プロダクトラインのスコープ定義における カバー率と対応度によるコアアセット可視化手法

位野木 万里[†1,†2] 深澤 良彰[†2]

プロダクトライン型ソフトウェア開発において，コアアセットの開発範囲を決定するスコープ定義は，プロダクトラインへの投資効果に影響を与える重要なタスクである．複数のプロダクトラインを有する企業は，スコープ定義の結果に基づき，投資先のプロダクトラインを適切に判断する必要がある．本稿では，コアアセットの範囲を計測するメトリクスである，カバー率と対応度を定義し，これらのメトリクスによりコアアセットのスコープを可視化する手法を提案する．本手法によりコアアセットのスコープと企業内の位置付けを明確にできる．

Product Line Scoping Method: Core Asset Visualization Based on Level of Coverage and Consistency

Mari Inoki[†1,†2] and Yoshiaki Fukazawa[†2]

Core asset scoping is an essential activity that provides economic impact on product line driven software development. In the enterprise that has several product lines, it is necessary to judge in which product line to invest appropriately by using the result of the scope definition. In this paper, we propose a core asset scoping method on the basis of two metrics: the levels of coverage and consistency of core assets. The method makes the scope of the core assets and the positioning of individual product lines in the enterprise visible.

1. はじめに

1.1 プロダクトライン型ソフトウェア開発におけるスコープ定義の重要性

ソフトウェア開発の生産性の向上，品質の安定，開発のリードタイムの短縮には，品質の保証されたソフトウェア部品を蓄積し，これらの再利用によりソフトウェアを開発することが有効である[1)]．再利用の対象は，プログラム，モジュールから，コンポーネント，フレームワーク等へと変化してきた．このような変化の中で，同一ドメインの製品開発に限定するものの，様々な種類の成果物を再利用の対象とすることで，再利用の効果を向上させるプロダクトライン型のアプローチが注目されている[2),3)]．このような考え方に基づき，ソフトウェアプロダクトライン構築の手法ならびに構築事例が報告されている[4),5)]．プロダクトライン型の開発では，製品開発のロードマップを描き，対象製品を開発するためのコアアセットと呼ばれるソフトウェア開発の資産をあらかじめ整備し，これらのコアアセットを利用し製品を開発する[6)]．

プロダクトライン型の開発における経済的な効果は，あらかじめ用意しておくコアアセットの開発および維持管理のための投資と，コアアセットの再利用による製品開発のコストとの差分等により判断する[7)]．経済的効果の向上には適切な投資が不可欠であり，そのためには，プロダクトラインのスコープ定義が重要である[6)]．スコープ定義では，プロダクトライン化する事業領域，対象とする製品群の特性と製品系列，ならびに，各製品系列に対応する製品を開発する際に再利用するコアアセットに何を含め，何を含めないかを決めることになる．コアアセットのスコープが広すぎれば，製品開発で使われないコアアセットを開発することになり，余分なコストが発生し，投資回収に影響が出る．スコープが狭すぎると，複数の製品間に重複する機能であっても，コアアセット化されていないために，製品開発ごとに開発することになり，無駄なコストが発生する[8)]．

現在までに，スコープ定義のための様々な手法が提案されている．Schmid は約 30 のスコープ定義手法を取り上げそれらの特徴を明らかにした[9)]．たとえば，スコープ定義の代表的な手法には，PuLSE-Eco[10)] や，

†1 東芝ソリューション株式会社
Toshiba Solutions Corporation
†2 早稲田大学
Waseda University

オブジェクト指向によるスコープ定義手法[11] がある．しかし，プロダクトラインの対象市場，利用技術，開発組織が様々な状況や条件にある中で，あらゆる観点からスコープ定義が可能な手法が提供されている状態には至っていない．

1.2 企業におけるスコープ定義の課題

我々は，プロダクトラインの開発計画と投資実行を決定する実例において，次のような課題に直面したが，これらは現状のスコープ定義の手法では明らかにされていない．

課題 1：ドメインによらずプロダクトラインを評価する手段がない．

複数のプロダクトラインを有する企業では，組織全体の最適化を考慮して，どのプロダクトラインへ投資するかを判断する必要がある．そのためには，製品の特性，利用技術，組織の条件に依存せず，組織の中でプロダクトラインを比較することが重要であるが，現状はそのような技術は明らかにされていない．

課題 2：分かりやすく簡便なスコープ定義手法が提供されていない．

プロダクトラインのスコープ定義において，経営トップに限らず，企画営業部門，開発部門等の様々な役割の関係者からの意見を取り入れることはリスク回避のためにも重要である．

たとえば，経営者は，事業の戦略との合致性，成長性，投資回収率等を中心に投資先を判断する．コアアセットとして，フレームワークやコンポーネント等のバイナリ成果物のみを蓄積するとコストは低く抑えられるが，業務ノウハウや設計の考え方を示した業務仕様や設計仕様を蓄積しておかないと，製品開発時のカスタマイズや仕様変更対応等で設計理解がしにくく，コアアセットの維持管理が困難になる．しかし業務仕様や設計仕様等の利益に直結しない成果物の開発の必要性は，経営幹部には伝わりにくい．

コアアセットがどのような状況にあるかは利害関係者で共有すべきであるが，対象プロダクトラインへの投資が決定していない状況で，見積り作業に多大なコストを費やすことは困難である．様々な利害関係者がスコープを把握しやすく，かつ，簡便に鳥瞰できる手法が必要であるが，現状のスコープ定義手法では，そのような点は考慮されていない．

課題 3：コアアセット全体の品質が考慮されていない．

あるドメインに対して，分析モデル，設計モデル，テストモデル，実装モデルが揃っており，テストモデルに基づいて，各モデルが検証済みであるコアアセットと，それぞれの要素は存在しているものの，テストモデルでの検証が行われていないコアアセットでは，前者の方は検証が完了している分，開発コストは高い．後者のコアアセットは，製品開発での再利用時や，維持管理の段階で，品質問題を起こすリスクがある．

コアアセットの投資コストには，開発量の視点だけではなく，コアアセットの要素間での対応関係や検証関係を保証するかどうかという，コアアセット全体の品質を確保するためのコストも必要である．しかし，現状のスコープ定義の研究では，コアアセットの量と質を利害関係者間で把握するためのメトリクスは定義されていない．

1.3 研究のアプローチ

本研究では 1.2 節であげた課題を解決するために，次のアプローチでスコープ定義のメトリクスおよびスコープ定義プロセスを定義する．とくに，開発者にしか分からなかったコアアセットの要素の品揃えや品質の状態を可視化し，利害関係者間での共通理解を促進し，適切なプロダクトラインの開発計画の立案に貢献する．可視化とは，個人の暗黙知となっている知識や考え方を企業内で共有可能な情報に変換することと定義する．本稿では，そのようなメトリクスとしてカバー率と対応度を提案する．

【本稿で提案するメトリクスの特性と考え方】

- 開発者にしか分からなかったコアアセットの要素の品揃えや品質面の観点からスコープを計測するメトリクスとしてカバー率と対応度を提案する．
- 複数の異なるドメインのプロダクトラインの比較に利用するため，メトリクスは，利用技術，製品特性，組織に固有の条件に依存しないものとする．
- 簡便な方法で計測でき，プロダクトラインの位置付けを利害関係者間で把握可能にする．

2. コアアセットのスコープ定義手法

2.1 スコープ定義プロセス

本稿では，プロダクトラインの開発計画と投資実行に関して，次のような状況を想定している．

- 企業は複数のプロダクトラインを保持または開発予定である．
- 経営層は，期単位で，新規も含めプロダクトラインの開発計画を審査し投資を決定する．
- 経営層の配下にあるビジネスユニット（以下 BU と呼ぶ）がプロダクトライン開発計画を立案し，経営層へ申請する．BU とはある特定の業務・業種分野の事業を行う組織単位とする．プロダクトラインは BU によって構築される．
- BU はプロダクトラインの開発計画を立案する．

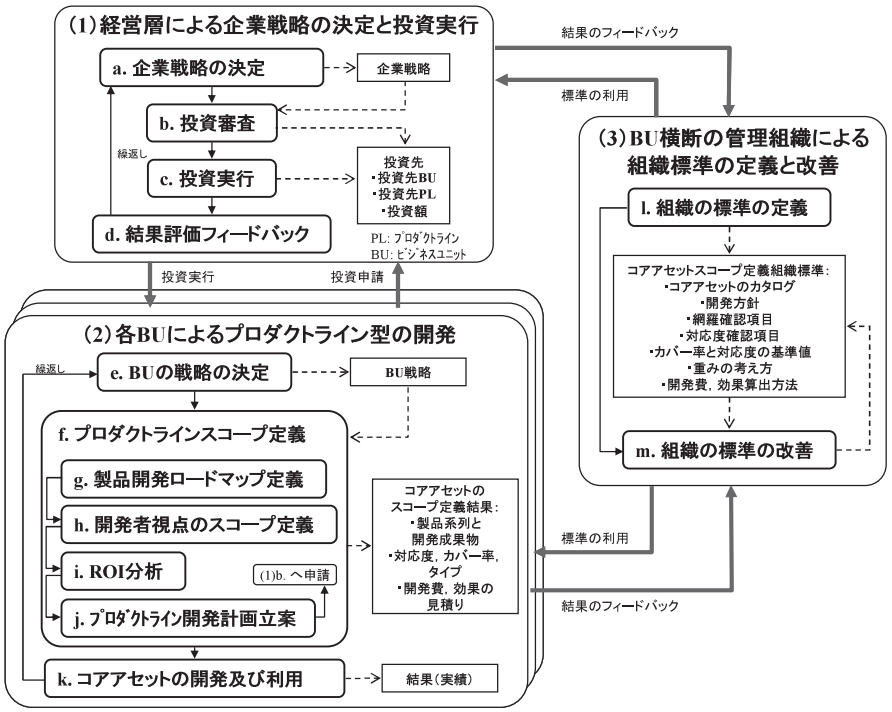

図 1 コアアセットのスコープ定義プロセス
Fig. 1 Processes for scoping core assets.

