# 対象別評価の導入によるソフトウエアプロセス最適化の検討

3W－2

銭谷由彦　山口智治　白石智

NTTネットワークサービスシステム研究所

## 1.はじめに

ｿﾌﾄｳｴｱの開発において、大規模になれば扱うﾃﾞｰﾀも増大し、工程管理・品質管理の評価はよりｼﾋﾞｱなものになる。本稿では、大規模ｿﾌﾄｳｴｱ開発における膨大な開発情報を分析し（ﾃﾞｰﾀﾏｲﾆﾝｸﾞ）、会社、個人のﾚﾍﾞﾙを対象としたﾌﾟﾛｾｽ評価を可能とし、その値の組み合わせによりﾌﾟﾛｾｽの最適化をはかる方法を提案する。

今回はﾃﾞｰﾀﾏｲﾆﾝｸﾞの手法として相関・判別分析のｱﾙｺﾞﾘｽﾞﾑの適用を検討しており、報告する。

## 2.現在の開発支援環境

図1にあるように開発作業のﾃﾞｰﾀを蓄積し、作業標準書に定められた指標値を計算し、進捗遅れや品質に問題があれば開発管理者に警告するｼｽﾃﾑを構築している。


図1　自動警告システム

## 3.データマイニング

ﾃﾞｰﾀﾏｲﾆﾝｸﾞとは大規模ｿﾌﾄｳｴｱ等の開発作業において日々蓄積される膨大な生ﾃﾞｰﾀから、従来のﾃﾞｰﾀﾍﾞｰｽ検索では発見することの出来なかった有用な情報、例えば、ﾃﾞｰﾀ間や事象間の関連、法則、傾向などを発見する事である。これを実行することにより、従来の作業標準書には載っていない情報を獲得でき、新たなﾉｳﾊｳとして活用できる。このﾉｳﾊｳは開発が続く限り増えつづけ、しかも陳腐化しない。

具体的には、例えば「設計作業でのﾄﾞｷｭﾒﾝﾄ品質の良し悪しが試験作業の進捗に影響をもたらす」といった「法則」や、「このﾍﾟﾝﾀﾞでは製造進捗はいつもほぼこのぐらいである」といった「傾向」を導く事である。

技術的にはｿﾌﾄｳｴｱﾌﾟﾛｸﾞﾗﾑの項目単位（ex.ｸﾗｽ群）毎に多次元の要素で表現し、その多次元要素を既存ﾃﾞｰﾀﾍﾞｰｽからﾃﾞｰﾀ取得したものをﾌﾟﾛｾｽ特性基本ﾃﾞｰﾀとする。要素の取り方には留意する。

＜ex. n個の要素を持つﾌﾟﾛｾｽ特性基本ﾃﾞｰﾀ ＞
ｸﾗｽ群 Aaa＝ (x1, x2, x3, x4, x5, x6, ..., xn)
＝（設計進捗分析値，ﾄﾞｷｭﾒﾝﾄ品質分析値，製造進捗分析値、規模変動分析値，試験進捗分析値，試験品質分析値，…，問題処理分析値）

そして全てのﾌﾟﾛｾｽ特性基本ﾃﾞｰﾀを多次元に解析（ex. 相関係数計算、判別分析、回帰分析等）し特性を抽出し、「法則」や「傾向」とする。

（方式1）
相関分析・・・要素間の相関を数値で表す
相関分析とは
・一般に相関係数（ﾋﾟｱｿﾝの積率相関係数）Rxy は次の様に定義される。
ある2つの変数x、yについてn個のﾃﾞｰﾀが得られているとき、
Rxy = xとyの共分散/ xの標準偏差＊yの標準偏差

$$Rxy = \frac{1/n\ \Sigma(xi-\bar{x})(yi-\bar{y})}{sqrt(1/n\ \Sigma(xi-\bar{x})^2) * sqrt(1/n\ \Sigma(yi-\bar{y})^2)}$$

（ここで、 $\bar{x} = 1/n\ \Sigma xi$ ， $\bar{y} = 1/n\ \Sigma yi$ ）

を計算し相関の仮説を立てる分析法である。（図2）
Rxyは－1から1の値をとる（ $-1 \leqq Rxy \leqq 1$ ）

例えばx2とx6の相関係数の計算結果により仮説を立てる
x2とx6の相関係数が高い　x2とx6には強い関連がある
x2とx6の相関係数が低い　x2とx6は関係がない
（相関係数計算により x2（ﾄﾞｷｭﾒﾝﾄ品質分析値）と x6（試験進捗分析値）の相関係数が 0.8（相関高）という値が出たら「ﾄﾞｷｭﾒﾝﾄ品質が比較的悪い状態のまま製造作業に移ればいずれ試験作業時に進捗遅れとして影響が出る。」という仮説が成り立つ。）

