**Development Process Document**

**Introduction**

This document outlines the standard Development Process adopted and fully implemented for the eServices Replatforming project at AltaGas. The approach aligns with enterprise DevSecOps best practices using Azure DevOps.

**Development Lifecycle Overview**

The development lifecycle, fully integrated into project operations, includes:

* Requirements Gathering
* Work Item Creation
* Development (Code & Configurations)
* Code Review & Pull Request (PR)
* CI/CD Pipeline Execution
* UAT & Production Deployment (Planned)

**Work Item Management (Azure Boards)**

* **Hierarchy Used:** Epic > Feature > User Story > Task > Bug
* **Linkage Implemented:** All code branches and PRs are linked to relevant work items.
* **Sprint Planning Conducted:** Two-week iterations with defined goals and deliverables.
* **Tracking:** Azure Boards dashboards are actively used for burndown, velocity, and progress tracking.

**Source Code Management (Azure Repos)**

* **Branching Strategy Adopted:** Gitflow (main, develop, feature/*, release/*, hotfix/\*)
* **Development Branch Protection:** The develop branch is protected. No direct commits are allowed.
* **Feature Branches:** Created by developers from develop for individual user stories.
* **Release Branches:**
  + release/qa: Used for QA environment deployments.
  + release/uat: Used for UAT environment deployments.
* **Branch Protection Enforced:** No direct commits to main or develop.
* **Pull Requests:** Required for all merges; minimum 1 reviewer; automated validation of build, lint, tests, and SAST is enforced.
* **Code Reviews:**
  + **Backend (.NET):** Ensure adherence to C# coding standards, architectural patterns (e.g., clean architecture), and secure coding practices. Code is reviewed for proper use of dependency injection, exception handling, and unit testing coverage.
  + **Frontend (ReactJS):** Code is reviewed for JSX best practices, component modularity, state management (Redux/Context), accessibility (a11y), and adherence to styling conventions. Linting and unit test coverage are checked.
* **Repository Naming Convention:**
  + WGLH-ESVC-FRONTEND: Front-end application code for eServices Replatforming
  + WGLH-ESVC-BACKEND: Back-end services and APIs for eServices Replatforming

**Code Quality & Security**

* **Tools Used:** SonarQube (SAST), GitHub Advanced Security
* **Policies Implemented:**
  + PR validation is mandatory
  + Code scanning and dependency scanning are enabled
  + Secret scanning and push protection are enforced

**CI/CD Pipeline (Azure Pipelines)**

* **Pipeline Names:**
  + WGLH-ESVC-FRONTEND-CICD: CI/CD for the frontend build and deployment
  + WGLH-ESVC-BACKEND-CICD: CI/CD for the backend build and deployment
  + WGLH-ESVC-SECURITY-SCANNING- For sonar scanning and Advanced Github Security
* **Pipeline Triggers:**
  + Triggered on PRs and commits
* **Stages Covered:**
  + Build → Test → Scan → Deploy
* **Environment Strategy:** Dev and QA environments are active and fully operational. UAT and PROD environments are planned and will be integrated into the pipeline in upcoming phases.
* **Deployment Gates:** Compliance and security validations are implemented before production deployments

**7. Secrets & Configuration Management**

* **Tool:** Azure Key Vault
* **Access Control Enforced:** Role-Based Access Control (RBAC)
* **Integration:** Secrets are securely injected into pipelines and runtime apps

**9. Tagging & Versioning**

* **QA Deployment Tags:**
  + Format: qa-<Major>.<Minor>.<Patch> (e.g., qa-1.0.0)
  + Semantic versioning where:
    - **Major**: Major functional improvements or architecture changes
    - **Minor**: New features and enhancements
    - **Patch**: Bug fixes or small changes
* **Development & Feature Branch Tags:** Used as per application development requirements
* **Best Practices Followed:** Tags are created only by release managers and are protected in Azure Repos

**Documentation & Governance**

* All repositories include README, .gitignore, and license files
* PR templates and issue tracking are enabled
* Changelog is maintained for every release

**Summary**

The development process has been fully implemented and standardized across the eServices Replatforming project. Dev and QA environments are fully operational, while UAT and PROD are planned for upcoming implementation. This ensures secure, high-quality, and compliant software delivery while promoting agility, transparency, and operational excellence.