

AI活用による開発・テスト自動化

2025.07



システム開発における課題

多くの企業が開発プロセスのボトルネックやテスト工数の増大という課題を抱えており、ビジネス機会の損失や品質低下を招き、人材不足を深刻化させ、企業の成長を妨げる要因となっています

開発プロセスのボトルネック

- 要件定義の曖昧さによる手戻りの多発とコミュニケーションコスト増大
- 属人化したプロセスとノウハウのブラックボックス化
- 複雑な業務ロジックの理解と実装にかかる膨大な時間
- 既存システム（ERR、SCMなど）との連携の複雑性

ビジネスへの深刻な影響

- 市場投入の遅れによる機会損失
- 品質問題による顧客信頼の低下、ブランドイメージの毀損
- セキュリティリスクの増大とコンプライアンス対応の負荷
- コスト増加による収益性の圧迫

テスト工数の肥大化と品質担保の限界

- テストケース作成・実行・管理の膨大な工数
- リグレッションテストの負担増と網羅性の低下
- テスト環境構築・維持のコストと手間
- 人手によるテストの見逃しリスクと品質のばらつき

エンジニアの疲弊と人材不足

- 単調な繰り返し作業によるモチベーション低下
- 長時間労働の常態化とバーンアウト
- 高度なスキルを持つエンジニアの採用難と育成の遅れ
- 日本企業では、特に人材不足が深刻な問題として認識

開発プロセスのボトルネックと非効率化

開発上のボトルネックは、市場投入の遅れ、品質問題、コスト増加、エンジニアの疲弊といったビジネス上の深刻な問題に直結してしまいます

手戻り多発とコミュニケーションコスト増大

要件定義の曖昧さや仕様変更の頻発により、開発工程での手戻りが多発し、プロジェクト遅延とコスト増大の主要因となっています
特にB2Bシステムでは、多数のステークホルダーが関与し、部門間の認識齟齬がコミュニケーションコストをさらに押し上げます

属人化したプロセスとノウハウのブラックボックス化

特定の担当者に依存した開発プロセスや、ドキュメント化されていない暗黙知は、担当者の不在や退職時に業務停滞リスクを生み出します
ノウハウが組織的に共有・継承されず、ブラックボックス化することで、組織全体の開発力向上が阻害されます

複雑な業務ロジックの理解と実装の負担

B2Bシステムは、業界特有の商習慣や法規制、企業独自の複雑な業務プロセスを反映する必要があります
これらのロジックを正確に理解し、システムに実装するには多大な時間と高度なスキルが要求され、開発の大きなボトルネックとなっています

既存システムとの連携の複雑性と困難さ

多くのB2B企業では、長年運用されてきた基幹システム（ERP、SCMなど）や多数のSaaSとの連携が不可欠です
これらのシステム間のデータ連携やプロセスの整合性を確保することは技術的に難易度が高く、開発の大きな負担となっています

テスト工数の肥大化と品質担保の限界

システムの複雑化に伴いテスト工数は増大し、リグレッションテストの負担や環境維持コストも増加します。また、人的ミスや品質のばらつきもあり、増え続ける工数の中での品質担保には限界があります

テストケース作成・実行・管理の膨大な工数

システムの複雑化に伴い、テストケースは指数関数的に増加します

これらのテストケースを手動で作成、実行し、結果を管理するには膨大な工数が必要となり、開発期間の大きな部分を占めています

複雑なシステムほどテストケースは増加し、管理負荷も上昇

リグレッションテストの負担増と網羅性の低下

機能追加や改修のたびに実施されるリグレッションテストは、変更が広範囲に及ぶほどテスト対象が増加し、開発チームの大きな負担となります

限られたリソースと時間の中で、十分な網羅性を確保することが困難になりつつあります

変更範囲の拡大に比例してテスト負担も増大

テスト環境構築・維持のコストと手間

本番環境に近いテスト環境を複数用意し、常に最新の状態に維持するには、多大なコストと手間がかかります

特に、多様なOS、ブラウザ、デバイス、外部システム連携を考慮した環境構築は複雑です

環境構築の複雑さがコストと時間を圧迫

人手によるテストの見逃しリスクと品質のばらつき

手動テストは、テスト担当者のスキルや集中力により品質が左右されやすく、ヒューマンエラーによるバグの見逃しリスクが常に存在します

これにより、品質のばらつきが生じ、安定した品質供給が難しくなります

人的要因による品質のばらつきが避けられない

自動テスト導入における課題

多くの企業が開発プロセスのボトルネックやテスト工数の増大という課題を抱えており、ビジネス機会の損失や品質低下を招き、人材不足を深刻化させ、企業の成長を妨げる要因となっています

限定的な自動化範囲

- テスト実行の一部や特定のビルドプロセスなど、限定的なタスクの自動化に留まる
- 要件定義、設計、複雑なテストシナリオ作成など、高度な判断を要する領域の自動化が困難

自動テスト設定・スクリプト保守の負担

- UIや仕様の変更により、自動テストの設定変更やスクリプト修正・再設定が頻発し、メンテナンス工数がかさむ
- 結果として、変化の激しい環境では、自動化の維持自体が開発現場の新たな負担となるケースも多い

複雑なシナリオへの対応不足

- B2B特有の複雑な業務フローや多数の分岐条件を含むシナリオの自動化は困難
- ユーザビリティ、パフォーマンス、セキュリティといった非機能要件のテスト自動化は、従来型ツールではカバーしきれない

その他の課題

初期投資と費用対効果の問題

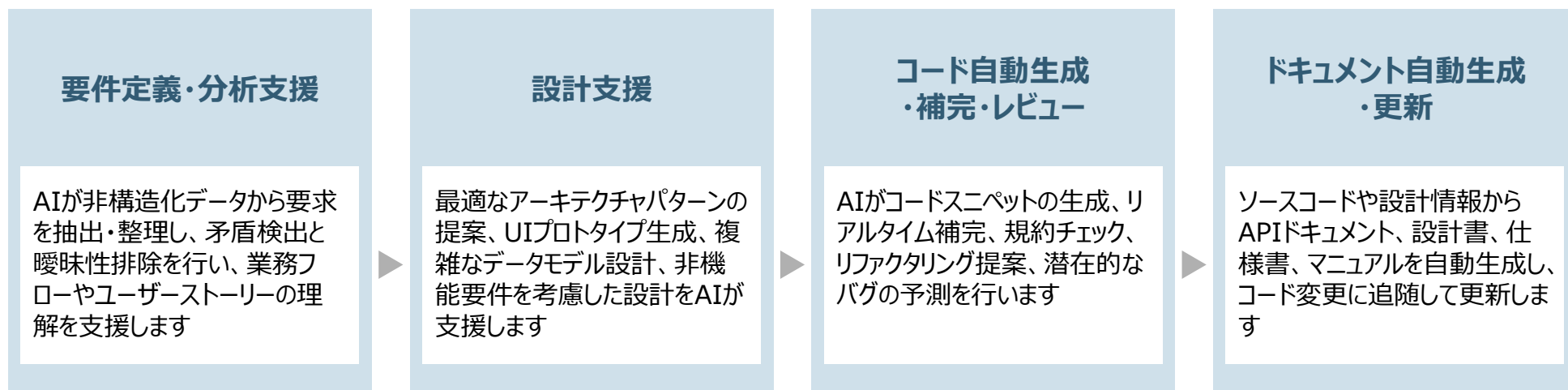
自動化のコストが手動作業のコストを上回るケースも存在

ツール自体の限界

特定の機能やフレームワーク、環境設定をサポートしていない場合がある

AIによる開発プロセスの自動化・高度化

開発ライフサイクルの上流工程から下流工程に至るまで、多岐にわたる領域でAIが介在することで、開発者はより創造的な業務に集中し、生産性と品質の向上を実現できる



AIによる開発プロセス自動化の主な効果

開発スピードの大幅な向上と手戻りの削減

コードの品質と一貫性の強化

エンジニアの創造的業務への集中を実現

従来手法との比較優位性

AIの自然言語処理は要件定義プロセスの変革をもたらすことができ、例えば議事録などから要求を自動で抽出し、矛盾や曖昧さを排除、さらに複雑な業務フローの理解やトレーサビリティを確保し、手戻りのない高品質な要件定義が実現できる

AIによる要件定義プロセス改革



自然言語処理による要求抽出・整理

顧客からのヒアリング議事録、メール、提案依頼書といった非構造化テキストデータから、AIが重要な要求事項を自動的に抽出・分類・構造化します
これにより、要求の抜け漏れを防ぎ、手作業による整理の負担を大幅に軽減します

