

日科技連・SPCからの情報発信

クオリティ ワン *Quality One*

● 特集

Quality with Strategy ～ソフトウェア品質を高めるには

財団法人 日本科学技術連盟

クオリティ ワン Quality One



特集 Quality with Strategy～ソフトウェア品質を高めるには

Quality with Strategy ～ ITアーキテクトの観点から「品質」をどう考えるか

日本アイ・ビー・エム株式会社 榊原 彰

Quality Architectによる戦略的なテスト

電気通信大学 西 康晴

システム開発を取り巻く状況とシステム開発に求められるもの

株式会社 OSK 小井土 亨

1

論文

CMMISM 連続表現で組織に合った改善を実施するために一脱!レベル取得一

株式会社東芝 飯田 卓郎 舩薙 匠 小笠原 秀人

12

SQuBOKのご紹介

『ソフトウェアの品質に関する知識体系のガイド』策定のご紹介

SQuBOK策定部会リーダー 岡崎 靖子(日本アイ・ビー・エム株式会社)

20

コミュニティの広場

24

ソフトウェア品質に関する国際規格の紹介

ソフトウェア品質に関する国際規格の紹介

東洋大学 野中 誠

29

SPC活動のご紹介

34

ソフトウェア品質関連ニュース

36

日科技連 教育・訓練活動のご紹介

37

特集

Quality with Strategy ～ソフトウェア 品質を高めるには

Quality with Strategy ～ ITアーキテクトの観点から「品質」をどう考えるか

日本アイ・ビー・エム株式会社 榊原 彰

Quality Architectによる戦略的なテスト

電気通信大学 西 康晴

システム開発を取り巻く状況とシステム開発に求められるもの

株式会社 OSK 小井土 亨

少し前に「ユビキタス」という言葉が流行りましたが、ソフトウェアはそのもの自体は実体が無いながらも、どこにでも存在するようになりました。

社会のインフラとして電気（電力）が不可欠な様に、人間が生み出した電気があるところなら、ほぼどこにでもソフトウェアが存在するといっても、過言ではありません。

このため、ソフトウェアの不具合によって私たちは大なり小なりの影響を受けることになります。

少々不便なら目をつむることができたとしても、看過できなかったり、我慢のならない不具合があれば、社会（場合によっては世界全体）に悪影響を及ぼす不具合すらあります。

その規模や性質、影響度によって、不具合の影響度は大きく異なるのです。

だからこそソフトウェア品質は、個人の問題（ミクロ）から、家庭、地域社会国家、世界（マクロ）と、コミュニティの規模を問わず、関心事となっているのです。

ソフトウェア品質の重要性は、その使い手にも作り手にも十分に認識されているにも関わらず、ソフトウェア（によって構成するプロダクトやサービス）が持つべき品質を実現し、要求レベルを満たすのは容易なことではありません。

よく言われる理由の一つとして、「ソフトウェア品質は漠然としているから」というものがあります。

良い品質とは何か？という問いを発したとき、立場によって答えは様々です。

例えば、以下のような答えなら、すぐに浮かんでくることでしょう。

- ・不具合がない
- ・欲しい機能を具備している
- ・細かなことを指定しなくても、要求通りの操作ができる
- ・欲しいとき（タイミング）に「そこにある」

ここから分かることは、個々人が捉える品質には主観が伴っているということです。

私たちは品質について語るとき、「だいたい」や「そこそこ」良ければよいというあいまいな言い方をよくしますが、こういった表現をしている限り、実際に期待しているソフトウェアの「質」に大きなばらつきができることに不思議はありません。

実際問題として、ソフトウェアに品質を作りこみ、それが期待通りであることを評価したり分析することは一筋縄では行きません。

まず品質を作りこむといっても、その手段は色々ありますし、求められる能力は様々だからです。

少し例を挙げて考えてみましょう。

- ・ソフトウェアに何が求められているかという、要求を的確に捉える能力
- ・要求を如何に現実世界において実現するかというアーキテクチャを構築する能力

- ・アーキテクチャをベースに仕様や設計を起し、それを実装する能力
- ・実装したソフトウェアが要求や仕様、設計通りであるかを検証し、妥当性を評価する能力

これらからは、ソフトウェアを開発するプロセス自体を総合的に整備しなくてはならないことも見えてきます。

このため、どのようなソフトウェア品質を、どれだけ達成すべきかということ、ソフトウェア開発全体に影響するわけです。

その品質の達成レベルというものは、プロダクトやサービスによって違ってきます。

プロジェクト・ゴールによって異なるとも言えるわけですが、そのゴール自体は、利益、納期、安全度、信頼度、魅力など、どういった品質に関わるファクターに重きを置いているかによって変化します。

また、一つのファクターのみが重要であるということは少なく、幾つかのファクターが優先度や相関関係を持っていることが自然でしょう。

だからこそ、ソフトウェア開発において、求められる品質を実現するには、開発プロセス全体を捉えることができる「グランドデザイン」が必須となるのです。

このグランドデザインは「戦略」ともいえるでしょう。戦略自体は、組織全体で取るべきものがあれば、市場や機種、時期によって変えたり、リスクの状況によって戦略AからBへ変更などといった方向性を変える場合もあります。

戦略は固定されたものでも「一つ」のものでもないわけですが、戦略抜きに多様性を持つ品質を実現することが困難なことは「事実」といえるでしょう。

そこで本特集では「Quality with Strategy」というテーマにて、品質を作り込んだり評価するためには、戦略的にどういったことをなすべきか、できるのか、といったようなことを多角的な視点で取り上げたいと考えました。

ソフトウェアとハードウェアとの組合せからなる実体を持つ「システム」全体の品質を定義するアーキテクトの立場からは榊原氏に、ソフトウェア品質を学問的に捉えつつ、現場での展開や実践につなげる、観察・分析から応用まで発展可能な研究を行っている立場から西氏に、実際にシステムを構築しければソフトウェアを「利用者が使える」ように品質ごと作りこむ開発者の立場から、小井土氏の各氏といった産学から3人の方より、寄稿いただきました。

日科技連 SPCステアリング委員会
ジャーナル担当 大西建児
(株式会社 豆蔵)

Quality with Strategy ～ ITアーキテクトの観点から 「品質」をどう考えるか

日本アイ・ビー・エム株式会社
東京基礎研究所 IBMディスティンゲイッシュト・エンジニア (技術理事)

榊原 彰

私たちの生活基盤は今日ITシステムによって支えられており、そのITシステムの脆弱性が社会生活全般を危機にさらすということを、昨今多くの方が認識するようになりました。残念ながらITシステムの品質を完璧に保証する策を現在私たちは持ちえていません。どこまでやれば、どこまでコストをかければ万全の品質を獲得できるのかということさえ、実は不明あるいは形式化されていないというのが現実です。開発ライフサイクルのV字モデルに見られるように、ITシステムの開発は品質の作りこみ工程と保証工程に大別されます。ITアーキテクトの役割は、その多くが品質の作りこみ工程におけるものですが、これは同時に品質の保証工程における重大なインプットを生成する作業ととらえることもできます。本稿では、ITアーキテクトがどのような視点で品質をとらえるのか、そして高品質のITシステムを開発するためにITアーキテクトが何をすべきかを概観します。

証券システムの障害、銀行ATMオンラインのダウン、公共料金の超過請求など、ITシステムの不具合によるトラブルの報道が後を絶ちません。私たちの生活はもはやITなしでは語れないほど、生活のすべての面において社会インフラとしてのIT活用が行われています。電気・ガス・水道といった「ライフライン」も含む、社会・経済

活動において欠くべからざる基盤なのです。ミッション・クリティカル、セーフティ・クリティカルという単語で表現されることの多いこれらのシステムは、裏を返せば障害や機能低下が社会に与える経済面や安全面での被害が甚大だということです。ここ数年の相次ぐシステム障害のニュースは、私たちの社会基盤がいかに脆弱なITシステムの上に成り立っているかを再認識する機会を十二分に与えてくれました。こうした状況に、経済産業省からは本年4月4日に「情報システムの信頼性向上に関するガイドライン」が公表されています。ここでは、ITに関わるステークホルダ全員（技術者のみならず管理者・経営者までも含む）を対象とし、ITの企画・開発から保守・運用に至るライフサイクル全般に関する配慮が記述されています。つまりITシステムの品質に関わる責任は、関係者全員の責任でもあるということになるのですが、特に先に述べた品質作りこみ工程におけるITアーキテクトの作業は、品質の「予防保守」を行う重要な側面を担っています。

一般に、ITシステムの規模が大きくなるほど当該システムの開発全体の負荷に占める欠陥除去活動の比率は、指数関数的に増えることが知られています。規模の増大は単純にケアレスミスによってバグを生む機会も増やす、というだけであれば両者は比例した関係のはずで

が、規模の増大化は複雑性をも増長させるため、実際には非線形な増加曲線を描きます。したがってこの問題の単純な解消策として（しかも効果的でもある）、システムをサブシステムなどに分割し、単一開発スコープにおける複雑性の増大を防ぐという手段を用います。適切な分割は、作業効率を高め細部にわたって開発情報を見渡すことが可能な体制を作り出すのです。ところが、分割の基準を間違えると（分割箇所を間違えると）、そこに新たな複雑性を生み出し、結果としてバグの温床を作りこむことになりかねません。そこで、ITアーキテクトという職種が必要になります。ITアーキテクトの責務とは、その名が示すとおりITアーキテクチャを構築することですが、問題はどのように構築するかという点なのです。

ITアーキテクトの視点

通常ITシステムは、ネットワークやサーバーなどの機器構成、それらの上で動作するミドルウェア群などからなるインフラ部分と、ビジネス・ロジックをソフトウェア・コンポーネントとして実装したり、それらを支える仕組みとしてのフレームワークなどからなるアプリケーション部分の組合せとして設計されます。

アプリケーション部分の品質に関しては本特集内の小井土氏の記事にその一端が書かれていますので、字数の少ない本稿において詳細を述べることはしませんが、ここでのアーキテクチャの観点からのキーポイントは、コンポーネントのインタフェース設計とコンポーネントの凝集度、コンポーネント間の結合度を意識した設計です。

一方、インフラ設計では通常以下のようなことを検討します。

- システム構成の検討
- パフォーマンスやキャパシティなど必要な性能・品質の検討
- 障害対策の検討
- セキュリティ対策の検討
- 保守・運用監視などの機能検討…等々

問題はいつもプロジェクトの早い時点で（時としてプロジェクトが始まる前に）物理的な構成ありきで検討が始まることです。コンポーネントの開発を行う際には、分析→設計→実装と段階を経て仕様の洗練度を高める（あるいは細粒度に落とし込む）のに、インフラを中心としたアーキテクチャ策定ではその手順がとられないことが大多数です。しかし、本来ハードウェア仕様や、使うソフトウェアは要求が確定した段階で最終決定される

べきもので、既存資源の使用という制約はあるにしても、もっと概念レベルからの設計思考法を取り入れる必要があるのです。物理的な要素を排除したモデルを作成しておけば、デザイン・パターンなどと同様にインフラ設計自体のパターン化も可能になり、何よりも要求に応じたハードウェア、ソフトウェアの調達にも柔軟性があるからです。

振り返ってみて品質とは

さて、ISO9126において定義されている品質特性をとらえる見方として、外部品質と内部品質というものがあります。これは設計品質と適合品質という観点でとらえる際に利用する品質特性の切り口ということになります。つまり、

- 外部品質： ユーザ要求と設計仕様の合致度合い
- 内部品質： 設計仕様と実装の合致度合い

ということであり、「正しいものを作っているのか」という視点と「正しく作っているのか」という視点です。もちろん、双方の品質に関してITアーキテクトは責任を負う立場にありますが、ITアーキテクトが戦略を持って決めていかなければならないのが外部品質です。つまり「正しいものを作る」という命題をもっていかにうまく作っていくかという点は、個々の要素技術のスペシャリストではなくITアーキテクトが決めなければならないのです。内部品質の観点での品質要求を満たすための技術はさまざまです。たとえば設計内容からの実装コードの自動生成や、テスト駆動開発などがよい例です。これに対し外部品質の品質特性に関しては、要求をいかに設計に反映させるかという点での技術が必要であり、これを満たすためには、まずモデルを作成するところから品質設計を始めなくてはなりません。モデルで実行時の品質がわかるのかと訝しく思われるむきもあるかもしれませんが、モデルを作成するという事は、品質上のさまざまな要素を明確にしていくという点で重要な取組みなのです。要求をどのように実現するかという確証を得るためには、システム開発の早い段階で品質に関する検討が可能となるようモデルが必要になります。ここで言うモデルとは、必ずしもUMLなどのダイアグラム描画だけを意図しているのではなく、さまざまな説明記述や見積り予測など数式も含んでいます。ソフトウェア構造のみならずインフラ構成もモデル化するからこそ可能になることも多く、品質を考えるならばアーキテクトはもっとモデル化ということに敏感になるべきです。モデル化

することによって設計がしやすくなる品質特性としては、ISO9126で言えば信頼性・効率性などの多くと機能性・保守性・移植性の一部が該当します。これに対して、使用性はモデル化することが最も難しい特性の1つであると言え、早期に顧客要求の満足度を推し量るという観点では、モデル化よりもプロトタイピングによる方が現実的です。

ビジネスとの整合性という戦略

前述の外部品質を考えるならば、「正しいもの」ということも併せ考える必要があるでしょう。ITシステムの使用において「正しいもの」とは、ビジネスの成功裏な遂行を支えるシステムのことです。したがって、いくら高可用性・高信頼性を備えたITシステムであっても、ビジネス上のフォーカスポイントを外したものであってはならないのです。24時間365日ノーダウンの受発注・在庫管理のシステムがあったとしても、適正在庫を確保できずに販売機会を逸することが頻発するのでは、システムとして役に立っていないということになります。このような観点においては、ITシステムの品質とは、ビジネス・パフォーマンスによって評価されなければいけません。昨今のシステム開発環境や開発手法は、ビジネスとITの垣根を取り払う方向に向かっています。ビジネスプロセス・モデリングを行い、WS-BPELなどのプロセス実行コードを生成し、Webサービス化されたコンポーネントを連携させていくSOAの手法などはその典型でしょう。ビジネス戦略を形にしてプロセスをモデル化し、IT

システムの開発・展開を経て、ビジネス・パフォーマンスの評価をビジネス戦略の次のステップにインプットする。ITアーキテクトの責務は、いまやITシステムの設計・構築のみならず、ビジネス・アーキテクチャの設計にまで及んでいます。ITアーキテクトにとってQuality with Strategyとは、Business Driven Development（ビジネス駆動型開発）そのものなのです。

略 歴

榊原 彰 (さかきばら あきら)

日本アイ・ビー・エム株式会社 東京基礎研究所
IBMディスティングイッシュト・エンジニア
(技術理事)

