

業務フロー設計シリーズ | 第1部

システム導入で失敗する前に 読んでほしい、業務フロー設計の進め方

～ AI時代の大規模システム刷新が失敗する5つの理由 ～

aurant technologies

著者紹介

PORTFOLIO

PORTFOLIO



Koki Soga

Aurant technologies株式会社 アーキテクト・コンサルタント

経歴・プロフィール

開発会社PM、コンサルティング会社を経て現職。一貫して企業のデータマーケティング基盤の構築やDX戦略の推進をリード。現在は大企業から中小企業まで、マーケティングとバックオフィスを統合する全体最適なシステムアーキテクチャの設計・実装を牽引

主なプロジェクト実績

アパレルブランド：

- CRM・MA（マーケティングオートメーション）基盤の構築

建築業：

- MA(マーケティングオートメーション)基盤の構築、全社DXの推進

その他

- クラウド会計システム導入および業務プロセス改善
- レガシーシステム（既存の独自DBやオンプレミス）からの脱却
- モダンデータスタック（CDP/DWH）への移行支援
- 複数SaaSの統合、生成AIを活用した業務効率化プロジェクト

02

何故いま

このシリーズを作ったのか？

なぜ今、このシリーズを作ったのか

日本企業のIT導入成功率は52.8%。約半数が失敗または期待効果未達のまま終わっている現状があります。こうした、失敗例を踏まえよりスムーズに皆さまの開発が行える一助となれば幸いです。

52.8%

IT導入プロジェクトの
失敗・期待未達率

出典: 経済産業省「DX推進指標」
およびIPAデータ白書をもとに試算

失敗1件あたり
5,000～7,000万円規模

① 開発費用のオーバーラン

当初見積の2～3倍のコストが発生するケースが多い。
追加要件・仕様変更・手戻りが連鎖して膨らむ。

② スケジュール遅延

6ヶ月予定が1～2年に延びることも珍しくない。
リリース遅延中も人件費・ライセンス費は発生し続ける。

③ 期待効果の未達

完成しても「現場が使わない」「工数が減らない」で終わる。
業務改善どころか業務が増える逆効果になることも。

03

「AI時代」に何が変わったのか —
なぜ今このテーマが重要か

「AI時代」に何が変わったのか — なぜ今このテーマが重要か

AI・生成AIの普及でシステム開発の景色が変わった。だからこそ「正しい進め方」を知らない人が増え、失敗も増えている

2023～2026年、AI開発ツールが変えた3つのこと

コードをAIが書く時代に

ChatGPT・Cursor・Claude Codeの普及。自然言語で指示するだけでコードが生成される「バンプコーディング」が2025年に爆発的に普及。

Cursor ARR: 1年で12倍成長

「自分で作れる」という幻想が広まる

「エンジニアなしでアプリが作れる」という情報が溢れた。中小企業のDX担当者が「外注より自社開発」を選ぶケースが急増。

外注費75～99%削減の報告も

それでもシステム失敗率は下がっていない

AI時代になっても、業務設計・要件定義・データ設計のミスによる失敗は変わらず続いている。むしろ「AI自社開発」という新しい失敗パターンが加わった。

新しい失敗パターン発生

「AIで作れるなら、なぜ頼む必要があるの？」への答え

AIは「AIはコードを書く」

✗「業務設計はできない」

「受注したら請求書を出す」はAIが書ける。でも「分割請求・前受金・代理店経由」の業務ルールを正確に実装するには、業務を理解した設計者が必要

AIは「AIは動くものを作る」

✗「業務で使えるものは作れない」

「動くこと」と「業務要件を正確に満たすこと」は別問題3ヶ月後に崩壊する技術的負債を生むのがAI自社開発の典型的なパターン

AIは「AIはコードを生成する」

✗「データ設計・連携設計は人間の仕事」

freeeとSalesforceをつなぐには、両システムのデータ構造・業務ルール・連携タイミングを全て理解した設計が必要。AIへの指示では代替できない

AIは「AIはバグを修正する」

✗「自分で修正できるのは自分が理解しているコードだけ」

AIが生成したコードを誰も読めない状態になると、バグが出たときに詰む。「誰も保守できないシステム」が生まれる

「AIで自分たちで作れる」——バイブコーディングが業務システムで失敗する理由

Cursor・Claude Codeでコードは書ける。でも「最初の 1~2ヶ月は動く・3ヶ月目に崩壊する」が典型的なパターン

1~2週目	爆速で動く ✓ 「1日でUIができた！」生産性の高さを実感
2~4週目	少しずつ歪む ● 新機能を足すと既存が壊れる。AIが「修正完了」と言っても直っていない
2~3ヶ月	設計が崩壊 ● コードがAIのコンテキストに収まらない。仕様書がない。誰も全体像を把握できない
本番稼働後	保守不能 ☠ バグが出ても直せない。経理業務が止まる。最悪は最初から作り直し

なぜ失敗するか — 根本の3理由

① 業務ルールはAIに教えられない

「前受金の振替処理」「インボイス税区分の設定」「代理店経由の請求先分離」
——これらは自社の業務を理解した設計者がいなければ正確に実装できない

② セキュリティ穴が静かに生まれる

AIはエラーを直すために認証チェックを削除することがある。APIキーや顧客データを扱うシステムでこれが起きると情報漏洩に直結する

③ 「誰も読めないコード」は保守できない

GitClear調査: AI導入後リファクタリング 60%減・コピー48%増。スケールすると誰も触れない技術的負債の山になる

AIは「コードを書く」仕事を助けてくれる。でも「業務を設計する」「システム全体を正しく動かす」という仕事は、依然として業務を理解した専門家が担う。これが今の現実。

「知らないから怖くない」が一番怖い — ノンエンジニア開発のセキュリティリスク

AIでコードが書いても、セキュリティの知識がなければ「穴だらけのシステム」を本番稼働させることになる

エンジニアがセキュリティを意識するのは「訓練されているから」。
ノンエンジニアが怖いのは「何が危険か知らない」こと。知らないリスクは対処できない。

🔑 ① APIキーの漏洩

情報漏洩

AIに「freeとつないで」と指示すると、APIキーをコードに直接書き込む。そのコードをGitHubにアップロードした瞬間、世界中に公開される。

⚠️ 実際にGitHubで「free APIキー」が検索され悪用された事例が複数報告されている

💉 ② SQLインジェクション・XSS

不正アクセス

入力フォームに悪意のあるコードを仕込むと、DBの中身が全部抜かれる。AIは「動くコード」を生成するが、入力値の検証(バリデーション)を省略することが多い。

⚠️ 入力フォームがあるWebページを公開すると標的になる。規模に関係なく自動スキャンで狙われる

🤖 ③ AIへの機密データ貼り付け

情報漏洩

「このエラーを直して」とAIに相談するとき、社内の顧客データ・売上数字・APIキーをチャットに貼り付ける。その情報は外部サーバーに送信・記録される。

⚠️ 会社が許可していないAIサービスへの業務データ送信は、個人情報保護法・社内規定違反になる可能性がある

👻 ④ 幽霊パッケージ攻撃

サプライチェーン攻撃

AIが存在しないライブラリ名を「ハルシネーション」で出力する。攻撃者がその名前でもルウェアを公開している。知らずにインストールすると社内システムが感染する。

⚠️ 2026年に入ってからこのパターンの報告が急増。AIコーディングに特有の新型攻撃手法

🔒 ⑤ 認証・権限の削除

不正アクセス

「ログインでエラーが出る」とAIに相談すると、AIがエラーを解消するためにログイン機能ごと削除することがある。「誰でもアクセスできる状態」で公開される。

⚠️ free・Salesforceと連携したシステムで認証が外れると、全顧客データ・取引データが無防備になる

📄 ⑥ ログ・監査証跡がない

コンプライアンス

「誰がいつ何を操作したか」の記録がない。情報漏洩が起きても何ヶ月後まで気づかない。会計システムと連携している場合、監査で証跡を求められても提出できない。

⚠️ 金融・医療・上場企業では操作ログの保存が法的に義務付けられているケースがある

SaaS乱立・システムのパズル化が止まらない

フロント・ミドル・バックがバラバラに導入された SaaSでつながっておらず、データ断絶と二重入力が常態化している



⚠ 各領域がバラバラに導入された SaaSは「フロント～バックを横断する視点」がないと絶対につながらない。つなぎ目の設計こそが最重要課題。

フロント～バックオフィス全体像(業務フロー俯瞰)

受注1件が最終的に「入金・決算」に至るまで、7つのステージを3つの組織レイヤーが跨いでいる

	①見込み開拓 リード獲得	②商談・提案 営業活動	③受注 契約締結	④調達・在庫 購買・仕入	⑤出荷・納品 フルフィル	⑥請求 インボイス	⑦入金・決算 ファイナンス
フロント オフィス 顧客接点・ 営業	広告・SEO SNS発信 MA連携	SFA商談 管理・提案 デモ実施	見積書作成 契約書受取 注文書処理 ⚠ Excel手作業	—	納品立会 顧客確認 連絡対応	—	—
ミドル オフィス 受注・在庫・ 物流	—	—	受注確定 在庫引当 発注指示 ⚠ 二重入力	仕入先発注 入荷検収 在庫更新 ⚠ Excelで管理	ピッキング 梱包・出荷 配送手配	納品書発行 送付・確認 連絡対応	—
バック オフィス 経理・財務	—	—	—	買掛金計上 仕入管理 支払処理	—	請求書発行 送付・売掛 計上 ⚠ Excel手打ち	入金消込 売掛回収 月次決算 ⚠ 手動照合

近年のシステム開発会社に求められる要件

フロント～バックオフィスツールまで、様々なシステムを経験し、事業に対する深い理解が求められる

	①見込み 開拓 リード獲得	②商談・ 提案 営業活動	③受注 契約締結	④調達・ 在庫 購買・仕入	⑤出荷・ 納品 フルフィル	⑥請求 インボイス	⑦入金・ 決算 ファイナンス
フロント オフィス			見積書作成				
顧客 サービス							
受注 管理							
バック オフィス				買掛金計上 仕入管理 支払処理		送付・売掛 計上	売掛回収 月次決算
経理・財務						⚠ Excel手打ち	⚠ 手動照合

様々なSaaSが企業にインストールされており、
個社毎の要件が複雑化している現状。

↓
各SaaSツールの仕組みを熟知しており、
マーケティング～バックオフィス・営業まで様々な範囲に対応できる
開発会社が必要とされている

04

システム開発の失敗要因

失敗① 要件定義がないまま進める

AI自社開発でも外注でも共通:「何を作るか」が曖昧なまま進むと、どんな方法を選んでも同じように失敗する

✖ 要件定義なし:よくある悪いパターン

01 「受注管理を効率化して」とだけ指示

「こんな感じで」「任せます」で発注。後で「聞いてない」が頻発する
AI自社開発:動くものは出来るが業務に合わない。外注:ベンダーが憶測で見積もる

02 複数社に「とりあえず見積を」と依頼

同条件でないから比較不能。最安値を選んで大失敗するパターンへ

03 開発中に「あ、これも必要だった」連発

追加要件・仕様変更で見積が150%超に。「追加費用」の連絡が来る

04 完成後に「思ったのと違う」

発注者とエンジニアの認識ズレ。手戻り・再開発で数百万追加費用発生

VS

✔ RFPあり:成功するフロー

As-Is業務フロー図を作成(第2部で詳述)

01 現状業務を図式化。「どこに問題があるか」を全員が共有できる状態にする

課題を定量化してRFPに書く(第3部で詳述)

02 「月80時間・転記ミス月12件」のように数値で書く。ベンダーが的確な提案をできる

複数ベンダーに同条件で提案依頼

03 同じRFPを使うから比較できる。金額・機能・アプローチを対等に評価できる

モックアップで認識を事前に合わせる

04 開発前に画面イメージを確認。「思ったのと違う」を事前に潰せる(失敗④で詳述)

失敗②「御用聞きベンダー」を選んでしまう

「要望を聞いて作るだけ」の業者と「事業を理解して提案する」パートナーは全く別物。見分け方を知っていれば避けられる

評価軸	✖ 御用聞きベンダー(避けるべき)	✔ 本物のパートナー(選ぶべき)
RFPへの反応	「わかりました、やります」と即答 → 理解度を確認しない。後で「聞いてない」が出る	「この要件のここが曖昧です。御社の業務では〇〇と解釈しましたが合ってますか？」 → 提案前に疑問点を自ら整理して持ってくる
提案書の中身	機能一覧と金額表だけ。「〇〇機能搭載」の羅列で終わり。なぜその機能なのかの説明がない	現状課題の理解→解決アプローチ→期待効果の順で説明。 「この順番で実装するとリスクが最小化される」と根拠を示せる
フロント～バック視点	担当領域(例:kintone構築)しか見ない。 「経理連携は別システムで」と言うだけ	「受注したらfreeeにどう連携するか」まで提案に含まれる。 システム全体の整合性を考えられる
課題への姿勢	「追加の仕様はご要望を頂ければ対応します」 → お客様任せ。問題を見つけて提案しない	「今の進め方だと〇〇のリスクがあります。こう変えた方がいいと思います」 → 言わなくていいことを言ってくれる
実績・事例	「実績多数あり」だけで具体例なし。 業種・規模が全く異なる事例しか出せない	「同業種・同規模での実装事例はこちらです。この部分が特に御社のケースと類似しています」 → 具体的・類似事例で説明できる

失敗③「特定ツール専門」ベンダーに依存する罠

「Salesforce専門」「kintone認定パートナー」はそのツールの中だけ最適化する。業務全体は見えていない

✗ 特定ツール専門ベンダーの「視野の狭さ」



kintoneに詳しくても「受注データが freeeにどう流れるか」「SFAからの案件情報をどう受け取るか」が設計できない。フロント～バックの全体像がないから、つなぎ目でシステムが失敗する。

