

Beyond the Black Box: Knowledge Overlaps in Software Outsourcing

出典 IEEE SOFTWARE Sept./Oct. 2004 pp3-10
著者 Amrit Tiwana, *Emory University*
訳者 大阪府立大学 工学研究科 中野 雅之
作成日 2005年8月17日

目次

- はじめに(論文の背景と概要)
- ソフトウェアを開発するにあたって
- 知識共有の重要性
- 調査内容
- 調査プロジェクトの特徴と分析
- 調査結果
- おわりに
- 付録1: “**Outsource or Insource?**”のフレームワーク
- 付録2: 評価指標の一覧

はじめに(論文の背景)

□ブラックボックスでのソフトウェア開発について

従来は、クライアント(発注者)、ベンダー(ベンダー)の両者ともソフトウェアの開発において相手の知識に精通している必要がほとんどない。



よって、ブラックボックスはアウトソーシングでのソフトウェア開発の核となる部分を担っている。

しかし、現在は

ブラックボックスを用いるべきではない開発計画も存在している

ここで明らかにしたいものは・・・

- ・相手側の知識が必要になるのはどのようなプロジェクトか？
- ・新規のプロジェクトを発足する場合、どのようにブラックボックスではないアプローチをとる必要があるのか？

はじめに(論文の概要)

論文ではアメリカにあるソフトウェア開発会社(クライアント)の209社、209プロジェクトを調査

- ・調査結果もとに以下の2つを作成(内容は「付録」で紹介)

クライアントとベンダーの知識共有の形を判定するフレームワーク
(=“Outsource or Insource?”)

アウトソーシング業務を改善する詳細なガイドライン

- ・調査して分かったこと

効果的なアウトソーシングには4パターンの知識共有がある

ソフトウェアを開発するにあたって

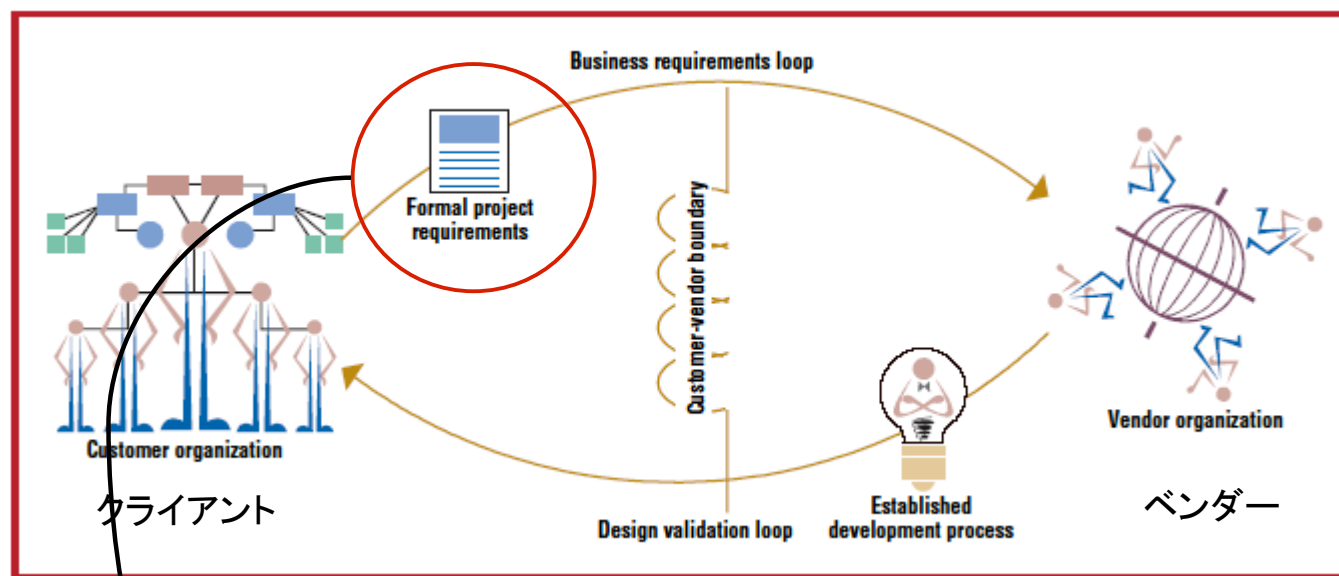
良いソフトウェアとは

問題ドメインの専門的知識と技術的な専門知識を正確に具現化するもの

※問題ドメイン:これから作るソフトウェアが解決しなければならない問題の発生している場所のこと

クライアントを満足させるにはこの2つとも必要

クライアントとベンダーの持っている知識の違い



クライアント

プロジェクトのニーズに関する知識を持っている

ベンダー

ソフトウェアを製造するための技術に関する専門的な知識を持っている

ブラックボックスの場合、お互いの知識を共有することはない。また、クライアントは設計書しかベンダーに渡さない。

知識共有の重要性

ブラックボックスでの開発について

ブラックボックスは良く知られた業務に対するソフトウェアを従来と同じ手順で開発委託するときには有効なのだが...

