

ソフトウェア開発プロジェクトにおける工数の構成分析

- 適切な管理工数とは -

Composition analysis of effort in software development project

- Appropriate management effort -

日本電気株式会社 ソフトウェアエンジニアリング本部

NEC Corporation, Software Engineering Division

○齊藤 拓也

柳田 礼子¹⁾

○Takuya Saito

Reiko Yanagida¹⁾

Abstract: In software development, effort is one of high interest items. According to the white paper on software development data issued by IPA/SEC, it is known that effort increases according to the development volume. But, effort includes direct effort to software development as well as indirect effort such as management effort and effort for customer response. In practice, management effort is often estimated as some percentage of development effort. Also, effort for customer response is frequently estimated separately from development effort. In this research, we analyze the composition of effort in software development projects. In software development projects, management effort is required. Therefore, we examine how much appropriate management effort is.

1. はじめに

ソフトウェア開発において、工数は関心が高い事項の一つである。IPA/SEC 発行のソフトウェア開発データ白書^[1]によると、工数は規模に応じて増加することがわかっている。しかし、工数にはソフトウェア開発に直接関係する工数以外に、管理工数や顧客対応工数といった間接的な工数も含まれる。これらは、すべて規模に直接関係するのであろうか。実務上では、経験値から、開発工数の何割かを管理工数とし、開発工数に加算して全体の工数を見積られるケースが多い。また、移行・展開やユーザ教育といった顧客対応工数は開発工数と別に見積られるケースが多い。

弊社では、工数を開発工数、管理工数、顧客対応工数（定義は表 1 参照）に分離し、収集している。そのデータを使った分析事例、SQiP2017「レビュー重視と品質・生産性の関係分析」^[2]では、開発工数と品質の関係を示している。しかしながら、管理工数、顧客対応工数については、適切に分離できているデータが少なく、十分な分析ができなかった。

本論文では、適切に分離できているデータを拡充し、ソフトウェア開発プロジェクトにおける工数の構成について分析を行う。またソフトウェア開発プロジェクトにおいて、管理工数は必要不可欠である。そこで、適切な管理工数はどの程度なのかの考察を行う。

本論文の構成は、2章で分析の前提となる、品質管理技法と開発プロセス、品質の判定方法、分析対象データとデータ項目について説明する。3章で開発規模や各工数間の関係についての分析結果を示し、4章で変数間の関係を踏まえて適切な管理工数の考察を行う。5章に考察をまとめ、6章で今後の課題を述べる。

 日本電気株式会社 ソフトウェアエンジニアリング本部

Software Engineering Division, NEC Corporation

東京都港区芝 4-14-1 Tel: 03-3798-9552 e-mail:t-saitou@ib.jp.nec.com

4-14-1, Shiba, Minato-ku, Tokyo Japan

1) 日本電気株式会社 ソフトウェアエンジニアリング本部 シニアエキスパート
Senior expert, Software Engineering Division, NEC Corporation

【キーワード：】 ソフトウェア開発、工数、開発工数、管理工数、開発規模

2. 分析の準備

2.1 品質管理技法と開発プロセス

はじめに、前提とする品質管理技法と開発プロセスについて説明する。弊社では、ソフトウェア開発の標準的な品質管理技法として品質会計を適用している^[3]。品質会計の前提とする開発プロセスを図1に示す。開発プロセスはV字モデルである。

基本設計からコーディングまでを「上工程」といい、単体テストからシステムテストまでを「テスト工程」という。各工程では、設計・コーディング・テスト・レビューといったソフトウェア開発に直接関係する作業の他に、管理作業や顧客対応作業といった間接的な作業もある。