✓ フロント～バックを設計できるパートナー

選定確認事項①

「受注から請求・入金まで一気通貫で設計した事例はありますか？」と聞く。答えられないベンダーはシステム全体を見ていない。

選定確認事項②

提案書に「他システムとの連携設計」が含まれているか確認。API設計・データフロー図がない提案書は危険信号。

選定確認事項③

フロント(SFA・MA)もバック(会計/給与)も、ひとつの会話の中でスムーズに話せるか確認する。

aurant technologiesの場合

フロント(MA・SFA)からバックオフィス(freee・kintone)まで一気通貫での設計・実装が専門領域。「つなぎ目の設計」が私たちの最も得意な領域。

失敗④ モックアップ(画面設計)なしで発注する

「家を建てるのに設計図なし」で工務店に丸投げするようなもの。完成後に後悔しても変更費用は莫大になる

🏠 住宅建築での例え

❌ モックアップなし(よくある発注)

「広いリビングと3LDK希望」とだけ伝えて設計図なしで工務店に施工依頼。完成後に「キッチンが狭い」「リビングが思ったより暗い」と後悔。でも変更費用は莫大。



✅ モックアップあり(正しい発注)

間取り図・外観イメージ・仕様書を事前にレビューしてから着工。「階段の位置を変えてほしい」という変更も図面段階なら低コストで対応できる。

💻 システム開発での例え

❌ モックアップなし(よくある発注)

「受注管理を効率化したい」とだけ伝えて画面設計なしで開発会社に発注。完成後に「この入力フォームが使いにくい」「検索機能がない」と発覚。でも開発後の変更は追加費用が発生。



✅ モックアップあり(正しい発注)

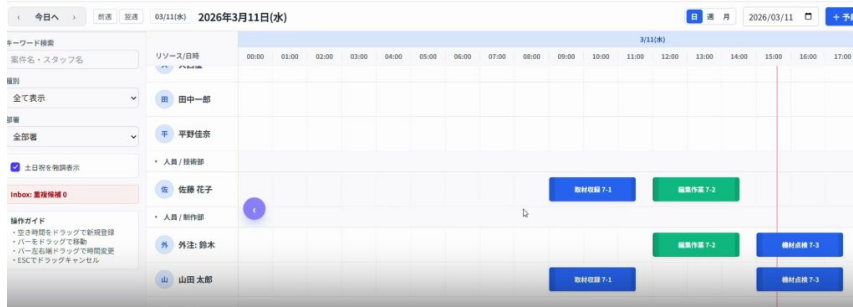
画面モックアップ・操作フローを事前に現場とレビューしてから開発開始。「このボタンはここに移動して」という意見を設計段階で反映できる。コスト追加ゼロ。

実践ポイント: FigmaやAdobe XDでなくてOK。PowerPointやGoogleスライドで作った簡単な画面イメージでも十分。「画面に何が表示されるか・どのボタンを押すとどう動くか」を絵にするだけで、認識齟齬の80%は防げる。

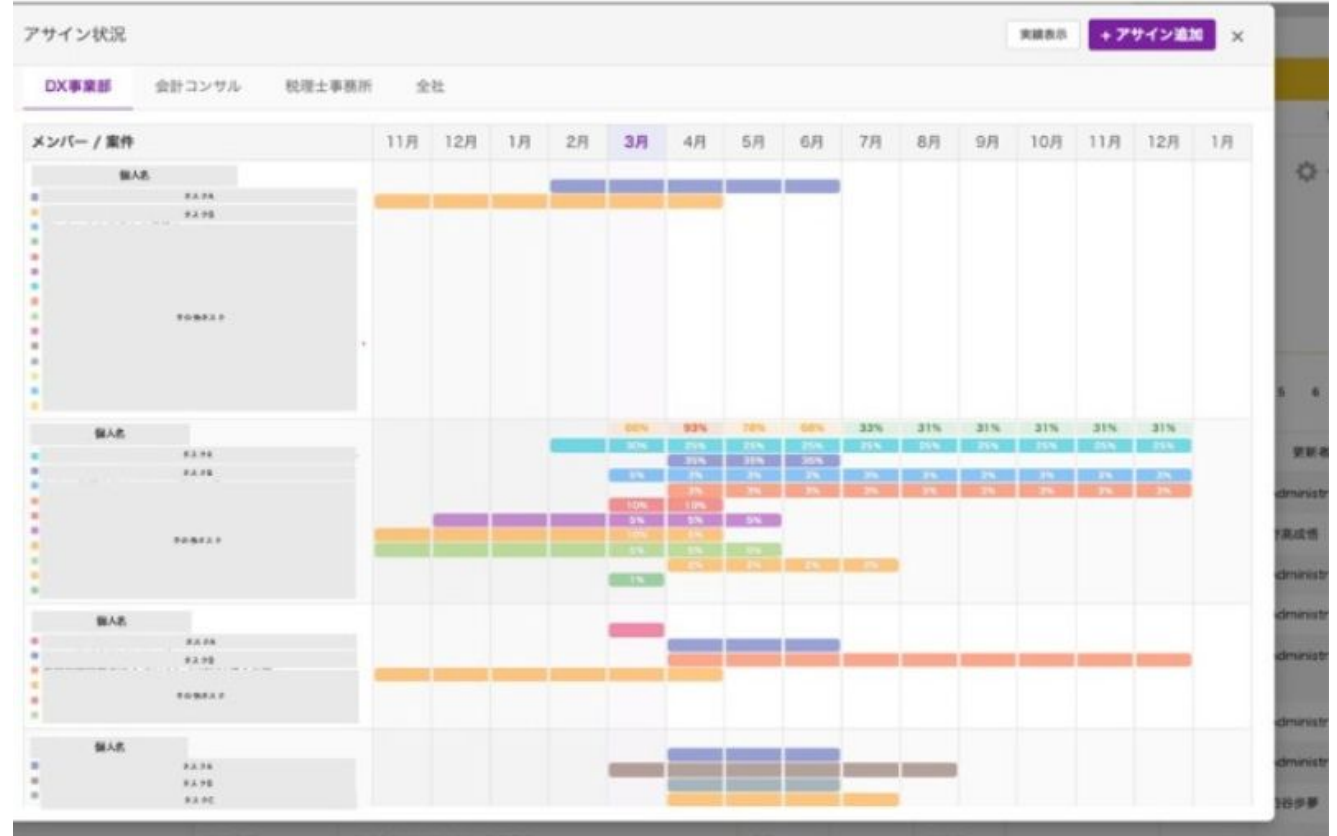
補足) モックアップは生成AIの発展によってより作りやすく

生成AIの発展により、様々なシステムが作りやすくなっています(モックアップや機能の作成といった単純な工程であればより早く)そのため、今後の手戻りを防ぐためにもモックアップの作成はお願いしてみると良いと思います。

打刻・スケジュール・施設予約



プロジェクト・アサイン管理



会計レポート(BI)



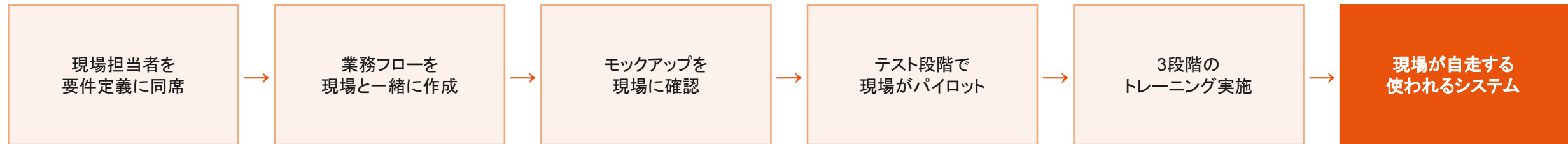
失敗⑤「現場を置き去り」にしたまま進める

経営層・IT部門だけで決めて現場に押し付けると、完成しても誰も使わない「幽霊システム」になる

❌ 失敗パターン: 現場置き去りフロー



✅ 成功パターン: 現場を巻き込むフロー



ポイント:「現場巻き込み」とは説明会の開催ではない。要件定義・モックアップ確認・テストの各フェーズで現場担当者が意見を言える場を設けること。「自分が参加して作ったシステム」という意識が生まれると、定着率は劇的に上がる。

業務フロー設計シリーズ 第1部 | まとめ

5つの失敗を知れば、あなたのシステム導入の成功確率は大きく上がる

① 要件定義をしてから進める(AI自社開発・外注どちらにも共通)

② 御用聞きではなく、事業を理解するパートナーを選ぶ

③ フロント～バックを横断して設計できるベンダーを選ぶ

④ モックアップ(画面設計)を事前に確認してから発注する

⑤ 現場担当者を最初から巻き込む

▶ 次回(第2部):システム開発前に揃えるべき「準備ドキュメント」の作り方

業務フロー設計シリーズ | 第2部

システム開発を発注する前に 「絶対に」揃えておくべき準備

～ ドキュメント整備・業務録画・As-Is業務フロー図の作り方 ～

aurant technologies

準備のゴールと3つの要素

「準備できている発注者」と「準備できていない発注者」では、ベンダーから受け取る提案書の品質が全く違う

01

As-Is業務フロー図の作成

現状の業務を担当者・手順・ツール・痛みポイントの4要素でスイムレーン図にする。「今何が問題か」を全員が視覚的に共有できる状態を作る。

02

現在使用ツールの棚卸し

今使っているSaaS・システム・Excelを全て書き出す。ベンダーはこのリストを見て「何と連携が必要か」を判断するため、必須の準備物。

03

業務動作の録画・ヒアリング会の設定

テキストで伝えにくい「実際の業務動き」をZoomやLoomで録画する。4時間のヒアリングと同等の情報量をベンダーに渡せる。

①As-Is業務フロー図(スイムレーン)の例

「受注～請求」の典型的な業務フロー。現状の痛みポイント(⚠)を可視化することがシステム開発の出発点になる

	①見積依頼 受付	②見積書 作成	③受注 確定	④在庫 確認・確保	⑤出荷手配	⑥請求書 発行
営業担当者	TEL/FAX 受付・メモ ⚠ FAX受信	Excelで 見積書作成 ⚠ Excelゼロから	注文書を FAX受信 ⚠ FAX/PDF	—	—	請求書 送付確認
在庫・ 物流担当	—	—	受注台帳 に手記入 ⚠ 転記作業	在庫表を 目視確認 ⚠ Excel在庫表	ピッキング 梱包・出荷	—
経理 担当者	—	—	—	—	—	Excelで 請求書作成 ⚠ Excel手打ち
顧客	見積依頼 (電話/メール)	見積書 受取・確認	注文書 送付(FAX)	—	商品 受取・検収	請求書 受取・振込

②現在使用ツール・システムの棚卸し

「今何を使っているか」をすべて書き出す。ベンダーはこのリストを見て連携設計・移行計画・リスク評価を行う

フロントオフィス	ミドルオフィス	バックオフィス
<p>Salesforce</p> <p>用途: 商談・顧客管理 形式: SaaS</p> <p>連携: × freeeに手動転記</p>	<p>受注管理Excel</p> <p>用途: 受注台帳管理 形式: Excel</p> <p>連携: ⚠ 手入力・属人化</p>	<p>freee会計</p> <p>用途: 会計・仕訳 形式: SaaS</p> <p>連携: × 手動仕訳入力</p>
<p>Mailchimp</p> <p>用途: メールマーケ 形式: SaaS</p> <p>連携: × CSV手動エクスポート</p>	<p>在庫管理Excel</p> <p>用途: 在庫数管理 形式: Excel</p> <p>連携: ⚠ リアルタイム不可</p>	<p>Excelテンプレート</p> <p>用途: 請求書作成 形式: Excel</p> <p>連携: ⚠ 全件手打ち</p>
<p>弊社ECサイト</p> <p>用途: オンライン受注 形式: 独自開発</p> <p>連携: ⚠ API未整備</p>	<p>kintone(検討中)</p> <p>用途: 業務管理基盤 形式: 検討中</p> <p>連携: ? 未定</p>	<p>マネーフォワード</p> <p>用途: 給与・勤怠 形式: SaaS</p> <p>連携: ○ 独立運用</p>

⚠ ポイント:「⚠ Excel手管理」「× 手動転記」が多い箇所こそ、システム化の優先ターゲット。この棚卸し表が RFPの核心部分になる。

補足) DB設計書・ER図があると開発品質が格段に上がる

ドキュメントが整っている発注者には、ベンダーが「設計に集中できる提案書」を出してくれる。整っていない発注者への提案書は「假定だらけ」になる

ER図(エンティティ関係図)の例

顧客マスタ

customer_id (PK)

company_name

contact_name

email

phone

created_at

1 : N
(1顧客が複数受注)

商品マスタ

product_id (PK)

product_name

unit_price

stock_qty

category

updated_at

受注テーブル

order_id (PK)

customer_id (FK)

order_date

status

total_amount

notes

1 : N
(1受注に複数明細)

受注明細

line_id (PK)

order_id (FK)

product_id (FK)

quantity

unit_price

subtotal

N : 1
(商品→明細)

ドキュメントが整うと何が変わるか

設計の手戻りが減る

ベンダーが「顧客IDと注文IDの関係は？」という確認作業に時間を使わなくなる。設計精度が上がり、要件変更リスクが下がる。

見積の精度が上がる

テーブル構造・データ量・連携先が明確だから、ベンダーが正確な工数を算出できる。「後から追加費用」が大幅に減る。

複数ベンダーを公平に比較できる

同じ設計書を渡せば、各社の提案が「同じ条件」での比較になる。金額だけでなくアプローチの違いも評価できる。

「ドキュメントがない場合」の準備法

既存システムがなく書類もない場合は、業務担当者へのインタビューを録画(Zoom/Loom)して渡すのが最速。詳細は次スライドで説明。

③業務動作を「録画」してベンダーに渡す方法

「ドキュメントが整っていない」場合でも、実際の業務画面を録画して渡すことで 4時間のヒアリングと同等の情報量をベンダーに渡せる

Step 1

録画ツールを用意する

Zoom(無料)またはLoom(無料プランあり)を使う。
Zoomは「レコーディング」ボタンひとつで録画できる。Loomはブラウザ拡張で画面録画が可能。

Step 2

録画する業務シナリオを決める

「受注メールを受け取ってから受注台帳に入力するまで」のように1本10~20分程度のシナリオを3~5本選ぶ。
重要:実際の業務データ(個人情報含む可能性)は必ずマスクして録画すること。

Step 3

担当者が実際に業務を実演しながら録画

画面を共有しながら「今これをxcellに転記しています。ここが月末に60件あって大変です」と口頭で説明しながら作業する。
説明しながらやること:ツールの切り替え、コピーペースト、電話確認、手書き転記など

Step 4

痛みポイント・疑問点を口頭でコメント

「このシステムが重くて毎回フリーズする」「在庫システムとExcelが連動していないので毎回目視確認する」など、現場の「困りごと」を実際の画面を見せながら話してもらう。

Step 5

ベンダーへの提供時の注意事項

録画ファイルをGoogleドライブ・OneDriveに置いて共有URL送付が最速。
個人情報映っている場合はモザイク処理ツール(Screencast-O-Matic等)を使う。
「As-Isフロー図」「ツール一覧」とセットで渡すと効果が倍増する。

綺麗な動画でなくても
OKです！

補足)紙は全部渡す

帳票類等は特に、カスタマイズ箇所が大きく、開発費が膨らむ要因となる箇所です。
ベンダーに依頼する際は、「帳票」をすべて渡すようにしておきましょう。それだけで開発ベンダー側の理解度は上がります。



Microsoft Excel spreadsheet showing a request form template. The title is "請求書" (Invoice).

