

AIエージェントは「開発ツール」ではない

AI駆動開発がもたらす「人月モデルの終焉」とIT組織の変革

ULSコンサルティング株式会社
取締役副社長
桜井 賢一

 **ULS** Consulting



桜井 賢一

ULSコンサルティング株式会社

取締役副社長

- 銀行子会社にてメインフレーム/COBOLからキャリアスタート。証券会社向けシステム開発に20年以上にわたって従事。Java/EJBの黎明期から、リアルタイムデータの非同期処理やインメモリを活用した高速処理を活用したシステムを多数構築。
- 現在はAI駆動開発推進の責任者として、エンタープライズ領域でのDevin導入を牽引。
- 発起人としてAI駆動開発コンソーシアムを設立に参加。



世界初の自律型AIソフトウェアエンジニア



自ら考え、調べ、開発タスクを実行するAIエージェント



設計 (Design)

システム設計、
アーキテクチャ策定



コーディング (Coding)

コードの生成、実装、
リファクタリング



テスト (Testing)

ユニットテスト、
バグ検出、デバッグ



デプロイ (Deployment)

環境構築、デプロイ
メントの自動化

既存システムの調査、ドキュメント作成、各種ツール操作までカバー

自律型AIソフトウェアエンジニア Devinの開発



チャットによるタスク指示
(プロンプト)

仕事を「任せて」
別の作業

成果物をレビュー
OKならマージ



 Devin

指示に従って作業計画を立てる

設計作業

指示を受けたら
あとは「自律的に」
作業を進める

テスト&デバッグ

成果物をアップ
(プルリクエスト)



Press Release



2025 年 5 月 28 日
ウルシステムズ株式会社
Cognition AI, Inc.

ウルシステムズと米 Cognition AI、「Devin」を日本のエンタープライズ市場に共同展開
～AI 駆動開発で企業のシステム内製化を促進～

ウルシステムズ株式会社（本社：東京都中央区、代表取締役社長：横山芳成、以下、ウルシステムズ）と米国 Cognition AI 社（本社：米国サンフランシスコ、CEO：Scott Wu、以下 Cognition AI）はこのほどパートナーシップの締結に合意しました。これを受けてウルシステムズは本日より Cognition AI の自律型 AI エンジン「Devin」を日本国内のエンタープライズ市場向けに展開してまいります。合わせて大規模開発プロジェクトへの AI 導入に関するプロフェッショナルサービスを提供します。

Cognition AI が 2024 年 12 月に一般公開した Devin は、人間の開発者と同様に開発作業を行う AI エンジンです。自然言語による作業依頼を理解し、開発の段取りを考え、必要な各種ツールを操作し、設計やコーディング、テスト、デプロイなどを自動で行います。既存システムの調査やドキュメントの作成にも対応できます。Devin は、これまで人手に依存していたシステム開発作業のあり方を一新し、自動化により少数精鋭チームでの大規模システム開発を可能にします。

ウルシステムズは創業以来、先進技術を用いた大規模なエンタープライズ開発を発注企業側に立って推進してまいりました。直近では本格的な生成 AI の業務導入に注力しており、Devin を活用したシステム開発の内製化支援でも実績を重ねています。いまだ人月ベースの受託開発モデルを脱しきれていない日本国内の業務システム開発のあり方を刷新し、新しい内製開発モデルを推進するため、この度 Cognition AI とのパートナーシップを発表するに至りました。

ウルシステムズと Cognition AI は、国内の企業を対象として以下のプロフェッショナルサービスを



事例 | みずほ証券株式会社 様

国内大手金融初「Devin」を大規模導入。金融機関の厳格なセキュリティ基準をクリア。



事業内容	金融
従業員規模	5,000～10,000名
支援部門	情報システム・DX推進部門
支援内容	セキュリティ基準適合支援、 専用環境構築、ユースケース定義、開発プロセス再定義

依頼の背景

- 中期経営計画にて「生成AIによる生産性向上」を重要テーマに掲げ、AI駆動開発に注力
- ビジネスニーズへの機敏な対応と、慢性的なエンジニア不足の解消が急務
- 国内金融機関としての厳格なセキュリティ基準のクリアや、既存開発フローへの適合に専門的な知見を必要としていた

ULSコンサルティングが実施したこと

- セキュリティ・インフラ構築支援：提供元(Cognition AI社)と連携し、金融機関の厳しい基準に適合する専用環境の構築。
- 実用性検証(PoC)の伴走：2025年9月からのトライアルにおいて、具体的な実用性の確認をサポート。
- 開発プロセスの抜本的見直し：生成AI導入による効果を最大化するためのユースケース提示と、従来の開発フローの再設計。
- ガイドライン整備の着手：2026年4月の本格運用に向けた、利用ルールの策定支援。

お客様の声

- 「ULSコンサルティングのサポートなくして、今回のDevin導入は困難だったと考えています。今後、開発プロセス全般の見直しや実際のプロジェクトへの適用など、AI駆動開発の推進に必要な活動全般をサポートいただくことを期待しています。」
(みずほ証券株式会社 IT・システムグループ 上級技術統括 杉谷剛 氏)

多くの日本企業の皆様にDevinをご利用いただいています

プレスリリース 2025年06月17日

ウルシステムズ、ぴあ内製開発チームにおける「Devin」の活用を支援

～自律型AIソフトウェアエンジニアをシステム開発実務に導入～

ウルシステムズ株式会社（本社：東京都中央区、代表取締役社長：横山芳成、以下、ウルシステムズ）はこの度、ぴあ株式会社（本社：東京都港区、代表取締役社長：年内廣、以下、ぴあ）のシステム内製チームにおける自律型AIソフトウェアエンジニア「Devin」の活用を支援しました。

プレスリリース 2025年12月24日

ULSグループ、札幌市の基幹システム標準化プロジェクトで「Devin」を活用

～150万ステップのテストコード書き換え工数を約1/4に削減、公共事業への生成AI適用を加速～

ULSグループ（本社：東京都中央区、代表取締役社長：横山芳成、以下、ULSグループ）はこの度、札幌市の基幹システム標準化プロジェクトで「Devin」を活用し、150万ステップのテストコード書き換え工数を約1/4に削減、公共事業への生成AI適用を加速しました。

