

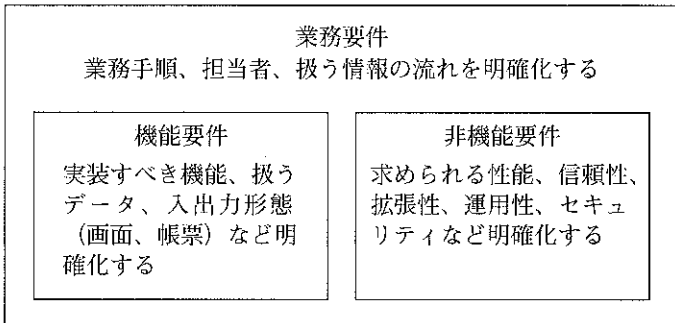
## 第2章 要件定義・設計

### 2.1 要件定義

要件定義では、現状の業務を分析して、どのように新しい業務のあり方をつくるべきかを明らかにしなければなりません。最初に業務の視点で要件をまとめます。

「誰の依頼でどのような業務を行い、誰に結果を返すのか」

この流れを要求事項としてまとめたものを業務要件と呼びます。業務要件の中からシステム化の範囲を決め、システムに要求される機能要件と非機能要件を洗い出します。



#### 2.1.1 業務要件

業務要件をまとめるためには、業務内容や業務フローを明確化しなければなりません。経営戦略、システム戦略、利用者のニーズなどを考慮し、新しい業務のあり方や運用をまとめた上で、業務上実現すべき条件や環境を定義します。後工程での齟齬を発生させないためにも、関係者間のヒアリングを十分行い、関係者間での共通認識が持てる資料作りが重要となります。業務要件は、システム化を前提にまとめる必要はありません。

#### 2.1.2 機能要件

業務要件の中からシステム化する要件を抽出し、システムが持つべき機能を明示したものが機能要件となります。求める機能を定義する上で、実装すべき機能、扱うデータ、画面・帳票などの入出力形態を明らかにします。機能要件では、必要とされる機能を文書化して定義するにとどめ、具体的な設計・実装方針は基本設計以降のフェーズに委ねます。

## 第2章 要件定義・設計

### 2.1.3 非機能要件

非機能要件とは、機能要件には含まれないシステムに求められる要求全般を表し、主にシステムの強度や品質に関わる要求のことを指します。定量的に評価できる要求項目を列挙する工夫が求められます。

特性	特性の説明	具体例
可用性	システムが継続して稼働できるか	システムは冗長化されているか サービス切替時間は規定時間未満か
性能/拡張性	どれだけ快適に使えるか。利用者が増えても大丈夫か	メモリの利用率は規定値未満か ユーザ数の上限は規定値以上か
運用/保守性	アフターサービスはきっちりとされているか	バックアップ作業は自動化されているか 保守作業が規定数以上自動化されているか
移行性	引越しや、乗り換えは簡単に出来るのか	移行期間が規定値未満であるか 移行対象のデータ量が規定値未満であるか
セキュリティ	ウィルス対策など、セキュリティ対策がされているか	認証機能が実装されているか 不正行為を検知するログが出力されるか
システム環境/ エコロジー	そのモノを置く設置環境は適切か。また環境保護に役立っているか	災害対策がされた場所に設置されているか システムが使用する製品が製品安全規格を取得しているか

参考資料 独立行政法人情報処理推進機構 システム基盤の非機能要求に関する項目一覧

## 要件定義のチェックポイント

- 作業開始前に計画が承認されているか
- 要求の実現性は検討されているか
- 要件定義に十分な期間が割り当てられているか
- スケジュールの妥当性が確認されているか
- システムの対象範囲が明確化されているか
- 要求発生元が追跡できる仕組みが考慮されているか
- 機能要件が定義した理由と共に明確化されているか
- 非機能要件が定量的に評価できる要求となっているか
- 要件定義書をレビューしているか

