

「CSAJ/JCSSA 情報システムの信頼性向上のための取引慣行・契約に関する検討委員会」

～情報システムの取引慣行・契約に関する実施ガイド～

<企画・開発に関するガイドライン>

社団法人コンピュータソフトウェア協会（CSAJ）  
社団法人日本コンピュータシステム販売店協会（JCSSA）

## 目次

1.	<b>総論</b>	1
1.1.	経緯	1
1.2.	目的	1
1.3.	ガイドの全体像	1
1.4.	用語	4
2.	<b>パッケージソフトウェアベース開発</b>	5
2.1.	概要	5
2.2.	主要プロセス	7
2.2.1.	業務要件定義プロセス	8
2.2.2.	開発プロセス	13
2.2.3.	移行・運用準備プロセス	17
2.3.	関係者の役割分担	19
2.4.	ベンダの説明事項と手順	21
2.4.1.	業務要件定義プロセス	21
2.4.2.	開発プロセス	28
2.4.3.	移行・運用準備プロセス	32
3.	<b>サービス調達 (SaaS、ASP)</b>	34
3.1.	概要	34
3.2.	主要プロセス	35
3.2.1.	企画プロセス	36
3.2.2.	要件定義、開発 (システム要件定義) プロセス	36
3.2.3.	開発 (設計・製作・テスト)、移行・運用準備プロセス	37
3.3.	SaaS での課題と必要な対応	37
3.4.	関係者の役割分担	39
3.5.	ベンダの説明事項と手順	39
4.	<b>アジャイル開発・プロトタイピング</b>	41
4.1.	概要	41
4.2.	主要プロセス	42
4.3.	企画・開発時の要点	42
4.4.	関係者の役割分担	43
4.5.	ベンダの説明責任事項	44
4.6.	留意事項	44
5.	<b>繰り返し型開発</b>	46
5.1.	概要	46
5.2.	主要プロセス	46
5.3.	企画・開発で起こるトラブル	46
5.4.	関係者の役割分担	47
5.5.	ベンダの説明責任事項	47
6.	<b>付録</b>	48
6.1.	付録1 パッケージソフトウェア開発でのトラブル例と必要な対応	48
6.2.	付録2 フェーズチェックリスト	62

6.3. 付録3 アジャイル開発適用のチェックリスト .....67

以下のチェックリストについては、本書の「添付資料1 (1)報告書本編」に添付していますのでそちらを参考にして下さい

- ・提案依頼書(RFP)のチェックリスト
- ・業務システム仕様書の記述レベル
- ・ユーザ IT 成熟度チェックリスト
- ・パッケージ選定のためのチェックリスト
- ・SaaS 選定のためのチェックリスト
- ・検収事前チェックリスト

# 1. 総論

## 1.1. 経緯

2007年4月13日に経済産業省から「情報システムの信頼性向上のための取引慣行・契約に関する研究会」最終報告書「情報システム・モデル取引・契約書」が発表された。その中で、パッケージを中心としたシステム導入の場合、反復繰り返し型の開発の場合、中小企業ユーザにおける活用の場合等、モデル取引・契約書で十分カバーできていない論点について、業界団体を中心としてさらに議論が深めることが必要であると指摘されている。特に今後ますます広がると考えられる中堅、中小企業でのパッケージ活用、SaaS、ASP 活用などに対する検討が重要と考え、CSAJ/JCSSA では、2006年に実施してきた共同作業部会を発展させ、情報システム・モデル取引・契約書に関する検討委員会で検討を進めることとなった。

## 1.2. 目的

「情報システムの信頼性向上のための取引慣行・契約に関する研究会」最終報告書を踏まえ、パッケージソフトウェア、SaaS、ASP を利用し情報システムの構築、サービスの利用を行う中小・中堅企業と、ソフトメーカー、システムインテグレータ、外部専門家との間の取引慣行・契約に関し課題を明確化し、パッケージソフトウェア、SaaS、ASP の企画、設計、構築、運用、保守にかかわる関係者（ユーザ、システムインテグレータ、メーカー、外部専門家等）の権利と義務、責任の明確化及び信頼性向上に資するガイドライン、モデル契約の策定、政策、普及策の提言を行うこととした。

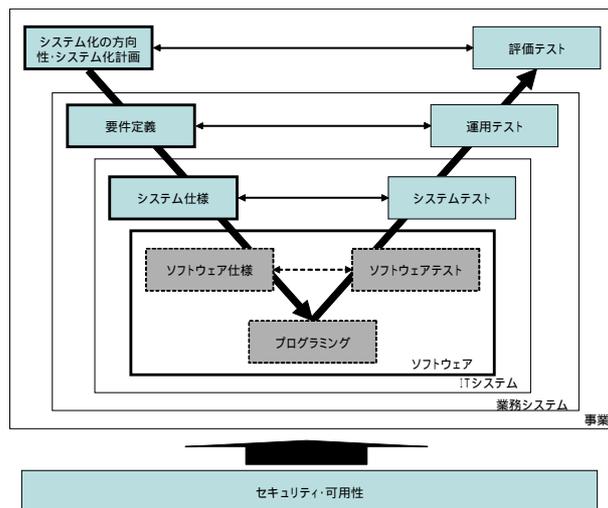
## 1.3. ガイドの全体像

### 1) 位置づけ

本ガイドは、平成 19 年 4 月に経済産業省が発表した「情報システム・モデル取引・契約書」を参考にしながら整備しているが、特に、パッケージソフトウェアや SaaS、ASP のシステムライフサイクルの中での企画、設計・開発の工程を対象としている。右図の V 字開発モデルの全体を対象としている。

保守、運用及びセキュリティ・可用性については、それぞれの分冊を参照されたい。

パッケージソフトウェアや SaaS、ASP ではないプログラミング主体の開発を行う場合には、本ガイドを見ながら不足している部分は、経済産業省「情報システム・モデル取引・契約書」を参照していただきたい。



## 2) 対象とする内容

本ガイドは、パッケージソフトウェアを主体としたプロセスを対象としている。このプロセスは共通フレーム 2007 に準拠する。特に以下の3種類の契約を想定している。

企画支援業務、業務要件定義等のパッケージソフトウェア選定及び要件定義支援契約（以下、「業務要件定義支援契約」と記載）

システム要件定義、カスタマイズ、アドオン、テスト等のソフトウェア設計・制作業務（以下、「システム要件定義支援契約」と記載）

システムの開発・実装を行う構築業務契約

上記の一貫契約

特に、ベンダの取引慣行とユーザの予算に起因する齟齬などからトラブルが多数発生している、提案書、見積書等での確認事項などを重点に以下の項目を検討する。

仕様、性能、信頼性に関する説明事項、手順

保証内容、保証期間（プロダクトライフサイクル）、トラブル発生時の対応窓口、保証対象とならない場合の説明事項、手順

最終的に双方が確認する重要事項説明書等の合意プロセスの組み込み

ここで、パッケージソフトウェアを主体といっても様々な導入形態があることから、以下のパターンを検討対象とする。

単体パッケージソフトウェアによる開発

✓ 概要

システム導入を1つのパッケージソフトウェア製品により導入する形態

✓ 例

ERPによる全社システム  
人事システムのための人事パッケージソフトウェア

パッケージソフトウェア・インテグレーション

✓ 概要

複数の機能別パッケージソフトウェアを組み合わせる総合的な機能の実現を図るパッケージソフトウェア・インテグレーション

✓ 例

販売管理パッケージソフトウェアと会計パッケージソフトウェアの連携

SaaS、ASP等のサービス利用

✓ 概要

システムを自社で持たず、インターネット経由でサービス提供を受ける。

✓ 例

会計ネットワークサービス  
CRM ネットワークサービス

また、ユーザが利用するアプリケーションにはパッケージソフトウェアを使わないが、システムの中核機能としてパッケージソフトウェアを使うミドルウェア・インフラパッケージソフトウェアを使用したシステムは、プログラミングをベースとしたシステムと同列の扱いとし、今回対象外とする。

### 3) 前提条件

本ガイドで記述する内容は、以下の条件を前提としている。

#### パッケージソフトウェアの範囲

対象とするソフトウェアは、一般化された業務フロー、処理、機能を持ち、主にパラメータの設定などで適用できる所謂「プロダクト・パッケージソフトウェア」とする。中核の機能だけ提供し、業務はカスタマイズでプログラムを組み構築するようなソフトウェアは対象としない。また、特定企業で作成したテンプレートしか持たず、汎用的な業務モデルになっていない製品も対象としない。

#### 契約当事者

専門知識を有しないユーザ

専門知識を有する IT ベンダ、システムインテグレータ、外部専門家

例 委託者(ユーザ): 民間中小・中堅企業、市町村地方自治体 等

受託者(ベンダ): IT ベンダ、システムインテグレータ 等

外部専門家 : コンサルタント 等

#### 開発モデル

経済産業省「情報システム・モデル取引・契約書」パッケージソフトウェア選定モデルに準じる。また、パッケージソフトウェアのカスタマイズ等はウォーターフォール型、アジャイル型、反復繰り返し型を想定する

#### 対象システム

企業基幹系、情報系システム

#### プロセス

共通フレーム 2007 による標準化されたシステム企画・開発・運用・保守プロセスによる

#### マルチベンダ契約

工程分割発注により、工程毎にベンダを変えることを可能とする。

## 留意事項

ユーザの IT リテラシーが高いとは限らないため、ユーザ側契約担当者が十分に契約内容を理解できない場合も想定される。契約交渉・締結時には、契約当事者間の情報の質と量、理解に差があることを理解し十分な配慮を行うことが必要である。特に、契約合意に至る交渉プロセスでは、お互いの理解内容を確認しながら進めるなど配慮を行う必要がある。

### 1.4. 用語

主要な用語は以下に記述する。下記以外の用語は、共通フレーム 2007 の用語解説を参照する。

- ・ モディファイ

パッケージソフトウェアのソースコードの変更を伴うカスタマイズ。

- ・ アドオン

パッケージソフトウェアのソースコードの変更を伴わず、API (Application Program Interface)、外部 I/F、ファイル交換等を利用した外部プログラム。単独で動作するものと、パッケージソフトウェア本体とともに動作する場合がある。

- ・ RFI

Request For Information、情報提供依頼書。

- ・ RFP

Request for Proposal、「提案依頼書」、「提案要望書」、「見積依頼書」などと言う。情報システム調達の際に、ベンダに詳細なシステム提案を行うよう要求すること、またはその調達要件をまとめた仕様書等をいう。

- ・ SaaS

Software as a service、サースもしくはサーズと読む。ユーザが必要とする機能だけを選択し利用可能にしたソフトウェアの形態。もしくは、それを実現するための提供形態・方法 (デリバリモデル) を指す場合もある。

- ・ ASP

Application Service Provider。アプリケーションソフトをインターネットを介してユーザに提供するサービス形態。もしくはサービス提供事業者を指す場合もある。

## 2. パッケージソフトウェアベース開発

### 2.1. 概要

#### 単体アプリケーションによる開発

必要とする情報システムを 1 つのパッケージソフトウェアにより構築する方式である。全社パッケージソフトウェアを ERP で導入する場合や、人事システムを人事機能専門の総合パッケージソフトウェアで開発する場合がこの方式である。

##### 1) メリット

1 つのパッケージソフトウェアをベースにシステム開発をするため、システム開発が容易である。また、パッケージソフトウェアを商品とする時点で、ベンダ内でシステム全体での検証が終わっているため信頼性が高い。

##### 2) デメリット

パッケージソフトウェアは、機能を汎用化するとともに経済性などで取捨選択するので、機能が不十分なことがある。また、パッケージソフトウェア全体のフレームは既に確立されているため、追加機能の付加などのモディファイ（改造）を行うときに制約がある場合がある。

#### パッケージソフトウェア・インテグレーション

必要とする情報システムを複数のパッケージソフトウェアを組み合わせて構築する方式である。販売や会計など業務分野ごとに最適なパッケージソフトウェアを選定し、一体として運用できるようにする場合がこの方式である。

##### 1) メリット

業務ごとに自社に合った最適のパッケージソフトウェアを選定するので、業務の専門性を高めたり、高度なサービスの実現が可能になる。また、機能に過不足のあるパッケージソフトウェアを無理して使ったりすることがなく、必要な機能を組み合わせられることから、システムを合理的に構成することができる。

##### 2) デメリット

パッケージソフトウェア間のデータの受け渡しなどに問題が生じることがある。また、パッケージソフトウェアごとに操作方法が違うなど、ユーザ・インタフェースが揃わない場合も多い。

#### ミドルウェア・インフラパッケージソフトウェアの使用（本ガイド対象外）

システムを構成する要素として、パッケージソフトウェアを導入することがある。ユーザの目には触れにくいだが、重要な、データ管理、出力などの機能を担うことも多い。

##### 1) メリット

専門性の高い部位、機能に、安定した高度なサービスを求めることができる。

## 2) デメリット

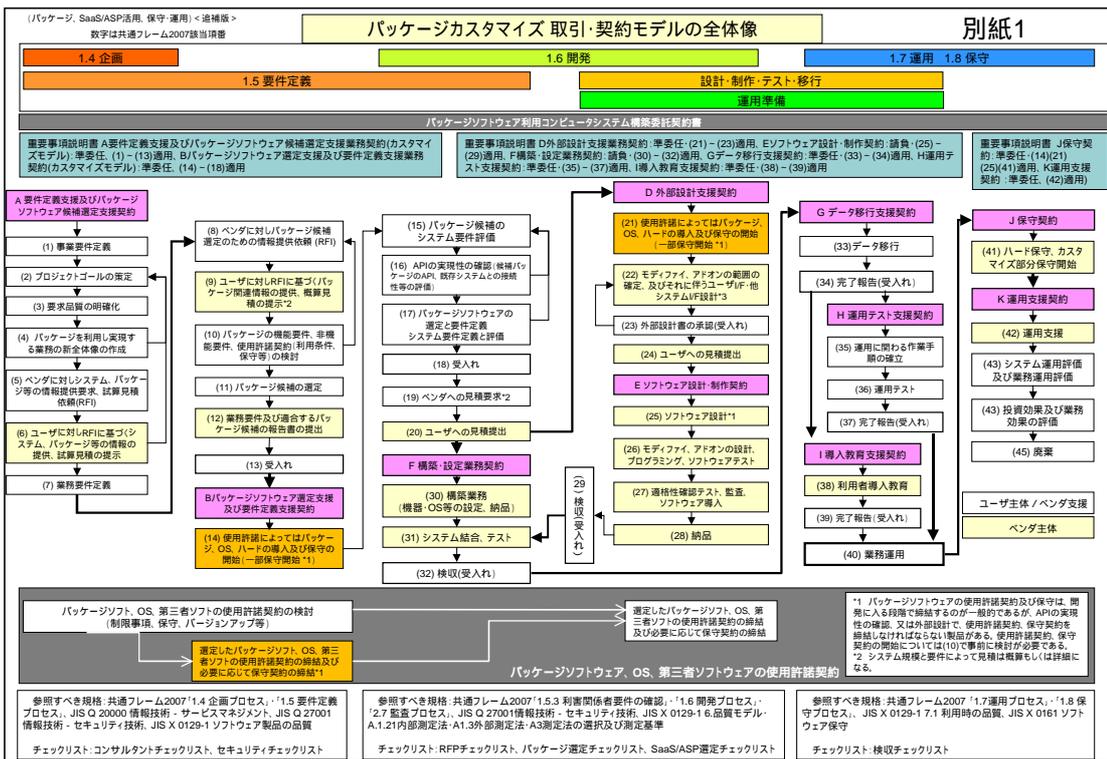
システム内に組み込まれているため、障害の解析などが難しい。

## 2.2. 主要プロセス

プロジェクトのゴールにむけて企画・開発を行っていく。「業務要件定義」、「開発（システム要件定義）」においては、準委任契約によりコンサルティング契約を外部と結び、その作業を委任することも多い。また、「開発（設計・制作・構築）」では、請負契約が一般的で、テストや教育を準委任契約で契約する場合が多い。契約は、中小規模の契約では一括して行われる場合も多いが、リスクを軽減する意味から、工程ごとに契約を行う多段階契約を行うことを検討する必要がある。

