メトリクスによるソフトウェア品質管理法とその評価

三宅武司[†] 竹中市郎[††]
古山恒夫[†] 中川豊[†††]

上流工程で品質を計測および評価する方法として，品質メトリクスを用いる方法 (SQM) が提案され，その有効性がすでに報告されている．しかし，品質メトリクスの実施例がまだ少なく，品質要求定義やメトリクスの導出など，作業の詳細が明確になっていないため，品質評価方法として適用するには再現性に問題があった．本論文では，品質メトリクスによる品質評価法の詳細を定め，それを実際の開発プロジェクトに適用・分析した結果を報告する．具体的には，以下の(1)から(3)に示す方法が有効であることを明らかにした．(1)品質メトリクスの適用に際して，ユーザの品質要求分析を的確に行うために，アンケート，一対比較法および文章分析法による品質要求分析法を併用する．(2)メトリクス導出のガイドとして，過去に使用したメトリクスをSub. criterion，および工程ごとに分類整理したデータベースを用意する．これにより，メトリクス導出が容易となるだけではなく，メトリクス導出の抜けと偏りを防ぐことができる．(3)メトリクスごとに品質向上の効果とその要因を分析することにより，各メトリクスの有効性の評価と設計法へのフィードバックを行う．さらに，品質メトリクスの適用により，品質向上と開発工数削減に有効という直接的な効果だけではなく，開発担当者のモラルの向上という間接的効果が得られることも示している．

Software Quality Control Method Based on Software Quality Metrics and Its Effectiveness

Takeshi Miyake,[†] Ichiro Takenaka,[††]
Tsuneo Furuyama[†] and Yutaka Nakagawa[†††]

In response to the growing demand for software quality, various software quality measurement models based on Software Quality Metrics (SQM) have recently been proposed and their effectiveness have been reported. In most cases, however, the details of such activities as quality requirements definition and metrics selection, have not been clarified. Therefore, people who would like to apply one to their projects find it difficult to actually apply it and obtain good results as the reports. This paper describes details of SQM methodology activities, in which quality requirements are defined, metrics to be measured are selected, metrics values are measured in actual development projects and effectiveness of the methodology was evaluated. The results are summarized as follows. (1) Using such methods together is effective to clarify quality requirements as; a questionnaire method, a method of comparing all factors one by one, and a method of counting a frequency of word occurrence in terms of quality factors in the requirement document. (2) Metrics database which stores metrics gathered in the previous development projects was useful for supporting metrics selection. This metrics database provide SQM users with metrics appropriate to evaluate required quality factors. (3) The effects of each metric on quality improvement were evaluated and fed back to the design and documentation rules. Also shown is that the SQM methodology is effective not only to improve the quality and reduce development efforts, but also to improve developers' morales.

1. はじめに

ソフトウェア品質の向上には，ソフトウェア要求仕様書がユーザの要求条件を正しく反映し，その後の成果物が規定どおり正確に作成されていることが重要である．各作業が正確に実行されたか否かは，成果物の品質の良否となって表れる．ソフトウェアの品質を定量的に把握するには，これら成果物の品質を定量的に

† NTT ソフトウェア研究所
NTT Software Laboratories
†† 久留米工業大学
Kurume Institute of Technology
††† NTT 情報システム本部
NTT Information Systems Headquarters

測る尺度と計測方法が必要となる．ソフトウェア品質メトリクス (SQM : Software Quality Metrics)*は，この要求に応えるべく開発された手法であり，従来より行われている試験工程における検出フォールト数に基づく品質管理だけではなく，仕様書，設計書等上流工程の成果物，および最終製品の品質を定量的に評価するための，評価尺度と計測手法を統合したものである．

ソフトウェア品質メトリクスの研究は，米国を中心に1960年代から始まった[1]．1976年にBoehm[2]が最初に階層的体系を発表し，その後，1977年には McCall が米国空軍 RADC で開発した工程別の詳細なメトリクスを含む品質評価体系を発表した[9]．これをもとに，より容易に適用可能な体系としたのが Murine[3] の SQAM である．

わが国でもこれを受けて1985年に日本電気が SQMAT を開発し，実際のソフトウェア開発に適用し，品質および生産性向上に大きな効果があることを示している[4]．また日本 IBM が 1985 年に発表した SQUALAS は，ビジネスアプリケーションに限定して，日本で開発されたハードウェアの品質管理手法である「品質機能展開」[5]をソフトウェアに適用したシステムである．

1980 年代後半から，これらの研究成果を基礎に，1つの統一的な品質評価の枠組みを構築しようとする活動が，IEEE[6]や ISO[7),10),11]で展開されてきた．さらに，これらを基礎に「品質機能展開」の考え方を加味した新しい定量化の方法が，ユーザ，技術者および管理者の3つの視点から品質を評価する方法として提案されている[8]．しかしながら，この枠組みを実際のソフトウェア開発に適用した例はなく，実用上の評価が待たれていた．これまでのメトリクス研究においては，ソフトウェアメトリクスを用いて品質の管理を行うことにより品質・生産性が向上することを，実際のプロジェクトで実証する方法が取られている．しかしながら多数のメトリクスの中から，具体的にメトリクスを選定，測定評価，さらにはメトリクスの妥当性の評価のための手順を，明確に示したものは少ない．また，同一のメトリクスを同一の開発チームで繰り返して使った場合の，学習効果による品質向上について評価したものは見あたらない．本論文では，これらの点に着目し，メトリクスによる品質評価の枠組みに基づく品質管理を設計工程に適用することにより，ソフトウェア製品の品質と生産性が大幅に向上すること，さらに，同一のメトリクスを同一の開発チームに繰り返して適用した場合の効果を示している．

2. ソフトウェア品質定量化の新しい枠組み

ソフトウェアの開発は，「ユーザ」からの要求に基づき，「開発技術者」によって実行される．しかし，通常，開発技術者（以後技術者と表記）とユーザの間には，開発の進行を管理したりユーザとの間の調整を行う「管理者」が存在する．これら3者とも，ソフトウェア品質に対し，それぞれ固有の評価尺度を持っていると考えるのが自然である．従って，ソフトウェアの品質評価体系も，これら3者の立場を考慮したものが望ましい．このような認識にたって，鳥居らは，IEEE および ISO の試案に品質機能展開の考え方を加味した新しい定量化の方法として，ユーザ，開発技術者および管理者（あるいは調整者）の3つの視点から品質を評価する方法を提案した[8),12)~14)]．新しい枠組みは文献8)に詳述されているが，本論文の理解を助けるために，その概要を2.1節，2.2節および2.3節に示す．

