

生成AI+LowCodeによる次世代開発 上流工程からの生成AI活用 モックアップ生成デモンストレーション

LLMによるローコード支援

ローコードツールで何かを作るとき、「作り始めを高速化したい」と思ったことはありませんか？

こんなアプリがあったらいいな



日立製作所はJasminesoftと共同で、LLM（大規模言語モデル）を活用してアプリケーションを迅速に「動作可能」にする技術の開発に取り組んでいます

出力

3

デモ概要説明

入力

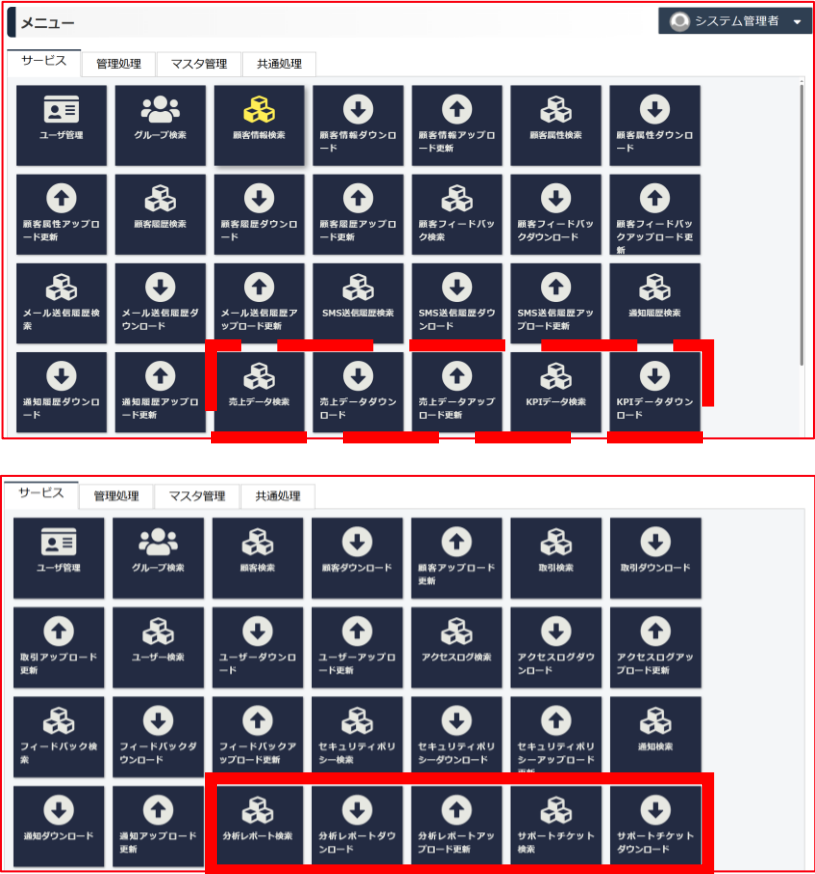
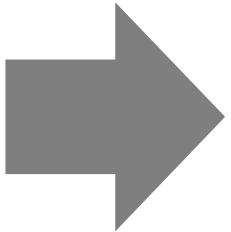
出力

前

```
"system_category": "顧客管理システム",
"client": "日立銀行",
"client_division": "法人営業部",
"theme": "非構造データの利活用"
```

後

```
"system_category": "顧客管理システム",
"client": "日立銀行",
"client_division": "法人営業部",
"theme": "顧客満足度の向上"
```



デモンストレーション

デモの流れ

Wagbyアプリ構築&マスタデータサンプル登録、ドキュメント生成を自動で実施

4行の要求からアプリと
マスタデータを生成

ドキュメント生成

要求を変更して
アプリ生成

①

- 日本語で4項目(4行程度)の
情報を入力
- アプリ要件と設計を生成
- Wagby Designerを使用し設
計に基づいたアプリを自動で
構築
- マスタデータサンプルを生成し
アプリ起動前に自動で登録

②

- アプリ生成と同時に、設計
書・要件定義書を生成
- > 画面一覧、エンティティ、デー
タのリレーションなど詳細な設計
情報が含まれる

③

- 4項目の入力内容を変更する
と、アプリの機能が変わる
- > “theme”項目を変更した場合
は、システムの大きな目的と基
本機能はそのままで、細かい
機能のみが変更される

デモ①

Wagbyアプリ構築&マスタデータサンプル登録、ドキュメント生成を自動で実施

4行の要求からアプリと
マスタデータを生成

ドキュメント生成

要求を変更して
アプリ生成

①

- 日本語で4項目(4行程度)の
情報を入力
- アプリ要件と設計を生成
- Wagby Designerを使用し設
計に基づいたアプリを自動で
構築
- マスタデータサンプルを生成し
アプリ起動前に自動で登録

②

- アプリ生成と同時に、設計
書・要件定義書を生成
- > 画面一覧、エンティティ、デー
タのリレーションなど詳細な設計
情報が含まれる

③

- 4項目の入力内容を変更する
と、アプリの機能が変わる
- > “theme”項目を変更した場合
は、システムの大きな目的と基
本機能はそのままで、細かい
機能のみが変更される

デモ②

Wagbyアプリ構築&マスタデータサンプル登録、ドキュメント生成を自動で実施

4行の要求からアプリと
マスタデータを生成

ドキュメント生成

要求を変更して
アプリ生成

①

- 日本語で4項目(4行程度)の
情報を入力
- アプリ要件と設計を生成
- Wagby Designerを使用し設
計に基づいたアプリを自動で
構築
- マスタデータサンプルを生成し
アプリ起動前に自動で登録

②

- アプリ生成と同時に、設計
書・要件定義書を生成
- > 画面一覧、エンティティ、デー
タのリレーションなど詳細な設計
情報が含まれる

③

- 4項目の入力内容を変更する
と、アプリの機能が変わる
- > “theme”項目を変更した場合
は、システムの大きな目的と基
本機能はそのままで、細かい
機能のみが変更される

