

ソフトウェア開発工程の在り方

河野善彌*, 陳慧†, Hassan Abolhassani ‡

この報告は、システム開発の要求分析/獲得と称される開発の初動期に焦点を当てる。上流では「目的の階層性」、以後は「機能の階層風開連鎖」を使うことが良い。企業活動は・入力から出力のデータフロー、・その制御構造、の両者の対の階層構造で表される。システムはデータ/ビジネスの流れと制御による階層的な論理モデルで表せ、プログラム化はこのモデルにより容易に行える。既に同思想/方針により成功された例がある。その発表を要約して紹介する。このモデル化とプログラム化は、発注者とベンダーの責任分担の新しい取組と言える。企業経営のニーズとそれに応えるハードウェア製造の様相も参考として報告した。

Processes of A Software System Development

Zenya Koono*, Hui Chen † and Hassan Abolhassani ‡

This paper focuses on the beginning period of a software system development, named as Requirement analysis/acquisition. Authors' studies show that the hierarchical detailing, in the initial stage, should be made by 'Hierarchy of object', then that, in later stages, should be made by 'Hierarchical decomposition chain of functions'. Based on these, activities of a business organization may be modeled by a hierarchical structure, consisting of pairs of data flow (business flow) and each control structure. An enterprise system development may be easily achieved by converting thus gained hierarchical structure to programs. This will be a new model for a contract between a company and the vendor. Hardware production process, responding management needs are also introduced.

1. はじめに

ここ数年来、我が国では情報処理システムの大きなトラブルが社会を震撼させている。この種のトラブルは、以前から日経コンピュータ誌に「動かないコンピュータ」として報じられ、トラブル毎の単行本も出版されている。運用開始延期やダウン等の明示的なトラブル以外に、能力や過負荷耐力の不足からおこるサービス水準の低下もある。これらから、学会の鼎の軽重を問われる迄に至っている。

技術問題のみではない。ガートナー社は数年前から、情報システムの投資効果について、経営者側からの声高な不満を伝えている。外部からの声でシステム統合時期を延期した後に、更に大幅延期を公表せざるを得なかった事例までであった。経営者は技術問題で責任を取られるほかに、自分の足許を揺るがされる。それやこれやで、各社の情報システム部門の能力不十分は否めない。しかし、これら業務は今や企業の中核業務である。「捕えて見れば我が子なり」でも、大事に育てねば

ならない。如何に扱えば良いのか？

全ての問題は、実態であるシステムの「作業工程」の中にある。本報告はエンタープライズ系システムの開発の上流に焦点を当てる。筆者等は「作業工程」の外部特性を調べ、また内部の知の働きを調べてきた。この視点では、問題は「ヒトの意図的行動」の枠組のあるべき姿からの逸脱から生じる。初めに各種の工程の在り方を概観し、筆者等の構造論を説明する。次に実態を下流の製作段階を見てその原因を作る上流の実態を明らかにする。この理想論と同思想で、既に成功した実例がある。簡単に紹介するので、是非原資料をご参照下さい。

2. 開発プロセス

2. 1 基本的な開発工程

モノ作りは歴史的に下の3段階で進化してきている。

- | | |
|---------------|----------|
| I. 製造 | 太古～19世紀末 |
| II. 設計→製造 | 19世紀末～ |
| III. 研究→設計→製造 | 20世紀初期～ |

常に先端が分化してきた。残された後続工程は基本技術は一応確立しているから、生産性や品質等の向上/高度化に重点を置けば良い。そこで先端部の高度化を行

* Creation Project koono@vesta.ocn.ne.jp

† 国士舘大学 chen@kokushika.ac.jp

‡ Sharif University of Technology Abolhassani@sharif.edu

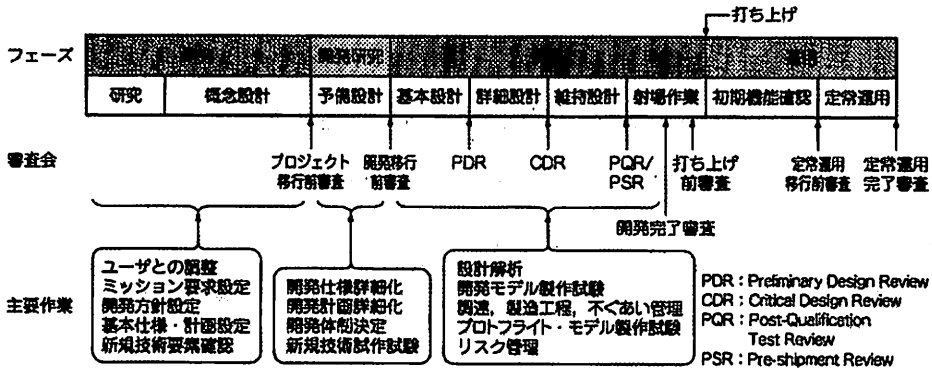


図1 人工衛星システムの開発工程

うパターンである。企業内の一般概念では、「設計」とは通常作業で可能なレベルの作業、「研究」とは「個々のトラブルを超えた問題の解決、特に後続設計の作業を可能にする」ことである。

IIIから各種変形が始まった。事業化の事前評価を強化する為の下が産まれた。

IIIa. 調査→研究（試作含む）→設計・

商品開発では、ニーズに応える為、漠とした研究は

IIIb. 調査→企画→計画→設計・

の如く、創る決定過程が商品向きに詳細化された。

更に進んで、アポロプロジェクト等成否が不明な段階から出発する場合も生じた。これに応じてシステム工学が誕生した。これは「分割し統治する」思想で工程を分割し、確実度を増す適当な手段を各個に取る。