プレスリリース 2026年01月22日

ULSコンサルティング、みずほ証券のAI駆動開発推進を支援

～大手金融機関として国内初となる大規模Devin導入プロジェクトを全面的にサポート～

ULSコンサルティング株式会社（本社：東京都中央区、代表取締役社長：横山芳成、以下、ULSコンサルティング）はこの度、みずほ証券株式会社（本社：東京都千代田区、取締役社長：浜本吉郎、以下、みずほ証券）の自律型AIソフトウェアエンジニア「Devin」導入を支援しました。

プレスリリース 2026年07月16日

ULSコンサルティング、サントリーホールディングスにAI駆動開発基盤「Cognition Platform」を導入

自律型AIソフトウェアエンジニア「Devin Cloud」、AIエージェント搭載の次世代IDE「Devin Desktop」を駆使してAI駆動開発を加速

ULSコンサルティング株式会社（本社：東京都中央区、代表取締役社長：横山芳成、以下、ULSコンサルティング）はこのほどサントリーホールディングス株式会社（本社：大阪府大阪市、代表取締役社長：鳥井信宏、以下、サントリー）に米Cognition AIが提供するAI駆動開発基盤「Cognition Platform」を導入したことをお知らせします。

商談数



200社以上

パイロット
プロジェクト



30件以上

エンタープライズライセンス
導入実績



10社以上

弊社開発プロジェクトでの成功事例

巨大基幹システム
Javaバージョンアップ

150万ステップの
テストコードを書換え

見積りは
200人月
半年以上

工数

200人月超



50人月

75%削減

期間

半年以上



2.5ヶ月

半分に短縮

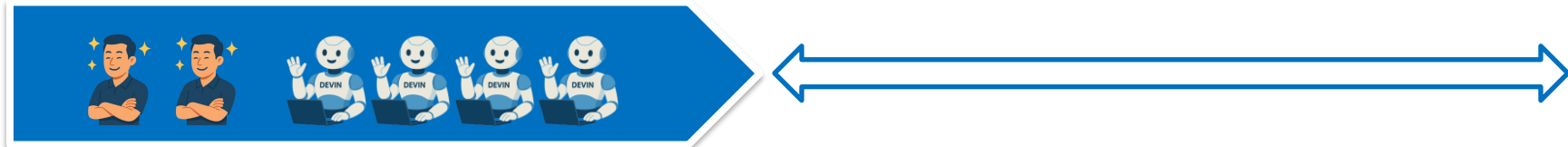
AI駆動開発による工数の削減、工期の短縮

AIへの指示をしっかりと与えることができれば
最も工数のかかる実装部分は大幅に短縮できる

従来の開発



AI駆動開発



IT業界で長らく続いてきた「人月 × 単価」による
コストモデルは崩壊！

AI導入は進んでいるが、成果につながっているのか



導入・利用は進んでいる

- トークンコストは毎月増加
- 最新のフロンティアモデルを常時利用
- 現場からは「便利」「ないと困る」



本来期待していた成果

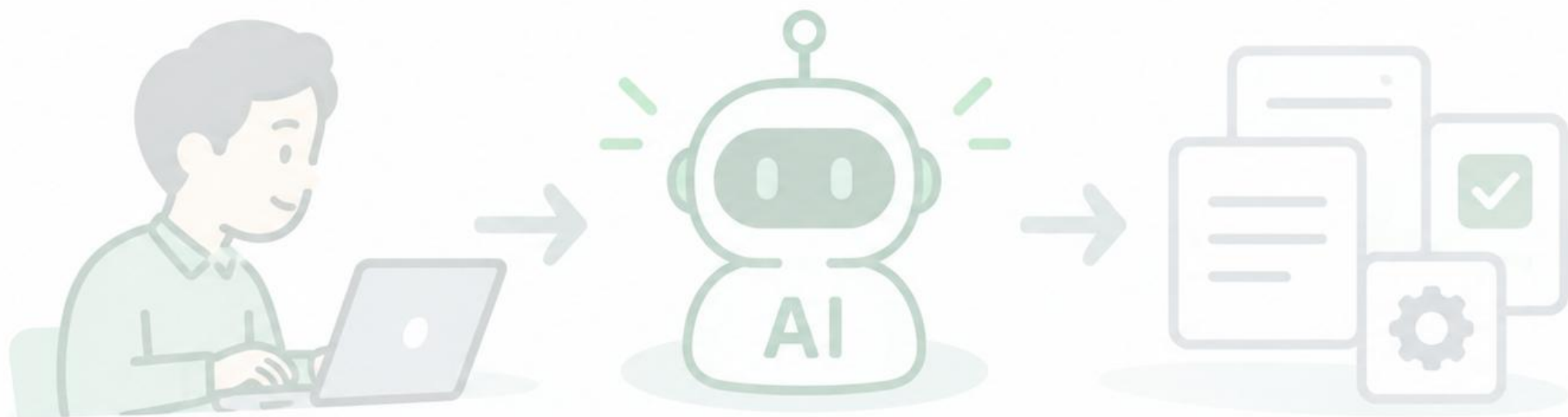
- 開発期間の短縮
- 必要人数の最適化
- 開発コストの削減
- 内製化の進展によるブラックボックス解消

