ST10282051

PROG7311 - POE

24/06/2025

Contents

Performance Optimisation of the Agri-Energy Platform Prototype 2
Performance Testing Guidelines for Final Software Development (MVC & SSMS-Based Systems) 4
Methodology Recommendation and Breakdown 6
DevOps Recommendation 6
Framework Recommendation 6
Prototype Description – Agri-Energy Connect 8
References 10

Performance Optimisation of the Agri-Energy Platform Prototype

Performance optimisation is essential for ensuring the Agri-Energy platform runs efficiently, scales effectively, and provides a responsive user experience. Built using ASP.NET MVC and SQL Server Management Studio (SSMS), the prototype can benefit from improvements across multiple levels of the software architecture.

General Optimisation Principles involve several layers. At the design level, reducing redundant network calls and combining queries for example, fetching dashboard metrics in a single call can reduce latency. Implementing efficient algorithms and data structures with time complexities such as O(n) or better can prevent scalability issues. On the code level, asynchronous controller actions, minimal model binding, and front-end asset minification all reduce processing time and page load speed. Client-side optimisations include bundling scripts and compressing images, while server-side enhancements like proper indexing and optimized Entity Framework usage improve data retrieval. These efforts should be balanced with maintainability and tested continuously (Odhiambo, 2018).

Prototype Review

1. **Registration and Login Autofill Issues**
   The prototype currently pulls all saved emails from the user's password manager during registration and login, creating a cluttered interface. Figure 1.1 below shows the clutter issue clearly.

