

ユーザー企業 ソフトウェアメトリックス ＝ 要点ハンドブック ＝

- (1) はじめに
 - (1-1) プロダクト志向とプロセス志向
 - (1-2) 指標（ソフトウェアメトリックス）を持つことの意義
 - (1-3) 日米の品質の差
 - (1-4) メトリックスの評価項目
 - (2) 開発のソフトウェアメトリックス
 - (2-1) システム開発の工期
 - (2-2) 要件定義後の工数・工期
 - (2-3) テストの標準工期
 - (2-4) 納入時の欠陥率・発注時の品質目標有無と換算欠陥率
 - (2-5) 発注時の品質目標
 - (2-6) 稼働後のバグ数とフォローフェーズの換算欠陥率
 - (2-7) 納入後に費やす負荷
 - (2-8) レビュー工数比率
 - (2-9) プロマネのスキルと工期遅延度
 - (2-10) 生産性
 - (2-10-1) 画面・帳票数からの工数の推定
 - (2-10-2) 画面・帳票数からの工数の推定 FP を用いる場合
 - (2-10-3) 画面数からの工数・費用の推定
 - (2-10-4) 品質と単価
 - (2-10-5) システムの重要度別の開発費用
 - (2-11) 顧客満足度
 - (2-11-1) 顧客満足度のコンセプト
 - (2-11-2) 要求仕様の明確さと顧客満足度
 - (2-11-3) 仕様変更と顧客満足度
 - (3) 保守のソフトウェアメトリックス
 - (3-1) 保守要員の守備範囲
 - (3-2) 保守作業の見積基準
 - (3-3) 保守作業の効率化
 - (3-4) 保守要員の業務内容
 - (3-5) 保守作業の品質
 - (3-6) システム保守費
 - (4) 運用費用のソフトウェアメトリックス
 - (4-1) システム運用評価指標
 - (4-2) 運用費用の変化
 - (4-3) 障害の件数
 - (4-4) 日本と米国のシステム品質の差
 - (4-5) 運用費の適正化取り組み状況
- 【付録】JUAS 各種調査報告書

◆本書の使い方

- ・ 本書は『JUAS ソフトウェアメトリックス調査報告書（2013 年）』ならびに『JUAS ユーザー企業IT 動向調査（2013）』から「使って便利、知っている」と得するノウハウを抽出したものです。
- ・ 本書では調査の背景や語句の定義、調査票などは割愛しています。不明な点はその都度、調査報告書に戻ってご確認ください。
- ・ 本書の図表番号は調査報告書と一致しています。毎年調査報告書の図表番号自体も2010年度から継承していますので、過去のデータを知りたい方は、同じ番号を参照して下さい。
- ・ 全ての指標は、システムの種類・規模によって異なります。特に生産性等は100人月以上の規模を想定しています。

以上

目 次

(1) はじめに	
(1-1) プロダクト志向とプロセス志向	3
(1-2) 指標（ソフトウェアメトリックス）を持つことの意義	3
(1-3) 日米の品質の差	4
(1-4) メトリックスの評価項目	4
(2) 開発のソフトウェアメトリックス	
(2-1) システム開発の工期	5
(2-2) 要件定義後の工数・工期	6
(2-3) テストの標準工期	6
(2-4) 納入時の欠陥率・発注時の品質目標有無と換算欠陥率	7
(2-5) 発注時の品質目標	8
(2-6) 稼働後のバグ数とフォローフェーズの換算欠陥率	8
(2-7) 納入後に費やす負荷	9
(2-8) レビュー工数比率	10
(2-9) プロマネのスキルと工期遅延度	10
(2-10) 生産性	11
(2-10-1) 画面・帳票数からの工数の推定	11
(2-10-2) 画面・帳票数からの工数の推定 FP を用いる場合	12
(2-10-3) 画面数からの工数・費用の推定	13
(2-10-4) 品質と単価	14
(2-10-5) システムの重要度別の開発費用	15
(2-11) 顧客満足度	15
(2-11-1) 顧客満足度のコンセプト	15
(2-11-2) 要求仕様の明確さと顧客満足度	16
(2-11-3) 仕様変更と顧客満足度	17
(3) 保守のソフトウェアメトリックス	18
(3-1) 保守要員の守備範囲	18
(3-2) 保守作業の見積基準	18
(3-3) 保守作業の効率化	19
(3-4) 保守要員の業務内容	20
(3-5) 保守作業の品質	20
(3-6) システム保守費	21
(4) 運用費用のソフトウェアメトリックス	23
(4-1) システム運用評価指標	23
(4-2) 運用費用の変化	24
(4-3) 障害の件数	24
(4-4) 日本と米国のシステム品質の差	25
(4-5) 運用費の適正化取り組み状況	25
【付録】JUAS 各種調査報告書	27

(1) はじめに

「ソフトウェアメトリックス」とはソフトウェアの品質 (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である。これを活用することにより、生産性の向上やプロジェクトの失敗を避けることができる。

(1-1) プロダクト志向とプロセス志向

現在、ハードウェアの品質管理技術は確立され欠陥率は非常に低い、ソフトウェアの欠陥(バグ)は「あって当たり前」と捉えられ、ハードに比べて大きく遅れをとっている。そのレベルを高めようと EASE (Empirical Approach to Software Engineering)の発想が登場した。今後の充実が望まれる。

図表 1-1-1 プロダクト志向とプロセス志向

	ハードウェア (他の産業、機械工業・建設業)	ソフトウェア (情報産業)
商品の保証	規格や標準に規定されている 規格違反は法律違反となる 規格の種類は国別に多数あり	商品の品質特性の規定は存在するが守るべき数値目標の規定はない
製造プロセスの規定	特に規定はない	開発フェーズ別になすべきActivityのガイドは存在する。
不良品	欠陥品個数は6シグマ以下(通念) 不良品は即時取り替えが原則	バグはあるのが当たり前 不良品がまかり通る世界から徐々に許されない世界へ
歴史	長年の歴史を持つ	数十年の歴史
今後	無欠陥商品の追究 不良品は刑事責任を問われる	EASE(Empirical Approach to Software Engineering)などが出現
考え方	プロダクト志向	現在: プロセス志向 プロダクト志向あつてのプロセス志向