2.1 評価の視点

ユーザ視点とは，ソフトウェア製品を使うものの立場から評価するものである．ユーザ視点でとらえた製品としてのソフトウェアの品質評価尺度を Factor, Sub. factor, Product metrics の3階層で定義する*．これは ISO における特性，副特性，メトリクスにそれぞれ対応している．一方，技術者視点とは，ユーザの品質要求を実現するための開発プロセスを評価するものである．つまり，品質の実現に有効な開発技術の選択とその評価を行うための尺度であり，開発技術の分類を表す Criterion，より詳細化した Sub. criteri-

* (1) JIS X 0129 では metric および metrics を測定法と訳している．しかし，本論文ではソフトウェア工学の慣例に従って，測定法をメトリクスと表記する．
(2) ソフトウェア品質メトリクスは，完成したソフトウェア製品の良否を使用する立場から評価するための「プロダクトメトリクス」と，ソフトウェア開発の各工程の作業が正確に実行されているか否かを評価する「プロセスメトリクス」に大別することができる．通常用いられる代表的なものを付録に例示する．プロダクトメトリクスは製品の品質を直接評価するためのものである点に着目して，ダイレクトメトリクスと呼ぶこともある．これに対して，プロセスメトリクスは最終製品の品質を予測するためのものであるため，プレディクティブ (Predictive) メトリクスと呼ぶこともある．

* 文献8)では Sub. factor, Sub. criterion をそれぞれ Ex. factor, Ex. criterion (Ex. : Extended の略) と表記しているが，本論文では，階層の下位概念を表す用語として，ISO および JIS 規格に用いられている Sub を用いる．

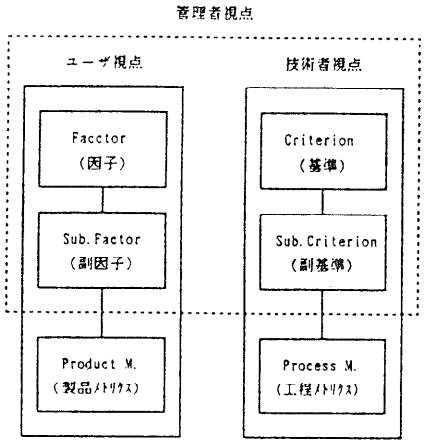


図 1 品質定量化の新しい枠組み
Fig. 1 A new quality framework.

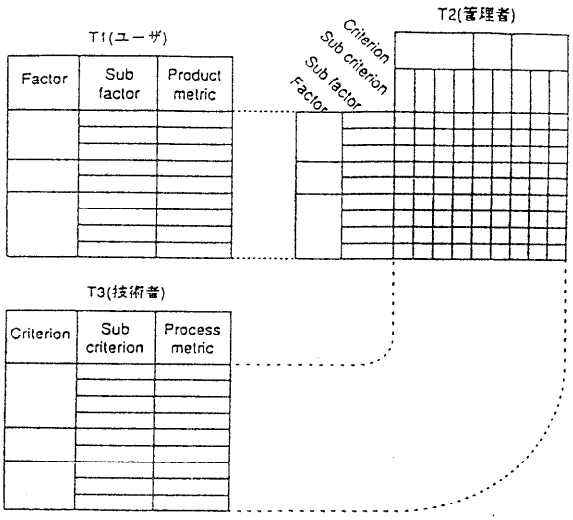


図 2 品質表 T1, T2, T3 文献4)
Fig. 2 Software quality table T1, T2 and T3.

on, ソフトウェアの開発工程に関するデータに基づく Process metrics の3階層で定義する.

実際のソフトウェア開発においては, コスト, 納期, 要員等の多くの制約条件が存在するため, 技術者がユーザ視点での要求品質を把握して, 品質の実現に有効な開発技術を選択し, 要求されたものを実現するのは難しい. そこで第3の視点として, 技術者に代わってこれら制約条件とユーザ要求を考慮・調整して, 実際に利用可能な技術を選択する管理者の視点が必要となる. 具体的には, Sub. factor と Sub. criterion の対応関係を規定するものを管理者視点と定義する. これらの関係を**図1**に示す.

2.2 品 質 表

それぞれの視点における評価尺度の階層構成をT1, T2, T3の3つの品質表で表現する(**図2**). T1表は品質特性展開表と呼ばれるユーザ視点のメトリクスを Factor, Sub. factor, Product metrics の3レベルの階層構造に分類したものである. T3 表は Criterion 展開表と呼ばれ技術者視点のメトリクスを Criterion, Sub. criterion, Product metrics の3段階に分類したものである. T2 表は品質対応表と呼ばれ, 管理者視点から見た Sub. factor と Sub. criterion の対応関係を示すものである. 対応関係は Sub. criterion の実施によって実現される Sub. factor との交点に○あるいは関係の強度を表す数値 (例えば1～5) をつけることによって表す.

2.3 枠組みの適用手順

実際の評価では, まず評価対象のプロジェクトからの要求品質に基づいて汎用的な品質表 T1, T2, T3 から必要な項目を抜き出して, 当該プロジェクトのための個別の品質表 t1, t2, t3 を作成する. これらの個別品質表を用いて具体的なメトリクスを決定し, 計測を行い品質の評価を行う.

(1) T1 表から t1 表の導出

品質管理担当は, あらかじめ使用すべきメトリクスおよび合否判定基準を設定する. このため, まずユーザ視点での Factor, Sub. factor, Product metrics のすべてを含んだ T1表から, 対象としている製品に要求される品質特性に合わせて必要な部分 (t1表) を抽出する. これは, 限られたリソースの範囲内で重点的に品質の管理を行い, 最大の効果を出すためである.

(2) T2 表による技術者視点のマッピング

重点化された品質特性を実現するために, 開発工程で測定すべきプロセスメトリクスを求める. このために**表1**に示す T2 表により, 求められた品質特性に対応する技術を表す Sub. criterion を求める. ユーザと技術者の間に立つ管理者は, 開発納期・工数等を勘案して T2 表の○のうち重要度の低いものは削除する等の調整を行って t2 表を得る.

(3) T3 表によるメトリクスの決定

T3 表は技術者の立場で実施すべき設計上の要点が, 階層的に分類され最終的にはプロセスメトリクスとして定義されている. 前項で求めた Sub. criterion の下に含まれるメトリクスが求めるものである. このようにして T1, T2, T3 表を順次適用することにより測定すべきメトリクスを求めることができる.

表 1 T2 表
Table 1 Table of T2.

