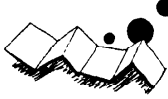


## 解説



## ソフトウェアからソフトウェア製品へ†

國井利泰††

## 1. プログラム, ソフトウェア, ソフトウェア製品 一序にかえて—

ソフトウェア製品あるいはパッケージド・ソフトウェアという言葉が人々に受け入れられるようになったのもそう昔のことではない。日本ではむしろまだソフトウェア製品というよりは単にソフトウェアと呼ぶか、あるいはその一部であるプログラムといった方がなじみやすいかもしれない。しかし、情報産業が一つの社会的に大きな比重を占めるものとなるにつれて、ソフトウェアも市場に製品として流通するソフトウェア製品としての形を整えざるを得なくなった<sup>1)</sup>。それではどこにソフトウェア製品とソフトウェア、あるいはその一部であるプログラムの差異があるのだろうか。

ソフトウェア製品はまずその作られ方がソフトウェアとは異なるとされる<sup>2)</sup>。ソフトウェアはプログラム・プロジェクト・チームにより作られ、同時にそれを後から保守するための内部仕様書、外部仕様書、保守マニュアルが作られる<sup>3)</sup>。これに対してソフトウェア製品は製品全体の開発に責任を持つただ一人のチーフ・プログラマの下に、プログラムを作り出すコーダ5人位が組み、これにそのドキュメントの作成、特にオンラインドキュメントの作成を支援するプログラム・ライブラリアン2人程が付いた合計8人程度のチームで編成される部隊が1つのユニットとなり製造にあたりとされる<sup>4)</sup>、<sup>10)</sup>、<sup>11)</sup>。この分業方式のこれまでの非分業方式に対する優位性と問題点とは、工業一般におけるこれまでの歴史から明らかな通りである。分業方式の優位性の一つである高生産性を保ちつつ、その欠点である単純作業の繰返しからの脱却を可能にするために、繰返し作業を分担あるいは代行するさまざまな対話型ならびに自動化ソフトウェア工具群が開発され、使用さ

れるようになったのも、ソフトウェアを製品として生産する場合に特徴的なことである。これも現代工業における生産オートメーションの進展と軌を一にする。たとえばソフトウェア製品には改訂が必要となった時点でただちに更新され得るオンライン・ドキュメント類、ユーザが製品を使用する際に予備知識無しに端末を通して対話的に利用できるインタラクティブ・ユーザマニュアル、ならびに製作チームとは異なる保守部隊が製品の保守を自動的に行うことができるように製品自体に組み込まれた組み込み保守機能が備わっているのが普通になりつつある。したがって、ソフトウェア製品は製作チームの手を離れて独り立ちできる商品としての資格を備えるに至った。

しかし日本の場合には、このようなソフトウェア製品が現在のコンピュータ産業の中での主流を占めるような製品となるまでにははまだまだかなりの紆余曲折を覚悟しなければならない。なぜならば、オンライン・ドキュメントの手法の完全普及には、まだ人間学的に優れた漢字端末の開発等さまざまな理由で障害がある<sup>4)</sup>。オンライン端末の普及自体までもう一步というところである。更に設計時、あるいは開発時に保守機能までも考慮するようなソフトウェア製品のこのような製造法は、初期投資コストが高いために、依然として敬遠されている。主として現在ハードウェアを対象に支払われているコンピュータ会社、あるいは情報関連会社への支払いがソフトウェア中心のものに切り替わり、その前提としてのソフトウェア有償化が進まない限り、しょせんこのようなソフトウェア製品の大規模普及は絵に書いた餅でしかないであろう。しかし、ソフトウェアの輸出等今後の日本の情報産業の発展、特に国際的発展を考えるならば、ソフトウェア製品に対する本格的取り組みは避けて通れないどころか、むしろ中心の問題でさえある。現時点でも、ソフトウェア会社はもとより、コンピュータメーカーでさえその製品コストの8割以上を占めるソフトウェアのコスト高により重圧を受けているのが実情である。

† Software Products Engineering—Evolution of Software—  
by Toshiyasu L. KUNII (Department of Information  
Science, Faculty of Science, The University of Tokyo).