IBM Academy of Technology メンバー

公 職：情報処理学会、PM学会、IEEE Computer Society、ACMの各正会員。ソフトウェアエンジニアリング・センター非機能要求/アーキテクチャ設計部会主査、ITスキル標準センター ITアーキテクト委員会主査、日科技連SPCステアリング委員会副委員長。

研究分野：アーキテクチャ設計技術、モデル駆動型開発

Quality Architectによる戦略的なテスト

電気通信大学
電気通信学部 システム工学科 講師

西 康晴

ソフトウェア開発において、最も戦略的でない工程は何でしょうか。多くのエンジニアは、テストを挙げるでしょう。残念ながら、頑張っているものの、今も多くの現場ではテストを行き当たりばつたりに進めているのが現状です。そこで本稿では、テストを行き当たりばつたりではなく戦略的に進めるために必要な考え方を、いくつか紹介します。

戦略的な品質向上を担う Quality Architect

戦略的に進めるとは、一体どういう意味なのでしょう。例えば局地的な戦闘のような、個々の局面における目標達成を指しているわけではありません。事業全体における大きな目的を達成するために、全ての局面を俯瞰し、長期的な視野で進めていくことを意味しています。個別最適ではなく全体最適を狙うのだ、と言い換えてもよいでしょう。すなわち、行き当たりばつたりなソフトウェア開発というのは、個別最適の罠に陥っているのです。

日本のソフトウェア開発の現場には、すっかりプロジェクトマネジメント（PM）という概念が定着しました。これによってマネジメント力が向上したのは事実です。しかしPMの概念を誤解している現場では、個々のプロ

ジェクトのQCDを達成することだけがPMだと勘違いして個別最適に陥るあまり、組織全体として開発力が疲弊しています。お手軽にコストダウンを進めようと、発注品質を向上せず安易にオフショア開発に手を出し手戻りが多発する、というのも、個別最適による誤謬でしょう。プロセス改善という概念を誤解している推進組織は、標準プロセスへの準拠の強制とCMMIのレベル取りという誤った個別最適に陥ってしまっています。品質を向上しようと品質保証（QA）とソフトウェアプロセス改善（SPI）とPMOの組織を設置したにも関わらず、現場はそれぞれの組織に報告義務を負ってしまい、かえって大変になってしまった組織もあります。能力の高いエンジニアを多く抱えているはずの日本のソフトウェア産業がこれほど赤字やトラブルに苦しんでいるのは、個別最適を狙うあまり全体最適を見失うという罠にはまっているためであると筆者は考えています。

したがって重要なのは、個々の技術力やマネジメント力を向上するだけでなく、戦略的にソフトウェア開発を進め全体最適を達成し、組織全体の開発力を向上することで長期的にコストを低減し品質を向上していくことです。とはいえ全体最適を狙う組織をまた新たに設置したのでは、元の木阿弥です。全ての個別最適はもともと、全体最適を狙う手段として始められたわけですから。必

要なのは、組織の構成員すべてが全体最適を狙って行動することであり、そのきっかけとなる役割を開発現場の内部に配置することです。これは各プロジェクトの戦術的成功を支援しつつ、組織全体の全体最適を推進するという高度な作業に他なりません。この役割を担う人財を"Quality Architect"と呼びましょう。

Quality Architectは、プロジェクトの品質マネジメント、レビューやテストなどのV&V、プロジェクト内のプロセス改善など品質に関わる活動を俯瞰的にデザイン・推進し、品質に関する責任を負います。同時に、プロジェクト内の品質達成が組織的な全体最適を推進するような施策を取らなくてはなりませんし、個々の開発技術者が全員参加で全体最適に寄与している感覚を持つよう泥臭い活動をする必要も必要です。プロジェクト内の戦略的な品質向上と、組織全体の戦略的な品質向上の2つの活動をバランス良く達成するというミッションを持つわけです。そのためQuality Architectは高い開発力やマネジメント力を備えるだけでなく、有機的に結合させた様々な品質技術を俯瞰的な施策と泥臭い活動によって実現するという高い能力を備えていなくてはなりません。品質に関わる技術者のキャリアの最高峰と言えるでしょう。

戦略的なテストによる全体最適の実現

次に、Quality Architectの視点からテストを眺めてみましょう。多くのテストの現場では、品質を向上するために大量のテスト項目を挙げ、人海戦術で実施しています。そうした現場では、品質を向上することは（テストの）コストを増大させることと同義です。

しかしこれは個別最適に他なりません。水際での膨大なテストというのは戦術的に仕方がないのかもしれませんが、長期的には組織が疲弊していきます。

逆に、テストの上手な現場ではどのように進めているのでしょうか。そうした組織では、バグが出やすいところを上手に狙ってテスト設計を行い、バグが無さそうなテスト項目を適切に間引いています。別に魔法を使っているわけでも何でもありません。注目すべきは、テスト技術者と開発技術者とが密接にコミュニケーションをとっている点です。一体全体、何を話しているのでしょうか。

腕の良いテスト技術者は、テスト対象に検出されたバグを様々な観点から頭の中で分析し、同じようなバグが他にも作り込まれていそうかどうかを推測しています。そして開発者に、似たようなバグが作り込まれている可能性と推測される機能を伝えるのです。それによって開

発者はデバッグの際に水平展開をして効率的にコードや設計、仕様を修正していきます。

また腕の良いテスト技術者には、頭の中で分析したバグのパターンが蓄積されています。

そこで分析や設計といった開発のレビューにも参加し、開発者がバグを作り込んでしまいそうな「匂い」のする部分を指摘することで、未然に不具合を防いでいます。気の利いた開発者であれば、そうした匂いをレビューのチェックリストや開発のガイドラインに反映させることでしょう。腕の良いテスト技術者は開発経験が少ない場合もありますが、豊富なバグのパターンを持っていることで開発者に深く信頼されますし、気軽に相談を受けます。特に信頼の深い場合は、仕上がりに心配な設計やレビューしきれなかった部分を開発者がテスト技術者に伝えることでカバーしてもらうこともあります。

一方彼らは別の相談もしています。特に組み合わせのテストは膨大になるため、適切な間引きが必要です。とはいえ何も考えずに間引いてしまうと、バグの検出漏れにつながってしまいます。そこで組み合わせのバグが発生してしまうような設計上の依存関係があるかどうかを開発者に検討してもらい、よりバグが漏れにくいテストの間引きを実現しているのです。開発の上流からテスト技術者が参画し並列に作業している場合は、組み合わせのテスト項目数が爆発しすぎないように、依存関係が集中し核になっている部分をスッキリと設計してもらうように依頼することもあります。

もちろんユーザの要求の重要度をヒアリングしてテスト項目の優先度に反映させたり、テストしきれずに出荷せざるを得ない部分の品質リスクが許容範囲内なのかどうかを相談する場合があります。それにより開発者自身も要求の不備に気づき、改善につながります。

こうした腕の良いテスト技術者が自然に実施している施策は、人海戦術によって大量のテストを消化し品質を確保する個別最適の罫にはまったテストとは異なります。テストの質を向上することで開発全体の質が上がるという、個別最適と全体最適とを同時に実現できる施策なのです。技術的にはそれぞれ不具合モードやWモデル、リスクベースドテストと呼ばれていますが、重要なのはこのような技術をただ導入することではなく、戦略的な位置づけ、すなわちQuality Architectの視点を十分反映させるように導入することです。

そして個別最適のための活動や努力が、全体最適にきちんとつながっていることを把握し確認し続けることです。それこそが全体最適を実現する戦略的なテストと言

えるでしょう。

Quality Architectは、現場が思いこんでいる「品質を上げようとするコストがかかる」という誤謬を払拭し、「品質を上げようとするコストが下がり、納期が短縮する」という正しい認識に改めることが使命です。テスト

トをきっかけに自組織の品質改善活動を棚卸しし、個別最適のために全体最適を阻害する施策になっていないかどうかを確認することを通じて、多くの組織でQuality Architectが育成されることを願っています。

略 歴

西 康 晴 (にし やすはる)

電気通信大学 電気通信学部 システム工学科 講師

Profile:

日科技連SPCステアリング委員会副委員長

ソフトウェアテストシンポジウム (JaSST) 共同実行委員長

日本ソフトウェアテスト資格認定委員会 (JSTQB) ステアリング委員長

組込みソフトウェア管理者・技術者育成研究会 (SESSAME) 副理事長

某社でソフトウェアテストに関するコンサルティング部門を立ち上げた後、東京・調布の電気通信大学にて研究や教育、コンサルティングを行う傍ら、ソフトウェアテストのエバンジェリストとして「現場に笑顔を」をキーワードに飛び回っている。

システム開発を取り巻く 状況とシステム開発に求 められるもの

株式会社 OSK
R&D本部 技術支援部 システム開発技術課
シニアアプリケーションスペシャリスト

小井土 亨

1 最近のシステム開発を取り巻く状況

1.1 情報システムの役割の変化

多くのビジネス分野において、情報システムは、単なる業務処理の効率化ツールから、ビジネスを提供するための基盤としての役割を担うように変化してきています。銀行、証券、保険業界はビジネスの成り立ちとして情報システムがビジネスの基盤となっているのは当然ですが、このようなビジネスドメイン以外でも、情報システムは、生産から加工そして販売までの一貫した商品のトレーサビリティの要求、インターネット電話、宅配便の配送状況、バスの運行状況のリアルタイム配信、大学講義の休講状況など、日常生活を行う上での必須インフラとして比重を高くしています。

1.2 情報システムへの新たな脅威

情報システムの役割の変化と適用領域の拡大によって、今までは存在しなかった以下のような様々な脅威が発生しています。

① 誤操作による多額の損害

株の売買やネット販売などでは、一人のオペレータの販売額のうっかり入力ミスによって、会社の存在自体を左右するような多額の損害が発生しています。

② 悪意のある攻撃

情報システムの社会に対する比重が高くなったこ

とによって、情報システムやそのインフラであるネットワークシステムは、テロや愉快犯の攻撃対象になっています。ウイルスやワームによる様々な情報の流出は、後を絶ちません。

③ 悪意のある利用

電子メールでの詐欺やクレジットカード番号や銀行口座IDの盗用による預金の引き出しや商品の購入なども、多発しています。また、悪意を持ったシステム運用者による、情報の持ち出しや転売が日々発生しています。

1.3 情報システムへの期待

以上のような状況によって、情報システムに対する期待は、さまざまな側面で高くなっています。日本版SOX法（金融商品取引法）でも、ITの利用が盛り込まれており、特に操作や業務処理のモニタリングなどは情報システムへの必須要件となりつつあります。

2 システム開発から見た状況の変化

2.1 情報システムの複雑さの増大

以前のシステム開発においては、入力画面、帳票出力や保存情報などに代表される機能要件だけに注目して開発が進められました。しかし、情報システムを取り巻く状況の変化により、セキュリティなどの一般的な非機能要求や、更には悪意ある攻撃があってもシステムやサー

ビスを安定して提供するなどの要件を考慮して開発を進める必要がでてきています。これらの要件を考慮することは、システムの複雑さを増大させることを意味しています。結果として、システム開発の難しさも増大しています。また、問題の中には、システムだけではなくミドルウェアやハードウェアなどに対して、総合的に対応する必要のあるものが少なくありません。このような状況が問題を更に複雑にしています。

2.2 ビジネスとシステム開発のベクトルのずれ

このように情報システムをインフラとしたビジネスを取り巻く状況は、問題や攻撃への対応や新しい機能の追加など、多くの変化を要求します。システム開発の初期段階で決定した方針は、時間の経過によって発生した様々な要因に対処するために変更したビジネスの方針とベクトルのずれが生じます。このずれを是正するためには、システム開発の進め方を変える必要があります。しかし、多くのシステム開発プロジェクトでは、現状の業務の効率化だけを目的としたシステム開発の契約形態や手法や体制を維持しようとしているため、最終的にユーザーにとって有効性の低いシステムを提供することになります。更に、開発組織や管理体制も、変化をさせないことや変化を最低限にすることだけを目的に構築されているため、この問題を大きくしています。結局、このずれが結果として、ビジネスの足を引っ張る情報システムという結果を生んでいるわけです。

一部の管理者は、ずれが発生するのは、先を予測できない開発者の言い訳であるという言い方をします。しかし、成功するビジネスは新しいアイデアを実現することが必要になることを考えれば、この意見は、開発者に預言者になることを要求していることは明白です。

3 システム開発に求められるもの

3.1 ビジネスに追従するためのシステム開発の準備

ビジネスに追従するシステムとは、一度開発したら後は不具合の修正だけというシステムではなく、様々な側面の要求に対応する、進化を続けるシステムです。実際、Web上でサービスを提供しているいくつかのシステムの中には、サービスを開始して1年以上経過しても、 β バージョンと表示しているものがあります。これらのシステムの多くは、常に新しいサービスの追加を行っており、それを継続することの決意表明として、このような表示を行っています。

このようなシステムを開発の側面から見た場合、良い設計に基づくシステムであることが必須条件となります。それは、良い設計を実現することで以下のようなシステム特性を実現することができるからです。

①正確性

仕様を正しく満たしている。

②理解性

設計結果が読みやすく理解しやすいこと。

③一貫性（統一性）

設計上の個々の概念が首尾一貫して、ぶれがない。

④変更容易性

機能強化などに伴う変更が容易であること。

⑤頑健性

誤った使い方に対して、システムが適切に対処できること。システムの一部のバグが全体に波及しないこと。

⑥移植性

いろいろなハードウェア、ソフトウェア環境へ容易に移植できること。

⑦効率性

実行効率、資源効率ともに十分実用に適応していること。

悪い設計のシステムは、変更時間に膨大なコストが必要になってしまう、あるいは、一部の修正が全体に影響するなどの問題が発生してしまうことになります。最悪、機能を追加する場合、変更する工数より作り直しの工数の方が少ないといったことになってしまいます。

3.2 テスト容易性の重要性

高い品質を維持した状態で進化を続けるためには、良い設計を実現する必要があるという、説明をしました。しかし、ここで大きな問題があります。機能拡張を行う前の良い設計が、機能拡張後も良い設計であるとは限らないということです。したがって、良い設計を維持するためには、設計の改善を繰り返すことが必要となります。

システム内部の設計を変更しても、外部に提供しているサービスには影響がないことをテストで証明し、品質の劣化が発生していないことに注意を払う必要があります。したがって、進化を続けるためには、常に多くのテストを行う必要があります。そこで有効な手段としては、テストの自動化があります。テストを自動化することで、人的工数を下げることができます。しかし、テストを自動化することは、一朝一夕に実現できるわけではありません。テストを自動化するためには、設計段階からテストすることを前提にして開発を行うことが重要です。また、すべてのテストを自動化することを目的にすると挫折してしまうので、自動テストできる範囲を広げることを目的にすべきです。

