

人の設計知識構造と定量評価 (2/2)

河野 善彌[†], 陳 慧[‡]

[†] 元 埼玉大学 工学部 〒251-0875 藤沢市本藤沢 2-13-5

[‡] 国士舘大学 情報科学センタ 〒154-8515 東京都世田谷区世田谷 4-28-1

E-mail: koono@vesta.ocn.ne.jp, chen@kokushikan.ac.jp

あらまし この報告は、先行報告の結果を承け、ソフトウェア開発の作業を「工程」の概念で説明する。基礎として工程の知識構造を説明する。工程の分割と定量化により問題を割出し、各種の管理とヒトを動かすことで、改善する。

キーワード 工程, 定量管理, 工程改善, プロジェクト管理, TQM

Structure of Human Design Knowledge And The Quantitative Evaluation (Part 2/2)

Zenya Koono[†] and Hui Chen[‡]

[†] Formerly Saitama University, SOHO: Honfujisawa 2-12-5, Fujisawa, 251-0875, Kanagawa, Japan

[‡] Center for Information Science, Kokushikan University, 4-21-1 Setagaya, Setagaya, 154-8515, Japan

E-mail: koono@vesta.ocn.ne.jp and chen@kokushikan.ac.jp

Abstract This paper reports on a software development process from a viewpoint of 'process' following to its previous report. It describes a knowledge structure of a process. Problems may be identified by thus divided processes with their quantitative measurements, and the process may be improved by using management methods including for people.

Keywords Process, Quantitative control, Process improvement, Project management and TQM

1. はじめに

工程や工程表という言葉は、元来は生産管理の用語であったが、今の日本では普通会話でも使われる。これは、品質や時間に世界一厳しい日本では、生産管理が社会に大きな影響を与えていることの現れであろう。この報告は、先行報告を受けて、工程の基礎と適用を報告する。

内容は、定量的で合理的な生産管理の理論モデル化である。人は合目的的に作業するように最適化した知の構造も持っている。それを経験的に取出し集大成して「生産管理」の体系が出来た。この報告では、知の立場から「工程」の構造を説明する。

人は単に「作業する」のではなく、作業しつつより合理的で経済的な仕方を求め改良して行く。作業を抽象化した「工程」から見ると、科学的計測し、定量的に判定して、ヒトの頭脳が改善する。この手続きや作業は全て「生産管理」の教科書類に示されている。

工程 (process) とは、(作業前状態の) 入力人間の作業で変換して出力 (作業後の状態) にすることである。設計や事務作業の工程は、入力情報から出力情報への変換が本質で、取扱の便宜のため文書図面で捕らえる。ハードウェア製造の工程は、モノの変換になる。両者の工程の中核は共に知的なヒトであり、環境や上長、同僚、部下に囲まれて働き、その影響を受ける。但し、このヒトは「人の特性を持つ1種の機構」と突放して客観的に見る。このヒトの特性から「工程」が発する。

この生産管理を初め各種の industrialization, 即ち「工業化」の体系は、19世紀末から形成された。英語では Industrial Engineering, IE と云い、日本語では「経営工学」[1]と云う。日本は先進国に遅れてこれを学び、高度成長経済を支えた。その後、元来は工程の特性を統計的な限界範囲内に留める為の回復手段であった、「改善手法」を限度を超えて適用すると品質がドンドン改善されることが見出された。品質に

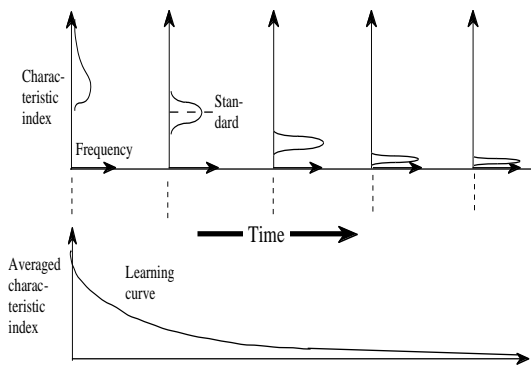


図1 工程特性の長期推移

厳しい日本では、これは非価格競争力を増す手法になる。これは工程を改善するから、適用対象を選ばない。適用を続ける内に遡ってトップも改善に参画し、業績は向上し全員が仕事に生き甲斐を見出す経営革命まで生じた。かくして船舶、半導体、自動車と、日本製品は世界に迎えられるようになった。出所の名を付けてTQCと呼ばれたが、今では Total Quality Management, TQM と改称している。

