

NECにおける定量的管理

～ソフトウェア品質会計とNECデータ白書に基づく改善活動の展開～

日本電気株式会社 宮崎 義昭 日本電気株式会社 誉田 直美

弊社ではソフトウェア開発全域において、長年にわたり、品質会計という標準的な品質管理手法を適用し、高品質なソフトウェア開発に努めてきた。また、開発途中で測定した膨大なプロジェクトデータを、グループ各社を含む全体規模で共通的に収集し、分析を加えた結果をデータ白書としてまとめ毎年発行している。

この分析から、上工程重視により品質向上と生産性向上が両立可能であること、CMMI成熟度レベルを向上することにより品質向上できることなど、多くの知見を得ることができた。更に、これらの知見を基に、組織的なプロセス改善のPDCAサイクルを回すことにより、ソフトウェア開発の品質と生産性の向上に取り組んでいる。本稿では、これまでの活動の概要と成果及び今後に向けた展開について報告する。

1 はじめに

NEC及びNECグループ各社では、高品質ソフトウェアを開発するための品質管理技法である「ソフトウェア品質会計」の適用を基盤とし、ソフトウェア品質の改善に向けた様々な取り組みを継続的に行っている。ソフトウェア開発途中で収集しているプロジェクトデータをグループ横断的に収集し、統計的な分析と考察を加えた結果を各組織にフィードバックすることで、データに基づくプロセス改善活動を全社的に展開している。これは、一部の事業部門において以前から行っていた活動を、2011年度から、グループ全体に対象範囲を広げ発展させてきたものである。現在では、グループ全体の開発量のうち、8割を超えるプロジェクトデータを収集するに至っている。

2 ソフトウェア品質会計

2.1 ソフトウェア品質会計の位置付け

このデータに基づくプロセス改善活動は、「ソフトウェア品質会計」(以降、品質会計)という弊社独自のソフトウェア開発の品質管理技法が基盤となっている [Honda2010]。品質会計は、グループ全体で標準的な品質管理技法として適用しており、この共通基盤があるからこそ、広範囲の事業領域において横断的な視点での取り組みが可能となったと考える。

2.2 品質会計の概要

品質会計は、基本設計からコーディングまでの工程(以降、上工程)で作込まれたバグを負債と考え、これをレビュー及びテストにおいて摘出/修正することで負債を返済し、残存バグがなくなり負債がゼロになった時点で出荷する、という考え方に基づく。その特徴は、上工程のレビューを充実させ、早い段階でより多くのバグを摘出する「上工程重視」と、開発途中の品質状況や実際に摘出したバグの分析により的確に「テスト終了判断」を行うことにある。

品質会計を構成する基本的な技術体系を図1に、また各技法の使用法と特徴を表1に示す。「バグ目標管理技法」は、開発途中のレビューやテストで摘出されるバグ数を目標管理するための技法であり、「上工程品質会計」「テスト工程品質会計」から成

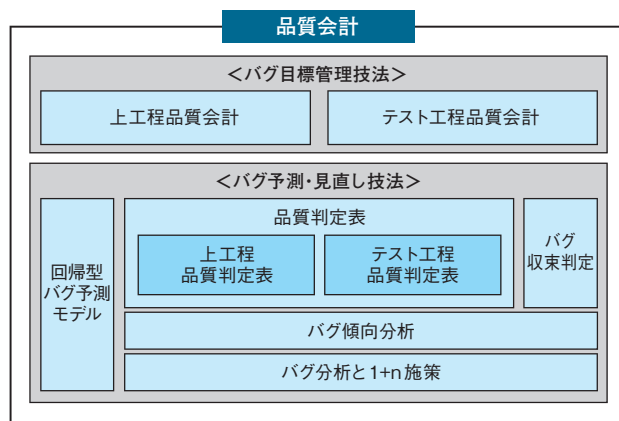


図1 品質会計の技術体系

る。「バグ予測・見直し技法」は、開発状況に応じて残りのバグ件数を予測し適宜見直しを行う技法であり、「回帰型バグ予測モデル」「品質判定表」「バグ傾向分析」「バグ分析と1+n施策」「バグ収束判定」の5つから成る。

