

SSM の活用によるリスクの可視化の試み ～ソフトウェア開発事例から見えてきたリスクの捕らえ方～

三 浦 信 宏

1. はじめに

金融商品取引法（通称：日本版 SOX 法）が 2006 年 6 月に成立し、これを受けて各企業は 2008 年 4 月から始まる日本版 SOX 法の適応に向けて内部統制に取り組んでいる。日本版 SOX 法の特徴のひとつは、アメリカ版 SOX 法にはない、「IT への対応」が含まれていることである。このことは、総称として IT と呼ばれている情報システムに関連する全てが企業経営に不可欠な要素として認知されたことを意味する。

従来、情報処理と言われていた業務は近年、IT サービスと呼ばれてはいるが、その内容はデータ処理、システム開発や運用保守関係等々で作業レベルとしては変わり映えはしないが、以前と大きく異なるのは、作業の外部委託化、つまりアウトソーシングが主流になっていることである。しかも、現在では、トーマス・フリードマン著の「フラット化する世界」等でも紹介されているように、アメリカの企業では、情報処理に関する作業は時差を利用してインドに外部委託している傾向を見ると、世界の中で、インドがアウトソーシング市場をリードする基盤作りをしている現状が見てとれる。IT サービスは既に、国境を越えているのである。

筆者は、コンピュータメーカーにおいて情報システムの開発を担当するプロジェクト・マネージャとして勤めてきた。現在、グローバルスタンダードとして Project Management Institute（以下 PMI）から発表された Project Management Body of Knowledge（以下 PMBOK）によると、プロジェクト・マネジメントとは「プロジェクトの事業主体や他のステークホルダーの当該プロジェクトに対する要求事項や期待を充足する、またはそれ以上の成果をあげるために、最適な知識、技術、ツールそして技法を適用すること」と定義されている。

本来、システム開発におけるプロジェクト・マネージャの仕事とは、顧客の要求する機能を具現化したソフトウェアを、決められた予算と期間内に開発することである。そのためには、プロジェクト計画の立案、トラッキングそして外部委託を含めた要員管理能力とともに、技術面と管理面の全てにわたるリスク・マネジメント能力が強く要求される。

この 30 年間、情報システムを取り巻く技術や開発方法論は数多く提案されてきたし、前述のグローバルスタンダードとしてのプロジェクト・マネジメント手法も整備されてきた。しかし、情報システムの品質の悪さが原因で、社会生活に大きな混乱をきたす現象が多々起こっている。本来、情報システムの品質というのは、リスク・マネジメントの視点から厳しく把握していくものなのである。それなのに、日経コンピュータの「動かないコンピュータ・フォーラム」に

見られる意見としては、その大部分が技術論や方法論ではなく、いわゆる“不条理”なプロジェクトの事例なのである。これは、筆者も長い間、システム開発を担当してきた経験から、現場の状況は手に取るように理解できる。

ソフトウェア開発というのは、他のハードウェア開発プロジェクトのように、そのプロセスが関係者に分かるような可視化（見える化）がしにくいのが特徴である。不可視化な製品開発においては、リスク・アセスといっても、リスク認識にはチェックリスト等と照合する作業が大部分を占めている。確かに経験者の蓄積された過去のノウハウをチェックリストに反映させることは、有効な手段のひとつであると言える。

しかし、ユーザー企業、自社、その他の政治的な諸事情からくる不条理な動きをチェックリストとして表現できるであろうか。そこで、本論文では、過去のソフトウェア開発プロジェクトの時代的背景とパラダイム・シフトによる環境の変化を整理し、この非科学的、または、ただの主観的発想として葬られてきたチェックリストには現れにくいリスクに対する感受性、つまり、リスク・マネージャとしての嗅覚を可視化できる方法をいくつかのメソッドロジーを使って試みることにした。

尚、本論文での“リスク”とは、例え、その情報システムが防災用のものであったとしても災害等の自然現象やテロ対策等に関する脅威となる可能性のものは扱わず、あくまでもソフトウェアの完成に対して損失をもたらす脅威が具現化する可能性のことを言い、投機的リスクに属し、プロジェクト・マネージャによって積極的にコントロールされるべき対象のものに限定している。

2. 情報技術の発展とパラダイムシフトによる課題

情報システムの変遷を調査していくと、先ず、ハードウェアとしては生まれも育ちも異なる大型汎用機、オフィス・コンピュータ（以下オフコン：ワーク・ステーションや後のサーバーも含む）、パーソナル・コンピュータ（以下PC）の誕生と、その技術的進歩と急激な普及の過程がある。次にプログラミング言語やそれに付随したソフトウェア群の整備と拡大、同時並行してソフトウェア工学と称する開発手法の提案、そしてプロジェクト・マネジメントに関する開発方法論と続いていく。コンピュータが世に出て60年程度というのに、この無目的な機械を取り巻く環境は衰えを知らない。その間に、コンピュータ、プロセッサ、ITと呼称は変わってきたが、本質は唯のデータ変換マシンには変わりはない。一時、IT革命なる意味不明な革命ごっこが、やたらはやったが、その結果、無目的な機械の新たな目的を発見し、ビジネスにも新しい視点が生まれ、それが年々着実に膨らんでいって今日の社会基盤としての原動力となった。

しかし、今日の情報システムの発展の本質を探っていくと、コンピュータと電話との出会いに辿り着く。電話のインフラ上をデジタル化されたデータが流れる。このことから、スタンドアロンの機械の目的が、単なるデータ処理機から一気にネットワーク・セントリックの主役と

