

調査報告

静岡県の地方ソフトウェア業における工学的技法導入に関する調査

伊 東 暁 人

・ 調査の目的

ソフトウェア工学分野で過去に研究開発され提唱されてきたさまざまな開発技法・管理手法が、地方に所在するソフトウェア開発企業・事業所においてどの程度、認知・普及し、その普及がソフトウェアの開発生産性や企業のパフォーマンスにいかなる影響を与えているかを明らかにするために、その実態把握を目的とする。

・ 調査対象

静岡県内に事業所のあるソフトウェア開発に関連すると思われる企業を、静岡経済研究所編『静岡県会社要覧』（2009年版）、(財)静岡産業創造機構のビジネスマッチングサイト“ビジネスパークしずおか”の企業データベース「IT企業ナビ」、NPO法人・静岡情報産業協会(SIIA)の会員一覧、日本商工会議所の商取引支援サイト「ザ・ビジネスモール」などによって抽出し、その代表者・事業所長など経営管理者とソフトウェア開発プロジェクトの管理者を対象とした。対象企業は445社である。

・ 調査方法：

調査票によるアンケート方式。郵送（宅配便業者によるメール便）にて調査票を発送、郵送返信封筒にて回収した。調査票は経営者・事業所の経営管理者向けとソフトウェア開発プロジェクトの管理者向けの2種類である。（本稿末尾の調査票参照）

・ 調査実施時期：

2009（平成21）年6月8～10日に発送、6月26日締切で返送依頼。その後、未返答企業にFAXと電話で回答と返送を依頼し、最終的には7月21日着信回収分までを集計対象とした。

・ 発送／回収数

上記445社を対象に、各社の経営者/事業所長を対象としたアンケートを各1通、計445通と従業員規模に応じてプロジェクト管理者向けアンケートを各1～3通（50名ごとを単位として最大3通まで）計584通を発送した。回収数は経営者/事業所長向けが116（回収率26.1%）、プロジェクト管理者向けが122（回収率20.1%）である。

（なお、本調査の実施にあたっては、NPO法人・静岡情報産業協会の協賛、ご協力をいただいた。ここに記して謝意を表する。）

(経営者／事業所管理者用)

事業所の概要

■直前期売上高	件数(件)	構成比(%)
1,000万円未満	2	1.7
1000万～5000万円未満	20	17.2
5000万～1億円未満	16	13.8
1億～5億円未満	32	27.6
5億～10億円未満	5	4.3
10億～100億円未満	11	9.5
100億円以上	3	2.6
無回答	27	23.3
総数	116	100.0

■営業利益額	件数(件)	構成比(%)
赤字経営	7	6.0
100万円未満	7	6.0
100～250万円未満	19	16.4
250～500万円未満	8	6.9
500～1000万円未満	8	6.9
1000～2000万円未満	8	6.9
2000～5000万円未満	6	5.2
5000～1億円未満	5	4.3
1億～5億円未満	10	8.6
5億～10億円未満	0	0.0
10億円以上	1	0.9
無回答	37	31.9
総数	116	100.0

■営業利益率	件数(件)	構成比(%)
赤字経営	4	3.4
1%未満	9	7.8
1～2%未満	7	6.0
2～3%未満	8	6.9
3～6%未満	9	7.8
6～10%未満	7	6.0
10～15%未満	5	4.3
15%以上	4	3.4
無回答	63	54.3
総数	116	100.0

■売上構成

ソフトウェア開発	件数(件)	構成比(%)
なし	4	3.4
10%未満	2	1.7
10%～20%未満	7	6.0
20%～30%未満	7	6.0
30%～40%未満	3	2.6
40%～50%未満	6	5.2
50%～60%未満	7	6.0
60%～70%未満	4	3.4
70%～80%未満	15	12.9
80%～90%未満	9	7.8
90%～100%未満	17	14.7
100%	19	16.4
無回答	16	13.8
総数	116	100.0

ハードウェア販売・リース	件数(件)	構成比(%)
なし	63	54.3
10%未満	8	6.9
10%～20%未満	10	8.6
20%～30%未満	5	4.3
30%～40%未満	4	3.4
40%～50%未満	4	3.4
50%～60%未満	2	1.7
60%～70%未満	1	0.9
70%～80%未満	0	0.0
80%～90%未満	1	0.9
90%～100%未満	1	0.9
100%	1	0.9
無回答	16	13.8
総数	116	100.0

システム運用・保守サービス	件数(件)	構成比(%)
なし	45	38.8
10%未満	16	13.8
10%～20%未満	17	14.7
20%～30%未満	7	6.0

システム関係人材派遣	件数(件)	構成比(%)
なし	72	62.1
10%未満	6	5.2
10%～20%未満	8	6.9
20%～30%未満	3	2.6

30%～40%未満	7	6.0
40%～50%未満	3	2.6
50%～60%未満	2	1.7
60%～70%未満	1	0.9
70%～80%未満	0	0.0
80%～90%未満	2	1.7
90%～100%未満	0	0.0
100%	0	0.0
無回答	16	13.8
総数	116	100.0

30%～40%未満	2	1.7
40%～50%未満	1	0.9
50%～60%未満	3	2.6
60%～70%未満	1	0.9
70%～80%未満	0	0.0
80%～90%未満	1	0.9
90%～100%未満	1	0.9
100%	2	1.7
無回答	16	13.8
総数	116	100.0

その他	件数(件)	構成比(%)
なし	81	69.8
10%未満	2	1.7
10%～20%未満	7	6.0
20%～30%未満	3	2.6
30%～40%未満	2	1.7
40%～50%未満	1	0.9
50%～60%未満	0	0.0
60%～70%未満	0	0.0
70%～80%未満	1	0.9
80%～90%未満	1	0.9
90%～100%未満	1	0.9
100%	1	0.9
無回答	16	13.8
総数	116	100.0

■従業員数	件数(件)	構成比(%)
1. 5人未満	26	22.4
2. 5～10人	23	19.8
3. 11～20人	25	21.6
4. 21～30人	2	1.7
5. 31～50人	9	7.8
6. 51～100人	9	7.8
7. 101～200人	6	5.2
8. 200人以上	8	6.9
無回答	8	6.9
総数	116	100.0

1-1 年間の平均開発件数(契約件数もしくは開発プロジェクト数など)は何件くらいですか？

	件数(件)	構成比(%)
0件	1	0.9
1件～10件未満	34	29.3
10件～20件未満	23	19.8
20件～30件未満	16	13.8
30件～50件未満	9	7.8
50件～100件未満	9	7.8
100件～250件未満	8	6.9
250件～500件未満	2	1.7
500件～1000件未満	1	0.9
1000件～1500件未満	0	0.0
1500件以上	1	0.9
無回答	12	10.3
総数	116	100.0

1-2 開発案件の契約別割合についてお答えください。

自社の製品開発	件数(件)	構成比(%)
なし	56	48.3
10%未満	12	10.3

ユーザー企業からの受託開発	件数(件)	構成比(%)
なし	13	11.2
10%未満	2	1.7

10%～20%未満	15	12.9
20%～30%未満	7	6.0
30%～40%未満	1	0.9
40%～50%未満	1	0.9
50%～60%未満	2	1.7
60%～70%未満	1	0.9
70%～80%未満	2	1.7
80%～90%未満	2	1.7
90%～100%未満	3	2.6
100%	5	4.3
無回答	9	7.8
総数	116	100.0