表1 品質会計を構成する技法とその特徴

カテゴリ	技法	使用法と特徴
バグ目標管理技法	上工程品質会計	上工程（設計～製造）用のバグ目標管理技法。バグの抽出工程と作り込み工程の両面からバグを目標管理する。
	テスト工程品質会計	テスト工程用のバグ目標管理技法。テスト開始時に残存する、プログラム全体の総バグ数を目標管理する。
バグ予測・見直し技法	回帰型バグ予測モデル	開発開始時に、今回の開発で作り込むであろう総バグ数を予測するためのバグ予測技法。
	品質判定表	開発途中に発生する変化を考慮して、バグ目標値を見直すバグ目標値見直し技法。上工程品質判定表とテスト工程品質判定表がある。
	バグ傾向分析	抽出したバグを様々な観点から整理することにより、バグの抽出傾向に偏りが無いかを分析する技法。
	バグ分析と1+n施策	バグ1件ごとに真因を分析することにより、開発上の細かい抜け・漏れを発見し、その抜け・漏れに対して、集中的なレビューやテストにより残存するバグを抽出する技法。バグ分析と1+n施策はセットで用いる。
	バグ収束判定	テスト度合いに対する累積抽出バグ数の推移により、バグ収束を判定する技法。

3 プロジェクトデータの収集と分析

3.1 収集データ項目の標準化

組織横断的なデータの収集と分析には、収集するデータ項目の標準化が不可欠である。データ項目は、開発するソフトウェアの規模や種類、工数やバグ数などの様々な管理項目に対する計画値と実績値に加え、統計的な分析に利用するための、組織名や業種などのプロジェクトに付随する属性情報、更に、開発が完了し出荷後に抽出されたバグ数などが含まれる。これらのデータ項目に対して、その意味と具体的な測定方法を定義することで、組織によるばらつきを最小限に抑えるようにしている。しかし、各事業領域によってソフトウェアの種類、開発方法、更には歴史的な経緯も様々であり、同じ名前のデータ項目が異なる意味で使われている例や、一部のデータ項目が収集されていない例もあり、データの精度には少なからず課題があるのも事実である。この点についても、継続的に共通化の方向に向けて改善を進めている。

3.2 データの収集と分析

各組織が管理するプロジェクトデータは、後述する全社的なデータ白書の作成に合わせて年1回の周期で収集し、多角的な視点から分析する。なお、ここで収集したプロジェクトデータの一部は、IPA/SECが発行するデータ白書用に提供させていた

3.3 NECデータ白書の発行

分析の結果は、年次レポートである「NECグループ版ソフトウェア開発データ白書」として発行し、グループ内に公開している。データ白書を作成する主要な目的は二つある。一つ目は定点観測である。どれだけの規模のソフトウェアをどのように開発しているのか、この情報を毎年蓄積することで、過去と比較して何が変わり、何が変わらないかが見えるようになる。この結果は、ソフトウェアを取り巻くビジネス環境が急速に変化する中にあって、それぞれの事業体やグループ全体の視点での経営判断にも重要な情報となる。

二つ目の目的は、ソフトウェア開発の品質／生産性に影響を与える要因を分析し、今後のプロセス改善の方針を考える材料を提供することである。収集したデータを成功プロジェクトと失敗プロジェクトの二つの層に分け、成功と失敗に大きく影響する要因を分析する。グループ全体のデータを対象にすることにより、まとまったデータ件数を確保できるため、事業部単位では実施しにくい細かな層別による傾向分析が可能となる。毎年、新たな視点を加えながら分析と考察を行いメッセージとして発信しており、これらの分析結果は、全社基準値などへ反映することにより、組織的なプロセス改善活動に活用している。

