

第4章 新基幹システム（ERP パッケージ）の構想と要件定義

第3章では、現状の業務プロセスやシステムに潜む課題を徹底的に可視化しました。これらの課題は、レガシーシステムが長年にわたり蓄積してきた「IT 負債」であり、企業の成長を阻害する大きな要因となっています。

この第4章では、その課題分析の結果を基に、目指すべき未来の姿、すなわち「To-Be（あるべき姿）」を具体的に描き出します。レガシーERP からの脱却は、単にシステムを入れ替えることなく、業務プロセスそのものを刷新し、企業の競争力を高めるための変革プロジェクトです。

本章では、新しい ERP パッケージの標準機能を最大限に活用しながら、未来の業務プロセスをどのように設計し、具体的な要件として定義していくかを詳述します。このプロセスを通じて、新しい ERP が単なるツールではなく、企業の成長戦略を実現するための「戦略的な武器」となるための道筋を示します。

4.1 新業務プロセスの設計と最適化：ERP 標準機能の最大限活用

新しい ERP の導入は、単なる既存の業務を新しいシステムに置き換える作業ではありません。それは、自社のビジネスモデルや市場環境の変化に合わせて、業務プロセスを再構築（BPR: Business Process Re-engineering）する絶好の機会です。

このセクションでは、ERP の標準機能という「ベストプラクティス」を最大限に活用しながら、未来の業務プロセスをどのように設計・最適化していくかを解説します。

4.1.1 To-Be 業務フローの策定と ERP 導入による業務変革

未来の業務プロセスである「To-Be 業務フロー」は、ERP 導入プロジェクトにおける羅針盤です。これを策定することで、関係者全員が目指すべきゴールを共有し、円滑にプロジェクトを進めることができます。

1. To-Be 業務フロー策定の基本原則:

- **Fit to Standard の徹底:** ERP 導入の最大のメリットは、業界のベストプラクティスが凝縮された標準機能を利用できることです。To-Be 業務フローは、この標準機能を最大限に活用することを基本原則とします。既存の業務プロセスを新しい ERP に合わせることで、カスタマイズを最小限に抑え、コスト削減、導入期間の短縮、将来のバージョンアップを容易にするというメリットを享受できます。
- **ビジネス要件の優先:** 標準機能に合わせることを目指しますが、それに固執しすぎるあまり、ビジネス上不可欠な要件を犠牲にしてはなりません。To-Be 業務フローは、経営目標やビジネス要件を達成するための最適なプロセスを追求します。

- **部門横断的な視点:** 特定の部門の最適化だけでなく、企業全体の視点からプロセスを設計します。部門間のデータのやり取りや連携をスムーズにし、全体最適を目指します。

2. To-Be 業務フローの策定ステップ:

- **ステップ 1: 理想の業務プロセスの議論:**
 - 第 3 章で特定した課題（ボトルネック、非効率）を解消することを念頭に、部門横断的なワークショップを開催します。
 - 参加者は、新しい ERP で「どのような業務を実現したいか」「現在の課題をどう解決したいか」を自由に議論します。
- **ステップ 2: ERP 標準機能とのマッチング:**
 - 議論で出た理想のプロセスを、検討中の ERP が提供する標準機能と照らし合わせます。
 - 「この課題は、ERP のこの機能で解決できる」「この業務は、標準のワークフローに乗せられる」といった具体的な対応策を検討します。
- **ステップ 3: To-Be 業務フローの図式化:**
 - BPMN などの表記法を用いて、最終的な To-Be 業務フローを図式化します。
 - この際、手作業だった部分が自動化される、承認ステップが簡素化される、といった「業務変革」の内容を明記します。
- **ステップ 4: 関係者による合意形成:**
 - 策定した To-Be 業務フローを、経営層、各部門のキーユーザー、IT 部門など、すべての関係者と共有し、**合意形成**を行います。この段階で合意を得ることで、プロジェクト後半での手戻りを防ぎます。

例えば、ある**国内製造業の事例**では、受注情報の手動転記によるミスや部門間の連携不足が大きな課題でした。新しい ERP の導入では、受注情報をシステムに一度入力するだけで、在庫確認、生産計画、出荷指示、請求処理までが自動で連携される To-Be 業務フローを策定。これにより、情報伝達のコストをなくし、業務リードタイムを大幅に短縮する道筋を描きました。

To-Be 業務フローは、ERP 導入を通じて企業がどのような変革を実現したいのかを明確にする、最も重要な成果物の一つです。

4.1.2 自動化、デジタル化による効率化と ERP の貢献

新しい ERP の導入は、これまで手作業で行われていた業務を**自動化・デジタル化**し、従業員の生産性を飛躍的に向上させる絶好の機会です。ERP は、単なる記録のシステ

ムではなく、業務プロセス自体を変革するツールです。

1. ERP による自動化・デジタル化の具体例:

- **ワークフローの自動化:**
 - 紙ベースで行われていた稟議や経費精算の承認プロセスを、ERP の電子ワークフローに移行。これにより、申請・承認状況が可視化され、プロセスが迅速化されます。
- **データの自動連携:**
 - 受注データが入力されると、自動的に在庫が引き当てられ、出荷指示が発行される。また、出荷が完了すると、自動的に請求書が生成・発行される。これにより、手作業によるデータ転記や連携のムダをなくします。
- **レポートの自動生成:**
 - 経営層が求める売上レポートや在庫分析レポートを、ERP のリアルタイムデータを基に自動で生成・配信。手作業でのデータ集計・加工が不要となり、経営判断の迅速化に貢献します。
- **RPA (Robotic Process Automation) との連携:**
 - ERP の標準機能で自動化できない、定型かつ反復的な作業（例：他社 Web サイトからの情報取得、データの CSV 出力）を、RPA と連携して自動化。ERP と RPA を組み合わせることで、さらなる業務効率化を実現します。

2. 効率化の評価指標 (KPI) 設定:

ERP 導入による自動化・デジタル化の効果を客観的に評価するためには、事前に評価指標 (KPI: Key Performance Indicator) を設定することが重要です。

- **業務リードタイムの短縮:** 「受注から請求までの期間を 3 日から 1 日に短縮する」など。
- **事務作業時間の削減:** 「経理部門の月次決算にかかる時間を 50%削減する」など。
- **人的ミスの削減:** 「手動でのデータ入力によるミスをゼロにする」など。
- **ペーパーレス化率の向上:** 「紙の伝票の使用を 80%削減する」など。

国内建設業の事例では、プロジェクトの進捗管理が紙の報告書に依存し、情報のタイムラグが課題でした。新しい ERP の導入にあたり、現場の担当者がモバイル端末から進捗状況をリアルタイムで入力できる仕組みを設計。これにより、管理部門は常に最新の情報を把握でき、報告書作成の手間も削減され、業務効率が大幅に向上しました。

自動化・デジタル化によって、従業員は定型的な作業から解放され、より創造的で付

加価値の高い業務に集中できるようになります。この効率化は、コスト削減だけでなく、企業の競争力向上に直結します。

4.1.3 ユーザーエクスペリエンス（UX）の考慮と ERP インターフェース

新しい ERP を導入しても、現場の従業員が「使いにくい」と感じてしまえば、その導入効果は半減します。そのため、新システムを設計する際には、ユーザーエクスペリエンス（UX）を深く考慮することが不可欠です。

1. なぜ UX が重要なのか:

- **従業員の生産性向上:** 直感的で使いやすいインターフェースは、操作ミスを減らし、トレーニング期間を短縮します。これにより、従業員はすぐに新しいシステムに習熟し、本来の業務に集中できます。
- **利用定着率の向上:** 複雑で使いにくいシステムは、従業員に敬遠され、結果としてシャドーIT の再発を招く可能性があります。優れた UX は、従業員が積極的にシステムを利用する動機付けとなります。
- **データの品質向上:** 使いやすい入力画面は、データの入力ミスを減らし、品質の高いデータを確保する上で不可欠です。

2. UX を考慮した ERP インターフェース設計のポイント:

- **直感的な操作性:**
 - 誰でも直感的に使えるよう、ナビゲーションを分かりやすく設計します。
 - ユーザーが次のアクションを予測しやすいよう、画面のレイアウトを統一します。
- **パーソナライズとカスタマイズ:**
 - ユーザーの職務や役割に応じて、必要な情報や機能をダッシュボードに表示する**パーソナライズ機能**を活用します。
 - 各ユーザーが自身の業務に合わせて画面をカスタマイズできる機能（例：頻繁に使うメニューのピン留め）を提供することで、利便性を向上させます。
- **モバイル対応:**
 - 現場での作業が多い従業員（例：製造ライン、倉庫、営業担当者）向けに、スマートフォンやタブレットからアクセスできるモバイルインターフェースを提供します。これにより、場所を選ばずにリアルタイムで情報を入力・確認できるようになります。
- **フィードバックの仕組み:**
 - エラーメッセージを分かりやすく表示する、入力内容に誤りがないかリ

アルタイムでチェックするなど、ユーザーに適切なフィードバックを返す仕組みを組み込みます。

国内製造業の事例では、既存システムが特定の端末からの入力しかできず、現場作業員が作業場を離れてから入力をするため、在庫情報にタイムラグが発生していました。新しい ERP の導入では、スマートフォンで QR コードを読み取るだけで入出庫データを登録できるモバイルインターフェースを開発。現場の使いやすさを追求した結果、データのリアルタイム性が確保され、在庫管理の精度が大幅に向上しました。

ERP のインターフェースは、従業員が日々接する「企業の顔」です。使いやすさを追求することで、ERP 導入の成功と、従業員一人ひとりの生産性向上に大きく貢献します。

4.1.4 ワークショップを通じた新業務プロセスへのチェンジマネジメント

To-Be 業務フローを策定するだけでなく、それを組織全体に浸透させるためには、従業員の意識と行動を変える「チェンジマネジメント」が不可欠です。新業務プロセスへの移行は、従業員にとって大きな変化であり、不安や抵抗感が生じるのが自然です。この変化を円滑に進めるために、ワークショップは非常に有効なツールとなります。

1. チェンジマネジメントの重要性:

- **従業員のエンゲージメント向上:** ワークショップを通じて、従業員が新業務プロセスの策定に主体的に関わることで、「やらされ感」ではなく、「自分たちが変えるんだ」という意識が芽生えます。
- **抵抗感の軽減:** 新システム導入の目的やメリットを直接伝え、従業員の懸念や疑問に耳を傾けることで、変化への抵抗感を和らげます。
- **潜在的な課題の早期発見:** ワークショップでの活発な議論を通じて、To-Be 業務フローに潜む潜在的な課題や、現場でしか知り得ない特殊なケースを早期に発見できます。

2. ワークショップの実施方法:

- **対象:** 経営層、IT 部門、各業務部門のキーユーザー、一般ユーザーなど、幅広い層を巻き込みます。
- **内容:**
 - **新業務プロセスの共有:** 策定した To-Be 業務フローを図や動画で分かりやすく説明し、新システム導入によって何がどう変わるのかを具体的に示します。
 - **役割と責任の明確化:** 新しい業務プロセスにおける各担当者の役割や

責任範囲がどう変わるのかを明確にします。

- **Q&A セッション:** 質疑応答の時間を十分に設け、従業員の疑問や不安を解消します。
- **フィードバックの収集:** 新業務プロセスに対する意見や改善提案を収集し、可能な範囲で設計に反映させます。

国内製造業の事例では、新 ERP 導入に伴い、製造現場での報告プロセスが大きく変わることになりました。プロジェクトチームは、現場の作業員を対象に数回のワークショップを開催。新システムがもたらすメリット（報告書の作成時間が半分になるなど）を具体的に示し、スマートフォンでの簡単入力操作を実際に体験してもらいました。この取り組みにより、当初は変化に抵抗感を示していた現場の従業員が、徐々に新システム導入に前向きになり、スムーズな移行を実現できました。