10%～20%未満	3	2.6
20%～30%未満	7	6.0
30%～40%未満	7	6.0
40%～50%未満	7	6.0
50%～60%未満	6	5.2
60%～70%未満	7	6.0
70%～80%未満	8	6.9
80%～90%未満	8	6.9
90%～100%未満	17	14.7
100%	21	18.1
無回答	10	8.6
総数	116	100.0

同業他社からの受託開発	件数(件)	構成比(%)
なし	50	43.1
10%未満	4	3.4
10%～20%未満	11	9.5
20%～30%未満	7	6.0
30%～40%未満	4	3.4
40%～50%未満	8	6.9
50%～60%未満	4	3.4
60%～70%未満	3	2.6
70%～80%未満	5	4.3
80%～90%未満	3	2.6
90%～100%未満	3	2.6
100%	4	3.4
無回答	10	8.6
総数	116	100.0

その他	件数(件)	構成比(%)
なし	99	85.3
10%未満	1	0.9
10%～20%未満	5	4.3
20%～30%未満	0	0.0
30%～40%未満	0	0.0
40%～50%未満	0	0.0
50%～60%未満	0	0.0
60%～70%未満	0	0.0
70%～80%未満	0	0.0
80%～90%未満	0	0.0
90%～100%未満	0	0.0
100%	1	0.9
無回答	10	8.6
総数	116	100.0

その他の内訳
<ul style="list-style-type: none"> ・ コンサル ・ コンサル・他社製品販売など ・ サポートのみ ・ 研究開発案件など ・ システム運用サービス

2-1 ソフトウェア開発の生産性管理はどの程度重要ですか。

生産管理の重要性	件数(件)	構成比(%)
1. とても重要である	56	48.3
2. ある程度重要である	38	32.8
3. 普通	13	11.2
4. あまり重要でない	2	1.7
5. 全く重要でない	1	0.9
無回答	6	5.2
総計	116	100.0

2-2 ソフトウェア開発の生産性を測定していますか。

生産性の測定	件数(件)	構成比(%)
1. すべてのプロジェクトで測定している	34	29.3
2. 半数以上のプロジェクトで測定している	20	17.2
3. 特定のプロジェクトのみで測定している	31	26.7
4. 全く測定していない	24	20.7
5. その他	1	0.9
無回答	6	5.2
総計	116	100.0

その他の内訳
・ 顧客の要求による

2-3 問2-2で1、2、3を選択された方にお聞きします。

生産性の測定をどのようにされていますか？

生産性の測定	件数(件)	構成比(%)
1. 工数(人月)ベース	78	67.2
2. SLOCベース	13	11.2
3. FPベース	10	8.6
4. その他	2	1.7
無回答	13	11.2
総計	116	100.0

FP(ファンクションポイント)ベース
・ IFPUG法

その他の内訳
・ 売価及び原価ベースでの測定も行っております
・ 工数(時間)担当別

3 貴社(事業所)では、ソフトウェア開発において標準化された開発方法を導入していますか。

標準化された方法の導入	件数(件)	構成比(%)
1. 導入している	50	43.1
2. 導入していない	55	47.4
無回答	11	9.5
総計	216	100.0

4 貴社では、ソフトウェア開発において品質基準の設定や品質保証の体制をとっていますか。

品質保証の体制の有無	件数(件)	構成比(%)
1. 体制をとっている	48	41.4
2. 体制をとっていない	60	51.7
無回答	8	6.9
総計	116	100.0

5 貴社(事業所)では、ソフトウェア開発を担う人材の育成のプログラムとしてどのような取り組みを行っていますか。(複数回答)

人材育成の取り組み	件数(件)	選択率(%)
1. 貴社内部で社員教育	73	62.9
2. 自己啓発	73	62.9
3. OJT	72	62.1
4. 外部育成講座等へ委託	38	32.8
5. 大学大学院への派遣など	2	1.7
6. その他	2	1.7
7. とくに取り組んでいない	5	4.3
無回答	4	3.4
総計	269	

総数(116)=100

その他の内訳
・ 書籍・ネットによる新技術の習得
・ 外部研修の参加

6 貴社(事業所)では、ソフトウェアの開発生産性を人事考課に反映させていますか？(複数回答)

	件数(件)	選択率(%)
1. 個々の社員として反映	62	53.4
2. プロジェクトチームとして反映	15	12.9
3. 組織レベルで反映	16	13.8
4. 反映させていない	28	24.1
5. その他	5	4.3
6. 無回答	4	3.4
総計	130	

総数(116)=100

その他の内訳
・ 適宜、賞与で反映させている
・ 一人でやっている
・ 生産性悪い場合は、その問題点を探り解決する。
・ 生産性といわれると難しい

7 貴社(事業所)では、ソフトウェア開発にかかわる資格の取得を人事考課に反映させていますか？

	件数(件)	構成比(%)
1. 反映させている	49	42.2
2. 反映させていない	58	50.0
無回答	9	7.8
総数	116	100.0

8 ソフトウェアの開発生産性と次の各項目関係について、「強くそう思う」から「まったく思わない」までの5段階で評価してください

開発方法論の導入は生産性の向上に寄与	件数(件)	構成比(%)
1. 強くそう思う	15	12.9
2. ややそう思う	53	45.7
3. 普通	31	26.7
4. あまり思わない	5	4.3
5. 全く思わない	1	0.9
無回答	11	9.5
総計	116	100.0

開発ツールの導入は生産性の向上に寄与	件数(件)	構成比(%)
1. 強くそう思う	35	30.2
2. ややそう思う	53	45.7
3. 普通	16	13.8
4. あまり思わない	1	0.9
5. 全く思わない	0	0.0
無回答	11	9.5
総計	116	100.0

会社が公的認証を受けることは生産性の向上に寄与	件数(件)	構成比(%)
1. 強くそう思う	2	1.7
2. ややそう思う	18	15.5
3. 普通	46	39.7
4. あまり思わない	28	24.1
5. 全く思わない	10	8.6
無回答	12	10.3
総計	116	100.0

品質管理の強化は生産性の向上に寄与	件数(件)	構成比(%)
1. 強くそう思う	23	19.8
2. ややそう思う	52	44.8
3. 普通	27	23.3
4. あまり思わない	2	1.7
5. 全く思わない	1	0.9
無回答	11	9.5
総計	116	100.0

その他	件数(件)	構成比(%)
1. 強くそう思う	1	0.9
2. ややそう思う	2	1.7
3. 普通	1	0.9
4. あまり思わない	0	0.0
5. 全く思わない	0	0.0
無回答	112	96.6
総計	116	100.0

共通化・部品化
提案の内容
管理者の育成

(経営者/事業所管理者用つづき)

