

ソフトウェア産業とソフトウェア研究の沈滞状況について

玉井 哲雄 (東京大学大学院総合文化研究科)

1 はじめに

SEA の幹事のメーリングリスト上で、幹事の新谷さんが「情報処理」にのった大須賀論文(「日本のソフトウェア問題についてー現状分析と将来対策ー」, 1997 年 8 月号)を取り上げて問題提起をされた。それでしばらく前に書いて放ってあった文章を思い出した。この元は、1995 年の 12 月に日本ソフトウェア科学会のソフトウェア工学の基礎研究会の招待講演で、「何がソフトウェア工学の基礎となり得るか: ソフトウェア工学の沈滞を打開するには」と題して話した内容の前半部分である。その時は OHP を使ってしゃべっただけだが、後でそれを材料に一部を文章化してみた。その後の動きを若干捕捉してあるが、それでも時機を失っている感があるかも知れない。しかし、これまでどこにも発表していないので、SEAMAIL に投稿してフィードバックを得るのも意味があるかと考えた。発表していない理由の一つは、最後に「ではどうしたらよいか」という自分なりの展望と提言を書こうと思ったからである。しかし、とりあえずこのまま出して、皆さんの意見を聞き、私の考えるところは別の機会に譲るとほぞを固めた。未完成なものをあえて皆さんの目に晒すことになるが、お許しいただきたい。

大須賀論文や新谷さんの問題提起とは必ずしも対応していないかも知れないが、議論のきっかけになれば幸いである。

2 ソフトウェア産業と研究の現状

現在、日本のソフトウェア産業にもソフトウェア研究にも、停滞感が漂っている。世上では目覚ましい製品も革新的な研究も、日本から生まれてこないといわれている。産業では、二重の意味での空洞化が進んでいる。すなわち、核となるようなソフトウェアは米国に支配され、大量の人手を投入して作られる適用型のソフトウェアは、技術力があって労働コストの安い国への生産の委託が進んでいる。企業では「作らずに買う」というのが、方針になっているという。買いにいく先は、米国やヨーロッパだったり、中国やインドだったりするわけだ。

日本のソフトウェア産業は、1990 年頃まではともかくも拡大を続けてきた。ソフトウェア危機という言葉が、初めはソフトウェア工学の必要性を叫ぶための枕詞として、後にはソフトウェア産業政策の正当化のための合言葉として、長らく呪文のように唱えられてきたが、それが世の中のソフトウェアに対する需要の増大に対して供給体制が追いつかないということを意味していたとすれば、ソフトウェア業界にとってはこれほど歓迎すべきことはなかったはずである。問題はプログラムの確保で、仕事を確保するための営業を必要とするような会社は、よっぽどおかしな会社だと思われていた。通産省によるソフトウェア技術者の 97 万人不足説(西暦 2000 年時点の予測)も、こうした背景で出された。しかしバブル経済ははじけ、1990 年代に入ってこれまでにない長い不況が続いている。この不況はまた、コンピュータやソフトウェアという不況の影響を受けにくかった産業に、きわめて大きな打撃を与えた点でも、これまでの景気後退や不況と異なる性格のものである。

経済のバブル化現象が顕著だった時期、並行してソフトウェアのバブルも大きく膨らみ、それがやはり 90 年前後にはじけたのではないだろうか。つまり、ソフトウェアの需要は次のような理由により、実需よりもかけがはるかに増大していた可能性がある。

- 同じようなソフトウェアを別々の組織が重複して作る。同じ業界の各社が横並びに、しかしそれぞれ別に、同じような機能の業務システムを作るようなケースである。
- 同じようなソフトウェアを同じ組織が何度も繰り返し作る。業務内容、利用環境、技術の変化に応じて、ソフトウェアシステムも変化せざるをえないが、設計技術や保守・再利用技術のまずさから、むだに作りなおしているケースである。
- 下請構造により全体の売り上げ高が増大する。ソフトウェア産業は非常に多くの企業を抱えるが、それらは階層的な下請構造を作っている。ソフトウェアの開発委託が、ユーザー企業からコンピュータメーカーや大手のソフトウェア会社に出されると、多くの場合、階層の上から下に何層にもわたって外注委託が流れていく。これらは経済活動としては重複して数えられ、みかけの産業規模は増大する。
- これらの要因の原因でもあり結果でもあるが、日本におけるパッケージ型のソフトウェアの流通は、個別発注型のソフトウェアの開発と比べてきわめて小さい。売上高でいえば、いまだにソフトウェア産業の15%程度を占めるに過ぎない。正確な比較データはないが、米国ではパッケージ型のソフトウェアの売り上げ比率は60%程度と言われる。
- 個々のソフトウェアの規模自体が、いたずらに増大している傾向もある。冗長な作り方、使われない機能などで水膨れになっているソフトウェアが多い。

