

# asagami AI × Cursor Rules 連携提案

## 学習データに基づく適応型開発環境の構築

### asagami AI

- ✓ ノートから問題を自動生成
- ✓ 理解度の正確なチェック機能
- ✓ 部門/プロジェクト別の集計分析
- ✓ 学習状況の可視化と傾向把握



### </>Cursor Rules

- ✓ コーディング規約の自動適用
- ✓ 個人に合わせたルール設定
- ✓ .cursorフォルダによる設定共有
- ✓ 開発品質の均一化と向上

### ★ 連携による主な価値

asagamiのノートから得られる **集計データ** を基に、個人の苦手分野や課題に合わせたCursor Rulesを **自動生成** することで、個人のスキルレベルに応じた「**適応型開発環境**」を構築します。これにより開発品質の向上と学習効率の最大化を同時に実現します。

## 私の考え：asagami AI × Cursor Rules 連携

### asagami AIの魅力は「ノートと問題を解く」こと

asagami AIの最大の魅力は、ノートから問題を自動生成し、それを解いてもらうことで理解度をチェックできる点です。

システム会社の部門やプロジェクトごとに、その部門やプロジェクトに関するノート管理者が従業員に問題を解いてもらい、蓄積された結果からベストプラクティスなルールも生成されるという考えがあります。

### Cursor Rulesとの連携で実現する「適応型開発環境」

この連携により、個人のスキルレベルに応じて開発環境そのものが最適化される「適応型開発環境」を構築できます。

従業員が解いた問題の結果（誰が何を苦手としているか）をCursor Rulesに送ることで、個人に合わせた「専用アシスト」が自動的に提供されるようになります。

### 「勉強して終わり」ではない継続的な成長サイクル

この連携の最大の価値は、「勉強して終わり」ではなく、勉強した結果がすぐに「明日の仕事を助けてくれる超優秀なアシスタント」を育て、そのアシスタントからのフィードバックがさらに学習を最適化するという継続的な成長サイクルを生み出すことです。

## 文化祭のカレー屋さんで例えると

「チーム専用のAI家庭教師つき教科書」と「超おせっかいなゲームの攻略アシスタント」が合体するイメージです。

### 1 asagami AI = チームの教科書メーカー

クラスのリーダーが「最高のカレーを作るための秘伝のレシピ」や「接客で絶対言ってはいけないNGワード集」を『**クラス専用ノート**』にまとめます。

クラスみんなはそのノートを読み、AIが自動で出す「**タマネギは何分炒めるのが正解？**」「**お客様に『いらっしゃいませ』の次に言うべき言葉は？**」といったクイズを解きます。

### 2 Cursor = 調理や接客を手伝う超優秀な助っ人AI

Cursorは、プログラミングを助けてくれるAIアシスタントです。文化祭の例では、「**調理や接客を手伝ってくれる超優秀な助っ人AI**」だと考えてください。

### 3 2つが合体すると魔法が起きる

クラスみんなが解いたクイズの結果（誰が何を苦手としているか）を、助っ人AI（Cursor）に教えると...

- **みんなの苦手から「チームの成功ルール」が自動生成：**

「クラスの半分以上が、隠し味のリンゴを入れるタイミングを間違えている」というデータから、調理中に「ストップ！今リンゴを入れるのは早い！レシピノートの3ページ目を確認して！」と自動でアシストしてくれます。

- **個人に合わせた「専用アシスト」：**

A君だけが「お釣りの計算をよく間違える」というデータがあれば、A君がレジに立つときだけ、「A君、注意！お釣りの計算は、もう一回確認しようね！」と特別なアドバイスをくれます。

# 高校生にもわかる説明



## asagami AI側

「みんなのテスト結果をまとめて、**苦手なところを見つける**よ。その情報を使って、クラス別の**勉強サポートプラン**（cursor rules）を作るね。」

### 具体例：数学の問題

クラス全体が「三角関数」につまずいていると検出→三角関数の問題を自動生成

- ✓ 一人ひとりの苦手に合わせて学習ができる
- ✓ 時間を効率的に使える（苦手な部分だけ集中して勉強）



## Cursor側

「実際に勉強や仕事をしているときに、**どこでつまづいたかを記録**するよ。その情報をasagami AIに送って、**もっと効果的な勉強方法**を考えてもらうね。だからまたテストやってね。」



**高校生Aさんの場合：**  
プログラミング部で Web アプリを作成中

- ① データベースの接続でエラー発生 → Cursorがエラーパターンを記録
- ② asagami AIがそのエラーに関する問題を自動生成
- ③ 部活の仲間も含めてデータベース接続の基礎から学び直せる

