

プロダクトライン開発におけるROI計算の一考察

—「クマのプーさん」のコア資産モデル—

Consideration about the ROI Calculation in Product Line Development

-Core Asset Model of “Winnie-the-Pooh”-

永 嶋 浩

NAGASHIMA, Hiroshi

1. はじめに

近年、注目を集めているモノづくりに米国 Carnegie Mellon大学ソフトウェア工学研究所 (SEI:Software Engineering Insutitute)の提唱したソフトウェアプロダクトライン(SPL: Software Product Line)なる開発形態がある。SPLは製品系列(製品ライン)を意識したソフトウェア開発の方法論である。プロダクトライン(Product Line)という意味は、戦略的な製品のシリーズ展開を実現するためにコア資産として体系化を図り再利用目的で構築したプロダクトを活用しながら開発する製品群を指す。SPLではコア資産開発、製品開発、管理からなる活動内容の枠組みが設けられている。SPLは日頃からコア資産の整備を心掛けながら、製品開発段階になったときにそのコ

ア資産を再利用して生産性を向上させ、費用対効果の面でも恩恵を得られるような運用体系を持つ。そこには再利用(Reuse)をキーワードにする開発の新しいパラダイムがある。従来のソフトウェア開発方法論は、ウォータフォールモデル(1970年Royce)をはじめとしてプロトタイプモデル(1980年Zelkowitz)やスパイラルモデル(1988年Boehm)などいろいろ提唱され、1990年代後半には「俊敏な(agile)」の意味をキーワードにしたアジャイル開発プロセスなる斬新な方法論も登場している。アジャイルの代表的な実践例には1996年にKent Beckの提唱したエクストリームプログラミング(XP:eXtream Programming)がある。

モノづくりはなにもコンピュータの絡むビジネス分野ばかりではない。世の中のありと

コア資産開発・・・開発対象製品の再利用価値を高めるための必須活動になる。コア資産にはモデル、パターン、ツール、フレームワーク、コンポーネントプログラム、テストケース、仕様書、開発環境、ソースコード等が関係する。

製品開発・・・・・・プロダクトラインスコープ、コア資産、プロダクトプラン、個々の製品要求が関係する。
管理・・・・・・技術レベルマネジメント、組織レベルマネジメントで構成され、リーダーシップのとれる管理者を置くことが必要になる。

キーワード: プーの物語、ソフトウェアプロダクトライン、コア資産、ROI
Key words : Pooh's Story,Software Product Line,Core Asset,ROI

あらゆるモノが対象になる。オブジェクト指向を標榜したシステム作りでは当然であり、世の中を見渡すとシリーズ化された文庫本、文学の世界の中にもモノづくりのストーリー展開を見ることができる。

2. プーの物語

A.A.ミルン(Alan Alexander Milne)が息子ロビンのために1926年に「クマのプーさん」^[1]、1928年に「プー横丁にたった家」^[2]を書き、大評判となって今に語り継がれている作品を取り上げる。ソフトウェア工学と呼ばれる言葉が誕生したのは、1968年のガルミッシュ(当時西独)で開催された「ソフトウェアエンジニアリングに関する国際会議」においてであったが、そこではモノづくりが家内工業的職人手作り風から工業製品を作り上げるかのごとく規格化された量産スタイル風の作り方が求められ議論されたものである。プーの物語とソフトウェア工学の話題では時系列や時代的背景、専門分野を考えると互いの接点はないが、よくプーの物語を分析すると「プーの世界」の中にモノづくりの真髄を垣間見、知見をえることができる。「プーの世界」には、ソフトウェア工学で取り上げられる以前からウォーターフォールモデルがあり、プロトタイプモデル、スパイラルモデル、アジャイルに至る数々のヒントが組み込まれている。モノづくりの開発手法を提供したのはRoyceなのか、Boehmなのか、Kent Beckなのか、それはプー(作者ミルン)ではないのかと思えるほどである。ミルンの著した二冊とも10章からなるストーリーで構成されている。その中で物語の結末が失敗に終わったのは、「クマのプーさん」では4章分、「プー横丁にたった家」では1章分ある。ミルンは、ストーリーの展開が

迷惑行為を目的としている内容を取り上げた結末は失敗扱いにしている。これらを除くと「クマのプーさん」の所で取り上げられた「プーがハチミツをとる話」、「モモンガーを追跡する話」、「ゾゾを捕まえる話」の3章分を失敗話で締めている。以下に4つの開発手法の例を示す。

2.1 プーがハチミツをとる話

プーは、カシの木のでっぺん付近にあるハチミツを採ろうとして、課題・実践・結果の3つの視点で計画を立てて仕様を決め、実践したが失敗に終わる。再度対策を講じて挑戦するが、再び失敗してしまう。計画には「起こり得る出来事」、「突然の計画変更」に対処できるように管理されていることが大事なのだと教えられる。図1にプーの手法と現在の工程要素の関係を示す。

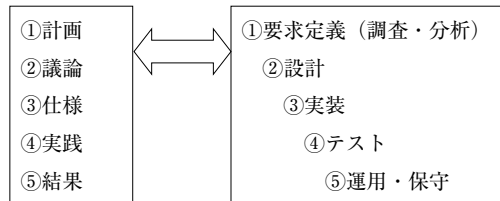


図1 プーの手法と工程要素の関係

表1 ハチミツ採取のストーリー

目的	ハチミツ採取(何かをゲットする)
示唆	モノ作りには予期せぬハプニングがあり、計画は常に変更を伴うことを認識しておくことが必要となる
結末	失敗
開発手法	ウォーターフォールモデル

