

Process と Product の定量評価と利用の方法

河野善彌, 陳慧[†], Hassan Abolhassani[‡]

筆者等は、先に経験工学的にソフトウェア開発過程（工程）の定量化モデルを報告した。この報告はこの定量化モデルが「人の意図的行動の知識モデル」を用いれば更に良く説明できることを報告する。2章は全て既報の経験工学レベルである。既報告に更に「工程が無統制な時には、特性は対数正規分布状になる」ことを補足し、更に工程を繰り返す時の特性向上の経験則である習熟効果を説明する。3章は人の知的処理のモデルは身体的行動から知的作業迄を含む「人の意図的行動」を表わせること、工程の外部特性は知的処理のモデルで説明でき、工程を繰り返す時の習熟効果は記憶を付加した知的処理のモデルで説明できる。4章は既報の工程の改善を知の集積で説明する。

Quantitative Evaluation of Process and Product and Their Use

Zenya Koono, Hui Chen[†] and Hassan Abolhassani[‡]

Authors reported previously quantitative model of development process, based on empirical engineering way. This report intends to explain the model better by using knowledge model for human intentional activities. Chapter 2 is empirical engineering way. It adds that when a work process is not constrained well, it shows lognormal characteristics. It also explains learning effect that appears in repeating the same work process. Chapter 3 discusses that 'human intentional activities' covers wide range human activities, from top managements works down to direct workers activities, and that they may be achieved by the same intelligent structure. It explains external characteristics of a work process, and by adding memories; it also explains the learning effect.

1. はじめに

ソフトウェアには、application, product および process の視点がある。現在は application が中心に発展しているが、作る観点からは process が重要である。筆者等は先に process について経験工学的に定量的に扱う為の定量化モデルを報告した[15]。この報告はこの先行報告についての付加分である。

2章の前半は、先の経験工学的な報告に加えて「工程の統制が弱い時には、各種特性が対数正規分布になる」ことを説明する。この対数正規分布は、数多くの要因が夫々積の形で作用して生じる。この為に、実験や計測には相応な配慮が必要である。後半には、3章で対比説明する為、余り知られていない習熟効果の概要を説明する。これも経験則である。

3章は人の知の視点を用いて、工程を別の視点から捕える。筆者等はプログラムの自動生成の研究を行ってきた。それは構造化設計等に見られる概念の階層展開を中心として研究を始めて、最終的に各種の知的処理を総合した構造が理解できた。最上流の経営段階では人の概念的な階層展開、降ってプログラムレベルではデータフロー中心の階層展開、最後の身体レベルでは階層展開の最終段で身体的な動作指令を出力すると考える。

人の脳裏では前記の全ては記号化されて、同一の階層展開機構が働く。これに技能、ルールおよび知識の3レベルの記憶を備えると、かなりヒトの知に近づく。外部的経験的に知られた習熟効果は、上記3種の記憶に蓄積され、高度な作業での習熟効果等も説明できる。

4章では、知の視点から見て最も大きな蓄積を作る「自らを改善し向上させる」過程が、定量的な改善の継続でできることを、改善手法と高度な改善努力を行った例で説明する。

奈良先端科学技術大学院大学 koono@vesta.ocn.ne.jp

[†] 国士館大学 chen@kokushika.ac.jp

[‡] Sharif University of Technology Abolhassani@sharif.edu

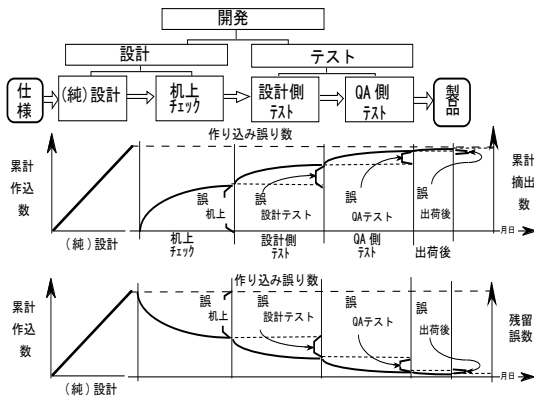


図1 ソフトウェア開発工程

2. 工程 (work process) とその特性

2. 1 基本特性

これは既報告[15]の延長なので、バラツキを重点に報告する。初めに図1[3]で用語を定義する。中段の仕様から製品への流れが工程 (work process) で、階層的に下位に展開する。工程は入力から出力への情報変換過程で、中核の人を隠蔽し客観的に扱う管理手段である。

工程の基本

1. 工程の名称は呼称に過ぎず、工程の定義は入力側文書規約 (例:フローチャート作成基準) と出力側文書規約 (言語文法) により行う。(文章で工程を定義すれば、下位概念を使うから定義にならない。)
2. 管理が目的だから、被制御対象内に入らず外部的特性 (例:工数) を計測して制御する。

Work Breakdown Structure (WBS)は工程の上位の説明～概要に当たる。

図2は前記の工程順に並べた外部特性の実績グラフ群で大きなバラツキを持つ。図2. a～dは両対数尺度で表示してあり、これら4グラフの実績プロットは帯状にバラついている。プロット群中央を中央傾向線が貫き、上方と下方に中央傾向線から等距離で並行な傾向線を引き、少数のプロットを異常値と見做して帯状領域外に出した。図2. a[4]とb[4]の右上には、帯状の分布を棒グラフで示す。その分布は粗く近似して釣鐘状であるから、図のプロットは対数正規分布である。図2.cとdも同様な帯状だから、これも対数正規分布である。

