

誤りの作り込みから再発防止迄のメカニズム

河野 善彌[†], 陳 慧[‡]

この報告は、経営段階からソフトウェア設計、更にはハードウェア製造作業、等に共通な知の働きに着目して、誤り関係の諸特性値の定量的な取扱の基礎を提案する報告である。上記作業は「人の意図的行動」であり、人の概念展開連鎖と見做せる。これを用いて、工数と誤り問題が定量的に取扱える。テストはその構成から誤りを定率に減衰する機構であり、同じく定量的に取扱える。終りに工程の概念から誤り再発防止策の立案を説明し、改良過程を知識の集積を行う習熟過程として定量的に取扱えることを示した。

From Build-in of Errors to Mechanism for Providing Preventive Means

Zenya Koono[†] and Hui Chen[‡]

This report proposes a basis for quantitative evaluation of error related problems, ranging from top level management, software design down to human direct work, from view point of human mental operations during them. They are human intentional activities, which may be regarded as a hierarchical decomposition chain of human concept. The idea enables to evaluate both man-hour and error problems. Test is decomposed to its components, and the characteristics may also be evaluated quantitatively. Finally, this report explains the procedure for providing a preventive mean not to repeat the same kind of error, and the improvement process is regarded as feedback system from quantitative evaluation, which may also be evaluated using Learning Effect.

1. はじめに

この報告は、ソフトウェアの誤りから誤りの再発防止までの技術に、モデル化によって定量的に扱い、分析し、評価する技術を確立することを目的としている。

2章では、基本的な知の働きの定量化を説明する。経営段階からシステムやソフトウェア設計、更にはハードウェア製造作業迄の人の知の働きは、ある目的に向けての「人の意図的行動」である。この為の階層展開モデルを用いて、ソフトウェア作業でもハードウェア製造作業でも、誤り数は誤り率と作業規模の積であることが説明できる。経験的な実績から、机上チェックやテストは誤りを定率で減衰させる誤り減衰機構である。

3章では、システムを設計とテストの複合系として扱い、主として開発終了後の全体評価と評価に基づくfeedbackを取上げて、習熟効果を用いて進歩が定量的に計測できることを示した。

この報告は、ハード製造等も含めるように、[河野04a]と[河野05]の該当部分を再編成し、品質関係各位のご意見を頂き易くした。忌憚無いご高見を戴きたい。

[†]奈良先端科学技術大学院大学情報科学研究科(客員)

Graduate School of Information Science, Nara Institute of Science and Technology

[‡]国士舘大学 情報科学センター

Center for Information Science, Kokushikan University

2. 人の意図的行動

2.1 設計の特性

軍事科学分野では「目的の階層性」を軍事行動計画の最高原理とする。これは[Cluasewitz1832]が指摘したもので、一般向けに表現すると次のようになる。

- 最高責任者には「最終目的0」が与えられる。その人は、目的0を実現する手段の群 $0_1 \sim 0_n$ (相互独立になるように機能分割した組織の行動計画)に階層展開して、次位の人々に夫々 $0_1 \sim 0_n$ を課し、実現に必要な資源を与える。
- 各部下は課された 0_i を目的として、実現する手段群 $0_{i1} \sim 0_{in}$ に展開して、自分の部下達の夫々に課し、実現に必要な資源を与える。
- これを繰返すと、最終的に、意図を達成する行動の計画が出来上がり、次にはこの実行に移る。
- ヒトの作業も同様で、意図する行動に向け脳裏で概念が階層展開され、最終的には肉体的動

これは、最上位の「経営」等の意図を、より具体化/詳細化する「設計」等の段階から、最後は具体的な「肉体的作業」の段階に至る迄、全てを一本化する原理を指摘している。これは「人の意図的行動」と見做せよう。本報告はこれら全てを統一的に扱う。

中心となるプログラム設計を更に詳しく見る。図1 [Koono et al. 87] は時計プログラムの設計作業の軌跡である。図の上部は設計の流れで、仕様が入力されるデータフロー設計、以後フローチャート設計、コーディング

を経て最右端のソースコードになる。

このデータフローとは、()をデータ、「」を機能として、(データ)→「機能」→(データ)の流れ図である。図左のデータフロー群は、仕様である外部機能「時計」に入出力の両データを加えて単位データフローである親概念「時計」にする。これを階層展開した第3段は、

(実時間クロック)→「時刻を得る」→(時刻)
 (時刻)→「時刻表示を求める」→(時刻表示)
 (時刻表示)→「表示する」→(時計盤面)

の3単位データフロー(時計の子概念群)で構成された詳細化されたデータフローになる。

第1単位データフローの左上のひしやげた樽形記号から破線が出て、第2,第3単位データフローを経て右上の樽形記号に至る制御の流れはフローチャートで、図の中央左の最上段に記した部分に対応する。

すなわち、単位データフローを概念として、

- ・親概念は子概念に階層展開される、
- ・子概念中の機能の実行順序はフローチャートになる、
- ・階層展開により、明確化、具体化、詳細化される。

図の最下段では、「分針を求める」では、60進制の分の値を6倍すれば針の角度が得られる。他の幅、長さと共に、実現手段であるプログラム言語によるコードに移行する。

・展開されて移行可能になると、実現手段に移行する。

図の例では、上位から下位に1対1に対応する。しかし、更に上位では機械的には対応せず、創造的に階層展開していく。経営やシステム設計の上位ではかような展開が普通である。この場合でも階層展開関係は維持される。

・各階層展開に於て、下記の知的処理が行われる。問題を(1)認知し、幾つかの展開関係を想起して、重みづけすることで何れかを(2)選択/決定して、下位概念に(3)変換/具体化する。

