

## 要件定義工程の進め方【全体概要】

要件定義工程の進め方は下記のとおり。

### 要件定義の流れ

1. 要件定義の計画作成
2. 業務要件の整理
  - 2-1. 現行業務の整理
  - 2-2. ヒアリング実施と課題分析
  - 2-3. 新しい業務の検討
  - 2-4. システム化範囲の検討
3. システム要件の整理
  - 3-1. プラットフォームの検討
  - 3-2. 機能要件と非機能要件の整理
4. まとめ／見積り
  - 4-1. 要件定義書の作成
  - 4-2. 基本設計以降の計画作成
  - 4-3. 見積りの算出

ざっくり説明すると、

1. どう進めるのか計画を立てて
2. どんな業務にしたいのか整理し、
3. どんなシステムが必要なのか整理し、
4. 要件定義書にまとめ、見積りをする

という流れだ。

## 1. 要件定義の計画作成

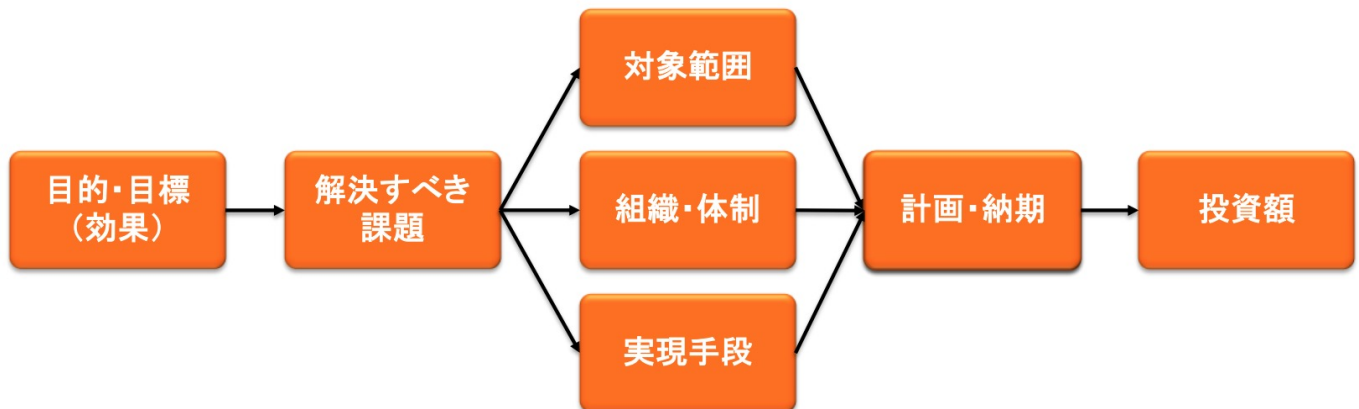
### 成果物

- ・ 要件定義の計画書

要件定義をはじめるにあたって、システム化の方針を明確にし要件定義のスケジュールや体制図を作成する。

と書いたものの、実際の現場では要件定義工程に入る前に計画書を作成している。でないと、顧客から作業費用がもらえないからだ。

また、要件定義をはじめた直後は5W2Hが曖昧なことが多いため、要件定義のなかで明確にしていこう。



Who	誰が？
What	何を？
When	いつ？
Where	どこで？
Why	なぜ？
How	どうやって？
How Much	いくらで？

<b>Who</b>	<b>組織・体制</b>
<b>What</b>	<b>解決すべき課題</b>
<b>When</b>	<b>計画・納期</b>
<b>Where</b>	<b>対象範囲</b>
<b>Why</b>	<b>目的・目標（効果）</b>
<b>How</b>	<b>実現手段</b>
<b>How Much</b>	<b>投資額</b>

出典：[IPA『高品質のための超上流工程における企業の課題・取組み事例集』](#),p.11

## 2.業務要件の整理

若手エンジニアのよくある失敗例として、「どんなシステム機能が欲しいですか？」というように、業務要件ではなく、システム要件のヒアリングをしてしまうことがある。

いきなりシステム要件をヒアリングすることがなぜ危険かという点、最初は顧客側も現状分析が十分にできていないことが多く、このような状況でシステムの質問をしても、顧客側から話を引き出せなかったり、後々の工程で機能を追加されたりと、トラブルの生じるリスクが高くなってしまふからだ。

