Varsity College Durban North

**PROG7311. POE. Part 3**

2023

by

Jordan Green

Student Number: st10083222

**Version 1.2**

**Changes made:**

* The rubric said I needed to; add an Introduction and Conclusion heading plus better heading for topics, which I have added in full.
* The rubric said I needed to; add an explanation to the images/graphs provided, give a reason for why they are there, which I have added underneath the image.
* The rubric said I needed to; add a new non-functional requirement, and the lecturer advised Maintainability as a good choice, which I've used and added.
* The rubric said I needed to; explain why NRF's are necessary in a business, business value and monetary value. I have added a better explanation on these points.
* The rubric said I needed to; Explain what pattern I will use, which is confusing because I clearly stated I am using a MVC pattern, a pattern I'm experienced in.

Farm Central Proposal

**Introduction**

Farm Central is searching for a website proposal which details a fully functioning, but also user-friendly website, and we; the developers have prioritized these non-functional requirements (nfr), when considering the build of the website.

A diagram of a type of non-functional requirements

Description automatically generated with low confidence

The image above visualises varying types of non-functional requirements; cited from (altexsoft, 2022)

**Non‐functional requirements of high importance:**

* **Performance**

~ System performance indicates how quickly a system can react to a certain user's activity while handling demands (Jafari, 2020). Ex: The loading period when using a website is indicative of the performance.

* **Security**

~ Security measures protect your program's safety from tampering and cyberattacks (Jafari, 2020). Even standalone systems need these features because you don't want anyone to have access to your private data (Jafari, 2020). Ex: Only a logged-in employee can add a new farmer to the database.

* **Usability**

~ This characteristic affects the users because it shows how well they can pick up and operate a system/website (Jafari, 2020). Ex: The website must be user-friendly.

* **Manageability**

~ The capacity to effectively manage a system and maintain its full functionality is what is meant by this feature (Jafari, 2020). Ex: When editing a page, the application can still be used in other aspects.

* **Maintainability**

~ A maintainable system can include extra characteristics such as modifiability, configurability, extensibility, and interoperability. It must be able to be maintained efficiently over the course of its intended lifetime. (boxUK, 2023)

These non-functional requirements above were chosen because the stipulations given must be that the website is safe and accurate, user-friendly and streamlined for multiple parties. Ill address these requirements by programming and designing with industry standard techniques to make a fully functioning website with the necessary features needed. It will also be designed with what the accountant, the head of marketing, and the store manager have specified.

**How these non‐functional requirements will affect final software build:**

Impact wise, the non-functional requirements specified above don’t impact the website in a negative way, as these are common ways to build a website regardless of which non-functional requirements are used.

I’ve personally added Manageability requirement even though it’s not necessary based on the specifications given to me by the client, but I feel it will add features which will improve upon the user-friendly-ness of the website. The NRF’s stated above, will positively impact the business by making the website run more efficiently and thus improve the user-experience.

**Are design patterns and architecture patterns important:**

Design patterns and Architecture patterns are vital in designing various programs, but to really understand why these options are important, one needs to know what these patterns mean.

Design patterns

~ These are common solutions to fix common issues found in software design (Lazuardy, 2019). Ex: A pre-made blueprint which was built to solve a certain problem.

There are three classifications of design patterns:

1. Creational patterns (Lazuardy, 2019).
2. Structural patterns (Lazuardy, 2019).
3. Behavioural patterns (Lazuardy, 2019).

Architecture patterns

* These patterns dictate the layout of one’s application, and one’s architecture pattern will come with advantages and disadvantages (Lazuardy, 2019). These patterns are similar to design patterns but have a larger scope on which they work (Lazuardy, 2019).

Three common architectures:

1. Layered pattern (Lazuardy, 2019).
2. Client-server pattern (Lazuardy, 2019).
3. Master-slave pattern (Lazuardy, 2019).

Both Architectural and Design Patterns describes the "concept," but Architectural Patterns focuses in on the idea's abstract nature, whilst Design Patterns focuses on implementing the concept (Lazuardy, 2019).