●設計とは人の意図的行動の創る面を表す。それには人

の記憶する概念の展開関係を用いて階層展開を重ねる。その結果、当初の意図は明確化/具体化/詳細化して、次のレベルの実現手段(コンピュータ言語や人の肉体的動作指令群)に移行する。

●誤りの大部分は自然言語/学術~技術用語での思考過程で生じ、実現手段の影響は最終段の変換に限られる。

誤りを一切許さない実現手段(プログラム)の特質から、誤って実現手段での作業に特殊性があると誤認される。難しさは、誤り無しに正しく展開することにある。

この知的処理群を定率展開する階層展開網モデルで近似する。図1の右下に添えた展開網で、白丸は情報、黒丸は単位的知的処理を表す。この網の特性は等比級数の公式により計算できる。展開段数nは充分に大として、単位的知的処理当りに微小時間τが消費されると仮定すれば工数関係が得られる。また、単位的知的処理が小さな確率εで誤ると仮定すれば、誤り関係が得られる。主要な結果は以下になる[Koono 96, 河野 04a].

- ・工数∝(規模)¹,
- ・生産性=工数/(規模)=一定,
- ・誤り数∝(規模)¹,
- ・誤り率=誤り数/(規模)=一定,
- ・これは線形系である,

これらは日常経験的に使われ、またハード製造での定量評価の基本関係でもある下記とも一致している。

総工数は(1個当り工数)×作業数、および

総誤り数は(1個当り誤り率)×作業数

前記の理論推計を実績により検証する[Koono 96, 河野 04a]. 図2の図aは工数に関するCOCOMO資料図[Boehm81]を90°回転させてプロットした。図bは誤り

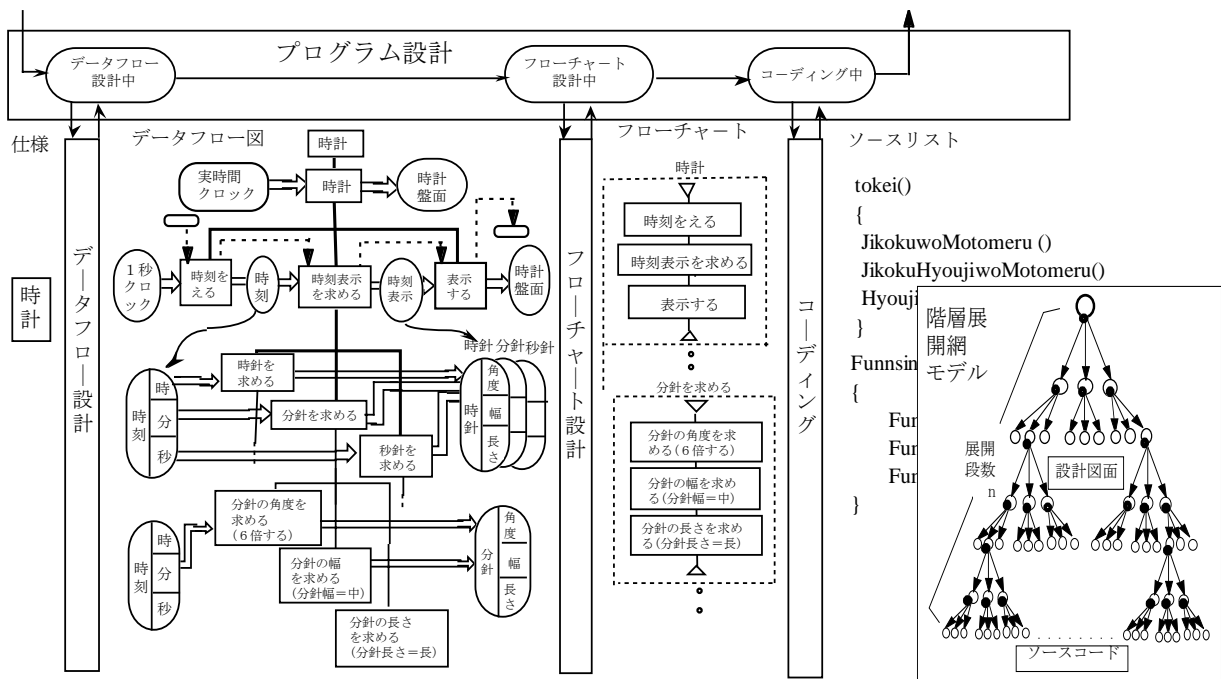
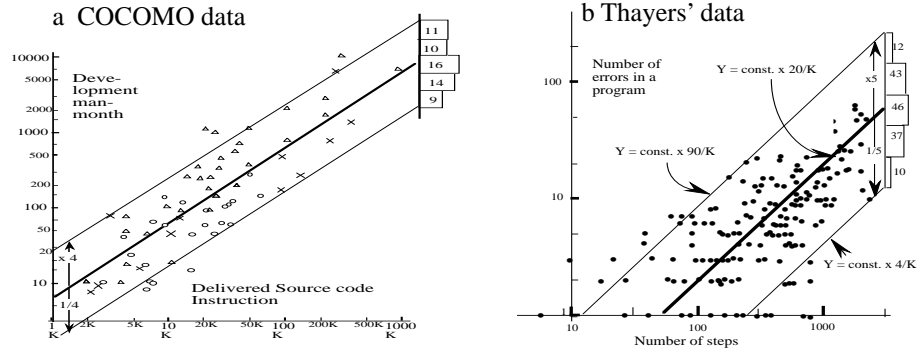


図1 時計プログラムの設計軌跡

図2 開発工数と誤り数の規模特性



に関する Thayers 資料 [Thayers76] の直線表示を読み取り両対数でプロットした。

図の太傾向線はプロット群を貫き、その上下の傾向線を等距離に保ちつつ、(実験データ整理の常道として)上下に少数のプロットを除外する。図は中央傾向線の切片、傾向線間の距離を調節した結果である。