大規模プラントの信頼度評価を行う人間信頼性工学では、単位操作の誤り率と工数は共に対数正規分布であることが証明されている[5]。

対数正規分布は、多くの独立要因が相乗積として働く時に現れる。その標準偏差 σ は平均値に等しく、中央値のN=3倍から1/3倍の範囲内に99.74%のプロットが収まるとされる。図の傾向線の $\times N(\times 1/N)$ は上方(下方)の傾向線がY軸上で中央傾向線のN倍上(1/N倍下)を示す。図2.d[7]はN=3で最も対数正規分布に近い。

このグラフは1種の組込みシステム(電話交換制御用システム)の出荷後の欠陥の累計曲線である。図は両対

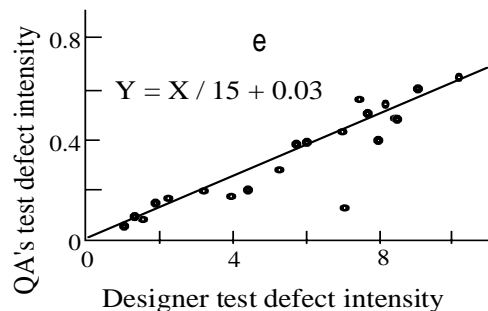
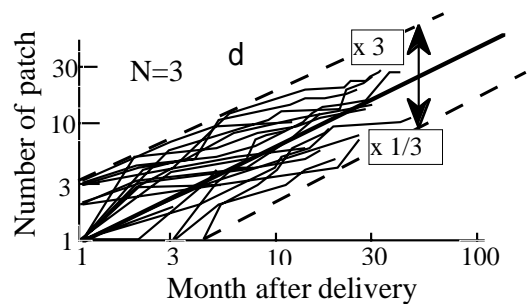
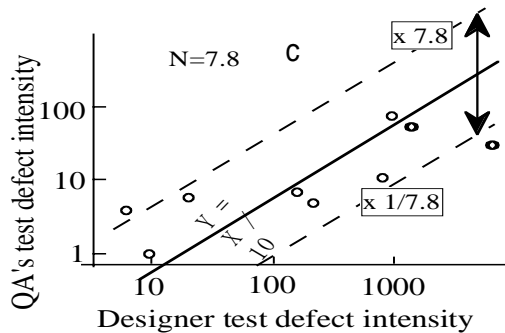
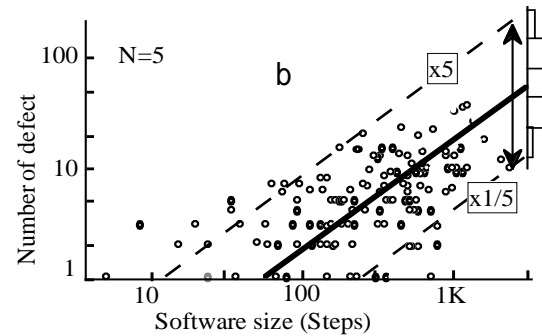
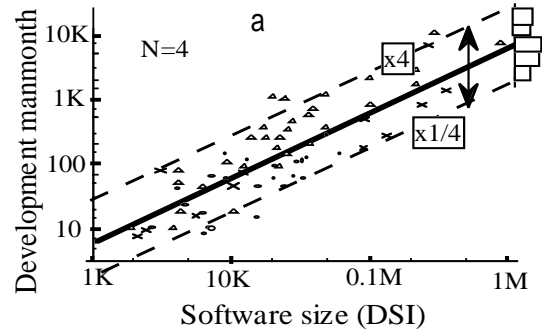


図2 各工程の様相

数尺度であり、サービス開始以後の日数を横軸に、縦軸に累計欠陥数を取り、各カーブはサイトに対応する。残留欠陥率は負の指数的に減衰するから、図の全体の傾向は中央傾向線は直線状で、カーブは中央値の×1/3倍～×3倍の範囲を変動する。N=3で理想的な対数正規分布に近いのは、人為的要因が関係しない為であろう。

遡って図2.c[8]は先行テスト(設計者)抽出欠陥数対後続テスト(検査, Quality Assurance, QA)の両抽出欠陥数の相関を示す。(各1連のテストは、1テスト毎の欠陥の減衰器^{脚注1}の縦列接続で、ある減衰量を持つ。)横軸は先行する設計側テストでの抽出欠陥数、縦軸は後続するQA側テストでの抽出数、各プロットはプロジェクトに対応する。大きなバラツキがある。両者には相関があり、先行テストは欠陥数を約1/10に減衰させる、と理解できる。N=7.8と4図中で最大である。

図2.cの相関の発見に基づき、1980年当時に最新の資料で精密な調査が行われた。その結果は図2.e[9]で、バラツキはごく少ない。この差異は工程の差にある。図2.cは初期の試作品等雑多な工程の、各種多様なシステムの資料で、欠陥密度もテストもバラツキが大きい。(対数正規分布状のテスト入力と対数正規分布状のテストの欠陥減衰量の両者の影響を受け、バラツキが増す)図2.eは、同一部内で同一規格で純化された工程である。以上から、純化され均質な工程であるからバラツキが減っている、と解する事ができる。