実は、自動テストを設計の早い段階から考慮し、実装の開始と同時に自動テストを作成することで、多くの設計上のメリットを得ることができます。

良いテストには以下のような条件が要求されます。

- ①テスト目的が明確である
何をテストするか明確である。その目的も明確で、更にひとつである。
- ②テストの判定が正しい
テストの成功、失敗が正しく判断されている。
- ③テストを独立して実行することができる
テストが他のテストに依存することなく、独立している。
- ④繰り返し実行することができる
何度でも繰り返して、テストを実行することができる。
- ⑤テストを実行しても、状態が変化しない
テストを実行し、成功した場合でも失敗した場合でも、テストを実行する前と後で何も変わらない。

これらの条件を満足するテスト対象は、以下のような良い設計の条件を満足することになります。

- ①テスト対象が一つの機能を実現している
明確な機能だけを提供する。
- ②結果を提供する
外部に対して、処理結果を判断できるような何らかの方法を提供する。
- ③独立度が高い
テスト対象が他の部分への依存度が低い。
- ④特定の環境への依存度が低い
特定のファイルやデータベース構造などに対する依存度が低い。

このような条件を満たして、高凝集と低結合のバランスの取れたシステムが実現されます。更に、テストは目的を表したもので、実装は実現方法であり、この2つの視点でシステムを明確にすることは、要求から実装までの関連を明確にすることになります。このことは、システムを変更するときに、変更を行うべき部分を明確にすることや変更が影響する範囲を確定することに、有効に作用します。

システムを高い品質に保ち、その品質を維持したまま、様々な機能拡張を繰り返すシステムに対して、テスト容易性を指針とすることは大変有効です。それは、良いテストが良い設計を証明することや、品質を維持したまま設計を改善するための重要な武器になるからです。

3.3 自動テストを支援するためのツール

最近、多くの開発言語ごとに、テストフレームワークが提供されています。これらの多くは、オープンソースのフリーソフトとして提供されており、利用することができます。これらのツールを利用する場合、ネット上に公開されている、既存の利用者情報が役立ちます。また、多くの書籍も出版されています。

最近、脚光を浴びているDIコンテナでは、独立性の高いコンポーネントの開発と、そのコンポーネントの組み立てを分離することで、コンポーネント単体のテストやコンポーネントの関連テストを容易に実現することができます。一部の開発者は、DIコンテナをテストコンテナと呼んでいます。

3.4 ディペンダビリティを確保するためには

ディペンダビリティを確保するためには、システム自体が良い設計を基に良い実装を持ったシステムであり、変更容易性やテスト容易性などの良い特性を持っていることが必須です。しかし、現在開発されているシステムで、良い設計を十分に実現されているものは決して多くないと思います。更に、十分な自動テストを持ち、テストも重要なシステム資産としてメンテナンスされているシステムは更に少ないと思います。システムが、ディペンダビリティを確保し続けるためには、新たに生まれる脅威に対して常に対応し続けることが求められます。

変化を受け容れることは、場当たりの開発とは、まったく異なります。常に、設計やテストにコストを払い、顧客のビジネス価値を上げるという方針を最優先に情報システムを提供することなのです。

システムを、様々な視点から十分な品質を確保し、システムを進化させながら、その品質を確保し続けるためには、変化に対して常に前向きに向かい合う、「カイゼンを実践するマインド」が開発者にも要求されていると考えます。

略 歴

小井土 亨 (こいど とおる)

株式会社OSK
R&D本部 技術支援部 システム開発技術課
シニアアプリケーションスペシャリスト

兼務公職：情報処理学会ソフトウェア工学研究会パターンワーキンググループ テストパターン リーダー
日本XPユーザグループ 運営スタッフ
INETA Japan 運営委員
Microsoft MVP
(Most Valuable Professional)
for Visual Developer ?
Solutions Architect
日科技連SPCステアリング委員会委員

専門・研究分野、関心のある分野：

- ・ソフトウェアパターン
- ・アジャイル・ソフトウェア開発

CMMISM 連続表現で組織に合った改善を実施するために

— 脱！レベル取得 —

Improve to Meet Your Organization Needs with reference to CMMISM Continuous Representation
- Good-bye, Level Hunting! -株式会社東芝 ソフトウェア技術センター
TOSHIBA Corporation Software Engineering Center○飯田 卓郎¹⁾
○Takuro Iida○艸薙 匠¹⁾
Takumi Kusanagi○小笠原 秀人¹⁾
Hideto Ogasawara

In the staged representation, there are problems that we face a crisis of level hunting and we cannot choose process areas that meet our organization needs. In order to improve to meet our organization needs, it is important that a priority of improvement items and choosing process areas based on business goals, SEPG activities and case study information. In the continuous representation, we can say good-bye to level hunting and we can choose process areas that meet our organization needs. In this paper, we show that the usefulness of the continuous representation and the lessons-learned by carried out appraisal.

1. はじめに

当社ではSW-CMM (Capability Maturity Model for Software®) を参照してソフトウェアプロセス改善活動を継続的に実施している^{[4][5]}。組織人員全員が改善活動に参加することは改善活動成功への必須条件である。上級管理者と開発担当者の意識の違いによる改善活動継続の危機はあちこちらで垣間見られ、その一つに管理者からのレベル取得プレッシャーが挙げられる。開発担当者は日々の業務が効率よく実施できるよう改善を継続し、改善を積み重ねた後にアセスメントで改善の状態を確認するよう努力しても、改善が積み重ねられる前に成果を求める上級管理者がレベル取得を目標に掲げる場合がある。上級管理者からのレベル取得プレッシャーと担当者が抱える問題解決との間に挟まれながら、改善活動の長期的視点やプロセス改善計画など提示して上級管理者のレベル取得を考え直させてきた。このような問題と向き合っているとき、CMMI (CMM IntegrationSM) 連続表現がレベル取得にならないモデル表現であると考えた。CMMIでは段階表現と連続表現の2通りの表現形式があり、改善活動の目的によりどちらを用いるか選択できる。CMMIモデルそのものは変わらず表現形式だけが変わるため、モデルとしての差はない。つまりまったく別の異なるモデルをあらたに導入しなくても、モデルの表現を変えるだけでよい。

段階表現では、何が改善されたかより成熟度レベルがいくつであるかに関心が集まり、レベル向上しさえすればQCDが向上するとの誤解の危険もある。連続表現ではプロセス領域別に能力レベルが設定されているため、様々な組み合わせの結果が出される。レベル到達したかレベル到達しないかの二つに一つの結果にならない点が、レベル取得から開放される要因と考えた。

そこで当社でCMMI 連続表現を用いた改善活動を実施し、SCAMPI (Standard CMMI Appraisal Method for Process Improvement、SW-CMMのアセスメントに相当) を実施しレベル取得ではない組織に合った改善が実施できるかを確認した。

当社においてもCMMIへ移行する組織が出現しているが、段階表現を参照する場合はほとんどで連続表現は参照されていない。また、社外に目を向けるとCMMIモデルを読み比べて段階表現と連続表現での比較は研究・調査としての報告事例はあるが、公表されている連続表現での改善

1) 株式会社東芝 ソフトウェア技術センター
TOSHIBA Corporation Software Engineering Center
神奈川県川崎市幸区小向東芝町1 Tel: 044-549-2405
1, Komukai-Toshiba-Cho, Saiwai, Kawasaki, Kanagawa Japan

事例は少なく、段階表現と連続表現を比較するための事例がそろえられないのが現状である。本論文の事例は、連続表現導入の際に活用できる内容である。

2. 当社における CMMI の推進事例

2000 年頃より SEPG^(注) が組織化され改善活動を継続的に実施している。2005 年 3 月時点で当社グループ会社に設置された SEPG は 49 組織にも増加している。現在 SW-CMM 利用組織は全体の約 8 割、CMMI 利用組織は約 2 割であり、SW-CMM から CMMI へ移行する組織も存在する。SW-CMM から CMMI へ移行する場合、SW-CMM 成熟度レベル 3 到達後に CMMI 成熟度レベル 3 への移行を推奨しているため、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 (CMU/SEI) が

2005 年 12 月で CBA-IPI (CMM Based Appraisal for Internal Process Improvement) のサービスを終了しても、当社では SW-CMM を用いて改善を継続する組織が多く存在すると推測している (図 1)。また、SW-CMM を参照して改善活動を推進する組織のために、社内アセスメント体系を整備し CBA-IPI 相当のアセスメントを継続して提供するための準備を進めている。

コーポレート SEPG では、2002 年 4 月から CMMI の調査を開始し、2003 年 3 月には SW-CMM から CMMI へ移行するためのガイドライン (CMMI を新規導入する場合のガイドラインを含む) を作成した。2003 年 12 までに SCAMPI リードアプレイザを 3 名養成し、社内でのアプレイザル実施準備を行った。CMMI モデルの理解と適切な解釈のため、推進者側を対象に CMMI 公式入門コースを調査段階の 2002 年 7 月に開催し、2003 年下期からは半年に 1 回の間隔で継続的に開催している。SEPG リーダを育成するための「SEPG リーダ教育コース」において、2003 年度から CMMI 概要教育を提供し、2006 年度では従来の SW-CMM を基盤とした内容から CMMI へ全面改訂する。また、CMMI モデル解釈の理解を助けるために、CMMI 解釈表 (全 25 表)、CMMI プロセス領域ガイド (全 23 冊) を作成しており 2005 年度中に完成する予定である。CMMI プロセス領域ガイドは社内事例を含めて解説しているため、社内での活用に適しており、社内で統一された解釈で改善を進めるための参考書となっている。図 2 に当社における CMMI 推進活動の概要を示す。

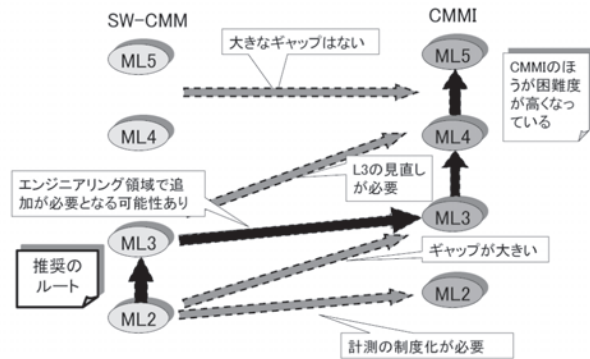


図 1 : SW-CMM から CMMI への移行ガイドライン

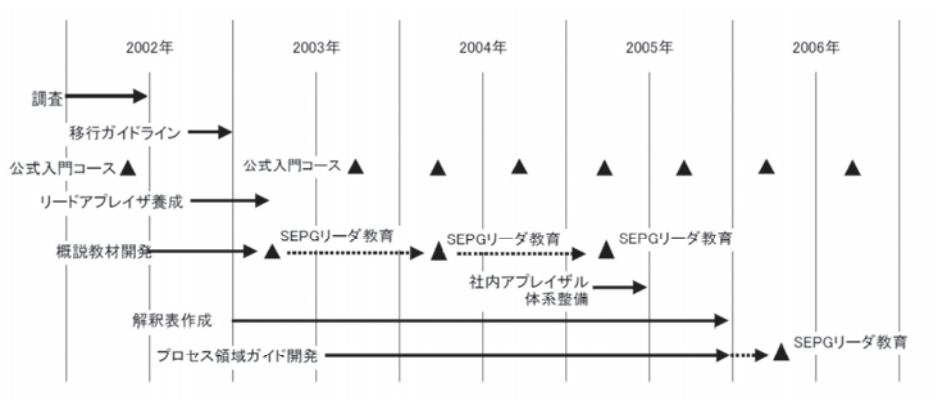


図 2 : CMMI 推進活動概要

(注) SEPG : Software Engineering Process Group の略。社内で SEPG という呼称が定着しているため CMMI の場合でも PG (Process Group) ではなく SEPG を用いる場合が多い。

3. CMMI 段階表現と連続表現

3.1. 段階表現と連続表現の特徴

CMMI モデルには段階表現と連続表現の 2 種類の表現がある。段階表現は成熟度レベル別にプロセス領域 (PA: Process Area) が割り当てられており、成熟度レベル 1 から順に成熟度レベル 5 へ段階的 (表 1) に改善を実施する筋道が示されている^{[1][2]} (表 3)。一方、連続表現は、プロセス領域に能力レベル (1~6) が存在し (表 2)、どのプロセス領域を選択するかは組織が決定でき、連続表現においては組織の考える優先度でプロセス領域を選択できる。例えば、組織の成熟度レベルが 1 である場合、通常成熟度レベル 2 のプロセス領域の改善に着手し、成熟度レベル 3 のプロセス領域は後回しとなる。連続表現においては、成熟度レベル 2 から成熟度レベル 5 までのプロセス領域から自由にプロセス領域を選択できるため、成熟度レベル 1 の組織であっても成熟度レベル 2 の要件管理、成熟度レベル 3 の要件開発を改善初期から選択することも可能である。しかし、成熟度レベル 4 の定量的プロセス管理と定量的プロジェクト管理だけを選択することは推奨されない。なぜなら、定量的管理を実施するための前提条件として組織としての計測が制度化されている必要があり、各プロセス領域間には順序関係が存在する。このような順序関係を考慮してプロセス領域を選択することが期待されている。

表 1 : 段階表現の成熟度レベル^{[1][2]}

成熟度レベル	説明
1	初期
2	管理された
3	定義された
4	定量的に管理された
5	最適化している

表 2 : 連続表現の能力レベル^{[1][2]}

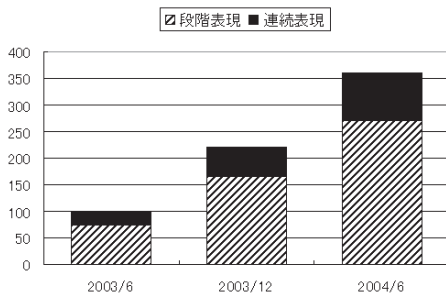
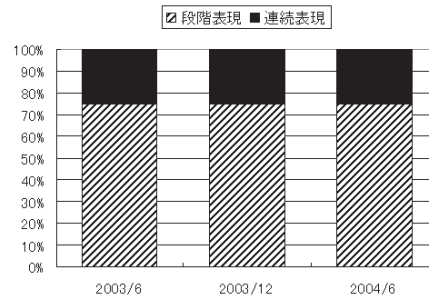
能力レベル	説明
0	不完全な
1	実施された
2	管理された
3	定義された
4	定量的に管理された
5	最適化している