日本の大学や研究機関におけるソフトウェア研究についても、風当たりが強い。ここでも、産業界と並行するように「空洞化」現象が見られる。新たな研究分野や基本的な概念は、米国から持ち込まれる。革新的な構想や研究成果が海外に向けて発信されることは少ない。そもそも、研究の道具として用いる基本的なソフトウェアも多くが米国製である。そしてよく批判されるように、大学における研究は実践との結びつきが希薄である。日本では一般にベンチャ型の企業あまり育っていないとよく言われるが、とくに大学から生まれたベンチャ・ビジネスはほとんどない。さらに、産業が元気をなくしているために、肝心の人材育成の面でも、たとい質の高い教育ができていたとしてもそれを産業界が受け入れてくれるかどうかに大いなる不安があるという有様である。総じて、日本のソフトウェアの産業界も学界も、自信を喪失しているようにみえる。

3 悲観論の蔓延

このような認識は基本的に間違っていないだろうが、あまりに悲観論に傾きすぎているようにも思われる。5-6年前を振り返ってみるとよい。当時、米国では日本のソフトウェアの信頼性、生産性の高さが一つの脅威として論じられていた。たとえば、M. Cusumano による日本の「ソフトウェア工場」についての報告は、その脅威に対して警鐘を鳴らしたものとして受け止められた。このままでは、鉄鋼、家電、自動車のように、ソフトウェアも日本にやられるという論調がよく聞かれたものである。

筆者にとって印象深いのは、1991年のテキサス州オースティンにおけるICSEである。そこで、W. Humphrey がCMMの日本と米国における調査結果を発表した。この2つの調査は、日米それぞれで開いたセミナーの参加者に、アンケートを取るという方法で行われたものである。対象者が、米国の場合は主に防衛関連を中心とした埋め込みソフトウェアの開発に携わる企業の技術者、管理者で、日本の場合はほとんどが事務処理システムに携わる企業からの参加者だったから、単純には比較できないということをハンフリーは強調した。それにもかかわらず、比較すれば米国の方がプロセスの成熟度が高いと読める結果に対して、米国人の聴衆が「その結果はおかしい」とする反応の激しさは驚くほどだった。その頃、米国で常識化していた日本のソフトウェア開発力の強さという思い込みとは、ずいぶん異なるメッセージだったからである。その認識がまったく逆転した今となっては、この6年前の状態が信じられないくらいである。

明らかにこの時の米国側は、日本の実力を過大視した。同時に脅威論を煽ることにより産業への補助政策や研究への予算拡大をねらう政治的な意図もあっただろう。現在では、このような認識はまったく逆転し

た。たとえば、1994年にスタンフォード大学の Feigenbaum 等が日米のソフトウェア企業 50 社あまりを調査して作ったレポートでは、日本の企業が品質を重視するあまり、売れる製品を市場に出すスピードに遅れを取っているという結論を導きだしているが、全体に米国の産業に対する自信に満ちあふれている。一方、1993 年に出版された棟上レポートは、やはり日米のソフトウェア産業を比較して見事な分析を行っているが、総じて日本の状況に対してきわめて悲観的である。

5 年前の米国における現象と、昨今の日本における悲観的な見方とを重ねあわせてみると、現状認識が極端から極端に振れる傾向が見てとれる。確かに日本のソフトウェア産業と研究が現在大きな問題を抱えていることは、認識しなければならない。しかし、必要以上に悲観視することも、生産的ではない。

ソフトウェアが問題だという認識は、実は日本に限った現象ではない。非専門家向けに解説されるソフトウェアの現状は、その供給力や信頼性に関しておそろしく否定的である。たとえば、Scientific American 誌の 1992 年 11 月号に、「ソフトウェアのリスク」と題する解説があるが、ソフトウェアの信頼性が不確実であるという事態に対し、可能な対処法としてあげているのが次のようなものである。

1. ソフトウェアに対しては要求仕様を定義するのを避ける。たとえば米国連邦航空管理顧問会議の出している回状では、航空機の種々の構成要素の重大故障発生確率を飛行時間当たり 10^{-9} とすることを取り決めているが、この対象からソフトウェアは定量的なエラー評価が不可能だとして明示的に除外されているという。
2. システム内でソフトウェアの果たす役割をあまり重大なものでないようにする。
3. 現状のソフトウェアの信頼性の限界を認めて、システム全体の安全性のレベルが多少甘くなるのを許容する。