経営層による審査および投資実行の決定後，BU はコアアセットの構築および維持管理を行う．

上記の想定のもと，コアアセットスコープ定義プロセスを**図 1** のように定義する．コアアセットのスコープ定義は 3 つのプロセスで構成する．1 つめは経営層が，投資すべきプロダクトラインを決定するプロセスである（図 1 (1)）．2 つめは，各 BU によるプロダクトライン型の開発のプロセスである（図 1 (2)）．最後は，BU に横断的な管理組織が，コアアセットとなる成果物の種類，後述するメトリクスの基準値，算出方法の重み，各種確認項目を定義し，組織の標準として共有可能にするプロセスである（図 1 (3)）．

以下，コアアセットのスコープ定義プロセスの詳細を示す．

(1) 経営層による企業戦略の決定と投資実行

a. 企業戦略の決定

企業の目指す姿とその実現のための戦略を決定しロードマップ，各種指標，目標値を定義する．

b. 投資審査

各 BU からの申請に基づき，a. の達成の観点で投資先を審査する．

c. 投資実行

b. の審査結果に基づき，投資を実行する．

d. 結果評価フィードバック

投資先である事業の進捗と各種指標の達成度を定期的に確認し，必要に応じ投資先や投資額を見直す．

(2) 各 BU によるプロダクトライン型の開発

e. BU の戦略の決定

企業全体の戦略に基づき BU の目指す姿，当該 BU がフォーカスするドメインとロードマップ，各種指標と目標値を定義する．

f. プロダクトラインスコープ定義

BU の戦略に基づき次の g.〜j. によってプロダクトラインのスコープを定義する．

g. 製品開発ロードマップ定義

対象とするドメインの分析を行い，製品系列を定義する．各系列の開発ロードマップを定義し，製品系列の開発順序を定義する．後述するプロダクトラインのタイプ目標を設定する．タイプの定義は 2.3 節で述べる．

h. 開発者視点のスコープ定義

本稿で提案するメトリクスを用いてコアアセットのスコープを定義する．以下に詳細を示す．

- 組織で定義したコアアセットの成果物のカタログからフォーカスする部分を選択する．
- 上記選択結果に基づき成果物を選択し，コアアセッ

トをどのような成果物で実現するかを定義する．

- メトリクスの計測：本稿で提案する「カバー率」と「対応度」を算出する．メトリクスの定義は 3，4 章で述べる：

  [カバー率の計測]：組織の標準で定めた網羅項目を利用して，対象コアアセットが満たす項目を確認する．問題領域のカバー率と，解決領域のカバー率を見積もり，コアアセットのカバー率を算出する．

  [対応度の計測]：組織の標準で定めた対応度確認項目を利用して，対象コアアセットが満たす項目を確認する．問題領域の観点からの対応度と，解決領域の観点からのカバー率を見積もり，コアアセットの対応度を算出する．
- タイプの導出：組織の標準で定めた基準値と比較し算出したカバー率と対応度から，プロダクトラインのタイプを導出する．設定したタイプ目標に合致しない場合には，実現する成果物の要素，対応度等を調整する．タイプ目標と合致するまで繰り返す．タイプの定義は 2.3 節で示す．

i. ROI 分析

製品系列を実現するためのコアアセットの成果物に対して，開発コストを見積もる．また，このコアアセットによる再利用効果を算出する．見積り結果に対して，投資対効果のバランスを検討し，目標に合致していることを確認する．必要に応じ h. に戻り i. で妥当となるまで繰り返す．

j. プロダクトライン開発計画立案

f. の定義結果を経営層へ申請する．投資審査によって投資実行が確定した場合，詳細なコアアセットスコープ定義を行い，計画実行の承認を経営層より得る．また，開発費用に変更があった場合等必要に応じ経営層に再度申請し投資実行の計画実行の承認を受ける．

k. コアアセットの開発および利用

詳細な開発計画の承認が得られた後，コアアセットの開発および，コアアセットの利用による製品開発を行う．

(3) BU 横断の管理組織による組織標準の定義と改善

組織の標準の定義と，実際の適用結果からのフィードバックと改善のサイクルを繰り返すことで，スコープ定義結果を最適に保つ．各 BU での経験やノウハウは，組織の標準として共有するとともに，継続的な改善により最適な標準を維持する．

l. 組織の標準の定義

コアアセットに含める成果物の種類のカタログ，コアアセットの開発方針，カバー率，対応度の基準値，算出式の重み，網羅確認項目や対応度確認項目，ROI 算出方式を定める．コアアセットに含める成果物の種類のカタログでは，たとえば付録に示す**表 3** の左 3 列分に示すような一覧を定義する．なお，コアアセットの開発方針とは，たとえば，組織は業務・業種のドメインの知識継承に重点を置くのであれば，役割軸では，ビジネスオブジェクトや問題分析モデル等の要素を蓄積すべきであるし，短期的な製品開発のコスト削減が重点事項であるならば，バイナリの成果物を重点的に作成するといった方針が考えられる．

カバー率，対応度の基準値とこれら算出式の重みは，組織内での議論によって定義する．重みを調整する（たとえば 1 より小さくする）ことは，算出対象の全体に対する影響度を小さくすることに相当する．算出対象が，アプリケーション開発のコストや品質に影響しないと判断できる場合に，重みを 1 より小さく設定する．

定義した結果は，組織の標準としてドキュメント化する．投資対効果の算出式も定義する．また標準を改善するタイミング（半期に 1 度や，製品開発のつど等）を定義する．

m. 組織の標準の改善

各 BU によるプロダクトラインのスコープ定義と実際の開発結果，製品開発での利用を通して得られた結果，経営層からの評価を分析し，組織標準を改善する．

たとえば，次のような改善が考えられる．

- 製品開発のつど開発することになった取扱い説明書を，コアアセットとするべき成果物の種類としてカタログに掲載する．
- 品質により重視したコアアセット作りを目指すため，品質に関するメトリクス（本稿では対応度）の基準値を引き上げる（たとえば，0.5 から 0.75 にする）．
- 組織全体で製品系列数が非常に少なく，製品系列の開発進捗を投資審査では重視しないため，網羅度（本稿ではカバー率）の重みを低く設定する．

2.2 メトリクス

プロダクトラインのスコープを定めるメトリクスとして，量を計測するカバー率と，質を計測する対応度を定義する．

2.2.1 カバー率

カバー率とは，コアアセットがどの程度の成果物を備えているかという，コアアセットの充実の度合いを測るメトリクスとする．カバー率の実測方法は 3 章で述べる．

コアアセットはあらゆる成果物が対象になる．しかし，現実のプロダクトライン開発では，投資対効果を

考慮し，組織のねらいやプロジェクトの条件によって作成するコアアセットを調整する．このような充実の度合いを計測できるメトリクスは重要である．カバー率が大きければ，そのコアアセットは内容が豊富で充実しているととらえることができる．

2.2.2 対 応 度

対応度とは，コアアセットの成果物間の整合性の度合いを示す，コアアセット全体の質を測るメトリクスとする．整合性の度合いとは，対象となるコアアセットの任意の要素間において，矛盾がないことの程度を示す．対応度の実測方法は，4 章で述べる．

たとえば，注文管理ドメインにおいて，注文管理設計モデル，注文管理コンポーネントを含む下記の 2 つのコアアセットでは，ケース 2 の整合性が高くなる．

ケース 1：注文管理設計モデルに基づいて，注文管理コンポーネントの検証が完了していない．たとえば，コンポーネントのある例外処理が，設計モデル上では実際の振舞いとは異なる仕様で記述されている．

ケース 2：注文管理設計モデルに基づいて，注文管理コンポーネントの検証が完了している．すなわち，注文管理コンポーネントは，設計モデルに定義された振舞いと構造に則り実装されており，注文管理コンポーネントの実際の振舞いは，注文管理設計モデルに記述された振舞いの仕様と一致している．

コアアセットの整合性は高い状態にしておく方が製品の品質の安定に貢献するが，そのためには高コストを要する．どの程度までコアアセット間の整合性を高めるかについては，組織のねらいやプロジェクトの条件により異なる．よって，そのようなコアアセットの整合の度合いを調整するメトリクスは重要である．対応度が高ければ，そのコアアセットは整合性が高く，再利用の結果，安定した品質が期待できる．

2.3 コアアセットのタイプによる組織内の位置付け定義

カバー率と対応度に対して，それぞれ基準値を設定すると，コアアセットの要素の構成方法は**表 1** に示す 4 つのタイプに分類できる．各プロダクトラインをタイプに分類しておくことで，組織全体のプロダクトラインの位置付けを把握し，プロダクトラインへの投資判断に利用できる．

(1) タイプ A：上流から下流までの一連の成果物を備え，各々整合がとれた状態．投資に対し効果が期待でき，組織の強みとなるドメイン向けのプロダクトラインが該当する．

(2) タイプ B：上流から下流までの一連の成果物を備えるが，成果物間の整理統合が不完全な状態．すでに複数のアプリケーションの開発経験があり，様々な成果物が蓄積されているが，製品の利益率が低い等の理由により，プロダクトライン構築への投資の決定が困難なドメイン向けのプロダクトラインが該当する．

(3) タイプ C：作成する成果物は限定するものの，成果物間の整合がとれている状態．新規参入スピードが必要なドメインに対するプロダクトラインが該当する．

(4) タイプ D：最低限のコンポーネントまたは設計仕様等が蓄積されているだけの状態．新規参入で，組織の強みとするかどうか，不確定要素の多いドメイン向けのプロダクトラインが該当する．

表 1 コアアセットのタイプ
Table 1 Types of core assets.

基準値との比較で2つに分類	対応度 ＼ カバー率	成果物間の対応がとれている	成果物間の対応は一部のみ
基準値との比較で2つに分類	コアアセットは充実している	タイプA	タイプB
	コアアセットは充実していない	タイプC	タイプD

基準値は，まずは中間値（0.5）を設定し，組織での実績を蓄積することで改善していくことが望ましい．対応度，カバー率の基準値を 0.5 とし，対象組織では，ほとんどのプロダクトラインがタイプ A と判定される場合には，プロダクトライン間で差異が明確とならないため，たとえばカバー率の基準値を 0.75 に引き上げる等の調整・改善を行う．本稿で提案する手法の導入開始時期では，基準値を 0.5 として出発し，組織全体で所有するプロダクトライン群の値を比較し，実際の開発結果の実績を評価することを通して，基準値を改善する．

2.4 プロダクトラインの開発計画における本手法の位置付け

前述したように，本稿で提案するスコープ定義手法は，企業がどのプロダクトラインに投資するかを決定する際に適用する．カバー率，対応度，タイプは，開発者がプロダクトラインの開発計画を立案する際に導出する．算出されたカバー率，対応度，タイプは，経営層の投資審査において利用する．各 BU におけるプロダクトラインの開発計画は，カバー率，対応度，タイプだけで立案するわけではない．経営層による投資実行が決定した後，詳細なプロダクトライン開発計画を立案する．たとえば，詳細なスコープ定義の例としては，PuLSE-Eco [10] がある．このような手法により，製品系列の定義，製品系列と製品特性の対応付け，各

製品特性と実現するコンポーネントの対応付け，コンポーネントの開発コスト見積りと優先度定義を行うものとする．

2.5 ドメインのとらえ方と組織の前提条件

プロダクトラインが対象とするドメイン（領域）は，問題領域と解決領域に分けられる．

問題領域は対象とする製品系列の性質を示すFeatureモデルで表現することが一般的である．Featureは，様々な利害関係者（ユーザ，開発者，マーケティング等）の視点で表現したドメインの特性であり，Featureを階層表現で記述したFeatureモデルにより，ドメインの共通/可変性を明らかにする[12]．このようなドメイン分析と市場分析を経て，製品系列を定義する．ここで製品系列とは，共通部分を共有する一連の製品群の総称と定義する．製品とは，特定の顧客向け等に製造・販売する個々のシステムとする．コアアセットは，製品系列に対応する各製品を実現する際に再利用される成果物ととらえることができる．

