

組織的アセット再利用サイクルの推進アプローチ

三上 徹也 高橋 辰徳 中山 清喬

The Approach for Promoting the Cycle of Systematically Reusing Assets

Tetsuya Mikami, Yoshinori Takahashi and Kiyotaka Nakayama

アセットベースビジネスの実現に不可欠な再利用の試みは1968年までさかのぼることができるが、企業規模のアセット再利用は成功例に乏しい。本論文では企業における再利用の阻害要因を分析し、アセット生成および適用のサイクルを整理し、そのサイクルが機能する必要条件を定義した。次に、サイクルを現実的に運用するために、依存性注入 (Dependency Injection) の適用など、6つのプラクティスを導出した。それらを、Business Transformation Outsourcing (BTO) の一環としてお客様の環境に適用することにより、その有効性を確認した。

Reuse is indispensable in realizing asset-based businesses, and trials in reuse can be traced back as far as 1968. However, there have been very few success stories on enterprise-wide asset reuse. In this paper, we have analyzed the disincentives against enterprise-wide reuse, organized the cycle of asset production/consumption, and defined the conditions necessary for the cycle to operate. To make this cycle practical, we derived 6 practices, including the application of the technology, Dependency Injection. By applying these practices to a customer environment as part of BTO, we were able to demonstrate its potency.

Key Words & Phrases: アセット再利用, 依存性注入, ソフトウェア開発プロセス, コンポーネント粒度, 利害のモデル化
asset reuse, dependency injection, software development process, component granularity, stake modeling

1. はじめに

アセット再利用に対する取り組みは業界全体で本格化しつつあり、日本IBMにおいても組織全体の生産性と品質の向上を目的とする戦略的アセットベースビジネスを推進している。今日「アセット」という用語は横展開可能な様々な資産に対して用いられるが、特に利用者による加工・修正が容易な設計書・提案書テンプレートなどについては再利用の敷居も低く、組織的再利用の成果が現れはじめています。

一方、ソフトウェア開発におけるコンポーネント再利用のアイデアは、1968年にMcIlroyが発表した“Mass-Produced Software Components [1]”がルーツと考えられ、部品化を促進するオブジェクト指向の登場など、ソフトウェア開発の歴史の中で幾度となく注目されてきたにもかかわらず、組織的なアセット再利用に成功した例は乏しく、混沌とし、いまだ普及に至っていない [2] [3] (表1)。

表1. アセット再利用の歴史

西暦	出来事・取り組み
1968	McIlroy, 利用の基本概念を提唱 [1]
1972	オブジェクト指向言語Smalltalk発表
1989	Shlaer/Mellorオブジェクト指向設計論
1999	IEEE, 再利用部品生成標準を勧告 [4]
2004	IBM Rational®, アセットベース開発のためのプロセス (RUP for ABD) を発表 [5] Microsoft®, 再利用を主眼とする新生産方式 Software Factories を提唱 [6]
2005	OMG, アセット標準仕様RASを規定 [7]

本論文では、以下、2章で組織的再利用の阻害要因を分析し、3章で阻害要因を解決するための考慮点と実践手法を検討する。続く4章で、実践方法の適用事例に基づきその有効性を報告し、5章で得られた知見と今後の検討課題をまとめる。

2. 組織的再利用の課題

2.1 再利用の阻害要因

ロバート.L.ラスらは著書「ソフトウェア開発 55の

提出日: 2006年8月31日 再提出日: 2007年3月22日

真実と10のウソ【3】の中で大規模な再利用がいまだ実現していない事実を指摘するとともに、その阻害要因として、開発現場の『再利用部品を利用せず、自ら開発したがる傾向』を指摘している。

実際の開発プロジェクトでは、求める機能要件に整合する再利用部品はログ機能や認証機能などの小規模なものや、Strutsのような汎用的なものに限定される。また、パフォーマンスやセキュリティなどの非機能要件の整合を含めて考慮すると、既存の部品に対してカスタマイズや修正が必要となる場合は少なくない。

