

〈論文〉

〈Paper〉

## ソフトウェア構成管理システムによる プロセス分析と改善手法についての考察

### A Study of Process Analysis and Improvement with Software Configuration Management System

吉 崎 浩 二

YOSHIZAKI Kouji

上武大学経営情報学部, 〒370-1393 群馬県高崎市新町270-1

*Faculty of Management Information Sciences, Jobu University, Takasaki, Gunma, 370-1393, Japan*

受付 2006年10月3日

Received 3 October 2006

## 抄 録

ソフトウェア開発プロジェクトにおいて開発プロセスの実態を見る努力と見せる仕組みを築き上げていくことは管理面のみならず、技術面においても大変重要で意義のあることである。しかしながらソフトウェア工学の研究とその実践が求められた1960年代の「ソフトウェア危機」の時代から今日まで実データに基づいたソフトウェア開発プロセスの実態を見せる仕組みはまだまだ十分といえない。

オフラインでバッチ的にデータを収集し分析するケースはまま見受けられるが、オンラインで構成管理の仕組みを活用してしかも組織的な取り組みで開発プロセスの実態を見せる仕組みはまだまだ十分といえない。さらに、構成管理システムのバージョン管理に関する事例研究はいくつか報告されているが、構成管理システムに保存されたデータを分析し、プロセス改善に活用する手法については十分に研究されていない。

本論分は、センター方式によるソフトウェア構成管理システムの活用推進方法とそれにとどまらず、ソフトウェア開発プロセスの品質、コスト、開発期間の改善面に貢献できるソフトウェア構成管理システムによるプロセス分析と改善手法について提言するものである。

キーワード：ソフトウェア開発プロセス、ソフトウェア構成管理、プロセス分析、プロセス改善

## Abstract

It is absolutely important and significant both in management and engineering to make efforts to develop a system to observe the actual status in current software development process (SDP). However, such a system to observe and analyze current status of SDP has not been thoroughly researched since the software crisis era in the 1960's.

The offline or batch system has been studied to some extent, but the online and organized system, with the help of software configuration management system (SCM), has not been developed enough to observe and analyze SDP. Several studies of SCM about version management have been done.

But, the study of the method to analyze the product and process data from SCM has not been given sufficient attention.

This report is to propose not only a centralized approach of SCM, but also an approach to analyze and improve quality, cost and time of SDP, with the product and process data of SCM.

*Key words and phrases:* software development process; software configuration management; process analysis; process improvement

# ソフトウェア構成管理システムによるプロセス分析と 改善手法についての考察

吉 崎 浩 二

## 目 次

1. はじめに
  2. 構成管理とは／構成管理の普及と現状
  3. SCMの位置づけ
    - 3.1 ISO9000とSCM
    - 3.2 CMMとSCM
    - 3.3 PDMとSCM
  4. センター方式によるソフトウェア構成管理の展開
    - 4.1 現状と課題
    - 4.2 システムの導入
    - 4.3 システムの普及について
    - 4.4 利用確認プロセスによるユーザの声の反映
    - 4.5 ソフトウェア開発管理力強化施策における位置づけ
  5. ソフトウェア構成管理システムによる開発プロセス分析と改善手法に  
ついての提言
    - 5.1 基本構想
    - 5.2 開発プロセス分析システムの仕組み
    - 5.3 段階的な成熟レベルアップへのアプローチ
    - 5.4 実施事例
  6. まとめ
- 参考文献

## 1. はじめに

構成管理 (Configuration Management, 以降 CM と称す) とは、製品の構成要素を識別し、これに対する変更を管理する手法である。ハードウェアの構成管理は、第2次世界大戦の初期からロッキードなどで手作業であるが開始された。当時、米国国家基準局などがいくつかの規定を作成しているが、コンピュータ・プログラムの構成管理についてはほとんど何も書かれていなかった。

1970年代になると、ソフトウェアも含むシステムとしての構成管理の重要性への認識が高まり、1980年代には、米国のDOD規格にソフトウェア構成管理 (Software Configuration Management, 以降 SCM と称する) が組み込まれ、米国の産業界に多くの影響をもたらしている。

日本の企業においては、SCMの導入率はせいぜい10%程度で十分に活用されているとはいえず、今後も構成管理を活用するだけでソフトウェア開発プロセスの改善に大きく貢献できる余地を残している。<sup>(3)</sup>

また、SCMがいくつかの事例研究により、ソフトウェア開発における変更管理やバージョン管理に大いに貢献できることが報告されているが、構成管理プロセスまたは構成管理システムに保存されるデータを分析し、プロセス改善に役立てる改善手法については十分研究され、報告されてはいない。

本論文は、センター方式によるソフトウェア構成管理システムの活用推進手法とそれに止まらずさらに、ソフトウェア開発プロセスの品質、コスト、開発期間の改善面に貢献できる「ソフトウェア構成管理システムによるプロセス分析と改善手法」について提言するものである。

## 2. 構成管理とは／構成管理の普及と現状

構成管理 (Configuration Management, 以降 CM と称す) とは、製品の構成要素を識別し、これに対する変更を管理する手法である。JISでは「システムの構成要素の機能的および物理的特性を文書化し、それらの特性の変更を管理し、変更の過程および実現状況を記録・報告し、指定された要件が満たされることを検証する (JIS X 0020)」と定義している。

ハードウェアの構成管理は、第2次世界大戦の初期からロッキードなどで手作業であるが開始された。構成管理は、米国の防衛産業において、低い品質、部品の誤発注、または、部品の不適合、などの問題を解決するために始まった。構成が単純であったころの生産は設計者個人でも管理できたが、ジェット機のような複雑な製品では、それが不可能になっ

