

## 解説



## ● 大規模ソフトウェア・プロダクトのライン管理†

本池 洵††

## 1. はじめに

一般の商用システムとしては、典型的な大規模ソフトウェア・プロダクトといえる。航空座席予約システムの開発作業に参画従事した。この開発過程を振り返って、ソフトウェア開発時の工夫、特にプロダクトライン管理の方法とこれを支える諸工夫について述べる。ここに述べる諸工夫は、航空座席予約システム(オンラインシステム)の開発において採用したものとすれば、本邦初の試みである。原始コードの開発からテスト、システムの生成への作業工程を円滑に進めるための各種支援ツールと支援環境の考え方は、ほかのオンラインシステムの開発においても採用できるものと考えられる。

## 2. システムの開発保守の現状と対策

システムの開発時には、そのシステムの開発期間および稼動時間を含めた長期に亘るニーズを可能な限り予測し設計する必要がある。しかしながら、システムの長期間の利用と多様化する環境の変化などのため、

- ① データ量の増大
- ② 新機能の必要性
- ③ 外界の変化(企業内政策、監督官庁の指導など)
- ④ その他(要求仕様の欠陥、実現の不備など)

が生じ、その結果、仕様変更を避けることができないことも事実である。しかし、システムの再開発は不経済であり極力回避したいので、設計その他を工夫しなければならない。我々は開発当初よりソフトウェア・プロダクトの維持管理機能についても考慮し、その方式と環境を設定した。この維持管理機能の骨子は、

- ① 運用システムとともに、これを改善強化する開発中のシステムの共存を認め両者を取扱うことができること。

② 運用システムの稼動および開発中のシステムの開発と検査が、同一コンピュータ・システムかつ同一操作環境の下で、実施可能であること。

③ 開発システムの開発および検査完了後、それを手直しすることなく、運用システムとして実用に供することができること、である。これが次章に述べる段階的開発作業を支える重要な機能になる。

## 3. 大規模システムの段階的開発

システムは通例、小さなモジュールの階層構造として構築され、徐々に目標とするシステムに近づいていくが、計画されたすべての機能要件を満たさなくても限定した利用範囲内での運用ができるシステムの早期実現が強く望まれている。このような考え方に従って、ソフトウェアを開発すると、本来必要とする完全なシステムへ接近する多数のシステム(システムの版(Version))が存在することとなる。当然これらの版を支えるモジュール群も爆発的に増加する。このような状況への対応は、計画の初期段階から用意しておかねばならない。開発要員および開発に要する機械、時間など有限な資源の有効利用と、開発済部分の品質把握を行うために、開発すべき機能の優先順位および開発期間の設定をした。開発期間の単位は、通常1カ月とし、単位期間ごとに総合作動確認ツールによる評価作業を実施した。この方法は、評価結果を開発作業に逐次反映させ、かつその作業範囲を拡大させることなく、円滑な作業推進を可能とした。なおこの開発と評価の同時並行作業は、次節に具体的に述べる維持管理機能により可能である。またシステム運用開始後の現在においても、この機能を利用し新しいニーズに対応する新機能の開発および検査を実施し、システムの改善を行っている。

## 4. 維持管理機能とその環境

一般の航空座席予約システムの開発において採用し

† Line Control for Large Scale Software Product by Makoto MOTOIKE (Nippon Univac Kaisha, LTD.).

†† 日本ユニパック(株)

た各種技法および設定した環境について述べる。

### 4.1 原始コード管理システム

多数要員による、多数のプログラム・モジュールの開発作業を、円滑に推進するために、すなわち開発要員一人一人の作業能率を最大限に向上させかつ多数のシステムの管理を可能とするために、開発開始時より開発支援技法について検討を重ね、原始コード管理システム (Source Code Control System-SCCS) の開発を行った。このシステムは、日本ユニパックのシリーズ 1100 の基本ソフトウェアにおける開発・維持管理支援ツールである SMART (Systems Maintenance And Reporting Tool) を基に、吟味拡張を計ったものである。

