## 要件定義書とは？

要件定義書はシステム開発を行う際に、クライアント（システム利用者）の要件をドキュメント化した文書になります。システムには詳しくない依頼者であっても、その内容を読むことで、期待したシステムが開発されることを確認できる必要があります。また、開発側ではこの後のフェーズである基本設計、詳細設計などを行う上でのベースとして、開発メンバーが情報を共有できることが期待されます。

## 要件定義書の目的

要件定義書はシステム開発の依頼者、開発者の間で齟齬をなくし、情報が適切に共有されていることを確認するために作成されます。現状を確認できる内容に加えて、システムに期待される効果（要件）についても記載されます。また、予算やスケジュールも明記することで、開発フェーズにおいて発生した課題に対して、予算と日程のいずれを優先するかなどを検討する際にも利用できます。

## 要件定義書（サンプル）

## システム名： 経費登録自動化システム

## 目的

この要件定義書は、経費登録を自動化するためのシステム要件について定義したものである。このシステムは、月末に行われる経費を手作業から自動化する事で、経費処理の効率化を図ることを目的としている。

## 業務要件

### 現状のフロー

現状のフローは次のようになっています。

1. 営業担当者が経費精算を依頼するため、営業事務担当者に手続きを行う。
2. 営業事務担当者は経費精算書を元に経費登録システムに手動で登録する。
3. 営業事務担当者は経費一覧を参照して、Excelで一覧表を作成する。
4. 営業事務担当者は一覧表をメールに添付して、営業主任に送信する。

### 構築後のフロー

システム構築後、フローは次のようになります。

1. 営業担当者が経費精算を依頼するため、営業事務担当者に手続きを行う。
2. 営業事務担当者は経費精算書をデスクトップに配置する。
3. 経費精算自動化システムを起動して、経費登録システムに自動的に登録する。
4. 自動処理後、一覧表を添付したメールが営業主任に送信される。

### 利用者一覧

* 経理担当者

### 規模

2人月（エンジニア1人×2）の予定です。

### 帳票

本システムでは以下のメールおよび帳票が出力されます。

•経費一覧送信メール

•経費一覧Excelファイル

•経費登録ファイル

機能要件：

1.経費が自動的に経費精算システムに登録される

2.登録された経費情報一覧をエクセルに書き込みメールに添付して送信する

非機能要件：

過去システムとの互換性は維持されます。経費登録自動化システムは経費登録システムに変更があった際でも改修無しで動作を保証致します。

-------------------------

要件定義書（要求仕様書）を元に業務フローを記載し視覚化していきます

機能要件とは、システムや製品がどのような機能を持っている必要があるかを示す要件です。具体的には、システムがどのような操作や機能を提供するか、どのような処理を行うか、どのような入出力を受け付けるかなどが含まれます。例えば、検索機能や注文機能、ログイン機能などが機能要件の一例です。

非機能要件は、システムや製品がどのような品質特性を持っている必要があるかを示す要件です。具体的には、システムがどのような性能を持っているか、どのような信頼性や安全性を持っているか、どのような規模や拡張性があるかなどが含まれます。例えば、システムの応答速度、セキュリティレベル、ユーザビリティ、保守性などが非機能要件の一例です。

簡単に言えば、機能要件はシステムが何をするかを示し、非機能要件はシステムがどのような性能や品質を持っているかを示します。両方の要件を考慮することで、ユーザーがシステムを使いやすく、運用者がシステムを維持しやすいシステムを開発することができます。