たばかりでなく、設計の継続的な変更が避けられなくなってきた。<sup>(3)</sup>

当時、米国国家基準局などがいくつかの規定を作成しているが、しかし、当時のシステムには実質的にはソフトウェアが含まれていなかったのもので、これらの規格はハードウェアに関するものが中心であり、コンピュータ・プログラムについてはほとんど何も書かれていなかった。

1970年代になると、それまでハードウェア中心の構成管理にも、ソフトウェアも含むシステムとしての構成管理の重要性に対する認識が深まり、ソフトウェア構成管理 (Software Configuration Management、以降SCMと称する) の必要性が高まった。1980年代には、米国のDOD規格に組み込まれ、米国の産業界に多くの影響をもたらしている。

日本においては、防衛産業でのアプリケーションシステムからではなく、メインフレームのオペレーティングシステムからはじまった。機器メーカーの埋め込みソフトウェアの分野では、いまだに、ハードウェアの構成管理にとどまり、かなり遅れているといわざるを得ない。ハードウェアの生産会社の多くは、製品がソフトウェア集約的な製品に変貌しているにもかかわらず、ハードウェア指向の生産体制に固執してきたといえる。SCMはダイナミックに変化するソフトウェアにおいて、製品の構成と状況、及び変更を管理する上で大きく貢献する。

欧米ではSCMはまさに成熟期を迎えているが、残念ながら日本では、まだまだ、未成熟期である。日本の企業においては、SCMの導入率はせいぜい10%程度で十分に活用されているとはいえず、今後も構成管理を活用するだけでソフトウェア開発プロセスの改善に大きく貢献できる余地を残している。<sup>(3)</sup> 特に、機器メーカーや中小規模のソフトウェア開発会社の実施例は少ないようである。

多くの事例は、プロジェクト毎の活用事例であり、企業組織の一部でしかも一時的な活用に終わってしまっている例が多い。企業組織全体が取り組む継続的なセンター方式によるソフトウェア構成管理の推進は十分とはいえない。

### 3. SCMの位置づけ

#### 3.1 ISO9000とSCM

ISO9000は、主としてハードウェアの製作を行う場合の品質管理基準について規定している。したがって、内容はソフトウェア開発・製作を十分に意識した内容にはなっていない。ISO9000-3はこの欠陥を補充するために設定されたもので、ソフトウェアの開発・製作に限定した基準のガイドラインを示す。

ISO9000-3の5ではソフトウェアの開発／運用プロセス (たとえば、要求仕様、設計と製

造、テストと妥当性確認、保守など)が設定されており、ISO9000-3の6ではプロセス全体を支える支援活動(たとえば、構成管理、文書管理、ツールおよび技術、購買、教育・訓練など)が設定されている。ISO9000で明示されているソフトウェア構成管理は、したがって、プロセス全体の管理をサポートするための基本的な管理技術の一つとして位置づけられている。(1)(2)(4)(5)

### 3.2 CMMとSCM

ソフトウェアのための Capability Maturity Model (CMM) は米国の公的研究開発センターであるカーネギーメロン大学 Software Engineering Institute (SEI) によって開発された。このモデルは当初、政府にソフトウェアを納入する企業の「ソフトウェア製造結果の予測性」を向上させるための支援システムとして、米国政府の要請 (DODのスポンサーシップにて) に応じて開発されたものである。CMMには次の5つの成熟レベルが設定されている。

レベル1－Initial (初期)

レベル2－Repeatable (繰り返し可能)

レベル3－Defined (定義づけされている)

レベル4－Managed (管理されている)

レベル5－Optimizing (最適化されている)

CMMではレベル2に到達するためのキープロセスのひとつとして、ソフトウェア構成管理を位置づけている。

ISO9000、ISO9000-3は品質管理システムへの要求を規定する標準である。契約の際に供給者の設計・供給能力を示すこと(認定)に使われることを意図している。

一方、CMMは組織がソフトウェア・プロセスの成熟度レベルを改善する助け(自主的なプロセス改善や品質改善への取り組み)となることを意図している。(1)(2)(4)(5)

### 3.3 PDMとSCM

PDM (Product Data Management) は製品ごとの情報を集中管理することによって、形状変更と機能変更要求にすばやく対応しようとすることを目標にして、導入・使用されるようになった管理技法である。両者とも構成部品とそのバージョンを管理するという点では共通しているが、PDMは設計情報と構成管理に重点を置いているのに対して、SCMは開発プロセスとリビジョン管理に重点を置いている。

また、組み込み系ソフトウェアはハードウェアに組み込まれてしまうケースが多く、この場合は、PDMが管理する製品の部品群の中に含まれるソフトウェアを管理する技術となる。(PDM>SCM) (1)(2)

#### 4. センター方式によるソフトウェア構成管理の展開

日本の企業においては、SCMの導入率はせいぜい10%程度で十分に活用されているとはいえない。今後も構成管理システムを活用するだけでソフトウェア開発プロセスの改善に大きく貢献できる余地を残しているにも関わらず、何ゆえ、活用と普及が進まないのだろうか。<sup>(3)</sup>

多くのSCM活用事例はプロジェクト毎の活用事例であり、企業組織の一部でのまたは一時的な活用に終わってしまっている例が多い。組織全体でしかも継続的にソフトウェアの構成管理を徹底することが実現されている例があまり見受けられず、組織全体でのセンター方式による徹底した構成管理システムの活用の推進は十分といえない。

