

第14回インターネット時代のCOBOL活用セミナー
 ～基幹システムの‘いま’と‘これから’を支え続けるCOBOLの力～

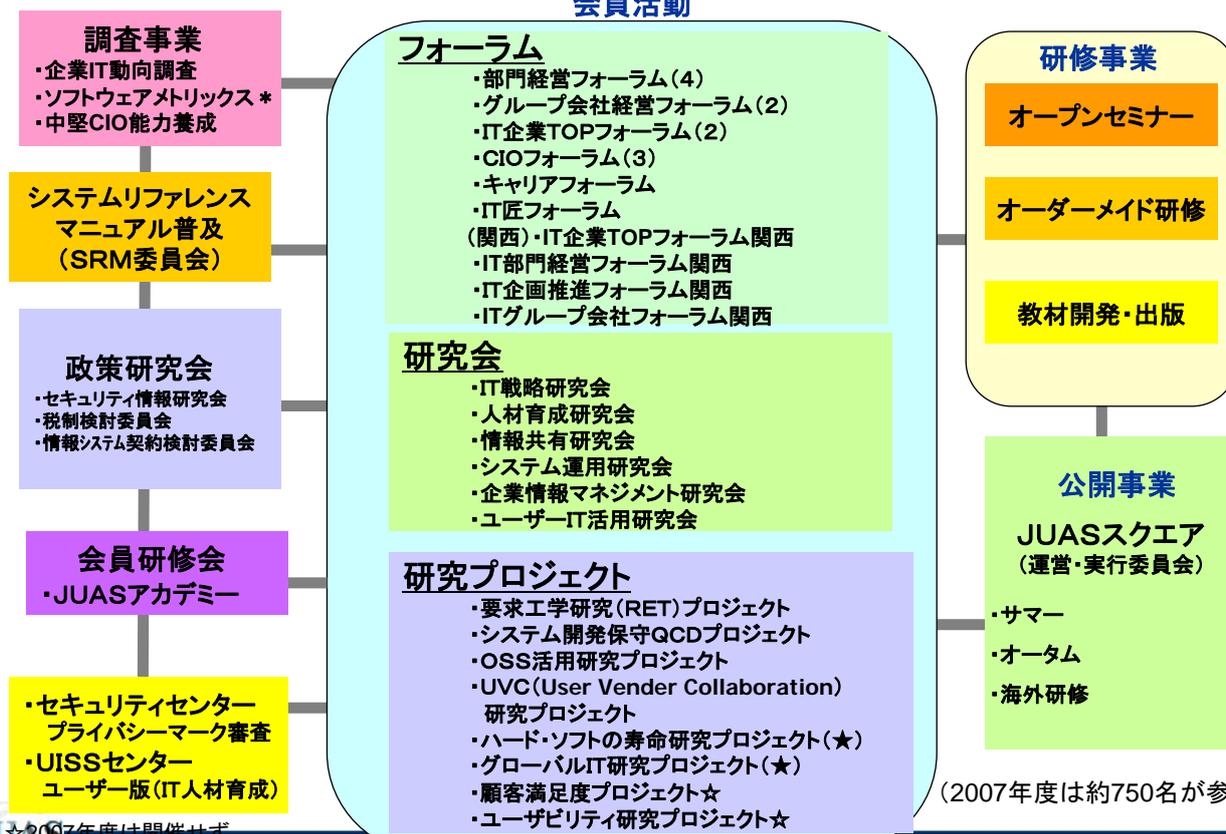
情報システム開発・保守、見える化への挑戦 ユーザー側から見たソフトウェアメトリクス

2007年12月13日

(社)日本情報システム・ユーザー協会
 専務理事 細川 泰秀

JUAS 2007年度活動関係図 (ユーザーの要求が未来を切り開く)

会員活動



(2007年度は約750名が参加)

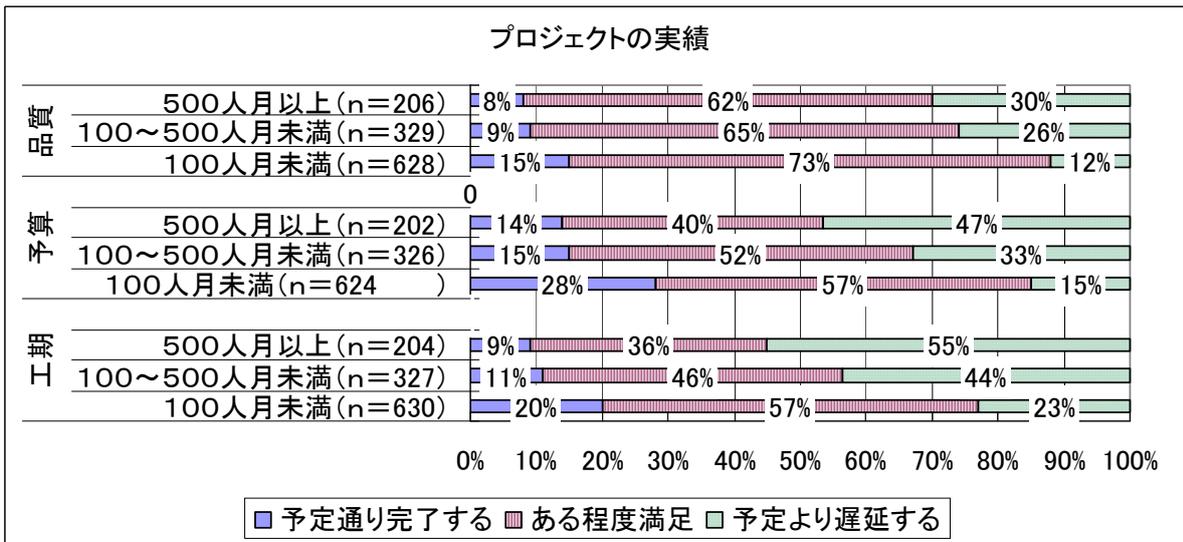
☆2007年度は開催せず
 ★2007年度開催予定

Agenda

1. 情報システム開発の実態、課題と対策
2. システム開発における工期の評価
3. システム開発における品質の評価
4. プロジェクト成功のための重要ポイント
5. システム保守についての分析

1. 情報システム開発の実態、課題と対策

日本企業におけるシステム開発の実績



企業IT動向調査2007

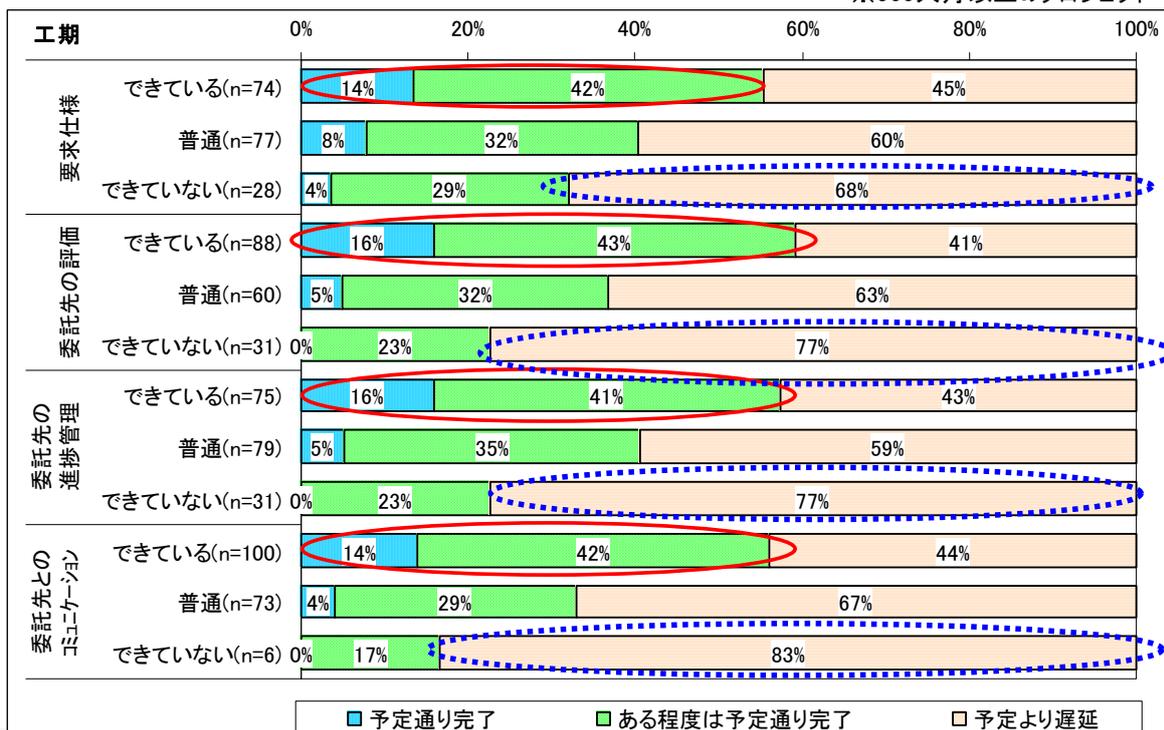
500人月以上のプロジェクトはおおよそ工期遅延5割、予算オーバー5割、品質不満が3割、5:5:3の法則がここ数年間継続中



