

平成21年度
ソフトウェア開発に関する調査票(受託者向け)
集計結果その1

平成22年6月

財団法人経済調査会

調査票集計結果について

本集計結果は、本調査にご協力いただいた各企業にお送りしているものです。

本調査は、日本ファンクションポイントユーザ会FP法利用検討会（JFPUG／FPSMSG）および国立大学法人奈良先端科学技術大学院大学との共同調査として財団法人経済調査会が実施しとりまとめました。

調査対象は、日本ファンクションポイントユーザ会（JFPUG）に登録されている企業を中心とした全国の情報処理サービス企業です。

本調査は、受託者（受注企業）への調査を想定しましたが、一部委託者（発注企業）の回答が含まれております。

平成22年6月

財団法人経済調査会

1 調査期間

平成22年1月～平成22年2月

2 調査票の構成と概要

調査票Ⅰ	調査先組織の概要に関する調査
調査票Ⅱ	調査先組織におけるソフトウェア開発規模計測手法と生産性等に関する調査
調査票Ⅲ	調査先組織における個別のソフトウェア開発事例に関する調査 同一組織の複数回答あり

3 調査票配布状況

配布数	588
-----	-----

4 調査票回収状況

調査票Ⅰ	85
調査票Ⅱ	80
調査票Ⅲ	134

5 集計結果の構成と条件

- 調査票Ⅰ・・・調査先組織の概要に関する集計結果

【前提条件】

回収された85社の調査票Ⅰを対象として主要な調査項目の集計を行いました。
集計結果については、外れ値の排除は行っておりません。

- 調査票Ⅱ・・・ソフトウェア開発規模計測手法と生産性に関する集計結果

【前提条件】

回収された80社の調査票Ⅱを対象として主要な調査項目の集計を行いました。
集計結果については、外れ値の排除は行っておりません。

- 調査票Ⅲ・・・個別のソフトウェア開発事例に関する集計結果

【前提条件】

回収された134件の調査票Ⅲを対象として主要な調査項目の集計を行いました。
集計結果については、外れ値の排除は行っておりません。

目 次

	ページ
1 調査票Ⅰ 組織の概要に関する集計結果	1
[1] 組織の立場	1
[2] 組織の企業規模	1
[3] 組織の所定労働日数、所定労働時間	2
2 調査票Ⅱ ソフトウェア開発規模計測手法と生産性等に関する集計結果	3
[1] 組織の代表的なソフトウェア開発規模計測手法	3
[2] ファンクションポイント法の具体的な手法	4
[3] I F P U G法またはI F P U G法をベースとした機能規模計測法の計測時間等	4
[4] 組織の代表的な開発規模	5
[5] 組織の平均的な開発言語別生産性	5
[6] 組織のシステム開発プロセスの標準化	19
[7] システム開発プロセスを標準化していない理由	19
[8] システム開発プロセス標準と共通フレームとのマッピング状況	20
[9] システム開発プロセス標準と共通フレームをマッピングしていない理由	20
[10] 組織のCMM	21
[11] 組織のオフショア開発の実績	22
3 調査票Ⅲ 個別のソフトウェア開発事例に関する集計結果	25
[1] 開発事例の新規開発案件、改造案件区分	25
[2] 開発事例の同異区分	25
[3] 開発事例の委託者の分類	26
[4] 開発事例の適用分野・業種	26
[5] 開発事例の開発工程範囲	28
[6] 開発事例のシステム構成	29
[7] 開発事例のOS	29
[8] 開発事例のデータベース	30
[9] 開発事例の開発言語または開発ツール	31
[10] 開発事例の開発方法論	32
[11] 開発事例のフレームワーク使用状況	34
[12] 開発事例のパッケージソフトウェア使用状況	35
[13] 開発事例のソフトウェア費用	35
[14] 開発事例のソフトウェア開発期間	36
[15] 開発事例のソフトウェア開発工数と規模	37
[16] 開発事例の画面数・帳票数・ファイル数・バッチプログラム数	45
[17] 開発事例のユースケース数・アクター数	46
[18] 開発事例の品質管理	47
[19] 開発事例の最大開発要員	48
[20] 開発事例のリスク管理	50

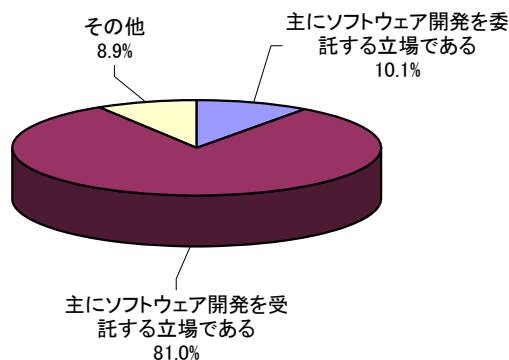
目 次

	ページ
[21] 開発事例の生産性変動要因	53
[22] 開発事例のソフトウェア開発工程別工数比率と職種別参画比率	54
[23] 開発事例のソフトウェア開発工程別設計書文書量	56
[24] 開発事例の評価	57
4 付属資料 ソフトウェア開発に関する調査票（受託者向け）	61

1 調査票Ⅰ 組織の概要に関する集計結果

[1] 組織の立場

選択肢	組織数	比率
主にソフトウェア開発を委託する立場である	8	10.1%
主にソフトウェア開発を受託する立場である	64	81.0%
その他	7	8.9%
合計	79	100.0%



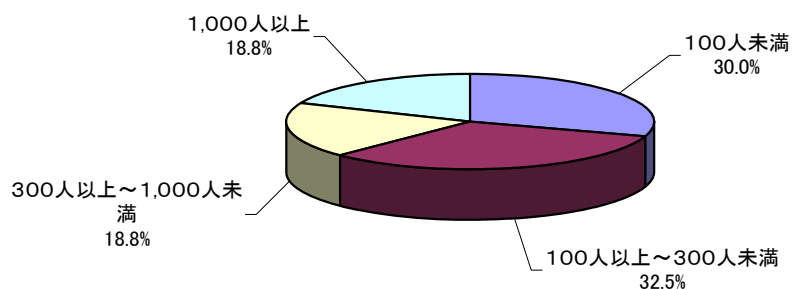
[2] 組織の企業規模

(1) 従業員数 (単位：人)

組織数	平均値	中央値	最大値	最小値
81	1,795	460	33,520	13

(2) ソフトウェア開発技術者数

選択肢	組織数	比率
100人未満	24	30.0%
100人以上～300人未満	26	32.5%
300人以上～1,000人未満	15	18.8%
1,000人以上	15	18.8%
合計	80	100.0%



[3] 組織の所定労働日数、所定労働時間

(1) 所定労働日数 (単位：日／月)

組織数	平均値	中央値	最大値	最小値
80	20.3	20	23	20

(2) 所定労働時間 (単位：時間／月)

組織数	平均値	中央値	最大値	最小値
80	157.1	157	180	140

2 調査票Ⅱ ソフトウェア開発規模計測手法と生産性等に関する集計結果

[1] 組織の代表的なソフトウェア開発規模計測手法（使用頻度=1が記入された項目の集計）

(1) ソフトウェア開発規模計測手法

選択肢	組織数	比率
ファンクションポイント法（簡易的な適用を含む）	22	27.5%
ステップ換算法	40	50.0%
その他	18	22.5%
合計	80	100.0%

(2) その他の具体的な開発規模計測手法

過去の経験。
経験則により、機能に細分化した規模に対し、特定の言語生産性（当社内で定めたもので、たとえばVBの場合、設計～テストまでの日/ksの基準生産性を使用）により計測する。（工程別にもっていないので、あえてステップ換算法とは別にした）
類推法、もしくはボトムアップ積上げ法。
過去の類似システムから予測。
開発規模が算定できない作業について、成果物に着目した規模見積を実施。
画面数、帳票数、（プログラム）モジュール数。
全社共通指標はステップ換算。継続契約しているもの、特定のアプリケーションドメインで画面数等を利用。
経験に基づく類似比較法、WBS積上げ法。
仕様追加・仕様変更する際に、過去の実績工数を基に、開発工数を見積もる。
パッケージにおける画面入力項目数、画面数、印刷帳票数、ファイル本数。
類推法、旧システムの保守要員工数から推測、機能別工数積上げ。
IPD：TPF系の見積標準に準拠した開発実施局面生産量をベースとし、実施全局面に拡張換算された開発生産量の総和（IPD数）をその開発に費やしたお客さま請求工数で除した値。
WBS法。
作業単位で必要な項目、工数、金額を見積り、全体を積上げる。
画面、帳票、DB数によりファンクションポイント値を概算するツールを使用。
直接工数算出法。
類似開発を参考。
機能ごとに画面数やファイル数をもとに過去の開発経験から工数を算出する。
I/Oからの工数見積もり。
開発工程ごとの作業時間（工数）。
設定パラメタ数など。
過去の類似プロジェクト実績を元にした工数積上による。
ファンクションスケール法（富士通独自の機能規模計測法）。
人月換算（作業工数で開発規模を代替的に表現）。
画面、帳票、バッチ数。
小規模開発の場合は、管理工数をファンクションポイントとは切り離して別途に稼動見込みで算出。稼動見込みは顧客の体質などを考慮して過去経験から導出。
開発規模計測を目的としたものではありませんが、以下を実施しています。画面、帳票、プロセスそれぞれについて複雑度を5段階に分類し、基準となる工数を算出する。
過去の類似案件の実績工数からの算出。
社内で標準化された工数換算表を利用
ステップ換算の簡易手法として、大・中・小分類による規模評価。
類推・類似法、積上げ法、テストケース数換算法など。
プログラム本数（exeファイルの数）。

[2] ファンクションポイント法の具体的な手法(複数選択あり)

選択肢	組織数
I F P U G 法	32
N E S M A 概算法	22
N E S M A 試算法	4
自社オリジナル機能規模計測法	22
その他	10
合計	90

[3] I F P U G 法または I F P U G 法をベースとした機能規模計測法の計測時間等

(1) 改造システムの具体的な計測手法(複数選択あり)

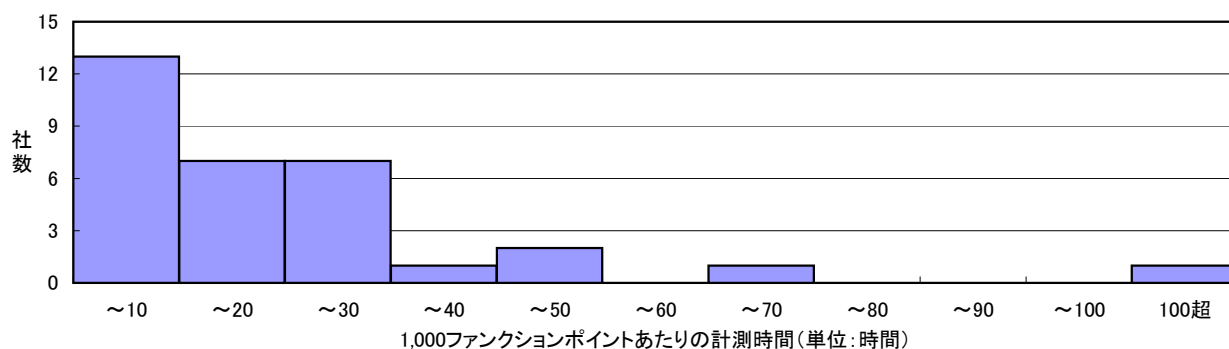
選択肢	組織数
E F P (Enhancement FP count)	11
E F P (Enhancement FP count) をベースとした考え方 (VAFを考慮しないなど)	11
A F P (Application FP count)	8
その他	11
合計	41

(2) 主な計測者

選択肢	組織数	比率
当該システムの関係者	46	90.2%
当該システムの関係者以外の専任の担当者	3	5.9%
その他	2	3.9%
合計	51	100.0%

(3) 1,000 FPあたりの計測時間 (単位: 時間)

組織数	平均値	中央値	最大値	最小値
32	22.5	16	140	1



[4] 組織の代表的な開発規模

(1) ソフトウェア開発工数

① ソフトウェア開発工数 (FROM) (単位：人月)

組織数	平均値	中央値	最大値	最小値
63	19.7	10	250	0.1

② ソフトウェア開発工数 (TO) (単位：人月)

組織数	平均値	中央値	最大値	最小値
62	286	100	2,755	1

(2) ソフトウェア開発規模

① ソフトウェア開発規模 (FROM) (単位：FP)

組織数	平均値	中央値	最大値	最小値
38	219.7	120	1,500	0.4

② ソフトウェア開発規模 (TO) (単位：FP)

組織数	平均値	中央値	最大値	最小値
37	3,636	2,000	20,000	31.5

[5] 組織の平均的な開発言語別生産性

回答数が10社以上の言語については、各社の「FROM値」と「TO値」を縦線に表示し、「FROM値」と「TO値」の平均値を昇順に並べたグラフを作成した。

(1) アセンブラの生産性

① アセンブラの生産性 (FROM) (単位：FP/人月)

組織数	平均値	中央値	最大値	最小値
1	6.8	6.8	6.8	6.8

② アセンブラの生産性 (TO) (単位：FP/人月)

組織数	平均値	中央値	最大値	最小値
1	21.3	21.3	21.3	21.3

③ アセンブラの生産性 (FROM) (単位：STEP/人月)

組織数	平均値	中央値	最大値	最小値
5	680	600	1,000	400

④ アセンブラの生産性 (TO) (単位：STEP/人月)

組織数	平均値	中央値	最大値	最小値
5	1,160	1,200	2,000	600

(2) A S Pの生産性

① A S Pの生産性 (FROM) (単位：FP/人月)

組織数	平均値	中央値	最大値	最小値
5	4.7	5	6.7	1.1

② A S Pの生産性 (T0) (単位：FP/人月)

組織数	平均値	中央値	最大値	最小値
3	14.6	7	30.9	6

③ A S Pの生産性 (FROM) (単位：STEP/人月)

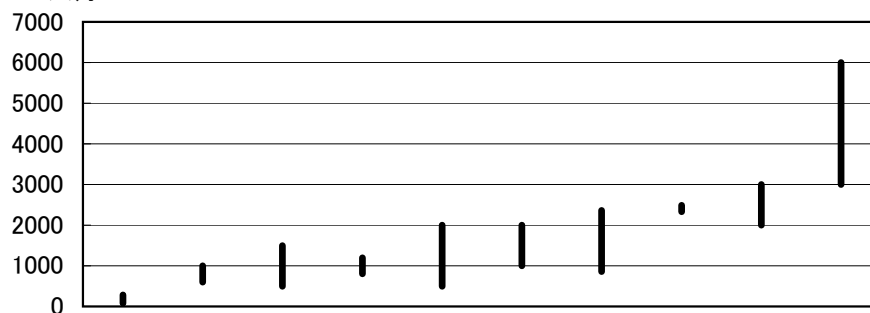
組織数	平均値	中央値	最大値	最小値
10	1,167	830	3,000	75

④ A S Pの生産性 (T0) (単位：STEP/人月)

組織数	平均値	中央値	最大値	最小値
10	2,184	2,000	6,000	287

ASPの生産性(STEP/人月)

STEP/人月



(3) ASP. NETの生産性

① ASP. NETの生産性 (FROM) (単位：FP/人月)

組織数	平均値	中央値	最大値	最小値
9	9	6.7	20	2

② ASP. NETの生産性 (TO) (単位：FP/人月)

組織数	平均値	中央値	最大値	最小値
9	22.7	20	64	4

③ ASP. NETの生産性 (FROM) (単位：STEP/人月)

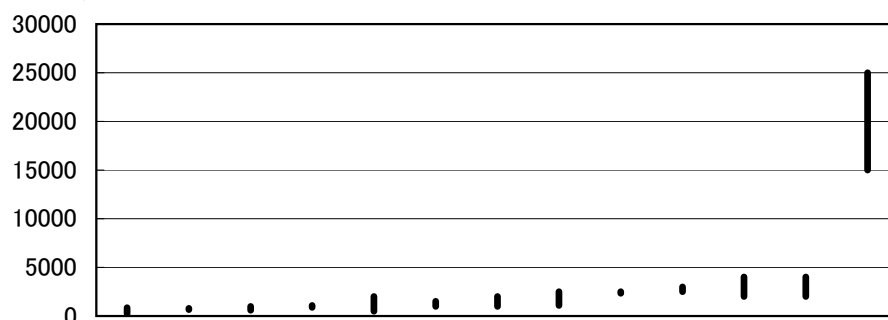
組織数	平均値	中央値	最大値	最小値
13	2,293	1,000	15,000	225

④ ASP. NETの生産性 (TO) (単位：STEP/人月)

組織数	平均値	中央値	最大値	最小値
13	3,865	2,000	25,000	800

ASP. NETの生産性(STEP/人月)

STEP/人月



(4) Cの生産性

① Cの生産性 (FROM) (単位：FP/人月)

組織数	平均値	中央値	最大値	最小値
4	12.6	13.6	20	3

② Cの生産性 (TO) (単位：FP/人月)

組織数	平均値	中央値	最大値	最小値
4	30.1	30.7	50	9

③ Cの生産性 (FROM) (単位：STEP/人月)

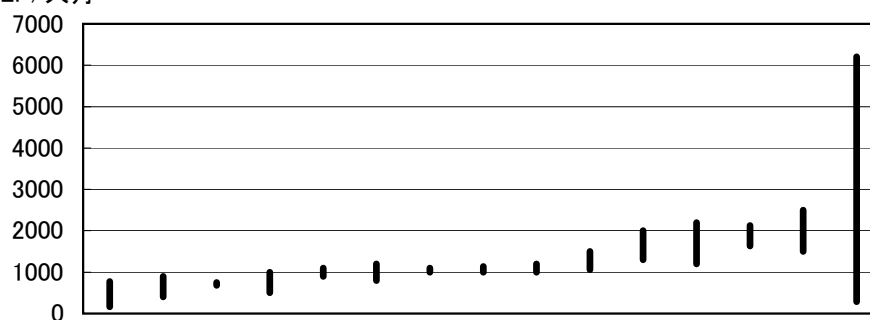
組織数	平均値	中央値	最大値	最小値
15	894.7	1,000	1,630	160

④ Cの生産性 (TO) (単位：STEP/人月)

組織数	平均値	中央値	最大値	最小値
15	1,713	1,200	6,200	750

C生産性(STEP/人月)

STEP/人月



(5) C++ (VC++を含む) の生産性

① C++の生産性 (FROM) (単位: FP/人月)

組織数	平均値	中央値	最大値	最小値
3	12.2	6.7	25	5

② C++の生産性 (TO) (単位: FP/人月)

組織数	平均値	中央値	最大値	最小値
4	34.1	28.1	70	10

③ C++の生産性 (FROM) (単位: STEP/人月)

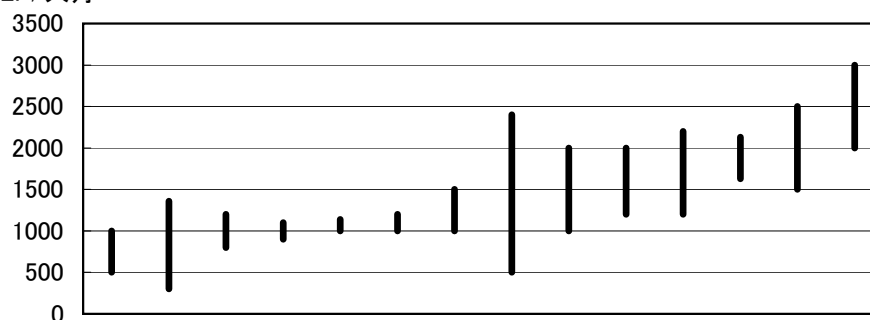
組織数	平均値	中央値	最大値	最小値
14	1,038	1,000	2,000	300

④ C++の生産性 (TO) (単位: STEP/人月)

組織数	平均値	中央値	最大値	最小値
15	1,882	2,000	3,500	1,000

C++(VC++を含む)生産性(STEP/人月)

STEP/人月



(6) COBOLの生産性

① COBOLの生産性 (FROM) (単位: FP/人月)

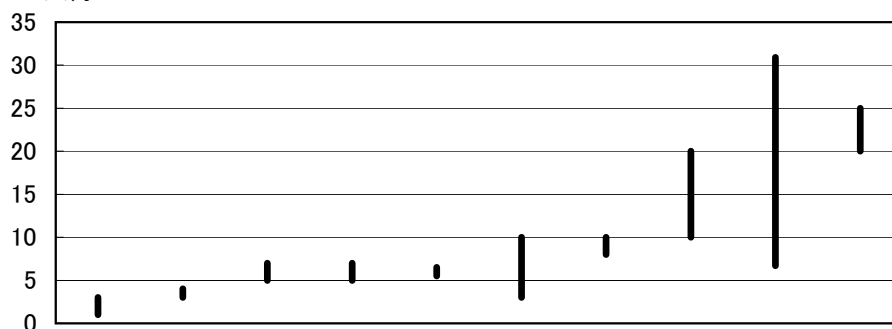
組織数	平均値	中央値	最大値	最小値
10	6.7	5.3	20	1

② COBOLの生産性 (TO) (単位: FP/人月)

組織数	平均値	中央値	最大値	最小値
10	12.3	8.5	30.9	3

COBOL生産性(FP/人月)

FP/人月



③ COBOLの生産性 (FROM) (単位: STEP/人月)

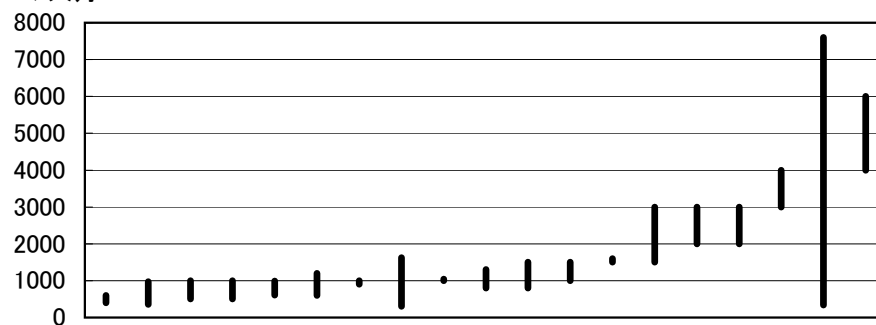
組織数	平均値	中央値	最大値	最小値
20	1,305	850	4,000	300

④ COBOLの生産性 (TO) (単位: STEP/人月)

組織数	平均値	中央値	最大値	最小値
21	2,330	1,500	7,600	600

COBOL生産性(STEP/人月)

STEP/人月



(7) HTMLの生産性

① HTMLの生産性 (FROM) (単位：FP/人月)

組織数	平均値	中央値	最大値	最小値
3	5.2	5	10	0.6

② HTMLの生産性 (TO) (単位：FP/人月)

組織数	平均値	中央値	最大値	最小値
2	15	15	20	10

③ HTMLの生産性 (FROM) (単位：STEP/人月)

組織数	平均値	中央値	最大値	最小値
3	1,933	2,000	3,000	800

④ HTMLの生産性 (TO) (単位：STEP/人月)

組織数	平均値	中央値	最大値	最小値
3	2,667	2,500	4,000	1,500

(8) J a v a の生産性

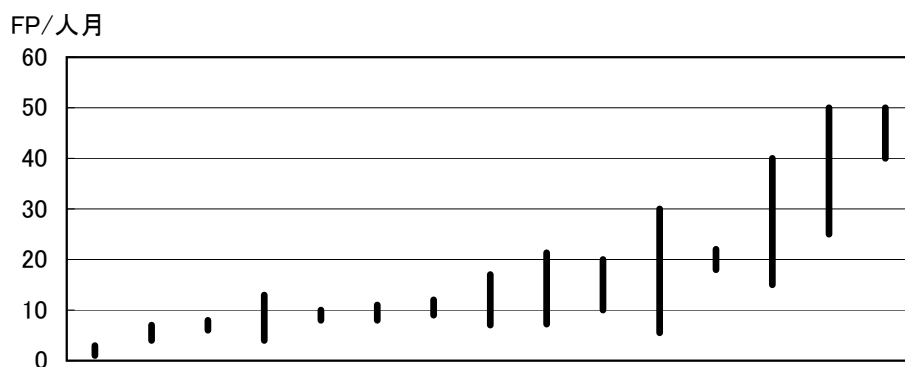
① J a v a の生産性 (FROM) (単位：FP/人月)

組織数	平均値	中央値	最大値	最小値
15	11.2	8	40	1

② J a v a の生産性 (TO) (単位：FP/人月)

組織数	平均値	中央値	最大値	最小値
16	20.4	15	50	3

Java生産性(FP/人月)



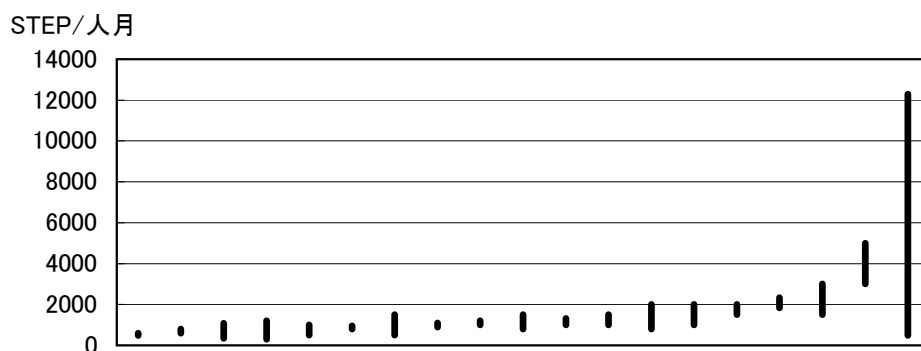
③ J a v a の生産性 (FROM) (単位：STEP/人月)

