

ソフトウェア産業の国際競争力を高めるには、ソフトウェア・エンジニアリング教育が重要。



IT(Information Technology:情報技術)が社会や生活に浸透する一方で、金融機関の情報システムのトラブルや、携帯電話の回収など、ソフトウェアの不具合が社会的な混乱を招いています。社会や生活に対するソフトウェアの影響力が大きくなる中で、ソフトウェアの品質を高めることがソフトウェア産業、ひいては日本の産業界の急務となっています。

電気通信大学 電気通信学部 システム工学科 ソフトウェア工学研究室の講師 西 康晴氏は、大学でソフトウェア・エンジニアリングを研究・教育する一方で、ソフトウェア・テスト技術者や組み込みソフトウェア技術者のコミュニティをリードし、さまざまな方向からソフトウェアの品質を高める技術を研究しています。

わが国におけるソフトウェア・エンジニアリングの現状と今後の目指すべき方向を、日本アイ・ビー・エム エグゼクティブITアーキテクトの榊原 彰が伺いました。

Special Interview ②

Education in Software Engineering is vital to strengthen the international competitiveness of Japan's software industry

While IT (Information technology) has become the indispensable part of the modern society and our daily lives, the computer failures caused by programming errors has become a new social menace, causing the financial institutions' information system troubles and led to a recall of defective cellular phones. It has become an urgent ask not only the software industry, but for the Japanese industry as a whole to improve the quality and reliability of software, as the society increases its dependence on IT.

Dr. Yasuharu Nishi wears many hats-as an educator, he teaches, as an assistant professor, the software engineering at University of Electro-Communications' Department of Systems Engineering. He is also a leading figure in the software engineering community, conducting a wide array of researches to improve the quality of computer software from various angles.

Akira Sakakibara, Executive IT Architect of IBM Japan, will interview Dr. Nishi to find out the current situation of Japan's software engineering and the directions it should seek in the future.

大学のソフトウェア・エンジニアリングと 企業のソフトウェア・エンジニアリング

私は、電気通信大学でソフトウェア・エンジニアリングの研究および教育を行っています。

ソフトウェア・エンジニアリングについては、大学の教官として、主として次の三つの活動にかかわっています。

一つは、研究活動です。企業や研究機関との共同研究を行っています。二つ目は教育であり、学生への研究指導です。電気通信大学で教えるだけでなく、ほかの大学でも講義を担当しています。三つ目が社会貢献活動としての普及活動です。後から詳しくご紹介しますが、幾つかの技術者コミュニティで、現場の技術者の方たちと共同研究を行っています。

また、大学に奉職する以前は、ある企業でコンサルティング部門を立ち上げて、ソフトウェアの品質やソフトウェア・テスト技術に関するコンサルタントとして働いてきました。

大学と企業という二つの異なる環境でソフトウェア・エンジニアリングにかかわってきた経験を基に、わが国におけるソフトウェア・エンジニアリングの現状と今後の目指すべき方向を、お話しさせていただきたいと思います。

企業におけるソフトウェア・エンジニアリングの現状

まず言えることは、企業・大学のいずれもがソフトウェア・エンジニアリングへ取り組むに当たって、解決すべき多くの課題を抱えているということです。

企業については、ごく一部の先進的な会社を除いて、全体的にソフトウェア・エンジニアリングの導入が遅れています。その理由は、端的に言えば、ソフトウェア・エンジニアリングを導入しなくても、ソフトウェアをつくることのできるからです。

自動車工学を勉強していない自動車エンジニアや、航空工学を勉強していない航空機エンジニアがいるとは思えません。ところが、ソフトウェア技術者がソフトウェア・エンジニアリングを勉強していないというこ

電気通信大学 電気通信学部
システム工学科
講師 工学博士
西 康晴氏

Yasuharu Nishi

Ph.D., Assistant Professor
Department of Systems
Engineering
The University of Electro-
Communications



とは多々あります。妙な話ですが、これが現状です。ソフトウェア・エンジニアリングを学ばずに、ソフトウェアをつくることで、そのしわ寄せが結果的に自分たちに戻ってくるということを、ソフトウェア技術者は認識すべきです。