**How should design and architecture patterns be applied in this build:**

These patterns should be included when building this application because it provides rules and structure to this build. It shows experience to use pre-built and tested ways to program an application. For example, this will be a website, so for Architectural Patterns, I’ll use Model-View-Controller (MVC), which is a pattern I’m experienced in using.

A diagram of a system

Description automatically generated with low confidence

MVC visualised above, showing how it relates to the user; cited from (GeeksforGeeks, 2021)

**How to improve the performance of the prototype:**

There are multiple ways to improve/optimize the performance of Farm Central’s built prototype website. Firstly, it is important to ensure that the website built is fully optimized for search engines so that can be readily available/easily found by future users. This can be done by ensuring that the website is mobile-friendly and responsive.

Another way to optimize the prototype website is to ensure that the site is built to for speed and performance. Ways to accomplish this is by compressing code, optimizing images and ensuring that the site is cache-ready. Continuing with the last point, having the site fully optimized for user experience, with clear navigation, easy-to-use menus, and clear calls to action.

**Guidelines to achieve peak performance:**

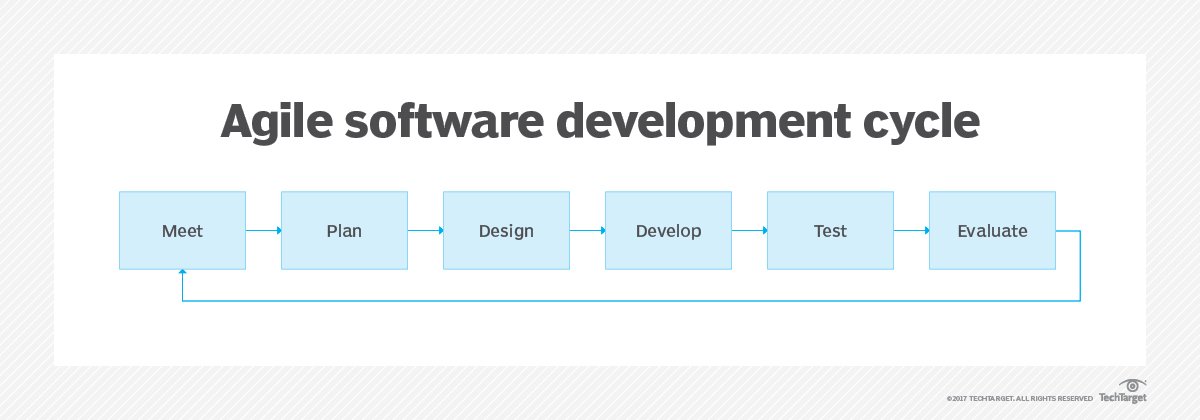
For guidelines, when developing the final presentation, there are several guidelines that should be followed to ensure that that it’s at an acceptable performance. These include making sure the code is clear and well-organized, utilizing a content management system (CMS) that is appropriate for the needs and objectives of the website, and making sure the site is adequately optimized for both user experience and search engine visibility.

In addition, it’s important to have security measures to protect the website from malicious attacks and to ensure that the website is fully accessible to users with disabilities. Finally, regular testing and monitoring the websites performance is essential to ensure it remains fully optimized and performing to acceptable standards. Overall, by following to these guidelines and best practices, the finished software can be produced to the highest standards, ensuring that the website is fully optimized, safe, and running up to expectations.

**Chosen software development methodology for this build:**

After analysing the non-functional requirements and the importance of design and architecture patterns for Farm Central, I would advise using the Agile software development technique for this project.

* Agile is a popular approach methodology that anticipates the need for flexibility, teamwork, and quick prototyping. For dynamic and complicated projects like web development, where numerous changes and iterations are required, agile approach is especially well-suited. (Brush, 2023)



The Agile software development cycle visualized; cited from (Brush, 2023)

Agile methodology consists of incremental development where requirements, design, and coding are tested continuously throughout the build. Agile methodology must have active participation from the development team, stakeholders, and the client. There must be active participation because the build requirements are rapidly changing, and it will allow the build to be delivered in a shorter time.

