



# 次期LMSシステム 「生成AI駆動開発準備作業」 キックオフミーティング

2025.3.3  
CTOグループ技術戦略本部

1. 体制
2. 弊社の生成AIによるシステム開発の生産性向上の取り組み概要
3. 目的
4. 作業内容、成果物
5. 弊社で実施した前回PoC(LMSプロトタイプ開発)の結果共有
6. その他(コミュニケーション/ツール)
7. Appendix

本調査に関するCTC側の技術的な問い合わせは以下3名を中心に対応いたします。

- **有馬 正行(Masayuki Arima)**
  - ー AIアドバイザー
  - ー [Masayuki.arima@ctc-g.co.jp](mailto:Masayuki.arima@ctc-g.co.jp)
- **玉井 龍(Ryo Tamai)**
  - ー AIアドバイザー(補佐)
  - ー [Ryo.tamai@ctc-g.co.jp](mailto:Ryo.tamai@ctc-g.co.jp)
- **菅原 和彦(Kazuhiko Sugawara)**
  - ー プロジェクト管理
  - ー [kazuhiko.sugawara@ctc-g.co.jp](mailto:kazuhiko.sugawara@ctc-g.co.jp)

## 2. 弊社の生成AIによるシステム開発の生産性向上の取り組み概要

### プロトタイプでのトライアル

2024年9月～11月

LMSプロトタイプ開発  
(AI駆動開発PoC)

16%の生産性向上を  
確認(製造工程では  
30%向上)

### 事前準備

2025年3月～

生成AI駆動開発  
準備作業

「AI駆動開発準備作業」の目的:

- ① 生成AIを活用できるツールを幅広く調査・評価し、最適ツールの選定を行う
- ② 選定ツールの活用方法や導入手順を整理し、本格的な開発着手前に不確定要素を排除する
- ③ プロトタイピングを通じ、生成AI導入による工数削減や品質向上の実績・データを取得し、次工程の工数見積に反映する

### 実プロジェクトでの試行

2025年6月～2026年3月

生成AI駆動開発

次期LMS開発を検討中  
順次、全社のSIへ適用開始

「生成AI駆動開発準備作業」の目的として以下を考えています。

1. 生成AIを活用できるツールを幅広く調査・評価し、最適ツールの選定を行うこと。
2. 選定ツールの活用方法や導入手順を整理し、本格的な開発着手前に不確定要素を排除すること。
3. プロトタイピングを通じ、生成AI導入による工数削減や品質向上の実績・データを取得し、次工程(※)の工数見積に反映すること。