デモ③

Wagbyアプリ構築&マスタデータサンプル登録、ドキュメント生成を自動で実施

4行の要求からアプリと
マスタデータを生成

ドキュメント生成

要求を変更して
アプリ生成

①

- 日本語で4項目(4行程度)の
情報を入力
- アプリ要件と設計を生成
- Wagby Designerを使用し設
計に基づいたアプリを自動で
構築
- マスタデータサンプルを生成し
アプリ起動前に自動で登録

②

- アプリ生成と同時に、設計
書・要件定義書を生成
- > 画面一覧、エンティティ、デー
タのリレーションなど詳細な設計
情報が含まれる

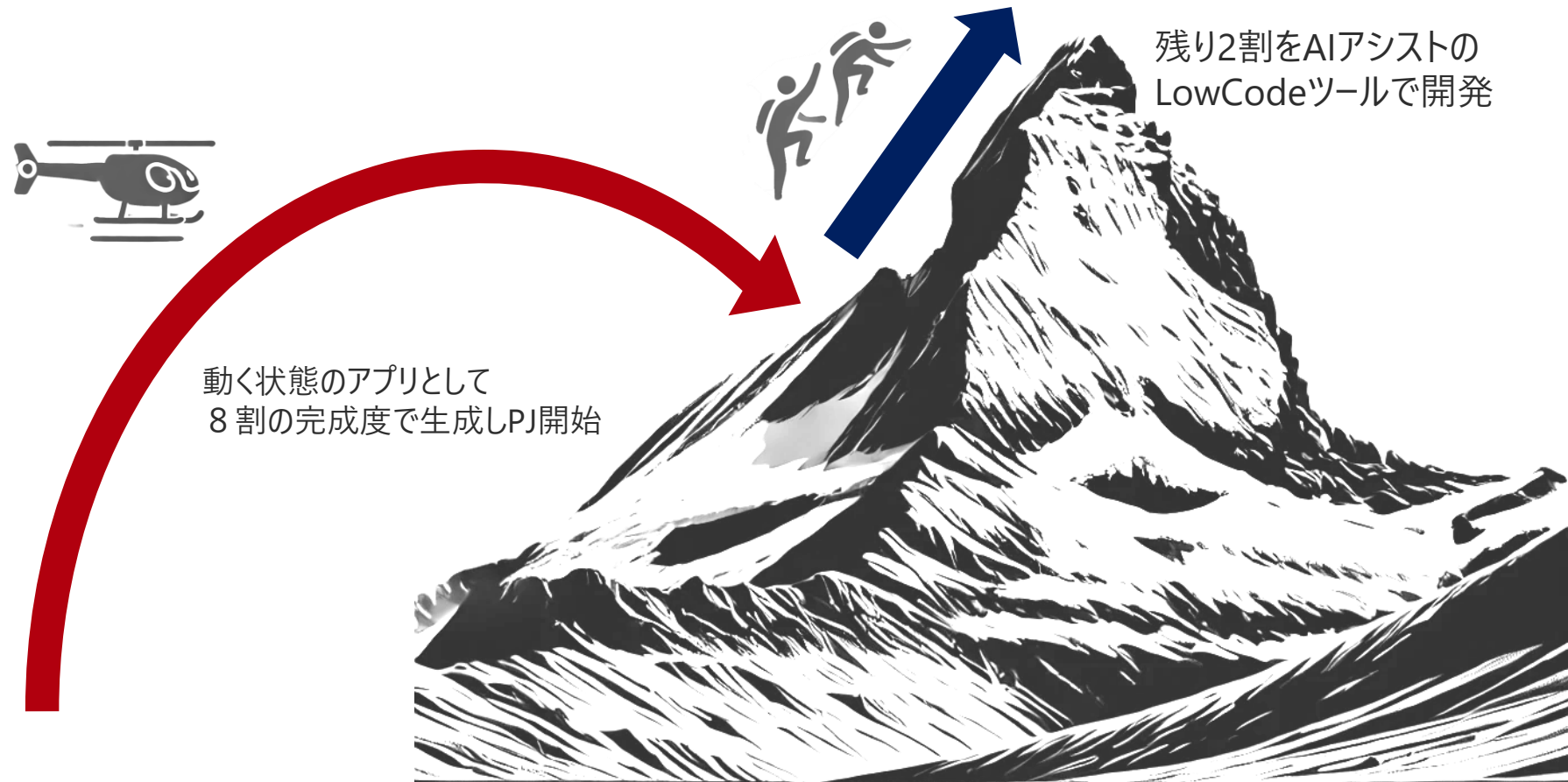
③

- 4項目の入力内容を変更する
と、アプリの機能が変わる
- > “theme”項目を変更した場合
は、システムの大きな目的と基
本機能はそのままで、細かい
機能のみが変更される

コンセプト

現状の技術レベルと従来開発手法の適切なバランスの実現

プロジェクト開始時に 8 割方完成している状態でスタート。残り 2 割は人力 + AIで完成に向かう



適用イメージ

LLM+LowCodeにより、アプリケーションの「8割」を短期間で開発

1 アプリ開発スタート

この分野の
開発は初めて
だから不安...



