|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UMGC Capstone | | | | |
|  |  | | |  |
| DevSecOps | | | | |
|  | | Project Plan |  | |

TABLE OF CONTENTS

[Introduction 4](#_Toc55324544)

[1. Executive Summary 5](#_Toc55324545)

[Business Needs 5](#_Toc55324546)

[Objectives 5](#_Toc55324547)

[Scope 6](#_Toc55324548)

[Definitions, Acronyms, Abbreviations 6](#_Toc55324549)

[2. Organization Overview 7](#_Toc55324550)

[Internal Structure 7](#_Toc55324551)

[External Structure 7](#_Toc55324552)

[Roles and Responsibilities 8](#_Toc55324553)

[Communication Management 9](#_Toc55324554)

[Cost Analysis 10](#_Toc55324555)

[3. Execution 11](#_Toc55324556)

[Methodology 11](#_Toc55324557)

[Key Deliverables 11](#_Toc55324558)

[Milestones 15](#_Toc55324559)

[Change Management 16](#_Toc55324560)

[Assumptions and Constraints 16](#_Toc55324561)

[Risk Analysis 16](#_Toc55324562)

[4. Test Plan 18](#_Toc55324563)

[Scope & Test Items 18](#_Toc55324564)

[Pipeline Infrastructure Risk Issues 18](#_Toc55324565)

[Strategy 18](#_Toc55324566)

[Test Cases 19](#_Toc55324567)

[5. Future Work 22](#_Toc55324568)

[Project schedule 24](#_Toc55324569)

[AppendIx 25](#_Toc55324570)

# Introduction

This document outlines the UMGC software engineering capstone DevSecOps project plan. The UMGC software engineering capstone project requires a comprehensive examination of tools, skills, and techniques of software engineering. The capstone project structure resembles a small company with individual development teams that each operate within the context of their own product. This document serves as the formal DevSecOps project plan as a guide to both project execution and project control. The primary uses of this document are to define planning assumptions and decisions to facilitate communication, identify scope, cost, and schedule.

This remainder of this document will operate within the context of the UMGC software engineering capstone project context, referring to the business as the class as a whole and project teams as development teams.

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reasons for change | Version |
| Glenn Goodlett | 9/3/20 | Initial Draft | V 0.1 |
| Dustin Emerson | 9/4/20 | Second Draft | V 0.2 |
| Glenn Goodlett | 9/8/20 | Internal Review Changes | V 1.0 |
| Glenn Goodlett | 9/21/20 | Change Request: ADF | V 1.1 |
| Glenn Goodlett | 10/10/20 | Change Request: Azure TDD | V 1.2 |
| Dustin Emerson | 10/11/20 | Internal Review Changes | V 2.0 |
| Glenn Goodlett | 10/15/20 | Change Request: Programmer Guide | V 2.1 |
| Glenn Goodlett | 10/18/20 | Change Request: Runbook | V 2.2 |
| Dustin Emerson | 10/19/20 | Internal Review Changes | V 3.0 |

# Executive Summary

## **Business Needs**

The business requires support for three development teams tasked with the discovery, collection, and implementation of a software project as identified by the development teams project plans. The three development teams included, and their projects are:

* City ChatBot
  + A suite of services with the ultimate goal of guiding users through the complicated process of acquiring city permits. Services included are a Chatbot, a zoning/permit management system, a map layer to serve zoning information.
* Virtual Letter of Life
  + An emergency medical services which will utilize QR codes on responders with pertinent medical conditions and illnesses when a patient cannot provide this information themselves.
* Project Tracker
  + A web-based application which allows customer, clients, and former students to submit detailed proposals of projects to be designed and implemented by UMGC SWEN 670 students as well as track the stages during the approval process.

The DevSecOps team complements the established project teams, and its charter is to assist in each of the projects until they have reached project completion. This project plan expresses how the ream will fulfill the objectives of the business needs.

## Objectives

The DevSecOps team exists to provide cohesion between the individual project reams and to serve as a unifying agent for the selection of industry-standard Tools, Techniques, and Practices (TTP). Guided by the Project Coordinator, the DevSecOps team primary objects are as follows:

* Develop an internal structure for the software development lifecycle across project teams
* Advise project teams through the respective Project Manager of best practices discovered through both external research and intra-team experiences,
* Create standardized instructional documentation for everyday project needs.
* Provide technical design documents to aid in continuity across the business.
* Innovate that future projects benefit from the outcomes of other projects.

