

まずはここから

はじめて学ぶ ロード開発



目次

- ✓ はじめに
- ✓ ローコード開発の定義
- ✓ 従来のアプリ開発との違い
- ✓ ローコード開発のメリット
- ✓ ローコード開発の基本要素
- ✓ 実現可能な業務用アプリケーション
- ✓ チーム編成と役割
- ✓ プラットフォームの選び方
- ✓ まとめ

はじめに

ローコード開発の重要性

ローコード開発は、プログラミングスキルが初級者でも迅速にアプリケーションを開発・導入できる手法であり、企業のデジタルトランスフォーメーションを加速する役割を果たしています。また、開発コストの削減や業務効率の向上にも寄与します。

本ホワイトペーパーの目的

本ホワイトペーパーでは、ローコード開発の基本的な概念や実現可能なアプリケーション、チーム編成や役割について解説し、初めてローコード開発に取り組む方々にとって有益な情報を提供することを目的としています。また、自社のローコード開発サービスの紹介も行います。

これらの情報をもとに、各セクションのテキストや図・イラストを作成していくことで、ホワイトペーパーがより具体的かつ視覚的に理解しやすくなるでしょう。



ローコード開発の定義

ローコード開発の定義

ローコード開発 (Low-Code development) とは、最小限の必要なソースコードのみを記述し、そのほかの多くの部分を「GUI (Graphical User Interface/グラフィカルユーザインターフェース)」とよばれる、視覚的に理解しやすく直感的に操作できる画面を用いて開発することです。

ローコード開発では、あらかじめ用意された機能単位（コンポーネント）を組み合わせることで開発を行います。ボタンを押すだけで一定の処理が行われるような機能が用意されており、それらをドラッグアンドドロップして組み合わせることで、アプリケーションを開発できます。

ノーコード開発との違い

ノーコード開発と似た開発方法として「ノーコード開発 (No-Code development)」があります。ノーコード開発とは、プログラミングを全く行わずに開発する手法です。ノーコード開発と同様、GUIを活用して、用意された機能単位をドラッグアンドドロップなどの直感的な操作で組み合わせることで開発を行います。まったくプログラミングを行わない分、開発も容易ですが、ローコード開発に比べると自由度は下がります。

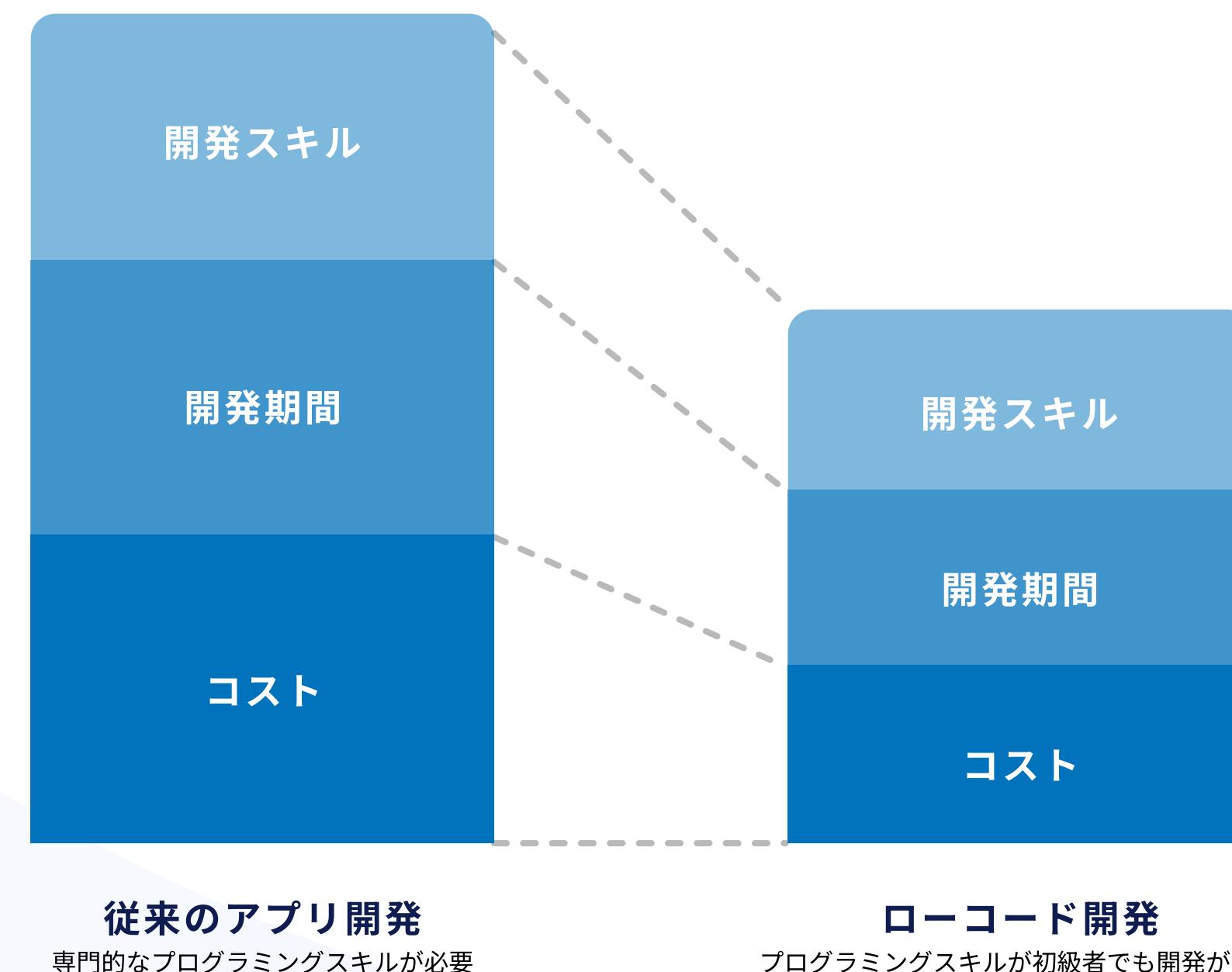


従来のアプリ開発との違い

従来のアプリ開発では、専門的なプログラミングスキルが必要であり、開発期間が長く、高いコストがかかることが一般的でした。一方、ローコード開発では、プログラミングの期間を大幅に圧縮できます。何ヵ月もかかっていたプログラミングを数時間から数日にまで圧縮でき、システム開発にかかる時間を大幅に短縮できます。プログラミングスキルが初級者でも開発が可能であり、迅速なアプリ開発と導入が実現します。

また、プログラミングをほとんど必要としないことから、ユーザー部門の担当者が開発を担えるため、プログラミングの前工程のヒアリングや要件定義も短縮できます。

運用を開始してからも、用意された仕様の範囲内であれば変更が可能なため、現場で使いやすく改善することも可能です。



ローコード開発のメリット

ローコード開発のメリット

ローコード開発の主なメリットは開発速度の向上、開発コストの削減、ノンプログラマーの参加可能性、アプリケーションの柔軟性、外部システム連携の向上です。これらにより、企業はデジタル化を加速し、業務効率を向上させることができます。