9 ソフトウェアの開発生産性を向上させるために経営上、有効と思われることがありましたらご記入ください。

- ・ 仕様の確定+新しいツールの使用+担当者のやる気
- ・ 上流工程での高い密度のつめがなされてないと、下流工程での工数が予想外に膨れる。
- ・ 目的は何か、目標は何かを明確にする事。経営理念と経営戦略の策定が重要。
- ・ 小さな会社である為、自社の標準化は顧客の標準化により実体が小さなものになってしまう。
- ・ 社内でのソフトウェア開発方法の統一化、マニュアル化。
- ・ 数名の小さな会社では個人のスキルに依存することが多く、知識武装することで、コミュニケーションもスムーズに流れ自然と協調性も生まれ、結果、生産性の向上につながっています。
- ・ 開発標準化と共有部品、サンプルコードの向上は生産性、品質に有効と考えています。
- ・ 途中退職を無くす。
- ・ 社内教育の実施に力を入れています。又、社外セミナー教育（基本的教育を中心）も行っている。今後は社内教育にもっと力を入れたいと思う。
- ・ ソフトウェアの開発生産性は個人のスキルレベルに非常に依存している。標準化等の良い方法があればよいと思うが・・・？
- ・ 弊社は地元企業の各種別基幹システムの開発を行っている為、業種別基本システムの品質向上が経営戦略に重要であります。
- ・ 標準化の推進は非常に有効と考える。開発後のメンテナンス時に対しても効率よくできると思う。又、標準化する事で品質も向上する。
- ・ 生産性の向上は、人的要素（レベルアップ）が必須と考えていますので、教育、人材育成が重要かと思います。
- ・ パーツ化あるいはパッケージ化をすることで品質も生産性も向上すると思われます。
- ・ 構造化、部品化の推進
- ・ 環境整備（S/W、H/W）
- ・ 生産性向上の行き着く先はどこか。見極める事が重要。最終的にはノンプログラミングでは事業構造そのものが崩壊する。浜松地区でWEBを中心としたシステム開発のグループを作っています。
- ・ 少人数の企業では1人プロジェクトが有効です。
- ・ 技術教育の強化及び先を見た段取りをする事で待ち時間を少なくできる。
- ・ 論理的な思考により生産性を上げることができる。（やみくもにテストしても意味がない。）（試した結果ではなく論理で組み上げていく事で全ての工程での時間短縮が可能。）
- ・ ベストの状態を少しでも多く維持させる事。（能力をフルに発揮させる）

- ・開発生産がお金と結び付くことを社員に認識させる。
- ・これまでも、ソフトウェアの生産性については色々議論されてはきましたが、バグが発見されればそれを撃退することが最重要課題で、あらゆる工程で一意的に方法論を適用しようとする逆生産性を損なう場合が多々あります。ドキュメントに関しても、そのほとんどはユーザーには不必要（目を通さない）で、特にツールを使用したドキュメントはユーザーには不評です。ユーザーによってはコスト削減の為、ドキュメント（取り説以外）を拒否する場合があります。ソフトウェアは時間をかけようと思えばいくらでも掛けることの可能な職種です。システム開発に於いてはポイントをいかに掴むか、ユーザーは何を真に要望しているのか、その為には最低限何を提供すれば良いかを早い段階で把握することが最良の開発効率と思い取り組んでおります。
- ・顧客のニーズを的確に把握出来る国語能力及び意志を正確に理解するコミュニケーション能力。すべての基本だと考えます。
- ・開発物件（受注物件）ごとの業務分析、基本設計、詳細設計、プログラム設計、プログラム開発、テスト、本稼働の指導などいくつかの工程に分け、各工程ごとに予算時間、実績時間、予算金額、実績金額、達成率等を仕掛（中間）状況及び開発物件完了時等、毎日確認する事により、問題点を抽出する。(物件別製番決算書作成)それを踏まえて、基準見直し会議、原価会議等を行うことにより、開発生産性向上に結びつくと思います。
- ・下請け企業にとっては業界全体の生産性向上は受注業務量が減少するだけで歓迎できる事ではありません。下請けの場合、元請けからの標準化の徹底等で独自の業務改善(方法論、ツールの導入)は全く認められません。開発スキルの向上のみが生き残る道です。
- ・方法やツールなどをどう有効的に運用していくか、またそれらの必要性を一人一人に理解させるかが大切であり、それができなければ何をしてもムダになってしまう。
- ・上流工程（調査・設計）をしっかりと行い、ドキュメントを残すことが重要。また、お客様の要求を具体化し、ソフトウェアに反映することも重要。技法も大切であるが、ソフトウェア開発では、関連部門とのコミュニケーションを充分に取り、問題点の早期検出、改善が総合的に見た場合、生産性の向上に寄与すると思う。オブジェクト指向設計の手法を実務でも段階的に取り入れていく必要があると思います。
- ・経営管理者も技術進歩がはげしいので、現場と物作りで対等に話ができるように、経営者側も物作りできるレベルを維持していないとダメです。
- ・社内コミュニケーション。協力関係の確立。
- ・弊社では、品質・スケジュール・コスト面からプロジェクトを管理しています。それぞれに一般的なもの、弊社独自のもののツールを導入しています。またツールなどの導入が儀式的にならないよう、チェック機能も導入しています。

- ・ 報連相をしっかりと行う。わからなければ質問をする。聞くこと。
- ・ 残業をしないという意識。
- ・ 案件ごとに各フェーズで関係した担当全員での反省会が重要と思う。(次期開発に活かすため)

(SE/プロジェクト管理者用)

1-1 あなたのソフトウェア開発の業務経験年数は何年くらいですか？

	件数(件)	構成比(%)
5年未満	2	1.6
5年～10年未満	13	10.7
10年～15年未満	15	12.3
15年～20年未満	18	14.8
20年～25年未満	33	27.0
25年～30年未満	19	15.6
30年以上	14	11.5
無回答	8	6.6
総数	122	100.0

1-2 あなたが過去に経験した開発プロジェクトの件数は全部で何件くらいですか？

	件数(件)	構成比(%)
5件未満	1	0.8
5件～10件未満	7	5.7
10件～20件未満	15	12.3
20件～30件未満	11	9.0
30件～40件未満	13	10.7
40件～50件未満	9	7.4
50件～60件未満	14	11.5
60件～100件未満	10	8.2
100件以上	25	20.5
無回答	17	13.9
総数	122	100.0

1-3 あなたが過去に担当した開発プロジェクトは、いわゆる「業務系」(エンタープライズ系)ですか？「組み込み系」(制御系)ですか？

過去の開発プロジェクトについて	件数(件)	構成比(%)
1. ほとんど業務系	84	68.9
2. ほとんど組み込み系	13	10.7
3. 半々くらい	10	8.2
4. その他	7	5.7
無回答	8	6.6
総数	122	100.0

その他の内訳

- ・ パッケージ製品
- ・ メーカー系サブOS、メーカー系ミドルウェア
- ・ FA,PA
- ・ パッケージ
- ・ ミドルウェア
- ・ 業務:70%、組み込み:30%
- ・ ホームページ

2-1 あなたにとってソフトウェア開発の生産性管理はどの程度重要ですか。

生産管理の重要性	件数(件)	構成比(%)
1. とても重要である	46	37.7
2. ある程度重要である	56	45.9
3. 普通	9	7.4
4. あまり重要ではない	3	2.5
5. 全く重要でない	0	0.0
無回答	8	6.6
総数	122	100.0

2-2 ソフトウェア開発の生産性管理を測定していますか。

生産管理の測定	件数(件)	構成比(%)
1. すべてのプロジェクトで測定	30	24.6
2. 半数以上のプロジェクトで測定	23	18.9

3. 特定のプロジェクトのみで測定	36	29.5
4. 全く測定していない	21	17.2
5. その他	2	1.6
無回答	10	8.2
総数	122	100.0

その他の内訳
<ul style="list-style-type: none"> 顧客の要求による 試験的に導入している

2-3 問2-2で1、2、3を選択された方にお聞きします。

生産性の測定をどのようにされていますか？

生産性の測定方法	件数(件)	構成比(%)
1. 工数(人月)ベース	87	71.3
2. SLOCベース	19	15.6
3. FPベース	9	7.4
4. その他	5	4.1
無回答	2	1.6
総数	122	100.0

FPベース(法)
<ul style="list-style-type: none"> 画面 IFPUG法 工数(時間)担当別

その他の内訳
<ul style="list-style-type: none"> 機能と関数の数と質、自分で考えています。 メーカー側の生産性基準 原価管理 品質管理(不具合管理など)