遡って設計の特性について、工数を図2.a[15]、欠陥数のを図2.b[15]に示す。(広範囲な種類で小規模から大規模迄の多数の資料で、設計方法が均質で再利用が少ない実績を求めた。)原資料な工数はCOCOMO[10]、欠陥数はThayersの第3プロジェクト[11]である。

図2.aはN=4で、理想系からやや外れている。図2.bの欠陥数はN=5と工数の場合より大きい、右上の帯状領域内の分布は図2.aより整然としている。この原因は、不適当な計数法の為に0.4±0.4のバラツキが重畳され、かつ資料が1システムのプログラム単位であることに起因する[15]。以上から下の結論を得る。

- ・各外部特性は中央傾向線の式である特性が、工程の実効的なバラツキの為にバラつく。
- ・各外部特性値を通じて特性値 \propto (規模)¹であり、工程は線形系で展開、統合が可能。
- ・両軸と中央傾向線の式が示す外部特性が、各工程の基本的特性であり、その値とそのバラツキは工程に依存する。
- ・線形性以外は、工程に依存するから、工程には物理・化学等のような定数は存在しない。

端面を保てば、工程は入替え可能で、自由に統合し、展開できる。各下位の外部特性値を統合すれば、上位の外部特性値が得られ、実績外部特性値にバラツキを考慮すれば、新規プロジェクトの計画に使える。

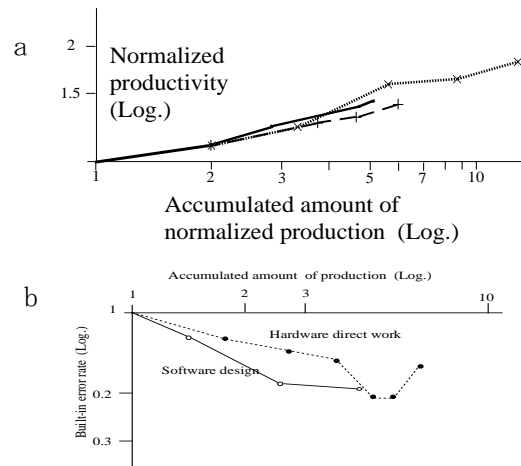


図3 習熟曲線

2. 2工程の習熟効果

あるスポーツやゲームを始めると、初めは急激に向上するが、その勾配は次第に低下する所謂「習熟効果」を示す。1936年に人の作業時間から飛行機工場の生産性に至る迄適用できる習熟効果の研究結果が公表され、研究成果はIndustrial Engineering中の習熟性工学[12]に集約されている。経験則^{脚注2}として広く使われている。人の習熟効果は対数習熟効果と呼ばれる。これは、初回の作業時間をKとして、X回目の作業時間Yは次式で表せる。

$$Y = KX^{-A}$$

但し、Xは繰返し回数、Aは習熟効果の指数。

これを使えば、少数回の実績傾向から将来が予測できる。ハードウェア作業者の標準作業時間予測、半導体の将来価格予想等は周知の応用例である。

筆者等はハードウェア作業のみでなくソフトウェア作業にも現れること、作業時間のみでなく欠陥率にも現れることを確認した。図3.a[13]は各種ソフトウェアの生産性の習熟効果を示し、図の横軸は、作業の繰返し回数として、正規化した開発規模を用い、縦軸は正規化した生産性である。図3.b[14]は、ソフトウェアの作り込み欠陥率とハードウェア作業の作り込み欠陥率を図3.aと同じく累計生産量で正規化した。両者共に直線状の傾向線が現れる。

同じ改善効果を継続すれば曲線は直線状になる。その勾配は改善効果に依存し、改善しないなら勾配は0になる。即ち、改善しないなら進歩は起こらない。

脚注1 1テストを「検査」と見ると、「検査」も誤りを免れない。正を邪とする第I種の誤りと、邪を正とする第II種の誤りがある。前者ではテスト後のチェック過程で誤りは解消する。後者では(第II種誤り確率)で欠陥を見逃す。テストの(第II種の確率)を E_c とすれば、出力の欠陥率は E_c 倍に減衰される。

脚注2 ハードウェア製造作業者の作業を中断すると、習熟効果は一旦は消えるが、速やかに復旧できる。そこで、習熟は人の記憶によると推定された。解明結果は3章で説明する。

以上の結果を纏める。

- ・習熟効果は工程の線形性と同様な基本特性で、その勾配(習熟効果の指数)は工程の繰返し、具体的には経営と管理の外部特性値である。
- ・習熟効果はヒトの知の記憶と推定されてきた。工場の仕掛けや設備は有形の記憶として働く。
- ・習熟効果は記憶の蓄積による改善と進歩の構造であり、習熟曲線はその良い指標になる。

3. 人の意図的行動とその特性

この章は、既報[15]を2章で拡張したものを、ヒトの知の働きで説明しようとするものです。

最初のソフトウェア設計法の提案は、段階的詳細化と云われる。構造化設計を学ぶと、「構造」と云う語が各所に現れる。図4[4]のプログラム設計の理想モデルで基本的な構造を説明する[4, 5, 15, 24]。左端のデータフロー図の最上段は、仕様である「時計」、第2段は、これに入出力データを付加して「単位データフロー」にする。これが最初の親概念になる。第3段で、この親概念を「時刻をえる」「時刻表示を求める」「表示する」の3単位データフローである子概念に展開する^{脚注3}。この段の左上、ひしゃげたピヤ樽状の開始記号から出発した制御の流れは、前記3機能を経て、右上の終了記号で終わる。図の中央のフローチャートはこれを示す。(これはプログラム段階での特殊性であり、以下の説明では省略する。)

