

# 1 プロダクトライン開発の 全体像と要求工学

岸 知二 北陸先端科学技術大学院大学<sup>☆1</sup>

プロダクトライン開発は、ビジネス目的や開発組織の状況に応じて体系化されるものであり、個々の開発形態はきわめて多様である。しかしながらそれらには共通する本質的な部分がある。本稿ではそうしたプロダクトライン開発の全体像について説明した後、そうした開発の最上流で必要となるスコーピングや要求定義に関する話題を紹介する。

## プロダクトライン開発の全体像

プロダクトライン開発は再利用に基づく開発形態である。したがって、再利用資産の開発と、開発された再利用資産に基づく個別プロダクトの開発との2つの活動が存在する。

図-1は、こうしたプロダクトライン開発の全体像を模式的に示したものである。図の上部が再利用資産の開発を、下部が個別プロダクトの開発を示す。前者をドメインエンジニアリングの活動、後者をアプリケーションエンジニアリングの活動と呼ぶこともある。中間部分が再利用資産であるが、プロダクトライン開発ではコア資産という呼び方をするのも多いのでここではその用語で示している。コア資産は実装で再利用されるコンポーネント<sup>☆2</sup>だけでなく、より上流の要求や設計、文書、テスト資産など、多様なものが含まれ得る。図ではこうした開発やコア資産について、代表的なものとして、要求、設計、実現を取り出して示している。

再利用資産の開発では、まずプロダクトラインへの要求定義を行う。ここでは開発対象となるプロダクト群全体への要求を獲得・分析し、すべてのプロダクトに共通する要求や、プロダクトによって変わり得る要求などを再利用可能な要求として定義する。プロダクトラインア

ーキテクチャの設計では、プロダクト全体に共通するアーキテクチャを設計する。このアーキテクチャは、プロダクト群全体に対する機能的・非機能的要求の達成の基盤となることができ、さらにプロダクトごとの違いに対応できるものでなければならない。再利用可能なコンポーネントの実現では、プロダクトラインアーキテクチャに整合するかたちで再利用可能なコンポーネントを整備する。

一方個別プロダクトの開発では、まずプロダクトへの要求定義を行う。この際にはプロダクトへの要求を獲得・分析しながら、再利用可能なかたちで定義された要求を活用しながら、共通的な要求は引き継ぎ、プロダクトごとにより変り得る要求は取捨選択しながら、対象プロダクトへの要求を定義する。プロダクトの設計では、プロダクトラインアーキテクチャに基づき、プロダクトごとの変更部分を考慮した設計を行う。この際に設計上の選択肢がアーキテクチャ中に用意されていればそれを活用するし、用意されていなければ新たに設計を加えることになる。プロダクトの実現では、再利用可能なコンポーネントを活用し、あるいは必要に応じて新たなコンポーネントを用意して対象プロダクトを開発する。なお、体系的かつ効果的な開発を行うために、要求、設計、コンポーネント間の対応関係を明示的に管理し、トレーサビリティを確保することが重要となる。

このようにプロダクトライン開発では、プロダクトラインに対する要求定義を明確に行い、それに基づいたプロダクトラインアーキテクチャを設計することで、対象プロダクトラインにとって効果的に使われる再利用可能なコンポーネントを開発する点が大きなポイントとなる。以下本稿では、要求定義にかかわる活動について、より詳細に見ていきたい。

## スコーピング

プロダクトライン開発ではコア資産をどういったプロダクトが利用するのか、その範囲を明確に定義する。この

<sup>☆1</sup> 現 早稲田大学 理工学術院 創造理工学部 経営システム工学科

<sup>☆2</sup> 本稿ではソースコード、ライブラリ、フレームワーク、コンポーネントソフトウェアなど実装に使われるさまざまなソフトウェア構成要素をコンポーネントと呼ぶ。

