

TSP の俊敏性向上に関する取り組みと評価

A Discussion on Improving Agility of TSP

九州工業大学

Kyushu Institute of Technology

○片峯 恵一 梅田 政信 荒木 俊輔 橋本 正明

○Keiichi Katamine Masanobu Umeda Shunsuke Araki Masaaki Hashimoto

Abstract Team Software Process (TSP) provides a high quality software development method which involves team building and team management. TSP uses a cyclic development process based on personal software process, and incorporates many best practices to realize high quality software within a schedule and a budget. However, if a software size is small enough and/or requirements are changed frequently, an agile software development method may be superior than TSP. This paper proposes several techniques to improve agility of TSP, and discusses the result of which the proposed techniques were applied to a TSP team of graduate school students.

1. はじめに

ソフトウェア開発では、大規模かつ短納期という非常に厳しい状況が続いており、さらに、要求変更への柔軟な対応と品質の確保も望まれている。著者らは、このような状況に対応できるソフトウェア開発手法として Watts S. Humphrey 博士の提唱するパーソナルソフトウェアプロセス (PSP) ^[1,2] とチームソフトウェアプロセス (TSP) ^[3,4] に着目している。

PSP は、高品質なソフトウェアを開発するための自己改善プロセスであり、TSP は、チームの構築とマネジメントのための枠組みである。また、TSP は、ソフトウェアプロセスに基づく循環的な開発プロセスを用いており、高品質なソフトウェアを計画通りに実現するための多くのベストプラクティスを組み込んでいる。Capers Jones によると PSP/TSP は、中規模以上のソフトウェア開発において最も優れた成果をあげているが、小規模開発においてはアジャイル手法が優れているとされている^[5]。TSP を提供している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 (以下、SEI と呼ぶ) では、アジャイルソフトウェア開発宣言に挙げられている 4 つの価値や、12 の原則の多くは TSP と類似した考えであると述べている。しかし、具体的な対応策や手段は示されていない。

九州工業大学では、SEI と連携して、PSP や TSP に関する科目を大学院向けに導入している^[6,7]。しかし、TSP に関しては、プロジェクトの立ち上げや計画立案に多くの時間がかかり、短い期間に多くのサイクルを実施することは困難な状況にある。そこで、TSP の俊敏性を向上させるために、開発戦略やアーキテクチャ設計等に関する新たな指針を提案する。本稿では、これらの指針を学生プロジェクトに適用した結果について報告する。また、プロジェクトサイクルを短くした際の品質への影響や、教育的な観点からの効果について議論する。

2. TSP の概要

TSP は、高品質なソフトウェアをチームで開発する際の枠組みを示している。TSP では、PSP を修得したメンバーから構成されることが前提となっているため、本節では、PSP と TSP の概要について述べ、TSP プロジェクトを実施する際の課題について述べる。

九州工業大学大学院情報工学研究院

Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology

〒820-8502 福岡県飯塚市川津 680-4 Tel: 0948-29-7649 e-mail:katamine@ci.kyutech.ac.jp
680-4, Kawazu, Iizuka-shi, Fukuoka 820-8502, Japan

【キーワード：】 TSP, PSP, ソフトウェアプロセス, プロジェクトマネジメント

2.1 PSP/TSP の概要

PSP/TSP は、SEI において Watts S. Humphrey 博士を中心に開発された。ソフトウェアの品質は、それを構成する最低品質の部品によって決まり、各部品の品質はそれを開発した個人とその時に用いたプロセスにより決まる、という考えに基づき、ソフトウェア品質の改善方法として、個人のスキル改善を対象とした PSP とチームの構築とマネジメントを対象とした TSP を開発している。ソフトウェア開発は、ソフトウェア技術者本人にしか分からない知識集約型の労働であることから、PSP の教育では、特に個人のスキル改善や規律の遵守に重点が置かれている。

図 1 は、PSP におけるプロセスの発展、および PSP と TSP との関係を示したものである^[2]。PSP0、PSP0.1 においては、欠陥記録、時間記録、規模測定、改善提案およびこれらを実践するための規律に従う。PSP1、PSP1.1 においては、要求の実現に必要な部品の同定とこれに基づく規模および時間の見積り、開発計画と進捗管理を行う。PSP2、PSP2.1 においては、品質の見積り、品質計画、設計とコードのレビュー、ならびに設計テンプレートと設計検証法を用いる。PSP のスキルは積み上げ式となっており、PSP2.1 までの知識とスキルを身につけたソフトウェア技術者が集まり TSP チームを構築する。