このシステムの機能は、

- ① 多数の開発要員が同時に、他要員とかかわりなく独自の開発作業を実施できること。
  - ② 複数の開発場所 (たとえば日本と米国) で同時開発を可能とし、国際分業および開発機能の相互利用ができること。
  - ③ 開発コードおよび付属情報の履歴の累積保持ができ、検索可能であること。
  - ④ 機能ごとに、システムへの組み入れ、取り外しが可能であること。
  - ⑤ 開発コードの組み入れモジュールに対応し、関連モジュールのコンパイルあるいはアセンブルおよび連結編集プログラムを起動させるための JCL を自動作成すること。
  - ⑥ システム生成が可能であること。
  - ⑦ システム生成単位 (システム・レベル) ごとに、作成日、組み入れコード数、不具合回避コード件数などの統計情報が把握可能であること。
  - ⑧ システムを構成するモジュール数およびモジュールごとの原始コード数が把握可能であること。
  - ⑨ 開発作業と検査作業が、同じ環境で継続して実施可能であること、
- である。これら機能を実現するために、次のように考えた。

#### 4.1.1 システム生成

システム生成の要素として以下に述べる要素がある。SCCS によるシステム生成と要素の関連を図-1 に示す。

① 基本原始コードファイル (Base Symbolics Image File-SI)

プログラム・モジュールの原始コードの集合で、そ

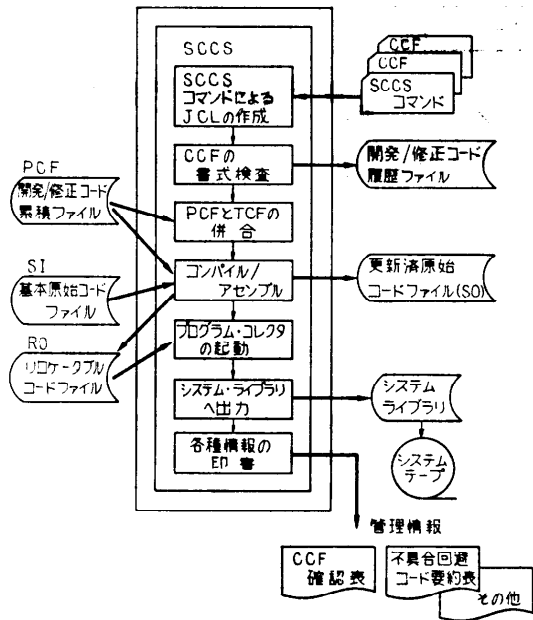


図-1 システム生成機能関連図

の内容は一定期間不変である。このことにより、原始コードごとに付加された文番号は不変であり、新規開発コードあるいは修正コードは、原始コードの文番号を見出しとして開発する。これは、原始コードを直接修正しないで、修正/開発の履歴を保存し、どの修正/開発時点のコードでも修復可能とする。

② 開発/修正コード累積ファイル (Permanent Correction File-PCF)

前項 SI の各モジュールに対応し、原始モジュールの文番号を見出し部とする開発/修正コードが、モジュールごとに累積されている。このファイルは、SI が固定している期間中累積される。PCF の例を表-1 に示す。この表は、プログラム・モジュール名 "B8 F5 FM" の PCF である。左欄はモジュール名と PCF のコードに対応するコード番号を示す。マイナス符号付きの数字は、"B8 F5 FM" の基本原始コードの文番号に対応し、たとえば -38 は、基本原始コードの文番号 38 と 39 の間に 2 つの PARAMETER 文を挿入することを意味し、-128, 128 は文番号 128 の文を削除し NO=BD8R NO-1 の文を挿入することを意味する。

③ 開発/修正コードファイル (Temporary Correction File-TCF)

