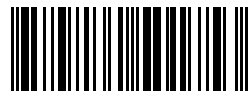


オーム社/雑誌局

ISBN978-4-274-50175-3

C3055 ¥1619E



9784274501753



1923055016194

定価(本体1619円【税別】)

SEC BOOKS

SEC BOOKS

プロセス改善ナビゲーションガイド

～ベストプラクティス編～

独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター 編

IPA® 独立行政法人 情報処理推進機構
ソフトウェア・エンジニアリング・センター

プロセス改善ナビゲーションガイド

～ベストプラクティス編～

独立行政法人 情報処理推進機構
ソフトウェア・エンジニアリング・センター 編

Ohmsha

SEC-TN07-007



プロセス改善ナビゲーションガイド

～ベストプラクティス編～

独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター 編

商標等の取扱いについて

- ・CMM[®], CMMI[®] and Capability Maturity Model[®] are registered with the U. S. Patent and Trademark Office.
- ・SEPG は、米国カーネギーメロン大学のサービスマークです。
- ・Microsoft, Windows, Visual Studio, Excel は米国 Microsoft Corporation の米国およびその他の国における登録商標です。
- ・Microsoft Word は米国 Microsoft Corporation の商品名称です。
- ・Windows の正式名称は、Microsoft Windows Operating System です。

上記にかかわらず、本ガイドに掲載されているシステム名、製品名等は、一般にその開発元の商標または登録商標です。本ガイドでは、本ガイドを製作する目的でのみそれら商品名、団体名を記載しており、編著者としては、その商標権を侵害する意思、目的のないことを申し述べておきます。

本書を発行するにあたって、内容に誤りのないようできる限りの注意を払いましたが、本書の内容を適用した結果生じたこと、また、適用できなかった結果について、著者、出版社とも一切の責任を負いませんのでご了承下さい。

本書は、「著作権法」によって、著作権等の権利が保護されている著作物です。本書の複製権・翻訳権・上映権・譲渡権・公衆送信権（送信可能化権を含む）は著作権者が保有しています。本書の全部または一部につき、無断で転載、複写複製、電子的装置への入力等をされると、著作権等の権利侵害となる場合がありますので、ご注意ください。

本書の無断複写は、著作権法上の制限事項を除き、禁じられています。本書の複写複製を希望される場合は、そのつと事前に下記へ連絡して許諾を得てください。

(株)日本著作出版権管理システム(電話 03-3817-5670, FAX 03-3815-8199)

JCLS <(株)日本著作出版権管理システム委託出版物>



はじめに

ソフトウェア・エンジニアリング・センター（SEC）では、増大化、複雑化するソフトウェア開発および保守において、高い品質を維持しつつ、その活動を効率的に進めるために、「プロセス改善」によって、組織能力を向上させる必要があると考えています。プロセス改善研究部会では、数々の「プロセス改善」の事例を取り上げ、それを紹介することを通して現場レベルから、経営者にいたる種々のステークホルダ（利害関係者）の意識変革にも取り組んでいます。

プロセス改善の進め方は、大きく2つに分類することができます。1つ目はベストプラクティスを抽象化し、モデル化したものと比較検討して、対象となるプロセスの診断をする進め方です。2つ目は現場で発生した課題の解決を積み重ねていく進め方です。プロセス改善の2つの進め方を見ると、今や国際語となった「KAIZEN」が、現場における地道な努力の積み重ねで実施されているように、「プロセス改善」も現場における実践が問題解決に活用されていることがわかります。

本書は、既に発刊されている「プロセス改善ナビゲーションガイド」シリーズの一環をなす～ベストプラクティス編～です。実際のソフトウェア開発現場で実施されており、他の参考になる事例、いわゆる「ベストプラクティス」を統一された様式でまとめています。

プロセス改善とは

QCDつまり品質（Quality）とコスト（Cost）、そして納期（Delivery）を維持、向上するために、ライフサイクル全体に渡って活動（プロセス）をより良くすることをプロセス改善といいます。

増大化、複雑化するソフトウェアに対して、高い品質を維持し、効率的に開発を進めるためには、各エンジニアによる知識とスキルの獲得のみならず、開発チームとしてソフトウェア開発・維持およびその管理にあたるのですから組織能力の

向上が必要です。

また、ソフトウェアの開発および保守にあたっては、単に開発チームという直接的に開発に関わっている組織の他に、発注者、製品開発企業、請負企業、人材派遣企業、運用企業など様々なステークホルダが存在します。よって、プロセス改善にあたっては、個別の組織・企業だけでなく、これらのステークホルダも視野に入れた取り組みが必要となります。

プロセス改善が目指すものは、ライフサイクル全体にわたって活動をより良くすること、例えば“QCD”の予測精度・制御精度を上げ、さらに有効性を高めることですから、その目的を達成するために、それぞれの組織・企業が目的になかったプロセス改善を進めることが必要になります。

SECにおけるベストプラクティス収集への取り組み

プロセス改善を進める上で、各種の事例は大変参考になるとの要望もあり、有効性が実証されたプロセス改善事例をひろく収集し、1つのポータルサイトにて検索ができるようになればと考えています。

各種コンファレンス、論文、訪問詳細調査等を行い、既に昨年初夏には第1次発表として3社についてベストプラクティス事例をSEC-Webサイト上で公開しています。今回、新たに7社を加え、10社のベストプラクティス事例を紹介しますので、ご活用ください。

2008年2月 著者らしるす



目次

第1章 効果的プロセス改善のための 情報共有の必要性	1
1.1 プロセス資産という考え方	1
1.2 プロセス資産の蓄積と活用	3
1.3 事例の共有化 — 使える情報から学べる情報へ	8
1.4 他組織に学ぶ	9
1.5 外部への情報発信の意義	9
第2章 プロセス改善とベストプラクティス	11
2.1 ベストプラクティスとは	11
2.2 プロセス改善におけるベストプラクティスの活用	12
2.3 ベストプラクティスの調べ方	14

第3章 ベストプラクティス集 17

事例1：設計からはじめる見逃しバグの防止	22
事例2：生産性を5年で2.5倍に！	36
事例3：究極の高品質ソフトウェア開発プロセスを目指して	54
事例4：生産性原理を具現化するプロセス改善	72
事例5：高品質ソフトウェア開発管理プロセスの実現	98
事例6：マネジメント自らが予防のための行動を取り、 問題が起こりにくい組織に	114
事例7：派生開発による品質および開発効率の向上	132
事例8：ソフトウェア部品による組立型開発と品質保証	146
事例9：全社レベルでのプロジェクト管理の改善	168
事例10：GDD（Genba Driven Development）	182

付 録 199

参考情報：公開されているプロセス改善事例	199
著者／インタビュー一覧	200
用語／略語	202
参考文献	203
索引	204

第1章 効果的プロセス改善のための 情報共有の必要性

ソフトウェアエンジニアリングに限らず、様々な製品やサービスを作り出すための知識体系としての工学の発展には、研究や実験等によって得られた発見的、普遍的な知識に加えて、それらをもつづくりの場で実践することによって得られた経験的な知識の蓄積が寄与していることに異論をはさむ余地はありません。

プロセス改善においては、より質の高い製品やサービスを、より安定的、効率的に作り出すための手段を追及するために、その実現のベースとなる理論や技法等を習得するとともに、それらを様々な局面で適用した結果から得られた経験知を蓄積する必要があります。経験知には、成功から得られたもの、失敗から得られたものがあり、双方とも貴重な教訓となります。しかしながら、経験知は、一部の関係者の頭の中のみ刻み込まれ、時間の経過とともに忘れ去られたり、人の異動とともに失われたりしがちです。このような状況を打破するためには、知識を格納するための仕組みを構築し、様々な知識を蓄積し、活用することが重要です。プロセス改善に活用できる様々な知識をプロセス資産と呼びます。組織的な改善活動を効果的に推進するためにプロセス資産が蓄積されたりポジトリの構築と活用が推奨されています。

この章では、プロセス資産の蓄積と活用による、効果的なプロセス改善の進め方について解説します。

1.1 プロセス資産という考え方

業務プロセスの改善に取り組む際には、改善の対象とする主要なプロセス領域を組織のビジネスゴールや解決したい課題に照らして絞り込み、それらのプロセス能力を向上していく必要があります。ISO/IEC 15504 第5部（プロセスアセスメントモデルの例）や CMMI（能力成熟度モデル統合）等のアセスメントモデルではプロセス能力を表1に示すプロセス能力水準に沿って段階的に向上させること

を推奨しています。

プロセスの能力水準を3またはそれ以上に向上させる過程において、プロセスの遂行はプロジェクトごとの個別最適から組織としての全体最適に進化します。能力水準2では、個々のプロジェクトの特性を考慮して、実践すべき作業を明確にし、工数の見積りやスケジューリング等を含む作業計画を立案して、作業状況を計画に照らしてコントロールすることが求められます。これに対して能力水準3では、能力水準2で確立された規範は維持しながら、組織内外のベストプラクティスを見出し、それをベースに標準プロセスを制定し、プロジェクトに展開し、定着させることが求められます。さらに、能力水準3では標準プロセスを実践する過程で得られた様々なノウハウや教訓を蓄積して、標準プロセスの強化やより効果的、効率的な実践を支援する仕組みの構築を行います。

このようなプロセス能力向上の過程で得られた様々なノウハウや教訓、組織知としての標準プロセス等が、プロセス資産に該当します。それらを蓄積し、共有可能にし、実務で活用されるようにするためには、情報の格納庫としてのリポジ

トリの構築に加え、組織的なナレッジマネジメント活動を活性化するための様々な施策が必要になることはいうまでもありません。

1.2 プロセス資産の蓄積と活用

リポジトリに蓄積されたプロセス資産は、組織的なプロセス改善を効果的に推進するための情報源となります。一般に、プロセス資産として、蓄積すべき情報には以下に示すようなものがあります。これらの情報の共有化と活用の意義を以下に説明します。

プロセス資産① 組織方針

計画立案、構成管理等、個々のプロセス領域の改善に取り組むに当たって、また、それらの標準プロセスの制定と展開に当たって、改善の必要性や達成すべき目標を明確にする必要があります。この方針が明確でなければ、改善活動に対する実務者の理解や協力を得ることが困難になります。これらの方針は、改善活動の立ち上げの段階で組織の成員に周知するとともに、改善活動の推進の過程でいつでも原点に立ち戻って方向性の確認ができるよう、プロセス資産として情報の共有化を図ることが重要になります。

プロセス資産② 標準プロセス

組織で制定した標準プロセスは、制定すること自体に意味があるのではなく、そのプロセスをプロジェクトに定着させることで、初めて期待する改善効果を達成することが可能になります。標準プロセスの制定や改正の際には、プロセス標準化の推進グループからその趣旨や要旨を関係者全員に伝え、それを実践する要員に必要なトレーニングを行うとともに、標準プロセス適用の局面で常にその内容を振り返って参照できるよう、プロセス資産として情報の共有化を図る必要があります。

プロセス資産③ テーラリングガイド

組織内外のベストプラクティスを集約して、組織の標準プロセスを制定したとしても、それらのプロセスが万能で、あらゆるプロジェクトに対してベストであるとは限りません。とくに組織が比較的大規模で、複数の製品群を開発しているような場合、製品の種類に応じて柔軟に標準プロセスを見直して適用する必要性

表1 プロセス能力水準

水準値	プロセス能力	概要
0	不完全なプロセス	その組織でのプロセスはまったく実行されていないか、プロセスの目的を達成していることを示す証拠となる成果が、ほとんどあるいはまったくない状況をいう。
1	実施されたプロセス	その組織でプロセスを実施した結果、適切な入力作業生産物が存在している状況をいう。
2	管理されたプロセス	その組織でプロセスの実施にあたって、作業と作業生産物の計画が立案され、実施を監視し、計画と差異があればこれを解消している状況をいう。
3	確立されたプロセス	組織の標準プロセスが確立されて、この標準プロセスを適切にテーラリングし、プロセスを実施している状況をいう。
4	予測可能なプロセス	プロセスの実施は定量的に管理され、組織の事業目標を達成するように予測可能な状態にあることをいう。
5	最適化しているプロセス	プロセスの実施は、組織の事業目標を達成するために、定量的な改善目標に基づいて、継続的に改善されている状況をいう。

が出てきます。しかしながら、その見直しをすべてプロジェクトの裁量に任せてしまうと、標準プロセスが棚上げになってしまい、標準化自体が無意味な活動になってしまう可能性があります。

テーラリングガイドは、標準プロセスのどの部分は全プロジェクトが必須として従わなければならないのか、どの部分は変更可能なのか、必須の部分を顧客からの要請等で遂行できない場合にはどのような手続きをとるのか、等を定めたものです。標準プロセスを、個々のプロジェクトに合致した形で適用し、実践するためのガイドとして情報を共有化しプロジェクトに周知することが推奨されます。

プロセス資産④ プロジェクトの定義されたプロセス

テーラリングガイドに沿って、標準プロセスを個々のプロジェクト向けに修正したプロセスを定義されたプロセスと呼びます。例えば、24時間連続稼働の基幹システムや人命に直結する機器の組込みシステム等、ミッションクリティカルなソフトウェアを開発する場合に、形式手法を導入して設計プロセスを見直すことや、テスト項目の設計基準やテスト終了の判断基準をより厳格にすることもあられるかもしれません。このようにプロジェクトの特性を考慮して標準プロセスをテーラリングしたプロセスが定義されたプロセスです。これらのプロセスは、将来別グループが同様の特性を持ったプロジェクトに取り組む場合の具体的なテーラリングの事例として活用できます。また、テーラリングによって大きな費用対効果が複数のプロジェクトで得られた場合には、テーラリングされたプロセスを標準プロセスに組み込むこともあり得ます。このような活用を図るため、定義されたプロセスをプロセス資産として蓄積、共有化を図ることが推奨されます

プロセス資産⑤ 手順/テンプレート

標準プロセスは、組織内の様々なプロジェクトに適用することを意図して制定されるため、作業の進め方についての記述はかなり抽象化して書かれているかもしれません。しかしながら、プロジェクト内の要員が一貫した方法で作業を進めるには、誰が、どのタイミングで何を実施し、その結果としてどのような成果物をどのような様式で作成するのかを明確にしておく必要があります。これらは、標準プロセスや定義されたプロセスを補完する手順書、あるいは成果物のテンプレートによって、明確にすることが可能です。個別プロジェクトで作成したこれ

らの情報をプロセス資産として蓄積、共有することで、他プロジェクトでの再利用や標準プロセスの強化等を図ることが可能になります。

プロセス資産⑥ ツール/技法

標準プロセスは、ある作業を実施する際に、組織として標準的に使用するツールや技法の使用を前提として定義しておく必要があります。また、プロジェクトで個別にツールや技法を選定する余地がある場合には、標準プロセスのテーラリングの一環として、それらの使用を前提に標準プロセスが見直されることとなります。しかしながら、詳細なツールの使い方や技法の基本的なコンセプトは、マニュアルやガイドといった形で文書化され、標準プロセスや定義されたプロセスから参照するほうが、使い勝手がよい場合があります。そのような場合には、それらのマニュアルやガイドをプロセス資産として共有することが望まれます。また、自製品、購入品を問わず、ツール自体を資産化し、必要に応じてリポジトリからダウンロードし、インストールしてプロジェクトで利用できるようにしておくといでしょう。

プロセス資産⑦ 実績データ

プロセス改善の目的は、組織目標の達成にあります。上位の組織目標は、定性的に提示されるかもしれませんが、その達成状況をトレースするには、目標を具体化し何らかの定量的な指標並びにその目標値を設定し、実績値と対比することが必要になります。個々の領域のプロセス改善の効果、およびプロセス改善活動の総括的な効果として、どのような指標を用いるのかを詳細に定義し、各プロジェクトに周知して、収集されたデータを組織レベルで蓄積することが望まれます。これらのデータは、定期的に組織レベルで集計、分析、報告し、プロセス改善の成果を顕在化させることによって、要員のモチベーションの向上につながることが可能です。また、期待通りの効果が表れていない領域では、標準プロセスを見直す場合もあるでしょう。組織の生産性基準、品質基準等を設定、公開して、より精度の高い見積りに活用することも可能です。さらに、個々の指標に関して、平均値と自プロジェクトとの差分や最高値と自プロジェクトとの差分等を比較できるようにして、他プロジェクトのプロセスを参考に、プロジェクトごとの自律的な改善を促すこともできるでしょう。ベンチマーキングに用いる場合には、自

組織内のデータに加え、ベンチマーク対象とする他組織の実績データを併せて提示することも、改善への動機付けや目標設定に有効です。

プロセス資産⑧ 事例

組織レベルおよびプロジェクトレベルでプロセスの改善に取り組むにあたって、参考になる成功事例、および教訓として学ぶべき失敗事例は、要員の改善に対する意識を高揚したり、自身が抱えるリスクを再認識することで、改善活動の活性化を促す効果があります。これらの事例は、必ずしも組織内に閉じて考える必要はなく、他社の事例も含めて共有化を図ることが望まれます。組織内の事例の共有化に関しては、プロジェクト完了後のプロジェクト報告書の作成と共有、成功事例および失敗事例の報告会の実施と資料の共有等が考えられます。また、社外事例の情報ソースとしては、業界紙、学会誌等の記事、シンポジウム等での発表資料、場合によっては特定の組織に対して実施したベンチマーキングの結果等が考えられます。

プロセス資産⑨ 教材

組織内またはプロジェクト内で標準的に用いるプロセスを定着化し、また、プロセス遂行に併せて利用する技法やツール等を効果的に活用するには、それらについての要員の理解を確実にし、実務を通してスキルを向上することが必要になります。特に新規のプロセス、技法、ツール等を導入する際には、クラスルーム形式やオンライン形式等での教育の実施を検討することもあるかもしれません。これらの教育に用いた教材は、新メンバの参入の際に再利用したり、既受講者が再度理解を確認したりできるよう、プロセス資産として共有化し参照可能にしておくことが望まれます。

以上に、プロセス資産の種類とそれらの活用目的を説明しましたが、プロセス資産として共有化を図るべき情報は、これらに限定する必要はありません。プロセス改善活動の活性化に向けて、より幅広い視点から資産の拡充に努めていくことが望まれます。

以下に、プロセス資産の蓄積と活用を活性化するための留意点を述べます。

① プロセス資産リポジトリの構造化

プロセス資産リポジトリの構築にあたって、組織として共有化を図るべき情報

の種類を明確にし、構造化し、利用者からの参照がしやすい形で情報を開示する必要があります。とくに、情報量が増えてきた段階ではこの点が重要であり、検索機能の付与等も考慮する必要があります。このような配慮がなく情報が蓄積されていくと、せっかくの有益な情報が実際には活用されないという状況に陥ります。

② 情報の有効性の維持管理

プロセス資産として蓄積された情報は、常に維持、更新が行われ、陳腐化した情報はアーカイブして直接参照できないようにする等の考慮が必要です。例えば、実績データに基づく組織の品質基準や生産性基準等は、現行の標準プロセスの下でどの程度のパフォーマンスが出せているかを示す指標なので、プロセス改善が活性化し、標準プロセスの改訂が活発に行われている段階では、それに併せた基準値の更新が必要になります。情報の有効性を維持管理するには、組織のプロセス資産リポジトリの管理責任者を設置して、その責任と役割を明らかにしておくことが必要です。扱う情報の量に応じて、標準プロセス、実績データ、内部事例、外部事例、教材等、情報の種類ごとに責任者を割り当てるのものを検討するとよいでしょう。

③ 情報提供者への動機付け

プロセス資産を有効に機能させるには、情報提供への動機付けが重要になります。たとえば、参照頻度の高い情報の提供者を報奨する制度を作る、高い実績を上げたプロジェクトの事例報告会を催す、事例の報告に対して有識者からの有役なフィードバックを得られる仕組みを作る等、情報共有の活性化の施策を検討する必要があります。

④ 開示範囲の設定

実績データ、事例等の情報は、場合によって顧客との契約によって開示が制限される場合があります。このような場合を想定して、情報の開示を制限する仕組みを構築しておく必要があります。また、個人ベースで収集されたデータをそのまま開示してしまうと、本来意図していなかった実務者一人ひとりの実績評価に用いられてしまう危険性があるため、このようなデータの扱いも留意が必要です。実績データの収集目的は、効果的なプロセス改善の推進ですので、その趣旨にそぐわないデータ活用は控えるべきでしょう。暗黙のうちに実績データが個人評価

に用いられるような場合、報告されるデータが実態と乖離していく可能性が大きくなります。

1.3 事例の共有化—使える情報から学べる情報へ

本書の主題は、「ベストプラクティスから学ぶ」という点にあります。組織の内を外を問わず成功事例を知るといのは、プロセス改善のきっかけを与える上で、または停滞気味のプロセス改善活動を活性化する上での、よい契機となるのではないのでしょうか。

ベストプラクティスに学ぶというのは、成功事例に紹介された仕事の進め方の見直し、技術やツールの導入、教育等によるスキル向上の仕方をそのまま導入するという点では必ずしもありません。組織やプロジェクトの規模、開発している製品の種類（業務アプリケーション、ミドルウェア、OS、組込みソフト等）、製品の特性（求められる品質の特性や水準、安全性への配慮の度合等）、外注依存の比率等によって、事例に紹介されているアプローチが合わないケースもあります。したがって、リポジトリに蓄積された事例の中から参考になりそうな事例を検索するために、組織特性、プロジェクト特性、製品特性等を明確にし、絞込みのための情報として明確にしておくことが重要です。

また、プロセス改善の事例は、単なる導入技術の紹介や標準プロセスの紹介に留まらないという側面もあります。このような事例を、ここではストーリー化された事例と呼びます。ストーリー化された事例には、改善への取り組みの背景、組織文化、設定した目標、直面した課題、課題解決のための施策、定着化のための工夫、得られた改善効果等が明らかにされます。またストーリー化された事例は、紹介されたプロセス、技術、ツール等がそのまま自組織の特性に合致しなかったとしても、その改善のアプローチ自体に感銘を受けたり、動機付けられたりするようなものを指します。

実際の組織やプロジェクトでの改善の取り組みを、上述のような形でストーリー化するのには手間のかかる作業です。しかしながら、ノウハウや教訓を次の世代、その次の世代へ引き継いでいくためにも、成功事例や失敗事例をストーリー化して共有するアプローチが推奨されます。

1.4 他組織に学ぶ

長年ソフトウェア開発のプロジェクトに従事していたとしても、自分自身が遭遇し経験する成功経験や失敗経験は極めて限られたものにならざるを得ません。組織内の他プロジェクト並びに組織外に目を向けると、現在自分たちが遭遇している課題が既存の技術導入によって解決されている、同種のソフトウェアをまったく異なるプロセスで開発して成果を上げている、というような例もあるでしょう。また、自分達と同領域の仕事で、極めて高い品質、生産性の水準を上げている組織があることに気がつくこともあるでしょう。このような気付きをきっかけに、さらに対象組織の業務プロセスを深く調査し、自プロジェクトまたは自組織のプロセス改善のヒントが得られることもあるでしょう。

しかしながら、とくにプロジェクトで開発に従事している管理者や開発者にとって、役に立つ社内外の事例を自身の裁量で収集し、プロジェクトまたは組織の改善活動を自律的に起動するというのは困難を伴います。したがって、組織レベルでプロセス資産リポジトリを構築し、参照するとよい情報を集約し、開発サイドからの参照を容易にすることが役に立ちます。

プロセス改善の主役は、ソフトウェア開発に従事している方々に他なりません。プロセス資産の効用は、これら改善の当事者に気付きや動機付けを与える点にあります。それがプロジェクトの自律的な、または組織的な改善のアクションに結びついていけば、プロセス資産リポジトリの構築の目的が達成されたこととなります。

プロセス資産リポジトリの管理者においては、自組織内の情報の蓄積に加えて、他組織の情報にも目を配り、有益な情報を組織内に展開することが重要な役割となります。とくに、表2に示すようなカテゴリの情報を継続的に監視して組織内で共有化を図る必要があります。

1.5 外部への情報発信の意義

事例に関して、他組織の取り組みに刺激を受けて改善に取り組み、所期の成果を上げた経験があれば、自らの取り組みを事例として報告し、自組織の他プロジェクトや将来のプロジェクトに貢献しようという意識が出てくるのではないでしょ

表2 継続的に監視すべき情報

カテゴリ	内 容
ツール/技法	<ul style="list-style-type: none"> ・ ツール展示会への参画 ・ 業界誌/学会誌/書籍等の調査 ・ ツールベンダへのデモ/カタログの請求
実績データ	<ul style="list-style-type: none"> ・ リサーチ会社の公開している関連データの調査 ・ リサーチ会社への調査依頼 ・ 情報処理関連のデータ白書の調査
事 例	<ul style="list-style-type: none"> ・ コンファレンス、シンポジウム等への参画 ・ 業界誌、学会誌、調査機関・研究機関のレポートや書籍等の調査 ・ 調査会社への調査依頼 ・ 他社ベンチマークの実施
教 材	<ul style="list-style-type: none"> ・ 教育会社のコース紹介の調査 ・ 受講した教育の報告 ・ 入手した教材の共有

うか。自組織内での実績データ共有、事例共有等を活性化するためには、プロジェクトメンバー各々が自身の担当しているプロジェクトでの役割を果たすとともに、その過程で自らが得たノウハウや教訓を報告して、組織全体の運営に貢献するという意識が重要です。また、組織としてもそのような活動の意義を認め、それに必要な工数の確保、報告の場の提供、優れた報告に対する報奨等を制度化することが必要です。

さらに、自組織の事例を、外部に公表するというのも積極的に推し進めるべき活動です。このような活動が活発化することで、業界全体の改善活動が活性化され、国際的な競争力の底上げにつながっていくことが期待されます。業界貢献というだけでは、個々の企業の事例報告の動機付けとしては必ずしも十分ではないかもしれませんが、公表する側のその他メリットとしては、自組織の改善活動への取り組みを公表することで自組織の業界内でのプレゼンスが向上する、報告の場で他組織の参加者から様々な意見を得ることによりさらなる改善に向けたヒントが得られる、等が挙げられます。

第2章 プロセス改善と ベストプラクティス

プロセス改善研究部会では、我が国のソフトウェア業界の国際競争力の強化を図るべく、プロセス改善活動の啓蒙、国際規格に準拠した日本版プロセス評価モデルと評価改善方式の開発、ベストプラクティスの共有、海外のプロセス評価改善スキームとの整合と協調等の活動を推進しています。本書は、ベストプラクティス共有を担当するワーキンググループの成果をまとめたものです。

事例をプロセス改善に活かすことを考える際、「失敗から学ぶ」という側面と「成功から学ぶ」という側面があります。

本書では、「成功から学ぶ」という側面に焦点を当て、ISO/IEC 15504 第5部やCMMI等のアセスメントモデルを用いるか否かを問わず、ソフトウェアプロセス改善に取り組んでいる、またはこれから取り組もうとしている方々にとって参考になるベストプラクティスを紹介しています。

この章では、ソフトウェアプロセス改善におけるベストプラクティスの考え方を明確にした上で、ベストプラクティス情報の調査方法や活用方法を説明します。

2.1 ベストプラクティスとは

組織目標の達成や直面している課題解決に向けた業務プロセスの改善に取り組むとき、先進企業の事例やノウハウを参考にすることは、改善活動を効果的かつ効率的に推進する上での有効なアプローチと考えられます。

ベストプラクティスは、一般的には、最も効果的かつ効率的な実践の方法、または模範的あるいは最良の事例を指し、マネジメント用語としては、「先進企業の成功事例、あるいは世界で最も優れていると考えられる業務プロセス／業務推進の方法／ビジネスノウハウを指し、自社の業務改革の目標となる最高水準の状態」等と定義されています。

ベストプラクティスをプロセス改善に活用することを考えるとき、すべての組

組織にとって、唯一無二のベストな実践方法があるのでしょうか。ソフトウェア開発の領域で考えるなら、ベストな構成管理、ベストな要件定義等があるのでしょうか。ある限られた領域のソフトウェア開発において、現状考えられるベストに近い方法は見つかるでしょうが、万能かつ唯一無二のベストはない、というのが正しい考え方だと思われます。つまり、何をベストと見るかは次のような要因で変わってくるということです。

- ・組織の風土、運営方針、体制、規模
- ・開発している製品のドメインや特性
- ・顧客の要求
- ・プロジェクトの特性

従って、ベストプラクティス共有による効果的プロセス改善の促進を考えるときには、ベストプラクティスをもう少し広義に解釈する必要がありそうです。さもなければ、ベストプラクティスは個別の組織、極端に言えば個々のプロジェクトにとってのものでしかあり得なくなり、ベストプラクティス共有という考え方自体が成り立たなくなります。

本書では、ベストプラクティスを「他組織が参考にし得る先進組織の実践」と捉え、選定しています。選定基準は次のような特徴を1つ以上備えていることとしています。

- ・業務プロセスを変えることによる効果が顕在化している
- ・原理原則に則って継続的にプロセス改善活動に取り組んでいる。つまり、改善活動が一過性の取り組みではなく、組織文化として定着している
- ・モデル等にかかれた要求事項、推奨事項の表面的な実践ではなく、改善効果の顕在化を目指した創意工夫が見受けられる

その上で、プロセス改善研究会では、過去の発表事例等を調査して選定を行いました。その範囲での推奨事例であることをお断りしておきたいと思います。

2.2 プロセス改善におけるベストプラクティスの活用

近年、ソフトウェア開発の分野では、ISO/IEC 15504 第5部、CMMI 等のアセスメントモデルを活用し、業務プロセスの改善に対して体系的かつ継続的に取り組

む組織が増加しています。たとえば、CMMI では、組織の能力を未成熟な状態から成熟した状態に向上するための道筋を組織成熟度レベルまたはプロセス能力レベルという概念を導入して指し示し、次の目標レベルに到達するために実践すべきプラクティスを定義しています。CMMI では、そのモデル自身を「製品とサービスに適用される開発と保守の活動を取り上げるベストプラクティス」と定義しています。つまり、アセスメントモデル自体が、業界の中での実践を通して効果性が確認されたベストプラクティスの体系であり、モデルを活用した改善アプローチ自体が、ベストプラクティスを活用した改善アプローチとなっているのです。

しかしながら、改善施策検討の拠り所としてアセスメントモデルを見ると、デファクトまたはデジュール標準として制定されたモデルは、汎用性を考慮して作成されており、プラクティスの定義の抽象度が高くなっています。定義された個々のプラクティスを、ベストプラクティスとして実践する際に充足すべき推奨事項が、典型的な作業成果物、実践のポイントに関する補足説明、プラクティスを一段詳細化したサブプラクティス等の形で記述されているものもありますが、個々の組織やプロジェクトでの実践という視点から見ると、抽象度は高いレベルに留まっています。

ある業界または業務領域におけるモデルを制定するという立場からすると、この抽象度の高さは致し方ないものであるとともに、抽象度を上げてプラクティスを定義していることによって、個々の組織やプロジェクトでの最適化に向けた工夫の余地が生まれるというメリットがあります。つまり、モデルで定義されているプラクティスは、ベストプラクティスから導き出された改善のポイントを指し示すものと捕らえるべきです。具体的な実践の方法は、モデルに記載された改善のポイントを踏まえつつ、組織や製品の特性に応じて、最適な方法を検討し、展開するのがモデルベースのプロセス改善の基本といえます。

この基本にそってプロセス改善活動に取り組むにあたり、より抽象度の低い、つまり個別具体的な事例としてのベストプラクティスを活用するための活動の流れとポイントを図1にまとめます。

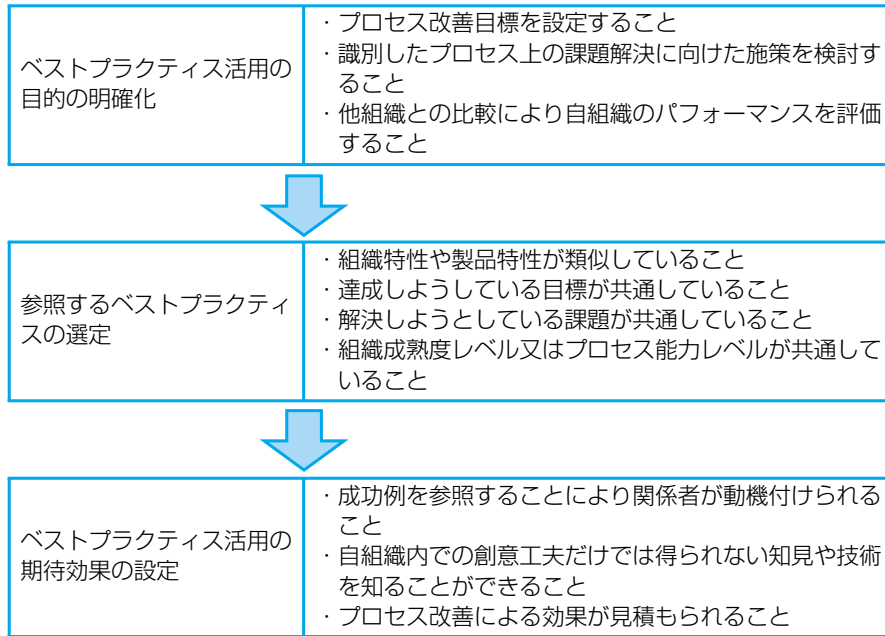


図1 ベストプラクティス活用の流れ

さらに、業界全体の底上げという観点からのベストプラクティス活用を考えると、業界におけるベストプラクティスの蓄積と共有の仕組みを構築することが有効と考えられます。その上で、技術の進歩に伴って、個々の業務領域において実践が期待されるプラクティスに変化が生じたなら、ベストプラクティスを抽象化してモデルを見直すことが重要と考えられます。

2.3 ベストプラクティスの調べ方

プロセス改善の推進に際して、ベストプラクティスを有効に活用するには、改善対象となる主要なプロセス領域を特定し、当該領域の改善に有効な情報を収集し、参照することが必要になります。

まずは、体制面で、改善領域ごとに外部の技術動向や社内外のベストプラクティス情報を継続的に監視する役割を割り当てることが重要です。主要なプロセス改

善領域ごとに PAT^{*1}を組織し、各 PAT にベストプラクティス調査の責任を与えるのも1つの方法です。

その上で、各担当者またはチームは、次のような手段によって、情報を入手し、自組織での情報共有や参考にすべき技術やアプローチの自組織への導入等を検討する必要があります。

- ① コンファレンス、シンポジウム等への参画
- ② 業界誌、学会誌、調査機関・研究機関のレポートや書籍等の調査
- ③ 調査会社への調査依頼
- ④ 他社ベンチマークの実施

①、②に関して、近年は情報の Web サイトへの公開、検索エンジンの活用等によって、ベストプラクティス情報の入手は比較的容易になってきています。プロセス改善研究部会では、ベストプラクティス活用による効果的なプロセス改善活動の一層の活性化を図るべく、継続的なベストプラクティス情報の調査と発信ならびにソフトウェア開発のベストプラクティス情報の収集を容易にするためのポータル構築等を推進していく予定です。

*1 PAT：プロセスアクションチーム：Process Action Team

第3章 ベストプラクティス集

本章では、プロセス改善研究部会で選定した10件のベストプラクティスを紹介します。各ベストプラクティスの概要は表1のとおりです。

表1 各社プロセス改善事例の概要

事例番号	タイトル	組織	概要
1	設計からはじめる見逃しバグの防止～F2Tの活用～	株式会社日立ハイテクノロジーズ ナノテクノロジー製品事業本部	独立検証組織である品質保証部で検出したバグの原因を分析した結果、設計部門で実施したテストの漏れと網羅性の欠如に起因したバグの比率が高いことが判明。その対策として、機能仕様の作成とテスト仕様の作成を、同時並行で対応付けを明確にしながら実施。その結果、テスト項目の漏れがなくなり、品質が向上。
2	生産性を5年で2.5倍に！～一人ひとりを改善の主役にするKAIZEN塾の設置～	株式会社富士通ソフトウェアテクノロジーズ	富士通ソフトウェアテクノロジーズは、その前身である富士通プライムソフトテクノロジ時代の2003年に「5年後に生産性を2.5倍にする！」という狙いを掲げ、全社規模のNPS (Nagare Kaizen Profit System) 活動に取り組む。NPS活動は、トヨタ自動車の強みは何であるのかを既存のプロセスモデルに捉われずに考え、改善によりムダを徹底的に排除する「トヨタ生産方式 (Toyota Production System)」に学び、現場をよくしようとする文化・風土を作り、生産性を改善していく活動。

(つづく)

(前頁より)

3	究極の高品質ソフトウェア開発プロセスを目指して～ソフトウェア独立検証および妥当性確認 Software IV&V～	独立行政法人宇宙航空研究開発機構 (JAXA)	究極の高品質ソフトウェア開発プロセスを目指して、SW IV&VとSQAを組み合わせた。SQAは開発部門から独立した組織が、開発プロセス規定に則って成果物が作成されたことを保証する活動。一方 SW IV&Vは、開発部門から独立した組織が、成果物の安全および品質（記述内容の論理性、一貫性、妥当性等）の検証および妥当性確認を行うための技術・枠組み。
4	生産性原理を具現化するプロセス改善～生産管理システム「ACTUM」の活用～	株式会社ジャステック	全社一体となってプロセス改善を前提とした生産性原理の具現化を図り、見積りからプロジェクト管理まで一貫した指標（開発量、生産性変数および環境変数）を持ち、独自の考え方に基づいた予実差異分析を行うことで、開発局面ごとのプロセス改善機会の創出促進。プロセス改善の「計る化（定量化）」を目指し、環境変数を統計処理し変動幅の是正や安定性の確保に取り組む。
5	高品質ソフトウェア開発管理プロセスの実現～QMTXによる品質メトリクスの活用～	日本電気株式会社 システムソフトウェア事業本部	開発物の品質のばらつきが1つの問題となっており、その対策として、組織的にプロセス改善活動を進める取り組みを実施。CMMI レベル5のプロセスを土台に、データを収集するための仕組みと収集されたデータを管理部門だけでなく、現場も含めて分析するための仕組み、分析された結果をもとに組織的な改善を推進するための仕組みを構築し、日常的に対応を行うことで高品質なソフトウェアを確保。それによって、出荷後品質に加え、納期短縮や生産性が向上。

(つづく)

(前頁より)

6	マネジメント自らが予防のための行動を取り、問題が起こりにくい組織に～Preventive Managementによる実現～	情報技術開発株式会社	SW-CMMのレベル3を達成した後、さらに赤字プロジェクトの撲滅を目指して、「Preventive Management」というキーワードのもとに展開した活動。プロジェクト計画からプロジェクト完了プロセスに加え、案件引き合い段階から、受注までのプロセスを定義し、それに準拠した活動を実施することで、全プロジェクトの黒字化に成功。
7	派生開発による品質および開発効率の向上～XDDP・USDMの徹底～	東京エレクトロンソフトウェア・テクノロジー株式会社 札幌テクノロジーセンター	従来製造拠点からの修正依頼に基づき、ソースコードを修正していたが、母体のドキュメントがなく、修正依頼→コード調査→コード修正→テストというプロセスを実施していた。その結果、修正漏れが起因となった不具合が発生し、修正コードを納品してからも不具合対応に追われているという有様となり、このような状況を打破するために、XDDPを取り入れ、手戻りの少ないソフトウェア保守を行うプロセスを実現。
8	ソフトウェア部品による組立型開発と品質保証～基幹システムでのコードの再利用～	住友電気工業株式会社 情報システム部&住友電工情報システム株式会社システムソリューション事業本部	エンタープライズ系システム開発の工業化を目指して、属人性を排除した設計手法の確立、部品組立型システム開発、統計的手法を使った品質管理を実践。20年を超える継続的な改善への取り組みの結果、大幅な開発工数とコストの削減、並びに出荷後欠陥数の削減を実現。

(つづく)

(前頁より)

9	全社レベルでのプロジェクト管理の改善 ~段階別にツールを導入し作業内容変更の最小化をはかる~	日立ソフトウェアエンジニアリング株式会社	プロジェクトの管理項目、各項目の評価方法等は規格化されているが、実際に使う管理ツールやデータの管理方法等は各プロジェクトの実情に合わせて決定可能。しかしそれが全社 PMO が各プロジェクトを監視する際の弊害になっていたため、全社で利用する統一的なプロジェクト管理ツールの導入を行う。導入は、まず簡単なシステムを導入し、システムのメリットを実証してシステム導入のコミットメントを得た後、システムの機能拡張をしてユーザを拡大、さらに操作性を改善し全社導入という、段階的な導入戦略をとることで、全社規模での導入に成功。
10	GDD (Genba Driven Development) ~現場主導によるアジャイル開発のすゝめ~	ソニー株式会社 PC アプリケーションソフトウェアセンタ設計部	開発現場のソフトウェアエンジニアの問題意識をきっかけにはじまったボトムアップでの改善実施事例。「eXtreme Programming」の手法を参照して、20人規模の開発プロジェクトにおいて、検証工程での不具合数の減少、レビューやリファクタリングによる知識の共有等を実現。

表2に、各事例が共通フレーム2007のどの領域の改善活動に該当するかを示します。

表2 各事例と共通フレーム2007の関係

ベストプラクティス事例 共通フレーム2007の プロセス/アクティビティ		事例1	事例2	事例3	事例4	事例5	事例6	事例7	事例8	事例9	事例10
		主プロセス	企画プロセス 要件定義プロセス 1.6.2 システム要件定義 1.6.3 システム方式設計 1.6.4 ソフトウェア要件定義 1.6.5 ソフトウェア方式設計 1.6.6 ソフトウェア詳細設計 1.6.7 ソフトウェアコード作成及びテスト 1.6.8 ソフトウェア結合 1.6.9 ソフトウェア適格性確認テスト 1.6.10 システム結合 1.6.11 システム適格性確認テスト 1.6.12 ソフトウェア導入 1.6.13 ソフトウェア受入れ支援								
保守プロセス 運用プロセス 管理プロセス											
組織プロセス	環境整備プロセス 改善プロセス 人的資源プロセス 資産管理プロセス 再利用プログラム管理プロセス ドメイン技術プロセス										
	文書化プロセス 構成管理プロセス 品質保証プロセス 検証プロセス 妥当性確認プロセス 共同レビュープロセス 監査プロセス 問題解決プロセス ユーザビリティプロセス										

BEST PRACTICE
事例1

設計からはじめる見逃しバグの防止

～F2Tの活用～

株式会社日立ハイテクノロジーズ
ナノテクノロジー製品事業本部

日立ハイテクノロジーズ ナノテクノロジー製品事業本部では、半導体検査装置等の組込みソフトウェアの開発を行っています。本活動の背景として、独立検証組織である品質保証部で検出したバグの原因を分析した結果、設計部門で実施したテストの漏れと網羅性の欠如に起因したバグの比率が高いことが判明し、その対策として、機能仕様の作成とテスト仕様の作成を、同時並行で対応付けを明確にしながら実施することで、テスト項目の漏れをなくすという取り組みを実施しています。

具体的には、システム方式設計とそれに対応するシステム総合テスト仕様書、並びにソフトウェア方式設計とそれに対応するソフトウェア結合テスト仕様書の開発を、同時並行で実施することで、作業のタイミングのずれによるテスト項目の設計漏れをなくすと共に、設計仕様とテスト仕様を同一文書内に併記することによって対応付けを明確にし網羅性を確保しています。この方式の基本アプローチは次のとおりです。

- ① Word のコメント機能を用いて設計書とテスト仕様書を同時に表示する
- ② 設計書の検証対象箇所ごとに ID を付与し、該当するテスト項目 ID にリンクをはる
- ③ 設計仕様 ID とテスト項目 ID の過不足を識別し、テスト項目の追加や仕様の見直しを行う

キーワード

- ・テスト
- ・トレーサビリティ
- ・組込みソフトウェア
- ・F2T

Function to Test の略称で、仕様設計者がその仕様の実装を確認するためのテストにも責任を持つという考え方に基づいて日立ハイテクノロジーズが自社開発したもの

- ④ テスト実施段階ではテスト結果もテスト仕様に記載し、仕様の検証状況をトレースする

1. プロセス改善活動の背景

1.1 改善方針

- 日立 HT 那珂事業所では、次の3つの改善方針で実施しました。
- ・検証できないソフトは無価値
 - ・機能設計者がテスト設計にも責任を持つ
 - ・品質を分割検証できるようにせよ