2.2 プーがイーヨーのしっぽを見つける話

プーは森を散歩しているときに元気のないロバのイーヨーに出会い、イーヨーのしっぽがないのに気づく。直ちに探しに行くが知恵もののフクロの意見を聞くためにフクロの家

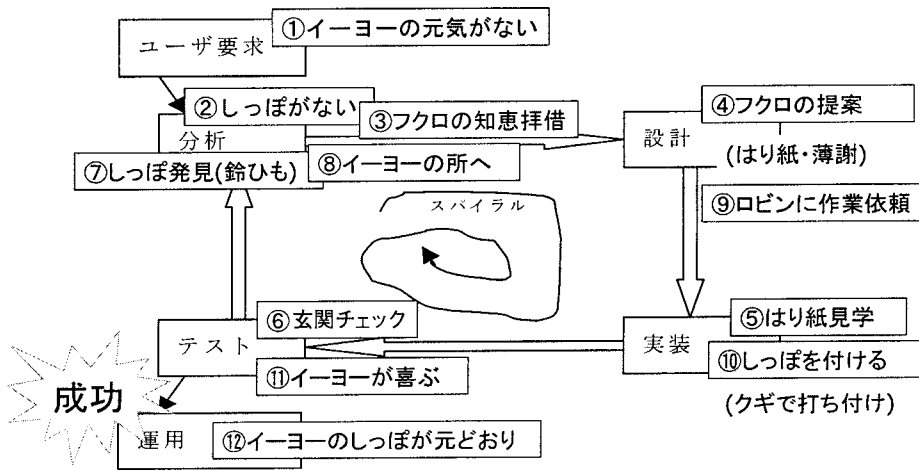


図2 スパイラルのストーリー展開

を訪ねる。そこでプーはフクロの玄関で発見する。玄関は1度目には気づかず素通りしているが、後でフクロに言われて再度玄関を見に行ったときに発見する。このイメージは縦

型検索のバックトラックに相当する。分析からテストの工程を何回か反復して作業を進めると、見過ごし防止ができると教えられる。

表2 しっぽ探しのストーリー

目的	しっぽ探し (探し物を検索する)
示唆	最初は見えないものが繰り返すことで見えるようになる
結末	成功
開発手法	スパイラルモデル、及びXPの共同所有権 (アジャイルプロセス分野)

2.3 ゾゾを捕まえる話

プーはコプタと一緒にゾゾを捕まえる決心をする。ゾゾに関する情報がないため、自分に置き換え写像の手法で計画を練る。わなには穴とエサを用意して捕獲されるのを待つ。しかし、プーはその後どうなったかが気になり深夜に起きて現場へと行く。そこで自分が

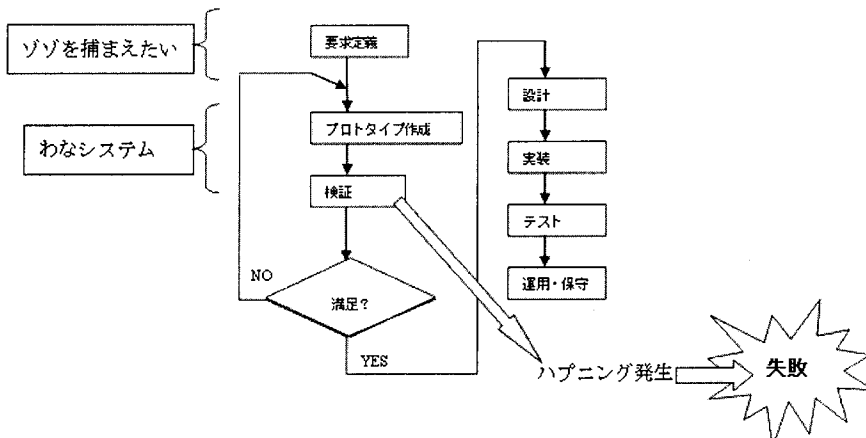


図3 プロトタイプのストーリー展開

わなにはまってしまう。試作品としての原型を作り改良して発展させていくのが本来の取り扱いであるが、途中での安易な検証を行ったため捕獲に失敗する。検証には事前にイメージトレーニングを行い手順通りにしないといけないということを教えられる。

表3 ゾゾ捕獲のストーリー

目的	ゾゾ捕獲（課題を検証する）
示唆	検証のためのテストには計画が必要になる（勝手な行動は失敗をまねく）
結末	失敗
開発手法	プロトタイプモデル

2.4 イーヨーの誕生日にプレゼントする話

年老いたロバのイーヨーが小川の岸で頭をかしげているので、プーが声をかける。イーヨーは自分の誕生日に誰も何もしてくれないのでしょげていた。プーは「イーヨーにお祝いをしよう」と考える。プーはコブタを仲間

表4 誕生日プレゼントのストーリー

目的	誕生日プレゼント（善意で実行する）
示唆	状況に応じた俊敏な対応が成功を導く
結末	成功
開発手法	XPのメタファー・共同所有権・オンサイト顧客（アジャイルプロセス分野）

に誘ってプレゼントを持って行くが、途中でハプニング（プーはハチミツを食べてしまい、コブタは風船を割ってしまう）があり事前に考えていたプレゼントとは異なってしまふ。