IIIc. 目的/目標の明確化→複数案から絞込→可能性検証→概念設計→開発・

図1 [杵原06]は人口衛星開発の実例で、新しい各種方法も取込み応用すると報告している。図の用語を数箇所変えれば、システム開発を確実にするように使える。

一般産業界では、常に現存する名称（概念）を尊重して用い、自らの作業を容易化し、他の理解も容易化する。往時はソフトウェア開発も上記の各パターンを用いていた。しかし、何時からかソフトウェア工学では、下が標準である。

S. 要求分析/定義→設計・

これは、「要求/仕様を変換してプログラムを得る」と考えるソフトウェア工学的な立場、或いは、発注者に対して受注者側からの発想で契約から出発する為であろうか？要求とは機能ではなく目的概念で、分析とは手段の概念である。これでは何をやるかが判らないから、「如何にすれば要求が獲得できるのか」が課題となった。ここで「研究」が現れないことは不思議に思える[河野95, Koono00]。メインフレーム系システムの開発技術が一応成熟した時期には、「要求」ではなく

Sa. 「要件」の定義→設計

が基本とされ、情報処理技術者試験の参考書等は、「要

求」と「要件」を区別して設計手順を説明していた。これについては後で触れる。

2.2 システムの開発作業

専用のシステムでは幅が絞られるから、前記IIIのいずれかで対処して不可なら研究している。冒頭に述べたように、最も難渋する所謂エンタープライズ系の上流を中心に、筆者等の研究結果を説明する。

情報システムの使命は現状から「何かを改善する」ことで、設備投資の一種である。図2で改善の原理を説明する。横軸は利用年数（製造設備なら製造個数）、縦軸は総支出金額で、太線部分は改善投資額に当たる。改善前は実線で表され、直線的に支出が増大する。改善投資をすると、年当り経費（1個当り製造原価）は減り（線の勾配は低下し）、総支出は改善投資額から出発した破線になる。2線の交点から先の領域では、投資した方が総支出は減る。（改善効果を増し、投資額を減らす。）

発注側の経営者は、1. 改善の効果/効用、2. 設備投資額、3. 導入時期やリスク、等を勘案して発注を認可し、約束厳守を命じる。経営者は各分野に対応するgeneralistであるから、通常3案程度を準備し、経営者に理解願える言葉で説明する。経営者は何れかを選択し、資金（カネ：経理）、人員（ヒト：総務）、場所（モノ：施設）の準備等を指令することが仕事である。

図3 [中村03]は多数の企業の情報化投資の目的を調査した結果で、圧倒的に原価低減が多い。

ハードウェア組立て原価低減の設備投資なら、製造（生産技術）部門が責任元になる。担当者は、各種作業の時間（工数）を計測し、新設備の導入後の低減した工数や増強後の能力等を計画する。前記諸条件を満たさねば事業運営の醜態を招く。確実にする為、全てを定量的に把握した上で夫々に適宜マージンを取り、少々のは自己

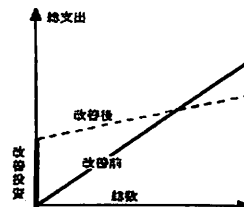


図2 改善投資の効果

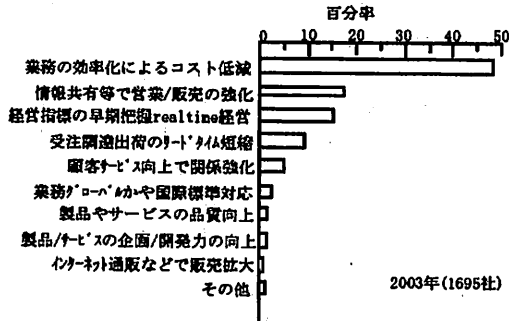


図3 情報化投資の経営上の目的

責任で処置可能にする。背負い切れないことは予め明示して、全体のリスク対処策を予め決める。これら定量化は Industrial Engineering, IE分野の教科書、また設備投資については生産技術系および会計関係書籍に書かれている。全投資責任がこのように動くから、経営者は情報システム部も同様と期待するし、経営者育成上から「どの部門にも決して甘い顔は見せてはならない」ことが原則である。

製造等の(直接)作業も研究/設計等(間接)作業も、具体化過程は「人の意図的行動」で説明できる。具体的には「人の概念展開連鎖」が基本原理で、最上流では「目的の階層性」[Calusewitz1832]を用い、以後は機能の階層展開の連鎖を用いる[陳97, Abolhassnai02, 河野04]。

「目的の階層性」は、過去の戦争の実績例から帰納された戦争計画の基本原則(例:図4a)で、一般向けに説明すると下記になる。

最高指揮者には、最終「目的」が課される。この「目的」を達成する為、これを実現する「手段」群に階層展開して、各「手段」を(海、空、陸などの直交的な機能毎組織の)部下に指令する。これは各部下の「目的」になる。夫々の部下は、これを実現する「手段」群に階層的に展開し、自分の各部下に与える。以下、これを繰り返す。目的は次々と階層的に展開され、最後の一兵卒の動きにまで至る。この上位階梯は戦略(計画)、すぐ下位は戦術と呼ばれる。(階層的組織が前提になっていることに注意。)

「戦略の誤りは戦術では取り返せない」と云われ、最上位は重要な決定である。経営者は最終目的である財務

諸表上の成績向上を考え、戦略を考えて経営資源(ヒト・カネ・モノ)を割振る。情報システム部長の仕事は、事業コスト、製品の市場評価や社外事故統計並びに大きな情報関係の技術動向から、事業の為に何が最も有効な策か/如何に有効にするか、であろう。

