

## 「要求獲得」の捉え方

——システム開発における問題認識とソリューションに向けて——

谷 地 弘 安

はじめに

本稿が焦点を合わせるのは、情報システムの開発工程において、超上流と呼ばれるフェーズであり、要求獲得工程と呼ばれるところである。すでに我々は、このフェーズにおける問題の所在について、端的な整理を試みた（谷地[2016]）。

一方、システム開発の分野には、要求獲得から開発、最終的な統合テストにいたるまで、業務遂行やプロジェクト・マネジメントのためのノウハウ、手引きがあふれるほど存在する。そして、形式化されたこれらの知識が成果物として流通し、我々は容易にアクセスできるようになっている。このような状態になっている分野は、恐らくほかにはないのではないだろうか。

これら成果物の流通目的は、もちろんシステム開発における問題・課題を整理し、それに対する解決策をあたえるためである。言い換えれば、それだけシステム開発にはさまざまな問題が存在し、開発を管理・遂行することには困難がともなうということである。だからこそ、解決に導くためのノウハウ、手引きが輩出されてきた。このような構図がある。

しかし、時間の流れで見えてみると、どうなのだろうか。問題の存在、それに対する解決策の提示がなされてきたとして、それによってシステム開発の問題は実際に解決されてきたのであろうか。あるいはシステム開発はより容易になってきたのであろうか。

そのように断言はできないように思われる。

なにより、解決に導くためのノウハウ、手引きはいまでも多数輩出されている。そして、そこでは必ずと言っていいほど、同じことが強調されている。すなわち、システム開発にはさまざまな問題が存在し、それを管理・遂行することには困難がともなう、という内容の文言である。

うがった見方をすれば、結局のところ、解決に導くためのノウハウ、手引きは問題の解決には十分結びついていないのかもしれない。そこで再び別のノウハウ、手引きが作成され、提供される。とはいえ、その間に内容上の大きな差はない。つまるところ、問題の存在、それに対する解決策の提示が、同じような内容のもとでループし続けているだけではないだろうか。

すると、それはなぜなのだろうか。ここには2つの見方がある。

1つは、現存のノウハウ、手引きには実務問題の解決力がない、ということである。

2つめに、解決力が存在しないのではなく、それを発現させるためには実はもっと異なる側面の問題をつかまえ、そこに対する解決策を考えねばならない、ということかもしれない。

本稿での我々の研究は、2つめの仮定からスタートする。ここで関わってくるのが、要求獲得を対象とする一連の研究とマーケティング研究との間にミゾがあることである。すなわち、前者でマーケティング研究に依拠したものはほとんど見る事ができない一方、マーケティング研究でシステム開発やそこでの要求獲得を題材にしたものを見ることもできない。しかし、要求獲得は明らかに売り手が買い手のニーズを探索するリサーチのフェーズである。したがって要求獲得を検討するのに、マーケティング研究に依拠することには、大きな可能性が秘められていると思われるのである。

とはいえ、マーケティング研究の援用には注意も必要である。本稿では、要求獲得に対するマーケティング研究の援用可能性を肯定しつつ、それを実あるものにするために必要な準備をすることにしたい。具体的には、システム開発において必要な要求獲得に関して、その営為をどのように認識するかについて、論述する。それによって、そこにはBtoB（産業財）の世界のなかでもユニークな特徴があることを明らかにしたい。それはマーケティングのみならず、製品開発論にまで及んでいる。また、実はシステム開発だけに閉じたものではなく、ほかのBtoB領域でも想定される一方、従来のマーケティング論が及んでいなかった課題がある。これらのことを指摘したい。

### 「情報システム」に関する認識

まず、そもそも要求獲得の対象となる「システム」について、言及しておく必要がある。なお、以降は「情報システム」を「システム」と略称する。

**業務タイプとシステム** ここでのシステムは、企業をもっぱらとする、組織での業務処理を対象としたものである。これら業務は企業としての機能と見ることができ、それぞれの機能について業務がある。機能としては経理、人事、販売・営業といったものがあるが、たとえば生産（製造業）や勘定（金融）といったように、業種によって企業が備えている機能に違いがある。そして、これら業務ごとに情報システムがある。

情報システムの分野では、企業における根幹となる業務を基幹系業務と呼び、それを支えるシステムを「基幹系システム」と呼ぶ。なにをもって基幹となすか。これはシステム・サイドから考えてみるとわかりやすい。すなわち、それはシステムが停止すると業務が滞ってしまうことである。基幹系システムが停止するということは基幹系業務が遂行できなくなるということである。たとえば、顧客からの注文を受けつけることができない、料金の計算ができない、商品の配送手配ができない、顧客に料金を請求できない、銀行だとATMが稼働しないといったことである。これが当該企業のビジネスに甚大な悪影響をおよぼすことは想像に難くない。

一方で、経営者をはじめ、管理者が非定型な業務を行うのを支援する情報システムがあり、これを基幹系に対して、「情報系システム」と呼ぶ。たとえば、新商品企画、出店計画、融資決定、予算配分、設備投資計画などのためのシステムである。

情報系システムも、これが停止すれば業務は滞る。しかし、基幹系システムが、いわば「止まってはいけないシステム」とされるのは、あらかじめ明確で厳格、そして一律なルールが存在し、データを入力することで、それを確実に処理していくことが求められるところにある。

一方、情報系システムは意思決定の高度化や創造性に資するアウトプットを提供するところ

に目的がある。たとえば、CRM（顧客関係管理）のシステムを考えると、これは頻繁に機能のバージョンアップが行われる。そのたびごとに製品発表会が実施され、有償のバージョンアップ情報が担当者に送られてくる。基幹系システムにおいてこのような頻繁なバージョンアップがなされたらどうだろうか。それによっていったいなにが変わるのだろうか。これまでのシステムよりも創造性の高い決済システムなど、そもそも概念的にありうるのだろうか。しかも、そんなことをシステムのユーザーが求めるのだろうか。

基幹系システムと情報系システムは、それぞれが対象としている業務の性格、実現すべき目的に違いがあることから分けられている。それゆえ、各システムに対するクライアントの要求にも違いが出てくるうえ、実現すべき要件にも違いが出てくることになる。とはいえ、基幹系にせよ情報系にせよ、それを導入するためには要求獲得というフェーズが必要となることに変わりはない。その意味で、本稿では特に業務のタイプ、およびそれにとまなうシステムのタイプについては、議論上峻別することはしない。

システムの利用スタイルと開発スタイル このシステムをどのようなかたちで利用するか、これについて言及が必要である。別稿で述べているように、システムには現在、その利用形態として、大きく「オンプレミス」と「クラウド」がある（谷地[2015]）。本稿で対象とするのは、もっぱらオンプレミス型のシステムであり、伝統的システムとも言い換えることができよう。もちろん、オンプレとクラウドをある案件で明確に二分することができないケースもある。その意味では、オンプレ「寄り」のシステムということで、オンプレミス型と呼ぶことにしたい。

