

Title	業務ITシステムの受託開発における事後不満解消のメカニズム
Author(s)	山口, 啓介
Citation	年次学術大会講演要旨集, 37: 931-934
Issue Date	2022-10-29
Type	Conference Paper
Text version	publisher
URL	<a href="http://hdl.handle.net/10119/18521">http://hdl.handle.net/10119/18521</a>
Rights	本著作物は研究・イノベーション学会の許可のもとに掲載するものです。This material is posted here with permission of the Japan Society for Research Policy and Innovation Management.
Description	一般講演要旨

## 2 F 0 3

# 業務 I Tシステムの受託開発における事後不満解消のメカニズム

○山口啓介（スーパーエンジニア株式会社）

### 1. はじめに

業務 I Tシステムを受託開発する際の重要なポイントは、顧客の業務内容を正確に把握した上で顧客の要望を取り込み、アナログ業務をデジタル化することにある。顧客プレーヤーである経営者、管理者、実務担当者などから顧客ニーズをヒアリングし、業務 I Tシステムとしてデジタル化する。

一方、システム開発分野では古くか存在するウォーターフォール手法や最近主流のアジャイル手法など、いくつもの開発手法が存在する。これらの開発手法にはセオリーや確認点・合意点があり、準拠して積み重ねていけば開発を成功へ導いてくれると説いている。

しかし、顧客ニーズを取り込み開発手法のセオリー通りに開発しても、納品後に顧客から不満が発生する事例が自社で確認されている。これらの事例は、開発中は適切な顧客プレーヤーからヒアリングをしていた。また適切な開発手法を選択し開発プロセスごとの確認書類の合意も得ていた。それにも関わらず納品後に事後不満が発生した。なぜ事後不満が発生するのか。その発生原因はどこにありどのような発生メカニズムなのか。発生メカニズムから解消するマネジメントはどのように考察できるのか。本報告では、これらの問いをリサーチクエストに設定し、事後不満解消のマネジメントを考察する準備作業として事後不満発生メカニズムの解明を試み、自社事例分析をもとにした仮説提案を行う。

### 2. 先行研究と理論的背景

#### 2-1 業務 I Tシステム関連の先行研究の調査

この分野の先行研究では技術論ばかりが多い。I T技術関連書も同様で、開発手法のセオリー[1]、開発の鉄則[2]、各プロセスで顧客と合意すべきポイントの解説[3]に終始している。あるいは過去の失敗例をもとに顧客と論争にならないようにする契約書の作り方で是正しようとする書籍もある[4]。

本報告は、技術論に従って業務 I Tシステムを構築したにも関わらず事後不満が発生しているという点から、先行研究では存在していない調査となっている。

一方で少数ながら技術論に終始しない先行研究もある。首藤の先行研究[5]は大変興味深く、経営学や組織論との関係性を述べている点がユニークである。また首藤の博士論文[6]も参考にしたが、公表されていないため仮説提案の補足として活用した。

#### 2-2 経営学・組織論との関連性

業務 I Tシステムは業務との関連性が強く、業務は組織との関連性が強いいため、本来業務 I Tシステムは経営学・組織論と深い繋がりがある。本報告で原因に辿り着いたいくつかは経営学・組織論[7]との関連性が含まれている。技術論だけでは解決できない背景が事後不満の原因や発生メカニズムにあると考える。

### 3. 研究手法と事例選択

#### 3-1 研究手法について

事例研究の題材には、筆者が経営するスーパーエンジニア株式会社（以下 S E社と表記）の事例を選択し、単一企業の複数事例研究とした。

まず、S E社事例をもとに事後不満の種類洗い出しから始めた。この事前準備として、顧客の不満が開発中に発生していた事例、そもそも開発手法の実施に問題があった事例は除外した。これらの事例は、開発中に解消しきれなかった不満が事後不満に繋がった、あるいは開発プロセスのどこかで適切ではないやり方が行われたなど説明のつくものが多いため、今回の研究対象外とした。

次に、事後不満が発生した開発プロジェクトに関連するプレーヤーの行為を、開発プロセスを遡って網羅的に整理を行った。一般に業務 I Tシステムの受託開発は、要件定義→概要設計→詳細設計→システムテスト→テスト運用→納品、といった流れになる。これら開発プロセスの流れをテスト運用から逆

に辿り、その各プロセスで開発に関わった様々なプレーヤーの発言との関連性を整理した。様々なプレーヤーには顧客と開発エンジニアの双方を含む。

最後にこれらから原因の抽出と論理を分析した上で、発生メカニズムを明らかにした。

### 3-2 事例について

本研究では、SE社の過去事例を対象にしている。事実収集の方法としては、開発当時の議事録メモ・設計書などのドキュメント資料をもとにした。また開発当時の顧客・開発エンジニアからのインタビューも含めた。ただし筆者自身の経験や記憶は客観的視点を重視すべく補足として活用した。

SE社は30年以上業務ITシステムの受託開発を専門に行っているシステム開発企業である。ターゲットは製造業・販売業が多く、顧客企業と直接契約により顧客ニーズを取り入れた業務ITシステムをフルスクラッチで開発してきた。特に企業全体を統括する基幹システムを数多く開発している。