結論として、両図とも理論推計と実績は良く一致する。

・両図の右上に、 $1/N \sim N$ の範囲を5等分した区間中のプロット数を棒グラフで表示した。これらは粗い近似として釣鐘状であり、プロットは対数正規分布である。

①中央傾向線と両側の傾向線との距離 N は、工数では $N=4$ 、誤り数では $N=5$ である。これは(帯状範囲外を除いた)最小:中央:最大は $1/N$ 倍 N : 1 : N 倍であって、工数では $\times 1/4 \sim \times 4$ の範囲、最大最小比は16誤り数では $\times 1/5 \sim \times 5$ の範囲、最大最小比は25である。(誤りの $N > 4$ の理由は後に説明)対数正規分布は下限 $1/N$ から立上がり1で頂点、裾を引いて上限は N 。

②人間信頼性工学(Human Reliability Engineering, HRE)[林84, 塩見96]では、単位的操作をする人の誤り率(Human Error Probability, HEP)を実測値から求める。それは前記同様の倍数の幅を付けて表示され、通常 $N=3$ 、高い信頼度を要するなら $N=5 \sim 10$ を勧告している。

③HRE分野の塩見は、Video Data Terminal, (VDT)の操作について精密に計測して、工数および誤り率は対数正規分布であることを報告している [Shiomi92]。

誤りで $N=5$ と大きいのは、通常の誤り計数法が正しくないことから起こる。

図3は、設計の後、設計側と検査(QA)側の後続2連のテストの様相を示す。ここで設計は、誤りを作り込む純設計、とこれに続く机上チェックに分かつ。純設計では誤りが工数(作業規模)に比例して作り込まれ、以後机上チェッ

ク、設計側テスト、QA側テストと摘出され、残留した誤りが出荷後数ヶ月間に出る様子をグラフで表した。そこで、作込んだ誤り総数は誤り摘出数の総和に等しい。

誤り作り込み数

=机上チェック摘出数+テスト摘出数+出荷後摘出数
最も効果的な摘出は(好まれないが)机上チェックである。Thayers資料や殆どの資料は机上チェック摘出数を計数していない。机上チェックの摘出率は0%から80%の間をバラつくから、これを計数しないと、

$$0 - 0.8 = 0.4 \pm 0.4$$

に見える。そこで、平均値は小さくなり、バラツキは同大に達して、何の因果関係も見えなくなる。Thayers資料は1プロジェクトの各プログラムである為のバラツキの増大もあり、両者で $N=5 > 4$ になっている。

図4右図で誤りのメカニズムを説明する。右の階層展開網の単位的知的処理は確率的に誤る。生じた誤りは下方に降って行き、最後に出力になる。図の左の棒グラフはアセンブラ言語からコンパイラ言語に切替る時期のもので、全プロジェクトの工程区分を、(3)詳細設計からコーディング、(1)仕様から基本が固まる迄、(2)両者の中間、の3区分に分け直し、作り込み誤り率の百分率にして、個別差を消した。奇麗に揃うのは、右の階層展開網で対応する3区分に分断して計数した為である。これは工程の線形性を立証している。

工数や誤りがバラツクのは、工程が実効的にバラツク結果である。バラツキを減らすには、第1に形式面では工程の統制を強化する、第2に工程の実体である知のバラツキを減らせば良い。図5は階層的工程を用いて工程を統制する原理図である。

第2段では、設計の開始端は仕様書、終端ではプログラム

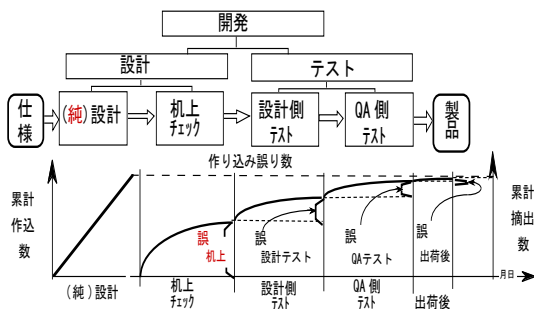


図3 誤り数の規模特性

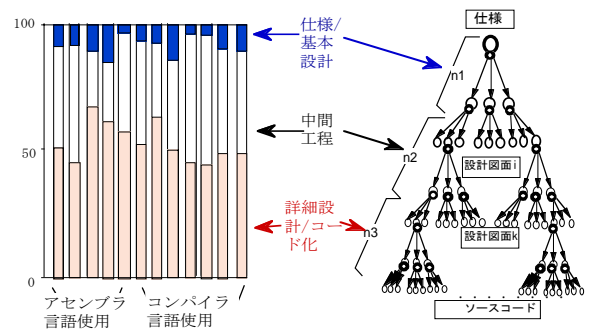


図4 設計の線形性(誤り)

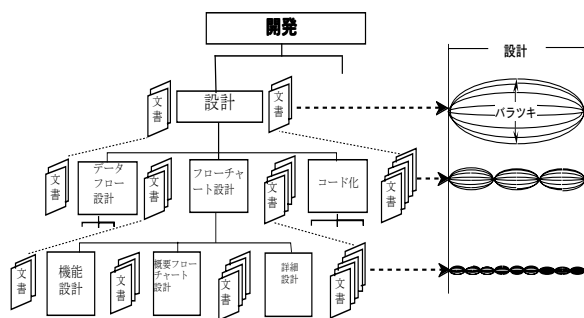


図5 階層的工程による統制の強化

言語で拘束されていた。この時には工程は弦の振動様の図の1/2サイクルになる。これを階層展開して、下位の各工程の端面で統制を加える。例えばデータフロー設計、フローチャート設計、コーディングと手順を押さえればバラツキが減る。右図の弦の振動様の図では、中間の2点を拘束された弦の振動になり、バラツキは更に減る。この階層展開による統制を降して行けば、工程はより厳しく縛られ、特性のバラツキは減る。バラツキは対数正規分布状→幅の広い正規分布状→幅の狭い正規分布状と変る。