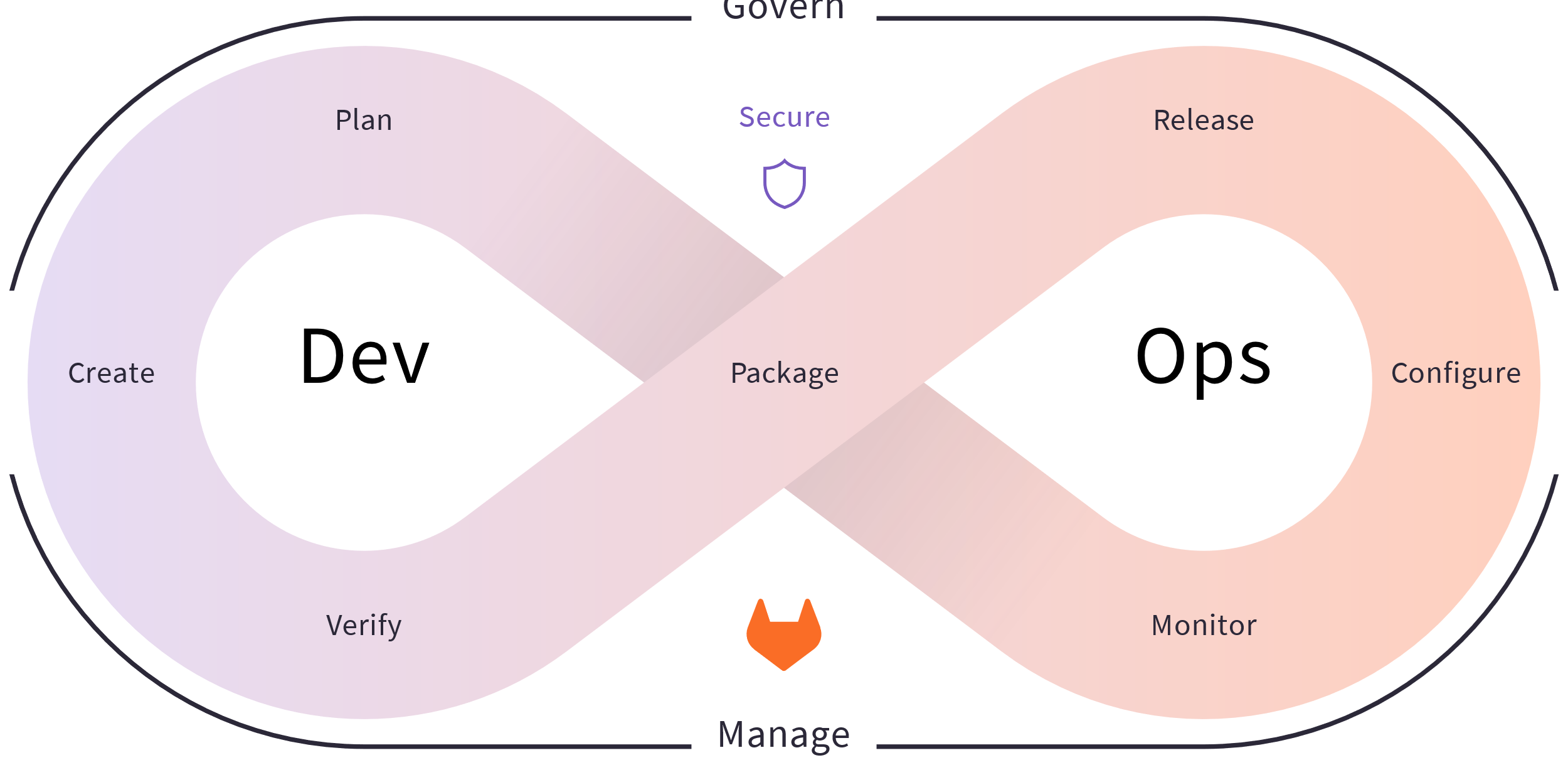
Agile methodology allows the development team to prioritize non-functional requirements alongside functional requirements throughout the development process, which will allow the final product to meet the client’s needs.

Lastly, agile methodology uses design patterns and architecture patterns in a flexible manner, which allows us (the development team) to be adaptable to changing requirements, and to integrate the patterns as per the project's needs. For example, Model-View-Controller (MVC) design, can be used in this development effort, taking into account the nature of the website.