他の設計/製造/品質部門の長は、夫々の分野で投資の狙い所を必死に探す。情報システム部門の長は、まずは全組織の共通な通信・計算機システム、次いで自部門では合理化計画ができない経理/総務/営業等の間接部門の合理化投資、自部門の経費が狙い所になる。システムの維持管理費用は、年々の維持変更拡張でソース規模が約10%/年で増えることにより増加するので、初回規模を小さくすることが肝要である。

経営の要諦の一つは、経営に関わる指標を決め、継続的に時系列、および横並びに監視して問題を早期に検出し改善することである。改善策を講じて、パレート図[水野00]に表した上位の数項目を取上げ、「選択と集中」の原理で集中的にカネ・ヒト・モノを投じて、期待効果を実現する。以後は未了項目がパレート図に浮上するから、前記を次々と繰返し、改善を抜ける。

情報システム部門の第1線では前記の「目的の階層性」を情報システム向きに階層展開し具体化して、

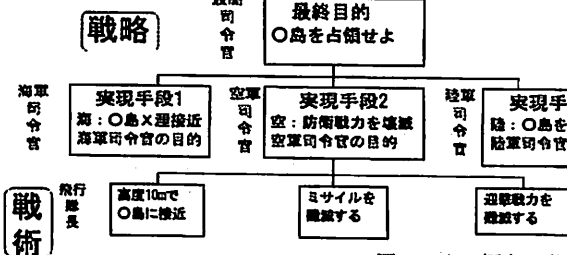
効果/効用→サービス→外部機能→内部機能に落とす[Koono90]。外部～内部機能レベルに落ちれば、機能を下のように階層展開することを繰り返す。

外部機能→内部機能→下位機能

最後に単位的処理になれば、この処理を実現するソースに変換すれば良い[陳97]。図4bはこの展開例で、入力(実時間クロック)から出力(時計盤面)迄の単位データフロー「時計」(親概念)が、下位の3単位データフロー(子概念)に詳細化され、また左上の開始記号から始まるフローチャートは3機能を抜き終了記号に至る。次は、下位の各単位データフロー毎に同様に階層展開する。展開毎に概念が具体化すること、データフローとフローチャートの対に注目されたい。これは、機能の階層展開のみ考える従来の構造化設計の欠陥を明確化し、Data Oriented Approach, (DOA)の基本原則になる。

機能の他に応答時間や操作性良好等の「非機能的要求」がある。これにはTQM (Total Quality Management) で開発された品質機能展開 (Quality Function Deploy-

a. 目的の階層性



b. 機能の階層展開

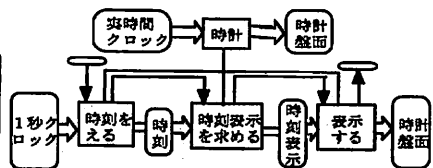


図4 人の概念の階層的展開

ment, QFD)[進藤 98]を応用する [Koono92].

例えば応答時間を指定値以下にするには、最終条件から出発して、外部的に定められた時間を階層的に定量的に各部分での計画値に配分する。方式設計ではディスクアクセス回数とその時間の枠を守らせ、プログラム設計では内部処理時間の枠を与えて設計させる。同様に、見易さ等の概念規定も階層的に展開して各個に具体化し数値化する。操作性等は、応答時間は〇秒以下等の定量的基準の他に、最下位で「文字は1行〇〇字、〇ポイント以上」等数値化して仕様に織込む。

サービスや外部機能等を階層的に具体化しながら、階層的な職制毎に了解を取る。最上位レベルは部長の了解、中間レベルは課長の了解、詳細レベルは(例えばモデル画面を用いて、経営幹部に対する場合と同じく)相手が判る言葉で話し、現状認識を合わせ、改善後の作業を説明し、協議する。紛糾する対人交渉等は、

工数は作業時間や操作数等

操作性は画面数、文字数/画面数、応答時間等
デザイン等は標準画面と操作手順等

包括的な定量指標値で予め了解を取る。要は、

- ・誠実に話し合い、先方に利があれば自らを修正し、約束は厳守する
- ・確実に約束できることを掌握し、必ずマージンを確保する(処理能力は70%になる等の経験則があれば、初めから、能力は1.4倍に計画する)
- ・計算機やソフトウェアに関する情報は、上記過程で含んではおろすが、露出せず、自由度を確保する
- ・詳細はシステム部門に一任願う。

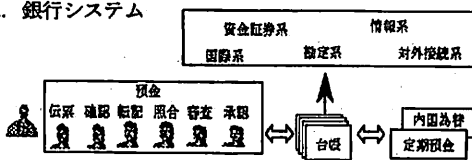
組織の階層的な職制に合わせて順次階層的に定めた指標で了解を取る。上長と取り決めた範囲内で具体化した結果をその部下達と打合せ、上位了解に反する事項には論争しない。了解範囲を超える異論は上位の職制内で始末して頂く。このように立場性や他分野知識に基づく異論を予め防ぎ、実現側の自由度を確保する。

以上の結果の仕様にベンダーに渡し見積らせ、仕様と期限を守らせる。内部で変更やミスがあっても、ベンダー側には極力影響させない努力をする。進行に応じ社内でも中間報告する。初めから旨くは行かない。発注の度に遣り方やマージンを改善して行く。これらはハードウェアの設備投資に際して皆が行っている。

