

テストプロセス改善の CMM 的アプローチ

An approach for test process improvement using CMM

主査

大川 鉄太郎 (日本ユニシス株式会社)

副主査

小笠原 秀人 (株式会社東芝)

研究員

木田 昌平 (株式会社セゾン情報システムズ)

崔 恩瑛 (三星 S D S 株式会社)

中島 則秋 (株式会社エス・キュー・シー)

福留 広晃 (オムロンソフトウェア株式会社)

藤村 薫 (株式会社産能コンサルティング)

横田 正樹 (パイオニアシステムテクノロジー株式会社)

研究概要

どのようなソフトウェアの開発でもテストプロセスは必要である。そして、テストの工数は、開発工数の中でも大きな比重を占めている。したがって、ソフトウェア開発の生産性向上および、ソフトウェアの品質強化のためにはテストプロセスの改善を図る必要がある。しかしながら、テストプロセスの改善は、開発プロセスの改善ほど研究が進められていない分野である。

そこで、本研究では一般的によく知られているソフトウェア開発プロセス改善モデルである SW-CMM を参考にして、テストプロセスを改善していくための成熟度モデルである Test-CMM を作成し、それを利用したテストプロセス改善について考察する。

Abstract

Test process is very important to assure software quality. Recently, the person-hour in software development is increasing rapidly. Considering that software is getting larger and more complex, quality must be maintained from the early phases such as requirements analysis, design and coding. To shift the viewpoint of building in quality to early phases, it is very useful to use results of test activities. However, the research for test process improvement is in their infancy.

The Capability Maturity Model for Software (SW-CMM) is a model for organizational capability that provides guidance for establishing process improvement programs. Now, many companies apply the SW-CMM to improve their software development process. In our research, we developed Test-CMM based on SW-CMM and TPI (Test Process Improvement Model). Furthermore, we proposed the improvement procedure for software development process using Test-CMM.

1．研究課題の選定理由と背景

近年、ソフトウェア開発においては「短納期」「品質向上」はごく当り前のこととして要求されている。それに加え、要求の多様化によるソフトウェア大規模化、テスト項目増大などに対応するため、ソフトウェア開発プロセスの改善が急務とされている。

このような状況の中で、ソフトウェア開発プロセスの実施に必要な要素として SEI によってソフトウェア成熟度モデル(以下、SW-CMM)が開発された。SW-CMM では、未熟な組織から成熟し規律がとれたプロセスへと移行していく上で組織がとるべき段階的な改善の指針を提示している。当分科会メンバでも、CMM に取り組んでいる組織があり、今後も増えていくことが予想される。

しかしながら、どのように開発プロセスを効率化したとしても、開発プロセスの最終であるテストプロセスが十分に機能しなければ高品質な製品をリリースすることはできない。また、最近脚光を浴びつつあるエクストリーム・プログラミング(XP)でもテスト・ファーストをポイントの一つとして挙げており、テストの重要性は従来に増して認識が深まりつつある。

そこで、当グループではソフトウェア開発プロセスを開発プロセスとテストプロセスととらえ、テストプロセスの改善に着目した。そして、テストプロセスの段階的な改善の指針として、SW-CMM の考え方に基づき、テストプロセスの成熟度モデルとして、Test-CMM を作成することにした。また、具体的なテストプロセスの改善手法として、IQUIP 社の構築した TPI モデルを利用し議論を進めることとした。

2．本年度活動の内容

当分科会では、前述の研究課題の選定理由と背景を踏まえ、次の内容を中心とした研究を行った。

- ・テストプロセスの重要性
- ・テストプロセス成熟度モデル(Test-CMM)の作成
- ・Test-CMM と TPI のマッピング
- ・Test-CMM と TPI によるテストプロセス改善の考察

研究内容については、以降の章で詳細を述べる。

3．テストプロセス

3.1 テストの重要性

現在、ソフトウェアの開発にあたり、多くの開発プロジェクトで採用されている開発手順はウォーターフォールモデルである。図 3.1 に一般的なソフトウェア開発の V モデルを示す。V モデルの左側を開発、右側をテストとすると、それぞれの開発プロセスに対応するテストが存在することになる。一般的にバグの修復には工程内で検出できなかった場合、手戻り工数が発生するためより多くのコストを要することになる。そして、最終工程であるテストでバグを検出できなかった場合、バグを含んだまま製品がリリースされることになり、その修復に要するコスト