この業務要件定義からシステム要件定義までの過程において、パッケージソフトウェアの持つ設計思想や機能を理解することになるが、その途中で当初考えていたゴール（実現目標）が変更になることもある。また、この過程において、ゴール達成のための必要な機能をパッケージソフトウェアで実現できるのかどうかの検証を行うフィット&ギャップ分析が重要である。

情報システムの信頼性向上のための取引慣行・契約に関する研究会報告書で、パッケージソフトウェア活用の場合のモデル契約プロセスが記述されているが、本ガイドでは共通フレーム 2007 との整合を取りながら、以下のように再整理を行っている。



また本モデル契約では、契約書を補完する意味で、重要事項説明書を活用することとしている。建設業界の重要事項説明書と同様に、契約書本体ではないが、この説明をベンダがユーザに行い、設計・開発の基本となる重要事項に関して役割と責任を明確にするとともにユーザの理解を求めることとする。重要事項説明書の説明においては、ユーザの理解に合わせて丁寧に説明を行う必要がある。

## 2.2.1. 業務要件定義プロセス

### A. システム化の方向性（事業要件定義、プロジェクトゴールの策定、要求品質の明確化）

このプロセスで、業務改革の責任者やシステム責任者は、システム導入を通じて実現したいことを明確にする。

情報システムの信頼性向上のための取引慣行・契約に関する研究会報告書では、以下のように記述している。

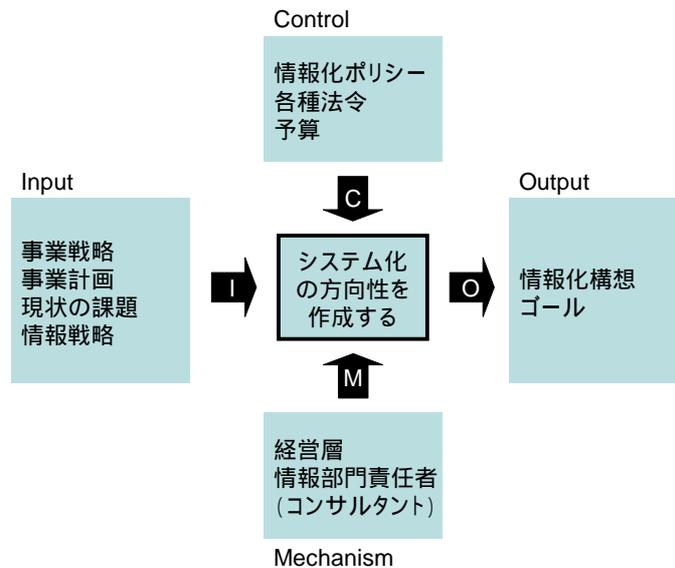
「システム化の方向性」、業務部門が、システム開発の前に、経営層が定めた経営方針、全体最適化計画に従い、事業上の目的、システム化の対象業務、システム化のニーズと課題、予算、事業環境を分析し、利害関係者からの要請やその数や役割に応じた規模などに配慮しつつ、システム化するビジネスモデルにつき十分な検討を繰り返し、取締役会等経営層による承認を受けてシステム化の方向性を決定するフェーズである。

ユーザは、ベンダに対し、このフェーズの成果を生かして RFP を発し、ベンダ等から「仮試算見積レベル」の見積提案を受け、これに検討評価を加えて、システム化計画に移行する。

ユーザは、必要に応じて、ベンダ、IT技術者、ITコーディネータ、システム監査人、情報セキュリティ監査人、公認会計士、弁護士等の専門家との間で支援業務を内容とする準委任契約としてのコンサルティング契約を締結し、これらの作業のための支援を受ける。

（共通フレームのアクティビティ）概ね「システム化構想の立案」に相当する。

業務上のゴールの策定においては、在庫の削減など、構想開始時に考えた視点はもちろんのこと、コストの削減、業務の高度化などの様々な視点から検討をしていく必要がある。パッケージソフトウェアを用いた開発を行う場合には、パッケージソフトウェアの持つ知見やアイデアを調査し、その内容を参照しながら現状の業務体系を見直す。ウォータフォールモデルのようにゴールを達成するための自社専用のシステムを一から作るのではないので、選定したパッケージソフトウェアの思想や機能によってゴールが変わることがあるが、仮のプロジェクトゴール（システム導入の成果）を検討、策定するフェーズである。



日本情報システム・ユーザー協会「ビジネスシステム定義研究 2004」で作成した業務システム仕様書の記述レベルのレベル1が求められる。

仕様の責任と記述項目	レベル1 ビジネス機能提示	レベル2 ビジネスプロセス提示	レベル3 業務フロー提示	レベル4 業務処理提示	レベル5 業務処理/データ項目提示
A ビジネス機能関連図			IS部門で企業/事業全体機能定義	IS部門で企業/事業全体機能定義	IS部門で企業/事業全体機能定義
B ビジネス連携図			業務と対外系/他部門間との連携	業務と対外系/他部門間との連携	業務と対外系/他部門間との連携
C ビジネスルール定義書			企業/業務上の戦略ルール	企業/業務上の戦略ルール	企業/業務上の戦略ルール
D システム化目標定義書	業務システム化の目標設定	業務システム化の目標設定	業務システム化の目標設定	業務システムのIT効果定義	業務システムのIT効果定義
1 ビジネス機能構成表	ビジネス機能の大分類定義	ビジネス機能の中小分類定義	ビジネス機能の細分類定義	ビジネス機能の細分類定義	ビジネス機能の細分類定義
2 ビジネスプロセス関連図		ビジネスプロセス間の関連定義	ビジネスプロセス間の関連定義	ビジネスプロセス間の関連定義	ビジネスプロセス間の関連定義
3 業務流れ図			業務処理フロー指示(含む例外処理)	業務処理フロー指示(含む例外処理)	業務処理フロー指示(含む例外処理)
4 業務機能関連図				DFD方式での上位DFDとして作成	DFD方式での上位DFDとして作成
5 業務ルール定義書				業務処理上の社内ルールを定義	業務処理上の社内ルールを定義
6 業務処理手順書				個別の業務処理手順を定義	個別の業務処理手順を定義
7 画面/帳票一覧			基本的に必要な画面/帳票一覧	基本的に必要な画面/帳票一覧	基本的に必要な画面/帳票一覧
8 画面/帳票レイアウト			画面/帳票レイアウトを定義	画面/帳票レイアウトを定義	画面/帳票レイアウトを定義
9 データ項目定義書					データ項目の属性を定義
10 運用・操作要件所			業務システムの運用・操作の条件設定	業務システムの運用・操作の条件設定	業務システムの運用・操作の条件設定

B. システム化計画（パッケージソフトウェアを利用し実現する業務の新全体像（BPR）の作成、ベンダに対し情報提供要求、概算見積要求（RFI）、ユーザに対しRFIに基づく情報の提供、ガイドライン適用状況の説明、概算見積の提示）

このプロセスで、ユーザ担当者は、インターネット、書籍、雑誌などを通じて導入できる製品やサービスの情報を収集（カタログ収集等）するとともに、ゴールを達成するための基本方針を検討し、概要の計画を作成する。

情報システムの信頼性向上のための取引慣行・契約に関する研究会報告書では、以下のように記述している。

「システム化計画」は、業務部門が、システム化の方向性を具体化するために、開発体制、予算、スケジュール、システム化する事業上の要求（例えば、システム化すべき新規事業、社外連携、組織改編、部門間業務分掌変更、法令・契約等のコンプライアンス要件、セキュリティ、個人情報保護、環境など）や対象業務上の要求（例えば、業務内容、業務形態、業務品質、性能目標、運用、移行要件、法令・契約等のコンプライアンス要件、セキュリティ、個人情報保護、事業継続性、環境など）を考慮して、業務範囲や業務分掌、関係者の教育及び訓練計画を定めたシステム化計画書を作成し、ステークホルダの合意を得てから経営層の方針稟議を求め、経営層による承認を受けて、業務部門及び情報システム部門における要件定義に進むフェーズである。

ユーザは、部門間の検討の後、システム化計画を作成して、レビューを繰り返して検討を加え、ベンダに対して RFP を発し、ベンダ等から「試算見積レベル」の見積提案を受ける。これを検討評価して要件定義フェーズに移行する。

ユーザは、システム化の方向性決定フェーズ同様、必要に応じて、ベンダ、IT 技術者、IT コーディネータ、システム監査人、情報セキュリティ監査人、公認会計士、弁護士等の専門家との間で支援業務を内容とする準委任契約としてのコンサルティング契約を締結し、作業のための支援を受ける。

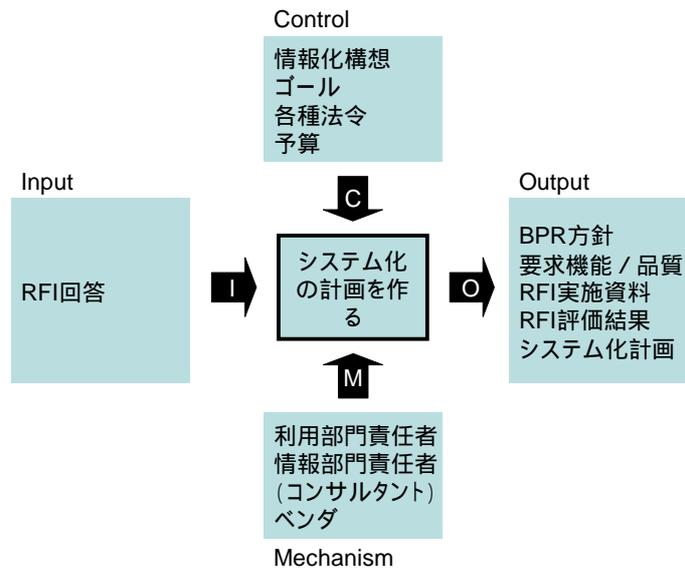
（共通フレームのアクティビティ）概ね「システム化計画の立案」に相当する。

ブラッシュアップした詳細なプロジェクトゴールや要求機能、BPR 方針を作成した後に、RFI によりベンダからの情報提供を受ける。それを踏まえ、システムの全体方向性、BPR が検討されるフェーズである。

パッケージソフトウェアは業務の知見を集積しソフトウェアとして提供するものであるため、自社の既存の業務にとらわれることなく、必要に応じてゴールの変更や業務方針の変更を図っていく必要がある。

この工程で作成した成果物にシステムのスケジュールや体制など、システムの具体的な内容を付加して、システム計画書として整理することが望まれる。この概要は、重要事項説明書の前半部分を記載し始めることにより、整理をすることができる。

ここで、(1)～(6)におけるプロセスは、仮説設定、検証を実施していくが、それぞれが独立したプロセスではなく、同時並行的、一体的に推移する場合もあり、大幅な手戻りや見直しが許容される。この時点から外部専門家（コンサルタント等）やベンダの参画を得る場合もある。



### C. 業務要件定義プロセス

このプロセスで、ユーザ担当者は、業務に必要な機能を明確にするとともに、現状の業務量や業務の流れを示して、ベンダに対して提案を依頼する。

情報システムの信頼性向上のための取引慣行・契約に関する研究会報告書では、業務要件定義とシステム要件定義をあわせて、以下のように記述している。

「要件定義」は、事業要件を反映したシステム化計画を受けて、業務部門が、業務上の要求を業務要件に、情報システム部門が、システムに実装すべきシステムの機能要件・非機能要件を定義し、経営層による実行稟議、承認を受けるフェーズである。

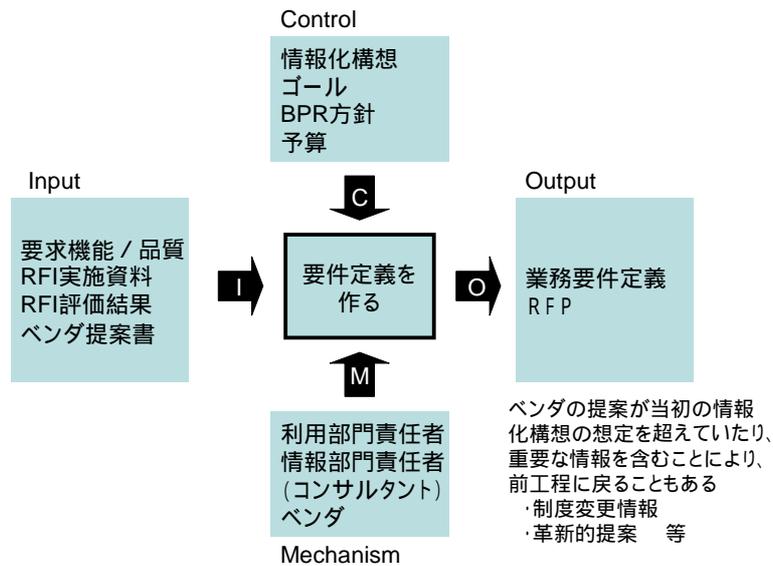
ここでは、ユーザは、業務要件及びシステム要件を検討して、要件定義書にまとめ、ベンダに RFP を発して「概算見積レベル」の見積提案を受け、これに点検評価を加えて、システム設計に移行する。

そのために、ユーザは必要に応じて、ベンダ、IT技術者、ITコーディネータ、システム監査人、情報セキュリティ監査人、公認会計士、弁護士等の専門家との間で、準委任契約を内容とする要件定義支援契約、仕様書作成支援契約を締結し、作業のための支援を受ける。

このフェーズの内容の妥当性の検証が、「運用テストフェーズ」である。

(共通フレームのアクティビティ) 共通フレーム 2007 で新設される「利害関係者要件の定義」、「利害関係者要件の確認」に相当する。

プロジェクトゴールや要求品質、BPR 方針、業務要求などとシステム化計画をもとに、それを実現するための要件仕様書を含んだ RFP を作成する。前工程でベンダから集めたパッケージソフトウェア情報などを参考にしながら、今回整備するシステムの業務の要件や使用許諾条件等を整備していく。詳細な検討を行うために再度ベンダに問い合わせることもあり、その場合には、再び RFI を実施することとなる。やはり、この工程は専門性も高いため外部専門家（コンサルタント等）やベンダの参画を得る場合も多い。



業務要件定義で作成される要件は、日本情報システム・ユーザー協会「ビジネスシステム定義研究 2004」で作成した業務システム仕様書の記述レベルのレベル2が最低限求められる。本来は業務フローを含むレベル3が必要であるが、それができない場合には次工程以降のフィット&ギャッププロセスと行き来することによって、後から再定義することとなる。