（方式2）
判別分析・・・ある要素の判別を他の要素を分析することで表す
判別分析とは
・判別したい2群の平均値とｻﾝﾌﾟﾙのばらつきも考慮した距離（ﾏﾊﾗﾉﾋﾞｽの距離）を求め、より近い方のｸﾞﾙｰﾌﾟに属していると判定する分析方法。
(1) 既存のﾃﾞｰﾀ(変数k個)を集め、第i群の重心を
$(x1\bar{i}, x2\bar{i}, ..., xk\bar{i})$，第i群に属するﾃﾞｰﾀを用いて計算した分散共分散行列を$\Sigma i$とする
(2) 判別したいﾃﾞｰﾀとそれぞれの群との重心との偏差ﾍﾞｸﾄﾙを

$$Ui = \begin{pmatrix} x1-x1\bar{i} \\ x2-x2\bar{i} \\ ..... \\ xk-xk\bar{i} \end{pmatrix}$$ とする

(3)ﾏﾊﾗﾉﾋﾞｽの距離 $Di = Ui' \Sigma^{-1} Ui$ を計算する
(4)距離が最小の群に属すると判定する。
xnがどちらの群（満足行く結果or問題のある結果）になるか事前に判別したい場合、ｻﾝﾌﾟﾙﾃﾞｰﾀから判別用の値(x1, x2, x3, x4, ..., xk)を取り出し値を計算し判断する。

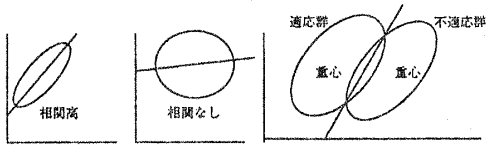



図2　方式1　相関分析




図3　方式2　判別分析

具体例）
設計ﾄﾞｷｭﾒﾝﾄのﾚﾋﾞｭｰ品質（ｺﾒﾝﾄ密度、ｴﾗｰ密度、ﾚﾋﾞｭｰ密度）について作業標準書で定めた閾値に違しないﾄﾞｷｭﾒﾝﾄがあった場合を例にとる。

通常であればこのようなﾄﾞｷｭﾒﾝﾄについては再ﾚﾋﾞｭｰ等の判定をしているが、本ﾃﾞｰﾀﾏｲﾆﾝｸﾞにおいては必要に応じてﾌﾟﾛｾｽ特性基本ﾃﾞｰﾀに設計作業に携わる開発担当者の個人ﾃﾞｰﾀを一部追加して分析を行う。通常では見落としがちな問題点や、その問題の解決となる仮説を導くとこに重点をおいて膨大な生ﾃﾞｰﾀから関連を導き出せそうなﾃﾞｰﾀを絞って分析を行うこととする。

Examination of software process optimization by introduction of evaluation according to object

Yoshihiko ZENIYA　Tomoharu YAMAGUCHI　Satoshi SHIRAISHI
NTT Network Service Systems Laboratories
9-11 Midori-Cho 3-Chome Musashino-Shi,Tokyo 180-8585 Japan

すなわち
＜分析に必要な追加ﾃﾞｰﾀ＞
個人 Bbb＝（個人 ID、ﾍﾞﾝﾀﾞ ID、設計ｽｷﾙﾚﾍﾞﾙ、開発経験年数、
ｴﾗｰ発見率、ｴﾗｰ原因率、再ﾚﾋﾞｭｰ時ｴﾗｰ発見率、
再ﾚﾋﾞｭｰ時ｴﾗｰ原因率…）

などのﾃﾞｰﾀを抽出し、上記ﾚﾋﾞｭｰ品質ﾃﾞｰﾀと相関分析を行う。
それにより、相関のあるもの、ないものが相当数得られるので一部挙げると
（相関高１）ｴﾗｰ密度が規定値に達しないﾄﾞｷｭﾒﾝﾄの開発担当者と開発経験３年未満の開発担当者の相関は高い
（相関高２）ﾚﾋﾞｭｰに参加したﾒﾝﾊﾞｰの平均開発経験年数とｴﾗｰ密度は正比例する
（相関高３）ﾚﾋﾞｭｰ実施時間（１２～２４）とｴﾗｰ発見率は反比例する
（相関無１）ﾚﾋﾞｭｰ参加者の設計ｽｷﾙﾚﾍﾞﾙとｴﾗｰ発見率は特に相関が無い。

また過去の開発ﾃﾞｰﾀを利用することにより、後工程のﾃﾞｰﾀも考慮した判別分析も行う。
それにより以下の判別情報が得られるので挙げると
（判別１）担当ﾍﾞﾝﾀﾞは、設計工程で再ﾚﾋﾞｭｰしたﾄﾞｷｭﾒﾝﾄにより製造したﾌﾟﾛｸﾞﾗﾑに対する試験作業において、作業標準値で定めた閾値の４／５以上のバグは試験期間内に検出できる
（判別２）担当ﾍﾞﾝﾀﾞが再ﾚﾋﾞｭｰしたﾄﾞｷｭﾒﾝﾄの製造作業において製造進捗遅れが発生する可能性は低い（確率２０％）
（判別３）再ﾚﾋﾞｭｰしたﾄﾞｷｭﾒﾝﾄにより製造されたﾌﾟﾛｸﾞﾗﾑに対する問題処理票の対応日数は再ﾚﾋﾞｭｰとならなかったものと判別はできない。