Copyright(C) 2007 JUAS All rights reserved

ユーザー側が適切に対応ができていない企業ほど大規模プロジェクトの工期は予定通り収まる傾向 **なお工夫が必要**

※500人月以上のプロジェクト

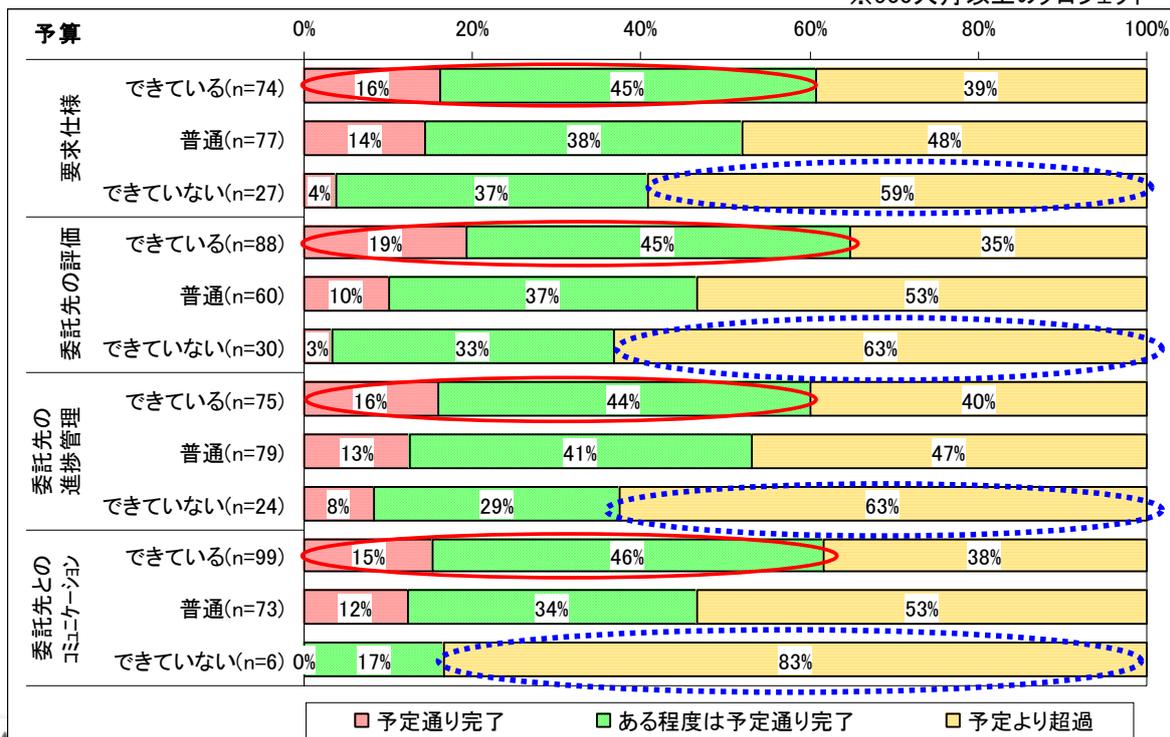


Copyright(C) 2007 JUAS All rights reserved

企業IT動向調査2007

ユーザー側が適切に対応している企業では、大規模プロジェクトの予算も同様に予定通り収まる傾向。 **しかし、なお改善の余地大**

※500人月以上のプロジェクト

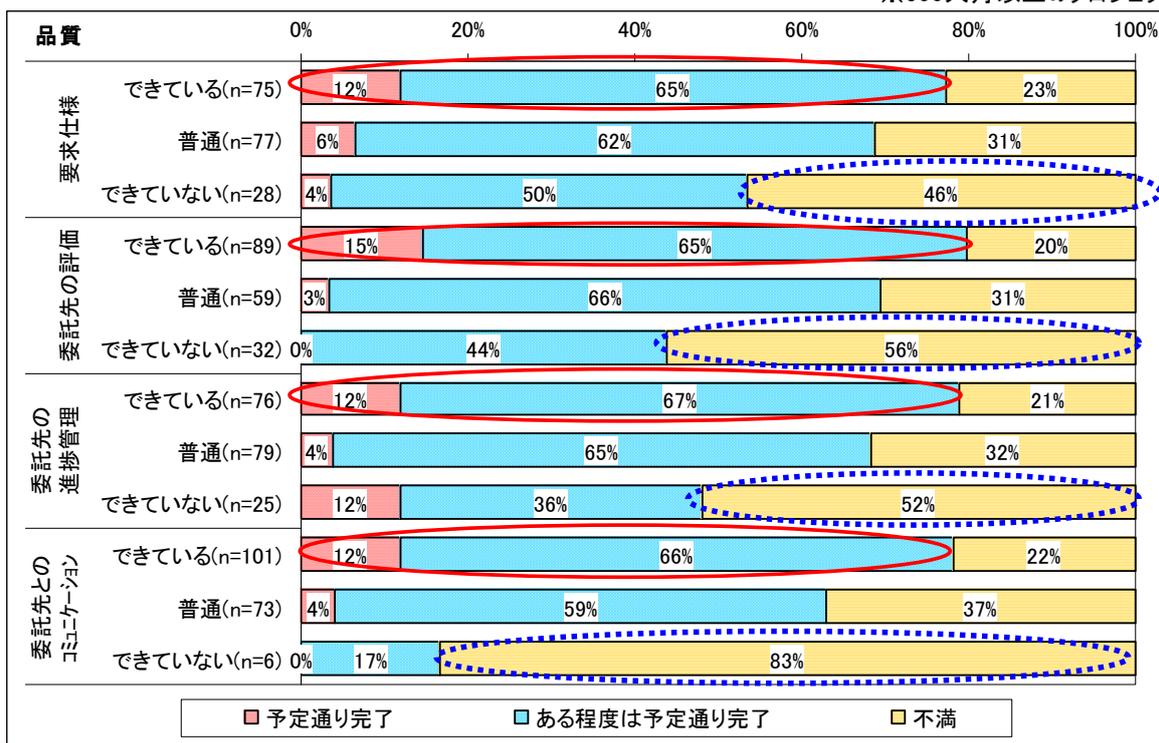


Copyright(C) 2007 JUAS All rights reserved

企業IT動向調査2007

品質についても同様に、ユーザー側の対応が適切である企業ほど満足度のいく仕上がりを得ている

※500人月以上のプロジェクト

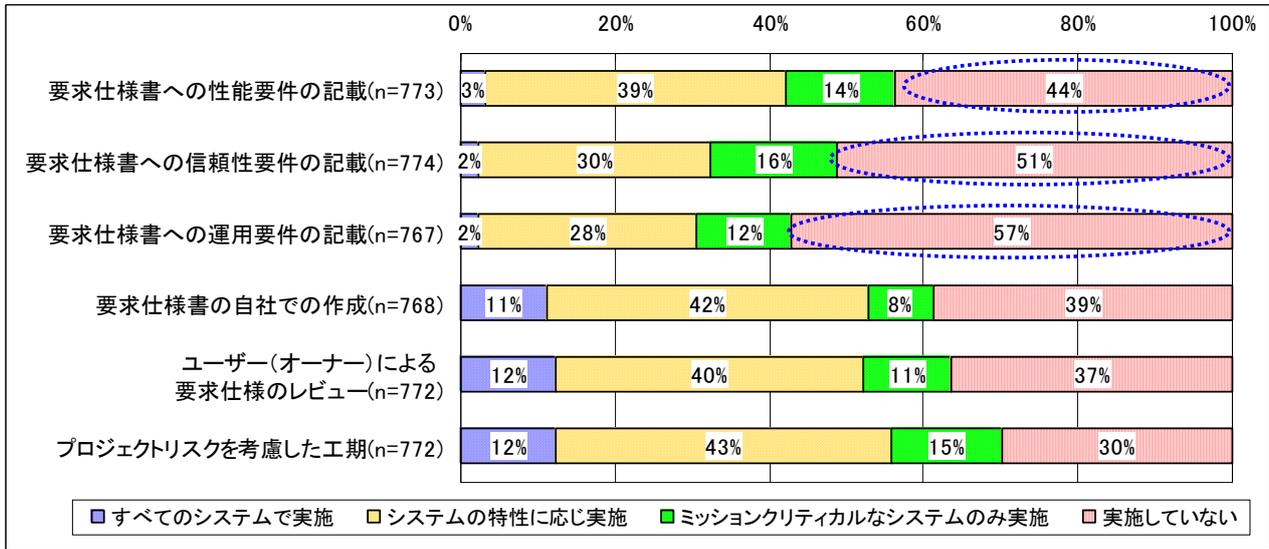


Copyright(C) 2007 JUAS All rights reserved

企業IT動向調査2007

企画・要求仕様段階では要求仕様への「性能」「信頼性」「運用」の非機能要件の記載充実が課題

企画・要求仕様段階での信頼性向上のための施策

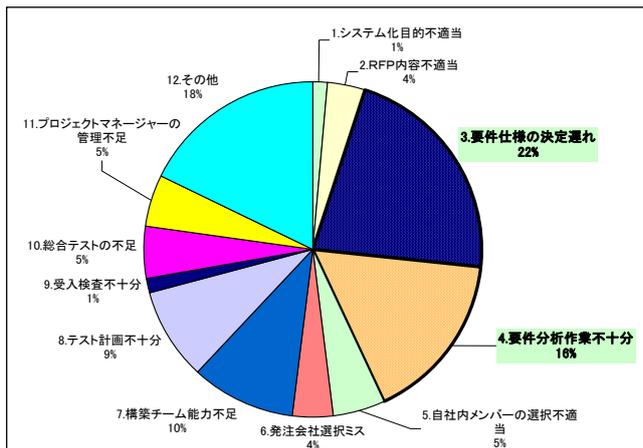


企業IT動向調査2007



Copyright(C) 2007 JUAS All rights reserved

工期遅延の理由:失敗の4割は要求仕様書の問題

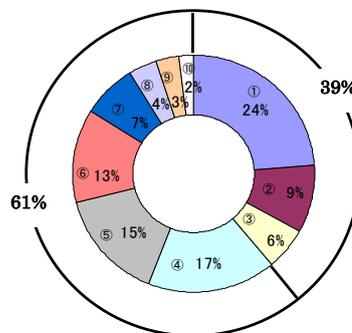


工期遅延理由
ソフトウェアメトリクス調査

「機能仕様をどのように書けばよいのか?」を記述したガイドはない

2006年、JUAS・UVC (user vender collaboration) プロジェクトにて「要求仕様ガイドライン」を発行

要求仕様遅延、要件定義不十分などのユーザーの責任は40%



品質とコストの関係(JISAデータ)

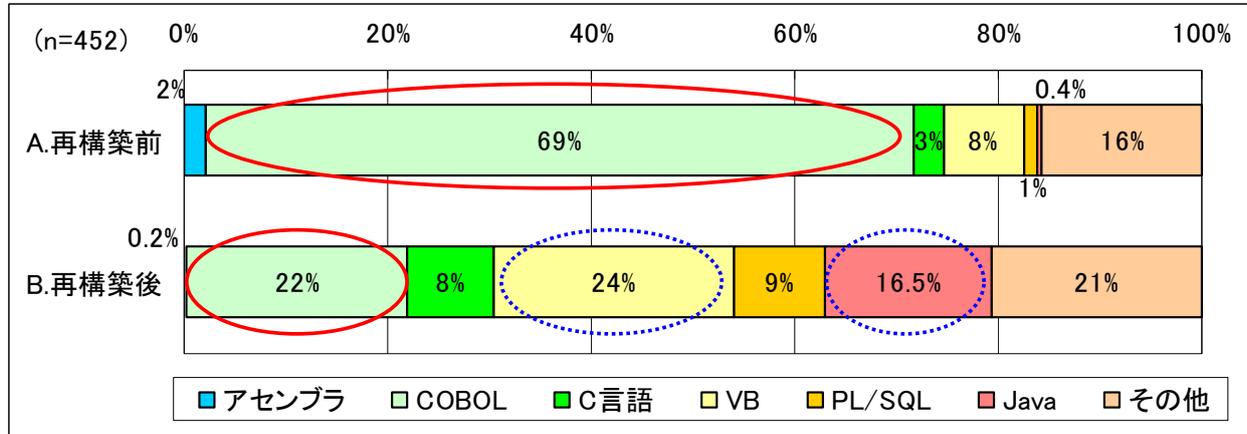


Copyright(C) 2007 JUAS All rights reserved

システム再構築におけるプログラム言語

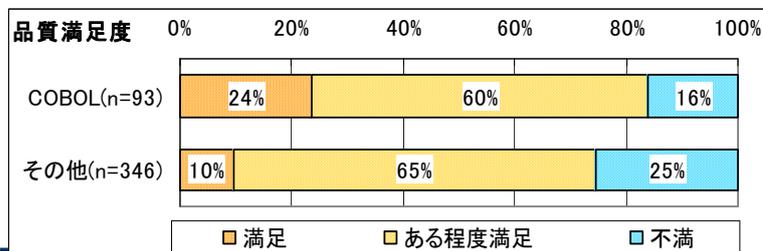
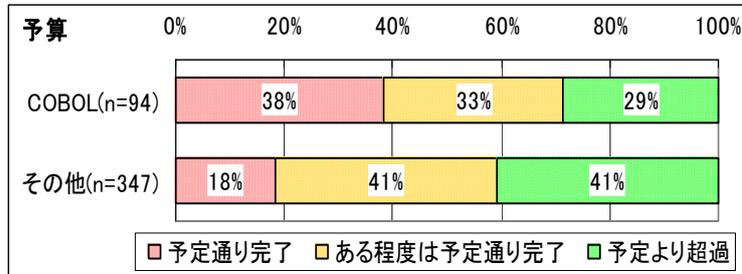
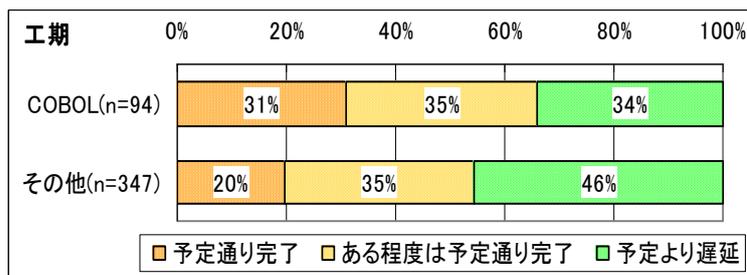
•「COBOL」が再構築前:69%→再構築後 22%に。

•「VB」と「Java」が急増、「VB」は8%→24%、「Java」は1%未満→17%に。



企業IT動向調査2006

システム再構築におけるQCD評価



COBOLで再構築を行った場合、他の言語と比べ工期・予算・品質の評価が高い



COBOLの安定性は大

再構築後のプログラム言語と再構築プロジェクトのQCDの評価 (企業IT動向調査2006)

見える化の必要性 目標値を持った管理を！

あなたは商品・サービスを買う時に、製造元のプロセスを考えて買いますか？
それとも商品の質・価格・納期で買いますか？

良い商品・サービスを作る方法とは？

①製造プロセスを確立すること(プロセス志向？)

* ISO * CMM

②最終商品の質(目標)を確保すること(プロダクト志向)

* ハードウェア……6シグマ……欠陥商品は直ぐに取り替えます

* ソフトウェア………バグがあるのは当たり前???

* Plan→Do→Check→Action

* EASE (Empirical Approach to Software Engineering) 奈良先端大学 鳥居教授

* 目標があるから、実績も評価でき改善アクションが見えてくる

* 貴社のシステム開発の品質？保守の品質？運用の品質？の

目標値とコストの関係は明確ですか？→ユーザとベンダー間の常識が必要

(優秀な商品・人が正當に評価される情報化社会を)

言語には関係ない



見える化のプロセス(JUAS・ソフトウェアメトリクス調査)

年度	開発	保守	運用
2004	開発プロジェクトの工期・品質・生産性		
2005	データの増加と精度の向上	保守プロジェクトの推進体制・カバー範囲・品質・	
2006	調査拡大	データの増加と精度の向上+対策の分析	運用体制・管理目標と実態(事前調査)
2007		調査拡大	本格調査

ソフトウェアメトリクス調査から得られた知見の例

- 開発工期: 2.4 × (投入人月の立方根) 品質: 1件/500万円は40%
- 保守範囲は3万LOC~100万loc/保守要員、平均は20万LOC
- 保守品質は85%/1回で受入完了 利用開始時でも7%の欠陥は残る

目標の提示(プロセスの改革)→更なるデータ蓄積と活用・普及対策



ソフトウェアメトリクス調査 プロフィール

収集条件:

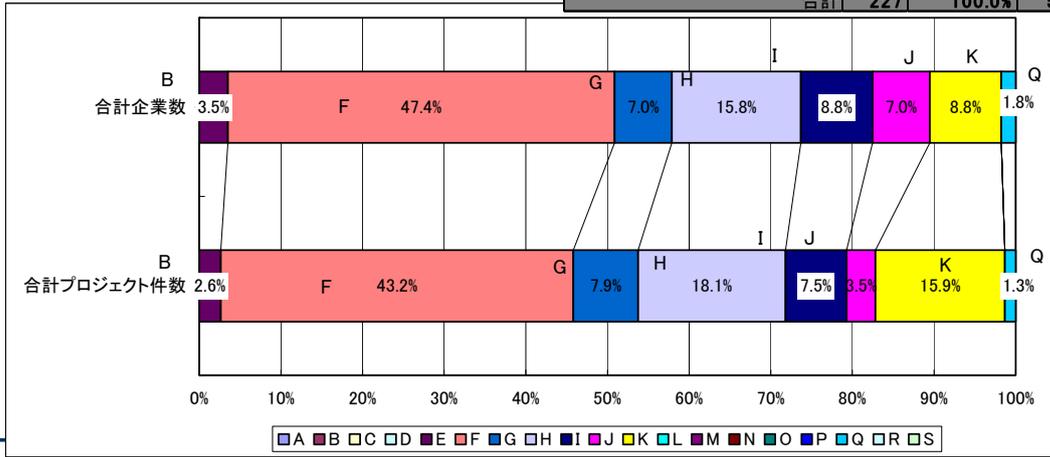
- ①「過去2年以内に開発完了」
- ②「開発コストが500万円以上」
- ③「新規または、改修プロジェクト」
(システム保守プロジェクトやマイナーチェンジの改修プロジェクトを除く)

集計結果:

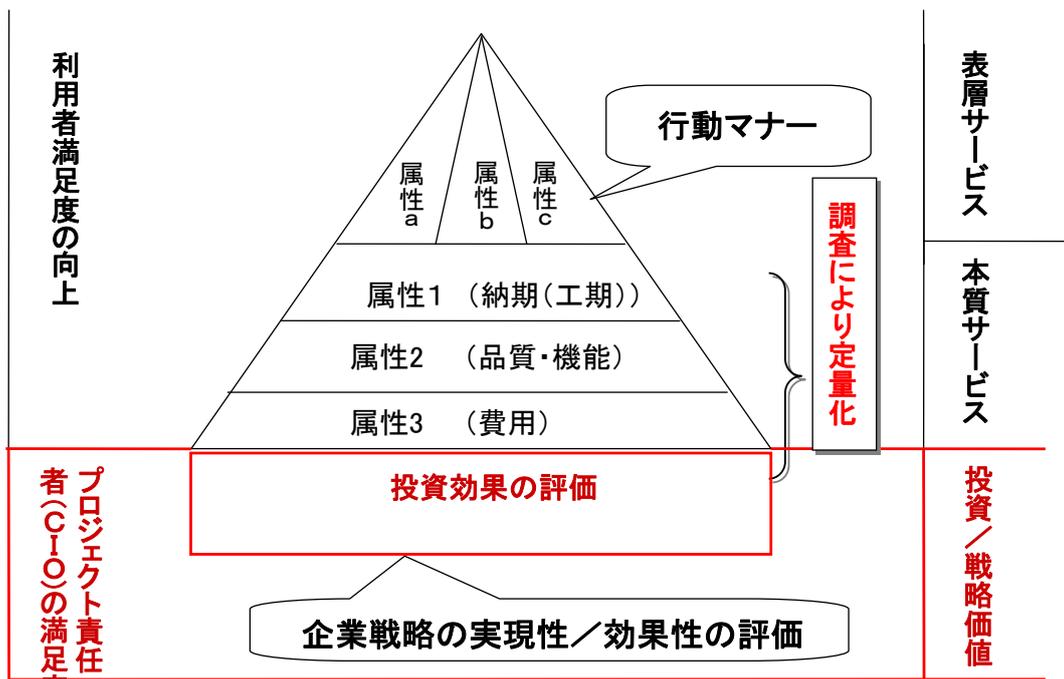
57、227プロジェクト/2006(開発+保守)