2 工程の基礎

2.1. 展望

工程は如何に進化するものか？日本でのラジオ製作工程の長期推移を図1の概念図で見よう。図の下部は工程の特性値「ラジオ1台当りの工数、生産性」の長期推移で、習熟曲線になる。上部は各時点毎の生産性の分布である。左端は（計算機の発達初期と同時期の）ST管型真空管の時代で、対数正規分布状を示す。全て人手作業で個別に組立しハンダ付けで布線する。個人毎に大きくバラつき、安定な生産にはほど遠かった。次第に技術が進歩し、生産量も増えた。作業時間の計測が行われ、ベルトコンベヤ作業が導入された。製造作業は、コンベヤの場所毎に単純な繰返し作業の連鎖に展開され、手順通り作業する。バラつきは減り、2番目の図のように正規分布状になった。

現在迄の50年間に、中心部品は真空管はST管からGT管、MT管と小型化し、更にトランジスタに変わったが、これも、IC、LSIと超小型化した。接続も、ハンダ付け、工具を用いる wire wrapping から、印刷基板上の銅箔と進化した。今では作業は自動機械化され、その運用の工数すらもゼロに近い。ラジオは携帯電話に変わり、大規模工場では年間数千万台が生産される。

この間のソフトウェアでの進歩は主としてモノであったから、ヒトについては、今でも上の1~2番目のグラフの中間にあり、労働集約状態に停留している。極端には、

50年経過しても、工業的な進歩は無かった。

複雑な加工を要するメカニズムや人手布線の多い電子装置を、数台作るとは現在でもある。その生産は図1の上部左端と同じで、作業は難波を極め、価格は天文学的になり、ソフトウェアの個別開発と何も違わない。「ハードウェアだから簡単で、ソフトウェアだから複雑」な訳では決

1 詳細設計やコーディング段階の作業は製造とは云えない。これらも製造作業も、ヒトの知の働きが中心である。

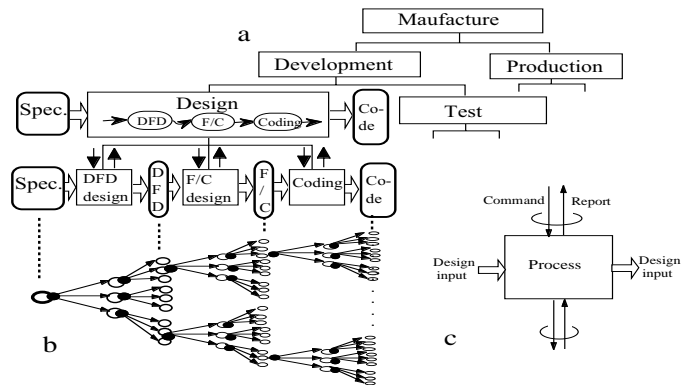


図2 階層的な工程と誤り累計曲線

してなくて、単なる industrialization の違いに過ぎない。ソフトウェアでハードウェアと同様な industrialization が起これば、同様な大飛躍が起こるのであろう。

2.2. 工程の構造

図2.aは製作(manufacture)の工程である。製造(production)は繰返し多量生産に特化した工程を云う。ソフトウェアでの製造¹は製品ファイルのコピーであり、最初から高度に自動化済である。左下に展開を進ると、

・下位になる程、具体化し、明確になり、詳細化する。これを続けると、最終的には微小な作業の単位的な知覚的処理[2]とそれによる微小動作に帰着する。工程とは、「目的の階層性」によるヒトの概念の階層展開である。

設計の実作業は図bの階層展開網で行われる。工程は、これと直交する垂直な階層展開網になる。試みに「設計」を見ると、ここでの作業は下位の「データフロー設計」「フローチャート設計」「コーディング」に展開され、実作業は図bの展開網が行う。図2.cの単位的工程の構成を見る。入力が左から工程に入り、右に出力する。「設計」工程は制御のみを行ない、それは状態遷移状(最小のObject)である。下位を動かす「指令」を出し、「報告」を受けて次に進む。「設計」の上にも同構造があり、

