

# ソフトウェア開発ライフサイクル（SDLC）におけるエージェント活用

---

GitHub Copilot の機能拡張により、ソフトウェア開発ライフサイクル（SDLC）の各工程で開発者を強力に支援できるようになりました。GitHub 上の Issue や Pull Request の取り扱い、外部サービスとの連携、そしてもちろんコード生成までを含みます。本ラボでは、これらの機能を実際のユースケースとともに体験し、活用のコツを学びます。

## ラボの概要

[!IMPORTANT] GitHub Copilot をはじめとする生成 AI は決定的（deterministic）ではなく確率的（probabilistic）に動作するため、生成されるコードや変更されるファイルは毎回同一ではありません。そのため、ラボ内のスクリーンショットやコード片と、皆さんの環境での結果に差異が生じることがあります。これは想定どおりで、この種のツールの性質です。

うまく動かない場合や壊れているように見える場合は、メンターに相談してください。

このラボでは、GitHub Copilot のエージェント機能でよくあるワークロードを順を追って体験します。

### 0. 環境のセットアップ

1. [Issue を GitHub Copilot コーディングエージェントに割り当てる](#)（非同期で作業を進める）
2. [Model Context Protocol（MCP）で外部サービスを設定・連携する](#)
3. [カスタムインストラクションで Copilot に文脈を与える](#)（指示ファイル・プロンプトファイル・チャット参加者）
4. [Copilot のエージェントモードを使ってサイト全体の更新を行う](#)
5. [Copilot コーディングエージェントの成果をレビューする](#)

## シナリオ

あなたは Tailspin Toys（DevOps をテーマにしたボードゲームのクラウドファンディングを提供する架空の会社）の新任開発者です。アプリケーションや DevOps フローに対する改善点を Issue として記述し、さらに「カテゴリ」と「パブリッシャー」でゲームを絞り込める機能を実装します。サイトと Copilot の機能を行き来しながら、反復的にタスクを進めていきます。

## はじめに

まずは[セットアップ](#)から始めましょう！