的な損失に加え、製品や企業のイメージダウンなどの社会的な損失まで招くことになる。

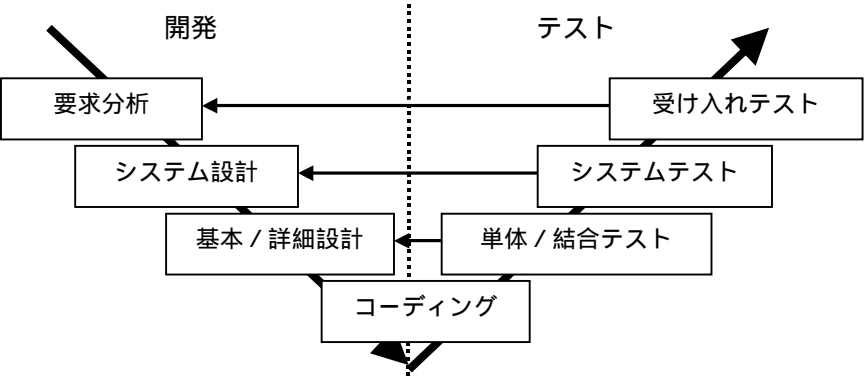


図 3.1 一般的なソフトウェア開発の V モデル

このように、テストは製品のバグを検出できる最終工程として重要となる。それであればこそ、開発のプロジェクトの多くではテストの工数を多く割り当てているのである。開発工程全体の工数に対するテストの工数の実情を表 3.1 に示す。

表 3.1 テストの工数割合

ソフトウェア開発		開発工数(人月)	テスト工数(人月)	テスト割合
パッケージソフト	新規開発	130.5	71.5	54.8%
	バージョンアップA	13.0	9.7	74.6%
	バージョンアップB	28.6	20.0	69.9%
SI	プロジェクトA	115.0	32.0	27.8%
	プロジェクトB	88.0	29.0	33.0%
	プロジェクトC	210.0	65.0	31.0%
組み込み	プロジェクトA	62.0	11.0	17.7%
	プロジェクトB	62.0	10.0	16.1%
	プロジェクトC	66.0	12.0	18.2%
開発工数 : SLCP-JCF98「1.4 開発プロセス」の工数				
テスト工数 : SLCP-JCF98「1.4.8 ソフトウェアコード作成及びテスト～1.4.12 システム適格性確認テスト」				

パッケージソフト系、SI 系では開発工程の約 3～7 割を占めており、組み込み系においても今後、ソフトウェア比重の増大により、テストの割合は増加していくことが予想される。

3.2 テストプロセスの改善

前節で述べたように、テストはソフトウェア開発工数全体からみても大きな割合を占めている。また、バグを含んだ製品のリリースを防ぐ意味でも重要なプロセスである。また、テスト自体では製品の品質を向上させることは不可能であるが、品質を向上するための有効な情報を提供できる。テストはソフトウェア開発における一工程であるが、開発期間の短縮と適用業務の複雑さの増大により、ソフトウェア開発の計画段階よりテストを考慮せざるを得なくなっている。テストの観点から他の上流工程に対してのフィードバックは、設計者や開発者がテスト結果の情報をを用いて製品の品質を改善することを可能にする。通常、修正のコストはシステム開発フェーズが進むにつれ急激に増大するため、レビューやテストは開発プロセスの上流工程から率先して行うのが工数の上でも効率的である。

テストをプロセスととらえ、ソフトウェア開発プロセスの中の一部として改善を行うことは、

バグの検出率を上げ、なおかつコストを削減することにつながるだけでなく、図 3.1 に示したように対応する開発プロセスにフィードバックを行うことで、ソフトウェア開発プロセス全体の改善へとつながるものとなる。

4 . Test-CMM

4.1 SW-CMM と Test-CMM

テストプロセスを改善するためには、参照するための改善モデルが必要となる。一方、本研究では、SW-CMM に取り組んでいるメンバも数人いることから、SW-CMM によるプロセス改善に取り組んでいる組織においても有効に利用できるように、SW-CMM をベースとしたテストプロセス改善モデルを作成することにした。ソフトウェア開発プロセス全体を開発とテストに分け、テストのみのプロセスの能力成熟度を考えたモデルが Test-CMM である。