Factor	Sub. Factor \ Criterion / Sub. Criterion	設計技術 階層化	構造化	モジュール化	ドキュメント	再利用	標準化	故障回避	故障診断	暗号化	HCI
機能性	合目的性	○									
	正確性		○	○	○						
	接続性					○	○				
	標準適合性	○				○					
	セキュリティ									○	
信頼性	成熟性	○	○	○	○	○	○				
	障害許容性							○	○		
	回復性								○		
使用性	理解性					○	○				○
	習得性					○	○				○
	運用性										○
効率性	実行効率性	○				○					
	資源効率性	○				○					
保守性	解析性	○	○	○	○		○		○		
	変更性	○	○	○	○		○				
	試験性	○	○	○							
	安定性					○		○	○		
移植性	環境適応性		○	○							
	移植作業性	○	○	○	○	○	○				
	規格準拠性	○				○	○				
	置換性	○	○	○		○					

（注）○印はSub. FactorとSub. Criterionの関係が強いことを表す

3. 新しい枠組みの適用

3.1 適用したシステムの特徴

社団法人トロン協会において標準化が行われているCTRON仕様に対する検定を行うシステム「CTRONインタフェース検定システム」[15]【注）TRON：The Realtime Operating System Nucleus, CTRON：Central and Communication TRON の略である】の新規開発時と，同プロジェクトの機能拡張時の2回にわたり，本枠組みに基づいたプロセスメトリクス（注：以下 SQM と表記する）を適用した[16),17)]．なお，2回とも開発担当者および品質管理担当者に変更はない．検定システムの性格から高品質が要求されるため，特に上流工程での品質の作り込みが重要であると考え，設計段階すなわち機能設計（FD）工程と詳細設計（DD）工程に重点をおいた品質評価を行った．CTRON 検定システムの諸元と SQM 適用規模は以下のとおりである．

開発言語：C言語
新規開発規模＝69.4 ks
（うち SQM 適用規模＝17.7 ks）
機能拡張規模＝31.3 ks
（うち SQM 適用規模＝12.3 ks）
合計＝100.7 ks
（総適用規模＝30.0 ks）

CTRON インタフェース検定システム[15]は大きく2つの部分で構成される．（1）OSのシステムコールを実行するプログラムの集合であるテストスイートと，（2）これらの実行のスケジューリング等を行う検定管理プログラムである．前者のプログラム構造はシステムコールの実行，結果の記録の繰り返しであり論理は単純である．一方，後者はシステムの状況に応じて実行の流れを変えたり，テスト実行のためのリソースの管理，テスト結果の編集等を行うため，論理も複雑でありシステム全体の品質に与える影響が大きい．このため SGM は後者を対象に適用した．

3.2 要求品質特性とメトリクスの選定

（1） 要求品質特性の決定

要求品質がシステムの要求仕様書に明確に記述されている場合は，その要求にしたがい品質管理を実施すればよいが，明記されていない場合には，何らかの方法で要求品質を明確にしなければならない．本実験では，品質要求を正確に把握するため次の3つの方法を併用して品質特性*の重要度を抽出した．

- アンケート法：ユーザまたは，その分野の専門家にアンケートを行う方法
- 一対比較法：品質特性を2つずつ取り出し，一対比較法で重要度を求める方法
- 文書分析法：要求仕様書に用いられている単語を分析し，品質関連単語出現頻度により品質の重要度を推定する方法[3)]

重要度の抽出結果を**図3**に示す．アンケート法では信頼性，使いやすさ，機能性の順に，一対比較法では信頼性，機能性，使いやすさ（使用性）の順に重要であるという結果を得た．また，文書分析法では機能性，信頼性，使いやすさの順に重要であるとの結果を得た．いずれの場合でも上位3位までは，同じ品質特性が選ばれている．今回はこれらの特性と下位3特性とに差が

* 文献8）ではユーザ視点の品質を規定するものを Factor あるいは Sub. factor と規定している．しかし本論文では，以後これらに相当する用語として，ISO および JIS 規格で規定された品質特性および副特性を用いる．

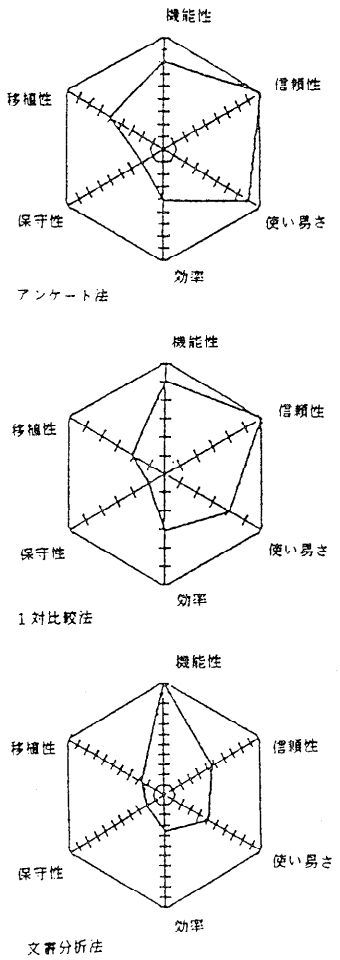


図 3 要求品質の重要度
Fig. 3 Priorities required quality factors defined by user.

あることから，これらを要求品質特性として選定した．

アンケート法と一対比較法の結果の違いは，アンケート法が副品質特性ごとに分析したのに対し，一対比較法では品質特性ごとに重要度を抽出したために生じたと考えられる．つまり，アンケート法では合目的性の重要度は高く抽出されたものの，接続性とセキュリティが低く抽出され，機能全体としては使用性にくらべ低く抽出された．一方，一対比較法では機能性は合目的性のイメージをもって重要度の抽出をされたため，使用性にくらべ高く抽出された．また，文書分析法では要求仕様書を分析対象としたが，一般に要求仕様書は実現されるべき機能について記述されているため，機能性に関する記述が突出していたと考えられる．

一般に，アンケート法や一対比較法は担当者の主観が入ったり，システムの経験がないと重要性が判断しにくいという問題がある．一方，文書分析法は客観的に判断できる方法であるが，品質要求が機能性に偏る可能性が高い．このため，文書分析法をアンケート法あるいは一対比較法と組み合わせ，互いに弱い部分を補完し合うことが有効であると考えられる．

