

4. GitHub Management

4.1. Development Workflow

GitHub Organization

```
https://github.com/  
genesys-emu-ai-  
enablement-platform
```

a) GitHub Repository Naming

```
1 <use_case_name>  
2
```

Examples:

- sales-document-ingestion
- customer-intent-classifier

b) Branching Strategy

Environment Branches

- dev
- main
- pre-release_{version}

Feature Branches

```
1 <DAE-ticket_number>  
2
```

Example:

DAE-1245

4.2. Code Development Process

a) Commit Guidelines

- Commit all changes into the **feature branch**.
- Every commit should reference the **JIRA ticket**.
- Maintain a **supporting Confluence page** describing changes.
- Follow meaningful, descriptive commit messages.

b) Pull Request (PR) Guidelines

PR Title Format

```
1 <ticket_number>: <description_of_change>  
2
```

Example:

DAE-1245: Added ingestion workflow for Seismic assets

After creating the PR, share it on the PR Chat: [GenIQ - PR](#) | [Business Solutions & AI](#) | [Microsoft Teams](#)

PR Description Template

```
1 Description:
2 <Short 2-3-line description about changes>
3
4 JIRA:
5 Link: <Confluence-page-link>
6
7 Testing:
8 <Short 2-3-line summary of testing performed>
9
```

4.3. Environment Workflow

Principles

- All migrations must follow the **dev** → **uat** → **prod** path.
- Min Approvals:

ENV	Min. Approvals
DEV	1
MAIN	1

- Ensure unit tests, integration tests, and validation checks are passed before promotion.
- Since DEV is a highly active environment, a constant interaction between the team is recommended for every merge
- No direct commits to **main** (This is blocked by default)

Promotion Flow

1. Developer completes work in **feature branch** (local testing is recommended in this phase)
2. Create PR into **dev** branch → merge upon approvals
3. Dev Deployment is triggered manually
4. After validation, raise PR from **dev** → **main**.
5. Once merged, resources are deployed to TEST
6. After UAT sign-off, create a pre_release branch as pre_release_{version}
7. To release, create a release TAG in the pre_release branch
8. The TAG must contain the CR number in the descriptions as CHANGE_REQUEST = CR1231231
9. Once the TAG is successfully created, everything is promoted to PROD

4.4. Versioning & Tagging

Versioning tags should be used following the standard version naming pattern:

```
1 v<major>.<minor>.<patch>
2
```

Example:

v1.3.2