業務フロー図の書き方

業務フロー図は直観的に業務の全体像と流れを把握できるため、業務効率化を目的とした現状業務の洗い出しや業務引継ぎなどを目的として作図されることが多いです。

しかしながら、初めて業務フロー図を作成される方にとっては作成することにハードルを感じてしまうかもしれません。

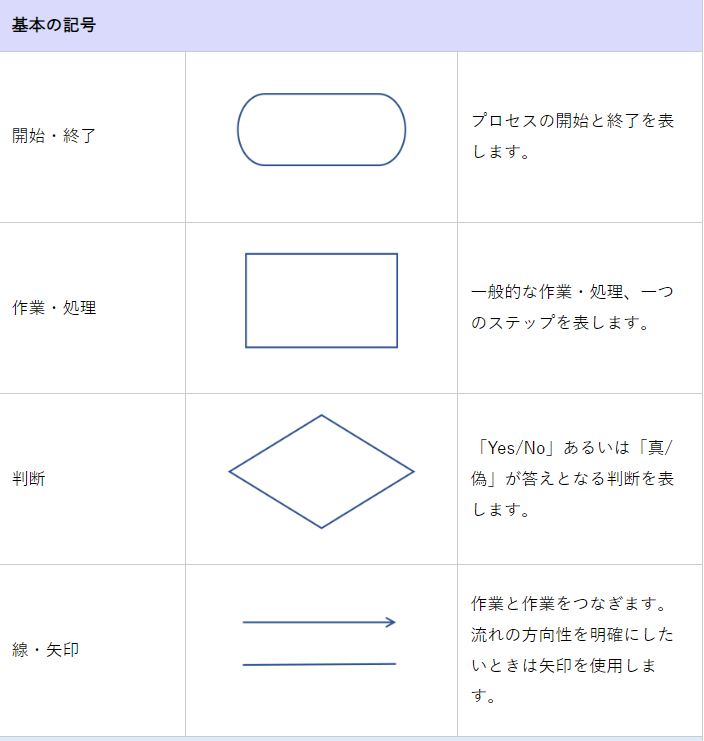
本記事では業務フロー図の作成方法について、「初めての方にもできるだけわかりやすく」解説していきます。

業務フロー図に使われる記号

業務フロー図に一般的に使われる記号を下図に表しました。

業務の流れのみであれば基本の記号だけで表現することが可能です。

必要に応じてサブの記号を追加し、目的に合った業務フロー図を作成しましょう。



図形

自動的に生成された説明

**業務フロー図を書く手順**

業務フローは下記の手順で行います。

### １．業務フロー図の目的を定義する

業務フローを作成する目的は何でしょうか。システム導入のために作図する場合と、業務引継ぎのために作図する場合ではどこまでを範囲とするのか、どこまで詳細に表現するのかが変わってきます。わかりやすい業務フロー図というのは、目的に沿ったものであることが前提になります。【今回の導入の目的：経費登録の作業時間短縮のため】

### ２．記載する部門と担当者を明確にする

業務フロー図に記載する部門や担当者を洗い出します。社内だけでなく、顧客や仕入れ先など業務に関わる全ての関係者を洗い出します。【担当部署を記入出来ます】

### ３．担当者へ作業についてヒアリングを行う

洗い出した部門の担当者にヒアリングを行い、手順や作業を把握します。  
ヒアリングする際は聞き出したい項目が記載されているヒアリングシートを使いながら行うと、部門ごとで偏りのない均一な情報を収集できます。横軸に作業項目、縦軸に時系列に作業を記載します。ヒアリングシートは実際に担当者に通常通りに作業を進めてもらいながら、ヒアリング担当者が記載していきます。その際、各手順の意味や背景、改善したいことなども聞き取りし、ヒアリングシートに追記します。【現在の業務改善プロセスが記入出来ます】

### ４．業務フロー図に記載する作業を確定する

ヒアリングした内容を業務フロー図に記載する必要があるかどうかを振り分けし、必要な作業だけを時系列に業務フロー図に記載します。【業務フローに処理などの記号が記載出来ます】

