

ソフトウェア開発のコストダウンモデルとその適用

青山 幹雄 小笠原 喜典 藤本 洋

富士通(株) 複合交換機事業部

近年のソフトウェア開発規模の増大に伴い、ソフトウェア開発コストは、その絶対額と共にシステム開発の総コストに占める比率も増加する傾向にある。実務においては、生産性、品質と共に、コスト管理が極めて重要になっている。このため、ソフトウェア開発のライフサイクルを通じた総括的なコストダウンモデルを開発した。本モデルでは、コストダウンを、要求仕様、開発プロセス、開発プロダクト、プロジェクト管理の4因子に分け、これらのコストダウンの積として定式化する。コストダウンを実務に適用するための技術面と組織面からの方法も開発した。この方法を実際の大規模開発に適用した結果、総コストを約10%削減できた。さらに、開発プロセスに沿ってコスト分析を行った結果、各工程において、コストダウンだけでなくコストアップの要因も見出した。

A COST-DOWN MODEL OF SOFTWARE DEVELOPMENT AND ITS PRACTICAL APPLICATIONS

Mikio Aoyama, Yoshinori Ogasawara and Hiroshi Fujimoto

*Business Switching Software Division
Fujitsu Limited*

2-12-5 Shimo-kodanaka, Nakahara-ku, Kawasaki 211, JAPAN

ABSTRACT

Software development costs are ever increasing at an alarming rate in both its amounts and its ratio to the total costs spent for developing systems. In practice, the importance of cost management is no less than the productivity and quality. Therefore, we have developed a cost-down model whose mission is to reduce the total costs of software development. In developing the model, we have identified four major factors which consists of software development costs: requirements, development process, software products, and development management. The total cost-down is formulated as a product-form of the four cost-down factors. To apply our cost-down model to practical development, we have developed technical methods as well as organizational methods of cost-down based on software assets reuse concept. We have attained some 10% cost reduction, as a whole, in developing a large-scale software system. Precise cost analyses along with development process have revealed various dynamic characteristics of development costs.

1. はじめに

近年、通信サービスが多様かつ高度になるに伴い、開発される通信ソフトウェアは、急速に大規模かつ複雑になっている。このため、通信システムにおけるソフトウェア開発コストは、その絶対額と共にシステム開発の総コストに占める比率も増加する傾向にある。従来から、ソフトウェア開発の生産性と品質を向上するため、技術と管理の両面からさまざまな努力が払われている。特に、ソフトウェア開発のコスト推定モデルの分野では、種々の方法が開発されている[1-4]。しかし、開発の総コストを削減する手法についてのアプローチは少ない。実務の観点からは、技術面と共にコスト管理が極めて重要である。さらに、このようなコスト削減活動をソフトウェア技術者に留まらず、企業において全社的に推進するしくみ作りが望まれる。

このような背景の下に、本稿は、ソフトウェア開発における総コストを削減するための「コストダウン」モデルとその実開発への適用経験について報告する。本稿で提案するコストダウンとは、生産性向上を包含して、より広範囲な視点から開発コストを低減するアプローチである。

まず、2章でソフトウェア開発コストの問題点を挙げ、3章で問題に対する解決アプローチを述べる。4章ではコストダウンのモデルを提示し、このモデルに基づくコストダウンの具体的方法について5章で述べる。6章は、本手法を大規模通信ソフトウェア開発に適用した経験を報告する。最後に、今後の課題についてふれる。

2. 問題の背景

本研究の狙いは、要求仕様の分析から開発、保守に至る通信ソフトウェア開発のライフサイクル全体に亘る包括的コストダウンの実現にある。コストダウンを行うために、通信ソフトウェアの開発において特に重要だと考えられる次の2点に着目した。

2.1 機能追加主体の開発

大規模通信ソフトウェアは長寿命であり、一般に長期に渡り機能追加が繰り返される。したがって、ソフトウェア開発コストのほとんどは機能追加に費やされる。さらに、通信ソフトウェアのような実時間分散処理アーキテクチャを採るソフトウェアの機能追加に伴う既存ソフトウェアへの影響は極めて複雑であり、機能追加でのソフトウェア生産性は新規開発に比べ低いことが指摘されている[5]。このため、対象システムのソフトウェア規模の増大に伴い開発コストは益々増大する傾向にある。

