

# OpenCentral 導入プロセス

## ——基幹系ソリューション導入方法論

### OpenCentral Implementation Process

#### ——Implementation Methodology for Enterprise Application Solution

小 口 修 一 郎

**要 約** 2002 年 3 月に小売本部ソリューションとして「OpenCentral」をジェネラル・リリースした。本ソリューションは小売本部の基幹業務で基本となる全ての業務を備え、その適用範囲は広い。本ソリューションの基本的な導入アプローチは、DeMarco (1978) が提唱した構造化方法論のウォーターフォール型の開発手法を取るが、導入初工程である「要件定義」の失敗が、プロジェクトの納期遅れやコスト増に直結することは言うまでもない。

にもかかわらずソフトウェア業界には数十年の経験がある中、いまだに要件定義を効果的に進める手法は確立されていない。

またシステムが大規模化し経営への結びつきが強くなるにつれて、要件定義の難易度は高くなる。そしてソリューション導入については、そのメリットを最大限実現するために、極力、ソリューションの標準機能を使用し、ソリューションに対する追加開発を最小化することも導入プロジェクトを成功に導く為に重要である。

本稿では当社が培ってきた経験をベースに OpenCentral の導入に関する手順と要件定義工程に特化した導入方法論の構築事例を述べる。

**Abstract** Nihon Unisys Ltd. released an IT Solution package for the retail business, called "OpenCentral" on March in 2002. This solution covers all core corporate functions of the chain office. The introduction of this system to our customers starts the definition of software requirements, based on Tom Demarco's waterfalls development model (1978).

This model has some risks that the failed requirements definition is directly linked to the seriously delayed delivery and increased cost of the project. The IT Software industry has faced these risks for several years and looked for more effective ways to solve them, but has not yet established the effective solutions.

Meanwhile, as the information system becomes more important factor of business decisions, the difficulty of requirements definition is getting higher.

To avoid these risks, we use general functions as much as possible while taking full advantage of our solutions.

This paper describes the standard approaches to development method of OpenCentral based on our experiences and introduces a case of OpenCentral implementation focusing on the requirements definition process.

## 1. は じ め に

当社は 2002 年 3 月に小売業向けに基幹系業務ソリューションとして「OpenCentral」のジェネラル・リリースを開始した。本ソリューションはスーパーマーケット、ホームセンタといった多店舗をチェーン展開している小売業の本部基幹業務で基本となる全ての業務機能を備

え、その適用範囲は広い。

本ソリューションの導入にあたっては、要件定義から始まり、設計、開発、テスト、そして既存システムからの移行を行い、かつ顧客企業の基幹業務機能の全般がシステム化の対象範囲となり 100 人月を超える開発規模になるケースが多く、その導入リスクも高い。

また、ソリューション導入のメリットを最大限に実現にあたっては、ソリューションが既に必要な業務機能を保持しているため、極力、顧客の業務機能をソリューションに合わせる事が重要であることは言うまでもないが、導入企業が他社との差別化を図り、強みである業務機能は、ソリューションに対し追加・変更を加える必要があり、逆にそうでない業務機能は排除、もしくはソリューションの業務機能に合わせていくといった見極めが必要となる。

一般のシステム開発同様に要件定義の品質がプロジェクト成功の可否を決定付け、失敗プロジェクトの多くは、この要件定義の精度の低さに起因するものが多い。筆者は、本ソリューションの顧客への適用をとおり、ソリューション導入工程の特に要件定義に特化したコンサルテーションのサービスレベルの向上化と均一化、そしてそれらを実施する要員育成と体制作り課題を感じてきた。

本稿では、これらの課題を解決すべく、ソリューション導入の要件定義工程に特化し、その導入方法論とその要員体制のあり方についてまとめる。特に導入方法論については、まだまだ今後の顧客適用の中で、改良、改善していく必要があるが、現場で使える方法論であることが重要であると考えます。

## 2. 当社基幹ソリューション OpenCentral の概要

OpenCentral の全体機能を図 1 に示す。小売本部の基幹業務機能として必要な商品管理、特売管理、発注管理、仕入管理、在庫管理、販売管理、買掛管理の 7 サブシステムから構成される基幹モジュールとデータウェアハウスを含む情報分析管理の情報モジュールにより、各サブシステムが統合化されたデータベースにより相互に連携されている統合システムであり、商談から仮説・検証までの小売マーチャンダイジングをトータルに支援している。

また、外部システムとの連携として、店舗システムとの商品・特売情報の送信と精算実績の受信、取引先との EDI による発注データの送信と納品実績の受信、通過型・在庫型物流センター、加工センター等の各種センターとの出荷指示の送信と納品・出荷実績の受信、そして給与システムとの勤怠データの送信、会計システムとの仕訳データの送信といった連携が必須となる。

ここでは各機能の詳細説明は割愛するが、我々が身近に買物をしているスーパーマーケットに並べられている生鮮、日配、雑貨等の商品、特売商品の取引先との商談に始まり、個店別の取扱いとその商品の売価・原価コントロール、これら商品の多種多様な物流形態を加味した発注と仕入、取引先への支払い処理、日次単位の分析を可能する販売管理情報と分析データの提供等、小売本部の基幹業務に必要な機能が広範囲に実装されているソリューションである。

## 3. 基幹系システム導入の現実

### 3.1 基幹系システム導入に期待すること

基幹系業務システムの刷新については、期待する効果は大きい。まずはソリューション・パッケージ選定の利点から期待効果がある。しかし、一般的に言われることであるが、システム

