

高度ICT人材育成に関する考察

峯 英一郎
Eiichirou Mine

1. はじめに

「高度ICT人材育成に関する研究会」報告書～我が国を支える高度ICT人材の自律的な育成メカニズムの構築に向けて（平成20年5月30日総務省）によると、ICTの適用範囲は拡大し続けており、産業・行政・社会の基幹システムとして活用されるとともに、国家の競争力を支える中核技術として、その期待はますます膨らんでいるとされています。しかし、その一方で、インド、中国、韓国等が、国家としてICT人材育成に注力することなどにより、我が国を激しく追い上げ、我が国におけるオフショア化が急速に拡大している状況を捉え、その危機感を募らせています。情報システム・ソフトウェアの構造変化（オープン化・モジュール化、SaaSの登場等）が進むことも踏まえて、高度ICT人材がますます必要とされている時代となっていることが指摘されています。

しかし、そういった背景があるにも関わらず、我が国の現状としては、

①産業構造：低収益、マーケットメカニズムが働きにくい構造

②人材の活躍の場（人材育成環境）：処遇面を中心としたICT人材の職業としての魅力が低下

③人材の育成の場（人材育成機会）：社会・経済・産業のニーズにあった教育（育成機会）が不十分

といった課題があり、高度ICT人材が不足する事態を招いていることが問題視されています。そのため必要な高度ICT人材が自律的に輩出されるようなメカニズム構築の必要性が唱えられています。

本考察は、こうした状況を踏まえ、ICT業界に潜む本質的な、実際の、現実的な課題を抽出するとともに、高度ICT人材育成に向けた業界構造の改善のための問題提起を行うものです。

2. 課題

大きな社会問題としてクローズアップされたみずほ銀行のシステム統合問題や、東京証券取引所の誤発注問題などにより露見されたように、ICTのトラブルが及ぼす社会活動への影響は非常に大きなものとなっています。ICTの現場では、こうした世間を騒がすほどのトラブルではないにしても、当事者にとって非常に深刻な納期遅延や品質問題は、日常茶飯事のように起こっています。まず、こうした納期問題、品質問題が後を絶たない根本的な原因はどこにあるのかを考える必要があります。

3. 現状のICT開発現場における問題

3.1 開発フェーズと契約手続き時期の齟齬により引き起こされる問題

人材の採用現場におけるICT業界のイメージは、その技術的、社会的飛躍とは裏腹に、いわゆる3K（きつい、きびしい、帰れない）あるいは7K（3Kに加えて、規則が厳しい、休暇がとれない、化粧がのらない、結婚でき

ない)と揶揄されるように、人気業界とは言い難い状況があります。その大きな原因のひとつに、開発現場が常に納期問題、品質問題と隣り合わせにすることがあげられます。

ICT業界は、他業界と比べると歴史的にも浅く、また、ソフトウェアという目に見えにくい商品であるという特性があります。ICT業界と同様に、設計して、製造し納品するというビジネスサイクルを持つ建築業界のそれと比較するとその違いが顕著に現れてきます。例えば、私たちがマンションを購入する場合、チラシやパンフレットの他に、詳細の間取り図や、キッチンやフロアに施されている特徴的な構造について、購入決定前に、必ず詳しく説明された書類を元に確認するということを行います。しかし、ICT業界の場合は、まず、建築でいうところの間取り図や、その他の詳細部分の説明等については、実際に使うエンドユーザ部門ではなく、情報システム部門が窓口となって調整、確認するケースが多くなっています。もちろん、情報システム部門は、実際に利用する人ではありませんから、機能レベルの有無については検証できますが、使い勝手という部分では検証することは難しくなるため、その部分については、エンドユーザ部門をプロジェクトに参画させ検証を行います。しかし、エンドユーザ部門も画面や仕様について確認することを要請されても、全体の業務の流れの中での使い勝手までを含めて事前にその善し悪しを判断することは非常に難しい作業となります。また、購入意志決定は、そういった詳細の仕様が作り上げられる前段階のベンダー側の提案書をベースに判断、決定されるケースが多く、検討を進める上で、発注者側の求めるものと、ベンダー側の作る予定の範囲のものとのギャップがますます大きくなり、大幅な仕様変更や納期遅延、あるいは予算調整などが発生するケースが多くなる一因となっています。