一方、このような長期にわたる機能追加主体のソフトウェア開発では、開発プロセスや開発プログラムの枠組みが定まっており、かつ、繰り返される。したがって、開発プログラムに加え開発プロセスなども資産として蓄積し再利用しやすい状況にある。

2.2 分散並行開発

顧客の通信ソフトウェアへの要求の多様化や開発サイクルの短縮などのために、複数の分散した開発拠点が並行して複数機能を開発する「分散並行開発」を行っている[6, 7]。このような開発形態では、複数の拠点間や各拠点内における複数グループ間での、開発資源の管理が極めて困難である。グループ間の開発資源の運用効率を向上し、システム全体の開発コストを削減する必要がある。

3. アプローチ

通信ソフトウェアのような長期に亘る機能追加型の開発では、従来のプログラムや設計情報などの開発プログラムに加え、要求仕様や、開発プロセス、さらには開発管理技術を含むより広範囲な開発情報を「資産」として活用することが期待できる。この点に着目し、「資産活用」に基づく開発コストダウンのアプローチを提案する。資産活用のアプローチは、従来の「ソフトウェア再利用」のアプローチを、成果物に留まらず開発プロセスやプロジェクト管理の手法へ拡張したものと考えられる。

3.1 ソフトウェア開発モデル

ソフトウェア開発を、図-1に示すマクロモデルで考える。ソフトウェア開発の入力は、顧客の要求仕様であり、出力はプロダクト（ソフトウェア製品）である。この製品を開発するため、開発プロセスに沿って開発資産を利用しながら開発資源を投下する。従って、開発の目標は、限られた予算内で最高の顧客満足度を達成するために、資産活用を最大にし、かつ、資源配分を最適化することにある。

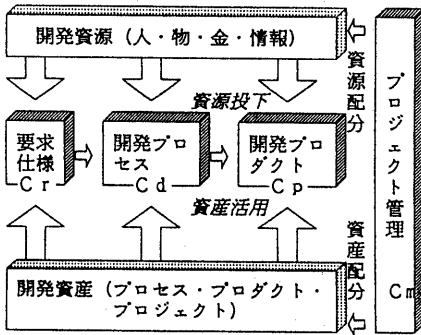


図-1 ソフトウェア開発モデルとコストダウン

3.2 ソフトウェア開発資産とは

ソフトウェア開発資産として、従来の開発プロダクトに加え、次の3つの資産の枠組みで捉える。

- (1) プロセス資産：ソフトウェア開発プロセスとそれに付随するノウハウなど。
- (2) プロダクト資産：ソフトウェア開発の生産物であり、ソースプログラム、設計情報、仕様などを含む。
- (3) プロジェクト資産：ソフトウェア開発プロジェクトを運営管理するための要員、設備、資金、情報など。

3.3 開発コストモデル

図-1の開発モデルに基づき、総コストダウンは、次の式(1)によって表わされる。

$$C = C_r \times C_d \times C_p \times C_m \quad (1)$$

但し、各コストダウン因子は次のように定義される。

- C_r：要求仕様によるコストダウン
- C_d：開発プロセスによるコストダウン
- C_p：開発プロダクトによるコストダウン
- C_m：開発プロジェクト管理によるコストダウン

4. コストダウンモデル

図-1のソフトウェア開発モデルに基づくコストダウンモデルを図-2に示す。複数プロジェクトに亘るシステム全体、あるいは、企業全体のソフトウェア開発コストダウンを図るため、仕様、開発、プロジェクト管理の観点から次のコストダウン要素を抽出した。