表 3 : 成熟度レベルとプロセス領域の区分に割り当てられたプロセス領域^{[1][2]}

		プロセス領域の区分			
		プロセス管理	プロジェクト管理	エンジニアリング	支援
成熟度レベル	5	組織改革と展開 (OID)			原因分析と解決 (CAR)
	4	組織プロセス実績 (OPP)	定量的プロジェクト管理 (QPM)		
	3	組織プロセス重視 (OPF) 組織プロセス定義 (OPD) 組織トレーニング (OT)	統合プロジェクト管理 (IPM) リスク管理 (RSKM) 統合チーム編成 (IT) 統合供給者管理 (ISM)	要件開発 (RD) 技術解 (TS) 成果物統合 (PI) 検証 (VER) 妥当性確認 (VAL)	決定分析と解決 (DAR) 統合のための組織環境 (OEI)
	2		プロジェクト計画策定 (PP) プロジェクトの監視と制御 (PMC) 供給者合意管理 (SAM)	要件管理 (REQM)	構成管理 (CM) プロセスと成果物の品質保証 (PPQA) 測定と分析 (MA)

3.2. 段階表現と連続表現の利用比率

CMU/SEI はプロセス成熟度プロファイルを半年に 1 回報告しており、SCAMPI に関する報告は現在 (2005 年 7 月) までに 4 回報告されている。プロセス成熟度プロファイルには SCAMPI クラス A 実施件数と段階表現または連続表現のどちらを選択したかの情報が含まれている^[3]。プロセス成熟度プロファイルによると SCAMPI クラス A 実施件数は、2003 年 6 月から 2004 年 6 月まで半年ごとに、100 件、223 件、367 件と増加しており (図 3)、連続表現と段階表現の実施件数比率は、ほぼ同じ比率で推移している (図 4)。

図3：段階表現と連続表現の件数^[3]図4：段階表現と連続表現の比率^[3]

4. CMMI 段階表現の問題点と連続表現への期待

SW-CMM を用いた改善活動の経験から、段階表現では次のような点を問題と認識している。

(1) 管理者などからの早期レベル到達のプレッシャーで本来すべき改善事項の優先度が下がる：

早期レベル到達のプレッシャーは、どの組織にも起こり得る問題であり、管理者または改善推進者（例：SEPG リーダ）がレベル到達へ無理に推進させることが多く、開発担当者がレベル到達を急ぐことは少ない。レベル到達のプレッシャーにより、開発担当者の改善意欲が低下し、職場の雰囲気が悪化するなどの影響も考えられる。もし、管理者がレベル到達を認証と誤解しているだけなら誤解を解くことでレベル取得の悪影響を回避できる。

連続表現においては、各プロセス領域別にレベル付け（能力レベル）が行われるため、レベルを到達したかどうか二つに一つではなく、選択したプロセス領域の数だけレベル到達・未到達が示される。7 プロセス領域中 6 プロセス領域が期待した能力レベルに達成、のような改善途中結果の報告が可能となる。段階表現では 6 プロセス領域到達という表現はなく、この場合は成熟度レベル未到達という判断が下される。プロセス領域別に判定されることがレベル到達プレッシャーから開放されると期待できる。

(2) 成熟度レベルごとにプロセス領域が割り当てられているため、組織が改善したいプロセス領域を優先的に選択できない：

段階表現では、成熟度レベルごとにプロセス領域が割り当てられており、段階的に成熟するための道筋を示している（表 3）。しかし、実際に発生している問題を解決するための改善を実施したい場合、本来改善したいプロセス領域を選択できない事態が発生する。これはレベル到達プレッシャーと関係があり、レベル到達を遅らせてでも本来改善すべきプロセス領域の優先度を上げられるかに関係する。

連続表現においては、プロセス領域を組織が考える優先順位で選択できるため、本来改善すべきプロセス領域を選択できると期待できる。

5. プロセス領域の選択

段階表現においては組織の成熟度が決まれば改善するプロセス領域は自動的に決まる（表 3）。組織の成熟度が 2 である場合、次の成熟度であるレベル 3 の 14 プロセス領域の改善に着手する。

連続表現では全 25 プロセス領域の中から自由にプロセス領域を選択できるため、組織のニーズに合致したプロセス領域を選択することが必要となる。ビジネス目標、改善活動項目、過去のトラブル事例に関するプロセス領域の関係をパレート図で分析した。段階表現においても、プロセス領域選択の分析は重要であるが、決められたプロセス領域をそのまま選択するが多い。

5.1. ビジネス目標とプロセス領域の関係

組織のビジネス目標はプロセス領域を選択する基準となり、文書化されたビジネス目標は半期毎に見直される組織方針に示されている。また、組織方針には明文化されていないが、今後のビジネスの展開を模索し、そのための準備もプロセス領域を選択する基準と成り得る。ビジネス目標とプロセス領域の対応関係を表 4 に示す。ひとつのビジネス目標は複数のプロセス領域に関係

付けられる。

表 4：ビジネス目標とプロセス領域の対応関係

ビジネス目標	対応するプロセス領域
顧客ニーズに合う製品を提供	要件管理、要件開発、検証、妥当性確認
精度の高い見積もりを行う	プロジェクト計画策定、プロジェクトの監視と制御、統合プロジェクト管理、リスク管理
技術者の教育を充実させ育成する 技術力を強化する	組織トレーニング
今までの技術や経験を継続維持し発展させる	組織プロセス重視、組織プロセス定義
規定や規則の遵守	プロセスと成果物の品質保証

5.2. トラブル事例とプロセス領域の関係

過去のトラブル事例を分析し、多く発生したトラブルに関係するプロセス領域を選択できる^[6]。

トラブル事例にプロセス領域を関係付けるには、表面的な問題に関係するプロセス領域と根本的原因に関係するプロセス領域の両方を関係付けた。よって、一つのトラブル事例には複数のプロセス領域が関係付けられている。図 5 にトラブル事例に関係するプロセス領域の割合を示す。プロジェクト計画策定 (PP) が最も多く、検証 (VER)、妥当性確認 (VAL) が続く。成熟度レベル 2 と成熟度レベル 3 のプロセス領域が混在している。

段階表現の場合、成熟度レベル 2 の改善途中なら、トラブル事例に関係する成熟度レベル 3 の検証 (VER)、妥当性確認 (VAL) は選択されないだろう。

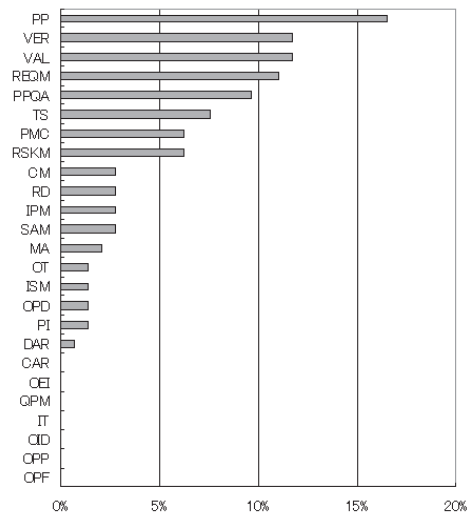


図 5：トラブル事例とプロセス領域の関係

5.3. 改善活動実績とプロセス領域の関係

2002 年より SEPG を組織して継続的に改善活動を実施しており、今までに積み重ねてきた改善活動の状態をアプライザルで確認することは有効である。SEPG が扱う改善項目とプロセス領域との対応を関係付けた (図 6)。

組織の規定整備はほぼ毎期実施しているため、組織プロセス定義 (OPD) の割合が高くなっている。プロジェクト計画策定 (PP)、プロセスと成果物の品質保証 (PPQA)、構成管理 (CM) と続き、成熟度レベル 2 と成熟度レベル 3 のプロセス領域が混在している。プロジェクト計画策定 (PP) では見積もりに関する改善事項を扱う場合が多く、プロセスと成果物の品質保証 (PPQA) では SQAG (Software Quality Assurance Group : ソフトウェア品質保証グループ) 確立を主に扱っている。

段階表現の場合、成熟度レベル 2 の改善途中なら、実際の改善活動に関係する成熟度レベル 3 の組織プロセス定義 (OPD)、組織プロセス重視 (OPF) などは選択されないだろう。

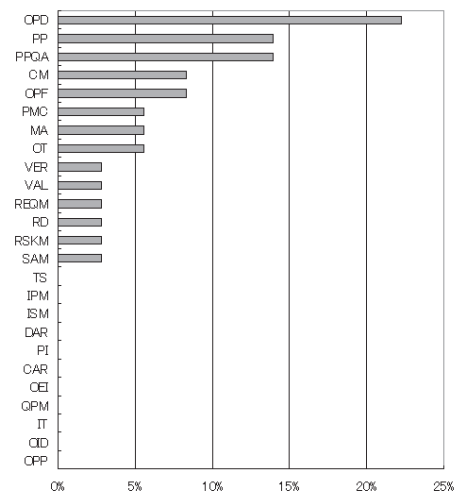


図 6：改善活動とプロセス領域

5.4. 選択したプロセス領域の適応率

実際のアプレイザルで選択されたプロセス領域は、プロジェクト計画策定 (PP)、プロジェクトの監視と制御 (PMC)、プロセスと成果物の品質保証 (PPQA)、測定と分析 (MA)、要件管理 (REQM)、要件開発 (RD)、リスク管理 (RSKM)、検証 (VER)、妥当性確認 (VAL) の 9 プロセス領域である。選択された 9 プロセス領域について、ビジネス目標、改善活動実績、トラブル事例に関係するパレート図の累積値 (プロセス領域の適応率) を図 7 に示す。

ビジネス目標の適応率は約 60%、改善活動実績の適応率は約 50%、トラブル事例の適応率は約 75%である。段階表現の場合、成熟度レベル 2 の改善途中なら、ビジネス目標 27%、改善活動実施 54%、トラブル事例 52%の適応率となる。

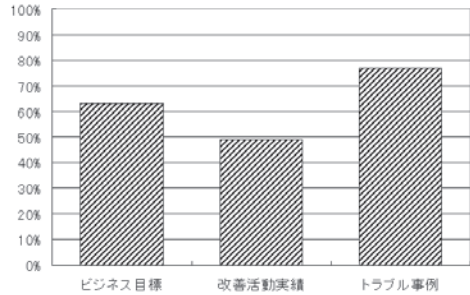


図 7: プロセス領域の適応率

6. 考察

6.1. 組織ニーズとプロセス領域

今回のアプレイザルではトラブル事例に関係するプロセス領域が多く選択されたことがわかり、直面している問題を解決するための手がかりとしたい意図が感じられる。一方、改善活動実績が約 50%に留まったのは、組織プロセス定義 (OPD) プロセス領域を選択しなかったためである。CMMI モデルの構造上各プロセス領域には共通のプラクティス (共通プラクティス) が含まれ、共通プラクティス 3.1 (GP3.1) に「定義されたプロセスを確立する」が含まれる。そのため、各プロセス領域の共通プラクティスに組織のプロセス定義が含まれるため、今回のアプレイザルでは組織プロセス定義 (OPD) プロセス領域を選択しなかった。

トラブル事例に関係するプロセス領域は多く選択されているが、ビジネス目標や改善活動項目が無視されているわけではない。短期的視点のトラブル事例とビジネス目標の中長期的視点がバランスよく取り入れられたプロセス領域の選択となっている。

また、ビジネス目標、改善活動項目、トラブル事例のそれぞれの割合の高いプロセス領域は成熟度レベル 2 と成熟度レベル 3 のプロセス領域が混在しており、プロセス領域を組織のニーズで選択できる連続表現の方が合致する。組織として関心の強いプロセス領域を優先的に改善したいと考えるのは当然のことである。

段階表現では成熟度レベル 2 でプロジェクト管理プロセス領域を改善し、次に成熟度レベル 3 でエンジニアリングプロセス領域を取り入れる順序が推奨されている。連続表現において組織の考える優先順序でプロセス領域を選択した場合でも、段階表現の推奨する順序とかけ離れたプロセス領域を選択しないことが確認できた。

6.2. レベル到達のプレッシャー

今回のアプレイザルではレベル到達へのプレッシャーが、アプレイザル準備段階から終了まで存在しなかった。そのため、今後の改善事項へ焦点が集まり、アプレイザル中においてもアプレイザルメンバは惜しみなく弱み (改善点や実装不十分な事項) を出し合い、改善のためのアプレイザルとなった。アプレイザル対象組織メンバが、成熟度レベルに到達すると安心してしまうため、あえてレベル到達させないために弱みを出そうとする場面さえあった。当然、アプレイザルではドキュメントレビューやインタビューで得られた観察事項に従って適切に判断するため、故意にレベル到達を操作することはできない。もしアプレイザル対象組織メンバがレベル到達のプレッシャーを少しでも受けていたら、積極的に弱みを多く抽出することは考えられない。このように弱みを多く抽出する雰囲気は連続表現の特徴と考えられ、段階表現のアプレイザル (経験上 CBA-IPI) ではありえない。

選択した 9 プロセス領域は成熟度レベルを重視していないため、成熟度レベルのレイティング

が実施できないプロセス領域の選択となっている。具体的には、成熟度レベル2の構成管理 (CM) と供給者合意管理 (SAM) が選択されなかったため、成熟度レベル2のレーティングは行えない。そのため、成熟度レベル到達のプレッシャーは皆無であった。つまり、アプレイザル準備段階から成熟度レベル到達は考慮せず、組織ニーズを重視していることがわかる。

また、能力レベル到達の宣伝効果は低く、「要件管理プロセス領域は能力レベル3ですが、要件開発と技術解は能力レベル1です。」といった宣伝に意味はない。能力レベル到達の宣伝効果が低いため、改善結果の効果 (例：QCDの改善) を宣伝するため本質的な改善に着目すると期待できる。目標が能力レベルの到達ではなく本質的な改善であれば、能力レベル到達のプレッシャーも低くなると考えられる。

6.3. レベル到達のプレッシャーとレベル到達の達成感

段階表現を用いたアプレイザルの結果として、成熟度レベル到達か非到達かの結果が出される。そのため、レベル到達への期待感は大きく、改善活動の集大成ともいえる一大イベントとなる。レベル到達が確認されたら、関係者でレベル到達パーティーを開催し、客先などへ宣伝、新聞発表や講演会などが実施されるかもしれない。レベル非到達の場合、これらの構想が妄想に終わり、レベル未到達のショックと関係者の期待に応えられなかった信頼喪失に悩まされる。

一方、能力レベルにおいては、プロセス領域別に到達レベルが示されるため、さまざまな組合せの結果が考えられる。そのため、連続表現においては能力レベル到達の達成感は成熟度レベル到達の達成感より低くなる。今回実施のアプレイザルにおいては、能力レベル到達の達成感は実際に低く、逆に能力レベル非到達のショック (悪影響) も少なかった。

