

2026年05月20日（水）

## IBM Bob サロン 第4回

Ver. 1.0

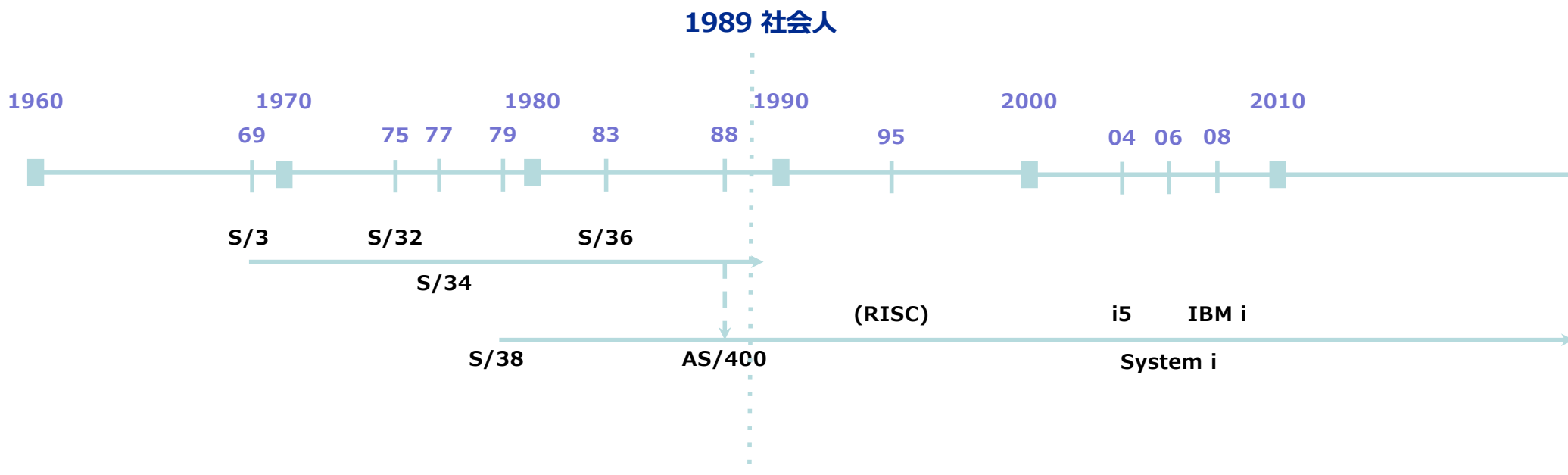
# AS/400 時代から IBM i を触ってきた私が IBM Bob に驚いた理由

ティアンドトラスト株式会社  
小川 誠



# 自己紹介

名前	小川 誠
会社	ティアンドトラスト株式会社
経歴	1989年から AS/400 / IBM i の開発および教育（37年）
社業以外の活動	技術情報発信 / セミナー登壇など
IBM 認定	IBM Champion (2024, 2025, 2026)



# IBM Champion としての Bob への正直な評価

効果的な使い方	丸投げするとがっかりする使い方
レガシーコードの読解・説明を頼む	業務ロジックの理解まで任せる
バグ修正の方針を一緒に考える	完成したコードを出させようとする
リファクタリングの設計を相談する	判断まで委ねる
仕様書・コメントの骨格を作らせる	自社固有の文脈を説明せずに頼む