研究手法で述べたとおりSE社の複数事例の中から「顧客の不満が開発中に発生していた事例、そもそも開発手法の実施に問題があった事例」は除外している。結果として「顧客ニーズを取り込み開発手法のセオリー通りに開発したにも関わらず事後不満が発生した事例」が研究対象となっている。本報告のリサーチクエスチョンとの適合点となる事例である。

## 4. 事例紹介と分析結果

### 4-1 真のヒアリング相手が異なっていた

顧客へのヒアリングを行う際には、適切なヒアリング相手の選択が重要なのは言うまでもない。例えば在庫管理をシステム化するには、資材調達・生産・販売の各部署からのヒアリングが対象となる。また、管理決定権のある部署長だけではなく、末端のオペレーション担当者からの情報も重要である。特に業務ITシステムが導入された結果、業務変更・業務フロー変更・管理仕様変更など、変更に関わる相手には注意する。変化が起きない業務に携わっているプレーヤーはヒアリング相手として除外される。

しかし業務フローが以前と同じでも「指示の取消から始まる業務の再試行」が悪影響を与えることがある。個別業務に変更はないが、この再試行が悪影響を与えるのは末端のオペレーション担当者になることが多い。再試行を考慮していないため「使い勝手が悪い」という事後不満が発生する。しかしオペレーション担当者がヒアリング相手に選ばれることはない。真のヒアリング相手は異なっていたのである。

事例：承認ワークフロー管理を導入する際、最も業務変化があるターゲットは管理者である。末端のオペレーション担当は業務に変更がないためヒアリング相手としては除外される。

しかし運用してみると管理者がラクをしようとした結果、「ボタン1つで安易な承認」から始まる取消作業が頻発し、業務の再試行が多発した。このシワ寄せは末端のオペレーション担当者であり「使い勝手が悪い」と事後不満が発生した。

### 4-2 形式知化が逆機能となる

業務熟練者の頭に入っている暗黙知を誰もがわかる形式知に変換することは経営学的にも重要とされている。たとえば生産部署においては、熟練した技術者の暗黙知を形式知化することで、経験の浅い技術者が熟練工に近い成果を出せるようになる。業務ITシステムを開発する上でも「暗黙知の形式知化」は重要で、ルール化されていない業務フローを形式知化することがシステム化の第一歩となる。

原材料を混ぜ合わせて作られる製品の製造プロセスを形式知化するには、混ぜ合わせるレシピと投入タイミングが必要となり、製品量が多くなると使用する原材料にズレが発生するので補正を必要とする。これらは熟練者が勘で行っている場合が多く、この暗黙知を形式知化しなければ生産管理システムは開発できない。また生産管理から繋がる在庫管理の正確性も同様である。

一方、業務ITシステムでの形式知化は仕様書を得て「システム運用時に顕在化」する。仕様書段階で承認されてもこれは技術的な正解であり、実は顕在化してはいない。また熟練者が気づいていない「さじ加減」が隠れていることがある。製造時に原材料の「さじ加減」投入が実施され、計算と合わない原材料の消費が発生する。製品の中には半製品と呼ばれる原材料扱いになるものもあり、これが別製品の原材料として使われる「さじ加減」が加わり、製品在庫数も計算と合わなくなる。

システム運用時に初めて顕在化した形式知は、隠れた「さじ加減」によって在庫計算の正確さを失う。実棚数と在庫計算の不一致が形式知化の逆機能として働いてしまう。

事例：塗料メーカーの生産管理では原材料成分の配分構成が記載されている「配合表」が必要となる。成分の関係で季節によって若干配合が異なり、また生産量によっても若干変わる。しかし条件分岐と計算式で実現可能であるため設計作業は進めやすく承認も得やすい。デジタル化によって顕在化した形式

知は「実棚と在庫数の不一致」という逆機能を生み出してしまい、「在庫計算が合わない」という事後不満が発生した。

#### 4-3 ITシステムと非ITシステムとの関係性の発見

ITシステムは形式知の集合である。形式知化できないものはITシステムにはならない。対して、暗黙知は非ITシステムである。前述した「さじ加減」は暗黙知のままであったため、これを「さじ加減で使った原材料は在庫調整入力をする」という機能を追加した。しかし「在庫計算が合わない」という事後不満は多少減ったが「職務外の時間が増えた」という新たな事後不満が発生した。

原材料数を合わせるという目的のために「調整」というプロセスを作り出して形式知化したが、「さじ加減」は非ITシステムからITシステムへ「代替」するものではなく「補完」するものであった。

暗黙知を形式知化することは「代替」となる。しかし、暗黙知の全てが「代替」できるわけではなく、その場合は「補完」という形で関与する。補完するにはITシステムと非ITシステムの橋渡しをする「非接合」が存在する。非接合は必要なものであり、無理な代替は非接合不足を発生させ業務全体を歪めることになる。熟練者の「さじ加減」は「代替」せずに「補完」という関係性が必要になることを発見した。