(1-2) 指標 (ソフトウェアメトリックス) を持つことの意義

「見えないものは測れない」

「測れないものは改善できない」

「何をどのように測れば良いのか？」

「何をすれば良くなるのか？」

などの疑問を解くためには、次のいずれかの STEP をとることになる。

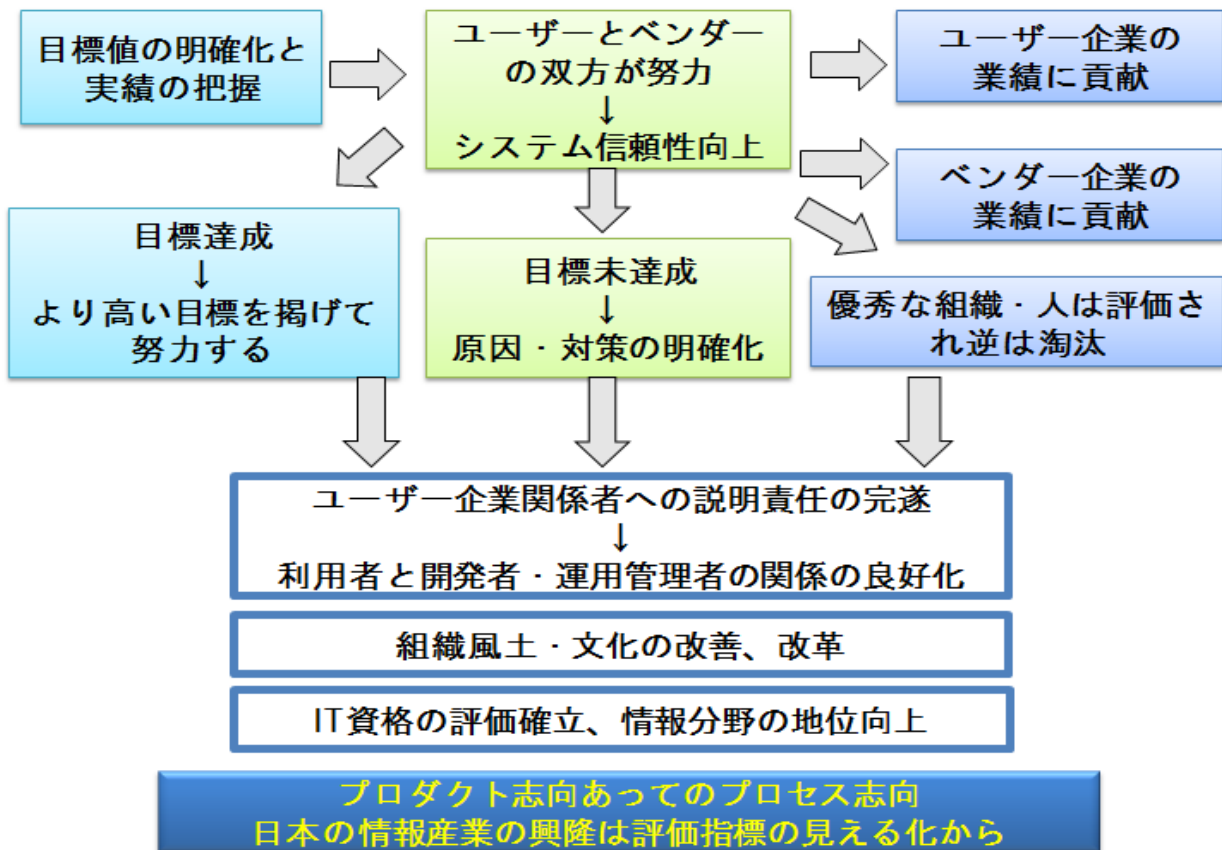
1 : 「事実の認識」 → 「データ収集」 → 「対策」 → 「改善」

2 : 「仮説の設定」 → 「データ収集」 → 「仮説の検証」 → 「対策」 → 「改善」

3 : 「理想状態の追究」 → 「改善」 → 「データ収集」

このソフトウェアメトリックスの活用により、多くのシステム利用者の不満の背景が顕在化するだけでなく、日本のソフトウェアの精度を客観的に示すことが出来る。

図表 1-2-1 ソフトウェア開発保守運用で指標を持つ意義



(1-3) 日米の品質の差 (参照 図表 9-11b)

システム開発完了後の1年間に発生した不具合数を日本と比較すると、米国は20倍、インドは13倍と大幅に悪い。情報システムが障害発生により停止する時間(月間)は米国が9倍近く多い。

(1-4) メトリックスの評価項目

JUASのソフトウェアメトリックスの評価項目は、品質(Quality)・費用(Cost)・工期(Delivery/Time)に加えて、ユーザー満足度(User Satisfaction)も重要な評価指標として採用している。

2：開発のソフトウェアメトリックス

(2-1) システム開発の工期（参照 図表 6-19）

$$\blacksquare \text{標準工期(月数)} = \text{投入人月の立方根} \times 2.5$$

(100 人月以上の場合)

例) 投入人月が、1000 人月の場合……1000 人月の立方根 \ast =10

$$10 \times 2.5 \text{ 倍} = 25$$

=> 標準工期 25 ヶ月

\ast a の立方根とは、 $b^3 = a$ を満たす b のこと

【使い方】

上記の式から得られた工期と実際に発注者から要求された工期を比較する。

- ・ 要求工期／標準工期が 1 以上の場合
→ 「工期は余裕がある。何故余裕があるのか、過去のプロジェクトを調べて特別な理由があるのか」を確認する。
- ・ 要求工期／標準工期が 1 以下の場合
→ 「工期不足の割合に応じて対策を取る。」
- ・ 0～25%短い場合・・・優秀なプロマネや SE を集める、ユーザー（顧客）の協力を仰ぐ等の対策が必要。
- ・ 30%以上短い場合・・・開発規模を見直す、工期を再設定し長くする等の対策が必要。

◆参考◆

- ・ この指標は、ベンダーの開発力と、発注者であるユーザー企業の協力体制によってばらつきがでる。
- ・ 過去、自社で「ノータラブルでリリースしたプロジェクト」、「大炎上を起こしたプロジェクト」について上記の指標との乖離（%）を見てみると良い。
使い慣れてくると、「自社の場合、-30%になると危険信号」などの目安ができる。
但し、短工期に慣れているユーザー企業の中には、この基準よりも短い工期で開発ができるケースもあるので、注意が必要である。

(2-2) 要件定義後の工数・工期 (参照 図表 6-162a)

■ 完成までの工数 = 要件定義の工数 × 10

例) 要件定義 (工数) が 10 人月だった場合……完成までの工数 (含: 要件定義) = 100 人月

【使い方】

要件定義に要した工数を 10 倍して全体工数を出し、実際の残工数と比較する。

- ・要件定義工数の 10 倍程度
→ 「予算としては妥当」
- ・要件定義工数の 10 倍を大きく下回る
→ 「(納期を順守すれば) 予算が不足するかも…」
「現時点で開発体制(ベテラン要員の比率)を見直すべき？」

◆ 参考 ◆

経済産業省のガイドでは要件定義フェーズを、原則として準委任契約で実施することを推奨している。多くの場合、感覚的に要件定義の 4-5 倍と答える人が多いが、実際にはプロジェクトの大きさには関わらず、ほぼ 10 倍である。

(2-3) テストの標準工期 (参照 図表 6-29)

■ テスト工期 = (設計工期 + 実装工期) の半分

例) 設計に 2 ヶ月 + 実装に 4 ヶ月だった場合…… $(2+4)/2 \Rightarrow$ テスト工期は 3 ヶ月

【使い方】

設計+実装工期に要した工期と結合テスト・総合テストの工期を比較する。

- ・テスト工期は、設計+実装工期の 50%程度
→ 「スケジュールとしては妥当」
- ・テスト工期は、設計+実装工期の 50%程度を大きく下回る
→ 「稼働準備不足で納期遅延するかも…」
→ 「テスト不十分で品質が担保できないかも…」

