

エンタープライズ系システムのありたき開発方式

河野善彌*, 陳慧†, Hassan Abolhassani ‡

この報告は開発方式の提案である。この種システムの狙いは合理化が多い。期待される効果/効用を起点として、目的とするシステムを、トップダウンに、階層展開を繰返しつつ、効用/効果を定量的に確保しつつ、具体化する、考え方および実際の方法を提案する。これにより、この種システムへの投資は、現行の設備投資と同程度の確実度と経営者層の理解を得ることができる。終に、此の種の領域の「工場制」企業の技術について触れた。

A Proposal of an Idealistic Process in Enterprise System Developments

Zenya Koono*, Hui Chen † and Hassan Abolhassani ‡

This paper reports on a development process. Major system developments of so-called enterprise system developments aim at attaining various rationalizations. Thus, it proposes the concept and the procedure to set 'effects and usefulness' as the final object to be attained, and repeats hierarchical decompositions in a top-down manner, evaluating the 'effects and usefulness' quantitatively. By these, the investment becomes as reliable as other investments in plant and equipment, and asks understanding of top managers. Finally, technologies of 'software factories' in this area are 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トラブルが社会を震撼させ、続いて組込系システムのバグ続発が報じられている。「火の無い所には煙が立たない」と云う諺がある。この機会に将来の問題を防止することが重要であろう。これらトラブルの再発防止を目指してIPA SECに参集された関係者各位のご努力が蓄積され、良い方向に次第に進み始めた。喜ばしいことだ。開発の初動期を「超上流」と名付け、ここにトラブルの大きな根があるとの認識は、将にその通りと思います。

前前報告[1]は、発注者側に焦点を絞った。その問題は、仕様を確定できないことである。続く前報告[2]は、ベンダー側に焦点を当てた。ここでは、技術の育成が弱く、プログラムの視点に限られている。この報告はエンタープライズ系システムの開発について「超上流」を中心に、ありたき姿/プロセスを提案する。

養老孟子先生は情報や脳でヒトを説明する「人間科学」(筑摩書房, 2002)を著された。本報告もこれに似て、知の視点から統一的に見ることで各種差異を超えた議論をする。2章では開発に於けるヒトの知の働きを説明して、ヒトの意図的行動をプロセスの基軸にする。

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

† 国士館大学 chen@kokushikan.ac.jp

‡ Sharif University of Technology Abolhassani@sharif.edu

3章はシステム設計の方法を提案する。これは経営者(判りやすく云えば社長)中心に階層展開する方式で、定量評価をも組み合わせる。

- ・「発注者は何を作るのか、自分でも判っていない」と主張される。発注者側の誰に着目すべきか? 「いや! 上の人は判らないから、判る我々がやれば良い。」と云う声が昔から強い。担当者レベルで始めると、A氏は○と云い、B氏は△と云う。誰が正しいのか?
- ・19世紀末に社会学が産れた。この初期に「人は自分の立場でモノを云う」ことが明らかになった。A, Bの立場が違えば、主張が違うから、担当者では決着が着かない。では、経営幹部に「恐れながら超上流もご理解/ご協力」をお願いするのであろうか?

エンタープライズ系システムに責任と権限のある人は、社長等の経営者である。システムへの投資は、{ヒトの採用や昇進}, {カネである予算決算}, と並ぶ重要な{モノに関する決定}である。そこで、

経営者の要件からシステムを階層的に決める。
仕様を決めたら、

具体化された仕様はある時点で凍結し、
変更は一切出させず受付もしない。

システムが落ち着いてから、追加変更だけを入れる。

4章は以後の具体化関係を補足して、残りは技術の有用性とその集積の実例を説明する。

2. 知の構造から見た開発プロセス

よく「ソフトウェア/システムは、他とは違う！」と聞かされ、あるいは、それを前提とする話を聞く。そうだろうか？人の知の共通基盤があるのではないか？

1970年代のWirthの「段階的詳細化」は設計方法の嚆矢といわれる。これは「まず親概念を決め、後で詳細化（子概念）する」行動的発想である。続いて各種の構造化が現れた。これは前記に「機能の階層構造化」を加える最適化である。当時は、これで決着した。

プログラム教育の最初は、全てを自然言語で説明することが普通であろう。最後迄そのままで押し進めたらどうなるか？図1 a[3]はそのような1設計例で、ヒトの知の共通基盤である概念展開を用いている。仕様である「時計」に、第2段では入出力データを付加して単位データフローにする。これを親概念とする。

第3段では、これを子概念に展開する。これは直列的な3単位データフローの群から成立つ。第4段では、中央の子概念「時刻表示を求める」が第5段で子概念に展開されが、ここでは3並列な単位データフロー群になっている。（Jackson法の特徴である入出力データの階層展開結果を用いている）。

「親概念」は、概念展開の度に、より具体的/より明確/より詳細になって行く。以降を見ると、詳細で単位的操作に落ちた最後の段階で、等価なプログラム言語表記に移行している。人の記憶を中心的に用い、繰返し展開することで、詳細化していることが判る。

図2は、各種の単位的な階層展開を示す。図a[1]は先の時計の最上部で、詳細化と共に各機能を連ねる小さなフローチャートが出てくる。プログラム設計の中心は、人の概念展開の繰返しで詳細化した後、計算機を

動かす実現手段であるプログラム言語の表記にする。プログラムは最後の「設計結果として産れる」実現手段である。機能以上にデータ+機能の階層性が本質で、機能のみは不十分とするData Orientedも理解できる。