ワークショップは、単なる情報伝達の間ではありません。それは、組織文化を変革し、新システムを成功裏に定着させるためのコミュニケーションの場であり、チェンジマネジメントの中核をなす活動です。

4.1.5 データ移行と連携の事前検討：円滑なシステム移行のために

To-Be 業務フローを策定し、新しいシステムがどのように動くかを明確にしたら、次は、そのシステムが利用する「データ」について具体的に検討する必要があります。データ移行と連携の計画は、円滑なシステム移行を実現するための重要なステップです。

1. データ移行計画の策定:

- **移行対象データの特定:**
 - 第 3 章で実施したデータアセスメントの結果を基に、マスターデータ（顧客、製品など）、および過去何年分のトランザクションデータ（受注、売上など）を新システムに移行するかを決定します。
 - すべてのデータを移行するのではなく、直近のデータのみを移行し、古いデータはアーカイブとして保管するなどの戦略も検討します。
- **データ移行方法の検討:**
 - 手動入力、CSV ファイルによるインポート、専用のデータ移行ツール（ETL ツール）の利用など、データの量や質に応じて最適な移行方法を検討します。
- **データクレンジングの計画:**
 - 移行するデータに不備や重複がある場合は、移行前にデータクレンジングを実施する計画を立てます。これにより、新システム稼働後のデータ品質問題を未然に防ぎます。

2. データ連携計画の策定:

- **連携対象システムの特定:**
 - 新システム（ERP）と連携が必要な周辺システム（CRM、SCM、EC サイトなど）をすべてリストアップします。
- **連携方式の検討:**
 - 各システム間の連携を、バッチ処理（決まった時間に一括処理）、または API 連携（リアルタイム連携）のどちらで行うかを決定します。
 - 特に、リアルタイム性が求められる重要なデータ（例：EC サイトからの受注情報）については、API 連携を基本とします。
- **データ連携基盤の検討:**
 - 複数のシステムと複雑に連携する場合、iPaaS（Integration Platform as a Service）などのデータ連携基盤の導入を検討します。これにより、連携の管理を効率化し、システムの柔軟性を高めます。

国内製造業の事例では、複数の拠点に分散していた生産・在庫データを新 ERP に統合する必要がありました。プロジェクトチームは、まず各拠点のデータを CSV 形式で抽出し、統合マスターデータに基づきクレンジングを実施。その後、専用の ETL ツールを用いて、段階的にデータを新システムに移行する計画を策定しました。これにより、複雑なデータ移行作業を効率的に進めることができました。

データ移行と連携の計画は、ERP 導入プロジェクトの後半で大きな問題となりがちです。この段階で十分に検討し、関係者間で合意しておくことで、プロジェクト全体の成功確率を高めることができます。

4.1.6 To-Be プロセスとデータに基づく要件定義書の作成

To-Be 業務フローとデータ移行・連携計画がまとまったら、これらを基に、次のステップである**要件定義**の主要な成果物である**要件定義書**を作成します。これは、ベンダーとの間で認識のズレがないように、新システムに求められるすべての要求を網羅的に文書化する作業です。

1. 要件定義書に含めるべき主要な項目:

- **はじめに:** プロジェクトの目的、背景、およびスコープを明確にします。
- **To-Be 業務フロー:** 策定した To-Be 業務フローを図と文章で詳細に記述します。
- **機能要件:** 「システムが何をするか」という要求事項を一覧化し、各機能の詳細（目的、トリガー、インプット、アウトプット）を記述します。
- **非機能要件:** 「システムがどのように動作するか」という品質に関する要求事

項（可用性、性能、セキュリティなど）を記述します。

- **データ要件:** 移行対象データ、データ連携、マスターデータ管理に関する要求を記述します。
- **移行要件:** システム移行のロードマップ、カットオーバー計画、トレーニング計画などを記述します。

2. 作成のポイント:

- **網羅性と具体性:** 要求事項が漏れなく、かつ曖昧な表現がないように記述します。特に、「できるだけ早く」や「使いやすく」といった主観的な表現は避け、具体的な数値（例:「3 秒以内に応答する」「画面遷移を 3 ステップ以内にする」）に落とし込みます。
- **関係者とのレビュー:** 作成した要件定義書は、ベンダーだけでなく、社内の各業務部門のキーユーザー、経営層と綿密にレビューを行い、すべての関係者が内容に合意したことを確認します。

国内製造業の事例では、要件定義書に曖昧な記述が多く、開発フェーズに入ってからベンダーとの間で認識のズレが多発しました。例えば、「生産計画の最適化機能」という要件に対し、ベンダーは単純なスケジューリング機能と解釈しましたが、ユーザーは複雑な制約条件（例：特定の設備は連続して使用できない）を考慮した最適化を求めています。この失敗から、要件定義書の作成は、単なる文書作成ではなく、コミュニケーションを通じて認識をすり合わせる最も重要な活動であることが再認識されました。要件定義書は、プロジェクトの憲法とも言える重要な文書です。この文書を丁寧に、かつ具体的に作成することで、その後の設計、開発、テストといった各フェーズでの手戻りを防ぎ、プロジェクト全体の成功へと繋がります。

4.2 ERP システム要件定義と機能・非機能要件の明確化

To-Be 業務フローで描いた理想の姿を実現するためには、それを具体的な**要件**として定義する必要があります。要件定義は、プロジェクトの目的、スコープ、期待される成果物を明確にし、ベンダーとの認識のズレを防ぐための最も重要なプロセスです。要件は、**機能要件**と**非機能要件**に大別されます。

4.2.1 ビジネス要件から ERP 機能要件へのブレイクダウン

ERP 導入プロジェクトの成功は、ビジネス上の課題や目標を、システムが提供すべき「機能要件」へと**正確に落とし込めるかにかかっています**。このプロセスをブレイクダウン（分解）と呼び、To-Be 業務フローで描いた理想を、具体的なシステムの振る舞いとして定義します。この作業が甘いと、稼働後に「求めていたものと違う」という深