3.4 データ白書による効果

弊社において、データに基づくプロセス改善活動が、取り組み直後から順調に進んだかと言うと必ずしもそうではない。当初は、組織横断的にプロジェクトデータを収集し分析することのメリットが十分に理解されず、データ収集や提供に消極的な組織も存在した。弊社の事業領域は、特定の顧客向けにカスタム開発した業務システム、サーバのOSやミドルウェアのような汎用ソフトウェア製品、更に、特定の機器向けの組込みソフトウェアなど、非常に広範囲にわたっている。事業領域が違えば、そのソフトウェアを開発するための開発プロセスや開発技術に違いが出るのは当然と言える。そのような条件の中で、データに基づく改善活動がグループ全体の取り組みとして定着した要因としては、ソフトウェア開発における品質を作り込むポイントには、ソフトウェアの種類によらず共通な面が多く、大量のデータを集め適切に統計的な分析を行えば、有益な情報が得られることが徐々に認知されてきたことが大きい。分析の結果から、データのばらつきの許容範囲や、複数のデータの関係が数式で示されることによって、これまで経験と勘に頼っていたノウハウが見える化され、基準値として設定したり、客観的な根拠を持った説明が可能になるなどの効果が現れてきた。

以下に、これまでのデータ白書の分析から得られた知見や、データに基づくプロセス改善活動の成果の一部を紹介する。

4 データ分析により得られた知見

4.1 上工程重視による品質向上と生産性向上の両立

ソフトウェアエンジニアリングにおいて、上流工程における品質確保の重要性は古くから言われている。品質会計では、上工程で作り込む全バグ数に対する、上工程終了までにレビューで摘出するバグ数の割合を上工程バグ摘出率と定義し、上工程バグ摘出率80%を目標とすることを推奨している。しかしながら、この目標の達成は、レビューに時間がかかるためにコストが増加し、品質が良くなっても生産性が悪化するのではないかと懸念の声が多かった。これに対して、上工程バグ摘出率が高いほうが品質が良いだけでなく生産性も高くなることを実証し、上工程バグ摘出率80%を達成することの重要性を示すことができた[Maruyama2017]。具体的には以下の通りである。

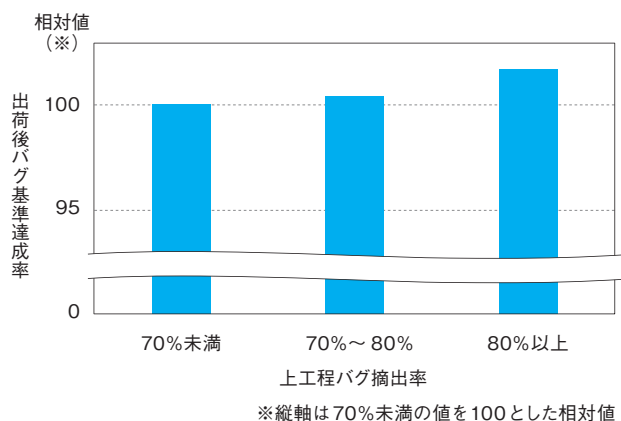


図2 上工程バグ摘出率と出荷後バグ基準達成率

図2は、データを上工程バグ摘出率70%未満、70~80%、80%以上の3つの層に分け、出荷後バグ基準達成率を比較したグラフである。出荷後バグ基準とは、出荷後12カ月に顧客で発生するバグ数を開発規模で除した数値の目標値である。この出荷後バグ基準を達成したプロジェクトの比率を、出荷後バグ基準達成率と呼ぶ。図2によると、上工程バグ摘出率が高いほど、出荷後バグ基準達成率は高くなり、品質が良くなることが分かる。更に、上工程バグ摘出率が高いほど、単位規模当たりの開発工数は減少し、生産性は高くなる(図3)。その理由は、後戻り工数の削減にある。上工程バグ摘出率で層別した3つの層を比較すると、単位規模当たりのテスト項目数には差は見られないことが分析の結果から得られている。一方で、テスト工程で摘出されたバグ数は、上工程バグ摘出率が高くなるほど減少し(図4)、同様にテスト工程の工数も減少している(図3)。これは、上工程でより多くのバグを摘出することにより、テスト開始時点に