4.2 Test-CMM とは

4.2.1 Test-CMM の構造

Test-CMM の構造を作成する前に、まず Test-CMM を適用する組織モデルと用語の定義を行った。組織モデルは、SW-CMM で言われているソフトウェアエンジニアリンググループが「開発グループ」と「テストグループ」で構成されるものと考え、両者のプロセスリレーションを「付録 4.2-1」に、適用する組織モデルを「付録 4.2-2」に、新しい用語定義を「付録 4.2-3」に、それぞれ示す。

Test-CMM の構造は、基本的に SW-CMM と同様にレベル 2 からレベル 5 の階層を設けてテストプロセス能力成熟度 / ゴールを表し、各レベル内にキープロセスエリア(以下、KPA)が存在し、KPA は更にソフトウェア開発でのテストプロセスに関するキープラクティス(以下、KP)に詳細化してその説明記述と TPI (後述) とのマッピングを示している。 マッピングについての説明は、4.4 節で記述する。

Test-CMM の KP は、SW-CMM の各レベルにおける KP をテスト用語に置き換えて、テストプロセスに係わると思われるものを選択して編成したもので、その内容を「付録 4.2-4」に示す。

4.2.2 Test-CMM の成熟度

前述したように、Test-CMM の成熟度はレベル 1 から 5 の 5 段階で表される。次に各レベルが対象とする組織・プロジェクトはどのような状態であるかを解説する。

(1) レベル 1 - 初期レベル

テストは混沌とした状態であり、プロジェクト間で成熟度に大きなバラツキがある。このバラツキは、成熟したプロセスと未熟なプロセスが存在し、組織として何の規則もない状態である。このレベルでは、テストはプロジェクトリーダーやテスト担当者など、個人の努力に頼っており、場当たりのテスト、バグ報告を行っている。

(2) レベル 2 - 反復できるレベル

組織としての基本的なテスト技法と方法論が制度化され、テストグループは計画、設計、実施といったテストのライフサイクルを繰り返し実行しているレベルである。これらの基本プロセスはできているが、依然としてプロジェクト間のバラツキは存在している。

(3) レベル 3 - 定義されたレベル

組織標準のテストプロセスが製品のソフトウェアライフサイクルに組み込まれている。テストプロセスが組織的に制御・監視され、標準化が進んでいるためプロジェクト間のバラツキは少なくなっている。また、テストで得られたデータはデータベースに記録されている。

(4) レベル 4 - 管理されたレベル

組織全体でテストプロセスが実装され、プロセスのデータが定量的に首尾一貫して収集されている。テスト品質管理はテスト品質計画に基づき実施され、テストは定量的に分析把握され制御されている。プロジェクト間のバラツキは限りなく平均値に近づき標準化されている。また、このレベルでは平均値の予測が可能である。

(5) レベル 5 - 最適化するレベル

テストプロセスは定常的にうまく運用され、組織が継続的なテストプロセス改善に集中できる。また、組織はテストプロセスからのデータを分析することにより欠陥を未然に防ぐ術を持っており、ここで得られた情報(テストプロセスの変更や新技術)は他のプロジェクトへフィードバックされる。このように、テストプロセスを最適化していることから、プロジェクト間のバラツキはなく、その平均値の制御ができる。

4.2.3 グループ間のフィードバック

Test-CMM はテストプロセスに限定した改善モデルであるが、SPI の観点からソフトウェア開発プロセス全体とのバランスが重要となる。開発プロセスが成熟していなければ、高品質のソフトウェアは開発できない。逆に、開発プロセスのほうが成熟している場合は、高品質のソフトウェアが開発できたとしてもそれを保証できる環境が整ってないことになる。したがって、テストプロセスを改善するとともに、開発プロセスとの間で互いに情報やアウトプットのフィードバックを行い、ソフトウェア開発プロセス全体をバランスよく改善していかなくてはならない。

4.3 TPI について

TPI とは「Test Process Improvement」の略で、テストプロセス改善アプローチモデルの意味であり、これはヨーロッパにおいて IT システムのテストに特化してきた IQIUP の経験と知識から得られたものであり、現在はその使用が急速に拡大し、オランダにおけるソフトウェア標準になろうとしている。

このアプローチは、CMMI(連続表現ではプロセスエリア毎で能力レベル評定ができるということ)と類似しており、テストプロセスの様々なキーエリア(以下、KA)について、組織の成熟度を

