Revision 0.40 2024/06/18

－　コンピュータシステム事故はトップマネジメントの責任！　－

Mignon M. Mazda （松田　實）

０．概要

ここ数年、私企業や組織体の運用を支えているコンピュータシステムが機能不全に陥り、製品やサービスの供給が停止するといった事故が多発しているのは何故なのか、品質管理の側面から分析し、トップマネジメントが実行せねばならない改善策の提言を試みる。

１．コンピュータシステムが動かないとどうなるか

最初に、私企業や組織体が製品やサービスを消費者へ供給するにあたり、それらの生産や提供に直接関与する人々からなる集合を「現業部門」と呼称する。ここでは、人間が製品やサービスの供給に直接関与しないとならず、現在の技術レベルではコンピュータシステムがそれに取って代わることはまず無理である。

次に、製品やサービスの供給を支えるために必要不可欠ではあるが、コンピュータによる自動処理が可能な範囲を一定の計画に基づいて実務に落とし込み、人間の手を直接必要としないように備える人々の集合を「システム開発部門」と呼称する。

もちろん、「システム開発部門」においても人間の手を直接的に介在させて、実務現場で使用するコンピュータシステムを開発し、実際の運用に移せねばならない工程（プロセス）は厳然として存在する。これは、私企業や組織体の業務遂行に必要なコンピュータシステムを開発し、目標を達成するのに必要じゅうぶんな機能と性能を有するハードウェアとソフトウェアを組み合わせる必要があるからである。

当然、これらのハードウェアとソフトウェアの両方がうまく嵌合（かんごう）しないと、私企業や組織体による製品やサービスの供給ができず、少なくとも社会の一部が機能不全に陥る。直近の身近な事例を挙げれば、江崎グリコ社におけるプリンなどの供給及び販売が長期間にわたって不可能になっていることからも明確に分かる。（註記１及び２）

２．コンピュータシステム開発における現業部門業務を文書化する重要性

コンピュータシステムを開発、導入する私企業や組織体が最初にやらねばならぬことは、従来人間の手で直接行ってきた業務内容を極力詳細精密に文書化することである。

この作業で絶対に必要なのは、私企業や組織体がいったい何をコンピュータシステムによる自動化の目標並びに目的としているかを明確にすることである。

この文書化作業の結果、具体的なシステム開発の目標（ターゲット）が明確になるから、コンピュータシステム化する業務内容に疎漏（そろう）がなくなる。逆に言うなら、ここに疎漏を含む間違いや混乱があれば、私企業や組織体のコンピュータシステムが機能不全に陥る要因を作り込むことになり、コンピュータシステム事故の発生を阻止できない。

よって「システム開発部門」は、すべての「現業部門」の業務内容を理解し、同時に文書化できるシステムエンジニアの集合体でないとその役割（責務）を果たすことができない。

すなわち、この部門で最も大切となる要件は、直接目に見えない業務を含めた現場実務を疎漏なく文書化できる能力を保有する人材を「適切な規模」で配置すべきことである。

ほぼすべての私企業や組織体の人員配置は、人事担当部門が管轄する業務に含まれるが、それを最終的に決済するのはあくまでもトップマネジメントの自由と責任に帰する行為であり、「システム開発部門」にどれだけの経営資源を費やすかで、コンピュータシステムが保有する「品質」そのものが大きく左右され、場合によっては致命的な事故に至ることを、トップマネジメントはきちんと理解し、遅滞と疎漏のない経営執務を実施する必要がある。

－　O　－

この後に続く第３項から第６項では、コンピュータシステム開発の基本的大原則を羅列的に記述しているだけですから、既にこの業種業界に関する知識見識をお持ちの方は、本文章の第４ページにある第７項から後の項目まで飛んで読み進めて下さっても結構です。

－　O　－

３．コンピュータシステムの設計（デザイン）と製造の原則

製品やサービスの供給を円滑順調に実施するために必要なコンピュータシステムの要件が文書化されたら、次に必要なことはそれを段階的に詳細な部品（パーツ）毎に分割して、より詳細な設計仕様書に落とし込む作業になる。

私企業や組織体の開発目標となるコンピュータシステムの規模によるが、ソフトウェア開発をする場合、一般的に「基本設計仕様書」、「中間部品設計仕様書」、「詳細部品設計仕様書」、そして「コーディング（プログラミング）設計仕様書」が必要になる。

これらの階層構造（以下、ヒエラルキーと表現する）を有する設計仕様書に基づき、実現すべきコンピュータシステムのソフトウェアが設計製造される。当然、この段階においても極力記述内容に疎漏がないことが最も重要である。

