

## ビジネスダイナミクスを活用したプロジェクト損益の変動予兆検知モデル Detecting Model and Simulator for Deterioration Signs of Project Profit Utilizing Business Dynamics

那須 弘明† 長岡 晴子† 中村 英生† 野口 昭人† 小野 俊之† 難波 康晴†  
Hiroaki Nasu Haruko Nagaoka Hideo Nakamura Akito Noguchi Toshiyuki Ono Yasuharu Namba

### 1. プロジェクト管理の背景と問題

企業にとって、大規模システム構築プロジェクト (PJ) の赤字撲滅は、重大な事業課題である。大規模 PJ は、一旦 PJ の状況が混乱するとその回復に甚大な追加工数を要し、それにより原価が膨れ上がり損益が悪化するため、企業の業績に与える影響が大きい。従ってその解決には、PJ を適切に管理し、損益の悪化の予兆を見極め、早期に対処を取ることが重要となる。

システム構築 PJ において、損益の変動に繋がる、PJ 管理項目は多岐に亘る。例えば、製品品質、作業品質、進捗の遅延度、コミュニケーションの円滑さなどである。その PJ 管理項目を体系化した知識体系に PMBOK(Project Management Body of Knowledge)がある[1]。PMBOK は、1996年に米国プロジェクトマネジメント協会(PMI)によって PMBOK ガイド(現第4版)に纏められ、現在では世界標準として広く利用されている。

一部の企業では、PMBOK の PJ 管理項目やリスクチェック項目のスコアリングシートを作成し、プロジェクトマネージャ (PM) が定期的にスコア付けして PJ 状況を把握することで、PJ 悪化を捉える手法が取られている[2]。しかし、項目間の影響を加味した損益への影響が具体的な金額として見えていないため、悪化の兆しを十分に捉えきれず、対策が遅れて大きな追加工数を発生させる場合がある。

PMBOK で推奨されており、損益を原価、進捗の観点から管理する手法として EVM(Earned Value Management)がある。EVM とは、PJ の進捗を作業の出来高によって定量化し、PJ の現在および今後の状況を評価する手法である[3]。しかし、EVM は、進捗や原価の計画値と実績値の差分から出来高の価値を定量化して最終損益を予測するため、進捗が遅れが開始してからしか、最終損益の悪化を発見できない。また、EVM は、進捗や原価で現状評価を行い、他の PJ 管理項目を考慮しないため、PJ の進捗や原価の予定と実績の不一致の原因までは明らかにできない[4]。PJ の悪化予兆を検知するためには、「進捗」「原価」「品質」が悪化する原因を特定し、それが連鎖的に波及することで最終損益にどの程度及ぼすかを定量的に測る必要がある。例えばコミュニケーションの円滑さの不十分さを検知し、コミュニケーションが円滑でないために生じる原価への影響が最終損益へどの程度、連鎖的に波及するかを定量化評価するというものである。

これらの問題に対し、本稿では、ビジネスダイナミクス (BD) というモデリング&シミュレーション技術を用い、スコアリング項目と損益との因果関係をモデル化し、各ス

コアリング項目のスコアから損益の変動を定量的に予測して、予兆を早期に検知する損益変動予兆検知モデルを提案する。BD は、事象内にフィードバック構造を含む連鎖的な要素間の因果関係を可視化し、定式化することで、構造上の問題を把握して定量評価を可能とする技術である[5][6]。BD は、システムダイナミクス (SD) の名称で知られているが、近年、ビジネスにおける問題解決に有用な手法としてその適用範囲を広げ、Sterman の著書「ビジネスダイナミクス」[6]でも有用な事例が多数示されている。本稿も適用先をビジネスに特化しているため、それに倣う。本稿は、2章で BD を活用した損益変動予兆検知モデルの概要を示し、そして BD を用いてスコアリング項目と損益の関係を定式化するにあたっての技術課題を述べる。そして3章でその解決手法を述べ、4章でモデルの検証を行う。

### 2. 解決アプローチとその技術課題

#### 2.1 損益変動予兆検知モデルの概要

損益変動予兆検知モデルは、PJ のある時点におけるスコアリング項目のスコアと業績シートの要員計画値を入力とし、項目間の因果関係により生じるスコアの低下/上昇と、それに伴う PJ への投入人月の追加/削減を予測することで最終損益を推定して、損益の変動予兆を検知する。入出力イメージを図1に示す。図1は、「評価対象PJの損益は、2009/11に赤字に転じる」と検知している例である。