しかし、イーヨー自身は気持ちを切りかえてプレゼントを喜ぶ。短い間隔で開発工程を繰り返し、仕様変更俊敏に対応するXPの手法が顕著に表れたストーリーになる。ここではXPの「12のプラクティス」の3番・8番・11番が使われている。

3. メタファー（Metaphor）

→システム開発に共通認識を共有する

8. 共同所有権（Collective Ownership）

→間違いは見つけた人が速やかに修正する

11. オンサイト顧客（On-Site Customer）

→顧客も開発チームの一員である

新仕様

1. ミツの入っていない「つぼ」を「重宝なつぼ」と定義する（プー）

2. つぼは物の保管に使うと定義する（イーヨー）

例：イーヨーは割れた風船をつぼに入れたり出したりしながら便利であると喜ぶ

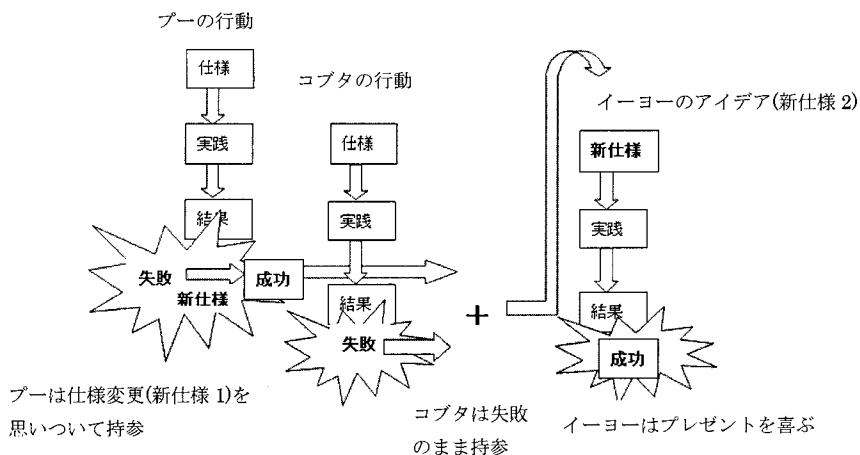


図4 XPのストーリー展開

3. プロダクトライン開発

SPLにはコア資産開発、製品開発、管理で構成された3つの必要な活動が存在する^[3]。これらの活動内容を効果的に運用するためドメインエンジニアリングとアプリケーションエンジニアリングの開発プロセスがある。プロダクトライン開発では、ドメインエンジニアリングとアプリケーションエンジニアリングの両プロセスがお互いに関係付けされながら並行作業で実践される。ドメインエンジニアリングではプロダクト分析後に行うコア資産の開発そのもの、つまりコア資産開発が中心になる。アプリケーションエンジニアリングでは目的別製品分析後にコア資産の再利用や製品毎に固有な機能の開発、さらにその開発の中からもコア資産化が可能かどうかの検討・抽出という活動が行われる。

図5にプロダクトライン開発の各活動の関係性をバス構成で表し、さらに図中の回転する矢印で用途に応じた活動が絶えず進化していくイメージを模して表現する。ドメインエンジニアリングはD1→(D2)→コア資産開発→D3のフローでコア資産を作り上げる。

アプリケーションエンジニアリングはA1→A2→(A3)→製品開発→(A4)→A5のフローで製品を作り上げる。

モノづくりには時代（局面）の要求に応じてきた開発手法の変遷がある。2.ではプーの物語の中に古典から最新に至る多くの開発手法が存在することを述べた。ここでは「クマのプーさん」にSPL型開発の適用を考える。まず森のメンバーのモノづくりの様子をモデル化する。モデル化のポイントはストーリーの中に計画、分析、プーのアイデア（他の仲間のアイデアも含む）、実践、実践中に結末がどうなったか等の調査をすべての物語に施してデータシートに整理する。その中からストーリー間で共通に使われている事柄に関してはコア資産として扱うようにする。コア資産開発に該当する箇所は、アイデアの前段にある分析／議論の所を割り当てる。このようにして図5のプロダクトライン開発を採用した構成のモデルを導出する。SPL型開発をプー用に作成したコア資産モデルを図6のような構成で示す。以下、このコア資産モデルを用いてコア資産開発やコア資産、製品開発に関するプーの事例を説明する。