（2） メトリクスの決定

現時点では，新しい枠組みで使えるような広く合意された品質表（T1, T2, T3）は得られていない．そこでメトリクスの導出を支援するため，メトリクスデータベースを構築した．まず過去の類似ソフトウェアで使用したメトリクスに加え，これらをもとにソフトウェアの特性やユーザ要求に応じて適宜修正したり新たに追加したメトリクスをデータベースに登録した．データベースの内容は，**表 2** に示すように，当該メトリクスを適用する工程，メトリクスが属する Criterion と Sub. criterion，メトリクスの定義，視点，計測対象成果物，分野，計測値の範囲が含まれる．文献等で報告のあった SQM の事例等から得られたものも含め，要求定義から試験工程までの全工程を対象に約 1000 個のプロセスメトリクスを各 Sub. criterion ごとに分類しデータベース化した．

CTRON 検定システムで求められた機能性，信頼性，使用性を実現するため Sub. criterion として，階層化設計，構造化，モジュール化，ドキュメント参照容易化技術，標準化，故障回避技術を選定し，その中から今回の開発に適したものとして FD 工程用として 21 個，DD 工程用として 18 個のプロセスメメリクスを求めた（**表 3**，**表 4**）．

ユーザの品質要求に対応する Sub. criterion として，T2 表から再利用，暗号化，故障診断および HCI (Human Computer Interface) 技術等も得られたが，対象とする検定システムは類似システムもなく再利用は困難であり，また広く一般ユーザに使われるものではないという管理者視点の判断で削除した．

次工程に進むための合否判定基準は，暫定的にメトリクス計測値の平均値が 0.9 以上で合格と定めた．

3.3 メトリクスに基づいた品質の計測

図 4 に品質の計測手順を示す．開発担当は各工程の最終段階で工程生産物のレビューを実施し，指摘箇所の修正を終えた工程生産物（機能設計書，詳細設計書）を品質管理担当に提出する．品質管理担当は，それまでに準備したメトリクスを用いて品質を計測し評

表 2 メトリクスデータベース
Table 2 Example of metrics database.

No.	955
ﾒﾄﾘｸｽ	要求機能実現率
定義	機能仕様書で定義されている機能数／要求された機能数
Sub-Criterion	階層化
Criterion	機能設計技術
視点	開発者
計測対象	機能設計書
計測値	0-1
出典	* * * * *
備考	
No.	956
ﾒﾄﾘｸｽ	入・出・処理定義率
定義	入力・出力・処理が明確に定義されているﾓｼﾞｭｰﾙ数／全ﾓｼﾞｭｰﾙ数
Sub-Criterion	構造化
Criterion	機能設計技術
視点	開発者
計測対象	機能設計書
計測値	0-1
出典	* * * * *
備考	

価する．

実験では FD 工程および DD 工程の生産物である機能設計書，詳細設計書を対象に計測を行った．計測には計測値を記録するための測定票および問題票を用意し，問題指摘箇所を問題票に記した．具体的な測定手順を FD 工程メトリクス No. 3「インタフェース定義率」を例にとって説明すると，「機能設計書で定義されている機能モジュール間で相互に関係のあるインタフェースの総数を分母にし，それらのインタフェースが正しく定義されているインタフェースの数を分子として計測値 (0〜1.0の値をとる) を求める」となる．同様にすべてのメトリクスの値を求めこれらの平均をとることにより，各品質特性および副特性ごとの測定値を得ることができる．このように測定値が集計された段階になると，「機能性が0.9」，「信頼性が0.85」，「使用性が0.7」などのように品質を特性ごとに数値で表すことができる．この値が，品質管理部門であらかじめ設定した合否判定基準を越えていれば合格として，問題票で指摘された不具合部分の修正を条件に次工程への移行を許可する．判定基準に満たない場合は，修正・レビューの後再度計測を行う．

一般に品質の計測作業は設計/開発担当とは独立した第3者が行うのが望ましい．今回は品質管理担当が不足していたため，同様の効果を得るため，開発担当2グループ間でクロスチェックを行う方法を採った．さらに計測の終了時に両者間で計測結果のレビューを行った．これは，メトリクスの定義に曖昧なものがあるため，計測者の主観が入ってしまい，計測結果にばらつきが出てしまう可能性があるからである．

ここで提案している方法は，従来から行われているレビューやインスペクションと本質的には同じものといえるが，以下の点を特徴としてあげることができる．

（a） 開発担当とは異なる第3者でもチェックが可能
（b） 評価結果が数値で示され，目標値管理が可能
（c） 枠組みの手順に基づいて実施するため，評価者依存性の少ない品質評価が可能

4. メトリクスの検証

新規開発時に用いたメトリクスについて，ソフトウェアの開発終了時点で個別に有効性を評価し，機能拡張時には効果の薄いメトリクスを次の判断基準に従って品質表から削除した．

A：機能拡張部分に計測対象のないメトリクス．
B：最終製品の品質への影響が少ないと認められるメトリクス．
C：計測の対象（メトリクスの式における分母の数）が少ない．

例えば FD 工程の場合，上記の基準に照らし，以下のメトリクスを除外した．

メトリクス	除外基準
・No. 4 ファイル一覧装備率	C
・No. 6 図表類参照距離3ページ以内率	B
・No. 9 章節改ページ率	C
・No. 12 出力帳票説明率	A
・No. 14 片仮名表記安定率	B

表 3 FD 工程用メトリクス
Table 3 Table of metrics used in functional design phase.

Sub Criterion	プロセスメトリクス		
	No	メトリクス	定義
階層化	1	要求機能実現率	実現されている機能／要求された機能
構造化	2	入・出・処理定義率	入力・出力・処理が明確に定義されている モジュール数／全モジュール数
モジュール化	3	インタフェース定義率	明確に定義されているインタフェース数／ 全インタフェース数
ドキュメント	4	ファイル一覧装備率	ファイル一覧表の有無
	5	用語一貫使用率	用語に一貫性のあるモジュール数/全モジュール数
	6	図表類参照距離3頁以内率	図表類の参照距離が3頁以内のモジュール数／ 全モジュール数
	7	文章非曖昧率	曖昧な文章のない頁数／全頁数
	8	ワープロ投入要領	ワープロ投入要領に準拠したページ数/全ページ数
	9	章節改ページ率	「章」「節」の区切りで改ページのされて いる章・節数 ／全章・節数
	10	機能階層関係図装備率	機能階層関係図の有無
	11	システムフロー図装備率	ｼｽﾃﾑ全体の流れ図の有無
	12	出力帳票説明率	明確に説明された出力帳票数／全出力帳票数
	13	ドキュメント読解率	ﾄﾞｷｭﾒﾝﾄの物理的な読解性の良い頁数／全頁
	14	片仮名表記安定率	片仮名表記にゆれのない頁数／全頁数
	15	レビュー指摘修正率	修正済みﾚﾋﾞｭｰ指摘項目数／全ﾚﾋﾞｭｰ指摘項目
標準化	16	データ形式標準化率	標準化されたﾃﾞｰﾀ形式の数／全ﾃﾞｰﾀ項目数
	17	入出力帳票形式 標準化率	標準化された入出力帳票数／全入出力数
	18	コマンド形式標準化率	標準形式ｺﾏﾝﾄﾞ定義数/全ｺﾏﾝﾄﾞ定義数
	19	関数形式標準化率	標準形式関数定義数／全関数の数
故障回避	20	異常処理定義率	異常処理手順の記述されたモジュール数／ 全モジュール数
	21	入力終了処理定義率	終了処理の定義された入力の数／全入力数

