



ソフトウェア開発における品質保証規格

井上 克郎*

Quality Frameworks for Software Development

Key Words : Software Process, Software Product, ISO9000, CMM, Standard

1. はじめに

ソフトウェアの社会的な重要性の増大に伴って、その品質の善し悪しが重大な問題になってきている。しかし、ソフトウェアは、年々大規模化、複雑化してきており、その品質を管理し、保証するのは非常に困難になってきている。本稿では、近年急速に普及し始めたソフトウェアに関する品質保証規格の中で最も一般的なISO9000-3¹⁾およびCMM²⁾について概説する。

2. ソフトウェア品質保証の考え方

今やソフトウェア製品は社会のシステムの一部組み込まれており、その品質の保持は、システムの安定・維持に必須である。しかし、その品質を知ることとは容易ではない。

ソフトウェアの最終製品やその途中で作られるドキュメントやライブラリなどを総じてプロダクトと呼ぶ。プロダクトの品質は、信頼性、効率、操作性、更新性など複数の品質特性から構成され、さらにそれぞれの品質特性は、多数のチェック項目や計測値(メトリクス値)から構成されると考えるのが一般的である。しかし、これらの多数のデータを収集し評価することは、特殊な高い信頼性を要求されるもの(例えば軍需や宇宙用など)を除いて、経済的な観点から現実的ではない。

そこで、最終的なプロダクトの品質のデータを直接観測するのではなく、そのプロダクトを作成する過程(プロセスと呼ぶ)を観測したり管理したりして、プロダクトの品質を予測したり品質向上をめざしたりすることが考えられる。うまくいけば、この方法で、より安く簡単に高い品質のプロダクトが得られることが期待される。

このような考えに基づき、ソフトウェア開発に関わる活動全体に対して指針を与え、生産性や品質を安定させたり向上させようとする動きが活発である。このような指針は、標準化や規格化され、実際にそれらに基づいてプロセスの評価や改善が行われるようになってきている。これらの活動により、組織の効率化や品質の向上に有益であったとの報告もある^{3,4)}。

このような指針は、一般に、品質に直接関する事項のみならず、生産性に関すること、契約や組織の責任に関すること、教育に関することなど多岐の事項を含む場合が多いが、ここでは、通常良く用いられるように品質保証規格と呼ぶ。これらの代表的なものとしては、ISO9000-3とCMMがある。

3. ISO9000-3

ISO9000シリーズは、ISO(International Organization for Standardization : 国際標準化機構)が制定する品質管理のための一連の国際規格である。これは、製品の開発や製造を行う際、顧客に良い品質の製品を提供するための最低限の品質管理のプロセスについて定めている。

これらの規格は、特に業種を固定せず、一般の製造業を念頭に書かれているが、ソフトウェアの製品は、他の一般的な工業製品(コンピュータのハードウェアも含まれる)とかなり違った特性をもっている。例えば、一つソフトウェアプロダクトが完成すれば、それと同じ品質のものを大量に作ることは非

* Katsuro INOUE
1956年8月16日生
1984年大阪大学大学院基礎工学研究科
後期課程修了
現在、大阪大学大学院基礎工学研究科
情報数理系ソフトウェア科学分野勤務、
教授、工学博士、ソフトウェア工学
TEL 06-6850-6570
FAX 06-6850-6574
E-Mail inoue@ics.es.osaka-u.ac.jp



常に簡単であるが、一般の製品では、品質を均一にするためには、生産ライン上での品質管理作業が必須である。

ソフトウェア製品の開発にISO 9000シリーズを適用するためのガイドラインであるISO9000-3は、表1のように、0～6章で構成されており、このうち0～3章は序文、適用範囲、引用規格、用語の定義などで、本規格が要求する品質管理作業は4～6章に書かれている¹⁾。この規格の特徴は以下のとおりである。

- ・購入者と供給者の二者の間の製品の取り引きの一部として、要求される製品の品質管理について規定している。
- ・ISO 9000-3の記述は、ソフトウェア開発の契約から保守までをフェーズという単位で分割しており、これは、通常よく用いられるウォーターフォールモデルと考えることができる。
- ・いろいろな作業は、文書によってその計画や結果を残すことが要求されている。また、その文書の承認・発行・変更に伴う管理作業が明確化されている。
- ・経営者による管理責任が明確化されている。
- ・他から購入するソフトウェア製品の管理が規定されている。