仕様の責任と記述項目	レベル1 ビジネス機能提示	レベル2 ビジネスプロセス提示	レベル3 業務フロー提示	レベル4 業務処理提示	レベル5 業務処理/データ項目提示
A	ビジネス機能関連図		IS部門で企業/事業全体機能定義	IS部門で企業/事業全体機能定義	IS部門で企業/事業全体機能定義
B	ビジネス連携図		業務と対外系/他部門間との連携	業務と対外系/他部門間との連携	業務と対外系/他部門間との連携
C	ビジネスルール定義書		企業/業務上の戦略ルール	企業/業務上の戦略ルール	企業/業務上の戦略ルール
D	システム化目標設定書	業務システム化の目標設定	業務システム化の目標設定	業務システムのIT効果定義	業務システムのIT効果定義
1	ビジネス機能構成表	ビジネス機能の中小分類定義	ビジネス機能の細分類定義	ビジネス機能の細分類定義	ビジネス機能の細分類定義
2	ビジネスプロセス関連図	ビジネスプロセス間の関連定義	ビジネスプロセス間の関連定義	ビジネスプロセス間の関連定義	ビジネスプロセス間の関連定義
3	業務流れ図		業務処理フロー指示(含む例外処理)	業務処理フロー指示(含む例外処理)	業務処理フロー指示(含む例外処理)
4	業務機能関連図			DFD方式での上位DFDとして作成	DFD方式での上位DFDとして作成
5	業務ルール定義書			業務処理上の社内ルールを定義	業務処理上の社内ルールを定義
6	業務処理手順書			個別の業務処理手順を定義	個別の業務処理手順を定義
7	画面/帳票一覧		基本的に必要な画面/帳票一覧	基本的に必要な画面/帳票一覧	基本的に必要な画面/帳票一覧
8	画面/帳票レイアウト		画面/帳票レイアウトを定義	画面/帳票レイアウトを定義	画面/帳票レイアウトを定義
9	データ項目定義書				データ項目の属性を定義
10	運用・操作要件所		業務システムの運用・操作の条件設定	業務システムの運用・操作の条件設定	業務システムの運用・操作の条件設定

## 2.2.2. 開発プロセス

開発は、要件定義に従って設計を行い、実際にシステムを構築していくフェーズである。

### A. システム要件定義

このプロセスで、ユーザ担当者は、ベンダからの提案を比較検討し、パッケージソフトウェアを選定するとともに、選定したベンダとともに、画面イメージなどシステムを構築するために必要な詳細機能を明文化していき、構築するシステムを詳細レベルまで定義する。

情報システムの信頼性向上のための取引慣行・契約に関する研究会報告書では、「要件定義」と「開発」に含まれる。「要件定義」は前項で引用しているのでここでは引用を省略する。「開発」の中では、以下のように記述されている。

企画・要件定義段階を経て、ベンダが、ユーザとの間でソフトウェア開発契約に基づいて情報システム開発を展開する段階である。システム設計（システム外部設計）、システム方式設計（システム内部設計）、ソフトウェア設計、プログラミング、ソフトウェアテスト、システム結合、システムテスト、導入・受入支援を経てシステム開発は終了する。

ウォーターフォール型の開発モデルでは、要件定義、外部設計、内部設計、プログラミング等の各工程を明確に区切り、順次実行される。

他方、反復型では、開発対象を小さな機能単位に分割、機能ごとに各フェーズを繰り返し適用して開発される。プロトタイプモデルでは、工程を区分せずに、ユーザの要望から試作品を作成・提示・評価していくことで開発される。パッケージソフトウェア導入の場合は、要件定義はフィット&ギャップ評価、システム設計・プログラミングはカスタマイズ（いわゆるアドオンも含む。）に置き換えられる。

しかしながら、どのような開発モデルにおいても、要件定義がそのシステムの挙動を定めること、要件定義にはユーザの参画が不可欠であることに変わりはない。

パッケージソフトウェアを使った開発を行う場合の特徴として、業務要件とパッケージソフトウェアが提供可能な機能の充足度の確認を行うフィット&ギャップ評価を行うことが重要である。

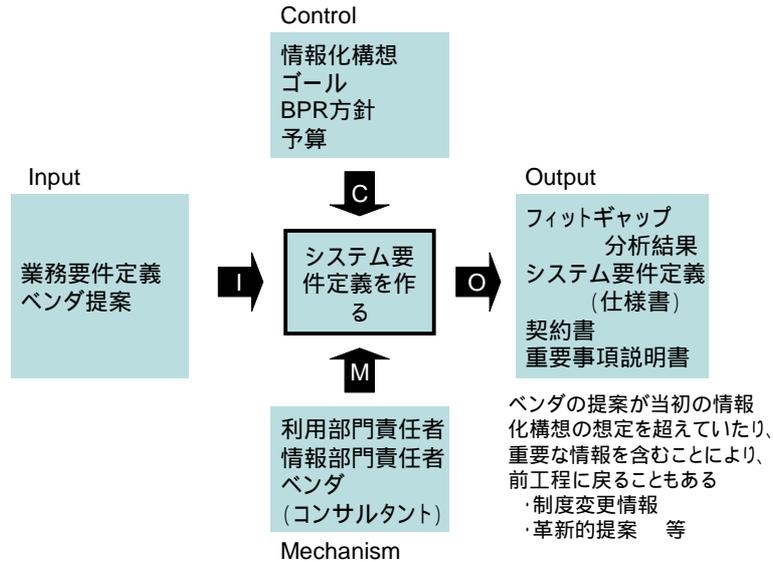
ベンダからの提案を基にしたフィット&ギャップ評価では、パッケージソフトウェアの適合性、カスタマイズ（モディファイ）やアドオンの必要性、外部インタフェース、既存システムからの移行、要員教育、将来にわたる仕様の拡張性などとともに、運用・保守体制、償却期間におけるトータルコストと得られる効果を、パッケージソフトウェアごとに検証する。

フィット&ギャップ評価の中で、今回の開発の基本事項に関する矛盾や新たな提案が発見される場合がある。その場合には、フィット&ギャップ評価に基づき、プロジェクトゴールや BPR の見直しなど、上流工程にさかのぼった変更が許容される。

フィット&ギャップ評価を得て、パッケージソフトウェアを選定した後に、必要なアドオン、カスタマイズ（モディファイ）、システム構成等を整理し要件定義として取りまとめる。機能の詳細レベルで再度フィット&ギャップ評価を行うこ

ともある。実機での検証などが必要な場合には、この段階でパッケージソフトウェアやOSの導入及び保守を開始する。

ここで、パッケージソフトウェアにない機能をカスタマイズ（モディファイ）で追加することもできるが、パッケージソフトウェアは一定の開発思想と経済合理性をもって設計されているため、ユーザ仕様に適合しない部分を無理にモディファイすることで、一部性能の大幅な低下や、将来の拡張に制限を招く場合がある。モディファイを行う場合には、モディファイによる制限事項やライフサイクルに対する影響に関して留意する必要がある。



パッケージソフトウェアの適合性フィットギャップ評価では、日本情報システム・ユーザー協会「ビジネスシステム定義研究 2004」で作成した業務システム仕様書の記述レベルのレベル3が求められる。業務フローレベルでの評価が求められる。

仕様の責任と記述項目	レベル1 ビジネス機能提示	レベル2 ビジネスプロセス提示	レベル3 業務フロー提示	レベル4 業務処理提示	レベル5 業務処理/データ項目提示
A	ビジネス機能関連図		IS部門で企業/事業全体機能定義	IS部門で企業/事業全体機能定義	IS部門で企業/事業全体機能定義
B	ビジネス連携図		業務と対外系/他部門間との連携	業務と対外系/他部門間との連携	業務と対外系/他部門間との連携
C	ビジネスルール定義書		企業/業務上の戦略ルール	企業/業務上の戦略ルール	企業/業務上の戦略ルール
D	システム化目標定義書	業務システム化の目標設定	業務システム化の目標設定	業務システムのIT効果定義	業務システムのIT効果定義
1	ビジネス機能構成表	ビジネス機能の中小分類定義	ビジネス機能の細分類定義	ビジネス機能の細分類定義	ビジネス機能の細分類定義
2	ビジネスプロセス関連図	ビジネスプロセス間の関連定義	ビジネスプロセス間の関連定義	ビジネスプロセス間の関連定義	ビジネスプロセス間の関連定義
3	業務流れ図		業務処理フロー指示(含む例外処理)	業務処理フロー指示(含む例外処理)	業務処理フロー指示(含む例外処理)
4	業務機能関連図			DFD方式での上位DFDとして作成	DFD方式での上位DFDとして作成
5	業務ルール定義書			業務処理上の社内ルールを定義	業務処理上の社内ルールを定義
6	業務処理手順書			個別の業務処理手順を定義	個別の業務処理手順を定義
7	画面/帳票一覧		基本的に必要な画面/帳票一覧	基本的に必要な画面/帳票一覧	基本的に必要な画面/帳票一覧
8	画面/帳票レイアウト		画面/帳票レイアウトを定義	画面/帳票レイアウトを定義	画面/帳票レイアウトを定義
9	データ項目定義書				データ項目の属性を定義
10	運用・操作要件所		業務システムの運用・操作の条件設定	業務システムの運用・操作の条件設定	業務システムの運用・操作の条件設定

システム要件定義では、日本情報システム・ユーザー協会「ビジネスシステム定義研究 2004」で作成した業務システム仕様書の記述レベルのレベル4が求められる。レベル5の内容は、次工程の設計以降で達成される。

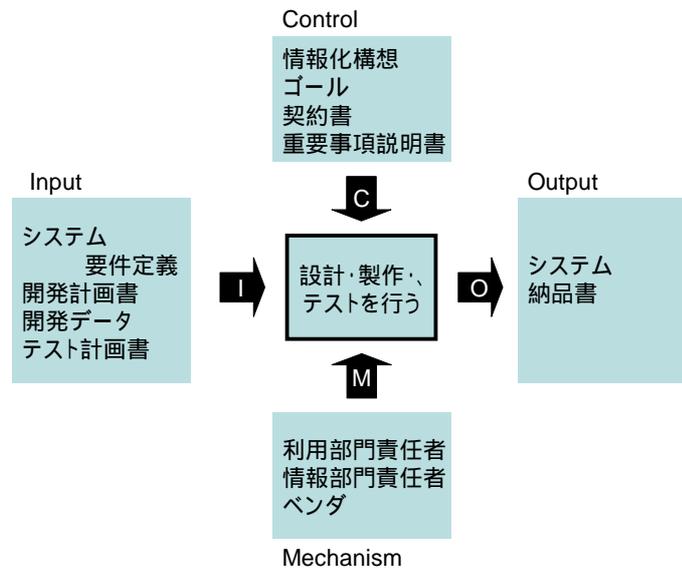
仕様の責任と記述項目	レベル1 ビジネス機能提示	レベル2 ビジネスプロセス提示	レベル3 業務フロー提示	レベル4 業務処理提示	レベル5 業務処理/データ項目提示
A	ビジネス機能関連図		IS部門で企業/事業全体機能定義	IS部門で企業/事業全体機能定義	IS部門で企業/事業全体機能定義
B	ビジネス連携図		業務と対外系/他部門間との連携	業務と対外系/他部門間との連携	業務と対外系/他部門間との連携
C	ビジネスルール定義書		企業/業務上の戦略ルール	企業/業務上の戦略ルール	企業/業務上の戦略ルール
D	システム化目標定義書	業務システム化の目標設定	業務システム化の目標設定	業務システム化のIT効果定義	業務システム化のIT効果定義
1	ビジネス機能構成表	ビジネス機能の細分類定義	ビジネス機能の細分類定義	ビジネス機能の細分類定義	ビジネス機能の細分類定義
2	ビジネスプロセス関連図	ビジネスプロセス間の関連定義	ビジネスプロセス間の関連定義	ビジネスプロセス間の関連定義	ビジネスプロセス間の関連定義
3	業務流れ図		業務処理フロー指示(含む例外処理)	業務処理フロー指示(含む例外処理)	業務処理フロー指示(含む例外処理)
4	業務機能関連図			DFD方式での上位DFDとして作成	DFD方式での上位DFDとして作成
5	業務ルール定義書			業務処理上の社内ルールを定義	業務処理上の社内ルールを定義
6	業務処理手順書			個別の業務処理手順を定義	個別の業務処理手順を定義
7	画面/帳票一覧		基本的に必要な画面/帳票一覧	基本的に必要な画面/帳票一覧	基本的に必要な画面/帳票一覧
8	画面/帳票レイアウト		画面/帳票レイアウトを定義	画面/帳票レイアウトを定義	画面/帳票レイアウトを定義
9	データ項目定義書				データ項目の属性を定義
10	運用・操作要件所		業務システムの運用・操作の条件設定	業務システムの運用・操作の条件設定	業務システムの運用・操作の条件設定

## B. 設計・製作・テスト

このプロセスで、ユーザ担当者は、開発の進捗についてベンダから報告を受けるとともに、実装機能の問い合わせなど、必要に応じてシステム実装に関する調整を実施する。

情報システムの信頼性向上のための取引慣行・契約に関する研究会報告書では、「開発」に含まれる。前項で引用しているのでここでは引用を省略する。

開発プロセスは、パッケージソフトウェアベンダによるもの、パッケージソフトウェアそのものの開発者ではないが導入のみ行うシステムインテグレータによるもの、それぞれの再委託によるものと多岐に渡ってさまざまなケースが想定される。守秘義務と品質保証、受入テスト、検収に至る一連のフェーズについて、役割の明確化と十分な事前合意が必要である。



### 2.2.3. 移行・運用準備プロセス

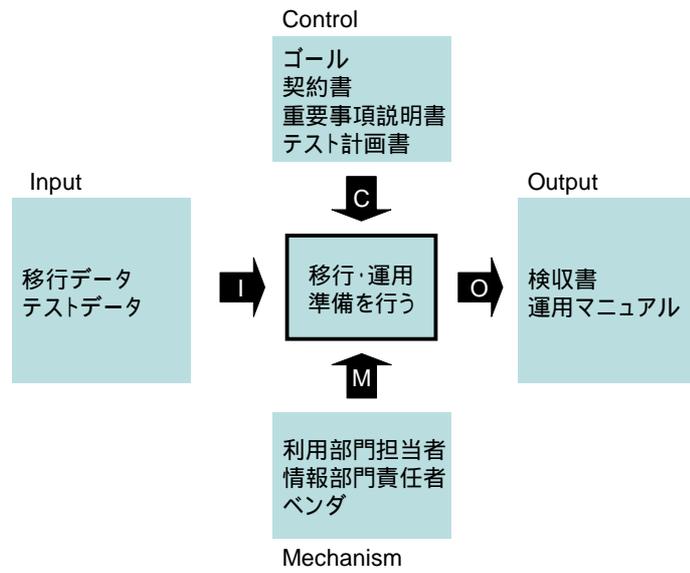
このプロセスで、ユーザ担当者は完成したシステムを実業務として実施してみ  
て、納品物の検収を行うとともに、利用部門の習熟を図る。

情報システムの信頼性向上のための取引慣行・契約に関する研究会報告書では、「開発」の中で「導入・受入支援」、「保守運用」の中で運用テストのみが以下のように記述されている。

「導入・受入支援」は、疑似環境又は実環境にソフトウェアを導入し、ユーザのソフトウェア受け入れレビュー及びテストを支援するフェーズである。  
 （共通フレームのアクティビティ）概ね「ソフトウェア導入」「ソフトウェア受け入れ支援」に相当する。

「運用テスト」は、疑似運用環境等での運用テストの実施と実運用環境に移行を実施するフェーズである。  
 （共通フレームのアクティビティ）概ね「運用テスト」「業務及びシステムの移行」に相当する。

パッケージソフトウェアの移行・受入準備では、パッケージソフトウェアの持つデータ移行機能の確認を行う必要がある。また、利用者の教育には、ベンダが開催するパッケージソフトウェア操作に関する基礎的な研修が用意されている場合も多い。研修開催スケジュールの確認やモディファイした部分の研修などを調整する必要がある。