アプリ開発者

2 ユーザー要望のヒアリング

こんな感じですか？

ちょっと違う

あ、ぼい

お客様

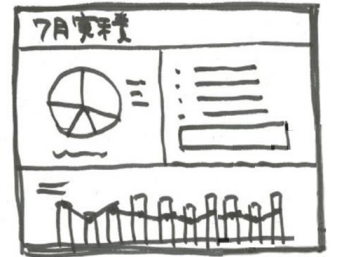
アプリ開発者



3 アプリ生成



アプリ開発者



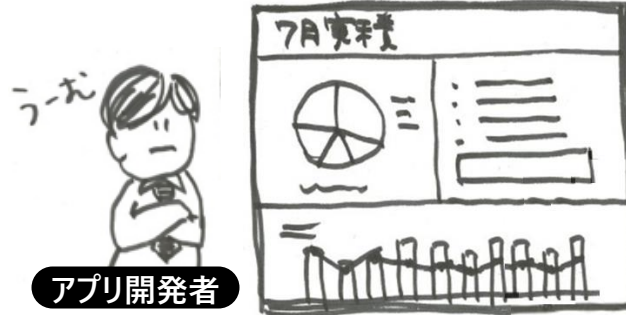
4 ユーザーに提示して確認



お客様

アプリ開発者

5 ユーザー意見を反映



アプリ開発者

6 ベースラインアプリをビルド

もうできたぞ！



アプリ開発者



注意事項

〔他社商品名称に関わる表示〕

- Wagbyは株式会社ジャスミンソフトの登録商標です。

生成AI+LowCodeによる次世代開発

上流工程からの生成AI活用 モックアップ生成デモンストレーション

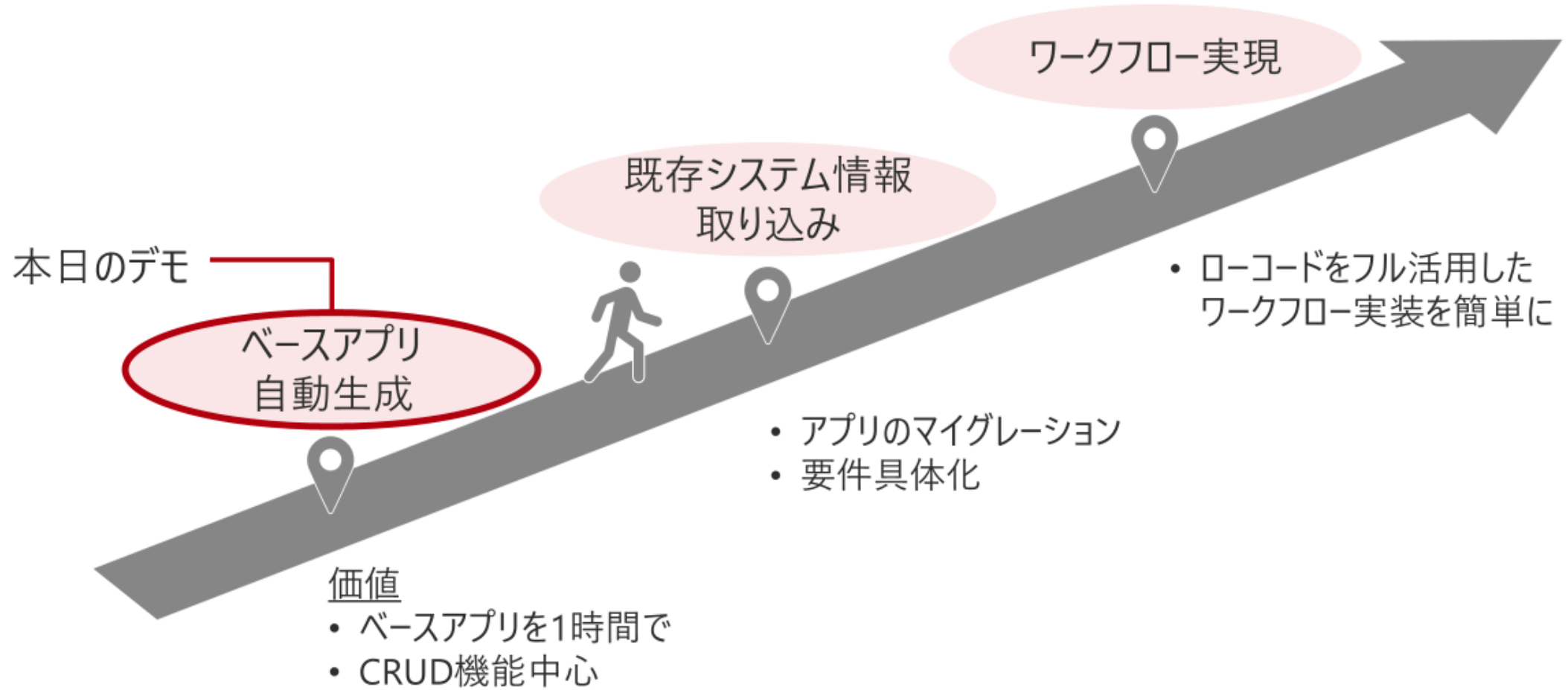
Follow us



HITACHI

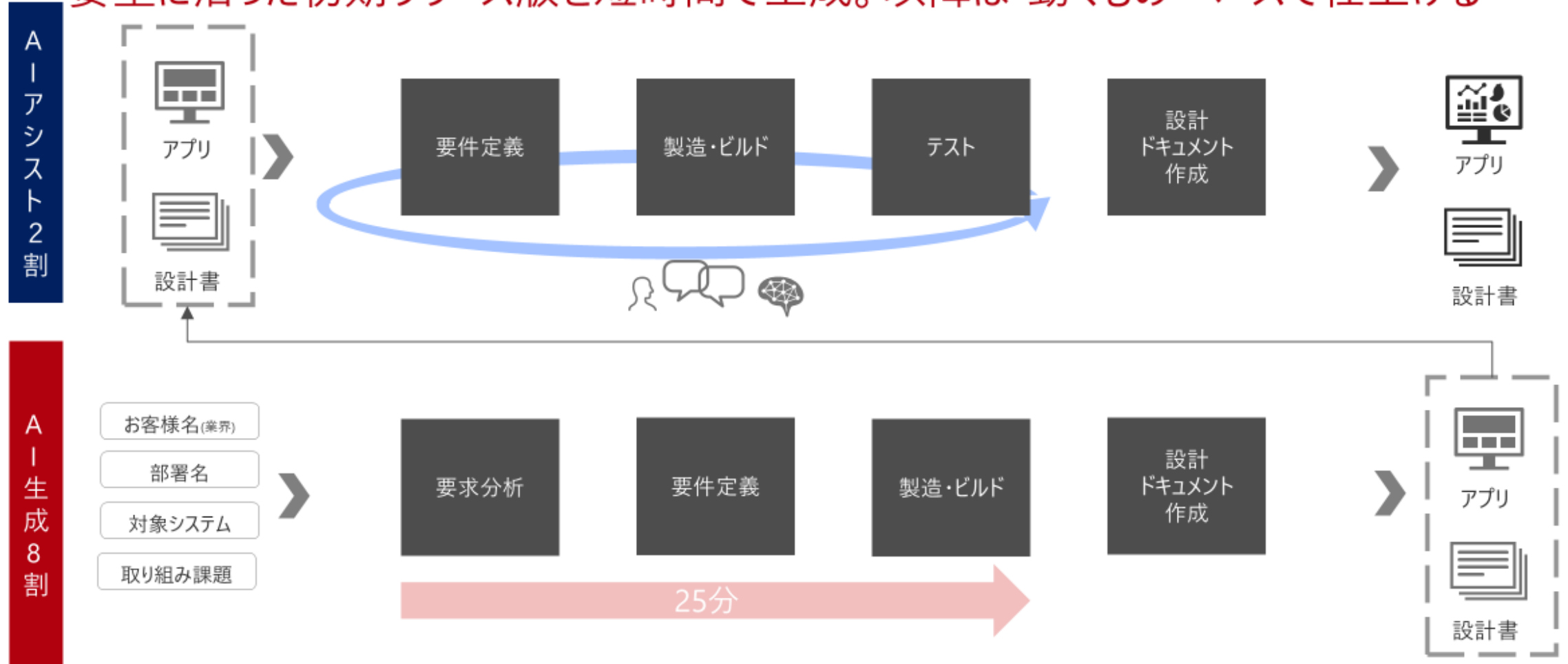
Appendix

ロードマップ



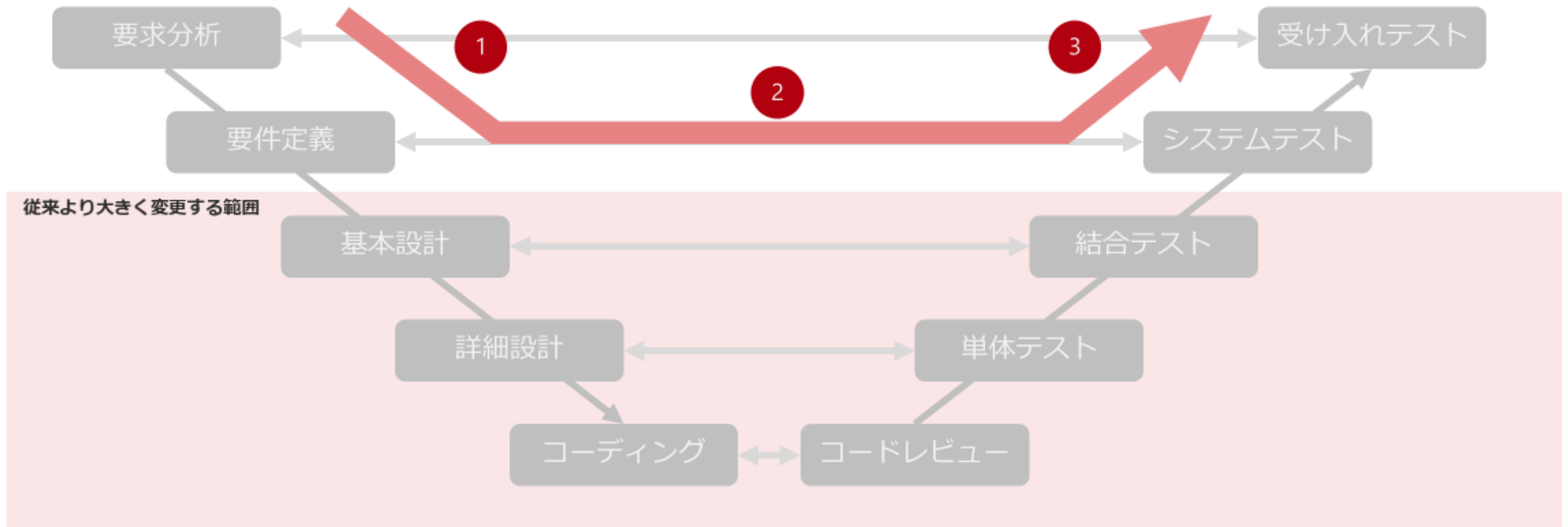
プロセスのイメージ

要望に沿った初期リリース版を短時間で生成。以降は“動くもの”ベースで仕上げる



実現したい姿

生成AIを全面適用し開発工程全般の効率化を図る



アプリケーション生成と同時に関連ドキュメントも生成

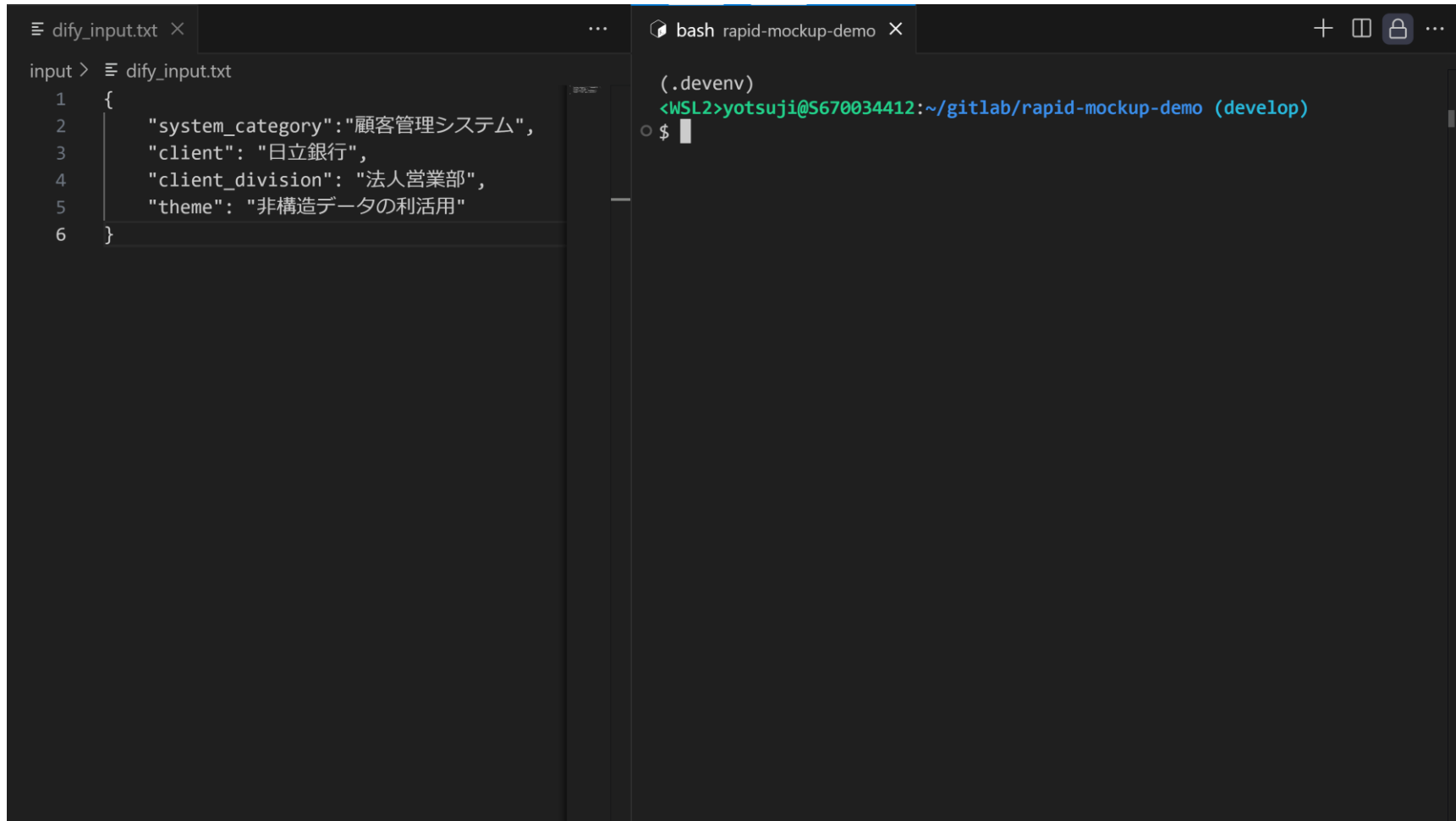
30年間の日立銀行の顧客管理システムにおける非構造化データの活用を基盤とした新たなシステムのあり方について、以下のポイントを考慮することが重要です。 1. データ統合プラットフォームの構築 多様なデータソースの統合: SMS、顧客フィードバック、取引履歴など、さまざまな非構造化データを一元的に管理できるプラットフォームを構築します。これにより、データのサイロ化を防ぎ、全体的な顧客理解を深めることができます。 APIファーストアプローチ: 各部署がデータソースやサービスとの連携を容易にするため、APIを標準とした設計を採用します。これにより、新しいデータソースの追加や機能の拡張が柔軟に行えます。 2. AIと機械学習の活用 顧客インサイトの生成: 自然言語処理 (NLP) や機械学習を用いて、非構造化データから顧客のニーズやトレンドを抽出し、パーソナライズされたサービスを提供します。これにより、顧客接点の向上が期待できます。 予測分析: 顧客の行動を予測し、適切なタイミングでのアプローチを可能にするための予測モデルを開発します。これにより、顧客のライフサイクルに合わせたサービス提供が実現します。 3. リアルタイムデータ処理 リアルタイムでの監視: 顧客からのフィードバックやSNSの投稿をリアルタイムで分析し、迅速な対応ができる体制を整えます。これにより、顧客の声に迅速に対応し、信頼感を築くことができます。 ダッシュボードと可視化: データの可視化ツールを導入し、リアルタイムでのデータ分析結果を関係者が簡単に理解できるように構築します。 