がっかりの多くは Bob の問題ではなく使い方の問題

- ・ 業務を理解すること
- ・ 言語化すること
- ・ 判断すること



自分たちの仕事

**IBM Bob は便利な道具ではなく、一緒に考えるパートナー**

---

# IBM i 開発の現実

---

# IBM i 開発現場の3つの深刻な問題

## ドキュメントがない



仕様書や設計書が整備されておらず、システムの中身が確認できない。

## 開発者に聞けない

当時のプログラムを書いた本人が既におらず、修正の背景が不明。

せめて変更履歴だけでも残っていれば・・・

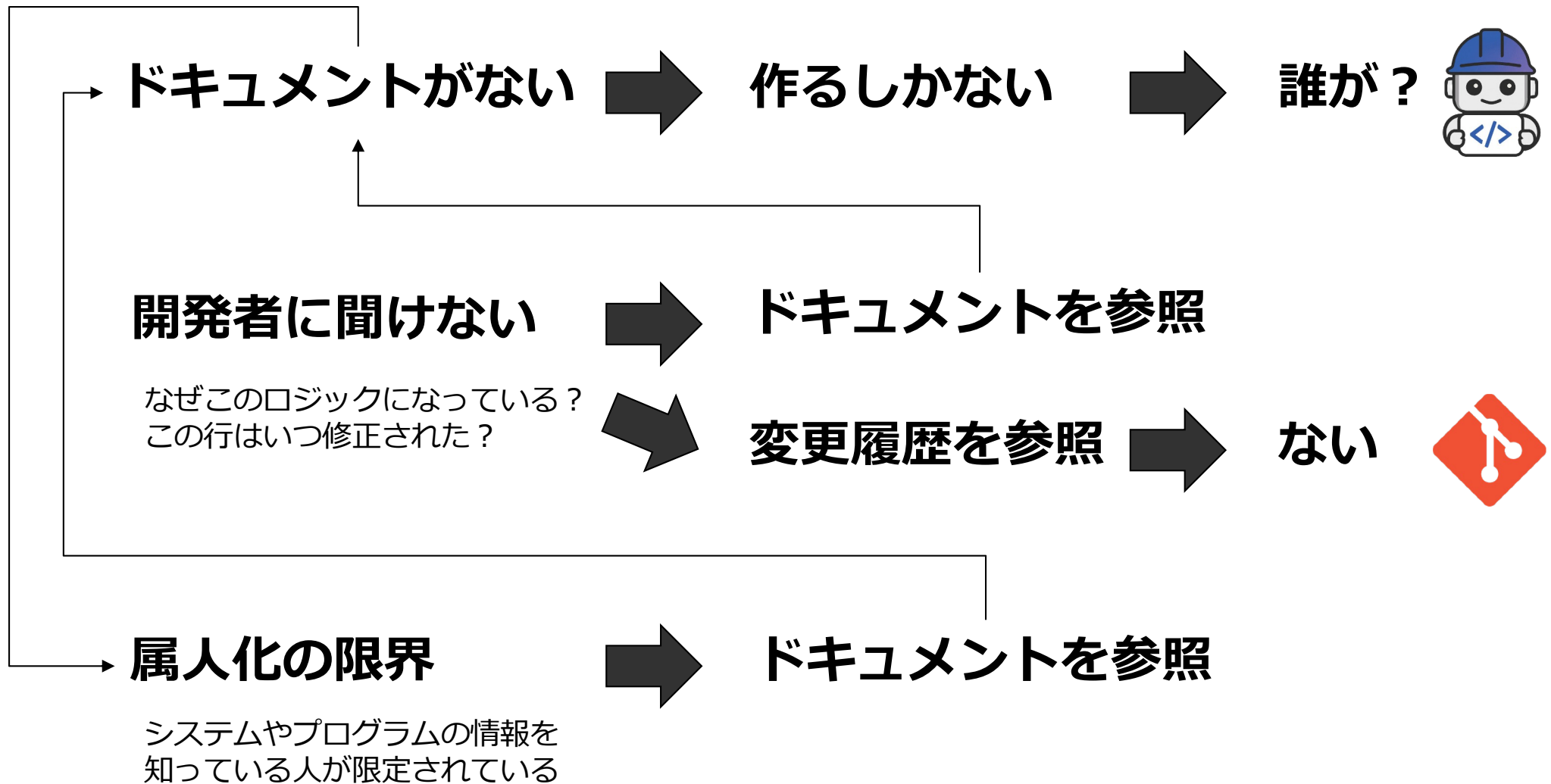


## 属人化の限界



特定の担当者一人に業務と知識が集中し、組織的な対応が困難。

# 3つの深刻な問題を解決するには？



---

# IBM Bob を使ってみた感想

---

# Bob との対話

□ Ask モードで部分確認 → Plan モードで仕様書に保存

	Ask モード	Plan モード
コードを読んで説明する	○	○
Markdown ファイルに書き出す	X	○
ソース・コードを変更する	X	X

→ Step 1 : Ask モードで不明箇所を Bob に聞く

→ Step 2 : Plan モードで仕様書を出力してもらう

# Bob と対話する - 1

## □ サブルーチン @RST の内容を聞いてみる

The image shows a screenshot of the IBM Bob interface. On the left, a code editor displays the following code:

```
182 C ENDSR
183 C @RST BEGSR
184
185 MOVEA '0000' *IN,60
186 MOVEL *BLANK X9MSG
187
188 ENDSR
189 BEGSR
190
```

A context menu is open over the code, with the following options:

- クイック修正
- Add to IBM Bob
- Explain with IBM Bob
- Improve with IBM Bob

An arrow points from the "Add to IBM Bob" option to a chat window on the right. The chat window displays the following code:

```
qrpgrsrc/pri04r.rpg:183-188
...
C @RST BEGSR
C**
C MOVEA'0000' *IN,60
C MOVEL*BLANK X9MSG
C**
C ENDSR
...
```

Below the code, the chat window contains the text: "このサブルーチンは何をしているのですか？" (What does this subroutine do?). At the bottom of the chat window, there is a "Ask" button and some navigation icons.