2. 3 システム開発のパターン

発展の経過で実態を調べる。入力から出力情報への

a. 銀行システム



データの流れを銀行システムで、次ぎにその流れの上位の制御構造の例を生産管理で説明する。

図5.aは銀行システムの概念図で、左の枠は預金の場合の業務の流れである。オンラインシステムの導入以前は、利用者から払出し伝票と通帳を受取り、以後何人もの女性達が夫々の確認/処理を行ない、最後に責任者が台帳記載/支払を承認する。今度は逆順に記入済通帳と現金が戻り、利用者に引渡される。(女性群がデータフローを構成している。) 定型的金銭処理が多く、早くからPCS (Punch Card System), 次いでアセンブラプログラムと計算機が用いられた。

1960年代半ばに第一次オンラインが稼働し、利用者がATM/CDを操作して処理は自動化され、人々が郵便局に代わり銀行を利用し始めた。ソフトウェアが多数の女性行員が不要にし、省人化を実現した。1970年代半ばの第2次オンラインでは、図下の科目毎の自動化から進んで全体的な自動処理が行われた。これで昼間処理の人員が省かれ、利用者の利便も更に増大した。1980年代半ばの第3次オンラインでは、通信網の共同利用が可能になって更に合理化が進み、システムの統合 System Integration により資金運用等の戦略的な運用やサービスが実現された。最上部はその区分を示し、全体規模は10MLの桁と称される。これらの進歩は、まずデータフロー中の女性群の処理を自動化し、次に省力化以外の色彩が増し、最後に戦略的機能/サービスが中心になっていることに注目されたい。

要求が仕様に直結するのはPCSやアセンブラ時代で終った。初期のシステムは、顧客先に無料で常駐したシステムエンジニア達も銀行員に協力して作成した。協力業者は、貴重な知識と技術習得の機会を得た。これで、標準的な工程、標準的なプログラム構造、外部機能/サービスの標準化技術が確立したと思われる。

この技術は業界に階層的に広まり蓄積された。一から十迄顧客が指導せねばならない幼稚な大工に大事な家を立てさせる人は居ない。情報業界は「要件を得て詳細は自分達で設計する」技術が身に付けて工程Saに進化し、他産業と同じビジネスモデルになった。

銀行の業務は、関係法令と示達、利用者との契約(約款)、業界の一般通念等で決まる。既に、発注受注間で裁判沙汰も出ている。発注者側が実損を蒙った場合には、会社は株主に裁判に訴えられる。現在では、

b. 生産管理システム

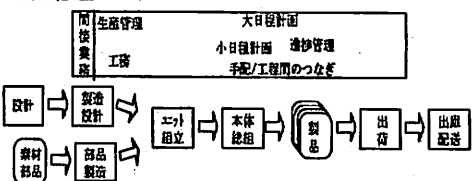


図5 各種のシステム

情報システム部門は(銀行業務なら法規から一般通念に至る迄)詳細で明解な仕様を文書で伝え、トラブル処理や負担も明確に契約で取決める必要がある。

図5bは製造業の作業の流れである。仕様からの流れは、設計を経て部品の種類個数が図面になり、製造設計を経て印刷回路基板の銅箔パターンや基盤間の接続図面になる。素材部品からの流れは部品群になり、電子回路ユニットが組立てられ、本体総組でユニットの間の配線を行ない、各ユニットを装着し、動作確認を行って製品を完成させ(管理区画である)倉庫に入れる。銀行での女性群の流れと同じく、各作業が流れを構成している。銀行での女性達がシステム化で省かれたと同様に、製造では各種設備で作業を減らし、最後に人が居なくなる。

製造業では、上記の製造ラインで働く人々が一定テンポで作業に専念し熱中できるように、その他の人々が最大限の努力をする。言葉によると、ラインの人々は仕事所内で最も尊重されている。(ソフトウェアでは?)

図の流れの上の枠は生産管理業務で、上記の為に存在する。(図4bのフローチャートの制御構造に当る)枠内の下は工務系の人々で、図面に記載した部品素材を発注し、作業開始に間に合わせる手配をする。入荷/先行職場が作業完了すると、各部署に現品を届ける。常にラインが全速力で継続的かつ定期的に安んじて作業に専念できるようにする。彼達は日程計画を現物ベースにより、話しながら顔色を見て進捗管理している。

枠の上部は生産管理系で大日程計画を立てて指令し、進捗を管理して、納期に合わせて出庫する。大日程計画で大工程に区切ると、工務が小日程計画を作り、職場毎に割付ける。部品の加工や組立て等の作業には必要な標準工数や標準日程が決めている。時間を横軸に日毎の所要標準作業員数の図(工数山積み図)を作る。

作業員は作業別能力別に格付けされ、標準員数から実人員に変換して、当該時期の未割当な保有人員数と照合する。凸凹の山があるから、ある時期には保有人員を超える。そこで各種の調整(山崩し)を行う。予め作り貯め、作業数を分割などして平準化する。これが不可なら、より優先度の低い作業をずらせ、または効率を下り原価は増えるが、他職場から応援を計画する。これも駄目なら外注を手配する等である。徹底的に定量的で合理的かつ科学的に職場毎の日程計画にする。

工程マンは、先手管理に徹しあらゆる智慧と手段を使い、何が何でも決まった日程計画を守り抜く。第一に「目的の階層性」を使い、策を考えて全て事前に手を打つ。約束しても各所で日程が狂うから、定期的に事前に再確認する。現場作業員達のモラルを監視し、各職場/発注先にも予定の再確認を度々行い、必要なら談判する。自分で処理できない遅延は、係長・課長・部長を動かして先方に交渉させ、優先度を上げさせる。