工程は制御中心で、同じ単位構造の繰返し。

制御対象はblack boxで階層性なら何でも良い。

担当設計者は設計作業に没頭するが、中堅以上の設計者や管理者は、工程処理で頭が一杯になる。リーダー/管理者の脳裏では、工程の「作る目的」の階層展開網の上位が頻りに参照され、設計者の脳裏では、「作る目的」の網のより下位の作業方法を、無意識かつ高頻度に参照する。設計は階層展開網を新しく創造することが中心であったが、工程の「作る目的を階層展開した」展開網は、専ら参照され、これに則って仕事する。

参照されるのは工程のみではない。自分の組織の構成や設計対象の機能の構成がある。普通、これらは階層的に構成するが、例えば管理者/リーダーの下のチームの構成は対象の機能構成と一致させ、一部が共用される。

正確で的確な用語で明確に構成された階層展開網は相互につながりあう。これらの記憶、従って理解も容易で、欠損が補われ、強固で正確な記憶になる。人の記憶の85%は意味ネットワークと云うが、これらは巨大な意味ネットワークをなしている。この仕組によって円滑に作業で

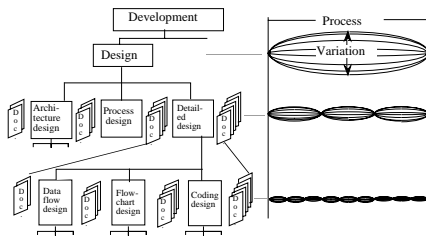


図3 工程の分割による純化

き、大規模で高信頼度なシステムの開発が可能になる。(ちなみに、ハードウェア産業では、図面やコードの体系で、階層的でない構成を許さないシステムになっている。)階層制は誤を減らす巨視的最適化²である。

2.3. 工程の分割

工程は線形で分割・統合できる。その条件は、

- ・階層的な展開であること、
- ・分断点や端面は単純であること、
- ・両端面インタフェースにより工程を定義する。

工程名称は定義ではなく、工程を表す符牒に過ぎない。

工程での変換機能は、中の作業に応じた特性を持つ。

- ・設計は入力を増幅し、誤りは「作り込み誤」になる。
- ・テストは入力の誤りを減衰し、誤りは残留誤になる。
- ・設計もテストも作業規模に応じて資源を消費する。

これらの具体的な様相は、先行報告で説明した。

「工程分割」は工程を純化する。図3 [3] の概念図で、ある工程を分割する。元よりも中間の間仕切りは増え(統制は強化され)、バラつきが減る。この何れかを更に分割して下位で統制すれば、バラつきは更に減る。すなわち、左の工程を階層展開して、下に降るほど、右の弦の振動の図のようにバラつき幅が減る。統制を更に強化するには、端面の文書図面の記述項目や記述度を規定し、基準に従い作業する。更に、「詳細設計」を「データフロー設計」「フローチャート設計」と「コーディング」と、作業を単一化すれば更に純化でき、定義である文書図面の規格や基準を詳細化すれば更に統制は強化できる。

3. 工程の実際

3.1. 原理的な工程とその特性

図4の最下部は、「設計」「机上チェック」と「テスト」からなる簡易な工程である。各工程は前記の各特性を持ち、また3者を纏めた全体工程の各特性も同様である。

ソフトウェア工学で広く知られたCOCOMOの思想は、開発全体の工数/規模等のデータに準拠して、新しいプロジェクトの工数等を計画する[4]。要点は「全体を捕らえる」ことにある。本研究では、COCOMO等の広汎なデータからは工程の線形性の結論を得るに留め、作業の計画は工程毎の自己実績に準拠する。

生産性、品質やテストの有効度には公定価格は無く、自己努力で幾らでも進歩できる。これらの改良手段も、同様で、その効果は場合により違う。客観的でも過去のデータより、主たる営業領域での自己実績に準拠することが