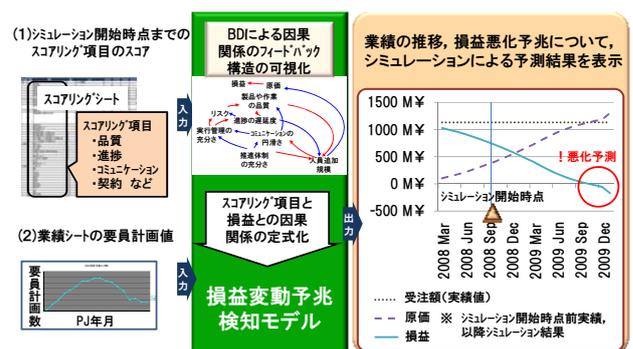


図1 損益変動予兆検知モデルの入出力イメージ

本モデルで入力に用いるスコアリングシートは、PMBOKを基にした約70項目のスコアリング項目から成る。スコアリング項目は、例えば、『進捗の遅延度』(以下『進捗』)、『コミュニケーションの円滑さ』(以下『コミュニケーション』)などである。各スコアは0~100点で危険なPJは毎月、安全なPJは3カ月置きという具合に、

† (株)日立製作所, Hitachi, Ltd.

定期的に PM, PMO(Project Management Officer)によって採点されるものである。以下, 説明で用いる『』はスコアリング項目を表す。

### 2.2 損益変動予兆検知モデル設計における技術課題

BD のモデル設計では, まず事象の要素間の因果関係を因果ループ図 CLD(Causal Loop Diagram)にグラフ化する。CLD では, 要素間の因果関係をリンクと呼ばれる矢印で結んで表す。要素の一方が上昇または増加するともう一方も上昇または増加する因果関係を表すリンクは同期リンクと呼ばれ, 矢印に“+”を付加して表現される。一方が上昇または増加するともう一方が下降または減少する因果関係を表すリンクは, 非同期リンクと呼ばれ, 矢印に“-”を付加して表現される。図 2 は, 複数の PJ の管理経験のある PMO にノウハウをヒアリングし, スコアリング項目と損益との因果関係を可視化した図(抜粋版)である。

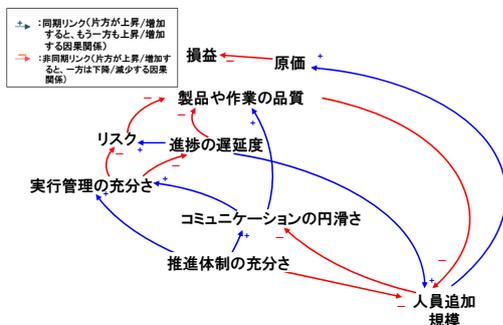


図 2 スコアリング項目間と損益との因果関係を可視化した CLD(抜粋版)

次に, CLD を基に定量モデルを作成し, 将来の各要素の変動度合いや施策の有効度を定量的に評価する。定量評価のための定式化は, 分析対象とする事象に対し, 個々の事象の特性を反映して定式化するアプローチがとられる。本モデルの場合, PJ マネジメント特有の性質を考慮してスコアリング項目間の因果関係を定式化する必要がある。プロジェクトマネジメントの分野に BD を用いた先行研究として, BD を用いた PJ リスク管理フレームワークを示した「Management and Modeling Project Risk Dynamics A System Dynamics-based Framework」がある[7]。しかし, このフレームワークでは, PMBOK に示されている6つのリスク領域を用いて可視化する概念のみを示し, その定式化手法については触れていない。よって PJ マネジメント分野に対する BD の定量モデル作成では, スコアリング項目と損益との因果関係に存在する PJ マネジメントの特性を反映する定式化が技術課題となる。

### 3. PJ マネジメント特性の定式化

PJ マネジメント特性とは次の3つのことである。1つ目は, スコアリング項目間の因果関係において, 原因側の項目(以下, 原因側)のスコアの値が, 結果側の項目(以下, 結果側)のスコアの変動に急激に影響を及ぼすことである。

例えば単位月当たりの追加人員の量が小規模であればコミュニケーションの円滑さは変わらないが, ある一定の追加人員を超えると円滑さが急に下がるというケースを指す。2つ目は, スコアリング項目間の因果関係には, 時間遅れが含まれることである。例えば, コミュニケーションの円滑さが不十分であるという影響が作業の品質に現れるのは, ある程度時間が経過してからである。3つ目は, PJ のどの工程であるかによってスコアリング項目間の因果関係の影響度が異なることである。例えば契約の詳細を締結していない状況は, 前工程では進捗に与える影響が小さいが, 後工程になるほど進捗に与える影響が大きい。以上から以下の(1)~(3)を踏まえた因果関係の定式化が技術課題となる。