センター方式により、開発したソフトウェアは100%、企業の技術資産として、保管し、変更を管理し、再利用を促進すべく、企業の組織体制にしっかりと組み込まれる必要がある。

そのためには、現在のICT時代では情報戦略の経営的責任者であるCIO (Chief Information Officer) が不可欠であると同様に、全社のソフトウェア資産の重要性和複雑性と多量性を考えるとき、また、企業ノウハウの大半が、いまや情報システムに保管されていることを考えるとき、その安全性と健全性を戦略的に管理するCSO (chief system & software Officer) なるべき立場の責任者が必須となってくるであろう。

現在のCIOはSCMをまだ十分には認識していないようである。SCMは技術者集団のボトムアップ的な活動では、一時的なまたは一部分的な活動に終わってしまう。SCMは欧米ではソフトウェア開発の必須の管理技法として位置づけられている。「品質確保のためのもっとも重要な管理方法は何か」との問いには、欧米人の多くは「SCM」と答えるといわれる。日本では、SCMという解決方法があるのに「知らない?」「やっつけられない?」「やりたくない?」などボトムアップの発言が多く、同じ種類のトラブルが永遠に繰り返される。<sup>(3)</sup>

そのためには、なんと言っても、構成管理システムの活用ルールの設定、活用推進と支援を実践できる組織すなわちソフトウェア開発管理力強化グループ (SWG) の存在とその指導者が欠かすことの出来ない重要な要素のひとつになる。

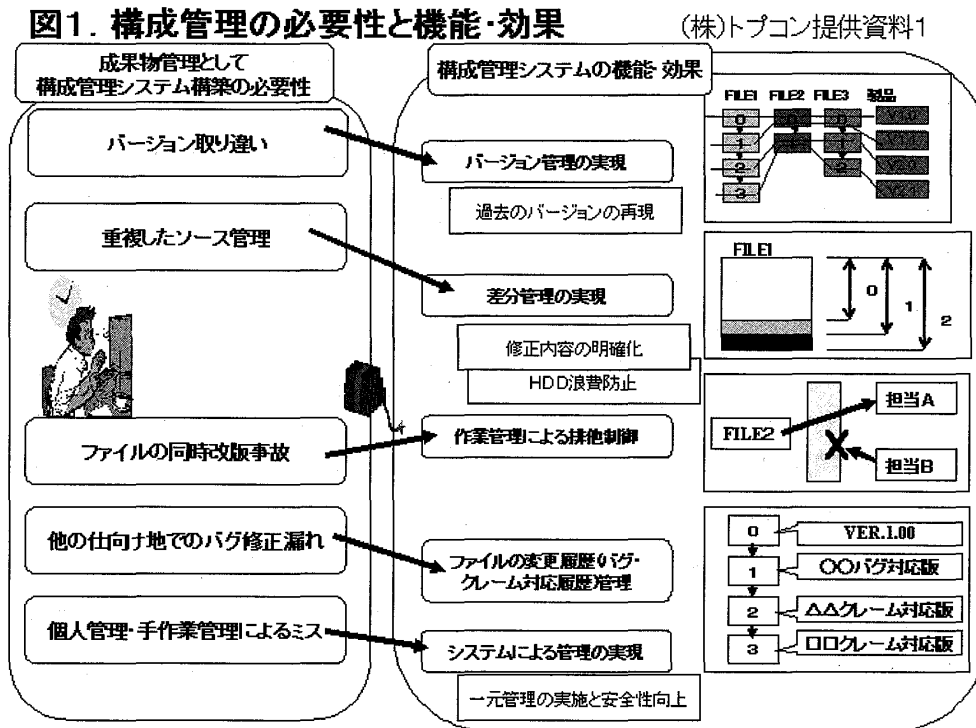
ここで、ソフトウェア構成管理システムを活用して、製品に組み込むすべてのソフトウェアを企業の技術資産管理として保管し、バージョン管理を実現し、再利用を促進することを目的とした光学機器メーカーのSWGグループの取り組み事例を示す。

#### 4.1 現状と課題

分野に限らず、多くのソフトウェア開発担当者や管理者が経験するように、この研究活動においても下記のような問題を抱えていた。(図1) (7)

1. ひとつのファイルを複数の開発担当者が同時に改版してしまい、ファイルの整合性が取れなくなってしまった。(ファイルの同時改版事故)
2. 思い込みや記憶違いで、誤ったバージョン(古いバージョン)のファイルをコピーして修正し、新しいバージョンとして適用してしまった。(バージョンの取り違い現象)
3. 必要なファイルであるにも関わらず、誤って消去してしまい、いつの間にか消失してしまった。(ファイルの誤消去事件)
4. 開発が進むうちに完成ソフトウェアのファイルが認識できなくなり、どのファイルが完成ファイルなのかわからなくなった。
5. バージョンごとに一式のファイルを保管し、重複したソース管理となってしまう、膨大な保管容量となってしまう。
6. 異なるバージョンで別々に同一バグを修正せざるを得なくなってしまい、修正ミスや修正漏れが発生してしまった。