組織数	平均値	中央値	最大値	最小値
19	966	800	3,000	300

④ J a v a の生産性 (TO) (単位：STEP/人月)

組織数	平均値	中央値	最大値	最小値
20	2,329	1,500	12,300	590

Java生産性(STEP/人月)



(9) JavaScriptの生産性

① JavaScriptの生産性 (FROM) (単位：FP/人月)

組織数	平均値	中央値	最大値	最小値
1	5	5	5	5

② JavaScriptの生産性 (T0) (単位：FP/人月)

組織数	平均値	中央値	最大値	最小値
1	9	9	9	9

③ JavaScriptの生産性 (FROM) (単位：STEP/人月)

組織数	平均値	中央値	最大値	最小値
6	2,580	1,740	5,000	500

④ JavaScriptの生産性 (T0) (単位：STEP/人月)

組織数	平均値	中央値	最大値	最小値
6	8,257	3,270	20,000	1,000

(10) JSPの生産性

① JSPの生産性 (FROM) (単位：FP/人月)

組織数	平均値	中央値	最大値	最小値
2	8.4	8.4	10	6.7

② JSPの生産性 (T0) (単位：FP/人月)

組織数	平均値	中央値	最大値	最小値
2	25.5	25.5	30.9	20

③ JSPの生産性 (FROM) (単位：STEP/人月)

組織数	平均値	中央値	最大値	最小値
3	873	800	1,300	520

④ JSPの生産性 (T0) (単位：STEP/人月)

組織数	平均値	中央値	最大値	最小値
3	1,457	1,800	2,000	570

(11) P H P の生産性

① P H P の生産性 (FROM) (単位：FP/人月)

組織数	平均値	中央値	最大値	最小値
1	5.4	5.4	5.4	5.4

② P H P の生産性 (T0) (単位：FP/人月)

組織数	平均値	中央値	最大値	最小値
1	37	37	37	37

③ P H P の生産性 (FROM) (単位：STEP/人月)

組織数	平均値	中央値	最大値	最小値
3	800	500	1,500	400

④ P H P の生産性 (T0) (単位：STEP/人月)

組織数	平均値	中央値	最大値	最小値
3	1,533	1,600	2,000	1,000

(12) R u b y の生産性

① R u b y の生産性 (FROM) (単位：FP/人月)

組織数	平均値	中央値	最大値	最小値
0	－	－	－	－

② R u b y の生産性 (T0) (単位：FP/人月)

組織数	平均値	中央値	最大値	最小値
0	－	－	－	－

③ R u b y の生産性 (FROM) (単位：STEP/人月)

組織数	平均値	中央値	最大値	最小値
1	1,500	1,500	1,500	1,500

④ R u b y の生産性 (T0) (単位：STEP/人月)

組織数	平均値	中央値	最大値	最小値
1	2,000	2,000	2,000	2,000

(13) SQL (PL/SQLを含む) の生産性

① SQL の生産性 (FROM) (単位: FP/人月)

組織数	平均値	中央値	最大値	最小値
7	5.8	5	15	0.2

② SQL の生産性 (TO) (単位: FP/人月)

組織数	平均値	中央値	最大値	最小値
7	12.4	7	30.9	2.5

③ SQL の生産性 (FROM) (単位: STEP/人月)

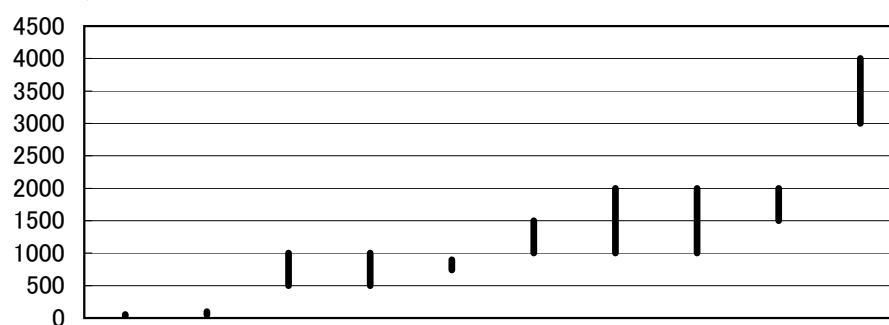
組織数	平均値	中央値	最大値	最小値
10	931	870	3,000	17.5

④ SQL の生産性 (TO) (単位: STEP/人月)

組織数	平均値	中央値	最大値	最小値
10	1,455	1,250	4,000	52.5

SQL生産性(STEP/人月)

STEP/人月



(14) VBの生産性

① VBの生産性 (FROM) (単位：FP/人月)

組織数	平均値	中央値	最大値	最小値
6	9.8	8.4	20	1

② VBの生産性 (TO) (単位：FP/人月)

組織数	平均値	中央値	最大値	最小値
7	25.2	21	50	2

③ VBの生産性 (FROM) (単位：STEP/人月)

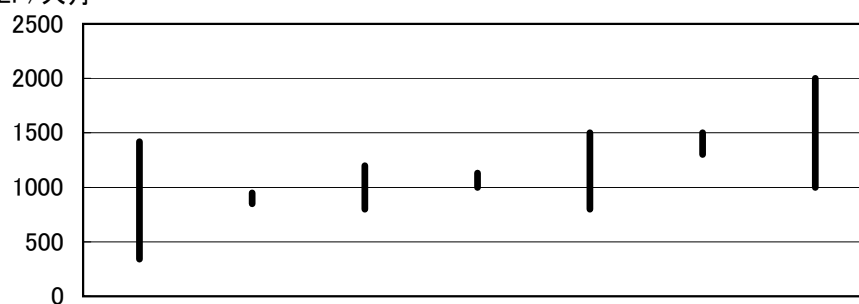
組織数	平均値	中央値	最大値	最小値
18	2,228	1,250	7,000	340

④ VBの生産性 (TO) (単位：STEP/人月)

組織数	平均値	中央値	最大値	最小値
19	3,460	2,500	10,000	2

VB生産性(STEP/人月)

STEP/人月



(15) VB. NETの生産性

① VB. NETの生産性 (FROM) (単位：FP/人月)

組織数	平均値	中央値	最大値	最小値
9	12	15	25.2	2

② VB. NETの生産性 (T0) (単位：FP/人月)

組織数	平均値	中央値	最大値	最小値
9	23.1	25.2	40	4

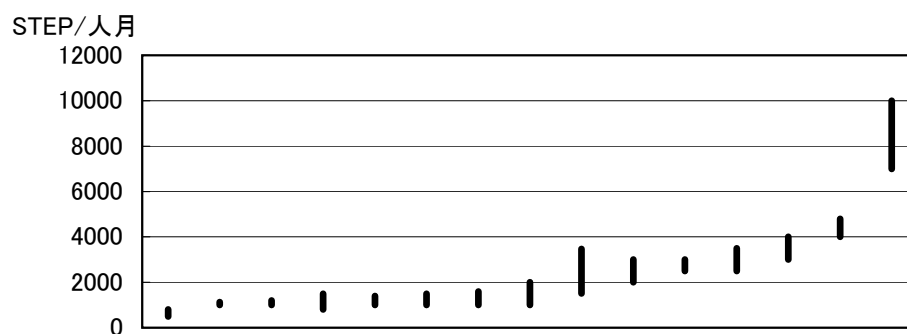
③ VB. NETの生産性 (FROM) (単位：STEP/人月)

組織数	平均値	中央値	最大値	最小値
15	1,987	1,000	7,000	500

④ VB. NETの生産性 (T0) (単位：STEP/人月)

組織数	平均値	中央値	最大値	最小値
16	2,744	1,800	10,000	800

VB. NET生産性(STEP/人月)



(16) VC#.NETの生産性

① VC#.NETの生産性 (FROM) (単位: FP/人月)

組織数	平均値	中央値	最大値	最小値
3	6.7	4	13	3.2

② VC#.NETの生産性 (TO) (単位: FP/人月)

組織数	平均値	中央値	最大値	最小値
4	24.6	16.2	59	7

③ VC#.NETの生産性 (FROM) (単位: STEP/人月)

組織数	平均値	中央値	最大値	最小値
7	1,534	1,000	4,000	370

④ VC#.NETの生産性 (TO) (単位: STEP/人月)

組織数	平均値	中央値	最大値	最小値
8	2,465	1,350	6,400	550

(17) XMLの生産性

① XMLの生産性 (FROM) (単位: FP/人月)

組織数	平均値	中央値	最大値	最小値
0	-	-	-	-

② XMLの生産性 (TO) (単位: FP/人月)

組織数	平均値	中央値	最大値	最小値
0	-	-	-	-

③ XMLの生産性 (FROM) (単位: STEP/人月)

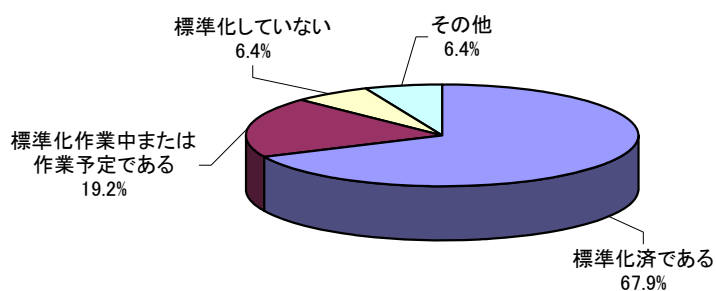
組織数	平均値	中央値	最大値	最小値
1	2,000	2,000	2,000	2,000

④ XMLの生産性 (TO) (単位: STEP/人月)

組織数	平均値	中央値	最大値	最小値
1	2,500	2,500	2,500	2,500

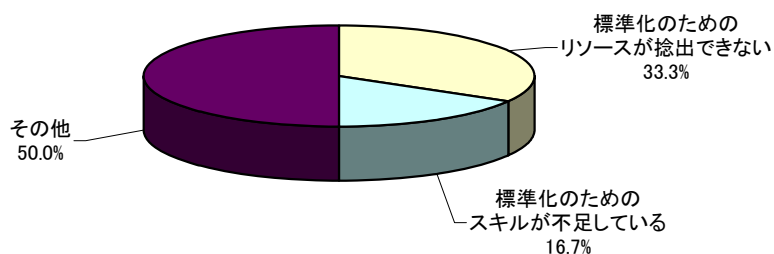
[6] 組織のシステム開発プロセスの標準化

選択肢	組織数	比率
システム開発プロセスは標準化済である	53	67.9%
システム開発プロセスは標準化作業中または作業予定である	15	19.2%
システム開発プロセスは標準化していない	5	6.4%
その他	5	6.4%
合計	78	100.0%



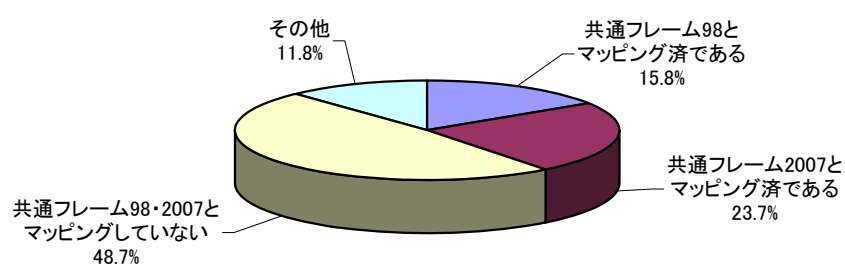
[7] システム開発プロセスを標準化していない理由

選択肢	組織数	比率
システム開発プロセスの標準化の必要性を感じていない	0	0.0%
システム開発プロセスの標準化の効果を期待していない	0	0.0%
システム開発プロセスの標準化のためのリソースが捻出できない	2	33.3%
システム開発プロセスの標準化のためのスキルが不足している	1	16.7%
その他	3	50.0%
合計	6	100.0%



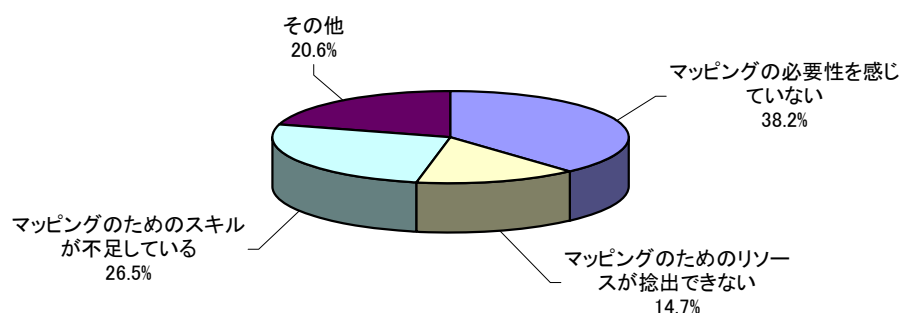
[8] システム開発プロセス標準と共通フレームとのマッピング状況

選択肢	組織数	比率
システム開発プロセス標準と共通フレーム98のアクティビティやタスクはマッピング済である	12	15.8%
システム開発プロセス標準と共通フレーム2007のアクティビティやタスクはマッピング済である	18	23.7%
システム開発プロセス標準と共通フレーム98・2007のアクティビティやタスクはマッピングしていない	37	48.7%
その他	9	11.8%
合計	76	100.0%



[9] システム開発プロセス標準と共通フレームをマッピングしていない理由

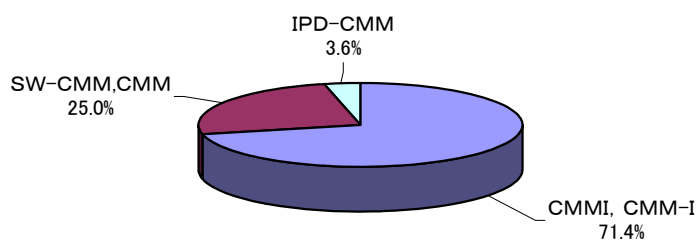
選択肢	組織数	比率
マッピングの必要性を感じていない	13	38.2%
マッピングの効果を期待していない	0	0.0%
マッピングのためのリソースが捻出できない	5	14.7%
マッピングのためのスキルが不足している	9	26.5%
その他	7	20.6%
合計	34	100.0%



[10] 組織のCMM

(1) CMMの種類

選択肢	組織数	比率
CMM I, CMM-I (CMM統合)	20	71.4%
SW-CMM, CMM (ソフトウェアCMM)	7	25.0%
SE-CMM (システムエンジニアリングCMM)	0	0.0%
IPD-CMM(統合製品開発能力成熟度モデル)	1	3.6%
SA-CMM (ソフトウェア調達能力成熟度モデル)	0	0.0%
合計	28	100.0%

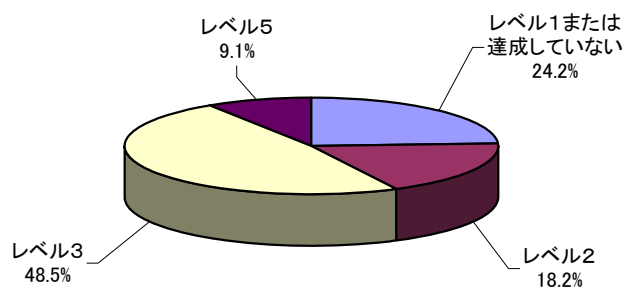


(2) アセスメント担当者

選択肢	組織数	比率
SEI (カーネギーメロン大学ソフトウェア工学研究所) 公認リードアセッサによる評価	16	66.7%
SEI (カーネギーメロン大学ソフトウェア工学研究所) 公認リードアセッサ以外による評価	8	33.3%
合計	24	100.0%

(3) 達成レベル

選択肢	組織数	比率
レベル1または達成していない	8	24.2%
レベル2	6	18.2%
レベル3	16	48.5%
レベル4	0	0.0%
レベル5	3	9.1%
合計	33	100.0%



[11] 組織のオフショア開発の実績

(1) オフショア開発の実績

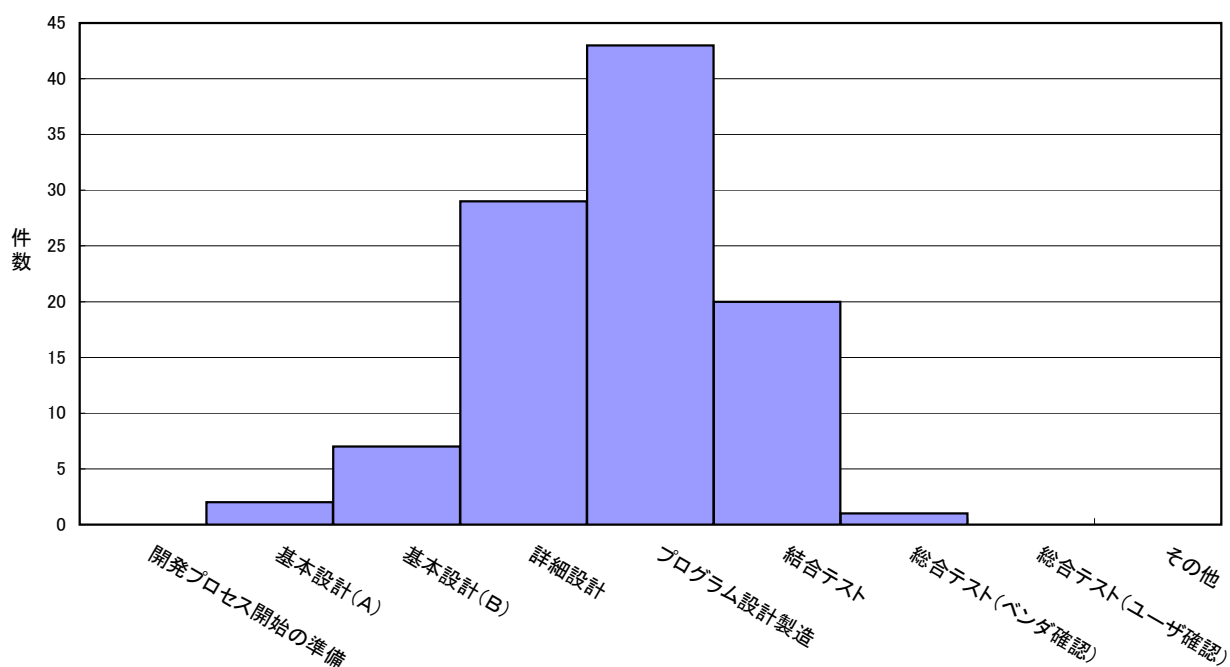
選択肢	組織数	比率
オフショア開発の実績がある	44	61.1%
オフショア開発の実績がない	28	38.9%
合計	72	100.0%

(2) オフショア開発先

選択肢	組織数	比率
中国	60	81.1%
ベトナム	9	12.2%
インド	4	5.4%
ロシア	1	1.4%
合計	74	100.0%

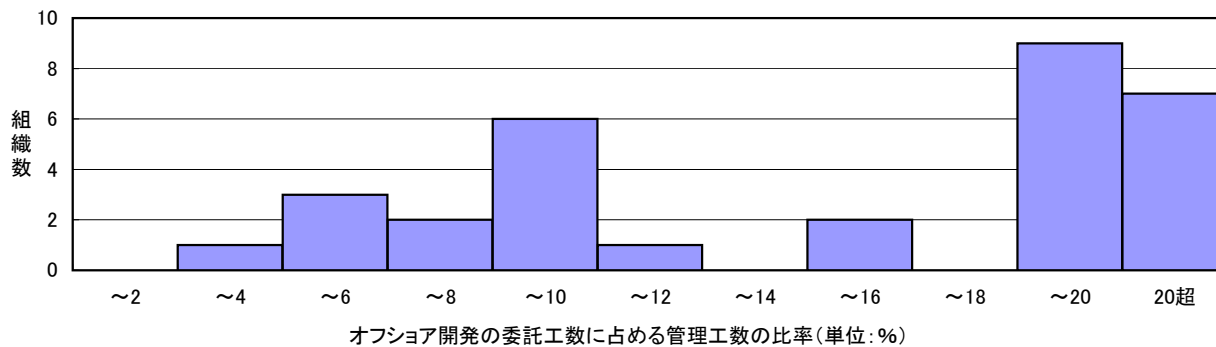
(3) オフショア開発の委託範囲(複数選択あり)

選択肢	組織数
開発プロセス開始の準備	0
基本設計 (A)	2
基本設計 (B)	7
詳細設計	29
プログラム設計製造	43
結合テスト	20
総合テスト (ベンダ確認)	1
総合テスト (ユーザ確認)	0
その他	0
合計	102



(4) オフショア開発の委託工数に占める管理工数比率（単位：％）

組織数	平均値	中央値	最大値	最小値
32	16.1	17.5	30	0



(5) 国内開発とオフショア開発の工数比率

① 国内開発工数・自社（単位：％）

組織数	平均値	中央値	最大値	最小値
29	49.9	50	95	5

② 国内開発工数・外部委託（単位：％）

組織数	平均値	中央値	最大値	最小値
29	32.3	35	90	0

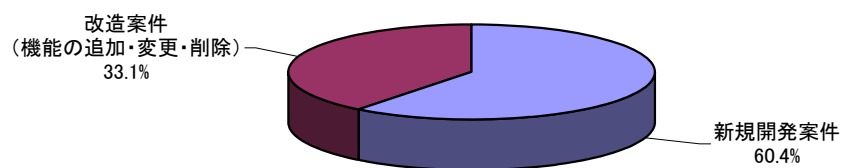
③ オフショア開発工数（単位：％）

組織数	平均値	中央値	最大値	最小値
29	17.8	10	65	1

3 調査票Ⅲ 個別のソフトウェア開発事例に関する集計結果

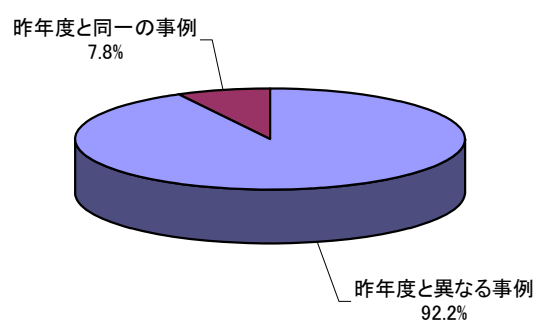
[1] 開発事例の新規開発案件、改造案件区分

選択肢	件数	比率
新規開発案件	81	60.4%
改造案件（機能の追加・変更・削除）	53	39.6%
合計	134	100.0%



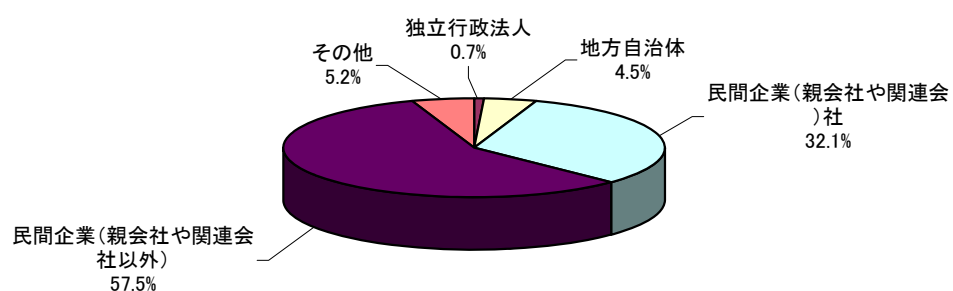
[2] 開発事例の同異区分

選択肢	件数	比率
昨年度と異なる事例	119	92.2%
昨年度と同一の事例	10	7.8%
合計	129	100.0%



[3] 開発事例の委託者の分類

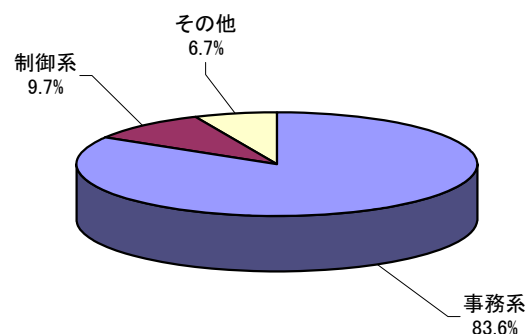
選択肢	件数	比率
府省庁	0	0.0%
独立行政法人	1	0.7%
地方自治体	6	4.5%
民間企業（親会社や関連会社）	43	32.1%
民間企業（親会社や関連会社以外）	77	57.5%
その他	7	5.2%
合計	134	100.0%