４．コンピュータシステムソフトウェアの試験と検査の原則

これまで述べてきた工程（プロセス）の産物として、ヒエラルキー的に下から挙げると、「コーディング（プログラミング）設計仕様書」に基づき、適切な各種プログラミング言語で書かれた「第１次ソース」部品、「詳細設計仕様書」の要求事項に従う「第２次パーツ」部品、「中間部品設計仕様書」の要求事項を満たす「第３次パーツ」部品、およびそれらを総合的に組み上げた「基本設計仕様書」どおりの機能と性能を有する「完成ソフトウェア」が設計並びに製造工程から出力される。

もちろんのことであるが、すべての設計製造工程では、各々のソフトウェア部品や完成品が必要な「設計仕様」を満足していることを確認するための試験と検査が必要である。

これらをまた下から順番に並べるなら、「単体テスト」、「組み合わせテスト」、「システムテスト」、そして実際のハードウェアに組み込んだ「仮運用（オフラインテスト）」となる。

これらの試験や検査を経て、ソフトウェアとハードウェア全体が必要じゅうぶんな機能と性能を保有することが確実であると確認できたら、ようやく私企業や組織体の実務現場での正式運用（本番運用）が開始できることになる。

５．コンピュータシステム事故発生の理論的メカニズム

ここまで述べてきたシステム開発プロセスを経たコンピュータシステムが、完全無欠なものであれば、「事故」は起きないはずだが、不完全な存在である人間が介在する作業である以上、事故要因をゼロにすることは統計学的に不可能である。

どんなに単純な作業を実施する場合でも、人間は約５，０００回に１回の間違いを起こすことが統計学的研究結果として得られている。（註記３）

これが何を意味するかというと、これまで述べてきた各種の設計仕様書や試験・検査の確認仕様書（チェックシート）には疎漏事項が含まれており、それを起因とする事故はいつか必ず発生し、回避することは不可能だと言い切っても決して過言ではない。

それ故、重要な問題とすべきは、システム事故発生要因の完全除去ではなく、事故発生を必須前提条件として考え、事故に起因する悪影響を最小限度に押さえ込むための代替手段を確保することである。

基礎的手法だと、システムの二重化によるオンライン処理ダウンの防止措置や、新規導入したシステムに問題が発生した場合、遅滞なく旧来のシステムを稼働させるロールバック措置などが代替手段になるケースが挙げられる。（註記４）

６．コンピュータシステム事故発生時の安心安全確保

コンピュータシステムの品質管理では、人間の誤操作あるいは機械の誤作動があっても、直接的もしくは間接的被害が発生しないよう、フールプルーフ機能やフェイルセーフ機能を備えておくことがシステム開発部門における大原則の一つになっている。

簡単な例を挙げるなら、金融機関のＡＴＭ（現金自動預け払い機）において、人間による入力に誤りがあった場合、必ずエラーメッセージを報告して再入力を求めるなり、すべての手順を最初からやり直すよう求めることが製品要求事項に含まれている。

また、ＡＴＭは電気で動く機器であるから、取り扱う人間が感電する事故や、ＡＴＭ自体からの発煙発火事故を起こさず、直ちに全電源を遮断する機能を備えておかねばならない。

特に後者はＰＬ法など関係法令によって厳しく規定されており、ＡＴＭ機器製造組織体に義務づけられている。

－　O　－

この文に続く各項目の記述には、本文章のメインテーマに沿った内容を書いてありますので、第３項から第６項を読み飛ばした方を含め、すべての読者の方々は是非御一読下さるようにお願いします。

－　O　－

７．コンピュータシステム事故発生の技術的メカニズムとその基本的特徴

この項では、コンピュータシステム事故発生の技術的なメカニズムとその基本的な特徴について考察する。

現在とこれから先の未来に渡り、製品やサービスの供給を行う私企業や組織体にとって、コンピュータシステムはとても便利な「道具」だが、事前にプログラムされた「定型処理」（ルーチンワーク）を実行する能力を有するだけで、「現業部門」の人間のみが対応可能な、いわゆる「飛び込み業務」と呼ばれる「例外処理」に対しては、まったく無力であるという技術的制約を持つ。

つまり、コンピュータシステム開発における業務の文書化のプロセスにおいては、必ずと言っても構わない程度含まれる疎漏事項によって、コンピュータシステム自体が保有するソフトウェアプログラムなどに潜む根本的かつ致命的欠陥（バグ）による事故を、ほぼ必ず発生させるものであることを認識すべき必要がある。