(1)仕様コストダウン

既存仕様の再利用や組み合わせなどにより、実際に開発が必要な仕様を削減する。

(2)開発コストダウン

開発コストダウンはプロセスとプロダクトのコストダウンにより実現される。

①プロセスコストダウン：開発プロセス分析に基づき、プロセスの自動化やムリ、ムダ、ムラの排除により開発工数を削減する。

②プロダクトコストダウン：ソースプログラムや設計情報などのプロダクトを再利用し開発量を削減する。

(3)プロジェクトコストダウン

各開発グループやリリース（版）ごとに必要な開発資源の配分を最適化し、プロジェクト間や工程間での「手待ち」など、資源と資産の運用ロスを削減する。

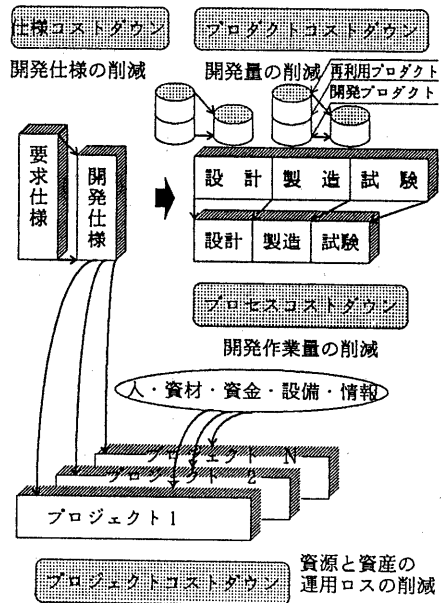


図-2 コストダウンモデル

5. コストダウンの仕組み

コストダウンモデルに基づいたコストダウンの仕組みと推進手段を具体的に述べる。コストダウンを実現するためには開発資産の活用による生産性向上の推進が必要であることから、図-3に示すようにコストダウンと生産性向上を一体化して推進する。

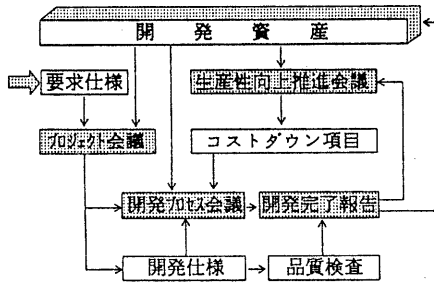


図-3 コストダウンの仕組み

5.1 仕様コストダウン

設計プロセスの開始に先立ち図-4に示す仕様コストダウンによって開発量の削減を図る。仕様コストダウンは、次の2つのステップによって実現する。

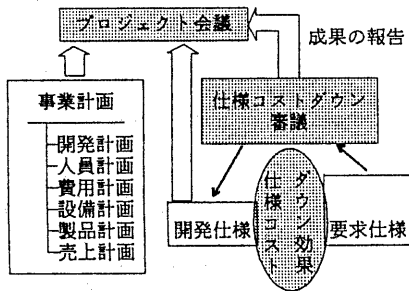


図-4 仕様コストダウンの仕組み

(1)仕様コストダウン審議

長期に渡って機能追加を繰り返すことによって開発仕様の蓄積は進んでいる反面、顧客の要求仕様を満たすために開発仕様の再利用が行われず、類似の仕様を多く設計している。また、設計を開始した後での蓄積された開発仕様の再利用は、手戻りも多く開発コストの増加を招く。そこで開発を着手する前に要求仕様を満たす開発資産を活用した開発仕様を実現することによってコストダウンが図られる。仕様コストダウン審

議では、開発資産の再利用に着目した開発仕様の決定と仕様コストダウン効果の妥当性を審議する。

(3)プロジェクト会議

要求仕様を満たす開発仕様が決された後、開発量及び開発工数に応じた開発資源の配分を目的とした開発計画の策定を行う。プロジェクト会議では、開発資源の効果的配分を図る目的で事業計画に占める開発資源の妥当性を審議する。

また、資源の配分だけではなく開発プロセスにおける開発コストダウンの目標を定量的に設定することにより開発者に対するコストダウン意識の高揚を図る。

5.2 資産活用による生産性向上の推進

コストダウンを推進するためには、コストダウン項目の抽出が必要となる。そこで開発資産の活用による生産性向上の施策をコストダウン項目として位置づけ、さらに開発工数及び開発量の削減効果を定量的に把握する仕組みが必要となる。

(1)生産性向上施策とコストダウン項目

生産性向上を達成するためには、具体的な方法を明確にする必要がある。そこで、生産性向上の具体的な方法を抽出するために、図-5に示すソフトウェアの3つの資産を活用する。この生産性向上の施策は、コストダウンの観点から効果を評価しコストダウン項目として蓄積する。