解決領域は製品系列に対応する製品を実現する構成要素（システム開発で作成される成果物）により表現する．開発される成果物は，抽象軸と専門軸によって分類できる．抽象軸は，成果物が開発や利用される開発工程を表す．専門軸は，成果物が表現しようとする特定の分野を表す．抽象軸や専門軸等によって，成果物全体をとらえるための視点として，すでに様々なものが提供されている．代表例として，Zachman Framework（以下，ZFと略す）があげられる[13]．ZFでは，抽象軸は，抽象度の高い順に，Scope, Business model, System model, Technology model, Detailed representationに分け，専門軸は，Data, Function, Network, People, Time, Motivationに分けている．

本稿ではドメインを問題領域と解決領域に分けてとらえる．また，スコープ定義を行う組織は，解決領域の成果物を分類する視点としてZFを導入しているものとする．

なお，ドメイン全体を表すモデルの中から，コアアセットとして開発する製品系列を決定することは，市場動向と企業戦略から判定される．本稿では，そのようなスコープ定義の手法は範疇外とする．

3. カバー率

カバー率とは，コアアセットがどの程度の成果物を備えているかという，コアアセットの充実度を測るメトリクスである．以下，カバー率の実測方法を定義するための考え方を整理し，算出式を定義する．

3.1 カバー率の考え方

各BUは，ドメイン分析の結果，ターゲットとして定めた製品系列の製品を開発する際に再利用するコアアセットを開発することを計画する．しかし，想定したドメインの製品系列の製品を開発するためのコアアセットを，すべて1度に開発するわけではなく，段階的に開発することになる．したがって，投資申請する費用で開発するコアアセットが，BUが想定したドメインの製品系列のどの部分に相当するのかを検討することは重要である．

コアアセットはあらゆる成果物が対象になるが，対象とする製品系列に対して，考えられるあらゆる成果物を作成するのではなく，組織のねらいやプロジェクトの条件によって，成果物の種類や量を調整することになる．そこで，問題領域と解決領域の双方に対して，充実の度合いを計測する考え方により，カバー率を定義する．

3.2 カバー率の算出方法

カバー率の算出式を，式 (1) に示す．カバー率は，問題領域と解決領域のカバー率の加重平均とする．問題領域に関するカバー率（式 (2)）は，BUの戦略において開発を計画する製品系列数に対する，投資申請で開発予定とするコアアセットがカバーする製品系列数の割合とする．

式 (3) は解決領域に関するカバー率の算出式である．式 (3) に示すように，これは，組織の標準で定めた網羅項目数に対する，コアアセットが満たす網羅項目数の割合とする．

$$X = \frac{X1 * wx1 + X2 * wx2}{wx1 + wx2} \tag{1}$$

$$X1 = \frac{n_{pd}}{n_{ps}} \tag{2}$$

$$X2 = \frac{n_{cd}}{n_{cs}} \tag{3}$$

X：コアアセットのカバー率
$X1$：コアアセットとして実現する問題領域のカバー率
$X2$：コアアセットとして実現する解決領域のカバー率
$wx1$：問題領域のカバー率の重み
$wx2$：解決領域のカバー率の重み
n_{pd}：投資申請で想定する期間において開発予定とするコアアセットがカバーできる製品系列の数
n_{ps}：BUの戦略において開発を計画する製品系列の数
n_{cd}：コアアセットが満たす網羅項目数
n_{cs}：組織の標準で定めた網羅項目数

カバー率は0〜1の数値となる．なお組織の標準で定めた網羅項目，各カバー率の重みについては，各組

織で基準を決定するものとする．組織の標準で定めた網羅項目数とは，成果物の種類の網羅項目数，成果物の種類別の網羅項目数の合計で構成する．

量を測る観点には，コンポーネント数，ユースケース数，プログラムのステップ数，機能量を示すファンクションポイント数，仕様書等のドキュメントのページ数等も考えられる．これらの指標は，コアアセットの開発または，コアアセットを利用した製品開発のライフサイクル全体において，コスト管理の指標として利用されることになる．本稿では，経営層が投資審査において，コアアセットの要素の規模を迅速に見積もるための指標として使うため，「カバー率」を式 (1)〜(3) により定義した．

3.3 網羅項目数とは

網羅項目数とは，組織で定めた成果物の種類の網羅項目数と，成果物の種類別の網羅項目数の合計で構成する．成果物の種類の網羅項目数とは，開発のライフサイクルで想定される工程と作業の分類別に，あらかじめ組織が定めた作成すべき成果物の種類数の合計とする．成果物の種類別の網羅項目数とは，上記で示した成果物の種類ごとに，ドメインの共通と特定製品に固有の成果物が準備され，製品開発に向けて成果物が網羅的かどうかを確認するための項目の数の合計とする．

たとえば，ソフトウェア開発の工程の例として，分析，設計，実装，テスト等がある．作業の分類の例として，データ，機能，環境，GUI 等がある．分析工程，データの作業の分類に対する，成果物の種類の例としては，概念データモデルの構造図と概念データモデルのデータ項目定義が考えられ，この場合，組織の標準で定めた成果物の種類数は 2 となる．その他の工程と作業の分類の組合せに対して，成果物の種類数を合計し，組織の標準で定めた成果物の種類を算出する．

成果物の種類として，概念データモデルの構造図と概念データモデルのデータ項目定義を定義したとする．これら 2 つの成果物の種類別の網羅項目として以下が考えられ，この場合の成果物の種類別の網羅項目数は 6 となる．分析工程，データの作業分類に関する成果物のみを組織が定めるコアアセットの種類とする場合は，組織の標準で定めた網羅項目数は，$2 + 6 = 8$ となる．

《成果物の種類別の網羅項目》

(a) 概念データモデルの構造図

(a-1) ドメインの共通データが，エンティティとして漏れなく構造図に定義されているか？

(a-2) 主要製品の固有データが，エンティティとして構造図に定義されているか？

(a-3) 開発完了/未定の製品の固有データが，エンティティとして構造図に定義されているか？

(b) 概念データモデルのデータ項目定義

(b-1) ドメインの共通データに対して漏れなくデータ項目仕様が作成されているか？

(b-2) 主要製品の固有データに対してデータ項目仕様が作成されているか？

(b-3) 開発完了/未定の製品の固有データに対してデータ項目仕様が作成されているか？

4. 対 応 度

対応度とは，コアアセットの要素間の整合性の度合いを示す，質を測るメトリクスである．以下，対応度の実測方法を定義するための考え方を整理し，算出式を定義する．

4.1 対応度の考え方

2 章において，ドメインは，問題領域と解決領域があることを述べた．ドメインの問題領域と解決領域に対応させ，コアアセットの対応度も 2 つの視点からとらえることにする．

4.1.1 問題領域の視点からの対応度

コアアセットが製品系列の共通特性と可変特性の双方を備え，カバー率が高い場合でも，コアアセット上で共通特性と可変特性に対応する部分の切り分けが不明瞭であるとき，製品開発の際，コアアセットの選択や拡張の方法が分かりにくく，製品の品質に影響が出るリスクがある．既存のアプリケーション群を再構成してプロダクトラインを構築する際に，単に既存の成果物を集めてコアアセットにしてしまうとこのようなことが発生しやすい．

製品系列の各特性が，コアアセットに正しく反映されているかについての確認はドメインごとに異なる．しかし，共通特性，可変特性が，明確に切り分けられてコアアセットとして表されているかについては，ドメインに依存しない．そこで，次の 2 つの観点で問題領域の対応度をとらえる．

(a1) 製品系列の共通特性の観点からの対応度

これは製品系列間の共通特性とコアアセットの対応度を示す．たとえば，顧客登録ドメインにおいて，法人向け顧客管理と，個人向け顧客管理の 2 つの製品系列を設定したとする．共通のデータ項目として，顧客 ID，顧客名がある．法人向けでは設立年月日を，個人向けでは生年月日を管理するという違いがある．論理データモデルの 1 つの顧客エンティティ上に，顧客 ID，顧客名，設立年月日，生年月日が混在して定義されている場合，カバー率では問題なくても，製品開発

ではカスタマイズが困難なコアアセットになる．共通特性である，顧客 ID と顧客名が，可変特性と明確に切り分けられている（たとえば，別のエンティティとして定義されている）場合，共通特性と論理データモデルとの対応がとれているとする．

(a2) 特定製品系列に固有の可変特性の観点からの対応度

これは特性製品系列に固有の可変特性と，コアアセットの対応度を示す．(a1) で取り上げた顧客管理ドメインの論理データモデルにおいて，製品系列別に法人顧客エンティティと個人顧客エンティティの差分部分が定義され，2 種類の差分部分が明確に切り分けられて定義されているとき，可変特性と，論理データモデルの対応がとれているとする．

なお，問題領域の対応度は，(a1) と (a2) の観点を含む確認項目を，組織であらかじめ設定しておき，その項目に対する対象コアアセットの合格確認項目の割合で求める．

4.1.2 解決領域の抽象軸の視点からの対応度

解決領域に抽象軸と専門軸からなる ZF を導入することを述べた．専門軸の考え方はドメインの違いにより異なるが，抽象軸は開発の工程に沿ったものになりドメインの違いに依存しない．抽象軸に着目すると，任意の複数のコアアセットは，同一の抽象度に対応づけられる場合と，異なる抽象度に対応づけられる場合が考えられることから，対応度を次の 2 つの観点からとらえる．

(b1) 同一の抽象度内の対応度

抽象軸を同一のレベルに固定した場合の，その抽象度におけるコアアセット間の対応度を示す．たとえば，顧客管理ドメインを例に，Business model に抽象軸を固定した場合を考える．コアアセットとしては，顧客登録データモデル，顧客登録プロセスが定義されているとする．この際，顧客登録プロセスへ入出力されるデータの仕様は，顧客登録データモデルで定義している各データの仕様（型，桁，意味等）と合致している場合，これらの間で対応がとれているとする．

(b2) 異なる抽象度間の対応度

ある抽象度におけるコアアセットと，別の抽象度のコアアセットとの対応度を示す．たとえば，顧客管理ドメインのコアアセットで，Data に関するコアアセットのうち，System model の論理データモデルと Technology model の物理データモデルに着目する．前者の論理データモデルの正規化されたデータの仕様に基づき，データの種類，データの識別子，データ間の関連等が，後者の物理データモデルに保持されて定義されている状態のとき，前者と後者の間で対応がとれているとする．

4.2 対応度の算出方法

(a1)，(a2)，(b1)，(b2) の観点によるコアアセット間の対応度を統合し，コアアセットの対応度を定義する．対応度の算出式を式 (4) に示す．対応度は，問題領域と解決領域のそれぞれの対応度の加重平均により求める．式 (5) は問題領域の対応度の算出式である．組織が定める問題領域の対応度確認項目数に対する対象コアアセットが満たす項目数の割合で求める．組織が定める問題領域の対応度確認項目は，(a1) および (a2) に示す製品系列間の整合性の確認項目である．