新規性を含んでいる開発プロジェクトにおいてブラックボックスを用いての開発は危険。

新規性には「概念的なもの」(クライアント側)、「開発プロセス的なもの」(ベンダー側)の2通りがある。

どちらの場合も知識を相手側に提供するのには困難であり、また新規性によってどちらが知識を提供するか異なる。

知識の提供を誤れば、計画通りにいくであろうプロジェクトも台無しになる恐れがある

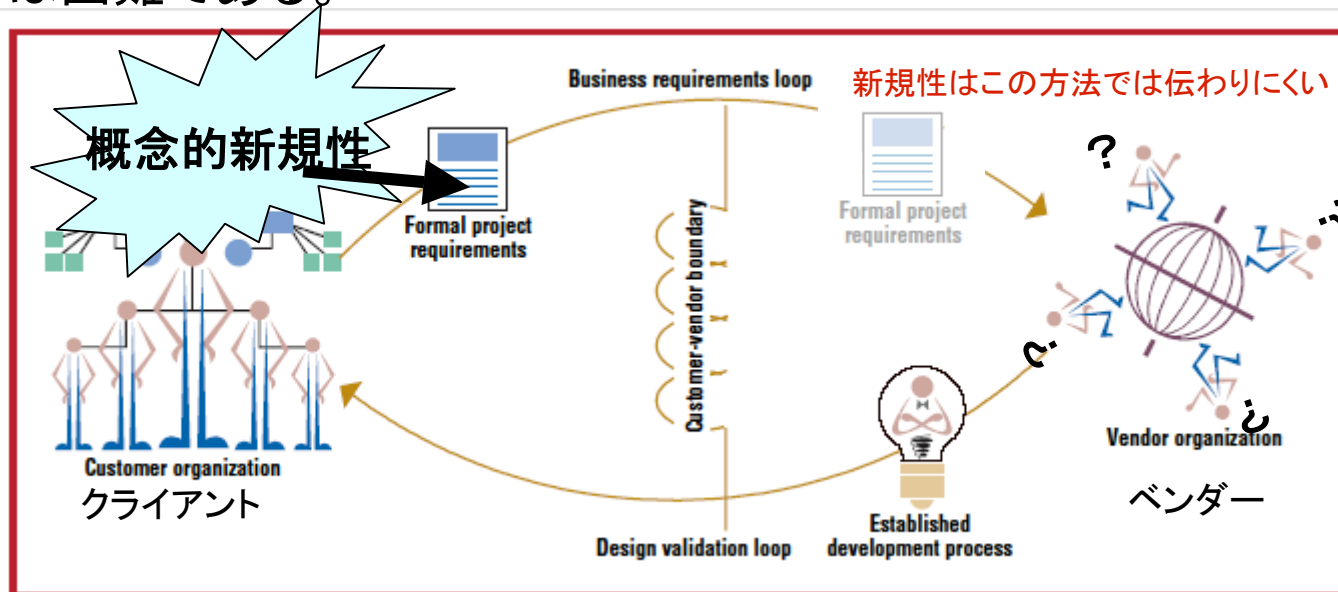
知識共有の重要性

2つの新規性

1) 概念的な新規性 (Concept novelty)

