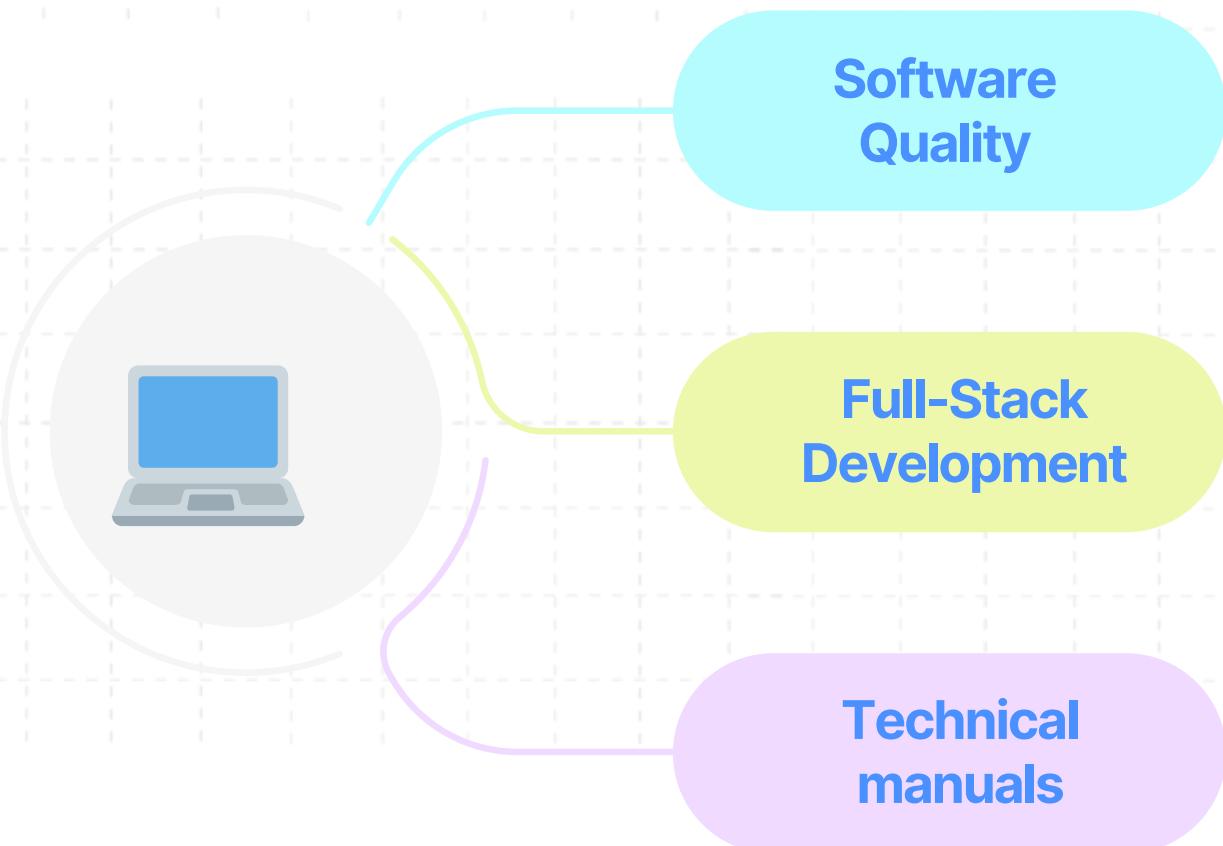


15 Coded Modules, Research and Technical Documents

A detailed study



Code and processes

15 modules developed

3 technical documents written

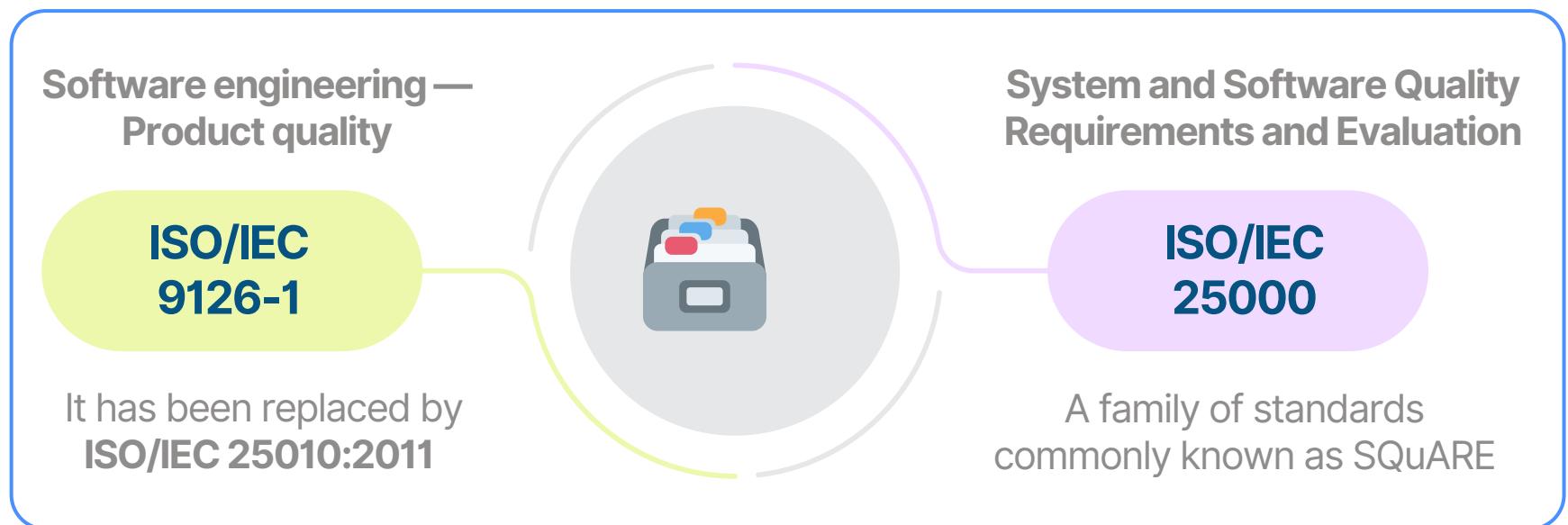


Mercedes Paz
Computer Systems Engineer



1. Software Quality as a Focus

Everywhere I could and where viable, I implemented improvements recommended by the following ISO/IEC standards. This scoped both full-stack development and a focus on UI/UX.



Software quality measurements can be split into **three parts**:

- process quality
- product quality** which includes internal and external properties
- quality in use, which is the effect of the software.

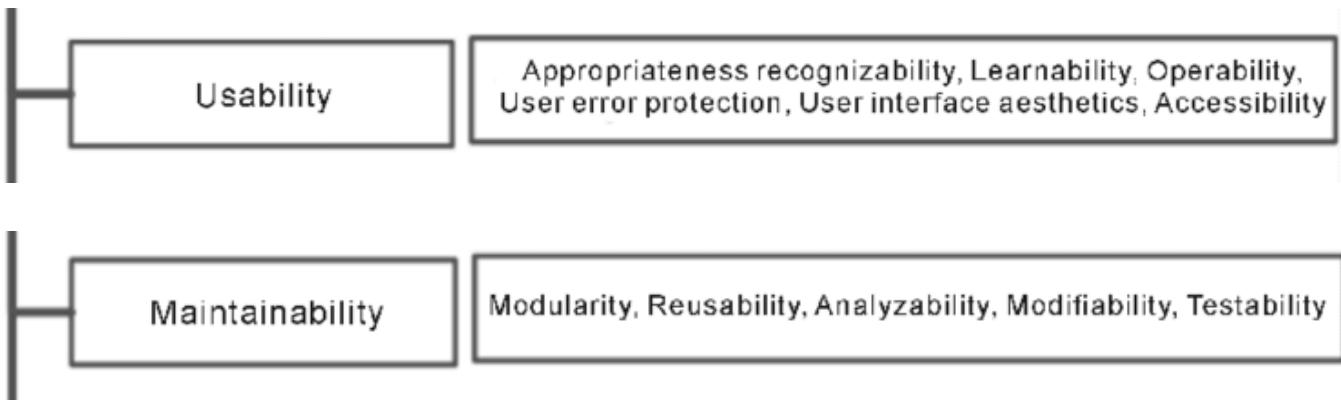


Mercedes Paz
Computer Systems Engineer

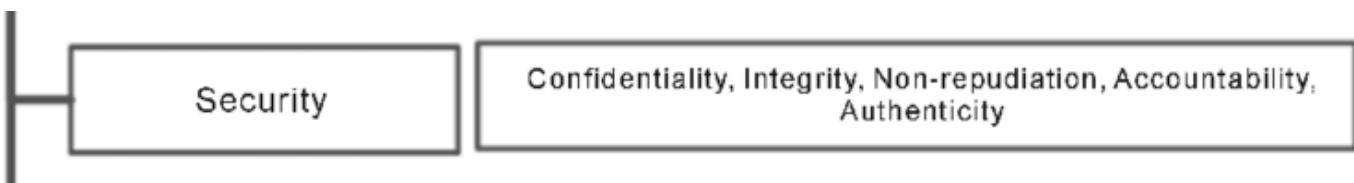


ISO/IEC 25010:2011

The maintainability and usability characteristics improvements were within my reach and scope. I focused on **implementing coding best practices** and **providing recommendations with action plans** for the overall software process.