1.2 達成目標

- 達成目標としては、シンプルに以下を掲げました。
- ・ソフトウェア品質向上

1.3 プロセス改善推進体制

SEPG 相当スタッフにより行いました (SEPG 相当スタッフというのは、開発組織内で改善活動を推進するメンバ)。

<p>■株式会社日立ハイテクノロジーズ ナノテクノロジー製品事業本部 (本書で用いる略称：日立 HT)</p> <p>●ビジネスゴール</p> <ul style="list-style-type: none"> ・最先端を、最前線へ <p>●製品・サービスの概要</p> <ul style="list-style-type: none"> ：下記領域で最先端のハイテク製品を開発 ・エレクトロニクス 高分解能 FEB 測長装置、光学式ウェーハ 	<ul style="list-style-type: none"> 検査装置、微小デバイス特性評価装置 他 ・メディカル 生化学自動分析装置、免疫自動分析装置 他 ・バイオシステム DNA シーケンサ、高速液体クロマトグラフ 他 URL http://www.hitachi-hitec.com/
---	---

1.4 プロセス改善対象組織の規模

実施後6ヵ月で日立HT那珂事業所全プロジェクト数の80%に展開しました。

1.5 改善活動の経緯

改善活動を行うことになったのは、プロジェクトに下記の状況が見られるようになり、早急な解決が必要とされたからです。

- ・テスト工程の遅延、テスト項目の漏れによる納期遅れの発生
- ・品質保証（以降QA：Quality Assurance）が発見したバグは是正並びに再発防止を実施
- ・再発防止策を統計処理すると、テスト工程の改善の必要性が他フェーズの倍以上

1.6 主要な課題と改善のフォーカス

主要な課題と改善のフォーカスを表1に示します。

表1 改善の課題とフォーカス

課題	改善のフォーカス
①アウトソースにより設計者と実装者の分割	・オフショア開発等に際しての、発注内容に関する相互の認識誤りの発生抑制。
②上流になるほどそれに対応するテストまでの時間が空く。また実施を忘れる	・設計とテスト項目作成を同時並行で実施。
③設計文書とテスト文書が別のフォーム	・設計文書とテストレビューの突合せを十分行える環境の実現。
④開発物の複雑化、トレーサビリティの維持が困難、不可能	・テスト項目と設計仕様とのリンクの明確化。

1.7 顕在化した改善効果

日立HT那珂事業所における改善効果を表2に示します。

表2 改善効果

項目	改善効果
定量（事例）	・仕様OKだがテスト漏れ：11件→0件
定性（アンケート）	・効果があると感じた人が67% ・テスト項目と設計仕様とのトレーサビリティの明確化
定性（その他）	・設計書の質の向上 ・ツールによるプロセス支援 ・プロセスとして標準化されている ・仕様変更に伴う追加テスト項目漏れの防止

2. ベストプラクティス

2.1 参照モデル上でのカバレッジ

- ① 当該ベストプラクティスに対応するISO/IEC 15504 第5部のプロセス
 - ・システムアーキテクチャ設計（ENG.3 System architectural design）
 - ・ソフトウェア設計（ENG.5 Software design）
 - ・ソフトウェア試験（ENG.8 Software testing）
 - ・システム試験（ENG.10 System testing）
 - ・文書化（SUP.7 Documentation）
- ② 当該ベストプラクティスに対応するCMMIのプロセス領域
 - 技術解（TS：Technical Solution）
 - 検証（VER：Verification）
- ③ 当該ベストプラクティスに対応する共通フレーム2007のプロセス/アクティビティ
 - ・開発プロセス/システム方式設計
 - ・開発プロセス/ソフトウェア方式設計
 - ・開発プロセス/ソフトウェア適格性確認テスト
 - ・開発プロセス/システム適格性確認テスト
 - ・文書化プロセス

2.2 プラクティスの説明

2.2.1 プロセス改善への取り組みの背景

日立 HT 那珂事業所では、半導体、医療、バイオ分野における最先端の測定、制御装置の開発を行い、製品分野ごとにユーザオペレーション支援およびデバイス制御の2系統のソフトウェアを開発しています。同社では「最先端を、最前線へ」という旗印のもと、ニッチな分野での高度な要求に基づいて、いわばパイオニアとして、どこも実現したことのない新たな製品を開発することがこの組織のミッションです。また、組織の方針として、最終的な顧客満足を掲げ、未踏領域の新製品であっても顧客が不安なく製品を使えることを重視し、その実現に向けて、製品不良の撲滅に力を注いでいます。

品質向上に向けた継続的かつ組織的な取り組みとして、落穂拾い、小集団活動、CMM/CMMI ベースのプロセス評価改善等、さまざまなアプローチを取り入れています。また、これらの活動に対するトップ層の関心も高く、その展開を自ら推進するとともに、事例報告等の場ではトップも交えた熱い議論が交わされています。以下に説明する F2T も、現場における課題分析を背景に、小集団活動の一環として構築された手法です。

これらの活動を背景とした業務プロセスの改善に当たっては、失敗や不良から学ぶという姿勢が貫かれています。不良は裏返すと他社に真似のできない業務ノウハウを見出すことのできる金脈になるという考え方が浸透しています。実際、過去の不具合解析の結果等から得られた開発ノウハウは、業務標準や設計テンプレートに埋め込まれてプロジェクトからプロジェクトに継承されています。

2.2.2 F2T が生まれた背景

日立 HT 那珂事業所では、製品の不具合撲滅に向けて、この組織では検査に力点を置いています。とくにソフトウェア開発の完了時、また製品開発完了時には、不具合の見逃しを防ぐべく QA による独立検証を実施しています。しかしながら、ソフトウェアおよび製品の最終段階での検査への過度の依存は、後戻り工数の増大による納期遅延を引き起こす可能性があります。

この問題に対処すべく、日立 HT 那珂事業所では、上工程での不具合の作りこみ

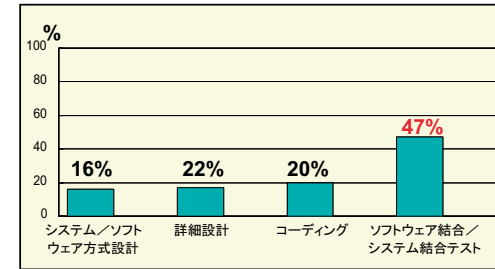


図1 工程別の改善点の比率

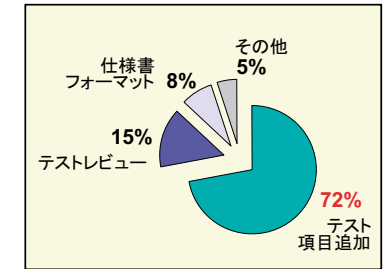


図2 テスト工程における改善施策の分類

を削減すべく、設計ノウハウを埋め込んだ設計テンプレートを用いて誤りの少ない作業を促進しています。また、しかるべきボリュームのレビューを実施して次工程および下工程への不具合の流出を防いでいます。

日立 HT 那珂事業所で、F2T を考案した部門でも、このような取り組みを行って、設計重視で開発に取り組んでいましたが、少なからず上工程での不具合の見逃しが生じ、テスト工程の実施の遅延により納期を遅らせるケースが散発していました。

そこで、QA による検査で見えられた不具合再発防止策を分析してみたところ、図1および図2のような結果が得られました。

この結果から、発生した現象に対して、予防的な措置を講じる必要のある工程はテスト工程が半数近くを占め、その約7割がテスト項目の漏れをなくすことによって防ぐことができるということが明らかになりました。

さらに、このような現象が発生している原因を掘り下げてみた結果、以下のような課題が明らかになり、F2T の開発が行われました。

- ・アウトソースによる設計者と実装者の分割によって、設計の意図が実装者に十分伝わらず、実装したソースコードのテスト観点に漏れや誤りが発生
- ・V字型モデルによる開発では、上流の仕様になればなるほど、その仕様に対するテストまでの時間が空くので、何をどのような基準に照らしてテストしなければならないかを忘れてしまう
- ・設計仕様とテスト仕様の様式は決まっているが、そのつながりが見通せるよう

な様式になっていない。それらの突合せはテスト仕様のレビューで行われるが漏れが発生

- ・開発するソフトウェアおよび製品が複雑化しており、成果物間のトレーサビリティを維持するのが困難

2.2.3 F2Tとは

F2Tは、Function to Testの略称で、仕様設計者がその仕様の実装を確認するためのテストにも責任を持つという考え方に基いて日立HT那珂事業所が自社開発したツールでWordのマクロ機能を使ったものです。F2Tは、前述の課題を解決のために「必要十分なテスト項目の抽出」を可能にし、また「テスト項目網羅の効率的な確認手段」を提供しています。機能仕様とテスト仕様の同時設計プロセスを課題解決の基本施策としています。

機能仕様とテスト仕様の同時設計プロセスでは、システム方式設計とそれに対応するシステム総合テスト仕様書、並びにソフトウェア方式設計とそれに対応するソフトウェア結合テスト仕様書の開発を同時並行で実施することによって、作業のタイミングのずれによるテスト項目の設計漏れをなくします。また、それらを仕様設計者が同時に行うことでコミュニケーションギャップによるテスト項目

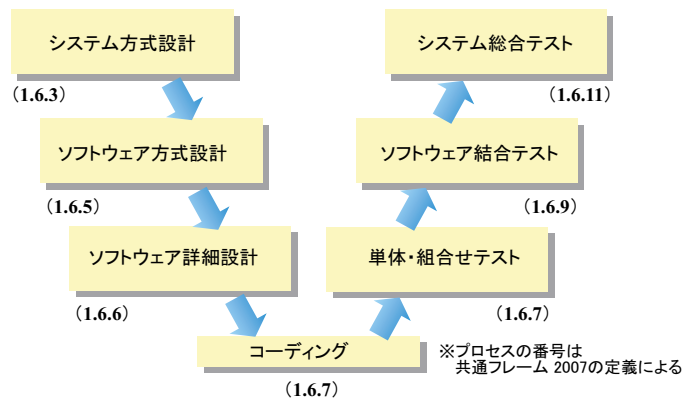


図3 ソフトウェア開発のライフサイクル

設計に漏れがなくなります。

改善施策の実現に当たっては、組織基準で定めた文書作成ソフトに、コメント機能を活用したツールを内製し組み込み、開発、展開、定着化の負荷を極力抑えています。

2.2.4 F2Tにより変わるテストの位置づけ

日立HT那珂事業所では、図3に示すようにシステム方式設計からシステム総合テストまでの作業をウォーターフォール型のライフサイクルに沿って実施しています。また、対応付く設計とテストをV字型に配することで、どの段階の仕様に基づいて成果物のテストを実施するかを明確にしています。

ウォーターフォールモデルのメリットは、一連の開発過程での関所が明確になることです。このモデルに沿った開発を成功裏に遂行するには、各工程の作業者が「品質の作りこみ」を意識し、工程内での不具合の抽出と修正を徹底的に行って、次工程に不具合を持ち込まないことが重要です。不具合が多く見逃されてしまうと検査工程から上工程への後戻り作業の負荷が増大し、その対処に工数が取られて、納期の遅れや工数の超過につながるようになります。

図4および図5に、テスト仕様作成作業についてF2T導入前後の位置付け変化

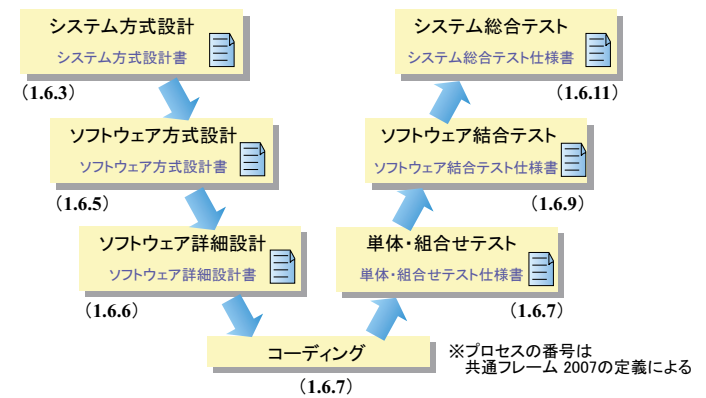


図4 機能仕様とテスト仕様の同時設計プロセス（変更前）

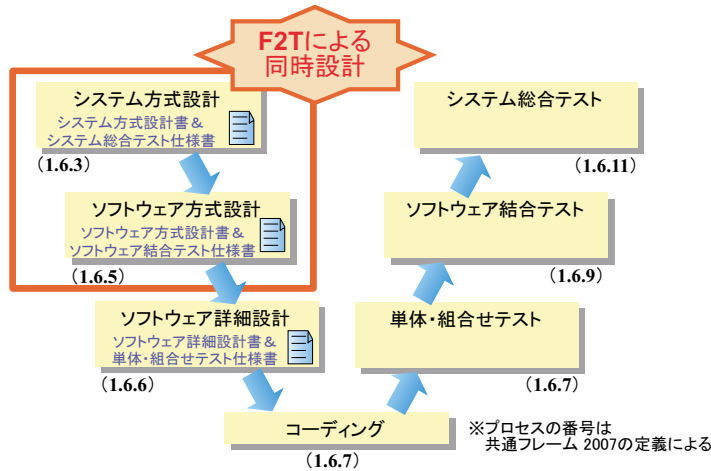


図5 機能仕様とテスト仕様の同時設計プロセス (変更後)

を示します。

ソフトウェアやシステムのテスト仕様を図4のようにテスト実施の直前に決めた場合、テスト仕様の作成時点になって設計仕様の誤り、抜け、曖昧さ等に気づき、設計段階からの作業をやり直さざるを得なくなる場合があります。

F2Tで実践している設計とテストの同時並行プロセスを採用すると、テスト仕様作成のタイミングが図5のようになり、早期に設計仕様の誤りや不備を認識して工程内で設計上の不具合に対処することが可能になります。

2.2.5 F2Tにより変わるテスト実施体制と役割

この組織では、ソフトウェア開発の終了段階並びにシステム開発の終了段階でのQAによるテストを実施しています。F2Tは、仕様設計者がその仕様の実装を確認するためのテストにも責任を持つという考え方に基づいて開発されたものですが、QAによる品質保証の必要性が皆無になる訳ではありません。テストには、成果物が設計仕様を実現していることを確かめる検証の視点と、利害関係者がその製品に求めているニーズを満たしているかを確かめる妥当性確認の視点があります。F2Tの導入によって、設計者は検証の観点からのテストのみならず、妥当

性確認の観点からのテストにも責任を持つという効果が出ています。その上でQAが客観的テストを行い、高品質化を目指しています。

また、この組織では、工程内で不具合を効果的に摘出するために、関係する開発部署およびQAグループ等のメンバがレビューに参画しています。その目的は、多様な知見を活用することで、思い込み等による不具合の見逃しをなくすことです。F2Tは、設計書の構造化や見易さにも効果を発揮します。それによって、成果物を開発した当事者以外のメンバが設計仕様の内容を正しく理解し、効果的にレビューを実施することが可能になります。

2.2.6 F2Tによる設計～テスト実施までの手順

ここでは、図6に示すフローに沿って、F2Tを利用すると設計からテストまでがどのような進め方になるのかを説明します。

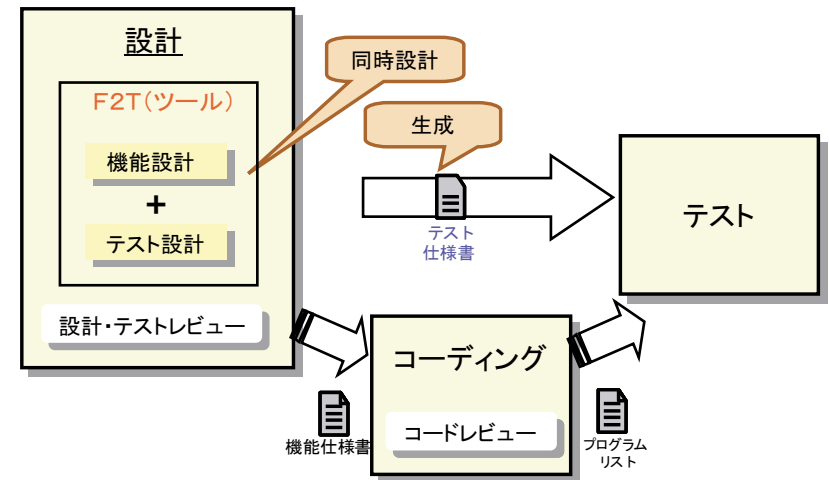


図6 F2Tを利用した新しいワークフロー

ステップ1 機能仕様とテスト仕様の同時設計

F2Tのプロセスでは、設計書とテスト仕様書の作成はいずれも設計者の責任で作成します。設計者が設計書の作成を終えたら、Wordのコメント機能を立ち上げて、テスト設計用のフレームをコメント表示欄に並列表示します。そして、相互

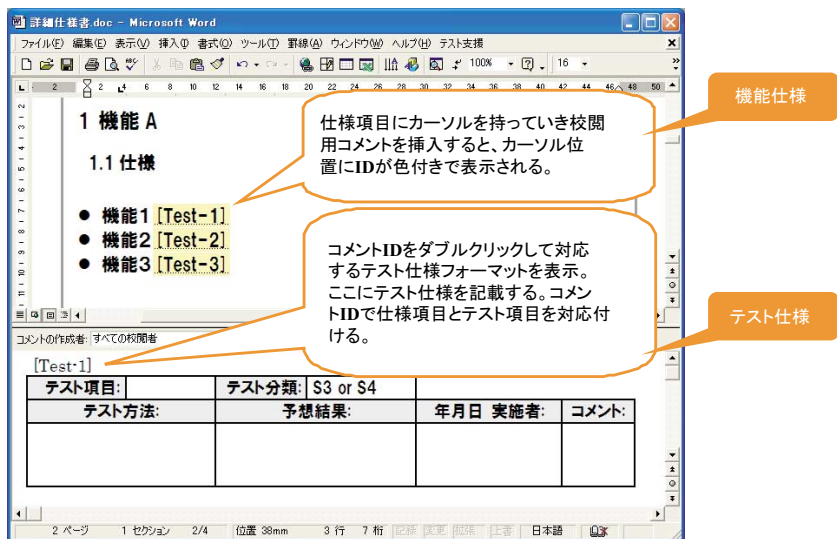


図7 F2Tによるドキュメント作成

のトレーサビリティを意識しながら、テスト設計作業を進めていきます。具体的には、図7に示すように、設計書の検証対象箇所ごとにコメント機能を利用し、テスト仕様を挿入していきます。設計仕様とテスト項目の対応は、コメントIDで関連付けられます。この同時並行作業を進める中で、設計書の抜けや不備に気付いたなら設計書を改訂し、併わせて改訂内容に合致したテスト項目を、リンクを明確にした上でテスト設計用のフレームに追記、訂正します。

ステップ2 トレーサビリティおよびテストボリュームの検証

次に、Wordに埋め込んだマクロを用いて、設計したテスト項目の総数を確認します。設計者は、テスト項目ごとに設計されたテスト件数の十分性を確認し、件数の不足している機能仕様に対するテスト項目の追加をのステップ1に戻って実施します。

ステップ3 テストの実施

テストの局面に至ったなら、テスト仕様に沿ってテストを実施し、コメント欄のテスト仕様にテスト結果を記述して、仕様の検証状況を確認します。

2.2.7 F2T導入によって得られた効果

F2Tは、現場のかかえる問題を解決すべく、小集団活動の一環として創出されたテスト方式です。この組織の強みは、現場主導のプロセス改善を組織的かつ継続的に展開している点にあります。F2Tの展開にあたっては、実施方針、手順、事例等を整備した上で、その方式を組織標準プロセスに組み込んでいます。また、開発者全員を対象にした教育を実施して、普及開始から6ヶ月で80%のプロジェクトに展開しています。

F2Tをうまく活用しているプロジェクトでは、トラブルシューティングの頻度が確実に減っています。表3は、あるプロジェクトのF2T適用前後の適用効果を示しています。

表3から、テスト項目設計漏れによる不具合の見逃しが11件から0件と大幅に改善されていることがわかります。

また、他のF2T適用プロジェクトメンバに対してオピニオンサーベイを実施したところ、F2Tがテスト項目の設計漏れの防止に効果があるとの答えが67%、テスト項目と仕様とのトレーサビリティの明確化に効果があるとの答えが76%を占め、現場からの支持の高さがわかります。

表3 F2T導入前後のテストプロセスに起因した不具合件数の比較

	(a) 仕様に記述あり だがテスト項目 なし	(b) 仕様記述不備に よるテスト項目 抜け	(c) 仕様変更に伴う テスト項目不十 分	(d) その他	合計
適用前	11	9	3	3	26
適用後	0	8	2	1	11

2.2.8 組織文化

日立 HT の組織文化は、日立グループの特色が出ています。具体的な取り組みとしては、日立グループの有名な「落穂拾い」、品質に対する徹底した議論、小集団活動等が挙げられます。「落穂拾い」の真髄は、技術者の驕りを指摘し、クレームに対して真摯に対応することです。これが、生産活動の精神的な根源になっています。また、小集団活動が活発で、事業所内発表にも必ず社長が参加することで、メンバのモチベーション向上に繋がっています。今回の F2T も、小集団活動の1つとして日立 HT 全体に発表された結果、高く評価されたため、外部への発表を決定しました。

日立 HT 那珂事業所の日常の改善活動は小集団活動で実行し、品質向上活動はトップダウンで方針を考え、実行されています。トップダウンとボトムアップの融合を見事に実施しています。

2.3 活用しているツール、技法

日立 HT 那珂事業所で使用している F2T の基本機能をまとめると次のようになります。

- ・ Word のコメント機能を活用して機能仕様とテスト仕様のリンク付けを行う
- ・ コメント欄に記述されたテスト項目を一まとまりのテスト仕様として抽出する
- ・ ファイル単位で、テスト項目数を抽出する

2.4 人材育成

日立 HT 那珂事業所では、F2T の教育を開発者全員に実施し、普及率は100%に達しています。教材には、F2T を実施するにあたっての方針と手順、並びに事例が盛り込まれています。教育担当者の言によれば、「従来方式のままでは自分たちが無駄をやっている」という認識を与えることを重視しているとのこと。

3. 学ぶべき点

日立グループで定着している落穂拾いの考え方に基づいて過去のバグ見逃しの実績データ分析から検証プロセスの改善に取り組んだ事例です。本組織では、モデルベースの改善と本例に見られるような課題ベースの改善の利点を融合しながら組織的なプロセス改善に取り組んでいて、F2T という課題解決の方式だけでなく、課題抽出から施策展開までの一連のアプローチが参考になります。また、検証方式の改善を Word の機能を用いて実装することで費用対効果を高めているとともに、現場への普及促進をしやすい点も見習うべきところがあります。

参考文献

株式会社日立ハイテクノロジーズ ナノテクノロジー製品事業本部：S-open & QuaSTom 共同イベント資料，2005年

株式会社日立ハイテクノロジーズ ナノテクノロジー製品事業本部：第3回世界ソフトウェア品質会議，発表資料，2005年

BEST
PRACTICE
事例2

生産性を5年で2.5倍に！

～一人ひとりを改善の主役にする KAIZEN 塾の設置～

株式会社富士通ソフトウェアテクノロジーズ

富士通ソフトウェアテクノロジーズは、その前身である富士通プライムソフトテクノロジー時代の2003年に「5年後に生産性を2.5倍にする！」という狙いを掲げ、全社規模のNPS活動※1に取り組みました。

NPS活動は、トヨタ自動車の強みは何であるのかを既存のプロセスモデルに捉われずに考え、改善によりムダを徹底的に排除する「トヨタ生産方式 (Toyota Production System)」に学び、現場をよくしようとする文化・風土を作り、生産性を改善していく活動です。また、当初約2年で「ムダ取り」を基本とするNPS活動を全社に展開するという方針を掲げています。

現場の身近な問題を洗い出し、自律的に改善していく問題解決手法により、全員参加型の改善活動を展開しています。しかし、改善を一過性で終わらせることなく、継続させるためには自律的な改善を実施できる人を育成し、改善に取り組む仲間、すなわち理解者を増やしていくことが大切です。また、改善の成果を経営層に見えるようにするためにも組

キーワード

- ・トヨタ生産方式
- ・人材育成
- ・KAIZEN 塾

NPS活動の推進役である伝道師を育成する機関

- ・ムダ取り
- ・NPS活動

あらゆるムダを徹底的に排除し、他の国々に真似のできない日本独自のソフトウェアのモノ作り標準を確立する運動

※1 NPS:Nagare Kaizen Profit System

織的な取り組みが重要です。そこで、改善を全社に浸透させるため、核となる人材（伝道師）の育成を目指し、「KAIZEN 塾」と呼ぶ社内トレーニングを行い、KAIZEN リーダの育成に邁進しました。

KAIZEN 塾では、自分達の仕事のやり方、価値観、方向性のばらつき等から生ずる問題に気づき、課題として取り上げて改善できる現場のKAIZEN リーダを組織的に育成しています。このように、改善し続ける組織を構築するために「人」という観点に着目し、NPS活動実践の核となる人材を育成することにより、改善を継続しています。

その結果、NPS活動は全社の理解を得ることに成功しました。この経験は、何年かの後にKAIZEN 塾出身者が同社の経営者になったときに、本当の成果として大きく花開くものであり、地道な改善活動を支える仕組みを維持する行為が、組織の文化、体質を作り出していくものでしょう。また、現場でのボトムアップによる改善活動が、経営者としてトップダウンで改善に取り組まなくてはならなくなったときに、本当に必要な仕組みを生み出す可能性があります。このKAIZEN 塾を作り出した力も、先輩経営者に、生産性向上のための強い意志があったために実現したものであることが推測できます。

プロセスモデルを組織の改善活動に当てはめて、モデルベースで取り組む改善手法ではなく、改善のための人材を育成し、現場にあった問題解決に集中しようとする姿勢は、現場でのやらされ感も少ないものです。また、改善は、いつでも、どこでもできる小さな改善と、組織だって大規模に行うことの両方が必要ですが、現場での小さな改善と、組織的なKAIZEN 塾という両方の取り組みをしていることが同社の特徴です。

株式会社富士通ソフトウェアテクノロジーズ (本書で用いる略称：FST)

●ビジネスゴール

顧客への深い理解と共感をベースとして、顧客の期待を超えるサービス/ソリューション/プロダクトを提供し続け、信頼されるパートナーとなります。とくに、スピードを重視します。

●製品・サービスの概要

- ・TRIOLE サービス & サポート
- ・Windows プラットフォームソリューション
- ・PI サービス、受託サービス

- ・BI コンサルティング
- ・アプリケーションソフトの開発
- ・情報アプライアンス (携帯電話、PDA、情報家電等)、組込み OS、ソフトの開発/サポート
- ・COBOL 再構築サービス
- ・ソフトウェアパッケージ開発、販売
- ・Web サイト診断/構築/運営
- ・eラーニング教材の作成/販売、eラーニングの運用代行サービス
- ・インターネットサービスプロバイダー (Web しずおか)

URL <http://jp.fujitsu.com/group/fst/>

1. プロセス改善活動の背景

1.1 改善方針

改善の方針は、次のとおりです。

- ・IT分野でトヨタ生産方式を参考に現場の改善を進める

1.2 達成目標

達成目標は、次のとおりです。

- ・超過勤務の低減とコスト低減（1人工単位での削減と新たな投資）
- ・最終ゴール…自律的に改善活動を推進する組織を形成して、生産性を上げる

1.3 プロセス改善推進体制

プロセス改善推進体制は、次のとおりです。

- ・経営企画室 NPS 推進室（専任7名）

1.4 プロセス改善対象組織の規模

プロセス改善対象組織の規模は、次のとおりです。

- ・全社（従業員数1,054名：2005年11月30日現在）

1.5 改善活動の経緯

富士通ソフトウェアテクノロジーズは、富士通インフォソフトテクノロジー、富士通プライムソフトテクノロジー、富士通ハイパーソフトテクノロジーの一部を母体として3社が合併し、2005年6月に設立されました。

合併直前の2003年に、株式会社富士通プライムソフトテクノロジーの当時の社長がトヨタ生産方式を参考にした新しい生産方式の確立を提唱し、トヨタグループ出身の改善コンサルタントを招いてトヨタ生産方式を学ぶことになりました。これが同社の改善活動の原点となっています。

あらゆるムダを徹底的に排除し、他の国々に真似のできない日本独自のソフトウェアのモノ作り標準を確立する運動をNPS活動と名付け、その運動の推進役で

ある伝道師を育成する機関としてKAIZEN塾がスタートしたのは、2004年4月のことです。

1.6 主要な課題と改善のフォーカス

5S（整理、整頓、清潔、清掃、躰）、ムダ取りを基本にし、改善ボードやタスクの見える化、一列待ち方式の導入により平準化を実現しています。

1.7 顕在化した改善効果

初年度の改善効果は、営業利益が前年度比18%増加しました（目標は年率17%）。

2. ベストプラクティス

2.1 参照モデル上でのカバレッジ

- ① 当該ベストプラクティスに対応するISO/IEC 15504 第5部のプロセス
 - ・人的資源の管理（RIN.1 Human resource management）
 - ・訓練（RIN.2 Training）
- ② 当該ベストプラクティスに対応するCMMIのプロセス領域
 - ・組織トレーニング（OT: Organizational Training）
 - ・組織改革と展開（OID: Organizational Innovation and Deployment）
- ③ 当該ベストプラクティスに対応する共通フレーム2007のプロセス/アクティビティ
 - ・改善プロセス
 - ・人的資源プロセス

2.2 プラクティスの説明

2.2.1 トヨタ生産方式

トヨタ生産方式は豊田喜一郎氏らが提唱し、大野耐一氏らによって体系化された工場における生産活動の仕組みであり、ジャストインタイム、自動化、7つのムダが柱となっています。トヨタ自動車では製造現場およびそれに付随するスタッフ部門だけでなく、間接部門や非製造業へも適用させており、トヨタ自動車を支

える1つの要因とされています。

トヨタ生産方式のソフトウェアへの応用は、名古屋地区では古くから取り組まれているようです。しかし、これまで必ずしも上手くいったわけではなく、理由としてはソフトウェアの品質の測定方法や、また測定値の利用方法を具体的に体系化できなかったことと、10年程度実践してきたベストプラクティスも、仕事の急激な増加や、開発環境の大幅な変更により、維持できなくなることがしばしばあったこと等が原因として挙げられます。

富士通ソフトウェアテクノロジーズでは取り組みをはじめたばかりですが、継続した改善を実践する意欲はKAIZEN塾によって継続され、今後とも品質を継続して測定することで何を測ればよいか、何をすればよいかの解が生み出されていくことでしょう。

2.2.2 ジャストインタイム

ジャストインタイムとは工程間の仕掛在庫を最少に抑えることを目的として、必要なものを、必要なときに、必要なだけタイムリーに生産する生産方式のことであり、カンバン方式ともいわれています。

ソフトウェア開発では、物による在庫がなく、必要なソフトウェアはネットワークで即時に提供可能であるため、ジャストインタイムは当たり前と考えることもできます。しかし、まだまだソフトウェア開発では、ジャストインタイムで試験プログラム、試験結果の提供はできておらず、改善の余地があります。

2.2.3 自動化

異常発見時には自ら停止する機能を備えた異常検知の仕組みと、徹底的なムダの排除を求められつつも現場で働く人の労働意欲を損なわないように配慮された自動化を、トヨタ自動車では自動化と呼んでいます。

KAIZEN塾の塾生は、現場の異常について、プロジェクトで活用されている改善ボードやタスク管理ボードに記されている内容をよく理解し発見する力を身に付ける努力を行っています。これにより、現場の反感をかうことなく改善を進めることができます。

2.2.4 7つのムダ

付加価値を高めない活動をムダと定義しており、「作り過ぎのムダ」、「手待ちのムダ」、「運搬のムダ」、「加工のムダ」、「在庫のムダ」、「動作のムダ」、「不良をつくるとムダ」を代表的な7つのムダとして挙げています。

FSTでは、個人、グループの作業を見える化し、「手待ちのムダ」、「動作のムダ」を排除する取り組みを行っています。

2.2.5 プラクティス推進者の横顔

NPS活動の推進者である高木氏は、「改善活動は人づくりからはじめることが肝要です」との信念を持っています。また、NPS活動の中で進められている伝道師育成による「人づくり」は、ソフトウェア技術者協会のSPIN-Meetingや、「実践!! IT屋のトヨタ生産方式」（風媒社）等で紹介されています。

2.2.6 NPS活動

NPSの組織の目的は、次のとおりです。

- ・現場の価値観、向かうべき方向性の共有
 - ・トヨタ生産方式を参考にして、仕事の流れを整え利益を生む方式の確立
 - ・NPS推進室の新設、KAIZEN塾の新設により組織体制の整備
- NPSの活動は、次のとおりです。
- ・具体的な原価低減目標値の達成を目指す
 - ・改善を進めながらの「人づくり」
 - ・現場からのボトムアップによる組織変革への取り組み

NPS活動で求められる人材とは、「小集団活動」、「現地・現物主義」、「実践主義」等のトヨタ生産方式の行動原則に基づき、ソフトウェア開発現場で改善の実践を持続・継続できる人です。

2.2.7 トヨタ生産方式採用の理由

FSTでは、人によって生産性に差がある要因として、次のものが考えられました。

- ・人の価値観、目指す方向性のばらつき
- ・プロセスのばらつき
- ・方法論やツールのばらつき
- ・現状否定における改善能力の欠如
- ・当たり前のことができていない

そして、これらの要因1つひとつを改善して現場を良くしようとする風土作りそのものが、トヨタ生産方式であると気付いたのです。

FSTでは、組織として力を発揮するため、世の中の流れに対して対応力を持つため、楽に仕事をするための組織文化・風土の再構築への取り組みに、トヨタ生産方式を採用にすることにしました。

2.2.8 KAIZEN 塾とは

KAIZEN 塾の目的は、NPS 活動推進のためのNPS 伝道師の育成であり、特徴は以下のとおりです。

- ・信念と情熱と忍耐をもって改善に取り組める人材育成
- ・異常を検知し、知恵を出し、実践し、周りを巻き込んで活動できる人材育成
- ・失敗を宝と捉え、二度と同じことを繰り返さないための解決策を考えることができる人材育成
- ・事例に対して「なぜ？」を5回繰り返し、真因が判明するまでとことん疑問を持つことができる人材育成
- ・後輩に対して動機付けをしてあげることができる人材育成
- ・物事の変化に敏感になり、小さな変化に気付くことができる人材育成
- ・実践主義、3現（現地・現物・現認）主義、人間尊重等の原理・原則に沿った人材育成
- ・KAIZEN 塾を卒業しないと幹部社員に昇格できない人事制度

KAIZEN 塾は、開発部門と事務部門に分かれ、さらにムダ取りと平準化のテーマ

に分かれて、半年間で現場の改善（開発のムダ、事務のムダ、開発の平準化、事務の平準化）に取り組んでいます。KAIZEN 塾は毎週金曜日に実施され、活動状況の報告とそれに対するフォローがあります。

2.2.9 改善に重要なこと

改善に最も重要なことは、改善しようとする人の「思い」だと高木氏は述べています。改善の必要を感じ取る能力(=改善しようという思い)がなければ、改善の行動は起こせるはずがないとまで言い切ることもあります。確かに、人から言われたことをその場だけの改善として取り組んでも、それは組織にとって一過性の処置でしかありません。組織(=企業)の発展の原動力は改善の持続力・継続力にあるので、組織に属する全員が改善しようという「思い」を常に持てるよう、育成していくことが大切です。高木氏は、人の価値観や夢等も変わりつつある中で、企業にとって必要な人材を確保するために、企業が自ら人を育てる以外に誰も育ててくれないと述べています。CMMI等のモデルベースによる改善は、どちらかというと標準化に注力されがちであり、「人づくり」が後回しになる傾向があるようです。

2.2.10 改善の効果

改善の効果としては、主に間接部門のムダ取りが効き、営業利益が前年度比で18%増加しました。

2.2.11 KAIZEN 塾の変遷

表1にKAIZEN 塾の発足以来の主要な出来事をまとめます。

2.2.12 KAIZEN 塾の今後の発展

高木氏は、KAIZEN 塾の卒業生（現場の改善リーダー含む）が組織の半数以上になる数年後には、企業の文化、風土となる土台ができ、組織の中で幹部社員や先輩が部下の指導をしていることを期待しています。そうなればKAIZEN 塾に必要性はなくなるため、解散の方向です。

表1 KAIZEN 塾史

時期	出来事
第1期 2004年4月～2004年9月	KAIZEN 塾を発足（名古屋）。 伝道師16名で発足。 ・TPSの思想理解 ・現地・現物の実践 ・KAIZEN 塾への参加を管理職昇格の条件化
第2期 2004年10月～2005年3月	伝道師8名、所属長8名。 前期の活動内容を継承し、以下を改善。 ・所属長もKAIZEN 塾に参加 ・伝道師育成制度の導入 ・TPS実践コンサルタントを導入 ・改善のモデル作り（ムダ取り、平準化） ・改善の見える化（KAIZEN ボード） ・運営モデル作り ・人材育成の重要性 ・効果と成果のモゴトリ（原価低減）認識
第3期 2005年4月～2005年9月	伝道師23名、所属長23名。 前期の活動内容を継承し、以下を改善。 ・指標管理方法の導入 ・分校（松本校／東京校）開校 ・交流会の定例化
第4期 2005年10月～2006年3月	伝道師24名、所属長24名。 前期の活動内容を継承し、以下を改善。 ・伝道師資格維持制度の導入 ・分校（本社／静岡校／神戸校）開校 ・NPS ルーム設置 ・部門 KAIZEN ボードの導入

しかしながら、ソフトウェア開発のムダは一時的な活動で切り切れるものではなく、全社に跨る継続的な取り組みが必要であり、KAIZEN 塾を改良した、次世代の活動も必要となることでしょう。

2.2.13 伝道師の現場での苦労話

NPS 活動開始当初には、現場より「なぜ朝礼をするのか?」、「なぜ作業実績を入力するのか?」といった疑問視の声がかなりありました。そこで、作業の押し

付けをするのではなく、目指すところはどこなのかを時間をかけて何度も繰り返して皆で話し合うことによって、「改善にはまず風土作り」の意識がメンバに浸透しました。

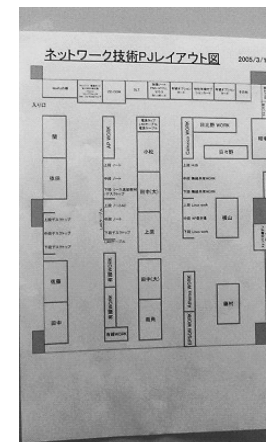
また、状況や成果が見えないと達成感がなく、負担だけが増えたように感じるので、NPS 活動の「見える化」にも取り組み、スキルマップや星取表等を壁に貼る等の工夫を凝らしています。

2.2.14 NPS 活動の事例

(1) 実験室設備改善の事例

実験室のあるべき姿について、以下の点に留意し、図1のようにレイアウト図を作成しました。

- ・スーパーマーケットのような陳列形式
- ・CD、DVD にはラベルを貼る
- ・小物収納のダンボール箱にもラベルを貼る
- ・足元のスペースには何も置かない



☆レイアウトのポイントと効果

- ・どの棚に何が置いてあるか
 - ・ロッカーには何が入っているか
 - ・机は誰が使用しているか
- 効果 ⇒ 見える化により検索時間短縮

図1 レイアウト図

2.2.15 ムダ取りカードによる改善（現場での小さな改善）

FSTでは、間接部門、直接部門を問わず全員に、毎日2件のムダを発見し、「ムダ取りカード」(図6、7)に記載することを義務付けられています。これは毎日休まず身の回りのムダを取り除くことで小さな改善を積み重ねて行き、大きな改善効果を得ようとするものです。

各人から出されたムダの解決は、チームで役割（発見者、対策者、実施者）を分担して対処します。

高木氏は、「毎日2件のムダを出し続けることはできないのでは?」と尋ねられたことがあるのですが、そのとき「2件のムダが出せない原因は、ムダ取りカードの仕組みにあるのかもしれない。」と答えています。ムダ取りカードそのものへの改善の可能性を考えることに、改善の真骨頂があるのでしょう。

ムダ・異常・気づき		提案・対策			
発見者	日	提案者	日	実施者	5回完了日
さん /		さん /		さん /	
ムダ単時	×	想定回数	×	人数	= 効果時間

図6 ムダ取りカード



図7 掲示されているムダ取りカード

2.2.16 ソフトウェアかんばん手法

ソフトウェアかんばんとは、トヨタの工場内における仕掛りかんばんを参考に、仕事（タスク）を紙に書き作業指示をホワイトボード等に貼って仕事をコントロールする手法です。このソフトウェアかんばんに、以下の4つのコンセプトを抽象化し、一般的な業務プロセスの改善に適合させた手法を、FSTではソフトウェアかんばん手法と呼んでいます。

(1) 見える化

プロジェクトのどんな異常でも検出できることがプロセスを良くすることにつながると考えます。したがって、良い報告のみをサマライズされたものは、安心はできても本質的にプロジェクトのプロセス改善のトリガーにはなりません。

見える化は、複数の人間に事実を表示することにより、異常の検出やリスクの共有化を図り、改善活動のニーズを引き出します。計画通りに進行している状態を見るためのデータだけではなく、異常を検出できるデータを見えるようにし、メンバ全員で異常が一目でわかるように工夫することが見える化の重要なポイントです(図8)。



図8 見える化の例

(2) 平準化

指示管理を集約し、担当間の能力と負荷のばらつきを最小化して、作業滞留時間のムダを省き、納期を短縮化して全体の処理能力を向上させます（図9）。

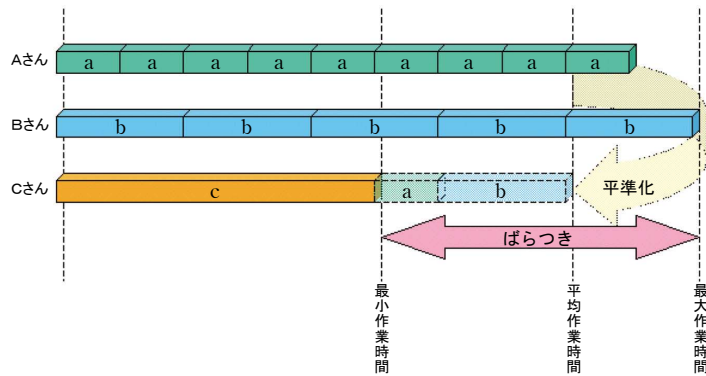


図9 平準化

(3) 一個流し

一個流しとは、ある単位（モジュール等）で品質を確保し、後工程の作業者が変わっても平準化を可能とするための考え方です。一個流しができていない場合は、部品、製品を製造するために必要となる規模見積りの基準（タクトタイムやサイクルタイム）が計れなくなる等、後工程に不良品質を送らないための予防（個々の部品、製品の品質を確保するためのテスト等）がしづらくなります。

(4) 一列待ち

作業が不規則に発生する場合、作業者に等配分して処理するよりも一列にして先頭から処理するほうが、納期が短くなるという待ち行列理論に基づき、作業をストア内（作業指示の保管場所のこと）に一列待ちキューとして管理する仕組みです。

2.2.17 正味作業と付帯作業の分類基準

正味作業とは、付加価値として顧客に対価を請求できる作業です。付帯作業は

正味作業を行うために必要な作業で、会議はその1例です。NPS活動では、この付帯作業をムダ取りとして削減することを目指しています。

2.3 活用しているツール、技法

ホワイトカラーの業務は「見えない」、「管理できない」という視点に立って、FSTではツールづくりと運用を行っています。またFSTという管理は、一般的に言われている管理とは少し違ってきます。

例えば、ホワイトカラーの改善目標や進捗状況を紙に書いて壁に貼ることにより、見える化し現場の異常検知を実現しています。普通紙を壁に貼るときに使う「はってはがせるスティックのり」も効果的なツールとして活用しています。

2.4 人材育成

FSTでは組織の文化・風土の形成を目的とした「KAIZEN塾」における教育の他、組織の技術・知識領域の形成を目的とした「業務ノウハウ」、「ITスキル」、「基本的な仕事のやり方」を教育する全社教育があります。

3. 学べき点

FSTの事例は、改善を継続するために、改善を実践する人の育成に着目し、改善活動の定着化に対する1つの答えです。プロセス改善を推進できる組織の構築を目指す上で、少なからず参考となるでしょう。

また、プロセスモデルを組織の改善活動に当てはめてのモデルベースでの改善手法ではなく、改善のための人材を育成し、現場にあった問題解決に集中しようとする姿勢は、現場の理解を得やすく、やらされ感が少ないのも良いことでしょう。