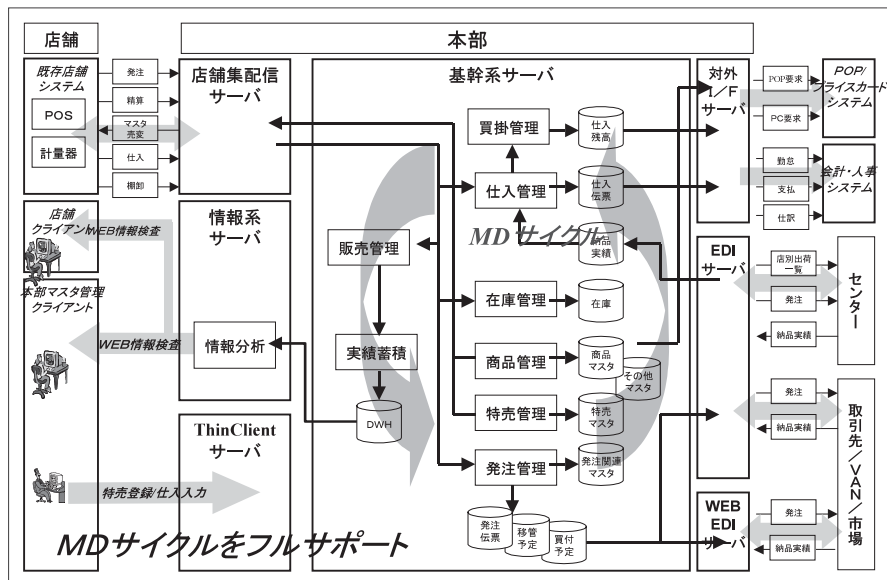


図1 OpenCentralの全体像

開発にあたり、長年にわたりユーザ個々の要望を積上げてきた結果、システムそのものが複雑になり現行システムが限界にきている企業は多い。複雑になった現行システムは、何か改修をするにしてもシステム全体を把握している技術者すら不在となっているケースもあり、かなり困難な状況になっているケースも少なくない。

その点ソリューション・パッケージは、統一されたアーキテクチャで、データベースを中核に一貫した設計思想のもとで開発されている。MDサイクル<sup>\*1</sup>をフルサポートする業界標準の業務プロセスが実装され、これらの業務プロセスの流れに沿ってデータベース構造があらかじめ設計されている。このことは、全ての企業が望んでいるスピーディなシステム全体の最適化とデータの一元管理の実現を可能とする。

次の狙いは、当然のことながら全て企業が直面している経営課題の解決である。必ずしも全ての経営者がシステムの全体最適を望んでいるわけではない。経営者の多くの悩みは売上高や利益率の向上、コスト削減、そして顧客満足度の向上といった経営そのものの経営課題が最優先で挙げられる。こういった経営課題を解決するために、ビジネススピードが企業の業績を左右する今日、大半の企業において新システム化の要件として、「スピード経営」や「見える経営」といったキーワードがあり、最も経営環境の変化に迅速に柔軟に対応できるシステムを重要視する。

これは基幹系業務システムの刷新により効率的な業務運営を可能とするだけでなく、日々発生、蓄積されるデータにより経営の状況をモニターし、計画や実行に変更が求められたときに、迅速に起動修正が出来ることを望む。言い換えると経営において必要なことは、日々のサイクルで管理が可能であり、結果としての業績だけでなく、業績を産み出す過程の業務プロセスの管理を可能とすることである。これらのことは、業務システムの刷新と同時に、各企業の業務プロセスを見直し、業務の簡素化や効率化といったことが重要となる。これらのことの実践により、多くの経営者が望む「スピード経営と変化への対応」の実現が可能となる。

### 3.2 開発プロセスと現場が抱える課題

基幹系業務システムの刷新に対する期待効果は大きい一方で、開発サイドとしての課題も多い。近年、プロジェクトマネジメントの話題をよく耳にするが、その背景にはソフトウェア産業が成熟した現在でも、いまだにシステム開発に対する失敗事例が多いからではないか。そして、システム化の対象が基幹業務系システムとなると、その機能範囲は広く、システム開発の失敗事例も多くなると言える。

最近のシステム開発の特徴として、前述した期待効果の「スピード」と言うキーワードからも、まずは従来のシステム開発に比べて相対的に期待される開発工期が短くなっていることが言える。次に、システム化にあたり、経営課題や業務課題と言った重点的な課題については、システム化の決定について、その判断基準として確認されているケースが殆どであるが、対象とする業務機能については全て検証を行ったわけではない。従って、開発サイドからすると要求仕様が固まっていない状態でシステム開発を始めなくてはならないケースが多い。そして、システム化の範囲が広くなり関連部門が多くなると、必然的にシステム化に対する利害関係者も多くなり、近年のビジネス変化の多いことも加わり、開発段階での要求仕様の追加・変更といったことが継続的に発生する可能性が高くなる。

結果、システム開発にはさまざま開発リスクが取り巻くが、その中でも要求リスクが高まり、開発プロジェクトそのものがコストのオーバーフロー、納期の遅れ、そして収束しないバグの発生等により、失敗するケースが多く発生する。

### 3.3 何故、開発プロジェクトが失敗するのか？

現在でも多くの開発プロジェクトでは、「ウォータフォール型」と言われる開発プロセスが採用されている。この開発プロセスは、要件定義工程でユーザのシステム要求の収集、機能定義を始めて、開発ライフサイクルの間、これらのシステム要求が変わらないことを前提に、各工程の完成された成果物に従い、次工程に進むことが想定されている。仮にこれらの成果物が不十分、または誤ったものであれば、当然、誤ったプログラム・コードへに導かれ、多くのケースでは工程終盤のテスト工程で、問題が発覚し、想像を絶する影響範囲を持った手戻りという形で跳ね返ってくる。そして、手戻りが発生すると無理やり仕様書やプログラム・コードを直すということが繰り返され、それが開発工数の増大、品質の歪みを累積する。結果的には、開発プロジェクトの納期の遅れ、コストのオーバーフロー、収束しないバグといった失敗に陥る。

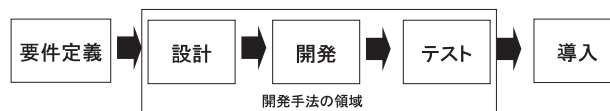


図2 ウォータフォール型開発プロセス

日経コンピュータ「2003年情報化実態調査」<sup>[4]</sup>でコスト超過の原因が記されているが、圧倒的に要求定義の不備による、追加の設計、開発作業が大きな割合を占めている。前述したウォータフォール型の開発プロセスの問題を解消するために、Rational社の開発プロセス・フレームワーク RUP (Rational Unified Process) に代表される「反復型」の開発プロセスがある。こ

れは小さなリリースを繰り返しながら全体的なシステムを構築していくというアプローチである。この小さなリリースのたびに統合が繰り返され何度も検証がされるので、最後になってから致命的な問題が発覚するということが防げる利点があると言われている。一方で、このプロセスの問題点は、全体を見極めないうちに作業が進んでしまい、プロジェクトの方向性を誤る危険性とプロジェクト管理の観点から最終成果物のスケジュール管理が難しく、システム開発の際には課題となる。また、基幹系業務システムの開発においては、業務領域が広く、かつ組織横断的な利害関係者も多い中、単に情報システムの構築には留まらず、要件定義工程の中で各部門間の業務プロセスの見直しも必要とすることから、この開発プロセスの適用は難しい。