さて、オンプレ型のシステムを対象とした場合、あらゆる商品と同様、そこには売り手と買い手がいる。売り手の方はシステムを開発し、それを買い手に納品する。以降、買い手がそのシステムを利用していくことになる。ここでは、システムの買い手を「クライアント」、売り手を「SIer」と呼ぶことにする（谷地[2015/2016]）。

クライアントが業務をシステム化するとき、SIerがその開発にあたって既存の製品や型を利用するかどうか、これによって分類する切り口がある。まず、「スクラッチ開発」とは、既存の製品や型を流用することなく、特定クライアント向けに最初から開発することである。特に、ほかからの流用がまったくない場合を「フルスクラッチ」と呼ぶ。一方で、特定の業務や特定の業種において、汎用的に利用できる、既成のソフトウェアを使ったシステムがあり、そのようなソフトウェアを「パッケージ」、それを使って開発されたシステムを「パッケージ・システム」と呼ぶことにしよう。

オンプレとクラウドの関係と同じように、スクラッチとパッケージは相対的なものであり、フルスクラッチを一方の極とすると、いっさいのカスタマイズなしにパッケージをそのまま導入したシステムがもう一方の極となる。実際には両極になることは稀であり、ある程度のパーツを流用したスクラッチ、パッケージをある程度、そのクライアントの業務に合わせてカスタマイズするというのが、ごく通常のありようである。

また、システムの開発には大きく「ウォーターフォール型」と「アジャイル型」のスタイルがある。このうち、前者については別稿（谷地[2016]）にて概要は説明した。一方、後者についての詳細は要求獲得の重要性やあり方に関わるものであり、また別の機会に深掘りすることにした。この点で、本稿は「ウォーターフォール型」の開発スタイルを前提とした議論を行うことにしたい。

とはいえ、本稿ではこれらがどのようなスタイルになるかにかかわらず、要求獲得は必要であるとする立場をとる。なぜなら、要求が明確にならないと、具体的なスタイルも明確にはならないからである。

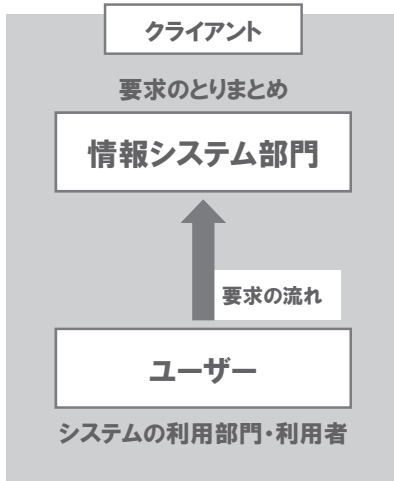
### 「要求獲得」をめぐる主体に関する認識

ここから先は、「要求」やそれを獲得するということが、いったいなにを指しているのか、掘り下げておきたい。そのために、まずは要求獲得をめぐる「主体」に関して、認識を共有しておこう。

図表1 要求獲得の主体 (1)



図表2 要求獲得の主体 (2)



要求獲得には、単純には①要求を投げかける側、そして②その要求を受け取る側がある。システム開発では、まずこの両者が誰であるのかを明確にする必要がある。

そこには基本的に2つの構図があることに留意したい。

1つに、①と②が異なる企業に属しているという構図である(図表1)。この場合、①がシステム開発の発注企業であり、②がシステム開発を行う企業である。①がクライアント、②がSlerである。

これがごくふつうの構図であるなか、2つめは①も②も同じ企業に属している構図がある。

それはクライアントの内部で①と②の間のやりとりがある、ということである。具体的・例示的には、②を担うのがクライアントの情報システム部門であり、①がシステムを利用する部門、ひいてはそこでシステムを利用するひと(ユーザー)である。そこでは、②がクライアントにおけるシステム要求のとりまとめ役となり、獲得した要求情報をSlerに渡す、という構図になる(図表2)<sup>1</sup>。

このような構図の存在がなにを意味するかというと、要求獲得というのはごく一般的なマーケティングの構図では売り手であるSlerの業務であり課題であるが、それだけではなく、買い手としてのクライアントの業務でもあり課題でもある、ということだ。

また、情報システム部門がクライアントのなかで要求のとりまとめ役になるということは、彼らを主体として、いかに要求獲得を効果的に行うかという問題があるということだ。それと同時に、獲得した

<sup>1</sup> 情報システム部門の業務内容によっては、ここがユーザーなどから要求を獲得することにくわえ、自らが要求を持ち、それをとりまとめてSlerに伝えることもある。それは、稼働したシステムの保守や運用を担当する場合が典型であり、それを遂行するに必要な要求や要件が対象となる。要件に照らすと、後述する非機能要件に大きく関わってくるのが、この情報システム部門自体ということがある。

要求をSIerに渡し、そのうえでSIerとの間でシステムのあり方について云々していくということは、彼らはシステムの調達管理者という顔も持ち合わせていることを意味する。いうなれば、彼らは売り手と買い手、2つの顔を同時に持った働きをするのである。

さらにいえば、システム導入プロジェクトをめぐるのは、クライアント企業にとっても情報システム部門がプロジェクト成功のカギを握っている。と同時に、SIerからしても、クライアント企業の情報システム部門の要求獲得能力がカギとなってくる。

とはいえ、売り手企業から見て、顧客と言ってもそこには複数のステークホルダーが存在することはBtoBのマーケティングでは珍しい話ではない。たとえば、企業がPC端末を利用する場合、そこには実際にその端末と向き合って作業を行うユーザーがいるが、このユーザーが端末の購入プロセスに関与するとは限らない。購入、言い換えれば調達にあたっては、それをもっぱらとする別部門が存在していて、そこが売り手企業との窓口になっていることは、むしろふつうである。

したがって、調達を担当する部門が窓口となってユーザーの実態や要求を汲み取り、それを購入品目確定に活かすことが行われている。

だが、情報システムの場合は、窓口役とユーザーを明確に分けることができないところに特徴がある。すなわち、この場合の窓口役は情報システム部門になるが、この部門はユーザーのためにシステムの導入・改修を司るだけではなく、ひとたびシステムが稼働すると、それを運用したり保守することに責任を持つ。つまり、情報システム部門は要求獲得をはじめとするシステム導入の窓口であるとともにユーザーでもあり、稼働したシステムをめぐる自らが要求や要件をSIerに投げかける。この場合、要件に照らすと、後述する非機能要件に大きく関わってくるのが、この情報システム部門である。SIerサイドからすれば、情報システム部門の能力やそこの関係のあり方が、自社の要求獲得成否を左右することになる。

### 要求とその獲得に関する認識

「要求」と「要件」 つぎに、システム開発の場で使われる言葉として、「要求」と密接に関連した「要件」について整理しておきたい。

まず、システム開発の全体プロセスを示したのが、図表3である(谷地[2015])。要求獲得はシステム開発プロセスの最初のフェーズであり、しばしば超上流工程と呼ばれる。