- (1) 原因側の項目のスコアと結果側の項目のスコアの因果関係を表す関数の定義
- (2) (1)の関数において, 時間遅れを表す時定数の設定
- (3) 工程別に異なり因果関係の影響度を表す関数の定義

図 3 は, 図 2 に示す CLD から『実行管理の充分さ』(以下『実行管理』)と『進捗』との因果関係を抜粋して(1)~(3)の技術課題を CLD 上に例示するものである。技術課題(1)は, 図 3 の『実行管理』と『進捗』との因果関係の影響を表す関数  $f$  を定義することである。 $f$  は, 原因側の『実行管理』のスコアを入力とし, 結果側の『進捗』の結果を出力する関数である。(2)は『実行管理』スコアを  $f$  に入力して, 因果関係により影響を受けて『進捗』スコアが変動するまでの時間遅れ  $\tau$  を定義することである。(3)は, 工程を立上げ期, 開発期, テスト期に大別し, 工程ごとに異なる因果関係の影響度の違いを工程別の関数  $f_{\text{立上げ期}}$ ,  $f_{\text{開発期}}$ ,  $f_{\text{テスト期}}$  で定義することである。これらの解決アプローチによる損益変動予兆検知モデルを提案する。技術課題(2)に関しては, 時間遅れを表す時定数を1か月とする。この理由は, スコアが最短約一か月間隔で評価されることから, スコアの変化を把握できる最小の時定数であるためである。最適な時定数の検討は今後の課題である。

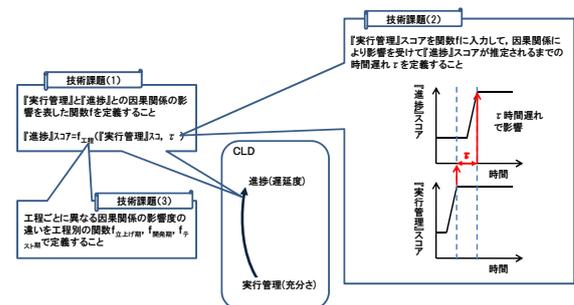


図 3 技術課題の CLD での位置付け

#### 3.1 因果関係の定式化の種類

技術課題(1)は, 原因側のスコアリング項目と結果側のスコアリング項目の関係が1入力1出力である場合と, N入力1出力である場合について定式化が必要となる。

図 3 の CLD から抜粋した図 4 の例で示すと, 前者は関数  $f$  を定義することであり, 後者は関数  $F$  を定義すること

である。関数  $F$  は『コミュニケーション』と『製品や作業の品質』(以下『品質』)との因果関係、『進捗』と『品質』との因果関係の両方の影響を考慮して定義する必要がある。技術課題(3)は、1入力1出力と  $N$  入力1出力の場合の両方で定義する。

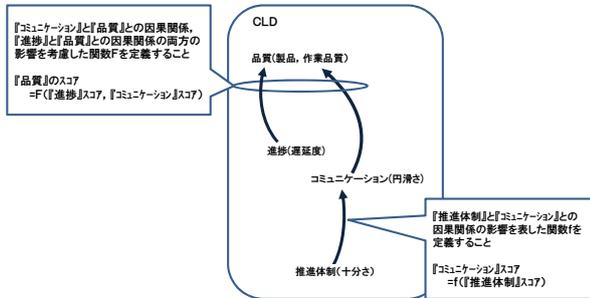


図4 因果関係定式化のCLDでの位置付け

3.2 スコアリング項目間の因果関係の定式化

3.2.1 1入力1出力である因果関係の定式化

(1) 定式化の考え方

PJのスコアリングでは、項目によっては、評価基準が定性的となるため、各評価者の経験や考え方がスコアに反映され、差異が生じる。例えば、『コミュニケーション』のスコアは、「必要なチーム、メンバ同士の十分なコミュニケーションがとれている(スコア 100)」「コミュニケーション面で孤立しているチームが1, 2存在している(スコア 60)」「SE, UP間のコミュニケーションが不十分(スコア 40)」といった基準に基づき、各評価者が判断して採点する。この場合、同じ状況下でも、例えば時点  $t$  で、評価者 A はコミュニケーションの円滑度が不十分とみなして、悲観的に『コミュニケーション』のスコアを 40 とする一方、評価者 B は、楽観的に 60 と評価するというように、スコアが評価者によって異なる。しかし、PJの状況が悪化、または改善するといった傾向の現れ方は人的差異が小さい。なぜならば、両者とも、自分の時点  $t-1$  の評価結果を基準にして、時点  $t$  の状況を比較し、良くなっているのか、変わらないのか、悪くなっているのかでスコアを付けるからである。従って、このような方法で得た評価データからの定式化には、スコアの絶対値ではなく、スコアの変動(上昇、不変、低下)に着目することが有効と考えられる。

