

Title	中小企業経営陣の戦略的情報システム活用のための能動的学習モデルの構築
Sub Title	Desining an active learning model for strategic operation of information system for small business management
Author	高橋, 良輔(Takahashi, Ryosuke) 当麻, 哲哉(Toma, Tetsuya)
Publisher	慶應義塾大学大学院システムデザイン・マネジメント研究科
Publication year	2011
Jtitle	
JaLC DOI	
Abstract	<p>多くの経営者は近視眼的であると言われる。それは即ち、経営を戦略的に行っていないとも言える。経営を戦略的に行うためには、まずは戦略を策定するための自社や競合相手の内的・外的要因を把握する必要がある。特に現代社会では「己を知る」つまり自社の内的・外的要因（例：自社の業務システム）を知ることが「敵を知る」ことよりも重要である。よって、自社の業務システムを知ることが経営をより戦略的に行うことに繋がるという仮説を立てることができる。</p> <p>自社の業務システムを理解するためのアプローチは色々ある。最も効果的なのはシステムの開発・保守・運用に参加することである。特に開発を経験すると、システムがどのようにして出来上がるののを見ることにもなるため、保守・運用にも役立つ。ただし、開発・保守・運用を行うためには専門的な知識が必要になるので、通常はSierと呼ばれる専門の業者に委託してしまう。しかし、資金や人材に乏しい企業にとっては、専門の業者に委託することもできず、自社で開発することもままならず、いつまで経っても自社の業務システムを詳しく理解することはできないジレンマに陥る。</p> <p>以上の問題を解決するにはどうすれば良いのか？一つの解決策として、顧客能動学習型（CAL: Customers' Active Learning）開発手法というものを提案する。これは、システムを開発する時、より積極的に顧客（特に経営陣）に参加してもらうことによって、システムの要求漏れを防ぎ、保守・運用しやすいシステムを実現すると同時に、システムの仕組みを知ってもらうことで、経営陣が企業経営や業務をより戦略的に行うようになることを目指した手法である。</p> <p>本研究では、まず実際に、ある小企業の業務システムの開発にCAL開発手法を適用して、システムを開発した後の経過も見守った。すると経営陣から、経営をより戦略的に行おうとする兆候が見えるようになった。次に、当該手法の有用性をより客観的に検証するために、200名超の中小企業経営者にアンケートを行った。その結果、当該手法を経験することによって、企業経営や業務をより戦略的に行えるようになる可能性があることが分かった。ITの発達により、今後、世の中のビジネスのスピードは増々速くなると考えられるにも関わらず、上記の要因を深く分析・理解しなければ、自らの利益を自覚できず、戦略を立てることもできない。しかし、CAL開発手法は、そうした問題を解決するための手法になるだろうと期待できる。</p>
Notes	修士学位論文. 2011年度システムエンジニアリング学 第51号
Genre	Thesis or Dissertation
URL	<a href="https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=KO40002001-00002011-0002">https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=KO40002001-00002011-0002</a>

慶應義塾大学学術情報リポジトリ(KOARA)に掲載されているコンテンツの著作権は、それぞれの著作者、学会または出版社/発行者に帰属し、その権利は著作権法によって保護されています。引用にあたっては、著作権法を遵守してご利用ください。

The copyrights of content available on the Keio Associated Repository of Academic resources (KOARA) belong to the respective authors, academic societies, or publishers/issuers, and these rights are protected by the Japanese Copyright Act. When quoting the content, please follow the Japanese copyright act.

中小企業経営陣の戦略的情報システム活用のため  
の能動的学習モデルの構築

高橋 良輔

(学籍番号：80933345)

指導教員 准教授 当麻哲哉

2011年9月

慶應義塾大学大学院システムデザイン・マネジメント研究科  
システムデザイン・マネジメント専攻

# 論 文 要 旨

学籍番号	80933345	氏 名	高橋 良輔
論文題目： 中小企業経営陣の戦略的情報システム活用のための能動的学習モデルの構築			
<p>(内容の要旨)</p> <p>多くの経営者は近視眼的であると言われる。それは即ち、経営を戦略的に行っていないとも言える。経営を戦略的に行うためには、まずは戦略を策定するための自社や競合相手の内的・外的要因を把握する必要がある。特に現代社会では「己を知る」つまり自社の内的・外的要因（例：自社の業務システム）を知ることが「敵を知る」ことよりも重要である。よって、自社の業務システムを知ることが経営をより戦略的に行うことに繋がるという仮説を立てることができる。</p> <p>自社の業務システムを理解するためのアプローチは色々ある。最も効果的なのはシステムの開発・保守・運用に参加することである。特に開発を経験すると、システムがどのようにして出来上がるのを見ることにもなるため、保守・運用にも役立つ。ただし、開発・保守・運用を行うためには専門的な知識が必要になるので、通常は <b>SIer</b> と呼ばれる専門の業者に委託してしまう。しかし、資金や人材に乏しい企業にとっては、専門の業者に委託することもできず、自社で開発することもままならず、いつまで経っても自社の業務システムを詳しく理解することはできないジレンマに陥る。</p> <p>以上の問題を解決するにはどうすれば良いのか？一つの解決策として、顧客能動学習型（<b>CAL: Customers' Active Learning</b>）開発手法というものを提案する。これは、システムを開発する時、より積極的に顧客（特に経営陣）に参加してもらうことによって、システムの要求漏れを防ぎ、保守・運用しやすいシステムを実現すると同時に、システムの仕組みを知ってもらうことで、経営陣が企業経営や業務をより戦略的に行うようになることを目指した手法である。</p> <p>本研究では、まず実際に、ある小企業の業務システムの開発に <b>CAL</b> 開発手法を適用して、システムを開発した後の経過も見守った。すると経営陣から、経営をより戦略的に行おうとする兆候が見えるようになった。次に、当該手法の有用性をより客観的に検証するために、200名超の中小企業経営者にアンケートを行った。その結果、当該手法を経験することによって、企業経営や業務をより戦略的に行えるようになる可能性があることが分かった。ITの発達により、今後、世の中のビジネスのスピードは増々速くなると考えられるにも関わらず、上記の要因を深く分析・理解しなければ、自らの利益を自覚できず、戦略を立てることもできない。しかし、<b>CAL</b> 開発手法は、そうした問題を解決するための手法になるだろうと期待できる。</p>			
キーワード（5語） 経営戦略、情報システム、ソフトウェア、アジャイル、能動学習			

## SUMMARY OF MASTER'S DISSERTATION

Student Identification Number	80933345	Name	Ryosuke TAKAHASHI
<p>Title</p> <p style="text-align: center;">Designing an Active Learning Model for Strategic Operation of Information System for Small Business Management</p>			
<p>Abstract</p> <p>It is said that many managers are myopic. That is, they don't make a strategic management. To do so, it is necessary for them to understand the internal and external factors of themselves and their competitors to plan their strategy at first. Especially in today, it is more important to "know themselves", that is, to understand these factors (ie: their business system) than to "know their enemy". Therefore, I can hypothesize that they will manage their business more strategically with knowing their business system.</p> <p>There are variously approaches for them to understand such a system. The most effective way for them is to participate in the development or maintenance or operation of the system. Especially, if they experience the development, it is also useful for the maintenance and operation because it comes to see how to complete the system. However, enterprises with scarce capitals and talents cannot consign to a professional (called "SIer") or to develop by themselves, and it falls into the dilemma that they cannot understand their own business system forever.</p> <p>How should we do to solve this problem? I propose Customers' Active Learning development Methodology (called "CAL") as one of solutions. This is a methodology that aimed for executives to make more strategic business management and service by having to participate more actively and understand how the system works on developing the system.</p> <p>In this study, I had applied CAL to the development of a business system in a small business and had watched its executives after. Then I have come to find them to go to perform management strategically. Next, in order to more objectively verify the usefulness of such approach, I had conducted a survey to over 200 small business owners. It has been found that having experienced CAL development could help for them to make a more strategic business management and service. The speed of the business will become faster and faster, but if executives cannot understand above factors, they cannot be aware of their own advantage and plan their strategy. I just expect that CAL will become a methodology to solve such a problem.</p>			
<p>Key Word(5 words)</p> <p>Strategic Management, Information System, Software, Agile, Active Learning</p>			

<b>1. 序論</b>	<b>10</b>
1.1. 現在の状況・問題点	10
1.2. 一般的な解決アプローチ	10
1.3. 本研究で主張する解決アプローチ	11
1.4. 本論の構成	11
<b>2. 現状分析</b>	<b>12</b>
2.1. 経営戦略	12
2.1.1. はじめに	12
2.1.2. 定義	12
2.1.3. 分類	12
2.1.4. プロセス	13
2.1.5. 問題点	14
2.2. 情報システム（開発・保守・運用）	15
2.2.1. はじめに	15
2.2.2. ライフサイクル	15
2.2.3. IT費用の推移と割合	16
2.2.4. 保守・運用費の推移と構成	18
2.2.5. 典型的な情報システムにおける機能の利用割合	19
2.2.6. プロジェクトの成功要因	19
2.3. 情報システム（開発手法）	21
2.3.1. システム開発手法の一覧	21
2.3.2. ウォーターフォール開発手法と反復型開発手法	22
2.3.3. 反復型開発手法	23
2.3.4. アジャイル開発手法	25
2.4. 情報システム（まとめ・問題点）	32
2.4.1. まとめ	32
2.4.2. 問題点	33
2.5. 教育・学習	34
2.5.1. 教える	34
2.5.2. 学ぶ	36

<b>3. モデル (CAL 開発手法) の提案</b> .....	<b>38</b>
3.1. CAL 開発手法とは? .....	38
3.2. 目的・目標.....	38
3.3. 対象領域.....	40
3.4. 前提条件.....	41
3.5. 原則 .....	42
3.5.1. はじめに .....	42
3.5.2. Agile 開発手法との比較.....	43
3.5.3. 他の開発手法との比較 .....	44
3.6. 役割分担.....	45
3.6.1. はじめに .....	45
3.6.2. 一般的な役割分担 .....	46
3.6.3. XP 開発手法における役割分担 .....	47
3.6.4. CAL 開発手法における役割分担.....	48
3.7. 顧客 (CAL ドライバー) の学習 .....	49
3.7.1. 教える .....	49
3.7.2. 学ぶ.....	50
3.8. 顧客 (CAL ドライバー) と開発者 (CAL ナビゲーター) の実装範囲.....	52
3.9. 費用 .....	53
3.9.1. 費用名目 .....	53
3.10. その他 .....	53
3.10.1. 開発スケジュール.....	53
3.10.2. まとめ.....	53
<b>4. 効果の検証 1 (能動的学習)</b> .....	<b>54</b>
4.1. 実験の対象.....	54
4.2. モデルの拡張.....	54
4.3. 実験方法.....	54
4.3.1. 概要.....	54
4.3.2. スケジュール .....	55
4.3.3. 体制・役割分担 .....	55

4.3.4.	作業管理 .....	55
4.3.5.	作業量の測定・分析 .....	55
4.3.6.	評価 .....	56
4.4.	実験結果.....	56
4.4.1.	開発結果 .....	56
4.4.2.	開発作業量の推移 .....	59
4.5.	アンケート・ヒアリング .....	64
4.5.1.	アンケートの目的・概要 .....	64
4.5.2.	初回リリース後の役員（CALドライバー）からのアンケート結果 .....	65
4.5.3.	最終リリース後の役員（CALドライバー）からのアンケート結果 .....	65
4.5.4.	役員（CALドライバー）からのアンケート結果の分析 .....	66
4.5.5.	最終リリース後の利用者からのアンケート結果.....	66
4.5.6.	利用者からのアンケート結果の分析.....	67
4.5.7.	ヒアリング .....	68
<b>5.</b>	<b>効果の検証 2（企業経営や業務の戦略） .....</b>	<b>69</b>
5.1.	はじめに.....	69
5.2.	目的 .....	69
5.3.	実施 .....	70
5.4.	検証 .....	70
5.4.1.	一般的な業務システムの開発・保守・運用の経験と、経営戦略の策定に対する積極性との関連性 .....	70
5.4.2.	個々の業務システムの開発・保守・運用の経験と、経営戦略の策定に対する積極性との関連性 .....	72
5.4.3.	一般的な業務システムの開発・保守・運用の経験と、システムを知るためのアプローチとの関連性 .....	72
5.4.4.	一般的な業務システムの開発・保守・運用の経験と顧客能動学習型（CAL）開発 .....	74
5.5.	考察 .....	76
<b>6.</b>	<b>結論および今後の課題と展望 .....</b>	<b>77</b>
6.1.	結論 .....	77

6.2. 課題 .....	78
6.2.1. 見積り .....	78
6.2.2. 契約 .....	78
6.2.3. その他 .....	79
6.3. 展望 .....	80
<b>7. 謝辞 .....</b>	<b>81</b>
<b>8. 文献目録 .....</b>	<b>83</b>
<b>9. 付録 .....</b>	<b>88</b>
9.1. 図表目次.....	88
9.1.1. 図目次 .....	88
9.1.2. 表目次 .....	89
9.2. 現状分析.....	91
9.2.1. 運用現場内の問題点 .....	91
9.2.2. 運用現場外の問題点 .....	92
9.2.3. 学習パターン .....	92
9.2.4. 学習パターン（コンテキスト毎） .....	93
9.3. 効果の検証1（能動的学習） .....	94
9.3.1. プロダクトバックログ（一部抜粋・一部省略） .....	94
9.3.2. スプリントバックログ（一部抜粋・一部省略） .....	96
9.3.3. 集計結果 .....	96
9.3.4. 抽出結果 .....	98
9.4. 効果の検証2（企業経営や業務の戦略） .....	123
9.4.1. 属性情報 .....	123
9.4.2. 質問内容・回答結果 .....	129
9.4.3. 分析結果 .....	138



略語一覧

略語	名称（英語）	名称（日本語）
<b>ASD</b>	Adaptive Software Development	適応型ソフトウェア開発
<b>CAL</b>	Customers' Active Learning	顧客能動学習型
<b>CMS</b>	Contents Management System	コンテンツマネジメントシステム
<b>CoBRA</b>	Cost Estimation, Benchmarking, and Risk Assessment	コブラ見積法
<b>CoCoMo</b>	Constructive Cost Model	ココモ見積法
<b>DB</b>	Database	データベース
<b>DSDM</b>	Dynamic Systems Development Method	動的システム開発手法
	Dynamic Solutions Delivery Model	動的な解決策配達モデル
<b>Evo</b>	Evolutionary Project Management	進化型プロジェクトマネジメント
<b>FDD</b>	Feature Driven Development	ユーザー機能駆動開発
<b>FP</b>	Function Point	ファンクションポイント見積法
<b>ID</b>	Identification	識別、一意な番号
<b>IEH</b>	Ideal Engineering Hours	理想エンジニアリング時間
<b>IID</b>	Iterative & Incremental Development	反復型開発
<b>IT</b>	Information Technology	情報技術
<b>KKD</b>	Kan Keiken Dokyo	勘・経験・度胸
<b>NTL</b>	National Training Laboratory	全米トレーニング研究所
<b>RDBMS</b>	Relational Database Management System	関係データベース管理システム
<b>SaaS</b>	Software as a Service	サービスとしてのソフトウェア
<b>SIer</b>	System Integrator	システムインテグレーター
<b>SQL</b>	Structured Query Language	RDBMS においてデータの定義や操作を行うためのプログラミング言語
<b>UP</b>	Unified Process	統一プロセス
<b>VBA</b>	Visual Basic for Applications	バイビーエー
<b>XP</b>	Extreme Programming	エクストリームプログラミング

## 用語一覧

用語	意味
CAL ドライバー	CAL 開発手法での用語。実装を担当する顧客。
CAL ナビゲーター	CAL 開発手法での用語。CAL ドライバーをサポートする開発者。
イテレーション	リリース期間。ライフサイクルは1つ以上のイテレーションから成る。
スプリント	Scrum 開発手法での用語。イテレーションと同義語。
ドライバー	XP 開発手法での用語。実装を担当する開発者。
ナビゲーター	XP 開発手法での用語。ドライバーをサポートする開発者。
バックログ	製品をリリースするまでに実現しなければならない要件の一覧。残務。
プラクティス	習慣行為。実践。
儀式度	成果物とそこに至るまでのステップの規模や数量の程度。 <sup>[1]</sup>
重大度	システムによってもたらされる被害の可能性。 <sup>[2]</sup>

## 1. 序論

### 1.1. 現在の状況・問題点

昨今、ビジネスの進展スピードの高速化は著しく、ビジネス環境の変化が激しくなっている。<sup>[3]</sup> そのため、ビジネスに戦略的に活用されている情報システムに対しても、ビジネス環境の変化に迅速に対応することが一層求められるようになってきている。<sup>[3]</sup>つまり、様々な情報が氾濫しており、それらが簡単に手に入るようになった現代社会において、企業が事業を継続しながら経営を拡大して他社との競争で優位に立つためには、顧客のニーズを捉えて、経営戦略を策定・実行した後に評価するというサイクルをスピードアップしつつ延々と繰り返さなければならない。そのためには、自社の作業フローをシステム化して、大量の注文を素早く捌いて生産・製造・販売に繋げると同時に、それらのデータの整合性を確保したり、システムのセキュリティを高めたりすることは必須である。要は、システム開発のスピードアップが求められている。

しかしながら、システム開発・保守・運用の現場を見ると、顧客の要求を実現するのに時間がかかったり、リリースしたものの不具合が多く使い物になっていなかったりして、とてもスピードアップに応じられないような状況であるとともに、運用でカバーする範囲も広がっており、技術が進歩しているにも関わらず技術者一人一人の負担はあまり変わっていないように見える。

このような状況において、一人一人がそれぞれ得意とする分野だけを担当するようなお役所仕事の取り組みをしては駄目で、顧客も含めて全員が短期間でスキルアップして、かつ心理的な垣根を超えてチームを組み、相互にリスペクトしながら助け合っていかなければならないと考える。

こうした世の中の流れの中で、必然的にシステムを速く正確に開発するための開発手法に注目が集まっている。古くはウォーターフォール開発手法と呼ばれるものが主流であったが、システムの開発を開始してからリリースするまで、要求分析・設計・実装・検証などの全ての工程は、それぞれ1回しか経由しないため、開発中のユーザー要求の変更を受け入れられず、現在のビジネス環境の急激な変化に伴うユーザー要求の変化や増加に対応しきれなくなっている。

### 1.2. 一般的な解決アプローチ

以上のような問題に対する解決アプローチとして、10年以上前に米国で提唱されたアジャイル開発手法というものが、最近、注目を浴びつつある。システムインテグレーター（SIer：System Integrator）と呼ばれる企業の中には、大小を問わず、この手法を採用してシステムを開発しているところもある。ところが、実はアジャイル開発手法と呼ばれるものは、4つの価値<sup>[4]</sup>・12の原則<sup>[5]</sup>というものを謳い、その価値や原則というものを踏まえた数多くの開発手法を総称したものである。

このようにアジャイル開発手法に分類される個々の手法は、いずれもウォーターフォール開発手法とは異なり、ユーザー要求の変化や増加に対応できるように、優先順位が高い要求だけを先に開発するとか、要求漏れを防ぐために顧客を開発現場に常駐させて各工程でレビューしてもらうなどの工夫が施されている。

しかし、アジャイル開発手法と呼ばれる手法が万能かというとは決してそうではなく、様々な前提条件や制約事項がある。例えば、開発のスタート時に要求が決まっていないことがあるため、日本におけるシステム開発の契約では一般的である一括請負契約に適していない。<sup>[6]</sup>また、開発者が如何に素早くシステムを開発できるかに重点を置いているため、顧客は開発者に業務知識を提供してプロダクトの出来具合をチェックするだけであり、自ら手を動かして設計や実装をすることはなく、リリース後のシステムを運用する運用者の負担軽減までは考慮に入れていない。

### 1.3. 本研究で主張する解決アプローチ

以上のようなアジャイル開発手法の制約事項に対して、本研究では、顧客能動学習型（CAL: Customers' Active Learning）開発というものを提案する。具体的には、アジャイル開発手法に分類される開発手法であるエクストリームプログラミング（XP: Extreme Programming）のオンサイト顧客（Onsite Customer）<sup>[7]</sup>およびペアプログラミング<sup>[8]</sup>というプラクティスを発展させて、経営者や運用者をシステムの開発チームに加え、経営者にユーザー要求を管理してもらい、かつ開発者が彼らに開発の仕方をコーチングしながら共同で開発する、というものである。そして従来の開発手法と比較して、開発者および運用者の負担を減らすとともに、経営者が企業経営および業務をより戦略的に遂行するようになることを目指している。

勿論、開発経験が殆どないような経営者や運用者に開発に積極的に参加してもらうこと、および彼らに開発の仕方を教えることは開発者にとって負担になりかねない。しかし、その懸念は、最適な指導および学習方法を採用・応用することによって乗り越えられると考えられる。

### 1.4. 本論の構成

本論は、2章の現状分析から始まり、3章でモデルの具体的な内容を説明している。そして、4章でモデルを能動学習の視点で検証した実験の内容を記し、5章で今度は企業経営や業務の戦略の視点から検証したアンケートの内容を記している。

## 2. 現状分析

### 2.1. 経営戦略

#### 2.1.1. はじめに

昨今の企業の寿命は「日本企業で約7年、米国企業で約5年」<sup>[9]</sup>とされている。そのような厳しい状況の中で、企業が事業を継続しようとする、経営および事業に関する目標を掲げることは必須であると思われる。そして、その目標を達成するためにはどうすれば良いのかをまとめた経営戦略も必要になってくると考えられる。しかし、その前に、そもそも経営戦略とは何なのか、それはどういうものなのか、それを策定するためにはどうすれば良いのか、といったことを知らなければならない。まずは、その辺から分析する。

#### 2.1.2. 定義

経営戦略という用語は、以下のように様々な定義がある。

- ・ 企業が環境変化に対応し、生存・成長するための鍵概念。<sup>[10]</sup>
- ・ 持続的競争優位を達成するためのポジショニングを構築すること。<sup>[11]</sup>
- ・ 企業が考えた競争に成功するためのセオリー。<sup>[12]</sup>
- ・ 企業を取り巻く環境との関わりについて、企業を成功に導くために何をどのように行うかを示したもので、企業に関与する人たちの指針となり得るもの。<sup>[13]</sup>
- ・ 「企業や事業の将来のあるべき姿とそこに至るまでの変革のシナリオ」を描いた設計図。<sup>[14]</sup>
- ・ 市場のなかの組織としての活動の長期的な基本設計図。<sup>[15]</sup>
- ・ 企業がその目標を達成するための資源展開のあり方を環境とのかかわりの中で方向づけるもの。<sup>[16]</sup>

このように経営戦略とは曖昧な概念であることが分かるが、本研究では、「経営または事業の目標を達成するための戦略」としておくに留める。

#### 2.1.3. 分類

経営戦略は、大きく分けて以下の3つに分類される。<sup>[10]</sup>

- ・ “Corporate Strategy”（全社戦略）…企業全体に関わる戦略。
- ・ “Business Strategy”（事業戦略）／ “Competitive Strategy”（競争戦略）…事業ごとの戦略。

- “Functional Strategy”（機能戦略）…各機能分野別の戦略。

上記の戦略は、組織の体制に合わせて階層化しており、図で表すと以下のようなになる。

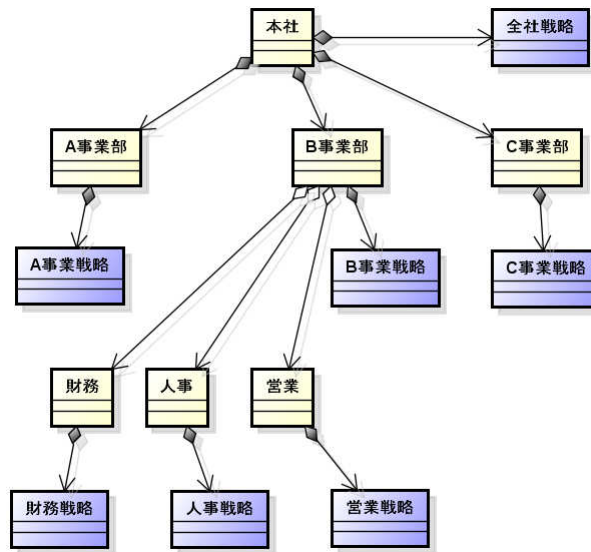


Figure 1 “Strategies Hierarchy” (Based on [10])

本研究では中小企業を対象としている。中小企業は事業部の数も機能分野（業務）の数も少なく、それぞれの戦略の規模も大きくないと思われる。よって本研究では、特に明記しない限り、戦略の種類を区別・意識しないこととする。

#### 2.1.4. プロセス

経営戦略のプロセスは、以下の3つに分かれる。<sup>[17]</sup>

- 戦略の策定
  - 自社や競合相手の内的要因（能力や資源など）、自社や競合相手を取り巻く外的要因を分析・評価する。
  - 上の評価に基づいて、目標を設定する。目標は、レベルに応じて長期的なものから短期的なものまでありえる。
  - それらの目標を達成するために必要な計画や手段を定める、戦略を策定する。
- 戦略の実行。
  - 戦略の実行に必要な資源を配分する。
  - 命令系統（階層構造）もしくはそれに代わる何らかの構造（例えばプロジェクト・チーム）を確立する。

- ・ 戦略の評価。
  - バランス・スコアカードや SWOT 分析などを通じて、戦略の有効性（合理性、実行可能性、正当性など）を評価する。
  - 戦略が有効でないか、状況の変化などが生じた場合には、必要に応じて戦略を変更・再策定する。

上記のような分類を見ると、真っ先に思い浮かぶのは、Plan（計画）・Do（実行）・Check（評価）・Act（改善）の4つプロセスで構成される PDCA サイクルである。PDCA サイクルと比較すると、Act が抜けていることに気づく。しかし、次項 2.1.5 でも述べるように、本研究では、戦略の策定に至るまでのプロセスに注目するので、そのことには拘らない。

### 2.1.5. 問題点

企業が事業を継続していくためには、中長期的な視野に立って策定される経営戦略は、無いより有った方が良いと思われる。しかしながら、経営戦略の策定の仕方を知らなくても企業経営者にはなれる訳であり、そのせいか、「たいていの経営者は、近視眼的である。」とも言われる。<sup>[18]</sup>

そのような経営者の特徴として、以下のような傾向があると考えられる。

- ・ 何となく不安を感じつつも、いつも勘・経験・度胸（KKD）に頼ってしまう。
- ・ 事前に入念に調査・分析・見積りすることを惜しむ。
- ・ 効果が見込めるかどうか分からないものに投資してしまう。
- ・ 結果・利益さえ出ていれば良しとしてしまい、継続的な検証・改善を行わない。

このような KKD に頼った経営を、戦略的な経営に変えていくためにはどうすれば良いのか？経営戦略の最初のステップである戦略の策定を行うためにはどうすれば良いのか？

このように立ち止まって考えてみると、まずは前項 2.1.4 でも述べたように、戦略策定のインプットとなる、自社や競合相手の内的要因や外的要因を把握しなければならない。特に現代社会では「己を知る」つまり自社の内的・外的要因（例：自社の業務システム）を知ることが「敵を知る」ことよりも重要である。自社の内的要因の最たるものとして、自社の業務を遂行する情報システムを挙げることができる。そこで次の 2.2 節では、この情報システムについて現状を分析する。

## 2.2. 情報システム（開発・保守・運用）

### 2.2.1. はじめに

経営者が自社の情報システムを把握しようとしても、それは一筋縄では行かない。何故なら、大抵の企業は自社の情報システムの開発を **SIer** と呼ばれる業者に発注しており、どのような仕組みになっているのか分からないからである。

特に大企業のように業務の種類も規模も多大であればある程、当然、システムは複雑になるので、経営者がシステムの仕組みを知ることは難しい。けれども大企業の場合、経営者以外でシステムの仕組みに詳しい人材が豊富にいることは十分に考えられ、そのような人材が戦略の策定に少なからず貢献していると想像できる。

一方、中小企業の場合、システムの開発を **SIer** に外注しようにも、大企業に比べて資金が乏しいので、その分、リスクは高くなる。例えば、システムの開発を外注して、完成・リリースした後、利用・運用することになって初めて、様々な不具合が見つかるとか、使い難いことが判明することはよくある。しかし、そういった問題を解決するための保守費を追加で支払う余裕はもうないために、システムを改善できないというリスクは、大企業よりも中小企業の方が高い。そもそも開発を外注に任せていたら、いつまで経ってもシステムの仕組みを詳しく知ることはできない。

このように考えると、とても悩ましい問題であることが分かってくる。話が繰り返しになるが、以下のようなジレンマが中小企業には付いて回る。

- (1) まずシステムの開発を **SIer** に発注しようにも資金が乏しい。
- (2) たとえ開発してもらったとしても、利用・運用しづらい恐れがある。
- (3) そういう問題が発生しても保守費を追加で支払えない恐れもある。
- (4) そして外注にシステムの開発・保守を委託している限り、システムの仕組みを知ることができない。
- (5) その結果、経営をより戦略的にすることに繋がっていかない。

上記の問題の内、(1)と(3)は中小企業としての制約事項であるので、これ以上の分析は控える。(5)は、(4)の問題が解決すれば自動的に解消するはずである。よって、(2)と(4)の問題を解決するにはどうすれば良いのか、次項以降で掘り下げて考える。

### 2.2.2. ライフサイクル

情報システムに限らずシステム全般について把握しようとした時、一つの物差しになる



のがライフサイクルである。システムのライフサイクルは、構想（Concept of Operations）・要求開発（Requirements Development）・設計（Design）・実装（Implementation）・検証（Verification）・妥当性確認（Validation）・リリース（Installation & Deployment）・運用（Operation）・保守（Maintenance）・廃棄（Disposal）などのフェーズで構成されている。情報システムのライフサイクルは、一般的に、開発（構想・要求開発・設計・実装・検証・リリース）・運用・保守というフェーズから成り立つ。<sup>[19][20]</sup>図で表すと以下のようなになる。

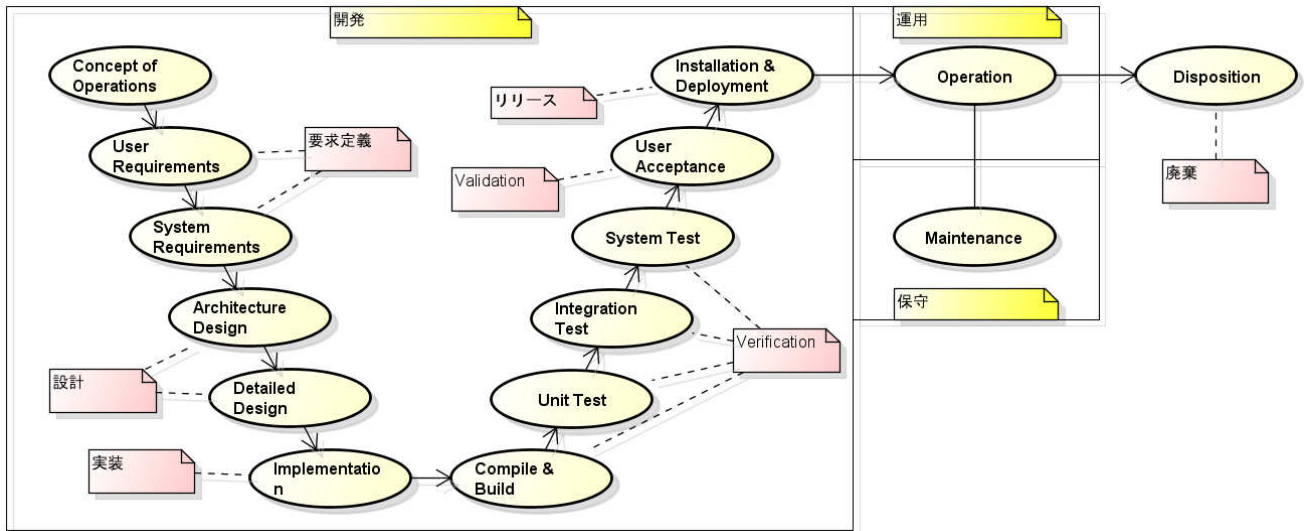


Figure 2 “Information System Lifecycle” (Based on [19][20])

上掲の Figure 2 のように工程面からシステムの開発・保守・運用を眺めてみると、開発がメインで、保守・運用がおまけのように見える。しかし、費用面で見ると様相が違ってくる。詳しくは次項以降で述べる。

### 2.2.3. IT 費用の推移と割合

システムの開発と保守・運用を費用面で眺めると何が見えてくるのか。まずは、一企業あたりの IT 費用に占める開発および保守・運用の費用の割合を、金額および百分率で表した二種類のグラフで示す。

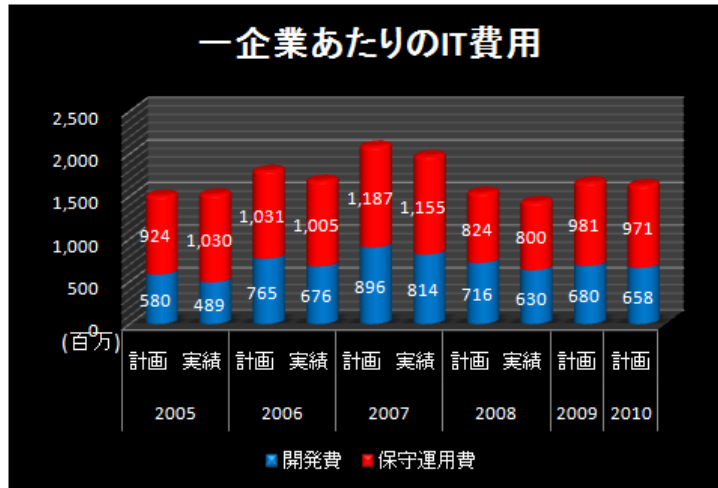


Figure 3 “IT cost amount per 1 company” (Based on [21][22][23][24][25][26])

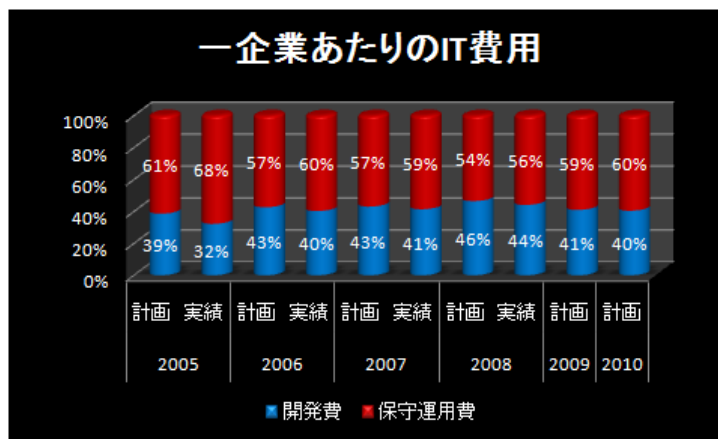


Figure 4 “IT cost percentage per 1 company” (Based on [21][22][23][24][25][26])

Figure 3 および Figure 4 のグラフが表していることは以下の通りである。

- ・ 開発費は、計画値だけを見た場合、もしくは実績値だけを見た場合、毎年増えている。逆に言えば、保守・運用費は毎年減っている。ここから、企業は開発に力を注ぎたいことが読み取れる。
- ・ しかしながら、2006 年以降、全体の IT 費用は、計画値よりも実績値の方が低い。つまり、毎年、IT 費用の予算は削られている。しかも、保守・運用費の IT 費用に占める割合は、計画値よりも実績値の方が増えている。これは、企業としては開発に力を注ぎたいけれども、保守・運用費は開発費と比べて削りにくいことを表している。

## 2.2.4. 保守・運用費の推移と構成

次に、そのような保守・運用費の内訳を年度毎の比率で表したものを、以下の表で示す。

Table 1 “Transition and composition of maintenance & operation costs” (Based on [27][28][29])

年度	計画 /実績	有効 回答	保守・運用費						
			1	2	3	4	5	6	7
2005	実績	680	26%	9%	13%		7%	30%	13%
2006	実績	408	21%	21%	10%		7%	28%	12%
2007	計画	408	21%	21%	9%		7%	29%	13%
2008	N/A	N/A							
2009	計画	489	23%	15%	17%	3%	8%	27%	7%

上記の Table 1 のヘッダに記載されている数字の意味は、以下の Table 2 に記した。

Table 2 “Breakdown of maintenance and operation costs” (Based on [30])

保守・運用費		
No	科目名	内容
1	ハードウェア費	ハードウェア機器(周辺機器を含む)購入、レンタル・リース料、保守費、償却費
2	ソフトウェア費	ソフトウェア購入費、レンタル料、償却費
3	ソフトウェア保守費	ソフトウェアの保守費用
4	処理サービス費	SaaS 等のサービス使用料
5	通信回線費	通信回線使用料、ネットワーク加入・使用料、携帯電話加入・使用料
6	外部委託費	保守・運用、コンサルティング等のアウトソーシング費用
7	その他	上記以外(含む社員人件費、運転管理費)

以上の Table 1 および Table 2 を見れば分かる通り、保守・運用費の中で外部委託費の占める割合が最も大きい。仮に、自社とアウトソース先の一人当たりの人件費が同じで、アウトソース先の企業のマージンが 1/3 であるとする、もし外部委託していた業務を自社で行うようにした場合、Figure 3 に記した数値を元に計算すると、最大で 9%~10% (=27% × 1/3~30% × 1/3) の費用を削減することができる。その上、実は運用費を半分にすることもできるという意見もある。<sup>[31]</sup>

こういった事情を考慮すると、いずれ運用費を削減する方向に世の中は流れていくと予想できる。しかし、運用現場では付録の Table 26 および Table 27 にあるような問題を抱えており、削減は容易でない。そもそも一般的に、システムの規模が大きくなればなる程、運用でカバーする範囲は広がるため、運用者の負担もそれだけ増すことになる。では、システムの規模そのものを小さくすることはできないのか？次項では、そのような観点で分析してみる。

### 2.2.5. 典型的な情報システムにおける機能の利用割合

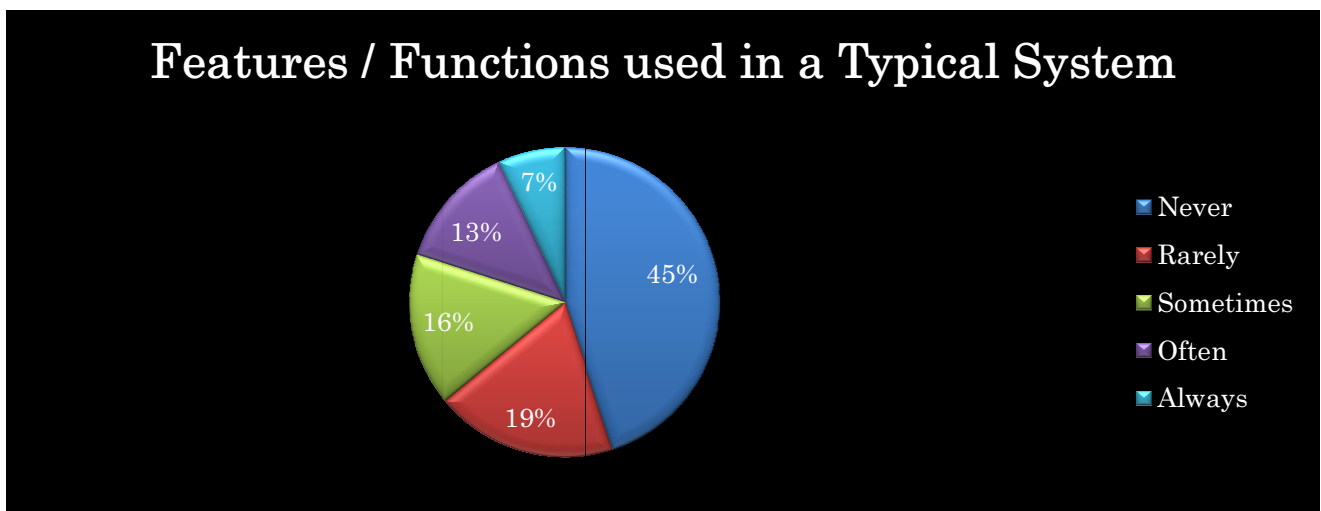


Figure 5 “Features / Functions used in a typical system” (Based on [32])