客観的に自己評価し、弱みと強みを示し、レベル付けをして改善に役立てるものである。

テストプロセスは開発プロセス全体の一部であり、図 4.1 に示すように総合的品質管理(TQM)やソフトウェアプロセス改善(SPI)を行う

組織において TPI はその一部であると言える。

TPI は、テストプロセスを構造化することによって

テストに関する多くの問題を解決することを

目標としており、構造化されたテストにより

テストプロセスの全ての視点をカバーする一連の

アクティビティ、手順、技法を示している。

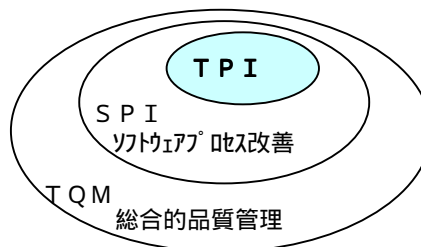


図 4.1 TPI の位置付け

TPI では、KA 毎にレベル付けが行われており、それを一覧表形式で表したものが TPI_Sheet(付録 4.3-1)であり、その構成内容は以下のようにになっている。

No：KA 番号とチェックポイント毎の通番を表す。(例：1-3)

キーエリア：テストツールの利用、テスト仕様化技法、報告などの様々な視点からテストプロセスを眺めた時の主要領域名を表している。

レベル：各 KA を成熟度レベルに分類したもので、A から D に向かって高レベルとなる。レベルが上がることは時間、費用、品質面で改善していることを意味する。

説明：キーエリアの各レベルの説明である。

チェックポイント：キーエリアにおける特定のレベルのチェックポイントを示しており、テストプロセスがそのレベルに評価されるために満足すべきポイントを表す。

依存性：そのレベルとして評価されるためのチェックポイント以外で満足すべき依存性のある(関連する)KA / レベルを示す。

(例：テスト戦略レベル A の依存性には「5A」と「11A」が記述されており、これは4つのチェックポイントの他に、「テスト仕様化技法」のレベル A「5A」と「コミットメントと意欲」のレベル A「11A」を達成していなくてはならないことを示している。)

4.4 Test-CMM と TPI

4.4.1 Test-CMM と TPI の関係

本研究では、Test-CMM においてアセスメントされたテストプロセスを TPI の手法を用いて改善を行う議論を行った。具体的には、Test-CMM の KP に TPI の KA のレベルをマッピングしてその KA のレベルを中心に依存性のある KA のレベルも同時に改善を行っていくものである。実際のマッピングでは、Test-CMM と TPI の親和性の問題から、Test-CMM の KP にマッピングしきれない KA や、Test-CMM の成熟度と TPI の成熟度で差が出てしまうものがあるため、Test-CMM の KP のゴールの理想形として合致する TPI の KA のレベルをマッピングした。Test-CMM と TPI のマッピングを「付録 4.2-4 "関連する TPI KeyArea Level"列」に示す。

実際の改善では、KA の依存性も考慮した改善を実施していくことになる。もし、テストプロセスがゴールとする KA のレベルまで達していない場合は、必然的に低いレベルからの改善

が必要となるが、それによってテストプロセス全体の成熟度の底上げが期待できる。次項で Test-CMM と TPI によるテストプロセス改善のライフサイクルを説明する。

4.4.2 テストプロセス改善のライフサイクル

図 4.2 に Test-CMM と TPI によるテストプロセス改善のライフサイクルを示す。

改善は、テストプロセスを Test-CMM の KP ごとにアセスメントし、ゴールを判定する。判定されたゴールは KP にマッピングされた TPI の KA のレベルのチェックポイントによって改善ポイントへ詳細化される。改善ポイントをもとに、改善の計画/実施が行われ、改善の成果の程度(ゴールが達成できたかどうか)が評価され、1 サイクルが終了する。評価によっては、次の改善のためのアセスメントを行うのではなく、改善の計画へ戻り再び同じ改善を実施する場合もあり得る。