式 (6) は解決領域の対応度の算出式である．解決領域の対応度（式 (6)）は，(b1) の同一の抽象度内での対応度（式 (7)）と (b2) 異なる抽象度間での対応度（式 (8)）との加重平均で表す．同一の抽象度内および異なる抽象度間の対応度は，組織が定めるそれぞれの対応度確認項目数に対する，対象コアアセットの合格数（目標数）の割合とする．

$$Y = \frac{Y1 * wy1 + Y2 * wy2}{wy1 + wy2} \tag{4}$$

$$Y1 = \frac{n_{qd}}{n_{qs}} \tag{5}$$

$$Y2 = \frac{Y2a * wy2a + Y2b * wy2b}{wy2a + wy2b} \tag{6}$$

$$Y2a = \frac{n_{ad}}{n_{as}} \tag{7}$$

$$Y2b = \frac{n_{bd}}{n_{bs}} \tag{8}$$

Y：コアアセットの対応度
$Y1$：問題領域からの対応度
$Y2$：解決領域からの対応度
$Y2a$：解決領域からの同一抽象度内の対応度
$Y2b$：解決領域からの異なる抽象度間の対応度
$wy1$：問題領域からの対応度の重み
$wy2$：解決領域からの対応度の重み
$wy2a$：解決領域からの同一抽象度内の対応度の重み
$wy2b$：解決領域からの異なる抽象度間の対応度の重み
n_{qd}：コアアセットが満たす問題領域の対応度確認合格項目数（または合格目標数）
n_{qs}：組織の標準で定めた問題領域の対応度確認項目数
n_{ad}：コアアセットが満たす解決領域の同一抽象度内の観点からの対応度確認合格項目数（または合格目標数）
n_{as}：組織の標準で定めた解決領域の同一抽象度内の観点からの対応度確認項目数

n_{bd}：コアアセットが満たす解決領域の異なる抽象度間の観点からの対応度確認合格項目数（または合格目標数）

n_{bs}：組織の標準で定めた解決領域の異なる抽象度間の観点からの対応度確認項目数

対応度は 0〜1 の数値となる．なお，各対応度の重み，成果物間の対応度確認項目の考え方は，あらかじめ組織で基準を決定しておくものとする．

ところで，質を測る観点には「対応度」で計測するコアアセット全体の品質以外に，コアアセットの各要素の品質も考えられる．たとえば，各要素に対するレビュー指摘件数，エラー密度等が考えられる．これらの指標は，コアアセットの開発または，コアアセットを利用した製品開発のライフサイクル全体において，品質管理の指標として利用されることになる．本稿では，経営層が投資審査において，コアアセットの全体の品質を迅速に判断するための指標として使うため，「対応度」を式 (4)〜(8) により定義した．

4.3 対応度確認項目とは

対応度確認項目とは成果物の種類の任意の組合せに対する，品質に関する考慮すべき確認項目とする．組織の標準として，成果物の種類を定義しておくと，任意の成果物間の関係に対して，品質確認のポイントをあらかじめ定義しておくことができる．上述したように，対応度の算出のため，組織は次の 3 種類の対応度確認項目を利用する．これらの項目の事例ならびにチェック結果を付録の表 4〜表 6 に示す．

(1) 組織の標準で定めた問題領域の対応度確認項目

これは，製品系列の共通および可変特性が，コアアセットに明確に反映されていることを確認するための項目である．共通特性と可変特性の両方の観点からの確認項目に分けられる．たとえば，論理データモデルには問題領域の共通特性が明確に表現されているか等を確認する．付録の**表 4** に例を示す．

(2) 組織の標準で定めた解決領域の同一抽象度内の観点からの対応度確認項目

これは，抽象軸を同一のレベルに固定した場合の，その抽象度におけるコアアセット間の対応関係を確認するための項目である．たとえば，論理プロセスモデルへ入出力されるデータの仕様は，論理データモデルで定義している各データの仕様（型，桁，意味等）と合致しているか等を確認する．付録の**表 5** に例を示す．

(3) 組織の標準で定めた解決領域の異なる抽象度間の観点からの対応度確認項目

ある抽象度におけるコアアセットと，別の抽象度のコアアセットとの対応関係を確認するための項目である．たとえば，論理データモデルの正規化されたデータの仕様に基づき，データの種類，データの識別子，データ間の関連等が，物理データモデルに保持されて定義されているか等がある．付録の**表 6** に例を示す．

5. 適 用 事 例

複数のプロダクトラインを保持する組織が，どのプロダクトラインへ投資すべきかを検討するケースを想定し，コアアセットのスコープ定義手法を適用した例を示す．本章での手順および各データは，実際のプロダクトラインへの投資計画を立案するために作成された各プロダクトラインのデータに基づき関係者へインタビューを行い，その結果に基づいて著者らが作成した．

5.1 組織の説明

経営者は，次期プロダクトライン開発への投資のため，各 BU からプロダクトライン型開発の候補を申請させ，これらの比較・分析を通して投資対象を決定しようとしている．投資額の合計はすでに決定しており，それらをどのように各プロダクトラインへ最適に分配するかを検討している．ここで申請されたプロダクトラインは次の 3 件である．

PL1：製品情報トレーサビリティシステム

本システムは，各ユーザが入力した，製品の配送経路（出発/着地点，出発/到着時期，製品の梱包状態等）の情報を登録管理するシステムである．トレーサビリティとは，製品の由来，運搬経路，内容，変更の履歴等の情報を後からトレース（追跡）できることである．本ドメインの製品系列は 2 つである．大型製品向けと，小型の市販製品向けの 2 つの系列を設定している．前者は，製品個別にトレースすることが特性であり，後者については複数の製品を同時に持ち運ぶことが多いため，梱包材や運搬容器に製品を集めた全体もトレースできるようにするという違いがある．

PL2：情報編集システム

対象ドメインは，特定分野向けの大規模な情報編集支援システムである．顧客は，従来は手作業で情報を編集していたため，顧客ごとにやり方が異なる．やり方の違いをパターン化し複数の製品系列を定義した．すでに市場には参入しておりプロダクトラインは構築されている．プラットフォーム刷新にともないプロダクトラインを再構成する．

PL3：事務処理業務支援システム

ある業種に固有の事務処理および費用処理を支援するためのシステムである．対象組織の規模の違いにより大規模向けと小規模向けの 2 つの製品系列がある．

5.2 スコープ定義手法の適用結果

考案したスコープ定義手法を適用し 3 つのプロダクトラインのコアアセットのカバー率と対応度を計測し，タイプを決定した．計測結果に基づき，3 つのプロダクトラインを比較した．

5.2.1 組織標準の定義

組織のコアアセットの候補となる成果物の種類を 59 個定義し，ZF に基づき抽象軸と専門軸のマトリクス上に分類した．定義結果は，付録の表 3 に示す．59 個の成果物の種類に基づき，網羅確認項目，問題領域の対応度確認項目，解決領域の対応度確認項目（同一抽象度内，異なる抽象度間）を定義した．カバー率，対応度の算出の際の重み（$wx1$，$wx2$，$wy1$，$wy2$，$wy2a$，$wy2b$）の考え方を定義した．本稿では重みはすべて 1 に設定した．カバー率，対応度の基準値は 0.5 と設定した．

5.2.2 各 BU によるスコープの定義

各 BU は 2.1 節に示したプロセス（2）に基づいてスコープを定義する．h. 開発者視点のスコープ定義のプロセスにおいて，カバー率と対応度を算出し，プロダクトラインのタイプを決定する．以下，PL1 を取り上げてカバー率と対応度の算出過程を示す．

- **タイプ目標の設定**

新規参入ドメインであり，投資額は限定される予定のため，タイプ目標を「タイプ C」と設定した．

- **製品系列の選択**

製品情報トレーサビリティシステムの製品系列は 2 つだが，新規ドメインでリスクも高く，開発のターゲットは，大型製品のトレーサビリティを支援する製品系列を対象とすることにした．

- **組織標準から成果物を選択**

組織の標準から成果物種類を選択した結果を付録の表 3 に示す．表 3 に示すように，専門軸としては，Data，Function，Network を選択した．抽象軸については，System model，Technology model，Detailed representation を選択した．データ中心の業務系アプリケーションの特性はデータモデルに現れるため，そのような特性に関するノウハウを関係者で継承したいこと，データモデル以外にも再利用効率の向上のためにバイナリのコンポーネントもコアアセットに含めたいこと等を考慮して，データに関する仕様，コンポーネントとそれらの仕様等を選択し，合計 24 種類の成果物を選択した．

- **カバー率，対応度の見積り結果**

3 および 4 章で定義した算出式に基づいて，カバー率，対応度を計算した．

BU が想定した製品系列は，大型製品向けと小型の市販製品向けの 2 つの系列である．投資申請で開発予定とするコアアセットがカバーする製品系列は大型製品向け 1 つとした．したがって，問題領域のカバー率は $1/2 = 0.5$ となった．

解決領域のカバー率は，組織が定めた網羅項目数に対する対象コアアセットの網羅項目数により求めた．組織が定めた網羅項目に対する対象コアアセットの網羅項目の結果を付録の表 3 に示す．今回の投資では，24 種類の成果物に対して共通部分は網羅させ，大型製品のみの製品系列を対象にすることにしたので，可変部分の網羅性は，合格数が低い状態を目指すことにした．これらの数値は，問題領域は半分をカバーし，これらを実現する成果物は開発投資コストの制約を考えて控えめにするという意味に解釈できる

表 3 に示すように解決領域のカバー率は 0.28 である．重み 1 によりカバー率を算出すると，$(0.5+0.28)/2 = 0.39$ である．

対応度は，対応度確認項目のチェック結果に応じて計測した．組織が定めたそれぞれ問題領域における対応度，解決領域（同一抽象度），解決領域（異なる抽象度）の対応度確認項目に対する，対象コアアセットが満たす対応度確認項目の見積り結果を付録の表 4〜表 6 に示す．問題領域の対応度，解決領域の同一抽象度内の対応度，解決領域の異抽象度間の対応度は，それぞれ 0.73，0.67，0.62 となった．少ないコアアセットの成果物種類でありながら，これらの間の品質は確実に保つという意味に解釈できる．重み 1 によって対応度を算出すると，$\{0.73+(0.67+0.62)/2\}/2 = 0.69$ となった．本例の場合には，カバー率，対応度の基準値をそれぞれ 0.5 と設定したのでタイプ C と判定でき，目標タイプと合致していることになる．目標タイプに合致していなければ，網羅確認項目や対応度確認項目で合格（目標）とした項目を，追加する/とりやめることによって，カバー率や対応度を見直し，目標タイプに合致するまで見積り作業を繰り返す．

- **投資対効果**

コアアセットの実現コストを算出する．算出方法は組織標準で定めたものを用いる．たとえば，Böckle らの算出式[7] に基づいて見積もる．必要に応じて，網羅確認項目や対応度確認項目で合格（目標）とした項目を，追加する/とりやめることによって，カバー率や対応度を見直し，コアアセットや製品の開発コストや品質が妥当になるように再検討する．

同様にして，PL2，PL3 もそれぞれ担当する BU が計測する．

表 2 3 つのプロダクトラインのカバー率と対応度の例
Table 2 Example of levels of coverage and consistensy for 3 product lines.