システム開発の問題の多くは、新技術・アーキテクチャの採用による技術リスクや要員リスクといったリスクも伴うが、明らかにシステム開発の初期段階である要件定義工程の成果物に原因がある。ソフトウェア工学上もソフトウェアの問題の多くは、そのソフトウェアへの要求を集めて、文書化し、同意し、修正する方法に欠陥がある<sup>[3]</sup>、と言われている。

### 3.4 要件定義の難しさ

それでは何故、要件定義がうまくいかないのか？まず要求定義における要件定義の位置付けを確認する必要がある。図3に要求の関連を示す。要求の中には、まずビジネス要求として、経営のミッション、組織が達成したいと望むスコープとしての要求がある。次に業務要求として、ビジネス目的を達成するために、どのように業務を進めるかといった業務設計としての要求がある。そして最後に、機能要求として、実際にシステムで作り込む機能要求がある。これらの要求に加え、企業方針、法律規制、業界標準、計算式といった業務ルールがあり、対外的な外部企業、システムとの接続要件があり、そしてシステムに必要な信頼性や運用ルールなどの条件も加わる。これらの異なって混在している様々な要求レベルを整理し、かつ要求の領域には非公式な情報、暗黙の機能、誤った情報や伝えられていない情報が存在し、これらを聞き出し整理し、成果物としての要件定義書をまとめあげるプロセスに問題があると言える。

業務的な視点でみると、各企業独自の商習慣というものは、普段は当たり前でなかなか意識することもないが、いざシステム化を検討すると結構多いのが普通である。特に小売業においては、取引先、物流センターとの複雑な流通チャネルをはじめ、取引先の系列による取引の制

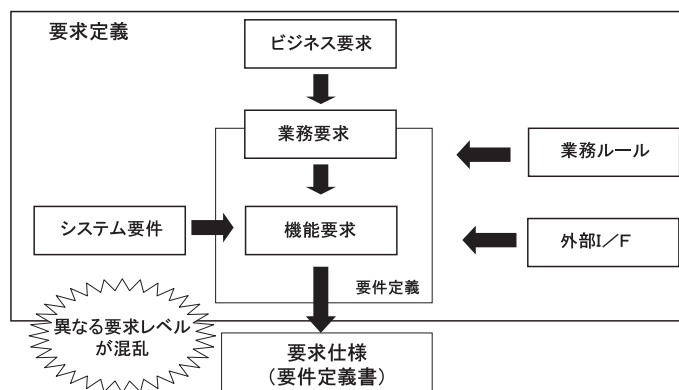


図3 要件定義の位置付け

約、多岐にわたるリポート等、企業固有の業務プロセスが多く存在する。企業の発注ひとつ取っても通常、企業で取り扱う商品分類は、生鮮(鮮魚・精肉・青果・惣菜)、日配、グロサリ、酒、住関、衣料、タバコ等様々であり、企業内でさまざまパターンが混在している。これらの様々なパターンとその業務プロセスを全て洗い出し、新システムでの対応方針を決定する必要がある。「その他」が残っていると本番運用開始後に実務が滞るといった問題が発生する。

一概に例外処理と言っても、広範囲な業務プロセスから多義的に業務ルールを定義すれば良い質のものと、長年の慣習によるルール違反、想定外の処理に大別され、その対応方針、方法を見極めることが重要であるが、非常に難易度が高い作業ある。

またソリューション・パッケージ適用の視点から見ると、ソリューションの実際の導入過程においては、現状機能で不足している機能を追加するというプラス面と、ソリューションの標準機能、もしくは代替機能により、結果的に既存システム比較し操作性が変わったり劣ったりするマイナス面が発生する。

当然のことながら新しい仕事の進め方、使用する画面・帳票の変更、新たな管理指標等に対する変化への抵抗は強い。導入メリットを最大限実現するためには、極力、ソリューションの標準機能を使用し、ソリューションに対する追加開発費用を最小化することが重要である。

#### 4. OpenCentral 導入ステップ概要

図5に標準のOpenCentral導入ステップを示している。本ソリューションの適用にあたっては、ソリューションのカスタマイズ・アドオン開発以外に店輔・対外接続系ならびに移行作業等、システム化の範囲が広範囲であり、要件定義後のプロジェクト計画の立案においては、これらを包含する必要がある。以下に各ステップの概要を述べる。

##### 4.1 プロジェクト発足

本ステップは新システム導入におけるプロジェクトの目的、プロジェクトの編成とプロジェクト導入の進め方、マイルストーン・スケジュールの決定である。プロジェクトの編成については、顧客側、当社のプロジェクト・マネージャの選定をはじめとしたプロジェクト体制と役割を明確にする。プロジェクトの目的については、特に顧客側プロジェクト・メンバの意識統一が出来ていることが必要である。

##### 4.2 要件定義 (Fit&Gap 分析)

本ステップは当社SEが中心となってOpenCentral適用のコンサルテーションを開始する。通常のシステム開発と異なり、ソリューションに通常業務のビジネスプロセスと業務機能は定義されており、顧客のビジネスプロセス/業務機能をソリューション標準のビジネスプロセス/業務機能と合うもの (Fit 機能)、合わないもの (Gap 機能) を明確にする Fit&Gap 分析が中心となる。

多くの Gap 機能が発生するが、それが標準機能だと何が損なわれるか整理するとともに、他業務プロセスとの関連と全体プロセスの最適化を視野に入れ解決案を検討、顧客への提案を行い、最終的に決定していく。その顧客特有の業務プロセス/業務機能は標準機能への追加開発となり、それらが少なければ少ないほど追加システム開発の費用、生産性は向上する。本ステップでは以下の要件を決定する。

- 1) Fit&Gap 分析によって決められた次期ビジネスプロセスのモデル化作成
- 2) OpenCentral 標準パラメタ設定の定義
- 3) Fit 機能の決定
- 4) Gap 機能に対するソリューションの解決方式と追加開発要件の決定
- 5) 店舗, 取引先接続等の対外接続方式と開発要件の決定
- 6) データ・業務移行の移行方式と開発要件の決定
- 7) 新アーキテクチャ, H/W,S/W 構成の決定
- 8) 以上の総見積りとスケジュールの確定

