

62-S-001

システム監査支援ツールの 調査研究報告書



昭和 63 年 3 月

JIPDEC

財団法人 日本情報処理開発協会

この報告書は、日本自転車振興会から競輪収益の一部である機械工業振興資金の補助を受けて昭和62年度に実施した「システム監査支援ツールの調査研究」の成果をとりまとめたものであります。

請求番号	JIPDEC 62-S001-7		登録番号	
著者名	システム監査支援ツールの			
書名	調査研究報告書			
所属	帯出者氏名	貸出日	返却 予定日	返却日

序

情報処理システムに対するシステム監査の必要性、重要性については、昭和60年1月の通商産業省のシステム監査基準の策定と昭和61年のシステム監査試験の開始に及んで、その認識が高まってきている。しかし、実際にシステム監査を実施している組織は、19.4%（昭和62年、情報化白書）と少ない。これは、システム監査の具体的な方法や手法が未成熟なこと、組織の体制が整っていないこと、開発段階においては納期第一であること等によるためと思われる。

システム監査の主要な目的である情報処理システムの信頼性、安全性の確保は、従来、運用中において継続的にシステム監査を実施することによって実現されるという考えが主流であったが、最近では、情報処理システムの運用時にはなく開発段階において、その信頼性、安全性への対応を十分に確保することが必要であるとの考え方が強くなってきている。

今後システム監査を幅広く普及・浸透させるには、組織側の努力は当然のことながら、システム監査の方法や手法の確立、特に開発段階のシステム監査の場合は、開発作業の進捗に影響を及ぼさない効率的な監査手法や支援ツールの導入が急がれる。

本調査研究は、以上のような状況を踏まえ、信頼性、安全性の観点から開発途上の情報処理システムを評価するシステム監査に的を当て、その活動を支援するためのツールを研究することを目的に、昭和61年度から2年計画で開始した。

2年目の今年度は、1年目の成果を踏まえ、情報処理システムの開発局面においてその信頼性、安全性の監査支援ツールの在り方、その機能要件等を検討した。また、システム監査エキスパートシステムについても調査した。

その結果、システム監査支援ツールは、監査対象の情報処理システムの開発支援ツールと切り離して考えることが出来ないため、それと連動するシステム監査機能を提案した。更に、開発支援ツールについても、開発作業を一貫してコントロールすることが重要であると認識し、開発コントロール機能を追加し強化することを提案した。

なお、調査研究を進めるに当たっては、前年度に引き続きシステム監査支援ツール研究委員会を設けて実施した。石崎委員長をはじめ各委員の方々には、御多

忙の中、積極的な参加と多大な協力を得ることとなった。ここに、深く感謝の意を表す次第である。

—システム監査支援ツール研究委員会（順不同、敬称略）—

委員長	石 崎 純 夫	（産業能率大学）
委 員	浅 輪 寿 男	（住友スリーエム㈱）
”	上 園 忠 弘	（日本アイ・ビー・エム㈱）
”	大 井 正 浩	（㈱三菱銀行）
”	高 橋 富 夫	（㈱富士通）
”	堀 内 一	（㈱日立製作所）
”	松 尾 明	（鯨中央会計事務所）
”	木 村 秀 夫	（アーサーヤングコンサルティング㈱）
”	菅 原 利 行	（アーサーアンダーセン アンド カンパニー）
”	土 屋 昭 治	（㈱富士通）
”	松 尾 谷 徹	（日本電気㈱）
”	山 本 欣 子	（㈱日本情報処理開発協会）
オブザーバー	若 月 俊 彦	（日本電気㈱）