### **５．作業を線でつなぐ**

業務フロー図に記載した作業を手順がわかるように線でつないでいきます。

【現状の業務フローが記載出来ます】

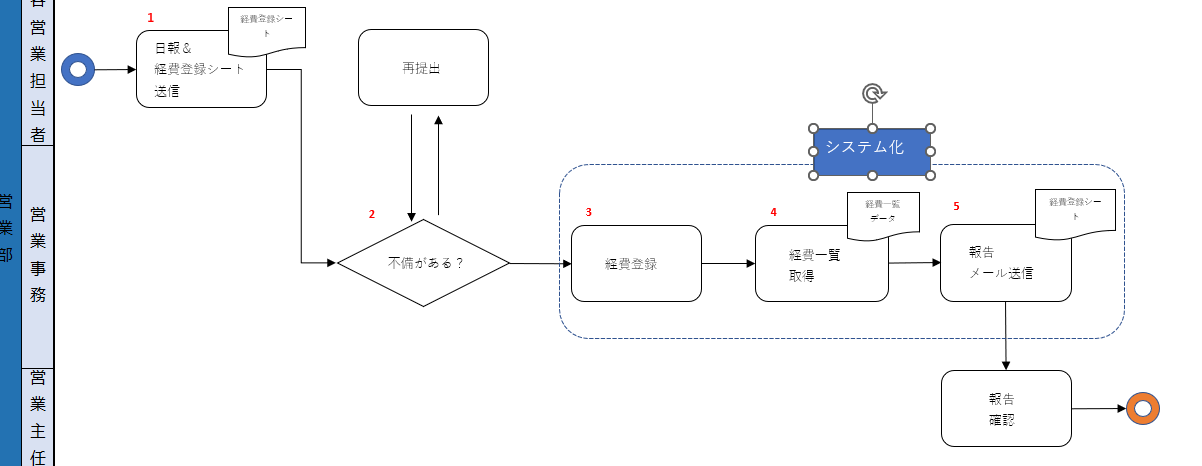
下記が現行の作業内容です。

ダイアグラム

自動的に生成された説明

ここから、システム化できそうな場所を洗い出します。

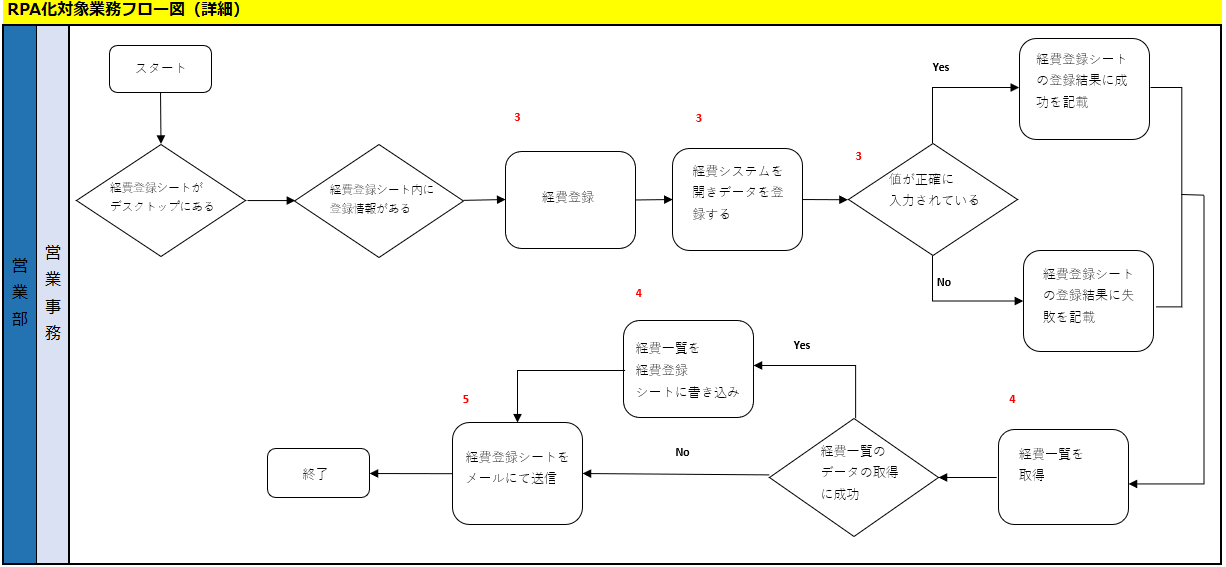
現行業務プロセスを見直したところ、経費登録作業,経費一覧を取得,営業主任にメールを送信の部分がシステム化出来そうです。



ここから、システム化できそうな箇所をプログラムレベルで細分化してフローチャートを作成してみましょう。

複雑ですがプログラムを作成するにあたりフローチャートをを用いて流れを作成していきます。

下記のような最初から大きなシステムではなく、簡単な物から作成して慣れていきましょう。



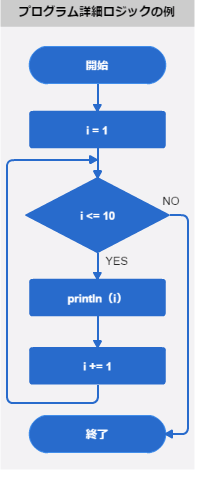
フローチャートとは？

フローチャートと業務フローは、どちらもプロセスを視覚的に表現するためのツールですが、異なる目的で使用されます。

フローチャートは、プロセスのステップや決定の流れを図解化するために使用されます。一般的に、フローチャートはより高度なプログラムやシステムを設計するために使用され、プログラマーやエンジニアが使用することが一般的です。フローチャートには、各ステップの名前、機能、決定点、および結果が含まれることがあります。

一方、業務フローは、企業や組織がどのように業務を実行しているかを示すために使用されます。一般的に、業務フローは、業務プロセスの改善や再設計を行うために使用され、マネージャーやプロセス改善チームが使用することが一般的です。業務フローには、各ステップの名前、担当者、必要な情報、およびプロセスの進行状況が含まれることがあります。