開発速度の向上

通常の開発手法とは違い、開発中にコードを記載する機会は最小限に抑えられるため、作業工数が自然と少なくなり開発速度の向上につながります。



開発コストの削減

システム開発の費用は、基本的に工数に比例します。ローコード開発を活用すれば開発期間を短縮できるため、開発費用を削減できます。利用開始後の改修も社内で行えるため、改修にかかるコストも抑えられます。



ノンプログラマーの可能

あらかじめ用意された機能単位を組み合わせることで、開発のほとんどを済ませることができるため、プログラミング初心者でも開発が行えます。



アプリケーションの柔軟性

ユーザーの反応に合わせてより良く改修したり、業務フローの変更に合わせて改修するといったことを現場のユーザーの手でスピーディーに行えます。



外部システム連携の向上

自社すでに利用しているシステムやアプリケーションなどと連携できると利便性も高いです。そのため、ローコード開発ツールの拡張性の高いものを選ぶことが大切です。

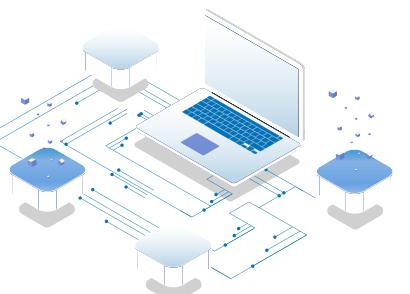
ローコード開発の基本要素

ドラッグ・アンド・ドロップ



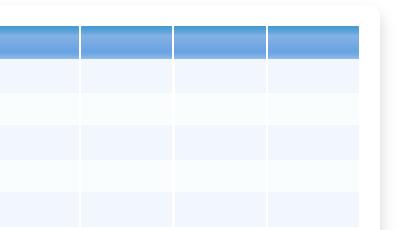
ドラッグ・アンド・ドロップ機能は、ユーザーが画面上の要素を選択し、マウスで移動させて配置することができる操作方法です。これにより、簡単に画面のレイアウトやデザインを調整することができます。

ビジュアルプログラミング



ビジュアルプログラミングは、プログラムのロジックを視覚的に表現し、ブロックやアイコンなどの要素を接続することでプログラムを構築する方法です。これにより、コードを書かずにアプリケーションの機能を実装することができます。

データベースの管理

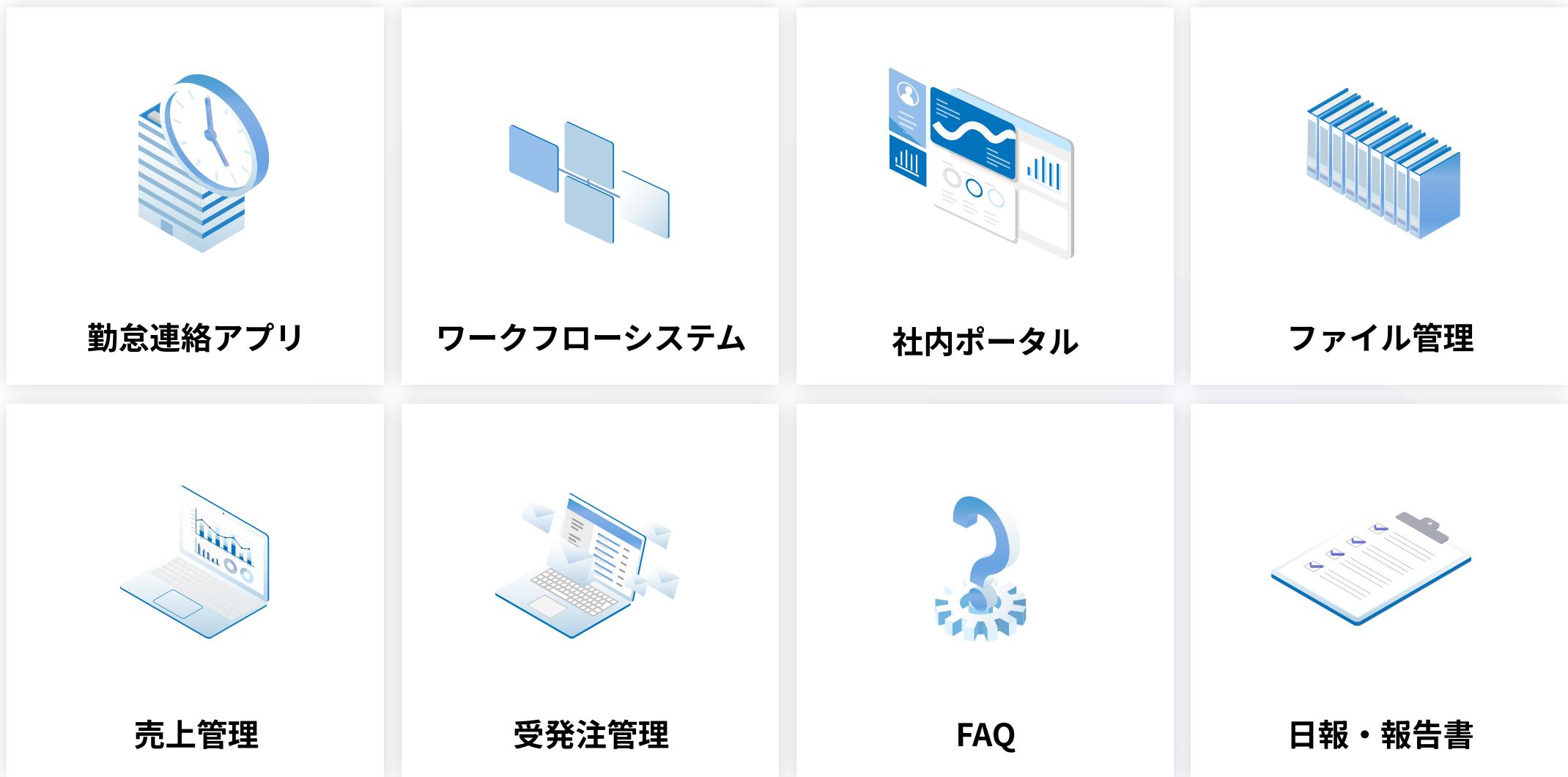


ローコード開発プラットフォームでは、データベースの設計や管理が視覚的に行えます。テーブルやカラムの作成、データの追加・更新・削除などの操作が簡単に行えるため、データ管理が容易になります。

実現可能な業務用アプリケーション

ローコード開発により、アイデアから実現への道筋が一気に近づきます。開発者ではない人々がビジネスプロセスを直接デジタル化できる、自分たちの問題を自分たちで解決する新しい可能性が広がっています。以降のページでは、ローコード開発を通じて制作可能なアプリケーションの具体的な例を紹介します。それぞれのアプリケーションはどのような機能を持つのか、どのような問題を解決できるのか、具体的なケースを通して見てきましょう。

代表的なアプリケーション



勤怠管理アプリの5つのポイント

- 1 **ユーザビリティ:** 社員の使いやすさは日々入力することを考えれば最優先事項であるべきです。直感的で簡単な操作性を確保し、複雑なプロセスを避けましょう。
- 2 **セキュリティ:** 個人情報を扱うため、データのセキュリティとプライバシーを確保することは言わずもがな必要です。パスワード保護、エンドツーエンド暗号化など、適切なセキュリティ機能を開発時に実装しましょう。
- 3 **柔軟性:** 時短勤務や時差出勤などユーザーの勤怠スケジュールや稼働時間を様々な要件に対応できるようにする必要があります。
- 4 **互換性:** アプリはPC / スマートフォンなど様々なデバイスやオペレーティングシステムで動作するように設計することで、可能なら多くのユーザーがアプリを利用できるようになります。
- 5 **連携機能:** 他のシステムやツールとの統合を可能にすることが重要です。たとえば、会社の既存の人事管理システムやメールシステムとの連携などを考えると良いでしょう。