ID	プロダクトライン名	製品系列		カバー率			対応度				タイプ	投資申請額順位	投資回収時期	効果順位
		想定数	設定数	問題領域カバー率	解決領域カバー率	カバー率	問題領域対応度	解決領域同抽象度対応度	解決領域異抽象度対応度	対応度				
PL1	製品情報トレーサビリティシステム	2	1	0.50	0.28	0.39	0.73	0.67	0.62	0.69	タイプC	3	3年	3
PL2	情報編集システム	5	2	0.40	0.88	0.64	0.09	0.20	0.21	0.15	タイプB	1	3年	1
PL3	事務処理業務支援システム	2	2	1.00	0.75	0.88	0.72	0.97	1.00	0.85	タイプA	2	3年	2

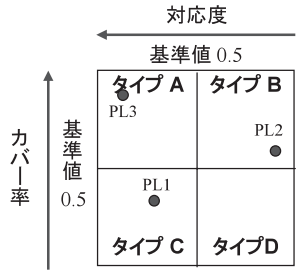

図 2 3 つのプロダクトラインの比較例
Fig. 2 Example for comparing 3 product lines.

5.2.3 組織による投資実行

経営組織に集められた各 BU によるプロダクトラインの計測結果を**表 2** に示す．表 2 の「想定数」と「設定数」は，それぞれカバー率の定義における n_{ps}，n_{pd} に相当する．

実際には開発コストや効果額等を示すべきだが，表 2 においては，投資額順位と効果順位を記載した．**図 2** に 3 つのプロダクトラインのカバー率と対応度の関係を 2 次元のマトリクス上にプロットしたイメージ図を示す．各 BU からの事業戦略の報告を考慮し，経営層により投資対象と投資額を決定する．

5.2.4 実際の投資結果との比較評価

実際の投資結果では，PL1 は投資回収額が低くても，開発計画が妥当であり投資実行となった．PL2 は，効果額が高くはあるが，投資額も多大であり，リスクが高い．今までの開発経験のあるドメインで既存製品の開発経験はあり，ビジネスルール等のノウハウは蓄積されているが，プロダクトラインを再構築するには規模が大きくリスクがある．そこで，プロダクトラインの開発計画を見直し，既存のプロダクトラインのリファクタリングで様子を見ることとなった．PL3 は，新規参入の大型ドメインである．組織の強みにすべく投資実行となった．

次に示すように，カバー率，対応度の計測結果から得られたタイプは，現実の投資判断の結果とマッチしたものになっている．したがって，本稿で提案したカバー率，対応度によるスコープ定義は，開発者の視点からのスコープを考慮し，今まで経営層で暗黙に行われていた投資判断のプロセスを可視化できる点で有用である．

《各プロダクトラインの可視化結果》

- PL1 は，新規参入の小規模プロダクトラインで開始することになるが，対応度を重視し，品質の安定したプロダクトラインであることがタイプ C として可視化された．
- PL2 に関しては，ドメインへの参入経験があり，高い投資効果が得られると予測できても，BU が対応度に関する関心が薄いと，その後の保守等でのコスト増大等のリスクが考えられる．その点が，タイプ B として可視化できた．
- PL3 は組織の強みとするドメインである．単に充実したコアアセットを作るだけではなく，品質面でもコストが必要であることを，事前に予測すべきであることがタイプ A として可視化できた．

6. 考　　察

考案したスコープ定義手法が 1.2 節で述べたスコープ定義の課題に対して妥当な解決策を提供できたかどうかを考察し，今後の課題を整理する．

(1) ドメインに依存せずにプロダクトラインを評価できるか？

カバー率は製品系列の数，製品系列を実現する成果物の種類数や成果物の種類別に共通部分や可変部分についてのカバー範囲がどの程度なのかを網羅確認項目で算出した．対応度は，問題領域の観点と解決領域の観点に分類し，同一抽象度内と異なる抽象度間の視点からの成果物間の対応関係の確認項目数，合格項目数により算出した．これらは，対象市場，利用技術，開発担当組織に依存せず算出可能である．

計測結果は 0〜1 の数値で表され，カバー率と対応度の結果からタイプに分類することにより，プロダクトラインの位置づけを明確にした．タイプに関する情

報とともに付録の表3〜表6等の統一した形式で記述したチェックと計測過程を用いることで，複数のプロダクトライン間の比較材料となり，投資先の判断に役立つ．

(2) 分かりやすく簡便な手法か？

カバー率と対応度は表3〜表6のチェックリストに基づいて計測でき，算出方法は容易である．プロダクトラインの利害関係者が，それぞれの立場で表3〜表6の形式を用いれば，コミュニケーションしやすくなり，スコープ定義がスムーズになると考えられる．

(3) コアアセットの全体の品質を考慮しているか？

対応度によりコアアセットの量だけではなく品質に関するスコープも定義可能である．従来はコアアセット間の整合性を直接的に可視化する手段がないため，整合性を保持するためのコストは見過ごされがちであった．しかし，表4〜表6の形式でコアアセット間の対応関係を示すことで，整合をとるための作業が明確になるだけではなく，コアアセットの開発に入る前に，検証方針も明確になる．対応度確認項目をチェックしていく過程で，開発者にも経営者にも品質を保つためにやるべきことがあるという認識を共有できる．

(4) 課題

コアアセットをどのような成果物で実現するのかについては，組織であらかじめコアアセットの候補を定め，共有しておく必要がある．今回は，ZFをベースに，抽象軸と専門軸を定義し，コアアセットの候補となる成果物を定義した．様々なドメインのプロダクトラインを想定している企業は，扱うプロダクトラインの違いによって，1つのコアアセットの候補の集合だけでは，不足する場合がある．今後は，ドメインの規模や業種・業務の違いより，複数のパターンを用意する工夫も必要である．

カバー率，対応度の重み変数，基準値，確認項目数の考え方については，組織であらかじめ定めておく必要がある．カバー率，対応度の見積りと，実際の開発コスト等を組にして蓄積し，コアアセットの実開発と製品への再利用のフィードバックを得て，見積り方法，重み，基準値，軸の持ち方等の組織の標準と改善の手法を充実させることが今後の課題である．

7. 関連研究

Schmidはスコープ定義手法を，対象と目的の組合せにより分類した[9]．スコープ定義の対象を，(a) Product portfolio，(b) Domain，(c) Assetの3つに分け，スコープ定義の目的を，スコープの認識，評価，最適化に分け，9個のカテゴリにスコープ定義手法を分類した．

我々の手法は，対象はSchmidの分類による(c) Assetで，目的はスコープ定義の最適化に相当する．Schmidの分類(c)は実行形式のコンポーネントにフォーカスしているが，我々の手法では，コンポーネントだけではなく，仕様，モデル，ドキュメント，各種ツール等もコアアセットの対象としている．なお我々の手法では，(b)の問題領域に関するスコープもカバー率によって一部を算出するが，主として(c)のスコープを算出するための手段として活用し，問題領域そのものを定める手法は範囲外である．

Schmidの分類において，我々の手法と同じ分類に相当するものに，Witheyの手法[14]とPuLSE-Eco[10]の手法がある．

Witheyの示すプロダクトラインのスコープ定義手法のプロセスは，(1)製品のポートフォリオの構築，(2)コアアセットのポートフォリオの構築，(3)スコープ定義からなり，各ポートフォリオの価値を決めるために決定木を用いて，合理的にスコープを定めることを目指す[14]．

我々の手法は(2)コアアセットのポートフォリオ定義を詳細化した手法であり，対応度やカバー率の計測結果をコアアセットのポートフォリオのエビデンスとして利用することができる．

DeBaudらは，PuLSE-Ecoと呼ばれるスコープ定義の手法を定義した[10]．PuLSE-Ecoでは，ビジネス上の目的に合致するFeature群を選択するためのプロセス，計算の考え方，各アウトプットの表現形態を提案している．

Schmidは上記のPuLSE-Ecoを拡張し，PuLSE-Eco V2.0[8]という手法を考案した．これは問題領域のスコープ定義も考慮に入れたうえで，解決領域のスコープ定義を厳密に定義した手法である．PuLSE-Eco V2.0は，Product Line Mapping (PLM)，Domain Potential Assessment (DPA)，Reuse Infrastructure Scoping (RIS)のプロセスから構成される．

我々の手法と，PuLSE-EcoおよびPuLSE-Eco V.2.0の違いは2点ある．1点目はプロダクトラインに含める要素として，PuLSE-EcoおよびPuLSE-Eco V.2.0がコンポーネントを対象としているのに対し，我々の手法ではコンポーネント以外の成果物も対象にしている点である．2点目はPuLSE-EcoおよびPuLSE-Eco V.2.0はアセットのスコープ定義において，製品系列，ビジネス上のリスク，実現可能性等を広く考慮していることに対し，我々の手法はコアアセットを実現する成果物の量や成果物全体の質にフォーカ

スする点である．我々の手法は，PuLSE-Eco の手法と組み合わせて適用することが可能であり，さらにスコープを算出する実測方法を詳細に定義している点は有用である．

Park らは，共通と可変部分の分析，可変部分の依存性分析，ドメインモデルの精査，コスト評価からなるスコープ定義のプロセスと，コスト評価の方法を提案した[15]．提案された手法の特徴は，可変部分のコストの算出において，可変部分の依存性（variability dependencies）をタイプ分けしたうえで，可変部分依存性の保持のためのコストを含めた点にある．

我々の手法には，製品系列の可変部分のスコープは含めているが，依存度は考慮していない．可変部分の依存度は対応度の算出に影響を与える．可変部分の依存度を考慮しなければならない場合は，そうでない場合と比較して，対応度は高く設定することになると考えられる．可変部分の依存度を考慮した場合の対応度の設定方法については，今後の課題である．

ところで，可変部分の依存性も含めたコアアセット開発のコストの見積りは，見積り作業だけで高コストとなる．我々の手法によって，組織がどのプロダクトラインに投資をするのか決定した後，各プロダクトラインの経済的な効果を詳細に見積もるというステップとし，後者の見積りの中で，Park らが提案するプロダクトラインのコスト算出を行うことが妥当である．

組織全体で自身のプロダクトラインのステータスを明らかにするための指標として BAPO モデルがある[16]．BAPO モデルには，Business，Architecture，Process，Organization の 4 つの側面に対して，メトリクスとレベル 1〜5 までの成熟度のモデルを定義した．しかし，BAPO モデルにおいて各メトリクスの算出方法は具体化されていない．我々の手法は Architecture の側面に対して，対応度とカバー率というメトリクスを定義し，算出式を具体化している．