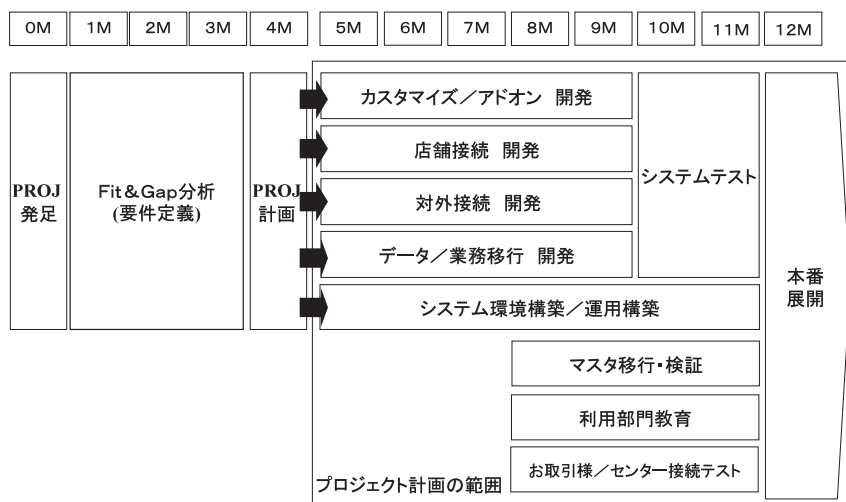


図4 OpenCentral 標準導入ステップ

#### 4.3 カスタマイズ/アドオン開発

カスタマイズには2種類のケースがある。一つは顧客のビジネスプロセスをソリューションが標準で持っているパラメータの設定とマスタの区分設定で対応するケースである。OpenCentralでは約70種のアプリケーション変数と約210種のマスタ区分をを用意している。これらの一部はFit&Gap分析の中でも、実際に実機で検証するものもあるが、最終的には本ステップの統合テスト工程で検証することとなる。

次のケースが顧客固有機能の追加開発および多くはないがOpenCentral標準機能への変更開発である。これらは本ステップにて「論理設計」～「統合テスト」を行う。

#### 4.4 データ移行・業務移行

本ステップでは、商品マスタ、取引先マスタと言ったマスタデータと、在庫データ、過去実績データ等のトランザクション・データの移行である。移行データには既存システムからシステム的に移行するものと、新規にデータを登録するものがあり、かつ既存システム側で本番稼働までに新システムに合わせメンテナンスしておくもの等、データの移行方式と移行データそのものが大量データになる為、本移行前に事前に移行出来るデータは移行する等の移行タイミングを決め、これら移行関連のプログラム開発とそのリハーサル・データ検証を行う。

業務移行では、システム化の対象範囲が広範であり、一斉に全ての業務モジュールの本番を

迎えるのは非常に顧客側のリスクが高い為、一定の期間内に一部の業務モジュールを先行して本稼働を実施していく。例としては、まず、数店舗での商品管理、特売管理の業務移行を実施し、その後2週間後に全店へ業務移行を実施し、さらにその後2週間後に発注・仕入管理、販売管理等の業務移行と段階的に本番を開始していく。その間、旧システムとの併行運用が必要となり、その間の両システムとのデータ連携方式とその連携プログラム開発を行う。

#### 4.5 店舗・対外接続

本ステップでは、本部ソリューションから店舗システムへの商品マスタデータ、特売データの配信や、店舗での精算実績データの集信等の連携といった店舗システムとのデータ集配信に関するプログラム開発を行う。同様に対外接続については、取引先や市場への発注データの送信や納品実績の受信、ならびに各種センターとへの出荷指示データの送信や出荷実績の受信といったプログラム開発を行う。

#### 4.6 システム環境/運用構築

本ステップでは、各種サーバ群の設置から開発環境、統合・システムテスト環境、本番環境、ならび保守環境の構築と新システムでの運用管理手順の定義を行う。

また、非常にシビアな処理時間を要求される発注締め処理や、大量データを扱う情報系 DWH へのデータ累積処理といったバッチ系の効率測定を中心としたベンチマーク・テストを実施する。

#### 4.7 システム・テスト

本ステップは、データ・業務移行の最終リハーサルによる移行データをもとに、本番開始可能なデータベースとシステム機能に対する顧客側の「検収テスト」の支援を行う。通常ここで使用するデータは、過去のある期間の本番データを利用し、テストデータの網羅性を合わせて検証する。

#### 4.8 本番展開

本ステップでは、本番稼働に向けたデータ・業務移行の最終移行と新システムの本番稼働であり、新システムへの顧客の定着を支援をする。同時に導入過程で発生した課題で、本稼働後に対応予定の課題解決と本稼働後に新たに見えてきた課題・要望に対する対応実施計画の策定を行う。

### 5. ソリューション導入方法論の必要性

ソリューション導入の要件定義工程では、前述した顧客業務の例外処理を含めた全ての業務プロセスの網羅性を確保し、顧客企業の新ビジネスプロセスを定義しなくてはならない。また、この工程の成果物の品質がプロジェクトの成功を確定付ける。

一方、これらの要件定義作業は非常に難易度が高く、当社でも小売業界を熟知し経験豊富なベテラン SE でないと実施が出来なかった。当然、そういった SE の数は少なく、かつ実務の経験が主であるから育成も難しい。

別の視点で、当社のソリューション・サービスの向上と均一化を考えた時、PM 要員の育成

に加え、ソリューション導入のコンサルティングが出来る SE の育成が必要となるが、ある程度経験を積んだ SE であれば、標準化されたナレッジ、導入方法論、そして導入ツールの使用により、ある程度均一なサービス・レベルの実現が可能になると考えてきた。

各導入プロジェクトで試行を続けている状況であるが、OpenCentral の導入に際し、要件定義工程を中心に導入方法論・ツールの整備をしてきた。現段階では大枠の方法論であり、厳密に定義されたフレームワークを解説したものはないが、現時点でこの大枠の方法論を使い五つのプロジェクトの要件定義を行った経験から、以下にその手法を述べる。

