

ソフトウェア・メトリクス調査2025 【ガイドブック】

2025 年版

2025 年4 月

一般社団法人 日本情報システム・ユーザー協会

目次

第1章 はじめに	4
(1-1) 情報システム再考	4
(1-2) 情報システムの役割の変化	5
(1-3) 情報システムの開発方法の多様化	5
(1-4) ソフトウェア・メトリクスを持つ意味合いの変化	6
(1-5) 分析・執筆	6
第2章 開発のソフトウェア・メトリクス	7
(2-1) システム開発の全体工数と全体工期の関係	7
(2-2) 開発フェーズ別の工数と工期	9
(2-3) プロジェクトの実績工期と標準工期	12
(2-4) 工期遅延度への影響	14
(2-5) 予定（計画）工期	15
(2-6) 品質の評価	17
(2-7) 換算欠陥数、換算欠陥率	17
(2-8) 開発総費用	21
(2-9) 工数区分別 平均単価	22
【コラム】本ガイドブックで用いた分析手法	24
参考：調査票でのフェーズの呼称と SLCP との対応表	27
第3章 JUAS ソフトウェア・メトリクス活用事例インタビュー	28
【事例1】大規模や未経験案件の事前リスク確認、失敗プロジェクトの事後評価に利用	28
【事例2】経営層への説明に業界標準として利用	30
【事例3】親会社との共通言語として活用	32
第4章 あとがき	34

第1章 はじめに

「ソフトウェア・メトリクス」は、情報システムに関する QCD「品質 (Quality)・費用 (Cost)・工期(Delivery/Time)」とユーザー満足度 (User Satisfaction) を評価する評価項目と基準値について、情報システムの開発・保守に取り組んできた企業のアンケート回答に基づいてまとめたものである。

その内容は過去の多くのプロジェクトを参考に自社プロジェクトを進めるに際しての指標となるものであり、開発生産性の向上を図り、プロジェクトの失敗を避けるための指針として活用していただければ幸いである。

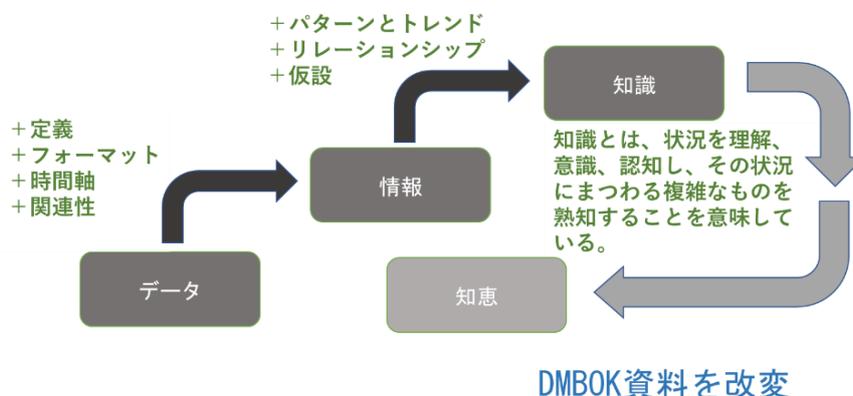
なお、「ソフトウェア・メトリクス調査 2025」は、従来の調査事項に対して、ユーザー企業を中心に活用状況についてのアンケート調査を行い、主に活用されているメトリクス 10 項目に絞り込んでアンケートを実施し、分析・考察・活用の仕方をまとめたガイドブックとしている。

(1-1) 情報システム再考

情報システムの使命は ISO/ANSI1985 で、「ビジネスに關与する人々の意思疎通を支援すること」と定義され、その基幹部分は「実世界のシミュレーション」「実世界の写像」とも言えるものであった。

情報システムの KPI ともいえるソフトウェア・メトリクスをまとめるに際し、情報システムとは何かを再度考えることから始めてみたい。

図表-1-1-1 情報システムの役割



図表 1-1-1 は、事実を表す「データ」は意味付けされて「情報」となり変化することを表している。「情報」は時間的な概念を有し、他との関連性を持ち、ものごとの内容や事情を表す。「情報」は、そのパターンやトレンド、関連性、仮説を持たせることで価値のある「知識」となり、それを人が活かすことで、意思決定のための「知恵」となる。企業における情報システムの重要性がよく理解できる。

「情報」を取り扱う「系」としての「情報システム」は、実ビジネスを写像するとともに人と組織の意思決定を支援する役割を果たしていると言える。少し表現を変えると、情報システムは Intelligence を生み出す力を持つサイクルとして考えることができる。このサイクルは「事実としてのデータ」を「意味のある情報」「価値を生む知識」「意思決定の知恵」と形を変えていく。サイクルの起点であり、根幹となるデータが、意味内容まで定義されて正確にシステムに反映されることが最も肝要である。

言い換えれば、「データ」こそが「情報システムの肝」であり、データを正確に把握し活用していく

ために情報システムが存在しているといえる。

(1-2) 情報システムの役割の変化

我が国の情報システムの導入は、主に大企業において1960年代から業務の効率化・省力化を目的として始まり、企業の競争力を高めることに大いに貢献してきた。それから半世紀、情報技術（IT）の進歩は「ムーアの法則」（18ヶ月毎に実装半導体数が2倍になる）を引き合いに出すまでもなく著しい発展を遂げ、企業経営に対して情報システムが極めて重要なリソースになり、我々もパソコン・スマートフォンなしでは仕事も日々の生活も成り立たない状況にまで至っている。

このような技術の進化の中で情報システムは、従来の「効率化・省力化」だけでなく、変革を促す「価値創造」が求められるようになり、情報システムの果たす役割も大きく変化してきた。

第1の役割は、「ビジネスの写像」として、組織が企業の目的を達成するために行う活動（オペレーション、マネジメント、意思決定など）を支援することとして位置付けられ、事業の継続性を担保するものであり、各企業の競争力強化・発展に重要な役割を果たしてきた。

第2の役割は「価値創造」であり、テクノロジーを組合せて新しいビジネスを創造したり、データ・情報から生まれた新しいアイデアを具体化・実現化したりするなかで、創造的破壊を引き起こしていくことを支援し、ビジネス創造のトリガーとしてITを活用していくことになる。まことに大きな変化といえる。

図表-1-2-1 情報システムの役割の変化

	役 割	効 果
①	人と組織の意思決定、目標達成活動の支援	事業の継続性担保
②	新技術によるイノベーションとしての企業変革を継続するトリガー	創造的破壊の支援

まさに、「情報システムの成否が経営の成否に直結する」と言っても過言ではない状況にまで至っており、21世紀をデジタルの世紀として切り開いていく力となっていると考えている。

(1-3) 情報システムの開発方法の多様化

企業では1960年代から順次システムを導入してきており、今では大部分の業務がシステム化されている。結果として、極めて複雑で大規模なシステム資産を保有するとともに、システム同士が複雑にからみあい、長年の保守・運用によってプログラムにも手を加え続け、今や「スパゲッティ」と称されるように、複雑化・ブラックボックス化したレガシーシステムを保有するに至っている。

システムそのものに①構築年代が異なり、利用技術が異なるシステム群が乱立、②設計書がない巨大なシステムが今も残存、③アドホックなシステム連携による“スパゲティ化”、④データマネジメントやデータガバナンスが不存在、⑤システム運用、情報セキュリティは個別最適、などサイロ型システムとしての問題を残すなかで、システム開発・保守を進めなければならないようになっている。

一方、開発方法に関しては、従来からのウォーターフォール開発に加えパッケージ活用開発、インクリメンタル・イテレーション開発、既存システム刷新のためのマイグレーションやリフォーム、変化への素早い対応を求めるアジャイル開発・超高速開発、さらにアプリケーションの疎結合を求めるマイクロサービス活用開発、既存システムとのデータ連携を保証するための汎用API活用、パブリッククラウ

ド利用による SaaS、生成 AI を用いたローコード／ノーコード開発、等々、企業は取り巻く状況やシステム活用の目的に応じて開発方法を拡大してきた。

また、開発生産性向上を目的とした開発フレームワーク活用も増加してきており、多くの企業で定着している。

データ保有も DB、DWH、データレイクなど、検索能力の向上とともに大きく多様化してきている。

(1-4) ソフトウェア・メトリクスの持つ意味合いの変化

このように情報システムの持つ役割と価値、開発方法が多様化する中で、ソフトウェア・メトリクスの果たす役割も限定的になり、その意味合いも大きく変化せざるを得なくなってきた。

約 15 年に亘って、企業の皆様のご協力を得てまとめてきたソフトウェア・メトリクスであるが、その適用範囲は、従来からのウォーターフォール開発を主とするものであり、パッケージ・クラウド利用開発、アジャイル・超高速開発などのさらに多様化する開発環境や開発方法に関して、ソフトウェア・メトリクスのあり方をあらためて検討していかなければならない状況に至っている。

しかしながら、今までに集められた「ソフトウェア・メトリクス」は貴重なデータであり、自社の指標の確認はこれをベースとして十分行えるものである。各企業のおかれた環境条件やシステム開発・保守の方法に合わせて本メトリクスを活用していただければ、KPI として十分役立てると考えている。

(1-5) 分析・執筆

- 分析／執筆（敬称略 所属は 2025 年 3 月現在）

佐々木 盛朗

藤本 礼久 （一社）日本情報システム・ユーザー協会 参与

- 事務局

近田 敦子 （一社）日本情報システム・ユーザー協会 マネージャー

橋本 葵 （一社）日本情報システム・ユーザー協会 マネージャー

第2章 開発のソフトウェア・メトリクス

調査・分析並びに考察について、以降に記載する。

なお、本ガイドブックに用いる回帰分析に関するコラムを章末に掲載しているので、分析手法についての理解を深めたい方は、ぜひ参照頂きたい。

(2-1) システム開発の全体工数と全体工期の関係

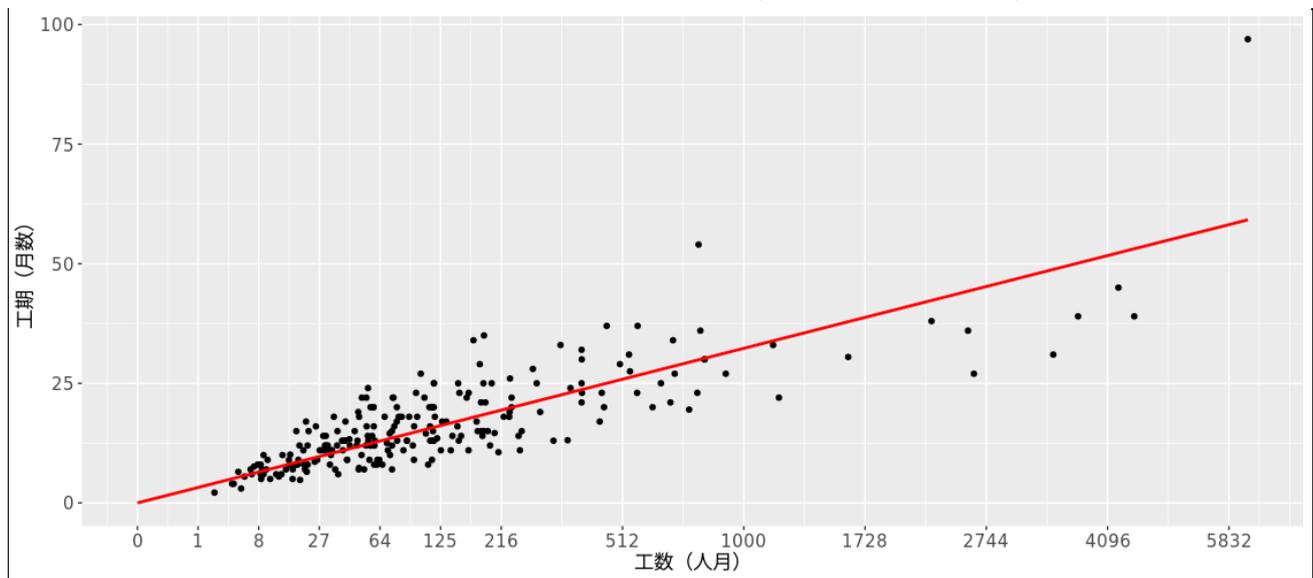
工数と工期の関係： $y = 3.23 \sqrt[3]{x}$

x ：投入工数（全体工数）、 y ：標準的な工期（全体工期）

<使用データ説明>

- 分析対象データは 232 件。
- 全データは 366 件で、実績工数と実績工期が明確なデータは 258 件。
- 工数の立方根と工期の比に関して上下 5%にあたる 26 件を外れ値として、258 件から除外。