刻な問題に直面するリスクが高まります。

1. ブレイクダウンの目的と重要性:

- **曖昧さの排除:** 「業務効率を向上させる」といった抽象的なビジネス要件を、「受注入力時間を 50%削減する」「リアルタイムに経営ダッシュボードを更新する」といった具体的な機能要件に変換します。これにより、ベンダーが開発すべき内容が明確になります。
- **ベンダーとの共通言語化:** 開発ベンダーやコンサルタントは、ビジネス上の課題だけでなく、技術的な側面から要件を理解する必要があります。ブレイクダウンを通じて、専門家が理解できる共通の言葉にすることで、コミュニケーションの齟齬を防ぎます。
- **スコープの明確化:** プロジェクトで実現すべき機能の範囲を明確にし、要件の追加や変更によるコストとスケジュールの遅延を防ぎます。特に ERP 導入では、要件が膨らみがちであるため、どの機能が必須で、どの機能が追加要件なのかを厳密に区別することが重要です。

2. ブレイクダウンの具体的なステップ:

- **ステップ 1: ビジネス要件の特定:**
 - ワークショップやヒアリングを通じて、経営層や各部門のキーユーザーからビジネス上の要求を収集します。
 - 例: 「月次決算の早期化」「在庫のリアルタイム把握」「製品品質のトレーサビリティ確保」など。
- **ステップ 2: To-Be 業務フローとの紐づけ:**
 - 特定したビジネス要件が、策定済みの To-Be 業務フローのどの部分で実現されるかを紐づけます。これにより、業務プロセスとシステムのつながりを明確にします。
- **ステップ 3: ユースケースの作成:**
 - 業務フローを基に、システムとユーザー間のインタラクション（相互作用）をユースケースとして記述します。ユースケースは「誰が」「何を」するのかを簡潔に表現するものです。
 - 例: 「経理担当者が仕訳伝票を登録する」「在庫担当者が棚卸しを実行する」「製造担当者が生産実績を登録する」。
- **ステップ 4: 機能要件への具体化:**
 - 各ユースケースを、より詳細な機能要件にブレイクダウンします。この段階で、画面の項目、入力チェックルール、帳票の出力条件などを具体的に定義していきます。
 - 例（ビジネス要件: 在庫のリアルタイム把握）:

- **機能要件:** 「入出庫時に自動的に在庫数を更新する」「在庫数が設定した閾値を下回った場合にアラートを通知する」「任意の条件で在庫状況を検索・集計する」「モバイル端末からリアルタイムで入出庫登録できる」など。

このブレイクダウンプロセスを丁寧に進めることで、ERP 導入プロジェクトは「システムを入れること」自体が目的になるのではなく、「ビジネスの変革」という本来のゴールを見失わずに進めることができます。

4.2.2 機能要件（ERP モジュール、画面、帳票など）

機能要件は、新しい ERP システムが提供すべき**具体的な機能**の集合体です。これらを詳細に定義することで、設計・開発フェーズにおける手戻りを最小限に抑えます。機能要件は、ERP の特性に合わせて、**モジュール、画面、帳票**の3つの観点から定義することが一般的です。

1. ERP モジュールの要件定義:

ERP は、販売管理、生産管理、会計管理、人事管理といった複数の**モジュール**で構成され、これらのモジュール間でデータが連携することで企業全体の情報を一元管理します。

- **定義内容:**
 - **対象モジュール:** どの業務領域のモジュールを導入するかを明確にします（例：販売管理、在庫管理、財務会計）。自社の事業特性や経営目標に合わせて、必要なモジュールを厳選します。
 - **モジュール間の連携:** 各モジュール間でどのようなデータが、どのようなタイミングで連携されるかを定義します。例えば、「販売管理モジュールで受注が確定したら、在庫管理モジュールで自動的に在庫が引き当てられる」といった連携ルールを明記します。

2. 画面（UI/UX）の要件定義:

従業員が日々操作する画面の要件は、システムの使いやすさ（UX）に直結し、業務効率やデータの品質を大きく左右します。

- **定義内容:**
 - **画面項目:** 各画面で表示・入力が必要な項目（例：受注登録画面における「顧客名」「製品コード」「数量」など）をリストアップします。
 - **入力チェック:** 入力されるデータの妥当性をチェックするルール（例：数量は正の整数のみ、日付は未来の日付に設定不可）を定義します。

- **ナビゲーション**: 画面間の遷移や、よく使う機能へのアクセス方法を定義します。ユーザーの役割に応じて、必要な情報に簡単にたどり着けるような導線を設計することが重要です。
- **カスタマイズ要件**: 標準画面では対応できない、自社の独自の項目やロジックが必要な場合は、その内容を明確にします。ただし、可能な限り標準機能で代替できないか検討を重ねるべきです。

3. 帳票・レポートの要件定義:

業務の可視化や経営判断に必要な帳票・レポートも、重要な機能要件です。これにより、データがどのように活用されるかが明確になります。

- **定義内容**:
 - **帳票の種類**: どの業務で、どのような帳票（例：納品書、請求書、月次売上レポート）が必要かを特定します。
 - **レイアウトと項目**: 各帳票に表示すべき項目、レイアウト（例：ロゴの位置、罫線の有無）を具体的に定義します。
 - **出力条件**: 帳票がどのような条件で、どのような形式（例：PDF、Excel）で出力されるかを定義します。

国内製造業の事例では、既存システムが特定の顧客向けの特殊な帳票レイアウトを必要としていました。新 ERP 導入にあたり、この帳票要件を**必須**と定義しつつも、ベンダーと連携して、帳票印刷をカスタマイズするのではなく、帳票ツール（BI ツールなど）と連携して出力する方式を採用しました。これにより、ERP 本体のカスタマイズを最小限に抑え、将来的なバージョンアップへの影響を低減することに成功しました。この 3 つの観点から機能要件を詳細に定義することで、新システムの全体像がより明確になり、開発ベンダーとのコミュニケーションも円滑になります。

