

Kiro

— AI と共に考え、共に作る。信頼できる開発パートナー


Amazon Web Services Japan
Solutions Architect

濱上 和也

2026/03/17



自己紹介

- 濱上 和也 (はまがみ かずや)
- アマゾン ウェブ サービス ジャパン
ソリューションアーキテクト
- 小売 (ファッションアパレル) のお客様を
中心にご支援
- 好きな AWS サービス
Amazon Connect 



本日のアジェンダ

- 1 AI コーディングとは、現在地と課題
- 2 Kiro 誕生秘話 — なぜ AWS が AI IDE を作ったのか
- 3 Kiro のアプローチ — 仕様駆動開発 ・ Steering ・ Hooks ・ Powers
- 4 Kiro の活用シーン — 日常業務 ・ 開発 ・ 運用
- 5 Kiro と共に成長する — Autonomous Agent ・ Enterprise

ゴール：Kiroの全体像をつかみ、自分ごとに活用ユースケースを想像できること

解像度
高めよう



AI コーディングエージェントとは？

AI エージェントが IDE を直接操作することで
単なる対話ではなく自律的な開発支援を行うことが可能になりました

従来の生成 AI コーディング

コードの書き方を教えて

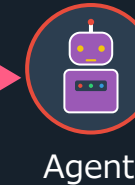
```
function example() { ... }  
こちらがサンプルコードです
```

- 質問と回答の繰り返し
- 開発者の判断でコードを取り込む
- あくまで **サポート** に留まる

AI コーディングエージェント

人間による指示

“既存のプロジェクトの中身を把握して、新機能として xx を追加して。追加した機能がちゃんと動くか、きちんとテストもしてね”



コード生成・実行

テスト作成・実行

デバッグ・エラー修正

ドキュメント生成

- 指示に基づき AI が自律的に開発
- テストやデバッグについても自動化
- **ナビゲーター** ではなく **ドライバー** に

AI コーディングの進化

2024

アシスタント

開発者のコーディングを
効率化

タイピング支援が中心

2025

エージェント

人が関与しながら
開発タスクを
最初から最後まで完了

2026

自律

制限された自律性を持って
開発タスクを最初から最後
まで完了

効率化

自律性



AI コーディングツールの現在地と課題

90%

Fortune 100 企業が
AI コーディングツール
を導入済み

85%

の開発者が日常的に
AI アシスタントを使用

42%

のコードが AI 生成

しかし、AI 生成コードを **完全に信頼** している開発者は **わずか 33%**

「使うかどうか」ではなく「どう信頼して使いこなすか」の時代へ

出典 : [TechCrunch \(Jul 2025\)](#) / [JetBrains DevEco 2025](#) / [Sonar State of Code 2025](#) / [Stack Overflow Survey 2025](#)



AI コーディングツールの現実

「ログイン画面を作って」
→ コードは出る。でも、**使えるコードにならない。**



既存環境との統合

認証・DB・API・インフラ...
既存の文脈や複数サービスの
組み合わせを理解できない



セキュリティ・規約

チームのルールや
セキュリティ要件を
守れない



品質・信頼性

生成コードの品質が不安定
テスト・レビューなしでは
本番に使用できない

Kiro 誕生秘話 - Vibe Coding におけるコミュニケーションの課題



- AI が生成したコードはよかったが、ソフトウェアの挙動は望んだものではなかった
- その都度、繰り返し最初から新しいプロンプトを試した - どんどん仕様の記述のようになっていった
- 「これは仕様書だ」

Clare Liguori — Senior Principal Software Engineer, AWS (re:Invent 2025 Keynote)