潜在するバグ数が減少するため、同等のテスト項目を実施しても、バグ修正のための後戻り工数が削減されたと考えられる。これが、生産性向上につながったのである。弊社では、これらの結果をもとに、上工程バグ摘出率に対して各組織で具体的な目標値を設定し、実績を評価する取り組みを全社的に推進している。

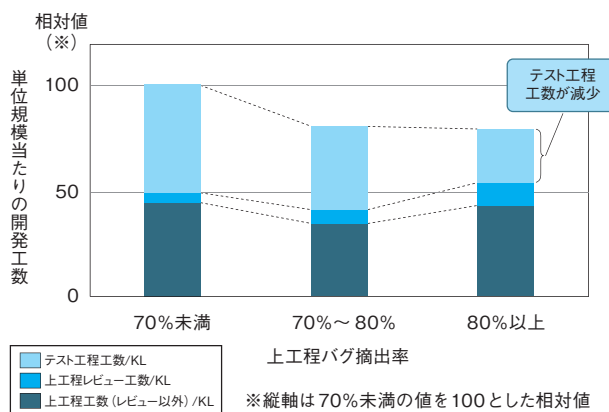


図3 上工程バグ摘出率と開発工数

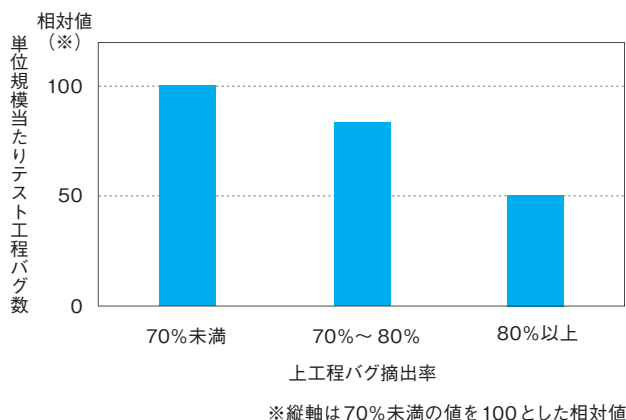


図4 上工程バグ摘出率とテスト工程バグ数

4.2 CMMI成熟度レベルと品質

CMMIの成熟度レベルが上がると各組織が開発するソフトウェアの品質が良くなることは、過去の先行研究においても示されており、これを弊社のデータで検証し、成熟度レベルが上がると、プロジェクトの特性によらず出荷後バグ基準達成率が高くなるという結果を得た[Yanagida2015]。

更に、プロジェクトの成否につながる具体的な要因を明らかにする目的で詳細な分析を行った[Yanagida2017]。成熟度レベル別に分類木を構築した上で、有意差検定及び相関分析を組み合わせて、出荷後バグ基準に対する達成と未達成に影響を与える要因を複合的に分析した。その結果、分類木において、成熟度レベル1と成熟度レベル2では共に開発規模が最も影響が大きい要因であり、達成と未達成を分ける開発規模の境界値は、成熟度レベル2は成熟度レベル1の4倍の値となった。これは、成熟

度レベル1と2の組織は開発規模が大きくなるとプロジェクトを制御できなくなることで、成熟度レベル2の組織は、レベル1と比べて4倍の規模のプロジェクトまでは制御できることを示している。また、レベル3になると、開発規模は影響要因ではなくなる。以上の考察から、成熟度レベルの向上は安定したソフトウェア開発に必須と判断した。

上記の分析結果に基づき、弊社では、各組織の成熟度レベルに応じて、遂行中のプロジェクトにおける品質確保の施策と、組織の成熟度レベルを上げることの両面で改善計画を立て、推進する取り組みを行っている。