†† 東京大学理学部情報科学科

## 2. ソフトウェア製品に至る歴史

さて、今まで述べたようなソフトウェアの発展段階をもう一度整理して、現在のソフトウェア製品に至るまでの歴史を簡単に振り返ってみよう<sup>5)</sup>。

まず、ソフトウェアの歴史は1950年前後に始まったプログラミングの時代で幕を開けたとされる<sup>16)</sup>。それは1962年まで続き、有能なるプログラマ個人がきわめて便利な個人用のプログラムの開発に腕を競った。この時代のソフトウェアの特色は、あくまでプログラムは個人限りもしくはその周辺限りの使い捨てのものであり、そこで主として用いられた技術は個人の名人芸、あるいは口伝えであった。

このいわば幸せなプログラミングの時代を過ぎて、1963年頃から1970年頃に至るソフトウェア開発の時代が到来した。この時代にはソフトウェアの社会的需要が増し、コンピュータ全体における経費的比重も増して、もはやその場限りの使い捨てのソフトウェアの生産が経費上も運営上も許されなくなった。それぞれの応用上の目的に合わせて強力なプロジェクト・チームが生まれ、同時にこのようにして生産されるソフトウェアの利用者による円滑な利用を助けるための数々のユーザマニュアルが作り出され、また手直しや保守のための内部仕様書が集積されるようになった。そこにおいてはプロジェクトごとに作られたプロジェクト・チームはかなりいわば均質な人間から成り立っており、なるべく有能なチーム編成を考えるにしても、そのチームの各人は設計、コーディング、マニュアル作成まで、それぞれこのチーム内の自分の分担に従ってほぼ均等に分担した。したがって、応用上のソフトウェア機能の分担に応じて各人がソフトウェア製造過

程の初期段階の計画から設計、製造、場合によっては保守までのソフトウェア・ライフサイクルの全般を幅広く分担したのが特色である。

ほぼ1971年頃からこのようなソフトウェアの生産法はもはやコスト的にも、あるいは利用者の納期上の要求に答えるためにも不十分なものとなってきた<sup>18)-20)</sup>、均質の数人が1つのプロジェクトを担当する代りに、数々のトレーニングを経て設計に対して十分な見通しを持ったリーダーがチーフ・プログラマの名でたてられ、コーディングの達人がそれを受けてプログラムの生産に専念し、オンライン・ドキュメント作成の専門家がプログラム・ライブラリアンとして利用者用、保守用あるいは教育用として必要となる多種多様のドキュメントをオンライン・ファイルにたたき込む生産方式がとられるようになった。これは前にも指摘した通り、既存のソフトウェア産業以外の工業のたどった道でもあり、いわば分業化が徹底的に進む生産の歴史の必然的発展の経路、もしくは結末であるともいえよう。それを支える理念もこれまでのプログラミングアート<sup>6)</sup>からソフトウェア工学、あるいはソフトウェアエンジニアリングといわれるものに発展を遂げた<sup>7), 20)</sup>。これまで名人だけが享受してきたモジュラ構成法、または階層的モジュラ構成法などの手法がソフトウェア産業の末端にまで普及した。当然ソフトウェアにおいても設計図が書かれ、設計図の基本となる構造あるいはその単位も明らかにされ、ソフトウェアの構造化設計法が業界の基本技術として普及するに至った<sup>8)</sup>。

## 3. ソフトウェア製品生産自動化における工具の位置

以上述べたようなソフトウェア生産方式の発展形態

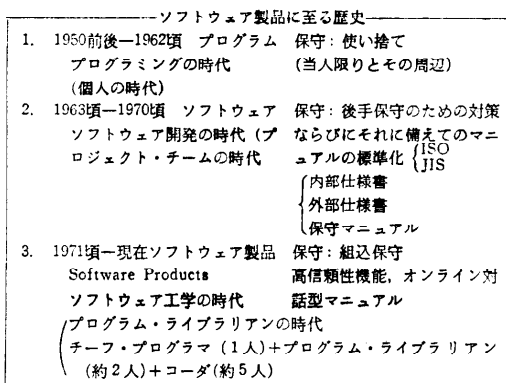


図-1

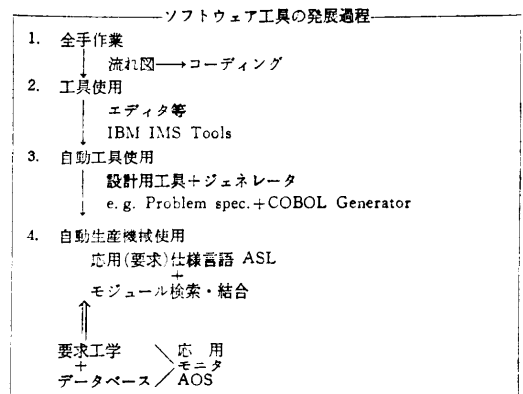


図-2

の変化を支えるために、当然そこで用いられる工具類も大きな変貌と発展を遂げた<sup>9)</sup>。プログラミングの時代には主として全手作業用の流れ図が頻繁に使われ、流れ図からコーディングというのが標準的な工法であったといえよう<sup>\*</sup>。その時代からソフトウェアの時代にかけてエディタあるいは数々の言語トランスレータ等のソフトウェア工具も使われた。