※次期LMS開発をAI駆動開発にて行うことを検討中

### 1. 作業内容

※詳細は別紙参照



20250205-AI駆動開発準備作業.pdf

#### 1. 生成AIツールの詳細調査

- ・ 設計・開発・テスト工程で使用する生成AIツールの調査・選定
- ・ 評価結果を踏まえ、導入候補を絞り込む

#### 2. プロトタイピング(PoC)実施

- ・ 生成AIツールを用いて設計・製造・テストを試行し、工数削減度・品質向上度を測定

#### 3. 定量効果・定性評価のレポート作成

- ・ プロトタイプで得た効果をまとめ、次工程の工数見積に反映可能な指標を作成

#### 4. ツール導入・利用手順のドキュメント整備

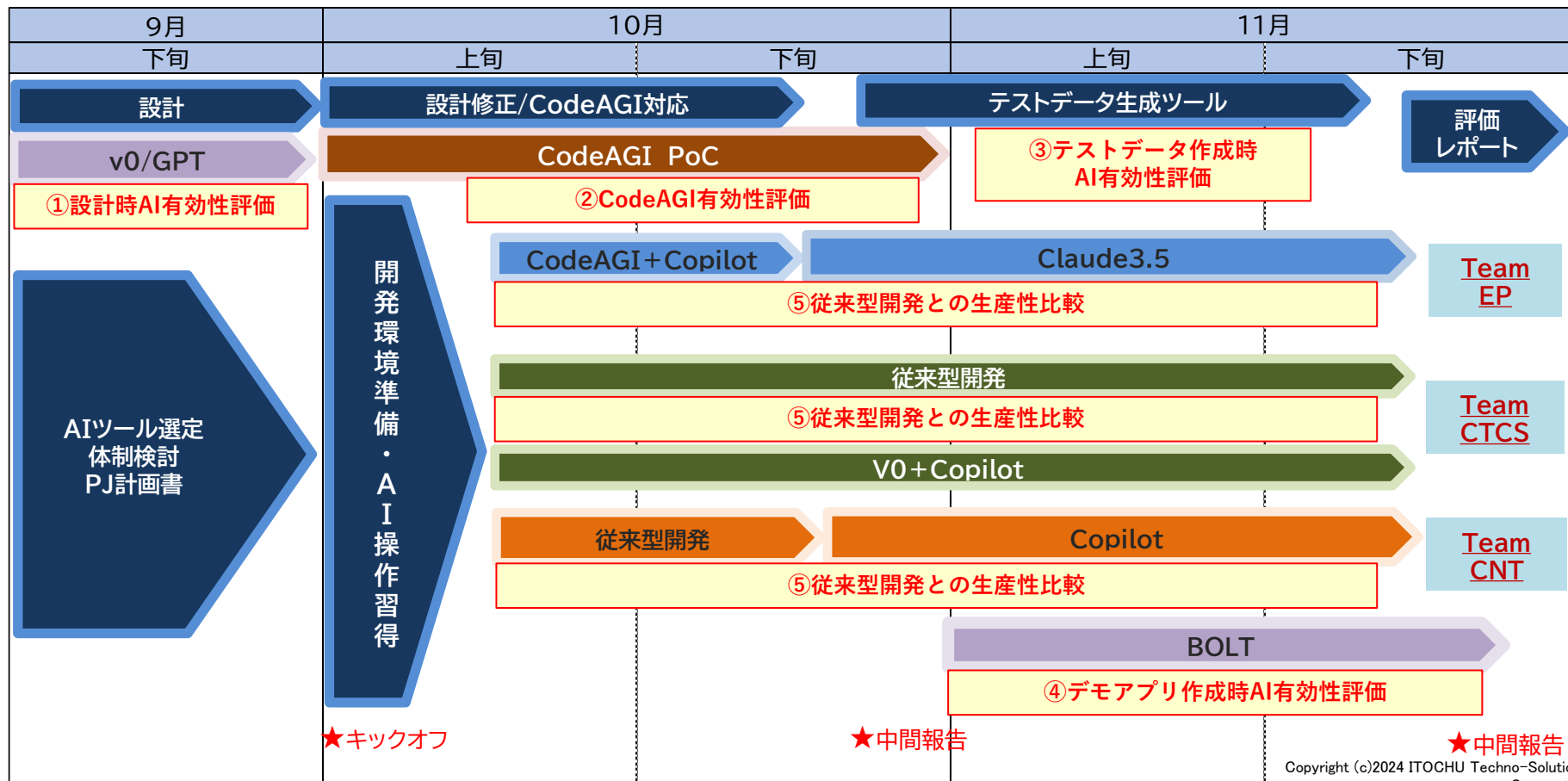
- ・ 導入するツールのセットアップ方法、運用手順、セキュリティ設定をまとめる

### 2. 成果物

1. 作業報告書: 毎月末
2. AIツール評価報告書: 適宜提出(ツール比較結果、PoC結果含む)
3. その他: プロトタイプ成果物、導入ガイド・運用ドキュメント

## 5. 弊社で実施した前回PoC (LMSプロトタイプ開発)の結果共有

## スケジュールと評価観点







実装フェーズやテストフェーズでの作業支援では効果が高かったが、設計や総合試験工程では、生成AIツールの効果は限定的であった。

(プロトタイプ開発で効果が確認できたのは下記赤枠)

