

作業時間に基づくソフトウェア保守ベンチマーキングの試み

角田 雅照^{1,2,a)} 門田 暁人¹ 松本 健一¹
大岩 佐和子³ 押野 智樹³

近年、受託開発ソフトウェアは新規の開発が頭打ちの状況に対して、開発済みのソフトウェア資産が増えることに伴い保守の比率が高まり、保守の重要性が高まっている。このため、保守単独で契約を結ぶ場合もある。保守とは、単にソフトウェア出荷後に発見された欠陥を除去することだけを指すのではない。ソフトウェアは利用されているうちに、ビジネス環境の変化により、機能の拡張、修正が必要となる。保守とはそれらの修正も含んだ概念である。本研究では、ソフトウェア保守における作業効率のベンチマークを確立することを試みる。ベンチマークを確立するためには、複数の組織から収集されたプロジェクトのデータ、すなわち企業横断的データを用いて、作業効率に影響する特性を明らかにする必要がある。本研究では、企業横断的データとして、一般財団法人経済調査会によって平成 24 年度に収集された 134 件の事例を用い、作業効率に影響する要因を分析した。作業効率は修正実施プログラム本数÷作業時間数と定義した。分析では、ベンチマーキングのために、箱ひげ図により各特性の作業効率の分布を示すとともに、開発言語が Java であるかどうか、ツールの制約が強いかどうか作業効率に影響することを示した。

1. はじめに

近年、受託開発ソフトウェアは新規の開発が頭打ちの状況に対して、開発済みのソフトウェア資産が増えることに伴い保守の比率が高まり、保守の重要性が高まっている。このため、保守単独で契約を結ぶ場合もある。保守とは、単にソフトウェア出荷後に発見された欠陥を除去することだけを指すのではない。ソフトウェアは利用されているうちに、ビジネス環境の変化により、機能の拡張、修正が必要となる。ソフトウェア保守とは、それらの修正も含んだ概念であり、JIS X0161[7]では以下の 4 つに分類している。

- 是正保守: ソフトウェアの出荷後に発見された問題の訂正
- 予防保守: 出荷後のソフトウェアの潜在的な障害が顕在化する前の是正措置
- 適応保守: 出荷後、変化している環境において、ソフトウェアを使用できるように保ち続けるために実施する修正
- 完全化保守: 出荷後のソフトウェアの性能また保守性を改善するための修正

ソフトウェアの保守に関する作業効率のベンチマーク(組織の作業効率を業界標準と比べて評価するための基準値[5])を確立することは非常に重要である。保守のサービスを提供する組織にとって、ベンチマークは業務改善の基礎となるものである。業務改善をした組織は、価格競争力が高まることが期待される。ユーザ(保守委託企業/組織)にとって、ベンチマークは契約を結んでいるベンダ(保守受託企業)の作業効率を評価することに役立つ。もしベンダの作業効率が低い場合、コストが割高である可能性がある。

り、契約内容の見直しの材料となりうる。

本研究では、ソフトウェア保守における作業効率のベンチマークを確立することを試みる。ベンチマークを確立するためには、複数の組織から収集されたプロジェクトのデータ(企業横断的データ)を用いて、作業効率に影響する特性(例えばシステム構成など)を明らかにし、その特性によってデータを層別する必要がある(利用時には、例えばシステム構成が同じデータから計算した基準値と比較する)。本研究では、一般財団法人経済調査会によって平成 24 年度に収集された 134 件の事例を用いた。

3 章で詳述するが、予備分析に基づき 1 年あたりの保守の作業時間をコストとみなし、1 年あたりの保守実施プログラム(以降 PG)本数を修正量とみなした。ベンチマーキングに用いる作業効率の定義は、実施 PG 本数÷作業時間とした。分析に用いたデータには実施ファンクションポイント(以降 FP)も記録されていたが、欠損値が多かったため、便宜的に修正プログラム本数を用いて作業効率を定義した。4 章の分析では、開発言語、業種などの各特性が作業時間に与える影響を重回帰分析により確かめた。作業時間を増加させる特性は、作業効率を低下させる特性である。また、保守のベンチマーキングのために、特性別に作業効率の分布(箱ひげ図)を示した。