矛盾検出と曖昧性排除

複数のドキュメントや過去の類似プロジェクトの仕様書をAIが分析し、要求間の矛盾点や曖昧な表現を検出します
これにより、早期に問題を発見し、手戻りを未然に防ぎます

B2B特有の業務フロー・業界標準の理解支援

金融業界におけるSWIFT連携、製造業におけるEDI標準、医療業界におけるHL7/FHIRなど、B2Bシステムでは業界特有の複雑な業務フローや標準規格への対応が不可欠です
AIはこれらのドキュメントを学習し、開発者が迅速に理解するためのサマリー生成や、関連情報の提示を行います

トレーサビリティの確保

要件と設計、コード、テストケース間の関連性をAIが追跡し、トレーサビリティマトリクスの作成を支援します
これにより、変更管理の効率化と影響範囲の正確な把握が可能になります

AIによる設計コーディング

AIによる最適なアーキテクチャの提案、UI/UXデザインの自動生成、リアルタイムのセキュリティチェックといった支援により、開発の生産性・品質・安全性を大幅に向上させることができます



アーキテクチャパターン提案

- プロジェクト特性や非機能要件（パフォーマンス、スケーラビリティ、セキュリティなど）に基づく最適なアーキテクチャパターン提案
- マイクロサービス、イベント駆動など、プロジェクトに適した技術スタックの選定支援
- 複雑なデータモデルや連携インターフェース設計の効率化と整合性確保



UI/UXデザイン自動生成

- 基本的な要件やターゲットユーザー情報を基にUIデザインのプロトタイプを自動生成
- 過去のユーザー行動データやヒューリスティック評価の知見を活用したユーザビリティ問題の予測
- ローコード／ノーコードプラットフォームとの連携による開発初期段階の自動化



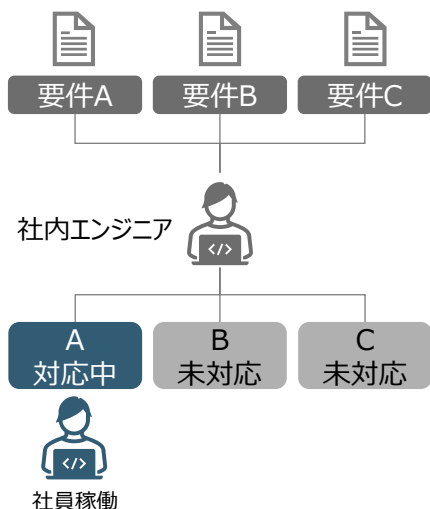
セキュアコーディング支援

- リアルタイムでコードを分析し、コーディング規約からの逸脱や既知のセキュリティ脆弱性を検出
- SQLインジェクション、クロスサイトスクリプティングなどの脆弱性に対する修正案の提示
- OWASP Top 10 などのセキュリティ標準に基づいたコーディングプラクティスの提案

AIエージェントを活用した開発の出力最大化

AIエージェントはfigmaデザインを元にしたコーディングやテストの自動化なども対応が可能であり、AIエージェント型組織を構築すれば、従来の開発リソースで数倍ものアウトプットを出すことも可能です

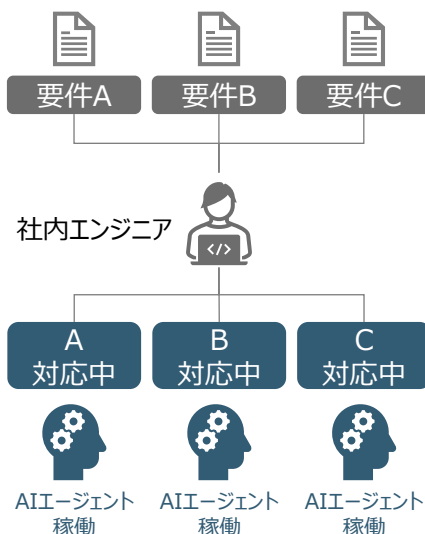
従来型組織



従来型組織の課題

- 社内の業務が属人化しており、後任の育成が難しい/または時間がかかる
- 休職や退職に伴い、社内業務を熟知している人材が急にいなくなるリスクがある
- 人材を採用して組織を安定させたいが、採用費や内定承諾率の変動が起こり採用が計画通りに進まない

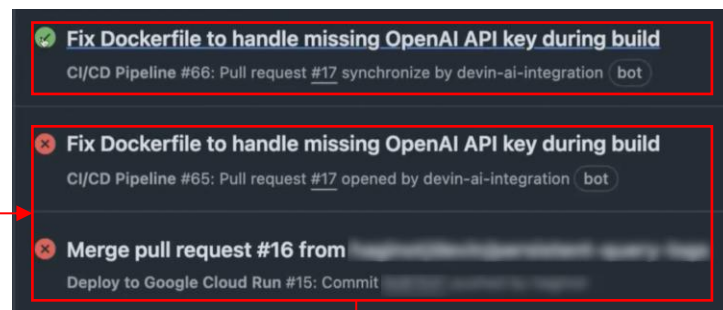
AIエージェント型組織



従来型組織の課題

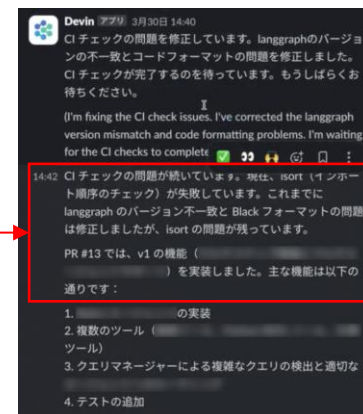
- 人間と同じようにやりとりが可能。下記はslackにてAIエージェントのDevinとのやりとりの例
- AIエージェントは既存システムのソースコードを学習していき、システムの仕様に習熟した開発者に育ちます

AIエージェントによる「開発⇄テスト」の取り組みの例



AIエージェントが自身で開発した内容をテストUIのスクリーンショットまで自動で投稿

AIエージェントが自身でコードの修正作業を実施修正が完了してテスト結果も問題なければ自動でプルリクエストを実施



AIによるテストプロセスの革新

AIによるテスト自動化は、システムの複雑性から増大するテスト工数を効率化し、テストの計画から実行、分析に至るまでのテストプロセス全体を革新的に変革させることができます



テスト計画・設計支援

- コード変更箇所、モジュールの複雑性、過去の障害履歴、ビジネス重要度を分析し、リスクベースのテスト戦略を立案
- 既存テストケースと要件・仕様の照合によるテストカバレッジの網羅性分析と最適化提案
- B2B特有の複雑なテストシナリオ（複数ユーザーロール、承認ワークフロー、外部システム連携）の設計支援



テストケース・データ自動生成

- 仕様書・要件定義書から自然言語処理技術を活用したテストケースの自動生成
- 正常系、異常系、境界値、組み合わせパターンなど多様なテストデータの自動生成
- 本番データの匿名化・マスキングと統計的特性を維持した高品質な合成データ生成
- B2B特有の複雑なデータ構造（取引データ、製品構成データなど）に対応した整合性のあるテストデータ生成



テスト環境構築・管理の自動化

- AIがテスト要件に応じて必要な環境構成を判断し、IaC (Infrastructure as Code) ツールと連携
- TerraformやAnsibleなどのツールを活用したテスト環境の自動プロビジョニング・設定・破棄
- 開発環境、ステージング環境、本番互換環境など、複数の環境間での設定差異検出と一貫性維持

開発テスト自動化AIエージェント構築プロセス

AIエージェントに対して、文書化された開発要件とユビキタス言語を取り込んだ上で、テストエンジニアによるAIエージェントの育成を経て、開発テストの完全自動化を実現する

開発要件の文書化

AIエージェントにおけるテストの自動化においては、文書化された開発要件と開発対象物のユビキタス言語を整理する

1

AIエージェントはテキストデータの処理を得意とするため、開発要件をテキストデータ化する

+

2



AIエージェントが開発要件を正しく理解できるようにユビキタス言語の一覧を整理する

||



AIエージェント

要件を正確に理解できました！



AIエージェントによるテストケースの生成

AIエージェントに各種情報をプロンプトに入れて学習させることで、AIエージェント自体にテストケースを考えられるようにして、テストケースの作成時間を削減していく

テキスト



プロンプト

AIエージェント

プロンプト

あなたはテスト設計の専門家です。以下の情報を基に、商品購入数量の境界テストケースを作成してください。【機能】：商品購入数量入力【制約条件】： - 購入可能数量：1～99個 - 入力形式：半角数字 - 在庫制限：あり【学習データ】：