2 反対は「プログラムの最適化は階層性ではない」局所論が多い。本論は大局的最適化を議論しており、根本は人の知の階層性にある。これを否定できるだろうか？

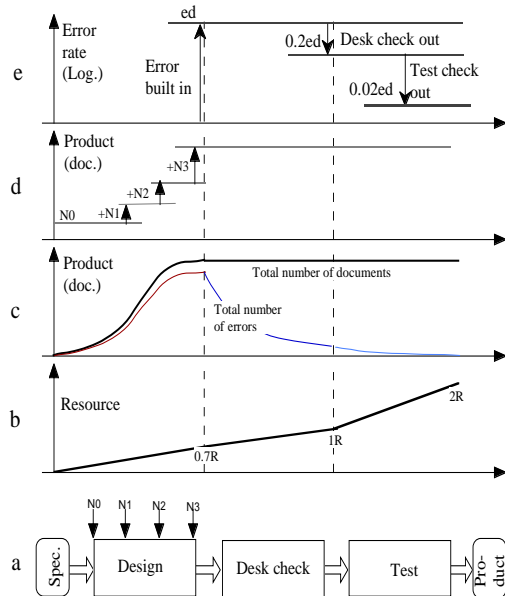


図4 工程の特性値の各種の表示

が、より確度も精度も高く、進歩に容易に適合できる。全体でなくて部分毎に分けて、実績を評価し、これを用いて推定し計画できる。

あるシステムの初開発を図4.aの工程で行ない、実績は図のb~eとする。図から色々な事が判る。例えば、図aの工数累計に追加投入はないし、図bも同様、なので、この開発は順調に進んだことが判る。

・図eは、搬送系のレベル図に準じた残留誤り率のレベル図である。品質が定量的に掴め、最適化も出来る。更に、以下の例などに発展できる。

- ・同一チームの同種実績数例と比較すれば、そのチームのバラつきが掴め、個人評価も出来る。時系列的データから全体工程の習熟効果が測定でき、小工程毎の習熟効果も評価できる。これらで定めた標準値を基準値とした、逆に各個の特性値から作業の様相や状況も判る。
- ・別プロジェクトと相互比較して、特性の差異と原因が判る。
- ・設計を3設計工程に縦列分解した時の夫々の文書量を図dに示す。仕様書(A4換算)枚数 N_0 ~コード行数 N_3 は、階層展開網の性質から $N_0:N_1:N_2:N_3$ は一定になる。実績例から各計数を求めれば、計画に使える。もし N_1 が計画値より小なら、最終規模は当初計画より小さくなる見込。成果物量は N_0 だが、消費工数が計画より大なら、最終的な生産性は計画値より低くなる見込。

任意に工程を組合せ合理的な工程設計が可能これは分割と定量化(Divide and conquer)による定量的合理的科学的な管理³の基礎になる。

3.2. 優れた開発の実績例

図5は、ある開発の総括図[5およびGTE社資料から筆者の責任で復元]で、棒グラフは開発工数の工程百分率、グラフ中の階層図はシステム構成、グラフは、横軸は工程の進行純に累計摘出誤り率(件/K行)の推移を示す。

テスト以前に82.5%の誤りを机上摘出している。(これは多数の従事者のプロジェクトでは最高のレベルである。)

3 従来は、勘と長い経験からより良い工程を求めた。そこで、皆は、現状を保つべく、現状変更を一切を嫌う。

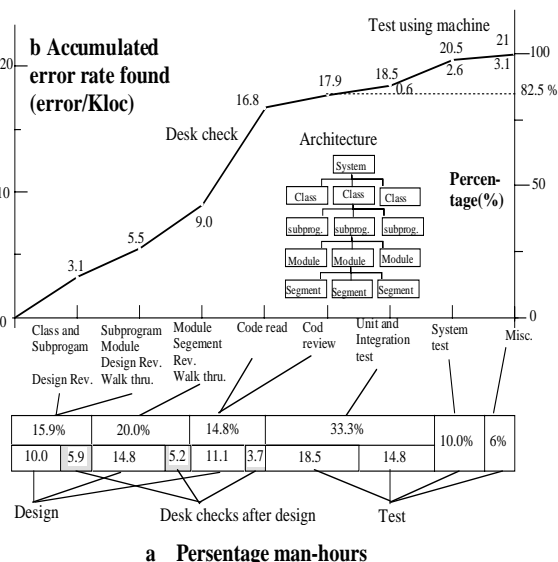


図5 優れた開発の実績例

(作り込み誤り率も21件/行と低い。) その工数(ハッチ部)は、夫々の純設計工数の約1/4~1/3程度に過ぎない。草創期だが構造化設計を行ない文書が豊富であり、これが高品質の鍵である。Design Review (DR)の草創初期だが、原義通りの上流DRの他に、各設計工程毎に作業の特性に応じた机上チェックをする。大問題が起り易い最上流と誤り数が多い最下流の設計には多めの工数を割く等、図から優れた管理が読取れる。