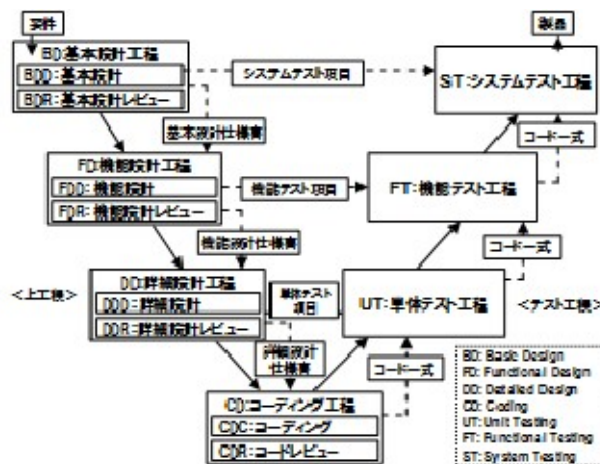


図1 開発プロセス

2.2 品質の判定方法

各プロジェクトが開発したソフトウェアの品質の良し悪しは、出荷後バグ基準に対する出荷後バグ数で判定する。

事業領域によって求められる品質レベルが違うため、弊社では、出荷後バグ基準値を事業領域毎に設定している。出荷後バグ基準値は、以下の式により算出する。

$$\text{出荷後バグ基準値} = \text{出荷後12か月以内の顧客摘出バグ数(件)} / \text{開発規模(KLOC)}$$

出荷後バグ実績値も同様の式で算出する。

出荷後バグ基準値に対する「達成」「未達」は、以下のように判定し、出荷後バグ基準値を「達成」したプロジェクトを「品質が良い」プロジェクトとする。

達成：出荷後バグ実績値 ≤ 出荷後バグ基準値

未達：出荷後バグ実績値 > 出荷後バグ基準値

2.3 分析対象データとデータ項目

本分析では、特定の顧客向けに開発を行う、SI系ソフトウェア開発プロジェクトを対象に分析を行う。プロジェクトデータは全社横断的に収集しており、製造業、通信業、金融業など様々な業種が含まれている。また、使用するプロジェクトデータは、2015年度または2016年度に完了したプロジェクトデータであり、収集時点に、データクレンジングを実施済みである。

表1は、弊社で収集している工数の定義を示したものである。工程ごとに設計製造テスト工数、レビュー工数の他に、管理工数、顧客対応工数を分離して収集している。開発に直接関わる工数である設計製造テスト工数とレビュー工数は、合わせて開発工数として定義している。工数の単位は人Hである。

表2は、弊社で収集している規模の定義を示したものである。新規規模と修正規模は、合わせて開発規模として定義している。規模の単位はKLOC(Kilo Lines of Code)である。

表1 工数の定義

データ項目		単位	定義
開発工数	設計製造テスト工数	人H	設計およびコーディング、テストに費やした工数 バグ対応工数を含む
	レビュー工数	人H	設計およびソースコード、テスト仕様書のレビュー工数
管理工数		人H	各管理作業工数（進捗管理、品質管理など）
顧客対応工数		人H	顧客関連の対応工数（移行・展開、ユーザ教育等）

表 2 規模の定義

データ項目		単位	定義
開発規模	新規規模	KLOC	新たに開発または追加されたソースコードの行数
	修正規模	KLOC	変更されたソースコードの行数

本分析では、上記で示したデータ項目のうち、開発規模、開発工数、管理工数、顧客対応工数を使用する。また、工数の構成を正しく考察できるように、開発規模、開発工数、管理工数、顧客対応工数のすべてが計上されている、292 件のプロジェクトデータを用いて分析する。

3. 分析結果

3.1 開発規模および各工数の回帰分析

開発規模および各工数間の回帰分析の結果を図 2, 3, 4, 6, 7 に示す。回帰分析の図では、青が出荷後バグ基準を「達成」したプロジェクト（以降、達成PJと呼ぶ）を、赤が「未達」のプロジェクト（以降、未達PJと呼ぶ）を示す。実線は達成PJの回帰直線を、破線は未達PJの回帰直線を示す。また、網掛けは回帰直線の95%信頼区間である。

回帰分析における相関は、IPA/SEC 発行のソフトウェア開発データ白書^[1]と同様の基準で判断する。基準を以下に示す。

強い相関： $|\text{相関係数}| \geq 0.85$

やや強い相関： $0.85 > |\text{相関係数}| \geq 0.70$

強い相関は認められない： $0.70 > |\text{相関係数}|$

3.2 開発工数の分析

図 2 は開発規模と開発工数の散布図と回帰直線である。達成PJでは相関係数が 0.76 であり、やや強い相関が認められる。これは開発規模に応じて工数がかかるという、これまでの知見と一致する。一方、未達PJでは相関係数が 0.50 であり、強い相関は認められない。また信頼区間が達成PJよりも広がっており、ばらつきが大きいことが確認できる。このことから、未達PJは開発規模に比した開発工数のかけ方になっていないことがわかる。これは、開発が予定通り進まず開発工数がかかった、もしくは、レビューやテストといった品質を確保するための工数を十分にかけていられずに出荷されたなど、適切に開発工数をかけられていないと考えられる。