なり、利用方法が大きく変化していった。ネットワークとの出会い、その応用技術が育たなければ、今日の高度情報化社会は実現できなかったと考えられる。

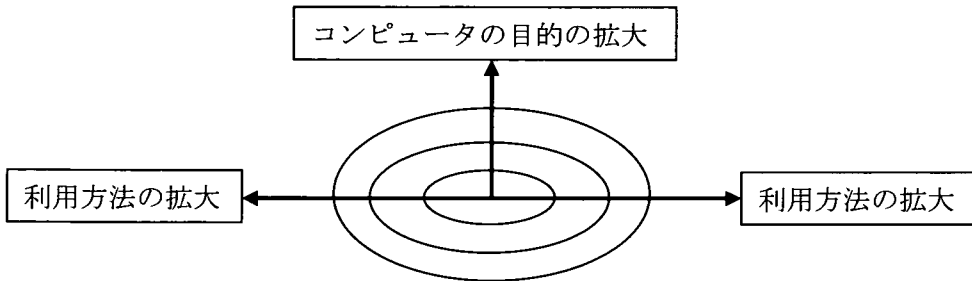


図1. コンピュータの利用方法の変化

図1で示したように、無目的な機械は、その利用方法を研究、拡大することで、コンピュータそのものの目的をも拡大してきたのである。

その後の展開は、オンライン・リアルタイム処理で代表される、大型汎用機を中心としたネットワークの構築による遠隔処理が情報システム開発プロジェクトの主役となっていった。

この頃のシステム開発環境は、主に大型汎用機のコンピュータ・メーカーがソフトウェアの開発も同時に受注し、全てのハードウェアも受注したメーカーのもとで構成されていたために全体像が掴みやすい上に、開発組織体制も外部委託はなく、全てをメーカーのSEで構成することができた。このような、開発体制が取れば、その分、プロジェクト・マネージャの仕事も、開発に対するリスク・マネジメントも今日とは比較にならないほどシンプルなものであった。むしろ、要員の経験不足や技術的な難しさから予測されるリスクの方が高かった。しかし、この頃でも、ユーザー企業の中には、機能要件を決める部署、サービス開始時期を決める部署、開発費を決める部署のそれぞれの組織が縦割りで、連携が取れてなく、不条理な条件の中での開発となったが、そのことは、チェックリスト等には、一切、反映されるものではなかった。

バブル崩壊後の長期的不況により、多くの企業では定期的な増設を伴ってきた汎用機関連の投資を抑制し、代りに驚異的な性能向上と低価格化が実現されたPCやオフコン（この頃にはサーバーと呼ばれるように姿を変えていたが）を中心とする情報システム構築が主流となり、これまでの大型汎用機を中心とした集中処理から、オープン・クライアント/サーバー・システム構想にのっとった分散処理へと変化していった。パラダイムシフトが叫ばれていた時代の代表的な情報システム構成であった。

前述したように、情報システムの品質こそが今後の企業経営に大きな影響をおよぼす中核的要素であると言える。しかし、現在では情報システムを取り巻く環境は大きく変わったのである。その範囲は、情報機器、開発環境、人的資源等全てにわたっている。この影響は、マーケティング分野に限らず、プロジェクト運営などにも顕著に現れている。例えば、製品のオープン化による構成の複雑化やそれに伴う人的資源の複雑化は、従来のプロジェクトとは比較にならないほどのシステム開発リスクを高めていった。

情報システムのパラダイムシフトは、構築環境面にも大きな影響を与える結果になった。そもそも情報システムを取り巻く環境としては、利用部門、情報システム部門、ソフトウェア・ハウス（ハードウェア・メーカーも含む）の三者間に図2のような関係がある。

先ず、誤解の無いように、ここで使用されている利用部門、情報システム部門、コンピュータ・メーカー、ソフトウェア・ハウスの言葉の意味を明確にしておきたい。

- ・利用部門とは、ユーザー企業のエンドユーザーを意味し、経営陣や現場の第一線にて社内開発されたシステムを活用している全ても部門を指す。
- ・情報システム部門とは、ユーザー企業の全社的規模のシステムを開発、保守している部門を指す。
- ・コンピュータ・メーカーは、主にハードウェアを提供する企業であるが、同時に、ユーザーのソフトウェア開発のサポートや、またはソフトウェア全体の開発を請け負う場合もある。
- ・ソフトウェア・ハウスは開発対象をソフトウェアのみに特化しており、依頼先が情報システム部門やコンピュータ・メーカーだけでなく、近年では、利用部門が直接依頼するケースが増えている。

従来の大型汎用機による集中処理システム開発では、図2の〔I〕面に相当する位置付けで情報システム部門とエンドユーザーとの間で機能要件を協業して、顧客としての要件としてまとめ、必要ならば情報システム部門の基で、ソフトウェア・ハウスの協力を得て、開発するのが一般的であった。コンピュータ・メーカーは、図2の〔II〕面に相当する位置付けで、この情報システム部門に対してハードウェアの導入やソフトウェアに関する技術的なサポートや製品に関する教育を行っていた。主役はあくまでも、ユーザー企業の情報システム部門であった。当然、プロジェクト組織も情報システム部門に置かれており、このような環境から、ユーザー企業にも多くのテクニカル・スキルや管理スキルを持った開発担当SEが育っていったのである。しかし、コンピュータ関連の知識に詳しい情報システム部門と、現場の業務関連に詳しいエンドユーザーとの間のコミュニケーションが円滑に行かず、出来上がった情報システムの使い勝手が、必ずしもエンドユーザーを満足させられるものではない場合もあった（これらのコミュニケーション・ギャップを埋める為に、“システム・アドミニストレータ”構想が出てきた）。