昭和63年3月

目 次

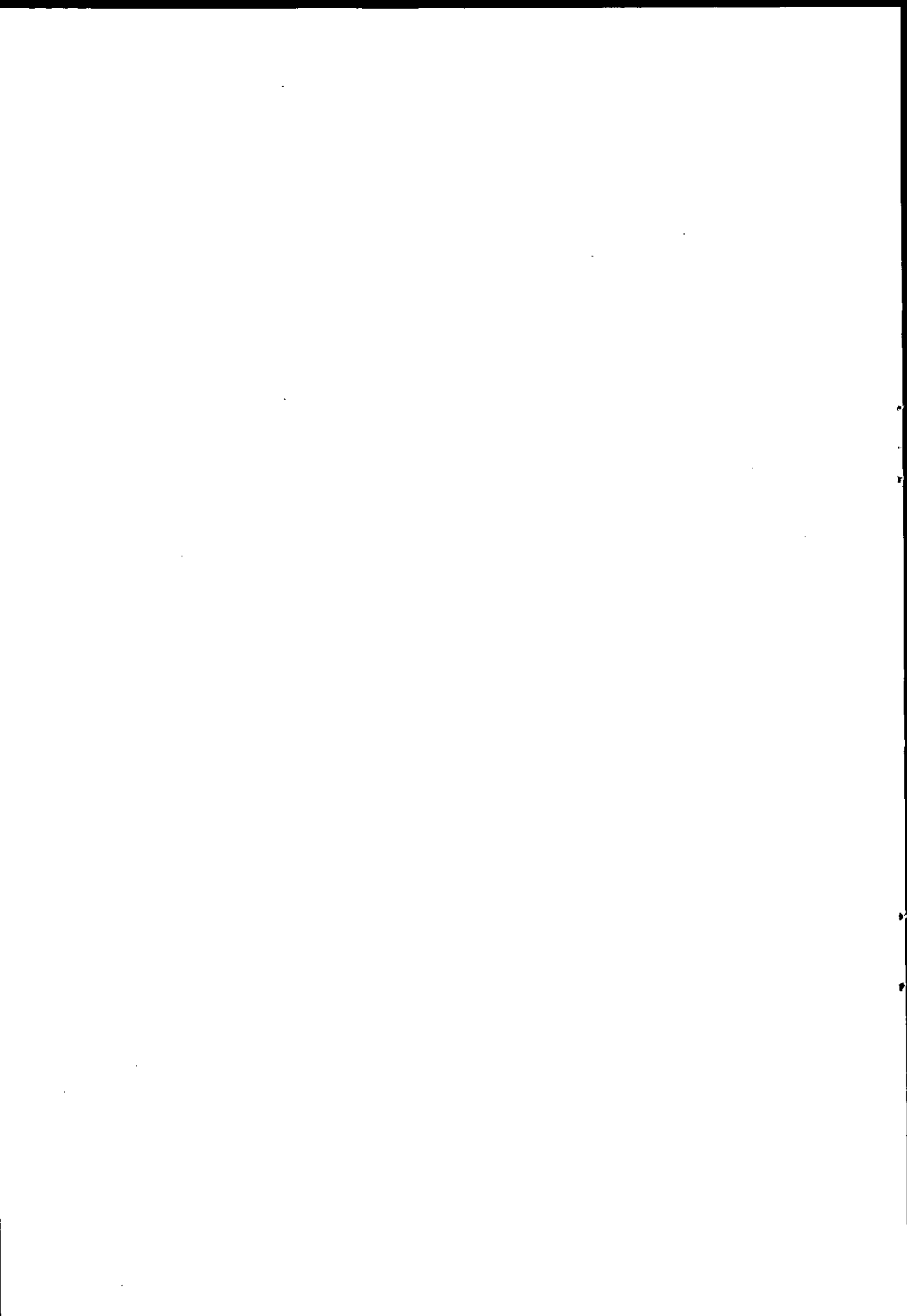
第 I 編 システム監査支援ツールの機能仕様	1
1. システム監査支援ツールの目的と要求仕様設定の基本方針	3
1.1 システム監査支援ツールの目的	3
1.2 要求仕様設定の基本方針	3
1.2.1 システム監査に関する方針	3
1.2.2 システム開発との関連に関する方針	8
2. システム監査支援ツールのコンセプト	10
2.1 システム開発におけるコントロール	10
2.1.1 システムライフサイクルとコントロールの重要性	10
2.1.2 コントロールの定義と機能	10
2.1.3 システムライフサイクルにおけるコントロールの目標	12
2.1.4 システムライフサイクルにおけるコントロールの機能要件	16
2.2 システム開発におけるシステム監査	23
2.2.1 システムライフサイクルにおけるシステム監査のモデル	23
2.2.2 システム監査支援機能	25
2.3 システム監査支援ツールのフレームワーク	28
2.3.1 機能構成	28
2.3.2 開発コントロール	30
2.3.3 監査 (SA/QA) 支援機能	30
2.3.4 ベース機能群	31
3. システム監査支援ツールの機能要件	32
3.1 品質管理項目と品質評価項目	32
3.1.1 システム開発におけるQC/QAの役割	33
3.1.2 品質の計測と評価	36
3.1.3 監査のチェックポイント	38
〔参考資料〕 A. 品質の分類と尺度	44
B. 品質の実現方法 (品質機能展開)	48