◆ 参考 ◆

「全体スケジュールが 20 ヶ月のプロジェクトがある。要件定義に 10 ヶ月かけたが、設計と実装に 8 ヶ月掛かるので、テストはなんとか残り 2 ヶ月で頑張っただけ」と無理を承知な予定を立てた。上記式では $8/2=4$ ヶ月テスト工期は必要である。結局このプロジェクトは稼働を 2 回延期し、テスト工期に 4 ヶ月を費やし完成した。

「最初から無理な工期設定をしなければ良かった」と反省した。

■ 換算欠陥率目標 = 換算欠陥数 / 開発工数 = 0.25 以下とする

■ 納入時のバグ(換算欠陥数) = 開発費 500 万円あたり 1 件まで

【欠陥数】

受入テストから、安定稼働(カットオーバー後 3 ヶ月)までに発見された欠陥数

【換算欠陥数】

換算欠陥数(重み付け欠陥数) = $2 \times$ 欠陥数(重度) + 欠陥数(中度) + $0.5 \times$ 欠陥数(軽度)

- 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)であり、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。

【換算欠陥率】

換算欠陥率(重み付け欠陥率) = 換算欠陥数 / 全体工数

【使い方】

システム開発発注時に、ユーザー(発注者)は「品質基準」を明示する。

・ 換算欠陥率(重み付け欠陥率) = 換算欠陥数 / 全体工数
→ 「換算欠陥率は 0.25 を達成するものとする」

・ 工数単価を 125 万円/人月(参照 図表 6-131)とすれば、欠陥数目標 = 発見欠陥数 / 500 万円以下になる。上記の値を上回る悪い品質の場合は、再テストに回すほうが良い。

◆参考◆

- ・ 発注時の品質目標の提示では、「納入時には『障害数 = 発注金額 / 500 万円以下』または『0.25 件 = 発生障害数 / 投入人月以下』にして欲しい」などと提示すれば良い。
- ・ 1 次開発 → 2 次開発など、さらに発注する場合にもこの目標を提示する習慣を持つと品質は安定する。
- ・ 調査では全体の 2/3 のプロジェクトが「換算欠陥率」 ≤ 0.25 以下を達成している。

(2-5) 発注時の品質目標 (参照 図表 6-59、60)

■ 品質目標をもって発注すれば、換算欠陥率は 1/2 に減少。

【使い方】

システム開発を発注する際は、前頁 2-4 のように品質基準を定めることが重要である。
品質基準を明示することで、換算欠陥率は約半数に減らすことができる。

換算欠陥率	品質基準			合計
	有り	無し	未回答	
件数	286	241	13	540
平均	0.29	0.65	0.97	0.47
割合	52.96%	44.63%	2.41%	100.00%
最大	10.12	12.73	6.36	12.73
最小	0.00	0.00	0.00	0.00

◆参考◆

現状、品質目標を提示しているプロジェクトは、おおよそ半分である。(参照図表 6-59)

(2-6) 稼働後のバグ数とフォローフェーズの換算欠陥率(参照 図表 6-126、6-128)

■ 稼働後の欠陥数 = 総合テストで出た欠陥数 × 27%(中央値)

■ フォローフェーズの欠陥数 = 全体工数 × 4%(中央値)

フォローフェーズ…システム稼働から3ヶ月後までの期間

例)総合テストで発覚した欠陥数が 100 の場合……稼働後に 25~30 個程度のバグが発生する。

【使い方】

ユーザー(発注者)の最終総合テストで発覚した欠陥数から稼働後の欠陥数を推定する。

- ・稼働後の欠陥数が多く発生することが予想される場合
 - 「カットオーバーを延期し、総合テストを再実施する」
 - 「本番稼働後に発生すると想定される欠陥数に合わせて、プログラム修正ができるプログラマを動員する」

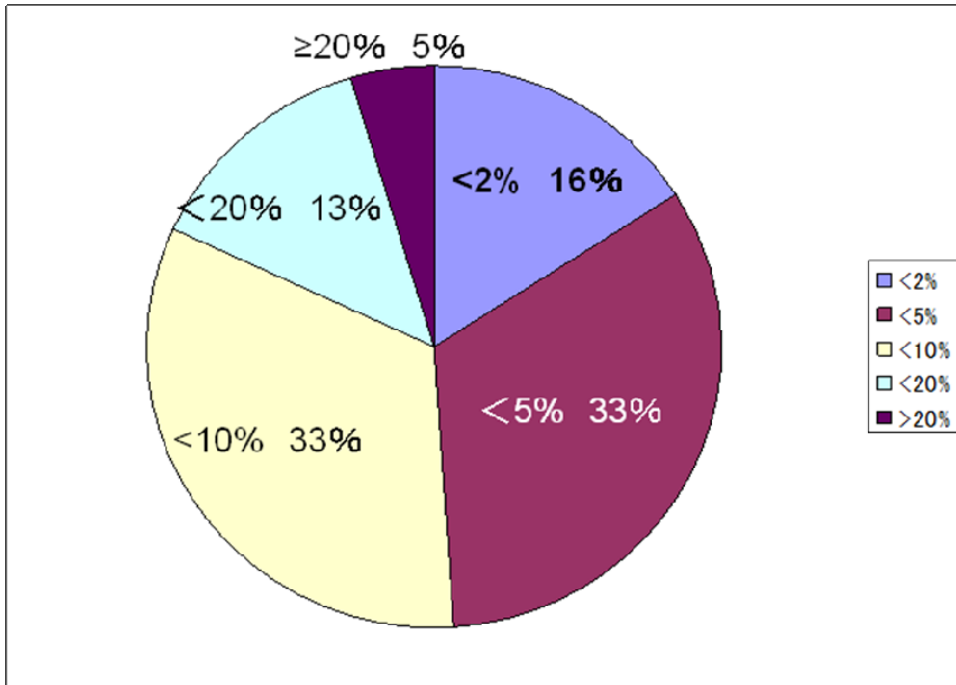
◆参考◆

- ・調査では、ユーザー(発注者)側の総合テスト時に出た欠陥数と稼働後に発生した障害率の割合は 27%(中央値)と算出された。簡単に「総合テストで見つかった欠陥数の 1/4 の数が稼働後に発生する」とすれば覚えやすい。
- ・この実績値は企業の基幹業務システムの場合であり、「重要インフラシステム」は欠陥ゼロを目指すこと。

(2-7) 納入後に費やす負荷 (参照 図表なし)

※下記図表は『JUAS ソフトウェアメトリクス調査報告書 2013』には掲載されていません。

■ 16%のプロジェクトが「フォロー工数／全体工数＝2%」を達成。



【使い方】

- ・ 稼働後、品質が悪いとフォロー工数が増加する。
- ・ 稼働後に費やす工数をプロジェクト管理の目標に設けること。

◆参考◆

- ・ 本調査を詳細に分析すると、およそ半数のプロジェクトは「フォロー工数／全体工数の割合は5%未満」と回答しており、残り半数が「5%以上（品質が悪い）」と回答している。
- ・ 一方で、「2%未満」と回答しているプロジェクトは16%存在している。この2%以下を目標にされたい。

(2-8) レビュー工数比率 (参照 図表 6-112)