システム生成時における開発/修正コードの集合フ

表-1 モジュール“B8F5FM”のPCF

B8F5FM		SSG PERMANENT STREAM NEW	
B8F5FM		***	
000001			B8F5FM NH-TKTF-451-00002 TICKET NH-R3-007-00
000002	000001	-38	PARAMETER DBG = 0
000003	000002		PARAMETER DRCNT = 2000
000004		-43	B8F5FM NH-TKTF-451-00001 TICKET NH-R3-004-00
000005	000001		INCLUDE RESLIBV.BBFREG
000006	000002		INCLUDE RESLIBV.BDFREG
000007		-55	B8F5FM NH-TKTF-451-00002 TICKET NH-R3-007-00
000008	000001		DIMENSION CMSC(2)
000009	000002		EQUIVALENCE (CMSC,COUT(4))
000010		-128,128	B8F5FM NH-TKTF-451-00002 TICKET NH-R3-007-00
000011	000001		NO = BDBRND - 1
000012		-130,131	B8F5FM NH-TKTF-451-00002 TICKET NH-R3-007-00
000013	000001		IF(MOD(NO,1000).NE.0) GO TO 2020
000014	000002		ENCODE(8005,CMSC) NO
000015		-137	B8F5FM NH-TKTF-451-00002 TICKET NH-R3-007-00
000016	000001		IF(BDBSTS.EQ.1.AND.DBG.EQ.0) GO TO 2900
000017	000002		SKIP NON SUPORT RECORD
000018		-270,270	B8F5FM NH-TKTF-451-00002 TICKET NH-R3-007-00
000019	000001		SUBSTR(OTC,PTR,11)-SUBSTR(TEMPC,1,11)
000020		-343,344	B8F5FM NH-TKTF-451-00002 TICKET NH-R3-007-00
000021		-413	B8F5FM NH-TKTF-451-00002 TICKET NH-R3-007-00
000022	000001		BDBRND = BDBRND - 1
000023		-418	B8F5FM NH-TKTF-451-00002 TICKET NH-R3-007-00
000024	000001		IF(DBG.EQ.0.AND.BDBRND.GT.DRCNT) GO TO,2910
000025		-421	B8F5FM NH-TKTF-451-00002 TICKET NH-R3-007-00
000026	000001	2910	B8F5FM NH-TKTF-451-00002 TICKET NH-R3-007-00

ファイルである。システムの生成時、TCF と PCF は、それぞれの SI 基本原始プログラム・モジュールに対応する文番号をキーとし併合され新しい PCF が作成される。TCF と PCF の併合例を図-2 に示す。この例では、モジュール α の TCF は、基本原始プログラム・モジュール α の文番号 1 と 2 の間に文 X を挿入し、更にモジュール α の PCF の -6 を見出し部とする開発/修正コードの文番号 1 の文すなわち文 D を削除し文 Y および Z を挿入することを意味している。後者の修正形式を相対修正形式(Relative Correction) という。

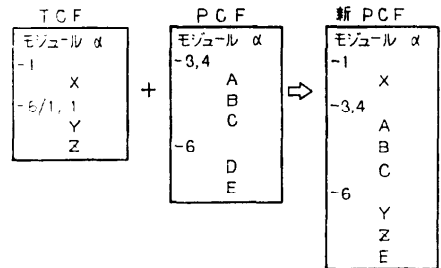


図-2 TCF と PCF の併合

④ リロケータブル (再配置可能) コードファイル (Relocatable Object File-RO)

システム生成時、SI の原始モジュールは、対応する PCF のモジュールを修正情報として入力しコンパイルあるいはアセンブルされ、そのリロケータブル・モジュールが、このファイルに出力される。

⑤ 更新済原始コードファイル (Symbolics Output File-SO)

システム生成時、SI の基本原始モジュールは、対応する PCF のモジュールを修正情報として入力しコンパイルあるいはアセンブルされ、その更新済原始モジュールがこのファイルに出力される。更新済原始モジュールの出力状況例を図-3 に示す。この例では、最初に PCF の示す -1 によって基本原始プログラム・モジュール α の文番号 1 と 2 の間に文 X を挿入する。次に -3, 4 によって文番号 3 および 4 の文ハと