図表-2-1-1¹ 工数と工期の関係、サンプル数 = 232 / 258 , $y = 3.23 x^{(1/3)}$, $\text{adj. } R^2 = 0.90$



<分析結果・考察>

- 自由度調整済み決定係数が 0.90 であるため、プロジェクト間の実績工期の差異の 9 割は、実績工数によって説明できる。
- 工数から工期を見積もる例（上記囲みの、工期と工数の関係、の使用例）：
125 人月のプロジェクトは、工数の立方根が 5 となるので、係数 3.23 を積算し、16.15 か月が標準的な工期（全体工期）となる。

¹ソフトウェア・メトリクス調査 2020 の図表 6-4-4 と対応している。

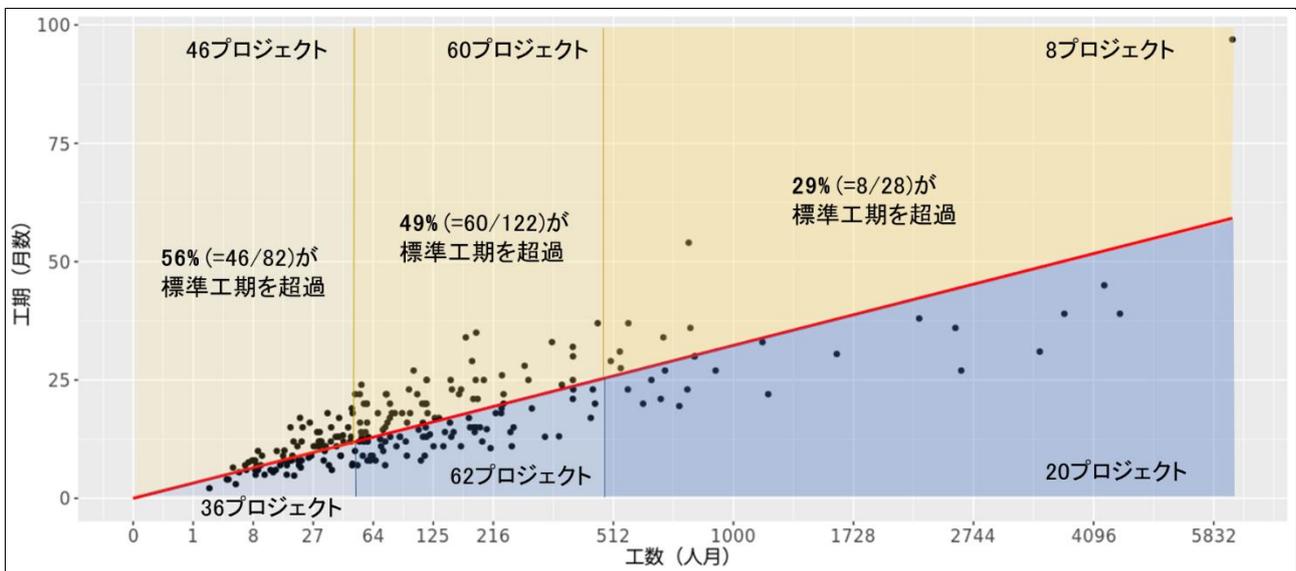
- ある工期で実施できるプロジェクトの規模（工数）の目安を得ることができる。

以下、実績工期が標準工期を超過する割合を分析した。

図表-2-1-2 データプロファイル（プロジェクト件数）

	実績工期が標準工期以下（件）	実績工期が標準工期を超過（件）	実績工期が標準工期を超過する割合（%）
全体	118	114	49
50人月未満	36	46	56
50以上500未満	62	60	49
500人月以上	20	8	29

図表-2-1-3 開発規模別の実績工期が標準工期を超過する割合



<分析結果・考察>

- 50人月未満のプロジェクトは、実績工期が標準工期を超過する傾向(56%)にあるため、標準工期に対して余裕を持って工期を計画することが望ましい。
- 500人月以上のプロジェクトは、実績工数が標準工期を下回る傾向(71%)にある。500人月以上の大規模プロジェクトでは、分担できる作業が多いことから、投入人員を増やすことによって短縮できる作業の割合が工期に影響すると考えられる。

(2-2) 開発フェーズ別の工数と工期

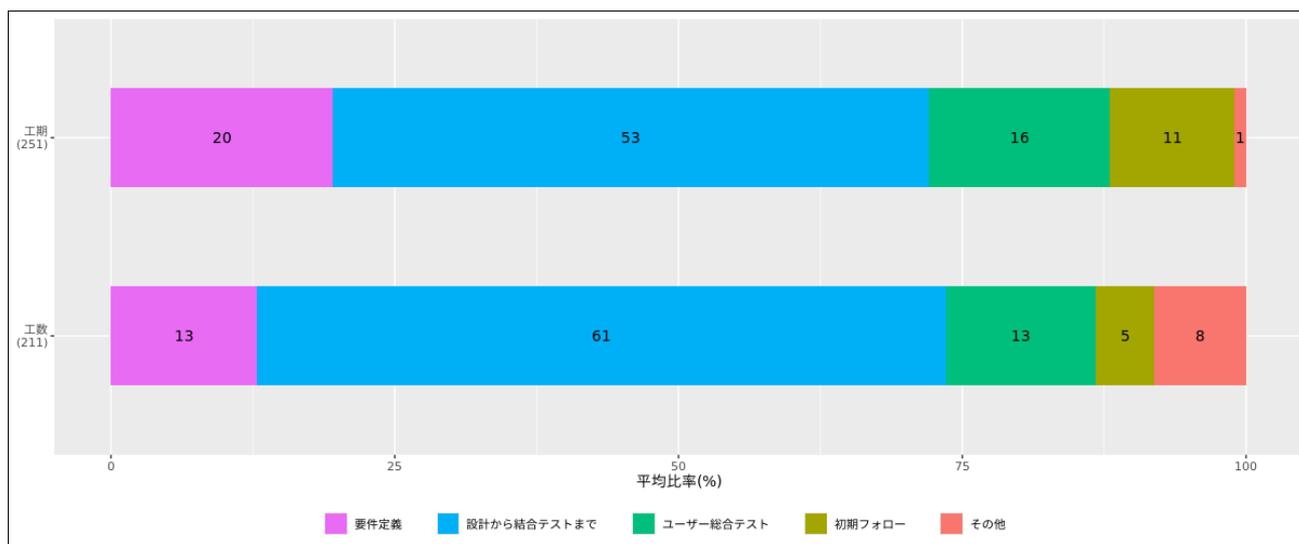
■ 工期 (要件定義：設計～結合テスト：総合テスト) = 20 : 50 : 30

■ 工数 (要件定義：設計～結合テスト：総合テスト) = 15 : 60 : 25

<使用データ説明>

- 工期、工数の比率に用いた開発工程とアンケート回答の開発工程との対比は、以下の通り。
要件定義：設計～結合テスト：総合テスト（ユーザー総合テスト、初期フォロー、その他）
- 工期に関する対象データは 251 件。分析対象データは、開発フェーズ別の工期が明らかであるデータ 211 件。
- 各フェーズの比率の小数点以下は四捨五入とし、上記囲みの比率は、5 刻みに丸めた。

図表-2-2-1² 開発フェーズ別の工数と工期の比率



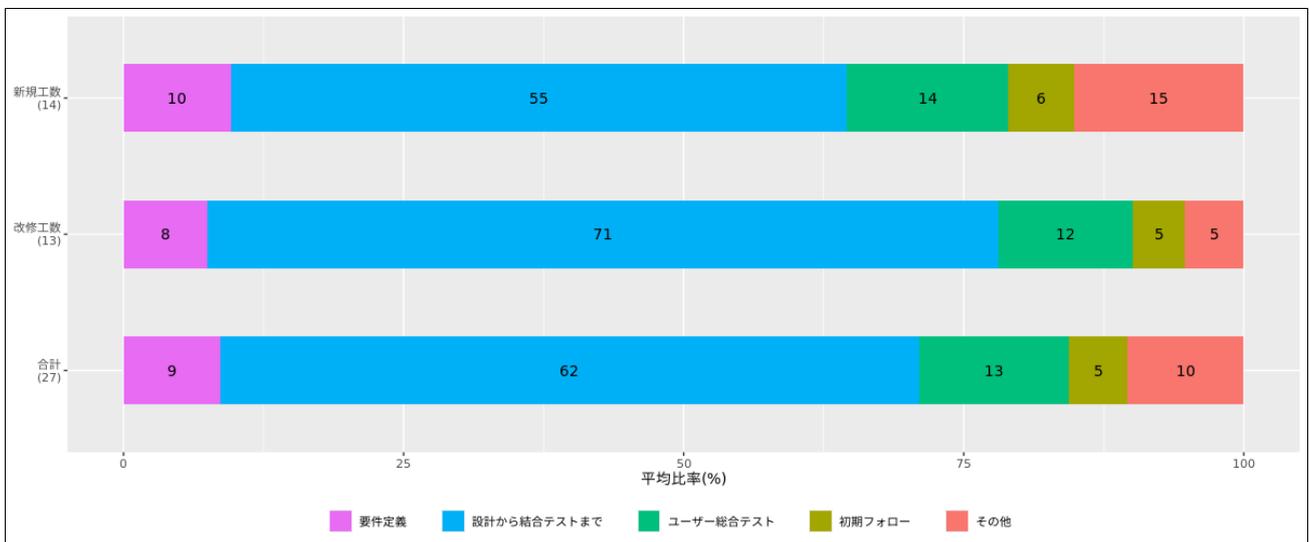
<分析結果・考察>

- 要件定義の全体に占める工期比率(20%)が、工数比率(13%)よりも大きい状態にある。要件定義は他の工程と比較して、作業を分担して進めることが適さないために工期比率が大きくなっているものと考えられる。
- 設計から結合テストまでの全体に占める工期比率(53%)が、工数比率(61%)よりも小さい状態にある。設計から結合テストまでの工程は他の工程と比較して、作業の分割が行いやすく作業分担によって工期比率が小さくなっているものと考えられる。
- 総合テストの全体に占める工期比率(28%)は、工数比率(26%)と概ね同じ状態にあるが、総合テストの内訳を比較するとそれぞれに特徴がある。