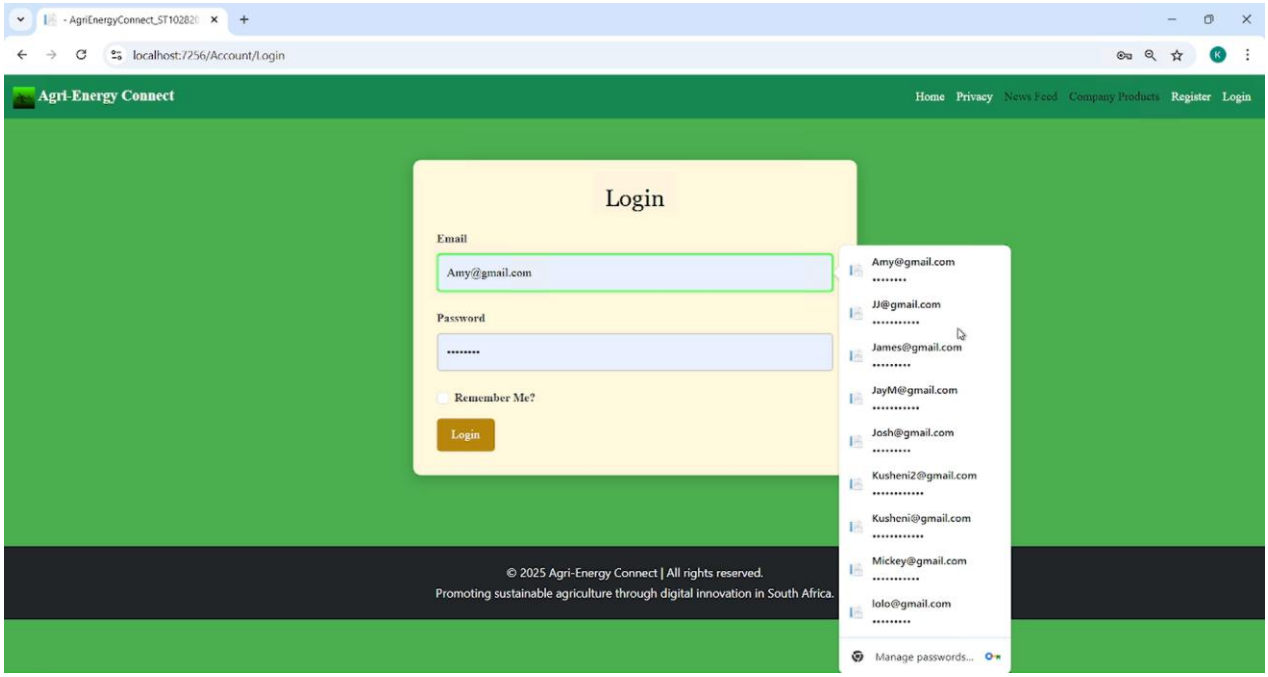

Figure 1.1

2. **User Dashboard Limitations**
   The dashboard lacks informative insights, which limits its utility for end users like farmers or energy coordinators. Figure 1.2 Admin Dashboard and Figure 1.3 Farmer Dashboard shows the lack of insight on both users end.