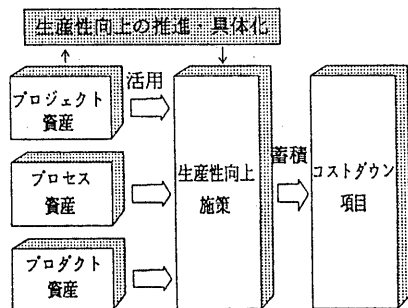


図-5 生産性向上とコストダウン項目抽出

(2)コストダウン効果の定量的把握

生産性向上の施策が、開発プロセスの中で期待するコストダウン効果を上げているか定量的に把握する仕組みを図-6に示す。開発仕様に基づく開発量は、開

発が進むにともない計画段階の見積もりと比較して増減することが予想される。そこで設計、製造及び試験の各段階で開発量の再評価を行い、最新の開発量に対するコストダウン効果の妥当性を評価する。

また、コストダウン効果だけでなく開発工数が計画段階の見積もりを超過する場合には、原因を究明し早期に対策を取ることでプロジェクトコストダウンを効果的に実現する。

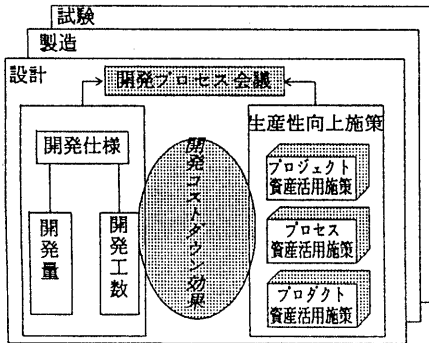


図-6 開発コストダウン効果の把握

5.3 コストダウン推進方法

コストダウンを推進する組織的な取り組みを表-1に示す。

表-1 コストダウン推進方法

推進手段	目的
プロジェクト会議	<ul style="list-style-type: none"> ○仕様コストダウン効果の承認 ○開発計画の審議 ○開発コストダウン目標の設定 ○プロジェクトコストダウン施策の策定
開発プロセス会議	<ul style="list-style-type: none"> ○開発プロセス定義に基づく開発工数の妥当性の評価 ○開発コストダウン効果の把握 ○各プロセスに対するプロジェクトコストダウン施策の設定
生産性向上推進会議	<ul style="list-style-type: none"> ○各資産の活用による生産性向上施策の計画 ○施策の実現
開発完了報告会	<ul style="list-style-type: none"> ○コストダウン成果の把握と蓄積 ○新たな資産活用に向けてのフィードバック

長期に亘る通信ソフトウェア開発の中でコストダウンを推進するには、開発管理技術だけでなく開発者のコストダウン意識と一体化して進める必要がある。これを推進する具体的なプロセスを以下に示す。

(1) プロジェクト会議

仕様コストダウンの効果及び開発計画の妥当性を審議する具体的な方法を図-7に示す。顧客からの要求仕様段階でのプロジェクト開発計画を策定し、要求条件の妥当性や資産活用に着目した要求仕様の審議を行う。審議結果、妥当と判断された要求仕様に関り具体的な実現を目的とした開発仕様の検討に着手し、完了次第、開発仕様に基づく開発計画書を策定する。その結果、要求仕様と開発仕様との開発量の差を式(2)の仕様コストダウン効果と定義する。

$$Cr = \frac{\text{要求仕様の開発量} - \text{開発仕様の開発量}}{\text{要求仕様の開発量}} \quad (2)$$

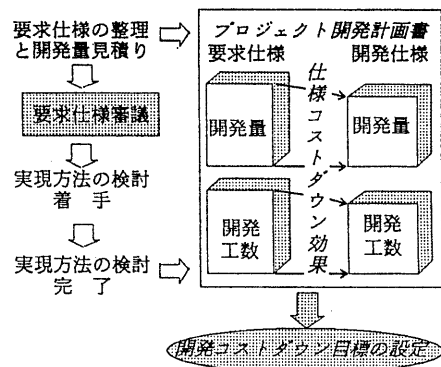


図-7 開発計画書の審議方法

図-8に示す開発計画書には、開発仕様に基づく開発量と開発工数に対する開発コストダウン後の目標値を明確にし、さらにコストダウン前後の目標開発費用を設定する。従って、全開発プロセスに渡り各プロセス毎のコストダウン目標値を設定する。この結果、開発計画書は開発プロセスにおける開発コストダウンのベースとする。開発プロセスが長期に亘る場合や開発が進むに伴い開発工数の見直しが生じた場合、コストダウン目標を見直し開発計画を再度策定する。従って常に最新の開発状況に応じた開発計画書は、コストダウンの目標だけでなく、開発全般を通した開発管理や第3者による開発状況を把握する手段としても活用出来る。

