

JAERI-M
90-114

JT-60制御用計算機システムにおける
ソフトウェアの品質管理

1990年7月

伊佐治 信明*・栗原 研一・木村 豊秋

JAERI-Mレポートは、日本原子力研究所が不定期に公刊している研究報告書です。
入手の間合わせは、日本原子力研究所技術情報部情報資料課（〒319-11 茨城県那珂郡東海村）あて、
お申しこしてください。なお、このほかに財団法人原子力弘済会資料センター（〒319-11 茨城県那珂郡
東海村日本原子力研究所内）で複写による実費領布をおこなっております。

JAERI-M reports are issued irregularly.
Inquiries about availability of the reports should be addressed to Information Division Department
of Technical Information, Japan Atomic Energy Research Institute, Tokaimura, Naka-gun, Ibaraki-
ken 319-11, Japan.

© Japan Atomic Energy Research Institute, 1990

編集兼発行 日本原子力研究所
印刷 ニッセイエプロ株式会社

JT-60制御用計算機システムにおけるソフトウェアの品質管理

日本原子力研究所那珂研究所JT-60試験部

伊佐治 信明*・栗原 研一・木村 豊秋

(1990年6月18日受理)

臨界プラズマ試験装置JT-60は、実験装置という性格から、装置及び制御系の改造、機能変更がかなりの頻度で行われる。この制御系の中核である計算機システムは、大規模なソフトウェア群を保有しており、そのソフトウェアの改造においても、ソフトウェアバグの発生を防止し、全体の品質を維持すること、即ち“品質管理”が、実験運転の効率向上の観点で極めて重要である。この品質管理を実行するには、標準的作業手順(作業標準)が規定され、その手順の各過程で作業者を支援するシステムの存在が必要となる。

本報告書では、これらJT-60制御系におけるソフトウェア開発の作業標準とそれらを支援するシステム、並びにこれらを実際のシステムで運用した結果について述べる。

Quality Control of the Software in the JT-60
Computer Control System

Nobuaki ISAJI*, Kenichi KURIHARA and Toyoaki KIMURA

Department of JT-60 Facility
Naka Fusion Research Establishment
Japan Atomic Energy Research Institute
Naka-machi, Naka-gun, Ibaraki-ken

(Received June 18, 1990)

The JT-60 Control System should be improved corresponding to the experimental requirements. In order to keep the integrity of the system even in the modification the concept of quality control(QC) was introduced in the software development. What we have done for QC activity are (1) to establish standard procedures of the software development, (2) to develop support tools for grasping the present status of the program structure, and (3) to develop a document system, and a source program management system.

This paper reports these QC activities and their problems for the JT-60 control system.

Keywords: Quality Control, Software Standard Procedure, Support Tool,
Software Management System, JT-60 Control System

* On leave from Japan Expert Clone Co. Ltd.

目 次

1. はじめに	1
2. ソフトウェアの品質管理	2
2.1 ソフトウェアの品質低下要因	2
2.2 JT-60制御系におけるソフトウェアの品質管理	7
3. ソフトウェアの作業標準	9
3.1 作業標準の概要	9
3.2 作業標準の効果	11
4. 支援ツール	13
4.1 ソフトウェアの現状把握支援システムとその効果	13
4.2 ドキュメント管理システムとその効果	21
4.3 ソースデータ管理システムとその効果	25
5. まとめ	28
謝 辞	28
参考文献	28

Contents

1. Introduction	1
2. Quality Control of the Software	2
2.1 Causes of quality deterioration of the software	2
2.2 Quality control of the software in the JT-60 control system	7
3. Standard Procedure of the Software Development	9
3.1 Outline	9
3.2 Effect	11
4. Support Tools	13
4.1 Support tools for grasping the present status of the software and their effects	13
4.2 Document control system and its effect	21
4.3 Source program management system and its effect	25
5. Concluding Remarks	28
Acknowledgments	28
References	28

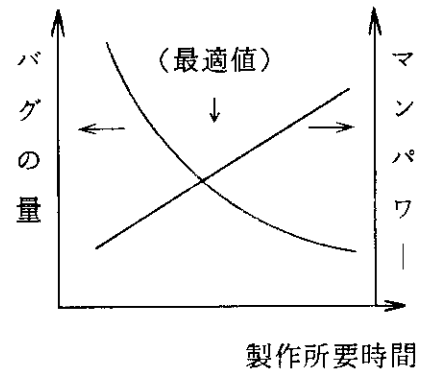
1. はじめに

臨界プラズマ試験装置JT-60（以降、「JT-60」という）は、実験装置という性格から、装置及び制御系の改造がかなりの頻度で行われる。JT-60制御系の中核である「全系制御設備計算機システム」は、7台の制御用計算機HIDIC-80E（㈱日立製作所製）を中心に構成されており、これら計算機が実行するプログラム数は約2500、またテーブル数が約3200の合計50万ステップと膨大な数である上、これらのプログラムは、制御用計算機システム特有の複雑な相互関係^{*1}を持っており、これらソフトウェアの改造作業を行う場合、改造部分の機能が正しく発揮されることは勿論で制御系全体の整合性を確保していることが要求される。この様な要求に答えられる1つの方法がソフトウェアの品質管理である。

ソフトウェアの品質管理の仕方については、数多くの成書が存在し、如何にソフトウェアバグ^{*2}（以降、「バグ」という）を最小限にしてソフトウェアを製作するかについて、それらのどれもが完璧なまでの手順分析を行っている。そして、その通りに行えばおそらくはバグを排除し、高い品質のソフトウェアが製作できることであろう。

しかし、我々はそれらをそのままの形で実用に移すことに強い抵抗があった。それは、実際のソフトウェア開発ではバグが少ないことと同様に manpower（時間×人）的に効率良くソフトウェアを製作することが重要な評価の一つである。換言すれば如何にバグが少なくても莫大な人手と時間がかかってしまったのでは、実用に供する品質管理とは言えない訳である。単純化すれば第1.1図の様な図式になっている。この最適値を模索し、JT-60制御系開発・改造の経験に基づいて、現段階での最適と思われる品質管理の姿を明らかにすることを試みたのが本報告である。品質管理の具体的な実施方法は、まず、「ソフトウェア開発の作業標準」を構築し、次にそれに従ったソフトウェアの開発作業を支援する「ソフトウェアの現状把握支援システム」、「ドキュメント管理システム」、「ソースデータ管理システム」の開発を行い、その後実運用（1986年～1990年）を行った。

第2章では、JT-60制御系におけるソフトウェアの品質管理体系について述べる。第3章では、確立したソフトウェアの作業標準について述べる。第4章では、ソフトウェアを開発する際に用いる（ソフトウェアの現状把握支援システム、ドキュメント管理システム及びソースデータ管理システム）に関する内容を述べる。最後にまとめを行う。



第1.1図 バグとマンパワーの製作所要時間に対する定性的関係

*1 制御用計算機では、プログラム（タスク）による別のタスクの起動、終了及び中断、再開、タイマを利用したタスクの周期的実行、外部要因によるプログラムの割り込み起動、複数のプログラムによる共有テーブルの参照及び占有・解放といった複雑なプログラムの関係が複数の計算機間で存在している。

*2 プログラムミスによる不具合。

2. ソフトウェアの品質管理

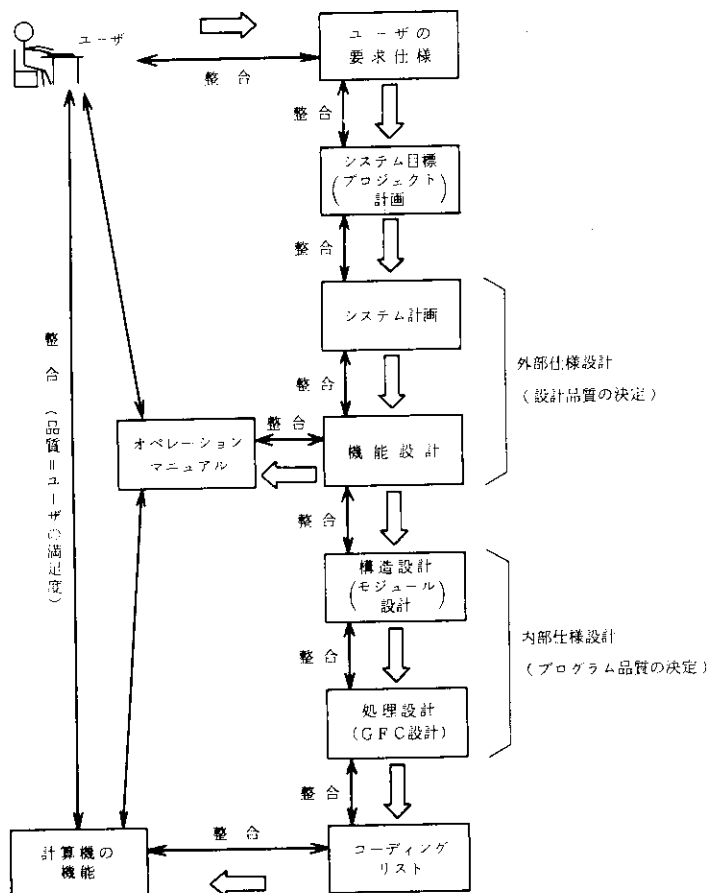
2.1 ソフトウェアの品質低下要因

本節では、ソフトウェアの品質管理をどのような体系で実施するかを決定するための検討結果について述べる。ソフトウェアの品質管理体系を確立するにあたっては、ソフトウェアの品質低下要因を検討し、それら検討結果をソフトウェアの品質管理体系構築の方針決定のための資料とした。

(1) ソフトウェアの品質低下要因

ソフトウェアの「品質」は、最終的には「ユーザの満足度」として定義出来る。高い満足度を得ることは、「プログラムがユーザの要求に整合して動作すること」である。ここでは、この条件を満足しないソフトウェアが出来る一般的な要因について調べる。

ソフトウェアの製作は、問題提起に始まり問題の解決をコンピュータに指示するための詳細な命令の巨大な集積に終る一連の処理である。従って、ソフトウェアの製作はいくつかの同値変換過程と見做せることから、変換時の間違いがソフトウェアの品質を低下させると考えることが出来る。第2.1.1 図はソフトウェアの開発工程を示しているが、各作業の成果物間の前後の整合が正しく取れていないと、ユーザの要求と計算機の機能が大幅に食い違ってしまう。各作業の成果物間において概略次の様なバグが生じる可能性がある。



第2.1.1図 ソフトウェアの開発工程と品質

① ユーザの要求から要求仕様書への過程

この過程では、要求仕様をユーザが作成する場合と、ユーザと製作者との打合せの場でユーザの要求を製作者で纏める場合がある。ここでのバグは、ユーザが自分の要求を適切に表現出来ない場合や、要求が誤って解釈される場合等である。