## Scope

Operating under the guidelines of the Project Coordinator and, to the extent possible, the guidance of the DevSecOps team, the recommendation provided to the individual project teams represents an opinionated architecture structure for the development, testing, delivery of each of the client's requirement. Within the DevSecOps scope of operations, the team contributes to the term's project execution by supporting the following tasks:

* Initialize and maintain official project repositories.
* Contribute to design documents for the implementation of project teams.
* Identify the best options for no-cost development/delivery of project teams.
* Support project managers in the collection and clarification of client requirements.
* Provide overall code reviews to establish and enforce an intra-team, agreed-upon development standards.
* Support project development teams through setup and utilization of recommended development technology stacks.

Notwithstanding, as an optional component of the project team's internal development strategy, deviation from any standardized architecture plan limits the DevOps team's ability to meet the above taskings. The DevOps PM communicates any such limitation, if any exists, through the respective team's Project Manager and the Project Coordinator, as described in the below communication plan.

## Definitions, Acronyms, Abbreviations

Terminology in this project plan targets individuals familiar with industry-standard software development planning, practices, and concepts. Also, it assumes a knowledge of general project planning vernacular. While most are self-explanatory, the following items are explicitly defined:

* DevSecOps: Development security and operations is the process of integrating secure development best practices and methodologies into development and deployment processes.
* TTP: Tactics, Techniques, and Procedures describe the process of applying industry standards to the development, testing, and delivery of robust software architecture.

# Organization Overview

## Internal Structure

The DevSecOps team consists of two members, a project coordinator that interfaces with each project team. Each of the two members shares all roles and responsibilities as it pertains to interfacing with each project team. To reinforce the continuity of operations, DevSecOps team members have constant internal engagement throughout the lifespan of the project life to provide consistent recommendations and best practices between the project teams.

## External Structure

While the DevSecOps team aims to provide technical assistance to all project team members, per the below communication plan, the primary external interface between each of the development teams is through or in coordination with the respective Project Manager for any given team. This structure ensures that any DevSecOps requests or decisions have the vetting of all team members, and between Project Managers when appropriate.