しかし、オフコンやPCの誕生と、それらの驚異的な低価格化と性能向上により、各企業ではダウンサイジングブームが起こり、情報システムは集中処理から分散処理を志向する傾向に変わっていった。ただ、この頃は、ブームが先行し、分散処理型のシステム運用に対するコストの高さは議論されないままだった。

分散処理となるとエンドユーザー主導で要件がまとめられ、情報システム部門から全社的なデータや技術的支援を受けながら、エンドユーザー部門が直接ソフトウェア・ハウスの協力を得て開発し、運用していく必要があるが、これを可能にしているのは、現場にシステム・アドミニストレータが根付いてきたことと、コンピュータ・メーカーが直接、エンドユーザーをサポートする体制に変わっていったことである。自社の業務に一番精通しているエンドユーザー

が自分達の満足するシステムを自らが開発することが理想とされ、業界では、エンドユーザー・コンピューティングという言葉がもてはやされていた時期でもある。

パラダイムシフトは図2の〔Ⅲ〕面を開発の現場に変えてしまったのである。この体制を推進するためには、企業としてのエンドユーザー・コンピューティングに対する明確な方針が必要になってくる。当時の資料を調査してみると、全国 IBM ユーザー研究会連合会によるアンケート調査報告 P61 では、57.5%（回答 1053 社、「方針有り」が 15%、「不十分ながら有り」が 42.5%）の企業が既にエンドユーザー・コンピューティングの方針を持っていることになる。特に金融業界などは、既に 72.2%に達している。

これにより、従来からの情報システム部門の存在は大きく変わっていった。例えば、ソフトウェア・ハウスによる SI（System Integration：統合情報処理システム・サービスとは、相手方との間に締結した契約に基づき、情報処理システムにつき、その設計、プログラム作成、試験、運用の準備、および保守のすべてを行なう役務をいう）サービスを活用して、開発の全面的な外部委託化や運用面でのアウトソーシング化等によって、仕事の内容が全社的な情報システムの企画立案とか、エンドユーザー・コンピューティング推進役として利用部門を支援するコンサルタント業務になっていった。このことは、前述のアンケート調査報告 P62 でも明らかでありエンドユーザー・コンピューティングの推進体制が利用部門にあると答えた企業が 9.9%であるのに対して情報システム部門にあると答えた企業が 45.7%になっている。但し、エンドユーザー・コンピューティングであるからといって、利用部門が PC 等で勝手に開発すればよいというものではない。あくまでも、大型汎用機、オフコン、PC を統合化した全社的な情報戦略の一部として考える必要がある。技術的な蓄積度合から考えて、推進体制が情報システム部門にあるのは当然のことといえるが、大型汎用機による集中処理的な発想を脱しなければならぬのも新たな課題である。そうでなければ、情報システム部門の存在が情報化推進の阻害要因になってしまう可能性も考えられからである。ところが、現在では、予想外の長期不況により、各企業では情報システム部門を本体から切離し、完全子会社化したり、コンピュータ・メーカー等と合併会社化して、本社の業務をアウトソーシングの対象会社として位置付けするようになった。

この開発環境の変化は、同時にソフトウェア開発に、今まで以上のリスクの増大をもたらす結果となった。それは、ハードウェアの多品種、高度化による複数化、ソフトウェアのパッケージ化の活用による複数化、ソフトウェア・ハウスを含めた開発要員の寄せ集め体制が蔓延したプロジェクトが常識化した現場、そして、相変わらず続く不条理な要求、それに堪えられないプロジェクト・マネージャ等々、どこを見ても、技術的、論理的に物事を処理すべき情報システム開発のセオリーが荒廃してしまったと思われるもしかたがない。

大型汎用機でシステムを構築していた時代は、ハードウェアの性能に制約があったために、ユーザー企業は、システム開発に慎重に取組んでいた。その後、PC が普及した後は、エンドユーザーが安価なハードウェアを購入し、パッケージ化されたソフトウェアによって、簡単にシステムが作れるようになり、長く使う気も感じられないシステムを作り散らかしている現状

が見て取れる。

一部の業界等ではオープン・クライアント/サーバー・システムに移行しても、大型汎用機から完全に脱却できず、システムの運用コストがかさんでしまう。そのことが運用部門の要員不足の問題を引き起こし、ソフトウェア・ハウス依存症につながっている。このような現状を総合的に検討し、数年前から、情報システム部門不要論が浮上しているものと考えられる。