さらにその 2 年後の Scientific American 誌（1994 年 9 月号）では、W. Gibbs という科学記者による「ソフトウェアの慢性的危機状況」という記事があり、デンバー空港事件を始めとする多くのソフトウェア開発の失敗プロジェクトを例にあげて、危機状況は慢性化しているということを、ややセンセーショナルに報じている。ソフトウェア業界で元気がいいのは、極言すれば米国でも W. Gates だけかもしれないのである。

4 ソフトウェア工学は何をしてきたか

ここで、ソフトウェア工学の状況に目を転じてみよう。よく知られるように、ソフトウェア工学は 1960 年代の終わりに興り、70 年代は構造化プログラミングという主導概念のもとに、理論面と実践面の相補的關係がかなりよく機能して、活発な研究、実践活動を展開した。

80 年代に入ると、ソフトウェアの大規模化、複雑化に対処しなければならないという掛け声とともに、ソフトウェア工学の重心が管理技術の方に移行していった。70 年代のソフトウェア工学の枠内で進められた、段階的詳細化などのプログラミング方法論、モジュール化、抽象データ型、などの分野は、次第にソフトウェア工学から外れていった。そのことで思い出すのが、1987 年にカリフォルニア州モンタレイで開かれた第 9 回 ICSE である。この時の基調講演者の一人が Lee Osterweil で、その後のソフトウェアプロセス研究隆盛のきっかけとなった「ソフトウェアプロセスもソフトウェアである」という題で講演をした。この講演は、後に与えた影響の大きさから、ICSE が毎年行っている 10 年前に開催された同会議の論文から選ばれる最優秀論文として、1997 年の ICSE で表彰を受けている。さて 10 年前の当日の講演後、第一に質問したのが Harlan Mills で、まず会場にチューリング賞の授賞者がいたら手を挙げてくれといい、一つの手も挙がらないことを確認した後、「Dijkstra, Hoare 達はどこへ行ってしまったのか。ソフトウェア工学の分野で、われわれはもう彼らから学ぶことはないのか。」と発言した。IBM の中にあっては理論派であった当時の Mills としては、管理技術に偏向して「反知性化」したソフトウェア工学の状況に我慢がならないという感じであった。

80年代のソフトウェア工学は、プログラミング言語やプログラミング方法論に関するテーマを、すでに解決済みのものとするか、あるいは小規模ソフトウェアにのみ通用する技術で、大規模なものには役立たないとして、意識的に遠ざけた。これは発展段階にあるあらゆる分野で起こる専門分化の現象の一つであったかもしれない。実際、たとえばACMではSIGSOFTが1976年にできて、SIGPLANとは路線を明確に分けることになった。しかし、この棲み分けのために、ソフトウェア工学は技術的な拠り所のかかなり大きな部分を失ったのではないだろうか。

80年代における実践的成果には、インスペクションやプロトタイピングがあり、これらはそれなりの効果をあげた。しかし、技術的な革新性という点での魅力には欠ける。新しい技術、とくにAIとの連携はやはり80年代の大きな話題だったが、結局のところそれほど目覚ましい結果を生んでいない。

80年代の後半にはソフトウェアプロセスが多くの関心と呼んだ。続いて90年代に入るとソフトウェアアーキテクチャという概念とその研究の普及に一部の研究者が力を入れた。これらは、プログラミングレベルではなく、システムレベルで、あるいは開発管理という面で、なんとか理論的なアプローチをしようとする研究者の苦肉の策と見ることもできる。ソフトウェア工学は管理であるとしてしまうと、理論の方ではやる事がなくなってしまう。そこでプロセスという管理面を含みながら技術的な対象となりうる分野とか、アーキテクチャといったマクロな設計対象を相手にして、そこに形式的なモデル化手法を持ち込み、理論的な取り扱いを試みようというわけである。プロセスの方は、CMMやISO9000のお陰で、同じ言葉を使いながら異なるものを想定している気味はあるものの、実務者の興味をもひくことにある程度成功した。しかし、アーキテクチャの方はその点が難しそうだ。

同じく80年代後半から、オブジェクト指向技術の分析や設計への適用が大いに関心を集めた。この動きは、形としては構造化プログラミングから構造化設計や構造化分析が生まれた70年代の流れと類似する。さらに、オブジェクト指向には、再利用・部品化というこれまたソフトウェア工学の長年の課題に、解決の糸口をつけるという期待もかけられた。70年代のソフトウェア工学が構造化プログラミングを指導原理として活況を呈したとすれば、90年代はオブジェクト指向技術を指導原理として同じような活気を取り戻せるかもしれないという期待を抱かせるに十分なものがあったといえよう。90年代も半ばを過ぎた現在、この期待は実現したといえるであろうか。

