ソフトウェア工学 73−11
（1990. 7. 20）

ソフトウェア・プロセスの設計教育用ツールへの適用及び評価

望月 純夫　山内 顕　市村 英昭　片山 卓也
三菱スペース・ソフトウエア（株）　東京工業大学

若手技術者のためのソフトウェア設計教育用ツール開発を目標として、実際のシステムの設計手順を解析し、ソフトウェア・プロセスを抽出した。しかし、このソフトウェア・プロセスをそのまま再利用して、ほかの設計問題に適用してみると種々の問題点があることが判明した。そこで、ソフトウェア・プロセスを再度整理し、設計内容を中心とした表現にしてほかの技術者にも理解し易く、また、応用範囲の広いものに改善した。今後、このソフトウェア・プロセスを中心としたＳＥ教育用ツールを開発し、充実させるつもりである。

SIG SE 73-11

APPLYING THE SOFTWARE PROCESS TO THE INSTRUCTION TOOL IN SYSTEM DESIGN AND EVALUATING IT

Sumio MOCHIZUKI　Akira YAMAUCHI　Hideaki ICHIMURA
Mitsubisi Space Software Corp,
MSS Yamazaki Bldg. 299, Yamazaki,
Kmakura-shi, 247, Japan.

Takuya KATAYAMA
Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo, 152, Japan.

To develop an instruction tool for inexperienced engineers, we chose one of the real-time processing systems to analyse its design process, and extracted the software process.

However, when we tried to reuse this software process to another design theme, we found some difficulties in it. We therefore rearranged the software process to express design process more clearly, and made it more understandable and applicable. We will develop a new instruction tool for the young system engineers, using these results.

1．はじめに

最近の情報処理産業の発展にはめざましいものがあり、ソフトウェア開発量の拡大は著しく、ソフトウェア開発を主業務とする企業は年々新しい優秀な技術者の確保に最大の努力を続けている。三菱スペース・ソフトウエアもその一つであり、事業の拡大に伴い、若手技術者の確保並びにその早期育成が大きな課題となってきている。

これまで、ソフトウェア設計技術は技術者各個人に属しており、その伝承は実際の開発業務を利用したＯＪＴしかないとされてきたが、若年層の割合が多くなると、そのＯＪＴの対象としてふさわしい業務がいつも必要な時にタイミングよくあるわけではなく、またＯＪＴを指導する熟練技術者の確保も容易ではない。そこで、何等かの設計技術教育用ツールを開発し、その教育効率を高めることが必要となってきた。設計教育においては、単なる技術的な知識を身につけるだけでは不十分であり、それを活かした訓練が不可欠である。

このような教育を考える場合、まずソフトウェア設計手順を明らかにする必要がある。我々は、プロセス・モデルＨＦＳＰを利用してソフトウェア設計の手順を明確にし、これを利用して若手技術者の教育訓練に適用しようとしているのである。[2]

2．ソフトウェア・プロセスの分析及び評価

2．1実用システムの設計プロセスの分析

教育用ツールとしてふさわしい優れたソフトウェア・プロセスを求めるためには、洗練された設計手順の基に設計されたシステムを選択し、その設計過程を分析する必要がある。我々は、これまで開発してきたシステムの内、人工衛星チェックアウト・システム（以下　”システムＡ”と呼称する）を選択し、その基本設計フェイズにおける設計手順を分析することとした。

