SURE 静岡大学学術リポジトリ
Shizuoka University REpository

地方ソフトウェア業とソフトウェア工学

メタデータ	言語: ja 出版者: 静岡大学人文学部 公開日: 2011-06-28 キーワード (Ja): キーワード (En): 作成者: 伊東, 暁人 メールアドレス: 所属:
URL	https://doi.org/10.14945/00005736

論　説

地方ソフトウェア業とソフトウェア工学

伊　東　暁　人

はじめに
１．調査からみた静岡県内ソフトウェア企業の概要
２．ソフトウェア工学と生産性
３．経営管理者の意識
おわりに

はじめに

コンピュータがこの世に登場して60余年、その間、コンピュータが情報処理の対象とする分野は著しく拡大し、時間を経るごとにその規模、複雑度、重要性は加速度的に増してきた。たとえば、携帯電話に組み込まれているコンピュータソフトウェア（≒プログラム）の量は10余年で約40倍となり、いわゆる第３世代といわれる携帯電話のプログラム規模は約500万行～1,000万行に達しているが、これは行数規模でみると1980年代後期から導入が進んだ都市銀行の第３次オンラインシステムの規模に匹敵する。また、近年製造されている標準的な自動車１台に搭載されているソフトウェアの規模は、90年代後期からの約10年間でおよそ1,000倍になったといわれている。[1]

しかし一方で、このように拡大し続けるソフトウェアの需要に対して質量ともにその供給能力の不足が慢性的に指摘され続けてきた。ソフトウェアの需要と供給能力のギャップは、いわゆる「ソフトウェア危機」[2]として繰り返し言われてきた。1960年代に最初のソフトウェア危機が言われてから、そのギャップを埋めるためにいかにしてソフトウェアの生産性を上昇させるかが課題となり、ソフトウェア工学（Software Engineering）が誕生し工学的な技法の導入によるさまざまな解決の道が探られてきた。こうしてコンピュータソフトウェアを作ることが産業として成立しソフトウェ

[1] 小川計介「標準化で開発効率を高める車載ソフト巨大化に立ち向かう」『日経Automotive Technology』2007年11月号、pp82-97

[2] ソフトウェア危機が最初に言われたのは、1960年代後期であった。1968年10月には当時急速に規模が拡大し複雑性を増しつつあったソフトウェア開発に対応するための会議“Software Engineering Conference”がNATO科学委員会主催で開催されソフトウェア工学という言葉が初めて登場した。

ア開発が工学的な議論の俎上に上がってから40余年が過ぎたが、コンピュータシステムの利用者の期待に対する満足度はいまだ十分に向上していない[3]のが現状である。開発企業は機能強化、品質向上、納期短縮、コスト削減など、ともすれば相反する要素を満たしながらより大きなソフトウェアの開発を行わなければならず、いかにして生産性を向上させるかが課題となっている。

本稿の目的は、ソフトウェアを開発する企業において、その生産性に影響を与える経営管理上の要因を明らかにすることである。そのために、2009年６月に静岡県下のソフトウェア企業を対象に実施したアンケート調査のデータを用い、要因別に生産性の差の検証を行なう。

１．調査からみた静岡県内ソフトウェア企業の概要

本調査は2009年６月に静岡県内の対象企業445社に各社の経営者/事業所長を対象としたアンケートを各１通、従業員規模に応じてプロジェクト管理者むけアンケートを各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、郵送にて回収を行った。回収数は経営者/事業所長向けが119（回収率26.7％）、プロジェクト管理者向けが124（回収率21.21％）である。（調査の実施方法等の詳細については、すでに『経済研究』14巻２号（2009年10月）において公表済みであるのでそちらを参照されたい。ただし、前回集計後に若干の回収が追加されている。）

回答を得られた企業の直前期売上高の平均は1433.97百万円、営業利益額の平均は41.92百万円、過去３年の平均利益率は8.00％である。また、ソフトウェア開発が売上の50％を超える企業（73社）に限定すると、営業利益率の平均は8.85％である。

平成20(2008)年特定サービス産業実態調査（確報）・県別の「ソフトウェア業」によると、静岡県内のソフトウェア企業については、259事業所、従業者数7,981人、売上高111,757百万円、営業費用97,154百万円となっている。また、情報処理推進機構（IPA）がソフトウェア・情報処理サービス業界の経営状況を調査してきた情報処理産業経営実態調査（全国）の2007年度調査[4]によると2006年におけるソフトウェア業[5]の１社当たり平均売上高は3,669百万円、売上高営業利益率は4.26％となっている。これらを比較すると、表１のようになる。

全国にソフト会社は１万社以上あり、営業利益率は平均約４～７％、その８割が年商10億円未満だが、年商10億円未満の７割超が営業利益率の平均値を下回っている。[6]また、IPAの情報処理産業経営実態調査によるソフトウェア業の1999～2006年までの平均営業利益率は約5.1％となっている。

---

[3] (社)日本情報システム・ユーザー協会（JUAS)「ユーザー企業　ソフトウェアメトリクス調査」(各年度)、日経BP社「日経コンピュータ顧客満足度調査」(各年度）など。