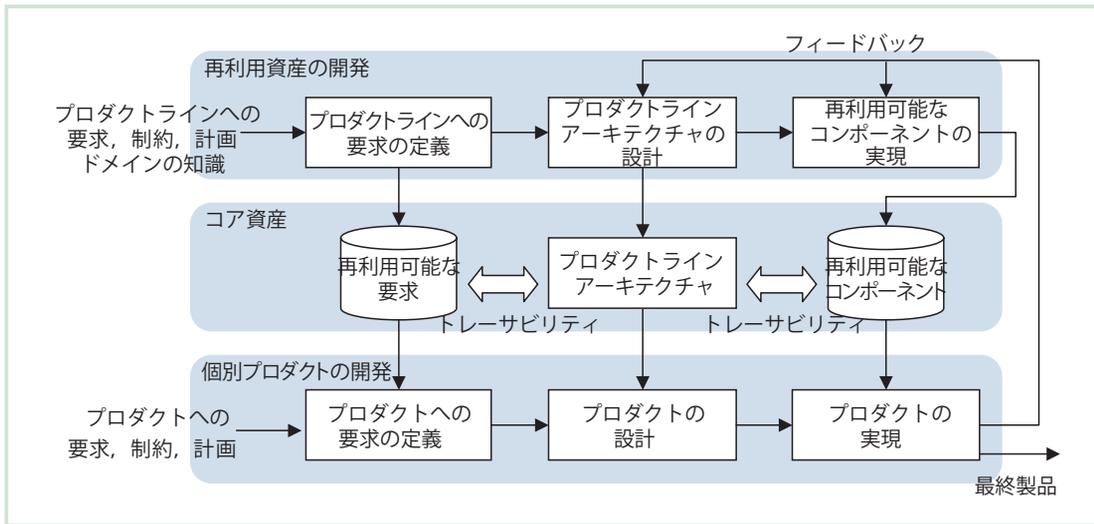


図-1 プロダクトライン開発の全体像

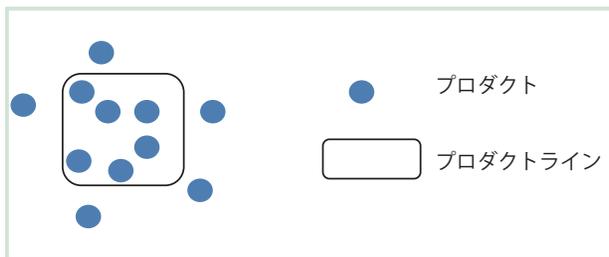


図-2 スコーピングのイメージ

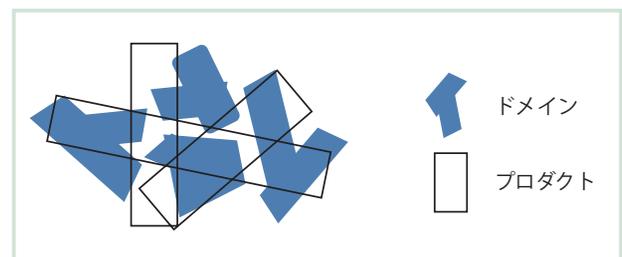


図-3 ドメインと製品の関係

作業をスコーピングと呼ぶ。SEI では、スコーピングは内側に含まれるものおよび外側にあるものの振舞いや特徴を定めることにより、システムあるいはシステム群を境界づけることであると定義している<sup>1)</sup>。

図-2 にスコーピングの直感的なイメージを示す。スコーピングは、コア資産を利用するのはどの製品なのか、どの製品は利用しないのかを明確にするということである。一般に再利用資産の開発において、多くの製品が共通に使えるようにすると、機能はより汎用的・一般的に作らざるを得なくなり再利用効果が小さくなる傾向がある。一方、特定の製品群に特化させるとよりそれらの製品にとって適した作り込みができ、再利用効果が大きくなるのが期待できる。しかしながら一部に特化させることは、再利用の範囲が狭まるということであり、再利用資産を構築するコストが見合うかどうか問題となる。再利用資産開発においては、このトレードオフを考えつつ適切な再利用範囲を明示的に決定する必要がある。

IESE<sup>☆3</sup> の提案するプロダクトライン開発手法である PuLSE (Product Line Software Engineering) 中でのスコーピングの手法<sup>2)</sup>を取り上げて、スコーピングについても少し詳しく見てみる<sup>☆4</sup>。PuLSE ではスコーピングを、再利用への投資効果と開発へのメリットが得られるよう