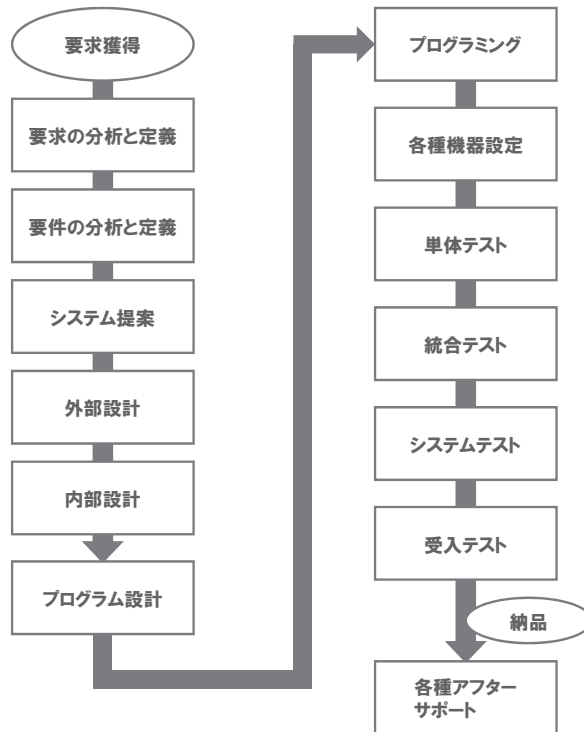
要求が獲得されると、それは「要件」へと翻訳されることになる。要求と要件は表裏一体と見なされる。すなわち、要求はクライアント、特にユーザーから見て、どのような業務をしたのか、その業務に関してシステムによってどんなコトをしたいのかを述べたものである。

一方で、要件とは要求を実現するために、システムが何をしなければならないのか、何を実現しなければならないのかを述べたものである。要求がユーザー、要件がシステムを視点に述べたものであるところに違いがある。

とはいえ、要求が明確でないと要件を決めることはできない。要件が決まらないとシステムの開発はできない。だからこそ、要件のまえに要求を獲得することが重要だ、ということになる。

要件から見ると、そこには大きく2つの領域がある。「機能要件」と「非機能要件」である。この分け方は、まずは機能要件があり、それ以外の要件が非機能要件であるとするものである。だが、後者に関してはこの言葉からはそれがいったい何を指しているのか、判然としない。

図表3 システム開発のプロセス



(谷地[2015]より作成)

システムの設計や開発で、機能要件はシステムの動作内容について定義されたものである。そして、非機能要件とは、システムが動作する方法が定義されたものである。

機能要件は要求に対するシステムの挙動という形で記述される。具体的には、システムの動作が明示されたり、数学的関数として表現されることもあれば、機能モデルとして説明される。非機能要件は、機能の全体的特性が、システムが要件を満たさなければならない形で記述される。その典型は「性能」である。仮にある特定の機能をシステムが実現できたとしても、アウトプットが出るまでに長時間、ユーザーを待たせるようであれば、その業務は滞ってしまう<sup>2</sup>。

なお、何が非機能要件なのかに関しては、2つの考え方がよく用いられる。1つは、日本情報システムユーザー協会 (JUAS) による「非機能要求仕様定義ガイドライン」(JUAS[2008]) である。そこでは10項目が挙げられている。①機能性、②信頼性、③使用性、④効率性、⑤保守性、⑥移植性、⑦障害抑制性、⑧効果性、⑨運用性、⑩技術要件である<sup>3</sup>。

<sup>2</sup> 「機能」と「性能」を峻別するところは、システム開発、システム工学の分野では特徴的に思われる。両者がまとめて議論されることは、技術経営論やイノベーション論では少なくない。

<sup>3</sup> このなかで①の「機能性」は機能要件・非機能要件の前者とは異なることに留意が必要である。非機能要件のなかの機能性とは、クライアントのニーズを満たす機能が提供できているかどうかであり、機能要求そのものではない。仮に個々の機能が一通りそろっているとしても、それがユーザーのニーズに即した形で配置されているかは別の問題であり、機能性は後者を問うものである。たとえば、ユーザーの業務の流れとシステムのインターフェース上で確認できる情報や画面遷移との間でギャップが存在すると、機能性には問題があると見なされる(野中・東[2014])。

2つめに、非機能要求グレード検討会が出した「非機能要求グレード」では、以下の6つの大項目が挙げられている。①可用性、②性能・拡張性、③運用・保守性、④移行性、⑤セキュリティ、⑥環境・エコロジーである<sup>4</sup>。

ここから読み取れるのは、システムを使ってクライアントがどのような業務を実現したいのか、それをシステムの持つ要件という視点で見たのが機能要件であり、そのような機能を実際に実現するにあたり、必要なシステム自体の備えるべき要件や、システムを長きにわたって使用していくにあたって必要となる要件を示したものが非機能要件である、ということである。

このような要件から見たとき、要求は機能要件、非機能要件いずれにも落とし込めるようなものとして獲得される必要がある。言い換えれば、いずれの要件も要求獲得対象になると本稿では考える。

「要求」とその獲得 そのうえで、ここでは「要求」に関して検討を進めよう。

要求は、もっぱら「言葉」として表現され、言葉としてやりとりされる。これを認識しておくことが重要である。この言い方には、つぎのような意味がある。

1つに、言葉はあくまでも媒介であるということだ。なぜか、それは、クライアントの人間が、まずその頭のなかにイメージとして要求を抱いているところにある。頭のなかに要求定義書が巻物のごとく存在しているのではない。また、システムのユーザーは、業務を遂行しているのであって、つねにシステムに対する要求を意識しているわけでもない。

2つめの意味は、情報システムというものが、もっぱら「機能」として構成されることである。情報システムは物理的な規定が難しい。ユーザーはPCなど、端末に対峙し、インターフェースを見ながらコマンドを出し、作業を行う。その意味でユーザーは機能を利用しているのであり、システムはユーザーに対してサービスを提供していると見ることができる。一方で、端末のみならず、システムはサーバーやストレージ、ネットワークなどの機器から構成されており、それら機器によってサービスが提供される。なので、システムを物理的な具象物として捉えることもできる。このように、システムには大きく2つの認識方法がある。

そこで、情報システムを利用するユーザーの視点に立てば、どのようなシステムが期待されるのか、現行のシステムにどのような不満を感じているのかといったことは、もっぱら機能に関してであり、それは言葉をもって語られることになる。目の前に物理的な具象物としてのシステムがあって、それを前にして網羅的に語るというものではない<sup>5</sup>。

ここから、情報システム部門やSIerは、言葉を通じてユーザーが求めている機能、現行の機能をめぐる不満を要求として得ようとする。しかし、認識しておくべきなのは、「言葉を収集すること」自体が目的ではないし、それが要求獲得のすべてではないことである。その言葉の背景を理解することが重要となる。形式化された言葉には背景がある。それはニュアンスやコンテキスト