[4] (独)情報処理推進機構『第29回情報処理産業経営実態調査報告書』(2008年３月）なお、本調査は第29回をもって終了しているので本調査が最新となる。

[5] 受注ソフトウェア開発、またはソフトウェアプロダクト販売が過半を超える企業

[6] 『日経ソリューションビジネス』(日経BP社）の各年度調査、田中克己『IT産業再生の針路』(日経BP社、2008年）など。

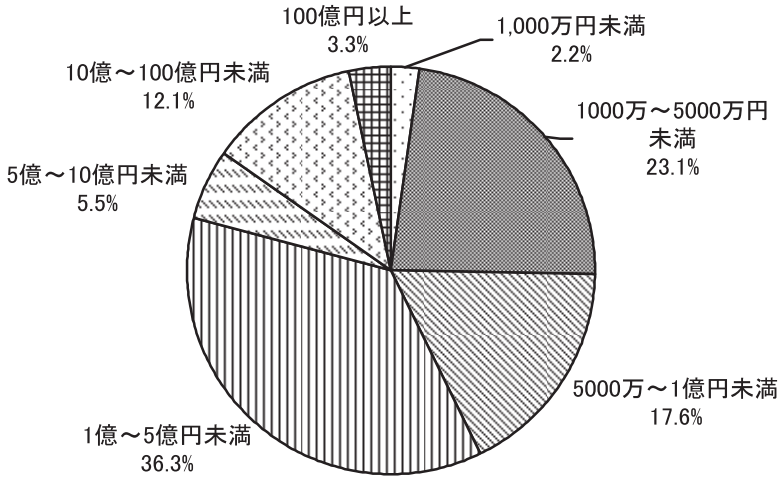
表１．各調査の比較

	本調査	特定サービス産業実態調査（2008年度確報・静岡県）	IPA経営実態調査（2006年・全国）
事業所数	119	259	587
平均売上高（百万円）	1434	432	3669
平均営業利益額（百万円）	42	56（計算値）	156
平均営業利益率（％）	8.00	13.07（計算値）	4.26
平均従業員数（人）	－	31	164

これらを比較すると、本調査の営業利益率の数値は、特定サービス産業実態調査の数値（約13％）ほどではないにせよ、高くなっていることがわかる。

回答のあった企業91社の直前期売上高の分布は図１の通りである。全国平均と同様、約８割強の企業の売上高が10億円未満である。

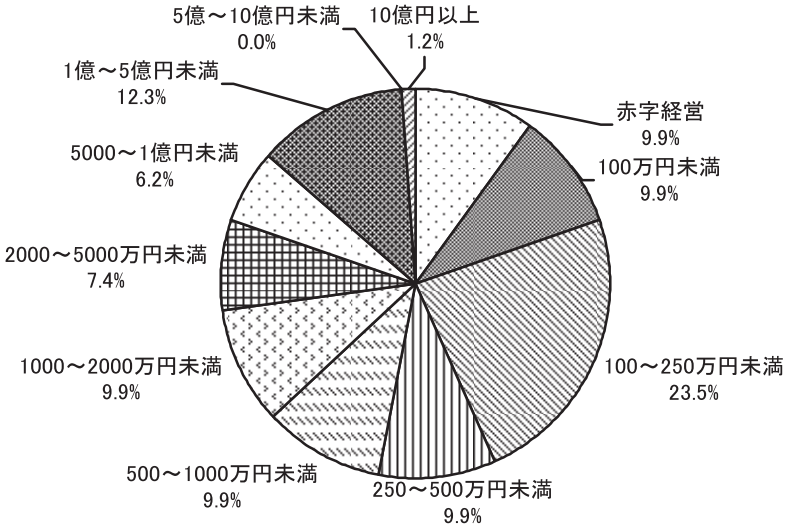
図１．直前期売上高（ｎ＝91）



売上構成を見ると、回答のあった103社のうち、約７割の73社がソフトウェア開発が50％以上となっている。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件数）は、回答のあった107社のうち、約７割が30件未満である。また契約別割合では、回答のあった109社のうち、６割強の67社がユーザー企業からの受託開発の売上が50％以上となっており、同業他社からの受託開発が50％以上は23社となっている。

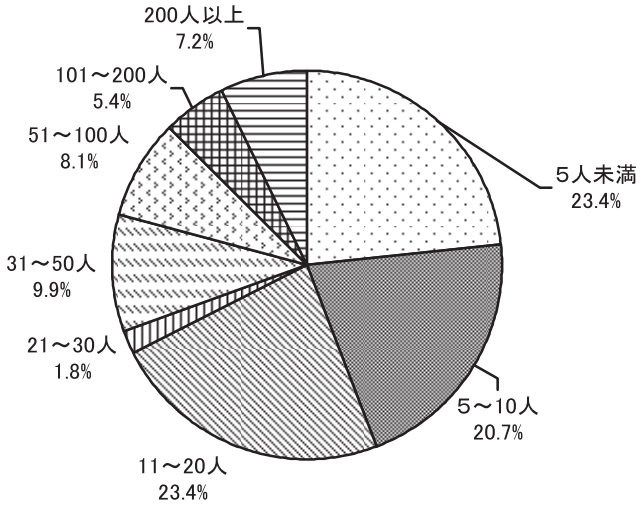
回答のあった企業81社の直前期営業利益額の分布は図２の通りである。約８割の企業の営業利益額が5000万円未満である。

