

システム開発プロジェクトの失敗リスク評価のためのGAPS²分析

GAPS² Analysis: a Method for Assessing Risks of Failures of System Development Projects

林浩一

Koichi Hayashi

ウルシステムズ株式会社

UL Systems, Inc.

要旨

本稿では、システム開発の失敗リスクを定性的に分析するために、GAPS²分析フレームワークを提案する。このフレームワークは、システム開発の失敗の主要因はステークホルダー間での知識ギャップにあると捉え、経営者との間の目的のずれであるゴール(Goal)のギャップ、システムのユーザとの間の業務の理解の差であるアクティビティ(Activity)のギャップ、プロジェクト管理が開発の状況に追従できないプロセス(Process)のギャップ、システム構築に必要な技術スキルが充足されないスキル(Skill)のギャップの4つの視点からプロジェクトの問題を分析する。

1. はじめに：システム開発の失敗リスク低減が急務

今日、情報システムは社会の隅々にまで行き渡り、社会機能の多くがその助けなしには成立できないまでに至っている。その一方で、社会的に重要な役割を担うシステムの不具合による大きな混乱もしばしば報告されている。経済の中核である証券取引所の停止や、ライフラインの根幹に関わる電話網の機能不全などの例は記憶に新しい。これほど影響の大きな話でなければ、企業内でのシステムトラブルによる業務停止や日々の業務の効率低下などの例は日常茶飯事である。また、開発プロジェクトが途中で破綻し、システムが構築できないことや、大幅に納期と予算を超過してしまうことも珍しくない。

システム開発の失敗は、そのシステムを構築している企業に大きな損害を与える。構築されたシステムが初期の目的を果たさなければ、ビジネス機会を失うばかりでなく損害賠償が求められる事態になることもある。また、システム構築の費用が予定を大幅に超過して経営を圧迫することもある。

システム開発の失敗が多く、かつその影響が深刻であるにもかかわらず、この問題が解決される兆しは一向に見えてこない。このためシステム開発が非常にリスクの高いビジネスとなり、優秀な技術者にとってSI業界やシステム技術が魅力の薄いものになりつつある。このように、システム開発の失敗リスクの低減は、SI業界ひいては社会にとって重要な課題である。本稿では、この問題の解決を図るための一手段として、失敗リスクを分析するためのGAPS²分析と呼ぶフレームワークを提案する。

2. 戦略コンサルティング手法の応用

2.1. プロセス改善を阻む要因

システム開発プロジェクトは、もともと失敗事例からの教訓を生かすにくい性格を持っている。一般に、繰り返し行われるプロセスは、失敗の教訓を生かして改善していくことができる。このために、計画(Plan)、実行(Do)、評価(Check)、改善(Action)からなる、PDCAと呼ばれるサイクルを繰り返すことで恒常的に改善を進める手法が広く使われている。しかし、システム開発のプロセスは以下のような性格があるため、このサイクルによる改善が難しくなっている。

- ・各開発プロジェクトはひとつひとつ異なる過程で進む

システム開発は、プロジェクト型と呼ばれる、目的達成に向けて必要な要員を集めた時限的な組織で遂行される。また、多くの場合、システム開発の実作業を行うのは専門技術を持つ開発ベンダーとなる。このため、開発プロジェクトはひとつひとつ、問題の領域も、関係する人物も、適用技術も異なったものになるので、繰り返しによる経験値が効果的に蓄積できない。

- ・システム開発の失敗を特定するのが難しい。

開発の成功や失敗は、しばしばシステムが構築され、稼働することができたかという観点から評価されがちであるが、それは表層的な評価に過ぎない。企業によるシステム開発であれば、投資の有効性が

問われなければならない。また、稼働していても著しく利便性が悪いシステムや、予定された期日やコストを大幅に超えて完成したシステムを成功とは言い難い。しかし、こうしたシステムの問題点が失敗として認められにくい事情がある。システム開発には多くのステークホルダー(利害関係者)が関わっているため、失敗を認めることは責任問題につながるためである。このため、システムが動作しないなどの明らかな問題でもない限り失敗とは認められにくく、また、問題があっても原因の究明は難しい。昨今は広い意味で失敗を捉えた事例の紹介[1]も始まり、開発に関連する投資効果[2]や、ステークホルダーの意識変化[3]に関する問題意識も高まりつつあるが、効果的な解決策が現れるには至っていない。