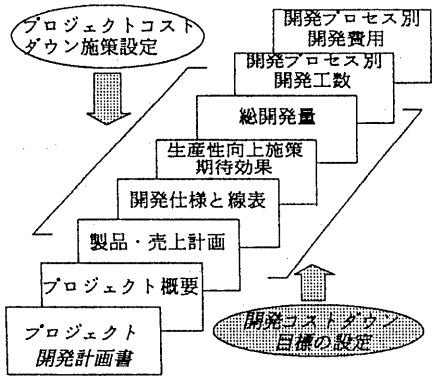


図-8 開発計画書の構成

(2)開発プロセス会議

開発プロセス毎に実施する開発プロセス会議の推進方法を図-9に示し、また本会議で用いる開発プロセス計画書の構成を図-10に示す。

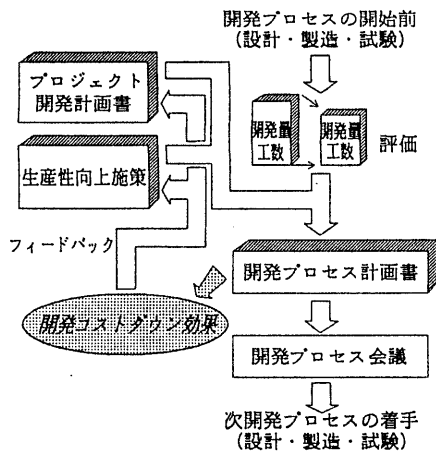


図-9 開発プロセス会議の推進方法

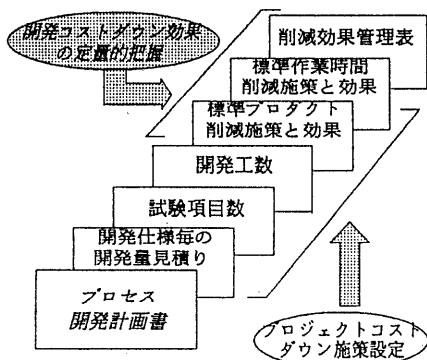


図-10 開発プロセス計画書の構成

開発プロセス計画書は、プロジェクト開発計画書のコストダウン前の開発量と開発工数を基準とし、開発コストダウン効果を定量的に把握する目的で用いる。

プロジェクトコストダウンCpは、各プロセスで設定した標準プロダクトのコストダウン効果による削減量として式(3)で定義する。

$$C_p = \frac{\text{標準プロダクト量} - \text{開発プロダクト量}}{\text{標準プロダクト量}} \quad (3)$$

例えば製造プロセスでは、プロダクト資産を活用することによりコーディング量を削減することが出来る。一方、機能追加主体の開発では、既存仕様を含む試験項目数の増加が想定される。プロセス会議では、コストダウン効果によるプロダクトの削減だけではなく、開発プロダクトの品質も含めて総合的に評価する。

プロセスコストダウンは、各開発プロセスに設定した標準作業時間のコストダウン効果による削減量として式(4)で定義する。

$$C_d = \frac{\text{標準作業時間} - \text{開発作業時間}}{\text{標準作業時間}} \quad (4)$$

開発プロセスは作業レベルに応じて階層化される。さらに階層化された作業プロセスは、作業要素毎に分析し、詳細化されている。詳細化した作業プロセスには、過去の開発プロジェクトで得た実績を基準とし標準作業時間を設定する。従って、プロセスコストダウンはコストダウン項目が適用される詳細プロセスの標準作業時間の削減効果によって把握出来る。図-11にプロセス分析とコストダウン効果の算出例を示す。