これら努力の結果、各職場は約束とおりの日時に必要材料類を受取る。各職場の作業員は計画は定量的合

理的であり、自分の給料や評価に響くから、全員がベストを目指して作業し、指定日時迄に工務に引継ぐ。両者は約束を守り、守らねばガンガン叱り飛ばす。製品の納期は確実に守られ、かくて工務マンは一息つく。

相互に緊張感を持つが、全員は同一の目的を目指す戦友同志であり、共に頑張って仕事を予定通りの完成が皆の誇り、また各自の達成感の源である。

最初からこれが出来た訳ではない。各所で改善を積み上げ、次第に高度化した。産業全体のレベルも向上し納期も確実にになった。トヨタのカンバン方式/Just-In-Timeも、トヨタのあくなき努力の元に各社が努力し、産業界のレベルが向上して実現できている。これら血のにじむ努力の成果として、今の日本製品の高い品質と確実な納期、他では時間通りに動く列車電車、が実現された。(所でソフトウェアではどうであろうか?)

1970~80年代には、各社の生産管理システムが発表された。この生産管理の働きは、データフロー状の製造ラインの制御構造で、生産の頭脳とも神経とも言える。この情報システム化は、各人が持つ技術を計算機に任せ替えることに他ならない。彼らは自分達の異常対処策や正常促進/確保策が正常な生産を支えていることを熟知している。生産管理システムの仕様化には、各社とも各職場からのベテラン/エースを出し、彼らは自分達が楽になる期待も持って、システム部門の人々から仕様作成の方法を教わり、集中作業で仕様を作り上げた。

単純なビジネスの処理フローと製造でのモノのフローを見た。ここでは、そのフローの上部の制御構造の例として、生産管理を説明した。これらは図4bのデータフローとフローチャートの対応と同等である。単純なひと流れの生産を説明したが、装置レベルの下位には部品製造のラインがあり、上位の装置レベルに同期して作業を進めている。即ち、階層を構成している。このようにフローと制御構造の階層が出来上がり、全体は階層的な論理構造を構成する。次にフロー中の各作業の作業/処理手順を明確化する。各作業/処理フローと制御構造をモデル化してプログラム化できれば自動化できる。間接業務でもこれは行なえる。

2.3 発注者とベンダーの実態

発注側とベンダー側の実態を見る。ソフトウェア開発従事者からは、顧客側「仕様が決まらない、遅れる、変る」と苦情が高い。仕様変更は、意気を喪失させてやる気を無くさせ、皆のモラルを大きく損ねる。まず第一に「ムダムラムリ」は退治すべきもの。実態は?

図6はCMM成熟度レベル5を誇るベンダー、ジャステック社の仕様変更の管理方式の説明図[太田05]であり、仕様変更に伴う廃棄や追加が明示されている。契約時に、変更タイミングと追加・廃棄量を取決め、ある時点で仕様変更を止める。更に、仕様変更の基準と仕様変

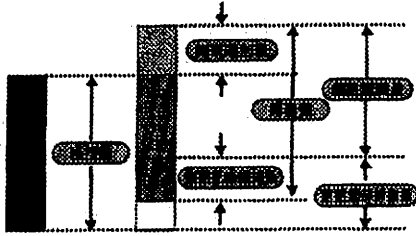


図6 変更の管理

更に伴う正味廃棄量を相互合意しておき、「予定量の仕様変更は尽きた。これで止めましょう」。ベンダー側が仕様変更に悩む様相が良く判る。実態は不明だが損失は〇十%と噂される。仕様変更の損失は、%の桁以下でなければ正常な品質高い作業は困難である。現状は産業界の常識では、ひどい異常事態である。「お客様は王様」とは云え、相互に襟を正すことが第一ではないか？

発注側はどうか？図7[中村03]は情報システム部門の悩みの調査結果で、平均値に当たる。仔細に図をご覧ください。(人員不足は全部門に共通の悩である。)

特異な項目を検討する。

- 「利用部門の意識が低い、経営者の意識が低い」-計66.3% 良く聞かれる声である。経営者はgeneralistで、技術者とは限らない。各部門の長に業務状況を管理させ改善を提案させて、最も効果的に経営資源(ヒトカネモノ)を配分する役割である。経営者や関係部門と改善の方向等を打合わせて経費の配分を受け、投資を決める。これらの中で、各種の具体的な問題点や各種新技術や動向を経営に關係させて伝え、トップと関連部署の理解を促し教育する。「経営者や利用者の意識が低い」のは、情報システム部門からの働掛が足りない結果と思えるが、如何か？
- 「情報化投資の投資効果が解らない」-36.4% 全体共通あるいは総務/経理/営業等の改善提案部署は情報システム部門である。これまで如何に職責を果たすかを具体的に縷々説明した。-投資の投資効果が解らない-とは！自己責任を意識しているのか疑問に思え、率の高さに大きな衝撃を受ける。

3. 成功事例

2.2の方式は理想ではあっても、現実には行い難い机上論と思われよう。たまたま、前記の方式で成功された事例が既にあったので、概要を紹介する。

大成建設の経営陣は、本業の技術開発費ほどにも膨張した情報システム関係コストに危機を感じ、木内里美氏を実質CIO(Chief Information Officer)、社長室情報企画部長に任命した。同氏は2年で3割引下げるコミットメントをしてプロジェクトを推進し、所期の目標を達成された。これが認められ、同氏は日経情報ストラテジー誌の初回の[CIO of the year 2003]を受賞された。