これらの変動は、CLDでは原因側のスコアが影響して生じるとしてモデリングする。すなわち図3の例で言えば、『推進体制の充分さ』(以下『推進体制』)のスコアが『コミュニケーション』のスコアの変動に影響を与えているとモデリングしていることになる。変動が起きるケースを具体例で示すと、評価者は、推進体制の人員数が少人数の間はコミュニケーションの円滑さに変化はないが、コントロールできる限界 100 人を超えると、チーム内でのコミュニケーションの円滑さが急に低下するという認識から、低下したスコアを付けるということである。この状況を図式化すると 100 人を超えた状態を表す『推進体制』スコアを閾値にするステップ関数になる(図5)。

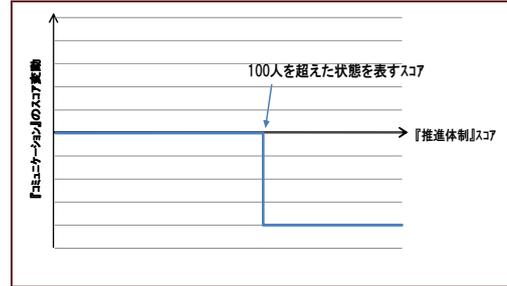


図5『コミュニケーション』と『推進体制』の因果関係

以上から因果関係は、結果側に低下、上昇変動を生じさせる原因側の閾値と、結果側のスコアの上昇、不変、低下変動を表すステップ関数  $g(x)$  で表せる(図6)。図6で結果側に低下変動、上昇変動を生じさせる原因側の閾値は、 $X1, X2$  である。つまり、スコアリング項目間の因果関係はスコア  $X1$  とスコア  $X2$  を閾値とするステップ関数  $g$  で定式化できると仮定する。

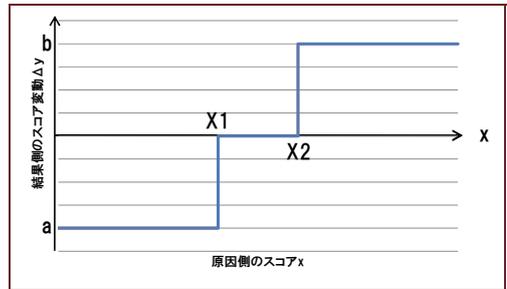


図6 結果側のスコア変動の関数  $g$

以上の考えに基づき、関数  $f$  を定義する。原因側の時点  $t$  カ月目のスコア  $x(t)$  を入力として、時定数である一か月後の結果側のスコア  $y(t+1)$ 、スコアの変動を表す関数  $g(x(t))$  を用いて以下のように定式化する。ここで  $a$  は関数  $g$  の低下変動の変動幅、 $b$  は  $g$  の上昇変動の変動幅を表す。

$$y(t+1) = f(y(t), x(t)) = y(t) + g(x(t)) \quad \text{ただし,}$$

$$x(t) \leq X1 \quad \text{ならば} \quad g(x(t)) = a < 0$$

$$x(t) \geq X2 \quad \text{ならば} \quad g(x(t)) = b > 0$$

$$X1 < x(t) < X2 \quad \text{ならば} \quad g(x(t)) = 0$$

項目間の因果関係によっては、閾値  $X1, X2$  はPJ工程ごとに異なる。例えば、リスクは後工程になるほど不確定要因が少なくなるため重要度が小さくなる。そのため『リスク』のスコアを上昇させる原因側のスコアの閾値は、基準が緩くなり閾値  $X2$  より低くなる。そこで、PJ工程を立上げ期、開発期、テスト期に大別し、各工程で原因側のスコアの閾値を定義する。結果側に低下を生じさせる工程別の閾値  $X1$  立上げ期、 $X1$  開発期、 $X1$  テスト期と結果側に上昇を生じさせる工程別の閾値  $X2$  立上げ期、 $X2$  開発期、 $X2$  テスト期を考慮すると、ステップ関数  $g$  は以下ようになる。

$$y(t+1) = y(t) + g(x(t)) \quad \text{ただし,}$$

$$x(t) \leq X1_i \quad \text{ならば} \quad g(x(t)) = a_i < 0$$

$x(t) \geq X2_i$  ならば  $g(x(t)) = b_i > 0$   
 $X1_i < x(t) < X2_i$  ならば  $g(x(t)) = 0$   
 ここで、 $i = \{\text{立上げ期, 開発期, テスト期}\}$