② 要求仕様書からシステム目標への過程

この過程では大きな変換は行われませんが、要求仕様の考慮すべき評価目標を見落とししたり、ユーザの要求に明示されていないが必要な目標を見落としてしまうとバグとなる。

③ システム目標から外部仕様設計への過程

この過程は、情報変換の量が非常に多くソフトウェア開発の中で最も重要な過程である。この過程では、多面的なレビューを行いユーザの要求事項や環境条件の見落とし等のバグを早期に発見することが重要である。

④ 外部仕様設計から内部仕様設計への過程

ここでは、仕様書をプログラムに焼き直す過程（人間が理解出来る言葉を計算機が理解できる言葉に置き換える作業）であり、情報を変換する際にバグが入り込む可能性が大きい。外部仕様と内部仕様の整合は、機能とプログラムとの対応が不明確になり易いので、保守性の面で問題となることが多い。

⑤ 内部仕様設計からコーディング及びパンチを行う過程

この過程では、ソフトウェアの論理的仕様をプログラミング言語で書かれたプログラムに変換し、プログラムの翻訳を行う。ここでのバグは、書き誤りや記述漏れが多くその量も多いが、修復はそれ程困難ではない。

以上の過程が終了した後、各設計、製作過程で決定した仕様が仕様通りに動作するかを計算機上で確認する作業を行う。確認する項目は、それぞれ処理設計仕様の確認、構造設計仕様の確認、機能設計仕様の確認、システム設計仕様の確認である。

(2) ソフトウェア品質低下の直接要因の検討

以上は、ソフトウェア製作過程の中で発生する品質低下要因を作業単位で検討してきたが、次に品質低下を直接起こすと思われる要因について検討していく。まずこれらを検討するため、JT-60 制御系開発関係者に、3つの設問を出して意見を収集した。①ソフトウェア開発にあたりその品質を保証するには何が必要か？。②高品質なシステム設計を行うためには何が必要か？。③何故プログラムの中にバグがあるのか？。それら意見を分析し関連性のあるものを集めていくつかのグループに分類した。各グループの問題点と対策の要点を以下に述べる。

(a) 要求の問題として以下が挙げられる。

①仕様において必要条件と十分条件の誤認識。

②頻繁な仕様変更。

③表面的には問題とは見えないが実際は矛盾している要求。

④要求を信用しすぎる。（ユーザの要求は通常最少のものが提示される。従って、すぐに新しい要求が発生することがある。）

これらは、要求を出す側と受ける側の問題といえよう。従って、ソフトウェアへの要求の

理解が双方で確実に行われていなくてはならない。

(b) 情報伝達、意志疎通の問題として以下が挙げられる。

- ① 会話や討論の不足。
- ② 悪い人間関係。
- ③ 設計思想の不協和や一貫性の欠如。
- ④ 自分の殻に閉じ籠り易い性格の人の存在。

ソフトウェア開発にあたっては、小規模のプログラムを除いて通常複数の人間で作業が進められる。このような場合、各人の作成するプログラムは互いに何らかのデータなどの取り合いを持つ。従って、話し易い雰囲気作りや、定期的な打ち合せを行うことで関係者間のコミュニケーションをとることが、人間の集合で製作する物の品質を保つために極めて重要なことである。特に、作業分担を明確にして誰が何をやっているのか互いに把握出来る様にし、製作ソフトウェアの関係を人の関係に置き換えることで、システムの整合性を図ることが重要である。

(c) 計画、資金の問題として以下が挙げられる。

- ① 予算不足。
- ② 厳しすぎる工程やマンパワー。
- ③ 製作責任者の予算執行能力が低い。
- ④ 作業者の個性や性格を無視した計画。

資金計画や作業工程は、“開発工数の見積り”と強い関係がある。見積りの正確さが計画の成否を決定するといっても良い。この見積り過程は、経験にたよる部分が大きく、一般的な方法論は無い。また、計画の遂行については、適切な調整役的存在が大規模ソフトウェアの製作に不可欠である。

(d) 整合性の問題として以下が挙げられる。

- ① 全てのシステムを理解している人がいない。
- ② 動作環境を十分考慮している人がいない。
- ③ 入出力の整合性や取り合いの検討が不足。
- ④ 全体の動きとの同期に対する検討不足。

ソフトウェアは、個々のプログラムが動作するだけでは不十分で、全体の整合性がとられて始めて機能を発揮する。従って、システムに対するソフトウェアの動作環境を理解し、システムと入出力関係の明確化などの検討は必要である。また、全体の動作に対して知識を有する人の存在が重要となる場合も多い。

(e) 計算機操作の問題として以下が挙げられる。

- ① 操作方法が解らない。
- ② 出力メッセージの内容が理解しづらい。
- ③ オペレーションマニュアルが理解しづらい。
- ④ オペレーション手順が繁雑。
- ⑤ 異常に対する対処方法が解らない。

以上は、総じてテスト及びデバック^{*1}時に現れることが多い項目である。これらは、計算機側の支援システムの整備やマニュアル作成時の考慮など、計算機製作メーカーに改善を期待することになる。

(f) ドキュメントの問題として以下が挙げられる。

- ① 複雑過ぎるドキュメント。
- ② 情報不足のドキュメントやドキュメント化されていない。
- ③ ドキュメントを作らずに端末の前に座ってしまう。
- ④ 十分な理解をしないうでドキュメントを作成する。
- ⑤ 曖昧な表現、難解な文章。

ソフトウェアは、プログラムを説明するドキュメントを作成することにより完備した製品となる。ドキュメントに必要事項が漏れなく記述されていれば、プログラムの製作過程や、システム運用時に発生する不具合の対処及び、システムの改造作業等における作業効率が高まることであろう。従って、体系的なドキュメントの作成や、ドキュメント記述形式の規定を行うこと及び、ドキュメントを作成するのに必要な時間を十分に取ることが重要である。

(g) 設計／プログラミング／保守の問題として以下が挙げられる。

- ① 設計仕様からプログラムへの翻訳ミス。
- ② ソフトウェア作成作業標準からの逸脱。
- ③ 複雑過ぎるプログラム構造。
- ④ 注釈の少ないプログラムの作成。
- ⑤ 想像力の不足（特に画面表示設計の際）。
- ⑥ 不適當なプログラム言語。
- ⑦ 自然言語から程遠い。

処理設計やコーディングの作業で発生するバグは、多種多様であるがその大半は、単純なミスによるものである。その代表例としては、要求仕様からプログラムへの翻訳ミスによるものである。上記の問題点①～④の裏返しでバグの低減は可能であろう。⑥、⑦は、将来的に解決されるべき問題として位置づけられる。

(h) 能力の問題として以下が挙げられる。

- ① 低い技術力。
- ② 学習不足。
- ③ 新しい技術の習得に挑戦しない。
- ④ 人の言うことを誤解し易い。
- ⑤ 不適切な人員の採用。
- ⑥ 作業内容と個人の性格とのズレ。

ソフトウェアの製作に限らず、品質の高い生産物を作成するには、高い技術力を持った人材が必要となる。上記に示す問題があると、要求したソフトウェアが出来なかったり、作成されたとしても品質が低かったりする。さらに、新しい技術の導入や工夫をするといった進

*1 ドキュメント上や計算機上でプログラム中に存在するバグを見つけ出して修正すること。

歩性のある生産物が期待出来ない。従って、講習や研修を行い各担当者が技術力を身につけることや、管理責任者が各担当者の技術力や適性を見極めて、見合った作業を与えることが必要である。

(i)環境の問題として以下が挙げられる。

- ①ひどい作業場所（暑／寒過ぎる、騒がしい、端末が少ない、連絡手段が無い）。
- ②必要なドキュメントが整っていない、何処に有るのか分らない。
- ③不適當な報酬（収入／地位）。

ソフトウェアは、人間の知的生産物であるので、作業環境の良否もソフトウェアの品質に影響を与えることがある。作業場所が悪く集中力を失ったり、作業に必要な道具が整っていないなかったり、ドキュメントが整理されていないため、それを探すのに時間がかかったりすることなどは、品質を低下させる原因となる。これらは特に、現場（計算機設置場所）で発生することが多い。ドキュメントについては、管理体系や検索ツールを整えることで解決出来ることが、第4章で示される。

(j)テスト、デバックの問題として以下が挙げられる。

- ①質的、量的に、テスト項目が不十分。特に、異常ルートの確認を怠る。
- ②テスト結果の確認が不十分。
- ③テストを行った自分自身を信用し過ぎる。
- ④テスト環境が不十分。
- ⑤徹夜作業等長時間労働での思考能力の低下。
- ⑥ドキュメントと計算機内容が一致していない。
- ⑦計算機が混んでいて作業が進まない。

プログラムにはバグが必ず存在すると思えなければならない。バグを見つけ出すことやそれを修正する作業の殆どは、テストの過程で実施される。また、プログラム中に存在するバグの大半は、単純なミスによるものと考えてよい。バグ自身を減少させることは、これら単純ミスを摘出するという地味な作業を注意深く行う姿勢が大事である。例えば、自分を過信しない、正常ルート以上に異常ルートの確認を抜け落ちなく行う、バグの発生を恐れず通常時以外の事象を想定したテストを行う、机上での確認を入念に行うといったことで、バグの減少が期待できる。

以上に述べた各項目の問題点のうち、人間関係、作業環境、資金・工程、人員計画等の問題は、品質管理を行うことによって解決出来る問題ではないが、その他の問題点は、ソフトウェアを製作する手順、計算機のプログラム構造の明確化、ドキュメント類の整備等を体系的に整えることで対応出来ると考えられる。

2.2 JT-60制御系におけるソフトウェアの品質管理

JT-60制御系におけるソフトウェアの品質管理手法の整備にあたっては、前節で述べたバグの発生原因の検討結果を考慮した。その第一点は、ソフトウェアを製作するにあたって必要となる作成順序や各作業の成果物間での作業規定に関するもので、ソフトウェアの開発・改造作業の一連の流れ（要求からシステムの運用まで）を示す作業標準を構築した。第二点は、この作業標準に従った作業を支援するための手法の整備である。現状の計算機内容の把握を可能としたシステムを開発し、またドキュメント及びソースデータの所在検索を容易に可能とするシステムを開発した。以降に開発した各品質管理の手法とそれらの概要を示す。

(1) ソフトウェア開発における作業標準の構築

ソフトウェアの開発・改造作業の実施における、要求から設計→製作→試験→整理作業→運用に至るまでの作業手順を定め、ドキュメント作成の手引となる規範を定めた。これにより、ソフトウェア開発過程の高効率化を図ることを目的とした。しかし、これは最適なソフトウェアの品質管理を行う上での必要十分条件ではなく、十分条件の記述である。具体的な適用をどの様にするかは第3章に述べる。