なプロダクトラインの分野(サブドメインや既存資産)と能力(フィーチャ)を識別し、境界づけるプロセスだと定義している。

ここでドメインとは関係を持つ概念の集まりといった意味だが、たとえば特定の処理、データベース管理、セキュリティなどといった単位で捉えられている。ドメインはさらに階層的にサブドメインによって構成される。個々の製品はいくつかのドメインにかかわるが、それらのかかわり合いを理解・分析しどの部分を再利用対象とすることが有用であるかを決定する。

図-3 は PuLSE の説明で使われるドメインと製品の関係を示す直観的な図である。PuLSE では、スコーピング作業を製品ライン・マッピングと、ドメイン・ポテンシャル・アナリシスという作業に分割している。製品ライン・マッピングは、製品やフィーチャを識別し、全体を概観することである。ここでは熟練者の知識、製品の計画、組織構造、既存システムなどの知識や分析を踏まえながら、どのような

☆3 ドイツ Fraunhofer 研究機関の Institut Experimentelles Software Engineering.

☆4 PuLSE は必ずしもまとまった形で発表されていない。本稿では SPLC2008 でのチュートリアル資料を参考にした部分が多いが、参考文献には入手可能な 2002 年の論文を掲載した。

ドメイン	フィーチャ	製品1	製品2	製品3	製品4
検索	キーワード指示	X	X	X	-
	地図上位置指示	-	X	X	X
通信	シリアル	X	X	-	-
	無線LAN	-	-	X	-
ファイル	HD	-	-	X	X
	CD-ROM	X	X	-	-

図-4 ドメイン・フィーチャ・マトリクスの例

プロダクトを開発するのか、どのようなドメインがそれにかかわっているのかを整理し、ドメイン・フィーチャ・マトリクスを作成する。

図-4はドメイン・フィーチャ・マトリクスの例である。縦方向には各(サブ)ドメインごとにフィーチャを、横方向には製品をそれぞれ列挙し、どの製品がどのフィーチャにかかわっているかを示している。

ドメイン・ポテンシャル・アナリシスでは、こうしたマトリクスを参考にしながら、どのドメインが再利用をする価値が高いかを決定する。ここではドメイン・フィーチャ・マトリクスを参考にしながら、ドメインの成熟度、安定性、プロダクト間での違い、既存資産などに対する考慮を行いつつ、作業を行う。PuLSEでは典型的なスコーピング作業のイメージとして、1週間程度の作業を行うことで、3～10プロダクト、100～150フィーチャ、5～12ドメイン程度を扱うという例示をしている。

スコーピングは、何を再利用資産とし、またそれらがどのプロダクト群から再利用されるかを決定する重要な作業である。その際には、再利用を行うことによる効果を、個々の製品の視点だけでなく、製品群全体の視点で捉えることが重要である。たとえば、再利用資産の利用は個々の製品にとっては必ずしも最適解ではなく、特注的に作った方がよりよい場合もあるかもしれない。そうした場合に、その製品を特注することで高まる製品価値と、そのためのコスト増などを評価して、どちらを選ぶことがビジネス全体として良いかを判断する視点が必要である。

## 要求定義

スコーピングによりプロダクトラインの範囲が定義されると、そのプロダクトラインへの要求を定義する必要がある。一般的に要求を定義するためには、要求の獲得、要求の分析といった作業が必要となる。こうした作業は通常のソフトウェア開発での作業と共通的な部分を含ん

でいるが、プロダクトライン開発では対象が単独プロダクトではなくプロダクトラインであり広範かつ捉えづらくなるため、より明示的な要求定義が求められる。

特にプロダクトラインへの要求定義においては、プロダクトライン中のすべてのプロダクトに対する共通的な要求と、個々のプロダクトごとに変わり得る可変的な要求とを区別して定義することが重要となる。なお、共通性や可変性がどのようなものか、またそれをどうモデリングするか等に関しては、本特集の他の解説に詳しいので、そちらを参照されたい。

