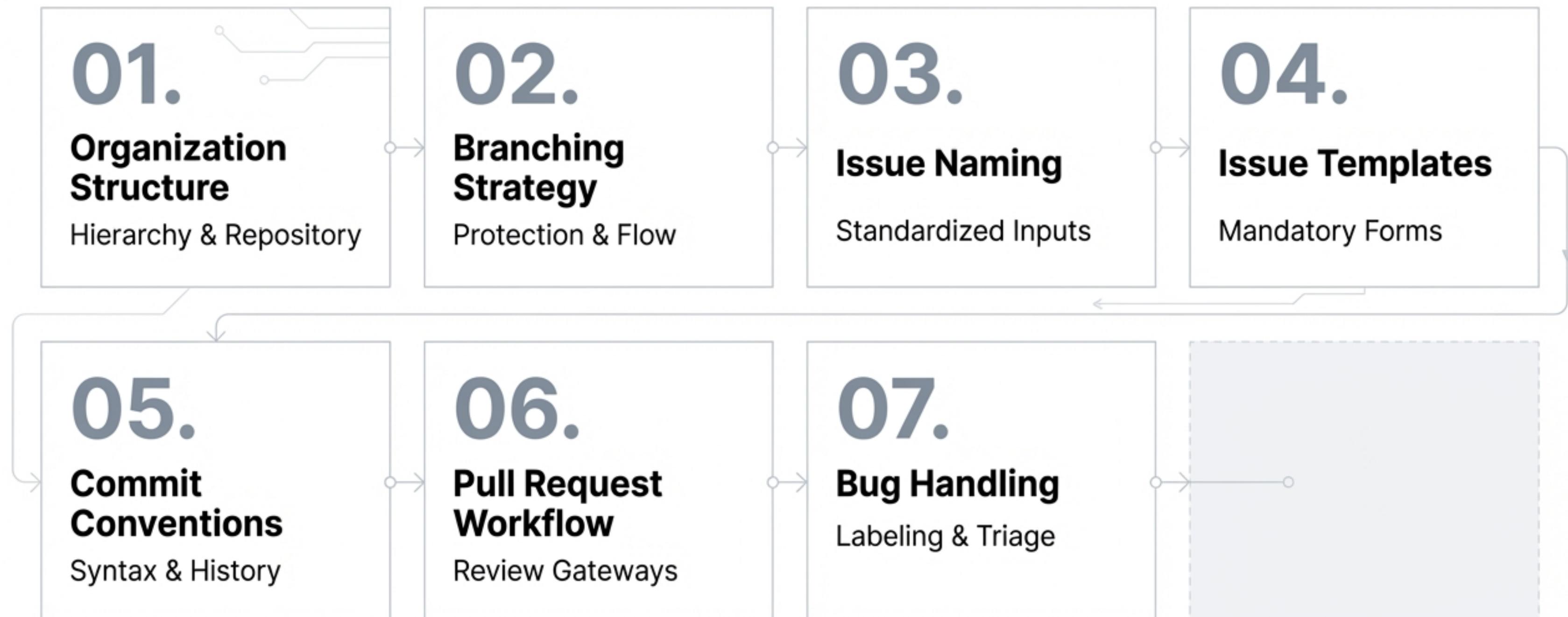


GitHub Development Rules & Guidelines