我々は文献[10]においても、企業横断的データを用いて保守における作業効率のベンチマークを確立することを試みている。文献[10]では技術者数を用いて作業効率を定義しており、作業効率を実施 PG 本数÷技術者数としている。これに対し、本研究では作業時間を用いて作業効率を定義している点が主要な違いである。技術者数は保守委託企業が把握しやすいという利点があるが、技術者数に基づいて定義した作業効率は、ややラフな定義となる。これに対し、作業時間は保守委託企業が把握しにくい、作業時間に基づく作業効率の定義は、より正確性が高いといえる。

以降、2 章で分析に用いた対象のデータセットについて説明する。3 章では保守費用などに関する予備分析につい

1 奈良先端大学院大学
Nara Institute of Science and Technology, Japan
2 近畿大学
Kinki University, Japan
3 一般財団法人経済調査会
Economic Research Association, Japan
a) tsunoda@info.kindai.ac.jp

表 1 各特性の詳細
 Table 1 Description of Attributes.

属性	詳細
保守費用	ソフトウェアに対する1年間の保守費用（原価ではなく契約金額）
受託側作業時間	ソフトウェア保守の受託側の年間作業時間
受託側作業員数	ソフトウェア保守の受託側の作業員数（派遣，専任，非専任など全て含む）
総作業時間	ソフトウェア保守の受託側と委託側の年間作業時間の合計
総作業員数	ソフトウェア保守の受託側と委託側の作業員数の合計（派遣，専任，非専任など全て含む）
作業効率	実施 PG 本数 ÷ 総作業時間
実施 FP	1年間に修正を加えた機能量
実施 SLOC	1年間にソースコードに修正を加えた行数
実施 PG 本数	1年間に修正を加えたプログラムの本数
実施画面数	1年間に修正を加えた画面の数
実施帳票数	1年間に修正を加えた帳票の数
実施ファイル数	1年間に修正を加えたデータファイルの数
実施 BAT 数	1年間に修正を加えたバッチファイルの数
保守種別	是正，予防，適応，完全化
システム構成	メインフレーム，Web系，クライアントサーバ
開発言語	SQL，Java，Java Script，Visual Basic，COBOL，HTML，C，JSP，C++など
業種	製造業，卸売・小売業，金融・保険業，サービス業，公務，建設業，情報通信業など
人的要因	保守プロジェクトあるいは保守組織の大きさと熟練度
問題要因	対象とする問題の型と重要度，要求仕様の構成，問題解の制約，波及度合い
プロセス要因	要求仕様言語，設計/プログラミング言語，開発方法論
プロダクト要因	対象システムの信頼性，規模，効率，制御構造，複雑度
資源要因	対象とするハードウェア，期間，予算
ツール要因	ライブラリ，コンパイラ，テストツール，保守ツールなど

て述べ、4章で作業時間に影響する特性を分析した結果について述べる。5章で関連研究について説明し、最後に6章でまとめを述べる。

2. データセット

分析対象としたデータは、一般財団法人経済調査会によって平成24年度に120社から収集された、ソフトウェア保守契約の事例134件である。うち107件が事務系のシステムの事例であり、また74件が定額契約（一定期間の保守業務を一定の金額で契約するもの[9]）の事例である。調査は、主にソフトウェア保守を受託する組織に対して行われた。

分析に用いた各特性の詳細を表1に示す。開発言語については、1システムで複数の言語を使用して開発されることもある。その場合は50%を超えるものを主開発言語として分類した。保守種別は、是正、予防、適応、完全化保守のいずれかの作業割合が、全体の50%を超える場合を取り上げ、それらを分類したものである。