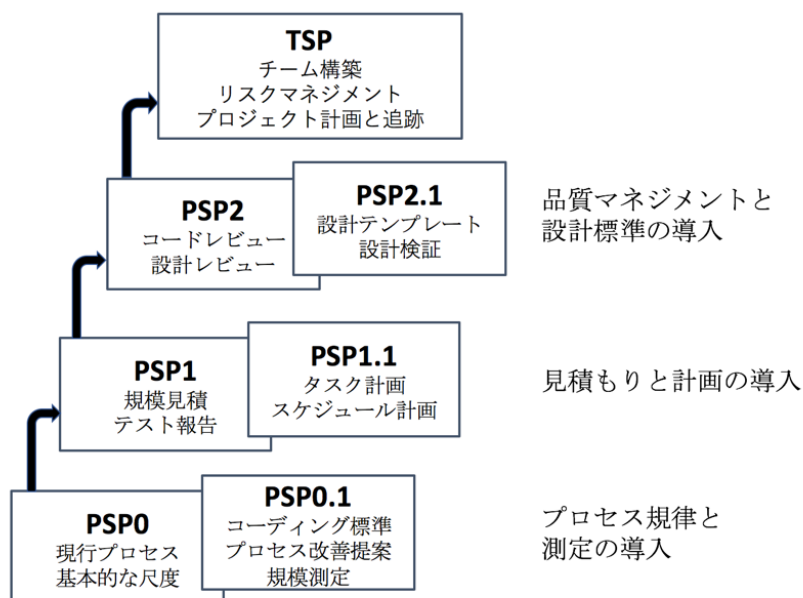


図 1. PSP プロセスの発展

TSP は、PSP を修得した技術者から構成される自律的なチーム作りと、そのようなチームによる高品質なソフトウェア開発を実現するための枠組みである。ここで自律的なチームとは、顧客要求を満たすチームゴールの設定、それを達成できる開発戦略と計画の立案、その遂行の計測と追跡をチーム自ら実施できることを言う。TSP には、プロジェクトの重要なプロセスの一つとして立ち上げ (Launch) がある。立ち上げは、チームメンバー間でゴールや責任を共有して結束力を高めることにより、自律チーム作りを支援するプロセスである。図 2 に立ち上げプロセスの概要を示す^[3]。この立ち上げにおいて、チームゴールを設定し、実施可能な開発計画や品質計画を作成できるのは、各メンバーの生産性等に関する定量的なデータが PSP を通じて揃っているからに他ならない。

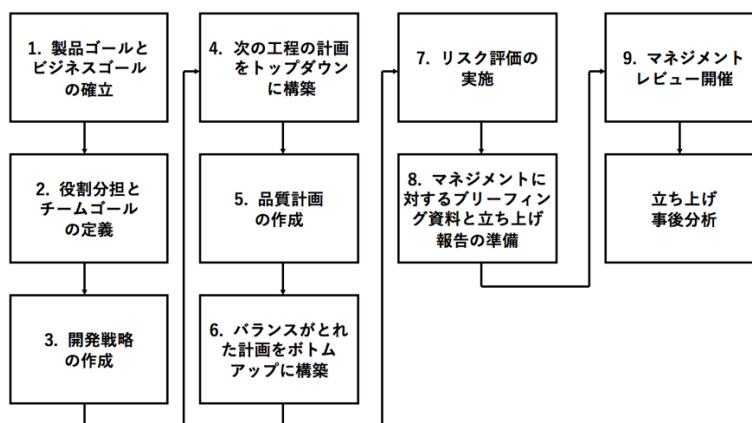


図 2. 立ち上げプロセスの概要

2.2 TSP 実施における課題

九州工業大学では、大学院生に対して、2007 年度から PSP を、2010 年度から TSP を教育用に簡略化した TSPi(Introductory TSP)^[4]による演習科目（以下、TSP コースと呼ぶ）を導入している。2015 年度までに 11 チームが TSP コースを実施し、多くのチームにおいて、第 2 サイクルでシステムテスト欠陥 0 を達成した。このことから、受講生には、品質を重視するという PSP/TSP の考え方が身につけていると考えている。しかし、TSPi プロセスに沿って作業を実施することが初めてであることや、エンジニアリング能力の不足から、多くのチームが第 1 サイクルで遅延を起しており、第 2 サイクルに十分な時間の確保が難しい状況が続いている。そこで、特に第 1 サイクルにおける遅延対策が必要である。なお、エンジニアリング能力に関しては、他の講義科目や演習科目と連携することにより対策を行っているが、根本的な解決には至っていない。