図b[1]は軍事科学で戦争計画の基本原則とする「目的の階層性」[16]の一例を示す。「最終目的」である「島を占領する」為に、司令官が海空陸の3軍を動かす「実現手段」に展開している。次の海軍司令官は、「X島にY海里接近する」目的を実現する為の幾つかの「実現手段」に展開して部下に指示する。以後、このような階層展開を繰返して詳細化する。組織に関係するが、組織も又階層状をなすことが判る。

図cは人の肉体的動作の場合を示し、最上位の行動の意図が階層的に展開され、下位の3動作になる。このように階層展開を続け、最下位では個々の筋肉への動作指令に展開され、意図した肉体的動作が始まる。

認知科学を初め、多くの科学では「ヒトは外界の事象を抽象化した情報に変えて、その情報を処理する」ことにより動くと考えている。この見方からは、外界の全ての差異はヒトの脳裏では消えて同一の知的機構が働くと言えよう。筆者等の研究[3, 10, 11]で人の意図的行動は、

- ・他の動物と同様な単純な記憶が中心的に働き、
 - ・他生物とは異なり、自然言語を用いて階層的な展開連鎖を構成させ、その上で自由自在に動き回られる。
 - ・得た知は情報として集積し、共同で、時と所を超えて利用でき、(他の生物は遺伝で継承するが)人は遺伝を離れて文化を継承して生きる、
- のである。図1, 2はこの階層展開連鎖の例と見做せる。

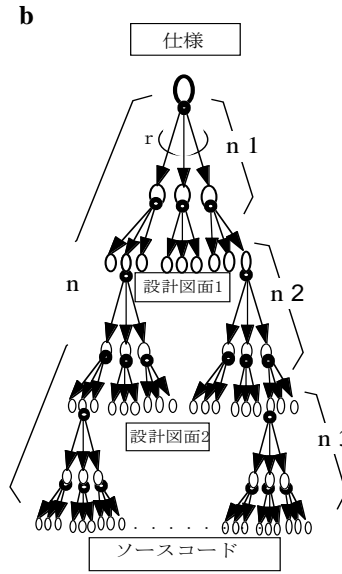
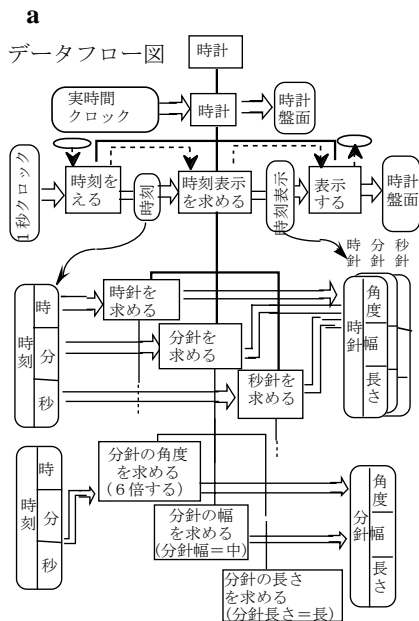


図1 プログラムの設計

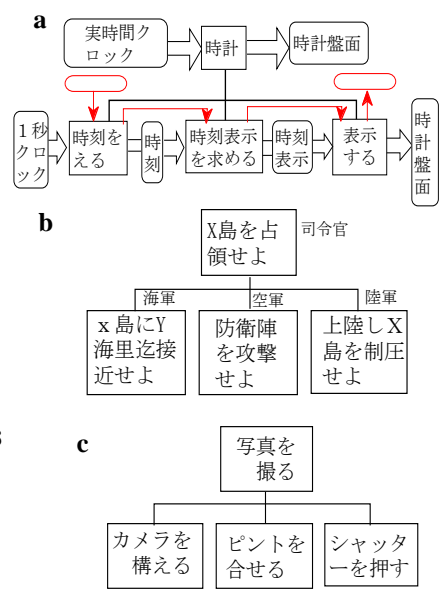


図2 各種の階層展開

図1の如く、プログラムは計算機を動かす媒体/実現手段に過ぎない。最終ランナーではあっても、プロセスの中心はこれ以前の概念展開連鎖にあり、

人の意図の具体化/創造過程は全て同一構造。

これは実現手段/モノに関わらず相互に独立である。ハードウェアのプロセス/管理の体系である Industrial Engineering (IE, 経営工学)[5]が、機械/電気/化学/生物/農業等から社会全体で使われている実績からも、これは明らかである。この中にシステム/ソフトウェア/プログラムも加わるに過ぎない。

3 エンタープライズ系システムの開発方式

3.1 効果/効用とその展開

「目的の階層性」は、全てのシステム開発の方法論を与える。だが、これは軍事用語だから、より相応しい名称にしたい。エンタープライズ系システムの多くは、間接要員の生産性/効率等の改善任務を持つ。この場合は「効果」の実現が義務になる。

ある場合には、システムの運用結果に具体的な「効果」を計上し難い。経済学は、経営幹部は『効用』を期待してITシステムに投資する」と説く。『効用』とは投資により意図した領域で得る定量的満足度と云う。以上から、「効果/効用」を最終目的の共通名称にする。

「効用」は定量性を前提とするから、評価は、評価水準を決めて○、△、× (○:5, △:3, ×:1) とでもすれば単純明解に行える。

現状の所謂ソフトウェア工学は定量性を欠き、実社会には受入れられ難い恨みがある。しかし、「効果/効用」は定量性がある。これを使えば、他の工学と同様な定量的合理的科学的な取扱が可能になる筈。「企業の利益/ヒトの利益」を定量的に扱ってこそ、「工学」になれる。)以上から、開発は以下を原則としよう。

