

統合パッケージソフトによるラピッドプロトタイピングシステム開発

片岡 信弘 (三菱電機)
伊藤 峰秋 (三菱電機)
岡 哲生 (三菱電機)
小泉 寿男 (三菱電機)

本論文で、我々は、統合パッケージを利用した新しい情報システム開発方式と、この時の開発工数モデルについて提案を行う。この開発方式の特徴は、パッケージを武器としたRAD (Rapid Application Development) とJAD (Joint Application Design)の活用である。パッケージ機能以外の追加開発を行わないという理想条件での予想開発モデル工数は、従来のウォーターフォール型での開発工数を100とした時、12.5となり、かなりの工数削減が図れることを提案した。この方式の有効性を実証するため、具体的なシステムへの適用を行った。本方式を採用した予測モデル工数が37人月に対して、適用事例の結果は、34人月となりモデルの有効性が検証できた。

New methodology for developing information management system which use application package software

Kataoka Nonuhiro (Mitsubishi Electric Corporation)
Itoh Mineaki (Mitsubishi Electric Corporation)
Oka Tetuo (Mitsubishi Electric Corporation)
Koizumi Hisao (Mitsubishi Electric Corporation)

This paper propose a new methodology for developing information management system.

This methodology use package software for a weapon of RAD (Rapid Application Development) and JAD (Joint Application Design). And also propose a new model for man power at the condition that we don't add functions beside package. Man power of the model is 12.5 when the man power of water fall model is 100. We apply this new methodology to a actual information management system. The case model manpower is 37 man month, and real man power is 34 man month.

1. はじめに

現在多くの企業で情報システムの革新の動きがある。これは、従来のシステムが次のような問題点を抱えているからである。

(1) ダイナミックに変化する経営環境に追従できない。

図1に従来の典型的なシステムの概念図を示す。独立したシステムのバッチ処理処理によ

る繋ぎ合わせであり、システムにまたがったリアルタイムの情報処理が困難であった。またデータベースが統合されていないため情報システムの構築に手間がかかっていた。

(2) コストパフォーマンスがビジネスの要求に合わない。

システム開発コスト、プラットフォームコスト、運用コスト等すべてにおいて、利用者の満足が得られていない。

(3) ウォータフォール型情報システム開発方式の問題点。

大量の手作りを前提としたカイクラムメイドのシステム開発であり、開発期間が極めて長かった。また利用者の理解しにくいドキュメントによるシステム仕様の記述のため、利用者を疎外した開発となっていた(1)。

以上のような問題点に対して要求される情報システムの在り方と開発方式は、ビジネスプロセスをシームレスにカバーする統合されたシステムであることと、ビジネスの変化に柔軟に対応するために迅速な開発ができることである。