### 2.3. 関係者の役割分担

システム開発ではユーザの利用部門の責任者が責任を負うが、情報システム部門は技術面から支援をする必要がある。また、トラブルの多発する RFP や要件の確定においては、現場とベンダが協力して行い、情報システム部門は調整を行う必要がある。

項目	ユーザ (現場)	ユーザ (システム)	ベンダ	備考
プロジェクトゴールの策定			-	
要求品質の明確化				
BPR 方針の策定				
システム化の方向性				
RFI				
ベンダ情報評価				
システム化計画				
RFP				
RFP の内容確認		調整役		
ベンダ提案受付				
フィット&ギャップ分析			支援	ベンダは必要な情報提供を行う
パッケージソフトウェア選定				
要件定義				
要件の詳細項目確認		調整役		
モディファイ・アドオンの設計、開発				

外部専門家が作業に参加するときには、ユーザのシステム部門と同じ立場になる。

ユーザ企業にシステム部門がない場合

ユーザ部門の現場が責任を持つが、実質的には外部専門家が作業を行い、ユーザ部門は確認のみ行う場合が多い。

項目	ユーザ (現場)	専門家	ベンダ	備考
プロジェクトゴールの策定	(確認)	(実質)	-	
要求品質の明確化	(確認)	(実質)		
BPR 方針の策定	(確認)	(実質)		
システム化の方向性	(確認)	(実質)		
RFI	(確認)	(実質)		
ベンダ情報評価	(確認)	(実質)		
システム化計画	(確認)	(実質)		
RFP	(確認)	(実質)		
RFP の内容確認	(確認)	(実質)		
ベンダ提案受付	(確認)	(実質)		
フィット&ギャップ分析	(確認)	(実質)	支援	ベンダは必要な情報提供を行う
パッケージソフトウェア選定	(確認)	(実質)		
要件定義	(確認)	(実質)		
要件の詳細項目確認	(確認)	(実質)		
モディファイ・アドオンの設計、開発	(確認)	(実質)		

## 2.4. ベンダの説明事項と手順

### 2.4.1. 業務要件定義プロセス

#### a. システム化の方向性

このフェーズで契約する場合の成果物は、情報化構想やシステム化の方向性に関する報告である。報告には、今後システム化を進めていく上での方針や要求すべき機能や品質の概要が記述される。

外部支援を受けたときに起こる契約上のトラブルは以下の事項である。

- 1) システム方針が経営者の満足する内容になっていない。

この工程におけるベンダの説明事項と手順、確認ポイントを以下に整理する

#### 説明事項

- ・システム方針（情報化構想）

システム全体の基本方針と将来構想を記述したもの。

- ・ゴール

業務やシステムを通じて実現したいゴールを記述したもの。

保証内容、保証期間（プロダクツライフサイクル）、トラブル発生時の対応窓口、保証対象とならない場合に関する説明事項

ここで決める内容はあくまでも概要であり、保証の対象外とする

#### 成果物イメージ等

##### 専門家選定時の説明事項、手順

この工程から外部専門家を活用する場合には、付録のチェックリストによる確認を実施し、当該外部専門家が実施プロジェクトに適しているか評価する必要がある。

##### 時の説明事項、手順

提案時には、ユーザの要望をヒアリングしたベンダが、ユーザの行いたいことを提案書として整理し、書面により確認を行う。

確認項目は以下の内容である。付録の外部専門家のチェックリストで評価する。

- ・背景
- ・目的
- ・実施したいこと
- ・目標指標
- ・実施手順

- ・ 成果物イメージ
- ・ 予算とその範囲
- ・ 体制と役割分担

また、契約時までには経営層の確認を受ける。

## 2) 程終了時の説明事項、手順

報告書（説明資料のみの場合もある）を基に、提案で記述された内容について説明していく。

途中の合意を持って、提案内容を変えた場合には、その変更内容も含めて確認を行う。

チェックリスト（フェーズ0）による確認

上記を確認後、経営層の承認を受けることで終了する。

## 3) 最終的に双方が確認するための合意プロセス

契約形態に依存するが、基本的には、業務部門が作成したものを情報システム部門がレビューし、最終的に、経営層が承認を与えることが必要である。

担当者レベルで内容確認

利用部門責任者による内容確認

合意文書で承認

## 運用・保守との連携に関する留意事項

このフェーズで運用・保守の詳細までは記述しないが、運用・保守が問題なく行える体制について検討をしていく必要がある。

## セキュリティ・可用性に関する留意事項

最低限維持すべきセキュリティ・可用性の要件を明確化する。「停止時間は8時間以内のこと」などのように具体的に記述する。詳細レベルまでは記述する必要はない。

別冊付録のセキュリティチェックリストで必要なセキュリティレベルの確認を行う。

## その他留意事項

なし

## B. システム化計画

このフェーズで契約する場合の成果物は、システム化計画書である。システム計画書には、システム化方式、機能の概要が記述されるとともに、具体的な実現方法、スケジュールなどが記述される。

外部支援を受けたときに起こる契約上のトラブルは以下の事項である。

- 1) システム計画書に具体性が無く、システム設計に詳細化していくことができない。
- 2) システム計画書が、技術や期間などの面から実現不可能であり、システム設計に進めない

この工程におけるベンダの説明事項と手順、確認ポイントを以下に整理する

### 説明事項

- ・ BPR 方針

業務改革を行う基本的な視点や方針を記述したもの。

- ・ 要求機能・品質

このプロジェクトで実現すべき機能の定義や要求品質を記述したもの。

- ・ システム計画書

システムの具体的な計画が記述されたもの。

保証内容、保証期間（プロダクツライフサイクル）、トラブル発生時の対応窓口、保証対象とならない場合に関する説明事項

ここで決める内容はあくまでも概要であり、保証の対象外とする

### 成果物イメージ等

#### 部専門家選定時の説明事項、手順

この工程から外部専門家を活用する場合には、付録の外部専門家チェックリストによる確認を実施し、外部専門家はそのチェック項目の内容をユーザに説明しなければならない。その情報をもとにユーザは、実施プロジェクトに適しているか評価する必要がある。

#### 案時の説明事項、手順

提案時には、ユーザの要望をヒアリングしたベンダが、提案書として、ユーザの行いたいことを整理し、資料により確認を行う。

確認項目は以下の内容である。付録の外部専門家のチェックリストで評価する。

- ・ 目的
- ・ 目標指標
- ・ 実施手順

- ・成果物イメージ
- ・予算とその範囲
- ・体制と役割分担

また、契約時までには経営層の確認を受ける。

### 3) 工程終了時の説明事項、手順

報告書（説明資料のみの場合もある）を基に、提案で記述された内容について説明していく。

途中の合意を持って、提案内容を変えた場合には、その変更内容も含めて確認を行う。

主な説明項目は以下の通りである。

- ・システム計画により初期目的が達成する根拠
- ・システム計画が実施可能な根拠

チェックリスト（フェーズ1）で確認する。

確認後、経営層の承認を受けることで終了する。

### 4) 最終的に双方が確認するための合意プロセス

契約形態に依存するが、基本的には、外部ベンダもしくは業務部門が作成したものを情報システム部門がレビューし、最終的に、経営層が承認を与えることが必要である。

## 運用・保守との連携に関する留意事項

このフェーズで要求機能や品質において、運用時間や必要な運用体制など、必ず留意しなければならない運用・保守の要件を記述する。この段階では、実行方法の詳細は不要としても、大まかなレベルの指定、設定は必要である。要件として定義できる程度に、要求が明確になっていなければならない。ベンダも技術的に実現可能かどうかの判断が求められる場合がある。

## セキュリティ・可用性に関する留意事項

このフェーズで、セキュリティ・可用性の要件を記述する。扱う情報の重要度や24時間運用など、構想時点で留意しなければならない項目を必ず記入する。

## その他留意事項

なし。

## C. 要件定義プロセス

業務要件を整理したうえで RFP による提案依頼が実施され、ベンダの提案を受け付ける。このフェーズで契約する場合の成果物は、業務要件定義書、提案依頼書（RFP）である。この作業により、今回構築するシステムが実装すべき業務機能や開発方法などが明確に規定される。

外部支援を受けたときに起こる契約上のトラブルは以下の事項である。

- 1) 業務要件定義に具体性が無く、システム要件定義に詳細化していくことができない。
- 2) 業務要件定義が、制度、技術や期間などの面から実現不可能であり、システム要件定義に進めない
- 3) 実施した RFP の記述内容が不十分であり、後工程でトラブルが発生する

この工程におけるベンダの説明事項と手順、確認ポイントを以下に整理する

### 説明事項

#### ・業務要件定義書

業務（システムではない）の要求機能、品質などを規定するもの。

- ✓ 最低限確保されるべき応答時間や処理時間などの操作性に代表される品質要件は BPR に密接に関連し、技術要件にも大きな影響を及ぼすため、早期に優先度と重要度を明確にすることが望ましい。また、セキュリティ、既存システムからの移行、運用・要員教育、保守等において特段に配慮すべき項目もあわせて抽出しておくことが望ましい。特に、(7)業務要件定義は、現場で運用に携わるオペレータや、利害関係者とプロジェクトゴールの共有や、システム化計画の周知と同意を得ておくことが重要で、プロジェクトチームと BPR に直面する現場の調整に留意すべきである。

#### ・RFP

業務要件とともに各種提案条件を記述するもの。ベンダに提示を行う。

### 保証内容、保証期間（プロダクツライフサイクル）、トラブル発生時の対応窓口、保証対象とならない場合に関する説明事項

RFP の記述内容の不備という問題が後工程で見つかった場合には、当該作業を実施した外部専門家は善管注意義務に問われることもあることを説明していく必要がある。ユーザ事由により要件未確定であったものの問題を問うことはできないが、常識的に必要な検討が行われていないことによる問題は外部専門家が責任を負う必要がある。

要件未確定部分がある場合には、そのことを書面によって明記することが重要である。

### 成果物イメージ等

#### 1) 業務要件定義時の説明事項、手順

業務要件を後工程で変更すると、予算、スケジュール、品質などに大きな影響が生じることを説明する必要がある。業務要件定義書のテンプレートを基に、記述内容にぬげがないか、記述内容は十分か確認を行う必要がある。

#### 2) RFP 実施時の説明事項、手順

RFP の意味、責任範囲、RFP で記述される要件の詳細、RFP の配布対象について説明を行う。特に、契約時に正式仕様を決定するが、そのもとになるのが RFP であり、RFP がユーザの責任において作成される公式文書であることへの理解を得ることが重要である。

RFP の記載内容を、チェックリストを使いながら確認を行う。

#### 3) FP での第三者評価の実施

RFP によるトラブルを防ぐために RFP の確認（監査）サービスを使うことが考えられる。その場合には、RFP のチェックリストのチェック内容を再確認するとともに、ユーザ IT 成熟度チェックリストにより、ユーザの IT に関するレベルも測定し、ユーザにも正しい処置を求める必要がある。

#### 4) パッケージソフトウェア選定時の説明事項、手順

パッケージソフトウェア選定時の評価項目をユーザに説明する必要がある。また、選定の対象とするパッケージソフトウェアの選定理由を明確に説明する。

#### 5) 工程終了時の説明事項、手順

報告書を基に、提案で記述された内容について説明していく。

途中の合意を持って、契約開始時の提案内容を変えた場合には、その変更内容も含めて確認を行う。

主な説明項目は以下の通りである。

- ・業務要件定義書 RFP の記載内容
- ・パッケージソフトウェア選定と判断根拠
- ・システム計画により初期目的が達成する根拠
- ・システム計画が実施可能な根拠

チェックリスト（フェーズ 2）で確認する。

これらの確認を行った上で、重要事項説明書の作成を行い、確認後、経営層の承認を受けることで終了する。

#### 6) 最終的に双方が確認するための合意プロセス

契約形態に依存するが、基本的には、ベンダもしくは業務部門が作成したものを情報システム部門がレビューし、最終的に、経営層が承認を与えることが必要である。各種成果物と重要事項説明書による確認を行った後、業務完了確認書兼検収書により双方の最終確認を持って合意する。

### 運用・保守との連携に関する留意事項

このフェーズで業務要件定義書の非機能要件の定義において、運用時間や必要な運用体制など、運用・保守の要件を明確に記述する。

#### セキュリティ・可用性に関する留意事項

このフェーズで業務要件定義書において、セキュリティ・可用性の要件を明確に記述する。要件は見積もりに影響するので、二重化などのシステム構成に影響する内容、バックアップに対する対応やアクセスログなどの機能を付加するなど、一般的な機能よりも多くの機能を要する場合には必ず記入する。

#### その他留意事項

なし

## 2.4.2. 開発プロセス

### A. システム要件定義

このフェーズで契約する場合の成果物は、システム要件定義書、重要事項説明書である。この作業により、今回構築するシステムの実装すべき機能や実際に適用される開発方法などが明確に規定される。

この工程で外部支援を受けたときに起こる契約上のトラブルは以下の事項である。

- 1) システム要件定義に具体性が無く、システム設計に詳細化していくことができない。
- 2) システム要件定義が、技術や期間などの面から実現不可能であり、システム設計に進めない

この工程におけるベンダの説明事項と手順、確認ポイントを以下に整理する

#### 説明事項

##### ・フィット&ギャップ分析結果

ベンダからの提案内容の要求への整合状況、パッケージソフトウェア選定の考え方について説明を行うもの。

- ✓ パッケージソフトウェアに対するモディファイについては、そもそもパッケージソフトウェアが想定していない運用を求めることによって、パッケージソフトウェアの構造そのものの変更などがありえる。そのため、モディファイやアドオン機能の工数と実現性について詳しく評価を行うべきである。
- ✓ フィット&ギャップ評価においてベンダの参加を求める場合、当該作業の内容と責任の所在を決めることが重要である。また、複数ベンダからの提案書およびフィット&ギャップ評価を求める場合は、書式の統一、用語の定義等に配慮し、相互理解に十分な時間をかけることが信頼性確保につながることに留意すべきである。

##### ・システム要件定義書

システムの要求機能、品質などを規定するもの。

- ✓ パッケージソフトウェアの保守期間は、前提となる OS やハードウェアの世代交代、保守打ち切りに影響される場合がある。パッケージソフトウェアの保守期間、ハードウェアの保守期間とともに OS の動向、保守打ち切りの際の移行、費用についても事前に調査、想定することが望まれる。
- ✓ 最終仕様の決定においては、画面の遷移や帳票の形式、運用手順等を、現場オペレータの参画と承認を得るとともに、導入に備えた教育計画が必須である。導入後の手直しや変更を最低限に留めることは、信頼性向上とコストに重大な影響を及ぼすため、十分に留意す

る。

- ✓ パッケージソフトウェアに対するカスタマイズによって、基本機能に制限が加わるなどの他の機能に及ぼす影響と、非機能要件に及ぼす影響について確認し、要件定義とする。
- ✓ ハードウェア、ネットワークの高性能化、低価格化に伴い、データ量の増大とデータの分散が顕著である。信頼性要件、セキュリティ要件の観点から、要件定義の最終評価を行うとともに、これらが付随的要件でないことに留意し信頼性を確保されたい。
- ✓ 対象となるパッケージソフトウェアのプログラムは、(1)パッケージソフトウェアの基本機能部分、(2)画面、帳票などの何らかの設定を前提としている部分、(3)新たに開発されるアドオン部分に分類される。プログラム変更、改修が(1)に係る場合は、信頼性に多大な影響を及ぼすとともに、将来に渡る保守が得られない場合もある。パッケージソフトウェアベンダと開発ベンダ、ユーザとの十分な相互理解と承認を得た上で、プログラム変更、改修を実施されたい。

