

ソフトウェア開発プロセスの改善効果予測と制御技術の確立

本多 八郎*, 相磯 正司**, 鈴木 圭一*

Controlling a Software Development Process by Predicting the Effect of Improvements

Hachiro HONDA*, Masashi AISO**, and Keiichi SUZUKI*

Abstract

We have established a continuous quality improvement process by applying the Capability Maturity Model Integration (CMMI) method to our software development activities. The process consists of the following steps: (1) analyzing the key performance indicators of each development phase, (2) predicting the effect of improvements based on the analysis, and (3) monitoring and controlling the quality and cost. In upgrading from the CMMI level 4 to level 5, we focused on improvement of quality in the early development phases and achieved three times better quality in system tests.

1. はじめに

富士フィルムでは医療を始めとする多業種分野で、高性能・高性能な製品を提供するための、ソフトウェアが必要不可欠な技術になってきている。さらにIT技術革新やソフトウェアシステムの巨大化・複雑化が進み、ソフトウェア開発力が一段と重要になったため、ソフトウェア開発管理、品質管理技術向上が求められている。

このような状況のもと、富士フィルム(株)ソフトウェア開発センターと富士フィルムソフトウェア(株)(以下われわれと略す)が共同で、QCD(Q:品質, C:コスト, D:期間)を改善するために、世界で広く認められているCMMI(Capability Maturity Model Integration)¹⁾のモデルを参考に、ソフトウェア開発プロセス改善を積み重ねている。

「X線画像診断システム用画像処理ユニットの開発組織」(以下、「医療系コンソール開発組織」と略す)を対象に、2010年3月に高成熟度モデルCMMIレベル4(以下L4と略す)の認定を受け、その成果を研究報告書「ソフトウェア開発における統計的プロジェクト管理手法の

導入と実践」²⁾として報告したが、今回、プロセスの管理手法をさらにブラッシュアップし、CMMIレベル5(以下L5と略す)の認定を獲得した。

L4からL5へ開発プロセスをブラッシュアップするために実施した改善活動の一部として、組織のデータ分析、改善施策の効果予測、品質と工数の監視・制御技術を、改善目標と改善活動の狙い、プロセス改善活動の取り組み、改善効果の確認、得られた制御技術の成果を踏まえて報告する。

2. 改善目標と改善活動の狙い

はじめにビジネス面から見た改善目標と、改善活動の狙いを明確化し、組織内で共有化した。

(1) ビジネス面から見たプロセスの改善目標

L4の達成に至るプロセス改善で、すでに開発コストを1/2以下に削減する実績を上げた。今回のL5に向けた改善では、「継続的なプロセス改善を組織的に定着させる」ことを主たる目的としたことから、L4段階からさらに品質を向上しつつ、5%の生産性向上を目標に設定した。

本誌投稿論文(受理2012年11月28日)

*富士フィルム(株)R&D統括本部

ソフトウェア開発センター

〒222-0033 神奈川県横浜市港北区新横浜2-10-23

*Software Research & Development Center

Research & Development Management Headquarters

FUJIFILM Corporation

Shin-yokohama, Kouhoku-ku, Yokohama, Kanagawa

222-0033, Japan

**富士フィルムソフトウェア(株)ソフトウェア技術本部
〒222-0033 神奈川県横浜市港北区新横浜2-10-23

**Software Corporate Engineering Division

FUJIFILM Software Co., Ltd.

Shin-yokohama, Kouhoku-ku, Yokohama, Kanagawa

222-0033, Japan

(2) 工数削減の考え方と改善活動の狙い

品質向上とコスト（工数）は相反する要素ととらえがちであるが、われわれは、L4の達成時に確立していた統計的な組織ベースライン分析の結果、上流での品質を向上すれば、全体として修正工数が少なくなり工数削減に繋がることわかった。そのため改善活動の狙いとして「上流工程の品質向上」に注目した。

3. プロセス改善活動の取り組み

目標とした5%の生産性向上効果を上げるために、組織の実績データの分析と改善予測から工程ごとに管理目標値を設定し、対策の監視までのプロセス改善活動に取り組んだ。

(1) 改善策の予測値

i) 組織のデータ分析と改善策の抽出

改善活動を行なった「医療系コンソール開発組織」の各プロジェクトがバージョンアップ開発であったことから、改善するための分析対象を、前バージョンの「最終評価のシステムテストでの不具合と、市場で発生した不具合」とした。

