

ソフトウェアプロダクトライン開発の 導入時における QFD の活用

吉田愛[†] 大原貴都[†] 近久真章[†] 小川秀人[†]

多くの分野の組込みシステムが様々なニーズに対応するため多品種化されている。一方で、開発コスト削減、開発期間の短縮、信頼性向上などへの要求はますます強くなっている。これらの要求に対応するために、組込みソフトウェア開発に対して様々な施策が講じられているが、近年高い注目を集める再利用パラダイムに Software Product Line Engineering (SPLE) がある。日立グループ内でも SPLE によってソフトウェア開発において成果が挙げられている。

SPLE は開発上流から下流までの様々な技術を含む概念であることから、SPLE の導入にあたって適用する技術の選定には SPLE を実践した者の経験やノウハウを要していた。本稿では、日立グループの SPLE に対する知見の集合と言える SPLE ガイドラインと、世界的に広く用いられている品質機能展開(QFD)を組み合わせて、SPLE 技術の導入手順を決定する方法と事例を示す。本方法を用いることで、SPLE を効果的に導入できる。

QFD based method for selecting activities in Software Product Line Engineering

Ai Yoshida[†] Takatoshi Ohara[†] Masaki Chikahisa[†]
and Hideto Ogawa[†]

Embedded systems in many fields have a wide variety of products to support various requirements. On the other hand, requirements of reduction in development cost, reduction in development time and improvement of product reliability have become increasingly strong. To make these requirements, we implement various measures for Embedded Software Development. In particular, Software Product Line Engineering (SPLE) which is software reuse techniques has received attention in recent years. In Hitachi Group, we introduce SPLE and make a result.

SPLE is a comprehensive concept of technology in upstream and downstream development. Therefore, in order to select activities in introducing SPLE, it required the experience and expertise of engineers who practiced SPLE. In this paper, we describe a method for selecting activities in introducing SPLE and an example. In the method, we combine the SPLE guideline in the Hitachi Group and QFD. Using this method, it can support to introduce SPLE.

1. はじめに

組込みソフトウェアが複雑化、大規模化することで、ソフトウェア開発のコスト増大や開発期間の遅延などが問題となっている。ソフトウェア開発のコスト削減と開発期間の短縮に向けた対策として、ソフトウェアの再利用が検討されている。複数製品にわたってソフトウェアを再利用することで、ソフトウェアの開発効率を高めるという考え方は古くから存在する。例えば、ソースコードの再利用として関数のライブラリ化、設計の再利用としてデザインパターンやソフトウェアアーキテクチャなどが挙げられる。近年では、この考え方をさらに推し進め、組織的、計画的にソフトウェアを再利用するソフトウェアプロダクトライン開発 (Software Product Line Engineering; 以下、SPLE) と呼ばれる開発手法が注目されている[1][2]。SPLE は製品系列における共通性、可変性を利用してソフトウェアの構造化、部品化を行い、再利用性を向上させる。また、そのソフトウェアを管理する体制を敷くことで、継続的な効率改善を図る。様々な製品メーカーにおいて SPLE の成功事例が報告されている[2][3][4][5][6]。同様に、日立グループでも SPLE を適用し、ソフトウェア開発を効率化しようという動きが活発化している。実際に、医用分析装置や自動車用エンジン制御システムなどに SPLE を適用している[7][8][9][10]。これまでの適用例では、SPLE を導入する際に、具体的な作業内容は製品毎に検討されている。そこで、再利用の規模や市場の予測可能性に基づいて SPLE の形態を分類する、SPLE の3つの適用アプローチが提案されている[11]。この分類を利用し、製品の状況に応じて SPLE の適用を検討できる。しかし、市場調査や先行投資の見積もりによって、SPLE の形態を決めることはできるが、実務上の具体的な作業内容は定義していない。つまり、SPLE では開発の体系を示してはいるが、現状ではどのように既存製品を SPLE に移行させるか、何を主眼において SPLE を導入するかについては、製品毎に検討されている。