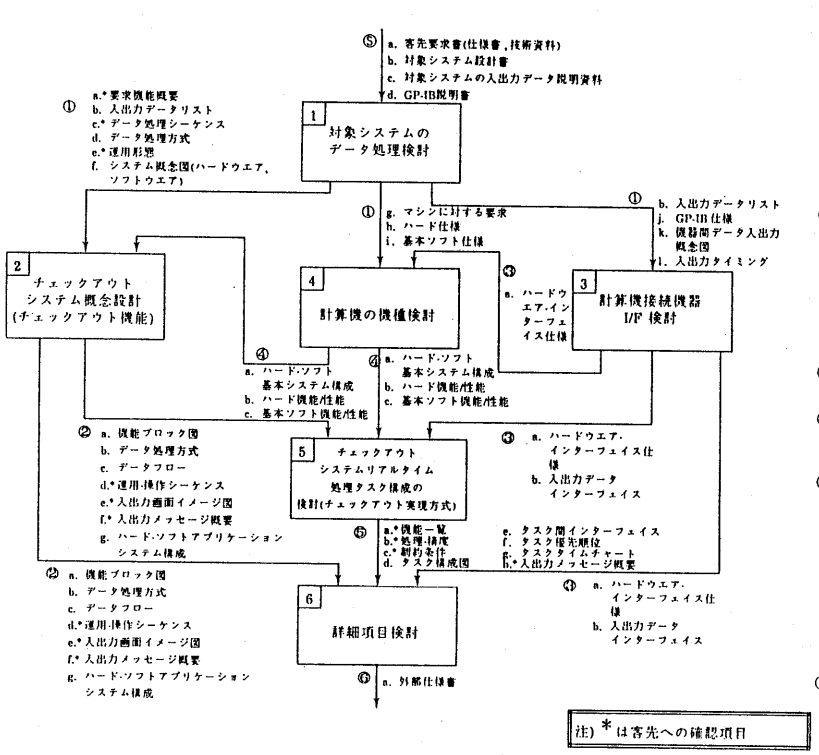

図１　人工衛星チェックアウト・システムの基本設計段階のソフトウェア・プロセス

まず、同システムの設計経験を持つ熟練ＳＥ６名により、過去の設計手順をプロセス・モデルＨＦＳＰに基づいて記述した。その結果、図１に示すようなソフトウェア・プロセスを得た。〔5〕

この内容については、分析作業に参加したＳＥの設計手順を十分に分析、吟味し、最終的に全員納得できるものを得ることが出来た。

２．２ソフトウェア・プロセスの評価実験及びその結果

前項で得たソフトウェア・プロセスを評価するために、別のＳＥ（二人）によりこのソフトウェア・プロセスを利用して、簡単な計測制御システム（以下”システムＢ”と呼称する。）を実験的に設計させた。システムＢは、計測制御システムの要件を含んだモデル・システムであり、対象物の温度を数点計測し、各点の温度を目標値に近づけるべくヒータのＯＮ／ＯＦＦ制御をするものである。

システムＢの概念図を図２に示し、前項のシステムＡとこのシステムＢの比較を表１に示す。

モデル・システムの設計に対しては、従来の我々固有の設計手法をそのまま利用したが、オブジェクト形式は、出来る限りＳＡ／ＳＤ技法の表現を利用することとした。

表１　ソフトウェア・プロセス分析対象システムの比較

システム	前回（システムＡ）	今回（システムＢ）
システム名	人工衛星チェックアウトシステム	計測データ収集・制御システム
機能概要	人工衛星のデータ収集解析（チェックアウト）	対象物からの計測データの収集・処理・制御
入力データ、点数	テレメトリー・データ ＧＰ－ＩＢ経由、多い	直接計測 ＧＰ－ＩＢ経由、少ない
データ処理内容	物理量への変換	温度制御
出力データ、点数	コマンド・データ ＧＰ－ＩＢ経由	直接制御 ＧＰ－ＩＢ経由

この評価実験の結果、図１のソフトウェア・プロセスに対し、次のような評価を得た。

（１）図１のソフトウェア・プロセスは大筋に於て正しい。

（２）ソフトウェア設計の手　　順は、正解が幾つもあり、あまり細かい部分まで一義的に定めるのは、適切でない。ソフトウェア・プロセスのレベルは、図１の程度にして、更に詳細なものは、学習者が解を得られなかった時に参照することとする。

（３）図１のプロセスは、実際のシステム開発業務の時間的な経緯を追って忠実に再現したものであり、実態をそのまま示している。しかし、そのためにいくつかのソフトウェア・プロセスの中に同じ（検討）項目が何度も登場して、解りにくい。時間的な推移を余り意識せず、ソフトウェア・プロセスの内容（設計内容）中心に記述すべきではないか。（新しいソフトウェア・プロセスの表現が必要）