テストエンジニアによるAIエージェントの育成

テストエンジニアの役割をAIにアドバイスをする先生とすることで、実務知見にも習熟したAIエージェントとなり自立が可能となる状態へと育てる

このテストケースは境界値が足りないよ

この観点が抜けているね

もっとこんなケースも試してみて



テストエンジニア

アウトプット提示

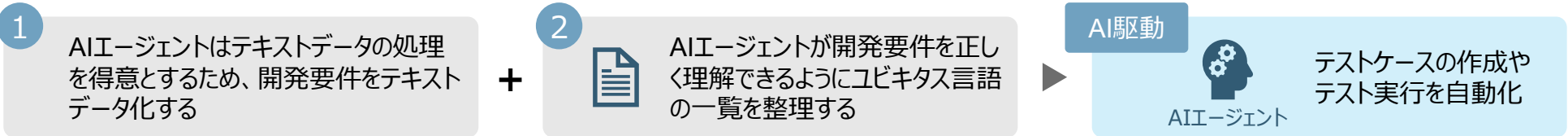
フィードバック



AIエージェント

テスト自動化に向けて

AIエージェントにおけるテストの自動化においては、文書化された開発要件と開発対象物のユビキタス言語を整理しておきたい



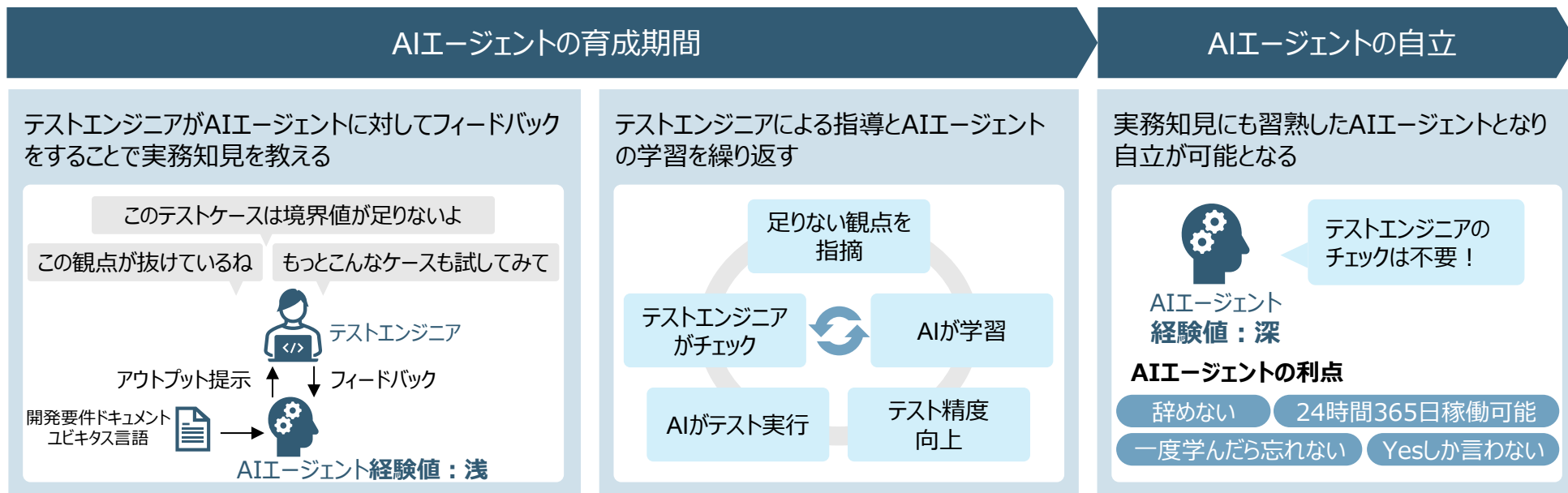
項目	想定イメージ	ユースケース・ロジック
アクティビティ情報エリアの移動	0.5秒長押しでその曜日のプレートが浮き上がる プレートの位置の移動動作に伴い曜日と日付の表示も変更される ・指が触れている間は、その曜日のプレートの位置を移動できる ・指を離すと一番、その配置に近い形で、その他の曜日にその時点で位置に近い形で、その他の曜日に曜日とそのプレートが入っていたプレートも配置される ・その配置に対応する ・配置を変更する	0.5秒長押しでスケジュール調整に対象のアクティビティ情報エリアが浮き上がる - 長押ししている間は浮き上がった状態が続く - 浮き上がった状態の間は、アクティビティ情報エリアの配置を変更することができる - 浮き上がった状態の間は、アクティビティ情報エリアは指の移動に合わせて上下移動ができる - 指をスマホ画面から離したら、その時点のアクティビティ情報エリアの位置から最も近い、曜日にアクティビティ情報エリアは配置される - 上記のアクティビティ情報エリアの配置に伴い、そのほかの日時に位置していたアクティビティ情報エリアの位置も移動する 月、曜日・日付は固定されており、アクティビティだけ動かせるUI - アクティビティは最初に配置された日時から+6日まで移動可能、また、最初に配置された週（月～日）の中であれば、－●日（最初に配置された週を跨がない限り）まで移動可能 ※ただし、過去には移動ができないので注意 - 曜日・日付は固定されており、アクティビティだけ動かせるUIになる - そのためこの日付の部分は、アクティビティのプレートを動かしても固定されたままで動かない - アクティビティ情報エリアの配置完了直後に、下記条件をユーザーの挙動が満たしたときポップアップでアラームのコメントを表示する
アクティビティ情報エリアの日付・曜日表示		
アクティビティ情報エリアのアラート表示	指定した条件をユーザーの挙動が満たしたときのみポップアップでアラームのコメントを表示する。	<表示ロジック> - ロジック1：週n回のプランを選択中において、連続でn日ランニングをスケジュールリングした場合、プレート配置完了に合わせて、「アクティビティの予定がn日続いています。怪我を避けるためにも、間に休息日ははさみましょう。」とアラームコメントを表示する - ロジック2：アクティビティには「ハード」と「ゆるめ」のどちらかの種類を分類するフラグが立てられています。「ハード」にフラグ立てされたアクティビティが連続して配置された場合、「負荷が強めのアクティビティが連続しています。出来るだけ中1日は間隔をあげましょう。」というアラームコメントを表示する - ロジック1と2を同時に満たす場合：ロジック1のアラームコメントを表示して、ロジック2のアラームコメントは表示しない - 左上のボタンを押すとポップアップを閉じることができる



AI駆動時におけるテストエンジニアの役割

テストエンジニアの役割をAIにアドバイスをする先生とすることで、実務知見にも習熟したAIEージェントとなり自立が可能となる状態へと育てる

従来の役割	ソフトウェアが正しく動くかをチェックする役割 - テストケースを作る: 「こんな操作をしたらどうなるか?」という質問リストを作成 - テスト方法を考える: 効率的にバグを見つけるための手順を設計
今後の役割	AIにアドバイスをする先生



補足：境界テストケース

1

AIエージェントはテキストデータの処理を得意とするため、開発要件をテキストデータ化する

AIエージェントの活用において、開発要件は「境界テストケース（ユーザーの状態をMECEにして、各状態でのバックエンドとの送受信データやUIでの表示内容、挙動等が全て定義されている状態）」でテキストで表現されていると理想的である



要素	例	表示ロジック	データロジック	関連画面
コメント	テストラン 10km 5:00/km以内のペースを目標に走りましょう	「アイコン」と「テストラン」という記載「大角重人コーチ」という記載は固定 「XXkmをX:XX/kmより早いペースで走るのを目指しましょう！」というテキストはXXの部分が可変	左記のXXの部分は、個々のテストランが有する指定距離と指定ペースが表示される（例：10kmを5:00/kmより早いペースで走るのを目指しましょう！）	特になし
計測エリア	通常のラン計測と同じ機能			
インジケーター		- インジケーターの横幅の長さは常に固定 - 走った距離に応じてインジケーターが右に色が変わって進んでいく 1%刻みでインジケーターが進んでいく	インジケーターの左端は走行距離0km、右端は走行距離を指定距離とする	特になし
計測終了ロジック		- 指定走行距離まで到達したら、強制的にランニングの計測を終了して、画面の停止中表示になる - 強制終了時にiOSのテキスト音声で「お疲れ様でした。テストランが終了しました。」と流れ、バイブレーションが2回振動する	指定走行距離まで到達したら、強制的にランニングの計測を終了して、再度計測を再開することはできない仕様になる	テストランを終了して、結果画面に遷移する。