- a) 「最終評価のシステムテストでの不具合と、市場で発生した不具合」の原因として多い項目の特性要因図を作成し、その根本原因を分析した (Fig. 1)。

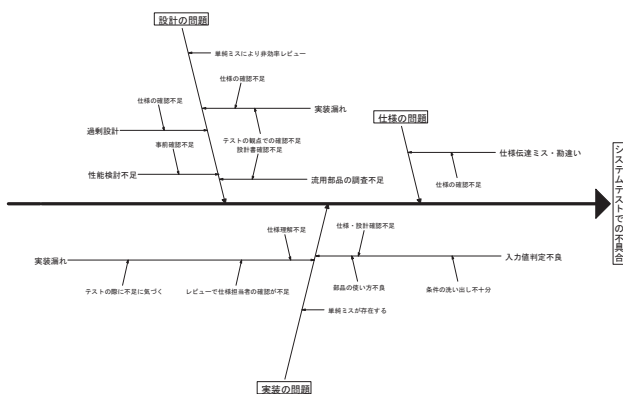


Fig. 1 Cause and effect diagram (fishbone diagram).

- b) a) の根本原因を系統図にまとめ、対応策を挙げ、その効果と実現性を評定して、実施すべき対応策として点数の高いものを選択した (Fig. 2)。

項目	原因	対策	効果	実現性	優先度	担当者	完了日	備考
設計の段階	単独レビューによる見落とし	複数人による見直し	○	○	○	〇		
	仕様書の誤り	仕様書の再確認	○	○	○	〇		
	仕様変更の遅延	仕様変更の迅速な対応	○	○	○	〇		
	仕様変更の漏れ	仕様変更の徹底	○	○	○	〇		
仕様の段階	仕様の誤り	仕様の再確認	○	○	○	〇		
	仕様の遅延	仕様の迅速な対応	○	○	○	〇		
	仕様の漏れ	仕様の徹底	○	○	○	〇		
	仕様の誤り	仕様の再確認	○	○	○	〇		
実装の段階	実装の誤り	実装の再確認	○	○	○	〇		
	実装の遅延	実装の迅速な対応	○	○	○	〇		
	実装の漏れ	実装の徹底	○	○	○	〇		
	実装の誤り	実装の再確認	○	○	○	〇		

Fig. 2 Causes and countermeasures.

ii) 改善策の加算

単独の対応策実施による改善効果1つ1つは大きくはないが、因果関係が独立な施策を対策範囲の規模の重みに換算して加算すると改善効果は大きくなる。この考えに沿って、過去の統計データをベースとした工程ごとに改善効果の加算が見えるツールを開発した (Table 1)。ツールで試算した結果、Table 1の「合計%」のように、狙いどおりに上流工程で不具合を先行して抽出すれば、下流工程で不具合を減らすことが可能であることが予測できた。

Table 1 Degree of quality improvements.

工程	工程 1	工程 2	工程 3	工程 4	工程 5	工程 6
施策 A	11%	11%	11%	-17%	-17%	-17%
施策 B						-16%
施策 C		6%			-6%	
施策 D					-5%	-7%
施策 E	-4%	-2%	-15%			
合計%	6%	15%	-4%	-17%	-28%	-39%

iii) 改善施策効果の簡易予測

平均の差の検定はt-検定を用いて実施するが、毎回正規な手法で実施すると、効果の算出に時間を要するため、改善施策の効果が出ているかどうか簡単に確認できるツールを新たに追加した (Table 2)。これにより、改善施策が統計的に有意差があるほど効果的かどうかを瞬時に予測できる。

Table 2 Simplified version of t-Test.

1. 平均値の差の場合 (t-検定)	
有意水準 α	5%
自由度 ϕ	23
併合分散 V	146.17
統計量 t0	2.77
t (ϕ, α)	2.07
t0 \geq t (ϕ, α) ?	yes (有意差あり)

iv) 採用した予測値

複数の施策による改善効果が、組織のプロセス改善活動の目標に届き、かつ有意差があるほど効果的かを事前に確認するために、ii) の改善策の加算と、iii) の改善施策効果の簡易予測を繰り返した後、最終的な改善施策を採用した。その上でシステムテスト時のソフトウェアの品質がどれくらい改善するかの目標をL4と比較して、システムテストの不具合密度 [件/千ステップ] を以下とした (Table 3)。

Table 3 Estimation of expected improvements in quality.