8. おわりに

本稿では，プロダクトライン型ソフトウェア開発のスコープ定義において，コアアセット全体の量と質を計測するメトリクスである，カバー率と対応度を定義し，これらのメトリクスによりコアアセットのスコープを可視化する手法を提案した．本手法を 3 つのプロダクトライン開発候補の計画の立案と投資判断に適用した．適用結果から，コアアセットのスコープを定義する際に，従来の手法ではフォーカスされていなかった開発者視点からのコアアセットの充実の度合いや，整合性の度合いを可視化する点と，可視化した結果を，組織全体のプロダクトラインの最適化に活用可能なことを明らかにした．

今後は，今回開発したスコープ定義に基づき，コアアセットの開発，コアアセットの再利用によるアプリケーションの開発を行い，結果をコアアセット設計手法にフィードバックし，スコープ定義の手法を具体化する．また，既存の手法と統合し，さらに精度の高いスコープ定義手法へと拡張する．

謝辞 研究の機会を与えてくださった東芝ソリューション（株）遠藤直樹技監，東芝ソリューション（株）IT 技術研究所の落合正雄所長，同 IT 品質ラボラトリの栄光宏室長に感謝します．

参考文献

1) Mili, H., et al.: *Reuse-Based Software Engineering, Techniques, Organization and Controls*, p.636, John Wiley & Sons (2002).
2) Birk, A., et al.: Product Line Engineering, the State of the Practice, *IEEE Software*, Vol.20, No.6, pp.52–60 (2003).
3) Sugumaran, V., et al.: Software Product Line Engineering, *Comm. ACM*, Vol.49, No.12, pp.29–32 (2006).
4) Families Project, Fact-based Maturity through Institutionalisation Lessons-learned and Involved Exploration of System-family engineering. http://www.esi.es/Families/
5) SEI, Product Line Approach to Software Development. http://www.sei.cmu.edu/plp/plp_init.html
6) Clements, P. and Northrop, L.: *Software Product Lines: Practice and Patterns*, p.608, Addison-Wesley (2001). 前田卓雄（訳）：ソフトウェアプロダクトライン，p.652, 日刊工業新聞社 (2003).
7) Böckle, G., et al.: Calculating ROI for Software Product Lines, *IEEE Software*, Vol.21, No.3, pp.23–31 (2004).
8) Schmid, K.: A Comprehensive Product Line Scoping Approach and Its Validation, *Proc. 24th International Conference on Software Engineering*, pp.593–603 (2002).
9) Schmid, K.: Scoping Software Product Lines, An Analysis of an Emerging Technology, *Proc. 1st Software Product Line Conference* (*SPLC1*), pp.513–532 (2000).
10) DeBaud, J. and Schmid, K.: A Systematic Approach to Derive the Scope of Software Product Lines, *Proc. 21st International Conference on Software Engineering*, pp.34–43 (1999).
11) Cohen, S. and Northrop, L.: Object-Oriented Technology and Domain Analysis, *Proc. 5th*

International Conference on Software Reuse, pp.86–93 (1998).
12) Kang, K., et al.: Feature-Oriented Product Line Engineering, *IEEE Software*, Vol.19, No.4, pp.58–65 (2002).
13) The Zachman Institute for Framework Advancement. http://www.zifa.com/
14) Withey, J.: Investment Analysis of Software Assets for Product Lines, Technical Report, CMU/SEI-96-TR-010, ESC-TR-96-010 (1996).
15) Park, S.Y. and Kim, S.D.: A Systematic Method for Scoping Core Assets in Product Line Engineering, *Proc. 12th Asia-Pacific Software Engineering Conference*, pp.491–498 (2005).
16) van der Linden, F., et al.: Software Product Family Evaluation, *Proc. 3rd International Software Product Line Conference* (*SPLC 2004*), pp.110–129 (2004).

付　　録

表 3 に成果物の種類および網羅項目の例を示す．表 4〜表 6 に，それぞれ，問題領域の対応度確認項目，同一抽象度内の対応度確認項目，異なる抽象度間の対応度確認項目の例ならびに PL1 における確認結果の例を示す．

表 3 組織が定めた網羅確認項目とコアアセットが満たす確認項目の例
Table 3 Example of exhausive inspection checklist and items to be passed in the target core assets.

No.	役割軸	専門軸	コアアセット候補（成果物の種類）	網羅に関する確認項目			
				作成	共通部分網羅性*1	可変部分網羅性1*2	可変部分網羅性2*3
1	Scope	Data	ビジネスオブジェクト				
2	Scope	Function	業務一覧				
3	Scope	Network	ビジネス拠点リスト				
4	Scope	People	組織リスト				
5	Scope	Time	ビジネスイベント/サイクルリスト				
6	Scope	Motivation	ビジネスゴール				
7	Business model	Data	概念データモデル				
8	Business model	Data	概念データベース検証仕様				
9	Business model	Function	概念クラス構造				
10	Business model	Function	概念クラス間相互作用				
11	Business model	Function	概念機能テスト仕様				
12	Business model	Function	概念機能テスト結果				
13	Business model	Function	概念機能テスト環境				
14	Business model	Network	エンタープライズシステム構成				
15	Business model	People	概念ユーザインタフェースモデル				
16	Business model	Time	概念プロセスモデル				
17	Business model	Motivation	問題分析モデル				
18	System model	Data	論理データモデル	レ	レ	レ	
19	System model	Data	論理データベース検証仕様				
20	System model	Function	パッケージ構造	レ	レ	レ	
21	System model	Function	システムユースケース	レ	レ	レ	
22	System model	Function	論理クラス構造	レ	レ	レ	
23	System model	Function	論理クラス間相互作用	レ	レ	レ	
24	System model	Function	機能テスト仕様				
25	System model	Function	機能テスト結果				
26	System model	Function	機能テスト環境				
27	System model	Network	論理システム構成	レ	レ	レ	
28	System model	People	ユーザインタフェースモデル				
29	System model	Time	論理プロセスモデル				
30	System model	Motivation	ビジネスルール				
31	Technology model	Data	物理データモデル	レ	レ	レ	
32	Technology model	Data	物理データベース検証仕様				
33	Technology model	Function	物理パッケージ構造	レ	レ	レ	
34	Technology model	Function	物理クラス構造	レ	レ	レ	
35	Technology model	Function	物理クラス間相互作用	レ	レ	レ	
36	Technology model	Function	結合テスト仕様	レ	レ		
37	Technology model	Function	結合テスト結果	レ	レ		
38	Technology model	Function	結合テスト環境	レ	レ		
39	Technology model	Network	物理システム構成	レ	レ	レ	
40	Technology model	People	ユーザインタフェース項目仕様				
41	Technology model	Time	物理プロセスモデル				
42	Technology model	Motivation	ビジネスルール設計モデル				
43	Detail representation	Data	データベース	レ	レ	レ	
44	Detail representation	Data	データベーステスト仕様				
45	Detail representation	Function	クラス仕様	レ	レ	レ	
46	Detail representation	Function	製品開発支援環境				
47	Detail representation	Function	ソースコード	レ	レ	レ	
48	Detail representation	Function	フレームワーク	レ	レ	レ	
49	Detail representation	Function	コンポーネント	レ	レ	レ	
50	Detail representation	Function	ビルド	レ	レ	レ	
51	Detail representation	Function	単体テスト仕様	レ	レ		
52	Detail representation	Function	単体テスト結果	レ	レ		
53	Detail representation	Function	単体テスト環境	レ	レ		
54	Detail representation	Network	モジュール配置	レ	レ	レ	
55	Detail representation	People	ユーザインタフェースプログラム				
56	Detail representation	People	ユーザインタフェーステスト仕様				
57	Detail representation	People	ユーザインタフェーステスト環境				
58	Detail representation	Time	プロセスコントロールプログラム				
59	Detail representation	Motivation	ビジネスルール記述				
コアアセットが満たす網羅項目数			内訳	24	24	18	0
			合計: n_{cal}				66
組織の標準で定めた網羅項目数			内訳	59	59	59	59
			合計: n_{cs}				236
解決領域のカバー率: $X2$							0.28

レ：該当行の成果物の種類が該当列の確認項目を満たす（または予定である）ことを示す．
*1 共通部分の網羅性：ドメインの共通部分が網羅されているか？
*2 可変部分の網羅性1：主要製品の固有部分（可変部分）が網羅されているか？
*3 可変部分の網羅性2：開発完了／未定の製品の固有部分（可変部分）が網羅されているか？

表 4 組織が定めた問題領域の対応度確認項目とコアアセットが満たす確認項目の例
Table 4 Example of quality inspection checklist for the problem domain and items to be passed in the target core assets.

No.	役割軸	専門軸	コアアセット候補（成果物の種類）	対応度に関する確認項目		
				作成	共通部分の整合性*1	可変部分の整合性*2
1	Scope	Data	ビジネスオブジェクト			
2	Scope	Function	業務一覧			
3	Scope	Network	ビジネス拠点リスト			
4	Scope	People	組織リスト			
5	Scope	Time	ビジネスイベント/サイクルリスト			
6	Scope	Motivation	ビジネスゴール			
7	Business model	Data	概念データモデル			
8	Business model	Data	概念データベース検証仕様			
9	Business model	Function	概念クラス構造			
10	Business model	Function	概念クラス間相互作用			
11	Business model	Function	概念機能テスト仕様			
12	Business model	Function	概念機能テスト結果			
13	Business model	Function	概念機能テスト環境			
14	Business model	Network	エンタープライズシステム構成			
15	Business model	People	概念ユーザインタフェースモデル			
16	Business model	Time	概念プロセスモデル			
17	Business model	Motivation	問題分析モデル			
18	System model	Data	論理データモデル	レ	レ	レ
19	System model	Data	論理データベース検証仕様			
20	System model	Function	パッケージ構造	レ	レ	レ
21	System model	Function	システムユースケース	レ	レ	レ
22	System model	Function	論理クラス構造	レ	レ	レ
23	System model	Function	論理クラス間相互作用	レ	レ	レ
24	System model	Function	機能テスト仕様			
25	System model	Function	機能テスト結果			
26	System model	Function	機能テスト環境			
27	System model	Network	論理システム構成	レ	レ	レ
28	System model	People	ユーザインタフェースモデル			
29	System model	Time	論理プロセスモデル			
30	System model	Motivation	ビジネスルール			
31	Technology model	Data	物理データモデル	レ	レ	レ
32	Technology model	Data	物理データベース検証仕様			
33	Technology model	Function	物理パッケージ構造	レ		
34	Technology model	Function	物理クラス構造	レ	レ	レ
35	Technology model	Function	物理クラス間相互作用	レ	レ	レ
36	Technology model	Function	結合テスト仕様	レ	レ	レ
37	Technology model	Function	結合テスト結果	レ		
38	Technology model	Function	結合テスト環境	レ		
39	Technology model	Network	物理システム構成	レ		
40	Technology model	People	ユーザインタフェース項目仕様			
41	Technology model	Time	物理プロセスモデル			
42	Technology model	Motivation	ビジネスルール設計モデル			
43	Detail representation	Data	データベース	レ	レ	レ
44	Detail representation	Data	データベーステスト仕様			
45	Detail representation	Function	クラス仕様	レ		
46	Detail representation	Function	製品開発支援環境		レ	レ
47	Detail representation	Function	ソースコード	レ		
48	Detail representation	Function	フレームワーク	レ	レ	レ
49	Detail representation	Function	コンポーネント	レ	レ	レ
50	Detail representation	Function	ビルド	レ	レ	レ
51	Detail representation	Function	単体テスト仕様	レ	レ	レ
52	Detail representation	Function	単体テスト結果	レ	レ	
53	Detail representation	Function	単体テスト環境	レ	レ	
54	Detail representation	Network	モジュール配置	レ	レ	
55	Detail representation	People	ユーザインタフェースプログラム			
56	Detail representation	People	ユーザインタフェーステスト仕様			
57	Detail representation	People	ユーザインタフェーステスト環境			
58	Detail representation	Time	プロセスコントロールプログラム			
59	Detail representation	Motivation	ビジネスルール記述			
コアアセットが満たす問題領域の対応度確認項目数			内訳		19	16
			合計: n_{qd}			35
組織の標準で定めた問題領域の対応度確認項目数			内訳		24	24
			合計: n_{qs}			48
問題領域の対応度: $Y1$						0.73

