**システム開発の進め方**

**1)ユーザー要求の把握**

a ユーザーがシステム化したいと考える要因について(基本構想)

① 事務量の増加

手作業の部分の作業量がどんどん増加して人手では既に許容範囲を超えてきている。

② 法令の改正

消費税とかプライス表示の表示改正など法令が変わったため作業量が一時的に多くなった

③ 経営方針の変更

人件費の削減・販路の拡大・多角経営への変更など

④ システムのレベルアップ

ダウンサイジシグなどで古いマシンからパソコンなどにシステムを移行せざるを得なくなった

b 予備調査(ユーザー聞き取り)

基本構想の完成後、予備調査を実施して新システムが当該事業体に及ぼす影響についての初期

評価を行うのに十分な情報を収集する。

【調査内容について】

① 現状作業の把握

実際に手作業又は現システムの内容を把握する。

② 問題点の洗い出し

システム化にあたり現状の問題点を把握する

③ 予想費用

ユーザーが考えている予算の把握

作業1

現状作業の洗い出し

実際におこなわれているであろう作業の概略を詳細に書き出してみる

文章でも図解でもどちらでもよい

現状作業概略図

作業2

現状作業の問題点を洗い出して発生の原因を考える。

現状問題分析調査表の作成

c 予備分析及び評価

予備調査をもとにシステムヘの検討を行う。現状の作業でもっと合理的に作業を変更できない

か、又省略できる点を分析する。そのためには以下の点をユーザーと討議する。

① 開発事項の確認と定義

実際にシステム化する範囲を確認しシステム化への定義をおこなう

② プロジェクトの範囲の決定

①よリシステム化のプロジェクトを発足して基本設計の範囲を決定する

③ 予想される効果・費用節約額及び想定される代替的システム解決策

（開発依頼の却下を含めて)を採用した場合の追加費用の見積りを行う）

**2)新システム化への構想**

a 基本設計案の作成

予備調査で把握した内容を元に、ドキュメント化する前の計画案を作成する。

① 基本設計の処理内容の整理

ユーザーから出された要求を元に新システムの流れを考える

② システム概念図

INPUTとOUTPUTの概略を書き出す

入力画面・処理・出力帳票・出カデータ等を書き出す

③ フィールドの洗い出し

システムに必要な項目を洗い出す

現状で使用している各種帳票にある項目を書き出す。

④ コード設計

システムで必要なコードを発い出す

区分・フラグなど決定している内容は洗い出す。

作業1

システム概要図を書く

作業2

新システムのDFDを書く

**3)コード設計**

システム開発をする上でコード設計は欠かせないものであり、コードを設計する上で考えなければならないのは以下の通りである。

①取り扱うデータの予測件数

a 区分のような既に有限で決まっているものについては最大数の桁数で決める

b マスタでも、日々増加するものについては前何桁を特定の区分にする

例えば商品マスタ

商品件数がメーカー毎に増加する場合、前何桁かをメーカーコードに設定してお

く

② コード設計の注意点

a ながながとしたコードはプログラムでの取り扱いが不便

b コードに意味を持たせすぎない

c 併用できるものは併用する

ひとつのコードを使いまわすことを考える

例えば発注・仕入れの時発注コード・仕入れコードといったようにそれぞれの処理

ごとにコードを設定するのではなく、発注コードを使いまわす

d 複数のキーで一つのキーにする

明細データの場合はキーを複数でユニ―クになるように設定する

例えば顧客毎の売上明細なら

販売日付・電話審号・商品番号でユニークにする

e電話番号などは、主キーとして使用しない

f キーのタイプはテキストタイブにする。

処理の途中で判断をするとき便利

③ 画面設計及び帳票設計をした後で出てくる項目もあるので、そのときにコード設計書に必す

追記しておく

作業3

コード設計書を書く

作業4

入出力関連図を書く

**4) 画面設計**

ユーザーからの聞き取りで現状の作業を洗い出しているので、必要な画面のみをレイアウト化する。

(注意点)