第4段は、上記1単位データフローから3本の単位データフローに(Jackson法だから)並列に階層展開する。第5段も同様に階層展開する。その結果の「分針の角度を求める」ことは「60進制の分の値を6倍すれば分針の角度になる」ことが、透けて見えるから、次にはこれらを実現手段であるソースコード表記に置換する。

- ・設計とは親概念から子概念への階層展開の繰返しである。展開するにつれ、より明確、より具

脚注3 MyersのSTS分割。入力(出力)から最も遠く入力(出力)の性質を失わない抽象度が最大の最大抽象点で分断する。

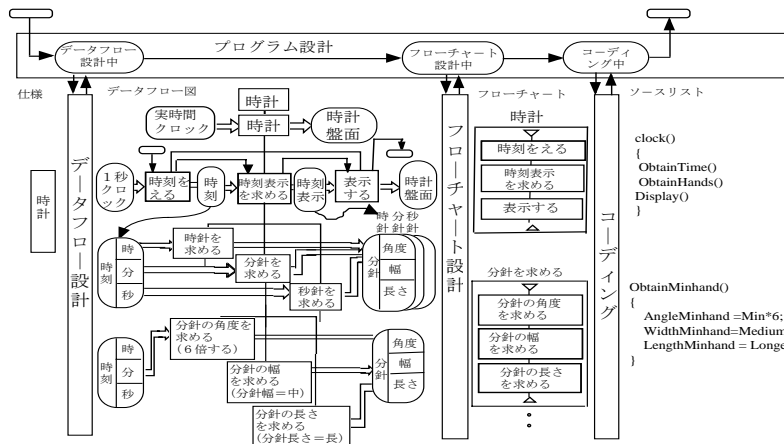


図4 時計プログラムの設計

体的に、より詳細になる。

- ・概念が実現手段で表記できるなら、移行する。人の脳裏では、各情報は記号化されて概念の親子関係を用いて展開処理される。更に上流の概念レベルおよび下流での身体的動作でも同一構造で処理される。

上流の例は戦争計画の最高原理で「目的の階層性」と呼ばれ、軍事教育の基本則である。19世紀プロシアの軍事哲学者 Clausewitz は過去の戦争例からを帰納した[16]。これを一般向きに説明する。

最高責任者には、最終目的(例:ある島の占領)が課される。彼/彼女はこれを実現手段群に展開し、部下達に課する。(海軍司令官:島から300Kmの地点に軍を送れ、空軍司令官:島の防衛戦力を撃滅せよ、陸軍司令官:上陸して島を占拠せよ)以後も同様に、各部下は課された命令を目的として、これを実現手段群に展開する。ここでは、「課された目的」が親概念であり、「実現手段群」が子概念になる。(概念レベルだからフローチャートはない。)

これは経営での事業計画にそのまま使えることはよく知られている。また、システム設計でも、

効果効用→サービス→外部機能

と展開して使える原理である[25, 26]。但し、システム設計の場合と同じく、もはやプログラムの制御の流れは見えない。なお、上流では可能な子は多数あり(ここでは無視するが)選択の知的処理が介在する。以後下流に降ると、選択幅は狭まり図5の下流ではほぼ1になる。

最下流の身体的動作の場合を見よう。軍事科学では、最終段階迄、前記の「目的の階層性」による階層展開を続けると考える。例えば、目的:「コーヒー茶碗を取出す」は、「棚を開け」、「茶碗を取り」、「机の上に置く」と展開する。ある目的の動作は、図4と同様に階層展開されて、実現手段である身体動作の指令になり、身体が動く。身体はI/O機器に例えることができる。

これらから、全てを「人の意図的行動」として一括して扱う。意図的行動の知的処理は階層展開の連鎖にな

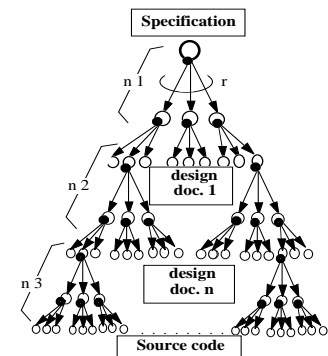


図5 階層展開網モデル

る。展開を単位的な進行・展開に揃えると、平均展開率は経験的に一定になる。そこで図5[4]の定率展開する階層展開網モデルでごく粗く近似する。白丸は情報、黒丸は単位的知的処理で、各処理は定率 r で展開する。

網の特性は等比級数で定量評価できる。黒丸の知的処理が微小時間を消費すると見做して計算すると工数が求まり、最終出力情報数で除せば生産性が求まる。また知的処理が微小確率で誤ると見做せば、誤り率が求まる[4]。このように、経験工学的に求めた、工程の線形性、生産性、欠陥作り込み率等が証明できる[4, 5]。以上で工数消費や欠陥作り込みのメカニズムが判った。生産性は知的処理速度に、欠陥率は知的処理の確実性に関わる。

最上流で決定に時間を要した場合は図5の上の $n1$ 段で、最下流で身体的動作に時間を要した場合は図5の上の $n3$ 段で、他より大きな単位的処理の工数を消費するが、これは線形性により平均値の増加をもたらすのみで、図5のモデルを覆す原因にはならない^{脚注4}。