上掲の Figure 5 は、2002 年度のデータであるが、典型的な情報システムが備える機能の内、一度も使用していないものが 50% 近くを占めている。たとえ一度も使われない無駄な機能であっても、システムを保守する度に影響が及ばないかどうかを調査しなければならないため、開発費のみならず保守・運用費を増加させる一因となりうる。開発者または開発チームが業務に精通していればいる程、不要な機能を実装する確率は減るだろうし、そのためには利用者または運用者の関与も必要になってくる。

### 2.2.6. プロジェクトの成功要因

Table 3 “Success factors of projects at 1995” (Based on [33])

Rank	Success Criteria	Points
------	------------------	--------

1	User Involvement	19
2	Executive Management Support	16
3	Clear Statement of Requirements	15
4	Proper Planning	11
5	Realistic Expectations	10
6	Smaller Project Milestones	9
7	Competent Staff	8
8	Ownership	6
9	Clear Vision & Objectives	3
10	Hard-Working, Focused Staff	3
	TOTAL	100

Table 4 “Success factors of projects at 2001” (Based on [34])

Rank	Success Criteria	Points
1	Executive Support	18
2	User Involvement	16
3	Experienced Project Manager	14
4	Clear Business Objectives	12
5	Minimized Scope	10
6	Standard Software Infrastructure	8
7	Firm Basic Requirements	6
8	Formal Methodology	6
9	Reliable Estimates	5
10	Other	5
	TOTAL	100

上掲の Table 3 および Table 4 は、現実のプロジェクト（失敗も含む）が終了した後に行われたアンケートの結果である。プロジェクトの成功要因の 1 位・2 位は「ユーザーの参加」と「（システムのオーナーとなる）経営陣のサポート」で占められている。ここで言う「ユーザー」とは、システムの利用者や運用者を指しているため、プロジェクトを成功さ

せるためには経営者や運用者を開発に関与させた方が効果的であることが分かる。

次の 2.3 節では、情報システム開発のプロジェクトを遂行するのに欠かせない開発手法について触れて、ユーザーをどのように関与させているのかを探る。

## 2.3. 情報システム（開発手法）

### 2.3.1. システム開発手法の一覧

情報システムを開発する際、モデルとなる主な開発手法として、主に以下のようなものを挙げることができる。<sup>[35][36][37]</sup>

- ・ ウォーターフォール開発手法 (Waterfall Development)
- ・ 反復型開発手法 (IID: Iterative & Incremental Development)
- ・ アジャイル開発手法 (Agile Development)

分かりやすく説明するために、今後はウォーターフォール開発手法を Waterfall 開発手法、反復型開発手法を IID、アジャイル開発手法を Agile 開発手法と略す。

まず、Waterfall 開発手法は、予測型 (predictive) の開発手法と言われ、開発期間中の変化を許容しない。それに対して、IID や Agile 開発手法は、適応型 (adaptive) の開発手法とみなされ、開発期間中の変化をできる限り許容する特徴を持つ。詳しい説明は次項以降で述べるが、それぞれの開発手法の比較を以下に示す。

Table 5 “Comparison of 4 Software Development Methodologies in target areas” (Based on [38][39][40])

開発の対象領域や種類	Waterfall	IID	Agile
ビジネス要求が変化する領域	×	✓	✓
リスクが高い領域	△	△	✓
市場競争が活発な領域	×	△	✓
大規模開発	△	✓	△
分散拠点開発 <sup>1</sup>	✓	△	△
組織 <sup>2</sup> 間をまたぐ開発チームによる開発	✓	△	△

<sup>1</sup> オフショア開発を含む。

<sup>2</sup> 会社を含む。

(凡例)

- ・ ✓ : 適している。
- ・ × : 適していない。
- ・ △ : どちらとも言えない。

上掲の表は、幾つかの文献を参考にして作ったが、分析・考察の余地はある。しかしながら、傾向は掴めると思う。つまり、要求が不明確であるとか変更が激しい場合、Agile 開発手法が適しているけれども、開発チームが複数に分散していて、お互いに歩調を合わせながら開発しなければならないような場合、Waterfall 開発手法の方が適している。

中小企業の規模を考慮すると、業務システムを開発する時、上掲の表にある「大規模開発」や「分散拠点開発」、「組織（会社）間をまたぐ開発チームによる開発」というものは想定しにくいので、次項以降では、それらを考慮に入れない領域で、それぞれのシステム開発手法を比較・分析する。

### 2.3.2. ウォーターフォール開発手法と反復型開発手法

以下の Figure 6 のグラフは、Waterfall 開発手法および IID というそれぞれの開発手法に基づき情報システムを開発した時の、システムを構成するプログラムのコード量の推移をプロジェクトの開始時点から比較したものである。

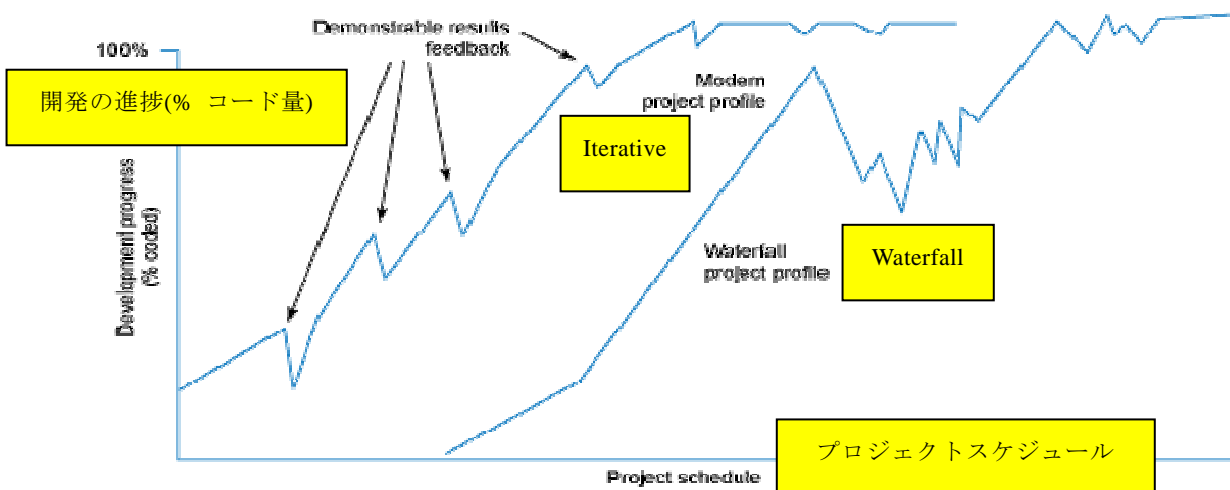


Figure 6 “Based on frequent demonstrations and stakeholder feedback, the small course corrections early in the lifecycle of an iterative (“modern”) project lead to more rapid success, compared to a “waterfall” project.” [41]

予測型の開発手法である **Waterfall** 開発手法は、ある工程が終わらないと次の工程に移れないため、実装（コーディング）の開始がどうしても遅れてしまう。

一方、適応型の開発手法である **IID** は、プロジェクトの初期からユーザーに対してデモンストレーションを行いユーザーからフィードバックを得ようとし、そのために試作品も実装したりするので、コード量が 100%に到達するのも速い。

また、ユーザーが関わるのは、通常、開発イテレーションの最初と最後である。ライフサイクルとイテレーションの関係は以下の通りである。

$$\text{ライフサイクル} = \text{イテレーション} \times (1 \text{ 以上})$$

イテレーションの目標はシステムの部分的なリリースであり、それらのリリースは、ライフサイクルの目標であるシステムの全体的なリリースを構成する。

**Waterfall** 開発手法の場合、1 ライフサイクル=1 イテレーションであるが、**IID** の場合、1 ライフサイクル=1 つ以上のイテレーションとなる。当然、**IID** の方がユーザーの関与が多くなる。次の項では、その **IID** について述べる。

### 2.3.3. 反復型開発手法

**IID** の例は、**Rational Unified Process (RUP)**<sup>[42]</sup>である。RUP が示すシステム開発のライフサイクルは、以下の Figure 7 のグラフで示すとおり、4 つの段階（phase）と 9 つのディシプリン（discipline）<sup>3</sup>から成る。

---

<sup>3</sup> A discipline in the RUP can be described as a grouping of interrelated activities and the artifacts or deliverables they produce. <sup>[37]</sup>



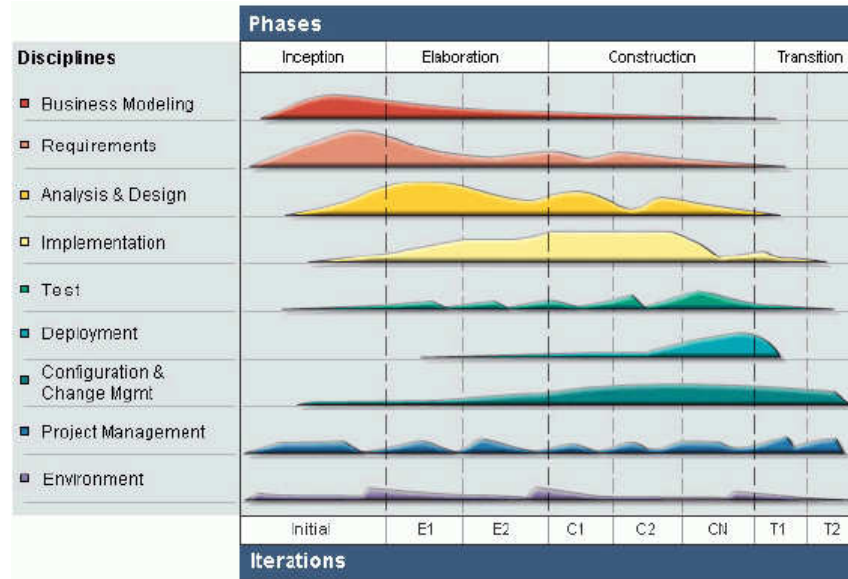


Figure 7 “The Rational Unified Process” [43]

各段階は、1 つ以上のイテレーションから構成される。そして、イテレーション毎に以下のディシプリンを上から順に辿って、そのイテレーションを完了させる。

- Business Modeling
- Requirements
- Analysis & Design
- Implementation
- Test
- Deployment

各イテレーションは直列で繋がっているため、前のイテレーションが完了していない限り、次のイテレーションには移れない。図で表すと以下の Figure 8 のとおりになる。

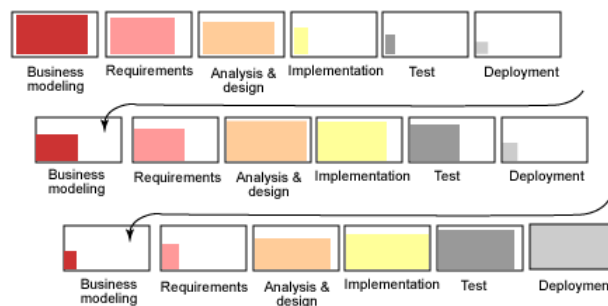


Figure 8 “The RUP's iterative development process” [44]

ユーザーの要求は、最初の“Business Modeling”および“Requirements”で取り入れられる。

もし、イテレーションの途中で要求漏れが見つかった場合、そのイテレーションでの要求の変更は原則的に受け入れられず、次のイテレーションで追加 (increment) される。よって、Waterfall に比べるとユーザーの関与は増えており、要求がより短期間で実現される可能性は増えているものの、例えば要求が不明確過ぎるとかビジネスを取り巻く環境の変化が激しい場合、要求を実現しきれないリスクは残っている。こういった IID の問題点を解消する一つの方法として、次項で Agile 開発手法について言及する。

### 2.3.4. アジャイル開発手法

#### (1) 概要

まず、Agile 開発手法とは、「不確実なビジネス環境の中で変化するニーズへの迅速な対応を目的としたソフトウェア開発手法」<sup>[45]</sup>である。ただし、Agile 開発手法が全ての領域で適用可能かと言うと、そうではないことは既に Table 5 で述べた。それを踏まえて、Agile 開発手法とその他のシステム開発手法を見比べると、まず以下の 4 つの価値<sup>[4]</sup>というものを宣言している点が特徴的である。

- ・ プロセスやツールよりも、個人との対話
- ・ 包括的なドキュメントよりも、動くソフトウェア
- ・ 契約交渉よりも、顧客との協調
- ・ 計画に従うよりも、変化へ対応する

これだけを見ても、他のシステム開発手法と比べて、儀式的なもの（例：ドキュメント作成）を省き、その分、より速くシステムを提供することを心掛けていることが伝わってくる。

更に Agile 開発手法は、上記の 4 つの価値の背後にある下記の 12 の原則<sup>[5]</sup>に従わなければならないとされている。<sup>[46]</sup>

Table 6 “12 principles behind the Agile Manifesto” (Based on [5])

#	Principle	Keyword
1	顧客満足を最優先し、価値のあるソフトウェアを早く継続的に提供します。	Early and continuous delivery
2	要求の変更はたとえ開発の後期であっても歓迎します。変化を味方につけることによって、お客様の競争力を引き上げます。	Adaptive planning

3	動くソフトウェアを、2-3 週間から 2-3 ヶ月というできるだけ短い時間間隔でリリースします。	Time boxing
4	ビジネス側の人と開発者は、プロジェクトを通して日々一緒に働かなければなりません。	On-site customer
5	意欲に満ちた人々を集めてプロジェクトを構成します。環境と支援を与え仕事が無事終わるまで彼らを信頼します。	Team environment
6	情報を伝えるもっとも効率的で効果的な方法はフェイス・トゥ・フェイスで話をする事です。	Face-to-face interaction
7	動くソフトウェアこそが進捗の最も重要な尺度です。	Working software
8	アジャイル・プロセスは持続可能な開発を促進します。一定のペースを継続的に維持できるようにしなければなりません。	Sustainable development
9	技術的卓越性と優れた設計に対する不断の注意が機敏さを高めます。	Technical excellence and good design
10	シンプルさ（ムダなく作れる量を最大限にすること）が本質です。	Simplicity
11	最良のアーキテクチャ・要求・設計は、自己組織的なチームから生み出されます。	Self-organizing team
12	チームがもっと効率を高めることができるかを定期的に振り返り、それに基づいて自分たちのやり方を最適に調整します。	Kaizen

上記の表の真ん中の列 **Principle** という列には元の資料<sup>[5]</sup>から転載した文章を並べた。この資料は後で比較するために再利用するが、その際、文章をそのまま使い回すのはくどいので、その右隣に **Abbreviation** という列を設けて、再利用するための略語を記載した。

## (2) 一覧

Agile 開発手法に分類されるものは数多くある。主なものは以下の通りである。<sup>[1][2]</sup>

- Adaptive Software Development (ASD)
- Agile Unified Process (AUP)
- Crystal family of methodologies (Crystal)
- Dynamic Systems Development Method (DSDM)
- Evolutionary Project Management (Evo)

- Extreme Programming (XP)
- Feature Driven Development (FDD)
- Lean Software Development
- Scrum

### (3) 前提条件

(2)で掲げた Agile 開発手法に分類される各開発手法は、開発対象のシステムやその業務・組織、および開発チームの特性に応じて、使い分けをする必要がある。このことは以下の表を見れば分かる。

Table 7 “Comparison of Agile Software Development Methodologies”<sup>[2]</sup>

条件	XP	Scrum	Lean	FDD	AUP	Crystal	DSDM
少人数のチーム	✓	✓	✓	×	×	△	✓
要件の変化が激しい	✓	✓	✓	✓	△	△	×
分散チーム	×	✓	✓	✓	✓	×	×
儀式度が高い	×	×	△	△	✓	△	✓
重大度が高いシステム	×	△	△	△	△	✓	×
複数の顧客／利害関係者がいる	×	✓	✓	△	△	△	×

(凡例)

- ✓：適している。
- ×：適していない。
- △：どちらとも言えない。

上掲の Table 7 は、一番左の欄に書かれた条件に適している手法とそうでないものを表している。XP を例にとると、開発チームが「少人数のチーム」である時や、ユーザーやシステムの「要件の変化が激しい」時には適しているが、開発者が別々の現場にいるような「分散チーム」には適していない。

ただし実際の問題として、上記の手法を適用するのに適しているとされる条件をピッチリ満たすような開発プロジェクトが常にあるとは限らないが、上記の Agile 開発手法と呼ばれる各手法は、お互いを組み合わせて開発プロジェクトに適用することが可能である。

#### (4) 詳細

以下、主な手法について簡単な説明を記しておく。

##### ① Crystal

Crystal には、「コバーン<sup>4</sup>の尺度」と言われる以下の Figure 9 の表がある。

	Clear	Yellow	Orange	Red	Maroon
Life (L)	L6	L20	L40	L80	L200
Essential Money (E)	E6	E20	E40	E80	E200
Discretionary Money (D)	D6	D20	D40	D80	D200
Comfort (C)	C6	C20	C40	C80	C200
	1-6	7-20	21-40	41-80	81-200

Figure 9 “Cockburn Scale”<sup>[2]</sup>

表の見方は以下の通りである。

- 横軸の数字：プロジェクトの規模。プロジェクトに関わる人数を表す。
- 縦軸：プロジェクトの重大度。システムが想定外の振る舞いをした場合、最悪、何が失われるのかを表す。“Comfort”(快適さ)、“Discretionary Money”(裁量の範囲内の資金)、“Essential Money”(組織の運営に欠かせない資金)、“Life”(生命)から成る。
- ヘッダ値：プロジェクトの色。プロジェクトの規模に応じて色分けする。“Clear”、“Yellow”、“Orange”、“Red”、“Maroon”から成る。
- セル値：プロジェクトのカテゴリ。アルファベットは縦軸（プロジェクトの重大度）の頭文字、数字は横軸（プロジェクトの規模）である。例えば、“E6”は、1～6人の規模で、システムが想定外の振る舞いをした場合、最悪、組織の運営に欠かせない資金が失われるプロジェクトを指す。

上掲の表は、以下の Figure 10 の表のように Crystal を含む各種 Agile 開発手法の分類にも利用される。

<sup>4</sup> Crystal の提唱者 Alistair Cockburn の名字。

Life (L)	L6	L20	L40	L80	L200	Scrum Crystal Lean
Essential Money (E)	E6	XP E20	DSDM E40	E80	AUP E200	
Discretionary Money (D)	D6	D20	D40	D80	D200	
Comfort (C)	C6	C20	C40	C80 FDD	C200	
	1-6	-20	-40	-100	-200	Number of People

Figure 10 “Table which shows evaluations based on experiences and observations of authors to the conformity of each development methodology with Cockburn's scale” [2]

この表の見方は以下の通りである。

- **XP** : 20 人迄の規模のプロジェクトに適用できる。しかし、システムが予期せぬ振る舞いをすると、最悪、生命（Life）が失われるようなシステムの開発プロジェクトには適用できないことになる。
- **AUP** : 200 人規模のプロジェクトにも適用できる。しかし、システムが予期せぬ振る舞いをしても、最悪、資金（Discretionary / Essential Money）だけが失われるようなシステムの開発プロジェクトに限られる。

本研究でも、この表を、開発手法を選択する時の判断材料にした。

## ② AUP

AUP は、前の 2.3.3 項で説明した RUP を Agile 開発手法にアレンジしたものである。AUP が示すシステム開発のライフサイクルは RUP と似ており、以下の図のとおり、4 つの段階（phase）と 7 つのディシプリン（discipline）から成る。つまり、RUP から“Requirements”と“Analysis & Design”の 2 ディシプリンを除いた、より実装に特化したものが AUP である。

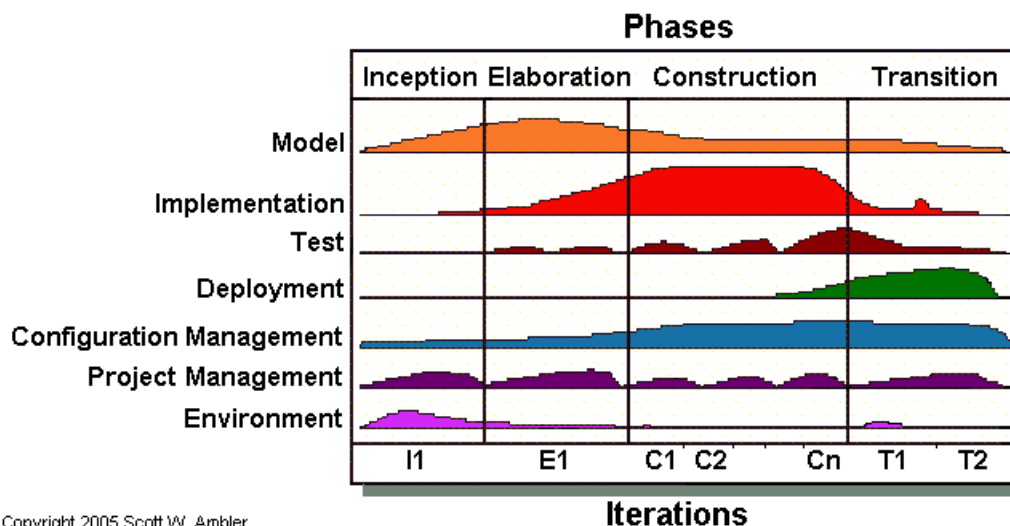


Figure 11 “The Agile Unified Process (AUP) lifecycle” [47]

上掲の Figure 11 のグラフの横軸（フェーズ）の目標（goal）は以下の通りである。 [48]

- Inception: プロジェクトの初期スコープやシステムの潜在的なアーキテクチャを特定することと、プロジェクトの初期資金や利害関係者の同意を得ること。
- Elaboration: システムのアーキテクチャを明確にする（prove）こと。
- Construction: プロジェクトの利害関係者の最優先のニーズを満たせるよう、定期的（regular）かつ漸進的（incremental）に、動くソフトウェアを構築すること。
- Transition: システムを検証して、本番環境にリリース（deploy）すること。

これに対して、縦軸（ディシプリン）の目標（goal）は以下の通りである。 [49]

- Model: 組織の業務やプロジェクトによって取り組まれようとしている問題領域を理解すること、そしてその問題領域に取り組むための実行可能な解決策を特定すること。
- Implementation: モデルを（プログラミング言語で書かれた）実行可能なソースコード（executable code）に変換すること、そしてテスト、特に単体テストの基本的なレベルを実行すること。
- Test: 品質を確保するために客観的な評価を下すこと。これは、欠陥を見つけ、システムが設計通りに動いているのかを検証し、要求が満たされるかを検証することを含む。
- Deployment: システムのリリース（deploy）を計画すること、そしてシステムがエンドユーザーに利用可能になるよう計画すること。
- Configuration Management: プロジェクトの成果物（artifacts）へのアクセスを管理すること。これは、成果物のバージョンを長期間に渡って追跡するだけでなく、その成果

物に対する変更を制御し管理することを含む。

- **Project Management:** プロジェクト上で起こる活動を監督すること。これは、システムが期間内 (on time) かつ予算内 (within budget) でリリースされるよう、リスクを管理したり、人々を監督したり (作業を割り振る、進捗を追跡する、等)、人々やプロジェクトの範囲外にあるシステムと連携することを含む。
- **Environment:** 適切な手順書やガイダンス (規格やガイドライン)、ツール (ハードウェア、ソフトウェア、等) がチームに必要なだけ利用できるよう保証することによって、残りの努力を支援すること。

本研究では、この開発手法を全面的には利用していないが、時間をフェーズで区切り、作業をディシプリンで分類するという考え方は参考になった。

### ③ Scrum

Scrum では、イテレーションをスプリントと呼んでいる。原則として、スプリントは 30 日間であるため、30 日毎にシステムの部分リリースを行わうことになる。プロジェクトの規模は特に決まっておらず、他の Agile 開発手法と組み合わせることができる。

各ステークホルダーとその役割は以下の Table 8 のとおりである。

Table 8 “Roles of stakeholders in Scrum” (Based on [50])

ステークホルダー	役割
プロダクトオーナー	プロダクトバックログを作成する。
スクラムマスター	スクラムチームをゴールへ導く。
スクラムチーム	スプリントバックログに書かれた要求を実現するために開発する。

プロダクトバックログとスプリントバックログの意味は以下の Table 9 のとおりである。

Table 9 “About backlogs in Scrum” (Based on [50])

バックログ	意味
プロダクトバックログ	優先順位付きの要求リスト。
スプリントバックログ	プロダクトバックログの上から順に抽出した、そのスプリントで実現する要求のリスト。



## ④ XP

XPでは、イテレーションの期間は概ね1~3週間である。作業成果物として、詳細な要求ドキュメントは必要ではない。その代わりに、現場に常駐する業務に精通した顧客（オンサイト顧客<sup>[7]</sup>）が、開発者に業務に関する詳細な情報を提供することが求められる。

「オンサイト顧客」は、Agile 開発手法の4つの価値の一つ「契約交渉よりも顧客との協調（Customer collaboration over contract negotiation）」<sup>[51]</sup>を具現化したものであると考えられ、システムを利用・運用する側の人間が開発現場に常駐して、ユーザー要求を開発者に伝え、ユーザー要求がシステムの機能として実現することを期待されている。

XPを適用した情報システム開発の成功事例<sup>[52]</sup>もあるが、「オンサイト顧客」を務める人間の間人性が開発現場に悪影響を及ぼし、開発者に心理的な負担を強いることになるかもしれないというリスクを排除することはできない。また、そうでなくても、「開発チーム内に顧客が居るということは、見せたくない部分も見せることにもなる。その顧客に決断してもらった場合もあるから、決定権のある人物でなければならない。期間や開発チームの居住環境などに対して、制約条件が多くなるのではないか。」<sup>[53]</sup>といった課題・リスクもある。よって「実際の現場では適用しづらい」<sup>[54]</sup>という方が現実的であると考えられる。

## 2.4. 情報システム（まとめ・問題点）

### 2.4.1. まとめ

以上、述べてきた情報システムの現状を箇条書きにしてまとめると、以下のようになる。

- ・ 開発費と保守・運用費の割合は約4:6である。すなわち保守・運用費は、企業のIT関連の予算の6割以上を占め、開発費を圧迫している。<sup>[55][22][23][24][25][26]</sup>
- ・ 保守・運用費の内、約3割が運用のアウトソーシングを含む外注委託費である。
- ・ システムが備える機能の約45%は全く利用されていない。しかし保守費はかかる。
- ・ システム開発のプロジェクトが成功するためには「ユーザーの参加」が欠かせない。
- ・ 予測型よりも適応型の開発手法の方が、開発のライフサイクルは短い。
- ・ Agile 開発手法は「ユーザーの参加」が必須になる適応型の開発手法である。
- ・ システムがリリースされて新しい業務が増える度に、運用者は新しい業務を学んでシステムの運用の仕方を覚えなければならない。
- ・ システムに不備があればある程、運用に皺寄せが及ぶ。運用者がカバーする領域は増える一方である。

- ・ 企業は、保守・運用費を削って開発費を増やしたいと考えているが、既に運用者は業務に追われて疲弊している。
- ・ また、システムが提供する機能が使いづらかったり不備があったりしてストレスを感じても、運用者はそれを使用して運用し続けなければならない。

#### 2.4.2. 問題点

前項の一覧を眺めると、「利用者・運用者にとって使いやすい機能がシステムに備わっていないばかりか、無駄な機能が付いている」ということが問題の共通要因として見えてくる。このような問題を解決する方法の一つとして、最適なシステム開発手法を選んで、ユーザー要求をきちんと実装したシステムを開発する、と考えることができる。

情報システムの開発手法は、時代の流れと共に発展していき、最近では Agile 開発手法が主流になりつつあり、大企業も注目・活用し始めている。現状分析の結果、条件付きではあるが、情報システムの開発費を抑えるためには、Waterfall 開発手法よりも IID、IID よりも Agile 開発手法の方が有効であることが分かった。

けれども、Agile 開発手法は、如何に速くシステムを開発するかという観点を重視しており、運用を如何にやりやすくするかという観点を持ち合わせていない。よって、Agile 開発手法を含む既存の開発手法では、必ずしも運用者の負担を減らすことができるとは言えない。

例えば、“How We Made Onsite Customer Work - An Extreme Success Story”<sup>[56]</sup>という従来研究においては、顧客が開発現場に常駐（オンサイト顧客）して、以下のような活動を行なったことが紹介されている。

- ・ オンサイト顧客（Onsite Customer）と開発チーム（Development team）の役割決め。
- ・ ユーザーストーリー（User Story）の細分化。
- ・ 様々な課題（issues）の解決。
- ・ 新しく開発する機能（features）を出来る限りシンプルにするように注文。

その結果、以下のような効果があったことが記されている。

- ・ システムをリリース（delivery）する遥か前に、顧客はシステムに詳しくなった。
- ・ スコープ（scope）と無駄な作業が減った。
- ・ 費用を減らし、製品の品質を向上させ、顧客に高い価値を届けられた。

しかしながら、システムの運用が楽になったとは書かれていない。それに、そもそも経

営者はシステムの仕組みを知らないままである。

このような問題に対して、経営者や運用者に開発の仕方を教えつつ開発に参加してもらうというアプローチで解決を図る。その際、懸念されるのは、どうすれば彼らは挫折することなく開発の仕方を学習してシステムの仕組みを理解することができるのか？である。次の2.5節では、この学習のやり方について述べる。

## 2.5. 教育・学習

### 2.5.1. 教える

「学問に王道なし」という言葉があるように、物事の学習の仕方には色々なやり方がある。同様に、人に物事を教える方法も、お互いの知識の量や性格や置かれた状況により異なってくるので、どのようなやり方を選択すれば良いのか悩む。ひとまず学生への一般的な指導方法のモデル<sup>[57][58]</sup>を引用して、一部を抜粋した表を以下に示す。

Table 10 “Comparison of teaching methods for students” (Based on [57])

#	教授法	指導者（教える側）の役割	学生（教わる側）の役割	情報
1	講義	専門家として：思考を導く・知識を持つ・学生を評価する。	受け手として：受動的・不活発・空っぽ。	指導者によって構成され与えられる。
2	直接指導	指導者として：学びを統制・暗記や模倣を誘導・学生を評価する。	弟子として：反応を示す・やや活発・指導者の指示を待つ。	ほとんど指導者によって構成され与えられる。
3	ケーススタディ	相談役として：講義や助言をすすめる・学習環境を整備・学生を評価する。	ゲストとして：反応を示す・やや活発・自分の経験を応用する。	〃
4	探求に基づいた発見学習	ミステリーの書き手として：部分をつなぎ合わせて発見へと導く・手がかりを与え、できごとを予示する・学生を評価する。	探偵として：手がかりを拾い上げる・やや活動的・証拠を探し出す。	〃
5	問題中	情報源として：内容や問題解決	問題解決者として：情報	〃

	心学習	を明確に教える・学生に適した問題を設定する・学生の世界へと転換させる・学生を評価する。	源を評価する・発散的な解決策を作る・活発。	
6	シミュレーション型学習	舞台監督として：状況管理・シミュレーションやゲームの進行を管理・袖から見守る・報告を聞く。	役者として：シミュレーションやゲームを体験する・現れる状況や変化に対して反応する・活発。	〃
7	ロールプレイ（専門家の外套）	旅行代理店として：グループ内からの学びを可能にする・任務（専門家を遂行するために知るべきことを発見させるための道筋を示す・旅の道案内・状況についての結果報告を聞く。	旅行者として：積極的に旅を体験する・歴史的な視点を通して行動する。	〃
8	問題に基づいた学習（PBL）	コーチとして：問題状況を提示・雛形を作りコーチし姿を消す・共同者としてプロセスに携わる・学びを評価する。	当事者として：積極的に状況の複雑さに取り組む・調査し、問題を内側から解決する。	指導者は学生が知るべきことを特定せず提示しない。 ほぼ全てが学生によって収集・分析される。

上記のモデルを、開発者（教える側）が運用者（教わる側）に開発のやり方を教えてシステムの仕組みを理解してもらう時に応用しようとする、以下のような懸念事項が浮かび上がってくる。

- ・ 教わる側が、物事の学習に対して消極的・受動的であるならば、#1 と#2 以外の方法は適用が難しいかもしれない。
- ・ 教わる側が、プライドが高くて人から物事を教わることに苦痛・抵抗を感じるならば、#8 の PBL (Problem-Based Learning) 以外の方法は、相手に直接訴えかけるプロセスを含んでいるため、適用が難しいかもしれない。

特に、企業経営者は普段は人に指図する立場にあり、人から指導されることには馴れていなくて、人によっては嫌がる人もいることは十分に考えられる。

よって、開発者が運用者に開発のやり方を教える時、相手によっては、教師が学生に指導するようなやり方をそのまま持ち込んで適用するのは難しいことが分かる。このような教わる側・学ぶ側の心理的な障害や障壁という問題を克服するために、次項 2.5.2 では学習ピラミッドおよび学習パターンというものを考察してみる。

## 2.5.2. 学ぶ

### (1) 学習ピラミッド

「学習ピラミッドは、米国の National Training 研究所 (NTL) の応用行動科学に源を發し」<sup>[59]</sup>ている。以下に図を示す。

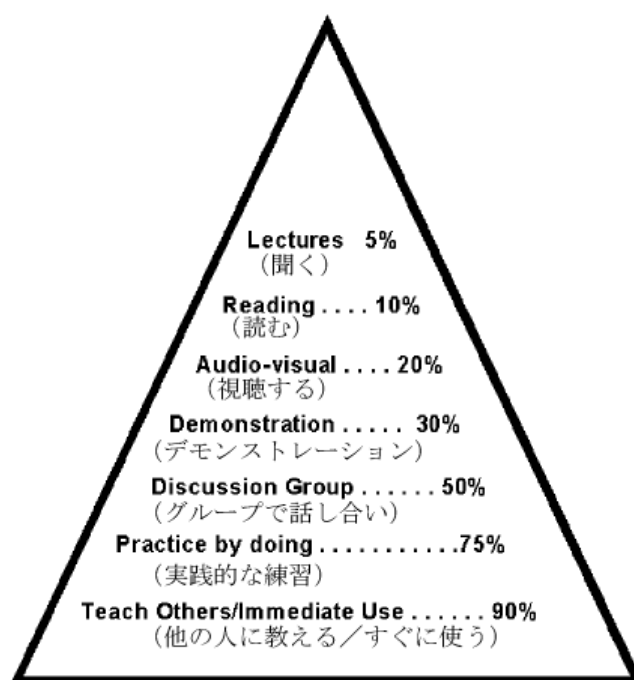


Figure 12 “Learning Pyramid”<sup>[59]</sup>

「パーセントの数字は、示された活動方法や教え方の後に情報を「記憶している割合」の平均です。実際に、この図は 1960 年代の初期に NTL によって開発され使用されました。しかし、NTL は、もはやこの数値の根拠となるオリジナルの研究結果を持たず見つけることもできません。1954 年に、わずかに異なった数字が示されている同様のピラミッドが、ニューヨークのエドガー・デール・ドライデン出版社によって出版された Audio-Visual

Methods in Teaching という本の中に出ていました。」<sup>[59]</sup>

学習方法が、ピラミッドでより下にあればある程、記憶効果が増すモデルであるから、あらゆる状況またはコンテキストで、常に一番効果が高い「他の人に教える／すぐに使う」ことができれば良いだろうが、前項でも触れたとおり、教わる側・学ぶ側の心理的な障害や障壁といった問題もあるから、現実的には難しいだろうと想像できる。

では、どの状況またはコンテキストで、どの学習方法を選択すれば良いのかを考える時、参考になりそうなのが、(2)で述べる学習パターンである。

## (2) 学習パターン

学習パターンとは、「よりよい学びを実現するためのコツを抽出・記述したもの」であり、「各パターンには、学びにおいて問題になりがちな状況と、それをうまく解消してくれるようなコツがまとめられて」いる<sup>[60]</sup> 一種の学習モデルである。付録の Table 28 に示すとおり、2011年7月17日現在、40個のパターンから構成されている。そして、同じく付録の Table 29 に示すとおり、各パターンはどのような状況またはコンテキスト（例：プロジェクトを立ち上げる。作品をつくっている。自分がいま何をすべきかわからない。）で活用できるのか定義されている。

Table 10 で示した学生への教授法と同様に、Table 28 で示した学習パターンも、学校（大学）教育の中で生み出されたものであり、想定する対象者が教員や学生であるのと、そもそも一種の方法論でもあるため、今回のような研究を行うコンテキストに適用するためにはカスタマイズが必要になるかもしれない。しかし学習パターンは、Table 29 の例でも示したとおり、「自分がいま何をすべきかわからないとき」といったマイナスの状況またはコンテキストも考慮しているので、教わる側・学ぶ側が心理的な障害や障壁を乗り越えられるよう配慮されていると考えられ、また本研究の実験でも、その様な状況が起こることを予想できたため、試す価値はあると考えた。

### 3. モデル（CAL 開発手法）の提案

#### 3.1. CAL 開発手法とは？

本研究で構築するものは、システム開発の発注側である顧客（特に経営陣）がシステムの仕組みを学習しつつシステムを開発するためのモデルである。その効果として、企業経営または業務を戦略的に遂行しきれていない経営者または経営陣が、それらを戦略的に行えるようになることを狙っている。このモデルを、顧客能動学習型（CAL: Customers' Active Learning）開発手法と名付ける。

まず、この CAL 開発手法が、従来のシステム開発とどう違うのかを簡単に表したものが以下の図である。

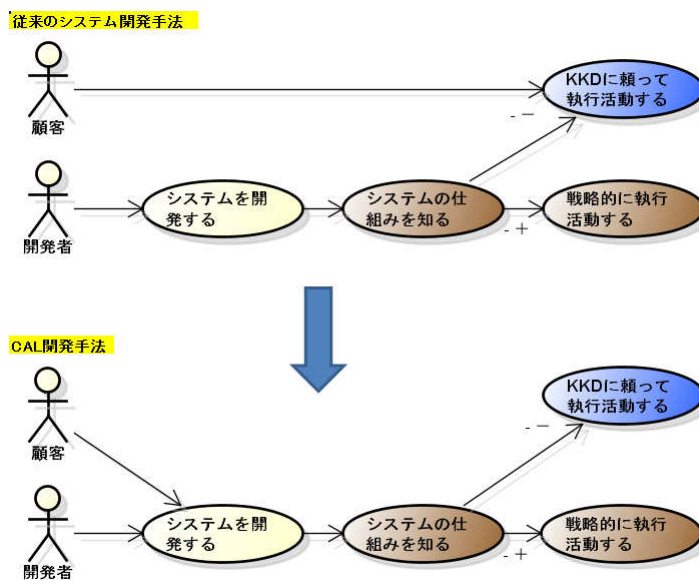


Figure 13 “Comparison of a traditional system development methodology and CAL”

上の図で使用している記号の説明は、この後の Figure 14 で行う。

#### 3.2. 目的・目標

目的を「最終的に到達すべき地点」<sup>[61]</sup>、目標を「目的にたどり着くための道筋やルート」<sup>[61]</sup>とするならば、CAL 開発手法の目的および目標は下記のとおりになる。

- ・ 目的：経営者または経営陣が、より戦略的に企業経営や業務を執行すること。
- ・ 目標：経営者または経営陣が、自社の業務システムの開発に積極的に参加することを通じてシステムの仕組みを知り、それを企業経営や業務の戦略策定に活かすこと。