<sup>4</sup> 正式には「システム基盤の発注者要求に見える化する非機能要求グレード検討会」という名称で、システム開発時における非機能要求の意識あわせが難しいという問題認識から、それを体系化するために発足した。メンバーは国内SIer 6社 (NTTデータ、富士通、NEC、日立製作所、三菱電機インフォメーションシステムズ、沖電気) で、2010年2月に「非機能要求グレード」の完成を受けて解散した。現在は情報処理推進機構に引き継がれている。成果は以下のURLにアクセスすることで入手できる。 <https://www.ipa.go.jp/sec/softwareengineering/std/ent03-b.html> なお、いずれの非機能要求類型も、ソフトウェア品質評価に関する国際規格ISO/IEC9126をベースとしていると思われる。

<sup>5</sup> 物理的な具象物としてユーザーが目にするのは端的にはインターフェースの画面である。

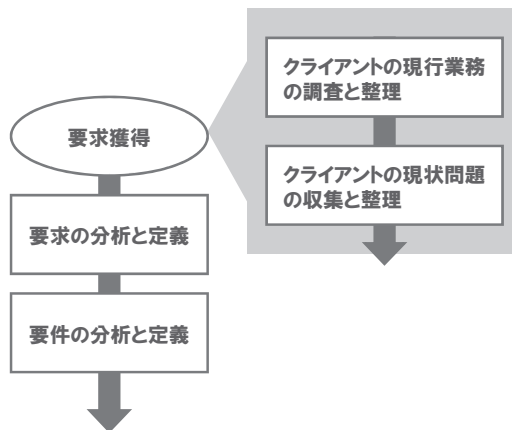
とも言われるが、知識経営学では暗黙知と位置づけられているものである (Nonaka=Takeuchi [1995]). そのように、ユーザーが「語っていない」「語りえない」暗黙的な知を読み取らねばならない。そうすることで、SIerや情報システム部門は、もともと顧客の頭のなかにあったイメージを再構成できるようになる。このように要求とその獲得を認識する必要がある。

言い換えると、言葉としての要求の背後にある暗黙知をユーザーは語るができない。SIerや情報システム部門はそれを読み取ることができない。ここから要求獲得をめぐる問題が生まれることになる。ということは、いかに暗黙知を読み取ることができるか。その能力が要求獲得には重要となる。ここからすれば、要求獲得問題がなかなか解消されないことや、それに対する方法がなかなか生み出されないことも必然と言える。

では、要求を獲得する主体である、SIerや情報システム部門にとって、要求を「獲得」するとは、いかなることをいうのか。

図表3で示したシステム開発プロセスの最初のフェーズを拡大したのが図表4である。要求獲得は、システムの対象となるクライアントの業務を明確にすること、そこにある問題点を明確にすることであり、これを行う主体の視点に立って、要求獲得という (水田[2011/2016])。すでに述べたように、その主体にはSIer、クライアント (情報システム部門) の一方、あるいは両方の場合がある。

図表4 要求獲得ステップ



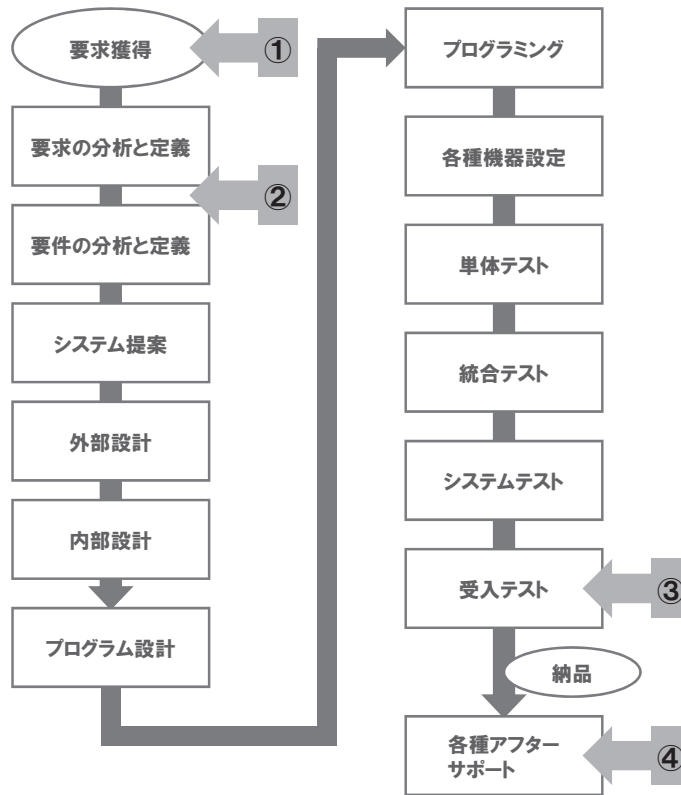
#### プロセス視点の要求獲得問題

これらのプロセスにもとづいて問題を示すと、図表5のように、4つのステップで発生するものと考えられる。

①は、要求獲得が不十分であるということで、この作業およびアウトプットの品質問題である。ここはさらに3つの問題に分岐する。1つは、要求獲得の主体が思うようにそれを実行できないという問題である。これはクライアントの組織的問題である。もう1つは、要求獲得主体の能力の問題である。これらについては谷地[2016]にて指摘されている。3つめは、要求獲得が不十分であるがゆえに、その後のフェーズで問題を生み出すというもので、それが②～④になる。



図表 5 要求獲得をめぐるシステム開発問題



まず、②は要求獲得が不十分である、あるいは要求獲得のステップが終了しないために、要求の分析、とりまとめができない、よって当然のことながらそれにもとづいた開発すべきシステムの要件も確定できないというものである。システム開発プロジェクトにカットオーバーの時期が設定されているとすると、このことは②のみならず、以降に続くフェーズの時間が減っていくことを意味する。

かくして起こるのが、③や④のような、後工程での問題である。

すなわち、要求獲得の品質が低いまま設計・開発されたシステム<sup>6</sup>がクライアントによって検証を受ける。前工程の低品質は当然のことながらアウトプットとしてのシステムの品質に反映される。かくして検証の段階になってクライアントからそのことを指摘されるのである（③）。つまり、このような最終工程になって、ようやく開発サイドのSierもクライアントも、当該システムの実品質を知ることになる。これと同じようなことは実稼働を開始した後も起こる（④）。つまり、納品をしてもクライアントから問題点を指摘される。これに対してSierは都度改修作業を行うことになり、ここに工数をとられていく。

<sup>6</sup> 要求獲得の品質が低いという意味には、要求が十分に獲得されていないという意味と、精確に獲得されていないという意味の2つがある。常識的には、このような状態で製品を設計・開発することは不可能であると思われる。しかし、こと情報システムではそれができてしまうところにも問題の原因がある。これについては谷地[2016]を参照のこと。