このサブルーチンは何をしているのですか？

# Bob と対話する - 2

@RST サブルーチンは画面表示のリセット処理を行っています。

## 処理内容

```
C      @RST      BEGSR
C**
C      MOVEA '0000' *IN,60
C      MOVEL *BLANK X9MSG
C**
C      ENDSR
```

## 具体的な動作

### 1. インジケータ60-63のクリア

- `MOVEA '0000' *IN,60`: インジケータ60から63を全てオフ('0')に設定
- これらのインジケータはエラー表示の制御に使用されています

### 2. エラーメッセージのクリア

- `MOVEL *BLANK X9MSG`: エラーメッセージフィールド(X9MSG)を空白にクリア

## 使用箇所

このサブルーチンは以下の場面で呼び出されています:

- @HED サブルーチン内: ヘッダー画面表示後
- @DTL サブルーチン内: 明細画面表示後

## 目的

画面遷移時や新しい入力を受け付ける前に、前回のエラー表示状態をクリアして、画面を初期状態に戻すための処理です。これにより、ユーザーは常にクリーンな状態で画面操作を開始できます。

「標識 60-63 はそれぞれどのような目的で使われていますか？」と追加で質問

# Bob と対話する - 3

標識(インジケータ)60-63の具体的な使用目的は以下の通りです:

## 標識60: エラーメッセージ表示制御

目的: エラーメッセージフィールド(X9MSG)の表示/非表示を制御

設定箇所:

- @CHKH: 受注番号が0の場合に '11' を設定 (IN60とIN61を)
- @CHKSF: 品目番号エラー時にON
- @CHKSF: 数量エラー時にON

表示されるメッセージ:

- ERR,1: 受注番号未入力エラー
- ERR,2: 品目番号エラー (品目マスタに存在しない)
- ERR,3: 数量エラー (0以下の値)

各標識ごとに目的や設置場所等を  
まとめて教えてくれる

## 標識61: 受注番号フィールドのエラー表示

目的: X1BANG (受注番号) フィールドを反転・カーソル位置指定で強調表示

設定箇所:

- @CHKH: 受注番号が0 (未入力) の場合

画面表示: 受注番号フィールドがDSPATR(RI PC)で反転表示され、カーソルが移動

## 標識62: 品目番号フィールドのエラー表示

目的: Y1HINB (品目番号) フィールドを反転・カーソル位置指定で強調表示

設定箇所:

- @CHKD: ループ開始時に '00' でクリア
- @CHKSF: 品目マスタ (TEITMP) に存在しない品目番号の場合

画面表示: サブファイル内の該当行の品目番号がDSPATR(RI PC)で反転表示

# 仕様書をゼロから作る

## PRI04R プログラム仕様書

### 1. プログラム概要

#### 1.1 プログラム名

- プログラムID: PRI04R
- プログラム名: 受注入力プログラム
- 画面ID: PRI04RD
- 画面名: 受注入力画面

#### 1.2 機能概要

受注データの登録・変更を行うプログラム。受注番号を入力し、品目の明細を最大15行まで入力・編集できる。

#### 1.3 処理モード

- 登録モード: 新規受注データの登録
- 変更モード: 既存受注データの変更

## 2. 使用ファイル

### 2.1 データベースファイル

ファイル名	用途	アクセス	キー	説明
TEITMP	定価マスタ	IF (入力)	K (キー付き)	品目の定価情報を管理
TEJUTP	受注明細ファイル	UF (更新)	K (キー付き)	受注明細データを管理

