

# 利用者主導のためのチェーン型開発に求められる発注側プロジェクト管理

林 浩一

## Project Management for Chain-Schemed System Integration

Koichi Hayashi

利用者主導でマルチベンダー開発を可能にする新たな開発モデルとしてチェーン型を提案し、その実現に必要なプロジェクト管理の役割について述べる。企業の基幹システムを構築するために従来行われてきた、多重請負によるピラミッド型の開発モデルには批判が強まっているが、外部発注をやめシステム開発の内製化に転換するのは、日本の企業にとって現実的ではない。チェーン型開発はユーザ企業側に開発に必要な枠組みを整えることでマルチベンダー統制を可能にする。

In this paper we propose *chain-schemed system integration*, which is a new type of multi-developer organization scheme designed for user oriented system integration. Recently, a widely adopted system development scheme is gaining bad reputation as the essential cause of project failures. It takes multi-layered contracts scheme that includes a lot of developers and forms a pyramid structure. Although some recommend to quit out-sourcing system development and return to in-sourcing, it is not a realistic solution for most Japanese companies. Chain-schemed system integration is another approach, which provide a standard development framework to enable users managing system integration projects under multi-developer environments.

Key Words & Phrases : マルチベンダー開発, 開発プロセス, 開発標準, 内製化, 多重請負  
multi-developer management, system development process, in-sourcing, multi-layered contract

### 1. はじめに

ICT の急速な進展によって、これまで基幹系企業システム開発の中心を担ってきたシステム開発(SI)のビジネスモデルが大きな転換点を迎えている。基幹系の企業システムを開発する際には、大手SI会社が開発案件をまとめて発注し、元請けとして複数の再委託業者の管理も含めて実施してもらう形態が広く採られてきた。

この形態でのシステム開発によって、汎用機の時代から現在のオープン系システムを中心としたシステムへの転換は進められてきたが、昨今はその見直しを求める議論が広がりつつある。問題提起の論点はさまざまだが、根本には開発受注側に大きく依存するこの開発モデルでは、発注側ビジネス要求に的確かつ迅速に応えられなくなっていることにある。

しかしながら、実際に開発をどう行うのかについて、現在の開発モデルに代わるビジネスモデルとして、目指すべき方向は明確になってはいない。一部には、開発を外注するのをやめて、自社内で実施するという「内製化」を進めているところも現れてはいるが、そのリスクは大きく、多くの発注側企業には現実的なアプローチではない。

著者の会社は、「利用者主導の開発」を実現するための支援を事業として展開してきた。この活動の中でから、丸投げでも内製でもなく、発注側の統制のもとで開発会社が開発を進める企業間協業のパターンを見出すに至った。本論文では、このシステム開発モデルを「チェーン型開発」と名付け、次世代の開発ビジネスのあり方を模索するためのリファレンスモデルとして提案する。

以下、2節で従来のシステム開発であるピラミッド型のモデル、3節で内製化と発注者主導のモデルの課題を述べた上で、4節にてチェーン型開発モデルの要件を明らかにする。5節にて、著者が展開しているAIST包括フレームワークと、それをういた札幌市の基幹システムの事例を紹介し、チェーン型開発のモデルの推進に必要なプロジェクト管理に関わる論点を提示する。

## 2. ピラミッド型開発モデルの限界

### 2.1 ピラミッド型モデルの利点

企業の基幹系システムを構築するにあたっては、システム開発の大手 SI 企業による一括請負の形態が広く採用されてきた。この開発モデルでは、システム開発全体に責任を持つ一次請け企業を頂点とし、二次請け、三次請け、(場合によってはそれ以上の)多重請負構造のピラミッドを形成して開発が行われることが多い。