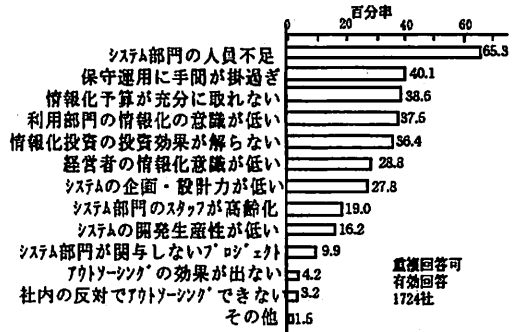


図7 システム部門の悩み

同氏はプロジェクトとその技術を各種発表[木内03, 04a, 04b, 05]されている。借越ながら筆者の責任でお話を抜出して切貼した。以下に紹介させて戴く。

われわれのように社長室に情報企画部門を置く企業も増え、CIOの立場も認知されつつある。CIOは全体を見回す立場に立ち、情報化戦略を進める。この実行の為に必要な執行権限と責任を持つ。部門間調整力、リーダーシップやコミュニケーション能力、トレンドを見ながら、話合える技術力も必要だ。自分の後継者もCIOになれるように育成している。

ITと経営の連携がよく話題になる。しかし、経営モデルとITへの取り組み姿勢が一致していないケースが多い。会社は縦割り組織で構成され運用される。それはそれで良いのだが、問題は縦割り構造に刺す横斬がないことだ。基本認識として、現在状況を改善するためには、経営者がITに取り組んでいく方向付けを行い、コミットメントを出して情報システム部は横断的な部分の企画を立て、提案していく必要がある。

このプロジェクトでは、情報インフラである基礎設備を整備してメインフレーム撤去等を行ないつつ、Power Projectを推進して業務改革を行った。Projectの対象は、人材管理、契約管理、会計管理等である。

社長室経営企画部と協調して、システム改善のために協議会を作り、様々な部署の関係者が入って改善し、運用・メンテナンスを情報部門が行っていく体制を作った。これらは社外の人には出来る領域ではない。情報技術者が出来る仕事でもない。業務に精通している人のみがデザインできる。業務システムのデザインは、現場の人たちが行わないと定着しない。留意すべきは、利益代表がリーダーになってはいけないこと、現在は他の部署と並列ではない社長室が見ている。

変革を成功させるポイントは、社内共通言語と共通手法を取込むことだ。可視化もそのひとつで、愚直なまでに基礎的な努力をきちんと行うことが重要である。ITコスト削減であれば、コスト構造が見えるように可視化し、着実に一つ一つ取り組む。

具体的には、最低限のコードの整備、標準化をまず先行させた。DOA, Data Oriented Approach 手法により、全体を Data Oriented な仕組みに統合してこうというアーキテクチャーである。全社情報マップを作る。これはシステムコンセプトを実現させるためのデータがどう動くかという全体がわかるマップである。モジュールのどこのデータがどうつながっているかの連動性を最初から考える。社内の情報管理図、管理マップも作った。全社の情報管理、ヒトモノカネの情報をマッピングした。マッピングした情報を小冊子にして各部門に配った結果、データの重複やつながりが悪い構造を避けるために役立った。

具体例：支店の総務系業務をすべて洗い出し、体系化した。業務をアクティビティごとに整理しチャートに描き出す。まず個々のプロセスを改善する。最初に業務体系表を作り、それぞれの人がこんな仕事を、こういう手順で処理しているというワークプロセスを全部フロー図に書き出す。このように可視化して、「この業務を省けるんじゃないの」とか、「この業務と似た業務と一緒にやればいいんじゃないの」とか、このように改善を話し合った。業務プロセスを可視化して、見直し、改善して、プロセス設計をきちんとしてから走り出さないと成功しない。

これで管理のポイントが明確化し、業務マニュアルができ、担当者のレベルと作業時間が判るから、これに金額を掛けると、コスト計算も明確にできる。各種の管理図類は、今も定期的に更新し続けている。

こういうことは『やらされている感覚』では駄目。でも、こういうことをじわじわやっている、「これでちょっとチェックしたい」とか、別の部署も言い出す。製造業の「改善」はもっと凄まじいと聞いている。こういうことをやるのは、やはりとても大変だ、けどどういことをやらないから、システムがちゃんと出来たのだ。他によい方法があればいいが、結局地道な手法しかないと思う。

情報システムは結果的に出来上がった業務プロセスを効率的に処理できるように組まれるだけだ。こういう前処理をしっかりとやるのがものすごく大事だ。

こういう手順で上流設計がしっかり出来たら、システムを作ることなんてそんなに難しいこととは思えない。

話しは変わるが、情報システムの受託ソフトウェア業界と建設とは、労働集約的/多層構造/プロジェクトマネジメント等似た所がある。しかし、プロジェクトマネジメントの方法が全く違う。建設業は期限を守らず品質を保証しないことはまずないが、ソフトウェアベンダーでは許されてしまう。(その他の詳細を下の一覧表に集約した。)

・労働単価が曖昧なこと

* PG, SE の経費込み単一単価を示すのみ

- * 建設業は基準労働単価があり、積算の元になる
- ・積算基準がない
- * 歩掛かりや標準積算の考えがない
- * 積算基準も持っていない
- ・出来高管理をしていない
- * 下請企業に対して完成払いが標準
- * 出来高管理型の労働管理や外注管理をしていない
- ・工期遵守意識が希薄
- * 工期の遅れ=ペナルティの意識がない
- ・設計変更契約の概念が無い
- ・建設業での品質確保の推進に関する法律等がない