「使われている」と「成果が出ている」は違う

個人の生産性向上だけでは、経営成果にはつながらない

なぜ、思ったほど成果がでないのか？

なぜ、思ったほど成果が出ないのか？



AIエージェントを 「開発ツール」として導入しているから

個人の生産性向上にとどまり、チーム・組織全体の生産性向上まで設計できていない

AIエージェントを「開発ツール」から「チームメンバー」へ

開発ツールとして使う

これまでのAIツールは
エンジニア一人ひとりのアシスタント

エンジニア個人がAIを使う
エンジニア個人の生産性向上
ローカルPC上でスケールアップ



開発リソースとして使う

Devinは仕事を任せられる
「チームメンバー」



AIに仕事を任せる
チームの処理能力を高める
クラウド上でスケールアウト



なぜ、思ったほど成果がでないのか？



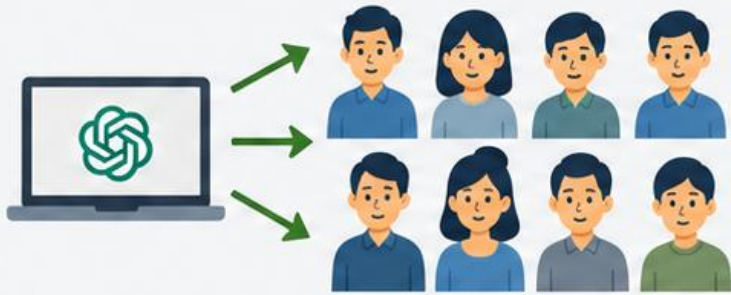
AI駆動開発導入時 ハマりがちな3つのパターン

AI駆動開発導入時にハマりがちな3つのパターン

開発ツールのつもりで導入するとハマります

⚠️ ① 全員配布の罠

「まず全員使えるようにする」
組織として使い方を決めない



起きること

- 使い方がバラバラ
- トークン消費だけ増える
- 成果が見えにくい

⚠️ ② コーディング偏重の罠

新規開発・実装工程だけに注目
工数のかかりどころを見ていない



起きること

- ブラウンフィールドの工数が減らない
- 調査・設計・テストが残る
- 開発全体は速くならない

⚠️ ③ エース最適化の罠

一番できるチームに選定を任せる
個人技のツール・設定に依存



起きること

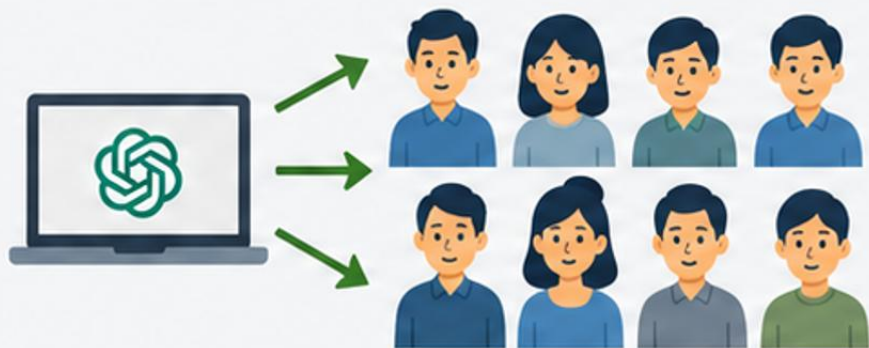
- 他チームがついてこれられない
- 全社展開できない
- 再現性がない

①全員配布の罠



① 全員配布の罠

「まず全員使えるようにする」
組織として使い方を決めない



起きること

- 使い方がバラバラ
- トークン消費だけ増える
- 成果が見えにくい

ハマりポイント

- 全員にアカウントを配る
- IDEやCLIツールの感覚で「自由に使って」と渡す
- 組織横断の権限・監査・利用量管理は利用拡大してから考える

起きること

- 使い方がバラバラ。成功ノウハウが**個人に蓄積されるだけ**で、チームをまたいだ標準化・横展開が進まない
- ローカルで動くAIエージェントの場合、利用環境を統制できない。
- 誰がどれだけ使っているのかわからない
 - 使っていないのかもしれない
 - 無駄な使い方をしているのかもしれない
- **トークンコスト爆発**問題
- 「便利だった」「コードが速く書けた」という感想しか残らず、投資対効果や全社展開の判断ができない

なぜ個人の生産性があがっても労働時間が減らないのか？



AIで個人の作業は速くなる

AIの活用により、これまで時間のかかっていた実装やテストなどが短時間で終わる



空いた時間は別の仕事で埋まる

- 不要な調査や会議、書類作成が増える
- 仕事量は与えられた時間まで膨張する

パーキンソンの法則



AIを良く使うための仕事が増える

- AIの出力をさらに良くする
- プロンプトや指示を作り込む
- 結局、締め切りまで仕事をしてしまう

②コーディング偏重の罠

⚠ ② コーディング偏重の罠

新規開発・実装工程だけに注目
工数のかかりどころを見ていない



起きること

- ブラウンフィールドの工数が減らない
- 調査・設計・テストが残る
- 開発全体は速くならない

ハマリポイント

- **新規開発・実装だけ**で評価
- 「コードを速く書く」ことだけをKPIとして追及
- バイブコーディングにより、ユーザーがどんどんアプリを製造

起きること

- 調査・設計・テストなどSDLC全体のボトルネックが残る
→ コード生成は速くなった
→ 一方、設計・テストなどの工程は**既存のプロセスを踏襲**
- 野良アプリの乱立

「新規システムの開発にAIを活用して3か月から1か月に短縮した。
これは確かに分かりやすい。」

毎年100人で保守しているシステム群を、80人相当の仕事量で
回せるようになった。
派手ではないが、企業経営から見るとインパクトは非常に大きい」

「いま大量の時間を使っている仕事は何か」を見るべき



③エース最適化の罠



③ エース最適化の罠

一番できるチームに選定を任せる
個人技のツール・設定に依存



起きること

- 他チームがついてこれられない
- 全社展開できない
- 再現性がない

ハマりポイント

- 技術力の高いトップエンジニアに評価を任せる
「一番イケてるツールを決めてくれ！」
- エースメンバーで上手くいった手法を標準にする

起きること

- 優秀な個人・チームに最適化した設定、ワークフローを作りこむ
- PoCや先行プロジェクトでは成果が出るが、他チームに展開すると**成果が再現できない**



成果を出している企業は
Devinどのように利用しているのか？

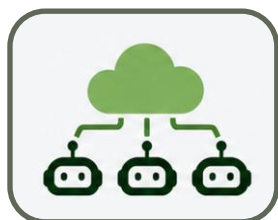


エンタープライズITでDevinが選ばれる3つの理由

実装だけでは成果が出ない。個人利用ではスケールしない。全社展開すると統制できない。
Devinはこの3つの壁に対応します



SDLC全体を支える強力な機能



Devinに仕事を任せる「仕組み」



全社利用の統制・管理

SDLC全体を支える強力な機能

DevinはただのコーディングAIじゃない
エンタープライズのSDLCを支える強力な機能

単なるコーディングAIではない SDLC全体をサポートする機能

影響調査から、実装・テスト、そして障害発生時の自動対応まで。SDLC(ソフトウェア開発ライフサイクル)のあらゆる工程でDevinが自律的にタスクを実行し、開発プロジェクトを圧倒的なスピードで推進します。