3 貴社(事業所)では、ソフトウェア開発において標準化された開発方法を導入していますか。

総数(122)=100

開発フェーズ	件数(件)	選択率(%)
要求仕様確定フェーズ	23	18.9
詳細設計フェーズ	27	22.1
プログラム生成フェーズ	25	20.5
テストフェーズ	22	18.0
その他	6	4.9
総計	103	

方法論の名称【要求仕様確定フェーズ】
<ul style="list-style-type: none"> 社内標準(6件) ウォーターフォール(2件) スパイラル開発技法、プロトタイプングのMix 構造化分析設計法 メーカー固有の方法 関与しているメーカー開発方法に準拠(SDAS等) 標準化、文書化、図式化 打ち合せ議事録とサンプル画面作成 RAD ヒアリング、プロトタイプ、UML、DFD 独自の基準はないが、ユーザ規格に準拠

方法論の名称【詳細設計フェーズ】
<ul style="list-style-type: none"> 社内標準 (7件) 構造化分析設計 構造化設計、オブジェクト指向設計 スパイラル開発技法、プロトタイプングのMix メーカー固有の方法 関与しているメーカー開発方法に準拠(SDAS等) 詳細設計所のひな型があり UML IPAの文書を参考にした独自方法 構造化分析設計法 ウォーターフォール 画面及び変数表、db定義表の作成 独自の基準はないが、ユーザ規格に準拠

方法論の名称【プログラム生成フェーズ】
<ul style="list-style-type: none"> 社内標準(5件) スパイラル開発技法、プロトタイプングのMix メーカー固有の方法 関与しているメーカー開発方法に準拠(SDAS等) 処理区分毎にプログラムのひな型がある パラダイム TDD IPAの文書を参考にした独自方法 ウォーターフォール コメントをプログラム単位で作成 コードレビュー、デバッグ フレームワーク(自社開発) 成果物がイメージしやすければ何でも採用 独自の基準はないが、ユーザ規格に準拠

方法論の名称【プログラム生成フェーズ】
<ul style="list-style-type: none"> 社内標準 (6件) 関数ベース、モジュールベース、全体テスト メーカー固有の方法 関与しているメーカー開発方法に準拠(SDAS等) 詳細設計時にテスト仕様書を作成 IPAの文書を参考にした独自方法 ウォーターフォール

方法論の名称【その他】
<ul style="list-style-type: none"> 自社で積み上げたもの(部品モジュールなどを利用)

- ・基本はウォーターフォールに近いイメージです。
- ・デバッグリストにて実施
- ・単体テスト、結合テスト、システム総合テスト
- ・独自の基準はないが、ユーザ規格に準拠

- ・ホワイトBox、ブラックBox テスト
- ・オブジェクト指向
- ・プロジェクト毎に開発の標準化方法を検討しています。
- ・顧客やプロジェクト、納期等の状況に応じて、対応が変化するので決まった方法というものはない
- ・メトロソフトウェア開発ガイドライン(弊社オリジナル)

4 あなたは、ソフトウェア開発のさまざまな技法や知識をどのように習得していますか。

(複数回答) 総数(122)=100

	件数(件)	選択率(%)
1. 貴社内部での社員教育	44	36.1
2. 自己啓発、自己研修	94	77.0
3. OJT(On-the-Job Training)	63	51.6
4. 外部の育成講座等への参加	24	19.7
5. 大学、大学院等への通学	1	0.8
6. その他	4	3.3
7. とくにとり組んでいない	4	3.3
無回答	8	6.6
総計	242	

その他の内訳

- ・書籍・ネットなど
- ・書籍
- ・前職に習得
- ・仕事上で習得

5 ソフトウェア開発の生産性を向上させるうえで、開発のフェーズ別の課題の有無を5段階で評価してください。

■要求仕様確定フェーズ	件数(件)	構成比(%)
1. とても課題が多い	33	27.0
2. やや課題が多い	50	41.0
3. 普通	23	18.9
4. やや課題が少ない	2	1.6
5. まったく課題がない	1	0.8
無回答	13	10.7
総数	122	100.0

■詳細設計フェーズ	件数(件)	構成比(%)
1. とても課題が多い	10	8.2
2. やや課題が多い	51	41.8
3. 普通	43	35.2
4. やや課題が少ない	4	3.3
5. まったく課題がない	1	0.8
無回答	13	10.7
総数	122	100.0

■プログラム生成フェーズ	件数(件)	構成比(%)
1. とても課題が多い	6	4.9
2. やや課題が多い	20	16.4
3. 普通	67	54.9
4. やや課題が少ない	14	11.5
5. まったく課題がない	2	1.6
無回答	13	10.7
総数	122	100.0

■デバッグ/テストフェーズ	件数(件)	構成比(%)
1. とても課題が多い	12	9.8
2. やや課題が多い	41	33.6
3. 普通	48	39.3
4. やや課題が少ない	6	4.9
5. まったく課題がない	2	1.6
無回答	13	10.7
総数	122	100.0

■その他	件数(件)	構成比(%)
1. とても課題が多い	3	2.5
2. やや課題が多い	2	1.6
3. 普通	3	2.5
4. やや課題が少ない	0	0.0
5. まったく課題がない	1	0.8
無回答	113	92.6
総数	122	100.0

その他の内訳
・ユーザー教育
・エンドユーザーテストフェーズ
・プロジェクト管理
・契約フェーズ(お金、期間等)
・●開発ツール選定時
●基本モジュール構築とひな型作成時
●規約作成時

6 開発プロジェクトと顧客(発注者)との関係について、あてはまる番号に一つだけ○をおつけください。

6-1 顧客の要求する仕様は明確ですか？

	件数(件)	構成比(%)
5. 非常に明確である	2	1.6
4. おおむね明確である	40	32.8
3. ふつう	22	18.0
2. あまり明確でない	46	37.7
1. まったく明確でない	2	1.6
無回答	10	8.2
総数	122	100.0

6-2 要求仕様を確定するうえで、顧客は関与しますか？

	件数(件)	構成比(%)
5. すべて関与する	26	21.3
4. おおむね関与する	66	54.1
3. ふつう	11	9.0
2. あまり関与しない	11	9.0
1. まったく関与しない	0	0.0
無回答	8	6.6
総数	122	100.0

6-3 顧客はあなたの設計を理解していますか？

	件数(件)	構成比(%)
5. 完全に理解している	0	0.0
4. おおむね理解している	58	47.5
3. ふつう	31	25.4
2. あまり理解していない	21	17.2
1. まったく理解していない	2	1.6
無回答	10	8.2
総数	122	100.0

6-4 顧客は開発したソフトウェアのテストに関与しますか？

	件数(件)	構成比(%)
5. すべて関与する	13	10.7
4. おおむね関与する	45	36.9
3. ふつう	22	18.0
2. あまり関与しない	31	25.4
1. まったく関与しない	1	0.8
無回答	10	8.2
総数	122	100.0

7 ソフトウェア開発における開発方法の認知と導入、効果について、それぞれ5段階で評価してください。

7-1 開発プロジェクト全体の方法論について(方法論の認知)

■ウォーターフォールモデル	件数(件)	構成比(%)
よく知っている	47	38.5
まあまあ知っている	23	18.9
普通	21	17.2
あまり知らない	6	4.9
全く知らない	15	12.3
無回答	10	8.2
総数	122	100.0