前例の無い問題を解決する場合や、開発する側にとって、プロジェクトの考え方がまるっきり新しくなる場合に発生する。

概念的な新規性が発生した場合、クライアントがその新規性を設計書だけでベンダーに伝えること(図のシステム)は困難である。



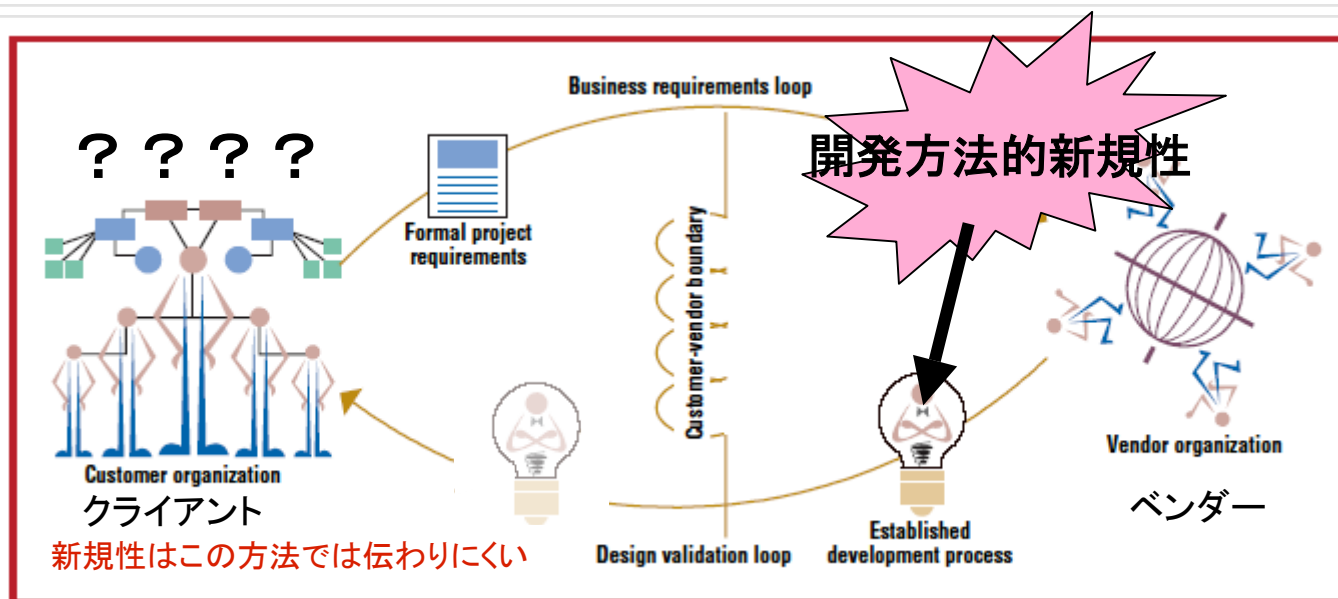
知識共有の重要性

2つの新規性

2) 開発プロセスの新規性 (Novel development processes)

クライアントの要望は明確であるがベンダーが新しいプロセスでソフトウェアの開発を試みる場合に発生する

プロセスが新しいために、クライアントはソフトの開発段階でベンダーをチェックしたり、フィードバックを行ったりすることができない



調査内容

調査の目的

いつ、そしてなぜ知識共有がアウトソーシングに必要なのかを決定すること

・調査は3段階

第1段階: 19社のプロジェクトマネージャーにインタビュー

第2段階: 818のソフトウェアプロジェクトに携わっている機関でアンケートをとる(232データ、返答率28.4%)