- すでにチームで上手く行っている方法に立ち返って見た
- 自分たちが新しい機能を実現するとき、どういう手順を踏むか…

1. どのようなものか記述、要件を洗い出す
2. Design Doc をレビューする
3. スプリントのタスクに落とし込む

同じアプローチが取れないか？

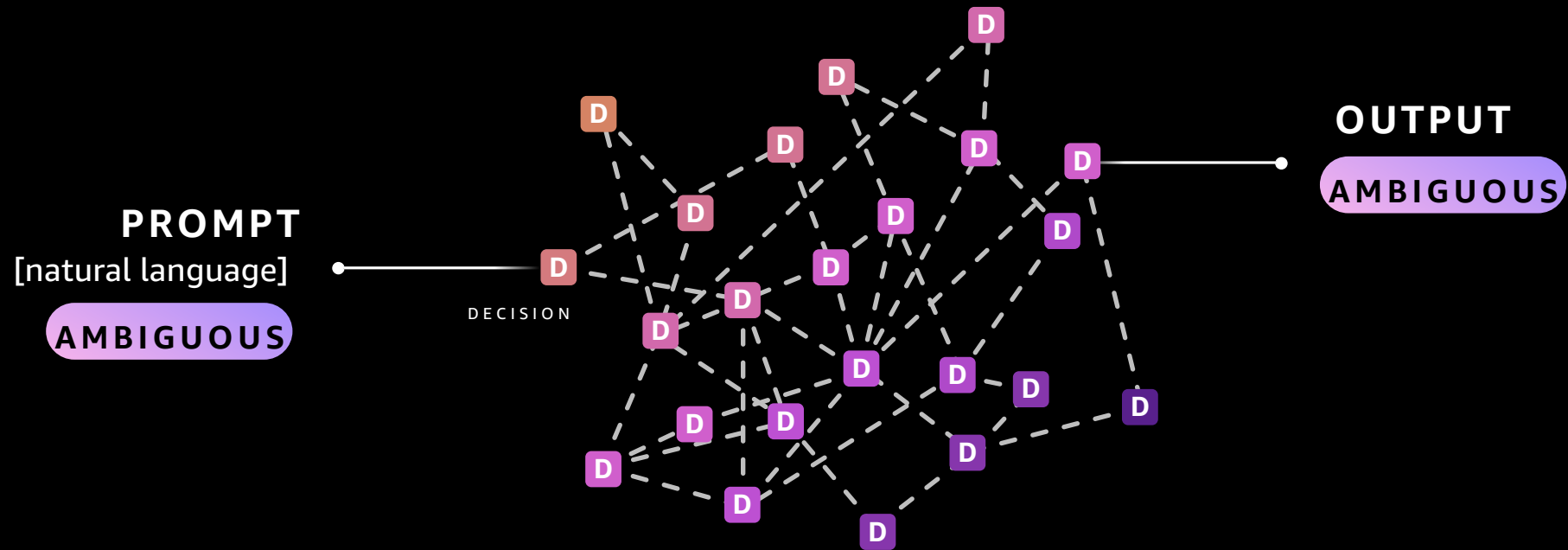
Requirements

Design

Tasks



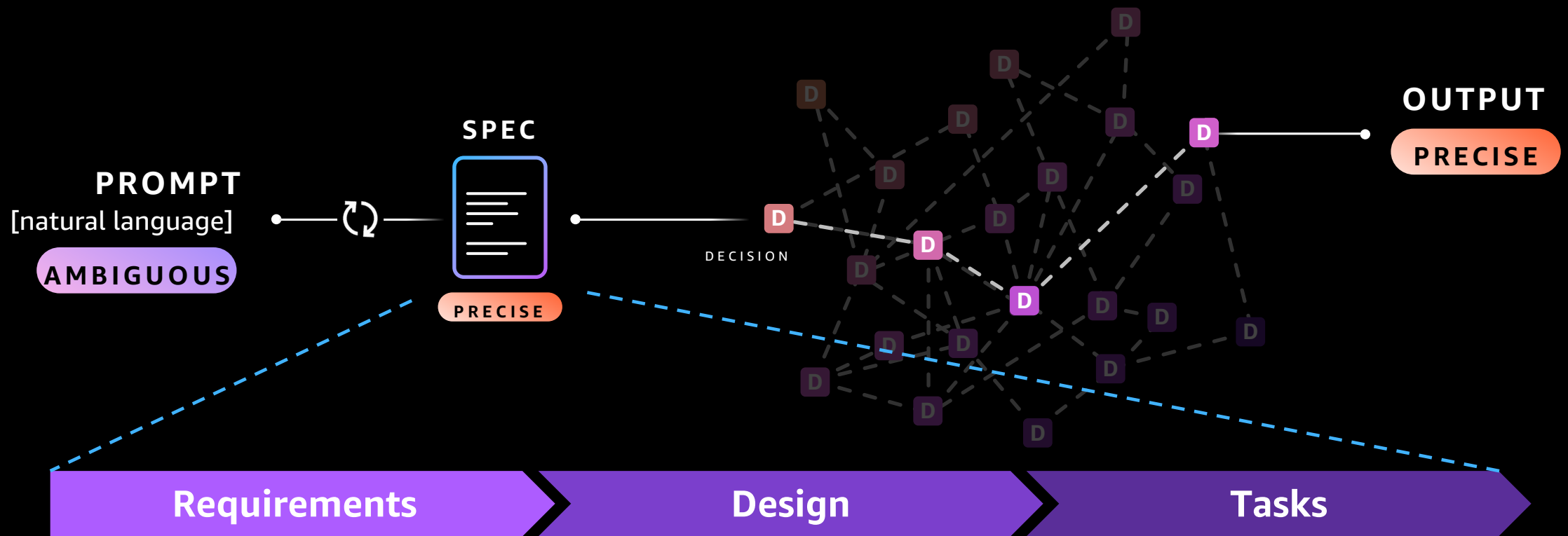
Before — 曖昧なプロンプト → 曖昧なアウトプット



Vibe Coding において、100万通りの成果物があり得る課題に対し、しばしば非常に曖昧な指示を与えがち。

しかし思い描いていたものは恐らくそのうちの一つだけのはず...

After — Spec で精度を確保 → 正確なアウトプット



同じように曖昧なタスクを与えても、いきなりコードに飛び込まず、
まず明確な Spec を確認し、方向性を合意する

Human to Human

[Rapid prototypes]

PRECISE

Human to Machine

[Specifications]