PRI04R は 203 行のコメントなしのRPGIII ソース

プロンプトに指定したのは pri04r.rpg だが、関連する表示装置ファイル pri04rd.dspf も読んで仕様書を作成

Bob が出力した Markdown ファイルをプレビュー機能で表示したもの

# バグが発生した場合

## □ 同じ依頼でも、結果は全く違う

「業務の背景まで  
伝えないと  
わからないはず」

	業務経験のない開発者	ベテラン IBM i 開発者
Bob への依頼	「SHPHOLD フラグを確認して引当がスキップされるバグを修正してください」  「フラグ名を伝えればわかるはず」 (AI だからやってくれるという期待)	「受注確定時に在庫引当を行う RPG プログラムです。出荷保留フラグ (SHPHOLD) が 'Y' の場合は引当をスキップすべきですが、現在はフラグに関係なく引当が実行されています。SHPHOLD='Y' のときは引当をスキップし、エラーログに記録する修正をお願いします。」
Bob の応答	「スキップすべきなのは 'Y' のときですか？エラーログは必要ですか？」 (確認が 3 ~ 4 往復続く)	条件分岐の修正案・エラーログの書き方・影響範囲の3点をまとめて提示
結果	的外れな提案が続く → がっかりする	計画書を受け取って採否を判断するだけ

IBM Bob だけでなく 他の AI も伝えた分だけ動く。それ以上でも、それ以下でもない。

# ベテランの開発者こそ、Bob を使いこなせる

長年の経験



業務ロジックの深い理解



「何をしたいか」を言語化できる



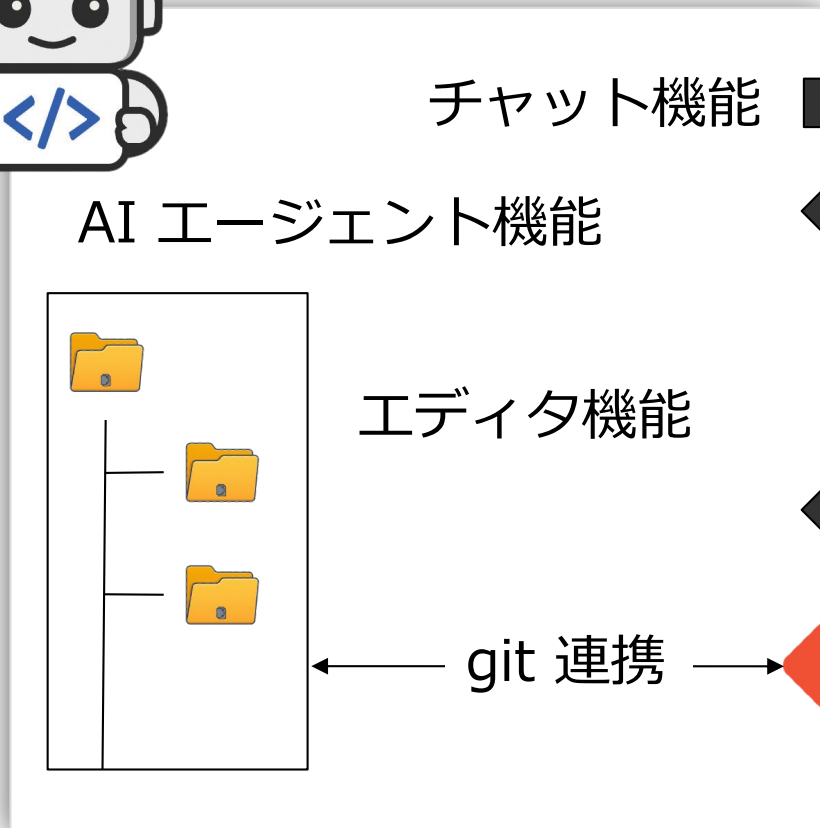
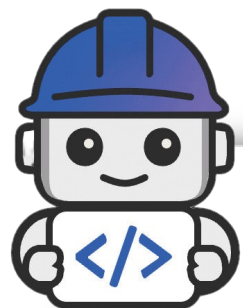
Bob を使いこなせる