[4] 開発事例の適用分野・業種

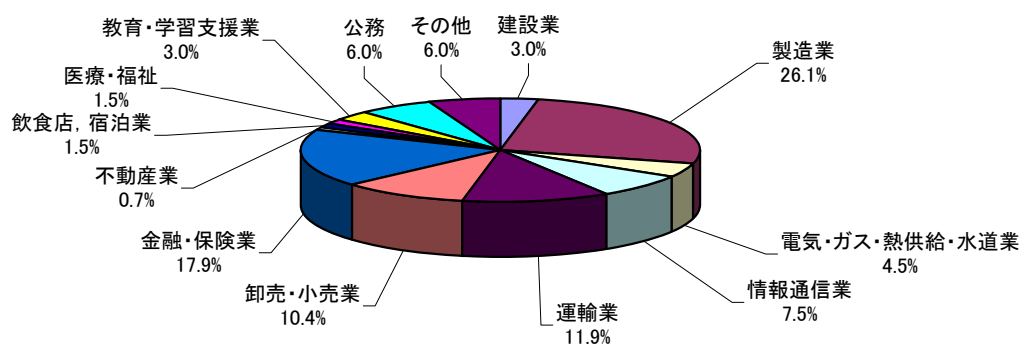
(1) 適用分野

選択肢	件数	比率
事務系	112	83.6%
制御系	13	9.7%
その他	9	6.7%
合計	134	100.0%



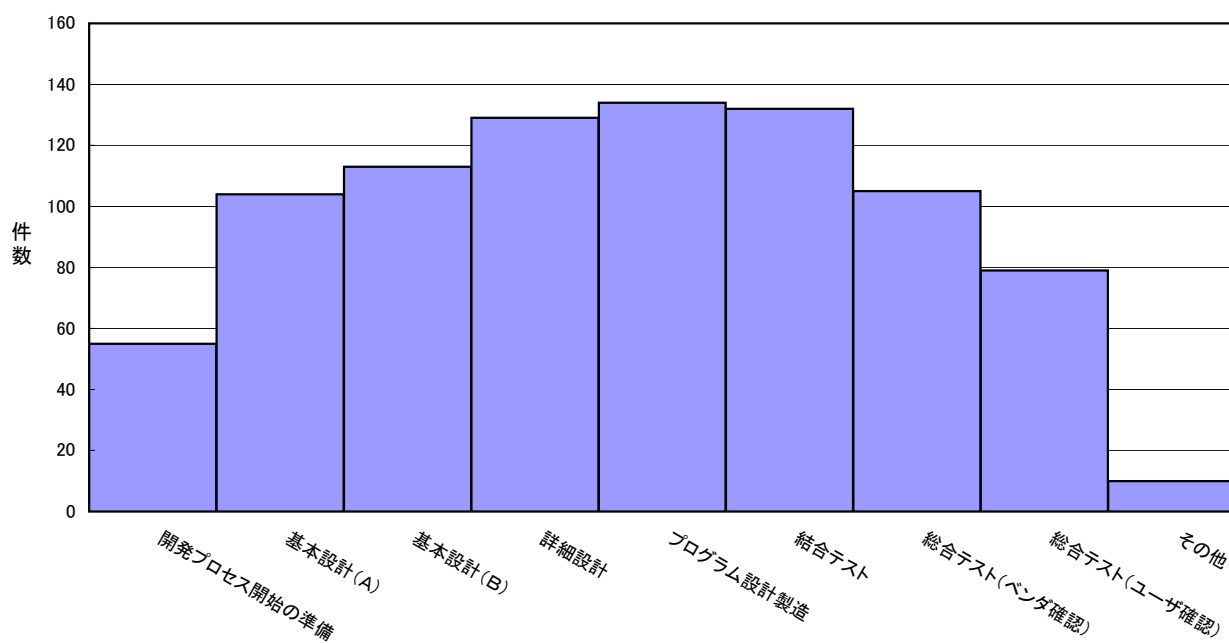
(2) 適用業種

選択肢	件数	比率
建設業	4	3.0%
製造業	35	26.1%
電気・ガス・熱供給・水道業	6	4.5%
情報通信業	10	7.5%
運輸業	16	11.9%
卸売・小売業	14	10.4%
金融・保険業	24	17.9%
不動産業	1	0.7%
飲食店、宿泊業	2	1.5%
医療・福祉	2	1.5%
教育・学習支援業	4	3.0%
公務	8	6.0%
その他	8	6.0%
合計	134	100.0%



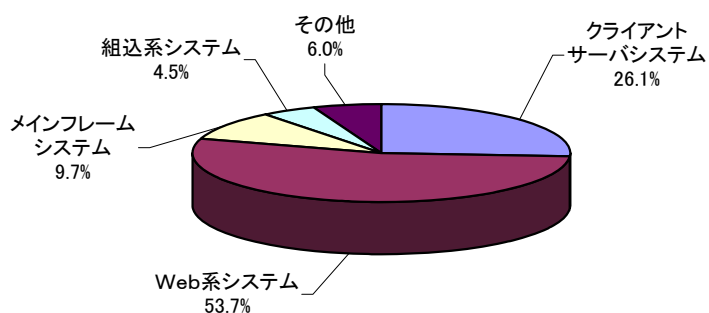
[5] 開発事例の開発工程範囲

選択肢	件数
開発プロセス開始の準備	55
基本設計（A）	104
基本設計（B）	113
詳細設計	129
プログラム設計製造	134
結合テスト	132
総合テスト（ベンダ確認）	105
総合テスト（ユーザ確認）	79
その他	10
合計	861



[6] 開発事例のシステム構成

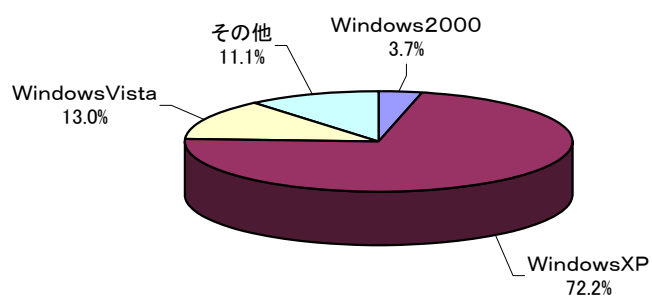
選択肢	件数	比率
クライアントサーバシステム	35	26.1%
Web系システム	72	53.7%
メインフレームシステム	13	9.7%
組込系システム	6	4.5%
その他	8	6.0%
合計	134	100.0%



[7] 開発事例のOS

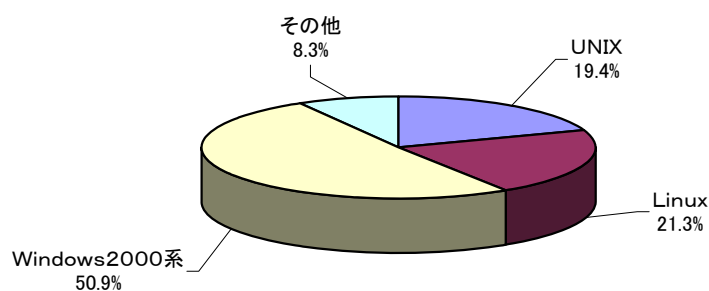
(1) クライアントOS

選択肢	件数	比率
Windows 2000	2	3.7%
Windows XP	39	72.2%
Windows Vista	7	13.0%
その他	6	11.1%
合計	54	100.0%



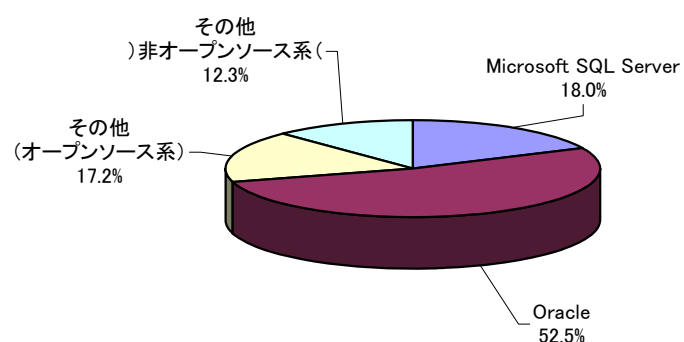
(2) サーバOS

選択肢	件数	比率
UNIX	21	19.4%
Linux	23	21.3%
Windows 2000系	55	50.9%
その他	9	8.3%
合計	108	100.0%



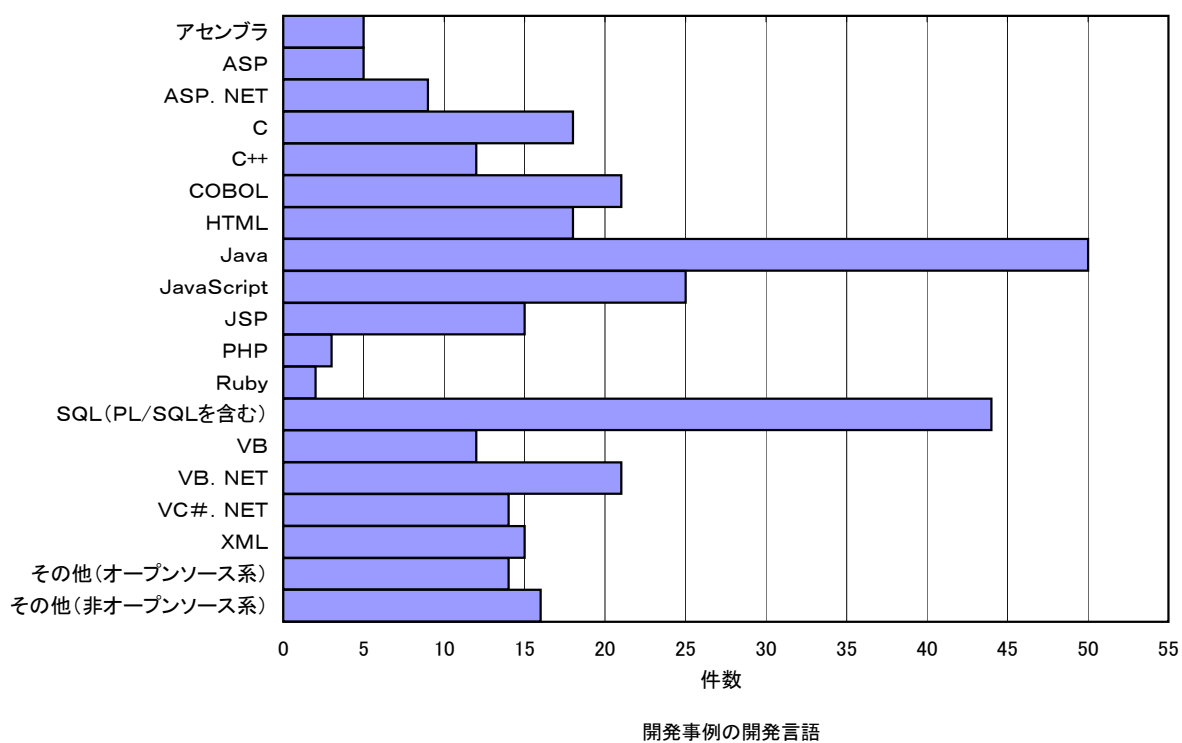
[8] 開発事例のデータベース

選択肢	件数	比率
Microsoft SQL Server	22	18.0%
Oracle	64	52.5%
その他（オープンソース系）	21	17.2%
その他（非オープンソース系）	15	12.3%
合計	122	100.0%



[9] 開発事例の開発言語または開発ツール（複数選択あり）

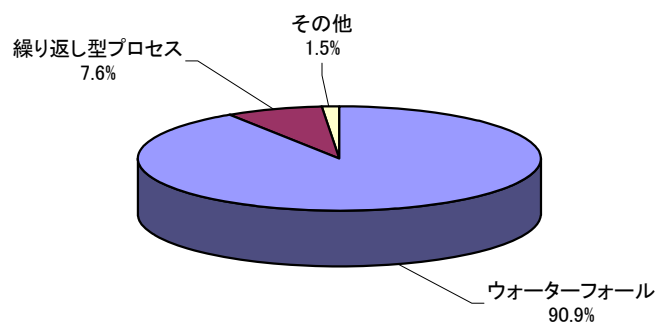
選択肢	件数
アセンブラ	5
ASP	5
ASP. NET	9
C	18
C++	12
COBOL	21
HTML	18
Java	50
JavaScript	25
JSP	15
PHP	3
Ruby	2
SQL（PL/SQLを含む）	44
VB	12
VB. NET	21
VC#. NET	14
XML	15
その他（オープンソース系）	14
その他（非オープンソース系）	16
合計	319



[10] 開発事例の開発方法論

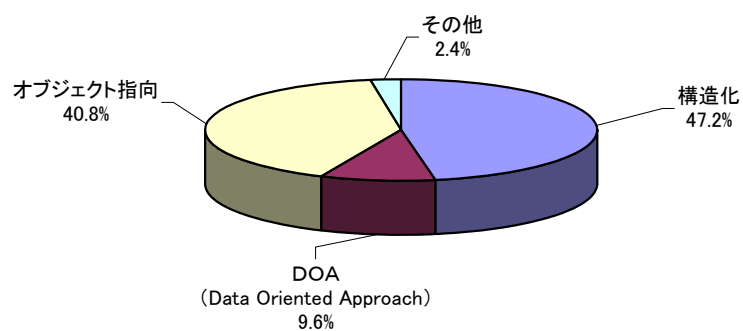
(1) プロセスモデル

選択肢	件数	比率
ウォーターフォール	120	90.9%
繰り返し型プロセス	10	7.6%
その他	2	1.5%
合計	132	100.0%



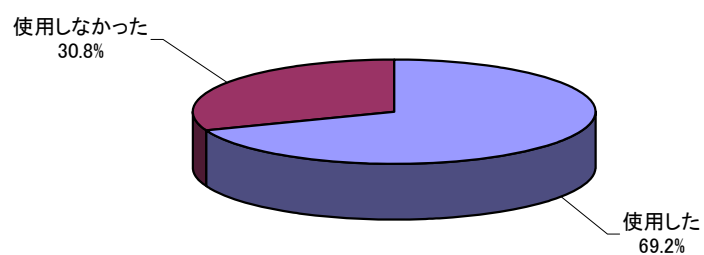
(2) 開発技法

選択肢	件数	比率
構造化	59	47.2%
DOA (Data Oriented Approach)	12	9.6%
オブジェクト指向	51	40.8%
その他	3	2.4%
合計	125	100.0%



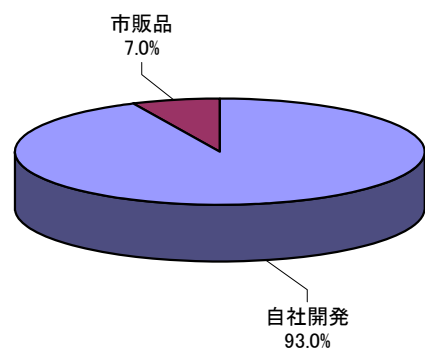
(3) システム開発標準

選択肢	件数	比率
使用した	90	69.2%
使用しなかった	40	30.8%
合計	130	100.0%



(4) システム開発標準の入手方法

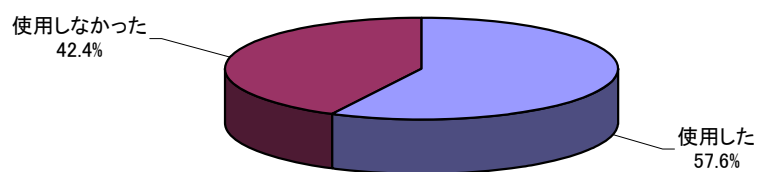
選択肢	件数	比率
自社開発	80	93.0%
市販品	6	7.0%
合計	86	100.0%



[11] 開発事例のフレームワーク使用状況

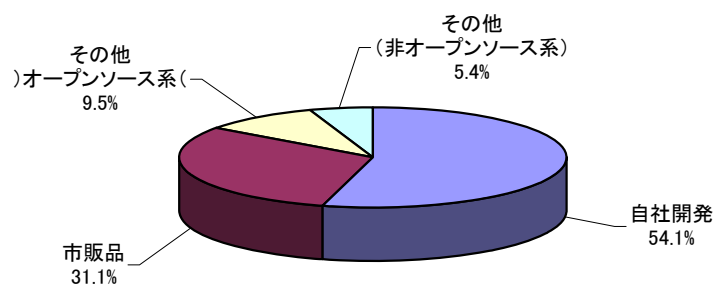
(1) フレームワーク使用状況

選択肢	件数	比率
使用した	76	57.6%
使用しなかった	56	42.4%
合計	132	100.0%



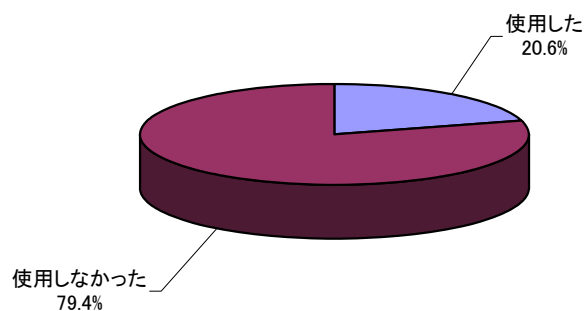
(2) フレームワークの入手方法

選択肢	件数	比率
自社開発	40	54.1%
市販品	23	31.1%
その他（オープンソース系）	7	9.5%
その他（非オープンソース系）	4	5.4%
合計	74	100.0%



[12] 開発事例のパッケージソフトウェア使用状況

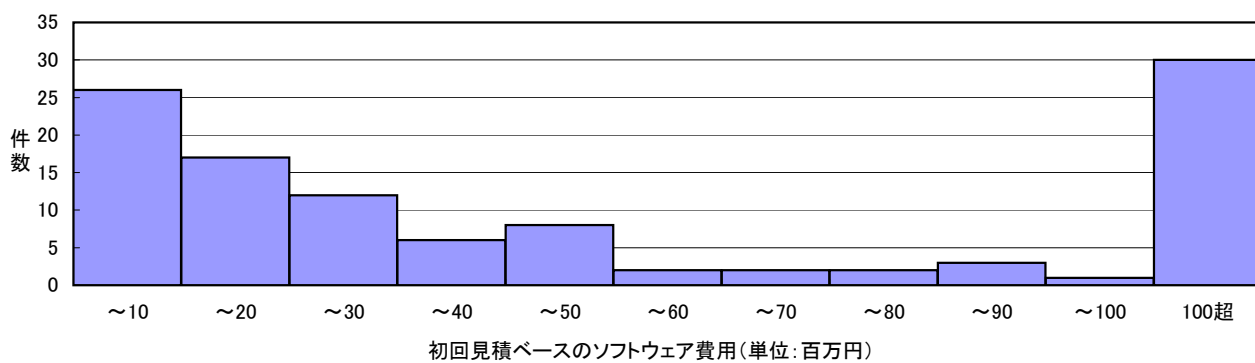
選択肢	件数	比率
使用した	26	20.6%
使用しなかった	100	79.4%
合計	126	100.0%



[13] 開発事例のソフトウェア費用

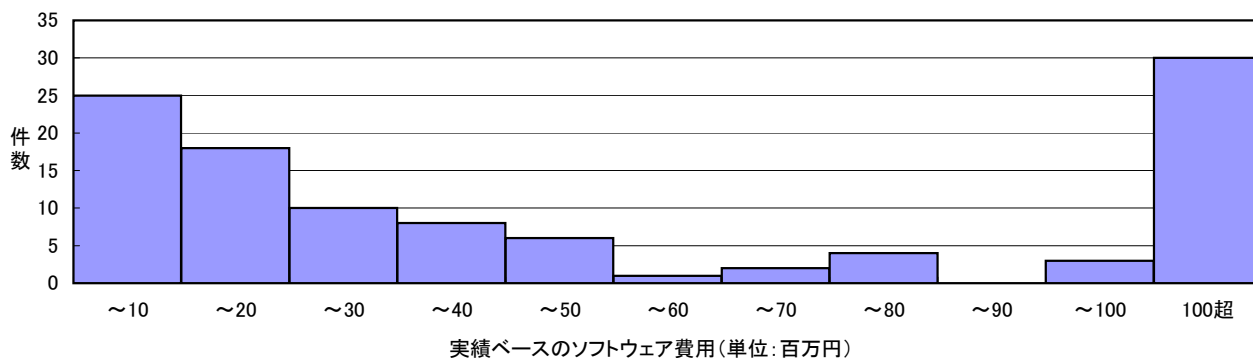
(1) 初回見積ベースのソフトウェア費用 (単位：百万円)

件数	平均値	中央値	最大値	最小値
109	107.8	30	935	0.7



(2) 実績ベースのソフトウェア費用 (単位：百万円)

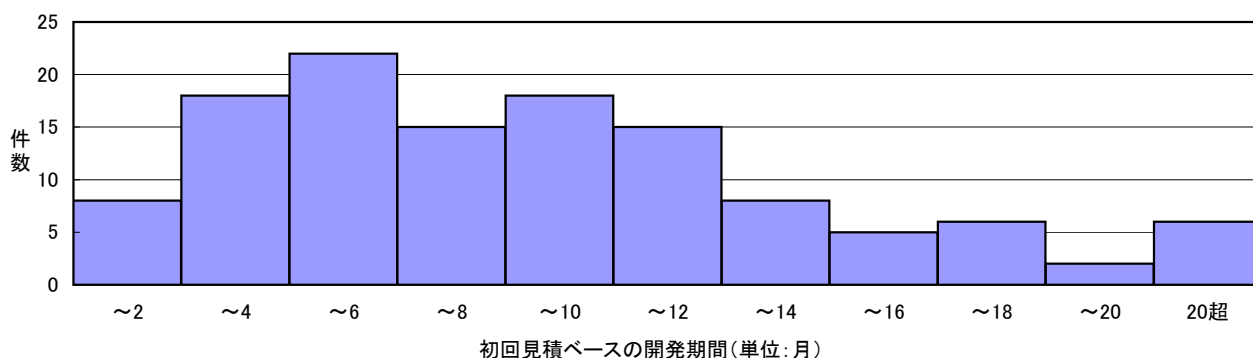
件数	平均値	中央値	最大値	最小値
107	120.2	31	1,148	1



[14] 開発事例のソフトウェア開発期間

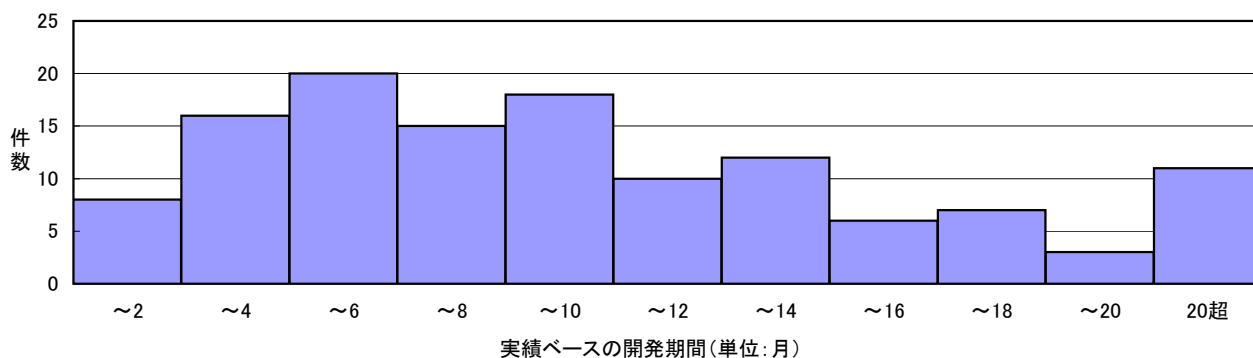
(1) 初回見積ベースの開発期間 (単位：月)

件数	平均値	中央値	最大値	最小値
123	9.9	8	110	0.5



(2) 実績ベースの開発期間 (単位：月)

件数	平均値	中央値	最大値	最小値
126	10.7	9	80	0.5

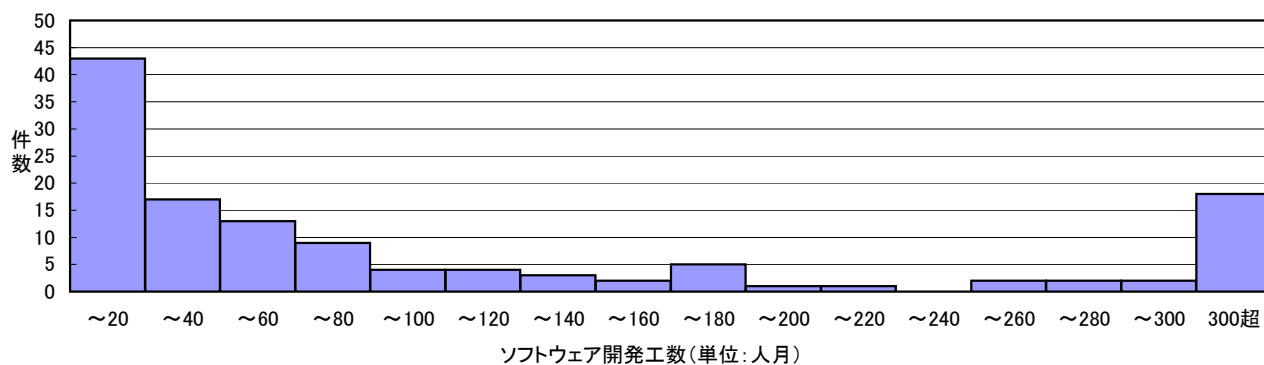


[15] 開発事例のソフトウェア開発工数と規模

(1) 実績ベースでの開発工数と規模

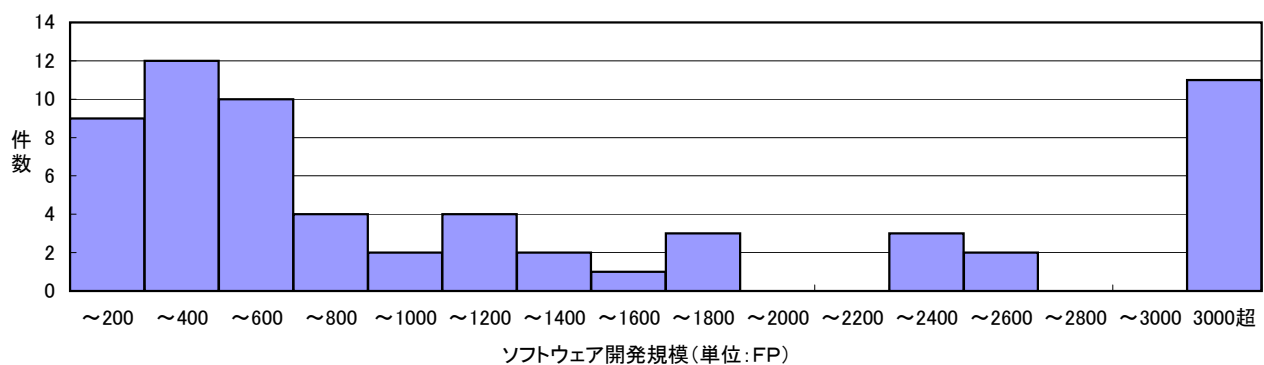
① ソフトウェア開発工数 (単位：人月)