### Sier視点の要求獲得問題

要求獲得の不十分さは、クライアントの満足度にダイレクトにつながる。ここは理解しやすい。一方、これがSierにとってどれほど重大な問題なのか、これを明確にするには、実はシステム開発における典型的な契約形態について言及しておく必要がある。

まず、クライアントがSierに支払う金額がどのように決まるのかについてである。

システム開発では、それは「工数積算」「人月積算」と呼ばれる(斎藤[2014])。これは、開発を行う人間に焦点を合わせ、その人間が1ヶ月業務を行うとして、それにどれくらいのコストがかかるかを決めておく。そのうえで、当該案件の開発ではどれくらいの間が、どれくらいの間、業務に従事するかを計算し、その合算としてトータルの金額を求めるというものである。たとえば、ひとりあたり月額100万円で、システムの開発に10ヶ月かかるとしたら、1,000万円という金額になる。そして、これを業界では人月という単位で表記するのである。この場合は10人月になる。

このような金額算定方法になっているのは、1970年代終わり、IBMのアルブレヒト(A.J. Albrecht)が考案した「ファンクション・ポイント(FP)法」に源流をたどることができる<sup>7</sup>。この方法では、プログラムを順に1つずつ作成すること、そこから1ヶ月間でプログラムを書く量が、ひとによって同じであることが前提となっており、それゆえに見積りの妥当性が確保されていた。しかし、その後はオブジェクト指向プログラミングといった方法が現れ、開発の生産性が向上する一方、設計の仕方によって1ヶ月あたりのエンジニアの工数が変わってくるようになった。エンジニアによって生産性が異なってくるということは、FP法的前提が崩れてしまったということである。しかし、これにとって代わる精度の高い見積り方法が出てくることもなく、実際には従来のFP法をベースにしつつ、これまでの経験によって開発の規模を推定し、積算的に見積りを行うようになったのであった(斎藤・後藤[2016])<sup>8</sup>。これはクライアントの観点からすると、見積金額の根拠が曖昧であることを意味する。

一方、クライアントから見て見積りの根拠が曖昧で、評価ができなくなると、どうなるか。それに対してクライアントは複数のSierに見積りをとる。そして各社から算定された見積りを比較することで、似たような結果になっていれば妥当と判断する。そうしてクライアントは見積金額がもっとも低いところを選定したり、あるいは同じような金額であれば、これまで自社のシステム開発に従事してきたSierを選定する。

これをSierの観点で捉えてみると、結局は見積り方法の性格ゆえに、価格重視の競争になってしまい、機能や品質といった、提案するシステムの内容とは関係ないところで案件受注の可否が決まってしまうことを意味する。いわば自分で自分の首を絞めているような話になってくるのである。こうした事情の元となっているのが、かねてから採用されてきた、工数積算・人月積算という方法なのである。

以下では人月積算という言葉で一括するとして、さてこの人月積算をベースにした見積りによって、ただでさえSierは厳しい価格競争に直面する。そのうえで、すでに示した①～④の

<sup>7</sup> これは、ソフトウェアの規模を測る方法であり、機能の数や複雑性からウェイト付けしたスコアを算出し、合計点数から開発の工数を見積もるものである。

<sup>8</sup> ただし、FP法でも経験が算定に重要であったことに変わりはない。FP法は機能数を求めるものであり、直接的に開発工数や開発期間が求められるわけではなかった。機能数をベースにして、そこにこれまでの経験・実績値を適用したうえで工数・期間を算定してきたのである。

ような問題が生まれると、どうなるかを考えてみよう。

仮に当初の金額が一定不変であるとすれば、後工程での追加・改修作業はすなわち工数増加、コスト増加につながり、順次利益が削られていくことになる。それが続くと、ひいては利益がマイナスになる、つまり赤字案件化する。

とはいえ、考えてみれば、いったんできあがったシステムに適宜改修をくわえる必要が出てきたのであれば、それはSIerにとっては見積もり外の工数が追加的に発生することを意味する。したがって、その分をやはり人月積算によってクライアントに請求すれば良いのではないか、という疑問が生まれてこよう。

ところが、システム開発ではそれができない状況がごくふつうなのである。

その原因となっているのが、「瑕疵担保責任」という契約条件である。これは、クライアントとの間で取り決められた仕様通りになっていなければ、SIerがその通りになるまで改修を義務づけられるという条件であり、システムだけではなく、民法で定められている（566/570条）。言い換えると、SIerはクライアントの望むように、成果保証の責任を負うことになるのである。

システムはすでに述べたように、人月積算で金額を見積もり、それに基づいて契約がなされている。支払金額に変更がないとすると、基本的にはSIerに改修義務が発生しても、そのコストはクライアントに課されることなく、SIerの追加コストになってしまうということである。この条件があるからこそ、システム開発の前工程で発生した問題は後工程に影響をおよぼし、それは利益という情報システム・ビジネスの基本目的に直接影響をおよぼすことになるのである。

このように、人月積算と瑕疵担保責任（成果保証）という契約が、システム開発ビジネスを根底から規定しており、そのうえで要求獲得の拙さがSIerのビジネス成果に対してマイナスの影響をもたらしているという構図がある。同時に、このような事態はクライアントのビジネスに対しても悪影響をおよぼすことになる<sup>9</sup>。ここからしても、要求獲得問題は、SIerとクライアント双方にとって重要なものであり、またいずれか一方の専心的努力だけで解決するわけではなく、双方の協力・協働が必要になってくるものであることがわかる。

## 要求獲得の失敗—2つの視点

こうして見ると、システム開発における問題の源流が、要求獲得にあることが明確になってくる。そこで、あらためて要求獲得が不十分になるのは、いったいなぜなのか、である。そもそも、不十分であるというのはどういうことなのか。そこには2つの視点がある。

1つめは、システムが備えるべき要件にヌケやモレがあることだ。これは要求獲得において、とりあげられるべき要求が結果としてとりあげられなかった、ということである。結果としてという言い方になるのは、要求獲得の段階ではそれがわからず、図表5に挙げた③や④になって顕在化するからである。

2つめは、いわば1つめの逆で、システムが備える必要のない要件が盛り込まれたというものである。これは要求獲得の情報をもとに、クライアントが必要だと考えている機能を実装したものの、システムが稼働してみると、使われていないという事態である。結果的には不要な機能が実装されてしまったわけであるが、もとをたどれば、クライアントが必要だと言ってい

<sup>9</sup> クライアントに対してどのような悪影響をおよぼすかについて、谷地[2016]で論じてあるので、ここで再論はしない。

たはずの機能である。これもまた③と④でわかることである。

以上の2つをまとめると、「あるべきものがない」「なくてもいいものがある」ということであり、源流に理由をたどったうえで、合わせて、「要求獲得が不十分である」と見るのである。

