

プログラミングの現場とソフトウェア工学

山崎 利治 (日本ユニバコ)

1 ソフトウェア工学とソフトウェア管理工学

ソフトウェア工学として、純粋技術的側面、すなわち、厳密科学の枠組における展開と、管理的側面からの諸議論、いわゆるソフトウェア管理工学が見受けられる。

Raymond Yeh [8] や David Gries [3] は前者であり、Barry Boehm [2] や Richard Gunther [4] などは後者といえる。Maurin Zelkowitz [9] は明快に二分してソフトウェア工学を紹介・展望している。ここで、ややややプログラミングの現場は、ソフトウェア工学からなにも学び得ぬことをどう許すかについて考えてみたい。いま学ぶべき多くは上の分類の前者にあり、その範疇 (paradigm) の形成のための現場の貢献を注意したい。

2 プログラミングの現場

さて、プログラミングの現場——事務計算、技術計算、基本ソフトウェアなどそれぞれのプログラミングの作成現場——ここでのスタッフとその仕事について振り返ってみたい。この産業側スタッフは、^{しばしば} しばしばその精神状況は、ホワイト・カラーに属する。だから、ときとして「教育」が議論されるが、プログラミングがそのための教育が熱心に行われることはない。ホワイト・カラーの基本能力のほかに、プログラミングの^{その} できが期待でき、また可能であるとす。ちょうど商業文を書く能力、事務に必要な計算能力あり、計算機なしに行なう技術計算能力ありと同一視するわけである。ここではしばしば「よいソフトウェア技術者である以前に、よい企業人である」といわれる。ホワイト・カラーであることはまた自然管理_に

向にもなる。他方また、いまなお現場にのこる——流小図を元のままコード化する作業——「コーディング」という確かにブルー・カラー制というより非経歴労働といふこともよいものがある（＝＝で書かれたコードを読んで整くことも多いが——＝＝でのプログラムは徒弟制の取人にはなっていない、すなわち、取人の技能に責任をもつ親方が存在である）。

計算機利用者が1000文程度のコードを書くとはい個別にみればさほど問題にはなってもよいかも知れない。このような計算機プログラムは、日記や私的ノートあるいは計算帖のようなものである。しかし計算帖を見直すと、反復使用するとき、さらにその計算帖を他人も使用し公的にも用いられるとき、プログラミングそのものに大きな問題が生ずる。いくつかのコードを融合利用するとき、計算規模が大きくなるとき、そのソフトウェアづくりそのもの問題は深刻になる。総型計数コードや構造解析コードなどの場合でも、制約式や変数、または自由度が大きくなると素人ではプログラムできなくなる。応用分野や数値計算に際する専門知識と経験以外に、ソフトウェアづくり自体に係るはずかこそ「経験」こそが発生したわけである。すなわち、応用分野ではなく、計算技術でもない、ソフトウェアそのものについての独立した技術分野が存在するのである。

3 ソフトウェア工学から学ぶもの

以上のような知見の上で、ソフトウェア工学＝プログラミング方法論の存在理由がよくわかるわけで、こゝまじわりの産業の現場が学ぶべきものである。かわりかわり仲間とともに、まず、Niklaus Wirthの二冊の教科書[6][7]を学び、Donald Knuth[5]やAho-Hopcroft-Ullman[1]を読み、さらにさきのGries[3]やYeh[8]などを実際に適用してゆけばならない。

4 ソフトウェア工学の確立のために

ソフトウェア危機の克服のためのソフトウェア工学は、ソフトウェア管理工学と峻別して存すべきである。船舶工学は造船管理工学のまゝに確立されねばならなかった。排気量計算が確立していなかったときに経済的な建造法も議論してはならない。正しいプロシージャは経済的につくることが正しくつくねばならない。ソフトウェア工学はソフトウェア管理工学のまゝにその範疇も確立しなねばならない。この範疇の形成はソフトウェア技術者を生むはずである。

この工学の確立のためにはまず教科書が欲しい。現場は、たとえば、抽象化の型も実際に応用すべきである。HIPOを使用するにせよよい。HIPOは抽象化も強制する図を用いる意味で極めて有効である(Dijkstraの指摘をまっぴらなく皮相的な見方が一般的であるか——この辺は科学に対する技術の態度はなるといふ言葉をももって聞いて面白い)。Parsonsの例を明記してあげては駄目で設計仕様もこの流儀で書いておねばならない。華納計算プロシージャに対する玉井・福永のアプローチ□□にみられるような仕事を発展させる必要がある。このような仕事が増積されてはじめてよい教科書が書けるわけだ。現場の改訂版には学術側も十分な注意を払うに違いない。応用の発展は現場のつとめであり、それが技術の進歩も促し、理論の深化をもたらす。そうしてソフトウェア工学の範疇が広まり、その存在自身がプロシージャの現場に与える影響もよほしはじめ、この理論や技術の教育も必須にし、職業性を生み、作成されるソフトウェアを通じて新しい型を創り影響を拡大する。この工学の有効性の認識とともに自然にソフトウェア管理工学と同期することになる。

5 データベース設計論

「ソフトウェア工学に欲しいものはデータベース設計論である。データベース意味論の展開が不可避であるが、現場でもっとも指針の欲しいもの一つである。ソフトウェア工学の育みに現場は大きな責任をもつことも意識すれば、課題もどんどん提提されてくるであろう。

参考文献

- [1] Aho, A.V., Hopcroft, J.E., Ullman, J.D., The design and analysis of computer algorithms. Addison-Wesley, 1974.
(邦訳) 野村浩・野下訳, アルゴリズムの設計と解析 I, II, サイエンス社. 1977.
- [2] Boehm, B.W., Software engineering, IEEE Trans. on Computers, C-25, 12. (1976) 1226-1241.
- [3] Gries, D., Programming methodology. Springer, 1978.
- [4] Gunther, R.C., Management methodology for software product engineering, John Wiley, 1978.
- [5] Knuth, D.E., The art of computer programming, 3 vols. Addison-Wesley. 1968.
(邦訳) 花沢敬也訳. 基本書局, 2 vols. サイエンス社 1978.
- [6] Wirth, N., Systematic programming, an introduction, Prentice-Hall, 1973.
(邦訳) 野下 訳. 系統的プログラミング. 近代科学社, 197

- [7] — , Algorithms + data structures = programs.
Prentice-Hall, 1976.
- [8] Yeh, R., Current trends in programming methodology.
4 Vols. Prentice-Hall, 1978.
- [9] Zelkowitz, M. V., Perspectives on software engineering,
ACM Computing surveys, 10, 2, (1978) 197-216.
- [10] 玉井・裕紀, 事務処理分野における検証適用のための仕様
記述の試み. TSC.Symp. on intelligent programming
systems, Tokyo Scientific Center, IBM Japan, 1978.