レベル到達を推進する側 (管理者や SEPG など) とレベル到達の活動を実施する側 (開発者など) のレベル到達の達成感には、レベル到達プレッシャーとの相関関係がある (図8)。活動推進側は、レベル到達のプレッシャーが無くてもある程度以上の達成感を得られ、プレッシャーが大きいほど達成感は高くなる。一方、活動実施側は、レベル到達のプレッシャーがなく知らないうちにレベル到達していた場合は達成感を得られず、適度なプレッシャーの結果にレベル到達することが、最も達成感を得られる。レベル到達プレッシャーが大きくなるとやらされ感が増し、アライの活動を実施すれば達成感を得られない。そもそもレベル到達の達成感を得たいということが、間違いなのかもしれない。

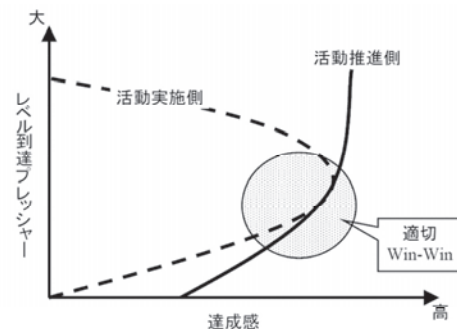


図8：プレッシャーと達成感の相関概念図

6.4. 連続表現を段階表現に変換した場合

連続表現は段階表現に変換することができるため、連続表現を選択したとしても成熟度レベルで示すことが可能である。この場合、連続表現を用いても段階表現での問題点と同じことが起こり得る。成熟度レベル到達のプレッシャーも起こり得るが、能力レベルを第1と考えることが定着していれば、成熟度レベル到達の関心は低くなる。成熟度レベルへの関心はゼロにはならないが、次の成熟度レベルへ到達するまでの中間を能力レベルの状態を確認できるため、成熟度レベルの到達だけを重視しなくてもよくなる。

改善の途中経過を確認できることが重要で、その役目を連続表現の能力レベルで実現できる。

6.5. その他の成功要因

連続表現を用いた利点を述べてきたが、連続表現の利点を活用できた理由を次に挙げる。

(1) 改善活動の推進者である管理者の推進力が強力であったことが挙げられ、毎週の SEPG 会議には管理者が毎回参加している。SEPG 会議での決定事項は、管理者自らが推進し、SEPG の決定事項に適切な権限を与えている。

(2) アプレイザルメンバの活動に報奨を与え、開発業務だけでなく改善活動の重要性を示した。

(3) SW-CMM を参照したことはあったがモデルに依存しない改善を継続的に実施していたため、CMMI モデルが導入でき、連続表現が受け入れられた。

(4) SEPG メンバがプロジェクトリーダーや課長、品質保証課担当者など組織全体から選出されたメンバで構成されているため、組織横断的な改善が実施できる環境があった。

(5) 新し物好きの組織で新技術や新手法を取り入れやすい雰囲気があった。

(6) 組織改革で異なる組織風土の課が集まったため、連続表現で課別にプロセス領域を選択できることが適していた。

このようにモデルや表現形式の選択だけでなく、改善活動を推進できる環境が整えられていたため連続表現の利点が生かせ、モデルに頼らず現実に起こっている問題を解決する参考情報としてモデルを用い、健康診断としてアプレイザルを実施できた。このような環境はどのような組織であっても必要と考えられる。

7. まとめ

段階表現ではレベル到達へのプレッシャーで本質的な改善ができない問題や、成熟度レベルごとにプロセス領域が割り当てられているため、組織の考える優先度でプロセス領域が選択できない問題があった。連続表現ではプロセス領域別に能力レベルが示され、成熟度レベル到達へのプレッシャーが軽減され、組織ニーズに合った優先順位でプロセス領域を選択できる。連続表現は、段階表現での問題点を解決するのに有効であることが確認できた。

レベル到達へのプレッシャーが軽減されることで、担当者の改善意欲を損なうことなく改善を実施でき職場の雰囲気の悪化を招かないなどの点でも有効である。これは改善活動がやりやすくなったといえる。

連続表現ではレベル到達の達成感を得られ難いが、改善項目が多く抽出され改善の推進力が強く働くことが確認できた。

段階表現の利点である改善の順序は、連続表現でも組織の考えるプロセス領域の順序を考慮する際に参考にでき、成熟度レベルの判定が必要な場合は必要に応じて変換が可能であるため、連続表現を用いたために段階表現の利点を捨てることにはならない。段階表現の利点である成熟度レベルに興味がある場合は、段階表現で陥るレベル到達のプレッシャーの問題が発生しないような配慮は必要である。

プロセス改善を成功させるためには、モデルの表現形式選択以外に多くの要因があり、それらが相互に良く関係し合うことが必要であることを再認識した。

今回の CMMI 連続表現の利用は当社初であり、社内におけるパイロットの位置付けを考えると今後の全社展開の方針やノウハウ蓄積に重要な役割を担った。

参考文献

- [1] CMMI Product Team: CMU/SEI-2002-TR-011 ESC-TR-2002-011, Capability Maturity Model® Integration (CMMISM) Version 1.1, Continuous Representation: Software Engineering Institute, Carnegie Mellon University: 2002/3
- [2] CMMI 成果物チーム著、JASPIC CMMI 翻訳分科会訳：CMU/SEI-2002-TR-011 ESC-TR-2002-011、能力成熟度モデル®統合、(CMMISM)、1.1 版、連続表現 公式日本語版：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：2002/3
- [3] Process Maturity Profile, CMMI® v1.1 SCAMPISM v1.1 Appraisal Results : Software Engineering Institute, Carnegie Mellon University: 2003/9,2004/3,2004/8,2005/3
- [4] 田村朱麗、小笠原秀人、藤巻昇：全社的な SPI 活動推進のためのフレームワーク：日本科学技術連盟 第 21 回ソフトウェア生産における品質管理シンポジウム：2002
- [5] 丸屋宏二、小笠原 秀人：全社的な SPI 活動支援体制の構築と実践：日本科学技術連盟 第 22 回ソフトウェア生産における品質管理シンポジウム pp.149-156：2003
- [6] 原聡、笠越義和、道正一郎：CMMI 連続モデルを使ったプロセス評価／改善：日本科学技術連盟 第 23 回ソフトウェア生産における品質管理シンポジウム pp.195-202：2004

『ソフトウェアの品質に関する知識体系のガイド』 策定のご紹介

SQuBOK策定部会リーダー

岡崎 靖子 (日本アイ・ビー・エム株式会社)

1. 策定の経緯

(財)日本科学技術連盟・SPC委員会と(社)日本品質管理学会・ソフトウェア部会は共同でSQuBOK策定部会を組織し、『ソフトウェアの品質に関する知識体系のガイド (SQuBOKガイド: Guide to the Software Quality Body of Knowledge、「スクボック」と読む)』の策定を行っております。このSQuBOKガイドの策定についてご紹介します。

2005年春に飯塚悦功先生(東京大学教授)を委員長とする(財)日本科学技術連盟・SPCステアリング委員会において、ソフトウェアの品質に関する技術全般に対し客観的・網羅的な知識ベースを付与し、国内の現場で品質管理に携わる方が読み解けるソフトウェア品質に関する知識体系の必要性が議論され、予備的な検討を行いました。そして、ソフトウェアの品質に関する知識体系構

造のドラフトと第1版正式版策定までのマイルストーン、および完成後は公表する方針であることを草案として示しました(図1)。

この草案を受けて同年9月にSPCステアリング委員会とソフトウェア部会の有志10名でSQuBOK策定部会を発足し、SQuBOKガイドの策定を開始致しました。その後、両組織下に各々、SQuBOK策定委員会とSQuBOK研究会を発足して、活動の母体となる組織体制を整えました(図2)。

我々のSQuBOKガイド策定の目的は次の5つです。

- 品質保証担当者の育成に役立つものにする
- ソフトウェア品質に関する日本の暗黙知を形式知化する
- ソフトウェア品質に関する最新のテーマを整理し、体系化する
- ソフトウェア品質技術の認知度向上を図る
- ソフトウェア品質保証プロセスを確立したい組織の助けとなる

2006年8月28日現在、SQuBOK策定部会には、28名の企業の品質に関わる担当者と大学関係者および国の研究機関所属者がおり、ボランティアでSQuBOKガイドの策定に当たっています。また、大場充先生(広島市立大学)、松尾谷徹先生(デバッグ工学研究所、法政大学)、保田勝通先生(つくば国際大学)(50音順)に顧問として助言をいただいています。

図1: SQuBOKはソフトウェアの品質に関する知識のアクセス・ハブ

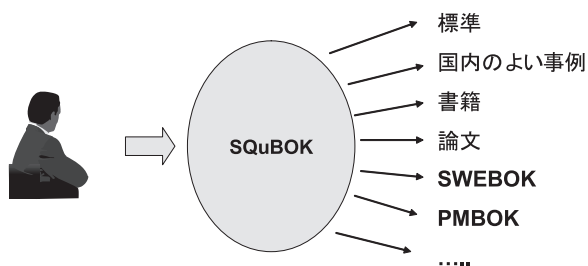
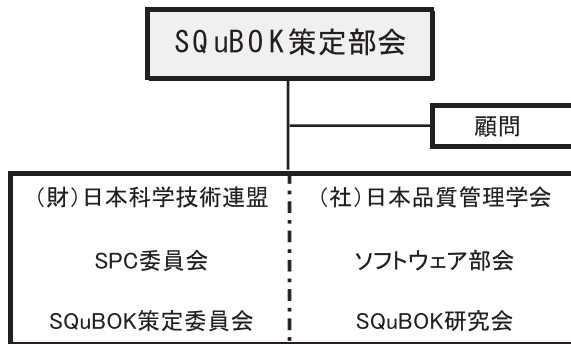


図2：SQuBOKガイドは2つの組織が協同して策定



2. SQuBOKガイド 第1版α版（有識者公開版）

SQuBOKを短期間で確実に形にしていくために、第1版では意図的に限定した範囲にスコープを絞って完成させ、第2版以降で順次内容を追加して充実していくことにしました。そこで、第1版ではソフトウェア品質の管理や評価に関するものを主に扱い、設計やコーディングなどの作りこみに関するものは第2版以降で検討します。

この第1版のα版を本年4月28日に完成させ、有識者の方々に公開してレビューをお願いしました。α版の樹形図の一部を図3にご紹介します。α版では、カテゴリ、知識領域、副知識領域、トピックスの4階層でソフトウェアの品質に関する知識体系を整理しています。図3は、カテゴリから副知識領域までの上位3階層を示した樹

形図です。

樹形図の最初のカテゴリ『ソフトウェア品質エンジニアリングの基本概念』では、品質の概念をはじめとする、ソフトウェア品質に関する基本的な概念や考え方を分類しています。このカテゴリでは、品質の概念からソフトウェア評価までの5つの知識領域と、品質保証とV&Vの2つの副知識領域を整理しています。

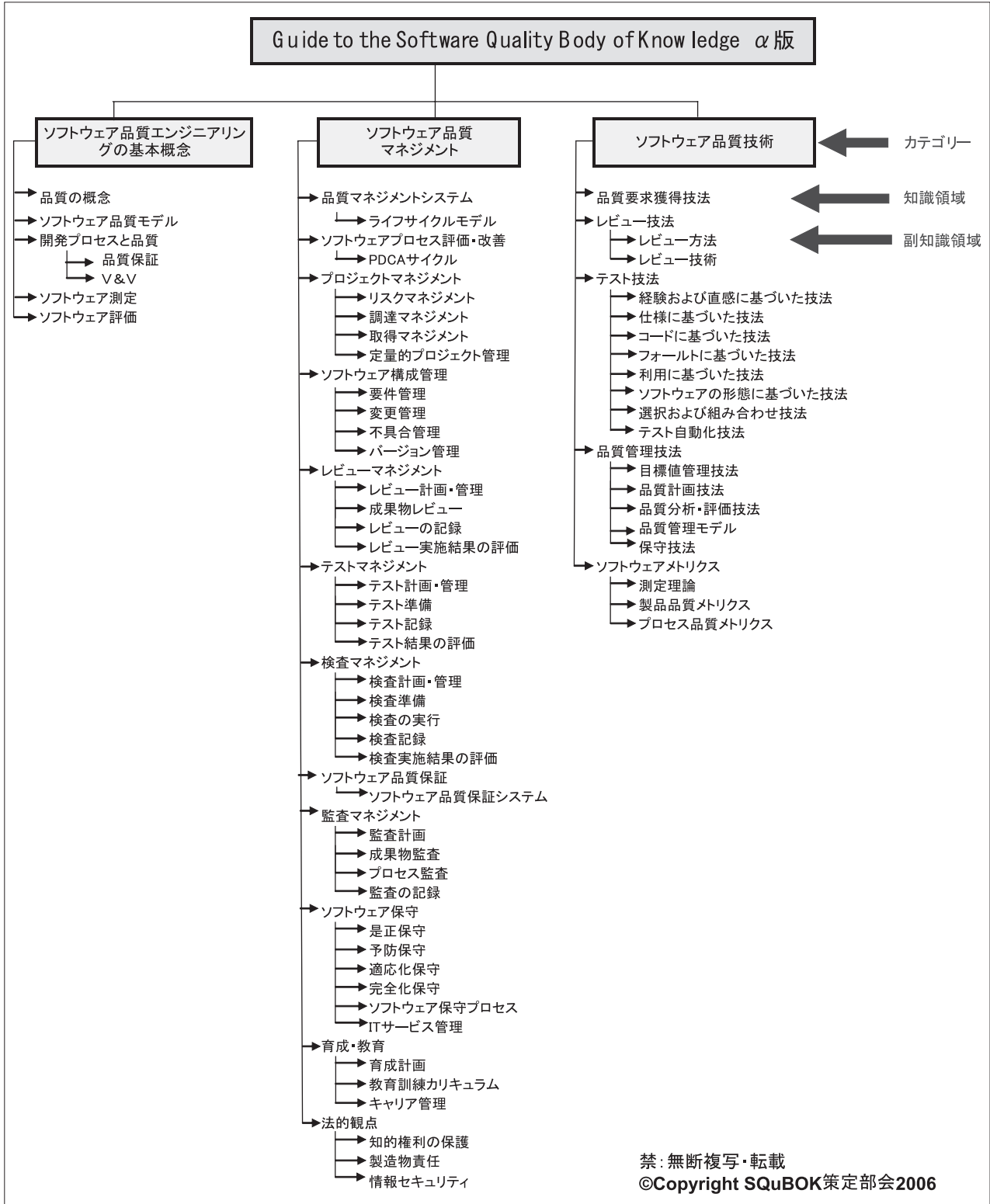
次のカテゴリ『ソフトウェア品質マネジメント』では、品質を管理するためのアクティビティを分類しています。このカテゴリでは、品質マネジメントシステムから法的観点まで12の知識領域とライフサイクルモデルから情報セキュリティまで40の副知識領域を整理しています。

最後のカテゴリ『ソフトウェア品質技術』では、ソフトウェア品質要求把握から品質評価に至るまでの具体的な手法・技法およびメトリクスを分類しています。このカテゴリでは、品質要求獲得技法からソフトウェアメトリクスまでの5つの知識領域とレビュー方法からプロセス品質メトリクスまでの18の副知識領域を整理しています。