勤怠連絡アプリをローコード開発で作るときに必要な項目

- ✓ ログイン/ログアウト機能: ユーザーが自身のアカウントでアプリにログインし、安全にログアウトできる機能が必要です。
- ✓ ユーザープロファイル: ユーザーの基本情報（名前、連絡先、所属部署等）を管理できる項目が必要です。
- ✓ 出勤/退勤時間記録: ユーザーが出勤時間と退勤時間を簡単に記録できる機能が必要です。
- ✓ 休暇・休業申請: ユーザーが休暇や休業を申請し、そのステータスを追跡できる機能が必要です。
- ✓ レポート/統計: 勤怠状況のレポートを作成し、統計情報を提供する機能が必要です。
- ✓ 管理者機能: 管理者が全ユーザーの勤怠情報を一覧・編集できる機能が必要です。
- ✓ 通知機能: 重要な情報やリマインダーをユーザーに通知する機能が必要です。
- ✓ データエクスポート: 必要に応じて勤怠データをエクスポート（CSVなど）できる機能が必要です。

不足しそうなら勤怠連絡アプリの開発手順

- ① 要件定義: 自社の勤怠管理アプリが果たすべき役割と機能を明確にします。具体的には、ユーザーが何をスムーズに達成できればいいのか、どのような情報が必要なのか、どのような操作が必要なのかなどを決定します。
- ② 設計: 要件定義に基づいてアプリの設計をします。ここでは、アプリのインターフェース、各画面のレイアウト、動作のフロー、データベースの設計などを詳細に考えます。
- ③ 開発: ノンコーディングでアプリを開発し、運用時に誰でも修正できるようマニュアルまで整備しておくのがポイントです。
- ④ テストと改善: 完成したアプリをテストし、問題点や改善すべき点を見つけ出します。ユーザビリティテストやセキュリティテストなどを実施し、問題を修正します。そして、ユーザーフィードバックに基づいてアプリを改善します。

ワークフローシステム開発の5つのポイント

- ① **ワークフローの定義**: ワークフローは、特定のタスクがどのように進行し、どのように完了するかを定義するものです。各ステップ、そのステップの役割、およびそのステップ間の関係を明確に理解し、定義する必要があります。
- ② **ユーザーと権限**: ワークフロー内で誰が何をするべきかを明確にするため、各ユーザーの役割と権限を定義します。これには、タスクの割り当て、編集、承認、およびその他の操作が含まれます。
- ③ **エラー処理と例外管理**: エラーや予期しない状況が発生した場合の対応策を定義します。これには、エラーメッセージの表示、再試行オプションの提供、または問題をエスカレートするための手順などが含まれます。
- ④ **データ管理と保管**: システム内でどのようにデータを収集、保存、取り扱い、保護するかを定義します。これは、データの入力、検証、更新、および削除に関するガイドラインを含むことができます。
- ⑤ **システムとの統合**: ワークフローシステムは通常、他のシステムやアプリケーションと連携する必要があります。そのため、どのシステムと統合するか、そしてそれらのシステムとどう

ワークフローシステムをローコード開発で作るときに必要な項目

- ✓ ワークフローの定義
- ✓ ユーザーロールと権限
- ✓ タスクとアクションの定義
- ✓ 通知とアラートの設定
- ✓ エラー処理と例外管理
- ✓ データ管理と保管
- ✓ システムとの統合
- ✓ レポートと分析ツール

ワークフローシステムの開発手順

①現状の調査分析

社内の各ドキュメントがどのようなプロセスを経て決裁完了に至っているのか、まずは現状の業務フローを明確にします。提出率や決裁までの時間、ヒューマンエラーの状況など現場にヒアリングし、問題点を洗い出してください。

②改善策の検討

作業効率や内部統制の観点から、工数や経路の改善を図ります。必要のないフローは省略し、コンプライアンス要件を満たしていかなければ承認者を増やすなど対応してください。
また、閲覧権限や編集権限については、ワークフローのプロセスごとや、文書ごとはもちろん、文書内の情報の種類ごとの権限についてもしっかりと検討してください。

③システムの担当者を決める

実装後もフローの見直しやフォーマットの変更が発生します。適切に運用するために、メンテナンス担当者を決めておくと安心です。全社に関わる部分は総務、部門に依る範囲は部門ごとに定めるとよいでしょう。

社内ポータルの5つのポイント

- ① **ユーザーニーズの理解**: 社員の要望や社内の問題点を状況把握し、それに基づいた社内ポータルの要件をまとめることが重要です。実際に使用想定する社員からのフィードバックを常に募り、反映させができる体制にしていきましょう。
- ② **直感的なユーザビリティ**: メンバーが迅速に情報にアクセスし、操作が容易で理解しやすいユーザーインターフェースを設計することが求められます。使いやすさと情報量の多さは社員の使用率と直結します。
- ③ **高度なセキュリティ**: 社内情報の保護は極めて重要です。データ暗号化、アクセス制御、不正アクセス検知などのセキュリティ対策を厳格に講じる必要があります。
- ④ **スケーラビリティの確保**: 企業の成長や変化に柔軟に拡充できる設計が重要です。将来的な機能追加を容易にする設計を心掛けましょう。
- ⑤ **他システムとの統合性**: 社内で他に使用しているワークフローシステムやサービスとのスムーズな連携を可能にし、情報の一元化と作業効率の向上を図ることが必要です。APIの連携などが必要であれば事前に要件として入れておく必要があります。

社内ポータルをローコード開発で作るときに必要な項目

- ✓ 組織図と従業員ディレクトリ: 従業員の役職、連絡先情報、部門などを一目で把握できるようにすることが重要です。
- ✓ ワークフロー自動化機能: ルーチン作業を自動化するための機能。これにより、効率を向上させ、ユーザーの作業負担を軽減できます。
- ✓ 学習管理システム (LMS) : 従業員が必要なスキルと知識を維持・向上させるための教育・訓練リソースへのアクセスを提供します。
- ✓ 意見箱やフィードバックツール: 社員からのフィードバックや提案を募り、それに応じて改善を図るためのシステムです。
- ✓ 社内SNS: 社員間のコミュニケーションを促進し、コラボレーションを強化するためのプラットフォームです。
- ✓ パフォーマンスマニタリングツール: 従業員のパフォーマンスを追跡し、管理するためのツールです。

社内ポータルの開発手順

①現状の調査分析

社内の各ドキュメントがどのようなプロセスを経て決裁完了に至っているのか、まずは現状の業務フローを明確にします。提出率や決裁までの時間、ヒューマンエラーの状況など現場にヒアリングし、問題点を洗い出してください。

②改善策の検討

作業効率や内部統制の観点から、工数や経路の改善を図ります。必要のないフローは省略し、コンプライアンス要件を満たしていかなければ承認者を増やすなど対応してください。
また、閲覧権限や編集権限については、ワークフローのプロセスごとや、文書ごとはもちろん、文書内の情報の種類ごとの権限についてもしっかりと検討してください。