PRECISE

ユーザーとは **Prototype** で、AI とは **Spec** で
明確なコミュニケーションを

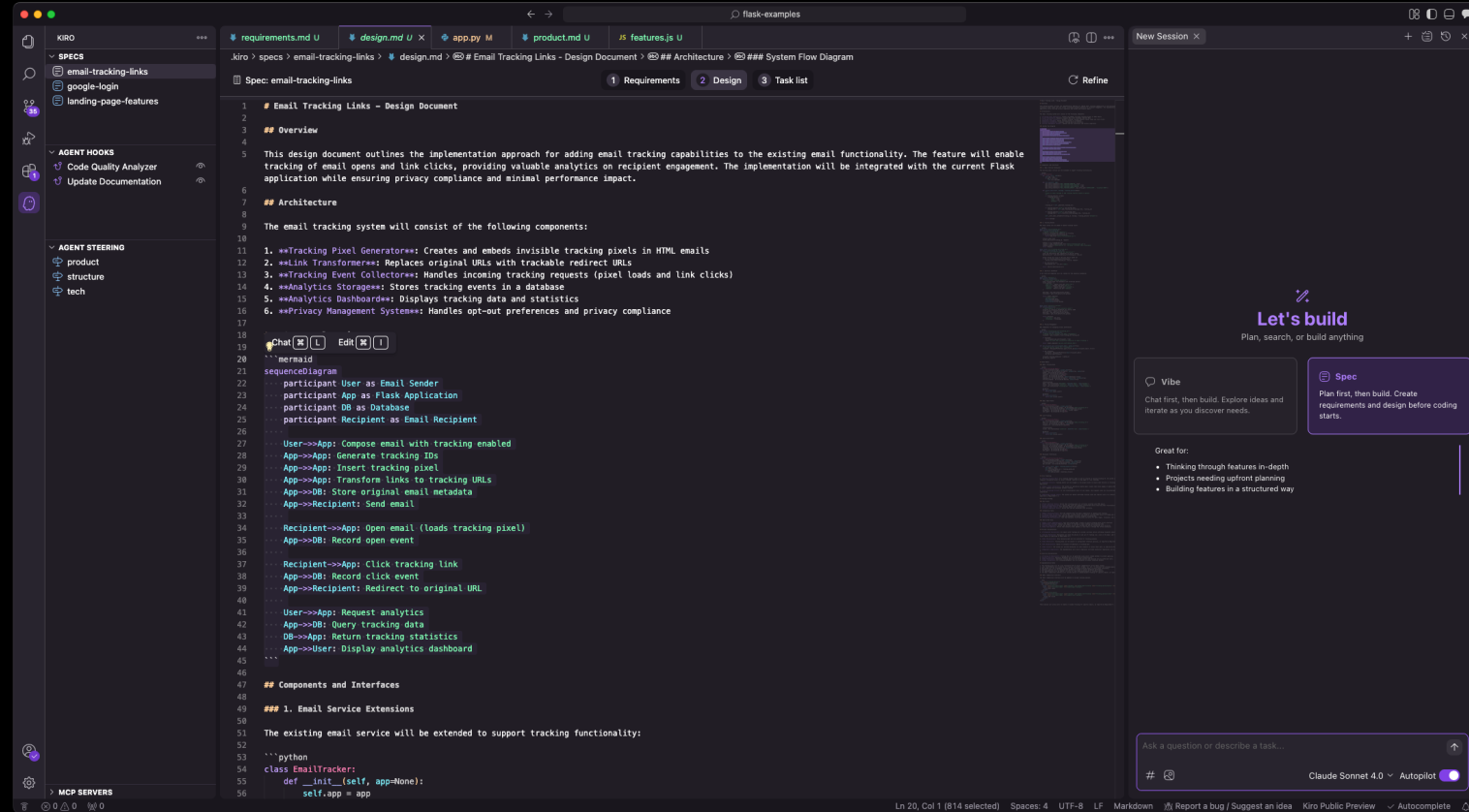
Kiro IDE

プロトタイプから本番までをサポートする AI 開発環境

VS Code ベースの 統合開発環境 (IDE)

自然言語で指示 → **仕様書を自動作成**
→ 合意してからコード生成

従来のコーディングアシスタントの課題
「AIが生成するコードは前提が不明瞭で本番環境で利用するには不安だ」
を解決



Kiro のアプローチ — 仕様駆動開発

- ▼ Specs — 機能やバグ修正ごとに要件・設計・タスクを定義（それぞれmdファイルを作成）

Requirements

Design

Tasks

- ▼ Steering / Hooks / Powers — プロジェクト全体を支える永続的な仕組み



Agent Steering

プロジェクトの技術スタック・
規約・構造を永続的に AI に伝達

毎回説明し直す必要なし



Agent Hooks

ファイル保存時に自動で
テスト実行・ドキュメント更新

品質を仕組みで担保



Kiro Powers

ドメイン専門知識と MCP 接続を
動的にロード

必要な知識を必要な時に

ユースケース① 開発 — ローカルアプリから AWS インフラまで

Kiro + AWS CDK で本番レベルのインフラをコードで構築



CDK



Lambda



API GW



DynamoDB



CloudWatch

自然言語で伝えるだけで、AWS CDK のコードを生成。
サーバーレス構成からコンテナ、データベースまで
AWS の全サービスを組み合わせたシステムを構築可能。

社内ツールだけでなく、本番サービス基盤も射程に。

日本の事例

SBI 証券

1 日 2 兆円超

の国内株式取引を AWS 上で処理

AWS CDK でインフラをすべてコード化
CI/CD パイプラインで本番環境を管理

1,300 万口座超の証券総合口座
1 日約 360 万件の取引処理
1 日 1 億アクセス

キャパシティ拡張: **オンプレ比 半分以下**

出典:

<https://aws.amazon.com/jp/solutions/case-studies/sbi-securities/>
<https://aws.amazon.com/jp/about-aws/whats-new/2024/04/sbi-securities-completes-migration-online-trading-systems-to-aws/>