参考文献

高木徹／富士通ソフトウェアテクノロジーズ：ソフトウェア技術者協会 2006
発表資料

『ソフトウェア会社のトヨタ生産方式導入から学ぶ』～FST 全社改善活動事例に見る伝道師育成による「人づくり」～, 2006年

http://www.sea.or.jp/SPIN/Reports/NPS_presentationTPS_060508_SPIN_0620R_pg6hc.pdf

山崎勲／富士通ソフトウェアテクノロジーズ：ソフトウェア技術者協会 2006
発表資料

『NPS 伝道師活動報告』, 2006年

http://www.sea.or.jp/SPIN/Reports/NPS_ActivitiesTPS_060508_SPINpg6hc.pdf

富士通プライムソフトテクノロジ：実践!! IT屋のトヨタ生産方式, 風媒社, 2005年

BEST
PRACTICE
事例3究極の高品質ソフトウェア開発プロセスを目指して
～ソフトウェア独立検証および妥当性確認 Software IV&V～

独立行政法人 宇宙航空研究開発機構 (JAXA)

SW IV&V と SQA の組み合わせにより、信頼性の高いソフトウェア構築を実施している、宇宙航空研究開発機構 (JAXA) を取り上げます。

SQA は開発部門から独立した組織が、開発プロセス規定に則って成果物が作成されたことを保証する活動です。一方 SW IV&V は、開発部門から独立した組織が、成果物の安全および品質（記述内容の論理性、一貫性、妥当性等）の検証および妥当性確認を行うための技術および枠組みです。

キーワード

・ IV&V

Independent Verification and Validation（独立検証および妥当性確認）
開発とは独立した機関や組織が検証および妥当性確認を実施する「方法」「体制」「枠組み」。ソフトウェアに焦点をあてた IV&V を SW IV&V と呼ぶ

・ SQA :

Software Quality Assurance（ソフトウェア品質保証）

・ 高信頼性

・ モデル検証

・ 形式記述

厳密であいまいさのない規則を用いた記述言語による仕様記述(JIS X 7001参照)

・ EM/PFM

PFM は Proto Flight Model の略で、実際に飛翔する人工衛星あるいはその搭載機器の動作を作る前に試験的に作る衛星あるいはその搭載機器のことをいう。その前に EM(Engineering Model)を作って基礎的な試験をすることもある

・ MISRA-C

Guidelines for the use of C language in critical systems で、当初は自動車向けに作られたコーディング標準であり、他の高信頼性が必要な領域にも適用が広がっている

SW IV&V 活動は、アメリカの NASA が1993年から実施しています。JAXA では1996年、NASA との宇宙ステーション共同開発で NASA から要請があり、はじめて実施しました。その後、1998年より JAXA 内の他のプロジェクト（例：人工衛星、ロケット等）にも展開がはじまりました。

JAXA における SW IV&V 担当者（実施者）は10名弱（外部支援含む）であり、衛星開発、ロケット開発、宇宙ステーション開発プロジェクトのソフトウェア開発上の成果物を技術的側面から論理性、一貫性、妥当性を検証しており、非常に過酷な業務となっています。ちなみに、ご本家 NASA では150人の SW IV&V 専門家と米国内各宇宙センター等（例：ケネディ宇宙センター、等）に常駐する SW IV&V 技術者が数100名規模で存在しています。

現在 JAXA では、上記実用衛星関連だけではなく、研究プロジェクトの一部にも SW IV&V の適用を試行するところまで来ています。費用・技術面を考えると、業種によってはこれらすべてを実装・展開することは不可能な場合もありますが、それぞれの組織で活用できそうな事項を積極的に利用されることをお勧めします。

■独立行政法人 宇宙航空研究開発機構
(本書で用いる略称：JAXA)

●ビジネスゴール

ビジネスゴールは、国際協力に基づく高信頼性・高安全性システムの提供です。SW 開発を含む全体の目標を、JAXA としての平成18年度の方針として次に示します。

『平成18年度においては、予定されたロケット・衛星の打上げを確実に成功させ、信頼性の確立に一層努めることはもとより、現中期目標期間が残すところあと2年となったことから、宇宙航空に関する科学の発展と技術基盤の拡充に努める等、中期計画の確実な達成に向け業務を遂行するとともに、管理業務改革を促進し一層の効率的経営に努めることとする。ま

た、第3期科学技術基本計画に基づき新たに実施すべき事業の検討や、アジア太平洋地域との連携強化等、次期中期計画に向けた取組みを図る。さらには、国民の理解が一層得られるよう情報発信やコミュニケーションの充実を図ることによって、社会からの信頼の維持・向上を目指す。』
原文は http://www.jaxa.jp/missions/plan/pdf/18th-year_plan.pdf。

●製品・サービスの概要

製品・サービスの領域としては、

- ・ロケット・輸送システム
H-IIA、H-IIIB、HTV*等
- ・人工衛星・探査機
地球観測、通信・測位・技術試験、月・惑星探査、天文観測等
- ・国際宇宙ステーション・有人宇宙開発

国際宇宙ステーション(ISS)、スペースシャトル等

・宇宙科学研究

天文研究、惑星科学、大気・宇宙プラズマ、宇宙情報システム工学等

・宇宙航空基盤技術研究

宇宙機技術研究（将来の宇宙機技術研究、人工衛星技術、…）等

・航空プログラム（Aviation Program）**

将来の航空機技術、安全技術、環境技術等

*H-IIA、H-IIBは、国産のN-1、N-2、H-1、H-IIロケットの後継ロケット。HTVは、「宇宙ステーション補給機(H-II Transfer Vehicle)」。

**航空プログラムとは、宇宙・航空機およびその構成要素に関わる研究開発、試験、施設設備運用および計算・解析を指す。このプログラムはソフトウェア用語ではなく一般用語。

●製品・サービス製品の特徴

常温常圧ではない環境での挙動の保証（動作してはいけない機能と動作しなければいけない機能、故障許容性の明確化、）、自動化・自律化の実現、および組み込みソフトウェアであるという特徴があります。

原文は http://www.jaxa.jp/missions/index_j.html

●4つの主要領域

① 人工衛星

・組み込み系ソフト：姿勢制御（バス系）、

データ処理系、ミッション系***、スタートラッカー(Star Tracker)/GPSR、センサ系

・Fail Safe から Fail Operative

・FDIR: Failure Detection Isolation and Recovery

・書き換え可能（バージョンアップ）

・放射能の影響

・言語：アセンブリ/C

***ミッション系は、衛星に搭載する機器で、衛星の打ち上げ目的である各種実験あるいは運用を行う構成要素。バス系は、衛星に搭載するミッション機器に電源を供給し、地上と交信を行う等の、衛星の基本的な機能を有する構成要素。

② 宇宙ステーション

・管制システム、マニピュレータ、実験装置

・2故障でも安全に！ 高い安全要求

・コマンド、データ、処理の独立性

・複雑で膨大なFDIR (Failure Detection Isolation and Recovery)

・システム構成に人間要素（クルー）を組込む

・言語：C/Ada

③ ロケット

・航行誘導制御系

・高信頼性／多数決

・ハードリアルタイム

・言語：アセンブリ/C

④ 地上管制システム

・管制システム

・安全性

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

1. プロセス改善活動の背景

1.1 改善方針

改善方針は、次のとおりです。

失敗のないソフトウェア開発組織

1.2 達成目標

宇宙関係の開発でも、「コスト低減」「開発期間短縮」は必須です。これを達成するために、以下の具体的活動を実施しています。

① 実施すべき事柄の明確化

暗黙知を形式知にします。

② 個人のノウハウのプロセスへのリンク

失敗事例等、個人に集積しがちな知識を外在化させ、開発プロセスの中に組み込みます。欧州宇宙開発機構(ESA)が失敗事例をデータベースに収集し、細かく更新しています。JAXAとしてこの方式の活用を検討しています。

③ プロセス性能の向上

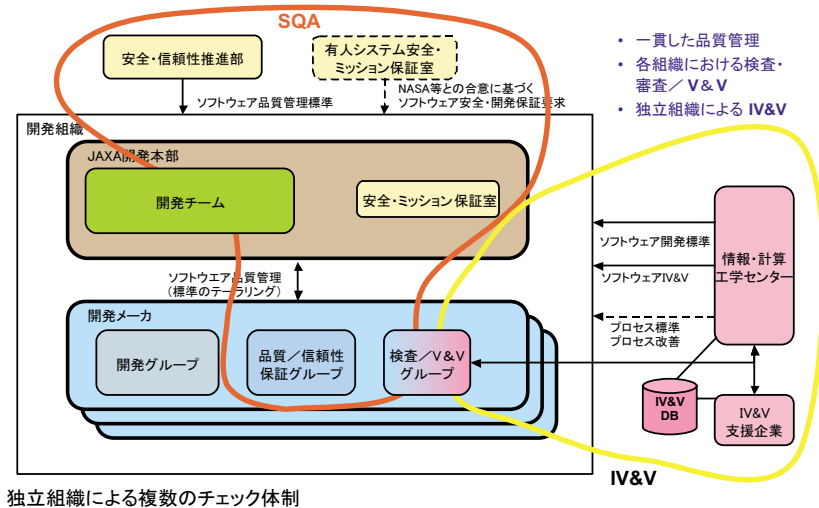
JAXA全体として開発工程能力の向上を進めます。種々の新しい手法・技術・ツールを、定量的な評価に基づき選択し、開発プロセスに組み込み、プロセス性能を向上させていきます。

ほとんどが委託開発であることから、委託先でのプロセス改善を推進する必要があります。

1.3 プロセス改善推進体制、SW IV&V 推進体制

図1に示しますが、推進の中心的組織は情報・計算工学センターです。実際の開発プロセスの中で安全・ミッション保証室、安全・信頼性推進部、有人システム安全・ミッション保証室（宇宙ステーションは特別）と協力して、品質・信頼性の確認等を実施していきます。

宇宙用高信頼性ソフトウェアの開発体制



独立組織による複数のチェック体制

出典：JAXA

図1 JAXAのソフトウェア開発体制

プロセス改善の推進は、3名が担当しています。内訳は、プロセス改善全般担当1名、SW IV&Vを兼ねたプロセス改善担当が2名です。

プロセス改善的体制としては、部門ごと/プロジェクトごとに推進担当者を置くということではなく、上記3名が必要に応じてプロジェクトと直接話し合いを持ち、個々のプロジェクトでの改善活動を推進しています。プロセス改善の対象組織としては、JAXA内部はもちろん、委託先も含まれています。複数の委託先が一同に会した改善検討会のような活動もはじまっています。

SW IV&Vの推進体制も上記プロセス改善推進体制と同じように、部門ごと/プロジェクトごとに推進担当者を置くということではなく、人員はプロセス改善推進と兼務2名に加えて、専任担当者7~8名(外部含む)で実施しています。

1.4 プロセス改善対象、SW IV&V 対象組織

JAXA全体が対象ですが、先行・先端・萌芽的研究は除いています。

1.5 改善活動の経緯

SW IV&V活動は、1996年NASAとの宇宙ステーション共同開発の取り組みに際し、その実施がNASAから義務付けられ、活動を開始しました。それ以降1998年からはロケット、人工衛星の開発へのJAXA内での横展開がはじまり、1998年のH-II A事故を期に、より強固なSW IV&Vの体制が確立しました。また欧州宇宙開発機構で利用されているSPICE for Space (S4S) というISO/IEC 15504に基づいたアセスメントモデルには、SW IV&V関係のプラクティスが含まれているので、JAXAとしてもプロセス改善とSW IV&Vを融合させて推進していく必要があります。

モデルに基づいたプロセス改善への取り組みの歴史は新しく、2002年頃からの実施です。標準類 (STD-24/NDC1-9-1等)、プロセスの課題、CMMによるギャップ分析結果とISO/IEC 12207、SPICE for Space、SA-CMM等を参考にした統合プロセスの検討・実装を行っています。

組織文化的な特徴としては、前身が国の機関であり、また対象ドメインが、簡単に手が届かない宇宙空間という特性からか、「壊れないものを作り上げるためにはなんでもする」「決められた方法をきっちり守る」という精神力があります。ただし、現在ではJAXAの規模は拡大し、分業化、委託先との協業が進みその精神力も次第に薄まりつつあるようです。

1.6 主要な課題と改善の焦点

「より効率的なSW IV&Vプロセスの実施」「SW IV&Vを含めた開発プロセス全体の明確化」に焦点をあてています。つまり開発者のスキル、プロジェクトの難易度を考慮した、もっとも効率的なSW IV&Vのプロセスを決定する枠組みができています。この枠組みにより開発プロセス上で、開発側とSW IV&V側との相互作用点とその深さが明確になります。JAXAではこの枠組みをさらに効率的にするために、詳細化・具体化する部分、一般化する部分、等々の見極めとその実施を行っています。

1.7 SW IV&Vでの顕在化した改善効果

定性的な効果の例としては、以下のような項目があります。

① 技術的欠陥の抽出

- ・通常の試験技術では発見が困難な上流工程の欠陥（開発担当者の思い違い等により混入したシステム挙動の矛盾や重要機能の欠落等）の発見の可能性。とくに、システム欠陥、異常・故障時の動作不成立等を発見。
- ・主開発プロセスにおいて未導入の新規検証技術（モデル検査、フォールトインジェクションによる網羅試験等）により、複雑な欠陥を点検可能。
- ・コードレベルでは、故障耐性不足（エラー処理等）、タイミング競合等の問題が解析により発見可能。

② 開発作業品質の向上

- ・ピアレビューと同様で、「他人が成果物を見る・検証する」ということで、開発担当者へ「ちゃんと作ろう」という気持ちを与える。これは重要な最初の一步。
- ・IV&Vにより発見された問題が引き金となり、開発担当者による自己点検の観点に取り込まれ、他の重要課題の抽出に繋がる。
- ・開発組織、対象製品の特性に合わせて、重要な工程、観点に焦点をあてた検証を行うことができ、信頼性向上のための効率的コスト配分の実現が可能。
- ・新規技術をIV&V技術として実証（シャドウプロセス）を進め、開発プロセスの飛躍的ステップアップに貢献（基礎体力としての信頼性向上、品質向上）。

③ 管理者への品質の可視化

- ・管理者に対して、客観的な立場での製品の評価報告を実施することにより、経営判断への重要な入力。

- ・検証（SUP.2 Verification）
 - ・妥当性確認（SUP.3 Validation）
- ### ② 当該ベストプラクティスに対応するCMMIのプロセス領域
- ・要件開発（RD: Requirements Development）
 - ・技術解（TS: Technical Solution）
 - ・プロセスと成果物の品質保証（PPQA: Process and Product Quality Assurance）
 - ・検証（VER: Verification）
 - ・妥当性確認（VAL: Validation）
- ### ③ 当該ベストプラクティスに対応する共通フレーム2007のプロセス/アクティビティ
- ・開発プロセス/システム要件定義
 - ・検証プロセス
 - ・妥当性確認プロセス

2.2 プラクティスの説明

2.2.1 概要

JAXAで実施しているSW IV&V活動について、開発者・評価者および技術支援者の視点に基づいた説明をします。これまでも述べたようにJAXAでは、大規模で複雑なシステム（ロケット、衛星、宇宙ステーション）を開発しています。費用・人命という視点から、いかに効率よく、信頼性が高く、安全なシステムを作るかに1つの焦点があてられ、そこには国の威信がかかっています。そのため、Independent Verification and Validation (IV&V) という開発組織とはまったく独立した組織が「検証および妥当性確認」を行う仕組みを作り、実践しています。

2.2.2 SW IV&Vとは

SW IV&Vは、米国NASAで1993年より本格的に実施された、ソフトウェアに対して最高の安全性・良好な費用効果を達成するために、開発とは独立した機関や組織が検証および妥当性確認を実施する「方法」「体制」「枠組み」を表しています。また、SW IV&Vはソフトウェア製品の正しさ（妥当性確認）および品質（検証）を、ソフトウェア開発ライフサイクル中で評価するために厳密な方法を用い

2. ベストプラクティス

2.1 参照モデル上でのカバレッジ

① 当該ベストプラクティスに対応するISO/IEC 15504 第5部のプロセス

- ・システム要求事項分析（ENG.2 System requirements analysis）
- ・品質保証（SUP.1 Quality assurance）

るシステム工学プロセスです。そして、ソフトウェアライフサイクル全体を通じてソフトウェア製品の品質・正確さを評価する厳格な手法を利用します。

SW IV&Vの特徴としては、以下の3点が挙げられます(<http://www.ivv.nasa.gov/ivvservices/whativv.php> 参照)。

- ① リスク特定・リスク対応技法を含む
- ② 検証と妥当性確認に対して、独立した評価・査定を提供する（検証組織が開発組織とは独自に検証技術を開発・獲得する）。

・検証

正しく製品を構築しているか？ 検証はソフトウェアの開発サイクルでの各フェーズで確立している（フェーズに対する）要求を満たしているか否かを決定するプロセスです。

・妥当性確認

正しい製品を構築しているか？ 妥当性確認はソフトウェアの開発サイクル全体を通じて、ソフトウェア製品が使命と顧客ニーズを満たしているかを評価することです。

- ③ 開発組織に対して、リスク特定・対応策、品質・安全性の向上、適時性・信頼性の改良、手戻りの低減という事項に対する支援を実現する

また独立性（IV&VのI）について、IEEE 1012 Software Verification and Validation - Description では、技術的独立、管理的独立、財政的独立の3つ視点で規定しています。

① 技術的独立

開発プロセスや製品の査定をするSW IV&V担当者の専門技術は、開発者のそれとは独立です。

② 管理的独立

SW IV&V実施工数はシステム実装に責任を持つ組織から分離されていることが要求されます。SW IV&Vは、解析・テストするソフトウェアおよびシステムのセグメント（部分）を独立して選択し、SW IV&V手法を選定し、IV&V実施スケジュールを立案し、取り組むべき特定の技術課題・問題を選択します。

③ 財政的独立

・シャドウプロセスとしての役割

【目的】

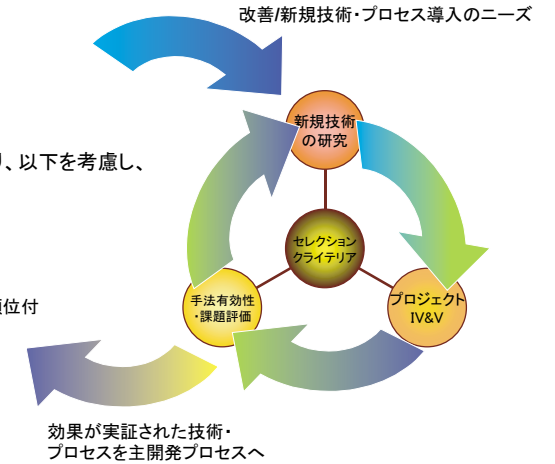
- ・新規技術・プロセスの実証
- ・不具合・問題点の発見

【実施内容選定】

- 各プロジェクトのIV&V実施にあたり、以下を考慮し、実施内容を選定・決定する
- ・不具合定量化による重点箇所
 - ・プロジェクト内作業員インタビュー
 - ・ソフトウェアの特徴・特性
 - ・データアクセス性
 - ・許容工数とスケジュールによる優先順位付
 - ・作業開始時期

【手法の有効性評価】

- 有効性を以下の基準で評価し、次プロジェクトへ反映する
- ・問題点発見総件数
 - ・重要問題点抽出比率
 - ・作業工数
 - ・手法利用時の課題
 - ・成果に対する顧客の評価



出典：JAXA

図2 JAXA SW IV&V 独立性

SW IV&Vは、開発組織より別で上位（通常は本社組織等）の組織から資金提供（予算立案）されるべきものです。通常の開発プロジェクトはSW IV&Vサービスを受けたことに対して、費用を支払います。

JAXAでは、上記②、③の独立性が維持される組織体制になっています。また、上記①「技術の独立性」に則って、図2のようなSW IV&Vプロセスを構築し、新規検証技術・手法の研究、プロジェクトへのSW IV&Vサービスの提供、検証手法の評価のサイクルを回しています。

2.2.3 SW IV&V への取り組み理由

ソフトウェアの高信頼化を開発プロセス上で実現するために、1996年にNASAとの宇宙ステーション共同開発で、NASAからの要請があり、JAXAで初めてIV&Vが取り入れられました。それ以降1998年からはロケット、人工衛星の開発にも適用しています。

2.2.4 SW IV&Vの推進方法

前述の様に、最初は1996年 NASA との宇宙ステーション共同開発で NASA から要請があり、SW IV&V 担当者2名（宇宙ステーション専用）で開始しました。NASA の SW IV&V の作業内容を参考にして、JAXA での実施内容を検討しましたが、「設計レビュー」「コード検査」という一般的なことが多く、それを第三者が実施することに“どのような特別な意味”があるのかという疑問を当時感じました。また、NASA の SW IV&V は開発プロセス全体と整合された SW IV&V プロセスは構築されておらず、個別検証作業項目の羅列のような状態で、そのような状況の中、NASA との協力、学会等の調査（例：ICSE 京都、等）を経て、2000年には現在のような検証メニューの体系化が完了しています。

2000年当時は、既に対象は宇宙ステーションだけではなく、ロケット・人工衛星にも広がっていましたが、専任担当者は相変わらず2名のままでした。それ以降、新しい技術を選択的に取り入れながら、継続的に検証メニューの改善を実施し、現在の体制は10名前後（外部支援含む）となっています。

JAXA のプロジェクトへの SW IV&V 推進方法としては、「プロジェクトの理解を得る」「プロジェクトの方々に SW IV&V の利点を感じていただく」ということを基本としており、6割程度のプロジェクトに対して SW IV&V サービスを提供しています。ただし、担当者の不足ですべての要望に応えることができていないわけではありません。

2.2.5 SW IV&V 高品質ソフトウェアのための組織体制

JAXA でのソフトウェア開発は企業への開発委託によって実施しています。委託会社は、人工衛星2社、宇宙ステーション7社、ロケット2社が代表的な主契約会社です。主契約会社～2次契約会社の関係は平均的には1～5ぐらいとなっています。委託内容は機能システム全体の開発委託の形態であり、ソフトウェア単体での開発委託の割合は非常に低くなります。

JAXA 内部には開発チーム（プロジェクト推進部門）とは独立した SQA/SW IV&V があり、各主委託会社内でも同様の体制確立（SQA/V&V）が原則必須になっています。JAXA 自身は主委託会社の体制確認は実施しますが、2次契約会社での

◆そもそも、NASA はなぜ IV&V をはじめた？◆

① 1996年以前、NASA は V&V 体制、つまり開発組織で Verification/Validation 作業実施体制を確立しており、それなりの効果も上げていた。しかし、1987年のスペースシャトル（チャレンジャー号）の爆破事故が発生し、原因究明が強力な体制で実施された。最終的な真の原因は固体ロケットブースター（SRB）のオーリングの不具合^{*}で、ソフトウェアの不具合ではありませんでした。

しかし、原因究明の過程でソフトウェア成果物についても専門家による再チェックが実施され、膨大な数の不具合（候補）が発見され、それらの中にはシャトル運用に甚大な被害をもたらす可能性のあるものも多く含まれていました。ロジャー委員会のソフトウェア部会報告書にあるように、これらソフトウェア不具合（候補）の中には「第三者がチェックすることで事前に取り除ける不具合」もたくさんあり、このことから、NASA では IV&V という検証プロセス確立に向けた作業が進められました。

^{*}シャトルの SRB はいくつかのセグメントをつないで作られており、そのつなぎ目にゴム製の Oリングと呼ばれるもので密着させています。そのゴムが当時の寒波による異常低温で弾性を失い内部の圧力を遮蔽できなくなりそこから炎が漏れ、その炎によって外部燃料タンクが爆発したとことです。http://history.nasa.gov/rogersrep/51lcover.htm に詳細な報告があります。また、http://sunnyday.mit.edu/accidents/index.html に NASA の宇宙事故等に関する情報があります。

② 「人の死に至ったあの飛行の安全性を損なう潜在的な危険を指摘していたエンジニアが何人もいたにもかかわらず、スケジュール通りに打ち上げを行うというプレッシャーのかかった管理者は、彼らの訴えを却下していたのです。」というお話もあるようです（ピート・マクブリー：ソフトウェア職人気質、P.39、ピアソン・エデュケーション、2002年より）。

V&Vの実装状況は、主委託会社が管理していることを前提に、直接JAXAが関与することはありません。JAXAのSQA/SW IV&Vと委託先のSQA/V&Vがそれぞれ協力をして、検証・妥当性確認作業を実施しています。

SQAは開発部門から独立した組織が、所定の開発プロセスに則って成果物が作成されたことを保証する活動です(=検証)。一方SW IV&Vは、開発部門から独立した組織が、成果物の安全および品質(記述内容の論理性、一貫性、妥当性等)を検証する活動です。次節で述べる検証実施項目の詳細が規定されています。

JAXAにおけるSW IV&V担当者(実施者)は、衛星開発、ロケット開発、宇宙ステーション開発プロジェクトのSW開発上の成果物を技術的側面から論理性、一貫性、妥当性を検証しています。

図1で示している情報・計算工学センターの役割は、

- ・ 民間IT技術の宇宙開発への適用
- ・ SW IV&V技術
- ・ SW工学の新技術獲得
- ・ プロセス改善
- ・ SW基盤技術獲得
- ・ 標準類の整備

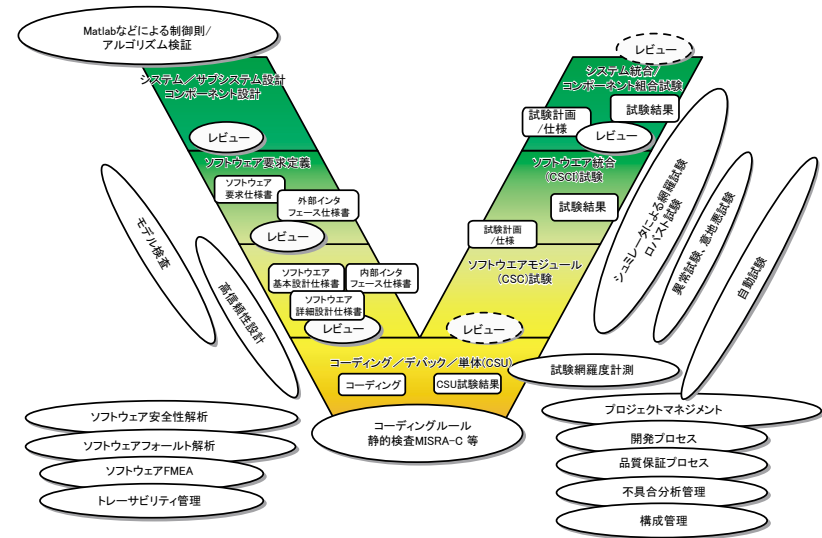
となっており、SPI(ソフトウェアプロセス改善)推進、SW IV&V推進の他にSW工学分野の技術開発・技術獲得も行っています。

2.2.6 SW IV&V 開発工程上の実施事項

まず、開発プロセス上での実施事項の概要を図3に示します。

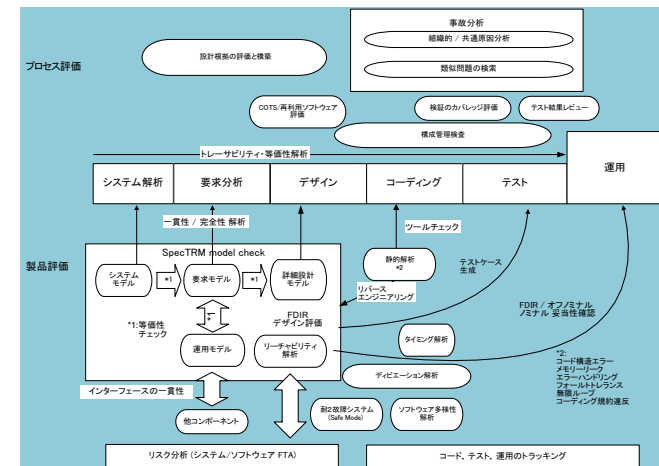
SW IV&Vでは開発プロセス上の成果物(例：設計書、コード、等々)に対して、プロジェクトの特徴に応じて、そのやり方(厳格さ、詳細さ等)を調整しながら、検証・妥当性確認を実施します(図4)。JAXAではEM/PFMという実機モデルでの検証をしていますが、そのモデルを使ってソフトウェア関連機能の検証できる部分はハードウェアとの整合性等、非常に限定されています。このような中、高品質・高安全性ソフトウェアを実現するため、仕様自体の一貫性や完全性の検証、また仕様と各工程での成果物の検証を着実に実施するための体制および開発プロセスが存在しています。その中心がSW IV&Vなのです。

次にSW IV&V実施項目を決定する具体的な方法を示します。開発フェーズごとに実施すべき項目、実施が好ましい項目を大まかに決定します。プロジェクトの



出典：JAXA

図3 開発工程とSW IV&V



出典：JAXA

図4 開発プロセスとSW IV&V

特徴によって実際に実施すべき解析項目とその実施レベル（手順）を調整します。手法選択とそのレベル設定の概要を図5に示します。例えば、完全性・一貫性解析という手法を例にとると、軽レベルではチェックリストの活用となりますが、重レベルではモデル化・モデル検査からシミュレーションの活用となります。

手順レベルで調整の例として、「コーディングに対するSW IV&V」では、以下のようなレベル分けによって手順が変わってくるようになります。

① レベルー軽 (Light)

開発側で、標準解析ツールを使用してコードチェックして、その結果をSW IV &V 担当者に報告。当該担当者はその報告をチェックし、不足・疑問等を指摘します。

② レベルー重 (Full set)

コードをSW IV&V 担当者に提供。当該担当者は、複数のツール・複数の視点からコードチェックを実施し、その結果を分析し、修正箇所・疑問等を指摘します。なお、レベルは非常に細かく分けており、多いものになると数十にもなります。

2.2.7 SW IV&V の費用

SW IV&V への取り組みにかかった費用は、非常に興味のあるところですが、個々の額は公表されていません。ただし、NASA からガイドラインが公表されており、ソフトウェア開発費の10%がSW IV&V の標準的な費用とされています。

JAXA での状況は、そのガイドラインに対して格段に少なく、また、新しい検証技術獲得の経費は研究開発予算を確保しており、個々のプロジェクトからの提供は行っていません。

2.3 活用しているツール・技法

JAXA でSW IV&V のために使用しているツールには、

- ・各種シミュレータ (Matlab、…)
- ・テストケース生成
- ・静的コード解析 (MISRA-C) 等
- ・モデル作成/モデル検証ツール等

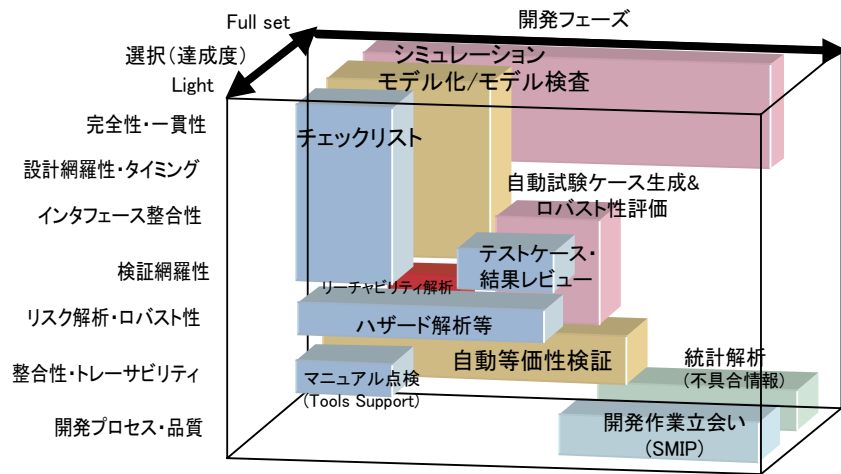
等があり、JAXA 内部で様々な改良を行っています。

手法・技法としては、主要なものとして、表1に記載したように23種類ほどあります。

また開発プロセスを規定する標準書としては、

- ・搭載ソフトウェア品質保証プログラム標準 (JMR-009A), 平成16年10月1日, A改訂
- ・プロセス定義文書目次 (NDC-1-9-1)
- ・ソフトウェア設計基準 (JERG-0-026), 平成16年4月1日, (NDC-1-9-1)
- ・標準プロセス (案)

等があり、さらなる改良・拡充作業を予定しています。



出典：JAXA

図5 検証手法とその達成度選択

表1 検証手法と工程

(凡例 ○：フェーズで活用が推奨される手法 △：開発進捗状況により活用が推奨される手法)

解析評価手法名	工程	要求分析	基本設計	詳細設計	コーディング	結合試験	実地検証	総合試験	運用
1 モード遷移解析		○	○	○					○
2 完全性解析		○	○	○					○
3 ノミナルシミュレーション						○	○		○
4 FDIR (1故障時) 網羅性解析		○	○	○					○
5 ソフトウェア逸脱解析			○	○	○				○
6 単一故障シミュレーション					○	○	○		○
7 FDIR 階層分析		○	○	○					○
8 複数 FDIR 並列実行解析		○	○	○	○	○	○		○
9 安全モード遷移解析(FMEA, FTA)		○	○	○	○	○	○		○
10 二重故障シミュレーション						○	○		○
11 HAZOP (Hazard and Operability) 解析						○	○		○
12 一貫性解析		○	○	○					
13 共有変数解析					○				○
14 コードクローン解析					○				○
15 階層モデルによる事故事例解析		△	△	△	△	△	△		△
16 自然言語仕様によるモデルの自動生成		○	○	○	○	○	○		○
17 トレーサビリティ解析		○	○	○	○				○
18 リーチャビリティ解析		○	○	○					○
19 運用制約識別		○	○	○	○	○	○		○
20 インタフェイス解析		○	○	○					○
21 タイミング解析			○	○					○
22 二重故障安全網羅テストケース解析				△	○	○	○		○
23 姿勢制御系搭載ソフトウェアチェックリスト		○	○	○	○				○

2.4 人材育成

SW IV&V 担当者を育成する組織的な活動はこれまでなく、要望に応じて現場の技術者に SW IV&V トレーニング (OJT) を実施等、個々の状況の中で OJT をしていました。今後は、組織的に SW IV&V 専門育成ができるようなシステムにしていくことが予定されつつあります。

具体的には、下記の取り組みです。

- ・開発者のスキル標準の制定
- ・SW IV&V スキル要件の検討 (NASA、欧州との共通化を目指す)
- ・教育プログラムの共有化 (NASA、欧州、大学)
- ・大学・研究機関との協調 (NASA では、米国の20大学と協調関係を構築。JAXA では、奈良先端科学技術大学院大学、大阪大学等との協調を進めています)

3. 学ばべき点

SW IV&V を適用するか、しないかを判断するための定量的・一般的基準はないようです。JAXA では、「製品がユーザにリスクを与える」ものを開発している場合には SW IV&V は効果があると考えています。例えば医療機器関係、運輸関係 (車・列車・飛行機)、社会インフラ関係 (金融、電気、ガス、水、通信…) のシステム開発では、この事例が参考になると考えます。

参考文献

ピート・マクブリー：ソフトウェア職人気質，P.39，ピアソン・エデュケーション，2002年

BEST
PRACTICE
事例4

生産性原理を具現化するプロセス改善

～生産管理システム「ACTUM」の活用～

株式会社ジャステック

ジャステックは、ソフトウェア開発の生産管理システム「ACTUM」を構築し、プロセス改善を前提とした生産性原理の具現化（生産性向上度合を評価し社員の給与へ反映すること、および生産性向上度合を評価し価格へ反映すること）を図り、見積りからプロジェクト管理まで一貫した指標（開発量、生産性および環境変数）を持ち、プロセス改善の「計る化（定量化）」を目指しています。

ジャステックのプロセス改善事例は、開発量（生産物量）および生産性に関わる開発環境の違いや品質要求の多寡による変動を吸収する「環境変数」を使用して、リスク低減およびコスト低減を可能としています。

環境変数を使用して、ジャステック独自の考え方に基づいた予実差異分析を行うことで、開発局面ごとのプロセス改善機会の創出を促しています。また、環境変数の差異を分析して改善策を導き出すとともに、改善策を講じた結果どの程度の改善効果があったかを改善

キーワード

- ・ 開発計画
- ・ 定量的管理
- ・ 見える化
- ・ 計る化
- ・ 生産物量環境変数
基準生産物量に対して品質要求の多寡による変動を吸収する変数
- ・ 生産性環境変数
基準生産性に対して開発環境の違いや品質要求の多寡による変動を吸収する変数
- ・ ACTUM
ジャステックで開発されたソフトウェア開発の生産管理システム

率として算出しています。

組織プロセス改善として開発部門の各階層（部門、課、チーム）ごとに環境変数を統計処理し変動幅の是正や安定性の確保に取り組んでいます。

ジャステックでは、創業以来、個人および集団の定量的なパフォーマンス評価（生産性、品質）に基づく能力主義を標榜し、その精錬に挑戦しています。技術者の生産性については、顧客の特質やプロジェクトの特性に左右されないように、環境変数を使用して評価値の補正を行い、生産管理システム「ACTUM」と評価システムとを連動させて、相互の運用精度を高めています。

また、顧客に提示する見積価格に関しては、ジャステックで設定した外部環境変数を顧客に公開し、調整を行うことで、リスク低減や価格低減の機会を創出しています。

さらに、組織マネジメントの観点からは、開発部門を中心に全社的なプロセス改善の支援体制を整え、経営者によるマネジメントレビューで承認された組織プロセス改善目標を予算化（年度ごとに全部門合計で約130の改善に関わる指標を持つ）し、達成度合を監視しています。

■ 株式会社ジャステック

● 経営理念

- ① 生産性原理の導入を図る。つまり生産性向上度合を評価し社員の給与へ反映すること、および生産性向上度合を評価し価格へ反映することである。
- ② 独立系のソフトウェアハウスとしてソフトウェア開発・販売専門を堅持する。とくに、開発関連業務の多角化ではなく、開発分野の総合化と流通化を図る。
- ③ 知識集約産業であるシステムインテグレータとしてのあるべき姿を、物真似せず独自に追及し情報社会に貢献する。

● 製品・サービス概要

- ・ ソフトウェア開発
- ・ 業務系・情報系ビジネスオートメーション、プロセスオートメーション、基本ソフト等。
- ・ システム販売

ソフトウェア生産管理システム、画像・映像類似検索パッケージ等。

- ・ 仕入商品販売
 - ・ 顧客との共同開発パッケージ（SAINTS、DBIR、MUSE等）、販売提携パッケージ、導入コンサルテーション、カスタマイズ、アドオン開発等。
 - ・ その他
- これまでに蓄積した開発技術および開発管理技術のシステム構築への適用コンサルテーション、情報システムの企画提案および要件定義に関するコンサルテーション、ならびに米国子会社を中心とした世界に通用するパッケージ（画像・映像類似検索パッケージ：仏「LTU Technologies」社の全株式取得済）の開発と販売。

URL <http://www.jastec.co.jp/>

1. プロセス改善活動の背景

1.1 改善方針

ジャステックでは、見積りからプロジェクト管理まで一貫した指標（開発量、生産性および環境変数）を持ち、プロセス改善の「計る化」を目指します。

1.2 達成目標

本書で紹介するプロセス改善は、開発量（生産物量）および生産性に関わる開発環境の違いや品質要求の多寡による変動を吸収する「環境変数」を使用したり、リスクおよびコストの低減を目的としています。

1.3 プロセス改善推進体制

ソフトウェア開発に関わるプロセス改善は、製造本部長のプロセス改善方針に基づき製造部長および製造課長が中心的な役割を担います。また、製造本部管轄の製造統括部および製造本部管轄外の品質／環境委員会、検査課、技術課、教育課、監査委員会は、プロセス改善推進を支援する役割として位置づけています（図1）。各組織の役割に関しては「2.2.7 支援環境と支援体制」で紹介しています。

1.4 改善活動の経緯

ジャステックでは、創業以来、労働集約型ではなく知識集約型のソフトウェア開発専門企業を標榜してきました。ソフトウェア開発における生産管理の取り組みは、労働集約型の派遣契約ではソフトウェア開発の生産性向上の努力を社員も経営者もなくなる、といった危機感からジャステック独自のソフトウェア開発における生産管理への取り組みがはじまりました。

1983年に労働者派遣法成立の機運が高まる一方で、脱マンパワーリースを率先垂範すべく、創業以来培ってきたプロジェクトマネジメント技術を集大成し、ソフトウェア開発の生産管理システム「ACTUM」を構築しました。

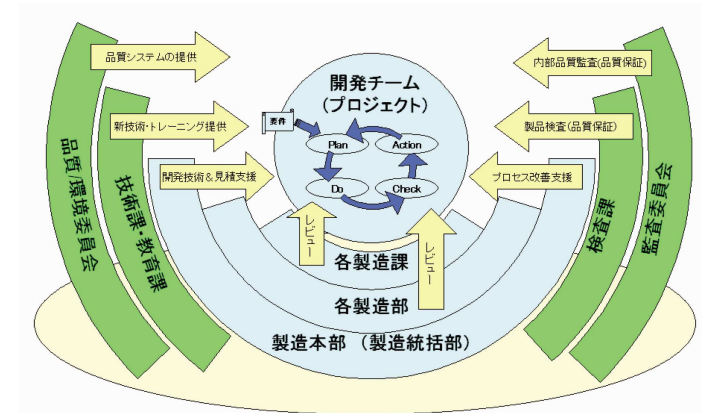


図1 製造本部を中心としたプロセス改善推進体制

生産管理システムの導入はプロセス改善を前提とした生産性原理の具現化を促しました。そして、見積りと連携した開発計画および出来高管理の定量化は、「見える化」にとどまらず「計る化」を標榜したプロセス改善の第一歩でした。

さらに、1996年には独自の生産管理システムを基軸に、品質保証の国際標準であるISO 9001を取り入れ組織レベルでの規程を整備し、運用を徹底しました。2002年1月にSEI（ソフトウェアエンジニアリング研究所）から、ソフトウェアプロセス改善の国際標準的なモデルであるCMMI（能力成熟度モデル統合）「V1.1」が、公開評価版として発表されました。CMMI レベル5は、ジャステックの生産管理システムのコンセプトと符合することを知り、CMMI レベル5（2003年「CMMI-SE/SW/IPPD/SS Version 1.1 連続表現」にて達成）を融合した独創的なプロセス改善体系を確立し、運用の徹底を図り、今日に至っています。

1.5 主要な課題と改善のフォーカス

ジャステックではCMMI レベル5（V1.1）の達成に引き続き、CMMI レベル5（V1.2）の達成を目指し、全部門（営業部門含め）において約130の改善指標の導入を図る等、全社的な取り組みを実践しています。

とくに、開発部門（製造本部）では、品質系およびコスト系へのプロジェクト

定量分析・測定・監視にフォーカスし、新たな定量的な生産管理を基軸としたプロセス改善の挑戦がはじまっています。

そのプロジェクト定量分析運用フローの概要を図2に示します。

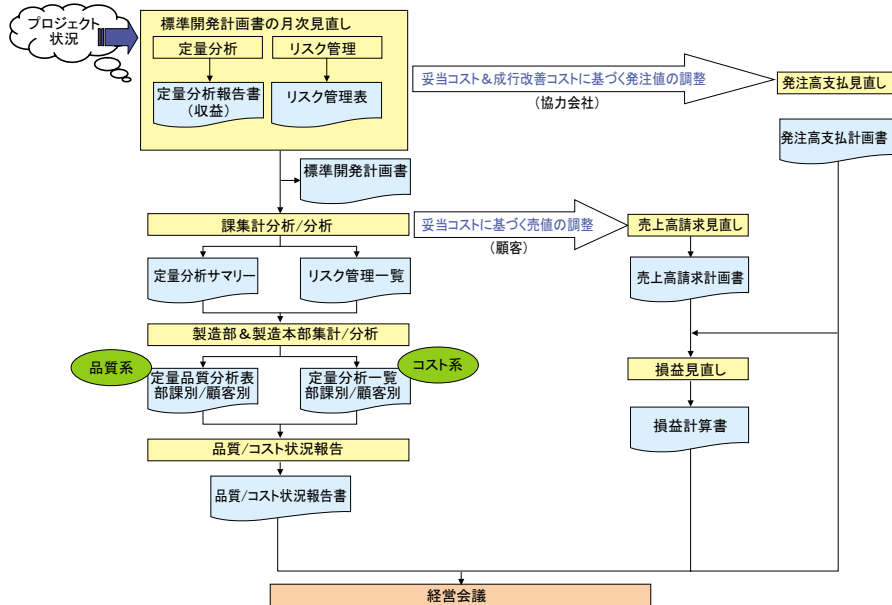


図2 プロジェクト定量分析運用フロー

1.6 顕在化した改善効果

ジャステック独自のソフトウェア開発の生産管理を基軸にしたプロセス改善の取り組みは、結果として業界屈指の高収益の達成を可能とした（過去3カ年平均の単独売上高経常利益率17.5%）。また、プロセス改善等の取り組み、および健全経営の評価を得て、平成16年度情報化促進貢献企業として経済産業大臣賞を受賞しています。