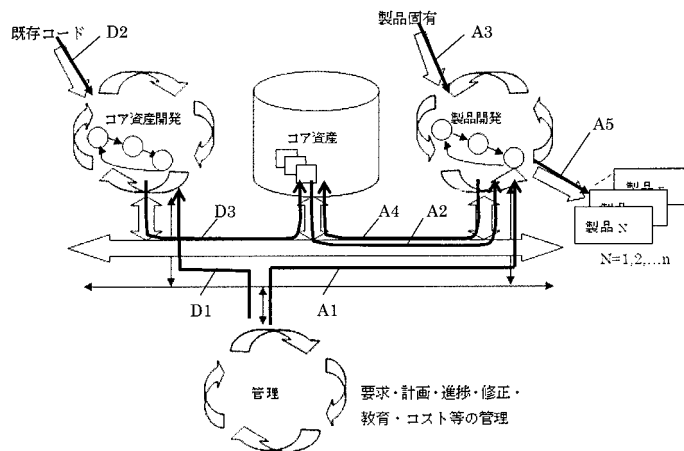


図5 プロダクトライン開発の構成

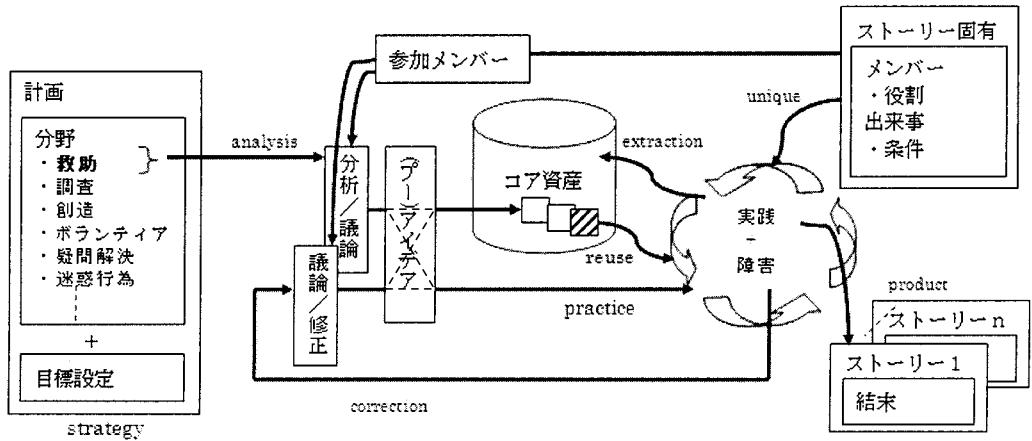


図6 「クマのプーさん」のコア資産モデル

3.1 コア資産開発

ここでは製品間の違いを明確にして再利用を実現するようなコア資産を開発する活動を行う。その中心にはKangら提案のフィーチャモデル (Feature Model) が使われる^[4]。フィーチャモデルの詳細な定義については省略するが、フィーチャモデルは4つのカテゴリ (能力、動作環境、ドメイン技術、実装技術) に領域分けがなされている。フィーチャモデルを考えるときの優先順位は、フィーチャ (特徴的な機能：ここでは能力の用語を使う) そのものが第一であり、次は動作環境という視点になる。

この考え方はモノづくりの基本形であり、フィーチャモデルはそれを視覚化した点で優れた表現モデルと言える。フィーチャモデルの表現を「プーが新しい遊戯を発明する話」に適用した例を図7に示す。プーの発明したプー棒投げは、森のみんなで橋の上から棒を落したあと下流側をながめた時に早く見えた棒を勝ちにする棒遊びである。この遊びのシステムは、プーの物語の「プー横丁にたった家」の中ではベストプラクティスに挙げられる。ちなみに「クマのプーさん」の中のベストプラクティスは「コブタを洪水から助ける話」になる。

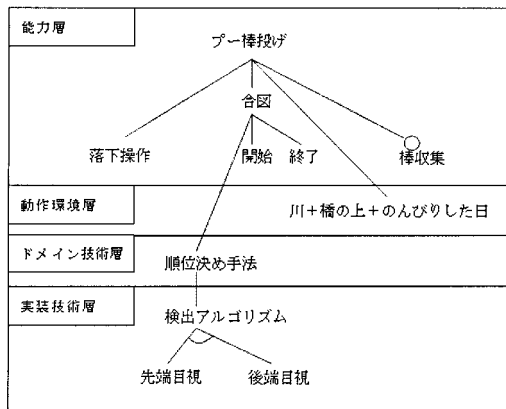


図7 プー棒投げのフィーチャモデル

3.2 コア資産

モノづくりにはその途中で問題発生が生じることがよくある。プーの物語においても例外ではない。問題発生に対して何をどのように対策したかは重要なポイントになる。この対策内容がコア資産として登録の対象になる。「クマのプーさん」の個所では問題発生に対して何も対策しなかった物語では結末は失敗に終わっている。唯一の例外は「イーヨーの誕生日にプレゼントする話」の所だけである。コア資産はSPL型開発において各種対策内容や再利用できるアイデアそのインプリメント内容、過去に開発されたものの中から可変性 (Variability) と共通性 (Commonality) を識別して蓄えておくことなどが要求される。そのためコア資産はリポジトリ (Repository) の役割があり、表記的にもデータベースの表記が使われる。図8にコア資産を形成する再利用可能な機能のかたまりや資産リポジトリそのものを示す。具体的にはプロダクトラインのプラットフォームとなるプロセスモデル、アーキテクチャ (パターンやフレームワーク)、コンポーネント、ツール、ドキュメントを挙げておく。さらに市販品をシステムに組み込んでいる場合にはCOTS (Commercial Off-The-Shelf) としてコア資産で管理される。

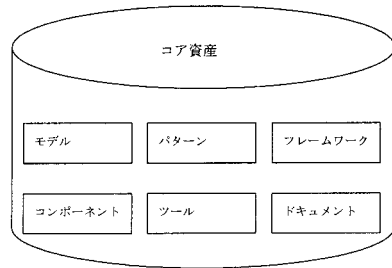


図8 プーに見るコア資産の構成要素

を開発することになる。開発プロセスにはアプリケーションエンジニアリングの趣旨が採り入れられるが、どのモデル (例えばウォーターフォールモデル、スパイラルモデルなど) で実施されるかは管理活動を担当している管理者の意向にも依存する。ここでは簡単に「コブタを洪水から助ける話」に製品開発の手法を実施した形を図9のようなマトリクスで表現してみる。