定量評価で方路を選ぶ階層展開の連鎖

階層展開の意味を説明する。階層展開の決定は、初動期ほど大きな影響を与える。軍事では「目的の階層性」の最上位の展開は、軍の最高指揮官が行う。それは政治上の最高責任者から「最終目的」を課されて行うもので、「国家の勝敗」につながる。「効果/効用」の場合でも、企業の成果/利益に大きな影響を与える。

軍事では初動期の展開を戦略と名付け「戦略の失敗は戦術では補えない」と云う。この重い責任を果たす為に、上級程大きな自由度を与える。それは「責任」に対応する「権限」である。最高責任者は最大の自由度を持つが、それは成功の責任を負っている。

責任と権限ある人の任命には、資質と云う可能性～実績と云う条件等を考慮する。最高司令官の必要能力は連隊長や師団長の戦術的能力とは違う。遠い先を透視する能力、優れた問題解決力、人の意表を衝く創造力、断行する決断力、統率力、等が望まれよう。

アレクサンダー大王やナポレオンは自力で成し遂げ

たが、現実には最高能力を持つ人が常にトップである保証はない。組織は常時に機能する必要があるが、「最高の人」ではなくても用は足りる。重要な場合の備えとして、軍事では司令官に参謀を付け、企業では経営者にスタッフ/スタッフ組織を付け、意見の具申や各種提案を義務付ける。これならトップは決定能力だけで良い。決定のクライテリアは全て「還元すればお金」に尽きる。「還元すれば」を付けた理由は、「肉を切らせて骨を切る」戦術から撒餌までであることによる。効果/効用も還元すればお金の評価になる。

トップダウンについて若干を補足する。展開は、(それが難しくとも)「後続作業が簡単になる」ようにする。上流段階であるほど効果の大きい手を使い(逃げや局部処置に陥らず)本格的な処置をする。上級者が問題を大きく解決すれば、下位者は極めて動き易くなる。そして今度は、この人も上長に倣い下が動き易くすべきだ(甘える人も出るから、しかるべき処置をする)。かような展開を続けると、終に近づいたある時点以降、急激に問題が氷解して行く。

大学人には「トップダウン/階層性は反対」が多いが、不完全なトップダウンの故に起る「階層性の誤解」では。

このように上級者程優れた解を出し、以下の人々も夫々努力を続けて行くことで、全組織の総力を最も効率良く配分して開発できる。勿論、成功するには責任者に高い資質を要し、(ある意味で)上級程苦勞が多く、全員を同一目的に向けて統率する為に、大きな努力を要する。それでも世界中の多くの人々がこの努力をする。それはこの道筋には最終的な成功があるからだ。

単純定型的な作業の場合には誰でも出来る必要があるであろう。しかし、重要度の高い仕事は、やれば出来る可能性があることが必須条件で、易しいことは常に必要な条件ではありえない。

皆が夫々努力して最善を尽くす。挑戦する行動を繰返し、良い結果がでればそれを踏襲し、悪い結果が出れば自今はそれを避け、成長を続ける内に優れたリーダー、良い部下達の居る優秀なチームが出来る。

ボトムアップでは、下から伸び上がった結果が要求と合致する可能性は保証されていない。伸上がったところで要求に合致させるムダな労力/コストを要する。これを怠れば、仕上がりは保証できない。ボトムアップが「罷り通る」のは、何かが正常ではない。

画家等が脳裏に設計図を持った上で、部分的な所から描く(実現に移す)場合、あるいは、若い/前線の人々の育成や実地体験を目的とする場合には構わない。

プログラムのレベルから見ると「超上流」に見える。図1の展開を上から下に追うと、外部的→構造的→細部と移り変わる。同様に、効果/効用→その実現メカニズム→その条件、と移り変わる。各時点毎に夫々の技術で対処すれば容易に作業できる。実現手段であるプログラムの視点だけではないことに留意されたい。

発注者側と受注者側の二元論がある。醒めた目も必要ではある。しかし、トップではなくても、自分ならどうするか苦心し考えた提案は、評価される。このように苦勞し積重ねると実力が着く。発注者でなくとも、相手(お客様)の立場に配慮した提案なり行動は、(競争~契約社会でも)それなりに相手に評価され、その積みあげが信頼を築き上げる。Customer satisfactionのひとつ。

3. 2 展開の具体的な方法

効果/効用を最終目的として、階層的かつ定量評価しながら開発する具体的な方法を説明する。効果/効用と投資の関係を図3に示す。横軸は(例えば事務作業の)負荷量で、縦軸は総費用額である。両者の関係は原点を通る直線と仮定する。これは負荷量が増えると、経費が直線的に増す。(これは単純化したモデルに過ぎない。単純化すればメカニズム/仕掛けを理解し易く、粗い近似でも、可視化し理解し話あえる利点がある。)

原点から縦軸に沿って伸びた太線分は、合理化システムの開発経費である。この最上部から右上方に伸びる破線は、システムの開発経費を含めた改善後の総経費を示す。仮に完全に合理化されて残存作業が無く運用経費がゼロなら、水平な直線になる。