また、企業におけるソフトウェア開発プロジェクトへの要求の一つに、要求変更への対応がある。TSP では、要求変更に応じて、実施プロセスを変更することも想定されている。その際には、そのプロセスを終結し、事後分析を行った上で、再立ち上げ工程を実施することとなる。しかし、事後分析や再立ち上げには時間がかかるという問題がある。そこで、特に要求分析工程において、効率の良い実施方法が望まれている。九州工業大学では、例年、PSP/TSPi の成果、およびその後実施している TSPi を使用した PBL の成果を、学外向けの報告会として実施しているが、企業の方から、要求変更を前提としたプロジェクトの実施要望が挙げられている。このため、要求変更への対応を考慮した TSPi の実施方法を検討する必要がある。

3. 俊敏性向上のための取り組み

本研究では、TSP の俊敏性を向上させるために、以下の 2 種類の取り組みを行った。これらの取り組みは、基本的には、九州工業大学における教育の一部として実施することを想定している。

(1) 開発サイクルの増加

従来の演習では、2 サイクルの循環プロセスであった。これをサイクルごとの開発規模を小さくし、3 サイクル以上で計画することとした。これにより、第 1 サイクルで TSP プロセスを体験し、その実績に基づき、複数のサイクルを体験することが可能となる。また、サイクルを短くすることにより、学生の改善活動を促進させることもできる。

(2) 開発戦略の追加

各サイクルで開発する製品は、顧客が要求の妥当性を確認できる機能を含むことを条件の一つに加えた。この条件を満たす各サイクルでの成果物は、顧客からのフィードバックに柔軟に対応できるようになる。また、製品の設計は、機能の追加や変更があることを前提とし、容易に対応可能となるようなアーキテクチャとするも条件に加えた。なお、実際の演習では、第 3 サイクル開始時に変更要求をインストラクタが提示し、その影響の分析や対応を検討してもらう。

4. 実験と結果

本稿で提案した俊敏性向上のための2つの方法を大学院生のTSPコースにおいて試行した。本節では、実験の概要と結果について述べる。

4.1 実験の概要

実験は以下のような条件のもとで実施した。

- (1) 開発システムは、TSPiガイドブック^[4]に記載してある変更カウンタとする。
TSPiガイドブックの付録Aには、変更カウンタの機能要求記述がサンプルとして記載されている。このプログラムは、バージョン管理システムの一つである。TSPコースでは、過去にもこの題材を取り扱っており、これまでの実施結果と比較するために同じシステムを開発対象とした。
- (2) 開発期間は、後期の第3クォータおよび第4クォータとする。
本プロジェクトは、9月末から2月上旬の約5ヶ月弱で実施する。ただし、受講生は、それぞれ大学院の他の科目の受講状況が異なるため、チーム内での分担内容や作業量の調整が必要となる。例年は、一人当たり週5~10時間程度の作業を行っている。
- (3) 開発プロセスには、TSPiを使用し、実施には、SEIの開発したSEIツールを使用する。
- (4) 2016年度の実験生は6名であり、1チームあたり3名として、2チームで実施する。
TSPiでは、5名前後のチームを想定して、チームメンバーの役割が定義されている。今回は、3名のチームのため、チームリーダー以外が複数の役割を兼任した。

4.2 実施結果

2チームの実験結果を開発、品質、改善提案の3つの観点から述べる。

(1) 開発

2つのチームは、それぞれ3サイクルでの開発を計画し、3サイクルおよび4サイクルを実施した。4サイクル実施したチームは、第3サイクルの途中で計画に大きな変更が発生したため要求分析プロセスの実施中に第3サイクルを完了させ、計画変更を含む新たなサイクルとして第4サイクルを開始した。このため、開発を完了したのは、両チームとも3サイクルとなる。各サイクルは、第1サイクルで8週間と12週間かかり、その後は3週間から6週間かかっていた。ただし、第1サイクルは、TSPiのプロセスに沿って実施するため、遅延がなくても8週間はかかる。なお、メンバーごとの開発時間や開発規模、および全体の開発期間は、例年とほとんど変わらなかった。また、第3サイクルに入る際に要求の変更を行った結果、この変更要求に対応するために、そのサイクルでの時間配分が変化したチームもあった。図3に時間配分の推移グラフを示す。第1サイクルでは、TSPiプロセスに慣れておらず、ミーティングに多大な時間(25%強)を使用していることがわかる。また、第3サイクルでは、要求変更に対応するために、要求分析プロセスの時間割合が増加していることがわかる。