4. セキュリティとプライバシーの強化 データガバナンスの確立: 非構造化データの取り扱いに関するポリシーを明確にし、データのセキュリティとプライバシーを確保します。顧客の信頼を得るためには、透明性のあるデータ管理が不可欠です。 コンプライアンスの厳守: 法律や規制の变化に対応できる柔軟なシステム設計を行い、常に最新のコンプライアンスに準拠する体制を整えます。 5. 顧客中心のアプローチ 顧客体験の向上: 非構造化データを活用して、顧客の期待を超えた体験を提供することを目指します。顧客の声を反映したサービス改善やサービスの開発を行います。 フィードバックループの構築: 顧客からのフィードバックをシステムに組み込み、継続的な改善を行う仕組みを整えます。顧客の意見を反映することで、より良いサービスを提供できます。 6. 柔軟なシステムアーキテクチャ マイクロサービスアーキテクチャ: システムをマイクロサービスとして構築し、新機能を独立して開発・運用できるようになります。これにより、迅速な機能追加や変更が可能になります。 クラウドベースのインフラ: クラウド技術を活用し、スケーラビリティや可用性を確保します。最新の技術に応じてリソースを柔軟に拡張できる環境を整えます。 これらの要素を組み合わせることで、日立銀行の顧客管理システムは、10年後においても柔軟に対応し続ける強力なシステムとなるでしょう。顧客のニーズに迅速に応え、継続的な成長を支える基盤を築くことが重要です。10年後の日立銀行の顧客管理システムにおける非構造化データの活用を基盤とした新たなシステムの具体的な構築を以下に詳述します。 1. データ統合プラットフォームの構築 多様なデータソースの統合: データレイクの構築: SMS、顧客フィードバック、取引履歴、ウェブサイトの行動データなど、さまざまな非構造化データを統合するデータレイクを構築します。これにより、データの収集、保存、分析が一元化され、データのサイロ化を防ぎます。 APIファーストの原則: データの抽出、変換、ロード (ETL) プロセスを自動化し、リアルタイムでデータを更新できる仕組みを整えます。これにより、常に最新のデータを基にした分析が可能になります。 APIファーストアプローチ: オープンAPIの提供: 外部の開発者やパートナー企業がデータの活用を促進するためオープンAPIを提供し、エコシステムを拡大します。これにより、新しいサービスやアプリケーションの開発が促進されます。	いりパフォーマンスを管理するためのAPI管理プラットフォームを構築し、APIのライフサイクルを管理するためのツールを導入します。また、APIのセキュリティを強化し、不正アクセスを防ぐための対策を実施します。 2. AIと機械学習の活用 顧客インサイトの生成: 自然言語処理 (NLP) や機械学習を用いて、非構造化データから顧客のニーズやトレンドを抽出し、パーソナライズされたサービスを提供します。これにより、顧客接点の向上が期待できます。 予測分析: 顧客の行動を予測し、適切なタイミングでのアプローチを可能にするための予測モデルを開発します。これにより、顧客のライフサイクルに合わせたサービス提供が実現します。 