² ソフトウェア・メトリクス調査 2020 の図表 6-2-3、図表 6-3-2 と対応している。

- ユーザー総合テストの全体に占める工期比率(16%)は、工数比率(13%)と近い値である。ユーザー総合テストは、多くのメンバーに参加してもらいシステム内容を理解してもらう目的で行い、稼働後早期にプロジェクトの目標を達成することが肝要である。
- 初期フォローの全体に占める工期比率(11%)は、工数比率(5%)と比較して2倍となっている。初期フォロー時は、対応事項が予め特定されていないことが多いため、対応方針の検討・策定の期間が必要となるため、工期比率が大きくなっているものと考えられる。
- その他フェーズの全体に占める工期比率(1%)は、工数比率(8%)と比較して小さい。その他フェーズに含まれる明確に分類できない開発業務は、アドホックに認識された課題の解決に費やされ、随時対応するために工数を要するが、工期に与える影響は小さいと考えられる。

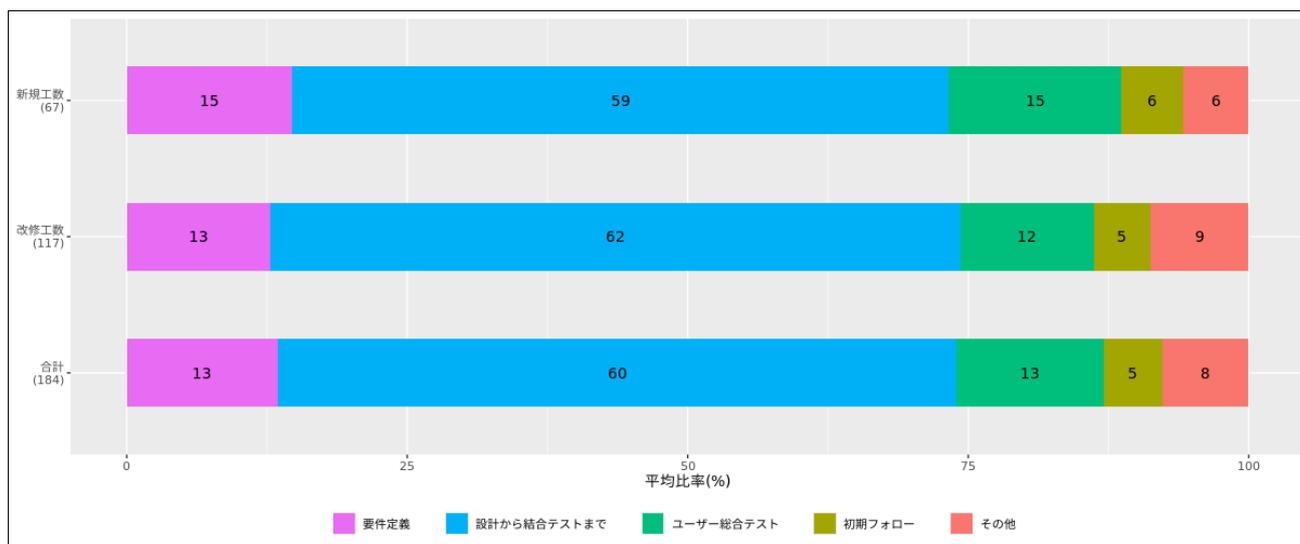
図表 2-2-2⁵ 500人月以上の新規/改修プロジェクト別の工数比率



<分析結果・考察>

- 500人月以上の新規/改修プロジェクトの設計から結合テストまでの新規工数比率(55%)は、改修工数比率(71%)よりも小さい状態にある。
500人月以上の改修プロジェクトは、要件定義・ユーザー総合テスト・初期フロー・その他の工程における業務が定めやすく、設計から総合テストまでに工数を集中させやすいと考えられる。
また、500人月以上の新規プロジェクトは、開発環境やドキュメント整備にかかる工数が改修プロジェクトに対して大きくなることから、設計から結合テストまでの新規工数比率が改修工数比率よりも小さくなると考えられる。
- 500人月以上の新規/改修プロジェクトのその他の新規工数比率(15%)は、改修工数比率(5%)よりも大きい状態にある。新規プロジェクトは、事前に定めきれない業務が多く発生していると考えられる。

図表 2-2-3⁵ 500 人月未満の新規/改修プロジェクト別の工数比率



<分析結果・考察>

- 500 人月未満のプロジェクトでは、新規と改修の工程比率の差異が小さい。500 人月未満のプロジェクトでは、新規・改修に関わらず同様な業務と業務量が求められているためと考えられる。

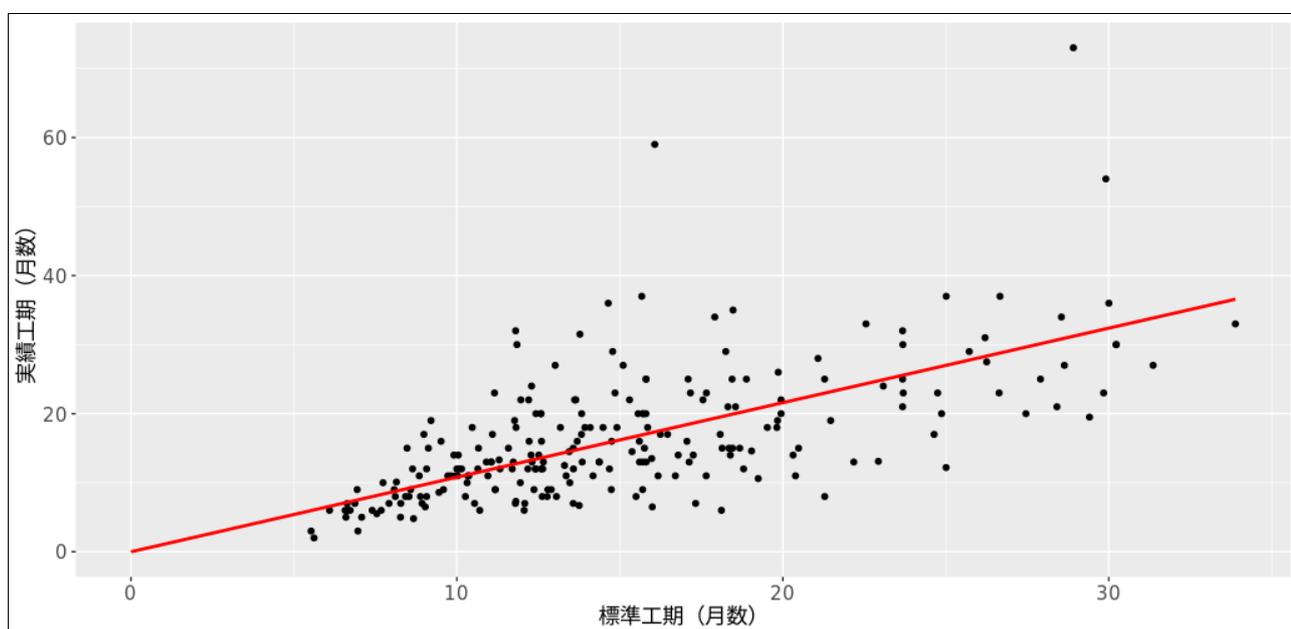
(2-3) プロジェクトの実績工期と標準工期

プロジェクトの実績工期=1.08×標準工期

<使用データ説明>

- 分析対象のデータは 232 件。
- 全データは 366 件で、実績工数と実績工期が明確なデータは 258 件。
- 工数の立方根と工期の比に関して上下 5%にあたる 26 件を外れ値として、258 件から除外。

図表 2-3-1³ 標準工期と実績工期の関係、サンプル数 = 232 / 258 , $y = 1.08x$, $\text{adj. } R^2 = 0.85$

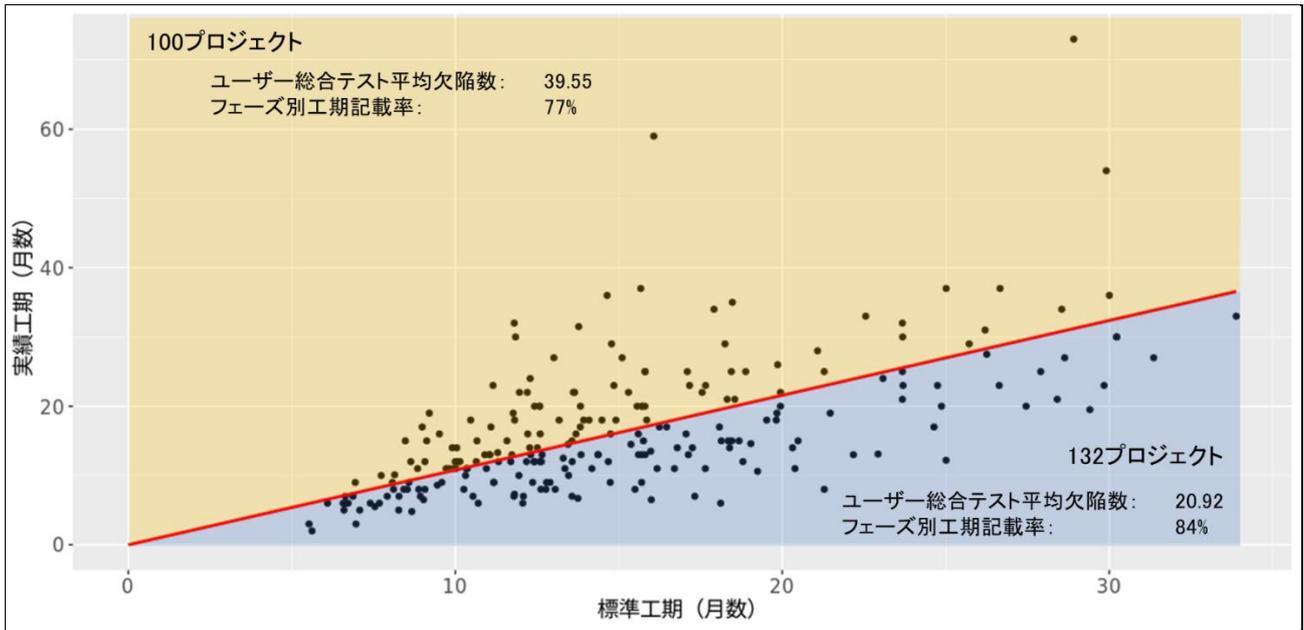


<分析結果>

- 自由度調整済み決定係数が 0.85 であるため、プロジェクトの実績工期の 85%は、標準工期 (= $3.23 \sqrt[3]{\text{実績工数}}$) によって説明できる。
- プロジェクトの実績工数が標準工期の 1.08 倍となっていることから、標準工期から 1 割程度余裕を持って計画を立案することが安全であると考えられる。

³ ソフトウェア・メトリクス調査 2020 の図表 6-2-8 と対応している。

図表 2-3-2 実績工期が標準工期を超えるプロジェクトと標準工期以下のプロジェクトの特徴



図表 2-3-3 データプロファイル

	実績工期が標準工期以下	実績工期が標準工期を超過
プロジェクト数	132	100
ユーザー総合テスト (平均欠陥数)	20.92	39.55
初期フォロー (平均欠陥数)	9.99	8.51
フェーズ別工期 (記載率/記載数)	84% / 111	77% / 77

<分析結果・考察>

- ユーザー総合テスト (平均欠陥数) が、実績工期が標準工期以下で 20.92 に対し、実績工期が標準工期を超過で 39.55 となっている。ユーザー総合テストの欠陥数を抑制することで、標準工期以下に実績工期を収められると考えられる。
- 実績工期が標準工期以下のプロジェクトのフェーズ別記載率は 84% で、実績工期が標準工期を超過のプロジェクトのフェーズ別記載率が 77% であることから、フェーズ別工期を管理しているプロジェクトの方が、標準工期以下に実績工期を抑制できる傾向があると考えられる。

