

PSP コースへのプロダクトライン技法の適用

日下部 茂
九州工業大学

kusakabe@csn.kyutech.ac.jp

要旨

著者は複数の大学院で個人レベルのソフトウェア開発プロセス PSP のトレーニングコースの講義を担当してきた。例えばその一つの九州工業大学大学院では実践的人材育成の一環として個人/チームの開発プロセス PSP/TSPi の教育を行い、大学院学生レベルでも生産性を落とさずに品質向上を達成するなどの実績をあげている。しかしながらそのようなコースの実施への取り組み方は大学院間で一様ではなく、コロナ禍の影響下での実施方式にも相違があった。さらに、ソフトウェア開発を取り巻く環境、大学に期待される社会的役割の変化などにより、求められるソフトウェアプロセスやその教育の多様性も増している。そのような状況に系統的に対応するために、ソフトウェアプロダクトライン技法を PSP コースに適用する。これまでもソフトウェアプロセスにプロダクトラインの技法を適用する取り組みはなされているが本研究は大学でのプロセスの教育・研究に焦点を当てている点に特徴がある。

1. はじめに

著者は複数の大学院で個人レベルのソフトウェア開発プロセス PSP (Personal Software Process)[1][2]のトレーニングコースの講義を担当してきた。PSP は米国カーネギーメロン大学ソフトウェアエンジニアリング研究所 (SEI) の Watts S. Humphrey により提唱されたソフトウェア技術者のための個人レベルの自己改善プロセスである。PSP のトレーニングコースの受講者は、PSP インストラクタによる講義とプログラム開発の演習、自己分析レポートを通じて、高品質ソフトウェア開発に必要なスキルを習得できる。

例えば、九州工業大学大学院(以降、九工大)では、文部科学省「先導的 IT スペシャリスト育成推進プログラム」の一環として、2007 年より SEI と連携し、PSP と TSP とを教育に取り入れ、高度情報通信技術者の育成に取り組んできた[3][4][5][6][7]。この大学院科目群は、PSP for Engineers コースに基づく PSP コースと、教育向けに設計された Introductory TSP (TSPi) に基づく TSP コース

からなる。九工大の PSP コースは、PSP-Planning と PSP-Quality をベースにした二つの演習科目からなり、PSP for Engineers コースと同様の内容を実施している。もともとの PSP コースは集中講義形式が想定されていたが、九工大では、通常学期中に実施する。受講者は他科目も履修することが想定される中で演習時間を十分確保するため、集中講義形式一日分の内容を一週間で実施し、全体を半期で終えるスケジュールとなっている。コース開設当時 SEI のライセンス制度があり、教員は PSP インストラクタ資格を取得した上で、SEI のライセンスに基づいて講義を行い、修了者には、SEI が実施するコース同様 PSP for Engineers コースの修了証が授与された。このような教育で、大学院生レベルでも生産性を落とさず品質向上を達成するなどの実績をあげてきた。

ライセンス制度下では PSP コースに関する講義資料や演習教材などは、SEI とのパートナー契約に基づいた配布・利用で、加えて一定の条件下で SEI の Web サイトからも一部入手可能という管理がなされていた。しかしながら 2018 年 10 月から Creative Commons ライセンス [8] の下、SEI Digital Library [9] から入手できるようになった。また SEI のライセンス制度も廃止されコース実施の自由度が増した。九工大の PSP コースも、コアの部分はもともとの PSP コースに従いつつ、SEI のライセンス要件に縛られずに実施するようになった。

産業界の実務者を想定した PSP トレーニングを大学院で実施すると、想定の違いに対処する必要が生じることがある[10]。また九工大では社会人向けコースの開設や教育研究での産学連携も考慮する必要がある[11][12]。大学院での継続的教育に加え、異なる背景と目的を持つ複数コースの系統的な開発・管理が必要となっている。このような九工大内での多様性に加え、教育機関ごとの多様性もある。プロセス教育に対する方針の違いに加え、プロセスデータの管理ツールや LMS (学習管理システム Learning Management System)、遠隔講義プラットフォームなどに多様性がある。このような背景から、PSP コースに後述するソフトウェアプロダクトライン (Software Product Line 以降 SPL) の技法を適用することを検討した。

本稿の残りの部分は以下のように構成される。第 2 節で、適用したプロダクトラインのアプローチの概要を述べる。第 3 節で PSP のプロセスの再利用における課題を論じる。第 4 節で分析内容について論じ、第 6 節でまとめと今後の課題について述べる。

2. プロダクトラインアプローチ

SPL は、共通の管理された特徴をもち、特定のマーケットやミッションのために、共通の再利用資産に基づいて作られる、ソフトウェア集約的なシステムの集合である[13]。そのような SPL の技法はソフトウェアそのものだけでなく、ソフトウェアプロセスのテーラリングへ適用することも提唱されている[14][15]。本稿でも SPL の手法を、プロセスに適用することを検討するが、特に PSP をコアとしたプロセスの教育・研究に適用することを検討する。