3.3 管理工数の分析

図 3「開発規模-管理工数」では、達成PJの相関係数が 0.59 となっており、強い相関は認められない。一方、図 4「開発工数-管理工数」では、相関係数が 0.74 となっており、やや強い相関が認められる。つまり、管理工数は、開発規模より開発工数との相関が強いことが確認できる。このことから、管理工数はソフトウェアの規模（開発規模）に直接関係するのではなく、開発の作業量（開発工数）に応じてかかることがわかる。これは、作業量が増えると管理作業が増えるという、今までの経験に合致すると考える。

未達PJは、図 3「開発規模-管理工数」において、相関係数が 0.44 と開発規模との強い相関が認められない。一方、図 4「開発工数-管理工数」では、相関係数が 0.71 とやや強い相関が認められるが、達成PJに比べて信頼区間が広がっており、ばらつきが大きいことが確認できる。

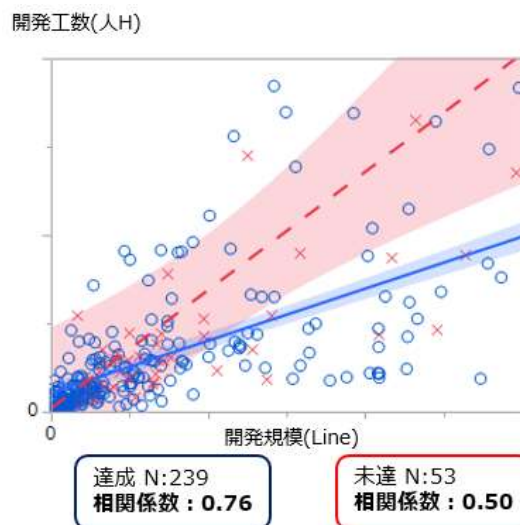


図 2 開発規模-開発工数

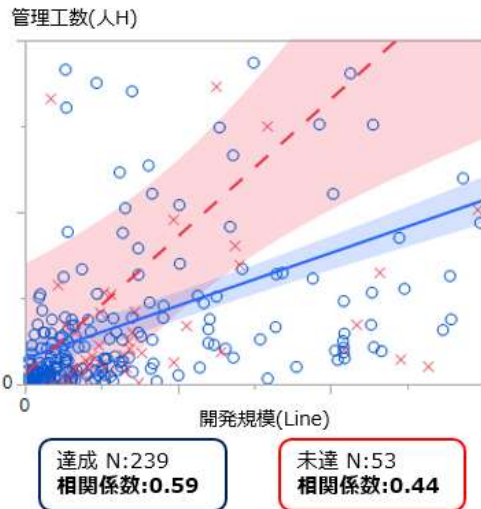


図3 開発規模-管理工数

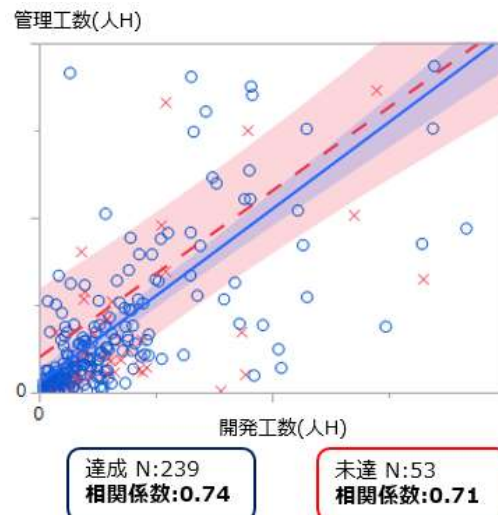


図4 開発工数-管理工数

また、達成PJと未達PJの管理工数のかけ方の違いを確認するために、開発工数に対する管理工数の割合（定義は下記）を定義する。定義式は以下である。

管理工数の割合 = 管理工数(人H) / 開発工数(人H)

図5「管理工数の割合の比較」の箱ひげ図にて、中央値を確認すると、達成PJは0.10、未達PJは0.07となっている。未達PJの方が小さい値となっており、達成PJと未達PJでは、Wilcoxonの順位和検定にて、5%水準で有意差が確認できた。未達PJにおいては、達成PJに比べて開発工数に対する管理工数が少なく、十分に管理されなかったことが品質悪化の一因となったと考えられる。