ランの一時停止中の表示・ロジックの詳細

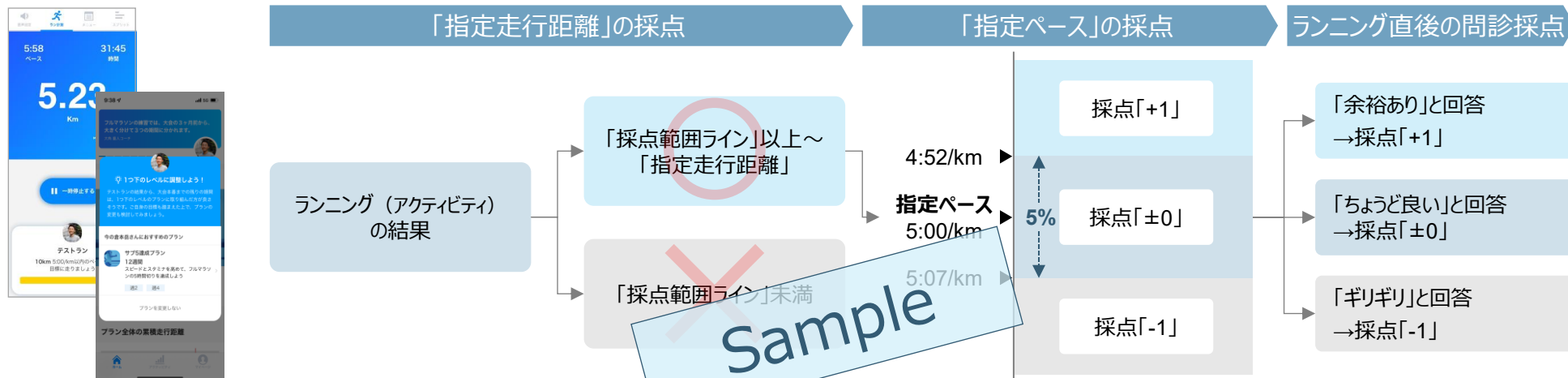
「指定距離」未到達時にユーザーが自分でラン計測と停止して、ランの一時停止中画面になった場合の画面	「指定距離」到達時にランの計測が強制終了された場合の画面
- 通常の場合と基本的には同じですがいくつか相違点があります <相違点> - テストラン計測中の画面と同様に、コメントとインジケーターのエリアが表示されています	- 指定走行距離まで到達した、計測を強制終了するため、「テストランの結果をみる」のボタンのみが表示される 「テストランの結果をみる」のボタンを押すと、テストランの結果画面に遷移する - コメントエリアのコメントは「ナイスラン！お疲れ様でした。」となる - ユーザーには、バイブレーションと音声で、終了したことを伝える。音声は「お疲れ様でした。テストランが終了しました。」をiOSのテキスト音声で流す

補足：境界テストケース

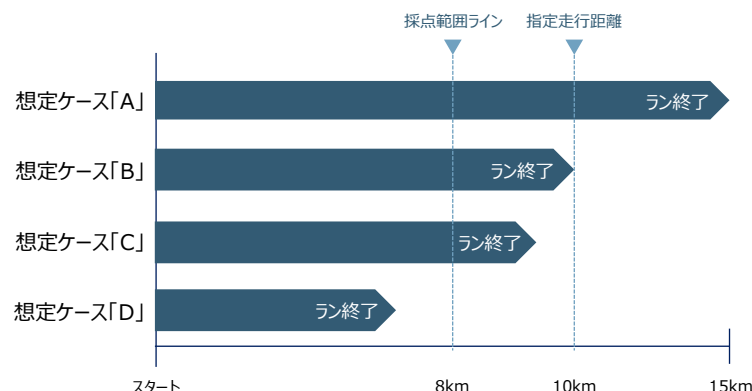
1

AIエージェントはテキストデータの処理を得意とするため、開発要件をテキストデータ化する

AIエージェントの活用において、開発要件は「境界テストケース（ユーザーの状態をMECEにして、各状態でのバックエンドとの送受信データやUIでの表示内容、挙動等が全て定義されている状態）」でテキストで表現されていると理想的である



「10km」を「5:00/km」で走ろう！のテストランの場合



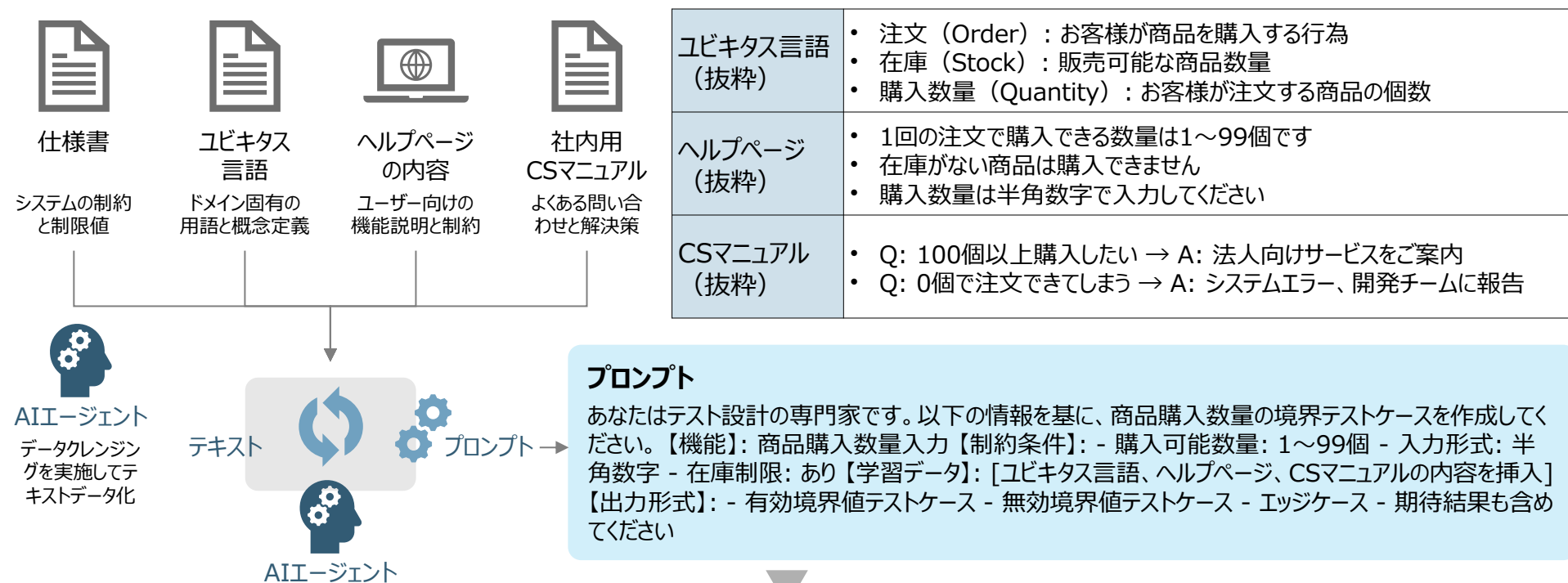
指定距離の採点	指定ペースの採点	ランニング直後の問診	採点結果	マッチ率の判定
採点範囲ライン以上だった場合	指定ペース 5% 上限より早い	「余裕あり」と回答	+1点	2点
		「ちょうど良い」と回答	±0点	1点
		「ギリギリだった」と回答	-1点	0点
	指定ペース 5% 範囲内	「余裕あり」と回答	+1点	1点
		「ちょうど良い」と回答	±0点	0点
		「ギリギリだった」と回答	-1点	-1点
採点範囲ライン以上未達だった場合	指定ペース 5% 加減より遅い	「余裕あり」と回答	+1点	0点
		「ちょうど良い」と回答	±0点	-1点
		「ギリギリだった」と回答	-1点	-2点
採点範囲ライン以上未達だった場合	不合格	評価なし	-2点	プランのレベルダウンを提案

AIエージェントによるテストケースの生成

1

AIエージェントはテキストデータの処理を得意とするため、開発要件をテキストデータ化する

AIエージェントに各種情報をプロンプトに入れて学習させることで、AIエージェント自体にテストケースを考えさせられるようにして、テストケースの作成時間を削減していく



有効境界値	TC001	数量=1 → 注文成功
	TC002	数量=99 → 注文成功
	TC003	数量=2 → 注文成功
	TC004	数量=98 → 注文成功

無効境界値	TC005	数量=0 → エラー「1個以上選択してください」
	TC006	数量=100 → エラー「99個まで購入可能です」
	TC007	数量=-1 → エラー「正の数を入力してください」

エッジケース	TC008	在庫1個、数量2 → エラー「在庫不足です」
	TC009	数量「abc」 → エラー「半角数字で入力してください」
	TC010	数量「1.5」 → エラー「整数で入力してください」
	TC011	空文字 → エラー「数量を入力してください」

ユビキタス言語の取り込み

2



AIエージェントが開発要件を正しく理解できるようにユビキタス言語の一覧を整理する

AIエージェントがあらかじめユビキタス言語（開発チーム全体が共通して使用する言語として用語や概念を定義したもの）を取り込んでおくことで、AIエージェントはテストケースを正しく理解することができる