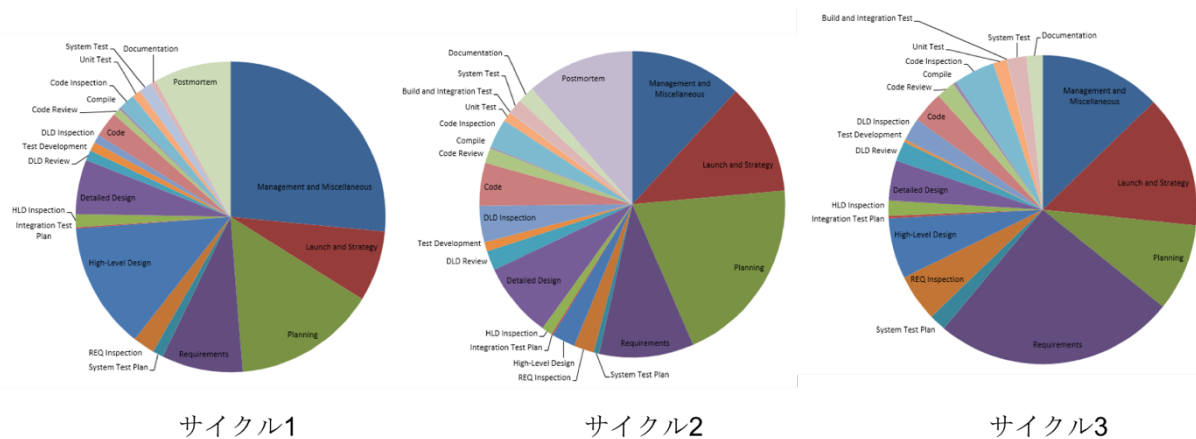


図3. 時間配分の推移

(2) 品質

システムテスト欠陥に関しては、開発が完了した2チーム6サイクルのうち、5サイクルでは0欠陥を達成し、残りの1サイクルでも見つかった欠陥は1つのみであった。また、プロセス品質指標 PQI の改善も見られた。PQI は、設計/コーディング時間、設計レビュー時間、コードレビュー時間、コンパイル欠陥数/KLOC、単体テスト欠陥数/KLOC の5つの指標からなり、それぞれを掛け合わせた PQI 値が 0.4 以上である場合、単体テスト以降でほとんど欠陥が見つからないことが知られている^[2]。図4に、各チームのサイクルごとの結果を赤と青の実線で示す。このように、システムテスト欠陥や PQI で示される5つの指標の向上は、従来の演習と比べても遜色はない。むしろ、サイクル数の増加により、PQI 値はさらに向上している。つまり、システムテスト欠陥だけではなく、レビューやインスペクションの品質も同程度以上であり、今回の試行における品質への悪影響はほとんどなかった。