図２．営業利益額（ｎ＝81）



回答のあった企業111社の従業員数の分布は図３の通りである。約７割弱の企業の従業員数が中小企業基本法で定める零細企業（製造業）である20人以下で、サービス業としての零細企業である５人未満が２割を超える。

図３．従業員数（ｎ＝111）



2．ソフトウェア工学と生産性

前述のとおり1960年代からソフトウェアの生産性向上が問題となり、それと同時に、ソフトウェアの生産性をどのように計測すべきかについても多くの議論がなされてきたが、いまだ決定的な方法が確立されていないのが現状である。生産性＝OUTPUT／INPUTであるが、まずOUTPUTが何なのかということが問題になる。目に見えないソフトウェアが目に見える形として生産されるのはプログラムの記述によってである。それゆえ、プログラムのソースコードの行数（SLOC: source lines of code）を生産物の規模や生産性の指標とすることが考えられ、実際、その簡便さから多く用いられてきた。しかし、SLOCは、同じ機能を持ったプログラムであってもプログラマの力量やアルゴリズムの違いにより記述行数が大きく異なり、冗長な記述のほうが規模が大きいものとして計測されることになる。また、プログラム言語が異なる場合は比較が困難である。

1970年代後期、SLOCの欠点を補う方法としてFP法（function point method）が提唱された。これはソフトウェアに実装される機能をその処理の複雑さに応じて点数（ポイント）化して積算するものである。IFPUG（International Function Point Users Group）が提唱する手法やCOSMIC-FFP法などがある。プログラム言語の違いや開発環境、プログラマの力量に左右されないという利点がある半面、計測のために一定のスキルが必要とされることや計測が煩雑で時間がかかるといった点が欠点として指摘されている。

さらに、そのソフトウェアを稼働するように実装させるのに必要となる資源（ソフトウェア開発の場合には、その多くが人的労働である）の量によって規模を推計する手法として広く用いられているのが人月（man-month）に代表される工数である。工数（一般に、工数＝時間×人）は従来から製造業の作業量を示す概念として使われてきたが、概念が単純なこともあり、ソフトウェア開発業界においてもシステム構築の際の見積もり費用積算などに応用されている。しかし、人月による管理については、担当するエンジニア・プログラマなどの力量の差を十分に反映していない、人と時間は交換不可能である（遅れているソフトウェアプロジェクトへの要員追加はさらに遅らせる[7]）といった様々な限界が以前から指摘されている。

ほかにも、SLOCと工数の間に一定の関係性を想定し、エンジニアやプログラマの能力等を係数化して規模を算出するCOCOMO（COnstructive COst MOdel）[8]やCOCOMOⅡ等がおもに見積もり手法として規模（工数・期間）を推定するために使用されている。

本稿では調査対象とした各企業の開発適用分野、開発方法や使用プログラム言語などが異なることを考慮し、ソフトウェアが生み出す価値（売価）に着目して、各企業の売上高営業利益率をもって生産性の指標としている。

---

[7] いわゆる「ブルックスの法則」。後述のF.P.Brooks Jr.『人月の神話』参照。
[8] Barry Boehmが1981年に提唱。

今回の調査では、経営者・事業所長（以下、経営管理者と記す）がソフトウェアの生産性を管理することについて回答を求めた。回答があった110事業所のうち、94の事業所の代表が「とても重要である」もしくは「ある程度重要である」と回答しており、８割強の経営管理者がソフトウェア開発において生産性管理が重要であることを認識していることが分かった。しかし、開発案件の一部であれ何らかの方法でソフトウェア開発の生産性を測定していると回答している企業の割合は８割弱にとどまり、約２割の企業は「全く測定していない」状況にある。生産性管理の重要度の認識と生産性の測定の有無の関係を表にまとめたものが表２である。

表２．生産性管理の認識と測定の関係

計測の度合→ ↓重要性の認識	全てのプロジェクトで計測	半数以上のプロジェクトで測定	特定のプロジェクトのみで測定	全く測定していない	その他	小計
とても重要である	25	8	19	6	0	58
ある程度重要である	7	11	11	9	1	39
普通	3	2	2	6	0	13
あまり重要でない	0	0	0	2	0	2
全く重要でない	0	0	0	1	0	1
小計	35	21	32	24	1	113

「生産性を測定している」と回答した企業のうち、85社からはその方法について回答が得られた。約９割の企業（78社）が工数（人月）ベースで測定をしており、コード行数（SLOC）ベースが13社、FPベースが10社、その他が２社となっている。

生産性を測定している企業と測定していない企業で生産性（売上高営業利益率）に差が生じているかどうかを検定したものが表３と表４である。検定に用いたデータは、売上高に占めるソフトウェア開発の割合が50％以上で利益率が把握できる（平均利益率の回答がある場合はその数値、ない場合は直前期売上高と営業利益額から計算）もので、±20％以上の利益率を外れ値として除外（７社）した51社で、その平均営業利益率は3.69％である。（以降、表中の有意確率は、＊ $p<0.05$　＊＊ $p<0.01$である。）