件数	平均値	中央値	最大値	最小値
126	135.9	45	1,275	0.5



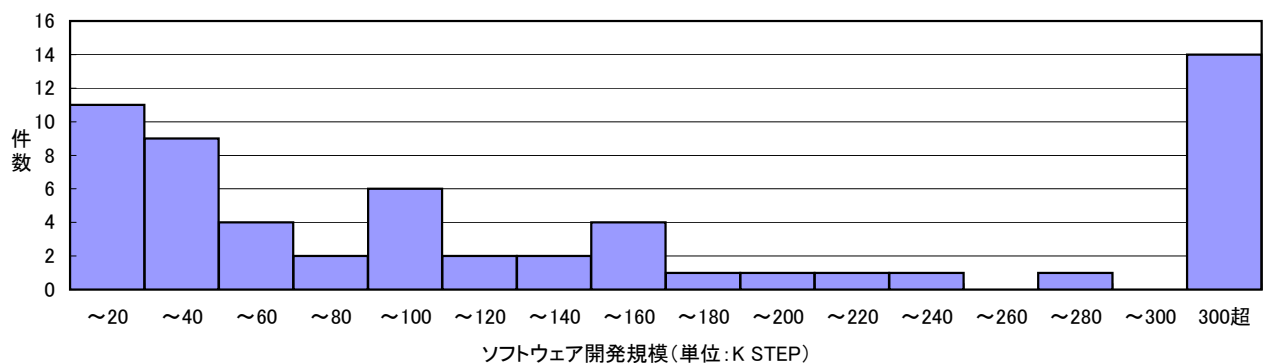
② ソフトウェア開発規模 (単位：FP)

件数	平均値	中央値	最大値	最小値
63	1,658	644	10,102	35



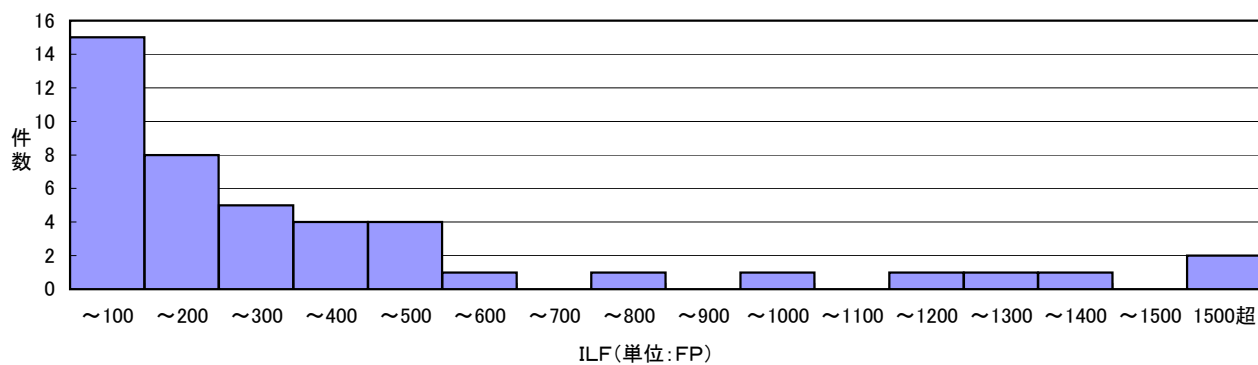
③ ソフトウェア開発規模 (単位：KSTEP)

件数	平均値	中央値	最大値	最小値
59	3,145	96	172,561	5



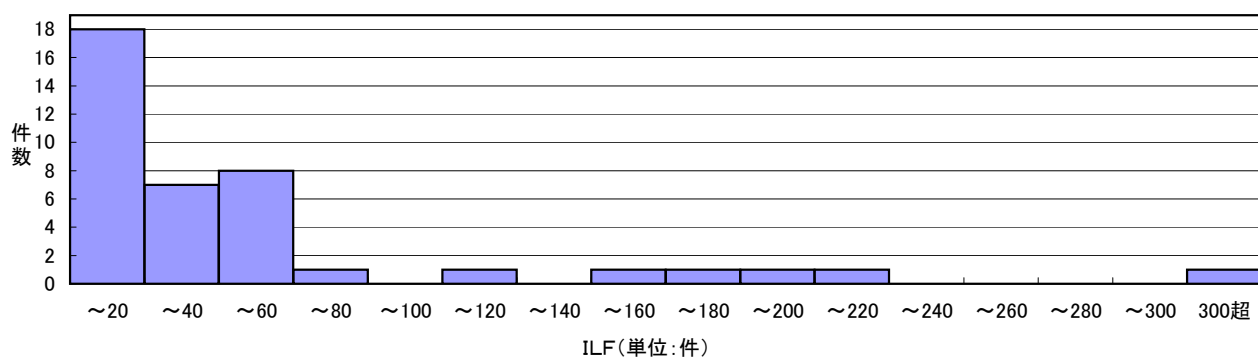
④ I L F (単位：FP)

件数	平均値	中央値	最大値	最小値
44	473.9	185.5	4,481	2



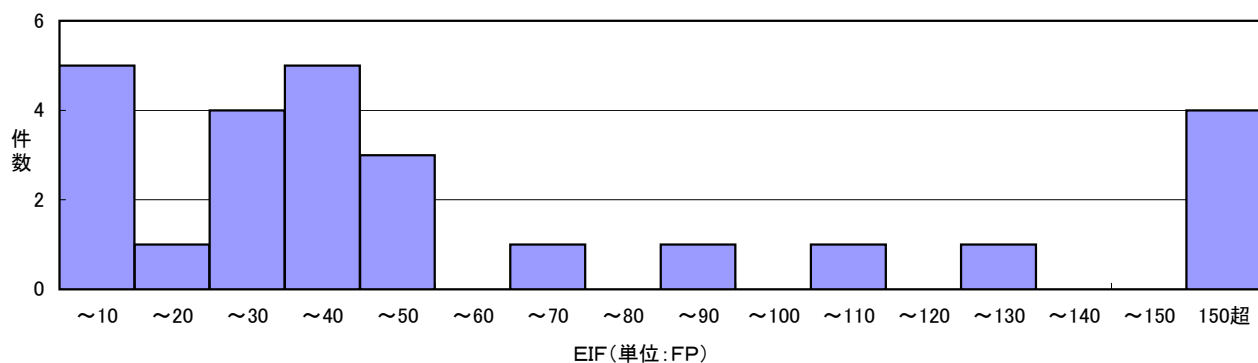
⑤ I L F (単位：件)

件数	平均値	中央値	最大値	最小値
40	56.9	26	571	1



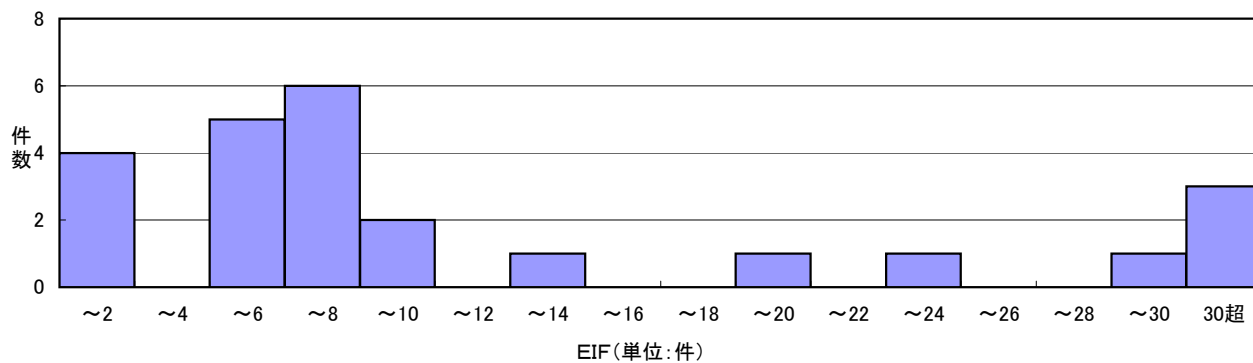
⑥ E I F (単位：FP)

件数	平均値	中央値	最大値	最小値
31	64.6	31	534	0



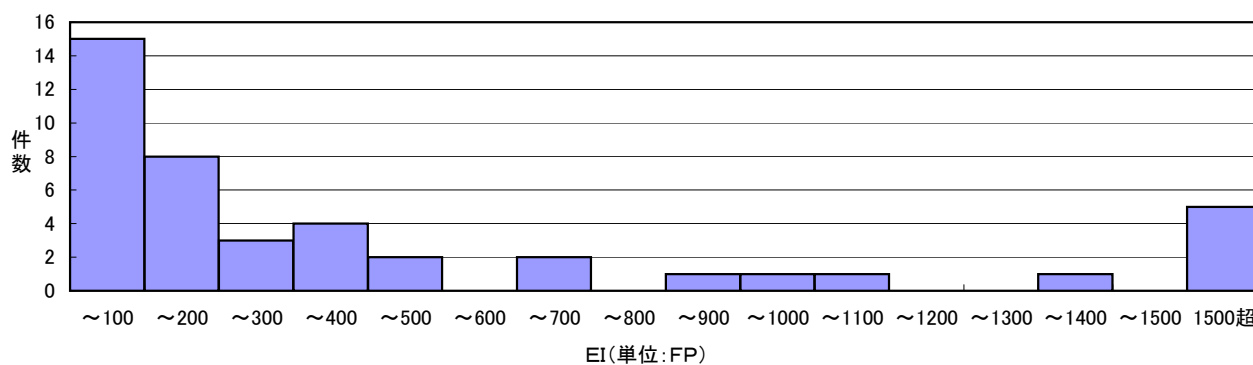
⑦ E I F (単位：件)

件数	平均値	中央値	最大値	最小値
27	12.9	7	76	0



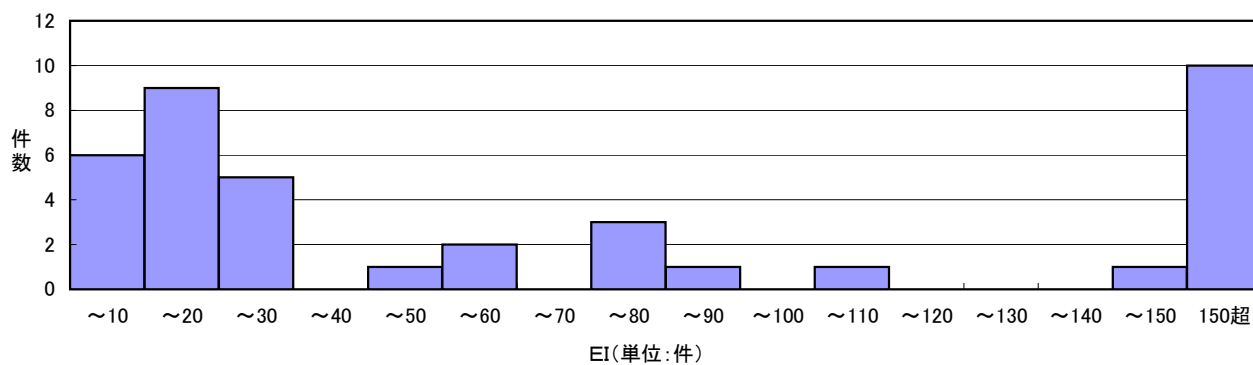
⑧ E I (単位：FP)

件数	平均値	中央値	最大値	最小値
43	601	152	4,208	3



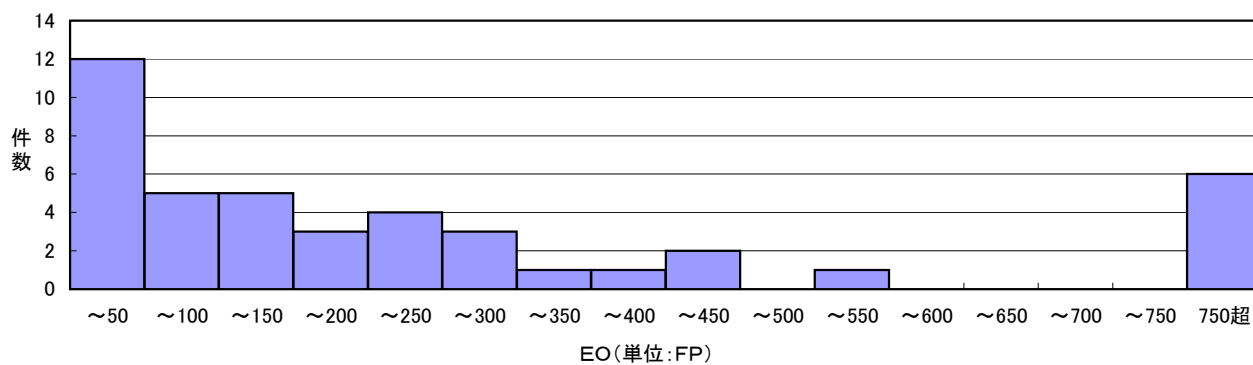
⑨ E I (単位：件)

件数	平均値	中央値	最大値	最小値
39	138.2	30	887	1



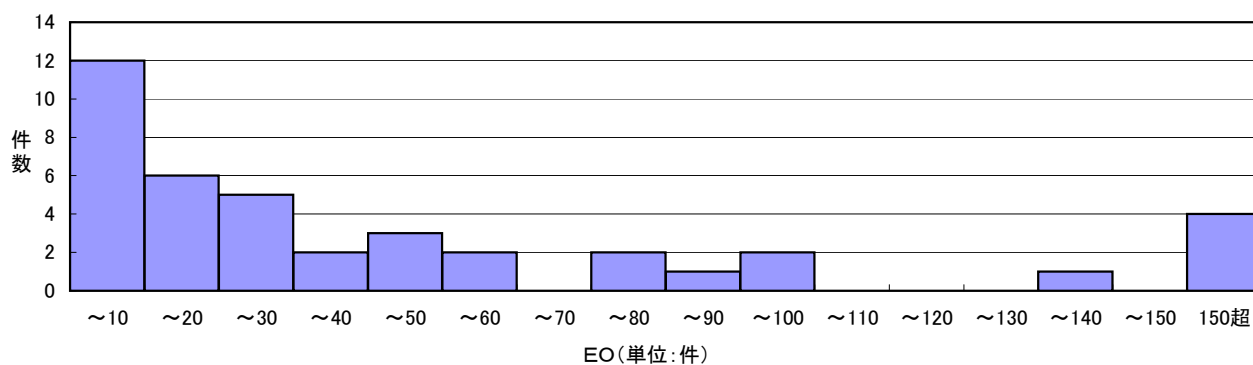
⑩ E O (単位：FP)

件数	平均値	中央値	最大値	最小値
43	327.4	123	2,030	5



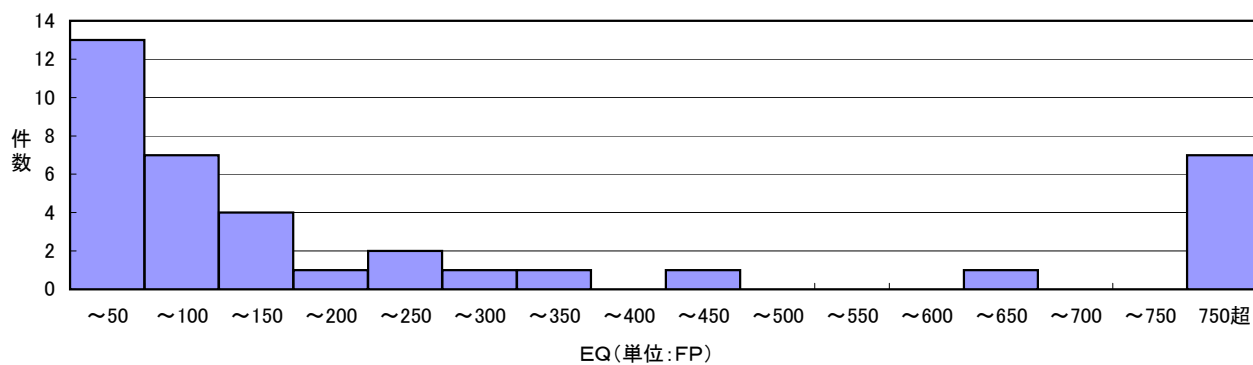
⑪ E O (単位：件)

件数	平均値	中央値	最大値	最小値
40	59.1	23	372	1



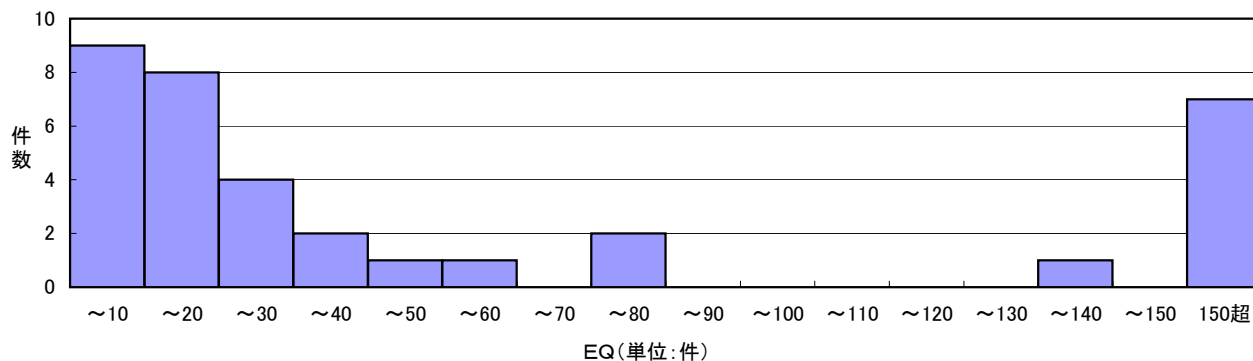
⑫ E Q (単位：FP)

件数	平均値	中央値	最大値	最小値
41	443.8	74	3,329	0



⑬ E Q (単位：件)

件数	平均値	中央値	最大値	最小値
37	115.3	20	866	0



(2) 改造案件における実績ベースのE F PとA F P

① E F P (単位：FP)

件数	平均値	中央値	最大値	最小値
7	752.6	281	2,548	35

② A F P (単位：FP)

件数	平均値	中央値	最大値	最小値
7	2,412	643	7,200	289

(3) 改造案件における実績ベースのステップ数

① 母体・非改造部分 (単位：KSTEP)

件数	平均値	中央値	最大値	最小値
15	1,407	311	6,679	60

② 削除・改造部分 (単位：KSTEP)

件数	平均値	中央値	最大値	最小値
10	4.5	2.5	29	0

③ 改造・改造部分 (単位：KSTEP)

件数	平均値	中央値	最大値	最小値
23	80.3	50	354	1

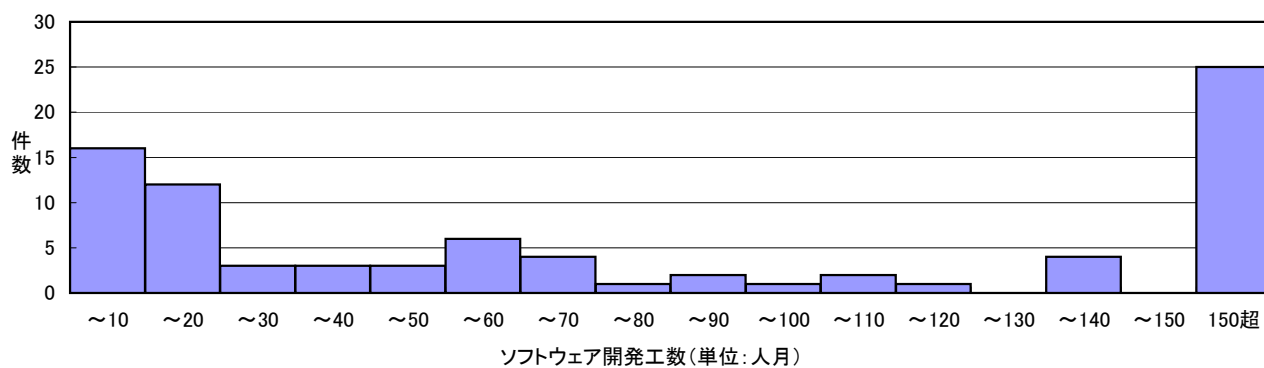
④ 追加・改造部分 (単位：KSTEP)

件数	平均値	中央値	最大値	最小値
18	77.4	40	320	1

(4) 初回見積ベースでの開発工数と規模

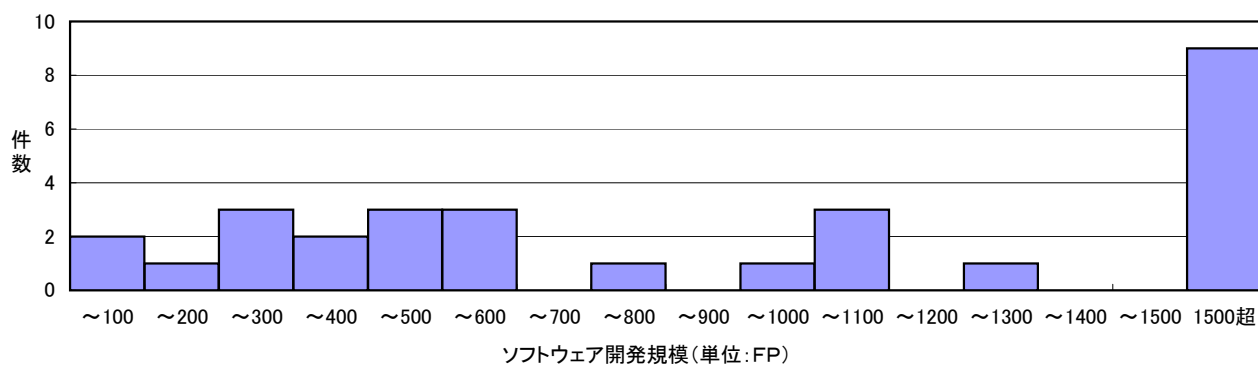
① ソフトウェア開発工数（人月）（単位：人月）

件数	平均値	中央値	最大値	最小値
83	150.6	59	1,039	0.5



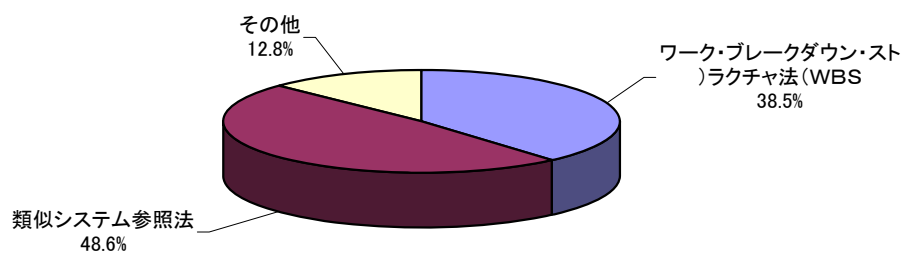
② ソフトウェア開発規模（単位：FP）

件数	平均値	中央値	最大値	最小値
29	1,787	785	10,264	44



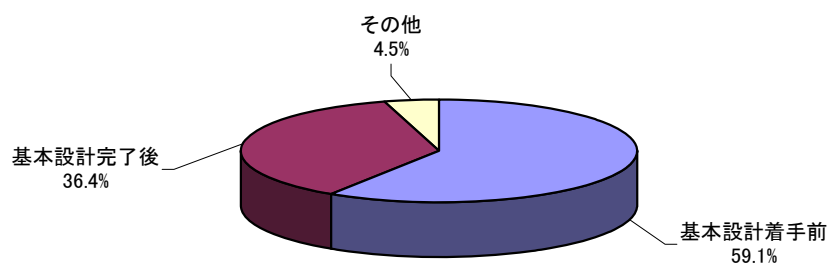
(5) 見積開発工数の算出方法

選択肢	件数	比率
ワーク・ブレイクダウン・ストラクチャ法（WBS）	42	38.5%
類似システム参照法	53	48.6%
その他	14	12.8%
合計	109	100.0%