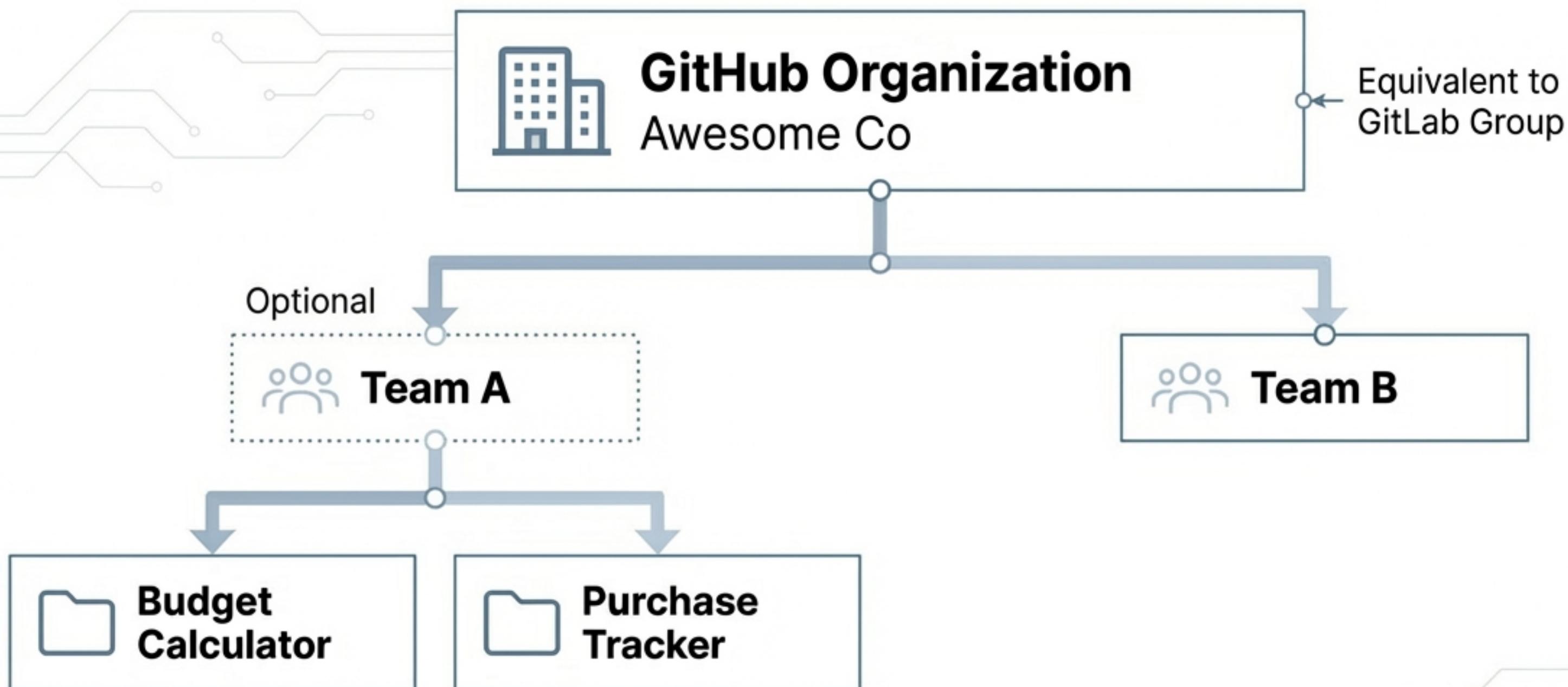
Standardized Workflows for High-Performance Teams