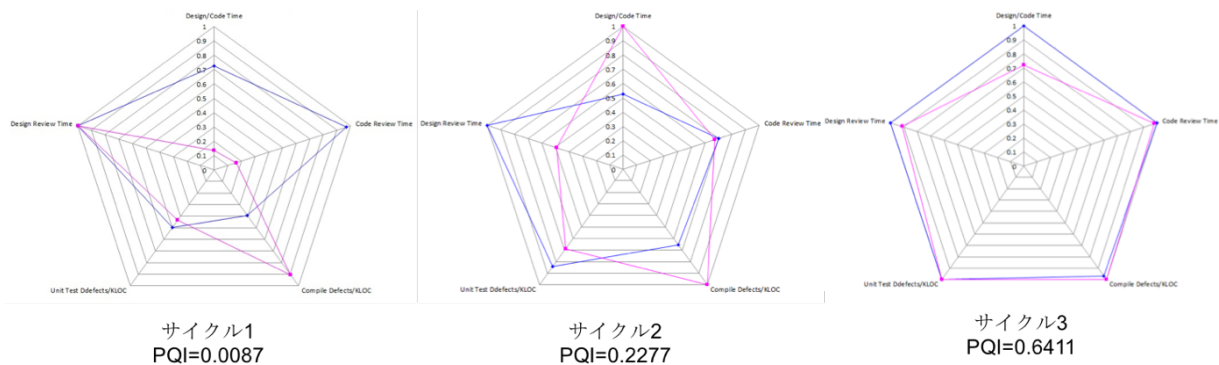


図4. プロセス品質指標 PQI の推移

(3) 改善提案

TSP では、データに基づく改善活動が重視される。つまり、事後分析プロセスにおいてプロジェクトのデータを分析し、次サイクルで活用する。今回の実施では、品質やリスクマネジメント、ミーティングに関する提案が行われた。品質に関しては、レビューやインスペクションの品質を向上するために、実施方法の工夫やチェックリストの更新が行われた。リスクマネジメントに関しては、他講義や学会発表、年末年始の活動等に関連するリスクを十分想定しておらず、遅延が発生したため、リスク項目の洗い出しに関する提案が行われた。ミーティングに関しては、各サイクルを短くしたため、通常 TSP や TSPi で実施する週次ミーティングでは、不都合が生じた。つまり、1週間に複数のフェーズを実施したため、情報共有に問題が生じた。そこで、週次ミーティングに加えて、フェーズ終了ごとにミーティングを開催した。このような問題を学生自身が気づき、対応策を実施することは望ましいことである。これは、サイクル数を増加することによる効果とも考えられる。なお、品質やリスクマネジメントに関する改善は、過去のプロジェクトでも実施していた。

5. 考察

今回の試行に関して、教育的な観点と開発の観点から考察する。

5.1 教育的な観点

大学院科目としての TSP コースは、今回の提案により効果を得た。特に開発サイクルの増加は、自己改善を含む改善活動の機会を増大させ、サイクルごとに振り返りと次サイクルへの提言をまとめるため、学生にとって気づきの機会が増える要因となった。また、開発時間や期間が変わらず、多くのサイクルでシステムテストの欠陥が 0 という良い結果も得られた。しかし、自己レビ

ューの品質やエンジニアリングに関する課題など従来からある問題に対する根本的な解決には至っていない。また、企業における教育においても、PSP や TSP によるマネジメントに関する知識やスキルの修得には効果があるが、実際の開発には対象ドメインに対応するエンジニアリングの教育も重要となる。このため、マネジメントとエンジニアリングの両輪を効率的に教育するための方法を検討していく必要がある。

5.2 開発の観点

今回の試行において、TSP は、小規模プロジェクトにおいても高品質なソフトウェアを開発することができることが分かった。しかし、要求変更に関して、変更内容によっては容易に対処することは難しく、開発の時間配分が崩れることがあることも判明した。今回の提案では、要求変更に対応可能なアーキテクチャを選択することを推奨していたが、どのような変更までを想定しているのかはエンジニアリング能力に依存する部分が多い。このため、もっと具体的な方針や戦略を検討する必要がある。

また、今回の方法により、一部の俊敏性は上がったが、これで十分だとは考えていない。俊敏性の高い開発手法としてアジャイルソフトウェア開発があり、スクラムや XP 等が知られている。これらの開発手法では、短いイテレーション（反復）を繰り返し、その中で様々なベストプラクティスが提案され導入されている。今後は TSP の品質を重視した考え方とプロセスを中心として、俊敏性が向上するベストプラクティスの導入を検討し、教育や開発に直接利用できるソフトウェア開発手法を研究していく予定である。

6. まとめ

本稿では、TSP の俊敏性を向上するために、開発サイクルの増加と開発戦略の追加を行った。この方法を大学院学生の TSP コースに適用し、改善活動に関する効果を得た。しかし、TSP は、マネジメントを中心とした開発手法であり、エンジニアリング能力を保持していることが前提である。このため、マネジメントとエンジニアリングの両方を教育できる方法を検討する必要がある。また、今回導入した手法は、俊敏性を向上させるための最初の小さな一歩である。今後は、TSP の品質を重視する考え方を中心として、さらに俊敏性を向上させるためのベストプラクティスの導入を検討していきたい。

7. 参考文献

- [1] Watts S. Humphrey: A Discipline for Software Engineering, Addison-Wesley (1995). (邦訳：パーソナルソフトウェアプロセス技法，共立出版，1999年)
- [2] Watts S. Humphrey: A Self-Improvement Process for Software Engineers, Addison-Wesley (2005). (邦訳：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] Watts S. Humphrey: TSP Leading a Development Team, Addison-Wesley(2005). (邦訳：TSP ガイドブック：リーダー編，翔泳社，2007年)
- [4] Watts S. Humphrey: Introduction to the Team Software Process, Addison-Wesley (1999). (邦訳：TSPi ガイドブック，翔泳社，2008年)
- [5] Capers Jones: Software Engineering Best Practices: Lessons from Successful Projects in the Top Companies, McGraw-Hill Education (2009).
- [6] 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，SEC journal, Vol.6, No.3, pp.118-125, 2010年.
- [7] 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，Vol.53, No.10, pp.1084-1087, 2012年.