他者のプログラムを読解し、修正し、テストするメリットがコストとリスクに勝る分界点を、NASAゴダード宇宙飛行センターのソフトウェアエンジニアリング研究所が行った。この報告では、既存部品のわずか20%をカスタマイズする必要があるから開発したほうが有利であると結論付けている【8】。実際には、修正を行うと解除される保守契約、スキル人材の確保など、再利用部品の利用には多くの考慮を必要とする。

つまり、現状の多くのプロジェクトでは「再利用を行うコスト(労力・時間・リスク)にメリットが見合わない」ため、再利用ではなく独自開発を選択しているのである。

2.2 利害的観点の必要性

前述の現状が示すように、再利用の実現にはコストとメリット。つまり利害の考慮が不可欠である。従来の再利用に関する技術・プロセス・ツールは、個々の領域における理論上の再利用促進効果やメリットが定性的に注目される一方、再利用活動全体に求められる必要コストが併せて語られることはまれであった。

例えば、オブジェクト指向において再利用促進効果は一つの技術的特長に過ぎず、再利用実現に求められるコストと効果の最適バランスはプロジェクトや状況に左右される。そのため、再利用性にフォーカスした具体的なオブジェクト指向設計方法論などは整備が難しい。また、Object Management Group (OMG) が規定するReusable Asset Specification (RAS)【7】

やRational Unified Process® for Asset Based Development (RUP® for ABD)【5】などは、優れた再利用の基本アーキテクチャーやプロセスを示すことを目的としているのであって、実現性の担保を目的とする定量的評価手段や実践的プラクティスを提供するものではない。

著者らは、組織的再利用全体を包括する利害関係をモデル化することで組織的再利用の課題を定量的に明らかにし、根拠に基づく実現条件と実践手法を導く必要があると考えた。

2.3 組織的再利用の構造モデル

まず組織的再利用の全体像を明らかにするために、RASやRUP for ABDを総合し、アセットの生成と消費の流れの全体像を図1のような構造モデルとして整理する(以後、「統一再利用サイクル」と言う)。

アセット生産/消費の当事者としては、現場レベルの各プロジェクトと、全社レベルの企業全体(または企業全体を代表するアセット推進部門など)が考えられる。また、このモデルにおいては両者間を流通するソフトウェア部品を、アセットと成果物とに明確に区別する。成果物が個々のプロジェクトと顧客の要件に基づき現場レベルで開発される一方、アセットは全社レベルの要件に基づき整備されるものである。現場レベルの成果物は汎化作業を経てアセットになり、アセットは特化作業を経てプロジェクトにおける開発物に取り込まれ統合される。

このモデルは、組織的再利用という活動が現場レベルと全社レベルに跨る単一のサイクルを構成しており、そのどちらで停滞しても再利用サイクル全体が機能しないことを示している。