PuLSEでは、プロダクトラインを分析し、ドメインモデルを作る手法を提供している。ドメインモデルはプロダクトラインへの要求を捉えるとともに、そのドメインモデルから特定のプロダクトのモデルを導出することが可能であるように構築される。そうした導出の判断をまとめた決定モデルもあわせて作成する。こうした作業においては、適切な情報源から要求を獲得しなければならない。その時点で想定されているスコープ中のプロダクトの情報だけでなく、既存のプロダクトの情報、あるいは対象とするドメインの知識などが不可欠となる。そのためにはドメインのエキスパートからの情報収集や、既存のシステムからのリエンジニアリングなどが有用となる。

またKangらはこうした要求定義においては、市場分析や戦略といったビジネス視点と、機能や非機能面の要求やプロダクトの配布の方法といった技術視点との両面からの検討が重要であると指摘している。たとえばプロダクトごとにバイナリが違うのか、同じバイナリを設定ファイルやパラメータで切り替えるのかといったプロダクトごとの作り分けの方法によって、製品の形態や配布の方法が異なってくるかもしれない。再利用技術というのは単に作る側だけの問題ではなく、それによって運用や販売の形態も変わってくる。こうした課題について、開発の早期に双方の視点から多面的に検討することが重要である。

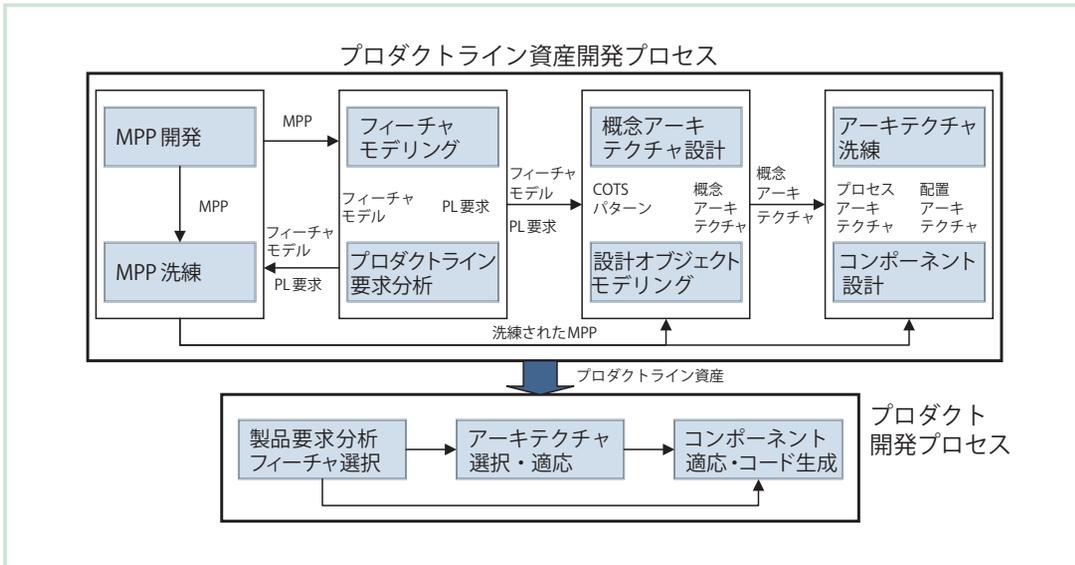


図-5 FORMの全体像

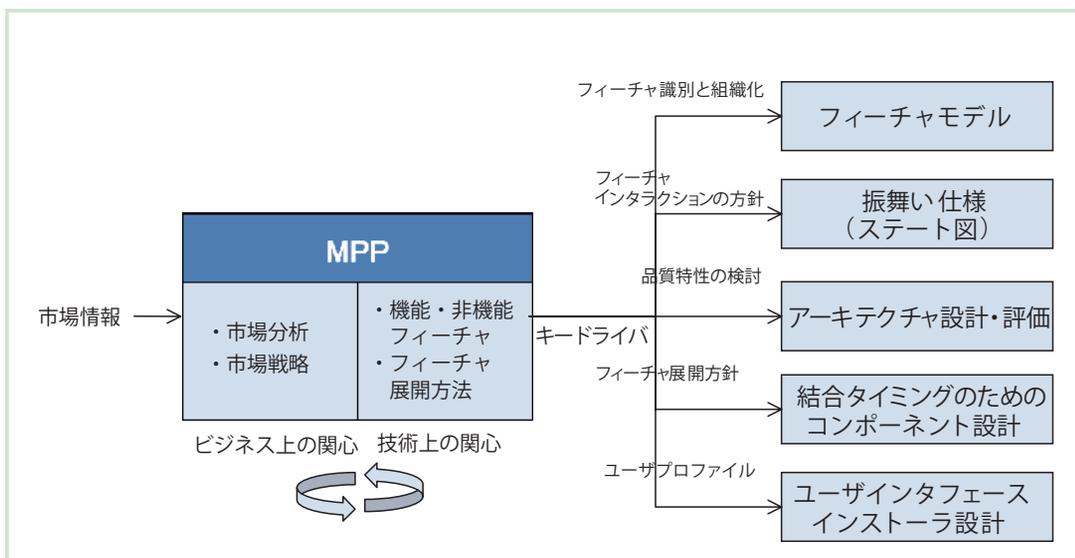


図-6 FORMのMPP