人的要因からツール要因の各項目（まとめて生産性変動要因と呼ぶ）は、文献[9]に基づいて定義された項目であり、生産性変動要因とは、ソフトウェア保守の生産性に影響すると考えられる要因を3段階（1：かなり厳しい，2：少し厳しい，3：厳しくない）で評価した項目である。なお、各特性には欠損値が含まれていたため、分析対象の項目によってデータ件数が異なっている。

作業効率を新たに定義した。作業効率とは、投入された人的資源に対する産出量であり、前者を総作業時間、後者を実施PG本数とみなして定義した。3章の分析結果に基

づき、これらの特性を用いた。

3. 予備分析

外れ値の影響を抑えて関連の強さを分析するために、スピアマンの順位相関係数を用いた。以降では単に相関係数と表す。さらに、重回帰分析により、複数の特性を考慮して目的変数に対する効果を確かめた。重回帰分析では、外れ値の影響を抑えるために、比例尺度の特性に対して対数変換を適用した。

3.1 保守費用に関連する特性

予備分析として、保守費用に影響する特性について分析した。ソフトウェア保守の費用は人件費に基づく、すなわち、受託側作業時間、または受託側作業員数によって決定していると考えられる。各特性が保守費用にどの程度影響を与えているのかを確かめるために、それぞれの相関係数を確かめた。表2に保守費用との相関係数を示す。受託側作業時間のほうが受託側作業員数よりも相関係数が大きかった。

受託側作業時間と受託側作業員数との相関係数を確かめると0.56であった。それぞれの特性が相互に影響している可能性がある。そこで相互関係を考慮して保守費用との関連を分析するため、重回帰分析を用いた。構築されたモデルの標準化偏回帰係数を表3に示す。受託側作業時間のほうが係数も大きく、また有意確率も5%を下回っていた。モデルの調整済みR²は0.73であり説明力が比較的高かった。

よって、保守費用はおおむね受託側作業時間で決定しているといえ、作業時間を減少させることはコストを抑える

表 2 保守費用と作業量との相関係数

Table 2 Correlation Coefficients between maintenance cost and workload.

項目	データ数	相関係数	有意確率
受託側作業員数	76	0.6	0%
受託側作業時間	81	0.79	0%

表 3 受託側作業員数, 受託側作業時間を用いたモデル

Table 3 The model using the number of the vendor's staff and working time of the vendor.

項目	標準化偏回帰係数	有意確率
受託側作業時間	0.76	0%
受託側作業員数	0.15	7%

ことにつながる。この結果は、作業時間および作業員数に基づいて定義された作業効率に着目した分析は、コストに着目した分析とみなせることを定量的に示している。これは当然の結果ではあるが、保守費用と作業員数、作業時間との関係を定量的に分析した研究は我々の知る限り存在せず、結果の信頼性を高めるためには必要不可欠な分析であると考えられる。

3.2 保守費用の影響要因

ソフトウェア保守の作業効率を定義するためには、修正量を示す特性が必要となる。実施 FP が修正量を示す特性として最も適しているが、欠損値が多かった。そこで実施 FP よりもデータ件数が多く、かつ実施 FP との関連が最も強い特性を修正量として採用し、以降の分析を行う。実施 FP 以外では、修正量を表す特性として、実施 SLOC, 実施 PG 本数, 実施画面数, 実施帳票数, 実施ファイル数, 実施 BAT 数が記録されている。これらと実施 FP との相関係数を表 4 に示す。実施 PG 本数の相関係数が最も大きく、また有意確率も 5% を下回っていた。そこで以降の分析では、実施 PG 本数を修正量として採用する。

なお、実施 PG 本数と実施 FP との相関係数は 0.54 であり、非常に関連が強いとまではいえない。よって、実施 PG 本数を用いた場合の分析結果は、実施 FP を用いた場合と異なる可能性があることに注意する必要がある。

3.3 作業量と修正量の関連

作業効率、作業時間に影響する要因の予備分析として、作業量（総作業時間）と修正量（実施 PG 本数）の関係について分析した。分析では、受託側作業時間（作業員数）ではなく総作業時間（作業員数）を用いた。これは、保守費用は受託側の作業量のみが影響するのに対し、修正量は委託側と受託側の両方の作業量が影響するためである。

