**2025オリエンテーション課題Q5**

**課 題**GitHubやGitLabのCD(継続的デリバリー)/CI(継続的インテグレーション)についてまとめること。GitLabでのイシュー(Issue)について、書き⽅をまとめること

**CI/CDとは**CI/CDは「継続的インテグレーション（CI）」と「継続的デリバリー（CD）」の略で開発からテスト、デプロイまでを自動化し、ソフトウェア開発を効率化・高速化する仕組み。

**CI（継続的インテグレーション）**開発者がコードをリポジトリにプッシュすると、自動でビルド・テストが実行される。  
エラーの早期発見と品質維持が可能。

**CD（継続的デリバリー）**  
CIによる統合や一体化をさらに拡張したもので、より高度にかつ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。

**CI/CDのメリット  
品質向上**：常に最新・安定した状態を保てる。  
**開発スピードの向上**：作業が自動化され手作業が減少。  
**ツール間のズレ予防**：統一されたプロセスでトラブルを回避。  
**開発に集中できる**：バグ修正などの負担が軽減。

**ベストプラクティス**コード品質の管理とレビューを徹底。  
初期段階では頻繁に調整とフィードバックを実施。  
セキュリティ対策を強化（認証、アクセス制限など）。  
設定ファイルの共有・再利用を推進。  
エラーは即時対応。  
デプロイごとに環境をクリーンアップ。

**成果の評価方法  
サイクルタイム**：開発の所要時間。 **作業時間**：無駄な作業の削減度合い。 **エラー率**：エラーの減少傾向。 **インフラコスト**：費用対効果の確認。 **開発者の声**：現場の使い勝手や満足度。

**CI/CDパイプラインとは**「ビルド→テスト→デプロイ」などの工程をパイプラインで連結し、自動で処理。効率的な連携・高速な開発を実現するCI/CDの中核。

**ツール選びのポイント**正確性・信頼性・速度を重視。無料トライアルなどを活用し導入効果を見極める。GitLabはCI/CD一体型の優れたツールを提供（他ツールとの統合不要）。

**GitLabでのイシュー（Issue）の書き方まとめ**

**イシューとは？**GitLabにおける「Issue（イシュー）」は、タスク、バグ、改善点、アイデアなどを記録・共有するためのチケットです。チーム開発において、やるべきことを明確にし、進捗を可視化する役割があります。

**イシュー作成時の基本項目**

|  |  |
| --- | --- |
| 項目 | 内容 |
| タイトル | 一目で内容がわかる具体的な表現 例：ログイン時にエラーが出る |
| 説明（Description） | イシューの詳細。 必要な情報を網羅的に記載 テンプレートを活用するのが望ましい |
| ラベル（Labels） | カテゴリ分け 例：bug, feature, frontend, urgentなど |
| 担当者（Assignee） | 実行する担当者を指定 |
| マイルストーン（Milestone） | 関連するプロジェクトの スプリント・期限などを設定 |
| 期日（Due date） | タスクの締め切りを明示 |

**説明欄の構成**

概要  
このIssueの目的・背景を簡潔に記載

発生している問題 / 要望  
何が起きているか or 何を実現したいのか

再現手順（バグの場合）  
1. ユーザーがログインする  
2. 特定のページにアクセス  
3. 500エラーが表示される

期待される挙動  
正常にページが表示される

スクリーンショット / ログ  
必要に応じて画像やログを添付

関連イシュー・MR  
#123, !456 などをリンク

**良いIssueのポイント**目的と背景が明確：何のために作られたIssueかが一目でわかる  
再現性がある情報：具体的な手順やエラーメッセージを記載  
簡潔だけど必要な情報が網羅されている  
誰が見ても理解できる：専門知識がなくても分かる表現を意識する

**よく使われるラベルの例**

|  |  |
| --- | --- |
| ラベル名 | 用途 |
| bug | 不具合報告 |
| feature | 新機能追加 |
| enhancement | 改善・最適化 |
| urgent | 優先度高 |
| backend / frontend | 作業区域の区別 |
| needs review | レビュー待ち |

**運用のベストプラクティス**

作成後すぐに担当者・ラベル・マイルストーンを設定  
Issueを作っただけで終わらせない：進捗に応じて更新  
完了したら「クローズ」するのを忘れずに  
大きな課題は子イシューに分割する（親子関係で管理）

**よくあるアンチパターン（NG例）**

|  |  |
| --- | --- |
| 問題点 | 改善方法 |
| タイトルが「バグ」 「修正お願いします」だけ | 具体的に「ログインボタンが機能しない」などにする |
| 説明なし | 再現手順や状況を書いてもらう |
| ラベル未設定 | 管理が煩雑になるため必ず設定する |
| 担当者不在 | 放置されるリスクがあるので すぐアサインする |