このように、プロダクトラインの要求定義においては、さまざまな情報源からプロダクトライン全体の要求を獲得、理解すること、プロダクトに共通する要求と、プロダクトごとに変わり得る要求とをモデリング技術などを活用しながら明示的に捉えること、またその際にはビジネス面と開発面の両方から十分に吟味することなどが重要となる。こうして作られたモデルは、再利用資産の開発におけるプロダクトラインアーキテクチャの設計、あるいは個別プロダクトにおけるプロダクトへの要求定義などの活動で活用されることになる。

### プロダクトライン開発の手法

プロダクトライン開発のための手法がいくつか提案されている。その中からいくつかを簡単に紹介する。

### ● FODA/FORM

FODA (Feature-Oriented Domain Analysis) はKangらの提案する手法で、ドメイン分析、フィーチャ分析、フィーチャモデルという3つの活動から構成される。プロダクトライン開発という用語が使われる以前に提案された。その後これを設計や実装にまで拡張し、FORM (Feature-Oriented Reuse Method)<sup>3)</sup> という手法を発表している。

図-5にFORMの全体像を示す。FORMでは最上流で、MPP (Marketing and Product Plan) という活動があり、ビジネス視点と開発視点からの検討方法を示している。

図-6はMPPの概要である。ここでは、市場分析や市場戦略といったビジネスの観点からの検討と、機能・非機能フィーチャやフィーチャの展開方法といった開発の観点からの検討を協調しつつ進める。この検討に基づいて、フィーチャモデル、振り舞い仕様、アーキテクチャ