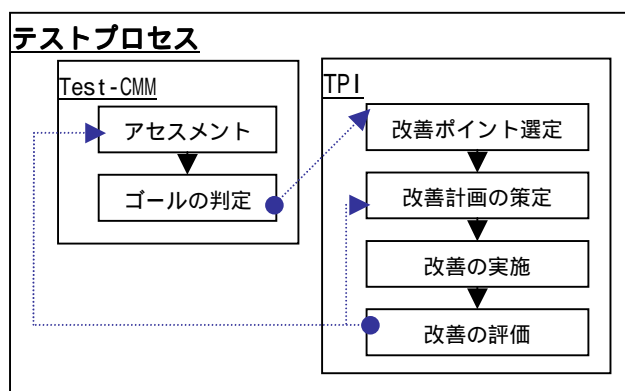


図 4.2 テストプロセス改善のライフサイクル

5 . Test-CMM および TPI を利用したテストプロセス改善の考察

本研究で作成した Test-CMM は、SW-CMM の考え方を踏襲しているため、2 モデル間の親和性は確保されている。それに対して、段階的に成熟していく Test-CMM と依存性があるものの各 KA を連続的に成熟させる TPI では、2 モデル間のマッピングにおいて親和性の確保は困難なものとなってしまった。

しかし、Test-CMM によってアセスメントを行い、TPI で改善を行うアプローチは有効な手段である。以下に 2 つのモデルの利点と欠点を示す。

- ・ Test-CMM の利点
 - 取り組むべきプロセスのロードマップとなる
 - 成熟度をレベルとして総合的に判断できる
- ・ Test-CMM の欠点
 - 自由度に欠け、考え方が固定化する
- ・ TPI の利点
 - 重視するプロセスにフォーカスが当てやすく、自由度が高い
- ・ TPI の欠点
 - 成熟のロードマップが見えにくい

このように、2 つのモデルは、お互いのモデルの欠点を補い合う関係にある。そのため、Test-CMM で改善のロードマップを示し、それに沿った改善を TPI で行えばよいわけである。

重要なことは、2モデル間の親和性を考えて TPI による改善を実施していくことである。そうすることによって、Test-CMM は SW-CMM を導入している組織においても十分有効性のあるテストプロセス改善モデルとなる。

また、TPI はレベルが上がるにつれて、テストチーム、プロジェクト、組織へとスコープが広がっていくモデルである。つまり、レベルが上がるほど開発プロセスとのつながりが強くなり、お互いのフィードバックが頻繁に行われるようになるので製品の品質も向上していく。

6 . まとめと今後の課題

一般的にテストは地味な工程というイメージがあり、テストを専門に行う技術者は少ない。2002 年 12 月に経済産業省から発表された ITSS(IT スキル・スタンダード)でも、専門分野としてのテスト技術者のカテゴリは存在しない。しかし、製品の品質向上のためにはテストプロセスの改善は必須である。テストに対する認識が深まり、テスト技術者が増加し、不断の改善活動によってテストプロセスそのものの質が向上するだけでなく、製品の品質も必ず向上するであろう。

今回の研究では、テストプロセスの成熟度モデルである Test-CMM を構築した。成熟度をあげるためには具体的に何をすればよいかについて TPI を利用する提案も行うことができた。実際のテスト作業に Test-CMM をあてはめた時、テストプロセスの改善点の発見に有効なレベルのものとなっている。しかしながら、Test-CMM と TPI の親和性については、その議論を十分に行うことができなかった。2つのモデル間に親和性をもたせる方法を策定することが今後の課題となる。

また、テストプロセスは製品の品質を検証することが主眼となりがちである。製品の品質向上は、テストプロセスの改善のみに留まらず、開発プロセスへのフィードバックまでおこなってこそ役立つことができる。具体的に何を、どのように、どのタイミングでフィードバックすれば製品の品質が向上するのかは、まだまだ議論する必要がある。

参考文献

- 1) Tim Koomen,Martin Pol 著:「テストプロセス改善 - CMM 流実務モデル - 」,構造計画研究所,2002 年
- 2) 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著:
「能力成熟度モデルのキープラクティス 1.1 版 SEA CMM 研究会 公式日本語版」,
ソフトウェア技術者協会,1999 年
- 3) 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著:
「成功するソフトウェア開発 - CMM によるガイドライン」,オーム社,1998 年
- 4) 保田勝通著:「ソフトウェア品質保証の考え方と実際」,日科技連出版社,1995 年
- 5) ADAMOS:「Software Quality Management Center」,<http://www.cam.hi-ho.ne.jp/adamosute>