企画・要件定義	設計・実装		テスト	保守・運用
<b>アイデア創出</b> AIとの対話によってアイデアを膨らませる。	<b>ソースコード・SQL自動生成</b> AIに作成してほしいコードの概要を入力し、プログラムやSQL文を生成させる。対話を通じてブラッシュアップする。		<b>テストケース生成</b> プロンプトにより、システムの機能テストや総合テストのテストケースを作成する。	<b>インフラ・リソース管理</b> AIにインフラやリソースの管理、デプロイを支援してもらう。
<b>要求・ニーズ分析</b> AIにユーザのペルソナを演じてもらい、要求やニーズを洗い出す。	<b>リファクタリング</b> 既存のソースコードから機能をそのままにデザインパターンに即したコードに修正したり、可読性の高いコードに修正する。	<b>コード変換</b> COBOLからJavaなど、あるプログラミング言語で記述されたコードを別の言語のコードへ変換する。	<b>テストコード生成</b> プロンプトにソースコードを入力し、それに応じた単体テストコードを作成する。	<b>脆弱性検知・修正</b> 使用ライブラリ管理支援など、システムに脆弱性がある場合に検知し、修正案を出してもらう。
<b>システム要件作成</b> AIに機能・非機能要件の案を生成させる。対話を通じてブラッシュアップしたり、作成した要件の漏れがないかレビューしてもらう。	<b>コード説明</b> AIに既存の複雑なコードの内容を要約・説明してもらい、ドキュメント作成やリファクタリングに繋げる。	<b>コードレビュー</b> AIにソースコードについて可読性や機能面、パフォーマンス、セキュリティ等の観点からアドバイスをもらう。	<b>バグ検出・修正</b> AIに作成したコードのデバッグ時のエラーメッセージを基に修正案を出してもらう。	<b>運用効率化</b> システムで発生したエラーメッセージなどを解析し、パッチ適用による修正案を出してもらう。
<b>UIデザイン作成</b> AIにUIデザイン案を出してもらい、対話を通じてブラッシュアップし、コードに変換してもらう。				
<b>情報調査:</b> 各工程で必要な情報を、対話やWeb検索により効率的に取得する。				
<b>ドキュメント作成支援:</b> プロンプトにより、システム開発に用いる各種ドキュメントの雛形や下書きを作成し、対話を通じてブラッシュアップする。				
<b>PJ管理支援:</b> PJ計画案・タスク案の作成、タスク消化実績の要約によるPJ進捗管理、チケットの要約による課題管理等をAIに支援してもらう。				

## 5.3.LMSプロトタイプ開発で検証した工程・AI開発ツール

実装フェーズやテストフェーズでの作業支援では効果が高かったが、設計や総合試験工程では、生成AIツールの効果は限定的であった。  
(プロトタイプ開発で使用した生成AIツールは下記赤字)