業種分類	合計プロジェクト件数	合計企業数
A. 農業	0	0.0%
B. 林業	0	0.0%
C. 漁業	0	0.0%
D. 鉱業	0	0.0%
E. 建設業	6	2.6%
F. 製造業	98	43.2%
G. 電気・ガス・熱供給・水道業	18	7.9%
H. 情報通信業	41	18.1%
I. 運輸業	17	7.5%
J. 卸売・小売業	8	3.5%
K. 金融・保険業	36	15.9%
L. 不動産業	0	0.0%
M. 飲食店・宿泊業	0	0.0%
N. 医療・福祉	0	0.0%
O. 教育・学習支援業	0	0.0%
P. 複合サービス業	0	0.0%
Q. サービス業	3	1.3%
R. 公務	0	0.0%
S. その他	0	0.0%
合計	227	100.0%
		57
		100.0%

図0-1 合計企業数:57社、プロジェクト数:227件



JUASユーザー満足度コンセプトを基にデータ収集 (JUASが考える、サービス充実度と満足度の関係)



(参考)顧客満足型マーケティングの構図 有斐閣社1994 嶋口充輝 慶応義塾大学大学院 経営管理研究科 教授

開発生産性指標

見積要因比較: 4つの評価要素を使い分ける

比較項目		FP	LOC	人月	価格
①この機能の価格はいくらか?	実績のあるスクラッチ	◎ 概算評価 画面数、帳票数を基に試算可能 (GUIの複雑性の評価に難あり)	○過去の実績からの推定	○過去の実績からの推定	○過去の実績からの推定
	実績の無いもの		○画面数、帳票数を基に試算可能(GUIの複雑性)	△LOCから試算可能	△人月から試算可能
	パッケージ	×ユーザーは評価困難 FP, LOCはベンダーしか判らない	×ユーザーは評価困難 FP, LOCはベンダーしか判らない	×ユーザーは評価困難 ベンダーのマネージメントしだい	○横並び評価は可能
②工期試算		◎FPから人月さらに工期 (COCOMO法など)	・LOCから人月換算	◎人月→工期	△過去の実績からの推定
③生産性評価		○ 総FPと投入人月の概算評価は可能 ○ 詳細設計～UT迄は個別評価も可能	○総LOCと投入人月の概算評価は可能	○FP/人月、LOC/人月の概算評価は可能	¥/FPあるいは¥/LOC
④品質評価	スクラッチ	◎欠陥数/FPが可能	◎欠陥数/ベンダー指定言語のLOCが可能	◎欠陥数/人月が可能	◎欠陥数/価格が可能
	パッケージ本体 (ユーザーの立場で)	×自社で見つけた欠陥数は可能(部分的評価)	×自社で見つけた欠陥数は可能(部分的評価)	×自社で見つけた欠陥数は可能(部分的評価)	◎欠陥数/価格で概要評価
	パッケージの追加、修正	◎欠陥数/FPが可能	○欠陥数/ベンダー指定言語のLOCは可能	◎欠陥数/人月が可能	◎欠陥数/追加のための価格で概要評価
スケジュール管理	基本設計～完了	×作業計画に反映しがたい	×作業計画に反映しがたい	◎WBSで人月使用可能	◎EVMで人月、価格あわせて使用可能



目標値の設定と分析方法

品質目標があると、品質満足度は向上する



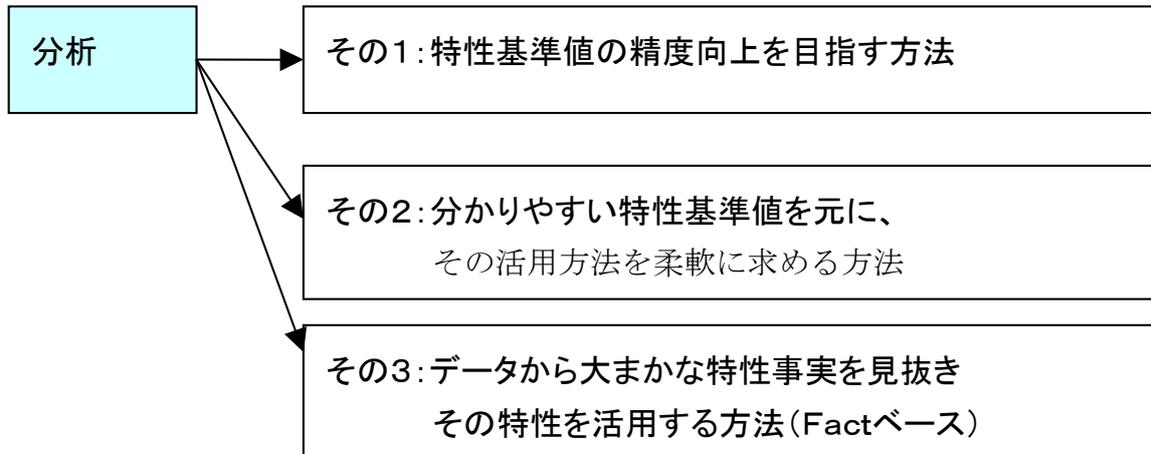
品質目標の提示と品質満足度の関係(企業IT動向調査2005)

	品質基準			計
	有り	無し	記入なし	
件数	44	54	1	99
比率	44.4%	54.5%	1.0%	100.0%
平均欠陥率	0.66	1.23	0.33	0.97
最大欠陥率	3.25	16.56	0.33	16.56
最小欠陥率	0.00	0.00	0.33	0.00

品質基準の有無と品質実績(ソフトウェアメトリクス調査2006)



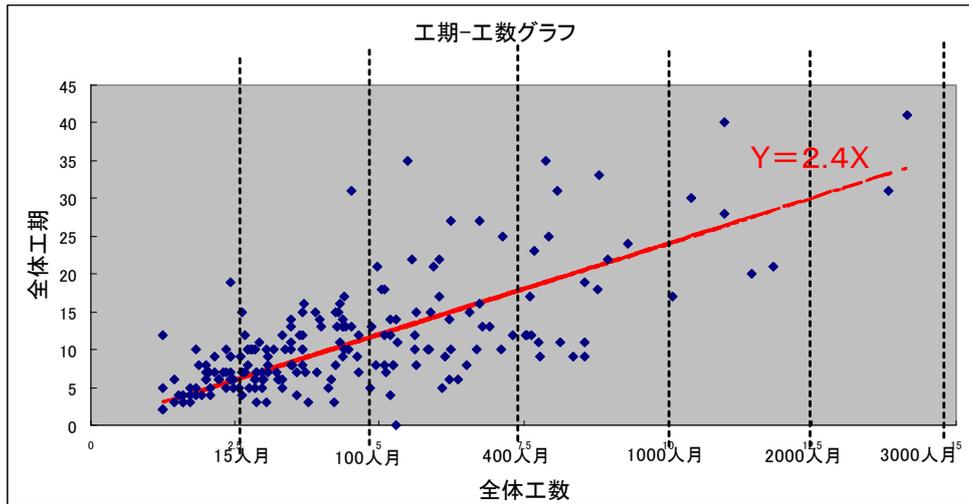
目標値の設定と分析方法



2. システム開発における工期の評価

(JUAS ソフトウェアメトリックス調査2007)

標準工期(適正工期)の考察



- COCOMO法によれば $工期 = 係数 \times (投入人月)^n$ 乗根 であるが、この各係数を詳細に求めても、実際は活用し難いので、**工期 = 係数 × (投入人月の立方根)** として活用しやすい形にした。
- プロジェクト全体工数と、全体工期がともに記入されている**203プロジェクト**について、**工数の3乗根と工期の関係**をグラフ化し、回帰直線を引いた。
- 工期・工数共に、実績の回答がある場合には**実績の工期・工数**を、計画しか回答がない場合には**計画工期・工数**を採用している。



工期の評価尺度とアクション

標準工期と実行工期の差(短縮率%)に着目してノウハウを蓄積する

	標準より長い工期	標準	25%工期短縮	25%以上工期短縮
工期の標準の考え方	金融等欠陥の発生を無くしたい品質重視のプロジェクトの場合	工数の立方根の2倍 (例: 1000人月のプロジェクトは20箇月)	・ユーザの要望 ・流通業のシステム化などに多い。	ユーザのやむを得ない外的事情で実施する場合 (対コンペ戦略、新商品の販売、株式の上場、企業の統合など)
スケジュールリングの対応策	十分なシステムテスト期間の確保	中日程計画の充実 (役割分担別WBS管理)	中日程計画の充実 (週間別管理)	小日程計画の充実 (日別管理)
その他の対応策	・品質重視のテスト計画書及びテストケースの緻密化 ・安定稼動のための分割立ち上げ等	・WBSによる総合計画と局面化開発 ・レビューの徹底 ・テストケース充実 ・コンバージョンデータのフル活用 ・確実な変更管理	同左 + ・PGの選抜 ・標準化の徹底と実力のある一括外注の採用。 ・システム範囲、対象の部分稼動 ・RAD+DOA ・性能事前検証 ・変更管理の強化	同左 + ・ベテランPMIによる采配と会社あげでの協力及び監視 ・パート図での計画 ・ベストメンバー選出 ・クリーンルーム手法 ・二交代制の配置 ・顧客主体のテストチーム設置 ・パッケージの活用 ・部品の再利用 ・オープンな進捗情報管理

工期短縮率の%に応じた対策を採用する。
この短縮率とアクションの関係は各企業によって異なる。



工期の適正度と遅延度の関係

標準工期 = 2.4 X 工数の三乗根と考え、工期が標準工期に対してどの程度短いかを表す尺度として、以下のように工期短縮率を定義する。

工期短縮率 = 1 - (実工期 ÷ 標準工期)。これを計算し分布を見た。

標準工期に対して、単工期、長工期の基準を、それぞれ全体の25パーセント程度となるように、**工期短縮率 > 0.2を短工期、工期短縮率 < -0.4を長工期**と定義した。

工期適正度	短工期	適正工期	長工期	全体
件数	50	104	48	202
割合	24.7%	51.5%	23.8%	100%
平均工期(月)	7.8	10.5	15.8	11.2
20%以上の遅延率	10.9%	12.0%	12.8%	11.9%

短工期でもプロジェクトマネジメント次第で遅延は防げる。
ただし異常な工期短縮率(企業によって異なる)のプロジェクトは失敗の可能性が高い

工期と欠陥率との関係

工期が標準よりも短かすぎると、十分なテストが出来ず、カットオーバー後にバグが多発すると
の仮説を基に、工期乖離度と欠陥率の関係に関する分析を行った。

工期乖離区分	欠陥率						計
	0	0.25未満	0.5未満	1未満	3未満	3以上	
不明	件数			1	1		2
	平均欠陥率			0.33	0.84		0.59
	最大欠陥率			0.33	0.84		0.84
	最小欠陥率			0.33	0.84		0.33
長工期	件数	5	5	9	6	6	38
	平均欠陥率	0.00	0.11	0.35	0.66	1.92	6.14
	最大欠陥率	0.00	0.23	0.43	0.85	2.93	16.56
	最小欠陥率	0.00	0.05	0.25	0.50	1.00	3.05
標準工期	件数	8	28	14	18	12	82
	平均欠陥率	0.00	0.12	0.38	0.67	1.77	3.20
	最大欠陥率	0.00	0.24	0.48	0.97	2.73	3.29
	最小欠陥率	0.00	0.00	0.29	0.51	1.00	3.11
短工期	件数	2	18	5	2	5	32
	平均欠陥率	0.00	0.10	0.33	0.63	1.67	0.41
	最大欠陥率	0.00	0.19	0.42	0.67	2.64	2.64
	最小欠陥率	0.00	0.01	0.25	0.59	1.05	0.00
計	件数	15	51	29	27	23	154
	平均欠陥率	0.00	0.11	0.36	0.68	1.79	5.49
	最大欠陥率	0.00	0.24	0.48	0.97	2.93	16.56
	最小欠陥率	0.00	0.00	0.25	0.50	1.00	3.05

- 長工期の平均欠陥率が最も悪いという、どちらかという、逆の傾向が見られた。
- 設計初期に十分なレビューが行われずにスケジュールが遅れ、仕様変更が多発し、テストも十分に行われず、結果的に品質が悪くなる、実態が現れている。
- 短工期であるからといって、品質を低下させて良いことにはならず、ベンダーが苦勞して納品している状況がうかがわれる。

規模(工数)別フェーズ別工期と比率

規模別フェーズ別工期比

PJ規模(工数)	件数	設計工期	実装工期	テスト工期	テスト比率
10人月未満	2	1.50	3.50	4.00	44.4%
50人月未満	45	2.31	3.38	2.73	32.5%
100人月未満	12	3.75	5.50	5.33	36.6%
500人月未満	33	4.42	5.18	5.06	34.5%
500人月以上	12	5.75	7.08	5.75	30.9%
記入なし	9	3.00	5.78	3.67	29.5%
総計	113	3.49	4.72	4.11	33.4%

- 設計工期:実装工期:テスト工期は、おおよそ平均で**6:8:7**になった。
- テスト工期の比率が高い。
- 一昨年の同調査の値が3:3:4、昨年度が5:7:7であるので、年々設計工期の比率が短くなっている事になる。
- 今年度の調査でははじめて実装工期の比率が最も高くなった。

工期フェーズ別・工数比率

工数	件数	設計工数率	実装工数率	テスト工数率
~10人月	9	20%	39%	41%
~50人月	49	24%	43%	33%
~100人月	14	21%	38%	41%
~500人月	33	26%	41%	33%
500人月~	12	29%	37%	34%
合計	117	24%	41%	35%

ソフトウェアメトリクス調査2007

- ・規模が大きくなると設計工期比が増加する
- ・テスト工期比が大きいのは、このフェーズで品質向上などやり残した仕事を実施する調節期間の意味が大きいからである。改善の可能性は大である

工期遅延度 計画値 VS 実績値

・工期の計画値、実績値がともにとれたプロジェクトは231件中**209件**であった。
 ・(実績工期－計画工期)／計画工期 を工期遅延度と定義してプロジェクト規模別の遅延度分析を行った。