③システムの担当者を決める

実装後もフローの見直しやフォーマットの変更が発生します。適切に運用するために、メンテナンス担当者を決めておくと安心です。全社に関わる部分は総務、部門に依る範囲は部門ごとに定めるとよいでしょう。

ファイル管理システム開発の5つのポイント

- ① セキュリティ：データの保護とアクセス制御は最優先事項です。ユーザーの権限管理やファイルの暗号化、不正アクセス防止のために必要な項目です。
- ② ユーザー増による拡充の意識：将来的にユーザー数やファイル数が増えたときに対応できるよう、予め設計しておくことが重要です。
- ③ ファイル検索：ユーザーが容易に目的のファイルを見つけるように、検索機能を実装します。
- ④ バージョン管理：ファイルの変更履歴を管理し、必要に応じて以前のバージョンに戻せるようにします。
- ⑤ バックアップとリカバリ：データを定期的にバックアップし、データが失われた場合でも迅速に復旧できるようにします。

ファイル管理システム開発をローコード開発で作るときに必要な項目

- ✓ ユーザー管理：ユーザーの登録、ログイン、権限設定などを行います。これは、誰がどのファイルにアクセスできるかを制御するために必要な項目です。
- ✓ ファイル検索：ユーザーが容易に目的のファイルを見つけるように、検索機能を実装しましょう。
- ✓ バージョン管理：ファイルの変更履歴を管理し、必要に応じて以前のバージョンに戻せるようにできる機能は検討の余地があります。
- ✓ データ暗号化：ファイルの内容が不正に盗まれることを防ぐため、データは暗号化できるようにしておきましょう。
- ✓ バックアップとリカバリ：データを定期的にバックアップし、データが失われた場合でも迅速に復旧できるようにしましょう。
- ✓ 監査ログ：ユーザーの行動を追跡し、必要に応じてログを検査できるようにするため、システムには監査ログ機能が必要です。

ファイル管理システムの開発手順

- ①要件定義：開発するファイル管理システムが何を解決するべきかを詳細に定義します。例えば、社内のメンバーが必要なドキュメントを迅速に見つけられるようにしたい、複数のバージョンを管理したい、特定のユーザーのみが特定のファイルにアクセスできるようにしたい等、具体的な要件をリストアップします。
- ②システム設計：要件に基づき、システムのアーキテクチャとデータベースを設計します。例えば、ドキュメントのメタデータ（作者、作成日、キーワード等）をどのように格納し、検索するか、どのようにバージョンを管理するか等を決定します。また、ユーザーが使いやすいUIを設計します。
- ③開発と実装：設計に基づいてソフトウェアを開発します。ここでは、ドキュメントのアップロード、検索、ダウンロードの機能や、ユーザーの権限管理、バージョン管理の機能等を実装します。
- ④テストとデプロイ：開発したシステムをテストします。具体的には、全ての機能が正しく動作するかを確認し、データが正しく保護されているかを確認します。問題がなければ、システムを本番環境にデプロイし、ユーザーに利用してもらいます。この段階では、ユーザーからのフィードバックを受け取り、必要に応じてシステムをアップデートします。また、適切に運用するために、メンテナンス担当者を決めておくと安心です。全社に関わる部分は総務、部門に依る範囲は部門ごとに定めるとよいでしょう。

売上管理システムの5つのポイント

- ① 要件定義：システムの要件を正確に定義することが重要です。例えば、どのようなレポートが必要なのか、どのような機能が必要なのか（例：顧客管理、在庫管理、予測モデルなど）を詳細にリストアップすることが重要です。
- ② セキュリティ対策：売上管理システムは企業の重要な情報を含むため、セキュリティは重要な考慮事項です。データ暗号化、アクセス制御、セキュリティアップデートとパッチ管理、システムの監視など、全面的なセキュリティ対策を考慮する必要があります。
- ③ データの検証とクリーニング：システムは正確なデータを扱う必要があるため、入力データの検証とクリーニングの仕組みが重要です。例えば、不正確な入力を防ぐための形式チェック、既存のデータとの整合性チェックなどがあります。
- ④ 柔軟な設計：企業の成長やビジネス環境の変化に対応できるよう、システムは柔軟であるべきです。これは、モジュール性の高い設計、設定やパラメータによる動作のカスタマイズなどを意味します。
- ⑤ 利用者の利便性：ユーザーフレンドリーな設計は重要なポイントです。必要な情報をすばやく見つけられるようなユーザーインターフェース、直感的な操作性、十分なヘルプとマニュアルがあるか、等を確認する必要があります。

売上管理システムをローコード開発で作るときに必要な項目

- ✓ 商品データベース：商品の情報（商品名、商品ID、価格など）を管理します。商品情報は一元管理され、新規商品の追加や既存商品の情報変更が容易に行えるようなシステムが求められます。
- ✓ 購入者情報の管理：顧客情報（顧客名、住所、購入履歴など）を追跡します。これにより、過去の購入パターンを理解し、個々の顧客に対する販売戦略を立てることが可能となります。
- ✓ 売上データの追跡：各商品の売上データ（販売数量、販売日、販売先顧客など）を正確に追跡します。これにより、販売トレンドを把握し、在庫管理や販売予測に利用することが可能となります。
- ✓ レポートの生成：売上データから各種レポート（日次売上レポート、月次売上レポート、商品別売上レポートなど）を自動生成できれば、それぞれの商品のパフォーマンスを把握し、戦略の調整に役立てることができます。
- ✓ 予測機能：過去の売上データを基に未来の売上を予測します。この機能により、商品の需要を予測し、適切な在庫管理や商品の再注文スケジュールを計画することができます。
- ✓ セキュリティ対策：顧客情報や売上データなど、ビジネス上重要なデータの安全性を確保します。データ暗号化、アクセス権限の制御、セキュリティアップデートとパッチの適用などを行なう必要があります。
- ✓ データバックアップと復元：データロスを防ぐために、定期的なデータバックアップと、必要な場合にはデータを迅速に復元するためのシステムが必要です。災害やシステム障害が発生した場合でも、ビジネスの運用を迅速に再開できるようにするためです

売上管理システムの開発手順

- ①目的設定・要件定義
どのような売上データを追跡するのか、どのようなレポートが必要なのか、どのような分析機能が必要なのかなど、ユーザーからの要求をリストアップします。顧客やステークホルダーとのミーティングを通じてこれらの情報を収集します。
- ②機能設計
収集したビジネス要件を基に、システムの詳細な機能を設計します。これには、商品管理、顧客管理、在庫管理、売上データの入力と追跡、各種レポートの生成、セキュリティ対策などが含まれます。
- ③データベース設計
システムが扱うデータとその関係を定義するためのデータベース設計を行います。商品データ、顧客データ、売上データなど、それぞれのデータ構造と、これらの間のリレーションシップを定義します。
- ④システム開発
設計を基に、システムの実際の開発を行います。具体的な機能ごとに開発を進め、各機能が要件を満たしていることを確認します。同時に、データの正確性を保証するためのデータバリデーションやエラーチェック機能も開発します。