工程ごとの  $a_i, b_i$  は過去 PJ5 件の学習データから設定する。 $a_i, b_i$  は因果関係別にスコア低下, 上昇変動の変動幅の平均値を分析し, 採用する。

(2) 定式化モデルのパラメータ設定

学習データを用い, 定式化モデルの検証と閾値  $X1_{\text{立上げ期}}, X1_{\text{開発期}}, X1_{\text{テスト期}}$  と閾値  $X2_{\text{立上げ期}}, X2_{\text{開発期}}, X2_{\text{テスト期}}$  を設定する。図 2 において, 1 入力 1 出力の因果関係は, 『実行管理』と『進捗』, 『実行管理』と『リスク』, 『推進体制』と『コミュニケーション』間の因果関係である。

因果関係別に, 結果側が低下, 上昇した際の原因側のスコアとその頻度を分析した。表 1 は, 工程別, 因果関係別に, 結果側が低下, 上昇に転じた際の原因側の項目でもっとも頻度の高かったスコアを閾値として表記している。

表 1 項目間の因果関係ごと, 工程ごとに見た結果側を変動させる原因側のスコア閾値

因果関係 \ スコア閾値	立上期		開発期		テスト期	
	X1	X2	X1	X2	X1	X2
『実行管理』と『リスク』	40	60	40	60	45	60
『実行管理』と『進捗』	40	60	40	69	40	60
『推進体制』と『コミュニケーション』	52	60	40	70	40	60

ここで, 変動の傾向が工程別で変化しやすい『実行管理』と『リスク』の因果関係を例に取り, テスト期の閾値  $X1_{\text{テスト期}}, X2_{\text{テスト期}}$  の設定の考え方を示す。結果側(『リスク』)に変動が生じた一か月前の原因側のスコアのヒストグラムを図 7, 図 8 に示す。図 7 は結果側のスコアに上昇変動が生じた際の, 一か月前の原因側のスコア分布を示している。これから, 原因側(『実行管理』)のスコアが 60 点になると結果側のスコアに上昇の変動が生じやすいことが分かる。図 8 は結果側のスコアに低下変動が生じた際の, 一か月前の原因側のスコア分布を示している。これから, 原因側(『実行管理』)のスコアが 45 点になると結果側のスコアに低下の変動が生じやすいことが分かる。

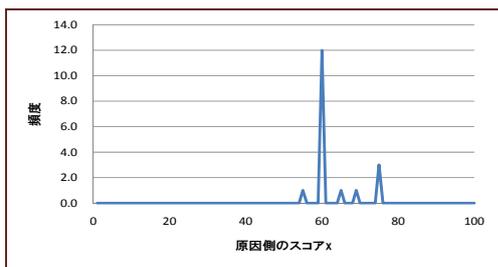


図 7 結果側のスコアが上昇したときの原因側のスコア分布

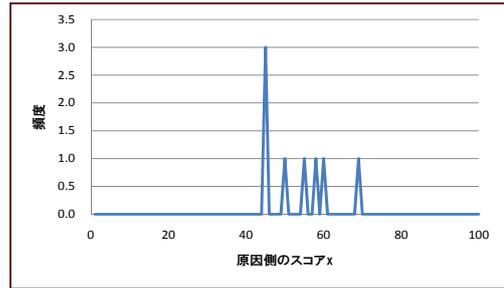


図 8 結果側のスコアが低下したときの原因側のスコア分布

これらの検証結果から, 『実行管理』と『リスク』との因果関係のテスト期の結果側の項目のスコア変動は, 原因側の項目のスコアの 45 点と 60 点を閾値にして起こっていると設定できる。すなわち,  $X1_{\text{テスト期}} = 45, X2_{\text{テスト期}} = 60$  である。また図 9 は結果側のスコアが不変であった際の, 一か月前の原因側のスコア分布を示している。これから, 原因側のスコアが 45 より大きく 60 点未満であると結果側のスコアは変動しない傾向にあることが分かる。