また、私企業や組織体がコンピュータシステムの開発に失敗した場合、それによって発生する事故に伴う製品やサービスの供給が全くできない完全機能不全（デッドロック）状態に陥る危険性（リスク）を正確に把握し、必要に応じた危機管理マネジメントを遅滞なく発動せねばならない。

身近な事例を挙げるなら、銀行など金融機関のオンライン業務がコンピュータシステム事故によって金融取引処理を完全に停止させた場合、それが回復するまでの間、社会全体に与える悪影響を想像していただければ理解が早いと思われる。

この完全機能不全（デッドロック）状態に陥った私企業や組織体のトップマネジメントは、遅滞なくフェイルセーフ機能を発動し、旧来のシステムを稼働（ロールバック）させるなど「原状回復措置」を取らねばならない。

また、事故の原因と場合にもよるが、現行の開発計画をいさぎよく断念し、コンピュータシステム開発全体を全くのゼロベースから再構築（スクラップアンドビルド）する経営判断をせねば、最悪の事態から逃れられないことを知るべきである。

ここで肝心なことは、「現業部門」の業務を無理にコンピュータシステムの技術的仕様に合致させる努力を払うのではなく、あくまでも「現業部門」において従来から実施していた「現場実務」の一部もしくは全部をコンピュータシステムに代替させて自動処理することを最も優先させる措置を講じねばならないという点に尽きる。

なぜなら、この項の最初で述べたとおり、現在のコンピュータシステムは「例外処理」を全く想定できず、またその処理を組み込むことができない技術的制約を持つからである。

また、この文章の中ほどで記述したとおり、コンピュータシステム事故が統計学的確率で発生することを避ける手段は存在しないが、その発生確率を学問的理論値に近い近似値に落とし込むことは可能である。

もちろん、そのためにはコンピュータシステムの開発と運用に対する継続的な品質向上施策を施すことが必要不可欠な要素（ファクター）であることをきちんと理解し、実施可能な範囲での品質向上施策をもれなく実践に移さなければならない。

８．コンピュータシステム事故発生の社会的メカニズム

この項では、製品やサービスの供給を行う私企業や組織体がコンピュータシステム事故発生要因除去策を講じているにもかかわらず、事故が後を絶たない社会的なメカニズムについて考察する。

第一に挙げるべき要因は、私企業や組織体が「システム開発部門」を自組織内に保有する重要性をあまりにも軽視していることである。

私企業や組織体のトップマネジメントが、経営体制の合理化や要員の省力化を優先するあまり、「システム開発部門」をほぼ全部下請け企業などへ丸投げし、外注化している事例は決して珍しくない。

当然、このような体制でコンピュータシステムを開発する場合、自組織の製品やサービスの供給を行う実務現場の業務を極力詳細精密に網羅（カバー）する文書に疎漏事項が含まれる危険性が非常に高くなる。

すなわち、システムの基本設計部分に欠落が含まれるのだから、システム開発の初期段階に基本的かつ致命的な欠陥（バグ）を自らの手で作り込むことになる。これでは目標に届く「品質」を保有するコンピュータシステムを作り出すことがほぼ確実に不可能である。

第二に挙げるべき要因として、システム開発の外注先では、あくまでも発注側が提示した契約文書（「発注仕様書」）に書かれている要求事項をすべて網羅するコンピュータシステムを納入すればそれで契約完了となり、同時に責任免除となっている状況がある。

この場合、欠陥（バグ）を作り込んだのはすべて発注側の責任であるから、万が一事故が発生したとしても、外注（アウトソーシング）側においては根本原因の究明に困難を極めるとともに、再発防止策を策定し実施する責任と権限を直接持たないという問題がある。

第三に挙げるべき要因は、製品やサービスの供給を行う私企業や組織体の側が、多重外注の制限を付けないシステム開発契約を締結してしまうことにある。

極端な話になるが、たとえば発注元が日本国内のシステム開発メーカと外注契約を結んだつもりでいても、実際にプログラミング作業を行うヒエラルキー末端に位置する組織体もしくは個人が世界中のどこにいても問題はない。

この場合、下請け先に制限がないので発注側が支払う金額から無制限に「金銭的中抜き」が可能であり、見積もり予算の数十分の一もしくは数百分の一のコストでプログラミング作業を強いられている実態がある。

かなり極端な事例だが、「システム開発部門」のヒエラルキー最下層にいてコーディング（プログラミング）作業を請け負う「ＩＴ土方（どかた）」という代名詞で表現される人々にとっては、実際の仕事量に対する利益率があまりにも低くコストパフォーマンスが悪いので、発注元から各種の文書で提供された個人情報等を第三者へ流出させて経済的利益を得る事件が後を絶たない。この情報漏洩事件は世界中を見渡しても決して珍しくない。