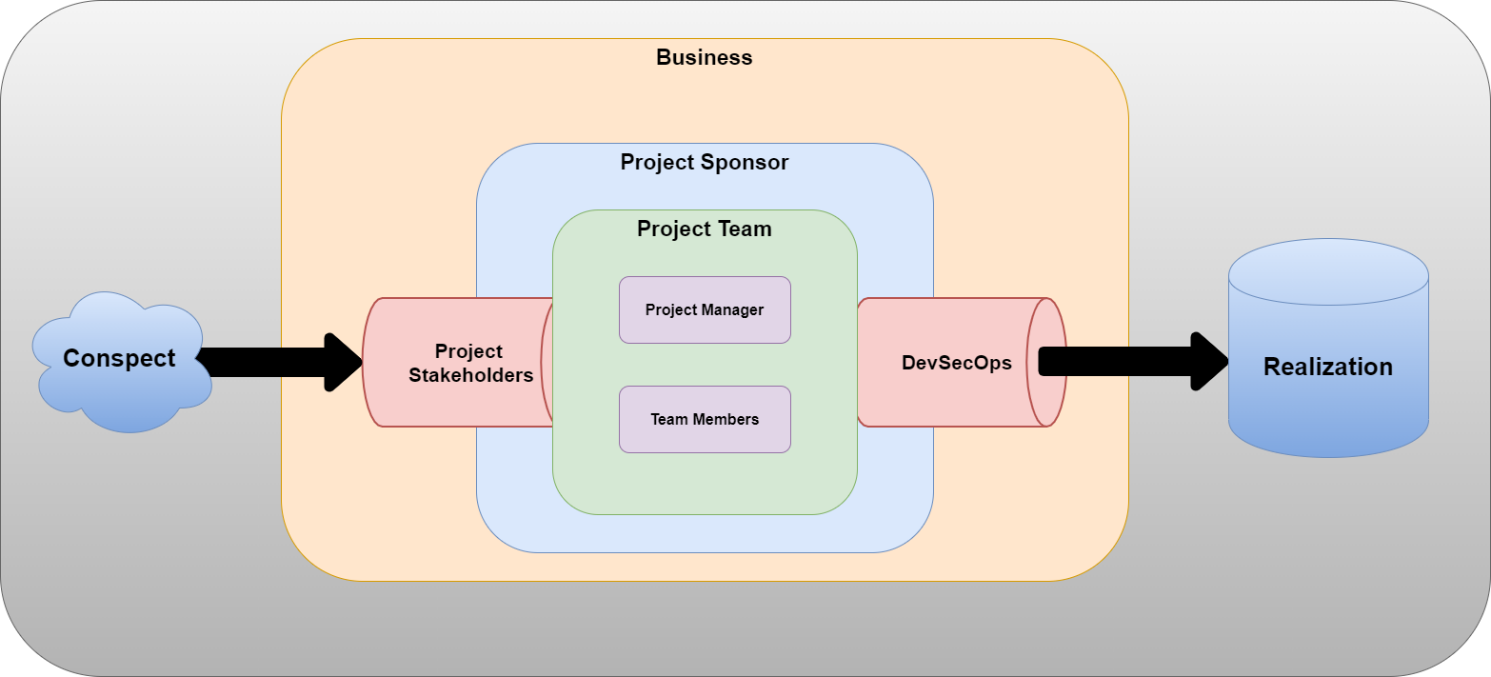


Figure Product Structure

## Roles and Responsibilities

At the time of this document inception, from a DevSecOps perspective, several vital roles are recognized and assigned to crucial business participants:

* Project Coordinator: Dr. Mir Assadullah is currently the sole proprietary of this role and serves as chief advisor of all projects. As such, the Project Coordinator may exercise authority to alter, limit, or expand any of the perimeters of the project(s) to meet Capstone requirements. This role is also known as the Project Advisor.
* Project Sponsors: Also identified as the stakeholder, client, or customer, the Project Sponsors are primary requirements drivers for each of the given projects. They are the evaluators who receive project deliverables after each of the project's development cycles and make the determination if deliverables meet the necessary acceptability, usability, and adherence to agreed-upon requirements.
* Project Managers: PMs are liaisons between their assigned team and all stakeholders. The PM communicates any needs or concerns from the teams to affected stakeholders and vice versa. Each team has identified a primary Project Manager; however, to allow for flexibility in team roles, teams may collectively decide to reassign the role during any particular phases of the project. Such a change must adhere to the communication plan to notify the core steering committee members
* Software Developers: Identified developers are responsible for utilizing Capstone-specified programming languages, tools, and infrastructure to write, test, and implement client specified applications. At times during the production effort, it may be necessary for DevOps team members to interface directly with a given project's assigned developers outside of the specified communication plan to establish processes, clarify requirements, or perform similar related items to meet the overall project goals.
* Steering Committee: Refers to those critical roles needed to make the majority of critical decisions that result in an impact on multiple team's internal processes or deliverables. As of this writing, those roles are:
  + Project Coordinator
  + Project Manager
  + Project Stakeholders when required

## Communication Management

Communication management ensures effective communication throughout the life of the project lifecycle. It indicates the means of how information is shared and follow up activities, which increases the effectiveness of the communication processes. This plan, coupled with the project teams' communication management strategies, plays an essential role in setting clear expectations for how and when the project updates are shared with the stakeholders and on increasing visibility of the project and status.

Approach

With interface requirements for each of the project teams, the DevOps PM leads the effort of ensuring effective communication within the Capstone construct. They communicate with the steering committee members to determine their preferred frequency and method of communication. At the current time, the steering committee is notified by weekly update emails to ensure projects on individual schedules. In addition to identifying communication frequency and method, the project's communication channels also identify and confirm that stakeholders have access to respective channels. After changes get approved, the plan is updated, and supporting documentation distributed to update the project team and all stakeholders.

Below are the current project team members and their assigned roles. Note: internal project team roles and responsibilities will not be outlined here just project managers and development team members

Business Directory

|  |  |  |
| --- | --- | --- |
| Name | Role | Contact |
| Professor Assadullah | Project Owner | mir.assadullah@faculty.umgc.edu |
| Jonny Lockhart | Project Sponsor | jlockhart2@student.umuc.edu |
| Roy Gordon | Project Sponsor | uspsrgordon@aol.com |

Chatbot Project Directory

|  |  |  |
| --- | --- | --- |
| Name | Role | Contact |
| Robert Lee | Project Manager | lee.robert001@gmail.com |
| Benjamin Fetterman | Development | ben.fetterman@gmail.com |
| Hanim Danur | Development | hdanur@student.umgc.edu |
| James Cornelius | Development | j aymcorn@gmail.com |
| Benjamin Murray | Development | b\_murray06@yahoo.com |

