

# GitLab と GitHub の違い

[ご購入はこちら](#)

久保 仁詩

GitLabはソースコードのGitバージョン管理システムです。特にオンプレミスでの運用が可能なので、社外にデータを置きたくない企業の需要が高い製品です。本稿では、GitLabとGitHubとの違い、GitLabの活用例について解説します。

## 機能やライセンスの比較

### ● 開発/運用/セキュリティに重点を置く

#### GitLab

GitLabとGitHubの最大の違いは、カバーできる業務領域です。GitLabは開発および運用のライフサイ

クル全体をカバーする機能を提供しているのに対し、GitHubは使いやすくシンプルな操作性とコーディングやコラボレーションに重点を置いており、他の工程については外部ツールとの連携が必要になることがあります。GitLabは単一のプラットフォーム上で開発(Dev)、セキュリティ(Sec)、運用(Ops)の各工程をシームレスに統合(DevSecOps)することによって、チームが迅速かつ安全にソフトウェア開発を行えるようになります。

また、GitLabのUIは日本語に翻訳されており、日本語での利用が可能です<sup>(1)</sup>。一方、GitHub UIは日本語に対応していません。表1に機能を比較します。

表1 GitLabとGitHubの比較(◎非常に良い、○良い、△あまり良くない)

開発段階	GitLab	GitHub	備考
Plan (プロジェクト計画)	◎	○	GitLabはプロジェクト管理機能が充実しておりイシュー、エピック、マイルストーン、ロードマップなどが利用可能。GitHubもプロジェクト管理機能を提供しているが、GitLabほど包括的ではない
Code (コーディング)	◎	◎	両者ともにコーディングとバージョン管理の機能が優れており、コード・レビューやブランチ管理などが可能
Build (ビルド)	◎	◎	GitLabは2014年からビルトインのCI/CD機能を提供しており、柔軟なパイプライン設定が可能。GitHubは2019年からGitHub Actionsを提供しており、CI/CD機能を拡張
Test (テスト)	◎	○	GitLabはCI/CDパイプラインにテスト機能を統合しており、自動テストが容易。GitHubでもGitHub Actionsを使用してテストを実行できるが、GitLabほど統合されていない
Release (リリース)	◎	○	GitLabはリリース管理機能を内蔵しており、タグ付けリリースノート、アセット管理が可能。GitHubもリリース機能を提供しているが、GitLabほど包括的ではない
Deploy (デプロイ)	◎	○	GitLabは最小限の設定でビルド・テスト・デプロイを自動的に行うAuto DevOps機能を統合しており、さまざまな環境へのデプロイが可能。GitHubでもGitHub Actionsを使用してデプロイを自動化できるが、GitLabほど統合されていない
Operate (運用)	◎	△	GitLabは環境管理、Kubernetesクラスタ管理、TerraformやGoogle Cloudとの統合などをサポート。GitHubにはこれらの機能はない
Monitor (監視)	◎	△	GitLabはPrometheusなどの監視ツールとの統合が可能で、パフォーマンス・メトリクスの追跡が容易。GitHubには直接的な監視機能はないが、外部ツールとの連携は可能
Security (セキュリティ)	◎	△	GitLabはSAST、DAST、脆弱性管理などのセキュリティ機能を提供。GitHubもセキュリティ機能を提供しているが、GitLabほど包括的ではない
Compliance (コンプライアンス)	◎	△	GitLabはライセンス・コンプライアンス、アクセス・コントロール、監査ログなどの機能を提供する。GitHubではコンプライアンス機能は限定的
Analyze (分析)	◎	△	GitLabはバリュー・ストリーム分析やDORAメトリクスなどの分析機能を提供し、開發生産性の可視化が可能。GitHubもInsightsと呼ばれる分析機能を提供しているが、GitLabほど包括的ではない
AI Assistant (AIアシスタント)	○	◎	AI領域ではGitHub Copilotが先行しており、リアルタイムのコーディング支援により個々の開発者やチームのコーディング効率が向上する。GitLabもGitLab Duoと呼ばれるAIを開発プロセス全体に統合し、生産性とセキュリティの向上を目指す