対象工程	システムテスト
不具合密度の改善目標	L4比 -48%

同様に、システム設計からシステムテストまでの総修正工数 [時間/千ステップ] の改善目標を以下とした (Table 4)。

Table 4 Estimation of expected improvements in performance.

	システム設計～システムテストまでの総修正工数合計
総修正工数の改善目標	L4比 -46%

(2) 対策効果の監視

実績データが予測の管理幅内で推移するかどうかの監視を以下の i) から iii) の方法で実践した。i) と ii) はL4から実施している既存技術で、iii) は新規に追加した監視技術である。

i) プロジェクトごとの工程内品質の監視

各プロジェクトのシステム設計以降の全工程で、サブシステムごとに不具合を確認した (Fig. 3)。そのなかで目標値の幅を逸脱する場合にはその原因を特定し、対策するという方法で監視を続けた。なお、今回の「医療系コンソール開発組織」の開発では一部で既存のソフトウェア部品の流用により、下限値を下回る事例が見られた以外は、管理幅からの逸脱はなく安定していたため、対策や目標値再設定の必要はなかった。

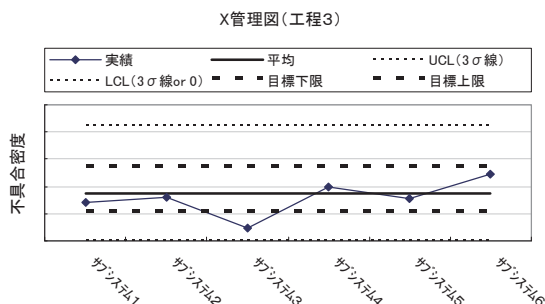


Fig. 3 Quality control graph in a process.

ii) 予測モデルを使った累積品質の監視

プロジェクトごとの各工程で、累積不具合予測モデルを使い、累積不具合が管理幅に入っているかどうかの監視を行なった (Table 5)。そこで、逸脱の予測結果が出た場合には品質の目標を見直すことができる。今回はここでの管理幅の逸脱は観測されず安定しており、予測モデル再設定の必要はなかった。

Table 5 Quality monitoring by our prediction model.

使用工程	予測モデルの工程	定数項	工程1	工程2	工程3	工程4	工程5	工程6	合計 [件/ks]	予測区間 (70%) 上限	予測区間 (70%) 下限
工程2	最終										
工程3	最終										
工程4	最終										
工程5	最終										

iii) 組織の改善効果実績データの監視

「医療系コンソール開発組織」全体で、品質と生産性 (総修正工数) の実績データの推移を週単位に確認した。このために開発した監視ツールを用い、複数のプロジェクトの各サブシステムの品質を入力することにより、組織目標値の管理幅以内かどうかを確認できる (Fig. 4)。

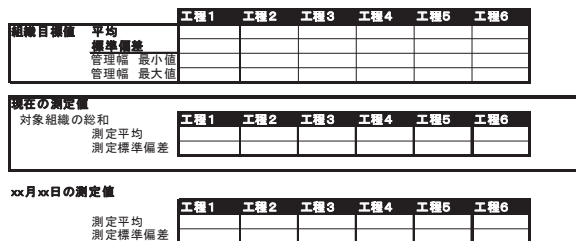


Fig. 4 Weekly monitoring of the organization's target.

4. 改善効果の確認

全プロジェクトの開発終了後、「3.プロセス改善活動の取り組み」の結果を検証した。

(1) 品質の改善施策の効果確認

品質の改善予測に対して結果はTable 6のとおりとなった。より修正工数のかかるシステムテスト工程での不具合密度がL4比で、平均69%下がり、品質が改善していることを確認した。標準偏差の比もL4を1とすると0.35に狭まり安定性が向上した。これは種々の施策の結果としてシステムテストでの不具合件数が大幅に減少したためである。

Table 6 Results of quality improvements.

対象工程	システムテスト	参考
不具合密度の改善目標	L4比 -48%	標準偏差の比 1 → 0.35
不具合密度の改善結果	L4比 -69%	

また、L4とL5でシステムテストの不具合密度をt検定すると、5%水準で有意差が認められるため、品質において統計的にも改善効果があったことが証明できている (Fig. 5)。

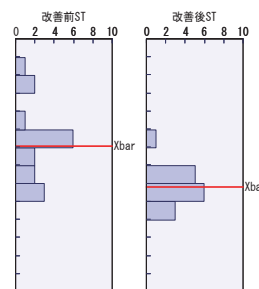


Fig. 5 t-test for quality improvements.