ゆえに、要件定義では「システム要件の整理」の前に「業務要件の整理」を行うのである。

業務要件の整理では下記の作業内容を行う。1つずつ説明していこう。

2-1.現行業務の整理

2-2.ヒアリング実施と課題分析

2-3.新しい業務の検討

2-4.システム化範囲の検討

## 2-1. 現行業務の整理

### 成果物

- ・ 現行業務フロー
- ・ 業務処理定義書

業務の「あるべき姿（To-be）」と「現状（As-is）」とのギャップが「業務課題」となる。

課題を抜けなく洗い出すためにも、現行業務（As-is）の整理は重要なのだ。

現行業務を整理する手法はいろいろあるが、「業務フロー」を作成するのが一般的で、フロー上  
に書ききれない詳細な内容は、「業務処理定義書」に記載する。

プロジェクトの特性に応じて、「データフロー」や「ユースケース」を用いて整理することもある。

また、現在は、部分的にシステム化をしている場合がほとんどなので、現行システムの調査も行  
い、業務フローにシステム機能を追記しておこう。

もし、現行システムの資料が無い場合は、要件の抜け漏れを防ぐためにも、下記のような資料を  
整理をしておいた方がいいだろう。

### 現行システムの資料

- ・ システム機能階層図
- ・ 画面一覧
- ・ 画面遷移図
- ・ 帳票一覧
- ・ ER図
- ・ エンティティ一覧
- ・ エンティティ定義書
- ・ 外部システム関連図
- ・ 外部インターフェース一覧
- ・ 外部インターフェース定義書

## 2-2.ヒアリング・問題分析・解決策検討

### 成果物

- ・ ヒアリング結果一覧
- ・ 問題/影響/原因/対策一覧

業務部門にヒアリングをしながら課題を分析し、対策を検討するタスク。

対策には、システム化による対策だけでなく、業務プロセスだったり体制の変更なども含める。

それでは下記3つの観点で説明していこう

- 1) ヒアリングの実施
- 2) 問題の分析
- 3) 解決策の検討

### 1) ヒアリングの実施

今回のプロジェクトのテーマ（つまり、事業課題）に沿った業務上の問題について、業務部門にヒアリングを行う。

ヒアリングの方法にもいろいろあるが、例えば、日立製作所では、RAカードと呼ばれるアンケートカードを用いて、「問題」「影響」「原因」「願望」という4項目に分けて記入してもらうそう。

問題：現在の業務で起こっている問題  
影響：問題の結果として発生している好ましくない状況  
原因：問題を引き起こしている最も重要な原因  
願望：問題の解決策や改善案