2.1. ソフトウェアプロダクトライン

SPL では共通の特徴が管理されており、再利用の状況把握や検討などの検討を的確に行える。共通のマーケットやミッションのために、ハイエンド、標準品、ローエンドといったラインナップを考え、再利用資産開発を含めた全体の対投資効果などを検討しやすい。共通の再利用資産の活用を前提とした体系的なアーキテクチャ設計が SPL 全体でなされるため、アーキテクチャ上の不整合を回避しつつ包括的な再利用資産を整備できる。

図 1 の参照モデル[16]のように SPL 開発のエンジニアリングには主に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がある。ドメインエンジニアリングでは SPL の共通性や可変性を管理し、コア資産を開発する活動を行う。コア資産は、SPL がベースとする再利用資

産で、ドメインアセットとも呼ぶ。その開発活動にはプロダクトラインスコーピング、コア資産の要求分析、設計、実現、検証と妥当性確認などがある。アプリケーションエンジニアリングではコア資産を再利用しながら SPL を構成する個々のアプリケーションを開発する。個々のアプリケーション開発のための、要求分析、設計、実現、検証と妥当性確認といった活動が含まれる。

そのようなエンジニアリングで開発した再利用資産はアセットベースに格納する。アプリケーションエンジニアリングで開発されたものでも、SPL 開発で再利用できるものはアプリケーションアセットとして格納される。直接的なエンジニアリング活動以外にも、SPL 開発への移行や SPL 開発の計画、ステークホルダとの関係などに関わる活動などは組織管理に分類される。構成管理や資産管理などに関する特定のライフサイクルのフェーズに依存しない横断的な活動は技術管理とされる。

2.2. プロセスのプロダクトライン

ソフトウェアプロセスライン

ソフトウェアプロセスもまたソフトウェアであるという考え方に基いて、SPL の手法をソフトウェアプロセスに応用したソフトウェアプロセスラインと呼ばれるものも提唱されている[14]。SPL の目的はソフトウェアのコア資産を効果的に活用しビジネス目標に則った高生産性と高品質を系統的に実現することである。そのような SPL のアプローチの有効性は、ソフトウェアプロセスが対象でも同様との考えの下、コア資産にもとづいてソフトウェアプロセスをテーラリングする。

ソフトウェアプロセスのテーラリングの典型的な実施は以下のようなものである。

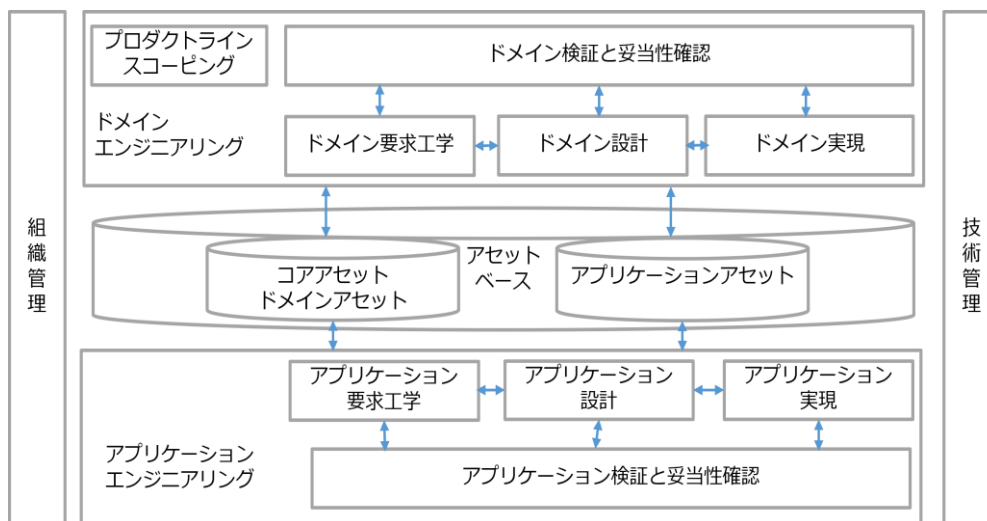


図 1 SPL のエンジニアリングと管理の参照モデル[16]

- プロジェクトの立ち上げが決まる
 - プロジェクトの特性を分析する
 - その特性に対応させて、標準プロセスの構成要素を選択調整し、または新たに作り統合する
 - 統合したプロセスを検証し妥当性を確認する
- ソフトウェアプロセスラインは上記のようなテラリングが以下のような SPL のエンジニアリング活動に類似している点に着目している。

- 製品計画が提示される
- 要求を分析する
- 要求に対応させてコア資産の要素を選択・調整し、または新たに作り統合する
- 統合した製品を検証し、妥当性を確認する