2.4 利害のモデル化

RUP for ABDでは、再利用に関する手続きを次の4つに定義している。

- ・生成：アセットの企画・開発・試験
- ・抽出：アセットの検索・発見・効果検証

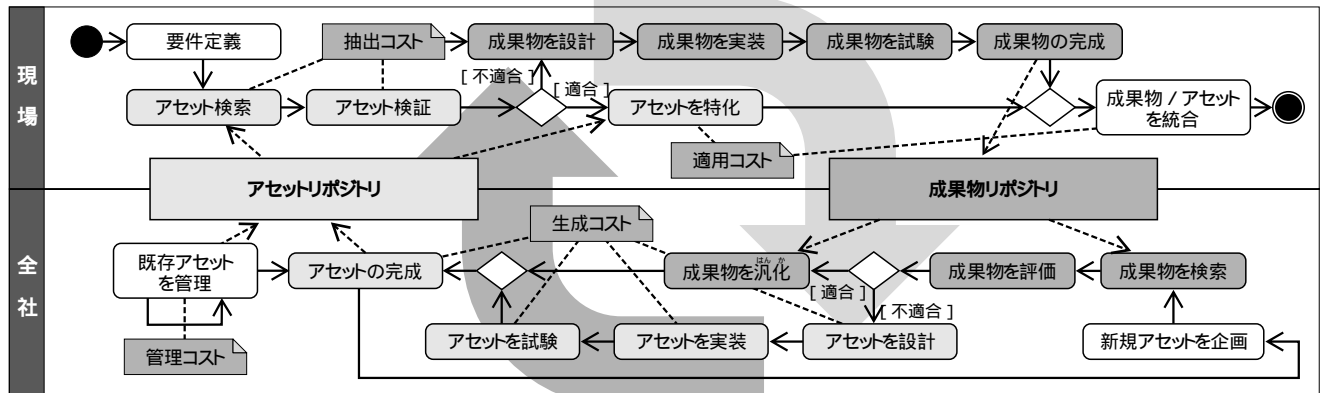


図1. 全社レベルと現場レベルを跨る統一再利用サイクル

- ・適用：アセットの取り込み・統合
 - ・管理：アセットの保存・公開
- これらそれぞれの活動に伴う労力・時間・リスクなどをコストに換算し、図1中に割り当てる。

統一再利用サイクル内におけるコストの明確化により、現場レベルと全社レベルの利害を以下のような数式で表すことができる。

$$\left\{ \begin{array}{l} \text{【現場利害条件】再利用のメリットが抽出・適用コストを上回る} \\ \text{削減可能開発コスト} > \text{抽出コスト} + \text{適用コスト} \quad (\text{A}) \\ \text{【全社利害条件】全現場のメリット総和がコストを上回る} \\ \text{削減可能開発コスト} > \text{生成コスト} + \text{管理コスト} \quad (\text{B}) \end{array} \right.$$

この連立不等式による利害モデル表現(以下、「連立利害条件」と言う)は、統一再利用サイクルが機能するための必要条件を表している。再利用は現場レベルと全社レベルに跨るため、その両式が満たされて初めてサイクル全体が機能する。

3. 組織的再利用実現へのアプローチ

連立利害条件から、再利用の推進に有効な2種類のアプローチを導くことができる。

- (1)不等式右辺(再利用の4コスト)を削減する。
- (2)不等式左辺(再利用回数と削減可能開発コスト)を増加させる。

つまり、このアプローチに基づき戦略的に設計されたプロセス、ツール、フレームワークや手法を活用することで、効果的に組織的再利用を促進することができる。

しかし、組織的再利用が普及していない現状は、このシンプルな2つのアプローチの運用が容易ではないことを裏付けている。以下、上記アプローチの実施に関する2つの考慮点を示した上で、具体的な実践手法を提案する。

3.1 連立利害条件の遵守

前述の両アプローチの実施以前の問題として、連立利害条件の不等式の遵守が必須である。特に、企業の全社レベルと現場レベルにおいて互いの利害を考慮せず再利用を試みると、連立利害条件の不等式自体が歪み、満たすことが困難となる点には特段の注意を要する。

例えば、生成コストは再利用コスト中で最も大きな負担だが、負担する主体が不明確か、暗黙的に現場に転嫁されている場合が少なくない(現場の成果物を一切汎化せずそのままアセットとする全社戦略。現場で良質の再利用部品を作る努力を行うプロジェクト)。全社レベルでは全社利害条件を容易に満たせるため組織的再利用を推進する一方、現場利害

条件を満たすことが極めて困難となる現場レベルではアセットを生成できない。

意識の高いプロジェクトでは全社横展開を意識した開発が検討されることもあるが、次の理由から現場では再利用を意識した開発をあえて行うべきではない。

- (1)再利用可能な部品を作るには通常の3倍の労力が必要と言われる[3]。つまり、再利用を意識した開発は、自社の都合のためにお客様に3倍の支払いを求めていることになる。
- (2)開発現場は、企業全体の再利用部品のニーズを把握していないため、再利用価値の低いアセット開発に過剰な投資を行う恐れがある。
- (3)プロジェクトの寿命はアセットの寿命より短く、アセットの利益や責任の大部分がプロジェクト終了後に発生する。プロジェクトは解散後までアセットを保守・管理できず、受け取れない未来の利益のためにプロジェクト期間中に生成コストを負担できない。全社レベルと現場レベルの双方で、再利用の単一サイクルにおける互いの利害の充足を目標とした負担と行動が求められる。