総作業時間と実施 PG 本数の関係に加えて、総作業員数, 実施 FP との関係も分析し、総作業時間と実施 PG 本数を用いることが妥当かどうかを確かめた。作業量と修正量の相関係数を表 5, 表 6 に示す。実施 PG 本数と総作業時間の関連の強さは中程度であり、その他の作業量と修正量を示

表 4 実施 FP と他修正量との相関係数

Table 4 Correlation Coefficients between FP modified and other amount of modifications.

項目	データ数	相関係数	有意確率
実施 PG 本数	14	0.54	4%
実施 LOC	22	0.37	9%
実施画面数	17	0.28	28%
実施帳票数	13	0.23	45%
実施ファイル数	14	0.46	10%
実施 BAT 数	14	-0.06	84%

表 5 総作業時間と修正量との相関係数

Table 5 Correlation Coefficients between whole working time and amount of modifications.

項目	データ数	相関係数	有意確率
実施 PG 本数	46	0.47	0%
実施 FP	14	0.65	1%

表 6 総作業員数と修正量との相関係数

Table 6 Correlation Coefficients for whole staff and amount of modifications.

項目	データ数	相関係数	有意確率
実施 PG 本数	45	0.58	0%
実施 FP	14	0.53	5%

す特性間の関連の強さも同程度であった。どの特性を用いた場合でも同程度の関連の強さであったため、実施 PG 本数と総作業時間を分析に用いても、特に問題は無いと考えられる。

総作業時間と実施 PG 本数との関係を単回帰分析により確かめた。以降の節において、実施 PG 本数以外に総作業時間に関連する要因を分析するが、予備分析として、実施 PG 本数のみによって総作業時間がどの程度決定しているのかを確かめた。構築されたモデルの調整済み R^2 は 0.32 であり、実施 PG 本数だけでは総作業時間を決定するには不十分であるといえる。

4. 分析

本章では、総作業時間を目的変数、各特性を説明変数として重回帰分析を行うとともに、作業効率と各特性との関係を箱ひげ図で示す。具体的には、重回帰分析では偏回帰係数より「ある特性（Web 系など）を持つシステムのほうが、総作業時間が多くなる」ことを確かめ、その後、ベンチマーキングに用いることを想定し、箱ひげ図により「ある特性を持つシステムの作業効率が低い」ことを確かめた。

箱ひげ図のひげは四分位範囲の 1.5 倍以内の値で最大と最小のケースを示し、丸印は四分位範囲から 1.5 倍以上離れたケース、星印は 3 倍以上離れたケースを示す。図の可読性を高めるため、一部の外れ値は省略した。

重回帰分析時に、名義尺度の変数についてはダミー変数化を行った。モデル構築時にはステップワイズ変数選択を

表 7 システム構成を用いたモデル

Table 7 The model using the system architecture.

項目	標準化偏回帰係数	有意確率
実施 PG 本数	0.51	0%
Web 系 (システム構成)	0.35	0%

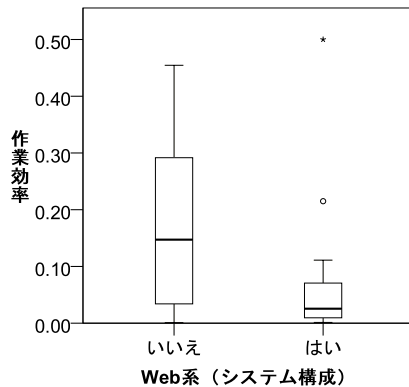


図 1 Web系システムの作業効率

Figure 1 Work efficiency of the web based system.