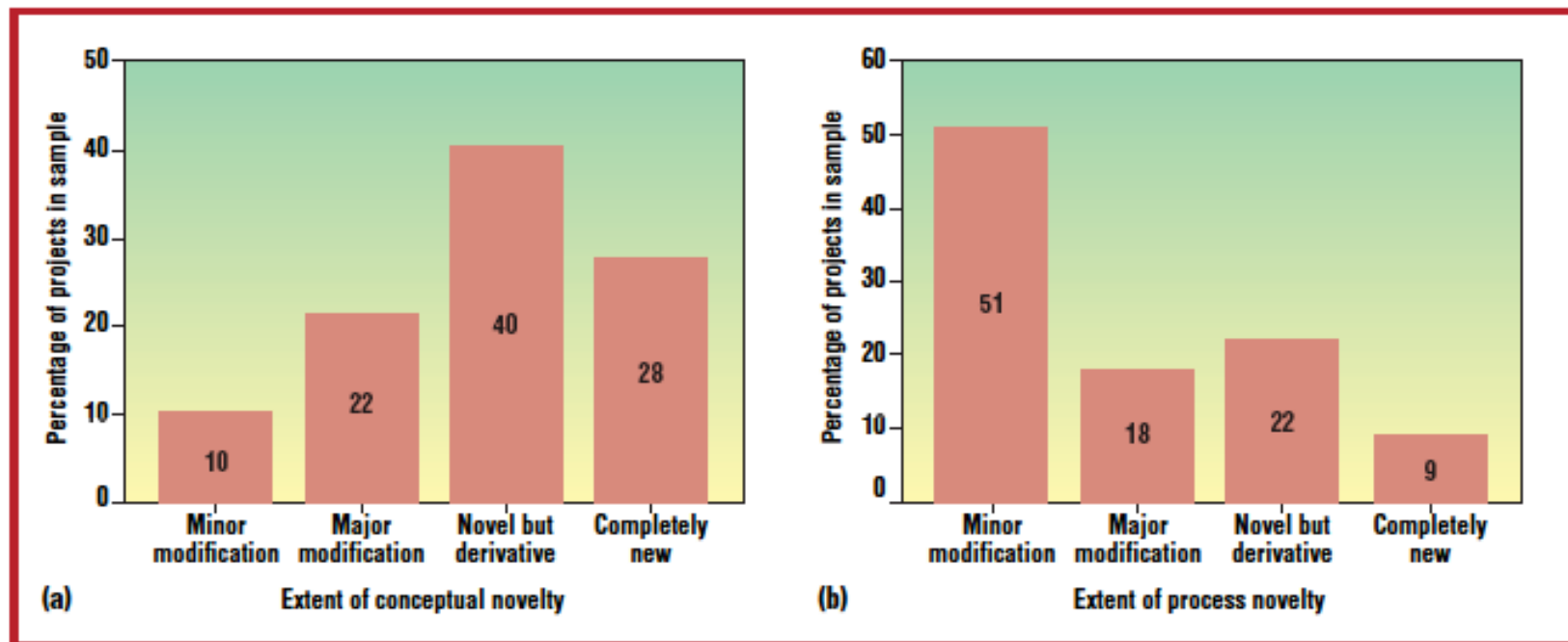
第3段階: 同プロジェクトに対するアメリカのクライアントの評価を収集(209データ)

※アンケートに用いた調査指標の一覧は付録に掲載

調査内容

調査したプロジェクトの特徴

プロジェクトの種類(左から小さな修正、大きな修正、新規だが派生的、まったくの新規なもの)の分布をあらわしている



概念的なものとはプロセス的なものでその分布が異なっている

調査プロジェクトの特徴と分析

調査したプロジェクトについて

- ・プロジェクトの期間(平均11ヶ月、最小1ヶ月、最大4年)
- ・構成人数の(平均16人、最小2人、最大150人)
- ・以前に組んだことのあるクライアントとベンダーの割合(59%)
- ・企業と付き合い期間(平均3.6年)
- ・どのくらいソフトウェア事業を行っているのか(平均7年、最小新規、最大47年)
- ・期間超過したプロジェクトは18.95%
- ・予算超過したプロジェクトは15.7%
- ・CMMの平均は2.5(0~5の六段階) ※CMM: Capability Maturity Model

分析方法

多重回帰モデルを使って分析を行った

統計的有意性は95%の信頼区間を用いている

→つまり結果は偶然のものではない。

調査結果～各プロジェクトの特徴～

調査から得られた結果

概念的な新規性がある場合 プロセスの新規性がある場合

クライアントが技術的な知識を持っている場合

ベンダーが業務ニーズの知識を持っている場合

Knowledge type	Routine	Project type Conceptually novel	Novel processes
Client organization's technical knowledge	-	No statistical evidence found	+
Vendor organization's business application domain knowledge	No statistical evidence found	+	+
Corresponding optimal client-vendor knowledge overlap pattern			