Fields include:

- XXXXX 御中 (Customer Name)
- 請求日 (Request Date)
- 請求番号 (Request Number)
- 件名 (Subject)
- ご請求金額 (Request Amount)

#	商品	商品コード	価格	数量	小計
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					小計
20					消費税
21					合計

備考 (Remarks):

- 設置サポート有無 (Installation support presence)
- 契約条件 (Contract conditions)

準備の有無で「受け取る提案書」がここまで違う

同じ「受注管理を効率化したい」という依頼でも、準備の質でベンダーの提案書の中身が 180度変わる

✖ 準備なし発注が受け取る提案書

ご提案内容

弊社のkintoneおよびSalesforce連携ソリューションで受注管理の効率化が可能です。

■ 機能概要

- ・受注データの一元管理
- ・自動通知機能
- ・レポート出力機能

■ 概算費用:300万円～

詳細はヒアリング後に確定します。

問題点:課題の具体性がゼロ。「なぜこの機能か」「御社のどの業務が改善されるか」の説明がない。費用の根拠も不明。

✔ 準備あり発注が受け取る提案書

現状課題の整理(ご共有頂いた資料より)

As-Isフロー分析:受注確定後の台帳転記作業に月80時間・ミス月12件が発生。主因はExcel台帳と在庫システムの非連携。

提案アプローチ:

- ①受注データのkintone一元管理(転記ゼロ化)
- ②在庫システムAPI連携(リアルタイム在庫把握)
- ③freee自動仕訳連携(月次3日→0.5日)

期待効果:月80h削減・ミスゼロ・3ヶ月ROI回収見込み

実装工数(詳細):kintone構築80h・連携開発120h・テスト40h

システムフロー図

ポイント:準備を整えた発注者への提案書は「御社固有の課題・解決アプローチ→期待効果→費用根拠」が揃っている。比較・検討できる内容になっている。

VS

業務フロー設計シリーズ | 第3部

システム開発の成功確率を劇的に上げる RFP(提案依頼書)の作り方 完全版

～ 機能要件・非機能要件・契約形態・ベンダー評価まで全解説 ～

aurant technologies

01

RFI,RFP,RFQの違い

RFP作成の前に知っておくべき「3つの依頼書」の違い

RFI・RFP・RFQは別物。プロセスの順番と目的を理解することで、ベンダー選定を正しい手順で進められる

RFI

情報提供依頼書

Request for Information

RFP作成前の
情報収集段階

目的

- ①どんなベンダーが市場にいるか
- ②どんな技術・アプローチがあるか
- ③大まかな費用感の把握

送付先: 候補 10～15社に送付

→ 期待結果: 候補を 3～5社に絞り込む

RFP

提案依頼書

Request for Proposal

ベンダー選定の
核心段階

目的

- ①詳細な業務課題・要件の提示
- ②ベンダーに具体的提案を依頼
- ③各社を同条件で比較評価する

送付先: 絞り込んだ 3～5社に送付

→ 期待結果: 最終候補 1～2社を選ぶ

RFQ

見積依頼書

Request for Quotation

仕様確定後の
最終価格決定段階

目的

- ①仕様が固まった後の正式見積
- ②価格・支払条件の最終確認
- ③契約に向けた数字の確定

送付先: 最終候補 1～2社に送付

→ 期待結果: 契約先ベンダーを決定する

このシリーズで最も重要なのが「RFP」。RFIとRFQは補助的な位置づけ。第3部では、良いRFPを構成する全要素を順番に解説する。

02

RFPの全体像

RFPを構成する「16の必須項目」全体マップ

良いRFPとは「16の項目が揃っているもの」。どこか欠けると、ベンダーは仮定で提案書を書かざるを得なくなる

A. プロジェクト概要

01 表紙・背景

なぜこのプロジェクトを立ち上げるのか。経営的な背景と目的を明示。


02 会社・事業概要

業種・規模・組織・主要サービス。ベンダーが業務文脈を理解するために必要

03 現行システム構成図

現在使っているシステム・SaaS・Excelを全て図示。連携状態も記載

04 As-Is業務フロー図

現状業務を担当者・ツール・痛みポイントでスイムレーン図化(第2部参照)

B. 課題と要件定義

05 課題の定量化

「月80時間・ミス12件」のように数値で書く。曖昧な形容詞は禁止

06 To-Be(実現したい姿)

改善後の業務フロー。「月80h→5h以下」のKPIも添える

07 機能要件(Must/Want)

実装してほしい機能をMust(必須)/ Want(あればよい)で分類する

08 非機能要件(6大項目)

性能・セキュリティ・可用性・移行性・保守性・環境。IPAグレード参考

C. 実施条件

09 データ移行要件

現行データをどう移行するか。件数・形式・クレンジング方針を書く

10 テスト要件

UAT(受入テスト)の体制・合格基準・スケジュールを明記する

11 教育・トレーニング要件

誰に・どの段階で・どんな教育をしてほしいかを書く

12 納品物・成果物一覧

プログラム以外に受け取る設計書・テスト結果・運用マニュアルを指定

D. 選定条件

13 スケジュール

月次・週次単位でのマイルストーン。自社側タスクも明示する

14 予算・契約形態

予算上限・準委任 or 請負の指定・フェーズ別契約可否を書く

15 プロジェクト体制要件

PMの経験年数・チーム規模・キーパーソンのCV提出を求める

16 選定基準・評価軸

価格・機能・実績・体制の重み付けを事前公開する(スコアカード)

機能要件の書き方 — Must / Want / 対象外の3分類

「機能一覧を羅列するだけ」の RFPは最悪。Must(必須)とWant(あればよい)を分けないとベンダーが見積れない

Must(必須)

これがなければ発注しない

欠けた場合は発注不可と明示する機能。最小限に絞ること。Mustが多すぎると提案書が「できません」だらけになる。

書き方例:

- 受注データの kintone登録
- freee自動仕訳連携(月次自動)
- 在庫リアルタイム同期
- PC・スマートフォン両対応

Want(あればよい)

あれば加点評価する機能

「提案があれば歓迎」な機能。評価時の加点対象。要件が固まっていない機能はWantに分類する。

書き方例:

- 受注予測・売上ダッシュボード
- Slackへの通知連携
- バーコードスキャン対応
- モバイルオフライン動作

Out of Scope(対象外)

今回は含まない・別途検討

明示しないと「それも含むのか?」という確認が増えて工数が膨らむ。今回対象外を書くことも重要。

書き方例:

- 給与・勤怠管理システム連携
- 海外拠点・多通貨対応
- EC(外部公開)機能
- CRM(Salesforce)のリプレイス

機能要件の3分類(画面・帳票・バッチ)で整理するとベンダーが工数算出しやすい

画面(UI)

受注一覧画面・受注入力フォーム・在庫照会画面・ダッシュボード
→ 一般的に1画面あたり設計・開発・テストで30~80時間が目安

帳票(帳票出力)

請求書PDF・納品書・受注確認書・月次売上レポートCSV
→ 1帳票あたり15~40時間が目安。帳票数が多いと費用が跳ね上がる

バッチ処理(自動化)

夜間在庫同期・月次 freee自動仕訳・売掛残高計算・メール自動送信
→ 1バッチあたり20~60時間。複雑なデータ変換がある場合は大幅増

API連携(外部接続)

freee API・kintone API・Salesforce API・在庫システムAPI
→ 連携先1つあたり20~80時間。先方のAPI仕様次第で大きく変動

非機能要件 — 見落とすと稼働後に必ず後悔する「裏側の要件」

機能が完成してもシステムが重い・落ちる・セキュリティが甘い・移行できない、これらは全て非機能要件の書き漏れが原因

① 可用性

Availability

システムが「いつでも使える状態」であることへの要求。

- 稼働率:99.5%以上(年間43時間以内の停止許容)
- 計画停止窓口:毎週土曜01:00~05:00 可
- 障害時の最大復旧時間(RTO):4時間以内
- 障害時のデータ損失許容量(RPO):1時間分以内

⚠ 書かないと:「落ちても責任取れません」と言われる

② 性能・拡張性

Performance

システムの「速さ・重さ・将来の拡張」への要求。

- 通常時ページ表示速度:3秒以内
- ピーク時同時アクセス:100人まで
- データ量:受注データ100万件まで対応
- 将来3年で利用者2倍想定、拡張可能な設計

⚠ 書かないと:繁忙期に激重になって現場が使えない

③ 運用・保守性

Maintainability

稼働後の「日常運用・バグ修正・バージョンアップ」への要求。

- バックアップ:日次自動・30日間保存
- ログ保存:操作ログ1年間保存
- 月次メンテナンス窓口の指定
- 障害発生時の連絡先・エスカレーション手順

⚠ 書かないと:障害時に誰に連絡すべきか不明になる

④ 移行性

Portability

「既存データの引越し」と「システム乗換時のリスク」への要求。

- 移行対象:受注データ5万件・顧客マスク3,000件
- データ形式:CSV/Excelで受け取り想定
- 移行テスト:本番環境で3回実施後に本番移行
- 契約終了時のデータエクスポート形式を確約

⚠ 書かないと:移行失敗で数ヶ月の二重入力が発生

⑤ セキュリティ

Security

「不正アクセス・情報漏洩」を防ぐための技術的・管理的要求。

- 認証:二要素認証(2FA)必須
- 通信:HTTPS/TLS1.2以上
- 権限管理:部門・役職別アクセス制御RBAC)
- 脆弱性診断:リリース前に実施・報告書提出

⚠ 書かないと:情報漏洩事故が起きても契約上無責任

⑥ システム環境

Environment

「動作環境・インフラ」への要求。クラウドvs オンプレも含む。

- 動作環境:Chrome最新・Edge・Safari(Mac)対応
- クラウド:AWS or GCPを推奨(オンプレ不可)
- モバイル:iOS/Android最新2世代対応
- 既存環境との共存期間:旧システム3ヶ月並行稼働

⚠ 書かないと:動かないブラウザ・OSで「仕様です」と言われる

「契約形態」の選び方 — 準委任 vs 請負を間違えると後で大揉めする

「請負にすれば安心」は誤解。フェーズごとに使い分けるのが正解。契約形態の選択がトラブル予防の最重要事項のひとつ

比較項目	準委任契約	請負契約
完成責任	なし →ベンダーは「作業する」義務。完成できなくても支払い義務が生じる	あり →ベンダーは「完成させる」義務。完成しなければ代金請求できない
追加費用のリスク	時間単価×工数で計算。作業が増えれば費用が増える →仕様変更が多い要件定義フェーズ向き	見積固定が多い。スコープ外は追加費用。 →要件が明確な設計・開発フェーズ向き
ベンダーの見積姿勢	リスクが低いので正直な工数見積になりやすい →実際の仕事量に近い金額が出やすい	リスクを嫌ってバッファを積む →見積が高くなる傾向がある。競争入札で引き下げを
向いているフェーズ	要件定義・ヒアリング・業務設計 →「何を作るか」がまだ決まっていない段階	設計・開発・テスト →「何を作るか」が決まってからの実装段階
実務でよくある失敗	「準委任なのにベンダーが完成責任を持つと思っていた」 →認識ズレでトラブルになる	「請負なのに仕様変更を繰り返した」 →追加費用が雪だるま式に増える

「良いRFP」の実際の文面 — 課題定量化から成功基準まで

この書き方で RFPを送ると、ベンダーは「御社の課題を理解している」と判断し、仮定なしの的確な提案書を返してくれる

【第1章: 背景と現状課題の定量化】

- 対象業務: 受注管理～請求業務(営業・物流・経理の 3部門)
 - 現状の課題(定量データ)
 - ① 受注台帳への手入力: 月 200件 × 8分 = 月26.7時間のロス(時給 2,500円換算で月6.7万円)
 - ② 転記ミス: 月平均 12件発生。修正対応 30分/件 = 月6時間の追加工数
 - ③ 在庫確認の電話: 受注ごとに物流担当へ電話。月 200回 × 5分 = 月17時間
 - ④ 請求書をWordで作成: 月 50件 × 20分 = 月17時間(入力ミス月 4件)
- 合計: 月66.7時間(年間800時間・人件費年間 200万円)が「システム化で不要になる工数」

【第2章: 実現したい状態(To-Be)と成功基準(KPI)】

- To-Be: 受注入力から在庫確認・出荷指示・freee自動仕訳まで一気通貫。転記作業ゼロを実現する。
- 必須KPI(Mustの成功基準)
 - ✓ 受注処理工数: 月 26.7h → 5h以下(80%削減)
 - ✓ 転記ミス: 月 12件 → 0件
 - ✓ 請求書発行: 月 17h → 2h以下(88%削減)
 - ✓ 在庫確認電話: 月 200回 → 10回以下(95%削減)
- ROI目標: 投資回収 12ヶ月以内(現状の年間コスト 200万円 ÷ 初期投資 X = 回収期間)

【第3章: 予算・スケジュール・契約形態】

- 予算: 初期費用 250～400万円(ROI12ヶ月以内が最重要評価基準)
- 稼働希望: 2025年10月1日(交渉可・但し11月以降は不可)
- 契約形態希望: 要件定義フェーズ → 準委任、設計～開発～テストフェーズ → 請負
- 提案書提出期限: ○月○日17:00まで。提案後プレゼン(1社1時間)を実施予定