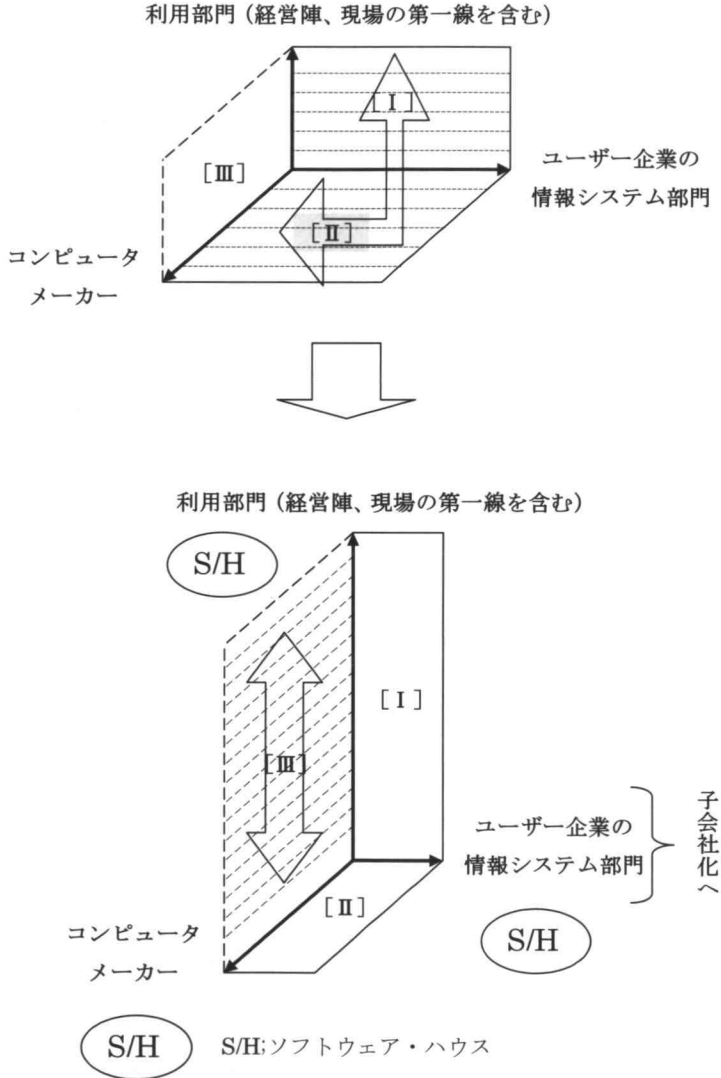


図2. 情報システム開発環境の変化

3. 情報システム開発要員の現状と課題

2007年問題がよく話題に上る。これは、いわゆる団塊の世代の大量定年退職による、技術の継承等が問題で、各業界はそれぞれ知恵を絞っていることを、よく耳にする。

情報システムの分野では、この2007年問題をどのように、乗り越えようとしているのか、調査してみた。調査期間は2004/8月から2006/10月までであるが、調査対象を、先ず運用部門に絞ってみた。定年退職対象の社員が若手に伝授している会社も多くあるが、この機会に大型汎用機での処理を、全面的にオープン・クライアント/サーバー・システムに移行するという冒険を犯そうとしている企業もあった。定年退職対象の社員の雇用延長が、大型汎用機からの脱皮を遅らせている原因である、という考え方である。若手社員の中には大型汎用機は世代遅れで興味を示さない、と言うことだが、あまりにも、短絡的で危険な解決方法であると考えられる。これは、全社的な情報処理のリスクを雪だるま式に増やす結果となることを忠告したが、どうやら、現状を明確に捉えられる経営者や情報システム関係者がいないために、コンサルタントと称するところに依頼した結果のことらしい。そのコンサルタントがどこまで責任を持っているのかは不明だが、このような方法で、2007年問題を乗り切ろうとしている企業があるのも現実なのである。

筆者の調査では、前述の企業の業務処理がいきなり分散型にとって変わるものではないと考えているが、ひとつのトレンドに流されているプロジェクト参加社員の感情面がリスクの増大を後押ししているように思われた。

そこで、さらに現状の情報システムの開発プロジェクトの現場を調査することにした。そして、システム開発に携わっている社員のキャリアについてまとめたのが表1である。

表1. 業界別 IT 要員の現状

ケース番号	業界	情報システム要員数の割合(%)	新入社時から配属されている社員の割合(%)	新入社時から希望して配属された社員の割合(%)	大学にて、コンピュータ関係学部出身者の割合(%)
1	CPG	3	52	18	15
2	CPG	5	70	15	20
3	CPG	8	60	10	18
4	CPG	10	65	15	10
5	CPG(*)	5 (6)	35 (85)	28	15
6	金融(*)	1 (12)	3 (90)	25	20
7	金融(*)	2 (8)	5 (78)	18	13
8	製造	10	78	70	80
9	通信(*)	1 (38)	80 (90)	82	76
10	運輸	10	85	45	68

CPG (Consumer Package Goods) は消費財メーカーを総称して使用している言葉である。

(*)印の企業は情報システム部門を子会社化し独立させている。よって当数字は本社に残っている情報システム部門の社員数の比率のものであり、カッコ内の数字は転籍または出向している元情報システム部門所属の社員の比率である。

この結果をみると、業界では製造、通信や運輸では、大学でコンピュータ関係を学んできた社員も多く、しかも、入社当時から変わっていないので、先輩からのスキルの継承もスムーズに行われるだけでなく、外部委託会社の管理にも慣れており、プロジェクトに関する技術的な問題は特に見られなかった。

金融は、大部分の企業が情報システム部門を子会社として独立させており、本社に残っている社員数はかなり少なくなっている。いわば、子会社との窓口業務が中心を成しているように感じた。しかし、子会社へ出向または転籍した社員は、システム全般に渡ってスキルも高く、先輩から若手への継承も進んでいるように思われた。

CPG に関しては、同じ業界内でも、それぞれの企業によって考え方に開きがあった。どこも情報システム部門に配属されて、キャリアを積んでいるのだが、外部委託依存症が強く感じられるところもあった。元々、初めから情報システムの開発を希望して、この業界に入ってきた社員が少ないために、最初は技術面を外部に頼っていたが、その内、業務内容に関しても外部委託先がスキルを付けていったため依存率が、さらに高まっていったものと考えられる。