・No. 18　コマンド形式標準化率　　A
・No. 19　関数形式標準化率　　A
・No. 21　入力終了処理定義率　　C

また，新規開発時のレビュー結果を参考に，機能拡張時に有効と思われるメトリクスを新たに追加した．

例えば FD 工程の場合，以下のメトリクスを新たに追加した．

・No. 1　要求機能実現率
・No. 2　入・出・処理定義率
・No. 15　レビュー指摘修正率
・No. 17　入出力帳票形式標準化率

5. 評価結果

5.1 測定結果と分析

表 5，**表 6** に FD 工程と DD 工程における新規開発時，機能拡張時における 1 回目のメトリクス計測値を示す．拡張時の DD 工程以外はいずれも 1 回では 0.9 の合格基準に達せず，修正後の再計測で合格し次工程に移行した．

これらの測定結果からいえることは，以下のとおりである．

【FD 工程】

・新規開発時にはドキュメント関連のメトリクスの大部分が平均値以下の得点を示しているが，機能拡張時にはほとんどが平均値を越えている．
・機能拡張時に新たに導入したメトリクス 4 個のうち，3 個までは平均値を下回っている．すなわち，はじめて適用したメトリクスでは得点が低い場合が多いが，2 回目以降は学習効果により大幅に改善されることを示している．
・メトリクス（No. 7 文章非曖昧率，No. 13 ドキュメント読解率，No. 20 異常処理定義率）は，平均値を大きく下回った．これは，新規開発時点では設計者があまり意識して設計していなかったためであ

表 4 DD 工程用メトリクス
Table 4 Table of metrics used in detailed design phase.

Sub Criterion	プロセスメトリクス		
	No	メトリクス	定義
階層化	1	要求事項実現率	ＤＤで実現されている要求事項／ＦＤで記述されている要求事項
	2	プログラム階層化率	ﾌﾟﾛｸﾞﾗﾑ・ｻﾌﾞﾌﾟﾛｸﾞﾗﾑ・モジュ・手順の階層構造を持つﾌﾟﾛｸﾞﾗﾑの数／全ﾌﾟﾛｸﾞﾗﾑ数
	3	文章・HCPチャート一致率	文章と一致しているＨＣＰチャート数／全ＨＣＰチャート数
構造化	4	レベル構造化率	レベル構造になっているモジュールの数／全ﾓｼﾞｭｰﾙ数
モジュール化	5	モジュール入口単一化率	入り口が単一であるモジュールの数／全モジュール数
	6	モジュール機能単一化率	機能が単一であるモジュールの機能／全モジュール数
ドキュメント	7	入力元記述率	入力元の定義された入力情報数／全入力数
	8	データ表現標準化率	表現の標準化されたデータ数／全データ数
	9	共通データ一覧表装備率	共通データ一覧表の有無
	10	グローバルデータ表装備率	グローバルデータ一覧表の有無
標準化	11	初期設定定義率	初期設定の定義されているデータ数／全データ数
	12	入出力フォーマット準拠率	標準フォーマット記述された入出力情報の数／全入出力情報数
	13	入出力データ範囲記述率	取り得る値の範囲が定義されている入出力データ数／全データ数
	14	データ定義標準化率	標準データ定義で定義されたデータの数／全データ数
	15	関数名付与基準準拠率	関数名付与基準に準拠した関数の数／全関数の数
故障回避	16	異常処理定義率	ＤＤで定義された異常処理数／ＦＤで定義された異常処理
	17	入力パラメータチェック率	パラメータチェックの行われている入力データ数／全入力データ数
	18	入力終了処理定義率	終了処理の定義された入力の数／全入力数

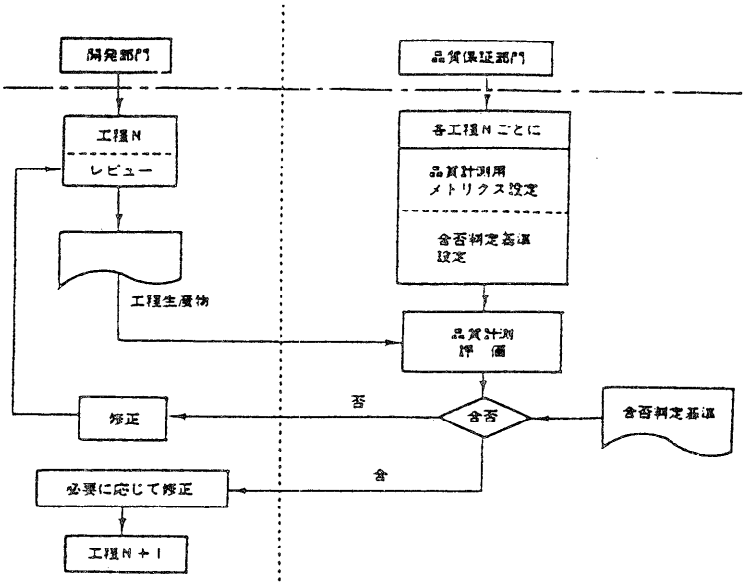


図 4 メトリクス計測手順
Fig. 4 Procedure for metrics scoring.

表 5 FD 工程メトリクススコア ▨：平均値以下
Table 5 Table of functional design metrics score.