ではさらに問い詰めてみよう。なぜこのようなことが起こってしまうのだろうか。その理由のなかに、要求獲得をどのように認識するか、エッセンスがあると思われる。

1つめ、本来あるべき要求が採り入れられないという問題について、深掘りしてみたい。

この問題をめぐってまず認識すべきは、システムのユーザーは、業務については専門家であるが、システム自体の専門家ではない、ということだ。彼らは基本的にシステム・エンジニアではない。

そのようなシステム・ユーザーが日々の業務のなかで、なんらかの不満を抱き、それを解消して欲しい、こういうことができれば、といった思いを抱くことがあっても、システムの技術的な専門家ではないので、そのような不満をどのように解決すべきか、自分が実現したいことを具体的にどう実現するか、その術を知らない。

しかも、それはすぐに解消・実現されるわけではない。システムの導入や改修という、ときどきのイベントを通じて、ようやくその機会が訪れるのである。だが、そのような機会が訪れたとき、過去に感じた不満や希望を言葉として表明できるかという点、そこに網羅性や完全性を期待することが難しい。そこには、すでに挙げた知識の暗黙性が関わってくるからである。

かくして、最初の獲得フェーズで要求が出尽くすことはなく、後になって五月雨式に出てきてしまう。「そのとき（最初の獲得フェーズで）気づかなかったが、実は重要な要求」が、開発プロジェクトの後半以降も発生することになる。

そして、ここからシステム開発に対する2つの視点が生まれることになる（谷地[2016]）。すなわち、上流工程では要求を完全に獲得できない、①だからこそ、完全な要求獲得を目標にするのはナンセンスであり、後から出てくることを前提にしてプロジェクトを進める。重要なのはプロジェクトが進捗したところで新たに出てきた要求に、どう対応するか、ここに課題がある、という認識である。

一方、②不可能であることはわかったうえで、どれだけ要求獲得を完全に近づけるか、その方法を追求し、作業として実施すること、ここに課題がある、という認識である。当然のことながら、ここには要求を獲得しようと試みる側の能力が関わってくる。すなわち、要求獲得能力には、このような実情があることをふまえ、そのうえでいかにユーザーに要求を出してもらいえるようにできるか、その能力がある。そして、その能力を実現するために、これまで多くの研究努力が投入され、さまざまな要求獲得の「技法」が開発、提案されてきたのである<sup>10</sup>。いずれの是非はここでは問わないけれども、それぞれで検討の対象や具体的な課題が大きく違ってくることは認識しておくべきであろう<sup>11</sup>。

2つめ、システムが備える必要のない要件が盛り込まれるという問題に照準を合わせてみよう。

クライアントの人間、ひいては実際にシステムを利用するユーザーですら、要求をあますところなく述べることができない。このことがある一方で、実際にはシステムがどのような機能を備えているべきかについて、過剰な要求が発生するという問題がある。ここで「過剰」とい

<sup>10</sup> 本稿は「要求獲得」の認識法をテーマとしているゆえ、これらの技法に関してはレビューしない。

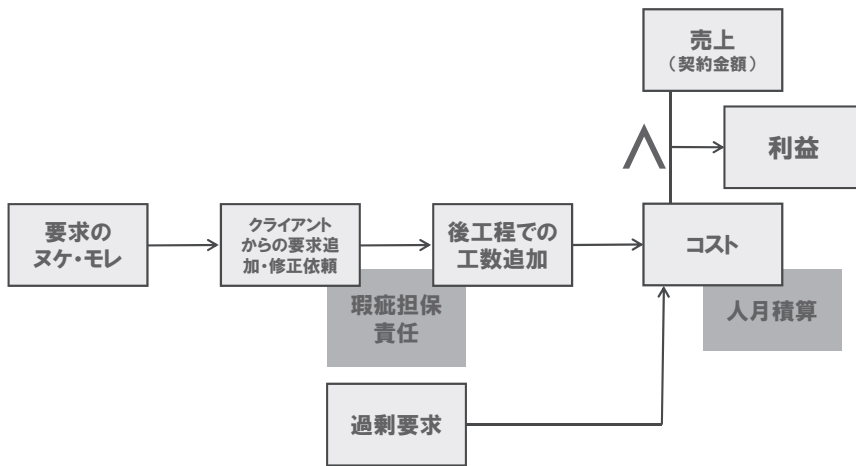
<sup>11</sup> さらに、実はここには3つめの見方がある。それは、要求が網羅されず、あとになって出てきても対応できるような、新たな開発のあり方を考える、というものである。ウォーターフォールに対するアジャイル開発発想は、その見方の産物と言えよう。

うのは、結果的には利用されないような機能がシステムに実装されてしまうということである。そして、機能が過剰になるのは、結果的に使われないことだけが問題なのではない。それが開発工程に組み込まれることで、無駄な工数になることが問題なのである。言い換えれば、工数増大は開発への負荷となり、それは品質の低下へとつながる。

要求獲得フェーズをはじめとする上流工程で、クライアントから提示される要求が多くなり、開発対象となる機能が多くなると、SIerの開発工数もまた増えていく。開発工数が増えていくということは、すでに述べた人月積算原則からすればコストが上がる。そして、それは見積金額といった価格に反映されることになる。

そのはずであるが、しかしそうならないことがある。1つは、すでに挙げたSIer間の競争関係である。機能増をそのまま価格に反映させていると、やもすれば競合のSIerに仕事を奪われてしまうかもしれない。それを考えると、おいそれと価格に反映させることができなくなるのである。

図表6 SIerのプロジェクト成果から見た要求獲得問題



一方で、クライアントのユーザーとしては、いまのうちに言えるだけのことを言うておくべきだということで、文字通り思いついた要求や機能をSIerや社内の情報システム部門に投げかける。そこに優先順位のような意識はなかなか生まれにくい。そもそもどのような切り口で順位をつけるのか。なにも切り口がなければつけようがなく、「どれも重要だ」という話になってしまう。これこそが結果的に不要な要求につながるのである。

以上、要求獲得の不十分さが有する2つの側面と、SIerの収益に対するインパクトを図式化してまとめたものが、図表6である。

### 要求獲得をめぐるクライアント組織の認識

業務に情報システムが導入される場合、それは業務のあり方自体をこれまでから少なからず変更させることがある。むしろ、当該部門の業務のあり方を見直すことを目的に、システムが

導入されることもある。また、ある部門の業務にシステムを導入すると、その部門にとどまらず、ほかの部門の業務にまで影響をおよぼすこともある。この点に関しては、すでに拙稿で述べた(谷地[2016])。

このようなシステム開発プロジェクトでは、システムをどうするかを考える以前に、業務がどう変わるのか、どう変えるのかについて検討する必要がある。そこでは、最終的なシステムの内容を決めるまでに、関係部門、関係者の合意形成が重要になってくると思われる。