このために最近注目されているのが、統合パッケージによる情報システムの開発である。統合パッケージ利用時の開発作業内容は次のようなものである。

(1) パッケージのカスタマイズ

- ・パラメータ設定による機能の選択（業務フローの選択を含む）
- ・組織等の情報システム取り巻く環境パラメータの設定
- ・各種マスターデータの設定

(2) 追加する不足機能の作り込み

一般にパッケージソフトにおけるカスタマイズは、パラメータの設定による上記のような作業であり、ソースの変更はモディフィケーションと呼ばれ原則としておこなわない。

これが注目されるようになってきたのは、以下の理由による。

(1) 企業に於ける様々な業務（営業、資材、経理、生産管理等）を統合し、これをシームレスにこれをカバーするような統合パッケージが出現してきたこと。

(2) 従来、中小企業では、当たり前であったパッケージソフト利用の概念が大企業にも浸透しだしたこと。

しかしながら、パッケージソフトによる開発は、いかにパッケージ機能の範囲で利用者を満足させるかの課題がある。

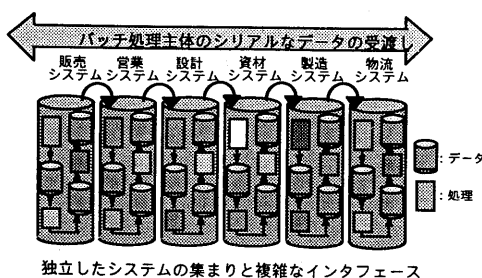


図1 従来の典型的な情報システム

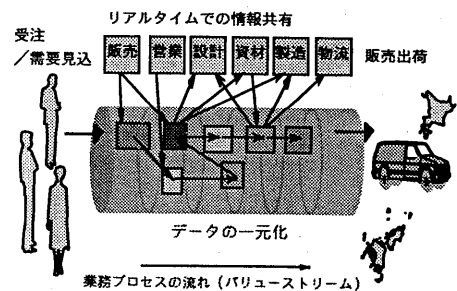


図2 新しい基幹情報システムにおける統合化

我々は、統合パッケージを利用した新しい情報システム開発方式の提案をおこなう。これはラビットプロトタイピングのツールとしてアプリケーションパッケージを用いる方式である。これによりRAD (Rapid Application Development)、JAD (Joint Application Design)を実現し、少ない工数で短期間にシステムを開発する方式である。

2章で新しい開発方式とこれを適用した時の開発工数モデルについて論じ、3章でこの方式の適用事例工数と予測モデル工数との差について論じる。4章で今回のモデルの有効性について述べる。

2. 新しい情報システムの開発方式の提案

図2に統合されたデータベースのもとで、各システムがリアルタイムに情報を共有している新しい情報システムの概念図を示す。このシステムでは、例えば営業の受注情報が設計部門に流され、これにより、設計部門が製造手配、資材購入手配を行うといった一連の業務プロセスが一元化されたデータのもとで処理されていく。

また、図3はこのようなシステムを、統合パッケージをツールとして利用して開発するRAD、JAD方式を示している。RADとはラビットプロトタイピングをくり返ししながら最終システムに仕上げていく手法であり、JADとはプロトタイピングをしたものを利用者に評価させることにより当初から利用者を巻き込みながらシステム開発をしていく手法である。

ユーザの要求に従い設計を行い、次にこれに基づきパッケージのカスタマイズを行い、これによりユーザの評価(JAD)を行う。このステップを繰り返すことにより、最終システムに仕上げていく。このステップにおける各作業内容を次に説明する。

設計：(1) パッケージのどのような機能を使うか、あるいは同一の機能が異なったサブシステムに存在する時にどちらの機能を使用するか等の選択。(2) パッケージ上のデータベースの項目と、実際に利用するデータとの対応づけ。

カスタマイズ：(1) 詳細パラメータの設定(詳細機能設定)(2) 動作確認のためのデータの設定。(3) これに基づく動作確認。

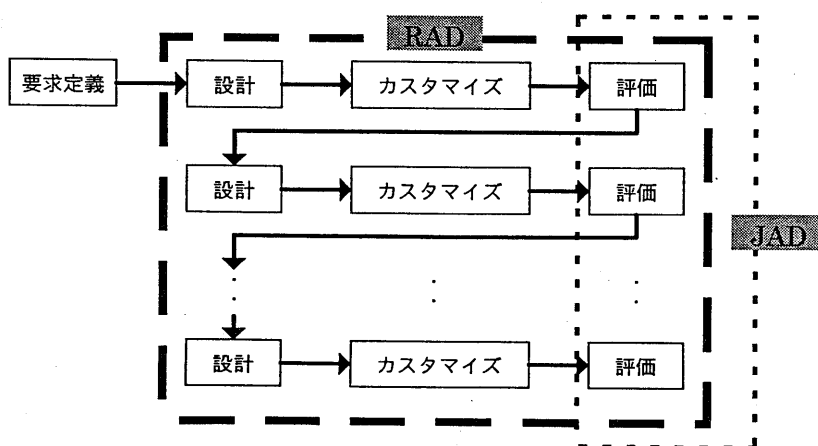


図3 RAD、JAD方式

評価：カスタマイズしたパッケージを実際に動作させることによるユーザによる機能の確認、使いがっての確認。

また、図2に示した統合された業務システムの全体についてこのようなRAD、JADを適用できることが大きな特徴であり、計画段階、要求機能定義段階で威力を発揮する。RAD、JAD方式に基づく開発ステージは図4に示す通りである。各ステージでの作業内容を以下順に示す。

ステージ1（計画）：

システム開発の計画作成を行うステージである。従来の方式との大きな違いは、新しく開発されるシステムの全体の概念をパッケージの標準プロトタイプデモにより、利用者に理解させることである。標準プロトタイプとは、細かな機能の選択は行わず、今回のシステムにヒットしたデータを用意してデモを行うものである。これにより、システム全体に対するコンセンサスを関係者全員で得ることが目的である。特にシステム間でのシームレスな情報の受渡し等、新システムのメリットを関係者に理解させることに有効である。