破線と実線がある所で交わる。負荷量がこの交点よりも右なら経費は前より減少し、差が合理化の「効果/効用」である。合理化を進めて残存を減らせば破線の勾配は減少して効果/効用は増大する。合理化システムの開発経費は合理化分に応じて増え、効果/効用は減少する。っそこで中間に最適点がある。合理化システムの効果/効用はかような図で表して可視化できる。(これは模式的な1例で、これに限る訳ではない。)

合理化効果が産れる現場を見る。図4は企業内での作業の流れである。図aはハードウェア製造の作業の流れで、左端で部品素材が入力され、作業者が手でユニットを組立しチェックする。以後同様な流れの後、製品は倉庫に入る。(モノと同時に管理用の伝票が流れる場合がある。)図bは、全てを人が行う原始的な銀行の例である。左端のユーザが払出伝票を作成し窓口で渡す。以後はこの伝票の流れに伴い各種の作業が行われる。最後

に右端の台帳確認が行われ、支払うお金とこの出金に関わる伝票が左に流れて行き、最終的にユーザにお金が渡る。

両者は図cのように抽象化できる。この流れは、両端に(モノあるいは情報という)成果物があり、中間は作業/処理を表す角無し長方形が繋がっている。図の例は全て人の作業だから、各作業を処理で置換え、合計員数で減少させれば効果/効用が生じる。(効用の場合には現状より処理時間が早くなる等、評価の視点が違う。)このような流れの図が第2の可視化である。

ハードウェア製造作業での合理化は、・製造用の機械や治具を導入し・作業手順を改善し・安い部品素材を・より少なく使う。事務系(間接)作業でも同様で、人を減らすことに(WPからサーバー迄の)計算機の力を使うが、作業手順の改善等も織込める。製造作業では機械等の設備コストと人にかかる経費のトレードオフであった。事務系(間接)作業では、WP等の汎用的な物を無視し簡略化すると、人の作業のコストと計算機化(ソフトウェア化)のコストのトレードオフになる。

続く第3の作業は、「各作業毎の単位作業当り工数×回数やコスト」と「作業を計算機に乗せる/プログラム化のコストの評価」である。階層構成を取入れ、前記の流れの○は①②③...に展開し、別の○の下位の①②③...等、共通分①を括出し、また○を統合して◎●等に整理する。旨く標準化し組合せると作業は容易になり、また合理化部分を変えて評価する等が遣り易くなる。プログラム化コストは例をコード化すれば推定できる。

初めから詳細精緻である必要は無く、粗い推定や近似から出発し、階層的に展開し詳細化しながら各部分の確度と精度を上げて計画する。計画に当たっては粗い経験ベースの推定から始め、常に目標値と推計値を定量的に比べる。バラつき、確度と精度を意識する。精度は細かく実際に近づければ向上できる。確度はサンプル実験/それに基づくシミュレーション、部分試作と評価試験、先行製作と試用等で上げることができる。バラつきは典型的時点の実測と各種異常事態の頻度と負荷倍率の推定に掛かっている。このようにすれば、結果は誰にも見易く整理し、伝え話しあえる。確認も必要な

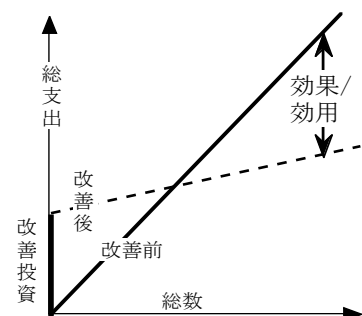


図3 改善投資の効果/効用

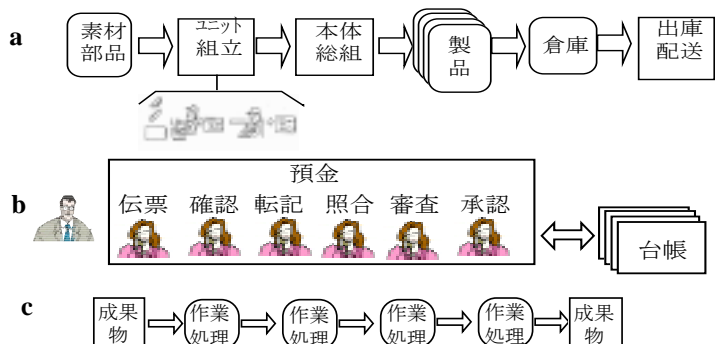


図4 作業の連鎖

所は高精度に、あるいは高確度に調整できる。

ハードで、事前評価が最も厳しいのは数が多い部品で、全て事前に行ない、要すれば先行試作/評価等をする。次はアナログ回路で、利得/周波数特性...と多数項目あり、今では殆どがシミュレーションできて、バラつきや各種の変動にも予め備える。ハード論理では、論理や遅延時間等のシミュレーションで確認する。定量化してメカニズムが簡単明確なら何でも如何様にも事前評価/確認ができる。ソフトウェア/システムも定量化すれば必ず同様に出来る。「作って見なければ判らない」と云わずに、出来ている例に学ぼう。

次に、あるマージンを必ず見込む。「出来上ったら能力70%！を繰返した」なら、140%化して計画する。計画と実際の間には必ず差があり、計画からの外れは必ず起る。チェックポイント毎に一回の外れが起ってもその中で対処可能にし、マージンの無い設計は許さない。監視/アクションが細かく高確度ならマージンは薄くできる。マージン幅は技術の反映と云える。

