

## PSP 課題プログラムの実践—プロセス成熟に伴う工数配分と欠陥数の変化

2C-9

阿部 正典, 野中 誠, 種田 圭吾, 東 基衛

早稲田大学大学院 理工学研究科 経営システム工学専門分野

## 1. はじめに

PSP は、ソフトウェア技術者個人のプロセスを測定することで、その改善を示すものである。PSP のプロセス成熟度を向上させることは CMM (Capability Maturity Model) で示された組織のプロセス成熟度を向上させるためにも有効で、欧米では大学や企業での技術者教育に取り入れ、実際の開発プロジェクトへの効果も報告されている。我々は、現在、PSP の課題を実践することにより、その有効性の評価を行っている。今回はこれまでの結果のうち、プロセス成熟に伴う工数配分および欠陥数の変化に関して報告する。

## 2. PSP 課題プログラムと測定内容

【1】には、1A から 10A まで 10 個の課題プログラムが用意されており、その内容は主に統計量を計算するものである。我々はこれらの課題プログラムを実践し、その過程を測定した。【1】は、開発プロセスを Planning・Design・Code・Compile・Test・Postmortem の 6 フェーズに分けている。そして予め用意された計測用テンプレートに従い、各フェーズに費やした時間及び開発中に注入・修正された欠陥等について、課題プログラムの実践と同時に計測していく。なお、本報告における実装言語は Java を用いた。

## 3. 実践結果

我々の中の一人が、1A から 6A までを完了した段階でのデータから、工数配分割合・開発中に発生した欠陥数および開発者の意識の変化について分析した結果を以下に示す。尚、この学生のプログラミング経験及びソフトウェア工学の知識は、平均的な大学院生よりも高い。

## 3.1 課題プログラムの実践に要した時間

1A から 6A までの課題プログラムを実践するのに、この学生の場合で約 25 時間、新規に作成したプログラム LOC (Lines of Code) は 381 行であった。またこの他にも、我々が輪講形式で【1】の 6A までに対応する範囲を読むのに約 15 時間の時間を要した。

## 3.2 工数配分割合の変化

各課題毎の総開発時間に占める、各フェーズの工数配分割合を以下の図 1 に示す。そして、特に配分割合の変動が顕著に見られた Planning・Design・Code・Compile フェーズについて分析結果を述べる。

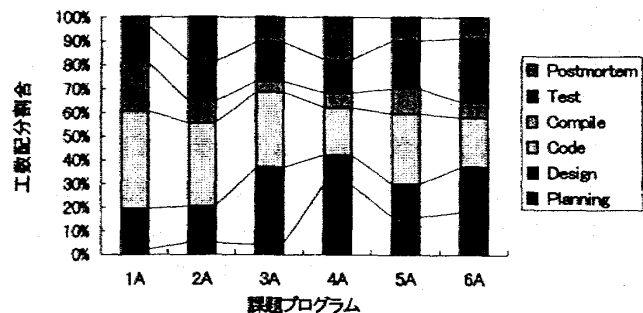


図 1: 開発時間に占める各フェーズの工数配分割合

Planning・Design の工数配分割合は、1A・2A では全体の 20%程度であるのに対して、3A 以降では全体の 30%~40%程度に上昇している。一方 Code・Compile の工数配分割合は、1A・2A では全体の 50%~60%程度であるのに対して、3A 以降では全体の 30%~40%程度に減少している。

これは、科学的な計測方法が開発者に対して意識の改革を促したことによるプロセス成熟の効果といえる。それまで疎かになっていた概要・詳細設計に対する姿勢の見直しがこの学生の中で促されたのが 3A 以降であり、それがコーディング及びコンパイルの生産性の向上へと繋がっているといえる。尚、4A の Planning が突出しているのは、Design において概要設計のミスによる手戻りが生じたためである。

A Case Study of PSP: An impact on effort distribution and defects density

ABE Masanori, NONAKA Makoto, OIDA Keigo and AZUMA Motoei

Graduate School of Science and Eng., Waseda University

### 3.3 欠陥数の変化

Planning・Design の工数配分割合の上昇と Code・Compile の工数配分割合の減少は、開発中に発生した欠陥数の変化にも影響を及ぼしている。各課題毎の 1KLOC (Kilo Lines Of Code) 当たりの欠陥割合を以下の図 2 に示す。