■プロトタイプング	件数(件)	構成比(%)
よく知っている	33	27.0
まあまあ知っている	31	25.4
普通	33	27.0
あまり知らない	6	4.9
全く知らない	9	7.4
無回答	10	8.2
総数	122	100.0

■スパイラル型	件数(件)	構成比(%)
よく知っている	15	12.3
まあまあ知っている	23	18.9
普通	40	32.8
あまり知らない	17	13.9
全く知らない	17	13.9
無回答	10	8.2
総数	122	100.0

■アジャイル型(XP、スクラムなど)	件数(件)	構成比(%)
よく知っている	4	3.3
まあまあ知っている	14	11.5
普通	32	26.2
あまり知らない	32	26.2
全く知らない	27	22.1
無回答	13	10.7
総数	122	100.0

■その他	件数(件)	構成比(%)
よく知っている	0	0.0
まあまあ知っている	0	0.0
普通	1	0.8
あまり知らない	1	0.8
全く知らない	6	4.9
無回答	114	93.4
総数	122	100.0

7-1 開発プロジェクト全体の方法論について(方法論の使用)

■ウォーターフォールモデル	件数(件)	構成比(%)
よく使っている	40	32.8
まあまあ使っている	21	17.2
普通	21	17.2
あまり使わない	8	6.6
全く使わない	22	18.0
無回答	10	8.2
総数	122	100.0

■プロトタイプング	件数(件)	構成比(%)
よく使っている	20	16.4
まあまあ使っている	23	18.9
普通	33	27.0
あまり使わない	15	12.3
全く使わない	20	16.4
無回答	11	9.0
総数	122	100.0

■スパイラル型	件数(件)	構成比(%)
よく使っている	8	6.6
まあまあ使っている	10	8.2
普通	21	17.2
あまり使わない	27	22.1
全く使わない	45	36.9
無回答	11	9.0
総数	122	100.0

■アジャイル型(XP、スクラムなど)	件数(件)	構成比(%)
よく使っている	3	2.5
まあまあ使っている	2	1.6
普通	12	9.8
あまり使わない	25	20.5
全く使わない	65	53.3
無回答	15	12.3
総数	122	100.0

■その他	件数(件)	構成比(%)
よく使っている	1	0.8
まあまあ使っている	0	0.0
普通	1	0.8
あまり使わない	0	0.0
全く使わない	8	6.6
無回答	112	91.8
総数	122	100.0

その他の内訳
・Vモデル

7-1 開発プロジェクト全体の方法論について(方法論の効果)

■ウォーターフォールモデル	件数(件)	構成比(%)
とても効果がある	9	7.4
まあまあ効果がある	20	16.4
普通	51	41.8
あまり効果がない	9	7.4
全く効果がない	8	6.6
無回答	25	20.5
総数	122	100.0

■プロトタイプング	件数(件)	構成比(%)
とても効果がある	11	9.0
まあまあ効果がある	37	30.3
普通	41	33.6
あまり効果がない	4	3.3
全く効果がない	7	5.7
無回答	22	18.0
総数	122	100.0

■スパイラル型	件数(件)	構成比(%)
とても効果がある	2	1.6
まあまあ効果がある	13	10.7
普通	42	34.4
あまり効果がない	12	9.8
全く効果がない	18	14.8
無回答	35	28.7
総数	122	100.0

■アジャイル型(XP、スクラムなど)	件数(件)	構成比(%)
とても効果がある	0	0.0
まあまあ効果がある	10	8.2
普通	27	22.1
あまり効果がない	14	11.5
全く効果がない	22	18.0
無回答	49	40.2
総数	122	100.0

■その他	件数(件)	構成比(%)
とても効果がある	1	0.8
まあまあ効果がある	0	0.0
普通	2	1.6
あまり効果がない	0	0.0
全く効果がない	2	1.6

その他の内訳
・Vモデル

無回答	117	95.9
総数	122	100.0

7-2 設計フェーズの方法論について(方法論の認知)

■構造化分析設計	件数(件)	構成比(%)
よく知っている	32	26.2
まあまあ知っている	28	23.0
普通	35	28.7
あまり知らない	7	5.7
全く知らない	9	7.4
無回答	11	9.0
総数	122	100.0

■オブジェクト指向分析設計	件数(件)	構成比(%)
よく知っている	23	18.9
まあまあ知っている	29	23.8
普通	42	34.4
あまり知らない	11	9.0
全く知らない	6	4.9
無回答	11	9.0
総数	122	100.0

■データ中心アプローチ	件数(件)	構成比(%)
よく知っている	23	18.9
まあまあ知っている	21	17.2
普通	49	40.2
あまり知らない	9	7.4
全く知らない	9	7.4
無回答	11	9.0
総数	122	100.0

■その他	件数(件)	構成比(%)
よく知っている	0	0.0
まあまあ知っている	2	1.6
普通	1	0.8
あまり知らない	0	0.0
全く知らない	5	4.1
無回答	114	93.4
総数	122	100.0

その他の内訳
・ ERP導入Fit & Gap

7-2 設計フェーズの方法論について(方法論の使用)

■構造化分析設計	件数(件)	構成比(%)
よく使っている	30	24.6
まあまあ使っている	29	23.8
普通	25	20.5
あまり使わない	11	9.0
全く使わない	16	13.1
無回答	11	9.0
総数	122	100.0

■オブジェクト指向分析設計	件数(件)	構成比(%)
よく使っている	17	13.9
まあまあ使っている	23	18.9
普通	43	35.2
あまり使わない	12	9.8
全く使わない	16	13.1
無回答	11	9.0
総数	122	100.0

■データ中心アプローチ	件数(件)	構成比(%)
よく使っている	20	16.4
まあまあ使っている	18	14.8
普通	35	28.7
あまり使わない	21	17.2
全く使わない	17	13.9
無回答	11	9.0
総数	122	100.0

■その他	件数(件)	構成比(%)
よく使っている	0	0.0
まあまあ使っている	2	1.6
普通	0	0.0
あまり使わない	1	0.8
全く使わない	5	4.1
無回答	114	93.4
総数	122	100.0

その他の内訳
・ ERP導入Fit & Gap

7-2 設計フェーズの方法論について(方法論の効果)

■構造化分析設計	件数(件)	構成比(%)
とても効果がある	13	10.7
まあまあ効果がある	34	27.9
普通	44	36.1
あまり効果がない	4	3.3

■オブジェクト指向分析設計	件数(件)	構成比(%)
とても効果がある	8	6.6
まあまあ効果がある	31	25.4
普通	49	40.2
あまり効果がない	5	4.1

全く効果がない	9	7.4
無回答	18	14.8
総数	122	100.0

全く効果がない	7	5.7
無回答	22	18.0
総数	122	100.0

■データ中心アプローチ	件数(件)	構成比(%)
とても効果がある	10	8.2
まあまあ効果がある	27	22.1
普通	51	41.8
あまり効果がない	6	4.9
全く効果がない	6	4.9
無回答	22	18.0
総数	122	100.0

■その他	件数(件)	構成比(%)
とても効果がある	0	0.0
まあまあ効果がある	1	0.8
普通	2	1.6
あまり効果がない	1	0.8
全く効果がない	2	1.6
無回答	116	95.1
総数	122	100.0

8 あなたが過去に使ったことがある開発ツールについてその経験の有無と生産性向上の効果について評価してください。

■プロジェクト管理ツール	件数(件)	構成比(%)
とても効果が高い	5	4.1
まあまあ効果がある	25	20.5
普通	29	23.8
あまり効果がない	9	7.4
全く効果がない	3	2.5
無回答(使用経験なし)	51	41.8
総数	122	100.0