- ✓ 実践で見つかった問題点をすぐに学習に活かせる
- ✓ 同じ間違いを繰り返さないようになる



## 学習と実践の正のループが完成！

 **学習**

asagami AIで学ぶ



 **実践**

Cursorで実践する

→ どんどん上達して、**自分の強みが見つかる！**

# 技術的な連携方法

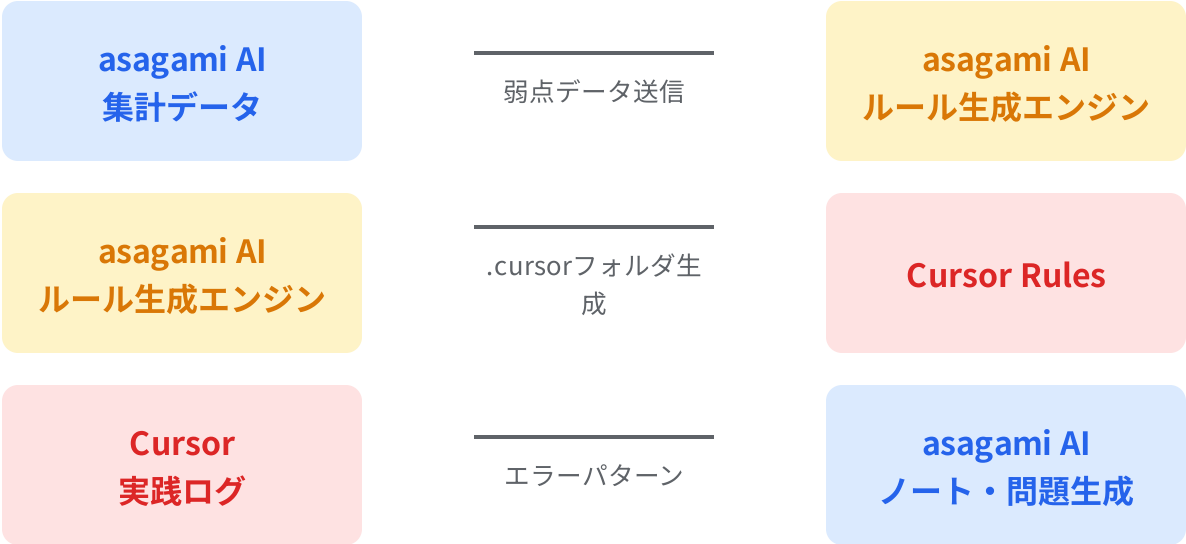
## API連携の仕様

### asagami AI → Cursor Rules

```
// 集計データの構造
{
  "department": "金融システム開発",
  "noteId": "sec-123",
  "weakPoints": [
    {
      "topic": "データ暗号化",
      "averageScore": 65,
      "problemAreas": ["鍵管理", "暗号化アルゴリズム選択"]
    }
  ],
  "individualData": [
    {
      "userId": "tanaka",
      "weakTopics": ["データ暗号化", "SQLインジェクション"]
    }
  ]
}
```

### Cursor Rules → asagami AI

## データの流れ



### 高校生にもわかる説明

**asagami AI側：**「みんなのテスト結果をまとめて、苦手なところを見つけるよ。その情報を使って、クラス別の勉強サポートプラン（cursor rules）を作るね。」

**Cursor側：**「実際に勉強や仕事をしているときに、どこでつまづいたかを記録するよ。その情報をasagami AIに送って、もっと効果的な勉強方法を考えてもらうね。だからまたテストやってね。」

# 高校の例で分かりやすく説明

## 文化祭のカレー屋さんで例えると

「チーム専用のAI家庭教師つき教科書」と「超おせっかいなゲームの攻略アシスタント」が合体するイメージです。

### 1 asagami AI = チームの教科書メーカー

クラスの担任の先生（管理者）が「最高のカレーを作るための秘伝のレシピ」や「接客で絶対言ってはいけないNGワード集」を『クラス専用ノート』にまとめます。

クラスのみんなはそのノートを読み、AIが自動で出す「タマネギは何分炒めるのが正解？」「お客様に『いらっしゃいませ』の次に言うべき言葉は？」といったクイズを解きます。

具体例：

2-A組の担任は「カレー作りの10の黄金ルール」をノートに追加。3日後のテストでは、28名中17名が「玉ねぎの炒め時間」を間違えたことが判明しました。

### 2 asagami AIがCursor Rulesを自動生成

集計データに基づいて、asagami AIが自動的にCursor Rules（.cursorフォルダ）を生成します。これが「調理や接客を手伝ってくれる超優秀な助っ人AI」の設定ファイルになります。