[5]は、ハードウェア技術者並に徹底的に定量的に実地検証する。詳細を知りたく思い、1981年に訪問調査した。技術追及心が非常に強く、合理的な思考である。例えば、[5]は単体テストを計算機実行と机上チェックで並行作業した結果を発表している。実験や結果は全員で検討・討論する。オープンに定量的客観的に議論するから、皆の見解が見事に揃う。チームワークが強いから、実証が済むと、工程を切替える。最高度成熟度チームの一つである。この優れた技術者の雰囲気をソフトウェア工学に齎したい。

4. 工程の計画と管理

4.1. 定量的なプロジェクト計画

定量的計画の原理を図6で簡単に説明する。製品企画から製品計画(要件書から仕様書)の段階で、図aのよう

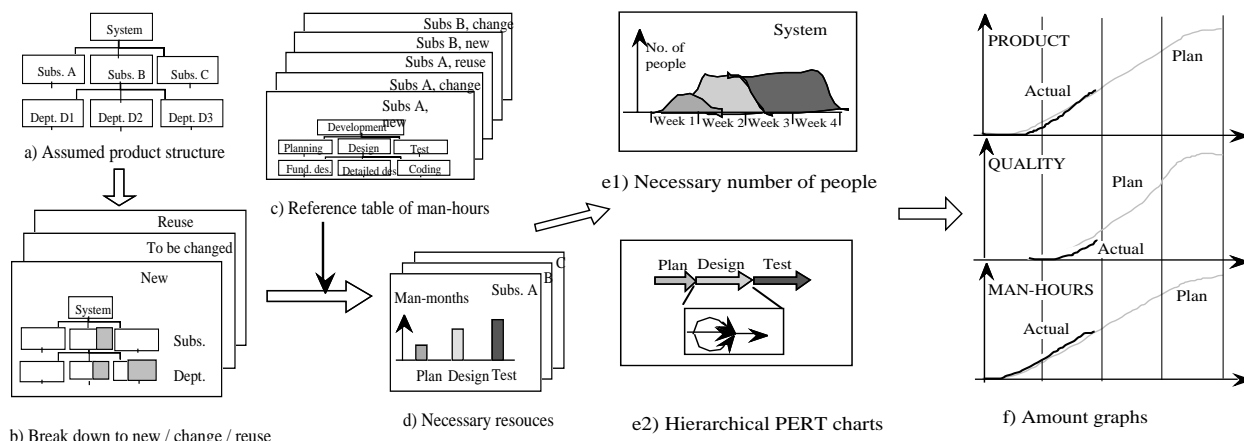


図6 定量的なプロジェクト管理

に実現構造を決め、これの新規/変更/流用等の規模を上位から仕分け(図b)する。既存プログラム群と対照し、図cの構成別工程別/組織別などの基準値表を参照し、新プロジェクトの条件を入れ、標準能力者での作業量を求める。図dは総括図である。次には、構造別の作業順序、利用可能な要員とその能力指標、能力別 derating、ピーク人員数等の拘束条件を加味して、必要なマージンを加え、主要な工程完期日を実現する作業計画の概要から詳細を計算する。図e1はある出力図の例である。生産管理のテキストブック等には、技術の基本が説明されている。

全ての計画は外乱で乱され、(神様が管理しても)ある割合で損失が出る。マージンを加えて計画し、速やかに検知し対策し復旧させる。マージン無しの計画は必ず崩壊する。

計画確定後の日程管理は、図e2のPART(Program Evaluation and Review Technique)を階層的に用いて、プロジェクトの実行を定量的に管理する。図fは監視対象の例で、原理は3.1で説明した。これを階層的に準備して全員のニーズに応じた進捗管理をして、管理者も実態が掴める。計画と実績を対比し、些細でも差異を検知すれば直ちに問題に対処して、素早く原計画に復帰させる。

ハードウェア企業では、ある規模以上なら工務担当を置き、製造と設計は本来業務に専念させる。ソフトウェアでも、進捗点検/異常検知/を含め、代替するシステムが望まれる。