こうなると、「システム開発部門」の制御がまったく効かない領域の現象が発生するわけであり、多重外注の仕組みには致命的欠陥が内包されていると断言できる。

情報漏洩事件の発生を防ぐ手段は限られるが、多重外注の仕組み自体を廃止するか、または制御可能な範囲で制限させることによって、情報漏洩事件の予防策を施す余地はあると言えるだろう。

９．コンピュータシステム事故を未然に防止する根本対策

この文章をここまで読んでいただいた方々にはもうお分かりいただけたと思うが、事故の発生をより確実に未然防止するために、製品やサービスの供給を行う私企業や組織体が、コンピュータシステムの「品質」を保持するため「システム開発部門」を丸ごと外注任せにせず、可能な限り私企業や組織体の内部部門へ戻すこと、並びにコンピュータ業界における「悪習」であるところの、多重外注の仕組みに発注側が一定の制限を設けることの二項目が必要不可欠になる。

これを実現するには、私企業や組織体のトップマネジメントによる適切な経営判断及び人的投資、ならびに設備投資を要するのが明白である。よって、可能な限り早急にこの対策を講じなければ、類似のコンピュータシステム事故や情報漏洩事件が、業種や業態を問わず、未来永劫繰り返し発生することになる。

現代では、コンピュータシステムは社会や生活の基盤（インフラ）を支える重要な「道具」だが、その実現について必要不可欠な能力を持つ人材の確保と、製品やサービスを供給する私企業や組織体の体制そのものの立て直しが早急かつ最重要課題の一つに位置付けられる。

直近の事例を挙げるなら、国内大手自動車メーカによるエンジンなどの型式認定データ改ざん事件の元凶は、品質よりも開発日程（経営効率）を優先させた結果に他ならない。

冒頭に書いたとおり、例えば消費者が要求するプリンをいつでも食べるように、供給側の私企業や組織体が保有するコンピュータシステムの「品質」をより完全なものに整えるには、トップマネジメントによる経営施策の革命的刷新と、経営効率重視から製品やサービスの品質重視へ転換回帰する組織体制の抜本的な再構築（リフォーメーション）が必要だと結論付けても決して過言ではない。

10. 結言

繰り返しになるが、製品やサービスを供給する私企業や組織体のトップマネジメントは、コンピュータシステム事故発生を未然防止し、供給者が顧客からの「適切な信頼感」を得るための「時間との戦い」が既に発生していることをもれなく知る必要がある。

すなわち、一刻も早い再発防止のための対応措置の確実な履行が必要であることを強く意識せねば、コンピュータシステムの品質管理体制の再構築と事故発生要因の根絶対策は「絵に描いた餅」に終わる。こういう事態の発生は何としても押さえ込まねばならない。

そこで第一に、私企業や組織体のトップマネジメント、およびハイレベルマネジメントを担う人々においては、従来から何も問題や事故が起きなかったから現状維持で良いという、「不確実な心理的根拠」や「過去の成功体験例」に基づく安穏な経営状態を放置せぬことが、早急かつ特別に強く求められているという認識を持つべきである。

また第二に、現在運用中もしくは開発中のコンピュータシステムが保有する「品質」の持続的向上（スパイラルアップ）を妨げる経営施策をもれなく排除するとともに、可能な限り早急に事故要因の徹底的な排除に取り組む「重大な社会的責任」（レスポンシビリティ）から逃れることはできないと常日頃から意識し、必須事項となるすべての経営執務にあたるべきだとの二項目を提言して、この文章を閉じたい。

結論をあえて一言に集約させると、「火のない所に決して煙は立たない」のである。

（おしまい）

註記１.

江崎グリコ社の基幹システム障害については、大規模な基幹システム開発実績に乏しいコンサルタント企業（デロイト　トーマツコンサルティング）に、システム開発のすべてを丸投げ外注したという報道情報がある。

註記２.

江崎グリコ社の基幹システム開発にあたっては、基本システムのソフトウェア技術仕様が半分以上ブラックボックスである「ＳＡＰ　Ｓ／４ＨＡＮＡ」システムが採用されたため、納期が大幅に遅延したとともに開発費用も大きく膨らみ、結果的に同社の営業利益が縮小したという報道情報がある。

註記３.

人間が行う作業の信頼性に関する誤りの発生確率は、(株)日立製作所・中央研究所が公表している数値を採用した。

註記４.

新規導入システムに不具合があった場合、旧来のシステムにロールバック措置を講じて障害の回復を図るのが常識的手段（セオリー）の一つであるが、江崎グリコ社はこれを採用せず、またその理由を２０２４年６月１１日の時点で明らかにしていない。