---

# IBM Bob の構造とコンテキスト

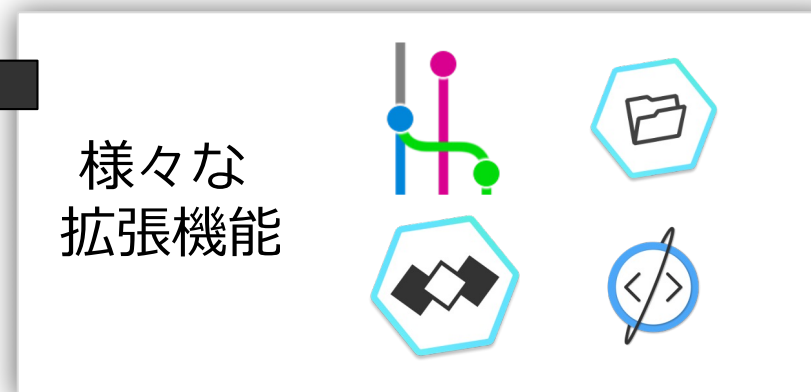
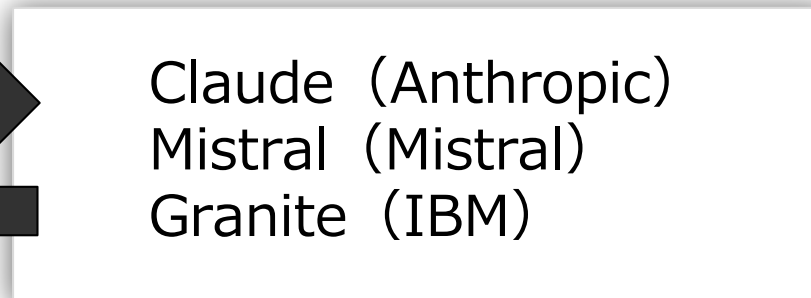
---

# Bob の構造



コンテキスト

AI モデル



# コンテキストについて

## → コンテキストとは

- ◆ AI に一度に渡される情報の総体
- ◆ プロンプトも履歴もファイルも全て含む
- ◆ トークンで測定 (20万トークン / 200K)