受発注管理システム開発の5つのポイント

- ① **データの正確性**: データ入力の簡易化（自動化やドロップダウンメニューの使用等）や、データのバリデーションチェックをシステムに組み込むことで、データの誤入力や不整合を防ぎます。また、定期的なデータ監査機能も重要です。
- ② **セキュリティ**: 顧客データや取引情報の保護は非常に重要です。システムには、データ暗号化、不正アクセスの検出、安全なログインメカニズム（2要素認証等）などのセキュリティ機能を組み込む必要があります。
- ③ **通知システム**: 在庫不足、注文のステータス変更、未払い請求書など、重要な更新情報を適時に通知します。
- ④ **統合**: 他のビジネスシステム（例えば、会計ソフトウェアやCRMツールなど）との統合を可能にします。これにより、情報を一元化し、業務効率を向上させます。
- ⑤ **商品・サービス管理**: 商品やサービスのカタログを管理し、価格、詳細、在庫状況などの情報を更新します。

受発注管理システム開発をローコード開発で作るときに必要な項目

- ✓ **注文管理**: 受注から出荷までのプロセスをトラッキングします。例えば、顧客からの注文の受け付け、注文内容の確認、出荷ステータスの更新などを行います。重要なのは、注文情報がリアルタイムで更新され、関係者が常に最新の状況を確認できることです。
- ✓ **在庫管理**: 商品の在庫数を一元管理し、在庫が不足した場合のアラート機能などを持つことが理想的です。例えば、特定の商品の在庫が設定した下限数を下回ったら、自動的に発注指示を出すなどの機能があると効率的です。
- ✓ **顧客管理**: 顧客の連絡先情報、注文履歴、支払い情報などを一元化します。これにより、過去の購入パターンを分析したり、特定の顧客に対するカスタマーサービスを向上させることが可能になります。
- ✓ **商品・サービス管理**: それぞれの商品やサービスについて、価格、詳細説明、画像、在庫状況などの情報を管理します。新商品の追加や価格の変更など、商品情報の更新が容易に行えることが重要です。
- ✓ **支払い管理**: 請求書の作成、送付、支払いの追跡を行います。例えば、特定の顧客が請求期限を過ぎても支払いを行わない場合、その顧客に対して自動的にリマインダーを送る機能があると便利です。
- ✓ **レポートと分析**: 売上、人気商品、顧客の購入傾向など、ビジネスのパフォーマンスを測定するためのレポートを生成します。例えば、過去の売上データを基に次の季節の商品発注量を予測するなどの分析が行えると良いでしょう。

受発注管理システムの開発手順

- ① **要件定義**: 受発注システムの開発は、まず必要な機能を明確にすることから始まります。ユーザーと一緒にワークショップを開き、どのような注文プロセスを管理するのか、どのような在庫情報を追跡するのか、どのようなレポートが必要なのかなど、詳細な要件をリストアップします。
- ② **設計**: 次に、要件を基にシステム設計を行います。これには、データベースの設計（どのような情報を格納し、どのように関連付けるか）、ユーザーインターフェースの設計（各画面でどのような操作を行うか）、システムアーキテクチャの設計（フロントエンドとバックエンドの連携、セキュリティ対策）などが含まれます。
- ③ **開発**: 設計した要件に基づいてソフトウェアのコーディングを行います。たとえば、商品カタログの管理機能、注文入力と追跡の機能、在庫の自動更新機能など、具体的な機能を一つずつ開発します。
- ④ **テストとデプロイ**: 開発した機能が正しく動作することを確認するために、各種テスト（ユニットテスト、結合テスト、システムテスト）を行います。全てのテストがパスしたら、システムを本番環境にデプロイします。本番環境での動作確認後、ユーザートレーニングを行い、システムの運用を開始します。

FAQの5つのポイント

- ① **ユーザーエクスペリエンス**: FAQシステムはユーザーが問題を解決するための道しるべであるため、使いやすさとアクセシビリティは最優先事項でなければなりません。ユーザーが求める情報を簡単に見つけられるようにすることが重要です。
- ② **カテゴライズとタグ付け**: すべての質問を適切にカテゴライズし、関連するタグを付けることで、ユーザーが必要な情報を素早く見つけられるようにすることができます。
- ③ **最新の情報を保つ**: FAQは定期的に更新され、最新の情報を反映している必要があります。情報が古いと、ユーザーの混乱を招く可能性があります。
- ④ **明確な言語を使用する**: FAQの回答は明瞭で簡潔でなければなりません。不明確な言葉や業界用語を使うと、ユーザーが混乱する可能性があります。
- ⑤ **ユーザーフィードバックを活用する**: ユーザーからのフィードバックは、FAQシステムの改善に大いに役立ちます。ユーザーが求めている情報を理解するために、これらのフィードバックを定期的に検討することが重要です。

FAQをローコード開発で作るときに必要な項目

- ✓ **導入目的が何なのか考える**: 開発するFAQシステムの導入目的を明確に定義します。それは製品のサポート、ユーザーガイド、技術サポートなど、具体的な目的に合わせて内容と設計を決定します。
- ✓ **検索機能の実装**: ユーザーが必要な情報を素早く見つけられるよう、効率的な検索機能を搭載することが重要です。キーワード検索、カテゴリ検索などを組み合わせることで利便性を高めます。
- ✓ **ナビゲーションの最適化**: FAQが多数存在する場合、ナビゲーションは非常に重要な要素になります。質問を適切なカテゴリに分け、ユーザーが情報を容易に探し出せるよう工夫します。
- ✓ **明瞭な質問と回答**: 質問は具体的でわかりやすい言葉を使用し、回答は簡潔で明確にすることが重要です。また、専門用語を使用する場合はその説明も提供すると良いでしょう。
- ✓ **ユーザーフィードバック**: ユーザーからのフィードバックを収集し、システムの改善に活用します。また、FAQが解決しきれない問題が発生した場合の連絡窓口も明示すると良いでしょう。

FAQの開発手順

- ① **要件定義と設計**
まず、システムの目的とその目標を明確に定義します。この段階では、システムの全体的な概念設計を作成し、何を達成しようとしているのか、どのような機能を備えているべきなのかを詳細に記述します。また、特定のユーザーグループをターゲットにしたシステムの設計や、必要なインタラクションを含むユーザーエクスペリエンスの設計も含まれます。
- ② **コンテンツ作成**
次に、FAQシステムで提供する質問とその回答を作成します。これには、ユーザーから最も頻繁に寄せられる質問を集め、それらに対する明確で簡潔な回答を作成する作業が含まれます。質問は適切にカテゴライズされ、タグ付けされるべきです。
- ③ **システム開発**
設計と内容が決まったら、システムの実際の開発が始まります。開発チームは設計ドキュメントをもとに、FAQシステムの各機能をコーディングします。この段階では、システムがユーザーの要求に応じて適切に動作することを確認するため、頻繁にテストが行われます。
- ④ **リリースとメンテナンス**
システムが開発され、テストされた後、リリースが行われます。しかし、リリースが行われた後も、システムは継続的にメンテナンスされ、更新される必要があります。これには、新しい質問と回答の追加、既存の内容の更新、バグの修正、そしてユーザーフィードバックに基づいた機能の改善などが含まれます。

