1.モデル及び周辺技術の発展

基盤モデル

- 大規模言語モデル
- 小規模言語モデル
- 画像動画生成モデル
- 音声生成/会話モデル
- 時系列基盤モデル
- マルチモーダル
- 高度推論

モデル I/O

- コンテキストサイズ拡張
- 検索拡張生成（RAG）
  - フルテキスト検索
  - ベクトル検索
  - セマンティック検索
  - グラフDB

フレームワーク

- チャットボット
- プロンプトチェーン
- AIエージェント
  - Dify/n8n/Copilot Studio
  - LangChain
  - LangGraph
  - LlamaIndex
  - etc

- マルチモーダルによる認知力の向上
- RAGにより生成AIの出力を制御可能
- フレームワークの充実
- コンテキストエンジニアリングの重要性

モデル及び周辺技術の発展により、AIエージェントの社会実装が昨年より可能になったことから、実装事例やプロダクトが多く登場中。

	Step1.初期導入（体験）	Step2.業務効率化	Step3.業務自動化
一般	社内AIチャットシステム導入	Microsoft 365 Copilot + α リサーチ/プレゼン/AIヘルプデスク、、	AX推進（AIエージェント開発＋DX） **業務自動化** **業務コンサル+エージェント開発**
エンジニア	プロンプトエンジニアリング 習熟度に個人差	ポイント・ソリューション（用途特化）の活用（業務活用推奨ツールの拡充）	AIエージェント導入＋プロセス変革 **生成AI駆動開発**
	調査研究・教育	GitHub Copilot + α GUIデザイン/仕様・設計QA/ホワイトボックスファズテスト/E2Eテスト、、	

© 2026 FUJISOFT INCORPORATED. All rights reserved.

3．業務変革

FUJISOFT INCORPORATED

業務分析
- 業務ヒアリング
- 業務工数/ログ分析

業務選定
- 業務難易度（中/低）より自動化効果の高いもの選定

自動化企画
- 業務フロー改善
- 業務データ標準化
- 業務フロー自動化企画（KPI設定）

自動化開発
- AIエージェント
- ワークフローオートメーション
- MCP
- RPA

評価&検証
- 自動化精度評価
- KPI評価
- 活用サポート

・主業務やサポート業務（問合せ、データ入力、データチェック、etc）の様に業務を分類

・業務担当者のレベルを分類

・業務工数を集計

・複数の部門で共通する業務フロー

・業務工数が多いもの

・サポート業務の様に付加価値が低い業務

・業務フローや、データを自動化出来るよう標準化

・自動化率を高めるための業務フロー改良

・自動化による工数削減効果を見積

【活動状況】
・全社の業務効率化の号令の下、コーポレート部門に伴走して業務コンサル＆自動化開発推進中
・全体工数の4割を占める【サポート業務】は、自動化＆改善のメインターゲット
・コーポレート部門要員は、ITに詳しくなく、又、真面目に【サポート業務】を行っており、自ら業務の非効率に気が付きにくく。業務コンサルのヒアリング無しでは明らかにしずらい

【活用状況（成果）】
・申請/受付業務の自動化
・社内QA対応業務の自動化
・データの転記/入力作業
・脆弱性情報の収集業務の自動化
・クラウドサービスのアセスメント業務の自動化
・etc

Dify/n8n/Copilot Studioなどを活用し、
2028年までに、コーポレート部門の業務を10万時間/年分、自動化目標で活動中。

活動初年度、2ヶ月で6千時間/年を削減。
（本年度中に1万2千時間削減目標）

© 2025 FUJISOFT INCORPORATED. All rights reserved.