3. PSP コース再利用の課題とスコーピング

PSP コースをソフトウェアプロセスラインにもとづいて再利用する際の課題とスコーピングについて論じる。

3.1. 再利用における課題

アーキテクチャ上の想定

ソフトウェアの再利用資産はそれが使われるソフトウェアアーキテクチャ上の想定を持っている。再利用資産を使う側のアーキテクチャがそのような想定と整合しないと再利用は難しい[17]。再利用資産の実際の利用時に、開発する側の想定から外れた使い方をしようとすると、手間やパフォーマンスの問題が生じたり、そもそも利用できない可能性が生じたりする。プロセスに関するアーキテクチャ上の想定は、フェーズ構成、プロセスデータの収集、作業成果物の構造や表現、利用するツールに関する想定などが考えられる。

アーキテクチャに関して SPL ではその製品が共有するソフトウェアアーキテクチャであるプロダクトラインアーキテクチャの検討を行う。ソフトウェアアーキテクチャはソフトウェアの品質特性に大きな影響を持つため、製品の品質特性を十分に考慮して設計する必要がある。SPL で開発される再利用資産がアドホックに設計されると開発が円滑に進まなくなる。インタフェースを規定することで、個々の製品開発時に再利用資産を組み合わせる作業が、体系だつて行える。

ここでは既存の PSP を標準プロセスとし、その標準プロセスのアーキテクチャを個別プロセスでも想定アーキテクチャとする。PSP のフェーズや主要プロセス要素などのアーキテクチャを図示すると図 2 のような構成となっている。品質の課題に取り組む PSP2 から加わる設計レビューとコードレビューを除き、PSP トレーニングの初期レベルからフェーズや主要プロセス要素の構成は共通である。

また、PSP ではプロセスデータに基づく改善が重要なポイントとなっている。見積もりや品質の改善のために関連するデータを収集し、次回以降の開発や改善に活用する。このような活動を支援するデータ処理のアーキテクチャは図 3 のようなものとなっている。

利用効果と汎用性のトレードオフ

再利用資産は、特定領域に特化するほどその利用効果が高くなるが、利用できるユーザは限定されてしまう。逆に汎用化させるとユーザは多くなるが、利用効果が低くなる。例えば PSP をベースにした標準的なプロセスと形式仕様記述を用いるプロセスを考える[18]。前者は自然言語や準形式的なモデル記述のような汎用性の高い表現法の仕様テンプレート、後者は欠陥の混入防止や除去に高い効果が期待できる形式的仕様記述言語のよう

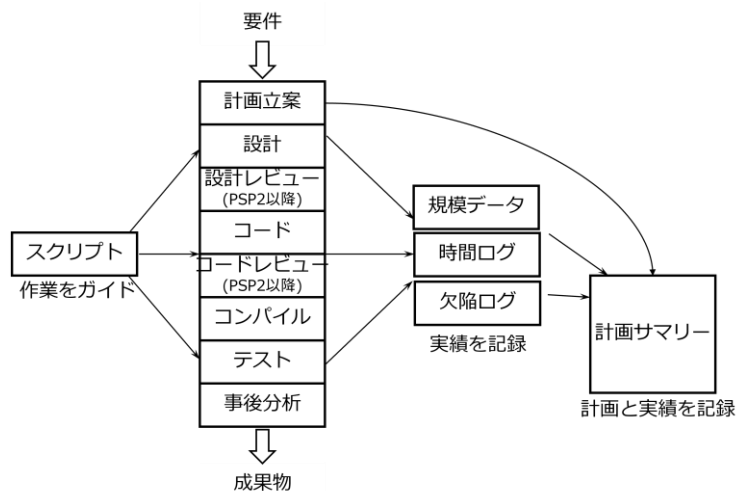


図 2 PSP 基本アーキテクチャ

な特定技術に特化した記法の仕様記述テンプレートを持ち得る。後者は数理的検証といった技術が効果的な対象に対して品質の向上が期待できるが利用者が限定され再利用資産の開発コストが見合わない可能性がある。一方、前者のような汎用的な再利用プロセス資産は多くのユーザが利用できるが、特別な技術による品質向上といったことは期待できない。

上記のようなトレードオフもふまえ再利用資産が実際に活用されるか考慮した上で、どのような分野のどのようなユーザを対象とするかというスコープの決定が重要となる。例えば九工大の取り組みでは、従来の大学院でのPSPトレーニングコースのユーザに加え、社会人コースの学生、共同研究先などが対象ユーザに含まれるとする。

ビジネス的観点

再利用の枠組みが期待通りに機能するかどうかは、再利用の技術的側面だけでなく、再利用の管理的側面が適切に実現され機能し維持されるかにも大きく依存している。再利用資産の開発や管理にはコストがかかり、再利用資産を作る側も、使う側もそれでもメリットがあるかが重要である。再利用資産を開発してもそのコストが見合わなければビジネス的には再利用は推進できず、再利用資産を利用するメリットがなければ再利用は進まない。再利用資産を開発する人、管理・提供する人、再利用資産を利用する人などといった様々なステークホルダと、それらの間でやりとりされる再利用資産、対価などから構成されるエコシステムの観点も必要である。