- *Implemented within the codebase*



- *Identified risks and provided recommendations in the technical documents written.*

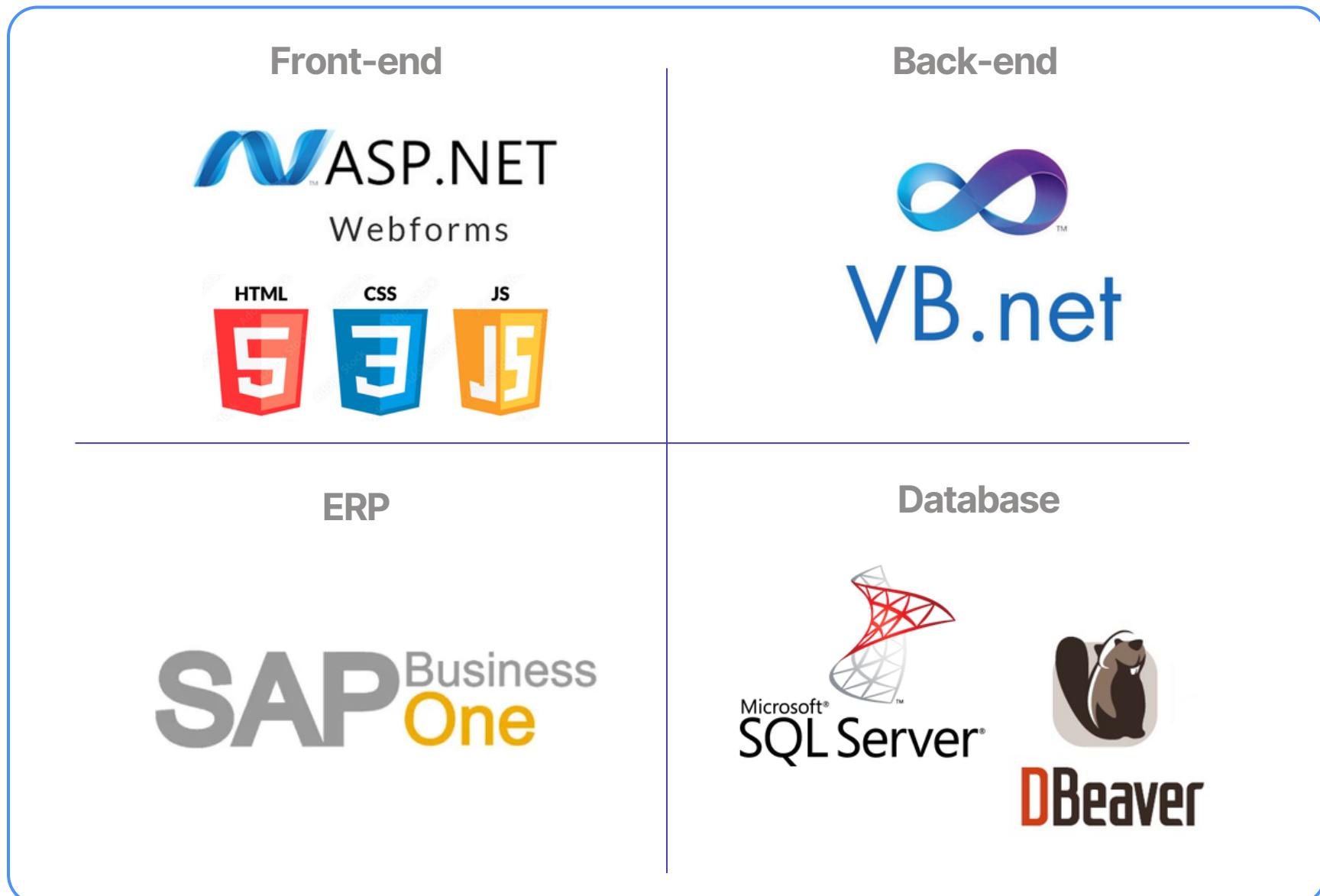


Mercedes Paz
Computer Systems Engineer



Tech Stack Used

Although the company used a tech stack I was unfamiliar with, I was able to quickly become productive due to **my understanding of development fundamentals and practices**, which apply to any programming language.