4.3 自律的なプロセス改善活動の推進

次に、各事業部が自律的に改善活動を進めるための取り組みについて紹介する。2年前から、各事業部において自組織のプロジェクトデータを自ら分析し、改善計画へ反映する取り組みを行っている。これは、グループ全体のデータと自組織データの比較分析を行うことで、グループ内における自組織の位置付けや課題を客観的に把握してもらうことを目的としている。また、この施策を実行できるよう、各事業部に統計分析ができる人材を育成するための教育プログラムを考案し実施している。これにより、各事業部の改善活動に、定量的な目標設定と実績確認のPDCAサイクルを回す仕組みを組み込むことが可能になった。

5 今後に向けた取り組み

5.1 対象領域の拡大

本稿では、ウォーターフォールモデルによるソフトウェア開発プロジェクトを対象にした改善の取り組みについて説明した。現在では、プロジェクトデータの収集と分析の対象範囲をアジャイル開発や、プラットフォーム構築^{*1}のプロジェクトまで拡大している [Honda2016] [Haraguchi2016]。これらの新たな領域に対しても、ウォーターフォールモデル開発と同様に、プロジェクトを特徴付ける品質指標データ項目を定義し、プロジェクトの成否との関係や、遂行途中に管理すべき項目について分析と検討を進めている。今後、データの蓄積が進むことで、新たな領

域でも、組織的な改善サイクルが軌道に乗るものと期待している。

5.2 分析から得られた知見の有効活用

本節では、データ白書を中心とした改善活動から得られた知見を応用した二つの取り組みを紹介する。

一つは、過去データを利用したシステムの提供である。このシステムは、過去データの分析から得られた成功プロジェクトに基づく見積もり機能を備えており、現在進行中のプロジェクトを過去データの中にプロットすることで、自プロジェクトの立ち位置の確認が可能である。また、事業領域ごとの統計値の表示など、収集した過去データを直接利用できるような工夫し、単にデータを参照できるだけでなく現場で有効に利用することを目指したものである。

もう一つは、データ分析で得られた知見を徹底的に利用し、高品質ソフトウェアを効率的に開発できるよう支援するプロジェクト管理システムの提供である。品質会計技法に基づく品質作り込み状況の自動判定、開発途中ででき上がったソースコードの複雑度やネストの深さなどによる品質自動判定など、開発途中でリアルタイムに収集した品質指標データに基づく問題の有無をシステムが自動判定し警告を出す仕組みである。更に、今まで収集した膨大なデータを使用し、AI(人工知能)によるプロジェクト成否の予測自動判定の実証実験を行っている。AIによる品質予測は非常に高い精度を示しており、本システムによる効果向上に寄与するものと期待している。

これらの取り組みは、いずれも、これまでのデータ収集と分析の蓄積があって初めて可能になるものである。

6 まとめ

組織的な改善活動を継続的に行うためには、活動の成果を常に定量的に管理できるようにすることが必要不可欠であり、今後も、ソフトウェア開発組織としての更なるレベルアップに向けた全グループ横断での活動を推進する。

^{*1} アプリケーションを動作させるために必要なハードウェア(サーバ、ストレージ、ネットワークなど)、OS、ミドルウェアを含むICT基盤の設計及び構築

参考文献

- [Honda2010] 菅田直美、ソフトウェア品質会計、日科技連、2010
- [Maruyama2017] 丸山志保、「レビュー重視と品質・生産性の関係分析」、ソフトウェア品質シンポジウム2017、2017
- [Yanagida2015] 柳田礼子、「効率的な品質改善に向けたCMMI成熟度レベル別の要因分析」、ソフトウェア品質シンポジウム2015、2015
- [Yanagida2017] 柳田礼子、野中誠、菅田直美、「CMMI成熟度レベル別に見たソフトウェア品質の良否にかかわる要因」、SECジャーナル No.49、2017
- [Honda2016] 菅田直美、「アジャイルと品質会計」、情報処理学会デジタルプラクティス Vol.7 No.3、2016
- [Haraguchi2016] 原口剛、「Approach to Build Prediction Model for System platform development projects」、ProMAC2016、2016