〔参考文献〕	50
3.2 システム開発支援ツール機能	51
3.2.1 システム開発支援ツールの歴史・遷移	51
3.2.2 システム開発支援ツールの現状と調査研究対象の選定	54
3.2.3 システム開発支援ツールによる開発工程の変化	58
3.2.4 システム開発支援ツールとシステム監査支援ツールの関係	60
3.3 開発コントロール機能	65
3.3.1 開発コントロール機能の目的	65
3.3.2 開発作業のリスク	66
3.3.3 開発コントロールの検討	70
3.3.4 システム開発支援ツールにおける開発コントロール機能の実現	74
3.3.5 作業ステップにおけるコントロール機能例	76
3.4 SA/QA支援機能	78
3.4.1 一貫性・完全性に関する測定支援機能	79
3.4.2 安全性・信頼性に関する測定支援機能	83
3.4.3 その他の測定支援機能	86
3.4.4 評価支援機能	87
3.4.5 監査人支援機能	89
4. システム監査支援ツール実現へのアプローチ	92
4.1 資源管理の充実	92
4.2 制御の命題化と集約	92
4.3 ソフトウェア構造の改善	93
4.4 開発方法論の変更	94
5. これからのシステム監査とその課題	96
5.1 内部統制について	96
5.2 開発段階におけるシステム監査の現状	98
5.3 近未来のシステム監査	99
5.4 不正プログラム機能の予防／発見の考察	102

第Ⅱ編 システム監査へのエキスパートシステム適用	107
1. 米国におけるシステム監査へのエキスパートシステム適用実態	
調査の概要	109
1.1 調査の目的	109
1.2 調査の結論	109
1.3 調査結果	109
1.3.1 調査のアプローチ	109
1.3.2 環境分析調査の概要	111
1.3.3 導入検討調査の概要	112
1.3.4 システム監査エキスパートシステム導入可能性検討の 概要	114
2. システム監査へのエキスパートシステムの適用の期待・可能性	118
2.1 システム監査へのエキスパートシステム適用の基本的な考慮 事項	118
2.2 システム監査局面別に見たエキスパートシステムの適用	120
2.2.1 監査計画局面へのエキスパートシステムの適用	120
2.2.2 監査実施局面へのエキスパートシステムの適用	122
2.2.3 監査報告局面へのエキスパートシステムの適用	124
2.2.4 フォローアップ局面へのエキスパートシステムの適用	124
2.3 システム監査へのエキスパートシステム適用の効果・期待	125
2.4 システム監査にエキスパートシステムを適用する上での 問題点・課題	128
2.4.1 システム監査固有の問題点・課題	128
2.4.2 エクスパートシステムの一般的な問題点・課題	129
付録 米国におけるシステム監査へのエキスパートシステム適用 実態調査の詳細	131