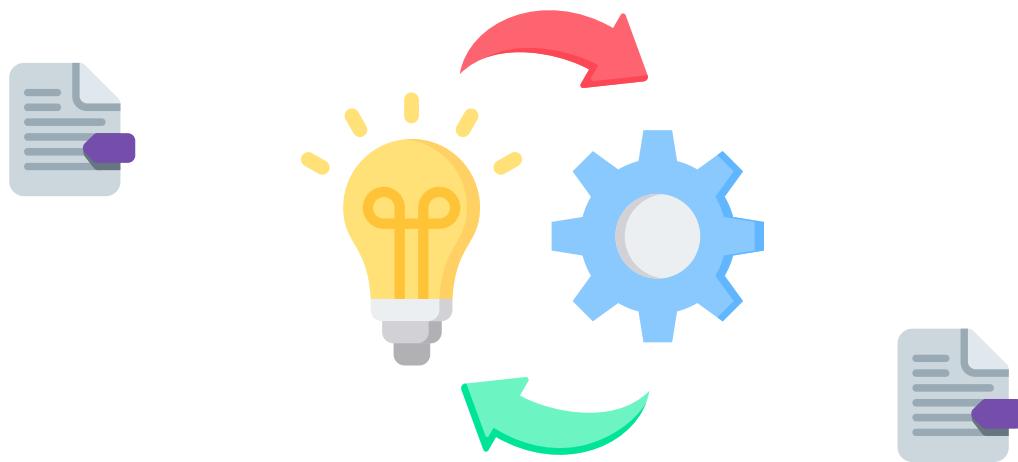


Mercedes Paz
Computer Systems Engineer



Maximizing Tech Stack Value

I saw working with a legacy system as an **opportunity to refine my refactoring, debugging and problem-solving skills**, all while still providing value in every developed module.



- ✓ I studied the stacks' official documentation **to understand its capabilities and limitations**.

- ✓ **Identified pain points** in the codebase, understood how they **impacted system performance**, and determined if my proposed solution was **viable** in its current environment.



Mercedes Paz
Computer Systems Engineer



Maximizing Tech Stack Value

Combined with a modern approach on **best practices and processes**, I was able to **identify bottlenecks** and provided solutions specific to the stack to ensure performance.



If **viable**, I implemented **fixes tailored to the existing architecture**, bringing out the best in the stack



This provided the stage for **adaptability, growth and problem-solving**.



Mercedes Paz
Computer Systems Engineer



Maintainability

My biggest concern during development was to make sure every module was **easily maintainable, easy to read and easy to understand** without having to play detective to understand it.



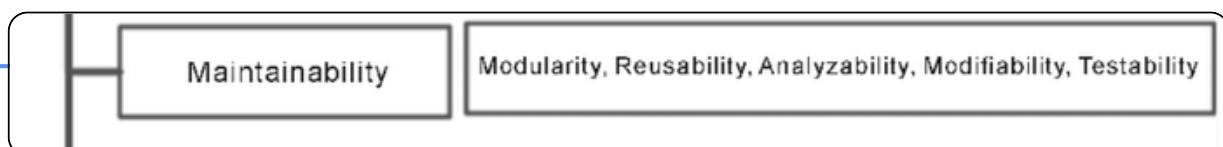
Identified **coding smells** and **applied refactoring techniques**, allowing the codebase to adhere to coding standards whenever possible and **reduce maintenance overhead** (+ technical debt!)



Ensured **documentation as code**, as well as comments that provided **brief context on the module's purpose** and how it related to the **overall project**.



Used coding best practices, like **SOLID principles, KISS, YAGNI, DRY**, among others, to **decrease mental load during maintenance**, if needed in the future.