親概念	子概念	孫概念	ひ孫概念	概要
スケジュール	—	—	—	プロセス、実施頻度に合わせて、いつ、何をすれば良いのかの日程表。プランの開始日から終了予定日までのアクティビティの行動計画である。
スケジュール	単位	—	—	スケジュールは1日単位である。
スケジュール	デイリースケジュール	—	—	スケジュールの1日単位の名前をデイリースケジュールとする。
スケジュール	実施制約	—	—	スケジュール内において、アクティビティは1日に最大1回まで実施可能。※フリーランは除外
スケジュール	アサイン	—	—	ユーザーが1日1つアクティビティをアサインされている。アサインされているアクティビティを実施した場合、その日中に他のアクティビティの実施はできない。
スケジュール	実施済み・未実施	—	—	実施済みのアクティビティは再度実施することができない。
スケジュール	頻度変更	—	—	ユーザーが選択中のプランの頻度を変更する。（例：フルマラソン完走プラン週4日ver↔フルマラソン完走プラン週2日ver）
スケジュール	頻度変更	アクティビティwith run分類	—	アクティビティwith run分類をハード目とゆる目の2つに分類する。
スケジュール	頻度変更	アクティビティwith run分類	ゆる目	アクティビティwith runを2つに分類した際の、負荷が低い方のアクティビティ。
スケジュール	頻度変更	アクティビティwith run分類	ハード目	アクティビティwith runを2つに分類した際の、負荷が高い方のアクティビティ。
スケジュール	頻度変更	ペア	—	同じプランで、頻度の異なる2つのプラン（例：フルマラソン完走プラン週4日ver↔フルマラソン完走プラン週2日ver）を比べた際に、同じ週にプリセットで配置された、ハード目のアクティビティは1：1対応でペアの関係性を有する。
スケジュール	頻度変更	ペア	消化	同じプランで、頻度の異なる2つのプラン（例：フルマラソン完走プラン週4日ver↔フルマラソン完走プラン週2日ver）を比べた際に、ユーザーが選択中の頻度で、1つのハード目のアクティビティを消化（＝実施済みになる）した際に、もう片方の頻度のペアのアクティビティも消化したと見なされて、プランの頻度変更を行う際に、残りアクティビティから除外される。
スケジュール	頻度変更	優先順位	—	ユーザーがその週に配置された全てのアクティビティを取り組むと仮定した際に、アクティビティに取り組み際の、取り組んで欲しい優先順位。
スケジュール	頻度変更	主従関係	—	ゆる目とハード目のアクティビティは、取り組む順番を決める際に、1：1のペアになり、「ゆる目→ハード目」の順番で取り組むことが推奨される。
スケジュール	頻度変更	配置	—	アクティビティを1週間の中でどの曜日に配置するのかということ。
スケジュール	頻度変更	可処分日数	—	プランの頻度変更を行う際に、その週に残された日数。
スケジュール	頻度変更	残りアクティビティ	—	プランの頻度変更を行う際に、変更先の頻度のプランでペア消化されずに残されたアクティビティの数。
スケジュール	頻度変更	配置ルール	—	アクティビティを配置する際のルール。
スケジュール	頻度変更	配置ルール	NGボタン1	ハード目のアクティビティを連続して配置すること。
スケジュール	頻度変更	配置ルール	NGボタン2	週n日のプランで、n日連続でアクティビティを配置すること。
スケジュール	頻度変更	配置ルール	アラーム	NGボタン1、2のどちらかがユーザーがスケジュール調整をしてアクティビティを配置した場合に表示する注意喚起画面
アクティビティ	—	—	—	スケジュールのなかで、1日にこなすべきワークアウトのメニューとその目的のまとめ。アクティビティには4つのタイプをもつ
アクティビティ	アクティビティ with Run	—	—	ランニングの計測機能を用いるアクティビティ。アクティビティ with Run（仮）は1つのウォームアップと1つのクールダウンを持つ
アクティビティ	アクティビティ without Run	—	—	ランニングの計測機能を用いないアクティビティ

AI活用による開発工程における効果

AI活用により、開発リードタイムやテスト工数を大幅に削減し、また品質向上による手戻り・修正コストの削減も実現することで、エンジニアをより創造的な業務に集中させ生産性の高い開発が可能になる

開発リードタイムの大幅短縮

要件定義・設計期間

20~40%

削減

コーディング時間

最大50%

削減

レビュー時間

30%以上

削減

- これらの効率化により、新機能や新サービスの市場投入までの期間を大幅に短縮し、ビジネスチャンスを迅速に捉えることが可能になります

開発リードタイムの大幅短縮

リリース後の不具合発生率

30~60% 削減

- 手戻り工数の削減：**
上流工程での矛盾検出や、開発中の早期バグ発見により、後工程での大規模な手戻り工数を大幅に削減
- バグ修正コストの削減：**
バグは発見が遅れるほど修正コストが増大する傾向がありますが、AIによる早期発見・早期修正により、トータルの修正コストを大幅に抑制

テスト工数とコストの削減

テストケース作成・実行時間

50~80%

削減

テスト仕様書作成工数

約40%

削減

- テスト環境コスト削減**
AIとIaCの連携によるテスト環境自動化は、環境構築・維持にかかるインフラコストや人件費を削減

生産性の向上

エンジニア一人当たりの生産性

10~30% 向上

- 開発チーム全体のキャパシティ向上：**
チーム全体の生産性が向上することで、より多くのプロジェクトや機能開発に同時に取り組むことが可能に
- 創造的業務への集中：**
AIが単調な作業を担当することで、エンジニアはより高度な分析や設計など、付加価値の高い業務に注力可能

定性的効果とROIの最大化

AI導入により製品品質やエンジニアの意欲、ビジネスの俊敏性を高める定性的効果も見込め、短期的なコスト削減だけでなく、長期的なビジネス価値向上によりROIを最大化できます

製品・サービスの品質向上

- テストカバレッジの向上による潜在バグの削減：AIは人間が見逃しがちなエッジケースや複雑な組み合わせを網羅的にテスト
- ユーザーエクスペリエンス（UX）の向上：品質の高いシステムによる使いやすさと満足度の向上
- システムの安定性と信頼性の向上：顧客の業務継続性を担保し、企業への信頼を強化

エンジニアのエンゲージメント向上

- 単調な作業からの解放と創造的・戦略的業務への集中
- 新しい技術習得の機会とスキルアップ
- 働きがいと満足度の向上、離職率の低下

ビジネスアジリティの向上

- 市場の変化や顧客ニーズへの迅速な対応
- 新機能の迅速なリリースと競争力の強化
- イノベーションの促進と市場シェア拡大

ROI（投資対効果）の最大化

投資と継続コストの把握

初期投資（ライセンス費用、導入コンサル費用、インフラ構築費用、人材育成）と継続コスト（運用・保守、AIモデル再学習・チューニング）を正確に把握

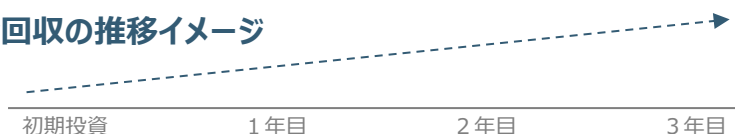
長期的視点での価値評価

短期的コスト削減だけでなく、品質向上、市場投入時間短縮、イノベーション促進などの長期的ビジネス価値を総合評価

ROI計算フレームワーク

初年度ROI = $\frac{(\text{SMTTC} + \text{QIC}) - (\text{AIC} + \text{AOC})}{(\text{AIC} + \text{AOC})} \times 100\%$
SMTTC:削減される手動テストコスト、QIC:品質向上によるコスト削減
AIC:自動化ツール導入・構築コスト、AOC:自動化運用・保守コスト

投資回収の推移イメージ



各業界における開発工程でのAI活用事例

業界を問わず、AIは開発工程での活用が進み始めており、特にテスト自動化では、工数を最大90%削減するなど顕著な成果を上げるなど、様々な業界でその有効性が実証されています

金融業界

課題 厳格なセキュリティ・コンプライアンス要件、複雑な金融商品のテスト

AI活用 リスク分析モデルのテスト自動化、コンプライアンスチェック自動化

事例

- 大手A銀行銀行は住宅ローン審査システムのテストを自動化し、リグレッションテスト工数を70%削減
- 振興FinTech企業B社はAPIテスト自動化で開発速度を40%向上

製造業界

課題 複雑な生産ライン制御システム、サプライチェーン連携、品質管理

AI活用 PCLプログラムのロジック検証、AI画像認識を用いた外観検査システムのテスト

事例

- 大手自動車部品メーカーC社は溶接ロボットの制御プログラムテストを自動化し、不具合流出を90%削減
- 中堅産業機械メーカーD社はIoTプラットフォームのテストにAIを導入し、開発効率30%向上