高度な管理体制を初めからは作れない。組織の実情に適合した、簡単な工程、簡易な定義、主要機能のみの段階から出発する。各種改善を積上げて生産性や品質を向上させる。管理も同様で、定量的に各種の研究をしつつ、開発組織と同期して定義や計測から始まる管理方式を改善し、適当な部分をシステム化する。初めから管理のみ強化しても、実態がついていけない。

管理の仕掛とシステムは自組織毎に作上げる。

4.2. 工程の改善

生産性や品質を向上するには、工程AからA'への切替える。これより優れたレベルに組織が進化することだが、工程の中核はヒトで、古い自分を否定する切替は容易でない。全員の参画が重要である。必ず客観的・定量的に行う。工程・対象システム構成・組織等を全て階層的に展開し、目指す指標値で計測して問題箇所を顕現させ、対策要因も定量的に割付けし、計画的な目標管理で向上させる。行動科学・管理的な全手段を動員し、全員をより優れたに育てる人ことである。

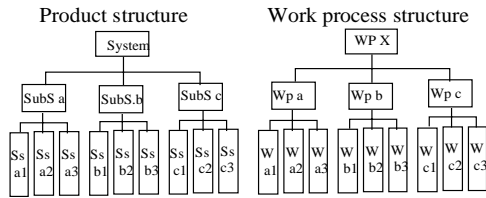


図7 状況を一覧させる図

4.2.1. 生産性の改善

定量評価で認識し、自分達で改善を工夫・提案し、検討/準備し実行して、達成感を味わう。これを積重ねて、積極的に改良改善する設計者に变身させる。プロジェクト終了後、実績資料を整理して全員で検討する。図7は、対象の構造(左側)と工程の構造(右側)の図である。(市場/競争状況や収益計算方式を事前教育して、)今回の損益概況を説明する。各箱に規模や生産性の数値や棒グラフで可視化し、問題を浮び上らせる。

管理者は事前説明の後、「ムダ(見切り発車のロス)、ムリ(無理な日程)、ムラ(追加注文)の撲滅」を、最初に取上ることを提案し、対策を自分が推進すると良い。担当者では処置し難い悩みを解決する、率先垂範が重要。

検討はブレインストーミングで行う。図により問題提起して改善案を募る。他人の案から触発された提案も推奨され、どしどし発言するのが良く、批判は禁止する。書記はメモ用紙に問題や提案の概要を書き、ボードに貼付ける。座長は、時には指名を交え、発言を促す。集中的に行い、最長でも1時間程度で一巡させて止める。別途検討して全員で提案を審議する。各提案の効果を表示し、新規ツール購入や事前確認等の付帯条件も付ける。

これは原価低減方法の1種で、Value Engineering (VE), Value Analysis (VA)と云う。(改善効果/投資)など実効効果で評価してソートし、図8の累計曲線⁴を書く。上位数項目の累計は、総計の過半になる。これらを採用し強く推進して実現させ、提案者や一同に喜びを与える。参画して、課題を発見させ、解決を探し、成功体験を重ねさせて、プログラマーから設計者、新技術を拓く技術者へ变身させる。次第に改善から改良に進み、更に新技術と技術的な先進度が上がり、経営資源の投入量が増す。管理者は賢明に重点化しつつ投入を増やし事業に成功させる。担当者管理者経営者が進化せねば、組織が進化できる訳がない。幹部はそれだけ多く勉強せよ。

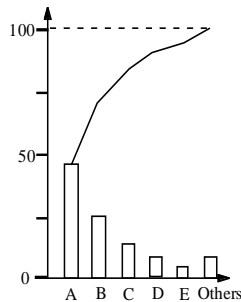


図8 パレート図の概念的な図

4 この種の図をパレート図と云う。これは社会の富の局部集中の法則性を発見したパレートにちなむ。不良分析にも使われる。

4.2.2. 品質の向上

教師が「これはこうせよ、これはするな」と、いくら教えても効き目がない。自らを磨いて自分自身を向上させるしかない。しかし、難しい壁がある。

自分の誤りは公開したくもなく云われたくもない

ある誤りについて、客観的に分析し再発防止策を作る方法を説明する⁵。本人も参加して客観的・科学的方法で話しあう。図9の上部は開発工程を示す。すべては、図下の右端、ユーザの苦情から始まる。