（４）ソフトウェア・プロセスが繁雑なために、手戻りが発生したときどこに戻れば良いか判断しにくい。

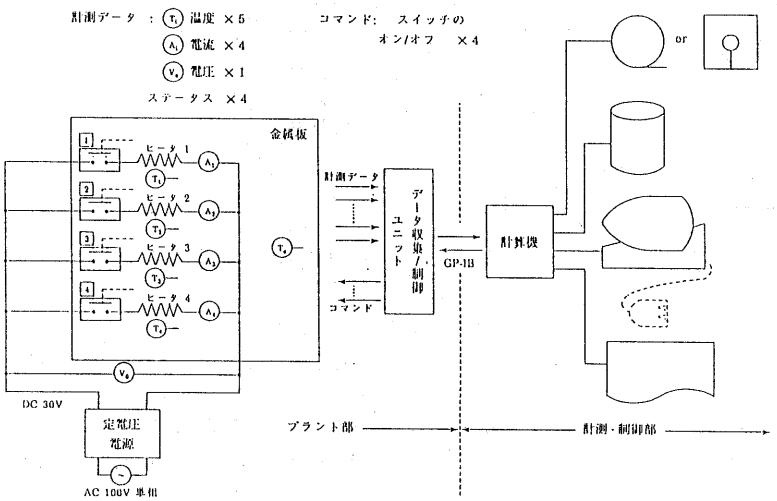

図２　計測制御モデル・システムの概念図

（５）オブジェクトも余りに細かいものが多いために、互いの関連性及び、そのオブジェクトの必要性、重要性がわかりにくい。

（６）システムＢのように開発すべきソフトウェアの規模が小さい場合はソフトウェア・プロセスも単純だが、システムＡのようにある程度規模が大きくなると、ソフトウェア・プロセスの内、いくつかを並行して進める必要がある。（ソフトウェアの規模によりソフトウェア・プロセスも異なる。）

（７）設計手順を良く整理して、分かりやすい表現をするように工夫すると、結果として得られるソフトウェア・プロセスも比較的単純になってくる。

（８）図１の設計手順は、純粋に技術的根拠に基づいたものだけではない。例えば、計算機選定のプロセスなどは、設計技術者の過去の経験に基ずく主観的な要素、技術的な好みとか興味などが支配的な要素となる場合があるが、ソフトウェア・プロセスとしては、記述しにくく、また、この経験がそのままほかのプロジェクトにも有効であるとはいいがたい。このようなプロセスは、ソフトウェア設計プロセス以外のレイヤ（例えば、プロジェクト管理レイヤ等）として扱うべきである。（図１の第３、４ブロック）