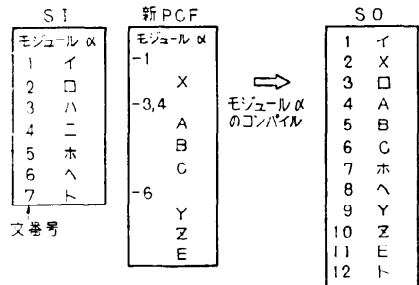


図-3 更新済原始モジュール

ニを削除し文 A, B, C を挿入する。更に -6 により文番号 6 と 7 の間に文 Y, Z, E を挿入し、更新済原始プログラム・モジュール α を出力している。

⑥ 変更管理書式 (Change Control Form-CCF)

開発/修正作業において作成するコードに、以下の付属情報を付加したものを CCF という。モジュール名、機能種別コード (パッケージ・Id)、コード作成年月日、コードの属性 (修正、開発、標準、非標準、な

ど), コード作成者名, 不具合の症状 (SYMPTOM), コードの説明 (CHANGE), 不具合報告書番号, 検査確認方法 (TEST METHOD)

SCCS は, システム生成時に, CCF の書式検査をし付属情報の完備していないものについては, エラー・CCF とする. 検査済 CCF は, 開発/修正コード履歴ファイル (CCF ファイル) と開発/修正コードファイル (TCF) に出力される.

⑦ CCF ファイル

CCF ファイルは, 前項で述べたようにシステムに組み入れ対象となった CCF の累積ファイルであり, それぞれの CCF は組み入れられたシステムの名前 (システム・レベル) が SCCS によって付加されている. このことにより, 基本原始コードファイルと, この CCF ファイルを使って任意の時点におけるシステム (任意のシステム・レベル) の修復が可能となっている.

また, 付属情報である機能種別コード (パッケージ・Id) は, 機能単位ごとに定め付加してあるので, この CCF ファイルからパッケージの抽出を可能にしている. 更に, 機能別のコード量および関連モジュール数などの機能別情報の把握を可能とし, 機能の拡張あるいは修正作業の影響度合とか作業予測などの有効な手段となっている.

⑧ レベル・Id

システム生成時に定義する, システムの名前.

⑨ システム生成管理情報

システム生成作業完了時に, SCCS によって以下の管理情報が出力される.

① CCF 検査表

システムに組み入れた CCF の検査表で, コード作成者が予定した通り組み入れられているかどうか確認するためのチェック・リストである.

② 不具合回避コード要約表

システムに組み入れられた不具合回避コード (あらかじめ定めた仕様どおりに作動しない機能を, 仕様どおりに作動するように組み入れたコード) の付属情報の要約表を出力する.

③ CCF 要約表

システムに組み入れた CCF の付属情報の要約表を出力する.

④ その他

システム生成時, コンパイルあるいはアセンブ

ルしたモジュール名の一覧表, JCL 表など.

4.1.2 運用方式と開発方式

同一コンピュータ・システムの下で, 運用と開発作業すなわちプログラマごとのシステム生成と検査を可能とするために, SCCS とシステム自身によって, 次の維持管理機能を支援している.

① 前項で述べたシステム生成環境をプログラマごとに設定し, システム生成作業の同時並行処理を可能にし, 複数プログラマが同時にシステムの開発と検査を実施している. ただし, システム生成に必要なファイルについては, 可能な限り共用ファイルを使用し, プログラマごとの設定ファイルは小規模に留めている. この場合, SCCS の運用形態を, 後で述べる組織的システム生成システム方式と対比し, プログラマシステム方式 (P-ライン) と呼ぶ. P-ラインの概略を図-4 に示す. P-ラインの A および B は, 図-1 に示したとおりである.