小売・EC業界

課題 大量のトランザクション処理、パーソナライズ顧客体験、オムニチャネル連携

AI活用 ECサイトのUIテスト自動化、レコメンドエンジンのテスト、在庫管理システム連携テスト

事例

- 大手ECプラットフォームE社はAIを活用したリグレッションテスト自動化により、週次機能アップデートのテスト時間を60%削減
- 専門小売チェーンF社はPOSとECサイト連携テストにAIを導入し、データ不整合を大幅削減

IT・SaaS業界

課題 アジャイル開発とCI/CDによる頻繁なリリース、マイクロサービスの複雑なテスト

AI活用 CI/CDパイプラインへのAIテスト統合、APIテスト自動化、障害原因の迅速な特定

事例

- B2B SaaSプロバイダーG社はAIテスト自動化プラットフォームを導入し、リグレッションテスト時間を80%削減、リリースサイクルを2週間から3日へ短縮
- SierのH社はAIによるテストケースの生成で設計工数を40%削減

医療業界

課題 患者情報の高度な機密性保持、医療機器との連携、厳格な規制遵守

AI活用 電子カルテシステムのテスト自動化、医療画像診断支援AIの検証、医療情報標準規格連携テスト

事例

- 医療情報システム開発I社はAIを活用して電子カルテのバージョンアップ時のテスト工数を60%削減
- 遠隔医療サービス提供J社はAIによる通信品質テストとセキュリティテストを自動化し、サービス信頼性を向上

建設業界

課題 BIMデータの活用と部門間連携、現場とオフィス間の情報共有、複雑なプロジェクト管理

AI活用 BIMモデルの整合性チェック、設計変更に伴う影響範囲の自動特定、施工管理アプリのテスト

事例

- 大手ゼネコンK社はAIを活用した、BIMモデルの自動チェックシステムを導入し、設計ミスによる手戻りを30%削減
- 建設テックスタートアップL社はAIによるテスト自動化で現場管理アプリのリリースサイクルを50%短縮

ユースケース：金融業界

与信審査、不正防止など多くの課題に対して、信頼性向上、開発効率化、収益性向上、コスト削減という具体的な効果をもたらします

金融業界の課題

- 与信審査の精度向上とリスク管理の高度化要求
- 不正取引検知・防止システムの継続的改善
- 厳格な規制遵守と監査対応の負担
- レガシーシステムとの連携テストの複雑さ
- 高度なデータ分析スキルを持つ人材の不足

開発テスト自動化の活用事例

リアルタイム不正取引検知システムのテスト自動化

複数口座情報、取引履歴、脅威インテリジェンスを用いた異常検知アルゴリズムの継続的なテスト自動化により、システム信頼性を向上

パーソナライズド金融サービスの品質保証自動化

顧客属性、取引履歴、ライフイベントを考慮した金融商品レコメンデーションの自動テストにより、提案精度を担保

コンプライアンス対応テストの自動化

国内外の法規制情報や監督当局からの通達に基づく自動回帰テストにより、規制遵守を効率的に検証

期待される効果

信頼性向上

不正取引検知精度の向上とリスク管理体制の強化

開発効率化

テスト工数削減と市場投入までの時間短縮

収益性向上

顧客エンゲージメント向上によるクロスセル機会創出

コスト削減

コンプライアンス遵守コストの削減とオペレーショナルリスクの低減

ユースケース：製造業界

製造プロセス監視、予知保全、サプライチェーン、設計・開発の各段階で品質向上、稼働率最大化、サプライチェーン最適化、開発サイクル短縮の効果がもたらされます

製造業の現状と課題

DXを推進できている製造業企業

22.6%

JUAS「企業IT動向調査報告書2023」より

- 部門ごとのデータ管理によるデータ分断
- 設計パラメータと製品品質の複雑な関係性
- 熟練技術者の経験や勘に頼ったノウハウ継承の難しさ
- 生産設備の故障予兆検知の精度向上要求
- 需要や在庫の変動を高精度に把握する必要性

開発テスト自動化の活用事例

製造プロセス監視システムの自動テスト

センサーデータ（温度、圧力、振動など）を用いた品質異常検知アルゴリズムの継続的な検証と精度向上

予知保全システムの自動検証

設備センサーデータ分析による故障予測モデルの精度検証と自動テストによる継続的改善

サプライチェーン最適化テスト

需要予測、在庫最適化、調達計画のアルゴリズム検証を自動化し、サプライチェーンの強靱性を向上

設計・開発プロセス検証の自動化

CADデータ、シミュレーション結果、試作品テストデータの自動検証によるリードタイム短縮

期待される効果

品質向上と不良率削減

製品品質の安定化、歩留まり改善によるコスト削減と顧客満足度向上

設備稼働率の最大化

突発的な設備停止の防止、メンテナンスコスト最適化による生産性向上

サプライチェーン最適化

在庫圧縮、欠品防止、市場変動への迅速な対応能力強化

開発サイクルの短縮

製品開発期間の短縮、イノベーション促進による市場競争力の強化

ユースケース：EC業界

クロスデバイス、決済連携、ユーザー行動シミュレーション、継続的インテグレーションを通じて、市場投入速度向上、コンバージョン率向上、リスク低減を実現し、ビジネス成長を加速させます

EC業界の課題

- 多様なデバイス・ブラウザでの表示・機能検証の複雑
- 複数の決済システム連携における安全性確保
- 頻繁な機能追加・UI変更に対する迅速なテスト実行
- サイト内顧客行動パターンの多様性
- 集客・マーケティング施策の効果検証

ECサイト平均のカート放棄率

63.3%

自動テストによるUX改善で低減可能

開発テスト自動化の活用事例

クロスデバイステスト自動化

複数のデバイス・ブラウザ環境での表示崩れや機能不全を自動検出。レスポンシブデザインの品質担保とテスト工数削減を実現

決済システム連携テスト

複数決済手段のエンドツーエンドテストを自動化。セキュリティ脆弱性の継続的検証と取引の整合性確認を効率化

ユーザー行動シミュレーション

実際の顧客行動パターンに基づく自動テストシナリオで、カート放棄率の原因となるUX問題を早期発見・改善

継続的インテグレーション

頻繁なサイト更新に対応する自動回帰テストで、新機能追加やUI変更による不具合を即時検出。リリースサイクル短縮化

期待される効果

市場投入速度向上

テスト時間短縮による継続的デリバリーの実現

コンバージョン率向上

UX問題の早期発見・修正によるカート放棄率低減

リスク低減

決済プロセスの堅牢性確保とセキュリティ強化

ユースケース：SIer業界

多様な要件、複雑な連携、短納期・高品質の両立、テスト環境構築の煩雑さなどの課題を解決し開発サイクル短縮、品質安定化、コスト効率化を実現し顧客満足度向上を実現します

業界の課題

- **多様な顧客要件への対応**
業種・規模・技術スタックが異なる顧客ごとに最適化が必要
- **短納期・高品質の両立**
厳しい納期制約下での品質担保と効率的なテスト実施
- **複雑な連携テスト**
複数システム間の連携や既存システムとの統合テストの複雑さ
- **テスト環境構築の煩雑さ**
顧客環境に応じた多様なテスト環境の構築・管理コスト

テスト自動化ユースケース

クロスプラットフォーム回帰テスト

複数OS・ブラウザ・デバイスでの一貫した動作検証を自動化し、マルチプラットフォーム開発の品質を担保

システム連携テスト自動化

API・マイクロサービス間連携のモック生成と自動テストによる複雑な統合環境での検証効率化

テスト環境自動構築

コンテナ技術とInfrastructure as Codeを活用した顧客環境に即したテスト環境の自動構築・復元

要件トレースability検証

顧客要件からテストケース・実行結果までの追跡を自動化し、要件充足の証跡を効率的に管理

期待される効果

開発サイクル短縮

テスト工数削減による納期遵守率向上

品質の安定化

人的ミス削減と一貫した検証品質

コスト効率化

プロジェクト間のテスト資産再利用

顧客満足度向上

高品質・短納期による信頼獲得

導入ステップ



テスト自動化成功のポイント

- プロジェクト間で再利用可能なテストフレームワーク構築
- 顧客環境の差異を吸収できる柔軟なテスト設計
- 開発・運用チームと連携したDevOpsプロセスの確立