分析・設計

Deep Wiki

既存のコードベースからドキュメントを生成。
多様な言語をサポートし、ドキュメントの存在しないプロジェクトの詳細を把握。

Ask Devin

コードベースの理解に基づいて問合せに回答。
新規開発における影響調査など開発開始前の計画・設計を支援。

開発

Devin Cloud Sessions

自律的な開発エージェントがクラウド上のセキュアな環境で開発を実施。
Linux・Windows・Macに対応

Devin Desktop & Devin CLI

ローカルPC上で対話形式の開発支援。ローカルとクラウドのエージェント群を一括して管理。

検証

Devin Review

AIによるコードレビューにより、品質を向上。
バグを見つけ、フィードバックを受けたセッションが自動修正

E2E Test

コードをエージェントのVMにデプロイし、UIテスト結果をビデオとスクリーンショットで提供

運用

Automation

インシデントやセキュリティースキャンのイベントをトリガーに、開発エージェントがコードを自動修正

エンタープライズITの主戦場は、手ごわい既存システム

- 30年前からずっと動いている
- 大きな改修はないが業務上重要なシステム
- 属人化しているが、仕様の全貌は誰も分らない
- ドキュメントが最新とは限らない
- 「The 現行踏襲」
- COBOLで書かれている
- VB6で書かれている
- 知らない言語で書かれている
- 数千万ステップ！
- 改修するときは副作用が怖いからコピー
- 使っていないコードも多数ある(が消してよいかどうかわからない)
- 定期的にバージョンアップ地獄がやってくる
- テストはすべて手動



どうしてレガシーシステムの改修はコストがかかるのか？

レガシー改修は、「コード生成」ではなく「調査・コード理解」から始まる

IPA 改良開発案件の工程別工数比率（平均）

工程	工数比率	主な作業目的
基本設計	16.4%	何を直すかを決める
詳細設計	16.1%	影響範囲を具体化する
制作	30.4%	実際にコードを修正する
結合テスト	20.9%	つながりを確認する
総合テスト	16.3%	全体として壊れていないか検証する

出典：IPA「ソフトウェア開発分析データ集2022」
改良開発580件の工程別工数比率（平均値）よりULSConsulting作成。

エンタープライズに効くDevin：DeepWiki

ソースコードを読み込み、システム構成に関するドキュメントを自動生成

概要

はじめに

システム構成

データベーススキーマ

バックエンドAPI

設定

認証システム

APIエンドポイント

費目API

記帳API

フロントエンドアプリケーション

アプリケーションセットアップ

認証とルーティング

UIページ

ダッシュボード

費目管理

記帳管理

スタイルとデザイン

デプロイとコンテナ化

概要

はじめに

システム構成

データベーススキーマ

バックエンドAPI

設定

認証システム

APIエンドポイント

費目API

記帳API

フロントエンドアプリケーション

アプリケーションセットアップ

認証とルーティング

UIページ

ダッシュボード

費目管理

記帳管理

スタイルとデザイン

デプロイとコンテナ化

概要

はじめに

システム構成

データベーススキーマ

バックエンドAPI

設定

認証システム

APIエンドポイント

費目API

記帳API

フロントエンドアプリケーション

アプリケーションセットアップ

認証とルーティング

UIページ

ダッシュボード

費目管理

記帳管理

スタイルとデザイン

デプロイとコンテナ化

概要

はじめに

システム構成

データベーススキーマ

バックエンドAPI

設定

認証システム

APIエンドポイント

費目API

記帳API

フロントエンドアプリケーション

アプリケーションセットアップ

認証とルーティング

UIページ

ダッシュボード

費目管理

記帳管理

スタイルとデザイン

デプロイとコンテナ化

概要

目的と範囲

このドキュメントは、Google OAuth認証を使用した家計簿WebアプリケーションであるAccountingBookアプリケーションの紹介を提供します。アプリケーションの目的、主要機能、および高レベルの3層アーキテクチャについて説明します。

詳細なセットアップ手順については、はじめにを参照してください。アーキテクチャの詳細については、システム構成を参照してください。特定のコンポーネントの情報については、バックエンドAPIおよびフロントエンドアプリケーションを参照してください。

AccountingBookとは

AccountingBook（家計簿アプリ）は、ユーザーがGoogleアカウントを使用して認証し、収入と支出を記録できるWebベースの家計簿アプリケーションです。このアプリケーションはDockerを使用してコンテナ化されており、フロントエンド、バックエンド、データベースの各層が分離された従来の3層アーキテクチャパターンに従っています。

出典: README.md 1-3

主要機能

アプリケーションは以下の機能を提供します：

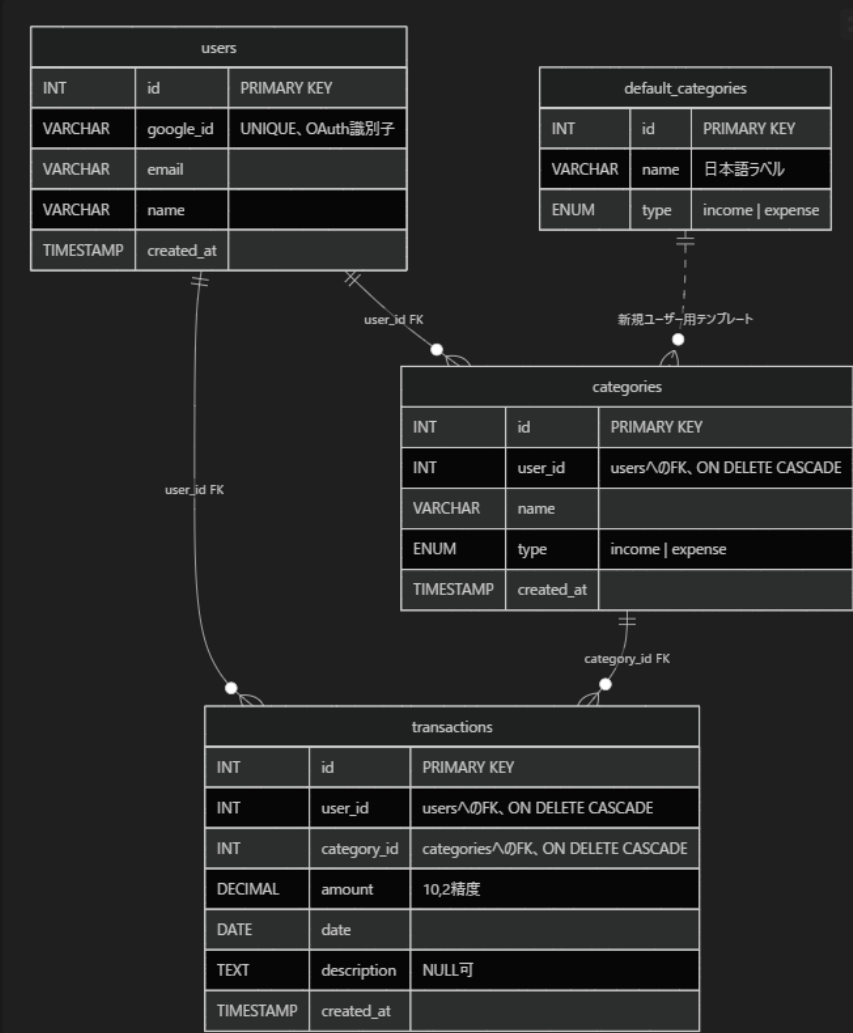
機能	説明	関連ページ
Google OAuth認証	ユーザーはGoogle OAuth 2.0を介して認証。パスワード管理は不要	認証システム
記帳管理	日付、費目、金額、説明を含む収入/支出の記帳を作成、読み取り、更新、削除	記帳API、記帳管理
費目管理	カスタマイズ可能な収入/支出の費目を使用して記帳を整理	費目API、費目管理
財務ダッシュボード	収入、支出、純残高の集計と最近の記帳履歴を表示	ダッシュボード
マルチテナントデータ分離	各ユーザーのデータは完全に分離。ユーザーは自分のレコードのみにアクセス可能	データベーススキーマ