ADAPTED FOR GITHUB WORKFLOWS • VERSION 1.0

The 7 Rules of Development

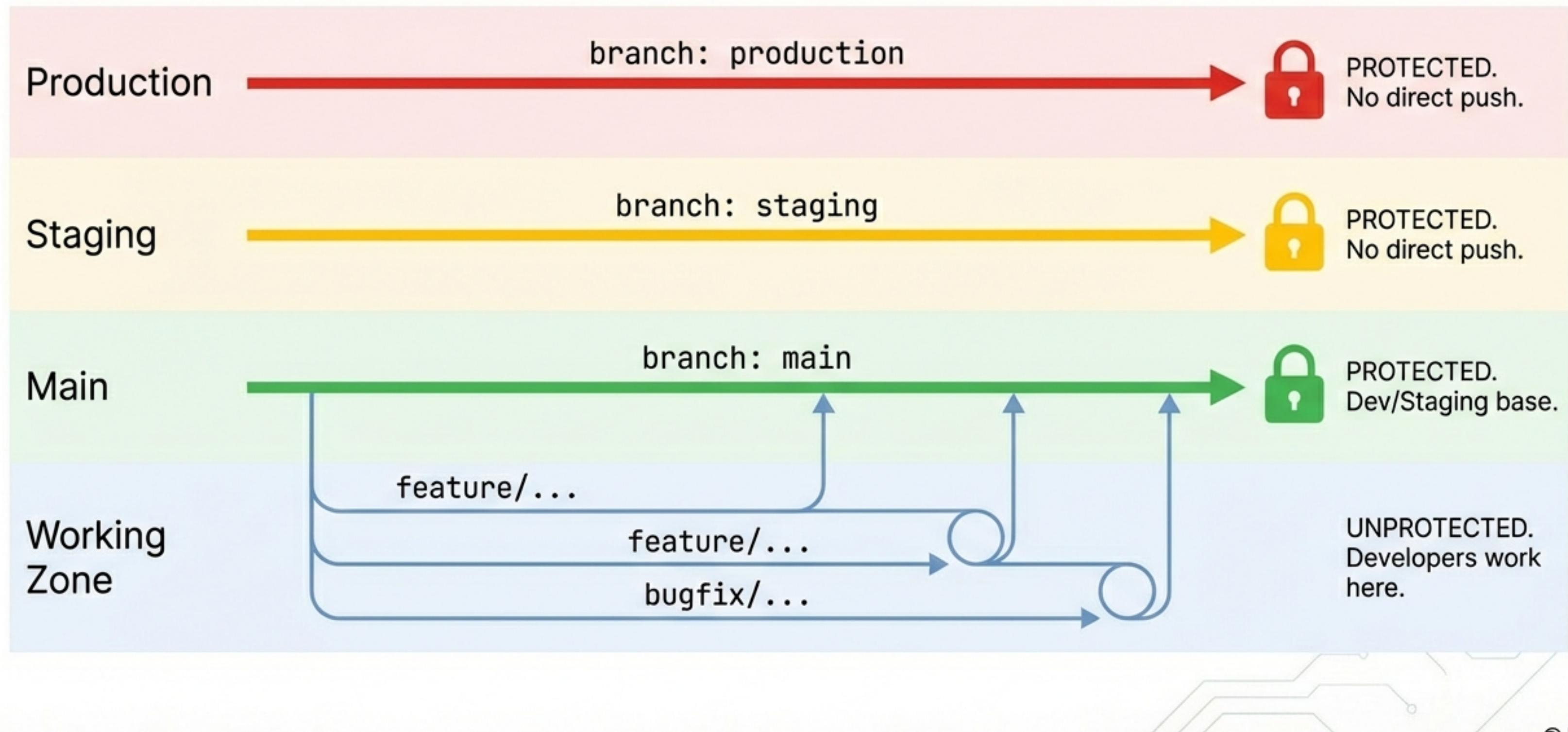


Rule 1: Organization & Repository Hierarchy



The Repository (Project) is where code lives.

Rule 2: Branching Strategy & Protection



Rule 2: Branch Naming Conventions

category/[issue-id]-description

Must select from
approved list

Links to Issue
Tracker

Short, kebab-case
summary

Feature Categories

- **feature/**: New features
- **bugfix/**: Minor bug fixes
- **hotfix/**: Critical production fixes
- **spike/**: Experiments/Prototypes

Maintenance Categories

- **refactor/**: No logic change
- **chore/**: Config, CI/CD
- **test/**: Writing tests
- **doc/**: Documentation
- **style/**: Formatting