4.2.3 非機能要件（性能、可用性、拡張性、セキュリティなど）

非機能要件は、システムの「品質」や「制約」に関する要件であり、システムの安定稼働と長期的な運用を保証するために不可欠です。機能要件が「何をやるか」であるのに対し、非機能要件は「どのようにやるか」を定義します。

1. 性能（パフォーマンス）:

- **要件**:
 - **レスポンス時間**: 画面の応答時間や検索結果の表示時間（例：「主要な画面は 3 秒以内に表示されること」）。
 - **処理能力**: バッチ処理やレポート出力の時間（例：「月次バッチ処理は 2

時間以内に完了すること)。

- **考慮事項:**
 - 同時接続ユーザー数、データ量、ネットワーク環境などを考慮して、具体的な数値を設定します。

2. 可用性（アベイラビリティ）：

- **要件:**
 - **稼働時間:** システムが利用可能である時間帯（例：「24 時間 365 日稼働」）。
 - **稼働率:** システムの稼働率（例：「稼働率 99.9%以上」）。
 - **災害対策（DR）:** 災害発生時に事業継続を可能にするための対策（例：「DR サイトを構築し、RPO/RTO を定義する」）。
- **考慮事項:**
 - どの程度のシステム停止が許容されるか、事業継続計画（BCP）に基づいて要件を定めます。

3. 拡張性（スケーラビリティ）：

- **要件:**
 - **ユーザー数:** 将来的なユーザー数の増加に対応できること。
 - **データ量:** 過去データの蓄積や取引量の増加に対応できること。
 - **機能拡張:** 新規事業や法改正に伴う機能追加に柔軟に対応できること。

4. セキュリティ：

- **要件:**
 - **認証・認可:** ユーザーの ID 管理、職務に応じたアクセス権限設定など。
 - **データ保護:** データの暗号化、バックアップ、改ざん防止など。
 - **ログ管理:** 誰がいつ、何をしたかを記録する監査ログの仕組み。
- **考慮事項:**
 - 法規制（電子帳簿保存法など）や内部統制の要件を満たしているかを詳細に確認します。

非機能要件は、システムの基盤となる部分であり、不十分な定義は後々の大きなトラブルに繋がります。ベンダーと綿密に連携し、自社のビジネス特性やリスク許容度に基づいた要件を定義することが重要です。

国内製造業の事例では、既存システムが特定のオフコン上で稼働しており、特定の OS が保守切れとなりセキュリティ上のリスクが懸念されていました。新しいクラウド ERP の導入にあたり、非機能要件として「データの保存場所は日本国内に限定する」

「国際的なセキュリティ認証を取得していること」などを明記。これにより、コンプライアンスリスクを解消し、システムの信頼性を確保しました。

4.3 ERP アーキテクチャデザインの検討

新 ERP システムの要件定義を終えたら、次にその要件を技術的に実現するための**アーキテクチャデザイン**を検討します。これは、システムの全体像を描き、将来的な拡張性や運用性を左右する重要なプロセスです。

4.3.1 クラウド ERP、オンプレミス ERP の選択

ERP を導入するにあたり、最も重要な意思決定の一つが、システムを自社内に構築する**オンプレミス**か、クラウド上で利用する**クラウド**かの選択です。それぞれのメリットとデメリットを理解し、自社の事業戦略や IT 方針に合わせて最適な方式を決定する必要があります。

1. オンプレミス ERP:

- **概要:** サーバーやネットワーク機器を自社で保有・管理し、システムを自社データセンター内で運用する形態です。
- **メリット:**
 - **高いカスタマイズ性:** 独自の業務プロセスや商習慣に合わせて、自由にシステムをカスタマイズできます。
 - **セキュリティのコントロール:** 物理的な機器を含め、セキュリティ対策を自社の管理下で厳格に実施できます。
- **デメリット:**
 - **高額な初期費用:** サーバーやライセンス、構築費用など、初期投資が大きくなります。
 - **運用・保守の負担:** システムの監視、障害対応、バージョンアップなど、運用・保守のすべてを自社で行う必要があります。
 - **拡張性の限界:** 事業規模の拡大に伴うサーバー増設などに時間とコストがかかります。

2. クラウド ERP:

- **概要:** ベンダーが提供するクラウド環境上でシステムを利用する形態です。SaaS（Software as a Service）や PaaS（Platform as a Service）などの種類があります。
- **メリット:**
 - **低額な初期費用:** サーバーなどの物理的な資産が不要で、初期費用を抑

えられます。

- **運用・保守の負担軽減:** ベンダーがシステムの運用・保守を担うため、自社の IT 担当者の負担が大幅に軽減されます。
- **高い拡張性:** ユーザー数やデータ量の増加に応じて、柔軟にリソースを拡張できます。
- **常に最新の機能:** ベンダーが提供する自動バージョンアップにより、常に最新の機能を利用できます。
- **デメリット:**
 - **カスタマイズの制約:** 標準機能からの逸脱が難しく、独自の業務プロセスへの対応が限定される場合があります。
 - **セキュリティの外部依存:** セキュリティ対策はベンダーの提供範囲に依存します。

3. 選択のポイント:

- **事業戦略:**
 - **国内製造業の事例**では、海外拠点展開の迅速化が経営課題でした。この場合、地理的な制約なく利用でき、短期間での導入が可能なクラウド ERP が有効な選択肢となります。
- **IT 方針とコスト:**
 - IT 資産を最小限に抑え、運用コストを変動費化したい場合はクラウドが適しています。
 - 自社で厳格なセキュリティ要件や独自のシステム資産を維持したい場合はオンプレミスが適しています。

4.3.2 ERP と周辺システム（CRM, SCM など）の連携戦略

ERP は企業の基幹システムですが、その力を最大限に発揮するためには、**周辺システム**とのシームレスな連携が不可欠です。例えば、営業部門の顧客管理システム（CRM）や、物流・サプライチェーン管理システム（SCM）との連携が挙げられます。