**DevOps and if its applicable with Agile software development methodology:**

The Agile software development methodology is a proven approach that is being used for this project, but DevOps can be utilized as it complements this methodology by enabling teams to automate the software's testing, deployment, and monitoring processes, as well as regular feedback and improvement.

* DevOps is described as a methodology for software engineering that encourages a culture of cooperation and shared accountability with the goal of integrating the work of development teams and operations teams (GitLab, 2023).



DevOps visualized; cited from (GitLab, 2023)

Implementing DevOps in an Agile environment allows for improved communication, collaboration, and transparency between teams. Additionally, it makes it possible for quicker responses to shifting client requests and more frequent releases. DevOps makes sure that the operations and development teams are linked in terms of goals and procedures, resulting in a highly effective software development process.

In conclusion, having and implementing DevOps to our agile software development methodology is highly recommended, as it carries multiple benefits as stated above.

**Whether to use ITIL, the Zachman framework or TOGAF:**

Based on the details provided above, I would recommend a combination of ITIL, the Zachman framework, and TOGAF to optimize Farm Central's built prototype website and improve its non-functional requirements.

ITIL:

* ITIL is a collection of best practices for managing IT services and raising the caliber of IT support and service. Making sure IT services are in line with business goals—even when those goals change—is one of ITIL's major aims. (IBM, n.d)

ITIL would be helpful for improving the website's security protocols and ensuring that users with disabilities may use it without any barriers. The framework offers a standardized method for managing IT services and can assist Farm Central in putting in place efficient security measures and accessibility guidelines.

The Zachman framework:

* An enterprise architecture ontology called the Zachman Framework employs a schema to arrange architectural artifacts like design documents, requirements, and models. The Zachman Framework for Enterprise Architecture seeks to consider and integrate the artifact targets and the specific problem being solved. (LeanIX, 2023)

The Zachman framework can be applied to ensure that the website's design and architecture align with Farm Central's business needs and objectives. When guiding enterprise architecture principles there are few more comprehensive taxonomies than The Zachman Framework which can enable optimal web development decisions concerning organizational goals.

TOGAF:

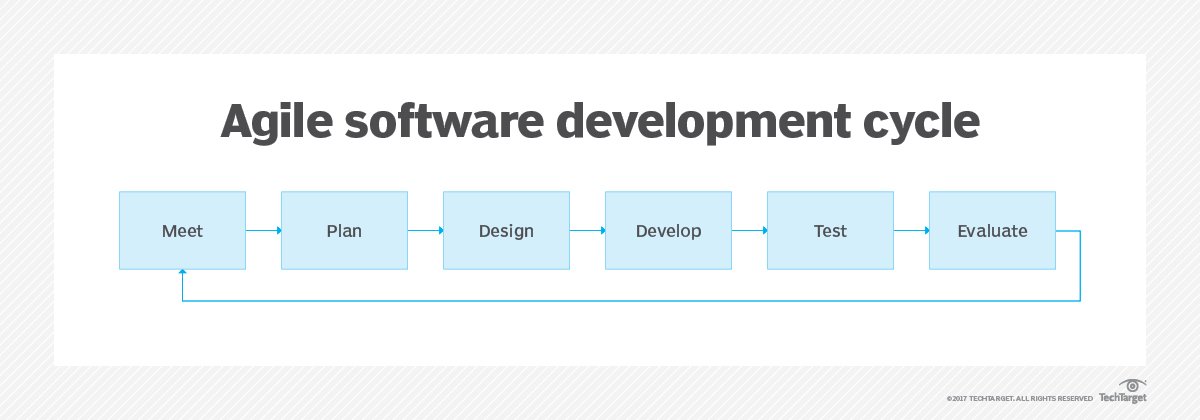
* A high-level framework for corporate software development is provided by the Open Group Architecture Framework (TOGAF), a methodology for enterprise architecture. Through a methodical methodology meant to cut down on errors, manage deadlines, keep within budgets, and integrate IT with business units to generate excellent outputs, TOGAF aids in organizing the development process. (CIO, 2022)

Lastly ensuring scalability alongside adaptability with leading best practices informed by TOGAF assures websites like those at Farm Central remain nimble over time making initiatives easy to maintain modularly.