そのため、SPLE に効果があることがわかっていても、特に、これまで SPLE を実践したことのない開発現場では、導入には高い障壁がある。日立グループでは、この状況を鑑みて、SPLE に関するワーキンググループを立ち上げた。そして、ワーキンググループの活動の一環として、SPLE を実践して培ってきた知見を集約した SPLE ガイドラインを作成している。SPLE ガイドラインには SPLE を適用するにあたって実施すべき具体的な項目がまとめられている。この SPLE ガイドラインを元にさらに SPLE の推進に取り組んでいる[12]。定義している全ての実施項目を適用することが望ましいが、同時に全ての実施項目を実践することはコストの面から現実的ではない。

[†] 株式会社 日立製作所 横浜研究所

Hitachi, Ltd., Yokohama Research Laboratory

そのため、実施項目の中で優先付け SPLE に対する開発現場の要求を踏まえて実施項目を決定する方法が課題となっている。

本報告では、品質機能展開(Quality Function Development; 以下, QFD) [13][14][15]を用いて SPLE の導入手順を決定する方法を検討し、効率的に SPLE を導入する手法を提案する。提案手法では、QFD を利用することで SPLE ガイドラインで示される具体的な実施項目を優先付けし、SPLE の導入を効率化する。

2. SPLE に関する日立グループにおける取り組みと課題

2.1 SPLE の概要

SPLE は、ある市場分野の製品において、複数の製品で共通利用されるコンポーネントや基盤となるアーキテクチャを構築し、その資産を用いることにより個々の製品を開発するソフトウェア開発手法である。資産としてはソフトウェアのコンポーネントのみではなく、製品に対する要求、試験、開発プロセス等も含んでいる。1990年代から欧米を中心に研究、適用がされ、体系化が進められている[16]。SPLE は、以下の3つの活動から構成される[3]。

・コア資産開発

複数の製品で共通利用されるコア資産を開発する。まず、どのようなコア資産が必要になるかを定義する。そのために、開発対象となる製品の品種群の要求仕様を分析して、製品に共通な要求仕様と、品種によって変化する要求仕様を獲得する。次に、その要求仕様を実現できるように、ソフトウェア全体の構造を決め、製品開発で利用するためのコア資産を開発する。

・プロダクト開発

コア資産開発で作りに出された資産を適用することにより、再利用に基づいた個々の製品開発を行う。まず、品種の要求仕様を獲得する。次に、コア資産開発で開発したソフトウェア部品を利用し、要求仕様を実現する。コア資産で実現できない場合は必要に応じて製品独自の機能を開発する。

・マネジメント

コア資産開発とプロダクト開発のリソースを調整し、コア資産を再利用した製品開発を推進し、管理する。具体的には SPLE 全体計画の策定、コア資産開発チームやプロダクト開発チームの編成、顧客対応などである。

		SPLE Level 1	SPLE Level 2	SPLE Level 3	SPLE Level 4	SPLE Level 5
コア資産開発	[Key area 1] スコープ定義		[Activity 1-a] SPLEの対象とする製品群			
	[Key area 2]		[Activity 2-a]	[Activity 2-b]	[Activity 2-c]	
	[Key area 3]		[Activity 3-a]	[Activity 3-b]	[Activity 3-c]	
					[Activity 3-d]	
	[Key area 4] ソフト部品開発		[Activity 4-a] レガシー解読	[Activity 4-c] 可変部品開発	[Activity 4-e] 使用形態の観察	
		[Activity 4-b] 共通部品開発	[Activity 4-d] 可変メカニズムの設計			
	[Key area ...]		[Activity ...]			
プロダクト開発	[Key area ...]		[Activity ...]	[Activity ...]	[Activity ...]	
				[Activity ...]		
マネジメント	[Key area ...]		[Activity ...]	[Activity ...]	[Activity ...]	
			[Activity ...]			

