

Spec-Driven Development with Kiro

AWS Classroom Training

コース概要

Spec-Driven Development with Kiro は、AI 支援によるエージェント型開発手法を採用する開発者、ソフトウェアエンジニア、テクニカルリード向けに設計された 2 日間のインストラクター主導型トレーニングコースです。本コースでは、エージェント型コーディングの原則と、ソフトウェア開発向けエージェント型 IDE である Kiro を使用してそれらを適用する方法について学習します。参加者は、Feature Specs および Bugfix Specs を通じて機能を計画・実装し、steering ファイルと実行モードでエージェントの動作を制御し、Model Context Protocol (MCP) サーバーおよび Kiro Powers でエージェントの機能を拡張する方法を習得します。また、Agent Hooks によるワークフローの自動化、Agent Skills やカスタム sub-agent による再利用可能な専門知識のパッケージ化、チームコラボレーションおよびエンタープライズガバナンス向けのプロジェクト構成についても扱います。ハンズオンラボは各日のモジュール内容と組み合わせて実施されます。1 日目には、学習者は各 Kiro 機能を学習しながら段階的に天気アプリケーションを構築します。2 日目には、事前構築されたマイクロブログアプリケーションのデプロイ、拡張、チーム対応を行います。

- コースレベル: 中級
- 期間: 2 日間

アクティビティ

本コースには、プレゼンテーション、ハンズオンラボ、デモンストレーション、グループ演習が含まれます。

コースの目的

本コースでは、以下について学習します。

- 仕様を用いて機能や修正の計画・設計・実行を行う、仕様駆動開発 (Spec-Driven Development) のワークフローを実装する
- steering ファイル、コンテキストプロバイダー、実行モード (Autopilot および Supervised)、カスタム sub-agent を使用してエージェントの動作を構成・制御する
- Model Context Protocol (MCP) サーバーおよび Kiro Powers を使用して、外部ツール、サービス、ドメイン専門知識を統合し、エージェントの機能を拡張する
- Agent Hooks、Agent Skills、共有 steering、カスタム sub-agent を使用して、開発ワークフローを自動化・スケールする
- チーム固有のワークフロー向けに、カスタム Kiro Powers、Agent Skills、カスタム sub-agent を作成し、再利用可能なエージェント機能を構築・共有する

- MCP レジストリの許可リスト化、モデルガバナンス、拡張機能管理に関するエンタープライズガバナンス制御を適用する

対象者

本コースは、以下の方を対象としています。

- ソフトウェア開発者およびエンジニア
- AI 支援型開発ワークフローを採用するテクニカルリード
- エージェント型コーディング手法を採用する開発チーム
- AI エージェントをデリバリーパイプラインに統合する DevOps エンジニア

前提条件

本コースの受講者には、以下を推奨します。

- 少なくとも 1 つのプログラミング言語に習熟していること
- Git バージョン管理の実務知識があること
- RESTful API およびクラウドアプリケーションの概念を基本的に理解していること
- 生成 AI の概念 (LLM、トークン、推論) に関する基礎的な知識があること

コースの概要 (アウトライン)

1 日目

モジュール 1: Kiro を用いたエージェント型開発の概要

- エージェント型コーディングのパラダイムと Kiro のコア機能
- 効果的なエージェント型開発の 5 つの原則
- Kiro IDE と Kiro CLI
- Vibe セッション、Spec セッション、モデル選択

モジュール 2: Feature Specs による仕様駆動開発

- 3 フェーズの spec ワークフロー (Requirements、Design、Tasks)
- Requirements-First ワークフローと EARS 記法
- Design-First ワークフロー (High-Level Design および Low-Level Design)
- プロパティベーステストによる正確性の確保

モジュール 3: Steering、コンテキスト、エージェント制御

- steering ファイルと基盤コンテキスト

- カスタム steering ファイルと適用モード (always、fileMatch、manual、auto)
- ワークスペーススコープとグローバルスコープ
- コンテキストプロバイダーとセッション管理
- Autopilot および Supervised 実行モード
- 信頼済みコマンドと信頼済みツール

モジュール 4: Bugfix Specs

- Bugfix Specs を使用する場面
- バグ修正の分析フェーズ: 現在の動作、期待される動作、変更しない動作
- 根本原因分析と設計
- 回帰防止のためのプロパティベーステスト

1 日目のラボ: Kiro による仕様駆動開発の基礎

2 日目

モジュール 5: MCP サーバーとツール統合

- Model Context Protocol の基礎とクライアント・サーバーアーキテクチャ
- ワークスペーススコープおよびユーザースコープでの MCP サーバーの構成
- サーバー、ツール、自動承認の管理
- チャットでの MCP ツールの使用とトラブルシューティング

モジュール 6: Kiro Powers

- コンテキスト過負荷の問題と Powers による解決方法
- Power のディレクトリ構造 (POWER.md、mcp.json、steering)
- IDE および kiro.dev からの Powers のインストールと使用
- カスタム Powers の作成と共有

モジュール 7: Agent Hooks、Skills、カスタム Sub-agent

- Agent Hooks: トリガー、アクション、作成
- Agent Skills: プログレッシブディスクローダーを備えたポータブルな命令パッケージ
- カスタム sub-agent: スコープ設定されたツール権限と委任
- Skills、Steering、Powers の比較

モジュール 8: チームコラボレーションとエンタープライズガバナンス

- .kiro ディレクトリを通じた共有ワークスペース構成

- チーム向けの steering ファイル構成と spec レビューの規約
- エンタープライズガバナンス: MCP レジストリの許可リスト化、モデルガバナンス、拡張機能レジストリ、Web ツール

2 日目のラボ: Kiro による拡張と自動化

Data sheet version: 9/2026