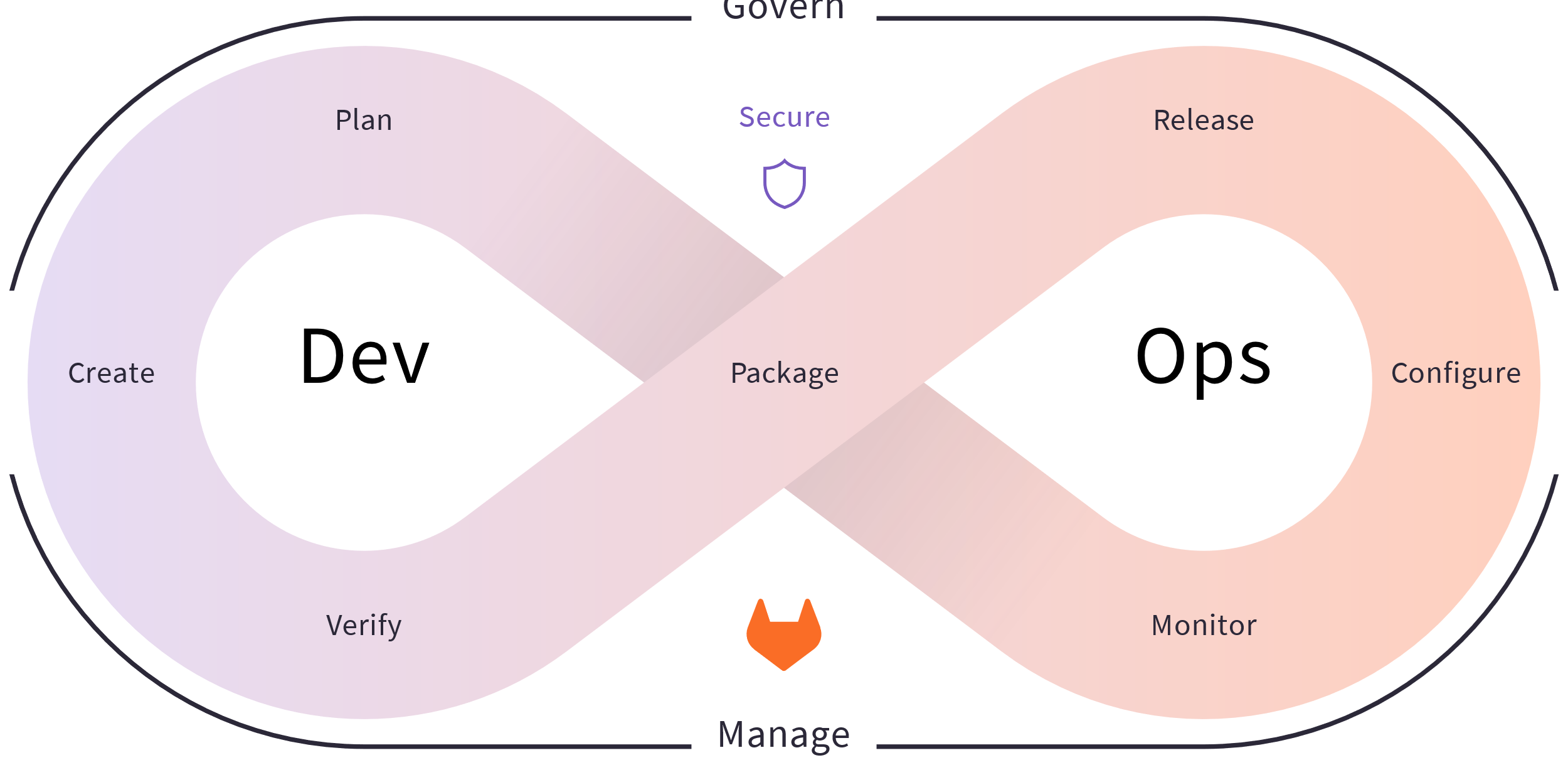
**Technical Solution Implemented in Prototype:**

* The technical solution implemented in the Farm Central prototype website involved developing a fully functional, user-friendly, and secure website that meets non-functional requirements such as performance, security, usability, manageability, and maintainability.