■ユーザー(発注者)が参加するレビュー時間は
“開発者のドキュメント作成時間の 20%以上”が必要。

例)開発者のドキュメント作成時間が 50 時間の場合…ユーザー(発注者)レビューは 10 時間以上が必要。

【使い方】

開発者のドキュメント作成時間から、ユーザーの参加すべきレビュー時間を算出する。

- 「レビューに必要な時間を確保するよう、事前にユーザー(発注者)に依頼しておく。」
- 「レビューに必要な時間の確保が難しい場合はユーザー(発注者)側で業務経験者を別途アサインし、緻密なレビューを行うための予算を積んでおく。」

◆参考◆

- ・上記の指標が 20%以上の発注者の場合、悪い品質のプロジェクトは存在していない。
- ・ベンダー任せでユーザーレビューを十分確保しないプロジェクトではバグ発生率は高くなることをユーザーに伝え、協力を依頼すること。

(2-9) プロマネのスキルと工期遅延度 (参照 図表 6-101、102 図表 6-50 図表 6-53)

■①高スキルの PM が短工期プロジェクトを成功に導く

■②長工期プロジェクトの方が品質は悪くなる

■③工期遅延のプロジェクトは換算欠陥率も高くなる。

【使い方】

- ① 短工期プロジェクトには、経験のあるベンダーのプロジェクトマネージャーを配置することを契約条件に入れて、開発を成功させる。
- ② ベンダーの、経験のあるプロジェクトマネージャーが短工期プロジェクトに配置されて良い品質のプロジェクトを作り出す。工期が十分にあると知って、目標値を持たずに悠長に仕事をしていると品質が悪いプロジェクトになる。経験豊かなベンダープロジェクトマネージャーを配置できない場合は PMO などの品質管理組織を使ってカバーすること。
- ③ 工期遅延を起こすプロジェクトほど換算欠陥率は悪くなる。計画工期を守れないプロジェクトは工期遅延を収めるために応急処置に走るために品質が低下しやすい事を意識してプロジェクト管理すること。

(2-10) 生産性

(2-10-1) 画面・帳票数からの工数の推定 (参照 図表 6-192a 6-132)

$$\blacksquare \text{全体工数} = 112.97 + 0.81 \times \text{画面数} + 0.42 \times \text{帳票数}$$

(新規開発及び再開発・改修の場合)

例)画面数 200、帳票数 50 のプロジェクトの場合…

$$\textcircled{1} 112.97 + 0.81 \times 200 + 0.42 \times 50 = 295.97 \text{ 人月}$$

$$\textcircled{2} 295.97 \times 110.83 \text{ 万円} \text{ (注1)} = 32,802 \text{ 万円}$$

注 1…図表 6-132 を参照

【使い方】

画面数・帳票数から、全体工数を推定する。

- ・算出された工数が、想定している工数を大きく上回る場合…
→ 「そんなに人月はかからないだろう」「見積もりは適切か(再吟味する)」
- ・算出された工数が、想定している指標を大きく下回る場合…
→ 「実現可能か」「この見積りで開発に入るなら、相当な覚悟と準備が必要」

◆参考◆

- ・上記の指標で出た工数に、単価を掛ければ全体費用を概ね推定することができる。
- ・仮に工数単価が不明な場合は、調査報告書 図表 6-132 の金額を代入すれば、おおよその金額を推定することができる。このデータによると 100～500 人月規模の単価は 110.83 万円/人月である。
上記の画面数 200、帳票数 50 のプロジェクトを例にとれば、295.97 人月×110.83 万円 = 32,802 万円となる。

$$\blacksquare \text{FP値} = 91.54 + 13.41 \times \text{画面数} + 40.33 \times \text{帳票数}$$

(新規開発+再開発WF型の場合)

例) 画面数 200, 帳票数 50 のプロジェクトの場合…

$$\begin{aligned} \textcircled{1} & 91.54 + 13.41 \times 200 \text{ 画面数} + 40.33 \times 50 \text{ 帳票数} && = 4790 \text{ FP} \\ \textcircled{2} & 4790 \text{ FP} \times 7.17 \text{ (万円/FP)} \text{ (注1)} && = 34,344 \text{ 万円} \end{aligned}$$

注1…図表 6-159 を参照

【使い方】

画面数・帳票類から FP を推定する。

- ・算出された工数が、想定している工数を大きく上回る場合…
→ 「そんなに人月はかからないだろう」「見積もりは適切？」
- ・算出された工数が、想定している指標を大きく下回る場合…
→ 「実現可能？」「やるなら、事前に相当な覚悟と準備が必要」

◆参考◆

FP 値あたりの工数単価が不明な場合は、ソフトウェアメトリクス調査報告書 2013(図表 6-159)の金額を代入すれば、おおよその金額を推定することができる。

■全体費用＝画面数×画面あたりの工数

図表 6-198 工数区分別画面数(ウォーターフォール型開発のみ)

プロジェクト規模	件数	システム当たりの画面数	画面当たりの工数(加重平均)
<10人月	22	18.82	0.40
<50人月	117	37.47	0.72
<100人月	48	60.63	1.18
<500人月	101	135.19	1.58
>=500人月	32	262.19	4.46
合計	320	92.98	2.21

システム規模別の画面数と工数の関係は上の表で示される。

帳票数など必要機能分を全て画面数で代表している。

例) 画面数 200 のプロジェクトの場合…

$$\textcircled{1} 1.58 \text{ (参照: 100~500 人月の画面あたりの工数)} \times 200 = 316 \text{ 工数}$$

$$\textcircled{2} 316 \text{ 工数} \times 110.83 \text{ 万円 (注1)} = 35,022 \text{ 万円}$$

注1…図表 6-132 を参照

【使い方】

要件定義フェーズで、画面数しか分からない場合の推定に使う。

◆ 参考◆

オンラインシステムの場合、図表 6-13、6-15(中央値)を参考にすると、帳票類は 1/3~1/4 ついて
いる。帳票類が多い場合は、この図表を参考にされたい。