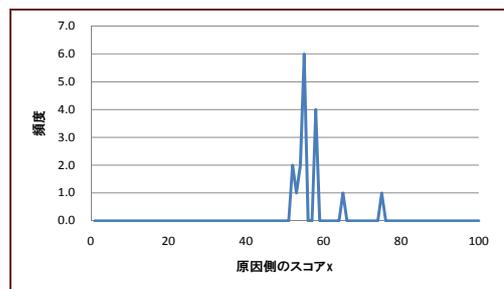


図 9 結果側のスコアが不変のときの原因側のスコア分布

分析の結果, 『実行管理』と『リスク』との因果関係の立上げ期の全てのスコア変動の内, 79%のスコア変動が, 閾値  $X1_{\text{テスト期}} = 45, X2_{\text{テスト期}} = 60$  で低下, 不変, 上昇していた。他の因果関係についても同様にし, 表 1 のように閾値を設定する。結果側のスコアの変動は, 因果関係別, 工程別に見て, これらの閾値で, 8割以上の変動を表せている。

3.2.2 N 入力 1 出力である因果関係の定式化

(1) 定式化の考え方

原因側の項目が複数ある場合, その構造に含まれる 1 入力 1 出力の因果関係は, 1 入力 1 出力の因果関係が  $n$  個あるとすると, スコア  $X1, X2$  を閾値とするステップ関数  $g_1, g_2, \dots, g_n$  と仮定できる。

本モデルは, スコアの変動を捉えて, 悪化を早期に予兆して検知することが目的である。そのため N 入力 1 出力の因果関係では,  $g_1, g_2, \dots, g_n$  に対して, スコアの低下を優先して定量化するモデルとなるよう定式化する。スコアの低下を優先する定式化とは, 結果側のスコアに低下を生じさせる閾値以下の入力値が 1 つでもあれば, スコアを低下させることである。よって以下のような論理和で  $g_1 \sim g_n$  を合成する関数  $G$  で因果関係を表す。 $g_1, g_2, \dots, g_n$  の入力スコアは,  $x_1, x_2, \dots, x_n$  で表記する。

$y(t+1) = F(y(t), x_1(t), x_2(t), \dots, x_n(t))$   
 $= y(t) + G(x_1(t), x_2(t), \dots, x_n(t))$  ただし,  
 $(x_1(t) \leq X1) \vee \dots \vee (x_n(t) \leq X1)$  ならば  $G = a < 0$   
 $(X1 < x_1(t) < X2) \wedge \dots \wedge (X1 < x_n(t) < X2)$  ならば  
 $G = 0$   
 全ての  $x_j(t) > X1$   $\wedge$  一つ以上の  $x_j(t) \geq X2$   
 ならば  $G = b > 0 (1 \leq j \leq n)$   
 なお,  $i = \{ \text{立上げ期, 開発期, テスト期} \}$

さらに工程による閾値の差異を考慮する合成関数  $G$  の定式化を考える. 例えば, システム品質は後工程になるほど重要になる. すなわち結果側の項目が『品質』である場合, 後工程では, より品質に対する基準が厳しくなるため『品質』のスコアに変動を生じさせる原因側のスコアの閾値は, 閾値  $X2$  より高くなる. よって立ち上げ期, 開発期, テスト期の各工程で原因側のスコアの閾値を設定して工程別に合成関数を定義する. 工程別の影響は, 『品質』などのように結果側の特性によって, 閾値に影響が出る. よって  $G$  は, 結果側の項目が低下する工程別の閾値  $X1_{\text{立上げ期}}, X1_{\text{開発期}}, X1_{\text{テスト期}}$  と上昇する工程別の閾値  $X2_{\text{立上げ期}}, X2_{\text{開発期}}, X2_{\text{テスト期}}$  を用いて以下のように定式化できる. なお,  $i = \{ \text{立上げ期, 開発期, テスト期} \}$ ,  $g$  は  $G$  に含まれる1入力1出力の因果関係を示す関数,  $j$  はその関数の数を表す.

$y(t+1) = y(t) + G$  ただし,  
 $(x_1(t) \leq X1_i) \vee \dots \vee (x_n(t) \leq X1_i)$  ならば  $G = a_i < 0$   
 $(X1_i < x_1(t) < X2_i) \wedge \dots \wedge (X1_i < x_n(t) < X2_i)$   
 ならば  $G = 0$   
 全ての  $x_j(t) > X1_i$   $\wedge$  一つ以上の  $x_j(t) \geq X2_i$   
 ならば  $G = b_i > 0 (1 \leq j \leq n)$

ここで, 学習データから閾値  $X1_i, X2_i$  設定の考え方を示す. 各  $g_j$  の各工程で, 1入力1出力の時と同じ方法で, 閾値が求まる. 各  $g_j$  の工程  $i$  における閾値は,  $X1_{i, g_j}, X2_{i, g_j}$  と表す.