これらから、同氏は、行うべきことを上から下迄的確に掴み、強い統制力とリーダーシップを揮われた様子が判る。またトップから始めて、「愚直」とも記されたがボトムまで徹底的にさらって推進された、その熱意と実行力は人の心を動かさずにはおかない。同氏とチームの作業は、自分達の Business Process Re-engineering, BPR を行いながらデータとフローを作り上げられたと云える。これは自分的で行わねば出来ないことである。これを元に処理構造を仕様にすれば、明確に理解できて、変更も格段に減り、作業効率も遥かに高くでき、品質も向上する。かくしてベンダーはシステム化中心に十分な力を蓄える。これは発注側のあるべき理想形を實踐し成功した例で、今後の模範になる。

他の実例[大和田01]は、カード/信販事業を行う株式会社ジャックスが IBM に発注した JANET システムである。規模 9.1M ステップで、現時点で民間では世界最大と云う。技術的には DOA によるもので、ジャックス社の多数の業務精通者から IBM 技術者が詳細を聞質して DFD, ERD を作り、ジャックス者が確認して仕様とした。(ハードウェアでの承認図方式) 以後 IBM 側でプログラム化し、COBOL 自動生成等を用いて作成した。他の第 1 の要は、両社が実務に長けて強いリーダーシップの方に大きな権限を与えられたこと、第 2 は組織的な管理である。詳細は原資料に詳しい。

4. おわりに

エンタープライズ系のあるべき開発工程、実態と成功例を説明した。これらから下記の知識を共有したい。

- ・要求からプログラム化できる対象もあれば高度で複雑な対象もある。各種の工程を考えて使いたい。
 - ・企業内の単位的な活動は、
 - 入力から出力へのデータ流 (データフロー)
 - その制御構造 (例: Calling sequence)
- の両者で構成された機能箱で表せる。これら機能箱群は上位のデータフローを形成し、その制御構造がある。これは最上位まで繰返される。原始的レベルでは全処理/作業は人が行なうが、最高レベルでは (銀行システムの如く) 全処理/作業は計算機

が自動処理する。今は平均的にこの中間にある。

・データの流れ/ビジネスフローが主役になる。

これらをソフトウェア化するビジネスモデル/工程を下記に詳細化しては如何か？

・基本モデル ⇒ (IIIa)

・商品開発特化モデル ⇒ (IIIb)

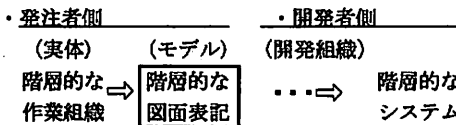
・高度複雑向けモデル ⇒ (IIIc)

各段階で、下記を使えば「何を如何にするか」が判る。

・上流 「目的の階層性」

・中～下流 「機能の階層展開連鎖」。

システム開発は下のフローで表され、作る機能/処理の流れ図等 DOA 方式が最もインタフェイスに適する。



作成指示～承認図

この原情報は発注者側にしかない。インタフェイス情報は自ら作るか、または、開発者側に伝え作成した結果を審査承認する。一方、システム構築技術は開発者側にしかない。発注者側で包括的な仕様を与え、打合わせながら詳細化する。このインタフェイスを責任境界として、明確で厳格な契約制に移行できる。

情報システム部門の人は、周囲が事業意識を軸にして感星状に動く地動説社会の中で、ひとり孤独で天動説の世界にあると懸念する。各位に期待される作業を繰々説明した。この他の所々に同傾向が現れる。各位は現代の基幹人材である筈。やれば出来る。業界は社会人を教育する必要がある。大学での教育にも織込むことが必要である。工学部/経営工学のJABEEは共通化し、理学系の情報科学と明確に分ける必要があると考える。

謝辞

本報告の研究と作成には多くの方々のご協力を戴いた。木内里美氏から、引用と切貼一本化のお許しを戴いた。皆様方のご協りに深く感謝し、お礼申し上げます。

参考文献

[Abolhassani02] Hassan Abolhassani, 河野善彌, ソフトウェアクリエーション: ルールによる自動設計と知識による自動設計, 情処研報, ソフトウェア工学 138-15, pp. 105-112, 2002.
[Cluasewitz1832] Cluasewitz, Karl von, Vom Kriege, 1832, 淡徳三郎(訳), 戦争論, 徳間書房, 1965.
[陳97] 陳 慧, Far B. H., 河野 善彌., ソフトウェア自動設計における系統的なエキスパートシステムの構築, -設計工程からの設計知識の獲得と再現-, 人工知能学会誌, Vol. 12, No. 4, pp. 616-626, 1997. 7.
[Chen98] Chen H., Tsutsumi N., Takano H. and Koono Z., Software Creation: An Intelligent CASE Tool Featuring Automatic Design for

Structured Programming, The Journal of Institute of Electronics, Information and Communication Engineers -Special Issue on Knowledge-Based Software Engineering-, Vol. E81-D, No.12, pp. 1439-1449, 1998. 12.

[橋原06] 橋原弘樹, 設計品質確保の思想-航空宇宙エレクトロニクスに学ぶ「信頼性設計」-, 特集1 ソフト&ハードの設計品質改善活動 (開発プロジェクトの破たんを防ぐ処方せん第1章), Design Wave, no. 2, p. 36-45, 2006. 6.

[木内03] 木内里美, 時めくパーソン木内里美氏「CIO オブ・ザ・イヤー2003」, @nifty 建設ポータルサイト C-PAS, インタビュー URL: <http://const.nifty.com/ipindex.htm>.

[木内04a] 木内里美, 大成建設における経営基盤の再構築と業務改革活動の実践～ITからアプローチするBPR～, 創造都市研究都市ビジネス専攻システムソリューション研究分野ワークショップ平成16年前期1, 2004年10月. 講演録, スライド URL: http://www.gscc-sa.com/J/workshop/2004/ws_11.html.