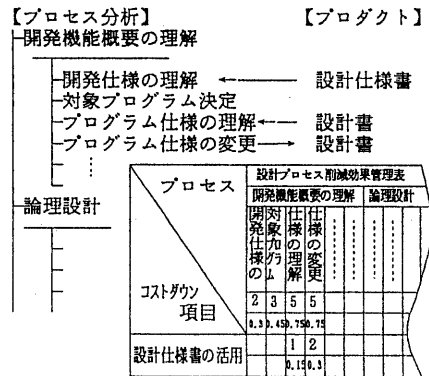


図-11 詳細プロセスとコストダウン効果

標準作業時間は、開発ソフトウェアの外部条件（言語、プロセッサ種別）に応じて設定するが、同一条件下の異なる機能追加では、同一の標準作業時間を設定する。また、開発条件や資産活用に伴って標準作業時間を開発プロセス計画書の中で補正し、作業プロセスと標準時間の適正化を図る。

(3)コストダウン活動のフィードバック

仕様コストダウンから始まったコストダウンは、開発プロセスに沿った開発コストダウンを繰り返し、さらに開発資源の最適化を図るためのプロジェクトコストダウンを加え、一連の開発プロセスの完了に伴い最終する。プロジェクト開発計画時に想定したコストダウン目標の達成度と開発資産の蓄積、さらに、今後のコストダウンへのフィードバックと課題の抽出を図るために図-12に示す開発完了報告会を開催する。本報告によりコストダウンを実践した開発者の継続したコストダウンの推進が期待出来る。

完了報告会はコストダウンの結果だけでなく、ソフトウェア製品の品質検査結果をも合わせて討議することにより品質向上活動と一体化して推進する。

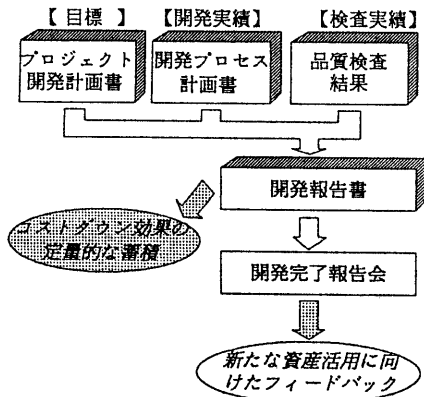


図-12 活動のフィードバック

6. 適用と評価

本方法を大規模通信ソフトウェア開発に適用した効果例を図-13に示す。開発量は仕様コストダウンに伴い21%削減し、また開発工数は開発計画時に7%削減の目標を設定したが、コストダウン効果により8%削減出来、当初の目標を達成した。開発費用は、

仕様コストダウン前の見積りに対し26%削減出来た。

なお、本活動を適用したソフトウェアは、十数ユーザに製品出荷され順調に稼働しており、品質面でも当初の目標を達成している。このように開発初期から一貫したコストダウンを推進することにより目標を上回るコストダウンを達成したことは高く評価出来る。

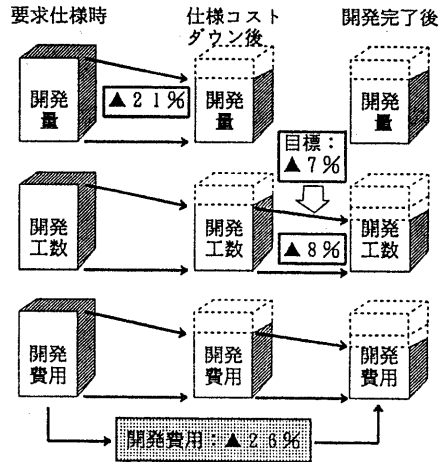


図-13 コストダウンの効果例

開発プロセスのコストダウン効果を各プロセス別の開発工数に着目し、分析した結果を図-14に示す。

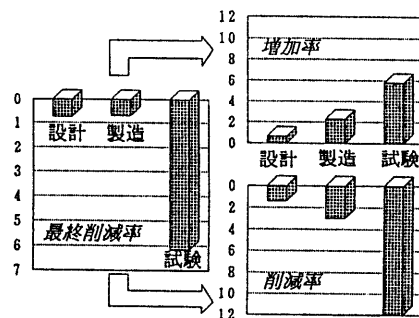


図-14 各プロセス別の開発工数増減率

開発プロセス毎に削減効果を分析した結果、試験プロセスの開発工数が最も削減出来た。一方、設計及び製造プロセスの削減率が低いことも判明した。さらに開発工数の増加要因を分析した結果、本適用システムでは、プロダクト資産の増加等から全開発プロセスで開発工数が8.5%増加していることが判明した。