(2) ソフトウェアの現状把握支援システムの開発

ソフトウェアの開発・改造作業を効率良く実施することを目的として、対象とするプログラムを中心とした関連情報を正確かつ迅速に把握出来る様にしたシステムを開発した。本システムは汎用計算機（㈱富士通製のFACOM M780）上に構築され、「対象プログラムとリンクされているプログラム、テーブルは何か」、「対象プログラムの上位プログラムは何か」、「対象テーブルを参照しているプログラムは何か」等を容易に知ることが出来る。汎用計算機には、外部情報（制御用計算機から出力されるプログラム登録時の結合・編集*1実行時に生成される結合・編集情報*2と、人間が調査したタスク間の起動関係情報及び、テーブルの参照関係情報）を収集してデータベースを作成し、それら情報を帳票形式で出力するシステムを開発した。

これにより、実験運転時に早急に対策をしなければならないソフトウェアの不具合が発生した時であっても、そのプログラムを修正したことによる、他のプログラムへの影響の有無等の情報が迅速に把握出来る。

*1 プログラムを計算機に登録する際に実施される手続きで、使用しているプログラムやテーブルを結合（飛び先番地の決定等）し、実行可能プログラムを作成すること。（以降、「LEDT」と称す場合もある）

*2 プログラムを計算機に登録する際に計算機より出力される情報で、内容は使用プログラム・テーブルの略称、種別（常駐サブルーチン、従属サブルーチン、非常駐サブルーチンの区別）、容量、結合配置等である。

(3) ドキュメント管理規定の整備

ソフトウェアの開発・改造作業の下調べやシステム完成後のドキュメント整理等に利用するために、各種ドキュメント（設計図書*1、フローチャート*2、テーブル構成図*3、プログラムリスト等）の所在検索と、ドキュメント修正からマスター管理等の管理手順を規定した。これにより、参照したい最新（現状の計算機内容と一致した）のドキュメントを迅速に検索することが出来る。

(4) ソースデータ管理システムの整備

制御用計算機のソースプログラムは、今まで紙カードで管理されていたが、紙カードは欠点が多く（紙質が時間と共に劣化しそれを修復するには再穿孔が必要となる、ちょっとしたことでバラバラになり易く持ち運び上の安全性に欠ける、カード読み取り装置でのカード読み飛ばしやジャム*4により作業効率が低下する、カード収納庫により有効な空間が狭くなる等）、ソースプログラム管理の効率向上とメンテナンスの簡略化を図るため、ワークステーション（㈱ソニー製のNEWS「NWS-841」）上に一元管理することにした。このため、必要なソースプログラムの所在検索システムを整備し、ワークステーションで開発、編集されたソースプログラムを制御用計算機に読み込ませる手順や、汎用計算機で開発されたソースプログラムをワークステーションに移す手順及び、ソースプログラムのマスター管理手順を規定した。これにより、参照したい最新（現状の計算機内容と一致した）のプログラムやテーブル等を迅速に検索・修正することが出来る。

*1 システムや機能の仕様を詳細に記述した書類。

*2 プログラムの手順を図式的に表現した書類。（以降、「GFC:General Flow Chart」と称す場合もある）

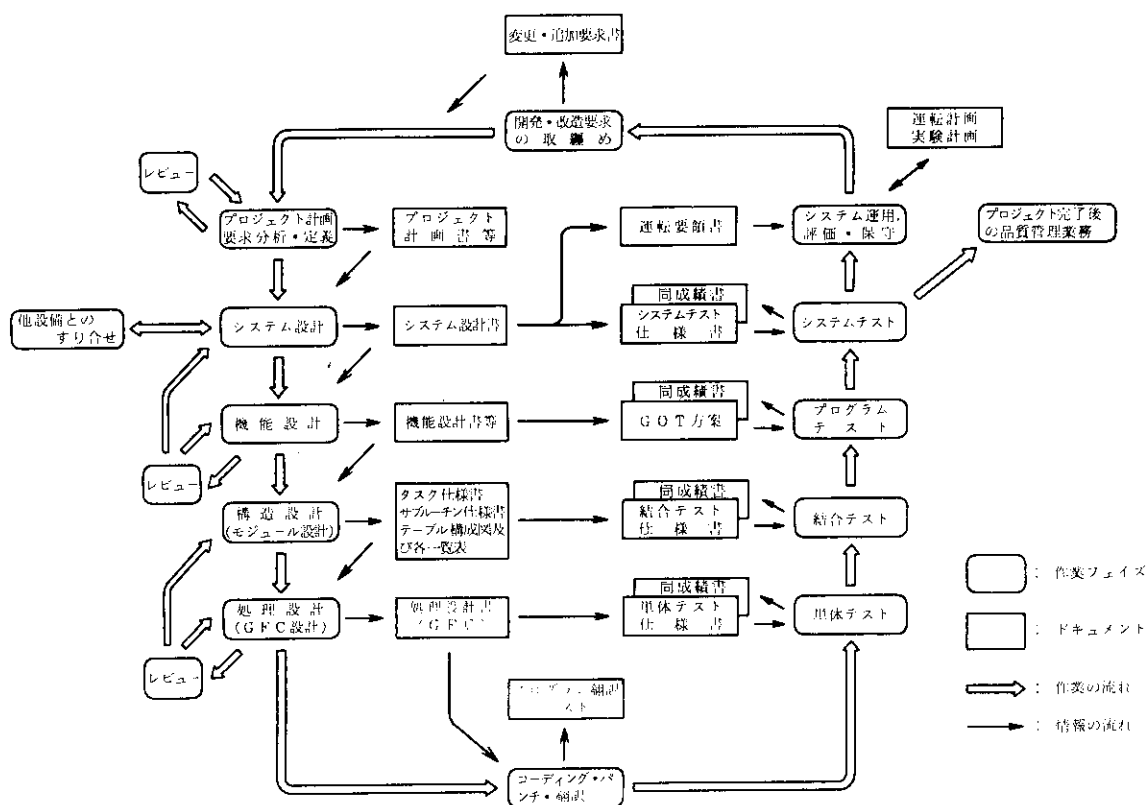
*3 テーブル形式に格納されたデータの構造やその参照関係を記述した書類。

*4 カードがカード読み取り装置内に巻き込まれるなどのトラブル。

3. ソフトウェアの作業標準

3.1 作業標準の概要

ソフトウェアの開発・改造作業の一連の流れを第3.1.1 図に示す。本節では、各作業の成果物間での作業内容と作成されるドキュメントの種類を中心にソフトウェアの作業標準（以降、「作業標準」という）の内容を示す。開発作業の一連の流れは、大別して「計画」、「設計・製作」、「試験」、「運用」の4つの段階に分類出来る。



第3.1.1図 ソフトウェアの開発・改造作業工程と作成ドキュメント

(1) 計画段階での作業

①開発・改造要求の取り纏め

ユーザから出された要求事項を整理し内容を吟味して、実現へ向けての検討事項の優先順位を決める。

②要求分析

ユーザからの要求仕様書等をよく検討し、ユーザと製作者の打合せ等で要求事項の分析、整理を行う。この過程でユーザの真意をつかみ、後で仕様変更となった事項も含めて、システムとしての必要条件を明確にする。また、この過程でシステム開発の必要性と可能性を吟味し、プロジェクト計画に進むか否かを決定する。

*1 ある目的に叶ったシステムを実現する際の一連の作業。

③プロジェクト計画

開発システムまたは改造についての概念設計を行い、費用対効果の分析、予算の都合、人的・物質資源計画、開発計画等を検討して、計画を実行に移すかどうかを決定する。実行と決まった場合は、計画を推進するために必要な体制、製品目標、日程計画、人員計画、機材の計画等を具体的に決定にする。また、新システムの移行・運用計画についても検討する。

④プロジェクト計画に関するレビュー

ユーザの要求事項を把握した後、①～③の過程についてのレビューを行う。

(2) 設計・製作段階での作業

①システム設計

プロジェクト計画における概念設計に基づき、システムの機能を確定する。また、その機能を実現するために必要な基盤（ハードウェア及びソフトウェアに渡るシステム構成）を明確にする。さらに、システムを機能単位に細分化しその仕様の決定を行い、入出力データの項目の洗い出しを行う。

②機能設計

細分化された機能に漏れが無い確認を行い、入出力データのテーブル、ファイル構造を明確にする。また、機能をプログラム（タスクまたはタスク群等）に分割し、プログラム間の処理の流れを決定する。

③外部設計仕様に関するレビュー

①、②の内容を対象に以下の項目についてレビューを行う。

- *設計がユーザの要求やシステムの環境条件に合致しているか、誤解や見落としが無いか。
- *信頼性、保守性、拡張性、操作性などの品質要求に対し、設計上の問題は無いか。
- *機能の定義は明確でかつ矛盾が無いか。
- *データ構造が適切でファイルやテーブルの構成は明確か。
- *入出力処理及び、タイミングの検討が十分か。
- *障害に対する設計が十分か。

④構造設計

機能設計に基づいて、各プログラムの内部構造設計を行う。即ち機能を実現するためにプログラムを主プログラム（タスク）、サブルーチンに分割し、各プログラムの処理仕様とプログラム間の結合関係を決定する。

⑤処理設計

プログラム内の詳細処理手順を設計し、定数値の決定が必要な場合は、この過程でテーブル・ファイル構成図を作成する。

⑥内部仕様設計に関するレビュー

構造設計、処理設計の設計内容が、機能設計の内容と合致しているか、プログラム分けは適当か、処理抜けが無いかな等を確認する。

⑦コーディング、パンチ、翻訳

処理設計までの各種設計書を基にプログラム、テーブルのコーディング、パンチ及び翻訳チェックまでを実施する。

(3) 試験段階での作業**① 単体テスト**

プログラム単位での動作チェックを行う。

② 結合テスト

プログラム(タスク)単位での動作チェックを行う。

③ プログラムテスト

機能設計で規定された各機能が、正しく動作するかをチェックする。

④ システムテスト

システム全体として所定の機能が、正しく動作するかをチェックする。

⑤ 整理作業

システムテストが完了したら、各種ドキュメント、ソースプログラム等の整理、及びソフトウェアの現状把握支援システムのデータベースの更新作業を行う。

(4) 運用段階での作業(システム運用と評価)

新システムの運用を開始し、運用実績に基づいて評価を行い、問題点が有れば整理して次の改造要求とする。また、システムテストまでの過程で見つからなかったバグが発生したら、迅速に対処する。

3.2 作業標準の効果**(1) 作業標準の有効性と問題点**

作業標準を作成してから4年間に、10数回のソフトウェアの開発・改造作業が実施された。それら開発・改造作業にあたっては、この作業標準に従って作業を実施した。その結果、作業標準に対して、次の様な有効性が確認され、問題点も指摘された。

① 有効性

作業標準は、ソフトウェアの開発・改造作業を初めて経験する人(経験の浅い人)に対して、ドキュメント等に記載すべき内容や設計作業上の注意事項等が纏められていることや、ソフトウェアの開発・改造作業がどの様に進められるのか、どの様な作業を行えば良いのか等の知識を学習させることが出来た点で効果的であった。また、各作業の成果物間での品質チェック項目が明確に示されているのでチェック漏れを減少出来たことや、各過程での作業内容が明確に示されているので、各過程で無駄な作業が減り過程から過程への切り替えを容易に進めることが出来た。