（９）設計開始時点は、あらゆる情報が不完全であり、それを取り扱うソフトウェア・プロセスが表現しにくい。

２．３新しいソフトウェア・プロセスの表現

前項の実験の結果を基にして、再度全員で新しいソフトウェア・プロセスの表現方法を検討した。その結果を次に示す。

ソフトウェア・プロセスは、設計内容を明確

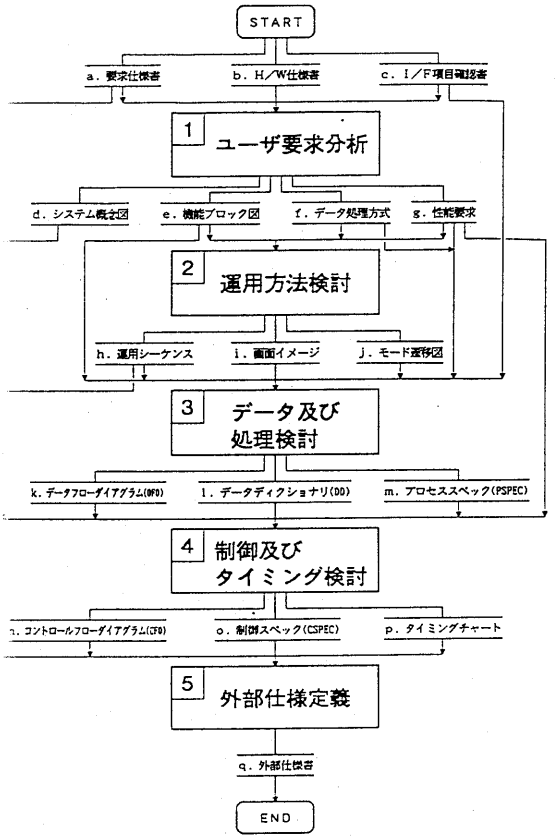

図３　新しいソフトウェア・プロセス
レベル１（トップレベル）

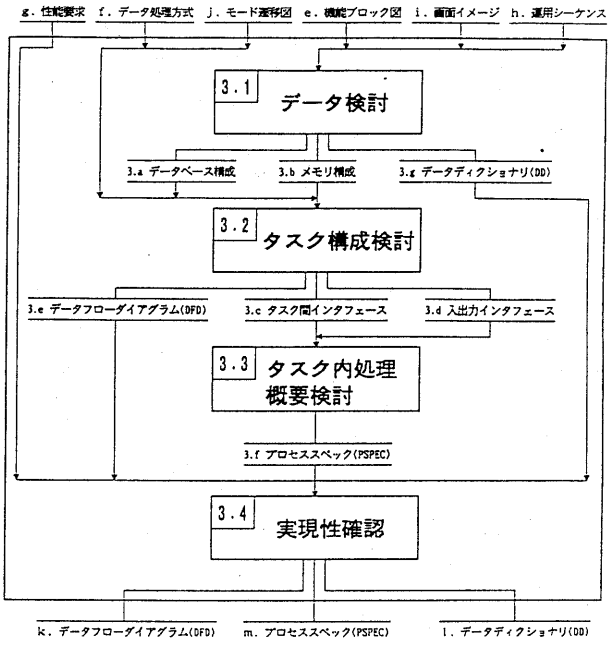

図４　新しいソフトウェア・プロセス
レベル２（図３の第３項目に対応）

に表現することに重点を置き、時間の経過とは切り離してまとめるべきである。この考えをつきつめて行くと、図１の中の１、２、５、６のプロセス —— 即ち、要求仕様の分析からソフトウェア設計までの手順 —— は、図３及び図４のような新しいソフトウェア・プロセスの表現となる。

この内容は、ソフトウェア設計の内容、及びその項目間のオブジェクトを中心にまとめたものであり、設計の時間的な経過に沿って厳密に表わしたものではない。設計作業は、ある時点では、複数のプロセスを同時に検討しつつそのオブジェクトを互いに交換し、部分的な調整を積み重ねながら進めている。このような並行作業の必要性は、対象プロジェクトの規模が大きくなるほど顕緒となる傾向にある。この関係を表２に示す。

即ち、小規模なプロジェクトでは、設計手順も単純で、極端に言えば、各プロセスは、ほかのプロセスとは独立に実行することが出来る。しかし、大規模プロジェクトでは、一つのプロセスを実行する場合、その前後のプロセス（特に後工程のプロセス）を同時に検討し、前後のプロセス間の一貫性を予め保っておかないと、先に進んで重大な手戻り作業の発生を招く結果となる。

図３のソフトウェア・プロセスと表２に示すプロセスの並行動作の関係は図５の様に示すことが出来よう。即ち、ソフトウェア・プロセスと時間は、２次元平面（実際には３次元空間）上に表現できる。ある時点Ｔにおける設計作業は、図中のＴ上の断面により表されてる。各プロセスの断面の大きさはその作業量の大きさを示す。プロジェクトが小規模の場合には、１時点に行うべき作業の種類は少ないが、大規模になるにしたがって、同時に並行して行うべき作業の種類、量が大きくなることがわかる。

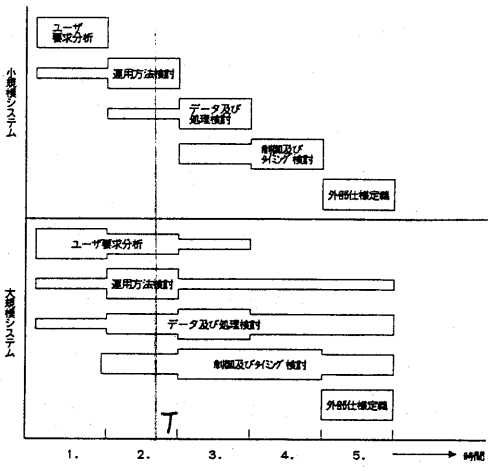

図５　ソフトウェア・プロセスと時間の経過と関係

図１のプロセスは、人工衛星チェックアウト・システムの設計手順を時間を追って忠実に表現したものであり、図５が示す並行作業を現実そのままに表していることに価値があるが、その反面ほかのメンバーには理解しにくい。また、このソフトウェア・プロセスは、対象システム（システムＡ）に密着したソフトウェア・プロ