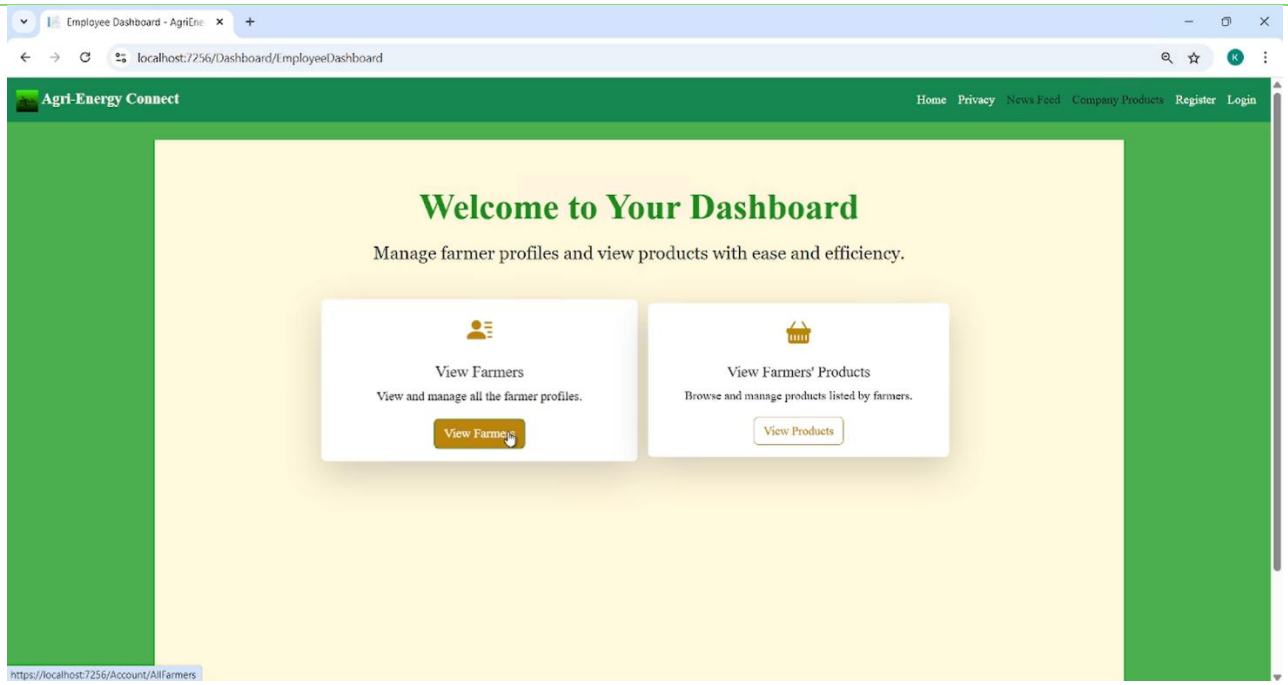

Figure 1.2

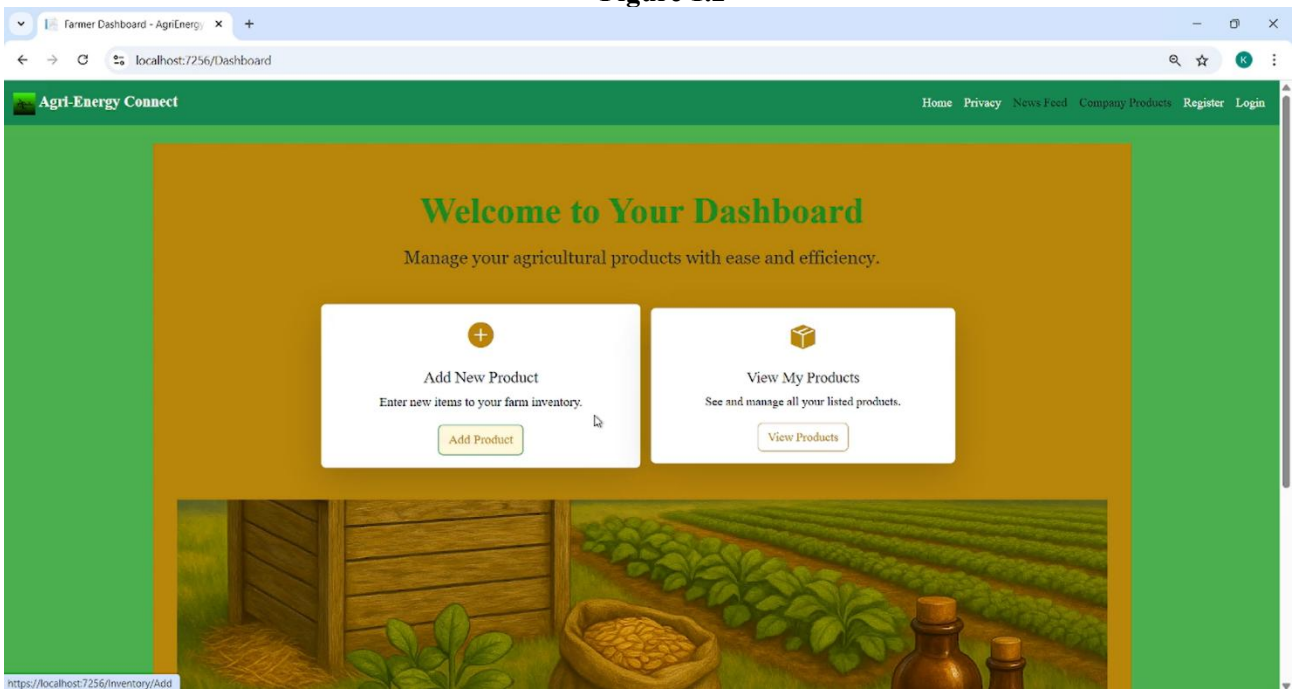

Figure 1.3

3. **Add Farmer (Admin Side)**
   Issue 1. Email Autofill Clutter: The email field shows all saved emails from the admin's browser, creating a cluttered and confusing experience.
   Issue 2. Admin Email Auto-Filled: When the page loads, the admin's own email is automatically inserted into the field, which should be blank by default.
   Impact:
   - Increases risk of submitting the wrong email
   - Slows down the process and causes frustration
   - May lead to data entry errors