表３．生産性の測定と利益率(1)

	「すべて」＆「半数以上」測定	「特定のみ」＆「測定せず」	等分散性の検定		t 検定	
データ数	20	31	統計量：F	1.0688	統計量：t	0.2718
平均	0.035	0.038	自由度１	19	自由度	49
不偏分散	0.001	0.001	自由度２	30	両側Ｐ値	0.7869
標本標準偏差	0.037	0.038	Ｐ　値	0.8988	片側Ｐ値	0.3935

表３は「すべて測定」と「半数以上測定」のグループ20社と「特定のプロジェクトのみ」と「全

く測定していない」のグループ31社の２つに分けて平均利益率の差を検証したものである。前者のグループの平均営業利益率が3.5％であるのに対し、後者のそれは3.8％と後者の方がやや高い値を示しているが有意な差は認められない。

表４．生産性の測定と利益率(2)

	「すべて」＆「半数以上」＆「特定のみ」測定	「測定せず」	等分散性の検定		t 検定	
データ数	45	6	統計量：F	1.3594	統計量：t	1.6434
平均	0.034	0.060	自由度１	44	自由度	49
不偏分散	0.001	0.001	自由度２	5	両側Ｐ値	0.1067
標本標準偏差	0.037	0.032	Ｐ　値	0.7985	片側Ｐ値	0.0533

表４は「すべて測定」と「半数以上測定」と「特定のプロジェクトのみ」を合わせたグループ45社と「全く測定していない」のグループ６社の２つに分けて平均利益率の差を検証したものである。前者のグループの平均営業利益率が3.4％であるのに対し、後者のそれは6.0％と後者の方がかなり高い値を示す結果となったが、こちらも有意な差とは認められない。以下、それぞれの計測手法と利益率について見て行きたい。

表５．工数（人月）と利益率

	人月を使用	人月使わず	等分散性の検定		t 検定	
データ数	35	3	統計量：F	2.2249	統計量：t	0.0827
平均	0.034	0.036	自由度１	34	自由度	36
不偏分散	0.002	0.001	自由度２	2	両側Ｐ値	0.9346
標本標準偏差	0.040	0.027	Ｐ　値	0.7166	片側Ｐ値	0.4673

なんらかの測定をしている企業45社のうち、38社から測定方法について回答が得られた。（複数回答可）38社のうち約９割の35社が工数（人月）による測定手法を採用しているが、利益率を見ると採用していない企業のほうがやや高い傾向がみられるが有意な差は認められない。

表６．コード行数（SLOC）と利益率

	SLOC使用	SLOC使用せず	等分散性の検定		t 検定	
データ数	6	32	統計量：F	3.9332	統計量：t	0.5151
平均	0.027	0.036	自由度１	5	自由度	36
不偏分散	0.000	0.002	自由度２	31	両側Ｐ値	0.6097
標本標準偏差	0.021	0.041	Ｐ　値	0.1311	片側Ｐ値	0.3048

６社がコード行数（SLOC）を用いて測定を行っている。比較では、SLOCを使っていない回答群のほうが利益率が高くなったが、有意な差ではない。

表７．FP法と利益率

	FP使用	FP使用せず	等分散性の検定		t 検定	
データ数	4	34	統計量：F	1.1538	統計量：t	1.1204
平均	0.055	0.032	自由度１	3	自由度	36
不偏分散	0.002	0.001	自由度２	33	両側Ｐ値	0.2700
標本標準偏差	0.041	0.038	Ｐ　値	0.6841	片側Ｐ値	0.1350

４社がFP法による測定を行っており、FP法を使っている回答群のほうが生産性（利益率）が高いことが示されているが、有意な差は認められない。

これらのデータを見る限りでは、いずれの方法であっても生産性を測定することは生産性（ここでは利益率）を改善することには必ずしもつながらないことが推定される。一般に製造業の世界では「測れないものは改善できない」[9]というのが常識となっていて、まず計測することが品質管理や生産性向上の出発点とされている。ソフトウェア開発の世界でも「測れないものは制御できない」[10]と言われてきた。経済産業省も「技術戦略マップ」のなかで、「民間企業内で実施されている実際のソフト開発プロジェクトに関する社内データ、顧客データ等の収集、評価、分析が必要」[11]としている。しかし、上記の分析は必ずしもそれが適用できない可能性―ソフトウェアの特殊性を示していると思われる。

ソフトウェアは一般的な工業製品とは異なる特質をもつ。1970年代に『人月の神話（The Mythical Man-Month: Essays on Software Engineering)』[12]でソフトウェアの開発管理の問題を論じたブルックス（F.P.Brooks Jr.）はその著書の中でアリストテレスの哲学概念を援用して、ソフトウェアは概念構造体（conceptual construct）としての「本質的（essential）性質」と周辺的な「偶有的（accidental）性質」を持つとしている。そして、その本質的性質のゆえに、４つの本質的困難（difficulty）―「複雑性（complexity）」、「同調性（conformity）」、「可変性（changeability）」、「不可視性（invisibility）」を持つと主張する。そして、これらの困難さが自動化や生産性を上げることを妨げると述べ、「銀の弾丸などない」[13]（＝ソフトウェア開発の生産性を劇的に改善させる方法はない）としている。