PSP の場合、産業界の組織プロセスの観点では、組織構成員個人のレベルのプロセスのベースとなる可能性も持っている。産業界での利用の場合、ソフトウェア開発プロセスに対するビジネス観点での分析が重要となる。ま

た九州工業大学での取り組みのような場合でも、従来の教育の観点に加え、社会人教育や産学連携を行う大学経営というビジネス観点での分析が重要となる[11][12]。

3.2. スコーピング

前述のような課題を念頭に PSP のプロダクトライン開発のスコーピングを行う必要がある。SPL のスコーピングでは、SPL に含まれるプロダクトを決定し、それらの間の共通性や可変性を分析し、プロダクトの計画を検討する。例えばどのソフトウェア製品が SPL に含まれ、どの製品が含まれないのかという、SPL の製品群の特徴づけを行う。今回の PSP の取り組みだとソフトウェア製品でなくプロセスについてこのような検討を行う。

スコーピングは技術的視点からだけではなく、前述のビジネス観点でコア資産を開発するための対投資効果の検討なども行う。SPL 開発では個別製品の個別最適ではなく、全体最適の視点が重要視されるため、個別プロダクトに対する最適解が必ずしも SPL 全体としての最適解とは限らない。例えば共通の再利用資産を使わず、個別に製品開発した方が、その製品のロードマップに適した機能や性能を実現しやすいかもしれない。また、ある製品だけ見ると個別開発の方が低コストに見えても、製品群全体の視点で見ると、共通の再利用資産の利用によって全体的な開発コストを下げられる可能性もある。

個人レベルのプロセスについて、各組織の慣習的な標準プロセスにもとづいて個別にアレンジする方が導入しやすく短期的には良い可能性がある。しかしながら、俯瞰的で長期的な視点での改善効果まで含めて考えると慣習的なプロセスよりよりも PSP をベースにした再利用資産を利用する効果が期待できる可能性がある。九工大での取り組みの場合、スコープの全体像を従来の大学院教

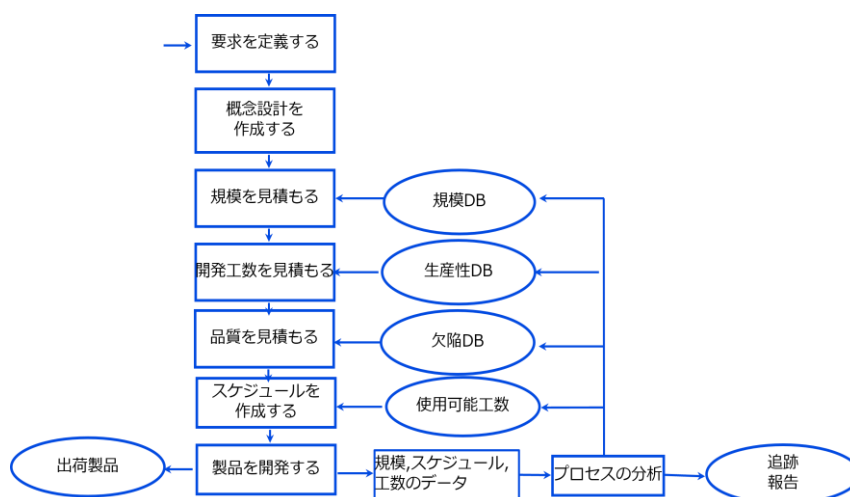


図3 PSPのデータ処理アーキテクチャ

育から、社会人教育や産学連携の共同研究まで含むものに変える必要がある。

上述のような背景の下、各プロセスのロードマップを検討する。SPL でのプロダクトスコーピングでは、個々のプロダクトのロードマップを決定する。例えば、SPL における各プロダクトが、対象とするマーケットのどのようなセグメントをカバーするのか、そしてそのために各プロダクトがどのような特徴を持つのか、などを決める。SPL のドメインスコーピングは、開発時にどのような機能領域が重要になるかを識別しプロダクトと対応づける活動である。SPL のアセットスコーピングは、どのようなコア資産を開発することが効果的かを決定する活動である。このようなことを、PSP をベースとしたプロセスに対して行う必要がある。

4. PSP 分析例

4.1. ドメインの概要

PSP は、CMMI のレベル 5 を実現するチームレベルプロセスの TSP を実践できる、個人レベルのプロセスという位置づけで作られており、それらのスキルセットの関係は図 4 に示すようなものとなっている。例えば PSP であれば、チームメンバになるためにも必要なスキルとしてプロセスの規律、パフォーマンス測定、見積もりと計測、品質管理のスキルが含まれている。