- 「設計」結果の良さ、生産性の高さ、誤り率の低さ、これのバラツキの少なさは、「設計」する人々の、問題領域での下記の反映である

「知の優秀性」と「知の仕掛/工程や工法の優秀性」
これら工程の外部特性値は、その人々と問題領域との組合せの値であり、物理/化学等の定数ではない。

2. 2 机上チェック/テストの特性

図3の最下部のグラフは摘出誤りの累計グラフであった。これを上下反転して、作り込み誤り数から始まる誤り減衰グラフを考える。粗く近似すると各部分の累計も、全体累計も、共に $(1 - e^{-ax})$ 状である。これは等比級数的～指数的減衰であるから、個々の小さなテストや机上チェックは小さな誤りの減衰器であり、これの連鎖がテストを構成している[河野93]。

2. 2. 1 机上チェック

机上チェックとは設計終結宣言後テスト開始迄の各種誤り摘出を云い、この摘出誤りを計数する。机上チェックの累計摘出誤り数と摘出日数のグラフはテストの誤り累計曲線と同形状で、 $(1 - e^{-ax})$ 状で近似できる[森本01]。

この全体を一つの「検査」と見做す。対象とする設計の誤り率を E_d とし、この「検査」で不正を正と誤認する第II種の誤り率を E_c とする。「検査」結果には、 $E_d \cdot E_c$ の誤りを含むから、「検査」は元の誤り率 E_d から $E_d \cdot E_c$ に、誤りを E_c 倍に減らす誤り減衰機構になる。

机上チェックを構造的に効率化する方法を説明する。設計対象が複雑ならその机上チェックも一般に複雑で、知的処理数は比例関係にある。今対象設計の全体をM区分し、それに対応する机上チェックもM区分して、その

誤り確率も $1/M$ にする。各区分毎に前記の机上チェックすれば、残留する誤り率は、 $M\{(E_d/M) \cdot (E_c/M)\} = (E_d \cdot E_c)/M$ となり、 $1/M$ 倍に低下する。実際に作業すると、チェックが非常に容易になる(例えば図2の小さな進行段階毎の机上チェックは極めて容易)ので、残留誤り率の低下はチェックし易さの効果に感じられる。

大規模に実行されている例は、数学や物理/化学の計算練習や試験答案の作成法である。小さな進行段階毎に展開結果を1行書かせて、念入りにチェックするよう指導する。これは教師の採点の容易性でなく学生への教育の筈。

2. 2. 2 テスト

テストでの誤り累計曲線は $(1 - e^{-ax})$ 状であるから、テスト数を対数尺度で表せば直線傾向線を示す。図6[Koono87]は電子交換システムでのテスト実績例で、横軸はテスト密度(テスト数/規模)縦軸は残留誤り率(残留する誤り数/規模)である。カーブの勾配は1テスト当りの誤り減衰率で、「テストの有効度」と名付ける。

あるプロジェクトの実績曲線は、 X = 総累計テスト密度、 Y = 最右端下の出荷後に一応安定傾向に入る迄に摘出された残留誤り率)から出発し、次の点は X は最終テスト密度を減じ Y は最終テストで摘出された誤り率を減じた点である。以後同操作を繰返し、最左端では Y 軸の切片になり、これはテスト開始時の残留誤り率である。(正確な分析は出荷後数ヶ月後の誤り数/残留誤り率を要する。)

○テストは誤りの減衰器である。テスト開始時の残留誤り率から出発し、区間毎の「テストの有効度」×テスト密度だけの減衰を重ねて、最後は出荷時の残留誤り率に至る。

実績に準拠すると、出荷品質水準を決めてそれをクリヤするようにテスト密度を決める、即ちテストを定量的に事前に設計することができる。

机上チェックと同様にテストを理論評価できる。1テストを「検査」と見れば、テストは誤り率を E_c 倍に減衰させる。「テストの有効度」を決めるメカニズム、そ

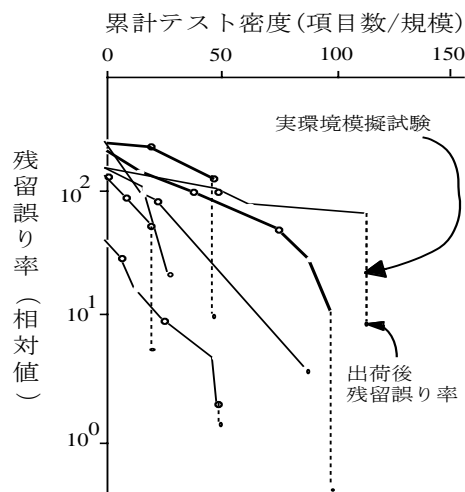


図6 累計テスト密度と残留誤り率

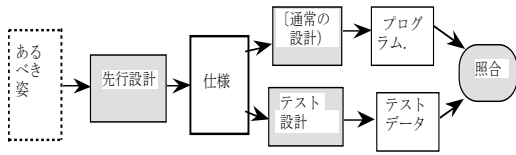


図7 テストの構造

の構造を図7に示す。何が如何に第II種の誤り率に寄与するかは、未詳ではある。しかし、図7の構成要素の誤り率を下げれば「テストの有効度」は改善できる。

世の進歩とともに許容できる残留誤り率は低下するから、テスト密度は年々向上する。1960～70年代には、所謂テストケースの思想から、数十項目/K行であったが、1990年代に高品質組織では500項目/K行に達している。これには、1テストケース内の各種パターンが一つづつ項目に昇格し、テスト条件や正解が明記されてテストの誤り率が低下することが貢献している。