#### ・重要事項説明書

ベンダがユーザに説明し、同意を取るべき重要事項が記述された文書であり、契約書の付帯文書である。

#### 保証内容、保証期間（プロダクツライフサイクル）、トラブル発生時の対応窓口、保証対象とならない場合に関する説明事項

システム要件定義書の記述内容や記述レベルはトラブルの最大の原因となる。要件定義書に関する内容の保証は契約上難しいので、十分な確認を行うことが重要である

#### 成果物イメージ等

##### 1) パッケージソフトウェアのフィット&ギャップ分析実施時の外部専門家契約時の説明事項、手順

フィット&ギャップ分析を行うときに、製品ベンダにフィット&ギャップの契約を別途行うことがある。このときに確認のみおこなう、追加費用の算定までおこなうなど、作業範囲を明確に示す必要がある。

##### 2) 提案時の説明事項、手順

提案時には、提案書として、ユーザの行いたいことを整理し、資料により確認を行う。  
確認項目は以下の内容である。

- ・実施手順
- ・成果物イメージ
- ・予算とその範囲
- ・体制と役割分担

また、契約時までには経営層の確認を受ける。

### 3) システム要件定義書確定時の説明事項、手順

システム要件を説明する前提として、フィット&ギャップ分析の結果を解説し、今回のパッケージソフトウェアが選定された検討プロセスとその結果を明確に説明する。また、システム要件の中で、パッケージソフトウェアから提供される機能、設定が必要な機能、カスタマイズを行う機能を明確に説明する必要がある。

付録のパッケージソフトウェア選定チェックリストも活用し、パッケージソフトウェア全体の確認をすることが望まれる。パッケージソフトウェアの設定について合意した場合には、この時点で、設定等合意書の作成を行う。

### 4) 契約書確定時の説明事項、手順

契約金額、期間等の契約基本事項を説明するとともに、モデル契約書との差異を説明する。特に、瑕疵担保期間など利害関係として大きく取り上げられる案件についてはユーザがわかるまで十分な説明をする必要がある。さらに、付帯文書である重要事項説明書を使って、ユーザとベンダの責任境界を明確にするとともに、両者がサインすることにより、基本事項について合意をしなければならない。

### 5) 工程終了時の説明事項、手順

チェックリスト(フェーズ3)により確認を行う。  
システム要件定義書、重要事項説明書の説明をもって工程を終了する。

### 6) 最終的に双方が確認するための合意プロセス

契約形態に依存するが、基本的には、外部ベンダもしくは業務部門が作成したものを情報システム部門がレビューし、最終的に、経営層が承認を与えることが必要である。  
契約がこの時点で終了する場合には、重要事項説明書で確認を行い、業務完了確認書兼検収書を作成する。また設定の合意をした場合には、設定等合意書も合わせて確認を行う。

## 運用・保守との連携に関する留意事項

このフェーズで作るシステム要件定義書において、運用・保守のシステム的な要件を具体的に記述する。運用については、運用サポート機能など機能的な内容だけではなく、必要な運用体制などについても言及をする必要がある。

## セキュリティ・可用性に関する留意事項

このフェーズで作るシステム要件定義書において、セキュリティ・可用性の要件を記述する。実装すべき技術標準やリカバリー手順など運用に必要な各種条件まで言及を行う。別冊付録のセキュリティ詳細チェックリストを活用する。

## その他留意事項

パッケージソフトウェアベンダとシステムインテグレーションベンダが異な

る場合、システムインテグレーションベンダで解決できない不具合や仕様変更が想定される。ベンダ間の協力体制、パッケージソフトウェアベンダとユーザにおける保守契約も合わせて選定評価の基準とすることが望まれる。大規模なカスタマイズなどの場合、進捗の報告について、時期、内容、終了の判断基準、進展の条件などの合意を事前に行うことも必要である。

## B. 設計・製作・テスト

このフェーズで契約する場合の成果物は、システム本体及び設計書、運用マニュアルなど付帯ドキュメントである。

この工程で外部支援を受けたときに起こる契約上のトラブルは以下の事項である。

- 1) システム要件定義に記述されていない追加要件などの扱い
- 2) システム要件定義に記述されていない、非機能要件の実装

設計、製造、テストでの、進捗の報告について、時期、内容、終了の判断基準、進展の条件などの合意を事前に行うことも必要である。同様に終了時での判断基準、終了条件を事前に行うことが必要である。

この工程におけるベンダの説明事項と手順、確認ポイントを以下に整理する

### 説明事項

- ・システム開発関連ドキュメント

システム開発に関連して生成されたドキュメント。

### 保証内容、保証期間（プロダクトライフサイクル）、トラブル発生時の対応窓口、保証対象とならない場合に関する説明事項

システムは瑕疵担保責任の範囲内で不具合の修正が行われる。ただし、システムの不具合は、瑕疵なのか仕様なのか、また誰の責任範囲で起こった問題なのか切り分けることは非常に難しい。障害分析などの初期作業を円滑に行うためにも、開発ベンダと保守契約しておくことが望ましい。保守契約を結ぶことにより保守窓口も明確になる。

### 成果物イメージ等

- 1) 提案時の説明事項、手順

提案時には、業務要件定義書で示されない項目の扱いを明確にするとともに、仕様変更発生時の手続きや費用の扱いなどについて説明を行う。

- 2) 工程終了時の説明事項、手順

設計書および付帯文書、重要事項説明書の説明をもって工程を終了する。

- 3) 最終的に双方が確認するための合意プロセス

契約形態に依存するが、基本的には、外部ベンダもしくは業務部門が作成したものを情報システム部門がレビューし、最終的に、業務責任者が承認を与えることが必要である。ベンダは作業完了報告書を作成し、ユーザは検査をした上で検査合格通知書兼検収書を作成する。

#### 運用・保守との連携に関する留意事項

このフェーズでは、早い段階から運用・保守担当者との調整を行っておくことが望ましい。テストフェーズにおいて、運用上の問題を明確にする。

#### セキュリティ・可用性に関する留意事項

このフェーズで作るシステム要件定義が実現されているかの確認を行う。

#### その他留意事項

受入テスト、検収については、実際の運用を想定したシナリオテストをベンダとユーザにおいて事前検討するとともに、シナリオテストに使用するデータによってテストの信頼性が大きく異なることから、使用データについてはベンダと十分な協議が望まれる。大規模なカスタマイズなどの場合、進捗の報告について、時期、内容、終了の判断基準、進展の条件などの合意を事前に行うことも必要である。

### 2.4.3. 移行・運用準備プロセス

このフェーズで契約する場合の成果物は、修正済みの設計書、運用マニュアルなど付帯ドキュメントである。ここで要件の実装状況が確認される。

- 1) 移行運用ドキュメントの不備。
- 2) プロジェクトゴールや業務要件と納品物のギャップ

この工程におけるベンダの説明事項と手順、確認ポイントを以下に整理する

#### 説明事項（報告書に下記項目を記載もしくは別添）

- ・システム要件定義書  
各システム要件が実装されていることを確認する。
- ・運用マニュアル  
ベンダからの提案内容の要求への整合状況、パッケージソフトウェア選定の考え方について説明を行うもの。
- ・各種設計ドキュメント  
保守が可能なレベルで記述された設計ドキュメント。
- ・重要事項説明書  
各重要項目が実現されていることを確認する。

保証内容、保証期間（プロダクトライフサイクル）、トラブル発生時の対応窓口、保証対象とならない場合に関する説明事項

移行や運用の保証は、通常は開発の保証に含まれる。

## 成果物イメージ等

### 1) 工程終了時の説明事項、手順

テスト結果の報告書及び教育訓練の報告をもって工程を終了する。成果指標を設定していた場合には、その検証も実施する。

検収のチェックリストを活用する。

### 2) 最終的に双方が確認するための合意プロセス

契約形態に依存するが、基本的には、外部ベンダもしくは業務部門が作成したものを情報システム部門がレビューし、最終的に、業務責任者が承認を与えることが必要である。

## 運用・保守との連携に関する留意事項

実現されているか確認を行う。

## セキュリティ・可用性に関する留意事項

実現されているか確認を行う。

## その他留意事項

受入テスト、検収については、実際の運用を想定したシナリオテストをベンダとユーザにおいて事前検討するとともに、シナリオテストに使用するデータによってテストの信頼性が大きく異なることから、使用データについてはベンダと十分な協議が望まれる。

### 3. サービス調達(SaaS、ASP)

#### 3.1. 概要

ユーザが自己保有するシステムにアプリケーションを導入し、自社もしくは委託先に運用を委託する従来の情報システム購入モデルがある。それに対しネットワークを通じてベンダが所有するシステムの上でサービスを利用し業務を行う情報サービス利用モデルがある。

また、SaaS はネットワークを通じて情報システムによるサービスを受けるが、システム開発と同様の特徴がある部分と差異がある部分がある。その特徴を整理する。

	SaaS	システム開発	備考
導入	低価格ですぐに導入できる	SaaS などに比べて導入までの時間がかかる	
サービス変更	月次等一定期間を区切りにして変更できるところと、できないところがある	一旦開発したら変更することはできない	システム開発の場合、サービスの変更は保守または変更になる。
途中解約	可能なところと、一定期間は不可なところがある	一旦開発したら、使うしかなない(または廃棄)	
サービス停止時の対応	回復を待つしかない	自社での対応が可能(体制や能力のある場合)	
データバックアップ	オプションサービスの有無による	自由に設定できる	
セキュリティ	ベンダの示した情報で判断	自由に設定できる	
民事再生時の対応	サービス停止	事前調整可能	
天災時の対応	回復を待つしかない	自社での対応が可能(体制や能力のある場合)	
ベンダの瑕疵	ベンダにより対応が分かれる	責任を問える	

#### 1) メリット

サービス導入が短期間にでき、高信頼なサービスが利用できる。また常に最新のバージョンを使い続けることが可能である。

#### 2) デメリット

個別対応ではなく共同で利用する仕組みなので、自社独自などの機能追加要望などに対して対応できない場合がある。インフラなどの信頼性は高いが、障害発生時に自分でコントロールをすることができない。

従来からも ASP といわれるネットワーク型のサービスは提供されていたが、SaaS では、

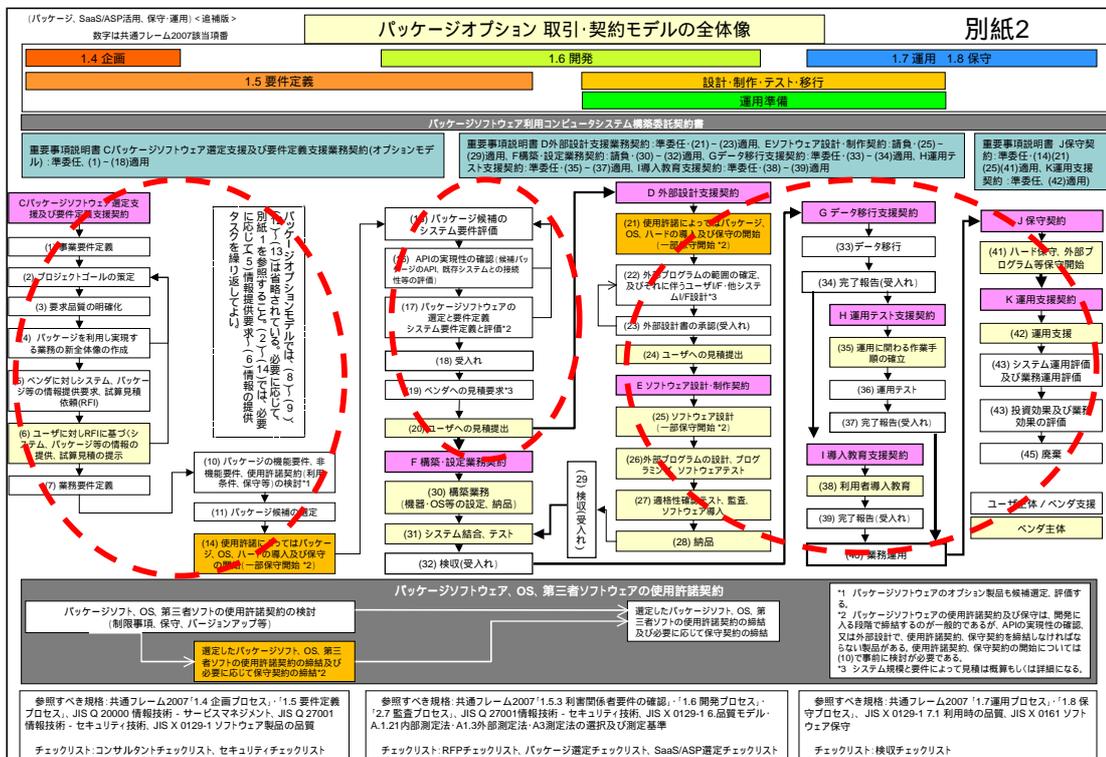
複数企業に同じプラットフォームを使ったサービスを行うマルチテナント方式を使うとともに、Three-Tier と呼ばれる。ユーザの画面、処理のプロセス、データベースが分離され、ユーザは画面表示のブラウザしか持たないモデルが一般的である。

実現形態	システム アーキテク チャ	ユーザ・インタフ エース	業務ロジック	データベース
		画面表示、イベン ト発生	画面遷移、デー タ処理、DB アクセス	DB 管理
自社システム	Stand-Alone	Client	Client	Client
ASP	Two-Tier	Client	Client	Server
SaaS	Three-Tier	Client	Server	Server

システムの導入ではなくサービスの利用になるので費用の構造もこれまでと変わってくる。サービス利用契約を行っている期間に定額料金を支払う定額料金型とユーザ数やデータ量によって費用を変動させる従量型の課金が一般的に導入されている。

### 3.2. 主要プロセス

SaaS では、開発ではなくあらかじめ提供されている機能を利用するという形態をとるためパッケージソフトウェア開発のように方針などの検討や機能の検討は短期間で一括して行われることが多い。また、開発がなく、各種設定と移行作業などが行われ導入される。一般的なプロセスを記述する。

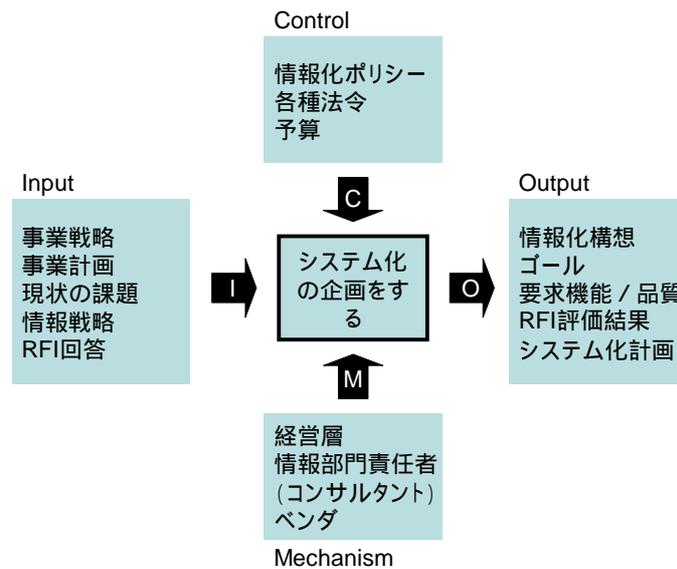


点線部分が一括して行われる。

### 3.2.1. 企画プロセス

このプロセスで、業務改革の責任者やシステム責任者は、システムを通じて実現したいことを明確にする。また、ユーザ担当者は、インターネット、書籍、雑誌などを通じて導入できる製品やサービスの情報を収集（カタログ収集等）するとともに、ゴールを達成するための基本方針を検討し、概要の計画を作成する。