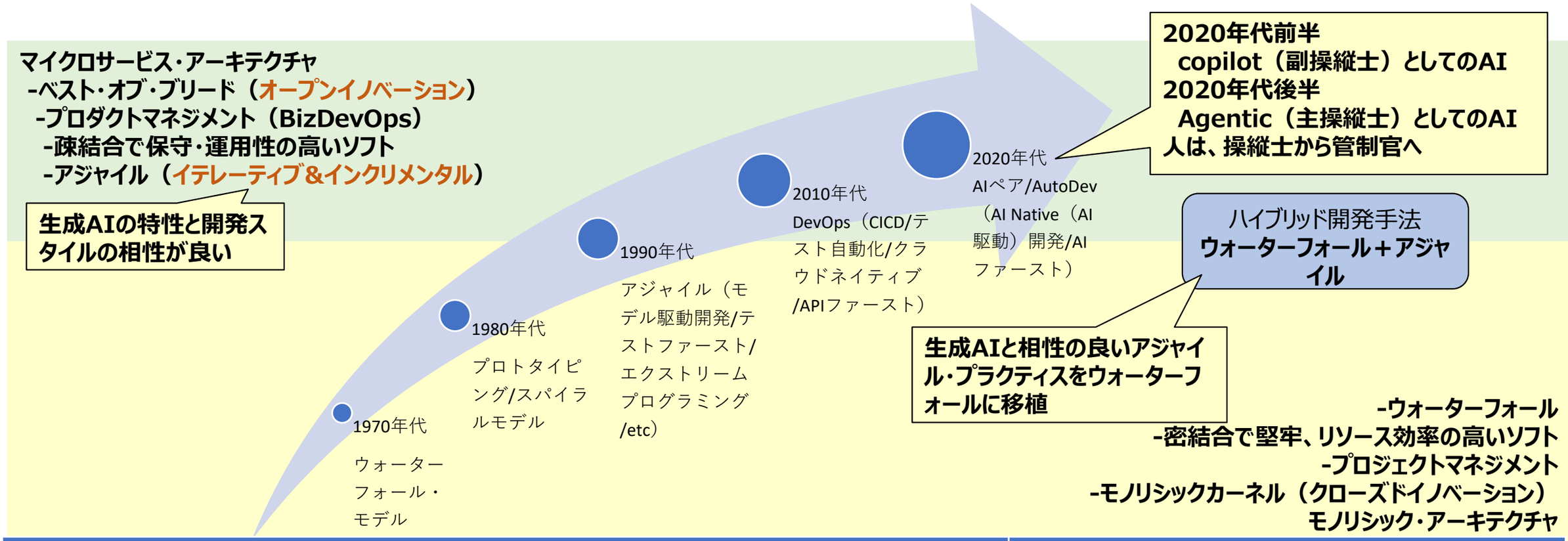
生成AIの事業活用を阻害する要因

業界共通課題	No.	カテゴリ	富士ソフト課題	AI導入阻害要因
①信頼性・セキュリティの懸念	1	権利・責任・契約・規則	社内ルールの不明確	**厳格な情報管理ルールにより、AI導入への理解やイメージが持たれにくい** 社内ルールでプロジェクト情報の翻訳すら禁止されている企業では、AIの活用に対する理解や許容度が低く、導入のイメージを持ちにくい。
	2		所有権/著作権違反	**著作権リスクと確認コストの影響で、従来の開発方法が優先される傾向にある** お客様との折衝の結果、AI生成物の著作権侵害を確認するコストが高いため、従来の開発スタイルの方が望ましいと判断された。（ディズニー訴訟やGitHub CopilotのOSS流用問題などが背景）
②契約形態・知財取り扱い問題	3			**データの所有権やAIのミス時の責任が不明確で、導入の障壁となっている** 「データの所有者」「AIの判断ミスの責任の所在」といった権利や責任の所在が曖昧で、導入リスクとなっている。
	4		契約・リスク回避	**保守契約は安定稼働が最優先でAI導入失敗時の責任が曖昧な新技術導入に消極的** 安定稼働を重視する既存の保守契約では、新技術導入に対し抵抗が強い。AI導入で期待通りの効果がでなかった場合の責任が不明確で導入の障壁となっている。
③ビジネスモデルのジレンマ	5	収益構造	ビジネス構造	**AIによる効率化（＝工数削減）が売上減につながる構造** AIによる効率化により、工数が減ると売上も減るという、ビジネスモデル（人月単価 × 人数 × 期間」で収益を上げるモデル）上のジレンマがある。
④プロンプトの難しさと精度の課題 （知識・スキル・マインドセット）	6	マインドセット・スキル	AI人財不足	**AI開発や運用の経験が乏しく、人材・体制の面で使いこなすのが難しい** AI人材が不足しており、多くは要件定義・設計・保守が中心で、AIモデル開発やMLOpsの経験がなく、体制もない。
	7			**AIへの理解不足から過度な期待や誤解が生まれ、PoCの設計が適切に行われない** 現場では「AIは魔法の箱」と誤解されることも多く、正しくPoCを設計できないケースがある。
	8			**多様なサービスの中で選定が難しく、PoCの進行が最新動向に追いつかない** サービス数が多く選択が難しい。PoC開始後により良いサービスやアップデートが登場し、現場適用が追いつかない。
	9			**教育不足で基礎知識が乏しく、会社の支援がなければ安全にAIを使えない** 現場の教育不足により、リテラシーやセキュリティを意識した基礎知識が不足している。会社が教育を支援しないと、安全にAIを使わせることができない。
	10		エンジニアの抵抗感	**業務がAIに置き換わることへの不安から、導入に対する抵抗感が生じる** 現場では「AIに仕事を奪われる」という不安から、導入に対して消極的・非協力的な反応が見られることがある。