次に、具体的にISO9000-3に従ったプログラミングの例について述べる。図1は、作業(楕円で表す)とその入出力のプロダクト(長方形で表す)の間を辺で結んだものである。ソフトウェア開発に本来必須なものは、プログラミングの作業及びソースコードのプロダクトの2つのみである。一方、ISO9000-3では、図に示すようにその他の付加的な作業および行ないそのドキュメントをプロダクトとして残すことを要求している。

すなわち、ソースコードをプロダクトとして作ると同時に、作業報告書も作成しなければならない。その作業報告書をもとにしてコーディング作業の品質を測定し、品質に関する文書を作成する(工程品質)。その文書は、承認を受けた後、管理済み文書として保管する。

ソースコードに対して、品質測定、構成管理の作業が行われ、それぞれ記録や結果を文書として承認を受けた後、管理済み文書として保管される。また、構成管理結果に基づいて変更管理が行われ、その結果も承認の後保管される。また、ソースコード自体

も承認作業の上、保管される。

ISO9000-3では、このような付加的な管理作業は、コーディング作業だけではなく、すべてのプロダクト作成作業に必須なものとして規定されている。この図で示したのは、この規格の一つの解釈で、実際、このとおりの順序で作業をする必要はない。また、プロダクトもこのとおり、準備しなくてもよい場合もある。

近年、企業や団体が、その生産活動がISO9000シリーズに合致しているかどうかの認証を受けることが多くなってきている。これは、その企業や団体と契約する際の一つの判断材料というだけではなく、契約するための前提となる場合もある。特に海外への製品の出荷などには、国際規格であるISO9000シリーズの取得は非常に有利である。

このような認証を行う第三者の認証機関が国内外に多数ある。認証を得たい企業や組織は、認証機関による文書の審査、審査官による訪問審査などを受ける。そして、ISO9000が定める品質管理に関するさまざまな要求が満たされているかどうかが判定される。その結果、満たしていると判定されると、認証済みの組織として登録され公表される。このような認証は、企業や組織全体で受ける場合やその一部の部所で受ける場合がある。

通常、認証を得るためには、経営者の意識変革、品質管理責任者の任命、品質管理に関する作業や文書の整備、組織内の教育など通常1年程度の準備とそのため費用が必要である。また、一度認証を取得した後も、半年や1年ごとのフォローアップの審査を受ける必要がある。このように、認証の取得、維持には、手間暇がかかるが、今後、ソフトウェア製品の取り引きが国際化するにつれて、その生産者の素性を明らかにするこの種の認証は、ますます重要になってくるものと思われる。

4. C M M

CMM(Capability Maturity Model:成熟度モデル)とは、米国のカーネギーメロン大学付属のSEI(Software Engineering Institute:ソフトウェア工学研究所)が、国防省の要請により作成した、より良いソフトウェア開発を行うための一連の規範である²⁾。CMMは、ソフトウェア開発を行う組織(会社やそのなかのグループ)がより良い製品が作れるようになるための数々の指針が記述されている³⁾。

表1：ISO 9000-3の構成

- 0. 序文
- 1. 適用範囲
- 2. 引用規格
- 3. 用語の意味
 - 3.1 ソフトウェア
 - 3.2 ソフトウェア製品
 - 3.3 ソフトウェア・アイテム
 - 3.4 開発
 - 3.5 フェーズ
 - 3.6 検証
 - 3.7 妥当性確認(ソフトウェアの)
- 4. 品質システム-フレームワーク
 - 4.1 経営者の責任
 - 4.2 品質システム
 - 4.3 内部品質システム監査
 - 4.4 是正処置
- 5. 品質システム-ライフサイクルでの活動
 - 5.1 一般
 - 5.2 契約レビュー
 - 5.3 購入者要求仕様
 - 5.4 開発計画立案
 - 5.5 品質計画立案
 - 5.6 設計と製造
 - 5.7 テストおよび妥当性確認
 - 5.8 検収
 - 5.9 複製、引渡しおよび据え付け
 - 5.10 保守
- 6. 品質システム-支援活動
 - 6.1 構成管理
 - 6.2 文書管理
 - 6.3 品質記録
 - 6.4 測定
 - 6.5 規則および取り決め
 - 6.6 ツールおよび技法
 - 6.7 購買
 - 6.8 支給ソフトウェア製品
 - 6.9 教育・訓練

CMMでは、組織の開発技能を図2のような五つのレベルに分けている。

レベル1 初期レベル(Initial Level)