図 1 SPLE ガイドライン

また、Family Evaluation Framework (FEF) と呼ばれる SPL 評価フレームワークが提案されている[17]。これは、組込みシステム(医療機器、携帯電話、自動車部品等)を対象とした欧州の産学協同プロジェクト(ESAPS, CaFÉ, FAMILIES)[18]の成果である。FEF は BAPO モデルによる分類を用いている。BAPO とはビジネス(Business)、アーキテクチャ(Architecture)、プロセス(Process)、組織(Organization)を表し、それぞれの観点について個々にレベルを評価することが特徴となっている。FEF は(1)取り組みが正しい方向に進んでいることの証明手段、(2)SPLE レベルを他社と比較するためのベンチマークツール、(3)プロセス改善のための意思決定ツールとして用いることを想定している。FEF はソフトウェア開発プロセス能力成熟度モデルであるCMMI(Capability Maturity Model Integration)[19]と同様、製品毎のレベルを評価できる。

2.2 日立グループの取り組み

日立グループでは、SPLE を様々な製品のソフトウェア開発において実践している。例えば、医用分析装置の開発プロジェクトに SPLE を適用し、情報処理部アーキテクチャを設計および作成している[9][10]。また、SPLE のコア資産開発において、複数

の自動車用エンジン制御システムのソフトウェアを分析し、製品間の可変性、共通性を評価している[11]。その他、搬送装置[12]や他のシステムなど、SPLE の適用範囲を拡大している。

しかし、これまでの活動は、対象製品や組織に合わせて実施内容を調整して決定しており、SPLE 実施者の経験やノウハウに依存するところが大きかった。そのため、SPLE を適用したことがない開発現場では、具体的に何から始めれば分からないといった問題に直面していた。

そのため、日立グループでは、これまでに様々な製品で培ってきた知見を集約し、共有することを目的とした、SPLE に関するワーキンググループを立ち上げた。ワーキンググループでは、活動の一環として、SPLE を実践した知見やノウハウを SPLE ガイドラインとしてまとめている。SPLE ガイドラインへ知識を集約することで、属人性を排除した効率的な SPLE の適用を実現することを目指している。

SPLE ガイドラインでは、FEF をベースとして、FEF では規定されていない、より詳細な技術をマッピングしている[14]。図 1 に SPLE ガイドラインの概要図を示す。SPLE ガイドライン内の、左端の行(Key area)は、SPLE における技術分野を示しており、コア資産開発、プロダクト開発、マネジメントの活動をさらに細分化している。また、上部の列(SPLE Level)は FEF に対応した SPLE レベルを設定している。そして、表内(Activity)には各 SPLE レベルにおける技術分野の具体的な実施項目（アクティビティ）とユースケースをまとめている。

2.3 課題

日立グループでは、SPLE ガイドラインを利用し、SPLE の推進に取り組んでいる。例えば、SPLE ガイドラインを評価分析に利用することで、現状を認識することや到達する SPLE レベルを設定するための参考としている。

しかし、SPLE ガイドラインでは SPLE のアクティビティを計 57 項目定義しているため、現状を分析した結果を踏まえても、今後どのアクティビティを選択するかを判断することは難しい。定義している全てのアクティビティを適用することが望ましいが、同時に全てのアクティビティに取り組むことはコストの面から現実的ではない。また、SPLE の導入を検討している製品毎に開発期間短縮や製品品質向上など SPLE に対するモチベーションが異なる。そのような状況において、どのアクティビティから着手すべきかを判断する方法がない。