[木内04b] 木内里美, 「CIOの役割とは ～経営戦略とIT～」, 日経デジタルコア, 経営の可視化とIT研究会, 第2回研究会, スピーチ1, 2004, 12, 14.

URL: <http://www.nikkei.co.jp/digitalcore/report/041130/index.html>.

[木内05] 木内里美, ユーザ企業から「勝る」経営とITの品質～品質の作り込みの観点から～, 品質管理学会第24回ソフトウェア品質シンポジウム, 特別講演, 2005年9月. スライド URL: http://www.juse.or.jp/software/pdf/050908_3.pdf.

[Koono90] Koono Z. and Soga M., (28) Structural way of thinking as applied to systems design, IEEE Global Telecommunications Conf. 1990, pp. 306. 1-7, Dec. 1990.

[Koono92] Koono Z., Structural way of thinking as applied to increasing customer's satisfaction, Proc. of 3rd European Conference on Software Quality, 5A 3, Nov. 1992.

[河野95] 河野善彌, 研究開発 vs. 要求仕様, 情処学会, サマワークショップ・イン・立山 論文集, pp. 121-127, 1995. 7.

[Koono00] Koono Z., Chen H., Abolhassani and Far, B. H., Structures in Requirement Engineering, Fourth and Fifth World Conference on Design and Process, pp. 14 (in CD-ROM), 2000, 6.

[河野04] 河野善彌, 陳慧, 人の設計知識構造と定量評価(1/2), 信学技報, KBSE2003-57, pp. 67-72, 2004.

[Koono90] Koono Z. and Soga M., Structural way of thinking as applied to systems design, IEEE COMSOC Global Telecommunications Conf. 1990, pp. 306. 1-7, Dec. 1990.

[太田05] 太田忠雄, ソフトウェアの生産管理に基づく見積もりモデル, SEC Jouanal, no. 2, p. 14-21, 2005.

[大和田01] 大和田尚幸, 実録・世界最大のSIプロジェクト, 日経コンピュータ, 2001年09月24日号, p. 170-179, 2001. 9.

[水野00] 水野 滋監修QC手法開発部会編集, 管理者・スタッフの新QC七つ道具, 日科技連出版社, 2000.

[中村03] 中村健助, 矢口竜太郎, 規模・業種問わず予算削減の圧力が強まる, 日経コンピュータ, 特集プロジェクト成功率は26%, 2003年11月17日, p. 64-71, 2003.

[新藤98] 新藤 久和 (編集), 赤尾 洋二, 吉沢 正, QFD 10年の成果, 日科技連出版社, 1998.

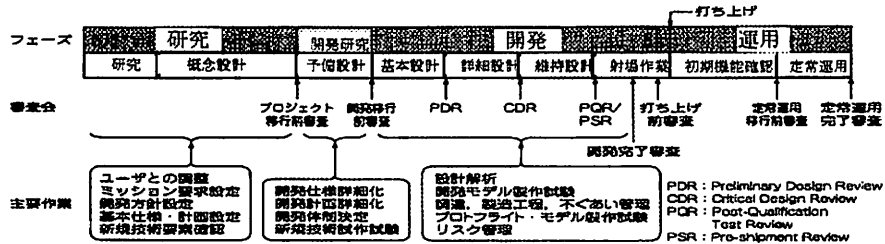
正 誤

情報処理学会 ソフトウェア工学研究会 研究報告 2006-SE-152 (2006年5月17日) の下記論文に誤りと不明瞭な図がありました。御手数ですが、下記(特に下線部分)によりご訂正ください。

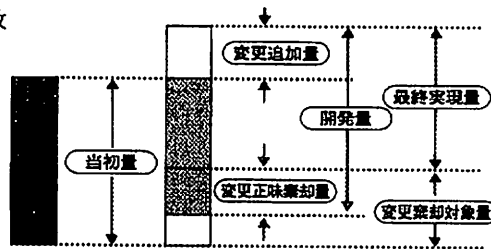
著者名：河野善彌，陳慧，Hassan Abolhassani

論文名：ソフトウェア開発工程の在り方

10頁 図1 ハッチ部文字改善



14頁 図6 ハッチ部文字改



字句訂正

14頁右下5行 誤 図6はCMM成熟度レベル5を誇るベンダー
正 図6はCMM I 成熟度レベル5を達成したベンダー

14頁右下2行 誤 変更タイミングと追加・廃棄量を取り決め、
正 変更タイミングと追加・棄却量を取り決め、

14頁右下2行 誤 更に、仕様変更の基準と仕様変更に伴う正味廃棄量を相互合意しておき、
「予定量の仕様変更は尽きた、これで止めましょう」。
正 つまり、仕様変更の基準と仕様変更に伴う正味棄却量を可視化した上での相互合意を前提に、「予定量の仕様変更は尽きたので、これで止めるか、新たな予算を追加し変更を継続するか調整しましょう」。

16頁右21行 誤 [木内05]木内里美，ユーザ企業から‘診る’経営とITの品質～品質の作り込みの観点から～，品質管理学会第24回ソフトウェア品質シンポジウム，特別講演，2005年9月。スライドURL: http://www.juse.or.jp/software/pdf/050908_3.pdf。
正 [木内05]木内里美，ユーザ企業から‘診る’経営とITの品質～品質の作り込みの観点から～，且科技連 第24回ソフトウェア品質シンポジウム，特別講演，2005年9月。スライドURL: http://www.juse.or.jp/software/pdf/050908_3.pdf (ピリオド削除)
以上