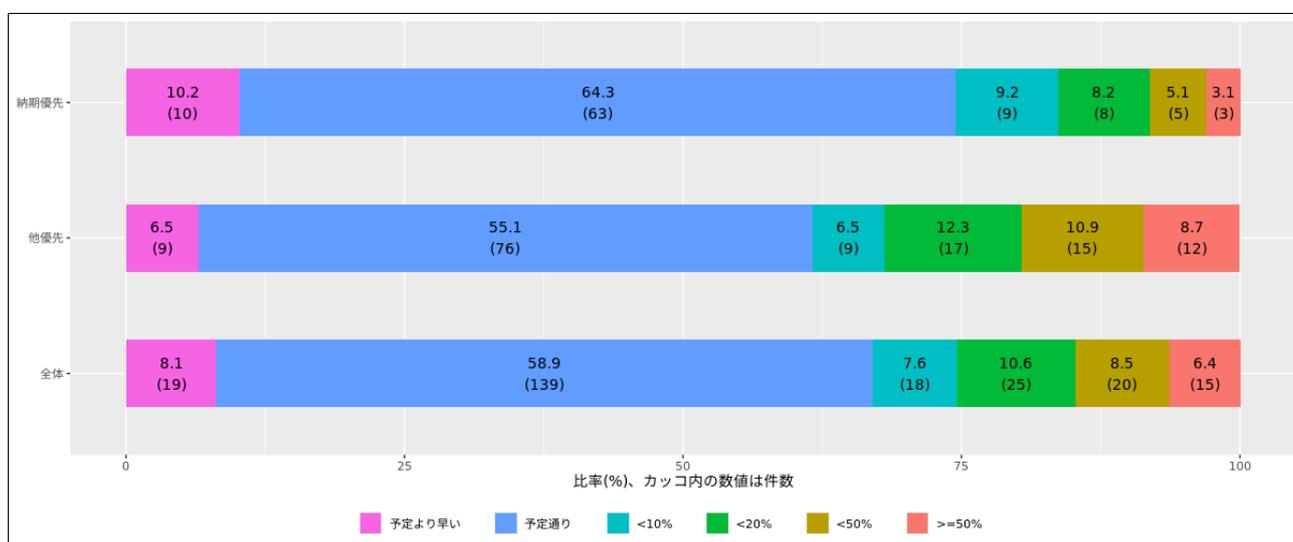
(2-4) 工期遅延度への影響

納期優先したプロジェクトは、予定工期で完了する傾向にある

<使用データ説明・定義>

- プロジェクトにおいて納期を優先することと、工期遅延との関係性を定量的に分析した。
ここで、工期の遅延度は、(実績工期/計画工期) - 1 と定義した。
- プロジェクトの最優先事項、計画工期、計画工数が明らかな 236 件を対象にした。

図表 2-4-1⁴ 納期を優先したプロジェクトの工期遅延度



図表 2-4-2 データプロファイル

	納期優先	他優先
予定より早く完了	10.2%	6.5%
予定通りに完了	64.3%	55.1%
計	74.5%	61.6%

<分析結果・考察>

- 納期を優先したプロジェクトは、予定工期までに完了(74.5%)することが多い。
- 納期を優先したプロジェクトが、20%以上の遅延度になることは少ない (8.2%)。
一方、他の事項を優先したプロジェクトの 20%以上の遅延度は 19.6%となった。

⁴ ソフトウェア・メトリクス調査 2020 の図表 6-2-9 と対応している。

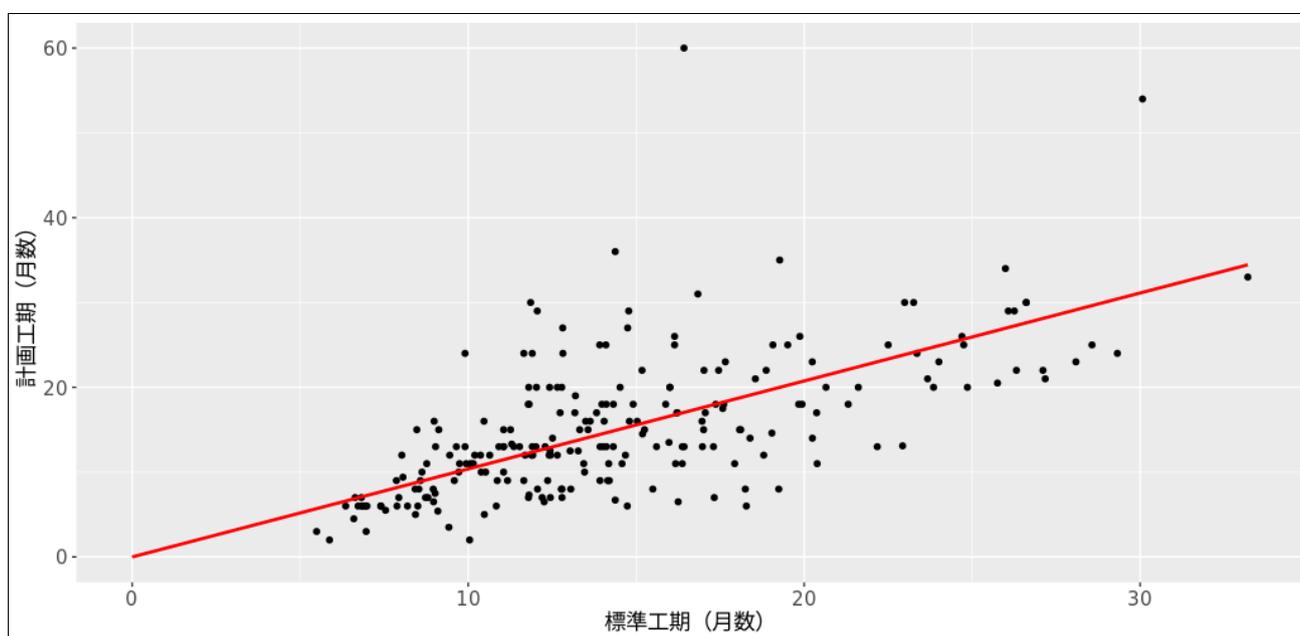
(2-5) 予定（計画）工期

プロジェクトの予定（計画）工期=1.04×標準工期

<使用データ説明>

- 分析対象のデータは 224 件。
- 全データは 366 件で、工数と工期が明確なデータは 250 件。
- 工数の立方根と工期の比に関して上下 5%にあたる 26 件を外れ値として、250 件から除外。

図表 2-5-1⁵ 標準工期と計画工期の関係、サンプル数 = 224 / 250 , $y = 1.04 x$, $\text{adj. } R^2 = 0.86$



<分析結果>

- 自由度調整済み決定係数が 0.86 であるため、プロジェクトの計画工期の 86%は、標準工期（= $3.23 \sqrt[3]{\text{計画工数}}$ ）によって説明できる。

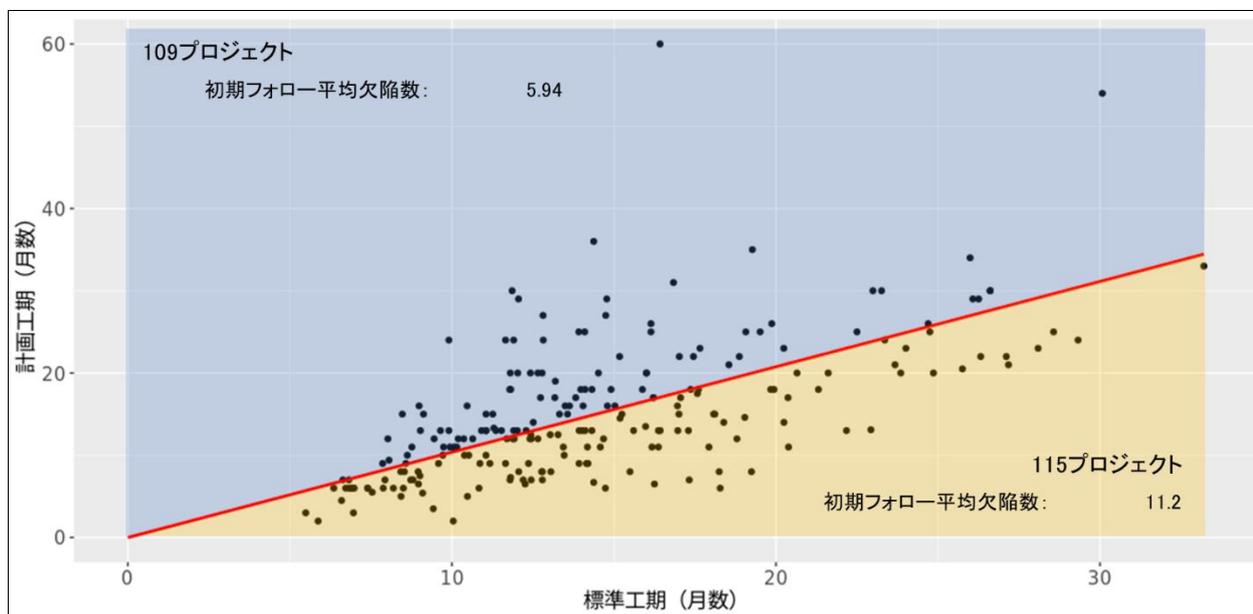
<活用方法・考察>

- プロジェクトの計画工期は、標準工期の 1.04 倍とほぼ標準工期に近い。
- プロジェクトの計画時の工期は標準工期とほぼ合致しているが、2-3 に示した通り、プロジェクトの実績工期は標準工期よりも長くなっていることに留意されたい。
- プロジェクト予定（計画）工期遵守を優先する場合に注意すべきこととして以下の点が考えられる。「納期優先の意識付け」、「要求仕様の明確化と仕様変更の最小化などの納期遵守の工夫」、「標準工期より余裕をもった計画工期を設定する」

⁵ ソフトウェア・メトリクス調査 2020 の図表 6-2-11-2 と対応している。

標準工期と計画工期の関係性が、初期フォロー平均欠陥数と関連があるかを定量的に分析した。

図表 2-5-2 計画工期が標準工期を超えるプロジェクトと標準工期以下のプロジェクトの特徴



図表 2-5-3 データプロファイル

	計画工期が標準工期以上	計画工期が標準工期未満
プロジェクト数	109	115
平均計画工期	19.5	11.2
平均欠陥数 (ユーザー総合テスト)	27.5	28.3
平均欠陥数 (初期フォロー)	5.94	11.2

<分析結果・考察>

- 平均欠陥数 (ユーザー総合テスト) は、計画工期が標準工期以上のプロジェクト(27.5)と計画工期が標準工期未満のプロジェクト(28.3)と大きな差異はなかった。
- 平均欠陥数 (初期フォロー) は、計画工期が標準工期以上(5.94)と比較して、計画工期が標準工期未満(11.2)の稼働後の欠陥数が多い。計画工期が相対的に短いことによって、システム使用方法の検討が浅かったり、例外事象への配慮が不十分であったり、各業務項目の工期を過小に見積もったことによって負担が生じたりしていることで、欠陥数が相対高位となっていると考えられる。

(2-6) 品質の評価

- ユーザーが、ユーザー総合テストと初期フォローフェーズにおいて発見した不具合を欠陥としてカウントし、その影響度の大きさによって大中小（重度・中度・軽度）に分類した。

