# ISO 27001 Compliant GitHub Code Repository Change Management Process Template

## Introduction:

This document describes the GitHub Code Repository Change Management process required for ISO 27001 compliance. Change management is critical in ensuring that code changes are controlled, tracked, and aligned with the organization’s Information Security Management System (ISMS) objectives. This process applies to all developers, stakeholders, and tools involved in managing code repositories hosted on GitHub for [COMPANY NAME].

## 1. Purpose of the Change Management Process

The Change Management process aims to:  
1. Ensure that all changes to code repositories are properly authorized and documented.  
2. Minimize risks associated with unauthorized or poorly executed changes.  
3. Maintain compliance with ISO 27001 requirements related to change control.  
4. Enable seamless collaboration and traceability for development teams.  
5. Facilitate rapid rollback and recovery in case of issues arising from code changes.

## 2. Change Management Process Flowchart

(Insert Flowchart Here)

Flowchart Description:

* Change Request Submission:  
  Triggered by new feature development, bug fixes, or security updates.  
  Key Inputs: GitHub issue tickets, feature requests, or incident reports.
* Impact Analysis:  
  Assess the potential impact of the change on the system, security, and stakeholders.  
  Stakeholders: Development team leads, IT security, and project managers.
* Approval Process:  
  Submit the change request for approval via pull request reviews in GitHub.  
  Responsible Parties: Designated code reviewers, project managers.
* Implementation:  
  Implement approved changes using version control best practices (e.g., feature branches, commit signing).  
  Tools: GitHub Actions, CI/CD pipelines.
* Testing and Validation:  
  Test the implemented changes in a staging environment.  
  Ensure compliance with security and functional requirements.  
  Key Outputs: Test results, validation reports.
* Deployment:  
  Deploy changes to the production environment following the approved schedule.  
  Tools: Deployment scripts, GitHub workflows.
* Post-Deployment Review:  
  Conduct a review to verify successful deployment and monitor for any issues.  
  Stakeholders: Development team, security team.
* Documentation and Feedback:  
  Document the change and update relevant records (e.g., change logs, GitHub issues).  
  Use feedback to improve future change management processes.

## 3. Roles and Responsibilities

Developers:  
- Create detailed pull requests with clear descriptions of changes.  
- Follow version control best practices during implementation.

Team Leads:  
- Conduct code reviews and approve changes.  
- Ensure alignment of changes with project objectives.

IT Security Team:  
- Review changes for security compliance.  
- Assess risks associated with changes and recommend mitigations.

Project Managers:  
- Coordinate testing and deployment schedules.  
- Oversee the resolution of issues arising during the change process.

GitHub Administrators:  
- Manage repository access and permissions.  
- Ensure the availability of tools required for compliance.

## 4. Narrative Context

The GitHub Code Repository Change Management process enables [COMPANY NAME] to maintain a structured approach to code development and deployment while mitigating risks associated with changes. For example, during a recent project, the structured approval process helped prevent the deployment of a critical vulnerability by catching it during the code review phase. This approach demonstrates the value of incorporating thorough testing and stakeholder involvement in achieving ISO 27001 objectives.

## 5. Revision and Maintenance

This document and the GitHub Code Repository Change Management process will be reviewed annually or as needed to reflect changes in organizational priorities, development practices, or regulatory requirements.

## Approval:

| Name | Title | Date |
| --- | --- | --- |
|  |  |  |
|  |  |  |