Sub-Criterion	No	メトリクス	スコア例			
			新規開発時		機能拡張時	
階層化	1	要求機能実現率			426/527	0.81
構造化	2	入・出・処理定義率			63/75	0.84
モジュール化	3	インタフェース定義率	23/27	0.85	18/18	1.00
ドキュメント	4	ファイル一覧装備率	49/50	0.98		
	5	用語一貫使用率	26/144	0.88	98/114	0.86
	6	図表類参照距離3頁以内率	43/49	0.88		
	7	文章非曖昧率	106/144	0.74	109/114	0.96
	8	ワープロ投入要領準拠率	52/63	0.83	101/114	0.89
	9	章節改ページ率	22/22	1.00		
	10	機能階層関係図装備率	4/5	0.80	6/9	0.67
	11	システムフロー図装備率	2/3	0.67	7/9	0.78
	12	出力帳票説明率	7/7	1.00		
	13	ドキュメント読解率	122/144	0.85	101/114	0.89
	14	片仮名表記安定率	269/283	0.95		
	15	レビュー指摘修正率			52/54	0.96
標準化	16	データ形式標準化率	12/12	1.00	12/27	0.44
	17	入出力帳票形式標準化率			5/7	0.71
	18	コマンド形式標準化率	2/2	1.00		
	19	関数形式標準化率	8/8	1.00		
故障回避	20	異常処理定義率	6/15	0.40	75/86	0.87
	21	入力終了処理定義率	82/84	0.98		
			平均値	0.88		0.85

る．例えば，異常処理定義率は，新規開発時にはエラーレベルなど異常処理が必要な対象自体決まっていなかったため得点が低かったが，エラーレベルを設定し，異常処理すべき対象を明らかにした機能拡張時には高い得点となっている．

【DD 工程】

・新規開発時，機能拡張時ともに，ドキュメント関連のメトリクスはいずれも高得点を示している．

・特に機能拡張時においては，既に新規開発時の FD 工程，DD 工程および機能拡張時の FD 工程と3回のメトリクス適用の経験を経たため，全体の得点レベルは高い．

・No. 11 の初期設定定義率と No. 13 の入出力データ範囲記述率のメトリクスは新規開発時には得点が低かった．これは FD 工程の場合と同様，新規開発時には設計者の考慮が不足していたためである．このようなメトリクスに対しては，フォーマットを決めるなどして抜けを防ぐことが可能になる．機能拡張時は記述フォーマットを規定したため定義抜けがなくなり高得点を得ることができた．

このことから，設計書の記述フォーマットを定義することは，記述漏れやチェック漏れの防止に役立ち，設計品質の向上に効果があることが明らかとなった．

・新規開発時の得点の特に低いのは，No. 3 の文章・HCP チャート一致率，No. 17 の入力パラメータチェック率，No. 18 の入力終了定義率である．これらは，いずれも技術者が DD 工程で初めて HCP チャートを適用したものであり，書き方の訓練が不十分であったためと判明した．

・No. 1 の要求事項実現率の得点も低かった．このメトリクスの計測に際して「要求事項が実現されているとは，事項に漏れがあるだけではなく，その内容の正否も考慮に入れ判断する」と規定した．このためドキュメントの記述誤りをすべてカウントしたた

表 6 DD 工程メトリクススコア ▨：平均値以下
Table 6 Table of detailed design metrics score.

Sub-Criterion	No	メトリクス	スコア例			
			新規開発時		機能拡張時	
階層化	1	要求事項実現率	16/32	0.50		
	2	プログラム階層化率	65/72	0.90	168/172	0.98
	3	文章・HCPチャート一致率	11/53	0.21	135/164	0.82
構造化	4	レベル構造化率	58/61	0.95	154/166	0.93
モジュール化	5	モジュール入口単一化率			144/144	1.00
	6	モジュール機能単一化率			143/144	0.99
ドキュメント	7	入力元記述率	92/96	0.96	243/248	0.98
	8	データ表現標準化率	507/515	0.98	194/200	0.97
	9	共通データ一覧表装備率	511/516	0.99		
	10	グローバルデータ表装備率	81/82	0.99	174/174	1.00
標準化	11	初期設定定義率	19/53	0.36	125/158	0.79
	12	入出力フォーマット準拠率	161/163	0.99	296/306	0.97
	13	入出力データ範囲記述率	62/163	0.38	223/235	0.95
	14	データ定義標準化率	295/295	1.00	654/663	0.99
	15	関数名付与基準準拠率			158/158	1.00
故障回避	16	異常処理定義率	71/90	0.79	61/87	0.70
	17	入力パラメータチェック率	15/72	0.21	22/25	0.88
	18	入力終了処理定義率	3/23	0.13		
			平均値	0.67		0.93

め得点が低くなった．

5.2 プロジェクトの改善効果

表 7 に工程ごとの工数比，フォールト発生比，計測工数比率を表す．工数比とフォールト発生比とは過去の類似ソフトウェアの開発において，SQM を適用しない場合の値を 100% としたものであり，計測工数比率とはメトリクスの設定から計測まで，メトリクス適用により新たに必要となった工数を，各工程の全工数に対する比率で表したものである．工数比と品質比についてグラフ表示したのが**図 5** である．これらから，SQM を適用することにより新規開発時，機能拡張時のいずれの場合も，大幅な工数削減と品質向上を達成できることが分かる．この効果は，上流工程で十分に品質の高い中間生産物（設計書類）を作成できたため，後工程で手戻りが起こらなかったこと，および試験工程を短縮できたためと考える．しかし，詳細に結果を分析してみると新規開発時と機能拡張時とでは，効果の現れ方に差異がある．以下にそれぞれの特徴的な事象について分析する．

5.2.1 品質向上効果

（1） 新規開発時

FD 工程では過去の類似プロジェクトの約 3 倍のフォールトを摘出した．このように上流工程で多くのフォールトを摘出し品質を作り込んだ結果，製造工程以降で大幅な削減が可能となった．グラフ中の設計時のフォールトとは，設計書レビュー時に指摘された指摘項目とメトリクス測定時に指摘されたものの合計である．また，製造時のフォールトとは，コードレビューおよび単体試験で指摘された項目である．

（2） 機能拡張時

FD 工程で過去の類似プロジェクトの約 6 倍ものフォールトを検出していることが特徴に挙げられる．この理由として，前回品質メトリクスによる計測作業を経験した開発担当者が，前回の経験に基づきメトリクスによる事前レビューを実施したため，より詳細なレビュー作業が可能になったことが考えられる．その結果，DD 工程ではフォールト摘出率が過去の類似プロジェクトの約 60% まで減少している．

表 7　品質測定結果
Table 7　Results of software quality.