② 問題点

作業標準の問題は、規定された内容が細かいため、それに従って作業を進めるとマンパワー的に厳しくなるということや、書かれている内容が一般的であり、本質的な部分とそうでない部分が考慮されていないということが挙げられる。この様な問題点を踏まえると次項に述べる改善案が考えられる。

(2) 作業標準の改善

作業標準は、こうすればどんな人でもバグを最小限化出来るという方法をうたったものである。従って、本来作業担当者の知識、経験、ソフトウェア開発能力、制御システムの理解度によって、その個人に応じた作業標準が存在すべきである。この様な視点から、作業標準を次の様に改善して実施する。

まず、必須項目とそうでない項目を区分する。例えば、ドキュメントの作成において、作成しなければならないドキュメントが何か、最低限盛り込まなければならない内容が何か、各過程でのレビュー項目において、必ず確認しなければならない項目（過去の統計による最も誤りの発生しやすい項目）は何かといった”ツボ”を押さえた手順とし、それを必ず実施する様にする。

次に、利用方法を各人各様になることを許容する。これは、一見”標準”と銘打って置きながら、それと矛盾してる様に見える。しかし、各人にとっての最適な手順を確立していく過程をガイドすることは、本当の作業標準の確立と言える。例えば、全く作業標準（ソフトウェアの開発・改造作業）に対して知識のない人は、基本的に全ての面で作業標準に従った作業を実施することを原則とする。そのためには、ソフトウェアの開発・改造作業の工程計画に、作業標準に目を通す工程（教育訓練等）を設けて、作業標準の内容や注意事項を前もって理解する必要がある。また、システムを理解していて、ある程度の技術力を身につけたと自ら判断できる人は、作業標準に記載されている必須項目を除く部分に関しては、作業自身自身の判断で決定し作業を進めていく。さらに、JT-60制御系システムや計算機の内容を理解している人や、ソフトウェア開発の経験が深い人は、本作業標準を補助的に利用し、必須ドキュメントの提出以外は、作業標準の順守を強制しない。

自分に最適なやり方を見出す手順として、最初はバグを最小限にするために必要な事項を網羅的に知り、その後自分にとって注意を払う必要がある部分を中心に利用する。あくまで目的はバグのないソフトウェアの実現であるのだから、前記の利用方法で苦痛を味わうことなくそれが可能となると言える。

4. 支援ツール

4.1 ソフトウェアの現状把握支援システムとその効果

ソフトウェアの現状把握には、大別して制御用計算機が保有しているソフトウェアのプログラム構造、プログラム（タスク）間の相互関係、プログラムの従属関係、テーブル（プログラム間で共有の）の参照関係等の計算機内部情報の把握と、プログラムリスト、フローチャート、テーブル構成図のドキュメントの所在や、プログラム、テーブル、ジョブ制御文^{*1}（JCL）が格納されているワークステーション上の所在を検索するドキュメント、及びソースデータ^{*2}に関する管理情報の把握がある。これらに対応して、制御用計算機からの情報と人間が調査した情報に基づいて、必要とするデータベースを作成するための機能を汎用計算機上に構築し、容易に利用出来る形態として出力する「ソフトウェアの現状把握支援システム」を整備した。第4.1.1 図に本システムの概略構成を示す。

計算機の内部情報は、制御用計算機より出力した情報と各種ドキュメントから人間が調査して抽出した情報を合わせて汎用計算機上でデータベースとして構築される。各翻訳の終わった主プログラムを結合・編集実行時に生成される結合・編集情報を制御用計算機より磁気テープ（以降、「MT」という）に出力させて、汎用計算機に入力して解析しデータベースが作成される。このデータベースを基に帳票を出力させて各種情報検索の支援に利用される。

本支援システムの特徴は、「対象タスクに対しどのタスクから起動、停止等のタスク管理マクロ^{*3}が発行されているか」、「対象プログラムがどの上位プログラムからCALLされているか」、「対象タスクからどのタスクに対し起動、停止等のタスク管理マクロが発行されているのか」、「対象テーブルを直接参照しているプログラムは何か」といった情報が迅速かつ正確に把握出来ることである。

ドキュメント等の管理情報は、各種ドキュメントより、人が調査した情報を汎用計算機に入力して処理し、帳票形式で出力させて検索の支援に用いられる。

*1 プログラムやテーブル等を計算機に登録する際に使用される制御文。

*2 ワークステーション上のプログラム、テーブル、JCLの各ソースを示す。

*3 タスク間で発行されるタスクを制御する命令。

《入力情報》

＜計算機内部情報の把握ツールで使用する情報＞

- ◆制御用計算機が出力する主プログラム単位の結合・編集情報より以下を抽出
 - *使用プログラム・テーブルの略称, 種別
- ◆人が以下を調査し抽出する
 - *タスク間相互関係情報 (タスクの起動・停止等)
 - *テーブル参照関係情報 (テーブル参照形式: READ, WRITE, RESERVE, FREE)

＜ドキュメント等の管理情報の把握ツールで使用する情報＞

- ◆人が以下を調査し抽出する
 - プログラムリスト, フローチャート, テーブル構成図の所在, 改造プログラムの履歴, プログラム・テーブルが格納されているワークステーション上の所在

↓ (入力)



《汎用計算機》

- ◆入力情報解析機能 (情報の入力, 編集, データベース作成, 帳票出力) により計算機内部情報の把握で使用するデータベースが作成される

- ◆手作業によりドキュメント等の管理情報の把握で使用するデータベースを作成する

↓ (出力)



《出力情報》

＜計算機内部情報の把握ツールで利用される帳票＞

- ◆入力情報解析機能により、以下の帳票が出力される
 - *タスク間関連情報リスト
 - *プログラム単位情報リスト
 - *テーブル参照情報リスト
 - *プログラム構成図
 - *プログラム単位テーブル参照情報リスト

＜ドキュメント等の管理情報の把握ツールで利用される帳票＞

- ◆検索情報リスト出力機能により、以下の帳票が出力される
 - *プログラム検索情報リスト
 - *テーブル検索情報リスト
 - *L E D T・J C L 検索情報リスト

第4.1.1図 支援システムの概略構成

(1) データベースの種類

① 計算機内部情報の把握ツールで使用するデータベース

a) タスク間関連マトリクス

制御用計算機内には多数のタスクが存在し、他タスクの起動、停止等を行う命令を発行して相互に関連している。本マトリクスは、これらのタスク間の関連についての情報を格納する。

b) プログラム構成マトリクス

プログラム、サブルーチンが、どのサブルーチンを直接CALLしているかという情報を持つ。主プログラムから順次検索していけばプログラム全体の構造を知ることが出来る。

c) 上位プログラム検索マトリクス

タスク、常駐サブルーチン、非常駐サブルーチンなど結合・編集する単位毎に、その全ての従属サブルーチンの情報を持つ。従属サブルーチンの側から調べると、サブルーチンがどの主プログラムに従属して使用されているかを検索することが出来る。

d) テーブル参照マトリクス

タスク、サブルーチンが、どのテーブルを直接参照しているかという情報を持つ。

② ドキュメント等の管理情報の把握ツールで使用するデータベース

a) プログラム検索情報ファイル

本ファイルは、プログラムについて、フローチャート、プログラムリストの所在（保管先のバインダー番号等）、改造プログラムの履歴及び、ソースプログラムが格納されているワークステーション上の所在情報を格納するファイルである。

b) テーブル検索情報ファイル

本ファイルは、プログラムが参照しているプログラム間で共有のテーブルについて、テーブル構成図の所在（保管先のバインダー番号、バインダー内の頁番号等）、改造テーブルの履歴、及びソーステーブルが格納されているワークステーション上の所在情報を格納するファイルである。

c) LEDT・JCL検索情報ファイル

本ファイルは、制御用計算機において、プログラムを登録するために行う結合・編集～計算機登録や、タスク生成実行時のJCLが格納されているワークステーション上の所在情報、及び結合・編集情報の出力を行なった際の結合・編集結果と実行可能プログラムのCOMPARE^{*1}実行結果情報を格納するファイルである。

(2) データベースを作成するための入力情報

① 計算機内部情報の把握ツールで使用する情報

a) タスク間関連情報

タスク間関連についての基礎情報は、タスク一覧表、フローチャート、プログラムリスト等より、人間が次の情報を調査する。調査する情報は、タスクが発行しているタスク管

*1 プログラムを計算機に登録する際に作成される実行可能プログラムと計算機既存の実行可能プログラムとの比較。

理マクロは何か、及び、どのタスクに対して発行しているかという情報である。データベースへの入力、汎用計算機のTSS端末*1より行う。

b) 主プログラム毎の結合・編集情報

プログラムの構成、プログラムの従属関係、上位プログラム、テーブル参照関係の情報は、制御用計算機より出力した結合・編集情報をMTを介して汎用計算機に入力し、これを解析、編集することにより得る。

c) テーブル参照形式の情報

テーブル参照関係の情報のうち、プログラムがテーブルをどのような形式*2で参照しているかという情報は、プログラムリスト、フローチャート等より、人間が調査して得る。データベースへの入力、調査結果をTSS端末より行う。

② ドキュメント等の管理情報の把握ツールで使用する情報

a) プログラム検索情報ファイルの情報

プログラムリスト、フローチャートの所在（保管先のバインダー番号等）、改造プログラムの履歴、及びソースプログラムが格納されているワークステーション上の所在等を人間が調査して、TSS端末より入力する。

b) テーブル検索情報ファイルの情報

テーブル構成図の所在（保管先のバインダー番号、バインダー内の頁番号等）、改造テーブルの履歴及び、ソーステーブルが格納されているワークステーション上の所在等を人間が調査して、TSS端末より入力する。

c) LEDT・JCL検索情報ファイルの情報

プログラム登録JCLが格納されているワークステーション上の所在情報、及び結合・編集情報の出力を行った際の結合・編集結果と実行可能語のCOMPARE実行結果情報等を人間が調査し、TSS端末から入力する。

(3) 出力される帳票の種類と用途

汎用計算機のデータベースに集積された情報を利用するための各種帳票の種類と用途を以下に示す。

① 計算機内部情報の把握ツールで利用する帳票

a) タスク間関連情報リスト

本帳票は、タスク間の関連情報を集約し、計算機の系統単位毎にタスク番号の昇順に印字したものであり、「対象タスクに対しどのタスクから起動、停止等のタスク管理マクロが発行されているか」、「対象タスクからどのタスクに対しタスク管理マクロが発行されているか」といった情報が得られる。帳票出力例を第4.1.2 図に示す。