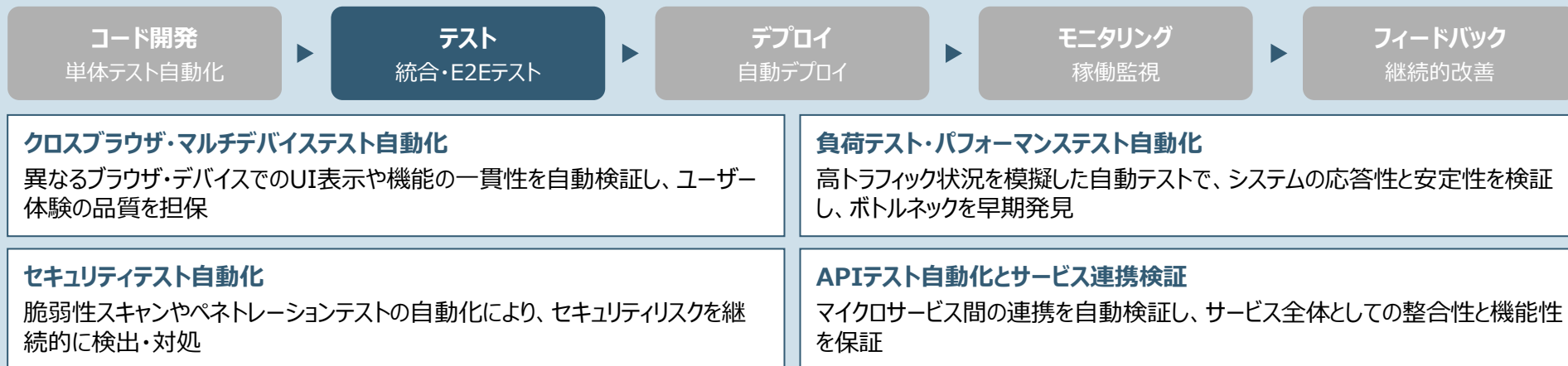
ユースケース：Webサービス業界

短リリースサイクル、多様なデバイス対応、高トラフィックの安定性、セキュリティなどの課題に対しリリース速度向上、不具合早期発見、品質向上、スケーラビリティを実現し、競争優位性を確立します

Webサービス業界の課題

- 短いリリースサイクルと継続的なアップデート要求
- 高トラフィック時のパフォーマンス維持と可用性確保
- マイクロサービス間の複雑な連携テスト
- 多様なデバイス・ブラウザでの一貫した体験提供
- セキュリティ脆弱性の継続的な検出と修正

Webサービス向け自動化戦略



自動化による効果

リリース速度向上

テスト時間の短縮により、市場投入までの時間を大幅に削減

不具合早期発見

開発初期段階での問題検出による修正コスト削減

品質向上

網羅的なテスト実施によるユーザー体験の一貫性確保

スケーラビリティ

テスト工数を増やさずに機能追加・拡張が可能

導入アプローチ

全社的なAI活用戦略とガバナンスの構築から始めることが重要であるが、初期的にはコスト効率と特定領域に限定したスモールスタートでのAI導入の検証も兼ねたトライアルを行うケースが多い

スモールスタートアプローチ

品質向上と不良率削減

- ・ 限られた予算と人材
- ・ AIや自動化に関する専門知識の不足
- ・ 導入・運用への心理的・技術的ハードル

AI活用例

- ・ 比較的安価なクラウドベースのAIテストツールの部分導入
- ・ 特定の繰り返し作業が多い業務へのスモールスタート
- ・ RPAツールとAIを連携させ、既存の自動化範囲を拡張

成功要因

- ・ 解決したい課題の明確化
- ・ 費用対効果の高いソリューションの慎重な選定
- ・ PoCを通じた段階的導入
- ・ 外部専門家やベンダーサポートの活用

事例企業

特定ニッチ市場向けSaaSを提供するO社（従業員50名）は、AI搭載のノーコードテスト自動化ツールを導入し、月20時間かかっていた手動リグレーションテストをほぼゼロに削減。開発者は新機能開発に集中できるようになりました

スモールスタートアプローチ

品質向上と不良率削減

- ・ 大規模かつ複雑な既存システム
- ・ 多数の部門間連携の必要性
- ・ 全社的な開発・テストプロセスの標準化の難しさ
- ・ レガシーシステムからの移行

AI活用例

- ・ AI活用のためのCoE（Center of Excellence）設立と全社戦略推進
- ・ 全社共通のAI開発・テスト自動化プラットフォーム導入
- ・ 既存システムとの連携を考慮した段階的なAI機能導入
- ・ 社内AI人材の育成プログラムと外部研修の積極活用

成功要因

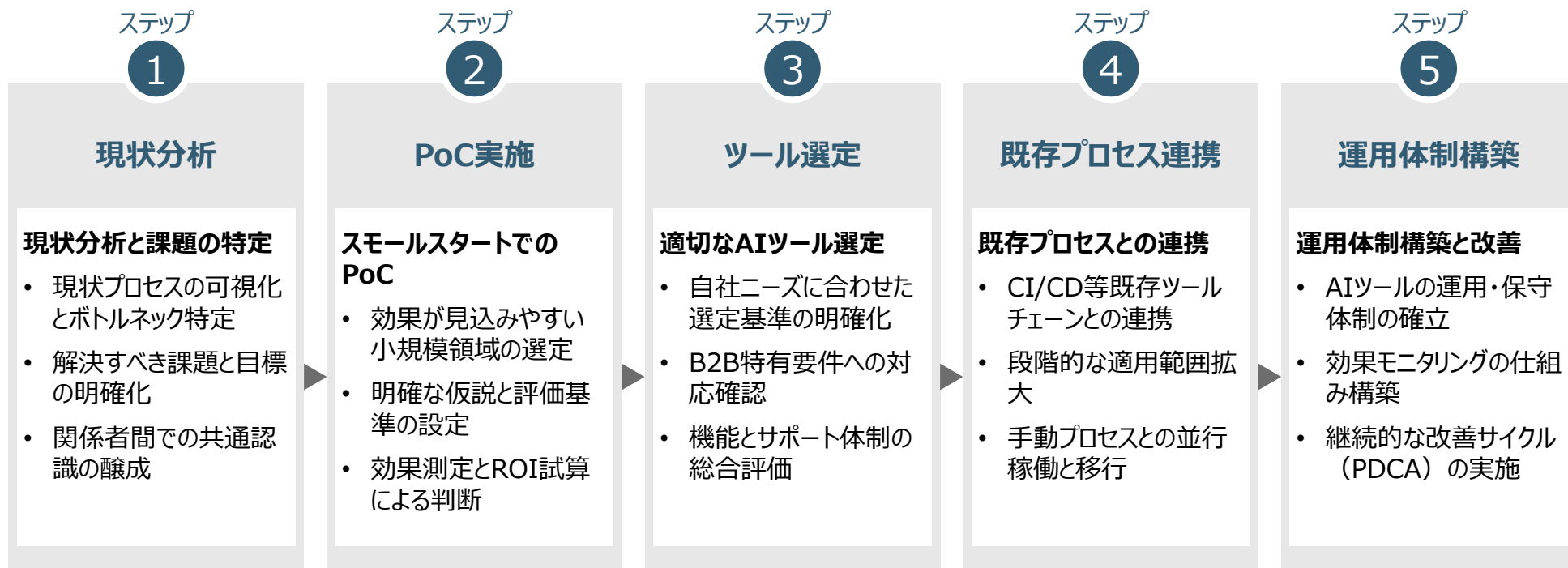
- ・ 経営層の強力なコミットメントとリーダーシップ
- ・ 全社横断的な推進体制の構築
- ・ 丁寧なチェンジマネジメント
- ・ アジャイルなアプローチによる継続的な改善と効果測定

事例企業

グローバルに事業展開する製造業P社は、AIを活用した品質管理システムを導入。設計データと生産データをAIで解析し、潜在的な不具合要因を早期に特定。テスト工程でのAI活用と合わせ、製品リコール率を過去5年で最低レベルに抑制しました

AI導入プロジェクトの進め方

AI導入は一気に全社展開するのではなく、スモールスタートで成功体験を積み重ね、徐々に範囲を拡大していくアプローチが現実的である



AI活用にあたり考慮すべき点

AI活用の成功には、データの質、セキュリティ、倫理の担保が不可欠であり、さらに計画的な人材育成と、経営層の強いコミットメントのもとで全社的にビジネスプロセス変革として推進することが重要です

データの質

AIの精度と効果は学習データの質に大きく依存します。B2B領域では、業界特有の専門知識や複雑なビジネスルールを反映した高品質なデータ整備が必要不可欠です。データの正確性、網羅性、最新性を確保するためのガバナンス体制構築が重要です

データ品質管理プロセスの確立がAI成功の基盤

セキュリティとプライバシー

AIツールへの機密情報や顧客データの入力に関するリスク管理が重要です。特にクラウドベースのAIサービス利用時には、データ保護対策を徹底する必要があります。B2B企業では、取引先情報や企業秘密の保護が信頼の基盤となります