まず、そのようなクライアント組織内部での合意形成をめぐり、誰が調整役となるのかがポイントになる。その場合、情報システム部門が担当になることが考えられる。あるいは、SIerがその役を背負うということもある。あるいは、これら二者が協働して調整にあたることも考えられる。

このことは、合意形成のための社内調整能力というものが、クライアントの情報システム部門やSIerにとってカギになってくることを意味している。なかでも、どこまでSIerがクライアントの社内調整にコミットするかは検討のしどころである。そこでは、立場、プライオリティの異なる、さまざまな人間とコミュニケーションをとり、相手の要望や意見を引き出し、要求や要件について合意を固めることが求められる。このようなスキルが、高いレベルで必要になるのは、要求獲得をはじめとする上流工程ならでのことと思われる(水田[2011])。

では、なぜこのようなスキルが重要になってくるのか。それは、システム導入や改修をめぐって、関係者の間には意見の相違が出てくることがあるからだ。

もっとも深刻なのは、ある部門から反対意見が出てくることである。その反対は、まさに業務のあり方が変わることに對するものである。これは従来から言われてきた組織変革の問題と考えられる(金井[2004]/山岡[2015])。そこには、業務に関する知識やスキルをイチから獲得し直さなければならない、それによる現業への悪影響を危惧してのことや、業務のあり方が変わることが組織内部でのパワー体系の変化と捉えられ、システム導入、業務体系変更に抵抗する、といったことがある。このような反対論は、システムが導入される部門からも、それによって業務上の影響をうける別部門からも起こりうる。

このような反対論に対してどのようにアプローチをかけ、システムの要件定義に向かって合意形成をはかっていくか。その作業は情報システム部門であれSIerであれ、大変な困難が予見されるものである。また、反対論ではなく、異なる部門間でシステムに対して異なった要求が出てくることも想定される。さらに、部門間の要求に相反が見られるような場合が出てくることも想定される。このような事態では、要求を獲得できたとしても、部門間で異なったり、相反するようなものをいかに要件に落とし込むか、ここで難航が予想される。

### 要求獲得問題の認識と課題

本稿では、システム開発において必要な要求獲得に関して、その営為をどのように認識するか、またそこでの問題をいかに認識するかについて、検討してきた。その作業を通じて、そこにはBtoB(産業財)の世界のなかでもユニークな特徴がある一方で、実はシステム開発だけに閉じたものではなく、ほかのBtoB領域でも想定される課題があることを指摘するのがねらいであった。最後にこの点を整理しておこう。

システム開発のオーバーラップ性 図表3でシステムの開発プロセスを見、また図表4で上流工程を見たうえで、そこで起きている要求獲得問題を見ていくと、システムをめぐっては企画、開発、そして製造工程がオーバーラップしている、つまり同時並行的に活動が実施される場合があることがわかる。

これは、システム開発に対する従来の見方が、非オーバーラップ、言い換えればシーケンシャル型であるとするものと、反する見解となる。従来からシステム開発は工程間分業がなされ、前工程が終了して、はじめて後工程に引き継がれ、作業が進行していくのがもっぱらで、それはウォーターフォール型と呼ばれてきた(谷地[2016])。

しかし、よく見ると、システムというのは企画と開発、開発と製造の切り分けが難しいところがある。システムの言葉でいう上流工程、すなわち要求獲得、要求分析、要求定義、そして要件定義では、要求獲得、要求定義、要件定義を企画と開発の両方にあてはめることができるし、要件定義からプログラミング、コーディング、ハードウェアの選定、各種テストといった下流工程にあっては、それぞれが開発と製造にあてはめることができる。つまり、システム開発の世界に閉じた視点であればシーケンシャル型に見えるが、ほかの商品と比較してみると、そこには工程間の重複があり、ここではそのことをしてオーバーラップがあると述べている。

問題は、このオーバーラップが、システムではどのように作用しているか、である。

なるほど、たとえば自動車では、開発工程内部においてオーバーラップがあったり、開発工程に量産のための工程開発が前倒しで組み込まれることは、従前から明らかにされてきた(Takeuchi=Nonaka[1986]/Nonaka=Takeuchi[1996], 藤本[1997])。しかし、システムの場合はオーバーラップ対象となる活動範囲が、実に広いことが特徴と言えそうである。

ここで重要なのは、自動車はそれによってリードタイムを縮減したり、結果として手戻りを抑制して開発工数(コスト)を縮減したりと、開発面での競争力向上を「意図」したものであるとされてきた一方<sup>12</sup>、一見するとオーバーラップであるものの、その内実は工程間での行きつ戻りつ、すなわち単に手戻りが発生しているにすぎない場合がある、という指摘があることだ。オーバーラップによって前工程での情報が後工程に向けて早期に伝達されることで結果的に後工程のリードタイムや工数が縮減されるのが理屈となる。しかし、肝心の前工程情報が後に変更されることがあるため、結局は後工程の作業に手戻りが発生したり、このような事態が予想されると後工程の担当が前工程終了まで作業に着手しないという事態が発生するというのである(糸久[2012])。

本稿で見てきた要求獲得問題とは、まさにこのような「見かけのオーバーラップ」での問題とみることができる。システム開発の場合は、そのようにならざるを得ないところがあり、オーバーラップに期待される効果とは真逆の結果につながることになる。そして、このような問題の発生源となっているのが、超上流の工程である、要求獲得フェーズである。

一方、これまでの製品開発研究で指摘されてきたオーバーラップとは、あくまで開発工程という企業内部での活動、もしくは当該企業とサプライヤーとの関係をもっぱらの対象としたものであり、顧客との関係が対象となることはまずなかったと言える。むしろ顧客の要求に関しては、それを所与とし、コンセプトというインプットに縮約して、以降の開発活動を分析することが多かった。

<sup>12</sup> なお、オーバーラップでの問題解決に対する視点は「問題(あるいは問題解決)のフロントローディング」と呼ばれてきた(Thomke=Fujimoto[2000])。

しかし、システム開発では、要求獲得という、開発に入るまえの超上流工程での活動が不十分なままに要件定義に至り、ひいては設計・開発にまで進んでいく。そして、そのプロセスで要求獲得の不十分さが露呈する。それがわかるのが、システムがかたちをなしていく下流工程であり、クライアントからの反応である。その反応は、要求追加であったり、修正であったりする。ここで再び要求獲得が発生することになり、設計・開発の手戻りとなる。これが表面的には工程のオーバーラップに見えるのであり、実際はシーケンシャル型開発での手戻り問題と同じである。そうして、リードタイムは伸びていき、カットオーバー、検収が遅れていく。SIerの工数は増加するが、受注金額は不変であるため、SIerの利益が縮減していく。

要求獲得というクライアントからのニーズ情報獲得フェーズを根因としてシステム開発をめぐる業務の見かけのオーバーラップが発生するのであり、ここにシステム開発を対象とした要求獲得のあり方を問う必要が出てくることになる。