3. リアルタイムデータ処理 リアルタイムでの監視: 顧客からのフィードバックやSNSの投稿をリアルタイムで分析し、迅速な対応ができる体制を整えます。これにより、顧客の声に迅速に対応し、信頼感を築くことができます。 ダッシュボードと可視化: データの可視化ツールを導入し、リアルタイムでのデータ分析結果を関係者が簡単に理解できるように構築します。 4. セキュリティとプライバシーの強化 データガバナンスの確立: 非構造化データの取り扱いに関するポリシーを明確にし、データのセキュリティとプライバシーを確保します。顧客の信頼を得るためには、透明性のあるデータ管理が不可欠です。 コンプライアンスの厳守: 法律や規制の变化に対応できる柔軟なシステム設計を行い、常に最新のコンプライアンスに準拠する体制を整えます。 5. 顧客中心のアプローチ 顧客体験の向上: 非構造化データを活用して、顧客の期待を超えた体験を提供することを目指します。顧客の声を反映したサービス改善やサービスの開発を行います。 フィードバックループの構築: 顧客からのフィードバックをシステムに組み込み、継続的な改善を行う仕組みを整えます。顧客の意見を反映することで、より良いサービスを提供できます。 6. 柔軟なシステムアーキテクチャ マイクロサービスアーキテクチャ: システムをマイクロサービスとして構築し、新機能を独立して開発・運用できるようになります。これにより、迅速な機能追加や変更が可能になります。 クラウドベースのインフラ: クラウド技術を活用し、スケーラビリティや可用性を確保します。最新の技術に応じてリソースを柔軟に拡張できる環境を整えます。 これらの要素を組み合わせることで、日立銀行の顧客管理システムは、10年後においても柔軟に対応し続ける強力なシステムとなるでしょう。顧客のニーズに迅速に応え、継続的な成長を支える基盤を築くことが重要です。10年後の日立銀行の顧客管理システムにおける非構造化データの活用を基盤とした新たなシステムの具体的な構築を以下に詳述します。 1. データ統合プラットフォームの構築 多様なデータソースの統合: データレイクの構築: SMS、顧客フィードバック、取引履歴、ウェブサイトの行動データなど、さまざまな非構造化データを統合するデータレイクを構築します。これにより、データの収集、保存、分析が一元化され、データのサイロ化を防ぎます。 APIファーストの原則: データの抽出、変換、ロード (ETL) プロセスを自動化し、リアルタイムでデータを更新できる仕組みを整えます。これにより、常に最新のデータを基にした分析が可能になります。 APIファーストアプローチ: オープンAPIの提供: 外部の開発者やパートナー企業がデータの活用を促進するためオープンAPIを提供し、エコシステムを拡大します。これにより、新しいサービスやアプリケーションの開発が促進されます。	10年後の日立銀行の顧客管理システムにおける非構造化データの活用を基盤とした新たなシステムのあり方について、以下のポイントを考慮することが重要です。 1. データ統合プラットフォームの構築 多様なデータソースの統合: SMS、顧客フィードバック、取引履歴など、さまざまな非構造化データを一元的に管理できるプラットフォームを構築します。