欠陥（重度）＝システムにとって致命的で緊急対応が必要な障害であり、5人日以上の負荷を要する場合

欠陥（中度）＝システムにとって致命的ではないが緊急対応が必要な障害（大でも小でもない障害）で、その復旧に要する時間が中程度である。

欠陥（軽度）＝軽微で緊急対応の必要がない程度の障害、その復旧に要する時間は半日以内で対応可能。

(2-7) 換算欠陥数、換算欠陥率

- 検出された欠陥を、次式にもとづいて「換算欠陥数」「換算欠陥率」として算出し指標化した。

換算欠陥数（重み付け欠陥数）＝ $2 \times (\text{欠陥数}_大) + 1 \times (\text{欠陥数}_中) + 0.5 \times (\text{欠陥数}_小)$

換算欠陥率（重み付け欠陥率）＝ $\text{換算欠陥数} \div \text{全体工数}$

- 換算欠陥率によって、A ランク（換算欠陥率=0）～F ランク（換算欠陥率 ≥ 3 ）に分類している

以降、プロジェクト品質を換算欠陥率によって6段階のランクに分類し、分析した。

A ランク：換算欠陥率=0

B ランク： $0.00 < \text{換算欠陥率} < 0.25$

C ランク： $0.25 \leq \text{換算欠陥率} < 0.50$

D ランク： $0.50 \leq \text{換算欠陥率} < 1.00$

E ランク： $1.00 \leq \text{換算欠陥率} < 3.00$

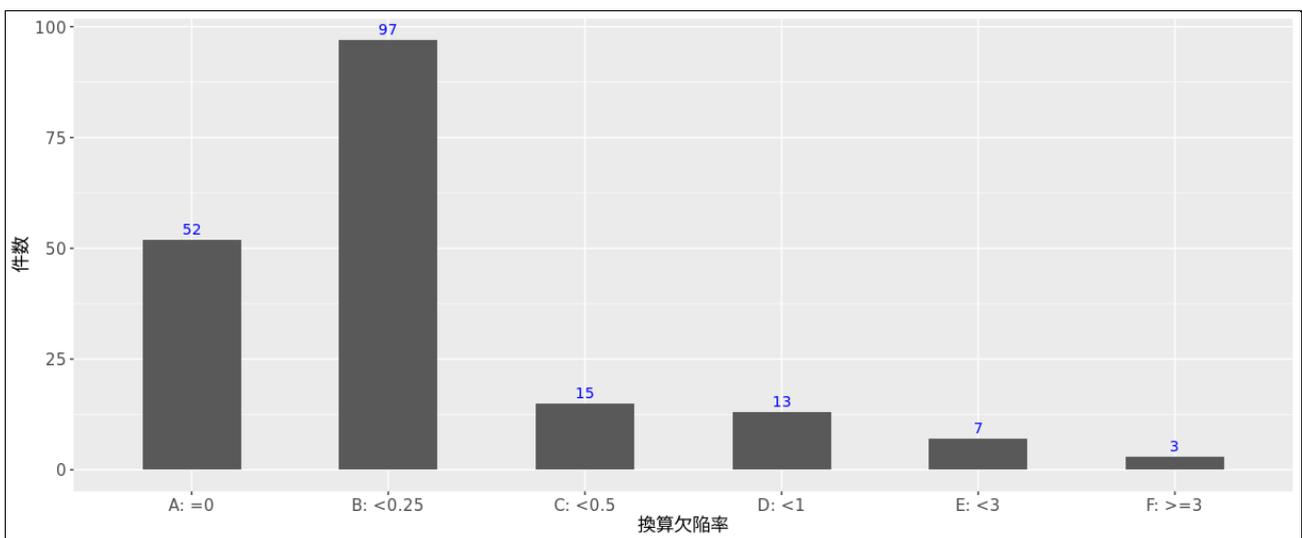
F ランク： $3.00 \leq \text{換算欠陥率}$

品質目標：換算欠陥率目標 = 換算欠陥数 / 開発工数 = 0.05 以下

<使用データ説明>

- 分析対象データは、品質目標・実績工数・欠陥数が記載されたプロジェクト 187 件。
- 換算欠陥率の平均値は 0.25、中央値は 0.05。

図表 2-7-1⁶ 換算欠陥率のヒストグラム



<分析結果・考察>

- 換算欠陥率の中央値が 0.05 であることから、換算欠陥率の目標を 0.05 とするのが妥当である。
また、換算欠陥率の平均値が 0.25 であることから、換算欠陥率の目標を 0.25 とすることも有益である。

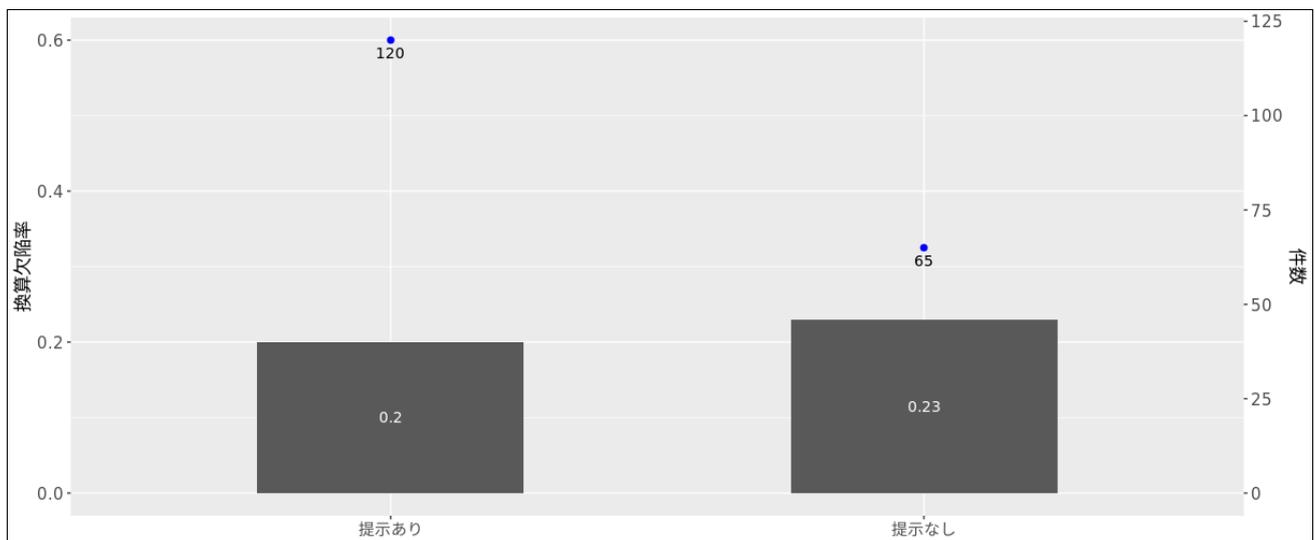
⁶ ソフトウェア・メトリクス調査 2020 の図表 6-5-1 と対応している。

品質目標の提示は、換算欠陥率を減らす傾向がある

<使用データ説明>

- 分析対象データは、品質目標・実績工数・欠陥数が記載された 187 プロジェクトから、F ランク（換算欠陥率が 3.00 以上）の 2 プロジェクトを除いた 184 プロジェクト。
- 品質目標提示ありのプロジェクトは 120 件、提示なしのプロジェクトは 65 件。

図表 2-7-2⁷ 品質目標の提示の有無による換算欠陥率の違い



<分析結果・考察>

- プロジェクトの 2/3 は品質目標を提示しており、品質目標を提示しているプロジェクトの換算欠陥率(0.2)は、品質目標を提示していないプロジェクトの換算率(0.23)と比較して、抑制されている。

⁷ ソフトウェア・メトリクス調査 2020 の図表 6-5-24 と対応している。

仕様が不明確、大きな仕様変更が発生するほど品質は劣化する

<使用データ説明>

- 分析対象データは、要求仕様変更の記載がない、または、適正工数が 10,000 人月を超えるプロジェクト 5 件を除いた 182 プロジェクト。

図表 2-7-3⁸ 要求仕様の変更と換算欠陥率

要求仕様変更	件数	平均換算欠陥率	最大換算欠陥率
変更なし	50	0.192	3.125
軽微な変更	97	0.213	3.404
大きな変更	33	0.481	4.576

<分析結果・考察>

- 要求仕様変更の程度が大きくなるにつれて、平均および最大換算欠陥率が増加した。
- アンケートでは「重大な変更」があったプロジェクトについても回答を得ているが、当該プロジェクトが 2 件とサンプル数が少なかったため、分析の対象外としている。

⁸ ソフトウェア・メトリクス調査 2020 の図表 6-5-44 と対応している。

(2-8) 開発総費用

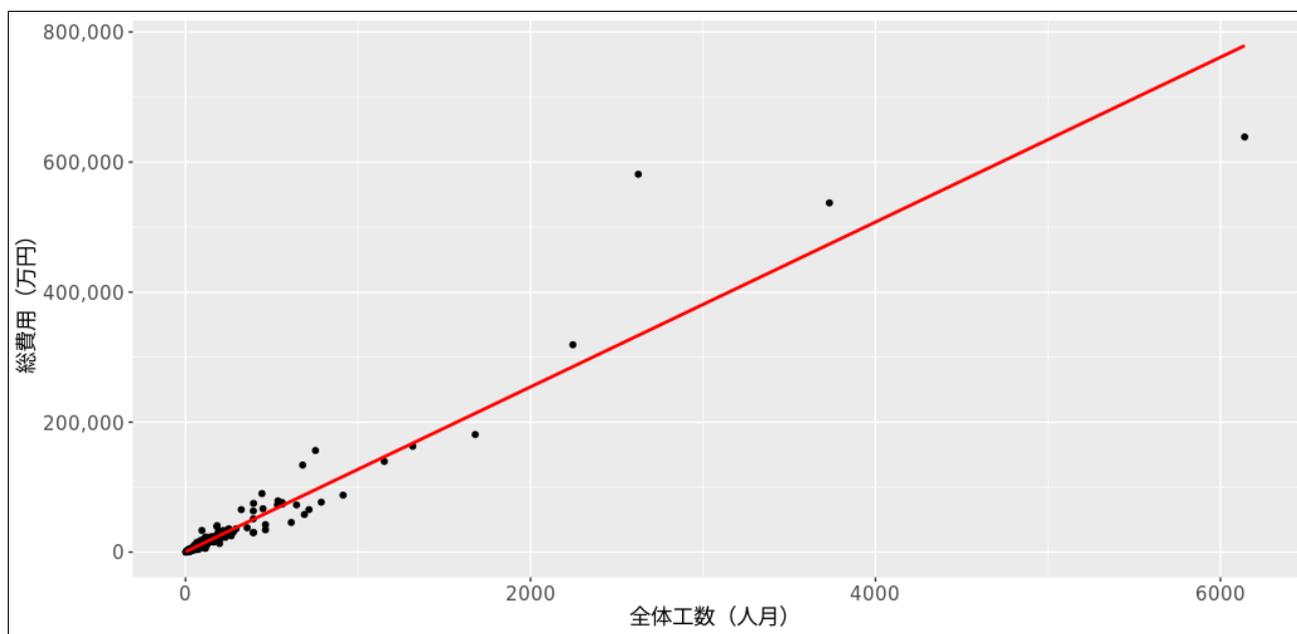
費用と工数の関係： $y = 127.14 x$

x ：実績工数（全体工数）、 y ：開発総費用（万円）

<使用データ説明>

- 分析対象のデータは、195 件。
- 全データ 366 件のうち、費用・工数・業務パッケージの活用方針が明確なデータ 223 件を対象とし、一般性を確保するために、単価が 50 万円以上 400 万円以下の 195 プロジェクトを分析対象とした。