ベンダー評価スコアカード — 感情ではなく「数値」で選ぶ

「安いから」「大手だから」でベンダーを選ぶのは最悪の意思決定。評価を数値化して全員が納得できる選定をする

評価カテゴリ・評価項目	重み	配点	ベンダーA	ベンダーB	ベンダーC
A. 課題理解力(重み25%)					
RFPの課題定量値を正確に理解・言及しているか	25%	/10	8	6	9
「なぜその機能が課題解決につながるか」を説明できているか	25%	/8	10	8	7
As-Isフロー図への具体的言及があるか	25%	/7	11	9	8
B. 解決アプローチの質(重み30%)					
フロント～バックの一气通貫設計が提案に含まれているか	30%	/12	7	8	10
他システムとの連携設計(API設計・データフロー)が明示されているか	30%	/10	6	8	7
実装の優先順位と理由が説明されているか	30%	/8	15	12	10
C. 期待効果の根拠(重み20%)					
KPI達成の根拠(なぜ月26h→5hになるか)が書かれているか	20%	/12	9	7	8
類似業種・規模での実績KPIが示されているか	20%	/8	5	6	7
D. 体制・信頼性(重み15%)					
PMの経験年数・同業種実績が明示されているか	15%	/8	6	8	7
リスク・懸念点を自ら開示しているか	15%	/7	4	5	4
E. コスト適正性(重み10%)					
費用内訳(設計・開発・テストPM)が明確か	10%	/6	8	6	9
ROI12ヶ月以内の条件を満たしているか	10%	/4	10	8	7
合計スコア(100点満点)			78	74	82

見落としがちな「移行・テスト・教育」要件 — 書かないと稼働後に大炎上

開発は成功したのに移行で失敗、テストが不十分で現場クレーム、誰も使えないで放置 — これら全て RFP に書けば防げる

📦 データ移行要件

現行データの「引越し計画」をRFPに書く

移行対象データ

受注データ:5万件(2020年～)
顧客マスタ:3,000社
商品マスタ:2,500点
在庫残高:現在時点の残高のみ

移行前のデータ整備

重複顧客の名寄せ(数百件想定)
住所・電話番号の表記ゆれ統一
廃止商品コードの処理方針を決定

移行テスト

本番移行の前に3回のリハーサルを実施
移行後のデータ件数・金額の一致確認
差異が出た場合の対処ルールを定める

✅ テスト要件

「誰が・何を・どの基準で」テストするかをRFPで指定する

単体・結合テスト

ベンダー責任で実施
テスト仕様書・結果報告書の納品必須
不具合はS/A/B/Cでランク付けし期限内修正

UAT(受入テスト)

発注者側の現場担当者が実施(3週間)
合格基準:Sランク不具合0件・Aランク不具合0件
UAT通過後に本番稼働を承認する

本番移行後の保証期間

本番稼働後3ヶ月間は無償で不具合対応
4ヶ月目以降は保守契約(別途見積)

🎓 教育・トレーニング要件

「誰に・いつ・どうやって」教えるかをRFPに書く

管理者トレーニング

システム管理者(2名)向け:操作・権限設定・バックアップ方法
リリース2週間前に4時間の集合研修を実施

現場ユーザートレーニング

営業・物流・経理各部門(計20名)向け
リリース1週間前に部門別2時間研修を実施
操作マニュアル(PDF)の納品必須

フォローアップ支援

本番稼働後2週間:ベンダー担当者が常駐(または随時リモート対応)
稼働1ヶ月後に「つまずきポイントQ&A会」を1回開催

⚠️ RFPでこの3要素を書いていない発注者には、ベンダーは「開発のみで納品します」の薄い提案書を返してくる。移行・テスト・教育まで含めることで「本当の導入成功」が実現する。

業務フロー設計シリーズ 第3部 | まとめ

良いRFPとは「ベンダーが仮定なしで提案書を書ける状態を作ること」

- ① RFI→RFP→RFQの順番でプロセスを進める(いきなり RFPを多数に送らない)
- ② RFPの16項目を揃える(As-Is・課題定量化・機能要件・非機能要件・契約形態・選定基準)
- ③ 機能要件はMust / Want / 対象外の3分類で書く(全部Mustにしない)
- ④ 非機能要件(IPA 6大項目)を書く(可用性・性能・保守・移行・セキュリティ・環境)
- ⑤ 契約形態を工程ごとに使い分ける(要件定義＝準委任、開発＝請負が基本)
- ⑥ 移行・テスト・教育要件を書く(「完成させるだけ」では導入成功にならない)
- ⑦ ベンダー評価はスコアカードで数値化する(感情・印象・価格だけで決めない)

業務フロー設計シリーズ | 第4部

せっかく入れたシステムを 「誰も使わない」悲劇を防ぐ

～ 現場定着の運用設計 完全版 ～

aurant technologies

定着しない3つの根本原因

「システムが悪かった」ではなく「定着設計がなかった」のが原因。この 3つを理解することが改善の出発点

01 現場が「置き去り」にされた

経営層とIT部門だけでシステムを選定し、現場担当者は「来月から新システムを使え」と事後告知された。自分が意見を言えなかったシステムへの愛着は生まれない。「なぜ私に相談しなかったのか」という不満が先に立ち、使う気持ちにならない。

→ 現場が「使わされている」と感じると定着率は 20%以下になる

02 トレーニングが「1回きり」だった

導入直前に1回だけ説明会を開いて「あとはマニュアルを読んでもください」で終わった。人間は1回の説明では定着しない。繰り返し使うことで初めて体に染み込む。1週間後には説明の半分以上を忘れている。「わからないから旧手順に戻る」のは当然の行動。

→ 稼働後のフォローなしでは 3ヶ月後の利用率は 30%台に低下する

03 旧システムへの「逃げ道」が残った

新システムを導入しながら、旧 ExcelやFAXも並行使用できる状態を放置した。逃げ道があると「急いでいるとき」「迷ったとき」「やり方がわからないとき」に必ず旧手順に戻る。これは意志の弱さではなく人間の自然な行動。並行運用ルール不在が諸悪の根源。

→ 並行運用期間が 6ヶ月を超えると旧手順への完全回帰が起きる

定着設計① 巻き込み設計 — 最初から現場を参加させる

「説明会の開催」は巻き込みではない。要件定義・モックアップ確認・テストの各フェーズで現場担当者が意見を言える場を設けることが「真の巻き込み」

✗ よくある「置き去り」パターン

経営・IT部門だけで
ベンダーを選定

現場への相談・説明ゼロ

仕様決定・発注

「完成したら使ってもらえばいい」

開発完了の直前
現場に初めて告知

「来月から新システムです」

説明会を1回開催

「あとはマニュアルで」

誰も使わない
幽霊システム完成

数百万円が無駄に

VS

✓ 定着する「巻き込み」パターン

現場担当者を
要件定義ヒアリングに同席

「自分の意見が反映された」という感覚

モックアップを現場に見せて
意見をもらう

「このボタンはここに移動してほしい」→ 反映

テスト段階で
現場がパイロット参加

「自分がテストした」という当事者意識

4段階トレーニング実施
(次スライドで詳述)

管理者→キーユーザー→全員→フォロー

現場が自走する
「使われるシステム」

定着率90%以上を達成

定着設計② 4段階トレーニングロードマップ

「1回の説明会で終わり」は定着しない。本番稼働前後を 4フェーズに分けて、段階的にスキルと自信を積み上げる

Phase 1	Phase 2	Phase 3	Phase 4
管理者トレーニング	キーユーザートレーニング	全ユーザートレーニング	稼働後フォローアップ
本番稼働 2週間前	本番稼働 1週間前	本番稼働 直前	本番稼働後 2~4週間
対象: 管理者2~3名 / 4時間	対象: 各部門1~2名 / 3時間	対象: 全ユーザー / 2時間	対象: 全員 継続支援 / 継続的
<ul style="list-style-type: none">システムの全機能を深く理解ユーザー権限設定の方法バックアップ・障害対応手順社内ヘルプデスクとしての役割	<ul style="list-style-type: none">自担当部門の操作を完全習得よくあるトラブルの対処法新人への教え方の練習「部門の先生」としての心構え	<ul style="list-style-type: none">自分の担当業務の操作だけに絞る全機能の説明はしない(情報過多を避ける)実際の業務データを使ったハンズオン「わからなければキーユーザーに聞く」を徹底	<ul style="list-style-type: none">毎日15分のQ&Aタイムを設けるチャットで質問をリアルタイム受付週次でキーユーザーと振り返りを実施小さな成功体験を積極的に共有・可視化
ゴール	ゴール	ゴール	ゴール
社内の「何でも聞いていい人」を先行育成する	「営業は田中さん、物流は鈴木さんに聞く」体制を作る	「自分が使う部分だけ」を体で覚えさせる	「わからない→旧手順へ逃げる」の連鎖を断ち切る

定着設計③ 移行設計 — 旧手順への「逃げ道」を計画的に塞ぐ

「逃げ道を塞ぐ」とは意地悪ではない。逃げ道があると現場は新システムに真剣に向き合えない。計画的に廃止することが現場への最大の愛情

July 17 決定事項①

旧システムの廃止タイムラインを決める

「新システム稼働から3ヶ月後に旧Excelの更新を禁止する」のように、具体的な日付を決める。

決めないと永遠に並行運用が続き、新システムの利用率が上がらない。タイムラインは関係者全員で合意しておくことが必須。

⚠ タイムラインを決めないと旧手順が半永久的に生き残る

決定事項②

並行運用期間のルールを明文化する

稼働直後は一定期間の並行運用が必要な場合もある。ただし「どのデータだけを二重管理するか」「並行運用はいつまでか」を明確に決めておく。

曖昧にすると「まあ旧Excelも使っているよ」という空気が生まれ、新システムへの移行が遅れる。

⚠ 並行運用ルールが曖昧だと6ヶ月後も旧手順が残る

決定事項③

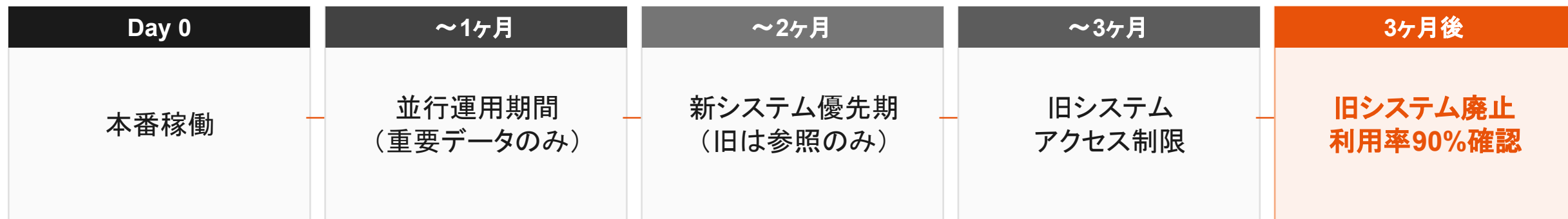
移行後の成功基準（KPI）を設定する

「稼働3ヶ月後に新システムの利用率90%以上」のような数値目標を設定する。

目標がないと推進チームのモチベーションが続かない。KPIがあることで「今どこにいるか」が見えて、行動の方向性が揃う。

⚠ 成功基準がないと定着の進捗が「感覚」になる

典型的な移行タイムライン例



定着KPIの設計 — 3層構造で「感覚」ではなく「数値」で測る

定着を「なんとなく感じる」で判断するな。第 1層の利用率から第 3層の満足度まで、月次で計測・報告することで改善アクションが生まれる

第1層

利用率KPI

システムが実際に使われているか

ログイン率

対象ユーザーの何%が週1回以上ログインしているか

目標: 90%以上

主要機能利用率

受注入力・在庫確認など主要機能が全案件の何%で使われているか

目標: 95%以上

旧手順の残存率

旧Excelや旧FAX手順が今も何件使われているか
(ゼロを目指す)

目標: 0件

第2層

業務改善KPI

導入効果が実際に出ているか

転記工数削減率

月26時間→何時間になったか。毎月計測して推移を追う

目標: 80%削減(5時間以下)

ミス件数

転記ミス・入力ミスが月何件発生しているか

目標: 0件

処理リードタイム

受注確定から出荷指示まで何日かかっているか

目標: 3日→1日以内

第3層

満足度KPI

現場の「気持ち」はどうか

ユーザー満足度スコア

月次アンケートで「このシステムを使いやすいか」を5段階評価

目標: 4.0以上

「旧に戻りたい」発言件数

キーユーザー経由で収集。ゼロになるまで改善を続ける

目標: 0件

サポート問い合わせ件数

月を追うごとに減少していれば定着の証拠。増加したら要因分析

目標: 月次で減少傾向

抵抗する人への向き合い方 — 3タイプとそれぞれの攻略法

「この人が悪い」ではない。抵抗には必ず理由がある。タイプを見極めて適切な対処をすることが定着推進リーダーの仕事

😞 タイプ①

「今のやり方の方が慣れてる」型

抵抗の理由

変化への抵抗ではなく「不慣れ」なだけ。新しい操作に自信がなく、失敗するのが怖い。悪意は一切ない。

よくある言動・サイン

- 説明会中は黙っている
- 「後で見ます」と言って動かない
- こっそり旧Excelを使い続けている

攻略法: 個別ハンズオンで「慣れ」を作る

キーユーザーが横に座って「一緒にやってみましょう」と1件だけ実際に操作してもらおう。成功体験が1つできれば、あとは自走する。

😞 タイプ②

「自分の仕事がなくなる」型

抵抗の理由

「Excel職人」として重宝されてきた自信と、それが失われる恐怖。変化への抵抗の裏に、「自分の存在価値への不安」がある。

よくある言動・サイン

- 「今のやり方で十分です」と繰り返す
- システムの欠点を過剰に指摘する
- 上司に「導入反対」を直接訴える

攻略法: 「役割の再定義」を具体的に伝える

「あなたの業務経験と知識は、これからのデータ分析や業務改善の提案で必要です。重労働から解放されて、より価値の高い仕事に集中してほしい」と正面から伝える。

😞 タイプ③

「このシステムが信用できない」型

抵抗の理由

過去のシステム導入が失敗した苦い経験。または「ITアレルギー」。「どうせまた使えないものが来た」という不信感が根底にある。

よくある言動・サイン

- 「前もそうだった」という発言
- 些細なバグを大げさに報告する
- キーユーザーになることを断る

攻略法: 「小さな成功体験」を積み重ねる

フェーズ4のフォローアップを丁寧に行い、「何かあれば即対応します」を実際の行動で示す。「転記作業が本当になくなった！」という体験の一つ作ることが最大の薬になる。

定着チェックリスト — 稼働前・稼働後の 10項目

この10項目をクリアした状態で本番稼働すれば、定着率は大きく上がる。逆に言うと、未チェックの項目があれば稼働を延期する勇気も必要

本番稼働前に確認する5項目

- 01** 現場担当者が要件定義・モックアップ確認に参加したか
→ 参加なしで稼働すると「押し付けられた感」が生まれ、定着率が低下する
- 02** 管理者トレーニングが完了しているか
→ 管理者が使い方を知らなければ、現場の質問に誰も答えられない
- 03** キーユーザーが各部門に1名以上配置されているか
→ キーユーザー不在の部門では「わからない → 旧手順へ逃げる」が起きやすい
- 04** 操作マニュアル(PDF)の納品を受けたか
→ マニュアルなしでは現場が「調べる場所がない」状態になる
- 05** UAT(受入テスト)を現場担当者が実施したか
→ テスト参加で当事者意識が生まれる。バグ発見も稼働前に完了できる