		遅延度						総計	遅延度 20%以上の割合	
		予定より早い	予定通り	10%未満	20%未満	50%未満	それ以上			
規模 (工数)	～10人月	件数	2	17		1	2	2	24	16.7%
		比率	8.3%	70.8%	0.0%	4.2%	8.3%	8.3%	100.0%	
	～50人月	件数	4	48	2	10	5	3	72	
		比率	5.6%	66.7%	2.8%	13.9%	6.9%	4.2%	100.0%	
	～100人月	件数	1	17	3		6		27	
		比率	3.7%	63.0%	11.1%	0.0%	22.2%	0.0%	100.0%	
	～500人月	件数	7	34	5	3	1	1	51	
		比率	13.7%	66.7%	9.8%	5.9%	2.0%	2.0%	100.0%	
	500人月以上	件数	1	9	6		3		19	
		比率	5.3%	47.4%	31.6%	0.0%	15.8%	0.0%	100.0%	
	未記入	件数		9	2		5		16	
		比率	0.0%	56.3%	12.5%	0.0%	31.3%	0.0%	100.0%	
	総計	件数	15	134	18	14	22	6	209	
		比率	7.2%	64.1%	8.6%	6.7%	10.5%	2.9%	100.0%	

- 予定通りの工期を確保できた割合は**70%以上**と高水準である。
- 規模の大きなプロジェクトほど、遅延度が高いとは言い切れないが、500人月以上の大きなプロジェクトでは、**計画内で収まる率は53%**と平均よりも低い。
- 失敗したプロジェクトのデータが調査票にて報告されていない事も考えられる。



工期遅延理由分析

工期遅延理由の件数

工期遅延理由	規模(工数)						合計
	10人月未満	50人月未満	100人月未満	500人月未満	500人月以上	記入なし	
1.システム化目的不適当			1				1 (0.5%)
2.RFP内容不適当		2	1	1	6	1	12 (5.7%)
3.要件仕様の決定遅れ	5	14	3	11	6	5	44 (20.9%)
4.要件分析作業不十分	5	7	2	5	3	6	28 (13.3%)
5.開発規模の増大	2	4	3	11	5	4	29 (13.7%)
6.自社内メンバーの選択不適当		1	3	1	3		9 (4.3%)
7.発注会社選択ミス			2		3	1	8 (3.8%)
8.構築チーム能力不足	1	6	3	5	1	2	18 (8.5%)
9.テスト計画不十分	3	4	5	2	3	1	18 (8.5%)
10.受入検査不十分				4	1		5 (2.4%)
11.総合テストの不足	1	5		3	2	2	13 (6.2%)
12.プロジェクトマネージャーの管理不足	2	1	3	2	2	2	12 (5.7%)
13.その他	1	4	2	2	1	4	14 (6.6%)
合計	23	52	23	57	26	30	211 (100.0%)

- ・理由コード1から5までの40%が、要件定義の問題と考えられる。これはユーザーとベンダーの両者の問題である
- ・上位工程での不具合が、全体工期の遅延につながる恐れが最も多いことがわかる。



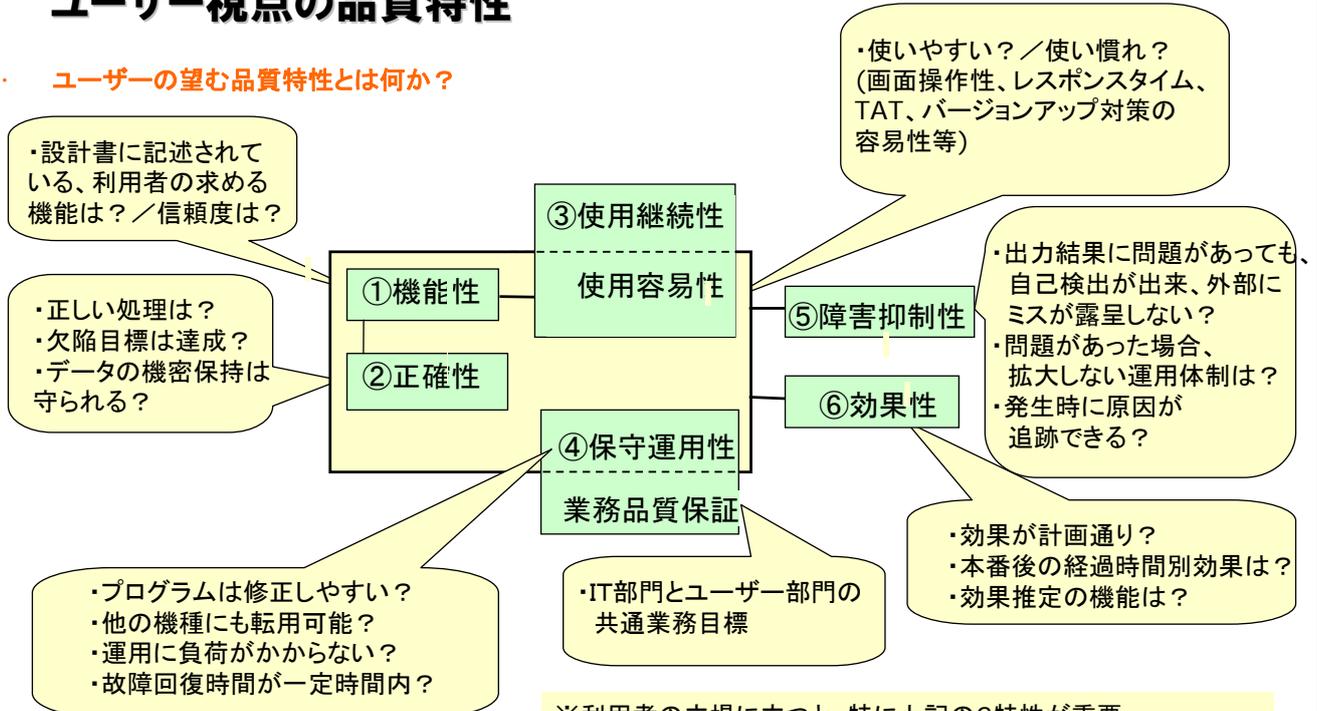
3. システム開発における品質の評価

(JUAS ソフトウェアメトリクス調査2007)



ユーザー視点の品質特性

ユーザーの望む品質特性とは何か？

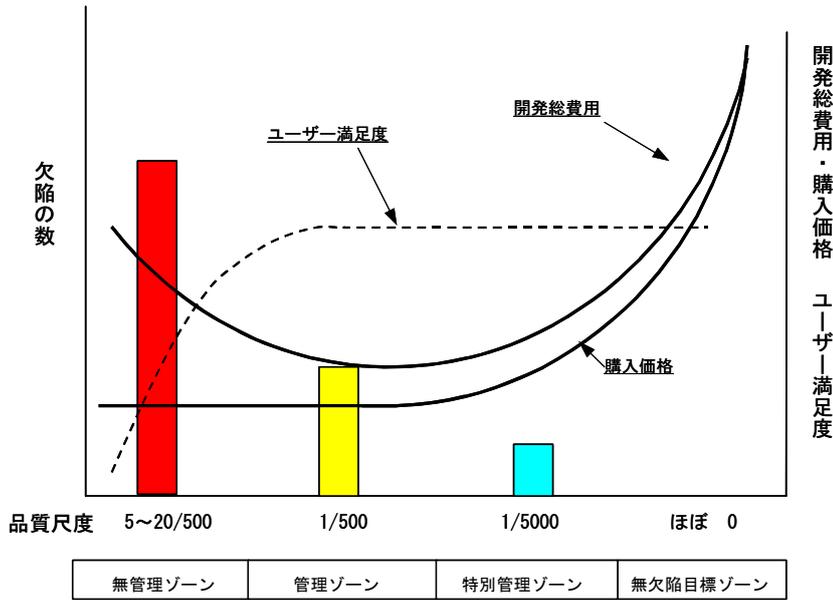


※利用者の立場に立つと、特に上記の6特性が重要。
※(JIS X0129-1994)の品質特性は機能性(5)、信頼性(3)、使用性(3)、効率性(2)、保守性(4)、移植性(4)の6特性と21の副特性(括弧内個数)から構成されている。



品質と費用の関連性

「仮に貴社で500万円に1件の品質保証ができている場合に、5,000万円に1件の障害に抑えて納入してくださいと顧客から要望が来たら、何倍の費用をもらえば対応が可能ですか？」



注1 品質尺度：(納入時～安定稼働期迄の欠陥個数) / 開発費用 (万円)
 注2 開発総費用と購入価格のギャップはテスト結果の確認、修正結果の確認のために要するユーザー側の負荷増加費用をイメージ化したもの。



非機能仕様の定義方法(ISO9126)

非機能	サブ非機能	概要
機能性 (非機能としては扱わない)	(合目的性)	要求仕様書に書かれた機能に対する充足度
	正確性	期待した結果に対しえられた実際の結果の対比(計算の精度検証を含む)
	セキュリティ	セキュリティ対策の確認
	相互運用性	データ互換性の正確性
	標準適合性	適用可能な規範、規格、協定との適合度
信頼性	成熟性	過負荷状態におけるシステムの故障頻度
	障害許容性	障害時に必要なレベルを維持している機能数/全機能数など
	回復性	障害時に復旧するまでの時間など
使用性	理解性	システム、操作ガイドなどの理解のしやすさ
	習得性	習得の容易性
	運用性・操作性	使いやすさ
	魅力性	魅力の持てる商品になっているか
効率性	時間効率性	正常時、過負荷時の時間効率性(レスポンスタイム、スループット等)
	資源効率性	正常時、過負荷時の資源効率性(メモリー消費量、ディスク使用量など)
保守性	解析性	システムが解析しやすいように作られているか
	変更性	システムが変更しやすいように作られているか
	安定性	外部から変更を加えた場合にもトラブルが少ないか
	試験性	テストしやすいように作られてあるか
可搬性	導入容易性	ポーティング作業のしやすさ、(適応性、共存、互換性、可搬性など)



稼働率とレスポンスタイムの管理状況

	オンライン稼働率	レスポンスタイム (社内向け)
目標を持っている企業の割合	74%	13%
実績を把握している企業の割合	43%	20%

- ・レスポンスタイムの定義から整理・確立する必要がある
- ・インターネット活用システムのレスポンスタイムはほとんど保証されていない

ほとんどのシステムは、End to Endのシステムを把握しておらず、利用者に「遅くなった」と言われて始めて、調査を始める受身型のシステムになっている

レスポンスタイムの種類と指定

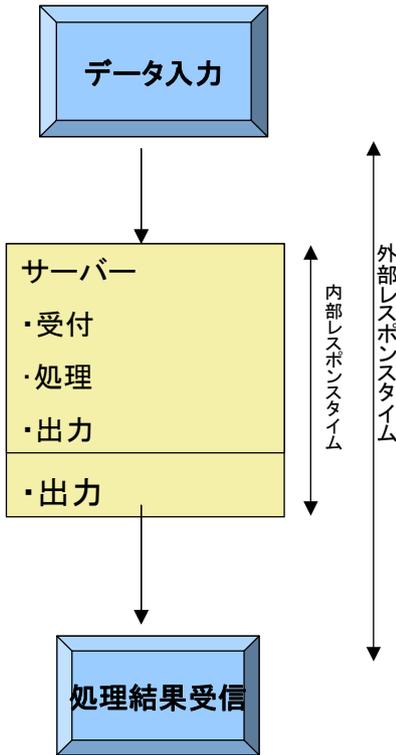
レスポンスタイムの種類	説明	保証
1: 開発マシン環境 端末機は1台	開発会社のテスト環境1台の端末機でのレスポンスタイム	A
2: 開発マシン環境 端末機はn台	開発会社の端末機をつないでのテスト環境でのレスポンスタイム、(n台同時入力して最後の入力の応答時間)	A
3: 本番環境 端末機は1台	本番環境ではあるが、他の入力はない場合のレスポンスタイム	A、B
4: 本番環境 端末機はn台 (n<99)	本番環境で同種データn個同時入力した場合の最終テストデータのレスポンスタイム	B
5: 本番環境、端末機はn台 (n>99)	本番環境で同種データn個同時入力、干渉が起きる他のプログラムが背景にながれている場合の、最終入力のテストデータのレスポンスタイムまたは90%以上の入力が制限時間以下	B
6: 本番環境、端末接続数は(n>100/1秒間)	本番環境で同種データn個同時入力した場合の最終テストデータのレスポンスタイム(nはインターネットからの三桁/秒以上の入力件数を想定した場合)ただし同時に干渉が起きる他のプログラムが背景にながれている。最後の入力データに対するレスポンスタイム または90%以上の入力が制限時間以下	C

A: プログラム、DBの作成方法の確認(保証者はプログラム作成企業)

B: プログラム、DBの作成方法の確認(保証者は本番環境提供ベンダー)

C: 本番環境でのレスポンスタイムの確認(対策としてGRIDコンピュータなどの活用も含む)

レスポンスタイムの把握



- ・入力画面表示
 - ・データ入力
 - ・データ送信
 - ・a.出力時間把握
-
- ・b.受信時間把握
 - ・(a-b)時間計算
 - ・サーバに送信
 - ・受信結果表示
 - ・(a-b)-(サーバ内処理時間)が伝送時間

この機能を、アプリケーションシステムに組み込むことによって、ユーザー体感のレスポンスタイムを把握できる。

しかし一般には行われていない。
(理由)

- ・余分な処理時間がかかる
- ・組み込む操作がわずらわしい
- ・企業内システムでは回線速度は問題にならない

常時このレスポンスタイムを把握しておき、予防保全を実施することが必要。利用者から「最近レスポンスタイムが遅くなった」といわれてから、対策をとるので遅い。

ユーザーによる非機能仕様検証(2007年度の課題)

どのフェーズで何を何と比較するのか？
納得性のある評価尺度を創り出す

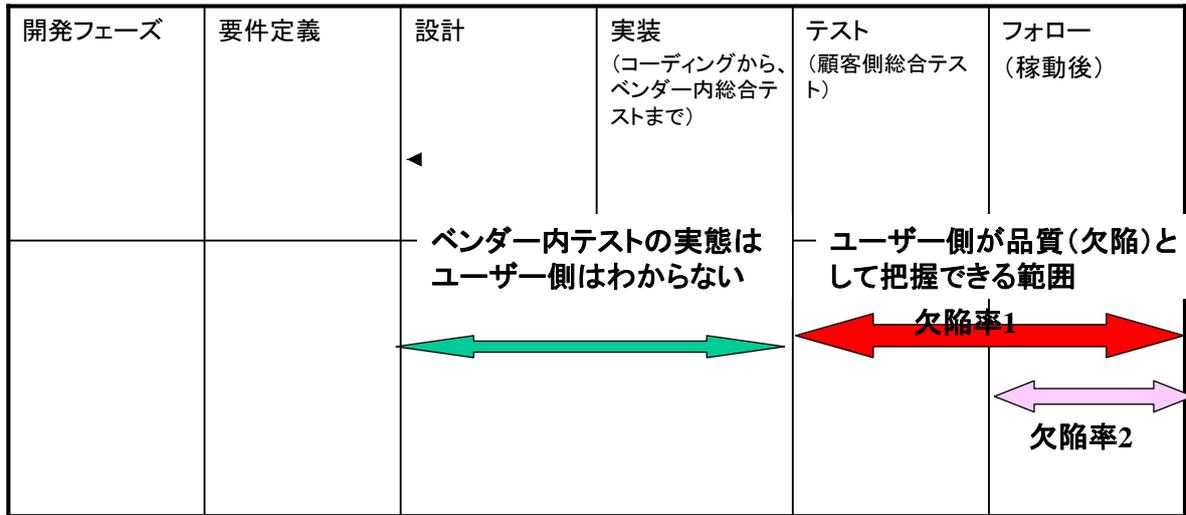
品質特性	副特性	説明	要求仕様書	設計書	プログラム	テストモジュール	単体テスト	結合テスト	総合テスト	併行運転	運用
機能性	合目的性		◎	◎			○	◎	◎	◎	◎
	正確性		◎	○		○	◎	◎	◎	◎	◎
	セキュリティ		◎	◎	◎				◎	◎	◎
信頼性	障害許容性		◎	○					◎	◎	◎
	回復性		◎	○	○				◎	◎	◎
使用性	理解性		◎	◎			◎	◎	◎	◎	◎
	学習性		◎	◎			◎	◎	◎	◎	◎
	操作性		◎	◎			◎	◎	◎	◎	◎
効率性	時間効率性	レスポンスタイム	◎	○			○	○	◎	◎	◎
	資源効率性						○	○	◎	◎	◎
保守性	解析性・変更性		◎	◎	◎						
	安定性									○	◎
	試験性		◎	◎		◎	◎	◎	◎	○	○
障害抑制性											
効果性											