*1 汎用計算機に接続されているキャラクター入出力装置。

*2 READ：テーブルからの読み込み。

WRITE：テーブルへの書き込み。

RESERVE：テーブルの資源占有。

FREE：資源占有の開放。の4つのテーブル参照形式がある。

b) プログラム単位情報リスト

本帳票は、プログラムに関する情報を集約し、系統単位毎にプログラム略称のアルファベット順に出力したものであり、計算機共通プログラム、上位プログラム、従属プログラム、参照テーブル等の情報が印字されていて、「対象プログラムがどの上位プログラムからCALLされているか」、「対象プログラムがCALLしている従属プログラムや当該プログラムが参照しているテーブルは何か」、「対象プログラムが他のどの系統（全系制御設備は7台の計算機で構成されており、同一プログラムでも複数の系統で使用されているプログラムも存在する）で共通に使用されているか」といった情報が得られる。帳票出力例を第4.1.3 図に示す。

c) テーブル参照情報リスト

本帳票は、テーブルを参照しているプログラムの情報をテーブルを中心として集約し、テーブル略称のアルファベット順に印字したものであり、「対象テーブルを直接参照しているプログラムは何か」、「対象テーブルをどの様な参照形式で参照しているか」といった情報が得られる。帳票出力例を第4.1.4 図に示す。

d) プログラム構成図

本リストは、上位プログラム（タスク、常駐サブルーチン、非常駐サブルーチン）という結合・編集する際の最上位プログラムに対し、それに従属する全てのプログラム群の構成をTREE形式で印字したものであり、対象プログラムの全体構成を把握したい時、また従属プログラムが対象主プログラムの中でどの様に使用されているか、といった情報が得られる。リスト出力例を第4.1.5 図に示す。

e) プログラム単位テーブル参照情報リスト

本リストは、対象プログラムを中心にプログラムが参照しているテーブルの情報をプログラム略称のアルファベット順に印字したものであり、プログラムが参照しているテーブルとその参照形式を同時に得られる。リスト出力例を第4.1.6 図に示す。

タスク間関連情報リスト (系統=IBM)														PAGE: 14
発行元タスク					自タスク				発行先タスク					
タスク番号	タスク名	プログラム略称	タスクレベル	マクロ	タスク番号	タスク名	プログラム略称	タスクレベル	マクロ	タスク番号	タスク名	プログラム略称	タスクレベル	
0541	CTHPP5	CTHPP	5	QUEUE	0532	GTSET5	GTSET	5	QUEUE	020D	STIODT	STIODT	5	
0563	GTINT9	GTINT	3	QUEUE					QUEUE	0555	GTDRV5	GTDRV	3	
0208	SYSOP	SYSOP	5	ABORT					QUEUE	057D	GTAUT5	GTAUT	5	
0508	SYSOP	SYSOP	5	ABORT					ASUSP		ALL TASK			
0208	SYSOP	SYSOP	5	RLEAS					ARSUM		ALL TASK			
0508	SYSOP	SYSOP	5	RLEAS										
0530	CTHPP1	CTHPP	5	QUEUE	0533	GTSSM1	GTSSM	5	QUEUE	020D	STIODT	STIODT	5	
0558	GTINT1	GTINT	3	QUEUE					QUEUE	0551	GTDRV1	GTDRV	3	
0208	SYSOP	SYSOP	5	ABORT					ASUSP		ALL TASK			
0508	SYSOP	SYSOP	5	ABORT					ARSUM		ALL TASK			
0208	SYSOP	SYSOP	5	RLEAS										
0508	SYSOP	SYSOP	5	RLEAS										
053E	CTHPP2	CTHPP	5	QUEUE	0534	GTSSM2	GTSSM	5	QUEUE	020D	STIODT	STIODT	5	
055D	GTINT3	GTINT	3	QUEUE					QUEUE	0552	GTDRV2	GTDRV	3	
0208	SYSOP	SYSOP	5	ABORT					ASUSP		ALL TASK			
0508	SYSOP	SYSOP	5	ABORT					ARSUM		ALL TASK			
0208	SYSOP	SYSOP	5	RLEAS										
0508	SYSOP	SYSOP	5	RLEAS										
053F	CTHPP3	CTHPP	5	QUEUE	0535	GTSSM3	GTSSM	5	QUEUE	020D	STIODT	STIODT	5	
055F	GTINT5	GTINT	3	QUEUE					QUEUE	0553	GTDRV3	GTDRV	3	
0208	SYSOP	SYSOP	5	ABORT					ASUSP		ALL TASK			
0508	SYSOP	SYSOP	5	ABORT					ARSUM		ALL TASK			
0208	SYSOP	SYSOP	5	RLEAS										
0508	SYSOP	SYSOP	5	RLEAS										
0540	CTHPP4	CTHPP	5	QUEUE	0536	GTSSM4	GTSSM	5	QUEUE	020D	STIODT	STIODT	5	

第4.1.2図 タスク間関連情報リスト

プログラム単位情報	マルチプログラム	上位プログラム	従属プログラム	参照テーブル
系統略称 : 1A	系統 種別	略称 種別	略称 種別	略称 種別
プログラム略称: CSUBV プログラム種別: RSUB	1B RSUB 1BR RSUB 2BA RSUB 2BB RSUB	LTRECV TASK SSSNOW RSUB	PRIN RSUB	
プログラム略称: CTBGL プログラム種別: ISUB	1B ISUB 1BR ISUB 2BA ISUB 2BB ISUB	CAMAC TASK	CERR RSUB KSET ISUB	HBCTFA GLB
プログラム略称: CTERR プログラム種別: ISUB	1L RSUB 1B ISUB 1BR ISUB	STCERR TASK		HBCTFA GLB
プログラム略称: CTLERR プログラム種別: ISUB	1AM ISUB	UDG TASK	AMODE ISUB CSET RSUB SSCMOV RSUB WFLSB ISUB	
プログラム略称: CTRPRT	1B TASK		HTASC ISUB	CAMTRG GLB

第4.1.3図 プログラム単位情報リスト

テーブル参照情報リスト

テーブル略称: CGINHB		テーブル種別: GLB		READ	WRITE	RESERVE	FREE
参照プログラム	系統略称	プログラム略称	種別				
	1A	CSINTW	ISUB	*	*		
	1AM	CSINTW	RSUB	*	*		
	1AM	CSULKB	RSUB		*		
	1AM	CTVCHG	TASK	*			
	1AM	CTVINT	TASK	*	*	*	*
	1AM	PTEVLP	TASK				

テーブル略称: CGINTF		テーブル種別: GLB		READ	WRITE	RESERVE	FREE
参照プログラム	系統略称	プログラム略称	種別				
	1A	CSINTR	ISUB	*			
	1AM	CSCANC	ISUB				
	1AM	CSDREQ	ISUB				
	1AM	CTDHST	TASK	*			

第4.1.4図 テーブル参照情報リスト

プログラム構成図 (系統=1A)

MAIN_#--+LEVEL.01-#--+LEVEL.02-#--+LEVEL.03-#--+LEVEL.04-#--+LEVEL.05-#--+LEVEL.06

```

ETREMA :I--+BKSBYC :R
        +-CCTF :R
        +-ESPTSA :I---VRTLYC :R
        +-KSRQIS :R--+CCTF :R
        |         +-KRDSK :R
        +-LSPIDA :I--+AMODE :I--+ISFTL :R
        |         |         +-SSCMOV :R
        |         +-CCTF :R
        |         +-CSET :R
        |         +-SPACK :R
        |         +-SSALM :R--+CCTF :R
        |         |         +-MQUEUE :R
        |         +-XTAC :I
        +-SSALM :R--+CCTF :R
        |         +-MQUEUE :R
        +-SSCCNO :R
        +-SSENGI :R
        +-SSFLQI :R---FPW :I
        +-SSLFLC :I--+CCTF :R---TRNF :R---CCTF :R
        |         +-CCTF :R---TRNF :R---CCTF :R
        |         +-TRNSYC :R
        +-SSPSEQ :R
        +-SSTMCK :I
        +-TRNSYC :R
        +-VRTLYC :R
    
```

第4.1.5図 プログラム構成図

ACCOM	1L	T	CMMT	G	*	8	ANML06	1AM	I	CDNAME	B	*	8
ACTCOM	1BR	I	RGCOND	G	*	8	ANML06	1AM	I	C0SETB	B	*	8
ACTCOM	1BR	I	RG0BUF	G	*	4	ANML06	1AM	I	CGJOTZ	G	*	8
ACTCOM	1BR	I	RGREF	G	*	8	ANML06	1AM	I	CGOEVF	G	*	8
ACT1RW	1AM	I	ACT1	G	**	C	ANML06	1AM	I	DBTIDX	B	*	8
ACT2RW	1AM	I	ACT2	G	**	C	ANML07	1AM	I	CDAIND	B	*	8
AIDMSK	1A	I	HBCTFA	G			ANML07	1AM	I	CDDIND	B	*	8
AIDMSK	1B	I	HBCTFA	G	*	8	ANML07	1AM	I	CDNAME	B	*	8
AIDMSK	1BR	I	HBCTFA	G	*	8	ANML07	1AM	I	CGJOTZ	G	*	8
AIDMSK	2BA	I	HBCTFA	G			ANML07	1AM	I	CGOEVF	G	*	8
AIDMSK	2BB	I	HBCTFA	G			ANML07	1AM	I	DBANDB	B	*	8
AIMSK	1A	I	HBCTFA	G			ANML07	1AM	I	DBANTM	B	*	8
AIMSK	1B	I	HBCTFA	G			ANML07	1AM	I	DBAPID	B	*	8
AIMSK	1BR	I	HBCTFA	G	*	8	ANML08	1AM	I	CDAIND	B	*	8
AIMSK	2BA	I	HBCTFA	G			ANML08	1AM	I	CDDIND	B	*	8
AIMSK	2BB	I	HBCTFA	G			ANML08	1AM	I	CDNAME	B	*	8
ALMED	1AM	I	ALMCB	G			ANML08	1AM	I	CGJOTZ	G	*	8
ALMED	1AM	I	ALMSF	B			ANML08	1AM	I	CGOEVF	G	*	8

第4.1.6図 プログラム単位テーブル参照情報リスト

②ドキュメント等の管理情報の把握ツールで利用する帳票

a)プログラム検索情報リスト

本帳票は、プログラムについて、フローチャート、プログラムリストの所在（保管先のバインダー番号等）、改造プログラムの履歴、及びソースプログラムが格納されているワークステーション上の所在情報等がプログラムの略称によって得られる。帳票出力例を第4.1.7図に示す。

b)テーブル検索情報リスト

本帳票は、プログラムで参照されている各種テーブルについて、テーブル構成図の所在（保管先のバインダー番号、バインダー内の頁番号等）、改造テーブルの履歴、及びソーステーブルが格納されているワークステーション上の所在情報等がテーブル略称によって得られる。帳票出力例を第4.1.8図に示す。

c)LEDT・JCL検索情報リスト

本帳票は、制御用計算機において、主プログラムの結合・編集～計算機登録、及びタスク生成を行う際のJCLが格納されているワークステーション上の所在情報や、結合・編集を行った際の結合・編集結果及び、COMPARE結果情報が主プログラムの略称によって得られる。帳票出力例を第4.1.9図に示す。