② システム自身もつ支援機能

前項の P-ラインによって生成されたシステムは, その検査時, 開発用に設定されたデータベースに対し入出力するよう考慮されている. すなわち, 同一コンピュータ・システムの下で, 開発中のシステム (P-ラインによって生成されるシステム) は, 開発用のデータベースに対し入出力し, 運用システム (組織的システム生成システム方式によって生成されたシステム) は, 運用データベースに対し入出力するよう考慮されている. 前者を開発方式, 後者を運用方式と呼び, それぞれ開発中のシステムと評価システムあるいは要員訓練システムなどの組合せによって利用形態は広く, 非常に有効であった. 現在においても, 運用システムの稼

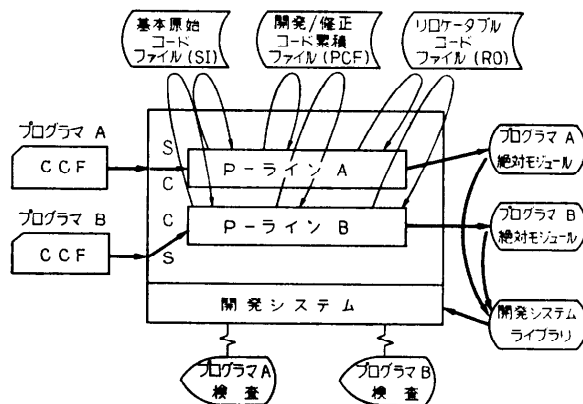


図-4 P-ライン

動時間中に新規機能の開発作業を、同一コンピュータ・システムの下で実施している。この概略を図-5 に示す。

#### 4.1.3 運用データベースと開発データベース

大規模データベース・システムにおけるデータベースの規模は、ディスク・パックにして数十パックも必要になる。しかしながら、システムの開発当初よりこれらをすべて装備することは不経済である。また前項で述べた、運用システムと開発中のシステムを共存させる場合、大規模データベースを二重に装備することは不経済であるし不可能である。そこで次のように考えた。

① 固定した大きさの、あるいは、年次に応じて増加量の少ないファイルは、開発当初より確保しておいた。年次に応じて大きく増加するファイルについては、ファイルの分割を行い必要の都度、その容量を増加させる機能を開発した。

② 開発データベースについては、固定量のファイル部は運用データベースと同様とし、可変量のファイル部は最小限の容量とした。この場合、ファイル識別コードは、運用データベースと開発データベースの対応する各ファイルについて、一定のバイアスをもたせ設定した。すなわち運用データベースの各ファイルの識別番号は400番代とし、開発データベースのそれは200番代とした。

#### 4.2 システム生成方法と組織的システム生成系統管理 (ライン・コントロール)

システム生成の要素とP-ラインについて述べたが、更にP-ライン(開発、検査)の運用から運用システムの生成過程について述べる。

##### 4.2.1 組織的システム生成系統管理 (ライン・コントロール)

運用システムは、あらかじめ定めた機能範囲の機能が仕様どおりに作動し、実用に供することを第一義とするが、一方では次期運用システムとしての新たな機能の開発が望まれるのが普通である。すなわち限定した機能の安定性の向上と新規機能の開発という裏腹な目的を同時に達成するために、次の2つのシステム生成ラインを定義し、SCCSによって運用している。ここでいう生成ラインとは、システム生成の流れを意味する。システム生成ラインを図-6 に示す。