セキュリティポリシーとAI活用の両立

AI倫理

AIの判断や提案の透明性と説明可能性を確保することが重要です。バイアスや差別的な結果を生まないよう配慮し、人間による適切な監督と最終判断を担保する体制を整えましょう。特にB2Bシステムでは、判断の根拠を明確に説明できることが重要です。

説明可能なAIの考え方を取り入れる

人材育成

AIツールを効果的に活用できるスキル開発が必要です。エンジニアのAI技術への理解と活用能力を向上させる教育プログラムを整備しましょう。日本企業では特に人材不足が深刻な問題となっており、計画的な育成と外部知見の活用が重要です

AIリテラシー向上を全社的な取り組みとして推進しましょう

経営層コミットメント

AI導入は全社的な取り組みとして位置づける必要があります。経営層の強いリーダーシップのもと、長期的視点での投資判断と継続的な改善サイクルを支援することが成功の鍵となります。AIは単なるツール導入ではなく、ビジネスプロセス全体の変革として捉えるべきです

トップダウンとボトムアップの両面からの推進

失敗を招くケースと対策

過度な期待、準備不足、現場の抵抗、短期的なROI重視によりAIの導入を失敗したととらえられるケースが多いが、スモールスタートでの検証から始め長期的な視点で取り組むことが不可欠です

過度な期待

AIを「万能薬」と捉え、現実的な検証や段階的な導入なしに、あらゆる問題が解決されると期待してしまう。その結果、実際の効果とのギャップに失望し、プロジェクト全体が頓挫するリスクがある

対策

明確かつ測定可能な目標設定を行い、スモールスタートでのPoC（概念実証）を通じて効果を検証する。AIの得意・不得意を理解し、適材適所での活用を目指す。経営層を含めた関係者間で現実的な期待値を共有する

不十分な準備

現状プロセスの可視化や課題の特定が不十分なまま導入を進め、データの質や整合性に問題があることに後から気づく。また、AI導入に必要なインフラや人材育成の準備が整っていないケースも多い

対策

現状のプロセスを詳細にマッピングし、ボトルネックや非効率を客観的に特定する。データの質と量を事前に評価し、必要に応じてデータクレンジングを実施。AI導入に必要なスキルセットを明確にし、人材育成計画を立案する

変化への抵抗

AIによる自動化に対する不安や抵抗感から、現場の協力が得られず、導入が形骸化する。特に「AIに仕事を奪われる」という恐れや、長年の経験に基づく判断がAIに置き換わることへの抵抗が障壁となる

対策

AIは「置き換え」ではなく「強化・支援」のツールであることを丁寧に説明し、現場を巻き込んだ形での導入を進める。AIによって単調作業から解放され、より創造的・戦略的な業務に集中できるメリットを具体的に示す。成功体験を共有し、段階的に理解と支持を広げる

短期的ROI重視

初期投資コストに対する短期的なリターンのみを重視し、AI導入の長期的な価値や定性的な効果を評価しない。コスト削減だけに注目し、品質向上や生産性向上といった総合的な価値を見落としてしまう

対策

定量的効果（開発リードタイム短縮、不具合削減率など）と定性的効果（品質向上、エンジニア満足度向上など）を総合的に評価するフレームワークを導入する。複数年にわたるキャッシュフローをシミュレーションし、長期的な投資対効果を経営層に示す

事業開発ならenableX

革新的なAIテクノロジーと高度な事業開発の専門性で次世代の事業開発サービスを提供します

経営タレントリソース

事業開発機能

AIテクノロジー

プロジェクト実績

平均継続率94%。中長期的に事業開発・テクノロジー活用の支援を行っています



数字で見るenableX

自社でもグローバルに事業立ち上げ・投資を行う、事業開発専門ファームです

事業開始2年で20億円以上の売上

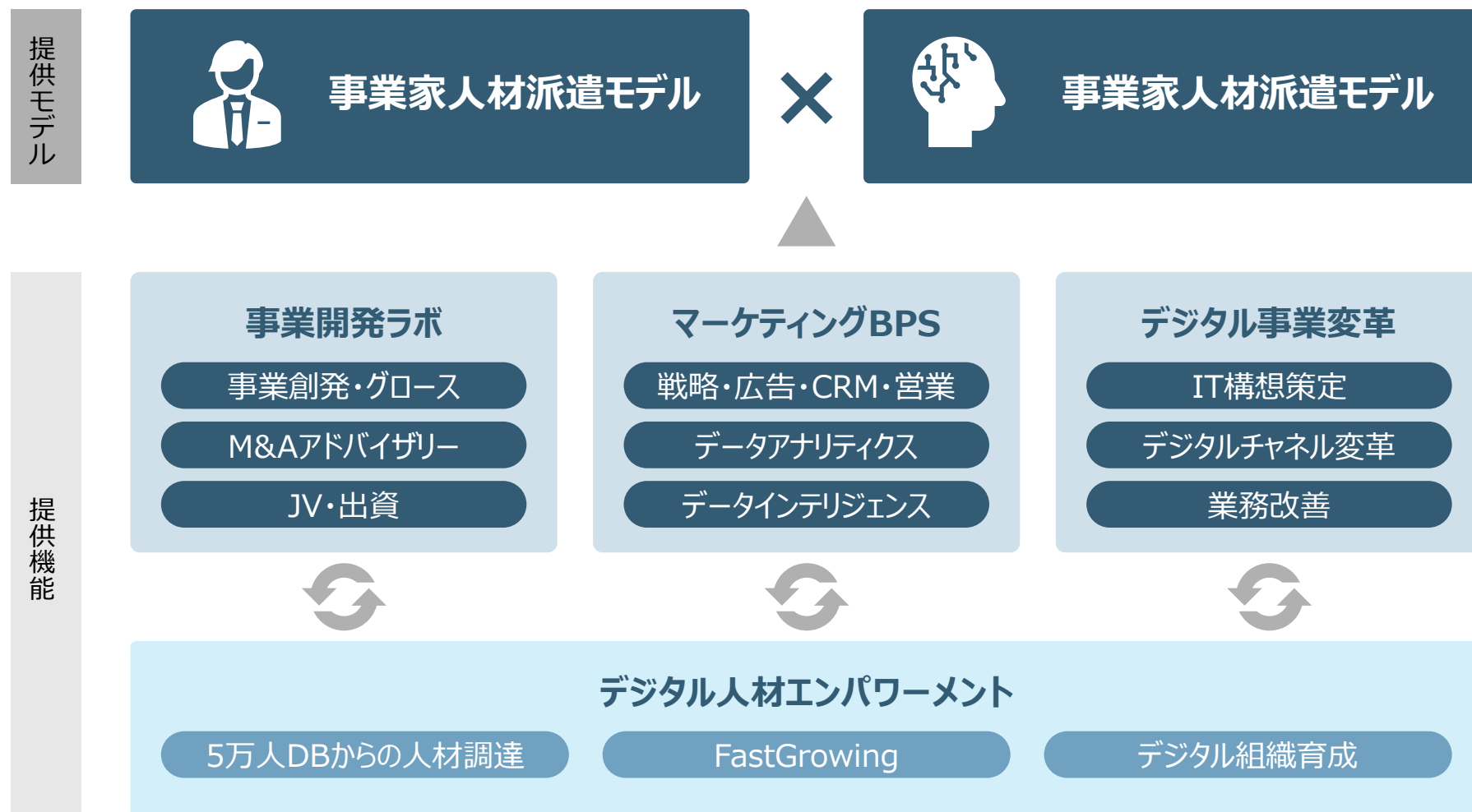
グローバルに100以上のプロジェクトを実施



保有事業会社5社

enableXの事業開発プラットフォームサービス

enableXが有する事業開発・グロースのための機能を、「事業家人材派遣 x eX AI agent」モデルで提供をすることで、結果創出に向けた効率性と確実性を担保します



お問い合わせ先

個別でのご相談などお気軽にお問い合わせください



担当部署

コンサルティング事業部



メールアドレス

contact@enablex-inc.com



ウェブサイト

<https://enablex-inc.com/>

無料個別相談会のご案内

貴社が抱える具体的な消費者理解の課題、導入目的、期待効果などについて、無料でご相談を承っております。

実際のAIペルソナと対話体験も可能です。

まずはお気軽にお問い合わせください。

資料請求・PoC提案

詳細な技術情報や、貴社の特定の課題・業種に特化した活用事例集をご用意しています。

小規模PoCを通じた具体的な価値検証も承ります



本資料は、お客様のために作成された文書であり、当社の書面による
事前承諾なしに、第三者に対し開示を行うことはお断りしております