役割分担は、大まかにはプロジェクトの全体管理・要件定義・基本設計などいわゆる上流工程を階層上位の企業が行い、詳細設計・実装・テストなど実際の開発実作業を階層下位の企業が担う。こうした多重請負による大規模プロジェクトの遂行形態は、IT ゼネコンと呼ばれることがあが、この呼称は、否定的な文脈で使われる傾向があるため、本論文ではピラミッド型と呼ぶ。

ピラミッド型の開発モデルの発注側にとっての利点は、商流と開発全体の責任を一次請け企業に集約できるところにあり、大きな規模の開発では広く採用されている。

開発会社にとっては、リスクの大きな仕事ではあるが、実施方法の自由度が高いため、金額規模が大きくなれば、内部原価の削減努力によって収益を確保することが可能である。固有のハードウェアやミドルウェアを導入することで、継続開発を競争なしに受注する顧客囲い込み戦略も収益確保には有効であるが、発注企業からはベンダーロックインと批判的に捉えられることもある。

### 2.2 ピラミッド型開発の課題

現在のようにシステムを用いたビジネス遂行が一般的になり多様性も広がると、要件が複雑になるとともに、要求への対応の迅速性が求められる。そのため、開発体制の規模が大きくなるピラミッド型のモデルでは、システムの利用者と開発の実作業者が遠く離れ、一次請け企業が状況共有のボトルネックになりやすい。

利用者からは作業の実状が把握できず、開発の実作業者からは利用者の要望を十分に理解できない、という状態になるため、プロジェクトの遅延や迷走、あるいは要件漏れや性能未達といった問題が起きやすい。

加えて、現在では、ハードウェアやインフラのクラウド化と、ミドルウェアのオープンソース化が進んだため、固有技術を前提とした顧客囲い込みも難しくなった。このため SI 企業にとって、大規模なフルスクラッチの開発は失敗リスクが高く、利益も上がらないビジネスになっている。

## 3. 内製化と新しい開発モデルの模索

### 3.1 内製化への回帰

SI 専門会社の存在とピラミッド型での開発は、国際的に見ると珍しく、日本の特異性として理解されるようになった[7]。このことから、日本でもシステム開発を外注せずに自社で内製化すべきであるという議論がある。

日本でも汎用機全盛の時代に内部の情報システム部門で開発をしていたが、部門をシステム子会社として切り離し、自社内には発注機能のみを残して外注するという流れで進んできた企業も多い。そのため、以前の状態に回帰するべくシステムを内製化の方向へ進もうとしている企業もある。

しかし、外注に大きく依存してきた発注側の情報システム部とシステム開発を実施してきた SI 企業では、経験と技術の両面で大きな開きができています。基幹システムの開発には、実際のシステム開発要員に加え、経験のあるプロジェクトマネージャとプロジェクト管理に必要な標準や体制の整備が必要であり、短期間には揃えられない。

内製化で特に留意が必要なことは、プロジェクトの失敗責任を受注側に転嫁できなくなることであり、現在のシステム開発は、複雑で多様化しており、SI 専門企業でも失敗することが珍しくない。そんな中で内製化を行うことは、多くの企業にとって現実解とはなりえない。

### 3.2 新しいシステム開発モデルの要件

ピラミッド型モデルに代わる開発モデルの模索は、実践と理論の両面から進められている[4]。アジャイル開発のように、利用者と開発者が一体的に開発に取り組むような考え方や、ずっと開発状態を続けて納品をしないモデル、プログラムの自動生成によって開発部分を大きく削減する方法などが実践に基づいて提案されている。これらは、オルタナティブ SI と総称されることもある[8]。

これらの提案に共通する点は、ピラミッド型のモデルで、一次請け企業への依存度が強くなりすぎて、「丸投げ」と呼ばれている状態を見直し、発注側自らが主体的に開発を管理することを重要視するところだ。しかし、いずれのアプローチも小規模で限定された適用にとどまっている。大企業の業務を支えるシステムの場合、数億円～数十億円の規模の開発になることは少なくないが、こうした大規模開発への適用にまでは至っていない。