出典: README.md 5-11

技術スタックとアーキテクチャ概要

3層アーキテクチャ

データベーススキーマとテーブルのマッピング



エンタープライズに効くDevin : Ask Devin

自然言語による質問から、複数リポジトリを横断してソースコードを調査

社員マスタの更新処理を行っている箇所はどこ？

Fast

Searched across devin-usage-guide, devin-test, AccountingBook, asset-lending-control-app, benchbase, customer-tutorial, esiritori, golang-gin-realworld-example-app, ksk-core-structure-refined, mahjong-score-app, nextjs-python-template, physical, project-from-design-docs, selfhost-gitlab, recruitment-legacy, survey-devin-data-analysis, tc2devin-backstage, terraform_deployment, TodoAppForDevin, workflow-template, gitlab-devin

Go deeper

提供されたコードベースを確認した結果、社員マスタの更新処理を直接行っている箇所は見つかりませんでした。

社員マスタ関連のコード

uls-devin-sandbox/recruitment-legacy リポジトリで社員マスタに関連するコードを発見しましたが、これは更新処理ではなく参照処理のみです：

EmployeeDao.java

src/main/java/jp/co/ressy/dao/EmployeeDao.java では、社員マスタ（EMPLOYEE テーブル）から採用担当者のメールアドレスを取得する処理のみが実装されています。 EmployeeDao.java:17-30

このクラスには getEmails メソッドしか存在せず、INSERT、UPDATE、DELETE などの更新処理は含まれていません。 EmployeeDao.java:12-15

他のリポジトリでの更新処理例

参考として、他のリポジトリでは以下のような更新処理が実装されています：

ksk-core-structure-refined: ユーザー情報の更新処理 repository_test.go:113-151

uls-devin-sandbox/recruitment-legacy src/main/java/jp/co/ressy/dao/EmployeeDao.java

9

10 // 各メソッドは、必要な変数に加えてConnectionをserviceから受け取って使ってください。

11

12 /**

13 * 社員マスタに接続するEmployeeDaoクラスです。

14 */

15 public class EmployeeDao {

16

17 public List<String> getEmails(Connection connection) throws SQLException {

18 String sql = "SELECT EMAIL FROM EMPLOYEE WHERE IS_RECRUITER = 1;";

19

20 List<String> emails = new ArrayList<>();

21

22 try (Statement statement = connection.createStatement();

23 ResultSet resultSet = statement.executeQuery(sql);) {

24 while (resultSet.next()) {

25 emails.add(resultSet.getString("EMAIL"));

26 }

27 }

28

29 return emails;

30 }

31 }

32

uls-devin-sandbox/ksk-core-structure-refined back/service/user/repository/userRepository/repository_test.go

110 assert.Equal(t, string(obj.ID()), record[0].ID)

111 })

112

113 t.Run("既存のマッチ結果を更新できる", func(t *testing.T) {

114 // 準備

115 tran := db.DBTransactor()

116 defer tran.Rollback()