■構成管理ツール	件数(件)	構成比(%)
とても効果が高い	7	5.7
まあまあ効果がある	6	4.9
普通	16	13.1
あまり効果がない	10	8.2
全く効果がない	4	3.3
無回答(使用経験なし)	79	64.8
総数	122	100.0

■設計支援ツール	件数(件)	構成比(%)
とても効果が高い	3	2.5
まあまあ効果がある	17	13.9
普通	29	23.8
あまり効果がない	8	6.6
全く効果がない	4	3.3
無回答(使用経験なし)	61	50.0
総数	122	100.0

■ドキュメント作成ツール	件数(件)	構成比(%)
とても効果が高い	1	0.8
まあまあ効果がある	22	18.0
普通	31	25.4
あまり効果がない	20	16.4
全く効果がない	6	4.9
無回答(使用経験なし)	42	34.4
総数	122	100.0

■デバッグ/テストツール	件数(件)	構成比(%)
とても効果が高い	15	12.3
まあまあ効果がある	33	27.0
普通	23	18.9
あまり効果がない	8	6.6
全く効果がない	2	1.6
無回答(使用経験なし)	41	33.6
総数	122	100.0

■CASEツール	件数(件)	構成比(%)
とても効果が高い	0	0.0
まあまあ効果がある	9	7.4
普通	19	15.6
あまり効果がない	11	9.0
全く効果がない	5	4.1
無回答(使用経験なし)	78	63.9
総数	122	100.0

■コードジェネレータ	件数(件)	構成比(%)
とても効果が高い	5	4.1
まあまあ効果がある	8	6.6
普通	24	19.7
あまり効果がない	11	9.0
全く効果がない	6	4.9
無回答(使用経験なし)	68	55.7
総数	122	100.0

■その他	件数(件)	構成比(%)
とても効果が高い	3	2.5
まあまあ効果がある	0	0.0
普通	0	0.0
あまり効果がない	1	0.8
全く効果がない	1	0.8
無回答(使用経験なし)	117	95.9
総数	122	100.0

その他の内訳
・バグ管理ツール

- ・ 統合開発環境 (Visual Studio, Eclipse)
- ・ ソースコード管理ツール (CVS, subversion)

9 ソフトウェアの開発生産性と次の各項目関係について、「強くそう思う」から「まったく思わない」までの5段階で評価してください

■ 開発方法論の導入は生産性の向上に寄与する

	件数(件)	構成比(%)
強くそう思う	10	8.2
まあまあそう思う	50	41.0
普通	41	33.6
あまり思わない	9	7.4
まったく思わない	2	1.6
無回答	10	8.2
総数	122	100.0

■ 開発ツールの導入は生産性の向上に寄与する

	件数(件)	構成比(%)
強くそう思う	18	14.8
まあまあそう思う	59	48.4
普通	28	23.0
あまり思わない	5	4.1
まったく思わない	1	0.8
無回答	11	9.0
総数	122	100.0

■ 会社が公的認証を受けることは生産性の向上に寄与する

	件数(件)	構成比(%)
強くそう思う	1	0.8
まあまあそう思う	15	12.3
普通	43	35.2
あまり思わない	40	32.8
まったく思わない	12	9.8
無回答	11	9.0
総数	122	100.0

■ 品質管理の強化は生産性の向上に寄与する

	件数(件)	構成比(%)
強くそう思う	16	13.1
まあまあそう思う	56	45.9
普通	29	23.8
あまり思わない	8	6.6
まったく思わない	2	1.6
無回答	11	9.0
総数	122	100.0

■ 自分もしくは開発メンバーが情報系の各種資格を取得することは生産性の向上に寄与する

	件数(件)	構成比(%)
強くそう思う	6	4.9
まあまあそう思う	31	25.4
普通	44	36.1
あまり思わない	22	18.0
まったく思わない	8	6.6
無回答	11	9.0
総数	122	100.0

■ 顧客(発注者)とのコミュニケーションは生産性の向上に寄与する

	件数(件)	構成比(%)
強くそう思う	66	54.1
まあまあそう思う	38	31.1
普通	6	4.9
あまり思わない	1	0.8
まったく思わない	0	0.0
無回答	11	9.0
総数	122	100.0

■ その他の生産性の向上に寄与する方法

	件数(件)	構成比(%)
強くそう思う	1	0.8
まあまあそう思う	0	0.0
普通	1	0.8
あまり思わない	0	0.0
まったく思わない	0	0.0
無回答	120	98.4
総数	122	100.0

- その他の内訳
- ・ スケジュールコストの管理

10 工期管理について、過去に計画工期に対して実績工期が遅延したことがありますか？

	件数(件)	構成比(%)
1. ある	106	86.9
2. ない	7	5.7
無回答	9	7.4
総数	122	100.0

■実績工期の遅延の理由(複数回答) 総数(106)=100

	件数(件)	選択率(%)
1 システム化目的不相当	13	12.3
2 RFP内容不相当	13	12.3
3 要件仕様の決定遅れ	85	80.2
4 要件分析作業不十分	51	48.1
5 開発規模の増大	74	69.8
6 自社内メンバーの選択不相当	12	11.3
7 業務委託会社選択ミス	7	6.6
8 構築チーム能力不足	22	20.8
9 テスト計画不十分	28	26.4
10 受入検査不十分	10	9.4
11 総合テストの不足	27	25.5
12 プロジェクトマネージャーの管理不足	26	24.5
13 その他	3	2.8
無回答	8	7.5
総数	379	

その他の内訳
・ サービス開始直前での大規模な仕様変更
・ ハードウェア・OS・ドライバのトラブル
・ 他真問題 (顧客提示要件誤り、担当外機能トラブル等)

11 開発プロジェクトで作業量の「見積もり」で何らかのトラブルが生じたことがありますか？

	件数(件)	構成比(%)
1. ある	98	80.3
2. ない	14	11.5
無回答	10	8.2
総数	122	100.0

■見積もりのトラブルの理由(複数回答) 総数(98)=100

	件数(件)	選択率(%)
1 要件があいまい	66	67.3
2 仕様の変更が多い	79	80.6
3 見積もりの作成・評価にかかる時間がない	28	28.6
4 客先に納得してもらえない積算根拠が示せない	10	10.2
5 開発に使用する技術・製品が多岐にわたりすぎる	12	12.2
6 開発チームのメンバーの力量把握が不十分	14	14.3
7 客先との理解の齟齬	40	40.8
8 その他	6	6.1
無回答	8	8.2
総数	263	

12 あなたが過去に使ったことがある「見積もり」技法についてその経験の有無と生産性向上の効果についてお答えください。

■類推法(概算法)	件数(件)	構成比(%)	■ファンクション・ポイント法	件数(件)	構成比(%)
とても効果が高い	3	2.5	とても効果が高い	1	0.8
まあまあ効果がある	23	18.9	まあまあ効果がある	14	11.5
普通	38	31.1	普通	23	18.9
あまり効果がない	3	2.5	あまり効果がない	4	3.3
全く効果がない	0	0.0	全く効果がない	1	0.8
無回答(使用経験なし)	55	45.1	無回答(使用経験なし)	79	64.8
総数	122	100.0	総数	122	100.0

静岡県の地方ソフトウェア業における工学的技法導入に関する調査

■WBS法	件数(件)	構成比(%)
とても効果が高い	4	3.3
まあまあ効果がある	8	6.6
普通	12	9.8
あまり効果がない	1	0.8
全く効果がない	1	0.8
無回答(使用経験なし)	96	78.7
総数	122	100.0