上記の目的・目標を達成するための仕組みを、2 章で行った現状分析を踏まえて説明す

ると、以下のようになる。

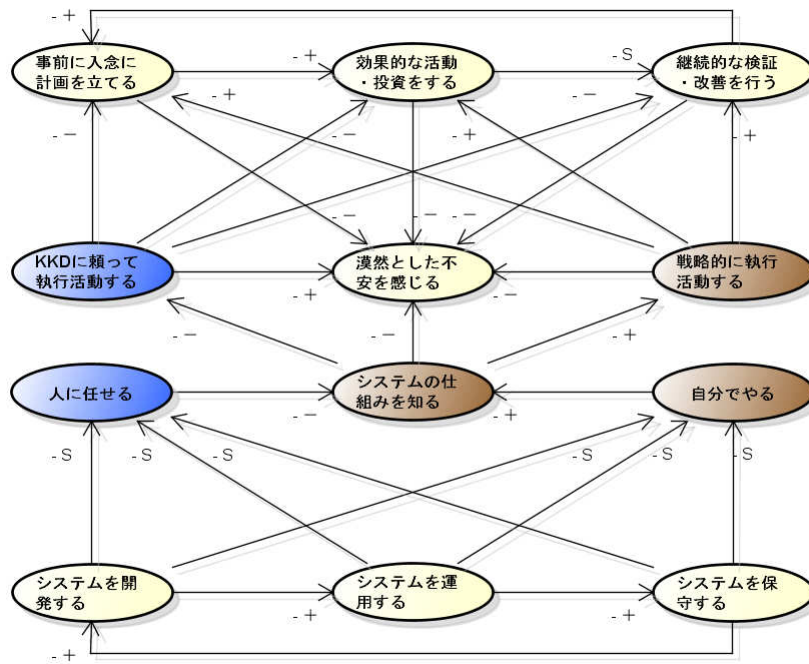


Figure 14 “Causal Relationship Diagram – Strategic Management” (Original)

【凡例】

- ・ 丸の中の文字：何らかの活動を表す。
- ・ 丸の背景色が青:CAL 開発手法を体験する前の経営者または経営陣の活動スタイル。
- ・ 丸の背景色が赤:CAL 開発手法を体験した後の経営者または経営陣の活動スタイル。
- ・ 矢印：ある活動が別の活動に対して後述する影響を及ぼすことを表す。
- ・ “+”：ある活動が矢印の先の活動に対してプラスの影響を与えることを表す。
- ・ “-”：ある活動が矢印の先の活動に対してマイナスの影響を与えることを表す。
- ・ “S”：ある活動が矢印の先の活動に対して殆ど影響を与えないことを表す。

通常、自社の業務システムを開発する時、業者に作業を委託する。しかし、CAL 開発手法においては、顧客は開発を業者に丸投げにするのではなく、業者から開発の仕方を教わりながら、顧客自身も積極的に参加して開発する。その結果、中小企業の経営者または経営陣が自社の業務システムの仕組みを知り、やがて企業経営や業務の執行により戦略的になることを目指している。それができれば、事前に入念に計画を立てたり、効果的な投資活動をするようになったり、業務の継続的な検証・改善を行うようになるといった効果も期待できる。



### 3.3. 対象領域

CAL 開発手法を適用する対象領域を、中小企業（特にシステムの開発や保守を業者に委託するための資金が乏しい企業）の業務システムのソフトウェア部分とする。しかし、これだけだと他のシステム開発手法との違いが分かりづらい。そのため、まずは 2.3.1 項の Table 5 を基準として比較する。

Table 11 “Comparison of 5 Software Development Methodologies in target areas” (Based on [38][39][40])

開発の対象領域や種類	Waterfall	IID	Agile	CAL
ビジネス要求が変化する領域	×	✓	✓	△
リスクが高い領域	△	△	✓	△
市場競争が活発な領域	×	△	✓	△
大規模開発	△	✓	△	×
分散拠点開発 <sup>5</sup>	✓	△	×	×
組織間をまたぐ開発チームによる開発	✓	△	△	×
組み込みシステム開発	✓	△	△	×
資金が乏しい企業の業務システム開発	△	△	△	✓

#### 【凡例】

- ・ ✓：適している。
- ・ ×：適していない。
- ・ △：どちらとも言えない。

上記の表は、元の表 Table 5 に対して、一番右の列に「CAL 開発手法」を追加した。更に一番下の行に「資金が乏しい企業の業務システム開発」という項目も追加した。その結果、一番右の列と一番下の行に新しくセルが増えたので、それぞれの評価も追加した。

各セルに記述した評価の経緯・理由は以下のとおりである。

まず、CAL 開発手法は、顧客に積極的にシステム開発に参加してもらうことで、変わりやすいユーザー要求を適時的に取り込むことも狙っているので、「ビジネス要求が変化する

<sup>5</sup> オフショア開発を含む。

領域」に“✓”マークを付けた。また、「資金が乏しい企業の業務システム開発」用に考案したので、それにも“✓”マークを付けた。

一方で、前提条件において、企業規模が小さい企業の業務システムを対象にしているため、そういった企業には縁がなさそうな「大規模開発」や「分散拠点開発」や「組織間をまたく開発チームによる開発」に“×”マークを付けた。また、ハードウェアも考慮に入れないので、「組み込みシステム開発」にも“×”マークを付けた。

上記以外の CAL 開発手法に関する項目については未知数なので、“△”マークを付けた。

最後に、「資金が乏しい企業の業務システム開発」に関して、他のシステム開発手法では特に考慮されていないので、こちらも“△”マークを付けた。

以上、改めて表を振り返ってみると、CAL 開発手法の対象領域は他の開発手法と大きく異なっていることが分かる。特に XP 開発手法をモチーフとしたにも関わらず、Agile 開発手法とも重なっていない。

### 3.4. 前提条件

CAL 開発手法を企業の業務システム開発に適用する場合、いくつか前提条件があるので、それを以下に記す。

- ・ 経営者または経営陣が、企業経営または業務を KKD に頼って遂行している。（戦略的ではない。）
- ・ 経営者または経営陣に、システムの開発・保守・運用に関与した経験が殆どない。
- ・ 経営者または経営陣が、システムの一部を開発する。

上記の前提条件についても、他の開発手法と比較を行う。2.3.4 項の(3)で、各種アジャイル開発手法の前提条件を比較した Table 7 を元に以下の表を作成した。

Table 12 “Comparison of CAL Development and Agile Software Development Methodologies”  
(Based on Table 7)

条件	XP	Scrum	Lean	FDD	AUP	Crystal	DSDM	CAL
少人数のチーム	✓	✓	✓	×	×	—	✓	✓
要件の変化が激しい	✓	✓	✓	✓	—	—	×	—
分散チーム	×	✓	✓	✓	✓	×	×	—

儀式度が高い	×	×	—	—	✓	—	✓	×
重大度が高いシステム	×	—	—	—	—	✓	×	×
複数の顧客／利害関係者がいる	×	✓	✓	—	—	—	×	—
顧客による開発	×	×	×	×	×	×	×	✓

#### 【凡例】

- ・ ✓：適している。
- ・ ×：適していない。
- ・ —：どちらとも言えない。

上記の表は、元の表 Table 7 に対して、一番右の列に「CAL 開発手法」を追加した。更に一番下の行に「顧客による開発」という項目も追加した。その結果、一番右の列と一番下の行に新しくセルが増えたので、それぞれの評価も追加した。

各セルに記述した評価の経緯・理由は以下のとおりである。

まず、CAL 開発手法は、3.3 節で説明したとおり、資金に乏しい中小企業の業務システムを想定しているので、当然、大人数の開発チームを構成することはできない。よって、ドキュメント作成などの作業に時間を割ける余裕がないので、「儀式度が高い」には“×”マークを付けた。また、「重大度が高いシステム」の場合、開発チームの規模が小さいと、様々なリスクを制御できない恐れがあるので、こちらも“×”マークを付けた。

一方で、繰り返しになるが、開発チームは少人数にならざるを得ないので、「少人数のチーム」に“✓”マークを付けた。また、顧客を支援して彼らに開発に積極的に参加してもらわなければならないので、「顧客による開発」にも“✓”マークを付けた。

上記以外の CAL 開発手法に関する項目については未知数なので、“—”マークを付けた。

最後に、「顧客による開発」は、他のシステム開発手法では元々想定に入っておらず、そもそも顧客の役割も決まっているので、全て“×”マークを付けた。

## 3.5. 原則

### 3.5.1. はじめに

この節では、CAL 開発手法の原則を説明して、それらを他のシステム開発手法のものと比較して、CAL 開発手法の位置付けを明らかにする。

### 3.5.2. Agile 開発手法との比較

まず、Table 6 で紹介した Agile 開発手法で定義している 12 の原則の観点で CAL 開発手法を説明する。

Table 13 “CAL Development Methodology in Comparison with 12 Principles of Agile Development Methodology” (Based on Table 6)

#	Agile Principle	CAL	Keep or Try in CAL
1	Early and continuous delivery	✓	原則、一ヶ月以内にシステムの一部または全体をリリースする。
2	Welcome changing requirements (ie: Adaptive planning, Requirements prioritization)	✓	顧客からの要求の追加・変更に応じる。ただし、要求の優先度も管理する。また、顧客が自身の要求を元に自らも実装する。
3	Time boxing	✓	原則、一ヶ月以内にシステムの一部または全体をリリースする。
4	On-site customer		顧客に開発現場への常駐を強要しない。
5	Team environment	✓	顧客の積極的な開発参加は必須である。
6	Face-to-face interaction	✓	メールよりも電話、電話よりも対面でのやり取りを重視する。
7	Working software	✓	開発者は顧客に言葉や文章だけで説明するよりも、プロトタイプやデモを見せて分かり易く伝える。
8	Sustainable development	✓	顧客も開発者も開発を途中放棄しないようにする。
9	Technical excellence and good design (ie: Refactoring)		初心者に開発してもらうので、高度な技術や洗練された設計は必ずしも求めない。
10	Simplicity		初心者に開発してもらうので、どうしても無駄が生じてしまう恐れがある。
11	Self-organizing team	✓	開発者は顧客からの指示を待つだけでなく、気づい

			たことは直ぐに顧客に伝える。また、自分で解決を図っても良い。
12	Reflections and tunings, adjustments (ie: Empirical process, Kaizen)	✓	KKD ばかりに頼った企業経営や業務の遂行を見直すきっかけを顧客に与えることができるように、開発者は作業の進め方の改善を促していく。

上記の表の左から 2 番目の列“Agile Principle”は、元の Table 6 の“Keyword”列から転記した。その右隣の列“CAL”は、CAL 開発手法が前述の“Agile Principle”を考慮に入れているかどうかを表している。そして一番右端の列“Keep or Try in CAL”は、その考慮に入れている理由や、どのようにして考慮しているのかを示している。さて、上記の 12 の原則を全て踏まえたものが Agile 開発手法に分類されると定義するならば、CAL 開発手法は、12 個の原則の内、9 つまでしか踏襲していないので、純粋な Agile 開発手法ではないが、それに準ずるものであるとみなすことができる。

### 3.5.3. 他の開発手法との比較

前項では Agile 開発手法そのものと比較したが、今度は具体的な開発手法と比較する。

Table 14 “CAL Development Methodology in Comparison with IID and XP, Scrum” (Based on [62])

#	Principle	IID	XP	Scrum	CAL	Keep or Try in CAL
1	Iterations and increments	✓	✓	✓	✓	リリースは繰り返し行い、その都度、新しい機能を追加する。
2	Internal and external releases	✓				開発企業内向けの非公開のリリースは行わない。
3	Time boxing	✓	✓	✓	✓	Table 13 の#3 に記載。
4	No change of started projects	✓		✓		イテレーションの期間中、経営陣はチームや個人に対して作業を追加できる。
5	Incremental deliveries	✓				リリースの度に常に新しい機能を追加する訳ではない。(不具合修正

					だけの時もある。)
6	On-site customer	✓	✓		Table 13 の#4 に記載。
7	Frequent face-to-face interaction	✓	✓	✓	Table 13 の#6 に記載。
8	Self-organizing teams	✓	✓	✓	Table 13 の#11 に記載。
9	Empirical process	✓	✓	✓	Table 13 の#12 に記載。
10	Sustainable discipline	✓		✓	Table 13 の#8 に記載。
11	Adaptive planning	✓	✓	✓	Table 13 の#2 に記載。
12	Requirements prioritization	✓	✓	✓	Table 13 の#2 に記載。
13	Fast decision making		✓		障害が発生しても、1 時間以内に判断しなければならない訳ではない。
14	Frequent integration	✓	✓	✓	Table 13 の#1 に記載。
15	Simplicity of design	✓			Table 13 の#10 に記載。
16	Refactoring	✓			Table 13 の#9 に記載。
17	Team code ownership	✓		✓	顧客と開発者はソースコードを共有する。(一方だけが管理しない。)

上記の表は元の資料<sup>[62]</sup>に対して、“CAL”という列と“Keep or Try in CAL”という列を追加した。意味は Table 13 の同名の列と一緒にある。上記の表と Table 13 は項目が一部被っている。これを見ると、CAL 開発手法は 17 項目中 10 項目の Principle に従っているが、その全ての項目は XP 開発手法にも含まれている。また、Scrum 開発手法とは 8 項目を共有している。一方で、IID とは 2 項目しか重なっていない。よって、これだけを見ると、CAL 開発手法は XP 開発手法から派生したサブセットとみなすことができるだろう。

### 3.6. 役割分担

#### 3.6.1. はじめに

この節から、CAL 開発手法の具体的な内容について説明する。また前節と同様に、他のシステム開発手法といくつかの視点・観点で比較をして、相違点を明確にする。次項以降で、CAL 開発手法を適用するシステム開発における顧客や開発者の役割分担を説明するが、

最初に、システム開発における一般的な役割分担と、2.3.4 項の(4)の④で説明したエクストリームプログラミング (XP) 開発手法の役割分担を、ユースケース図を使って紹介・説明した後、最後にそれらを CAL 開発手法での役割分担と比較する。

### 3.6.2. 一般的な役割分担

まず、システム開発における一般的な役割分担を図で表すと以下のようになる。

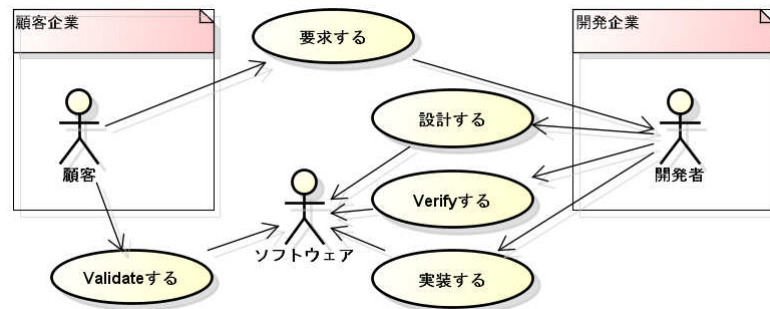


Figure 15 “General roles in a system development” (Original)

Figure 2 でも示したが、システム開発の工程を時系列で並べると、要求→設計→実装→Verification→Validation となる。それを上の図 Figure 15 と照らし合わせると、顧客は主に、最初の工程の要求と最後の工程の Validation でしか関わらず、中間の設計・実装・Verification については開発企業任せ、いわゆる外注任せ・丸投げ状態になる。

しかし、このような役割分担では、開発者がソフトウェアの Verification まで終わらせて、顧客企業まで赴いてソフトウェアを Validation してもらうまで、要求漏れや要求誤りが判明しないリスクを抱えることになる。

このようなリスクの回避を目指したのが、次項で示す XP における役割分担である。

### 3.6.3. XP 開発手法における役割分担

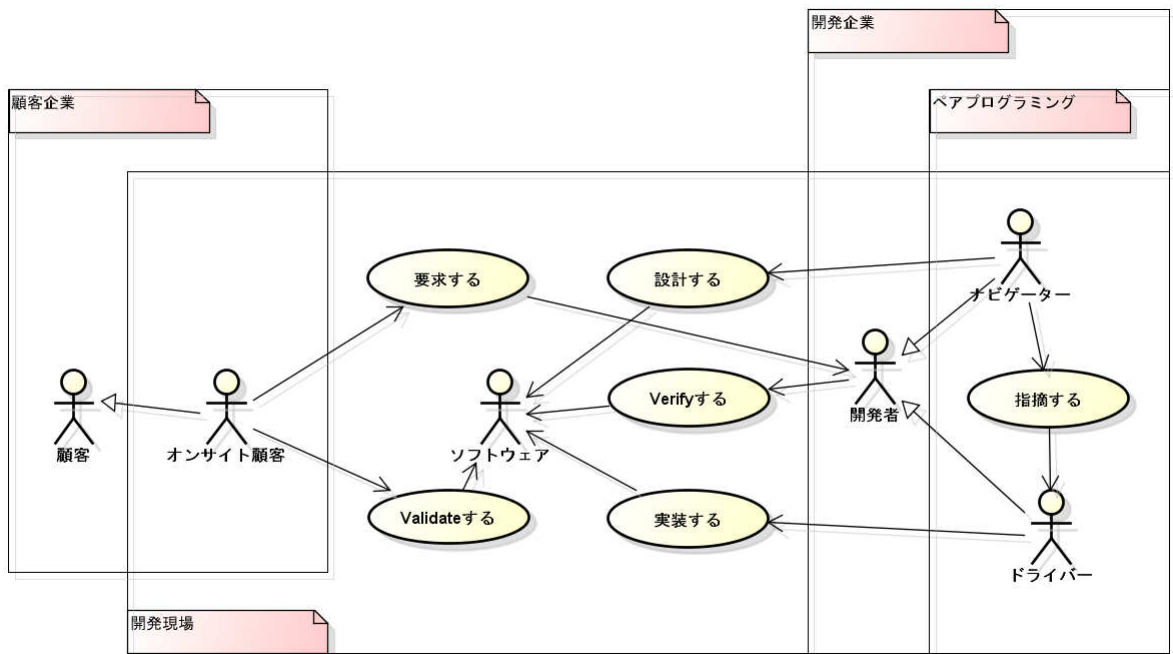


Figure 16 “Roles in XP Development” (Original)

まず、図の右側を見ると、「ナビゲーター」と「ドライバー」という二種類の開発者がいて、その二者をペアプログラミングという枠で囲っていることが分かる。ペアプログラミングとは文字どおり、二人の開発者がペアになって、一つのモニターを一緒に見ながら、一つのキーボード・マウスを共有してプログラミングを行うことである。ここで特徴的なのは、ドライバーと呼ばれる開発者が実装を担当する一方で、ナビゲーターと呼ばれる開発者はそれ以外のことを考える点である。プログラミングを車の運転に例えるなら、運転席に座るドライバーが車を運転して、助手席に座るナビゲーターはドライバーが運転に集中できるように周囲に気を配るようなものである。つまりペアプログラミングは、ドライバーが実装に集中できるように配慮されたプラクティスである。このペアプログラミングの考え方は CAL 開発手法にも取り入れているが、詳しくは 3.7 節で述べる。

一方、図の左側には、オンサイト顧客という開発現場に常駐する役割のアクターがいる。一見、Figure 15 の顧客と役割が変わらないように見えるかもしれないが、開発現場に常駐して会議や成果物のレビューに参加することによって、顧客の開発者に対する要求伝達やソフトウェアに対する Validation の回数が飛躍的に増えて、その結果、開発工程の最後の方で要求漏れや要求誤りが発生するリスクを減らすことになる効果を期待できる。

しかしながら、資金や人材に乏しい中小企業にとって、業務に詳しい人材を業務から切



り離して、開発現場に常駐させて開発のチェックをするのは、少なからず困難である。

このような問題を解決するのが、次項で示す CAL 開発手法での役割分担である。

### 3.6.4. CAL 開発手法における役割分担

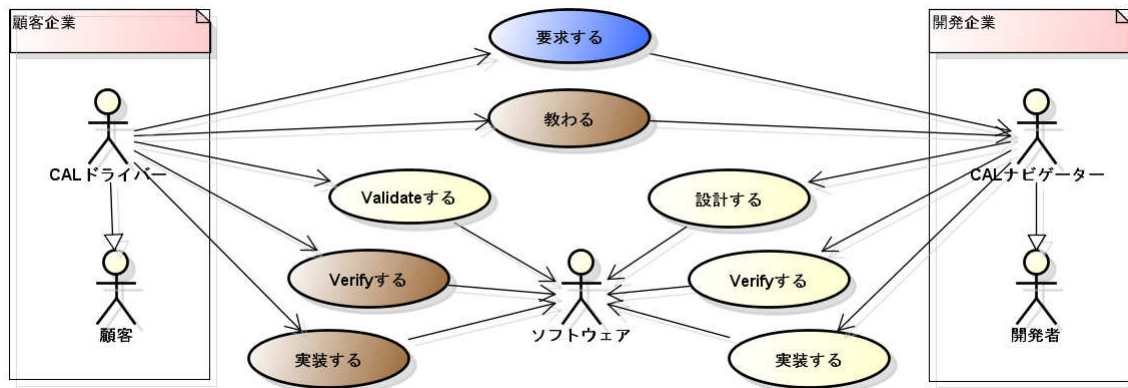


Figure 17 “Roles in CAL Development” (Original)

まず、上のユースケースには、「CAL ドライバー」と「CAL ナビゲーター」というアクターが登場している。それぞれ、前の 3.6.3 項で述べた XP 開発手法のペアプログラミングにおける「ドライバー」と「ナビゲーター」に類似している。

説明を簡潔にするために、以降、それぞれ「XP ドライバー」、「XP ナビゲーター」とする。相違点は、XP ドライバーが純粋な開発者であるのに対して、CAL ドライバーは顧客が務めている点であり、また、XP ナビゲーターは実装を行わないのに対して、CAL ナビゲーターは実装も行う点である。

これを車で例えると、XP 開発手法におけるドライバーとナビゲーターは、運転席にしかハンドルが付いていない一般的な車に乗って運転（実装）するようなものであるのに対して、CAL 開発手法におけるそれは、助手席にもハンドルが付いている自動車教習所の車に乗って、共に運転（実装）しているようなものである。

以降の説明では、特に断りがない限り、顧客は CAL ドライバーを指し、開発者は CAL ナビゲーターを指すこととする。

更に上の図で色が付いているユースケースは、Figure 15 や Figure 16 と比較して、主体となるアクターの負担の増減が大きいものを表している。即ち、赤色のユースケースは負担が増えるものであり、反対に青色のユースケースは負担が減るものである。これらを従来の開発手法と比較したメリットおよびデメリットとして捉えると、下記のとおりになる。

### (1) 顧客 (CAL ドライバー) のメリット

- ・ 開発の仕方が分かり、自分で開発できるようになれば、開発者に要求を伝える必要がなくなる。

### (2) 顧客 (CAL ドライバー) のデメリット

- ・ ある程度、開発の仕方を覚えなければならない。挫折するかもしれない。

### (3) 開発者 (CAL ナビゲーター) のメリット

- ・ 顧客から要求をヒアリングする手間が省ける。
- ・ 顧客の方で開発してもらえれば、その分、実装・Verify したり、顧客に Validate してもらったり、再実装したりする手間を省くことができる。

### (4) 開発者 (CAL ナビゲーター) のデメリット

- ・ 顧客が実装したものを Verify しなければならず、場合によっては再実装しなければならない。
- ・ 開発のノウハウやコツを顧客に教えてしまうことになり、自らの収入を減らすことに繋がりがねない。

Figure 17 を見ただけなら、顧客の負担が一方的に増えているように見えるが、上記のように双方のメリット・デメリットを並べて総合的に見ると、果たしてそうとも言えない。しかし、双方のデメリットを放置することは何らかのリスクを増やすことになるので、これらのデメリットをできるだけ抑えなければならない。そのため、(2)で示したデメリットの対策として、次の 3.7 節で顧客への指導の仕方について説明する。また、(4)で示したデメリットに対しては、その後の 3.8 節で顧客と開発者がそれぞれ実装する範囲を説明する。

## 3.7. 顧客 (CAL ドライバー) の学習

### 3.7.1. 教える

前項 3.6.4 の(2)でも触れたとおり、CAL 開発手法のネックになるのが、まずは顧客が開発の仕方を学習しなければならない点である。2.5.1 項の最後の方でも少し触れたが、教える時に注意しなければいけないのは、相手のプライドを傷つけないことである。特にプライドが高い場合、指導方法は限られてくるだろうと想像できる。Table 10 で紹介した指導方法を、生徒の特徴に合わせて振り分けてみると以下の表のようになると考えられる。

Table 15 “Teaching Method Matrix” (Original)

生徒の特徴	消極的	積極的
プライドが高め・ 変化に弱い	<ul style="list-style-type: none"> <li>・ ケーススタディー</li> <li>・ 探求に基づいた発見学習</li> </ul>	<ul style="list-style-type: none"> <li>・ 問題中心学習</li> <li>・ シミュレーション型学習</li> <li>・ ロールプレイ</li> </ul>
プライドが低め・ 変化に強い	<ul style="list-style-type: none"> <li>・ 講義</li> <li>・ 直接指導</li> </ul>	<ul style="list-style-type: none"> <li>・ PBL</li> </ul>

本研究では、開発者が経営者や運用者の特徴を見抜いて、Table 12の中から最適な指導方法を選択する。

### 3.7.2. 学ぶ

#### (1) 学習ピラミッド

前の 3.7 項では教える側の立場に立って、指導するモデルを説明した。今度は、学ぶ側の立場に立って、学習するモデルを説明する。まず、顧客がシステムの仕組みを知るためにどのようなアプローチを取れば良いのか？本研究では、2.5.2 項で説明した「学習ピラミッド」を応用する。表にすると以下のようなようになる。

Table 16 “Comparison of approaches to know mechanism of a business system with Learning Pyramid” (Original)

#	学習ピラミッド	システムの仕組みを知るためのアプローチ	重み
1	N/A	知りたくない	0
2	Lectures	詳しい人から話を聞く	5
3	Reading	ドキュメントを読む	10
4	Demonstration	デモンストレーションを見る	30
5	Discussion Group	勉強会に参加する	50
6	Practice by Doing	運用に参加する	75
7	Teach Others / Immediate Use	開発・保守に参加する	90
8	Audio-visual	その他のアプローチを採用する	20

上掲の Table 16 の項目の意味や他の項目との関連性を説明する。まず、Figure 12 の「学

習ピラミッド」の各アプローチを縦に並べている。続いて、それに対応する「システムを知るためのアプローチ」を洗い出している。一番右端の「重み」には、同じく Figure 12にあるパーセンテージの数字（記憶している割合）から%を外した数値を入れている。

このモデルが何を意味しているのかと言うと、CAL 開発手法の目標は、3.2 節でも説明したとおり経営者または経営陣がシステムの仕組みを知ることであるが、その目標をどの程度達成できるかの目安になる。式で表すと以下のようなになる。

「システムの仕組みへの理解度」＝

「システムの仕組みを知るためのアプローチ」×「重み」

この「システムへの理解度」が高ければ高いほど、経営者または経営陣はシステムの仕組みを知ることになり、それによって彼らは企業経営や業務をより戦略的に遂行するようになる。

## (2) 学習パターン

しかしながら、最適な指導方法や学習方法を選択したとしても、経営者または経営陣が教わることや学ぶことにストレスを感じるなどして、逃避・挫折してしまうリスクは十分に考えられる。その原因として考えられるのは、学ぶことに対するワクワク感が欠けているとか、逆に学ぶことから逃げない覚悟が足りないとかであると考えられる。こういった不安やリスクを減らすために、更に 2.5.2 項の(2)で紹介した学習パターン<sup>[63]</sup>を利用する。

システム開発時には、Table 29 で示した 3 つのコンテキストをまず想定することができる。その内、特に「自分がいま何をすべきかわからないとき」が、学習者つまり顧客に起こり得るコンテキストであると考えられる。そのため、CAL 開発手法では学習パターンを応用して、顧客が学習を継続できるように体制・役割・環境を整える。具体的には、以下の表のようなになる。

Table 17 “Learning Patterns when I don’t know what to do.” (Based on [63])

#	Learning Pattern	CAL
13	アウトプットから始まる学び	顧客にできる限り開発（実装）を任せる。ただし、実装に失敗した時のリスクをコントロールできるよう、修正時の影響範囲が小さい箇所に限定する。
20	広がり掘り下	顧客はシステムの開発方法について、せいぜい狭く浅くしか知ら

げの「T字」	ない。そのような場合、あれもこれも広く教えようとする、学習するのが大変になる。よって、まずは一つのことを深く知ってもらうよう、特定の作業だけを任せるようにする。
29 「はなす」こと わかる	顧客は、分からないことがあれば直ぐに開発者に相談する。また逆に、顧客が気軽に相談ができるように、開発者は相談を受けやすいような環境を作っておく。

### 3.8. 顧客（CAL ドライバー）と開発者（CAL ナビゲーター）の実装範囲

3.6.4 項の最後で触れたように、開発経験がない顧客が実装を行う場合、開発者は顧客の実装をレビューしなければならなくなるはずである。実装が複雑で難しい箇所をいきなり顧客が担当するのはリスクが大きいので、どこからどこまでを顧客および開発者がそれぞれ実装するのか、棲み分けをする必要がある。CAL 開発手法では、この棲み分けを、実装するファイルを修正した時の影響範囲で実現する。

Table 18 “The extent of the impact on modification” (Original)

#	修正した時の 影響範囲	システムまたはシステムの構成要素 (ファイル・データベース等)	実装担当者
1	修正箇所のみ	テキストファイル、画像ファイル、音声ファイル、...	顧客または開発者
2	同じ要素内	HTML <sup>6</sup> 、XML、...	顧客または開発者
3	他の要素	CSS、JavaScript、PHP、Java、SQL、DB、...	開発者

即ち、ファイルを修正した時、その影響範囲がその箇所に限られるようなファイル(#1)なら、顧客に実装を担当していただいても問題ない。しかし、CSS や JavaScript などのように、修正すると他のファイルにも影響を及ぼすようなもの(#3)の実装を顧客に担当していただくのはリスクが大きいため、従来どおり開発者が担当した方が良い。両者の中間である#2の修正もリスクが小さいとは言えないが、修正した時の影響範囲は同じファイル内に限られるので、顧客に実装を担当していただいても構わない。

また、開発者は顧客が目の前の実装に集中できるように開発環境を整えたりして配慮し

<sup>6</sup> CSS や JavaScript 部分を除く。

なければならない。顧客にあれもこれも手広く担当してもらうのではなく、一つのことに集中できるようにした方が効率は良いからである。このことは 3.6.3 項で説明したペアプログラミングの話からも分かる。つまり、実装工程における顧客と開発者の関係は、ペアプログラミングにおけるドライバーとナビゲーターの関係に似ている。

## 3.9. 費用

### 3.9.1. 費用名目

システムを開発する時、費用を支払う発注側、即ち顧客は、通常その費用を「開発費」として認識・計上する。しかし、CAL 開発手法を適用したシステム開発では、顧客も積極的に開発を行うので、顧客にしてみれば、「何で俺達が開発しているのに開発費を支払わなければならないのか!？」という疑問が起きて、彼らの開発に対するモチベーションを削ることにもなりかねない。また、開発者による開発にかかった費用と、顧客自身が開発したものの費用を区別しないと、顧客による開発と開発者による開発のそれぞれの効果の違いが分かりにくい。

従って CAL 開発手法では、開発者による開発にかかった費用を従来どおり「開発費」とみなし、顧客による開発の費用を「教育費」とみなせば、それぞれの開発効果を区別できるとともに、前述した顧客のジレンマを取り除くことにも繋がるだろうと考えられる。

## 3.10. その他

### 3.10.1. 開発スケジュール

Agile 開発手法は、2.3.4 項でも述べたように 12 の原則<sup>[5]</sup>というものがあり、その中でイテレーションを「2-3 週間から 2-3 ヶ月」と決めている。CAL 開発手法でもイテレーションは同様とする。何故なら、システム開発の対象となる企業の規模は小さいことを前提にしているため、業務の規模もそれに比例して大きくないと考えられるので、必然的にイテレーションも Agile 開発手法より長くはならないだろうと想定できるからである。

### 3.10.2. まとめ

以上、振り返って見ると、CAL 開発手法は XP 開発手法から派生したにも関わらず、共通しているのは原則ぐらいで、対象領域や前提条件や役割分担は大きく異なっており、斬新な開発手法であることが分かる。

## 4. 効果の検証 1（能動的学習）

### 4.1. 実験の対象

今回、前の 3.3 節で述べた前提条件を満たす企業を探した。その結果、苗場スキースクールという企業が、会員・行事申込システムのリプレースを検討していることが分かり、経営者が CAL 開発手法を適用することに応じていただいたため、システム開発をしながら実験を行うことにした。

### 4.2. モデルの拡張

CAL 開発手法は他の Agile 開発手法と組み合わせて適用することができるので、Table 7 を頼りに当該実験の開発プロジェクトにふさわしい手法が他にないかどうか検討した。その結果、当該開発プロジェクトは、「少人数のチーム」、「要件の変化が激しい」、「分散チーム」、「複数の顧客／利害関係者がいる」という条件を有しているため、Scrum と Lean が最適であることが分かった。

次に、Figure 10 と照らし合わせてみた結果、システムが想定外の振る舞いをして、代わりに電話での対応が増えるだけであり、即ち“Comfort”（快適さ）が失われるだけであるため、Scrum と Lean のどちらでも良いことが分かった。

最終的に、開発者は Lean よりも Scrum に馴染みがあるので、今回の実験においては Scrum を採用して、これを CAL 開発手法と組み合わせて、当該開発プロジェクトに適用した。

### 4.3. 実験方法

#### 4.3.1. 概要

試作品を開発してプロダクトオーナーに提示した後、製品を開発する。Scrum のイテレーション（スプリント）は原則 30 日なので、開発の開始から 30 日後に初回リリースを実施する。その後、最終リリースまでイテレーションを複数回繰り返す、最終リリースに至るようにする。

開発前に作業量を見積り、実際の作業量と比較する。また、運用者の作業量を計測し、増減の推移をグラフ化し、傾向を分析する。最後に運用者からアンケートを取り、どのような効果があったのかを検証する。

#### 4.3.2. スケジュール

開発スケジュールは、以下の通りである。

- ・ 試作品開発イテレーション：2010年9月1日～9月30日
- ・ 製品開発イテレーション：
  - 第一期：2010年10月1日～10月31日
  - 第二期：2010年11月1日～11月23日
  - 第三期：2010年11月24日～12月18日

Scrum 開発手法では、イテレーションを原則 30 日間としているので、できる限りそれを超えないようにした。

#### 4.3.3. 体制・役割分担

開発体制は、Scrum に倣って、開発者（CAL ナビゲーター）と顧客（CAL ドライバー）で開発チームを構成する。それとは別に顧客の中からプロダクトオーナーも選ぶ。開発者が試作品を開発した後、開発者と顧客が共同で開発を行う。プロダクトオーナーがプロダクトバックログを管理して、開発者がスプリントバックログを管理した。

#### 4.3.4. 作業管理

開発者および顧客の開発作業をプロダクトバックログおよびスプリントバックログで管理する。プロダクトバックログとは、「プロダクトオーナーが作成する優先順位付けられた欲しい物リスト」である。<sup>[50]</sup> 今回の実験で作成したものは、Microsoft Excel ファイル形式である。参考までに、その一部を抜粋して付録（Table 30 “Product Backlog”）に掲載しておく。各項目は優先順位の高い順で並んでいる。

一方、スプリントバックログとは、「開発イテレーション（スプリント）計画中に、プロダクトバックログから優先順位の高い順で抜き出した欲しい物の小さなリスト」である。<sup>[50]</sup> 今回の実験では、Redmine というバージョン管理システムを使って実現した。プロダクトバックログと同様に、その一部を抜粋して付録（Figure 27 “Sprint Backlog”）に掲載しておく。

#### 4.3.5. 作業量の測定・分析

開発者および顧客の作業量を測定する。具体的には、コーディングの量を測定し、時間



の経過と共にどのように変化したのかをグラフ化する。

#### 4.3.6. 評価

前の 4.3.5 項で得られた結果を参考に、CAL 開発手法の評価を行う。また、顧客や利用者からのアンケート結果を集めて、更に直接ヒアリングして、フィードバックを得ることができたら、その結果を分析する。

### 4.4. 実験結果

#### 4.4.1. 開発結果

##### (1) システムのアーキテクチャ

開発者（CAL ナビゲーター）が顧客（CAL ドライバー）と共にシステムを開発するにあたり、顧客が実装した箇所ができる限りシステムの他の要素に影響を及ぼさないようにするために、システムを複数のモジュールに分けて、モジュールがお互いに疎結合している状態にしなければならないと考えた。その課題を解決する方策として、コンテンツマネジメントシステム（Contents Management System: CMS）に着目して、その中からデータベース（Database: DB）とスクリプト言語 PHP で構成される WordPress を当該業務システムのプラットフォームとして採用した。

##### (2) スケジュール

実は開発者は、2010 年 9 月に試作品を開発し始める半年以上前に、当該企業に CAL 開発手法での開発を提案した。しかし、その時は良い返事を得られなかった。プロダクトオーナーである経営者は決断をためらっていた。けれども、その後、営業シーズンが近くなった 8 月に再び提案したところ、他に解決の道がないこともあり、提案を受け入れてもらった。初回リリースは 3 日遅れてしまい、その後もしばらく遅延が続いたが、最終リリースには間に合った。

##### (3) 役割分担

まず、顧客企業の経営者にはプロダクトオーナーになっていただき、要求一覧の管理をご担当いただいた。また、顧客企業の役員に CAL ドライバー（以降、役員と略す。）になっていただいた。本来の肩書きは主任であるが、業務の執行に責任を有しているため、実質的には役員（経営陣）と

みなすことができ、CAL 開発手法が前提条件とする、システム開発に対する経営陣の積極的な参加を満たせると判断したためである。そして、本研究の研究者である高橋が CAL ナビゲーター（以降、開発者と略す。）を務めた。

4.3.3 でも述べたとおり、原則として、役員には影響範囲が限られた部分を修正してもらう。しかし、今回の実験で採用した WordPress は、システムに修正を加える時、基本的に管理画面上での操作に限られる。よって、最初に述べた原則を元に、開発者と役員の役割を分担すると、以下の Table 19 のようになる。

Table 19 “Roles and impacts of working by stakeholders” (Original)

役割	開発者	役員	利用者	影響範囲
テーマの変更、編集	○			サイト全体
プラグインの有効化、編集	○	○		サイト全体
ユーザーの編集	○			サイト全体
ファイルの編集	○			サイト全体
設定の管理	○	○		サイト全体
カテゴリーの管理	○	○		全ページ
リンクの管理	○	○		全ページ
ページの編集	○	○		1 ページ
ファイルアップロード	○	○		複数ページ
記事の投稿、編集	○	○		1 ページ
閲覧	○	○	○	1 ページ

上記の Table 19 にあるように、役員は影響範囲が主に 1 ページ内にとどまる作業が大半であり、サイト全体に及ぶものは殆どない。これは、サイト全体に影響が及ぶような作業に関する専門的知識を役員が有していないことと、そのような作業を行って、万が一、異常が発生した時に対応できないリスクを回避するためである。

一方で、ページの編集など影響範囲がページ内にとどまる作業を役員に委譲することにより、役員は開発の初期段階で WordPress の操作方法を学習することにもなった。これは学習の大半が運用・保守フェーズではなく開発フェーズの実装・テスト段階で発生することを意味しており、かつ開発者の作業負担を減らす要因にもなった。具体的には、通常であれば開発者が作成するはずのページの殆どを、役員が作成・編集した。そのことは、以下の Table 20 に記した。

Table 20 “Work results of operators at 2011/02/03” (Original)

	開発者	役員	合計
テーマの変更、編集	1	0	1
プラグインの有効化、編集	22	2	24
ユーザーの編集	4	0	4
ファイルの編集	0	0	24
設定の管理	63	12	75
カテゴリの管理	3	0	3
リンクの管理	23	0	23
ページの編集	79	153	232 <sup>7</sup>
ファイルアップロード	1	61	62
記事の投稿、編集	7	0	7
閲覧	?	?	38,041

上記の Table 20 を見てもらえれば分かるとおり、サイト全体もしくはページ全体に影響が及ぶような作業は、基本的に開発者が行なった。けれども、影響範囲が 1 ページ内に収まるような作業の大半は運用者が行なった。

この結果、以前の業務システムの時と比べて、役員の作業量は増えたものの、以前はページの作成・編集を全て開発者に依頼しており、出来上がるまで約 1 週間かかっていた。しかし、今回は自分自身でページを作成するので、その場で作業成果を確認することができるようになり、その分、運用上のストレスからは解放された。