図表 2-8-1⁹ 全体工数と総費用の関係、サンプル数 = 195 / 221 , $y = 127.14 x$, $\text{adj. } R^2 = 0.92$



<分析結果・考察>

- 自由度調整済み決定係数が 0.92 であるため、総費用の 92%は、上記メトリクスによって説明できる。
- 回帰係数が 127.14 であるため、分析対象データに対する人月単価は 127 万円が妥当である。
- パッケージの活用によって、外注コストにおける人月単価の違いが見られた。(詳細は後述)

⁹ ソフトウェア・メトリクス調査 2020 の図表 6-7-3 と対応している。

(2-9) 工数区分別 平均単価

図表 2-9-1¹⁰ スクラッチ開発プロジェクトの工数区分別の平均単価

工数区分	<10人月	<50人月	<100人月	<500人月	>=500人月	全体
件数	6	23	22	36	12	99
外注コスト平均	850	2951	7570	19381	146901	27273
工数平均	5	31	70	205	1542	284
加重平均単価	185	95	109	95	95	96

<使用データ説明>

- 分析対象データは、外注コストが記載されているスクラッチ開発プロジェクト 99 件。

<分析結果・考察>

- 全体の加重平均単価は 96 万円であった。
- 開発規模が 10 人月を超えるプロジェクトでは、開発工数規模による単価の差異は小さかった。スクラッチ開発は広く用いられ、実績が蓄積されている開発手法であるため、プロジェクト規模による影響が小さいものと考えられる。

図表 2-9-2¹¹ パッケージ利用開発プロジェクトの工数区分別の平均単価

工数区分	<10人月	<50人月	<100人月	<500人月	>=500人月	全体
件数	1	13	8	12	5	39
外注コスト平均	560	2890	10407	28844	166583	33344
工数平均	9	31	70	221	1089	232
加重平均単価	63	94	150	130	153	144

<使用データ説明>

- 分析対象データは、外注コストが記載されているパッケージ利用開発プロジェクト 39 件。

<分析結果・考察>

- 全体の加重平均単価は 144 万円であった。
- パッケージ利用開発の全体加重平均単価 (144 万円) は、スクラッチ開発の全体加重平均単価 (96 万円) よりも高い結果となった。パッケージ利用開発は、パッケージ使用のスキル・ノウハウが必要とされるため、スクラッチ開発と比較して高単価となっていると考えられる。また、パッケージ開発技術者が需要に対して十分に供給できていない状況にあるとも考えられる。
- 開発規模が 100 人月以上のプロジェクトでは、100 人月未満のプロジェクトよりも単価が高い結果

¹⁰ ソフトウェア・メトリクス調査 2020 の図表 6-7-5-3 と対応している。

¹¹ ソフトウェア・メトリクス調査 2020 の図表 6-7-5-4 と対応している。

となった。開発規模が大きくなることで、パッケージについてのさらに高度なノウハウが必要となるなどの理由によって、高単価人材が必要になったと考えられる。

図表 2-9-3 SaaS 利用開発プロジェクトの工数区分別の平均単価

工数区分	<10人月	<50人月	<100人月	<500人月	>=500人月	全体
件数	0	4	6	5	0	15
外注コスト平均		3715	8010	42832		18472
工数平均		40	67	273		128
加重平均単価		93	120	157		144

<使用データ説明>

- 分析対象データは、SaaS を使用したプロジェクト 15 件。

<分析結果・考察>

- 全体の加重平均単価は 144 万円で、パッケージ利用開発プロジェクトの全体の加重平均単価(144 万円)に等しかった。
- SaaS 利用開発の加重平均単価がスクラッチ開発に対して高位となり、パッケージ利用開発と同程度となっているのは、SaaS 利用開発においても SaaS 使用のスキル・ノウハウが必要とされるためと考えられる。
- 尚、SaaS 利用開発プロジェクトの工数と単価については、サンプル数が少ないため、参考情報としていただきたい。

【コラム】本ガイドブックで用いた分析手法

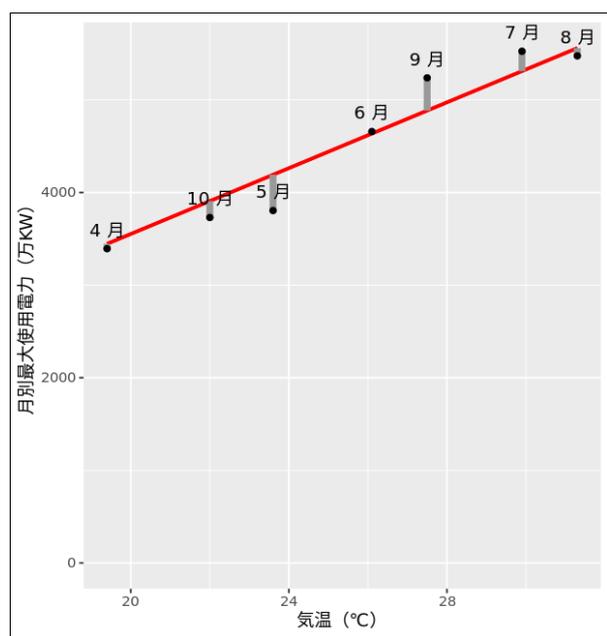
本ガイドブックでは、回帰分析を実施しています。より詳細には、線形単回帰分析¹²を用いており、単というのは y に対して一つの変数 x のみとの関係性を考慮するということの意味し、線形というのは $y = ax + b$ のように x と y の関係が平面上の直線として表現できることを意味します。例えば、 y を費用（万円）、 x を工数（人月）とし、 x と y の間に $y = 127.14x$ で表される関係が成立していることがわかったとします。前述の関係式（回帰式）がわかれば、既知の数値から未知の数値を推定できるため、100人月のプロジェクトにかかる費用を1億2714万円と推定できます。

次の左のグラフは、気温¹³と月別最大使用電力¹⁴のデータを回帰分析した結果です。各データ点との距離（グレーの縦線の長さ、残差と呼ばれます）が最小になる直線を表す回帰式

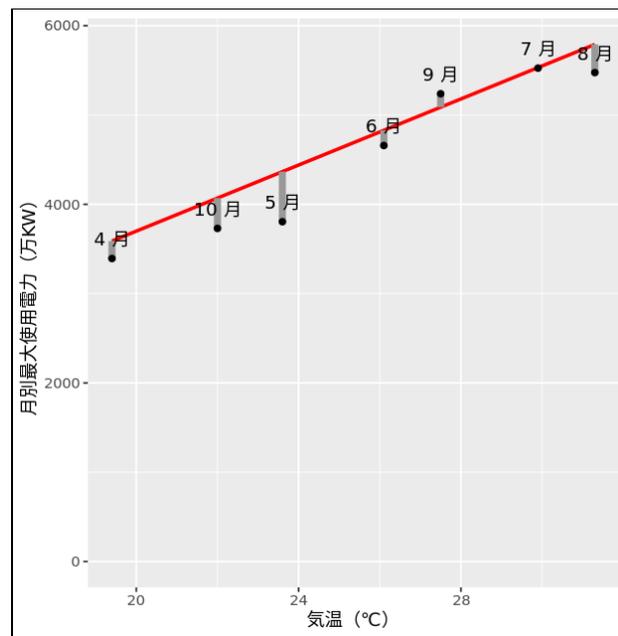
$$\text{「月別最大使用電力} = 177.62 \times \text{気温} \text{」}$$

を求めています。右のグラフは比較用に「月別最大使用電力 = 185 × 気温」の直線を引いています。左のグラフより残差が大きくなっているのがわかります。

回帰分析の結果の残差を示すグラフ



最適ではない係数に対する残差を示すグラフ



前記のイメージに基づいて、線形回帰分析に関して、より一般的に説明します。数値を知りたい変数（前記の月別最大使用電力）を目的変数と言い、そのために用いる変数（前記の気温）を説明変数と言

¹² 線形単回帰分析は、複数の変数 ($z = ax + by + c$ など) を用いる線形重回帰分析や、多項式回帰分析などの非線形の回帰分析に比べるとシンプルですが、読者の理解の一助になればという意図で本コラムにおいて説明します

¹³ 気象庁、過去の気象データ検索、<https://www.data.jma.go.jp/stats/etrn/index.php>

¹⁴ 東京電力、月別使用最大電力（1日最大、サービスエリア内全ての電力需要の合計値）、<https://www.tepco.co.jp/corporateinfo/illustrated/power-demand/peak-demand-monthly-j.html>

います。説明変数の数は複数であることが多いです。前記の例のように、一つの要因で事象を説明できることは少ないためです。以下、説明変数の数を k 個と想定します。

求めたい式は、目的変数を y 、説明変数を $x_i (i=1, 2, \dots, k)$ と書くとする、

$$y = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k$$

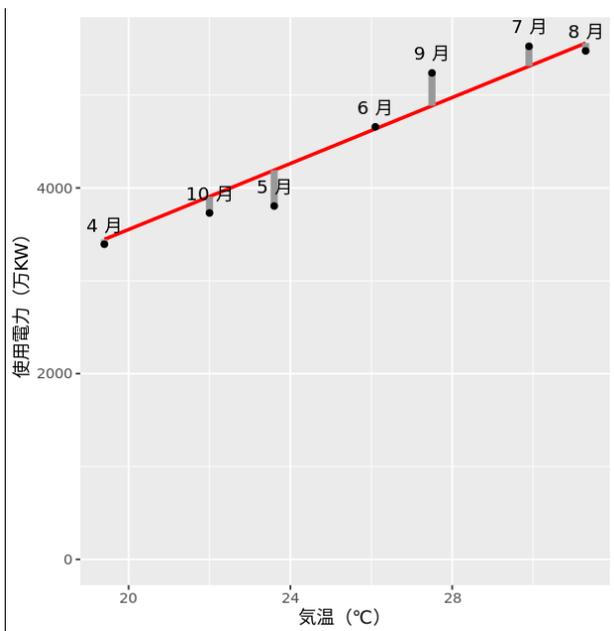
です。回帰分析では、 $\beta_0, \beta_1, \dots, \beta_k$ の値を適切に定めます。適切にというのは、残差の二乗の和を最小化するという意味です。この残差の二乗の和は、残差平方和 (Residual Sum of Squares, RSS) と呼ばれます。最小化には最小二乗法が用いられますが、その詳細については関連する文献に譲りたいと思います。ここでは、回帰式の係数はコンピュータが適切に定めてくれる、という理解を頂ければ幸いです。

回帰式を算出したら、その回帰式を評価します。回帰式はあくまで残差平方和を最小化したもので、あてはまりが悪い場合もあるためです。評価指標としては、決定係数 (R^2)、特に自由度調整済み決定係数 (adjusted R^2) が用いられます。