各業界の情報システム部門スタッフが言う共通の話題は、ソフトウェア・ハウスのスキルは千差万別である、ということである。得意分野に特化しているソフトウェア・ハウスもあれば、大学時代に、リテラシー程度を学んできた社員を、まともに教育もせずに、現場に出して先輩の下でOJTによって訓練しているところもある。この傾向は筆者が現役の頃から存在していたものだが、このような体質のソフトウェア・ハウスは離職率が高いうえに、個人に依存しているために、プロジェクトの途中で辞めていかれると生産性が大きく低下して、計画を狂わす原因となる。請負契約であれば、ソフトウェア・ハウス社内で解決すべき問題であるが、派遣契約の場合は、ユーザー側に管理責任があるため要員管理を難しくしている。この現象もリスクの温床になっている。

また、複数社のソフトウェア・ハウスと契約しているプロジェクトでは、組織も複雑になり、品質を保つためには、標準化を徹底する必要がある。この標準化が観念的で、誤解を与えるようなものがユーザー企業の一部にあった。標準化は具体的に定められていないと、害になるものである。

4. 事例が語る現場の状況と課題

筆者の調査で最も注目したのは、リスクの温床の大部分は、技術的な問題や、組織的な仕組み、社内開発要員や外部委託先のスキルよりも、人間関係や権力者のエゴが約8割を占めていることである。

(1) 「本社側が充分調査せずに、他社に対抗して開発指示を出した」

このケースでは、完成時期が会社の創業記念行事の一環とされており、開発する意義を持っていないまま、プロジェクトがスタートした。完成時期だけが決まっており、その他は、それに合わせた機能を絞り込んで開発したもので、開発要員の士気がまったく上がらないプロジェクト

であった。

(2) 「合併のための権力闘争に、情報システム部門とメーカーが巻きこまれた」

このケースでは、合併前のユーザー企業が、それぞれ別々のメーカーのコンピュータで運用していたために、メーカーをも巻きこんだ権力闘争下での、新システム開発となったために、コンセンサスを得るのに時間がかかり、さらに感情的な面も加わって、開発時に重要なコミュニケーションも不足し、結局、テストが不十分のまま完成時期を迎えたため、混乱を引き起こした。

(3) 「経営者や情報担当役員クラスは、新技術に過度の期待をし、それらを使用した情報システムの開発に着手するよう、指示を出し、その後、役員が暴走していったため、プロジェクトが途中で崩壊した」

今日の情報技術の発展は、インターネット関連の発展、Web2.0、ユビキタス・コンピューティングの研究・開発等々、スピードが目覚しく、メーカーも近未来の社会像を提案しているが、セキュリティ面が解決されていないものも多々あり、時期尚早のものも少なくない。“実験”として、扱うつもりなら良いのだが、いきなり、本番業務に適用するのは、かなりのリスクを伴う。新技術に対しては機能を理解し、冷静な判断が求められる。

(4) 「開発費の削減のためにパッケージ・ソフトウェアを購入し、カスタマイズした」

このケースでは、パッケージ・ソフトウェアを購入して、早期に稼動開始を目指したものが、その考え自体に間違いがあったとは思わないが、購入したパッケージ・ソフトウェアを自分達の使い慣れている情報システムになるべく近づけようと、大幅なカスタマイズを実施したことである。基本的にはパッケージ・ソフトウェアは“そのまま使う”ことが重要であって、中身を直してしまうと、新規開発よりも、遥かにリスクを伴うものなのである。パッケージ・ソフトウェアの導入は、エンド・ユーザーにとっては、今まで使い慣れていた操作方法が変わるわけだから、当然、反発を招く場合が多く、開発部門が押し切られる形で違和感が少なくなるように、従来のシステムに近づける傾向があるが、これは、間違った発想である。

(5) 「エンド・ユーザーの参加がプロジェクト・チームを混乱させて、完成時期が大幅に遅れた」

情報システムの開発には、現場のエンド・ユーザーを参加させることは、非常に重要なことである。しかし、このケースでは、エンド・ユーザーが中途半端にコンピュータに関する知識を有しており、開発部門が、それに振り回された結果、あまりに、細かいところまで情報処理用語を使って要求してくるので、仕様書のとりまとめに時間を大幅に取られてしまった。全社的システムの開発は、情報システム部門がリーダーシップを発揮すべきである。

5. 管理手法とメソドロジー

ソフトウェア開発プロジェクトのために、今まで、多くの手法や方法論が発表されてきた。ここでは、その代表的なものを紹介し、適応上の課題を考察する。

3.1 CMM (Capability Maturity Model)

1986年に米国国防省 (DoD) の後押しで、カーネギーメロン大学に SEI (Software Engineering Institute) が設立された。これを機に、当時 IBM フェローだったワッツ・ハンフリー博士達が大学に移って、ソフトウェアの品質は開発のプロセスの品質に依存するとして、従来のプログラミング終了時のテストケースを数多く設定する方式から、開発プロセスの品質に関する研究が始まった。

この研究所で1987年に“プロセス・マチュリティ・フレームワーク”としてまとめられた手法がCMMと呼ばれるようになり、その後バージョンアップを繰り返し今日に至っている。

要約するとCMMというのは、“組織診断プログラム”とも呼ばれていて、ソフトウェア開発の各プロセス (開発の全局面) に対して“品質マネジメントの5段階の成熟段階”に沿って評価する物差しであると考えられる。(場当たりの仕事ではレベル1、定量的にまとめられていればレベル4、再現性があり、必要ならば元に復元できる仕組みを備えていればレベル5と言った様に、仕事のプロセスの品質に着目している)。

不可視性の高いソフトウェア開発のプロセスを、このような物差しにより、客観的に組織として評価できる仕組みを提案している。

3.2 CMMI (Capability Maturity Model Integration)

CMMIは、CMMをシステム・エンジニアリングやソフトウェア調達、さらには個々のエンジニアの成熟度を位置付けるPSP (Personal Software Capability) などに個別に広げ、それらを統合する形でプロジェクト全体に適用するように拡大したものである。