もちろん、現場の技術者は日々の開発に追われていて、あまりにも余裕がないことが問題なのですから、マネジメント側の責任も大きいですね。

その一方で、ごく一部ですが、ソフトウェア・エンジニアリングの重要性に気付き、意欲的に導入している企業もあります。特に車載機器の組み込みソフトウェアの分野は、人命に直結していることから、ソフトウェア・エンジニアリングの導入に積極的です。

とはいえ業界全体としては、成熟したものづくりのレベルに達していないと言ってよいでしょう。これは非常に憂慮すべき状況です。

大学におけるソフトウェア・エンジニアリングの現状

一方、大学におけるソフトウェア・エンジニアリングの現状に目を向けると、その地位が非常に低いといえます。

根本的な問題として、大学の情報系の学部・学科はソフトウェア・サイエンスが主流であり、ソフトウェア・エンジニアリングのコースをつくりたくても、教官の絶対数が少ないという問題に突き当たります。

また、ソフトウェア・エンジニアリングの学問体系は、

ソフトウェア・サイエンスとはずいぶん異なりますが、それにもかかわらず、ソフトウェア・エンジニアリングのカリキュラムをソフトウェア・サイエンスの一部として考えている人が多いという問題があります。これは、先進的な米国でさえその傾向が見られます。

特に日本の場合、情報系の学部・学科は、電気や機械、数学などの学部・学科の一分野としてもともと位置付けられていたからでしょう。また、日本では、経営工学などのものづくりの一つとしてソフトウェア・エンジニアリングを研究することも少なくありません。いずれにせよ、ソフトウェア・エンジニアリングを体系的に学べる学部・学科が日本にはほとんどありません。

学会における議論も、学会であるからにはアカデミックになるのは当然としても、本来は表裏一体である「技術」と「管理」を分けて議論しがちであるなど、開発の現場をまったく考えていない場合が見受けられます。

特に、大学の研究者がソフトウェア産業におけるオピニオン・リーダーになり得ていないことが問題です。企業の技術者は、所属企業の利益に反した発言をするのは難しいはずですし、仮に個人の立場で発言したとしても、その背景を勘ぐられることがあるでしょう。そういった懸念が研究者にはないにもかかわらず、ソフトウェア産業が進むべき方向を示すことがなかなかできないでいます。むしろオピニオンを持っているのは、企業の技術者であることが少なくありません。ソフトウェア・エンジニアリングは実学なので、研究者はもっと産業全体をリードするという気概を持つべきだと考えています。

ソフトウェア・エンジニアリング教育の品質認証を

大学におけるソフトウェア・エンジニアリング教育を考えたときに、私自身も委員として参加している情報処理学会 アク্রেディテーション委員会の活動が、一つの方向を示せるのではないかと思います。

アクレディテーション委員会は、IT(Information Technology: 情報技術)関連の教育機関における品質認証を検討し、各分野での学習/教育目標を達成するために必要な教育内容やカリキュラムを定める

ことを目的としています。

米国などの事例を参考に積極的な取り組みを進めていますが、こうした取り組みが実効力を持つには、解決すべき問題がまだまだあります。

企業の方には、ISO9000を例に取って説明すると分かりやすいでしょう。今日、多くの企業がISO9000を取得していますが、それにより企業の品質マネジメントが改善に向かっていくのか、それとも取得のための文書やマニュアルでがんじがらめになってしまうのかは、トップの意思の強さに依存します。自社の技術や品質を向上させたり、生産性を上げたいとトップが強く思わない限り、ISO9000のみならずこの種の品質システムは形がい化し、それほど役に立ちません。

アクレディテーションも同じです。アクレディテーションのカリキュラムを活用することで、教育レベルを上げようという意思が大切です。アクレディテーションを取得するには、大学側にはさまざまなアクティビティが求められますが、そうした努力は、単に取得することを目的としている限り無意味でしょう。「ほかの大学も取得しているから」というスタンスではなく、アクレディテーションの意味を十分に理解した上での取り組みが大切です。さもないと、産業界への優秀な人材の輩出は望むべくもありません。