レ：該当行の成果物の種類が該当列の確認項目を満たす（または予定である）ことを示す．
*1 共通部分の整合性：製品系列の共通部分が可変部分と分離してあり明確か？
*2 可変部分の整合性：製品系列に固有の可変部分がそれぞれ独立しており明確か？

表 5 組織が定めた解決領域の同一抽象度内の対応度確認項目とコアアセットが満たす確認項目の例
Table 5 Example of quality inspection checklist in the same abstract degree for the solution domain and items to be passed in the target core assets.

No	抽象度	成果物種類①	成果物種類②	同一抽象度内の対応度確認項目	該当*1	確認*2
1	Scope	ビジネスオブジェクト	業務一覧	ビジネスオブジェクトで定義された仕様(オブジェクト，役割)に基づいて，業務一覧は定義されているか？		
2	Scope	業務一覧	ビジネス拠点	業務一覧の業務は，ビジネス拠点で定義された拠点で，運用可能か？		
3	Scope	組織リスト	ビジネス拠点	組織リストの組織は，ビジネス拠点で定義された拠点に，配置可能か？		
4	Scope	ビジネスイベントリスト	業務一覧	ビジネスイベントリストで定義された仕様(イベント，タイミング)に基づいて，業務一覧は定義されているか？業務の起動条件は妥当か？		
5	Scope	業務一覧	ビジネスゴール	業務一覧で定義された各業務を実行すると，ビジネスゴールを達成できることが証明できるか？実例や実証説明が可能か？		
6	Business model	概念データモデル	概念データモデル検証仕様	概念データベース検証仕様に基づいて，概念データベースの検証が完了しているか？		
7	Business model	概念データモデル	概念クラス構造	概念データモデルで定義されている各データの仕様(型，桁，意味)に基づいて，概念クラス構造が定義されているか？		
8	Business model	概念データモデル	概念クラス間相互作用	概念データモデルで定義されている各データの仕様(型，桁，意味)に基づいて，概念クラス間相互作用が定義されているか？		
9	Business model	概念クラス構造	概念機能テスト仕様	概念機能テスト仕様に基づいて，概念クラス構造の検証が完了しているか？		
10	Business model	概念クラス間相互作用	概念機能テスト仕様	概念機能テスト仕様に基づいて，概念クラス間相互作用の検証が完了しているか？		
11	Business model	概念機能テスト仕様	概念機能テスト結果	概念機能テスト仕様に基づいて，概念機能テスト結果が記述されているか？		
12	Business model	概念クラス構造	概念クラス間相互作用	概念クラス構造で定義された各クラスの仕様(属性，操作，役割)に基づいて，概念クラス間相互作用の仕様は定義されているか？		
13	Business model	概念クラス構造	エンタープライズシステム構成	概念クラス構造で定義された各クラスは，エンタープライズシステム構成に，配置可能か？		
14	Business model	概念ユーザインタフェースモデル	概念プロセスモデル	概念ユーザインタフェースモデルで定義された各インタフェース定義は，概念プロセスモデルにおいて，想定通りの使われ方で定義されているか？		
15	Business model	概念プロセスモデル	概念クラス間相互作用	概念クラス間相互作用は，概念プロセスモデルで定義された業務の流れを想定して定義されているか？		
16	Business model	問題分析モデル	概念データモデル	問題分析モデルで宣言された対象ドメインに対する各要件に基づいて，概念データモデルは定義されているか？		
17	Business model	問題分析モデル	概念クラス構造	問題分析モデルで宣言された対象ドメインに対する各要件に基づいて，概念クラス構造は定義されているか？		
18	Business model	問題分析モデル	概念クラス間相互作用	問題分析モデルで宣言された対象ドメインに対する各要件に基づいて，概念クラス間相互作用は定義されているか？		
19	Business model	問題分析モデル	エンタープライズシステム構成	問題分析モデルで宣言された対象ドメインに対する各要件に基づいて，エンタープライズシステム構成は定義されているか？		
20	Business model	問題分析モデル	概念ユーザインタフェースモデル	問題分析モデルで宣言された対象ドメインに対する各要件に基づいて，概念ユーザインタフェースモデルは定義されているか？		
21	System model	論理データモデル	論理データモデル検証仕様	論理データベース検証仕様に基づいて，論理データベースの検証が完了しているか？	レ	レ
22	System model	論理データモデル	論理クラス構造	論理データモデルで定義されている各データの仕様(型，桁，意味)に基づいて，論理クラス構造は定義されているか？	レ	レ
23	System model	論理データモデル	論理クラス間相互作用	論理データモデルで定義されている各データの仕様(型，桁，意味)に基づいて，論理クラス間相互作用は定義されているか？	レ	レ
24	System model	論理データモデル	論理プロセスモデル	論理データモデルで定義されている各データの仕様(型，桁，意味)に基づいて，論理プロセスモデルは定義されているか？	レ	
25	System model	論理クラス構造	論理クラス間相互作用	論理クラス構造で定義された定義された各クラスの仕様(属性，操作，役割)に基づいて，論理クラス間相互作用の仕様が定義されているか？	レ	レ
26	System model	論理クラス構造	機能テスト仕様	機能テスト仕様に基づいて，論理クラス構造の検証が完了しているか？	レ	レ
27	System model	論理クラス間相互作用	機能テスト仕様	機能テスト仕様に基づいて，論理クラス間相互作用の検証が完了しているか？	レ	レ
28	System model	機能テスト仕様	機能テスト結果	機能テスト仕様に基づいて，機能テスト結果が記述されているか？		
29	System model	ユーザインタフェースモデル	論理プロセスモデル	ユーザインタフェースモデルで定義された各インタフェース定義(レイアウト，画面遷移)に基づいて，論理プロセスモデルは定義されているか？	レ	レ
30	System model	論理プロセスモデル	論理クラス間相互作用	論理クラス間相互作用は，論理プロセスモデルで定義された業務の流れを想定して定義されているか？	レ	レ
31	System model	ビジネスルール	論理データモデル	ビジネスルールで定義した業務の特性に基づいて，論理データモデルが定義されているか？	レ	レ
32	System model	ビジネスルール	論理クラス構造	ビジネスルールで定義した業務の特性に基づいて，論理クラス構造が定義されているか？	レ	
33	System model	ビジネスルール	論理クラス間相互作用	ビジネスルールで定義した業務の特性に基づいて，論理クラス間相互作用が定義されているか？	レ	
34	System model	ビジネスルール	パッケージ構造	ビジネスルールで定義した業務の特性に基づいてパッケージ構造が定義されているか？	レ	
35	System model	ビジネスルール	ユーザインタフェース	ビジネスルールで定義した業務の特性に基づいて，ユーザインタフェースが定義されているか？		
36	System model	ビジネスルール	論理プロセスモデル	ビジネスルールで定義した業務の特性に基づいて，論理プロセスモデルが定義されているか？		
37	Technology model	物理データモデル	物理データモデル検証仕様	物理データベース検証仕様に基づいて，概念データベースの検証が完了しているか？	レ	レ

(次頁に続く)

表 5 組織が定めた解決領域の同一抽象度内の対応度確認項目とコアアセットが満たす確認項目の例（前頁続き）
Table 5 Example of quality inspection checklist in the same abstract degree for the solution domain and items to be passed in the target core assets (cont'd from previous page).