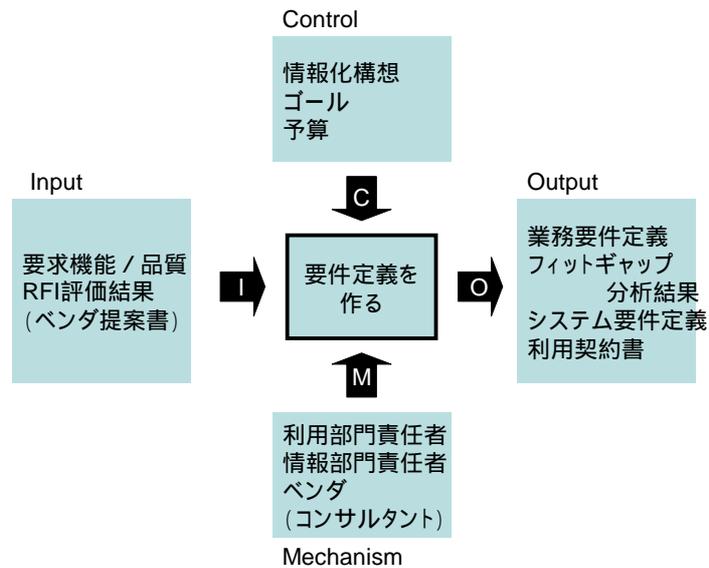
このプロセスの流れは、パッケージソフトウェア開発と同じであるので記述を省略する。（2.2.1 参照）



### 3.2.2. 要件定義、開発(システム要件定義)プロセス

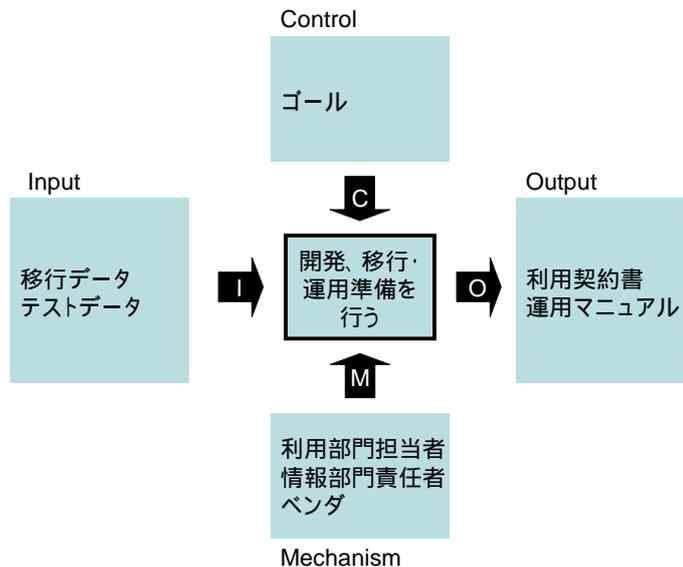
このプロセスで、ユーザ担当者は、業務に必要な機能を明確にするとともに、現状の業務量や業務の流れを示して、ベンダに対して対応可否などの提案を依頼する。（グループウェアなどのカスタマイズを要しないシステムでは、必ずしも提案を求めない）また、ユーザ担当者は、仕様（ベンダから提案があるときは提案）を比較検討し、サービスを選定する、このとき、多くの SaaS ベンダでは試行利用を可能としているので、試行の中で機能を確認する。

また利用するオプションや、社内の他システムとの連携方法を定義していく。



### 3.2.3. 開発(設計・製作・テスト)、移行・運用準備プロセス

利用契約を結んだ上で、ユーザ担当者はオプションなどの機能の設定を行うとともに、他システムの連携などがある場合にはその確認をする。また、研修を行い利用部門の習熟を図る。



### 3.3. SaaS での課題と必要な対応

これまでの ASP の導入、SaaS 導入を通じて、ユーザの不満や要求はあるものの契約上のトラブルはほとんど報告されていない。システムのようにこれから未知のものを構築するのではなく、既にあるサービスを納得して使うというのが SaaS のポイントであり、そのため契約上のトラブルまでは発展していない。しかし、利用者にまったく不満や不安がないわけではなく以下のような注意点に配慮する必要がある。

## 1) 機能不足

サービスの機能をユーザが理解していない、もしくは誤解が生じる場合がある。ユーザ側の「こういう機能を期待していた」「標準的な機能かと思っていた」などの期待値がそのままクレームへと発展する場合があるため、ベンダは、サービスの機能について詳細にユーザ側に説明する必要がある。ユーザも試行利用期間等を通じて検証を十分に行う必要がある。

## 2) サービスの停止（ベンダ都合）

ベンダ側の都合（サーバメンテナンス、プログラム修正、バックアップ作業など）で一時的にサービスを停止する際に、「事前に聞いていなかった」「その時間帯はサービスをどうしても使いたい」といったユーザのクレームが起こる場合がある。ベンダは、契約時にその旨をしっかりと伝え、さらに停止の際にはユーザに事前に承諾もしくは通知する必要がある。ユーザも停止があることを理解し、業務を停止したり、社内にローカルバックアップしたデータで代用するなど、必要に応じて対策を講じる必要がある。

## 3) サービスの停止（故障など）

災害時や通常環境でのハードウェアの故障、ソフトウェアの不具合などでサービスが停止してしまう場合がある。ベンダ側は停止しないよう最大限の努力を行う必要があるが、発生してしまった場合は速やかに復旧の作業を行うとともに、回復予測などの情報公開を行う。また、ベンダは復旧にかかる時間（保証時間）、その場合の課金の扱いを利用契約などであらかじめ明示しておく必要がある。

## 4) サービスの中止

サービス提供会社の倒産などでサービスの維持が難しくなった場合、サービスそのものが中止される場合がある。事前に倒産時のサービス維持についての契約事項を盛り込むことが必要である。

## 5) レスポンスの遅延

サービスレベル自体が通信環境に左右されるという特性を持っているため、ユーザ側の通信環境によっては満足いくレスポンス速度が得られないという場合がある。事前に実環境での運用テストを行う必要がある。また、ベンダは利用規約、場合によっては契約書にサービスの実行環境についての条件を明示する必要がある。

## 6) 個別環境（ハードウェア、ソフトウェア）での不具合

ユーザ側の個別の環境（ハードウェア、ソフトウェア）ではそれぞれソフトウェアやハードウェアの組み合わせによる競合によってサービスがうまく動かない場合がある。ユーザは、事前に実環境でのテストを行う必要がある。また、ベンダはあらかじめサービスの実行環境について詳しい条件を明示する必要がある。

## 7) 途中解約とスケールダウン

ユーザ側の都合により途中解約やユーザ数の減少などのスケールダウンをする場合がある。この場合の契約条件や手続きが明示されていないと、ユーザに予想外の違約金などが発生することがある。契約前に確認をする必要がある。

## 8) データの保全

ベンダがデータの流出、誤消去などを行ったときの補償範囲を明確にしておかないと事故発生時に問題が発生することとなる。契約等で確認をする必要がある。

## 9) データ移行

ユーザの都合で他者サービスや自社システムへの変更などを行うときには、これまでのデータを移行することを求められる場合が多い。データ移行のためのデータの変換または、移行ツールの提供をしているかを、ベンダはユーザに明示する必要がある。

## 10) マッシュアップサービスでの不整合

複数のサービスを組み合わせて使うマッシュアップサービスでの不整合は標準化されたインタフェースを使う SaaS では通常おきないが、これまでに事例などがある場合には、ユーザに開示する必要がある。

## 11) SaaS プラットフォームにおける障害

SaaS 提供企業が他社のプラットフォーム上でサービスを提供する場合、プラットフォーム停止時の責任を明示する必要がある。

## 12) 価格改定

サービス利用途中での価格改定が行われると、ユーザにとっては受諾せざるを得ない場合も多い。価格改定の方針、猶予期間などの有無等についてユーザに明示する必要がある。

### 3.4. 関係者の役割分担

システムの検討から導入まで一貫してユーザ責任によって作業を実施する。外部専門家が導入を支援する場合でも、基本的にユーザ業務の実装だけが業務であり、また、利用開始後にユーザがサービス内容を変更することも可能なことからユーザの責任によって作業を行う。

SaaS ベンダの責任範囲は利用規約に記述されている範囲内であり、その内容を十分に説明する義務を負っている。このためユーザは利用時でのリスクに関して十分かつ詳細な確認事項を準備してあたる必要がある。

### 3.5. ベンダの説明事項と手順

SaaS 導入時の説明は、ホームページによる説明と書面の利用規約（利用契約）のみで行われる場合も多い。また、小規模での試行導入からスタートするユーザも多いことから、トラブル以前にサービスをすぐに中止するユーザもいる。前項で記述された課題を回避するために、ベンダは以下の説明事項をユーザに明示していく必要がある。

#### 説明事項

- ・ サービス機能の詳細説明

提供する機能について記述したもの。

- ・利用時の規約（制限事項など）

停止予定など利用時に制限事項があることなどを記述したもの。

- ・安全性

安全性についてどのような対策を採っているか記述したもの。

- ・データ保全方法

データのバックアップ方法や管理方法について記述したもの

- ・運用保証

どのくらいの稼働率を保証するのか、または、実績を公開するのかを記述したもの

- ・運用体制の説明

ユーザ側に必要な運用体制について記述したもの

- ・移行・導入作業の流れ

移行と運用の流れについて記述したもの

## 保証内容、保障期間（プロダクトライフサイクル）、トラブル発生時の対応窓口、保証対象となる免責事項に関する説明事項

サービスは利用契約に記述された範囲で保証されるが、一般に、インターネットプロバイダなどの SaaS ベンダに起因しない基盤に関する障害に関しては保証されない。

## 手順

ベンダからの情報が書面などによる場合には、付録のチェックリストにより確認を行っていく。不明な点に関してはベンダに問い合わせを行う。ベンダはチェックリストに提示されている事項について情報開示していくことが望まれる。

その後納得したら、利用契約を締結する。

## 運用・保守との連携に関する留意事項

サービス停止時での業務継続方法については、ユーザ側で考慮が必要。

## セキュリティ・可用性に関する留意事項

ユーザから見えないところでシステムが稼働しているので、データのバックアップ体制などに関する確認を念入りに行う必要がある。各種認証の有無、ディスクの多重化、データ拠点の多重化、アクセス管理方法などを確認する。特にデータが国外にある場合には、国内法が適用されない場合があるので留意が必要である。

## その他留意事項

なし

## 4. アジャイル開発・プロトタイピング

### 4.1. 概要

情報システムが企業内の業務で使われるだけでなく、インターネットを經由して一般の利用者にも直接利用されるようになって久しい。今では非常に多くの人がオンラインショッピング、オンラインゲーム、株などの金融取引や行政手続までをインターネットを經由して利用している。情報技術は日進月歩し、利用者の数は急激に増加した。新しいサービスも次から次に生み出されている。言い換えると日々進化する情報システム、新しい価値を提供できるシステム、予測できない事態に早急に対応できるようなシステムが求められるような状況になったといえる。また、オープンソフトウェアに代表されるように、実現する技術についても取捨選択していく時代になった。このような状況を考えると、従来のように結果をしっかりと予測しーから順番に積み上げていくウォーターフォール型の開発では予期せぬ事態に即座に対応できない、もしくはもっと別な方法のほうがより適切であるというようなケースも生まれてくる。そこで採用されることになったのがアジャイル型開発という考え方である。

アジャイル開発はまず目的と、その目的を果たすための主要な機能を実現し（プロトタイプ作成）必要に応じて適宜修正を加えていく考え方である。常に利用者に対しては稼動する完成品の形を成している。当然、計画的に機能を追加していく（反復開発的要素）ようなケースもある。重要なことは目的につながるシステムの価値が常に提供され続けることである。アジャイル型開発における反復は、その結果として何らかの価値が提供されなければならない。

#### 1) メリット

変化の激しい環境において、ゴールに向けて短期間に臨機応変にサービスが提供できるので、業務に対する価値を提供しやすくシステム開発リスクが小さい。

#### 2) デメリット

ゴールに向かってとはいえ、予算の全体像を立てにくい。また、ユーザとベンダとの強固な信頼関係が重要である。

アジャイル型開発や反復繰り返し型と呼ばれる開発手法は、実際に機能するソフトウェアの提供に重点を置く手法である。実際に機能するという意味で、そのソフトウェアの実用度を早期に確認できるという特徴がある。

ウォーターフォールモデルは上流から下流にむけて企画 要件定義 基本設計 詳細設計 運用テスト 運用というプロセスを踏むことから、万一、要件定義そのものに誤りがあった場合、その誤りはリリースまで判明しないという構造的欠陥が指摘されている。言い換えれば、ウォーターフォールモデルでは、要件定義、外部設計でのレビューにおいて、設計書段階でのシミュレーションと確認を重ねることによって、手戻りを防止するモデルである。さらに、要件定義確定後の業務の変更や拡大が少なく、変更があっても予想可能な範囲にとどまるビジネスモデルであるとともに、発注者側が業務全般の説明能力に長けていることが信頼性確保の前提である。

アジャイル型開発は要件定義そのものを、ユーザとともにプロトタイプの開発を通

じて行ってしまおうという考え方にたつ。いわば、対話型のプログラムやユーザ・インタフェースを実環境で実際に構築し、使い勝手やレスポンスの評価を得て、要件そのものを定義していくアプローチである。ユーザ自身で要件定義が困難であったり、ビジネスの範囲が成長、拡大している際に有効なアプローチである。反復繰り返し型異なるのは、オンサイト（ユーザのオフィスや実際のシステムの使用場所）開発、一定期間の間に顧客が選択した機能を作りこむなどの、独特のプラクティスに従う点にある。

機能要件を決定し、非機能要件を満たすリソースを求めるという意味で、銀行のATMシステムの開発においては、ウォーターフォールモデルが明らかに優位である。あらかじめ、ATMで提供される機能は要件として定義が可能であり、画面のレスポンス等の非機能要件もユーザがビジネス上の重要な要件として定義することができる。こうした機能要件、非機能要件を実装できるハードウェア、通信環境、システム環境を選択し、必要なリソースを確保して開発にかかることで、運用テストにおける諸問題がどこにあるのかを追求するのも容易であると言える。

反面、ユーザ自身が機能要件を決定できない場合や、応答性や画面遷移といったユーザビリティが顧客満足度や生産性に影響を与えるシステム構築では、ウォーターフォールモデルのシミュレーションでは限界がある。反復繰り返し型開発やアジャイル型開発は、このような状況において適用が可能な開発モデルである。

#### 4.2. 主要プロセス

アジャイル開発手法のプロセスを図示すると次のようになる。一旦完成したシステムは、さらに要求分析を繰り返し、改良を続けていくような形となる。ある意味では完成することはないとも言える。



システムに求められる価値は常に提供され続ける。結果として時間をかければかけるほど価値のあるシステムとなることが求められる。一方で価値にならない機能を提供することがあってはならない。そのため、価値のある機能の検討精度を上げるためにも、リリース後の効果測定を行いながら要求を検討することも必要である。

#### 4.3. 企画・開発時の要点

主要な参加メンバーが実現するサービスや業務について熟知していることが前提に

なる。さらに実装をするメンバーを含むメンバー間のコミュニケーションが重要になる。結果としてプロジェクトメンバーの数には制約が出る、参加できるメンバー個々の能力もそれなりに要求されることになる。