## 5.1 前 提 条 件

導入プロジェクトを成功に導く為に、顧客側に求める二つの前提となる条件がある。一つはプロジェクト導入の過程で、要件定義を行い、設計、開発、テストといった工程があるが、プロジェクトのスタートにあたり、顧客側の意識改革である。導入プロジェクトはソリューションの適用であるが、単に既存システムの置き換えではない。新システム導入には狙いがあり、導入企業の企業経営/企業戦略の視点から業務プロセスのあり方を見直す業務改革、BPR（ビジネス・プロセス・リエンジニアリング）のプロジェクトだと認識することが必要である。そして新しい価値観のもとで改革を進める為に、社内の各部門の意識改革が必要であり、プロジェクト・スタート時点で、何を目的として具体的にどんな方向で改革を進めるのを明確にすることが重要であり、顧客のトップ・マネジメントが関与することが必要である。そしてトップ・マネジメント自ら社員全員の意識改革を促す啓蒙活動を行うことが、より効果的である。

言い換えると全ての利害関係者が効果的に要求プロセスに従うことが重要要件であり、高品質の要求は効果的なコミュニケーションから生まれると言っても過言ではない。図 5 に顧客側に求める理想的な体制を示す。以下に各役割の概要を述べる。

### 1) プロジェクト・オーナー

プロジェクト推進の総責任者であり、全ての最終決定に関する権限を持つ。通常は役員がこの任にあたる。

### 2) プロジェクト管理者

プロジェクトの全体計画と体制を決定し、プロジェクト運営にあたり全体を管理していく実質的なリーダーである。プロジェクト・メンバの組織化を図り、メンバとの円滑なコミュニケーションを維持していくことが重要である。

### 3) ステアリング・コミッティ

プロジェクトの進捗・課題を認識し、承認・解決に向けた意思決定の最高機関である。通常はプロジェクト関連部門の役員、部門最高責任者がこの任にあたる。部門最高責任者としての立場から、他部門と協力をして全社的にプロジェクトを支援する立場である。ケースとしては、プロジェクト管理者と兼ねる場合もある。

### 4) 業務チーム

業務の専門的な立場から要件定義工程の中心的な役割を果たす。現状の業務プロセスを把握しており、ビジネス要求、業務要求、そして OpenCentral の標準機能から新しい業務プロセス、機能要求を決定していく。

通常は商品部の主要部門、経理部、店長といった現場部門の代表者から少数精鋭のメンバで構成する。特に現状の業務運用に固執せず、プロジェクトのスコープを意識し行動出

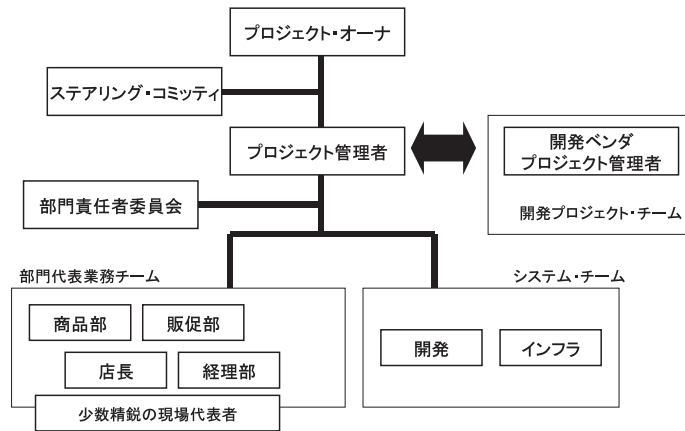


図5 理想的な顧客体制図

来る資質のメンバを選定する必要がある。

#### 5) システムチーム

現行アプリケーションの開発，保守要員，そしてネットワーク，HW,SW の保守要員の代表から構成され，新システム化要件の決定に際しい，業務チームの支援を行う。

### 5.2 当社のプロジェクトの体制と役割

以下に要件定義工程における当社のプロジェクト体制と役割を簡単に述べる。

#### 1) プロジェクト・マネージャ

当社標準のプロジェクト管理方法論である TeamMethod/PM, ISBP に準拠し，顧客の新システム導入プロジェクトが期待通りの品質で納期通り，かつ予算範囲内に本稼働させることを使命とする。その実現に向けて，顧客のプロジェクト・マネージャに対し，プロジェクト全般に対するコンサルティング，各種提案，報告，そしてプロジェクト総責任者に対する報告を行う。担当する要員はプロジェクト専任とする。

#### 2) ソリューション・スペシャリスト

ある程度の小売業務知識と OpenCentral の業務機能・データベースの知識をベースに，後述する導入方法論に沿って，Fit&Gap 分析の中心的な役割を果たす。基本的には1～2日/週の顧客との打合せの中で，Fit 機能の定義，Gap を中心とした課題解決を行う。Gap を中心とする課題に対しては，解決方式案の検討&策定を実施し，顧客への提案，支援等を行う。課題の中には難易度が高いものもあり，それらは後述するソリューション・アーキテクトとともに解決方式案の策定を行う。担当する要員はプロジェクト専任とする。

#### 3) ソリューション・アーキテクト

小売業務知識と小売業界へのシステム導入経験が豊富にあり，かつ OpenCentral の全機能/データベース構造を熟知しており，各導入プロジェクトの横断的な品質管理と支援を実施する。具体的には，各プロジェクトの中間成果物の検証を行う。特に Gap で発生した課題に対する解決方式の検証は重要であり，難易度が高い課題については，ソリューション・アーキテクト自らが解決案を策定し，顧客への提案を行うケースもある。

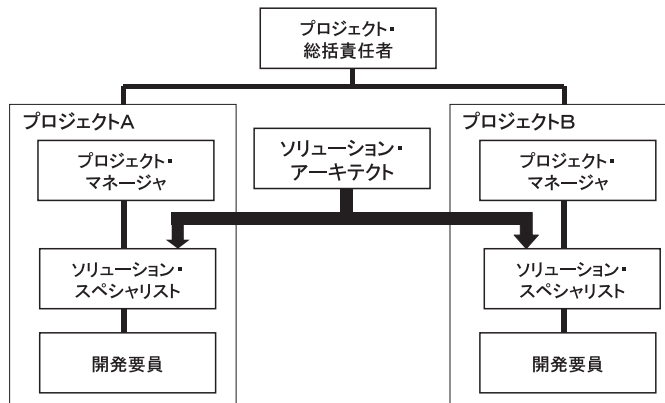


図6 当社の体制と役割

#### 4) プロジェクト総括責任者

プロジェクト・マネージャに対するプロジェクト・マネージメント・レビュー、品質レビューと多面的なアドバイスおよび支援等を行う。

### 5.3 OpenCentral の導入方法論

本章において OpenCentral の導入方法論を述べる。本方法論で述べる工程は要件定義工程であり、Fit&Gap 分析を中心に行い、最終的な目標は、実行可能なビジネスプロセスを構築することである。