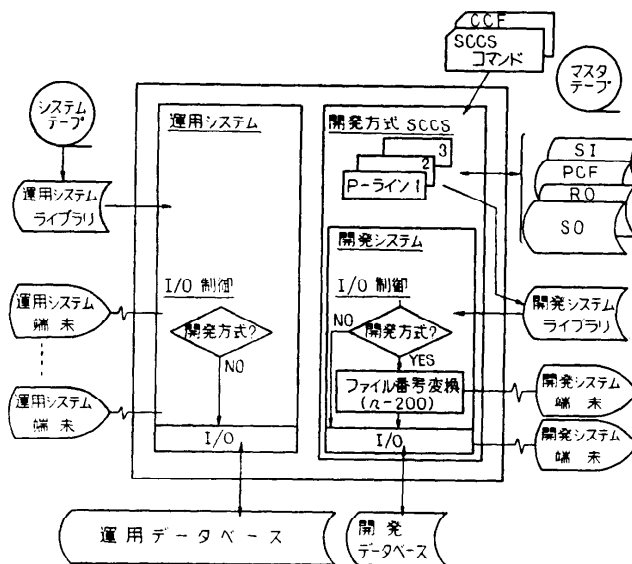


図-5 運用方式と開発方式

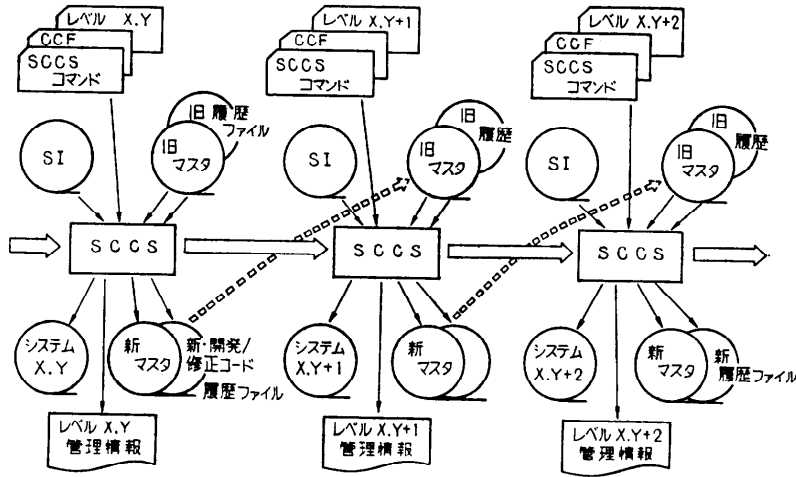
① 主システム生成ライン (メイン・ライン)  
新規機能コードの組み入れを行い、将来の運用システムとして開発推進するシステム生成ライン。

② 従システム生成ライン (サイド・ライン)  
次期運用システムとして、限定した機能範囲の機能が仕様どおりに作動することを目的とするシステム生成ライン。このシステムには、不具合回避コードのみを組み入れる。

図-7 は、メイン・ラインとサイド・ラインの運用形態を示す。新規機能の開発に、長期開発期間を要する場合、たとえば、新規機能 A と B の開発期間と C の開発期間は、それぞれシステム・レベル X、Y の生成日からレベル X、Y+3 およびレベル X、Y+8 の生成日の期間とすると、機能 A と B については、サイド・ライン・システムの中に包含されるが、機能 C については包含してはならない。この場合には、次の二通りの運用形態がある。

① 長期開発期間を要する機能コードは、マクロ命令 ON/OFF 句でかこみメイン・ライン・システムに組み入れ、サイド・ライン・システムにおいては注釈行とする運用形態。

② 機能種別コード(パッケージ・Id)を付加して、メイン・ライン・システムに組み入れ、CCF ファイルを媒体としてサイド・ライン・システムに組み入れる。この時、サイド・ライン・システムとして不要な



注) マスタ・ファイル: PCF, RO, SO ファイルをもつ.

図-6 システム生成ライン

機能コード (パッケージ) を組み入れ除外する運用形態.

4.2.2 開発・検査 (P-ライン) から組織的システム生成への手順

各プログラマは、最新のメイン・ライン・システムから出力された更新済原始プログラム・モジュールに対し開発すべきコードを考え、SI および PCF の文番号を確認し、CCF を作成する。SCCS の P-ライン方式によって CCF の検査確認の上、完成した CCF を、CCF 収集箱に提出する。組織的システム生成に先立って、原始コード管理者と開発責任者によって、CCF 検査確認会議が開催され、組み入れる CCF を確定する。その後システム生成作業を実施する。システム生成後、作動確認ツールによって評価し、不具合があれば不具合報告書を作成し開発担当グループに報告する。そして P-ラインの環境に新しいシステムを設置する。開発・検査・システム生成サイクルを図-8 に示す。