生徒が実際にカレーを作る時、Cursorという賢いアシスタントが「玉ねぎは透き通るまで弱火で15分以上炒めましょう！」と適切なタイミングでアドバイスしてくれます。



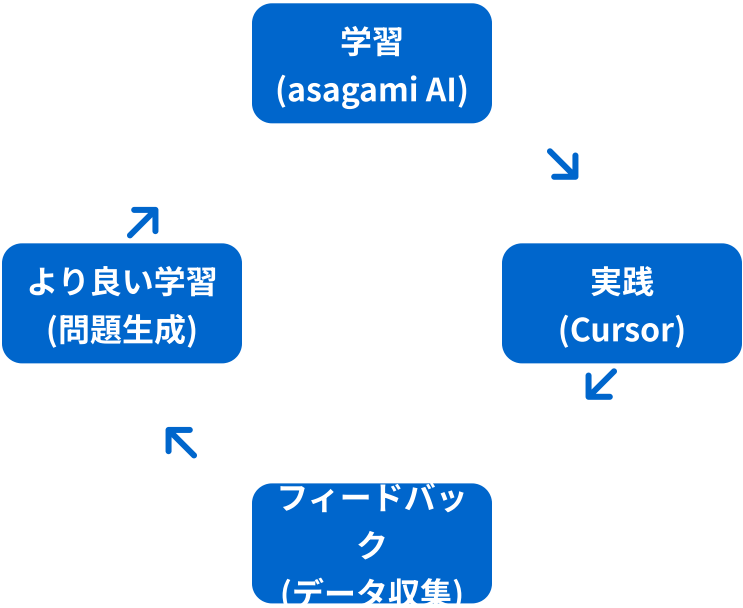
### 3 Cursorからのフィードバックで正のループ

カレー作りや接客の実践中のログ（誰がどこでつまづいたか）をCursorが記録し、それをasagami AIにインポートすると、AIが自動で「よくある間違いまとめノート」を作成して新しい問題を生成します。

これにより、「学習→実践→フィードバック→より良い学習」という正のループが生まれます。

実践例：

文化祭当日、新たに「お客さんが多いときの対応」の課題が発生。Cursorの記録から、asagami AIが翌週のテストで「混雑時の効率的な作業分担」という新しい学習項目を自動生成しました。



学習・実践の継続的な改善サイクル

# 具体的な連携シナリオ

## トップダウン式ナレッジ浸透モデル

### 金融システム開発チームの例

#### 1 研修ノート作成

リーダーが「最新の金融セキュリティ要件」に関する資料をasagami AIのノートにまとめる



#### 2 習熟度テスト

AIがノートから「4択問題」や「穴埋め問題」を自動生成。チームメンバーが解答し、理解度を可視化



#### 3 弱点分析とルール適用

👤 **Aさん**：「データ暗号化」の正答率が低い → 暗号化関連コードを書く際、自動で社内推奨ライブラリを提示

👤 **Bさん**：「個人情報保護法」の理解が浅い → 個人情報にアクセスする処理を書くと、警告と関連ドキュメントへのリンクを表示

👤 **チーム全体**：「セキュリティログ分析」が弱点 → チーム全員に適用されるログ処理コードレビューの強化ルールを自動設定

#### ★ 期待される成果・ベネフィット

- ✓ セキュリティ知識の均一化と実装品質の向上（脆弱性の80%削減）
- ✓ 個人に合わせたサポートによる学習効率の向上（習得時間30%短縮）
- ✓ コンプライアンス違反のリスク低減と事前予防の実現

## ボトムアップ式スキル向上モデル

### 新人エンジニアのオンボーディング例

#### 1 エラー検知と分析

新人のCさんが開発中に以下の課題が発生：

⚠️ ReactのHooksに関する同じエラーを3回繰り返し

⚠️ コードレビューで状態管理に関する指摘が複数回



#### 2 補強問題の自動生成

Cursorが検知したデータをAPI経由でasagami AIに送信。AIが弱点に特化した学習コンテンツを生成



#### 3 個別学習プランの実行

Cさんのasagami AIに以下が自動配信：

📖 「React Hooks集中特訓問題セット」（10問）

</> 「状態管理実践ガイド」（実装例付き）



#### 4 ルールの最適化と成長追跡

Cさんの理解度向上（テスト正答率85%）に伴い：

🔒 過剰なコード補完・警告ルールは緩和

📊 スキル成長データがマネージャーと共有され、次の育成計画に活用

#### ★ 期待される成果・ベネフィット

- ✓ 新人の自律学習促進と成長速度の加速（従来比2倍）
- ✓ エラー再発防止と品質向上（同一エラー発生率70%減）
- ✓ 先輩エンジニアの指導工数削減と効率的なナレッジ移転