設計は作ったらピシヤリと合う努力をする！
方法を整備し構造を決める等標準化する。

効果/効用をひねり出す方向の候補がでたら、次には「選択と集中」あるいは「パレートの法則」の応用で、対象の絞込みを徹底する。品質分野では「損失の過半は少数項目から生じる。見方/視点で少数項目の浮かび方が違う。対策はこの影響の大きな少数項目について行え」と教える。これはZipfの法則から生じるので、ヒトの知恵を使う全てに起る。項目の取り方等を変えて項目毎に{(効用/効果)/価格}比を試算し、有力な少数項目を選ぶ。集中が見えたら今度は逆に幅を拡げ、対象部署/機能全体をゼロにする、少なくとも今後は2度と合理化する必要が無い様に奇麗さっぱりさせる。全てを定量化し評価すれば、「選択と集中」が可能になる。

社会には各種の変更があるから、時と共に詳細化し煩雑化し複雑化して、情報システムの開発規模は増える。「一寸待って！」ネット銀行やネット証券の隆盛は、利用者に安価なサービス/大きな利益を与えるからで、それは図4bの中間の各処理を抹消して単純化した結果である。競争力を保つ為に、システム化は、

現作業を計算機に転嫁するのは不可で、
必ず現状を叩き切って簡素化して行う。

通常システム規模に比例して年毎の変更/更新(俗に云う保守)コストが増加する。規模は小さい方が良い。

この地道な経費節減的視点の他に、戦略的視点がある。銀行業務では、投資先選定技術、デリバティブ、金融工学等各種がある。製造関係なら生産管理技術で、トヨタのJust-in-timeで棚残を減らす例、コンビニでは売れ筋の管理やお客の到来に合せた商品配送等である。これらは全て図4cから出発して、モデル化～数式化できる。各種の数学的な最適化やシミュレーションで様相の把握が行え、IT系技術を使った戦略的なシステムが作れる。

米国ではIT系投資が全設備投資の1/2近くに達し、

工場建設等の投資の2倍(朝日新聞、2006年10月31日朝刊13版9面、「点検米国経済」と云う。システム投資はハード設備投資以上に厳しく統制するべきであろう。

社長の立場では、

1. 投下できる資金とその時期、見返りを期待する額、時期、期待する確実度
2. 効果/効用を期待する投資の方向付け
3. 効果/効用の産み出し方のモデル

位から後は、枠や方向を明示して部長クラスに任せられる。体制を作り社長の方向付けにより全体を動かす。

- ・社長直属の経営企画組織を設けて計画を統括する。
- ・作業を2分して発注する。

(SE作業) システム構築方式/プログラム仕様書迄(開発作業) プログラム作成/単体～システムテスト

- ・全てのSE作業が出来る人は居ない。協業が必要。

1. ビジネスコンサルタント

ハードウェア製造作業等で対象の定量的な特性から、合理的に改善投資計画立案に至る過程をリードできる能力があり、システムにも拡幅したい希望のある人

2. 社内の情報系システム/設備等の概要から詳細をSE作業のメンバに説明できる人
3. ポイントとなる技術領域(含む最適化技術)につき方式や技術を推奨/評価できる高度技術者
4. 本システムの統括担当システム会社の担当

以上を経営企画組織の本件担当者が統括し、各種プログラム仕様を作成させ、システム開発作業を統括する。我々は以下のように評されている[1]。

建設業では期限を守らず、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。

この事態を起こす根は、

- ・明確に「仕様」が定義できず作業が明確でない。
- ・顧客からベンダ、依頼元と受託先、両者間の関係がビジネスライクでなく、あるべき姿が不明。

にあると考えて、本報告の提案に至った。

この方法は、系統的かつ可視的に、定量的かつ階層的に計画を固められ、事前確認やマージン確保等もできる。多分、一度やれば、2回目からはもっと旨く行く。ゴールが明解だから仕事も為し易くなり、システムが明確な階層構造になり、協力会社/組織に組織/階層的に明確な責任を負わせて委託できる。

これらはごく当たり前だが、その当たり前がしっかりできれば、確実に良い仕事ができる。

4. 具体的なシステム開発

本報告は、以下を主張する。

変更箇所は、続けて変更が起こり易い。変更が出続け修正し続けると、何が何か不明確になり変更の信頼度が急激に落ち、心理的に荒れ始め皆が短絡的心理に陥る。一旦立止まり、纏めて作業させる。

仕様作成/評価/レビューの後、仕様を凍結して開発作業をする。発注者側/受注者側共に変更は出さず、受付けない。システムテストで先の見通しがつかぬ迄、作業する。外力が全くないから作業は進め易く、自分の中で全体を統一できる。作業にも慣れて能率も向上する。

作業中に各種の不良を修正する。正常と損失の両工数を計測する。修正の損失率は10%を超える可能性がある。従来、これは仕様変更で隠蔽されていた。原因を調査し己の弱点を知り再発を防止する。

システムテストで抽出される不良は、システム設計自体の不良と先行テストでの抽出漏れである。凍結により外乱の憂いなく、自分の中で統一できるから、従来より早くに安定になる。先の見通しがついたら、凍結した変更を織込む。従来よりより効率が必ず良い。

正規作業についてはこれ以上云うことは無い。以後は、技術重視を訴えたい。

2. 1で説明した人の意図的行動は階層展開網状で、図1bの定率階層展開網モデルで近似でき下の関係が得られる。

$$\text{総工数} \propto (\text{規模})^{1.0}$$

これは実績で裏付けられる。図5はBoehmのCOCOMOの実績資料を両対数尺度で再プロットした[12]。プロット群の中央を上記の傾向線が貫き理論通りである。プロットの分布は右上に示した釣鐘状になり、対数正規分布状である。人の作業の特性値は殆ど対数正規分布状になり、理想状態では平均値の1/3倍から3倍の範囲に99.74%が分布する。(図5