2. ベストプラクティス

2.1 参照モデル上でのカバレッジ

- ① 当該ベストプラクティスに対する ISO/IEC 15504 第5部のプロセス
 - ・プロジェクト管理 (MAN.3 Project management)
 - ・リスク管理 (MAN.5 Risk management)
 - ・測定 (MAN.6 Measurement)
- ② 当該ベストプラクティスに対応する CMMI のプロセス領域
 - ・プロジェクト計画策定 (PP: Project Planning)
 - ・プロジェクトの監視と制御 (PMC: Project Monitoring and Control)
 - ・測定と分析 (MA: Measurement and Analysis)
 - ・統合プロジェクト管理 (IPM: Integrated Project Management)
 - ・リスク管理 (RSKM: Risk Management)
 - ・組織プロセス実績 (OPP: Organizational Process Performance)
- ③ 当該ベストプラクティスに対応する共通フレーム2007のプロセス/アクティビティ
 - ・開発プロセス/システム要件定義
 - ・保守プロセス
 - ・管理プロセス

2.2 プラクティスの説明

2.2.1 プロセス改善に関わる指標

ジャステックでは、年度ごとに下記 (①～⑧) のプロセス改善への契機から発生するプロセス改善実績を収集・分析し、経営者によるマネジメントレビューを通して各部門の改善項目および指標値を設定しています。

- ① 経営者によるマネジメントレビュー
- ② ISO 9001定期検査
- ③ CMMI レベル5の外部および内部アプレイザル
- ④ 内部品質監査
- ⑤ 製品検査

- ⑥ 予算提案および業務改善提案
- ⑦ 生産性向上事例発表会
- ⑧ 見積りおよびプロジェクト管理

毎年、全部門合計で約130（そのうち製造本部は約100）のプロセス改善に関わる指標が予算目標として上がっています。例えば、開発部門（製造本部）では、納品検査合格率、納期達成率、出荷後のバグ密度および改善成功率等があります。

本書で紹介するプロセス改善の指標は、プロジェクトのコストに関わる要素として、開発量（生産物量）および生産性に関する指標、ならびに、それらの要素において開発環境の違いや品質要求の多寡による変動を吸収する指標（環境変数）にフォーカスしています。その基本アルゴリズムを「2.2.2 生産管理方式に基づく見積りモデル」に示します。

2.2.2 生産管理方式に基づく見積りモデル

ジャステックの生産管理方式に基づく見積りモデルを紹介します。加えて新規開発におけるコスト見積りモデルの基本アルゴリズムを示します。

ある開発工程 i の基準生産物量を V^B_i 、基準生産性を P^B_i （以下「基準値」）で表現すると、生産物量 V_i 、生産性 P_i および開発コスト C_i は次式で求めることができます。

$$V_i = V^B_i \times (1 + a_i) \quad (a_i = \sum \alpha_{ij})$$

$$P_i = P^B_i \times (1 + b_i) \quad (b_i = \sum \beta_{ij})$$

$$C_i = V_i \times P_i$$

a_i は生産物量環境変数と呼ぶパラメータであり、 V^B_i に対して品質要求の多寡による変動を吸収する変数です。また b_i は、生産性環境変数と呼ぶパラメータであり、 P^B_i に対して開発環境の違いや品質要求の多寡による変動を吸収する変数です。 a_i および b_i は品質特性と環境特性から影響される独立した変動要素 (α_{ij} , β_{ij}) から構成されています。生産物量環境変数 a_i は表1に、生産性環境変数 b_i は表3、表4に示します。さらに、環境変数には改造型開発特有の環境変数があります。改造型

開発特有の生産物量環境変数 a'_i は表2に、生産性環境変数 b'_i は表5に示します。

なお、環境変数には、外部環境変数と内部環境変数（外部環境変数と同じ特性項目で構成されるが変動要因および変動率は異なる）があります。外部環境変数には「顧客の特性による変動」および「プロジェクトの特性による変動」があり、価格に反映されます。内部環境変数は「開発ベンダの能力による変動」であり、ソフトウェアの価格に反映されませんが、プロジェクトの実行計画には反映されます。

表1 品質特性変動要素評価表（生産物量に影響する外部環境変数）($\alpha_i = \sum \alpha_{ij}$)

主特性	副特性	変動要素	基準値からの変動率(%)			
			要件定義	設計	製作	テスト
機能性	合目的性	利用者／利害関係者の広がり、コンテンツエンシー対応、不正移行データ対応等の該当事象数	0～50	0～50	0～50	0～50
	正確性	正確性（検証）に関する標準テスト密度を基準にしたテスト項目量への要求水準	—	—	0～20	0～50
	接続性	他システムとの接続によるコード変換、フォーマット変換数	0～5	0～20	0～20	0～20
	セキュリティ	対応が必要なセキュリティ実現機能数、ただし機能要件に定義されている部分は除く	0～20	0～20	0～20	0～20
信頼性	成熟性	故障低減に必要な実現機能数	0～5	0～10	0～10	0～10
	障害許容性	異常検知に必要な機能数	0～5	0～10	0～10	0～10
	回復性	再開処理に必要な実現機能数	0～5	0～10	0～10	0～10
使用性	理解性	理解性向上（機能等）のためのプレゼンツール等の作成対象数	—	0～10	—	—
	習得性	習得性向上（使い方等）のためのマニュアル等作成対象数	—	0～10	—	—
	操作性	操作性向上（心理的／肉体的配慮、運用やインストール容易性等）のための実現機能数	0～10	0～20	0～20	0～20
保守性	解析性	解析に必要な実現機能数	—	0～10	0～10	0～10
	変更作業性	作成する保守用ドキュメントの数	—	0～10	—	—
	試験性	試験に必要な機能数	—	0～5	0～15	0～20

表2 現行資産特性変動要素評価表（生産物量に影響する外部環境変数） $(a_i = \Sigma \alpha'_{ij})$
 (改造型開発特有の環境変数)

主特性	副特性	変動要素	ベースラインからの変動率(%)			
			要件定義	設計	製作	テスト
機能性	正確性（既存母体）	改造／流用母体が正しく動作しない場合のテスト量（現行保証）に及ぼす影響	-	0～5	0～15	0～20

表3 環境特性変動要素評価表（生産性に影響する外部環境変数） $(b_i = \Sigma \beta_{ij})$

主特性	副特性	変動要素	基準値からの変動率(%)			
			要件定義	設計	製作	テスト
業務特性	業務ナレッジ	顧客の開発対象業務に対する業務ナレッジが生産性に及ぼす影響	-10 ～50	-10 ～10	-	-10 ～10
ハードウェア特性	安定度／信頼度／使用度	システムもしくは製品となるハードウェアの安定度・信頼度	-	-5 ～5	-	-5 ～5
ソフトウェア特性	安定度／信頼度／使用度	システム／製品に組み込む他社作成ソフトウェアもしくはCOTSの安定度・信頼度	-	-5 ～5	-	-5 ～5
コミュニケーション特性	顧客窓口特性	意思決定能力（期限遵守、決定事項の覆る度合）	-10 ～20	-10 ～10	-	-7 ～7
	工期の厳しさ	基準工期（月）=2.7×（人月）1／3に対し▲30%限度とした短期化度合	0～10	0～10	0～5	0～7
	コミュニケーション基盤	開発拠点分散、資料等情報共有、電子媒体・システム具備等物理的基盤充実度	-10 ～10	-5 ～5	-3 ～3	-3 ～3
開発環境特性	レビュー体制	無駄なレビュー（重複多段階等）の排除およびレビュー効率向上への工夫度合	-5 ～5	-5 ～5	-3 ～5	-5 ～5
	開発手法／開発環境	開発手法・環境（ソフト／ハード／ツール）の信頼性、占有率等を考慮した使用実績	-3 ～3	-3 ～3	-5 ～5	-5 ～5
	テスト手順書水準	テスト手順の具体化度（操作手順&入出力の具体化の要求水準）	-	-	-3 ～3	-3 ～3

(つづく)

(前頁より)

工程入力情報特性	業務関連資料	必要資料の具備状況（正確性、信頼性を含む）および使いやすさ（検索性、理解性）	-10 ～10	-7 ～7	-3 ～3	-3 ～3
	他システム関連資料	必要資料の具備状況（正確性、信頼性を含む）および使いやすさ（検索性、理解性）	-10 ～10			
	規約・標準化関連資料	必要資料の具備状況（正確性、信頼性を含む）および使いやすさ（検索性、理解性）	-7 ～7			

表4 品質特性変動要素評価表（生産性に影響する外部環境変数） $(b_i = \Sigma \beta_{ij})$

主特性	副特性	変動要素	基準値からの変動率(%)			
			要件定義	設計	製作	テスト
機能性	合目的性（要求仕様の網羅性）	要求の記述水準および網羅性。要件定義については新規性、方針明確性、ステークホルダの多様性等を考慮	0～100	0～30	-	0～10
	正確性	正確性（検証）に関わる標準レビュー工数（各工程8%）を基準にした要求水準	0～5	0～5	0～3	0～5
	接続性	基準単位（100ks）に対する社内／社外システムとのインタフェース先の数	-10 ～10	-5 ～10	-	-5 ～5
効率性	整合性	整合をとる社内／社外の規格・基準の数、全体適合性やグローバル化対応も含む	-10 ～10	-5 ～10	-3 ～5	-3 ～5
	実行効率性	実行効率に対する一般的要求水準（既知）の標準事例を基準にした要求水準	0～5	0～10	0～5	0～10
	資源効率性	資源効率に対する一般的要求水準（既知）の標準事例を基準にした要求水準	0～5	0～10	0～5	0～10
保守性	解析性	ソースコードの解析性をコード化規約に定めるコメントに対する要求水準により評価	-	-	0～5	-

(つづく)

(前頁より)

	安定性	ソフトウェア変更に対しシステム品質維持可能とする水準をライフサイクル目標年数の長さにより評価	0~5	-3 ~7	-3 ~5	-
移植性	環境適用性 移植作業性 規格準拠性 置換性	ソフトウェアをどの程度、多様な環境に移すことができるかに対する要求の水準	0~28	0~28	0~12	0~25

表5 現行資産特性変動要素評価表 (生産性に影響する外部環境変数) ($b'_i = \sum \beta'_{ij}$)
(改造型開発特有の環境変数)

主特性	副特性	変動要素	ベースラインからの変動率(%)			
			要件定義	設計	製作	テスト
改造・再構築特性	母体調査ツールの機能水準	調査ツールの機能の数 ・ 絞込み ・ モニタリング (ルート解析) ・ ドキュメント生成 (リバース) ・ 実機での稼働確認 ・ データディクショナリ ・ その他調査ツール	-	-20 ~15	-10 ~5	-20 ~10
	既存設計書具備状態	改造/流用母体の設計書有無および設計書が存在した場合のメンテナンス状態	0~10	0~30	0~30	0~25
	既存のテスト環境流用水準	既存のドキュメントおよびテストデータ (テスト環境を含む) の流用可能度合	-	-	-20 ~0	-30 ~0
	リソース管理水準	ソースとロードモジュールのバージョン管理水準	-	-	0~15	0~15
	既存母体品質 (正確性)	既存システムが正しく動作しない場合の生産性に及ぼす影響度	0~8	0~10	0~10	0~10
	既存母体品質 (解析性)	既存システムの改造箇所特定 (改造設計) における標準化項目の遵守度合によるソースコードの解析容易性	0~10	0~25	0~25	0~15

(つづく)

(前頁より)

		・ ソースコメント ・ 修正履歴 ・ モジュールサイズ ・ 規約 ・ その他顧客規約事項				
	既存母体品質 (環境適用性)	現行システム資産を別環境へ移植し改造する開発における、別環境への適用容易性 (適用可能な環境の個数) ・ ハードウェア ・ OS ・ ネットワーク ・ フレームワーク (ミドル) ・ DB/DC ・ バッチ運用システム	0~10	0~25	0~25	0~15
改造・再構築特性	既存母体品質 (環境適用性)	現行システム資産を別環境へ移植し改造する開発における、別環境への適用容易性 (適用可能な環境の個数) ・ ハードウェア ・ OS ・ ネットワーク ・ フレームワーク (ミドル) ・ DB/DC ・ バッチ運用システム	0~10	0~25	0~25	0~15

2.2.3 見積りからプロジェクト管理まで指標を統一

見積りは単独の活動として扱うものではなく、プロジェクト管理と一体として扱うことにより開発計画の精度を高め、開発計画達成へのコントロールを可能にし、さらには開発プロセス改善の機会を創出します。

ジャステックには、契約への入力となる標準見積書以外に、プロジェクト開始時に作成する標準開発計画書および開発工程着手時に作成するプロジェクト実行計画書が存在します。プロジェクト実行計画書は実行体制の確立が前提になります。開発進行中にはプロジェクトの監視と制御に関する各種分析書および報告書があり、代表的なものとしてプロジェクト実績報告書があります。さらに、完了

時点ではプロジェクト完了報告書が存在します。

プロジェクト実績報告書は、図3のプロジェクト実行計画書と同一項目となっており、予実対比を可能にしています。プロジェクト完了報告書はプロジェクト実績報告書に基づき、次プロジェクトまたは他プロジェクトへのプロセス改善に結び付いた是正処置を明記しています。

とくに、見積りからプロジェクトの計画、出来高管理および実績評価まで、コストに関わる要素「開発量、生産性および環境変数」を、同一指標で取り扱うことにより、見積りとプロジェクト管理の一体化が可能となります。

図3に標準見積書、標準開発計画書およびプロジェクト実行計画書における指標の連携を示します。

2.2.4 環境変数を利用したプロセス改善

(1) 見積りとプロジェクト計画段階

見積りとプロジェクト計画段階では、顧客の見積要求仕様書に基づき標準開発

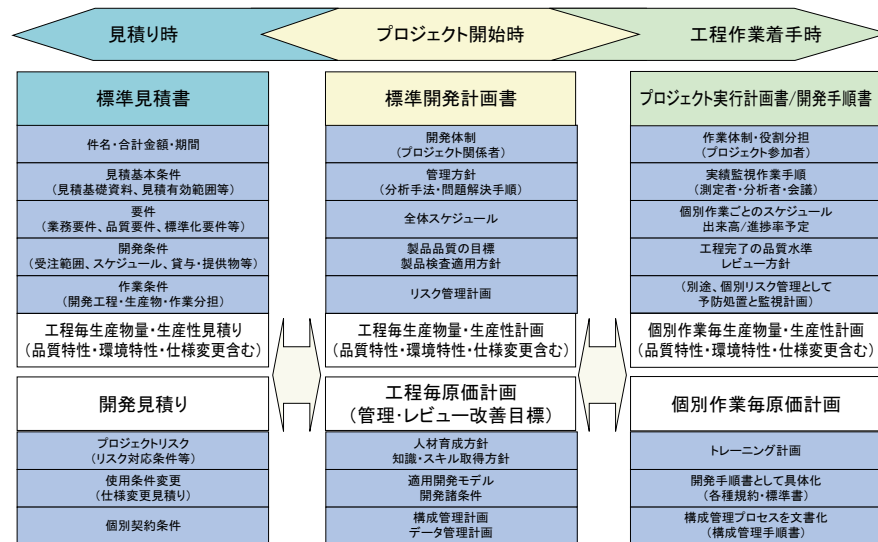


図3 ジャステックにおける見積りとプロジェクト計画との指標連携

計画書および顧客見積書を作成し、価格（開発量、生産性および環境変数）および納期等を顧客と合意します。なお、顧客からの品質要求は開発量および生産性の環境変数に反映されます。

この段階の見積りコスト「 Cl_i 」を次式に示します（「2.2.2 生産管理方式に基づく見積りモデル」参照）。

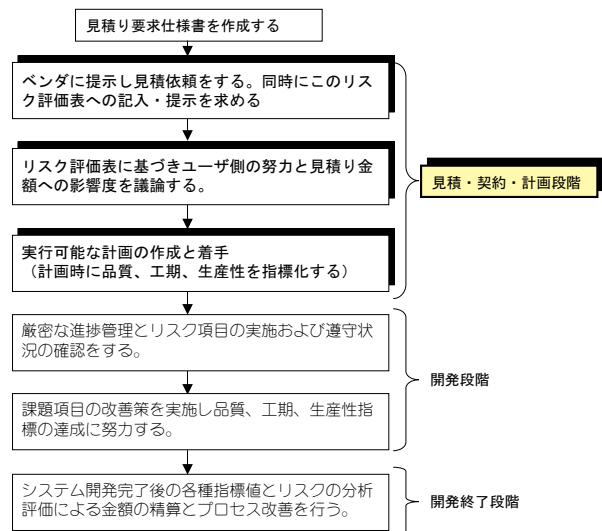
$$Cl_i = V^p l_i \times (1 + al_i) \times P^p l_i \times (1 + bl_i)$$

al_i は生産物量環境変数、 bl_i は生産性環境変数です。

この外部環境変数 al_i および bl_i を利用したプロセス改善について以下に説明します。

ジャステックでは外部環境変数ごとに開発量および生産性への影響度を評価し、見積価格に反映するとともに、その影響度を顧客へ公開することを原則としています。顧客とジャステックは影響度の認識を摺り合わせた上で、相互に協力して外部環境変数ごとにプロセス改善が可能な点を議論し、具体的なリスク低減または改善策を講じる機会を創出しています。表1～表5の外部環境変数表は概要を示しました。実運用上は副特性、変動要素および工程ごとに基準値からの変動率（変動要因ごとに5段階に定義）を評価し見積価格（開発量、生産性）に反映しています。顧客との見積り（契約）段階で、5段階の中から、どの変動率を適用するかがポイントです。ジャステックが提示した外部環境変数（変動率）に対し開発量の削減や生産性の向上につながる具体的な工夫改善が提起され、顧客とジャステックの相互合意に基づき見積価格を低減することが可能となります。ただし、既に紹介しましたが外部環境変数は「顧客の特性による変動」および「プロジェクトの特性による変動」を対象としています。さらに、顧客とジャステックが是々々々な関係を維持しつつ相互に協力する信頼関係を構築することが前提となります。

なお、日本情報システム・ユーザー協会（JUAS）のシステム・リファレンス・マニュアル（SRM）では、この外部環境変数をリスク評価表として取り上げ、使用する手順を紹介しています。図4にその手順を示します。



(出展) システム・リファレンス・マニュアル(SRM:JUAS)
 図4 リスク評価表(外部環境変数)を使用する手順

(2) プロジェクトの開発進行段階

プロジェクトの進行段階は、プロジェクト実行計画書に基づき、「測定→状態監視(問題発生)→問題調査および分析→対策立案および実施」といった監視と制御サイクルが一般的な手順です。

この監視と制御サイクルにおいて、プロセス改善の契機となるのが予実差異分析です。コストに関わる要素「開発量、生産性および環境変数」は「見積りとプロジェクトの計画段階」(以下「見積り段階」)において評価し目標を指標化しているため、「見積り段階」と「開発進行段階」との予実差異分析により改善の効果を定量的に確認できます。しかし、「見積り段階」の開発コストには不確実性が内在するので、プロジェクトの「開発進行段階」では以下の方法を適用しています。

「開発進行段階」の予実差異分析は、「予」を妥当コスト Cm_i に関わる外部環境変数 (am_i 「生産物量環境変数」、 bm_i 「生産性環境変数」) および内部環境変数 ($a'm_i$ 、 $b'm_i$)、および「実」を成行コスト Cn_i に関わる外部環境変数 (an_i 、 bn_i) および内部環境変数 ($a'n_i$ 、 $b'n_i$) を対象にすることで定量的な改善目標を定めています。

妥当コスト Cm_i は「開発進行段階」での改善を配慮した場合の想定実績コストに改善を配慮した残期間の「予測コスト」を加算したコストです。

成行コスト Cn_i は、「開発進行段階」での実績コストに改善を配慮しない残期間の「予測コスト」を加算したコストです。

プロジェクトの「開発進行段階」のコスト削減目標 Cs_i は次式となります。

$$Cs_i = Cn_i - Cm_i \quad (Cn_i > Cm_i)$$

コスト削減目標 Cs_i に対応する外部環境変数 (as_i 、 bs_i) および内部環境変数 ($a's_i$ 、 $b's_i$) は、以下のとおりです。

$$as_i = an_i - am_i \quad (an_i > am_i)$$

$$bs_i = bn_i - bm_i \quad (bn_i > bm_i)$$

$$a's_i = a'n_i - a'm_i \quad (a'n_i > a'm_i)$$

$$b's_i = b'n_i - b'm_i \quad (b'n_i > b'm_i)$$

よって、外部環境変数の削減目標 (as_i 、 bs_i) および内部環境変数の削減目標 ($a's_i$ 、 $b's_i$) に対応するプロセス改善を対策します。

なお、改善によるコストを成行改善コスト Ck_i といいます。実運用では妥当コスト Cm_i と成行改善コスト Ck_i を管理指標として扱っています。

成行改善コスト Ck_i は「開発進行段階」での実績コストに改善を配慮した残期間の「予測コスト」を加算したコストです。

(3) プロジェクトの開発完了時

「開発完了時」における環境変数を利用した予実差異分析は、次案件および他案件の「見積り段階」でのコスト削減への契機となり、また、顧客との客観的な精算行為を促します。

コストに関わる要素「開発量、生産性および環境変数」は「見積り段階」にて評価し目標を指標化しているため、「開発完了時」においても、予実差異分析によ

り改善の効果を定量的に確認できます。さらに、次案件向けの継続的なプロセス改善を推進するため、以下の方法を適用しています。

「開発完了時」の予実差異分析は、「予」を妥当コスト Cp_i に関わる外部環境変数 (ap_i, bp_i) および内部環境変数 ($a'p_i, b'p_i$) および「実」を実績コスト Cq_i に関わる外部環境変数 (aq_i, bq_i) および内部環境変数 ($a'q_i, b'q_i$) を対象にすることで、改善目標を定めています。

妥当コスト Cp_i は、「開発完了時」での改善を配慮した場合の想定実績コストです。実績コスト Cq_i は、「開発完了時」での実績コストです。

以下に改善目標値（コスト削減目標）に関する設定方法を紹介します。プロジェクトの「開発完了時」のコスト削減目標 Cr_i は次式となります。

$$Cr_i = Cq_i - Cp_i \quad (Cq_i > Cp_i)$$

コスト削減目標 Cr_i に対応する外部環境変数 (ar_i, br_i) および内部環境変数 ($a'r_i, b'r_i$) は、以下のとおりです。

$$ar_i = aq_i - ap_i \quad (aq_i > ap_i)$$

$$br_i = bq_i - bp_i \quad (bq_i > bp_i)$$

$$a'r_i = a'q_i - a'p_i \quad (a'q_i > a'p_i)$$

$$b'r_i = b'q_i - b'p_i \quad (b'q_i > b'p_i)$$

この外部環境変数の削減目標 (ar_i, br_i) および内部環境変数の削減目標 ($a'r_i, b'r_i$) に対応するプロセス改善を、ジャステックでは製造統括部 (SEPG) 等の支援を得てチームで対策します。その対策は組織プロセス改善の資産として蓄積していきます。

2.2.5 組織プロセスの改善への契機

環境変数の基準値は年度ごとに蓄積された全プロジェクトの環境変数実績値を

分析し、全社一本で一意的に設定しています。

全プロジェクトの環境変数は外部環境変数および内部環境変数ごとに、統計処理し全社平均値と変動幅を求めています。現在の外部環境変数の変動幅を表6に示します。

この全社平均値の低減による生産性向上、および変動幅の是正による安定性の確保が定量的な組織プロセス改善の目標となります。

表6 外部環境変数のとり得る値の範囲（特異点を排除した値）

種別	特性		副特性	基設	パ設	プロ	統テ	シテ	
				本計	パッケージ計	プログラム成	ス	ス	
生産物量環境変数	品質特性	機能性	正確性、他2項目						
		信頼性	成熟性、他2項目	-20	-20	-15	-30	-30	
		使用性	操作性、他2項目	~	~	~	~	~	
		効率性	実行効率性	+20	+20	+15	+30	+30	
		保守性	解析性、他2項目						
	仕様変更		変更棄却対象率	-5	-5	-5	-5	-5	
			変更正味棄却率	~	~	~	~	~	
			変更追加率	+25	+20	+15	+10	+10	
	生産性環境変数	品質特性	機能性	合目的性、他3項目					
			効率性	実行効率性、他1項目	-30	-20	-10	-10	-20
保守性			解析性、他1項目	~	~	~	~	~	
移植性			環境適用性、他3項目	+30	+20	+10	+10	+20	
環境特性		業務特性	業務ナレッジ						
		ハードウェア特性	使用実績度						
		ソフトウェア特性	使用実績度	-40	-30	-10	-20	-30	
		コミュニケーション特性	組織複合度、他4項目	~	~	~	~	~	
		入力情報特性	業務関連資料、他2項目	+40	+30	+10	+20	+30	
	開発環境特性	開発手法・開発環境、他1項目							

(注1) ジャステック標準工程定義と乖離がある場合、環境変数影響度が限界値を超えることがある。

(注2) ジャステックは環境変数影響度が上記範囲内に収まるよう作業改善を推進する。

さらに、組織プロセス改善目標は年度予算にて開発部門の階層ごと（本部レベル、部レベル、課レベル、チームレベル）に努力目標を設定し、予算執行を行います。

- 環境変数を利用した組織プロセス改善への契機となった事例を次に示します。
- ・環境変数がベストプラクティス（基準値からの変動率が生産性向上および規模の削減に著しく寄与している）となっているプロジェクトを参考に他のプロジェクトへの横展開を図ります。
 - ・組織の階層ごとにプロジェクトに共通する弱みを発見し改善を図ります。
 - ・環境変数が突出して悪化しているプロジェクトを発見し是正を図ります。

2.2.6 プロセス改善効果の還元

プロセス改善は生産性向上と密接に繋がっています。また、向上した生産性向上は、技術者へのインセンティブおよび顧客への還元を配慮することで、プロセス改善活動を活性化することを期待しています。

ジャステックでは創業以来、個人および集団の評価（生産性および品質）を定量的に捉えた能力主義を標榜し、その精錬に挑戦しています。

（1）技術者の生産性と品質を定量的に捉えるための2つの工夫

1つ目の工夫は、ジャステックの生産管理方式（見積りモデル含む）のプロジェクト固有の変動パラメータ（環境変数）を可視化することで、プロジェクトの特異性に左右されない技術者個人の生産性を定量的に捉えることを可能としていることです。

2つ目は、生産管理システムと評価システムとが連動することで相互の運用精度を高めていることです。ただし、評価システムはオープン（評価基準および評価結果）な能力主義と全社員参加型による社内民主主義に基づいていることが大前提となります。

（2）実施局面の留意事項

相互に享受可能な関係を維持するには、顧客とジャステックが是々非々な関係を維持しつつ相互に協力する信頼関係を構築することが前提となります。また、

効果の還元は業界標準（ベースライン）等の配慮や相互の工夫度合に対応する効果を客観的に捉えておく必要があります。

その他のプロセス改善効果を促す取り組みについては、年度予算策定への参加を可能とする予算提案制度、全社員参加の年度予算発表会および生産性向上事例発表会があります。生産性向上事例発表会に関しては「2.2.8 環境変数を使用した改善事例」にまとめます。社員モチベーションの高揚は、独立系ソフトウェア・ハウスとして社内民主主義を維持していくこと、および社員の自立や自律を醸成させる考え方が基本にあります。

2.2.7 支援環境と支援体制

プロセス改善を推進するには、支援環境および支援体制を整える必要があります。

（1）支援環境

支援環境としては、ソフトウェア開発における独自の生産管理システム「ACTUM」を使用しています。「ACTUM」の中心となるのがソフトウェア開発の見積りモデルです。

（2）支援体制

環境変数を利用した組織プロセスの改善に関係する支援体制について図4に示します。標準見積りおよび標準開発計画は顧客のRFP（提案依頼書）に基づき、営業部門と開発部門（ジャステックでは製造部と呼ばれています）とが共同でサーベイ作業を行い生産物量、生産性および環境変数等を見積もっています。標準開発計画は開発部門の達成責任になります。

営業部門の責任は価格折衝責任以外に、生産物量、生産性および変動パラメータの外部環境変数を顧客に公開し、相互共有と協力関係を構築することです。

（3）開発部門の牽制と支援する3つの組織

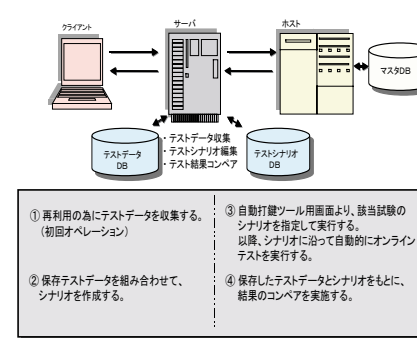
1つ目の組織は製造統括部（SEPG）です。プロジェクトの計画と実績との乖離

(前頁より)

標準的なプロジェクト	品質特性 (共通)	機能性	10	5	0	0	2
		効率性	0	5	0	0	0
		保守性	0	0	0	0	0
		移植性	0	0	0	0	0
		小計	10	10	0	0	2
	環境特性 (改造型)	母体調査ツールの機能水準	0	0	0	0	0
		既存設計書具備状態	5	10	0	0	0
		既存のテスト環境流用水準	0	0	0	-5	-5
		既存母体品質 (正確性)	0	0	0	0	0
		小計	5	10	0	-5	-5
Aプロジェクト	環境特性 (共通)	業務特性	10	6	0	0	8
		ハードウェア特性	0	0	0	0	2
		ソフトウェア特性	0	0	0	0	0
		コミュニケーション特性	10	6	0	3	7
		開発環境特性	3	3	5	5	5
		工程入力情報特性	3	0	0	0	0
		顧客の協力特性	5	0	0	0	0
		小計	31	15	5	8	22
	品質特性 (共通)	機能性	15	5	0	0	8
		効率性	0	5	0	0	0
		保守性	0	0	0	0	0
		移植性	0	0	0	0	0
		小計	15	10	0	0	8
	環境特性 (改造型)	母体調査ツールの機能水準	0	10	5	5	7
		既存設計書具備状態	5	10	0	0	0
既存のテスト環境流用水準		0	0	0	0	0	
既存母体品質 (正確性)		0	0	0	0	0	
小計	5	20	5	5	7		

図6の「事例1：リグレッションテストの自動化」は表8の「既存のテスト環境流用水準副特性」改善事例に対応した実現手段で、「事例2：要件追跡管理の充実」は表8の「母体調査ツールの機能水準副特性」改善事例に対応した実現手段です。

事例1:リグレッション・テストの自動化



事例2:要件追跡管理の充実

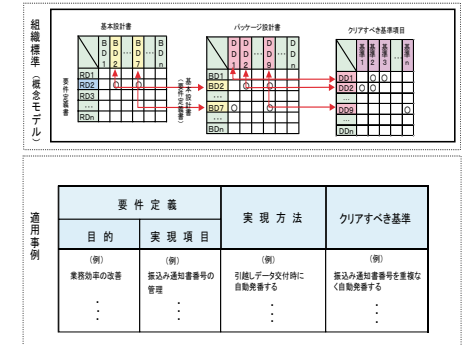


図6 実現手段の事例

表8 差異分析による改善事例

	差異分析結果	改善策および環境変数の改善率(%)	
Aプロジェクト	【開発環境特性】 ・ライブラリー管理機能不備のため、ミスによる再作業の増加	・ライブラリー管理機能の充実 (基本機能にEDIT履歴採取、操作ミス破損防止対応、復元機能等を追加)	6%
	【既存のテスト環境流用水準】 ・保守案件が発生する都度、現行機能を保証するためのテストケース作成、テストデータ作成およびテストオペレーションの負荷が増大	・利用ユーザを巻き込んで、保守用テストリソースを保全し維持管理する (リソース：テストケースおよびテストデータ) ・オンラインテスト (リグレッションテスト) での人手入力を自動打鍵・コンパツールにてテスト負荷を軽減	10%
	【母体調査ツールの機能水準】 ・改造母体が大規模で、ドキュメント (仕様書・設計書等) のメンテナンスが不備の状況下、不慣れな要員が改良を行うと、調査段階から工数が増加する	・要件とソフトウェア・アイテムとの関係図を作成し、改造影響範囲の見極め効率を向上させ、かつテスト漏れを防止する	8%

2.3 活用しているツール・技法

ジャステックでは、ソフトウェア開発の見積りモデルを中心とした独自の生産管理システム「ACTUM」を使用しています。

2.4 人材育成

ジャステックでは、3つの教育方針を基軸とした人材育成教育体系（表9）に基づき、人材育成を実践しています。

- ・技術偏重でなく全人格を重んじた教育
- ・ライフサイクルに根ざした教育体系
- ・義務でなく自主性と自助努力を前提とした教育

●プロセス改善の視点から見た人材育成に関わる3つの特徴

- ・新入社員教育から創造性開発として創意工夫等の体験学習を行っており、生産管理システム「ACTUM」やCMMIレベル5の基礎素養の取得に配慮しています。
- ・教育そのもののプロセス改善を指標化しています。例えば、受講レポート、講師実施報告および要望調査表による効果改善度合い、およびITSSレベル6～7までの人材育成像ごとの達成人数等があります。
- ・経営層がソフトウェア開発に傾注し、プロセス改善を自ら率先垂範しています。

3. 学ぶべき点

ジャステックでは、創立以来ソフトウェア開発技術・管理技術の改善・向上にむけた努力の集大成として「プロセス改善を前提とした生産性原理の具現化」が達成されています。過去実績のベースとした組織能力ベースの見積り（成果物・生産性）と定量的進捗管理が全社的に展開されている数少ない会社であると考えています。このレベルに達するにはかなりハードルは高いと思われませんが、長期的な目標の達成イメージとして大変参考になる事例です。

表9 人材育成教育体系

基礎教育	躰・エチケット	ビジネスマンとしての心構え、基本的素養の習得
	基礎技術	ソフトウェア開発に関する基礎技術および基礎知識の修得
	マネジメント基礎	プロジェクトリーダーの能力およびマネジメント基礎能力の育成
	経営感覚	経営参加に合意した社員による経営への理解
	教養（1）	偏りのない幅広い知識の習得（講座により選択）
応用教育	専門技術教育	最新技術の理解および新技術の発見ならびに開発への応用
	マネージメント	プロジェクトリーダーとしての部下の育成方法および管理技法の開発と実践能力の開拓
	創造性開発	既存の技法、知識を超えて、かつ技術者自身が意識改革を通じて、新しい分野に挑戦する中で創造力を開発
	教養（2）	各界の著名人（学者、文化人、教育者、宗教家、科学者等）の主張または考え方に触れることにより専門的素養の習得と意識の改革
キャリア開発教育	脱ソフトウェア事業家教育	ソフトウェア開発技術者から事業起業家として新しい分野への挑戦
	転職技術教育	ソフトウェア開発以外の事業に転身する技術者へ、その事業に携わるに必要な知識の教育および能力の開拓

参考文献

- 独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター：
ソフトウェア開発見積りガイドブック，オーム社，2006年
- 独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター：
ソフトウェア改良開発見積りガイドブック，オーム社，2007年
- 社団法人 日本情報システム・ユーザー協会（JUAS）：システム・リファレンス・マニュアル，2006年

BEST
PRACTICE
事例5

高品質ソフトウェア開発管理プロセスの実現

～QMTXによる品質メトリクスの活用～

日本電気株式会社
システムソフトウェア事業本部

日本電気株式会社システムソフトウェア事業本部では、オープンミッションクリティカルシステムの基盤となるミドルウェアソフトの開発を行っています。しかし、品質の良い製品やそうでもない製品のばらつきが大きいことが1つの問題となっており、その対策として、組織的にプロセス改善活動を進める取り組みを実施しました。

具体的には、CMMI レベル5のプロセスを土台に、データを収集するための仕組みと収集されたデータを管理部門だけでなく、現場も含めて分析するための仕組み、分析された結果をもとに組織的な改善を推進するための仕組みを構築し、日常的に対応を行うことで高品質なソフトウェアを確保しています。この方式の基本的アプローチは次のとおりです。

- ・ソフトウェア開発に関わるプロセスを組織全体の共通プロセスとして定義する
- ・各プロセスのインプット／アウトプットの定量的および定性的指標を定義する
- ・とくに品質関連については、「品質会計制度」という定量的品質管理の仕組みを全員に周知徹底し、品質の作り込みを推進する

キーワード

- ・定量的プロジェクト管理
- ・品質保証
- ・高信頼性
- ・QMTX

Quality Metrics expatiate systemの略で、品質を定量的に把握し、かつその詳細を明示できる品質システムの内部名称。定量的な品質および生産性情報の他、品質システムに関わる各種規程、是正処置や予防処置実施状況等定性的な情報も含めて維持・管理される

- ・定義されたデータは、開発計画書をベースに自動的に組織データとして収集する仕組みを構築する
- ・組織データや計画書内データをもとに、開発を立案時もしくは計画立案後の早い段階で、

■日本電気株式会社 システムソフトウェア事業本部
(本書で用いる略称：NEC-SS)

●ビジネスゴール

- ・高信頼性システムの提供
「継続的改善を図り、顧客が安心して使っていただける品質を確保し提供する」

●製品・サービスの概要

システムソフトウェア事業本部は、ミドルウェア製品の開発とITプロフェッショナルサービスを行っています。主な開発製品／サービスとして以下のような領域を担当しています。

- ① 統合基盤ソフトウェア、セキュリティソフトウェアの企画／開発／販売
ハードウェア／ソフトウェア障害を自動的に監視し、復旧する自律コンピューティング技術を用いた“統合運用管理ソフトウェア「WebSAM」”や、企業秘密および個人情報を社外に漏らさないようにする情報漏えい対策、サイバー攻撃対策等をトータルなセキュリティとして実現する“協調型セキュリティ「Info-Cage」”等が主力の製品となります。
- ② コピキタスソフトウェア、Webサービスソフトウェアの企画／開発／販売
人と人とがリアルタイムに、しかもお互いの場所や状態に応じて最適な方法でコミュニケーションできる「StarOffice 21」や安全かつスピーディにインターネット・イントラネット対応の業務システムを構築する「WebOTX」等が主力の製品

となります。

- ③ 業務構築運用基盤ソフトウェアの企画／開発／販売
アプリケーションの保守性、運用性、可用性を高めるための基盤コンポーネントである「MCOne」、オープン基盤業務システムを安く、早く、安全に構築するための“システム構築基盤製品「AP-BASE」”等が主力の製品となります。
- ④ IT基盤に関するプロフェッショナルサービスの提供
業種・業務を問わず、顧客のビジネス戦略を支えるためのサービスとして、顧客の課題を引き出し、解決に向けた技術適用提案「IT基盤コンサル」や、要件を確実に満たす個別システムの設計／導入支援「個別プロフェッショナルサービス」等が主力の製品となります。
- ⑤ システムマネジメントサービスの提供
障害を起こさないようにするために、定期的および必要に応じて運用保守作業を実施する「プロアクティブサービス（予防保守）」と障害が発生した場合は迅速に対応し、システムの他の部分やサービスへの影響範囲を最小限に抑える「リアクティブサポート（障害復旧）」を合わせ持った「ハイアベイラビリティサポート(HAサポート)」等ミッションクリティカルシステムの運用をサポートするサービスが主力の製品となります。

URL <http://www.nec.co.jp/>

記載項目の不整合、不適合もしくはリスクに該当する内容を自動的に抽出できるようにする

- ・数値データだけでなく、仕様書や分析結果等の成果物も情報共有化する仕組みを構築する

1. プロセス改善活動の背景

1.1 改善方針

NEC-SSにおけるプロセス改善活動は、CMMI レベル5（2003年3月達成）のプロセスをベースに継続的に取り組んでいます。本取り組みは、顧客／市場の理解、組織能力の理解や最新の設計技法・ツールの動向等をインプットとし、プロセス設計に組み込み、さらなる組織能力成熟度の向上、顧客満足度の向上、および個人の技術力向上を目指すといった活動です。

ここでは主に、組織能力の理解に基づくプロセス改善活動を紹介します。組織能力の理解に基づくプロセス改善は、具体的には次の方針で活動を推進しています。

- ① 開発手順、開発手法、ツール、開発管理方法等を組織として標準化し、組織の全員が、これに基づき開発を進めること
- ② 開発の状況を、データに基づいて定量的に評価し、問題点の把握と適切なフィードバックを行い、ユーザーニーズを満たした製品開発を行うこと
- ③ 顧客の要望・期待や市場環境の変化に応えるために、あらゆる機会を捉えて課題を収集し、真の原因分析と改善策の実施を継続的に進めること

1.2 達成目標

達成目標は、次のとおりです。

- ・ソフトウェア品質向上「リリース後のバグ0件」

1.3 プロセス改善推進体制

NEC-SSでの改善推進体制は、複数の事業部（開発／サービス提供部門）と、それらを横断して支援する事業本部直属部門（スタッフ部門）に分かれています。開発体制は、東京地区を中心とし、神戸、広島、北陸、静岡、岡山等の開発拠点の他、最新の技術動向を調査研究するためのNECアメリカ駐在やインド、中国を第二開発拠点とする海外開発拠点があります。事業本部直属部門には、経営管理機能を行う計画グループと、組織全体のプロセス管理と改善を行う品質保証&CS推進グループがあります。

開発部門は、開発する製品ごとにマネージャを置き、製品責任者として該当製品の企画、設計、開発、保守を担当しています。各事業部には、開発グループの他に支援スタッフとして製品企画グループ、販売推進グループがあります。開発グループ内には、プロセスおよび品質改善推進を担当する品質保証委員がそれぞれ1～2名配置しています。

事業本部直属部門である品質保証&CS推進グループの主な役割は、プロセス改善を推進するための品質システムの構築、維持／改善、開発部門への教育と定着化等があり、開発グループ内に設置した品質保証委員と共同し活動を実施しています。

品質保証&CS推進グループでは、具体的に次の業務と責任を担っています。

- ① プロセス改善推進のための品質システムの構築、改善（品質マニュアル含む事業本部共通の規程類や手順書類作成）
- ② 内部監査計画の立案と実施
- ③ 設計審査（開発計画審査、テスト計画審査、出荷判定）
- ④ フィールド品質／CS情報管理
- ⑤ 是正／予防処置（事業本部共通事項）計画立案、実施および状況フォロー
- ⑥ 組織トレーニング計画の立案、とりまとめ、状況フォロー

1.4 プロセス改善対象組織の規模

プロセス改善対象組織の規模は、次のとおりです。

- ・システムソフトウェア事業本部内の全開発プロジェクトへ展開

1.5 改善活動の経緯

NEC-SSでは、プロセス改善は20数年来実施していますが、ISO 9001の2000年版（2000年12月取得）およびCMMI レベル5（2003年3月達成）をきっかけに、より体系的にプロセスを継続的に改善し、顧客満足の向上を推進しています。

1.6 主な課題と改善のフォーカス

組織およびプロセス全体の状況を把握し、是正処置や予防処置を確実に実施していくことで、大きな改善というよりも小さな改善を積み重ね、後から見れば大きな改革となっているよう、日常的で着実な改善活動の推進をテーマに実施しています。

1.7 顕在化した改善効果

表1に改善効果を示します。

表1 改善効果

項目	改善効果
定量効果	<ul style="list-style-type: none"> プロセス実績データに基づく定量管理の定着 現場および協力会社における過去データを活用した目標管理や分析力の向上 目標/実績差分に対する適切なプロセス制御 組織全体のプロセス品質と能力評価の実施と改善
定性効果	<ul style="list-style-type: none"> 出荷後品質：年間バグ受付数が5年間で約40%減少 納期：納期遅れ（1日遅れ含む）が3年間で約30%改善 生産性：全体生産性(Line/人時)が3年間で約20%改善

2. ベストプラクティス

2.1 参照モデル上でのカバレッジ

① 当該ベストプラクティスに対応するISO/IEC 15504 第5部のプロセス

- ・組織管理 (MAN.2 Organizational management)
- ・品質管理 (MAN.4 Quality management)