筆者等は「人に倣ったソフトウェア自動設計」を掲げ、プログラムの制約条件を入れず、共通的な人の知に留意して、知的CASEツール群の研究を行った。初めは図4の概念展開の内の機能の階層展開を構造化チャートPADを用いて実現した。これらが実際の設計図面であり、確実な設計情報であることから選んだ。この段階以降は、人の知のシミュレータを逐次近似的に高度化して作る意図を

明確に持った。次には、データフロー図を加えて図4のプログラム様式で表記した人の意図的行動の再現に成功し、その特性を得た[17, 18]。以上の概要を図6により簡単に説明する。

階層展開以外の知的処理として、意図的行動の各断面には、「認知」→「決定」→「変換」の3段の知的処理がある。次は、人間工学のZipfが証明したヒトの「労力最小化の原則」がある。これは「ヒトは問題に直面すると、まず最简单的解を試み、もし不可ならより高度な解を試みることを、解を得る迄繰返す」[19]。Rasmussenはこの内の3典型を提唱した。「人の知識には、(反射的行動の)技能レベル、ルール(で解を作る)レベル、最後に(基礎知識から創造的に解を組立てる)知識のレベルがある」[20]。筆者等は「労力最小化の原則」が人の基本特性と考え、単純化の為にRasmussenの3レベルで人の設計動作の再現を研究した。

- 技能のレベル 親子の対を単位知識として知識ベースに蓄え、親を指定すると子が反射的に得られる。
- ルールのレベル フレーム形式のルールを用いる。詳細化した子データフローの基本文形をデータ、状況に合わせて完成させるメソッドからなる。
- 知識のレベル 基本概念の知識の辞書および解決法対応の処理論理から成立つ。

技能レベルの結果を説明する。図6. a[17]は、設計を繰返す時に新規設計ルールが蓄積される様子を直線尺度で示す。これを図6. b[17]に両対数尺度で表示した。図より対数習熟効果が判る。(これは重要だから、設計者を変え2回試行して確認した。)図6. c[17]はCASEツールを使う設計時間を習熟性工学を使い、定量評価したもので、ほぼ対数習熟効果を示す。図6. d [17]は習得知識を再利用して自動設計できる度合いを示し、早く昇るが約80%で頭打ちで浅智慧の限界を示

脚注4 19世紀末F. W. Taylorは、ある作業を下位の作業に展開し、ストップウォッチで時間を計測し、良い作業を組合せ、生産性を向上させた。ここでは線形性の生産性の一定性も保証は何も無く経験的に実施した。20世紀の前半に品質関係の技術が発達し1940年代には統計的品質管理が創始された。ここでは欠陥率=一定を前提にしている。これらは企業利益を増すから普及し、巨大なIEの体系が成長した。図4の設計の構造は、IEの基礎原理を夫々100年後、60年後に理論証明したと云える。

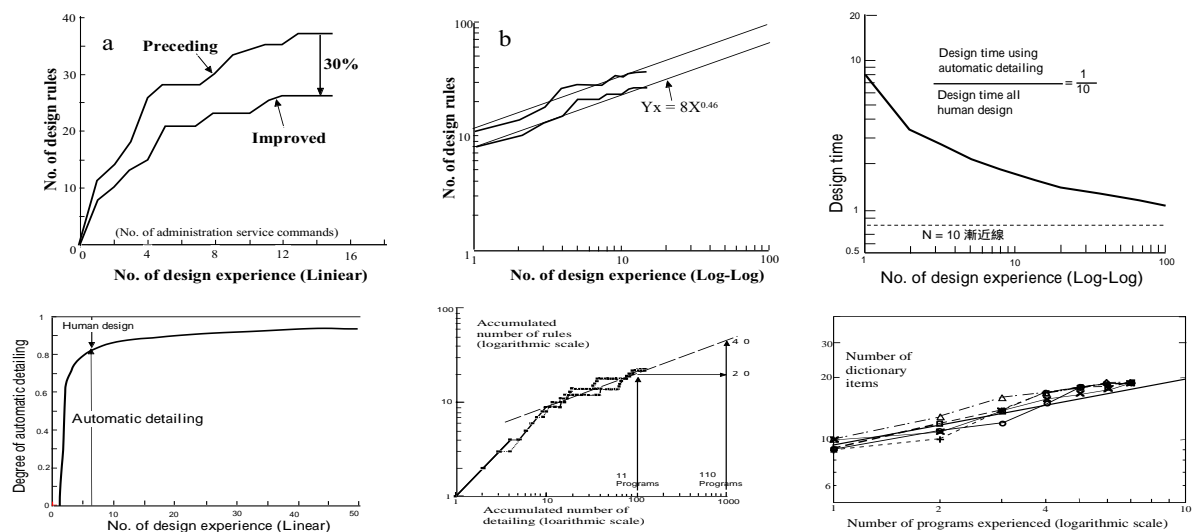


図6 人に倣った自動設計システムの特長

すと云える。

ルールレベルの研究結果はルールは図 6. e [18]で、同様に対数習熟効果を示す。更に進んだ知識のレベルでも、辞書の項目数も図 7. f [18]の対数習熟効果を示す。(これは Zipf の法則から生じるので、全てに現れる)