本番稼働後1ヶ月以内に確認する5項目

- 06** 毎日の質問窓口(チャット等)が機能しているか
→ 稼働直後が最も質問が多い。窓口がないと「わからない → 諦める」になる
- 07** 週次でキーユーザーと振り返りを実施しているか
→ 問題の早期発見と対処ができる。キーユーザーの疲弊も防げる
- 08** 利用率KPIを計測・報告しているか
→ 測らないと問題に気づけない。数値で見える化することが改善の第一歩
- 09** 抵抗している担当者への個別フォローができているか
→ 放置すると周囲にも「使わなくていい」という空気が伝染する
- 10** 旧手順への逆戻りが起きていないか確認しているか
→ 逆戻りは早期発見・早期対処が鉄則。1ヶ月放置すると定着が困難になる

業務フロー設計シリーズ | 全4部 総まとめ

正しい準備と正しい進め方を知っていれば、システム導入の失敗確率は大幅に下げられる

第1部	第2部	第3部	第4部
システム導入が失敗する「5つの理由」	発注前に揃える「4つの準備」	RFP(提案依頼書)の作り方 完全版	現場定着の「運用設計」
① 要件定義がないまま進める	① As-Is業務フロー図を作る	① 機能要件: Must/Want/対象外の3分類	① 巻き込み設計(要件定義から現場を参加させる)
② 御用聞きベンダーを選ぶ	② ツール棚卸し表を作る	② 非機能要件: IPA 6大項目	② 4段階トレーニング(1回説明会で終わらせない)
③ 特定ツール専門ベンダーに依存	③ 業務動作を録画してベンダーに渡す	③ 契約形態: 準委任 vs 請負を使い分ける	③ 移行設計(旧手順への逃げ道を計画的に塞ぐ)
④ モックアップなしで発注	④ 課題を定量化する(時間・件数・日数)	④ ベンダー評価: スコアカードで数値化	④ 定着KPI 3層(利用率・業務改善・満足度)
⑤ 現場を置き去りにする			

aurant technologiesは、フロント～バックオフィスの一気通貫設計・実装・定着支援が専門です。
「自社に当てはめたら？」という相談はこちらのリンクからどうぞ。
<https://aurant-technologies.com/contact/>

Aurant Technologies - 会社説明資料

AI×開発による

事業が変わるビジネス体験

01

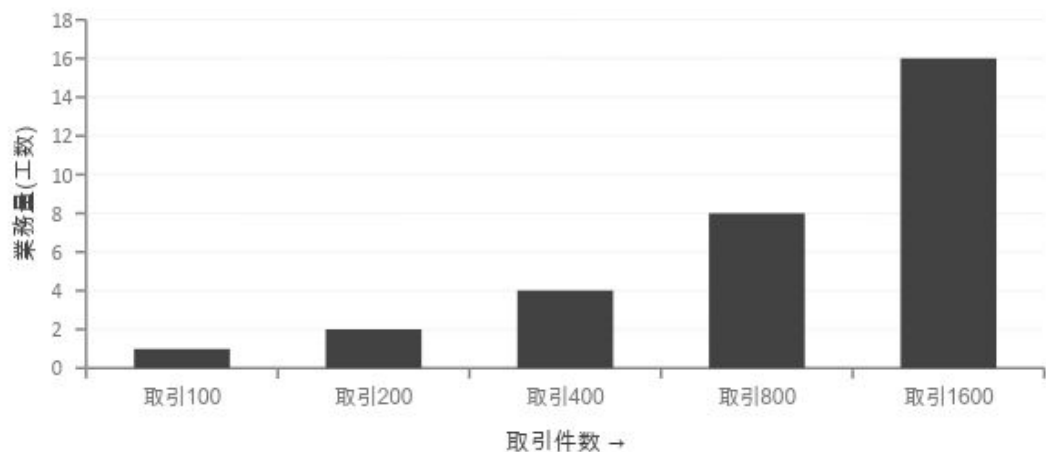
事業紹介 -

Our Business

なぜ多くの企業でDXが形骸化するのか？

SaaSを導入しても「人がシステム間の橋渡し」をしている限り、取引が増えれば人も増え続ける

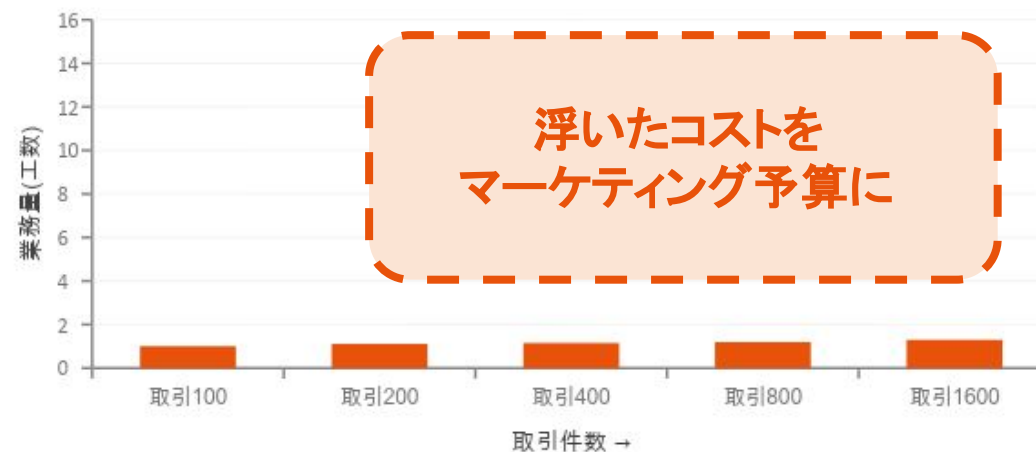
手作業モデル(従来)



取引2倍 → 人員2倍

SaaSを入れても、システム間の橋渡しは人間。
事業拡大=人員増のジレンマが続く

AIエージェント・自動化モデル



取引2倍でも人員据え置き

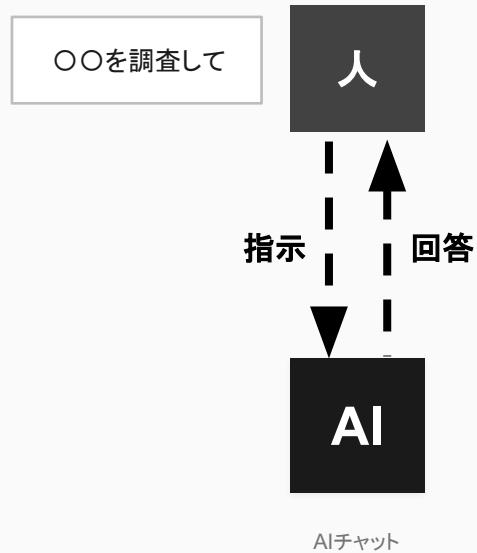
APIとAIエージェントがシステム間を繋ぐため、
取引件数が増えても業務量はほぼフラット

ここを解決するのが、Aurantのソリューションです

時代は、AIチャットボットからエージェントへ

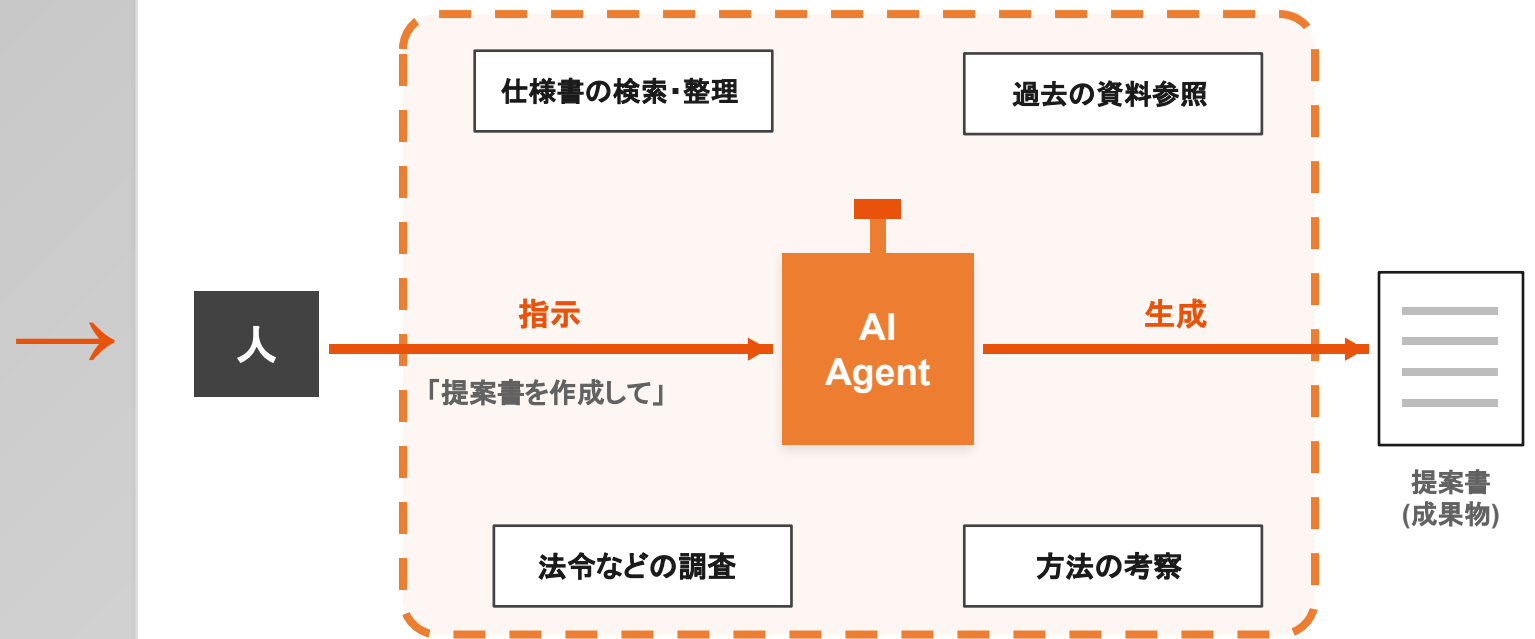
MCPにより「手足」を得たAIは、自ら考えシステムを操作し業務を完結させる。これが真の AIトランスフォーメーション

汎用チャットボット



情報は教えてくれるが、
最終的に手を動かすのは人間

AIエージェント



業務のゴールを伝えるだけで、AIが必要タスクを自律的に実行・完結させる

システム開発の第三の選択肢

AI(Claude Code) × WebAppで実現する、SaaSでもスクラッチ開発でもない第三の選択肢で貴社の DXを実現

SaaS・ローコードツールの悩み

導入は早いけど、自社の特殊な業務に合わない
人数が増えるとアカウント代が高すぎる

結局、エクセル管理



ユーザー課金



スクラッチ開発の悩み

自社の業務にぴったり合うものを一から作ろう！
と見積もったら、初期費用が高い。時間もかかる、
出来上がったが、使いにくい

システム開発の長期化



使いにくい



提案

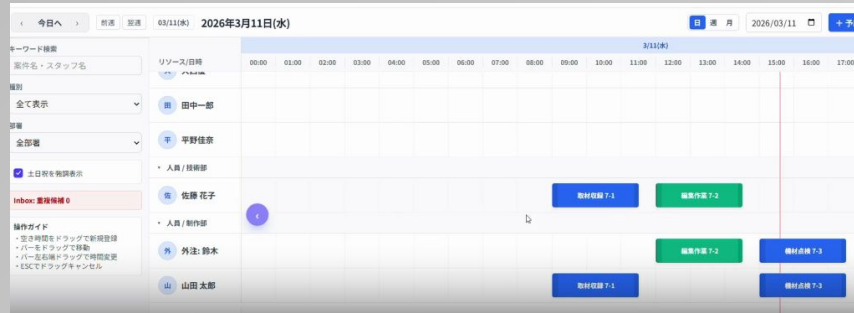
Claude Code × Web APP

で実現する、システム開発
(固定料金、Good UI、)