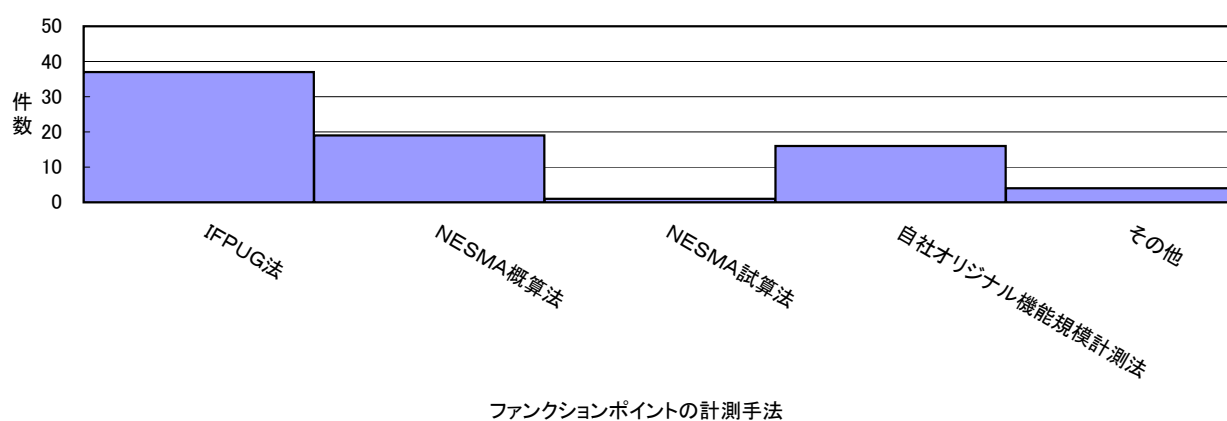
(6) 見積開発規模 (FP) を計測したタイミング

選択肢	件数	比率
基本設計着手前	26	59.1%
基本設計完了後	16	36.4%
その他	2	4.5%
合計	44	100.0%



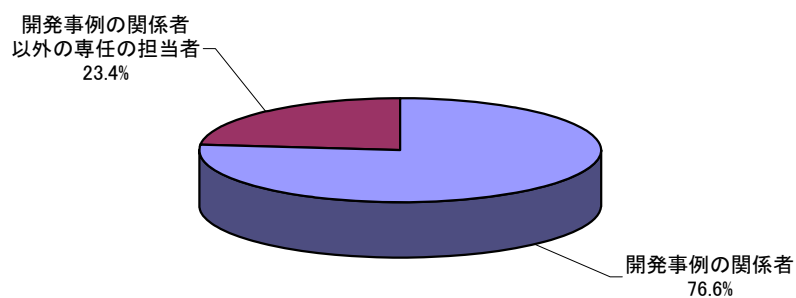
(7) ファンクションポイントの計測手法 (複数選択あり)

選択肢	件数
I F P U G 法	37
N E S M A 概算法	19
N E S M A 試算法	1
自社オリジナル機能規模計測法	16
その他	4
合計	77



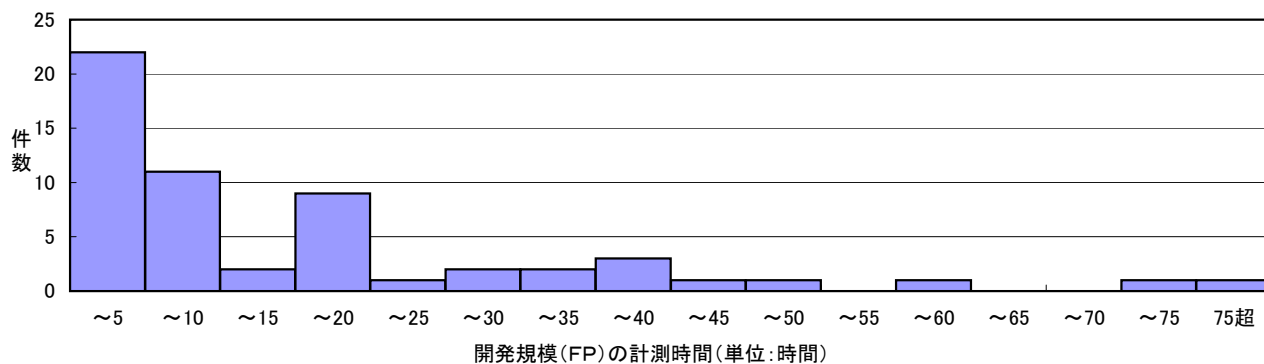
(8) 開発事例のファンクションポイントの計測者

選択肢	件数	比率
開発事例の関係者	49	76.6%
開発事例の関係者以外の専任の担当者	15	23.4%
その他	0	0.0%
合計	64	100.0%



(9) 開発規模（F P）の計測時間（単位：時間）

件数	平均値	中央値	最大値	最小値
57	16	8	90	0.5

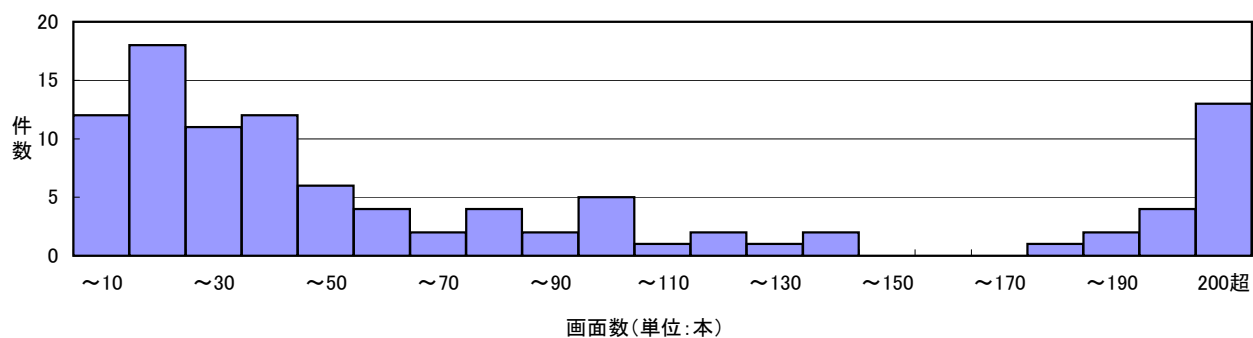


[16] 開発事例の画面数・帳票数・ファイル数・バッチプログラム数

(1) 画面数

(単位：本)

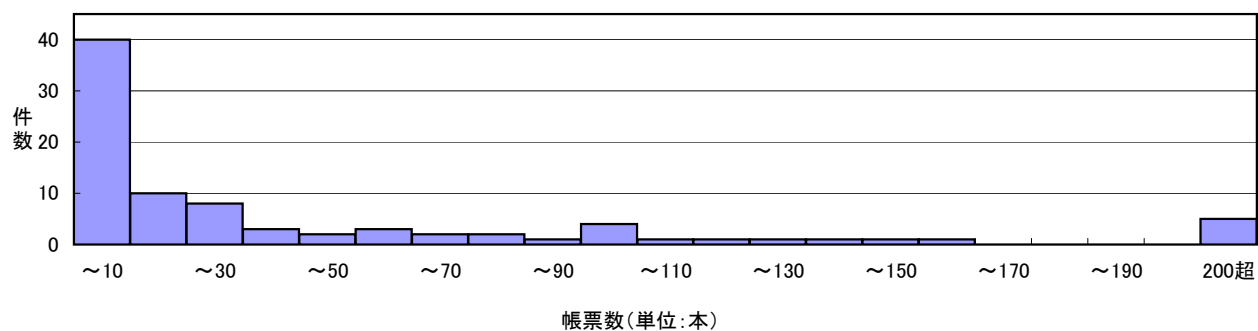
件数	平均値	中央値	最大値	最小値
104	102	40	1,250	0



(2) 帳票数

(単位：本)

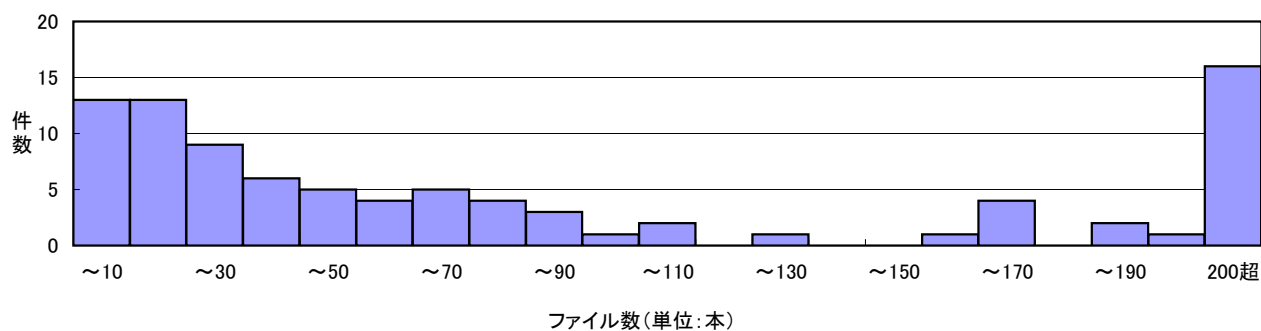
件数	平均値	中央値	最大値	最小値
104	44	7	800	0



(3) ファイル数

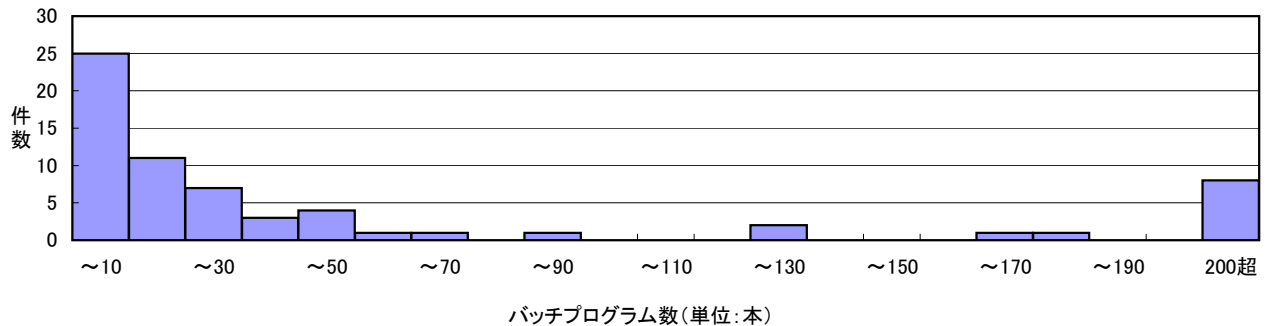
(単位：本)

件数	平均値	中央値	最大値	最小値
104	90	36	722	0



(4) バッチプログラム数 (単位: 本)

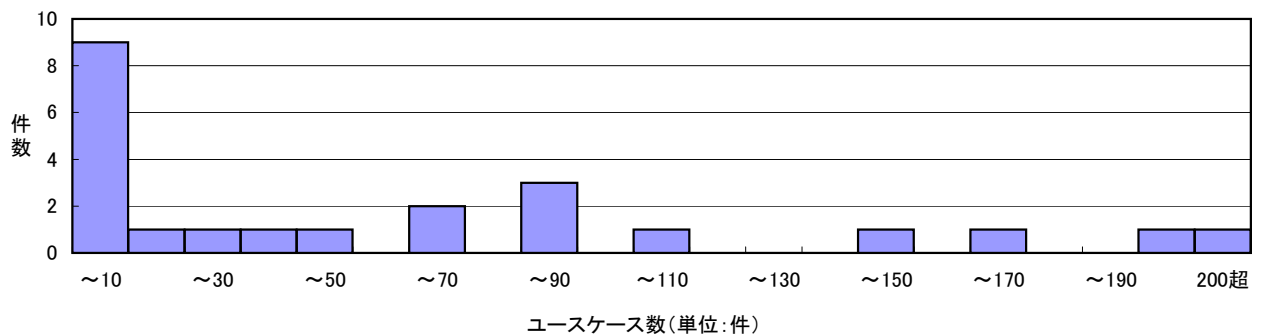
件数	平均値	中央値	最大値	最小値
104	62	4.5	1,540	0



[17] 開発事例のユースケース数・アクター数

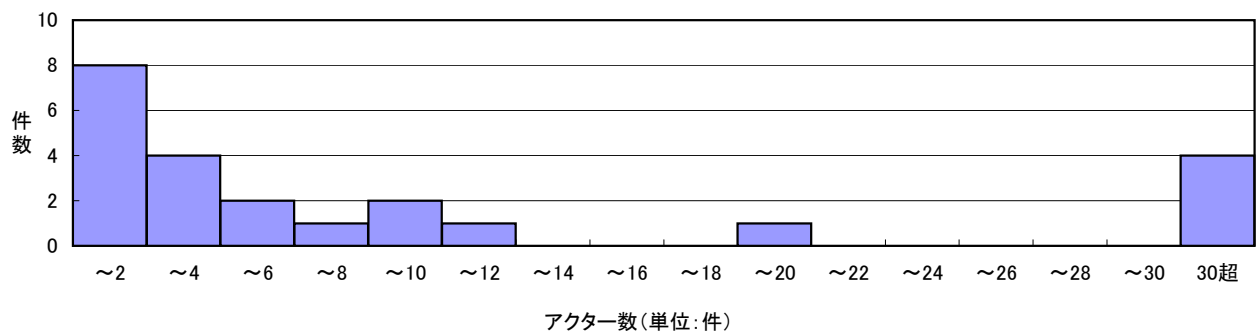
(1) ユースケース数 (単位: 件)

件数	平均値	中央値	最大値	最小値
23	61	39	219	1



(2) アクター数 (単位: 件)

件数	平均値	中央値	最大値	最小値
23	234.2	4	5,000	1



[18] 開発事例の品質管理

(1) 内部レビュー回数と指摘件数

① 内部レビュー回数・設計レビュー (回) (単位: 回)

件数	平均値	中央値	最大値	最小値
74	102.8	5	3,000	1

② 内部レビュー回数・コードレビュー (回) (単位: 回)

件数	平均値	中央値	最大値	最小値
54	42.9	4	578	0

③ 内部レビュー回数・テストレビュー (回) (単位: 回)

件数	平均値	中央値	最大値	最小値
64	299.7	4	16,337	1

④ レビュー指摘件数・設計レビュー (単位: 件)

件数	平均値	中央値	最大値	最小値
73	863.8	100	12,426	0

⑤ レビュー指摘件数・コードレビュー (単位: 件)

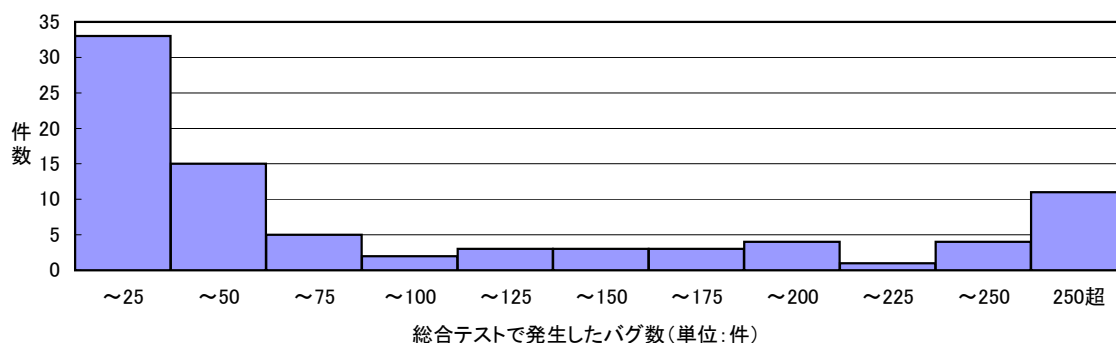
件数	平均値	中央値	最大値	最小値
52	340.6	30	2,914	0

⑥ レビュー指摘件数・テストレビュー (単位: 件)

件数	平均値	中央値	最大値	最小値
58	294.1	24	3,098	0

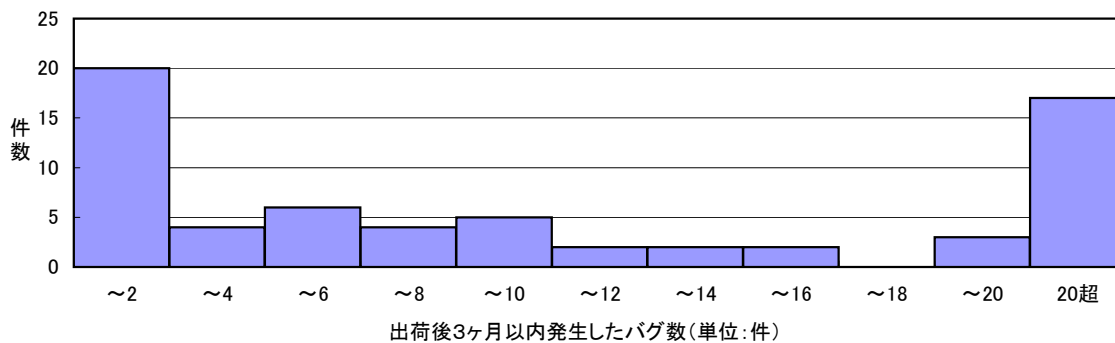
(2) 総合テストで発生したバグ数 (単位: 件)

件数	平均値	中央値	最大値	最小値
91	116.9	32	1,656	0



(3) 出荷後3ヶ月以内発生したバグ数 (単位: 件)

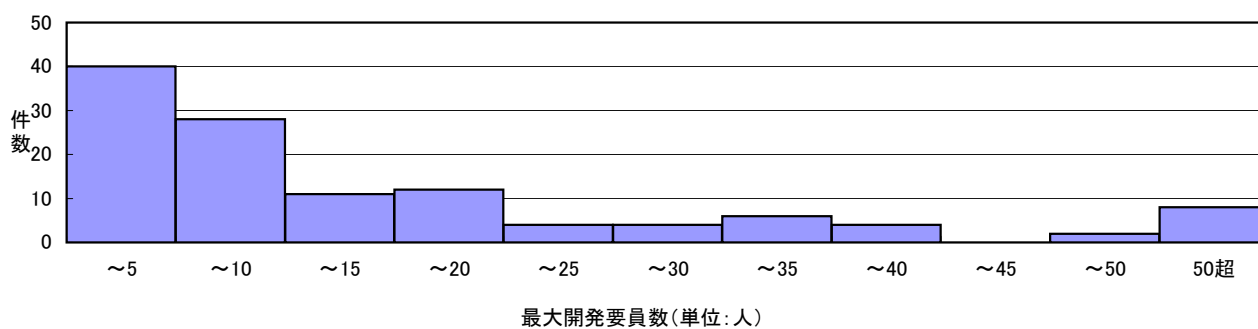
件数	平均値	中央値	最大値	最小値
86	19.3	3	247	0



[19] 開発事例の最大開発要員

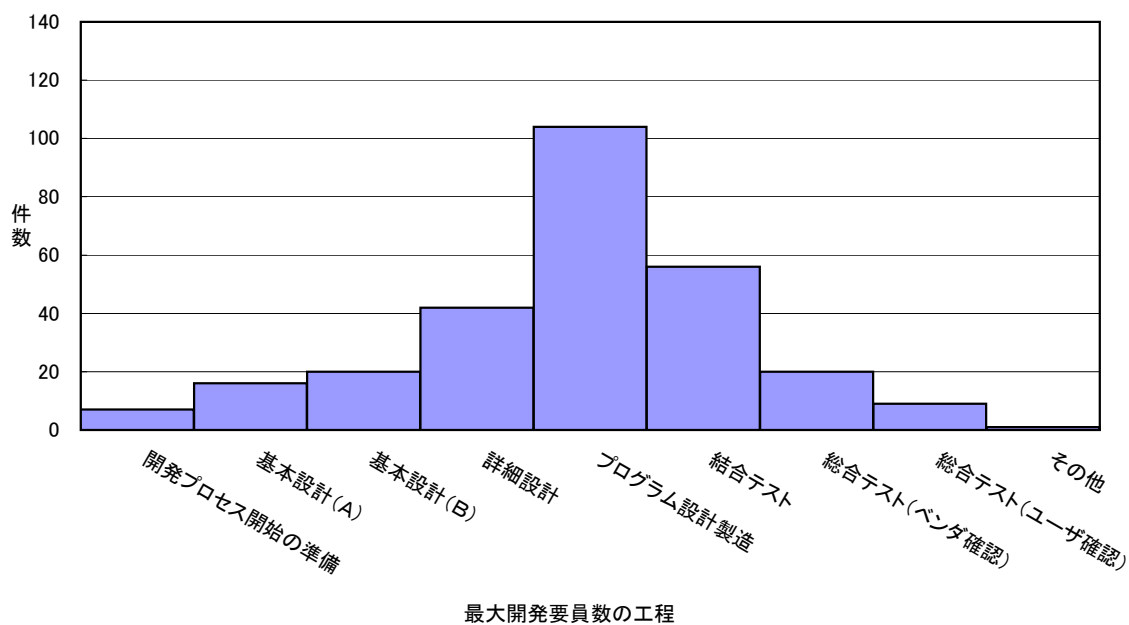
(1) 開発事例の最大開発要員数 (単位: 人)

件数	平均値	中央値	最大値	最小値
119	19.5	10	300	1



(2) 開発事例の最大開発要員数の工程（複数選択あり）

工程	件数
開発プロセス開始の準備	7
基本設計（A）	16
基本設計（B）	20
詳細設計	42
プログラム設計製造	104
結合テスト	56
総合テスト（ベンダ確認）	20
総合テスト（ユーザ確認）	9
その他	1
合計	275



[20] 開発事例のリスク管理

(1) 見積もりベース（受注前）で行った、当該システムのリスク管理の内容

懸念された重大な問題は発生しなかった。ただし予兆に気を配り、顕在化する前に対応した。
顧客との要件の食い違いを減らすため新旧比較の画面イメージ資料などユーザによりわかりやすい資料を準備してユーザレビューの回数を多く入れて要件のズレを減らした。
結合テスト以降に納期遅延の恐れがあるため、懸念段階の基本設計段階から納期延長の交渉を申し入れ、納期変更を合意の上、行った。
総合テスト（ユーザ確認）の直前に異動になったため、ユーザへテスト実施支援体制を整備し、対応した。
開発量の見積りにリスク分を上乗せした。
要件変更受付期間を基本設計工程が開始した時点で設定し、それ以降の要件変更については、別工数および、スケジュールの引きなおしを行った。
新顧客のため、情報収集を密にした。
業務経験が無い場合、業務解説書、旧システムのヒアリング等を実施した。
経験者の採用。
短期間開発のため、進捗管理を徹底。
新規ユーザのため、納品後に要望が多く発生することを懸念していたため、開発の途中で出来る限り実際の画面を見せて確認を行った。
実行ベースでのパフォーマンスが出るか、検証する時間がない。
要件が不明な場合、打合せを増やし対策した。
Web経験者が少ない→技術力のある協力会社へ一括発注した。
他プロジェクトとの競合。
オフショア開発（他社）。
性能に関するリスク管理実施。
OracleReport 未経験。
製造工程以降に発生する仕様変更・追加。
実績ベース。
プロジェクト要員の確保。
リッチクライアント方式の技術検証を基本設計の段階で実施。問題の早期発見を行った。
コンテンジェンシープランのユーザとの合意。
仕様の確定、進捗。
自社PMOに進捗分析を実施してもらった。
顧客による検証段階で、仕様の改善・追加要望対応による工数肥大が発生。
他システムとの連携で仕様変更が発生。
積算範囲外の機能要求が発生する可能性があるため、対応案件一覧をベースラインに顧客と合意しておく。
大規模な開発のためスケジュール遅延、品質悪化の可能性がある。
品質管理基準を定め、品質悪化の原因分析し、根本的な原因を特定する会議を開催し解決する。
顧客のシステム開発経験が少ないので、妥当性の確認を基本設計時のプロトタイプ等で検証、確認した。
顧客都合で運用開始が遅れたが、当社が支援できる部分を行いサポートした。
着手遅延や他要因による機能追加による作業遅延（納期遅延リスク）の場合、客先と納期を最重要課題と合意し、対応範囲を検討。
仕様変更発生（スコープ変更リスク）の場合、変更運用ルールを客先と合意し、残予算と残人的資源の範囲内で実施することで合意。
既存リソースの二重管理（品質低下リスク）の場合、構成管理、リリース前バージョンチェックの徹底。
移行障害（運用開始遅延）の場合、リスク分散目的のリハーサル実施。旧環境への戻し手順の準備。
スコープ変更。
スケジュール変更。
管理台帳を作成し、定期的な管理を行った。

テスト項目以外のテスト検証を行い、テスト項目以外の項目で障害が発生しないようにする。
規模確定後の機能の増減を変更管理 要員の欠員、欠勤に対し、作業が止まらない様な体制作り。
改正内容が判明し、規模が大きくなる事。また、予定期間外に作業が発生する事が判ったため、対応予定内容の一部を後の対応とする事によって、品質の向上を計った。
長期間にわたるプロジェクトにおいて、既存システムの改修を行うことから、開発要員がほかに優先度の高い案件にアサインされ十分な要員確保ができなくなる。
開発期間が短期間であったので開発要員については、生産性の高いメンバーで構成するようにリスク管理を実施した。
開発課題の進捗管理。
プロジェクトの進捗管理（WBS表、スケジュールなど）。
実績費用管理。
要件工程から機能の漏れの管理。
要件未決定事項の管理。
株券電子化を契機にした新しいビジネスモデルのため、業務運用面で問題がないか評価できない。予算も限られているため、機能をもれなく実装できるのであれば、業務運用の簡便さを追及しないようにする。
ハードウェアが日程どおりに上がってこないリスク。
ハードウェアとのインタフェースで仕様どおりに機能しないリスク。
要件定義、概要設計終了後の再見積の通知。
作業範囲、納品物の明確化。
各フェーズに必要な実工数だけの工数を確保するのではなく、一定のマージンを見て工数管理を行った。
プロジェクト内において「リスク登録簿」を作成、管理し、月1回の社内プロジェクト状況報告会にて社内全体で情報共有および対策検討を行った。
担当者が多忙の為、工程別のレビューが遅延したり、ユーザーの受入検証が遅くなってしまった為、仕様変更が多少発生した。