そして、図3には示していませんが、副知識領域の下位の4階層目に約220のトピックスを挙げています。トピックスは、例えば、用語の定義、モデル、ISO/IEC・JIS・IEEEなどの規格、品質管理活動、技法などとなっています。SQuBOKガイドをご覧になる方が、このトピックス名を手がかりにしてご興味分野をより詳しく調べていただけると幸いです。

SQuBOKのご紹介

図3 SQuBOKガイド α版 カテゴリー、知識領域および副知識領域までの樹形図



(この樹形図は2006年4月28日現在のものです。内容は事前の予告なしに変更する場合があります。)



3. 今後の予定

現在当部会は α 版に対して有識者の方々から頂戴したコメントを反映しつつ、 α 版で積み残した内容も盛り込んで β 版を策定しているところです。現在策定中の β 版は、 α 版に対して下記の変更を検討しています。

(1) 品質の作りこみ関係の知識領域の位置づけを明示

前述のように、第1版では、策定期間の関係で意図的に設計やコーディングなどの品質の作りこみに関する知識領域はスコープ外としています。 α 版ではそれらは樹形図からも除外としていました。しかし、 β 版では樹形図の中にそれらの挿入場所だけは明示するように検討中です。この方が、全体的に見渡せる包括的な体系となり、品質に関する知識領域の位置づけがわかり易くなるからです。

(2) 知識領域の並べ替え

3つのカテゴリーのうち、真ん中の「ソフトウェア品質マネジメント」の知識領域が膨らみすぎたため、既存のよく知られた枠組みのいくつかを参考にして少し整理しなおそうとしています。例えば、ISO/IEC 12207の「組織に関すること」、「プロジェクトに関すること」、「支援に関すること」という大分類が流用できないか検討中です。

(3) トピックスの充実（網羅度向上、解説文付与）

前述の約220トピックスに、有識者レビューを通じてご教授いただいたトピックスや、当部会メンバが気づいたトピックスを追加中です。加えて、トピックス

についても各々に簡単な解説文を作成しようと予定しています。 α 版では樹形図のカテゴリーから副知識領域の上位3階層までに解説文を付けていましたので、 β 版では、4階層全てに解説文がつくことになります。

(4) 推奨文献リストの厳選

α 版では当部会メンバが推奨する文献を全て推奨文献リストに掲載していましたが、 β 版ではその中でも複数の部会メンバが推奨する文献に厳選して掲載しようと考えています。その準備として、現時点で複数の推奨がない文献には部会メンバを割り当て、文献の内容を吟味しているところです。なお、国内のITエンジニアが入手しやすいように、推奨文献は日本語のものを充実するように努めております。

今後、SQuBOKガイド第1版正式版までは、次のステップを予定しています。

- SQuBOKガイド第1版 α 版に対する有識者コメントを反映して β 版を策定（現在）
- SQuBOKガイド第1版 β 版を一般公開してコメントを募集
- SQuBOKガイド第1版 β 版に対するコメントを反映して正式版を作成
- SQuBOKガイド第1版正式版を一般公開

β 版は一般公開し、皆様からコメントを頂戴したいと考えています。その折には、皆様のご支援・ご協力をよろしくお願い申し上げます。

コミュニティ の 広場

S-open (ソフトウェア技術ネットワーク)

S-openはソフトウェア技術者が
気軽に参加できる相互研鑽の場を提供します。

S-openホームページ : <http://www.s-open.net>

■ ネットワーク拡大！ソフトウェアピープルの輪

志のある技術者が自主的に集い、自由な雰囲気の中で相互研鑽し、輝かしい未来を創造する…「ソフトウェア技術者ネットワーク (Software Professional Engineers' Network: S-open)」はそんなコミュニティを目指しています。

S-openでは、会員からの提案で、新しい研究グループ「SIG」を結成し、仲間を集め、明日から議論を始めることができます。S-openの研究活動は成果を形にする事がモットーです。図書や論文など、眼に見える成果が上がっています。

S-openは、常にその時代で注目される話題を、セミナー「ホットセッション」で提供しています。S-openの持っているネットワークを駆使して、著名で実力のある方々にご講演頂いています。そのコストパフォーマンスは相当高いレベルにある！と自負できます。

S-openは会員交流を何よりも大切に考えています。電子メールを利用した「コミュニティ」、「ホットセッション」出席者による「BOF」と「BOFメーリングリスト」、合宿討論会「ホットスパセッション」など、多様なスタイルのコミュニケーションチャンスをご提供しています。

■ S-openは魅力満載！

S-openはあなたの情報ネットワークを広げる身近なコミュニティです。

—交流は、SWプロフェッショナルとしての視野を広げます

—最新の技術動向を低価格で得られます

—会社を超えた人的ネットワークを構築できます

—もちろん、S-openで得たことは、仕事に活きます

S-openに是非ご参加下さい！

■ 主な活動

(ア) 研究グループ活動

—成果創出を目的とした研究グループ【SIG】～①ソフトウェアメトリクス、②モチベーション研究、③感性によるソフトウェア研究～

—ネット上の議論を中心としたサロン／テーマ別グループ【コミュニティ】

(イ) 最新情報を低価格に提供

—参加型セミナー【ホットセッション】

(ウ) 成果の発信

—成果報告書の発行。【SIG報告会】

(エ) 会員交流の場の提供

—共有テーマを会員同士で議論。【BOF】【ホットスパセッション】【メーリングリスト】

■ 2006年度 イベントスケジュール

・創立4周年記念講演「お隣のお兄さん」的Webサイトへの挑戦～前田建設ファンタジー営業部の顛末～(5/10 (水) 午後)

・第18回「大型システム構築の超上流」(7/28 (金))

・第19回「ソフトウェア再利用への挑戦」(8/29 (火)) 関西開催

・第20回「見える化(仮)」(11/6 (月) 予定)

・第21回「人材育成・モチベーション(仮)」(予定)

・第22回「SIG報告会」(2/2 (金) 予定)

・ホットスパセッション開催

・テーマ「S-Openの今後の研究テーマについて考える」(仮) 10/6 (金) ～7 (土) ラフォーレ修善寺

S⁺open

年会費：5,250円

主な活動拠点：関東・関西

お問い合わせ先：E-mail: s-open-quote_2006@s-open.net

コミュニティ の 広場

QuaSTom（高品質ソフトウェア技術交流会）

…でしたら、QuaSTomはいかがですか？

— 高品質ソフトウェア技術交流会（QuaSTom）へのお誘い —
QuaSTomは、ソフトウェアの高品質化を追求する会員が自主運営する任意団体です。
QuaSTomホームページ : <http://www.quastom.gr.jp/>

(1) すぐに現場に持ち帰れる、身近な事例研究が QuaSTom（「カスタム」と読む）の自慢です

何をそんなにお悩みですか？

ふむふむ、その品質データ、もうちょっと効率的に分析して、気の効いたまとめ方をしたいのですね？

…でしたら、QuaSTomはいかがですか？

今年度は「Kowaza」という研究テーマを設定し、会員自らがExcelの関数やマクロ、Windowsのバッチファイル等で工夫した、ちょっとしたツールや帳票を持ち寄って改善事例を紹介します。一つのツールからそれを利用する業務そのものについてまで、ディスカッションは広がります。アンケー

トでは、参加者全員に「参考になった」と好評でした。いくつかのツールはQuaSTom Madeとして会員限定で自由にご利用できます。

今年度開催した事例研究では他に、QuaSTom会員によるPSP演習実践の体験報告もありました。いつも例会で会う“仲間”の体験談に、書物や講演だけでは得られない気付きや元気を得、自らもまた実践してみたいとの感想を持たれた参加者もいました。

並行して取り上げたトピックス「ファシリテーション」も、ワークショップ形式で体験重視です。議論し体験し身体で覚えて、翌日からすぐに自らの業務に取り入れられる、そんな事例研究がQuaSTomの自慢です。





(2) 継続は力なり、分科会こそがQuaSTomの本質です

何をそんなにお悩みですか？

ふむふむ、プロジェクト管理手法や改善手法の勉強はいろいろしたけれど、いざ実務となると使いこなせない。もっと継続的に深く勉強して、タイムリーに使いこなしたいのですね？

…でしたら、QuaSTomはいかがですか？

QuaSTomでは、メンバーをある程度固定し、年間を通じて特定のテーマを検討し、現場に活用しやすい形で成果を残す分科会活動を行っています。今年度はプロジェクトマネジメント分科会、現場改善技法検討分科会の2つの分科会が活発に活動を進めています。

メンバーが固定されている事で、ある意味、安

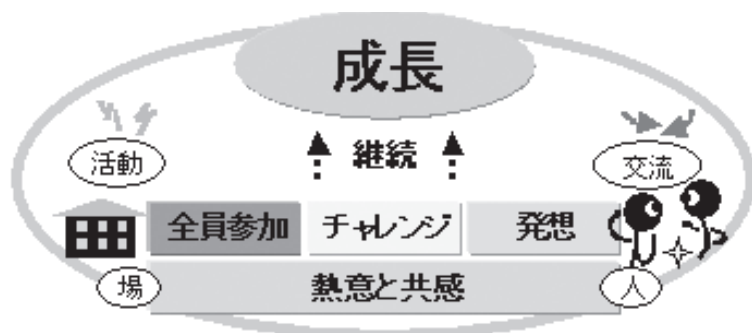
心して本音の議論もできます。継続的な本音の議論、これこそがQuaSTomの本質です。

(3) もちろん、講演会も合宿もあります

講師を招いての本格的な講演会、首都圏近郊で一泊二日の合宿研修も好評です。テーマも、見積もり、要求の仕様化、アーキテクチャ設計、ソフトウェアテスト、メトリクスによる進捗管理、等多彩です。どのようなテーマでも、必ず「技術交流会」の名前を意識し、演習や講演者を交えたディスカッションを重視したプログラムで開催しています。

どうですか？より具体的で実践的な交流を通じて、お互いを高め合い成長していく場として、QuaSTomはいかがですか？

皆様の参加をお待ちしております。



QuaSTom

現場に
ゆとりと熱意をもたらすための
ディスカッション&ネットワーク！

これがQuaSTomの活動方針です。

お問い合わせ先：E-mail：secre@quastom.gr.jp

コミュニティ の 広 場

PS 研究会

PS 研究会 紹介

PS研究会ホームページ : <http://www.ps-tb.jp>

PSの考え方

企業価値や成長性を測る尺度として従業員満足（ES：Employee Satisfaction）が良く知られています。ESは、企業の人的資源のクオリティを現していると考えられます。同じように、プロジェクトにおける人的資源のクオリティや健全性を測ることが望まれます。

プロジェクトは、特定企業の従業員だけで構成されているとは限りません。むしろ、多様な所属形態によって構成されており、ESを用いて計測することが困難です。そこで考えられたのがパートナー満足（PS：Partner Satisfaction）です。

ESは、企業の労働環境を従業員の視点で計測します。PSは、プロジェクトの労働環境をプロジェクトに参加したメンバーや関係するステークホルダの視点で計測します。メンバーやステークホルダには顧客、従業員、外注、派遣などさまざまな所属形態を含みます。このような多様な人々のことをパートナーと考えます。

PSは、プロジェクトにおける人的資源のクオリティを測ることから始まりましたが、測る目的は人的資源のクオリティを高めるためです。

PS研究会

PS研究会は、プロジェクトにおける人的資源のクオリティを高めるため、プロジェクトメンバーの働きやすい環境について、計測、モデル化、事例分析、実践など幅広い活動を行なっています。人的資源のクオリティを高めることは、結果的に

プロジェクトのパフォーマンスを高めることとなります。

PS研究会は、日科技連SPC配下において2002年から始まり、プロジェクトマネジメントと関連が深いため、PMAJ（日本プロジェクトマネジメント協会）とも連携して活動を行っております。

研究活動 FST（Field Study Team）

PS研究会の活動は、個別の研究活動とその成果を**実践**する活動に分かれています。研究活動はFSTと呼び、研究フィールドを共有しながらお互いに専門性を高めることを目的とするタスクチームに分れて活動しております。

PS研究会メンバーによる研究活動成果は、日科技連だけでなくPM学会の論文、ProMAC国際会議、PMAJシンポジウムなどで高く評価されており、この分野で大学の研究と遜色のないレベルに達しています。

現在、次の4つのタスクチームが活動しております。

- 1) セミナーFST：プロジェクトが元気になるような、研究で得たものを現場で実践してもらうことを目的とし、セミナー開発などを行っている。
- 2) メンタルヘルスFST：蓄積疲労やストレスなどプロジェクトにおけるメンタルヘルスに関する研究を行っています。産業カウンセラーが中心となって活動しています。
- 3) 普及FST：CMMを始め、さまざまなツールに

対してプロジェクトは拒否反応を示します。人間側面を考慮し役立つ普及について研究を行っています。

- 4) MM4：今年立ち上がった新しいタスクです。現場のリーダーやプロジェクトマネジャー（以下PM）を対象とし、「メンバーを元気にするPM行動、ダメにするPM行動」をテーマに研究を始めました。

PS研究会の実践活動

PS研究の成果を実践する目的で始めた活動です。

- 1) PS調査：「測れないものは制御できない」ことから、PSを計測するサービスを日科技連から提供しております。企業や部門を対象とし、PS調査を行い、専門家が分析します。分析は課題を明らかにするとともに改善提案を行います。
- 2) PSセミナー：現場のリーダーとして、人間側面から仕事を進めるために必要な理論とノウハウを学ぶセミナーを日科技連から提供しています。2日コースと、1日コースがあります。現在募集中のものは次の2つです。
「PS（パートナー満足）とTB（チームビルディング）」
・2日コース '07年1月18日－19日
・1日コース '06年11月28日
(1日コースのみPMAJと共催 PDU対象コース)

- 3) PSシンポジウム：PS活動の成果を知ってもらうため、07年2月15日－16日に初回のシンポジウムを計画しております。PSを導入した企業の成果や、FSTでの研究成果を中心に発表いたします。

PS研究会への参加について

PS研究は、参加費を払って情報を得る場ではありません。PS活動に共感した方が、フィールドを共有しながら、研究し実践する場です。研究はフィールドを共有しますが、共同研究に限定せず、単独の研究でもかまいません。実践の中心は、参加者が属する企業や組織に役立つ形で行います。

<現在募集中のタスク：MM4>

プロジェクトの活性化をメンバーの意欲の視点でとらえ、PMとしてどんな行動が効果的かについて研究します。経験豊かなPMから、人間側面のプロジェクト運営に不安を持っているPM、問題意識を持っているスタッフの方まで、幅広い参加を期待しております。