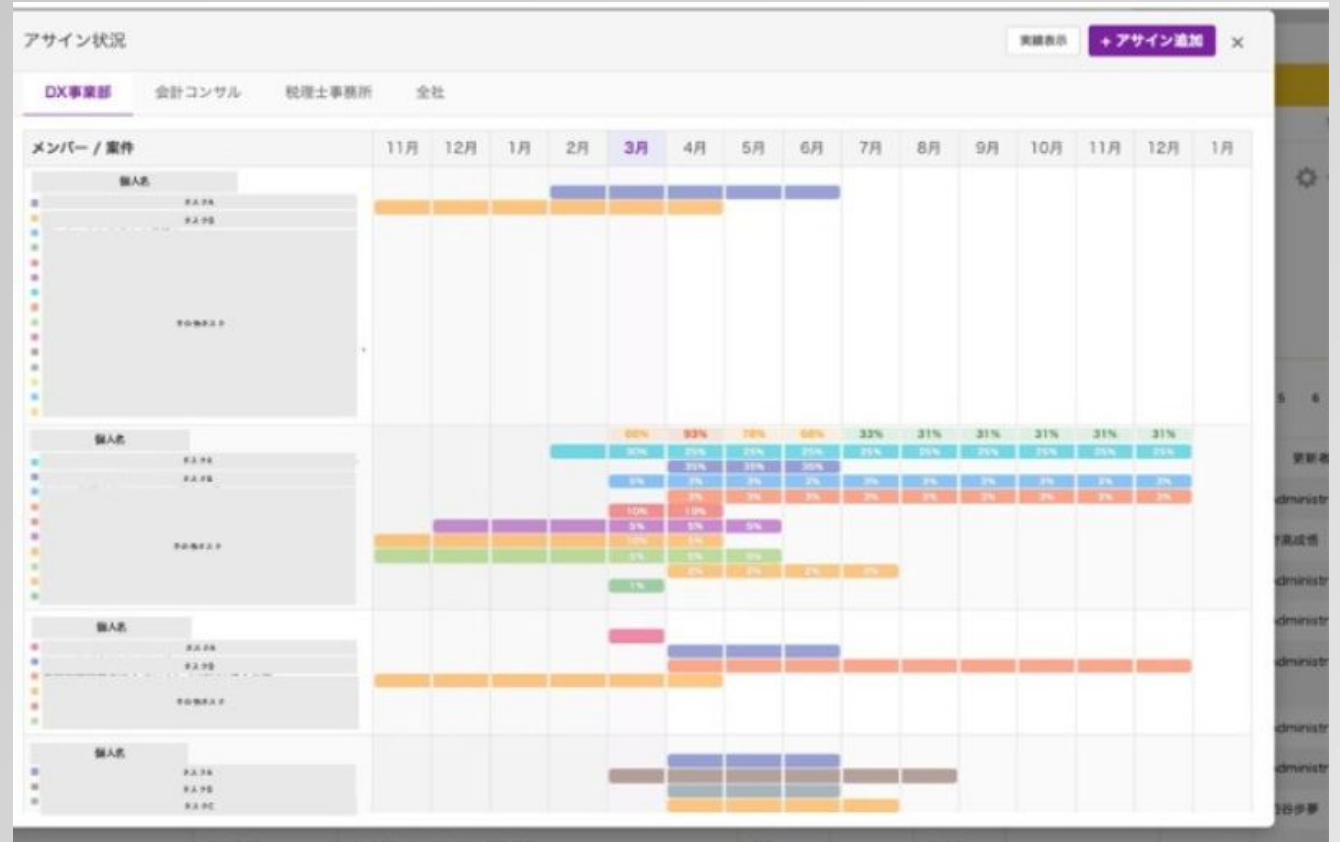
Claude Code × Web APPシリーズ

過去に下記の様なWebAPPを開発・導入済みです。(新規開発もちろん受け付けております)
自社での取り組みは下記記事をご覧ください。(<https://aurant-technologies.com/blog/internal-practice-saas-webapp-part1/>)

打刻・スケジュール・施設予約



プロジェクト・アサイン管理



会計レポート(BI)



Aurantが解決できること

営業から経理・マーケ・経営まで、どの部門のどの課題にどのソリューションが対応するかの一覧

部門 / ソリューション	独自システム 開発	AIエージェント	LINE開発	BI・経営	AI伴走支援	主な施策・成果
営業・CRM	◎	◎	○	◎	◎	商談音声→CRM自動入力、提案書10分生成、トークスクリプト最適化
経理・バックオフィス	◎	◎	—	○	◎	請求・仕訳・入金消込の完全自動化
マーケ・CX	○	◎	◎	◎	◎	LINE OMO、広告自動化、真のROAS可視化
経営管理	○	○	—	◎	◎	CFOLレポート、ダッシュボード、予実管理
人事・採用	○	◎	—	○	◎	面接→評価→入力を自律化
製造・在庫	◎	○	—	○	○	BOM・在庫・購買の統合管理

◎ 最重点領域 ○ 対応可 — 対象外

02

解決策 -

Our Solution【営業・経理・開発】

【営業・CRM】商談音声 → CRM自動入力・提案書 10分生成・トップ営業の技

商談後の作業(メモ転記・入力・提案書作成・次アクション判断)を全て AIが代行。営業は商談に集中でき、トップ営業のトークスクリプトを分析・営業力の強化

することで、組織全体の営業力を強化・最適化します



【提案フェーズ】自動メール送信 (AI-DM)

高品質なメール作成・ターゲティングが可能です。

自社のソリューション情報や、送付先の情報を基にしたメールを送信することができます。(下記は AI生成で実際に送付したデモメールです)



To info, Bcc: 自分 ▼

ご担当者様

と申します。

おかやまIT経営力特別賞のご受賞とIT導入補助金のご活用を拝見し、DXへの積極的な姿勢に共感してご連絡いたしました。

創業60年以上の歴史を持ちながら、医療用サポーター・コルセットの企画から製造まで自社で一貫して手がける御社が、IT経営でも先進的に取り組まれていることは、大変刺激を受けております。

▼ AI時代に求められるのは「データをつなぐ」こと

近年、生成AIの活用が急速に広まる中で、多くの企業様が気づき始めているのが「AIを入れる前に、データがバラバラのままでは使いこなせない」という現実です。

受発注・品質・在庫・売上のデータがExcel・基幹システム・紙に分散していると、AIがあっても参照できる情報が限られ、自動化の効果が半減してしまいます。

弊社では、AIエージェントの導入支援だけでなく、BIツールによる経営可視化や、既存の業務管理ツールの整備・統合を一体で行います。

データが整備・統合されて初めて、生成AIが現場の力になります。

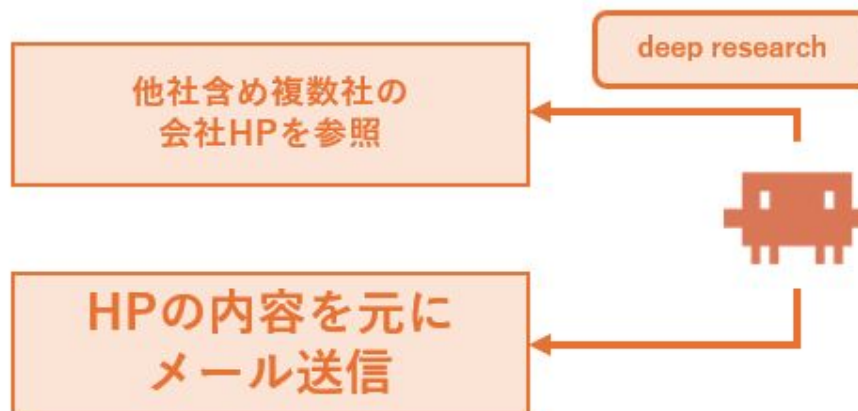
▼ 生成AIで「こんな自動化」ができます

たとえば御社のような製造現場であれば、

品質検査の記録データをもとに、報告書の文章を生成AIが自動ドラフト
受注メールの内容を読み取って、業務管理システムに自動登録
月次の在庫・生産データを集計し、BIツールで経営レポートを自動生成
設計変更の仕様書を自然言語で問い合わせると、該当箇所をAIが即座に返答

こういった自動化が、特別なプログラミングスキルなしに実現できる時代になっています。

クライアントの属性にあつたメールを送信

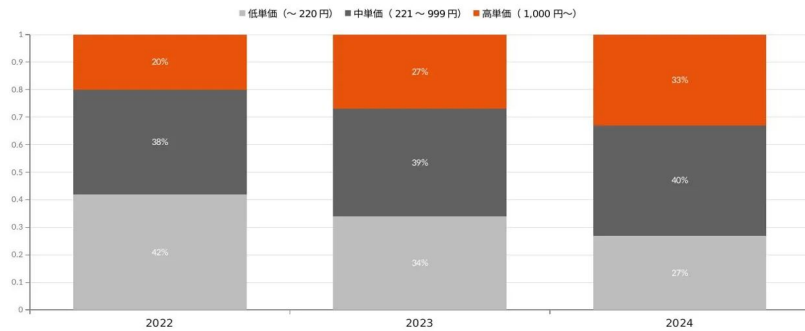


【商談・提案フェーズ】スライドの自動生成 (AI-Slide)

高品質な自社のテンプレートに合わせた資料作成が可能です。(下記は AI生成で実際に作成したデモスライドです)

書籍の価格帯別 売上構成比推移

低単価帯 (220 円以下) のシェアが縮小し、高単価帯 (1,000 円以上) が拡大、単価アップ戦略が奏功



事業ポートフォリオ評価

国内ブックオフが「安定収益源」の位置づけ。プレミアムは成長性は高いがシェア獲得が課題



アプリ会員の店舗購買導線

811 万人のアプリ会員がネット→店舗受取→ついで買いへ段階的に移行し、リアル店舗の売上に寄与

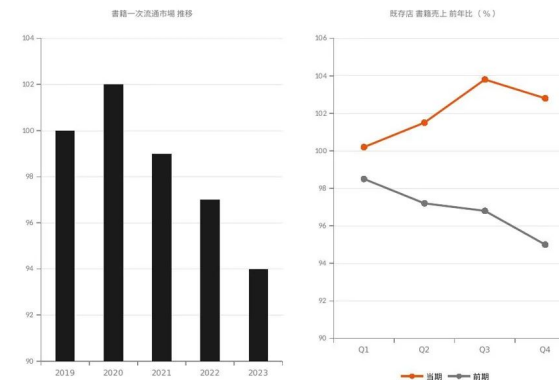


ポイント

ネット→店舗受取→ついで買いの導線は、アプリ会員の 2 割弱を店舗売上の底上げに繋げる仕組みとして機能。公式アプリを起点とする統合顧客体験の典型例。

市場環境と自社の取り組み

一次流通が 5 年で 6% 縮小する中、ブックオフ書籍の売上前年比 102.8% と逆行して成長



- ◎ 一次流通の縮小が続く中、ブックオフの書籍前年比は 2023 年から上昇トレンドに転換。高単価冊作り・デジタル連携の相乗効果が市場と逆方向の成長を実現
- インフレによる新刊書籍の平均単価 +9% 上昇に連動し、中古価格も引き上げ。低単価帯の比率が 42%→27% へ縮小、高単価帯が 20%→33% へ拡大
- △ ソフトメディアもレトロ需要が後押しし、本と合わせて安定推移。次フェーズではデジタル強化が重要
- トレカ・ホビーは相場変動リスクあり、書籍安定収益とのバランス管理が継続課題

【経理・中小企業】free × kintone連携モデル

kintoneで業務を管理しながら、freeへ会計データを自動連携。請求・仕訳・入金消込まで一気通貫でつなぐシステム構成です

中小企業向け



API連携
データ自動同期

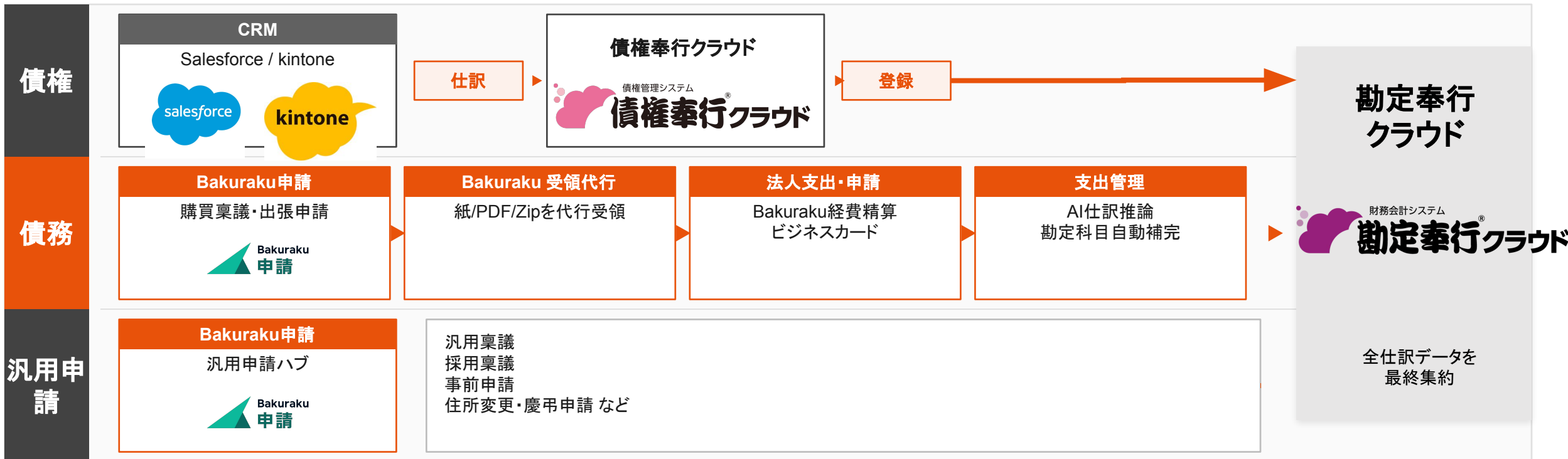


kintoneで業務を柔軟に設計しつつ、freeに会計データが自動集約。中小企業に最適なコストバランスで「手作業ゼロ経理」を実現

【経理・大企業】Salesforce × 勘定奉行 × Bakurakuモデル

CRMの受注データをトリガーに、消込処理。 Bakurakuで申請・経費を処理し、勘定奉行クラウドへ仕訳を自動連携。 AI仕訳推測で科目補完

大企業向け



大企業特有の CRM連携機能を備え。稟議・経費・購買フローを全て Bakurakuで標準化し、AI仕訳推論で勘定奉行クラウドへ自動連携。月次締めめの劇的短縮と統制強化を両立