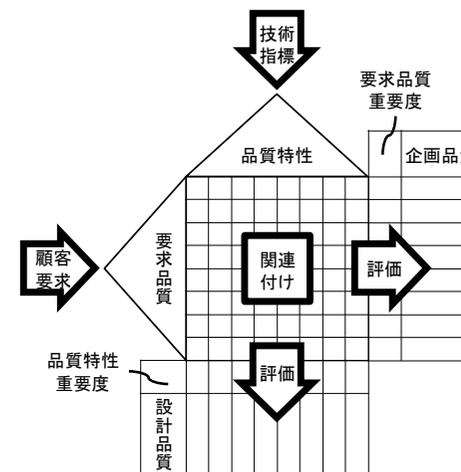


図 2 品質展開表

3. 提案手法の概要

3.1 課題の解決方針

本報告では、開発現場の要求を踏まえて、SPLE ガイドラインからアクティビティを選択する手法を提案する。複数人からの複数個の要求を整理し、複数の技術と対応させて、定量的な方針決定を促す手法として QFD が提案されている。QFD は二次元マトリクス計算をベースとした要求分析手法で、顧客の要求を技術的指標に変換することで、定量的な判断材料を与える。図 2 に典型的な品質展開表を示す。今回は、開発現場の要求と SPLE ガイドラインのアクティビティを優先付けし、開発現場の要求と SPLE ガイドラインとの対応を取るために QFD を利用することとした。つまり、図 2 に示す、要求品質が開発現場の要求項目となり、品質特性が SPLE ガイドラインのアクティビティとなる。

QFD を利用する利点はその他にも考えられる。QFD は、品質特性の項目間で依存関係を示すことができる。そのため、SPLE ガイドラインのようなアクティビティに SPLE レベルという段階的な依存関係がある場合においても、その依存関係に応じてアクティビティの重み付けができる。また、QFD は品質展開表を作成することで、要求品質と品質特性の関連を可視化しており、対応を確認しやすい。そのため、ある品質特性を選択した場合、その結果解決される要求品質がわかる。今回の提案手法を用

		SPLE ガイドライン				
	要求項目 重要度	[Activity 1-a]	[Activity 2-a]	[Activity 2-b]	[Activity 2-c]	[Activity ...]
要求項目1	5				○	
要求項目2	2	◎	△			
要求項目3	3		△			
要求項目4	4			○		
要求項目5	2		○		△	
要求項目6	3	○				
	品質特性 重要度	19	11	12	17	...

対応関係 換算表	
◎ 5	:強い対応
○ 3	:対応あり
△ 1	:対応が予想

図 3 作成される品質展開表の例

いて、ある一時点において品質展開表を作成し、あるアクティビティが実施され、ある要求項目が解決されるとする。その結果、品質展開表を見直し、その時点での要求項目の追加要求項目重要度の変更を踏まえて品質展開表を修正することで、再度次に実施すべきアクティビティを決定できる。つまり、QFD は継続的に利用できる。SPLE では SPLE レベルによって段階分けがされていることからわかるように、継続的に改善要求を汲み取って、時系列的な改善活動していくことでより効果が得られる。以上のような QFD の特徴をふまえて、提案手法には QFD を応用することは適していると判断した。

3.2 QFD に基づく定量的な SPLE 導入項目決定方法

QFD を利用することで、SPLE ガイドラインから実施すべきアクティビティを選択する方法の手順を述べる。

まずは、SPLE に対する開発現場の要求を明らかにする。明らかになった要求に対して品質展開表を用いて、SPLE ガイドラインに定義しているアクティビティの優先付けをする。提案手法の結果として、QFD により図 3 に示すような二次元表が作成される。QFD を利用する実施手順は下記の 5 項目である。

- (手順 1) 製品担当者へのヒアリング
- (手順 2) ヒアリング結果のグルーピング

- (手順 3) 要求項目重要度の設定
- (手順 4) 要求項目と SPLE ガイドラインとの対応付け
- (手順 5) 品質特性重要度の算出

(手順 1) 製品担当者へのヒアリング

(手順 1)の目的は、ヒアリングを実施することにより、ヒアリング実施者が現状の開発状況を把握するとともに、開発現場の抱える課題を抽出することである。ヒアリングでは、主に BAPO に関する内容について質問する。(手順 1)の結果、製品担当者の要求の生データを得る。

(手順 2) ヒアリング結果のグルーピング

(手順 2)の目的は、ヒアリング結果をグルーピングすることにより、開発現場の真の要求を洗い出すことである。ヒアリングから得た情報は、ヒアリング参加者である製品担当者の経験に基づく具体的な内容が多い。つまり、参加者は必ずしも真の要求を述べているわけではない。従って、参加者が発言に至った原因を想定することで、要求の本質を捉える必要がある。