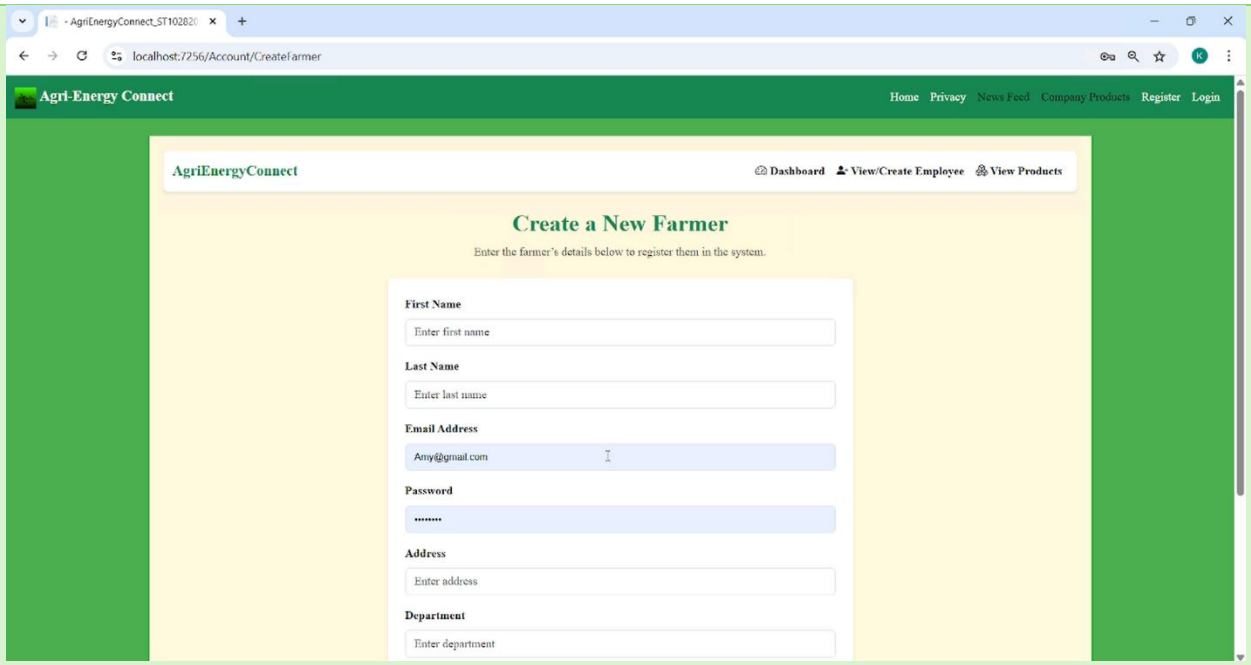

Figure 1.4

4. **Security Shortcomings**

The prototype includes basic security but lacks key protections. It uses role-based access to separate admin and farmer users during registration, which, if not optimized, could impact performance causing redundant queries, slow role checks, and increased load times.

Addressing these issues will improve both security and system performance under load.

5. **Response Time Delays**

Some backend operations are slow, especially when retrieving data from SSMS.

(Troelsen & Japikse, 2022).

Performance Testing Guidelines for Final Software Development (MVC & SSMS-Based Systems)

To ensure the final software version delivers reliable and efficient performance, it's essential to follow a structured performance testing strategy. This process evaluates how well the system operates under various loads and conditions not just in terms of functionality, but also responsiveness, scalability, and resource usage. Figure 1.5 demonstrates these tests (Bigelow, 2020).

