

今も続くソフトウェア危機に SECの役割を期待する



IPA 顧問

松田 晃一

■ OS/360 は今年 50 歳に

今年 2014 年はシステム 360 が生まれて 50 歳の記念の年だそうです。1964 年に IBM が発表したシステム 360 には、当時としては画期的な OS/360 が搭載されました。その OS/360 開発の責任者であったブルックスが、大規模なソフトウェア開発に苦勞した話を「人月の神話」と題した本にまとめたのがご存じの通りです。表紙に掲げられた、ターンの沼に足を取られ身動きできなくなった巨獣の図は、ソフトウェア開発プロジェクトの困難さを端的に表す図として印象に残っています。OS/360 発表の数年後の 1968 年には、NATO 主催の会議が行われ、初めて「ソフトウェア工学」という言葉が使われたとのこと。増大するソフトウェアの需要に対して、人海戦術による手作りの域を出ないソフトウェア開発は、生産性が低くとも供給が追いつけない状況、いわゆる「ソフトウェア危機」が叫ばれました。「このままでは世界中の人類すべてがプログラマにならないと、ソフトウェアの供給が間に合わない」と言われるほどで、このような危機意識のもとでソフトウェア工学が提唱されたのがご存じの通りです。

以来 50 年弱の年月が流れたわけですが、さてソフトウェア危機は解消されたのでしょうか？

■ 今も続く「ソフトウェア危機」

相変わらずソフトウェアの生産性の低さは大きな問題です。ソフトウェアの欠陥によって運用中のシステムが停止し、社会に大きな混乱をもたらす事故は後を絶ちません。SEC journal で 2010 年から連載している IT システムの事故の集計を見ても、マスコミで報道されるよ

うな事故だけでも月に 2～3 件のペースは変わらず、むしろ増える傾向にあります。更に近年では、組込みソフトウェアの問題も大きくなってきています。例えば、自動車の組込みソフトウェアの不具合が原因で大量のリコールが行われた件数が、今年初めから 8 月までで既に 10 件近くに上っています。

また、開発予算の大幅な超過や、完成時期を大きく遅らせ損害賠償を請求され開発会社が赤字に陥ってしまう、といった問題もあります。更に問題なのは、開発が途中で頓挫してしまう、あるいは開発は終わったが使い物にならずお蔵入りしてしまう失敗プロジェクトがかなりあるという事実です。このようなプロジェクトの失敗は、IT システムの事故以上に表面化し難いため、実態が把握できないのですが、米国スタンディッシュ・グループの 2013 年のレポートでは、完成に至らず途中でキャンセルされたり、完成したが使われなかった完全な失敗プロジェクトが 18%～21% というデータが示されています。ほかに、大幅な予算超過、納期遅延、機能不足などを起こしたプロジェクトが 42%～43% で、大きな問題なく成功したプロジェクトはわずかに 37%～39% 程度と報告されています。残念ながら、日本にはこの種のデータが全くありませんが、会計検査院が官公庁システムの開発失敗を指摘したり民間企業同士が開発の失敗を巡って裁判で争うなどによって明るみに出ることがありますが、多分氷山の一角でしょう。特許庁のシステムやスルガ銀行のシステムなどでは、開発の失敗によって数十億円規模の損失が報道されたのは記憶に新しいところです。それほど高額ではなくても、開発の失敗によって億円単位の損害賠償を争う訴訟や調停に持ち込まれる事例は結構多いようです。私もここ数年、東京地裁で IT 関係

の民事訴訟や調停に専門委員・調停委員として参画するようになって、このような実態にはじめて触れるようになったのですが、現場では問題が山積している印象です。

このように見てくると、今も「ソフトウェア危機」は解消されたとはとても言えず、一層困難な状況にあると思います。

■ 古くて新しい課題だが...

もう少し具体的な課題に触れてみたいと思います。例えば、プロジェクトが頓挫する失敗について見れば、システムに対するユーザの要求をいかに消化し適切な形にまとめて合意を得ながら設計を進めていくのか、そして開発のために必要なリソースや期間を早い段階でどう見積もれば良いのか、といった古くからの課題にまだ満足な答えが得られていないからこそ、このような失敗が起きます。更に外注形態の開発における発注者と受注者との間の協力義務やプロジェクト管理義務、善管注意義務などプロジェクトに取り組む姿勢の問題や適正な契約のあり方など、技術面以外の問題もあります。これらも、決して新しい課題ではありませんが、まだまだ解決したとは言えません。

また、ソフトウェアの欠陥による事故はどうでしょう。欠陥を作らないための開発の工夫はもちろん重要ですが、それでもなお欠陥は残ることを前提に運用・管理を行うことも重要です。エンタープライズシステムのように専門の運用者・管理者が居るケースに比べると、消費者が直接操作する組込み機器においては、信頼性や安全性、使い易さ、判り易さなどの品質はとくに重要です。人間の要素を加味したリスク分析や間違いの無い設計・実装の保証、品質の客観的な検証・確認などはまだまだ手探りの状態です。

更に、日々激しく変化するビジネス環境に応えられるシステムを開発するには、開発内容や計画の全体を開始時に決めて後戻りしないウォーターフォールモデルでは、限界があることは明らかですが、それに代わる開発モデルは確立できたのでしょうか？有力な候補であるアジャイル開発は、外注を使わざるを得ない大規模開発にも適用できる形に洗練されてきたのでしょうか？

課題を挙げれば、キリがないのでこのあたりにしますが、どの課題も古くから言われてきた課題です。しかし、今なお残されている新しい課題でもあるのです。

■ ソフトウェア・エンジニアリングのナショナルセンターに

では、SECはどんな役割を果たすべきなのでしょう？

ここまで述べてきた色々な課題解決をSECが直接カバーすることはもちろん不可能です。SECは、このような課題に取り組む産学官の多くの人々の「拠り所」としての役割を果たすことにあると思います。ソフトウェアにかかわる産学官の多くの人達が、利用者、開発者、保守者、研究者、教育者など様々な立場から現場の問題を共有し、解決に向けた方向を議論し、活発な情報交換を行う「拠り所」となり、そこで得られた多くの知見を結集して、現場の課題解決を牽引するナショナルセンターとしての機能を果たすことこそ、SECの大きな役割だと思います。

現在SECが重点的に取り組んでいるソフトウェアの信頼性は、市民が安心して利用できるITシステムのために重要であることに疑いはありません。しかし、その実現はソフトウェア・エンジニアリングの幅広い技術の基盤に支えられて初めて得られるものです。基盤となるソフトウェア・エンジニアリング技術全般を底上げすること、そして取引や契約のあり方の問題、知的財産権の問題、人材の問題など技術面だけに偏らず幅広く課題を取り上げて行って欲しいと思います。

2009年から2010年にかけて事業仕分けに翻弄されたことを思い出します。組織のあり方を不断に見直すことはもちろん重要ですが、大衆受けを狙ったパフォーマンスに振り回されるのはご免です。これからの社会におけるソフトウェアのあり方を見据えて必要な役割は自信をもって果たしていくべきです。

ソフトウェアが様々な形で社会を動かす「ソフトウェアこそが主役」の時代を迎え、「ソフトウェア危機」は一業界の問題ではなく、あらゆる産業の競争力、イノベーションを制する問題であり、社会全体の力を決定づける基本的な問題です。ますますSECの役割の重要性と期待は高まっています。