<sup>7</sup> ページの編集の合計は、純合計ではなく延べ数。



Figure 18 “Comparison with the old site and the new site” (Original)

#### 4.4.2. 開発作業量の推移

##### (1) DB のアーキテクチャ

WordPress の場合、作成した Web ページのデータは、WP\_POSTS というテーブルに 1 ページにつき 1 レコードという形で保存される。更に、Web ページを修正した場合、新しいデータは以前のデータとは別のレコードに保存される。つまり、あるページを作成した後、3 回修正すれば、レコードは 4 件 (=最新のページが 1 件+履歴が 3 件) になる。

WP\_POSTS テーブルに保存されるページのデータには、ページのタイトルや内容の他、ページの ID、ページの種類、ページを作成した日時や修正した日時、ページの編集者の ID が含まれる。このため、この仕組みを利用すれば、データを取得して加工することによって、「誰が、いつ、どのページを、どれぐらい作成したのか？」ということが分かる。

ただし、今回の特殊なケースとして、WordPress の機能を拡張するために、いくつかのプラグインをインストールしているが、その内、一部のプラグインが Web ページのデータの一部を別のテーブルに保存しているため、これらのテーブルからもデータを取得しなければならない。

##### (2) データの抽出および集計

データの抽出作業をフロー図で表すと、以下の Figure 19 のとおりになる。

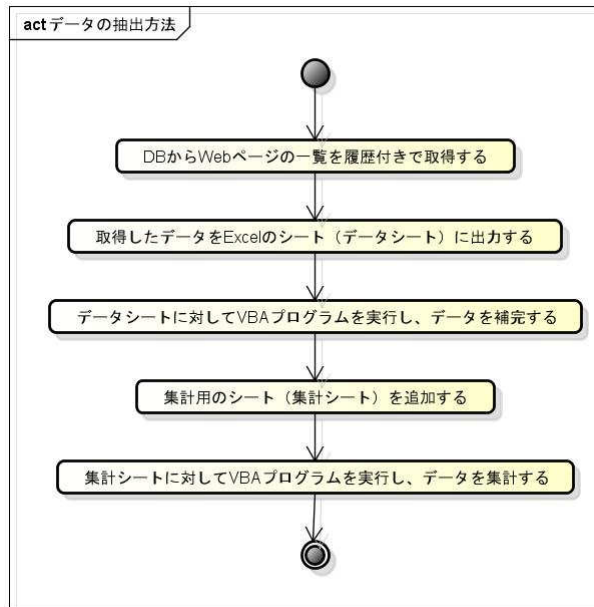


Figure 19 “Flow chart of works to extract data of web pages and summarize them” (Original)

まず、DB にログインしてアクセスしたら、SQL プログラム（コード不掲載）を実行して、Web ページのデータを取得した。

次に、取得したデータを Excel のシート（データシート）に出力した後、データ補完用の VBA プログラム（コード不掲載）を実行して、上記の SQL プログラムでは取得できなかったデータを補完した。今回、DB から取得・補完したデータは、Table 32 にある通り、全部で約 700 件に上った。

最後に、データを集計するためのシート（集計シート）を追加した後、集計用の VBA プログラム（コード不掲載）を実行して、データを集計する。今回、集計した結果は、Table 31 の通りである。

### (3) データの分析

前項で得られた Web ページのデータの集計結果を元に、分析結果を得ることができた。

#### ① ページの文字数

まず、全てのページから抽出した文字の数を担当者毎に区切ったグラフを示す。

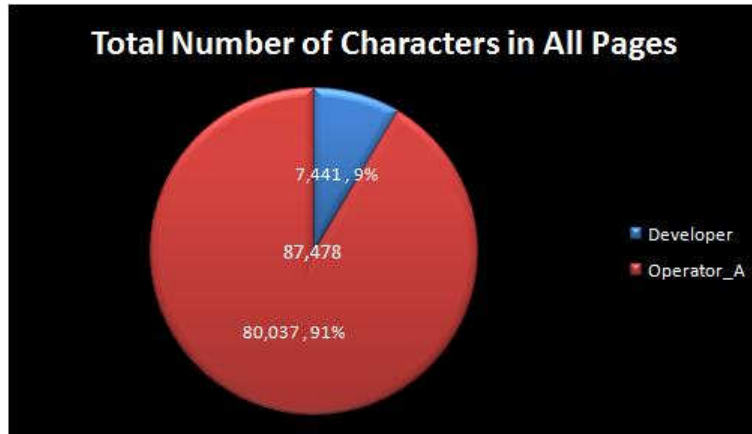


Figure 20 “Total number of characters in all pages” (Original)

Figure 20 のグラフが示す通り、ページに含まれる文字の数は9万近くになるが、その内の9割以上を役員が追加している。つまり、ページの内容の殆どは役員によって記述されたことを表している。

次に、ページに対して追加または削除した文字の累積数を時系列で示したグラフを示す。

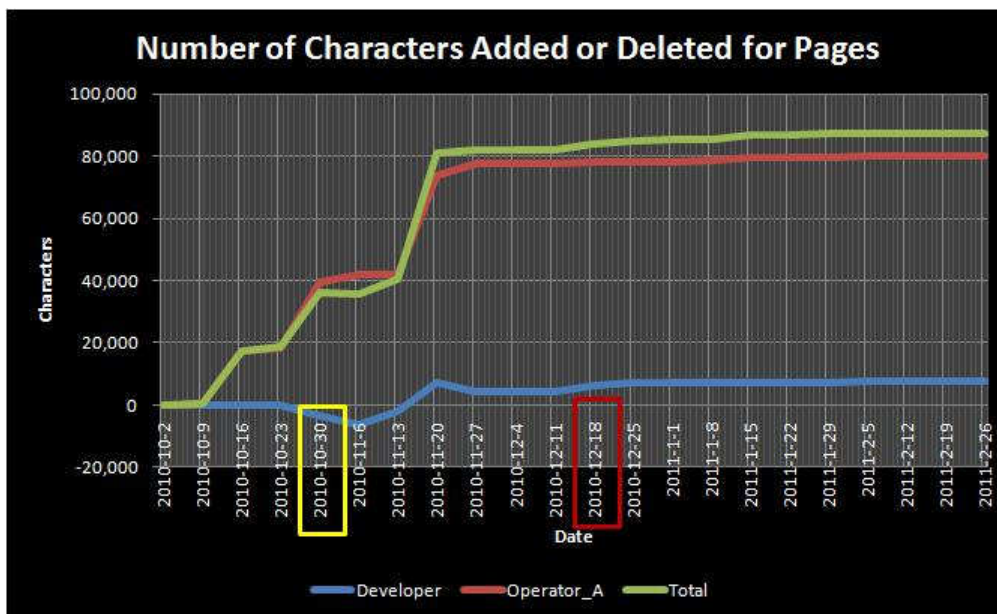


Figure 21 “Number of characters added or deleted for pages” (Original)

Figure 21 のグラフの横軸である Date の「2010年10月30日」と「2010年12月18日」を、それぞれ黄色と赤色の枠で囲っているが、これらはそれぞれ「リリース日」と「スクール開校日」であることを表している。

作業内容の詳細は、以下の②および③で詳しく述べるが、役員のパージ編集のペースが

3 段階に分かれて急激に伸びていることが見て取れる。これは、その直前のペースが緩やかな期間に役員はページ編集の仕方を学習しているのである。

## ② 役員（CAL ドライバー）の作業内容

まず、役員グラフの背景となる作業内容は以下の通りである。

- 役員は当初、ページを追加・削除・編集するという作業を行うに当たって、WordPress の操作方法を学習しなければならなかった。「2010 年 10 月 2 日」～「2010 年 10 月 9 日」までの 1 週間は、主にそういった作業を試しながら操作方法を学習していった。
- 操作方法を一通り学習した後、役員は、「2010 年 10 月 9 日」～「2010 年 10 月 30 日」まで、リリースまでに必要な情報をページに載せていった。
- 「2010 年 10 月 16 日」～「2010 年 10 月 23 日」まで、役員によるページ編集のペースが緩やかになっている。この要因として以下の事情がある。
  - 「2010 年 10 月 9 日」～「2010 年 10 月 16 日」まで、役員は主に旧業務システムのページに記載されていた情報をそのまま転記するだけの作業を行っていた。
  - 「2010 年 10 月 16 日」～「2010 年 10 月 30 日」まで、役員は今度、開発者が作成したページの修正に乗り出した。前半の 1 週間で、どのように修正するのか当たりを付け、後半の 1 週間で修正に着手した。
- リリース後の「2010 年 11 月 13 日」～「2010 年 11 月 20 日」、役員によるページ編集のペースが急激に伸びている。これはリリース後でも良いとされたコーチのプロフィールを紹介する「コーチ紹介」ページの作成・編集を行なったことによる。このページは、<table>タグで囲まれた複雑なページであり、コーチが新しく入校するとか、コーチのプロフィールが変更になる度に、ページを編集しなければならない。旧業務システムにも同様のページがあるが、そのまま持ち込むと、<table>タグ内の HTML 要素を細かく修正しなければならず、ページを編集する度に少なからず労力を必要とする。このため、「2010 年 11 月 6 日」～「2010 年 11 月 13 日」の間、開発者と役員は、このページを楽に編集できるよう、共同で解決策を模索した。その結果、<table>タグで囲まれたページを簡単に編集できるプラグイン<sup>8</sup>を見つけた。そして、「2010 年 11 月 13 日」～「2010 年 11 月 20 日」の 1 週間で、このプラグインを利用して、運用者が「コーチ紹介」のページを仕上げた。
- 役員は、「2010 年 11 月 27 日」までに、スクール開校日である「2010 年 12 月 18 日」

---

<sup>8</sup> <http://wordpress.org/extend/plugins/wp-table-reloaded/>

前に必要な情報を掲載する作業を終えた。

### ③ 開発者（CAL ナビゲーター）の作業内容

一方、開発者のグラフの背景となる作業内容は以下の通りである。

- 開発者は、リリース日である「2010年10月30日」までに、旧業務システムの情報を運用者が作成・編集したページのHTMLタグの修正やページの階層構造の見直し（ページの最適化）を行う予定であった。特に、役員が作成・編集したページには無駄なHTMLタグが数多く含まれていたため、それらを次々と省いていった。その結果、ページの文字数は減少していった。ただし、その作業はリリース日までには間に合わず、「2010年11月6日」までずれ込んだ。
- 「2010年11月6日」～「2010年11月20日」まで、開発者は「会員申込」と「行事申込」という特殊なページを作成した。このページは、複数のHTMLコンポーネントを扱い、かつメール送信機能も付加する複雑なページであるため、役員ではなく開発者が作成・編集を担当した。この作業は、本来であれば、リリース日までに行わなければならないものであったが、他の作業に時間を割かなければならない事情もあり、着手・完成が1～2週間ほど遅れた。
- 「2010年11月20日」～「2010年11月27日」まで、再び文字数が減少しているが、これは業務システムのレイアウトを変更したことによって、再びページを最適化したためである。

### ④ ページの件数

以下は、ページを編集した延べ枚数を担当者毎に区切ったグラフである。

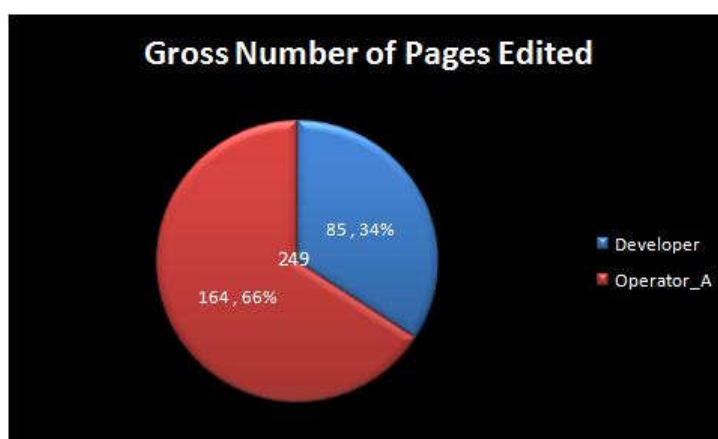


Figure 22 “Gross number of pages edited” (Original)



Figure 22 のグラフが示す通り、総ページ数は 249 枚となっているが、あくまでも編集を行なった延べ枚数であり、実際のページ数は約 1/3 である。

次に、編集したページの述べ枚数を時系列で示したグラフを示す。

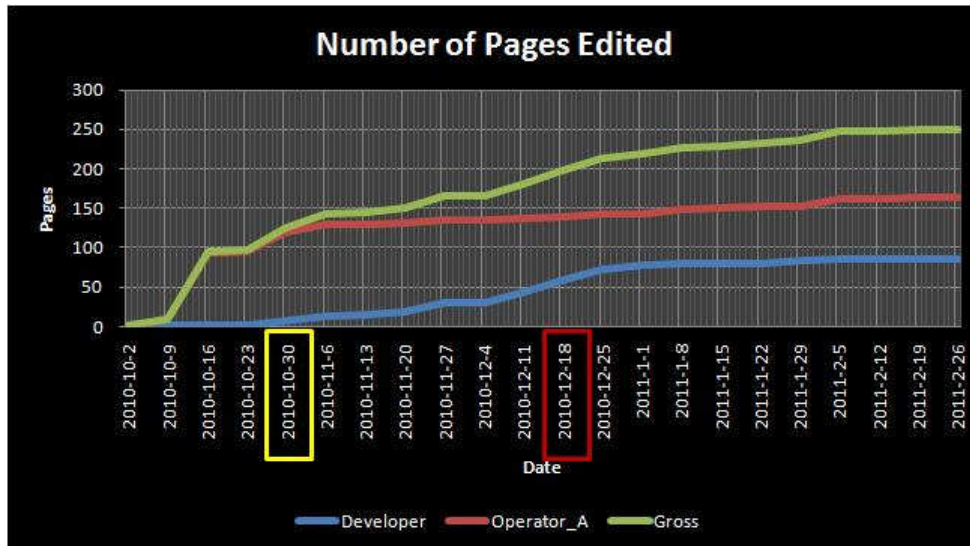


Figure 23 “Number of pages edited” (Original)

Figure 23 のグラフの「2010年10月30日」と「2010年12月18日」を、それぞれ黄色と赤色の枠で囲っている意味は、Figure 21 のグラフと同様である。

## 4.5. アンケート・ヒアリング

### 4.5.1. アンケートの目的・概要

実験結果を検証するために、役員や業務システムの利用者からアンケートを取った。

役員からアンケートを取る目的は、今回の開発と前回とを比べて、業務システムの運用状況がどのように変化したのかを調査するためである。役員に対して行った質問事項は、以下の5項目である。

- ・ 学んだこと・勉強になったこと (Learn)
- ・ 楽しかったこと・嬉しかったこと (Fun)
- ・ 今後も続けていきたいこと (Keep)
- ・ 今後、改善していきたいこと (Problem)
- ・ 今後、チャレンジしてみたいこと (Try)

これは、Agile 開発手法を採用している情報システム開発現場において、各開発者が「ふりかえり」をする時に問われる KPT (Keep, Problem, Try) の3項目に、役員が如何に能動的

に取り組んだのかを調べるための項目を追加した内容となっている。

一方、利用者からアンケートを取る目的は、今回の開発と前回とを比べて、業務システムのユーザビリティや利用者の満足度がどのように変化したのかを調査するためである。アンケートに使用した項目は、事前研究で使用された質問事項<sup>[64]</sup>をアレンジして持ち込んだものである。

ともに回答数は少なく、分析するには不十分なデータ量であるが、逆に貴重なものでもあるので、集計せずに全回答を掲載する。

#### 4.5.2. 初回リリース後の役員（CAL ドライバー）からのアンケート結果

##### (1) 学んだこと・勉強になったこと（Learn）

- ・ WordPress という新しいツールを学ぶことができた。

##### (2) 楽しかったこと・嬉しかったこと（Fun）

- ・ 今まで、原本が個人持ちされており、その個人しか更新することができなかった。それが、Web 上で原本を共有する仕組みに変わり、複数人間が更新できるようになった。
- ・ 原本が個人持ちでないため、どの環境（自宅、スクール、その他）でもログオンさえすれば更新ができるようになった。
- ・ ページの追加や更新が容易である。また、構成（どのページ配下にする、順番をどうする等）が簡単に換えられるようになった。

##### (3) 今後も続けていきたいこと（Keep）

- ・ 回答なし。

##### (4) 今後、改善していきたいこと（Problem）

- ・ 回答なし。

##### (5) 今後、チャレンジしてみたいこと（Try）

- ・ 回答なし。

#### 4.5.3. 最終リリース後の役員（CAL ドライバー）からのアンケート結果

##### (1) 学んだこと・勉強になったこと（Learn）

- ・ WordPress というツールの使い方を学んだ。
- ・ ローカルで更新してアップするという従来の方式ではなく、Web 上で更新が可能な

仕組みがあるという事を知った。

(2) 楽しかったこと・嬉しかったこと (Fun)

- ・ 自分たちの手で HP を作成することができた。
- ・ 作成した HP について他の人が話題にしているのを聞くと嬉しい。

(3) 今後も続けていきたいこと (Keep)

- ・ ホームページの改善。
- ・ 更新し続ける (合格者、コーチ紹介、ブログ) べきページの更新。

(4) 今後、改善していきたいこと (Problem)

- ・ ブログの更新がなかなか行われていない。
- ・ レッソンの流れを写真入りで紹介したいができていない。

(5) 今後、チャレンジしてみたいこと (Try)

- ・ 回答なし。

#### 4.5.4. 役員 (CAL ドライバー) からのアンケート結果の分析

4.5.2 と 4.5.3 において採取したアンケート結果は、回答者が少ないため、ここから説得力のある結論を導き出すことは難しい。また、殆どの回答が予想できる若しくは当たり前の内容であった。しかし逆に言えば、システム開発の経験を積んだ開発者にとって当たり前のことが、開発経験のない役員にとっては新鮮であるということが分かった。

#### 4.5.5. 最終リリース後の利用者からのアンケート結果

最終リリース後、約1ヶ月間、業務システムの利用者からアンケートの回答を得られた。この目的は、本研究で提案した開発手法に則って開発した業務システムが、利用者の満足度とどのような関係にあるのかを調査するためである。以下の Table 21 は、その質問と結果である。

Table 21 “Web Site User Satisfaction Scorecard” (Original)

#	大分類	小分類	質問	平均	標準偏差
1	Information	Relevancy	以前のサイトよりも今のサイトの 方が、情報は的確である。	3.33	0.71
2		Accuracy	〃、情報に間違いがない。	3.22	0.67

3		Comprehensibility	//、情報が分かりやすい。	2.33	1.41
4		Comprehensiveness	//、情報の量は多い。	2.78	0.97
5		Ease of Use	//、使い勝手が良い。	2.13	1.13
6	Connec tion	Entry Guidance	//、トップページから知りたい情報に簡単に辿り着ける。	2.63	1.06
7		Structure	//、構成は明確である。	2.50	1.41
8		Hyperlink	//、(ハイパー)リンクのリンク先は何なのかが分かる。	3.00	1.07
9		Speed	//、表示は速い。	3.38	0.74
10	Layout	Layout	//、レイアウトは良い。	2.00	1.07

このアンケートは、“The conceptualization and empirical validation of web site user satisfaction”<sup>[65]</sup>において使用されているアンケートを元に、一部の質問を和訳して作成したものである。有効回答者数は9で、回答者は全員、以前の業務システムを利用したことがある人達である。回答は全て5段階評価で、回答とその値は以下の Table 22 のとおりである。

Table 22 “Answers of Web Site User Satisfaction Scorecard” (Original)

回答	回答値
全然そうは思わない	1
そうは思わない	2
変わらない	3
そう思う	4
強くそう思う	5

#### 4.5.6. 利用者からのアンケート結果の分析

有効回答者数が少ないため、誤差が大きいと考えられる。よって、このアンケートは参考程度に掲載しておき、結論を出すことは控える。

#### 4.5.7. ヒアリング

経営者や役員から、上記のアンケートとは別の機会に何度かヒアリングした結果、業務システムの開発を行う前と後とで経営に対する姿勢の変化を感じ取ることができた。

(2)項でも触れたとおり、開発者は当初、2010年の2月に経営者に対してCAL開発手法での開発を提案していた。既にその時点で、旧システムに対する変更の難しさを感じていたものの、新しいシステムを開発することに躊躇していた。理由はいくつかあるだろうが、おそらく経営者が、ソフトウェアという複雑で境界・際限が見えにくいものに対して、コントロールしきれないと感じていたことが最大の要因であろうと思われる。

しかし、半年後の2010年8月に開発者が再び開発を提案したところ、次の営業シーズンが迫っていて選択の余地がなかったために、経営者は翻って提案を受け入れた。

開発が始まると、経営者は役員と共に打合せには積極的に参加し、ユーザー要求も提示して、それらの優先順位は付けていただいた。しかしながら、設計から検証までの細かい作業が多いフェーズに参加することには消極的であった。

その後、最初のイテレーション（スプリント）が終わり、現行システムの運用が始まってから半年も経たない内に、経営者の中で、そのシステムに対する新たな要望がいくつも生じてきた。けれども、今度は以前と違って直ぐにシステムの改善計画を立てるようになった。理由を聞いてみると、消極的とはいえ今回のシステム開発に参加して当該システムの仕組みをある程度理解したおかげで、今後どうすべきか分かったからだと述べた。

つまり、CAL開発手法で自社の業務システムを開発したおかげで、それまで業務システムというものをコントロールできないと感じて、それを戦略的に活用するに至らなかった経営者でも、システムを戦略的に活用しようとするのが分かった。

## 5. 効果の検証 2（企業経営や業務の戦略）

### 5.1. はじめに

2章での現状分析の結果、自社の業務システムの仕組みを理解することは、そのまま自社の内的要因を把握することになり、それが戦略を策定するための重要なインプットになることが分かった。企業の経営者や業務システムの利用者・運用者が、自社の業務システムの仕組みを理解するためには、システムの開発を人任せにするよりも、自分自身も積極的・能動的に開発プロジェクトに参加して取り組んだ方が効果的であるということが見えてきた。具体的には、利用者にとって本当に必要な機能だけが実装されて、無駄な機能が実装されなくなることによって、システムの開発・保守・運用費を削減できる可能性が増えることや、運用者がシステムの利用・運用方法をプロジェクトの早い段階から学習する機会を得ることによって、リリース間近に慌てて学習しなくても済むようになり、運用者の負担を減らすことにも繋がると思われる。

しかしながら、現状分析の結果として見えてきたことは、全て推測に基づく仮説である。即ち、「経営戦略を策定するためには、業務システムの仕組みを理解することが重要である」ということや、「業務システムの仕組みを理解するためには、システムの開発・保守・運用に積極的・能動的に参加すべきである」といったことは、確かにその通りだと思われるが、定性的または定量的には検証されていない。

更に、4章でCAL開発手法をシステム開発に適用して検証してみたものの、たった1プロジェクトでしか実験していない上に、対象企業や業務システムの規模も小さいため、実験結果としては少し物足りなかった。しかし、そのようなプロジェクトでも開発し終えるのに4ヶ月近くかかってしまう。よって、実際の業務システムを開発するのにCAL開発手法を適用しても、更にせいぜい1~2個のプロジェクトしか実験できない。

という訳で、4章で行った検証を保管するためにも追加でアンケートを実施して、これらの推測・仮説を検証して、CAL開発手法の妥当性を確保する。

### 5.2. 目的

前節5.1で述べた推測・仮説を検証して妥当性を確保するために、アンケートを実施する。具体的には、以下の推測・仮説に対して、アンケートの結果を用いて検証を行う。

- ・ 経営戦略を策定するためには、業務システムの仕組みを理解することが重要である。
- ・ 業務システムの仕組みを理解するためには、システムの開発・保守・運用に積極的・

能動的に参加すべきである。

### 5.3. 実施

アンケートの実施は、株式会社マクロミル（英文社名：MACROMILL, INC.）に代行を依頼した。属性情報・質問内容・回答結果は、付録 9.2 に掲載したとおりである。対象者は、200 名超の企業経営者・CAL ドライバーと、同じく 200 名超の起業に興味がある 20 歳以上の男女である。

### 5.4. 検証

アンケートの回答結果を分析して、前節 5.1 で述べた推測・仮設の妥当性を検証する。

#### 5.4.1. 一般的な業務システムの開発・保守・運用の経験と、経営戦略の策定に対する積極性との関連性

まずは、中小企業の経営者が、戦略を策定すべき業務に対して、策定に積極的に関わるかのかどうかと、一般的な業務システムの開発・保守・運用を経験したことがあるのかどうかの関連性を調べた。

Table 23 “Relativity to the activeness or passiveness for the development of management strategies with the experience of the development and/or maintenance and/or operation of general business system(s)” (Original)

戦略を策定する 対象の業務	※1	回答数	戦略の策定に積極的			戦略の策定に消極的		
			※2	※3	合計	※4	※5	合計
経営	○	68	<b>77.9</b>	<b>11.8</b>	<b>89.7</b>	<b>7.4</b>	2.9	10.3
	×	95	67.4	9.5	76.8	5.3	<b>17.9</b>	<b>23.2</b>
営業・販売	○	68	<b>66.2</b>	<b>22.1</b>	<b>88.2</b>	4.4	7.4	11.8
	×	95	50.5	21.1	71.6	<b>10.5</b>	<b>17.9</b>	<b>28.4</b>
企画・マーケティング	○	68	<b>58.8</b>	<b>22.1</b>	<b>80.9</b>	8.8	10.3	19.1
	×	95	47.4	16.8	64.2	<b>10.5</b>	<b>25.3</b>	<b>35.8</b>
経理・会計	○	68	<b>50.0</b>	10.3	<b>60.3</b>	<b>26.5</b>	13.2	39.7
	×	95	33.7	<b>22.1</b>	55.8	22.1	<b>22.1</b>	<b>44.2</b>

総務・人事	○	68	<b>58.8</b>	11.8	<b>70.6</b>	<b>16.2</b>	13.2	29.4
	×	95	40.0	<b>18.9</b>	58.9	15.8	<b>25.3</b>	<b>41.1</b>
法務・知財	○	68	<b>42.6</b>	13.2	<b>55.9</b>	<b>25.0</b>	19.1	44.1
	×	95	25.3	<b>13.7</b>	38.9	17.9	<b>43.2</b>	<b>61.1</b>
物流・調達	○	68	<b>45.6</b>	<b>20.6</b>	<b>66.2</b>	<b>19.1</b>	14.7	33.8
	×	95	30.5	8.4	38.9	18.9	<b>42.1</b>	<b>61.1</b>
生産・製造	○	68	<b>45.6</b>	<b>20.6</b>	<b>66.2</b>	<b>16.2</b>	17.6	33.8
	×	95	23.2	6.3	29.5	12.6	<b>57.9</b>	<b>70.5</b>
IT	○	68	<b>60.3</b>	<b>13.2</b>	<b>73.5</b>	<b>17.6</b>	8.8	26.5
	×	95	24.2	12.6	36.8	11.6	<b>51.6</b>	<b>63.2</b>
研究	○	68	<b>47.1</b>	<b>14.7</b>	<b>61.8</b>	<b>14.7</b>	23.5	38.2
	×	95	14.7	8.4	23.2	10.5	<b>66.3</b>	<b>76.8</b>
その他専門職	○	68	<b>33.8</b>	<b>13.2</b>	<b>47.1</b>	<b>16.2</b>	36.8	52.9
	×	95	26.3	8.4	34.7	11.6	<b>53.7</b>	<b>65.3</b>
上記以外	○	68	<b>23.5</b>	<b>5.9</b>	<b>29.4</b>	<b>11.8</b>	58.8	70.6
	×	95	11.6	5.3	16.8	7.4	<b>75.8</b>	<b>83.2</b>
合計	○	68		<b>65.8</b>			34.2	
	×	95		45.5			<b>54.5</b>	

(凡例)

- ・ ※1 … 業務システムの開発・保守・運用の経験の有無。
  - ○ … いずれかの経験がある。
  - × … いずれも経験がない。
- ・ ※2 … プラニングから積極的に関わって意見を出す。
- ・ ※3 … ある程度任せながら自分の意見を入れる。
- ・ ※4 … 担当者に任せて最後に承認する。
- ・ ※5 … ビジネス戦略立案に関与しない。
- ・ (太字) … 各業務を、列毎に、※1の○(上段)と×(下段)同士で比較して、高い方の数値。

上掲の Table 23 をグラフで表すと以下のようなになる。



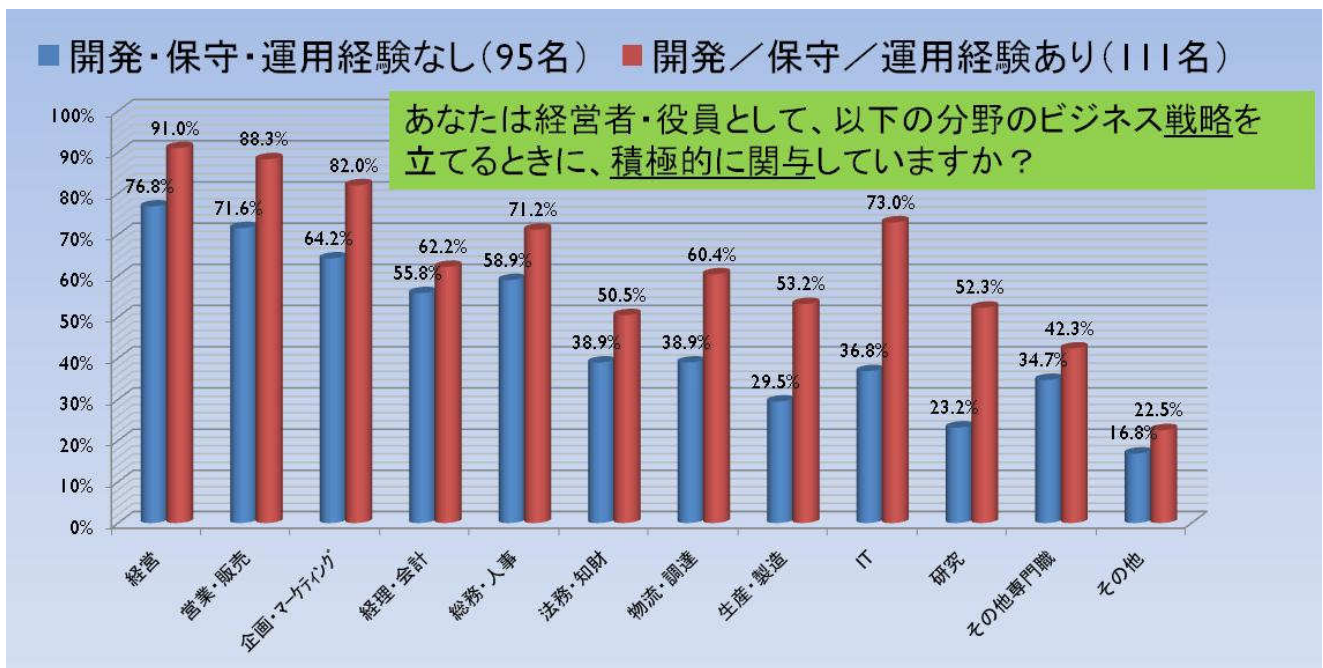


Figure 24 “Relativity to the activeness or passiveness for the development of management strategies with the experience of the development and/or maintenance and/or operation of general business system(s)” (Original)

上掲の Table 23 および Figure 24 が示すとおり、殆どのケースにおいて、一般的な業務システムの開発・保守・運用の経験と経営戦略の策定への積極性に、高い相関性が見えた。つまり、システムの開発・保守・運用の経験が全くない人に比べて、何らかの経験がある人の方が、戦略の策定に積極的に関与していることが分かった。

#### 5.4.2. 個々の業務システムの開発・保守・運用の経験と、経営戦略の策定に対する積極性との関連性

更に、個別の業務システムの開発・保守・運用の経験との関連性についても比較してみた。結果の表は行数が長いため付録の Table 53 に載せた。グラフも省くが、Table 23 と同様に、ここでも業務システムの開発・保守・運用を経験している人ほど、経営戦略の策定に積極的であることが分かった。

#### 5.4.3. 一般的な業務システムの開発・保守・運用の経験と、システムを知るためのアプローチとの関連性

次に、業務システムの仕組みを知ろうと思った時、どのようなアプローチを採用するの

かについて、業務システムの開発・保守・運用のいずれかで1年以上経験してきている人達と、いずれも全く未経験の人達とで比較した表を以下に示す。

Table 24 “Comparison of approaches to know mechanism of a business system by old-timer engineers and amateurs” (Original)

#	システムの仕組みを知るための アプローチ	重み	開発・保守・運用経験	
			なし(95名)	あり(111名)
1	知りたくない	0	9.5%	4.5%
2	詳しい人から話を聞く	5	55.8%	57.7%
3	ドキュメントを読む	10	13.2%	24.8%
4	デモンストレーションを見る	30	8.4%	21.6%
5	勉強会に参加する	50	13.7%	23.4%
6	運用に参加する	75	11.6%	23.9%
7	開発・保守に参加する	90	7.9%	27.5%
8	その他のアプローチを採用する	20	15.3%	8.6%
<b>学習効果</b>			<b>32.32</b>	<b>67.91</b>

上掲の Table 24 の最下段にある「学習効果」は以下のように算出している。

$$\text{「学習効果」} = \sum (\text{「システムの仕組みを知るためのアプローチ」} \times \text{「重み」})$$

ついでにこの表をグラフで表すと、以下のようになる。

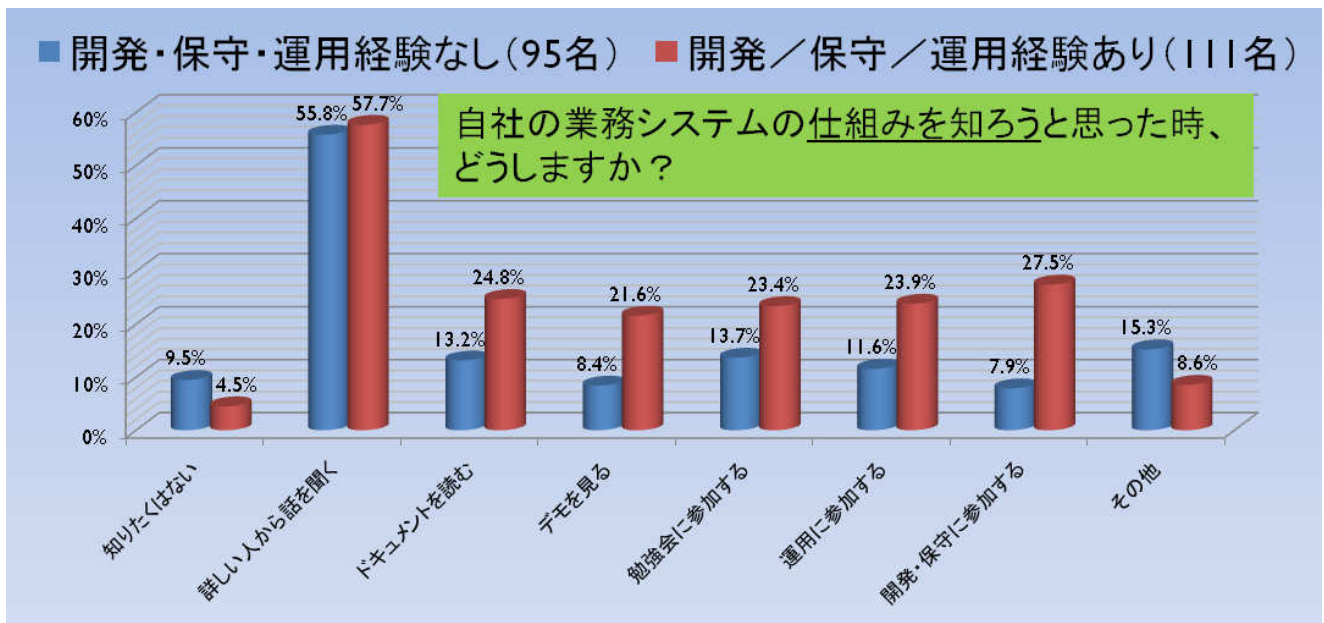


Figure 25 “Comparison of approaches to know mechanism of a business system by old-timer engineers and amateurs” (Original)

改めて、Table 24 を眺めると、システムの仕組みを知ろうとした時の、1年以上の経験者と未経験者の学習効果は、67.91 : 32.32 と約 2 : 1 の比率になる。このことから、システムの開発・保守・運用の経験が全くない人に比べて、何らかの経験がある人の方が、システムの仕組みを深く知ろうとして、より学習効果の高いアプローチを採用する傾向にあることが分かる。更にこれに加え、お互いが持っている事前の知識・経験・技量の差が、それぞれの学習効果の差を広げることになるはずである。

また、重み付けをする前の両者のアプローチの割合を比較すると、その値が大きく開いているのは、#6の「運用に参加する」と#7の「開発・保守に参加する」であり、いずれも能動的な学習アプローチであることは明白である。よって経験者がシステムの仕組みを知ろうとした時、未経験者と比べてシステムの開発・保守・運用に積極的に参加して、能動的に学習することを重視していることが分かる。

#### 5.4.4. 一般的な業務システムの開発・保守・運用の経験と顧客能動学習型（CAL）開発

最後に、顧客が自社の業務システムを開発する時、「プロの開発者のサポートを受けられるのであれば、自社のシステムの開発・保守をやってみたい」と思うかどうか、即ち、CAL 開発手法でもってシステム開発することに興味・関心があるかどうかを尋ねてみた。以下の表にその結果を示す。

Table 25 “Interests in the application of CAL system development Methodology on their own business system being developed?” (Original)

興味・関心	開発・保守・運用経験	
	なし(95名)	あり(111名)
やってみたいと思わない	62.6%	20.7%
自分を除く社内の皆でやってもらいたい	4.2%	6.8%
自分一人でやってみたい	10.5%	22.5%
自分を含む社内の皆でやってみたい	22.6%	50.0%

上掲の表をグラフにすると以下のようなになる。

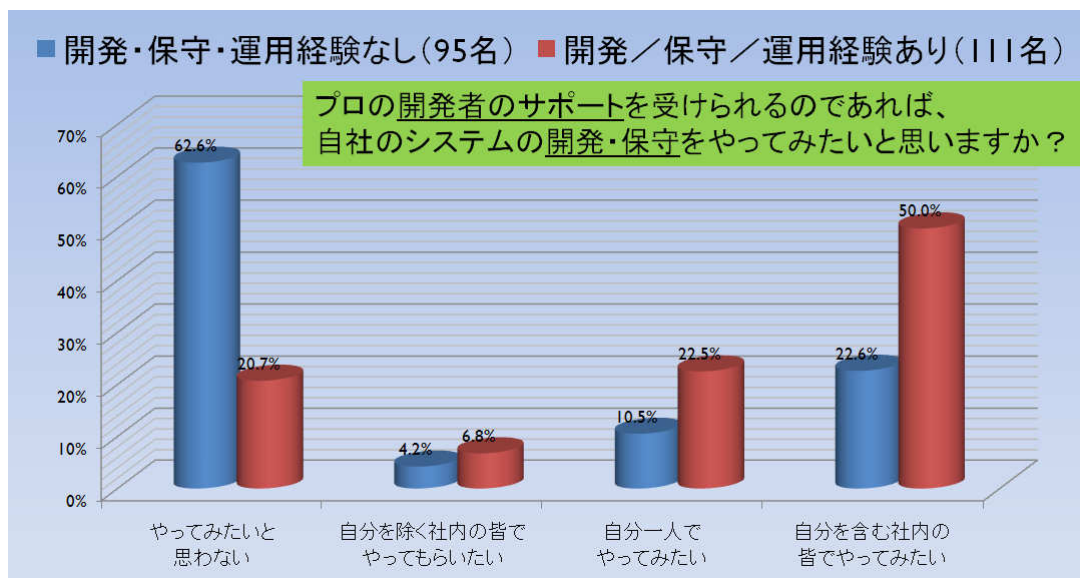


Figure 26 “Interests in the application of CAL system development Methodology on their own business system being developed?” (Original)