デマルコは前述のように1980年代初からソフトウェア開発における工学的技法の導入（＝計測し

[9] William Edwards Demingの言葉と言われている。

[10] "*Controlling Software Projects*", Tom DeMarco, 1982（邦題：品質と生産性を重視したソフトウェア開発プロジェクト技法―見積り・設計・テストの効果的な構造化）の中の言葉。"You can't control what you can't measure."

[11] 経済産業省「技術戦略マップ2009」の「ソフトウェア分野」（http://www.nedo.go.jp/roadmap/index.html、2009年）

[12] "*The Mythical Man-Month: essays on software engineering*", Brooks, F. P., Addison Wesley, 1975, 1995 邦訳：『人月の神話　狼人間を撃つ銀の弾はない』（原著発行20周年記念増訂版、新装版）、フレデリック・P・ブルックスJr.、滝沢・牧野・宮沢（訳）、ピアソン・エデュケーション、2002年

[13] "*No Silver Bullet - essence and accident in software Engineering*", Brooks, F. P., Proceedings of the IFIP Tenth World Computing Conference, pp. 1069-1076, 1986.

コントロールすること）を長年主張してきたが、近年ではそれだけでは限界があり、ソフトウェア開発における問題の核心はすでにそこから移動していることを述べている。

Software development is and always will be somewhat experimental. The actual software construction isn't necessarily experimental, but its conception is. And this is where our focus ought to be. It's where our focus always ought to have been.[14]

ソフトウェア開発がはたして工学的な生産手法・管理に馴染むものなのかということについては以前から多くの議論がある。たとえば、マクブリーンはその著書[15]の中で、「工業製品の工場での生産にあたるのは、ソフトウェアにおいては出来上がってテストを終了したバイナリをCD-ROMにコピーしているときである。コーディングやテストですら、工業製品で言えば開発・試作工程にあたる。ゆえに、工学（エンジニアリング）の世界での大量生産の成功例を当てはめる試みはうまくいかない。」と述べ、ソフトウェア工学の持つ限界と問題点を指摘し、ソフトウェアが人によって作られるものであるという基本に回帰する必要性を提唱している。

とはいうものの、労働集約的といわれながらもこれまでにこれだけの膨大なソフトウェアを生産することを可能にしてきたのは、ソフトウェア工学が新たなプログラム言語やさまざまな開発ツール、開発方法論などを生み出すことで偶有的複雑性の面ではその克服と改善に成果を出してきた結果といえよう。定型化された業務を処理するような比較的構造が明確であるソフトウェア開発の場合には、パッケージ化されたソフトウェアを基盤として開発を行ったり、過去のソースコードの流用、部品化とその共通化、コードジェネレータ、あるいは完成したプログラムのテストツール利用といったことが有効である。[16]後述のように今回の調査においても、経営者、プロジェクト管理者ともに開発ツールの導入が生産性の向上に寄与することについては肯定的であった。今回、調査の中で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を自由記述してもらったが、その中にも「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」、「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」といった意見が散見された。これらのことからも、（ウォーターフォールモデルで言うところの）「下流工程」における生産性は不十分ながらも向上しているが、要求仕様の確定、設計といった「上流工程」は依然として工学的な手法による生産性向上は困難であることが推察される。

以下、本稿では計測以外の経営管理の要素と生産性（利益率）の関係を見ていく。

---

[14] "*Software Engineering: An Idea Whose Time Has Come and Gone?*", DeMarco, T. IEEE Software, July/August 2009, pp. 96–95

[15] "*Software Craftsmanship: the New Imperative*", McBreen, P. Addison-Wesley, 2002（邦訳：「ソフトウェア職人気質」ピアソン・エデュケーション、2002年）

[16] 福浩邦，笹島三佳代「システムの信頼性向上に関する調査―ソフトウェア工学的アプローチ＆失敗学的アプローチ」（IPA、2002年）など。

表８．品質保証体制と生産性

	体制あり	体制なし	等分散性の検定		t 検定	
データ数	21	30	統計量：F	1.2883	統計量：t	1.7225
平均	0.047	0.030	自由度１	20	自由度	49
不偏分散	0.002	0.001	自由度２	29	両側Ｐ値	0.0913
標本標準偏差	0.0039	0.034	Ｐ　値	0.5227	片側Ｐ値	0.0456*

対象51社のうち、品質基準の設定や第三者レビューなどの品質保証の体制をとっている会社が21社であったが、体制をとっていない会社の利益率が3.0％であるのに対し、とっている会社の利益率は4.7％と高く（５％水準で有意）、品質を向上させることが結果的に生産性の上昇につながる可能性を示している。