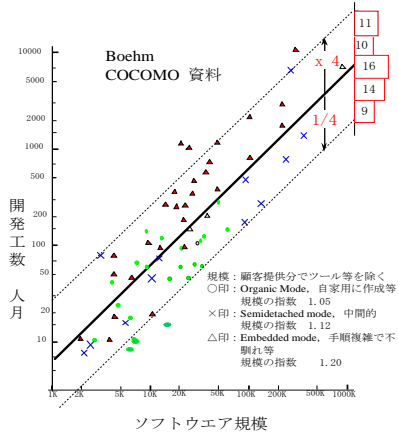


図5 規模と工数の関係 (実績例)

では1/4~4倍の範囲に分布し、理想状態より悪い。)誤り数も同様である。これら知的作業中心のソフトウェアの特性は、ハードウェア直接作業と同じ定量的特性が現れる。ハードウェア製造作業では、これら定量的特性値を使い定量的合理的科学的な生産管理の体系IE[5]を確立した。これは19世紀末から始まり20世紀前半に固まった。日本は第2次大戦の後に取入れ、後進の利で最後には最先端に立った。

終戦後に「労」も加え産労学官が一体になり日本生産性本部を中心として、戦後の復興と技術のキャッチアップを行った[6]。大会社の製造部門には、1960年代初めにIEが標準時間制と共に導入され、以後全産業界に普及した。(この技術の中心の一つは日本能率協会で、多くのコンサルタントが育ち産業界に技術を広めた。)遅れの激しかった品質技術は日本化した後、日本的なTQC (Total Quality Control, 現在ではTQM, Management)に発展する。これはプロセス欠陥除去技術を用いて欠陥を(統計的管理範囲内に留めるのでなく)除去し続けると、品質や歩留まりは上昇し続け、コストは低下し対他社競争力が強化される。この技術的発見は、組織を動かす術と組合わせて普及する。実際に仕事する人自身が、定量的手段を用い(言うなれば科学者になって)欠陥を見出し合理的に改善を工夫する。このプロセス改善に取り組む小集団化活動を社長をトップとした体制が推進する。社長から末端迄が、同一目的に向けて一致して向上努力する。日科技連はこの運動のセンターになった。この活動は1970年代に日本の半導体を世界に広める力になり、続けて電子機器類や自動車等が世界に進出する力になった。運用方法は会社毎に違うが、全員参加/定量的/科学的/合理的な基礎があり、永年向上努力を続け広めることで大きな成果が上がる。トヨタでは改善努力の永年の継続の結果、世界で革新的な生産技術が産れた。向上技術は日本中の各領域に広まり、「高度の定時運行をする鉄道」や「何時行っても必ず欲

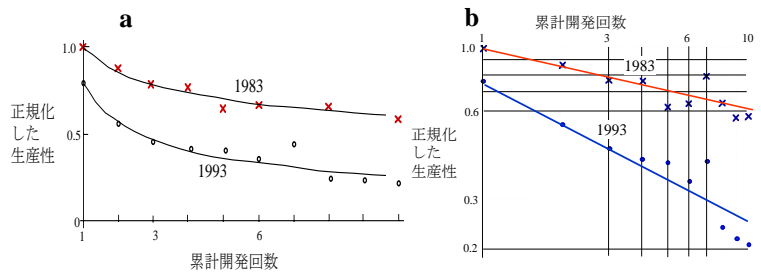


図6 プログラム生産性と習熟効果

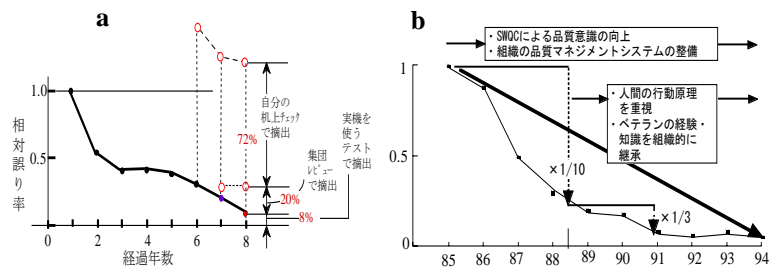


図7 品質の習熟効果

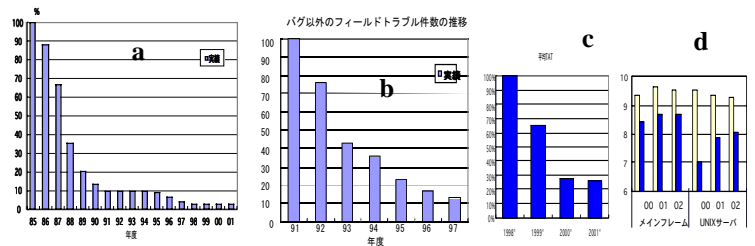


図8 各種の品質特性値の向上

しいモノが買えるコンビニ」等が産れた。これらは世界にも稀な存在であることを知って欲しい。ソフトにIEを本格的に使えば、かならず世界トップになれる!