つまり、フローチャートはより具体的な技術的な設計に使用され、業務フローはより広範な業務プロセスを把握するための管理的なツールとして使用されます。



2. フローチャートを書く３つの目的

IT企業では、若手プログラマーは必ずフローチャートを書くよう指導を受けますが、そもそもなぜフローチャートを作るのか「目的」を理解してない人も多いかと思います。まずは目的からしっかり理解する事が重要です。

2.1. プログラム構造の整理

１つめの目的は、プログラム構造を整理するためです。プログラムは、書き方によって処理スピードや可読性に大きな影響を与えます。実際にプログラミングをする前に、最適な設計がされていることが望ましいです。

熟練プログラマーは、フローチャートを書かなくても最適な書き方が頭に入っていますが、若手プログラマーはそうはいきません。しっかりフローチャートを書いて、先輩にレビューしてもらいましょう。

2.2. プログラミング速度を上げるため（効率化）

2つめの目的は、プログラミングの速度を上げることです。しっかり設計ができていれば、その通りに効率的にプログラムを書くことができます。設計ができていなければ、後々修正が多く発生し余分に時間が掛かかってしまいます。若手プログラマーは、プログラミングの速度を上げるためにもフローチャートを書きましょう。

2.3. プログラム品質の向上

フローチャートを書くことでプログラムの全体像が明確になり、設計漏れ・やバグを減らすことができます。つまり、プログラムの品質が上がります。フローチャートを使って他者と事前レビューをすることで、更に品質を上げることもできます。

この章では、フローチャートを書く３つの目的について解説しました。優秀なプログラマーの領域に到達するには、フローチャートをたくさん書いて基本を頭にインプットすることが一歩です。若手プログラマーは積極的に活用して下さい。



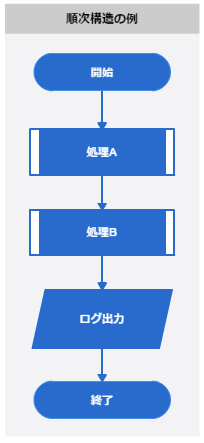
グラフィカル ユーザー インターフェイス, テキスト, アプリケーション

自動的に生成された説明

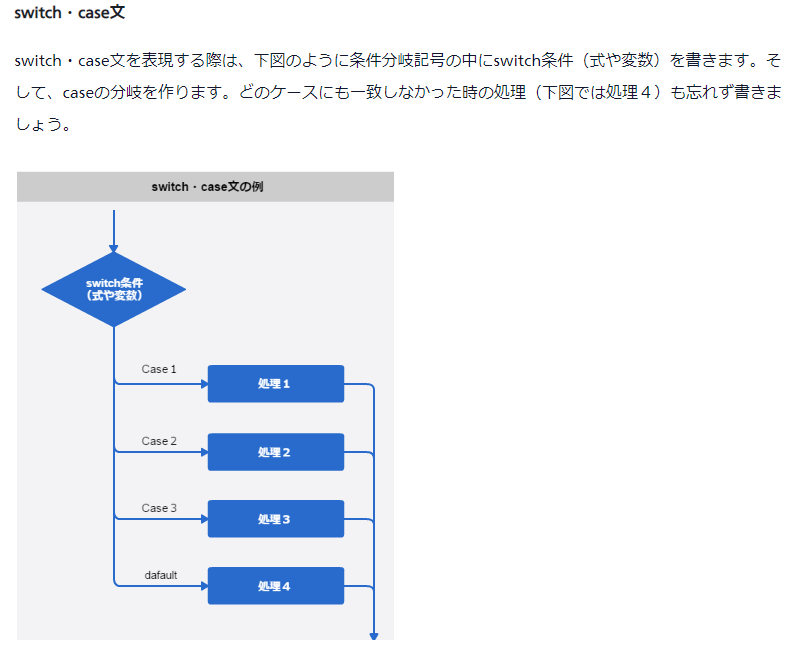
### よく使用される例

### 5.1. 順次構造

順次構造とは、処理する順番に記述されているプログラム構造のことです。  
単純に、上から下に処理が流れていくシンプルな構造をしています。  
前述したフローチャートの記号を並べて矢印でつなぐだけでＯＫです。



### 5.2. 分岐構造



ダイアグラム

自動的に生成された説明

ダイアグラム, じょうごグラフ

自動的に生成された説明

ダイアグラム が含まれている画像

自動的に生成された説明

ダイアグラム

中程度の精度で自動的に生成された説明

テーブル

自動的に生成された説明