表９－１．人事考課への生産性の反映（個々の社員に反映）

	反映している	反映していない	等分散性の検定		t 検定	
データ数	30	20	統計量：F	2.0075	統計量：t	0.1304
平均	0.037	0.038	自由度１	29	自由度	48
不偏分散	0.002	0.001	自由度２	19	両側Ｐ値	0.8968
標本標準偏差	0.042	0.029	Ｐ　値	0.1164	片側Ｐ値	0.4484

回答のあった50社のうち、開発生産性を個々の従業員レベルで人事考課に反映させていると回答した企業が30社であった。反映させている企業の利益率は3.7％、反映させていない企業の利益率は3.8％で、わずかに個々に反映させていないが利益率が高くなる傾向があるが、有意な差ではない。

表９－２．人事考課への生産性の反映（プロジェクトチーム単位に反映）

	反映している	反映していない	等分散性の検定		t 検定	
データ数	9	41	統計量：F	12.6822	統計量：t	2.3956
平均	0.012	0.043	自由度１	8	自由度	48
不偏分散	0.000	0.001	自由度２	40	両側Ｐ値	0.0205*
標本標準偏差	0.011	0.038	Ｐ　値	0.0008**	片側Ｐ値	0.0103*

対象50社のうち、開発生産性を開発プロジェクトのチーム単位で人事考課に反映させていると回答した企業が９社であった。反映させている企業の利益率は1.2％、反映させていない企業の利益率は4.3％で、プロジェクトチーム単位で反映させていないほうが利益率が高くなる傾向（５％水準で有意）が確認できた。（ｔ検定：Welchの方法でも両側Ｐ値：0.0001**、片側Ｐ値：0.00003**）

表９－３．人事考課への生産性の反映（部課レベルで反映）

	反映している	反映していない	等分散性の検定		t 検定	
データ数	8	42	統計量：F	1.0466	統計量：t	0.1414
平均	0.036	0.038	自由度１	7	自由度	48
不偏分散	0.001	0.001	自由度２	41	両側Ｐ値	0.8882
標本標準偏差	0.037	0.037	Ｐ　値	1.0483	片側Ｐ値	0.4441

対象50社のうち、開発生産性を部課など開発プロジェクトを超えた組織レベルで人事考課に反映させていると回答した企業が８社であった。反映させている企業の利益率は3.6％、反映させていない企業の利益率は3.8％で、部課レベルで反映させているほうが利益率がやや低くなる傾向が示されたが有意な差とは確認できない。

表９－４．人事考課への生産性の反映（反映させていない）

	まったく反映させていない	なんらかの反映をさせている	等分散性の検定		t 検定	
データ数	12	38	統計量：F	1.6931	統計量：t	0.2345
平均	0.035	0.038	自由度１	11	自由度	48
不偏分散	0.001	0.002	自由度２	37	両側Ｐ値	0.8156
標本標準偏差	0.030	0.039	Ｐ　値	0.3520	片側Ｐ値	0.4078

対象50社のうち、開発生産性を人事考課にまったく反映させていないと回答した企業が12社であった。反映させている企業の利益率は3.8％、反映させていない企業の利益率は3.5％で、何らかの形で人事考課に反映させているほうが利益率がやや高くなる傾向が示されたが有意な差とはいえない。

表10．標準化された開発方法と生産性

	導入している	導入していない	等分散性の検定		t 検定	
データ数	27	22	統計量：F	1.2660	統計量：t	0.4078
平均	0.036	0.040	自由度１	26	自由度	47
不偏分散	0.002	0.001	自由度２	21	両側Ｐ値	0.6853
標本標準偏差	0.040	0.035	Ｐ　値	0.5860	片側Ｐ値	0.34427

対象51社のうち、ソフトウェア開発において標準化されたな開発方法を導入していると回答した企業が27社、導入していないと回答した企業が22社（無回答２社）であった。導入している企業の利益率は3.6％、導入していない企業の利益率は4.0％で、標準化された開発方法を導入しない企業の方が利益率が高くなるという傾向が示されたが有意な差とは確認できない。（別の条件で抽出し

たデータ群では逆の傾向が示された。）

表11．資格の取得を人事考課に反映させることと生産性

	反映させている	反映させていない	等分散性の検定		t 検定	
データ数	21	30	統計量：F	2.2415	統計量：t	2.2168
平均	0.024	0.046	自由度１	20	自由度	49
不偏分散	0.001	0.002	自由度２	29	両側Ｐ値	0.0313*
標本標準偏差	0.027	0.040	Ｐ　値	0.0644	片側Ｐ値	0.0157*

従業員がソフトウェア開発にかかわる資格（技術士、情報処理技術者など）した場合、それを人事考課に反映させているかを尋ねた。51社のうち、21社が何らかの形で反映させていると回答したのに対し、30社は反映させていないと回答している。利益率を比較してみると、反映させている回答群が2.4％であるのに対し、反映させていない方は4.6％となり、この差は有意な差（５％水準）と認められる。（ｔ検定：Welchの方法でも両側Ｐ値：0.0215*、片側Ｐ値：0.0108*）