- ①クレームに対応しつつ「何をした時如何なる現象が生じたのか」を掴み、異常現象を再現する。
- ②現象からメカニズムを掴み誤ソースコードを割出す。
- ③設計文書上で工程を遡り、誤り作込み工程を突止める。それは「出力文書は誤りに関わる情報を含むが、入力文書には誤りの痕跡が無い」工程である。
- ④人の「認識」「決定」「変換~詳細化」の何処で誤ったか推定する。階梯毎に誤り対策の方向が大きく違う。
- ⑤最も強く働いた「誤りの動機的原因」を決める。設計入力から出力へのヒトの頭脳内のメカニズム(原因-結果関係)の連鎖中で影響を受けた箇所と影響した要因を決める。
- ⑥「誤りが再度起らない」ようにするには、この原因結果網中のどこかを変える。それが再発防止策になる。単なる抽象的な内容ではなく、具体的な作業の仕方を変える。組織の技術力にする為全員に徹底する。

抽象的な内容でなく体的な作業の仕方を変える。仕掛けが良い。組織の技術力にする為全員に徹底する。

多数の誤りを改善するには、「作り込み誤り率」の工程毎の分布を作る。大多数に適用できる動機的原因からパレート図化等で優先再発防止策を決め実行する。常に下流工程の誤りが多い。小工程/原因別等の適当なグループに細分化し、重点項目を選ぶ。ムダ、ムリ、ムラが解消すると、誤り退治が進む。根気良く原因を潰す内に、氷山が溶けるように小さくなる。途中で必ず伸び悩むから、視点をプログラム中心、論理中心、構造問題、機能問題、製品計画問題と管理者が自分で勉強して誘導せねば駄目。

4.3 サイクル毎の向上

開発-評価-改善実施→次の開発と改善を繰り返す内に、習熟曲線に従い進歩する。伸びは、(知恵~技術×努力~粘り×経営資源×環境)で決まる。改善しないなら何も向上しない。使う手法は目標管理である。

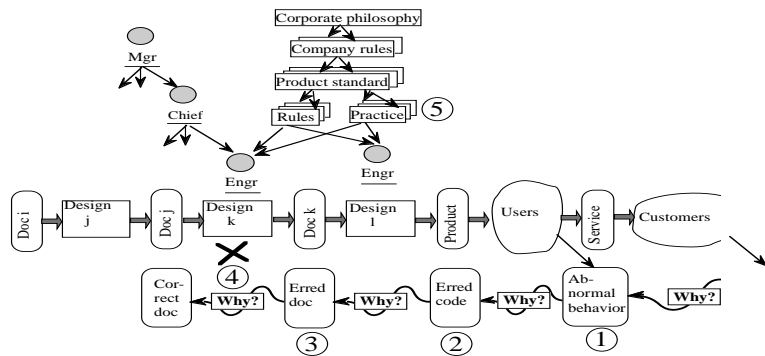


図9 誤りの原因と結果の図

5 TQM (Total Quality Management) での方法を、経験をもとにしてプロセス中心の説明に変えてある[6]。

開発終了後に実績資料から定量的に評価する。対策の原理は、(生産性と品質向上の例のように,)常に定量化して階層展開して問題を顕現させ、原因-結果の網から実効度の高い向上策を選び、目標管理の手法で組織全体を高度化させる。経営は市場と技術動向を見極め、戦略的に改善から改良、新技術/新製品中心に成長しつつ、経営資源の重点集中かつ多量投入に移行し⁴、必勝の態勢にしていく。経営者が、最大の責任を自覚し研鑽し、実態を掌握し、下からの意見を吸上げ、皆を活かし続ける。(今 KAIZEN で有名なトヨタの改善は、40年以上継続の筈)

継続は力である。

5. おわりに

先行報告の定量的な取扱を基礎として、本報告では工程とその技術を説明した。これらには、

- ・工程を分割すること
- ・特性値を定量計測することを中心手段として、
- ・地道に改善を積上げ、
- ・継続させること
- ・ヒト対象を含め、管理には定型的手法があることを明らかにしようと努めた。

おわりに若干を付け加えさせて頂きたい。

急速に市場は世界規模に拡大は良いが、唯一市場しかない状態ともいえ、関連技術の急速な進歩に裏付けられた急激な成長と裏返しにの急速な陳腐化も現れている。19世紀から20世紀初めにハードウェア産業の初期のリーダー(ノーベルもフォード等)が大勝を取めた。丁度100年経った今、ソフトウェア業界も同じ状況にある。他に先駆けて各種の英知を結集して急速に事業を展開して、勝ちを収めている。基礎研究を大学に委ねたい等の主張を通り越して、MOT (Management of Technology)が云われるのはこのような事態の急進への対処であろう。