Rule 3: Issue Naming Standards

Format: [Type] Issue Name

✓ Valid Examples

[Feature] Create user profile

[Bug] Login button unresponsive

[Refactor] Optimize recursion logic

[Doc] Update CHANGELOG spelling

[Test] Add unit tests for auth

✗ Invalid Examples

Fix the login button ← Missing Type

Update code ← Too vague

Bug found in production ← Incorrect format

Rule 4: Mandatory Issue Templates

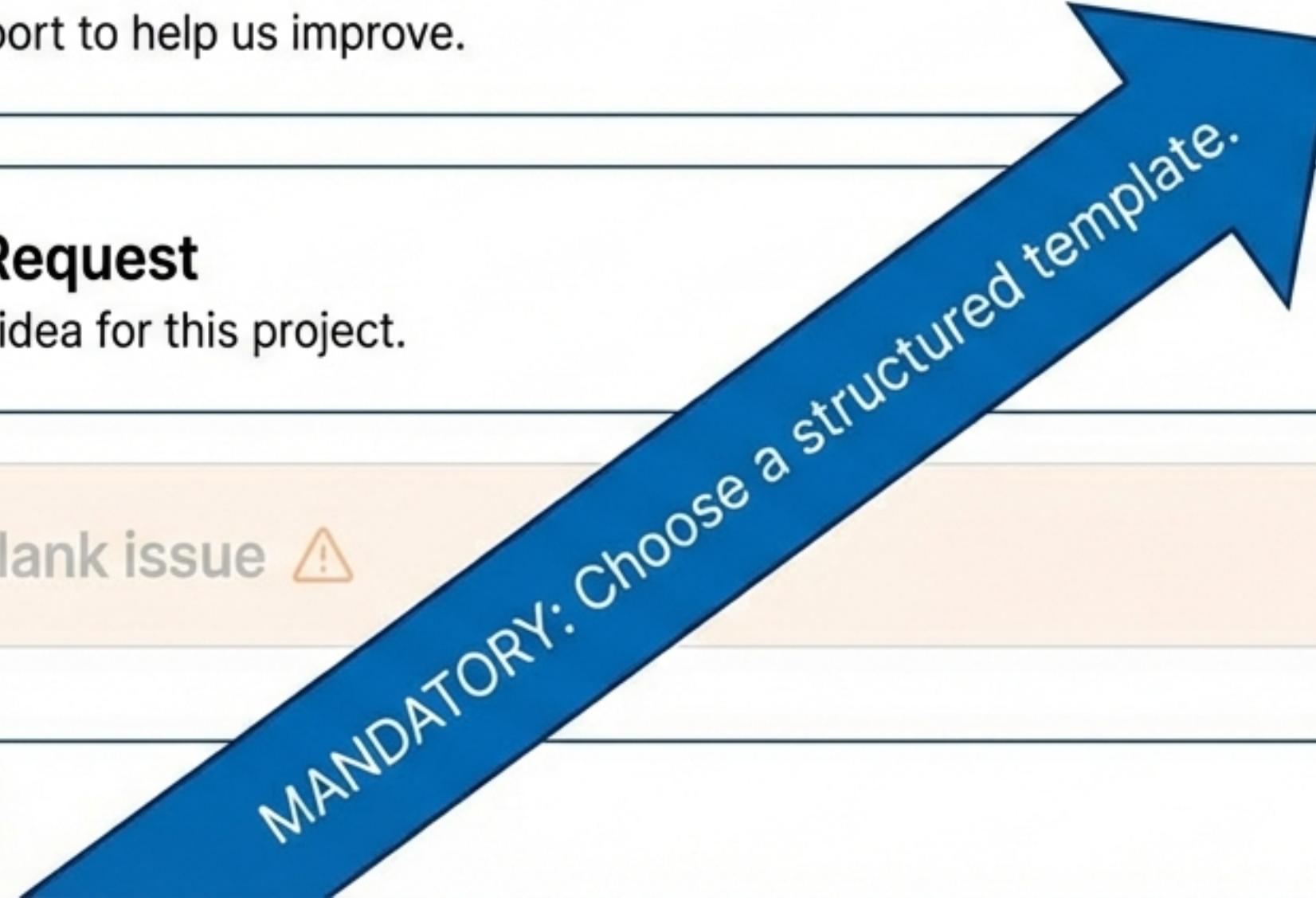
New Issue

Bug Report
Create a report to help us improve.

Feature Request
Suggest an idea for this project.

Open a blank issue 

MANDATORY: Choose a structured template.



Get started

Get started

Rule 4: Feature Request Anatomy

```
.github/ISSUE_TEMPLATE/feature_request.md
```

```
---
```

```
name: Feature Request
```

```
labels: Type::Feature, Status::Open
```

```
---
```

Summary
(Description of the idea)

Tasks

- [] Task 1
- [] Task 2
- [] Definition of Done

Checklists define the scope of work.