以上、表３～表11に示されたものと同様の検定を別の抽出条件によるデータ群[17]で行ったが、ほぼ同様の結果となった。

IPAが集計しているデータに基づく分析では、一般に規模の小さいプロジェクトは生産性のばらつきが大きく、プロジェクト管理者のスキルの分布もばらつく傾向がみられる。また、規模の大きいプロジェクトは生産性が低くなる傾向がみられるのと同時に、プロジェクト管理者のスキルが高い傾向がみられる。[18]今回分析に使用したデータも比較的規模が小さいプロジェクトが多いため、生産性のばらつきが大きくなった可能性が予想される。

ここまでの分析をまとめると、

１．生産性を測定することは生産性（利益率）の向上には直接的には結び付かない。
２．品質保証体制をとることは生産性向上に結び付く可能性がある。
３．生産性と人事考課の間に関係を見出すことは難しい。むしろプロジェクトチーム単位に反映させた場合にはマイナスの効果が出ることもありうる。
４．標準化された開発方法を導入することが生産性の向上に資するとは言えない。
５．ソフトウェア開発にかかわる資格取得を人事考課に反映させることは生産性の向上に資するとは言えない。むしろ反映しないほうが利益率の向上につながる可能性がある。

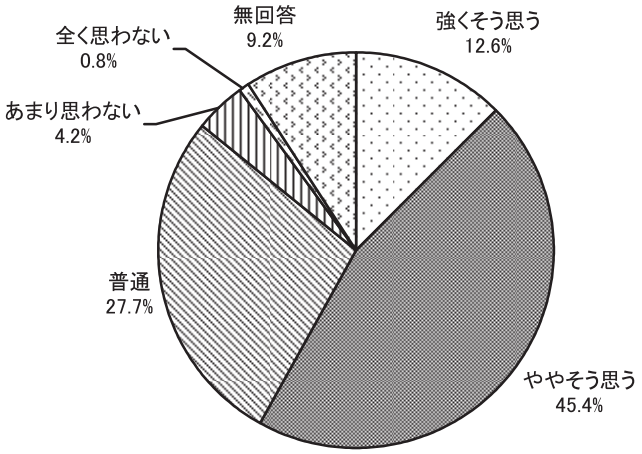
といった、いくつかの点で従来の経営管理の考え方を否定することが明らかとなった。

[17] 売上高40百万円以上、従業員数５人以上、売上に占めるソフトウェア開発比率40%以上（ただし従業員数50人以上の場合はそれ未満も対象）の49社でその平均営業利益率は3.53％である。
[18] (独)情報処理推進機構『ソフトウェア開発データ白書』（各年版、日経BP社）

3．経営管理者の認識

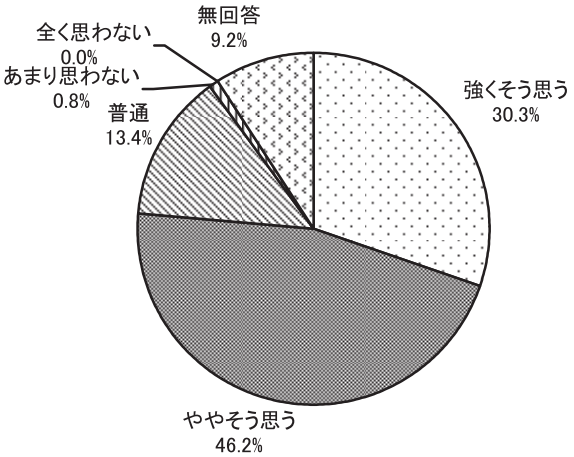
本調査では、経営者が開発生産性とさまざまな要因についてどのように認識しているかについても調査（5段階評価）を行った。

図4．開発方法論の導入は生産性の向上に寄与する（n＝119）



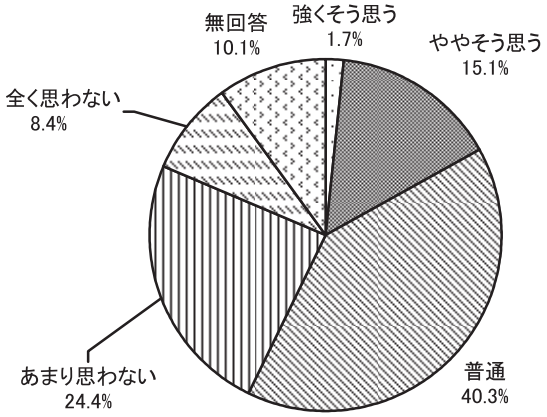
開発方法論の導入については肯定的な認識が半数以上を占める（図4）が、上記の検証はそれを裏付けるものとはなっていない。今回同時に実施したプロジェクト管理者向けのアンケートにおいても、肯定的な認識は5割弱にとどまる。経営管理者の自由記述においても「あらゆる工程で一意的に方法論を適用しようとすると逆に生産性を損なう場合が多々ある」といった意見も見られた。