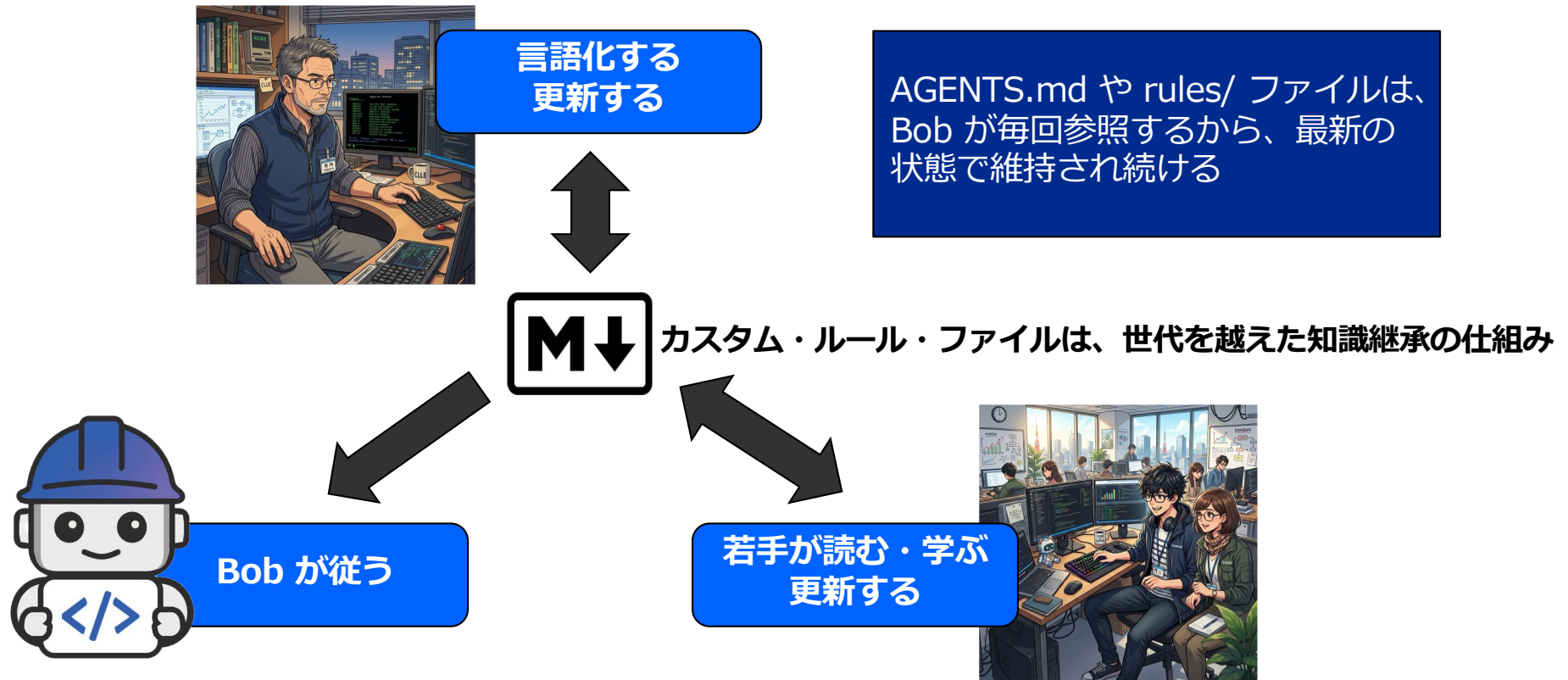
## → コンテキストに含まれるもの



種類	タイミング	内容	追加方法
システム指示	常時	モードの役割・ツール定義	自動
AGENTS.md・rules ファイル	常時	プロジェクト・モード別のルール	ファイルを置くだけで自動
MCPツール定義	常時	外部サーバーのツール一覧	MCP接続時に自動
Skills	有効化時	再利用可能な指示セット	有効化した時に自動
会話の履歴・ツール出力	蓄積	やり取りの蓄積・コマンド実行結果	タスク内で自動蓄積
@メンション・選択コード	都度	ファイル・フォルダー・選択範囲など	手動で指定
ユーザーのプロンプト	都度	今回の質問・指示	手動で入力

# システムの暗黙知を言語化してチームの財産に

	AGENTS.md	rules/
仕組み	プロジェクトの「文脈」	開発の「知恵」
内容	このシステムは何か・方針など	命名規則・エラー処理パターン・設計判断など



カスタム・ルール・ファイルの詳細は <https://bob.ibm.com/docs/ide/configuration/rules> を参照のこと

---

# まとめ

---

# まとめ

「AI に乗り遅れたと思っていたが、  
実は、一番重要な立場にいた。」



これまで	Bob が「居る」世界
仕様書がない・書いた人もいない	10分で仕様書の骨格ができる
一人でバグの原因を探し続ける	計画書を受け取って判断しながら進める
言語化できるか、使いこなせるか不安	長年の経験がそのまま武器になる



まず、「何ができるか」を知るために、とことん Bob を使ってみてください。もちろん Bob コインは消費します。でも、それだけの価値はあります！

# 【大切なこと】 誰が・いつ・何のために変えたかを残す

## → レガシーコードへの後悔

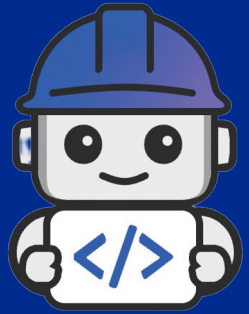
- ◆ この行は誰がいつ変えたのか
- ◆ なぜこのロジックになっているのか
- ◆ 変更履歴があれば防げた苦勞が、どれだけあったか

## → AI 開発における変更管理の重要性

- ◆ AI が生成したコードだからこそ「なぜそう変えたか」の記録が必要
- ◆ チケット駆動開発：変更の「なぜ」をチケットに残す
  - コードと理由が紐づく
- ◆ カスタム・ルール・ファイル 自体も git で管理する
  - いつ・なぜこのルールを追加したかが残る
  - チームで共有できる



過去の歴史を再構築することはできない  
でも、これからの歴史の記録をはじめることができる



まず「何ができるか」を知るために若い方もベテラン開発者も、とことん使ってみてください

---

AS/400 時代を知る人間として、  
IBM Bob を自信を持って  
お勧めします！！

# ティアンドトラスト株式会社の紹介

「優しさ」と「信頼」で  
お客様のビジネスに “Good Cycle” を。

IBM i +  $\alpha$  = T&T Trust  
Tenderness and Trust  
<https://tat.co.jp>

## 受託開発

信頼性においては定評のある IBM i をメインとしたアプリケーション開発を行います。お客様とのコミュニケーションを第一に考え、最適なソリューションをご提案しています。

## 顧客研修

外部研修にて IBM i コースを担当する専任のインストラクターがいます。また、お客様のご要望に応じたオーダーメイドの研修コースの提供も行います。

## 技術開発

IBM i の最新技術だけでなく、お客様にとって必要になるであろうコンピュータ技術を日々蓄積しています。

## ティアンドトラスト株式会社

〒111-0053

東京都台東区浅草橋4-16-4 ウイングイトビル6F

フリーダイヤル：0120-913-474

代表：03-5821-3666