ストーリーは2ステージに分かれている。最初のステージではプーのもとにコブタからの手紙が流れてくるが、プーは字が読めないため風雨で危険であるにも関わらず勇敢にもロビンの所へつば船で行く。ここまでが第1解決になる。次のステージではプーとロビンがコブタを傘船で助けに行き第2解決となる。このステージの救助作業がコア資産を活用したり登録したりする状況となる。実際の運用ではドメインエンジニアリングにおける再利用箇所の分析 (例: フィーチャモデル)、コア資産への登録などは実施済の状態になっている。

3.3 製品開発

製品開発での活動は、コア資産を活用しながら各製品 (アプリケーションソフトウェア)

モデル	・・・	ウォーターフォールモデル、プロトタイプモデル、スパイラルモデル、XP
パターン	・・・	迷惑行為、救助、調査・検索
フレームワーク	・・・	雨、晴、雪、風、霧などの季節、川、山、日時
コンポーネント	・・・	ロビン、プー、ウサギ、フクロ、イーヨー、プー & コブタの各アイデア
ツール	・・・	ハチミツ、風船、傘、つば、ピン、穴、棒
ドキュメント	・・・	はり紙、手順書、回覧板、手紙、のし紙

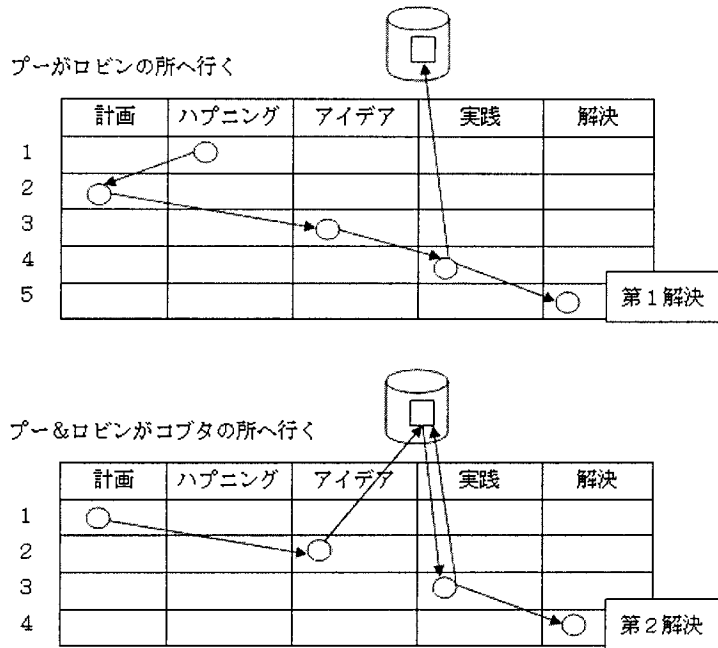


図9 アプリケーション(救助)の開発プロセス

4. 「クマのプーさん」のコストモデル

プロダクトラインの視点でプーの物語から導出したコストモデルは、7つのコスト要素で構成できる。ストーリー系列はプロダクトP₁、プロダクトP₂、・・・プロダクトP_nを想定し、個々のストーリーは、リポジトリに相当するコア資産と丸印のプロセスの中に実践

と障害を置いて表現する。

この丸印の個所が製品開発に該当する。一方、製品開発にコア資産の再利用を促すコア資産開発は、C_{strategy}とC_{analysis}のコスト要素からなるパラメータ入力の形で表現する。図10にプーのコストモデルを示す。このコストモデルは後のROI計算の所で利用する。

- C_{strategy} …… モノづくりの戦略コスト。どのようなテーマの分野を扱うかを決め、プロダクトが森のみんなに喜んでもらえるかどうかの市場調査コストも含む。
- C_{analysis} …… ストーリーをどのように組み立てるかを分析するコスト。コア資産開発コストのほか、コア資産へ登録できるかどうかを調べるコストも含む。
- C_{unique} …… ストーリーの固有部分を開発するコスト。
- C_{reuse} …… ストーリー展開のためコア資産を利用するコスト。
- C_{practice} …… ストーリー展開を滞りなく実践するコスト。
- C_{correction} …… 障害対策のため修正に要するコスト。
- C_{extraction} …… ストーリー展開の中からコア資産化できる部分を抽出するコスト。
- C_{usualway} …… モノづくりに従来型開発を採用した場合の製品コスト。

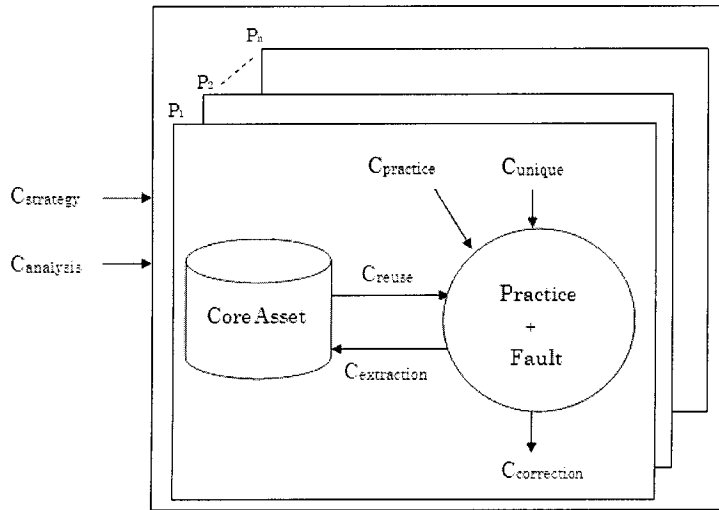


図10 プーのコストモデル構造

図10のコストモデルからプーの物語のプロダクトラインを構成するコストを下記のような式(1)で表現する。コストモデルに組み込んだ $C_{strategy}$ は、プロダクトラインを組織するコストでモノづくり戦略に絡む重要なコスト要素であるが、本稿では開発コスト計算に関する他のパラメータと同様のレベルで扱うことにする。