CCSE導入の期待と課題

ソフトウェア・エンジニアリングの知識体系としてSWEBOOK(Guide to the Software Engineering Body of Knowledge: ソフトウェア・エンジニアリング基礎知識体系)はよく知られていますが、このようなコンセプトから教育のカリキュラムを構築しているものにIEEE(Institute of Electrical and Electronics Engineers: 米国電子技術者協会)/ACM(Association for Computing Machinery: 米国電子計算機学会)のCCSE(Computing Curricula/Software Engineering: コンピューティングとソフトウェア・エンジニアリングの教育カリキュラム)があります。

ところが、日本の大学でCCSEに基づいたカリキュラムを導入するには、三つの難点があるのです。

一つは、高校までの情報教育の不足です。CCSEを

表1. CCSEを基にアクレディテーション委員会が策定したカリキュラムJpn1の主要科目

科目名	単位数	科目説明
SE1 構築技術とツール及び演習	2.0	構築技術とツールについて学ぶ。構築技術やツール、設計の基本概念、コンポーネントの設計などを扱う。またツールを用いた演習を行う。
SE2 形式手法とモデル論	2.0	形式手法とモデル論について学ぶ。仕様記述言語、正当性検証、モデリングの概念、事前条件や事後条件などを扱う。
SE3 工学基礎	2.0	工学基礎について学ぶ。実験の手法、統計解析、能力評価、セキュリティ、スケーラビリティ、工学的設計の基本概念、ソフトウェア工学以外の分野における工学などを扱う。
SE4 エンジニアリングエコノミクス	2.0	エンジニアリングエコノミクスについて学ぶ。ライフサイクルにおける価値の変化、参加型デザインやステークホルダー分析、QFDなどによる製品企画、費用対効果分析、価値評価、経済的な影響などを扱う。
SE5 組織論とマネジメント	2.0	組織論とマネジメントについて学ぶ。グループダイナミクス、人間の心理、マネジメントの基本概念、プロジェクトマネジメント、構成管理などを扱う。
SE6 コミュニケーションスキル及び演習	2.0	コミュニケーションと要求獲得の技術について学ぶ。コミュニケーションスキル、インタビュー、ブレインストーミング、プロトタイプング、仕様化とドキュメンテーション、表現、折衝などを扱う。また演習を行う。
SE7 ソフトウェアモデル及び演習	2.0	ソフトウェアモデリングについて学ぶ。モデリングの基本原則、さまざまなモデル、要求の評価、要求の管理などを扱う。
SE8 ソフトウェア設計(1)及び演習	2.0	ソフトウェア設計について学ぶ。設計の概念、設計戦略、アーキテクチャ設計、設計の表記法とツール、設計の評価などを扱う。
SE9 ソフトウェア設計(2)及び演習	2.0	ソフトウェア設計について学ぶ。ユーザインターフェース設計、SA/SD、JSD、オブジェクト指向設計、デザインパターン、コンポーネントベース設計、インターフェース設計などを扱う。
SE10 ソフトウェア検証(1)及び演習	2.0	ソフトウェアテストについて学ぶ。テストの概念、単体テスト、例外の処理、さまざまなカバレッジ、結合テスト、ブラックボックステスト、システムテスト、受け入れテスト、品質特性、回帰テスト、テストツールなどを扱う。
SE11 ソフトウェア検証(2)	2.0	ソフトウェア検証について学ぶ。検証の基本概念、レビューやインスペクション、ユーザインターフェースのテスト、障害の分析と管理などを扱う。
SE12 開発プロセスと保守	2.0	開発プロセスと保守について学ぶ。プロセスの基本概念、プロセスの種類、品質管理、プロセス改善、規格や標準とテラリング、PSP、TSP、保守の基本概念、保守のプロセス、コストと計画、影響分析、リバースエンジニアリング、リファクタリングなどを扱う。
SE13 ソフトウェア品質	2.0	ソフトウェアの品質について学ぶ。品質の基本概念、社会的意義、品質コスト、品質特性、品質管理、プロセス改善、規格や標準とテラリングなどを扱う。

出典 : <http://blues.se.uec.ac.jp/acc-se/IPSJ-SE-Curriculum.html>

導入するには、プログラミングの経験をはじめとする基礎的な知識を習得していることが前提となりますが、現在の高校教育にそれを期待することはできません。そのため大学でCCSEに基づいたカリキュラムを組もうとすると、CCSEで規定された内容よりはるかに基礎的な教育から始めざるを得ません。