まずは製品担当者の要求の生データを整理する。データを整理するために、参加者の発言をより端的な表現に直す。そして、関連する他の要求項目でグルーピングする。グルーピングは各要求項目が現れたと想定した原因毎に行う。(手順 2)の結果、開発現場の要求項目を得る。

(手順 3) 要求項目重要度の設定

(手順 3)の目的は、(手順 2)で抽出した複数の要求項目を重要度毎に分類することである。(手順 2)まででは、それぞれの要求の重要度はわからない。要求項目の中には、ヒアリング参加者の個人的な内容や思い込みなども含まれる場合もある。そのため、製品担当者間で要求項目をレビューしながら、要求項目に重要度を付けていく。(手順 3)の結果、要求項目に対応する要求項目重要度を得る。

(手順 4) 要求項目と SPLE ガイドラインとの対応付け

(手順 4)の目的は、(手順 5)で品質特性重要度を算出するために、(手順 2)でまとめた要求項目と SPLE ガイドラインの対応付けをすることである。

要求項目を縦軸に SPLE ガイドラインを横軸に設定した二次元表を作成する。要求項目と SPLE ガイドラインとの対応付けは、SPLE ガイドラインのユースケースを参考に行う。要求項目と SPLE ガイドラインに対応関係があれば、二次元表の縦軸と横軸の交点に対応関係を示す記号を記入する。(手順 4)の結果、要求項目と SPLE の対応関係の表を得る。

(手順 5) 品質特性重要度の算出

(手順 5)の目的は、要求品質表から品質特性重要度を算出することで、SPLE ガイドラインの定義するアクティビティの中から、開発現場の要求を満たす項目を選択する

ことである。SPLE ガイドラインのアクティビティである各列において、これまでの手順で規定した要求項目重要度と対応関係を示す記号を換算した数値との積算結果を和算して品質特性重要度を算出する。(手順5)の結果、得られた品質特性重要度に従って、SPLE ガイドライン中から、実施するアクティビティを選択できる。

4. 検証

提案手法をある組込みシステムにおいて実践した。今回対象とするのは3製品(製品A, 製品B, 製品C)である。3製品はある機能を同様に有している。例えば、製品Aで開発した機能を他の製品Bや製品Cで流用して開発をしている。ヒアリング参加者は製品A, 製品B, 製品Cの各担当者の計5名である。また、ヒアリング実施時間は、1名あたり約1時間とした。提案手法の各手順を実践した内容を下記に示す。

(手順1) 製品担当者へのヒアリング

ヒアリングに備えて、ヒアリングの内容を参加者間で合わせるため、ヒアリングの必須項目を事前検討した。項目はBAPOの視点を踏まえて作成した。ヒアリングの際は必須項目を起点とし、内容を掘り下げて話を聞いた。また、ヒアリングの際には、書記を配置した。質問への回答以外の会話の中に、重要な意見が隠れている可能性があるため、書記は可能な限り、全ての会話を書き留めた。これは、必要に応じて、ボイスレコードなどの機器を用いることで代用できる。

(手順2) ヒアリング結果のグルーピング

まずは、ヒアリングで得た内容から、要求となる項目を抽出し、端的な表現に直した。具体的には、1文に2つ以上の内容を有することのないように分割して、表現を修正した。各項目について、ヒアリング参加者が発言に至った原因を想定した。そうすると、表現が異なる2者間の発言でも、本質的には同じ原因から起因しているということもあった。原因毎に各項目をグルーピングし、グルーピングの結果を要求項目とした。

(手順3) 要求項目重要度の設定

(手順2)で作成した要求項目について、ヒアリング参加者以外の製品担当者も含め、レビューを実施した。レビューでは、開発現場として要求項目に抜け漏れがないか、参加者間で要求項目の内容について認識の差はないかを確認した。確定した要求項目に対して、今回は、重要度を5段階(5:重要である, 1:重要ではない)で設定した。