## レビューのチェックポイント

- 事前に作成者が誤字脱字を直しているか
- レビュー対象物を事前にレビュアーに配布しているか
- レビュアーは事前に問題点や質問事項を整理しているか
- レビューの時間は長くても2時間程度としているか
- レビュー参加者は多くても6、7名程度に絞られているか
- レビューの指摘内容を記録に残しているか

## 代表的なレビューの種類

インスペクション	レビュー進行のモデレータ（推進者、調整者）が、参加者を招集して会議形式で実行する問題点が発見された場合、解決策を検討し処置の是正作業を指示する
ウォークスルー	仲間内で自主的に会議召集し、問題点をチェックする。原則管理者は出席しない問題点の発見を主目的とし、解決策までは検討しない

参考文献 IEEE 1028 Software Reviews and Audits

## 2.2 基本設計

要件定義書を元に、求められるシステムの機能／仕様／構成を体系的に整理し定義します。システム間の連携や構成を明確にし、機能一覧、画面や帳票など入出力に関する項目、生成されるデータの概要など、システムの基礎的な仕様を明らかにする工程となります。基本設計書の取りまとめには、顧客企業や外部システムの開発チームとの仕様調整を初めとする下記課題に対して、方針を決定しなければなりません。

- ・全体最適の視点による設計
- ・あいまいな要件定義の明確化
- ・高品質を確保する実装方針の決定
- ・オープンシステムの最適な組み合わせ設計
- ・制約条件の克服
- ・他の連携システムとの仕様調整
- ・非機能要件の考慮
- ・明確な文書の作成（オフショア先への意識）

これらの課題に取り組みながら、システムを実現するための方式設計を行い、機能設計、その他設計を行います。

### 基本設計におけるタスク

基本設計	方式設計	プラットフォーム設計
		アプリケーション・アーキテクチャ設計
		開発標準やテスト方式の決定
	機能設計	ビジネスロジック設計
		データベース設計
		画面・帳票設計
		バッチ設計
	その他の設計	性能／信頼性設計
		セキュリティ設計
		移行設計
		運用設計

## 基本設計のチェックポイント

- 要件定義書は、顧客の責任者が承認しているか
- 運用および保守の基本方針を定めて設計しているか
- 顧客の利便性を考慮して入出力帳票および画面等を設計しているか
- データベースは業務の内容およびシステム特性に応じて設計しているか
- データの保全性を確保しているか
- ネットワークは業務の内容に応じて設計しているか
- 情報システムの性能は、要件定義を満たしているか
- システム構成はピーク時を想定して設計しているか
- 他の情報システムとの整合性を考慮して設計しているか
- 運用に必要な性能管理等の技術的な実現方法等を設計しているか
- 情報システムの障害対策を講じているか
- 不正防止、機密保護等の機能を設計しているか
- テスト計画は、目的、範囲、方法、スケジュール等を明確にしているか
- 情報システムの利用にかかわる教育の方針、スケジュール等を明確にしているか
- 基本設計書をレビューしているか

## 2.3 詳細設計

基本設計の成果物を元に、求められる機能の動作仕様をプログラムでどのように実現するのかを設計します。

初めに機能を実現するための要素を大まかに切り出し、各要素間での役割を定め、各要素間の入出力データを洗い出します。その後、各要素を構成する機能を分解し、個々の機能が単純で簡素になるまで局所化を進め、各機能の入出力インタフェースの設計を行います。

特に出力設計では、正常時の出力だけでなく異常が発生した際の処理についても明確に決定する必要があります。異常処理は「業務的な異常」と「システム的な異常」があり、双方の異常を検出した段階でどのように振舞うかを決定しなければなりません。

処理種別	例	処理の例
正常処理	—	業務処理を記述する
業務的異常処理	データ異常	ログを出力する 利用者に再入力を求める
システムの異常処理	メモリ不足 ディスク不足	ログを出力する

## 詳細設計のチェックポイント

- 基本設計書は、顧客および開発の責任者が承認しているか
- 基本設計書に基づいて、詳細設計をしているか
- テスト要求事項を定義し、文書化しているか
- 詳細設計書およびテスト要求事項をレビューしているか
- 詳細設計時に発見した基本設計書の矛盾は、基本設計の再検討を行って解決しているか