技能レベルからルールレベルに進むと自動化率は、ほぼ 100% 近い。大部分の設計作業は、殆ど考えもしないで、半ば自動的に手が動く設計者の実感に近い。

これら 3 モードを併置すると、一度知識レベルで解いた問題は、次回は技能レベルで高速に解けて、習熟効果は強くなる。技能レベルは最速であるが記憶容量が大きく、知識レベルでは記憶は最小でも速度は遅い。巧みな速度と記憶量のトレードオフである。この巧みな脳の仕組みは、場合毎に手を変えて、困難な問題を克服する能力と有限な記憶容量を活用する。生産性や欠陥作り込み率の習熟効果を見ると、前者は多面的な直接的な設計知識を技能よりはむしろより高次の記憶に、後者ではこれらの記憶の欠陥を補う知識を集積していると考えられる。生産性の差異の内、工程に関わる部分を取れば、それは、第 1 に「知の働きの反応速度」と理解でき、第 2 には「知の集積」によると理解できる^{脚注 5}。以上から既報告 [4, 5, 15] を更に強めて以下に纏める。

- ・(知的～身体的を問わず) 人の意図的作業は、線形性等の共通的基本特性を持つ。
- ・人の意図的作業の反復的あるいは多面的な繰返しは、知を集積する働きを持つ。
- ・人の意図的作業の外部特性(生産性等から夫々のバラツキを含む)は、人や組織の知の集積で定まり、集積量は能力上限を示す。

Taylor は作業効率を人の知の高さによると知り「出来高払い方式」を提案した(1896 年)。以後各種発展をとげて、Industrial Engineerin, IE と呼ばれる体系になった。1920 年代末にはモラル/志気～意欲の存在が知られ、Huamn Relation が作業管理に取入れられる。日本では 1955 年に産学官に労組上部組織を加えて「生産性向上運動」が展開され、1960 年代に大企業では全社員に IE の基礎を教育して、「標準作業時間制」が導入された。「全員を投じて〇〇を△日で作れ」等の無茶がない合理的なシステムとして各労組が受入れ、やがて全国に普及した。「俺達の技を時間で計るとは何事だ！」とすごむ方も居た製造現場は透明で合理的になった。上級設計者は「A1 図面当り作業時間や relay 当り設計時間は夫々ほぼ一定」等を見出し実地に応用した。1960 年代半ばからソフトウェア人員急増の波が、IE の先導的ハードウェア会社でもあるメインフレームメーカーに押寄せた。経営者達は当然の事として「合理的で透明

脚注 5 初期の知的 CASE ツールで 1998 年に技能レベルの習熟効果の理論付けができ、今回更に一般の習熟効果を含む人の意図的行動の理論付けができた。日本のソフトウェア工場の出発からでも約 40 年後になる。繰返してこれらを強調するのは、理論基礎がなくても実際の効果があれば産業界は利用する。「理論根拠が無いと採用しない」考えは産業上有害である。公理や定理から出発する理論的(所謂科学的)学問・技術とは根本的に違うことをご理解頂きたい。

な工場システム」の採用を決めた。これが世界に稀な「日本のソフトウェア工場」誕生の歴史である。その後は各社個別に自社技術を発展させている。

工学は強い実利主義を持っている。吉川弘之博士(現産業技術総合研究所所長、元東大校長、工学部長)は設計論の大家である。工学を次のように定義された [23]。

工学とは技術の科学である... 工学の体系は改変・保全・操作する対象自身の持っている性質と、快適という価値に導かれる人間の意図と、これら両者の相互作用とを包含する体系である。このことが、工学が理学と著しく異なる構造を持つことの原因である... 工学は理学の応用体系ではなく、理学が無くても工学は成立する.. その関係は、本質的なものではない。

IE では 19 世紀末から線形性や生産性の一定性を経験的に前提として出発した。謎とされたその基礎に筆者等の答えを出したのは 100 年後である。経験則でも十分な再現が見込めれば、理論根拠は必要ではない。更に、この見解は (自らは学の立場で働かれたが)、価値を産の立場に置かれていることは刮目に値する。

4. process と product の改善 (知の側面)

[6, 15] では改善の概要を報告したので、ここでは知の側面を報告する。割愛したが、成果物である product は process の外部特性と同じ立場だから、同様な対数正規分布に従う大きなバラツキを持つ。現場では process と同様に各種の統制を加える。product のソフトウェア規模については詳しく調べて [2] に報告してある。

ソフトウェア開発は知の集積が発揮される場である。知の集積は、大区分すると次の 3 段階でなされる。

1. Best practice 期 小学校 / 中学校と基礎を学ぶ。
2. 自領域の知識吸収 大学～就職後は、自ら吸収する。好例は GTE 社の優れた開発チーム [15] である。
3. 改善努力期 学んでも事業上では、必ず不満足な所がある。自分で原因を探索し自分で修正する。

技術問題は、全てを定量的かつ実証的に行う事が原則である。実測にはバラツキと、バラツキの要因が多数有り、その制御や再現は難しい。有意な結果は、以下の 3 方法で得られる。

1. 広範囲多様な資料から傾向を浮上らせる [28]。
2. 修正箇所以外は同条件と見做せる場合。・修正前と修正後の外部特性の相違は修正の結果と見做せる。・自チームのあるプロジェクトと後続した同一人員でもプロジェクトである遣り方を変える。
3. 高い純度が保証される場合 (例: テストならテスト設計～実施と照合と関係要因は少なく高純度化容易) 又は高成熟度組織なら微小傾向も鮮明に浮上る。通常使えるのは 2 項。工程は多元的ランダム性で他では再現し難い。定数は存在せず同一影響は他では保証できない。常にある傾向があった報告しかできない。実は