+は正の相関を、-は負の相関をあらわしている

一番下の行はプロジェクトタイプごとの知識のオーバーラップの形を示している

- ・ブラックボックスでの開発が有効なのはroutineのプロジェクト(左端)だけ

調査結果～知識共有のパターン～

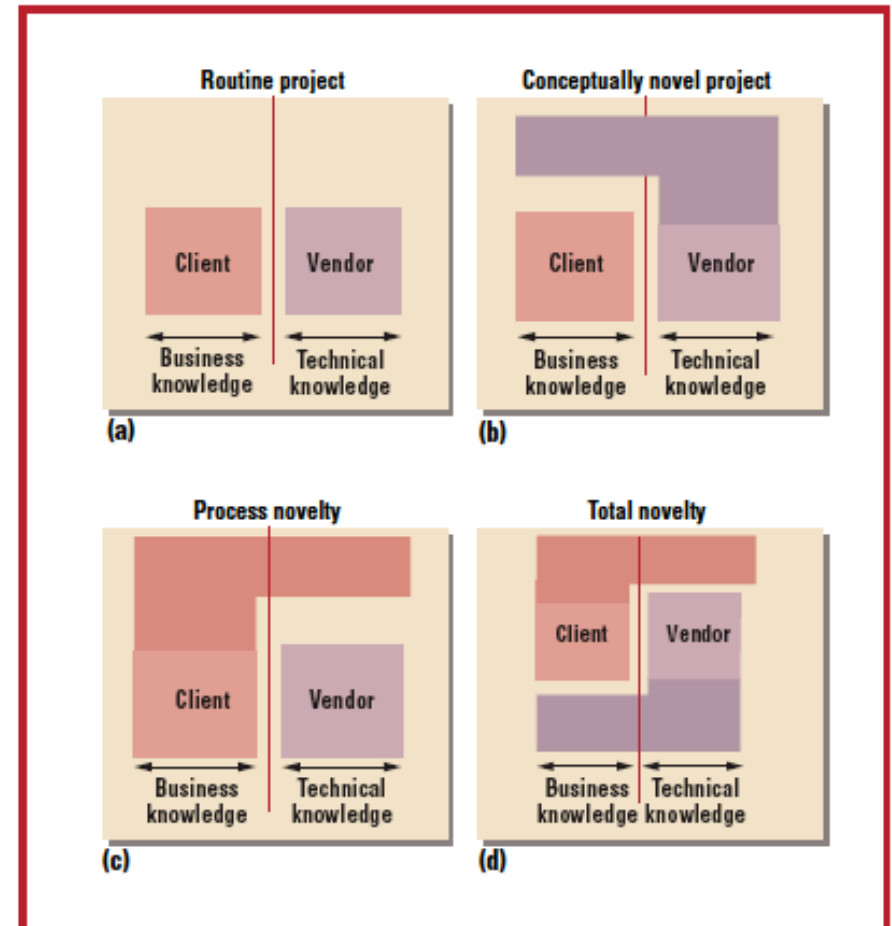
調査から得られたポイント

左の図は知識のオーバーラップの種類を示している

- ・business Knowledge とはクライアントの業務ニーズについてのもの
- ・Technical knowledge とはベンダーが持っている知識

プロジェクトの知識の形が図と一致するなら、設計過程の質を高めることができる

設計過程の質の向上は開発プロセスの改善につながる



調査結果～指針になるもの～

アウトソーシングのガイドライン

- 1) 日常的に使うものを開発している製造業者はクライアントの知識を得ることで利得を直接得られることは無い
また、この場合はクライアント側が高い技術知識を持っているとかえって計画の効率が悪くなる
- 2) プロセスの新規性を含んでいるプロジェクトの場合、ベンダーがクライアントの知識を持っているとかえって効率が悪くなる
ベンダーがクライアントの知識を持っていると時にクライアントの問題点が分からなくなってしまう
→むやみにクライアントの知識を得ることはクライアントに技術的な専門知識を教えるぐらいベンダーには危険な行為である
- 3) アウトソーシングを考えているマネージャーは図(一つ前のスライド)に合うようなアウトソーシングの関係を注意深く形成しなければならない

調査結果～指針にならないもの～

図の4つのうち、最も効率的な平衡メカニズム4dが開発プロセスを通じてのクライアントとベンダーの相互関係に近い

その他の開発調整ツールについての検証

検証するのは6つの開発調整ツールとベンダーのCMMレベル、準備段階の設計段階への大幅な投資

6つの開発調整ツール

要求管理者 (requirements managers)

構造設計者 (Architectural modelers)

試験の自動化 (test-automation tool)

テストケースの開発 (test case development tool)

構成管理者 (configuration managers)

失敗や変更の追跡機能 (defect and change request-tracking tools)

検証したはいいがこれらが開発プロセスの改善につながる証拠を何も見つけることができなかった

調査結果～今後の可能性～

その他の検証

ポストウォーターフォールモデルは注目に値するものである

これは調査した209のプロジェクトにおいて何が最も役に立つのかと言うものの核心に触れている

しかし、過去にこの様なモデルを扱っていない限りは無理に採用するべきではない。

→早急な採用はプロセスの新規性を生み出すから

おわりに

まとめ

調査からは銀の弾丸(特効薬)は見つからなかった。

しかし、今回の調査で、効果的なアウトソーシングの関係を設計するため、そしてどんなときにアウトソーシングを完全に避けるべきかを定めるためのシンプルでパワフルなアプローチを発見できた。