●「テストの有効度」は生産性や誤り率と同様な工程の外部特性であり、人の知と知の作る仕掛けで決まる。この場合には、誤りの定義、設計自体とテスト設計法/テスト用紙記録法/テスト手順等を整備すると進化する。

高信頼度あるいは実時間システムでは、一渡りのテストの後にも、信頼度/処理能力/反応時間等の向上の為のテストが続く。図8[Koono87]はその時期のMTBFの向上を示す。抽出誤りを修復しつつ、適宜に実環境の模擬負荷を掛けて不倒時間(MTBF)を計測しプロットする。残存誤りが少なくなると、対数表示上で直線的に向上する。これを用いれば所望のMTBFに達する時期を予測でき、工夫すれば残存誤り数も予測できる。(誤り累計曲線の全体を信頼度成長曲線と捕えるのは如何なものか疑問。図8の指数的直線領域に入った以降を信頼度成長と呼ぶ方が相応しいのではないのでしょうか?)

この他、検査/QA段階では抜取検査に当たる探針[菅野79]を行い、事前に品質を知る、検査に相応しい品質確認を行ってから検査する等、で有用である。

3. システム

システムのレベルでは、前記各項を纏めた形になる。初めに総括を展望し、開発後の手順を説明する。

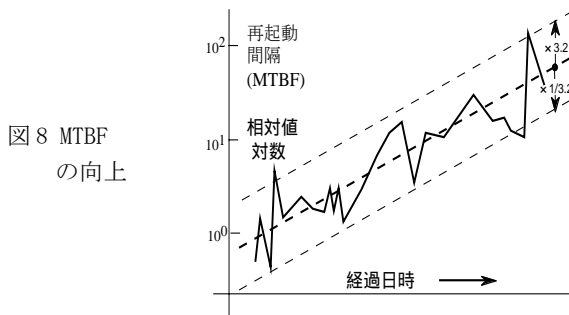


図8 MTBFの向上

3. 1 総括

システムレベルの優れた例を図9[河野05, これはDaly et al. 79およびGTE社内部資料から著者責任で再現した]に示す。このシステムの構成を図9のグラフ右下の階層図に示した。各種統計はこの階層を工程にしている。上のグラフは誤り摘出の累計グラフ、下は工数の内訳を百分率棒グラフを示したものである。

この開発のテスト抽出誤り率は3.1件/K行であり、当時の通常の水準20～60件/K行より約1桁小さい。それは作り込み誤りの82.5%を机上チェックで抽出したことで達成された。工数の棒グラフの各設計の工数の内、机上チェックのハッチ部分とハッチされない純設計部分の工数比を見ると、純設計工数の約1/4～1/2を机上チェックに費やした。約85%の抽出とは、各設計の机上チェックは当該設計で行った作業を厳しく念入りに調べたが、その漏れが15%あった。

各段階のチェック方式は、上のグラフの横軸下部に記してある。最上流では、通常的设计者の能力範囲を超える領域(営業/建設/運用/保守や系構成技術)のエキスパートが製品/設計仕様や基本構造を審査チェックする(原義とおりの)Design Review, 続いて(内部高度技術者中心に)内部構成の動作を確認するWalk Throughと進む。(これまで専ら聞き役であった設計者が次第に中心になり)以後は各構成部分毎に機能を動作/頭脳シミュレーションでのチェックに移行し、最後はコードを読合せで、宣言単位の頭脳シミュレーションをしている。これらで構造化設計の階層毎モジュール毎の文書がチェックされる。

●工程の始端と末端には設計文書がある。工程毎に、その設計結果につき、小さな進行段階毎に正確/明解/疑義無いように記し、机上チェックは工程入力と対照しつつ、小段階毎に、厳密/正確/几帳面にチェックする。完全に抽出した積りでも最終的には80%程度に留まる。

このチームが優れた成果を挙げた大きな理由を説明する。訪問して議論した。経営トップから担当まで優秀な人々である。皆が新技術に明るく見解が揃ってブレがない。中

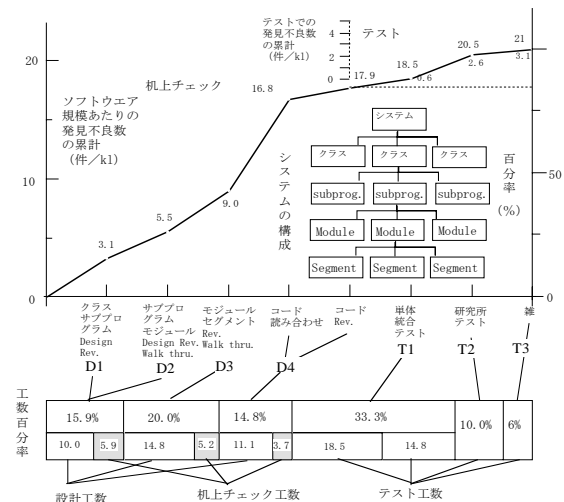
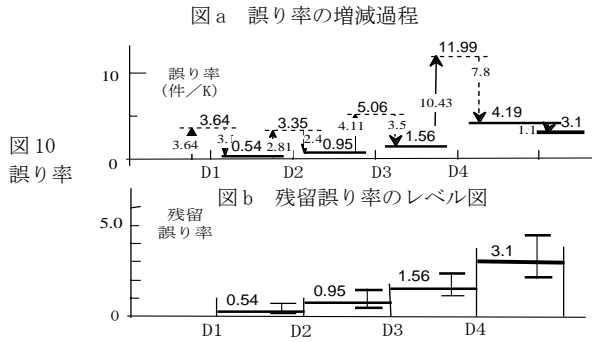