the first of these is the fact that the
the second is the fact that the
the third is the fact that the
the fourth is the fact that the
the fifth is the fact that the
the sixth is the fact that the
the seventh is the fact that the
the eighth is the fact that the
the ninth is the fact that the
the tenth is the fact that the
the eleventh is the fact that the
the twelfth is the fact that the
the thirteenth is the fact that the
the fourteenth is the fact that the
the fifteenth is the fact that the
the sixteenth is the fact that the
the seventeenth is the fact that the
the eighteenth is the fact that the
the nineteenth is the fact that the
the twentieth is the fact that the
the twenty-first is the fact that the
the twenty-second is the fact that the
the twenty-third is the fact that the
the twenty-fourth is the fact that the
the twenty-fifth is the fact that the
the twenty-sixth is the fact that the
the twenty-seventh is the fact that the
the twenty-eighth is the fact that the
the twenty-ninth is the fact that the
the thirtieth is the fact that the
the thirty-first is the fact that the
the thirty-second is the fact that the
the thirty-third is the fact that the
the thirty-fourth is the fact that the
the thirty-fifth is the fact that the
the thirty-sixth is the fact that the
the thirty-seventh is the fact that the
the thirty-eighth is the fact that the
the thirty-ninth is the fact that the
the fortieth is the fact that the
the forty-first is the fact that the
the forty-second is the fact that the
the forty-third is the fact that the
the forty-fourth is the fact that the
the forty-fifth is the fact that the
the forty-sixth is the fact that the
the forty-seventh is the fact that the
the forty-eighth is the fact that the
the forty-ninth is the fact that the
the fiftieth is the fact that the
the fifty-first is the fact that the
the fifty-second is the fact that the
the fifty-third is the fact that the
the fifty-fourth is the fact that the
the fifty-fifth is the fact that the
the fifty-sixth is the fact that the
the fifty-seventh is the fact that the
the fifty-eighth is the fact that the
the fifty-ninth is the fact that the
the sixtieth is the fact that the
the sixty-first is the fact that the
the sixty-second is the fact that the
the sixty-third is the fact that the
the sixty-fourth is the fact that the
the sixty-fifth is the fact that the
the sixty-sixth is the fact that the
the sixty-seventh is the fact that the
the sixty-eighth is the fact that the
the sixty-ninth is the fact that the
the seventieth is the fact that the
the seventy-first is the fact that the
the seventy-second is the fact that the
the seventy-third is the fact that the
the seventy-fourth is the fact that the
the seventy-fifth is the fact that the
the seventy-sixth is the fact that the
the seventy-seventh is the fact that the
the seventy-eighth is the fact that the
the seventy-ninth is the fact that the
the eightieth is the fact that the
the eighty-first is the fact that the
the eighty-second is the fact that the
the eighty-third is the fact that the
the eighty-fourth is the fact that the
the eighty-fifth is the fact that the
the eighty-sixth is the fact that the
the eighty-seventh is the fact that the
the eighty-eighth is the fact that the
the eighty-ninth is the fact that the
the ninetieth is the fact that the
the ninety-first is the fact that the
the ninety-second is the fact that the
the ninety-third is the fact that the
the ninety-fourth is the fact that the
the ninety-fifth is the fact that the
the ninety-sixth is the fact that the
the ninety-seventh is the fact that the
the ninety-eighth is the fact that the
the ninety-ninth is the fact that the
the hundredth is the fact that the

第I編 システム監査支援ツールの機能仕様



1. システム監査支援ツールの目的と要求仕様設定の基本方針

1.1 システム監査支援ツールの目的

システム監査は、我が国においてはその必要性が徐々にではあるが認識され始めて来た。しかし、実施状況を見ると、金融機関等一部の先進的組織を除いては、まだまだ不十分な状況にあると言えよう。特に開発中のシステムについて、本格的にシステム監査を実施している所は、その意義は十分に理解はしていても極めて小数である。