対象工程	工数比(%)		フォールト発生比(%)		計測工数比率(%)		メトリクススコア	
	新規	拡張	新規	拡張	新規	拡張	新規	拡張
機能設計	118	123	310	630	8.7	2.0	0.88	0.85
詳細設計	58	40	80	55	10.4	6.7	0.67	0.93
製　造	49	32	76	35				
試　験	47	50	16	65				

注：メトリクスによる品質計測は，機能設計工程・詳細設計工程のみ実施
工数比とは，今回の実績工数と過去の類似ソフトとの比である
フォールト発生比とは，今回の実績フォールト摘出数と過去の類似ソフトとの比である
メトリクススコアとは，全メトリクスの平均値である

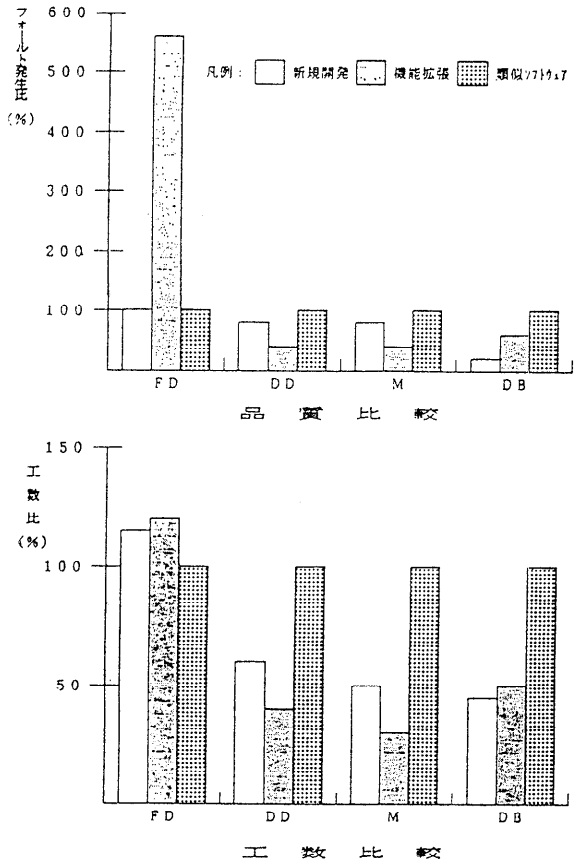


図 5　SQM 実施結果（類似ソフトウェアを 100 とする）
Fig. 5　Results of appling SQM for actual project (Compared with same category software).

5.2.2　工数削減効果

上流より品質の確保が可能となったため，試験工程の工数を約 50% 削減できた．これはメトリクスによる品質管理手法の間接的な効果である．なお，メトリクスを用いることによるオーバヘッド工数はそれぞれ以下のとおりである．

新規開発時：

FD 工程 8.7%，DD 工程 10.4%

機能拡張時：

FD 工程 2.0%，DD 工程 6.7%

ここでオーバヘッド工数は，品質管理グループによるメトリクス設定のための準備作業と開発担当によるクロスチェック作業に要した工数（人時）である．機能拡張時には，新規開発時に使用したメトリクスを再利用できたため，メトリクス設定の工数が減少している．

5.2.3　開発担当者のモラル向上効果

開発担当は新規開発時には，メトリクスとはどんなものか事前に知ることはできなかった．しかし，機能拡張時には品質が数値で評価されることと，メトリクスから品質向上のキーポイントを知ったということで，早い段階から担当者レベルにまで品質に対する意識が浸透した．表 7 に示すフォールト発生比において，技術者によるレビューで検出したフォールト件数の割合(A)とメトリクス測定時に指摘されたフォールト件数の割合(B)を以下に示す．

	A	B		A	B
新規開発時 FD 工程	2	8	DD 工程	5	5
機能拡張時 FD 工程	9	1	DD 工程	8	2

新規開発時は，メトリクス測定時の指摘項目が多いのに対し，機能拡張時はほとんどの不具合が技術者自身によるレビュー時に摘出されていることがわかる．また，FD 工程および DD 工程のメトリクススコア（表 5, 6）からも，機能拡張時の各メトリクスのスコアは新規開発時に比較し向上していることがわかる．特にドキュメンテーションのメトリクスについてそれが顕著に見られた．

6. おわりに

ソフトウェアの品質をユーザ，技術者，管理者の 3 視点から捉え，体系的に評価することをねらった新しい品質定量化の枠組みを，実際のソフトウェア開発に適用し，その有効性を確認した．要求品質は 3 種類の方法を用いて明確化した．これを実現するためのプ

ロセスメトリクスとしてFD工程で21個，DD工程で18個のメトリクスを選定し，それぞれ機能設計書，詳細設計書の品質測定を行った．この結果，品質，生産性とも過去の同様のソフトウェアの値に比較して品質では試験工程のフォールト摘出件数で約50%向上，生産性では試験工程で約40%の工数削減効果が認められた．また品質を点数で表すため，技術者に与えるインパクトは大きく，品質に関するモラルの向上にも大きな効果が認められた．さらに，設計フォーマットを定義することにより，設計品質が向上することを確認した．

今後は，この品質表（T1, T2, T3）の標準化を推進する必要がある．ただし，すべてのソフトウェア分野に共通の品質表の作成はむずかしく，当面は適用事例を増やして分野ごとに標準品質表を作成する方向で検討を進めるのが良いと思われる．また，品質計測の精度の向上のため，メトリクス計測ツールの開発が望まれる．

謝辞 本研究の機会を与えて下さったNTTソフトウェア研究所細谷僚一所長ならびに，CTRON検定システム開発におけるデータ収集に協力を得たNTT情報システム本部小田英雄氏に感謝の意を表します．