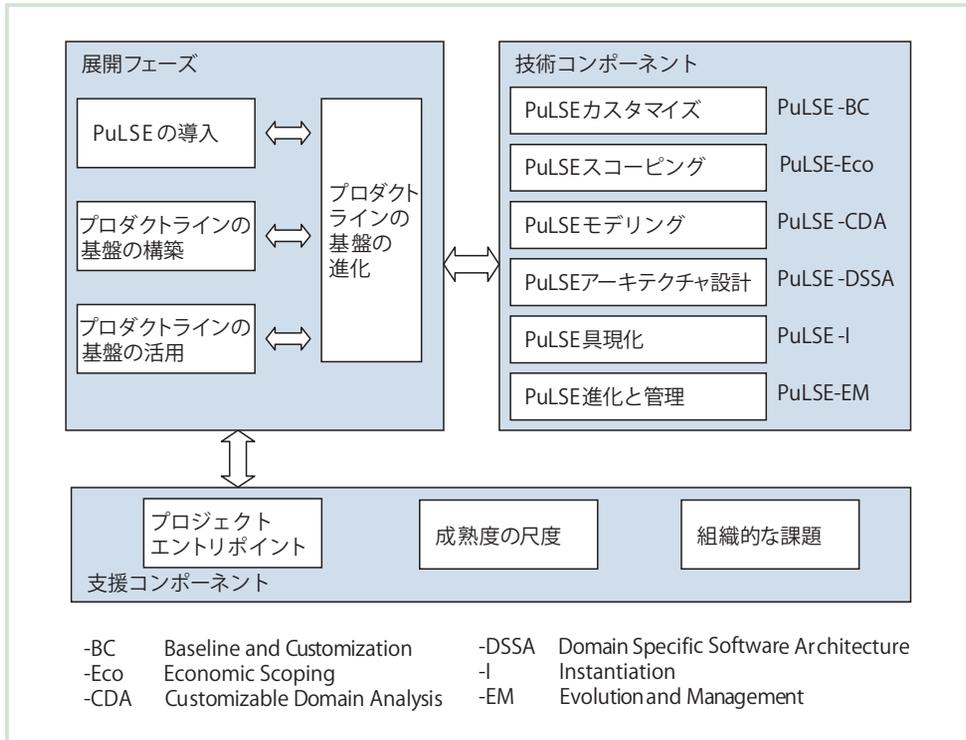


図-7 PuLSEの全体像

設計、コンポーネント設計、ユーザインタフェースやインストールの設計へとつなげる。

●PuLSE

IESEの提唱するPuLSE<sup>☆5</sup>は、プロダクトライン開発の全体を対象とした手法である<sup>4)</sup>。組織に応じたカスタマイズができるようにモジュール化された構造となっている。

図-7にPuLSEの全体像を示す。展開フェーズはPuLSEの導入(カスタマイズ)、プロダクトラインの基盤の構築(再利用資産の開発)、プロダクトラインの基盤の活用(個別プロダクトの開発)、およびプロダクトラインの基盤の進化(基盤や管理方法の進化)という4つの活動から構成される。技術コンポーネントや支援コンポーネントはそうした活動を支援するものであり、目的に応じて取捨選択して使われる。

技術コンポーネントには、PuLSEのカスタマイズ、スコーピング、ドメインモデリング、アーキテクチャ設計、具現化、進化と管理といったモジュールが用意されている。支援コンポーネントにはプロジェクトエントリーポイント(導入のパターンにあわせたカスタマイズ)、成熟度の尺度(プロダクトラインを導入する際の尺度)、組織的な課題(開発や管理のガイドライン)が用意されてい

る。前述した、スコーピングの手法は、技術コンポーネント中のPuLSE-ECO (Economic Scoping)、プロダクトラインの分析やドメインモデルの作成は同じくPuLSE-CDA (Customizable Domain Analysis)に対応する。

●SEIのフレームワーク

米国SEIではソフトウェアプロダクトライン実践のフレームワーク(A Framework for Software Product Line Practice)として開発の枠組みを提示している<sup>5)</sup>。この中で、プロダクトライン開発の本質的な活動としてコア資産開発、プロダクト開発、および管理の3つを挙げ、それらの活動を遂行するために必要な活動を、3つのプラクティスエリアとしてまとめている。スコーピングや要求工学は、コア資産開発の活動に含まれている。

図-8に3つのプラクティスエリアを示す。ソフトウェア工学プラクティスエリアは、コア資産やプロダクトを生成し進化させるために必要な技術を適用するためソフトウェア工学、それらを生成・進化させるために必要な管理のための技術管理、およびプロダクトラインの活動を調整・組織化するために必要な組織管理の各プラクティスエリアから構成されている。表-1に各プラクティスエリアの内容を示す。