2.2. フレームワークによる解決

本稿で提案するのは、戦略コンサルティングの分野でフレームワークと呼ばれる考え方をシステム開発のプロセス改善に応用しようとするものである。戦略コンサルタントは、企業の経営戦略や事業戦略に対して助言を行う職業である。彼らが解決しなければならない問題は、システム開発と共通の特性を持っている。まず、それぞれの企業の経営環境やそこに登場するステークホルダーはそれぞれ全く異なっている。そして、何をもちて戦略の失敗・成功を決めるのかは容易ではなく、しかも失敗が認められにくい事情についても同じである。こうした状況の中で、それぞれの企業に有効なサービスを提供するために、使われているのがフレームワークと呼ばれる考え方である。

フレームワークは、企業の種類によらず経営や事業の分析を行うための視点を汎用的に用意したものである。具体例としては、会社の経営環境を分析する3Cや、企業の戦略を検討するためのSWOTなどが知られている。いずれも、極めて単純なものではあるが、マクロな視点で経営の状況を把握して、必要な意志決定の指針を示すことができるようになっている。

2.3. 失敗リスク評価のためのフレームワークの要件

著者らは、システム開発の失敗を捉えるためのフレームワークを用いて、プロジェクトの個別性に左右されることなくシステム改善のためのノウハウを共有してゆくことが可能になると考えた。システム開発の領域でフレームワークとして使える枠組みには、いくつかのものが知られている。例えば、ITガバナンスの観点からシステムの企画から運用までのプロセスを定めたCOBITの体系、プロジェクト管理に関するPMBOKの知識体系などがある。しかし、これらをシステム開発の失敗リスクの評価に利用するのは難しい。

システム開発の失敗リスクの分析のためのフレームワークは、以下の要件を満たす必要がある。

(1) 広い範囲のシステム開発に適用できる汎用性があること

システム開発プロジェクトの多様性に関わらず共通の視点を提供できる必要がある。

(2) 失敗要因を特定し対策を打つのに利用できること

システム開発における複数の失敗要因を明示し、対策を考える助けにならなければならない。

(3) 様々なステークホルダーの間で理解・共有できること

経営者を含め全員が理解・共有できる単純で明快な枠組みでなければ、意志決定につながらない。

上述したCOBITやPMBOKは、いずれも(1)は満たし、(2)の用途に使うことも可能ではあるが、複雑になりすぎるため、経営者も含めた多くの人の理解できるレベルにはなく(3)が満たせない。

3. GAPS²分析フレームワーク

本稿では、上述した要件を満たすことのできる、システム開発の失敗リスク評価のためのフレームワークとしてGAPS²(gaps of GAPS)分析の枠組みを提案する。

一口にシステム開発と言っても様々なものがある。各部署の課題を解決するために既存のパッケージを導入するものや、社内のインフラとして世の中の標準的なシステムを導入するだけのものもある。GAPS²分析フレームワークが対象とするシステムは、企業の経営的な目的達成のために構築され、企業内の複数部門の業務に関わる複雑なシステムである。こうしたシステムは、既存のパッケージを導入するだけでは済まず、フルスクラッチ(ゼロからの作り込み)による開発が行われることが多い。このため構築には大きなリスクがあり、また、企業の根幹のビジネスに直結するため失敗の影響も大きい。

著者らは[4]の中で、こうしたシステム開発の失敗要因について、ゴール(目的: Goal)、アクティビテ

イ(業務：Activity)、プロセス(工程：Process)、スキル(技術：Skill)のギャップという視点で分類する GAPS モデルの考え方を示した。GAPS² 分析は、これらの視点に従ってプロジェクトの問題事象を図 1 に示す形で整理・検討するためのフレームワークである。

これら 4 つのギャップは、システム開発に関わる 4 種類の主要なステークホルダーの視点からの失敗に対応する。すなわち、ユーザ企業における、(1)開発の最終意志決定者である経営者、(2)システムのユーザとなる業務部門、(3)システム開発のプロジェクト管理に責任を持つ情報システム部門(以下、情シス)、そして(4)開発の実作業を行う開発ベンダーである。各ギャップは、以下に示すように、各ステークホルダーが担っている責務が果たせない状態である。