ユースケース② 日常業務 — Agentic Chat + MCP

ドキュメント・資料作成

「この設計書を要約して」 →
要約・翻訳・資料作成を自然言語で

コードレビュー・リファクタリング

「このコードをレビューして」 →
品質改善・リファクタリング提案

アーキテクチャ図の生成

「この設計書からアーキテクチャ図を
生成して」 → 構成図を自動作成

技術質問への回答

社内の技術質問に回答。
最新機能の情報にも対応

(参考) MCPとは — 外部サービスとの連携

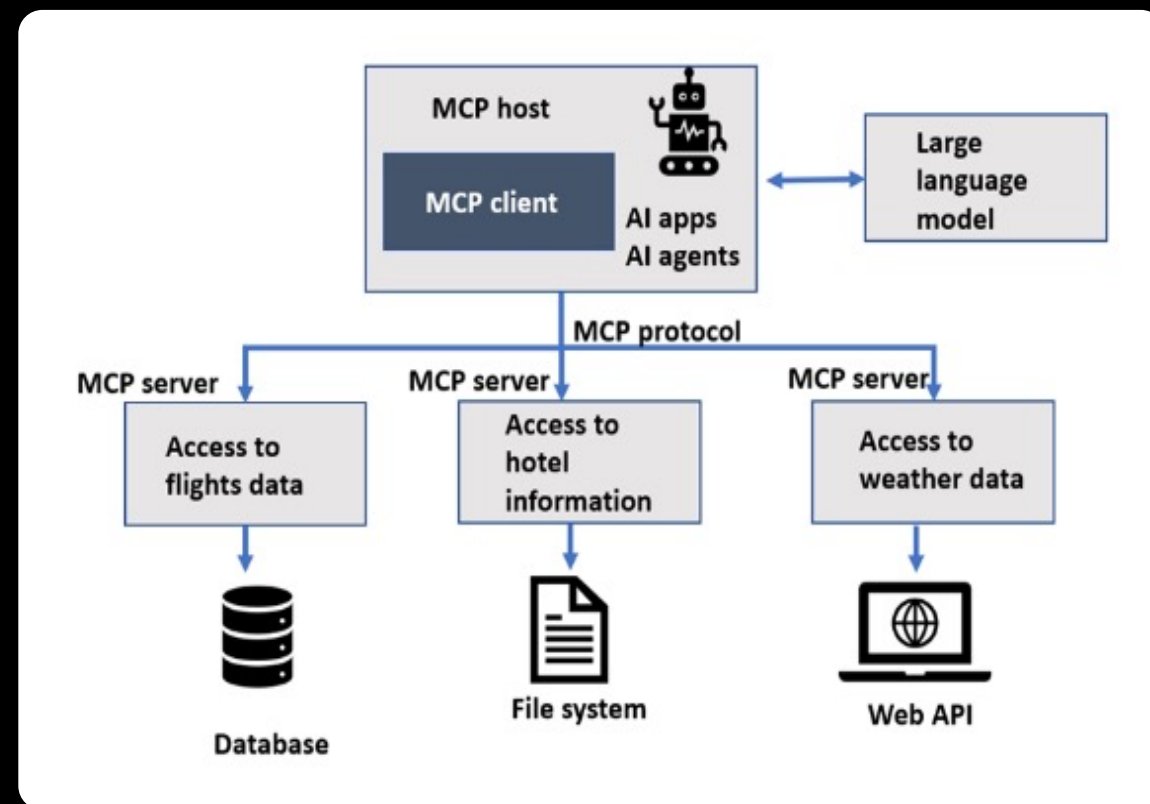
Model Context Protocol

AI エージェントが外部ツール・データソースに安全にアクセスするための標準プロトコル

AWS 公開の MCP サーバーで

CloudWatch、**S3**、**コスト情報**など

AWS サービスに自然言語でアクセス可能



MCP アーキテクチャ概要

ユースケース③ 運用 — 自然言語でオペレーション



運用業務を自然言語で完結

Agentic Chat + MCP / Powers で

Before

5 ~ 6 画面を手動で
行き来



After

Kiro に聞くだけで
完結

主なユースケース

- 1 **AWS コスト分析**
「先月の AWS コストを分析して」
→ 可視化・改善提案
- 2 **運用モニタリング**
「CloudWatch のログとメトリクスを確認して」 「RDS スナップショットを作成して」
→ 自然言語で調査、依頼
- 3 **障害分析・SLO 監視・セキュリティ調査**
Powers で専門知識をロード
→ 高度な運用を自然言語で