二つ目は、CCSEに基づいてカリキュラムを組もうとすると、ソフトウェア・エンジニアリングだけで13科目になってしまうことです(表1)。ところが、ソフトウェア・サイエンスを扱う情報系の学部・学科では、それを1科目で済ませているのが現状です。教官の不足と相まって、ソフトウェア工学科やソフトウェア工学部がほとんど存在していない日本では、CCSEのカリキュラムを組むのは現実的にほとんど不可能なのです。

三つ目の問題として、ソフトウェア・エンジニアリングを学んだ学生を、企業が積極的に採用するとは限らないということです。これでは、ソフトウェア・エンジニアリングを体系的に学ぼうとする学生のモチベーションは上がらないでしょう。

また、個々の企業がエンジニア教育のためにSWE-BOKやCCSEを導入するのも、そう簡単にはいきません。なぜなら企業の教育は、企業内教育だけで完結しない場合がほとんどだからです。外部のセミナーなどとの連携を考えると、整合性を取るのには難しいでしょう。

従ってSWEBOKのような知識体系や、CCSEのようなカリキュラムをソフトウェア産業全体で検討し、導入を推し進めていく必要があると考えています。ITSS (IT Skill Standard: ITスキル標準)や組み込みスキル標準は、よいきっかけになるのではないのでしょうか。

徹底的に考え抜くことが大切

学生にソフトウェア・エンジニアリングの指導をする際に注意していることを、ここで紹介しましょう。

多くの学生を対象とする講義では、「唯一無二の絶対の解が存在するわけではない」ということを強調し

ています。ソフトウェア・サイエンスでは、メトリックスを幾つか決めることで「最も速いアルゴリズム」のような最適解が存在します。しかしソフトウェア・エンジニアリングでは、現実世界という制約があり、その制約を十分に理解しつつ、しかもそれに引きずられないようにバランスの取れたモデルにする必要があります。つまり、状況が変わることで、正解に成り得るモデルも変わってくるということです。「正解がない」ということは、学生にはずいぶんインパクトがあるようですね。

また、マン・ツー・マンに近い教え方となる研究指導では、「きちんと考える」ということを意識して教えています。

具体的には、抽象的な概念をビジュアルに考えること、すなわちモデリングの大切さを教えています。それから日本語の読み書きや、論理的構成能力、コミュニケーション能力といった、日本語を使って考えるということを教えています。三つ目は、これは難しいのですが、妥協しないで徹底的に考え抜くことを意識させています。つまり、ロジックを考えると、徹底的に小さな単位にまで落とすようにさせるわけです。現実の問題の原因を分析する際には、抽象度を下げていかなばなりません。学生に考えさせると、いつの間にか抽象度が上がってループに陥ったり、あるいは論理が飛躍したりということがよく見られます。例えば、ある事象の原因を徹底的に追求していくと「社会全体の構造が悪い」とか「不景気が悪い」という結論に陥ることが少なくありません。こうした論理展開では問題の解決に結び付きませんから、徹底的にブレークダウンし、考え抜かせるようにしています。学生にとってはなかなか辛い作業のようです。

実は企業でも同じようなことがよくあります。

例えば品質問題の分析を依頼されたときには、最初にクライアント側の方に分析していただくのですが、その結果「社長が悪い」とか「組織が悪い」「お客様が悪い」という方向に帰着する 경우가少なくありません。そこで私たちが適切な分析を行うのですが、そうすると6～8割は自分達が原因であることが分かります。その結果、腹をくって改善活動に取り組みだすわけです(笑)。

SESSAMEの活動について

今まで述べてきたように、企業にしても大学にしても、ソフトウェア・エンジニアリングへの取り組み状況は相当厳しいものがありますが、これからご紹介するコミュニティ活動が一つの突破口になるのではないかと考えています。

私がかかわっている幾つかのコミュニティの中から、まずはSESSAME(Society of Embedded Software Skill Acquisition for Managers and Engineers:「組み込みソフトウェア管理者・技術者育成研究会」)について紹介しましょう。