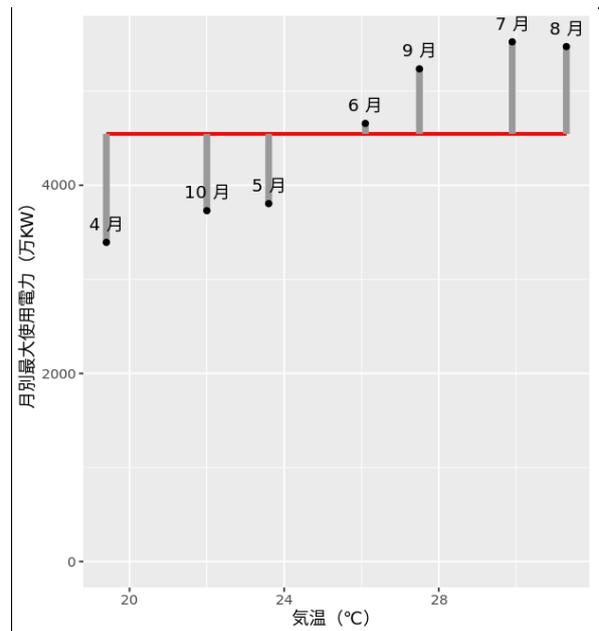
R^2 は、直観的には次の二つのグラフの残差の違いです。左のグラフの回帰式に対する残差平方和を RSS、右のグラフに示した目的変数 (例えば月別最大使用電力) の平均との差の二乗和である全平方和 (Total Sum of Squares) を TSS と書くと、 R^2 は次のように算出します。

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}$$

RSS はグレーの線の長さの平方和



TSS はグレーの線の長さの平方和

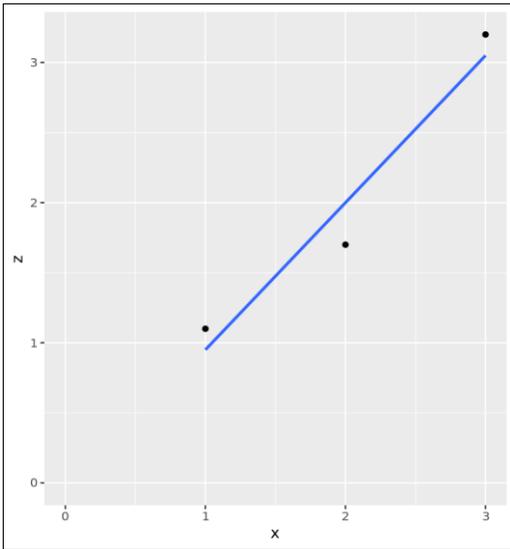


R^2 の値が大きいほど、適合度が高い、つまり目的変数の変動を説明変数で説明できる、ことを意味します。 R^2 の最大値は、 $RSS=0$ のときの 1 であり、目的変数の値と、説明変数と回帰式から求めた値が完全に一致する場合に相当します。最小値は $RSS=TSS$ のときの 0 であり、右図のように目的変数を常に平均で推定する場合に相当します。目的変数 y の値が n 個あるとき (データが n 個あるとき) の TSS と RSS は、 y の平均を \bar{y} 、回帰式から求めた y の値を \hat{y} とすると、次のように記述できます。

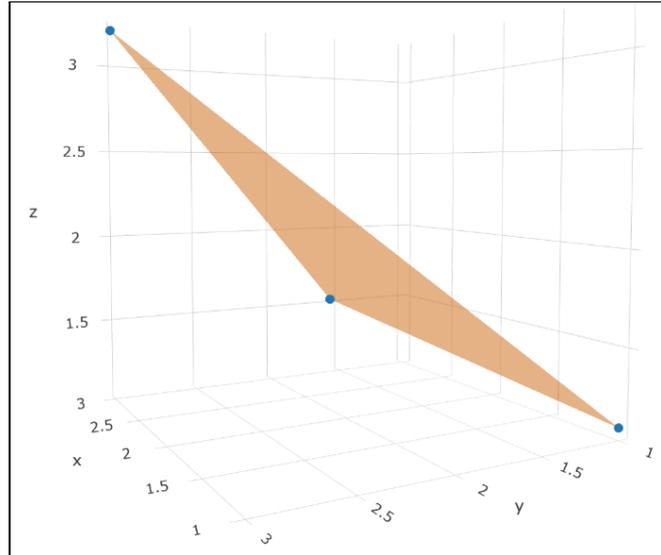
$$TSS = \sum_{i=1}^n (y_i - \bar{y})^2 \quad RSS = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

決定係数 R^2 は広く用いられていますが、より正確には自由度調整済み決定係数 (adjusted R^2) が用いられます。理由の一つは、目的変数との関連性がない説明変数であっても、追加するだけで R^2 が大きくなるためです。例えば、 $(x, z) = (1, 1.1), (2, 1.7), (3, 3.2)$ というデータがあったとします。これを平面上にプロットしつつ回帰直線を引いたのが次の左図です。見て取れる通り、 R^2 は 1 未満です。ここで説明変数 y を追加して、 $(x, y, z) = (1, 1, 1.1), (2, 2.1, 1.7), (3, 3, 3.2)$ というデータに対して回帰した結果を平面として表したのが右図です。空間上の平面は 3 点で固定できるため、 R^2 は 1 になります。

3 データに対する回帰直線



3 データに対する回帰平面



Adjusted R^2 を説明する前に、母分散と標本分散の違い、及び不偏分散に触れます。一例として、全ての日本人 (仮に 1 億人) の身長データを母集団、そのうち 1000 人の身長データを標本としてランダムに抽出したとします。母分散は母集団 (1 億個のデータ) の分散で、標本分散は標本 (1000 個のデータ) の分散です。母分散は前述の TSS をデータ数で割った値になり、各データの身長から減じる値は母集団の平均 (母平均) です。一方で、標本分散は 1000 個のデータとその平均 (標本平均) から同様に求めますが、母平均と標本平均は別の値です。これに起因して、標本分散ではなく、TSS をデータ数よりも 1 少ない値で割った値を用いるのが適切です。これは不偏分散と呼ばれます。不偏分散の式の導出に興味のある読者は、関連する文献等を参照頂ければと思います。

Adjusted R^2 は、データ数を n 、説明変数の数を k として、次のように計算されます。

$$adj. R^2 = 1 - \left(\frac{RSS}{TSS} \times \frac{n-1}{n-k-1} \right)$$

R^2 は $1 - (RSS/TSS)$ でした。ただし、母平均と標本平均の差異を調整するために TSS は $(n-1)$ で割って不偏分散にすべきで、RSS ではさらに説明変数の数を考慮して $(n-k-1)$ で割るべきです。この調整を反映した数値が、自由度調整済み決定係数になります。

参考：調査票でのフェーズの呼称と SLCP との対応表

2018 年版呼称	SLCP プロセス/アクティビティ	SLCP の定義
企画段階		【SLCP 定義にない本調査独自アクティビティ】 投資申請、承認する前段階のことを示しています。
要件定義	システム計画の立案 システム要求分析 ソフトウェア要求分析	企画者は、システム計画の基本要件の確認を行い、実現可能性の検討、スケジュール作成、システム選定方針の策定、プロジェクト推進体制の策定、システム移行やシステム運用・保守に対する基本方針の明確化、環境整備・教育訓練・品質に対する基本方針の明確化を行い、計画を作成・承認を受ける。 開発者は、品質特性仕様を含めて、ソフトウェア要求事項を確立し文書化する。また、設定した基準を考慮して、ソフトウェアの要求事項を評価し文書化。さらに、共同レビューを行い、要求事項に関する基準線を確立する。
設計から統合（結合）テスト	システム方式設計 ソフトウェア方式設計	開発者は、ソフトウェア品目に対する要求事項をソフトウェア方式に変換する。最上位レベルのソフトウェア構造、コンポーネント、データベースの最上位レベルでの設計、利用者文書の暫定版の作成、ソフトウェア結合のための暫定的なテスト要求事項及び予定等を明らかにする。また、共同レビューを実施する。
	ソフトウェア詳細設計	開発者は、ソフトウェア品目の各ソフトウェアコンポーネントに対して詳細設計を行う。ソフトウェアコンポーネントは、コーディング、コンパイル及びテストを実施するユニットレベルに詳細化する。また、インターフェイス、データベースの詳細設計、必要に応じて利用者文書を更新、ユニットテストのためのテスト要求事項及び予定を定義する。共同レビューを実施する。
	ソフトウェアコード作成及びテスト	開発者は、ソフトウェアユニット及びデータベースを開発する。また、それらのためのテスト手順及びデータを設定する。さらに、テストを実施し、要求事項を満足させることを確認する。これらに基づいて、必要に応じて利用者文書等の更新を行う。
ユーザー総合テスト	ソフトウェア結合 システム結合 ソフトウェア適格性確認テスト システム適格性確認テスト	開発者は、ソフトウェアユニット及びソフトウェアコンポーネントを結合して、ソフトウェア品目にするための計画を作成し、ソフトウェア品目を完成させる。また、結合及びテストを行う。必要に応じて利用者文書等の更新を行う。共同レビューを実施する。 開発者は、ソフトウェア品目の適格性確認要求事項に従って、適格性確認テストを行う。必要に応じて利用者文書等の更新を行う。また、監査を実施する。
	ソフトウェア導入支援 ソフトウェア受け入れ支援	開発者は、契約の中で指定された実環境にソフトウェア製品を導入するための計画を作成し、導入する。 開発者は、取得者によるソフトウェア製品の受け入れレビュー及びテストを支援する。また、契約で指定するのとおり、取得者に対し初期の継続的な教育訓練及び支援を提供する。
初期フォロー		【SLCP 定義にない本調査独自アクティビティ】 リリース後数か月、開発プロジェクト体制がフォローしている時期を示しています
	運用プロセス	ソフトウェア製品の運用及び利用者に対する運用支援を行う。運用者は、このプロセスを管理するために具体化した管理プロセスに従って、運用プロセスの基盤となる環境を確立する、など。

(備考 1) SLCP の定義は、規格のアクティビティを要約したものである

第3章 JUAS ソフトウェア・メトリクス活用事例インタビュー

【事例1】大規模や未経験案件の事前リスク確認、失敗プロジェクトの事後評価に利用

A社：

業種	諸工業・印刷業／ITグループ会社
----	------------------

今回調査対象のご活用 JUAS ソフトウェア・メトリクス
<ul style="list-style-type: none">・全体工期と全体工数の関係・標準工期と実績工期の対比・総費用（万円）と全体工数（人月）の関係・工数区分別工数単価（予算／人月）・工数区分別・品質区分別の工数単価

1. A社の状況

スクラッチ開発が主流であったが、最近はSaaSサービス利用によるシステム導入が増えている状況にある。スクラッチ開発の場合、FP法や類推法を基に工数見積もりを実施しているが、FP法については自社の有識者が工数見積もりツールを作成して、多くの案件で活用している。見積内容は、上位者、有識者への見積レビューを実施し、過去の経験や案件特性を考慮しての指摘を行い、当該案件の見積内容の精査を行っている。

自社のメトリクスを保有している訳ではないが、自社の工数見積ツールにより、プロジェクト全体の工数・工期算出の標準化、妥当性向上をはかっている。また、外部委託先の工数・工期妥当性確認にも活用することがある。

算定された工数に対して、工程、人材レベルに応じて設定された単価を乗じてプロジェクト全体のコストを試算するプロセスを確立している。

2. メトリクス活用の組織体制

現状、メトリクスを公式に活用していないが、現場開発部門や品質保証部門で、頻度は少ないながらメトリクスを第三者視点の評価として参考にすることがある。

品質保証部門は、IT全般統制・PMO・品質保証の3つの機能で構成されており、7名を配置している。IT全般統制とPMOとしての大型案件の開発進捗状況の追跡と共に、今後、プロジェクトリスク低減や障害の再発防止活動などの品質保証業務の強化を検討していくが、メトリクス活用を広く活用する方針であれば品質保証部門が推進役の体制になる可能性がある。