Project Proposal Project Directory

|  |  |  |
| --- | --- | --- |
| Name | Role | Contact |
| Kathryn Stewart | Project Manager | kathryns222@gmail.com |
| Tarun Lava | Project Manager | tejalava@gmail.com |
| Yonas Mekete | Development | ymekete@yahoo.com |
| Marc Bueno | Development | mcbueno2k@hotmail.com |
| Ephrem Kefle | Development | efysha@gmail.com |
| Bereket Tamrat | Development | bereket.m.tamrat@gmail.com |

Virtual Letter of Life Project Directory

|  |  |  |
| --- | --- | --- |
| Name | Role | Contact |
| Michael Shaw | Project Manager | 1michaelshaw@gmail.com |
| Heather Barnes | Project Manager | jluhbarnes86@gmail.com |
| Kimberly Van Allen | Development | kimberlyavanallen@gmail.com |
| Andrew Coleman | Development | coleman.andrew.d@gmail.com |
| Michael Marcucci | Development | onlycutch@gmail.com |
| Panhavorn Hok | Development | onehok@gmail.com |

## Cost Analysis

It is the mission of the DevSecOps team to provide all project teams and the Project-Sponsors a little to no cost solutions to on which to develop, deliver, and prototype software deliverables. Within these boundaries, the expected cost of all project-related expenditures is non-existence to neglectable. Providing a no-cost solution limits the capability of the deliverable products to operate in the realm of open-source, freely available platforms. If accepted, the Project Sponsor may expand the capabilities of the specified deliverables at their discretion; however, they accept all cost increase from the expansions of said products.

While the project teams may exercise their independence and utilized any for-cost products in their internal development operations; however, the team cannot pass such expenses on to the client among project deliverables. Lastly, after the Capstone specified timeline, the teams must offer their deliverables in a non-expenditure acquired deployment solution.

# Execution

Given the support nature of the DevSecOps roles and responsibility within a business structure, it is logical that any purposed internal project schedule put forth is significantly influenced by a dependent project teams' ability to meet and deliver artifacts on schedule. As such, the DevOps team reserves the right to make changes to the deliverable timeline within five days of a set milestone, utilizing the below communication plan.

## Methodology

The business has outlined a hard deadline of ten November. Development will be contained within a 12-weeks project lifecycle, starting 19 August and ending 10 November. A four-phase approach has been identified by Milestones one thru through, each time-box evenly in 3-weeks blocks. With this specification, each phase has a definitive start/stop marker, with measurable criteria for success (identified by the team-specified deliverable) that closes before the next phase begins. However, acknowledging the dynamic nature and scope of each software development initiative, changes, or decisions from earlier phases may have to be revisited. The included project schedule provides an outline of each of the major phases, with expected milestones.

## Key Deliverables

Project Plan

This formal document provides a plan to both project execution and project control with the primary use to define planning assumptions and decisions to facilitate communication, identify scope, cost, and schedule

Version 1

This version is the initial release of the project plan.

Version 2

This version addresses project changes as required.

Version 3

This version addresses project changes as required.

Version 4

This version addresses project changes as required.

DevSecOps Plan

This formal document provides a guide to standardize project teams' development environment and life cycle to improve the core competencies associated with the software production effort, including a foundation of efficiency, assurance, and maintainability of the code creation process and delivery flow.

Version 1

This version is the initial draft (internal release) of the DevSecOps plan with the intent to give project teams a general direction of the DevSecOps team. Additionally, this draft will be made available to each project team for review and suggestions.

Version 2

This version is the initial release of the DevSecOps plan that is formalized and incorporates project teams' suggestions.

Version 3

This version addresses project changes as required.

Version 4

This version addresses project changes as required.

Source Control Management

This artifact is realized by the implementation and access of the project teams to a source control tool along with its supporting technologies for facilitating CI/CD functions.

Version 1

This version addresses the initial access to source control tools for each project team and its members.

Version 2

This version addresses the initial integration of the selected tooling outline in the DevSecOps plan.

Version 3

This version addresses changes to the source control management as need by project teams as additional functionality is needed.

CI/CD Management

This artifact is realized by the integration and automation of technologies that make possible application build, test, static analysis, and deployment.

Version 1

This version is the initial integration and automation outlined in the DevSecOps plan.

Version 2

This version addresses changes to the CI/CD management as need by project teams as additional functionality is needed.

Advance Development Factory (ADF)

This artifact is realized in the form of a Docker image that provides basic development environment packages and dependencies.