日報・報告書の5つのポイント

- ① 報告書閲覧機能：作成された報告書を検索し、閲覧できる機能
- ② コメント・承認機能：報告書に対するコメントや承認作業が行える機能
- ③ データエクスポート機能：報告書のデータをエクセルやPDFなど他のフォーマットにエクスポートする機能
- ④ セキュリティ管理：ユーザのパスワード管理やデータの暗号化など、情報のセキュリティを担保する機能
- ⑤ 報告書入力機能：使用者が日報・報告書を簡単に作成できる入力フォーム

日報・報告書をローコード開発で作るときに必要な項目

- ✓ テンプレート管理：報告書の形式を設定するためのテンプレートを作成・管理できる機能が必要です。これにより、ユーザーはテンプレートを選ぶだけで一貫した形式の報告書を作成できます。
- ✓ 入力フォーム設定：報告書に必要な情報を入力するためのフォーム設定機能が必要です。ユーザーが報告書を作成する際に、項目名や入力形式（テキストボックス、ドロップダウン、日付選択など）を自由に設定できるようにします。
- ✓ データ連携：他のシステムやデータベースとの連携機能が重要です。例えば、CRMやERPと連携することで、既存の顧客情報や販売データを報告書に自動的に反映させることができます。
- ✓ 承認フロー設定：報告書の承認フローを設定できる機能が必要です。特定の報告書が上司の承認を必要とする場合、そのフローを自由に設定し、承認者に通知を送ることができます。
- ✓ アクセス権管理：ユーザーやグループに対して、報告書の閲覧・編集・承認などの権限を設定できる機能が必要です。これにより、情報のセキュリティを確保しつつ、必要なユーザーに対して適切なアクセス権を提供できます。
- ✓ 検索・フィルタリング：作成された報告書から特定の情報を素早く見つけるために、検索やフィルタリング機能が必要です。これにより、ユーザーは日付、作成者、キーワードなどに基づいて報告書を簡単に検索できます。

日報・報告書の開発手順

要件定義：この段階では、日報・報告書のシステムの目的と具体的な機能要件を詳細に明確化します。例えば、どのような報告書形式をサポートするのか、どのような検索機能が必要なのか、承認フローはどのように動作するのかなどを決定します。

設計：要件をもとに、システムの全体設計を行います。具体的には、データベース設計（どのような情報をどのように保存するか）、ユーザーインターフェース設計（報告書の作成や閲覧の流れ、画面レイアウト）、システムアーキテクチャ設計（サーバーやフロントエンドの構造）などを行います。

開発：設計を元に、具体的なコーディングを行います。日報・報告書作成機能、編集機能、閲覧機能、検索機能、承認機能などの開発を進めます。この段階では、コードレビューや単体テストを行いながら、品質を確保します。

統合テスト：各機能が一緒に動作するかを確認するための統合テストを行います。また、実際のデータを使って、パフォーマンスやセキュリティをチェックします。

チーム編成と役割

プロジェクトマネージャー



プロジェクトマネージャーは、ローコード開発プロジェクト全体の進捗管理やリソース調整、品質管理などを担当します。彼らはチームを統括し、プロジェクトの目標達成に向けて最適な方向性を示します

ビジネスアナリスト



ビジネスアナリストは、業務要件の分析や仕様策定、ユーザーのニーズを把握する役割を担います。彼らは実際の業務プロセスを理解し、アプリケーション開発に必要な機能やワークフローを定義します。

デザイナー



デザイナーは、アプリケーションのユーザーインターフェース（UI）やユーザー経験（UX）をデザインします。彼らはビジュアルデザインや操作性を考慮し、アプリケーションの使いやすさや魅力を高めます。

開発者



開発者は、ローコード開発プラットフォームを利用してアプリケーションを実装します。彼らはドラッグ・アンド・ドロップやビジュアルプログラミングを使い、ビジネス要件を満たす機能やワークフローを開発します。

プラットフォームの選び方

まとめ

必要な機能と価格帯

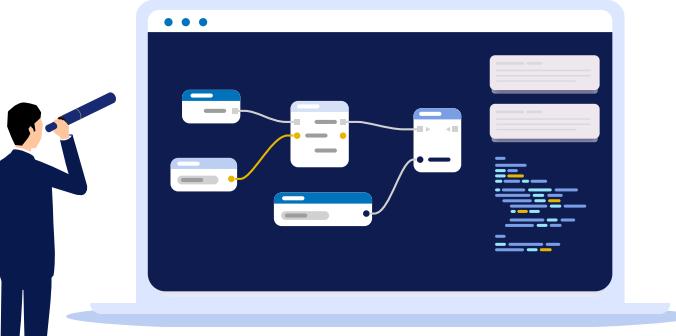
選ぶローコード開発プラットフォームには、自社の業務要件や予算に合った機能と価格帯が重要です。機能面では、必要なコンポーネントやワークフロー、データベース管理機能などをチェックしましょう。価格帯は、利用規模やサポート範囲によって異なるため、比較検討が必要です。

サポート体制

サポート体制も、ローコード開発プラットフォーム選びにおいて重要な要素です。質問やトラブルシューティングに対応してくれるサポートチームや、継続的なアップデート・メンテナンスが提供されるかどうかを確認しましょう。

他システムとの連携性

ローコード開発プラットフォームは、既存のシステムやサービスとの連携が容易であることが望ましいです。API連携やプラグインの提供など、他システムとの連携性を評価する際のポイントを把握しましょう。



ローコード開発の今後の展望

ローコード開発は今後も技術革新が進み、さらなる機能拡張やユーザビリティの向上が期待されます。AIや機械学習の進化により、アプリケーション開発がより簡単かつ迅速になるでしょう。また、より多くの業界や業務領域でローコード開発の活用が広がり、デジタルトランスフォーメーションの推進に貢献すると予想されます。



本ホワイトペーパーで得た知識の活用方法

本ホワイトペーパーで得た知識を活用し、自社の業務改善や効率化を図るためのローコード開発プロジェクトを立ち上げましょう。適切なプラットフォーム選定、チーム編成、連携システムの整備など、ローコード開発の成功に必要な要素を網羅的に考慮し、効果的なアプリケーション開発を実現してください。

サービス概要

業務プロセスのデジタル化・自動化を実現するシステム共通基盤



攻めのDXと守りのDXを実現

豊富な機能を取り揃えたアプリケーションプラットフォーム

「intra-mart」は、企業内に存在する様々な業務システムを同一のプラットフォーム上に集約し、最新のデジタル技術を活用することで、**IT投資の効率化と業務プロセスの最適化・標準化**を実現します。さらに、業務プロセスのフルオートメーション化をサポートする機能とAPIコンポーネント群を多数取り揃えており、**スピーディかつ柔軟なローコードアプリケーション開発**を可能にします。グループ企業全体での共同利用はもちろん、クラウド利用も可能です。

INDEX

intra-mart Accel Platformは、企業特有のニーズにあわせたカスタマイズしやすいOpenな開発環境と全社員がすぐに使えるEasyさを備えています。システム開発に欠かせない機能も豊富に取り揃えており、企業のあらゆる業務課題をサポートします。