3.3 EVMS (Earned Value Management Systems)

以前はC/SCSC (Cost Schedule Control System Criteria) と呼ばれていたが、その後、EVMSと呼ばれるようになり1998年にANSI/EIAのガイドライン規格として発行されている。

要約すると“経営・発注視点からのプロジェクトの進め方”を評価する物差しであると考えられる。技術的達成度を評価する物差しは多く存在するが、経営面からの視点でソフトウェア開発のプロセスを評価する仕組みが無かった。Earned Valueは訳すと“稼ぎ高”となるように、あくまでもコスト評価に焦点を当てている。各開発局面ごとに、Baseline (計画値) と Earned Value を比較し、“作業量の金銭価値化”と“達成量の金銭価値化”を比較評価しながらプロジェクト全体の進捗測定を判断していくための規定 (クライテリア) が述べられている。

従来の進捗管理では“プログラムの出来具合”に関して管理していたが、これからは各局面毎に“金の掛かり具合”も評価する必要がある (このようなことは当たり前のことであるが、大型汎用機時代のシステム開発では、ハードウェアメーカーがソフトウェア開発も請け負って

きた関係で、ハードウェアの利益でソフトウェア開発のコストオーバーを補うことも可能だったために、ソフトウェア開発のみに対するコスト管理が不明確な部分があった）。

3.4 PMBOK (Project Management Body of Knowledge)

1969年、PMI (Project Management Institute) が設立され、現在、世界で約20万人の会員を要する世界最大のプロジェクト・マネジメント関連を研究する団体となっている。このPMIが1984年、PMBOK (Project Management Body of Knowledge: プロジェクト・マネジメントの基礎知識体系) を発表した。その後、PMBOKは米国標準規格としてANSI (米国標準協会) に認定され、国際標準ISO10006のベースになっている。

PMIはPMBOKに沿ったプロジェクト・マネージャに関する資格試験であるPMP (Project Management Professional) を実施し、プロジェクト・マネジメントに関するスキルの向上に努めている。

プロジェクト・マネージャの仕事の標準化として、PMBOKは9つのマネジメント領域を提案している。内容は、それぞれ、次のような機能を有している。

表2. PMBOKの構成

	マネジメント名称	マネジメントの内容
1	統合マネジメント	プロジェクトが抱えている種々の要素をバランスの取れた形にするためのマネジメント
2	スコープ・マネジメント	プロジェクトの最終目標を成功裡に達成するために必要な全ての作業が過不足なく、かつ確実に実行されることを保証するためのマネジメント
3	タイム・マネジメント	プロジェクトを工程通りに完成するために調整するためのマネジメント
4	コスト・マネジメント	プロジェクトを承認予算内で完了させるために調整するためのマネジメント
5	品質マネジメント	プロジェクトが初期の要求を満足していることを保証するために必要な一連の業務を確認するマネジメント
6	組織マネジメント	要員に、その持てる力をプロジェクトの目的に従って効果的に発揮してもらうための環境を提供したり、調整するためのマネジメント
7	コミュニケーション マネジメント	プロジェクトに関する情報の作成、収集、配布、蓄積、最終処理といった一連のプロセスをタイムリーかつ的確に行なうためのマネジメント
8	リスク・マネジメント	プロジェクトのリスクを発見、識別しプロジェクトへの影響を最小限にするためのマネジメント
9	調達マネジメント	必要に応じて外部から商品や役務を取得するための取引を扱うためのマネジメント

3.5 SSM (Soft System Methodology)

「ソフト・システム方法論」と呼ばれ、英国国立ランカスター大学のピーター・チェックランド (Peter Checkland) 教授によって開発された経営・組織管理の方法論である。このSSMの思考方法の基礎をなすのが「ソフト・システム思考法」(Soft System Thinking) と呼ばれている。これに対し、従来から使われているシステム工学 (System Engineering) の思考方法の基礎をなすのが「ハード・システム思考法」(Hard System Thinking) と呼ばれる。

チェックランド教授は、化学会社 (ICI) に研究者として15年間勤務した後、ランカスター大学大学院に30年間所属していた。最初は「ハード・システム思考法」を用いた研究やコンサルタント活動をしていたが、プロジェクトがうまくいかないことを痛感し、1970年代中頃に、SSMの原型を作り上げた。1980年代に入り、国家医療、保健関係プロジェクトにSSMを適用し、成果を上げたことが注目され、ヨーロッパでは広く利用されるようになった。

SSMの特徴は、7つのステージと呼ばれるプロセスで構成されている。詳細は省くが、まず、初めのステージでは、プロジェクト関係者が、それぞれ考えている事を、絵や言葉で表現する。この絵は“リッチピクチャー (Rich Picture)” と呼ばれ、自分の置かれている状況を絵に描いてみて、自分自身の見方や考え方を確認する。また、多くのメンバーのリッチピクチャーを持ち寄ることにより、1つの問題に対する複数の見方や考え方を理解することができるようになる。これにより、自分自身の思い込みからの脱却も図れる。

次にメンバーが、問題をどのように捉えているかを言葉で表現する。これは関連システム (Relevant System) と呼ばれている。関連システムは、メンバーに思いつくままに挙げてもらい、その後、何回かミーティングを繰り返し、関連システムを数個にまとめていく。