図9 優れた開発例 1970年代末米GTE社PBX交換機



心人物は新技術の吸収に熱心で、調査/検討させて、皆で議論しては実体を定量的に実証させる。(前記論文は、作業方式A/Bを試行して各種効果を定量的に比較研究した実証的報告である。)次回の開発では、優れた方が選ばれ、試行経験から更に改善した方法手順で実地適用する。これは positive feedback に働く。率先垂範して新技術に取組めば、志気も技術も挙がる。公に定量評価すれば皆の意見も揃う。本例を基に誤り率の問題を検討する。(設計工程(D1~D4)の机上チェックの摘出率=全体平均値(85.2%)から作り込み誤り率を近似的に求めて↑で表し、机上摘出誤り率の実績を↓で表して、残留分を求める過程の図aを作る。図bは残留レベルであり、最右端でテスト開始時の残留誤り率3.1件/K行に繋がる。(同様にして、テストのレベル図も作れる。[河野05])

この設計過程をハード製造の例でたどらう。原料の粉末は、混ぜ合され棒状になり、焼成され、棒状に切削加工して研磨され、規定寸法に長さ方向に加工され切断される、と行った1貫的工程と似ている。図bは各工程で不良が累加されるのに相当する。ここで、不良品(オシャカ)が出れば、破壊して調べる。例えば混和物を棒状に伸ばす時に泡が入っている事が判れば、当該工程の加工方法を改善する。小さな問題でも、一つ一つ取上げて工程を改善すれば確実に工程が改善される。

一応の安定に向えば、図bの残留レベルの平均線を中心に上下限を設定し、不良率を限度幅内に収める(統計的な管理)努力をする。ハード製造は繰り返し生産だから、一つ一つこなせば、この状態に近づける。ソフトウェア開発もハード製造も工程の性質も様相も似ている。

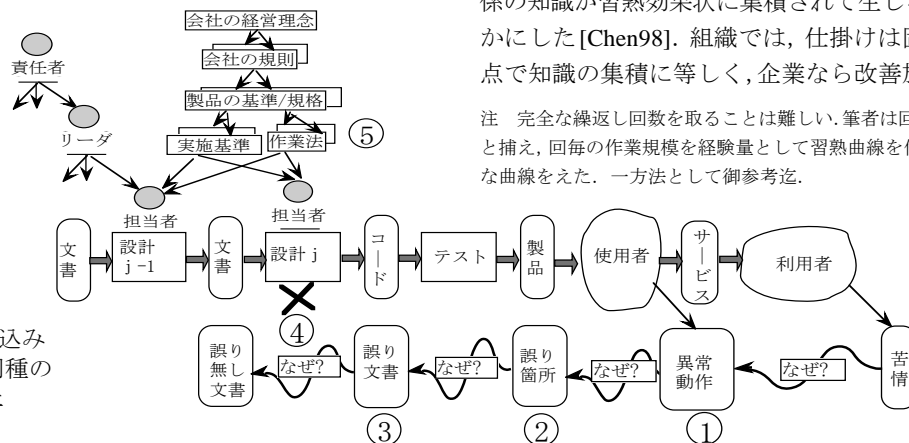


図11 誤り作り込み工程と同種の再発防止

しかし、ソフトウェア開発は、必ずしも繰り返しのではない。ここは大きく違い、標準化の努力をせねばならない。

3. 2事後分析とフィードバック

開発の終了は新製品を世に送出すことで、その製品に生命が息ぶきはじめ育って行く。開発従事者の内、ある人はこの製品と共に暮らし続け、ある人は別の開発へと移って行く。Boehmの指摘しているサイクルの度毎の進歩は、多くの人々が実感する。経営科学の観点からは、Plan-Do-Check-Action, 所謂PDCAが強調される。定量評価を徹底させれば、効率面も品質面も、何が良くて何が悪いか、容易に切分けできる。悪い所が判ればそれを意図して修正する。PDCAは、それを合理的に行わせ、動機付けて歩ませる仕掛けである。(詳細は参考書で研究されたい)

ここでは、これらの進歩を習熟効果で説明する。新しくゲームやスポーツを始めた時、初期には、やる度毎に進歩を実感する。この回毎の進歩は、次第に鈍り始めるが、しかし、依然としてやればやるだけの進歩を感じる。この進歩状況は習熟効果と呼ばれる。図12.a[Tsuda93]はその例で、新入社員チームがプログラム開発作業を繰り返し行った時の相対工数を縦軸に、横軸に経験回数^注を表示してプロットしてある。初めに急激に低下し、やがて緩やかな低下になって行く。これを両対数尺度でプロットすると図12.bの直線傾向線が現れる。これを対数習熟効果[諸岡94, Salvendy82]と云う。図12.b[Tsuda93より作成]から明らかにように、初めの数回の定量的計測結果で得た勾配は将来の特性を予言できる。

習熟効果は、個人、チームのみならず企業や産業のレベルで現れる。部品の将来コストの予測等はその例である。筆者等はハード製造作業とソフト設計作業での誤り作り込み率につき図13の習熟効果を観測した。習熟効果は製造作業者の作業工数の研究から出発し、記憶によると推定されている。筆者等はヒトの設計作業の習熟効果とその理由が、図1に示した階層展開関係の知識が習熟効果状に集積されて生じることを明らかにした[Chen98]。組織では、仕掛けは固定的効果の点で知識の集積に等しく、企業なら改善施設が知識の

注 完全な繰り返し回数を取ることは難しい。筆者は回数は累計経験量と捕え、回毎の作業規模を経験量として習熟曲線を作成し、一応安定な曲線をえた。一方法として御参考迄。