ローコード／アジャイル開発

このような課題を解決

- ✓ IT人材不足を解消したい
- ✓ ビジネスニーズの変化に合わせて、素早く内製でシステム開発をしたい



業務プロセス改善

このような課題を解決

- ✓ 紙運用の業務をデジタル化したい
- ✓ 業務プロセスを見直して最適化したい



システム共通基盤

このような課題を解決

- ✓ 他システムとの連携でユーザビリティを高めたい
- ✓ バラバラな業務システムを効率化させたい

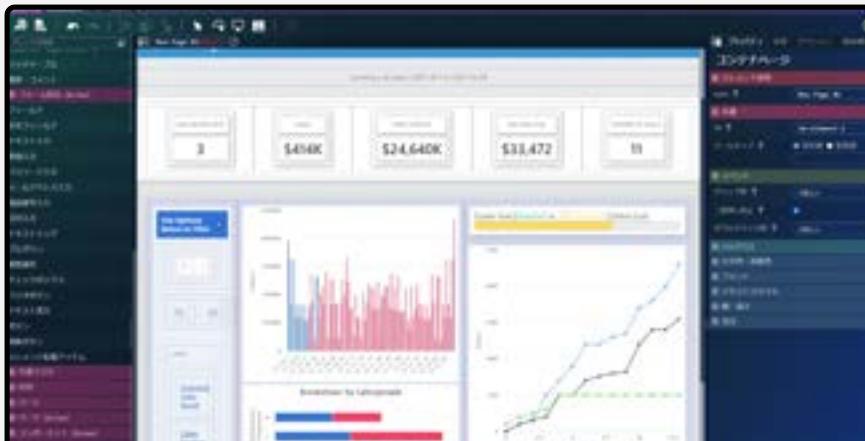
ローコード／アジャイル開発

誰でも作れる、変えられる、運用できる 高い業務生産性を実現する「ローコード開発」

簡易なフォーム画面から複雑な業務画面まで、ドラッグ＆ドロップなどの操作で簡単に画面を作成できます。様々なUI部品も備えており、作成した画面はintra-martのワークフロー/BPMとも連携可能です。また、PCやスマートフォン等様々なデバイスで実行できます。

01 Webブラウザ上でかんたん画面作成

簡易なフォーム画面から複雑な業務画面まで、ドラッグ＆ドロップなどの操作で簡単に作成することができます。intra-martのBPM/ワークフローとも連携可能です。作成した画面は、PCやスマートフォン、タブレットなど様々なデバイスで実行可能です。



02 ノンコーディングで業務ロジック作成

プログラミングの知識がない方でも、様々な業務処理の部品をドラッグ＆ドロップで配置し線でつなげるだけで、業務ロジックを作成することができます。今までコーディングが必要だった処理ロジックもコーディング不要になり、開発生産性の向上が期待できます。



03 柔軟な拡張性と高いカスタマイズ性

システム運用後に生じた変更も、運用を止めることなくWebブラウザ上で設定変更してリリースすることができます。intra-martが持つ様々な業務コンポーネント群と連携できるため、エンタープライズに必要なアプリケーションもお客様に合わせて自由にカスタマイズ可能です。



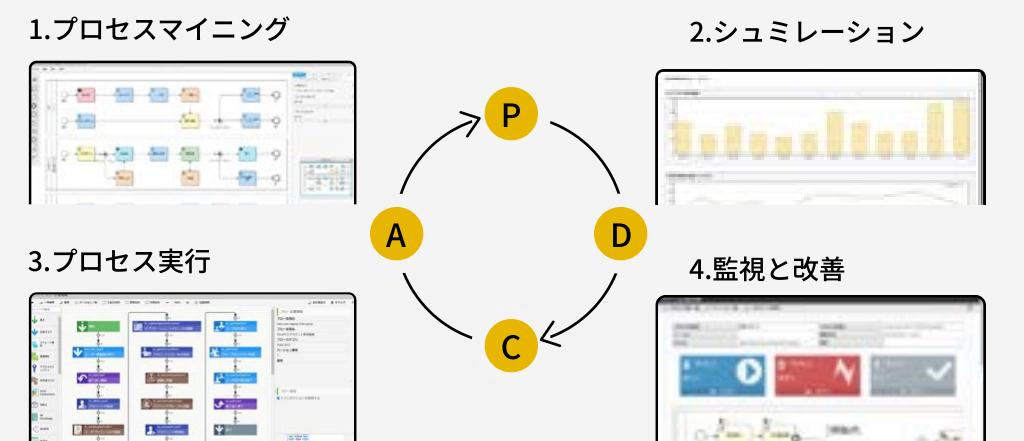
業務プロセス改善

業務プロセスのデジタル化・自動化を実現 業務プロセスのスピードアップ

紙で運用している業務のデジタル化はもちろん、システムの個別導入によって分断された様々な業務を一連のプロセスとして可視化し、継続的な改善によって効率化を実現する、業務プロセス管理ツールが揃っています。

01 継続的な業務プロセス改善を実現

各部門の業務プロセスを可視化し、定義～実行～モニタリング～改善という、継続的な業務改善の仕組みを確立することができます。現在のパフォーマンス状況を表示したり、過去の状況や今後の予測も可能です。



02 ワークフロー×ローコード開発による 高い開発生産性

intra-martのBPM/ワークフローとローコード開発ツールを組み合わせることで、システム構築において高い開発生産性を実現します。カスタマイズ性が高く、業界問わずお客様の業務・運用に合わせたシステムが構築可能です。

03 包括的なDX業務改革をトータルサポート

BPMを活用した業務プロセス改善を実践するための様々なサポートサービスを提供しています。DX人材を育成するプログラムやるべき業務プロセスの策定、製品の導入、ア�파ーフォローまでトータルで支援しますので、初心者の方も安心してご利用いただけます。



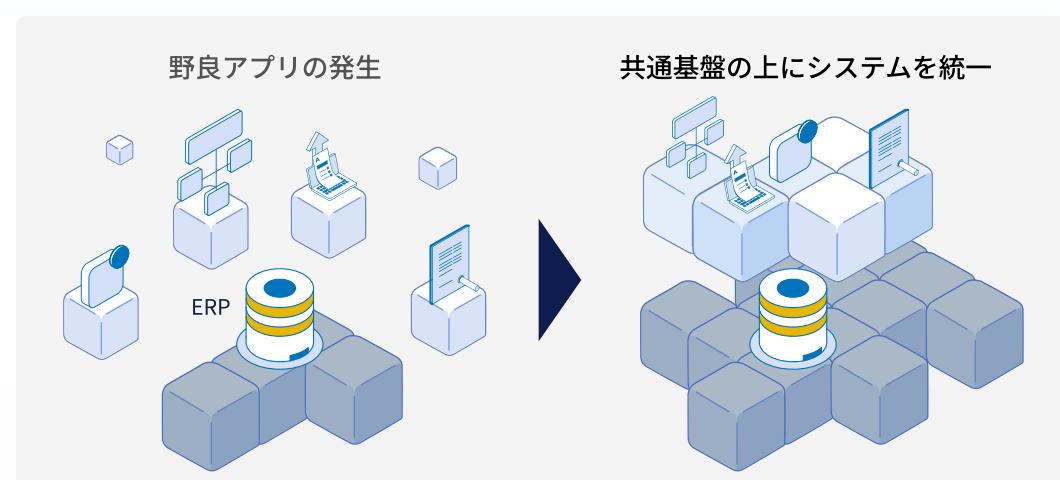
システム共通基盤

混在する業務システムを集約することで IT投資の効率化・標準化を実現

業務システムの集約によって、IT投資の効率化と標準化を実現します。ガバナンスを保ちながら、小規模利用から全社・グループ展開へと、ビジネスの成長に合わせて利用範囲を拡張させることができます。