図5．開発ツールの導入は生産性の向上に寄与する（n＝119）



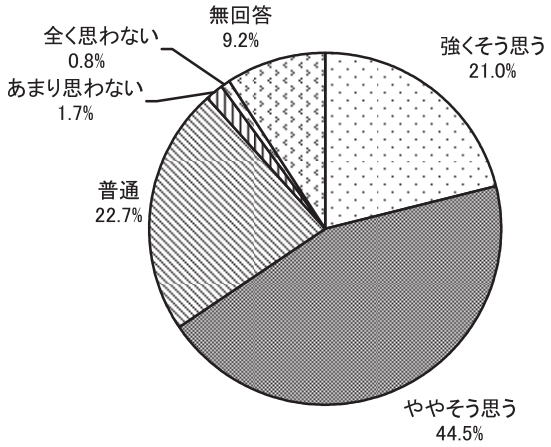
開発ツールの導入については肯定的な認識が７割以上を占める（図５）。プロジェクト管理者向けのアンケートにおいても、開発ツールの導入は開発方法論や公的認証といった他の要因よりも生産性の向上に寄与すると評価する（63.2％）傾向が見られる。しかし、一方で、経営者からは「ツールを使用したドキュメントはユーザーには不評」、「方法やツールなどをどう有効的に運用していくか、またそれらの必要性を一人一人に理解させるかが大切であり、それができなければ何をしてもムダになってしまう。」といった限界も聞かれる。

図６．会社が各種の公的認証（ITSS, CMM, SI事業所登録認定など）を受けることは生産性の向上に寄与する（ｎ＝119）



公的認証と生産性向上の関係については肯定的な認識が２割未満にとどまるという結果になった。むしろ、生産性との関係においては否定的な認識が３割を超えている。（図６）。プロジェクト管理者に対する調査でも否定的な認識が４割を超えている。今回、公的認証の取得の有無、種類等については調査項目から除外したので、公的認証の取得と生産性の関係については検証できないが、もし実際に生産性向上に結び付かないのであれば企業にとって公的資格を取得する動機づけが弱いものになっていることが予想される。

図７．品質管理の強化は生産性の向上に寄与する（ｎ＝119）



品質管理の強化と生産性の向上については肯定的な認識が６割以上を占める（図７）。プロジェクト管理者に対する調査でも６割弱が肯定的な認識を示している。また、上記の検証結果もそれを裏付ける結果を示している。（表８）ソフトウェアの品質管理を強化し、品質を高めることが結果的に無駄な手戻りや修正作業を減少させ、生産性の向上に資するものと考えられる。

おわりに

ソフトウェア開発プロジェクトがうまくいかず、多くの開発技術者が過酷な労働を強いられている状況をヨードンが「デスマーチ」と称してから10余年が過ぎた。[19]この言葉は死語になるどころか（「デスマ」と略され）広く世間に流布され、同時にソフトウェア開発業界は新３Ｋの魅力のない仕事といわれるようになってしまった。多くの優秀な若者がソフトウェア開発業界への就職を忌避する事態となっている。一方、地域経済の停滞に加え、取引の首都圏一極集中化が進行[20]するとともに、オフショア開発の拡大によって仕事の国外流出が進行し、地方ソフトウェア業における状況は以前にもまして一層深刻な状況にある。こうした状況を打開するためにも生産性を上げることは重要な課題である。

今回はおもに経営管理者の認識や管理の在り方を中心に生産性との関係について分析を行ったが、調査のなかでは顧客や関連する諸部門とのコミュニケーションを十分にとることが生産性の向上に重要であるとの指摘が多く寄せられた。今後は、同時に実施したプロジェクト管理者を対象とした

[19] “*Death March: The Complete Software Developer's Guide to Surviving “Mission Impossible” Projects*”, Yourdon, E. Prentice Hall, 1977（邦訳：「デスマーチ－なぜソフトウエア・プロジェクトは混乱するのか」）トッパン、1998年

[20] たとえば静岡県の場合、売上高で2006（平成18）年と平2008（平成20）年を比較すると96.9％（全国は107.0％）となっており、就業者数を見ても東京一極集中がさらに進行している。（特定サービス産業実態調査）

データの分析を行うとともに、生産性の向上に資するコミュニケーションのあり方について検討を行いたい。また、小規模な会社が多いために、生産性の向上が個々の技術者のスキルレベルに依存するという声もいくつか聞かれた。どのようなスキルの獲得が生産性の向上に寄与するのかを明らかにすることも今後の課題であろう。

［参考資料・文献］

・経済産業省「特定サービス産業実態調査」（各年度）
・（社）日本情報システム・ユーザー協会（JUAS）「ユーザー企業　ソフトウェアメトリクス調査」（各年度）
・（独）情報処理推進機構「情報処理産業経営実態調査」（各年度）
・（社）情報サービス産業協会『情報サービス産業白書』（各年度）
・（社）情報サービス産業協会「魅力ある地域情報サービス産業の確立に向けた課題と展望」（2010年）
・（独）情報処理推進機構『ソフトウェア開発データ白書』（各年度、日経BP社）
・玉井哲雄『ソフトウェア工学の基礎』（2004年、岩波書店）
・大槻繁『ソフトウェア開発はなぜ難しいのか「人月の神話」を超えて』（2009年、技術評論社）

＊本稿は、（独）日本学術振興会平成22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。