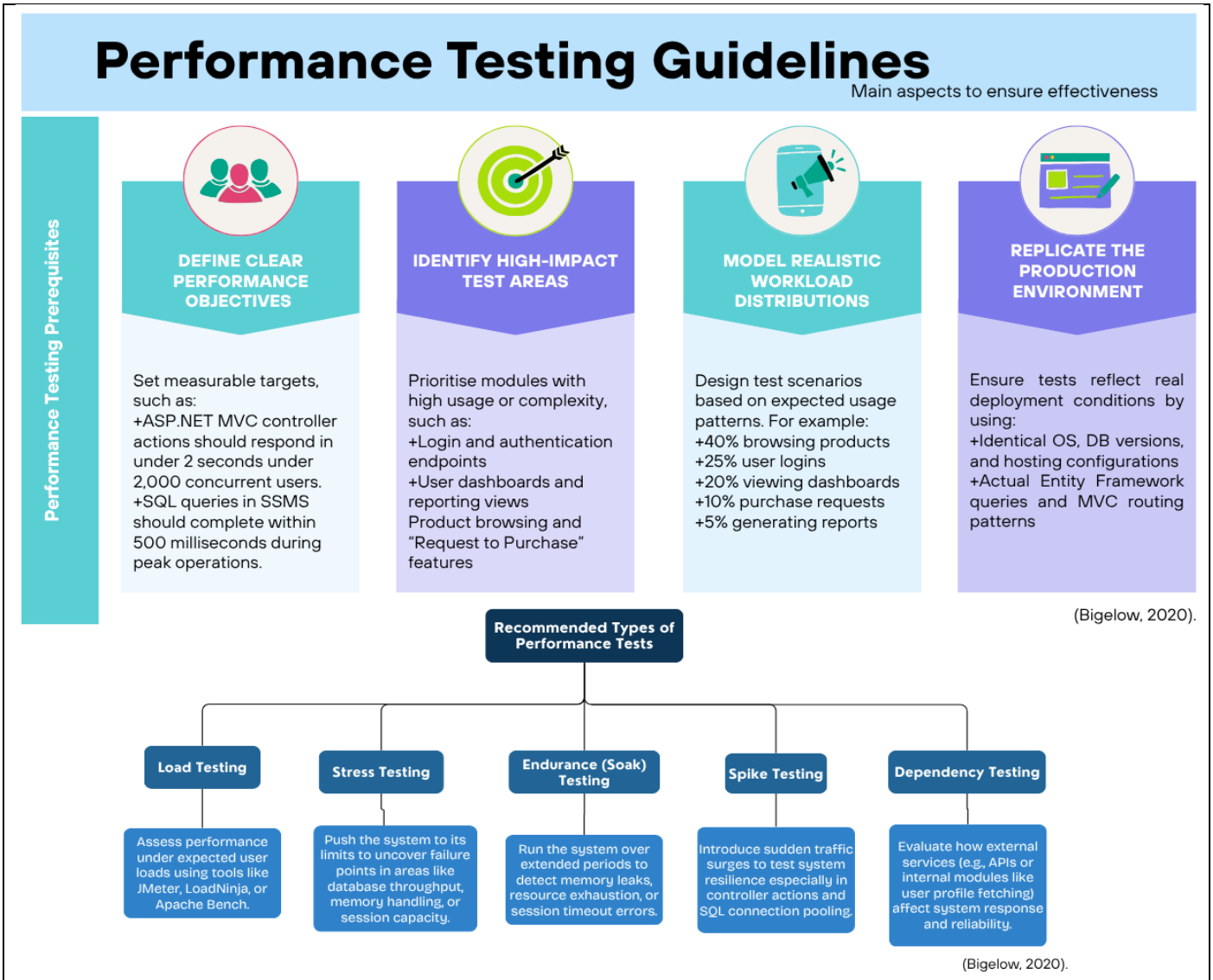

Figure 1.5

(Bigelow, 2020).

Methodology Recommendation and Breakdown

The Agile Incremental methodology is ideal for developing the Agri-Energy Connect platform, as it combines structured modular development with the adaptability of Agile practices.

The Incremental Model breaks the system into functional modules such as the Sustainable Farming Hub, Green Energy Marketplace, and Training Resources each built, tested, and delivered independently. Early increments are usable, allowing users to benefit from key features before the full system is completed.

Agile principles enhance this by introducing short development cycles (sprints), continuous testing, user feedback, and collaboration. Each sprint delivers a working piece of the platform, which is reviewed and refined based on stakeholder input. This encourages rapid improvement and the flexibility to adapt to evolving user needs.

The method promotes cross-functional collaboration between developers, farmers, and green energy stakeholders, ensuring relevance and real-world impact. Early delivery of core functionality like the "Request to Purchase" feature allows the platform to provide value from the start.

Given the platform's dynamic nature and the diverse needs of its users, this hybrid approach ensures scalable, user-driven development. It supports ongoing innovation in sustainable farming and energy, making Agile Incremental the most effective methodology for both initial implementation and future growth (Nikitin, 2024).

DevOps Recommendation

Implementing DevOps is strongly recommended for the Agri-Energy Connect platform, which is developed using MVC and SSMS and follows the Incremental Agile methodology. DevOps enhances efficiency by automating the build, test, and deployment processes, which accelerates delivery and aligns with Agile's goal of frequent, iterative releases.