行い、 $p < 0.05$ の時にモデルに取り込み、 $p > 0.1$ の時にモデルから除外した。

4.1 システム構成との関連

システム構成が作業量、すなわち総作業時間に与える影響について分析した。重回帰分析を用いて総作業時間に対するシステム構成と実施 PG 本数の影響を分析した結果、構築されたモデルの調整済み R^2 は 0.43 となった。Web 系を加えることにより、実施 PG 本数のみの場合に比べ、調整済み R^2 は改善していたが、0.5 はを超えておらず、総作業時間を決定するには十分であるとはいえない。

構築されたモデルの標準化偏回帰係数を表 7 に示す。変数選択の結果、Web 系以外のシステム構成の項目がモデルに残らなかった。Web 系の標準化偏回帰係数が正の値であり、これは、実施 PG 本数、すなわち作業量が同じ 2 つのソフトウェアがある場合、Web 系のほうが総作業時間が多くなる傾向があることを示している。

ベンチマーキングの参考とするために、Web 系システムと作業効率の箱ひげ図を図 1 に示す。図より、明確に作業効率が低くなっていることがわかる。

4.2 開発言語との関連

開発言語が総作業時間に与える影響について分析した。重回帰分析を用いて総作業時間に対する開発言語と実施 PG 本数の影響を分析した結果、構築されたモデルの調整済み R^2 は 0.52 であった。調整済み R^2 が 0.5 を超えていることから、Java と実施 PG 本数により、総作業時間を決定することができるといえる。

構築されたモデルの標準化偏回帰係数を表 8 に示す。変数選択の結果、Java 以外の開発言語の項目がモデルに残らなかった。Java の標準化偏回帰係数は正の値であり、開発

表 8 開発言語を用いたモデル

Table 8 The model using the programming language.

項目	標準化偏回帰係数	有意確率
実施 PG 本数	0.58	0%
Java (開発言語)	0.44	0%

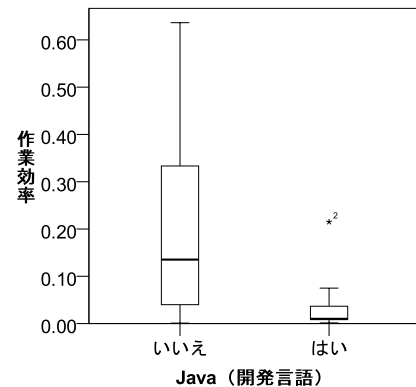


図 2 Java の作業効率

Figure 2 Work efficiency of Java language.

言語が Java の場合、総作業時間を増加させる傾向があるといえる。

ベンチマーキングの参考とするために、Java と作業効率の関係を示す箱ひげ図を図 2 に示す。図より、Java の利用有無で開発効率が大きく異なっていることがわかる。

4.3 業種との関連

業種が総作業時間に与える影響について分析した。重回帰分析を用いて業種の影響を分析した結果、構築されたモデルの調整済み R^2 は 0.37 となった。調整済み R^2 が 0.5 を下回っていることから、業種と実施 PG 本数だけでは、総作業時間を決定できないといえる。

重回帰分析により構築されたモデルの標準化偏回帰係数を表 9 に示す。変数選択の結果、金融・保険業の業種の特徴のみがモデルに残った。金融・保険業の係数は正の値であり、業種が金融・保険業の場合、総作業時間が増加する傾向があるといえる。

金融・保険業の業種と作業効率の関係を示す箱ひげ図を図 3 に示す。金融・保険業に該当する場合、開発効率が明確に低かった。

4.4 生産性変動要因との関連

生産性変動要因が総作業時間に与える影響について重回帰分析を用いて分析した。構築されたモデルの調整済み R^2 は 0.41 であった。調整済み R^2 が 0.5 を下回っていることから、生産性変動要因と実施 PG 本数だけでは、総作業時間を決定できないといえる。

表 10 に生産性変動要因の標準化偏回帰係数を示す。変数選択の結果、ツール要因以外の項目がモデルに残らなかった。ツール要因の標準化偏回帰係数は負の値であった。これは、ツール要因の値が大きい (要因が厳しくない) 場

表 9 業種を用いたモデル

Table 9 The model using the business sector.