以上の表・グラフを見ると、業務システムの開発・保守・運用の経験が全くない人達の大半は、たとえサポートを受けられたとしても、自社の業務システムの開発・保守には消極的であることが分かる。対照的に、そういった経験がある人達の大半は積極的であることが分かる。経験者といえども誰もが最初は未経験であることは自明であるから、やはり未経験者には経験を積んでもらうことによって、次第にシステム開発・保守に積極的になり、それが自社の企業経営や業務の戦略策定にも積極的に関わるようになることが予想できる。

## 5.5. 考察

改めて、以下の推測・仮説に対する検証を振り返ってみると、

- ・ 企業経営や業務の戦略を策定するためには、業務システムの仕組みを理解することが重要である。
- ・ 業務システムの仕組みを理解するためには、システムの開発・保守・運用に積極的・能動的に参加すべきである。

1つ目の推測・仮説は、5項と5.4.2項で明らかになったことによって裏付けられたと考えられる。また2つ目は、5.4.3項と5.4.4項で分かったことによって裏付けられたとみなして差し支えない。この結果、3章で提示したモデルの一部である Figure 14 の妥当性を確認・証明できたと考える。すなわち、システムの開発・保守・運用のいずれかに積極的に参加することによって、システムの仕組みを詳しく知ることができ、その結果、企業経営や業務の戦略策定に積極的に関与できるようになる、という一連の因果関係の妥当性を説明できた。

## 6. 結論および今後の課題と展望

### 6.1. 結論

今回の研究では、顧客能動学習型（CAL: Customers' Active Learning）開発という新しいシステム開発手法を提案し、それを実際のシステム開発に適用することで、有効性を検証した。その結果、CAL 開発手法は、開発者の実装の作業工数を減らすことに繋がる事が分かる。今回のケースで言えば、ページのコンテンツの9割以上を顧客が実装したことにより、その分、開発者は一から十まで実装しなくて済んだ。それに対して顧客は、開発初期から能動的に開発に関わることによって、システムを運用する時、受動的・消極的ではなく能動的・積極的に取り組める可能性があることが分かる。このことは、運用者の心理的な負担を下げることも言える。それから、システムのリリース間近になって、開発者が顧客にシステムの運用方法を伝える必要がなくなる。今回の実験では、顧客が自分で何度もページを作成・編集を繰り返す内に、システムの運用の仕方も自然と学習していき、その結果、システムがリリースされる頃には、顧客はシステムの運用の仕方を殆どマスターしてしまい、学習する必要が殆どなかった。これによって、システムをどのように運用すれば良いのか？という不安、即ち心理的な負担を抑えることにも繋がる。そして、4.5.7 項でも述べたとおり、システムをコントロールできないでいた経営者または経営陣は、CAL 開発手法を経験すると、システムを戦略的に活用しようと前向きになることも分かった。

一方、200 名超の中小企業経営者に対するアンケートの回答を分析した結果、以下の 3 点の気づきを得た。

- ・ システムの開発・保守・運用の経験が全くない人に比べて、何らかの経験がある人の方が、戦略の策定に積極的に関与している。
- ・ システムの開発・保守・運用の経験が全くない人に比べて、何らかの経験がある人の方が、システムの仕組みを深く知るために学習効果の高いアプローチ（例：システムの開発・保守・運用への参加）を採用する傾向にある。
- ・ 未経験者には経験を積んでもらうことによって、次第にシステム開発・保守に積極的になり、それが自社の企業経営や業務の戦略策定にも積極的に関わるようになる。

これらは、3.2 節の Figure 14 で示した因果関係図の正確性を証明しており、CAL 開発手法の妥当性を確保することに繋がっている。

以上、CAL 開発手法は、開発者および顧客を様々な負担から解放し、運用現場から疲弊感を取り除くと共に、経営者または経営陣にシステムの戦略的な活用を促し、ひいては組織の活性化に繋がる開発手法であることが分かった。

## 6.2. 課題

### 6.2.1. 見積り

どのような開発手法であっても、システムを開発する時、作業工数を見積もることによって、作業計画を立てるとか、作業単価を算出することができる。

以下、作業工数の見積方法の一覧である。<sup>[66]</sup>

- Cost Estimation, Benchmarking, and Risk Assessment (CoBRA)
- Constructive Cost Model (CoCoMo)
- Function Point (FP)
- Ideal Engineering Hours (IEH)

本来なら、この中から今回の実験に適した方法を選んだ後、開発前に全体の作業工数を見積り、最終リリース後、実際の全体作業工数を測定し、比較・分析するつもりでいた。

しかしながら、以下の理由で工数見積りを断念した。

まず、CoBRA 法、CoCoMo 法、FP 法は、いずれも入力すべき項目が多く、更にいくつかの制約のために、その入力項目を導き出すことが困難または不可能であったため、見積りを先送りにした。果たして、これらのお見積方法が Agile 開発手法に馴染むのかどうかについて更に調査する必要があると感じた。

IEH 法は、Agile 開発手法で推奨されている方法ではあるが、今回は以下の理由で見積もりを先送りにした。

- 顧客が提示したストーリーは割りと明確であったが、イテレーションが最大で 30 日という短い期間で、しかも実際には平日は、開発者および顧客の都合で、半日（4 時間）すら作業できなかったため、それを工数に算出するのは時間的に厳しかった。
- 顧客が、果たしてどれぐらいのペースで作業を進めることができるのか、全く予想がつかなかった。

特に 2 つ目の理由として挙げた、顧客の作業工数を見積もれないことは、開発を計画通りに進めることを困難にしてしまう。今後、この問題をどのように解決するかは十分に検討していかなければならない。

### 6.2.2. 契約

当初は、前の 6 項で見積もった作業工数を元に、CAL 開発手法でシステムを開発する時の

契約金額を算出するつもりでいた。以下は、契約金額の算出方法の一覧である。<sup>[67][68][69][70]</sup>

- the “Sprint Contract”
- Fixed Price / Fixed Scope
- Time and Materials
- Time and Materials with Fixed Scope and a Cost Ceiling
- Time and Materials with Variable Scope and Cost Ceiling
- Phased Development
- Bonus / Penalty Clauses
- Fixed Profit
- “Money for Nothing, Changes for Free”
- Joint Ventures
- Collaborative Agile Contracts

最終的にどの形態の契約を結ぶかは、顧客側と開発側の利害が一致するものになるが、そもそも作業工数を見積もれなかったのが、具体的な契約金額の算出を断念した。

一方で、今までの算出方法と変わってくるはずであるとも考えている。従来の開発手法に則って開発するのであれば、「これだけやるから、これだけください。」ということになる。しかし、今回の CAL 開発手法なら、「教えてあげるから、これだけください。」という風になるはずである。即ち、発注者側にとって、開発費の代わりに教育費が発生することになる。この教育費というものが、開発者側の新たな収入源になると考えられる。

### 6.2.3. その他

顧客は情報システム開発に関する専門的な知識を殆ど有していないため、たとえ能動的に学習しても、できる範囲は開発者と比べると限りがある。そのため、例えば、開発者が顧客に「砂場」の様なものを提供して、その中で自由に開発してもらい環境を整えたりして、顧客が楽しく開発できるように導くことが重要である。

また、顧客が開発したシステムはユーザビリティなどの標準から外れていることも少なくない。そこは開発者がフォローに回って修正を手助けしてやるなど、顧客をサポートすることも重要であると考えられる。

それから、そもそも顧客がシステムの開発に関わることは監査上よろしくないという意見もある。勿論、全てが駄目とは言えないだろうし、今回の実験では、特に監査的に問題



はなかったので、果たしてどこまでが OK で、どこからが NG なのかを調査する必要がある。

### 6.3. 展望

今後も世の中のシステム開発のスピードは増々速くなっていくと考えられるが、システムを受入れる企業側もそのスピードに応じていかなければならない。そこで重要になってくるのが、お互いが相手のやっていることを素早く理解することである。CAL 開発手法は、システム開発の発注側である顧客企業の経営者または経営陣が、システムの仕組みを深く理解することによって、システム運用時の心理的な負荷の軽減を狙ったものであるが、それだけではなく、システムに精通することにより脳の中でゲシュタルトが構築され、それが経営戦略に活かされる効果も期待している。

また、開発者も顧客に教えるだけでなく、金銭以外の何らかの知的な見返りを得ることができれば、真に Win-Win の関係になりえる訳で、そうしたことも実現できるよう、このモデルの更なる改良を重ねていきたい。

## 7. 謝辞

本研究の主査として、貴重なお時間を割いてレビューしていただいた慶應義塾大学大学院システムデザイン・マネジメント研究科の当麻哲哉准教授に深く感謝申し上げます。情報システム開発の視点しか持ち得なかった私に対して、ビジネスや教育の視点から本研究の価値を見出していただき、研究の幅を広げることができました。また、決して相手を否定しない、その代わり相手にも「できない」を言わせない独特の指導方法のおかげで、限られた時間の中、モチベーションを保ち続けて研究に取り組むことができました。心より感謝しております。更に、私の体調についてもご心配かけていただき、負担がかからないように配慮していただけたことに、御礼申し上げます。

同研究科の小木哲朗教授、そして神武直彦准教授には、副査を務めていただき、ご指導いただけたことに深く感謝申し上げます。小木教授には、私の研究の内容を理解しようと辛抱強くレビューしていただき、かつ論文のストーリーの重要性を何度も繰り返しご指摘くださったことが、非常に印象に残っております。神武准教授には、システム開発手法に関する論文の一般的な書き方をご教示いただき、また論文のコアとなる部分を充実させることを強く推奨していただけたことを、とても有難く思っております。

同研究科の前研究科委員長である狼嘉彰教授、そして中野冠教授には、以前、副査を務めていただけたことを今でも感謝しております。狼教授には、JAXAにおけるWebサイト開発時のエピソードを取り上げて、本研究の成果と比較して研究の価値を認めていただき、自信を深めることができました。また、中野教授には、本研究の目的と成果を厳しくレビューしていただき、おかげ様で研究のレベルを向上させることができたことと確信しました。

苗場スキースクール校長の園部健氏、副校長の園部修氏、主任の近藤洋氏には、本研究の実験のために同スクールの業務システムの開発および保守を私に委任していただき、心より感謝しております。特に園部健氏と近藤氏には、隔週のミーティングに参加していただき、また他のスクールのスタッフを巻き込んで開発に関する意見をヒアリングしていただき、ご尽力いただいたことに深く御礼申し上げます。園部修氏、スタッフの鎗田明子氏からは、開発に関して重要な意見を述べていただきました。更に、近藤洋氏は、運用者として、時には共同開発者として、当該システムの開発・保守・運用に関わっていただき、非常に貴重な実験データを得ることができました。当初、業務システムの開発の仕方を殆ど知らない状態から、開発方法を能動的に学習していただいた結果、私の予想を遥かに上回るペースで開発が進んだことは、驚きを禁じえませんでした。

前任開発者の山本博之氏には、ご多忙を極める中、ご自身の開発作業を振り返っていただき、そのおかげで、今回の開発と比較・検証を行うのに大いに助かりました。誠に有難うございます。

更に、株式会社マクロミルの成澤知美および三浦志保には、短期間でのアンケートの実施および回答結果の収集をしていただき大変助かりました。また平日の遅い時間帯や土日休日にもご対応していただき、非常に有り難く思っております。

また、同研究科の岡啓氏、株式会社永和システムマネジメントの鍛冶舎浩氏、中西卓哉氏および北川朋央氏からは、豊富なシステム開発の知見や経験を元に様々なアドバイスをいただけたおかげで、研究にリアリティを持たせることができたと思っております。また、研究の進め方にも助言や時には苦言を呈していただき、誠に有り難うございました。

最後に、仕事が忙しいにも関わらず私が研究のために時間を割くことを承認していただいた、株式会社シンプレクス・コンサルティングの服部省治氏や菊池俊輔氏、高野一利氏をはじめ、昔または現在の上司である、伊東詩音氏、飛澤宗則氏、木村耕司氏、中野明彦氏、宮川悠氏、住田昌樹氏、山下幸男氏、辻昌佳氏、木村正弘氏、そしてその他の同僚の皆さまには、私のことを暖かく見守っていただいたことに深く感謝申し上げますとともに、その分、仕事に時間を費やすことができなかつたことを深くお詫び申し上げます。

## 8. 文献目録

1. **CoffinRod , LaneDerek.** 7つのアジャイル開発手法の実践ガイド（第1回）. internet.com. (オンライン) 2007年2月6日. <http://japan.internet.com/developer/20070206/27.html>.
2. -. 7つのアジャイル開発手法の実践ガイド（第2回）. internet.com. (オンライン) 2007年3月20日. <http://japan.internet.com/developer/20070320/26.html>.
3. **IPA.** 非ウォーターフォール型開発 WG 活動報告書. 日本 : IPA, 2011. ページ: I .
4. **Agile Alliance.** Manifesto for Agile Software Development. Manifesto for Agile Software Development. (オンライン) 2001年. <http://www.agilemanifesto.org/>.
5. -. Principles behind the Agile Manifesto. Manifesto for Agile Software Development. (オンライン) 2001年. <http://www.agilemanifesto.org/iso/en/principles.html>.
6. **IPA.** 非ウォーターフォール型開発 WG 活動報告書. 日本 : IPA, 2011. ページ: 75.
7. **FowlerMartin.** オンサイト顧客. Martin Fowler's Bliki. (オンライン) 2003年12月2日. <http://capsctrl.que.jp/kdmsnr/wiki/bliki/?OnsiteCustomer>.
8. **小野和俊.** そして、ペア・プログラミングが始まる. 小野和俊のブログ. (オンライン) 2007年7月4日. <http://blog.livedoor.jp/lalha/archives/50172573.html>.
9. **日経ビジネス.** 【会社の寿命】今や"寿命"はわずか5年. 日経ビジネス ONLINE. (オンライン) 2009年2月18日. (引用日: 2011年7月27日.) <http://business.nikkeibp.co.jp/article/manage/20090212/185916/>.
10. **新宅純二郎.** 経営戦略とは. UT オープンコースウェア. (オンライン) 2006年. [http://ocw.u-tokyo.ac.jp/wp-content/uploads/lecture-notes/Eco\\_07/shintaku-01.pdf](http://ocw.u-tokyo.ac.jp/wp-content/uploads/lecture-notes/Eco_07/shintaku-01.pdf).
11. **KluyverA. deCornelis , PearceA.John.** Strategy: A View From The Top. 出版地不明 : Prentice Hall, 2003.
12. **BarneyJay.** Gaining and Sustaining Competitive Advantage. 出版地不明 : Prentice Hall, 2010.
13. **浅羽茂.** 経営戦略の経済学. 出版地不明 : 日本評論社, 2004.
14. **伊丹敬之 , 加護野忠男.** ゼミナール 経営学入門. 出版地不明 : 日本経済新聞社, 2003.
15. **伊丹敬之.** 経営戦略の論理. 出版地不明 : 日本経済新聞社, 2003.
16. **一橋大学商学部経営学部門, (編).** 経営学概論. 出版地不明 : 税務経理協会, 1999.
17. **LambBoydenRobert.** Competitive Strategic Management. 出版地不明 : Prentice Hall, 1984.

18. **大前研一**. 戦略論. 出版地不明 : ダイヤモンド社, 2007.
19. **BarrP.C. , CallenderD.E.** Life-cycle Management & Documentation Concepts for the Space Station Program. USA : IEEE, 1988. ページ: 33.
20. **JUAS**. 企業 IT 動向調査 2008. 日本 : 日本情報システム・ユーザー協会, 2008. ページ: 159. V 字型開発から U 字型開発へ.
21. -. 企業 IT 動向調査 2011. 日本 : 社団法人日本情報システム・ユーザー協会, 2011. ページ: 46. 10 年度調査.
22. -. 企業 IT 動向調査 2010. 日本 : 社団法人日本情報システム・ユーザー協会, 2010. ページ: 76. 09 年度調査.
23. -. 企業 IT 動向調査 2009. 日本 : (社) 日本情報システム・ユーザー協会, 2009. ページ: 87. 08 年度調査.
24. -. 企業 IT 動向調査 2008. 日本 : (社) 日本情報システム・ユーザー協会, 2008. ページ: 59. 07 年度調査.
25. -. 企業 IT 動向調査 2007. 日本 : (社) 日本情報システム・ユーザー協会, 2007. ページ: 61. 06 年度調査.
26. -. 企業 IT 動向調査 2006. 日本 : (社) 日本情報システム・ユーザー協会 (JUAS), 2006. 05 年度調査.
27. -. 企業 IT 動向調査 2010. 日本 : 社団法人日本情報システム・ユーザー協会, 2010. ページ: 78. 09 年度調査.
28. -. 企業 IT 動向調査 2008. 日本 : (社) 日本情報システム・ユーザー協会, 2008. ページ: 65. 07 年度調査.
29. -. 企業 IT 動向調査 2006. 日本 : (社) 日本情報システム・ユーザー協会 (JUAS), 2006. ページ: 24. 05 年度調査.
30. -. 企業 IT 動向調査 2010. 日本 : 社団法人日本情報システム・ユーザー協会, 2010. ページ: 52. 09 年度調査.
31. **熊澤壽**. IT 部門と中間管理職の事なかれ主義が日本企業を駄目になっている. 日経情報ストラテジー . (オンライン) 2011 年 1 月 27 日 .  
<http://itpro.nikkeibp.co.jp/article/COLUMN/20101224/355651/>.
32. **JohnsonJim**. Chaos Report. USA : The Standish Group International Inc., 2002.
33. **ClancyTom**. Chaos Report. USA : Stadish Group, 1995. ページ: 8.

34. **Standish Group**. Extreme Chaos. USA : THE STANDISH GROUP INTERNATIONAL, INC., 2001. ページ: 4.
35. **JovanovicDejan , DogGaTomai**. Comparison of software development models and their usage in Computer-Telephony systems. USA : Conference on Telecommunications, 2003.
36. **LarmanCraig , BasiliR.Victor**. Iterative and Incremental Development: A Brief History. USA : IEEE, 2003.
37. **MunassarMohammed AliNabil , GovardhanA**. A Comparison Between Five Models Of Software Engineering. USA : IJCSI, 2010.
38. **IPA**. 非ウォーターフォール型開発 WG 活動報告書. 東京 : IPA, 2011. ページ: 10-11.
39. **三橋和広**. 分散開発のABC(前編). @IT. (オンライン) 株式会社オープントーン, 2005年7月20日. (引用日: 2011年8月13日.) <http://www.atmarkit.co.jp/farc/rensai/abc01/abc01a.html>.
40. **安保秀雄**. やはり, ソフト開発手法の確立を組み込み技術者に期待する. 日経エレクトロニクス. (オンライン) 日経 BP, 2009年3月3日. (引用日: 2011年8月14日.) <http://techon.nikkeibp.co.jp/article/TOPCOL/20090302/166587/>.
41. **AmblerW.Scott , KrollPer**. Best practices for lean development governance, Part II: Processes and measures. IBM developerWorks Rational Technical Library. (オンライン) IBM. (引用日: 2011年7月22日.) [http://www.ibm.com/developerworks/rational/library/jul07/kroll\\_ambler/](http://www.ibm.com/developerworks/rational/library/jul07/kroll_ambler/).
42. **IBM**. Rational Unified Process (RUP). IBM. (オンライン) 2010年11月14日. <http://www-01.ibm.com/software/awdtools/rup/index.html>.
43. **PortierBertrand , HodgkinsonGregory**. Exploring the fundamentals of architecture and services in an SOA. IBM developerWorks. (オンライン) 2007年9月4日. <http://www.ibm.com/developerworks/architecture/library/ar-archserv1/>.
44. **WessberMats**. Introducing the IBM Rational Unified Process essentials by analogy. IBM. (オンライン) 2005年10月12日. <http://www.ibm.com/developerworks/rational/library/05/wessberg/>.
45. **IPA**. 非ウォーターフォール型開発 WG 活動報告書. 日本 : IPA, 2011. ページ: 8.
46. **TDD.NET**. ソフトウェアの新たな開発手法、「アジャイル開発」って? TDD.NET. (オンライン) 2010年1月12日. <http://www.tdd-net.jp/2010/01/zdnet-japan-8d8.html>.
47. The Agile Unified Process (AUP) - Overview. Ambysoft. (オンライン) 2005年. <http://www.ambysoft.com/unifiedprocess/agileUP.html#Overview>.
48. The Agile Unified Process (AUP) - Serial in the large. Ambysoft. (オンライン) 2005年.

<http://www.ambysoft.com/unifiedprocess/agileUP.html#Serial>.

49. The Agile Unified Process (AUP) - Iterative in the Small. Ambyssoft. (オンライン) 2005 年.  
<http://www.ambysoft.com/unifiedprocess/agileUP.html#Iterative>.

50. **Scrum Alliance**. Scrum Is an Innovative Approach to Getting Work Done. Scrum Alliance. (オンライン) 2011 年 2 月 17 日. [http://www.scrumalliance.org/pages/what\\_is\\_scrum](http://www.scrumalliance.org/pages/what_is_scrum).

51. Manifesto for Agile Software Development. Manifesto for Agile Software Development. (オンライン) 2001 年. <http://www.agilemanifesto.org/>.

52. **井上英明** , **森山徹**. 要件未定でも納期は厳守 “アジャイル開発”で乗り切る. ITpro. (オンライン) 2009 年 7 月 15 日. <http://itpro.nikkeibp.co.jp/article/COLUMN/20090701/333041/>.

53. **富本達明**, **ほか**. 小規模・短納期プロジェクトに X P を導入する際の課題. 出版地不明 : JUSE, 2002. ページ: 6.

54. アジャイルはなぜ失敗するのか? ~教科書には載っていない反復型開発の 3 つの掟. プログラマの思索. (オンライン) 2010 年 10 月 9 日 .  
<http://forza.cocolog-nifty.com/blog/2010/10/3-9be5.html>.

55. **TucciJoe**. Keynotes - EMC World 2010. USA : EMC Corporation, 2010.

56. **Sabre Airline Solutions**. How We Made Onsite Customer Work - An Extreme Success Story. 出版地不明 : IEEE, 2007.

57. **TorpLinda** , **SageSara**. Problems as Possibilities: Problem-Based Learning for K-16 Education. 出版地不明 : Association for Supervision & Curriculum Dev, 2002.

58. **富山高等専門学校**. 他の教授方法との比較. 富山高等専門学校. (オンライン) (引用日: 2011 年 7 月 17 日.) [http://www.pbl.toyama-nct.ac.jp/?page\\_id=21](http://www.pbl.toyama-nct.ac.jp/?page_id=21).

59. **富山工業高等専門学校**. 参考文献一覧. Problem-Based Learning. (オンライン) (引用日: 2011 年 7 月 20 日.) [http://www.pbl.toyama-nct.ac.jp/?page\\_id=15](http://www.pbl.toyama-nct.ac.jp/?page_id=15).

60. **井庭崇**. 学習パターンとは. Learning Patterns. (オンライン) 慶應義塾大学. (引用日: 2011 年 7 月 17 日.) <http://learningpatterns.sfc.keio.ac.jp/introduction.html>.

61. **大森久永**. 目的と目標を混同するな. IT Pro. (オンライン) 日経 BP, 2011 年 3 月 30 日. (引用日: 2011 年 8 月 14 日.) <http://itpro.nikkeibp.co.jp/article/COLUMN/20110322/358537/>.

62. **LarmanCraig**. Agile and Iterative Development: A Manager's Guide. 出版地不明 : Addison-Wesley Professional, 2003.

63. **井庭崇**. Learning Patterns. 『Learning Patterns』 (SFC オリジナル版) . (オンライン) (引

用日: 2011年7月17日.) <http://learningpatterns.sfc.keio.ac.jp/LearningPatterns2009.pdf>.

64. **MuyllleSteve, MoenaertRudy , DespontinMarc.** The conceptualization and empirical validation of web site user satisfaction. Elsevier B.V. 2003.

65. **MuylleaSteve, MoenaertbRudy , DespontincMarc.** The conceptualization and empirical validation of web site user satisfaction. 出版地不明 : Elsevier B.V., 2003.

66. **澤部直太.** ソフトウェア開発におけるコスト見積もり手法. アジア人財資金構想プログラム産学連携専門講義 . (オンライン) <http://www.tuat.ac.jp/~asiaprogram/courses/project/lesson09/#sec4>.

67. **SimsChris.** Contracts for Agile Software Development. InfoQ. (オンライン) 2009年5月4日. <http://www.infoq.com/news/2009/05/Agile-Contracts>.

68. **StevensPeter.** 10 Contracts for your next Agile Software Project. AgileSoftwareDevelopment.com. (オンライン) 2009年4月29日. <http://agilesoftwaredevelopment.com/blog/peterstev/10-agile-contracts>.

69. **ThorupLars , JensenBent.** Collaborative Agile Contracts. BestBrains. 出版地不明 : Denmark.

70. -. アジャイル開発に適した契約 ~ ハイブリッド契約の例. TDD.NET. (オンライン) 2010年5月6日. <http://www.tdd-net.jp/2010/05/bent-jensen-e04.html>.

71. **波田野裕一.** 現場視点からの運用方法論. Think IT. (オンライン) 2010年12月2日. <http://thinkit.co.jp/story/2010/12/02/1903>.

72. An Expansive View of Reusable Software. **HorowitzEllis , MunsonB.John.** 5, USA : IEEE, 1984年, IEEE Transactions on Software Engineering, 第 SE-10 巻, ページ: 479.

73. **Standish Group.** 2006 FIRST QUARTER RESEARCH REPORT. USA : The Standish Group International Inc., 2006.



## 9. 付録

### 9.1. 図表目次

#### 9.1.1. 図目次

Figure 1 “Strategies Hierarchy” (Based on [10]) .....	13
Figure 2 “Information System Lifecycle” (Based on [19][20]) .....	16
Figure 3 “IT cost amount per 1 company” (Based on [21][22][23][24][25][26]).....	17
Figure 4 “IT cost percentage per 1 company” (Based on [21][22][23][24][25][26]).....	17
Figure 5 “Features / Functions used in a typical system” (Based on [32]) .....	19
Figure 6 “Based on frequent demonstrations and stakeholder feedback, the small course corrections early in the lifecycle of an iterative ("modern") project lead to more rapid success, compared to a "waterfall" project.” <sup>[41]</sup> .....	22
Figure 7 “The Rational Unified Process” <sup>[43]</sup> .....	24
Figure 8 “The RUP's iterative development process” <sup>[44]</sup> .....	24
Figure 9 “Cockburn Scale” <sup>[2]</sup> .....	28
Figure 10 “Table which shows evaluations based on experiences and observations of authors to the conformity of each development methodology with Cockburn's scale” <sup>[2]</sup> .....	29
Figure 11 “The Agile Unified Process (AUP) lifecycle” <sup>[47]</sup> .....	30
Figure 12 “Learning Pyramid” <sup>[59]</sup> .....	36
Figure 13 “Comparison of a traditional system development methodology and CAL” .....	38
Figure 14 “Causal Relationship Diagram – Strategic Management” (Original).....	39
Figure 15 “General roles in a system development” (Original).....	46
Figure 16 “Roles in XP Development” (Original) .....	47
Figure 17 “Roles in CAL Development” (Original) .....	48
Figure 18 “Comparison with the old site and the new site” (Original).....	59
Figure 19 “Flow chart of works to extract data of web pages and summarize them” (Original) .....	60
Figure 20 “Total number of characters in all pages” (Original).....	61
Figure 21 “Number of characters added or deleted for pages” (Original) .....	61
Figure 22 “Gross number of pages edited” (Original) .....	63
Figure 23 “Number of pages edited” (Original).....	64

Figure 24 “Relativity to the activeness or passiveness for the development of management strategies with the experience of the development and/or maintenance and/or operation of general business system(s)” (Original) .....	72
Figure 25 “Comparison of approaches to know mechanism of a business system by old-timer engineers and amateurs” (Original).....	74
Figure 26 “Interests in the application of CAL system development Methodology on their own business system being developed?” (Original).....	75
Figure 27 “Sprint Backlog” (Original).....	96

### 9.1.2. 表目次

Table 1 “Transition and composition of maintenance & operation costs” (Based on [27][28][29]) .....	18
Table 2 “Breakdown of maintenance and operation costs” (Based on [30]).....	18
Table 3 “Success factors of projects at 1995” (Based on [33]).....	19
Table 4 “Success factors of projects at 2001” (Based on [34]).....	20
Table 5 “Comparision of 4 Software Development Methodologies in target areas” (Based on [38][39][40]) .....	21
Table 6 “12 principles behind the Agile Manifesto” (Based on [5]) .....	25
Table 7 “Comparison of Agile Software Development Methodologies” <sup>[2]</sup> .....	27
Table 8 “Roles of stakeholders in Scrum” (Based on [50]) .....	31
Table 9 “About backlogs in Scrum” (Based on [50]).....	31
Table 10 “Comparison of teaching methods for students” (Based on [57]).....	34
Table 11 “Comparison of 5 Software Development Methodologies in target areas” (Based on [38][39][40]) .....	40
Table 12 “Comparison of CAL Development and Agile Software Development Methodologies” (Based on Table 7).....	41
Table 13 “CAL Development Methodology in Comparison with 12 Principles of Agile Development Methodology” (Based on Table 6) .....	43
Table 14 “CAL Development Methodology in Comparison with IID and XP, Scrum” (Based on [62]).....	44
Table 15 “Teaching Method Matrix” (Original).....	50

Table 16 “Comparison of approaches to know mechanism of a business system with Learning Pyramid” (Original) .....	50
Table 17 “Learning Patterns when I don’t know what to do.” (Based on [63]) .....	51
Table 18 “The extent of the impact on modification” (Original) .....	52
Table 19 “Roles and impacts of working by stakeholders” (Original).....	57
Table 20 “Work results of operators at 2011/02/03” (Original) .....	58
Table 21 “Web Site User Satisfaction Scorecard” (Original).....	66
Table 22 “Answers of Web Site User Satisfaction Scorecard” (Original).....	67
Table 23 “Relativity to the activeness or passiveness for the development of management strategies with the experience of the development and/or maintenance and/or operation of general business system(s)” (Original) .....	70
Table 24 “Comparison of approaches to know mechanism of a business system by old-timer engineers and amateurs” (Original).....	73
Table 25 “Interests in the application of CAL system development Methodology on their own business system being developed?” (Original).....	75
Table 26 “Voice concerns inside operation sites” <sup>[71]</sup> .....	91
Table 27 “Voice concerns about relationship with external organizations or relevant departments” <sup>[71]</sup> .....	92
Table 28 “Learning Patterns” <sup>[63]</sup> .....	92
Table 29 “Learning Patterns for particular contexts” <sup>[63]</sup> .....	93
Table 30 “Product Backlog” (Original).....	94
Table 31 “Summary of data of web pages with history” (Original).....	96
Table 32 “Data of web pages with history acquired from DB” (Original).....	98
Table 33 “Questionnaire – Attribute Information - Gender” (Original).....	123
Table 34 “Questionnaire – Attribute Information - Age” (Original) .....	123
Table 35 “Questionnaire – Attribute Information - Prefecture” (Original) .....	124
Table 36 “Questionnaire – Attribute Information - Area” (Original).....	126
Table 37 “Questionnaire – Attribute Information - Married” (Original).....	126
Table 38 “Questionnaire – Attribute Information - Child” (Original).....	127
Table 39 “Questionnaire – Attribute Information – Household Income” (Original).....	127

Table 40 “Questionnaire – Attribute Information – Job” (Original) .....	128
Table 41 “Questionnaire – Attribute Information – Student” (Original).....	128
Table 42 “Questionnaire – Attribute Information – Cell” (Original) .....	129
Table 43 “Questionnaire – Question1 & Answers” (Original).....	129
Table 44 “Questionnaire – Question2 & Answers” (Original).....	131
Table 45 “Questionnaire – Question3 & Answers” (Original).....	131
Table 46 “Questionnaire – Question4 & Answers” (Original).....	132
Table 47 “Questionnaire – Question5 & Answers” (Original).....	133
Table 48 “Questionnaire – Question6 & Answers” (Original).....	134
Table 49 “Questionnaire – Question7 & Answers” (Original).....	135
Table 50 “Questionnaire – Question8 & Answers” (Original).....	136
Table 51 “Questionnaire – Question9 & Answers” (Original).....	136
Table 52 “Questionnaire – Question10 & Answers” (Original).....	137
Table 53 “Relativity to the activeness or passiveness for the development of management strategies with the experience of the development and/or maintenance and/or operation of various business system(s)” (Original) .....	138

## 9.2. 現状分析

### 9.2.1. 運用現場内の問題点

Table 26 “Voice concerns inside operation sites” [71]

<b>運用現場内部の悩みの声</b>
業務が多岐にわたり、すべてを把握することが困難になっている。
次々と新しいサービスが開発され、その運用に現場での対応が十分でないまま、次の新たなサービスの受け入れに追われている。
運用のためのドキュメントが作られていない。あっても更新されていない。
どんなドキュメントが必要なのが分からない。書き方が分からない。
特定の人間にしかできない業務があり、その人に業務が集中している。
属人化が進み、ノウハウの継承ができていない。

異動により、現場が混乱することが多い。(上と同じ。)
設計思想が失われた、古い運用業務がある。これが現場の負担になっている。
人が育たない。優秀な人が入ってこない、定着しない。(スキル不足。低評価。)
突発的な業務が多く、計画通りに作業が進まない。残業でカバーしている。
目標が後ろ向き(稼働率など 100%からのマイナス評価)で、がんばっても評価されない。
トラブルが多く、前向きな改善に着手する余裕がない。
ツールが使いにくい、改修にはコストと期間が必要なため、我慢して使っている。
新規のツールを設計したいが、どんな要求があるのか現場でも分かっていない。

### 9.2.2. 運用現場外の問題点

Table 27 “Voice concerns about relationship with external organizations or relevant departments”  
[71]

<b>対外組織や関連部署との関係における悩みの声</b>	
サービス設計導入時の検討漏れや実装が間に合わない部分を「運用でカバーする」というように、設計側による“その場しのぎ”の影響を直接受ける。「えー、カバーするの俺たちなの？」という声が運用現場から上がるものの、上から「がんばれ」と言われるとやるしかない、という現場も多い。	
設計と運用を両方やっているため、つい「運用でカバー」することにしてしまう。こうして自分たちの業務を圧迫しながら、次の設計に追われ、疲弊しつつある現場もある。	
依頼されてから動き出すまでのリード・タイムが長くなってしまっている。変化のスピードが速い昨今では、特にリード・タイムに対する要求が厳しくなっているが、応えきれていない。	
声の大きいユーザーや部署に対して、通常の想定を大幅に超えたサポートを強いられている。	
コスト削減要求が強いが、どこをどう削減すべきなのかが見えない。	

### 9.2.3. 学習パターン

Table 28 “Learning Patterns” [63]

#	学習パターン (0~19)	#	学習パターン (20~39)
---	---------------	---	----------------

0 学びのデザイン	20 広がりと掘り下げの「T字」
1 学びのチャンス	21 隠れた関係性から学ぶ
2 創造的な学び	22 右脳と左脳のスイッチ
3 学びをひらく	23 鳥の眼と虫の眼
4 学びの竜巻	24 量は質を生む
5 知のワクワク！	25 自分で考える
6 研究への情熱	26 目的へのアプローチ
7 まずはつかる	27 捨てる勇気
8 「まねぶ」ことから	28 学びの共同体をつくる
9 教わり上手になる	29 「はなす」ことでわかる
10 身体で覚える	30 ライバルをつくる
11 成長の発見	31 教えることによる学び
12 言語のシャワー	32 外国語の普段使い
13 アウトプットから始まる学び	33 小さく生んで大きく育てる
14 プロトタイピング	34 魅せる力
15 学びの中の遊び	35 「書き上げた」は道半ば
16 動きのなかで考える	36 ゴール前のアクセル
17 フィールドに飛び込む	37 セルフプロデュース
18 偶発的な出会い	38 断固たる決意
19 フロンティアアンテナ	39 突き抜ける

#### 9.2.4. 学習パターン（コンテキスト毎）

Table 29 “Learning Patterns for particular contexts”<sup>[63]</sup>

# プロジェクトを立ち上げる時	# 作品をつくっているとき	# 自分がいま何をすべきかわからないとき
26 目的へのアプローチ	14 プロトタイピング	13 アウトプットから始まる学び
33 小さく生んで大きく育て	16 動きのなかで考える	20 広がりと掘り下げの「T

る			字」
34 魅せる力	22 右脳と左脳のスイッチ	29	「はなす」ことでわかる
37 セルフプロデュース	35 「書き上げた」は道半ば		
38 断固たる決意	36 ゴール前のアクセル		

### 9.3. 効果の検証 1 (能動的学習)

#### 9.3.1. プロダクトバックログ (一部抜粋・一部省略)

Table 30 “Product Backlog” (Original)

項番	コメント内容	回答	現在の状況
1	(前略)なので、ページとしては存在するが、プルダウンメニューとしては表示されないように、(後略)	子項目は見せない。	子項目は見せないようになっている。ただし、プルダウンメニューの表示は現在は改善されている。
2	トップページに電話番号とメールアドレス(要するに連絡先)を入れた方が良い(後略)	サイドバーまたはフッターに表示する。	サイドバーに表示した。
3	(前略)「http://naebass.jp/」がトップページになるようにしてほしい(後略)	“04”を表示させない。	トップページは、 http://naebass.jp
4	左側に「Pages:」とありますが、「最近更新されたページが表示されている」ということが伝わりにくい(後略)	「更新情報」に変更する。	日本語を表示しようとする と文字化けしてしまうので、保留中。
5	配下のページのURLが「http://naebass.jp/04/?page_id=42」などと、数字で表示されていますが、どうやらパーマリンクという機能？を使うと、数字ではなく意味のある英語にできるようです (http://naebass.jp/04/kentei/ski みたいな)。そうしたいです。(後略)	パーマリンク化する。	パーマリンクを使用する。 ちなみに URL に日本語が混じっていると、色々と面倒になる。

<p>6 メーカーのロゴが本来の縮尺よりもつぶれている気がします。(中略)本来の縮尺でやれば良いと思います。</p>	<p>「アスリート」(Columbia)とリンクする。 LANG/DYNASTER のどちらかのロゴがないので、それも追加する。</p>	<p>中央上部と下部に表示した。</p>
<p>7 もう少し写真使っても良いのでは。(後略)</p>		<p>写真を外した。</p>
<p>8 二段あるメニュー、上を開くとしたのメニュー文字とぶつかり何と書いてあるか分かりにくいです。(後略)</p>	<p>変える。</p>	<p>変えました。</p>
<p>9 メニュー下は、開くとさらに、再度クリックが必要になります。(後略)</p>	<p>#1 と同様。</p>	<p>対応中です。現在は埋込み形式で表示するようにしています。</p>