しかし、ソフトウェアの時代にはプログラミングは全体の一部分にすぎず、大量のソフトウェアを生産するためにオンライン・エディタ等の工具とかHIPOのような文書化ツールが多用されるようになり、デバッグのためのデバッグ工具類も組織的に整備されるようになった。

ソフトウェア製品の時代に入ると設計仕様言語でソフトウェアの設計を書き、ジェネレータでソフトウェアの生産をするというジェネレータ方式が、たとえばコンパイラ・ジェネレータ等のようにしばしば使われるようになった<sup>9)</sup>。更には進んで客の要求を書く要求仕様言語が発達し、それを入力として今まで蓄積されたプログラム・モジュールあるいはデータ・モジュールを自動的に結合・評価し、客の要求に答えるソフトウェアを自動的に生産しようとする試みが重点的に行われるようになった。現在このような要求仕様技術やそのための要求仕様言語は要求工学として徐々に発展しつつある<sup>10)</sup>。その一方のソフトウェア資源の共有化のためのプログラムあるいはデータのモジュール化の技術は、現在きわめて大きな努力が払われているにもかかわらず、今一つその基本となるところが明確でないきらいがある。しかし、データ定義とその参照のブロックの組み合わせがいかなる構造もしくは階層でつながるかを明確に仕様化するデータ・フローアナリシスの技術、ならびにその制御形態を明らかにするコントロール・フローアナリシスの技術により、データベース技術の発展とあわせて、現在このモジュール化技術もしだいにその姿を明確化しつつあるといっても良いであろう<sup>11)</sup>。設計評価のためのシミュレーション技術も、対話型設計技術との組み合わせで対話型シミュレーションの方向に進歩しつつある。

#### 4. ソフトウェア製品への要求

ソフトウェア製品はそれでは主としていかなる社会

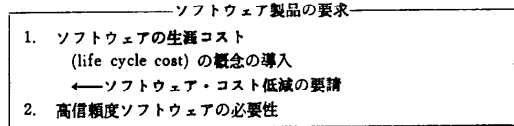


図-3

的要求に応じるために生まれているのであろうか。種類の要求を分析しまとめてみると、そこには大きな2つの要求があるように思われる。その第一はソフトウェアの生涯コスト (life cycle cost) の概念の導入によるものである。ある1つのソフトウェアに注目したときに、それに対する要求が発生することにより誕生してからよりすぐれた製品にとってかわられて不要になるまでの、ライフサイクル総コスト低減の要求である。ソフトウェアを単に作れば済むものとする時代は終わった。ソフトウェアを個々の顧客の要求を越えて流通させるもの、との考えにたつて製造部隊、販売部隊、保守部隊を完全に分け、その全体を管理するための管理層を置く、いわゆるソフトウェア会社が社会的要求に応じて多数生まれるようになったのも、このようなライフサイクル管理の分業化によるコストダウンの動きの反映であるということができよう<sup>9)</sup>。今一つは社会的にソフトウェアの占める比重、あるいは場面が増し、そのために高信頼度ソフトウェアが要求されるようになったということが挙げられる。たとえば列車の運行、売上げの集計、給与の支払い、あるいは原子力発電所等のエネルギー機関の運転等を始めとするありとあらゆる社会の場面でソフトウェアが中心的役割を果たすにつれ、社会機能の維持・保全のためにソフトウェアの高信頼度化が当然要求されるようになった。そのためのソフトウェア製品工学がきわめて大きな発展を遂げ、ソフトウェア品質保障の技術が信頼度向上を目標として着実に進歩を遂げてきた<sup>12)</sup>。現在のところ、このような高信頼度技術は主としてアメリカの軍用ソフトを中心に発展したといわれているが、航空機と同じようにもはやこれらの信頼度の技術は軍用製品の域を離れて完全に民需の世界でもコスト的にひきあう、あるいは当然要求されるべきものとなりつつある<sup>13)</sup>。我が国ではこれらの信頼度向上の問題は今からむしろ焦点になりつつあると考えて良いであろうが、そのための数々の方法、道具、あるいはその背景となる方法論 (methodologies) についても大きな発展が必要となっている。