混沌としたプロセスで、個人の開発能力や管理者のリーダーシップによっては、プロジェクトが成功しない場合もあるようなレベル。予算超過や納期の遅れなどが頻発し、プロジェクトが途中で中止されることも多い。プロダクトを出荷できる場合もあるが、納期遅れや予算超過が発生する。開発の中心になる人物がいなくなると、同じようなプロダクトの開発もできなくなる。

レベル2 反復可能レベル(Repeatable Level)

プロジェクトを進めるためのノウハウをもっており、過去に行ったのと同じようにプロジェクトを計画し実行することができる。しかし、過去に経験のないような突発的な問題には対応できない場合もある。管理者は、プロジェクトの節目(マイルストーン)ごとに、中間プロダクトを見て、プロジェクトの現状をおおまかに把握することができるが、節目と節目の間に行われる各作業の詳細は、各開発者に任されており、把握していない。

レベル3 規定レベル(Defined Level)

標準的な開発プロセスが規定に沿って文書化され、それが組織で広く用いられている。その標準的なプロセスに基づいてプロジェクトを繰り返

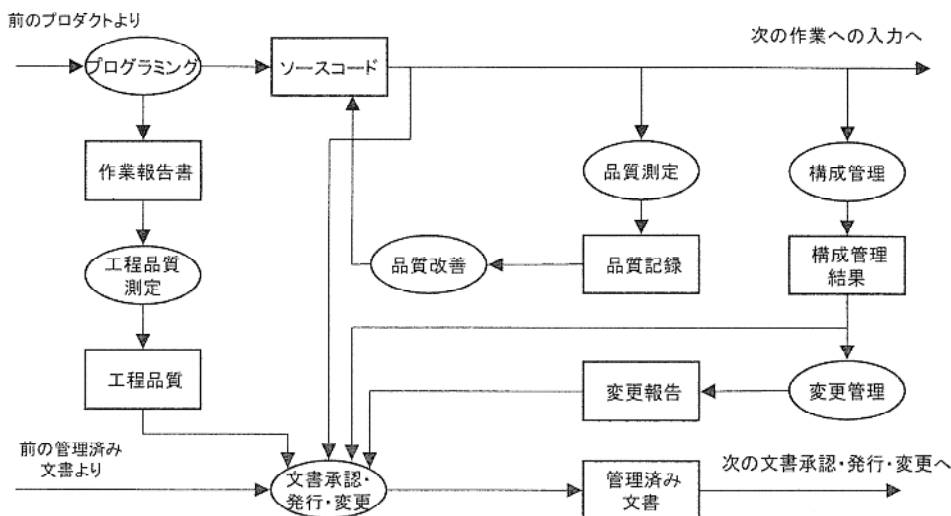


図1：ISO9000-3が要求する作業例

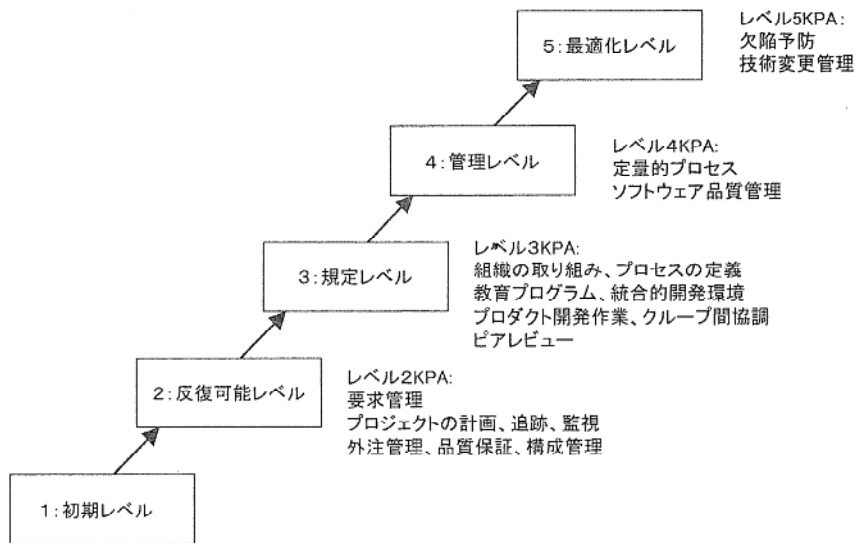


図2：CMMの5つのレベル

し、安定して実行することができる。予算やスケジュールも制御できており、大幅に予算超過したり納期が遅れることはない。各作業も標準的なプロセスに基づいて決められた方法に従って行われるので、管理者は、中間プロダクトだけではなく、個々の作業の詳細も知ることができる。