### 9.3.2. スプリントバックログ（一部抜粋・一部省略）

✓ #	▲ トラッカー	親チケット	ステータス	優先度	題名	担当者	更新日	カテゴリ	対象バージョン	予定工数	進捗 %
1	機能		終了	今すぐ	【開発】WordPressをインストールする。	TAKAHASHI Ryosuke	2011/01/03 19:21:43			1.0	<div style="width: 100%;"></div>
2	サポート		終了	通常	【管理】チケットを作成する時、「担当者」に自分の名前が表示されるようにする。	TAKAHASHI Ryosuke	2010/09/26 05:13:02			1.0	<div style="width: 100%;"></div>
3	サポート		進行中	高め	【管理】Redmineをカスタマイズする。	TAKAHASHI Ryosuke	2010/12/24 16:30:39			8.0	<div style="width: 87.5%;"></div>
4	サポート	サポート #3	終了	低め	▶【管理】チケットを作成する時、「担当者」に自分の名前が表示されるようにする。	TAKAHASHI Ryosuke	2010/09/27 00:35:06			1.0	<div style="width: 100%;"></div>
5	サポート	サポート #3	終了	低め	▶【管理】Redmineユーザーを削除する。	TAKAHASHI Ryosuke	2010/09/27 00:33:59			1.0	<div style="width: 100%;"></div>
6	サポート	サポート #3	終了	低め	▶【管理】進捗の算出方法を最適化する。	TAKAHASHI Ryosuke	2010/09/26 05:56:24			1.0	<div style="width: 100%;"></div>
7	サポート	サポート #3	終了	低め	▶【管理】チケットの題名を変更できるようにする。	TAKAHASHI Ryosuke	2010/09/27 00:31:35			1.0	<div style="width: 100%;"></div>
8	機能		終了	今すぐ	【開発】WordPressにプラグインを導入する。	TAKAHASHI Ryosuke	2011/01/03 19:24:54			2.0	<div style="width: 100%;"></div>
9	機能		進行中	今すぐ	【開発】WordPressにプラグインを導入して有効化する。	TAKAHASHI Ryosuke	2011/01/10 00:33:18			111.0	<div style="width: 100%;"></div>
10	機能		解決	急いで	【開発】WordPressにプラグインを導入して有効化する。	TAKAHASHI Ryosuke	2010/09/27 01:13:09			2.0	<div style="width: 100%;"></div>
11	機能	機能 #9	解決	高め	【開発】WordPressにプラグインを導入する。	TAKAHASHI Ryosuke	2011/01/03 19:26:01			2.0	<div style="width: 100%;"></div>
12	機能		却下	低め	【開発】Twitterと連動できるようにする。	TAKAHASHI Ryosuke	2010/12/31 21:13:23				<div style="width: 0%;"></div>
13	機能		解決	高め	【開発】WordPressにプラグインを導入して有効化する。	TAKAHASHI Ryosuke	2010/10/02 23:14:03			4.0	<div style="width: 100%;"></div>
14	機能		終了	高め	【開発】DBを定期的にバックアップする仕組みを検討・導入する。	TAKAHASHI Ryosuke	2011/01/03 19:11:02			4.0	<div style="width: 100%;"></div>
15	機能		進行中	低め	【開発】動画サイトと連動できるようにする。	TAKAHASHI Ryosuke	2011/01/02 10:44:15				<div style="width: 0%;"></div>
16	機能		新規	高め	【開発】WordPressにプラグインを導入する。	TAKAHASHI Ryosuke	2010/09/26 09:24:24			8.0	<div style="width: 0%;"></div>
17	機能		解決	急いで	【開発】プロタイプをWordPressに移行する方法を検討・実施する。	TAKAHASHI Ryosuke	2011/01/02 14:05:54			16.0	<div style="width: 100%;"></div>
18	機能		解決	高め	【開発】ユーザーを作成する。	TAKAHASHI Ryosuke	2011/01/02 14:07:12			2.0	<div style="width: 100%;"></div>

Figure 27 “Sprint Backlog” (Original)

### 9.3.3. 集計結果

Table 31 “Summary of data of web pages with history” (Original)

年月日	Developer						CAL ドライバー					
	ページ数		文字数		文字数/ページ		ページ数		文字数		文字数/ページ	
	文字追	文字	文字追	文字削	文字追	文字削	文字追	文字削	文字追	文字削	文字追	文字削
	加	削除	加	除	加	除	加	除	加	除	加	除
2011-2-26												
2011-2-19							2		115		57.5	
2011-2-12												
2011-2-5	1		14		14		9		198		22	

年月日	Developer						CAL ドライバー					
	ページ数		文字数		文字数/ページ		ページ数		文字数		文字数/ページ	
	文字追	文字	文字追	文字削	文字追	文字削	文字追	文字削	文字追	文字削	文字追	文字削
	加	削除	加	除	加	除	加	除	加	除	加	除
2011-1-29	3	2	297	-65	99	-32.5						
2011-1-22							2	1	62	-16	31	-16
2011-1-15							1	1	1,303	-1	1,303	-1
2011-1-8	1	1	70	-25	70	-25	4	1	317	-1	79	-1
2011-1-1	5		161		32		1		6		6	
2010-12-25	11	2	934	-12	85	-6	1	2	252	-121	252	-61
2010-12-18	13	2	1,648	-82	127	-41	2		335		168	
2010-12-11	12	2	1,299	-962	108	-481	1	1	3	-82	3	-82
2010-12-4												
2010-11-27	9	2	486	-3,688	54	-1,844	4		3,955		989	
2010-11-20	4		9,257		2,314		1		26,849		26,849	
2010-11-13	2		4,404		2,202							
2010-11-6	5	1	187	-3,230	37	-3,230	8	3	2,793	-9	349	-3
2010-10-30	4	2	138	-3,527	35	-1,764	16	7	21,140	-253	1,321	-36
2010-10-23							2		1,344		672	
2010-10-16							48	37	22,998	-6,238	479	-169
2010-10-9							4	5	3,372	-2,986	843	-597
2010-10-2	1		137		137							
2010-9-25												
合計	71	14	19,032	-11,591	268.06	-827.9	106	58	89,744	-9,707	846.64	-167.4

### 9.3.4. 抽出結果

Table 32 “Data of web pages with history acquired from DB”<sup>9</sup> (Original)

Title	Modified_Date	Author	列 D <sup>10</sup>	列 E <sup>11</sup>
バッジテスト (スキー検定・スノーボード検定)	2010-10-23 17:48:11	CAL ナビゲーター	0	0
バッジテスト (スキー検定・スノーボード検定)	2010-10-10 09:27:33	CAL ドライバー	0	0
スノーボードレッスン・検定	2010-10-05 23:18:09	CAL ドライバー	0	-137
スクール紹介	2010-10-04 23:30:33	CAL ナビゲーター	137	0
スクール紹介	2010-10-04 23:29:30	CAL ナビゲーター	137	0
紹介	2010-09-25 21:41:47	CAL ナビゲーター	137	137
IMG_5124	2010-10-03 22:14:30	CAL ドライバー	0	0
スキーバッジテスト (検定)	2011-01-06 22:25:22	CAL ドライバー	2977	0
スキーバッジテスト (検定)	2011-01-06 22:24:59	CAL ドライバー	2977	95
スキーバッジテスト (検定)	2010-10-23 17:36:59	CAL ドライバー	2882	0
(済) スキーバッジテスト (検定)	2010-10-23 14:12:13	CAL ドライバー	2882	0
(済) スキーバッジテスト (検定)	2010-10-10 09:36:55	CAL ドライバー	2882	0
(済) スキーバッジテスト (検定)	2010-10-10 00:49:34	CAL ドライバー	2882	0
(済) スキーバッジテスト (検定)	2010-10-09 23:30:22	CAL ドライバー	2882	988
(済) スキーバッジテスト (検定)	2010-10-09 23:23:21	CAL ドライバー	1894	126
(済) スキーバッジテスト (検定)	2010-10-09 21:50:40	CAL ドライバー	1768	1
(済) スキーバッジテスト (検定)	2010-10-09 18:59:03	CAL ドライバー	1767	-9
(済) スキーバッジテスト (検定)	2010-10-09 17:38:39	CAL ドライバー	1776	0
バッジテスト	2010-10-09 17:37:44	CAL ドライバー	1776	4
バッジテスト	2010-10-09 17:36:07	CAL ドライバー	1772	513
バッジテスト	2010-10-09 16:57:16	CAL ドライバー	1259	481
バッジテスト	2010-10-09 16:56:27	CAL ドライバー	778	752

<sup>9</sup> ID など機密性の高い列や、ページの内容など文字数が多い列は、非表示。

<sup>10</sup> ページの内容の文字数を表す列。集計用の VBA プログラム (コード非掲載) を実行して作成。

<sup>11</sup> ページの前の内容と今回の内容の文字数の差分を表す列。

Title	Modified_Date	Author	列 D <sup>10</sup>	列 E <sup>11</sup>
バッジテスト	2010-10-09 16:55:27	CAL ドライバー	26	26
レッスン (スキー・スノーボード)	2010-10-23 14:09:07	CAL ナビゲーター	0	0
レッスン (スキー・スノーボード)	2010-10-10 09:26:29	CAL ドライバー	0	0
スキーレッスン・検定	2010-10-05 23:15:50	CAL ドライバー	0	0
レッスン・検定	2010-09-13 23:05:51	CAL ナビゲーター	0	0
お申し込み・お問い合わせ	2010-11-21 11:52:46	CAL ナビゲーター	0	-3686
お申し込み・お問い合わせ	2010-11-20 11:47:30	CAL ドライバー	3686	126
お申し込み・お問い合わせ	2010-11-20 11:46:42	CAL ナビゲーター	3560	15
お申し込み・お問い合わせ	2010-11-20 11:45:36	CAL ナビゲーター	3545	146
お申し込み・お問い合わせ	2010-10-25 20:50:42	CAL ナビゲーター	3399	0
お申し込み・お問い合わせ	2010-10-23 21:42:52	CAL ドライバー	3399	0
お申し込み・お問い合わせ	2010-10-23 17:53:54	CAL ドライバー	3399	0
(済) お申し込み・お問い合わせ	2010-10-23 15:52:32	CAL ドライバー	3399	10
(済) お申し込み・お問い合わせ	2010-10-23 15:51:42	CAL ドライバー	3389	0
(済) お申し込み・お問い合わせ	2010-10-23 15:50:25	CAL ドライバー	3389	3389
お申し込み・お問い合わせ	2010-10-23 14:19:53	CAL ドライバー	0	0
お申し込み・お問い合わせ	2010-10-10 22:45:20	CAL ドライバー	0	0
お申し込み・お問い合わせ	2010-10-10 09:34:46	CAL ドライバー	0	0
お問い合わせ	2010-10-06 01:04:16	CAL ドライバー	0	0
お問い合わせ	2010-10-05 23:26:27	CAL ナビゲーター	0	0
お問い合わせ	2010-10-05 23:20:09	CAL ナビゲーター	0	0
お問い合わせ	2010-09-15 21:47:11	CAL ナビゲーター	0	0
よくあるご質問	2010-12-18 06:34:56	CAL ナビゲーター	11068	0
よくあるご質問	2010-12-18 06:32:18	CAL ナビゲーター	11068	0
よくあるご質問	2010-10-30 16:48:23	CAL ドライバー	11068	0
よくあるご質問	2010-10-30 16:47:15	CAL ナビゲーター	11068	-3230
よくあるご質問	2010-10-30 01:26:41	CAL ドライバー	14298	0
よくある質問	2010-10-25 20:50:52	CAL ナビゲーター	14298	0

Title	Modified_Date	Author	列 D <sup>10</sup>	列 E <sup>11</sup>
よくある質問	2010-10-24 23:46:45	CAL ドライバー	14298	-3
よくある質問 (更新中)	2010-10-24 23:46:03	CAL ドライバー	14301	7
よくある質問 (更新中)	2010-10-24 23:30:51	CAL ドライバー	14294	14294
よくある質問 (更新中)	2010-10-23 21:42:35	CAL ドライバー	0	0
よくある質問 (更新中)	2010-10-23 17:55:51	CAL ドライバー	0	0
よくある質問	2010-10-23 14:20:25	CAL ドライバー	0	0
よくある質問	2010-10-10 22:45:11	CAL ドライバー	0	0
よくある質問	2010-10-10 09:35:55	CAL ドライバー	0	0
よくある質問と回答	2010-09-15 21:48:15	CAL ドライバー	0	0
ジュニアテスト (検定)	2010-10-23 17:43:30	CAL ナビゲーター	1838	0
ジュニアテスト (検定)	2010-10-23 17:42:07	CAL ドライバー	1838	0
(済) ジュニアテスト (検定)	2010-10-10 23:07:37	CAL ドライバー	1838	0
(済) ジュニアテスト (検定)	2010-10-10 22:36:40	CAL ドライバー	1838	0
(済) ジュニアテスト (検定)	2010-10-10 16:02:16	CAL ドライバー	1838	-1
(済) ジュニアテスト (検定)	2010-10-10 16:01:42	CAL ドライバー	1839	13
(済) ジュニアテスト (検定)	2010-10-10 15:42:26	CAL ドライバー	1826	0
(済) ジュニアテスト (検定)	2010-10-10 15:41:24	CAL ドライバー	1826	99
(済) ジュニアテスト (検定)	2010-10-10 15:34:12	CAL ドライバー	1727	1
(済) ジュニアテスト (検定)	2010-10-10 15:33:20	CAL ドライバー	1726	1
(済) ジュニアテスト (検定)	2010-10-10 09:50:12	CAL ドライバー	1725	0
(済) ジュニアテスト (検定)	2010-10-09 22:45:52	CAL ドライバー	1725	-1
(済) ジュニアテスト (検定)	2010-10-09 22:45:20	CAL ドライバー	1726	-1
ジュニアテスト (検定)	2010-10-09 22:45:08	CAL ドライバー	1727	-766
ジュニアテスト (検定)	2010-10-09 20:01:37	CAL ドライバー	2493	-2
ジュニア検定	2010-09-20 05:12:04	CAL ドライバー	2495	2495
山荘ジュニアスキースクール	2010-10-23 17:39:21	CAL ナビゲーター	2172	0
(済) 山荘ジュニアスキースクール	2010-10-23 14:14:40	CAL ドライバー	2172	0
(済) 山荘ジュニアスキースクール	2010-10-10 17:55:40	CAL ドライバー	2172	0

Title	Modified_Date	Author	列 D <sup>10</sup>	列 E <sup>11</sup>
(済) 山荘ジュニアスキースクール	2010-10-10 17:54:38	CAL ドライバー	2172	0
(済) 山荘ジュニアスキースクール	2010-10-10 17:47:18	CAL ドライバー	2172	-4
(済) 山荘ジュニアスキースクール	2010-10-10 17:40:48	CAL ドライバー	2176	-3
(済) 山荘ジュニアスキースクール	2010-10-10 17:30:13	CAL ドライバー	2179	-1474
(済) 山荘ジュニアスキースクール	2010-10-10 09:42:58	CAL ドライバー	3653	0
(済) 山荘ジュニアスキースクール	2010-10-07 23:28:21	CAL ドライバー	3653	-1
山荘ジュニアスキースクール	2010-10-07 23:23:53	CAL ドライバー	3654	22
山荘ジュニアスキースクール	2010-10-06 22:32:21	CAL ドライバー	3632	-1
山荘ジュニアスキースクール	2010-09-20 05:18:40	CAL ドライバー	3633	3633
ホテルジュニアスキースクール	2010-10-23 17:39:32	CAL ナビゲーター	1713	0
(済) ホテルジュニアスキースクール	2010-10-23 14:14:49	CAL ドライバー	1713	0
(済) ホテルジュニアスキースクール	2010-10-14 22:33:22	CAL ドライバー	1713	0
(済) ホテルジュニアスキースクール	2010-10-10 17:59:07	CAL ドライバー	1713	-1
(済) ホテルジュニアスキースクール	2010-10-10 17:58:05	CAL ドライバー	1714	-1025
(済) ホテルジュニアスキースクール	2010-10-10 09:44:38	CAL ドライバー	2739	0
(済) ホテルジュニアスキースクール	2010-10-07 23:33:22	CAL ドライバー	2739	21
ホテルジュニアスキースクール	2010-09-20 05:28:46	CAL ドライバー	2718	2718
ジュニア検定・チャレンジ・キャンプ	2011-01-06 00:36:40	CAL ナビゲーター	1291	-25
ジュニア検定・チャレンジ・キャンプ	2011-01-06 00:35:48	CAL ドライバー	1316	5
ジュニア検定・チャレンジ・キャンプ	2010-10-23 17:42:16	CAL ドライバー	1311	0
(済) ジュニア検定・チャレンジ・キャンプ	2010-10-23 14:16:16	CAL ドライバー	1311	0
(済) ジュニア検定・チャレンジ・キャンプ	2010-10-10 23:08:45	CAL ドライバー	1311	0
(済) ジュニア検定・チャレンジ・キャンプ	2010-10-10 23:07:55	CAL ドライバー	1311	0
(済) ジュニア検定・チャレンジ・キャンプ	2010-10-10 22:36:59	CAL ドライバー	1311	0
(済) ジュニア検定・チャレンジ・キャンプ	2010-10-10 16:46:21	CAL ドライバー	1311	0
(済) ジュニア検定・チャレンジ・キャンプ	2010-10-10 16:05:21	CAL ドライバー	1311	-1
ジュニア検定・チャレンジ・キャンプ	2010-10-10 16:04:26	CAL ドライバー	1312	-485
ジュニア検定チャレンジキャンプ	2010-10-10 09:48:28	CAL ドライバー	1797	0

Title	Modified_Date	Author	列 D <sup>10</sup>	列 E <sup>11</sup>
Jr 検定チャンレンジキャンプ	2010-10-10 09:48:17	CAL ドライバー	1797	0
Jr 検定チャンレンジキャンプ	2010-10-10 09:46:51	CAL ドライバー	1797	-1
Jr 検定チャンレンジキャンプ	2010-10-10 09:45:43	CAL ドライバー	1798	-2
Jr 検定チャンレンジキャンプ	2010-09-20 05:49:23	CAL ドライバー	1800	1800
シーズンを通したジュニアテスト結果報告	2010-10-23 17:44:35	CAL ナビゲーター	0	-2805
シーズンを通したジュニアテスト結果報告	2010-10-23 14:16:27	CAL ドライバー	2805	0
シーズンを通したジュニアテスト結果報告	2010-10-10 23:06:18	CAL ドライバー	2805	0
シーズンを通したジュニアテスト結果報告	2010-10-10 23:05:25	CAL ドライバー	2805	-1
シーズンを通したジュニアテスト結果報告	2010-10-10 23:05:13	CAL ドライバー	2806	0
シーズンを通したジュニアテスト結果報告	2010-10-10 09:53:41	CAL ドライバー	2806	0
シーズンを通したジュニアテスト結果報告	2010-09-20 05:52:44	CAL ドライバー	2806	2806
ジュニアテスト合格者	2010-10-23 17:44:14	CAL ナビゲーター	0	0
ジュニアテスト合格者	2010-10-10 23:04:40	CAL ドライバー	0	0
ジュニアテスト合格者	2010-10-10 09:53:21	CAL ドライバー	0	0
ジュニアテスト合格者	2010-09-20 05:57:24	CAL ドライバー	0	0
苗場ジュニア技術選手権大会リザルト	2010-10-23 17:41:21	CAL ナビゲーター	0	-722
苗場ジュニア技術選手権大会リザルト	2010-10-23 17:40:12	CAL ドライバー	722	0
苗場ジュニア技術選手権大会リザルト	2010-10-23 14:15:29	CAL ドライバー	722	0
苗場ジュニア技術選手権大会リザルト	2010-10-20 23:50:33	CAL ドライバー	722	0
苗場ジュニア技術選手権大会リザルト	2010-10-10 09:53:59	CAL ドライバー	722	0
苗場ジュニア技術選手権大会リザルト	2010-09-20 06:06:51	CAL ドライバー	722	722
レーシング	2010-12-16 03:02:12	CAL ナビゲーター	219	122
レーシング	2010-10-23 14:18:54	CAL ナビゲーター	97	0
レーシング	2010-10-13 23:28:59	CAL ドライバー	97	7
レーシング	2010-10-13 23:26:56	CAL ドライバー	90	0
レーシング	2010-10-13 23:25:54	CAL ドライバー	90	-6
レーシング	2010-10-13 23:24:47	CAL ドライバー	96	20
レーシング	2010-10-13 23:23:34	CAL ドライバー	76	-2333

Title	Modified_Date	Author	列 D <sup>10</sup>	列 E <sup>11</sup>
レーシング	2010-10-12 23:38:27	CAL ドライバー	2409	0
スキーレーシング	2010-10-10 23:03:17	CAL ドライバー	2409	0
スキーレーシング	2010-10-10 09:31:58	CAL ドライバー	2409	0
スキーレーシング	2010-10-05 23:25:48	CAL ドライバー	2409	0
suki- レーシング	2010-10-05 23:25:27	CAL ナビゲーター	2409	0
レーシング	2010-10-05 23:25:20	CAL ナビゲーター	2409	0
一般レッスン	2010-09-22 20:32:27	CAL ナビゲーター	2409	2409
スクール紹介	2010-11-03 09:39:19	CAL ナビゲーター	0	0
スクール紹介	2010-10-21 22:51:42	CAL ナビゲーター	0	0
スクール紹介	2010-10-10 22:47:00	CAL ドライバー	0	0
スクール紹介	2010-10-10 22:46:49	CAL ドライバー	0	0
スクール紹介	2010-10-06 01:07:01	CAL ドライバー	0	-40
スクール紹介	2010-09-13 19:32:46	CAL ナビゲーター	40	40
経営理念	2010-11-03 09:31:20	CAL ナビゲーター	1676	0
経営理念	2010-11-03 09:29:45	CAL ナビゲーター	1676	0
経営理念	2010-10-23 16:27:37	CAL ナビゲーター	1676	4
(済) 経営理念	2010-10-23 16:26:51	CAL ドライバー	1672	-4
経営理念	2010-10-23 16:26:23	CAL ドライバー	1676	4
経営理念	2010-10-23 16:25:38	CAL ドライバー	1672	0
経営理念	2010-10-23 16:25:34	CAL ドライバー	1672	-4
経営理念	2010-10-23 16:22:27	CAL ドライバー	1676	-228
経営理念	2010-10-23 14:07:32	CAL ドライバー	1904	0
経営理念	2010-10-10 09:02:32	CAL ドライバー	1904	-1
経営理念	2010-10-05 23:17:37	CAL ドライバー	1905	0
経営理念	2010-10-05 23:17:07	CAL ナビゲーター	1905	0
経営理念	2010-09-13 20:00:26	CAL ナビゲーター	1905	1905
インストラクター紹介	2010-11-14 15:51:55	CAL ナビゲーター	26849	0
インストラクター紹介	2010-11-14 15:52:58	CAL ドライバー	26849	26849



Title	Modified_Date	Author	列 D <sup>10</sup>	列 E <sup>11</sup>
インストラクター紹介	2010-10-23 17:11:57	CAL ドライバー	0	0
インストラクター紹介	2010-10-23 14:07:19	CAL ドライバー	0	0
インストラクター紹介	2010-10-10 09:38:14	CAL ドライバー	0	0
スタッフ	2010-10-06 01:03:45	CAL ドライバー	0	0
スタッフ	2010-10-06 01:03:29	CAL ナビゲーター	0	0
会員のご案内	2011-01-03 20:27:18	CAL ドライバー	3915	39
会員のご案内	2010-11-03 11:27:59	CAL ドライバー	3876	-1
会員のご案内	2010-11-03 11:26:54	CAL ナビゲーター	3877	4
会員のご案内	2010-11-03 11:26:19	CAL ドライバー	3873	62
会員のご案内	2010-11-02 23:44:31	CAL ドライバー	3811	606
会員のご案内	2010-10-30 16:57:29	CAL ドライバー	3205	0
会員のご案内	2010-10-23 17:50:20	CAL ドライバー	3205	0
(済) 会員のご案内	2010-10-23 14:18:00	CAL ドライバー	3205	0
(済) 会員のご案内	2010-10-12 23:40:54	CAL ドライバー	3205	-1
(済) 会員のご案内	2010-10-12 23:40:14	CAL ドライバー	3206	-1
(済) 苗場スキースクール会員のご案内	2010-10-10 23:02:48	CAL ドライバー	3207	0
(済) 苗場スキースクール会員のご案内	2010-10-10 09:30:40	CAL ドライバー	3207	-1
(済) 苗場スキースクールメンバーズ (会員)	2010-10-09 16:20:50	CAL ドライバー	3208	-7
(済) 苗場スキースクールメンバーズ (会員)	2010-10-09 16:18:25	CAL ドライバー	3215	1
(済) 苗場スキースクールメンバーズ (会員)	2010-10-09 16:17:41	CAL ドライバー	3214	-4
(済) 苗場スキースクールメンバーズ (会員)	2010-10-09 16:17:28	CAL ドライバー	3218	-5
苗場スキースクールメンバーズ (会員)	2010-10-09 16:17:14	CAL ドライバー	3223	1446
苗場スキースクールメンバーズ (会員)	2010-10-09 16:04:40	CAL ドライバー	1777	1255
苗場スキースクールメンバーズ (会員)	2010-10-08 23:59:56	CAL ドライバー	522	-2807
苗場スキースクールメンバーズ (会員)	2010-10-08 23:58:06	CAL ドライバー	3329	0
苗場スキースクールメンバーズ (会員)	2010-10-08 23:57:46	CAL ドライバー	3329	2868
苗場スキースクールメンバーズ (会員)	2010-10-08 23:56:28	CAL ドライバー	461	461
苗場スキースクールメンバーズ (会員)	2010-10-08 23:56:08	CAL ドライバー	0	0

Title	Modified_Date	Author	列 D <sup>10</sup>	列 E <sup>11</sup>
スノーボードバッジテスト (検定)	2011-01-13 22:49:17	CAL ドライバー	2147	-1
スノーボードバッジテスト (検定)	2011-01-13 22:48:47	CAL ドライバー	2148	1303
スノーボードバッジテスト (検定)	2010-10-23 17:37:32	CAL ドライバー	845	0
(済) スノーボードバッジテスト (検定)	2010-10-23 14:13:51	CAL ドライバー	845	0
(済) スノーボードバッジテスト (検定)	2010-10-10 22:37:22	CAL ドライバー	845	0
(済) スノーボードバッジテスト (検定)	2010-10-10 09:49:34	CAL ドライバー	845	0
(済) スノーボードバッジテスト (検定)	2010-10-10 01:19:33	CAL ドライバー	845	0
スノーボードバッジテスト (検定)	2010-10-10 01:18:43	CAL ドライバー	845	0
スノーボードバッジテスト (検定)	2010-10-10 01:18:41	CAL ドライバー	845	845
ナイターバッジテストレッスン (スキー)	2010-10-23 17:37:20	CAL ドライバー	812	0
(済) ナイターバッジテストレッスン (スキー)	2010-10-23 14:13:23	CAL ドライバー	812	0
(済) ナイターバッジテストレッスン (スキー)	2010-10-13 20:43:59	CAL ドライバー	812	0
(済) ナイターバッジテストレッスン (スキー)	2010-10-10 22:36:26	CAL ドライバー	812	0
(済) ナイターバッジテストレッスン (スキー)	2010-10-10 09:49:45	CAL ドライバー	812	0
(済) ナイターバッジテストレッスン (スキー)	2010-10-10 08:42:56	CAL ドライバー	812	0
ナイターバッジテストレッスン (スキー)	2010-10-10 08:42:38	CAL ドライバー	812	15
ナイターバッジテストレッスン	2010-10-10 08:41:24	CAL ドライバー	797	0
ナイターバッジテストレッスン	2010-10-10 08:41:01	CAL ドライバー	797	797
ナイターバッジテストレッスン	2010-10-10 08:32:25	CAL ドライバー	0	0
ご挨拶	2010-11-03 09:29:44	CAL ドライバー	422	0
ご挨拶	2010-10-23 17:07:57	CAL ナビゲーター	422	0
(済) ご挨拶	2010-10-23 15:54:01	CAL ドライバー	422	0
(済) ご挨拶	2010-10-23 15:52:58	CAL ドライバー	422	0
(済) ご挨拶	2010-10-23 15:18:28	CAL ドライバー	422	0
ご挨拶	2010-10-23 14:54:12	CAL ドライバー	422	221
ご挨拶	2010-10-23 14:50:16	CAL ドライバー	201	0
ご挨拶	2010-10-23 14:49:11	CAL ドライバー	201	201
ご挨拶	2010-10-21 22:52:10	CAL ドライバー	0	0

Title	Modified_Date	Author	列 D <sup>10</sup>	列 E <sup>11</sup>
ご挨拶	2010-10-10 09:01:23	CAL ドライバー	0	0
ご挨拶	2010-10-10 09:01:16	CAL ドライバー	0	0
ご挨拶	2010-10-10 09:01:05	CAL ドライバー	0	0
ご挨拶	2010-10-10 09:00:57	CAL ドライバー	0	0
ご挨拶	2010-10-10 08:59:52	CAL ドライバー	0	0
プライバシーポリシー	2010-10-23 17:12:07	CAL ドライバー	2908	0
プライバシーポリシー	2010-10-23 17:07:04	CAL ドライバー	2908	0
プライバシーポリシー	2010-10-23 17:06:41	CAL ドライバー	2908	2908
プライバシーポリシー	2010-10-23 14:08:24	CAL ドライバー	0	0
プライバシーポリシー	2010-10-10 09:06:30	CAL ドライバー	0	0
プライバシーポリシー	2010-10-10 09:06:25	CAL ドライバー	0	0
採用情報	2010-12-18 05:25:15	CAL ドライバー	1310	-34
採用情報	2010-10-23 17:12:15	CAL ナビゲーター	1344	0
採用情報	2010-10-23 17:11:00	CAL ドライバー	1344	4
採用情報	2010-10-23 17:08:24	CAL ドライバー	1340	0
(済) 採用情報	2010-10-23 14:30:10	CAL ドライバー	1340	-4
採用情報	2010-10-23 14:08:32	CAL ドライバー	1344	0
採用情報	2010-10-20 23:48:21	CAL ドライバー	1344	0
採用情報	2010-10-20 23:47:46	CAL ドライバー	1344	254
採用情報	2010-10-20 23:16:56	CAL ドライバー	1090	1090
採用情報	2010-10-10 09:21:14	CAL ドライバー	0	0
採用情報	2010-10-10 09:21:04	CAL ドライバー	0	0
スキルアッププログラム	2010-12-13 02:38:22	CAL ドライバー	0	0
スキルアッププログラム	2010-10-23 14:17:19	CAL ナビゲーター	0	0
スキルアッププログラム	2010-10-10 23:02:22	CAL ドライバー	0	0
スキルアッププログラム	2010-10-10 22:47:58	CAL ドライバー	0	0
スキルアッププログラム	2010-10-10 09:28:19	CAL ドライバー	0	0
スキルアッププログラム	2010-10-10 09:28:16	CAL ドライバー	0	0

Title	Modified_Date	Author	列 D <sup>10</sup>	列 E <sup>11</sup>
ジュニア（子供向け）プログラム	2010-10-23 14:14:00	CAL ドライバー	0	0
ジュニア（子供向け）プログラム	2010-10-10 22:47:51	CAL ドライバー	0	0
ジュニア（子供向け）プログラム	2010-10-10 09:29:06	CAL ドライバー	0	0
ジュニア（子供向け）プログラム	2010-10-10 09:29:05	CAL ドライバー	0	0
国内・海外ツアープログラム	2010-12-13 02:39:53	CAL ドライバー	0	0
国内・海外ツアープログラム	2010-10-30 16:57:17	CAL ナビゲーター	0	0
国内・海外ツアープログラム	2010-10-23 14:18:18	CAL ドライバー	0	0
国内・海外ツアープログラム	2010-10-12 23:41:16	CAL ドライバー	0	0
国内・海外ツアープログラム	2010-10-10 23:02:40	CAL ドライバー	0	0
国内・海外ツアープログラム	2010-10-10 09:29:25	CAL ドライバー	0	0
国内・海外ツアープログラム	2010-10-10 09:29:23	CAL ドライバー	0	0
苗場スキースクール公式ブログ	2010-12-15 03:03:08	CAL ドライバー	174	0
苗場スキースクール公式ブログ	2010-12-15 03:00:42	CAL ナビゲーター	174	0
苗場スキースクール公式ブログ	2010-12-15 03:00:03	CAL ナビゲーター	174	0
苗場スキースクール公式ブログ	2010-12-15 02:56:55	CAL ナビゲーター	174	0
苗場スキースクール公式ブログ	2010-12-15 02:56:34	CAL ナビゲーター	174	0
苗場スキースクール公式ブログ	2010-12-15 02:53:26	CAL ナビゲーター	174	0
苗場スキースクール公式ブログ	2010-12-15 02:52:21	CAL ナビゲーター	174	-3
苗場スキースクール公式ブログ	2010-12-15 02:51:03	CAL ナビゲーター	177	3
苗場スキースクール公式ブログ	2010-12-15 02:49:33	CAL ナビゲーター	174	94
苗場スキースクール公式ブログ	2010-12-13 02:24:42	CAL ナビゲーター	80	0
苗場スキースクール公式ブログ	2010-12-13 02:20:32	CAL ナビゲーター	80	0
苗場スキースクール公式ブログ	2010-10-23 17:52:57	CAL ナビゲーター	80	80
苗場スキースクール公式ブログ	2010-10-23 14:18:31	CAL ドライバー	0	0
苗場スキースクール公式ブログ	2010-10-10 23:02:55	CAL ドライバー	0	0
苗場スキースクール公式ブログ	2010-10-10 22:45:53	CAL ドライバー	0	0
苗場スキースクール公式ブログ	2010-10-10 09:31:23	CAL ドライバー	0	0
苗場スキースクール公式ブログ	2010-10-10 09:31:20	CAL ドライバー	0	0

Title	Modified_Date	Author	列 D <sup>10</sup>	列 E <sup>11</sup>
苗場レーシングブログ	2010-12-15 03:03:18	CAL ドライバー	207	0
苗場レーシングブログ	2010-12-15 03:00:45	CAL ナビゲーター	207	0
苗場レーシングブログ	2010-12-15 02:57:56	CAL ナビゲーター	207	111
レーシングブログ	2010-12-13 02:25:00	CAL ナビゲーター	96	0
レーシングブログ	2010-12-13 02:21:09	CAL ナビゲーター	96	0
レーシングブログ	2010-10-23 14:19:04	CAL ナビゲーター	96	0
レーシングブログ	2010-10-13 23:28:11	CAL ドライバー	96	96
レーシングブログ	2010-10-10 22:45:44	CAL ドライバー	0	0
レーシングブログ	2010-10-10 09:32:58	CAL ドライバー	0	0
レーシングブログ	2010-10-10 09:32:32	CAL ドライバー	0	0
たけのこやま山荘	2010-12-16 03:09:30	CAL ドライバー	300	184
たけのこやま山荘	2010-10-25 20:50:24	CAL ナビゲーター	116	0
たけのこやま山荘	2010-10-23 21:43:13	CAL ドライバー	116	0
たけのこやま山荘	2010-10-23 14:19:15	CAL ドライバー	116	0
たけのこやま山荘	2010-10-13 23:31:04	CAL ドライバー	116	2
たけのこやま山荘	2010-10-13 23:30:28	CAL ドライバー	114	114
たけのこやま山荘	2010-10-12 23:37:50	CAL ドライバー	0	0
たけのこやま山荘のご紹介	2010-10-10 22:45:37	CAL ドライバー	0	0
たけのこやま山荘のご紹介	2010-10-10 09:33:26	CAL ドライバー	0	0
たけのこやま山荘	2010-10-10 09:33:21	CAL ドライバー	0	0
狭山校	2010-12-16 03:11:59	CAL ドライバー	252	151
狭山校	2010-10-25 20:50:34	CAL ナビゲーター	101	0
狭山校	2010-10-23 21:43:02	CAL ドライバー	101	0
狭山校	2010-10-23 14:19:27	CAL ドライバー	101	0
狭山校	2010-10-13 23:33:34	CAL ドライバー	101	0
狭山校	2010-10-13 23:32:32	CAL ドライバー	101	101
狭山校	2010-10-10 22:45:29	CAL ドライバー	0	0
狭山校	2010-10-10 09:33:46	CAL ドライバー	0	0

Title	Modified_Date	Author	列 D <sup>10</sup>	列 E <sup>11</sup>
狭山校	2010-10-10 09:33:39	CAL ドライバー	0	0
S A J 公認 1・2 級バッジテストコース (スキー検定コース)	2010-12-25 18:56:20	CAL ドライバー	2592	6
S A J 公認 1・2 級バッジテストコース (スキー検定コース)	2010-12-25 18:55:27	CAL ナビゲーター	2586	5
S A J 公認 1・2 級バッジテストコース (スキー検定コース)	2010-12-25 18:55:10	CAL ナビゲーター	2581	3
S A J 公認 1・2 級バッジテストコース (スキー検定コース)	2010-12-25 18:53:58	CAL ナビゲーター	2578	0
S A J 公認 1・2 級バッジテストコース (スキー検定コース)	2010-12-25 18:53:20	CAL ナビゲーター	2578	0
S A J 公認 1・2 級バッジテストコース (スキー検定コース)	2010-12-25 18:52:39	CAL ナビゲーター	2578	0
S A J 公認 1・2 級バッジテストコース (スキー検定コース)	2010-12-25 18:46:54	CAL ナビゲーター	2578	9
S A J 公認 1・2 級バッジテストコース (スキー検定コース)	2010-12-25 18:44:51	CAL ナビゲーター	2569	0
S A J 公認 1・2 級バッジテストコース (スキー検定コース)	2010-10-23 17:37:07	CAL ナビゲーター	2569	0
(済) S A J 公認 1・2 級バッジテストコース (スキー検定コース)	2010-10-23 14:12:26	CAL ドライバー	2569	0
(済) S A J 公認 1・2 級バッジテストコース (スキー検定コース)	2010-10-12 23:03:38	CAL ドライバー	2569	1
(済) S A J 公認 1・2 級バッジテストコース (スキー検定コース)	2010-10-12 22:57:53	CAL ドライバー	2568	1
(済) S A J 公認 1・2 級バッジテストコース (スキー検定コース)	2010-10-10 22:34:55	CAL ドライバー	2567	-1
(済) S A J 公認 1・2 級バッジテストコース (スキー検定コース)	2010-10-10 15:02:58	CAL ドライバー	2568	-3
(済) S A J 公認 1・2 級バッジテストコース	2010-10-10 14:40:40	CAL ドライバー	2571	273
(済) S A J 公認 1・2 級バッジテストコース	2010-10-10 14:21:54	CAL ドライバー	2298	12
(済) S A J 公認 1・2 級バッジテストコース	2010-10-10 14:20:37	CAL ドライバー	2286	2286
S A J 公認 1・2 級バッジテストコース	2010-10-10 13:58:31	CAL ドライバー	0	0
S A J 公認 1・2 級バッジテストコース	2010-10-10 13:58:23	CAL ドライバー	0	0
S A J 公認 1・2 級バッジテストコース	2010-10-10 13:41:09	CAL ドライバー	0	0
S A J 公認 1・2 級 シルバー・バッジテストコース (スキー検定コース)	2010-10-23 17:37:13	CAL ドライバー	1281	0

Title	Modified_Date	Author	列 D <sup>10</sup>	列 E <sup>11</sup>
(済) S A J 公認 1・2 級 シルバー・バッジテストコース (ス キー検定コース)	2010-10-23 14:13:04	CAL ドライバー	1281	0
(済) S A J 公認 1・2 級 シルバー・バッジテストコース (ス キー検定コース)	2010-10-10 22:35:53	CAL ドライバー	1281	-1
(済) S A J 公認 1・2 級 シルバー・バッジテストコース (ス キー検定コース)	2010-10-10 15:03:18	CAL ドライバー	1282	0
(済) S A J 公認 1・2 級 シルバー・バッジテストコース (ス キー検定コース)	2010-10-10 15:02:29	CAL ドライバー	1282	0
(済) S A J 公認 1・2 級 シルバー・バッジテストコース (ス キー検定コース)	2010-10-10 15:02:06	CAL ドライバー	1282	0
(済) S A J 公認 1・2 級 シルバー・バッジテストコース (ス キー検定コース)	2010-10-10 15:02:05	CAL ドライバー	1282	1282
山荘ジュニアスノーボード・キャンプ	2010-10-23 17:39:49	CAL ドライバー	1357	0
山荘ジュニアスノーボード・キャンプ	2010-10-30 19:22:58	CAL ドライバー	1357	0
(済) 山荘ジュニアスノーボード・キャンプ	2010-10-23 14:14:17	CAL ドライバー	1357	0
(済) 山荘ジュニアスノーボード・キャンプ	2010-10-10 17:48:36	CAL ドライバー	1357	-2
(済) 山荘 J r スノーボード・キャンプ	2010-10-10 17:22:11	CAL ドライバー	1359	0
(済) 山荘 J r スノーボード・キャンプ	2010-10-10 17:21:23	CAL ドライバー	1359	256
(済) 山荘 J r スノーボード・キャンプ	2010-10-10 17:15:36	CAL ドライバー	1103	-79
(済) 山荘 J r スノーボード・キャンプ	2010-10-10 17:14:33	CAL ドライバー	1182	1182
(済) 山荘 J r スノーボード・キャンプ	2010-10-10 17:14:13	CAL ドライバー	0	0
ジュニアテスト	2010-10-23 14:15:49	CAL ドライバー	0	0
ジュニアテスト	2010-10-10 22:48:45	CAL ドライバー	0	0
ジュニアテスト	2010-10-10 22:43:42	CAL ドライバー	0	0
自動下書き	2010-10-10 22:43:31	CAL ドライバー	0	0
苗場 S S 小回り & コブ 特別講習	2010-12-07 23:51:01	CAL ドライバー	1552	3
苗場 S S 小回り & コブ 特別講習	2010-12-07 23:52:04	CAL ドライバー	1549	-82
苗場 S S 小回り & コブ 特別講習	2010-12-07 23:48:47	CAL ドライバー	1631	0

