Need For Engineering Documentation

Table Of Contents

Problem Statement

Hyland's Technology organization is facing significant challenges due to the lack of standardized practices for engineering documentation. Engineers currently rely on ad hoc approaches, resulting in inconsistent and fragmented documentation that is difficult to locate and often varies in format, structure, and detail. This has created knowledge silos, where critical information resides with individuals rather than being accessible across the organization. Consequently, new hires struggle to onboard efficiently, and cross-functional teams, including solution engineers, services, and tech support, face challenges leveraging engineering outputs. The absence of automation and integration tools for documentation further exacerbates these issues, requiring manual effort and introducing inconsistencies. Additionally, poorly documented or undocumented systems slow developer productivity, as engineers must spend significant time interpreting existing code and systems. These challenges increase technical debt, hinder scalability, and slow time-to-market for new solutions. Addressing these issues by implementing clear standards, tools, and workflows for documentation would foster better collaboration, enhance developer productivity, and improve system maintainability.

Business Values in Creating Engineering Documentation

Investing in standardized engineering documentation early as Technology moves into our new value/work streams enhances operational efficiency and scalability. By addressing knowledge silos and improving accessibility, documentation reduces dependencies, accelerates onboarding, and supports collaboration across teams. It minimizes technical debt, simplifies maintenance, and fosters innovation by providing clarity around code and systems. Automating updates ensures documentation evolves with the code, creating a sustainable process that supports long-term growth. Additionally, high-quality documentation strengthens relationships with customers and partners while signaling a commitment to excellence, reinforcing your organization's ability to adapt and succeed.

Key Objectives Of Creating Engineering Documentation

- **Accelerate Development and Delivery:** Streamline workflows to reduce time spent interpreting code and systems, enabling faster delivery of features and products.
- **Enhance Knowledge Sharing and Collaboration:** Create accessible documentation to eliminate knowledge silos and improve cross-functional collaboration among engineering, solution engineers, services, and support teams.
- **Improve Onboarding Efficiency:** Provide clear and consistent documentation to reduce onboarding time for new team members, ensuring they become productive faster.
- **Automate Documentation Maintenance:** Integrate documentation generation and updates into CI/CD pipelines to ensure it evolves alongside code and remains reliable.
- **Support Long-Term Scalability:** Build processes and practices that grow with the organization, enabling sustained operational excellence and adaptability.

Expected Outcome To Achieve

- **Accelerate Development and Delivery**
  - **Outcome:** Decrease time spent on understanding existing code and systems.
  - **Metric:** Average time to complete new feature development decreases by 20% within six months.
- **Enhance Knowledge Sharing and Collaboration**
  - **Outcome:** Improve accessibility and usability of documentation across teams.
  - **Metric:** 90% of team members report finding relevant documentation in under five minutes during quarterly surveys.
- **Improve Onboarding Efficiency**
  - **Outcome:** Reduce time-to-productivity for new hires.
  - **Metric:** Average onboarding time decreases by 30% within the first year of implementation.
- **Automate Documentation Maintenance**
  - **Outcome:** Ensure documentation evolves with code changes and reduces manual effort.
  - **Metric:** 90% of repositories integrate automated documentation generation and updates within CI/CD pipelines within the first year.
- **Support Long-Term Scalability**
  - **Outcome:** Establish scalable processes that adapt as the organization grows.
  - **Metric:** 100% compliance with documentation standards across repositories within a year; measurable efficiency gains in team operations as the organization scales (e.g., onboarding, project throughput).

ReadMe Documents Learning Path:

General Path: Apply to all types of Tech Stacks

1. Read through parent readme document to understand the need of Engineering Documentation and Code Standards to successfully develop a project in line with Modern Development Practices for documentation.

2. Read through Guiding Principles For Engineering Documentation Standards to understand the principles which we have used to develop the projects present the repository.

3. Read through OpenAPI - An Introduction to understand key elements of OpenAPI.

4. Read through OpenAPI Documentation Standards - understanding of which is very important for creating OpenAPI documentations.

Tech Stack Specific Path:

1. Tech Stack: .NET 8
    - Read through Documentation Standards For Code Comments
    - Read through <a DotNet8ProjectInstallationInstruction.md">Project Information and Installation Instructions

🟥 **The Tech Stack section is still under process and will be updated as other tech stack projects are added**

Namespace DotNetOpenAPI.Controllers

Classes

EmployeesController

This controller is responsible for handling requests related to employees CRUD operations. Will provide the necessary http response codes and data. The controller will interact with the ICURDActions interface to get the data.