これにより、データのサイロ化を防ぎ、全体的な顧客理解を深めることができます。 APIファーストアプローチ: 各部署がデータソースやサービスとの連携を容易にするため、APIを標準とした設計を採用します。これにより、新しいデータソースの追加や機能の拡張が柔軟に行えます。 2. AIと機械学習の活用 顧客インサイトの生成: 自然言語処理 (NLP) や機械学習を用いて、非構造化データから顧客のニーズやトレンドを抽出し、パーソナライズされたサービスを提供します。これにより、顧客接点の向上が期待できます。 予測分析: 顧客の行動を予測し、適切なタイミングでのアプローチを可能にするための予測モデルを開発します。これにより、顧客のライフサイクルに合わせたサービス提供が実現します。 3. リアルタイムデータ処理 リアルタイムでの監視: 顧客からのフィードバックやSNSの投稿をリアルタイムで分析し、迅速な対応ができる体制を整えます。これにより、顧客の声に迅速に対応し、信頼感を築くことができます。 ダッシュボードと可視化: データの可視化ツールを導入し、リアルタイムでのデータ分析結果を関係者が簡単に理解できるように構築します。 4. セキュリティとプライバシーの強化 データガバナンスの確立: 非構造化データの取り扱いに関するポリシーを明確にし、データのセキュリティとプライバシーを確保します。顧客の信頼を得るためには、透明性のあるデータ管理が不可欠です。 コンプライアンスの厳守: 法律や規制の变化に対応できる柔軟なシステム設計を行い、常に最新のコンプライアンスに準拠する体制を整えます。 5. 顧客中心のアプローチ 顧客体験の向上: 非構造化データを活用して、顧客の期待を超えた体験を提供することを目指します。顧客の声を反映したサービス改善やサービスの開発を行います。 フィードバックループの構築: 顧客からのフィードバックをシステムに組み込み、継続的な改善を行う仕組みを整えます。顧客の意見を反映することで、より良いサービスを提供できます。 6. 柔軟なシステムアーキテクチャ マイクロサービスアーキテクチャ: システムをマイクロサービスとして構築し、新機能を独立して開発・運用できるようになります。これにより、迅速な機能追加や変更が可能になります。 クラウドベースのインフラ: クラウド技術を活用し、スケーラビリティや可用性を確保します。最新の技術に応じてリソースを柔軟に拡張できる環境を整えます。 これらの要素を組み合わせることで、日立銀行の顧客管理システムは、10年後においても柔軟に対応し続ける強力なシステムとなるでしょう。顧客のニーズに迅速に応え、継続的な成長を支える基盤を築くことが重要です。10年後の日立銀行の顧客管理システムにおける非構造化データの活用を基盤とした新たなシステムの具体的な構築を以下に詳述します。 1. データ統合プラットフォームの構築 多様なデータソースの統合: データレイクの構築: SMS、顧客フィードバック、取引履歴、ウェブサイトの行動データなど、さまざまな非構造化データを統合するデータレイクを構築します。これにより、データの収集、保存、分析が一元化され、データのサイロ化を防ぎます。 