全てモノヅクリには、productとprocessの両面がある。processには、技能的な能力も必要ではあっても、思考と何らかの創造性は必ず必要である。ソフトウェアでは、これまでの進歩は主にproductに依存してきた。色々と言明したようにprocessの中核はヒトにある。現在でも未知なヒトに対処するには、これを中心に据えねばならない。全ては現実物現場から出発して、経験→経験則と抽象化して行くことが工学では科学的な態度であり、この上に、理論モデル化→新しい工学と使い易くすることと考える。工学の価値は(工学の使命は今未来での貢献であるにせよ)産業に対する貢献で評価される。ソフトウェア工学者は新しいengineerに変身しての登場が期待される。

翻って足下を見ると、世界的な流れに乗り、かつては生命線と言われた各種システムすらアウトソーシングに移行しつつある。10年前には世界に稀な「日本のソフトウェア工場[8]」を産み、高い信頼度を誇った日本の代表的なシステム分野でのトラブルが続出し、優位にあった組込みシステムでも品質低下と開発遅延、などがマスコ

ミに伝えられる。日本のハードウェア産業では、標準時間制の導入は現場のムリムラを退治した。経営側の理解を頂くには、R&Dのコストや期間、残留誤りとその影響等を定量的に説明し理解願ひ、経営の視点からの正しい意思決定と発注の原則を明確にして、悲劇防止が重要である。

最後に、明るい展望を持ちたい。次世代は知の時代と云われる。筆者等はボトムアップに「人に倣ったソフトウェア自動設計」を推進してきた。それは先行および本報告に各種反映させたが、次世代の鍵は「人間知能」にあると思われる。今後は「人間シミュレータ」に発展させて、ささやかでもお役に立てたいと考えている。

6. 謝辞

19世紀末ハードウェア生産の混沌の中で現場から立上がって路を拓いたIEの先達に感謝します。本報告の母胎技術の研究を育てて頂いた埼玉大学情報システム工学科のFar先生(現Calgary大学)、Abolhassaniさん始め研究室の学生諸君、学科の皆様に感謝します。著者河野は、日立製作所(元)戸塚工場生産技術部、検査部、製造部、ソフトウェア関係および幹部の方々から貴重な実地教育を受けた。皆様に深く感謝します。

文献

- [1] Salvendy, G. (eds), Handbook of industrial engineering, John Wiley & Sons, 1982. (訳本) IEハンドブック, 日本能率協会, 1986.
- [2] 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, 一設計工程からの設計知識の獲得と再現一, 人工知能学会誌 Vol. 12, No. 4, pp. 616-626, 1997.
- [3] Koono, Z. and Far, B. H., Quantitativ design of development process, Fourth European Conference on Software Quality, pp. 173-181, 1995.
- [4] Boehm, B. W., Software engineering economics, Prentice Hall, 1981.
- [5] Daly, E. B. and Mnichowicz, D. A., The management of large software development for stored program switching systems, International Switching Symoisium 1979, pp. 1287-1291, 1979.
- [6] Koono, Z., Toiya, M., Matsuida, T. and Soga, M., In-Service quality improvement activities, IEEE COMSOC Journal on Selected Areas in Communications, Vol. 6, No. 8, pp. 1299-1304, 1988.
- [7] JISA 基本統計調査, 情報サービス産業協会, 2003.
- [8] Cusumano, M. A., Japan's software factory: A challenge to U. S. management, Oxford University Press, 1991. (訳) 富沢宏之, 藤井留美, 日本のソフトウェア戦略ーアメリカ志気経営への挑戦, 三田出版会, 1993.
- [9] 河野 善彌, 陳 慧, 人の設計知識と定量評価(1/2), 信学技報, Vol. 103, KBSE, (2004・3)

4 JISA統計では研究開発比率は永年1%程度しかない。2002年度の実績も、教育費用を含めた加重平均値で0.99%設に留まり、今年加わった備備投資率、情報化投資率も夫々1.0%, 1.99%である。改善投資を増やして利益を増大して社員を活性化できないか?