☆募集人数 5名程度 連絡先：メールにて連絡ください

PS@ps-tb.jp

☆既存のFSTに参加希望がありましたら、同様にご連絡ください。



ソフトウェア品質に関する国際規格の紹介

東洋大学
経営学部 助教授

野中 誠

ソフトウェア品質に関する国際標準（global standard）は、ISO¹とIEC²の合同技術委員会によって標準化されています。本稿では、それらの国際規格と技術報告書³の概要を紹介します。

国際規格を利用するメリット

ソフトウェア品質に関する国際規格に頼らなくとも、各社各様、ソフトウェア品質の確保・向上のために様々な取り組みを積み重ねていることでしょう。そのような経験に長けた組織においてさえも、国際規格は次のような点で有用です。

概念および用語の統一化：品質評価者によってソフトウェア品質特性の概念理解に違いがあったり、用語を異なる意味で用いたりしているのでは、複数のソフトウェアを同一の基準で評価できません。また、“方言”とも呼ぶべき組織特有の用語の使用は、協業企業とのコミュニケーションに支障を来しかねません。国際規格の利用は、品質に関する概念を正しく理解し、メン

バー間で共有し、評価基準の客観性を高めるのに貢献します。

包括的・重点的なアプローチ：自組織で実施している品質確保・向上のための取り組みが、ソフトウェア品質に対して全方位的にアプローチしていることを確認するためには、その規範としての国際規格が必要です。これまで未検討だったが重要な特性、自社製品にとって重視すべき品質特性を識別するにあたって国際規格が役立つでしょう。このような利用方法は、自社製品の強みを一層強化することに結びつきます。

品質評価技術の客観性向上：ソフトウェア品質の改善には、その第一歩として客観的な測定・評価が必要です。そのため、ソフトウェア品質を測定・評価する技術には、再現性および客観性の高さが要求されます。測定者が異なるだけで測定手順や評価結果にブレがあるようでは、信頼に足る品質評価は不可能です。国際規格は、客観性を重視して長年にわたり議論されていることもあり、国際規格の知見を借りることで客観性の高いソフトウェア品質評価が期待できます。

¹International Organization for Standardization（国際標準化機構）

²International Electrotechnical Commission（国際電気標準会議）

³国際規格（IS: International Standard）が公的な規格であるのに対して、技術報告書（TR: Technical Report）はISO/IECによって作成・発行された参考出版物という位置づけです。

ソフトウェア品質に関する国際規格と技術報告書

本稿の最後に記載した表1に、ソフトウェア品質に関する国際規格および技術報告書のうち、現時点で一連のシリーズがすべて発行された文書の一覧を示します。規格番号にTRと付いているものは技術報告書を表します。

9126群と14598群は、いずれも1991年に発行された国際規格ISO/IEC 9126を起源としています。この規格が、ソフトウェア品質モデルとメトリクスに関する9126群と、ソフトウェア品質評価プロセスに関する14598群に分岐しました。しかし現在、これらの規格はSQuaRE (Software Quality Requirements and Evaluation) と呼ばれるISO/IEC 25000シリーズに再び一本化されており、策定作業が進行中です。なお、SQuaREの全体的ガイドを述べた国際規格ISO/IEC 25000は、2005年に出版済みです。

表1に挙げた文書はすべて日本語版が存在します。9126-1はJIS X 0129-1、14598群はJIS X 0133群、15939はJIS X 0141として規格化されています。また、TR 9126-2からTR 9126-4はすべて対訳版が発行されています。

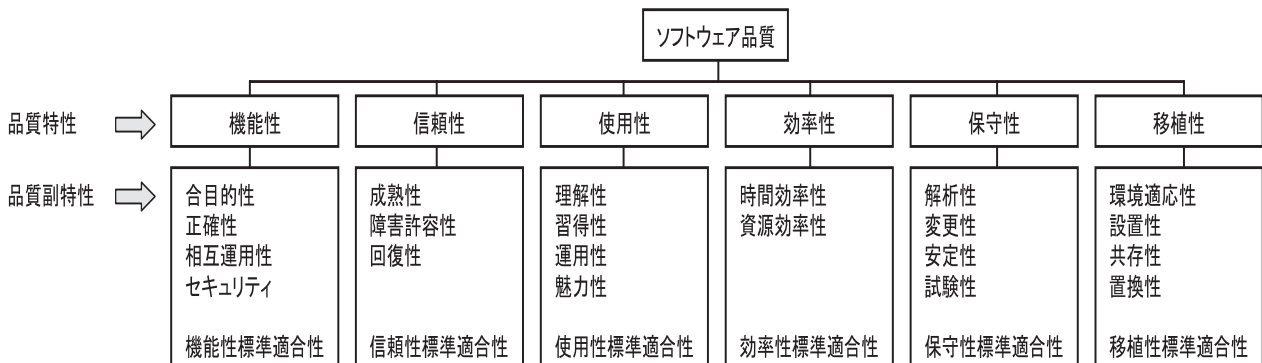
以降では、表1の分類別に概要を紹介いたします。

ソフトウェア品質モデル

ソフトウェア製品の品質モデル⁴は、図1に示した通り、6つの品質特性と細分化された品質副特性によって表されます。企業活動や社会への影響度の大きさから、多くの場合は信頼性が最も重要視されますが、ソフトウェア品質を構成する要素は信頼性だけではないということを理解しなければなりません。機能性を例にとれば、いかに信頼性の高いソフトウェアであっても、顧客や利用者の目的に即した機能が提供されていないソフトウェアは“品質が悪い”ことになります。

ISO/IEC 9126-1で規定されたソフトウェア製品の品質モデルは、世界的によく知られたモデルであり、長年の議論によって得られた国際的な合意の産物です（残念ながら、科学的な整合性の追究による産物）。品質モデルに関する議論は1970年代から数多く行われており、今なお、このモデルの不備や曖昧さを主張する論文が発表されることもあります。しかし、改版によって細かな変更があったものの、1991年のモデルからその構造はほとんど変わっておらず、安定した国際規格と言えます。

図1 ソフトウェア製品の品質モデル（その他、各特性には標準適合性という副特性が含まれる）



⁴ISO/IEC 9126-1では、“ソフトウェア製品の品質モデル”に加えて“利用時の品質モデル”も規定しています。本稿では“利用時の品質”に関する議論は省略します。

品質メトリクス

ソフトウェア品質モデルの各副特性について品質を測定する具体的方法として、TR 9126-2には外部メトリクス、TR 9126-3には内部メトリクスが例示されています。これらは技術報告書なので、あくまで例示の範囲に過ぎません。

ここで、外部と内部の違いについて簡単に説明します。図2に、評価対象ソフトウェアの状態別に、TR 9126群の品質メトリクスおよび直接の測定対象との関係を図示します。

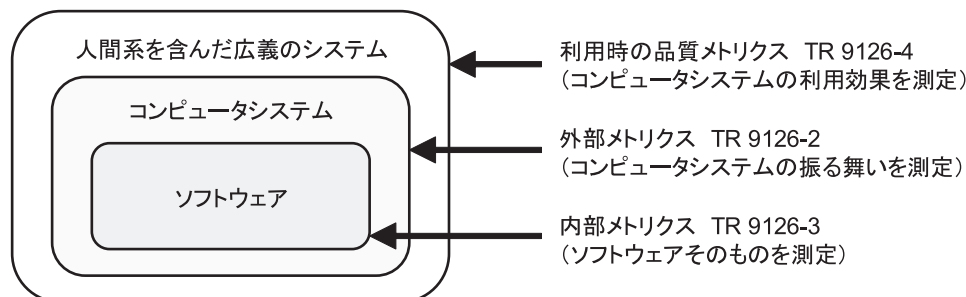
内部メトリクスは、ソフトウェアそのものを測定するメトリクスであり、ソフトウェアを測定静的なものです。コード行数や、複雑度を表すサイクロマティック数などは内部メトリクスに相当します。ソフトウェア品質モデルに関するメトリクスの例として、レビューでの欠陥除

去率が挙げられます。これは、信頼性の副特性である成熟性について、開発プロセスのデータを用いることでソフトウェアの内部品質を間接的に測定（または推定）していると言えます。

一方、外部メトリクスは、ソフトウェアを動作させそのシステムとしての振る舞いを測定します。一例として、テスト段階で用いるバグ密度というメトリクスは、成熟性について外部品質（ソフトウェア実行時の品質）を測定しています。

また、TR 9126-4では、利用時の品質を測定するメトリクスの例が示されています。これらは、利用者が特定の利用状況においてソフトウェアを利用した際に、どれだけ有効に、生産的に、満足度を持って目的が達成できたかを測定するメトリクスです。利用者の振る舞いや作業の成果が直接の測定対象になりますが、その性質上、利用状況によって測定結果が異なります。

図2 ISO/IEC TR 9126群と評価対象ソフトウェアとの関係



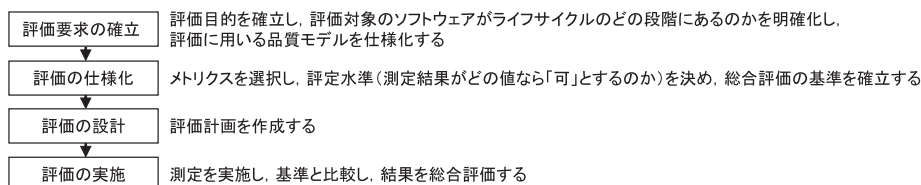
品質評価プロセス

9126群ではソフトウェア品質の測定技術を扱っていますが、14598群では品質を評価するプロセスについて規定しています。評価プロセスは、開発者、取得者、第三者機関の評価者など立場によって異なるため、それぞれのプロセスを14598-3から14598-5に規定しています。いずれの場合にも共通する評価プロセスとして、14598-1では図3に示した4ステップを共通の枠組みとして規定し

ています。

14598-3から14598-5までが具体的な評価手順に言及しているのに対して、14598-2および14598-6は全社的または部門レベルにおける評価支援について規定しています。14598-2では、品質評価に責任を持つ支援組織が、品質評価の計画と管理について実施すべき事項を規定しています。14598-6では、評価技術をパッケージ化したもの（評価モジュール）を文書化する際の構造と内容を規定しています。

図3 ソフトウェア品質評価プロセス

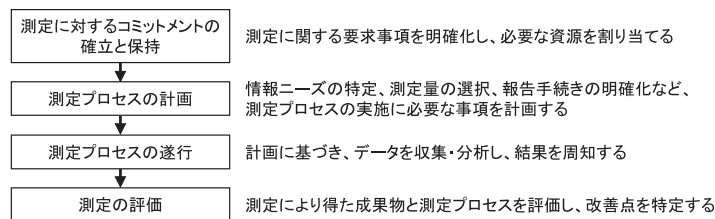


これらの規格に記述された内容の多くは、1970年代頃から日本の品質管理をソフトウェアに適用し続けてきた組織にとっては、いわば当たり前のことに思えるかもしれませんが、ソフトウェアおよびその周辺技術の急激な変化のために、かつて構築した品質評価制度が崩壊してしまった組織や、品質評価体制が不十分ないわゆる成熟度の低い組織にとっては、学ぶべきところが大いにあることでしょう。

品質測定プロセス

品質に限らず、ソフトウェア測定全般に関するプロセスと、測定に関連した用語の定義を規定した国際規格が15939です。図4に、測定プロセスの概要を示します。15939に規定された測定プロセスは、ソフトウェア品質の測定にも適用可能です。ソフトウェア品質測定技術の客観性をより高いレベルに引き上げるには、15939は大いに参考になることでしょう。

図4 ソフトウェア測定プロセスの概要



測定に関する用語について特筆すべきことは、この規格の中では“メトリクス”という用語は一切使われていないことです。メトリクスの代わりに“測定量 (measure)”という用語が使われています。メトリクスという用語は、1999年の14598-1の時点では測定方法 (measurement method) と測定尺度 (scale) をセットにしたものとして定義されています。しかし、metricという語は元来“メートル法の”という意味しか持たないため、メトリクスという語の曖昧さについて過去に何度も議論が起きています。とはいえ、産業界においても学术论文の世界でもすでに“手垢のついた”言葉であるため、“測定量”という日本語になかなか馴染めないのが筆者の正直なところでは。

SQuaRE (ISO/IEC 25000) シリーズ

以上の内容から分かる通り、9126群と14598群は、ソフトウェア品質の測定・評価技術について包括的な指針を与えています。しかし、規格の番号体系が異なる、全体として各規格の位置づけが分かりにくいなどの理由から、SQuaREシリーズとして一本化するべく標準化作業が行われています。

SQuaREシリーズでは、合計14件の規格群を次の5つの区分で体系的に整理しています。

品質マネジメント (25000, 25001)

品質モデル (25010)

品質測定 (25020, 25021, 25022, 25023, 25024)

品質要求 (25030)

品質評価 (25040, 25041, 25042, 25043, 25044)

表1と対比すると、品質要求という要素が新たに追加されたことがわかります。そのほかにも、基本メトリクス (base metrics) と呼ばれる、外部および内部メトリクスに共通して用いられるプリミティブなメトリクスが、品質測定の区分に新たな要素として加えられています。さらに、15939の測定プロセスとの連携もSQuaREでは図られています。

おわりに

ソフトウェア品質に関する国際標準は、1985年から策定作業が開始され、20年が経過した現在に至っても、ゆっくりとはいえ着実に進化しています。客観性の高い品質評価技術を目指しているものの、科学的な整合性の追究というより合意の産物としての側面が、完全に属人性が排除された客観的な評価技術と言えるかは疑問の余地もあります。しかし、非科学的だから受け入れないとする姿勢よりは、知恵と経験の詰まった国際規格を活用することを考えた方が、とりわけ産業界においては生産的な結果をもたらすことでしょう。

ISO 9000シリーズや米国カーネギーメロン大学のCMMIなどに比べると適用事例の報告が圧倒的に少ないものの、本稿で紹介した国際規格の適用事例は着実に積

み重ねられつつあります。これらの規格を様々なタイプのソフトウェアに適用した場合の考察や、プロセス改善の過程において品質評価・改善をどのような手順で取り入れるか等、研究論文や事例報告などが国際会議の場でしばしば報告されています。特にソフトウェア品質モデルは、要求分析フェーズにおける非機能要求の抽出において、品質特性の観点が活用されている事例が多数あります。

短期間の小規模プロジェクトや人数の少ない組織には、これだけ重厚な品質測定・評価技術を適用するのは現実的ではないとの声も聞こえてきます。しかし、ソフトウェアの品質に関する社会的関心が高まり、ベンダー

としての責任が問われる中、長年の蓄積の上に成り立つ品質測定・評価技術を少しずつでも取り入れることは、品質向上に向けた確実な一歩となるでしょう。そして何よりも、高品質のソフトウェアを世の中へ確実に送り出す組織体制を作り上げ、その制度を継続的に改善し続けることが、スタッフ部門およびソフトウェア技術者の「誇り」に結びつくことを。