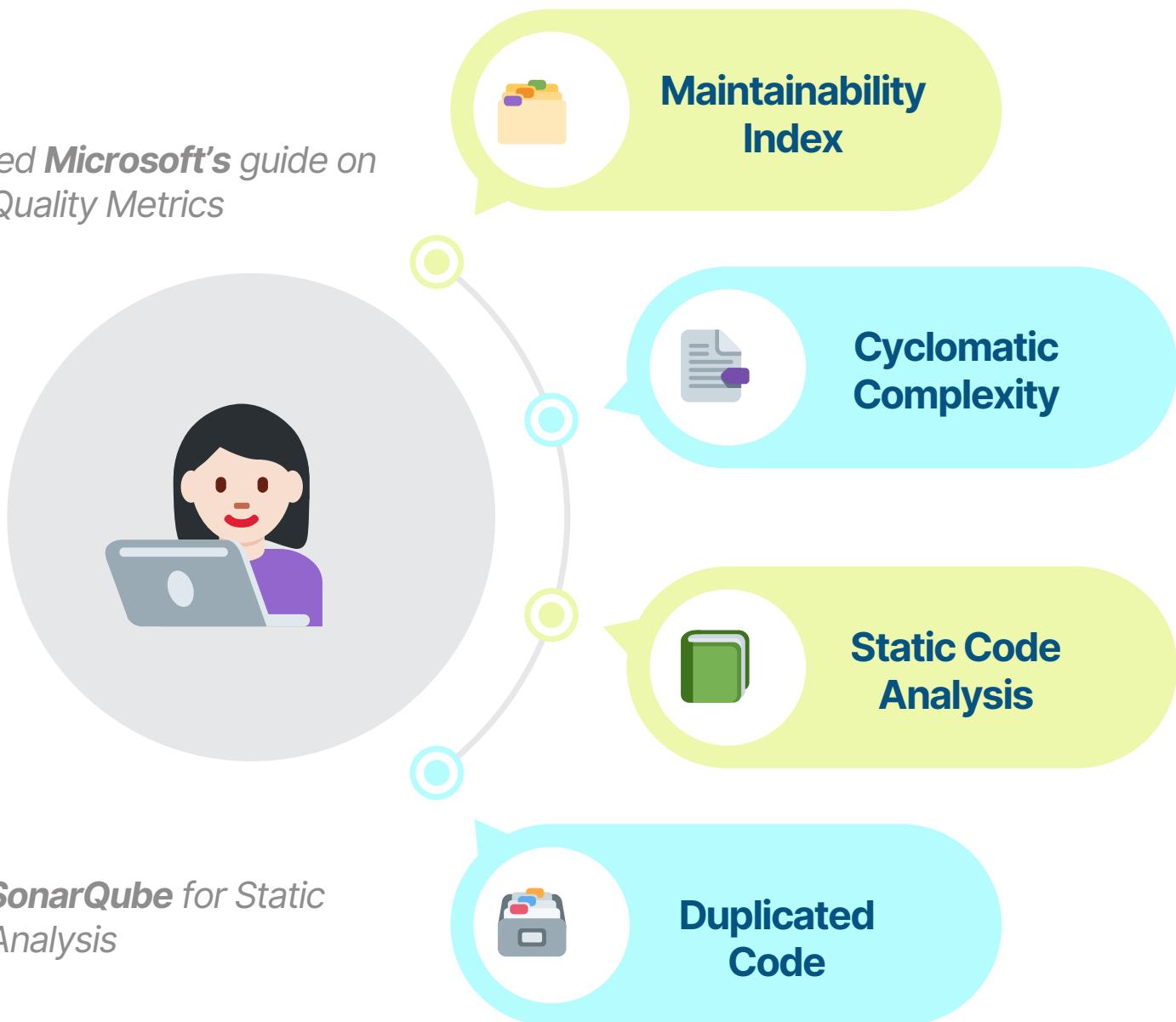
Mercedes Paz
Computer Systems Engineer



Maintainability

I used **metrics** to ensure the maintainability focus was being met. This provided a bigger understanding of pain points and how I could improve the codebase **while keeping the code up to standards**.

Followed **Microsoft's** guide on
Code Quality Metrics



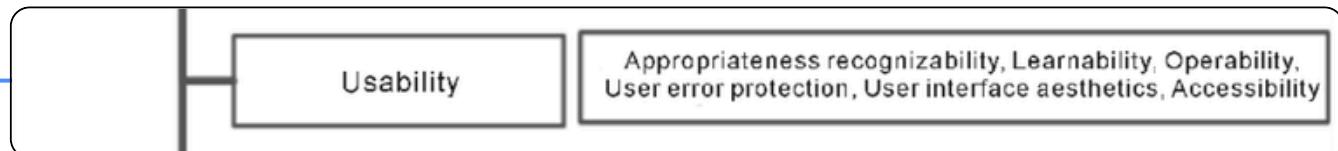
Mercedes Paz
Computer Systems Engineer



Usability

I also set out to maintain a balance between code quality and the user experience.

- ✓ Created **wireframes** prior to the front-end, identifying bottlenecks that would hinder the user. This also reduced front-end development time during revisions and testing.
- ✓ Created **user flows** to ensure users could complete their tasks within the WebApp in a **seamless workflow** focused on prioritizing their experience and productivity.
- ✓ Designed all the modules developed following **UI best practices**.



Mercedes Paz
Computer Systems Engineer



Security

The security characteristic was mostly implemented as **recommendations in the technical documents written**. Like the previous characteristics, this was done after a careful analysis procedure.



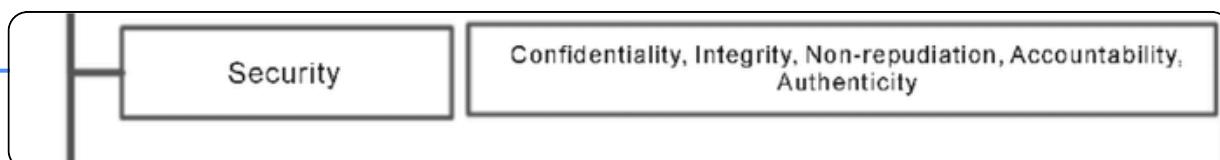
Proposed routine checkups for vulnerabilities, such as cyberattacks, geographical disasters, human mistakes, and system failures.