Devinに仕事を任せる「仕組み」

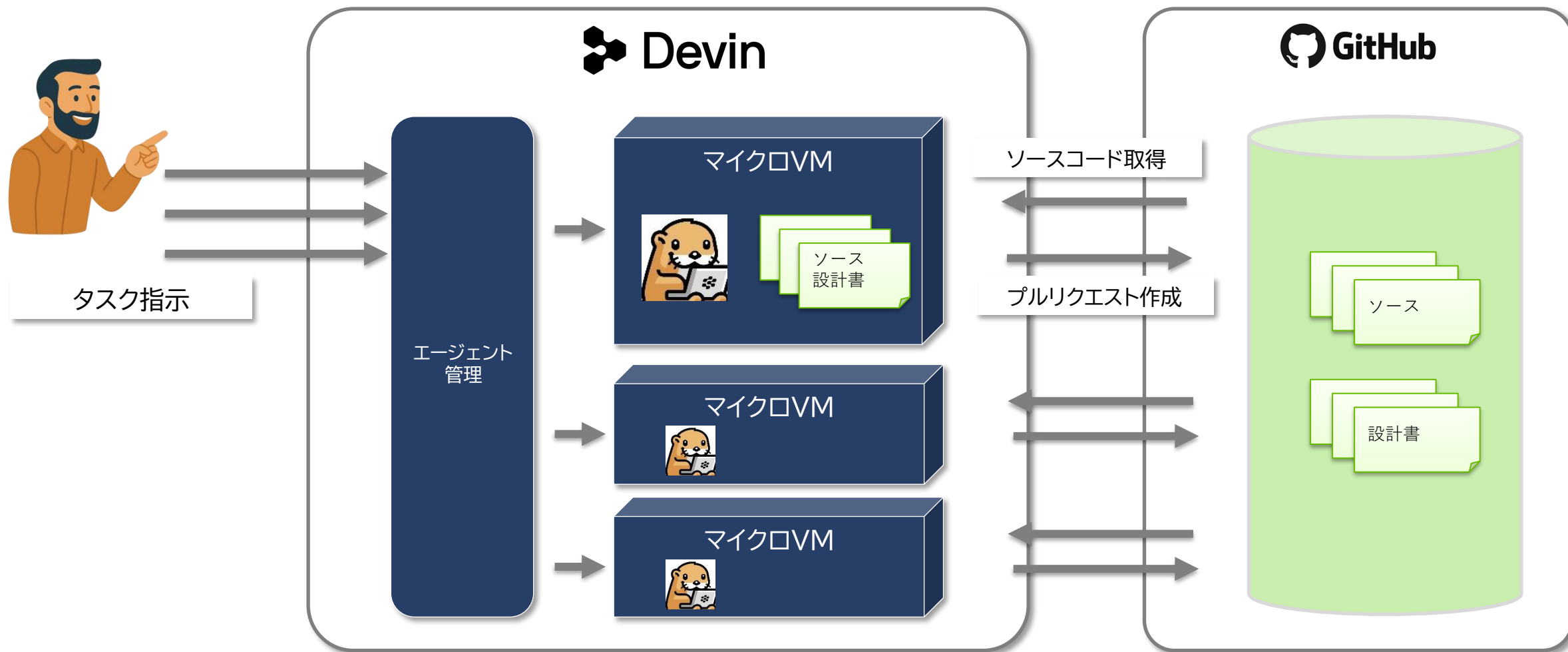
Devinに企業の知識やルールを伝え、
自律的に仕事ができる「チームメンバー」へ

クラウド上で無限にスケールする 凄腕エンジニア集団



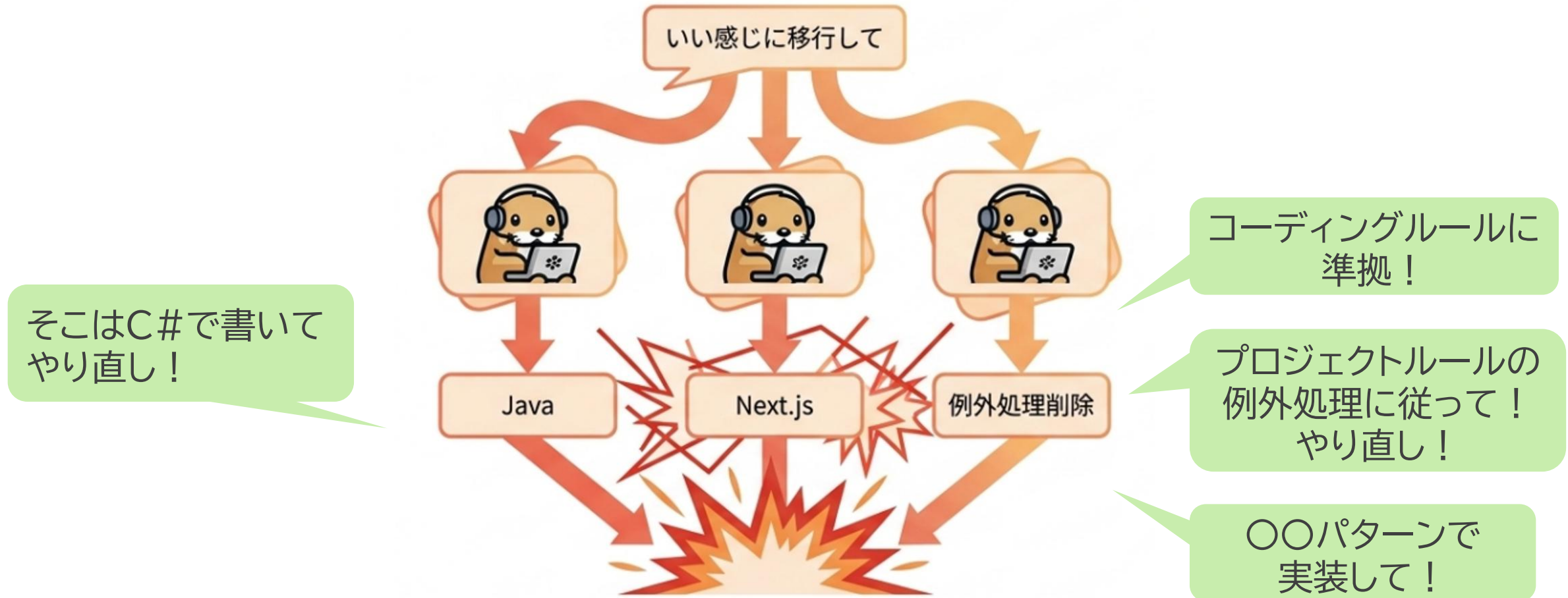
Devinはクラウド上でスケールアウト＝増員できる

タスクを指示すると、クラウド上にAIエンジニアと開発環境がセットになった「マイクロVM」が立ち上がります。マイクロVMは同時にいくつでも起動できるため、複数タスクを並列に処理させることができます