-	11	組織・文化的な未整備・顧客起因	開発スタイルのアンマッチ	**ウォーターフォール文化が強く、AIに必要な柔軟な進め方が浸透しにくい** SI現場ではウォーターフォール型の「最初に要件を固める」文化が根強く、AIに必要な「試行錯誤しながら精度を高める」アジャイル的な進め方が受け入れられにくい。
	12		リスクを伴う技術導入への慎重な姿勢	**失敗を許容度が低いSIerの文化があり、AIに必要な試行錯誤が進めづらい** AIは試行錯誤を前提とするが、SIerの現場は失敗の許容度が低く、チャレンジが評価されにくい。
	13		費用対効果	**AI導入の費用対効果（コスト削減、売上増）が不透明で、効果を数字で示しにくい**
	14		データ環境・整備不足	**顧客データのサイロ化、AI学習用データの収集やテストデータの作成に手間がかかる** 業務毎に顧客データが分断され、AIの学習用データを収集しづらい。テストデータも手作業でシステムに入力している

【施策】
クローズドな生成AI活用環境構築

© 2020 FUJISOFT INCORPORATED. All rights reserved.

4.開発変革（開発プロセス変革）



アジャイル（エクストリームプログラミング）の開発プラクティス		生成AI活用
テスト駆動開発	テストファースト、イテレーティブ/インクリメンタル開発、テスト自動化（自己検証/繰り返し可能）	スペック駆動、テスト駆動、Issue駆動など、タスク委任型/自立・協調型のエージェンティックな開発への移行
YAGNI原則	今必要（仕様に関係）無いコード（処理）は実装しない	
リファクタリング	テスト自動化の元、安全に内部構造の見直し	補完型、対話型での開発への移行
ペアプログラミング	ナビゲーターとドライバーとで二人三脚	

© 2020 FUJISOFT INCORPORATED. All rights reserved.

4.開発変革（検証）

①業務ドメインに特化したコードに対してKrugleがどこまで対応出来るかを検証

観点	チェック内容
業務用語の理解	「口座種別」や「保険料率」など用語の意味をどこまで理解出来ているか？
業務ロジックの再現性	計算式や条件分岐から業務の流れを正確に理解できるか？
業務特有の処理	月次処理、締め処理など理解出来ているか？
非機能要件	ログ、エラー処理、セキュリティなど
保守性	可読性、命名規則、コメントなど

②開発業務においてKrugleが有用なユースケースの検証

工程	ユースケース
要件定義・仕様策定	要望一覧・議事録等から、仕様書ドラフト生成/仕様書から差分抽出
アーキテクチャ設計	設計パターンのプロトタイプ生成/ブロック図やシーケンス図の生成
詳細設計	関数仕様書の生成/状態遷移図の生成
実装	プロトタイプ生成/テンプレート生成/リファクタリング
テスト	テストケースの生成/モックコードの生成/テストコードの生成/テストログの解析
デバック・検証	システムログの解析/バグの原因推定
ドキュメント作成・保守	コードからドキュメント生成

© 2020 FUJISOFT INCORPORATED. All rights reserved.