■『品質の良いシステム開発は価格が高い』とは言えない

【使い方】

ベンダーから納入された開発データ(システム規模、品質データ)を元に、横軸に平均単価、縦軸に換算欠陥数をプロットし評価する。

ベンダーを品質と価格によって区分して見ると次の4種類に分けられる

(a)価格は高いけど品質も良い (b)価格は高いが品質は悪い

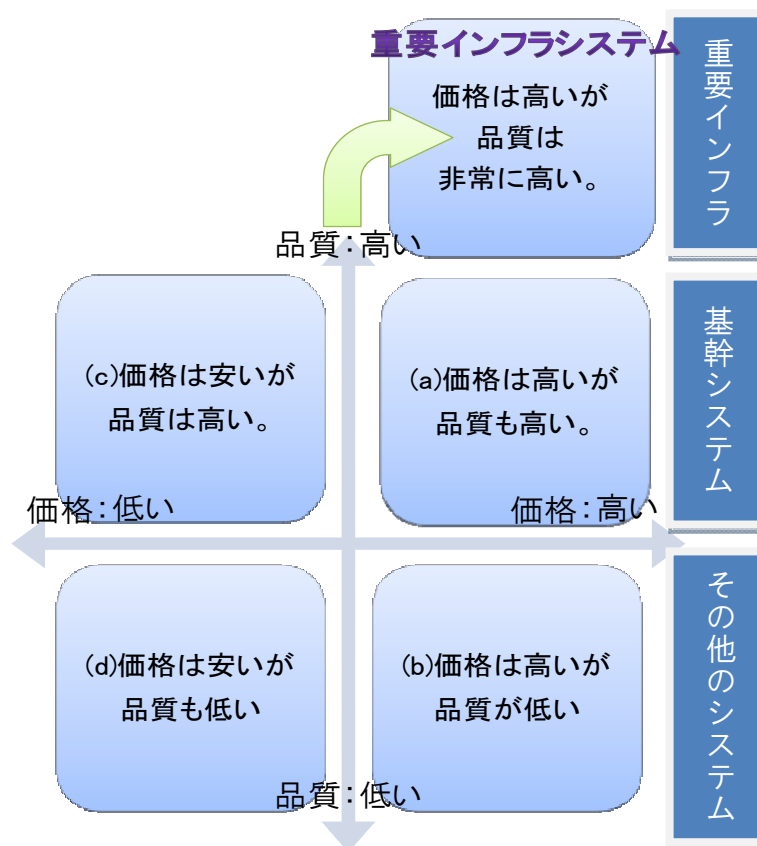
(c)価格は安いけど品質は良い (d)価格は安いけど品質も悪い

→ (b)(d)のベンダーには注意を促す。

→ (a)(c)のベンダーは評価する。

◆参考◆

- ・『品質の良いシステム開発＝価格が高い』は成立しない」は様々なデータを集約した結果であるが、中には「価格の高いベンダーの品質は良い」結果が得られている企業もある。
開発プロジェクトが終了した後に品質結果をふりかえり評価をすることが、重要である。
- ・発注時に品質目標を記述することにより、(b)(d)プロジェクトになることを避けることができる。



※上記図表は『JUAS ソフトウェアメトリックス調査報告書 2013』には掲載されていません。

(2-10-5) システムの重要度別の開発費用 (参照 図表 6-166、6-167)

■開発費用＝生産物量×生産性×単価

■重要インフラ等システム＝基幹システム×1.8×1.5 倍

【使い方】

高品質を特に期待する重要インフラプロジェクトは、費用が 2.7 倍かかるとして予算を準備する。

◆参考◆

- ・FP 生産性比較をすると、重要インフラ:13.58 FP/人月、企業基幹システム:20.54 FP/人月となり、企業基幹システムでは FP 生産性は重要インフラの 1.5 倍になる。
- ・単価アップと生産性アップで $1.8 \times 1.5 = 2.7$ 倍となる。

(2-11) 顧客満足度

(2-11-1) 顧客満足度のコンセプト (参照 図表 6-108)

■換算欠陥率が 0.25 より悪いプロジェクトは評価されない

■換算欠陥率がゼロでも顧客満足度は 100 点をとれない。

【使い方】

- ・顧客満足度は成果品質(コスト、工期、品質)とプロセス品質(好印象、正確性、迅速性、柔軟性、共感性、安心感)の両方で評価される。
- ・ステークホルダーの特徴に合わせたプロセス品質を確保すること。

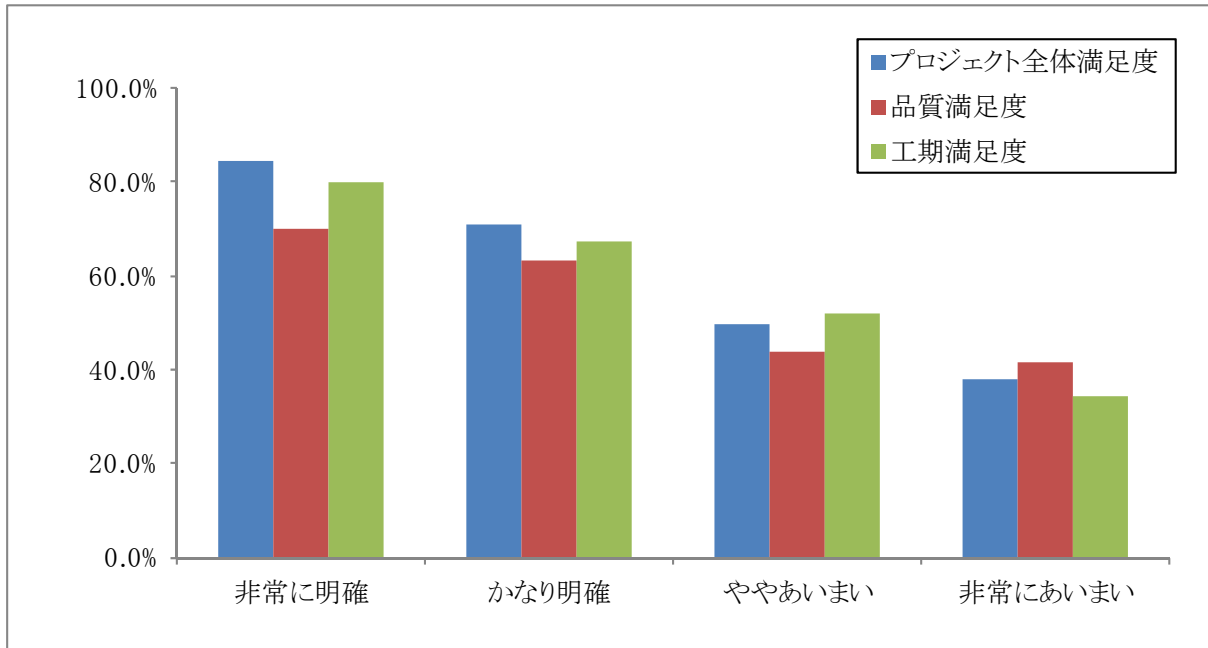
◆参考◆

- ・本調査では、71%のプロジェクトが『換算欠陥率 0.25 以下』を達成している。

■ 仕様の明確さを欠くほど顧客満足度が確保できない。

仕様の明確度と満足度の関係は「ややあいまい」「非常にあいまい」になるにつれて「やや不満」「不満」の割合が増加してくる。

図表 6-215a 仕様明確度別の満足の割合



【使い方】

- ・高いユーザー満足度を得るためには要件定義、基本設計段階で仕様を確実につめること
- ・仕様を確実に詰められずアジャイル・高速開発等をする場合は、その特徴を十分にユーザー（発注者）に説明し、協力をあおぐこと