プログラム検索情報リスト (NEWS) PAGE: 31

プログラム略称	リスト番号	系	種	地別	ページ番号	プログラム	ディレクトリ名	ファイル名	備考
LPRNT	003-140	1B	I		27	DTGNRT	/QA/SRCMSTR/P1B	LPRNT	
LPRNTR	005-005	1B	T		30		/QA/SRCMSTR/P1B	LPRNTR	KP:TO-LRNTR1
LPRNTS	003-143	1B	I			VLSET	/QA/SRCMSTR/P1B	LPRNTS	
LREAD	004-003	1AM	I		14		/QA/SRCMSTR/P1A	LREAD	
LRNTR1	005-005	1B	I		30		/QA/SRCMSTR/P1B	LRNTR1	KP(LPRNTR)
LSARS1	009-135	1A	I		5	LTCOMM	/QA/SRCMSTR/P1A	LSARS1	
LSARS2	009-136	1A	I		5	LTCOMM	/QA/SRCMSTR/P1A	LSARS2	
LSBSJM	009-143	1A	I				/QA/SRCMSTR/P1A	LSBSJM	
LSCTB	002-189	1B	I		13	STCNST	/QA/SRCMSTR/P1B	LSCTB	
LSEDT1	009-058	1A	I		5	LTTRNS	/QA/SRCMSTR/P1A	LSEDT1	
LSEDT2	009-089	1A	I		5	LTTRNS	/QA/SRCMSTR/P1A	LSEDT2	
LSEDT3	009-090	1A	I		5	LTTRNS	/QA/SRCMSTR/P1A	LSEDT3	
LSEDT4	009-091	1A	I		5	LTTRNS	/QA/SRCMSTR/P1A	LSEDT4	
LSEDT5	009-092	1A	I		5	LTTRNS	/QA/SRCMSTR/P1A	LSEDT5	
LSEDT6	009-093	1A	I		5	LTTRNS	/QA/SRCMSTR/P1A	LSEDT6	
LSERR2	009-118	ETC	I		5	LTTRNS	/QA/SRCMSTR/P1A	LSERR2	
LSERS1	009-119	1A	I		5	LTABMN	/QA/SRCMSTR/P1A	LSERS1	
LSERS2	009-120	1A	I		5	LTABMN	/QA/SRCMSTR/P1A	LSERS2	
LSERS3	009-121	1A	I		5	LTABMN	/QA/SRCMSTR/P1A	LSERS3	
LSFMT2	009-124	1A	I		5	LT1BDS	/QA/SRCMSTR/P1A	LSFMT2	
LSMENT	009-150	1A	I				/QA/SRCMSTR/P1A	LSMENT	
LSDFLC	009-153	1A	I				/QA/SRCMSTR/P1A	LSDFLC	
LSDPND	009-115	1A	R				/QA/SRCMSTR/P1A	LSDPND	
LSPIDA	009-116	1A	I				/QA/SRCMSTR/P1A	LSPIDA	

第4.1.7図 プログラム検索情報リスト

テーブル検索情報リスト (NEWS)

テーブル略称	リスト番号	グループ名	ページ	ディレクタリ名	ファイル名	備考
EDBACL	105-010	MC	9-	/QA/SRCMSTR/TMC	EDBACL	
EDDVSG	105-017	MC	10-	/QA/SRCMSTR/TMC	EDDVSG	
EDEGNO	105-018	MC	11-	/QA/SRCMSTR/TMC	EDEGNO	
EDFCPT	105-070	FC	2-	/QA/SRCMSTR/TFC	EDFCPT	
EDG	-	-	-	/QA/SRCMSTR/TUDG2S	EDG	G UDG
EDGEID	-	-	-	/QA/SRCMSTR/T		X
EDGFID	105-146	MC	86-	/QA/SRCMSTR/TMC		X 62. 7.31-5 KJ (A)
EDGPID	105-146	MC	85-	/QA/SRCMSTR/TMC		X 62. 7.31-5 KJ (A)
EDHOSY	105-012	MC	12-	/QA/SRCMSTR/TMC	EDHOSY	
EDICT	105-093	MC	87-	/QA/SRCMSTR/TMC	MTCR	
EDMCTA	105-090	MC	13-	/QA/SRCMSTR/TMC		X 62. 7.31-3 KJ (A)
EDN	-	-	-	/QA/SRCMSTR/TUDG2S	EDN	G UDG
EDNFID	105-147	MC	84-	/QA/SRCMSTR/TMC		X 62. 7.31-5
EDN11D	105-147	MC	83-	/QA/SRCMSTR/TMC		X 62. 7.31-5
EDT	-	-	-	/QA/SRCMSTR/TUDG2S	EDT	G UDG
EDTFID	105-110	MC	82-	/QA/SRCMSTR/TMC		X
EDTLID	105-011	MC	14-	/QA/SRCMSTR/TMC	EDTLID	

第4.1.8図 テーブル検索情報リスト

LEDT JCL 検索情報リスト (NEWS)

プログラム略称	リスト番号	系統	種別	LEDT	COMP結果	ディレクタリ名	ファイル名	備考
CCLNK	001-168	1BR	R		0	/QA/SRCMSTR/L1BR	CCLNK03	
CCLNX	011-021	1B	R		0	/QA/SRCMSTR/L1B	CCLNK04	
CCLVX	001-200	2B	R		0	/QA/SRCMSTR/L2B	CCLNK06	
CCLNK	001-200	2BB	R		0	/QA/SRCMSTR/L2B	CCLNK06	
CCSB11	007-033	1L	R		X	/QA/SRCMSTR/L1L	CCSB11	
CCSB12	007-034	1L	R		X	/QA/SRCMSTR/L1L	CCSB12	
CCSB13	001-007	1L	R		X	/QA/SRCMSTR/L1L	CCSB13	
CCSB14	001-008	1L	R		0	/QA/SRCMSTR/L1L	CCSB14	
CCSB15	001-009	1L	R		0	/QA/SRCMSTR/L1L	CCSB15	
CCSB18	001-010	1L	R		0	/QA/SRCMSTR/L1L	CCSB18	
CCSB19	001-011	1L	R		0	/QA/SRCMSTR/L1L	CCSB19	
CCSB2	007-031	1L	R		0	/QA/SRCMSTR/L1L	CCSB2	
CCSB20	001-021	1L	R		0	/QA/SRCMSTR/L1L	CCSB20	
CCSB21	001-012	1L	R		0	/QA/SRCMSTR/L1L	CCSB21	
CCSB22	001-013	1L	R		0	/QA/SRCMSTR/L1L	CCSB22	
CCSB23	001-014	1L	R		0	/QA/SRCMSTR/L1L	CCSB23	
CCSB27	001-015	1L	R		0	/QA/SRCMSTR/L1L	CCSB27	
CCSB5	001-028	1L	R		X	/QA/SRCMSTR/L1L	CCSB5	
CCSB9	007-032	1L	R		0	/QA/SRCMSTR/L1L	CCSB9	
CCTF	006-092	1A	R		0	/QA/SRCMSTR/L1A	CCTF01	
CCTF	011-068	1AM	R		C	/QA/SRCMSTR/L1AM	CCTF02	
COINT	-	PM	T			/QA/SRCMSTR/LPM		

第4.1.9図 LEDT・JCL 検索情報リスト

(4) 支援システムの有効性

① 計算機内部情報の把握ツール

本システムは、主プログラム単位の結合・編集情報を基に作成されたもので、プログラムより下位の情報と上位の情報を得ることが出来る。本システムを使わずに情報を調査する場合、プログラムより下位の情報については、制御用計算機からリスト出力される結合・編集情報を調査すれば情報が得られるが、調査に要する時間は本システムを用いる場合に比べて膨大となる。また、プログラムより上位の情報については、結合・編集情報には出力されず、調査をすると非常に時間が費やされ、かつ調査ミスも改造対象プログラムの数に比例して増えると想定される。この様に本システムを用いることで、ソフトウェアの開発・改造作業において、必要な情報を迅速かつ正確に把握することが出来た。

② ドキュメント等の管理情報の把握ツール

本システムは、ソフトウェアの開発・改造作業において、最も有効に利用されたシステムといえる。ソフトウェアの開発・改造作業に必要なドキュメント（フローチャート、テーブル構成図、プログラムリスト）及びソースデータ（ソースプログラム、ソーステーブル、プログラム登録JCL）の所在が明確になっていることで、それらドキュメントを探す時間が大幅に減少した。特に、我々が扱っている様な大規模な計算機システムでは効果があった。また、所在検索だけでなく改造を実施した場合の履歴も記録されているので、不具合が発生した場合の調査に役立てることも出来た。

(5) 支援システムの問題とその対策

① 計算機内部情報の把握ツール

このシステムの問題は、情報の一部（タスク間相互関係情報，テーブル参照関係情報）は、人間が調査しなければ得られず、多大な時間を費やしてしまうといった点と、本システムは汎用計算機上に構築されているため、制御用計算機の出力情報を一旦MTに出力して汎用計算機に出力するといった人間が介入する手続きがある点である。

これらの問題に対しては、現状のシステムと全情報が抽出可能なシステムを制御用計算機上に構築することが理想とされる。最新の制御用スーパーミニコンピュータやワークステーションでは、この様なことが可能であろう。しかし、現システムの制御用計算機（1978年に製作されたミニコンピュータ）では、容量的に余裕がないことと、そのソフトウェアを開発するには多大な費用と時間が必要となるため開発は無理な状況となっている。この様な状況で解決策を検討した結果、情報の調査をプログラム製作担当者が的確に実施することで、情報調査時間の短縮や信頼性の向上が期待出来る。

② ドキュメント等の管理情報の把握ツール

本システムの問題は、ソフトウェアの開発・改造作業により、データベースの更新作業や帳票の出力及び、整理作業が面倒という点である。現状の更新作業は、変更になった情報を抽出し、現状の帳票に変更内容を書き込み、その帳票を見ながらデータベースを更新し、帳票を出力して旧帳票（現状の帳票に変更内容の書き込みを行った帳票）と差し替えるといった内容である。作業は単純であるが、変更内容が多い場合、工数を要する作業となる。

この問題に対しては、更新作業の方法を変えることで対処する。今までの更新作業は、一つのソフトウェアの開発・改造プロジェクト毎に一連の更新作業（変更内容の書き込みから旧帳票の差替えまで）実施していたが、変更内容を書き込みまでをプロジェクト毎に行い、データベースの更新作業、帳票出力及び、帳票の差替え作業は、適当な時期（5プロジェクトに一度または、～3年に一度）を見て一括して行う。こうすることで、データベース更新以降の作業効率が向上する。

4.2 ドキュメント管理システムとその効果

(1) 管理対象ドキュメント

本システムで対象としているドキュメントの種類を以下に示し、ドキュメントの内容をに説明する。

① 設計図書

機能設計書，システム設計書，設計計算書，各種ハード関連の図面等を示す。

② フローチャート

主にフローチャートだけであるがタスクの場合は、フローチャートの先頭にフローチャートの処理概要書が付けられ、サブルーチンの場合は、フローチャートの先頭にサブルーチン仕様書が付けられている。