- ・測定 (MAN.6 Measurement)
 - ・知識管理 (RIN.3 Knowledge management)
- ② 当該ベストプラクティスに対応するCMMIのプロセス領域
- ・測定と分析 (MA: Measurement and Analysis)
 - ・組織プロセス定義 (OPD: Organizational Process Definition)
 - ・統合プロジェクト管理 (IPM: Integrated Project Management)
 - ・定量的プロジェクト管理 (QPM: Quantitative Project Management)
- ③ 当該ベストプラクティスに対応する共通フレーム2007のプロセス/アクティビティ
- ・管理プロセス

2.2 プラクティスの説明

2.2.1 プロセス改善への取り組みの背景

データが収集されていても、データ分析、活用が十分でない場合が多くあります。そのため、定量管理の規範を整備するためには、データの収集、蓄積、分析、

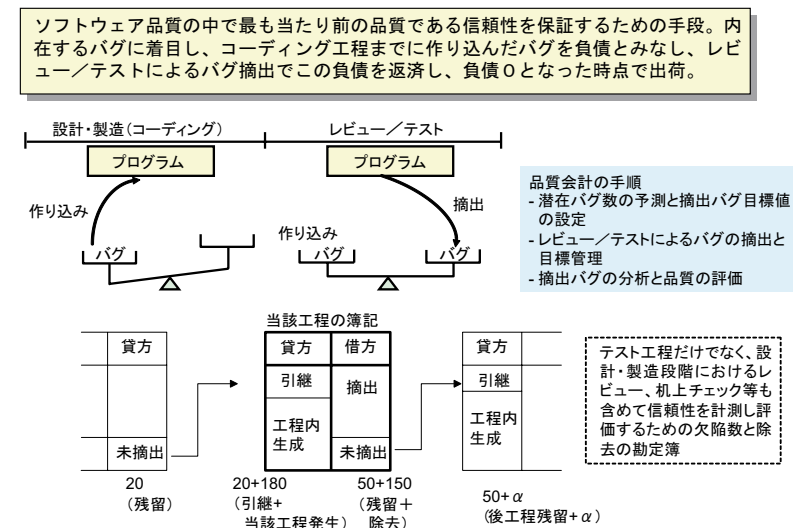


図1 品質会計制度概要（定量的品質管理（品質会計制度））

活用をサイクリックに展開していく必要があります。この計量管理を効果的かつ効率的に展開するための施策として、品質会計制度と QMTX を基盤とした定量管理活動の支援環境整備を行っています。

品質会計制度では、過去の実績データおよび当該ソフトウェアの規模や難易度を考慮して各工程で抽出すべきバグ数の目標値を設定し、実績との対比を行い、都度品質を制御します。品質会計制度の概要を図1に示します。

この品質会計制度に基づく品質管理を拡張して、組織ごと、プロジェクトごとに制定したプロセスおよびプロダクトの品質目標（規模、スケジュール等を含む）をフェーズ単位に展開し、多角的な観点からの定量管理を実施しています。

2.2.2 QMTXとは

QMTXとは、品質を定量的に把握し、かつその詳細を明示できる品質システムの実現のために設定した NEC-SS での内部名称です。プロセスの実施状況を監視するためには、定義されたプロセスからのアウトプットや統計データを吸い上げ、組織実績として整備し、改善を推進していく方法が課題となります。

この組織では、これらの課題を解決するために、次の方法を取ることにしました。その運用イメージを図2に示します。

- ・ 規程や標準類および品質記録の全面電子化、電子化データ活用機会の増大
- ・ ワークフローを用いた情報の収集
- ・ 開発計画書による情報管理（統計データ／記録）と組織 DB への自動収集
- ・ 収集されたデータを品質システムホームページで一元管理

QMTXでは、定量的な品質および生産性情報のほか、品質システムに関わる各種規程、是正処置や予防処置実施状況等定性的な情報も含めて維持・管理されます。QMTXで中核としている情報の1つに開発計画書があります。開発計画書には、当該開発における開発の狙いや目的、機能性、信頼性、操作性、効率性、その他品質目標、開発日程計画、体制、品質計画（バグ抽出目標、レビュー予定工数）、タスク管理計画等の定量的な目標および定性的な計画の概略も含めてプロジェクトの目標および全体像が記されます。

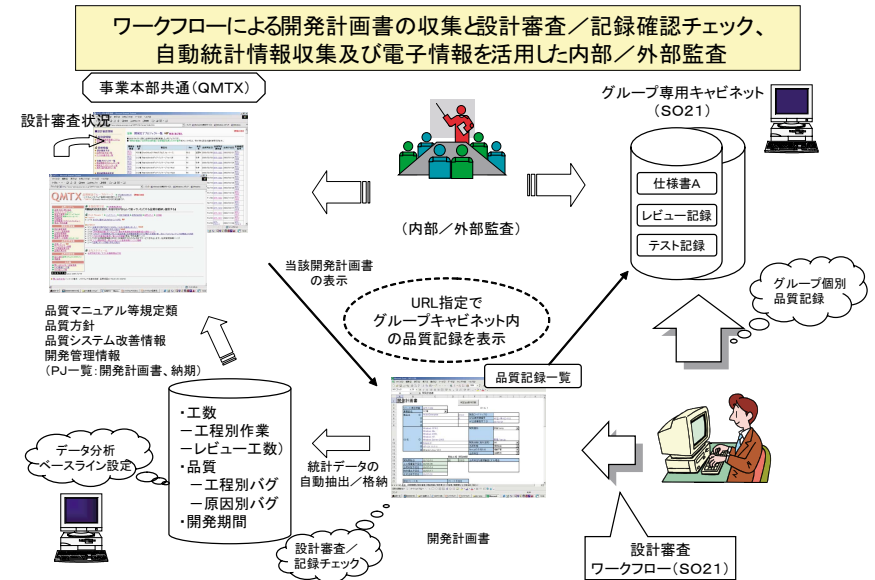


図2 統計情報の収集と活用(品質情報(統計データ／記録)の収集と管理)

この開発計画書は、これ以降に発生する設計審査、開発作業の実績管理、出荷判定、製品（媒体）の出荷に至る基本帳票として利用します。そしてデータだけではなく、計画書に対応した品質記録を参照することが可能です。つまり、現物（成果物）による確認が可能です。開発計画書は、Excelで作成し、1ファイル17シートで構成されています。その一部を図3に示します。

開発計画書では、未入力チェックの他にデータ間の整合性や組織ベースラインと差が大きい場合のチェック、実績が目標値からの基準値（管理限界値）を超えた場合の抽出等が自動的に行われ、計画立案および工程管理が効率よく実施できるよう工夫しています。その概要を図4に示しますが、具体的には、開発計画書で記載された様々な目標に対して実績データを同開発計画書に入力します。すると、目標との差分（相対誤差）の管理限界値を超えた場合には、自動的に開発計画書（理由見解シート）内で該当項目にアラーム記号（×）の文字が表示されま

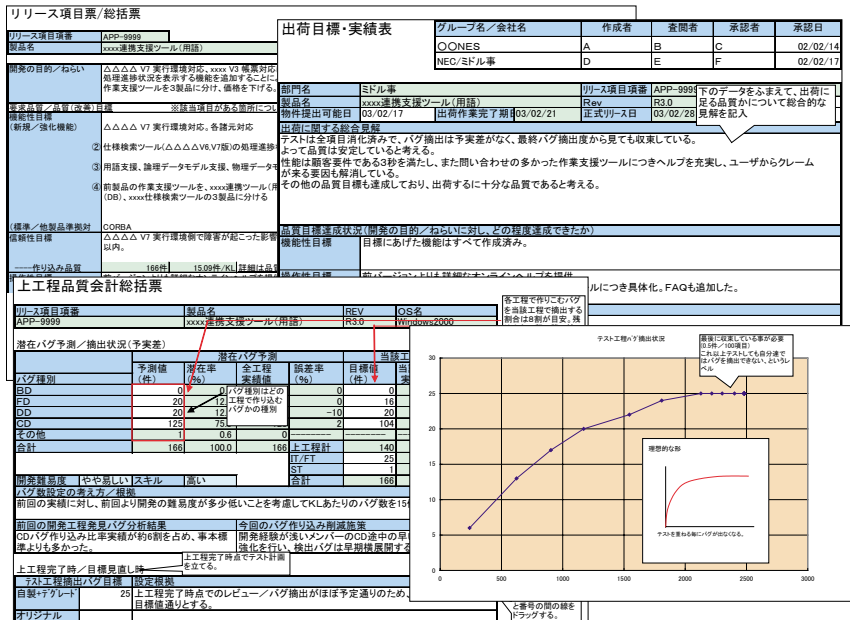


図3 開発計画書（一部）

す。それを不適合（データ分析／計画見直しタイミング）とし、該当項目に対して理由／見解（分析結果）を開発側が記載します。該当項目に対し、理由／見解が記載されていない場合には、開発計画書承認時にアラームが表示される等、各工程完了時に、状況把握と今後の見通しが確実に実施されるように工夫しています。

工数の見積りは、規模と開発の難易度およびスキルレベルにより、組織DBに蓄積された過去実績データからの回帰モデルを利用して算出されます。ただし、この値をそのまま利用するのではなく、該当算出値をベースに+αで考えられる要因を考慮し決定します。

各開発工程別の工数は、組織DBの過去実績情報をもとにした全体工数の割合から得ることができます。なお、工程別工数検討時には、工程に応じ工数をドラフト作成工数、レビュー工数、レビュー対応工数の3つに分け算出します。ドラフ

開発計画書のExcelによる自動チェック

- プロジェクトの失敗事例等をもとにチェック項目を随時追加、チェックそのものは自動的にチェック
- 目標管理（予定／実績差）
- 管理限界値外項目のアラーム
- 管理限界値外に対する、原因を分析し理由・見解を明示

人手によるチェックは、記載内容の妥当性確認中心へ



図4 統計データを活用したプロジェクト管理（データによるプロジェクト管理（工程管理）の実施）

ト作成工数は、ある程度安定した値を求めることができますが、レビュー工数およびレビュー対応工数はドラフト版の品質状況やレビュー者数等の変動要因を考慮して見積る必要があります。

より合理的な見積りができるように、開発計画書内において、仮に設定した値からドラフト版に関する生産性（ページ／人時）、全体開發生産性（Line／人時、レビュー等含む）および工程別生産性、工数配分、レビュー工数比率（当該レビュー工数／当該工程工数：％）、文書化率（ページ／規模）等複数の指標が自動計算され、その情報をもとに妥当性の確認を行えるようになっています（図5）。

NEC-SSの個々のプロジェクトでは、前述の方式を整備し、定量的情報をもとにした改善を進めています。また並行して、それらの開発計画書情報を収集し、組織全体としての効果を管理し、改善していく仕組みも整備しています。具体的には、図6に示すように組織のプロセス実績情報を掲示し、四半期ごとおよび半期、年度での組織全体の状況、プロセスごとの状況、あるいは事業部ごとや事業部比較等の定量データを組織全メンバに公開し、現状認識を正しく行うための仕組みとしています。そして、四半期ごとの定期的な品質システム改定に合わせ、前回の改善施策で期待通りの効果が出ていない場合の原因分析や追加施策検討、次に

- ・組織プロセス実績データをベースラインとして提示
- ・基本データ(入力値)により指標データを提示
- ・ベースラインとの比較により計画値の妥当性を評価
- ・計画値と実績値比較により妥当性を評価

管理指標

※品質データ情報掲載URL： <http://asou-plans.asou.nec.co.jp/QMTX/dq/quality-index.htm>

リリース項目番号	APP-9999	OS名	Windows2000
製品名	xxxx連携支援ツール(用語)	REV	R3.0

品質管理指標	基本設計	機能設計	詳細設計	GD/UT	IT/FT	システムテスト	上工程	下工程	全体
バグ抽出数(件/KL)	標準値 >0.1	4±3	4±3	8±5	<5.5	<0.5	16±3	<6	20±10
予定	1.45	1.82	9.63	2.31	0.09	12.86	2.41	15.32	15.32
実績	1.45	1.82	9.72	2.31	0.09	13.06	2.31	15.32	15.32
(件/人)	標準値 <0.5	<1.0	<1.2	<1.5	<0.15	<0.05	<1.0	<0.10	0.2±0.1
									0.27
									0.28
バグ抽出配分(%) (合計=100)									100
									100
レビュー工数比率(%)									17.6
									18.4
レビュー網羅度(H/KL)									28.70
予定	9.09	5.45	9.26	2.78	1.85	24.07	4.83	28.70	28.70
実績	7.64	4.91	11.11	2.31	1.85	23.89	4.17	28.06	28.06
テスト網羅度(H/KL)									>25
予定	4.63	21.30	7.41	4.63	29.70	33.33			33.33
実績	4.63	23.15	9.26	4.63	32.41	37.04			37.04
文書化率(ヘッダ/KL)	標準値 >0.5	>2.5	>3		>1.5	>0.5	>6	>2	>10

図5 組織データの活用(組織ベースライン値および計画値との比較)

実施すべき弱点領域の施策検討等、それぞれ効果を確認しながらの改善サイクルを回し、組織全体のプロセス改善を実施しています。

2.2.3 QMTXの生まれた背景

NEC-SSでは、品質およびプロセスを改善する上でこれまでに示した方針で活動を実施し、顧客ニーズに的確に応え、安心して使えるソフトウェア製品およびサービスを継続的に提供できる組織にすることを推進しています。規程や手順書類の電子化および一元管理は、一般的なことであり、ホームページ等での掲示も容易です。しかし品質記録は、定義されたプロセスから生じる文書や記録類であり多種多様で、数の多さからも文書の管理として一番やっかいな部類のもので管理工数もかかるため、品質記録も含めてすべて電子化し、シンプルに一元管理する必要があります。また、日常的な記録類を随時確認するだけでは、プロジェクトの状態の判断結果が属人的になってしまうため、プロジェクトの状況を判断する基準を構築する必要があります。

- プロセス能力ベースライン
- ・出荷品質：出荷後フィールドソフトミス件数/半期
 - ・生産性：新規+改造規模(Line)/人時
 - ・期間：単位規模当たりの開発日数
 - ・工数分布：全体工数に占める各工程の割合、設計工程におけるレビュー工数
 - ・作り込み欠陥率：新規+改造規模当たり作り込みバグ数
 - ・欠陥分布：工程別作り込み欠陥率
- 等

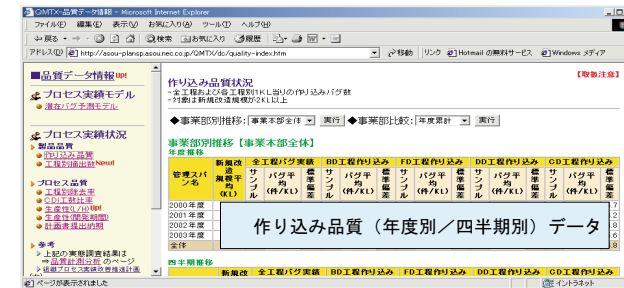


図6 組織プロセス能力の確認(組織プロセス実績情報)

2.2.4 QMTXによって変わるプロジェクト管理の位置付け

内部監査および外部監査実施の際、品質記録確認はすべてディスプレイ(電子プロジェクタ)で確認します。日常での設計審査時も同様に電子キャビネット内の記録を確認する仕組みをとっています。すなわち、何かあったら関連資料を開発部門から提示し確認するというのではなく、日常的に第三者がプロジェクトの実施状況を現物確認しながら監視ができるということです。

また、工程の制御は、QCDに到る様々な定量的目標値が存在します。データを人手で確認する際の漏れを防ぐために、自動チェックという仕組みを構築することで、プロジェクトがうまくいっているのかどうかを目標との差として明確にし、判断基準が確立され、適切なタイミングでのデータ分析および計画の見直しも含めた対策が確実に実行できるようになります。

2.2.5 QMTXによって変わる開発監査体制と役割

開発計画審査により、開発計画書の内容について、プロジェクトが計画通りに遂行できる内容か、とくに品質面に関して適切な考慮が払われているかを品質保証部や企画審査部門等の第三者も含めた審査が可能となります。そのため、品質

目標値（摘出バグ目標値やレビュー予定工数等）の設定が妥当か、また設計段階とテスト段階の完了予定日程、工数計画は適切か等を主体に自動チェック結果をガイドとして利用し、より適切な確認を効率的に行うことができます。

2.2.6 QMTX 導入によって得られた効果

QMTXにより、プロセス実績データがソフトウェアメトリクスとして、ソフトウェア開発プロセスまたはソフトウェア製品の定量的な特徴付けが可能となりました。

例えば、開発計画立案時には、完了済みプロジェクトのデータを参考に新規の開発計画の立案、目標設定が可能となります。これによりプロジェクトのプロセス制御が、計画値と実績値の比較によるプロジェクトの状態確認、または差分による適切なアクションの実施等の方法により、適切に実施できるようになりました。

また、組織全体のプロセス品質と能力等の評価や改善を行う際、メトリクスデータを用いることで、経年変化の傾向や強みや弱み等を定量的に把握でき、改善に結び付くようになりました。

これらの結果、定量的には表2に示す効果が現れています。

表2 QMTXによる効果（定量）

対象	効果
①出荷後品質	年間バグ受付数が5年間で約40%減少
②納期	納期遅れ（1日遅れ含む）が3年間で約30%改善
③生産性	全体生産性(Line/人時)が3年間で約20%改善

2.2.7 組織文化—不具合から組織が学ぶ

NEC-SSは、1件のバグでも大切に考える文化があります。その一例をバグの事前摘出という観点と出荷後のバグ発生という2つの観点で取り上げます。

（1）バグの事前摘出

プロジェクトで設定したバグ摘出目標は、コミットメントとして捉え、1件の誤差も許さないという考え方が浸透しています。例えば100件のバグ摘出目標値を

設定していた場合、99件の摘出実績があっても、まだ1件はあるはずと考え、再レビュー／追加テストを行い、バグを見つけようとします。その結果、どうしても見つけきれなかった場合には、はじめに設定した目標値の妥当性を再検証し、目標の見直しが必要と判断した場合には、そう判断した理由と目標値を99件と下方修正し潜在バグはないと再宣言を行います。

実際には、残存するバグが0ということはなく、出荷後にバグが発見されることはあります。その際、バグ0と思ったにも関わらず発生した事実に対し、なぜなぜ分析を実施します。出荷時に0と思って分析するのと、「バグはあるに決まっている」と思って出荷するのでは、その後のバグ分析に大きく影響します。もうないと判断したからこそ、なぜなぜ分析が生きてくるという考え方があります。

「1件のバグであっても出荷先にとっては100%起きる障害」という認識のもと、バグ1件ごとに行うなぜなぜ分析は、品質グループと開発グループで相当数の工数をかけ、何度も分析結果を議論し、次のプロセス改善へとインプットしています。なぜなぜ分析結果は、当該プロジェクトのノウハウとするだけでなく、組織関係者が全員参加の品質分析会議（毎月）で紹介され、他プロジェクトでもノウハウとして横展開されています。

（2）出荷後のバグ発生

横展開することによって、他のプロジェクトでも改善が進んだ1例として、「レビュー体制の確認強化」を取り上げます。NEC-SSでのレビュー体制は、インスペクション方式および品質会計制度を採用しています。通常の開発であれば、この併用方式で問題はないのですが、スケジュールがかなりシビアなプロジェクトで、かつ遅れが生じはじめたような場合に様々な対策を打つこととなります。その対策の中で、通常は各人の旗振り役、つまり制御や成果物のレビュー等を実施しているリーダーがQCD確保の中心役となっていました。しかし、上記状態となったことで、該当リーダーも多くの実作業を受け持つことになり、その結果生じた遅れを取り戻すことを挽回の施策の1つとして実施しました。該当リーダーは、通常の実作業もこなし、実作業にも十分対応できたのですが、そこに1つの落とし穴がありました。その落とし穴とは、該当リーダーが作成した成果物レビューの体

制でした。

該当リーダーが作成した成果物をレビューする際、該当リーダーと同等以上の力を持つ担当者が当たらなかったため、該当リーダー自身が該当成果物をレビューするレビューリーダーも兼務しました。レビューは実施したものの、各レビューアの気持ちや、該当リーダー自らが作成した成果物なのだから大丈夫という思い込みが発生し、さらにはレビューアのスキル不足から致命的な障害を見逃すといったことが発生したのです。それを契機に、レビューリーダーと成果物作成者が同じになることを不適切な体制とし、「上位管理者が実施責任を持つ」という改善を実施するようになりました。そして品質システムも改定され、上記状態抽出と上位管理者の対応状況が第三者にもわかるようになりました。

2.2.8 活用しているツール、技法

開発計画書は、Excelとマクロ機能を活用しています。マクロによるチェックや計算式は2,000個以上設定され、マクロには、Excelで設定されたデータを自動的に組織DBへ格納する機能や、工程途中でのバグ見直し予測モデルによる推奨値提示等があります。ユーザインタフェースは、Excel帳票画面のため、特別な環境がなくとも誰もが本機能を活用しプロジェクト管理を実施することが可能です。

2.3 人材育成

QMTX含む品質システムの改善は、四半期ごとに実施しています。その際は、開発グループ内の各品質保証委員向けに説明会をその都度実施し、組織内の定着／改善を図っています。個別には開発計画立案技術、レビュー技術、テスト技術等体系的な集合教育の定期的な実施や、計画登録時に開発計画書個別教育必要とチェックしたプロジェクトに対する個別説明等を実施し、適切な利用促進を図っています。

また、新人教育や異動者教育の一環として、品質システム概説教育（初級教育）や、新たに異動されてきた開発者に対する個別説明会を実施し、品質システムに関する一般的な知識教育を行っています。

3. 学ぶべき点

データを用いて改善するという方法は、誰もが理解し取り組んでいることですが、それを組織的に徹底して実施するというのがNEC-SSの事例です。本事例でのプロジェクトと組織をうまく連携させ、組織的なプロセス改善に取り組むアプローチは参考になります。また、開発計画書をExcelの機能を用いて実装することで、費用対効果や保守性を高めており、現場への普及促進をしやすい点も見習うべきところがあります。

参考文献

織田巖：ソフトウェアレビュー技術～基礎から実践までのノウハウ～，ソフトリサーチセンター，2006年

Capers Jones：ソフトウェア開発の定量化手法，共立出版，1998年

Robert B. Grady：ソフトウェアメトリクス，日経BP社，1990年

真野俊樹：見積の方法，日科技連出版社，1993年

瀧本法良：ソフトウェアプロセス改善と品質保証の実際，日本テクノセンター，2004年

BEST
PRACTICE
事例6

マネジメント自らが予防のための行動を取り、問題が起こりにくい組織に

～Preventive Management による実現～

情報技術開発株式会社

情報技術開発株式会社は、プロセス改善のモデルとしてSW-CMMを採用し、tdiソフトウェアプロセス標準（以下、「tdi-SPS」）を構築、2004年2月にはSW-CMMレベル3を請負開発の全部門で達成しました。当時CMMのレベル達成は、各社で取り組まれていましたが、全社規模でのレベル3の達成は、数少ない事例です。

しかし、「tdi-SPS」のプロセスを構築し、SW-CMMレベル3を達成しても、時には、トラブル（赤字）プロジェクトが発生してしまうことがあり、このことが経営面で大きな課題と認識され、その原因が分析されました。その分析の結果、トラブルプロジェクトの予兆は、案件引合い段階から受注までの間に確認できることがわかりました。そこで、SW-CMMレベル3のキープラクティスを中心に、受注後のプロジェクト計画からプロジェクト完了までのプロセスを中心に構築された「tdi-SPS」に、案件引合い段階までのプロセスを追加することで対応することとしました。

その結果、「案件会議」（※1）、「リソース会議」（※2）、「提案レビュー」（※3）を追

キーワード

・ Preventive Management

英語の Prevent、「起こらないようにする、予防する、防止する」という意味に Management を付加した言葉。つまり、トラブルが起こらないように日常からのマネジメントが重要であり、これらをチームで解決することが重要であると TDI では社内に指導している

・ tdi-SPS

TDI Software Process Standard

・ MAGMA

TDI Order Management and Resource Matching System

加する等、案件引合い～プロジェクト計画の承認までのプロセスを強化し、このプロセスの遵守を徹底することで、その期に受注した一定規模のプロジェクト全てが黒字に転換したのです。

これら一連のプロセス改善活動は、開発部門と営業部門がトップダウンで取り組んだものです。2つの関連する組織の上位マネジメントが参加した取り組みであったことが大きな特徴であり、その後の TDI のプロセス遵守の継続性に大きく影響を及ぼしました。また、印象に残るシンプルなキーワード「Preventive Management」を掲げることにより、「問題が起こらないように、マネジメント自らが予防のための行動を取り、チームで取り組む」思想が各部門へ展開され、継続できたという事例です。

※1～※3のプロセスは、「2.2.2 TDIのプロセスの概要」を参照願います。

■ 情報技術開発株式会社 (TDI CO., LTD.)
(本書で用いる略称：TDI)

● ビジネスゴール

- ・ サービス価値の向上
- ・ 顧客満足度の向上
- ・ 利益率の向上
- ・ 生産性の向上

● TDIの製品・サービスの概要

- ・ 認証システム「@Face シリーズ」
- ・ 統合IT資産管理ソフトウェア「LANDesk Management Solutions」
- ・ ウイルス・スパム対策とセキュリティの PandaGateDefender「Panda」
- ・ メール暗号化ソリューション「PGP」
- ・ 企業内データアクセスソリューションの

「Intellisync Mobile Suite」

- ・ データ持ち出しによる情報漏えい防止・監視システム「4thEye」
 - ・ 検疫ソリューション「LANDesk Apresia 連携 検疫システム」
 - ・ 組込ソリューションサービス&CASE ツール「ZIPC」
 - ・ コールセンターサポートシステム「おてがるコールセンター」
 - TDIのサービス
 - ・ ソフトウェア開発
 - ・ 情報処理サービス
 - ・ エンベデッド・ユビキタス/半導体関連
- URL <http://www.tdi.co.jp/index.htm>

1. プロセス改善活動の背景

1.1 改善方針

改善方針は、次のとおりです

- ・ 請負ビジネスの拡大に向けた活動の推進
- ・ ラインスタッフの主導、製販協業による確実な提案プロセスの構築

1.2 達成目標

達成目標は、次のとおりです。

- ・ プロセス遵守率90%以上の徹底

遵守率 = (実施した会議、レビュー回数 / 計画されている会議、レビュー回数)

- ・ 案件会議、提案レビュー、SQA レビュー開催それぞれの開催率90%以上
- ・ プロジェクト課題のトラッキング (トラッキング率90%以上、期限内達成率80%以上)

1.3 プロセス改善推進体制

コンピテンシー部が中心となり、開発部門との兼任メンバーからなる SQA 担当者、SEPG 担当者と共に、活動を推進しています。

① SQA メンバ (開発部門の約7%弱)

SQA メンバには、コンピテンシー部と部長推薦の開発部門の社員を登録し、必要時に SQA レビューに参加しています。また SQA 担当者は、毎年見直します。

② SEPG メンバ (開発部門の約2%弱)

SEPG メンバには、コンピテンシー部と開発部門の社員 (兼任) がプロセス改善に従事しています。

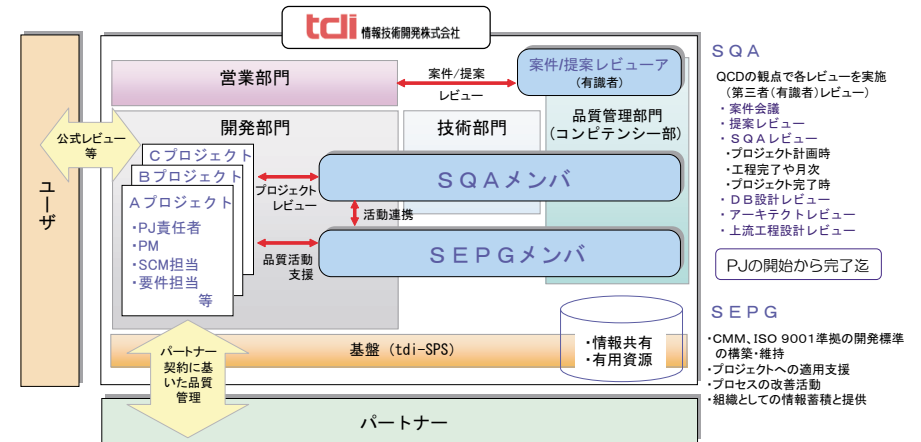


図1 プロセス改善推進体制

1.4 プロセス改善対象組織の規模

プロセス改善の対象組織は、営業本部、開発本部およびEUL事業本部の3組織です(図2)。

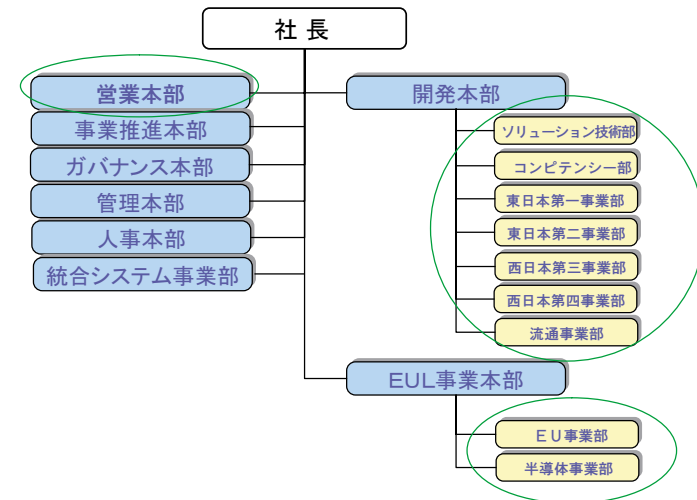


図2 プロセス改善対象組織

1.5 改善活動の経緯

TDIはソフトウェアの品質管理の向上に着目し、2000年に開発現場とは独立した「品質管理部」を設置しました。この品質管理部は、主に問題プロジェクトが発生した場合に、該当するプロジェクト関連ドキュメントを中心に調査、原因分析を行い、社内に改善提案を行っていました。

その後「プロジェクト監理部」と改名され、プロジェクトの順調な進行に向けた監督・指導、およびプロジェクトへの支援機能を組織のミッションに加え、本格的なプロセス改善を開始しました。

その結果、2004年2月にはSW-CMMレベル3を請負開発の全部門で達成しました。さらに、2005年には過去の失敗プロジェクトの教訓をもとに、「案件引合い」から「プロジェクト計画の承認」までのプロセスを見直し、全社一丸となってプロセス遵守に取り組んだ結果、期中の赤字プロジェクト撲滅という成果を得ることができました。2007年4月以降「プロジェクト監理部」を「コンピテンシー部」として改名し、機能拡充とともに「Preventive Management」をキーワードとしたプロセス改善活動を推進しています。

1.6 主要な課題と改善のフォーカス

TDIでの主要な課題と改善のフォーカスを表1に示します。

1.7 顕在化した改善効果

「1.2 達成目標」に示した目標を達成したことにより以下に記述する改善効果が得られました。

(1) 案件の早い段階からの提案方針の検討やリスク、課題のチェックとその対策の完遂

① 案件会議では、どのような方針で提案するかを迷っている段階から、各部門の上位マネジメントからの提案方針に関する意見やアドバイスが得られるため、提案作成段階での手戻りが減少しました。さらに案件会議以降のプロセスへ課題トラッキングが引き継がれ、発見された課題は、すべて期限内に解決されて

表1 主要な課題と改善のフォーカス

主な課題	改善のフォーカス
案件の早い段階での ・提案方針の検討 ・リスク、課題のチェックとその対策の完遂	・案件会議、提案レビュー、リソース会議、プロジェクト計画レビュー等でのリスクや課題の洗い出しと継続的なトラッキング
課題解決方針決定のスピードアップと首尾一貫性の確保	・各会議、レビュー等に開発本部長（開発の意思決定者）と営業本部長（営業の意思決定者）が参加
課題に対する具体的な指示 ・対応スピード向上 ・対策の有効性向上等	・SQAレビューア育成と有識者の参加
全社的な経営資源の活用	・開発本部を横断的な組織に変更して開発要員のタイムリーな異動を実現
SQAレビュー結果(QCDの○△×評価)の社内共有化、見える化	・SQAレビュー結果を経営層の会議で報告、また社内サーバによる共有化

次工程に進むことが可能となりました。

② 提案レビュー時の見積り内容の検証、TDIに発注していただく価値が盛り込まれた提案であるか、等の確認段階において適切なアドバイスが多く出されるようになり、リスク回避につながりました。

●指摘事項の主な分野

- ・見積り根拠の曖昧さ
- ・工程比率の是正
- ・テスト計画の曖昧さ
- ・工数見積りの甘さ、漏れ是正
- ・標準、適正単価の検証
- ・受入検査、レビュー等の工数漏れ是正、WBS検証
- ・開発工程の重なり（要件定義、外部設計、内部設計等）
- ・規模感とスケジュールの乖離是正

(2) 課題解決方針決定のスピードアップと首尾一貫性の確保

各会議やレビューの場に開発本部長（開発の意思決定者）と営業本部長（営業

の意思決定者)が参加することにより、首尾一貫した指示が出されるようになり、速やかな課題解決につながるようになりました。

(3) 課題に対する具体的な指示

各会議やレビューの場に開発本部長(開発の意思決定者)と営業本部長(営業の意思決定者)が参加することにより、その場で解決方針が得られるため、事後のネゴが不要となり、決定が速くなりました。

また、その場での指摘内容は具体的なものとなるよう努めたため、指摘を受けた人が「どのように対処すればいいかが理解できる」ようになりました。

- ・指示が具体的なので課題対応内容も効果的である
- ・意思決定者からの指示なので手戻りがほとんどない
- ・担当者の課題対応スピードが向上する

(4) 全社的な経営資源の活用

開発本部をリソースの有効活用を考慮した横断的な組織に変更したことにより、全社的な開発力、ノウハウ、リソース等が有効に活用できるようになりました。

- ・部署、部門の壁を越えた横串のプロジェクトマネージャのアサインやプロジェクト支援
- ・開発要員のタイムリーな異動
- ・プロジェクトマネージャ、プロジェクトリーダークラスのタイムリーな異動
- ・全社共有サーバに蓄積された社内事例の共有化

(5) SQA レビュー結果(QCDの○△×評価)の社内共有化、見える化

SQA レビューの結果は毎月○△×で評価され、社長・取締役等が参加する会議で、プロセス品質やソフトウェア品質に関するよい点や改善点が報告されます。このことにより、「レビューされ、評価されることは、効果がある」という認識が社員に浸透し、良い状態になっています。また共有サーバに情報が格納されることで「見える化」にもつながってきています。

2. ベストプラクティス

2.1 参照モデル上でのカバレッジ

- ① 当該ベストプラクティスに対応する ISO/IEC 15504 第5部のプロセス
 - ・プロジェクト管理 (MAN.3 Project management)
 - ・品質保証 (SUP.1 Quality assurance)
 - ・検証 (SUP.2 Verification)
- ② 当該ベストプラクティスに対応する CMMI のプロセス領域
 - ・組織プロセス定義 (OPD: Organizational Process Definition)
 - ・要件管理 (REQM: Requirements Management)
 - ・プロジェクト計画策定 (PP: Project Planning)
 - ・プロジェクトの監視と制御 (PMC: Project Monitoring and Control)
 - ・検証 (VER: Verification)
 - ・プロセスと成果物の品質保証 (PPQA: Process and Product Quality Assurance)
- ③ 当該ベストプラクティスに対応する共通フレーム2007のプロセス/アクティビティ
 - ・管理プロセス
 - ・品質保証プロセス
 - ・検証プロセス

2.2 プラクティスの説明

2.2.1 Preventive Management

失敗プロジェクトの原因として、以下の2つに着目しました。

- ① ソフトウェア開発部門のほとんどに開発および管理プロセス記述が存在しない
- ② 開発および管理プロセス記述が存在していても、実行していない

TDIは、SW-CMMをもとに①のプロセス記述として「tdi-SPS」を構築しましたが、TDIのビジネスゴールを達成するためには、さらなるプロセス改善が必要でした。

その改善ポイントとしては、過去のトラブルプロジェクトの教訓をもとに「案

件引合い」～「プロジェクト計画」の承認までのプロセスを見直し、追加するとともに三権分立を目指した組織化（営業、製造、プロジェクト管理）を行ったことが挙げられます。これにより、客観的なチェック機能が働き、プロセス遵守につながりました。

次に②のプロセスの実行を「Preventive Management」の方針の下で予防管理を徹底指導して、プロジェクトマネージャだけに負荷をかけるのではなく、関係者全員でプロセスを遵守することにより赤字撲滅の成果を得ることができました。

実行したプロセスおよび活動の一例を表2に示します。これらの活動には、開発部門および営業部門の上位マネジメントが参加し、課題やリスクへの指摘等を徹底指導しています。

表2 実行プロセスと活動の一例

プロセス	活動	ポイント	プロセスオーナー
見積提案プロセス	案件会議	良いソリューション提案につなげるための検討会議	開発本部
	提案会議	見積提案内容のレビュー、見積り根拠の確認、等	営業本部
	リソース会議 (必要時)	リソースの最適化	開発本部
プロジェクト計画プロセス	プロジェクト計画レビュー	実現可能な計画策定のレビュー	開発本部

<以下、省略> 詳細は、「2.2.2 TDIのプロセスの概要」を参照

プロセスを作り過ぎるとその弊害として、現場スタッフに「そのとおりにやれば良い」という気持ちが蔓延しがちとなり、改善しよう、問題はないか、等の意識が薄れることが想定されます。プロセスが存在し、ただ実行しているだけでは、間違いなくトラブルは起きてしまいます。「Preventive Management」こそ、トラブルの予兆に早く気づき、早期に対策を打つための、最も重要なアクションなのです。

作業は、部下やメンバにまかせられたとしても、「今のやり方で問題はないか?」という問題意識は、マネージャ自らの自意識によらないと発見できません。また、プロジェクトマネージャに指示してもすべてが解決するというものではなく、起

きている事実を自ら確認するとともに、その対策をマネージャを含めたチーム全員で解決することが大切です。

「開発および管理プロセスを確実に実行する」という当たり前のことを実行するのは、大変難しく、各社共に悩まれていることです。TDIの事例は、当たり前のことをマネジメント自らが行動で示すとともに「Preventive Management」を組織に展開することで成果が得られ、組織力が向上した事例といえます。

2.2.2 TDIのプロセスの概要

以下に「Preventive Management」のもとで実施しているTDIの品質保証活動の概要を示します(図3)。なおSQAレビューは、品質保証活動の1つとして位置づけて実施しています。

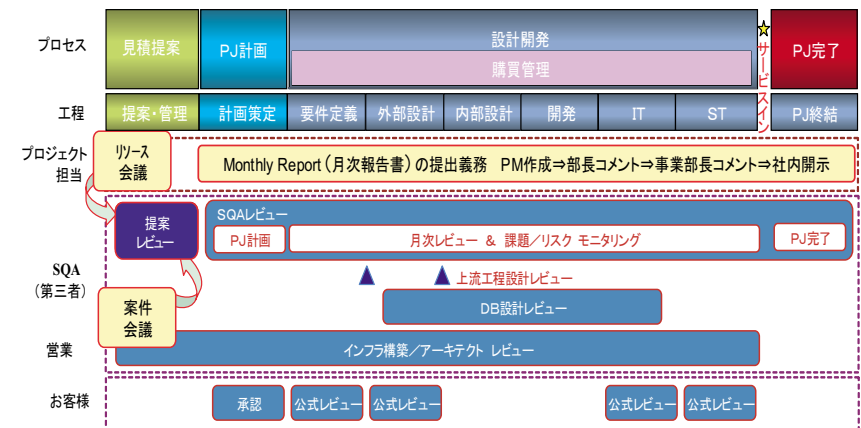


図3 TDIの品質保証活動

TDIでは、顧客満足を第一に考え、高品質の製品・サービスを将来にわたり提供するために「tdi-SPS」を構築しています。

「tdi-SPS」は、以下の5つのプロセス群から構成されています。

- ・見積り提案プロセス
- ・プロジェクト計画プロセス
- ・設計・開発プロセス

- ・プロジェクト完了プロセス
- ・購買管理プロセス

この「tdi-SPS」は、プロセス改善の専門部署であるコンピテンシー部により、現場で実用可能なプロセスとなるよう留意して作成されました。実際には、「〇〇手続き」という Word 文書と Excel によるテンプレートから構成されています。

各開発プロジェクトにおいては、開発部門を兼任している SEPG により、各プロジェクトが円滑に進むようサポートしています。また、プロセスの構築と改善活動の推進においては、「決まっているから実施する」ではなく、「今のやり方で問題はないか？」という問題意識を常にマネージャを含む関係者が持ち続けるといった雰囲気作りを心がけました。それが、「Preventive Management」というキーワードに現れており、またこのキーワードにより全社一丸となつての活動に発展することができたと思います。

以下に、各プロセス群の概要を記述します。

● 見積提案プロセス

顧客の要求事項を早い段階から全社横断的に検討してシステム化の目的を達成するために効果的な見積りを提案します。より良い提案に結び付けるための「案件会議」、タイムリーな全社リソース活用のための「リソース会議」、営業本部、開発本部および有識者により提案・見積り内容を評価する「提案レビュー」、技術専門家による「テクニカルレビュー」等により、提案品質を確保します。

● プロジェクト計画プロセス

システム化の目的、マスタースケジュールおよび製品品質確保のためのレビュー計画、テスト計画等は、プロジェクト計画としてとりまとめられ、SQA および有識者によるレビューを受けてプロジェクトが成功できる計画として立案されます。必要な場合は、このプロジェクト計画を顧客にも説明し、安心して依頼されるようにしています。

● 設計・開発プロセス

プロジェクト計画に従って、設計から顧客に納品するまでのプロセスを明確にして設計・開発します。また社内標準が適用されたプロジェクトは、工程別または月次で SQA および有識者のレビューを受けて、品質、納期、コスト等が計画通

りにできているかを確認します。

主な管理局面には、以下のものがあります。

(要件管理、課題管理、変更管理、リスク管理、構成管理、品質保証管理、進捗管理、プロジェクト管理)

● プロジェクト完了プロセス

プロジェクトが計画通りに実施され、顧客が満足できるような結果が出せたかを SQA および有識者でレビューして確認します。またプロジェクトから得られた各種実績データや教訓等は、次の見積提案やプロジェクト計画のためにソフトウェアプロセス資産として社内に蓄積され活用されます。

● 購買管理プロセス

TDI のプロジェクトにご協力いただく国内および海外のパートナー向けのプロセスと役割分担を明確にして双方ともにプロジェクトの成功を目指します。

2.2.3 計画レビューチェック項目と様式

品質保証活動の具体例として、TDI が重要視している SQA 計画レビューでのチェック項目の一例を以下に示します (表3、図4 および図5)。

これらのチェック項目に従い、SQA が中心となって、マネジメント同席のもと、状況を確認していきます。

表3 SQA 計画レビューでのチェック項目の例

計 画	チェック項目
①一般事項	<ul style="list-style-type: none"> ・プロジェクト名称、期間、作業場所、ソフトウェア・ハードウェア構成、ステークホルダ、プロジェクトの概要、主要要求事項等 (プロジェクト状況報告書)。 ・上記について説明を受け、プロジェクト側の課題認識を共有。 ・また性能要求、検収基準、納品物、品質基準、開発標準等の顧客要求に対応できる体制か否かを確認。
②見積り資料	<ul style="list-style-type: none"> ・見積り額とその根拠を確認 (提案レビュー時の資料で可)。 ・見積り額には、開発体制はもとより開発ツール、利用ソフトウェア製品、通信費、開発場所、オフショア活用の有無等のコストを含めて、見積り根拠の妥当性と採算性について確認。

(つづく)

(前頁より)