03

解決策 -

Our Solution【マーケ・CX】

【LINE・マーケ・CX】LINEをハブにしたOMO体験構築

ネイティブアプリ不要でLINEだけで会員証・予約・クーポンが完結。来店データがCRMに自動蓄積され、効果的なマーケティング戦略を実現



会員証・バーコード

順番待ち・呼出

パーソナライズ配信

予約・電子チケット

クーポン配布

ミニアプリ化

Salesforce等各種CRM・MAツールとの連携も可能、LINEとつながる一気通貫CXを実現

【LINE・マーケ・CX】LINEをハブにしたOMO体験構築

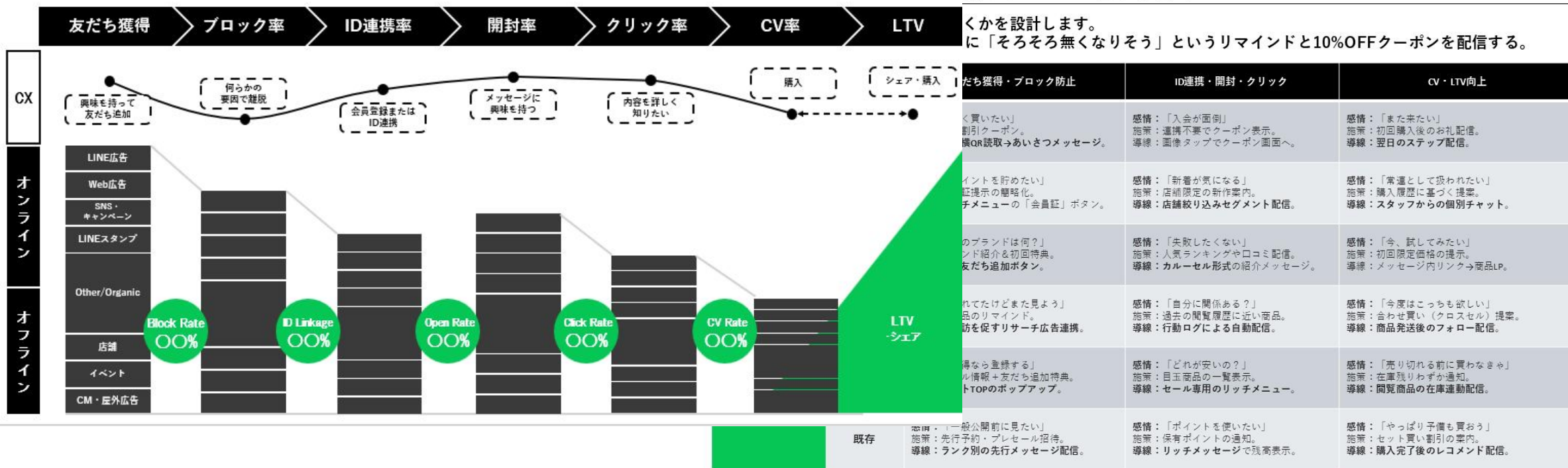
ブログ記事にて、LINE関連のマーケティング戦略構築に関しても発信中です
<https://aurant-technologies.com/blog/line-official-account-crm-ltv-practice-guide/>

LINE CRM戦略の構築

LINE運用でよくある失敗は、「目的が曖昧なまま、なんとなく全員に配信してしまうこと」です。
 まずは、皆様の会社が今「新規獲得」と「既存育成」のどちらに注力すべきフェーズなのかを定義し、追うべき指標（KPI）を整理しましょう

開く「タイミング」を想像する

くかを設計します。
 に「そろそろ無くなりそう」というリマインドと10%OFFクーポンを配信する。

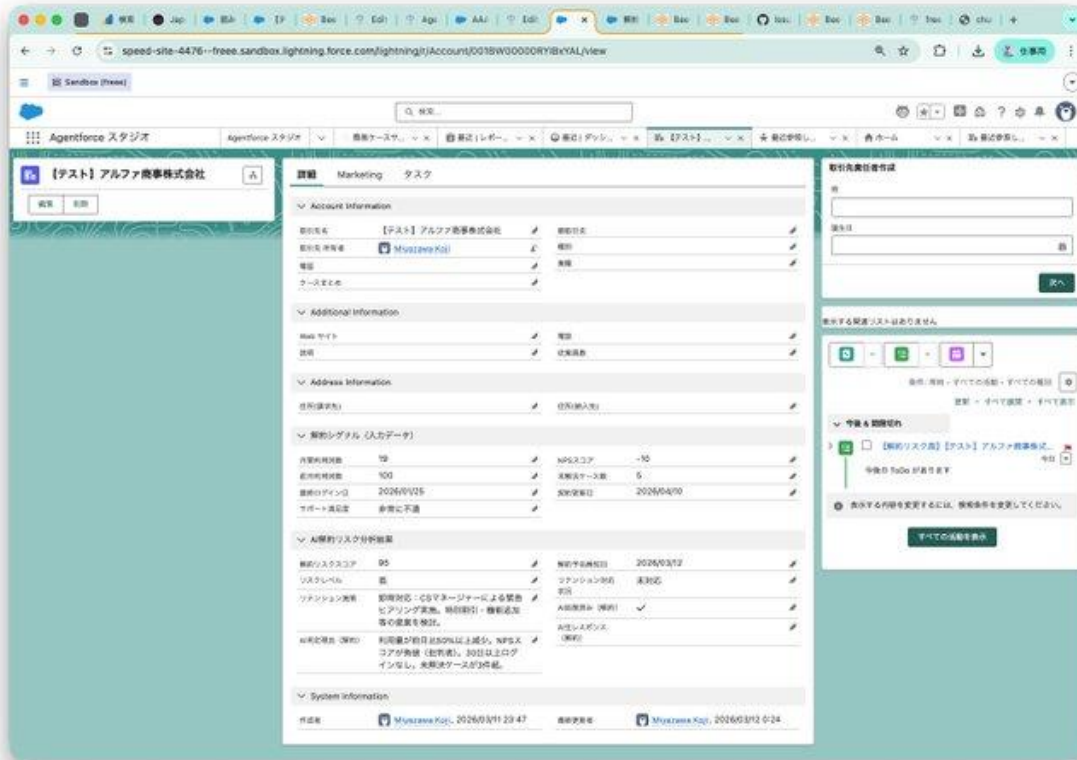


豊富な経験を持ったコンサルタントが貴社の成長をお手伝いいたします

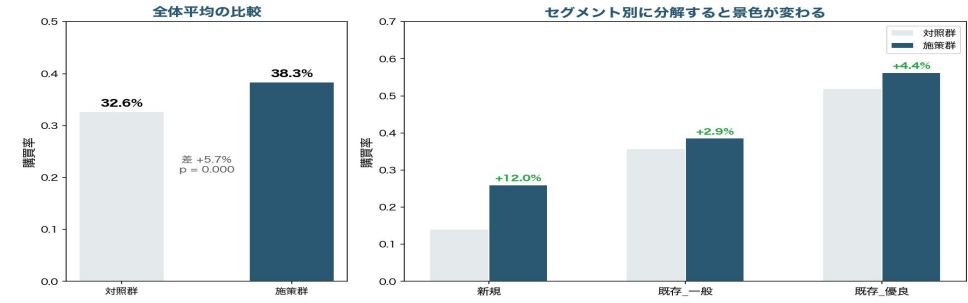
【マーケ・CX】Salesforce運用の高度化

表面的なデータ(全体平均や過去の事象)だけで判断するのではなく、データに隠れた「本当のリスクや予兆」を解像度高く抽出(分析の高度化)し、確実に利益を生むアクションの実行や自動化(マーケティングの高度化)へと繋げるアプローチが AIの力によって可能になります

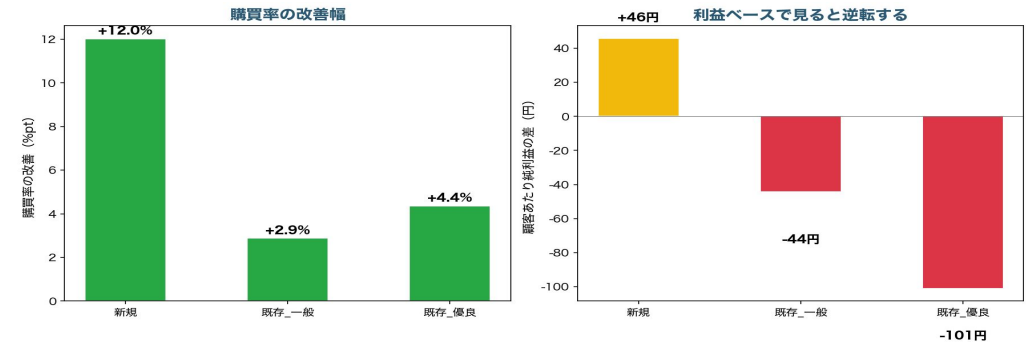
Salesforceの設定・開発



キャンペーン効果分析



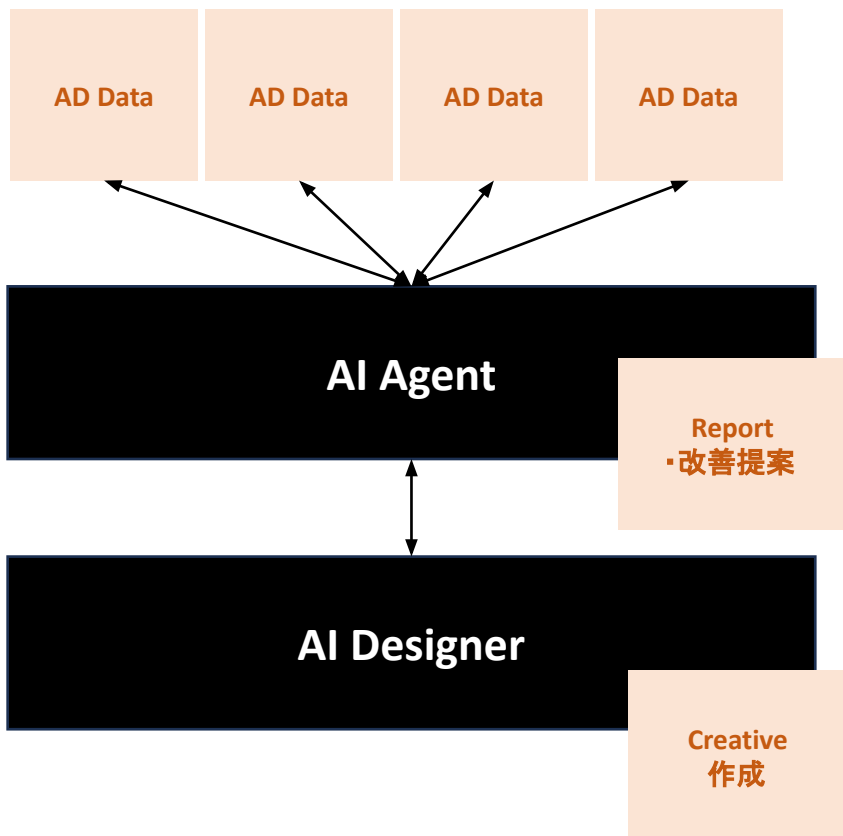
クーポン効果分析



Salesforceの高度活用によるマーケティング施策への移行や、Salesforceの開発支援まで

【Web・マーケ】Web広告の完全自動化(AD-AI)

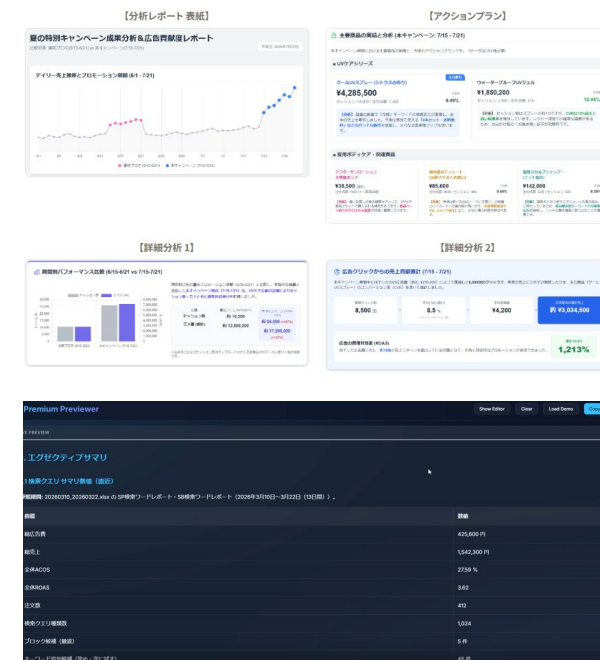
Google/Meta広告の予算調整・A/BテストをAIが自動実行。表面的なCPAではなくLTV・商談化率まで含めた真のROASを可視化



広告用 クリエイティブの作成



レポート自動化



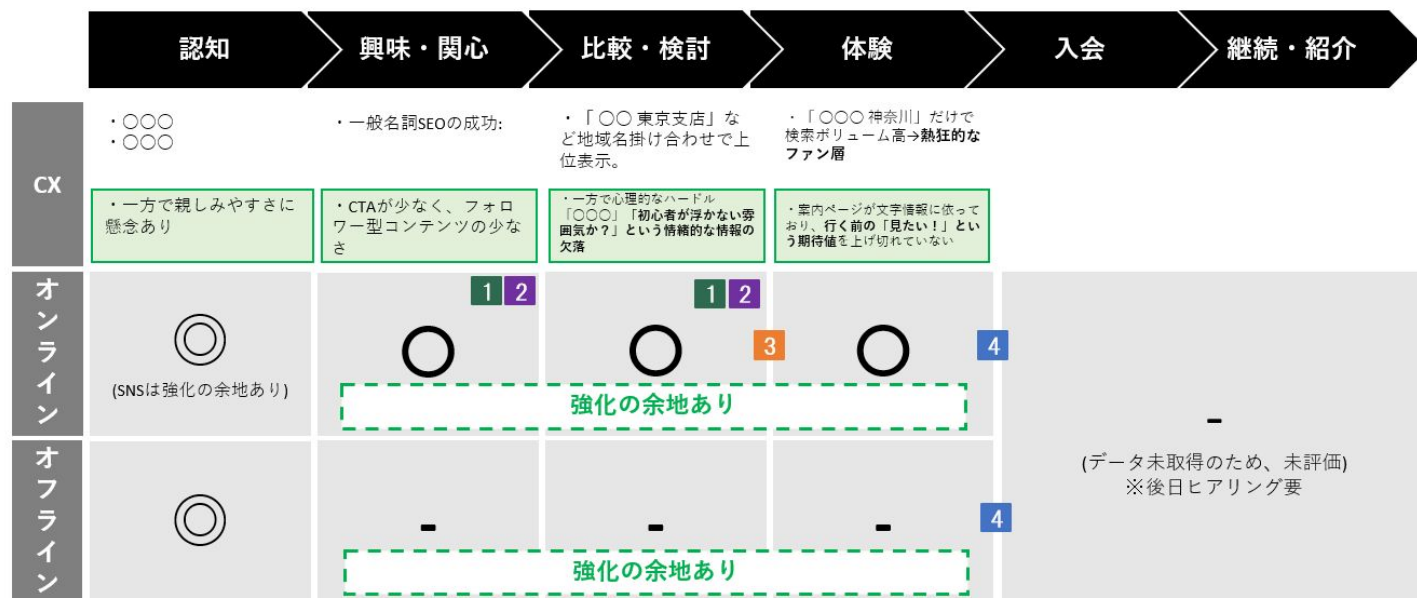
従来の5倍のスピードで判断を行い、4倍のスピードで広告を作成 = 20倍の速度で運用可能