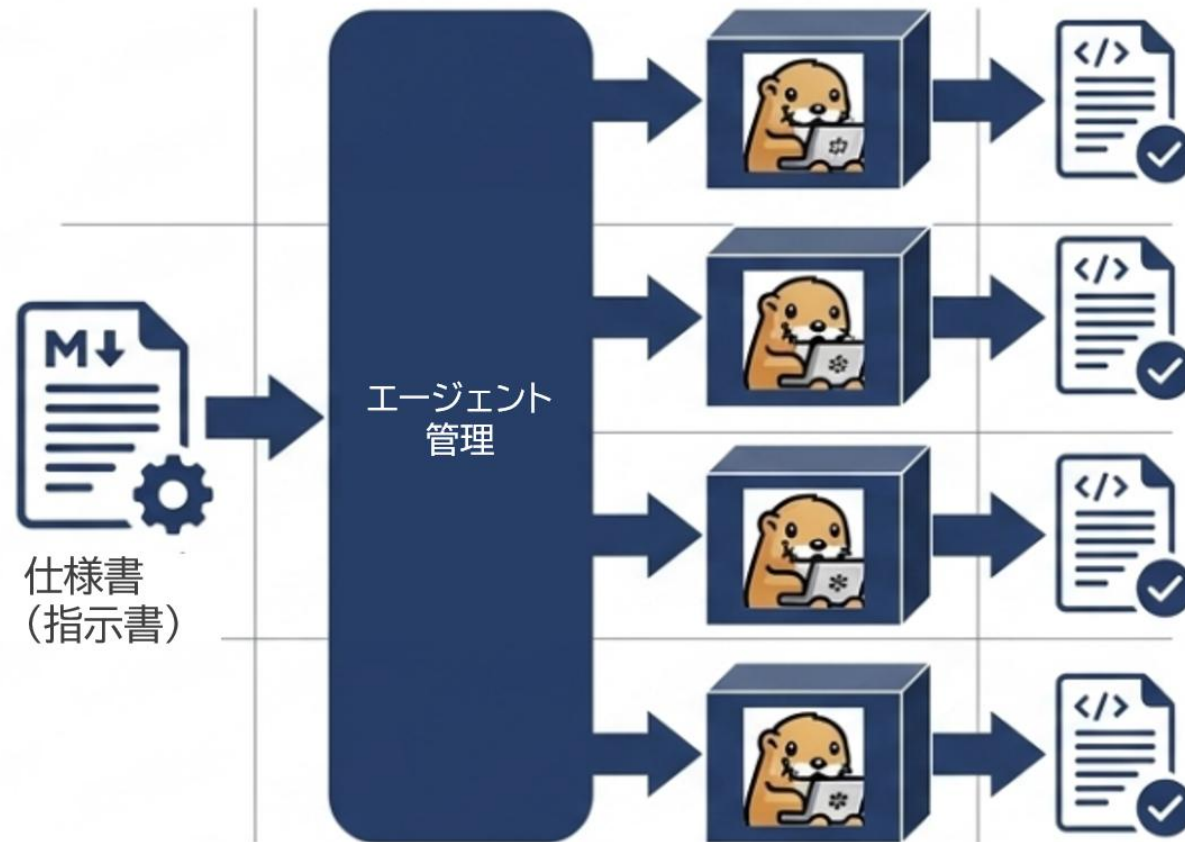
「自律的」であることの落とし穴

Devinは、できるだけ指示を受けずに仕事を進めるようにチューニング
指示があいまいなら、自分で「行間を推測」して進める



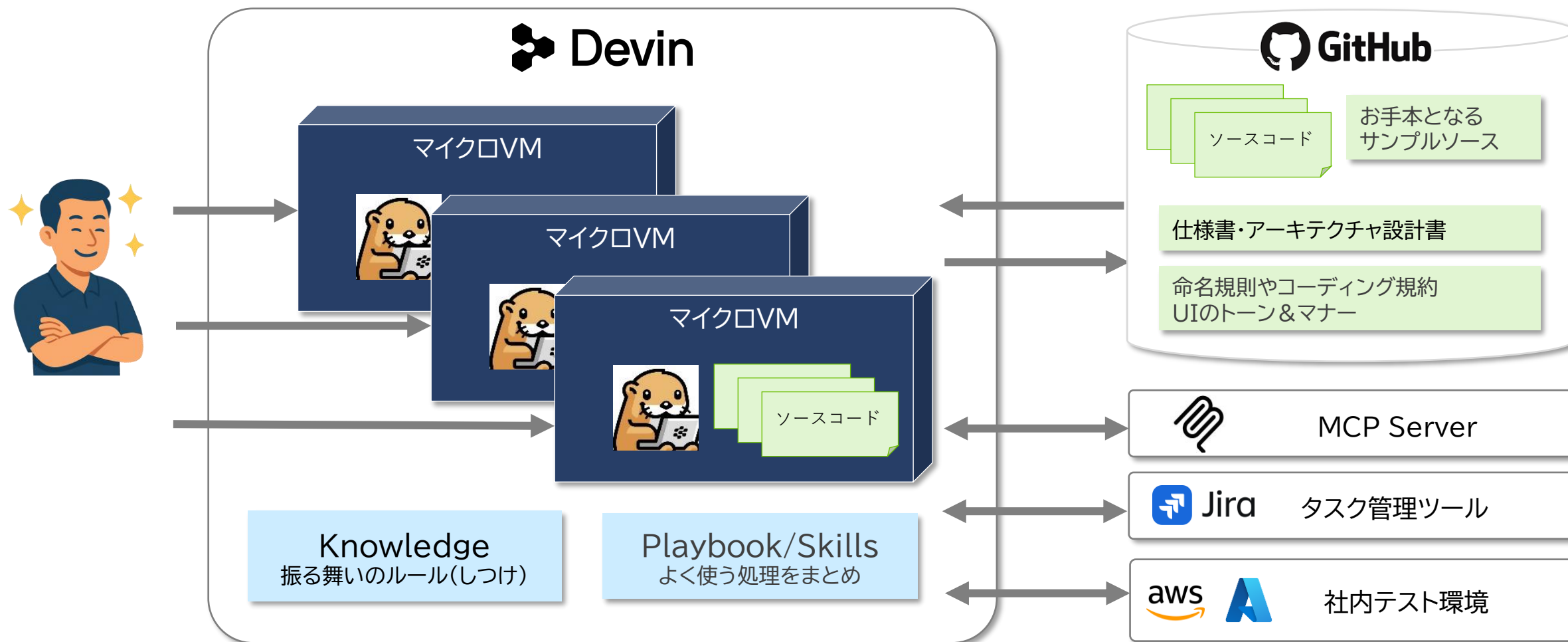
自律型AIソフトウェアエンジニア Devinに「仕事を任せる」

必要最低限の指示一発で、欲しかった成果物を出す
これができたら「仕事を任せる」ことができる



Devinの自律性を実現するプラットフォーム

仕事を任せるための企業の知識・作業手順・ツール・実行環境を整備
ハーネスを構築、運用するためのプラットフォーム



全社利用の統制・管理

数百人単位の組織でDevinを活用する
セキュリティとガバナンス

Devinを組織で運用するための管理機能

当初から「大規模組織で運用する」ことを中心に設計されています。
SOC2準拠の世界水準のセキュリティとデータ管理体制



統制・ガバナンス

安全に、効率的に、AIを活用するための管理機能



SSO / RBAC

シングルサインオン・ロールベースの権限管理



シングルテナント

自社専用のセキュアな実行完了を構築



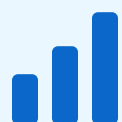
監査ログ

誰が・何を・いつ実行したか



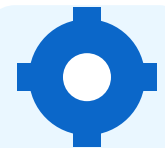
Outpost

「データは自社から出さない」



ACU管理

利用量チェック・上限管理・コストの可視化



Organization

プロジェクト単位でのノウハウの共有

個人の生産性向上から、組織の開発力向上へ

エンタープライズITに必要なのは、最も賢いコーディングAIではなく、組織として運用できるAIプラットフォーム

Devinが選ばれる3つの理由

1 SDLC全体を担う

コードを書くだけでなく、
開発全体をAI化



- 調査・コード理解
- 影響分析・計画
- 実装・テスト・レビュー
- 既存システム・
ブラウнフィールドにも対応

2 自律・並列に働く

クラウド上のAIエージェントが
プロフェッショナルチームとして動く



- 非同期・自律実行
- 複数エージェントの並列稼働
- ハーネスによる安全な実行環境
- 1~6時間の中〜大タスクを委譲
- 人はレビューと意思決定に集中

3 全社で統制する

多数のAIエージェントを
安全に、組織的に活用できる



	SSO / RBAC
	監査ログ・追跡性
	利用量管理 (ACU)
	複数チーム・複数プロジェクトの統制
	VPC / セキュリティ要件対応

開発プロセスを「実装して」全社に展開する

AIエージェントと協働する「新しい業務フロー」
標準化された開発プロセスを実装して全社へ展開

AIDD CoEの役割



標準化

ベストプラクティスを整備し
企業の『開発の型』を策定



展開

ハーネスを整備し
各チームへ横展開



教育

AI駆動開発のスキル・
ノウハウを育成



改善

運用から得た知見を反映し
継続的にアップデート

各チームへの定着を支援

コンテキスト（共通仕様・設計）

AIが前提として読む共通仕様・設計原則・用語を、書いて置いておく

品質ゲート