(参考) Powers の中身 — 実際に何が定義されているか

POWER.md

概要 + 使い方ガイド

- 利用可能なツール一覧
- ワークフロー例
(障害調査、性能分析…)
- ベストプラクティス
- クイックスタート手順

読み込み時に即座にロード

mcp.json

MCP サーバー接続設定

- CloudWatch MCP
- Application Signals MCP
- CloudTrail MCP
- AWS Documentation MCP

読み込み時に自動接続

steering/

シナリオ別の詳細手順

- incident-response.md
- log-analysis.md
- performance-monitoring.md
- security-auditing.md
- alerting-setup.md
- observability-gap-analysis.md

必要な時に動的ロード

Power を読み込む → **知識**と**ツール接続**が即座に手に入り
詳細手順は必要な場面で自動追加 → **MCP** だけでは得られない「**知識**」がセット

出典 : github.com/kirodotdev/powers/tree/main/aws-observability

Autonomous Agent

バックグラウンドで自律的にタスクを実行

- 1 最大 10 タスク並列実行**
設計・実装・テストを同時に進行
- 2 複数リポジトリ横断**
影響範囲を自動特定し統合処理
- 3 作業メモリの保持**
過去の仕様・コミットを踏まえた提案

Name ↓	Task status ↓	Repository ↓	Source ↓	Date created ↓	Last updated ↓
Investigate reported issue for user login delays	In queue	kyleandmikesrazyadventure/docker-file-example	Kiro	6 seconds ago	-
Move Timely and Differential-Dataflow Dependencies to Hydro-Deps Repo	In progress	2 repos	Kiro	1 minutes ago	1 minutes ago
The settings page is broken on mobile	Completed	-	GitHub	11 hours ago	10 hours ago
Refactor the sign-in flow	Completed	kyleandmikesrazyadventure/shopping-cart	Kiro	1 days ago	1 days ago
Implement Database + Front-End For TODO App	Completed	kyleandmikesrazyadventure/docker-file-example	Kiro	2 days ago	2 days ago
Upgrade the dependencies for these repos	Completed	6 repos	Kiro	2 days ago	2 days ago
Implement Database And Front-End For Simple TODO App	Cancelled	kyleandmikesrazyadventure/docker-file-example	Kiro	2 days ago	2 days ago
Build issue	Completed	-	GitHub	2 days ago	2 days ago

Autonomous Agent タスクダッシュボード

Kiro Enterprise — 組織で安心して使える



認証・チーム管理

IAM Identity Center で SSO ログイン | サブスクリプション・利用状況をダッシュボードで管理



ガバナンス

MCP サーバーの管理者制御

組織で使える MCP を管理者が指定
未承認の外部接続を防止

AI モデルの固定

使用モデルを管理者がポリシーで指定



セキュリティ・データプライバシー

コンプライアンス

VPC エンドポイント（閉域利用）
ログ記録（S3）

データプライバシー

コードはサービス改善に不使用
(Free Tierの場合、オプトアウト設定を有効化することで不使用)

まとめ

- 1 AI コーディングは「**どう信頼して使いこなすか**」の時代
- 2 Kiro は「**自然言語に精度をもたらず**」ために生まれた
- 3 **仕様駆動開発**で、まず合意してから作る。品質を仕組みで担保
- 4 日常業務から **AWS インフラ構築・運用**まで幅広く活用可能
- 5 Kiro は「**一緒に考え、一緒に作るパートナー**」

では、ハンズオンで実際に体験しましょう



Thank you!