世界に稀な「日本のソフトウェア工場」[7]がメインフレームに誕生したのもその波及効果である。設計者達も標準時間制や基礎特性の全社教育を受け、ソフトウェア作業に使えることが気付かれ工場制が発足した。

製造は繰返作業で定量計測するから、習熟効果による生産性改善が早くに知られ、習熟性工学[8]が発達した。以下これを中心に説明する。(図6, 7, 8の縦軸は基準点から正規化してある。)図6a[9]は日立の報告例で、新入社員チームのプログラム作業での生産性の習熟効果のグラフである。図は作業の経験回数と正規化工数の関係を示す。両カーブは初めに急激に下がるが、次第にその向上率は下がる。しかし何時までも延び続ける。これはゴルフ等のスポーツやゲーム等の知的作業に現れる習熟効果である。

図6b[4]は(筆者等の責任で)ソフト規模は平均値一定として累計値を求め、両対数尺度で同一資料を表示した。直線傾向線が現れる。この習熟効果は対数習熟効果と云い、人の作業の殆どの特性値に現れる。この直線は「向上努力が大なら勾配は大きく」、「その努力が一定なら直線状の伸び」の特性を示す。

筆者らは図1に示した階層展開を用いるソフトウェア自動設計を実現し[10, 11]、習熟効果を研究した。

対数習熟効果は対数習熟状に知識が集積することにより起ることを見出した。習熟効果による技術向上は知の集積量の増大により起る[10, 11].

図6の2曲線は、新人達の頭脳がツール類、仕掛や作業方法を記憶して生産性が向上した事を示す。また、1983年と1993年の2曲線を比べると、後者の生産性は初回から約20%位向上し向上率もより早い。これは(新入社員とは関係ないから)設備/仕掛の改善と推定できる。工場制とは、設備や仕掛類を設け、一定技量の人なら一応の生産性、品質、期間で生産できる体制を云う。ここの生産技術部門は、定量的に計測/評価し、研究して設備、仕掛け、教育への投資を具申し実現する。(効果/効用と投下資金等を明示して実現するから、Software Engineering Groupとは異なり、経営者に信頼される。)

図7a[13]は日立の通信部門のあるチームの成果、図7b[14]はNECのある部門の品質向上の成果で、共にある領域の中に集中した約10年間の習熟効果を示す。図8はNECのトップから全員の小集団活動も含む品質向上の成果で、図8aは20年間の基本的特性値の習熟効果、また他の図は顧客側に立った各種特性値で向上を展開した時に習熟効果を示す。何れも10~20年の長期間の努力により、各特性値は習熟曲線状の推移で約1/10あるいはそれ以下になった。これは技術の集積である。

品質やバラつき等は永い期間を掛けて知/技術を集積し続けることで低減される

図7bから図8は、第25回ソフトウェアシンポジウムNEC会長 佐々木元氏の基調講演「NECにおける品質革新の取組みから」[14]の引用で、流れを以下にご紹介する。

終戦後の1946年から品質管理を始め、Zero Defect運動など、経営を絡めた品質向上運動を推進した。1960年代以降は前記の全国的な技術革新に乗って全社品質管理体制の構築と強化を図った。1981年以降は『品質を追求すれば生産性は後からついて来る』理念と「お客様が喜んで買ってくれて満足し、更に社会に貢献するソフトウェアの実現」目標を定める。社長をトップとする推進体制を築き、以後20年余りこれをSWQCの名で推進を続けている。これは品質保証を強化する等の縦のトップダウンと共に、ボトムアップの小集団活動を他の柱としている。[14]はNECの流れを的確に纏めてある。

他の特色は、

- ・本体のみでなく系列各社を包括すること、
- ・情報処理/通信システムだからハードとソフトを併用する組込み系システムに昔から取組み続けて来た、ことである。ソフトとハードのシーケンス性を有限状態機械でモデル化し、両者に殆ど同内容の図面類を用いる階層化設計手法を推進して成果を挙げ、日本電気アイシーマイコンシステム(現NECマイクロシステム)は1987年にはデミング賞実施賞を獲得された。この推進の詳細は、全体統括者内丸清社長の著書[15]に詳しい報告がある。階層化設計手法は進化を続け、現在も組込ソフトとLSI開発に適用されている。

上記の各施策のNECの成果が図7bと図8に見える。図8は長期の活動状況を示すので、元資料から説明を引用する。まず図の期間は各5年程度毎に、1985~品質10倍作戦、1992~トラブル低減作戦、1997~CS向上活動と、重点を移し幅を拡げて続けている。図8aは基本特性値フィールドバグで、その削減を掲げて運動した長期実績を示す。習熟曲線は始めは急降下して、遣り甲斐がある。しばらく経つと、何をやっても曲線にはほんの少しの改善しか見えなくなるから、皆が意気阻喪しかねない。実態は各種の技術を広範囲で蓄積しているから、ある期間毎に拡張/進化した次のテーマ(例:1985~トラブル低減作戦等)を掲げ、新しい特性値(図8bフィールドトラブル件数)をも示し皆を力付ける。

前記の期間の初めは・トップダウン型品質向上体制の構築、・SWQC活動によるボトムアップ型品質向上活動の開始である。次は・品質システムの構築・改善、目出度く・ソフト部門で初のISO9001認証を獲得した。以上を経て図dはCS度改善がメインフレームとサーバーの評価値を示し、・出荷後バグの管理からCS向上へ軸足を変え、・サービス品質の向上活動を開始し、・CS調査結果でお客様毎フォローアップ、と進歩の跡を記した。