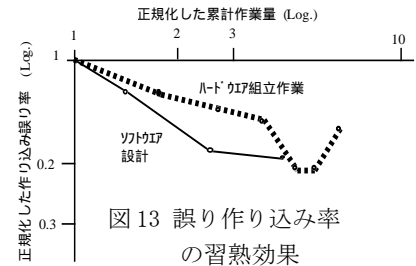
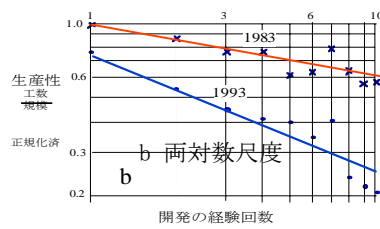
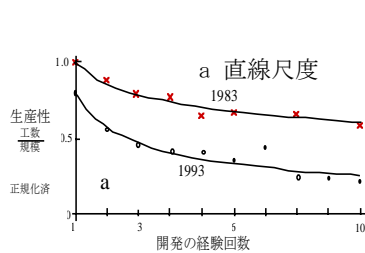


図 12 生産性の習熟効果

図 13 誤り作り込み率の習熟効果

集積として働く。そこで組織や企業/産業のレベルで習熟効果が生じる。習熟効果の勾配は、その改善努力(投資を含む)の大きさとその効果の大きさで定まる。何らかの遅滞や異常負荷等ですぐに曲線の乱れが生じる。

以上から、各組織で行う作業については単にその度の作業ではなく、長期的な向上努力の1貫として、毎回の開発終了時には多面的な評価を行い、その結果からの改善項目を洗い出して実行し、それらが各種の指標値として継続的に記録されトレースできることが望ましい。生産性や誤り率等を考えれば納得できよう。

具体的な改善の行動として、誤りの再発防止を例にとり、図 11 で再発防止手順 [Koono88b] を説明する。

図の中央は工程の流れで、右端では製品は使用者に引渡されサービスの利用者が居る。使用して問題があると、困った利用者の或る比率の人が苦情を申立てる。「何を如何にしたら生じたか?」等を伺うと、異常動作①が判る。再現して突止めると、異常発生メカニズムや誤り箇所②も判り、対策して終わる。これが通常。

・②から設計文書を遡る。当該コードに対応するフローチャートを見ると、ここでも誤っている③。更に遡ると、情報は色々変わるが、何処かの工程(工程内の文書間でも良い)で、その入力には誤りは無いが、出力には誤りかそれを誘発しかねない箇所がある工程④がある。それが誤りを作り込んだ工程になる。

・当該誤った文書を前にして、周辺文書等を調べて設計の全断面で現れるトリオ(認知/決定/変換)の何れを誤ったか(Pとする)本人と話しながら、推定する。

・原因Cと結果Rの因果関係を逆に辿る。認知/決定/変換の誤りPを引起したR, そのRを起すCと遡り、根本的原因⑤を突止める。(何度か辿ることがある。本人は直視したくない事もある。)

・同種の誤りを再度起こさない/起こさせない為に、何か要(かなめ)を変更する。これが再発防止策になる。当該原因と結果の関係を良く調べる。・誤り側Pを変える、

- ・そもそもPを無くす等～基本的に変える、
- ・Pを変える、
- ・Pに影響するRまたはGから変える、

等の各種方策がある。この領域では常識的、洞察的の2面のバランスや大小をよく考えて決める。

・再発防止策を決まったら自分や他人によく伝わり、実行されるようにする。

●誰も自分の失敗等、云いたくもなし、云われたくもない。自分の非を自認してはならない社会、全て誤りは他の原因に帰する文化。色々な問題がある。TQC/Mでは「な

ぜなぜ問答」と云うが、それでは一般人は理解できない。誰でも、オープンに皆と一緒にできるように本モデルを得た。客観化したので、常常仲間て夫々自分の失敗を披露しながら他人の意見を容れる等、日常的に行うことが一つの解決と考える。学生達にもプログラム教育の柱として、自己矯正の方法として教育するべきでは?

1970年代、それまでの造船関係以外に、半導体や自動車等の各種工業製品が世界に進出した。それは品質/品位の良さが大きな力となった。これは、品質に厳しい日本人ユーザ、品質/品位が非価格競争力になる狭い産業、「仏の顔も三度迄」と気軽に詫びて再発防止でお話し願える社会、等の背景と品質保証/品質管理技術者が Total Quality Control/Management の Process Improvement を各企業内部に持ち込んだことが大きい。

●上記等の再発防止は品質向上策と捕えられるが、正確には合理的な工程改良法が相応しい。工程は外部特性を定量的に計り、工程を細分化して局所化するが、作業そのものには入らない。工程改良法は何にでも適用できる。前記の1970年代の日本産業の世界進出を実現した。同じことはソフトでも可能である。

この例は誤りであるけれど、その他の場合にもこの科学的搜索法は利用できる。

改善/改良には成功と失敗の明瞭なパターンがある。出発点は現場の第1線になる。しかし、そのみで出来ることは限られており、コーディングルール作成位で効果は頭打ち、活動は尻つぼみになる。図1の例で示したようにコードだけでは駄目で、真の問題は意味の領域だから、図面化する等の方向に誘導せねばならない。それを卒業しかかったら、設計では「事前に複数案を評価して行う」と云う技術者として当り前の教育を充実し、結果を実測再確認することで、本人達の技術意識を向上させる。システムのレベルに進むと、管理者が先頭に立って仕様凍結を相手に事前約束させ励行させるとか、ハード側でシミュレーション体制を準備して間違ってもソフト側に負担を掛けない仕掛けにする等、ムラ、ムダ、ムリの巨悪退治が必要になる。これらは管理者層が全力を挙げて闘う戦場の筈。ハードではこれと並行して設備投資を推進せねばならない。最後には経営者を革新側に組み込まねばならない。この進歩過程の中で、担当者達には成功感を味わわせて、上位階級者には加担することでシンバを育てる。終始一貫して効果は企業収益に反映させ、次第に成功者の輪を拡げ