Rule 4: Bug Report Anatomy

```
.github/ISSUE_TEMPLATE/bug_report.md
```

```
---
```

```
name: Bug Report
```

```
labels: Type::Bug, Status::Open
```

```
---
```

Steps to reproduce

1. Go to...
2. Click on...



Steps must be reproducible.

Behavior

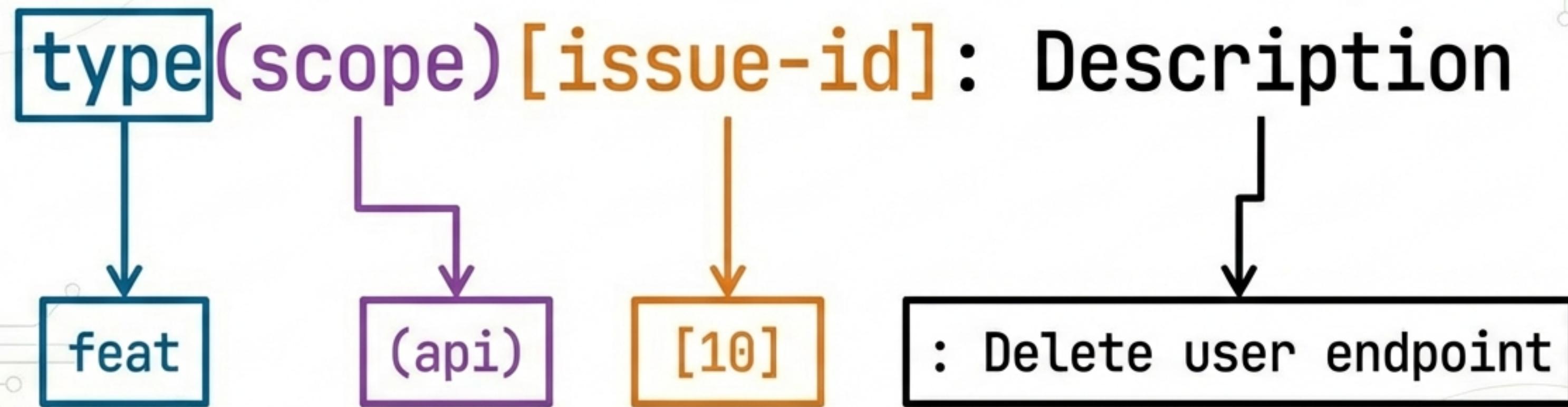
Current: Error 500 appears

Expected: User redirects to dashboard

Evidence

(Place logs or screenshots here)