*◎主作業として実施 ○従作業として実施

◎○は単なる現時点のサンプル

ただし◎であっても、作業内容、比較対称は大幅に異なる

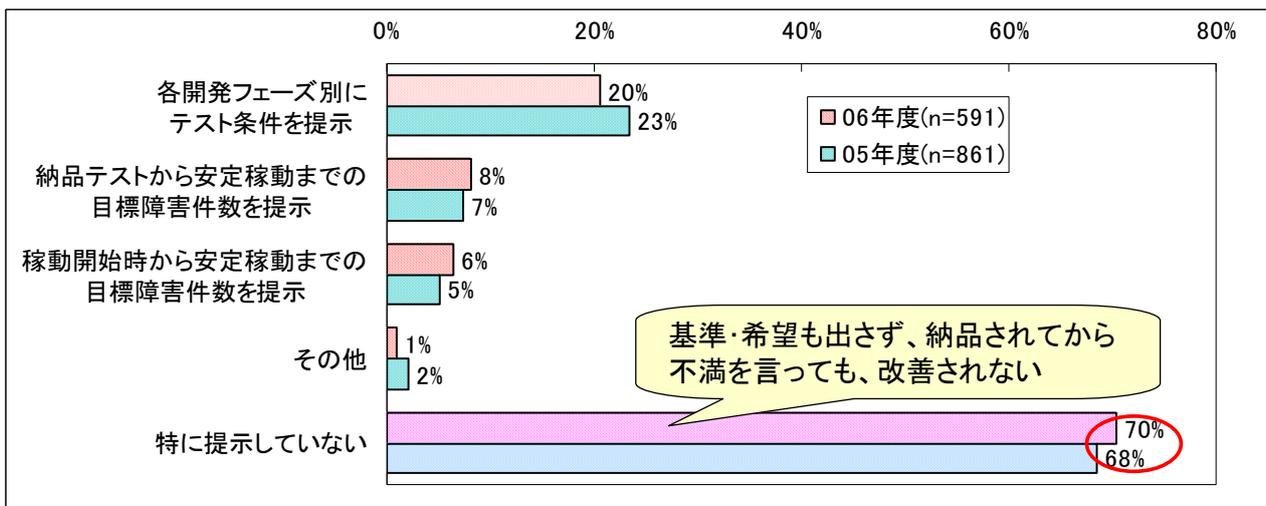
ユーザーの品質(正確性)の定義



欠陥率1 = 「ユーザが発見した欠陥数の密度」 =
 (総合テスト2～フォローのフェーズで発見された不具合の数) ÷ プロジェクト全体計画工数
 0.2個/人月(おおよそ1個/500万円)であれば、優秀である

欠陥率2 = 「稼動後に発生した欠陥数」 =
 (稼動後～フォローのフェーズで発見された不具合の数) ÷ プロジェクト予算
 大きな欠陥はなく、4個/億円以内であれば、一般的な企業内ユーザーは受入可能。

システム開発を外部に委託する際に、 品質目標を「特に提示していない」企業が7割



企業IT動向調査2007

自社基準がなければ、

- 納品以降の品質目標「5百万円に1個以内」、
 - 稼動後の品質目標は「1億円に1～4個以内」
- などの、JUASの目標値を活用してほしい

ユーザー評価品質の実態

	欠陥率						計
	Aランク	Bランク	Cランク	Dランク	Eランク	Fランク	
	0	0.25未満	0.5未満	1未満	3未満	3以上	
件数	15	51	29	27	23	9	154
比率	9.7%	33.1%	18.8%	17.5%	14.9%	5.8%	100.0%

- ・平均は0.8個。
- ・0.25個/人月は、おおよそ1個/500万円に相当するが、4割のプロジェクトが確保できている。
- ・これは国際的基準からみると非常に高レベルである

品質と価格の関係

	品質区分(欠陥率)						総計
	A(0)	B(~0.25)	C(~0.5)	D(~1)	E(~3)	F(3~)	
件数	9	38	24	24	17	8	120
単価(平均)	96.2	113.5	105.8	101.6	113.7	117.3	108.6
単価(最大)	140.4	272.7	162.5	285.7	175.7	250.0	285.7
単価(最小)	71.5	46.2	43.2	41.4	70.8	45.7	41.4

「品質が良いものは高い」はシステム開発の世界では証明されていない。
 もっとも安いプロジェクトが最も品質が良いとの逆の結果になっている。
 良い品質のプロジェクトは最小単価が高い(無茶な値切りをしていない)

発注者は品質について努力目標を提示する必要がある

外国と比較して、日本のソフトウェア品質は高い？

	日本	米国	インド	欧州他
プロジェクト数	27	31	24	22
生産性 プログラマーが1人月に記述 するコード行数	469	270	209	436
ソフトウェアの品質 システム導入後1年間に発見 された1Kあたりの不具合報告 (中央値)	0.020	0.400	0.263	0.225

出典:IEEE software(2003年11,12月号)
ソフトウェア開発の生産性と品質に関する国際比較

上記日本の値はJUASのソフトウェアメトリックス調査の実績と近似している



品質と価格の関係が理解しがたい理由

- ①ユーザーが品質目標を示していない場合が多い
したがって納入してくる品質が安定してこない
→目標を示せば変化する。
- ②高い品質を創り出すSE, PGの性格・作業分析が進んでいないのではないかと
・頭脳明晰、理解度の早いSE, PGがミスを犯す確率が低いとはいえない
・頭脳明晰、理解度が早くなくてもミスをほとんど犯さないSE, PGが存在する
(しかしこの緻密なSE, PGが優秀であるとはほとんどの企業は評価していない。
この種類のSE, PGは自分のした仕事をフォロー、見直しを着実にする。
決してあわてないなどの特質を備えている)
- ③システム実装(設計、コード化、テスト)のエンジニアリング化、計数管理を、さらに進めることにより「価額と品質」の関係は明確になる
・品質、生産性を価格および個人・企業評価に反映する仕組みを!



欠陥率と顧客満足度の関係

欠陥率と顧客満足度(品質):50人月以上

小規模プロジェクトでは満足度が甘くなる可能性があるため、50人月以上のプロジェクトで再計算した。

欠陥率		顧客満足度(品質)				計	満足率
		満足	やや不満	不満	未回答		
0	件数	6				6	100.0%
	平均	0.00				0	
0.25未満	件数	22	9		2	33	71.0%
	平均	0.09	0.11		0.11	0.10	
0.5未満	件数	8	6		2	16	57.1%
	平均	0.37	0.31		0.32	0.34	
1未満	件数	4	4	4		12	33.3%
	平均	0.67	0.77	0.64		0.69	
3未満	件数	6	4	2		12	50.0%
	平均	1.85	1.97	1.53		1.84	
3以上	件数	1	1			2	50.0%
	平均	3.05	4.35			3.70	
計	件数	47	24	6	4	81	61.0%
	平均	0.46	0.76	0.94	0.21	0.57	

- 欠陥率が多く、品質が悪くなると満足率も下がっている。
- 品質(欠陥率)で評価されるためには0.25個/人月以上の高品質にする必要がある
- 欠陥が多く発生してもフォローの仕方である程度の満足度が得られる場合がある

品質基準の有無と欠陥率

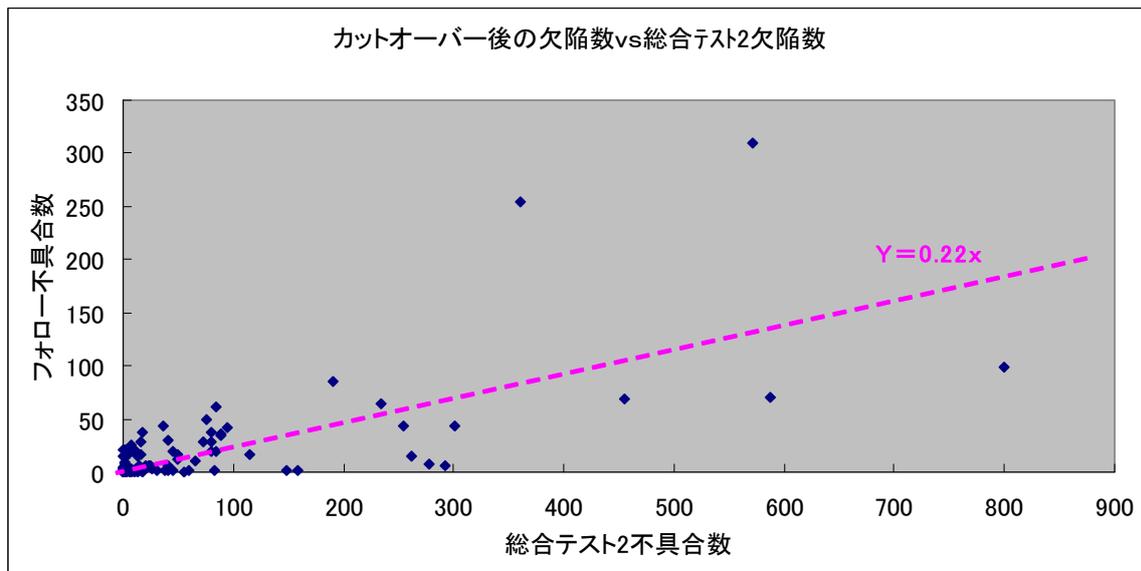
品質基準の有無と欠陥率

欠陥率の計算できた154プロジェクト、換算欠陥率が計算できた145プロジェクトについて、品質基準の有無と欠陥率の関係を調べた。

	品質基準					品質基準			
	有り	無し	記入なし	計		有り	無し	記入なし	計
件数	69	84	1	154	件数	65	79	1	145
比率	44.8%	54.5%	0.6%	100.0%	比率	44.8%	54.5%	0.7%	100.0%
平均欠陥率	0.60	0.99	0.33	0.81	平均換算欠陥率	0.37	0.78	0.17	0.59
最大欠陥率	3.25	16.56	0.33	16.56	最大換算欠陥率	2.62	11.89	0.17	11.89
最小欠陥率	0.00	0.00	0.33	0.00	最小換算欠陥率	0.00	0.00	0.17	0.00

- 換算欠陥率 = (大規模欠陥 × 2 + 中規模欠陥 + 小規模欠陥数 × 1/2) / 工数
- 換算欠陥率を用いて比較すると、品質基準を持っていないプロジェクトでは、**換算欠陥率が2.1倍**になっていた。
- 換算欠陥率を使用したほうが両者の差が広がって見える

ユーザー総合テストと稼動後の欠陥数の関係



顧客総合テストで見つかった欠陥数のおおよそ1/5のバグが稼動後に発生する。

総合テストで見つかったバグが1件/500万円だとすれば、4件/億円程度発生するので、この数値をユーザーに提示し稼動時のトラブル解消への努力を呼びかけると良い。

大規模プロジェクトの場合はかなりの数になるので、この欠陥数を減らしたければさらに総合テストを繰り返す必要がある。

再度総合テストを繰り返せば、 $0.2 \times 0.2 = 0.044\%$ の欠陥数の発生に抑えられる

4. プロジェクト成功のための重要ポイント

(JUASソフトウェアメトリックス調査2007)

生産性のデータ

(代表例) 1000人月以下のプロジェクトの場合

FPあたりの生産性	10FP/人月	規模が大きくなると生産性は下がる
KLOCあたりの生産性	1.0KLOC/人月	

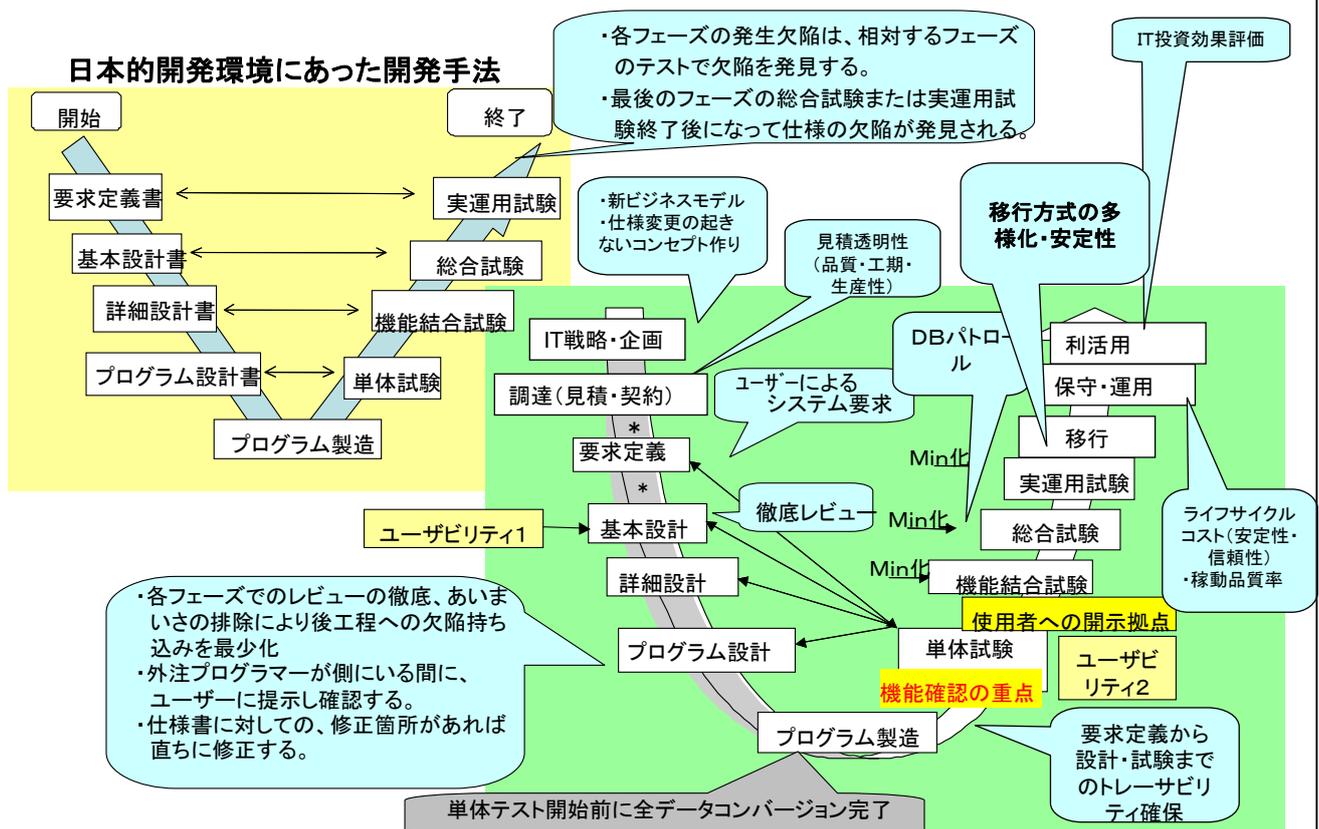
パッケージの追加開発	142万円/人月	パッケージ追加開発の単価は50%増し ただし工期の短縮、グローバル標準会計などの魅力有り
自社スクラッチ開発	95万円/人月	