これらの相関関係、及び、判別情報から次の法則、傾向を導くことが出来る。

（法則１）開発経験３年未満の設計ﾄﾞｷｭﾒﾝﾄ数の何割かは再ﾚﾋﾞｭｰになる
（法則２）ﾚﾋﾞｭｰは平均開発年数を考慮してトータルの人選を行う。
（法則３）ﾚﾋﾞｭｰは夜遅くなるようなら無理をせず翌日行う
（法則４）ﾚﾋﾞｭｰの個人人選は設計ｽｷﾙﾚﾍﾞﾙは考慮しなくてよい

（傾向１）担当ﾍﾞﾝﾀﾞの再ﾚﾋﾞｭｰしたﾄﾞｷｭﾒﾝﾄはほぼ目標バグ密度値を満足しそうである。
（傾向２）担当ﾍﾞﾝﾀﾞの再ﾚﾋﾞｭｰしたﾄﾞｷｭﾒﾝﾄの製造進捗遅れはなさそうである。
（傾向３）担当ﾍﾞﾝﾀﾞの再ﾚﾋﾞｭｰしたﾄﾞｷｭﾒﾝﾄに対する問題処理対応能力は通常の場合と大差なさそうである。

## 4.プロセス評価

ﾃﾞｰﾀﾏｲﾆﾝｸﾞにより抽出された情報を元に、開発作業を会社、個人ﾚﾍﾞﾙで細かく評価していき、具体的な数字を提示することにより、各々最善をはかるように取り組ませることがより現実味を帯びたものになる。どこまで値をあげなくてはいけないという明確な目標が与えられるからである。その単位としては、作業標準書で定めた工程単位である場合もあるし、工程をまたがった作業でもそれを一つのﾌﾟﾛｾｽとしてまとめるのが良い場合もある。したがって、従来の作業標準書にはない柔軟な作業単位の設定が必要である。

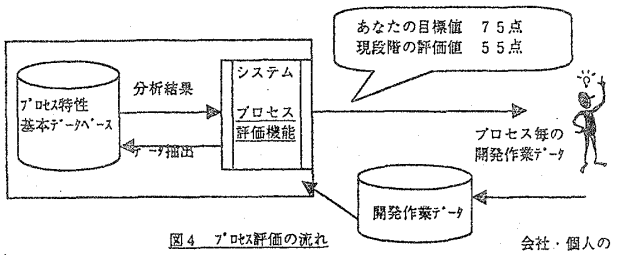
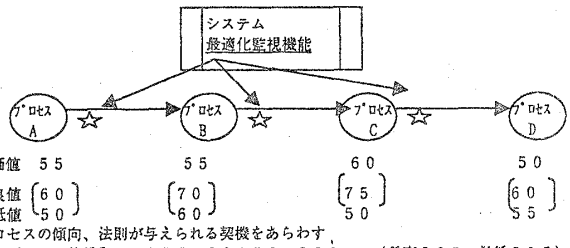


図４　ﾌﾟﾛｾｽ評価の流れ

会社・個人の
各ﾚﾍﾞﾙにて評価

## 5.プロセス最適化

ﾃﾞｰﾀﾏｲﾆﾝｸﾞにより抽出された情報により開発者の各単位でﾌﾟﾛｾｽ評価が実現できれば、それを最適化することが一番求められていることである。つまり一つのﾌﾟﾛｾｽの開始から終了までたどり着く最良の道を求めることである。具体的には図５のように、開発作業をﾌﾟﾛｾｽ単位で区切って、各ﾌﾟﾛｾｽの優良な評価値（＞平均値）を設定し、それにたどり着くための「法則」や「傾向」をﾀｲﾑﾘｰに開発当時者に知らせ、その実行状態を把握し、道から外れないようにﾌｫﾛｰしていく流れをｲﾒｰｼﾞしている。



☆はプロセスの傾向、法則が与えられる契機をあらわす。
今回プロセス値総和５５＋５５＋６０＋５０＝２２０　（最高２６５、最低２１５）
*プロセス最適化とは　各ﾌﾟﾛｾｽ評価得点の総和|A|+|B|+|C|+|D| を最大にすること*

図５　ﾌﾟﾛｾｽ最適化の流れ

## 6.本方式による効果

本方式により、開発作業のﾌﾟﾛｾｽ評価対象が明確になり、その評価も客観的に行え、その総和を最大にする努力をﾌｫﾛｰすることによりﾌﾟﾛｾｽ評価値の最適化、つまりはﾌﾟﾛｾｽ最適化につながる効果が見込まれる。

## 7.まとめ

本稿では、ｿﾌﾄｳｴｱ開発における膨大な情報をﾃﾞｰﾀﾏｲﾆﾝｸﾞによりﾌﾟﾛｾｽ評価に役立て、ﾌﾟﾛｾｽ最適化に利用する構想を示した。今後詳細な検討を進めて実現に必要な仕様を確認する。

## 参考文献

[1]現代人の統計２　新版　多変量解析法　柳井・高根共著（朝倉書店）
[2]ｿﾌﾄｳｴｱﾌﾟﾛｾｽ成熟度の改善 Watts S.Humphrey 著 藤野訳（日科技連）