などなどである。



## 4.2 システムの導入

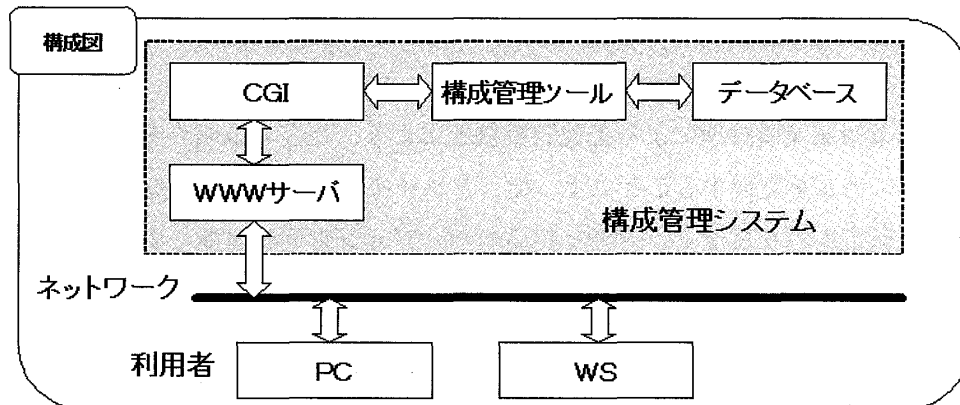
そこで、下記の機能を持つソフトウェア構成管理システムの導入を検討し、適用することによって、上記に記載した開発担当者や管理者が抱える諸問題の解決に対応することとした。(図1)

- (A) 作業領域環境管理による排他制御機能により、現在の改版作業者が明確になるシステムを実現することによって、ファイルの同時改版事故を防止できる。
- (B) ファイルのバージョン管理と製品のバージョン管理が可能なシステムにより、誤コピーによるバージョンの取り違いの阻止。
- (C) ファイル構成を可視化することにより、完成ソフトウェアのファイルがわからなくなることを防止できるシステム。
- (D) ファイルの差分管理を可能とし、バージョンごとに一式のファイルを持つことを必要としないシステム。
- (E) ひとつの変更内容がどのバージョンに反映すればよいのか容易になり、異なるバージョンで別々に同一修正をする必要がないシステム。
- (F) バックアップをセンターサーバで保証することにより、万が一でのファイル消失を防止できるシステム。

システムの導入と普及の展開に当たって研究活動を通じてえられた配慮すべき点は下記のとおりである。

- ・ 市販のソフトウェア構成管理システムを選定するに当たっては、上記に掲載した機能を持つシステムを選定する。
- ・ 調査の結果では、市販のシステムはソフトウェア開発のプロジェクト単位での導入を基本としている。今回の活動では、全社でのすべてのソフトウェア・プロダクトを一括して一元管理することを狙いとしているので、市販システムの導入だけでは実現できないことがわかる。
- ・ 市販システムと開発担当者のワークステーション間はイントラネットの形態でネットワーク化し、ワークステーションの配置と数量の柔軟性を持たせることが重要である。そのためのイントラネット用ユーザインタフェースの開発は欠かせないものであった(図2)
- ・ 企業の技術資産管理として、保管の安全性を保证する必要がある。そのためには、サーバの2重化とハードディスクの(例えばRAIDハードディスクシステム)多重化などによるシステムの信頼性・安全性を確保することが欠かせない要件となる。

図2. 構成管理システムの構成



(株)トプコン提供資料2

#### 4.3 システムの普及について

- ・ 普及に当たっては、まず第1段階としてモデル部門を選定し、開発部門とモデル部門（活用部門）で編成する共同グループで導入システムの試行を開始することである。
- ・ ここで、もっとも重要なことは、システムの活用によって得られる有効性をモデル部門に肌で実感してもらうことである。モデル部門がシステムを活用し始めることによって、そのあり難さを感動してもらえかがシステム導入の成功の鍵となる。
- ・ 幸いにもソースコード管理の安全性が高まることと個人管理にわずらわされる負担が軽減されるなどを実感し、モデル部門からの好評を得ている。
- ・ 第2段階としては、開発完了の製品に関しては全製品のソフトウェアを構成管理システムに保管し、安全と一元性を保証することである。
- ・ 第3段階としては、開発中の製品のソフトウェアを定期的に構成管理システムに保管し、万が一開発中のソフトウェアに消失や破損などの不祥事が発生しても、センターのサーバによるバックアップによって、最小限の損失にとどめることを可能にすることである。

#### 4.4 利用確認プロセスによるユーザの声の反映

- ・ システムの活用ケイパビリティをさらに向上させるためには、ユーザ部門の意見を収集し、要望にこたえていく体制（例えば、利用確認会）を維持することが重要である。当研究活動の結果では

- ・ モデル部での試行評価では、パスワード入力機能追加と担当範囲外の製品に関するソフトウェアへのアクセスを制限するなどの機能が要求され、システム改善を実施している。
- ・ さらに、その後ユーザ部門から、開発担当者、開発リーダー、管理者によって、利用できる機能範囲を設定してほしいという要求が引き続き提案され、システム改善を継続している。

#### 4.5 ソフトウェア開発管理力強化施策における SCM の位置づけ

なお、ソフトウェア開発管理力強化施策を検討するに当たっては、大きく次の4つの施策を展開させながら、実施した。

- 1、設計プロセスの改善と機械化
- 2、開発管理の強化
- 3、技術者育成
- 4、改善活動の定着化

「設計プロセスの改善と機械化」に関する施策としては例えば、

- ① ソースコードの静的分析
- ② プログラムの動的な分析
- ③ テストの自動化
- ④ オブジェクト指向技術の適用

など有効な設計技法の適用と CASE ツールの導入がこれに相当する。

「開発管理の強化」に関する施策としては

- ① プロジェクト管理手法の適用
- ② PERT 手法の適用
- ③ COCOMO やファンクションポイント法による見積もり手法の適用
- ④ デザインレビュー手法の適用などに加え
- ⑤ 「構成管理手法の適用」が欠かせない。