<p>③作業計画 (大日程計画、WBS) と成果物</p>	<ul style="list-style-type: none"> プロジェクトのスケジュール管理方法と報告ルールの説明、スケジュール変更等をする時の手順方法、WBS 詳細化の時期等について説明を受け、これらの妥当性を確認。 粒度：直近フェーズは、1週間単位までWBSで詳細化されていること。 記載内容：担当者、期間、工数、成果物が網羅されていること。 成果物：全工程の成果物、顧客提出の成果物と工程の完了基準等が明確であること。
<p>④開発体制</p>	<ul style="list-style-type: none"> プロジェクト状況報告書で説明を受けて、作業計画を実施する上で要員割当の妥当性、未充当を確認。 また再委託があれば当該企業の概要や契約内容も確認。
<p>⑤会議体と報告</p>	<ul style="list-style-type: none"> プロジェクトの会議体と報告ルールについて以下の説明を受け、確認。 <ul style="list-style-type: none"> 会議やレビューの種類、実施方法およびタイミング等の確認。 顧客のレビュー参加者の適切性の確認。 顧客の承認者の適切性の確認。
<p>⑥受入基準、検収、支払いについて</p>	<ul style="list-style-type: none"> 受入基準、承認者(顧客/TDI)、検収、支払いについての説明を受け、納品物、納期、検収時期、支払い時期等の妥当性を確認。
<p>⑦変更管理</p>	<ul style="list-style-type: none"> 要件、仕様等の変更管理手順と文書化(顧客の責任者との合意)、証跡について以下の説明を受け、確認。 変更要求の文書化は、定められた手続きをとられているか。 変更要求には、契約範囲内か否かをTDIから遅滞なく文書で回答をするように取り決められているか。 証跡の管理方法。
<p>⑧リスク管理</p>	<ul style="list-style-type: none"> プロジェクトが認識しているリスクと対策(回避策、緩和策、コンティンジェンシープラン)についてプロジェクト状況報告書で説明を受け、認識しているリスクの漏れや対策、コンティンジェンシープラン発動タイミングの妥当性について確認。 コンティンジェンシープランとは、リスクが現実化する事が確実になったときにプロジェクト目標を達成するために実施する代替プラン(施策、コスト、期間、納期)のことです。
<p>⑨品質管理</p>	<ul style="list-style-type: none"> 開発の各工程の成果物の品質保証に関する手順の説明を受け、工程の終了や成果物納品が妥当な手順になっているかを確認。 例：要件定義～システムテストのドキュメントのレビュー(つづく)

(前頁より)

<p>⑩コスト管理</p>	<p>(レビューア、承認者、等のルール)、単体テスト/結合テストのテストケース数、バグ検出数。</p> <ul style="list-style-type: none"> コスト管理方法について説明を受け、妥当性を確認。
---------------	--

tdiソフトウェアプロセス標準<計画レビュー資料>

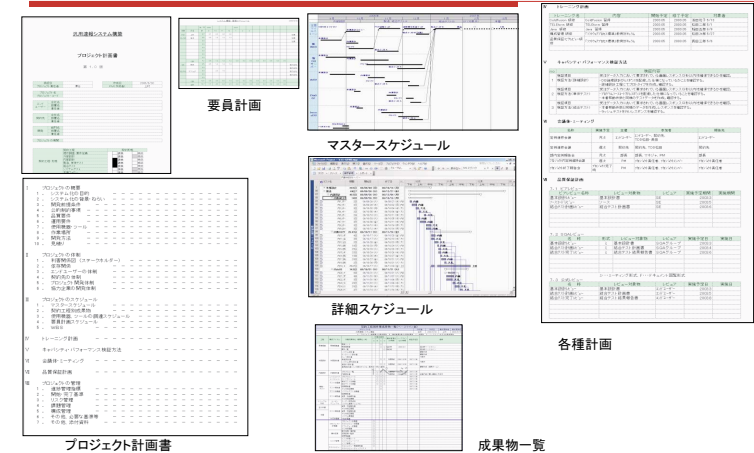


図4 チェック項目の一例と様式(1)

tdiソフトウェアプロセス標準<月次のレビュー資料>



図5 チェック項目の一例と様式(2)

2.2.4 「tdi-SPS」展開にあたって

TDIでは、「Preventive Management」というキーワードを掲げ、全社一丸となって活動に取り組んでいます。しかしその活動は決して容易なものではなく、推進者の多大な努力があってなされたものです。以下に、本活動を推進するにあたって、苦労した点を記述します。

(1) モデル（SW-CMM）の理解

手続き作成者もSW-CMMをゼロから学ぶ必要があったため、まずモデルを理解することに苦労しました。とくに、リスク管理、構成管理（ベースライン）、要件管理（トレーサビリティ）といった概念の理解に苦労しました。

(2) 「プロセス」の理解と定義

「プロセス」とよく耳にしますが、その概念を理解することに苦労しました。さらに、SW-CMMの記述は、英語の訳であるため、日本語としてその概念をよく理解できず、関係者での共通理解を得ることに苦労しました。

(3) 手順の文書化

SW-CMMで記述されているプラクティスを現場が行う手順として手続き文書を作成することに苦労しました。

モデルを解釈し、手順として文書化する活動は、当初の予想以上にスキルが必要な作業でした。

(4) 現場の理解を得ること

当初は、現場からの反発も少なからずありました。そこで、できる限り本活動の意義、目的を関係者全員に理解していただくよう、説明に努めました。手続き文書を作成した社員（5名）が講師を担当しました。1回あたり、10人から20人程度を対象にした2時間ほどの集合教育を企画し、延べ1,000人程度にむけて説明会を実施しました。

2.2.5 今後に向けた課題

① CMMIへの対応

現在の「tdi-SPS」は、SW-CMM L3の領域をベースに作成してあります。現在、CMMIを視野に入れた改善を計画中です。

② エンジニアリング領域の強化

現在の「tdi-SPS」は、プロジェクト管理面の強化を重点的に行ったものです。今後は、品質の向上に向けて、エンジニアリング領域、とくにCMMIのプロセス領域である要件開発、要件管理への対応が課題です。

③ 支援領域の強化

現在の「tdi-SPS」で対応したプロジェクト管理能力をより高めるために、「測定と分析」の強化を計画しています。

2.3 活用しているツール、技法

TDIでは、特別なツール等を使用せずに、現場レベルでの利便性を持たせるためExcel等を使用しています（表4）。

表4 TDIで利用しているツール

ツール	用途
TDIの基幹システム（MAGMA）	・社内開発したシステムで、売上／利益、リソース等を全社一元管理
TDIの社内ウェブサイト（Project Management Office）および共有サーバ	・「tdi-SPS」の手続きや様式およびプロセス資産等を全社で共有
各プロセスで使用する様式等	・「tdi-SPS」の手続きで指定または推奨されている「プロジェクト計画書」、「プロジェクト完了報告書」等は、大半はExcelで作成 ・マクロを活用しての入力の省力化
WBS、スケジュール	・市販パッケージのMicrosoft Projectやプロジェクトの特性に合わせてExcel等も活用

2.4 人材育成

TDIでは、人材の育成のため「人材開発センター」を設置し年間教育計画に従って育成を行っています。また毎年SQAレビューア育成コース等も開催し、レビュー品質向上にも取り組んでいます。さらに実践型エンジニア育成として社内リアルケースを用いた手作りの研修も企画し、プロジェクトマネージャ、アプリケーションエンジニア等を中心に早期育成に努めています。

3. 学ぶべき点

SW-CMMまたはCMMIのモデルを採用したプロセス改善は、多くの会社組織で行っていますが、トップの参加の管理面（マネジメント）等では、どのように取り組まれているのでしょうか。本事例は、「Preventive Management」というキーワードをもとにトップ参加の管理面（マネジメント）を重要視したプロセス改善に取り組み、成果が得られた事例として参考になります。

参考文献

能力成熟度モデルのキープラクティス 1.1版, 1999年

開発のためのCMMI 1.2版, 2007年

BEST
PRACTICE
事例7

派生開発による品質および開発効率の向上

～XDDP・USDMの徹底～

東京エレクトロンソフトウェア・テクノロジーズ株式会社
札幌テクノロジーセンター

東京エレクトロン・ソフトウェアテクノロジーズは、1991年に東京エレクトロンから分離した会社で、親会社ブランドの半導体製造装置、FPD製造装置、コンピュータシステム関連機器のソフトウェア・制御システムの開発を行っています。本社は東京ですが、今回の事例は九州、山梨、東北の製造拠点で製品を作り、札幌でソフトウェア開発を行っている分散開発事例です。ソフトウェア製品の開発は新規開発ではなく、適応保守が中心のソフトウェア保守であり、組込み系でよく派生開発と呼ばれているものです。

キーワード

・派生開発

「ソフトウェア保守」の定義を含み、社会や環境の変化に対応するための機能の追加・削除・移植等の仕様の変更によって新たな製品を生み出したり、既存のソースコードの流用を前提としたソフトウェア・システムの改造を伴う改変行為という（株）システムクリエイツ清水吉男氏考案定義。

・適応保守

JIS X 0161:2002 (ISO/IEC 14764) ソフトウェア保守での保守タイプ、他には是正保守、予防保守、完全化保守がある。

・ソフトウェア保守

JIS X 0161:2002 (ISO/IEC 14764)。

・XDDP

eXtreme Derivative Development Process、(株)システムクリエイツ清水吉男氏考案の派生開発専用のプロセス。

・USDM

Universal Specification Description Manner、(株)システムクリエイツ清水吉男氏考案の要求仕様を表現する方法。

従来は製造拠点からの修正依頼が来ると、ソースコードを直し始めていました。母体のドキュメントがないこともあり、修正依頼→コード調査→コード修正→テストというプロセスを実施していました。その結果、修正漏れが起因となった不具合が発生し、修正コードを納品してからも不具合対応に追われているという有様でした。このような状況を打破するために、XDDPを取り入れ、手戻りの少ないソフトウェア保守を行うプロセスを実現したのです。

XDDPを適用しても工数には変わりはないのですが、「開発への不安がなくなった」「仕様漏れに早く気付くので計画が崩れにくくなった」と現場の声は導入効果そのものであり、ソフトウェア保守の安心さを実現しました。

1. プロセス改善活動の背景

1.1 改善方針

改善方針は次のとおりです。

- ・できることから始める

1.2 達成目標

達成目標は次のとおりです。

- ・ソフトウェア品質向上、生産性向上

1.3 プロセス改善推進体制

TELSTにおけるプロセス改善推進体制は、開発グループの代表者で構成する委員会です。ここでは、ソフトウェア品質向上のための情報共有活動、改善プロセスの導入活動支援を行っています。プロセス改善推進委員会を図1に示します。

■東京エレクトロンソフトウェア・テクノロジーズ株式会社 札幌テクノロジーセンター
(本書で用いる略称：TELST)

●ビジネスゴール

- ・高品質のソフトを、素早く市場に提供する

●製品・サービスの概要

半導体製造装置 FPD製造装置、コンピュータシステム関連機器のソフトウェア・制御システムの開発

URL <http://www.tel.com/jpn/about/japan/telst.htm>

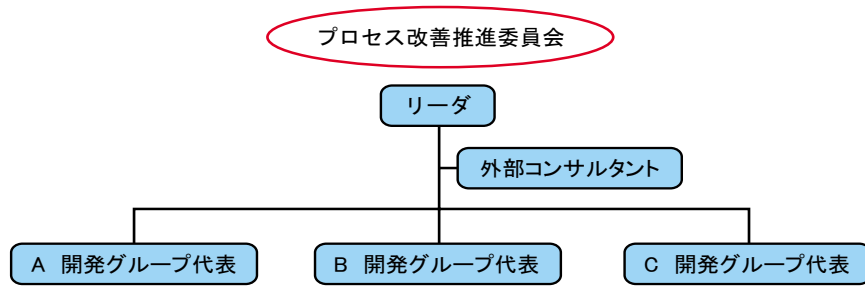


図1 推進体制図

TELSTにおけるUSD Mの導入は、牽引役のリーダーとUSD Mに関心を示した10人の開発者ではじめました。リーダーは外部コンサルタントとの窓口も行っています。そのため、開発者からの質問、取り組み結果の検証依頼がリーダーのところに集まるようになり、自動的に情報が集約されます。

外部コンサルタントの役割は、開発者が作成した「変更要求仕様書」のデザインレビューであり、要求、仕様の粒度、仕様の階層化等に関する問題指摘を行っています。

1.4 プロセス改善対象組織の規模

・2年間でTELSTの70%の開発部門に展開（図2）。

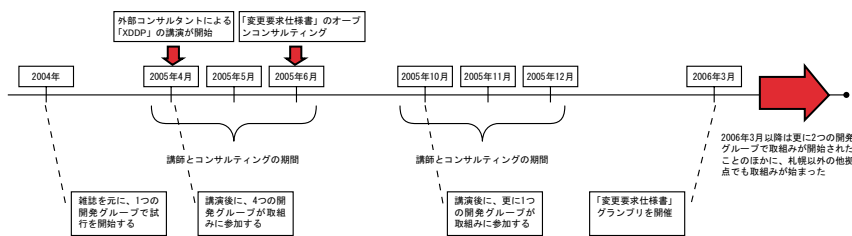


図2 社内での派生モデル開発の展開の様子

1.5 改善活動の経緯

ここで紹介するプロセス改善活動に取り組む前は、仕様の取り違い、仕様漏れ

がプロジェクトの後半になってから見つかり、計画通りの作業が困難になることもありました。

2004年秋には、グループ会社に向けて、ソフトウェアの要件開発に関する外部コンサルタントを招き講演会を開きました。外部コンサルタントの提唱する派生開発向けの要件定義手法は、それまでも雑誌記事等でも取り上げられており、プロセス改善推進委員会メンバがもともと関心を持っていたことも相まって、講演内容の採用による改善効果が期待できるとの結論になり、プロセス改善推進委員会を中心となって、USD M手法の導入開始となりました。

1.6 主要な課題と改善のフォーカス

主要な課題と改善のフォーカスを表1に示します。

表1 課題と改善のフォーカス

課題	改善のフォーカス
分散開発でのソフトウェア保守を行う際の情報不足	課題を補うコミュニケーション手段
仕様が複雑になっているなかでの追加・変更による修正漏れの発生	修正漏れを防止

1.7 顕在化した改善効果

表2に改善効果（定性）を、表3に改善効果（定量）を示します。

表2 改善効果（定性）

項目	改善効果
要求仕様合意後の手戻り	減少
仕様漏れと分類されていた不具合	減少
計画遅延の機会	減少
要求仕様の明確化に伴い、プログラム変更仕様設計とテスト設計の並行進行	可能になり、納期短縮

表3 改善効果（定量）

項目	改善効果
保守工数と不具合修正工数（手戻り工数）の比率	適用前は6：4 適用後は8：2 現状は9：1 (全体の工数は変わっていない)
工程内工数比率	適用前：要求定義：1、設計：2、実装：5、テスト：2 適用後：要求定義：4、設計：2、実装：1、テスト：3

要求定義を確実に実施し、十分なテストを行うことで、不具合の削減を実現しました。

2. ベストプラクティス

2.1 参照モデル上でのカバレッジ

- ① 当該ベストプラクティスに対応する ISO/IEC 15504 第5部のプロセス
 - ・ソフトウェア要求事項分析 (ENG.4 Software requirements analysis)
 - ・ソフトウェア設計 (ENG.5 Software design)
 - ・ソフトウェア結合 (ENG.7 Software integration)
- ② 当該ベストプラクティスに対応する CMMI プロセス
 - ・要件開発 (RD: Requirements Development)
 - ・技術解 (TS: Technical Solution)
 - ・成果物統合 (PI: Product Integration)
- ③ 当該ベストプラクティスに対応する共通フレーム2007のプロセス/アクティビティ
 - ・開発プロセス/ソフトウェア要件定義
 - ・開発プロセス/ソフトウェア方式設計
 - ・開発プロセス/ソフトウェア結合
 - ・開発プロセス/ソフトウェア適格性確認テスト
 - ・保守プロセス

2.2 プラクティスの説明

2.2.1 プロセス改善への取り組みの背景

TELSTでは、親会社ブランドの半導体製造装置、FPD製造装置、コンピュータシステム関連機器のソフトウェア・制御システムの開発をしています。ミッションは、工場にいち早く高品質な製品ソフトウェアを提供し、工場から顧客に製品を提供できるようにすることです。開発体制としては、ハードウェア製造とソフトウェア製造の両方を行っている拠点が、山梨、九州、東北であり、今回の事例の組織である札幌は、ソフトウェア製造のみとなっており、上記の各拠点からソフトウェア製造依頼があります。東京は、ソフトウェア製造のみですが、基礎開発的なことを行っています。このようにハードウェアとソフトウェア開発の拠点が分かれた製品があり、ソフトウェアも拠点間での分散開発となっているため、情報共有の不足、コミュニケーションの不足が起因となり仕様の取り違えが起きることがありました。USDMに取り組んだ背景には、分散開発によるコミュニケーション不足が起因となった失敗を少なくし、はじめから完成度の高いソフトウェアを納めることにより、納品後の手戻りの心配をすることなく、次の開発に全力で取り組める環境を作ることになりました。

2.2.2 派生モデル開発への取り組みの背景

TELSTでは、要求仕様に時間をかけず、「まずは動くものを作る」の精神で作業を進めていました。そのため仕様の抽出が不十分であり、設計またはコーディングをしながら仕様を詰めていくという作業をしていました。その結果、「テスト工程で要求の取り違い」や「仕様漏れが見つかって」、「計画が崩れる」ことがあり、開発メンバ自身が作業を計画通りに進める自信をなくしていました。

この状況を打開するために、2004年にUSDM（「2.2.3 USDMとは」を参照）を用いた要求の仕様化に取り組みました。USDMはXDDPのなかで紹介された要求仕様を表現する方法で、派生開発の場合は「追加機能の要求仕様書」や「変更要求仕様書」として活用します。「変更要求仕様書」は、変更の特化した要求仕様書の記述方法を用いているので、開発メンバの理解した変更要求の範囲を表現しやすくし、仕様の漏れが発見しやすくなる工夫がされています。

2.2.3 USDM とは

USDMはUniversal Specification Description Mannerの略で要求仕様を表現する方法です。新規開発はもとより、追加機能の要求、仕様変更の要求を主とする派生開発でも使える表現方法です。

その特徴は、以下のとおりです。

① 要求を「振る舞い」で表現する

要求とは、「実現して欲しいこと」。実現して欲しいことを操作やソフトの動きがイメージできるように「振る舞い」で表現することで、要求提供者の要求を開発者が理解しやすくなります。

② 「理由」を付けて要求補強する

要求には「理由」や「背景」があります。

要求と理由を一緒に表現することで、要求の「履き違い」を押さえることができます。

③ 4つの分割基準を提供し、要求を分割・階層化して要求の範囲を制御する

要求を実現する仕様を抽出しやすくするように要求の分割・階層化を行います。

要求を分割する際には、次の4つの基準を用います。

- ・「時系列（時間軸に着目）」
- ・「構成（機能や構成）」
- ・「状態（状態という概念で要求の階層化）」
- ・「共通（共通する機能を切り出し独立させる）」

といった基準で分割します。要求を分割することで、実現する範囲がしぼられるため、仕様抽出がしやすくなります。

④ 要求と仕様を階層構造で表現する

4つの基準で分割された要求の下に仕様を展開します。仕様は要求を実現するうえでの具体的な記述です。具体的な記述とは、仕様からソースコードの変換がイメージできる記述としています。

⑤ 範囲が制御された要求の中で仕様を抽出する

1つの要求が多数の仕様で実現される場合があります。仕様の数が多い場合は、それらをいくつかのグループに分けて分類します。グループは、グループをあら

わす記号(< >)を使って適当なグループ名（グループ名は名詞形）を挟んで表現します。

●要求仕様書の特徴部の例

要求を分割・階層化して仕様を抽出することで、実現して欲しい「要求仕様」を関係者が認識することができるようになります。USDMで詳細に表現された要求仕様書で合意がなされれば、それ以降は、安定した開発を進めることができます。

以上の手順に沿って作成した要求仕様書の特徴部の例を図3に記します。

顧客による座席予約機能	要求	理由	説明	備考欄
カテゴリー名(記号)	RSY		操作画面から、指定された席会議室を予約する。	振る舞い(動き)がわかるように動詞で表現する。
			お客様が希望する人数に応じた会場を操作画面を介して予約したい。	理由を書くことで、要求の履き違いをなくす。
			販売は予約システムに接続していません。ここではネットとの接続方法については扱わない。複数の席から同時に同じ会議室を予約する可能性がある。	
	要求	理由	RSY.01 会員番号の入力を求める 説明 この操作は、既に会員登録された人にも提供されるもの	
			<会員番号の入力>	
			<input type="checkbox"/> RSY.01.1 会員番号の入力を求める	
			<input type="checkbox"/> RSY.01.2 入力された会員番号が登録されていることを確認する	
			<input type="checkbox"/> RSY.01.3 会員登録されていない場合は「その番号は登録されていません」を2秒間表示し、元の画面に戻る	
			<開始画面の表示>	
			<input type="checkbox"/> RSY.01.8 会員であることを確認できれば予約操作が出来る画面を表示する	
	要求	理由	RSY.02 開始画面から会議室を選択することで、予約状況を知ることができる	
		理由	予約を急ぐかどうかの判断の材料を提供する	
		説明	予約の対象とする会議室は指定のファイルに登録されている。予約状態が随時更新されている。	
			<会議室一覧の表示>	
			<input type="checkbox"/> RSY.02.1 「一覧」ボタンで会議室一覧画面を表示する	
			<input type="checkbox"/> RSY.02.2 会議室ファイルを開いて、会議室一覧をファイルに登録されている順に表示する	
			<予約状況の表示>	
			<input type="checkbox"/> RSY.02.7 会議室番号をキーにして、予約ファイルから予約状況を入力して画面に表示する	
			<input type="checkbox"/> RSY.02.8 予約状況の表示は、一覧表示されている画面の上から順に表示する	
			<input type="checkbox"/> RSY.02.9 表示の種類は、 ・予約済み ・空き の2種類	
			<input type="checkbox"/> RSY.02.10 予約状況表示のタイミングは、 1) 一覧画面を最初に表示したとき 2) 「更新」ボタンが押されたとき	
			<会議室一覧の表示、(予約状況の表示)と操作と画面の挙動がイメージできるように時系列で表す。>	
			<input type="checkbox"/> RSY.02.10	
				グループ化することで考える範囲が特定されて、仕様の抽出がしやすくなる。

図3 要求仕様書の特徴部の例

2.2.4 USDM による要求定義以降の開発効率の改善

USDMを取り入れてから、最初のうちは、変更要求仕様書の完成までに全体開発工数の約4割を投入していましたが、USDMでの表現方法が組織で安定すると、変更要求仕様書の作成時間は短くなり、全体開発工数の約3割に落ち着きました。変更要求仕様書作成作業の手順は次のとおりです。

① 要求に対する実現方法（仕様）を検討する

- ② 既存資産（ソースコード、ドキュメント）を確認し、変更箇所を洗い出す
- ③ 変更要求仕様書の構成を検討する
- ④ 変更要求仕様書を関係者間でレビューする

ここで注目したい箇所は、「② 既存資産の確認作業」です。XDDPではソースコードの変更箇所＝仕様変更と捉え、USDMの表記方法にしたがって、変更要求仕様書上で仕様の分類と整理を行います。その後に設計～実装作業に移るため、作業を確実にこなせるようになっていきます。導入以前は、極端に言えば、手当たり次第にソースを直し、テストしてから漏れが発見され、そして前に戻るという作業を行っていました。

XDDPとUSDMを取り入れる前は、ソースコード上で該当箇所を見つけしだい修正し、動作を確認するといったことを行っていたため、変更箇所を記録した資料が残っていませんでした。また、コーディング終了後にその資料を作成するといったことをしていましたが、この作業は効率的ではなく、記録上の漏れも起きやすく、再利用されることも少ないものでした。

USDMを取り入れてから、開発効率の面で改善されたことは、

- ・設計～実装段階で“仕様漏れ（検討漏れ）”が減少
- ・テスト段階で“仕様漏れ（検討漏れ）”が減少
- “仕様漏れ（検討漏れ）”が原因の手戻りが少なり、開発効率が改善
- ・USDMで表現した変更要求仕様書には具体的な変更内容が書かれているため、変更点を狙い撃ちしたテストケースが考えやすい

2.2.5 USDMによる品質の向上

TELSTでの製品の品質の決定は「知見者が参加するレビュー」と「最終工程のテスト」が大きな割合を占めています。

下記のようにUSDMによって、品質は向上しており、その品質は開発メンバに自信を与えています。

（1）レビュープロセスの改善とその効果

- ① 関係者を交えたレビューが機能する

レビューは変更要求仕様書が書かれたところで、作成者、知見者、依頼元との間で行います。

知見者は変更要求仕様書を通して“仕様漏れ（検討漏れ）”を検出します。

依頼元は依頼した内容が正しく伝わったことを要求と理由から確認します。また、時系列で書かれた要求、仕様からは動きを確認することができます。

② レビューによる欠陥検出が向上

USDMで書かれた変更要求仕様書は仕様漏れが発見しやすい表現方法なので、レビューで欠陥検出ができます。また、変更要求仕様書はテストケース作成時の入力文書としても使っているので、テストケース作成時に仕様の誤り、仕様漏れに気が付くため、テスト工程に進む前に欠陥を除去することができます。

（2）テストプロセスの改善とその効果

① テスト段階での仕様漏れ不具合が減少

テストケースのレビューで仕様漏れを防ぐことができるので、スケジュールに影響を与える作り込み不足はほぼ発生しません。

② テストケースが作りやすくなる

変更要求仕様書には変更点がBefore/Afterの表現で書かれているため、変更点を狙ったテストケースが作りやすくなります。また、変更要求仕様書には変更点以外のことは扱いません。ここに書かれていないことの確認をテストケースで表現し、これを作成者と確認することで“仕様漏れ（検討漏れ）”を防ぐことができます。

③ 最終テストの目的が変化

最終テストは不具合を発見する工程から、期待する動作を確認する工程に変化しています。

2.2.6 変更要求仕様書の組織への展開

組織への展開は次のように進めました。

はじめにUSDMの考案者である外部コンサルタントを招き、USDMの話をして頂き、開発メンバ全員の知識と意識を合わせました。

各開発グループはUSDМを取り入れ、その取り組み結果を直接外部コンサルタントに指導していただきました。

指導方法は、

① オープンコンサルティング

これは、個人または、開発チームと外部コンサルタントとのやりとりの様子を聴衆にみせる方法です。開発者の工夫と外部コンサルタントからのアドバイスを聴衆側も直接確認できるよさがあります。また、約40名の聴衆を前にコンサルティングが行われるので、発表者には発表の訓練にもなります。

② 夕食会を利用したコンサルティング

食事をしながら、コンサルティングをする方法です。食事会の参加者は10名程度、オープンコンサルティングに比べて少人数なのと、食事で気持ちが和らいでいることで参加者からは質問しやすいと好評です。

③ メールを介したコンサルティング

電子メールは担当窓口を介して行います。担当窓口を介すことで、取り組む組織が把握でき、質問事項が整理できるので、先に挙げた2つのコンサルティングの対象を担当窓口がメールでの質問内容・量を観ながら、適切に選ぶことができました。

上記のイベントは、USDМ導入当初は1～2ヵ月おきに開催していました。定着した現在は、半期に一度程度で行っています。

2.2.7 USDМによって得られた効果

ベースとなるソフトウェアに対する追加、変更を主とする派生開発を担当する開発グループにおいて、USDМを使った変更要求仕様書は、「階層化された要求に対して<グループ>で範囲を特定した中で仕様を展開する」、「変更されることがだけを記述する」、「ソースコード=仕様ととらえる」をルールに仕様を表現するため、開発者がどこに変更を加えて、要求を実現しようとしているかを理解するのに適したものでした。

USDМを取り入れた効果として、開発メンバから次のような意見が上がっています。

- ・各開発工程での手戻りが減った
- ・計画通りに作業を進めることができるようになった
- ・変更要求仕様書があれば、作業途中からの応援参加がしやすい
- ・テスト担当、設計担当を分けて並行開発が可能となった
- ・仕様の取り違え（未検討を含む）による不具合はほとんどなくなった
- ・出荷後の問い合わせが減った
- ・安心して開発ができるようになった

USDМで表現した変更要求仕様書は組織に定着しています。派生開発のため、性質の似た機能要求が多くあります。性質の似た機能間で仕様漏れがおきにくいように、要求の階層化、仕様のグループ化をパターン化して、これを「テンプレート」として活用する工夫も行われています。各グループの工夫を開発メンバで確認できるように、年に一度、「要求仕様書グランプリ」と題して、参加者全員で講評し合っています。

2.2.8 組織文化

TELSTの組織文化は現場主導での実施です。各開発グループが改善したいことをそれぞれで実施します。各開発グループの改善活動は、開発グループの代表で構成されるプロセス改善推進委員会で紹介されるので、改善活動を開発グループ間で共有することができます。

共有したことは、開発グループの代表を通じて各開発グループに展開されます。プロセス改善推進委員会に参加する代表者は開発経験が2～5年の若い社員です。若い社員ほど、既存の開発スタイルにとらわれず、改善意欲が高いことも組織文化であります。また、トップの品質への関心も高く、半期に一度、社員全員が集まる全体会議では必ず品質に関わるテーマが扱われています。

2.3 活用しているツール、技法

USDМでの要求仕様の階層化はExcelを使って表現されるため、Microsoft Office環境があれば、誰でも試すことができます。

2.4 人材育成

TELSTでは、XDDPとUSDMで表現した要求仕様書を普及展開するための教育を実施し、講師から要求仕様書の説明会を受けています。受講できなかったメンバには、ビデオ教材を提供し、自習できる環境を用意しています。

また、開発者が書いた要求仕様書を講師がレビューし、コメントをフィードバックして、要求仕様書の作成能力を高める機会も提供しています。

3. 学ぶべき点

仕様漏れ、仕様の取り違いに気が付きやすい表現で要求仕様書を作成することで、高品質のソフトウェアを開発できるようになった事例です。要求に対する理由付け、要求仕様の階層化を通して開発メンバが自分の思考を整理しながら開発を進める姿勢に変化したことも注目すべきところです。

参考文献

清水吉男：要求を仕様化する技術・表現する技術－入門＋実践 仕様が書けていますか？，技術評論社，2005年

清水吉男：「派生開発」を成功させるプロセス改善の技術と極意，技術評論社，2007年

BEST
PRACTICE
事例8

ソフトウェア部品による組立型開発と品質保証

～基幹システムでのコードの再利用～

住友電気工業株式会社

情報システム部

&

住友電工情報システム株式会社

システムソリューション事業本部

住友電気工業株式会社、住友電工情報システム株式会社ではエンタープライズ系システム開発の工業化を目指して、(1) 属人性を排除した設計手法の確立、(2) 部品組立型システム開発、(3) 統計的手法を使った品質管理を実践しています。

属人性を排除した設計手法は、T字形ER手法の導入により実現されています。T字形ER手法は少ないルールで作図でき、作成されたデータベース設計に差異が少ないという特徴があります。仕様の不明確さや、不整合があると作図できないため、仕様書の品質が一定以上であることを保証します。属人性を排除することによりSEの経験やスキルに左右されない設計が可能となり、手戻りによるコストが削減されました。

キーワード

・組立型開発

住友電工(株)情報システム部、および住友電工情報システム(株)で実施されているソフトウェア部品化/再利用+コードジェネレーションを活用したソフトウェア開発

・定量的プロジェクト管理

・統計的品質管理

・JaX

プログラムの規模を測定する指数

・組立型プログラミング

・サイクロマティック複雑度

if、while、for等のプログラムの流れに基づいてプログラムがどれだけ複雑かを計測する指標。一般に管理上限値は20とされています。Javaプログラムの場合、自動計測を行うEclipseのプラグインがオープンソースで公開されている

部品組立型システム開発は、業務に特化した部分の再利用率を高め、生産性と保守性を向上させることを目標として取り組まれたもので、受注番号や受注日等のデータ項目や、画面、帳票等を含む約2,000の部品が蓄積されています。また、適切な部品を選択し、利用できるようにするため、トレーニングコースの整備、ネーミングルールの標準化、部品や画面の自動選択ツールの開発を行っています。

統計的手法を使った品質管理は、部品組立型システム開発に対応した規模指数の導入、u管理図による欠陥数の管理、プログラム中のif文の数を用いた欠陥数の予測により実現しています。また、一連の工程での欠陥の作り込み数、除去数をグラフ化することで開発チームの品質管理能力を明示しています。統計的品質管理に用いる管理図等の各種グラフは自社開発の構成管理ツールにより自動生成されるため、開発チームの手を煩わせることなく状況を容易に確認できます。

品質向上を議題とした全社大会の定期開催、トレーニングの構築と実施、品質管理部門による管理支援等も行われており、3つの取り組みを含めて、定着、改善が図られています。

1. プロセス改善活動の背景

1.1 改善方針

SEI/SISでは、システム開発の工業化をキーワードとして以下の3つを柱として改善を進めています。

- ・属人性を排除する設計手法の導入

■住友電気工業株式会社 (Sumitomo Electric Ind., LTD) 情報システム部

(本書で用いる略称: SEI)

■住友電工情報システム株式会社 (Sumitomo Electric Information Systems Co., Ltd) システムソリューション事業本部

(本書で用いる略称: 略称 SIS)

●ビジネスゴール

- ・統計的品質管理手法による継続的品質改善
- ・年率 5~10% の継続的開発コスト削減

●製品・サービスの概要

- ・基幹系システムの開発
- ・パッケージ製品の開発、販売
- 楽々 Framework II (基幹システム開発用ソフトウェア部品、開発環境)
- 楽々 Workflow II (楽々 Framework II と連携するワークフロー・ソフト)
- 楽々 Procurement(Web 購買システム)
- QuickSolution (全文検索システム)

URL <http://www.sei.co.jp/>

URL <http://www.sei-info.co.jp/>

- ・部品組立型のシステム開発
- ・プロセス改善による QCD の向上

1.2 達成目標

達成目標は、次の3点です。

- ・データ中心設計による20%以上の工数削減
- ・再利用部品開発による開発コストの半減（対 COBOL 比）
- ・定量的品質管理による出荷後欠陥数の半減

1.3 プロセス改善推進体制

SEI は、20年以上前から新技術の評価し導入する部署が設置されており、新しい開発手法や開発ツールの導入を伴う革新的なプロセス改善を推進しています。

SIS は、主としてビジネスシステムの設計・開発・運用を担当し、2003年には CMM レベル3を達成、漸進的なプロセス改善を推進しています。

2007年からは統計的手法を使った定量管理を実現するために SEI、SIS の両社が連携し、プロセス改善を推進しています。

① 新技術（ツール、手法）の評価・導入の担当

- ・SEI システム技術グループ

② プロセス改善の担当

- ・SIS

プロセス改善推進委員会（委員：社長、部長、課長および SEPG）

品質管理推進室

プログラム改善ワーキング・グループ（WG は必要に応じて結成）

1.4 プロセス改善対象組織の規模

- ・開発者約200名（プロジェクト管理者、設計者、開発者等）

1.5 改善活動の経緯

SEI では約20年以上前から技術の評価・導入のための組織を設置し、常に社内開発チームのニーズや世間の動向を見ながら技術的な改善活動を続けています。2006年からはシステム開発の工業化に向け、CMMI レベル4、5達成を目指して、統計的手法を使った開発手法の開発に取り組んでいます。

SEI/SIS ではシステム開発の工業化を実現するため、「属人性を排除した設計手法」、「バイナリ部品の再利用」、「統計的品質管理」の3つの観点から継続的に改善活動を行っています。表1に、これまでの取り組みを示します。

表1 SEI/SIS の主な取り組み

年	取り組み内容
1989年	プログラム開発の生産性向上を目指してソースコード・ジェネレータを自社開発
1991年	ソースコード・ジェネレータを全社展開
1994年	生産性向上ワーキング・グループ活動結成 設計工程を含む開発生産性向上を目指してデータ中心設計を導入
1999年	組立型開発の実現を目指して Java によるフレームワークを開発
2003年	ソフトウェア部品が充実し、組立型開発を開始
2003年	CMM レベル3を達成
2006年	品質大会開始
2007年	品質向上を目指して統計的手法を使った品質管理を実施

1.6 主要な課題と改善のフォーカス

- ・属人性を排除した設計手法の導入
- ・ソフトウェア部品の開発と利用技術の普及
- ・属人性を排除した品質コントロール手法の確立と普及

1.7 顕在化した改善効果

SEI/SIS における改善効果を表2に示します。

表2 改善効果

項目	改善内容
データ中心設計によるシステム開発コスト	30%削減
組立型開発導入	COBOL 比で約3倍の生産性向上
統計的手法を使った品質管理の導入	出荷後欠陥数半減

2. ベストプラクティス

2.1 参照モデル上でのカバレッジ

- ① 当該ベストプラクティスに対応する ISO/IEC 15504 第5部のプロセス
 - ・品質管理 (MAN.4 Quality management)
 - ・測定 (MAN.6 Measurement)
 - ・資産管理 (REU.1 Asset management)
 - ・再利用計画管理 (REU.2 Reuse program management)
 - ・知識管理 (RIN.3 Knowledge management)
 - ・ドメインエンジニアリング (REU.3)
- ② 当該ベストプラクティスに対応する CMMI のプロセス領域
 - ・測定と分析 (MA: Measurement and Analysis)
 - ・技術解 (TS: Technical Solution)
 - ・組織プロセス定義 (OPD: Organizational Process Definition)
 - ・定量的プロジェクト管理 (QPM: Quantitative Project Management)
 - ・組織プロセス実績 (OPP: Organizational Process Performance)
- ③ 当該ベストプラクティスに対応する共通フレーム2007のプロセス/アクティビティ
 - ・開発プロセス/ソフトウェア要件定義
 - ・開発プロセス/ソフトウェア方式設計
 - ・再利用プログラム管理プロセス
 - ・ドメイン技術プロセス

2.2 プラクティスの説明

2.2.1 ソースコード・ジェネレータ技術の確立

一般的にホストコンピュータ上でのプログラム開発は抽出、ソート、マージ、帳票出力等に分類されたプログラムの雛形に必要なコードを追加することが主でした。SEI/SIS では、オープンシステムの開発でもこの考え方を取り入れました。当時は Informix-4 GL で開発をしており、telnet ベースの画面が中心でした。1本のプログラムで照会・登録・変更・削除等の機能があり、COBOL 利用時の雛形よりも、より高度な雛形が必要でした。雛形を使用する際のコピー後のプログラム修正工数を削減するために、1989年にソースコード・ジェネレータを自社開発し、雛形コピー時にはソースコード・ジェネレータにパラメータを入力するようにし、ソースコードに必要なパラメータが自動的に埋め込まれるように工夫しました。この機能によりプログラマは、使用する雛形の名前、受注番号や受注日といったプログラム固有のデータ項目を指定するだけで、とりあえずは動作するプログラムの自動生成が可能でした。そしてプログラマは自動生成されたソースコードに必要なビジネスロジックを追加し、プログラムを完成させます。ソースコード・ジェネレータが使用する雛形は簡単に追加・変更することができたため、プログラムの自動生成が幅広くできるようになり、1991年にはソースコード・ジェネレータを使ったプログラム開発が SEI/SIS 全社に定着しました。この取り組みにより以下の技術基盤が確立し、ソフトウェア部品開発の基礎となりました。

- ・雛形の開発技術すなわち粒度の大きいソフトウェア部品の抽出技術の確立
- ・雛形利用を前提にした設計の定着
- ・プログラムの構造やコーディングスタイルは自動生成されたソースコードを真似てプログラムを開発するため、プログラム構造等の標準化の進展

2.2.2 データ中心設計の導入と定着化

実装レベルでのソフトウェア部品を再利用するためには、システムのアーキテクチャや設計思想を統一する必要があります。ところがソフトウェア設計は非常に自由度が高いものであるため、SE の経験やスキルによってさまざまに設計され、結果的に、どの設計が正解であるかという客観的な判断基準を示すことができな

いという課題があります。

1994年に結成した生産性向上ワーキンググループ活動を通じて、データベース設計の品質がプログラム開発の生産性に大きな影響を与えることが判明し、データベース設計手法の検討に着手しました。調査・検討の結果、属人性を排除しやすいT字形ER手法を採用し、そのサンプルとして図1にT字形ER図を示します。T字形ER手法は少ないルールで作図できるため、教育が比較的容易です。また、正解が1つであるためレビュワーによっての指導内容が差異が少なく、設計品質を一定に保ちやすくなります。そして大阪、東京、横浜、名古屋等の各開発拠点でT字形ER手法の講習会を実施した後、実プロジェクトでの個別指導を実施した結果、約2年間でデータ中心設計手法がSEI/SISに定着しました。これにより設計思想が統一され、ソフトウェア部品の前提になるデータベース構造が安定し、開発した部品が多くプロジェクトで利用できる基盤が確立されました。

T字形ER図は、仕様が不明確であったり、仕様間に不整合が生じていたりすると作図（エンティティ間のリレーションシップの線の描画）ができません。そのためT字形ER図が完成した際には、外部仕様書の品質が一定以上になっていることを示していることになります。SEI/SISではT字形ER手法の導入により、外部設計から統合テストのそれぞれのフェーズで約30%のコスト削減が実現できました。外部設計を原因とする手戻りの減少が、コスト削減につながった例です。

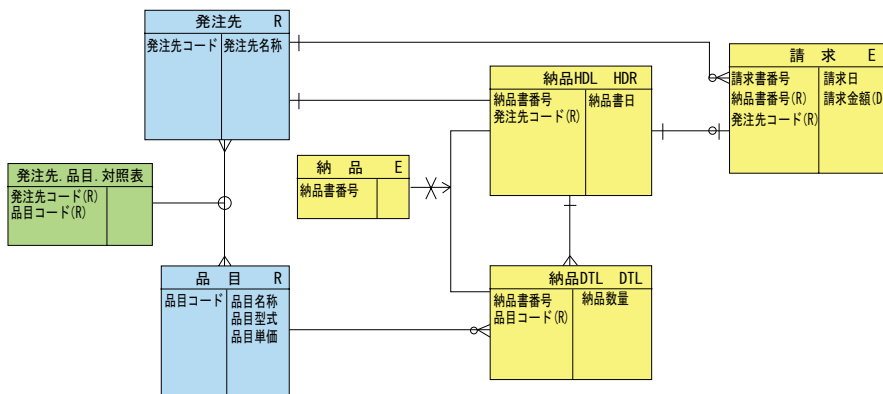


図1 T字形ER図のサンプル

2.2.3 組立型開発への取り組み方針

ソースコード・ジェネレータ技術によってプログラム開発の生産性は約30%改善することができましたが、継続的に生産性を向上させるためにはソフトウェア部品による組立型開発が不可欠でした。ジェネレータ方式では、次の2つの欠点がありました。

- ・基本的にソースコードのコピーによる再利用であり、保守対象のコード量が増加
- ・改善により自動生成率は向上するものの、工数の削減効果のあるビジネスロジックの自動生成が難しいため、工数効果は約30%で頭打ち

組立型開発ではこれらの欠点を克服するため、次の方針で進めました。

- ・ソフトウェア部品はバイナリで提供し、保守性の向上を図る
- ・システム全体の8割程度が再利用できるよう、システムよりの部品から業務部品まで広範囲にわたっての部品提供の実施
- ・部品の蓄積と部品の利用技術の改善により、継続的に生産性を向上できるようにする

2.2.4 ソフトウェア部品の開発

1999年、サーバ側で動作するJava Servletによる基幹系システム構築事例が雑誌に掲載され、それに触発されたSEI/SISではJavaによる組立型開発を目指したソフトウェア部品の開発を開始しました。基幹システムのプログラムには日付チェック部品等のシステムよりのソースコードと、受注番号、受注日といった適用業務に特化したソースコードが含まれていますが、従来の部品化技術では業務に特化した部分の部品化は困難で、部品化の割合はせいぜい2～3割程度でした。そのため、組立型開発実現のためには業務に特化した部分のソースコードを再利用できるようにすることが必要不可欠です。SEI/SISではソースコード・ジェネレータ技術でプログラムを分割するノウハウがそれまでに蓄積されていたため、受注番号、受注日といったデータ項目をプログラムのメインルーチンから分離した雛形が多数あり、また、前述のとおり、データベース設計手法が本社レベルで標準化されており、再利用の環境が整っていたのです。

一般的には、オブジェクト指向言語を使ったシステム開発ではクラス図を使った設計が行われます。クラス図を使った設計ではシステム内の共通的な処理を1箇所にまとめることができるため、システム内の再利用を進めることができます。しかし、このアプローチは異なるシステム間での再利用が難しく、再利用できるのは、ログ管理といった汎用機能の部品に限定されます。