1. 連携の重要性:

- **データの一元管理:** 各システムに散在するデータを ERP に集約し、全社で一貫したデータに基づいた意思決定を可能にします。
- **業務プロセスの自動化:** 例えば、CRM に入力された受注情報が自動で ERP に連携され、生産計画や在庫管理プロセスが自動的に開始される、といった業務効率化を実現します。
- **リアルタイム性の向上:** 必要な情報をリアルタイムで共有することで、迅速な

ビジネス判断や顧客対応を可能にします。

2. 連携戦略の策定:

- **連携対象の特定:** どの周辺システムと連携が必要か、その優先順位を明確にします。
- **連携方式の選択:**
 - **API 連携:** リアルタイム性が求められる場合（例：EC サイトからの受注データ）に採用します。
 - **バッチ連携:** 定期的なデータ同期で十分な場合（例：月次の会計データ）に採用します。
- **連携基盤の検討:** 複数のシステムと複雑に連携する場合、iPaaS（Integration Platform as a Service）のような連携基盤の導入を検討します。これにより、連携の管理を効率化し、システム全体の柔軟性を高めます。

国内製造業の事例では、販売管理の ERP と生産管理の MES（製造実行システム）が連携しておらず、生産指示のデータ転記に多くの工数がかかっていました。新 ERP 導入にあたり、API 連携を前提としたアーキテクチャを設計。ERP の受注情報に基づいて、MES にリアルタイムで生産指示を連携する仕組みを構築しました。これにより、生産計画の精度が向上し、手動入力によるミスも大幅に削減されました。

4.3.3 データプラットフォーム、AI/ML 基盤との連携

現代の ERP は、単なる業務処理システムではなく、企業データを活用した**経営戦略の推進基盤**へと進化しています。そのため、ERP が蓄積した膨大なデータを、**データプラットフォーム**や **AI/ML（機械学習）基盤**と連携させるアーキテクチャが重要になっています。

1. データプラットフォームとの連携:

- **目的:** ERP のトランザクションデータ（受注、売上、在庫など）を、データプラットフォーム（データウェアハウスやデータレイク）に統合し、BI ツールなどで多角的に分析できるようにします。
- **連携のメリット:**
 - **経営の可視化:** リアルタイムな売上動向、在庫状況、生産コストなどを可視化し、迅速な経営判断を支援します。
 - **データドリブンな意思決定:** 勘や経験に頼るのではなく、データに基づいた客観的な意思決定を可能にします。

2. AI/ML 基盤との連携:

- **目的:** ERP のデータを AI/ML 基盤で分析し、予測や最適化に活用します。
- **活用例:**
 - **需要予測:** 過去の販売データや市場データから、将来の需要を予測し、生産計画や在庫最適化に役立てます。
 - **品質予測:** 製造プロセスにおける各種データから、製品の品質を予測し、不良品の発生を未然に防ぎます。
 - **メンテナンス予測:** 機器の稼働データから、故障時期を予測し、予防保全を効率化します。

国内製造業の事例では、新しい ERP の導入に際し、ERP が生成する生産実績や品質データを、クラウド上のデータレイクに連携させるアーキテクチャを構築しました。このデータレイク上のデータを AI で分析することで、不良品の発生要因を特定し、品質改善につなげることに成功しました。このように、ERP と AI/ML 基盤を組み合わせることで、業務効率化だけでなく、企業の競争力向上という新たな価値を生み出しました。

4.4 ERP パッケージ選定と評価基準

ERP システムの要件定義とアーキテクチャデザインが固まったら、いよいよ具体的な **ERP パッケージの選定**に入ります。市場には多種多様な ERP パッケージが存在するため、自社の要件に最も合致する最適な製品を見極めることが、プロジェクトの成否を大きく左右します。

4.4.1 主要 ERP パッケージの比較検討

ERP パッケージは、その機能範囲、対象とする企業規模、業界特性などによって多岐にわたります。主要なパッケージを比較検討する際は、それぞれの特徴を理解することが重要です。

1. グローバル ERP パッケージ:

- **概要:** 世界中の大企業に導入実績があり、財務会計から SCM（サプライチェーンマネジメント）、CRM まで広範な機能を網羅しています。
- **代表的な製品:** SAP S/4HANA, Oracle Cloud ERP など。
- **特徴:**
 - **広範な機能:** あらゆる業界の業務プロセスに対応できる豊富な標準機能を備えています。
 - **グローバル対応:** 多言語、多通貨、各国の法規制に対応しており、海外拠点を持つ企業に適しています。
 - **導入期間とコスト:** 機能が多岐にわたるため、導入期間は長くなる傾向

があり、コストも高額になることが一般的です。

2. 国内 ERP パッケージ:

- **概要:** 日本特有の商習慣や法制度にきめ細かく対応しており、中堅・中小企業を主なターゲットとしています。
- **代表的な製品:** OBIC7, D3CRM など。
- **特徴:**
 - **日本商習慣への適応:** 日本独自の複雑な商慣習（例: 業界特有の歩引きやリベート計算）に標準機能で対応しています。
 - **サポート体制:** 国内ベンダーが提供するため、サポートや保守サービスが日本語で受けられ、迅速な対応が期待できます。
 - **コスト:** グローバル製品と比較して、導入コストやライセンス費用が比較的安価なものが多いです。

3. 業界特化型 ERP パッケージ:

- **概要:** 特定の業界（製造業、建設業、小売業など）に特化した機能を強化したパッケージです。
- **代表的な製品:**
 - **製造業向け:** 生産計画、原価管理、品質管理、IoT 連携などを強みとします。
 - **建設業向け:** プロジェクト管理、原価管理、見積もり、労務管理などを強みとします。
- **特徴:**
 - **業務適合性:** 業界特有の業務プロセスや専門用語に標準機能で対応しているため、カスタマイズが最小限で済みます。