ステージ2（要求定義）：

このステージでは、パッケージの機能と要求機能のギャップ分析を行い、パッケージの適用範囲、追加開発の概要仕様の決定を行う。このステージに於いて重要なのは、細かなギャップ分析を行うのではなく、現状のパッケージ機能にいかに関業を合わせるかとの観点より検討である。このために、業務の要求に合わせてパッケージの大まかなカスタマイズ（機能選択）を行い、これのデモにより、パッケージに業務を合わせても業務がなりたつことを納得させることにパッケージのプロトタイピングを有効活用する。

ステージ3（設計）：

このステージは、詳細機能のヒアリングによる設計、パッケージの詳細カスタマイズ、これを動作させることによるユーザの評価を行う。これによりさらにヒアリングを詰めていくというサイクルをまわして行く。またこれ共に、追加機能の設計を行う。

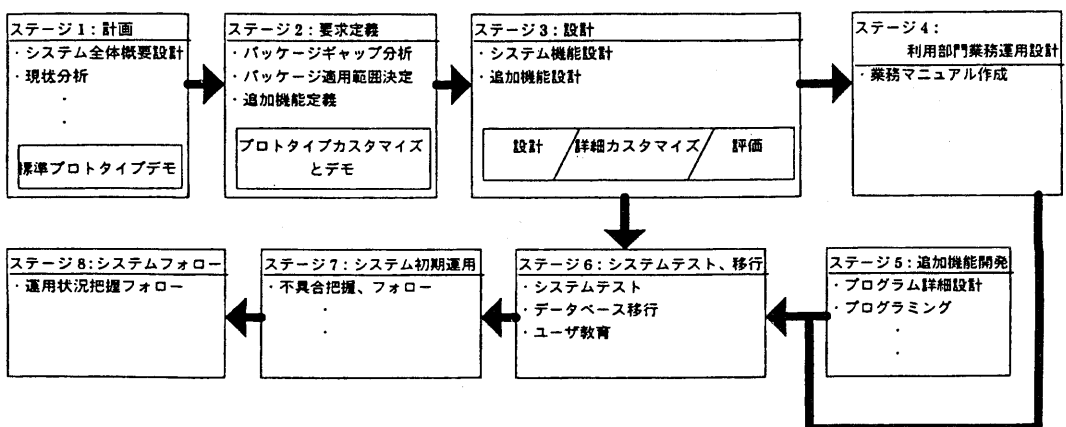


図4 新しい情報システム開発方式

ステージ4（利用部門業務運用設計）：

利用部門での業務遂行のために必要なマニュアルの作成を行う。ここにおいても、プロトタイプリングされたシステムを動作させながら、マニュアルの作成を効率よく行う。

ステージ5（追加機能開発）：

追加プログラム、追加／変更画面、帳票のプログラム設計プログラミング、テストを行う。

ステージ6（システムテスト、移行）：

システムテストの計画、実施、評価であり従来の作業と基本的には同じである。しかし、この段階においても、カスタマイズの一部変更によるシステム機能の変更が可能である点が従来の方式と大きく異なる。またデータの移行、ユーザ訓練等を行う。

ステージ7（システム初期運用）：

ユーザの運用の支援、初期不具合の把握、改善である

ステージ8（システムフォロー）：

運用状況の調査、把握フォローである。

次にこの開発方式を従来のウォーターフォール型モデルと比較したものを表1に示す。

開発の作業内容はステージ1，2，3はフェーズ1，2，3に、またステージ6，7，8はフェーズ7，8，9にそれぞれ対応する。

ステージ4，5はフェーズ4，5，6に対応するが、利用者運用マニュアルの作成を除いては、ステージ4，5が追加開発のみに対するものであるのに対してフェーズ4，5，6はシステム全体であることが大きく異なる。

表1に示す従来のウォーターフォール型モデルにおける各フェーズの工数比率（当社の長年のデータより）をもとに、新しい開発方式での作業効率の向上を組込んだ予想作業工数モデルを表2に示す。