品質向上を狙うバグ削減という事後のフィードバックから始め、高度化を客観的に認められるレベルに達し、更に「お客様」アタックと創造フェーズと進化して、

攻勢に出はじめた。

「ソフトウェア工場」である2社の公表資料から、日立は生産技術の活動、NECは品質向上の活動を解説した。何れも1960年代末の創世期から既に40年経過している。終始1貫して定量的合理的科学的な歩みを続け、技術を集積して今日に至った。

エンタープライズ系全体で、このような技術の集積がなされているであろうか？前報[2]のように研究費/売上高比率は教育費を含めても1%程度に留まる。プロジェクトが終わった後、個人毎の終了後の評価と改善に向けてのフィードバックすら行われずに、次のプロジェクトに投入される人々が多いと聞く。お客様が変わり、作業方法が変わり、メンバも変わるとも云う。これでは習熟効果は生じないし、技術も集積して行かない。これが以下を引起す諸悪の原因である。

- ・ プロジェクト毎の特性値が大きくばらつく。
→実績資料から有意な結論が引出し難い。
→定量的取組みがなされず、因果関係も不明。
→定量的合理的科学的な基礎～工学が伸びない。
- ・ 自ら能動的に改善するする意欲が沸かない。
→定量的に物事を判断し弁別する力が着かない。
→自分自身で思考せず、他人や有力者に従う。

第1線の人達の切なる願いは「終わったらボーッと休息し切替えたい」こと、と云う。もう一つ、経営者の方々に切望したい。ローテーションや転換教育を除き、専門領域の技術を組織と我々に蓄積させたい。

5. おわりに

本報告は、エンタープライズ系システム開発の解決策であるプロセスを提案した。

1. 良いプロセスに改善する第1歩として、仕様を確定し、凍結して作業し、安定後に更新する。
2. 経営者の期待する効果/効用から始めてトップダウンにかつ定量評価しながらシステムを浮び上らせる。これは従来からあるシステム工学の方法論でもある。
3. 定量的合理的科学的な進め方の数例を紹介した。
4. 乱れた時にはバッサリやる。乱暴ではあるが、実効の上がる策を提案した。(これらは経済産業省で推進して居られる所と狙いは同一と思う。)
5. 我々の業界は「技術の認識と技術の集積が充分なのか」充分にご検討頂きたい。
6. 工学部の情報系教育は、見直す必要がある。ミクロな作る技術のみが必要な訳ではない。

浅学非才な提案ですが、忌憚ないご意見を戴きたい。本年初めからコンサルタント方式や文書の標準化の動きが新聞報道されている。同趣旨と思うのでお役に立てれば幸いです。本提案に関心ある方々と共に、研究し改善するフェーズに入りたく思います。関心ある企業研究所大学等の方々からご連絡頂ければ幸いです。

謝辞

本報告の基礎は筆者等が埼玉大学在職中の研究です。Software Creation Projectの関係者各位のご貢献に感謝します。また引用をご許可下さった日立、著者大野治氏ほか、およびNEC、著者佐々木元氏、ご協力下さった外所和博氏にお礼申し上げます。ご両社のご許可は公開資料の引用許可であり、本報告の責任は筆者等にあります。

参考文献

- [1] 河野善彌, 陳慧, Hassan Abolhassani, ソフトウェア開発工程の在り方, 情処研報, 2006 SE-152(2), pp. 9-16, 2006. 5.
- [2] 河野善彌, 陳慧, Hassan Abolhassani, システム/ソフトウェア開発ビジネスの在り方, 情処研報, 2006-SE-153(6), pp.39-46, 2006.
- [3] 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.
- [4] 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.
- [5] 日本能率協会 IE ハンドブック翻訳委員会, IE ハンドブック, 日本能率協会, 1990. Salvendy, G. eds., Handbook of industrial engineering, John Wiley & Sons, 1982.
- [6] 社会経済生産性本部, 生産性運動50年史, (財団法人) 社会経済生産性本部, 2005.
- [7] 富沢宏之, 藤井留美 訳, 日本のソフトウェア戦略-アメリカ方式経営への挑戦, 三田出版会, 1993. Cusmano, M. A., Japan's software factory: A challenge to U.S. management, Oxford Press, 1991.
- [8] 諸岡孝次, 習熟性工学(改定版), 建帛社, 1982.
- [9] 森岡洋介, 長野文子, 大野治, EAGLE/Pを用いたプログラム開発の習熟, 情処学会全国大会(平成3年後期), 1K-3, 1991.
- [10] Abolhassani, H., 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.
- [11] Koono, Z., Abolhassani, H. and Chen, H., A new way of automatic design of software (Simulating human intentional activity), Fujita, H and Mejri, M. eds., New trends in software methodologies, tools and techniques (Proc. of SOMET 06), pp. 407-420, IOS Press, 2006.
- [12] Koono, Z. Chen, H. and Far, B.H., Expert's knowledge structure explains software engineering, Proc. of Joint Conference on Knowledge-Based Software Engineering 1996, pp. 193-197, 1996.
- [13] Koono, Z., Ashihara K. and Soga M., Structural way of thinking as applied to development, Proc. of IEEE/IEICE Global Telecommunications Conf. 1987, pp. 26. 6. 1-6, Nov. 1987.
- [14] 佐々木元, [基調講演]NECにおける品質革新の取組み, 第25回ソフトウェア品質シンポジウム 発表報文集, pp. 13-39, 日科技連, 2006.
- [15] 倉原文照, 内丸清, 岡本進, 技術集団のTQC, 日科技連, 1990.
- [16] 淡徳三郎(訳), 戦争論, 徳間書房, 1965, Clausewitz, Karl von., Vom Kriege, 1832.