3.2 再利用コスト削減と回数増加

連立利害条件における変数において、管理コストと削減可能開発コストは考慮から外しても差し支えない。なぜなら、管理コストはアセットの保管や公開の仕組みを確立した後は低く抑えられ、一方の削減可能開発コストは、アセットが提供する機能によってほぼ確定するため故意に制御することが困難だからである。つまり、残る生成・抽出・適用コストの削減と再利用回数の増大が求められるアプローチとなる。

特に再利用回数については、全社レベルで①時間的に長期にわたり、②空間的に多数のプロジェクトで、再利用されるような戦略的な取り組みを行うことが必要となる。時間的・空間的な再利用機会の増大に関しては、一般に以下の2つの原則が知られている。

3.2.1 部分劣化原則

部分劣化原則は、以下の2つで説明される。

- (1)部品は、依存する部品と同一速度で劣化する。
- (2)部品は、複数の部品に依存する場合、最も劣化速度の速い部品と同一速度で劣化する。

例えばJava™に依存する部品は、Javaが使われなくなった時に使われなくなる。また、大きな部品ほど一部の劣化に伴い全体価値が減損するリスクが高い。一般にアセットは、全体価値の減損を防ぐために劣化した一部を修正または置換しながら利用され続ける。

価値が減損し全社利害条件を充足不能となったアセットをリポジトリから除去する手続きが規定されて

いるべきである。また、同式を満たす見込みがないものはアセット企画時に排除されるべきである。

なお、「有効なアセットに求められる最低限の再利用回数」が経験的に3回程度であることが、「ピガスタフの3の法則」「マジックナンバー・スリー」として複数の文献で紹介されており[3][9]、全社利害条件を継続的に充足する再利用回数の目安と考えることができる。

3.2.2 粒度背反原則

空間的再利用には、アセットの粒度が影響する。具体的には、粒度と再利用効果は比例し、粒度と適用可能な機会は反比例する。小さく汎用的なアセットは多くの利用プロジェクトのニーズにマッチするが、工数削減効果は小さい。一方、特化された大規模アセットは、ほぼ同一の限られたコンテキストでしか再利用できないが工数削減効果は大きい[10]。

3.3 再利用を推進する6つのプラクティス

統一再利用サイクル、連立利害条件、そして前項の考慮点に基づきアセット再利用サイクルを推進する実践手法として、6つのプラクティスを提起する(表2)。

表2. 6つのプラクティスと期待される効果

#	プラクティス	利用回数	コスト削減		
			生成	抽出	適用
1	既存成果物からの汎化	-		-	-
2	レイヤリングの実践	-*		-*	-
3	DIの適用	-*		-*	-
4	アセット属性による検索		-		
5	調整可能な部品粒度		-	-	-
6	アセット検索を含むプロセス		-		-

*プラクティス1との併用で効果を示す

3.3.1 既存成果物からの汎化

アセットを一から開発せず、既存成果物から汎化することで生成コストを抑えられる。

具体的には、全てのプロジェクトは成果物を成果物リポジトリに登録する。全社レベルでは玉石混交の中から再利用価値が高い成果物を選び、汎化作業後、アセットリポジトリに登録する。リポジトリを分離することで、再利用性が高いアセットが再利用性の低い成果物に埋もれる事態を避けることができる。結果、プロジェクトはアセットリポジトリから洗練されたアセットを効率良く取得できる。

3.3.2 レイヤリングの実践

ユーザーインターフェースなどの技術は日進月歩で進化するのに対し、ビジネスモデルは比較的变化が緩やかである。よって、後者に対応するドメイン層は前者に対応するプレゼンテーション層やインテグレーション層より安定である[11]。このプラクティスは、

層の責務やライフサイクルの違いを、特性の違いと捉え、成果物をその特性の違いに分類して作成することを推奨する。成果物の特性を特定することで汎化コストを、アセットの特性を明確化することで抽出コストを抑えることができる。

レイヤリングの実践により、特性による分業が可能になる。図2で示す業務による分類は、構造化開発やカード型開発などの業務単位での分業を示すが、全てのチームまたは担当者に多くの特性のスキルを要求することを意味する。一方でレイヤ別に分業することで、担当者に要求するスキルを特定することができる。