レベル4 管理レベル(Managed Level)

組織全体として、数量的な品質目標が立てられており、そのためのプロセスやプロダクトのデータが収集されている。管理者は、そのデータを利用して、プロジェクトの現状を知り、問題があれば、早急に対策を立てることができる。新たなプロジェクトに関して、過去のプロセスのデータの裏づけに基づいてスケジュールを立てることができる。

レベル5 最適化レベル(Optimized Level)

組織全体として、継続的にプロセスの改善に取り組んでいるレベル。その組織は、自分たちの弱いところ、強いところを知るための方法を持っている。新しい技術の評価方法があり、新たな技術を採り入れてプロセスを改善する方法を持っている。改善は既存のプロセスの手直しや革新的な技法の導入などによって行われる。

組織は、成熟するにつれて、このレベルを順番に

登っていき、レベルを飛び越えることはできない。レベルを上げることで、組織の生産性は向上し、見積もりはより正確になっていく。

重要作業領域KPA(Key Process Area)は、それぞれのレベルが、多様な品質向上作業のなかでどのような特性について着目しているかを示す。そのレベルに到達するための改善の指針となる。

図3にCMMのレベル2で要求する種々の管理作業を採り入れたプログラミング作業の一例を示す。この図でわかるように、レベル2では、行う作業に関して、作業報告書の作成、その監視、評価、調整作業などが必要である。また、プロダクトに関して、構成、変更管理、レビュー、およびそれに基づく品質改善などの作業が必要になってくる。

CMMに基づいて開発組織を審査するには、以下のような六つのステップを順に行う。

1. まず、評価作業をするチームの選定を行う。
2. 評価対象の組織の代表者にアンケートを行う。
3. アンケートの結果を解析する。
4. 面接とドキュメントのレビューによる実地調査を行う。
5. 組織の特徴、問題点を表した調査結果を作る。
6. 各KPAが満たされているか満たされていないかの一覧を作る。

すべてのKPAが満たされているレベルがその組織のレベルになる(レベル3ならばレベル3と2の全てのKPAが満たされている必要がある)。また、

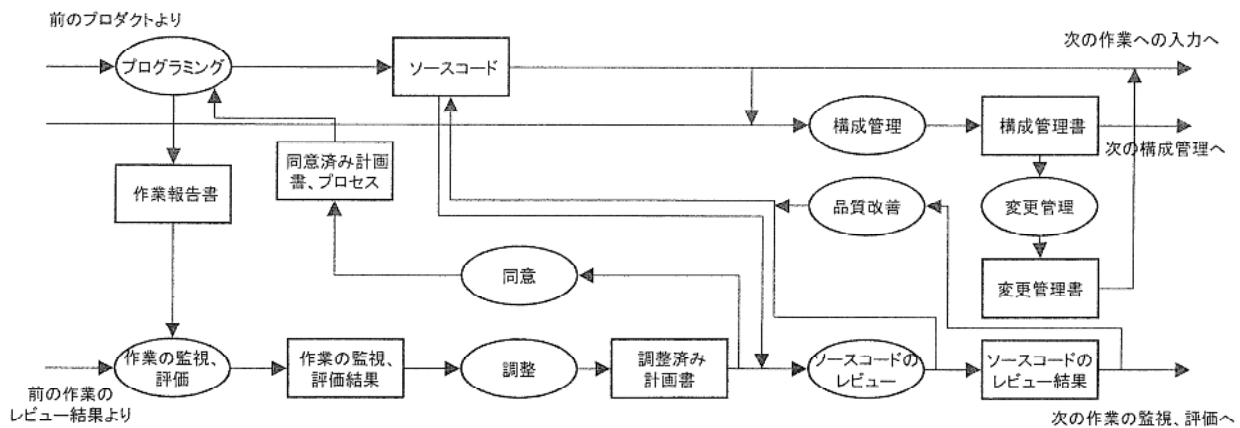


図3：CMMレベル2で要求される管理作業の例

上のレベルのKPAのどれが満たされているか、いないかを見ることによって、組織の強いところ、弱いところを知ることができる。これは、今後、組織のプロセス改善を行うための基礎資料とすることができる。

このCMMが最初に発表された時期に行われた調査では、レベル1 またレベル2の組織がほとんどであった。文献6)によると、1987年から1990年にかけて、米国168、日本196件のプロジェクトを調べた結果、87%がレベル1、11%がレベル2となっていた。

