate the problematic component and update the code. However, in our case, the code is broken as a result of the poor logic implementation that the Intern has added mentioned in the comment area. There were multiple lines of code that were incomplete and commented out. Therefore, I will finish the remaining code one by one with the assistance of the expert developer and test the update. Upon completing this, I will begin refactoring the existing code, which I think to be one of the key processes because it is preferable to restructure rather than rewrite the code because the latter demands a significant amount of time and money to add resources to create. In order to make sure the code is clean, I will perform a quality and code coverage analysis. By following the above mentioned process I believe I can implement modernization into the existing legacy code at my firm.

## Conclusion

The adoption of DevOps concepts together for Shinty Software in the process of transitioning of from legacy systems to more modern practices represents a journey of re-engineering, restructuring, and encouraging businesses to operate in new ways with automated, and intelligent processes. In this assignment we have considered of implementing DevOps into a small part of the whole project, but when analysed in a high level the modernization roadmap of the whole system consists of different activities which includes standardization, automation, shift left etc. and this would take around two to three years to completely implement because this is all together a new way of working for the existing the employees in the firm.

We examined technical difficulties in great detail, but teams working on old systems must also adopt Agile and CD practices if they are to successfully adopt the DevOps technique for a legacy system. These teams frequently spend a long time building the functionalities without deploying them into an environment similar to production. A product undergoes a one-time complex and challenging deployment when it is released into the production environment and handed over to the operation team. The effectiveness of a CD pipeline that can release features into production on a daily basis depends on how quickly features are developed and how well the operations team can handle regular updates from the development team.

Through this assignment I have understood that agile alone cannot bring the expected results in the early stages of development, but the combination with DevOps methodologies can do so to the organisation because of the continuous integration, testing and deployment. I believe that the testing is the most important phase of SDLC. Automating the testing much early in the development can help errors be found and resolved more quickly, allowing developers to concentrate on the build's quality rather than waiting for mistakes to happen. Additionally, they can avoid longer delays by providing feedback and implementing iterative adjustments as soon as possible. Due to the fact that it is more cost-effective to patch a bug in development than in production, this procedure also lowers the cost of bug fixes, which is one of the major profitability that the business look forward to.

## References & Bibliography

Amdaris. (2021). How DevOps can bring legacy systems back to life. [online] Available at: https://amdaris.com/how-devops-can-bring-legacy-systems-back-to-life/.

Bellairs, R. (n.d.). 8 Tips for Working With Legacy Code. [online] Perforce Software. Available at: https://www.perforce.com/blog/qac/8-tips-working-legacy-code#what [Accessed 25 Nov. 2022].

says, img alt=’’ src=’https://www spec-india com/wp-content/wphb-cache/gravatar/a00/a007dff416a7f8a6cd31a397899fc1eax42 jpg’ srcset=’https://www spec-india com/wp-content/wphb-cache/gravatar/a00/a007dff416a7f8a6cd31a397899fc1eax84 jpg 2x’ class=’avatar avatar-42 photo’ height=’42’ width=’42’ /> A.M. (n.d.). A DevOps Approach To Legacy System Modernization. [online] SPEC INDIA. Available at: https://www.spec-india.com/blog/legacy-modernization-with-devops [Accessed 25 Nov. 2022].

Urolime Blog. (2019). How to Implement DevOps in Legacy System? Part 2. [online] Available at: https://www.urolime.com/blogs/how-to-implement-devops-in-legacy-system-part-2/ [Accessed 25 Nov. 2022].

Urolime Blog. (2019). How to Implement DevOps in Legacy System? [online] Available at: https://www.urolime.com/blogs/how-to-implement-devops-in-legacy-system/.