③プログラムリスト

ソースプログラムを翻訳してリスト出力させたものである。

④テーブル構成図

データの構成、用途、定数値(定数テーブルの場合)等が示されたものである。

⑤タスク一覧表

計算機に登録されているタスク情報の(タスク略称、プログラム名称、タスク起動時の優先レベル、コアブロック番号*1といった計算機登録時の情報)をタスク番号の昇順に記述したものである。

⑥サブルーチン仕様書

計算機に登録されているサブルーチンの仕様を記述したものである。

⑦ソフトウェア開発・改造管理シート

ソフトウェア開発・改造管理シート(以降、「管理シート」という)は、ソフトウェアの開発・改造担当者が、開発・改造作業を行う前に作成するシステムの概要、作業計画、変更予定のドキュメント、ソースデータが何であるか等を記述した「開発・改造の要求シート」と、開発・改造作業が完了した後に作成する実際に行った改造内容と新規作成、修正、削除を行ったドキュメント、ソースデータの一覧を記述した「プロジェクト完了報告書」で構成されている図書である。

⑧ソフトウェアの開発・改造における書き込み図書(旧版図書)

本システムの管理対象ドキュメントに対して、ソフトウェアの開発・改造において生じた修正内容を改造前のドキュメントに書き込みを行ったものである。修正が大規模等で書き込み図書で対処出来ない場合は、旧版の図書(修正前の図書)を管理する。

(2) 各ドキュメントの管理体系と検索

本システムで管理しているドキュメントは、総て各図書の最優先訂正版をマスター図書、即ち、原図として扱って管理しており、ソフトウェアの開発・改造作業におけるドキュメントの修正は、この図書に対して実施する。また、各ドキュメントの保管は、プログラムリストを除いて、全てA4サイズのバインダーで行うこととした。以下に各ドキュメントの管理体系と検索方法を述べる。

①設計図書

設計図書は、各機能に分類して管理している。機能の分類は、大きく分けると3つ(ハードウェア関連、ソフトウェア関連、一般)に分類され、さらに細かく分類(各計算機の機能に対応した)している。分類された各機能には、機能分類コードを設けてドキュメントの検索に利用されている。

設計図書の所在を検索する場合は、機能分類コードによる方法と設計図書に付されている整理番号(以降で述べる)による方法、及びバインダー番号による検索がある。それらは総

*1 主記憶に常駐しないタスクが主記憶上に読み込まれる位置を示した番号である。尚、この番号は、計算機システムの初期設定時に、適当な容量(1ブロックあたり1~4kw)に分割されていて、タスク登録時に読み込み位置の指定を行う。

て、「設計図書の目録」*1に管理されている整理番号主体の一覧表、機能分類別一覧表、バインダー別一覧表を利用する。ソフトウェアの開発・改造作業等で設計図書が追加及び削除された場合は、「設計図書の目録」の各種一覧表及び設計図書のバインダーの目次等を更新し、設計図書を該当するバインダーに保管(削除)する。

②フローチャート(GFC, 処理設計書)

フローチャートは、各機能に分類して管理している。機能の分類は、①の設計図書と同様である。

フローチャートの所在を検索する場合は、機能分類した機能名称より検索する方法とプログラム略称より検索する方法がある。機能名称より検索する方法は、プログラムに該当しそのような機能名称を探し、バインダー番号を抽出してそのバインダーに保管されているフローチャートを探す。プログラム略称より検索する方法は、「プログラム検索情報リスト」を利用して検索する。ソフトウェアの開発・改造作業等でフローチャートが追加及び削除された場合は、「プログラム検索情報リスト」及び、フローチャートのバインダーの目次等を更新し、フローチャートを該当するバインダーに保管(削除)する。

③プログラムリスト

プログラムリストは、リストバインダーにとじて管理している。プログラムリストの所在検索は、「プログラム検索情報リスト」を利用して行う。ソフトウェアの開発・改造作業等でプログラムが追加及び削除された場合は、「プログラム検索情報リスト」を更新し、プログラムリストを該当するバインダーに保管(削除)する。

④テーブル構成図

テーブル構成図は、計算機の登録先別、及び定数テーブルとワーク(作業用)テーブルに分類して管理している。

テーブル構成図の所在検索は、「テーブル検索情報リスト」を利用して行う。ソフトウェアの開発・改造作業等でテーブルが追加及び削除された場合は、「テーブル検索情報リスト」及びテーブル構成図のバインダーの目次等を更新し、テーブル構成図を該当するバインダーに保管(削除)する。

⑤タスク一覧表

タスク一覧表は、計算機の系統別に分類して管理している。タスク一覧表が保管されているバインダーには、系統別にインデックス(見出し)をつけて検索を容易にしている。

タスク一覧表の情報検索は、バインダーの系統略称インデックス以下に存在する頁から、検索対象のタスクを探していく。尚、各タスクは、計算機に登録されているタスク番号の昇順に記載されているので、タスク番号が分かっている場合は容易に検索出来る。ソフトウェアの開発・改造作業等でタスクが追加、削除及び、計算機登録内容が変更された場合は、タスク一覧表の内容を更新する。

*1 設計図書の管理台帳。

⑥ サブルーチン仕様書

サブルーチン仕様書は、計算機の系統別またはサブシステム別に分類して管理している。サブルーチン仕様書のバインダーには、系統またはシステムのインデックスをつけて検索を容易にしている。

サブルーチン仕様書の所在を検索する場合は、索引が無いのでプログラム略称より「プログラム検索情報リスト」を利用して系統を抽出し、その系統のインデックス以下の頁を一つ一つ探していく。ソフトウェアの開発・改造作業等でサブルーチンが削除された場合は、サブルーチン仕様書のバインダーより対象頁を削除する。尚、追加された場合、最近ではサブルーチン仕様書をフローチャートの先頭に付けて納入しているので、そのままフローチャートと一緒に保管している。

⑦ ソフトウェア開発・改造管理シート

管理シートは、開発・改造作業前に作成した「開発・改造の要求シート」と開発・改造作業完了後に作成した「プロジェクト完了報告書」を一諸にして管理している。

管理シートの所在を検索する場合は、「世代管理台帳*1」に管理シートの一覧表があり、一覧表に記述されているバインダー番号とインデックス番号より所在を検索する。ソフトウェア開発・改造により、管理シートを作成した場合は、「世代管理台帳」と管理シートが保管されているバインダー内の目次に内容を追記し、管理シートを保管する。

⑧ ソフトウェアの開発・改造における書き込み図書

本図書は、ソフトウェアの開発・改造作業のプロジェクト毎に管理している。本図書を保管する場合は、「世代管理台帳」の一覧表と保管先のバインダーの目次に内容を記入し保管する。

本図書を検索する場合は、「世代管理台帳」に書き込み図書一覧があるので、一覧表の記述情報よりドキュメントの検索を行う。

尚、本図書は、開発・改造作業が発生す毎に増えるので、古いプロジェクトの図書については時期を見て廃棄している。

(3) ドキュメント修正の規定

ドキュメントを修正する際は、必ずマスター図書をコピーしてコピーに対して書き込み修正を行い、システムが収束したら書き込み修正の内容を基にマスター図書を修正する。マスター図書のコピーへの書き込みは、鉛筆（修正が可能な筆記用具）を使用して修正箇所を雲型マークで囲んで分かり易くする。

マスター図書を修正する場合は、原図に対して修正液等を使用して修正する。また、表紙の来歴欄に来歴（改訂番号（一回目の修正を1とし以降一つずつ更新していく）、プロジェクト名（開発・改造の件名）、修正者、プロジェクト名の略号（プロジェクト名をアルファベットで省略した記号）を記入し、さらに修正した各頁に改訂番号とプロジェクト名の略号を記入する。

*1 旧版図書の管理台帳。

(4) 設計図書の整理番号

設計図書を管理する上で必要となる整理番号は、付番基準に従って全図書に対して行い「設計図書の目録」に登録する。

(5) 管理システムの有効性

本管理システムの有効なことは、最新版のドキュメントが整理されていることと、必要なドキュメントの所在が容易に検索出来ることである。ドキュメントを整理し、所在検索が容易であることは、一見当り前のことの様であるが、我々の経験では管理されていない計算機システムが多く、時間的にも作業上においても色々と支障が生じることが多い様である。本システムを用いたことによりこの様な不都合が生じなかった。

また、書き込み図書を管理していることにより、改造作業が終わり運転再開後に発生した不具合の対策において、何処をどう変更したかが明確であったため、不具合の原因が速く解明出来たこともある。

(6) 管理システムの問題点

本管理システムの問題は、ドキュメントの修正方法（当初のドキュメント修正方法は、修正箇所を丁寧に示し、各修正部分に修正を実施した日付、担当者名、プロジェクト名を記述するといった規定であった）の規定が細か過ぎるということや、変更になった情報の更新作業が面倒であるといった点である。

4.3 ソースデータ管理システムとその効果

(1) システム全体構成

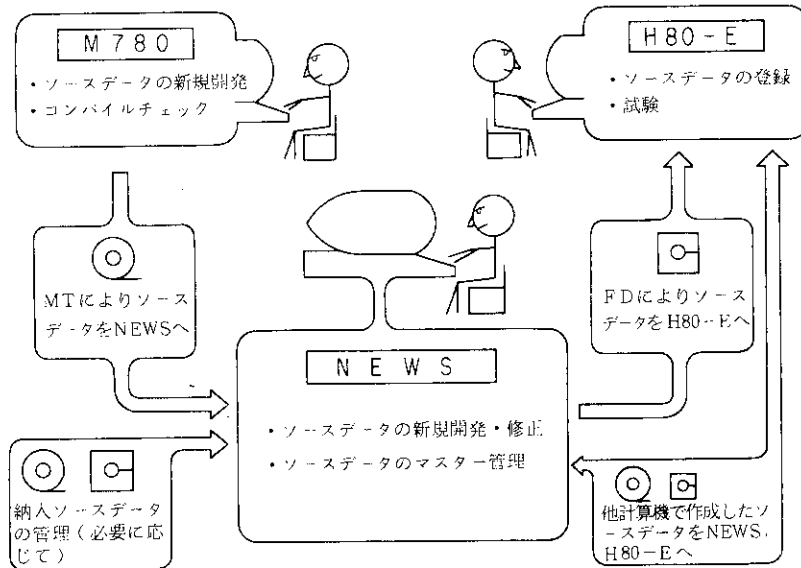
本システムは、プログラム、テーブル、プログラムの結合・編集～計算機登録実行時に使用する JCL をワークステーション上で管理するものであり、ソースデータの修正、新規開発を可能とする。

ソースデータの開発・改造作業で、ワークステーションを使用して作成されたソースデータは、フロッピーディスク（以降、「FD」という）を介して制御用計算機側で処理される。