また PSP コースでは段階的に高度化するプロセスを用いる(図 5 参照)。前述のスキルのうち、まずは規律と計測に関する要素が PSP のバージョン PSP0 のプロセス系列に含まれている。その後、見積もりと計画立案に関する要素がバージョン PSP1 の系列に、品質管理と計測に関する要素がバージョン PSP2 の系列に含まれている。このような特徴と構成を持つ PSP コースに対して、SPL の技法を用いる。

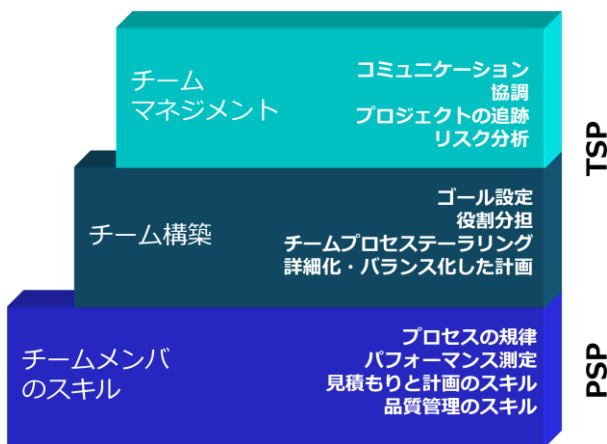


図 4 TSP と PSP のスキルセット

4.2. 共通性と可変性

SPL 開発の大きな特徴として、共通性と可変性に着目して開発を進める点がある。共通性とは SPL のソフトウェア製品すべてに共通する機能的・非機能的な特徴で、可変性とは SPL 中の製品ごとに変わりうる機能的・非機能的な特徴である。一般的に再利用では、複数のソフトウェアの共通部分の再利用を考える。SPL 開発では、共通性がある部分だけでなく可変性がある部分の再利用も考慮し、より再利用の効果を高めることを目標とする。

SPL で共通性と可変性の記述に用いられているものの一つとしてフィーチャモデル[19]がある。フィーチャモデルは、ドメイン分析のために提案されたモデルで SPL 開発において広く使われている。フィーチャとは対象ソフトウェアあるいはソフトウェア群の重要な、もしくは明確にユーザから観測できる側面、品質、特徴である。フィーチャモデルでは、SPL の持つフィーチャを階層的に記述する。必須フィーチャはその上位階層のフィーチャが選ばれれば必ず選ばれるフィーチャ、選択フィーチャは上位階層が選ばれてもその選択は任意のものである。要求元のフィーチャを選択した場合に要求先のフィーチャが必要になることを示す要求関係や、両端のフィーチャを同時に選ぶことができないことを示す排他関係を示す有方向のリンクも記述できる。フィーチャグループは多重度の制約の範囲で任意のフィーチャを選べることを示す。

本稿の取り組みはソフトウェアではなくプロセスが対象であるが、フィーチャモデルの考えベースにプロセスの共通性と可変性の分析を行った。上述の図 5 で示したような、段階的に高度化するプロセスで用いられるプロセス要素について表 1 のように分析した。表中の要素は資産候補と考える。下線がないものはコア資産、下線がつい

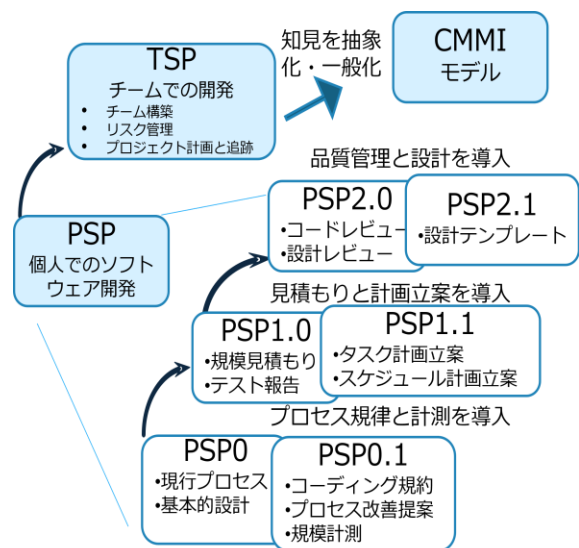


図 5 PSP の段階的の高度化

たものはアプリケーション資産の候補である。PSP は見積もりや品質管理のプロセスのスキルの習得を目標としているが、見積もり手法や設計手法は特定のものである必要はない。例えば PSP では具体的な見積もり手法としては PROBE(PROxy-Based Estimation)法を採用しているが、図 2 や図 3 のアーキテクチャと整合すれば他手法でも利用可能である。そのため PROBE という特定の技法に関するプロセス要素はアプリケーション資産の候補とした。品質に関連するプロセス要素である設計テンプレートもシャンプーの4象限にもとづくものが用意されているけれど上述の PROBE 法と同様に、整合すれば他の設計テンプレートも利用可能と考える。SPL の手法によりこのような PSP プロセス中の共通性と可変性が明確化された。