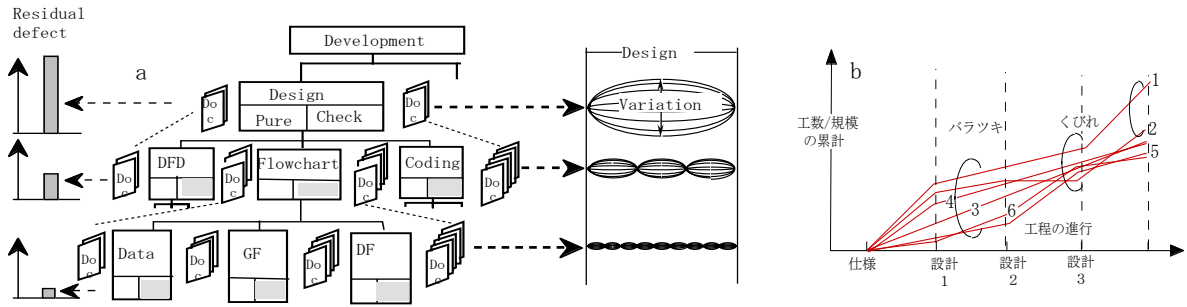


図7 工程の統制とその効果

ハードウェアで工程に関わることは、この認識で行われている。また、責任を持って云える人は本人しか居ない。従って process に関わることは、現場の人達しかできず、内容は改善報告になる。TQMはこの方式である。企業から学会に報告する時には下記を参考にされたい。

- ・外部公表は、ある効果等を見易く整理してメカニズムを明確にして、理解され易い結果を提示する。基本的に必ず正規化してコストや品質に関わる生数値は出さず、ノウハウになるレベル以下のある効果や関係に留める、以上なら公表できる筈なので、必ず社内認可を得て公開する。

図7.aを例として改善を知の視点から説明する。図の「設計」は最もバラツキが大きく、残留欠陥率が高かったとする。これを下位に展開する。(それには両端面を保存することが条件である。)中間が統制されて右図のようにバラツキが(展開数=3だから)約1/3に減る。各小設計工程毎に机上チェックすれば欠陥は摘出し易くなり、減少する。目安は1/3である。以上の理論的な見込み値で次プロジェクトが計画出来るならこれで良く、不足なら更に展開を続けるか、あるいは他の工程を改善する。

原理を学んでも、実際に効果を生むには適合させる研究が要る。図7.bはその実例で、横軸は工程の進行を、縦軸は累計工数/実績規模を、また各線はチームに対応する。図の右には実績カーブ群にくびれがある。それはこの端面が一連の作業の終着点で、生産性一定が現れている。その前には大きなバラツキが見え、これは問題である。現物を調べると、これらの端面の各文書の量がまちまちであった。そこで項目小項目を決め、(辻褄あわなくても良い)サンプルを準備した、問題は解消する。以後これを繰り返す。これらは自分で科学捜査を実行せねばならないことは明らかであろう。

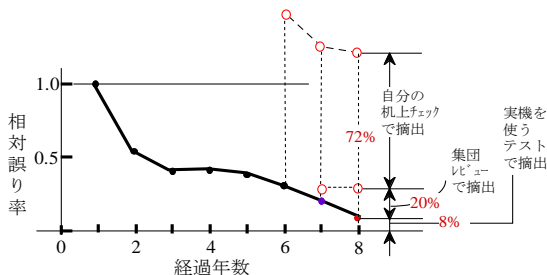


図8 保守10年間のテスト抽出欠陥密度

これら改善では、工程の外部特性を計測して良否を判定する。計測結果が悪いならその工程が悪い。原因と結果の関係が単純である。次に定量評価は何処が如何に悪いか、を明確化する。次に'Divide and Conquer'で、詳細化された次段階に進む。経営上問題になるコスト、品質、経過時間等の基本要素が全て計測できる。これだけ有用な metrics は他に無い。

知の集積の高い実例を紹介する。図8は、初期のアセンブラ(所謂スパゲッティ)プログラムである電子交換システムの保守を約10年間継続した記録である。横軸に開始後の年数、縦軸はテスト抽出欠陥密度を正規化してある。当時は少数のエキスパートが大多数の新人を率いて仕事した。彼は常に純設計=机上チェックで作業させた。全員が揃って計画を議論して、達成の為の具体策を出し、全員がコミットした。何処かのチームが遅れ始めると、彼はそのチームに飛び込み遅延を回復させる。既存文書以外に必要な設計文書を作り、皆で検討する。彼は皆が紳士としての誇りを持って仕事することを求めた。皆他人の設計の検討に耳を傾け、自分の設計ではその種の誤りを犯さないように努力する。仕事を始めてすぐに顧客から高い品質を評価され受注が殺到し、欠陥密度はしばらく足踏みする。しかし、自らが自分の誤りを事前抽出する癖が効果を奏し始め、図の右端では92%の欠陥を事前抽出した。約6~7年経過した頃から、このチームは事業所第一の精鋭チームと評価され始めた。単に技術に明るいのみでなく、仕事には常に積極的に前向きに取り組む、難題でも快く引受け確実に完成させる。接する人には、正しい礼儀と人への暖かい思いやりを感じさせる。このリーダーは、仕事を通じて部下達の技術のみでなく人格迄を向上させた。改善努力による知の積上げが如何に大きな効果を挙げるかが理解できる。(これも先の図3のチームも何れもTotal Quality Managementの優れたリーダーに率いられた優れたチームであった。