ヒアリングした内容は、一覧表にまとめて後々確認できるようにしておこう。

### 2) 問題の分析

ヒアリングの結果をもとに、問題・影響・原因を構造的に整理する。

整理するときは、「問題」から整理していく良いだろう。

### 参考：問題分析

影響：問題に対して、「だから何？」

（問題）出荷伝票の書き間違いがある。

↓ だから何？

（影響）誤出荷に繋がっている。

原因：問題に対して、「なぜ？」

（問題）出荷伝票の書き間違いがある。

↓ なぜ？

（影響）出荷依頼をメールや電話、FAXなどで受けて、伝票を書き起こしているため。

影響に対しては「だから何？」を繰り返して、「売上／コスト／顧客満足度／コンプライアンス」といった経営レベルの観点につなげよう。

原因に対しても「なぜ？」を繰り返して、「業務プロセス／システム／制度・ルール／組織・担当者／設備・機器」といった観点到し込んでおくと対策が検討しやすい。

また、それぞれの問題に対して優先順位を仮決めしておく、解決策の検討対象を絞ったりシステム化の対象範囲を決めたりする際に役立つだろう。

## 3) 解決策の検討

掘り下げた原因に対しての対策を検討する。

前述した5つの観点を考えると対策が出やすい。（システム化だけが対策ではない）

- ・業務プロセス
- ・システム
- ・制度・ルール
- ・組織・担当者
- ・設備・機器

ここでの“システム”に当たる部分がシステム要件へとつながっていく。

## 2-3.新しい業務の検討

### 成果物

- ・新業務フロー

現行業務の資料に検討した解決策を反映させる。

現行のデータフローやユースケースを整理していた場合は、改善後の資料も整理しておこう。

## 2-4.システム化範囲の検討

### 成果物

- ・システム化の対象範囲

新しい業務の姿が描けた後はシステム要件の整理へと入っていくが、その前に今回のプロジェクトでのシステム化範囲を検討する。

要件定義開始時の範囲から外れていなければ問題ないが、問題分析・解決策の検討を行った結果、システム化範囲が変わってきている可能性があるからだ。

具体的な作業としては、要件定義開始時の5W2HのWhere（対象範囲）やHow（実現手段）をチェックする。

当初と比べてシステム化範囲が拡大している場合、範囲を絞ったり、プロジェクト予算を増やしたりする必要があるだろう。（要件定義のスケジュールが延長となることもある）

## 3.システム要件の整理

業務要件の整理が終わったら、システム要件の整理を行う。

システム化の範囲やプラットフォームに変更がないかを再検討し、システムの機能要件と非機能要件を具体的にしていくという作業だ。

では、下記2つの作業内容を説明していこう。

### 3-1.プラットフォームの検討

### 3-2.機能要件と非機能要件の整理

## 3-1.プラットフォームの検討

### 成果物

- ・ハードウェア構成
- ・ソフトウェア構成
- ・ネットワーク構成

プラットフォームとは、システム機能を提供するためのネットワーク、ハードウェア、OS、ミドルウェア（DBなど）のこと。

前作業「システム化範囲の検討」の結果を受けて必要なプラットフォームを整理していく。プラットフォームの変更は、プロジェクトに多大な影響を及ぼすためしっかりと検討しておきたい。

## 3-2.機能要件と非機能要件の整理

新しい業務を実現するうえでのシステム要件を整理する。

システム要件は「機能要件」と「非機能要件」に分類できる。

機能要件 : 業務を代行、または支援する機能

非機能要件 : 性能・信頼性・操作性など、機能以外の要件

### 機能要件の成果物

- ・システム機能要件
- システム機能階層図



- ・画面要件
  - 画面一覧
  - 画面遷移図
  - 画面レイアウト
- ・帳票要件
  - 帳票一覧
  - 帳票レイアウト
- ・データ要件
  - ER図
  - エンティティ一覧
  - エンティティ定義書
- ・外部インターフェース要件
  - 外部システム関連図
  - 外部インターフェース一覧
  - 外部インターフェース定義書

### 非機能要件の成果物

- ・ユーザビリティ及びアクセシビリティ要件
- ・システム方式要件
- ・規模要件
- ・性能要件
- ・信頼性要件
- ・拡張性要件
- ・上位互換性要件
- ・継続性要件
- ・情報セキュリティ要件
- ・情報システム稼働環境要件
- ・テスト要件
- ・移行要件
- ・引継ぎ要件
- ・教育要件
- ・運用要件
- ・保守要件

機能要件は、画面や帳票といった入出力機能から整理していくと良いだろう。

非機能要件は、IPAの『[非機能要求グレード](#)』を参考にすると整理をしやすい。

機能要件と非機能要件は下記の記事でも説明しているので、参考にしてほしい。



### 機能要件・非機能要件の書き方【サンプル有り】

2018年8月17日

## 4.要件定義のまとめ／見積り

要件定義のまとめとして、下記の作業を行う。

- 4-1.要件定義書の作成
- 4-2.基本設計以降の計画作成
- 4-3.見積りの算出

### 4-1.要件定義書のまとめ

#### 成果物

- ・ 要件定義書

これまで作成してきた資料をまとめる作業だ。

[>> 「要件定義書の書き方」を見る](#)