(2) 実績ベース（受注後）で行った、当該システムのリスク管理の内容

考えられるリスクをリストアップし、発生の軽減策と、発生時の対応策を設定した。
受注単位を分割することによりリスクを回避。
自社フレームワーク開発および標準化が未経験であった事に対する品質不良・スケジュール遅延の恐れに対して、自社フレームワーク開発標準化経験のある社内他部署の支援および開発会社の確保を行った。
開発要員の増員が必要となった場合の確保について、開発会社へのリソース増強の余裕の確認と条件に合致する開発会社を複数候補リストアップして問題発生に備えた。
客先の希望納期が早く、スケジュールが厳しいリスクについて、開発範囲のフェーズ分けの提示を行った。
人事異動により、お客様（ユーザ）が変わる事による要件変更が発生する。
要件定義工程が完了した時点で基本設計工程以降の工数算出を行った。
新顧客のため、情報収集を密にした。
業務経験が無いため、業務解説書、旧システムのヒアリング等を実施した。
短期間開発のため、進捗管理を徹底。
新規ユーザのため、見積範囲を明確にして、受注後に規模が膨らまないように提案書を作成した。
要件でペンディング事項の取込方法・時期が明確でない、リスク分。
移行作業を含めたスコープ。
下流工程の体制。
保守費用と要員。
開発中の変更依頼工数、納品後の対応工数。
オフショア開発（他社）。
OracleReport 未経験。
仕様変更に関するリスクと、性能実現のリスク、人員確保のリスクを発現時の影響分析、回避検討を実施。

体制の強化。
新技術（Solaris版Oracleが未経験）。
開発中の仕様変更、規模増大についての取り扱いを前もって顧客とプロジェクトで調整を行った（工数増大が一定以上になるものの対応はリリース後とするなど）。
見積ベース。
プロジェクト要員の確保。
新業務J-SOX法対応で仕様変更のリスクを低減するようにした。
3層C/S型リッチクライアント方式の技術的検証。
新規顧客。
コストコンテエンジェンシーの確保。
スコープの変更。
機能が見えない。複雑そうな機能については、見積り金額を割り増しした。
新業務（顧客側）。
契約に含まれる仕様が一部確定していない。
仕様の漸増が予測される。
制度対応のため、外部要件によるカットオーバーの延長が発生する可能性がある。
新顧客でシステム構築等のノウハウを学びたい意向があった。
当社標準のシステム構築を基本として開発を行った。
他案件からの改修、機能追加要件発生(スコープ変更リスク) →プロジェクト計画時に、客先ステークホルダーと必要不可欠な改修以外、保守凍結することを合意。また変更発生時は残予算と残人的資源の範囲内で実施することで合意。
キーマンが他プロジェクトへアサインされる（要員不足）→予備を持ったスケジュールを立てる。
定例会議（レビュー含む）のみは出席できるようコミュニケーション計画で事前に決めておく。
受注確度。
外部調達。
直結する取引所の更改による接続仕様の提示時期の遅延。
管理台帳を作成し、定期的な監視を行った。
仕様を出来るだけ明確にする。
見積もりと実績の差を埋める為、規模見積りの決定を基本設計後に調整。
改正内容が見えない部分もあったため、システムテスト終了予定を早めに設定し、不明点による工数増大に備えた。また、テスト期間について、以前の類似した改正対応に比べて多めに設定した。
見積前提条件の変更に伴う規模の変動。
仕様変更の多発による作業工程の見直し。
各工程における検収条件が不明確。
外注費の調整。
詳細設計以降の外注要員が未確定。
複数外注会社への依頼により品質確保が懸念。
お客様ネットワーク回線の速度が低い場合があり、レスポンスが懸念。
お客様側に外部アドバイザーがいる為、スコープ変更が懸念。
仕様が不確定の部分は、仕様のブレをMAXで想定し見積もった。
長期間にわたるプロジェクトにおいて、既存システムの改修を行うことから、開発要員がほかに優先度の高い案件にアサインされ十分な要員確保ができなくなる。
要件の確認もれ。
開発期間が短期間であったので開発要員については、生産性の高いメンバーで構成するようにリスク管理を実施した。

手戻り想定として、見積り金額に10%上乗せした。
製造業ではなく金融業のシステムとしてとらえ、見積り金額を上乗せした。
要件定義に時間がかかると予想される案件だったので、工数を多めに見積もった。
中国オフショア分の品質指標値の見直し。
金融部門のため、証券業務の知識が乏しい。
証券部門から有識者の支援を求める。
要件定義、概要設計終了後の再見積の通知。
作業範囲、納品物の明確化。
主要部分以外であいまいな部分がある。
ユーザからの問合せや軽微な仕様変更、予定外の旅費等、未確定な工数に対応できるよう、プロジェクト管理費を算出した。また過去案件より、基本設計・詳細設計の見積は15%増で算出。
分割納品にて、第一次リリース時期が決まっていたが、なかなか注文が頂けなかった。

[21] 開発事例の生産性変動要因

(1) 発注者側の生産性変動要因(各項目が選択された事例の件数)

選択肢	影響度判定(生産性)				
	1	2	3	4	5
機能性	13	31	51	21	3
信頼性	3	23	50	26	17
プラットフォーム	1	8	74	22	13
開発スケジュール要求	7	10	75	22	4
発注要件の明確度・安定度	18	58	30	9	4
参画割合	3	16	63	24	12
合計	45	146	343	124	53

(2) 受注者側の生産性変動要因(各項目が選択された事例の件数)

選択肢	影響度判定(生産性)				
	1	2	3	4	5
先行モデルの流用と標準モデルの採用	16	11	47	36	8
プロジェクト管理者の経験と能力	6	17	59	23	13
アナリストの経験と能力	23	28	43	13	4
SE・プログラマの経験と能力	2	14	38	39	26
合計	47	70	187	111	51

[22] 開発事例のソフトウェア開発工程別工数と職種別参画比率

(1) 開発工程別工数の記入

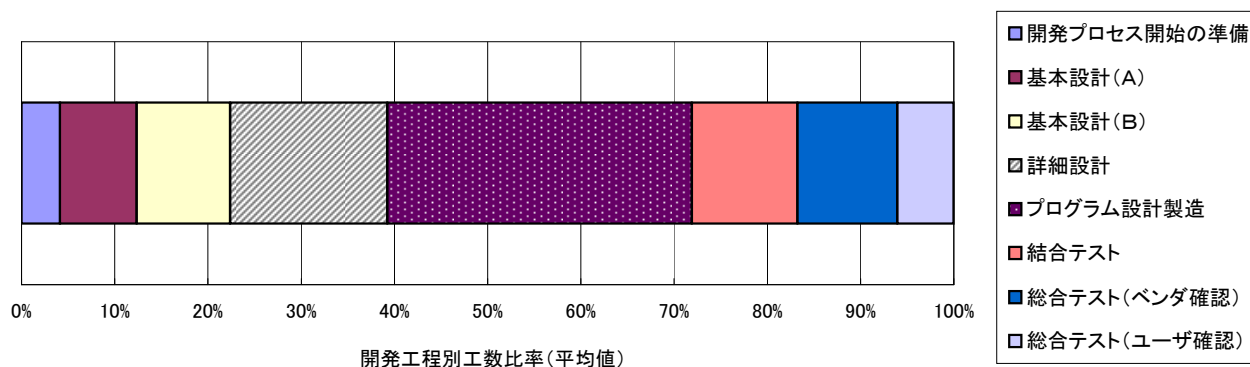
選択肢	件数	比率
初回見積ベースの記入	21	22.1%
実績ベースの記入	74	77.9%
合計	95	100.0%

(2) 開発工程別工数比率

(全工程に工数が記入された事例を集計)

(単位：%)

工程	件数	平均値	中央値	最大値	最小値
開発プロセス開始の準備	24	4.12	4.00	10.18	0.00
基本設計 (A)	24	8.24	6.67	22.22	2.00
基本設計 (B)	24	10.04	8.95	25.54	2.00
詳細設計	24	16.86	16.31	33.33	6.93
プログラム設計製造	24	32.65	33.33	60.00	11.26
結合テスト	24	11.33	11.11	22.22	2.90
総合テスト (ベンダ確認)	24	10.71	9.55	34.20	1.87
総合テスト (ユーザ確認)	24	6.05	4.27	25.00	0.00



(3) 各開発工程の契約形態

選択肢	委任件数	請負件数	委任比率	請負比率
開発プロセス開始の準備	5	30	14.3%	85.7%
基本設計 (A)	15	61	19.7%	80.3%
基本設計 (B)	11	61	15.3%	84.7%
詳細設計	5	83	5.7%	94.3%
プログラム設計製造	6	87	6.5%	93.5%
結合テスト	5	87	5.4%	94.6%
総合テスト (ベンダ確認)	9	61	12.9%	87.1%
総合テスト (ユーザ確認)	14	41	25.5%	74.5%

(4) 開発工程別職種別参画比率

(技術者の参画比率計が100%の事例を集計)

① 開発プロセス開始の準備

(単位：%)

職種	件数	平均値	中央値	最大値	最小値
プロジェクトマネージャ	45	50.24	50.00	100.00	0.00
システムエンジニア 1	45	44.46	50.00	100.00	0.00
システムエンジニア 2	45	3.97	0.00	50.00	0.00
プログラマ	45	1.33	0.00	30.00	0.00

② 基本設計 (A)

(単位：%)

職種	件数	平均値	中央値	最大値	最小値
プロジェクトマネージャ	75	26.20	18.00	100.00	0.00
システムエンジニア 1	75	57.27	60.00	100.00	0.00
システムエンジニア 2	75	13.69	0.00	100.00	0.00
プログラマ	75	2.84	0.00	100.00	0.00

③ 基本設計 (B)

(単位：%)

職種	件数	平均値	中央値	最大値	最小値
プロジェクトマネージャ	69	22.46	14.00	100.00	0.00
システムエンジニア 1	69	51.33	50.00	100.00	0.00
システムエンジニア 2	69	22.70	10.00	100.00	0.00
プログラマ	69	3.51	0.00	60.00	0.00

④ 詳細設計

(単位：%)

職種	件数	平均値	中央値	最大値	最小値
プロジェクトマネージャ	90	12.04	10.00	70.00	0.00
システムエンジニア 1	90	41.74	35.50	100.00	0.00
システムエンジニア 2	90	30.79	30.00	100.00	0.00
プログラマ	90	15.43	0.00	100.00	0.00

⑤ プログラム設計製造

(単位：%)

職種	件数	平均値	中央値	最大値	最小値
プロジェクトマネージャ	96	7.53	5.00	66.70	0.00
システムエンジニア 1	96	22.65	20.00	100.00	0.00
システムエンジニア 2	96	17.90	13.50	100.00	0.00
プログラマ	96	51.92	50.00	100.00	0.00

⑥ 結合テスト

(単位：%)

職種	件数	平均値	中央値	最大値	最小値
プロジェクトマネージャ	94	10.56	10.00	60.00	0.00
システムエンジニア 1	94	36.77	30.00	100.00	0.00
システムエンジニア 2	94	26.14	26.50	90.00	0.00
プログラマ	94	26.53	22.00	100.00	0.00

⑦ 総合テスト（ベンダ確認）

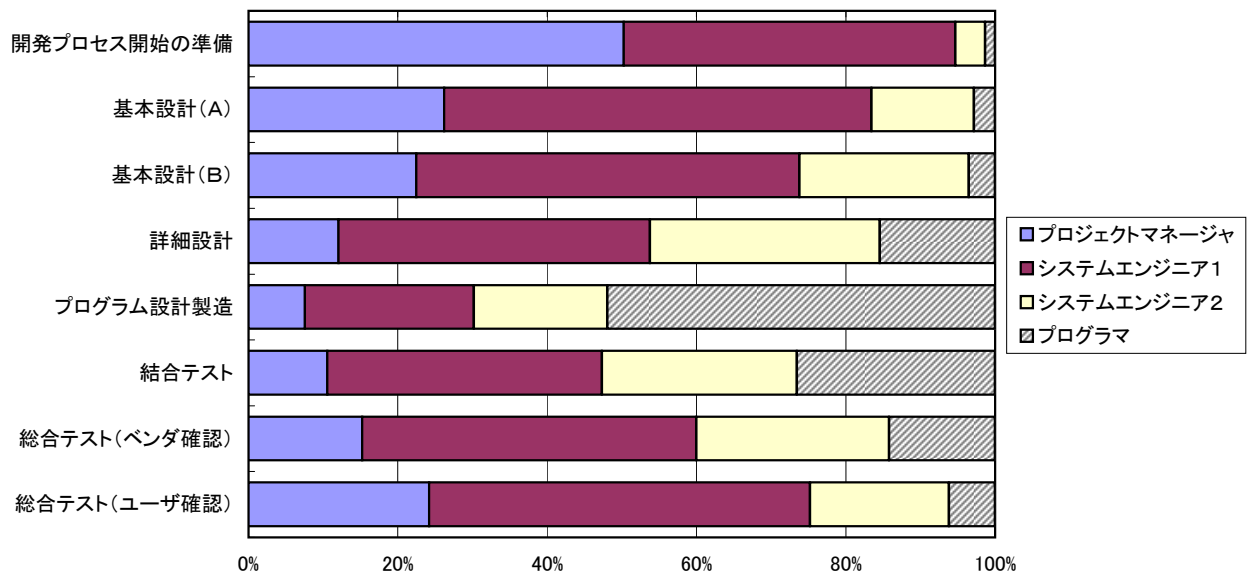
（単位：％）

職種	件数	平均値	中央値	最大値	最小値
プロジェクトマネージャ	69	15.21	10.00	80.00	0.00
システムエンジニア 1	69	44.80	40.00	100.00	0.00
システムエンジニア 2	69	25.81	25.00	85.00	0.00
プログラマ	69	14.19	0.00	100.00	0.00

⑧ 総合テスト（ユーザ確認）

（単位：％）

職種	件数	平均値	中央値	最大値	最小値
プロジェクトマネージャ	53	24.17	10.00	100.00	0.00
システムエンジニア 1	53	51.04	49.00	100.00	0.00
システムエンジニア 2	53	18.63	0.00	90.00	0.00
プログラマ	53	6.17	0.00	80.00	0.00



開発工程別職種別参画比率 (平均値)

[23] 開発事例のソフトウェア開発工程別設計書文書量

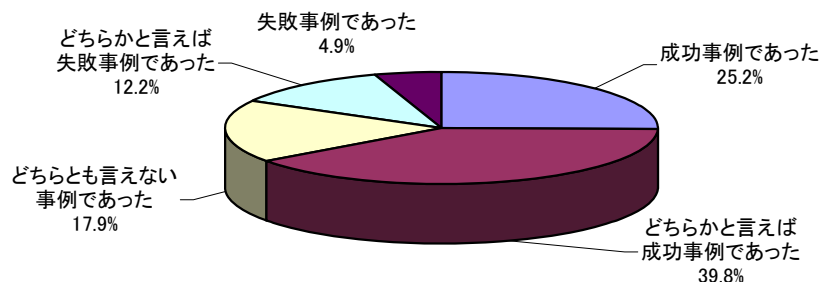
（単位：枚）

工程	件数	平均値	中央値	最大値	最小値
開発プロセス開始の準備	29	4.47	1.00	58.00	0.00
基本設計（A）	58	14.00	4.50	150.00	0.05
基本設計（B）	51	17.34	6.00	155.00	0.05
詳細設計	74	9.29	4.30	34.00	2.00
プログラム設計製造	51	23.12	6.72	200.00	0.12
結合テスト	67	41.79	15.50	315.00	0.20
総合テスト（ベンダ確認）	45	21.10	8.20	200.00	0.10
総合テスト（ユーザ確認）	34	20.72	5.00	400.00	0.10

[24] 開発事例の評価

(1) ソフトウェア開発実績事例の評価

選択肢	件数	比率
成功事例であった	31	25.2%
どちらかと言えば成功事例であった	49	39.8%
どちらとも言えない事例であった	22	17.9%
どちらかと言えば失敗事例であった	15	12.2%
失敗事例であった	6	4.9%
その他	0	0.0%
合計	123	100.0%



(2) 「成功事例であった」ケースの具体的な評価理由

納品後のトラブル発生率が低い為。
決められた予算内で納期を厳守出来た為。
要件定義工程とその他工程とで、工数/スケジュールを切り離し→工数乖離の阻止。 仕様齟齬防止を目的とした要件確認資料の作成および承認手続き→仕様/障害切り分けにおける主導権の掌 独自形式ながら、FP法のWebアプリ適用ルールができた為。
Ruby + Ruby on RailsベースのWebアプリ開発案件の自社指標ができた為。
短期間開発かつ関連システムが多岐に渡るという特性の中で、納期、品質、COSTともに問題なく、完了す ることができた為。
予定していた工数内に収まり、本稼働後のバグもほとんど発生せず安定稼働している為。
納期が予定通りとなり、高い利益を確保できた為。
利用部門の参画比率が高く、要件の整理が円滑に実施することができた為。また、システムテストについ ても十分な協力が得られ、致命的な障害も無く、品質の高いシステムとなった為。
当初のS/Iを守れた。想定したGPを確保できた。VUP時、S/Iの障害が発生しなかった為。
QCDが守られたこと。特に本番稼働後の障害が0件だった為。
要件が明確であり、その要件を計画通りの期間・コストで実現できた為。
機能に熟練した要員がアサインできたことにより、計画時点に比べ生産性が高くなった為。
社内受注判定時の収支を達成できたことにより、お客様から完成システムについて高評価された為。
極秘案件であった為、ユーザーから案件を聞く事が出来なかったが、過去の経験から作った案件がほぼ ユーザーに満足いただけるものとする事が出来た為。
本番後のトラブル発生率が低い為。
自社フレームワークと資産を利用できたことと、高スキルの要員で体制を組むことができた為。
工数、スケジュールとも予定通りであり、品質面においても問題なかった為。
実証実験を含む運用サポートまで行ったが、地域新聞等に当該成果が取り上げられた為。（発注者、関係 者との安定した協働ができた）
ユーザ検証中、毎日ヒアリングを実施し、要望に即応した為。
協力会社のマネジメント問題を早期に改善したことにより、QCDとも満足できる結果となった為。
プロジェクト計画時から客先との目標の共有、スコープやリスクに対する合意をし、仕様変更に対する対 応も柔軟に対応できた為。
品質、コスト、納期の全ての要件を満たして完了した為。
開発の多くを協力会社に請負させた事で当社はPM作業に専従でき、効率よく協力会社を活用できた為。
見積と実績の比較において、ほぼ見積時の想定通りであった為。

当該システムの開発案件では、ケータイ端末を使用したWebシステムの開発であった。今後のモバイル開発においてのノウハウの蓄積およびケータイ端末向けの開発手法、開発スキルの向上に繋がり、開発メンバーのスキル向上とモバイル開発の経験を積めた事は、非常に成功と言える事例であった為。
会社の収益に貢献し、お客様から完成システムについて高評価された為。
経理システムと連携を行うことで、レスポンス向上、ミス漏れが防止できるようになった為。
事業所毎に異なっていた管理方法が、全社一貫システムを取り入れた事により全社共通認識での業務が遂行できた為。
在庫の棚卸制度が悪いのが改善された為。
SOX法対応を取り入れた事により、セキュリティ面でも向上した為。
Excelで手作業管理していた業務をシステム管理することで、作業工数の低減が見込めた為。

(3) 「どちらかと言えば成功事例であった」ケースの具体的な評価理由

お客様から求められた納期は、機能的に全てを間に合わせることは不可能であったが、求められている機能を分割し、フェーズを3つに分けてサービスインすることにより、お客様に必要な機能を必要な時期にそれぞれ納品することができた為。
納期・コストともに満足し、且ついくつかの不具合は出たものの、プログラム改修しながら現在、システムは安定稼働しており、多くのユーザに利用して頂いている為。
ユーザ確認で障害はあったものの、エンドユーザの満足度が高かった為。
品質も高く大きなトラブルになっていない為。
納期・コスト・品質の面で顧客の満足を得られている。既存システムを引き継いだ形の開発だったが、既存の設計書とプログラムの理解が十分できていない時点で工数見積を行ったために、設計書の修正と既存プログラムのテストに予定よりも多くの時間を必要として、開発原価は計画より増加してしまった為。
QCD目標達成。客先評価は普通の為。（非常に良い訳ではない）
一次開発要員と同じメンバーで開発した為。
再利用による効率化（フレームワーク、画面、共通開数など）の為。
リリース遅延が無い為。
オフショアの利用率拡大の為。
開発メンバーのスキルアップの為。
大きな障害は発生していない為。
QCD共に、充実していた（目標に達成していた）為。
経験者の選別により、高い生産性が得られた為。
再利用の効果があった為。（フレームワーク、画面、共通開数）
本番稼働後1ヶ月間の障害発生件数が目標以下であった為。
計画費用内で開発できた為。
リリースがほぼ計画通り実現できた為。
フレームワークを作成して、再利用性を高め、生産性向上を図った為。
共通部品の再利用性を高めた為。
要員の経験者を選別した為。
Q：出荷後のバグは発生していない（成功）、C：実績工数が見積工数を超過した（失敗）、D：納期は予定通り（成功）
ほぼ予定していた期間で完了した為。
同じフレームワーク経験者が多く、生産性を高く出来た為。
基本設計の途中で規模が少し増加したが、結果として納期通りに高品質なアプリケーションを納品できた為。
リリース後のトラブルはゼロの為。
利益率が高い為。
納期が守れた為。
スケジュール面、コスト面について予定通りに進捗した為。
上流工程にお客様にしっかり入ってもらい、後工程で大きな齟齬がなかった為。
想定よりも工数が少なく完了できた為。

運用開始後の大きな不具合も少なく、現在でも多くのユーザが使用しており、安定している為。
開発期間、工程毎のスケジュール、見積工数との乖離、移動後の障害発生率はほぼ想定された範囲内であった為。
チャレンジablなシステム方式をきちんと実現したうえで工完できた為。
計画に対し工程遅延があったが、顧客との調整・合意できる範囲であった為。
工数的には+10%程度で終了した為。
システムのにもユーザークレームは少ない為。
会社としての利益も目標70%であった為。
予定工数に差異がなかった為。
導入後の品算課題等が発生していない為。
当社パッケージ基盤によるシステム開発で事前にFit&Gapである程度の要求は満たせたと感じている。導入後も大きな問題は発生しておらず、さらなる拡張へとシステム提案を行っている事は評価できる為。
本番稼働後、障害が発生してしまった事において反省点であり、マイナス要素であるが、ユーザ各社において滞りなく証券事務/運用が行えた事から概ね成功だったと判断する為。

(4) 「どちらとも言えない事例であった」ケースの具体的な評価理由

品質・工数に問題は無かったが、途中でレスポンスの問題があり、解決に時間を要した為。
納品したソフトウェアがまだ使用されておらず、フィードバックが得られていない為。
コスト的にはオーバーであった。納期・品質に関しては良好であった為。
開発ボリュームは、当初想定通りであった為。
コスト面ではオーバー。工期面では予定通り。品質面では想定内の為。
リリース日から日も浅いため、障害報告もなく成功・失敗の評価は困難の為。
オフショアを利用しコミュニケーションの難しさを痛感した点。作業期間が短くTV会議を利用し、仕様説明や外注管理を行ったが、品質の問題からスケジュール遅延等発生させた為。
コスト的には8%増の結果であったが、品質と納期面では良好であった為。
新技術を使用した今までにない開発であったので、全体的に見れば成功と思うが、汎用的なものにし、次の受注につなげるには至らなかった為。
短期間での開発で、他品目の繁忙によりプロジェクトメンバーのほとんどが他グループや協力会社の構成であった。グループ内では当たり前となっていることができていない部分が多々あり、バグにつながったお客様からの要望や変更依頼があったが、営業・開発で相談の上、良い落としどころで対応した為。

(5) 「どちらかと言えば失敗事例であった」ケースの具体的な評価理由

要件定義段階でのスコープで、抽象的な内部統制対応というものの重さが高くなっていたため、そのままシステムでの承認や点検などの実装という形になり、結果としてユーザが非常に使いにくいシステムとってしまった。既存への改修要件でデータ構造が複雑になりすぎた結果、データ項目追加やテーブル構造の見直しを行う事となり、移行などにおいて、データ移行の難易度が高くなり、困難を極めた。
ハードウェア系の開発の遅れにひきずられ、ソフトウェアの納期が遅れた為。
当初の日程に対して遅延、出荷後に市場問題発生した為。
一括請負契約で要件定義工程から総合テストまでの作業を行ったため、顧客の仕様変更とテスト工程での不具合対応が混在して行われ、結果として、当初見積もりの工数を大幅にオーバーした。
パッケージソフトを前提とした開発であったが、パッケージ適用の経験の蓄積が乏しく、要件定義が難航した為。
出荷後の障害が10件以上発生したため、今後のテスト体制の見直しにつながった。また、スケジュールの関係上、遅れが生じ、テスト期間の確保のため、今後の積み残しが発生した。工数についても、予定の15人/月から、18人/月に増えた為、予算の超過につながった。
サブプロジェクトの1つにおいて、コスト・品質の両面でスコープをいちじるしく満足できなかった為。 (その他のサブプロジェクトは許容範囲であった)
基本設計着手前に初回見積もりを実施したため、過少見積もりとなった。それに伴い開発体制の見直しを図ったが、タイミングよく要員投入ができず計画コストを超える結果となったので「4. どちらかと言えば失敗事例」としている。同種業務システムのノウハウを流用したため、実績ベースの生産性は比較的高くなっている。
フィード品質不良の為。
度重なる要件変更、追加に対する顧客への適切な対応がとれなかったために、当初見込み作業量を大幅に超越した為。
品質：本番稼働後、性能確認テストもれに起因するトラブル1件発生した。 コスト：2度にわたり計画変更を行い、変更数値どおりとなるが、初期計画から大きくブレた。 スケジュール：初期計画通りにリリースできた。これら三点から、判断した。