#### 5.3.1 導入ツール資源

本方法論には7個のツール資源があり、これらのツール資源を使用してプロトタイピングを行い、結果としてソリューションにビジネスプロセスを統合的に取り入れていく。ツール資源は、概念ビジネスレベルとシステム・レベルに分けられ、その概要を以下に記述する。

##### 1) 概念ビジネスレベル

###### ① 標準ビジネス・プロセス・フロー

OpenCentral Navigator と称し、OpenCentral の標準ビジネスプロセスの流れを各業務プロセス単位に表現する。これらがソリューションの導入において実際に実現可能なビジネスプロセスを表現しており、導入顧客にとって新システムの適用範囲を決定するとともに、新ビジネスプロセスを決定するモデルとなる。顧客プロジェクト・メンバに対し、ソリューションのビジネス・モデルと、かつその流れの中でユーザが使用する各画面、帳票を視覚的に理解させることが可能である。

###### ② ビジネス・ルール定義書

OpenCentral の各業務プロセスのビジネス・ルールを詳細に定義しており、OpenCentral が標準で保持しているビジネス要件の理解が可能である。

###### ③ ヒヤリング・ワークシート

各業務プロセスの確認ポイントを網羅的に記述しており、特にこの中で記述している顧客業務内での例外処理の確認ポイントが、若干、経験の浅いソリューション・スペシャリストに対する知識ベースにもなる。

## ④ データモデル

OpenCentral のベースとなるデータベースの ER (EntityRelationship) モデル図と標準データ辞書から構成される。通常、初期段階では使用せず、新ビジネスプロセスが固まった段階で、各業務プロセスの管理項目の検証と確定に使用する。

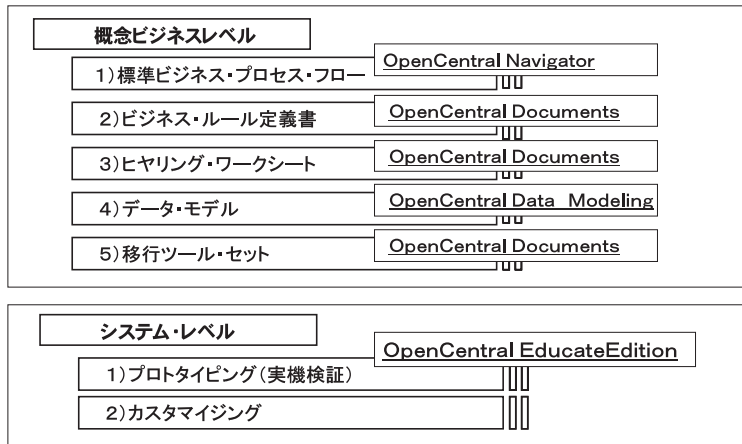


図7 OpenCentral 標準資源ツール

## ⑤ 移行ツール・セット

データ・業務移行に関する標準のツール・セットであり、要件定義工程の最終段階で行う移行関連の要件定義時の使用する。OpenCentral が標準で装備する移行プログラムと、顧客既存システムとのデータ・マッピング用の標準シートから構成される。

## 2) システム・レベル

## ① プロトタイピング

プロトタイピングは要件定義工程の中の実機検証であり、一般的には CRP (ConferenceRoomPilot) とも呼ばれる。ここでは実機検証用のデータベース環境ともに導入後、即稼働可能な環境を提供する。また、要件定義の初期段階ではユーザーもソリューションに対する習熟度は低い為、実機検証用のサンプル・シナリオとそれをナビゲートする手順書を用意し、それに沿って徐々に習熟度を上げていくことが可能となる。これらのセットを OpenCentral では OpenCentral Educate Edition と称している。

## ② カスタマイジング

カスタマイジングは顧客の新ビジネスプロセスに対する目的や要求に従い、OpenCentral が標準で保持している様々なパラメタ、ならびに各種マスタで保持しているビジネス・ロジックに影響する項目区分の設定内容を、前述したプロトタイピングでの実機検証とともに決定を行う。

## 5.3.2 導入方法論の実行手順

図8に Fit&Gap 分析の概念図を示す。まず各業務プロセス単位に机上での概念レベルの検証を実施し、次に概念レベルの仮決定事項をシステム・レベルの実機による検証を行ない、再度、概念レベルの検証を繰り返す。そして次の業務プロセスに進んでいくが、必要に応じて前

の業務プロセスに戻り、再度、検証を繰り返す。その過程で、ソリューションの標準ビジネスプロセスと顧客が要望するビジネスプロセスとの間に Fit 機能と Gap 機能が発生し、各々のタスク検証を繰り返し、ソリューションで実現可能なビジネスプロセスを決定していく。ここで重要なことは、既にソリューションそのものがデータベースを中核に各業務機能の整合性が保たれていることから、ビジネスプロセスの検証に際しては、各ビジネスプロセス間のデータ整合性維持の観点からデータモデリングについて検証を行っていることである。

実行の形態は、概念レベルの検証が顧客との打合せをベースとして1～2日/週を実施し、その中で実機検証のカスタマイジングについても実施する。プロトタイピングの初期段階は若干の支援を要するが、顧客主体の作業として実施が可能である。

また、通常のパッケージ適用においては、当然、パッケージ機能の理解が重要であるから、まず、パッケージ教育を実施し、その後、要件定義となるが、ソリューションの適用においては、業務範囲が非常に広範囲でもあり、より効果的な理解を求めるために、各プロセスの中で融合的に実施している。