今後、本格的にピラミッド型の開発モデルを置き換えていくためには、発注者主導の特性を満たしつつ、大規模システムへの適用ができる開発モデルが必要である。

内製		ピラミッド型		チェーン型		
	発注側	受注側	発注側	受注側	発注側	受注側
統制	○	×	×	○	○	×
開発	○	×	×	○	×	○

図1 内製, ピラミッド型, チェーン型

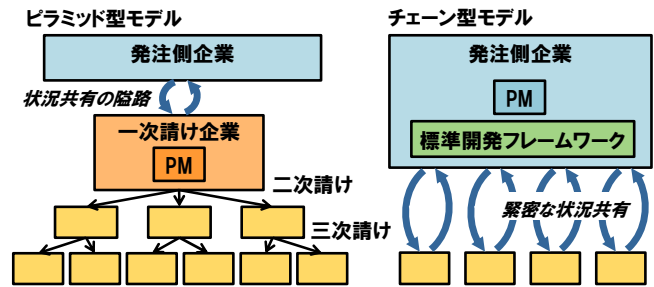


図2 ピラミッド型とチェーン型の体制の違い

#### 4. チェーン型開発モデル

##### 4.1 チェーン型モデルの要件

前節で挙げた要件を満たす新しい開発モデルとして、本論文ではチェーン型開発モデルを提案する。このモデルは、システムの開発を、開発統制の役割と開発実施の役割に分けたときに、開発統制を発注側、開発実施を受注側で行うモデルである(図1)。ピラミッド型の開発モデルは、開発統制と開発実施の両方を受注側企業が行う。内製化は反対に、両方の役割を発注側企業が行う。

体制の違いは典型的には、大規模なシステム開発を機能ごとに分割し、複数の開発会社によって進める形態である。このモデルを特徴づける重要な要素は、発注側にシステム開発の進め方や、構築するシステム全体のアーキテクチャなど開発の枠組み定めた「標準開発フレームワーク(FW)」を用意するところである。

こうした開発の枠組みはピラミッド型の開発であれば、一次請け企業が定めていたものであるが、ユーザ企業が用意し、開発企業に遵守してもらうことで、複数企業が並列して開発を行っても、プロジェクト全体の進捗を的確に把握でき、適切に全体連携するシステムの構築が可能になる。

このモデルを「チェーン型」と名付けたのは、ユーザ側が定める標準を軸にして、複数の開発企業が連なることができることからである。チェーン型開発モデルは、標準開発FWを媒介にして、ユーザ企業側と開発の実作者を近づけ、より直接的に状況の共有を可能にする。

チェーン型開発の要件は次の3点になる。

##### A) フラットストラクチャ

発注側の開発標準に従う複数の受注企業が、それぞれ独立に開発実作業を行うこと。

##### B) ユーザコントロール

発注側のプロジェクトマネージャー(PM)が自社の開発標準に従って受注企業を統制すること。

##### C) スケーラビリティ

小規模プロジェクトから大規模プロジェクトまで同じ考え方を適用できること。

#### 4.2 プロジェクト管理の特徴

発注側のプロジェクト管理の定常的作業は、発注側で用意する標準開発FWにしたがって、受注企業の開発の進捗を管理し、成果物のレビューを実施していくことになる。開発標準FWは、次の3種類の統制について定める。

##### (1) 開発手順の統制

開発を進める手順について定める。要件定義、設計、実装・テスト、といった工程の分け方や工程を進める判断基準などを定める。異なる開発会社は異なる開発手順に従うため、この統制がないと全体の状況把握ができなくなる。

##### (2) 開発成果物の統制

各開発工程で残す成果物を定める。要件定義書や設計書といった成果物の種類とそれぞれの記述内容を定める。開発手順と同様、開発成果物も会社によって異なるため、この統制がないと内容も品質基準もバラバラになるため、内容の整合性の判断ができず、レビューもできなくなる。