■LOC(Line of Code)法	件数(件)	構成比(%)
とても効果が高い	0	0.0
まあまあ効果がある	4	3.3
普通	10	8.2
あまり効果がない	3	2.5
全く効果がない	3	2.5
無回答(使用経験なし)	102	83.6
総数	122	100.0

■ユースケース・ポイント法	件数(件)	構成比(%)
とても効果が高い	0	0.0
まあまあ効果がある	1	0.8
普通	8	6.6
あまり効果がない	3	2.5
全く効果がない	1	0.8
無回答(使用経験なし)	109	89.3
総数	122	100.0

■COCOMO・COCOMOⅡなど	件数(件)	構成比(%)
とても効果が高い	0	0.0
まあまあ効果がある	1	0.8
普通	2	1.6
あまり効果がない	2	1.6
全く効果がない	1	0.8
無回答(使用経験なし)	116	95.1
総数	122	100.0

■その他	件数(件)	構成比(%)
とても効果が高い	0	0.0
まあまあ効果がある	2	1.6
普通	0	0.0
あまり効果がない	1	0.8
全く効果がない	0	0.0
無回答(使用経験なし)	119	97.5
総数	122	100.0

その他の内訳
・ストーリーポイント

見積もり技法	件数(件)
特に利用していない	31

(SE・プロジェクト管理者用つづき)

13 ソフトウェアの開発生産性を向上させるために、有効と思われることがありましたらご記入ください。

- ・ 目的目標を明確化する。ユーザー、経営者、開発側すべて同じ目的・目標に向い各々の立場で情報の共有化をする。
- ・ カスタマイズしたCASEツールの利用同様にカスタマイズしたコードジェネレーターの利用（カスタマイズしていなければ、全く意味はない。）
- ・ 技術者個々のスキルアップは勿論だが、プロジェクトマネージャーの管理及びフォローのやり方によるところが大きいと思う。
- ・ ユーザーと良好な関係を保つ。
- ・ 開発フレームワークについてはあまり多くの選択肢を持たず、標準化および方式設計を吟味した上、社内標準を決定する事で製造工程以降の工数を低減できるのではないかと考えます。
- ・ 業務毎に有用なやり方をアサインできる能力が必要と感じます。
- ・ 要求事項の明確化及び提案力、説得力。まとめてしまえば、顧客とのコミュニケーションと、PMのコミュニケーション能力が重要。（有能な社員がいることも大事。自信のある方に入社してもらいたい。）
- ・ 人と人とのコミュニケーション
- ・ 環境整備、特にテストの自動化
- ・ 選択と集中
- ・ 標準化（プログラムのサンプル）を進め、誰が作成しても同様なQCDを満たすプログラムを作成できる環境にしていくことが重要であると考えています。
- ・ プロジェクトメンバー間の意思疎通がとても重要だと思います。全員がチームとしてゴールに向いている事を常に意識して、スムーズに開発を進めていくには「その時一人一人が自分は何をすればいいか？」を考え行動できていれば、おのずと開発生産性は上がると思います。
- ・ 顧客とのコミュニケーション、メンバーの意思疎通→開発におけるフィーリング、同構想8割を伝え、その先を理解できる能力。
- ・ 1. 各工程の標準化 2. 各工程毎の上位者のチェック、レビュー実施、徹底。
- ・ 生産性は計測（予測）がほぼできない場合が多い。上流工程を重視して”戻り”をそのフェーズ内に留める。
- ・ 弊社では、ウォーターフォールモデル主体であるが、上流工程の精度で生産性が決定してしまう。要件定義を確実に行うことが重要となっている。
- ・ 要求仕様の確定、レビュー

・開発プロジェクトで考えた場合、いかに上流工程で要求仕様を明確にできるかが、生産性のポイントかと思う。下流工程に行くほど、課題がでたときの後戻り工程が大きくなることは自明の通り。また、例えば製造工程では、プログラムソースの流用化率を上げることも単純ではあるが、有効と考えられるが、流用可能なソースを多く知っているためには、多くのソースに触れている必要があり、経験や環境に左右されやすいところではある。

・いかにツールなどで効率をあげていかなど考えなければ、何事も面倒と感じてしまい生産性の向上には働かない。

・オブジェクト指向言語でない場合（COBOL等）、F.P法での見積もり算出は難しい部分があるため、積算法が有効である。

・プログラム生成フェーズの前段階までの精度を高めること。

・他人が読まないような自己満足的なドキュメントを作ることを減らす ・wiki等で情報を共有する ・他人が見て分かりやすいソースを書く。またソース内にコメントを残す。

・顧客調整（Q,C,D） ・各開発フェーズでの品質強化（レビュー、テストの充実）

・システム受注時にシステム全体の要件を見える（だれにも説明できる）ところまで追求してまとめる。あとは、適切な方法論で、手法・ツールでまとめればOKだと思う。プロジェクトは何をどのようにしたいとかはっきりした顧客の知識及び協力が絶対不可欠です。（プロジェクトの成功or失敗は顧客の協力で95%は確定する）

各方法論・手法論には長所、欠点もあるので単純にきめた方法にこだわって作業してはいけない。あくまでこれはツールです。現実には単純なアンケートだけで正しい調査などできない。誤差ブレが多いと思う。どのように使っているかも？

（例）あるプロジェクトでこのようなことで大問題が発生し、その時の考え方をこの方法論・手法論で見事に解決したというところまで調査して、その解決プロセスを公表してこそ静岡県及び日本の生産性向上に寄与すると思います。補助金でやるならそこまで徹底的にやって補助金を生かしてほしいですね。

最近では技術の進歩が激しくSE手法と物作り現場手法が違い大変です。本屋にはオブジェクト指向という本が多くあり使うと効率よいが、設計からその作りでないと出来ません。SE及び物作り現場でどれほど同期をとってオブジェクト指向でおこなっているかが問題です。現実あまり使っていないという話も多い。現在、中国の優秀な技術者の台頭があり、日本の大学生の工学ばなれなど、大学でも本当に使えるシステム・プログラムを教育してくれる先生がすくないなど日本国内の問題も多いと聞いております。

・開発者間のコミュニケーションと教育。過去に作成されたプログラムの再利用。

・こちらが教えてもらいたいくらいですが、1つは経験です。状況や業種、チームの能力によって

も、管理方法は大きく異なります。20年には満たない経験ですが、血を流して学んだことが一番役に立っています。

- ・コミュニケーションを十分に。中間報告、確認チェック。
- ・案件ごとに各フェーズで関係した担当全員での反省会が重要と思う。
- ・ユーザーが小売ということもあり、要件や仕様の明確化を求めることは経験上困難。当社の場合「機能追加・変更が発生しても短時間で」をモットーとし、それらが可能であるようなパッケージ作りに力を入れている。
- ・顧客の理解度の向上のためのコミュニケーション強化や打ち合わせの確認

*本報告は、(独)日本学術振興会平成21年度科学研究費補助金(基盤研究(C))「地方ソフトウェア業における工学的技法導入の経営に与える影響に関する研究」(課題番号:20530319)による成果の一部である。

問4 あなたは、ソフトウェア開発のさまざまな技法や知識をどのように習得していますか。あてはまる番号すべてに○をおつけください。（複数選択可）

- 1. 貴社内部での社員教育・研修システム
- 2. 自己啓発、自己研修
- 3. OJT (On-the-Job Training)
- 4. 会社派遣による外部の育成講座等への参加
- 5. 大学、大学院等への通学
- 6. その他（具体的に)
- 7. とくに取り組んでいない