(手順4) 要求項目とSPLEガイドラインとの対応付け

要求項目とSPLEガイドラインの対応付けについては、SPLEガイドラインの要求技術を実現したとき、要求に対する効果の有無を考慮して設定した。今回は、対応がある場合を「○」、対応がない場合を「無印」、要求達成の妨げになる場合を「×」の

印で表記した。例えば、「開発工程を簡略化したい」という要求項目がある場合、あるプロセスが追加されるようなアクティビティに対しては、必ずしも開発工程の簡略化するという要求を満たすことに繋がらない可能性がある。そのような場合が考えられる際に、要求達成の妨げになるという対応関係を表記した。

(手順5) 品質特性重要度の算出

(手順4)で付加した印を、「○」を5点、「無印」を0点、「×」を-5点と点数換算し、品質特性重要度を算出した。算出した品質特性重要度を含め、品質展開表を作成した。結果、あるアクティビティが選択された。

5. 考察

実験結果から、SPLEガイドラインの中の中でも、ソースコードのアーキテクチャ設計に関するアクティビティが選択された。SPLEを実現するアーキテクチャは、複数製品にわたって、共有するものである。しかし、現状では3製品間のアーキテクチャは異なる。SPLEを適用するために、まずは3製品間で共通利用できるアーキテクチャの設計を目指すことは現状を踏まえて考えても適切である。よって、これまで製品毎に実施者の経験やノウハウに応じて独自に取り組んでいたSPLE導入に対して、定量的な優先付けをすることで、アクティビティの選定が効率的に実施できると考える。

一方、今回選択されたアクティビティはSPLEガイドラインにおいてSPLEレベル4に設定されている。SPLEガイドラインのSPLEレベルは段階的に実現することを想定しているため、SPLEレベル4を達成するためにはSPLEレベル2のから順々に実現することになる。これは、SPLEレベルが高い方がより与える効果が高いため、対応付けをしていくと、単純に対応関係が多くなり、品質特性重要度の値が大きくなるためと考えられる。そのため、今後SPLEレベルを考慮して、アクティビティ間での重み付けを設定するような対応が必要である。具体的にはSPLEガイドラインのアクティビティに実現するための難易度を設定し、それを重要度に反映するような仕組みを用意するなどである。

また、手順の中での留意点がいくつか見えてきた。

その他、(手順1)において、ヒアリングの中でSPLEは対象製品を製品群として扱うこととなる。そのため、製品固有の話だけでなく、他の製品との連携についても聞くことで、アクティビティが決定した後の考慮材料として役に立つ情報である。また、製品固有の話なのか、他の製品とも関連した話なのかを切り分けておくことで、要求項目をまとめた後に、製品全てで同じ要求が出ている場合は重要な要求である、逆にある製品固有の要求であれば影響は少なく重要度は小さいなど、(手順3)の作業で重要度を設定する際の一助となると考えた。

また、(手順 4)において、今回は SPLE に初めて取組む開発現場において実践したため、対応付けは SPLE の有識者が実施した。SPLE に関して知識が乏しい開発現場では、対応付けの段階で実施が難しくなることが予想される。しかし、この問題については、アクティビティに関して、作業内容だけではなく、その効果について記載することで対応できると考えている。一方で、初めて取組む開発現場においては、対応付けの段階で、有識者が SPLE ガイドラインの各アクティビティの説明を通して、共に対応付けを実践することで SPLE の知識を共有し、普及することができる。そのため、一度実践すれば、今後開発現場内で同様に実践できると考えている。

6. おわりに

本報告では、これまで製品毎に実施者の経験やノウハウに応じて独自に取り組んでいた SPLE 導入に関して、定量的な開発方針決定方法を提案した。具体的には、SPLE 経験者の知見を集約した SPLE ガイドラインから、QFD によって、定量的にアクティビティの優先付けをする方法である。実際に、提案した手法をある組込みシステムの開発現場で実践した。その結果、製品担当者の要求を明らかにしながら、アクティビティを選択できた。そのため、提案手法により、効率的に SPLE を導入できる。