##### (3) 開発技術の統制

構築するシステムで共通に利用する技術を定める。構築するシステムについて、使用言語をはじめ、利用するフレームワークやライブラリ、システム連携の方法などを定める。この統制がないと異なる会社の作成したシステムが全体として、適切に連携して動作できなくなる。

#### 5. チェーン型開発の実践

##### 5.1 AIST 包括FW とガラスボックス化

筆者らは、システム開発をAIST包括フレームワーク(FW)に基づいた利用者主導の開発の展開を行ってきた。このフレームワークは、国立研究開発法人産業技術総合研究所(略称 AIST)で開発された利用者主導の開発の枠組みで、発注側に備えるべき開発標準を体系化したものである[1]。

前節で述べた、標準開発FWは、AIST包括FWを一般化したものだが、この枠組みを適用すれば必ずチェーン型の開発になるわけではない。

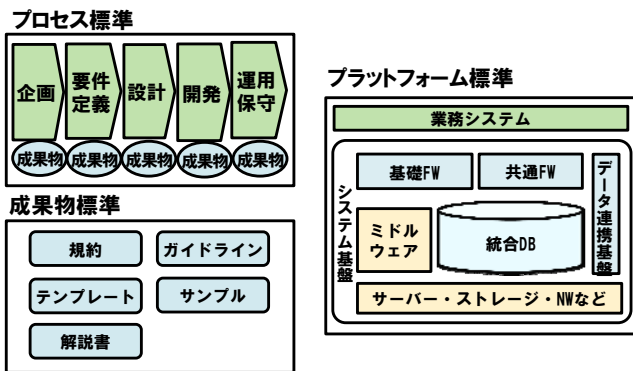


図3 AIST 包括フレームワーク (FW)

AIST 包括 FW は、チェーン型モデルの標準開発 FW で示した構成要件を満たす要素を持つが (図 3)、これらを、トレーサビリティ、利用者による可読性、標準技術の活用、の3つを重視して定めている [3]。これらはシステム開発の透明性を高めるものであり、この特性を持たせることをガラスボックス化と呼ぶ。ガラスボックス化は、AIST 包括 FW の重要な特徴の一つであるが、チェーン型開発に必須の要件になるかどうかは、現時点では明らかではない。

#### (1) プロセス標準

開発手順を統制するために、企画、要件定義、設計、開発、運用保守の工程について進め方を記述している。プロセス標準で特に重視しているのがトレーサビリティである。トレーサビリティは、発注側が意図した要件が確実に機能として実現されることを保証するものである。トレーサビリティを確保することで、発注者側の要員は自分が理解できる範囲でチェックした範囲を、プログラムコードの中身まで理解できなくても、ブラックボックスとして管理することが可能になる。

#### (2) 成果物標準

開発成果物の統制を行うために、各工程で作成される成果物について、規約、テンプレート、サンプル、ガイドラインなどを用意することで、どのようなレベルの記述が求められているのかを明確にしている。成果物標準で特に重視しているのが利用者による可読性である。受注側が主導で開発を進める場合、利用者による理解より、構築の都合が優先される。具体的には、業務フローやユースケース記述といった、特殊な知識をもたなくても理解できる成果物を選定している。

#### (3) プラットフォーム標準

開発技術の統制を行うために、共通に使う技術として、採用している開発フレームワークやミドルウェアについて具体的に定めている。プラットフォーム標準で特に重視しているのが標準技術の活用である。広く使われているオープンソース

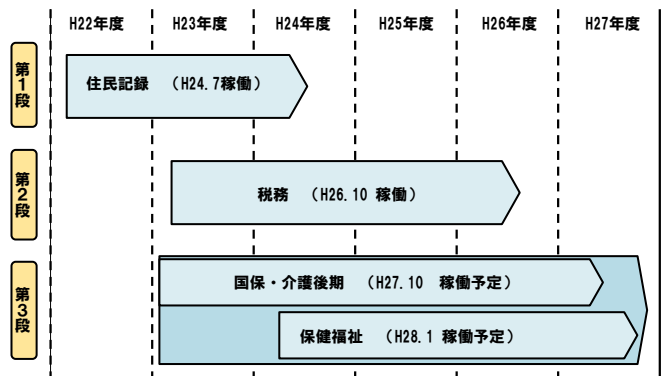


図4 札幌市基幹システム再構築事業 (文献[9]図2を引用)