これは、システム開発に与えられた期間が短く、したがって開発者または組織の第一の関心が目標の時期迄にシステムを完成させ運用に漕ぎ着けることにあるからである。これに加えて、開発中のシステムを監査するための方法論が十分に確立されておらず、また効率的に監査するためのツールが皆無であること、及び体制が確立されていないことも原因している。

本調査研究においては、情報処理システムを安全且つ効率的に運用させることを目的としているシステム監査の主旨を活かすためには、運用時のシステム監査も重要ではあるが、開発局面で信頼性が高く安全なシステムであることを保証するシステム監査がより重要であるとの考え方を持って、システム監査支援ツールの要求仕様を提案するものである。

本ツールは、開発中のシステムを監査するための監査活動を効率的に支援し、また監査の精度を向上させると共に、併せて開発局面のシステム監査の普及を促進させることを目的としている。

1.2 要求仕様設定の基本方針

ここに提案するシステム監査支援ツールは、以下に示す基本方針の下にその要求仕様を設定することとした。

1.2.1 システム監査に関する方針

(1) 監査ポイント

システム監査基準によると、システム監査は、「信頼性」、「安全性」、「効率性」の三つの監査ポイントでシステムを監査することである。ここで、信頼性とは、①システムの品質が高いこと、②エラー、事故の発生を未然に防止できること、③万一発生した場合には、影響を最小限に食い止

め、迅速に回復できることであり、安全性とは、①自然災害からシステムを保護できること、②不正アクセスや破壊行為からシステムを保護できることである。更に、効率性とは、①システムのリソースを最大限に活用していること、②コスト／パフォーマンスの向上が図られていることであると同基準では定義している。

本調査研究では、システム監査人の監査活動をコンピュータで支援するシステム監査支援ツールにターゲットを絞って調査・研究を行うものである。その上、本ツールで支援する監査活動の対象は、設計・開発中のシステムである。

そこで、有効性の意味合いを持つ「効率性」に関しては、設計開始以前に評価されているものであり、また、コンピュータ支援の対象とはなりにくいため、本ツールが対象とする監査ポイントから除外することとした。なお、性能の意味合いを持つ「効率性」に関しては、ユーザ要件の一つと考え「信頼性」に含まれるものと捉えている。

監査ポイントとした「信頼性」、「安全性」についても、支援対象の監査活動が開発段階にあるシステムに対するものであることを考慮して、次のように見直した。

開発段階にあるシステムの監査では、「開発段階での成果物（仕様書、設計書、プログラム等）」だけでなく、「開発作業」自身も監査対象とする必要がある。何故なら、監査対象のシステムは、コントロールされた状態の下で設計・開発されていることがシステム監査の前提であり、先ず開発作業をその観点で監査しておく必要があるからである。

したがって、開発作業に対する監査は、別の監査ポイントで評価することが必要である。開発作業の監査で重要視すべきことは、開発技術基準、開発作業基準、ドキュメント基準など開発作業を規定している基準に準拠して作業が行われているかという「準拠性」の面と、開発作業中に機密度の高いドキュメント、プログラム、データなどの資源を無許可で見ているかという「安全性（セキュリティ）」の面である。開発段階での成果物に対する監査は、信頼性及び安全性の評価で十分であると考えられる。

以上のような検討により、本システム監査支援ツールが対象とする監査ポイントは、次に示すものとした。

- (a) 開発作業に対する監査ポイント
 - ① 準拠性
 - ② 安全性
 - (b) 開発段階での成果物に対する監査ポイント
 - ① 信頼性
 - ② 安全性
- (2) 本調査研究での監査ポイントの定義
- (a) 開発作業に対する監査ポイント
 - ① 準拠性…開発作業が開発基準／標準に則って作業が行われ、正規の承認手続きを得て作業が進められていること。
 - ② 安全性…開発作業中に機密度の高いドキュメント、プログラム、データ等がアクセス規約に従って保護されていること。保護の概念には、不正アクセスからの保護以外に、過失による破壊からの保護も含んでいる。
 - (b) 開発段階での成果物に対する監査ポイント
 - ① 信頼性…各成果物が、上位の成果物で規定された要件を満たすように作成されていること。
 - ② 安全性…完成したシステムが運用時に安全に運用できるようにするために必要な安全対策機能が必要十分に設計・実現されていること。