Proposed the lecture of the General Data Protection Regulation (**GDPR**) as well as **ISO/IEC 270001:2022**



Suggested **steps to improve** the Data recovery and restoration procedure.



Mercedes Paz
Computer Systems Engineer



2. Modules Developed

Additional to the software quality implemented, I designed and developed 15 modules for the administrative user's WebApp within 6 months.

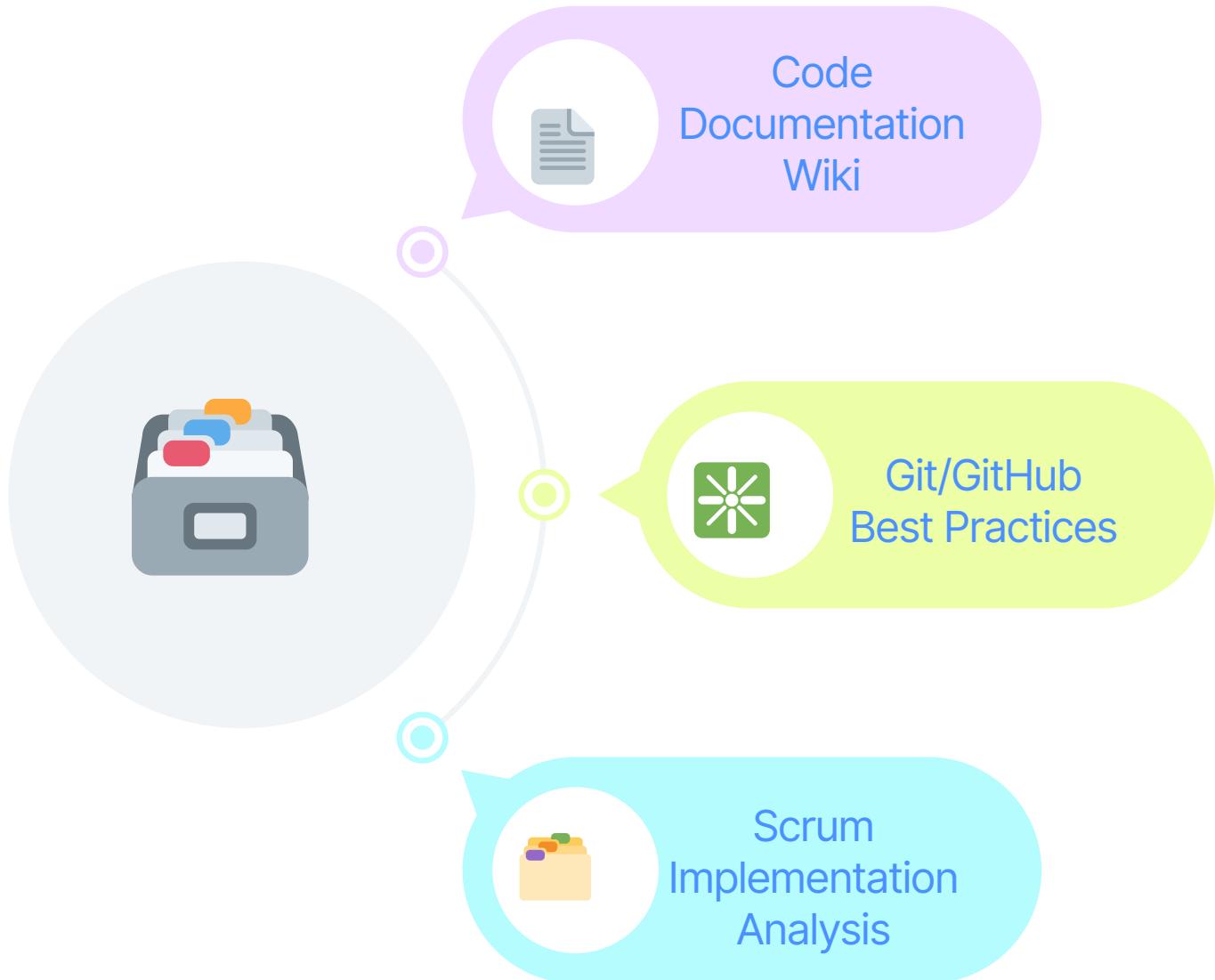
- Implemented full-stack modules with **CRUD** operation capabilities.
- Created a **SAPB1 emulator** that allowed users the visualization of data within the WebApp, without the need to use SAPB1.
This included tax, invoice, payment, credit note, debit note, order sales and collection plans information.
- Created **wireframes and user flows** to maximize the user experience and development time.