以上のことから、管理工数は開発工数に応じてかける必要があることがわかる。また未達PJは、ばらつきが大きく、開発工数に対する適切な管理工数がかけられていないと考えられる。

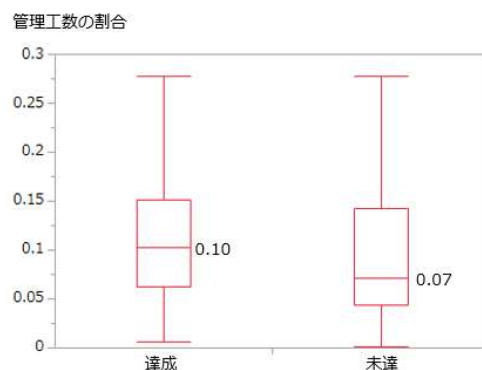


図5 管理工数の割合の比較

3.4 顧客対応工数の分析

図6「開発規模-顧客対応工数」では、達成PJの相関係数は0.42となっている。また、図7「開発工数-顧客対応工数」では、相関係数が0.46となっており、顧客対応工数と開発規模、顧客対応工数と開発工数ともに強い相関が認められない。これは顧客対応工数が、移行・展開、ユーザ教育等の開発以外の付帯作業にかかるため、開発規模もしくは開発工数と直接的に関係するわけではないと考えられる。

未達PJについても、相関係数がそれぞれ0.31と0.56となっており、強い相関が認められず、達成PJと同様の傾向を示す。

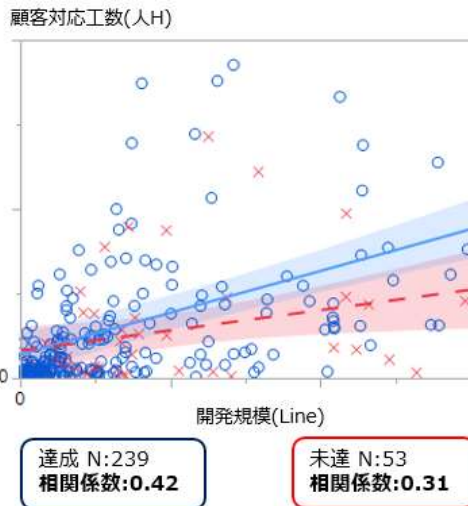


図6 開発規模-顧客対応工数

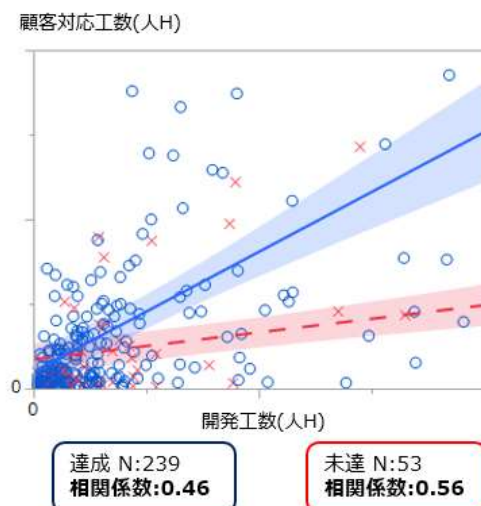


図7 開発工数-顧客対応工数

4. 適切な管理工数

3.3「管理工数の分析」により、管理工数は、開発工数に応じてかける必要があることがわかった。そこで、適切な管理工数を算出するために重回帰分析を行った。目的変数に管理工数を、説明変数に開発規模、開発工数、顧客対応工数を使用し、分析対象は達成PJのみとした。達成PJのみにした理由は、未達PJが、開発工数に対する適切な管理工数がかけられていないためである。

重回帰分析の結果、開発工数および顧客対応工数が、説明変数として5%水準で有意な結果となった。このことから、管理工数は、開発工数と顧客対応工数に応じてかかることがわかる。