フトウェア(OSS)などで構成することで、開発会社への依存性をなくすことができる。開発者に対してすべての技術内容が開示されるという観点での透明性を高めることができ、この状態をホワイトボックスの状態であると表現する。

## 5.2 札幌市の基幹システム開発

AIST 包括 FW を適用した開発は、産総研にはじまり、横浜市、札幌市といった形で公共系の案件から採用されてきた [5][6]。最大の適用案件である札幌市の事例にはチェーン型開発の特性が強く現れている。札幌市の開発事例は、文献 [9][10] の記述にあるように、以下の特徴を持っている。

- 開発対象は、汎用機を中心に構築されてきた、人口 194 万人の住民記録・税務・国民健康保険・保険福祉などの住民サービスを行う職員が使う基幹業務システム。
- 開発費は当初の計画では約 130 億円、その後、番号制度や制度変更対応により 150 億円。
- AIST 包括 FW に基づき札幌市用に拡張したもの(札幌市版)を活用した職員主導での開発。
- 関わった開発業者数は 84 社。同時に最大で 15 のプロジェクトが進行。
- 3 段階に分かれ、第 1 弾は住民記録系システムで平成 24 年に、第 2 弾は税務系システムで平成 26 年 10 月に稼働。第 3 弾は国保・介護後期、および、保険福祉システムで、本論文執筆時点で開発中。

チェーン型開発の要件について整理すると以下のようなになる。

#### A) フラットストラクチャ

札幌市では、各業務システムの開発プロジェクトが、要件定義、設計、開発、に分けて調達され、それぞれ独立した体制で実施された。札幌市の場合には、全体規模が大きいいため、各体制がさらに多重請負のピラミッド型を形成している。

## B) ユーザコントロール

札幌市では、AIST 包括 FW に対して、さらに独自要素を加えた上で、各受注企業を統制している。職員も利用者情報システム部門とで役割分担をして、チェックをする体制が敷かれている[3]。

## C) スケーラビリティ

札幌市の案件は、全体の規模は 100 億円を超えるプロジェクトであるが、業務ごと、工程ごとに分割され、それぞれが独立した案件として発注されている。小さなもので、数千万円、大きなものは、十億円を超えるプロジェクトの集合体となっている。これらのすべてが、同じ AIST 包括 FW 札幌市版の規定に従って進められている。

### 5.3 プロジェクト管理上の課題

札幌市の案件への AIST 包括 FW の適用を通じて、多くのことを得ることができたが、本節では、その中でチェーン型開発のプロジェクト管理について特徴的な論点を示す。

チェーン型開発における発注側のプロジェクト管理の活動は、進捗や成果物確認など定常的な活動と、随時発生するリスクと課題の解決活動に分けられるが、それぞれ特徴的な部分がある。

#### ① 成果物レビュー経験の蓄積

チェーン型開発では、発注者が主体的に関わるために成果物についてのレビューが重要になる。成果物の記述をどこまでの詳細度で記述するのかについては、仔細に規定することは難しく、最後まで構築行った経験からの獲得が不可欠である。

経験が蓄積されると、スケジュールの逼迫度や参画要員のスキルレベルに合わせて、レビューの進め方を変えるようなことも可能になる。例えば、スキルの高い要員に先に実施してもらって、それを他の要員に伝達するなどといった工夫により、品質を維持したまま工数を減らすことが可能になる[2]。チェーン型開発は、同じ開発手順を繰り返すので、こうした経験の蓄積には有効に機能する。