Title	Modified_Date	Author	列 D <sup>10</sup>	列 E <sup>11</sup>
苗場 S S 小回り & コブ 特別講習	2010-10-23 17:49:15	CAL ドライバー	1631	0
(済) 苗場 S S 小回り & コブ 特別講習	2010-10-23 14:17:28	CAL ドライバー	1631	0
(済) 苗場 S S 小回り & コブ 特別講習	2010-10-13 23:20:25	CAL ドライバー	1631	0
(済) 苗場 S S 小回り & コブ 特別講習	2010-10-12 23:02:02	CAL ドライバー	1631	0
(済) 苗場 S S 小回り & コブ 特別講習	2010-10-12 23:01:53	CAL ドライバー	1631	1631
青木哲也・スキルアップ キャンプ	2010-10-23 17:49:30	CAL ドライバー	997	0
青木哲也・スキルアップ キャンプ	2010-10-30 17:29:33	CAL ドライバー	997	0
(済) 青木哲也・スキルアップ キャンプ	2010-10-23 14:17:51	CAL ドライバー	997	0
(済) 青木哲也・スキルアップ キャンプ	2010-10-13 23:20:40	CAL ドライバー	997	0
(済) 青木哲也・スキルアップ キャンプ	2010-10-12 23:35:44	CAL ドライバー	997	0
(済) 青木哲也・スキルアップ キャンプ	2010-10-12 23:35:37	CAL ドライバー	997	997
苗場ジュニア技術選手権大会	2010-10-23 17:40:02	CAL ドライバー	1938	0
(済) 苗場ジュニア技術選手権大会	2010-10-23 14:14:31	CAL ドライバー	1938	0
(済) 苗場ジュニア技術選手権大会	2010-10-13 20:32:32	CAL ドライバー	1938	-2
(済) 苗場ジュニア技術選手権大会	2010-10-13 20:32:03	CAL ドライバー	1940	1
(済) 苗場ジュニア技術選手権大会	2010-10-13 20:31:07	CAL ドライバー	1939	1
(済) 苗場ジュニア技術選手権大会	2010-10-13 20:29:07	CAL ドライバー	1938	-6
(済) 苗場ジュニア技術選手権大会	2010-10-13 20:28:55	CAL ドライバー	1944	0
(済) 苗場ジュニア技術選手権大会	2010-10-13 20:28:26	CAL ドライバー	1944	0
(済) 苗場ジュニア技術選手権大会	2010-10-13 20:28:18	CAL ドライバー	1944	1944
スキー・グループ・レッスン	2010-10-23 17:34:55	CAL ドライバー	731	2
(済) スキー・グループ・レッスン	2010-10-23 15:19:43	CAL ドライバー	729	-2
(済) スキー・グループ・レッスン	2010-10-23 14:09:36	CAL ドライバー	731	0
(済) スキー・グループ・レッスン	2010-10-13 21:22:25	CAL ドライバー	731	0
(済) スキー・グループ・レッスン	2010-10-13 21:21:46	CAL ドライバー	731	0
(済) スキー・グループ・レッスン	2010-10-13 21:21:36	CAL ドライバー	731	731
スノーボード・グループ・レッスン	2010-10-23 17:35:11	CAL ドライバー	607	0
(済) スノーボード・グループ・レッスン	2010-10-23 14:10:38	CAL ドライバー	607	1



Title	Modified_Date	Author	列 D <sup>10</sup>	列 E <sup>11</sup>
(済) スノーボード・グループ・レッスン	2010-10-13 21:41:19	CAL ドライバー	606	-1
(済) スノーボード・グループ・レッスン	2010-10-13 21:40:16	CAL ドライバー	607	607
(済) スノーボード・グループ・レッスン	2010-10-13 21:39:50	CAL ドライバー	0	0
スキー・プライベート・レッスン	2010-10-23 17:35:03	CAL ドライバー	785	0
(済) スキー・プライベート・レッスン	2010-10-23 14:10:13	CAL ドライバー	785	1
(済) スキー・プライベート・レッスン	2010-10-13 21:51:37	CAL ドライバー	784	-1
(済) スキー・プライベート・レッスン	2010-10-13 21:50:35	CAL ドライバー	785	0
(済) スキー・プライベート・レッスン	2010-10-13 21:50:28	CAL ドライバー	785	785
スノーボード・プライベート・レッスン	2010-10-23 17:35:18	CAL ドライバー	769	1
スノーボード・プライベート・レッスン	2011-01-06 23:02:21	CAL ドライバー	768	-1
(済) スノーボード・プライベート・レッスン	2010-10-23 14:10:49	CAL ドライバー	769	0
(済) スノーボード・プライベート・レッスン	2010-10-13 21:52:28	CAL ドライバー	769	0
(済) スノーボード・プライベート・レッスン	2010-10-13 21:52:22	CAL ドライバー	769	769
団体レッスン	2010-10-23 17:35:25	CAL ドライバー	595	0
(済) 団体レッスン	2010-10-23 14:10:58	CAL ドライバー	595	2
(済) 団体レッスン	2010-10-13 21:59:28	CAL ドライバー	593	-2
(済) 団体レッスン	2010-10-13 21:58:26	CAL ドライバー	595	12
(済) 団体レッスン	2010-10-13 21:57:46	CAL ドライバー	583	0
(済) 団体レッスン	2010-10-13 21:57:41	CAL ドライバー	583	583
春スキー・レッスン	2010-10-23 17:35:32	CAL ドライバー	472	0
(済) 春スキー・レッスン	2010-10-23 14:11:06	CAL ドライバー	472	0
(済) 春スキー・レッスン	2010-10-13 22:26:14	CAL ドライバー	472	34
(済) 春スキー・レッスン	2010-10-13 22:25:25	CAL ドライバー	438	-1
(済) 春スキー・レッスン	2010-10-13 22:24:58	CAL ドライバー	439	11
(済) 春スキー・レッスン	2010-10-13 22:24:26	CAL ドライバー	428	0
(済) 春スキー・レッスン	2010-10-13 22:24:24	CAL ドライバー	428	428
ジュニア・グループ・レッスン	2010-10-23 17:39:04	CAL ドライバー	950	0
(済) ジュニア・グループ・レッスン	2010-10-23 14:15:38	CAL ドライバー	950	3

Title	Modified_Date	Author	列 D <sup>10</sup>	列 E <sup>11</sup>
(済) ジュニア・グループ・レッスン	2010-10-13 22:39:04	CAL ドライバー	947	-3
(済) ジュニア・グループ・レッスン	2010-10-13 22:38:02	CAL ドライバー	950	0
(済) ジュニア・グループ・レッスン	2010-10-13 22:37:51	CAL ドライバー	950	950
水曜日はコブ小回り	2010-10-23 17:49:22	CAL ドライバー	517	0
水曜日はコブ小回り	2010-10-30 19:18:50	CAL ドライバー	517	0
(済) 水曜日はコブ小回り	2010-10-23 14:17:40	CAL ドライバー	517	0
(済) 水曜日はコブ小回り	2010-10-13 23:19:28	CAL ドライバー	517	0
(済) 水曜日はコブ小回り	2010-10-13 23:19:12	CAL ドライバー	517	517
kouchou	2010-10-23 14:53:03	CAL ドライバー	0	0
バッジテスト合格者	2010-10-23 17:46:56	CAL ドライバー	0	0
バッジテスト合格者	2010-10-23 17:46:02	CAL ドライバー	0	0
バッジテスト合格者	2010-10-23 17:45:46	CAL ドライバー	0	0
シーズンを通したバッジテスト結果報告	2010-10-23 17:47:14	CAL ドライバー	0	0
シーズンを通したバッジテスト結果報告	2010-10-23 17:46:25	CAL ドライバー	0	0
シーズンを通したバッジテスト結果報告	2010-10-23 17:46:18	CAL ドライバー	0	0
苗場スノーボードスクールブログ	2010-12-15 03:03:28	CAL ドライバー	183	0
苗場スノーボードスクールブログ	2010-12-15 03:01:15	CAL ナビゲーター	183	0
苗場スノーボードスクールブログ	2010-12-15 03:01:07	CAL ナビゲーター	183	99
苗場スノーボードスクールブログ	2010-12-13 02:25:15	CAL ナビゲーター	84	0
苗場スノーボードスクールブログ	2010-12-13 02:22:00	CAL ナビゲーター	84	0
苗場スノーボードスクールブログ	2010-10-23 21:45:22	CAL ナビゲーター	84	0
苗場スノーボードスクールブログ	2010-10-23 21:43:30	CAL ドライバー	84	0
苗場スノーボードスクールブログ	2010-10-23 21:41:02	CAL ドライバー	84	-8
苗場スノーボードスクールブログ	2010-10-23 21:39:10	CAL ドライバー	92	92
スノーリモ	2010-12-18 04:53:06	CAL ドライバー	165	-87
スノーリモ	2010-12-18 04:51:34	CAL ナビゲーター	252	252
スノーリモ	2010-12-18 04:48:05	CAL ナビゲーター	0	0
スノーリモ	2010-10-23 21:48:57	CAL ナビゲーター	0	0

Title	Modified_Date	Author	列 D <sup>10</sup>	列 E <sup>11</sup>
スノーリモ	2010-10-23 21:43:41	CAL ドライバー	0	0
スノーリモ	2010-10-23 21:41:47	CAL ドライバー	0	0
自動下書き	2010-10-23 21:41:32	CAL ドライバー	0	0
日本モンブランクラブ	2010-10-23 22:15:22	CAL ドライバー	0	0
日本モンブランクラブ	2010-10-23 22:11:47	CAL ドライバー	0	0
日本モンブランクラブ	2010-10-23 22:11:40	CAL ドライバー	0	0
間もなくリリース	2010-12-18 07:22:17	CAL ナビゲーター	54	0
間もなくリリース	2010-11-20 12:09:33	CAL ナビゲーター	54	0
間もなくリリース	2010-11-03 10:23:21	CAL ナビゲーター	54	0
間もなくリリース	2010-11-03 10:22:22	CAL ナビゲーター	54	0
間もなくリリース	2010-11-03 10:21:17	CAL ナビゲーター	54	0
間もなくリリース	2010-11-03 09:58:38	CAL ナビゲーター	54	0
間もなくリリース	2010-11-03 09:57:45	CAL ナビゲーター	54	0
間もなくリリース	2010-11-03 09:43:30	CAL ナビゲーター	54	0
間もなくリリース	2010-10-24 16:26:10	CAL ナビゲーター	54	33
間もなくリリース	2010-10-24 16:25:22	CAL ナビゲーター	21	21
ミッフィースキーキャンプ	2010-12-18 04:52:34	CAL ドライバー	252	252
ミッフィースキーキャンプ	2010-12-18 04:47:46	CAL ナビゲーター	0	0
ミッフィースキーキャンプ	2010-10-25 20:50:16	CAL ナビゲーター	0	0
ミッフィースキーキャンプ	2010-10-25 20:49:33	CAL ドライバー	0	0
ミッフィースキーキャンプ	2010-10-25 20:49:25	CAL ドライバー	0	0
月山・小回り&コブ・キャンプ	2010-10-30 23:10:13	CAL ドライバー	759	1
月山・小回り&コブ・キャンプ	2010-10-30 23:39:56	CAL ドライバー	758	758
月山・小回り&コブ・キャンプ	2010-10-30 16:55:45	CAL ドライバー	0	0
月山・小回り&コブ・キャンプ	2010-10-30 16:54:17	CAL ドライバー	0	0
月山・小回り&コブ・キャンプ	2010-10-30 16:54:03	CAL ドライバー	0	0
乗鞍サマーベーシックキャンプ	2010-10-30 23:50:16	CAL ドライバー	819	1
乗鞍サマーベーシックキャンプ	2010-10-30 23:51:21	CAL ドライバー	818	-1

Title	Modified_Date	Author	列 D <sup>10</sup>	列 E <sup>11</sup>
乗鞍サマーベシクキャンプ	2010-10-30 23:49:55	CAL ドライバー	819	-7
乗鞍サマーベシクキャンプ	2010-10-30 23:45:45	CAL ドライバー	826	826
乗鞍サマーベシクキャンプ	2010-10-30 16:55:56	CAL ドライバー	0	0
乗鞍サマーベシクキャンプ	2010-10-30 16:54:54	CAL ドライバー	0	0
乗鞍サマーベシクキャンプ	2010-10-30 16:54:38	CAL ドライバー	0	0
ニュージーランドキャンプ	2010-10-30 23:54:48	CAL ドライバー	539	1
ニュージーランドキャンプ	2010-10-30 23:55:51	CAL ドライバー	538	538
ニュージーランドキャンプ	2010-10-30 16:56:06	CAL ドライバー	0	0
ニュージーランドキャンプ	2010-10-30 16:55:25	CAL ドライバー	0	0
ニュージーランドキャンプ	2010-10-30 16:55:11	CAL ドライバー	0	0
1204120773	2010-11-01 22:50:16	CAL ドライバー	0	0
1204120773	2010-11-01 22:50:40	CAL ドライバー	0	0
リリースしました！	2010-12-18 07:23:01	CAL ナビゲーター	183	0
リリースしました！	2010-11-20 12:08:40	CAL ナビゲーター	183	0
リリースしました！	2010-11-03 10:29:20	CAL ナビゲーター	183	0
リリースしました！	2010-11-03 10:28:12	CAL ナビゲーター	183	0
リリースしました	2010-11-03 10:26:45	CAL ナビゲーター	183	0
リリースしました	2010-11-03 10:25:56	CAL ナビゲーター	183	0
リリースしました！	2010-11-03 10:20:05	CAL ナビゲーター	183	0
リリースしました！	2010-11-03 10:15:21	CAL ナビゲーター	183	1
リリースしました！	2010-11-03 10:14:19	CAL ナビゲーター	182	0
リリースしました！	2010-11-03 10:12:46	CAL ナビゲーター	182	0
リリースしました！	2010-11-03 09:45:35	CAL ナビゲーター	182	53
リリースしました！	2010-11-03 09:43:48	CAL ナビゲーター	129	0
リリースしました！	2010-11-03 09:42:23	CAL ナビゲーター	129	0
リリースしました！	2010-11-03 09:41:45	CAL ナビゲーター	129	68
リリースしました！	2010-11-03 09:41:18	CAL ナビゲーター	61	61
1204120773	2010-11-06 15:53:15	CAL ドライバー	0	0

Title	Modified_Date	Author	列 D <sup>10</sup>	列 E <sup>11</sup>
メール送信フォームサンプル	2010-11-14 18:33:37	CAL ナビゲーター	551	0
メール送信フォームサンプル	2010-11-14 18:33:03	CAL ナビゲーター	551	0
メール送信フォームサンプル	2010-11-14 18:27:17	CAL ナビゲーター	551	0
メール送信フォームサンプル	2010-11-07 09:04:52	CAL ナビゲーター	551	0
メール送信フォームサンプル	2010-11-07 06:36:56	CAL ナビゲーター	551	0
メール送信フォームサンプル	2010-11-07 06:36:46	CAL ナビゲーター	551	551
[非公開] メール送信フォーム	2010-11-07 06:35:03	CAL ナビゲーター	0	0
メール送信フォーム	2010-11-07 06:34:49	CAL ナビゲーター	0	0
メール送信フォーム	2010-11-07 06:34:21	CAL ナビゲーター	0	0
申込書(会員) 2010-2011 サンプル	2010-11-20 04:20:44	CAL ナビゲーター	3853	0
申込書(会員) 2010-2011 サンプル	2010-11-11 01:18:53	CAL ナビゲーター	3853	0
申込書(会員) 2010-2011 サンプル	2010-11-11 01:18:31	CAL ナビゲーター	3853	0
申込書(会員) 2010-2011 サンプル	2010-11-11 01:17:44	CAL ナビゲーター	3853	3853
申込書(行事) 2010-2011 サンプル	2010-11-20 12:07:49	CAL ナビゲーター	2702	0
申込書(行事) 2010-2011 サンプル	2010-11-20 21:09:56	CAL ドライバー	2702	0
申込書(行事) 2010-2011 サンプル	2010-11-14 17:01:16	CAL ナビゲーター	2702	0
申込書(行事) 2010-2011 サンプル	2010-11-14 16:59:38	CAL ナビゲーター	2702	0
申込書	2010-11-14 16:58:52	CAL ナビゲーター	2702	2702
会員申込	2010-11-21 11:47:25	CAL ナビゲーター	3853	0
会員申込	2010-11-20 11:37:18	CAL ドライバー	3853	0
会員申込	2010-11-15 01:27:14	CAL ナビゲーター	3853	0
会員申込	2010-11-14 18:45:58	CAL ナビゲーター	3853	0
会員申込	2010-11-14 18:45:05	CAL ナビゲーター	3853	0
会員申込	2010-11-14 18:44:42	CAL ナビゲーター	3853	0
会員申込	2010-11-14 18:41:34	CAL ナビゲーター	3853	0
会員申込	2010-11-14 18:40:53	CAL ナビゲーター	3853	0
会員申込	2010-11-14 18:38:53	CAL ナビゲーター	3853	0
会員申込	2010-11-14 18:38:03	CAL ナビゲーター	3853	0

Title	Modified_Date	Author	列 D <sup>10</sup>	列 E <sup>11</sup>
会員申込	2010-11-14 18:36:07	CAL ナビゲーター	3853	0
会員申込	2010-11-14 18:35:59	CAL ナビゲーター	3853	0
会員申込	2010-11-14 18:28:47	CAL ナビゲーター	3853	3817
会員申込	2010-11-14 18:28:02	CAL ナビゲーター	36	36
行事申込	2010-11-21 11:47:15	CAL ナビゲーター	2702	0
行事申込	2010-11-20 11:37:27	CAL ドライバー	2702	0
行事申込	2010-11-15 01:27:10	CAL ナビゲーター	2702	0
行事申込	2010-11-15 01:21:25	CAL ナビゲーター	2702	0
行事申込	2010-11-14 18:47:27	CAL ナビゲーター	2702	0
行事申込	2010-11-14 18:47:13	CAL ナビゲーター	2702	2702
行事申込	2010-11-14 18:46:27	CAL ナビゲーター	0	0
メール申込のページを追加しました！	2010-12-18 07:23:36	CAL ナビゲーター	474	0
メール申込のページを追加しました！	2010-12-18 07:19:33	CAL ナビゲーター	474	0
メール申込のページを追加しました！	2010-11-21 16:05:17	CAL ナビゲーター	474	1
メール申込のページを追加しました！	2010-11-21 16:04:49	CAL ドライバー	473	151
メール申込のページを追加しました！	2010-11-20 12:26:03	CAL ドライバー	322	1
メール申込のページを追加しました！	2010-11-20 12:25:24	CAL ナビゲーター	321	0
メール申込のページを追加しました！	2010-11-20 12:24:12	CAL ナビゲーター	321	0
メール申込のページを追加しました！	2010-11-20 12:23:36	CAL ナビゲーター	321	0
メール申込のページを追加しました！	2010-11-20 12:22:15	CAL ナビゲーター	321	-2
メール申込のページを追加しました！	2010-11-20 12:22:03	CAL ナビゲーター	323	0
メール申込のページを追加しました！	2010-11-20 12:21:32	CAL ナビゲーター	323	9
メール申込のページを追加しました！	2010-11-20 12:20:54	CAL ナビゲーター	314	9
メール申込のページを追加しました！	2010-11-20 12:20:44	CAL ナビゲーター	305	5
メール申込のページを追加しました！	2010-11-20 12:19:37	CAL ナビゲーター	300	3
メール申込のページを追加しました！	2010-11-20 12:18:51	CAL ナビゲーター	297	297
お申し込み・お問い合わせ先	2010-12-18 06:27:37	CAL ドライバー	3807	0
お申し込み・お問い合わせ先	2010-12-18 06:27:33	CAL ナビゲーター	3807	12

Title	Modified_Date	Author	列 D <sup>10</sup>	列 E <sup>11</sup>
お申し込み・お問い合わせ先	2010-12-18 06:26:26	CAL ナビゲーター	3795	-10
お申し込み・お問い合わせ先	2010-12-18 06:24:25	CAL ナビゲーター	3805	26
お申し込み・お問い合わせ先	2010-12-18 06:20:36	CAL ナビゲーター	3779	0
お申し込み・お問い合わせ先	2010-12-18 06:15:47	CAL ナビゲーター	3779	5
お申し込み・お問い合わせ先	2010-12-18 06:14:25	CAL ナビゲーター	3774	38
お申し込み・お問い合わせ先	2010-12-18 06:11:42	CAL ナビゲーター	3736	0
お申し込み・お問い合わせ先	2010-12-18 06:10:24	CAL ナビゲーター	3736	25
お申し込み・お問い合わせ先	2010-12-18 06:04:27	CAL ナビゲーター	3711	0
お申し込み・お問い合わせ先	2010-12-18 06:03:00	CAL ナビゲーター	3711	0
お申し込み・お問い合わせ先	2010-12-18 06:01:57	CAL ナビゲーター	3711	0
お申し込み・お問い合わせ先	2010-12-18 06:00:55	CAL ナビゲーター	3711	9
お申し込み・お問い合わせ先	2010-12-18 05:59:57	CAL ナビゲーター	3702	21
お申し込み・お問い合わせ先	2010-12-18 05:58:37	CAL ナビゲーター	3681	5
お申し込み・お問い合わせ先	2010-12-18 05:56:57	CAL ナビゲーター	3676	-2
お申し込み・お問い合わせ先	2010-12-18 05:55:51	CAL ナビゲーター	3678	0
お申し込み・お問い合わせ先	2010-12-18 05:54:42	CAL ナビゲーター	3678	0
お申し込み・お問い合わせ先	2010-11-21 11:56:03	CAL ナビゲーター	3678	1
お申し込み・お問い合わせ先	2010-11-21 11:49:57	CAL ドライバー	3677	3677
お申し込み・お問い合わせ先	2010-11-21 11:48:21	CAL ドライバー	0	0
お申し込み・お問い合わせ先	2010-11-21 11:46:07	CAL ドライバー	0	0
お申し込み・お問い合わせ先	2010-11-21 11:45:52	CAL ドライバー	0	0
英語のページも作成します。	2010-12-18 07:24:07	CAL ナビゲーター	80	0
英語のページも作成します。	2010-12-07 19:51:27	CAL ナビゲーター	80	9
英語のページも作成します。	2010-12-07 19:06:59	CAL ナビゲーター	71	5
英語のページも作成します。	2010-12-07 19:04:37	CAL ナビゲーター	66	-10
英語のページも作成します。	2010-12-07 18:41:06	CAL ナビゲーター	76	5
英語のページも作成します。	2010-12-07 18:40:31	CAL ナビゲーター	71	0
英語のページも作成します。	2010-12-07 18:38:25	CAL ナビゲーター	71	71

Title	Modified_Date	Author	列 D <sup>10</sup>	列 E <sup>11</sup>
We will create English pages.	2010-12-07 19:48:46	CAL ナビゲーター	177	9
We will create English pages.	2010-12-07 19:48:18	CAL ナビゲーター	168	0
We will create English pages.	2010-12-07 19:48:10	CAL ナビゲーター	168	38
We will create English pages.	2010-12-07 19:05:55	CAL ナビゲーター	130	0
We will create English pages.	2010-12-07 18:50:54	CAL ナビゲーター	130	0
We will create English pages.	2010-12-07 18:49:21	CAL ナビゲーター	130	130
About us	2010-12-07 22:02:26	CAL ナビゲーター	0	0
About Us	2010-12-07 22:11:53	CAL ナビゲーター	0	0
About school	2010-12-07 19:09:50	CAL ナビゲーター	0	0
About	2010-12-07 19:09:13	CAL ナビゲーター	0	0
Lessons	2010-12-07 19:22:55	CAL ナビゲーター	0	0
Lesson	2010-12-07 19:12:58	CAL ナビゲーター	0	0
自動下書き	2010-12-07 19:12:38	CAL ナビゲーター	0	0
Ski Private Lesson	2010-12-07 19:17:28	CAL ナビゲーター	16	-952
Ski Private Lesson	2010-12-08 23:15:25	CAL ナビゲーター	968	968
Sli	2010-12-07 19:17:16	CAL ナビゲーター	0	0
Snowboard Private Lesson	2010-12-07 19:20:00	CAL ナビゲーター	16	16
Snowboard Private Lesson	2010-12-07 19:19:57	CAL ナビゲーター	0	0
Applications and Inquiries	2010-12-07 19:33:37	CAL ナビゲーター	16	16
Applications and Inquiries	2010-12-07 19:32:55	CAL ナビゲーター	0	0
FAQ	2010-12-07 19:34:38	CAL ナビゲーター	16	16
FAQ	2010-12-07 19:34:26	CAL ナビゲーター	0	0
Privacy Policy	2010-12-07 22:01:05	CAL ナビゲーター	16	16
PrPriPrivPrivaPrivaPrivPrivPriv	2010-12-07 22:00:18	CAL ナビゲーター	0	0
ブログ	2010-12-15 03:16:45	CAL ナビゲーター	927	4
ブログ	2010-12-15 03:15:46	CAL ナビゲーター	923	16
ブログ	2010-12-15 03:14:24	CAL ナビゲーター	907	0
ブログ	2010-12-15 03:13:33	CAL ナビゲーター	907	0



Title	Modified_Date	Author	列 D <sup>10</sup>	列 E <sup>11</sup>
ブログ	2010-12-15 03:12:48	CAL ナビゲーター	907	-79
ブログ	2010-12-13 02:17:03	CAL ナビゲーター	986	0
ブログ	2010-12-13 02:02:49	CAL ナビゲーター	986	0
ブログ	2010-12-13 02:00:32	CAL ナビゲーター	986	0
ブログ	2010-12-13 02:00:01	CAL ナビゲーター	986	0
ブログ	2010-12-13 01:57:41	CAL ナビゲーター	986	0
ブログ	2010-12-13 01:55:17	CAL ナビゲーター	986	0
ブログ	2010-12-13 01:52:56	CAL ナビゲーター	986	4
ブログ	2010-12-13 01:51:00	CAL ナビゲーター	982	54
ブログ	2010-12-13 01:49:47	CAL ナビゲーター	928	28
ブログ	2010-12-13 01:49:03	CAL ナビゲーター	900	0
ブログ	2010-12-13 01:40:10	CAL ナビゲーター	900	801
ブログ	2010-12-13 01:36:08	CAL ナビゲーター	99	99
特別プログラム	2010-12-13 02:50:57	CAL ナビゲーター	0	0
特別プログラム	2010-12-13 02:37:31	CAL ナビゲーター	0	0
特別プログラム	2010-12-13 02:37:24	CAL ナビゲーター	0	0
苗場たけのこやま山荘ブログ	2010-12-15 03:07:40	CAL ナビゲーター	213	0
苗場たけのこやま山荘ブログ	2010-12-15 03:06:59	CAL ナビゲーター	213	213
提携プログラム	2010-12-18 04:46:55	CAL ナビゲーター	0	0
提携プログラム	2010-12-18 04:44:12	CAL ナビゲーター	0	0
レッスン開始！	2010-12-18 06:59:57	CAL ナビゲーター	270	270
レッスン開始！	2010-12-18 07:13:45	CAL ナビゲーター	271	0
レッスン開始！	2010-12-18 07:07:53	CAL ナビゲーター	271	0
レッスン開始！	2010-12-18 07:07:01	CAL ナビゲーター	271	271
表示エラーのお詫び	2010-12-26 01:54:17	CAL ナビゲーター	144	0
表示エラーのお詫び	2010-12-26 01:54:08	CAL ナビゲーター	144	49
表示エラーのお詫び	2010-12-25 19:13:33	CAL ナビゲーター	95	0
表示エラーのお詫び	2010-12-25 19:12:22	CAL ナビゲーター	95	95

Title	Modified_Date	Author	列 D <sup>10</sup>	列 E <sup>11</sup>
naebass-background-2011-h4.jpg	2011-01-03 21:08:30	CAL ナビゲーター	70	70
自動下書き	2011-01-04 22:39:54	CAL ドライバー	0	0
自動下書き	2011-01-04 22:44:15	CAL ナビゲーター	0	0
	2011-01-06 00:31:38	CAL ドライバー	178	178
2010～2011 スキー 1 級合格者	2011-01-18 22:44:18	CAL ドライバー	23	-16
2010～2011 スキー 1 級合格者	2011-01-18 22:41:38	CAL ドライバー	39	0
2010～2011 スキー 1 級合格者	2011-01-18 22:36:51	CAL ドライバー	39	39
2010～2011 スキー 1 級合格者	2011-01-06 23:08:48	CAL ドライバー	0	0
2010～2011 スキー 1 級合格者	2011-01-06 22:59:49	CAL ドライバー	0	0
2010～2011 スキー 1 級	2011-01-06 22:57:42	CAL ドライバー	0	0
2010～2011 スキー 1 級	2011-01-06 22:57:00	CAL ドライバー	0	0
2010～2011 スキー 1 級	2011-01-06 22:20:26	CAL ドライバー	0	0
2010～2011 スキー 1 級	2011-01-06 22:20:11	CAL ドライバー	0	0
2010～2011 スキー 2 級合格者	2011-01-18 22:44:58	CAL ドライバー	23	0
2010～2011 スキー 2 級合格者	2011-01-18 22:46:06	CAL ドライバー	23	23
2010～2011 スキー 2 級合格者	2011-01-06 23:04:28	CAL ドライバー	0	0
2010～2011 スキー 2 級合格者	2011-01-06 23:03:35	CAL ドライバー	0	0
2010～2011 スキー 2 級合格者	2011-01-06 23:03:25	CAL ドライバー	0	0
2010～2011 スキー 3 級合格者	2011-01-06 23:06:18	CAL ドライバー	0	0
2010～2011 スキー 3 級合格者	2011-01-06 23:05:51	CAL ドライバー	0	0
2010～2011 スキー 3 級合格者	2011-01-06 23:05:37	CAL ドライバー	0	0
2010～2011 スキー 4 級合格者	2011-01-06 23:07:35	CAL ドライバー	0	0
2010～2011 スキー 4 級合格者	2011-01-06 23:07:01	CAL ドライバー	0	0
2010～2011 スキー 4 級合格者	2011-01-06 23:06:38	CAL ドライバー	0	0
2010～2011 スキー 5 級合格者	2011-01-06 23:08:19	CAL ドライバー	0	0
2010～2011 スキー 5 級合格者	2011-01-06 23:08:09	CAL ドライバー	0	0
2010～2011 スキー 5 級合格者	2011-01-06 23:07:54	CAL ドライバー	0	0
2010～2011 スノーボード 1 級合格者	2011-01-06 23:12:53	CAL ドライバー	0	0

Title	Modified_Date	Author	列 D <sup>10</sup>	列 E <sup>11</sup>
2010～2011 スノーボード1級合格者	2011-01-06 23:12:17	CAL ドライバー	0	0
2010～2011 スノーボード1級合格者	2011-01-06 23:11:52	CAL ドライバー	0	0
2010～2011 スノーボード1級合格者	2011-01-06 23:11:45	CAL ドライバー	0	0
2010～2011 スノーボード2級合格者	2011-01-06 23:13:41	CAL ドライバー	0	0
2010～2011 スノーボード2級合格者	2011-01-06 23:13:25	CAL ドライバー	0	0
2010～2011 スノーボード2級合格者	2011-01-06 23:13:16	CAL ドライバー	0	0
2010～2011 スノーボード3級合格者	2011-01-06 23:14:59	CAL ドライバー	0	0
2010～2011 スノーボード3級合格者	2011-01-06 23:14:43	CAL ドライバー	0	0
2010～2011 スノーボード3級合格者	2011-01-06 23:14:29	CAL ドライバー	0	0
2010～2011 スノーボード3級合格者	2011-01-06 23:14:02	CAL ドライバー	0	0
2010～2011 スノーボード4級合格者	2011-01-06 23:16:02	CAL ドライバー	0	0
2010～2011 スノーボード4級合格者	2011-01-06 23:15:50	CAL ドライバー	0	0
2010～2011 スノーボード4級合格者	2011-01-06 23:15:36	CAL ドライバー	0	0
2010～2011 スノーボード4級合格者	2011-01-06 23:15:17	CAL ドライバー	0	0
2010～2011 スノーボード5級合格者	2011-01-06 23:16:41	CAL ドライバー	0	0
2010～2011 スノーボード5級合格者	2011-01-06 23:16:26	CAL ドライバー	0	0
2010～2011 スノーボード5級合格者	2011-01-06 23:16:15	CAL ドライバー	0	0
2010～2011 ジュニア1級合格者	2011-01-06 23:19:28	CAL ドライバー	0	0
2010～2011 ジュニア1級合格者	2011-01-06 23:19:16	CAL ドライバー	0	0
2010～2011 ジュニア1級合格者	2011-01-06 23:19:01	CAL ドライバー	0	0
2010～2011 ジュニア2級合格者	2011-01-06 23:20:19	CAL ドライバー	0	0
2010～2011 ジュニア2級合格者	2011-01-06 23:20:05	CAL ドライバー	0	0
2010～2011 ジュニア2級合格者	2011-01-06 23:19:52	CAL ドライバー	0	0
2010～2011 ジュニア3級合格者	2011-01-06 23:21:09	CAL ドライバー	0	0
2010～2011 ジュニア3級合格者	2011-01-06 23:20:54	CAL ドライバー	0	0
2010～2011 ジュニア3級合格者	2011-01-06 23:20:40	CAL ドライバー	0	0
2010～2011 ジュニア4級合格者	2011-01-06 23:21:48	CAL ドライバー	0	0
2010～2011 ジュニア4級合格者	2011-01-06 23:21:35	CAL ドライバー	0	0

Title	Modified_Date	Author	列 D <sup>10</sup>	列 E <sup>11</sup>
2010～2011 ジュニア 4 級合格者	2011-01-06 23:21:26	CAL ドライバー	0	0
2010～2011 ジュニア 5 級合格者	2011-01-06 23:22:26	CAL ドライバー	0	0
2010～2011 ジュニア 5 級合格者	2011-01-06 23:22:14	CAL ドライバー	0	0
2010～2011 ジュニア 5 級合格者	2011-01-06 23:22:01	CAL ドライバー	0	0

## 9.4. 効果の検証 2（企業経営や業務の戦略）

### 9.4.1. 属性情報

#### (1) 性別

Table 33 “Questionnaire – Attribute Information - Gender” (Original)

SEX	性別 単一回答	N	%
1	男性	340	80.8
2	女性	81	19.2
	全体	421	100.0

#### (2) 年齢

Table 34 “Questionnaire – Attribute Information - Age” (Original)

AGEID	年齢 単一回答	N	%
1	12 才未満	0	0.0
2	12 才～19 才	0	0.0
3	20 才～24 才	4	1.0
4	25 才～29 才	40	9.5
5	30 才～34 才	35	8.3
6	35 才～39 才	33	7.8

7	40才～44才	49	11.6
8	45才～49才	67	15.9
9	50才～54才	65	15.4
10	55才～59才	44	10.5
11	60才以上	84	20.0
	全体	421	100.0

### (3) 都道府県

Table 35 “Questionnaire – Attribute Information - Prefecture” (Original)

PREFECTURE	都道府県	N	%
	単一回答		
1	北海道	31	7.4
2	青森県	0	0.0
3	岩手県	2	0.5
4	宮城県	6	1.4
5	秋田県	3	0.7
6	山形県	1	0.2
7	福島県	4	1.0
8	茨城県	4	1.0
9	栃木県	7	1.7
10	群馬県	2	0.5
11	埼玉県	17	4.0
12	千葉県	16	3.8
13	東京都	85	20.2
14	神奈川県	39	9.3
15	新潟県	5	1.2
16	富山県	2	0.5
17	石川県	7	1.7

18	福井県	2	0.5
19	山梨県	0	0.0
20	長野県	5	1.2
21	岐阜県	5	1.2
22	静岡県	11	2.6
23	愛知県	27	6.4
24	三重県	5	1.2
25	滋賀県	4	1.0
26	京都府	6	1.4
27	大阪府	40	9.5
28	兵庫県	16	3.8
29	奈良県	1	0.2
30	和歌山県	3	0.7
31	鳥取県	2	0.5
32	島根県	2	0.5
33	岡山県	7	1.7
34	広島県	14	3.3
35	山口県	2	0.5
36	徳島県	3	0.7
37	香川県	1	0.2
38	愛媛県	2	0.5
39	高知県	0	0.0
40	福岡県	21	5.0
41	佐賀県	1	0.2
42	長崎県	3	0.7
43	熊本県	2	0.5
44	大分県	2	0.5
45	宮崎県	2	0.5
46	鹿児島県	1	0.2

47	沖縄県	0	0.0
	全体	421	100.0

#### (4) 地域

Table 36 “Questionnaire – Attribute Information - Area” (Original)

AREA	地域	N	%
	単一回答		
1	北海道	31	7.4
2	東北地方	16	3.8
3	関東地方	170	40.4
4	中部地方	69	16.4
5	近畿地方	70	16.6
6	中国地方	27	6.4
7	四国地方	6	1.4
8	九州地方	32	7.6
	全体	421	100.0

#### (5) 未既婚

Table 37 “Questionnaire – Attribute Information - Married” (Original)

MARRIED	未既婚	N	%
	単一回答		
1	未婚	116	27.6
2	既婚	305	72.4
	全体	421	100.0

## (6) 子供の有無

Table 38 “Questionnaire – Attribute Information - Child” (Original)

CHILD	子供の有無	N	%
	単一回答		
1	子供なし	147	34.9
2	子供あり	274	65.1
	全体	421	100.0

## (7) 世帯年収

Table 39 “Questionnaire – Attribute Information – Household Income” (Original)

HINCOME	世帯年収	N	%
	単一回答		
1	200 万未満	6	1.4
2	200～400 万未満	45	10.7
3	400～600 万未満	90	21.4
4	600～800 万未満	84	20.0
5	800～1000 万未満	69	16.4
6	1000～1200 万未満	35	8.3
7	1200～1500 万未満	27	6.4
8	1500～2000 万未満	12	2.9
9	2000 万円以上	9	2.1
10	わからない	24	5.7
	無回答	20	4.8
	全体	421	100.0



## (8) 職業

Table 40 “Questionnaire – Attribute Information – Job” (Original)

JOB	職業	N	%
	単一回答		
1	公務員	11	2.6
2	経営者・CALドライバー	206	48.9
3	会社員(事務系)	48	11.4
4	会社員(技術系)	46	10.9
5	会社員(その他)	33	7.8
6	自営業	44	10.5
7	自由業	8	1.9
8	専業主婦(主夫)	15	3.6
9	パート・アルバイト	5	1.2
10	学生	1	0.2
11	その他	2	0.5
12	無職	2	0.5
	全体	421	100.0

## (9) 学生種別

Table 41 “Questionnaire – Attribute Information – Student” (Original)

STUDENT	学生種別	N	%
	単一回答		
1	小学生	0	0.0
2	中学生	0	0.0
3	高校生・高専生	0	0.0
4	専門学校生	0	0.0
5	短大生	0	0.0

6	大学生	1	100.0
7	大学院生	0	0.0
8	その他学生	0	0.0
	全体	1	100.0

## (10) 割付セル

Table 42 “Questionnaire – Attribute Information – Cell” (Original)