4.3 SCCS の運用方式と運用システムの方式

SCCS の運用方式は、その運用形態によって次の 3 つの運用方式が可能である。

- ① 開発方式 (P-ライン)
- ② システム生成方式
- ③ 統計資料作成方式

運用システムの方式は、次の 4 つがあり、特に検査方式は、24 時間 365 日運用を前提とする航空座席予約システムでは必須である。

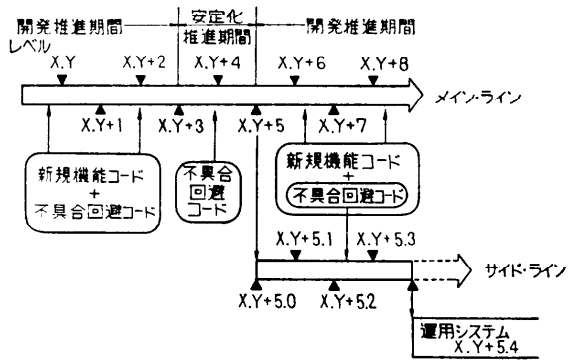


図-7 メイン・ラインとサイド・ライン

- ① 稼動方式
- ② 訓練方式
- ③ 開発方式
- ④ 検査方式

4.3.1 訓練方式

この方式は、システム運用開始後の要員訓練のための方式で、データベースからの入力には運用データベースから、出力については訓練用データベースへ、という機能を提供し、運用システムの下で要員の訓練を可能にしている。

4.3.2 検査方式

この方式は、運用システムにおいて不具合が発生し、回避コードを作成したが運用データベースを使用しなければ検査確認できない場合に使用する方式である。データベースに対する入出力は、訓練方式と同様な機

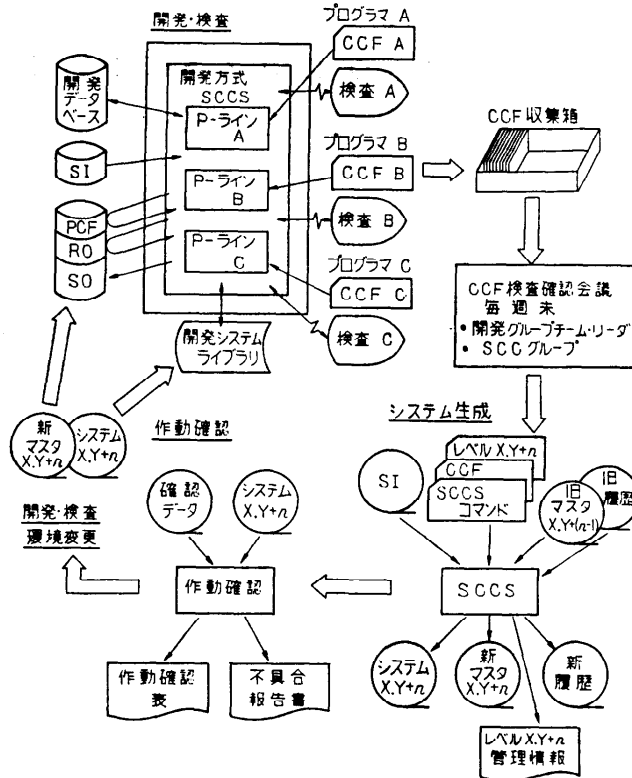


図-8 開発・検査・システム生成サイクル

能によって、入力は運用データベースから、出力は検査データベースへ出力される。システムは、SCCS によって変更された被検査システムを使用する。

### 5. むすび

今般の開発作業を通じて、プロダクトライン管理、開発支援システムおよび開発支援環境の重要性と効果を身をもって感じた。プログラミング技術については、常々議論される場面が多いが、プロダクトライン管理とか開発支援の技術についても、もっと議論されるべきであろう。プログラマの作業能率の向上を計り、か

つ目的とするシステムを確実に構築するための開発支援システムと環境と手続きを考えねばならない。今後の多様化する大規模システムの開発・構築作業においては、更に求められる分野となる。試行を重ね、より良い技法の確立を計りたい。

### 参考文献

- 1) 藤田 献：リアルタイム予約システム。
- 2) 山谷正己，秋山義博：仮想計算機。

(昭和 55 年 7 月 2 日 受付)