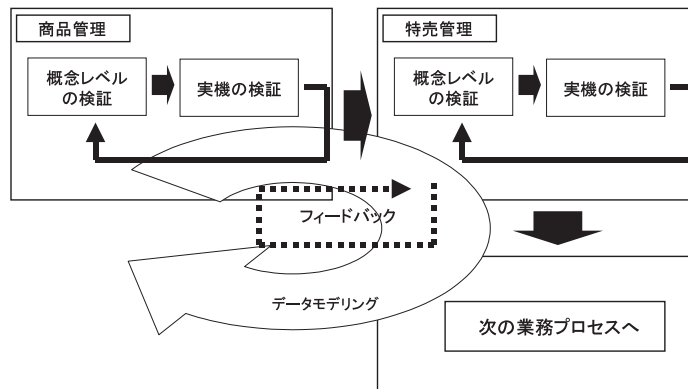


図8 Fit&Gap 分析の概念図

図9にFit&Gap分析のタスクフローを示している。以下に各タスクの概略を記述する。

#### 1) 新ビジネスプロセスの検討

本タスクは、新ビジネスプロセスの検討を行う。手順としては、初期段階のプロジェクト・メンバはソリューションの知識が乏しいため、OpenCentralNavigatorによる標準ビジネスプロセスと各業務機能、およびビジネス・ルール定義書による各業務プロセスのビジネス要件の説明を中心に行い、プロジェクト・メンバのソリューションに対する習熟度を上げる。次のステップでは、顧客によって記入されたヒヤリング・ワークシートを元に、顧客要望のビジネス要件に対するソリューションとのマッピングを行う。そして、このマッピングの結果、ソリューションに対するFitプロセスとGAPプロセスの候補が出される。

#### 2) GAP抽出

本タスクでは、候補として出てきたGAPプロセスに対する評価を実施する。要求の妥当性を判断する基準として、要求元と有効性の二つの基準から評価を行う。要求元の判断基準として優先順位の高いものからみる以下のようなになる。

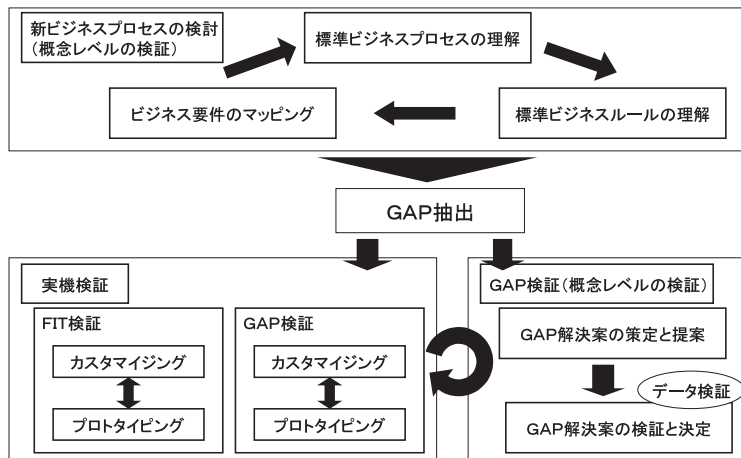


図9 Fit&amp;Gap 分析のプロセスフロー

- ① 法律、条令で要請されているもの
- ② 商習慣上で必要とされるもの
- ③ 顧客（新システム導入先からみた顧客）から要求されるもの
- ④ 顧客企業の方針からくるもの
- ⑤ 部門単位の要望からくるもの
- ⑥ 個人の要望からくるもの

また、有効性の高いものから見ると、以下のようになる。

- ① 企業存続に必須の機能
- ② 競争優位上、省けない機能（差別化要因）
- ③ 全部門にまたがり効果が期待できるもの
- ④ 部門の効果が期待できるもの
- ⑤ あれば便利、必要性が不明確、使用頻度が低い機能

以上の評価基準から GAP プロセスに対する解決応方針を決定する。

通常、解決方針には、①業務プロセスをソリューションに合わせる（現行の仕事の仕方を変える）、②現行の業務プロセスを廃止する、③アドオンとして追加開発する、の3方式があり、①、②については、Fit プロセスとして次プロセスに進む。③のアドオンに関しては、追加開発のための追加コストと開発期間、他業務プロセスへの影響度を充分考慮し、決定を行う。

### 3) 実機検証（Fit 検証）

Fit 候補案件に対する実機検証を行う。手順としては、Fit 候補案件に対して、ソリューション標準のパラメタ設定内容、ならびに各種マスタで管理されている区分の設定内容を決定し、OpenCentralEducateEdition のサンプル・シナリオと手順設定内容に対する結果検証を含め、Fit プロセスに対する検証を行う。また、このタスクの実行の終了後、顧客のソリューション習熟度は、ほぼ完成される。

### 4) GAP 検証

GAP プロセスの内容は千差万別であるが、プロジェクトのスタート当初はプロジェク

ト・メンバのソリューションに対する習熟度の低さから来る不安により、操作面に対する GAP も少なくない。その為、その様な GAP 案件も、実機検証により検証を行う。手順については、Fit 検証の実機検証と同様である。

次に GAP 解決案の策定と提案については、GAP 評価でのアドオンとして追加開発する GAP プロセスが対象であり、解決案の策定にあたっては、当社のソリューション・アーキテクトとの代替解決策の有無等、検討を実施し、解決案の顧客提案を行う。

GAP 解決案の検証と決定では、提案した解決案に対し、顧客との協議の上、正式に追加案件と決定する。

#### 5) ビジネスプロセスの決定

決定された Fit 業務プロセスと GAP 業務プロセスに対する解決方式から、ビジネス要件とシステム要件を決定し、新ビジネスプロセスを決定する。以上のタスク・フローは、基本的には業務プロセス単位に進めており、この段階で最終的に他業務プロセスとの関連を検証し、決定する必要がある。

#### 6) 開発見積確定

全開発案件を業務プロセス毎に、画面、帳票、バッチ別に機能分解表に展開し、見積りを実施する。その後、本番稼働スケジュール案とともに顧客提案を実施し、顧客プロジェクト・オーナーの承認後、次工程へ進む。

### 5.3.3 Fit&Gap 分析のマネジメントのポイント

GAP 解決策でソリューションに対する追加・変更は、開発工数を増やし、開発納期、開発コストへの影響を与える。最終的な決定に際してはビジネス目的である品質目標を達成をさせること、プロジェクト管理から最終的な納期、コスト目標を厳守すること、のどちらをどれくらいのバランスで目指すのかを判断基準とする必要がある。

まず品質目標の達成のためには、

- 1) 解決案に対し、収益向上、コスト削減といった財務的な効果は？
- 2) 企業ブランドの向上といった非財務的な効果は？
- 3) 法改正、センター等の外的要因
- 4) 実現難易度に成功の可能性は？