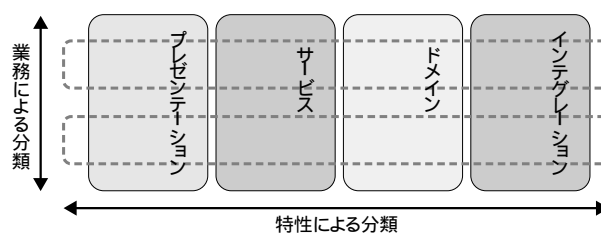


図2. レイヤリングと特性による分類

3.3.3 DIの適用

このプラクティスは、プロジェクトにおける成果物開発に依存性注入(Dependency Injection; DI)技術を用いることを推奨する。DIを用いることで成果物は容易に汎化可能となり、また既存のアセットを容易にプロジェクト開発物に統合可能となる。

DIとは、複数の部品を疎結合する技術である。2006年、国内でも大手都市銀行が相次いで採用し注目を集めているが[12]、DIのアイデアや機構自体は古くから優れた開発者の間で行われており、一過性の技術ではない。

表3は、従来手法とDIを利用した手法との比較である。従来手法でもオブジェクト指向により部品化がなされるが、最終成果物として部品を統合する過程で、個々の部品の中に他の部品の利用がハードコーディングされ密結合となる。一方、DIを用いた開発では、各部品は内部に依存を記述する代わりに、専用の構成ファイルに依存や部品の設定パラメータを記述する。稼働時、部品Xを利用する場合は、直接Xを取得せず、「DIコンテナ」と呼ばれるフレームワークにXの生成を依頼する。DIコンテナは、構成ファイルの記述に従い、まずYとZを生成し、Xに紐付け、完成したXを要求側に返す。

DIはアプリケーションをシンプルかつ柔軟にし、テストを容易にする道具として注目されている(表3-(I)列)[13]。しかし本論文は、DIを用いた開発成果物が「多くのプロジェクト固有要件を含む構成ファイル」と「固有要件が少ない分解済み部品群」という形態である点から、DIが再利用サイクルの生成および適用

表3. 従来手法とDIを利用した手法の比較

手法	成果物の形態 (X Y Zは部品)	(I)成果物開発		(II)アセット生成・再利用	
		Xのテスト	YをY'に交換	Xのみをアセット化	アセットWを統合しXから利用
従来		困難	困難	困難	困難
		Y Zも併せてテストする必要有	Yを利用するようXのソース修正が必要	Xのソースを修正しY Z利用部分の除去が必要	Wを利用するようXのソース修正が必要
DI利用		容易	容易	容易	容易
		ダミーY Zを利用するよう構成ファイルを修正	Yを利用するよう構成ファイルを修正	Xをそのままアセットとする	XがWを利用するよう構成ファイルに追記

表4. アセット属性と期待される効果

属性項目	期待される効果
機能概要	アセットを検索しやすくなり
キーワード	抽出コスト削減
適用対象レイヤ	
適用プロセスとタスク	
テストケース	
実績	
サポート体制	適用コスト削減
ガイドライン	
サンプル	

コストを大幅に削減する効果を持つことに新たに着目した(表3-(II)列)。従来の再利用サイクルはオブジェクト指向という「部品化技術」を持ちつつも、部品を再利用しやすく繋ぐ「結合技術」を欠いていたと考えられる。

3.3.4 アセット属性を用いた検索

全社利害条件が示すように、アセットを揃えたとしても、プロジェクトに利用されなければ再利用メリットを上げることはできない。

例えば表4に示すような属性をアセットに付加する。要件に適合する属性を持つアセットを検索する仕組みを提供することで、抽出および適用コストを下げることに繋がる。

3.3.5 調整可能な部品粒度

図3は、粒度背反原則に基づき部品粒度、削減可能開発コスト、および再利用機会の関係を示したものである。粒度を変えられない(不変粒度)小さなアセット()は、再利用による削減可能開発コストが低いが適用可能な機会は多い。一方、不変粒度の大きなアセット()は再利用の機会は少ないが多くの開発コストを削減できる。アセットの粒度は、アセット利用者であるプロジェクトの要件にアセットがどの程度適合するかに依存し、プロジェクトにより異なる。よって、アセットの再利用粒度を選択可能にし、再利用の総効果を増やすことを推奨する。具体的には、アセットを小さな部品が組み合わされた構造体として扱い、組み合わせ方を変えることで再利用粒度を調整可能とする。部分劣化原則に照らしても、このプラクティ