各パッケージの特徴を理解した上で、自社の事業規模、グローバル展開の有無、業界特性などを考慮し、検討対象を絞り込んでいきます。

4.4.2 機能要件、非機能要件、コスト、ベンダーサポート、導入実績

ERP パッケージを評価する際には、多角的な視点から総合的に判断することが不可欠です。以下に示す 5 つの評価基準を設け、客観的に比較検討します。

1. 機能要件への適合性:

- **評価項目:** 策定した機能要件（第 4.2.2 項参照）に対し、パッケージの標準機能でどれだけ対応できるか。

- 評価ポイント:
 - Fit & Gap 分析: 要件と標準機能のギャップを詳細に分析し、カスタマイズが必要な部分を特定します。ギャップが大きいほど、導入コストやリスクが高まります。

2. 非機能要件への適合性:

- 評価項目: パッケージが非機能要件（第 4.2.3 項参照）をどれだけ満たしているか。
- 評価ポイント:
 - 性能: 大量のデータ処理や多数のユーザーが同時に利用する際のパフォーマンスは十分か。
 - 可用性: クラウドの場合、ベンダーの SLA（サービス品質保証）が自社の要件を満たしているか。
 - セキュリティ: データの暗号化、アクセス制御、認証方式などが、自社のセキュリティポリシーに準拠しているか。

3. コスト:

- 評価項目: 導入にかかる総費用（TCO: Total Cost of Ownership）を評価します。
- 評価ポイント:
 - 初期費用: ライセンス、サーバー、コンサルティング費用など。
 - ランニングコスト: 保守費用、ライセンス更新費用、クラウド利用料など。
 - 隠れたコスト: カスタマイズ費用、運用担当者の人件費、トレーニング費用など、見落としがちなコストも考慮に入れます。

4. ベンダーサポート体制:

- 評価項目: パッケージを導入・運用するベンダーのサポート体制は十分か。
- 評価ポイント:
 - 導入コンサルタントの専門性: 自社の業界知識や ERP の専門知識が豊富か。
 - サポートサービス: 障害発生時の対応時間、サポートの範囲、日本語対応の可否など。
 - パートナーエコシステム: 導入ベンダーだけでなく、周辺システム連携や追加開発を支援するパートナーが豊富か。

5. 導入実績:

- **評価項目:** ベンダーが提供するパッケージの導入実績と、自社と同業種・同規模の企業での成功事例の有無。
- **評価ポイント:**
 - **事例の確認:** 類似企業の導入事例を参考にし、実際にどのような課題を解決し、どのような効果を得られたかを確認します。
 - **ユーザーの声:** 可能であれば、実際にシステムを利用している企業にヒアリングを実施し、生の声を聞くことも有効です。

4.4.3 PoC（概念実証）の活用

理論上の評価だけでなく、実際の環境でパッケージを試す PoC（Proof of Concept: 概念実証）は、選定の精度を飛躍的に高める有効な手段です。PoC は、システムの一部を実際に構築し、要件が技術的に実現可能かを検証します。

1. PoC の目的とメリット:

- **要件の実現可能性検証:** 策定した要件が、実際に候補となる ERP パッケージで実現可能かを検証します。特に、カスタマイズが必要な部分や、周辺システムとの連携がスムーズに行えるかを確認します。
- **ユーザーエクスペリエンスの体感:** 実際の画面を操作することで、従業員が「使いやすいか」「業務プロセスに合っているか」を肌で感じることができます。これにより、導入後の利用定着率を予測できます。
- **ベンダーの評価:** PoC を通じて、ベンダーの技術力や対応力、コミュニケーション能力などを直接評価できます。

2. PoC の進め方:

- **ステップ 1: 目的と範囲の明確化:**
 - PoC の目的を「特定の複雑な業務プロセスの実現可能性を検証する」といった具体的なものに絞り込みます。
 - 全機能を検証するのではなく、最も重要な機能やリスクが高い部分に焦点を当てます。
- **ステップ 2: 期間と予算の設定:**
 - PoC は通常、1～3 ヶ月程度の短期間で実施します。無駄なコストをかけないように、あらかじめ期間と予算を設定します。
- **ステップ 3: 評価基準の策定:**
 - PoC の結果を客観的に評価できるよう、事前に評価基準を策定します。
例: 「特定の業務プロセスがエラーなく完了すること」「画面の応答時間が 3 秒以内であること」など。

- **ステップ 4: 実施と評価:**

- 候補ベンダーに PoC を依頼し、実際のデータや業務プロセスを再現して検証します。
- 実施後、策定した評価基準に基づいて結果を客観的に評価します。

国内製造業の事例では、海外工場とのリアルタイムなデータ連携が必須要件でした。複数の ERP 候補の中から、クラウド ERP を選定するにあたり、特定の海外拠点と連携する PoC を実施。これにより、実際にデータ連携の速度や安定性を確認し、非機能要件を満たすことを検証できました。PoC の結果、当初は懸念されていた通信遅延の問題も解消できることが分かり、安心してパッケージを決定できました。

4.5 データ戦略と ERP におけるデータモデルの検討

ERP システムは、企業のあらゆる業務データを一元的に管理する「データの器」です。そのため、ERP 導入プロジェクトでは、システム設計と並行して、そのデータをどのように扱い、活用していくかという「データ戦略」を明確にすることが不可欠です。

4.5.1 ERP データ連携基盤の構想

新しい ERP は、単体で完結するシステムではありません。企業の IT エコシステム全体の中で、周辺システムや外部サービスとシームレスにデータを連携させる必要があります。この連携を効率的かつ柔軟に行うための基盤が、**ERP データ連携基盤**です。

1. データ連携の課題と連携基盤の必要性:

- **課題:**
 - 既存のシステムは、システムごとに連携方式が異なり、複雑な「Spaghetti Architecture (スパゲッティアーキテクチャ)」になりがちです。
 - 新たなシステムを追加するたびに、連携のための開発コストと時間がかかります。
 - 連携部分に問題が発生した場合、原因特定に時間がかかります。
- **連携基盤の必要性:**
 - **標準化:** 連携方式を統一し、システム間の依存関係を整理します。
 - **効率化:** 連携ロジックを再利用可能にし、開発を効率化します。
 - **一元管理:** 連携状況を可視化し、障害発生時の原因特定を迅速化します。