(6) 「失敗事例であった」ケースの具体的な評価理由

見積金額の甘さ及び極端に編重したユーザーニーズの吸収に振り回された為。
新技術に対する調査不足（工数、期間とも甘い）の為。
途中の機能の増減、スケジュール変更等顧客管理を調整できなかった為。
システム難易度に見合うスキルの技術者を投入できず、工数超過が増大した為。
プロジェクト管理工数を十分にかけず、またスキル不足もあり、マネジメントの問題に対する発見が遅れた為。
システム構築規模に見合う要件定義、基本設計の工数を十分に投入しなかったため、顧客予算内での実現範囲の判断が正しく出来ず、予算調整が間に合わなかった為。
仕様外の大量の作業が発生した為。
コストオーバーの為。

4 ソフトウェア開発に関する調査票（受託者向け）

平成 2 1 年度
ソフトウェア開発に関する調査票
(受託者向け)

平成 2 2 年 1 月
財団法人経済調査会

調査票ご記入にあたって

本調査票は、記入用紙の他に入力用Excelファイルをご用意しました。

入力用Excelファイルご要望の場合には、会社番号(調査票右上に記載の番号)、会社名、所属、氏名、E-mailアドレスを明記のうえ、経済調査会 押野(er352@zai-keicho.or.jp)までご連絡ください。

ご連絡いただいた方には、会社番号をパスワードに設定したファイルをお送りいたしますので、その後、E-mailにて上記アドレスにご返信いただくか、印刷後封入してご返信いただくようお願いいたします。

1 調査票の構成と概要

調査票は、調査票Ⅰ～Ⅲから構成されています。

① 調査票Ⅰ

・貴組織の概要に関する調査です。

② 調査票Ⅱ(品質管理部門またはそれと相当の部門の方にご記入願います)

・貴組織または貴部門におけるソフトウェア開発規模計測手法と生産性等に関する調査です。

・昨年度の調査にご協力をいただいた組織におかれましても、再度ご記入願います。

③ 調査票Ⅲ(当該プロジェクトのプロジェクトマネージャまたはそれと相当の方にご記入願います)

・貴組織または貴部門におけるソフトウェア開発実績事例に関する調査です。

・ファンクションポイント法によるソフトウェアの機能規模計測手法については、出来るだけIFPUG法で計測された事例についてのご記入をお願いいたします。また、調査票内のFPとは、未調整ファンクションポイントを指しています。

・1社につき3事例分の記入用紙またはご要望に応じて入力用Excelファイル(E-mail)をお送りいたしますので、可能な範囲で複数件のご記入をお願いいたします。

2 調査票ご記入にあたっての前提条件

調査票のご記入にあたり、下記の前提条件をご考慮願います。

① 別紙「調査票ご記入にあたっての参考資料」を参照願います。

② 調査対象システムの受発注間における契約形態は、請負契約または委任契約とします。

③ 調査対象システムは、パッケージソフトウェア(ERPパッケージなどのアプリケーションパッケージソフトウェア)を利用したソフトウェア開発は対象外とします。

④ 調査票Ⅱの調査項目は、各社における新規開発案件と改造案件を含めた平均的な値のご記入をお願いいたします。

⑤ 調査票Ⅲの調査項目は、一部を除いて実行環境では無く、開発環境を対象としています。

⑥ 調査票Ⅲの調査対象システムは、過去2～3年以内に受注された新規開発案件または改造案件を対象とします。新規開発案件とは、システムの再構築やダウンサイジングによるソフトウェア開発も含まれます。なお、システムのマイグレーションは対象外とします。

⑦ 調査票Ⅲは、組込系システムも調査の対象としています。調査票Ⅲには組込系システムに馴染まない調査項目も含まれていますが、可能な範囲でご記入願います。

⑧ 昨年度の調査にご協力をいただいた組織におかれましては、調査票Ⅲの個別のソフトウェア開発事例は、昨年度と同一の事例を対象としていただいても結構です。

3 参考情報

JFPUG(日本ファンクションポイントユーザ会) <http://www.jfpug.gr.jp/>

国立大学法人奈良先端科学技術大学院大学情報科学研究科 <http://isw3.naist.jp/home-ja.html>

共通フレーム2007 第2版(株式会社オーム社発行)

調査票Ⅰ（貴組織の概要について）

組 織 名	
所 在 地	〒
氏 名	
部 課 名	
電話番号	
E-mail	@

お名刺を添付いただいても結構です。

[1] 貴組織の立場についておたずねします。

あてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 主にソフトウェア開発を委託する立場である
- 2 主にソフトウェア開発を受託する立場である
- 3 その他(具体的な内容を下欄にご記入ください)

[2] 貴組織の企業規模についておたずねします。

① 従業員数をご記入ください。

 人

② IT系技術者数について、あてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 100人未満
- 2 100人以上～300人未満
- 3 300人以上～1,000人未満
- 4 1,000人以上

[3] 貴組織の月あたりの所定労働日数, 所定労働時間についておたずねします。

① 所定労働日数をご記入ください。

 日／月

② 所定労働時間をご記入ください。

 時間／月

調査票Ⅱ（ソフトウェア開発規模計測手法と生産性等について）

貴組織または貴部門（以下「貴組織」という）におけるソフトウェア開発規模計測手法と生産性に関する事柄について、以下におたずねします。

また、本調査票は、品質管理部門またはそれと相当の部門の方にご記入願います。

[1] 貴組織における代表的なソフトウェア開発規模計測手法についておたずねします。

下記の中から使用頻度の高い順に【 】に番号(1～3)をご記入ください。

- | | | | |
|--------------------------|-------|---|---|
| ・ファンクションポイント法(簡易的な適用を含む) | | [|] |
| ・ステップ換算法 | | [|] |
| ・その他(具体的な内容を下欄にご記入ください) | | [|] |

--

[2] 「ファンクションポイント法」を使用されている方におたずねします。

具体的な手法について、あてはまるものを下記の中からすべて選んで数字を○で囲んでください。

- 1 IFPUG法
- 2 NESMA概算法
- 3 NESMA試算法
- 4 自社オリジナル機能規模計測法
- 5 その他(具体的な内容を下欄にご記入ください)

--

[3] 項番[2]で“1”を選択された方におたずねします。

① 機能改良プロジェクトのソフトウェア規模を計測するために使用されているIFPUG法についてあてはまるものを下記の中からすべて選んで数字を○で囲んでください。

なお、EFPとAFPの定義については、別紙「調査票ご記入にあたっての参考資料」の「4 機能改良プロジェクトファンクションポイントの計測」を参照願います。

- 1 EFP(Enhancement FP count)
2 EFP(Enhancement FP count)をベースとした考え方(VAFを考慮しないなど)
3 AFP(Application FP count)
4 その他(具体的な内容を下欄にご記入ください)

--

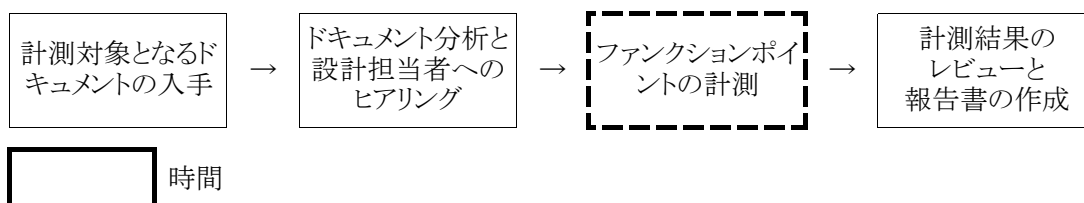
② 貴組織におけるファンクションポイントの主な計測者についてあてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 当該システムの関係者
- 2 当該システムの関係者以外の専任の担当者
- 3 その他(具体的な内容を下欄にご記入ください)

③ 1,000ファンクションポイント(FP)を計測する際に必要となる作業時間をご記入ください。

【前提条件】

ファンクションポイント計測に関連する作業の流れと範囲は下図のとおりです。
調査の対象となる作業範囲は破線内の「ファンクションポイントの計測」を指します。その前後の作業は調査の対象となる作業範囲に含みません。また、VAF(調整係数)の計測作業も含みません。



[4] 貴組織における代表的なソフトウェア開発工数(人月)と規模(FP)についておたずねします。

① ソフトウェア開発工数(人月)をご記入ください。

 ~ 人月

② ソフトウェア開発規模(FP)をご記入ください。

 ~ FP

[5] 貴組織における平均的な開発言語別生産性についておたずねします。

【前提条件】

生産性の適用範囲は、ソフトウェア開発工程における「開発プロセス開始の準備」～「総合テスト(ベンダ確認)」を想定しています。

なお、ソフトウェア開発工程の定義については、別紙「調査票ご記入にあたっての参考資料」の「1 ソフトウェア開発工程の定義」を参照願います。

① アセンブラの生産性をご記入ください。

 ~ FP/人月 ~ STEP/人月

② ASPの生産性をご記入ください。

 ~ FP/人月 ~ STEP/人月

③ ASP.NETの生産性をご記入ください。

 ~ FP/人月 ~ STEP/人月

④ Cの生産性をご記入ください。

～ FP/人月 ～ STEP/人月

⑤ C++(VC++を含む)の生産性をご記入ください。

～ FP/人月 ～ STEP/人月

⑥ COBOLの生産性をご記入ください。

～ FP/人月 ～ STEP/人月

⑦ HTMLの生産性をご記入ください。

～ FP/人月 ～ STEP/人月

⑧ Javaの生産性をご記入ください。

～ FP/人月 ～ STEP/人月

⑨ Java Scriptの生産性をご記入ください。

～ FP/人月 ～ STEP/人月

⑩ JSPの生産性をご記入ください。

～ FP/人月 ～ STEP/人月

⑪ PHPの生産性をご記入ください。

～ FP/人月 ～ STEP/人月

⑫ Rubyの生産性をご記入ください。

～ FP/人月 ～ STEP/人月

⑬ SQL(PL/SQLを含む)の生産性をご記入ください。

～ FP/人月 ～ STEP/人月

⑭ VBの生産性をご記入ください。

～ FP/人月 ～ STEP/人月

⑮ VB.NETの生産性をご記入ください。

～ FP/人月 ～ STEP/人月

⑯ VC#.NETの生産性をご記入ください。

～ FP/人月 ～ STEP/人月

⑰ XMLの生産性をご記入ください。

～ FP/人月 ～ STEP/人月

- [6] 貴組織におけるシステム開発プロセスの標準化についておたずねします。
あてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 システム開発プロセスは標準化済である
- 2 システム開発プロセスは標準化作業中または作業予定である
- 3 システム開発プロセスは標準化していない
- 4 その他(具体的な内容を下欄にご記入ください)

- [7] 項番[6]で“3”を選択された方におたずねします。
下記の中からあてはまる理由をひとつだけ選んで数字を○で囲んでください。

- 1 システム開発プロセスの標準化の必要性を感じていない
- 2 システム開発プロセスの標準化の効果を期待していない
- 3 システム開発プロセスの標準化のためのリソースが捻出できない
- 4 システム開発プロセスの標準化のためのスキルが不足している
- 5 その他（具体的な内容を下欄にご記入願います）

- [8] 貴組織におけるシステム開発プロセス標準と共通フレーム98・2007とのマッピング状況についておたずねします。
あてはまるものを下記の中からすべて選んで数字を○で囲んでください。

- 1 システム開発プロセス標準と共通フレーム98のアクティビティやタスクはマッピング済である
- 2 システム開発プロセス標準と共通フレーム2007のアクティビティやタスクはマッピング済である
- 3 システム開発プロセス標準と共通フレーム98・2007のアクティビティやタスクはマッピングしていない
- 4 その他(具体的な内容を下欄にご記入ください)

[9] 項番[8]で“3”を選択された方におたずねします。

下記の中からあてはまる理由をひとつだけ選んで数字を○で囲んでください。

- 1 マッピングの必要性を感じていない
- 2 マッピングの効果を期待していない
- 3 マッピングのためのリソースが捻出できない
- 4 マッピングのためのスキルが不足している
- 5 その他（具体的な内容を下欄にご記入願います）

--

[10] 貴組織におけるCMMについておたずねします。

① 貴組織が導入されているCMMの種類についてあてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 CMMI, CMM-I (CMM統合)
- 2 SW-CMM, CMM (ソフトウェアCMM)
- 3 SE-CMM (システムエンジニアリングCMM)
- 4 IPD-CMM(統合製品開発能力成熟度モデル)
- 5 SA-CMM(ソフトウェア調達能力成熟度モデル)

② アセスメント担当者についてあてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 SEI(カーネギーメロン大学ソフトウェア工学研究所)公認リードアセッサによる評価
- 2 SEI(カーネギーメロン大学ソフトウェア工学研究所)公認リードアセッサ以外による評価

③ 達成レベルについてあてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 レベル1 または達成していない
- 2 レベル2
- 3 レベル3
- 4 レベル4
- 5 レベル5

[11] 貴組織におけるオフショア開発の実績についておたずねします。

① オフショア開発の実績についてあてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 オフショア開発の実績がある
- 2 オフショア開発の実績はない

② 項番 [11]①で“1”を選択された方にオフショア開発先についておたずねします。
オフショア開発先の国名・都市名をご記入ください。

国名	都市名

③ 項番 [11]①で“1”を選択された方にオフショア開発の委託範囲についておたずねします。
オフショア開発として委託したソフトウェア開発工程についてあてはまるものを下記の中からすべて選んで数字を○で囲んでください。

なお、ソフトウェア開発工程の定義については、別紙「調査票ご記入にあたっての参考資料」の「1 ソフトウェア開発工程の定義」を参照願います。

- 1 開発プロセス開始の準備
- 2 基本設計(A)・・・システム要件定義, ソフトウェア要件定義
- 3 基本設計(B)・・・システム方式設計, ソフトウェア方式設計
- 4 詳細設計・・・ソフトウェア詳細設計
- 5 プログラム設計製造
- 6 結合テスト・・・ソフトウェア結合, システム結合
- 7 総合テスト(ベンダ確認)・・・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
- 8 総合テスト(ユーザ確認)・・・ソフトウェア導入, ソフトウェア受入れ支援
- 9 その他(具体的な内容を下欄にご記入ください)

--

④ 項番 [11]①で“1”を選択された方にオフショア開発の貴組織の管理工数についておたずねします。

オフショア開発の委託工数に占める貴組織の平均的な管理工数の比率をご記入ください。

	%
--	---

⑤ 項番 [11]①で“1”を選択された方にオフショア開発の工数比率についておたずねします。

貴組織における国内開発とオフショア開発の工数比率をご記入ください。

国内開発工数(自社)		%
国内開発工数(外部委託)		%
オフショア開発工数		%
合計	100	%

調査票Ⅲ(個別のソフトウェア開発事例について)

貴組織または貴部門(以下「貴組織」という)のソフトウェア開発実績事例に関する事柄について、以下におたずねします。

なお、本調査票は、一部を除き、ソフトウェア開発終了後の実績ベースでご記入願います。

また、本調査票は、当該プロジェクトのプロジェクトマネージャまたはそれと相当の方にご記入願います。

[1] 当該システムが新規開発か改造(機能の追加・変更・削除)かについておたずねします。

あてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 新規開発案件
- 2 改造(機能の追加・変更・削除)案件

[2] 当該システムのソフトウェア開発実績事例についておたずねします。

あてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 ソフトウェア開発実績事例は、昨年度の調査票Ⅲ記入事例と異なる事例
- 2 ソフトウェア開発実績事例は、昨年度の調査票Ⅲ記入事例と同一の事例

[3] 当該システムの委託者の分類についておたずねします。

あてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 府省庁
- 2 独立行政法人
- 3 地方自治体
- 4 民間企業(親会社や関連会社)
- 5 民間企業(親会社や関連会社以外)
- 6 その他(具体的な内容を下欄にご記入ください)

--

[4] 当該システムの実行環境における適用分野・業種についておたずねします。

① 適用分野についてあてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 事務系
- 2 制御系
- 3 その他(具体的な適用分野を下欄にご記入ください)

--

② 適用業種についてあてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 建設業
- 2 製造業
- 3 電気・ガス・熱供給・水道業
- 4 情報通信業
- 5 運輸業
- 6 卸売・小売業
- 7 金融・保険業
- 8 不動産業
- 9 飲食店, 宿泊業
- 10 医療, 福祉
- 11 教育, 学習支援業
- 12 公務
- 13 その他(具体的な適用業種を下欄にご記入ください)

--

[5] 当該システムのソフトウェア開発工程範囲についておたずねします。

該当する工程についてあてはまるものを下記の中からすべて選んで数字を○で囲んでください。
なお, ソフトウェア開発工程の定義については, 別紙「調査票ご記入にあたっての参考資料」の「1 ソフトウェア開発工程の定義」を参照願います。

- 1 開発プロセス開始の準備
- 2 基本設計(A)・・・システム要件定義, ソフトウェア要件定義
- 3 基本設計(B)・・・システム方式設計, ソフトウェア方式設計
- 4 詳細設計・・・ソフトウェア詳細設計
- 5 プログラム設計製造
- 6 結合テスト・・・ソフトウェア結合, システム結合
- 7 総合テスト(ベンダ確認)・・・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
- 8 総合テスト(ユーザ確認)・・・ソフトウェア導入, ソフトウェア受入れ支援
- 9 その他(具体的な内容を下欄にご記入ください)

--

[6] 当該システムのシステム構成についておたずねします。

あてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 クライアントサーバシステム
- 2 Web系システム
- 3 メインフレームシステム
- 4 組込系システム
- 5 その他(具体的なシステム構成を下欄にご記入ください)

--

[7] 当該システムのOSについておたずねします。

項番[6]で“1”の「クライアントサーバシステム」を選択された方は①と②について, “2”の「Web系システム」を選択された方は②について, “3”の「メインフレームシステム」を選択された方は③について, “4”の「組込系システム」を選択された方は④についてご記入ください。

① クライアントのOSについてあてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 Windows 2000
- 2 Windows XP
- 3 Windows Vista
- 4 その他(具体的な名称を下欄にご記入ください)

--

② サーバのOSについてあてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 UNIX
- 2 Linux
- 3 Windows 2000 系
- 4 その他(具体的な名称を下欄にご記入ください)

--

③ メインフレームのOS名称を下欄にご記入ください。

--

④ 組込系システムのOS名称を下欄にご記入ください。

--

[8] 当該システムのデータベースについておたずねします。

データベースについてあてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 Microsoft SQL Server
- 2 Oracle
- 3 その他(オープンソース系:具体的な名称を下欄にご記入ください)

--

- 4 その他(非オープンソース系:具体的な名称を下欄にご記入ください)

--

[9] 当該システムの開発言語(又は開発ツール)についておたずねします。

① 使用した開発言語(又は開発ツール)についてあてはまるものを下記の中からすべて選んで数字を○で囲んでください。

- 1 アセンブラ
- 2 ASP
- 3 ASP.NET
- 4 C
- 5 C++(VC++含む)
- 6 COBOL
- 7 HTML
- 8 Java
- 9 Java Script
- 10 JSP
- 11 PHP
- 12 Ruby
- 13 SQL(PL/SQLを含む)
- 14 VB
- 15 VB.NET
- 16 VC#.NET
- 17 XML
- 18 その他(オープンソース系:具体的な名称を下欄にご記入ください)

--

- 19 その他(非オープンソース系:具体的な名称を下欄にご記入ください)

--

② 項番「9」①で複数の開発言語を選択された方は、代表的な開発言語名とその言語のソフトウェア機能量比率をご記入ください。

なお、ソフトウェア機能量比率を把握されていない場合は、大まかな値で結構です。

開発言語名	機能量比率
	%
	%
	%
	%
合計	100 %

[10] 当該システムの開発方法論などについておたずねします。

① プロセスモデルについてあてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 ウォーターフォール
- 2 繰り返し型プロセス
- 3 その他(具体的な内容を下欄にご記入ください)

--

② 開発技法についてあてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 構造化
- 2 DOA (Data Oriented Approach)
- 3 オブジェクト指向
- 4 その他(具体的な内容を下欄にご記入ください)

--

③ システム開発標準についてあてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 使用した
- 2 使用しなかった

④ 項番「10」③で”1”を選択された方にシステム開発標準の入手方法についておたずねします。

あてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 自社開発
- 2 市販品(具体的なメーカー名と製品名を下欄にご記入ください)

メーカー名	製品名

[11] 当該システムのフレームワークの使用状況についておたずねします。

① フレームワークについてあてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 使用した
- 2 使用しなかった

② 項番 [11]①で“1”を選択された方にフレームワークの入手方法についておたずねします。

あてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 自社開発
- 2 市販品(具体的なメーカー名と製品名を下欄にご記入ください)

メーカー名	製品名

- 3 その他(オープンソース系:具体的な名称を下欄にご記入ください)

--

- 4 その他(非オープンソース系:具体的な名称を下欄にご記入ください)

--

[12] 当該システムのパッケージソフトウェアについておたずねします。

① パッケージソフトウェアの使用状況についてあてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 使用した
- 2 使用しなかった

② 項番 [12]①で“1”を選択された方は具体的なメーカー名と製品名をご記入ください。

メーカー名	製品名

[13] 当該システムのソフトウェアの費用についておたずねします。

① 初回見積ベースのソフトウェア費用をご記入ください。

百万円（税抜き）

② 実績ベースのソフトウェア費用をご記入ください。

百万円（税抜き）

[14] 当該システムのソフトウェア開発の期間についておたずねします。

① 初回見積ベースの開発期間をご記入ください。

ヶ月

② 実績ベースの開発期間をご記入ください。

ヶ月

[15] 当該システムのソフトウェア開発工数(人月)とソフトウェア開発規模(FPとSTEP)についておたずねします。

① 実績ベースのソフトウェア開発工数(人月)とソフトウェア開発規模(FP)をご記入ください。

ソフトウェア開発工数 人月

ソフトウェア開発規模 FP

K STEP

ILF	<input type="text"/>	FP	<input type="text"/>	件
EIF	<input type="text"/>	FP	<input type="text"/>	件
E I	<input type="text"/>	FP	<input type="text"/>	件
EO	<input type="text"/>	FP	<input type="text"/>	件
EQ	<input type="text"/>	FP	<input type="text"/>	件

② 項番[1]で“2”の改造案件を選択された方におたずねします。
実績ベースでのEFPとAFPおよびその内訳をご記入ください。なお、EFPやAFPの定義については、別紙「調査票ご記入にあたっての参考資料」の「4 機能改良プロジェクトファンクションポイントの計測」を参照願います。

EFP	<input type="text"/>	FP
AFP	<input type="text"/>	FP

ADD	<input type="text"/>	FP
CHGA	<input type="text"/>	FP
CFP	<input type="text"/>	FP
DEL	<input type="text"/>	FP
UFPB	<input type="text"/>	FP
CHGB	<input type="text"/>	FP

③ 項番[1]で“2”の改造案件を選択された方におたずねします。
実績ベースでの改造STEPの内訳をご記入ください。改造STEPの内訳とは、下図のとおり「A:母体(非改造部分)」「B:削除(改造部分)」「C:改造(改造部分)」「D:追加」に該当する機能のSTEPを指します。

A: 母体 (非改造部分)		D: 追加 (改造部分)
B: 削除 (改造部分)	C: 改造 (改造部分)	

A: 母体(非改造部分)	<input type="text"/>	K STEP
B: 削除(改造部分)	<input type="text"/>	K STEP
C: 改造(改造部分)	<input type="text"/>	K STEP
D: 追加(改造部分)	<input type="text"/>	K STEP

④ 初回見積ベースのソフトウェア開発工数(人月)とソフトウェア開発規模(FP)をご記入ください。

ソフトウェア開発工数	<input type="text"/>	人月
ソフトウェア開発規模	<input type="text"/>	FP

⑤ 項番[15]④でソフトウェア開発工数(人月)についてご記入された方におたずねします。

初回見積ベースのソフトウェア開発工数(人月)の算出方法についてあてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 ワーク・ブレイクダウン・ストラクチャ法(WBS法)
- 2 類似システム参照法
- 3 その他(具体的な内容を下欄にご記入ください)

--

⑥ 項番[15]④でソフトウェア開発規模(FP)についてご記入された方におたずねします。

初回見積ベースのソフトウェア開発規模(FP)の計測タイミングについてあてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 基本設計着手前
- 2 基本設計完了後
- 3 その他(時期を下欄にご記入ください)

--

⑦ 項番[15]①②④でソフトウェア開発規模(FP)をご記入された方におたずねします。

ファンクションポイントの具体的な手法についてあてはまるものを下記の中からすべて選んで数字を○で囲んでください。

- 1 IFPUG法
- 2 NESMA概算法
- 3 NESMA試算法
- 4 自社オリジナル機能規模計測法
- 5 その他(具体的な内容を下欄にご記入願います)