オブジェクト指向は確かに豊かな概念であり、現在でも分散オブジェクト、設計パターンやフレームワークといった高レベルの部品化、ビジネスオブジェクトなど、活気のあるテーマを産み出している。しかし一方で、90年代はもはや70年代のように、単一の原理ですべてをくくれるほど単純ではなくなっているともいえよう。さらに学会の細分化傾向も関連する。SIGSOFTとSIGPLANの棲み分けについて述べたが、OOPSLAを主催するのはSIGPLANである。OOPSLAは相変わらず多くの参加者を集める派手な学会となっているが、SIGSOFTとしてはよい商品を先に取られたというところであろう。意地のせい、か、棲み分け原理が働くのか、SIGSOFT主催のオブジェクト指向をテーマにした会議は目にしないし、ICSEやFSE(ACM SIGSOFT Symposium on the Foundations of Software Engineering)でオブジェクト指向をキーワードに入れたような論文はきわめて少ない。ただし、日本やヨーロッパでは、もともとソフトウェア人口が少ないし、米国の学会の暗黙の掟（そのようなものが実際に存在しているかどうか確かではないが）にとらわれる必要もないので、ソフトウェア工学関連の研究会やシンポジウムにオブジェクト指向の話が出てくるのは珍しくないどころか、むしろ目立って多い。

ソフトウェア工学が、ソフトウェア生産の現実を追いついていない面もあるかもしれない。大規模化、複雑化という言葉が、ソフトウェア工学の存在意義を説明するのに使われ続けてきたが、現実には、小規模多品種のソフトウェアの生産など、さまざま異なる性質の要求があるだろう。また、ソフトウェア工学が想定していた対象システムは、米国ならたとえば防衛関連の巨大システム、日本ならたとえば銀行システムのような民間の巨大システムで、いずれにせよ特定の顧客向けの受注型システムであった。たとえばシュリンクラップ型のソフトウェアをどう作ったらよいかという問題は、旧来のソフトウェア工学が真正面から取り組んでこなかったものいえるだろう。

その点で、マイクロソフトのお膝元たるシアトルで開かれた1995年のICSEはある意味では象徴的であった。当初はビル・ゲイツを基調講演者という話もあったが、最終的にはマイクロソフトの関係者の発表

や展示はほとんどなかった。そのかわり基調講演の一人が M. Cusumano で、その題目がなんとマイクロソフトであった。5 年前は日本のソフトウェア工場に目をつけ、今度はマイクロソフトというのはビジネススクールの教師らしく目端がきくというべきか。しかし、マイクロソフトほど一般のマスコミで取材されている対象に対して、新たな情報や知見を披瀝するのはクスmanoにしても荷が重い。話としてはつまらなかった。聴衆の反応も冷ややかだったが、印象に残った質問が一つある。質問者は大学の人らしく、「クスmano氏の話によれば、マイクロソフトでは大学で一生懸命やってはいけないと教えていることを常にやっているのではないか」というものである。聴衆が少し湧いたが、マイクロソフトの技術や商売のやり方を批判したい気持ちはよく分かる。とくに情報として目新しいものはないが、マイクロソフトを持ち上げる内容の講演の後であってみれば。しかし一方で、筆者には変な質問のようにも思えた。そもそもクスmanoはほとんど技術の話をしていないので、ソフトウェア工学上好ましからざることをマイクロソフトがやっているかどうか判断できるような内容はなかったと思う。だから、この質問は講演内容によらず質問者が先入観念として持っているものに違いない、というのがおかしく感じた理由の一つである。同調者も同様の発想であろう。また、この質問者の意図は、大学で正しく教えているソフトウェア工学からすると、マイクロソフトの開発プロセスは劣悪であるという主張であるが、皮肉にとれば、大学で教えているようなやり方ではソフトウェア企業としては成功しない、ともいえないことはない。質問者とその同調者がそれに気づかないのもおかしく思った。

5 これからのソフトウェア工学

ここを書く予定であった。実際書こうとしているのだが、理由があって少し先になる。そこで、最初にも述べたように、狡猾な戦略に出ることにした。このままで SEAMAIL に出し、幸い何らかのフィードバックが得られれば、これから書くものに参考にしようというのである。というわけで尻切れとんぼのようだが、ご意見を賜りたい。