「技術者の育成」に関する施策としては

- ① 新しい言語や設計手法など先端的なソフトウェア技術の教育と同時に
- ② 開発過程に得られた成功事例と失敗事例に関する技術移転の機会を持つことが重要となる。

「改善活動の定着化」に関しては

- ① 組織的にソフトウェア改善グループを定着させ、継続的に改善プロセスが遂行できる体制が必要である

- ② と同時に改善が見える目標値としてのメトリックス (指標) を設置することが重要になる。

本論文の主テーマである「構成管理手法の適用」は開発管理の強化施策として取り上げている。

## 5. ソフトウェア構成管理システムによる開発プロセスの分析と改善手法についての提言

前節の結果、

- ・ 日本の企業においては、SCMの導入率はせいぜい10%程度で十分に活用されているとはいえない。
- ・ 多くの事例は、プロジェクト毎の活用事例であり、企業組織の一部でしかも一時的な活用に終わってしまう例が多い。
- ・ センター方式により、開発したソフトウェアは100%、企業の技術資産として、保管し、変更を管理し、再利用を促進すべく、企業の組織体制にきっちり組み込まれる必要がある。

ことがいえる。すべての企業は、ソフトウェア構成管理システムを導入・構築し、ソフトウェアの資産管理と変更管理を徹底し、ソフトウェア資産の安全性、信頼性、有効性を高め、健全性を保証することが不可欠である。

と同時に、ソフトウェア開発プロセスの分析とその更なる改善としての新たな取り組みとして、下記の構想を提言する。

構成管理システムを単に、ソフトウェア資産の保管と、変更管理の活用にとどめるのではなく、構成管理を必要とするソフトウェア開発プロセス全範囲において、構成管理システムに管理されるデータならびに構成管理をアクセスするプロセス〔過程〕自体の情報を意図的に取り入れ、効率的に活用し、ソフトウェア開発プロセス自体を目視化し、分析可能にすることによって、ソフトウェア開発プロセスの改善につながる手法を提言するものである。

従来の開発者による定性的なまたは不正確な (残念ながら) 情報による分析から成果物情報とプロセス情報の累積を基に、正確なかつ静的な (スタティックな) または動的な (ダイナミックな) 情報による適確な分析を可能にするものである。

そして、ソフトウェア開発にかかわる品質改善、工期の短縮、開発コストの低減に貢献できることを狙いとするものである。1999年にこの構想を提言しはじめてから今日まで、具体的なシステム構想・設計・開発・活用推進活動を継続し、いくつかの成果を得ている。

(8)(9)(10)

## 5.1 基本構想

基本構想としては次の3つのプロセス分析と改善の仕組みを作ることを狙いとする。

- ① 成果物を管理する構成管理システムを発展させ、これにプロセス(ソフトウェア開発過程)に関する情報を加え、開発プロセスを可視化し、分析できる仕組みを作る。
- ② 開発プロセスの可視化と現状分析により、開発過程のプロセスに対して、まず、動的な生産状況の現状把握から着手し、順次、ソフトウェアまたはシステムの品質特性の把握(品質管理)、工数実績の把握(工数管理)、工程の進捗状況の把握(工程管理)へと段階的に改善を強化する仕組みを作る。
- ③ また、これらの現状分析データを中長期的に蓄積することによって、製品別、部門別の統計的なデータを分析する仕組みを作り、製品別、組織別の基準値を見出せることを可能とし、また製品別、部門別の問題点を分析すると同時に全組織での問題点を把握する仕組みを作る。

## 5.2 開発プロセス分析システムの仕組み

構成管理システムと開発プロセス分析システムの関連を図3に示す。

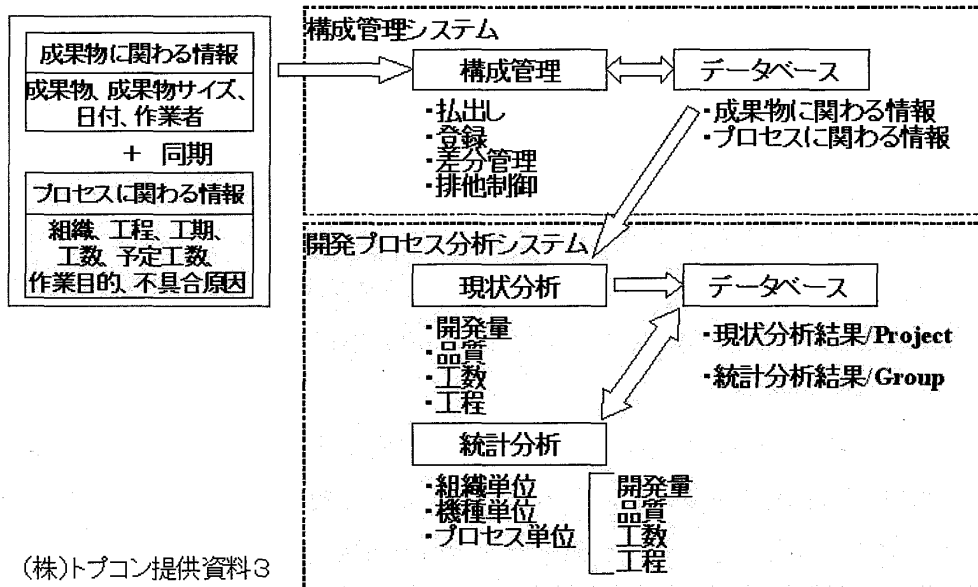
構成管理システムの対象はシステム開発プロセスのシステム設計、ソフトウェア設計、プログラム設計、検査に関する設計情報、ならびにソースコードとなる。