国際規格の日本語訳（翻訳JISおよび対訳版）は、日本工業標準調査会（JISC, <http://www.jsa.or.jp>）および日本規格協会（JSA, <http://www.jsa.or.jp>）のwebサイトにて検索・入手できます。

表1 ソフトウェア品質に関する国際規格と技術報告書の一覧

分類	規格番号（発行年）	主な内容
品質モデル	9126-1（2001）	ソフトウェア製品の品質特性および品質副特性を定義
品質メトリクス	TR 9126-2（2003）	外部品質を測定する外部メトリクスの例
	TR 9126-3（2003）	内部品質を測定する内部メトリクスの例
	TR 9126-4（2004）	利用時の品質を測定するメトリクスの例
品質評価プロセス および 品質マネジメント	14598-1（1999）	ソフトウェア品質評価に関する全体的な内容
	14598-2（2000）	計画と管理に関する事項
	14598-3（2000）	開発者のための評価プロセス
	14598-4（1999）	取得者のための評価プロセス
	14598-5（1998）	第三者機関などの評価者のための評価プロセス
	14598-6（2001）	評価モジュールの文書化に関する事項
（品質）測定プロセス	15939（2002）	（品質に限らず）ソフトウェアを測定するプロセス

SPC活動のご紹介

1.1 財団法人日本科学技術連盟

財団法人 日本科学技術連盟（以下、日科技連という）は、1946年5月1日に設立された科学技術団体で、1962年には急速な社会の進歩に対応すべく科学技術庁（現 文部科学省）所管の財団法人として組織を整備し、今日にいたっています。

日科技連は創立以来、日進月歩の科学技術を企業経営に応用する諸活動を通じ産業界に寄与することを基本方針として、これを実現するために各種の事業を推進しています。多数の会員企業をはじめ科学者・技術者の協力により順調な発展を遂げ、わが国産業界の製品とサービスの品質・信頼性の向上と科学的経営基盤の強化に寄与しています。

今日、日科技連では調査・研究・開発、教育訓練、研究集会、技術相談、国際協力、広報・出版、エクステンションサービスの7つのカテゴリーをもって各種の事業を展開し、科学技術と経営管理技術の普及・進歩・発展に努めています。特に品質管理を中心とする経営管理技術については日科技連の創設直後から研究開発、教育、普及の活動を開始し、以来これらの活動を主柱とした事業を展開しており、その成果は国内はもとより世界各国から注目を集めるにいたっています。

1951年に創設された「デミング賞」は、2005年までに延べ188社がデミング賞実施賞を受賞されています。デミング賞実施賞を受賞し、受賞年度を含め5年以上経過した企業を対象とした「日本品質管理賞」は延べ19社が

受賞されています。

1962年に始められたQCサークル活動は第二次産業のみならず第三次産業にまで普及し、2006年8月現在QCサークル本部に433,166（累積）のサークルが登録され、全国で活発な活動が展開されています。また、QCサークル活動は、今や世界128カ国地域以上で実施されています。

1. SPC事業の趣旨

「ソフトウェア生産管理研究委員会」（現在、「SPC（ソフトウェア品質）委員会」）は、わが国におけるソフトウェア製品の品質向上と効果的開発の方法論の確立を目指して、日本の品質管理をソフトウェア生産に適用するための調査・研究・普及を行うことを目的として、1980年に日科技連に設置された研究委員会です。

1.2 SPCの設置

日本における近代的品質管理は1949年に始まりました。その当初は主に重化学産業で適用され、1960年代以降には機械・電気の組立産業や自動車産業に拡大し大きな成果を上げました。そして1970年代の半ばごろより、いわゆる非製造業にも適用されるようになりました。非製造業で初めて日本のTQMに興味を持ったのは建設業です。その後サービス産業、ソフトウェア産業などがTQMを導入するようになりました。

日本のTQMの普及において常に中心的な役割を果た



してきた日科技連は、TQM適用の拡大に応じて各種研究会の設置、シンポジウム開催、セミナー開催などの活動を拡大してきました。ソフトウェア産業へのTQMの適用の動きに合わせて、ソフトウェア産業へのTQMの適用を目的とする研究会の設置が提案され、1980年にSPC研究委員会が設置されました。

本研究委員会を設置するにあたって掲げられた方針は、以下のようなものです。

- (1) ソフトウェア産業への日本的TQMの考え方、方法論、手法の普及を行い、TQMとソフトウェア工学の「結婚」による新たなパラダイムの構築をめざす。
- (2) この目的のために、ソフトウェアに関わる全ての関係者（産・官・学）に相互研鑽の場を提供する。
- (3) 我が国のソフトウェア産業の健全な発展を第一義とし、ソフトウェア・メーカ主導で運営する。

1.3 SPC活動における3つの視点

上述したようなアイデンティティを意識してSPC活動を実施していくにあたって、私たちは以下の3つの視点を常にこころにとめたいと考えています。

- 1) 実践
- 2) 普及・啓蒙
- 3) ソフトウェア品質

「実践」とは、机上の空論や無用な理論の精緻化ではなく、実践に基づく「ソフトウェア品質学」の体系化を

めざすということです。現実世界に置いて発生する問題やニーズに真正面から取り組み、実践の場に置いて役に立つ提案をしていきたいと思います。その積み重ねが、結果として、ソフトウェア品質の維持と向上のための体系化につながるだろうと考え行動していきたいと思います。それが真の意味での実学であろうと信じます。

「普及・啓蒙」とは、ソフトウェア科学、ソフトウェア工学の先駆的な研究よりは、そのような分野での開発された理論の適用研究、ソフトウェア開発・保守の現場への新しい思想・方法論・技法の普及、ソフトウェア分野の多く「普通の」同士への啓蒙を重視するという意味です。先端的な理論の開発というのは魅力的な言葉ですが、そのような役割は学会が担うべきであって、私たちが力点を置くべきは、実用化前の理論というよりはむしろ、実践的でごく一般のソフトウェア関係者が入手し使用できるような理論を「普及・啓蒙」することであろうと考えています。

「ソフトウェア品質」とは、ソフトウェア品質の面に特に重点をおいていることを意味します。

高い品質こそがソフトウェアの発展を推進し、それは我が国だけではなく、世界のソフトウェアの発展に貢献できるものと確信しています。そのためにこそ、あらゆるソフトウェア業界に従事する関係者が主導する形で、ソフトウェアの発展に寄与する実学的研究を進めたいと考えています。

ソフトウェア品質関連ニュース

本コーナーでは、ソフトウェア品質に関連した、読者の皆さまに有益な情報をご案内いたします。



JaSST (ソフトウェアテストシンポジウム)

URL : <http://www.jasst.jp/>

今、ソフトウェア品質への関心の高まりとともにソフトウェアテストが脚光を浴びています。

ソフトウェアテストシンポジウム実行委員会 [主催：特定非営利活動法人ソフトウェアテスト技術振興協会 (ASTER：アスター)] は、2003年より「JaSST (ソフトウェアテストシンポジウム)」を開催してきました。

本シンポジウムは、ソフトウェアテスト分野の最新の研究発表、実践事例にとどまらず、ツールの適用事例や活用ノウハウ、ソフトウェア品質マネジメント関連トピックなど幅広い情報を得る絶好の機会です。

ソフトウェアテストに興味を持つソフトウェア開発技術者同士のネットワーキングの場としても是非ご活用ください。



JaSST'06 (ソフトウェアテストシンポジウム2006) 札幌

いよいよ今年、北の大地 北海道に「JaSST」が上陸します！

【開催概要】

開催日：2006年10月17日 (火)

場所：小樽商科大学 札幌サテライト
(北海道・札幌)

主催：特定非営利活動法人 ソフトウェアテスト技術振興協会 (ASTER)
JaSST'06 in Sapporo 実行委員会

共催：小樽商科大学

基調講演：山浦 恒央氏 (東海大学)



JaSST'07 (ソフトウェアテストシンポジウム2007) 東京

【開催概要】

開催日：2007年1月30日 (火) ~ 31日 (水)

場所：目黒雅叙園 (東京・目黒)

主催：特定非営利活動法人 ソフトウェアテスト技術振興協会 (ASTER)
JaSST'07 in Tokyo 実行委員会

お問い合わせ先 特定非営利活動法人ソフトウェアテスト技術振興協会 (ASTER) 事務局
〒108-0014 東京都港区芝5-3-7 三田奥山ビル4F 株式会社クラフトワン内
TEL：03-5444-7601 FAX：03-5444-8095 E-mail：jasst@blues.se.ucc.ac.jp

日科技連 教育・訓練活動のご紹介

日科技連では、SPC活動のほか、TQMやQCサークル、信頼性などに関する教育・訓練活動を積極的に展開しています。

掲載しているセミナーは一部のセミナーです。

日科技連ホームページ <http://www.juse.or.jp/>

【TQM・品質管理】 → <http://www.juse.or.jp/tqm/index.html>

- ・ 役員のための品質経営セミナー
- ・ 品質経営セミナー部課長コース
- ・ 品質管理セミナー 係長・主任コース
- ・ TQMセミナー入門コース
- ・ 方針管理セミナー実践コース
- ・ 品質トラブルの未然防止セミナー
- ・ 品質管理セミナー ベーシックコース
- ・ 品質管理セミナー入門コース
- ・ 品質機能展開セミナー 入門コース
- ・ 品質機能展開セミナー 実践コース
- ・ 通信教育 品質管理基礎講座

【QCサークル】 → <http://www.juse.or.jp/qc/index.html>

- ・ 品質改善のための問題解決力実践コース
- ・ 管理者のためのQCサークル指導・支援力強化コース
- ・ QCサークル推進者のための指導・支援実践力強化コース
- ・ QCサークルリーダーのための総合力養成コース
- ・ 品質トラブル防止のためのヒューマンエラー防止策と目で見える管理の実践セミナー

【信頼性】 → <http://www.juse.or.jp/reliability/index.html>

- ・ 信頼性セミナー基礎コース
- ・ 信頼性セミナー入門コース
- ・ 信頼性セミナー初級1日コース
- ・ 信頼性技法実践講座 FMEA・FTA
- ・ 信頼性技法実践講座 デザインレビュー

【応用統計】 → <http://www.juse.or.jp/statistics/index.html>

- 実験計画法セミナー実践コース
- 実験計画法セミナー入門コース

【医薬応用・医薬研修】 → <http://www.juse.or.jp/medical/index.html>

- 臨床試験セミナー 統計手法コース
- 臨床試験セミナー入門コース

【ソフトウェア品質】 → <http://www.juse.or.jp/software/index.html>

- セミナー —
- 個々のソフトウェア生産技術・品質管理技術を磨ける場！
- ・ ソフトウェア品質マネジメントセミナー 管理者コース
- ・ ソフトウェア品質マネジメントセミナー 技術者コース
- ・ ソフトウェア品質マネジメントセミナー 入門コース（初級編）
- ・ ソフトウェア品質技術実践講座 デザインレビューコース
- ・ ソフトウェア品質技術実践講座 プロジェクトマネジメントコース
- ・ ソフトウェア品質技術実践講座 品質改善実践コース
- ・ ソフトウェア品質技術実践講座 デバッグ工学とテスト技法コース
- ・ ソフトウェア品質技術実践講座 原因分析コース
- ・ ソフトウェア品質技術実践講座 ソフトウェア機能量測定コース
- ・ ソフトウェア品質技術実践講座 組込みソフトウェアのための開発管理コース
- ・ PS（パートナー満足）とTB（チームビルディング）
- ・ SESSAME認定・組込みソフトウェア技術者・管理者向けセミナー 初級者向けコース
- ・ SEI認定・CMMI®入門コース（段階及び連続）

— シンポジウム —

- ソフトウェア品質に関わるすべての方々が一室に会し、現場で役立つ実践的な技術や経験、ノウハウ、研究成果を発表し意見交換を行う場！
- ・ ソフトウェア品質シンポジウム

— 研究会 —

- ソフトウェア開発に関連した教育と事例研究を行うことを目的として、毎年4月から2月の1年間サイクルで開催！
- ・ ソフトウェア品質管理研究会

— ソフトウェア品質国際ワークショップ（海外調査団） —

- 世界各地のソフトウェア関連組織・企業などを訪問し、実際に現状を「見て」、「聴いて」、『技術交流、パートナー交流』することを目的に原則年に1回派遣！

— ミニセミナー —

- ソフトウェア品質に関するタイムリーな話題を半日のセミナー形式（主に講義とQ&Aの構成）で開催し、参加者全員の意見交換ができる情報収集の場！

クオリティ ワン Quality One

前『Quality One』を発行してから1年。
最近、月日の流れをすごく早く感じます。

前号も「ソフトウェア品質シンポジウム」の開催に併せて発行した本誌は、
今回も「第25回ソフトウェア品質シンポジウム」開催に併せて発行いたしました。
シンポジウムと冊子の同時準備は原稿締切や校正時期が重なるなど、
かなりつらかったです（涙）。

今回も「ソフトウェア品質」をキーワードに、かなり内容の濃い1冊になりました。

特集「Quality with Strategy」ではアーキテクチャ、研究者、開発者の、IT業界にて産学から著名な三氏にそれぞれの視点からご寄稿いただき、とてもわかりやすい内容にまとまっております。

本誌をまとめるにあたり、前号に引き続き企画の段階から多大なるご協力をいただきましたSPCステアリング委員会 ジャーナル担当の大西委員をはじめ、執筆者の方々やSPCステアリング委員会の委員、ご協力していただきましたすべての関係者の方々に心から厚くお礼申し上げます。

日科技連・SPC担当

『Quality One』2006年9月14日発行

編集：日科技連・SPCステアリング委員会

発行：財団法人日本科学技術連盟

〒166-0003 東京都杉並区高円寺南1-2-1

TEL:03-5378-9813 / FAX:03-5378-9842

E-MAIL:spc@juse.or.jp

http://www.juse.or.jp/

Printed in Japan

印刷・製本 日本印刷株式会社

※本誌の全部または一部を無断で複写複製（コピー）することは、
著作権法上での例外を除き、禁じられています。

**本誌に関するご意見・ご感想、
または日科技連の活動に関する
お問い合わせは下記までお寄せ
ください！！**

皆様の声をお待ちしております。

財団法人日本科学技術連盟

クオリティマネジメント一課

SPC担当

E-MAIL:spc@juse.or.jp

TEL:03-5378-9813

フリーダイヤル 0120-207-560(企画広報室)



クオリティ ワン
Quality One

2006年9月14日発行

編集：日科技連・SPCステアリング委員会
発行：財団法人 日本科学技術連盟

〒166-0003 東京都杉並区高円寺南1-2-1
TEL 03-5378-9813/FAX 03-5378-9842
E-mail: spc@juse.or.jp
<http://www.juse.or.jp/>