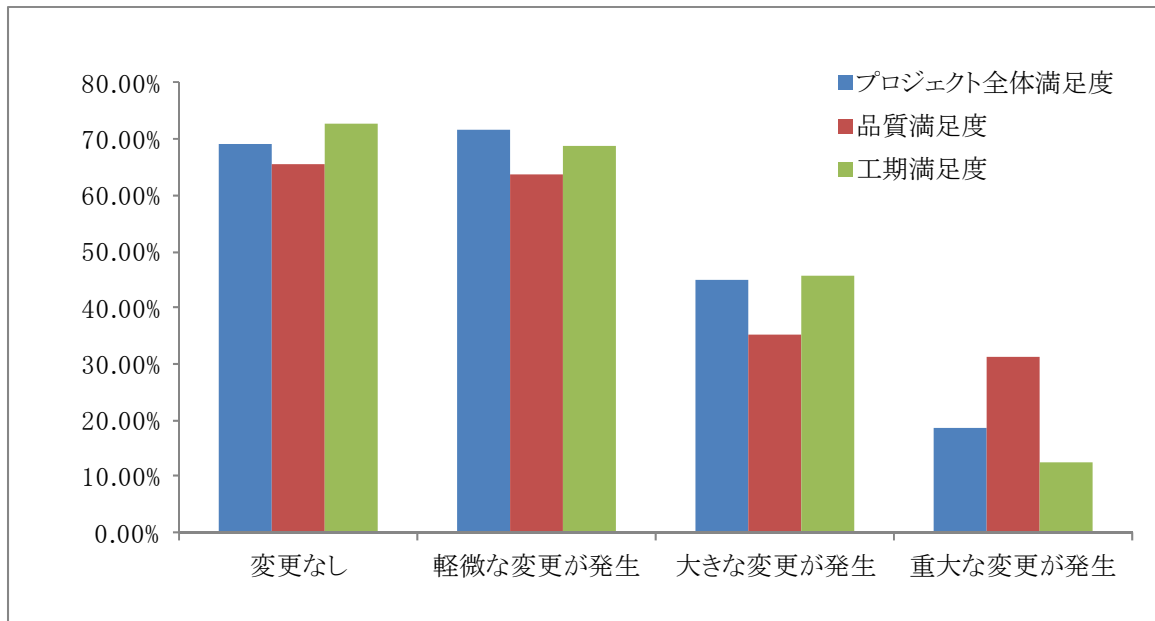
◆ 参考 ◆

品質、工期への影響については、品質は図表 6-105、工期は図表 6-219 を参照のこと

■大きな仕様変更が発生すると、満足度が低下する。

全体、品質、工期満足度と仕様変更の発生程度との関係は「大きな仕様変更が発生」「重大な仕様変更発生」と進むほど満足度は低下する。

図表 6-220a 要求仕様の変更発生とプロジェクトの満足度(複数回答)



【使い方】

- ・開発途中の大きな、あるいは重大な仕様変更は、多くの手戻り作業が発生するので、ユーザー(発注者)にレビューの参加を促し、仕様確認を十分にすること
- ・やむを得ず修正をする場合は、工期と費用を十分に見込むこと

◆参考◆

仕様変更と工期遅延度との関係は図表 6-217 を参照、品質との関係は図表 6-223 を参照のこと

3：保守のソフトウェアメトリックス

保守作業はシステムを使い続ける限り永久に続く作業であるにも関わらず、ここに焦点が当たることは少ない。保守作業に目標値を持ち込むことはシステムの安定稼働やユーザーの利用支援に大きく貢献する上、費用の面でも削減効果は大きい。

(3-1) 保守要員の守備範囲（参照 図表 7-15a）

- 保守専任者のカバー範囲…36.5 万 LOC/人(平均)
- 保守専任者のカバー範囲…16.8 万 LOC/人(中央値)
- 保守専任+非専任者のカバー範囲…29.8 万 LOC/人(平均)
- 保守専任+非専任者のカバー範囲…12.6 万 LOC 人(中央値)

【使い方】

- ・現在開発中のプロジェクトの規模から、必要保守要員を算出する目安として活用する。
- ・自社の開発プロジェクトは完成後、「保守フェーズ要員は何人必要か、開発チームの中から誰を保守者として残すか」を配慮して、育成しておかねばならない。

◆参考◆

保守専任者の作業カバー範囲は平均 22%、中央値で 33%。非専任者を含めた場合よりも広い。

(3-2) 保守作業の見積基準（参照 図表 7-56、7-57）

- 保守作業の工数見積基準は 6 種類ある。

【使い方】具体的には、以下の 6 手法を組み合わせる。

1. 修正内容の箇所数、負担程度を加算・見積
2. 修正の影響範囲まで配慮して予測・見積
3. リスク要因まで配慮して見積もる方法
4. WBSを準備して見積もる方法
5. 担当者の熟練度を考慮する方法
6. 改修母体の品質を配慮する方法

◆参考◆

保守作業の工数見積基準があるプロジェクトは半数、保守作業を行う担当者が業務見積を行う場合も半数、が実態である。

■「設計ドキュメントのレベル向上」と「テスト環境整備」は 62%以上のプロジェクトが実施済み。

図表 7- 63 保守負荷を低減する主なしくみの分布 (複数回答) (単位: 件, %)

保守負荷を低減する	2012 度		全体	
	件数 (件)	割合 (%)	件数 (件)	割合 (%)
1.保守用調査ツール	9	16.7%	72	23.2%
2.設計ドキュメント	41	75.9%	201	64.8%
3.テスト環境整備	33	61.1%	193	62.3%
4.ドキュメント解析容易性	5	9.3%	48	15.5%
5.移植環境適合性	4	7.4%	26	8.4%
6.開発時のバグ徹底	5	9.3%	28	9.0%
7.複数案件の要件等、同時着手	23	42.6%	46	14.8%
8.その他	3	5.6%	16	5.2%
合 計	123	—	630	—

【使い方】

アンケートには含まれていないが、リポジトリの活用が保守作業の効率化に大きな効果があるので検討すること。

◆参考◆

時間的・予算的制約から開発時に実施ができなかった保守が容易な仕組みの整備を保守計画に盛り込み実行するのが望ましい。

(3-4) 保守要員の業務内容 (参照 図表 7-45)

- アプリケーションプログラムの保守作業のうち、31%が「利用者から来る問い合わせ」
- 本来的な是正保守(欠陥の修正作業)への対応は 17%

【使い方】

アプリケーションプログラムの保守作業のうち、利用者の問い合わせ対応に 1/3 の負荷がかかっている
ので、利用者教育を十分にす、マニュアル等を整備するなどの対策をとること。
是正保守(バグの修正作業)が 17%もあるので、開発時の欠陥を減少させること。

◆参考◆

是正保守と問い合わせ作業を合わせると半分を占める。
それ以外は、計画的に保守作業ができるので、十分な計画を作成して保守作業を推進すること。

(3-5) 保守作業の品質 (参照 図表 7-51)

- 受入確認即時合格率*は平均で 50%(中央値 80%)
- 本番組込後の保守欠陥率は平均値で 10%(中央値で 3%)

※受入確認即時合格率…保守作業完了後検査に回り、それが合格となった比率

【使い方】

保守作業品質の目標値として、二つの指標「受入確認即時合格率」「本番組込後の保守欠陥率」を設定し努力すること。

◆参考◆

稼働直後の初年度の保守品質は
・開発時のバグが潜在していること
・保守作業に慣れていないこともあって、上記指標の半分の悪い値である。
2 年目以降は安定した品質実績になっている

■ 自社開発システムの保守費用(初年度～5 年間)

$$= \text{初期開発費} \times 2.16$$

■ パッケージを利用したシステムの保守費用(初年度～5 年間)

$$= \text{アドオン開発初期費} \times 3.01 + \text{パッケージの保守費/年} \times 1.59$$

■ 5 年目以降に ERP パッケージの更新費用を請求される場合は、自社開発よりパッケージ利用の方が高くなる場合が多い。

例)