4.1 ROIの計算式

情報システムの投資効果は経営判断をする上で重要な指標であり、一般にはROI(費用対効果：Return On Investment)での評価で把握される。今日モノづくりで問題になっているシステムの高度で複雑な構成は、製品系列で勝負するメーカにとっては工事進行のリスク増大を伴う恐れがあるし、あるいは情報システムを導入して情報化経営を目指す企業にとっては見積もりコストの妥当性把握に障害

をもたらす要因にもなっている。現在、タイムリーなことに従来の商取引慣行を改める会計基準の変更が行われ、2009年4月から工事進行基準の導入が適用となりムリムダを排除する見積もりが求められ、見積もりコストの精度向上や明朗化を指向する環境が整ってきた状況にある。

従来型開発(非SPL型開発)でも再利用の手法にコンポーネントをベースにした作り方があるが、多品種向け製品開発で生ずるムリムダを解消する効率よい開発の実現には至っていない。一方SPL型開発は組織から分析、設計、インプリメントの仕方に至るまで新しい概念を導入するため少なからずハードルの高い開発手法になる。そのためSPL型開発の導入に対する費用対効果を予測することは投資タイミングを判断する上で重要な課題になる。企業の投資案件を評価するROIの計算式は式(2)で表現される。

$$\begin{aligned} \text{プロダクトラインを構成するコスト} &= C_{strategy} + C_{analysis} \\ &+ \sum_{i=1}^n (C_{unique}(P_i) + C_{reuse}(P_i) + C_{practice}(P_i) + C_{correction}(P_i) + C_{extraction}(P_i)) \end{aligned} \quad (1)$$

$$ROI = \frac{\text{利益}}{\text{コスト}} \quad (2)$$

式(2)で求めた値はパーセント扱いで表現する。SPLに用いるROI計算は式(2)を改良して用いる^[5]。分子の利益は、従来型開発コストからSPL型開発コストを差し引いた値で表現する。分母のコストは、 C_{strategy} と C_{analysis} を加算した投資コストで表現する。従来型開発コストは、 C_{usualway} のコスト要素を定義し、 $i=1 \cdots n$ 毎に製品コストを加算する形で表現する。以上によりROIの計算式は式(3)のとおりになる。

4.1.1 コストの数値化

プーをはじめとする森のメンバーがストーリーを組み立てるために要する日数とメンバー数を把握してコストを数値化する。森のメンバーは擬人化して取り扱う。

コストの数値化は、プーの物語を体制・工数・計画性・難易度・再利用度・信頼性・開

発モデルの各視点で数値に換算にする。さらに本例題ではストーリーの対象（救助）が4つあるため得られた数値の平均をとる。

次に基本コストの1.3PDを使い、コスト要素分けを行う。PD数値化された各値はスケール調整と重み付けを行い、ROI計算のために準備する。スケール調整と重み付けの例として C_{unique} では1/2（スケール調整）と0.8（重み付け）が掛け算されて0.52PDの値を得ている。このような数値化の流れを図11に示す。

例題：プーの物語で救助に関係する4つのストーリー（プーが身動き出来なくなる話、コブタを洪水から助ける話、トラーが木からおりられない話、フクロの家が倒壊する話）を考える。

前提：ストーリーの再利用度や構成の複雑さ、計画性等に関してはいずれの場合も同程度のレベルとする（平均化）。またストーリーの結末に要する時間は、3日で4人掛かるものとする。基本コストは1日につき1.3人（PD:Person-Days）の計算になり、それを1.3PD(PD数値化の値)で扱う。4つのストーリーでは5.2PDになる。

$$\frac{\sum_{i=1}^n C_{\text{usualway}} - (C_{\text{strategy}} + C_{\text{analysis}}) + \sum_{i=1}^n (C_{\text{unique}}(P_i) + C_{\text{reuse}}(P_i) + C_{\text{practice}}(P_i) + C_{\text{correction}}(P_i) + C_{\text{extraction}}(P_i))}{C_{\text{strategy}} + C_{\text{analysis}}} \quad (3)$$