--

⑧ 項番[15]①②④でソフトウェア開発規模(FP)をご記入された方におたずねします。

ファンクションポイントの主な計測者についてあてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

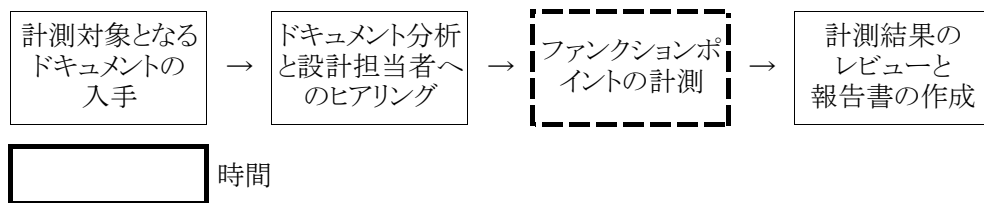
- 1 当該システムの関係者
- 2 当該システムの関係者以外の専任の担当者
- 3 その他(具体的な内容を下欄にご記入ください)

--

⑨ 項番[15]①②④でソフトウェア開発規模(FP)をご記入された方におたずねします。
ソフトウェア開発規模(FP)の計測作業時間をご記入ください。

【前提条件】

ファンクションポイント計測に関連する作業の流れと範囲は下図のとおりです。
調査の対象となる作業範囲は破線内の「ファンクションポイントの計測」を指します。その前後の作業は調査の対象となる作業範囲に含みません。また、VAF(調整係数)の計測作業も含みません。



[16] 当該システムの画面数・帳票数・ファイル数・バッチプログラム数についておたずねします。
画面数・帳票数・ファイル(またはテーブル)数・バッチプログラム数をご記入ください。

画面数		件	ファイル数		件
帳票数		件	バッチプログラム数		件

[17] 当該システムのユースケースについておたずねします。
ユースケース数・アクター数をご記入ください。

ユースケース数		件	アクター数		件
---------	--	---	-------	--	---

[18] 当該システムの品質管理についておたずねします。

① 各工程における内部レビュー回数と指摘件数をご記入ください。

	設計レビュー	コードレビュー	テストレビュー	
レビュー回数				回
レビュー指摘件数				件

② 総合テスト(ベンダ確認)工程において発生したバグ件数をご記入ください。

なお、ソフトウェア開発工程の定義については、別紙「調査票ご記入にあたっての参考資料」の「1 ソフトウェア開発工程の定義」を参照願います。

③ 出荷後から3ヶ月以内に発生したバグ件数をご記入ください。

なお、出荷後から3ヶ月以内のバグ件数を把握されていない場合には、把握されている月数と出荷後からその月数以内に発生したバグ件数をご記入ください。

[19] 当該システムの最大開発要員(ピーク時の要員)についておたずねします。

① 最大開発要員数をご記入ください。

人

② 最大開発要員数に該当するソフトウェア開発工程についてあてはまるものを下記の中からすべて選んで数字を○で囲んでください。

なお、ソフトウェア開発工程の定義については、別紙「調査票ご記入にあたっての参考資料」の「1 ソフトウェア開発工程の定義」を参照願います。

- 1 開発プロセス開始の準備
- 2 基本設計(A)・・・システム要件定義, ソフトウェア要件定義
- 3 基本設計(B)・・・システム方式設計, ソフトウェア方式設計
- 4 詳細設計・・・ソフトウェア詳細設計
- 5 プログラム設計製造
- 6 結合テスト・・・ソフトウェア結合, システム結合
- 7 総合テスト(ベンダ確認)・・・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
- 8 総合テスト(ユーザ確認)・・・ソフトウェア導入, ソフトウェア受入れ支援
- 9 その他(具体的な内容を下欄にご記入ください)

[20] 当該システムのリスク管理についておたずねします。

① 見積ベース(受注前)で当該システムのリスク管理を行った場合には、その具体的な内容を下欄にご記入ください。

② 実績ベース(受注後)で当該システムのリスク管理を行った場合には、その具体的な内容を下欄にご記入ください。

[21] 当該システムの生産性変動要因についておたずねします。

生産性変動要因は、発注者側の要因と受注者側の要因に分かれています。

なお、生産性変動要因の定義と影響度の評価については、別紙「調査票ご記入にあたっての参考資料」の「2 生産性変動要因の定義」を参照願います。

① ソフトウェア開発の生産性に影響を与えたと考えられる「発注者の要因」について、下表の選択肢(1～5)の中からあてはまるものをそれぞれひとつだけ選んで数字を○で囲んでください。

生産性 変動要因		選択肢				
発注者の要因	機能性	1	2	3	4	5
		全体的に過度に要求された	部分的に過度に要求された	適度に要求された	部分的に要求された	要求されなかった
	信頼性	1	2	3	4	5
		ソフトウェア障害は人命または社会基盤にかかわる損失が想定された	ソフトウェア障害は財政上の大規模な損失が想定された	ソフトウェア障害は復旧可能な中規模の損失が想定された	ソフトウェア障害は簡単に復旧可能な小規模な損失が想定された	ソフトウェア障害は軽微な損失が想定された
	プラットフォーム	1	2	3	4	5
		全体的に適合性に欠けた	部分的に適合性に欠けた	適合性があった	かなり適合性があった	非常に適合性があった
	開発ル ス要 求 ジ ュ ー	1	2	3	4	5
		スケジュールは当初の想定より25%以上短縮された	スケジュールは当初の想定より10%～25%前後短縮された	スケジュールは当初の想定どおりであった	スケジュールは当初の想定より10%～50%前後許容された	スケジュールは当初の想定より50%以上許容された
	発注・要件 安定の 度明確	1	2	3	4	5
		全体的に不明確・不安定であった	部分的に不明確・不安定であった	明確で安定していた	非常に明確で安定していた	先行モデルがあり非常に明確で常に安定していた
	参画割合	1	2	3	4	5
		全く参画しなかった	一部参画した	主要な工程について適度に参画した	全工程について適度に参画した	全工程について適切な担当者が適度に参画した

② ソフトウェア開発の生産性に影響を与えたと考えられる「受注者(開発者)の要因」について、下表の選択肢(1～5)の中からあてはまるものをそれぞれひとつだけ選んで数字を○で囲んでください。

生産性 変動要因		選択肢				
受注者（開発者）の要因	標準モデルの 先行モデル の採用	1	2	3	4	5
		先行モデルは存在しなかった または 適合する標準モデルは存在しなかった	先行モデルは存在したがほとんど 流用出来なかった または 適合する標準モデルは存在したが採用しなかった	先行モデルは部分的に流用出来た または 適合する標準モデルは部分的に採用出来た	先行モデルはかなり流用出来た または 適合する標準モデルはかなり採用出来た	先行モデルは全面的に流用出来た または 適合する標準モデルは全面的に採用出来た
	管理者の プロジェクト 経験と能力	1	2	3	4	5
		経験無かった	少数の小中規模プロジェクトの管理を経験していた	多数の小中規模プロジェクトの管理を経験していた	少数の中大規模プロジェクトの管理を経験していた	多数の中大規模プロジェクトの管理を経験していた
	アナリスト の経験と能力	1	2	3	4	5
		経験無かった	少数の小中規模プロジェクトのアナリストを経験していた	多数の小中規模プロジェクトのアナリストを経験していた	少数の中大規模プロジェクトのアナリストを経験していた	多数の中大規模プロジェクトのアナリストを経験していた
	SE・プログラマ の経験と能力	1	2	3	4	5
		要員の過半数は経験無かった	半数はある程度経験していた	過半数はある程度経験していた	半数は十分な経験残り半数はある程度経験していた	過半数は十分な経験をしていた

③ 項番[21]①②で定義した生産性変動要因以外に重要と考えられる要因があれば下欄にご記入ください。

[22] 当該システムのソフトウェア開発工程別工数と職種別参画比率についておたずねします。

ソフトウェア開発工程と職種の定義については、別紙「調査票ご記入にあたっての参考資料」の「1 ソフトウェア開発工程の定義」と「3 職種区分の定義」を参照願います。

① 項番[22]の②③について、次の中からあてはまるものをひとつだけ選んで数字を○で囲んでください。

- 1 初回見積ベースで記入
- 2 実績ベースで記入

② 各ソフトウェア開発工程の工数(人月)をご記入ください。また、各ソフトウェア開発工程の契約形態を「委任」か「請負」からあてはまるものをひとつだけ選んで○で囲んでください。

工 程	開発プロセス 開始の準備	基本設計		詳細設計	プログラム 設計製造	結合テスト	総合テスト (ベンダ確認)	総合テスト (ユーザ確認)
		(A)	(B)					
工数(人月)								
契約形態	委任・請負	委任・請負	委任・請負	委任・請負	委任・請負	委任・請負	委任・請負	委任・請負

③ 各ソフトウェア開発工程における職種別参画比率をご記入ください。

工 程 職 種	開発プロセス 開始の準備	基本設計		詳細設計	プログラム 設計製造	結合テスト	総合テスト (ベンダ確認)	総合テスト (ユーザ確認)
		(A)	(B)					
PM								
SE1								
SE2								
PG								
合 計	100%	100%	100%	100%	100%	100%	100%	100%

[23] 当該システムのソフトウェア開発工程別の設計書文書量についておたずねします。

各ソフトウェア開発工程で作成した設計書の文書量(枚数)をご記入ください。なお、文書はA4サイズを想定した大まかな値で結構です。

工 程	開発プロセス 開始の準備	基本設計		詳細設計	プログラム 設計製造	結合テスト	総合テスト (ベンダ確認)	総合テスト (ユーザ確認)
		(A)	(B)					
設計書(枚)								

[24] 当該システムのソフトウェア開発実績事例の評価(成功事例・失敗事例)についておたずねします。

① あてはまるものを下記の中からひとつだけ選んで数字を○で囲んでください。

- 1 成功事例であった
- 2 どちらかと言えば成功事例であった
- 3 どちらとも言えない事例であった
- 4 どちらかと言えば失敗事例であった
- 5 失敗事例であった
- 6 その他

② 項番[24]①で選択された評価の理由について、その具体的な内容を下欄にご記入ください。

最後に、本調査票に対するご意見やソフトウェア開発費の積算、ソフトウェア開発規模計測等に関するご意見等ありましたら下欄にご記入願います。

ご協力ありがとうございました。

調査票ご記入にあたっての参考資料

調査票のご記入にあたり、下記１～４の定義を考慮の上ご記入願います。

１ ソフトウェア開発工程の定義

本定義は、ソフトウェア開発の工程と共通フレーム2007のアクティビティとタスクを対比させることを目的としています。

ソフトウェア開発に際して、ソフトウェア開発の工程範囲や作業内容が曖昧なため、委託者と受託者間で認識がくい違う場合があります。

本定義は、調査票Ⅱ・Ⅲのソフトウェア開発工程に関するご記入の際に、貴組織におけるソフトウェア開発工程の定義と比較してご利用ください。なお、太線内の項目が本調査の調査対象範囲です。

工程	共通フレーム2007の定義
システム化計画	1.4.3:システム化計画の立案 1.4.3.1:システム化計画の基本要件の確認 1.4.3.2:対象業務の内容確認 1.4.3.3:対象業務のシステム課題の定義 1.4.3.4:対象システムの分析 1.4.3.5:適用情報技術の調査 1.4.3.6:業務モデルの作成 1.4.3.7:システム化機能の整理とシステム方式の策定 1.4.3.8:システム化に必要な付帯機能,付帯設備に対する基本方針の明確化 1.4.3.9:サービスレベルと品質に対する基本方針の明確化 1.4.3.10:プロジェクトの目標設定 1.4.3.11:実現可能性の検討 1.4.3.12:全体開発スケジュールの作成 1.4.3.13:システム選定方針の策定 1.4.3.14:費用とシステム投資効果の予測 1.4.3.15:プロジェクト推進体制の策定 1.4.3.16:経営事業戦略,情報戦略,システム化構想との検証 1.4.3.17:システム化計画の作成と承認 1.4.3.18:プロジェクト計画の作成と承認
要件定義	1.5.2:利害関係者要件の定義 1.5.2.1:利害関係者のニーズの識別と制約事項の定義 1.5.2.2:業務要件の定義 1.5.2.3:組織及び業務環境要件の具体化 1.5.2.4:機能要件の定義 1.5.2.5:非機能要件の定義 1.5.2.6:スケジュールに関する要件の定義 1.5.2.7:実現可能性とリスクの検討 1.5.3:利害関係者要件の確認 1.5.3.1:要件の合意と承認 1.5.3.2:要件変更のルール決定

工程		共通フレーム2007の定義
開発プロセス 開始の準備		1.6.1:プロセス開始の準備 1.6.1.1:開発作業の組立て 1.6.1.2:必要な支援プロセスの組込み 1.6.1.3:開発環境の準備 1.6.1.4:開発プロセス実施計画の策定 1.6.1.5:非納品品目の使用の容認
基本設計	(A)	1.6.2:システム要件定義 1.6.2.1:システム要件の定義 1.6.2.2:システム要件の評価 1.6.2.3:システム要件の共同レビューの実施 1.6.4:ソフトウェア要件定義 1.6.4.1:ソフトウェア要件の確立 1.6.4.2:ソフトウェア要件の評価 1.6.4.3:ソフトウェア要件の共同レビューの実施
	(B)	1.6.3:システム方式設計 1.6.3.1:システムの最上位レベルでの方式確立 1.6.3.2:利用者文書(暫定版)の作成 1.6.3.3:システム結合のためのテスト要求事項の定義 1.6.3.4:システム方式の評価 1.6.3.5:システム方式設計の共同レビューの実施 1.6.5:ソフトウェア方式設計 1.6.5.1:ソフトウェア構造とコンポーネントの方式設計 1.6.5.2:外部,コンポーネント間の各インタフェースの方式設計 1.6.5.3:データベースの最上位レベルの設計 1.6.5.4:利用者文書(暫定版)の作成 1.6.5.5:ソフトウェア結合のためのテスト要求事項の定義 1.6.5.6:ソフトウェア方式設計の評価 1.6.5.7:ソフトウェア方式設計の共同レビューの実施
詳細設計		1.6.6:ソフトウェア詳細設計 1.6.6.1:ソフトウェアコンポーネントの詳細設計 1.6.6.2:ソフトウェアインタフェースの詳細設計 1.6.6.3:データベースの詳細設計 1.6.6.4:利用者文書の更新 1.6.6.5:ソフトウェアユニットのテスト要求事項の定義 1.6.6.6:ソフトウェア結合のためのテスト要求事項の更新 1.6.6.7:ソフトウェア詳細設計及びテスト要求事項の評価 1.6.6.8:ソフトウェア詳細設計の共同レビューの実施

工程	共通フレーム2007の定義
プログラム 設計製造	1.6.7:ソフトウェアコード作成及びテスト 1.6.7.1:ソフトウェアユニットとデータベースの作成及びテスト手順とテストデータの作成 1.6.7.2:ソフトウェアユニットとデータベースのテストの実施 1.6.7.3:利用者文書の更新 1.6.7.4:ソフトウェア結合テスト要求事項の更新 1.6.7.5:ソフトウェアコード及びテスト結果の評価
結合テスト	1.6.8:ソフトウェア結合 1.6.8.1:ソフトウェア結合計画の作成 1.6.8.2:ソフトウェア結合テストの実施 1.6.8.3:利用者文書の更新 1.6.8.4:ソフトウェア適格性確認テストの準備 1.6.8.5:ソフトウェア結合テストの評価 1.6.8.6:ソフトウェア結合の共同レビュー実施 1.6.10:システム結合 1.6.10.1:システム結合計画の作成 1.6.10.2:システム結合テストの実施 1.6.10.3:利用者文書の更新 1.6.10.4:システム適格性確認テストの準備 1.6.10.5:システム結合テストの評価 1.6.10.6:システム結合の共同レビュー実施
総合テスト (ベンダ確認)	1.6.9:ソフトウェア適格性確認テスト 1.6.9.1:ソフトウェア適格性確認テストの実施 1.6.9.2:利用者文書の更新 1.6.9.3:ソフトウェア適格性確認テストの評価 1.6.9.4:ソフトウェア適格性確認テストの共同レビューの実施 1.6.9.5:監査の支援 1.6.9.6:納入ソフトウェア製品の準備 1.6.11:システム適格性確認テスト 1.6.11.1:システム適格性確認テストの実施 1.6.11.2:システムの評価 1.6.11.3:システム適格性確認テストの共同レビューの実施 1.6.11.4:利用者文書の更新 1.6.11.5:監査の支援 1.6.11.6:各納入ソフトウェア製品の準備 1.6.11.7:運用、保守に引き継ぐソフトウェア製品の準備
総合テスト (ユーザ確認)	1.6.12:ソフトウェア導入 1.6.12.1:ソフトウェア導入(インストール)の計画の作成 1.6.12.2:ソフトウェア導入の実施 1.6.13:ソフトウェア受入れ支援 1.6.13.1:取得者の受入れレビューと受入れテストの支援 1.6.13.2:ソフトウェア製品の納入 1.6.13.3:取得者への教育訓練及び支援

2 生産性変動要因の定義

本定義は、ソフトウェア開発の生産性を変動すると考えられる要因とその程度を明確にすることを目的としています。

ソフトウェア開発においては、様々な要因が生産性を大きく左右することが確認されていますが、それらの要因を一般化する手段として、次のとおり“委託者の要因”と“受託者の要因”に分類定義しました。

本定義は、調査票Ⅲの[21]をご記入の際にご利用ください。

生産性 変動要因		生産性変動要因の説明	選択肢				
委託者の 要因	機能性	機能性とは、下記の要件を指す。 ・難易度の高い機能(合目的性) ・精密性の高い計算(正確性) ・他システムとの接続(相互運用性) ・税法、OSI(Open Systems Interconnection)規格等の公的規則や公的標準、社内規則や社内標準等(標準適合性) ・機密保護やアクセス管理(セキュリティ)	全体的に 過度に要求された	部分的に 過度に要求された	適度に要求された	部分的に 要求された	要求されなかった
	信頼性	信頼性とは、下記の要件を指す。 ・故障発生率(成熟性) ・システムダウン(障害許容性) ・システムダウンからの再開・回復時間(回復性)	ソフトウェア障害は 人命にかかわる 損失が想定された	ソフトウェア障害は 財政上の大規模な 損失が想定された	ソフトウェア障害は 復旧可能な中規模の 損失が想定された	ソフトウェア障害は 簡単に復旧可能な 小規模な損失が 想定された	ソフトウェア障害は 軽微な損失が 想定された
	プラットフォーム	プラットフォームとは、ハードウェアとソフトウェア(OSやDBMSなど)の複合体であり、その適合性(ニーズ、性能など)を指す。	全体的に 適合性に欠けた	部分的に 適合性に欠けた	適合性があった	かなり 適合性があった	非常に 適合性があった
	開発スケジュール 要求	開発スケジュール要求とは、開発スケジュールの制約度合を指す。	スケジュールは 当初の想定より 25%以上 短縮された	スケジュールは 当初の想定より 10%～25%前後 短縮された	スケジュールは 当初の想定 どおりであった	スケジュールは 当初の想定より 10%～50%前後 許容された	スケジュールは 当初の想定より 50%以上 許容された
	発注要件の明確 度と安定度	発注要件の明確度とは、発注時点における発注仕様書の明確度合を指す。 発注要件の安定度とは、ソフトウェア開発期間中に発注者の要因により仕様変更が発生する度合を指す。	全体的に 不明確・不安定 であった	部分的に 不明確・不安定 であった	明確で 安定していた	非常に 明確で 安定していた	先行モデルがあり 非常に明確で 常に安定していた
	ユーザの参画割合	ユーザの参画割合とは、発注者の関与度合を指す。	全く 参画しなかった	一部 参画した	主要な工程について 適度に参画した	全工程について 適度に参画した	全工程について 適切な担当者が 適度に参画した

生産性 変動要因		生産性変動要因の説明	選択肢				
受託者の 要因	先行モデルの流 用と標準モデルの 採用	先行モデルの流用とは、対象業務のシステム 化に開発済みの類似システムを流用出来る 度合を指す。 標準モデルの採用とは、ミドルウェアやフレ ームワークの採用度合いを指す。	先行モデルは 存在しなかった または 適合する標準モデル は存在しなかった	先行モデルは 存在したが ほとんど 流用出来なかった または 適合する標準モデル は存在したが 採用しなかった	先行モデルは 部分的に流用出来た または 適合する標準モデル は部分的に 採用出来た	先行モデルは かなり流用出来た または 適合する標準モデル はかなり 採用出来た	先行モデルは 全面的に流用出来た または 適合する標準モデル は全面的に 採用出来た
	プロジェクト管理 者の経験と能力	プロジェクト管理者の経験と能力とは、プロ ジェクト管理者に求められる経験と能力の度 合を指す。	経験無かった	少数の小中規模 プロジェクトの管理を 経験していた	多数の小中規模 プロジェクトの管理を 経験していた	少数の中大規模 プロジェクトの管理を 経験していた	多数の中大規模 プロジェクトの管理を 経験していた
	アナリストの経験と 能力	アナリストの経験と能力とは、アナリストに求め られる経験と能力の度合(業務経験含む)を指 す。	経験無かった	少数の小中規模 プロジェクトの アナリストを 経験していた	多数の小中規模 プロジェクトの アナリストを 経験していた	少数の中大規模 プロジェクトの アナリストを 経験していた	多数の中大規模 プロジェクトの アナリストを 経験していた
	SE・プログラマの 経験と能力	SE・プログラマの経験と能力とは、下記の要 件を指す。 ・業務の経験と能力の度合 ・開発技法の経験と能力の度合 ・開発方法論の経験と能力の度合 ・プラットフォームの経験と能力の度合 ・言語とツールの経験と能力の度合	要員の過半数は 経験無かった	半数はある程度 経験していた	過半数はある程度 経験していた	半数は十分な経験 残り半数はある程度 経験していた	過半数は 十分な経験をしていた

3 職種区分の定義

本定義は、ソフトウェア開発技術者について、職種の区分とその役割を明確にすることを目的としています。
 なお、付記した主担当工程はその技術者が主となることを表し、他の技術者の参画を排除するものではありません。
 本定義は、調査票Ⅲの[22]をご記入の際にご利用ください。

職種区分	職種の役割	主担当工程
プロジェクト マネージャ (PM)	<ul style="list-style-type: none"> ・システム開発計画の全体構想、プロジェクト体制の構築及び後工程のプロジェクト管理指標決定 ・プロジェクトのスケジュール、必要コスト等の管理及びプロジェクト全般の意思決定 ・開発に必要な資源の供給・管理及びプロジェクトの進捗管理、コスト管理、成果管理 ・成果物管理及びコスト評価などプロジェクト全体の評価 	全体
システム エンジニア1 (SE1)	<ul style="list-style-type: none"> ・業務のモデル化、情報システム化の計画を策定 ・システムの機能設計及びシステムの具体化の中心的役割 ・テスト環境整備等とシステム全体のテスト・評価及びマニュアル作成等の中心的役割 ・システム運用・保守にあたってのシステム変更の支援 	<ul style="list-style-type: none"> ・基本設計 (A) (B) ・総合テスト(ベンダ確認)
システム エンジニア2 (SE2)	<ul style="list-style-type: none"> ・システムの基本設計をもとにした詳細設計作成の中心的役割 ・結合テストの中心的な役割 	<ul style="list-style-type: none"> ・詳細設計 ・結合テスト
プログラマ (PG)	<ul style="list-style-type: none"> ・プログラミングの中心的役割 ・プログラムモジュールやプロセスごとのテストを実施 	<ul style="list-style-type: none"> ・プログラム設計製造

4 機能改良プロジェクトファンクションポイントの計測

本定義は、機能改良プロジェクトのファンクションポイント計測のルールを明確にすることを目的としています。

EFPとAFPの計算式とEFPとAFPを構成する各変数の定義については、ファンクションポイント計測マニュアル(R 4. 1. 1)からの抜粋です。

本定義は、調査票Ⅱの[3]と調査票Ⅲの[15]をご記入の際にご利用ください。

機能改良プロジェクトファンクションポイントは次の計算式を用いて求めます。

$$\text{EFP (Enhancement FP count)} = \lceil (\text{ADD} + \text{CHGA} + \text{CFP}) * \text{VAFA} \rceil + (\text{DEL} * \text{VAFB})$$

ADDは機能改良プロジェクトで追加される機能の未調整ファンクションポイント

CHGAは機能改良プロジェクトで変更される機能の未調整ファンクションポイント

CFPは移行のために追加された機能のファンクションポイント

VAFAは機能変更後のアプリケーションの調整係数

DELは機能変更で削除される機能の未調整ファンクションポイント

VAFBは機能変更前のアプリケーションの調整係数

機能改良プロジェクト実施後のアプリケーションファンクションポイントの計測は次のとおりです。

$$\text{AFP (Application FP count)} = \lceil (\text{UFPB} + \text{ADD} + \text{CHGA}) - (\text{CHGB} + \text{DEL}) \rceil * \text{VAFA}$$

UFPBは機能改良プロジェクト開始前のアプリケーションの未調整ファンクションポイント

CHGBは機能改良プロジェクトにより変更となった機能の未調整ファンクションポイント

$$\text{EFP} = \text{B} + \text{C} + \text{D}$$

$$\text{AFP} = \text{A} + \text{C} + \text{D}$$

A:母体(非改造部分)		D:追加
B:削除	C:改造	

禁無断転載

ソフトウェア開発に関する調査票の集計結果
(受託者向け)
非売品

平成22年6月

財団法人経済調査会 経済調査研究所 調査研究部 第三調査研究室

〒104-0061 東京都中央区銀座 5-13-16 東銀座三井ビル

TEL (03)3543-1462

FAX(03)3543-6516