By integrating Continuous Integration (CI) and Continuous Delivery (CD) pipelines, DevOps ensures each increment is stable and tested, improving quality and reducing bugs early. This reliability is critical for a platform dealing with sensitive agricultural and financial data.

DevOps also enhances collaboration between development and operations teams, breaking silos and enabling smoother deployments essential for a modular platform that must evolve with new features and green energy innovations.

Through Infrastructure as Code (IaC) and containerization, DevOps supports scalability and easier maintenance, allowing rapid adaptation to growing user needs. Automated monitoring tools also provide real-time feedback, which feeds directly into Agile planning and sprint retrospectives.

Overall, DevOps complements Agile by ensuring each sprint result is deployable, trackable, and high-quality, while reducing risks and enabling faster response to change. This makes DevOps a natural fit and essential component of the platform's long-term success (Opstree, n.d.).

Framework Recommendation

TOGAF + ITIL

1. **TOGAF (The Open Group Architecture Framework)**

   TOGAF offers a clear structure for designing, planning, implementing, and governing the platform's enterprise architecture. Since Agri-Energy Connect is modular (e.g., farming hub, green marketplace, training modules), TOGAF's Architecture Development Method (ADM) helps align the platform's evolving business goals with technical development. TOGAF supports scalability, integration, and long-term sustainability critical for rural deployment and future expansion (Dey, 2017).

2. ITIL (Information Technology Infrastructure Library)

ITIL focuses on IT service management and aligns well with platform operations post-deployment. It ensures efficient incident handling, user support, availability management, and continuous service improvement vital for a platform supporting multiple user roles (farmers, admins, partners) with real-time services and data (Dey, 2017).