このため SEI/SIS では、再利用が難しい具体的なクラス（例えば、受注クラス）の実装は行わず、受注クラスや出荷クラスを汎化したスーパークラスを実装しています。部品化するのはエンティティ・クラスであり、受注や出荷はメモリ中に存在するソースコードを持たないオブジェクトとして扱っています。同様に受注番号や受注日等のデータ項目もデータ項目クラスのオブジェクトとして扱っています。受注画面や受注リスト等も画面クラスや帳票クラスのオブジェクトとして扱えば、簡単に再利用できるようになります。受注番号や受注日の属性（桁数、データ型、データラベル、入力方法、エラーチェック等）はリポジトリと呼ばれるメタデータのデータベースに格納しておき、オブジェクトをメモリ中に生成する際、リポジトリから属性を読み込むことで、業務に対応したプログラムがメモリ中で動作できるようになります。受注画面もメモリ中のオブジェクトとして実装され、画面の入力項目、表示項目、対象となるデータベースのテーブル名等をリポジトリから読み込むことで業務画面の表示が可能となります。

このように高度に抽象化された部品に対してメタデータを与えてやることで販売管理や生産管理といった適用業務とは関係なく広範囲のプログラムで再利用が可能になります。これらのソフトウェア部品群は体系化し、楽々 Framework II と名付け、全社展開しています。2007年には、合計約2,000のソフトウェア部品が登録され、このうち、約500がメインルーチンに相当するパターンと呼ばれる部品です。

● 楽々 Framework II

図2に楽々 Framework II で開発したシステムの内部構造を示します。ブラウザから画面表示の要求があると Servlet が起動し、要求されたプログラムのメタ情報をリポジトリ (XPD:XML Program Definition) から読み出します。このメタ情報にはメインルーチンとなる部品の名前と画面やデータベースで使用する項目名等が含まれています。Servlet はメインルーチンになる部品に項目名等のメタ情報を

引き渡し、図中に Pattern と示しているメインルーチンを起動します。データ項目の各種属性は、DD (Data Dictionary) をオブジェクト化した項目オブジェクトがメモリ中に格納されています。メインルーチンは必要に応じて項目オブジェクトを呼び出すことで設定された桁数で画面表示を行うことができます。メインルーチンや項目オブジェクトには、あらかじめ用意したインターフェースを使ってロジックを追加することができます。この追加したロジックのプログラムをプラグインと呼んでいます。プログラムはプラグインを開発することで、例えば金額がマイナスのデータを赤色で表示するプログラムを作成することができます。システム全体で複数のプログラムから呼び出されるロジックは事前に開発しておきプラグインを通じて呼び出すことができます。例えば、在庫管理のロジックは入庫と出庫のデータを扱うため、図中の Business Logic で示される部分に共用部品として実装されるケースが多くあります。

楽々 Framework II を構成している部品は、顧客や社内開発者の要望に応じて日々機能拡張が行われており、完成度が向上しています。例えば画面用の部品は表3のような機能が実装されていますが、開発工数はほとんど増やすことなく、ユーザーの操作性を大幅に向上させることが可能になっています。

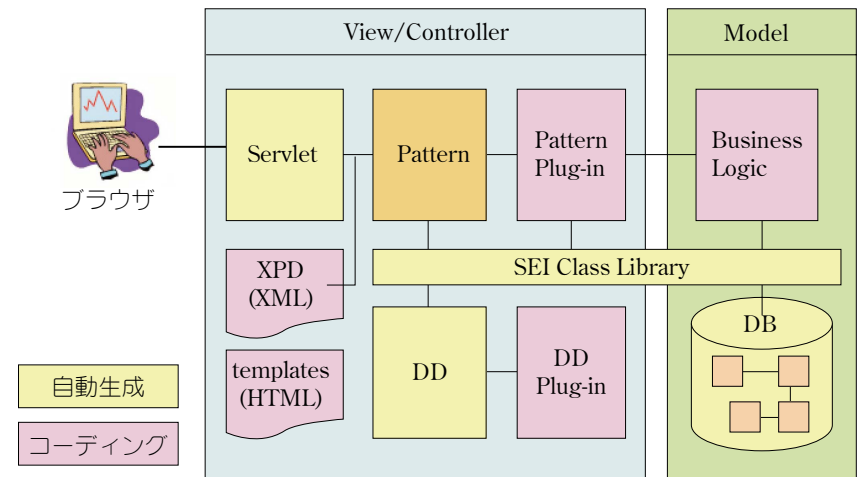


図2 楽々 Framework II で開発したシステムの内部構造

表3 楽々 Framework II の主な機能

項目	機能
ブラウザの入力項目の移動	一般にTABキーだが、カーソルキーやリターンキーによる移動が可能
コード入力時	対応する名称の自動表示
一覧表示画面	設定項目でページ切り替えが可能であり、また、データ項目ごとに昇順、降順のソート可能
日付型項目への入力	3ヵ月分のカレンダーをポップアップすることが可能であり、マウスで選択入力することが可能

2.2.5 組立型開発の実践

組立型開発の1つの課題はソフトウェア部品の構築ですが、もう1つの課題はソフトウェア部品の利用技術です。多数の部品を用意しても適切な部品を選択し、部品を利用する技術（可用性）がなければ生産性向上は困難です。そこでSEI/SISでは以下のように再利用ができる体制を構築しました。

① プロジェクトの開発に合わせたタイムリーな人材養成

設計者向け、開発者向けのトレーニングを毎月実施し、プロジェクトの開発に合わせてタイムリーに人材を養成できるようにしました。

トレーニングは演習を中心に行い、開発者は約3日間で簡単なプログラムの組み立てが可能。その後、開発現場で2～3週間OJTによる指導後一人前のプログラマとして活躍。

② 部品のネーミングルールによる部品名からの機能類推

メインルーチンの部品は、照会、登録、変更、削除、帳票といった機能区分と画面の構成（例えば、ヘッダー&ディテール形式）で絞り込めるようにするため、部品名の先頭3文字を機能区分、4文字目から画面構成およびオプション情報を示す。

（例：EntFaは、登録機能（Ent）を持つヘッダー（F）&ディテール（a）形式の入力画面）

③ 画面の自動組立

データベースの設計情報から利用可能な部品を自動選択可能とし、自動組立を

実現することで複数の画面候補を自動的に設計者に示す。設計者は自分のイメージに近い画面を選択できる。

④ 画面デザインツール

ブラウザ上で画面デザインを編集できるツールを提供し、容易な組立を実現。

2.2.6 組立型開発の効果

組立型開発による生産性向上の結果を表4に示します。生産性はファンクション・ポイントに対するプログラム開発工数で測定したものを比較しています。

表4 従来開発言語の生産性を1とした場合の組立型開発の生産性

比較対象の開発言語	組立型開発の生産性
COBOL	約3倍
Visual Basic	約2倍
Cold Fusion	約1.5倍

表4での組立型開発のコーディング量は、照会・登録・変更・削除を備えた受注や出荷のプログラム1本あたり、コメント行を除いた論理行数の平均値が約180行です。提供している画面のソフトウェア部品には操作性を改善する多数の機能が実装されており、部品を利用しない場合、数千ステップの規模になると予想されるため、コーディング量は1/10以下になったと考えられます。

2.2.7 組立型開発における品質管理

十分なソフトウェア部品が準備された組立型開発では、プログラム開発が必要となるシステム固有の部分がビジネスロジックに限定されるため、定量的な管理がしやすいという特徴があります。

具体的に品質向上策としてとられた施策は次のとおりです。

- ・部品の組み込み方法に関する欠陥の再発防止
- ・McCabe等の指標による複雑度の制御
- ・プログラムの規模指数の決定と欠陥数評価
- ・if文の数による欠陥数の予測

2.2.8 部品の組み込み方法に関係する欠陥の再発防止

SEI/SIS では過去発生した欠陥を定期的に分析し、頻度の高い欠陥について全社的に周知し、トレーニングを実施することで、再発を防止しています。例えば、Java の浮動小数点の計算では計算誤差が発生するため、浮動小数点計算をする場合には SEI/SIS で提供した計算用部品を使用します。テスト時に、整数や短い桁数の数値が入力された場合、欠陥が検出できない可能性があり、全プログラマへの教育を行った例です。

2.2.9 McCabe 等の指標による複雑度の制御

SEI/SIS では、プログラムの品質・保守性を高めるため、McCabe のサイクロマティック複雑度の上限を決め、上限を超えないよう管理することでプログラムの管理を行っています。McCabe のサイクロマティック複雑度の上限を決めることでプログラムの品質を管理できるのは、組立型プログラミングの特徴によります。図3に組立型プログラミングにおける開発内容の内訳を示します。画面出力や画面遷移等はパターンで、基本的なエラーチェックは項目オブジェクトで、共通化された部品として実装され、この部品が開発内容の80%を占めています。プログラミングは残りの20%の部分となる業務ロジック部分となります。ソースコードには業務に必要な部分のみが書かれるため、それ以外の部分でサイクロマティック複雑度が上昇してしまいがちありません。本質的な業務ロジックの部分のみをサイクロマティック複雑度に反映することが、効果的な導入につながっています。

サイクロマティック複雑度は Eclipse のプラグインによって自動計測され、プログラミング中にもリアルタイムで表示されます。プログラマが、常にサイクロマ

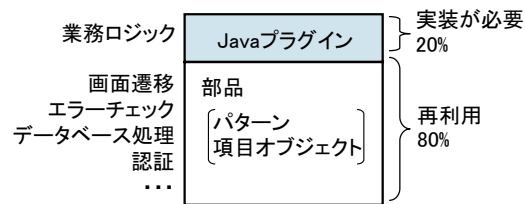


図3 組立型プログラミングにおける開発内容の内訳

ティック複雑度を意識することで、簡潔なプログラムが可能となっています。サイクロマティック複雑度が20を超えるものについては、重点的なコードインスペクションを行いレポートすることを定めています。

2.2.10 プログラムの規模指数の決定と欠陥数評価

組立型プログラミングでは Java での実装量が極めて少なく、Java ソースコードのステートメント数だけではプログラムの規模を決定できません。規模が決定できないことにより工数や欠陥数が定量的に管理できず、例えば、1本のプログラムの欠陥数が多いのか少ないのかを判断できないといった問題があります。そこで、Java ソースコードのステートメント数、および部品のパラメータといった Java 以外の記述全体から、プログラムの規模を測定する指数 JaX を決定しました。プログラムの開発工数がプログラム規模と相関があるため、JaX はプログラムの開発工数のばらつきを最小にするように統計的に決定されています。

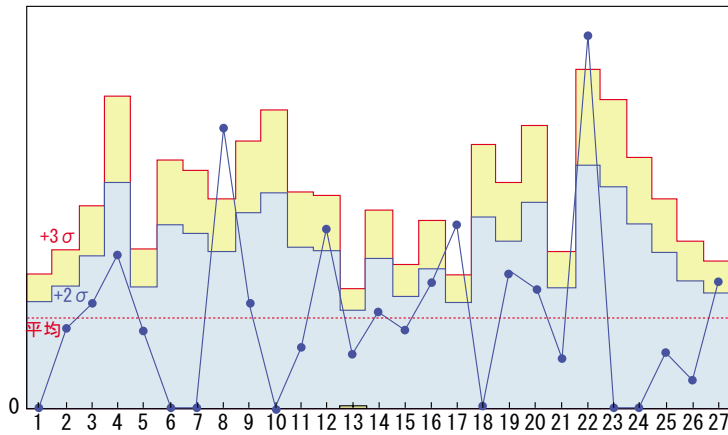
規模指数に JaX を用いることで欠陥のあるプログラムの監視も行いやすくなります。図4(a)は横軸に各プログラムを取り、Java ソースコードの1,000ステートメントあたりの欠陥数を用いて u 管理図を描いたものです。同様に図4(b)は1,000 JaX あたりの欠陥数を用いた u 管理図となっています。JaX ではプログラム規模を適切に推定して規模のばらつきを抑えるため、図4(a)に対して、図4(b)は狭い幅で安定しており、管理範囲内から外れたプログラムを見つけやすくなっています。

2.2.11 if 文の数による欠陥数の予測

SEI/SIS ではプログラム要素と欠陥の相関を分析しており、部品の再利用性が高く業務ロジックのみを書く組立型プログラミングの特徴から、プログラム中の if 文の数と欠陥の数には相関があることが判明しています。if 文の数と欠陥数の相関を用いることで、テストで除去すべき欠陥数を推定することができます。

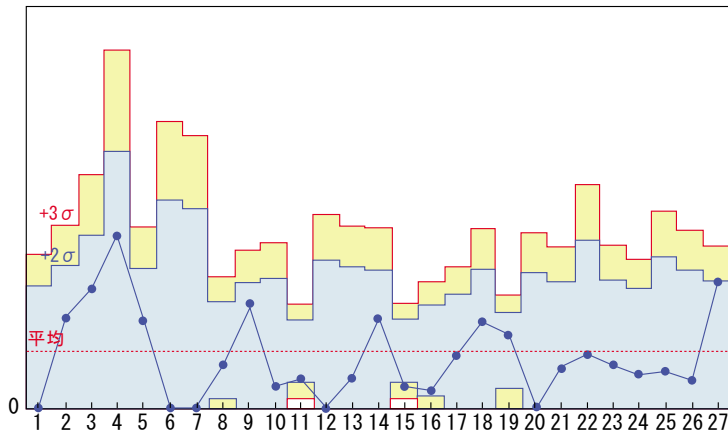
図5はあるプロジェクトの if 文の数とテストで除去した欠陥の数の相関を示しています。横軸に if の合計数、縦軸にはテストで除去した欠陥数を取り、各プログラムをプロットしています。点線で示した基準値の範囲に点が入っていれば、

欠陥検出数 / K ステートメント数 (Java)



(a) Java の1,000ステートメントあたりの欠陥数を用いた u 管理図

欠陥数 / KJaX



(b) 1,000JaX あたりの欠陥数を用いた u 管理図

図4 Java ステートメント数に対する欠陥数と JaX に対する欠陥数の対比

テストで適切な数の欠陥が除去できていることとなります。多くの場合この範囲内におさまりますが、範囲を外れている場合、作られたプログラムの品質が悪い、

あるいは、テスト工程が不十分である可能性があることを示しています。この場合は範囲を外れた理由を分析し、必要に応じて重点的なテストやプログラム製造者のトレーニングを行っています。

テストで除去すべき欠陥数を標準化し、これに達するまでテストを行うことでプログラムが一定の品質にあることを保証しています。

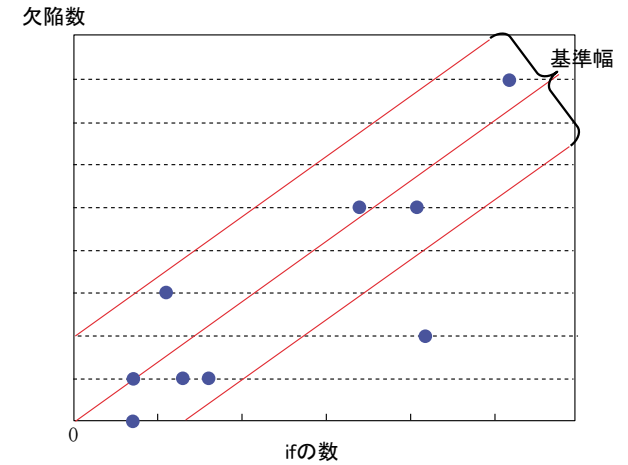


図5 ソース中の if 文の数と除去欠陥数の相関図

2.2.12 プログラム品質管理の成果

SEI/SIS ではプログラムの欠陥の定量的管理を行うため管理図を用いています。管理図にはいくつかの種類がありますが、規模の異なるプログラムに対する欠陥の数を扱える u 管理図を用いています。u 管理図は工業生産の場では品質管理のために一般的に用いられていますが、ソフトウェア開発への適用例は少なく、SEI/SIS の特徴です。

図6は欠陥数の u 管理図を示したものです。横軸に各プログラム、縦軸に規模あたりの欠陥数を取り、ある1プロジェクトの結果を示しています。管理図では 2σ の範囲に95%、 3σ の範囲に99.7%の点がおさまる分布となっており、 3σ の範囲を出ると管理から外れた異常な点として対処が必要となります。プログラ

ム製造の場合、 3σ の範囲から外れたプログラムは品質に問題があるものとして、コードインスペクションやテストが行われます。

●組立型プログラミング固有の改善、定量的管理の結果

以上のような組立型プログラミング固有の改善、定量的管理を行うことにより、SEI/SISでは欠陥の数が20件/KJaXから10件/KJaXに半減しました。また、図6に示すように各点が 2σ の範囲におさまっており、プログラム製造工程が安定していることがわかります。

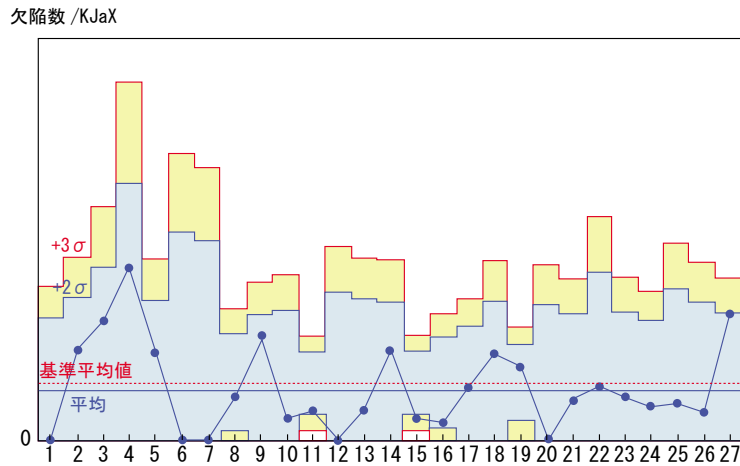


図6 プログラム欠陥数のu管理図

2.2.13 本番稼働後のシステムの品質管理

SISは、2003年にCMMレベル3を達成し、ファンクション・ポイントあたりの欠陥数を管理してきました。しかし、工程単位の管理は、各工程が終わった段階でしか目標の達成度合いが評価できないため、実際のプロジェクトでは本番出荷後の品質を制御することが困難でした。今回導入した定量的な品質管理を実施することでプログラム設計、プログラム開発の進行中に品質保証ができるようになり、工程終了時の品質が制御できるようになりました。この結果、本番稼働後の品質もよりよく制御できるようになりました。

図7は、プロジェクト単位で欠陥の作り込みと除去を示したものです。この図はレビューやテストの結果からリアルタイムに作成されます。そのため、基準値との差異をこの図から判断し、次工程（例えば統合テスト）での摘出欠陥数の目標値の変更が可能となり、最終品質の制御が可能となります。また、欠陥フロー図の山のピーク位置は開発チームのレベルが上がるほど上流工程に移動し、全体の欠陥数が少なくなる傾向があります。欠陥フロー図の形は開発チームの品質管理能力の目安になります。

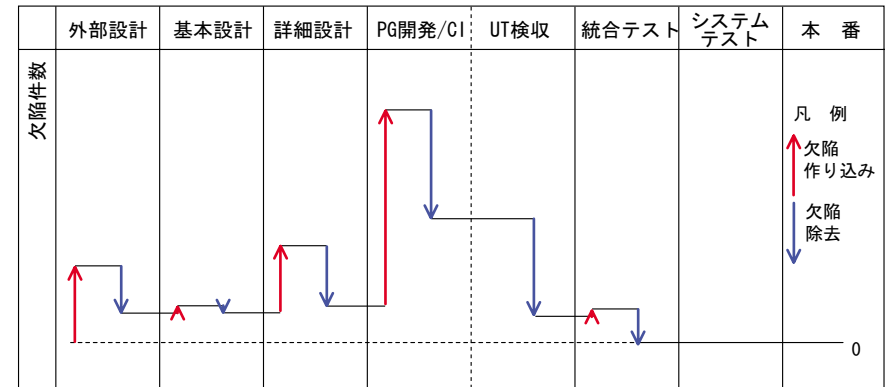


図7 欠陥フロー図（推移図）

2.2.14 管理ツールの開発

統計的品質管理で使用する管理図等のグラフは作成することが目的ではなく、管理図を使ってプロジェクトをコントロールすることが目的です。多忙なプロジェクトリーダーに毎週管理図をチェックしてもらってプロジェクトの状態を把握してもらう必要がありますので、自動的に管理図、相関図が出力できるツール（図8）を自社開発しました。u管理図を手軽に描画できるソフトが入手できないことも自社開発の理由です。

プロジェクトのQCDを管理する為には成果物（仕様書やソース）とプロセス実績のデータが必要となるため、構成管理ができる文書管理ソフトを開発し、各文書（ソースの場合は文書に添付）にプロセス実績のデータが登録できるようにしました。成果物の規模が自動測定され、成果物に対するレビュー／テスト記録か

ら自動的に管理図等が出力できます。管理図を含めすべてのドキュメントはブラウザで閲覧できるようにし、管理職や SQA 担当者、SEPG も簡単にプロジェクトの状況が確認できるようにしています。

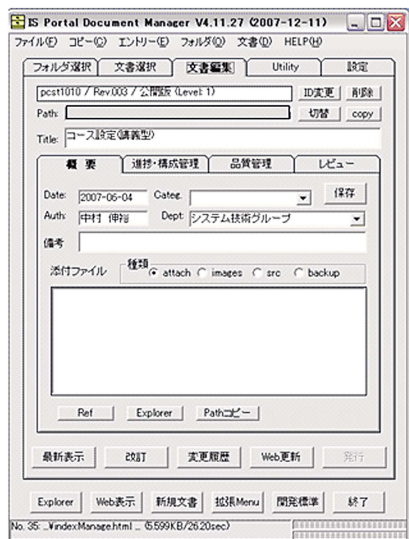


図8 構成管理・グラフ作成ツール

2.2.15 プロセス改善の全社展開

SEI/SIS では、これまで技術的改善を全社展開してきましたが、CMMI レベル5の組織改革と展開のプラクティスを参考にしながら改善施策の展開の制度化に取り組みました。改善策の展開にあたり、以下のような問題が想定されます。

- ・担当者は、業務遂行に手一杯で管理に十分な時間が取れない
- ・今までのやり方を変えたくないという意識が強い
- ・担当者が目的を正しく理解できておらず、余計な仕事と思っている
- ・改善活動に理解を示さない課長が改善活動の工数を確保してくれない
- ・ツールの使い方がわからない

これらの問題を解決するため①課長と担当者のプロセス改善に対する意識を変

え、②管理活動の工数を確保し、③スキルを身に付ける、という基本方針に従って以下の活動を実施しています。

(1) プロセス改善推進委員会

月1回開催する会議で社長を委員長とし各部門の部課長と SEPG メンバが参加してプロセス改善の方針を決定し、改善の進捗状況を確認します。全社の管理職の意識を合わせ、改善施策の展開を加速させます。

(2) 品質大会

ソフトウェアの品質向上を目指して品質大会と称する全社大会を2006年より年4回実施しています。

品質大会では、①品質や生産性を向上させるために実施した取り組みと成果の発表、②新たに取り入れた手法の紹介、③失敗事例の紹介等を行っています。

品質大会は社長以下全社員参加で実施し、全社員で改善活動の状況や方向性を共有しています。他のプロジェクトを知ることで、担当者の品質や定量管理に関する意識が向上しています。発表回数を重ね、現在ではより効果的で定量的な発表内容になっています。

(3) 定量管理トレーニング

定量的プロジェクト管理の概念や統計に関する知識、品質のコントロール手法を習得するための概要編と、ツールを使った具体的な手順を習得するための実践編を組み合わせたトレーニングカリキュラムを構築し、定期的に各地区で実施しています。

概要編では実プロジェクトの管理図による具体的な品質コントロール状況の説明や、統計手法と管理図の一般知識に関する説明を行っています。実践編では実際に管理図や相関図を作成することで理解を深めてプロジェクトの担当者への定着化を図っています。

(4) 個別指導および支援

定量管理トレーニングで基本的な手法や手順を学習しますが、各プロジェクトで実践するには品質管理推進室の担当者が個別に指導しています。

また品質管理推進室では各プロジェクトが作成した管理図を定期的にチェックし、必要に応じて管理図の作成支援や指導を行っています。

(5) SQA 検証や納入レビューでの利用

月1回開催する SQA 検証でも統計的品質管理の実施状況をトレースして問題分析の結果確認も行っています。社内テスト完了後に納品可否判定を行う納入レビューでは品質指標値を採取して分析と対策が実施されている事を合格の条件としています。

(6) プロジェクト内の定例トレース会議での取り組み

プロジェクト内の定例トレース会議では進捗状況やコスト状況の他に品質指標である McCabe や欠陥数の管理図を見て異常値を分析し対策を検討しています。

上記6点の取り組みの結果、管理図を使った品質管理が全システム開発で実施されるようになりました。

2.3 活用しているツール、技法

- ・データ中心設計（T字形 ER 手法）
- ・楽々 Framework II
- ・IS Portal Document Manager（構成管理、自動測定、管理図、EVM 作成ツール）
- ・組立型開発／組立型プログラミング

2.4 人材育成

部品化・再利用を推進するために以下のトレーニングを実施しています。

① 楽々 Framework II プログラム開発トレーニング（3日間）

部品の使い方を習得します。

② 楽々 Framework II 外部設計トレーニング（半日）

既存部品が活用できる外部設計のポイントを習得します。

③ 定量管理トレーニング（3日）

統計の基礎を理解し、管理図の作成方法を演習を通して習得します。

3. 学ぶべき点

部品化再利用では、仕組みは作ったが部品が登録されない、部品は溜まるものの活用が進まない等の課題に直面して頓挫しがちです。この事例では、20年以上にわたる継続的な創意と工夫によって、部品化再利用による生産性と品質の向上を実現しています。技術的なアプローチもさることながら、定着化に向けた施策、データに基づく品質の制御の仕方も見習うべきところがあります。

参考文献

佐藤正美：T字形 ER データベース設計技法，ソフト・リサーチ・センター，1999年

佐藤正美：データベース設計論 T字形 ER，ソフト・リサーチ・センター，2005年

BEST
PRACTICE
事例9

全社レベルでのプロジェクト管理の改善

～段階的にツールを導入し作業内容変更の最小化をはかる～

日立ソフトウェアエンジニアリング株式会社

日立ソフトウェアエンジニアリングでは、複数の事業部それぞれが多くのプロジェクトを実施しています。各プロジェクトはプロジェクトマネージャ（PM）が管理し、その状況は課、部、本部、事業部、全社 PMO へとエスカレーションされています。

一般にプロジェクト、課、部、本部、事業部の各レベルでプロジェクト管理項目は異なり、全てのプロジェクトにおいて詳細項目を細かく管理していますが、課、部と上位レベルの管理になればなるほど概要レベルでの監視の傾向が強まります。

日立ソフトウェアエンジニアリングではプロジェクトの管理項目、各項目の評価方法等が規格で決まっていますが、実際に使う管理ツールや、データの管理方法等は各プロジェクトの実情に合わせて決めています。このなかで得られた良い管理方法は、課内、部内に伝えられ、利用されてきました。本部、事業部へのエスカレーションも、同様に工夫されたものとなり、事業部、本部ごとに、少しずつ異なった管理方法を実施していました。

そして、プロジェクトごとの違いおよび標準ツールを使用していないことは、本部長、事業部長、全社 PMO が各プロジェクトを監視する際の弊害になっており、全社的な改善の困難さをもたらしていました。

そこで、プロジェクト管理の標準化をすすめ、全社の主要プロジェクトの状況を関係者がいつでも監視できるようにすることに取り組み、全社で利用する統一的なプロジェクト管理ツールの導入を進めました。

今回の取り組みにおける一番の課題は、これまでもプロジェクトそれぞれについてき

キーワード

- ・プロジェクト管理
- ・全社プロジェクト管理システム：EPM（Enterprise Project Management）

ちんと管理されており、システムを導入しても、直接個別のプロジェクトの管理の改善とはなりません。したがって、今回の取り組みはPMにとってのメリットがわかりにくく、また管理方法が部、本部レベルである程度標準化されている部署でのメリットもわかりにくいため、ユーザの協力の不安がありました。

そこで、まず簡単なシステムを導入し、システムのメリットを実証してシステム導入のコミットメントを得た後、システムの機能拡張をしてユーザを拡大、さらに操作性を改善し全社導入という、段階的な導入戦略をとることにより、ユーザの理解を得、全社規模での導入に成功しました。日立ソフトウェアエンジニアリングでとった導入のフェーズは以下の3フェーズです。

- ・フェーズ1：効果を知ってユーザのコミットメントを得る
- ・フェーズ2：ユーザの拡大
- ・フェーズ3：全社活用

1. プロセス改善活動の背景

1.1 改善方針

今回の改善は、プロジェクト管理の標準化と、全社の主要プロジェクトの状況を関係者がいつでも監視できるような可視化でした。

留意すべき点は、改善以前から各プロジェクトの状況はPMが把握し、問題点等があれば、上位管理者にエスカレーションされていたということでした。つまり、PMの立場から見ると、当初は特別なメリットがないことになります。そのため、どんなに高機能なシステムを開発しても、システムが利用されない恐れがあり、これが一番のリスクです。このリスクを避けるため、次の方針でシステムの開発と導入を進めました。

- | | |
|---|---|
| ■日立ソフトウェアエンジニアリング株式会社
(本書で用いる略称：HSW) | ・システム開発
・サービス
・プロダクト&パッケージ
・情報処理機器 |
| ●ビジネスゴール
・持続的成長の実現 | |
| ●HSWの製品・サービス | URL http://www.hitachisoft.jp/ |

- ① 利用者にとって、追加コスト（作業等）と効果、利点等がバランスすること
- ② システムの導入の前後で主要な関係者の役割や作業が大きく変わらないこと
- ③ 改善以前のやり方からスムーズに移行できるようにシステムを設計すること
- ④ 改善に当たり、利用者からの要望、要求にタイムリーにフィードバックすること
- ⑤ フェーズを決めて段階的にリリースすること

1.2 達成目標

HSWでの今回の改善は、システムの開発と提供だけでなく、そのシステムを使い、効果的にプロジェクトの監視が行われることを目指し、具体的には、以下の3点を達成目標としました。

- ① プロジェクト管理に関する情報を1つのシステムで統合管理する
- ② プロジェクトのステータス情報を管理するための1つのシステムを全員が使うようになる
- ③ 上級管理者がプロジェクトのステータスを確認するときには、個別にPMに確認する前に、システムを利用して確認する

1.3 プロセス改善推進体制

HSWでは、準備したシステムを確実に利用されることを目指し、最初から適用の拡大を意識した次の体制としました。システムが定着した後は、全社PMOとシステム部門がシステムの維持改善を行っています。

- ・全社PMO：利用部署への説明と運用支援
- ・プロセス改善技術センタ：マスタープランの策定とシステムの開発
- ・利用部署：とりまとめ者を置き、利用部署内の適用支援、問題点の吸い上げを実施

1.4 プロセス改善対象組織の規模

- ・全社員約5,000人

1.5 改善活動の経緯

HSWの今回の改善への取り組みの動機には、プロジェクトの監視の効率向上とプロジェクト管理の質向上の2つがありました。管理が不十分では、プロジェクト実施に問題が生じるので、タイムリーに効率よくプロジェクトの状況を全社PMO等が監視する必要がありました。

1.6 主要な課題と改善のフォーカス

HSWでは、もともと原価管理システムや、作業量管理システム等があり、プロジェクトの管理に利用できるデータが社内の各種システムで管理されていたのですが、これらのデータを有効利用するためには、各種システムそれぞれからデータを入手する必要があり不便でした。この課題を解決するために、共通的なプロジェクト管理ツールの開発と定着化を推進しました。

また、改善を進めるにあたっては以下の懸念の払拭が必要でした。

- ① 各管理者の管理スタイルができあがっていて、これまでそのスタイルでプロジェクトの管理を成功させてきており、管理者は、標準化のための変更の必要性を感じていない
 - ② 上級管理者も、自分の管理範囲では、これまでのやり方で困っておらず、また、自分の見やすい方法で報告させたい
 - ③ 技術者、管理者は、定量的で詳細な報告は面倒であり、報告することによるメリットを感じていない
- これらの課題を解決するために、段階的アプローチを採り、ツール統一化のメリットと負荷のバランスを図りながら推進を行いました。

1.7 顕在化した改善効果

表1に改善効果を示します。

表1 顕在化した改善効果

内容	改善効果
(1) 管理、報告	・各プロジェクトの管理データの標準化の進行 ・報告方法と資料が標準化され、PMの報告資料とその作成業務が削減
(2) 報告のリアルタイム性	・プロジェクトマネージャがプロジェクトの状況をシステムに登録することで、関係者はいつでも、最新の状況を把握可能
(3) 過去データの扱い	・データベースより過去のすべてのプロジェクトの状況を入手可能
(4) 全社プロジェクトの一覧性	・全社のプロジェクトの状況を一覧画面で確認可能
(5) 問題プロジェクトの発見速度	・プロジェクトの一覧から、動きのおかしいプロジェクトを早期に発見可能

2. ベストプラクティス

2.1 参照モデル上でのカバレッジ

- ① 当該ベストプラクティスに対応するISO/IEC 15504 第5部のプロセス
 - ・プロジェクト管理 (MAN.3 Project management)
- ② 当該ベストプラクティスに対応するCMMIのプロセス領域
 - ・プロジェクトの監視と制御 (PMC: Project Monitoring and Control)
- ③ 当該ベストプラクティスに対応する共通フレーム2007のプロセス/アクティビティ
 - ・管理プロセス

2.2 プラクティスの説明

HSWが導入したプロジェクト管理システムは、特段開発が困難なものでもなく、システムの機能もとくに先進的なものではありません。が、大人数の組織にシステムを導入し、確実に利用されるようにすることを念頭に置いて、「小さくはじめて、小さな改善ループと大きな改善ループを回す」ことでシステムを強化しています。

2.2.1 段階的アプローチ

HSWでは、まずシステムが便利であり、効果があることをスタッフに納得してもらうことを最重点テーマとしました。次に、PMにとって便利で効果があるようにシステムの機能を拡張しました。最後の段階で、全社に広げることを念頭に、操作性等を改善しました。表2に、各フェーズの目的、ターゲットとする利用者、システムの概要を示します。

表2 HSWでの段階的アプローチ

フェーズ	目的	利用者	システムの概要
1	・関係者がシステムの利点を知る ・コミットメントを得る	一部プロジェクト一部PM	・スプレッドシートベースの報告帳票の標準化 ・上記帳票の収集と、スプレッドシートベースのDBへの集約 ・スプレッドシートの管理情報をWebで公開
2	・ユーザの拡大 ・実際のプロジェクトでの適用	一事業部多くのPM	・指標の追加 ・工程会議で利用する帳票を出力 ・既存の原価管理システム等と接続
3	・操作性の改善	全社	・スプレッドシートからDBMSへの変更 ・プロジェクトの状況を容易に確認できる画面の追加 ・Webからのデータチェック、更新機能

2.2.2 関係者と機能の絞り込み

HSWでは、効率の良い管理システムを検討すると、プロジェクトの全員が進捗、品質等のデータを管理システムに登録し、集約するようなシステムが求められます(図1)。このシステムは計算機システムとして、効率のよいシステムですが、移行を考えた場合、必ずしも、ベストとは考えられません。

担当者全員にデータをタイムリーに入力させるのは、入力方法の変更等大変な労力を必要とします。そのため、HSWでは、新しい管理システムの利用による作業内容の変更が必要となる要員はPM以上とし、プロジェクトの他のメンバは従来どおりのやり方としました。実際の作業では、標準化されたプロジェクトの管理帳票システムに、PMが登録作業を行います。上位管理者は、システムを参照してプロジェクトの状況を知ることができ、進捗等の会議では、システムに登録されているデータを使うこととしました。このようなシステムの概要を図2に示します。

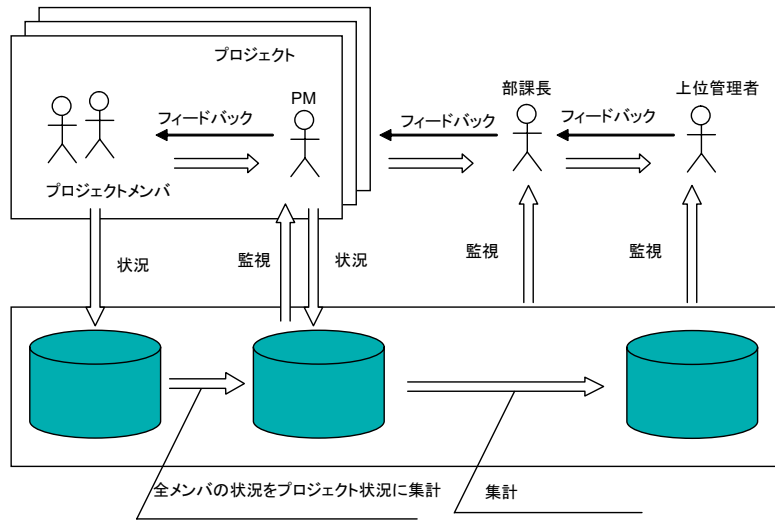


図1 理想の管理システム

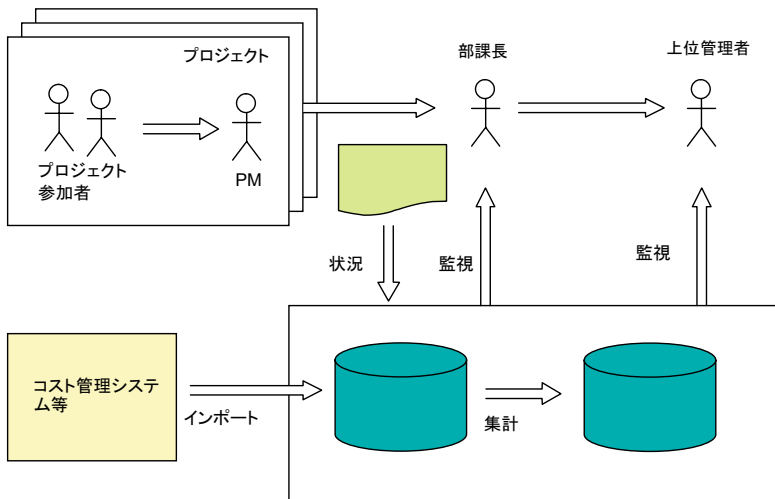


図2 実現すべきシステム

2.2.3 各フェーズの概要

(1) フェーズ1：コミットメントを得ることを目的とした最初のシステム

HSWでは、はじめにシステムの開発と、そのシステムを上位管理者が利用することのコミットメントを得ることを目的としたシステムを開発しました。このシステムでは、上位管理者がプロジェクトの一覧を見ることができるようになることで、どんなメリットがあるかを関係者に示しました。

また、このシステムを実際に適用するためには、PMがシステムにデータを入力する必要がありますが、この作業の手間が少ないことが重要です。また、社屋外からも利用できる必要があります。

最初のシステムのポイントを、

- ・上級管理者に便利な情報を提供できること
- ・そのためのPMの負担が少ないこと

の2つに絞りました。システムは、大量データを処理できる必要はなく、きめ細かいユーザ対応も不要としました。

この条件を満たすためのシステムの構成は図3のようになりました。PMは直近の状況に更新された標準管理帳票をシステムにメールで送付します。システムは、受け取ったデータを元に管理情報を管理情報用のスプレッドシート（図3上の集計データ）に書き込みます。そして、システムは上位管理者にスプレッドシート（図3上のサマリーデータ）を提供するとともに、スプレッドシートのマクロによりHTML形式に書き出し、Web公開します。

このため上位管理職は、プロジェクトの一覧をスプレッドシートで入手するだけでなく、Webでも閲覧できます。

また、PMへのデータ入力の促進は、従来からプロジェクト状況の確認、一覧作成および報告を行っていた、とりまとめ担当者の協力により行いました。

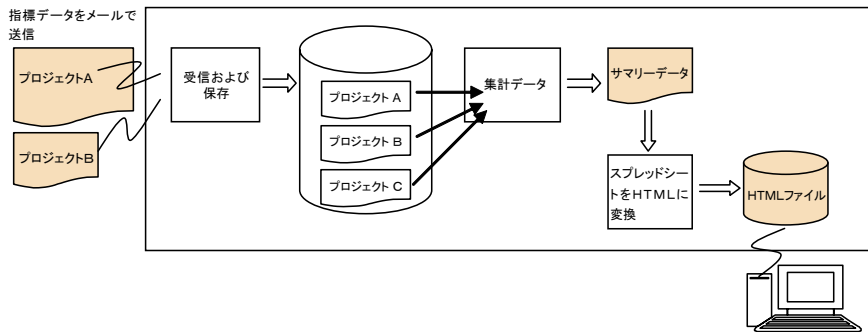


図3 フェーズ1におけるシステム構成

(2) フェーズ2：ユーザ拡大のための2番目のシステム

2番目のシステムでは、実際のプロジェクトで利用することを念頭に利用者の利便性の向上を図るため、標準管理帳票に、社内の各システムで管理している原価情報等を取り込む機能を追加しました。これにより、管理に利用できるデータをPMが容易に利用できるようになり、利用者の拡大に寄与しました。このシステムの概要を図4に示します。

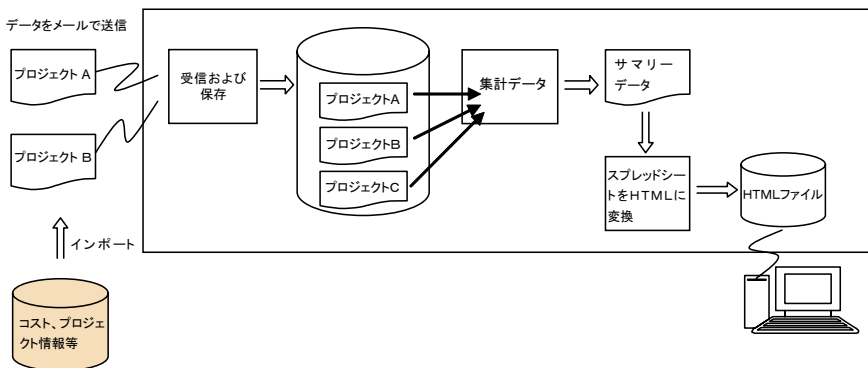


図4 フェーズ2でのシステム構成

(3) フェーズ3：全社利用のための3番目のシステム

全社に広げるには、効果があるだけではなく、使いやすいことが重要となります。そこでデータを格納するDBMSを導入し、各種状況を表すグラフの表示を可能とし、また、進捗等の確認の会議で利用する帳票を出力する機能も付加されました。

その一方で、データを入力する仕掛けは従来と大きく変えず、メールでスプレッドシートを送るとデータベースに自動的に格納されます。このシステムの概要を図5に示します。フェーズ3では、システムのアーキテクチャは確定しており、利用者也全社に拡大しました。従って、その後は、必要に応じた機能追加、操作性の改善を続けています。

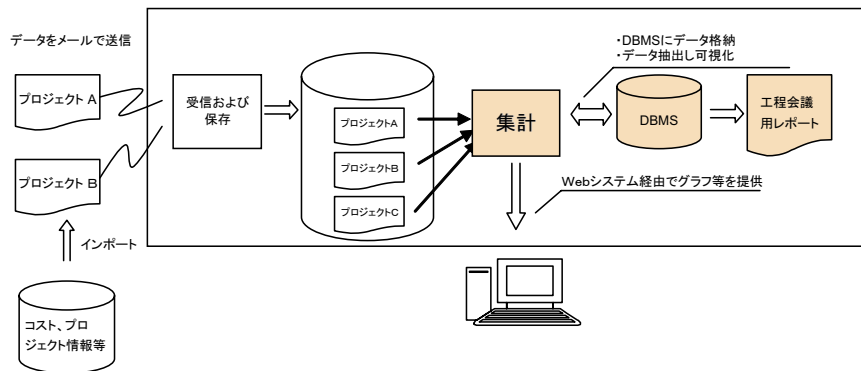


図5 フェーズ3でのシステム構成

2.2.4 システムの全体像

さて、前節まではシステムの移行方法（構築方法）に焦点をおいて説明しましたが、ここからは、全社プロジェクト管理システムの機能面について説明します。図6に今回構築したシステムの全体像を示します。

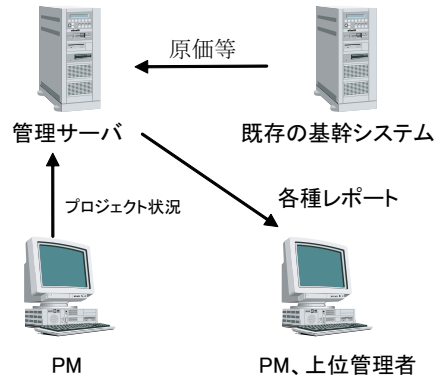


図6 システムの全体像

PMは、プロジェクトの状況を図7に示すようなスプレッドシートで管理します。これを管理サーバにメールで送ります。いったん送った後は、Webで更新したり、データを追加したりすることもできます。