大きな規模のシステムになればなるほど、金額も大きく、実績としての価値が大きいことも多いため、ベンダーは少々無理をしても受注したいと考えるのが市場原理の必然です。そのためかなり無理をして受注することもしばしばであり、ベンダーとしてはできる限り約束していないことまではしたくない、というよりもできないという状況でプロジェクトがスタートするケースが多くなります。逆に、発注者側は、提案段階のこんなに良くなる、というイメージを前提に購入決定をしているため、時間が経てば経つほど、つまり工程が進めば進むほど、知識も増え、より具体的にこうして欲しい、ああして欲しい、という欲求が膨らんで行くこととなります。

通常、開発工程は、概ね下記の通り進むとされています。

要件定義→U I 設計→詳細設計→プログラム設計→プログラミング→プログラムテスト→結合テスト→結合機能テスト→システムテスト→運用テスト→運用開始

以上の流れの中で、要件定義までがユーザ主導のフェーズ、U I 設計からがベンダー主導のフェーズかのように運用されることが多く、具体的には、要件定義がまとまった段階で、R F Pを出し、ベンダーからの提案を受け、発注先を決定するという手法が取られています。しかし、この要件定義の段階では、先ほどの建築の例でいうところの間取り図的なものは、そのある一例が示されているに過ぎなかったり、特徴的な部分が紹介されている程度の状況であったり、決して固定的なものとは位置づけられておらず、極めて流動性の高い状況にあります。間取り図や部品が固定的に決まるのは、詳細設計の段階となるため、製造者へ何を幾らで、ということを決定事項として意思伝達できるのは、本来、詳細設計を終えて、プログラム設計以降を発注する段階であるにも関わらず、その前段階で購入意志が決定されることとなります。ここに業界としての構造、商習慣上の問題、トラブルの元となるその大きな原因があるのです。

3.2 開発段階に引き起こされる問題

次に、詳細設計以降の問題を取り上げます。こうした作業を請け負い対応する専門家はシステムエンジニア（通称SE）と呼ばれています。SEには、優れたSEもいれば、必ずしもそうでないSEも存在します。システム開発はその開発する人によって大きく影響するとされていますが、ベンダーには、優れたSEばかり何人も存在しているわけではなく、大規模なシステム開発になればなるほど、必然的にそれほど優秀ではないSEもたくさんプロジェクトに参画してくることになります。すぐれたSEが行う設計は、難易度も高く、そうでないSEには理解が難しくなります。そのためプロジェクト全体としては、優秀なSEの設計レベルではなく、そうでないSEの設計レベルに合わせて仕事をするということが習慣化していきます。こうして進められたプロジェクトは、3.1で示されたように、元々発注者側と受注者側の思惑のギャップと相まって、仕様のにも品質的にも満足するものにならないため、顧客は仕様を追加し品質向上を強く要請するようになります。そうするとベンダー側は、その解決策として、プロジェクトマネジメントを強化し、プロジェクト運営を改善するという報告をせざるを得なくなります。本来、品質を向上させるためには、プロジェクトマネジメントを強化するだけでなく、根本的な設計レベルを向上させる必要があるのですが、それができないため、プロジェクトマネジメントを強化せざるを得ないことになります。しかしそれでは、やはり実際に品質をあげることができないため、増やされた管理工数と、追加やり直しになった開発工数により、どんどんコストばかりが積み重なって行くという負のサイクルに陥ってしまうのです。

3.2.1 品質の高い設計とは何か

こうした負のサイクルから脱却するためには、(1) ユーザ側との主体的分解点を、要件分析ではなく、詳細設計段階とすること。(2) プログラム品質を向上させるためには、プロジェクトマネジメントを強化するだけでなく、品質の高い設計を行うこと、言い換えると、品質の高い設計ができるSEにより設計させ、そのSEが他のSEを教育しSE全体の設計レベルを引き上げるということが必要になります。

システム開発は通常、大きく、ハードウェア、OS (operating system)、ミドルウェア、DB (database)、パッケージソフト、アプリケーションソフト、という構成要素から成り立っています。SEという専門家は、これら全体構成の決定とインテグレーション及びアプリケーションソフトの開発ということを含めてシステム開発として請け負います。このアプリケーションソフトの作り方に大きな問題が潜んでいます。アプリケーションとして開発する部分の一つに「データベースを読み込んで処理を行い必要なアウトプットを行う」という機能があります。データベースの処理は、データが増えれば増えるほど処理時間に大きく影響を及ぼしますので、できるだけシンプルに、すなわちデータベースへのアクセス回数を減らし、最小限のアクセスで求めているデータを抽出することが求められます。しかし、先ほど触れた優秀なSEには、そうした設計が可能ですが、ほとんどのそうでないSEは、この設計を行うことが出来ません。