「ソフト・システム思考法」が「ハード・システム思考法」と大きく異なるのは、目的やゴールの決定に有効な方法論であるところである。目的やゴールを決めるということは、最上流工程局面の重要な仕事である。しかし、ソフトウェア開発の現場では、システムの要件が関係者間 (エンド・ユーザーや情報システム部門間でも) で十分に検討されておらず、開発作業全体に遅延をもたらす問題の原因を作ってしまうことになる。

システム工学は情報システム開発に、多大な貢献を果たしてきた。「ハード・システム思考法」では、システムティックに秩序だったアプローチが要求された。これは、問題を定式化し、客観的に解決策を見出す方法として、SEの基本的思考方法とされていた。

しかし、「ハード・システム思考法」をソフトウェア開発プロジェクトに当てはめて考えたとき、開発対象が明確で、要員、組織、コスト、開発期間等、全てが客観的に矛盾なく議論されてから、スタートできるプロジェクトは極めて稀なことである。特に、上流工程では要件が曖昧なまま (当事者は曖昧とは思っていないが)、スタートするケースが圧倒的に多く、そのことが、リスク・マネジメントの根幹を成していることは、多くのプロジェクト・マネージャは認識しているのである。しかも、最初から不合理的理由で作られたプロジェクトでは、本質的な目的は無いに等しい、と言える。

そこで、プロジェクトには、二面性があることを訴えてきた筆者が注目したのは、要件を決めるまでの局面に、「ソフト・システム思考法」を、外部設計以降を「ハード・システム思考法」を適用したプロジェクトの実施である。しかも、各局面の開始時に「ソフト・システム思考法」により、リスクの潜在化を調査する時間を設けることである。

図3に示すように、要件定義の段階では、経営者側の思惑や、現場での聞き取り調査等、論理的、定量的にプロジェクトの姿をイメージすることは非常に困難な作業である。多くのプロジェクトを調査してみても、失敗したプロジェクトの最大の要因は、やはり最上流工程にあると思われる。それは、大部分のプロジェクト・マネージャが、初めからシステム工学的な「ハード・システム思考法」で、要件定義をまとめていこうとしていることである。この段階(図3の①の作業)では、目的やゴールもはっきりしていない場合も少なくない。この工程では、積極的に「ソフト・システム思考法」を取り入れるべきではないかと感じた。開発対象がはっきりしてくれば(図3の②の作業)、従来の「ハード・システム思考法」による論理的なプロジェクト運営が適していると思われる。

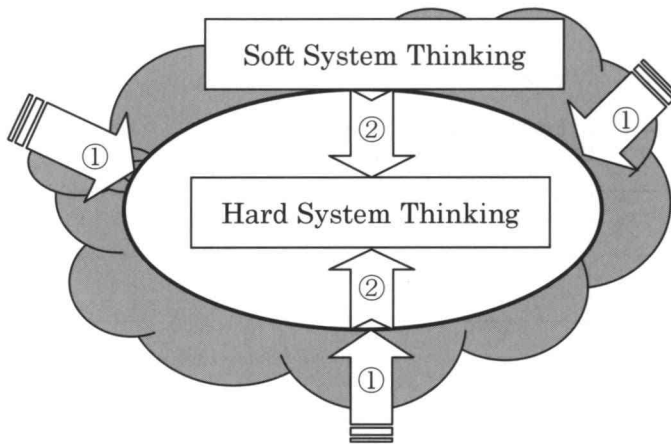


図3. プロジェクトの二面性

ただし、方法論としての芸術性や論理性を追求するあまり、現場での実践可能性と乖離が起きないよう注意が必要となる。

6. 開発局面とSSMの適応

筆者は、現在進行中のあるプロジェクトを、担当のプロジェクト・マネージャからの依頼で図4に示すように各局面の間に、「ソフト・システム思考法」を使ってみた。特に重要な人間関係を表すには、組織図を活用して絵を描き、これをプロジェクト共通のリッチピクチャーとした。これだけで、どの局面にどのような要員が関わり、それぞれのスキル・レベルを記入することにより、プロジェクトの弱点を浮き彫りにすることができる。同時に、それぞれの分野

の意思決定者や責任者も明確にできるので、各人の自覚を促すのにも好都合であった。特に、SSM2とSSM3には、エンド・ユーザーを参加させたが、現場の責任者にも好評を博した。

いつまでたっても変わらないプロジェクト開発現場の姿や、失敗すると項目ばかり増えるリスク・チェックリスト、このチェックリストをこなすための、仕事のための仕事を続けているプロジェクト・マネージャのために、筆者は、今まで多くの方法論を試してきたが、その目的は、全てプロジェクトに関するリスクの可視化のためでもあった。非論理的で主観的なため、本音が出にくい人間関係（プロジェクトでは最重要なはずなのに）をリッチ・ピクチャーは、表現することができる。今後、管理面と実施面で「ソフト・システム思考法」と「ハード・システム思考法」とを組み合わせ、活用することを推奨する。