Version 1

This version of ADF includes the software and design requirements document.

Version 2

This version of the ADF includes the initial ADF Docker image that includes basic developer needs.

Version 3

This version of the ADF includes changes need to support additional requirements discovered during development lifecycle.

Azure TDD

This artifact is realized in the form of an Azure pipeline technical design document and outlines the pipeline design, strategy, and approach.

Version 1

This version of Azure TDD is the initial release of the TDD document.

Version 2

This version address changes to the TDD as needed.

General Support

This artifact is realized by the general role DevSecOps play within a business group. This is only noted as a critical deliverable because DevSecOps plays a vital role within each project team's success, and it's worth mentioning as a critical activity of the DevSecOps team.

DevSecOp Programmers Guide

This artifact is realized in the form of formal document detailing everything DevSecOps with the intended audience of future DevSecOp engineers.

Version 1

This version of DevSecOps Programmers Guide is the initial release of the TDD document.

DevSecOp Runbook

This artifact is realized in the form of formal document detailing how to deploy DevSecOps pipelines with the intended audience of future DevSecOp engineers.

Version 1

This version of DevSecOps Runbook is the initial release of the TDD document.

## Milestones

Milestone 1

* Project Plan v1
* DevSecOps Plan v1
* Source Control Management v1

Milestone 2

* Project Plan v2
* DevSecOps Plan v2
* Source Control Management v2
* ADF v1
* Azure TDD v1

Milestone 3

* Project Plan v3
* DevSecOps Plan v3
* Source Control Management v3
* DevSecOps Programmers Guide v1
* DevSecOps Runbook v1
* CI/CD Management v1
* ADF v2
* Azure TDD v2

Milestone 4

* Project Plan v4
* DevSecOps Plan v4
* Source Control Management v4
* CI/CD Management v2
* ADF v3

## Change Management

Change Management, in this context, refers to the changes that deviate from the outline in this document. Whether changes are trivial or complicated, the project team must manage changes to prevent scope creep. The CGMP includes submitting, reviewing, and approving changes at all echelons of the development process. All requests and submissions follow the change control process described below to ensure changes are approved, tracked, and managed per this CGMP.

The proposed CGMP approach confirms that changes throughout the project lifetime are defined, reviewed, and agreed on and adequately implemented and communicated to all involved parties. Also, it ensures that only changes within the scope of the project are considered. By following these processes, it is the goal of the DevSecOps team to prevent unnecessary changes from occurring and help the team members to focus only on beneficial changes within the project scope.

## Assumptions and Constraints

To enable each project team the freedom to operate as independent entities while also meeting the expectations levied by the Project Organizer, DevOps operates under the following assumptions and constraints:

* [Assumption] Timetables: As mentioned earlier, any provided timetable and schedules are heavily influenced by the ability of individual teams to formulate and deliver software.
* [Assumption] Production Value: All software architectures and implementation are design based on providing proof-of-concept and product prototyping.
* [Constraint] Architecture Requirements: Unless prior coordinate with the Project Coordinator, the selection of specific technologies is limited to this expressed in earlier requirements documents.
* [Constraint] Capstone Resources: Unless otherwise offered by the Project Sponsors, the Capstone provides no financial support for development efforts.

## Risk Analysis

The initial risk assessment attempts to ascertain, describe, rank, and track risks and their mitigations before project commencement. The risk assessment is continuously examined and kept up to date for the project lifecycle. Time should be allocated at each steering committee communications to identify possible new risks and determine mitigations.

Risk Category: CI/CD Account Access

The account used for CI/CD model or web hosting/deployment exceeds provided credits.

Risk Level: Medium-Low Even Likelihood: Medium-Low

Previous projects have demonstrated a viable technology toolset within the business project context; however, the risk could increase if DevSecOps intends to deviate from previously used technologies.

Risk Category: Project Team dependencies

Project teams are unable to complete project milestones or provide adequate artifacts to support DevSecOps functions.

Risk Level: Medium Even Likelihood: Low

It is unlike that project teams fail to deliver on project milestones; however, additional DevSecOps support will be required in the event artifacts are inadequate to support DevSecOps functions.

Risk Category: Azure Adoption

The business has required project teams to leverage Azure as the cloud computing service. Currently, the DevSecOps team has no previous experience with the Azure platform.

Risk Level: Medium Even Likelihood: Low

Although the DevSecOps team members have no Azure platform experiences, Azure has an extensive learning program and is widely used. It is unlikely the DevSecOps team will be unable to learn Azure.