表２　新しいソフトウェア・プロセスの各フェイズ（１－５）における検討項目の比重

（１）　小規模システム

SP ＼ 作業	ユーザ要求分析	運用方法検討	データ及び処理検討	制御及びタイミング検討	外部仕様定義
1.	大	小	－	－	－
2.	－	大	小	－	－
3.	－	－	大	中	－
4.	－	－	－	大	－
5.	－	－	－	－	大

（２）　大規模システム

SP ＼ 作業	ユーザ要求分析	運用方法検討	データ及び処理検討	制御及びタイミング検討	外部仕様定義
1.	大	中	小	－	－
2.	中	大	中	中	－
3.	小	小	大	大	－
4.	－	小	中	大	－
5.	－	小	中	中	大

セスであり他のシステム（特に規模の異なるシステム）には、そのまま適用しにくい、融通性の少ないソフトウェア・プロセスであるともいえよう。

このように、ソフトウェア・プロセス同士の並行処理の必要性は、設計対象システムの規模及び複雑さに対応して臨機応変に判断すべきものであり、ＳＥが自ら会得すべき重要な応用動作である。教育用ツールにおいては、この訓練のために難易度の異なる演習問題を用意し、学習者にソフトウェア・プロセスの相違を自ら体験させることを考えている。

２．４　教育用ツールのためのソフトウェア・プロセス

ソフトウェア設計教育用ツールのためのソフトウェア・プロセスは、次のようなものでなければならない。

（１）熟練技術者に共通な手順を若手技術者に解り易く把握させるものであること。

（２）詳細なレベルのプロセスは、熟練技術者の間でも個人差があるため、学習者が自ら考え、創造することを原則とする。但し、学習者の要求があればその一例を表示する。

（３）学習者がソフトウェア・プロセスの使い方を工夫することにより、種々の設計問題（規模及び複雑さが異なる）に対応できること。いわゆる広い応用が可能であること。

（４）オブジェクトの内容が、採用しているソフトウェア設計技術と一致すること。（図３、図４のオブジェクトは、ＳＡ／ＳＤ手法を意識している。）

このような検討結果から、我々は、図３のソフトウェア・プロセス及び表２を教育用ツールとして選択し、詳細なレベルのプロセス（例えば、図４）は、学習者の要求により表示することとした。

３．教育用ツールの開発

３．１　ＳＥ教育の問題点

ＳＥ活動の内、ユーザ要求に基づくシステム設計、開発は最も根幹の業務である。

実戦上役に立つ設計技術を育成するためには、単なる技術的知識の学習だけでは不足であり、どうしてもその応用力をつけるための訓練が必要である。

一般にシステム設計の解は、複数存在し、熟練技術者はその中から優れていると思われる手法、手順を選択し把握しているのである。この為には、長年の経験の中で意識的な努力の積み重ねが必要である。

このようにして得られた技術を伝承するために、実際のプロジェクトを対象としたＯＪＴを行ってきた。しかし、教材として適切なプロジェクトがいつも都合良く、しかも学習者の人数分あるとは限らない。そこで、我々は、小規模ではあるが、システム設計の要件を含んだ教育用モデル・システムを用いて、ＯＪＴの疑似体験をさせることとした。