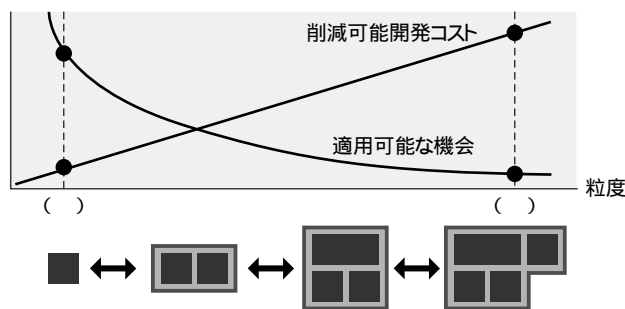


図3. 部品粒度と再利用効果の関係

スに従えば、部分劣化にも対応しやすく、長期にわたり再利用されやすくなる。

しかし、単に部品を小さく分解するだけではプロジェクトにおける抽出および適用コストが上昇する。そこで本論文では、開発プロセスのタスク単位で部品を予め組み合わせることを着想した。これは、Grant Larsenが提唱する、再利用レシピの概念[10]を応用したものである。Rational Unified Process (RUP)の分析/設計の「サブシステム的设计」を例にとると、設計モデル、設計サブシステムなどを作成する際に再利用可能なアセットとして、ステレオタイプ、コンポーネント、デザインパターン、フレームワーク、ガイドなどをまとめて再利用レシピとして用意する。

3.3.6 アセット検索を含む開発プロセス

プロジェクトが拠り所とする開発プロセスに、アセット検索のタイミングが明示されない開発者は判断できない。例えば、開発プロセスと連携可能なABDの適用事例が少ない理由としてRUPやADSG for e-businessにはABD利用のタイミングが明示されていない点が考えられる。

このプラクティスでは、各タスクやフェーズにおいて、実現すべき成果物の要件が決まった段階で、その要件を満たすアセットの検索するよう開発プロセスに明示を求める。

4. 実践と評価

著者らはBusiness Transformation Outsourcing(BTO)の一環として、お客様の環境で本論文提案手法を実

表5. お客様の開発業務改革への適用結果

プラクティス	実践内容	効果(コスト削減/再利用回数増加/その他)	
1. 既存成果物からの汎化	成果物rを元にアセットAを生成(図4-②).	生成	13.2人月(COCOMO-IIで見積算)から3.7人月に削減(72%削減)
2. レイヤリングの実践	4層モデル(Presentation, Service, Domain, Domain, Integration)を実践(図4-④).	生成	今回は,レイヤ化されていない成果物からの汎化のため,効果を示現せず.
		他	現場の開発期間を従来比54%削減.
3. DIの適用	各レイヤのコンポーネントと各アセットをDIコンテナにより統合(図4-④).	生成	今回は,DIを利用していない成果物からの汎化のため,効果を示現せず.
		適用	Bで使用されるDから派生されたDを生成.Bへの影響は設定Fileの修正のみ.
		他	成果物の複雑度が改善し品質向上.
4. アセット属性を用いた検索	ガイド,テストケース,サンプル,レイヤを属性として採用.また,統合にMaven[14]を利用(図4-④).	抽出	今回は属性による検索機構を整備しなかったため,効果を示現せず.
5. 調整可能な部品粒度	Bを一つのアセットではなく,CとDに分割(図4-③).	適用	依存する全アセットはビルド時に自動的に取得・統合され,適用コストを削減.
6. アセット検索を含む開発プロセス	OOプロセスを拡張し,再利用手順を明示(図4-④).	回数	B全体の再利用は粒度が大きく要件不適合な別案件にて,Dのみの再利用を実現.
		抽出	プロセスに従い開発を行うことで,再利用機会損失を回避できた.