組織における要求相反 SIerにとって、その商品たる情報システムは、クライアントの業務体系を大きく変えるという特質を持つ。であるがゆえに、SIerは自ずとクライアントの内部組織問題に多かれ少なかれ関与せざるを得ない。マーケティングにあっては、顧客のことをよくよく知らねばならないという一般則など、あまりに当然至極のことであり、そもそもそれを知らねば商品自体を企画できない。しかし、ほかの分野であれば、極端に言えば、顧客をよく知らなくても商品を企画することは不可能ではない。問題はそのような商品が顧客に受容されるかどうか、受容してくれる顧客が存在するかどうかである。

だが、システムはそれとは異なる。顧客と相対するなか、相手のことを知らねば企画自体が不可能なのである。つまり、なにをつくるべきかが顧客とのやりとりのなかで見えてくること、これが基本であることを出発点としなければ、問題をつかまえることもできないのである<sup>13</sup>。

だからこそ、相手のことを知る作業がひとえに重要となってくる。これを要求獲得と呼ぶ。そして、その要求獲得にこそ、難しさがある。再び一般的な言い回しである「顧客」を「クライアント」に戻すと、クライアントとの相互作用とは言うが、なによりもクライアントと言ってもそこには立場の異なる部門・人間がおり、システム導入や改修に対する基本的見解に違いがある。そこでどう合意形成をなすかが問題となる。

このような問題は、それ自体は指摘されてきてはいる。だが、それにどうアプローチするかということでは、結局はクライアントのことをよく知った上で考えるべきという話で終わっており、マーケティング研究ではほとんどこれといった成果を見ることができない(高嶋・南[2006])。ここはシステム開発に限らず、組織を顧客とするBtoBマーケティングでの残存課題と思われる。

顧客リサーチ研究の適用可能性 また、クライアントの人間が、どのようなシステムを求めているか、そこに結びつくような情報(要求)を上流工程で提示することが難しい。だからこそ、後工程に向かうなかで五月雨式に追加の要求が提示されたり、当初の要求にもとづいた要件のなかに変更のリクエストが出てきたりする。さらにだからこそ、システム開発では手戻りとい

<sup>13</sup> なお、要求獲得が不十分であるにもかかわらず、以降の要求定義から要件定義、ひいては設計・開発まで工程を推し進めることができるところに、システム開発の特徴がある。ここは谷地[2016]を参照のこと。



うかたちで企画、開発、生産の各工程が並行展開されてしまうのである。

「何が欲しいか」が明らかになって、「何をつくるか」が決まることに変わりはない。だが、「何が欲しいか」、すなわち要求がその獲得フェーズで出尽くすことはなく、後の工程で出てくる。なかでも、開発フェーズやテスト・フェーズのように、クライアントがシステムに触れることができるようになって要求が出てくることがある。そこから新たに設計・開発が行われる。これは、「何が欲しいか」と「何をつくるか」が一過的、シーケンシャルに進むのではなく、繰り返されることを意味する。これがシステム開発における工程のオーバーラップである。

そのうえでどのように作業を展開するか。そこにはすでに述べた2つのアプローチがあるが、ここで注目するのは2つめのアプローチである。上流工程でクライアントの要求をあますところなく獲得することはむずかしい。だが、そのうえでなお、いかにそれに近づけていくか、その方法を問うていく必要がある。このとき、顧客のニーズを引き出すためのマーケティング研究の知見が活かされると考えられるが、そこで重要となるのが、今回検討してきた要求獲得をどう認識するかである。

なかでも、インターフェースくらいでしかシステムに触知できないというなかで、それを扱うユーザーが、いかにして要求をモレなく言葉で表明してくれるか。そこには、ユーザーの頭のなかに言葉として要求が入っているのではなく、あるのはイメージであって、それを言葉に形式化するプロセスがある。そして、言葉自体の背後には状況という名の暗黙知がある。SIerやクライアントの情報システム部門はこの認識のうえで言葉としての要求とともに、背後の暗黙知を引き出す必要がある。それがあってこそ、要求はより精確に要件へと翻訳されることになる。

ここで、ひとくちに顧客リサーチの研究がマーケティングにはあるといえども、そこにはBtoCを対象にしたものもあれば、BtoBを対象にしたものもある。単純に見れば、情報システムの分野で利用できそうなものはBtoBを対象にした研究と考えられる。だが、実は要求獲得を本稿のように認識したうえで顧客をリサーチするような状況を対象にした研究は意外に少ない。むしろ、BtoC対象の研究の方が、顧客という生身の人間の有り様をベースにニーズを引き出すことを考えており、こちらの研究の方が利用できる可能性がある。この意外性を看過すべきではないと思われる。

## 参 考 文 献

- 藤本隆宏 (1997) 『生産システムの進化論—トヨタ自動車にみる組織能力と創発プロセス』有斐閣。  
糸久正人 (2012) 「オーバーラップ型製品開発におけるフロントローディングの効果」『赤門マネジメント・レビュー』第11巻第4号, 237-254ページ。  
金井壽宏 (2004) 『組織変革のビジョン』光文社。  
水田哲郎 (2011) 『手戻りなしの要件定義実践マニュアル』日経BP社。  
水田哲郎 (2016) 「要件定義の実践テクニック」(第1回～第6回) 日経SYSTEMS, 2016年5月～10月。  
Nonaka, I., H. Takeuchi (1995) "The Knowledge-Creating Company : How Japanese Companies Create the Dynamics of Innovation", Oxford University Press (梅本勝博訳『知識創造企業』東洋経済新報社, 1996年)。  
野中誠・東基衛 (2014) 「ソフトウェア非機能要求の定義—品質の良いソフトウェアを作るために」『情報処理』第55巻第1号, 31-37ページ。  
斎藤昌義 (2014) 『システムインテグレーション崩壊』技術評論社。  
斎藤昌義・後藤晃 (2016) 『システムインテグレーション再生の戦略—いまSIerは何を考え、どう行動すれ

- ばいいのか?』技術評論社.
- 高嶋克義・南知恵子 (2006) 『生産財マーケティング』有斐閣.
- Takeuchi, H., I. Nonaka (1986) "The New Product Development Game", Harvard Business Review, January-February, pp.137-146.
- Thomke, S., T. Fujimoto (2000) "The Effect of "Front-Loading" Problem-Solving on Product Development Performance", Journal of Product Innovation Management, 17, pp.128-142.
- 谷地弘安 (2015) 「情報技術の業界外観—ITマーケティング研究にむけた基盤的ワーキング」『横浜経営研究』第36巻第2号, 118-131ページ.
- 谷地弘安 (2016) 「IT産業における「システム開発」の要求獲得問題—BtoBにおける顧客リサーチ研究にむけて—」『横浜経営研究』第37巻第1号, 219-234ページ.
- 山岡徹 (2015) 『変革とパラドックスの組織論』中央経済社.

〔やち ひろやす 横浜国立大学大学院国際社会科学研究院教授〕

〔2017年5月27日受理〕