表1 パッケージ利用モデルと従来のウォーターフォールモデル型モデルの対比

パッケージ利用型モデル		ウォーターフォール型モデルでの対応			
				*工数 比率(%)	作業目的対応度 (対レベル利用モデル)
ステージ1	計画	フェーズ1	計画	3	同じ
ステージ2	要求定義	フェーズ2	システム設計	6	同じ
ステージ3	設計	フェーズ3	サブシステム設計	15	同じ
ステージ4	利用部門業務運用設計	フェーズ4	利用部門業務運用設計 EDP 処理設計	19	・運用マニュアル作成は 同じ
ステージ5	追加機能	フェーズ5	プログラム作成	25	・システム全体のプログラ ム作成となる。
		フェーズ6	EDP システムテスト	15	
ステージ6	システムテスト、移行	フェーズ7	システムテスト、移行	10	同じ
ステージ7	システム初期運用	フェーズ8	システム初期運用	6	同じ
ステージ8	システムフォロー	フェーズ9	システムフォロー	1	同じ

*当社の長年のデータにもとづく

表 2 パッケージ利用モデル工数モデル

	ステージ1	ステージ2	ステージ3	ステージ4	ステージ5	ステージ6	ステージ7	ステージ8	合計
ウォーターフォール型での工数	3	6	15	59	10	6	1	100	
新しい方式での効率向上度	1	4	4	—	4	4	4	—	
新モデルでの工数	3	1.5	3.8	0	2.5	1.5	0.25	12.5	

ウォーターフォール型開発モデルを 100 とした時の新モデルでの工数

開発の工数は追加機能の量によりシステムごとに大きく異なるため、このモデルでは、追加開発がゼロである理想的なケースを想定している。このモデルの算出根拠を以下に示す。

ステージ 1：計画策定作業であり効率は従来方式と同じと考える。

ステージ 2，3：機能定義、設計作業の R A D，J A D による作業効率向上を次の理由により 4 倍と考える。一般に技術者の職場における作業要素時間の調査によると、資料作成が 24 %、チェックが 7 %、検討、考察が 19 %、会議が 11 % を占めており、パッケージを利用することにより設計資料作成、検討、考察がほぼゼロとなれば上記の合計時間は、61 % が 18 % となる(2)。さらにプロトタイプング見ながらの会議であることの効率向上を加味し効率向上を 4 倍とする。

ステージ 6，7，8：パッケージソフトの利用であり、システム品質が高いためのシステムテストの作業効率向上を 4 倍と考える。一般にシステムテストにより、残存不具合を 1 / 110 程度まで減少させており、当初よりこの程度の品質であれば、作業工数は 1 / 4 と想定した。

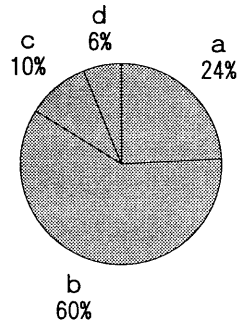
この結果、表 2 では、ウォーターフォール型での開発工数を 100 とした時、パッケージ利用モデルでの予測工数は 12.5 となり、かなりの工数削減が図れることになる。

次に表 3 に開発作業分類での工数比率比較を示す。過開発作業の内容を以下の 4 つに大分類した。

表 3 開発作業分類での工数比率比較

	分類内容	ウォーターフォールモデル		パッケージ利用モデル	
			工数比		工数比
a	計画、システム設計 (システム、サブシステム)	フェーズ 1	24%	ステージ 1	69%
		フェーズ 2		ステージ 2	
		フェーズ 3		ステージ 3	
b	プログラム開発 (設計、作成、テスト)	フェーズ 4	59%	ステージ 4	0
		フェーズ 5		ステージ 5	
		フェーズ 6			
c	システムテスト	フェーズ 7	10%	ステージ 6	20%
d	システム初期運用	フェーズ 8	6%	ステージ 7	12%