① マスタファイルヘのメンテナンス画面については、登録・修正・削除を一つの画面でできる

ようにしておく

② データの入力画面(伝票入力画面など)のように一度に大量のデータを入力する画面は、マウ

ス操作は使わない。すべてキーボートで入力できるようにしておく

③ 画面入力は左から右、上から下に順次関連項目を入力できるようこしておく

④ ガイダンスの項目名・説明は必要最低限にとどめておく。

もし長々と説明するときは、別にポップアップメニューなどでヘルプを表示できるようにし

ておく

⑤ エラーメッセージについてはメッセージ番号を決め、すべての画面で共通のメッセージを使

用するようにする

⑥ 画面の背景色と前景色については原色を避け、ガイダンスは意味なく赤を使用しない

⑦ 文字の大きさなどは統一する（11ポイントなど）

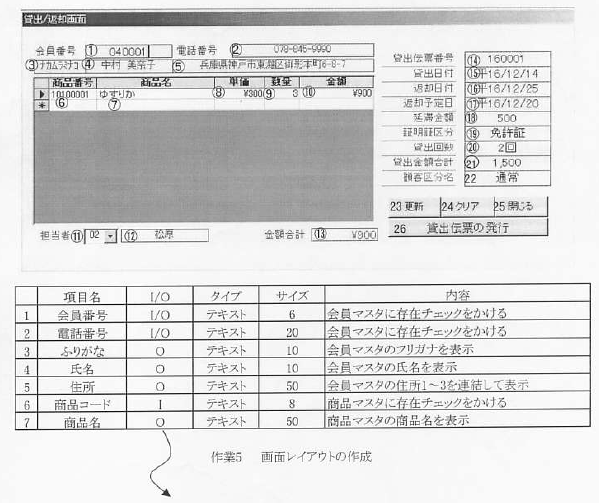
⑧ 画面の大きさは一画面で表示できるようにし、項目が多い場合はワンスクロールで表示でき

るようにする（解像度なるべく統一する）

⑨ 更新・終了・クリアボタンなどはどの画面も同じ場所に設置する

(書き方)

画面レイアウトだけでは、どの項目が入力又は出力項目なのかわからないし、また編集方法がわからないので、画面の項目に番号を振って説明書を別途作成する。



作業5

画面仕様書を書く（画面レイアウトの作成）

**5)帳票設計**

ユーザーからの開き取りで現状の作業を洗い出しているので、必要な帳票をレイアウト化する。画面レイアウトと同じく帳票設計も番号を付与して説明文を書く

(注法点〉

①伝票類については既存のものをそのまま使用するかどうかの検討が必要

現状を使用する場合は、データのみの出力になるので注意

② 帳票の場合、連帳か単票かを明示する

③ 行数を何行で改ベージするか、又合計行についても明示する

④ 帳票の表示に際し、何順で表示するかソートキーを明示する

⑤ 表題については、作成日付・ページ番号を表示する

⑥コードの表示について

営業分析などで使用するものに関しては、各種コードは必す表示する

顧客に送付する資料については、会員番号以外については表示しない

⑦帳票出力のタイミングを明示する

作業6

各種帳票レイアウトの作成

**6)ファイル設計・データベース設計**

予め入出力設計で決めていたファイルの分類と画面・帳票のレイアウトより各種ファイルの項目を確定する。

(注意点)

①主キーは必ずテキストタイプにする

数値だと、けた数が変化するため扱いづらい

② 項目名はなるべく短くシンプルにする

③ 同じ項目を別々のファイルで定義するときは同じ項目名に統一する

④ サイズは必ずコード設計で定義した長さにする。

⑤ 各マスタファイル・累積ファイルには必ず以下の項目を入れる

登録日付 レコードを登録した日

更新日付 データの内容を変更した日付

削除フラグ レコートを削除したらTrueにする

削除日付 レコードを削除した日付

⑥ 各明細ファイル・一次ファイルについては上記の項目はつけない

⑦ ファイル同士のリレーション図を書く

特にマスタより項目を持ってきたり、マスタに項目を追加するファイルについては、リレー

ションをはっきりと書いておく(図でもよい)

作業7

ファイル設計書を書く