参考文献

1) Murine, G. E.: Software Measurement Techniques, *ASQC Quality Congress Transactions* (1988).
2) Boehm, B. W., Brown, J. R. and Lipow, M.: Quantitative Evaluation of Software Quality, *2nd ICSE*, pp. 592-605 (1976).
3) Murine, G. E. et al.: Applying Software Quality Metrics, *ASQC Quality Congress Transactions* (1983).
4) Sunazuka, T., Azuma, M. and Yamagishi, N.: Software Quality Assessment Technology, *8th ICSE*, pp. 142-148 (1985).
5) 情報処理振興事業協会技術センター(編)：品質機能展開による高品質ソフトウェアの開発手法「活用事例編」，(株)コンピュータ・エージ社，東京 (1989).
6) IEEE Quality Metrics Standard Committee, P1061: A draft standard for software quality metrics, Document Number, Ver. 12 (1987).
7) 日本規格協会：マンマシンインタフェースおよびソフトウェア評価の標準化に関する調査研究報告書，日本規格協会，東京 (1988).
8) 鳥居，菊野，松本，西田，岸田，浦野：ソフトウェア品質の定量化のための枠組みの検討報告，電子情報通信学会 フォールトトレラントシステム信頼性全国研究会，pp. 57-64 (1988).
9) McCall, J. A., Richards, P. and Walters, G.: Factors in Software Quality, three volumes, NTIS AD-A049-014, AD-A049-015, AD-A049-055, November (1977).
10) ISO/IEC 9126: Software Product Evaluation-Quality Characteristics and Guidelines for Their Use, ISO/IEC JTC 1/SC 7 (1991).
11) JIS X 0129：ソフトウェア製品の評価—品質特性およびその利用要領，日本工業標準調査会審議 (1994).
12) Torii, K.: A New Framework for Software Quality Assurance, *1st International Workshop on Software Quality Improvement* (1989).
13) 菊野，松本，鳥居：ソフトウェア信頼性の実現技術—現状と今後の展望—，電子情報通信学会誌，Vol. 73, No. 5, pp. 454-460 (1990).
14) 日本規格協会：システムの高信頼性技術に関する調査研究報告書第2部，日本規格協会 (1990).
15) 竹中，小田，森廣：OSインタフェース検定システム，情報処理学会論文誌，Vol. 34, No. 5, pp. 1107-1116 (1993).
16) 三宅，竹中，小田：ソフトウェア品質メトリクスの適用，第40回情報処理学会全国大会論文集，pp. 1057-1058 (1990).
17) 三宅，高橋，竹中，小田：品質メトリクスによるソフトウェア品質保証，NTT R&D, Vol. 39, No. 11, pp. 1575-1584 (1990).

付　録

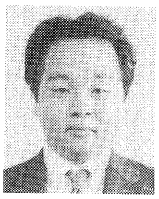
プロダクトメトリクス

・ユーザ改良要求率：
　ユーザによる改良要求の件数/出荷後の経過月数
・操作説明書と実動作の合致度：
　一致する機能項目/全機能項目数
・不正操作検出率：
　検出可能不正操作種類/不正操作種類数
・稼働率：
　稼働状態にあった総時間数/観測時間数
・平均故障発生間隔 (MTTF)：
　総稼働時間/故障発生件数
・平均ダウン時間：
　非稼働時間/ダウン回数
・平均再開時間：
　復旧に要した総時間数/故障回数
・ターンアラウンドタイム：
　処理要求の発生から完了までの時間
・応答時間：
　要求指示の終わりから応答の開始までの時間
・スループット：

単位時間内の仕事量
・CPU 使用率：
システム負荷（最大，平均）に対する CPU 使用率
・コマンド形式統一化率：
標準コマンド形式の種類数/コマンド形式の種類数
・画面操作方式統一化率：
標準に準拠した画面操作数/画面操作の種類数
・ヘルプ機能装備率：
ヘルプ付き入力操作数/全入力操作数

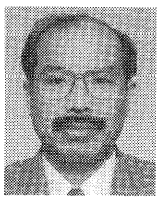
プロセスメトリクス
【設計工程】
・要求仕様展開率：
設計済み要求仕様項目/全要求仕様項目
・データ形式標準化率：
標準形式準拠データ数/全データ項目数
・コマンド形式標準化率：
標準形式準拠コマンド数/全コマンド数
・設計標準化率：
標準設計（構造化，OOD 等）準拠モジュール数/全モジュール数
・設計書レビュー指摘項目密度：
設計レビュー指摘項目数/推定ソースコード行数
【製造工程】
・モジュール見出し記述率：
見出しのあるモジュール数/全モジュール数
・コメント率：
コメント行数/全ソースコード行数
・コード自動生成率：
自動生成コード行数/全ソースコード行数
・プログラム再利用率：
再利用ソースコード行数/全ソースコード行数
・プログラム複雑度：
Halstead の係数，McCabe のサイクロマチック数等
・コードレビュー指摘項目密度：
コードレビュー指摘項目数/全ソースコード行数
【試験工程】
・テストデータ自動生成率：
自動生成テストデータ数/全テストデータ数
・テスト項目密度：
テスト項目数/全ソースコード行数
・テストカバレッジ：
C0, C1, C2 カバレッジ
・フォールト検出密度：
検出フォールト数/全ソースコード行数
・残存フォールト密度：
推定残存フォールト数/全ソースコード行数
【検査工程】
・検査データ自動生成率：
自動生成検査データ数/全検査データ数
・検査項目密度：
検査項目数/全ソースコード行数
・検査（または検定）カバー率：
検査済み機能数/全機能数
・フォールト検出密度：
検出フォールト数/全ソースコード行数
・残存フォールト密度：
推定残存フォールト数/全ソースコード行数

（平成6年6月8日受付）
（平成6年10月13日採録）



三宅 武司（正会員）

1961 年生．1985 年法政大学工学部経営工学科卒業．1987 年慶応義塾大学大学院理工学研究科管理工学専攻修士課程修了．同年日本電信電話(株)ソフトウェア生産技術研究所入所．以来，ソフトウェア開発管理システム，ソフトウェアプロジェクト管理法の研究実用化に従事．現在，NTT ソフトウェア研究所にて，ソフトウェア品質保証法，ソフトウェアプロセス評価技術，ソフトウェアメトリクスなどの研究実用化に従事．ソフトウェア科学会会員．



竹中 市郎（正会員）

昭和 18 年生．昭和 41 年九州大学工学部電子工学科卒業．昭和 43 年同大大学院修士課程修了．同年日本電信電話公社電気通信研究所入社．以来，交換用ソフトウェア開発環境，ディジタル交換網構成，ソフトウェア品質管理技術等の研究に従事．平成 6 年久留米工業大学知能工学研究所教授，現在に至る．工学博士．電子情報通信学会会員．

古山 恒夫（正会員）

1945 年生．1968 年東京大学工学部計数工学科卒業．1973 年同大学大学院博士課程修了．同年日本電信電話公社入社．横須賀電気通信研究所で，拡張型言語，Ada, Common LISP などの言語処理プログラムの研究実用化に従事．日本電信電話(株)ソフトウェア研究所でソフトウェアプロジェクト管理法，ソフトウェア品質保証法，ソフトウェア見積り法などの研究実用化に従事．平成6年度山下記念研究賞受賞．電子情報通信学会，日本ロボット学会各会員．工学博士．

中川 豊（正会員）

1946 年生．1969 年九州大学工学部電子工学科卒業．同年日本電信電話公社（現 NTT）入社．以来，オペレーティングシステム，ソフトウェア開発管理データベースシステム，ソフトウェア開発支援ネットワークの研究実用化，ソフトウェア信頼性モデルの研究等に従事．現在，NTT 情報システム本部主幹技師．電子情報通信学会会員．