#### ② コードレビューができる技術要員

プラットフォーム標準には、共通の API を提供するだけでなく、プログラムのコードに関する規約も含まれる。参加している開発企業において、実際にその規約が守られているのかどうかを判断するには、コードレビューが必要になる。つまり、必要に応じて開発側の要員が書いたコードの中まで踏み込んで指摘できる技術要員が発注側にいなければ、開発を統制することはできない。

ただし、必ずしも発注企業の社員である必要はなく、例えば開発を受注しているのは別の開発会社に、第三者的な監査の役割としてコードレビューを依頼するのは現実的な選択肢である。

## ③ 発注側と受注側の双方との関係

チェーン型開発の発注側のプロジェクトマネージャは、通常のシステム開発以上に、利用者側と受注側の両者からの信頼関係が重要になる。

この開発モデルでは、受注企業での開発の進捗も成果物の品質も透明性高く、発注側から丸見えになってしまう。発注側がその状況を有利に使うて無理な要求がされないように、守ってもらえるといった安心感がなければ、受注側にとっては苦しいだけになる。

逆に受注者側に立った事情説明をすることで、利用部門から開発者の代弁者のように見られてしまうことも起こる。受注側、発注側の両方の事情を理解して、バランスを保ったプロジェクト管理をすることが求められる。

## ④ 最初の輪の重要性

チェーン型の開発では最初の輪、つまり標準開発 FW に従って行われる最初の開発を成功させることが極めて重要になる。経営層をはじめプロジェクト外の周囲の目からは、チェーン型開発によるシステム実現の試金石となる。初回の成功によって利用者側の開発への期待も高くなり、追い風となる。また、次以降の開発を円滑に進めるための知見も多く得ることができる。

一方で、初回ゆえの難しさもある。新しい開発手順にしたがって本当にシステムができあがるかどうかを実証されていない状態なので、発注側、受注側の双方にとって、不安な中でプロジェクトを遂行することになる。この払拭のためにも最初の開発は小規模に進めることが望ましい。

特に技術面での難しさとして、注意すべき点は、基盤システムの動作が適切かどうかは、その上に載せる個々の業務システムを動作させるまで検証ができないところである。業務システムに不具合が生じた時は、どちらに原因があるのかの切り分ける必要があり、お互いに責任を押し付け合うなど、開発者同士の摩擦が起きやすい。

関係する技術者同士の信頼関係が不十分だと、解決に余計な手間がかかり開発効率が低下することになる。このため、特に最初の開発において、基盤システムの開発者が、業務システム開発者への技術支援を十分に行うことで、こうした状況の回避に務めることが大切である。

### 5.4 ビジネスモデルの可能性

チェーン型モデルは開発の進め方にとどまらず、複数の企業間の活動によるビジネスとしてのモデルにも関わる。そうしたビジネス上の課題についての可能性について論じる。

### ① 新規調達のコストとのバランス

札幌市の事例は公共であるため、参加する企業については競争入札が必須となる。しかし、チェーン型開発も、AIST 包括 FW も、適用対象を公共に限るものではない。

新規参入の開発会社の採用にはリスクがある。開発会社にとっては、新しい開発プロセスを学習しなければならないので、初回参画時の学習コストは高い。競争入札が必須でなければ、少数の特定の会社と長期的な関係を築くことが有用である。受注会社固有の開発手順や技術に依存すると、発注側には選択の幅がなくなり、ベンダーロックインの状態になる懸念があるが、発注側の標準に従ってもらうことから、逆に他社との入れ替えが起こり得るという緊張感を持って開発を進めることになる。

### ② 複数企業横断の標準開発 FW

将来的には、開発標準 FW を複数の利用企業で共有して利用するという可能性が考えられる。一社で、開発標準 FW を維持していく負担は大きい。

とりわけ、プラットフォーム標準は新しい技術が現れるとそれに合わせた見直しが常に迫られる。例えば、データベースについて NOSQL のような技術、また、テストでは継続的インテグレーションのツールなどが普及することで、開発プロセスや成果物に大きな影響がある。