項目	標準化偏回帰係数	有意確率
実施 PG 本数	0.58	0%
金融・保険業(業種)	0.24	5%

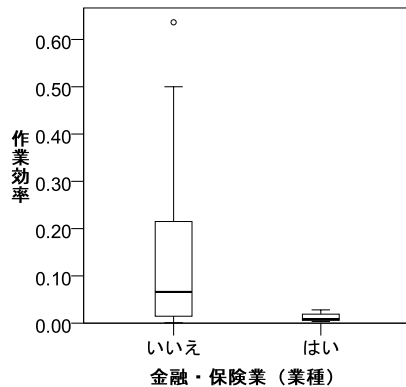


図 3 金融業・保険業の作業効率

Figure 3 Work efficiency of banking and insurance.

合、総作業時間を減少させる傾向があることを示している。

ツール要因と作業効率の関係を示す箱ひげ図を図 4 に示す。紙面の都合上、ツール要因以外の箱ひげ図を省略するが、どの生産性変動要因の場合も、値が 3 の場合、開発効率が高かった。特にツール要因の場合、箱ひげ図においても開発効率の差が顕著だった。

4.5 保守種別との関連

保守種別が総作業時間に与える影響について分析した。重回帰分析を用いて保守種別の影響を分析したが、保守種別がモデルに残らなかった。そこで変数選択を行わずに、データ数の多い是正保守と適応保守を用いてモデルを構築した。構築されたモデルの調整済み R^2 は 0.41 であった。調整済み R^2 が 0.5 を下回っていることから、保守種別と実施 PG 本数だけでは、総作業時間を決定できないといえる。

重回帰分析により構築されたモデルの標準化偏回帰係数を表 11 に示す。適応保守の有意確率は 5% に近く、係数は負の値であった。適応保守の場合、総作業時間を減少させる可能性があるが、有意確率が 5% を上回っているため、結論を下すためには今後さらに分析が必要である。

保守種別と作業効率の関係を示す箱ひげ図を図 5 に示す。適応保守の場合、是正保守と比べて開発効率が高い傾向があることが読み取れる。

4.6 複数の特性との関連

4.1 節から 4.5 節の分析では、ユーザまたはベンダがある特性に着目してベンチマーキングを行うことを想定し、それを支援するために各特性と総作業時間との関係について分析した。例えば、システム構成に着目したベンチマーキングを支援するために、システム構成は作業量に影響しているのかを重回帰分析により示し、システム構成により作

表 10 生産性変動要因を用いたモデル

Table 10 The model using the productivity factors.

項目	標準化偏回帰係数	有意確率
実施 PG 本数	0.44	0%
ツール要因	-0.39	0%

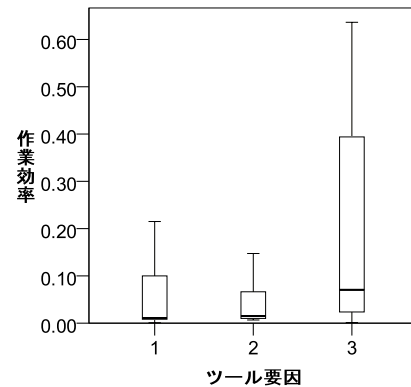


図 4 ツール要因の作業効率

Figure 4 Work efficiency of tool factor.

業効率にどの程度の違いがあるのかを箱ひげ図により示した。

本節では、総作業時間に影響を与えていた特性のうち、特にどの特性が影響を与えているのかを明らかにするために、それらの特性全てと修正 PG 本数を説明変数として重回帰分析を行った。説明変数の候補は、Web 系（システム構成）、Java（開発言語）、ツール要因（生産性変動要因）、金融・保険業（業種）、適応保守（保守種別）、修正 PG 本数である。