Mercedes Paz
Computer Systems Engineer



3. Technical documents

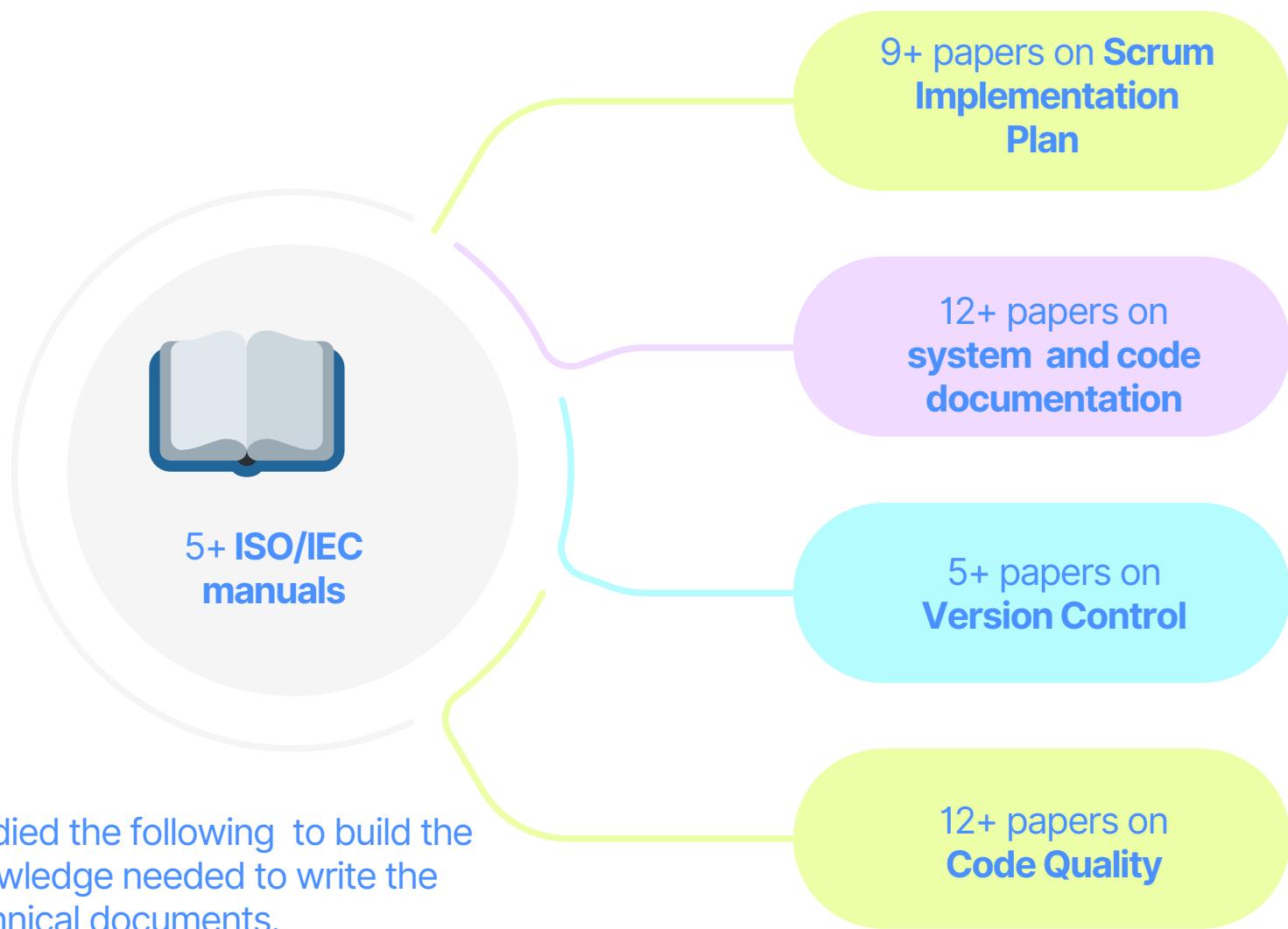


Mercedes Paz
Computer Systems Engineer



Research

To ensure the correct information This required extensive research, which played a key role in **ensuring a successful analysis process.**



Mercedes Paz
Computer Systems Engineer

Summary

Each technical document was created after a thorough analysis of the department's workflow and limitations.

The documentation wiki included a **50% coverage of the developed modules**, including **UML** diagrams for additional clarification on the **database design**.

The Git&GitHub document included an analysis of the **current practices** and a guideline of **personalized workflow** to further facilitate the software development life cycle.

Included targeted recommendations and educational material for a future **Scrum implementation**.