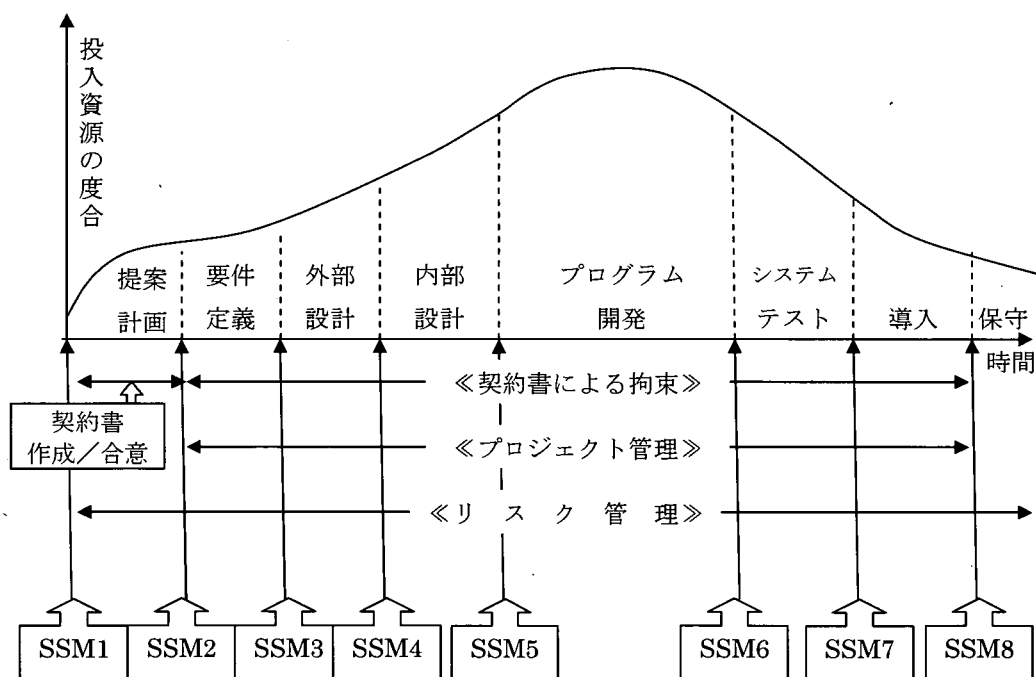


図4. 局面毎のSSMの活用

7. おわりに

リスクは現場で顕在化するが、上部組織は、現場だけで、どこまでリスクに対応できる能力があるかを把握しておく必要がある。しかし、全てを現場だけにリスク処理を任せてはプロジェクト全体の遅延につながる。PMBOKでは、プロジェクト・マネジメントの中にリスク・マネジメントを位置付けしているが、全社レベルでリスク・マネジメント・ポリシーを確立し、その管理下で、プロジェクト・マネジメントを行なっていくべきであると考え。当然、現場だけにリスクの責任を負わせない。しかし、現場を一番知っている人をリスク・マネジメント・チームに入れる必要はある。いつまでも、「結果オーライ」でプロジェクトを行なっている、

将来はない。計画的なリスク・マネージャの育成や全社的な標準化、共有化、知識管理 (Knowledge Management) ができる仕組みを持った会社が強くなる。

今回、数社のリスク・チェックリストを見させていただいて感じたことは、表現方法が単語の羅列ではなく、ストーリー的 (シナリオ化) に表現されていれば、経験の浅いプロジェクト・マネージャにも理解しやすいのではないかと、ということである。今後、このような分野にも SSM の適応を考えていきたい。

文 献

- ・福沢 恒 2000 プロジェクトマネジメント ダイアモンド社
- ・トーマス・フリードマン 2006 フラット化する世界 上・下 日本経済新聞社
- ・Gueldenzoph, L. E. & May, G. L. (2002). *Collaborative peer evaluation; Best practices for group member assessments*, Business Communication Quarterly, 65, 9-20.
- ・Gueldenzoph, L. E. & May, G. L. (2003). *The Impact of Social Style on Student Peer Evaluation Ratings in Group Projects*, Association for Business Communication Annual Convention.
- ・Johnson, C. B. & Smith, F. I. (1997), *Assessment of a complex peer evaluation instrument for team learning and group processes*, Accounting Education, 2 (1), 21-41
- ・Kotter, J. P. 1990. *A force for change: How leadership differs from management*. New York: The Free Press.
- ・Kouzes, J. M., Posner, B. Z. 1987. *The leadership challenge*. Jossey-Bass.
- ・Leimback, M. (1991), *Research use of the Social Style Profile*, Edina, MN; Wilson Learning Research and Development Corp.
- ・三浦信宏 2004 プロジェクト・マネジャーのタイプ分けと適性評価 愛知淑徳大学論集 (コミュニケーション学部篇) 第4号
- ・三浦信宏 2005 プロジェクト・マネジャーのタイプ分けと適性評価 愛知淑徳大学論集 (ビジネス学部篇) 創刊号
- ・三隅二不二 1984 トップマネジメントリーダーシップのPMスケール作成とその妥当性の研究 組織科学 91-104
- ・宮川雅明 2003 [入門] プロジェクトマネジメント PHP 研究所
- ・長尾精一 2007 問題プロジェクトの火消し術 日経BP社
- ・能澤 徹 1999 国際標準プロジェクトマネジメント 日科技連出版
- ・財団法人 1997 エンジニアリング振興協会 プロジェクトマネジメントの基礎知識体系 (PMBOK Guide 和訳版)
- ・不条理なコンピュータ研究会 2006 「IT失敗学」の研究 日経BP社
- ・ピーター・チェックランド, ジム・スクールズ/著, 妹尾堅一郎/監訳 2004 ソフト・システムズ方法論 有斐閣
- ・社団法人 電子情報技術産業協会 ソリューションサービス事業委員会 編著 2007 SLAガイドライン第三版
- ・社団法人 電子情報技術産業協会 ソリューションサービス事業委員会 編著 2007 SLA チェッ

クポイント 294

- ・独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター機構 日経 BP 社
2007 ITプロジェクトの「見える化」上流工程編
- ・独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター機構 日経 BP 社
2007 ITプロジェクトの「見える化」下流工程編
- ・独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター機構 日経 BP 社
2007 ソフトウェア開発データ白書 2007