といった判断ポイントがあり、納期、コストの達成のためには、

- 1) 実施による納期の影響は？
- 2) 実施によるコストへの影響は？
- 3) ハードウェア、ソフトウェア等のリソースへの影響は？

といった判断ポイントがある。

実際の導入にあたっては、これら両方の目標を達成することが難しいケースが発生する。そして、多くのケースで、最終判断はその企業のマネジメント方針に委ねられるケースが多い。しかし、プロジェクト運営としては、これらのバランスをプロジェクトで明確に定義し、決定事項についてプロジェクト・メンバ全員が共有していくということが重要である。

### 5.3.4 Fit&Gap 分析の日程プラン

Fit&Gap 分析は、当社と顧客との打合せが主体であり、この打合せ密度の濃さが本工程の

品質と作業進捗に左右する。この打合せを行う顧客メンバには現場の代表者も参画しており、現行業務との併行作業になる。従って、本工程の計画段階で、いついつの打合せで何を議論すると言った打合せプランの策定が必要になる。

現在、本方法論の標準打合せ回数は、導入企業のシステム機能範囲にも若干影響はするが、図 10 に示すとおり 24 日間で終わらせることが可能である。各サブシステムでの標準日数とそ  
の中で議論する内容を 1 日単位で定義しており、これに従い進めていけば、進捗の把握も可能となるメリットもある。

また、この工程の標準期間は 3 か月としている。よく同計画の提案時に、もう短縮出来ないか、と言われることがある。しかし、顧客にとって当社との打合せ以外に、顧客はプロトタイピング、Gap を始めとした課題対応方針に対する社内調整等のアクションアイテムを実施する必要があり、当社は Gap 解決方式案の策定ならびに、中間成果物作成&検証といったアクションアイテムを実施する必要がある。また、顧客メンバの時間の制約もあり、打合せに取れる時間は 1~2/日と想定するべきであり、仮に 2 日とすると 12 週間を要することになり、この期間が必要であり、妥当であると考える。

そして要件定義工程で重要なことは、計画した期間で、顧客企業の基幹業務の全てを掌握し、そのビジネスプロセスを再定義することであり、かつ以降の工程である設計から本稼働までのプロジェクト運営において、要件定義後に策定されたプロジェクト計画との乖離がないことが重要である。

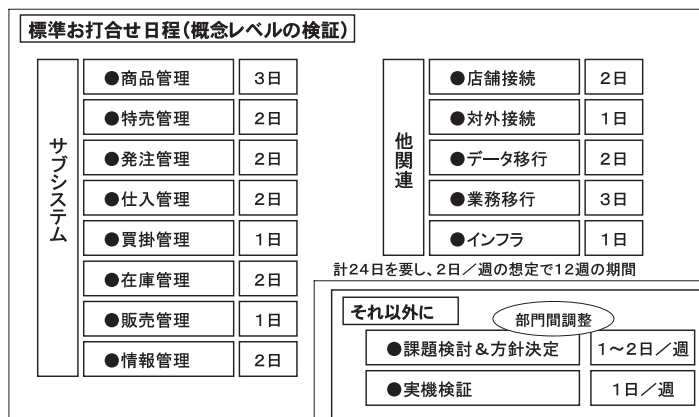


図 10 Fit&Gap 分析標準日程プラン

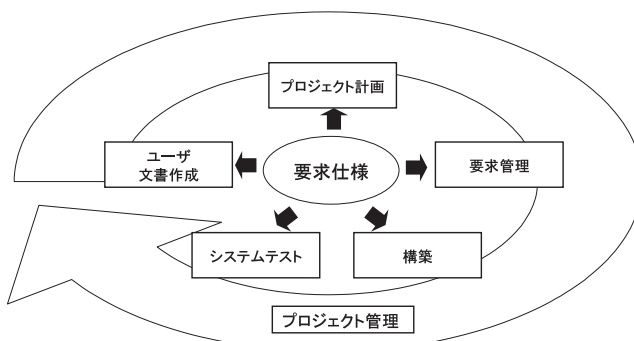
## 6. お わ り に

プロジェクト管理の重要性は言うまでもないが、実際の開発においては、要求仕様がプロジェクト計画のベースラインとなり、そして設計工程以降の変更管理、構築作業、システムテストといった全ての工程のベースラインとなる。そういう意味から、プロジェクト管理とソフトウェア開発の両アクティビィの基盤となり、要求仕様の精度そのものが明らかに、プロジェクト成功の可否を決定付けることは言うまでもない。

また多くの人が、要求の議論に時間を費やせば、かかった時間だけ単純に納期が遅れるという誤解を持っているが、要件定義に対して、工数、スケジュールといったリソースのより多くの部分を充てたプロジェクトの方が、実質的に費用も少なく、スケジュール超過も少なかった、

という実績も昔から多くの文献で言われている。

筆者は今後も多くの企業に当社の OpenCentral が導入され、各企業がシステム刷新の目的を確実に達成することを期待すると同時に、この方法論を継続的にブラッシュアップさせ、より洗練された手法として進化させて行きたい。



ソフトウェア開発とプロジェクト管理の両アクティビティの基盤

図 11 プロジェクト管理と要求仕様の関係

- \*1 MD サイクル：Merchandising 一企業（小売業者）が自社の行う私的流通の全体を設計、計画、コントロールすること。言い換えると適正な商品を適正な場所（店舗・売場）で、適正な時期に、適正な数量を、適正な価格で提供すること

- 参考文献** [1] 伊藤 重光, 「ERP プロジェクト こうすれば成功する」, 日本経済新聞社, 2003  
 [2] 佐野 勝久, 「日本ユニシスの ERP 導入標準アプローチ」, 日本ユニシス, 技報第 69 号, 2001.  
 [3] Karl E. Weigers SOFTWARE REQUIREMENTS [邦訳：渡部洋子監訳, ソフトウェア要求－顧客が望むシステムとは, 日経 BP ソフトプレス, 2003 年]  
 [4] 日経コンピュータ, 2004.6.14 号

**執筆者紹介** 小口 修一郎 (Shuichiro Oguchi)

1983 年工学院大学化学工学科卒業。同年日本ユニシス(株)入社。製造, 流通分野でアプリケーション開発, SE サービスに従事。1999 年より小売業向けソリューション開発とその適用に従事。現在, 日本ユニシス・ソリューション(株)製造流通サービス本部製造流通第一統括部システム 3 部部長。