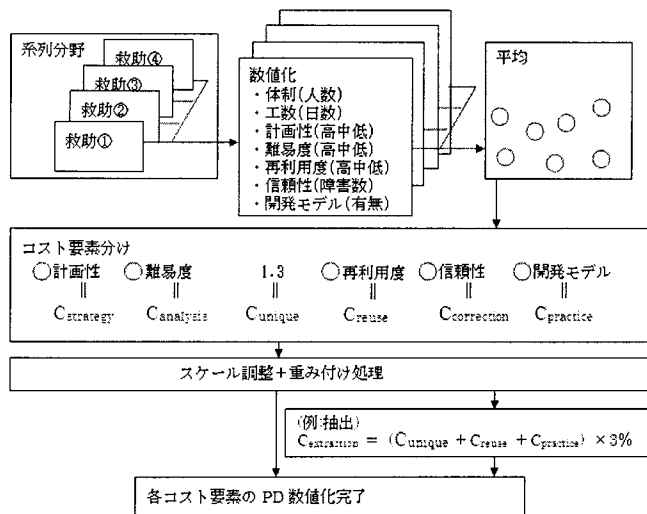


図11 PD数値化の流れ

得られたPD数値化の値は、次のとおりである。

$$C_{\text{strategy}} = 0.625\text{PD} \quad C_{\text{analysis}} = 0.6875\text{PD} \quad C_{\text{unique}} = 0.52\text{PD} \quad C_{\text{reuse}} = 0.125\text{PD}$$

$$C_{\text{practice}} = 0.025\text{PD} \quad C_{\text{correction}} = 0.075\text{PD} \quad C_{\text{extraction}} = 0.0645\text{PD}$$



$$C_{\text{strategy}} + C_{\text{analysis}} = 0.625\text{PD} + 0.6875\text{PD} = 1.3125\text{PD}$$

$$C_{\text{unique}} + C_{\text{reuse}} = 0.52\text{PD} + 0.125\text{PD} = 0.645\text{PD}$$

$$C_{\text{practice}} + C_{\text{correction}} + C_{\text{extraction}} = 0.025\text{PD} + 0.075\text{PD} + 0.0645\text{PD} = 0.1645\text{PD}$$

4.1.2 SPL型開発導入の視点

プーの物語の救助に関係するストーリーから導く従来型開発とSPL型開発のコスト比較は、どの時点でSPL型開発を採用すべきか否かのヒントを与えてくれる。物語から得られた数々のパラメータであっても現実の世界に適用できる内容があり、それらはコスト節約の検討にも利用することができる。SPL型開発の導入は一般に投資コストを伴う。この投資コストが、いくつかの製品系列を開発していくとある時点で従来型開発のコストを下回る形になり分岐点の存在が浮かび上がる^[6]。その分岐点はコスト要素の扱い方次第で操作することができる。ストーリー（製品）数がゼロの初期値において従来型開発に必要なコストは0PDであり、SPL型開発に必要なコストは1.3125PDである。以後ストーリー（製品）数が1、2、・・・と増えるにつれて図12のような傾向を持つグラフになる。例えばストーリー（製品）数=1の場合、従来型開発に必要なコストは1.3PD、SPL型開発に必要なコストは2.122PDになる。

従来型開発とSPL型開発のコスト比較を一般化して考えた場合、分岐点の個所が重要になる。後述するROIとの兼ね合いを考えると、3～4の製品数を分岐点に考えるのが望ましい。分岐点に関する正しい呼び名は、payoff point（利益獲得点）という。

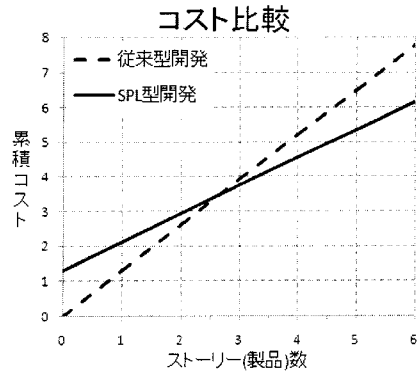


図12 従来型開発とSPL型開発のコスト比較 (救助)

4.2 ROI計算

PD数値化の値を式(1)に代入する。次に式(3)を使い従来型開発コストからSPL型開発コストの引き算を行う。最後に引き算で得られた値の利益を C_{strategy} と C_{analysis} を加算した投資コストで割り算する。解は小数点付きの値となるが、この値に100を掛けてROIの値にする。ここで求めた製品数4のROIの値は50% (=49.5%)である。

一般にROIの値が50%以上あるいは60%以上であればSPL型開発の採用を可とする判断ができる。このような手続きでプロダクトラインによる開発手法を採用すれば製品系列に関するシステム開発のコスト節約が実現できるようになる。

ストーリー（製品）数 = 4 の場合

$$0.625PD + 0.6875PD + 4(0.52PD + 0.125PD + 0.075PD + 0.025PD + 0.0645PD) = 4.5505$$

$$4 \times 1.3 - 4.5505 = 0.6495$$

$$0.6495 \div 1.3125 = 0.494857 \quad \therefore \text{ROI} = 50\%$$

ストーリー（製品）数の値を0～7まで変化させたときのROIの値の関係は図13のようなグラフで表現できる。

4.3 ROI考察

図13で表現のグラフは単調増加の一次関数である。この一次関数を近似式で表すと式(4)のとおりになる。

横軸を横切るポイントのbreakeven pointは $y=0$ のときであり、製品数の値は2.67になる。グラフを見ると製品系列数は3シリーズ分以降がSPL開発型の対象になることを示している。式(3)で考えると分子の前項と後項が等しい場合に導ける。つまり、「従来型開発による製品コスト」と「投資コスト+ α 」の値の一致した時点が損益なしのbreakeven pointになる。式(5)はbreakeven point条件を示している。