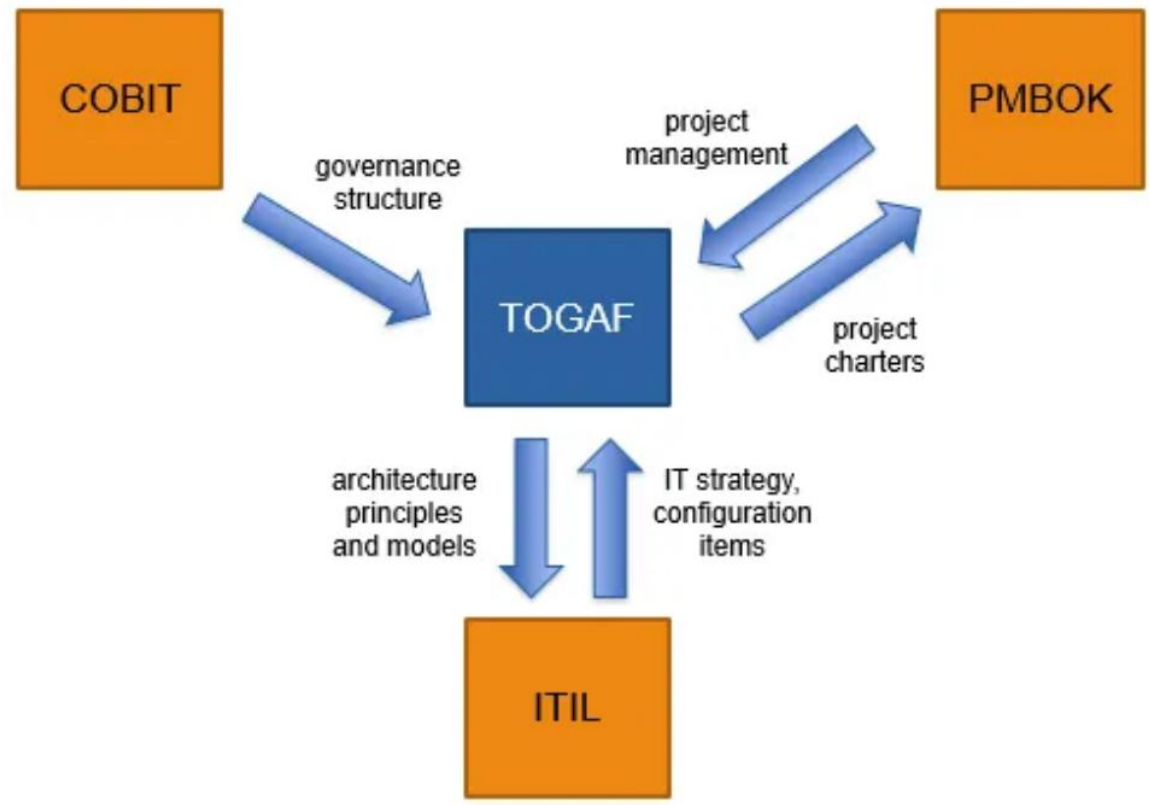

Figure 4.1

(Dey, 2017).

Prototype Description – Agri-Energy Connect

The Agri-Energy Connect prototype is a web-based platform developed using the ASP.NET MVC framework, integrated with SQL Server Management Studio (SSMS) for structured and secure data management. The solution is designed to serve as a central hub for farmers, green energy providers, and sustainability experts to collaborate, share resources, and engage in project-based initiatives.

Technical Architecture Overview	**ASP.NET MVC Framework**	The prototype uses the Model-View-Controller architecture to separate concerns, ensuring clean code structure, easier maintenance, and rapid feature delivery. This supports the Incremental Agile methodology by allowing independent development and testing of platform modules like the Sustainable Farming Hub, Green Energy Marketplace, and Training Resources.
	SQL Server Management Studio (SSMS)	The backend database supports relational data storage and indexing for efficient queries. It handles user registration, forum discussions, product listings, course content, and collaboration project data ensuring data consistency and scalability as platform adoption grows.
	Entity Framework (Code First)	Used as the Object-Relational Mapping (ORM) tool to simplify data access, reduce boilerplate SQL, and accelerate development cycles. This enabled the team to quickly implement business logic changes based on feedback during iterations.
	Responsive Front-End & Optimizations	HTML5, CSS, and JavaScript (with bundling and minification) ensure the platform is mobile-friendly and loads quickly even in low-bandwidth rural areas directly improving accessibility for farmers and users in remote communities.
	Security & Access Control	Authentication (via ASP.NET Identity) restricts access to sensitive areas such as project collaboration tools, ensuring data privacy and trust, which are essential for platform credibility.

(Troelsen & Japikse, 2022).

GitHub Link - https://github.com/VCPTA/bca3-prog7311-portfolio-of-evidence-poe-ST10282051.git

References

Bigelow, S. J., 2020. *Software performance testing requirements and prerequisites.* [Online]
Available at: https://www.techtarget.com/searchsoftwarequality/answer/Required-prerequisites-for-performance-testing
[Accessed 20 June 2025].

Dey, A., 2017. *Enterprise Architecture & Frameworks -TOGAF, ITIL, COBIT, PMBOK.* [Online]
Available at: https://medium.com/@arindey/enterprise-architecture-frameworks-togaf-itil-cobit-pmbok-88a3e4dca82c
[Accessed 22 June 2025].

Nikitin, V., 2024. *Software development methodologies: types, selection criteria & stats.* [Online]
Available at: https://www.itransition.com/software-development/methodologies
[Accessed 19 June 2025].

Odhiambo, D., 2018. *Performance Optimization in Software Development.* [Online]
Available at: https://medium.com/the-andela-way/performance-optimization-in-software-development-ae7952ab885e
[Accessed 20 June 2025].

Opstree, n.d. *DevOps Explained: What It Is, How It Works, and Why It Matters.* [Online]
Available at: https://opstree.com/blog/2025/05/15/devops-how-it-works-and-why-it-matters/
[Accessed 21 June 2025].

Troelsen, A. & Japikse, P., 2022. *Pro C# 10 with .NET 6.* 11th ed. New York: APress Media.