重回帰分析により構築されたモデルの調整済み R^2 は 0.56 であった。調整済み R^2 が 0.5 を上回っていることから、開発言語、生産性変動要因、実施 PG 本数により、総作業時間を決定できるといえる。モデルの標準化偏回帰係数を表 12 に示す。変数選択の結果、Web 系（システム構成）と金融・保険業（業種）、適応保守（保守種別）はモデルに残らなかった。これらの特性は、特性間の場合、総作業時間に影響を与えているとはいえなかった。ツール要因の標準化偏回帰係数は負の値であり、特性間の関係を考慮した場合でも、ツール要因が厳しくない場合、総作業時間は小さくなる傾向があるといえる。

係数の絶対値の大きさから、実施 PG 本数が総作業時間に最も影響を与えており、Java を利用しているかどうか次に影響を与えるといえる。ツール要因は最も総作業時間に与える影響が小さいが、係数の有意確率は 5% を下回っており、ツール要因も総作業時間に影響しているといえる。

5. 関連研究

いくつかの研究において、保守の作業効率に影響する要因の分析がされている。Jørgensen[4]はある企業のデータを分析し、母体ソフトウェアの規模と開発言語（第 3 世代言

表 11 保守種別を用いたモデル

Table 11 The model using the maintenance type.

項目	標準化偏回帰係数	有意確率
実施 PG 本数	0.60	0%
是正保守 (保守種別)	-0.13	38%
適応保守 (保守種別)	-0.27	7%

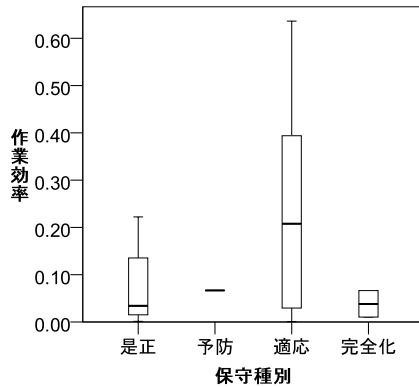


図 5 保守種別ごとの作業効率

Figure 5 Work efficiency of maintenance type.

表 12 複数の特性を用いたモデル

Table 12 The model using the multiple factors.

項目	標準化偏回帰係数	有意確率
実施 PG 本数	0.47	0%
Java (開発言語)	0.37	0%
ツール要因	-0.29	1%

語と第 4 世代言語) は作業効率に関連がないことを示している。Ahn ら[1]は生産性変動要因と類似の概念(技術的要因, 人的要因など)の属性を保守工数見積モデルに用いている。ただし, これらの研究では, 本研究で用いたような企業横断的データを分析対象としていない。

ソフトウェア開発に関する企業横断的データについては, ISBSG (International Software Benchmarking Standards Group)[2]が収集しており, ベンチマーク[5]や分析[8]に広く用いられている。ソフトウェア保守の場合も同様に, ISBSG が収集した企業横断的データが存在する[3]。文献[9]において, このデータを分析しているが, 企業横断的データにかかわらず, 欠損値などにより, 実際には数社のデータしか分析対象となっていない。これに対し, 本研究では多様な企業が分析対象となっており, 結果の一般性が高いことが期待される。

ソフトウェア保守に関する企業横断的データを分析した研究, 報告が少数ではあるが存在する。経済産業省と日本情報システム・ユーザー協会との調査[6]では, 企業横断的データを用い, 技術者あたりの保守対応件数を作業効率とみなして, 業種別の技術者あたり対応件数を示している。我々は文献[10]において企業横断的データを用い, 作業効率(修正プログラム本数÷技術者数)に関連する要因を分

析している。これらの研究では技術者数に基づいて作業効率を定義しているが, ややラフな定義といえる。これに対し, 本研究は作業時間に基づいて作業効率を定義している。作業時間は保守委託企業が把握しにくい, 作業効率の定義としては, より正確性が高いといえる。

6. おわりに

本研究では, ソフトウェア保守における作業効率のベンチマークを確立するために, 企業横断的データを用いて, 作業効率に影響する特性を明らかにした。予備分析では, 保守費用と作業時間との関連が強いことを示すとともに, 実施 PG 本数の代わりに用いる修正量として, 実施 PG 本数が最適であることを示した。この結果に基づき, 作業効率を実施 PG 本数÷作業時間と定義し, 分析を行った。