しかし、プロセス改善活動の重要性は次第に認識されてきている。モトローラ社では、プロセス改善を行なっていく段階で得られたいろいろなプロジェクトのデータを用いて、CMMのレベルと品質や生産性との関係を調べている⁴⁾。それによると、レベル5ではコード行数あたりの欠陥発生率はレベル2に比べて7分の1になっている。また、単位時間あたりのコード作成量は2.8倍になっている。

これらの認識のもと、次第に上位のレベルに移行する組織が増えてきつつある。特に、信頼性を要求される航空機、宇宙、軍需、基本ソフトウェア(OS)などの開発を行っている組織は一般に高いレベルにある。

CMMとISO9000-3とはソフトウェア開発の品質を高めるという同様な目的をもつ規格文書である。それぞれの規格の中でいろいろな品質管理に対する要求が記述されている。どういったことが要求されているかを項目を挙げて整理することによって、それぞれの規格の特徴、弱点などを知ることができる⁷⁾。

多くの項目は、両規格で共通して要求されており、

同等である。ISO9000-3にあってCMM(レベル2～5すべて)にはないという、ISO9000-3の特徴としては、

- ・ 経営者による管理責任の明確化
- ・ 他から購入するソフトウェア製品の管理
- ・ 文書の承認、発行、変更にとともなり管理作業が厳密
- ・ プロセスの継続的改善
- ・ 見積もり作業に対する管理の要求
- ・ 計画プロセス変更に伴う同意が必要

などがある。CMMとISO9000-3はこのようにお互いに欠けている部分があり、対応がすべて取れるわけではないが、強いて言えばISO9000-3はCMMのレベル3までで要求される項目と重なる部分が多いと言えよう。

5. む す び

ソフトウェア品質保証規格のISO9000及びCMMについて概説した。CMMと似た構造や目的を持つ規格としてSPICE⁸⁾がある。これはCMMの1～5レベルに対して、0～5のレベルがあり、その評価法も作業の種類と各レベルの特性によるマトリクスを用いるなど、より構造的に評価ができるようになっており、近々ISO15504として国際規格となる予定である。

これらの規格の普及とともにそれらの認証システムも整備されつつあり、社会的に大きな影響を持つようになってきた。一方、近年のインターネットの急速な普及に伴い、従前の方法では捉え切れないシ

システム開発が行なわれるようになってきている。今後、このようなソフトウェア品質保証規格がどのようにインターネットベースのシステム開発やコンテンツ作成に生かせるかどうか、新たな考えが求められている。

参 考 文 献

- 1) ISO : *ISO 9000-3 Guidelines for the Application of ISO 9001 to the Development, Supply, and Maintenance of Software*, International Organization for Standardization (1991), 飯塚悦功編, ソフトウェアの品質保証 -ISO9000-3 対訳と解説 一, 日本規格協会, 1992.
- 2) CMU-SEI : *The Capability Maturity Model : Guidelines for Improving the Software Process*, Addison Wesley, MA, 1995.
- 3) 青山幹雄 : ソフトウェアプロセス・リエンジニアリング, 情報処理, Vol.36, No.5, pp.442-447, 1995.
- 4) M. Diaz, J. Sligo : *How Software Process Improvement Helped Motorola*, *IEEE Software*, Vol.14, No.5, September, pp.75-81, 1997.
- 5) W. S. Humphrey : *Managing the Software Process*, Addison-Wesley, MA, 1989, 藤野喜一監訳, ソフトウェアプロセス成熟度の改善, 日科技連1991.
- 6) W. S. Humphrey, D. H. Kitson, J. Gale : *A Comparison of U.S. and Japanese Software Process Maturity*, *International Conference on Software Engineering*, pp.38-49, Austin, TX, 1991.
- 7) 井上克郎, 渡辺淳志, 飯田元, 鳥居宏次 : ソフトウェア品質保証規格ISO9000-3に基づく管理プロセスの記述とその比較, 情報処理学会論文誌, Vol.36, No.2, pp.453-462, 1995.
- 8) K. El Emam, J. Drouin, W. Melo : *SPICE, the Theory and Practice of Software Process Improvement and Capability Determination*, *IEEE Computer Society*, Washington D.C., 1998.
- 9) 井上克郎, 松本健一, 飯田元 : ソフトウェアプロセス, 共立出版, 2000.