事例：前述の「形式知化が逆機能となる」にあらたに「原材料在庫調整入力」という機能を追加したが、これを使用できるのは熟練者のみであり熟練者に不要な操作をさせることになった。「さじ加減」を形式知化する手段として顧客と合意した結果であったが、あらたな事後不満が発生してしまった。

#### 4-4 影響システムへの悪影響

集計値による報告書・分析資料は、基になる情報が詳細でなければ分析ができない。業務ITシステムで営業報告資料機能を開発する場合、報告のための情報をわざわざ入力させる仕様は認められない。そのため、必須となる機能から集計・分析して報告書を作る機能が重要となる。当然、詳細な情報を得るには詳細に項目入力することが必要となる。ここで便利機能の1つとして簡易的な情報入力画面を用意し、詳細入力・簡易入力のどちらからも報告書が作成できると、営業担当の負荷が軽減される。簡易入力はあくまでも非定型サブ機能として存在する。

しかし残念なことに、本来詳細入力すべきことも簡易入力優先となってしまうと、報告としては問題ないが細かい分析が進まない。大ざっぱな分析によって管理者の判断材料が少なくなり、現場を動かす影響システムへ悪影響が発生する。

事例：営業担当者から正確な情報を報告させるために見積書に付加情報を追加して、全ての案件に対して見積書入力を義務づける仕様とした。しかし一部営業担当者から簡易版も必要との意見があり経営者・管理者も承諾の上で開発した。運用後は簡易版ばかりが使用され、特に管理者から「詳細情報が取得できない」と事後不満が発生した。

#### 4-5 問題の原因

本報告の研究範囲では3つの原因が分かった。まず「組織との関連性」が挙げられる。真のヒアリングの相手が異なっていたのは、ヒアリング相手を「対象者」として認識してはいたが「組織との関連性」に着目しなかったところに原因があった。

2つ目は暗黙知から形式知への変換において「逆機能の発生」を予測できず「暗黙知と形式知の非結合不足」を考えていなかった。形式知化が逆機能を発生することはあり得る。さらに形式知化は暗黙知からの「代替」ではなく「補完」が重要であり、非結合の存在に気づかずにいたことが原因である。

3つ目は非定型サブ機能が定型化することで影響システムにまで悪影響を与えることを想定できなかった点。経営者目線では承認できても、直属の管理者目線で承認できない結果となっていた。影響システムに悪影響ということは部下へのインセンティブ提供、行動管理まで悪影響が広がる可能性があるため見逃せない原因である。

### 5. 結論・貢献・課題

実務では原因分析より課題解決を優先することが多く、また1つの案件単位のみで検討していたが、本研究によって多方面を分析することでいままで見えていなかった原因を知ることができた。事後不満の原因は経営学・組織論に関わる部分も多々あり、技術論だけでは解決できないと考えられる。

また本報告の結論として、事後不満発生メカニズムを2つ発見した。1つ目は、開発手法に準拠すればするほど「点」で見えてしまうこと。「点」の例としてヒアリングターゲットとなる担当者が挙げられる。事例では真のヒアリング相手は「点」として存在していなかった。視野狭窄とも言い換えられるが、視野を広げても該当する「点」として認識できないと不満発生メカニズムに陥る。「点」に重要な該当

者を入れられないというメカニズムが存在する。

2つ目は「IT万能信仰」。業務ITシステムを構築するという目的に、なんでもIT化できるという穿った考えを信仰してしまうというメカニズム。暗黙知は全て形式知化できるという「万能信仰」も含まれる。代替ではなく補完や非結合の存在というITが万能ではないという事象を何もかもをIT化できるという方向へ進めてしまう。業務ITシステムに関与するプレーヤーが経営学・組織論を無関係だと決めつけ、技術論先行としてしまうことも「IT万能信仰」へ結びついている。

システム開発中に、これらメカニズムに該当すると事後不満に繋がると考察した。したがって、この考察から事後不満解消のマネジメントへ繋がれば、事後不満減少へと貢献できる。

本報告をさらに進めるため、①経営学・組織論との関係性の精微化、②調査範囲を他社事例に広げる、③解消のマネジメントの確立、以上3点を今後の課題としたい。

## 参考文献

- [1] 赤俊哉, システム設計のセオリー, リックテレコム, (2016)。
- [2] 田村昇平, システム発注から導入までを成功させる 90 の鉄則, 技術評論社, (2017)。
- [3] 赤俊哉, だまし絵を描かないための要検定のセオリー, リックテレコム, (2018)。
- [4] 細川義洋, なぜ、システム開発は必ずモメるのか?, 日本実業出版社, (2013)。
- [5] 首藤聡一郎「形式化推進装置としてのIT」『一橋研究』一橋大学大学院一橋研究編集委員会, 第30巻第2号, 2005a, pp.43-pp.56.
- [6] 首藤聡一郎「電子系情報システムと人間系情報システムのかかわり：電子情報機器と企業組織の関係」一橋大学大学院商学研究科・修士論文, 2003.
- [7] 伊丹敬之, 経営戦略の論理 第4版, 日本経済新聞出版社, (2012)