<sup>\*</sup> Parnas によると、5,000~10,000 以上の命令数のプログラムの場合には、流れ図はもはやプログラム仕様として有効な工具たり得ないという<sup>14)</sup>。

## 5. おわりに

世界第2のコンピュータ設置数を有する日本の立場からして、現時点でソフトウェア製品に関する本格的開発体制の整備が急務になっている。本論はその第一歩として、ソフトウェア製品の備えるべき資質とはいかなるものかを明らかにしようと努めた。最後に、ソフトウェア製品の問題はもはやコンピュータ・メーカーあるいはソフトウェア会社にとって問題であるだけでなく、コンピュータの社会普及度とソフトウェア比重の急激な増大に伴い、国家あるいは社会全体の政策上の問題となっていることを指摘して本論の終わりとす。

## 参考文献

- 1) たとえば Datamation 誌に毎年連載の Datapro 社主催ソフトウェア・パッケージの利用者による品評会を参照。
- 2) Metzger, P. W.: *Managing a Programming Project*, Prentice-Hall (1973).
- 3) Brooks, Jr. F. P.: *The Mythical Man-Month—Essays on Software Engineering*, Addison-Wesley Pub. (1975).
- 4) Yamada, H.: *A Historical Study of Typewriters and Typing Methods: from the Position to Planning Japanese Parallels*, *Journal of Information Processing*, Vol. 2, No. 4, pp. 175-202 (1980).
- 5) Gunther, R. C.: *Management Methodology for Software Product Engineering*, Wiley-Interscience Pub. (1978). 日本語版, 水野幸男監訳。
- 6) Kunth, D. E.: *The Art of Computer Programming (Second Edition)*, Vol. 1-3, Addison-Wesley Series in Computer Science and Information Processing (1973).
- 7) 情報処理学会ソフトウェア工学研究会報告シリーズ. *IEEE Transactions on Software Engineering*, *ACM SIGSOFT Newsletters*.
- 8) Yourdon, E.: *Managing the Structured Techniques (Second Edition)*, Prentice-Hall (1979).
- 9) 國井利泰: ソフトウェア・ツールとその位置付け, *情報処理*, Vol. 20, No. 6, pp. 477-480 (1979).
- 10) 國井利泰監修: *ソフトウェア工学—要求仕様技術—*, 共立出版 (1978).
- 11) Kunii, T. L. and Harada, M.: *SID: A System for Interactive Design*, *Proc. AFIPS National Computer Conference*, pp. 33-40 (1980).
- 12) Razouk, R. R., Vernon, M. and Estrin, G.: *Evaluation Methods in SARA—The Graph Model Simulator*, *Proc. 1979 Conference on Simulation, Measurement and Modeling of Computer Systems*, pp. 189-206 (1979).
- 13) *Proceedings of Specifications of Reliable Software*, IEEE Computer Society (1979).
- 14) 國井利泰: 「生命型」製品による産業構造の転換, 昭和54年電気四学会連合大会, pp. 5-1~5-4 (1979).
- 15) Parnas, D. L.: *On the Criteria to be Used in Decomposing Systems into Modules*, *CACM*, Vol. 15, No. 12, pp. 1053-1058, (1972).
- 16) たとえば Goldstein, H. H. and von Neumann, J.: *Planning and Coding Problems for an Electronic Computing Instrument*, Part II, Vol. 1, report prepared for the U. S. Army Ordinance Department (1947).
- 17) たとえば Brandon, D. H.: *Management Standards for Data Processing*, Van Nostrand Reinhold (1963).
- 18) Mills, H.: *Chief Programming Teams, Principles and Procedures*, IBM FSD Report, FSC 71-5108, Gaithersburg, Maryland (1971).
- 19) Baker, F. T.: *Chief Programmer Team Management of Production Programming*, *IBM Systems Journal*, Vol. II, No. 1, pp. 56-73 (1972).
- 20) Landy, B. and Needham, R. M.: *Software Engineering Techniques Used in the Development of the Cambridge Multiple Access System: Software—Practice and Experiences—*, Vol. 1, No. 2, pp. 167-173.

(昭和55年6月30日受付)