Rule 5: Conventional Commits Syntax

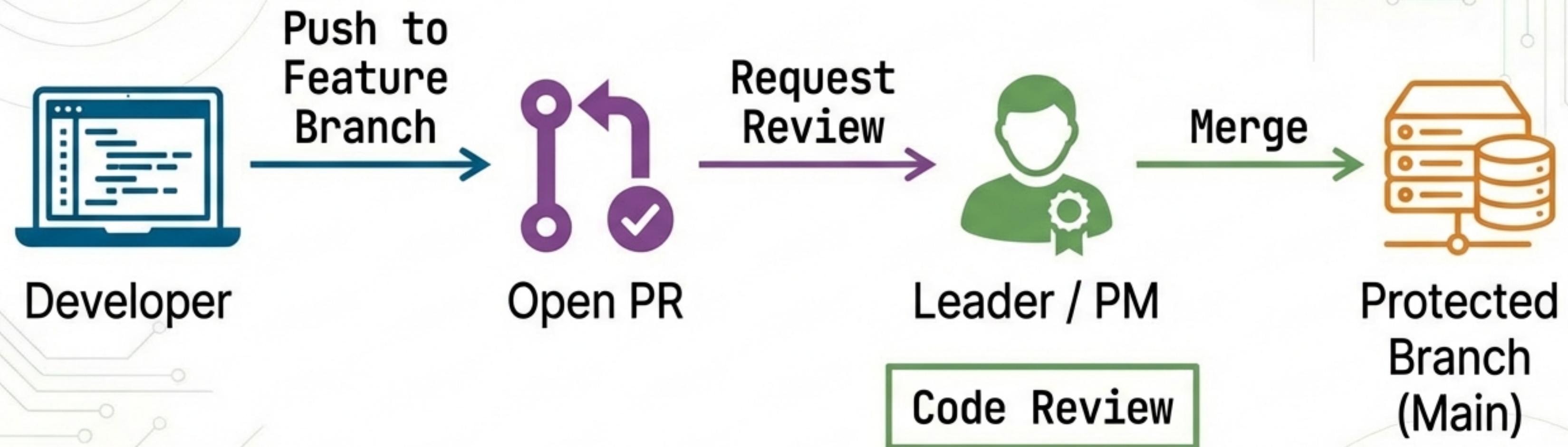


Example: **feat(api)[10]: Delete user endpoint**

Rule 5: Commit Types Reference

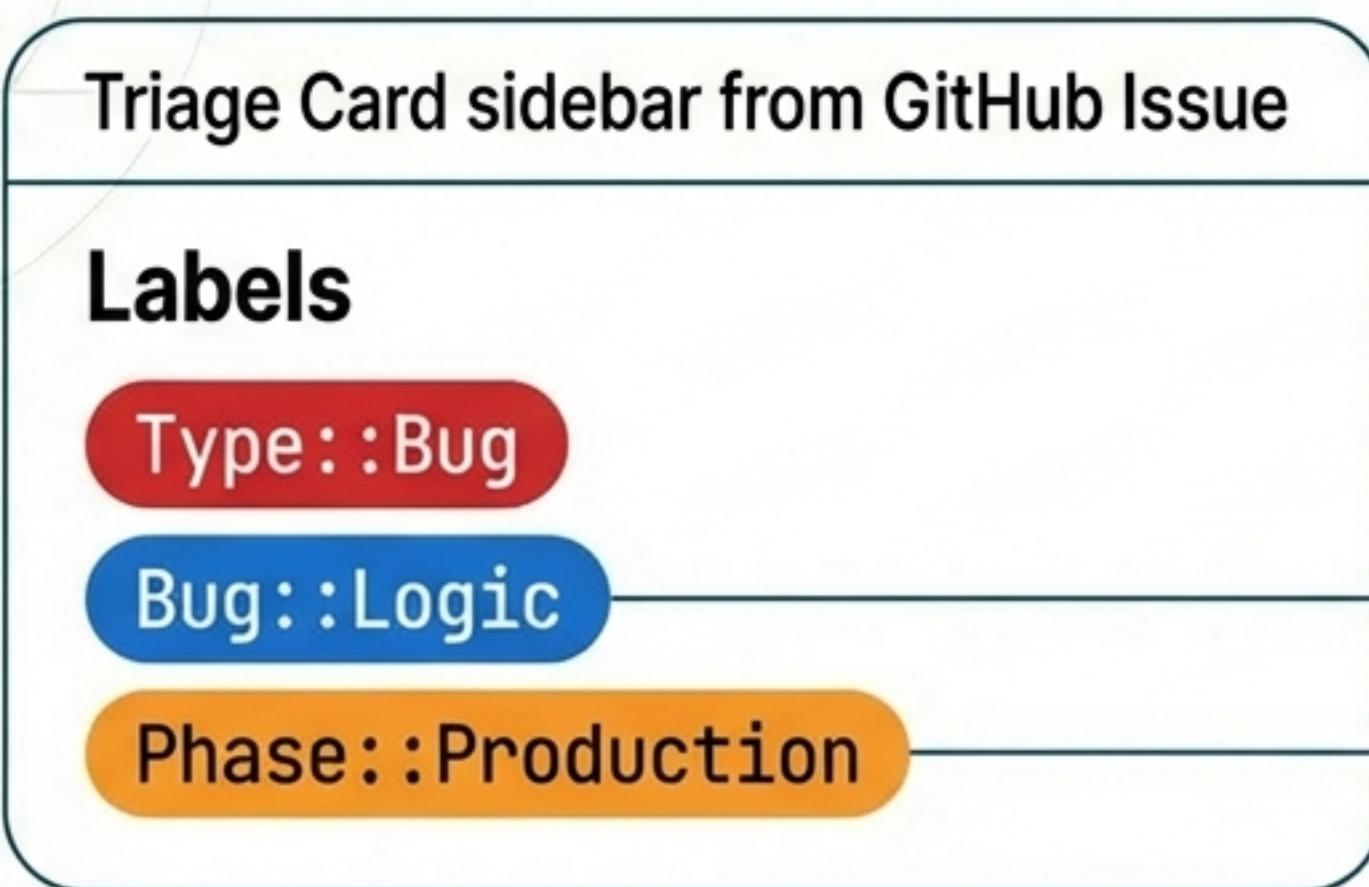
| | | |
|---|--|--|
| feat New feature (Value Add) | fix Bug fix (Value Add) | hotfix Urgent production fix |
| refactor Code change, no logic change | style Formatting, white-space | test Adding missing tests |
| doc Documentation only | chore Build process, aux tools | spike Experiment/Prototype |