本モデルはスコアの低下を捉えて, 損益悪化を早期に予兆検知することが目的である. そのため閾値には, 工程ごとに, 全ての  $g_j$  の閾値の中で, 最も大きい値を採用する. つまりスコアを上昇させにくく, 低下させやすい値となる.

$$X1_i = \text{MAX}(X1_{i, g_j})$$

$$X2_i = \text{MAX}(X2_{i, g_j})$$

このようにして求めた閾値  $X1_i, X2_i$  を用いた関数合成  $G$  のイメージ図を, 図 10 にテスト期の例で示す. 工程ごとの  $a_i, b_i$  の学習データからの設定は, 各  $g_j$  のスコア低下, 上昇変動の変動幅の平均値  $a_{i, g_j}, b_{i, g_j}$  を求め, 全ての  $a_{i, g_j}, b_{i, g_j}$  の中で各々最小値を  $a_i, b_i$  として採用する. その理由は, 本モデルがスコアの低下を捉えて, 損益悪化を早期に予兆検知することが目的であるからである.

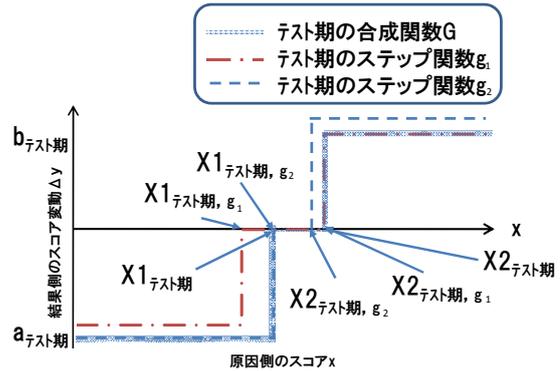


図 10 合成関数 G の例

(2) 定式化モデルのパラメータ設定

工程別に学習データである過去 PJ 5 件のスコア変動データを分析して, 因果関係ごとに, 結果側が低下, 上昇する際の原因側のスコア閾値の変動を分析し, 閾値を設定する. 表 2 は, 各工程, 各因果関係で, 結果側のスコアが低下, 上昇に転じた際の原因側のスコアの閾値をまとめた表である. 結果側のスコアの変動は, 因果関係別, 工程別に見て, これらの閾値で, 7 割以上の変動を表せている.

表 2 工程ごと, 因果関係ごとに見た, 結果側が低下, 上昇した際の原因側のスコア閾値

スコア閾値 結果側項目	立上げ期		開発期		テスト期	
	X1	X2	X1	X2	X1	X2
『品質』	40	60	49	60	40	70
『実行管理』	51	60	40	60	54	70
『推進体制』	52	60	48	60	40	60

4. 提案モデルの評価

3 章までに述べた定式化および閾値設定結果に基づいて, 図 2 に示す CLD の定量モデルとなる損益変動予兆検知モデルを開発した. このモデルを, 3 章でモデルの学習用に用いた過去 PJ データとは別の 6 件の PJ①~⑥を検証データとして用いて妥当性と有用性を検証する.

4.1 損益変動予兆検知モデルの妥当性検証

モデルの妥当性は, PJ 開始時からシミュレーション開始時点までのデータを用いて評価する. どの工程からも最終損益を予測し, 悪化予兆を検知できるか否か評価するため, シミュレーション開始時点は, 立上げ期 (要件定義~基本設計), 開発期 (詳細設計~納入), テスト期 (システム/総合テスト~稼働) のそれぞれから一時点を選ぶ. 各工程で選出したシミュレーション開始時点とその根拠を以下に示す.

立上げ期: スコア, 要員計画値のデータが揃う最初の時点. 選出根拠は, PJ の最初の時点で最終損益を予測できるか評価するため.

開発期 : スコアリング項目の半数以上に変動が生じた最初の時点. モデルがスコア変動を捉えて損益悪化, 改善を予測できるか評価するため.

テスト期 : システム/総合テスト開始時. モデルが, PJの終了に近い時点で, 実際の最終損益を予測できるか評価するため.

その各時点のシミュレーションが算出した赤字/黒字の結果と実際の最終損益の赤字/黒字の結果の一致を見て, 合致度を検証する(図11). その結果とPJの期間, 実際の損益の推移を表3に示す. 各工程でシミュレーション結果と最終損益が一致した場合を○とし, 一致しない場合は×として評価した結果, 立上げ期では6件中4件, 開発期では6件中5件, テスト期では6件中5件一致した. この結果から, 実用に十分耐え得ると考える. これにより本モデルの挙動の妥当性が検証できた.