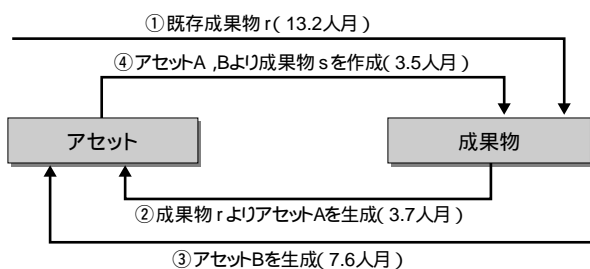


図4. 検証における再利用サイクルの実践

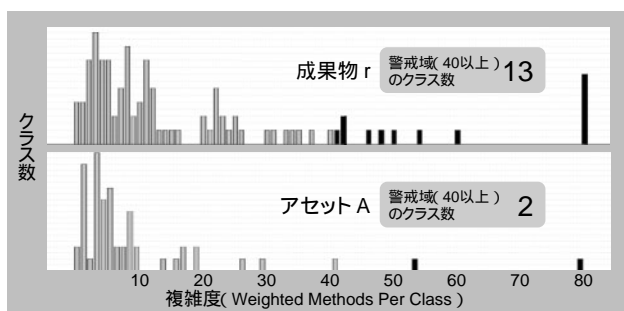


図5. 成果物rとアセットAの複雑度分布

践した。

現場レベルと全社レベルの利害を考慮の上,図4のようなサイクルでアセットの生成および成果物の作成を行った。

13.2人月の生成コストをかけた既存成果物rを元に,3.7人月の生成コストをかけアセットAを生成した。また,アセットAおよび別途開発したアセットBから成果物sを生成した(図4)。②③は全社レベルの活動,①④は現場レベルの活動であり,図1に対応した統一再利用サイクルを構成している。

表5の検証結果が示すように,各プラクティスは生成・抽出・適用の各コストおよび再利用回数の改善に効果を示し,アセット再利用サイクルの推進に有益であった。加えて,一部のプラクティスは現場レベルにおける成果物開発においても効果を示し,開発期間約54%,開発コスト約72%の削減を実現した。また,既存成果物rを提案プラクティスでアセットAに汎化した結果,図5のように複雑度の高いクラスが大幅に減少して,品質向上と保守の容易化も実現している。

最終的に,再利用サイクルが機能する必要条件である連立利害条件は以下の結果となり,現場レベルと全社レベルの双方の利害関係を満たした(アセットA,Bの再利用は多数計画されているが,3.2.1で述べた再利用最低回数の目安3回で試算した)。

$$\left\{ \begin{array}{l} 9.7 > 0.1 + 1.0 \quad (A') \\ 9.7 \times 3 > (3.7 + 7.6) + 0.0 \quad (B') \end{array} \right.$$

(単位:人月)

実際,その後も部品の開発が継続的に実施されており,次期プロジェクトからの成果物のアセット化も検討されている。

5. おわりに

お客様の開発業務改革に本論文提案手法を適用し,以下の知見を得た。

- (1)本論文提案のプラクティスは再利用コストの削減,再利用回数の増加に効果を示した。
- (2)特にDIコンテナの利用のように,現場レベル・全社レベル双方の利害充足に有効に働くプラクティスは双方に受け入れられやすく,相乗効果による高い効果が期待できる。
- (3)現場レベルと全社レベル,双方の利害を満たした結果,再利用サイクルが機能し,本論文提案のプラクティスと基盤をなす原則の妥当性が示された。一方,本論文提案のアプローチの効果的実践には,以下のような考慮も必要となる。
 - (1)各コストに関して厳密な実績調査を試みると,調査コストが増大し再利用サイクルを阻害することがある
 - (2)拡張開発プロセスに基づく開発体制の確立や,比較的新しいDI技術の習得のため,現場レベルに学習コストが必要な場合がある
上記を踏まえ,さらに多数のプロジェクトにおける適用を経て,本論文のアプローチをさらに改良していくことを検討している。特に,本論文では主にコンポー

ネット粒度の部品を議題としたが、今後Webサービスなどのより大きな粒度での再利用への応用や、永田らが示したパターン・UML拡張仕様・プロセスなどのSOA関連アセット[15]との関連を含め、更なる検討を深めていきたいと考える。