アジャイル開発手法を採用する場合には何を實現しようとしているのか、開発しようとしているシステムの目的はどこにあるのかというような点を十分に検討して採用すべきである。明確な目的が設定されていないと、要求を判断する際の指標が無いため、正しい判断が行えないためである。目的が明確でなければ、検討の方向性が曖昧になり、結果として出来上がるシステムも曖昧な形になることが多く、結果として無駄な投資がかさむことが証明されている。目的が複数設定されている場合は、目的に優先順位を設定することが求められる。実際のプロジェクトでは複数の目的が設定されていることも珍しくない。

目的に繋がる価値の検討において、しばしば議論になるのは、価値に繋がる要求の大きさである。ビジネスの価値とその要求の大きさは直交する関係にあるため、要求の大きさは大小まちまちである。また、この要求の大きさは開発計画に密接に関係する。

あらかじめ達成すべき目的が予測可能な時、つまり誰がいつ、どのような使い方をし、どれくらいのトラフィックがあるのかなどがわかっているとき、また将来変動するような可能性が無いような場合にはアジャイル開発手法を採用するメリットは生まれてこない。

以下、アジャイル開発手法の代表的なものとされるエクストリームプログラミング（以下 XP）について重要とされる点は以下のようなものである（参考：エクストリームプログラミング入門、wikipedia）

（XP における基本的事項）

XP では次の「4つを価値」が必要なものとして挙げられている。

1) コミュニケーション

顧客、管理者、プログラマがそれぞれに対し即時に率直に質問できること。

2) シンプルであること

動作させるために必要な最低限のところから始めて必要なものを付け加えている戦略をとること。

3) フィードバックが常になされること

常にテストによって確認し、評価し、計画を修正し、結果を見せていくこと。

4) 勇気を持つこと

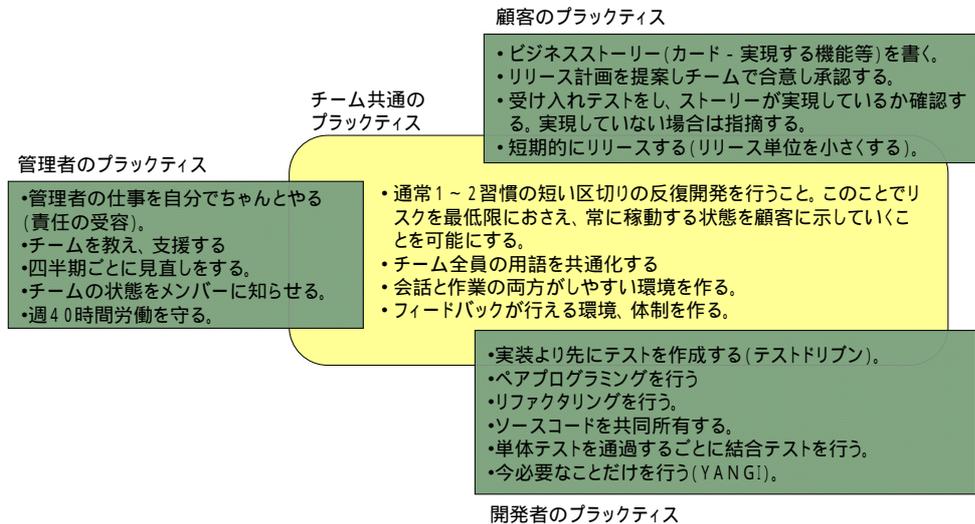
上記の3つを前提にして、大胆な取捨選択、改善、後退、挑戦を行う勇気を持つこと。

#### 4.4. 関係者の役割分担

従来、ウォーターフォール型開発では、開発依頼者が開発者に対してシステム構築を丸投げするような形を取ることがあった。開発依頼者はどういうものが出来上がるかということ、どういう機能を実現したいかということに対して常に決断と決断の責任を負う。開発者も要求される要件に対してその実現の可否の判断を速やかに言い、採用できる技術に基づき合意できる決着点を示す必要がある。したがってシステム化の

対象となるビジネスを深く理解している人間が深く関与することは、アジャイル開発には特に欠かせないものである。

同様に、アジャイル開発手法の代表的なものとされるエクストリームプログラミングについての関係者のプラクティスは以下のようなものである（参考：エクストリームプログラミング入門、wikipedia）



これらの中には単独でも機能するものを多く、部分的な採用も多く行われている。

但し、実装の部分に関連しているものが多く、要求定義の部分はビジネスストーリーとして与えられるという点には注意を要する。

#### 4.5. ベンダの説明責任事項

開発依頼者に対して、採用する開発手法の選択に当たって十分な説明を行いその合意を得なければならない。さらに採用するにあたっては、開発依頼者とベンダー（開発者）側はお互いの役割分担とコミュニケーションの確立に当たってお互いに十分な理解を得なければならない。特に開発依頼者がアジャイル開発手法に対する理解、開発への参加の度合いが非常に濃密なものになることを理解することは不可欠である。

#### 4.6. 留意事項

アジャイル型は、反復繰り返し型をもとに発展的にさまざまなプラクティスや原則を定義したもので、さまざまなモデルが存在している。モデルによっては大規模開発に向かないと言われているものもあるため、留意点が異なってくる。

アジャイル型では、開発手法によってさまざまなプラクティスが提唱されており、ユーザとベンダがそのプラクティスを正しく理解することが信頼性確保においては重要である。代表的な手法であるエクストリーム・プログラミング(XP)では、オンサイト開発や、テスト手順を定めてから実装を行うテスト駆動型開発、反復ごとにユーザによる受け入れテストを行うなどのプラクティスが定められており、従来のソフトウェア開発の慣行とは大きくかけ離れた特徴がある。開発者にとっても高い技術力と従来にない管理を要求することから、反復繰り返し型の留意点を踏まえ、ユーザ、ベンダの双方のプラクティスの実現方法、管理方法を契約書に明記する必要がある。

動作するシステムから得られる効果を検証しながら要件定義が確定し、プログラムが完成した段階で設計が完成するため、ソースコードの保守性、可読性を高めるためのリファクタリングと呼ばれる作業が、信頼性向上においては大きく影響を及ぼす。リファクタリングの実現方法について、ユーザ、ベンダでの詳細な取り決めは、将来の保守性を高めるために有用である。

反復繰り返し型、アジャイル型を含め、プラクティスにドキュメントが定められていないことを理由にドキュメントを作成しないことは誤りである。可読性の高いコードの創出とともにユーザにおける信頼性、保守性を確保するためのドキュメントの作成は重要である。XP においてもテクニカルライタの重要性が指摘されており、プロジェクトの特徴と優先順位を鑑みたドキュメントのあり方を、ユーザ、ベンダで合意することに留意されたい。特に取扱説明書のようなプログラムで表現できないドキュメントについては、それ以外のドキュメントが事前に作成されない関係上、より精緻なものが要求されることとなる。

## 5. 繰り返し型開発

### 5.1. 概要

システム開発を行うとき、さまざまな事情から、最初にすべて実現すべき内容が決定できるとは限らない。そのような場合、すでに内容が決定している主要な機能から順次、開発、実現していくような形が考えられる。実際のシステム開発は現状ではウォーターフォール型の開発といえども何らかの意味で繰り返し型開発的な要素を持っているといえる。大きな違いは、開発されるユニットを 1 つの価値としてそれぞれに確立していくという点である（契約単位にすること）。

### 5.2. 主要プロセス

次に図示するようなプロセスである。



### 5.3. 企画・開発で起こるトラブル

どういうシステムが求められているのか明確に開発依頼者と開発者が合意しなければならないことなどウォーターフォール型開発と基本的にまったく同様である。利用者に対してリリースできるレベルでテストまでを含むプロセス単位を小分けにするので、プロジェクトとしてのリスクを軽減することが出来る。

反復繰り返し型の場合は、上流から下流にむけての流れを繰り返すことによって、「統合・テストがすんで安定している『部分として』完成したシステムをリリースすることである。」とされている。言い換えれば、初期に仕様を完全に定義するのではなく、開発構想書や概要レベルの要求一覧表などをもとに、数週間単位でイテレーション（要求分析、設計、実装、テスト）を繰り返し、段階的に詳細化して仕様の完成度を高めていく開発手法である。イテレーションの流れはウォーターフォールモデルに準じており、企画、要件定義の不備は運用テストで明らかになることから、これらの不備に対しては次のイテレーションの企画、要件定義で反映されることになる。同様にイテレーションにおける実装段階では、要件の変更は行わず、次のイテレーションでの課題として引き継がれていく。こうすることで、定められた期間での進捗と完成度を求め、全体の工数見積りに反映させていく。

・反復繰り返し型の場合でも、要件定義が信頼性に大きく影響を及ぼすことは同様である。開発構想書や概要レベルの要求一覧表で想定していなかった課題を要求事項として反映するための企画、要件定義プロセスをおろそかにすると、無用のイテレーションを繰り返すこととなる。また、次のイテレーションへのフィードバックを得る

ためのテストにおけるユーザの関与（評価）方法とテスト計画の変更管理、新たなイテレーションでの企画、要件定義の変更管理が必要である。一連の管理が不十分なままイテレーションが繰り返されると信頼性が大幅に損なわれることに留意する。

・要件定義の先送りなどによる無用なイテレーションを防ぐため、実現が必要な機能の優先順位の設定と、量的制限 を契約書に明記しておく必要がある。ユーザとベンダで、見積における積算根拠と作業結果の評価方法を事前に合意しておき、あらかじめ定めた反復回数で作業の見直しを行うことを契約に定めておくことが重要である。場合によっては、期間を定めた数回のイテレーションを見積のために契約し、その後、詳細見積、契約を締結するなどを検討すべきである。

#### 5.4. 関係者の役割分担

ウォーターフォール型の開発と基本的に同様である。

#### 5.5. ベンダの説明責任事項

ウォーターフォール型の開発と基本的に同様である。

## 6. 付録

### 6.1. 付録 1 パッケージソフトウェア開発でのトラブル例と必要な対応

企画設計に起こるトラブルは、RFP に起因する内容とその後の要件仕様確定によって起こることが多い。以下が多くの場合で指摘される主な課題である。

- (1) 不十分なRFP（要求仕様書）
- (2) 過大なモディファイ、アドオン
- (3) パッケージソフトウェアの機能・サービスレベル不足
- (4) 仕様外の要求
- (5) 検収時点での要求不一致
- (6) 既存、追加ソフトウェアとの不整合
- (7) 優越的地位の利用
- (8) 知的財産の帰属
- (9) 開発中止時の精算
- (10) パッケージソフトウェア間インターフェースに起因する問題
- (11) システム内で障害が切り分けられない場合
- (12) パッケージソフトウェアのバージョンアップに起因する問題
- (13) パッケージソフトウェアのサポート切れの問題

(1) 不十分な RFP (提案要求書)

ユーザの作成した RFP の内容が不十分であり、提案内容が絞りきれない場合がある。そのため提案内容に過不足が発生し、その後の要件定義の段階で詳細化しないまま進めると、設計途上や検収において、どちらの責任か問題となることがある。ユーザ側の意見として「プロならば RFP や仕様の不十分さを専門知識で補完せよ」、ベンダ側の意見として「仕様の確認はユーザとベンダとの間で書面で交わしているので、記述されていないことは対象外」といった議論が起こることが多い。

ユーザ側は、業務の暗黙知を仕様で表現できていないことも多く、また、ベンダの技術的な提案を理解できていない場合が多い。ベンダも、説明や要件を聞きだす努力が不足していることも見られるため。十分な留意をする必要がある。

また、RFP の作成に外部専門家を活用する場合もあるが、外部専門家が必要十分な仕様を作成できていない場合もある。コンサルティングを依頼するときには成果物イメージと記述レベルを契約前に十分に打ち合わせする必要がある。

	ユーザ	ベンダ	外部専門家
原因	RFP の記述方法を知らない。 RFP を作る時間が取れない	不十分な RFP を容認している RFP の内容を勝手に解釈している	RFP 作成支援において、十分な RFP を作っていない場合がある
主な対策	ユーザの業界に精通する第三者による RFP 評価・監査的なものを入れる	要件定義にする段階で十分な詳細化を図る	ユーザの業界に精通する第三者による RFP 評価・監査的なものを入れる
契約での留意点	RFP に記載されていない追加変更事項の扱いを契約書に記載する	同左	RFP に必要な項目が記載されない等、明確な不備があった場合の善管注意義務があることを契約締結時に確認する。

## (2) 過大なモディファイとアドオン

フィット&ギャップ分析後にパッケージソフトウェア選定を行うが、この後の詳細要件を定義していく中で、各種パラメータの設定、パッケージソフトウェア本体に改造を加える「モディファイ」、パッケージソフトウェアに機能を加える「アドオン」を実施することとなる。このときにユーザが過大な要求をすることにより、システム全体のオリジナルのパッケージソフトウェアの占める割合が低くなり、システムの信頼性、安全性、性能が低下する要因となるとともに想定外の費用が発生することがある。また、モディファイの影響でバージョンアップが受けられなくなるトラブルも散見する。パッケージソフトウェア選定後であっても、大規模改修が必要とわかった場合には、BPR方針の見直しやパッケージソフトウェア選定のやり直しなど、思い切った対策が必要な場合も出てくる点に留意が必要である。

	ユーザ	ベンダ	外部専門家
原因	パッケージソフトウェア導入のポイントを理解していない。 現場が既存の機能などに固執する	パッケージソフトウェアが提示する業務の優位性を十分に説明していない モディファイ・アドオンの増加が売り上げ増加につながる場合、容易にモディファイ・アドオンを受けてしまう	モディファイ・アドオンの必要性、投資対効果が説明できていない
主な対策	パッケージソフトウェア導入時には、パッケージソフトウェア側の業務パターンにあわせることも重要であることを利用現場に啓発する。	企画段階でシステム化方針としてモディファイ・アドオンを少なくする方策を決定する 設計着手前に、十分なユーザ教育を行う	モディファイ・アドオンに対する実施可否の評価基準やチェックリストを提示する
契約での留意点	契約上の留意点はない。	システムのバージョンアップに制限が入ること等を契約書に明記する。 パフォーマンス低下の可能性を示唆する。	契約上の留意点はない。

### (3) パッケージソフトウェアの機能・サービスレベル不足

フィットギャップ分析時に、当該機能有りと判断された機能について、設計段階においてユーザのイメージと大幅に異なることが判明し、モディファイの要否、その費用の負担などが問題になる場合がある。応答時間や使い勝手といった非機能要件、ユーザビリティにおいて問題が発生することが多い。

特に、あるユーザ企業用に作ったシステムをそのままパッケージソフトウェアと称して販売する「流用品」では機能の整備が十分ではないケースが見受けられる。システム構築の企画段階からパッケージソフトウェアを視野に入れて開発しシステム完成後に、パッケージ化を図る商品も、初期ユーザの仕様に依存していることが多く汎用的になっていない場合があり注意が必要である

	ユーザ	ベンダ	外部専門家
原因	パッケージソフトウェア選定時の検討が不足している	パッケージソフトウェアの機能や制約を正確に説明していない	パッケージソフトウェアの精査が十分でない
主な対策	パッケージソフトウェア導入時には、パッケージソフトウェア側の業務パターンにあわせることが重要であることを利用現場に啓発する。	重要事項説明を実施する。 流用パッケージの場合には特に注意をする 性能などの非機能要件は早めの確認を行う	パッケージソフトウェア選択の評価基準やチェックリストを提示する
契約での留意点	仕様に記述した機能が、既存機能より低下するときや一般常識的に用意されるべき機能が不足している場合の対応を明記する。	パッケージソフトウェアとして提供されるべき基本機能の不足等、紛争対応を明記する。	同左