# Test Plan

The nature of DevSecOps provides limited testing functionality; however, to ensure a seamless development process testing, the infrastructure provides by DevSecOps is outlined below.

## Scope & Test Items

The central artifact produce by the DevSecOps team is the pipeline infrastructure and will be the focus of the test plan. Testing the pipeline will be conducted continuously and frequently thought the capstone course as a deficiency in the infrastructure will limit other teams' capabilities.

## Pipeline Infrastructure Risk Issues

Identify below are the critical areas to test in the pipeline infrastructure:

* Application build process
* Application test process
* Application promote process
* Git pull request integration triggers
* Application test result publication

## Strategy

Testing the pipeline infrastructure is conducted by running the pipeline and observing the results. The testing results are considered successful if the critical areas outlined above are completed depending on the application's context. Pipelines are run on every pull request, and because the pipelines are declared in the applications git repo as a source-controlled file, pipelines are tested every time code changes. This ensures a systematic and continuous pipeline testing strategy. The following sections outline the test cases necessary to demonstrate the pipeline infrastructure capabilities.

## Test Cases

Test Case 01: Build

**Description**: This test validates the pipeline infrastructure can build an application.

**Requirements**: The system shall build applications.

**Prerequisites**: A pipeline infrastructure has been created to build a given application.

**Steps**:

* Trigger the pipeline via the Azure UI.

**Expected Output**:

* The Azure pipeline starts and builds the application verified by the Azure logs.

Test Case 02: Test

**Description**: This test validates the pipeline infrastructure can test an application.

**Requirements**: The system shall test applications

**Prerequisites**: A pipeline infrastructure has been created to build and test a given application.

**Steps**:

* Trigger the pipeline via the Azure UI.

**Expected Output**:

* The Azure pipeline starts and tests a given application verified by the Azure logs.

Test Case 03: Promote

**Description**: This test validates the pipeline infrastructure can promote an application.

**Requirements**: The system shall promote build artifacts.

**Prerequisites**: A pipeline infrastructure has been created to build, test, and promote a given application.

**Steps**:

* Trigger the pipeline via the Azure UI.

**Expected** **Output**:

* The Azure pipeline starts, builds, tests, and uploads artifacts to the configured location verified by the Azure logs and the artifact storage solution.

Test Case 04: Git Pull Requests

**Description**: This test validates the pipeline infrastructure can gate git pull requests.

**Requirements**: They system shall gate git pull requests based on pipeline success.

**Prerequisites**: A pipeline infrastructure has been and linked via git webhooks.

**Steps**:

* Create a test pull request.

**Expected Output**:

* The azure pipeline automatically triggers when a git pull request is created.
* After the pipeline is successful, git allows a merge into master.

Test Case 05: Publish Test Results

**Description**: This test validates the pipeline infrastructure can publish an application test results'.

**Requirements**: They system shall publish the results of application tests.

**Prerequisites**: A pipeline infrastructure has been created to build, test, and publish a given application test results'.

**Steps**:

* Create a test pull request.

**Expected Output**:

* The azure pipeline automatically triggers when a git pull request is created.
* In the git pull request requirements test results are viewable.

# Future Work

This section captures concepts to work towards in the future. Below are some ideas for future DevSecOps teams to take on during a capstone session.

Central Application Logging and Monitoring

Collection of application logs such that they can be ingested by a central logging services to allow for dashboards, logging, and alerts such as ELK.

Protentional Technologies:

* Syslog
* Elk
* Splunk

Project Issue Tracking

As applications mature an issue tracking system that allows for continuity between capstone groups such that know issues are not rediscovered each session.

Protentional Technologies:

* GitHub
* Jira

Project Feature Tracking

Like the project issue tracking this system would track request features for continuity between capstone groups.

Protentional Technologies:

* GitHub
* Jira

Docker Image scanning

This system would scan Docker images as they are promoted from a master/main branch.

Protentional Technologies:

* Docker Hub
* Twist Lock

# Project schedule

# AppendIx

* Project Management Institute. (2004). *A guide to the project management body of knowledge (PMBOK guide)*. Newtown Square, Pa: Project Management Institute.
* Eldridge, I., Eldridge, I., Wieldt, T., Casey, K., & Paul, F. (2020). SecDevOps: Injecting Security Into DevOps Processes., Retrieved from https://blog.newrelic.com/technology/what-is-secdev