CELL	割付セル	N	%
	単一回答		
1	経営者・CALドライバー	206	48.9
2	起業 20代 有職	38	9.0
3	起業 30代 有職	38	9.0
4	起業 40代 有職	38	9.0
5	起業 50代 有職	38	9.0
6	起業 60歳以上 有職	38	9.0
7	起業 20代 その他	5	1.2
8	起業 30代 その他	5	1.2
9	起業 40代 その他	5	1.2
10	起業 50代 その他	5	1.2
11	起業 60歳以上 その他	5	1.2
	全体	421	100.0

### 9.4.2. 質問内容・回答結果

#### (1) 業種

Table 43 “Questionnaire – Question1 & Answers” (Original)

<b>Q1</b>	あなたは現在、以下のどの業種に従事していらっしゃいますか？※複数の業種に従事している方は、あてはまるものをすべてお選びください。		
-----------	--	--	--

	複数回答	N	%
1	農林水産業	2	0.5
2	鉱業	1	0.2
3	建設業	39	9.3
4	製造業(繊維)	2	0.5
5	製造業(化学)	5	1.2
6	製造業(鉄鋼金属)	3	0.7
7	製造業(機械)	7	1.7
8	製造業(食料品)	7	1.7
9	製造業(出版印刷)	7	1.7
10	製造業(その他)	34	8.1
11	卸売業	33	7.8
12	小売業	34	8.1
13	料理・飲食・旅館業	10	2.4
14	金融・保険業	17	4.0
15	不動産業	29	6.9
16	運輸・通信・公益事業	7	1.7
17	ソフトウェア・情報サービス業	36	8.6
18	教育・学習支援業	20	4.8
19	学術・開発研究機関	0	0.0
20	医療・福祉業	21	5.0
21	その他のサービス業	90	21.4
22	上記以外【    】	31	7.4
23	専業主婦(主夫)	15	3.6
24	学生(社会人学生を除く)	1	0.2
25	無職	2	0.5
	全体	421	100.0

## (2) 従業員数

Table 44 “Questionnaire – Question2 & Answers” (Original)

Q2	あなたの主な勤務先の会社の規模はどれぐらいですか？※本社、支社、営業所などを含めた会社全体の従業員数をお答えください。※現在会社に属していない方でも、以前に属した経験のある方は、直近で属した会社についてお教えてください。	単一回答	N	%
1	10 人未満		155	36.8
2	10 人～99 人		129	30.6
3	100 人～999 人		62	14.7
4	1000 人以上		60	14.3
5	会社に属したことはないが、働いている(働いたことがある)		12	2.9
6	今まで働いたことはない		3	0.7
	全体		421	100.0

## (3) 年商

Table 45 “Questionnaire – Question3 & Answers” (Original)

Q3	あなたの主な勤務先の会社の年商はどれぐらいですか？※現在会社に属していない方でも、以前に属した経験のある方は、直近で属した会社についてお教えてください。	単一回答	N	%
1	100 億円以上		53	13.1
2	10 億円以上		51	12.6
3	1 億円以上		84	20.7
4	1 億円未満		162	39.9
5	分からない		56	13.8
	全体		406	100.0

#### (4) 業務経験

Table 46 “Questionnaire – Question4 & Answers” (Original)

Q4	あなたは、以下の業務を経験されたことがありますか？経験がある方は、何年ぐらい経験されていますか？これまでの経験の合計の期間をそれぞれお答えください。※業種は問いません。 単一回答	全体	1	2	3	4	5	6
			経験したことがない	合計1年未満	合計1年以上～3年未満	合計3年以上～6年未満	合計6年以上～10年未満	合計10年以上
1	経営	418 100.0	184 44.0	9 2.2	38 9.1	41 9.8	42 10.0	104 24.9
2	営業・販売	418 100.0	132 31.6	23 5.5	36 8.6	50 12.0	36 8.6	141 33.7
3	企画・マーケティング	418 100.0	208 49.8	26 6.2	44 10.5	26 6.2	26 6.2	88 21.1
4	経理・会計	418 100.0	226 54.1	26 6.2	30 7.2	31 7.4	30 7.2	75 17.9
5	総務・人事	418 100.0	222 53.1	21 5.0	37 8.9	31 7.4	27 6.5	80 19.1
6	法務・知財	418 100.0	323 77.3	15 3.6	13 3.1	9 2.2	12 2.9	46 11.0
7	物流・調達	418 100.0	284 67.9	13 3.1	21 5.0	17 4.1	21 5.0	62 14.8
8	生産・製造	418 100.0	305 73.0	16 3.8	17 4.1	15 3.6	12 2.9	53 12.7
9	IT	418 100.0	284 67.9	9 2.2	15 3.6	21 5.0	24 5.7	65 15.6
10	研究	418	337	7	19	9	9	37

		100.0	80.6	1.7	4.5	2.2	2.2	8.9
11	その他専門職	418	277	7	17	19	18	80
		100.0	66.3	1.7	4.1	4.5	4.3	19.1
12	上記以外【 】	418	381	7	6	6	2	16
		100.0	91.1	1.7	1.4	1.4	0.5	3.8

## (5) 戦略への関与度

Table 47 “Questionnaire – Question5 & Answers” (Original)

Q5	あなたは経営者・CALドライバーとして、以下の分野のビジネス戦略を立てるときに、どの程度関与していますか？最も多いケースを、それぞれお答えください。 単一回答	全体	1	2	3	4
			プランニングから積極的に関わって意見を出す	ある程度任せながら自分の意見を入れる	担当者に任せて最後に承認する	ビジネス戦略立案に関与しない
1	経営	206	149	25	11	21
		100.0	72.3	12.1	5.3	10.2
2	営業・販売	206	116	50	15	25
		100.0	56.3	24.3	7.3	12.1
3	企画・マーケティング	206	111	41	18	36
		100.0	53.9	19.9	8.7	17.5
4	経理・会計	206	85	37	51	33
		100.0	41.3	18.0	24.8	16.0
5	総務・人事	206	95	40	34	37
		100.0	46.1	19.4	16.5	18.0
6	法務・知財	206	64	29	45	68
		100.0	31.1	14.1	21.8	33.0
7	物流・調達	206	73	31	40	62

		100.0	35.4	15.0	19.4	30.1
8	生産・製造	206	60	27	31	88
		100.0	29.1	13.1	15.0	42.7
9	IT	206	87	29	28	62
		100.0	42.2	14.1	13.6	30.1
10	研究	206	56	24	24	102
		100.0	27.2	11.7	11.7	49.5
11	その他専門職	206	59	21	27	99
		100.0	28.6	10.2	13.1	48.1
12	上記以外【 】	206	30	11	18	147
		100.0	14.6	5.3	8.7	71.4

## (6) 業務システム全般の経験

Table 48 “Questionnaire – Question6 & Answers” (Original)

			1	2	3	4	5	6
Q6	<p>あなたは、業務システムに関して、次の工程で関与した経験がございますか？</p> <p>経験がある方は、何年ぐらい経験されていますか？これまでの経験の合計の期間をそれぞれお答えください。※ここで言う「運用」とは、利用者がシステムをつつがなく利用できるように システムの稼働環境を維持することを指します。※業種は問いません。</p> <p>単一回答</p>	全体	経験したことがない	合計 1 年未満	合計 1 年以上 ～ 3年 未満	合計 3 年以上 ～ 6年 未満	合計 6 年以上 ～ 10 年未満	合計 10 年以上
1	開発	418	276	16	23	28	7	68
		100.0	66.0	3.8	5.5	6.7	1.7	16.3
2	保守	418	278	12	22	27	15	64

		100.0	66.5	2.9	5.3	6.5	3.6	15.3
3	運用	418	250	14	29	28	22	75
		100.0	59.8	3.3	6.9	6.7	5.3	17.9

### (7) 個々の業務システムの経験

Table 49 “Questionnaire – Question7 & Answers” (Original)

Q7	あなたは、以下のそれぞれの業務システムに関して、開発・保守・運用のいずれかに関与したことがありますか？ある方は、その関与の仕方をお教えてください。(複数回答可)※ここで言う「運用」とは、利用者がシステムをつつがなく利用できるようにシステムの稼働環境を維持することを指します。※業種は問いません。	全体	1	2	3	4	5	6
			マネージャーとして	メンバーとして	社内/他部署に委託	他社に委託	その他	関与したことはない
1	顧客管理	204 100.0	108 52.9	45 22.1	7 3.4	4 2.0	3 1.5	50 24.5
2	会計管理	204 100.0	77 37.7	23 11.3	24 11.8	12 5.9	4 2.0	71 34.8
3	人事管理	204 100.0	89 43.6	38 18.6	12 5.9	2 1.0	2 1.0	68 33.3
4	知財管理	204 100.0	56 27.5	19 9.3	13 6.4	4 2.0	4 2.0	111 54.4
5	物流・調達・販売管理	204 100.0	84 41.2	38 18.6	13 6.4	4 2.0	3 1.5	69 33.8
6	在庫管理	204 100.0	84 41.2	40 19.6	15 7.4	2 1.0	2 1.0	70 34.3



7	設備管理	204	69	32	11	4	5	86
		100.0	33.8	15.7	5.4	2.0	2.5	42.2
8	不動産管理	204	54	21	11	4	5	112
		100.0	26.5	10.3	5.4	2.0	2.5	54.9
9	その他【 】	204	25	8	6	0	7	159
		100.0	12.3	3.9	2.9	0.0	3.4	77.9

### (8) 仕事の取り組み方

Table 50 “Questionnaire – Question8 & Answers” (Original)

Q8	あなたは以下の仕事に取り組む時、どんな基準に従っていますか？主にあてはまるものをそれぞれお選びください。 ※現在働いていない方も、仕事に取り組むことを想定してお答えください。 単一回答	全体	1	2	3	4
			これまでの自分の経験	個々のマニュアル	全体的に標準化されたプロセス	あてはまるものはない
1	日常的で決められた業務	421 100.0	260 61.8	51 12.1	89 21.1	21 5.0
2	プロジェクトなど有期性の業務	421 100.0	232 55.1	43 10.2	72 17.1	74 17.6
3	検証・振り返り	421 100.0	221 52.5	50 11.9	78 18.5	72 17.1
4	改善活動	421 100.0	266 63.2	37 8.8	57 13.5	61 14.5

### (9) システムの仕組みを学習するアプローチ

Table 51 “Questionnaire – Question9 & Answers” (Original)

Q9			1	2	3	4	5	6	7	8
----	--	--	---	---	---	---	---	---	---	---

自社の業務システムの仕組みを知ろうと思った時、どうしますか？あてはまるものを、それぞれいくつでもお答えください。※現在企業に属していない方も、企業に属することを想定してお答えください。		全体	知りたくはない	詳しい人から話を聞く	ドキュメントを読む	デモンストレーションを見る	勉強会に参加する	運用に参加する	開発・保守に参加する	その他
1	システムの規模が小さい場合	421 100.0	19 4.5	262 62.2	123 29.2	69 16.4	92 21.9	94 22.3	74 17.6	35 8.3
2	システムの規模が大きい場合	421 100.0	25 5.9	241 57.2	116 27.6	102 24.2	151 35.9	97 23.0	78 18.5	37 8.8

### (10) システムの開発サポート

Table 52 “Questionnaire – Question10 & Answers” (Original)

Q10	プロの開発者のサポートを受けられるのであれば、自社のシステムの開発・保守をやってみていと思いますか？あてはまるものを、それぞれお答えください。※現在企業に属していない方も、企業に属することを想定してお答えください。※以下の選択肢の中よりお答えください。	全体	1	2	3	4
			やってみたいと思わない	自分一人でやってみたい	自分を含む社内の皆でやってみたい	自分を除く社内の皆でやってもらいたい
1	システムの規模が小さい場合	421 100.0	138 32.8	116 27.6	147 34.9	20 4.8
2	システムの規模が大きい場合	421 100.0	167 39.7	47 11.2	188 44.7	19 4.5

### 9.4.3. 分析結果

Table 53 “Relativity to the activeness or passiveness for the development of management strategies with the experience of the development and/or maintenance and/or operation of various business system(s)” (Original)

戦略を策定する 対象の職種	業務システ ムの種類	※1	回 答 数	戦略の策定に 積極的			戦略の策定に 消極的		
				※2	※3	合計	※4	※5	合計
経営	顧客管理	○	92	<b>78.3</b>	12.0	<b>90.2</b>	<b>5.4</b>	4.3	9.8
	システム	△×	114	67.5	<b>12.3</b>	79.8	5.3	<b>14.9</b>	<b>20.2</b>
	会計管理	○	63	<b>81.0</b>	9.5	<b>90.5</b>	<b>6.3</b>	3.2	9.5
	システム	△×	143	68.5	<b>13.3</b>	81.8	4.9	<b>13.3</b>	<b>18.2</b>
	人事管理	○	85	<b>84.7</b>	9.4	<b>94.1</b>	3.5	2.4	5.9
	システム	△×	121	63.6	<b>14.0</b>	77.7	<b>6.6</b>	<b>15.7</b>	<b>22.3</b>
	知財管理	○	54	<b>87.0</b>	7.4	<b>94.4</b>	3.7	1.9	5.6
	システム	△×	152	67.1	<b>13.8</b>	80.9	<b>5.9</b>	<b>13.2</b>	<b>19.1</b>
	SCM	○	74	<b>82.4</b>	9.5	<b>91.9</b>	4.1	4.1	8.1
	システム	△×	132	66.7	<b>13.6</b>	80.3	<b>6.1</b>	<b>13.6</b>	<b>19.7</b>
	在庫管理	○	76	<b>78.9</b>	10.5	<b>89.5</b>	5.3	5.3	10.5
	システム	△×	130	68.5	<b>13.1</b>	81.5	<b>5.4</b>	<b>13.1</b>	<b>18.5</b>
	設備管理	○	70	<b>81.4</b>	<b>12.9</b>	<b>94.3</b>	4.3	1.4	5.7
	システム	△×	136	67.6	11.8	79.4	<b>5.9</b>	<b>14.7</b>	<b>20.6</b>
	不動産管理	○	57	<b>89.5</b>	7.0	<b>96.5</b>	3.5	0.0	3.5
	システム	△×	149	65.8	<b>14.1</b>	79.9	<b>6.0</b>	<b>14.1</b>	<b>20.1</b>
その他の	○	23	<b>91.3</b>	8.7	<b>100.0</b>	0.0	0.0	0.0	
システム	△×	183	69.9	<b>12.6</b>	82.5	<b>6.0</b>	<b>11.5</b>	<b>17.5</b>	
営業・販売	顧客管理	○	92	<b>66.3</b>	21.7	<b>88.0</b>	4.3	7.6	12.0
	システム	△×	114	48.2	<b>26.3</b>	74.6	<b>9.6</b>	<b>15.8</b>	<b>25.4</b>
	会計管理	○	63	<b>66.7</b>	19.0	<b>85.7</b>	4.8	9.5	14.3

	システム	△×	143	51.7	<b>26.6</b>	78.3	<b>8.4</b>	<b>13.3</b>	<b>21.7</b>
	人事管理	○	85	<b>69.4</b>	20.0	<b>89.4</b>	3.5	7.1	10.6
	システム	△×	121	47.1	<b>27.3</b>	74.4	<b>9.9</b>	<b>15.7</b>	<b>25.6</b>
	知財管理	○	54	<b>68.5</b>	22.2	<b>90.7</b>	1.9	7.4	9.3
	システム	△×	152	52.0	<b>25.0</b>	77.0	<b>9.2</b>	<b>13.8</b>	<b>23.0</b>
	SCM	○	74	<b>67.6</b>	23.0	<b>90.5</b>	4.1	5.4	9.5
	システム	△×	132	50.0	<b>25.0</b>	75.0	<b>9.1</b>	<b>15.9</b>	<b>25.0</b>
	在庫管理	○	76	<b>64.5</b>	<b>25.0</b>	<b>89.5</b>	3.9	6.6	10.5
	システム	△×	130	51.5	23.8	75.4	<b>9.2</b>	<b>15.4</b>	<b>24.6</b>
	設備管理	○	70	<b>67.1</b>	24.3	<b>91.4</b>	4.3	4.3	8.6
	システム	△×	136	50.7	24.3	75.0	<b>8.8</b>	<b>16.2</b>	<b>25.0</b>
	不動産管理	○	57	<b>71.9</b>	21.1	<b>93.0</b>	5.3	1.8	7.0
	システム	△×	149	50.3	<b>25.5</b>	75.8	<b>8.1</b>	<b>16.1</b>	<b>24.2</b>
	その他の	○	23	<b>82.6</b>	17.4	<b>100.0</b>	0.0	0.0	0.0
	システム	△×	183	53.0	<b>25.1</b>	78.1	<b>8.2</b>	<b>13.7</b>	<b>21.9</b>
企画・マーケティング	顧客管理	○	92	<b>60.9</b>	<b>20.7</b>	<b>81.5</b>	7.6	10.9	18.5
	システム	△×	114	48.2	19.3	67.5	<b>9.6</b>	<b>22.8</b>	<b>32.5</b>
	会計管理	○	63	<b>68.3</b>	12.7	<b>81.0</b>	6.3	12.7	19.0
	システム	△×	143	47.6	<b>23.1</b>	70.6	<b>9.8</b>	<b>19.6</b>	<b>29.4</b>
	人事管理	○	85	<b>65.9</b>	16.5	<b>82.4</b>	7.1	10.6	17.6
	システム	△×	121	45.5	<b>22.3</b>	67.8	<b>9.9</b>	<b>22.3</b>	<b>32.2</b>
	知財管理	○	54	<b>70.4</b>	16.7	<b>87.0</b>	5.6	7.4	13.0
	システム	△×	152	48.0	<b>21.1</b>	69.1	<b>9.9</b>	<b>21.1</b>	<b>30.9</b>
	SCM	○	74	<b>64.9</b>	17.6	<b>82.4</b>	8.1	9.5	17.6
	システム	△×	132	47.7	<b>21.2</b>	68.9	<b>9.1</b>	<b>22.0</b>	<b>31.1</b>
	在庫管理	○	76	<b>60.5</b>	19.7	<b>80.3</b>	7.9	11.8	19.7
	システム	△×	130	50.0	<b>20.0</b>	70.0	<b>9.2</b>	<b>20.8</b>	<b>30.0</b>
		設備管理	○	70	<b>65.7</b>	<b>20.0</b>	<b>85.7</b>	8.6	5.7

	システム	△×	136	47.8	19.9	67.6	<b>8.8</b>	<b>23.5</b>	<b>32.4</b>
	不動産管理	○	57	<b>66.7</b>	19.3	<b>86.0</b>	<b>8.8</b>	5.3	14.0
	システム	△×	149	49.0	<b>20.1</b>	69.1	8.7	<b>22.1</b>	<b>30.9</b>
	その他の	○	23	<b>78.3</b>	17.4	<b>95.7</b>	4.3	0.0	4.3
	システム	△×	183	50.8	<b>20.2</b>	71.0	<b>9.3</b>	<b>19.7</b>	<b>29.0</b>
<b>経理・会計</b>	顧客管理	○	92	<b>47.8</b>	13.0	<b>60.9</b>	<b>27.2</b>	12.0	39.1
	システム	△×	114	36.0	<b>21.9</b>	57.9	22.8	<b>19.3</b>	<b>42.1</b>
	会計管理	○	63	<b>66.7</b>	12.7	<b>79.4</b>	15.9	4.8	20.6
	システム	△×	143	30.1	<b>20.3</b>	50.3	<b>28.7</b>	<b>21.0</b>	<b>49.7</b>
	人事管理	○	85	<b>55.3</b>	14.1	<b>69.4</b>	22.4	8.2	30.6
	システム	△×	121	31.4	<b>20.7</b>	52.1	<b>26.4</b>	<b>21.5</b>	<b>47.9</b>
	知財管理	○	54	<b>59.3</b>	<b>18.5</b>	<b>77.8</b>	20.4	1.9	22.2
	システム	△×	152	34.9	17.8	52.6	<b>26.3</b>	<b>21.1</b>	<b>47.4</b>
	SCM	○	74	<b>54.1</b>	14.9	<b>68.9</b>	23.0	8.1	31.1
	システム	△×	132	34.1	<b>19.7</b>	53.8	<b>25.8</b>	<b>20.5</b>	<b>46.2</b>
	在庫管理	○	76	<b>51.3</b>	15.8	<b>67.1</b>	22.4	10.5	32.9
	システム	△×	130	35.4	<b>19.2</b>	54.6	<b>26.2</b>	<b>19.2</b>	<b>45.4</b>
	設備管理	○	70	<b>52.9</b>	15.7	<b>68.6</b>	22.9	8.6	31.4
	システム	△×	136	35.3	<b>19.1</b>	54.4	<b>25.7</b>	<b>19.9</b>	<b>45.6</b>
	不動産管理	○	57	<b>57.9</b>	15.8	<b>73.7</b>	22.8	3.5	26.3
	システム	△×	149	34.9	<b>18.8</b>	53.7	<b>25.5</b>	<b>20.8</b>	<b>46.3</b>
その他の	○	23	<b>56.5</b>	<b>26.1</b>	<b>82.6</b>	13.0	4.3	17.4	
システム	△×	183	39.3	16.9	56.3	<b>26.2</b>	<b>17.5</b>	<b>43.7</b>	
<b>総務・人事</b>	顧客管理	○	92	<b>53.3</b>	17.4	<b>70.7</b>	<b>17.4</b>	12.0	29.3
	システム	△×	114	40.4	<b>21.1</b>	61.4	15.8	<b>22.8</b>	<b>38.6</b>
	会計管理	○	63	<b>65.1</b>	19.0	<b>84.1</b>	9.5	6.3	15.9
	システム	△×	143	37.8	<b>19.6</b>	57.3	<b>19.6</b>	<b>23.1</b>	<b>42.7</b>
	人事管理	○	85	<b>60.0</b>	<b>21.2</b>	<b>81.2</b>	11.8	7.1	18.8

	システム	△×	121	36.4	18.2	54.5	<b>19.8</b>	<b>25.6</b>	<b>45.5</b>
	知財管理	○	54	<b>63.0</b>	<b>20.4</b>	<b>83.3</b>	13.0	3.7	16.7
	システム	△×	152	40.1	19.1	59.2	<b>17.8</b>	<b>23.0</b>	<b>40.8</b>
	SCM	○	74	<b>60.8</b>	14.9	<b>75.7</b>	16.2	8.1	24.3
	システム	△×	132	37.9	<b>22.0</b>	59.8	<b>16.7</b>	<b>23.5</b>	<b>40.2</b>
	在庫管理	○	76	<b>53.9</b>	17.1	<b>71.1</b>	<b>17.1</b>	11.8	28.9
	システム	△×	130	41.5	<b>20.8</b>	62.3	16.2	<b>21.5</b>	<b>37.7</b>
	設備管理	○	70	<b>58.6</b>	18.6	<b>77.1</b>	<b>17.1</b>	5.7	22.9
	システム	△×	136	39.7	<b>19.9</b>	59.6	16.2	<b>24.3</b>	<b>40.4</b>
	不動産管理	○	57	<b>64.9</b>	<b>21.1</b>	<b>86.0</b>	12.3	1.8	14.0
	システム	△×	149	38.9	18.8	57.7	<b>18.1</b>	<b>24.2</b>	<b>42.3</b>
	その他の	○	23	<b>60.9</b>	<b>21.7</b>	<b>82.6</b>	<b>17.4</b>	0.0	17.4
	システム	△×	183	44.3	19.1	63.4	16.4	<b>20.2</b>	<b>36.6</b>
<b>法務・知財</b>	顧客管理	○	92	<b>40.2</b>	12.0	<b>52.2</b>	<b>23.9</b>	23.9	47.8
	システム	△×	114	23.7	<b>15.8</b>	39.5	20.2	<b>40.4</b>	<b>60.5</b>
	会計管理	○	63	<b>54.0</b>	9.5	<b>63.5</b>	15.9	20.6	36.5
	システム	△×	143	21.0	<b>16.1</b>	37.1	<b>24.5</b>	<b>38.5</b>	<b>62.9</b>
	人事管理	○	85	<b>42.4</b>	<b>15.3</b>	<b>57.6</b>	<b>23.5</b>	18.8	42.4
	システム	△×	121	23.1	13.2	36.4	20.7	<b>43.0</b>	<b>63.6</b>
	知財管理	○	54	<b>55.6</b>	<b>18.5</b>	<b>74.1</b>	20.4	5.6	25.9
	システム	△×	152	22.4	12.5	34.9	<b>22.4</b>	<b>42.8</b>	<b>65.1</b>
	SCM	○	74	<b>43.2</b>	12.2	<b>55.4</b>	<b>23.0</b>	21.6	44.6
	システム	△×	132	24.2	<b>15.2</b>	39.4	21.2	<b>39.4</b>	<b>60.6</b>
	在庫管理	○	76	<b>42.1</b>	11.8	<b>53.9</b>	<b>22.4</b>	23.7	46.1
	システム	△×	130	24.6	<b>15.4</b>	40.0	21.5	<b>38.5</b>	<b>60.0</b>
	設備管理	○	70	<b>42.9</b>	<b>14.3</b>	<b>57.1</b>	<b>24.3</b>	18.6	42.9
	システム	△×	136	25.0	14.0	39.0	20.6	<b>40.4</b>	<b>61.0</b>
	不動産管理	○	57	<b>50.9</b>	<b>17.5</b>	<b>68.4</b>	21.1	10.5	31.6

	システム	△×	149	23.5	12.8	36.2	<b>22.1</b>	<b>41.6</b>	<b>63.8</b>
	その他の	○	23	<b>56.5</b>	<b>17.4</b>	<b>73.9</b>	13.0	13.0	26.1
	システム	△×	183	27.9	13.7	41.5	<b>23.0</b>	<b>35.5</b>	<b>58.5</b>
<b>物流・調達</b>	顧客管理	○	92	<b>43.5</b>	<b>18.5</b>	<b>62.0</b>	<b>20.7</b>	17.4	38.0
	システム	△×	114	28.9	12.3	41.2	18.4	<b>40.4</b>	<b>58.8</b>
	会計管理	○	63	<b>52.4</b>	11.1	<b>63.5</b>	19.0	17.5	36.5
	システム	△×	143	28.0	<b>16.8</b>	44.8	<b>19.6</b>	<b>35.7</b>	<b>55.2</b>
	人事管理	○	85	<b>48.2</b>	12.9	<b>61.2</b>	<b>21.2</b>	17.6	38.8
	システム	△×	121	26.4	<b>16.5</b>	43.0	18.2	<b>38.8</b>	<b>57.0</b>
	知財管理	○	54	<b>55.6</b>	<b>18.5</b>	<b>74.1</b>	18.5	7.4	25.9
	システム	△×	152	28.3	13.8	42.1	<b>19.7</b>	<b>38.2</b>	<b>57.9</b>
	SCM	○	74	<b>52.7</b>	<b>16.2</b>	<b>68.9</b>	16.2	14.9	31.1
	システム	△×	132	25.8	14.4	40.2	<b>21.2</b>	<b>38.6</b>	<b>59.8</b>
	在庫管理	○	76	<b>46.1</b>	<b>19.7</b>	<b>65.8</b>	18.4	15.8	34.2
	システム	△×	130	29.2	12.3	41.5	<b>20.0</b>	<b>38.5</b>	<b>58.5</b>
	設備管理	○	70	<b>50.0</b>	<b>18.6</b>	<b>68.6</b>	<b>20.0</b>	11.4	31.4
	システム	△×	136	27.9	13.2	41.2	19.1	<b>39.7</b>	<b>58.8</b>
	不動産管理	○	57	<b>57.9</b>	14.0	<b>71.9</b>	<b>21.1</b>	7.0	28.1
	システム	△×	149	26.8	<b>15.4</b>	42.3	18.8	<b>38.9</b>	<b>57.7</b>
	その他の	○	23	<b>65.2</b>	<b>17.4</b>	<b>82.6</b>	8.7	8.7	17.4
	システム	△×	183	31.7	14.8	46.4	<b>20.8</b>	<b>32.8</b>	<b>53.6</b>
<b>生産・製造</b>	顧客管理	○	92	<b>37.0</b>	<b>16.3</b>	<b>53.3</b>	<b>17.4</b>	29.3	46.7
	システム	△×	114	22.8	10.5	33.3	13.2	<b>53.5</b>	<b>66.7</b>
	会計管理	○	63	<b>38.1</b>	<b>14.3</b>	<b>52.4</b>	14.3	33.3	47.6
	システム	△×	143	25.2	12.6	37.8	<b>15.4</b>	<b>46.9</b>	<b>62.2</b>
	人事管理	○	85	<b>37.6</b>	<b>14.1</b>	<b>51.8</b>	<b>17.6</b>	30.6	48.2
	システム	△×	121	23.1	12.4	35.5	13.2	<b>51.2</b>	<b>64.5</b>
	知財管理	○	54	<b>42.6</b>	<b>18.5</b>	<b>61.1</b>	<b>16.7</b>	22.2	38.9

	システム	△×	152	24.3	11.2	35.5	14.5	<b>50.0</b>	<b>64.5</b>	
	SCM	○	74	<b>41.9</b>	<b>14.9</b>	<b>56.8</b>	13.5	29.7	43.2	
	システム	△×	132	22.0	12.1	34.1	<b>15.9</b>	<b>50.0</b>	<b>65.9</b>	
	在庫管理	○	76	<b>40.8</b>	<b>17.1</b>	<b>57.9</b>	14.5	27.6	42.1	
	システム	△×	130	22.3	10.8	33.1	<b>15.4</b>	<b>51.5</b>	<b>66.9</b>	
	設備管理	○	70	<b>44.3</b>	<b>17.1</b>	<b>61.4</b>	<b>15.7</b>	22.9	38.6	
	システム	△×	136	21.3	11.0	32.4	14.7	<b>52.9</b>	<b>67.6</b>	
	不動産管理	○	57	<b>47.4</b>	<b>17.5</b>	<b>64.9</b>	<b>15.8</b>	19.3	35.1	
	システム	△×	149	22.1	11.4	33.6	14.8	<b>51.7</b>	<b>66.4</b>	
	その他の	○	23	<b>47.8</b>	<b>21.7</b>	<b>69.6</b>	8.7	21.7	30.4	
	システム	△×	183	26.8	12.0	38.8	<b>15.8</b>	<b>45.4</b>	<b>61.2</b>	
<b>IT</b>	顧客管理	○	92	<b>57.6</b>	13.0	<b>70.7</b>	<b>16.3</b>	13.0	29.3	
	システム	△×	114	29.8	<b>14.9</b>	44.7	11.4	<b>43.9</b>	<b>55.3</b>	
	会計管理	○	63	<b>66.7</b>	12.7	<b>79.4</b>	9.5	11.1	20.6	
	システム	△×	143	31.5	<b>14.7</b>	46.2	<b>15.4</b>	<b>38.5</b>	<b>53.8</b>	
	人事管理	○	85	<b>60.0</b>	11.8	<b>71.8</b>	<b>15.3</b>	12.9	28.2	
	システム	△×	121	29.8	<b>15.7</b>	45.5	12.4	<b>42.1</b>	<b>54.5</b>	
	知財管理	○	54	<b>59.3</b>	<b>16.7</b>	<b>75.9</b>	<b>16.7</b>	7.4	24.1	
	システム	△×	152	36.2	13.2	49.3	12.5	<b>38.2</b>	<b>50.7</b>	
	SCM	○	74	<b>64.9</b>	12.2	<b>77.0</b>	12.2	10.8	23.0	
	システム	△×	132	29.5	<b>15.2</b>	44.7	<b>14.4</b>	<b>40.9</b>	<b>55.3</b>	
	在庫管理	○	76	<b>59.2</b>	<b>14.5</b>	<b>73.7</b>	<b>15.8</b>	10.5	26.3	
	システム	△×	130	32.3	13.8	46.2	12.3	<b>41.5</b>	<b>53.8</b>	
	設備管理	○	70	<b>62.9</b>	<b>15.7</b>	<b>78.6</b>	11.4	10.0	21.4	
	システム	△×	136	31.6	13.2	44.9	<b>14.7</b>	<b>40.4</b>	<b>55.1</b>	
	不動産管理	○	57	<b>61.4</b>	<b>15.8</b>	<b>77.2</b>	12.3	10.5	22.8	
	システム	△×	149	34.9	13.4	48.3	<b>14.1</b>	<b>37.6</b>	<b>51.7</b>	
		その他の	○	23	<b>60.9</b>	<b>17.4</b>	<b>78.3</b>	8.7	13.0	21.7



	システム	△×	183	39.9	13.7	53.6	<b>14.2</b>	<b>32.2</b>	<b>46.4</b>
<b>研究</b>	顧客管理	○	92	<b>39.1</b>	10.9	<b>50.0</b>	<b>13.0</b>	37.0	50.0
	システム	△×	114	17.5	<b>12.3</b>	29.8	10.5	<b>59.6</b>	<b>70.2</b>
	会計管理	○	63	<b>42.9</b>	<b>12.7</b>	<b>55.6</b>	6.3	38.1	44.4
	システム	△×	143	20.3	11.2	31.5	<b>14.0</b>	<b>54.5</b>	<b>68.5</b>
	人事管理	○	85	<b>41.2</b>	10.6	<b>51.8</b>	<b>11.8</b>	36.5	48.2
	システム	△×	121	17.4	<b>12.4</b>	29.8	11.6	<b>58.7</b>	<b>70.2</b>
	知財管理	○	54	<b>50.0</b>	<b>14.8</b>	<b>64.8</b>	9.3	25.9	35.2
	システム	△×	152	19.1	10.5	29.6	<b>12.5</b>	<b>57.9</b>	<b>70.4</b>
	SCM	○	74	<b>43.2</b>	10.8	<b>54.1</b>	9.5	36.5	45.9
	システム	△×	132	18.2	<b>12.1</b>	30.3	<b>12.9</b>	<b>56.8</b>	<b>69.7</b>
	在庫管理	○	76	<b>39.5</b>	<b>13.2</b>	<b>52.6</b>	10.5	36.8	47.4
	システム	△×	130	20.0	10.8	30.8	<b>12.3</b>	<b>56.9</b>	<b>69.2</b>
	設備管理	○	70	<b>44.3</b>	<b>12.9</b>	<b>57.1</b>	10.0	32.9	42.9
	システム	△×	136	18.4	11.0	29.4	<b>12.5</b>	<b>58.1</b>	<b>70.6</b>
	不動産管理	○	57	<b>49.1</b>	<b>12.3</b>	<b>61.4</b>	10.5	28.1	38.6
	システム	△×	149	18.8	11.4	30.2	<b>12.1</b>	<b>57.7</b>	<b>69.8</b>
その他の	○	23	<b>47.8</b>	<b>17.4</b>	<b>65.2</b>	<b>13.0</b>	21.7	34.8	
システム	△×	183	24.6	10.9	35.5	11.5	<b>53.0</b>	<b>64.5</b>	
<b>その他専門職</b>	顧客管理	○	92	<b>29.3</b>	9.8	<b>39.1</b>	<b>15.2</b>	45.7	60.9
	システム	△×	114	28.1	<b>10.5</b>	38.6	11.4	<b>50.0</b>	<b>61.4</b>
	会計管理	○	63	<b>36.5</b>	<b>11.1</b>	<b>47.6</b>	7.9	44.4	52.4
	システム	△×	143	25.2	9.8	35.0	<b>15.4</b>	<b>49.7</b>	<b>65.0</b>
	人事管理	○	85	<b>32.9</b>	8.2	<b>41.2</b>	<b>14.1</b>	44.7	58.8
	システム	△×	121	25.6	<b>11.6</b>	37.2	12.4	<b>50.4</b>	<b>62.8</b>
	知財管理	○	54	<b>35.2</b>	<b>14.8</b>	<b>50.0</b>	<b>14.8</b>	35.2	50.0
	システム	△×	152	26.3	8.6	34.9	12.5	<b>52.6</b>	<b>65.1</b>
	SCM	○	74	<b>35.1</b>	<b>10.8</b>	<b>45.9</b>	10.8	43.2	54.1

	システム	△×	132	25.0	9.8	34.8	<b>14.4</b>	<b>50.8</b>	<b>65.2</b>
	在庫管理	○	76	<b>30.3</b>	<b>10.5</b>	<b>40.8</b>	11.8	47.4	59.2
	システム	△×	130	27.7	10.0	37.7	<b>13.8</b>	<b>48.5</b>	<b>62.3</b>
	設備管理	○	70	<b>32.9</b>	<b>11.4</b>	<b>44.3</b>	11.4	44.3	55.7
	システム	△×	136	26.5	9.6	36.0	<b>14.0</b>	<b>50.0</b>	<b>64.0</b>
	不動産管理	○	57	<b>36.8</b>	<b>14.0</b>	<b>50.9</b>	<b>14.0</b>	35.1	49.1
	システム	△×	149	25.5	8.7	34.2	12.8	<b>53.0</b>	<b>65.8</b>
	その他の	○	23	<b>47.8</b>	<b>21.7</b>	<b>69.6</b>	13.0	17.4	30.4
	システム	△×	183	26.2	8.7	35.0	<b>13.1</b>	<b>51.9</b>	<b>65.0</b>
上記以外	顧客管理	○	92	<b>17.4</b>	4.3	<b>21.7</b>	<b>9.8</b>	68.5	78.3
	システム	△×	114	12.3	<b>6.1</b>	18.4	7.9	<b>73.7</b>	<b>81.6</b>
	会計管理	○	63	<b>22.2</b>	4.8	<b>27.0</b>	7.9	65.1	73.0
	システム	△×	143	11.2	<b>5.6</b>	16.8	<b>9.1</b>	<b>74.1</b>	<b>83.2</b>
	人事管理	○	85	<b>18.8</b>	4.7	<b>23.5</b>	8.2	68.2	76.5
	システム	△×	121	11.6	<b>5.8</b>	17.4	<b>9.1</b>	<b>73.6</b>	<b>82.6</b>
	知財管理	○	54	<b>24.1</b>	<b>7.4</b>	<b>31.5</b>	<b>9.3</b>	59.3	68.5
	システム	△×	152	11.2	4.6	15.8	8.6	<b>75.7</b>	<b>84.2</b>
	SCM	○	74	<b>20.3</b>	4.1	<b>24.3</b>	8.1	67.6	75.7
	システム	△×	132	11.4	<b>6.1</b>	17.4	<b>9.1</b>	<b>73.5</b>	<b>82.6</b>
	在庫管理	○	76	<b>19.7</b>	3.9	<b>23.7</b>	6.6	69.7	76.3
	システム	△×	130	11.5	<b>6.2</b>	17.7	<b>10.0</b>	<b>72.3</b>	<b>82.3</b>
	設備管理	○	70	<b>20.0</b>	<b>5.7</b>	<b>25.7</b>	<b>10.0</b>	64.3	74.3
	システム	△×	136	11.8	5.1	16.9	8.1	<b>75.0</b>	<b>83.1</b>
	不動産管理	○	57	<b>26.3</b>	<b>7.0</b>	<b>33.3</b>	<b>10.5</b>	56.1	66.7
	システム	△×	149	10.1	4.7	14.8	8.1	<b>77.2</b>	<b>85.2</b>
	その他の	○	23	<b>52.2</b>	<b>17.4</b>	<b>69.6</b>	4.3	26.1	30.4
	システム	△×	183	9.8	3.8	13.7	<b>9.3</b>	<b>77.0</b>	<b>86.3</b>

(凡例)

- ・ ※1 … 業務システムの開発・保守・運用の経験の有無。

- ○ … マネージャー／リーダー／メンバーとして関与したことがある。
- △ … 他部署／他社に委託している。
- × … 関与したことがない。
- ※2 … プラニングから積極的に関わって意見を出す。
- ※3 … ある程度任せながら自分の意見を入れる。
- ※4 … 担当者に任せて最後に承認する。
- ※5 … ビジネス戦略立案に関与しない。
- **(太字)** … 各業務を、列毎に、※1の○（上段）と×（下段）同士で比較して、高い方の数値。