今後は、SPLE ガイドラインの整備を進める。また、作成した品質展開表を SPLE の取り組みと合わせて継続的に使用する方法を検討して行きたい。

参考文献

- 1) John D. McGregor, Dirk Muthing, Kentaro Yoshimura, and Paul Jensen, "Successful Software Product Line Practices," IEEE Software, Vol. 27, No. 3, (2010).
- 2) Klaus Pohl, Gunter Bockle, and Frank van der Linden, "Software Product Line Engineering : Foundations, Principles And Techniques," Springer-Verlag New York Inc, (2005).
- 3) Paul Clements and Linda M. Northrop, Software Product Lines: Practices and Patterns, Addison-Wesley, (2001).
- 4) 吉村健太郎: 製品間を横断したソフトウェア共通化技術 -ソフトウェアプロダクトラインの最新動向, 情報処理学会誌, Vol.49, No.2 (2007)
- 5) Jan Gerben Wijnstra, "Component Frameworks for a Medical Imaging Product Family," Proceedings of the International Workshop on Software Architectures, for Product Families (2000)
- 6) Robert L. Nord, "Meeting the Product Line Goals for an Embedded Real-Time System," Proceedings of the International Workshop on Software Architectures for Product Families, (2000).
- 7) 竹辺靖昭, 近久真章, 島袋潤, 高木由充, 塙 俊英: 多様な製品展開を支える再利用型組込みソフトウェア生産技術, 日立評論, Vol.91, No.5 , p.422-425, (2009).

- 8) Yasuaki Takebe, Naohiko Fukaya, Masaki Chikahisa, Toshihide Hanawa, and Osamu Shirai, "Experiences with Software Product Line Engineering in Product Development Oriented Organization," SPLC '09 Proceedings of the 13th International Software Product Line Conference, p.275-283.
- 9) 吉村健太郎, ガネサンダルマリンガム, ムーティック ディルク: プロダクトライン導入に向けたレガシーソフトウェアの共通性・可変性分析法, 情報処理学会論文誌, vol.48, No.8, p.2482-2491, (2007).
- 10) 明神智之, 深谷直彦, 小川秀人: 段階的開発指向型組織におけるソフトウェアプロダクトライン開発の適用, 第 73 回全国大会講演論文集, Vol. 2011, No.1, p.225-227, (2011).
- 11) Charles Krueger, "Eliminating the Adoption Barrier, Point-Counter Point Column," in IEEE Software, (2002)
- 12) Kentaro Yoshimura, Yoshitaka Atarashi, Jun Shimabukuro, Takatoshi Ohara, Chikashi Okamoto, Shinobu Koizumi, Shigeru Watanabe and Kazumi Funakoshi, "Key Activities for Introducing Software Product Lines into Multiple Divisions: Experience at Hitachi," SPLC '11, (2011).
- 13) 赤尾洋二: 品質展開入門, 日科技連出版社 (1990)
- 14) 大藤正, 赤尾洋二, 小野道照: 品質展開法(1), 日科技連出版社 (1990)
- 15) 大藤正, 赤尾洋二, 小野道照: 品質展開法(2), 日科技連出版社 (1994)
- 16) 岸知二, ほか: ソフトウェア再利用の新しい波 -広がりを見せるプロダクトライン型ソフトウェア開発, 情報処理学会誌, Vol.50, No.4, pp.265-310 (2009)
- 17) Frank van der Linden, Klaus Schmid, and Eelco Rommes, "Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering," Springer-Verlag New York Inc (2007)
- 18) Frank van der Linden, "Software product families in europe, The esaps & café projects," IEEE Software, Vol.19, No.4 (2002)
- 19) Mary Beth Chrissis, Mike Konrad, and Sandy Shrum, "JASPIC CMMI V1.1 翻訳研究会訳, CMMI 標準教本," (2005).