一般的な開発現場で行われている開発とは、一つのアプリケーション処理を行うために、まず、関連するデータベースを全部抽出します。そして、その抽出されたデータベースに検索に必要な条件を一つずつ施していきます。つまり条件の回数だけ検索するという設計を行います。当然、データベース検索を行う回数だけ処理時間が積み重ねられます。しかし優れた設計を行うSEは、データベースを検索する回数が最小限に止めるにはどうすればいいかということを考慮した設計を行います。当然、データベースに検索を行う回数が最小限となり処理時間を要しない設計がなされます。一般的には前者の設計が行われており、後者の設計はなされていません。こうした状態は、UI設計以降はベンダーに任せられているため、発注者側からブラックボックスとなっていることが多く、発注者

により指摘されることはありません。ブラックボックス化しているためプロセスではなく結果だけが追求されることとなります。もうひとつの事象として、こうした設計が習慣化していることがあげられます。習慣化した要因には、技術開発の歴史とともに生まれた背景もあることも否めません。しかし、時代の変化とともに開発手法や手順も進化させる必要があり、そういった進化が遅々として進んでいない現状があることを認識する必要があります。また、開発現場は、納期問題、品質問題と常に隣り合わせに居るため、会社の開発標準の手法や、従来の開発手法を踏襲することが前提化しており、従来手法にこそ、これまで述べてきたリスクが存在しているにもかかわらず、リスクを冒してまで開発標準外の手法を用いたり、前例のない開発手法を用いることをしません。用いることが許されない重圧が現場にはあると表現した方が適切なのかもしれません。納期延伸や品質問題が後を経たない最も大きな原因の一つは、ここにも存在しているのです。にもかかわらず、こうした背景や要因から、この問題がクローズアップされることはほとんどありません。

4. 業界の構造上の問題解決

こうした問題は、非常に多くの問題が複雑に絡み合っているために一筋縄では解決することはできません。解決が難しいからこそ今なお習慣化が浸透し続けています。そのため、解決に向けては、まず、建築でいうところの間取り図を確認した上で金額と購入意志を決定することができるように、発注者側と受注者側の責任分界点と工程フェーズを合致させることが必要です。現実に習慣化されている商習慣を一朝一夕に変えることは容易ではありませんので、発注者側、受注者側、双方にとってメリットがあること示し、広く浸透させていく不断の努力が必要になります。

次に、ベンダーの開発手法の問題も、従来型の開発手法ではなく、明らかに性能も品質も向上する開発手法を常識化させるためには、同手法による開発実績を積み重ね、こうした手法を取りやすい環境を作ることが必要です。また、多くのSEが取り扱えるように教育機会を増やし広く普及させていくことも必要になります。

こうした取り組みを行うことによって、開発現場は仕様変更や納期延伸の苦悩から解放され、創造的で活気ある現場に生まれ変わることができます。そうすれば、3Kや7Kといった評価もされることなく、ひいてはより優秀な人材の集う業界へと変貌していくことができます。こうした現場の実態、現場に潜んだ本当の課題、それらを解決することこそが業界のイノベーションこそが、高度ICT人材が自動的に排出されるメカニズム作るための大きな第一歩になるのです。

5. おわりに

自動車や家電などものづくりにおける日本の技術は世界一位の呼び声が高い中で、同じものづくりであるICTは、まだまだ欧米の先行を許し、それを追従するという地位に甘んじています。しかし、一人ひとりの技術者や、技術力そのものが後塵を拝しているのではなく、今回問題を指摘している通り、業界構造に起因するものであると言えます。ですから、こうした業界構造を一変させるような業界にイノベーションを起こせる革新的な人材を輩出していく必要があります。優れた技術者を輩出することももちろん最重要課題の一つですが、併せて、優れた技術者を束ねながら、商習慣を打ち破り、品質の高く生産性の高いコストパフォーマンスの優れたシステム開発を広げることこそが、ICTに向けられた日本の国家競争力を刺さる中核技術としての期待に応えるために必要不可欠な道なのです。

引用・参考文献

「高度ICT人材育成に関する研究会」報告書 平成20年5月30日 総務省

「情報通信白書」2009年 総務省

ソフトウェア開発プロフェッショナル ステイブ・マコネル／著 日経BP社 2005.1

企業IT動向調査2009 (社)日本情報システム・ユーザー協会 (JUAS) 2009年4月8日

計算不可能性を設計する ITアーキテクトの未来への挑戦 神成淳司・宮台真司 株式会社ウェイッ 2007年4月
30日

日本のソフトウェア産業がいつまでもダメな理由 久手堅憲之 株式会社技術評論社 平成20年4月25日

ベンチャー社長で技術者で <http://el.jibun.atmarkit.co.jp/g1sys/> 生島 勘富