“outsourcing or insourcing?”(この部分の訳はまだ完成していません)でクライアントとベンダーの提携に関する枠組みを作ることができた。

今回の調査で、どういう構成がアウトソーシングにおいて、ぴったり合うのかを評価する強力な枠組みを提供することができた

付録1: “Outsource or Insource?”のフレームワーク

このフレームワークはクライアントと開発もとの知識の関係がどのようなものであるか、そして外注すべきかどうかを判断するもの

Step1: プロジェクトの新規性のタイプを調べる

概念的な新規性を知るために、ベンダーの実績、自身が持っている技術的な知識に関して Figure Aを用いて得点化する。

各項目に1～4点の点数をつける

合計点が11点以上なら概念的な新規性が高いと判断できる。

プロセスの新規性を知るために、開発の方法と以前ベンダーが使っていたツールに関して Figure Bを用いて得点化

各項目に1～4点をつける

合計が6点以上ならプロセスが新規であると判断できる

Assessment area	Score
Design concept	3
System design	4
System functionality	3
Business application problem domain	4
Conceptual novelty score (add rows)	14

1 = minor modification of a pre-existing system,
2 = major modification of a pre-existing system,
3 = completely new design based on approved concept
4 = unprecedented design

Figure A. Assessing a project's concept novelty.

Assessment area	Score
Development methodology	2
Software development tools	4
Process novelty score (add rows)	6

1 = existing processes with minor modification,
2 = existing processes with major modification,
3 = new processes derived from existing ones
4 = entirely new and untried processes

Figure B. Assessing a project's process novelty.

付録1: “Outsource or Insource?”のフレームワーク

Step2: 自社の技術的な知識のレベルを調べる

Figure Cを用いて自社の技術知識のレベルを調べる。

各項目に1～10の点をつける

合計が34点以上なら、クライアントの技術的な知識のレベルは高いと判断される

Assessment area	Score
Detailed systems design	10
Understanding of technical design constraints	6
Programming language to be used for this project	4
Understanding of various development methodologies	10
Testing and debugging procedures for software code	10
Familiarity with software development coordination tools (such as those used for requirements management, modeling, test case development, configuration management, and defect/change request tracking)	7
Client's technical knowledge score (add rows) ‡	47

Figure C. Samples scores in assessing client organization's internal expertise.

Step3: ベンダーの知識レベルを調べる

Figure Dを用いてベンダーが今回依頼するプロジェクトに関する自社(クライアント)の業務(ビジネス)についてどれくらい知識を持っているのかを調べる。

各項目に1～10の点をつける

合計が34点以上なら開発もとの知識レベルは高いと判断できる

Assessment areas	Score
Business processes	3
Application problem domain	4
Day-to-day business routines	9
Business rules to be implemented in the system	10
Your business objectives for this project	1
Familiarity with existing systems in your organization with which this system must interoperate	4
Vendor's business application domain knowledge score (total)	31

Figure D. Samples scores in assessing vendor's business application domain knowledge.

付録1：“Outsource or Insource?”のフレームワーク

Step4: 外注するかどうかを判断する

Step 1～3とクライアントと開発もとの知識構成(Figure 4 後述)を使って、プロジェクトを外注すべきかどうかを判断する。

そのためにFigure E を使う。

・・・と書いているがFigure E が Figure D と同じ表のため最終ステップがどのような方法で判断するのか分からない

Assessment areas	Score
Business processes	3
Application problem domain	4
Day-to-day business routines	9
Business rules to be implemented in the system	10
Your business objectives for this project	1
Familiarity with existing systems in your organization with which this system must interoperate	4
Vendor's business application domain knowledge score (total)	31

Figure E. Determining the feasibility of outsourcing a project according to project profile.

付録2: 調査指標一覧

クライアントの技術知識

プロセス、開発、プログラミングなどに関する知識
に関する指標

ベンダーのクライアントの業務に対する知識

クライアントが抱えている問題ドメインに関する知識
に関する指標

知識共有度

上の2つの知識をどれだけコヒーレントに一体化させているかを評価する指標

クライアントとベンダーの協力度

プロジェクトのライフサイクルを超えた協力の度合いを評価する指標

開発調整ツールの使用度

前述の6つの開発調整ツールをどの程度使用しているかを評価する指標

構造設計重点度

準備段階の開発設計に割く時間に関する指標

概念的新規性

新しいコンセプトか、技術的な前例がないかを評価する指標

開発プロセス的新規性

ソフトウェアの開発方法の新規性を評価する指標