APIファーストの原則: データの抽出、変換、ロード (ETL) プロセスを自動化し、リアルタイムでデータを更新できる仕組みを整えます。これにより、常に最新のデータを基にした分析が可能になります。 APIファーストアプローチ: オープンAPIの提供: 外部の開発者やパートナー企業がデータの活用を促進するためオープンAPIを提供し、エコシステムを拡大します。これにより、新しいサービスやアプリケーションの開発が促進されます。	10年後の日立銀行の顧客管理システムにおける非構造化データの活用を基盤とした新たなシステムのあり方について、以下のポイントを考慮することが重要です。 1. データ統合プラットフォームの構築 多様なデータソースの統合: SMS、顧客フィードバック、取引履歴など、さまざまな非構造化データを一元的に管理できるプラットフォームを構築します。これにより、データのサイロ化を防ぎ、全体的な顧客理解を深めることができます。 APIファーストアプローチ: 各部署がデータソースやサービスとの連携を容易にするため、APIを標準とした設計を採用します。これにより、新しいデータソースの追加や機能の拡張が柔軟に行えます。 2. AIと機械学習の活用 顧客インサイトの生成: 自然言語処理 (NLP) や機械学習を用いて、非構造化データから顧客のニーズやトレンドを抽出し、パーソナライズされたサービスを提供します。これにより、顧客接点の向上が期待できます。 予測分析: 顧客の行動を予測し、適切なタイミングでのアプローチを可能にするための予測モデルを開発します。これにより、顧客のライフサイクルに合わせたサービス提供が実現します。 3. リアルタイムデータ処理 リアルタイムでの監視: 顧客からのフィードバックやSNSの投稿をリアルタイムで分析し、迅速な対応ができる体制を整えます。これにより、顧客の声に迅速に対応し、信頼感を築くことができます。 ダッシュボードと可視化: データの可視化ツールを導入し、リアルタイムでのデータ分析結果を関係者が簡単に理解できるように構築します。 4. セキュリティとプライバシーの強化 データガバナンスの確立: 非構造化データの取り扱いに関するポリシーを明確にし、データのセキュリティとプライバシーを確保します。顧客の信頼を得るためには、透明性のあるデータ管理が不可欠です。 コンプライアンスの厳守: 法律や規制の变化に対応できる柔軟なシステム設計を行い、常に最新のコンプライアンスに準拠する体制を整えます。 5. 顧客中心のアプローチ 顧客体験の向上: 非構造化データを活用して、顧客の期待を超えた体験を提供することを目指します。顧客の声を反映したサービス改善やサービスの開発を行います。 フィードバックループの構築: 顧客からのフィードバックをシステムに組み込み、継続的な改善を行う仕組みを整えます。顧客の意見を反映することで、より良いサービスを提供できます。 6. 柔軟なシステムアーキテクチャ マイクロサービスアーキテクチャ: システムをマイクロサービスとして構築し、新機能を独立して開発・運用できるようになります。これにより、迅速な機能追加や変更が可能になります。 クラウドベースのインフラ: クラウド技術を活用し、スケーラビリティや可用性を確保します。
--	---	--	--