5. プロセス系列の導出と検証

SPL では製品導出で共通の再利用資産から特定のソフトウェア製品を構築する。可変性の管理、コア資産の構築、それら間の対応関係の管理を体系立てて行うことによって、効果的に製品を構築できる。フィーチャモデルにより SPL の可変性を表現し、アーキテクチャ上で可変性に対応した可変点が定義され、可変点に対応するコンポーネントを定める。個別ソフトウェア製品を導出する際は、その製品が持つフィーチャを明確にし、フィーチャの共通性と可変性を識別することで、必要な再利用資産が特定でき、求めるソフトウェア製品を開発できる。

組織標準プロセスを対象にしたテーラリングへの SPL のエンジニアリング適用の事例は既に存在する。事例

[15]ではプロジェクト特性をフィーチャモデル化し、プロセス定義のデータ構造を定義した上で記述し、プロセス定義情報へフィーチャを割り当て、テーラリングパターンを定義している。

PSP のようなプロセスおよびそのトレーニングコースは、ソフトウェアと異なり実行主体が計算機でなく人や組織である点も重要と考える。PSP プロセスの場合でも、要素間のインタフェースやデータ構造の検証はソフトウェア同様機械的な確認が容易な一方、その実行に関しては、計算機が実行するソフトウェアと異なり、機械的な処理に大きく依存はできない。学生を対象とした PSP トレーニングコースの場合、職業的にソフトウェアを開発する組織で標準プロセスが既に定着している、といったプロセス実行者に対する前提も確実ではない[10]。学生、教員、部局、大学などの実践者やステークホルダは決定的な動作を行うとは限らず、個別の変動に加え、取り巻く社会環境による影響を受けることも多いため変動を前提に分析する。

PSP コースのプロセス系列の開発に人間中心設計で使われることが多いシナリオベースの手法を用いる。シナリオ分析には図式表現を使うものも含め様々なものが考えられるが、ここでは発想を支援する図式表現を使うシナリオ分析として機能共鳴分析手法 (Functional Resonance Analysis Methods, 以降 FRAM) の分析手法[20]を用いる。FRAM での記述は、例えばソフトウェア開発で一般的な統一モデリング言語 (UML: Unified Modeling Language)のように観点ごとに図式表現を使い分けられないため、多面的な分析を同じ表記をベースに実施できると考える。例えば文献[21]の分析では、シナリオ進行の図

表 1 プロセス要素の分析

プロセス	プラクティス	利用技術	スクリプト	フォーム	標準	テンプレート	課題
PSP 0	規律と測定		計画, 開発, 事後分析	Project Plan Summary, 時間記録, 欠陥記録	欠陥型		①
PSP 0.1	同上			プロセス改善提案	コーディング, 規模計数		②
PSP 1	同上+見積りと計画	PROBE	PROBE			PROBE規模見積, テストレポート	③
PSP 1.1	同上	PROBE				タスク計画, スケジュール計画	④
							中間レポート
PSP 2	同上+品質管理	レビュー			コードレビューチェックリスト		⑤
PSP 2.1	同上	設計(4象限: 操作, 機能, 状態, 論理)	設計レビュー		設計レビューチェックリスト	設計テンプレート(4象限)	⑥⑦⑧
							最終レポート