3. JUAS ソフトウェア・メトリクスの活用方法

① 工数と工期

1)類推法の適用が難しい過去経験したことの無い案件、大規模な案件やリスクの高い案件など、

過去実績を基に策定している社内基準では不安を感じるプロジェクトに対し、JUAS メトリクスによる評価との対比によって見積もりの妥当性を確認することがあるが、全社で、常時 JUAS メトリクスを活用している状況ではない。また、パートナー（外部委託）の提示した工数の妥当性評価時においても、同様の活用をすることがある。

2)実績工数管理に基づくプロジェクト収支分析を行っている。社内ルール化は行っていないが、赤字案件に対し、見積もり時の工期の妥当性を確認するために、JUAS のメトリクスを用いて振り返ることがある。

② 単価

親会社とは役割別単価（個人の経験年数に基づく工程別・ランク別の単価）を用いて契約しているが、親会社の業務に精通している強みと信頼関係から、本単価の妥当性検証などには利用されていない。

③ 品質

JUAS が提示している品質指標は現在利用しておらず、品質管理に役立つ指標や考え方について、社内で検討を行っている状況。検討にあたって、JUAS の品質メトリクスの考え方は参考になると捉えている。

現状の社内での品質保証の考え方は、メトリクスよりも、プロジェクトの重点マイルストーン時に、要件や設計内容の確認、テストや移行計画の確認することと、稼働後の品質障害発生時の原因分析と情報共有、再発防止策の展開を主に実施している状況にある。

4. 課題

SaaS サービスやパッケージなどによるシステム導入方法の変化、開発手法や開発環境の変化に対し、既存の工数見積手法、品質担保のための施策も状況にあわせて変化していく必要があるが、有効性高い手法、施策、指標等をどのように策定していくべきかが具体化できていない。第 3 者の考え方、指標などを参考にしながら、客観的な妥当性の高い方策を実施していきたい。

【事例2】経営層への説明に業界標準として利用

B社：

業種	金融業
----	-----

ご活用の JUAS ソフトウェア・メトリクス
<ul style="list-style-type: none">・投入工数別フェーズ別新規改修区分別工期比・全体工期と全体工数の関係・標準工期と実績工期の対比・総費用（万円）と全体工数（人月）の関係・工数区分別工数単価（予算／人月）

B社における生産性指標の活用事例

1. B社の状況

B社は、国内システムの約6割をスクラッチで開発している。システム統合などにより複雑化したシステムの生産性向上を目的に、2015年からメトリクス評価を開始した。

①生産性

B社は自社の生産性目標管理手法を策定するため、過去プロジェクトのデータ（約1,500件／2010年～2017年）を分析。言語別STEP数と工数実績（要件定義～移行）を基に、見積もりKSTEP数から総工数を予測する算出式を導出し（R2：0.845）、自社内の他プロジェクトや他社ベンチマークと生産性を比較している。生産性管理は社内で広く活用されているが、アジャイル系開発やパッケージ利用開発への適用は未済である。

②工数

一定金額規模以上のプロジェクト（パッケージ開発などKSTEPでの評価が適合しないシステム開発を除く）では、開発工数見積もり・実績の生産性評価をルール化し、審査部門が見積もり審査・実績評価を行っており、評価にあたっては自社データ以外に外部データも参考にしている。また、社内の標準工数から大きく外れる見積もりに対しては、差異が発生している理由を確認・分析している。

③工期

工数規模に対する開発期間の妥当性評価にはCOCOMO標準を採用している。また、開発着手前に自社独自のリスク・アセスメントツールを用いてプロジェクト・リスクを確認しているが、COCOMO標準を用いた工期の評価結果もリスク・アセスメント項目の一つとしている（工期が短い場合は、リスク事項として取り扱う）。なお、リスク・アセスメントツールは、各種リスク項目とプロジェクト苦戦の相関分析結果に基づいて制定されており、特にプロジェクトの成否を決める上流工程での準備不足リスクに着目して評価している。

④ 品質

品質評価に関する標準的な自社メトリクスは策定できていない状況である。開発部署は「不具合密度」や「テスト件数」などで品質を管理しているが、多くの業務領域で高いテスト品質（≡不具合が限りなくゼロに近い状態）を求めるため、テスト品質と生産性との相関を分析することが難しい（データが少ない）ためである。また、品質を確保するために十分な期間を確保するため、プロジェクト期間は長くなる傾向にある。

2. メトリクス活用の組織体制

自社プロジェクト・メトリクスの管理はシステム企画部門が担っており、生産性管理に必要なデータ収集・加工に約 0.3 人、生産性や開発期間の分析・評価に約 0.2 人の業務量で対応している。なお、データ収集・加工のプロセスは自動化等を進めることで効率化を図っている。

3. JUAS ソフトウェア・メトリクスの活用事例

① 工程別工数比率（自社メトリクス）の外部比較データとして利用

自社メトリクスの SLOC 総合生産性業界比較（開発工数見積・実績評価シート）に、システム別標準（理論値工数）と工程別工数比率を掲載し、参考情報としている。

② 経営層への工程別工数比率の妥当性説明

過去に開発スピード向上やコスト削減を目指し、トップダウンで開発工数全体のうちの要件定義の比率やテスト工数比率の削減を推進したが、それによりプロジェクトが苦戦するケースが増加した。この状況に対して JUAS の工程別工数比率を参考にすることで、経営層に対して適切な工数配分の必要性を説明し、理解を得ることができた。

4. 課題

パッケージ開発における生産性を評価する指標がないことやアジャイル開発などスピード重視型の開発に対して従来型の COCOMO の指標は適用できない点が課題である。また、高いテスト品質（≡不具合が限りなくゼロに近い状態）が社内の標準的な基準となっている中、日本が求める品質に届かない海外パッケージをどのように評価していくべきかなど、開発環境の変化に伴う自社メトリクスの見直しが必要である。

AI による生産性評価スキーム導入の大きな方向性も見据えているが、JUAS 始め同じ課題を共有する企業との情報交換の場の必要性も感じている。

【事例3】親会社との共通言語として活用

C社：

業種	エネルギー業/ITグループ会社
----	-----------------

ご活用のJUASソフトウェア・メトリクス
<ul style="list-style-type: none">・全体工期と全体工数の関係・標準工期と実績工期の対比・換算欠陥率の度数分布と基本統計量・総費用（万円）と全体工数（人月）の関係

1. C社の状況

開発案件・保守対象システムの殆どが親会社から受託する基幹系システムで、スクラッチ開発が大部分を占めている。開発方式もウォーターフォール型が殆どだが、親会社はクラウドサービスやパッケージの利用が進みつつある。

① 工数

新規開発案件は、親会社と合意した見積方法（FP法）を用いて、プロジェクトの特性を加味し、外部委託分を含め自社にて設計工程以降を見積もっている。エンハンス案件は、見積算定時に親会社から提示された見積算定シートに、要求規模に対するStep数（言語換算あり）や工程や業務別の単価を入力することで、見積が作成される。インフラなどStepを持たない案件は、作業積算による見積もりを実施している。いずれの場合においても、プロジェクト体制図を併せて提示している。

尚、企画工程の見積もりは、過去のプロジェクトでの検討期間などを参考とした類推法にて行っている。

② 品質

システム規模やプロジェクトの特性に応じて指摘数やレビュー時間、テストケース数や欠陥数などのメトリクスに目標を定め、レビューで確認している。単位あたりFPやキロStepについて平均指摘数、平均欠陥数などの社内標準のメトリクスを保有している。社内標準は年間で見直しを実施している。

2. メトリクス活用の組織体制

品質部門がメトリクスを活用している。特に非機能要件は品質確保に重要となることから、2024年度に受託案件のプロジェクト支援・モニタリングを行う機能を再編し、品質部門は上流工程における非機能要件の確実な反映を支援する機能を持つ組織に統合している。経営層へのレポート、プロジェクト品質計画の審査やプロジェクト品質結果の実績まとめ（障害発生件数や障害原因の特定）、機能要件と非機能要件のプロジェクトマネジメントを支援する機能を担っている。プロジェクトマ

ネジメント 3 人、品質管理 3 人、非機能要件策定 10 人の支援体制（外部委託を除く）となる。

3. JUAS ソフトウェア・メトリクス の活用方法

① 親会社との共通言語として活用

受託開発のゲート（契約審査）では、親会社が全体工期と工数の関係について、JUAS メトリクスを参考にチェックで活用しており、標準からの乖離に対するリスクや対策内容を確認している。JUAS ソフトウェア・メトリクスが両社の共通言語とされている。

② 業界動向の把握

信頼できるハイレベル企業のプロジェクトデータが集約されているので、業界動向の把握に活用している。メトリクス策定の背景にある考え方が、プロジェクト推進時に必ずやらなければならないこと、やらなくてもいいことの判断に参考となる。特に、企画工程で作成するドキュメントが非常に多く、昨今のプロジェクト短期化に対応するため、やらなくてもいいことの精査に利用している。また、非機能要件は品質に大きく影響するため、特に大規模案件が非機能要件の検討工数を確保する際にも有効となる。

経営層にも「世の中の統計値はこうである」と客観的に説明することができる。「教科書」ではなく、「リファレンス」として活用している。

4. 課題

特になし

第4章 あとがき

2020年ソフトウェア・メトリクス調査報告書の発行以来、4年間調査を中断していましたが、JUAS会員の皆さまにアンケート方式で報告書の利用状況を調査したところ、主なメトリクスについて活用されていることが確認できました。今回のガイドブックでは従来の多岐にわたる調査項目を見直し、活用度の高いメトリクス上位項目に絞り込んで調査を実施したことで、多くの有効回答数を得ることができました。調査方針見直しについてご理解いただくとともに、メトリクス調査にご協力を頂いた全ての皆さまに感謝を申し上げます。

また、活用状況調査時にソフトウェア・メトリクス調査報告書をご存知でない方も多く見受けられました。ソフトウェア開発環境は、スクラッチ開発型からアジャイル開発型、パッケージソフト利用型、クラウド利用型などへと変化しつつありますが、まだまだスクラッチ開発に取り組んでいる企業も多いです。本ガイドラインでは、今回取り上げたメトリクスをどのように利用しているのかについての活用事例を紹介することで、各社が自社の活動に取り込んでいただけることを期待しています。また、インタビューに対応いただいた3社にお礼を申し上げます。

情報システムが、SoR (System of Record) から SoE (System of Engagement) へ多様化し、世の中の変化に迅速に対応していくことが求められる状況下においては、ウォーターフォール開発からアジャイル開発へ、自前のインフラからクラウド利用へ、作るシステムから使うシステムへ、プロセスドリブンからデータドリブンへ、ベンダーへの発注から自社でイニシアチブをとることへと、企業もシステム開発・保守に携わる人の意識も変わらざるを得なくなってきました。また今後も変わり続けることが予想されますが、形態を変えていくとはいえ情報システムが使われ続け、進歩していくことは間違いありません。

2026年ソフトウェア・メトリクス調査では、今回の調査に加え、これらソフトウェア開発環境変化に伴い、生産性・品質についてどのように考えていけば良いのかについてのプロジェクトを発足させます。皆さまが抱える課題に協会として対応していけるよう、努めていきますので、引き続きご理解とご協力を賜りますよう、よろしく申し上げます。

最後に、本ガイドブックが皆さまのお役に少しでも立つことを願っております。

2025 年版

「ソフトウェア・メトリクス調査 2025」ガイドブック

発行日：2025 年 4 月

発 行：一般社団法人 日本情報システム・ユーザー協会

〒104-0045 東京都中央区築地 1-13-14 NBF 東銀座スクエア 2 階

TEL 03-6264-1312

URL <http://www.juas.or.jp/>

(禁無断転載)