予算オーバー平均額は5% (少しバッファを用意した方が安心)

詳細データはJUAS「ソフトウェアメトリックス調査2007」を参照



V字型開発からU字型開発へ(日本の開発環境に適した手法を)



U字型開発法(補足)

- ①企画段階のシステムコンセプト確立・・・
要求仕様の変更は、企画段階のユーザー・リーダーの理解度向上で解消
- ②見積の透明性・・・リスク要因についてのベンダーとの対話
- ③ユーザーによるシステム要求・・・利用者がシステム要求を、裏を読めはダメ
- ④徹底レビュー・・・作成時間の10%以上時間をかけて慎重に
- ⑤ユーザービリティの確保・・・ユーザービリティテストによる利用容易性確保
- ⑥UVC判りやすい仕様の書き方と仕様の一貫性
- ⑦目標を持って管理(EASE)・・・目標の有効性とフォロー
- ⑧単体テストの徹底・・・データコンバージョンプログラムは単体テスト開始前に準備
- ⑨単体テスト完了でユーザーに開示・・・プログラマーが側にいる間に完了を
◎プログラマーと会話ができる間にほとんどすべての修正を
- ⑩結合、総合、実運用試験のmin化・・・単体テストの徹底
- ⑪DBパトロール・・・データベース間の整合性チェック
- ⑫移行計画は早期準備・・・開発当初より計画し準備移行方式の多様化・安定性
- ⑬C/O日は定時帰宅を・・・no-trouble で、
- ⑭利活用が最重要・・・投資評価の実施、使いこなしが最重要
- ⑮ライフサイクルコスト・・・利用期間の長期化を配慮した総合コストでの判定

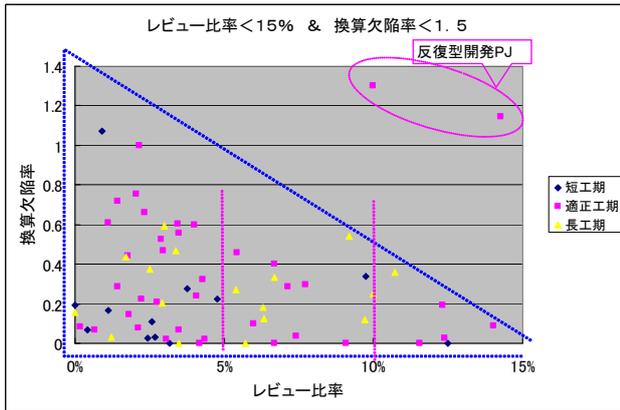
文章問題(エレベーター)

機能	要求番号	区分	要求機能
積載制限	SK01 (原則:ユーザーが書く)	要求機能	乗り込んだ客の重量を予測し次ぎの一人が乗ったら満員になると予測された場合は警告を発する
		理由	お客が乗り込んでから、降りてもらう不快感と時間のロスを避ける
	SK01-1 (SEが書く)	要求機能仕様	お客が乗り込んできたら、その都度時間をカウントし始めると同時に体重を積算し、(制限重量-最後の一人分の余裕)以上になった場合は「これが最後の方です」とアナウンスして伝える
		理由	乗ってから降りてもらう手間を省くため
	SK01-2	要求機能仕様	前の客が乗り込んでから3秒以上たっても次の客が乗車しなくては、制限重量以下であることを確認し、ドアを閉め、始動する
		理由	お客が少ない場合は、早めに発車するため

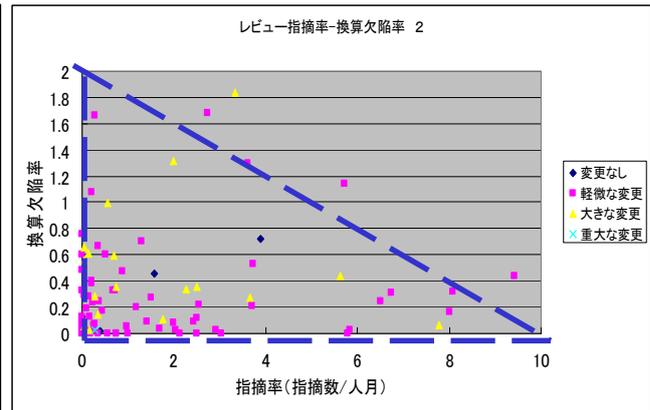
- ・要求機能・仕様を書き、その下に何故それが必要かの「理由」を書く
- ・仕様変更率=変更仕様数/総仕様数 これを一定の率に収める努力をする
- ・要求番号を、RFP、設計書、プログラムシート、変更管理にまで一貫して要求番号で参照できるようにする。

レビューと欠陥率

レビュー比率－換算欠陥率



レビュー指摘率－換算欠陥率



・散布図(欠陥率の尺度には換算欠陥率を使用)を書いてみるといずれも、**グラフの右上の範囲にデータが少なく**、レビュー比率またはレビュー指摘率の高いプロジェクトでは欠陥率の高いデータが少ないことをあらわしている。

レビュー比率>10%のエリアのところでは、目だって大きな換算欠陥率の点がプロットされておらず、逆に5%より少ないエリアでは大きな換算欠陥率の点が散見される

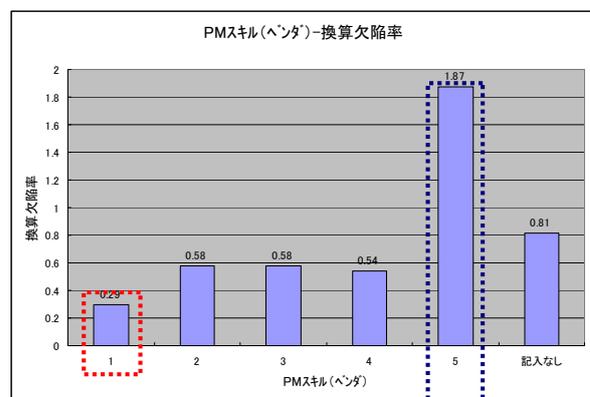
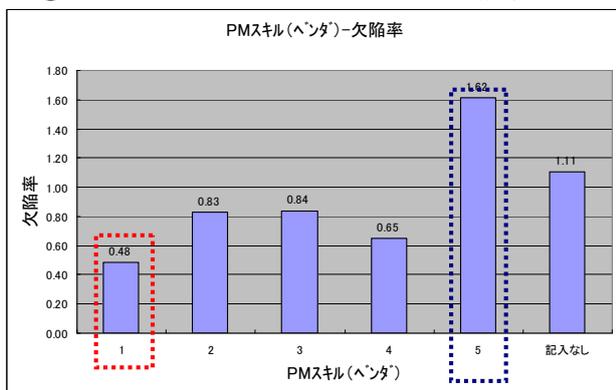
・設計時間の10%をかけてレビューすれば欠陥の多いプロジェクトは発生し難くなる



PMの能力と欠陥率(1)

PM(ベンダ、ユーザ)の能力と欠陥率の関係(能力が低いと出来上がり後のバグが多い)を確かめるために、PMスキル、PM業務精通度、PM技術精通度別に欠陥率、換算欠陥率を調べた。

①ベンダーPMスキルと欠陥率・換算欠陥率



- **ベンダPMスキルが低いと欠陥率も高い。**
- **ベンダPMスキル1が担当したプロジェクトは、平均の欠陥率が一番小さい。**
- **換算欠陥率で比べるとさらに顕著な差になって現れる。**

PMスキル

- 1.多数の中・大規模プロジェクトの管理を経験
- 2.少数の中・大規模プロジェクトの管理を経験
- 3.多数の小・中規模プロジェクトの管理を経験
- 4.少数の小・中規模プロジェクトの管理を経験
- 5.プロジェクト管理の経験なし

プロジェクトを成功させるためには、良いベンダーPMが必要

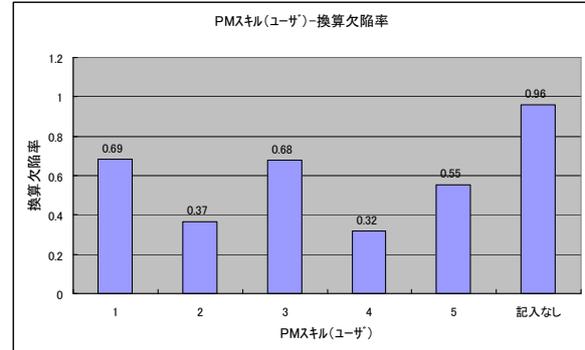
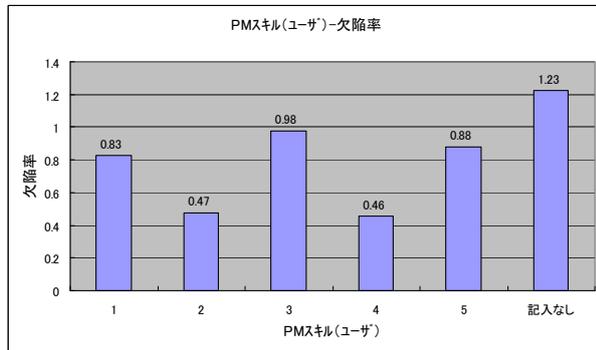


PMの能力と欠陥率(2)

②PM(ユーザ)スキルと欠陥率・換算欠陥率

PMスキル

- 1.多数の中・大規模プロジェクトの管理を経験
- 2.少数の中・大規模プロジェクトの管理を経験
- 3.多数の小・中規模プロジェクトの管理を経験
- 4.少数の小・中規模プロジェクトの管理を経験
- 5.プロジェクト管理の経験なし



- ユーザ側のPMの能力と品質に関しては、あきらかな傾向は見られない。

PMスキル、PM業務精通度、PM技術精通度全てにおいて全ベンダー側のPMの能力が高いと欠陥率が低い(品質が良い)が、

ユーザ側のPMの能力と品質に関しては、明らかな傾向は見られない。

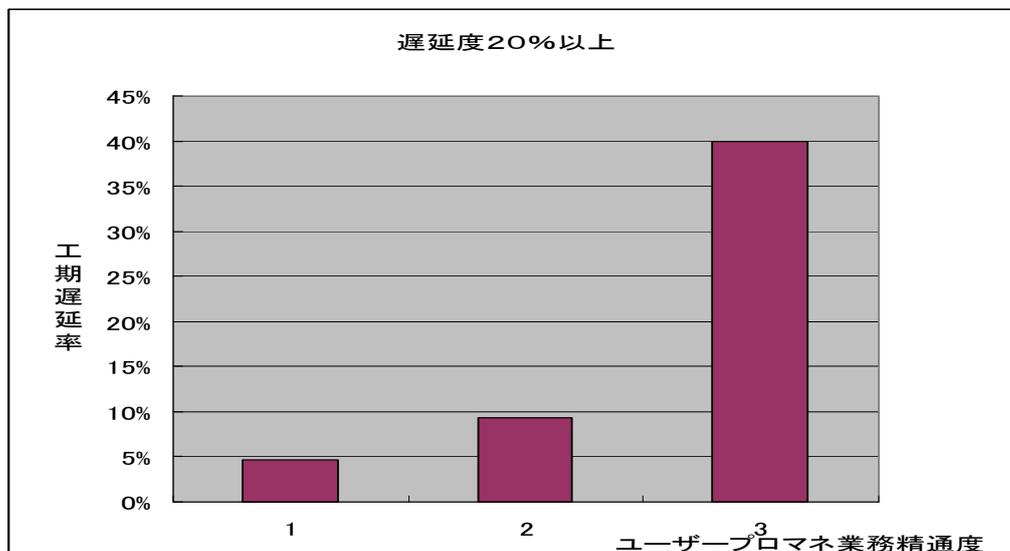
3年連続して同じ傾向が確認できた。



ユーザーPMの工期遅延への影響

ユーザーPMの業務精通度は工期遅延に影響する。

仕様決定のデシジョンができるユーザーPMを選ぶことが成功への第一歩



<業務精通度>

1. 十分精通していた
2. ある程度のレベルまでは精通していた
3. 精通していたとはいえない

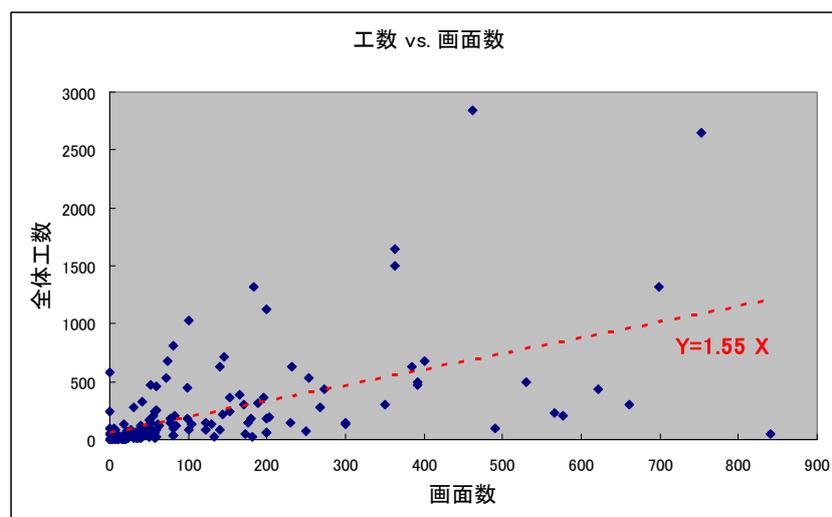


プロジェクトの成功に大きな影響を与える顧客キーマンの資質分析

	顧客キーマンの資質		
	高	中	低
システム開発のオーナー意識	高い	曖昧	無し
契約金額	必要と認めた分は承認	できるだけ安いが良い	値切りに値切る
システム要求仕様書の定義	完全	曖昧	皆無 後付けの智恵の仕様増加
ソフトウェア仕様書のチェック	約束納期厳守 緻密なレビュー	約束納期甘い 曖昧レビュー	約束納期守れず レビューは無し
開発規模	ほぼ予定通り	やや増加	大幅増加
仕様変更	殆どなし	頻発	多発
開発時の追加支払い	不要、あれば全額支払い	大幅な値切り	要求すれどゼロ回答
カットオーバー	計画通り又は早期化	納期確保に苦戦	納期遅れ常態化
粗利率	高い	計画より低下	低い、場合によっては赤字
運用時のトラブル	殆ど無し	時々	頻発
運用費用	低コスト	増加しがち	高コスト
キーマンの常用言	当社分の開発責任作業は 実行します。	できるだけ開発会社 が実行してください	当社ではITは判らないから 全て開発会社にお任せします
完了時の発言	又次の開発もお願いします	曖昧	次は別のベンダーと開発 したい
開発担当ベンダーSEの発言	又次の開発も下さい	(キーマンが変れば考えます)	2度と一緒に仕事したくない