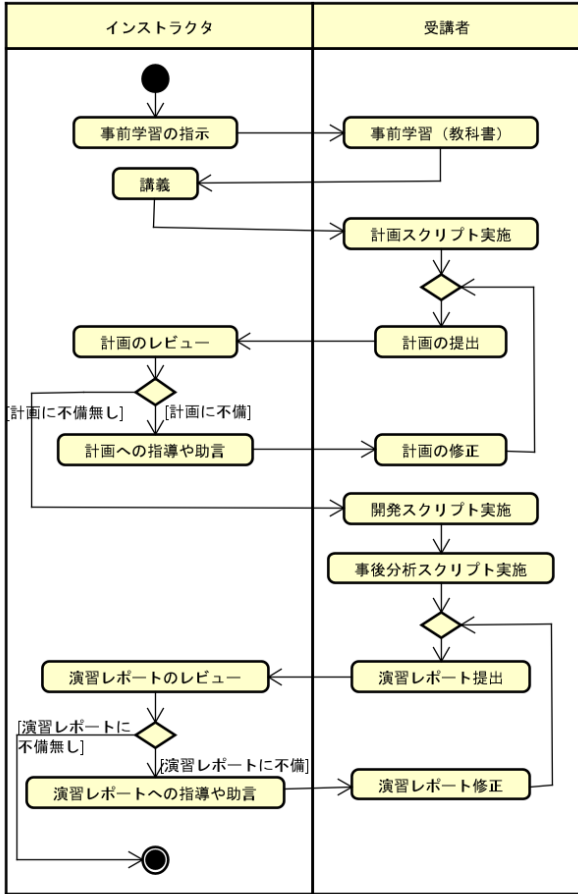


図 6 インストラクタと受講者のやり取りの記述例

式表現として図 6 のような UML のアクティビティ図を用いている。このような図はシナリオに沿った処理の流れの表現には有用と考える。これに対し FRAM では機能・アクティビティの記述の際に、図 7 に示すような入力 I と出力 O の間の接続による処理の流れの記述に加え、前提条件 P、時間 T、制御 C、資源 R、の合わせて 6 つの aspek トを含めた記述が可能である。そのため、機能・アクティビティを含むシナリオの分析の際に、そのような aspek トの観点を含めて分析が行えるため気づきが多いと考える。その際 FRAM のシナリオ記述とシナリオ実行者のペルソナを併用した文献[22]のような手法を用いる。

例えば PSP のプロセス要素にスクリプトがあり、プロセスの実行者はこのスクリプトに沿って作業を行う。そのスクリプトに沿った作業シナリオを FRAM で記述すると、作業アクティビティごとに上述の aspek トについて意識することになる。実例として、計画スクリプト中に” 要求記述を制作もしくは獲得すること” とあるが実施者によってはこの要求記述でつまづくことがある。学生の場合、大学ではプログラミング演習は経験しているが、要求記述の学習経験が無い場合も多い。このように産業界で PSP コースを実施する場合には問題とならない箇所が大学では問

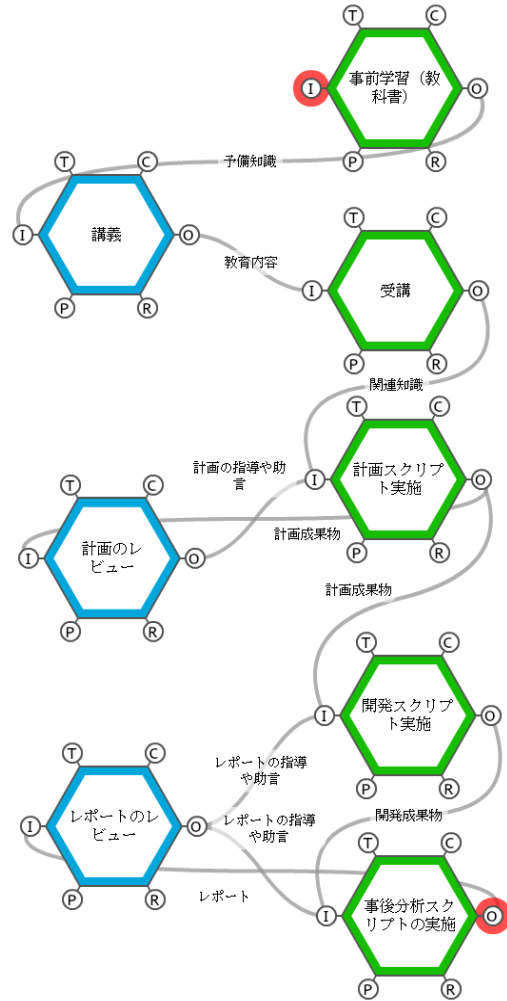


図 7 FRAM でのベース記述の

題となってしまうことがある。FRAM で記述されたシナリオでは前述のようにアスペクトも記述できるため、作業の前提条件といったアスペクトと実施者のペルソナの確認によりこのような問題への気づきと対策が促進される。このような問題への対策の成果物は、コース実施者の属性に違いに対応するため必要に応じて資産化される。

6. おわりに

大学でソフトウェア開発プロセスのコースを実施、発展させる観点で、PSP トレーニングコースへプロダクトライン技法を適用することについて論じた。ソフトウェア開発を取り巻く環境、大学に期待される社会的役割の変化などにより、求められるソフトウェアプロセスやその教育の多様性も増している。従来の PSP トレーニングコース実施の継続であっても、比較的軽微な修正を中心とした保守を行う必要がある。さらに、社会人向け教育や産学連携共同研究などに対応するための比較的大がかりなコース開

発もあり得る。そのような状況に系統的に対応するために、ソフトウェアプロダクトライン技法を PSP トレーニングコースに適用することを検討した。

本稿では、そのアプローチと課題について論じた。これまで組織内プロセスのテーラリングにプロダクトラインの技法を適用する、といったプロセスを対象とした取り組みはなされており、その知見の多くは PSP トレーニングコースの場合にも当てはまる。SPL の手法により PSP トレーニングコース中の共通性と可変性が明確化された。

また、本稿では、想定する受講者が異なるプロセストレーニングコースのプロダクトラインの分析ため、図式的記法を持つ FRAM を用いペルソナを併用したシナリオベースの分析法を用いることを論じた。提案する方法で、学生といった、特定の利用者層を対象とした PSP トレーニングコースの実施で典型的な問題を明らかにし、その対策に関する成果物を資産化できると考える。