【Web・マーケ】SEOの完全自動化(SEO-AI)

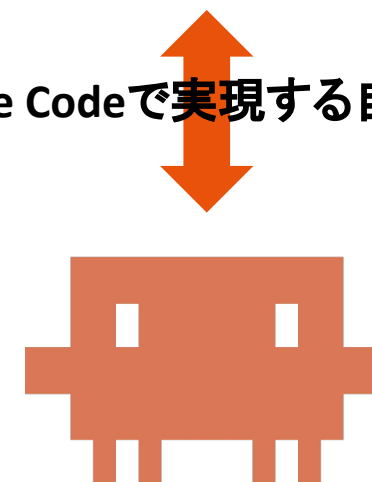
SEOの記事作成をAIで自動化、HP全体の戦略提案～リブランディングまでお手伝いいたします
 ブログ記事にて、マーケティング戦略構築に関する発信中です
マーケティング戦略立案の裏側

カスタマージャーニー別診断

現状のWeb流入は他社ドメインと比較しても多く、重要なキーワードでランキングを取れているので、流入に関しては特に改善の余地なし
 一方で、集客後の「体験・入会」までの導線・フォローづくりの体験設計に関しては改善の余地あり



Claude Codeで実現する自動化



3か月で総表示回数 100万回突破、クリック数増加率：1,000%超のAI時代のSEO対策

04

解決策 -

Our Solution【経営・財務】

【経営】CFOLレポートの自動生成

free APIからデータを取得し、AIがキャッシュフロー分析・回収リスク評価・推奨アクションを自動出力（Aurant自社でも稼働中）

データフロー

01 free API

残高・取引・売掛データを
リアルタイム取得

02 AI分析エンジン

キャッシュフロー分析
回収リスク評価
シナリオ・ストレステスト

03 CFOLレポート

PDF/Slack/Looker
で自動配信

CFOLレポート自動出力の中身(例)

① キャッシュ必要額レンジ

当月 / 3ヶ月 / 6ヶ月先の
必要キャッシュを3ケース試算

② 取引先別 回収リスク評価

回収遅延確率スコア
売掛集中度の警告

③ ストレステスト

売上 ▲30%/回収50%遅延の
悲観シナリオでの耐性分析

④ 推奨アクション

「X社に来週連絡」等
タスク管理ツールへ自動登録

05

事例-

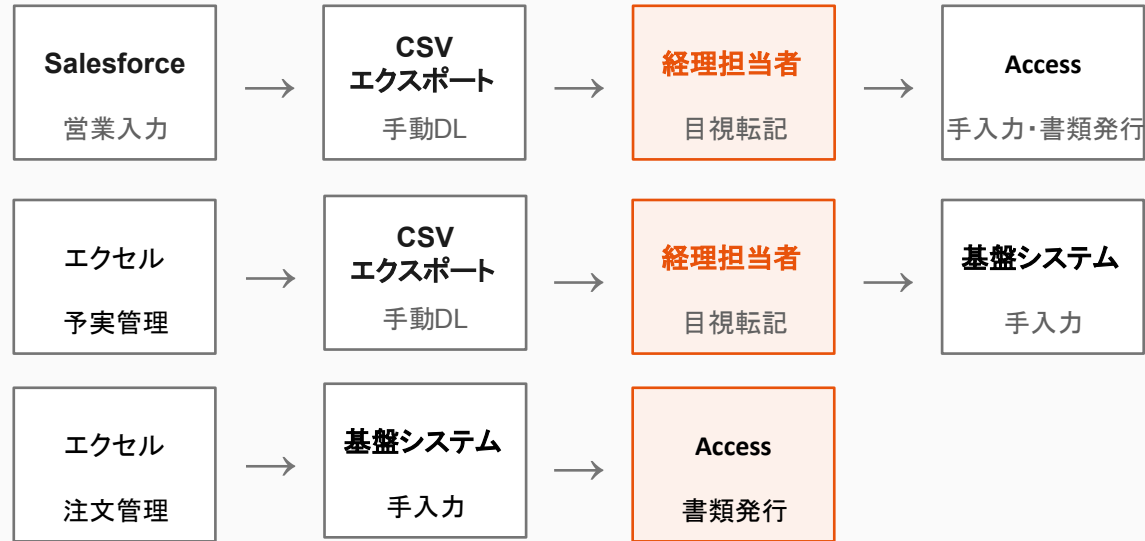
Case Study

導入事例 – 大手広告代理店 A様

Access+基盤システムを WebAPP×各種 SaaSツールとの連携によって、リプレイス。
既存ランニングコストを年間 2,000万円強削減

業種: 広告代理店 / 既存システムリプレイス

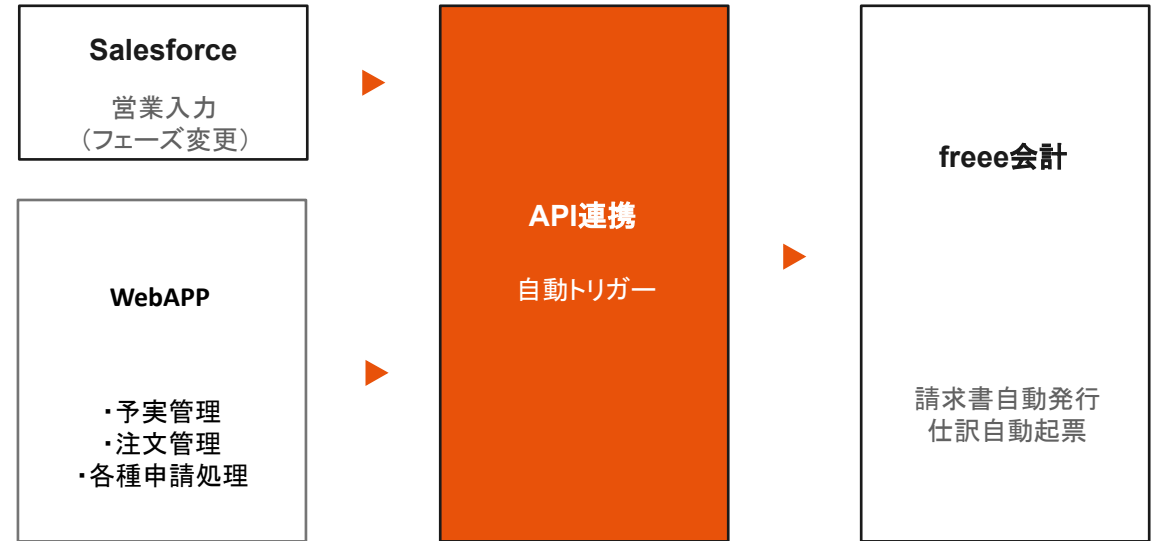
BEFORE 基盤システムのスピードが遅く、経営に支障



問題点

- ・月末月初に経理担当者が深夜残業(請求書の処理や、予実のズレ)
- ・ヒューマンエラーによる請求漏れ頻発
- ・既存システムの高いランニングコスト
- ・会計基準や、各種法整備に伴う既存システムへの改修
- ・料率の改定など、経営で取りたいアクションが取れなかった(システム改修に時間がかかる)

AFTER API連携による完全自動化



実現したこと

- ・営業が案件を「請求フェーズ」に進めると裏で API起動
- ・freee上で請求書+仕訳伝票が同時生成
- ・転記ミスが構造的に発生し得ない
- ・WebAPP基盤に移し替えることで、入力箇所の UI・UXの向上。入力のしやすさにより、営業側や経理側の業務効率もアップ

導入事例 - 医療機器販売 M様

Salesforce × freee のAPI完全連携により、請求処理を 0秒化。人員据え置きで取引 2倍に対応する体制を実現

業種: 医療機器販売 / 課題: 請求業務の月末残業

BEFORE 手作業による転記ボトルネック



問題点

- ・月末月初に経理担当者が深夜残業
- ・ヒューマンエラーによる請求漏れ頻発
- ・取引増で人員増やさないと対応不可

AFTER API連携による完全自動化



実現したこと

- ・営業が案件を「請求フェーズ」に進めると裏で API起動
- ・freee上で請求書+仕訳伝票が同時生成
- ・転記ミスが構造的に発生し得ない

請求書1件あたりの処理時間

0秒

従来: 数分の手作業

転記ミス・請求漏れ

ゼロ

物理的に発生し得ない構造

取引件数の増加

2倍

導入後のスケーラビリティ

経理人員の増員

0人

据え置きで事業拡大に対応

導入事例 - 製造業 A様(BOM・在庫・購買の統合)

紙・Excel・個人PCに散らばっていた情報を kintoneに集約。受注と同時に部材引き当てと購買アラートが自動で走る

業種: 製造業 / 課題: 情報散在による属人化と赤字案件

BEFORE 情報が社内に散在



- ・受注時に部材の空き状況が分からず納期遅延
- ・材料費高騰が見積に反映されず赤字案件
- ・ベテラン社員の頭の中に知見が閉じ込められている

AFTER kintone中心に統合



- ・受注時に在庫引き当て・購買アラートが自動実行
- ・材料費変動が即座に見積に反映される
- ・ベテランの知見が全社員に共有される構造へ

赤字案件の撲滅 / 納期遅延ゼロ化 / 属人化からの脱却

導入事例 - スクール事業 B様(顧客誘導の自動化)

Web/LINE経由の申込から日程調整・入金・ Slack招待まで全自動。担当者は面談だけに集中できる体制に

業種:教育・スクール事業 / 課題:顧客対応工数

顧客誘導の自動化フロー



BEFORE

担当者工数

申込1件あたり30~60分(メール返信・日程調整・請求書作成・Chatwork招待)

AFTER

担当者工数

申込1件あたり 0分。
担当者は面談そのものだけに集中できる

導入事例 - 外国人派遣 J様(在留カードOCR自動読取)

スマホで撮影 → OCR → kintone自動入力 → 期限切れアラートまで完全自動化。コンプライアンス強化と事務工数削減を両立

業種: 外国人派遣 / 課題: 在留資格管理の工数と期限切れリスク

在留カード登録の自動化フロー



1件あたりの登録時間

数分 → 数十秒

撮影から登録完了まで

入力ミス・誤字

1/10

OCR高精度で物理的に排除

期限切れ雇用リスク

完全排除

自動アラートで先回り対応

06

料金-

Price menu

サービス料金

サービス名	料金・契約形態	項目／フェーズ	具体的な支援・実施内容
AI伴走支援	月額 150,000円(税抜) ※単月契約 ※出張費・交通費等は別途実費	<ul style="list-style-type: none"> ・月2回の定例MTG ・月3時間のレクチャー ・AIツールの導入支援 ・現場での活用支援 	<p>【定例MTG】 業務の現状分析と課題抽出/ ツール利用状況の分析と戦略のPDCA 【レクチャー】 対象者(開発/ビジネス部門)に合わせたカリキュラム/ 最新LLM動向の共有 【導入支援】 Claude Code等のアカウント開設・環境構築/ 安全な社内AI利用ルールの策定 【活用支援】 実務直結のユースケース構築/ 講師と共にAIを動かす実践的なペアプログラミング</p> <p>【事例】 <ul style="list-style-type: none"> ・社内エンジニアへのClaude Code導入・定着支援 ・非エンジニア部門へのChatGPT活用・業務効率化レクチャ </p>
SaaS・AI 組み込み開発	要見積り ※プロジェクト型契約 ※要件規模や複雑さに 応じて個別算出	<ol style="list-style-type: none"> 1. 要件定義 2. プロトタイプ作成 3. 本開発 4. QA(テスト) 5. リリース 	<p>【要件定義】 ビジネス要求のシステム要件化/ UI・UX設計 / 最適なAIモデルと技術の選定 【プロトタイプ】 コア機能(MVP)の短期間開発/ 実機を用いた仮説検証と方向性の早期修正 【本開発】 拡張性を考慮したアジャイル開発/ RAGやAPI連携によるセキュアなAI機能の組み込み 【QA】 単体・結合テストの実施/ ハルシネーション対策などAI特有のエッジケース検証 【リリース】 クラウド環境へのデプロイ/ マニュアル作成およびスムーズな運用移行サポート</p> <p>【事例】 <ul style="list-style-type: none"> ・kintone開発 / freee会計、freee人事労務導入支援 ・kintone ⇄ freee 連携システムの構築 ・Salesforce開発 / Salesforce ⇄ freee 連携 ・勤定奉行、バクラク連携システムの構築 </p>
コストダウン支 援 (AI・SaaS活用)	要見積り ※プロジェクト型契約 ※削減効果に応じた 成果報酬プラン等もご相談可能	<ul style="list-style-type: none"> ・現状分析とコスト可視化 ・削減プランの策定 ・システム移行・自動化実装 ・効果測定と継続改善 	<p>【現状分析】 既存システム維持費や外部委託費、非効率な業務プロセスを洗い出し、削減余地を特定 【プラン策定】 高額なレガシーシステムから安価なSaaSへの移行、AIを用いた業務代替プランの提案 【実装・自動化】 代替SaaSへの移行作業、手作業をAI・RPAで自動化するワークフローの構築と内製化支援 【効果測定】 削減コスト(時間・費用)の明確な算出と、リソース最適化に向けた継続的なモニタリング</p> <p>【事例】 <ul style="list-style-type: none"> ・高額なフルスクラッチ基幹システムから安価なSaaS(kintone等)へのリプレイス(1,000万円規模) ・高額なフルスクラッチ基幹システムから安価なSaaS(kintone等)へのリプレイス(2,000万円規模) ・外部委託していたデータ入力・分析業務のAI自動化による外注費80%削減 </p>

AIの真価をお届けします