また、プロセス情報としては、作業目的、変更要因、開発工程、日時、担当者名、所属部署名、工数、予定工期、予定工数などプロセス分析に必要なプロセスデータで構成される。

これらのプロセス情報を入力する機能は市販の構成管理システムではまだ標準化されておらず機能も十分ではない。したがって、この研究では、wwwを利用したカスタマイズの容易さと使いやすさを考慮したデータマネージャーを開発し適用している。<sup>(5)(6)</sup>

構成管理システムはプログラムやドキュメントなどの成果物を登録し、その変更履歴を管理する。格納される情報は成果物、成果物サイズ、登録者名、登録日時などであり、これを成果物情報という。

図3. 開発プロセス分析システム機能図



今回提言しているプロセス分析機能を持つ構成管理システムは現時点では市販の構成管理システムの外付けとしてのソフトウェアをユーザごとに開発していかなければならない不便さが残っている。開発のアプローチとしては二通り考えられる。

ひとつは、プロセス分析機能をエディターの役割としてエディターへの組み込み型である。もうひとつはプロセス分析機能を構成管理システムの機能として位置づける構成管理システムへの組み込み型が考えられる。

分析の詳細化とシステムの軽さからいえばプロセス分析機能をエディターの機能に取り込みながら開発するほうがユーザ視点としては喜ばしい。

しかし、今後、世界的な規模でのソフトウェア分散開発の形態が進展してゆく状況下では、構成管理システムの機能は必須であり、開発環境の主体とならざるを得ないことを考え、後者の構成管理システムへの組み込み型を適用した。

ただし、構成管理システムを日毎にまたは修正ごとにチェックインすることを考慮すると、相当量のデータ通信量となるので、チェックイン/チェックアウトの所要時間の課題が残る。通信網の高速化に配慮すると同時にファイルの構成要素にも配慮し、無駄なファイルの転送を反復しないようにも工夫する必要がある。

理想的には設計プロセス（従来、設計書が個人の机の上や引き出しにある時のように）ではエディターへの組み込み型で開発プロセスを分析し、組織的な認証の元でのリビジョンアップやバージョンアップ時のプロセス解析は構成管理システムへの組み込み型を適用するのがよい。

市販のエディターや構成管理システムにはリビジョン記録を残すなどのプロセス管理機能が徐々に充実しつつあるが、いずれもツール主体型である。われわれのアプローチはソフトウェア開発プロセスの分析と改善機能主体型であるところにアプローチの根源的な違いが見られる。

SCMはプロセス管理、リレーション管理、バージョン管理の3つの管理層体系からなる。そのうち、この論文の論点はプロセス管理にもっとも関わるものであるが、観点の相違は一方がベースラインの管理が中心なのに対して、本論文は、成果物の変化を量的または質的に分析することによるプロセスの分析と改善に主な狙いをおくものである。

また、プロセス分析機能を明確にし、向上させるためにはひとつのチェックイン／チェックアウトでは一種類の作業内容が対応するのがよい。例えばクレームに対する修正作業であるならば、ひとつのクレームに対する作業ごとにチェックイン／チェックアウトを行うことが有効であり、複数のクレーム修正に対してひとつのチェックイン／チェックアウトを行うことは好ましいものではない。当研究での事例では、開発、テスト、検査、仕様変更、クレーム対応など標準的なプロセスを組織全体で標準化し、ポップアップメニューで選択するなどプロセスの標準化と操作の標準化に配慮している。使い慣れてゆくにしながら、このような活用に関するルールは成熟していくものと思われる。

### 5.3 段階的な成熟レベルアップへのアプローチ

開発プロセスを分析するためには、構成管理システムが持つ本来の成果物情報に加え、分析に必要なプロセス情報を加味しなければならない。しかし一度にすべてのプロセス情報を入力することを求めることは、開発技術者にとっても管理者にとっても大きな負担となる。そこで、この研究では、開発プロセスとして下記のように段階的に分析システムを発展させていく成熟レベルアップのアプローチを考案した。

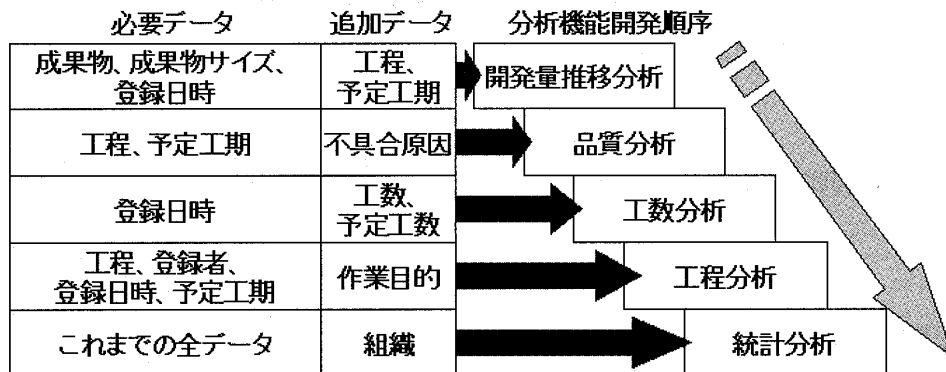
構成管理システムが持つ本来の成果物情報に加えて、各機能を実現するために必要にして最小限のプロセス情報を追加してゆくことによって、下記の分析機能を段階的に実現しながら提供してゆくことが可能になる。(図4)

- ① 各プロジェクトのプロセス毎の動的な生産状況のモニタにより、プロセスの状況を把握する。例えば
  - (1) 開発着手の遅れ期間の状況。
  - (2) 開発が進んでいるのか。停滞しているのか。
  - (3) 開発量や修正量が増え続け、予定通りに開発が終わりそうにない。
  - (4) リリース後も安定していない。
  - (5) 予定通りの開発量が出来ているのか。