ウォーターフォール型モデル



パッケージ利用型モデル

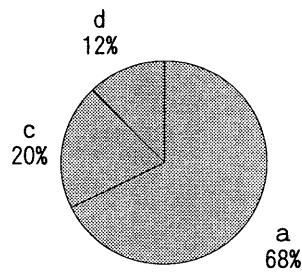


図5 開発作業分類での工数比率円グラフ

- a. 計画、システム設計（システム、サブシステム）
- b. プログラム開発（設計、作成、テスト）
- c. システムテスト
- d. システム初期運用

これらの分類ごとの各フェーズの対応と、ステージの対応、および工数比率を記入した。来
のウォーターフォール型のモデルで大きな比率を占めていた、bのプログラム開発作業が大幅
縮小することにより、計画、システム設計作業が極めて大きな比率を占めることとなる。
したがって従来大きく外注作業にたよっていた情報システム部門の開発体制も大きく変更を
要求されることとなる。

3. 新しい開発方式の適用評価と考察

3. 1 新しい開発方式の適用

この方式の有効性を実証するため、典型的な具体的システムへの適用をおこなった。この
システムは、経理、資材の全システムおよび生産管理システムの一部である。システムの規
模は従来のCOBOL換算で数百Kステップであり、従来のウォーターフォール型での開発工
数は約300人月と想定される。これに基づき、予測モデル工数と今回の適用結果工数を比
較したものが表4である。この表では、ステージ1（計画）の工数が予想モデル工数と適用
事例工数で大きな差が存在するものの、それ以外のステージでは、かなり近い値を示してい
る。各々のフェーズでの予想モデル工数と適用事例工数の差は、次の理由と考えられる。
ステージ1：予想モデル工数よりもかなり小さくなっている。今回の事例では、計画時点で、
詳細な計画立案を行わず、とにかくパッケージソフトを適用するとの前提でプロジェクトを
開始した為と考えられる。このステージについては、今後のデータを待って評価したい。
ステージ2，3：予想モデル工数よりも、多少大きくなっている。この工数増加は、今回の
プロジェクトの主要員4名の内1名を除いては、当統合パッケージの未経験者であり、これ
が工数増加要因と考えられる。

ステージ6：予想モデル工数とほぼ同じであることは、システムテスト時の手戻り作業が従
来は大きなウェイトを占めていたことを示している。

表4 パッケージ利用工数モデルの適用例

	ステージ1	ステージ2	ステージ3	ステージ4	ステージ5	ステージ6	ステージ7	ステージ8	合計
モデル工数	9	4.5	11.4	0		7.5	4.5	0.75	36.9
適用事例工数	2	6	16	0		8	2	未	34.0

ステージ7：予想モデル工数よりさらに小さくなっている。このステージでのシステム品質による工数低減率は、ステージ6以上に大きく寄与する為と考えられる。すなわちシステム品質が高いとしても、システムテストとして一通りのデータを流す必要があるが、ステージ7では、不具合が発生しなければ、作業内容は、純粹にシステム運用のための支援活動のみとなるからである。

3. 2 考察

提案した新しい情報システムの開発方式においては、パッケージ利用開発モデル工数で示すように、開発工数の低減に有効であることの可能性が実証された。パッケージソフトを利用したシステム開発では、従来の手作りのシステムと比較して、大部分をパッケージの機能の利用でおこない、一部の不足機能を作り込むのみのため、大幅にプログラム作成作業（ステージ5）の工数が減少することは当然である。しかしこれ以外の設計、システムテストの工数もかなり小さくなることが実証された。これは、仕様の検討工数と共に、これをドキュメントとして纏める設計書作成工数が大きく減るためと考えられる。また実際に動作するシステムを見ながら仕様の確定作業を行うために、仕様決定の効率が良いためと考えられる。

4. まとめ

本論文では、パッケージソフトを武器としたJAD、RADによる新しい情報システム開発方式、これに基づき、追加開発がゼロの時のモデル工数について論じた。パッケージソフトを利用することは、単にプログラム作成の工数削減だけでなく、仕様の確定の作業も大きく効率向上させることができることが明らかとなった。今回は、開発工数についてのみモデル化をおこなったが、開発期間に対するモデルの課題が残っており今後取り組みたい。またJADで利用者の参加による意志決定のメカニズムがどのようなものであり、意思決定の公立向上にどのような要素が大きく寄与するか解明も行いたい。

文献

-
- (1) 「システム生産標準"SPIRINGAM"」三菱電機技報、67, No10(平3)
 - (2) 増井・田中・三好：「グループウェア機能を備えたエンジニアリングオフィスシステム」電学誌, 113, No12(平5)
 - (3) 高木：「グループ意思決定会議のコミュニケーション構造」電学論C, 114, No3(平6)