・ 自社開発プロジェクトの保守費用 = 1 億円 \times 2.16 = 216,000,000 円

→ このプロジェクトの開発～5 年間の保守費用概算見積もりは 2 億 1600 万円

・ パッケージ利用でアドオン初期費用が 3000 万円、パッケージ本体の保守費が 1000 万円/年

$$(3,000 \text{ 万円} \times 3.01) + (1,000 \text{ 万円} \times 1.59) = 10,620 \text{ 万円}$$

→ このプロジェクトの開発～5 年間の保守費用概算見積もりは 1 億 620 万円

【使い方】

・ 上記の指標を用いて、5 年間の保守費用の概算を算出してみる。

・ 開発までのリソース、その後の利用年数なども勘案し、自社開発かパッケージ利用かを決断すると良い。

◆参考◆

保守費用分析 (平均値を採用)	自社開発 A				パッケージ本体費用 B			
	保守費用(件数)		開発費用(件数)		アドオン開発費用 C			
	A1		A2		本体保守(件数)		開発保守(件数)	
初年度総保守費用	8.5%	265	19.7%	310	12.9%	87	50.8%	58
2年目総保守費用	8.5%	218	15.5%	253	10.6%	67	37.5%	48
3年目総保守費用	8.8%	173	14.4%	198	10.4%	52	37.4%	41
4年目総保守費用	8.4%	138	10.5%	154	13.5%	46	33.6%	37
5年目総保守費用	9.3%	114	11.8%	130	11.6%	38	41.7%	28
年間平均	8.7%	—	14.4%	—	11.8%	—	40.2%	—
初期開発費用	A				B		C	
合計費用比較	$A + A \times 0.231 \times 5 = 2.155 \times A$				$1.590 \times B$		$3.010 \times C$	

保守費用にはさまざまな費用がかかっている。

A 自社開発の場合には保守チームの費用(A1)に加えて、一期開発で実現できなかった機能を開発チームが追加開発をしている分(A2)の二つの費用がかかってくる。

B パッケージを使う場合はパッケージ本体の保守費用(B)とパッケージにない機能を追加開発するアドオン部分(C)の二つの費用がかかってくる。

各々のプロジェクトの A、B、C の金額によって 5 年間の合計費用は変わってくる。

5 年間の総費用は $2.155(1.000 + 0.231 \times 5 \text{年間})A$ と $(1.590B + 3.010C)$ の比較で損得が決まる。B、C の係数も A の係数と同様に保守費用の年平均割合に 5 年を掛けて算出している。B、C の係数は、それぞれ $B(1.000 + 0.118 \times 5 \text{年間} = 1.000 + 0.590)$ 、 $C(1.000 + 0.402 \times 5 \text{年間} = 1.000 + 2.010 = 3.010)$ によって算出される。なお、A、B、C は稼働までの費用である。

※ 2012 年度の費用(平均値)の実績平均値は、 $A=64,168$ (万円)、 $B=35,980$ (万円)、 $C=26,741$ (万円)であった。

4：運用費用のソフトウェアメトリックス

(4-1) システム運用評価指標（参照 図表 9-11）

■システムの運用段階の評価指標は、「保守運用費用」「稼働」「稼働品質」「顧客満足」「投資評価」の5種類

図表 4-1-1 システム運用評価指標（参照 図表 9-11）

大区分	評価項目	評価式	評価	参考(システムのタイプ別の目標)
保守運用費用	保守運用費用率	保守運用費用/IT費用全体	対他社比較	なすべきことを実施した上でならば安い方が良い。投資不足の場合もあるので注意が必要
		保守運用費用/売上高		
稼働	稼働率	実績稼働時間/計画稼働時間	1に近いほど良い	99.999%(5分停止/年)以上
	延べ稼働率	延べ時間-計画停止時間-障害停止時間/延べ時間		顧客の使用可能時間の評価(99.95%以下になると不満がでるなど)
稼働品質	業務停止回数	業務停止回数/年	0に近いほど良い	基幹業務システムは0.06件/年・運用費の標準値あり
	規定時間外停止回数	規定時間以上停止した回数/年	0に近いほど良い	15分以上停止した回数/年 復元対策が重要になる
	オンライン平均応答時間	規定内応答回数/全応答回数	1に近いほど良い	例:300件/分の入力で2秒以内の応答率が95%など
顧客満足	お客様迷惑度指数	お客様に迷惑をかけた回数×重要度/年間	0に近いほど良い	お客様に迷惑をかけた回数点/運用費などで他社比較
	ユーザー満足度	品質、価格、納期、振る舞い、投資効果で評価する	別途	別途
投資評価	投資・費用	IT投資金額/売上高 IT投資金額/ユーザー当り	戦略による	類似企業との比較
	効果	ROI、KPI、ユーザー満足度	1以上が理想	計画値との比較、プロセスの評価も加えた評価体系の確立

【使い方】

システムの企画時点から、上記の(図表 9-11)を参考に評価指標の体系化と目標値を設定する。

- 保守運用費*は IT 費用全体の 53%にまで低下
 - クラウドの利用などを含めると更なる低下も予想される。
- *保守運用費…開発費以外の費用を指す。

【使い方】

自社の長期的保守運用費用の低下状況を確認し、他社と比較するために使用する。

◆参考◆

情報システム全体の費用はハードの低廉化・仮想化技術の活用などにより、10 年間で半分に低下した。この事実を踏まえた自社の保守運用費用の目標値を設定し努力すること。

- 保守運用費1億円当たり 0.06 件
= 保守運用費17 億円あたり 1 件の重大なシステム障害*が発生
- *重大なシステム障害…担当役員に報告される規模のシステム障害

例)システムの保守運用費が 50 億円の場合

$0.06(1 \text{ 億円当たりの障害件数}) \times 50(\text{億}) = 3 \Rightarrow \text{約 3 件/年の重大な障害が想定される。}$

【使い方】

重大な障害件数 ÷ 自社の運用費(億に換算)と、0.06 との比較をする。

・指標を大きく上回る場合…

→ 「システムトラブルが多すぎる。」「根本的な問題が潜んでいないか」

「発生した障害を早期に解決させる対策がなされているか」などを考慮する。

・指標を大きく下回る場合…

→ 「品質が担保されている」「品質を担保しコスト削減を目指せないか」を検討する。

◆参考◆

JUAS 企業 IT 動向調査と JUAS ソフトウェアメトリックス調査の二つの調査で、障害度数と運用費用の関係は、同じ結果が得られている。

(4-4) 日本と米国のシステム品質の差 (参照 図表 9-11b)

■ 日米の運用障害発生割合は 9 倍日本の方が少ない。

【使い方】

ソフトウェアの品質はハードウェアの品質並みに良くしてほしいとの日本人の気質が表れている。
費用をかけないでこの品質以上のシステムを作り、運用するための一つの目標として考えてほしい。