No	抽象度	成果物種類①	成果物種類②	同一抽象度内の対応度確認項目	該当*1	確認*2
38	Technology model	物理データモデル	物理クラス構造	物理データモデルの仕様に基づいて，物理クラス構造が定義されているか？	レ	レ
39	Technology model	物理データモデル	物理クラス間相互作用	物理データモデルで定義されている各データの仕様に基づいて，物理クラス間相互作用が定義されているか？	レ	レ
40	Technology model	物理クラス構造	物理パッケージ構成	物理クラス構造で定義された各クラスは，物理パッケージ構造に，配置可能か？	レ	レ
41	Technology model	物理クラス構造	物理クラス間相互作用	物理クラス構造で定義された各クラスは，物理クラス間相互作用の仕様において，想定通りの使われ方で定義されているか？	レ	レ
42	Technology model	物理クラス構造	結合テスト仕様	結合テスト仕様に基づいて物理クラス構造の検証が完了しているか？	レ	レ
43	Technology model	物理クラス間相互作用	結合テスト仕様	結合テスト仕様に基づいて物理クラス間相互作用の検証が完了しているか？	レ	レ
44	Technology model	結合テスト仕様	結合テスト結果	結合テスト仕様に基づいて結合テスト結果が記述されているか？		
45	Technology model	ユーザインタフェース項目仕様	物理プロセスモデル	ユーザインタフェース項目仕様で定義された仕様（入出力項目，アクション）は，物理プロセスモデルにおいて定義されているか？		
46	Technology model	物理プロセスモデル	物理クラス間相互作用	物理クラス間相互作用は，物理プロセスモデルで定義された業務の流れを想定して定義されているか？		
47	Technology model	ビジネスルール設計モデル	物理データモデル	ビジネスルール設計モデルで定義した業務の特性が物理データモデルに考慮されているか？		
48	Technology model	ビジネスルール設計モデル	物理クラス構造	ビジネスルール設計モデルで定義した業務の特性が物理クラス構造に考慮されているか？		
49	Technology model	ビジネスルール設計モデル	物理クラス間相互作用	ビジネスルール設計モデルで定義した業務の特性が物理クラス間相互作用に考慮されているか？		
50	Technology model	ビジネスルール設計モデル	物理システム構造	ビジネスルール設計モデルで定義した業務の特性が物理システム構造に考慮されているか？		
51	Technology model	ビジネスルール設計モデル	ユーザインタフェース項目仕様	ビジネスルール設計モデルで定義した業務の特性がユーザインタフェース項目仕様に考慮されているか？		
52	Technology model	ビジネスルール設計モデル	物理プロセスモデル	ビジネスルール設計モデルで定義した業務の特性が物理プロセスモデルに考慮されているか？		
53	Detail representation	データベース	データベーステスト仕様	データベーステスト仕様に基づいて，データベースの検証が完了しているか？	レ	レ
54	Detail representation	データベース	ビルド	データベースで定義されている各データは，ビルドと連動して動作するか？	レ	レ
55	Detail representation	データベース	フレームワーク	データベースで定義されている各データは，フレームワーク上で動作するか？	レ	レ
56	Detail representation	データベース	コンポーネント	データベースで定義されている各データは，コンポーネントと連動して動作するか？	レ	レ
57	Detail representation	クラス仕様	ソースコード	クラス仕様で定義された各クラスの仕様に基づいて，ソースコードは作成されているか？	レ	
58	Detail representation	ソースコード	単体テスト仕様	単体テスト仕様に基づいて，ソースコードの検証が完了しているか？	レ	
59	Detail representation	ビルド	モジュール配置	ビルドは，モジュール配置に基づいて，配置することが可能か？	レ	
60	Detail representation	ユーザインタフェースプログラ	モジュール配置	ユーザインタフェースプログラムは，モジュール配置に基づいて，配置することが可能か？		
61	Detail representation	ユーザインタフェースプログラム	ユーザインタフェースプログラム単体テスト仕様	ユーザインタフェースプログラム単体テスト仕様に基づいて，ユーザインタフェースプログラムが検証されているか？		
62	Detail representation	プロセスコントロールプログラム	モジュール配置	プロセスコントロールプログラムは，モジュール配置に基づいて配置することが可能か？	レ	
63	Detail representation	プロセスコントロールプログラム	単体テスト仕様	プロセスコントロールプログラムは，単体テスト仕様に基づいて検証されているか？	レ	
64	Detail representation	単体テスト仕様	単体テスト結果	単体テスト仕様に基づいて単体テスト結果が記述されているか？	レ	
65	Detail representation	ビジネスルール記述	コンポーネント	ビジネスルール記述で定義した業務特性がコンポーネントに実現されているか？		
66	Detail representation	ビジネスルール記述	ユーザインタフェースプログラム	ビジネスルール記述で定義した業務特性がユーザインタフェースプログラムに実現されているか？		
67	Detail representation	ビジネスルール記述	プロセスコントロールプログラム	ビジネスルール記述で定義した業務特性がプロセスコントロールプログラムに実現されているか？		
コアアセットが満たす解決領域の対応度確認項目数（同一抽象度内）: n_{ad}						20
組織が定めた解決領域の対応度確認項目数（同一抽象度内）: n_{as}						30
解決領域の同一抽象度内の対応度: $Y2a$						0.67

*1 レ：該当行の成果物種類①と成果物種類②の間の「対応度に関する品質確認項目」が，確認すべき項目であることを示す．
*2 レ：該当行の成果物種類①と成果物種類②の間の「対応度に関する品質確認項目」は，対象コアアセットが満たしている（または満たす予定）であることを示す．

表 6 組織が定めた解決領域の異なる抽象度内の対応度確認項目とコアアセットが満たす確認項目の例
Table 6 Example of quality inspection checklist in the different abstract degrees for the solution domain and items to be passed in the target core assets.

No	成果物種類①		成果物種類②		異なる抽象度間の対応度確認項目	該当 *1	確認 *2
	抽象度	成果物種類	抽象度	成果物種類			
1	Scope	ビジネスオブジェクト	Business model	概念データモデル	ビジネスオブジェクトで定義された主たるオブジェクトの仕様に基づき，データの意味，種類，識別方法，データ間の関連が，概念データモデルに保持されて定義されているか？		
2	Business model	概念データモデル	System model	論理データモデル	概念データモデルで定義されたデータの仕様に基づき，データの意味，種類，識別方法，データ間の関連が，論理データモデルに保持されて定義されているか？	レ	
3	System model	論理データモデル	Technology model	物理データモデル	論理データモデルの正規化されたデータの仕様に基づき，データの種類，データの識別子，データ間の関連が，物理データモデルに保持されて定義されているか？	レ	レ
4	Technology model	物理データモデル	Detail representation	データベース	物理データベースモデルでの定義結果通りに，データベースが実装されているか？	レ	レ
5	Scope	業務一覧リスト	Business model	概念クラス構造	業務一覧リストに定義された業務の仕様に基づき，業務をサポートする観点でシステムの概念クラス構造が定義されているか？		
6	Scope	業務一覧リスト	Business model	概念クラス間相互作用	業務一覧リストに定義された業務の仕様に基づき，業務をサポートする観点で，業務の流れのシナリオに基づいて，システムの概念クラス間の相互作用が定義されているか？		
7	Business model	概念クラス構造	System model	論理クラス構造	概念クラス構造で定義されたクラスの仕様に基づき，クラスの属性，操作，クラス間関係が，論理クラス構造に保持されて定義されているか？	レ	
8	Business model	概念クラス間相互作用	System model	論理クラス間相互作用	概念クラス間相互作用で定義されたクラス間の相互作用の仕様に基づき，論理クラス間の相互作用が定義されているか？	レ	
9	System model	論理クラス構造	Technology model	物理クラス構造	論理クラス構造で定義されたクラスの仕様に基づき，クラスの属性，操作，クラス間関係が，物理クラス構造に保持されて定義されているか？	レ	レ
10	System model	論理クラス間相互作用	Technology model	物理クラス間相互作用	論理クラス間相互作用で定義されたクラス間の相互作用の仕様に基づき，物理クラス間の相互作用が定義されているか？	レ	レ
11	Technology model	物理クラス構造	Detail representation	クラス仕様	物理クラス構造で定義されたクラスの仕様に基づき，クラスの属性，操作，クラス間関係がクラス仕様に保持されて定義されているか？	レ	レ
12	Technology model	物理クラス構造	Detail representation	ソースコード	物理クラス構造での定義結果通りに，ソースコードが記述されているか？	レ	レ
13	Technology model	物理クラス間相互作用	Detail representation	クラス仕様	物理クラス間相互作用で定義されたクラスの呼び出し関係の仕様に基づき，クラスの属性，操作，クラス間関係がクラス仕様に保持されて定義されているか？	レ	レ
14	Scope	ビジネス拠点リスト	Business model	エンタープライズシステム構成	ビジネス拠点リストで定義された拠点の仕様に基づき，配置場所，拠点役割を考慮して，エンタープライズシステム構成が定義されているか？		
15	Business model	エンタープライズシステム構成	System model	論理システム構成	エンタープライズシステム構成の仕様に基づき，サーバ配置，役割を考慮して，論理システム構成が定義されているか？	レ	レ
16	System model	論理システム構成	Technology model	物理システム構成	論理システム構成の仕様に基づき，システム／サブシステムの配置，役割を考慮して，物理システム構成が定義されているか？	レ	
17	Technology model	物理システム構成	Detail representation	モジュール配置	物理システム構成の仕様に基づき，ハードウェア，ソフトウェア，ネットワークの仕様を考慮して，モジュール配置が定義されているか？	レ	
18	Scope	組織リスト	Business model	概念ユーザインタフェースモデル	組織リストに定義された各組織と役割に基づいて，概念ユーザインタフェースモデルが定義されているか？		
19	Business model	概念ユーザインタフェースモデル	System model	ユーザインタフェースモデル	概念インタフェースモデルで定義されたインタフェースの仕様に基づき，インタフェースモデルが定義されているか？		
20	System model	ユーザインタフェースモデル	Technology model	ユーザインタフェース項目仕様	ユーザインタフェースモデルの役割，特性，レイアウト等の仕様に基づき，ユーザインタフェース項目仕様が定義されているか？		
21	Technology model	ユーザインタフェース項目仕様	Detail representation	ユーザインタフェースプログラム	ユーザインタフェース項目仕様通りにユーザインタフェースプログラムが実装されているか？		
22	Scope	ビジネスイベントリスト	Business model	概念プロセスモデル	ビジネスイベントリストに定義されたイベントの仕様に基づいて，イベントの時期，役割，目的，関連組織等を保持し，概念プロセスモデルが定義されているか？		
23	Business model	概念プロセスモデル	System model	論理プロセスモデル	概念プロセスモデルで定義された仕様に基づき，業務手順，入出力，アクターに基づき，論理プロセスモデルが定義されているか？		
24	System model	論理プロセスモデル	Technology model	物理プロセスモデル	論理プロセスモデルで定義された業務フローに基づき，物理プロセスモデルが定義されているか？		
25	Technology model	物理プロセスモデル	Detail representation	プロセスプログラム	物理プロセスモデルに基づき，プロセスプログラムが実装されているか？		
26	Scope	ビジネスゴール	Business model	問題分析モデル	ビジネスゴールに基づき，問題分析モデルが定義されているか？		
27	Business model	問題分析モデル	System model	ビジネスルール	問題分析モデルに基づき，業務要件や運用要件等を考慮して，ビジネスルールが定義されているか？		
28	System model	ビジネスルール	Technology model	ビジネスルール設計モデル	ビジネスルールに基づいて，ビジネスルール設計モデルが定義されているか？		
29	Technology model	ビジネスルール設計モデル	Detail representation	ビジネスルール記述	ビジネスルール設計モデルに基づいて，ビジネスルール詳細仕様が定義されているか？		
コアアセットが満たす解決領域の対応度確認項目数（異抽象度間）: n_{bd}							8
組織が定めた解決領域の対応度確認項目数（異抽象度間）: n_{bs}							13
解決領域からの異抽象度間の対応度: $Y2b$							0.62

*1 レ：該当行の成果物種類①と成果物種類②の間の「対応度に関する品質確認項目」が，確認すべき項目であることを示す．
*2 レ：該当行の成果物種類①と成果物種類②の間の「対応度に関する品質確認項目」は，対象コアアセットが満たしている（または満たす予定）であることを示す．

(平成 19 年 2 月 1 日受付)
(平成 19 年 11 月 6 日採録)

位野木万里（正会員）

1989 年早稲田大学理工学部数学科卒業．1991 年同大学大学院修士課程修了．同年（株）東芝入社．現在，東芝ソリューション（株）IT 技術研究所に所属．ソフトウェア生産技術の研究・開発，開発部門への適用・普及業務に従事．早稲田大学大学院理工学研究科博士後期課程に在籍．IEEE，ACM 各会員．

深澤　良彰（正会員）

1976 年早稲田大学理工学部電気工学科卒業．1983 年同大学大学院博士課程修了．同年相模工業大学工学部情報工学科専任講師．1987 年早稲田大学理工学部助教授．1992 年同教授．工学博士．ソフトウェア再利用技術を中心としてソフトウェア工学の研究に従事．電子情報通信学会，日本ソフトウェア科学会，IEEE，ACM 各会員．