### 4-2.基本設計以降の計画作成

#### 成果物

- ・ WBS

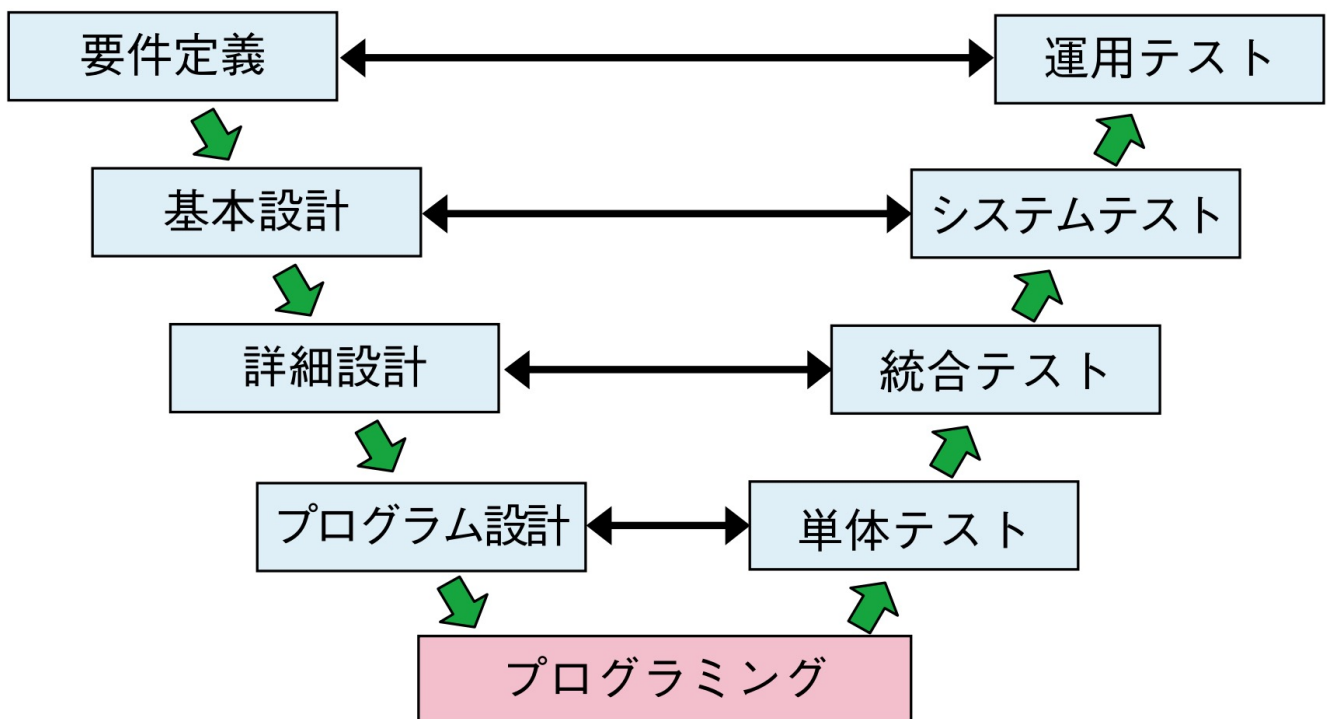
要件定義書の作成後は、基本設計以降の作業を洗い出し、スケジュールを立てよう。

ここで作成した資料は、見積りや、後続工程でのスケジュール・コスト管理にも使用する。

### 基本設計以降の流れ

- ・ 基本設計（外部設計）
- ・ 詳細設計（内部設計）
- ・ プログラム設計
- ・ プログラミング
- ・ 単体テスト
- ・ 統合テスト（結合テスト）
- ・ 総合テスト
- ・ 運用テスト

### システム開発のV字モデル



出典 : [IPA『ソフトウェアテスト見積りガイドブック』](#), p.8

## 4-3.見積りの算出

### 成果物

- ・ 見積書

要件定義書が完成した後は、システム開発の見積りを算出し予算をオーバーしていないかをチェックする。

参考までに、下記に代表的な見積り手法を3つ紹介しておこう。

- 1) 類推見積り
- 2) パラメトリック見積り
- 3) ボトムアップ見積り

## 1) 類推見積り

過去の類似プロジェクトから、見積りを類推する方法。

予算獲得時など、上流工程の初期見積りで使われることが多いが、精度が粗いため、要件定義以降ではほとんど使われない。

## 2) パラメトリック見積り

機能数や画面数などに係数をかけて見積りを算出する。

係数の設定次第で見積り精度が変わってくるが、利用実績の多い企業では係数の信頼度も高く、ユーザー企業の理解も得やすいため、最も利用される手法だ。

## 3) ボトムアップ見積り

プロジェクトの作業を洗い出し、工数を算出する方法。

作業ベースで見積るので最も精度が高く、プロジェクト開始後のスケジュール・コスト管理にも流用できる。

[IPA『ソフトウェア開発見積りガイドブック』](#), pp.48-59 を参考に記載

## さいごに

以上、要件定義の進め方を説明してきた。

ここに紹介してきた要件定義の進め方は、情報処理推進機構（IPA）や日経BPの記事をベースにしているため、やや”お堅い”内容になっている。

私の経験上、自社システム開発ではここまでしっかりとした要件定義をしていない。だが、顧客のシステム開発を請け負う場合には、作業ボリュームに大きい小さいはあるものの要件定義が行われることは認識しておいた方がいいだろう。

この記事が参考になれば幸いだ。

要件定義工程の関連記事はこちら。



### 要件定義工程の進め方

2021年2月7日



### 要件定義の目的は「プロジェクト5W2Hを明確にして失敗を減らすこと」

2021年2月6日



### 要件定義における成果物一覧と書き方（要件定義書サンプルあり）

2021年2月5日



### 機能要件・非機能要件の書き方【サンプル有り】

2018年8月17日

## 参考文献

[IPA『共通フレーム2013』](#)

[IPA『経営者が参画する要求品質の確保』](#)

[IPA『ユーザのための要件定義ガイド』](#)

[IPA『非機能要求グレード』](#)

[IPA『ソフトウェア開発見積りガイドブック』](#)

[日経BP『要求定義の方法論を知る』](#)

[日経BP『手戻りなしの要件定義「業務の分析・設計」』](#)

[経済産業省『情報システム・モデル取引・契約書（受託開発（一部企画を含む）、保守運用）＜第一版＞』](#)

システム開発ノウハウ

ツイート

はてブ

feedly

シェア

LINE

Pocket

コメントを残す

メールアドレスが公開されることはありません。

コメント