2. データ連携基盤の構想:

- **連携方式の選択:**

- **API 連携（リアルタイム）**：受注情報や在庫情報など、リアルタイム性が求められる重要なデータ連携に採用します。
- **ファイル連携（バッチ）**：月次の会計データや日次の売上データなど、定期的な同期で十分なデータ連携に採用します。
- **基盤ツールの選定**：
 - 複数のシステムと複雑に連携する場合、iPaaS（Integration Platform as a Service）の導入を検討します。これにより、連携の設計・開発・管理をクラウド上で一元的に行えます。
- **データセキュリティ**：
 - 連携基盤においても、データの暗号化、アクセス制御、ログ監視など、厳格なセキュリティ要件を定義します。

国内製造業の事例では、ERP と EC サイト、WMS（倉庫管理システム）が複雑に連携しており、システムごとに異なる連携方式が採用されていました。新 ERP 導入にあたり、iPaaS を導入して連携基盤を再構築。EC サイトからの受注データは API でリアルタイムに ERP に連携し、ERP から WMS への出荷指示はバッチ連携で行うという、連携方式の標準化と整理を実現しました。これにより、連携部分のメンテナンス性が大幅に向上し、新たなサービス追加への対応も迅速化しました。

4.5.2 マスターデータ管理（MDM）と ERP への統合

ERP が生成するトランザクションデータ（取引データ）の品質は、その前提となるマスターデータ（顧客、製品、社員など）の品質に大きく依存します。そのため、ERP 導入と並行して、企業全体でマスターデータを一元管理するマスターデータ管理（MDM: Master Data Management）の仕組みを構築することが重要です。

1. MDM の課題と ERP 統合のメリット

- **課題**：
 - 複数のシステムでマスターデータが重複・不整合を起こし、データ分析や経営判断の信頼性を損なう。
 - マスターデータの登録・変更プロセスが統一されておらず、手作業によるミスが発生する。
- **ERP 統合のメリット**：
 - **単一真実（Single Source of Truth）**：新しい ERP を、企業全体のマスターデータの「単一真実」として位置づけます。他のシステムは、ERP からマスターデータを参照する仕組みを構築します。
 - **データガバナンスの強化**：マスターデータの登録・変更プロセスを ERP

の承認ワークフローに統合し、誰がどのような権限でデータを操作したかを記録します。

- **データ品質の向上:** 入力時のチェック機能や重複データの自動排除機能などを活用し、マスターデータの品質を向上させます。

2. MDM と ERP 統合の進め方:

- **ステップ 1: マスターデータの定義:** 顧客、製品、社員など、企業全体で共通して使用するマスターデータの種類と、その定義項目を標準化します。
- **ステップ 2: データオーナーシップの明確化:** 各マスターデータについて、どの部門が最終的な登録・変更の責任を持つか（データオーナー）を明確に定めます。
- **ステップ 3: ERP へのデータ統合:**
 - 移行対象のマスターデータを精査し、不備や重複を修正する**データクレンジング**を実施します。
 - クレンジング後のデータを ERP に統合します。
- **ステップ 4: 運用ルールの策定:** ERP 稼働後のマスターデータの登録・変更ルール、承認プロセス、監査体制などを定めます。

国内製造業の事例では、既存システムでは生産管理システムと販売管理システムで異なる製品マスターが使用されていました。新 ERP 導入にあたり、製品マスターの定義を統一し、ERP をマスターデータの単一真実と位置づけました。さらに、生産管理システムは新 ERP の製品マスターを API で参照する仕組みを構築。これにより、データの不整合が解消され、生産計画と販売計画の連携がスムーズになりました。

4.5.3 分析用データモデルの検討

ERP に蓄積される膨大なデータは、単に業務を処理するだけでなく、ビジネスの現状を把握し、未来を予測するための重要な資産です。このデータを効果的に活用するためには、**分析用データモデル**をあらかじめ検討しておくことが重要です。

1. 分析用データモデルの必要性:

- **目的:** ERP のトランザクションデータは、日々の業務処理に適した形式（正規化された構造）で格納されていますが、そのままでは複雑な集計や分析には向きません。分析用データモデルは、このデータを分析しやすい形式（非正規化されたスター型スキーマなど）に変換し、データウェアハウスなどに格納する際のモデルを指します。
- **メリット:**

- **分析効率の向上:** 複雑な SQL（データベース問い合わせ言語）を使わずとも、BI ツールなどで簡単に集計や分析ができるようになります。
- **意思決定の迅速化:** 経営層や管理者が、必要な情報を迅速に抽出し、データドリブンな意思決定を行えるようになります。

2. データモデルの検討ポイント:

- **ビジネス要件のヒアリング:** 経営層や各業務部門に「どのような分析をしたいか」「どのような指標を追いたいのか」をヒアリングし、分析の目的を明確にします。
- **データウェアハウスの構築:** ERP データと、周辺システム（CRM など）のデータを統合するためのデータウェアハウス（DWH）やデータレイクの構築を検討します。
- **BI (Business Intelligence) ツールの選定:** 分析用データモデルで整備されたデータを可視化するための BI ツールの選定を行います。
- **レポートとダッシュボードの設計:** リアルタイムな経営ダッシュボードや、月次の定型レポートなど、具体的なアウトプットを設計します。

国内製造業の事例では、新しい ERP 導入と並行して、ERP の販売実績データを自動でクラウド上のデータウェアハウスに連携する仕組みを構築しました。データウェアハウスでは、製品マスターや顧客マスターと連携した分析用データモデルを定義。これにより、BI ツールを使って「地域別の売上推移」「製品ごとの利益率」といった分析を簡単に行えるようになり、経営層の迅速な意思決定に貢献しました。