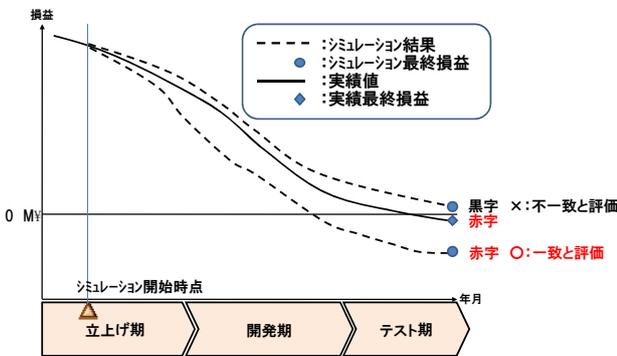


図11 モデルの妥当性検証の評価の考え方

表3 モデルの妥当性検証結果

検証PJとその属性		検証結果		
PJ名	実際のPJ期間, 損益の推移	立上げ期	開発期	テスト期
PJ①	23 か月, 途中で損益が変動	×	○	○
PJ②	30 か月, 途中で損益が変動	×	○	○
PJ③	30 か月, 損益は終始赤字	○	○	○
PJ④	24 か月, 損益は終始赤字	○	○	○
PJ⑤	12 か月, 損益は終始黒字	○	○	○
PJ⑥	19 か月, 最終月, 損益が変動	○	×	×

4.2 損益変動予兆検知モデルの有用性検証

モデルの有用性は, 損益悪化の早期予兆という本来の課題を解決できるか否かで評価することとし, 従来技術と比較して次のように検証した. 評価に用いるPJは, PJの途

中段階では最終損益が黒字と予測されていたが, 実際の最終損益が赤字となった過去PJとなる. このような過去PJには, ①, ②が相当した. PJ①, ②に対して, 本モデルと従来のスコア監視方法, および, EVMの予兆検知できたタイミングを比較した. 結果, 本モデルは, 従来のスコア監視方法より最終損益の悪化予兆を平均6.5ヶ月先行して検知できた(表4). EVMとの比較においては, PJ①では, 5ヶ月先行して予兆検知でき, PJ②では, 同じタイミングで予兆を検知できた. 以上から本モデルは, PJ①, ②では有用と言える. しかし, サンプル数が少なく検証が十分とはいえないためデータを増やし, 今後検証していく.

表4 モデルの有用性に関する従来手法との比較結果 予兆検知できたタイミング

PJ名	実際の見込み損益の変動月	本モデル	スコア監視法	EVM
PJ①	12ヶ月目	7ヶ月目	12ヶ月目 予兆失敗	12ヶ月目 予兆失敗
PJ②	24ヶ月目	16ヶ月目	24ヶ月目 予兆失敗	16ヶ月目

5. まとめ

PJマネジメント項目間の因果関係をモデル化し, 各項目のスコアから損益の変動を定量的に予測し, 予兆を早期に検知する損益変動予兆検知モデルを提案した. このモデルを過去6件のPJに適用し, 工程ごとに設定した開始時点から損益の挙動をシミュレーションしたところ, その挙動は, 立上げ期では6件中4件, 開発期では6件中5件, テスト期では6件中5件一致した. また本モデルは, 従来のスコア監視方法より, 最終損益の悪化予兆を平均6.5ヶ月先行して検知でき, 最終損益を予測する従来技術EVMより, 平均2.5ヶ月先行して検知できた.

参考文献

[1] “プロジェクトマネジメント知識体系ガイド第4版 - PMBOKガイド,” Project Management Institute, 2009.  
 [2]井上茂, “リスク管理,” UNISYS TECHNOLOGY REVIEW 第67号, NOV. 2000.  
 [3] “EVM 活用型プロジェクトマネジメント導入ガイドライン,” 情報処理振興事業協会, 2003.  
 [4] “プロジェクト管理における定量化手法 - 「EVM」を利用したプロジェクト管理の可視化-,” LS 研, 研究成果報告, 2003, <http://jp.fujitsu.com/family/lsken/activity/work-group/03/abstract/pdf/2003-15abstract.pdf>.  
 [5]J. W. Forrester, “The Beginning of System Dynamics,” System Dynamics Society, 1989.  
 [6]John D. Sterman, “Business Dynamics, Systems Thinking and Modeling for a Complex World,” McGraw Hill Higher Education Press, 2000.  
 [7]A. G. Rodrigues, “Management and Modeling Project Risk Dynamics A System Dynamics-based Framework,” PMI Europe 2001, London UK, 6/6-7, 2001.