以降の設定には、過去に経験したプロジェクトを総合評価して、あるいは直近のプロジェクトを評価してご回答ください。

問5 ソフトウェア開発の生産性を向上させるうえで、開発のフェーズ別に課題の有無を「5（とても課題が多い）」から「1（まったく課題がない）」までの5段階で評価してください。

	とても課題が多い ← 5 4 3 2 1 → まったく課題がない				
要求仕様確定フェーズ	5	4	3	2	1
詳細設計フェーズ	5	4	3	2	1
プログラム生成フェーズ	5	4	3	2	1
デバッグ/テストフェーズ	5	4	3	2	1
その他（具体的に)	5	4	3	2	1

問6 開発プロジェクトと顧客（発注者）との関係についてお答えください。あてはまる番号に1つだけ○をおつけください。

- 6-1. 顧客の要求する仕様は明確ですか？
 - 5. 非常に明確である
 - 4. おおむね明確である
 - 3. ふつう
 - 2. あまり明確でない
 - 1. まったく明確でない
- 6-2. 要求仕様を確定するうえで、顧客は関与しますか？
 - 5. すべて関与する
 - 4. おおむね関与する
 - 3. ふつう
 - 2. あまり関与しない
 - 1. まったく関与しない
- 6-3. 顧客はあなた（もしくは開発チーム）の設計を理解していますか？
 - 5. 完全に理解している
 - 4. おおむね理解している
 - 3. ふつう
 - 2. あまり理解していない
 - 1. まったく理解していない
- 6-4. 顧客は開発したソフトウェアのテストに関与しますか？
 - 5. すべて関与する
 - 4. おおむね関与する
 - 3. ふつう
 - 2. あまり関与しない
 - 1. まったく関与しない

問7 ソフトウェア開発における開発方法の認知と導入、効果についてお聞きします。5段階で評価し、それぞれ該当するマスの数字に○をおつけください。

7-1. 開発プロジェクト全体の方法論について

方法論の認知	よく知っている 5 4 3 2 1 → まったく知らない				
ウォーターフォールモデル	5	4	3	2	1
プロトタイプ	5	4	3	2	1
スパイラル型	5	4	3	2	1
アジャイル型（XP、スクラムなど）	5	4	3	2	1
その他（具体的に)	5	4	3	2	1

方法論の使用	よく使っている 5 4 3 2 1 → まったく使わない				
ウォーターフォールモデル	5	4	3	2	1
プロトタイプ	5	4	3	2	1
スパイラル型	5	4	3	2	1

裏面へおすすみください

アジャイル型（XP、スクラムなど）	5	4	3	2	1
その他（具体的に)	5	4	3	2	1

方法論の効果	とても効果がある	ふつう	まったく効果がない		
ウォーターフォールモデル	5	4	3	2	1
プロトタイピング	5	4	3	2	1
スパイラル型	5	4	3	2	1
アジャイル型（XP、スクラムなど）	5	4	3	2	1
その他（具体的に)	5	4	3	2	1

7-2. 設計フェーズの方法論について

方法論の認知	よく知っている	ふつう	まったく知らない		
構造化分析設計	5	4	3	2	1
オブジェクト指向分析設計	5	4	3	2	1
データ中心アプローチ	5	4	3	2	1
その他（具体的に)	5	4	3	2	1

方法論の使用	よく使っている	ふつう	まったく使わない		
構造化分析設計	5	4	3	2	1
オブジェクト指向分析設計	5	4	3	2	1
データ中心アプローチ	5	4	3	2	1
その他（具体的に)	5	4	3	2	1

方法論の効果	とても効果がある	ふつう	まったく効果がない		
構造化分析設計	5	4	3	2	1
オブジェクト指向分析設計	5	4	3	2	1
データ中心アプローチ	5	4	3	2	1
その他（具体的に)	5	4	3	2	1

問8 あなたが過去に使ったことがある開発ツールについてその経験の有無と生産性向上の効果についてお答えください。（使用経験のあるツールの枠の○をつけ、効果について評価してください）

	経験あり	とても効果が高い ←	ふつう	→	まったく効果がない	
プロジェクト管理ツール		5	4	3	2	1
構成管理ツール		5	4	3	2	1
設計支援ツール		5	4	3	2	1
ドキュメント作成ツール		5	4	3	2	1
デバッグ/テストツール		5	4	3	2	1
CASE ツール		5	4	3	2	1
コードジェネレータ		5	4	3	2	1
その他（具体的に)		5	4	3	2	1

問9 ソフトウェアの開発生産性と次の各項目関係について、「5（強くそう思う）」から「1（まったく思わない）」までの5段階で評価してください（記入例をご参考にしてください）。

	強く思う ← ← ← ← ← ふつう → → → → → まったく思わない				
開発方法論の導入は生産性の向上に寄与する	5	4	3	2	1
開発ツールの導入は生産性の向上に寄与する	5	4	3	2	1
会社が各種の公的認証 (ITSS、CMM、SI 事業所登録等) を受けることは生産性の向上に寄与する	5	4	3	2	1
品質管理の強化は生産性の向上に寄与する	5	4	3	2	1
自分もしくは開発メンバーが情報系の各種資格を取得することは生産性の向上に寄与する	5	4	3	2	1
顧客 (発注者) とのコミュニケーションは生産性の向上に寄与する	5	4	3	2	1
その他 (具体的に)	5	4	3	2	1

問10 工期管理について、過去に計画工期に対して実績工期が遅延したことがありますか？

1. ある 2. ない
 ↓ (その理由の番号に○をつけてください、複数選択可)
 1 システム化目的不相当 2 RFP(Request For Proposal) 内容不相当 3 要件仕様の決定遅れ
 4 要件分析作業不十分 5 開発規模の増大 6 自社内メンバーの選択不相当
 7 業務委託会社選択ミス 8 構築チーム能力不足 9 テスト計画不十分
 10 受入検査不十分 11 総合テストの不足 12 プロジェクトマネージャの管理不足
 13 その他 ()

問11 開発プロジェクトで作業量の「見積もり」で何らかのトラブルが生じたことがありますか？

1. ある 2. ない
 ↓ (その原因の番号に○をつけてください、複数選択可)
 1 要件があいまい 2 仕様の変更が多い 3 見積もりの作成・評価にかかる時間がない
 4 客先に納得してもらえない積算根拠が示せない 5 開発に使用する技術・製品が多岐にわたりすぎる
 6 開発チームのメンバーの力量把握が不十分 7 客先との理解の齟齬 (くいちがい)
 8 その他 ()

問12 あなたが過去に使ったことがある「見積もり」技法についてその経験の有無と生産性向上の効果についてお答えください。(使用経験のある技法の枠の○をつけ、効果について評価してください)

	経験あり	とても効果が高い ← ← ← ← ← ふつう → → → → → まったく効果がない				
類推法 (概算法)		5	4	3	2	1
ファンクション・ポイント法		5	4	3	2	1
WBS(WorkBreakdownStructure)法		5	4	3	2	1
LOC(Line of Code)法		5	4	3	2	1
ユースケース・ポイント法		5	4	3	2	1
COCOMO・COCOMO II など		5	4	3	2	1
特に利用していない		5	4	3	2	1
その他 (具体的に)		5	4	3	2	1

問13 ソフトウェアの開発生産性を向上させるために、有効と思われることがありましたらご記入ください。

ご協力誠にありがとうございました。6月26日 (金) までに同封筒にてご投函ください。