|  |
| --- |
| GitHub Governance for Product Engineering Team Organization |
|  |
| Version 1.0 - 10 / 20 / 2023 |

Contents

[Introduction 3](#_Toc149203434)

[Repository Management 3](#_Toc149203435)

[Roles and Responsibilities 5](#_Toc149203436)

[Security and Data Protection 5](#_Toc149203437)

[Best Practices 6](#_Toc149203438)

[Runners Execution 6](#_Toc149203439)

[Conclusion 7](#_Toc149203440)

# Introduction

This document outlines the policies and procedures that govern the use of GitHub by DA Technology Services - Product Engineering Team. Its purpose is to foster effective collaboration, ensure compliance with security policies, and efficiently manage resources on GitHub.

**Purpose**

The purpose of this governance document is to:

* Establish guidelines for source code collaboration and version control.
* Define roles and responsibilities related to GitHub usage.
* Implement security measures and data protection protocols.
* Promote best practices in software development.

# Repository Management

**Repository Creation**

Members of the Product Engineering Team may create new repositories as required, provided that they adhere to a clear and descriptive naming convention

Example: [Project-Name]-[Repo Type or Service]

Members of the Product Engineering Team are strongly encouraged to create new repositories using approved boilerplates whenever feasible.

Approved boilerplates are pre-defined project templates or structures that have undergone scrutiny and received the endorsement of a dedicated committee for the sake of ensuring consistency, adherence to best practices, and compliance with governance standards.

Utilizing approved boilerplates serves to guarantee uniformity, minimize redundant efforts, and encourage adherence to established best practices.

When an approved boilerplate is unavailable for a particular project, members should request a review from the Backend or Frontend Committees to create a new repository from scratch.

Repositories must incorporate comprehensive documentation that clearly outlines their purpose, structure, and contribution guidelines.

**Repository Access**

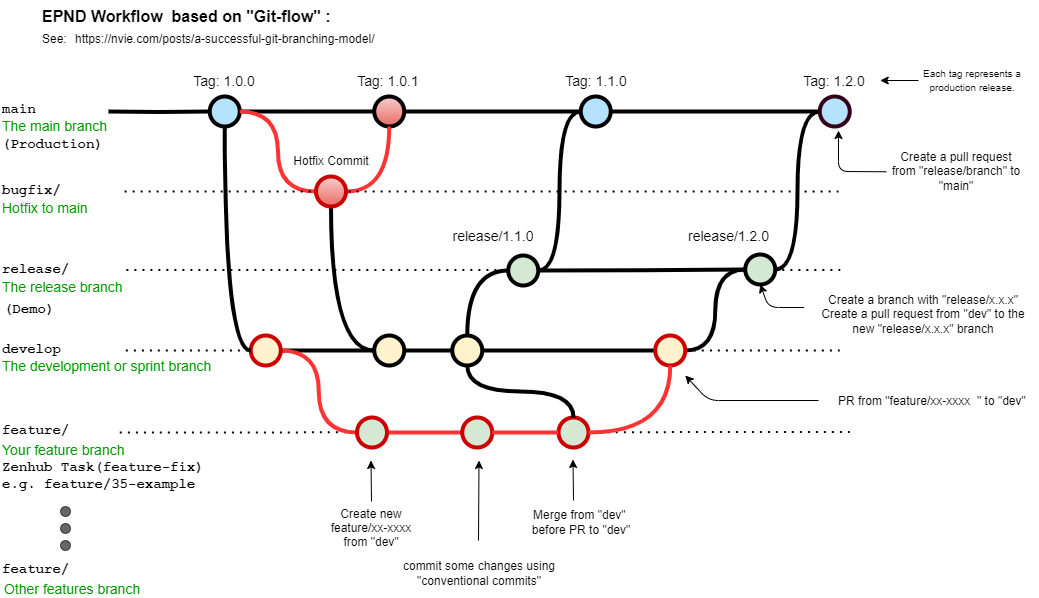
Access to repositories should follow the "need-to-know" principle. Members should only have access to repositories relevant to their roles and responsibilities.

Just Repository Owner and administrators could delete repositories.

Permission management will be done through the GitHub organization and will be periodically reviewed for consistency.

**Branching and Workflow**

Repositories should follow a branching workflow, such as GitFlow or GitHub Flow, to maintain proper version control and facilitate collaboration.



Pull Requests must be used for code review and approval before merging changes into the dev and main branch.

# Roles and Responsibilities

**Repository Owner**

The repository owner is responsible for its maintenance, including reviewing and approving pull requests and managing repository-related issues.

Validate developers maintain the code generated in the cloud.

**Collaborators**

Collaborators are team members who can make contributions to the repository but are not responsible for its management.

**Organization Administrators - Owners**

Organization administrators have the ability to manage permissions and resources within the GitHub organization.

The following individuals are designated as Organization Administrators with the responsibility for repository management:

1. Richard Saavedra
2. Ignacio Henriquez
3. Erick Jamet
4. Bruno Rodriguez.
5. Amandeep Singh.

# Security and Data Protection

Security policies of the organization must be followed when working with sensitive data.

* All repositories must be created as Private.
* Repositories containing confidential information must be marked as private and restricted to necessary personnel only.
* Two-Factor Authentication (2FA) must be implemented for all GitHub accounts.
* Use GitHub's built-in secrets management feature to securely store and manage sensitive information. (base64)
* All keys, credentials, and variables required for running software should be stored as secrets or environment variables in GitHub repositories.  These should never be hard coded in the source code or committed directly to the repository.
* Ensure that only authorized personnel have access to secrets, and regularly review and update them as needed.
* When a person leaves the organization, their access must be promptly revoked from all repositories and resources.

# Best Practices

Best practices in software development, such as proper documentation, unit testing, and code review, should be followed.

Developers should follow the recommendations presented in TS [Co-HUB](https://product-engineering-team.github.io/) according to the technology used.

Team members are encouraged to use, Azure Devops, GitHub's project management and issue tracking tools.

**Review and Updates**

This governance document will be reviewed and updated periodically to ensure relevance and effectiveness. Updates will be communicated to all members of the Product Engineering Team.

# Runners Execution

Any runners or scripts executed from repositories to external servers must comply with the guidelines and policies outlined in this governance document.

Prior to executing runners or scripts, team members should ensure that they have reviewed and follow the governance document's guidelines related to repository management, security, and best practices.

# Conclusion

Effective GitHub governance is essential for maintaining a collaborative and secure development environment within our Product Engineering Team. All team members should familiarize themselves with these policies and adhere to best practices when using GitHub.

Signature: Richard Saavedra   Date: 17 – Oct – 2023

Reviewed by:

Ignacio Henriquez