- (6) 累積開発量と開発期間による静的生産性の分析。
- ② 変更内容や修正量などの分析による品質分析を可能とする。例えば
  - (1) 開発プロセスにおいて開発量が安定しない。変動が激しい。仕様の見直しや追加が繰り返されている。
  - (2) テスト期間に入ってからの変更量が多い。(バグが多い)
  - (3) テスト期間での変更量が安定しない。(テストが収束しない)
  - (4) 出荷後の修正量が多い。(出荷後のバグ修正が多い)
  - (5) 特定のモジュールに修正が集中している。
- ③ 変更日時による工数推定や実績工数の入力により、コスト分析を可能とする。例えば
  - (1) 開発プロセスにおける生産性(ステップ/日)が基準値よりも低い/高い。
  - (2) テストプロセスにおける工数比(修正ステップ/総ステップ)が基準値よりも低い/高い。
  - (3) 保守プロセスにおける工数比(修正ステップ/総ステップ)が基準値よりも低い/高い。
  - (4) 各プロセスの工数比率の適正を分析できる。
- ④ 予定工期と実績工期の分析により工程(工期)分析を可能とする。例えば、
  - (1) システム設計/プログラム設計/テスト/保守プロセス等の工程(工期)分析により進捗状況とその要因の把握が可能となる。
  - (2) システム設計/プログラム設計/テスト/保守プロセスの工期比率の適正性分析を可能とする。
- ⑤ 現状データを積み上げ統計分析することにより、企業全体、部門、機種、工程別の基準値を得ることが可能となる。たとえば
  - (1) 開発期間に対する開発量の統計分析により、生産性の基準値を得ることができる。これにより、より精度の高い開発コストの予測や開発工程計画の立案が可能となる。
  - (2) 品質阻害要因を分析し、品質改善活動の重点項目を設定することができる。
  - (3) 工程ごとの工数と期間を統計分析することにより、基準となるマンパワー曲線を得ることができる。

図4. 段階的な成熟レベルアップへのアプローチ

追加するデータを絞り込み、入力する仕組みを設ける



各機能を利用するのに必要な情報のみ追加して機能提供していく

(株)トプコン提供資料4

#### 5.4 実施事例

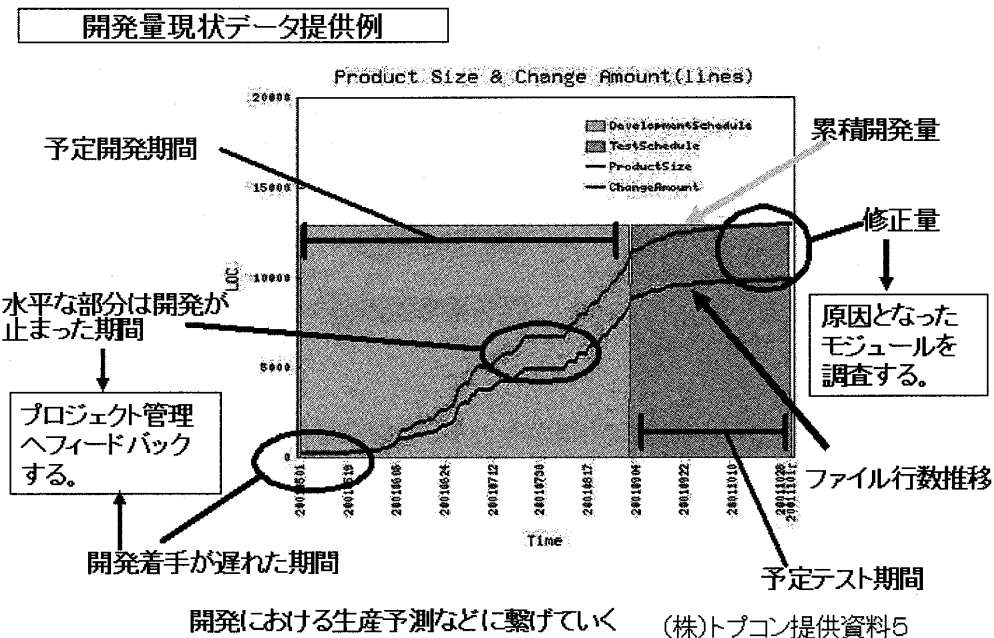
上記までの基本構想に基づいて、開発されたセンター方式による構成管理システムと開発プロセスの分析システムの実施例を示す。

・センター方式による構成管理システムの導入と実施については

- ① 具体的にソフトウェア開発管理力強化グループを技術企画部門に設立。  
4.5で示したソフトウェア開発管理力強化策を推進。その中で「開発管理の強化」施策の柱として本研究テーマである「構成管理の適用」を実施した。
- ② 市販ツールの選定と導入。ただし、市販ツールはプロジェクト単位での活用を基本とするので、
- ③ センター方式の構成管理を実現するためのイントラネット方式のデータマネージャーを開発。(図2)
- ④ ソフトウェア資産管理の信頼性を確保するための2重化システム、Raidハードディスクシステム、DATによるデイリーバックアップシステムの実現。
- ⑤ ほぼ100%のソフトウェア製品の保管管理。
- ⑥ 管理面では安全性の向上、個人管理の不要、一元管理、過去のバージョンの確実な再現が可能となったなどの評価を得ている。
- ⑦ 開発面では、開発中のファイル破損時のバックアップ、同時改版事故の防止、差分管理による開発履歴の明確化などが高く評価されている。