画面数と工数の分布

工数と、最も高い相関を示した画面数との関連



- **工数(人月) = 1.55 x 画面数**
- サンプル数は163、相関係数は0.67(昨年度の相関係数は0.66)である。
- 昨年度の3変数回帰／画面と工数による回帰のサンプル数が99であったので、両回帰分析ともに、サンプルが増えても、回帰係数、相関係数ともに概ね同じ結果が得られたことになる。

ソフトウェアメトリックスの活用法

・ソフトウェアメトリックスは、全業界一律のデータであり、一般的には企業係数をかけて実際に使う必要がある。

(業種別に多数のデータが集まり分析できたとしても、企業係数、業務種類別係数は必要になる)

・したがって、JUASのソフトウェアメトリックス調査を参照しながら、自社用の業務種類別標準値を持つことが必要となる。

・システムが大規模になればなるほど、生産性は低下するが、10億円以上の、大規模システムは数が少ないので、精度アップは今後の課題である。

・品質と生産性の関係などなお見極めねばならない課題が山積である

・プロジェクトの数が少ない企業は、自社用のデータを集めることが難しい。

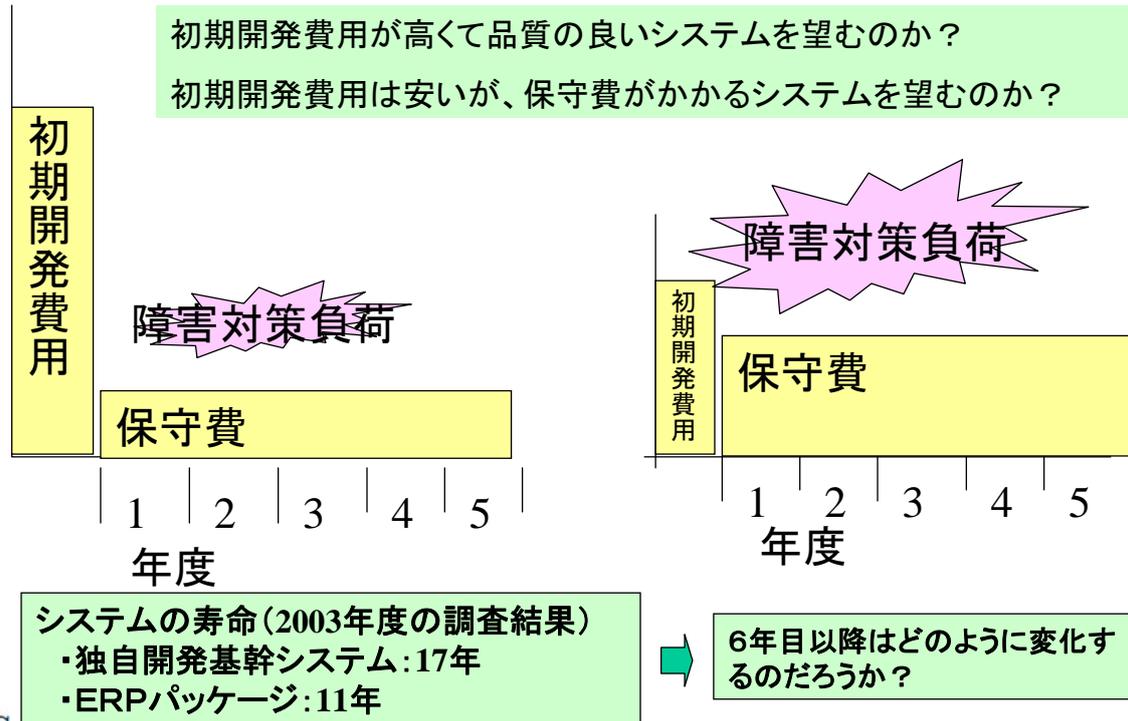
その場合には、仮目標を設定してから開発作業を実行すれば良い。

何も目標がない場合よりもはるかに効果がある。

5. システム保守についての分析

(JUASソフトウェアメトリックス調査2007)

(1-1) システムライフサイクルコスト システムライフサイクルコストの比較ポイント



(1-2) システムライフサイクルコスト カットオーバー後の開発費用・保守費用分析

バラつきが小さく実態をよく表していると思われる中央値を用いて、
自社開発・パッケージ開発それぞれの保守費用モデルを以下のように作成した。

保守費用分析 (中央値を採用)	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)	合計	本体保守(件数)	開発保守(件数)				
初年度総保守費用	10.00%	61	10.30%	42	20.30%	21.90%	16	13.00%	15
2年目総保守費用	10.90%	50	8.20%	34	19.10%	20.40%	10	11.50%	11
3年目総保守費用	8.90%	41	7.30%	27	16.20%	23.50%	9	14.40%	8
4年目総保守費用	13.60%	27	9.60%	14	23.20%	22.00%	7	8.40%	6
5年目総保守費用	13.60%	17	8.30%	9	21.90%	18.30%	4	3.80%	3
年間平均	11.40%		8.80%		20.10%	21.20%		10.20%	
初期開発費用	a					b		c	
合計費用比較	$a + a \times 0.2 \times 5 = a \times 2$			$b + b \times 0.21 \times 5$		$c + c \times 0.1 \times 5$			

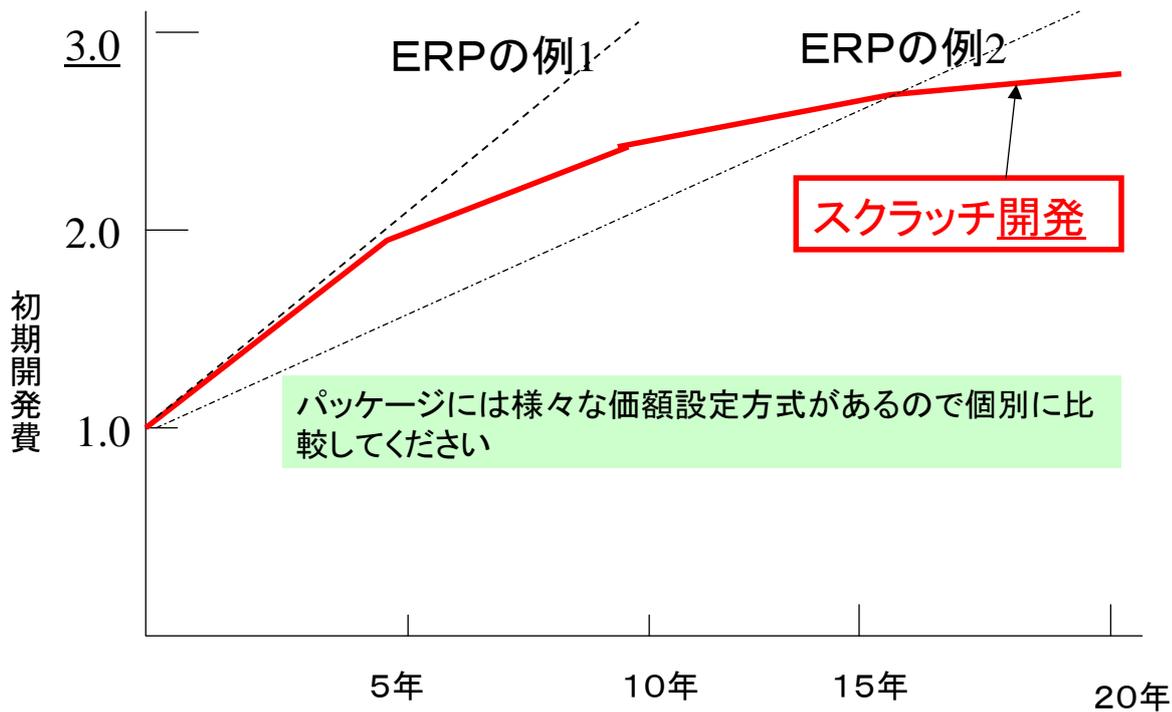
5年間で自社開発では初期開発費用の2倍の総費用がかかる。

パッケージ開発における保守費用比率としては、パッケージ本体の保守費用は単年で本体金額の20%前後、アドオン開発については、開発部分の10%前後が典型的な姿と、考えられる。
アドオン開発が増えればそれ以上に、ライフサイクルコストが増えることになる。

これらは参考値であるので、実際の適用はケースごとでの考慮が必要である。

B=0.8a、c=0.3a等の実際の数値をおいての比較が必要、係数のみで優劣比較をしないこと

(1-3) システムライフサイクルコスト 長期的システム費用

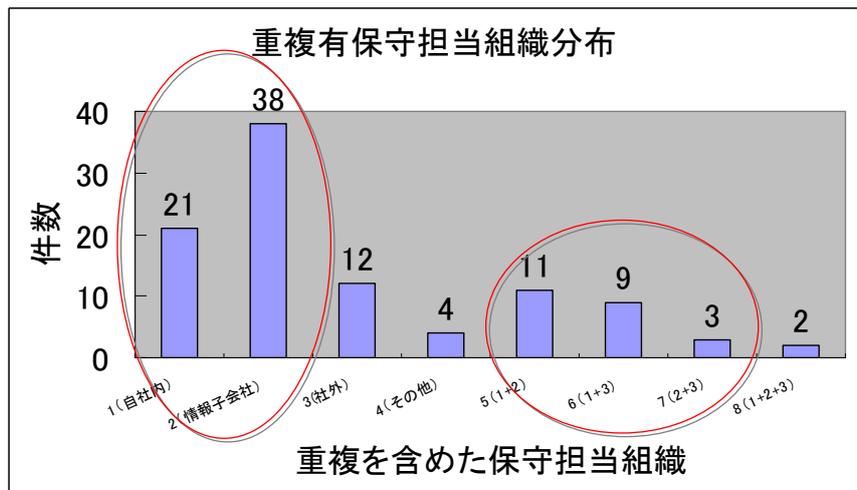


(2) 保守担当組織と専任・非専任

1) 保守担当組織

データ区間	頻度
1(自社内)	21
2(情報子会社)	38
3(社外)	12
4(その他)	4
5(1+2)	11
6(1+3)	9
7(2+3)	3
8(1+2+3)	2

	平均	N数
保守要員総数	6.7人	98
専任者割合	36%	96
非専任者割合	44%	96
社外要員割合	20%	96



自社あるいは情報子会社を中心に8割の保守作業は実施されている。システムは自社のものという意識が表れている。

(3-1) 保守範囲(非専任含む)

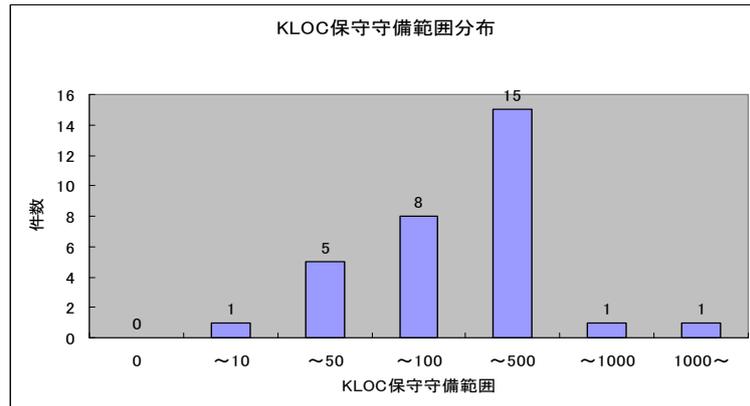
1) 要員1人当たりの保守守備範囲(FP値)

2) 要員1人当たりの保守守備範囲(KLOC値)

KLOC守備範囲	
平均	218.5847044
標準誤差	55.21826053
中央値(メジアン)	100.7428571
最頻値(モード)	#N/A
標準偏差	378.5573218
分散	143305.6459
尖度	29.58420175
歪度	4.995185475
範囲	2499.965857
最小	0.034142857
最大	2500
合計	10273.48111
標本数	47
信頼区間(95.0%)	111.1484873

FP保守守備範囲	
平均	4586.151119
標準誤差	3426.267223
中央値(メジアン)	848.5
最頻値(モード)	#N/A
標準偏差	17470.60343
分散	305221984.2
尖度	25.6692063
歪度	5.052985616
範囲	89998.94817
最小	1.051829268
最大	90000
合計	119239.9291
標本数	26
信頼区間(95.0%)	7056.524494

- ・中央値は約100KLOC/人
850FP/人であるが、バラつきが大きい
- ・1000KLOC/人以上の極端に範囲が広いプロジェクトも存在する。



(3-2) 保守範囲(専任のみ)

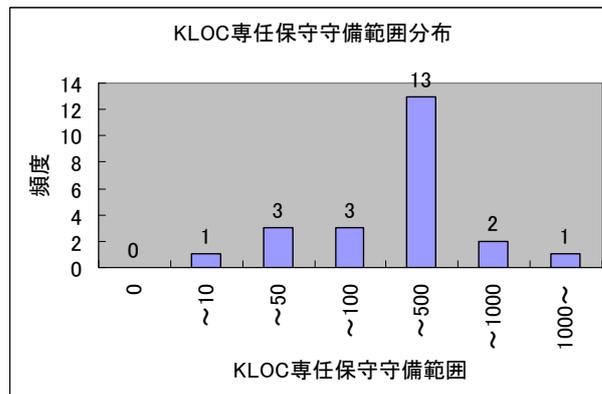
3) 専任保守要員1人あたりのKLOC値

KLOC専任保守守備範囲	
平均	268.4870865
標準誤差	62.59110802
中央値(メジアン)	195.738
最頻値(モード)	#N/A
標準偏差	300.1764089
分散	90105.87647
尖度	5.893429696
歪度	2.21305823
範囲	1306.626833
最小	0.039833333
最大	1306.666667
合計	6175.202989
標本数	23
信頼区間(95.0%)	129.8061525