レビュー観点をチェックリスト化し、
AIに一次判定させる

手順書（Playbook / Skill）

工程・承認・分岐を書き下し、
毎回同じ手順で走らせる



チームA



チームB



チームC

システム開発の世界にもようやくAXが！
エンタープライズITが大きく変わる



2か月間のDevin スターターパック



EnterpriseライセンスとPoCコンサルティングをセット
Devinをすぐに活用するためのパッケージです

500万円～

環境構築

- 貴社専用のDevin環境を準備します

検証

- 事前に検証ユースケースとKPIを設定。
- PoC実施により導入効果を検証します。

トレーニング

- お客様がDevinを使用できるようになるためのトレーニングを行います

結果報告

- PoCの成果をレポートニング
- 本格導入に向けたポイントをご提案します。

AI駆動開発導入のご支援

企業様ごとに現状の課題感とAI駆動開発に対する期待は異なりますが、まずは効果のありそうな個別システムをパイロットとして選定し、PoCからスタートすることが一般的です。

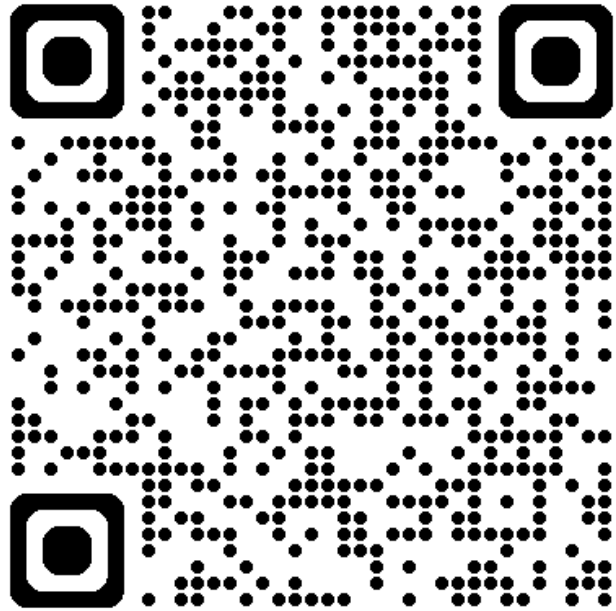
レベル	個別システムで試す	プロセスを作る	組織を変革する
期待効果	- 開発工数の削減 - 開発リードタイム（工期）の短縮 - ドキュメント生成	- AI駆動ベース開発プロセス構築 - ドキュメント/アーキテクチャ標準化	- AI駆動開発を基盤とした内製化体制の強化 - IT部門のコスト構造とベンダー体制の抜本的な見直し
課題	- ベンダー見積が予算の数倍になることがある 「3月末までに終わらせねばならない開発が終わらない」といった事態が発生 - ドキュメント不足で、正しい仕様が分からない	- システムが多岐にわたり、開発プロセスがバラバラ - 各自が承認なしにAIツールを無秩序に使用し、ガバナンスが機能していない	- IT部門がシステムを十分に把握できず「ベンダーがいないと何も分からない」状態 - システム増大により、数百人規模のベンダー体制化 → ITコストが増大



DevinではじめるAI駆動開発 ～AIエージェントがもたらす開発現場の革命～

出版社 : 日経BP
発売日 : 2026/7/24
著者 : UJSコンサルティング

Contents	
第1章 Devinとは何か └ AIエンジニアに仕事を任せるという選択肢	第6章 ハーネスエンジニアリング └ 注意力ではなく仕組みで品質を支える
第2章 Devinの機能と使い方 └ 機能を知り、任せ方を設計する	第7章 チーム運用 └ 個人から組織へスケールする
第3章 プロンプトエンジニアリング └ 「指示は設計である」を実務に落とす	終章 バイブコーディングとの共存
第4章 セッション実務 └ 仕様を書いて、渡して、検証する	付録A ユースケースカタログ (実践パターン集)
第5章 コンテキストエンジニアリング └ Devinに渡す情報を設計する	付録B 統制設計ガイド (セキュリティ・コンプライアンス)
	付録C 意思決定のための指標と運用 (KPI・ROI)



<https://www.ulsconsulting.co.jp/devin/>

 **ULS** Consulting



 **ULS** Consulting

となりで、
かなえていく。