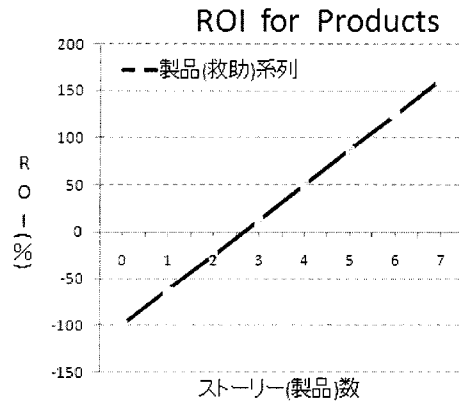


図13 製品数とROIの関係

$$y = \frac{3}{8}x - 1 \quad (4)$$

$$\sum_{i=1}^n C_{\text{usualway}} = (C_{\text{strategy}} + C_{\text{analysis}} + \sum_{i=1}^n (C_{\text{unique}}(P_i) + C_{\text{reuse}}(P_i) + C_{\text{practice}}(P_i) + C_{\text{correction}}(P_i) + C_{\text{extraction}}(P_i))) \quad (5)$$

1992年にJohn GaffneyとBob Cruikshankは、再利用する開発のスタイルでシステム開発したpayoff pointが1.67から4.86の間にある場合のROI計算式は、式(6)で扱えると提案している^[3]。(N=製品数、N0= payoff point)

$$\text{ROI} = \left(\frac{N}{N0} - 1 \right) \times 100\% \quad (6)$$

$$Y_{\text{ROI}} = \frac{1}{2.67584} X_N - 1 \quad (7)$$

ここでGaffneyらの式(6)と今回導いた式

(4)のROIを検証してみる。まず正確なpayoff pointを求めることから始める。図12のグラフの源データを用いると、SPL型開発は座標が始点(0,1.3125)と終点(10,9.4075)を通る直線になる。

一方、従来型開発は $y=1.3x$ の直線式で表現できる。以上から2つの線の交点を求めると(2.67584,3.47859)になる。従ってpayoff pointの値は $N0=2.67584$ になる。式(6)に $N0$ の値を代入して整頓する(計算上%は除外しておく)

と式(7)が得られる。

プーのコストモデルから導いた式(3)のROI、Gaffneyらが提唱した式(7)のROI、図13のグラフから近似した式(4)のROIを比較したグラフを図14に示す。近似式から得られた式でのROIが他の2つの方法で得られた値に比べて多少の違いが値上見られるが、ほぼ3種類とも同じROIの値の傾向となる。従って提案したコストモデルの妥当性が評価できたものとする。しかも従来型開発とSPL型開発のpayoff pointを正しく把握することが正確なROIを計算する上で重要であるとわかる。

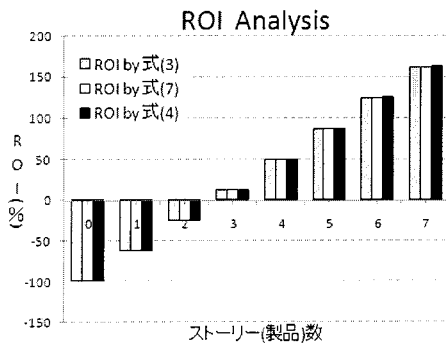


図14 3種類のROI分析

5. おわりに

モノづくりは、古代ギリシアの時代から追求されてきたテーマである。今回「クマのプーさん」の中にはいろいろな開発モデルが存在することを指摘し、これら開発モデルの存在からプロダクトライン開発での検討も可能だと仮定して論じてきた。実際に3.ではコア資産モデルを提示し、4.ではコストモデルを導出してROI計算に用いた。さらに従来型開発とSPL型開発の導入視点をROI計算で論じ、そこにはbreakeven pointなるファクタがあり、その値の正確な把握も重要であることを示した。

再利用がキーワードのプロダクトライン開発を導入するためには、経営者も技術者もそれぞれに新しい視点で組織化される必要がある。そこには投資コストや回収プランの評価、高度な技術基盤の確立、組織や技術面でのマネジメント力が求められる。当然、実務で実践されているプロダクトライン開発とは異なる局面もストーリーの中では存在したが、本稿がSPLを把握する上での一助となり、今後のプロダクトライン開発の取り組み・学習の進展に寄与することを期待したい。

参考文献

- [1] A.A.Milne, "Winnie-the-Pooh." Methuen, 1926. "クマのプーさん," 石井桃子 (訳), 岩波書店, Oct.1957.
- [2] A.A.Milne, "The House at Pooh Corner." Methuen, 1928. "プー横丁にたった家," 石井桃子 (訳), 岩波書店, Sep.1958.
- [3] P.C.Clements and L.M.Northrop, "Software Product Line: Practices and Patterns," Addison-Wesley, 2001. "ソフトウェアプロダクトライン," 前田卓雄 (訳), 日刊工業新聞社, pp.36-56, pp.274, 2003.
- [4] K.Lee, K.C. Kang and J.Lee, "Concepts and Guidelines of Feature Modeling for Product Line Software Engineering," ISCR7, LNCS, vol.2319, pp.62-77, Austin, U.S.A., Apr.2002.
- [5] G.Böckle, et al., "Calculating ROI for Software Product Lines," IEEE Software vol.21, no.3, pp.23-31, May.2004.
- [6] 岸他, "特集「ソフトウェア再利用の新しい波」," 情報処理, vol.50, no.4, pp.289-294, Apr.2009.