謝辞

本論文提案手法の実践は、日本IBM製造セクターにおける支援活動として行われた。この活動に携わった、エンジニア、マネージメントの各位、特に、高塚誠一氏、山口幹彦氏、柳一弘氏、長井昭史氏、小島慶子氏の協力とサポートに深謝する。また、多くの助言を頂いた西原裕善氏、杉本宣男氏に感謝する。

参考文献

- [1] M.D.McIlroy: "Mass-Produced Software Components," *Nato Science Committee* (1968)
- [2] 田中 淳: "ソフトウェア部品化/再利用を阻む壁," <http://itpro.nikkeibp.co.jp/free/ITPro/OPINION/20030918/1/> (2003.9.18).
- [3] ロバート.L. グラス: "ソフトウェア開発 55の真実と10のウソ," 日経BP社, ISBN4-8222-8190-6 (2004).
- [4] IEEE: "Software Lifecycle Process-Reuse Processes," *IEEE Standard 1517 "Standard for Information Technology"*, IEEE (1999).
- [5] IBM Corp.: "RUP for Asset-Based Development Plug-In v1.0," <http://www-128.ibm.com/developerworks/rational/library/4145.html> (2004.5.12).
- [6] Jack Greenfield et al.: "Software Factories," Wiley, ISBN0-4712-0284-1 (2004).
- [7] Object Management Group: "Reusable Asset Specification, Version 2.2," <http://www.omg.org/technology/documents/formal/ras.htm> (2005.11).
- [8] McGarry et al: "An Approach to Software Cost Estimation," NASA Software Engineering Laboratory, SEL-83-001 (1984).
- [9] スコットW. アンブラー他: "エンタープライズ統一プロセス," 翔泳社, ISBN 4-7981-0934-7 (2006).
- [10] Grant Larsen: "Using Assets To Tame Complexity," *ASWEC2004* (2004)
- [11] ロバート.C. マーチン "アジャイルソフトウェア開発の奥義," ソフトバンク, ISBN 4-7973-2336-1 (2004).
- [12] "UFJ銀などが相次ぎ「DIコンテナ」採用," 日経コンピュータ 2006年8月7日号, 日経BP社 (2006).
- [13] 須賀幸次 他: "Seasar入門 はじめてのDI&AOP," ソフトバンク, ISBN4-7973-3196-8 (2006).
- [14] The Apache Foundation: "Maven2," <http://maven.apache.org> (2006).

- [15] 永田葉子他: "ラショナル統一プロセスによる実践的SOA構築手法," *ProVISION*, No.51, pp.73-79 (2006).



日本アイ・ビー・エム
システムズ・エンジニアリング株式会社
BIS ビジネスインテグレーション部
ITスペシャリスト

三上 徹也 Tetsuya Mikami

[プロフィール]

オブジェクト指向、ソフトウェア再利用、RationalをスペシャリティとするITスペシャリスト。オブジェクト指向デザイン・アーキテクチャー、開発プロセスの構築・標準化をメイン・タスクとする。Rational Unified Processを適用したRationalソリューションの構築、組織的ソフトウェア再利用にも力を入れている。
tmikami@jp.ibm.com



日本アイ・ビー・エム
システムズ・エンジニアリング株式会社
BIS ビジネスインテグレーション部
ITスペシャリスト

高橋 辰徳 Yoshinori Takahashi

[プロフィール]

2000年IBM入社。Webサービス、Gridなど先進技術展開の担当を経て、現在はSOA、再利用を担当するITスペシャリスト。さまざまなお客様のプロジェクトにて、アーキテクチャー策定、フレームワーク設計、先進技術、プロセス策定の支援を行っている。サービス再利用の実践を目指し、SOAの推進に努めている。
yoshitk@jp.ibm.com



日本アイ・ビー・エム
システムズ・エンジニアリング株式会社
AIS テクノロジーイノベーション部
ITスペシャリスト

中山 清喬 Kiyotaka Nakayama

[プロフィール]

オープン・ソース関連技術のITスペシャリスト。入社以来、大規模基幹プロジェクトやSier戦略組織において標準化やフレームワーク設計支援に携わる。現在はIBM内オープン・テクノロジー戦略推進チームの一員として、新興先進技術とIBMコンピテンシーとを融合させた高付加価値サービスの確立・普及に努めている。
knakayam@jp.ibm.com