#### (4) 仕様外の要求

システム開発途中で、仕様を書いていなかったことが要求されることがある。それは、ユーザが仕様を書き忘れていた場合や、事業環境が変化したことに起因することも多い。ベンダとユーザの力関係で作業を押し切られてしまうことも多く、手順の明確化が必要である。

	ユーザ	ベンダ	外部専門家
原因	仕様の書き方や位置づけが理解できていない。  仕様確定後の利用部門からの要望をコントロールできない。	仕様確定時に十分な情報提供を行っていない	パッケージソフトウェアの精査が十分でない
主な対策	仕様を基にチェックを行う。(日本情報システム・ユーザ協会「要求仕様定義ガイドライン」等)仕様の意味付けを利用現場に啓発する。	企画段階でシステム化方針としてモディファイを少なくする方策を決定する。  設計着手前に、十分なユーザ教育を行う。  仕様書に記述がない項目は、仕様変更にあたるという事を啓発する。	パッケージソフトウェア選択の評価基準やチェックリストを提示する
契約での留意点	仕様変更の費用分担判断基準や判断プロセスを明確にする	仕様に明記されていない部分での意識のずれ違いに関する判断基準を明確化しておく	同左

(5) 検収時点での要求不一致

システム検収時点で現場の利用者が来て、これでは使えないなどの指摘をすることも多い。企画、設計などを情報システム部門にまかせっきりのユーザにおいて多発するトラブルであり、フローのあり方や操作性などの指摘を受けることも多い。仕様では明確に記述されていないことも多く、トラブルに発生することが多い。

	ユーザ	ベンダ	外部専門家
原因	システム部門がユーザ部門と十分な確認を取っていない。  完成イメージを勝手に思い込み、ベンダに伝えていない。	完成イメージを勝手に思い込み、ユーザに伝えていない	-
主な対策	プロトタイプやシナリオモデルによるレビューを行う。	プロトタイプやシナリオモデルによるレビューを行う。	嗜好や感覚に属する部分が多い事、現物（プロトタイプなど）での確認の重要性を説明する。
契約での留意点	契約上の留意点はない。	明文化されている仕様を実現していない場合はベンダの責に帰すべき要求不一致であり、明文化されていない仕様についてはベンダに帰すべき責でない旨を記載する。	-

## (6) 既存・追加ソフトウェアとの不整合

システムの開発環境では問題なく稼働していたが、実運用機で動かしてみると既存システムと新システムで使用しているドライバが不整合を起こすなどで正常に稼働しないことがある。逆に、追加システムを入れたことにより納入済のシステムが動かなくなることもある。また、セキュリティ対策などによるOSなどの変更もシステム障害を引き起こす可能性がある。

	ユーザ	ベンダ	外部専門家
原因	稼働環境をベンダに事前に連絡していない。 テスト環境を提供しない。	事前に環境確認を行っていない。	-
主な対策	事前に稼働環境をベンダに開示する。	事前環境調査を実施する。	ソフトウェアはその稼働環境の設定が繊細である事を十分に説明する。
契約での留意点	事前環境の確認を仕様に記述しておく。	システム開発終了後の別システム追加によるシステム障害は免責事項として記述する。	-

## (7) 優越的地位の利用

ユーザとベンダは、本来は対等なパートナー関係を築く必要があるが、発注側であるユーザが相対的に優越的地位になることが多い。そのため、価格や仕様変更のリスクに備えフェージング契約をしていたとしても、後工程での金額の交渉ができない場合も多い。そのため、初期の概算見積りのまま、開発をせざるを得なくなりトラブルになることがある。

また、グループ経営をしている企業の情報システムを構築するときに、企画まではグループの中核会社を実施し、その後、グループ会社と個別契約をすることもあがあるが、責任主体が明確でなくトラブルに発展することがある。

	ユーザ	ベンダ	外部専門家
原因	当初合意していた事項を変更する。	契約条項などに記載していても、強く主張していない	-
主な対策	合意内容を変更しない。	トラブルになりそうな項目は、契約書や議事録で確認を行う。仕様書に明記されていない項目は、追加または変更にあたるという事を啓発する。	仕様書に明記されていない項目は、追加または変更にあたるという事を事前に啓発する。
契約での留意点	次フェーズで合意に至らない場合の解約条項を入れておく。 また、関連契約が影響を及ぼす場合は明記しておく。 「仕様変更・追加が契約に影響する場合は、契約の変更管理プロセスを用いて対応することを合意する。」ように契約に明記し、関係者に啓発、喚起する。	次フェーズで合意に至らない場合の解約条項を入れておく。 また、関連契約が影響を及ぼす場合は明記しておく。 「仕様変更・追加が契約に影響する場合は、契約の変更管理プロセスを用いて対応することを合意する。」ように契約に明記し、関係者に啓発、喚起する。	-

## (8) 知的財産の帰属

著作権は開発者に帰属することが基本であるが、ユーザの知識不足による著作権保持の主張や、複数社が関わっての仕様検討等により、業務フローなどのビジネスモデルについてユーザが権利を主張する場合もある。特にビジネスモデルについては、特異なモデルではなく汎用的モデルについての権利を主張される場合、ベンダにとっては応じかねることがある。

	ユーザ	ベンダ	外部専門家
原因	必要以上に著作権を主張する	業務秘密をパッケージソフトウェアなどで流用することがある	業務秘密を他社へのコンサルティングなどで流用することがある
主な対策	著作権と秘密保持契約の関係を啓発する	モラル啓発を行う。 パッケージソフトウェア化を前提とした契約を行う（利益の先渡しは行わない）。	モラル啓発を行う。
契約での留意点	モデル契約を使って、常識的な契約を行う。 必要であれば秘密保持契約を締結する。	業務秘密がユーザ固有の物である場合、汎用的で有る場合、業務秘密を流用した場合の扱いを契約書に明記する。	業務秘密を流用した場合の扱いを契約書に明記する。

### (9) 開発中止時の精算

開発中にユーザ、ベンダそれとも双方の何らかの事由により開発を中止にせざる得ない場合がある。そのときに、かかった費用の負担や中止に伴う損害に対して問題化することがある。

	ユーザ	ベンダ	外部専門家
原因	仕様決定の遅れや頻繁な仕様変更など、ベンダが対応できない状況になってしまう。	技術力の不足などでプロジェクトが遂行できない。	左記両原因
主な対策	プロジェクト管理を強化する。ベンダ選定を強化する。	プロジェクト管理を強化する。	プロジェクト管理を強化する。
契約での留意点	開発中止時の解約条件を明記する。	同左	同左

(10) パッケージソフトウェア間のインタフェースに起因する問題

システム構築の際に各パッケージソフトウェアの持つ詳細仕様の整合性が取れないことが明確になり、業務要件やシステム要件を満たさなくなることがある。メッセージの連携ができずに統合運用ができない場合や、システム基本ソフトのバージョンに制約があり、同じハードウェア上では同居できないなどの場合もある。そのため、運用が当初想定していたものより手間がかかるようになり、ハードの重複投資が必要になったり、最悪、システム実現ができない場合がある。

	ユーザ	ベンダ	外部専門家
原因	-	組み合わせに関する十分な調査を行っていない	機能面だけでパッケージソフトウェア選定をしてしまう
主な対策	複数社のパッケージソフトウェアを使う場合には、システムインテグレータなど、その分野の知見を持つ企業に委託するか、パッケージソフトウェアベンダの1社にシステムインテグレーション契約を行う	先行事例の調査を行う。事例がない場合には事前検証を行う。	各社に対して実績などの調査を行う
契約での留意点	障害発生時に対応ができる条項を入れておく。(解約条項など)	-	-

### (11) システム内の障害が切り分けられない場合

障害発生時に傷害の原因がわからない場合には、ベンダからテクニカルサポート料などの調査費用を要求されることも多い。1パッケージソフトウェアしか搭載されていない場合でもOSとの相性などを理由に自社パッケージソフトウェアの非を認めないベンダも多いが、パッケージソフトウェア・インテグレーションで構築されたシステムにおいては、特に、組み合わせ先パッケージソフトウェアの問題として迅速に対応してくれない場合が多い。

特に、原因が最後まで明確にならない障害については修正ができないままユーザが一方的に不利な状況になることがある。

	ユーザ	ベンダ	外部専門家
原因	-	責任回避をしようとする体質	-
主な対策	複雑なシステムにはシステムインテグレータを活用する  迅速な対応を促すため保守契約を結ぶ	障害履歴を蓄積することによる、対応の高度化  サービス姿勢の改革	-
契約での留意点	障害発生時に対応ができる条項を入れておく。	-	-

(12) パッケージソフトウェアのバージョンアップに起因する問題

ミドルウェアのバージョンアップによって、システムのサポートの継続性が得られないこともあるし、機能拡張時に新しいバージョンのミドルウェアに切り替えたいという要求が出るときもある。そのときに、バージョンアップを行った影響が既存の機能に出る場合もある。最近のソフトウェアは自動バージョンアップを行うものも提供されており、注意が必要である。

	ユーザ	ベンダ	外部専門家
原因	-	設計時にバージョンアップを考慮していないことがある	-
主な対策	バージョンアップの可能性やその対処方法を開発初期の段階で確認しておく	バージョンアップの影響を受けにくい設計にする。	-
契約での留意点	-	-	-

### (13) サポート切れ

システム内で利用しているミドルウェアなどのパッケージソフトウェアソフトが、サポート切れで、メーカーから今後の対応が受けられなくなることがある。システム自体の寿命がまだ長い場合には、ミドルウェアの入れ替え、システムそのものの入れ替えを求められる場合もある。

	ユーザ	ベンダ	外部専門家
原因	-	設計時にサポート期限を考慮していないことがある	-
主な対策	サポート切れの可能性やその対処方法を開発初期の段階で確認しておく	当該パッケージソフトウェアの影響を受けにくい設計にする。	-
契約での留意点	-	-	-



## フェーズレビュー チェックシート

### 6.2. 付録2 フェーズチェックリスト

#### フェーズレビュー0（システム化の方向性）

評価者：

分類	質問	評価	備考
事業	事業の目的は明確になっていますか？		
事業	ゴールは明確に定めていますか？		
事業	予算の実施により事業・業務目的は達成されますか？		
事業	緊急性、必要性はあるか？		
事業	自社で実施すべき妥当性がありますか？（他社サービスで代替できませんか）		
効果	利便性向上などの価値を生み出す効果は明確に記述されていますか？		
効果	業務改革効果は明確に記述されていますか？		
効果	投資対効果を評価していますか？		
効果	必要な要求品質は検討され明記されていますか？		
予算	予算額は妥当ですか？		
内容	代替案、システムの依存性、類似性は検討されていますか？		
リスク	リスクは許容できるレベルですか？		
ポートフォリオ	戦略性と実現性を勘案して、適当な事業ですか？		
体制	予算執行に必要な体制は発注側、受注側にありますか？		

評価 A：質問を満たしている

評価 B：不十分な部分もあるが概ね満たしている（要備考記述）

評価 C：質問内容を満たしていない

- : 質問事項に該当しない



## フェーズレビュー チェックシート

フェーズレビュー 1 (システム化計画)

評価者：

分類	質問	評価	備考
事業	事業目標を実現するための主要成功要因が明確化されていますか？		
事業	要求内容は、関係者全体から収集されていますか？		
事業	システム化計画は整備されていますか？		
改革	BPR の方針は明確ですか？ 納期短縮、コスト削減等		
改革	RFI は適正な対象（ベンダ）に依頼されていますか？		
改革	RFI により、各社のパッケージソフトウェアの特徴など、必要な情報は収集されていますか？		
現状評価	既存の業務・システムがある場合、そのまま継続するのは駄目ですか？		

評価 A：質問を満たしている

評価 B：不十分な部分もあるが概ね満たしている（要備考記述）

評価 C：質問内容を満たしていない

- : 質問事項に該当しない



## フェーズレビュー チェックシート

フェーズレビュー 2 (業務要件定義)

評価者：

分類	質問	評価	備考
事業	主要成功要因が業務機能に展開されていますか？		
事業	事業や業務上の仮説に変化はないですか？		
E A	機能分析は、設計に引き継げるレベルで行われていますか？		
E A	情報分析は、設計に引き継げるレベルで行われていますか？		
E A	業務プロセス分析は、設計に引き継げるレベルで行われていますか？		
E A	インタフェース分析は、設計に引き継げるレベルで行われていますか？		
E A	ハード、ソフト、ネットワークは、設計に引き継げるレベルで整理されていますか？		
改革	他者事例の調査はしましたか？		
成果物	RFP はきちんと整備されていますか？		
成果物	外部専門家を使う場合、外部専門家は善管注意義務について理解していますか？		
効果	当初見込んだ効果が得られそうですか？		
予算	当初見込んだ予算内に収まりそうですか？		
引き継ぎ	基本設計に引き継げますか？		

評価 A：質問を満たしている

評価 B：不十分な部分もあるが概ね満たしている（要備考記述）

評価 C：質問内容を満たしていない

- : 質問事項に該当しない



## フェーズレビュー チェックシート

フェーズレビュー3（システム要件定義）

評価者：

分類	質問	評価	備考
事業	事業や業務上の仮説に変化はないですか？		
外部仕様	関係者は明確になっていますか？		
外部仕様	機能は明確になっていますか？		
外部仕様	インタフェースは明確になっていますか？		
外部仕様	開発モデルについての合意を得ていますか？		
パッケージソフトウェア	フィット&ギャップ分析は適正な評価項目で行われましたか？		
効果	当初見込んだ効果が得られそうですか？		
予算	当初見込んだ予算内に収まりそうですか？		
リスク	システムに不測の事態が発生した場合の対策が検討されていますか？		
成果物	システム要件定義書が整備されていますか？		
引き継ぎ	詳細設計に引き継げますか？		

評価 A：質問を満たしている

評価 B：不十分な部分もあるが概ね満たしている（要備考記述）

評価 C：質問内容を満たしていない

- : 質問事項に該当しない



## フェーズレビュー チェックシート

フェーズレビュー4（サービス開始時、移行・運用準備プロセス）

評価者：

分類	質問	評価	備考
事業	事業や業務上の仮説に変化はないですか？		
品質	品質は許容できるレベルまで収束していますか？		
品質	機能として致命的な問題はないですか？		
品質	過負荷に対する試験をしましたか？		
効果	当初見込んだ効果が得られそうか？		
予算	当初見込んだ予算内に収まりそうか？		
リスク	システムに不測の事態が発生した場合の対策ができていますか？		
引き継ぎ	関係者は操作に習熟していますか？		
引き継ぎ	保守体制は明確になっていますか？		
引き継ぎ	サービスレベルの指標は明確になっていますか？		

評価 A：質問を満たしている

評価 B：不十分な部分もあるが概ね満たしている（要備考記述）

評価 C：質問内容を満たしていない

- ：質問事項に該当しない

### 6.3. 付録3 アジャイル開発適用のチェックリスト

	分類	チェック項目	評価
1	開発側	関係者全員がアジャイル開発を理解し、それを採用することに同意していること	
2	開発側	サーバー管理やマニュアル作成等の担当を除く全員がコーディングできること	
3	相互条件	開発に使うソフトウェアを開発者が選べること	
4	相互条件	タスクをチーム内で決定できること	
5	相互条件	実際に作業する人が顧客と話せること	
6	相互条件	無駄な決まりごとを作らないこと	
7	相互条件	チーム全員が自由に発言できること	
8	相互条件	質疑応答が迅速に出来る環境が整備されていること	
9	相互条件	ソフトウェアの設計を書かなくても良いこと	
10	顧客側	ステークホルダーが5人以下であること	
11	顧客側	開発者の要請に応じて時間が取れること	