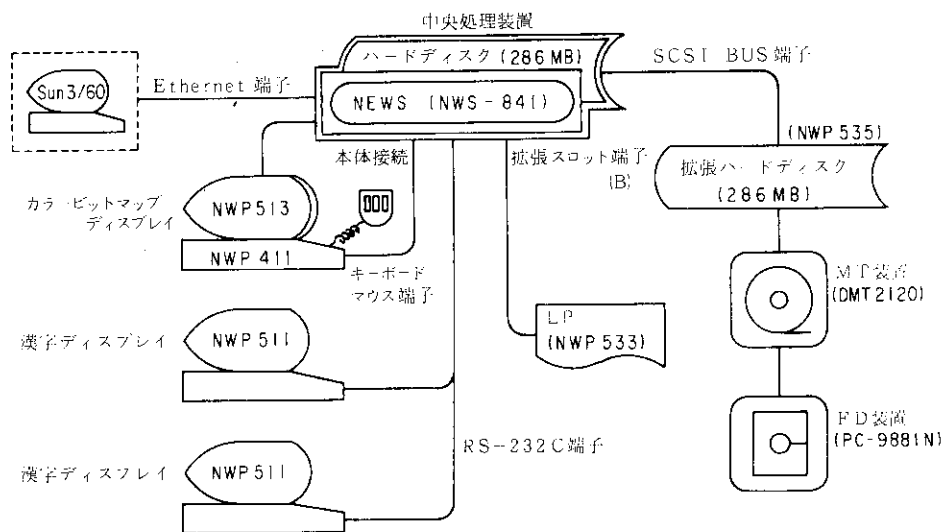
また、ソースデータの新規開発時には補助的な手順として、汎用計算機でも実施することが可能だが、この場合作成されたソースデータは、MTを介してワークステーション側に転送する。第4.3.1 図に本管理システムを用いたプログラム開発作業の関連についてその全体構成を示す。

(2) ワークステーションのハードウェア構成

ワークステーションは、(株)ソニー製のNEWS (NWS-841) で中央処理装置を中心にUNIXというオペレーティングシステムの下で動作する複数（現在3台）の端末処理装置で構成され、周辺装置として、IBMフォーマット対応の8インチFD装置及びMT装置、さらにレーザービームプリンタ（以降、「LP」という）が接続されている。第4.3.2 図にハードウェア構成を示す。



第4.3.1図 システム全体構成

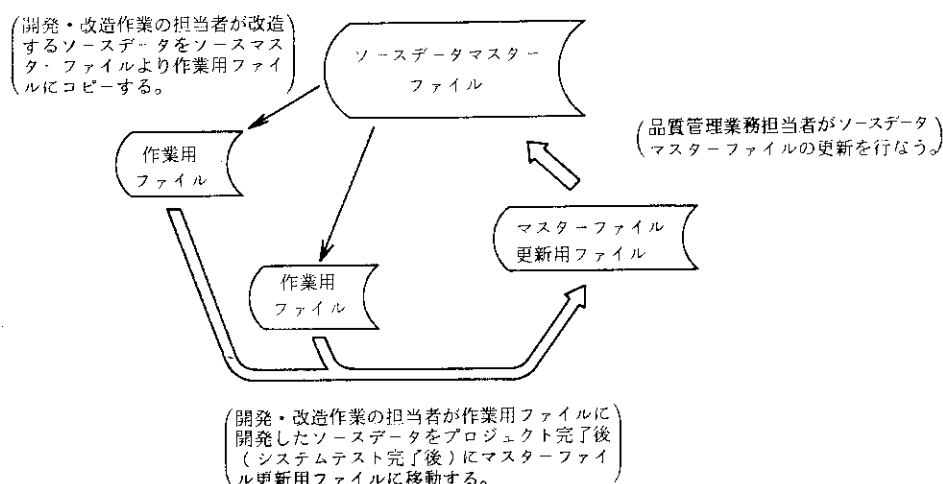


第4.3.2図 ワークステーションのハードウェア構成

(3) ソースデータの管理体系

ソースデータは、ワークステーションのディスクファイル上で管理される。このディスクファイルでは、それらを計算機の系統別に分類してソースデータマスターファイル上のマスターソースデータとして管理する。

ソースデータの修正を行う際は、ソースデータマスターファイルからソースデータを作業用ファイルにコピーして、作業用ファイル上で修正する。ソースデータマスターファイルの更新は、ソフトウェア開発・改造作業のプロジェクト毎に、プロジェクトが完了した時点で品質管理業務担当者が行う。ソースデータマスターファイルのアクセスは、ソースデータ管理用ユーザ（品質管理業務担当者のみが更新可能で、その他の者は参照のみ可能）とソースデータ開発・改造用ユーザ（ソフトウェアの開発・改造作業担当者であれば誰でも更新可能）の2つのユーザに限りアクセスを可能とする。第4.3.3 図にソースデータの管理体系を示す。



第4.3.3図 ソースデータの管理体系

(4) ソースデータの検索

ソースデータが格納されている所在検索は、各種検索情報リスト（「プログラム検索情報リスト」，「テーブル検索情報リスト」，「LEDT・JCL検索情報リスト」）により行う。各種検索情報リストの詳細については、4.1節を参照のこと。

(5) 本管理システムの有効性

本管理システムで有効な点は、当初の紙カードでの管理体系に比べて、ソースデータの所在検索、ソースデータの修正、プログラム及びテーブルの計算機登録の手間、磁気媒体になったことでの耐久性等、あらゆる面で使い易くなり、各作業が時間的に短縮されことは特筆すべき点である。

(6) 管理システムの問題とその対策

本管理システムの問題は、ワークステーションで開発・改造されたプログラム及びテーブルを登録する際にFDを介さなければならず時間や手間がかかるという点と、ワークステーションと制御用計算機間でのデータの受渡し（ワークステーションの標準入出力コマンドを使用した場合）の際に、化けてしまう文字があるという点であった。

前者の問題については、ワークステーションと制御用計算機をオンラインで結ぶことで、プログラム、テーブルの翻訳作業や計算機登録等の作業に費やされる時間が、かなり減少出来ると考えられる。

後者の問題については、ワークステーション内で変化した文字を補うプログラムを組み込んで対処を行った。

5. まとめ

本品質管理で対象となったJT-60全系制御設備計算機システムのソフトウェアは、運用前2年間の試験を行ったため、運用当初の段階でかなり高い品質を有していた。その後、実験装置の制御用ソフトウェアである点から、その開発・改造作業が重ねられたが、本品質管理が行われたことで、その品質を低下させることなく実験運転が進められたと考えられる。

今後もこの品質管理を改善して続けることで、最適なソフトウェアの開発・改造作業の確立を志向するものである。

謝 辞

JT-60制御系におけるソフトウェアの品質管理手法の構築にあたって、初期の開発を含めて多大な便宜を図って頂いたJT-60第2試験室長 近藤育朗氏に感謝致します。また、本品質管理手法の開発及びその適用に際して、様々の協力を頂いたJT-60第1試験室制御グループの方々に感謝の意を表します。さらに本作業を遂行するにあたり、JT-60第1試験室長当時より適切な助言をして下さったJT-60試験部次長 白形弘文氏、並びに、終始御支援賜りましたJT-60試験部長 田中裕二氏に感謝致します。

参考文献

- (1) 菅野文友, 大川幹彦, 片岡雅憲, 額田啓三, 萬木敏昭: "ソフトウェアデザインレビュー 日本的デザインレビューへのアプローチ", 日科技連出版社, 東京 (1982).
- (2) Glenford J. Myers: "Software Reliability-Principles and Practices", 有澤誠 訳, "ソフトウェアの信頼性 ソフトウェアエンジニアリング概説", 近代科学社, 東京 (1977).
- (3) 中原啓一, 三次衛 共編: "新編 システムエンジニアハンドブック", オーム社, 東京 (1982).

5. まとめ

本品質管理で対象となったJT-60全系制御設備計算機システムのソフトウェアは、運用前2年間の試験を行ったため、運用当初の段階でかなり高い品質を有していた。その後、実験装置の制御用ソフトウェアである点から、その開発・改造作業が重ねられたが、本品質管理が行われたことで、その品質を低下させることなく実験運転が進められたと考えられる。

今後もこの品質管理を改善して続けることで、最適なソフトウェアの開発・改造作業の確立を志向するものである。

謝 辞

JT-60制御系におけるソフトウェアの品質管理手法の構築にあたって、初期の開発を含めて多大な便宜を図って頂いたJT-60第2試験室長 近藤育朗氏に感謝致します。また、本品質管理手法の開発及びその適用に際して、様々の協力を頂いたJT-60第1試験室制御グループの方々に感謝の意を表します。さらに本作業を遂行するにあたり、JT-60第1試験室長当時より適切な助言をして下さったJT-60試験部次長 白形弘文氏、並びに、終始御支援賜りましたJT-60試験部長 田中裕二氏に感謝致します。

参考文献

- (1) 菅野文友, 大川幹彦, 片岡雅憲, 額田啓三, 萬木敏昭: "ソフトウェアデザインレビュー 日本的デザインレビューへのアプローチ", 日科技連出版社, 東京 (1982).
- (2) Glenford J. Myers: "Software Reliability-Principles and Practices", 有澤誠 訳, "ソフトウェアの信頼性 ソフトウェアエンジニアリング概説", 近代科学社, 東京 (1977).
- (3) 中原啓一, 三次衛 共編: "新編 システムエンジニアハンドブック", オーム社, 東京 (1982).

5. まとめ

本品質管理で対象となったJT-60全系制御設備計算機システムのソフトウェアは、運用前2年間の試験を行ったため、運用当初の段階でかなり高い品質を有していた。その後、実験装置の制御用ソフトウェアである点から、その開発・改造作業が重ねられたが、本品質管理が行われたことで、その品質を低下させることなく実験運転が進められたと考えられる。

今後この品質管理を改善して続けることで、最適なソフトウェアの開発・改造作業の確立を志向するものである。

謝 辞

JT-60制御系におけるソフトウェアの品質管理手法の構築にあたって、初期の開発を含めて多大な便宜を図って頂いたJT-60第2試験室長 近藤育朗氏に感謝致します。また、本品質管理手法の開発及びその適用に際して、様々の協力を頂いたJT-60第1試験室制御グループの方々に感謝の意を表します。さらに本作業を遂行するにあたり、JT-60第1試験室長当時より適切な助言をして下さったJT-60試験部次長 白形弘文氏、並びに、終始御支援賜りましたJT-60試験部長 田中裕二氏に感謝致します。

参考文献

- (1) 菅野文友, 大川幹彦, 片岡雅憲, 額田啓三, 萬木敏昭: "ソフトウェアデザインレビュー 日本的デザインレビューへのアプローチ", 日科技連出版社, 東京 (1982).
- (2) Glenford J. Myers: "Software Reliability-Principles and Practices", 有澤誠 訳, "ソフトウェアの信頼性 ソフトウェアエンジニアリング概説", 近代科学社, 東京 (1977).
- (3) 中原啓一, 三次衛 共編: "新編 システムエンジニアハンドブック", オーム社, 東京 (1982).