使用AIツール	実業務での利用	使用する場合の制限事項/使用できない理由
CodeAGI	×	実業務で利用できない理由: ・メーカーと毎回PoC契約による改修が必要 ・設計書のフォーマットによる制約が多い ・チャットによる微修正がなく <b>毎回一からコードを生成するため、コード修正後の仕様変更が困難</b> ※オプトアウト機能あり
ChatGPT (gpt-4o)	○	実業務で利用する場合の制限: ・アウトプットは人間のチェックが必要 ・効果は使用者の技量次第 ・顧客情報や顧客ソースをLLMに入力できること ・ <b>毎回結果が異なることを許容できること</b> ※オプトアウト機能あり
v0	△	<b>プロトタイピング</b> 、UIデザインなどで利用できる可能性あり 実業務で利用する場合の制限: ・ <b>オプトアウトが未実装</b> なため、顧客データ入力不可 ・ <b>毎回結果が異なることを許容できること</b> ・言語、フレームワークに制限あり
Github Copilot	○	実業務で利用する場合の制限: ・アウトプットは人間のチェックが必要 ・効果は使用者の技量次第 ・顧客情報や顧客ソースをLLMに入力できること ・ <b>毎回結果が異なることを許容できること</b> ※オプトアウト機能あり パブリックコード利用制限機能あり
Claude (3.5 sonnet)	○	実業務で利用する場合の制限: ・アウトプットは人間のチェックが必要 ・効果は使用者の技量次第 ・顧客情報や顧客ソースをLLMに入力できること ・ <b>毎回結果が異なることを許容できること</b> ※オプトアウト機能あり ・有料版でも利用制限あり(5時間に45回) ・現在では性能もOpenAIよりも低い
Bolt.new	△	<b>プロトタイピング</b> などで利用できる可能性あり 実業務で利用する場合の制限: ・ <b>オプトアウトが未実装</b> なため、顧客データ入力不可 ・ <b>毎回結果が異なることを許容できること</b> ・商用利用には、TeamsプランかEnterprise & Self-hostedプランが必要 ・言語、フレームワークに制限あり

### 各観点における評価結果

#### ①設計時AI有効性評価

- ChatGPT、Claudeを利用して、PlantUMLやMermaidによりシーケンス図やER図を作成するなど設計書作成の補助で活用。画面項目やDBカラムの洗い出しにも効果的。
- V0を使って画面デザインの補助に使用できた。

#### ②製造工程でのAI有効性評価

- CodeAGI(Excel設計書からコードとテストケースを自動生成するツール)は、生成コード/テストケース共に精度が低く、**毎回一から生成するため仕様変更が困難**であったため、実用性なしと判断。
- Github Copilotなどエンジニア補完型のツールは、単純ロジック実装には非常に効果的、単体テストコードも出力することが可能。(30%程度の工数削減効果)

#### ③テストデータ作成時AI有効性評価

- 単体テストコードは、AIツールにより自動作成できるが、コード生成をAIで行っているため、ケアレスミスが少なく効果は限定的であった。
- 結合試験/総合試験のAIによる自動化は、今回実施できていない。

#### ④デモアプリ作成時AI有効性評価

- Bolt.new(AIエージェント型ツール)を用い、本PJ開発対象5機能のデモ用アプリを作成。
- DBやビジネスロジックは実装されていないが、作成に要した時間は1画面あたり10分程度と非常に有効であった。

- コミュニケーション

- Zoom

- 定例打合せ(毎週木曜日11時は如何でしょうか)
    - Zoom翻訳字幕(検討中、サードパーティで何か良いものがあれば教えてください)

- 議事録(録画、トランスクリプトを共有します)

- ZoomIDを作成します
    - 定例議事録の作成をお願いします

- Slack(チャット)

- 使用可能か確認をお願いします。問題なければチャンネルを作成します

- Box(ファイル共有)