重回帰分析の結果、管理工数の予測式は、下記の通りとなった。

$$\text{管理工数(人H)} = 0.094 * \text{開発工数(人H)} + 0.121 * \text{顧客対応工数(人H)}$$

図8は管理工数の予測値と実績値による散布図と回帰直線である。予測値と実績値の相関係数は0.75と、やや強い相関が認められた。

また、予測値と実績値の相対誤差(定義は下記)を「±20%以内」と「+20%超もしくは-20%未満」で層別すると、達成PJの割合が、それぞれ90.5%と79.5%となっており、「±20%以内」の方が達成PJの割合が高いことが確認できる。(表3参照)

予測値と実績値の相対誤差は、以下の式で定義する。

$$\text{予測値と実績値の相対誤差} = (\text{実績値} - \text{予測値}) / \text{予測値}$$

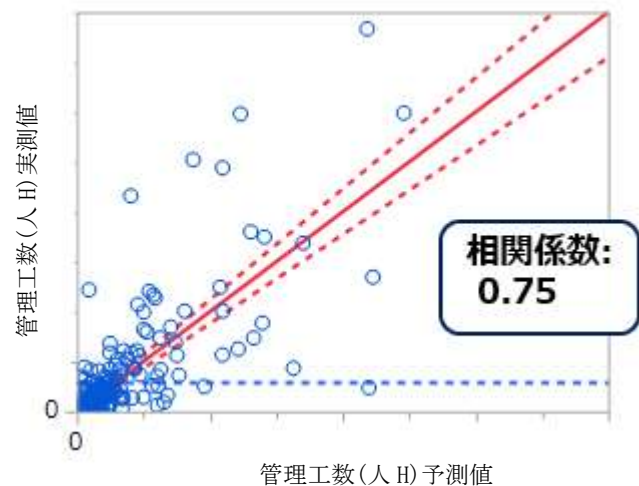


図8 管理工数 予測値-実測値

表3 適切な管理工数と品質

予測値と実測値の相対 誤差	達成PJ		未達成PJ	
	N	%	N	%
±20%以内	57	90.5	6	9.5
+20%超 or -20%未満	182	79.5	47	20.5

上記の結果より、算出された管理工数の予測式が妥当であると判断した。予測式の開発工数の係数は0.094、顧客対応工数の係数は0.121であり、いずれも1割程度になっている。このことから、管理工数は、開発工数および顧客対応工数の1割程度が適切だと考えられる。

5. まとめ

3.2「開発工数の分析」により、達成PJの開発工数は、開発規模に応じてかけていることがわかる。これは、開発規模に応じて工数がかかるということが実証されたと考える。一方、未達成PJは、開発規模に応じた開発工数になっておらず、ばらつきが大きい。未達成PJは、品質確保のために工数を十分につけずに出荷に至った場合や、プロジェクト遂行中に問題が発生して結果的に工数が多くかかった場合など、プロジェクト個々に様々な事情があるものと考えられる。

達成PJの管理工数は、3.3「管理工数の分析」からわかるように、開発工数に応じてかけている。これは、作業量に応じて管理作業が増えるという、今までの経験に合致すると考える。一方、未達成PJは、開発工数に応じた管理工数が適切にかけられていないことが、品質悪化の一因になったと考える。4.「適切な管理工数」により、適切な管理工数は、開発工数と顧客対応工数の1割程度であり、予測値との相対誤差が20%以内から外れると品質が悪化している。これは、必要十分な管理工数をかけて、適切に管理することがプロジェクトの品質確保のために重要であることを示唆している。

3.4「顧客対応工数の分析」により、顧客対応工数は、達成PJ・未達成PJともに、開発規模や開発工数に応じてかかるわけではないことがわかる。これは、顧客対応工数が、移行・展開、ユーザ教育等の開発以外の付帯作業にかかるためと考える。

6. 今後の課題

本論文では、主に達成PJにおける、工数の構成および適切な管理工数について考察を行った。一方、未達成PJについては、様々な失敗要因が考えられ、明確な要因の解明までには至っていない。今後、開発途中の工数のかけ方を分析することで、失敗の兆候捕捉の解明につなげていく所存である。

参考文献

- [1] 独立行政法人情報処理推進機構（IPA）技術本部 ソフトウェア高信頼化センター（SEC）、
「SEC BOOKS ソフトウェア開発データ白書 2016-2017」、P26、154、169、2016
- [2] 丸山志保、SQiP2017 レビュー重視と品質・生産性の関係分析、2017
- [3] 誉田直美、「ソフトウェア品質会計」日科技連出版社、P22、2010