SESSAMEは「世界との競争の中で、組み込みソフトウェアを日本の将来のコア・コンピタンスに」という目標を掲げ、2000年11月に発足しました。この趣旨に賛同した200名以上のメンバーが産・官・学の各分野から参加しておりまして、私自身は世話人として、その実、雑用係として(笑)活動させていただいています。具体的な活動内容としては、セミナーやワークショップを開催したり、組み込みソフトウェアの管理者・技術者向けのテキストや知識体系・用語集・文献集などをまとめてWebページで公開しています(図1)。

日本のソフトウェア産業は、創造性型産業において米国に劣り、またコスト競争型産業でアジア諸国に負けるといわれています。しかし日本にも強みがあります。その代表が組み込みソフトウェア(ファームウェア、埋め込みソフトウェア、エンベデッド・システム)です。



図1. SESSAMEのWebサイト(<http://www.sesame.jp/>)

組み込みソフトウェアは、ハードウェアのキャパシティーや、パフォーマンスなどに多くの制約があり、独特のスキルやノウハウが必要です。一方で、世の中のユビキタス化に伴い需要は急増しており、開発規模は爆発的に大きくなっています。また、それに伴う技術者の増大もうまくコントロールしていく必要があります。

そこでSESSAMEでは、組み込みソフトウェア技術者や管理者を数十万人育成するために、カリキュラムを整備し、テキストや用語集などを提供しています。これはすべて、日本の産業競争力を強めたいと願うボランティアの活動です。

組み込み系ソフトウェアへの期待

先ほども述べたように、日本のソフトウェア産業を諸外国と比較したときに、新しい発想で新しいものをつくっていく力は欧米にかなわないでしょう。また、コスト競争では中国・インドをはじめとするアジア圏の国々に勝てません。しかし日本には「きめ細かさ」や「すり合わせ能力」といった武器があります。

そういった能力の基盤となるのが品質技術です。

東京大学の東京大学ものづくり経営研究センターの藤本 隆宏先生は、日本の自動車産業が高い国際競争力を持ち続けているのは、すり合わせ型の開発/生産体制を維持しているからだと分析されています。

ハードウェアの場合、欧米・韓国のメーカーが得意とするのは、業界標準の部品を買ってきて組み立てるオープン・モジュラー型の開発・生産ですが、日本メーカーは、自社開発の仕様の部品をすり合わせていく技術、すなわちクローズ・インテグラル型の開発・生産が強みを発揮しています(図2)。

しかしソフトウェアは少し異なります。マーケットのスピードに追いつくために、オープンな部品を調達し組み合わせてシステムを構築しなくてはなりません。一方で、業界標準のOS(Operating System)なり、データベースなり、ミドルウェアを買ってきて、それをそのまま組み合わせればすぐ動くということにはなりません。すなわち、オープンな部品を迅速にすり合わせるノウハウが競争力の源泉なのです。オープン・インテグラルな開発形態といえるでしょう。これは、組

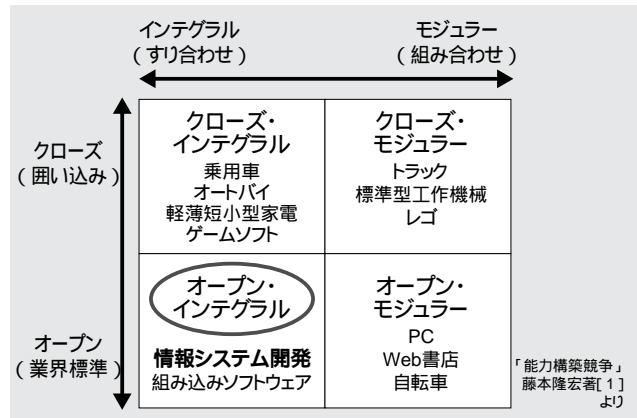


図2. オープン・インテグラル型とクローズ・モジュラー型

み込み系もエンタープライズ系も同様です。

こうしたソフトウェア製品の特徴を考えると、日本のソフトウェア産業が進むべき方向を見つけなければならないでしょう。

TEFの活動について

私の別の活動としてはTEF(Testing Engineer's Forum: ソフトウェア・テスト技術者交流会)というコミュニティも主宰しています。このコミュニティは、情報交換や技術向上を目指したソフトウェア・テスト技術者のためのものです。主な活動はメーリング・リストによる情報交換ですが、不定期に勉強会や研究会も行っています。ソフトウェア・テストに携わっている技術者であれば誰でも参加することができ、現在、約800名のメンバーが参加しています。

