

テーマ2 業務を効率化する情報処理

1

複雑な情報処理の改善

データ整理・変換、PHP+DB活用

高度な情報処理の実現

外部サービス連携、XML/SOAP 対応

具体的なデータ処理の悩みに応える

実務型の内容です

【スライド1】

業務を効率化する情報処理
複雑処理／高度処理の改善

2

****業務を効率化する情報処理****

～複雑処理／高度処理の改善～

日々の業務で発生するデータ処理。

手作業で時間をかけている作業の多くは、

適切なツールと方法で大幅に効率化できる。

実務に即した情報処理の改善方法をご紹介します。

【スライド2】 よくあるデータ処理の悩み

- * Excel が重くなる
- * 毎回手作業でコピー
- * データ形式がバラバラ
- * 外部システムとデータが連携しない

3

現場の声から見える課題

* **Excelが重くなる**

- 数万行のデータで動作が遅い
- ファイルが開かない、保存に時間がかかる
- 計算式が多すぎてエラーが出る

* **毎回手作業でコピー**

- 同じ作業を繰り返している
- ミスが発生しやすい
- 時間がもったいない

* **データ形式がバラバラ**

- CSV、Excel、PDF、紙…
- 統合するのに手間がかかる
- 集計が困難

* **外部システムとデータが連携しない**

- 手動でデータ移動
- 二重入力が発生

【スライド3】 複雑処理が生まれる原因

- * データの構造が不統一
- * XML → CSV → Excel の変換が多い
- * サイズが大きい
- * プログラムで処理できる形になっていない

4

****なぜデータ処理が複雑になるのか****

*** **データの構造が不統一** ***

- 人によって入力方法が違う
- 「鈴木」「すずき」「スズキ」などの表記ゆれ
- 日付形式がバラバラ（2025/1/1、令和7年1月1日など）

*** **XML → CSV → Excel の変換が多い** ***

- システム間のデータ受け渡しで形式変換
- 変換のたびに手作業が発生
- データの欠落や文字化けのリスク

*** **サイズが大きい** ***

- Excelの限界（約100万行）を超える
- メモリ不足でフリーズ
- 処理に数十分かかる

*** **プログラムで処理できる形になっていない** ***

- セル結合、余計な装飾が多い

1セルに複数情報が入っている

【スライド4】 整理の基本

- * 1行 = 1件
- * 同じ列は同じ型
- * ID / 日付 / 種別の整理
- * フォルダ構成を統一

5

データ処理の大原則

* **1行 = 1件**

- 1つのデータは1行に収める
- セル結合は使わない
- 空白行を入れない

* **同じ列は同じ型**

- 数値の列に文字を混ぜない
- 日付は統一形式で（例：2025-01-01）
- 単位は別列に分ける（例：「金額」列と「単位（円）」列

* **ID / 日付 / 種別の整理**

- 一意のIDを付ける（連番など）
- 日付はソート可能な形式で
- カテゴリーは統一された値を使う（プルダウン推奨）

* **フォルダ構成を統一**

- 命名規則を決める（例：20250101_売上データ.xlsx）
- 年度別、月別などで整理

【スライド5】 PHP + MySQL の活用

- * データ処理を自動化
- * 大量データ処理が高速
- * Web ベース管理が可能

6

****大量データ処理の強い味方****

*** **データ処理を自動化****

- 定型処理をプログラム化
- ボタン一つで完了
- 夜間に自動実行も可能

*** **大量データ処理が高速****

- 数十万件のデータも数秒で処理
- Excelでは不可能な規模に対応
- 複雑な条件での抽出・集計が可能

*** **Webベース管理が可能****

- ブラウザからアクセス
- 場所を選ばず利用できる
- 複数人での同時作業も可能
- スマホからも確認可能

****なぜPHP + MySQLか****

【スライド6】 外部連携（SOAP & REST）

- * 他サービスとデータやりとり
- * 申請、集計、地図情報など
- * 自動連携で人的ミス削減

7

****システム間の自動連携****

*** **他サービスとデータやりとり****

- API（Application Programming Interface）を活用
- 自動でデータを送受信
- リアルタイム連携が可能

*** **申請、集計、地図情報など****

- 申請フォームからデータベースへ自動登録
- 複数システムの情報を一元集計
- 住所から地図表示を自動生成
- 在庫情報の自動同期

*** **自動連携で人的ミス削減****

- 二重入力がなくなる
- 転記ミスがゼロに
- データの整合性が保たれる

****SOAP vs REST****

【スライド7】 高度処理の成功事例

- * 毎月 4 時間かけていた集計が 2 分に
- * 報告書の自動生成
- * CSV 自動ダウンロード+加工

8

実際の改善効果

* **毎月4時間かけていた集計が2分に** *

- 【Before】 複数のExcelファイルを手作業で集計
- 【After】 データベースから自動集計、レポート自動生成
- 【効果】 年間48時間の削減、ミスゼロ

* **報告書の自動生成** *

- 【Before】 テンプレートに手作業でデータ入力、30分/件
- 【After】 データベースからWordファイル自動生成
- 【効果】 1分/件、レイアウト崩れなし

* **CSV自動ダウンロード+加工** *

- 【Before】 毎日3つのサイトからCSVダウンロード、統合
- 【After】 自動ダウンロード→統合→必要項目抽出
- 【効果】 毎日20分の作業が完全自動化

その他の事例

- イベント参加申込の自動集計・名簿作成

【スライド8】トラブルを防ぐデータ設計

- * 同じ意味の項目は統一
- * ID 採番ルール
- * データの持ち方を「正規化」

9

****長期運用を見据えた設計****

*** **同じ意味の項目は統一** ***

- 「氏名」「お名前」「名前」→「氏名」に統一
- マスタデータを作成 - プルダウン入力で表記ゆれ防止

*** **ID採番ルール** ***

- 連番、日付+連番など明確なルール
- 一意性を保証
- 手入力ではなく自動採番が理想
- 例：2025010001（年+月+連番4桁）

*** **データの持ち方を「正規化」** ***

- 重複情報を排除
- 更新時の不整合を防ぐ
- データベース設計の基本原則
- 例：顧客情報と注文情報を別テーブルに分離

****正規化の具体例****

【スライド9】 業務フロー改善ステップ

1. 現状把握
2. 問題箇所の抽出
3. 自動化できる箇所を選定
4. ツール/スクリプト化
5. 運用ルール化

10

****段階的な改善アプローチ****

****1. 現状把握****

- どんな作業にどれくらい時間がかかっているか記録
- ボトルネックを特定
- 関係者へのヒアリング

****2. 問題箇所の抽出****

- 手作業が多い箇所
- ミスが発生しやすい箇所
- 時間がかかりすぎている箇所
- 優先順位をつける（効果×実現性）

****3. 自動化できる箇所を選定****

- 定型作業は自動化しやすい
- データ変換、集計、転記など
- まずは小さく始める
- すぐに効果が出るものから

****4. ツール/スクリプト化****

- Excelマクロ、Python、PHP など

【スライド10】 データ処理チェックリス

- * データ形式が統一されている
- * 項目名が一貫している
- * 自動化できる作業は特定済み
- * 外部連携が必要か整理した
- * フォルダ構造にルールがある
- * DB 保守体制が明確
- * バックアップの仕組みがある

【スライド11】 個別相談の例

- * XML の扱い
- * DB 連携の方法
- * 自動化スクリプトの提案