現在トヨタの好成績が評判で、世界のトップGMを間もなく追い抜く勢いである。少し前、Just-In-Timeが評判になり、最近ではKAIZENがクローズアップされ、全ての改善が鍵であると社会に理解し始めた。現在でも従業員一人当たり年平均2件の改善提案があると云う。総員の知の集積が今日のトヨタの好成績を作り上げた。

5. むすび

本報告は既報への付加分ではあるが、最後は全体を纏める。

1. 人の意図的行動である process の結果である product と該 process の外部特性は、上流から最下流の身体的行動迄、共通的な特性があり、それは共通的な人の知の働きである。

1. A 本方式な何にでも使える。

2. 人の意図的行動である process の外部特性は、共通的に線形系に見える。

3. 工程の外部特性は、中央傾向線式で表される値に多数の要因の積であるバラツキが重畳している。そのバラツキは統制の強さ / 純度に対応している。

4. この特質をソフトウェア側で使うことは、1960年代半ばから「ソフトウェア工場」で行われ、(報告は無いが) 夫々実績を積重ねている。

5. プロジェクト等では習熟効果が現れる。この繰返しの上に、各個人あるいは組織がある。その能力は、自らが調べ分析して改善され、集積して行く。これは知の集積と云える。(我々は既に知の世界に入っている。) 改善とは斯様な研鑽の集積した知である。

謝辞

この研究の前半は、主として河野が行った研究と発表に基づく。ご指導戴いた日立製作所幹部の方々、力になって下さった同僚各位や第一線の諸君達に感謝します。永い途を拓くはるか先の光を与えて戴いた母校の先生方に感謝します。後半は埼玉大学情報工学科で行った研究と発表に基づく。B. H. Far 先生はじめ研究室の学生諸君に感謝します。

引用文献

1. DeMarco, T. et al., Software development: State of the art vs: state of the practice, ICSE 1989.
2. Koono Z., Kondo T., Igari M. and Ohtsu K., Structural way of thinking as applied to good design (Part 1. Software size), Proc. of IEEE Global Telecommunications Conf. 1991, pp. 24. 3. 1-8, Dec. 1991.
3. 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, 情処技報 2005 - SE-147(12), pp. 89-96, Mar. 2005.
4. 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, Sept. 1996.
5. 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, Mar. 2004.
6. 河野善彌, 陳慧, 人の設計知識構造と定量評価(2/2), 信学技報, KBSE2003-58, pp. 73-78, Mar. 2004

7. 河野善彌, 陳慧, 誤りの作り込みから再発防止迄のメカニズム, 第24回ソフトウェア品質シンポジウム発表報文集, pp. 325-332, Sept. 2005.
8. 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), 情処学会研究会資料ソフトウェア工学 95-5, pp. 31-38, Nov. 1993.
9. 渡辺, 緒方, ソフトウェアの品質および生産性予測法の一事例について, 第2回ソフトウェア生産における品質管理シンポジウム, 1982.
10. Boehm, B. W., Software engineering economics, Prentice Hall, 1981.
11. Thayers, T. A., Software reliability study, RADC-TR-76-238, Rome Air Development Center, Aug. 1976. (British Library に有)
12. Salvendy, G. ed.: Handbook of Industrial Engineering", John Wiley and sons, 1982.
13. Koono, Z., Tsuji, H. and Soga, M., Structural way of thinking as applied to productivity, IEEE International Conference on Communications 1990, pp. 204.2.1-7, 1990.
14. Koono Z., Igawa K. and Soga M., Structural way of thinking as applied to improvement process, Proc. of IEEE Global Telecommunications Conf. 1988, pp. 40. 1. 1-6, Dec. 1988.
15. 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, 情処研報, 2005-SE-147 (12, pp. 89-96. Mar. 2005.
16. Cluasewitz, Karl von, Vom Kriege, 1832. 淡徳三郎(訳), 戦争論, 徳間書房, 1965.
17. Chen H., Tsutsumi N., Takano H. and Koono Z., Software Creation: An Intelligent CASE Tool Featuring Automatic Design for Structured Programming, JIEICE Vol. E81-D, No. 12. pp. 1439-1449, Dec. 1998.
18. 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, July 2002.
19. G.K. Zipf, Human Behavior and the Principle of Least Effort," Hafner Publishing, 1972.
20. Rasmussen, J., The role of hierarchical knowledge representation in decision making and system management, IEEE Trans. on Software Engineering, 18, 6, pp. 523-533, 1985
21. 社団法人情報サービス産業協会, JISA 基本統計調査
22. (株) 東レ経営研究所, 研究開発促進税制に係る企業ニーズ把握のための調査, 平成16年度産業技術調査報告書 (平成16年度経済産業省委託)
23. 吉川弘之, 工学, プリタニカ国際大百科事典, TBS プリタニカ, 1995.
24. 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年7月.
25. 河野善彌, 構造的な思考のシステムへの適用, 電子情報通信学会 信学技報 SE89-165, pp. 13-18, 1990年3月
26. Koono Z. and Soga M., Structural way of thinking as applied to systems design, Proc. of IEEE Global Telecommunications Conf. 1990, pp. 306. 1-7, Dec. 1990.
27. 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.