Mercedes Paz
Computer Systems Engineer



Tech Stack

All technical documents were written in Astro Starlight due to the framework's core characteristics of **optimized fast, content-driven** websites. It is also written mostly in markdown.

The Git/GitHub and Scrum implementation documents were deployed in **Vercel for easy access** throughout devices. The technical documentation is entirely local due to sensitive information.



Mercedes Paz
Computer Systems Engineer



Documentation Wiki

Technical Documents * Technical Documents

The WebApp's large codebase greatly benefited from a **centralized unit of knowledge across all developers**. I set in motion a process to strengthen this **pain point** using **Astro Starlight as the main tool**.

- Covered **50% of developed modules**, with a focus on **architectural design, technical design, code documentation**.
- Created **UML diagrams for each responsible schema** per module to bring **further clarification** of the database design.
- Included **targeted recommendations** for maintenance and security protocols based on current processes.
- Provided all the **relevant information about maintainability, usability and security** studied through research to **facilitate intellectual property** of all members over development.



Mercedes Paz
Computer Systems Engineer



Agile Scrum Proposal

Due to lack of staff and lack of available time, the Scrum implementation could not be executed. Following this, I wrote this **technical document as a digestible and easy to read source of information for future reference.**

This will allow the department to all **gain the same intellectual property** without the overwhelm that come with investigating a new methodology from the ground up.

- ✓ Presented the most relevant information from the **Agile Manifesto, Essential Scrum and Google's Agile Management Course**
- ✓ Presented **case studies** from companies in a similar position as the company and how they managed to implement the Scrum methodology.
- ✓ Presented a **step by step plan** that allows progressive implementation to **avoid setbacks** due to sudden changes in management.



Mercedes Paz
Computer Systems Engineer



Agile Scrum Proposal

EIS.AGILE

Search Ctrl K

Ágil

- Metodología agil
- Marcos de trabajo

Scrum

- Roles
- Artefactos
- Ceremonias

Implementación

- Implementación

Metodología agil

La **metodología agil** no es solamente el manejo de proyectos. Es un enfoque global y filosofía. Bajo esta metodología, existen diversos marcos de trabajo, como Kanban, Scrum, XP, entre otros. Estos marcos de trabajo se explican más adelante.

Ágil

El término **ágil** se refiere a moverse rápidamente y eficientemente, teniendo flexibilidad y la voluntad de cambiar y adaptarse a las circunstancias actuales. Esto generalmente se hace utilizando un enfoque iterativo.

On this page

- Overview
- Breve historia
- Valores fundamentales
- Individuos e interacciones sobre procesos y herramientas
- Software funcional sobre documentación comprensiva
- Colaboración con el cliente sobre negociación contractual
- Responder al cambio sobre seguir un plan
- Principios ágiles
- Entrega de valor
- Colaboración empresarial
- Dinámicas de equipo y cultura
- Retrospectivas y aprendizaje continuo
- Metodologías tradicionales vs ágiles

Implementación

La implementación de Scrum en el departamento queda a completo juicio del mismo debido a la falta de espacio proporcionado para su implementación. Sin embargo, se recomienda la implementación de los siguientes puntos:

- Implementación de backlog detallado
- Implementación de iteraciones
- Implementación de retrospectivas (iteración o entrega de sistema)
- Implementación de requisitos de sistema mediante épicas e historias de usuario
- Seguimiento de proyecto en Asana y prácticas de Scrum

Casos de estudio

Los siguientes son casos de estudio e implementaciones de Scrum dentro de



Mercedes Paz
Computer Systems Engineer



Git/GitHub Best Practices

- ✓ Proposed a new team workflow by employing Issues to track anything relevant to the project. This will help the team gain a 100% centralization of error reports to avoid losing critical tasks due to human mistakes.
- ✓ Included step-by-step guide on the implementation of issues templates, which allows the team to write a full report without friction.
- ✓ Included step-by-step guide on the implementation of issues templates, which allows the team to write a full report without friction.
- ✓ Included step-by-step branch use best practices such as branch strategies and branch protection.
- ✓ Provided commits best practices strategies, such as conventional commits for easier tracking.



Mercedes Paz
Computer Systems Engineer



Git/GitHub Best Practices

The screenshot displays two pages from a GitHub-style documentation site:

- Top Page (Introducción):** Describes GitHub as a collaborative development platform using Git version control. It mentions Visual Studio Community 2019 for managing Git commands via VSCommunity's minimalist graphical interface.
- Bottom Page (Reportes):** Provides a JSON schema for bug reports, including fields for name, description, labels, body, attributes, and validations.

On this page:

- Introducción
- Reportes
- Nuevo sistema



Mercedes Paz
Computer Systems Engineer