TEFを立ち上げたきっかけは、ソフトウェア・テストに関する情報が少ないことに私自身が困ったからです。当時は大学の博士課程で、ソフトウェア・テスト技法の研究を進めていたのですが、ポーリス・バイザーやG.J.マイヤーズの訳書がある程度で、日本語で読める技術的な文献がほとんどありませんでした。

一方、米国では、ソフトウェア・テスト技術関連の本の刊行数も多く、またメーリング・リストやニュース・グループ、Webサイトもたくさんあって活発に活動していました。

「日本にもソフトウェア・



TEF勉強会の模様

テスト技術者はたくさんいるはずだし、米国のようなコミュニティがあればいいですね」という話のある方にしたところ、「それなら自分でつくればいい」と勧められたのがきっかけです。

TEFでは、ソフトウェア・テスト技術に関する書籍をメンバーで共同翻訳するなど、意欲的に活動を続けています。ソフトウェア・テスト技術者や品質技術者に対しては、それなりの寄与をしてきたと思いますが、今後は、開発技術者や管理者などに向けた上流への貢献が必要だと思っています。

例えば具体的には、ソフトウェア・テストを意識したソフトウェア設計を行うといったことが挙げられます。ソフトウェア・テストの工数は、プロジェクト全体の少なくとも3割、多いときには9割を占めるといわれています。ですから、仮にソフトウェア・テストの生産性を2倍に上げられれば、プロジェクト全体の生産性が大幅に向上しますし、コスト削減にもつながります。プロジェクト全体のボトルネックとなっているソフトウェア・テストを改善すれば、生産性やコスト、品質に大きく影響するわけですね。

そのため、開発技術者やマネジャー、さらには経営者層にもソフトウェア・テストの重要性を意識してもらえるように、上流に対する働き掛けをしていかなければならないと考えています。

JaSSTについて

また、2003年からはJaSST(Japan Symposium on Software Testing:「ソフトウェアテストシンポジウム」)の共同実行委員長として、シンポジウムを開催して、ソフトウェア・テストに関する活発な議論や情報交換を行っています。

2004年1月に東京コンファレンスセンター品川で開催されたJaSST04には、2日間で延べ800名の来場者があり、とても盛況でした。

その理由は、一つには構造化分析(Structured Analysis)やピープルウェア(Peopleware)などを提唱されたトム・デマルコ氏をお迎えして基調講演やチュートリアルを行っていただいたことが挙げられるでしょう。何といても氏の影響力は絶大ですから。



JaSST04 パネル・ディスカッションの様相

私は、ワインバーグやブルックス、デマルコといった方々がソフトウェア開発の「技術」ばかりではなく、「人」に焦点を当てた著作を発表してきたことは、過去40年にわたるソフトウェア・エンジニアリングの歴史の中でも、最も偉大な貢献であると考えているほどです。

しかし最大の理由は、現場のエンジニアがソフトウェア・テストで困っていて、解決のヒントを求めているということではないでしょうか。

もちろんシンポジウムが成功した要因として、企画や研究発表の内容が粒ぞろいであったことは言うまでもありません。実行委員の方たちが「こうした企画を立てれば人が集まるだろう」ではなく、「自分が聞きたいものを企画する」というスタンスで楽しみながら取り組んだことも大きいでしょう。

JaSSTはソフトウェア・テストに関する技術交流を行う技術見本市であり、品質意識の向上や、ソフトウェア・テスト技術への投資に対する意識を向上させることを目的としていますが、それだけでなく、ソフトウェア・テスト技術者のモチベーションを高めるカンフル剤としても必要だと思っています。実は私は東京の下町出身で、子供のころは目黒不動の近所に住んでいて、10日に1回は縁日があってとても楽しかった思い出があります。JaSSTを、ソフトウェア・テスト技術者たちの縁日のように盛り上げたいと思っています。