展望

以上、プロダクトライン開発の全体像と、その最上流にあたるスコーピングと要求定義について紹介した。

☆5 PuLSEのスコーピングの資料と同様の理由で、本稿ではいくつかの文献を参考にしたが、参考文献では入手可能な比較的初期の論文を掲載した。

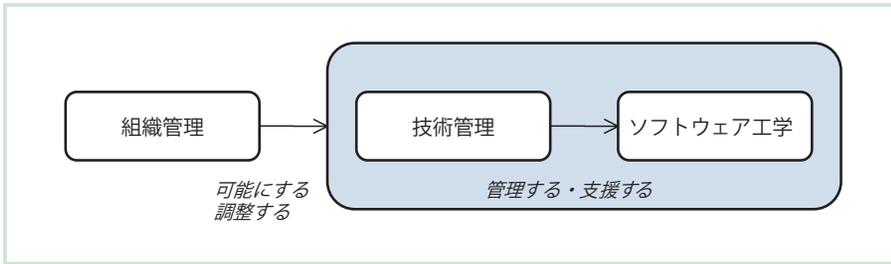


図-8 SEIのプラクティスエリア

プラクティスエリア	内容
ソフトウェア工学	アーキテクチャ定義, アーキテクチャ評価, コンポーネント開発, 既存資産のマイニング, 要求工学, ソフトウェアシステム統合, テスト, 関連ドメインの理解, 利用可能な外部ソフトウェアの利用
技術管理	構成管理, 開発・購入・マイニング・委託の分析, 測定と追跡, プロセスの規律, スコーピング, 技術面での計画, 技術面でのリスク管理, ツール支援
組織管理	ビジネス事例の構築, 顧客インタフェース管理, 獲得戦略の開発, 資金調達, 開始・準備, 市場分析, 運営, 組織計画, 組織面からのリスク管理, 組織の構造化, 技術予測, 要員教育

表-1 プロダクトライン開発のプラクティスエリア

プロダクトライン開発に関する成功事例<sup>6)</sup>として、Celcius Tech社の艦艇システムであるShip System 2000、米国Naval Research Laboratoryの攻撃機A-7Eの操作プログラム、Bell研、Philips社、Ericsson社などの交換機などの事例が紹介されたこともあり、どちらかという重厚長大な開発にしか使えないのではないかという印象を持たれた時期もあった。しかしながら、そうした分野だけではなく、HP社のプリンタファームウェア開発、LSIロジック社のRAID制御ファームウェア、Nokia社の携帯電話、Philips社のテレビや医療画像装置、Cummins社やBosch社のエンジン制御システムなど、多様な成功事例が発表されている。また従来はエンジニアリング分野の事例が多かったが、ビジネス系やエンタープライズ系の事例報告もなされている。

そうした背景には、プロダクトライン開発の実践方法が多様化してきていることが挙げられる。先行投資としてコア資産を開発するような形態だけでなく、既存資産をコア資産へと再構築したり再利用資産を徐々に整備したりしながら緩やかに移行する方法などが検討され実践されている。また比較的ライトウェイトに実践することで小規模なプロジェクトでの成功事例も多く報告されるようになってきた。またPhilips社の医療画像装置のようにX線、磁気共鳴、コンピュータ断層といった複数のプロダクトラインをさらに束ねてプラットフォームを整備したりするなどより高度な実践もなされている。

一方、研究動向としては要求工学、アーキテクチャ設

計、管理技術といったそれぞれの側面から研究がなされているが、たとえば変動性のモデリング、アーキテクチャの評価や移行、効果の見積もりといった観点の研究などがある。また近年ではモデル駆動開発との接点などの議論も活発である。

プロダクトライン開発は、すべてのソフトウェア開発に適したものではないが、ビジネス観点と技術観点のバランスの中で組織目的にあわせた体系的な再利用をするための技術であり、最も進んだ開発形態の1つであるといえる。今後より一層の進展が期待される。

参考文献

- 1) <http://www.sei.cmu.edu/>
- 2) Schmid, K.: A Comprehensive Product Line Scoping Approach and its Validation, Proceedings of the 24th International Conference on Software Engineering, pp.593-603 (2002).
- 3) Kang, K. C., Lee, J. and Donohoe, P.: Feature-Oriented Software Product Line Engineering, IEEE Software, Vol.19, Issue: 4, pp.58-65 (July/Aug 2002).
- 4) Bayer, J. et al.: PuLSE: a Methodology to Develop Software Product Lines, Proceedings of the 1999 Symposium on Software Reusability, pp.122-131 (1999).
- 5) Northlop, L. M.: SEI's Software Product Line Tenets, IEEE Software, Vol.19, Issue: 4, pp.32-40 (July/Aug. 2002).
- 6) <http://www.splc.net/>

(平成 21 年 2 月 23 日受付)

岸 知二 (正会員)

tkishi@jaist.ac.jp

京都大学大学院情報工学科修士課程修了, 北陸先端科学技術大学院大学情報科学研究科博士課程修了, 博士(情報科学)。電気メーカ勤務を経て, 2003年より北陸先端科学技術大学院大学情報工学科, ソフトウェア設計・検証, プロダクトライン開発に興味を持つ。