る。これらに成功して、造船、半導体、自動車の世界進出が実現できた。改善/改良運動は単に現場にやらせればよいものとは全く違う。それは担当者を初め皆を目的に向かってたくましく戦い勝利する部隊に仕立て上げ、その結果として成果を挙げることに尽きる。

4. おわりに

この報告では、経営段階からシステムやソフトウェア開発、更にはハードウェア製造作業などを貫く「人の意図的行動」に着目し、これが人の概念の階層展開であることから、経験的な要素を含めて、一本化した構造と定量化を可能にした体系を提案するものである。

19世紀末 F. W. Taylor 達は、

- ・総工数は一つ当たり工数×個数である、
- ・工程は線形系である

という経験則に立脚して **Industrial Engineering** を拓き、これはやがて品質関係の諸学問に発展した。本提案は、

- ・人の知の内部構造で上記諸関係が裏付けられる、
- ・ソフトウェア作業は純粋に人の知の作業、

ことを用いて経験則から一歩進めた。

この結果は、知の作業でもハード製造作業でも、工程は同じ外部特性を示すことに着目した、**Cusmano** の云う「日本のソフトウェア工場」の理論基礎でもある。

メインフレーム計算機時代に比べ、現在はコンピュータパワーが湯水の如くに使える、将に夢の時代である。ソフトウェアの人口も格段に増え、多様なビジネスモデルと多様な作業様式の時代に入った。しかしながら、ソフトウェア人から定量的な計測評価の意識が消失した感がある。およそ科学的技術的である為には、定量的な評価、正しい原因と結果の関係の把握が求められると筆者等は信ずる。

謝辞

筆者河野は、際限無しに沸き出すバグと闘い続けた日立製作所戸塚工場の仲間達にお礼申し上げます。特に検査部の方達には苦しい所で助けられた他、種々教えられることが多かった。これらがこの研究を纏める軸になった。歴代工場長はじめ元同僚各位の御教育に感謝します。この研究は埼玉大学情報システム工学科の中で熟成して行き、ここに纏めることができた。諸先生方、学生諸君特に **Software Creation Project** の関係者にお礼申し上げます。

参考文献

[Chen98] Chen H., Tsutsumi N., Takano H., 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, Dec. 1998.

[Cluasewitz1832] Cluasewitz, Karl von, Vom Kriege, 1832. 淡徳三郎(訳), 戦争論, 徳間書房, 1965.

[Boehm81] Boehm, B. W., Software engineering economics, Prentice Hall, (1981).

[Daly et al. 79] Daly, E. B. and Mnichowicz, D. A., The management of large software development for stored program switching systems, International Switching Symposium 1979, pp. 1287-1291, (1979).

[林84] 林喜男, 人間信頼性工学—人間エラーの防止技術—, 海文堂, (1984).

[菅野79] 菅野文友, ソフトウェア・エンジニアリング—ソフトウェアの開発・生産と品質保証, 日科技連, 1979.

[Koono87] Koono, Z., Ashihara, K. and Soga, M., Structural way of thinking as applied to development, IEEE COMSOC Global Telecommunications Conference 1987, pp. 26.6.1-6, (1987).

[Koono88a] Koono, Z., Yamato, M. and Soga, M., Structural way of thinking as applied to high quality design, IEEE COMSOC International Conference on Communications 1988, pp. 8.2.1-7, (1988).

[Koono88b] Koono, Z., Toiya, M., Matsuida, T. and Soga, M., In-Service quality improve, emt activities, IEEE COMSOC Jour. Selected Areas in Commun. Vol. 6, No. 8. pp. 1299-1304, Oct. 1988.

[Koono 90] Koono, Z., Tsuji, H. and Soga, M., Structural way of thinking as applied to productivity, IEEE COMSOC International Conference on Communications 1990, pp. 204.2.1-7, (1990).

[河野93] 河野善彌, 大坪東光, ソフトウェアの誤りと除去の評価 (Industrial Software Engineering の立場から), 情処研報, ソフトウェア工学 95-5, pp. 31-38, 1993.

[Koono 96] Koono Z., Chen H. and Far B. H, Expert's Knowledge Structure Explains Software Engineering, Joint Conference on Knowledge Based Software Engineering 1996, pp. 193-197, (1996).

[森本01] 森本祥一, 須藤紀久夫, 陳慧, 河野善彌, ソフトウェアクリエーション: 統合的CASEツールの画面処理, 信学技報, KBSE, Vol. 80, 2001-3, pp.25-28, 2001.

[河野他04a] 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)

[河野他04b] 河野善彌, 陳慧, 人の設計知識と定量評価 (2/2), 信学技報, Vol. 103, KBSE, (2004・3)

[河野他05] 河野善彌, 陳慧, ソフトウェアプロセス定量化モデルの提案, 情処学会研報, 2004-SE-147 (12), 2005/3

[諸岡94] 諸岡孝次, 習熟性工学(改訂版), 建帛社 1994.

[Salvendy82] Salvendy, G. eds., Handbook of Industrial Engineering, John Wiley and Sons. 1982. 訳, IEハンドブック, 日本能率協会, 1990.

[塩見96] 塩見弘, 人間信頼性工学入門, 日科技連, (1996).

[Shiomi92] Shiomi, H., On analysis and summarization of human reliability data for simple VDT operation., BICRMS 92, pp. 372-377, (1992).

[Thayers76] Thayers, T. A., et al., Software reliability study, Final Technical Report, RADC-TR-76-238, Rome Air Development Center, (1976).

[Tsuda76] Tsuda, J. et al., Productivity analysis of software development with an integrated CASE tool, ICSE 1993.