The website was designed and programmed using industry-standard techniques and architectures, such as the Agile development methodology, Model-View-Controller (MVC) architecture pattern, and DevOps, to ensure flexibility, teamwork, and quick prototyping, as well as automated testing, deployment, and monitoring processes.



The Agile software development cycle visualized; cited from (Brush, 2023)



DevOps visualized; cited from (GitLab, 2023)

Additionally, the ITIL, Zachman Framework, and TOGAF frameworks were utilized to optimize the website's non-functional requirements, ensuring that it aligns with Farm Central's business needs and objectives, as well as complying with accessibility guidelines and established security measures.

By following these industry-standard best practices in web development, the finished software can be produced to the highest standards, ensuring that the website is fully optimized, safe, and running up to expectations.

**Conclusion**

In conclusion, considerable attention has been paid to non-functional requirements such as performance, security, usability, manageability, and maintainability when developing the proposed Farm Central website. Model-View-Controller (MVC) is used as the architectural pattern, and design patterns and architecture patterns have been combined to enable flexibility, teamwork, and speedy prototyping.

Agile software development methodology has combined DevOps to ensure better teamwork, cooperation, and transparency. Additionally, the Zachman framework, TOGAF, and ITIL have all been used to verify that the best web development decisions have been made with regard to corporate goals, adherence to accessibility standards, and security precautions.

In conclusion, the software can be developed to the greatest standards, guaranteeing that the website is fully optimized, secure, and performing as expected.

# References

altexsoft. (2022, July 26). *Non-functional Requirements: Examples, Types, How to Approach*. Retrieved March 28, 2023, from altexsoft: https://www.altexsoft.com/blog/non-functional-requirements/

boxUK. (2023). *Guide to non-functional requirements: types and examples*. Retrieved June 20, 2023, from boxUK: https://www.boxuk.com/insight/guide-to-non-functional-requirements-types-and-examples/#:~:text=What%20is%20maintainability%20in%20non,%2C%20configurability%2C%20extensibility%20and%20interoperability.

Brush, K. (2023). *Agile software development*. Retrieved June 19, 2023, from TechTarget: https://www.techtarget.com/searchsoftwarequality/definition/agile-software-development

CIO. (2022, May 30). *What is TOGAF? An enterprise architecture methodology for business*. Retrieved June 20, 2023, from CIO: https://www.cio.com/article/228328/what-is-togaf-an-enterprise-architecture-methodology-for-business.html#:~:text=TOGAF%20is%20an%20enterprise%20architecture,objectives%20around%20enterprise%20software%20development.

GeeksforGeeks. (2021, June 30). *Benefit of using MVC*. Retrieved March 29, 2023, from GeeksforGeeks: https://www.geeksforgeeks.org/benefit-of-using-mvc/

GitLab. (2023). *What is DevOps?* Retrieved June 19, 2023, from GitLab: https://about.gitlab.com/topics/devops/

IBM. (n.d). *What is IT Infrastructure Library (ITIL)?* Retrieved June 20, 2023, from IBM: https://www.ibm.com/topics/it-infrastructure-library#:~:text=ITIL%20stands%20for%20Information%20Technology,and%20printed%20them%20for%20distribution.

Jafari, L. (2020, May 31). *What Are Non-Functional Requirements? Types and Examples*. Retrieved March 28, 2023, from WIN a TALENT: https://winatalent.com/blog/2020/05/what-are-non-functional-requirements-types-and-examples/#:~:text=System%20performance%20is%20the%20most,fall%20exclusively%20under%20these%20requirements.

Lazuardy, E. (2019, October 6). *Design or Architecture Pattern?* Retrieved March 29, 2023, from Medium: https://ezralazuardy.medium.com/design-or-architecture-pattern-5314ee71ed6c

LeanIX. (2023). *Zachman Framework*. Retrieved June 20, 2023, from LeanIX: https://www.leanix.net/en/wiki/ea/zachman-framework