J23375

細井 琉吾

GitHubとGitLabにおけるCI/CD

1. CI/CD

CI/CDとは、Continuous Integration (継続的インテグレーション) /Continuous Delivery(継続的デリバリー)の略称です。ソフトウェアの変更を常にテストし、自動で本番環境に適用できるような状態にしておく開発手法の事をいいます。

1. CI

CIとは、開発者がコードを修正し、その都度ビルドやテストを自動で行うプロセスや仕組みのことです。CIによって継続的にテストを行うことで、プログラムの不具合をいち早く見つけ、すぐに修復することができ、コードの品質を高く保つことができます。

3.　CD

　CDとは、ユーザーに継続的にアプリケーションを提供することを指し、自動的にシステムに変更が反映されるように設計された環境や概念のことをいいます。

4．GitHubとGitLabにおけるCI/CD

　GitHubとGitLabのCI/CD機能は、開発の自動化において大きな違いがあります。

4.1 GitHubのCI/CD

　GitHubの特徴は、外部のCI/CDツールを柔軟に統合できる点です。これにより、プロジェクトの特性やチームに合わせて最適なツールを選択することができます。また、GitHubのCI/CD機能である、GitHub Actionsを使用することでGitHub内でCI/CDを構築することができます。

4.2 GitLabのCI/CD

GitLabの特徴は、標準でCI/CDを提供している点です。GitLab内でビルド、テスト、デプロイを行えるため、他のツールとの結合を考慮することなく、すべて１つのツール内で完結することができます。また、視覚的にわかりやすいUIなのも特徴です。

5. GitLabでのIssue

GitLab上でのIssueとはタスクを指します。チームでの進捗共有、追跡に不可欠です。

Issue作成時には、いくつか意識することがあります。

タイトルはissueの内容がひと目見ただけでわかるようにしておきます。テンプレートはチームで用意、統一されたものを使います。ラベル、マイルストーン、担当者、期限も必ず設定します。説明は、何をするissueなのかを具体的に書き、だれでも再現・理解できるようにします。具体的には、issueの概要・目的、再現手順ややらなければいけないこと、期待される動作と実際の動作、その他の補足情報(備考)が書いてあるとよいです。

6. まとめ

GitHubとGitLabのCI/CDの違い、メリットを知った上で、自らの開発にはどちらが合っているのかを考え、開発を行いましょう。

参考文献

Valerie Silverthorne.” CI/CDとは？ベストプラクティスやメリットも解説”.GitLab home page.

<https://about.gitlab.com/ja-jp/blog/2022/02/03/how-to-keep-up-with-ci-cd-best-practices/>

友永 慎哉.” CI/CDとは何かをわかりやすく図解、具体的なツールや取り組み方とともに紹介する”.ビジネス＋IT.

<https://www.sbbit.jp/article/cont1/81640>

りょーちん.”【初心者向け】めっちゃ簡単!?GitLab issueの作り方”.Zenn.

<https://zenn.dev/secondselection/articles/_create_issues>

TECH BLOG.” GitHubとGitLabにおけるCI/CD機能の違い”.ISSOH.

<https://www.issoh.co.jp/tech/details/4219/#GitHubGitLabCICD>