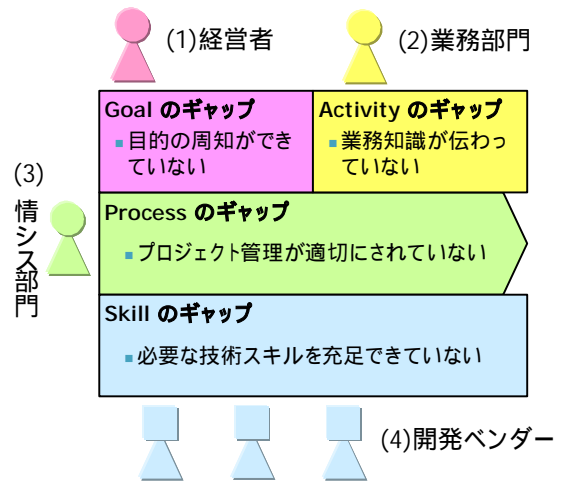


図1 GAPS²分析フレームワーク

- ・ゴールのギャップは、構築したシステムが順調に稼働しているにもかかわらず経営者の目的を達成しない場合で、主な原因は経営的な目的が業務部門や情シス部門に周知できなかったところにある。
- ・アクティビティのギャップは、システムが計画通りに構築できたにもかかわらず業務部門の支援ができない場合で、主な原因は業務内容を情シス部門や開発ベンダーに伝えられなかったところにある。
- ・プロセスのギャップは、システムは構築されるが、情シス部門に与えられた予算と納期を大幅に越えてしまう場合で、主な原因はプロジェクト管理が開発状況の変化に追従できなかったところにある。
- ・スキルのギャップは、開発ベンダーが要件どおりにシステムを構築することができない場合で、主な原因はシステム構築に必要なスキルを持つ要員を揃えられなかったところにある。

4. GAPS²分析の適用

4.1. トラブル事例への適用

図2は、システム開発で起きたトラブルの概要をGAPS²分析のフレームワークで整理したものである。

事例1は、インターネットでのサービスサイトを構築した事例である。この事例では、必要な機能を実現するために特殊なプロトコルの実装が必要だった。ところが、開発ベンダーにプロトコル実装の経験を持つ要員がいなかったため、その部分がネックになって遅延が発生し、その遅延をカバーするために主要部の技術リーダーを投入した結果、遅延が全体に波及してしまったものである。

事例2は、基幹システムをフルスクラッチで構築した事例である。この事例では、ユーザ部門の業務分析を上流工程のコンサルタントが行い、その結果に基づいてシステム構築を行うものだった。ところが、業務分析に漏れがあったため、その部分を開発ベンダーが行うことになり、前工程への手戻りが発生した。さらに分析の結果、要件のスコップ(範囲)が膨らんでしまい、当初の想定工数を大幅に上回るようになってしまったものである。

事例3は、既存と新規を含む複数のシステム連携に必要な連携機能を構築した事例である。この事例では、コスト削減のために、連携の大変な旧システムは廃止し、連携の中心を新システムにするという方針であった。しかし、旧システム廃止の方針を知らない現場担当者からの要望に対応して、要件定義を進めたために必要な連携が増え、要件のスコップが大きく拡大してしまった。さらに、このことに気づかず、拡大した要件で関連部門との調整を進めた結果、後戻りできなくなってしまったものである。

4.2. 適用結果についての考察

いずれの事例も、結果的には開発プロジェクトを計画どおりに進めることができなくなってしまっている。このため、プロジェクト管理上の問題と捉えることもできる。しかし、根本的な要因はそれぞれ別のところであり、有効な対策を検討するには要因を構造的に理解する必要がある。

事例1は、必要とされる技術スキルを持つ要員を配置できずにシステム構築ができなかったことが原因であり、スキルのギャップである。早期に気づけば全体に影響のない体制構築ができた可能性がある。