日本アイ・ビー・エムへの期待

JaSSTの活動に取り組んで3年になりますが、日本アイ・ビー・エムに期待することが二つあります。

日本アイ・ビー・エムの技術者の方々は、品質に対する意識が高く、また高度なスキルも持っています。

しかもワールドワイドの視野を持つ技術者が少なくありません。以前からそうした方々にお手伝いしていただいておりますが、引き続きJaSSTの活動に参加していただければ、業界全体の底上げができると思っています。これが一つです。

二つ目は、ソフトウェア・エンジニアリング技術の発展のために、今まで以上にこの分野に投資をしていただきたいということです。日本アイ・ピー・エムは、ソフトウェア品質に関しても世界をリードしているのですから、そういった形で業界全体を支援していくこともリーディング・カンパニーの役割ではないでしょうか。業界全体への貢献が、結果的には日本アイ・ピー・エムの利益にもつながるはずですよ。

例えばJaSSTのようなコンファレンスを後援し、ソフトウェア・テスト技術者のスキル・アップや意識向上を支援することは、そのまま日本アイ・ピー・エムのソフトウェア品質に関するスキルをアピールすることにつながるはずですよ。さらに言えば、日本アイ・ピー・エムがこの分野に積極的に投資することで、業界全体、特にユーザー企業に「ソフトウェアの品質は高くなくてはいけない」という意識が浸透していくのではないのでしょうか。

つい最近、ある技術誌に「ソフトウェアのプロジェクトの成功率は26.7%」という統計結果が掲載されました。ハードウェア産業では信じられない数字であり、ソフトウェア産業の品質は現状では極めて低いと言わざるを得ません。ソフトウェア・エンジニアリングにかかわる者として、この結果はくやしい限りです。

とはいっても、第二次世界大戦直後の日本では、ハードウェアの品質もそれほど高くはありませんでした。業界全体の努力によって、今日の世界に誇る水準に至ったわけですよ。日本アイ・ピー・エムをはじめとするソフトウェア産業のリーディング・カンパニーが、業界全体の技術の発展に寄与することで、ソフトウェアの品質もハードウェアに負けないものとなっていきましょう。

これからの若いエンジニアに求めるもの

ソフトウェア・エンジニアリングや品質技術について

て学ぶエンジニアの方たちへのコメントとしては、「笑顔で仕事ができること」という言葉に尽きますが、具体的には三つのアドバイスがあります。

一つ目は「マネジメント」です。これはプロジェクトマネジメントという意味もありますが、むしろ広い意味での「経営」を意識するということです。目の前の技術だけを見るのではなく、自分のつくったものがお客様にどのように使われるのか、それによりお客様が幸せになれるのかどうか、そしてそれがどんな社会的影響力を持つのか、ということを考えてほしいのです。自分の働きによって会社や業界がどう変わり、どう良くなっていくのかということを常に意識できる技術者になっていただきたいと思います。

二つ目は、エンジニアとしての「モラル」です。技術倫理や、品質意識を持つことです。いい意味でのプライドの高いエンジニアになって、腕に自信のある頑固おやじのような存在になってほしいと思います。

三つ目は「勉強し続ける」ことです。勉強を始めるのはたやすいことですが、それを続けていくのは困難です。ドッグイヤーといわれるようにITの世界は日進月歩ですから、足を止めた瞬間に時代遅れになりかねません。昨日より良い仕事をするために勉強をし続けて、常に新しいものを吸収する技術者になってください。

最後に、経営者層の方々に一言付け加えさせてください。長い間、日本人は「水」と「安全」と「品質」は無料で手に入るものだと思ってきました。しかしながら、今やミネラル・ウォーターはガソリンより高価ですし、家族や家の安全のために、一般家庭でさえセキュリティ会社と契約を結ぶような時代になっています。では、品質はどうでしょうか？ 少なくとも、現場が頑張りさえすれば、品質は確保できるという考え方は間違いです。品質も、水や安全と同様に、それ相応のコストを支払わなければ手に入りません。

しかし、飲めばなくなってしまう水とは異なり、品質に投資すれば、それ以上のリターンがあります。ソフトウェアの品質への投資は、長期的に見ると、必ずや自社の大きな利益となり、アドバンテージとなるのです。

参考文献

[1] 藤本 隆宏「能力構築競争」中央新書、2004年6月20日、7版