分析では, 保守のベンチマーキングのために, 特異別に作業効率の箱ひげ図を示すとともに, 開発言語, 業種などの各特性が作業時間に与える影響を重回帰分析により確かめた。作業時間を増加させる特性は, 作業効率を低下させる特性である。修正 PG 本数と各特性を説明変数とし, 作業時間を目的変数として重回帰分析した結果, Web 系(システム構成), Java (開発言語), ツール要因(生産性変動要因), 金融・保険業(業種), 適応保守(保守種別)が作業時間との関連があった。これらの特性を同時に説明変数として用いた場合, Java (開発言語)とツール要因(生産性変動要因)が作業時間に関連があった。

本研究の主要な貢献は, 多数の企業が含まれた企業横断的データを用い, 作業時間に基づいて定義した作業効率を用いて, 作業効率に関連する特性を明らかにしたことである。本研究で示した箱ひげ図に基づいて, 簡易的にベンチマークをすることは可能であると考えられる。ただし, 作業効率の定義に実施 PG ではなく実施 PG 本数を使っており, また作業効率の分散が大きいため, ベンチマークを絶対視するのではなく, 判断材料のひとつとすべきである。なお, 開発効率と関連があった各特性のうち, ツール要因は変更できる可能性があるが, 開発言語などは開発済みのソフトウェアであるため変更することができない。開発効率と関連が強く, かつ変更することが可能な要素(例えば保守の体制, 保守の手順など)を明らかにすることは今後の課題である。

謝辞 本研究の一部は, 文部科学省科学研究補助費(基盤 C: 課題番号 25330090)による助成を受けた。

参考文献

- [1] Ahn, Y., Suh, J., Kim, S. and Kim, H.: The software maintenance project effort estimation model based on function points, *Journal of Software Maintenance: Research and Practice*, Vol. 15, No. 2, pp. 71-85 (2003).
- [2] International Software Benchmarking Standards Group,

<https://www.isbsg.org/>

- [3] International Software Benchmarking Standards Group (ISBSG): *ISBSG Maintenance & Support Data Suite Release 4*, ISBSG (2010).
- [4] Jørgensen, M.: Experience With the Accuracy of Software Maintenance Task Effort Prediction Models, *IEEE Transactions on Software Engineering*, Vol. 21, No. 8, pp. 674-681 (1995).
- [5] Lokan, C., Wright, T., Hill, P. and Stringer, M.: Organizational Benchmarking Using the ISBSG Data Repository, *IEEE Software*, Vol. 18, No. 5, pp. 26-32 (2001).
- [6] 日本情報システム・ユーザー協会 編：ソフトウェアメトリクス調査 2008, 日本情報システム・ユーザー協会 (2008).
- [7] 日本規格協会：JIS X 0161 ソフトウェア技術—ソフトウェアライフサイクルプロセス—保守, 日本規格協会 (2008).
- [8] Rodríguez, D., Sicilia, M., García, E. and Harrison, R.: Empirical findings on team size and productivity in software development, *Journal of Systems and Software*, Vol. 85, No. 3, pp. 562-570 (2012).
- [9] ソフトウェア・メンテナンス研究会：第4年度 ソフトウェア・メンテナンス研究会報告書, ソフトウェア・メンテナンス研究会 (1995).
- [10] Tsunoda, M., Monden, A., Matsumoto, K. and Oshino, T.: Analysis of Software Maintenance Efficiency Focused on Process Standardization, *Proc. of International Workshop on Empirical Software Engineering in Practice (IWESEP)*, pp.3-8, Nara, Japan (2011).
- [11] Tsunoda, M. and Ono, K.: Pitfalls of Analyzing a Cross-company Dataset of Software Maintenance and Support, *Proc. of International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)* (to appear).