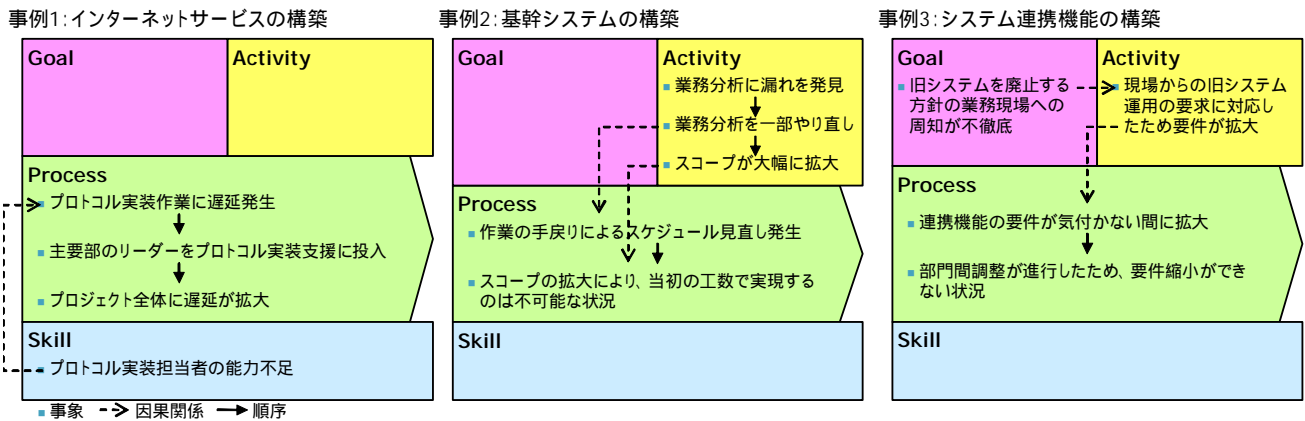


図2 GAPS²分析の適用

事例2は、業務分析に漏れがあったため、そのままでは業務の支援が不十分なものになるので、アクティビティのギャップである。これは本来やるべき作業が行われていなかったことを意味し、計画の見直しは避けられない。事例3は、経営上の目的をユーザ部門や開発ベンダーに対して徹底できていなかったことが真の原因であり、ゴールのギャップである。同時に要件定義の拡大という変化を捉えられなかったというプロセスのギャップも生じている。当面の要件拡大への対応だけでなく、これらのゴールとプロセスのギャップに早めに手を打たなければ他の問題を引き起こす可能性がある。

このように、GAPS²分析のフレームワークは、システム開発のトラブルについて、要因の構造をステークホルダーに示し、必要なアクションを促すための手段として用いることができる。このことから、著者らは、GAPS²分析フレームワークは以下の要件を満たすものと考えている。

- (1)十分な汎用性があり、様々なシステム開発に適用できる。
- (2)異なる種類の要因を特定して、対策の検討に用いることができる。
- (3)経営者やユーザ部門に理解できる、単純かつ明快な指針を示せる。

GAPS²分析の活用はまだ試行錯誤の段階である。今回の適用例は、トラブルが判明した後で行ったものだが、リアルタイムでの状況把握に利用できるようになれば、リスク回避に役立つことが期待できる。

5. まとめ：重要なのは情シス部門のガバナンス力強化

本稿では、システム開発の失敗を回避するためには、定性的なリスク分析を可能にするフレームワークが必要であることを述べ、そのひとつとしてGAPS²分析フレームワークを提案した。さらに、トラブル事例への適用を行い、失敗要因の分析に活用できることを示した。筆者らは、GAPS²分析以外にも、様々なリスク評価のフレームワークを用意していくことで、リスク低減が可能になると考えている。

しかし、システム開発のプロセスを改善するにはフレームワークだけでは十分ではない。フレームワークによる状況分析の結果を用いて、PDCAのサイクルを回していく主体が必要である。著者らは、この役割を果たすのは、ユーザ企業の情シス部門であるべきだと考えている。問題の根本原因が、ステークホルダー間の知識のギャップにあるのだとすると、外部の開発ベンダーにそれを埋めることはできないからである。ユーザ企業の情シス部門が、開発ベンダーへの依存から脱却し、自ら主体的にシステム開発をリードできるガバナンス力をつけることこそが、システム開発の失敗リスクを低減させる最も重要な鍵になるはずである。

参考文献

- [1] 不条理なコンピュータ研究会, 「IT失敗学」の研究, 日経BP社, 2006.
- [2] 栗山敏, “情報システム投資の有効性評価に対する合意形成手法適用の提案”, 第1回情報システム学会研究発表大会, 2005, A-02.
- [3] 石井信明, “プロジェクト過程のステークホルダー意識変化とマネジメント”, 第1回情報システム学会研究発表大会, 2005, A-05.
- [4] ウルシステムズ, 間違いだらけのシステム開発, 翔泳社, 2006.