データ区間	頻度	割合
0	0	0%
~10	1	4%
~50	3	13%
~100	3	13%
~500	13	57%
~1000	2	9%
1000~	1	4%

中央値は約200KLOC/人
バラつきが大きい。

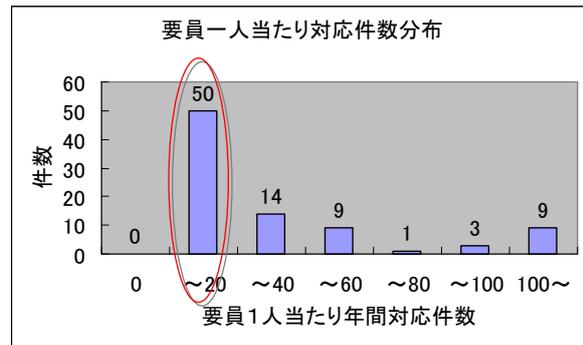
極端に守備範囲が広いプロジェクトもある。該当システムが環境変化をどの程度吸収する必要があるかで守備範囲が決定される。



(4-1)保守作業の引き受け形態と年間対応件数/人

保守作業引受形態

引受形態	割合
契約要員数	11%
対応工数の範囲	38%
内容判断による	45%
その他	6%



システムごとの年間依頼件数は50件(中央値)、その要求に対して保守対応率は平均86%、すべて保守しているシステムも60%ある。

依頼対応できなかった理由	
理由(n=46)	平均値
不要判断	24.4
人手不足	12.3
経済的理由	19.1
即時対応不要	32.1
工期不足	4.2
スキル不足	1.7
その他	6.2

保守組織	平均	中央値	最小	最大	標本数
全体	37.0	14.4	0.5	286.0	86
自社内	43.3	20.6	0.9	210.0	19
情報子会社	23.5	13.3	0.5	105.0	36
社外	86.0	34.7	1.0	286.0	8
その他	52.7	50.0	3.3	104.7	3
自社+情子	21.4	6.6	0.5	138.2	10
自社+社外	66.3	32.5	4.8	231.0	6
情子+社外	15.6	18.2	5.0	23.5	3
自社+情子+社外	6.7	6.7	6.7	6.7	1

対応件数は14件/人年前後である。しかし、システム内容や保守組織によってばらつきは大きい。中には1人で100件以上の案件をこなす場合も見られる。

業務内容、対応の緊急性、欠陥発生時の影響範囲、品質、予算などによって対応方法を選択することになる

(4-2)年間対応件数/人

* 保守環境整備との関連

	平均	中央値	最小	最大	標本数
要員一人当たり対応数 24hテスト可能	47.0	14.3	1.3	286.0	27
要員一人当たり対応数 柔軟にテスト可能	28.5	12.0	0.5	231.0	41
要員一人当たり対応数 テスト時間に制約	41.2	34.6	0.5	138.2	18

本番環境と同じ環境で保守作業できるかどうか、保守作業時間の状況で差が見られるか調べてみた。品質重視で24時間テスト可能システムは、対応件数が低下する可能性がある。

是正保守とは問題を訂正する保守のこと

適応保守とは環境変化に適合させる保守のこと

完全化保守とは性能や保守性を改善する保守のこと

5) 保守作業割合との関連

	平均	中央値	最小	最大	標本数
保守の問合せ	32.9%	30.0%	0.0%	100.0%	92
保守の基盤整備	8.8%	5.0%	0.0%	50.0%	92
是正保守	23.7%	14.0%	0.0%	100.0%	92
適応保守	23.7%	10.0%	0.0%	90.0%	92
完全化保守	10.8%	1.0%	0.0%	100.0%	92

保守作業割合の調査結果を見ると、保守作業において「保守の問合せ」に占める割合が圧倒的に大きい。是正保守含めると約6割の負担になる。

この作業項目が年間対応件数に大きく影響するものと考えられる。

(5-1)保守作業発生理由

回答のあった97件のデータで、各項目別の保守作業割合の統計を算出

保守理由	平均	中央値	最小	最大	標本数
保守理由 システムバグ	19.8%	10.0%	0.0%	100.0%	97
保守理由 制度ルール変化	17.3%	10.0%	0.0%	95.0%	97
保守理由 業務方法変化	18.0%	10.0%	0.0%	70.0%	97
保守理由 経営目標変化	4.9%	0.0%	0.0%	70.0%	97
保守理由 ユーザビリティ変化	7.8%	5.0%	0.0%	50.0%	97
保守理由 担当者要望	20.1%	10.0%	0.0%	90.0%	97
保守理由 その他	12.2%	0.0%	0.0%	87.0%	97

担当者要望、システムバグ、制度ルールの変化、業務方法の変化 の4項目が保守作業の主たる理由

(5-2)保守作業割合

作業負荷の割合は簡単な作業が1/3

作業負荷分類	割合(%)
保守作業半日以内	31.5
保守作業1日以内	17.9
保守作業3日以内	16.0
保守作業1週間以内	12.8
保守作業1ヶ月以内	12.8
保守作業1ヶ月以上	9.1

(5-3)保守作業フェーズ別負荷

作業フェーズの割合は3:3:3:1

	割合
修正箇所の調査	29.7
修正作業	30.4
テスト確認	28.7
ドキュメント修正	11.2

(6) 業種別保守作業分析

業種ごとの特性、管理思想の差が表れている

保守作業負荷割合区分	全体(n=92)		金融(n=10)		製造(n=63)		サービス(n=19)	
	平均値	中央値	平均値	中央値	平均値	中央値	平均値	中央値
半日以下	31.5%	22.5%	10.5	2.5	30.4	20.0	46.2	50.0
1日以内	17.9	10.0	5.0	5.0	18.6	10.0	22.3	17.0
3日以内	16.0	10.0	16.1	15.0	17.7	10.0	10.3	10.0
1週間以内	12.8	7.0	19.0	15.0	12.7	10.0	9.8	5.0
1ヶ月以内	12.8	6.5	17.4	10.0	13.9	8.0	7.0	1.0
1ヶ月以上	9.1	0	32.0	10.0	6.8	0	4.4	0

保守理由	金融(n=10)		製造(n=66)		サービス(n=21)	
	平均値	中央値	平均値	中央値	平均値	中央値
システムバグ	12.9	11.5	19.4	10.0	24.0	10.0
制度ルールの変化	19.0	10.0	15.5	10.0	22.0	15.0
業務方法の変化	19.2	10.0	18.1	12.5	17.1	15.0
経営目標変化	13.5	5.0	4.5	0	2.0	0
ユーザビリティ変化	9.6	5.5	8.4	5.0	5.4	0
担当者要望	18.7	15.0	20.4	14.0	19.5	5.0
その他	7.1	5.0	13.6	0	9.9	0

(7) 保守品質

保守作業品質基準として、初年度保守欠陥率、2年目以降保守欠陥率、受入確認即時合格率を調査した。

保守の品質とは？

➤初年度保守欠陥率：

保守初年度に本番に組み込み運用開始後に発生した欠陥数/年間保守件数

➤2年目以降保守欠陥率：

保守2年目以降に本番に組み込み運用開始後に発生した欠陥数/年間保守件数

➤受入確認即時合格率：保守作業後の受入確認が1回でOKの割合

	平均	中央値	最小	最大	標本数
初年度保守欠陥率	18.3%	5.0%	0.0%	100.0%	68
2年目以降保守欠陥率	9.0%	5.0%	0.0%	100.0%	56
受入確認即時合格率	60.9%	89.5%	0.0%	100.0%	54

初年度保守欠陥率は平均約18%、中央値5%という結果であった。

1年以内に保守した案件は、リリース後10件中1.8件は欠陥が出ているという実態がうかがえる。

カットオーバー当初はバグが多く、落ち着けば安定するという予想された結果が得られているが、それでも9%の保守欠陥率になっている。

「保守作業が完了しました」とSEが言ってきたシステムの40%は再修正作業が発生する。保守仕様の不十分さ、テストの不十分さなどが原因であるが、保守作業の難しさが伺われる。



(8) 保守作業の目標とドキュメント修正度

保守作業の品質目標

区分	割合
品質目標あり	38%
品質目標なし	61%

品質目標が無いシステムが60%であり、PDCAが回っているとはいえない

ドキュメントの修正程度

修正区分	割合
①完全に修正	21%
②ほぼ完全に修正	33%
③一部不完全	34%
④不十分な修正になっている	9%
⑤ほとんど修正しない	1%

ドキュメントを正確、確実に修正するのは難しい忍耐力を必要とする作業である。

更なるツールの利用など工夫の余地が残されている



(9) 保守の工期

	平均	中央値
納期遅延率	8.6%	5.0%

- 納期遅延率は平均9%(中央値5%)
- 遅延理由は、保守担当者から見ると他責要因が多い(①③⑤で1位選択の84%を占める)

	1位選択	2位選択	3位選択	データ数	順位得点	平均順位点
①他の作業が割り込んだ	30	11	6	47	118	2.51
②工数見積もりが甘かった	6	8	9	23	43	1.87
③保守仕様の変更があった	14	18	4	36	82	2.28
④作業中にミスが多発した	3	1	2	6	13	2.17
⑤潜在バグの影響	3	10	7	20	36	1.80
⑥その他	0	1	8	9	10	1.11
選択肢合計	56	49	36	141		

(10) 保守作業の見積

見積者	割合
①チーム内見積者	49%
②チーム外見積者	0%
③担当者が見積る	46%
④その他	4%

見積基準	割合	納期遅延率平均
あり	40%	8.3%
なし	60%	8.6%

保守者 一人当たり対応数/年	平均値	中央値	N数
①チーム内見積者	47.7	21.5	42
②チーム担当者の見積	25.7	12.2	44

見積者は保守チーム内の見積者と担当者任せが二分する。

①チーム内の見積専任者が見積する場合の方が生産性は高そう。

見積担当者有無と欠陥率の関係	見積担当者あり		見積担当者なし	
	平均値	中央値	平均値	中央値
初年度保守欠陥率	13.9	2.2	22.7	8.7
2年目以降保守欠陥率	7.6	1.0	10.3	5.0
受入時の保守欠陥率	43.7	10.0	34.7	13.0

規模、内容にもよるが、見積専任者がいるほうが、保守作業要件、条件が検討・整理されて、手戻りなどが少ないため生産性・品質が向上する。

見積責任者兼保守作業管理者の設置は重要である

(11) 保守作業見積方法

見積方法	割合(複数回答)
①帳票・画面の修正内容	18%
②ロジック変更の修正内容	23%
③DB変更の修正内容	13%
④DB項目追加の修正内容	16%
⑤その他の修正内容	5%
⑥範囲から負荷予測:巻き込み含む	18%
⑦範囲から負荷予測:巻き込み含まず	3%
⑧リスク要員から負荷予測	18%
⑨WBSから負荷予測	10%
⑩その他	2%

保守作業の見積方法は多様。
WBSの作成方法含めてエンジニアリング手法を持ち込む必要がある

(12) 保守可能時間と品質

	24時間テスト可能 割合は29%		柔軟にテスト可能 割合は51%		テスト時間に制約 あり割合は20%	
	平均値	中央値	平均値	中央値	平均値	中央値
一人当たりの対応数	47.1	14.3	28.5	12.0	41.2	34.6
初年度保守欠陥率	18.3	5.0	17.7	5.0	21.1	20.0
2年目以降保守欠陥率	9.0	2.5	9.0	3.8	8.8	5.0
受入時不合格率	44.7	20.0	42.5	15.0	10.0	10.0

テスト時間に制約がある保守作業は、運用に持ち込んだ最終品質が低下

(13) 保守生産性向上の仕組み

保守負荷低減の仕組み

	割合
あり	47%
なし	53%

保守負荷低減の方法

	割合
保守作業を考慮しドキュメントを作成する	31%
既存のテスト環境の整備を十分に行い維持する	22%
開発時に保守用調査ツールをあわせて作成する	13%
ドキュメントソースを特定するための解析容易な仕組みを取り入れる	13%
別環境に移植したときの環境適合に関する配慮を行う	5%
開発母体の潜在バグを徹底的に摘出し品質を高める	2%
その他	5%

保守担当者の開発への参画

	割合
①開発要員の移行	71%
②開発レビューへの参加	15%
③開発ドキュメントの査閲	5%
④その他	7%

保守作業の生産性向上は開発チーム内の要員が保守チームへシフトすることによって確保するケースが一番多い。今後保守作業を重視して対策を強化する必要がある。

保守作業の効率化はドキュメントの改善と環境の整備が重点である



(14) 保守環境

テストツール

使用している	23%
使用していない	76%



ツールの種類(複合回答)

テスト結果の比較	65%
テスト手順再現	22%
データの整合性チェック	13%
テストケース生成	0%
その他	4%



(15) 開発から保守への引継ぎとユーザー満足度

引継ぎ条件(引継ぎ基準ありの割合)

時間	8%
方法	20%
資料	36%

保守容易性確保のガイドライン

あり	16%
なし	84%

ユーザー満足度

非常に良い	5%
良い	49%
普通	39%
やや悪い	5%
非常に悪い	0%

概ね良好な関係であると保守責任者は考えている。
開発時の制約を守るための開発SEと日々の利用者の業務活動を支援する保守SEの性格の差が表れている

(16) 保守担当専任組織、専任者、教育体系

保守作業専門組織あり	51%
保守作業専門組織なし	49%

保守専任担当者あり	59%
保守専任担当者なし	41%

保守専任要員の教育体系あり	14%
保守専任要員の教育体系なし	82%

保守作業の教育は何をすれば良いのか？
大半は悩み中であるが、すでに対策をとり始めている企業も14%ある。

まとめ

①時代の流れ

* 新機能競争→価額競争→品質競争(安定化)

②プロセス志向から「プロダクト志向あつてのプロセス志向」

* 見える化の重要性

③従来の方法では、ユーザーは満足させることができない

→改善、改革への発想の転換→新技術へも挑戦

④問題があるところに、改善の種があり。

改善を考えるとところに創造の喜びあり

* 従来の開発・保守・運用方法は喜びの種がたくさんある

ご清聴ありがとうございました

■本資料で使用している調査データ
・企業IT動向調査2005、2006、2007
・ソフトウェアメトリックス調査2006、2007
<http://www.juas.or.jp/project/survey/index.html>

■本資料に関するお問い合わせ
(社)日本情報システム・ユーザー協会(JUAS)
担当:佐藤(03-3249-4101/info@juas.or.jp)