各プロセス別に開発工数の増減状況を分析した結果、設計プロセスでは仕様コストダウン時に実施した設計仕様の具体化検討の成果によって1.2%の工数が削減出来た。一方、開発資産の増加に伴い設計範囲の抽出に関わる作業時間が、標準作業時間を越えたために0.4%の工数が増加した。従って、総工数は0.8%の削減が出来た。

製造プロセスでは、他プロセスに比べ自動化等の生産性向上が最も進んでいる。さらに新たな資産活用施策の導入や開発者による作業の改善を実施し、3%の工数が削減出来た。一方、並行開発に伴う資産競合や施策の導入不慣れ等から2.2%の工数が増加した。従って、総工数は0.8%の削減が出来た。

また、試験プロセスでは、プロダクト資産の活用により試験項目や試験手順書の作成量が軽減されたことにより1.2%の工数が削減出来た。一方、機能追加主体の開発による既存資産の品質を確保するための試験工数の増加等から5.6%の工数が増加した。従って、総工数は6.4%の削減が出来た。

以上の分析結果から各プロセスでは、コストダウンの効果だけでなくコストアップに結びつく要因が明らかになった。したがってコストアップの要因を抑制することがソフトウェア開発のコストダウンを推進する上で重要である。

7. 今後の課題

今後のソフトウェアの需給予測を考慮すると、本稿で提案したコストダウンを一層推進する必要がある。本稿の適用事例は、単一プロジェクトであったため、プロジェクトコストダウンの評価は得られていない。今後、複数プロジェクトにまたがるコストダウン効果の評価を行う必要がある。また、コストダウンを効果的に進めるためには、ソフトウェア開発の根本に立ち帰り、開発プロセスの改善が必要である。コストダウンモデルを実際の開発に適用して得られた定量的なデータに基づき、今後、開発プロセスの分析と改善を推進する。また、プロダクト、プロジェクト管理を含めたソフトウェア開発資産の活用を合わせて推進する。

8. まとめ

ソフトウェア開発のマクロモデルに基づく、開発コストダウンモデルとその実現方法、ならびに大規模開発への適用成果について報告した。

システム開発コスト占めるソフトウェア開発コストが70%を越えている現状では、このようなソフトウェア開発の包括的コストダウンの推進が急務である。今後のソフトウェア開発需給の傾向を考慮すると、このようなコストダウンのモデル化とその実現方法の開発を推進する必要があると高まるであろう。

一方、コストダウンを実現する根本は開発プロセスの改善にある。本稿で述べたコストダウンの狙いは、コストの観点から開発プロセスを評価し、改善する手段を提供することにある。

本稿のアプローチは個々の開発でも成果が得られているが、分散並行開発においては複数プロジェクトに渡って適用することにより、シナジー効果が期待できる。全社品質管理(TQC)のように「全社コスト管理(TCC: Total Cost Control)」の推進を提唱したい。

最後に、本稿で述べたコストダウンモデルの開発と適用にご協力頂いた、当社複合交換機事業部のソフトウェア開発部門の関係各位に感謝致します。

参考文献

- [1] B. W. Boehm, Software Engineering Economics, Prentice-Hall, 1981.
- [2] T. DeMarco, Controlling Software Projects, Yourdon Inc., 1982.
- [3] R. H. Thayer (ed.), Software Engineering Project Management, IEEE Computer Society, 1988.
- [4] B. W. Boehm and P. N. Paraccio, "Understanding and Controlling Software Costs", IEEE Trans. Software Engineering, Vol. 14, No. 10, Oct. 1988, pp. 1462-1477.
- [5] W. Yu, et al., "Software Productivity Measurements", AT&T Technical Journal, Vol. 69, No. 3, May. 1990, pp. 110-120.
- [6] 青山 誠 "交換ソフトウェアの分散並行開発支援環境", 電子情報通信学会・交換システム研究会, SSE90-25, 1990年6月, pp. 7-12.
- [7] M. Aoyama, "Distributed Concurrent Development of Software Systems: An Object-Oriented Process Model", Proc. IEEE COMPSAC '90, Nov. 1990, pp. 330-337.