- 使用可能か確認をお願いします。問題なければフォルダを作成します

- プロジェクト管理・タスク管理

- Redmaineの使用を考えています

## 2. 対応体制案

オンサイト（日本）



案件責任者

御社担当者

Neusoft

日本現地統括責任者

梁 艶

担当営業

単 海格

SE兼技術リーダー

龍 拳兵

SE

劉 馨遥

オフショア（瀋陽）

Neusoft

SE兼リーダー

蔡 玉潔

SE

宋 奕博

SE

王 徳晨

### 役割説明

#### オンサイト側

- 推進方針の検討・策定
- 要件検討及び推進方針の検討・策定
- 作業進行中の問題点に関する情報連携、コミュニケーション効率確保

#### オフショア側

- 推進方針の検討・策定へ参画
- 要件検討及び推進方針の検討・策定へ参画
- AI検証作業の実施
- PoC成果物、検証レポートの作成

### コミュニケーション計画

- 朝会／夕会 頻度：毎日 ※合意要
- 週例会 頻度：毎週の金曜日 ※合意要
- 業務検討会 頻度：随時
- チームズチャット連絡 頻度：随時

- 開発環境、使用ツール候補
  - 生産性が上がるツールを御社の知見からご提案ください。
  - 例)
    - Markdownエディタ(AIが直接読み書きできるため、markitdown,typora)
    - Github Enterprise(コード管理、弊社の社内リポジトリ)
    - ChatGPT o1/o1 pro/o3-mini/Deep Research
    - Claude 3.7 sonnet
    - Vscode
  - クラウド環境
    - 想定するクラウドをお教えてください
  - 言語、フレームワーク
    - 想定しているものをお教えてください
  - 開発環境、ツールは御社でご用意ください。(費用はご請求ください)



- 中国国内で規制されているか確認が必要なサービス
  - Google系(Geminiなど)
  - OpenAI系(o1,o3-miniなど)
  - Bolt.new、v0など開発系開発ツール
  - DeepSeek(米国規制の対象となる恐れがあるため今回は除外)
  - その他

- 設計工程
  - Mermaid,PlantUMLなどによる設計図表の作成
  - OpenAI o1/o1 pro/o3-miniの推論機能による設計
  - OpenAI Deep Researchの調査能力
- テスト工程へのAIツール適用(GCで単体テストコードは出力)
  - 結合、総合など外部テスト(ツールも未選定)

Visual Studio Code interface showing the PlantUML editor and preview.

**Left Panel (Explorer):**

- Files: test.pu, uml
- WORK: dgx1, doc, gpt4all, uml
- test.pu
- ws12
- zoltraak
- zoombot-pj
- zoombot-source.zip

**Editor (test.pu):**

```
1 @startuml
2 actor 顧客 as customer
3 participant "システム\n(受付審査システム)" as system
4 participant "受付センター\n\n担当者" as staff
5 participant "電話番号照合\n\nシステム" as phone
6 participant "信用情報機関\n\n(CIC)" as cic
7 participant "支店\n\n担当者" as branch_staff
8 participant "与信\n\nモデル" as credit_model
9 participant "クレジット発行手続き\n\nシステム" as issue_system
10 participant "銀行の運用\n\n担当者" as operator
11
12 customer -> system : クレジットを申し込む
13 system -> system : 申し込みデータの内容を確認
14 system -> staff : 申し込みデータを送信
15 staff -> cic : 信用情報の問い合わせ
16 cic -> staff : 信用情報を返す
17 staff -> system : 自動審査の開始
18 system -> system : デフォルト率の算出
19 system -> system : クレジットカードの合否やキャ
20 system -> customer : メール通知
21 system -> operator : 与信モデルの管理
22 system -> phone_system : 登録電話番号を確認
23 alt 電話番号が存在しない
24 phone_system -> customer : 存在しない旨をメール
25 else 電話番号が存在する
26 phone_system -> cic : 信用情報を照合する
27 cic -> system : 信用情報を返す
28 alt 照合がない
29 system -> branch_staff : システム連絡をする
30 branch_staff -> credit_model : 与信モデルにか
31 else 照合がある
32 system -> credit_model : 与信モデルの判断に必要
33 credit_model -> system : デフォルト率と顧客格付
34 alt 当確
35 system -> issue_system : クレジット発行手続きを
36 issue_system -> customer : 発行の件をメールする
37 else 謝絶
```

**Preview (PlantUML プレビュー):**

The diagram illustrates the credit application process flow, including steps like 申し込みデータの内容を確認 (check application data), 申し込みデータを送信 (transmit application data), 信用情報の問い合わせ (credit information inquiry), 信用情報を返す (return credit information), 自動審査の開始 (start automatic review), デフォルト率の算出 (calculate default rate), クレジットカードの合否やキャッシング枠を算出 (calculate credit card eligibility and cashing limit), メール通知 (email notification), 与信モデルの管理 (credit model management), 登録電話番号を確認 (check registered phone number), and クレジット発行手続きをする (perform credit issuance process).