今後の課題としては、社会人向けの PSP コースや、産学連携の共同研究先での PSP コースの展開などにも提案手法を適用、評価していく予定である。

参考文献

- [1] Humphrey, W. S.: A Discipline for Software Engineering, Addison-Wesley (1995), (邦訳:松本 正雄 監訳, ソフトウェア品質経営研究会訳:『パーソナルソフトウェアプロセス技法』, 共立出版, 1999 年).
- [2] Humphrey, W. S.: A Self-Improvement Process for Software Engineers, Addison-Wesley (2005), (秋山 義博監訳, JASPIC TSP 研究会訳:PSP ガイドブック ソフトウェアエンジニア自己改善, 翔泳社, 2007 年).
- [3] 秋山義博, 片峯恵一, 梅田政信, 橋本正明, 乃万 司: 九州工業大学におけるパーソナルソフトウェア プロセス教育—ソフトウェア品質向上のためのスキル 修得—, SEC Journal, Vol. 6, No. 3, pp. 118-125 (2010).
- [4] Katamine, K., Umeda, M., Hashimoto, M. and Akiyama, Y.: Changing Software Management Culture from Academic, in TSP Symposium 2011 Proceedings, pp. 12–18 (2011).
- [5] Katamine, K., Umeda, M., Hashimoto, M. and Akiyama, Y.: A STRATEGY IN EFFECTIVE TEACHING OF SOFTWARE ENGINEERING PROCESS FOR GRADUATE STUDENTS, in The Proceedings of IADIS International Conference Information Systems 2012, pp. 259–266 (2012).
- [6] 梅田政信, 片峯恵一: PSP/TSP による実践的な ICT 人材育成の取り組み, 情報処理学会誌, Vol. 53, No. 10, pp. 1084-1087 (2012).
- [7] Umeda, M., Katamine, K., Hashimoto, M. and Akiyama, Y.: Improving Introductory TSP for Creating High Performance Student Teams, in TSP Symposium 2013 (2013).
- [8] Creative Commons : CC BY 4.0 リーガル・コード, <https://creativecommons.org/licenses/by/4.0/legalcode.ja> (2019).
- [9] Software Engineering Institute : Team Software Process (TSP) and Personal Software Process (PSP) Materials, <https://www.sei.cmu.edu/go/tsp> (2019).
- [10] 日下部茂, 梅田政信, 片峯恵一, 荒木俊輔:安全工学由来のモデリング STAMP を用いた大学院 PSP 教育改善活動の促進, ソフトウェアプロセス改善カンファレンス 2023 (SPI Japan 2023), https://www.jaspic.org/events/sj/spi_japan_2023/
- [11] 片峯恵一, 安永卓生:社会人向け PSP/TSP コースについて, PSP/TSP 成果報告会, 2024
- [12] 加藤雅和, 秋丸義輝:PSP にたどり着いた本当の理由, PSP/TSP 成果報告会特別講演, 2024
- [13] Clements, P. & Northrop, L.: Software Product Lines – Practice and Patterbsm Addison-Wesley, 2001
- [14] 林好一: ソフトウェアプロダクトラインエンジニアリングをプロセステーラリングに応用する, 第 25 回ソフトウェア品質シンポジウム, 2006
- [15] 池永直樹: プロセステーラリングへのプロダクトライン適用の実践, ソフトウェアプロセス改善カンファレンス 2023 , https://www.jaspic.org/events/sj/spi_japan_2023/
- [16] ISO/IEC : ISO/IEC26550, Reference Model for Software Product Line Engineering and Management, 2010
- [17] Garlan, D., Allen, R.& Ockerbloom, J.: Architectural mismatch: why reuse is so hard, IEEE Software, vol. 12, no. 6, pp. 17-26, Nov. 1995
- [18] 日下部茂, 大森洋一, 荒木啓二郎, プロセス改善における形式手法の活用 of の検討, ソフトウェアシンポジウム SS2013 予稿集, <http://sea.jp/ss2013>, 2013
- [19] Kang, K. C., Cohen, S. G., Hess, J. A., Novak, W. E., & Peterson, A. S: Feature-Oriented Domain Analysis (FODA) Feasibility Study, Technical Report, CMU/SEI-90-TR-21 ESD-90-TR-222, 1990
- [20] Hollnagel, E. (小松原明哲監訳). 社会技術システムの安全分析-FRAM ガイドブック, 海文堂出版, 2013
- [21] 梅田政信, 片峯恵一, 荒木俊輔, 橋本正明, 日下部 茂: ソフトウェアプロセスの自己改善は自学自習でも可能なのか?, ソフトウェアシンポジウム 2019 予稿集, <http://sea.jp/ss2019>, 2019
- [22] 日下部茂, 有田大作: 遠隔授業改善でのターゲットのペルソナ化とステレオタイプ修正, 人間中心設計推進機構・機構誌, Vol.20 No.1, pp10-17 (2024)