原価等は、本システムの導入以前から別システムで管理されていますので、これを定期的にも本システムの管理サーバに取り込みます。これらの情報を見やすいように加工し、PMや、上位管理者が参照できるようになっています。

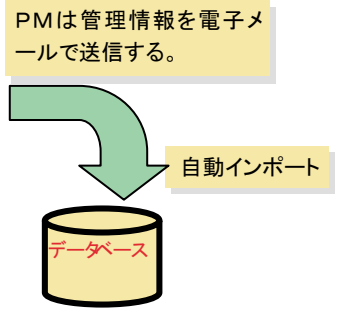


図7 プロジェクト状況情報の収集

2.2.5 全プロジェクト状況の可視化

上位管理者は、自分の管理対象である多くのプロジェクトについて、問題に早期に対策する必要があります。そのため、プロジェクトの状況を一覽し、気になるプロジェクトについて、より詳細に確認できるようになっています。図8にプロジェクトの一覽の例を示します。

図8 プロジェクト一覽

この画面では、管理者ごとに、担当のプロジェクトの一覽が表示され、原価の利用状況、規模の推移、スケジュール、品質の状況について表示されます。この表示はあらかじめ決められた基準に従い、緑、黄、赤の3色で色分けされます。この表示をもとに気になるプロジェクトがあれば、詳細を確認することができます。

2.2.6 プロジェクト状況の可視化

プロジェクトの一覽で気になるプロジェクトがあったとき、詳細情報として、PMが登録したすべての情報と、原価システムで管理している原価の状況を品質保証担当者がモニタし、報告している情報を確認することができます。進捗に関しては、計画と実績の差異と、仕上がり状況を表示します(図9)。品質に対しては、プロジェクトの概要と、品質保証担当者の報告を表示します(図10)。

さらに、進捗の問題があるときの問題分析のツールとして、原価の使用状況と、

進捗の状況を併せて確認する画面などを用意しています（図11）。

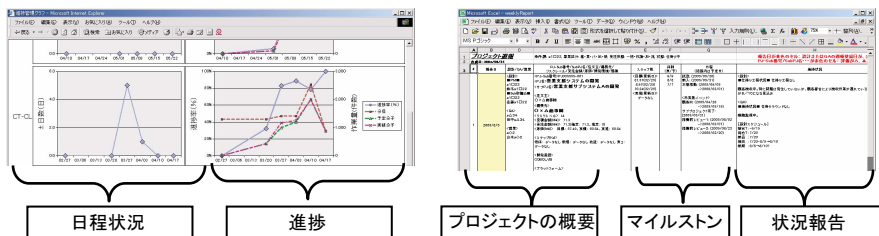


図9 進捗状況の可視化

図10 品質報告

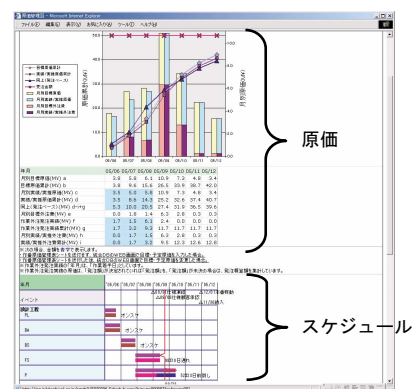


図11 原価・進捗

2.2.7 今後の方向

現状収集しているデータは、現状での活用以上にいろいろな方法で活用できます。今後、これをより広い範囲で役立つためには、データを分析し、各種の定量的なパフォーマンスベースラインを確立し改善し続けることが必要です。

また、このシステムはPMから情報を収集し、プロジェクトの管理制御に利用しはじめていますので、今後、新たにデータ必要になったときには、この枠組みに載せることで比較的短時間で新たなデータを収集、活用できるようになります。データを利用した客観的で定量的な管理のためにこのシステムを役立てることが

できます。

2.3 活用しているツール、技法

一般的なツール、技法は使用していません。

2.4 人材育成

全員を対象とし、トレーニングを実施。

3. 学ぶべき点

プロセスを改善し、それを定着させるのは容易なことではありません。改善したいプロセスを実施している本人が、改善の必要性を感じていない場合も多くあります。とくにプロセスを改善する関係者が多い場合、また、改善の効果を得る人とプロセスの改善をする人が異なる場合はより定着が困難となります。

HSWでの事例は、プロセスの改善により、作業方法の変更を余儀なくされるPMには直接的な見返りが少なく、大きい見返りを得るのは上位管理者であり、理解を得るために各フェーズでの注意が必要でした。また、PMの管理自体の問題改善ではなく、全社標準化のために作業プロセス改善したケースです。

このようなケースでは、改善を小さくはじめて、その後の不断の繰り返しにより、戦略的に改善を成功させた事例であり、改善方法の戦略作成の参考となります。

BEST
PRACTICE
事例10

GDD (Genba Driven Development)

～現場主導によるアジャイル開発のすゝめ～

ソニー株式会社

PC アプリケーションソフトウェアセンター 設計部

ソニーのPCアプリケーションセンターでは、現場開発者達の感じていた不安や不満が発端になり、現場主導による改善活動がはじまりました。当時現場では、スケジュールに追われた場当たりの開発が横行し、無理矢理に全機能を盛り込み、ソフトウェアをリリースする等し、そのため、その後の検証工程では、当然の事ながら手戻りが多発し、不具合改修に追われ、ゆとりのない時間をさらに圧迫していました。その状況下で、まずは現場の声を集め解決すべき課題を3つに集約しました。

- ・課題a レビューによる他者のフォローが無いため、間違いが多発
- ・課題b 実装完了時点で品質保証が行われていないため、検証工程で手戻りが多発
- ・課題c 形式知になっていないため、プロジェクトメンバを助けられない・助けてもらえない

これらの課題を解決するために、以下の4つの利点を持つeXtreme Programming (以下XP)を参考にしました。

- ・XPの利点a ソフトウェアは人間が作成するということを意識
- ・XPの利点b 実装経験者なら直感で良いと感じる内容
- ・XPの利点c 具体的なプラクティス群の提示
- ・XPの利点d 導入費用が不要

時間にゆとりがない中での課題解決のために、現場で実感の持てる改善から実行しよう、

キーワード

- ・現場
- ・アジャイル開発
- ・プロジェクト改善

というのが改善活動のテーマでした。改善活動を現場の担当者のみで行うことで、調査・報告・汎用性の考慮・文書化等の作業から解放されるため、本質的な改善活動に集中できるのでは、という狙いもありました。

XPの上記4つの利点とプラクティス群の一部を参考にし、「XP実装」(2.2.2 XP実装)、「Iterative Postmortem」(2.2.3 Iterative Postmortem)を実践し、一定の効果をえました。この2つのプラクティスは、XPのプラクティスである『短期リリース』、『テスト駆動開発』、『リファクタリング』、『共同所有』を参考にし、プロジェクトにて以前から行っていたプラクティスである『常時結合』、『リリース前検証』、『ポストモーテム』と連動させました。

「XP実装」では、設計時点での品質の作り込みを行うことができ、その結果、検証工程で検出される不具合数が減少するという効果をえました。また、レビューやリファクタリングによって、知識の共有も行うことができ、「Iterative Postmortem」ではプロジェクトの課題を全員で共有することを実現、常に改善を行う体質を身に付けました。この2つのプラクティスは、現場の開発者だけで十分に実施することが可能なものであり、かつ、より小さいプラクティスに分割して実施することも可能です。

1. プロセス改善活動の背景

1.1 改善方針

改善方針は、以下のとおりです。

- ① 現場でできることから実感の持てる改善を進める (時間に余裕がない中で課題を解決する)
- ② 改善活動を現場の担当者中心で進める (本質的な改善活動に集中する)

■ソニー株式会社 PCアプリケーションソフトウェアセンター 設計部
(本書で用いる略称：SPC)

●製品・サービスの概要
音楽管理用アプリケーションソフトウェア「SonicStage CP」

●ビジネスゴール

製品の価値を高めるPCアプリケーションの提供

URL <http://www.walkman.sony.co.jp/sscp/>

1.2 達成目標

達成目標は、次のとおりです。

- 目標 a できるだけ早く間違いに気付く
- 目標 b 検証工程での不具合指摘を減少させる
- 目標 c 独りで抱え込まない・込ませない
- 目標 d プロジェクトの推進上の課題を常に改善する

1.3 プロセス改善推進体制

プロセス改善推進体制は、次のとおりです。

- ・改善推進メンバ（開発プロジェクトメンバから3名）
- ・開発プロジェクトメンバ全員

改善対象プロジェクトのメンバのうち、3名が中心となって推進し、改善活動は開発プロジェクトメンバ全員で実施しました。

1.4 プロセス改善対象組織

プロセス改善対象組織は、ソニー株式会社 PC アプリケーションソフトウェアセンター 設計部 OpenMG Audio Client SDK for Windows 開発プロジェクトです。本活動では、組織ではなくプロジェクトレベルで活動を実施しました。

1.4.1 開発プロジェクトメンバ

・人数：プロジェクトマネージャ：1名、プロジェクトリーダー：1名、開発者：10名、設計内検証者：8名
 （設計内検証者とは、ホワイトボックステスト可能な、設計者と連携して検証を行う検証者です）

- ・特長① メンバ変更がほぼない
- ・特長② 開発初心者がいない

図1に開発プロジェクト（OpenMG Audio Client SDK for Windows）と本活動の推進メンバの位置付けを示します。

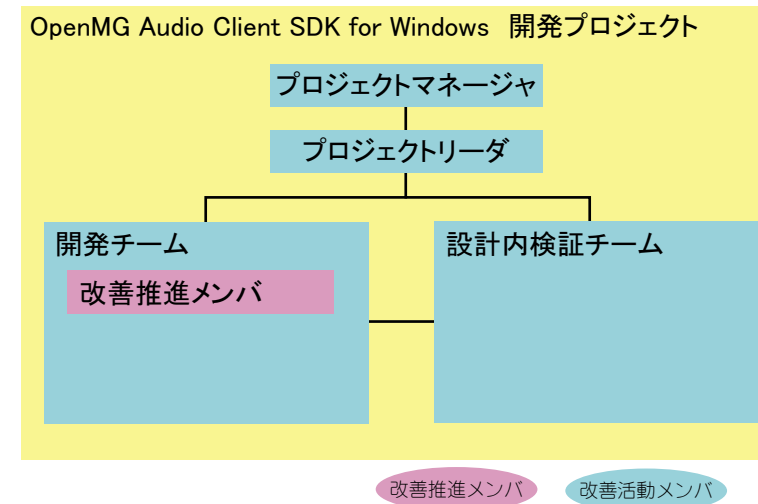


図1 開発プロジェクトと推進メンバの位置関係

1.4.2 開発の概要

- ・音楽ファイルを取り扱う Windows Middleware（UIなしのアプリケーション）
- ・社内外に提供
- ・Client Application：22種類
- ・配信サービス：3種類
- ・接続可能機器：11種類
- ・対応 OS：Windows Vista、Windows XP、Windows 2000
- ・1999年から現在まで継続している大規模プロジェクト
- ・約1,000 (KLOC)
- ・全世界展開

1.4.3 開発言語・環境

- ・C++
- ・C
- ・Microsoft Visual Studio .NET 2003

1.5 改善活動の経緯

SPCでは、4年前（2003年当時）、現場の「なぜ自分だけがこんなに大変なのか」という生の声をきっかけにして、改善活動がはじまりました。具体的にはチームメンバーの中で「XP」に注目したメンバー数人でWebやセミナー参加を通して検討していきました。その過程で、「自分たちが動きはじめなければ、自分たちは楽になれない」と考え、改善活動を開始しました。また自分自身が「良いと体験できた仕事」はチームメンバー全員にぜひ味わってほしいと感じました。以下に当時感じていた「大変さ」の例を挙げます。

- ・人のコードを調査、修正、検証するのに、相当な時間がかかる
- ・無理矢理盛り込んだ全機能をリリースした後は、不具合改修の日々が続く
- ・検証チームの負荷の高まりが目に見えてわかる
- ・成果物リリース後に安心して家に帰りたい
- ・スケジュールに追われた場当たりの開発をやめたい
- ・イベントやマイルストーンはあるが、開発プロセスが明確ではない
- ・プロセス改善の本を読んでも、具体的に何をやって良いのか不明
- ・現場からすると、CMMIベースの活動は重荷
- ・開発者からすると、意味のある作業だと思えないToDoが多い

上記問題の解決を組織やマネジメントに依頼したとすると、まず、問題の分析・理解のためのイニシャルコストが発生します。また、確実に伝わるという保証もありません。そこで、SPCでは厳しいスケジュールの中で、確実に問題解決するために、現場主導を選択しました。

SPCにおいて現場主導の改善活動を開始してから4年経過しますが、姿形は変わってきていますが、活動は継続しています。実際の開発者が考えたプラクティス群は、継続され、洗練され、プロジェクト力が生まれ、モチベーションが維持されています。改善活動は、アジャイル開発の中でもXPの思想を基に活動を行っています。

1.6 主要な課題と改善のフォーカス

主要な課題を表1にまとめました。

表1 課題

課題		状況
a	仕様とコード、文書のレビューによる他者のフォローがないため、間違いが多発	<ul style="list-style-type: none"> ・ソフトウェア開発は、人への依存度が高いため、間違っ て当然 ・考え方や理解度が見えないため、物事の解釈が人それ ぞれ ・ハードウェアに比べて、ソフトウェアはオートメーショ ン技術が貧弱で、ほぼ手作業となる ・心や体の調子や意識によって、勘違いや誤解が発生する ・独りでの開発のため、上記のようなヒューマンエラー による簡単な間違いに気付かない
b	実装完了時点で品質保証が行われていないため、検証工程で手戻り多発	<ul style="list-style-type: none"> ・不具合への対応は、下流工程になるほど、高コスト ・プロジェクト全体で、最小でも0.25(人日/件)*のコス トを要する ・検証すればするほど不具合が発生するため、プロジェ クトメンバー全員が疲弊してしまう
c	形式知になっていないため、プロジェクトメンバーを助けられない・助けてもらえない	<ul style="list-style-type: none"> ・独りでの開発は、プレッシャーが大きくなっていき、 心を壊してしまう ・チームワークが悪くなり、効率が低下する

※は、OpenMG Audio Client SDK for Windows 開発プロジェクトにおける計測結果

SPCでは、上記課題を解決するために、表2のような対策を実施しました

① 対策a フィードバックを高速化する

- ・課題a・bに有効
- ・XP実装によって実現する
- ・テスト駆動開発、コード修正ごとのUnit Test、メンバーによるレビュー、毎晩実行のAuto Test、リリース前検証によって、1つのイテレーション（短期リリース）の中で、仕様・コードの不具合検出の機会を大幅に増加することで、フィードバックを高速化する

② 対策b 設計時点から品質を考慮し、検証工程での不具合指摘を減少させる

- ・課題bに有効

- ・XP実装によって実現する
 - ・対策aによりリリース前に不具合検出が行われ、検証工程前に不具合改修を行う
- ③ 対策c 暗黙知を形式知化し、タスク・ソースコード・問題をメンバ間で共有する
- ・課題cに有効
 - ・XP実装、Iterative Postmortemによって実現する
 - ・XP実装のメンバによるレビューによって、仕様やコードの知識を共有する。また、Iterative Postmortemによって、プロジェクトの課題や対策のためのタスクを共有する

1.7 顕在化した改善効果

1.7.1 全体的な効果

全体的な効果を図1に示します。

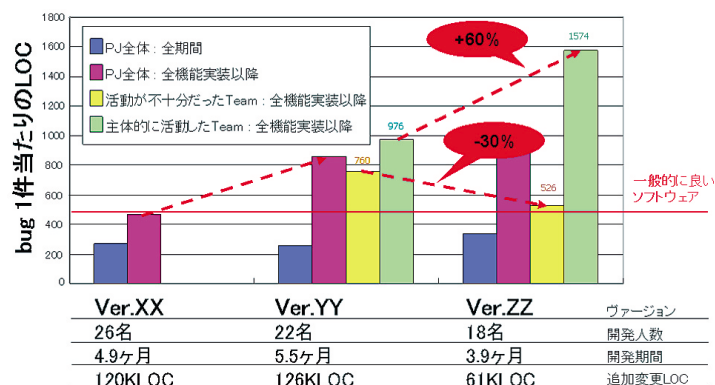


図1 全体的な効果

活動を推進したことで、「1.2 達成目標」で記載した目標は達成されました。

検証工程前に設計内検証を行うことによって、全体として品質が60%向上しました。改善活動前 (Ver.XX) と比べ、改善活動後 (Ver.YY / Ver.ZZ) では、品質が向上しているのが確認できます。また、改善活動が不十分だったチームと比較しますと、主体的に活動したチームは、継続した品質向上が実現しています。

このグラフでは表現できていませんが、プロジェクトメンバ間での知識の共有が行われることで、助け合いやプロジェクトの課題改善を行える体質へと変化しました。

1.7.2 各課題に対する効果

(1) 課題a (レビューによる他者のフォローがないため、間違いが多発)への効果

- ・レビューによる不具合検出率1.92 (件/時)

活動開始時点の測定値は存在していません。一般的な値として、約0.29 (件/時) (IBM Santa Teresa 研究所調べ, “ピアレビュー” Karl E. Wieggers 著) があります。この一般値は、コードインスペクションとして測定された結果のため、SPCが行っているピアレビューと純粋な比較はできませんが、高速フィードバックという点ではピアレビューが有効であることを確認できました。

- ・プロジェクトメンバ同士で相談ができる状態になった
- ・チェックし合う体質になった

(2) 課題b (実装完了時点で品質保証が行われていないため、検証工程で手戻り多発)への効果

- ・60%品質向上 (品質 = 変更LOC / 検証工程で指摘される不具合数)
- ・検証工程にて指摘される不具合が大きく減少
- ・設計時の品質作り込みのできる体質実現

(3) 課題c (形式知になっていないため、プロジェクトメンバを助けられない・助けてもらえない)への効果

- ・タスクとソースコードを、担当者固定ではなく、プロジェクトメンバで共有できるようになった
- ・休暇を取得しても、プロジェクトメンバ間で、他者のタスクを拾い合える体質になった

2. ベストプラクティス

2.1 参照モデル上でのカバレッジ

- ① 当該ベストプラクティスに対応する ISO/IEC 15504 第5部のプロセス
 - ・ソフトウェア要求事項分析 (ENG.4 Software requirements analysis)
 - ・ソフトウェア設計 (ENG.5 Software design)
 - ・ソフトウェア構築 (ENG.6 Software construction)
 - ・ソフトウェア試験 (ENG.8 Software testing)
- ② 当該ベストプラクティスに対応する CMMI のプロセス領域
 - ・要件開発 (RD: Requirements Development)
 - ・技術解 (TS: Technical Solution)
 - ・検証 (VER: Verification)
- ③ 当該ベストプラクティスに対応する共通フレーム2007のプロセス/アクティビティ
 - ・開発プロセス/ソフトウェア要件定義
 - ・開発プロセス/ソフトウェア方式設計
 - ・開発プロセス/ソフトウェア詳細設計
 - ・開発プロセス/ソフトウェアコード作成およびテスト
 - ・開発プロセス/ソフトウェア適格性確認テスト
 - ・改善プロセス
 - ・問題解決プロセス

2.2 プラクティスの説明

2.2.1 プラクティス実施タイミング

SPC の事例では、「XP 実装」「Iterative Postmortem」の2つのプラクティスについて説明します。開発プロセスにおけるそれぞれの位置付けは、図2のとおりです。

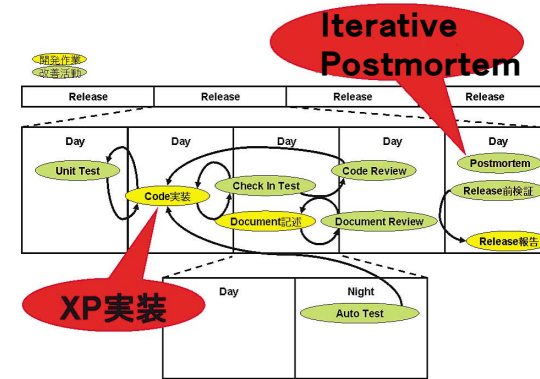


図2 開発プロセスにおける「XP 実装」「Iterative Postmortem」の位置付け

2.2.2 XP 実装

このプラクティスは、『テスト駆動開発』、『Unit Test』、『レビュー』、『リファクタリング』、『自動ビルド』、『自動テスト』、『リリース前検証』をシームレスに結合しています。当初は、1つひとつのプラクティスを個別に実施していましたが、現在では一連の作業として連結しています。

(1) 実施の流れ

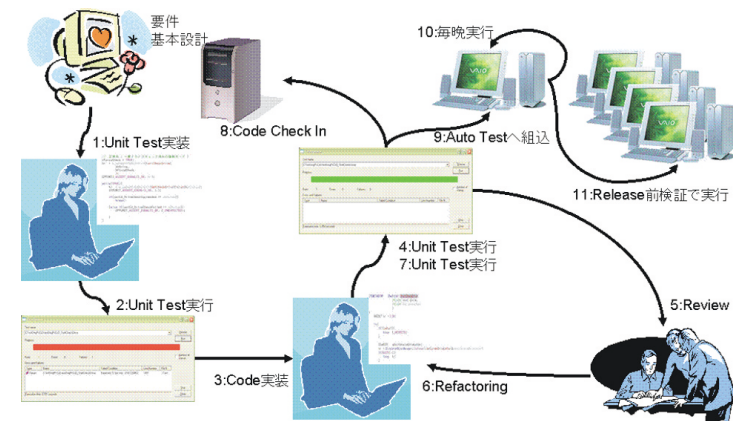


図3 XP 実装の流れ

(2) 実施方法

1 Unit Test 実装

実施項目	<ul style="list-style-type: none"> 要件・仕様から単体テストを実装（成果物となるコードに着手する前に行う） 該当コードの担当者が、実装予定となるコードのためのテストコードを実装する 仕様不具合に気付いた場合は、再度、設計・要求分析を行う
コメント	<ul style="list-style-type: none"> テストコードによって仕様を表現することにより、コード実装前に仕様不具合を検出できる

2 Unit Test 実行

実施項目	<ul style="list-style-type: none"> 「1 Unit Test 実装」で作成したテストを実行し、失敗する事を確認する（成果物となるコードは未だ実装されていないため、失敗することが期待値となる）
コメント	<ul style="list-style-type: none"> 失敗により、テスト自体の正当性を確認する

3 Code 実装

実施項目	<ul style="list-style-type: none"> 仕様を満たすために、成果物となるコードを記述・修正
コメント	<ul style="list-style-type: none"> なし

4 Unit Test 実行

実施項目	<ul style="list-style-type: none"> 「1 Unit Test 実装」で作成したテストを実行し、成果物が正しく動作することを確認する（不具合は、この時点で発見され、直ぐに修正を行う）
コメント	<ul style="list-style-type: none"> 不具合発見から改修までの間に、他者を介在させないため、フィードバックが高速になる（実際には、「3 Code 実装」と「4 Unit Test」を何度も繰り返して実装を行うことになる） 検証工程に進む前の実装工程で、動作が保障されることになる

5 Review

実施項目	<ul style="list-style-type: none"> プロジェクトメンバにて、コード・ドキュメントのレビューを行う レビューの選定は、コード・ドキュメントを書いた開発者が行う レビューは指摘のみを行い、修正方法の検討は、依頼者が検討する
コメント	<ul style="list-style-type: none"> レビューに知識が広まり、形式知化される インスペクションはコストが大きいので、ピアレビューの形式を推奨する レビューミーティングでは、事前準備等なく、その場で確認することになりやすいため、レビュー箇所を指定し、デスクチェックしてもらうと効率が良い

6 Refactoring

実施項目	<ul style="list-style-type: none"> レビューによる指摘項目を、コードに反映する（「5 Review」での指摘内容や解決策が不明な場合には、レビューに相談する）
コメント	<ul style="list-style-type: none"> 人に見られる事を意識するため、読みやすさを考慮したコードとなり、保守性が向上する レビューが読みやすいコードとなるため、他のメンバでも修正が容易になる

7 Unit Test 実行

実施項目	<ul style="list-style-type: none"> 「1 Unit Test 実装」で作成したテストを実行し、リファクタリングによる不具合混入が無い事を確認する（リファクタリング後は、必ずテストを実行する） リファクタリング後のコードレビューを行うために、「5 Review」の作業に戻る 「5 Review」の作業に問題がなければ、「8 Code Check In」の作業に進む
コメント	<ul style="list-style-type: none"> テストが成功することを確認できるまでは、リファクタリング完了とはいえない

8 Code Check In

実施項目	<ul style="list-style-type: none"> コードをレポジトリにチェックインする
コメント	<ul style="list-style-type: none"> テストを通過し、形式知化されたコードが保存され、リリース対象となる 不具合の存在しないコードのみが成果物として結合される

9 Auto Test (自動テスト) へ組み込み

実施項目	<ul style="list-style-type: none"> 「2 Unit Test 実行」の作業で作成した Unit Test を、自動テストとして組み込む
コメント	<ul style="list-style-type: none"> なし

10 毎晩実行

実施項目	<ul style="list-style-type: none"> 自動ビルドされた成果物に対して、Unit Test が毎晩深夜に実行される
コメント	<ul style="list-style-type: none"> 全ての最新コードが統合された成果物が検証されるため、他のメンバの修正による影響も1日以内に検出される

11 Release 前検証で実行

実施項目	<ul style="list-style-type: none"> 成果物をリリースする前にも、自動テストを実行する（他 OS での実行も容易に行える）
コメント	<ul style="list-style-type: none"> 「10 毎晩実行」にて実行されているテストなので、ヒューマンリソースをほぼ使わずに、全機能の検証が実現できる

2.2.3 Iterative Postmortem

SPC では、プロジェクトのレビューを毎週実施します。このプラクティスは、課題に対する対策よりも、もう1レベル上の話になり、常に改善できる体質を構築することを狙っています。

(1) 実施の流れ

XP 実施の流れを図4に示します。

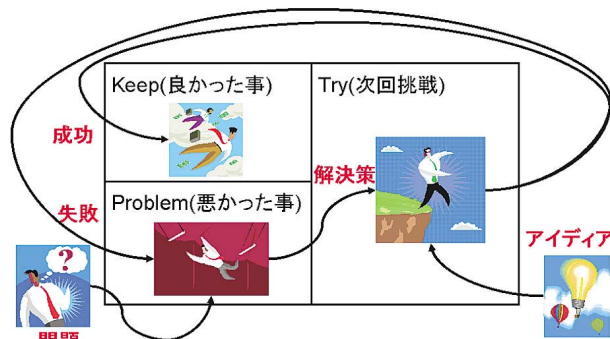


図4 XP 実施の流れ

(2) 実施方法

実施にあたっては、以下をグラドルールとし、下記の1~3を行いました。

- ・毎週30分
- ・全員発表
- ・Try 担当者決定
- ・業務外ネタ OK

1 プロジェクトメンバ全員を集める

実施項目	<ul style="list-style-type: none"> ・プロジェクトメンバ全員の時間を確保する ・集合前までに、今週の Try/Problem/Keep を検討しておいてもらう
コメント	<ul style="list-style-type: none"> ・毎週決まった開始時間・開催場所とすることでリズムを生み出す ・最初は、アイスブレイク等で1人1回発言してもらうと意見が出やすい

2 Keep/Problem/Try を発案してもらう

実施項目	<ul style="list-style-type: none"> ・各自が考えてきた内容を発表してもらう (Problem は、表面的な問題だけを発表するケースが多いので、本質的な問題になるまで質問をする)
コメント	<ul style="list-style-type: none"> ・今週あったイベントやリリースを記述しておくと思い出しやすい ・発表していない人は、ファシリテータがガイドし、挙げてもらう

3 Try 担当者を決める

実施項目	<ul style="list-style-type: none"> ・不平等感が出ないように、1人1個、同じ大きさにしたタスクを割り振る (自主的なサインアップが基本だが、誰も挙手しない場合は、ファシリテータがアサイン)
コメント	<ul style="list-style-type: none"> ・なし

2.2.4 プラクティスに付随した効果・経験

(1) XP 実装

XP 実装の結果について、表3にまとめます。

表3 XP 実装の結果

結果	内容
測定できた効果	<ul style="list-style-type: none"> ・成果物への不具合潜在時間が、1日以内 ・実装時点でテストコードが完成するため、それ以降、テストコード作成の工数が0 (人日) ・全検証完了のためのヒューマンコストが、0.5 (人時/OS)^{*1}
良かった点	<ul style="list-style-type: none"> ・コード実装前に仕様不具合に気付ける ・開発者自身が、コード修正後に容易に実行できるため、リグレッションの不安が激減 ・自動テストにより毎日テストされているため、リリース後の不安がないリファクタリングにより保守性が向上しているため、調査コストが減少 ・レビューにより形式知化できているため、安心して休暇を取得できる
悪かった点	<ul style="list-style-type: none"> ・テストコード作成コストが増加 ・簡単な関数だとテストに無駄感がある
教訓	<ul style="list-style-type: none"> ・テストコードの品質は開発者次第 ・テストコードにもリファクタリングが必要 ・自動テストが有効だとしても、省略できない手動テストが存在 ・納期が短い場合、テストコード実装の工数確保が困難

*1 改善活動前は、全て GUI アプリケーションでの手動テストを行っていたため、18 (人時/OS) となっていた。

(2) Iterative Postmortem

Iterative Postmortem の結果について、表4にまとめます。

表4 Iterative Postmortemの結果

結果	内容
測定できた効果	・毎週、最低3つはプロジェクトの問題が改善される ^{*1}
良かった点	・個人の「Problem」がプロジェクト全体を改善 ・「Try」を決めるので、“愚痴で終わり”にならない ・親睦が深まる
悪かった点	・なし
教訓	・小さいTryは実施される ・硬い雰囲気を実施すると意見が出難い ・短いスパンで開催しないと気付きを忘れてしまう ・個人のKeepは、全体のKeepにはなりにくい

※1 改善活動前は、プロジェクト完了後にのみ実施していたため、プロジェクト前半の問題が挙げられず、かつ、問題の本質を覚えていないため、改善活動につながりにくかった。

2.3 活用しているツール、技法

- ・XP 実装
 CppUnit (Open Source Software)
- ・Iterative Postmortem
 ホワイトボード

2.4 人材育成

- ・改善推進メンバで書籍やWebから情報を収集し、XPについて学習した
- ・使用するプラクティスのみを抽出し、プロジェクトメンバに説明会を実施した
- ・不明点は、実施しながら説明した
- ・プロジェクトメンバ全員で検討し、自分たちに合うようカスタマイズした

3. 学ぶべき点

開発プロセスの改善活動を推進し成果をあげるためには、組織のトップマネジメントの意志とコミットメントが重要であるといわれています。しかしSPCの事例では、トップマネジメントの意志とコミットメント以上に、現場の開発者の「強い疑問と課題認識」を背景にした現場の開発者自身のモチベーションによる改善活動を推進した結果、一定の成果が得られたことが報告されています。

それぞれの組織において改善活動の進め方を検討される際、1つのアプローチとして、参考にしてください。

参考文献

オブジェクト倶楽部 <http://www.objectclub.jp/>

Karl E. Wiegers：ピアレビュー 高品質ソフトウェア開発のために、日経BP社、2004年

遠藤功：見える化（強い企業を作る見える化の仕組み）、東洋経済新聞社、2005年

付録 参考情報： 公開されているプロセス改善事例

注：下記の情報は、本書発行時点のものであり、また変更される場合があります。

名称：SPES（ソフトウェア プロセス エンジニアリング シンポジウム）
主催：JISA（社団法人 情報サービス産業協会）
期間：2003年～
URL：<http://www.jisa.or.jp/seminar/spes2007/announce.html>

名称：SPI Japan（ソフトウェアプロセス改善カンファレンス）
主催：JASPIC（日本 SPI コンソーシアム）
期間：2003年～
URL：<http://www.jaspic.jp/>

名称：JUAS Square（IT ガバナンス）
主催：JUAS（社団法人 日本情報システム・ユーザー協会）
期間：2001年～
URL：<http://www.juas.or.jp/project/evnt/index.html>

名称：SQiP（ソフトウェア品質シンポジウム）
主催：JUSE（財団法人 日本科学技術連盟）
期間：1982年～
URL：<http://www.juse.or.jp/software/symposium.html>

名称：SS（ソフトウェアシンポジウム）
主催：SEA（ソフトウェア技術者協会）
期間：1980年～
URL：<http://www.sea.jp/Events/symposium/ss-index.html>

名称：JaSST（ソフトウェアテストシンポジウム）
主催：ASTER（特定非営利活動法人 ソフトウェアテスト技術振興協会）
期間：2003年～
URL：<http://www.jasst.jp/>

著者／インタビュー一覧

(役職名・敬称略)

各事例の原稿を執筆するにあたっては、以下の皆様のご協力を頂きました。

事例 1 設計からはじめる見逃しバグの防止 ～F2Tの活用～

株式会社日立ハイテクノロジーズ
ナノテクノロジー製品事業本部 研究開発本部
児玉 隆一郎
飯泉 紀子

事例 2 生産性を5年で2.5倍に！ ～一人ひとりを改善の主役にする KAIZEN 塾の設置～

株式会社富士通ソフトウェアテクノロジーズ
NPS 推進室 高木 徹
(現在：富士通株式会社 生産革新本部 SI 生産革新推進統括部
TPS 適用推進担当部長)

事例 3 究極の高品質ソフトウェア開発プロセスを目指して ～ソフトウェア独立検証および妥当性確認 Software IV&V～

独立行政法人 宇宙航空研究開発機構 情報・計算工学センター
ソフトウェアエンジニアリングチーム
片平 真史
静永 誠

事例 4 生産性原理を具現化するプロセス改善 ～生産管理システム「ACTUM」の活用～

株式会社ジャステック
営業本部 太田 忠雄
製造本部 高橋 亨
品質／環境委員会 黒柳 鉄夫

事例 5 高品質ソフトウェア開発管理プロセスの実現 ～QMTXによる品質メトリクスの活用～

日本電気株式会社 システムソフトウェア事業本部
品質保証部
織田 巖
(現在：NEC システムテクノロジー株式会社 プラットフォーム事業本部 PF
統括本部)

事例 6 マネジメント自らが予防のための行動を取り、問題が起りにくい組織に ～Preventive Managementによる実現～

情報技術開発株式会社 開発本部 コンピテンシー部
平山 洋一
秋月 伸晃
大島 道夫

事例 7 派生開発による品質および開発効率の向上 ～XDDP・USDMの徹底～

東京エレクトロンソフトウェア・テクノロジーズ株式会社
本多 慶匡

事例 8 ソフトウェア部品による組立型開発と品質保証 ～基幹システムでのコードの再利用～

住友電気工業株式会社 情報システム部 システム技術グループ
中村 伸裕
中塚 康介
住友電工情報システム株式会社 品質管理推進室
山邊 人美

事例 9 全社レベルでのプロジェクト管理の改善 ～段階的にツールを導入し作業内容変更の最小化をはかる～

日立ソフトウェアエンジニアリング株式会社
技術開発本部 角田 文広
プロジェクトマネジメント統括本部 浅沼 信行

事例 10 GDD(Genba Driven Development) ～現場主導によるアジャイル開発のすゝめ～

ソニー株式会社
PC アプリケーションソフトウェアセンター
設計部
小原 優

用語・略語

- ISO/IEC 15504……………ISO/IEC 15504 Information technology-Process assessment
プロセスアセスメントに関する国際規格群 本書発行時点で以下の
部が発行されている。対応する JIS 規格群 (JIS X 0145) も順次発
行予定。
- Part 1 : Concept and vocabulary
 - Part 2 : Performing an assessment
 - Part 3 : Guidance on performing an assessment
 - Part 4 : Guidance on use for process improvement and pro-
cess capability determination
 - Part 5 : An exemplar Process Assessment Model
- CMM……………Capability Maturity Model for Software
CMMI……………Capability Maturity Model Integration
IEC……………International Electrotechnical Commission
ISO……………International Organization for Standardization
IV&V……………Independent Verification and Validation (独立検証および妥当性)
PM……………Project Manager
PMO……………Project Management Office
SEI……………Software Engineering Institute (米国カーネギーメロン大学・ソフ
トウェア工学研究所)
SEPG……………Software Process Engineering Group
SLCP……………Software Life Cycle Processes
SPICE……………Software Process Improvement and Capability Determination
SQA……………Software Quality Assurance (ソフトウェア品質保証)
SWIV&V……………Software Independent Verification and Validation (ソフトウェア独
立検証および妥当性)
WBS……………Work Breakdown Structure (作業細分化構造)
XP……………Extreme Programming
共通フレーム2007……………ユーザ・ベンダのソフトウェア取引の透過性の確保、発注者が行う
べき作業の可視性、開発者・運営者が行うべき作業の可視化のため
の共通規範 (書籍：独立行政法人 情報処理推進機構 ソフトウェ
ア・エンジニアリング・センター編, 共通フレーム2007～経営者,
業務部門が参画するシステム開発および取引のために～, 2007,
オーム社)。

参考文献

- カーネギーメロン大学・ソフトウェア工学研究所：開発のための CMMI®1.2版, 2007年
- 独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター編：共通
フレーム2007, オーム社, 2007年
- ISO/IEC 15504- 5 : 2006 : Information technology - Process assessment - Part 5 : An exem-
plar Process Assessment Model
- JIS X 0160 : 2007 : ソフトウェアライフサイクルプロセス
注) 対応国際規格：ISO/IEC 12207 : 1995, Information technology - Software life
cycle processes, Amendment 1 (2002) 及び Amendment 2 (2004)

索引

あ 行

アジャイル開発	182
アセスメントモデル	1
一列待ち	50
一個流し	50
落穂拾い	34

か 行

過去データ	102
環境変数	72
カンバン方式	40
管理的独立	62
技術的独立	62
形式知	182
欠陥フロー図	163
検証	62
顧客満足	102
コミットメント	110

さ 行

マイクロマティック複雑度	146, 158
財政的独立	62
システム・リファレンス・マニュアル	85
システム総合テスト仕様書	22
システム方式設計	22
自動化	40
ジャストインタイム	40
全社標準化	181
ソフトウェア方式設計	22
ソフトウェア保守	133
ソフトウェアメトリクス	110

ソフトウェア結合テスト仕様書	22
----------------	----

た 行

妥当性確認	61, 62
定量管理	102
データ中心設計	151
テーラリング	4
テスト	141
伝道師	37
トヨタ生産方式	36

な 行

日本情報システム・ユーザー協会	85
能力成熟度モデル	1

は 行

バグ	111
標準プロセス	3
品質会計制度	98
品質システム	112
プロセス改善	I, 1
プロセス資産	1
プロセス資産リポジトリ	7
平準化	50
ベストプラクティス	I, 8

ま, や, ら 行

ムダ取り	36
ユーザニーズ	100
リポジトリ	1

数 字

3 現主義	42
5 S	39

アルファベット

ACTUM	72
CMMI	1, 11
EM/PFM	54
EPM	168
F2T	22
ISO/IEC 15504 第5部	11
Iterative Postmortem	183
ITSS	96
IV&V	54, 61
JUAS	85
KAIZEN 塾	36
MISRA-C	54

NPS	36
NPS 活動	36
PAT	15
PM	168
PMO	168
Preventive Management	114
QA	24
QMTX	98, 104
S4S	59
SEPG	91
SPICE for Space	59
SQA	54
SRM	85
SW IV&V	61
Try/Problem/Keep	194
T 字形 ER 手法	146
USDM	138
XP	182

執筆者：経済産業省 プロセス改善研究部会 WG2

リーダー	込山 俊博	日本電気株式会社
	穴田 直也	株式会社大和コンピューター
	小川 清	名古屋市工業研究所
	北島 義弘	株式会社 PM Academy
	河野 文昭	株式会社アドヴィックス
	田中 一夫	アイエックス・ナレッジ株式会社
	徳永 享	富士ゼロックス株式会社
	服部 祐二	ブラザー工業株式会社
	和田 典子	ソニー株式会社

経済産業省 プロセス改善研究部会メンバ

主査	菊島 靖弘	東京海上日動システムズ株式会社／株式会社アイネス
副主査	新谷 勝利	独立行政法人 情報処理推進機構
	赤坂 幸彦	株式会社 NTT データ
	安達 賢二	株式会社 HBA
	足立 久美	株式会社デンソー
	穴田 直也	株式会社大和コンピューター
	岩佐 洋司	住友電工情報システム株式会社
	江崎 美保	株式会社日新システムズ
	小川 清	名古屋市工業研究所
	片平 真史	宇宙航空研究開発機構
	北島 義弘	株式会社 PM Academy
	北野 敏明	独立行政法人 情報処理推進機構／新日鉄ソリューションズ株式会社
	小泉 浩	マイクロソフト株式会社
	河野 文昭	株式会社アドヴィックス
	込山 俊博	日本電気株式会社
	近藤 聖久	三菱電機株式会社
	佐藤 健哉	同志社大学
	砂塚 利彦	砂塚コンサルティングサービス株式会社
	田中 一夫	アイエックス・ナレッジ株式会社
	谷川 浩	トヨタ自動車株式会社
	徳永 享	富士ゼロックス株式会社
	中村 伸裕	住友電気工業株式会社
	服部 祐二	ブラザー工業株式会社
	伏見 諭	株式会社情報数理研究所
	堀田 勝美	株式会社コンピータジャパン
	松原 友夫	松原コンサルティング
	和田 典子	ソニー株式会社
	石谷 靖	独立行政法人 情報処理推進機構／株式会社三菱総合研究所

編 者 紹 介

独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター
2004年10月に独立行政法人 情報処理推進機構 (IPA) 内に設立されたソフトウェア・エンジニアリング・センター (SEC) は、エンタプライズ系ソフトウェアと組込みソフトウェアの開発力強化に取り組むとともに、その成果を実践・検証するための実践ソフトウェア開発プロジェクトを産学官の枠組みを越えて展開している。

[所在地] 〒113-6591 東京都文京区本駒込2-28-8
文京グリーンコート センターオフィス
電話 03-5978-7543, FAX 03-5978-7517
<http://sec.ipa.go.jp/index.php>

- 本書の内容に関する質問は、オーム社雑誌部「(書名を明記)」係宛、書状またはFAX (03-3293-6889)にてお願いします。お受けできる質問は本書で紹介した内容に限らせていただきます。なお、電話での質問にはお答えできませんので、あらかじめご了承ください。
 - 万一、落丁・乱丁の場合は、送料当社負担でお取替えいたします。当社販売管理課宛お送りください。
 - 本書の一部の複写複製を希望される場合は、本書扉裏を参照してください。
- JCLS** <(株)日本著作出版権管理システム委託出版物>

SEC BOOKS

プロセス改善ナビゲーションガイド

～ベストプラクティス編～

平成 20 年 2 月 12 日 第 1 版第 1 刷発行

編 者 独立行政法人 情報処理推進機構
ソフトウェア・エンジニアリング・センター

発 行 者 佐 藤 政 次

発 行 所 株式会社 オ ー ム 社

郵便番号 101-8460

東京都千代田区神田錦町 3-1

電 話 03 (3233) 0641(代表)

URL <http://www.ohmsha.co.jp/>

© 独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター 2008

印刷・製本 報光社

ISBN 978-4-274-50175-3 Printed in Japan

プロセス改善ナビゲーションガイド

～なぜなに編～

A 5 判・124頁

プロセス改善ナビゲーションガイド

～プロセス診断活用編～

A 5 判・160頁

共通フレーム 2007

～経営者、業務部門が参画するシステム開発および取引のために～

B 5 変形判・320頁

ソフトウェア改良開発見積りガイドブック

～既存システムがある場合の開発～

A 5 判・160頁

ソフトウェア開発見積りガイドブック

～IT ユーザとベンダにおける定量的見積りの実現～

A 5 判・232頁

経営者が参画する要求品質の確保

～超上流から攻める IT 化の勘どころ～

第 2 版

A 5 判・128頁・CD-ROM 付き

※【事例検索システム】URL : <https://sec.ipa.go.jp/enterprise/index.php>

SEC journal

A 4 判 (年 4 回発行)

もっと詳しい情報をお届けできます。

◎書店に商品がない場合または御注文の場合も
右記宛にご連絡ください。



ホームページ

<http://www.ohmsha.co.jp/>

TEL/FAX

TEL.03-3233-0643 FAX.03-3293-6224