Rule 6: Pull Request (Merge) Workflow



**NO DIRECT PUSH TO
MAIN/STAGING/PRODUCTION.**

Rule 7: Bug Analysis & Labeling



→ Category/Root Cause
→ Where was it found?

Definition: A bug is an issue requiring code correction, detected during development or production, where the root cause lies in implementation logic.

Maintainer Setup Guide

1. Create Templates

```
repo/  
  .github/  
    ISSUE_TEMPLATE/  
      bug_report.md
```

UI

Terminal

2. Commit Config

```
$ git push origin main
```

3. Protect Branches

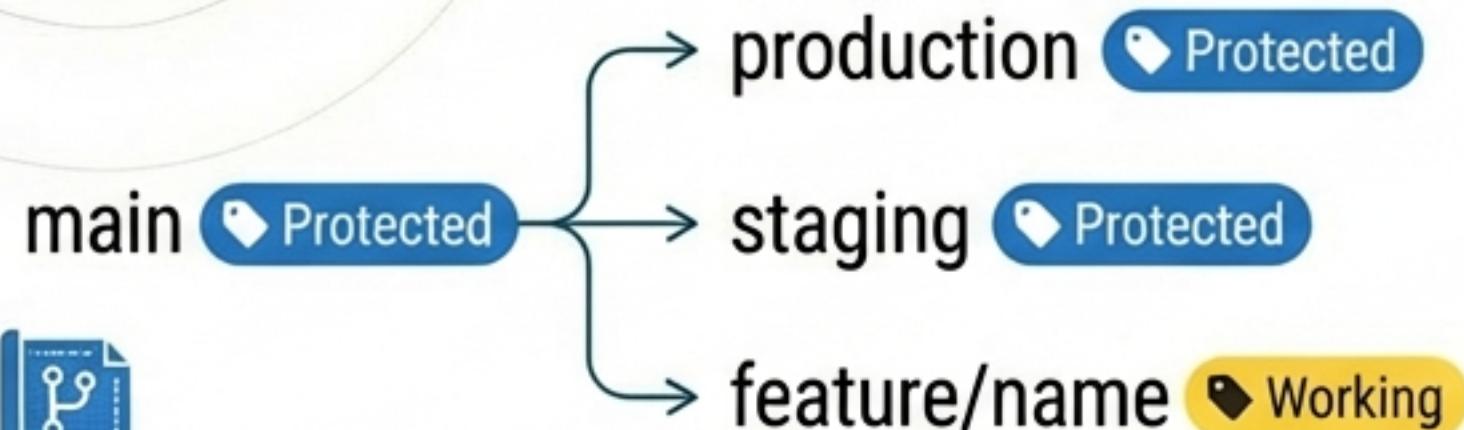
Settings Menu UI

Branch protection rule

- Require pull request reviews before merging
- Include administrators

The Developer's Cheat Sheet

Branching



Naming Syntax

Branch: category/[id]-desc

Issue: [Type] Title

Commit: type(scope)[id]: desc

Commit Types

feat : New Feature

fix : Bug Fix

refactor : Code Change

style : Formatting

test : Maintenance

style : Formatting

test : Tests

doc : Documentation

chore : Maintenance

Workflow



Label bugs with Type::, Category::, Phase::

Type::Bug , Category::Logic , Phase::Production