(2) 総修正工数に対する改善施策の効果確認

総修正工数の予測に関して、結果はTable 7のとおりとなった。上流工程施策で品質を向上できたことと、施策「ソフトウェア構造の改革」により、「システムテスト」の修正工数が削減できたため、総修正工数（システム設計以降の各工程の不具合密度と修正時間を掛け合わせた総和）でもL4比で、平均62%下がり、改善施策効果が見られた。予測値は最低限の効果予測で見積もっていたため、結果として修正工数が予測値より、よい値となっている。標準偏差の比もL4を1とすると0.84に狭まり安定性が向上した。これは改善施策により解析時間が短くなり安定したことが原因である。

Table 7 Results of performance improvements.

	システム設計～システムテストまでの総修正工数合計	参考
総修正工数の改善目標	L4比 -46%	標準偏差の比 1→0.84
総修正工数の改善結果	L4比 -62%	

また、総修正工数をt-検定すると5%の水準で有意差が認められるため、生産性（修正工数）において統計上の改善効果があったことが証明できている（Fig. 6）。

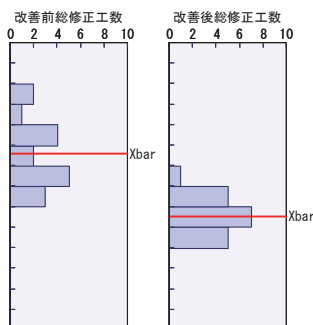


Fig. 6 t-test for performance improvements.

(3) ビジネス上の効果

一連の改善施策と監視の結果、開発工数平均は6%の削減になっており（Fig. 7）、5%の生産性向上の目標を達成し開発費削減に貢献できた。

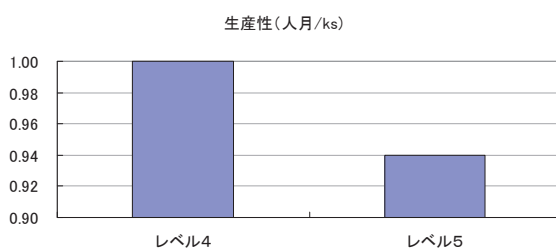


Fig. 7 Productivity improvements from CMMI-L4 to L5.

またL4からL5への活動で、システムテストでの品質は不具合密度の値で3倍に向上した（Fig. 8）。

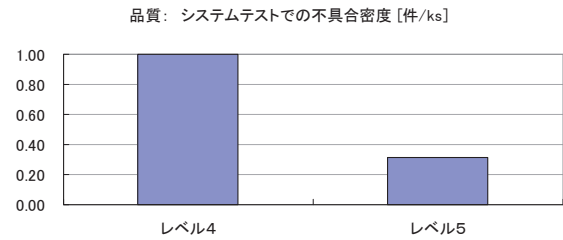


Fig. 8 Quality improvements from CMMI-L4 to L5.

5. 得られた制御技術の成果

プロセス改善活動をとおり、L4達成時と比較して予測から監視までの制御技術確立の成果が得られた。

○改善策の効果予測技術

- ・改善策を加算した改善効果の予測手法が確立できた。
- ・品質と生産性について統計的な効果の有無を事前に予測する手法が確立できた。

○組織全体の目標設定から監視までの制御技術

6. まとめ

CMMIのモデルを参考に、2002年以来ソフトウェア開発プロセス改善を積み重ねてきた。今回さらにCMMIの高成熟度モデルを「医療系コンソール開発組織」に導入することで、組織のデータを分析し、複数の改善施策の効果を統合的に予測し、品質と工数を監視する制御技術を確立した。

公式アプレイザルにて2011年12月にCMMI（V1.3）-L5（最適化している段階）を達成できたと同時に生産性向上目標を達成し開発工数削減に貢献することができた。

改善は現在も進行中であり、今後L5を達成した「医療系コンソール開発組織」のソフトウェア開発プロセスにおけるノウハウや仕組みを、社内のソフトウェア開発プロセスの改善に適用して、開発組織全体のレベルアップに繋げていく。

参考文献

- 1) 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. CMMI for Development, Version 1.3 (2010).
- 2) 相磯正司ほか. ソフトウェア開発における統計的プロジェクト管理手法の導入と実践. 富士フィルム研究報告. No.56, 31-34 (2011).

（本報告中にある“CMMI”は、米国・カーネギーメロン大学の登録商標で、同大学のソフトウェアエンジニアリング研究所が開発した、開発組織のプロセス成熟度を5段階で評価する指標です。）