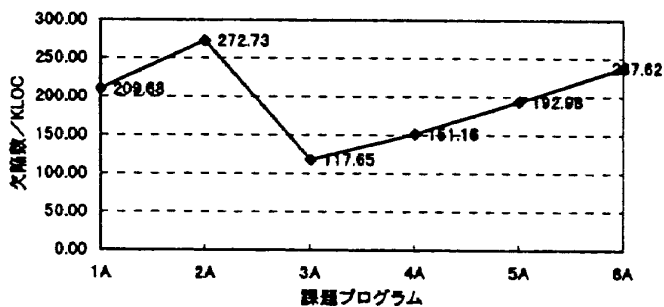


図 2：各フェーズにおける 1KLOC 当たりの欠陥数

2A までは欠陥数が 1KLOC 当たりで 200 個から 300 個程度へ上昇しているが、Planning・Design の工数割合が上昇し始める 3A で欠陥数が 100 個以上急激に減少している。しかし、6A までの結果を見ると最初の 1A の欠陥数を上回ってしまっている。

発生した欠陥のうち Compile・Test で発見された割合がそれぞれ 65.5%・29.8%で、全体の 95.3%と非常に高い。特に Compile での発見が顕著で、発見された欠陥タイプのうち Syntax によるエラーが 58.2%であった。このタイプの欠陥 1 個当たりの修正時間は 1min であったことから、ケアレスミスによる欠陥注入がその原因と考えられる。こうしたミスによる欠陥注入を取り除くことができれば、欠陥数も減少し生産性および品質の向上が期待できる。

### 3.4 開発者の意識変化

この学生は、これまでに自分の開発過程のデータを記録したことはない。そして開発では、概要設計及び詳細設計を綿密には行わずにコーディングする傾向があった。しかし、用意されたテンプレートに従って開発過程を記録し自身のプロセスを数値として眺めることで、これまで場当たり的であった開発姿勢を改めるという意識変化が生じたと思われる。それが 3A 以降に、Planning・Design の工数配分割合の上昇として現れている。また、3A での欠陥数の顕著な減少がその効果として捉えることができる。

## 4. 考察および今後の課題

今回得られた実践結果の分析から PSP の有効性と品質について考察し、今後の課題を述べる。

### 4.1 PSP 課題実践の効果

現時点での被験者が 1 人であったこと、実践した課題プログラムが全体の 6 割であったことから、実践効果の統計的な裏付けは不可能である。しかし、2A と 3A を境界に工数配分割合および欠陥数に顕著な変化が見られたように、科学的な測定により自身の開発プロセスを記録し、それを数値として表現することが、開発者本人の改善意識の喚起を促す一要因になり得る。これをプロセスの成熟として捉えれば PSP の効果として判断することができ、今回の実践においてもその効果が見られたといえる。

### 4.2 品質向上の判断

開発中に発生した欠陥数の変化は 2A から 3A では急激に減少してはいるものの、それ以降再び上昇傾向にあり品質という側面での効果はあまり見られていない。これは 1A から 6A まで、ケアレスミスによる欠陥注入が改善されないまま継続して行われてきたことが主な原因と考えられる。しかし【1】は、6A までの課題で達成されるべきタスクとして測定・見積もり及び計画を挙げており、この時点では品質管理のタスクを割り当ててはいない。レビュー作業等を取り入れた品質管理は次の 7A 以降の課題で実践されるため、継続して実践することで欠陥数の減少および品質の向上が期待できる。

### 4.3 対象被験者の違いによる変化の差違

前述のように今回対象とした被験者は、プログラミング経験が平均より高い。そこで、プログラミング経験の殆どない学生を対象に同様の課題を実践させた場合、工数配分割合および発生する欠陥数の変化に前者の学生とどのような差違が見られるか今後検証していく予定である。また、複数の被験者からデータを収集することで今回の被験者から得られた PSP の有効性を統計的に検証できれば、組織のプロセス成熟度の向上も期待できるとと思われる。

### 参考文献

- 【1】Humphrey, W., A Discipline for Software Engineering, Addison-Wesley, 1995