- ・ 構成管理システムによる開発プロセス分析システムについての実施事例の一例として、「開発量推移分析」を示す。出力例を図5に示す。

図5. 開発プロセスの現状分析の例



成果物のサイズと開発量のサイズが開発プロセスの進捗に応じて、きめ細かく観察される。

たとえば

- ① 修正等の作業が差分として見て取れる。
- ② 開発プロセスでは仕様の変更や設計のやり直しが多いかどうかを観察できる。
- ③ テストプロセスではバグの修正が観察できる。
- ④ 保守プロセスではクレームの頻度と修正時期や修正量が観察できる。

ことが期待できる。

- (a) 具体的な事例として図5の例で見られるような開発プロセスでの停滞状況 (stagnation) が観察された。これは、開発プロセス中に生じた仕様の変更のため、一時開発を停止せざるを得ないことを示していた。
- (b) 急激なソースコードの増大 (Rapid Increasing) が観察された。これは多くの場合、すでに開発されソースコードの再利用が実施されたことを示す。どの程度、ソースコードの再利用率が達成されているか観測できると同時に、再利用ソースコードの修正割合も観測することができる。また、再利用率と開発工数との相関関係を観察することによってより適切な開発工数の見積もり手法も確立できる。
- (c) 開発ソースコード量とプロダクトコード量との差の急激な変化が観測された。たとえ

ば、

- ・ 開発プロセスにおいては仕様の変更要求が入ることによる急激な変化が観察される場合、
- ・ デバッグプロセスやテスト（検査）プロセスにおけるソースコードの修正作業が頻繁におこなわれる場合、
- ・ 保守プロセスにおけるクレーム対応における修正が頻繁に行われる場合
- ・ または、テスト期間に投入されていたテスト用のコードを最終製品にするとときに除去した場合などの現象が観測されている。

この開発プロセス分析システムを活用することにより、また、さらにこのシステムの機能を充実させていくことにより、きめ細かなソフトウェア開発プロセスの状況把握が可能になることが期待できる。

そのためには、推進部門の熱意あるオリエンテーションによって、「見せる努力と見る努力」を積み重ね、継続していくことが、今後最も重要である。

有効なシステムを導入し、カスタマイズし、時には必要にして不可欠なソフトウェアを開発し、ソフトウェアの開発プロセスに発生するさまざまな現象をたとえソフトウェア開発になじみがない管理者や部門要員に対してもわかり易く見せてゆく努力が不可欠である。

と同時に、ソフトウェア開発をソフトウェア技術者だけに任せる無責任な組織機能から一歩踏み込んで、開発部門だけでなくそれを取り巻くすべての部門がソフトウェア開発の実態を見る努力と支援をすることがさらに大切となる。

## 6. まとめ

ソフトウェア開発プロジェクトまたはソフトウェア開発プロセスにおいてプロセスの実態を見る努力と見せる仕組みを築き上げていく努力は管理面においても技術面においても大変重要で意義あることである。

しかしながら、ソフトウェア工学の研究とその実践が求められた1960年代の「ソフトウェア危機」の時代から今日まで、実データに基づいた見せる努力と見る努力が十分にかつ具体的にこなされてきたかという点はまだ満足できるものではない。

オフライン的またはバッチ的にデータを収集してデータ分析するケースはままだ見受けられるにしても、特に、オンラインで構成管理の仕組みやエディターの仕組みを利活用し、開発プロセスの現状把握と分析を、しかも組織的に取り組み、実現しているケースはまだ少ないといえる。

今後、オンラインでのプロセスデータの分析機能による、プロジェクト推進中でのコスト管理、品質管理、工程管理などへのフィードバックの仕組みやいくつかのプロジェクト

結果における統計的データの分析結果を、次期ソフトウェア開発プロジェクトへのフィードフォワード的な管理へ活用する仕組みをさらに追及すべきと思える。尽力していきたい。

なお、この研究をまとめるにあたり、(株)トプコン技術品質グループ設計合理化推進部、特に杉山昭洋氏、大木雅彦氏には多大なご協力をいただいた。深く感謝申し上げます。

## 参考文献

- 1) 徳田弘昭著『ソフトウェア構成管理～SCMでソフトウェア開発の革新を～』ソフト・リサーチ・センター、1999年
- 2) 徳田弘昭著『ソフトウェア構成管理ハンドブック～ソフトウェア資産管理のキイテクノロジー～』ソフト・リサーチ・センター、2004年
- 3) (社)情報サービス産業協会『SCM (ソフトウェア構成管理) に関する調査研究報告』、平成9年
- 4) 東陽テクニカ『CASEフェスタ'99ソフトウェア構成管理で工程遅延を防止』、1999年
- 5) 東陽テクニカ『CASEフェスタ'01 ソフトウェア構成管理の高度な実践方法』2001年
- 6) 吉崎・会沢他『ソフトウェア構成管理プロセス改善一手法(1)』情報処理学会第60回全国大会、2000.
- 7) 吉崎・大木他『ソフトウェア構成管理プロセス改善一手法(2)』情報処理学会第60回全国大会、2000.
- 8) 吉崎・大木他『構成管理データに基づく開発プロセス分析の一手法』情報処理学会第64回全国大会、2002
- 9) 吉崎・会沢他『構成管理システムを活用したソフトウェア開発プロセス改善への取り組み事例』第66回情報処理学会全国大会、2003
- 10) Aizawa, 『An Approach to Improvement of Software Project Management by Utilizing Data from SCM System』3rd World Congress for Software Quality、2005年