◆参考◆

日米の運用障害発生割合は日本が 1 に対して、米国は 9 である。
開発のみならず保守運用含めて厳しい管理をしていることになる。
障害頻度の高いシステムに慣れれば、障害発生時もそれなりに対応する方法を見つけて処置することも可能だろうが、現在の日本のユーザーでは許容できない可能性がある。

(4-5) 運用費の適正化取り組み状況 (参考 図表 8-23 a b, 8-23 c)

【使い方】

・27 項目の評価項目をつけることによって自社の運用費用削減の要努力項目が判断できる。
他社比較で自社の方が遅れていると思われる場合は、それで良いのか議論して対策をとること。

図表 4-4-1 運用費削減への努力評価表 (図表 8-23 a, b, 8-23 c)

回答区分: 5.実施中 4.一部実施中 3.検討中 2.検討したが実施せず 1.未検討 0. その他

区分	施策	対策事項		回答区分			
				平均	中央値	標準偏差	回答数
1	システム 資産 棚卸	不要 HW・SW 排除		4.5	5.0	1.0	61
2		過剰リソース契約解除		3.6	4.0	1.5	60
3		保守契約棚卸		4.5	5.0	1.0	61
4	ベンダー 契約 見直し 改善	集中購買	PC	4.8	5.0	0.8	62
5			サーバー	3.8	5.0	1.6	60
6			ネットワーク機器	3.8	5.0	1.6	59
7			OS 等	4.2	5.0	1.3	60
8		特定ベンダー集中購入		3.7	4.0	1.5	60
9		競争入札		4.2	5.0	1.2	61

10		過剰保守契約見直し		4.1	5.0	1.3	59
11		保守コスト低減		3.6	4.0	1.6	57
12	運用・運行 プロセス 改善	効果的な外 部能力活用	外部委託	4.2	5.0	1.2	61
13			無人化	2.9	3.0	1.6	58
14		運用プロセス改善		3.7	4.0	1.4	61
15		コールセンター低廉化		2.4	2.0	1.5	59
16		仮想技術 活用	サーバー		4.4	5.0	0.8
17	ストレージ		4.1	5.0	1.2	61	
18	PC		3.1	3.0	1.4	60	
19	システム 再構築	運用簡素化		3.3	4.0	1.5	57
20	クラウド 活用	インフラ		3.5	4.0	1.5	58
21		基幹業務		2.7	3.0	1.5	58
22		メール等		3.4	4.0	1.3	58
23	省エネ 省資源 (BPRを 伴う)	機器統合・ 削減等	多機能プリンター導入等	4.2	5.0	1.4	59
24			個人プリンター廃止	4.4	5.0	1.2	58
25		紙帳票出力 廃止・削減 (社内)	電子帳票化等	4.3	4.0	0.9	58
26		紙帳票出力 廃止・削減 (社外)	帳票類の Web 配信等	3.9	4.0	1.2	58
27		省電力徹底	省電力機器		4.4	5.0	1.2

ソフトウェアメトリクス調査 2013

～生産性向上に役立つメトリクス分析

ソフトウェアの開発保守運用に役立つ評価指標を見つけ出し、ユーザーとベンダーが、あるいはベンダーの発注者と受注者が相互にコミュニケーションをとる場合の基準を作りたいと志してから9年目の報告書にあたります。工期、品質、生産性（価格）、顧客満足度などの目標値設定、基準の活用法など、評価基準はユニークで使いやすいと評価をいただき、日本はもとより、近年では海外からも注目されています。自社のIT活動を見直すために、自社で取り組み可能な指標ひとつからご活用ください。

【ご購入はamazonから】

2013年7月発売



¥5714+税

企業IT動向調査報告書 2013

～ユーザー企業のIT投資・活用の最新動向（12年度調査）

株高や円安が急速に進んだ2013年は、企業が「守り」から「攻め」に転じ、イノベーションを加速させる年になるでしょう。経営環境の変化に対応しながら、競争力を維持・向上するには、企業内外の経営資源を最大限に活用し、経営革新に取り組むことが不可欠です。そのために、企業はどのようなIT戦略を実行していけばよいのでしょうか。——東証1部上場企業とそれに準じる企業の計4000社を対象に実施（有効回答は1030社）した国内最大規模*の調査結果から、2013年度のIT投資・活用の最新トレンドを解説します。

*企業の情報システム部門長を対象に実施するIT投資動向の調査として

★CD-ROM付き（報告書の全文をPDFで収録）

著者・編者：JIAS 発行：日経BP社 発売：日経BPマーケティング

【ご購入はこちら】<http://www.juas.or.jp/servey/it13/book.html>

2013年7月発売



¥14,200+税

情報システム管理の神髄

～企画・開発・保守・運用の改善のヒント

開発、保守、運用まで一貫して「各フェーズでの仕事の仕方のHow（どのように）」を解説。一歩進んだプロジェクト管理方法はないのかを求めたWG活動の知恵を凝縮。稼動最初から100%の完成を要求される再構築プロジェクトなど、より繊細緻密なプロジェクト管理が求められる現代にふさわしい、プロジェクト管理の注意事項満載。プロジェクト管理の開発保守運用にご活用ください。

【ご購入は amazon から】

2013年1月発売



¥3,429+税

5W4Hで解くプロジェクト管理

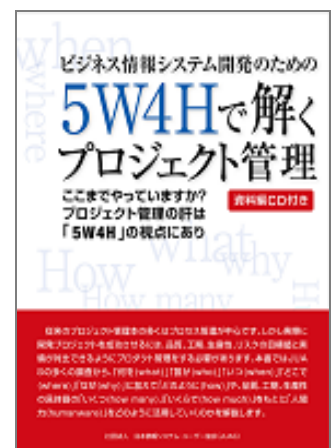
～プロジェクト管理の実践的解説

従来のプロジェクト管理本の多くはプロセス管理が中心です。しかし実際に開発プロジェクトを成功させるには、品質、工期、生産性、リスクの目標値と実績が対比できるようにプロダクト管理をする必要があります。本書ではJUASの多くの調査から、「何を(what)」「誰が(who)」「いつ(when)」「どこで(where)」「なぜ(why)」に加えて「どのように(how)」や、品質、工期、生産性の具体値の「いくつ(how many)」「いくらで(how much)」をもとに「人間力(humanware)」をどのように活用していくのかを解説します。

★サンプルドキュメントがダウンロードできます。

【ご購入は amazon から】

2011年1月発売



¥5,524+税

「ユーザー企業 ソフトウェアメトリックス 要点ハンドブック」

発行日：2014年2月 第1版

2014年3月 第2版

発行：一般社団法人 日本情報システム・ユーザー協会

〒103-0012 東京都中央区日本橋堀留町1-10-11 井門堀留ビル4階

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

本ハンドブックは『JUAS ソフトウェアメトリックス調査2013』、『JUAS ユーザー企業IT動向調査2013』より、指標・要点をまとめたものです。

調査の背景や調査票、分析に関するご不明点は、『JUAS ソフトウェアメトリックス調査2013』、『JUAS ユーザー企業IT動向調査2013』をご確認ください。

(禁無断転載)

一般社団法人 日本情報システム・ユーザー協会(JUAS)
定価 (本体価格 186 円+税金)