３．２　ＳＥ教育用ツールの基本方針

教育用ツールを構築するに当たり、次の方針で望むこととした。

（１）基本的な設計手順の流れを理解させることとし、詳細レベルのプロセスは、学習者に考案させる。（詳細レベルの解は、要求により表示する。）

（２）複数存在する正解の中から、最も優れていると思われる設計手順を伝承する。

（３）前提となる要素技術を抽出し、事前教育の対象とする。

（４）純技術的でない判断基準に基づくプロセスは、別途プロジェクト管理などのレイヤとして扱う。

３．３　ＳＥ教育用ツールの試作

教育用ツールの基本となるソフトウェア・プロセスとしては、図３のトップレベルのもの及び表１を使用することとした。この教育用ツールの最終目的は、設計技術の習得とそれに関連

する要素技術の習得及び応用であるが、このために、次のようなものを備えるべく考えている。

・前提となる要素技術の説明
・いくつかの練習問題（図２の計測制御システム・モデルは、その一つ）
・各段階毎に作成されるオブジェクトの例（正解例）
・オブジェクトをまとめるための標準的形式
・各段階毎のチェックリスト

このような要件を基に、若手技術者（経験２－３年）にソフトウェア設計の疑似体験をさせる。

このソフトウェア設計教育用ツールは、あくまでもソフトウェア設計手順の内、純粋に技術的な部分に焦点を当ててその範囲に絞っているが、実際の設計現場における業務には、前述の計算機機種決定のような曖昧な根拠に基づく判断、プロジェクト管理、リスク管理などのような総合的な判断、意思決定などが包含される。[3][4]

我々は、このような部分に対しては、今回抽出したソフトウェア・プロセスとは別のレイヤに位置づけて究明して行くつもりである。図６にソフトウェア設計作業とそれを取り巻くプロジェクト管理業務との関係を示す。このＳＥ業務全体に及ぶ分析は、今後の課題である。

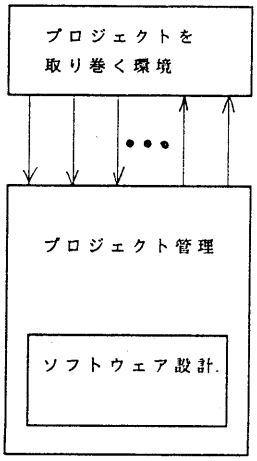

図６　ソフトウェア・設計と
プロジェクト管理関係

４．まとめ

我々の最終的ゴールはＳＥ教育用ツールであるが、そのために、まず現実のシステム設計の手順を地道に分析し、ソフトウェア・プロセス（図１）を抽出した。

実際のシステム設計は、きわめて多くの不完全な情報を整理統合し、補完しつつ進められるためシステム設計手順そのものも煩雑なものとなっている。

このソフトウェア・プロセスは、システム設計の実態を示しているという意味での価値は十分にある。しかし、他の技術者がこれをそのまま利用（REUSE）して別の設計課題を解く場合、種々の問題があることが判明した。

そこで、ソフトウェア・プロセスを再度整理し、理解し易くかつ応用範囲の広いもの（異なった規模のソフトウェアにも適用できる。）をまとめた。

今後このソフトウェア・プロセスを中心とする教育用ツールをさらに充実させて、実際に若手技術者の教育、訓練に適用し、評価、改善を続ける所存である。　なお、これまでのソフトウェア・プロセスの分析は、設計技術そのものに焦点を当てているが、今後は、それ以外の事項をとりあえず一括してプロジェクト管理のレイヤとして捉えて究明することとし、ＳＥ作業そのものを浮き彫りにしたいと考えている。

参考文献

[1] L. Osterweil, "Software Processes are Software too, "Proceeding of the Ninth International Conference on Software Engineering, Montery, California, pp. 2-13, April, 1987.

[2] T. Katayama, "A Hierarchical and Functional Software Process Description and its Enaction, "Proceeding of the 11th International Conference on Software Engineering, pp. 343-352, 1989.

[3] B. W. Boehm, "A Spiral Model of Software Development and Enhancement, "ACM

Software Engineering Notes, vol. 11, no. 4
, pp. 14-24, August, 1986.
[4] David Dixon, "Integrated Support for
Project Management, "Proceedings of the
10th International Conference on Software
Engineering, pp. 49-58, 1988.
[5]望月純夫、山内顕ほか：ソフトウェア・プロ
セス一実時間処理システムにおけるケース・ス
タディー：情報処理学会研究報告，90-SE-71 ，
pp. 139-148(1990).