標準開発フレームワークは、様々な業務アプリケーションを動かし、他の様々なシステムの連携を媒介するという性格上、異なる会社でも利用することが可能である。複数の利用企業と複数の開発企業が、共通の標準開発 FW を使うことで、コミュニティを形成し得る。そのコミュニティで情報交換をし、お互いのプロジェクト管理スキルを高めながら、開発を進める可能性は、今後の ICT の発展を担える有望な将来像ではないだろうか。

## 6. おわりに

ピラミッド型開発のビジネスモデルは、汎用機の時代から Web システムへの転換期において、多くの企業の基幹システムを整備していく上で重要な役割を果たしてきた。長く続いたこのビジネスモデルも、その歴史的役割を終えようとしているように見える。このモデルには功罪の両面あるが、大きな弊害を残したことは否定できない。

最大の不幸は、プログラミングという高度専門スキルを多重請負のピラミッドの最低階層に位置づけて、その価値を下げたことにより、かつては存在した企業システムの開発という職業の誇りと輝きを毀損してしまったことだ。

それによって、優秀な人材の参入意欲を削ぎ、教育の機会を減らし、結果としてスキル不足の技術者を増やし、技術空洞化の原因となった。ピラミッド型の多重請負開発モデルの終焉をチャンスと捉え、延命を図るのではなく新しいビジネスモデルを模索すべき時機が来ている。

本論文で提案したチェーン型開発モデルもその模索のひとつである。創造的なビジネスの企画者と高度なシステム技術者が相互にリスペクトし合って進めるシステム開発の実現を目指したい。

## 参考文献

- [1] 林浩一, 和泉憲明: "ガラスボックス化により地方自治体での利用者主導の開発を実現する AIST 包括フレームワーク", 情報システム学会第 5 回全国大会・研究発表大会, 2010.
- [2] 山根基, 林浩一, 和泉憲明: "成果物品質の課題分析に基づくソフトウェアプロセス改善の考察", プロジェクトマネジメント学会 2012 年度春季研究発表大会, 2012.
- [3] 林浩一, 和泉憲明, 山根基, 丸山大輔, 岡田隆志, "Java/Web でできる大規模オープンシステム開発入門", 丸善出版, 2012.
- [4] 大槻繁: "ソフトウェアデザインプロセスの呪縛からの開放 - 人働説から知働説への転回 - ", Design シンポジウム 2012, 2012.
- [5] 日経コンピュータ: "IT 戦略強化の決め手 基幹系の刷新 札幌市 - ユーザー主導の分割発注ガラスボックス化で 20 年品質" 日経コンピュータ 2012/12/20 号, 2012.
- [6] 日経 SYSTEMS: "事例 ケーススタディー 問題解決の現場 - 横浜市 初の共通システム基盤作り 利用部門の要望で DB 見直す", 日経 SYSTEMS 2012/04 号, 2012.
- [7] 谷島 宣之: "ソフトを他人に作らせる日本, 自分で作る米国", 日経 BP 社, 2013.
- [8] 日経コンピュータ, "オルタナティブ SI - 未来を開く五つの鍵", 日経コンピュータ 2015/02/05 号, 2015.
- [9] 田中寛純, 千葉匡, 岩間雅巳: "情報システム部門のあり方を再構築する, 札幌市基幹系システム再開発の取組み(1) - 発注者主導を実現する「ガラスボックス」の仕組み", 自治体ソリューション 2015 年 10 月号, 2015.
- [10] 田中寛純, 千葉匡, 岩間雅巳: "情報システム部門のあり方を再構築する, 札幌市基幹系システム再開発の取組み(2) - マルチベンダー体制での競争入札を支える「システム基盤」", 自治体ソリューション 2015 年 10 月号, 2015.