アプリケーション生成と同時に関連ドキュメントも生成

©Hitachi, Ltd. 2025. All rights reserved

操作イメージ

実行前



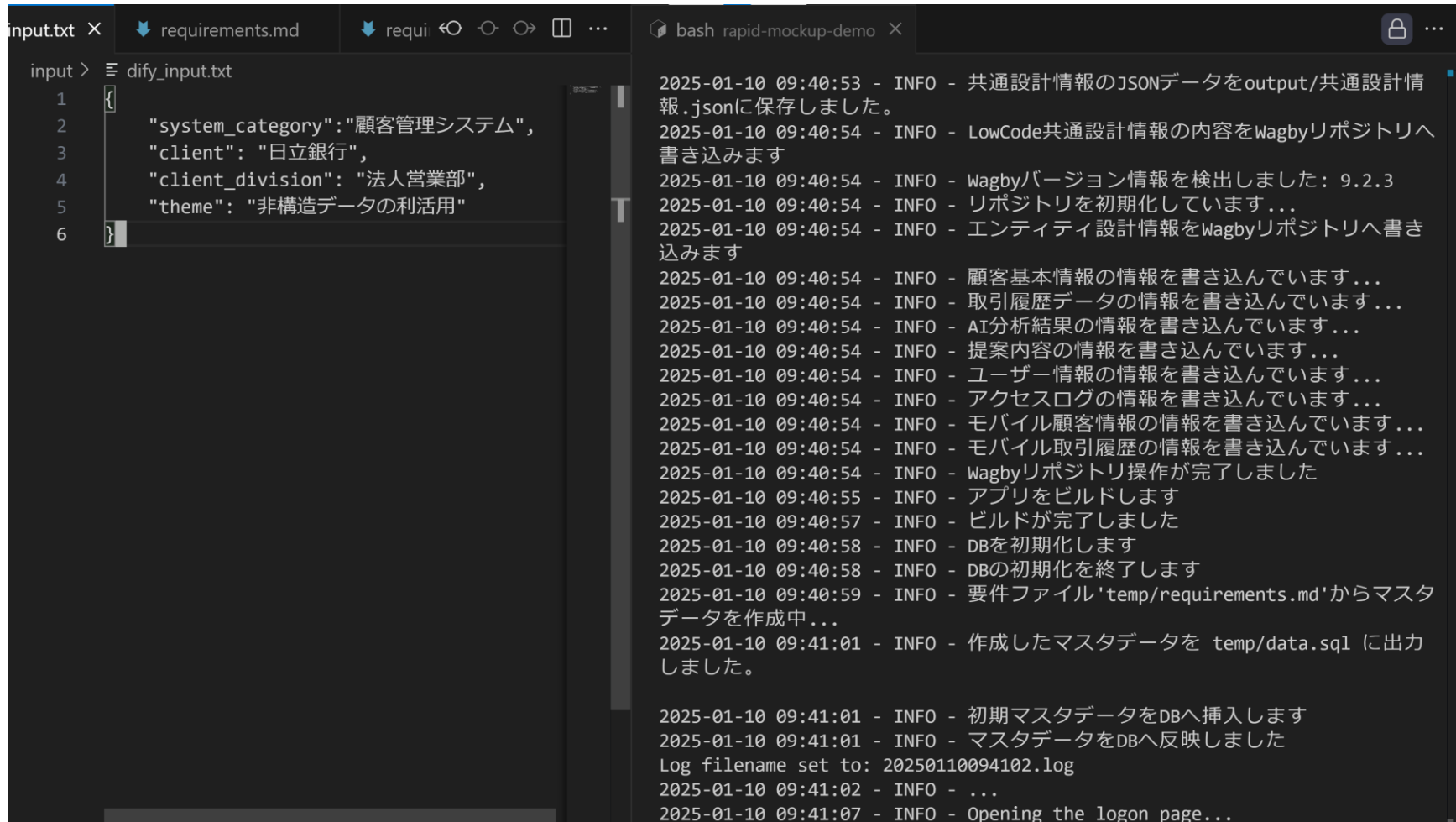
The image shows a code editor and a terminal window side-by-side. The code editor on the left has a tab labeled 'dify_input.txt' and contains a JSON object with Japanese text. The terminal window on the right has a tab labeled 'bash rapid-mockup-demo' and shows the output of a shell command, including the current directory and branch.

```
input > dify_input.txt
1 {
2   "system_category": "顧客管理システム",
3   "client": "日立銀行",
4   "client_division": "法人営業部",
5   "theme": "非構造データの利活用"
6 }
```

```
(.devenv)
<WSL2>yotsuji@S670034412:~/gitlab/rapid-mockup-demo (develop)
$
```

操作イメージ

実行中



The screenshot shows a code editor with two tabs: 'input.txt' and 'requirements.md'. The 'input.txt' tab is active, displaying a JSON file named 'dify_input.txt' with the following content:

```
1 {  
2   "system_category": "顧客管理システム",  
3   "client": "日立銀行",  
4   "client_division": "法人営業部",  
5   "theme": "非構造データの利活用"  
6 }
```

The 'requirements.md' tab is also visible, showing a file named 'requi'. Below the code editor, a terminal window titled 'bash rapid-mockup-demo' displays a series of log messages:

```
2025-01-10 09:40:53 - INFO - 共通設計情報のJSONデータをoutput/共通設計情報.jsonに保存しました。  
2025-01-10 09:40:54 - INFO - LowCode共通設計情報の内容をWagbyリポジトリへ書き込みます  
2025-01-10 09:40:54 - INFO - Wagbyバージョン情報を検出しました: 9.2.3  
2025-01-10 09:40:54 - INFO - リポジトリを初期化しています...  
2025-01-10 09:40:54 - INFO - エンティティ設計情報をWagbyリポジトリへ書き込みます  
2025-01-10 09:40:54 - INFO - 顧客基本情報の情報を書き込んでいます...  
2025-01-10 09:40:54 - INFO - 取引履歴データの情報を書き込んでいます...  
2025-01-10 09:40:54 - INFO - AI分析結果の情報を書き込んでいます...  
2025-01-10 09:40:54 - INFO - 提案内容の情報を書き込んでいます...  
2025-01-10 09:40:54 - INFO - ユーザー情報の情報を書き込んでいます...  
2025-01-10 09:40:54 - INFO - アクセスログの情報を書き込んでいます...  
2025-01-10 09:40:54 - INFO - モバイル顧客情報の情報を書き込んでいます...  
2025-01-10 09:40:54 - INFO - モバイル取引履歴の情報を書き込んでいます...  
2025-01-10 09:40:54 - INFO - Wagbyリポジトリ操作が完了しました  
2025-01-10 09:40:55 - INFO - アプリをビルドします  
2025-01-10 09:40:57 - INFO - ビルドが完了しました  
2025-01-10 09:40:58 - INFO - DBを初期化します  
2025-01-10 09:40:58 - INFO - DBの初期化を終了します  
2025-01-10 09:40:59 - INFO - 要件ファイル'temp/requirements.md'からマスターデータを作成中...  
2025-01-10 09:41:01 - INFO - 作成したマスターデータを temp/data.sql に出力しました。  
  
2025-01-10 09:41:01 - INFO - 初期マスターデータをDBへ挿入します  
2025-01-10 09:41:01 - INFO - マスターデータをDBへ反映しました  
Log filename set to: 20250110094102.log  
2025-01-10 09:41:02 - INFO - ...  
2025-01-10 09:41:07 - INFO - Opening the logon page...
```

モックアップ自動生成ツールのイメージ