01 システム共通基盤によるガバナンス

企業内に存在する業務システムを一つのプラットフォーム上に集約することで、IT投資の効率化と業務の標準化を実現します。また、PaaS基盤としてグループ企業内の共同利用も可能です。

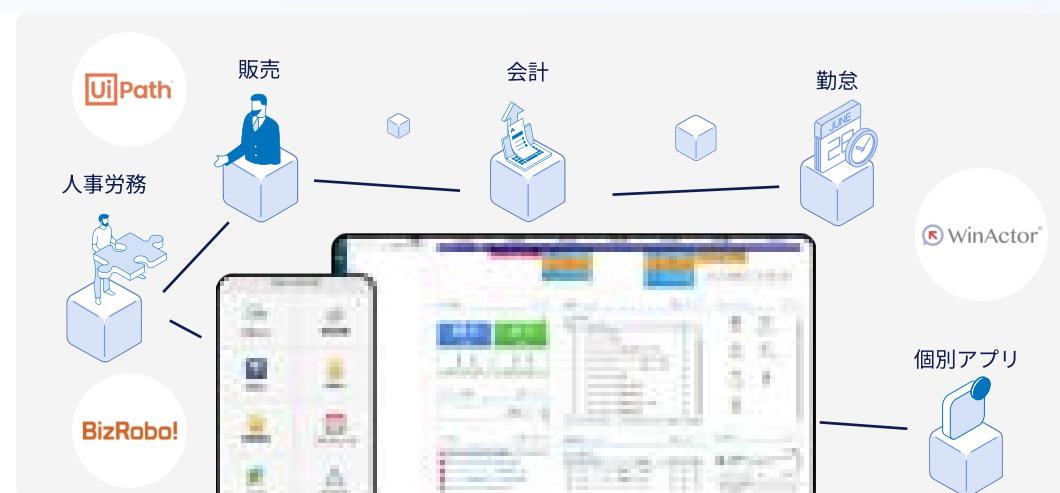


02 開発ツールの組み合わせで自由にカスタマイズ

プラットフォーム上に用意されている、ローコード開発を含む豊富な業務コンポーネントを活用することで、複雑なWebシステムもスピーディに構築できます。ソースコードも公開しているため、独自のフレームワークとしてカスタマイズ可能です。

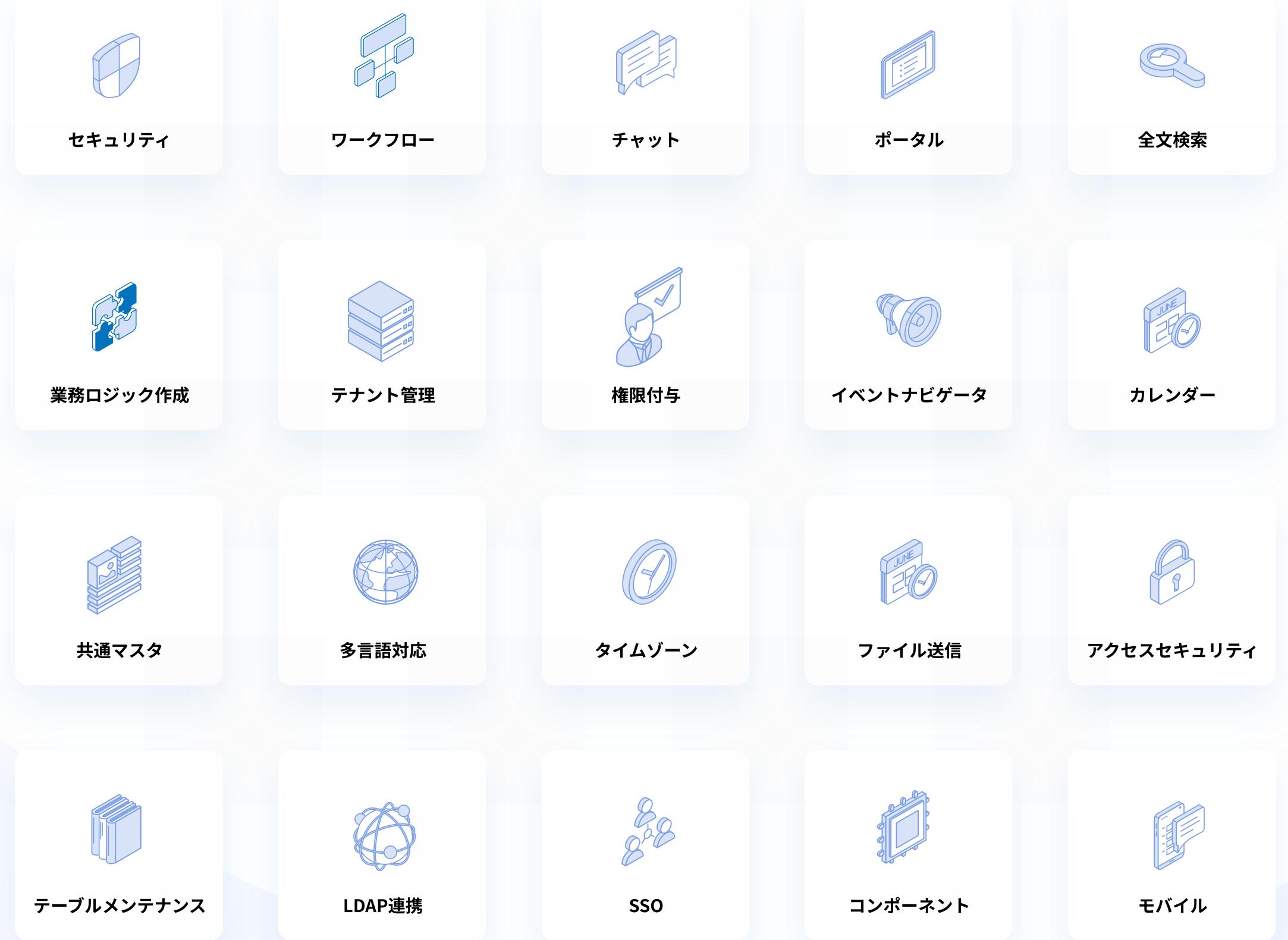
03 柔軟なシステム連携でユーザビリティを向上

外部システムをつなぐモジュールによって、グループウェアといった情報系システムや会計などの基幹システムに同一画面からアクセスできるため、操作性を大幅に高め、業務効率を向上します。



その他の機能

Webシステム開発でよく利用される機能をAPIで数百種類提供しています。これらを自由に組み合わせてシステム開発を行うことで、お客様の業務に合ったアプリケーションを短期間で柔軟に構築することができます。





株式会社NTTデータ イントラマート

東京都港区赤坂四丁目15番1号 赤坂ガーデンシティ5階

<https://www.intra-mart.jp/>

- ・掲載内容は2024年10月現在のものです。
- ・intra-martは株式会社NTTデータ イントラマートの登録商標です。
- ・その他の各種製品名は、各社の製品名称、商標または登録商標です。