システム監査対象と監査ポイントの関連を図 1-1 に示す。

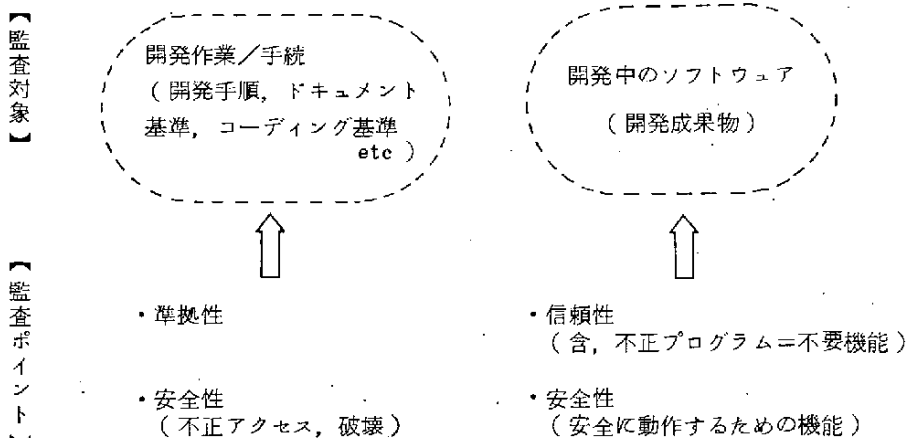


図 1-1 監査対象と監査ポイント

(3) システム監査支援ツール機能の考え方

システム開発のリスクを見ると、開発成果物に関するリスクと、開発作業／手続（含む、環境）に関するリスクの2つに大別される。システム開発におけるシステム監査の役割は、これらのリスクを取り除くための一つの管理活動と捉えることが出来る。そこで、今まで述べてきた、システム開発段階のシステム監査の監査ポイントを、リスクに対応させて見ると次のことが分る。

開発作業／手順に対する準拠性、安全性の監査は、作業や手続きの規定を守らずに作業したために起こるリスクの発生を防止するためのものであり、また、開発成果物に対する信頼性、安全性の監査は、開発または設計の内容が正確でない、矛盾している、足りない等のリスク発生を検出し設計・開発にフィードバックし、結果的にリスク発生を防止する活動である。

開発作業／手順に関する準拠性、安全性の監査は、そのタイミングを見ると、一つ一つの作業に直接に働きかけて時々刻々と実施しなければ、その作業より生まれた開発成果物は当該組織の基準に違反したものとなっているかもしれないので意味がない。

したがって、この監査を支援するツールとしては、作業を刻々監視し基準違反を防止させる、即ちコントロールのイメージを持った機能が必要と考えている。

一方、開発成果物に対する信頼性、安全性の監査は、各工程で出来上がった成果物を対象とするため、その品質を評価することに近いイメージを持っており、したがって、それを支援する機能群の集まりがこの監査を支援するものとして、システム監査支援ツールに必要と考えている。

リスクと監査ポイント及び支援機能との関連を、表1-1と図1-2に示す。

表1-1 開発過程のリスクとその監査ポイント、支援機能の関連

リ ス ク	監査ポイント	支 援 機 能	
1.開発作業／手 続（含、環境） に関するリスク	・準 拠 性 ・安 全 性	開発作業コント ロール支援	開発作業を刻々監視し、基準 違反を防止させることを支援 する。
2.開発対象に関 するリスク	・信 頼 性 ・安 全 性	品質評価支援	各工程で出来上がった成果物 の品質を評価することを支援 する。

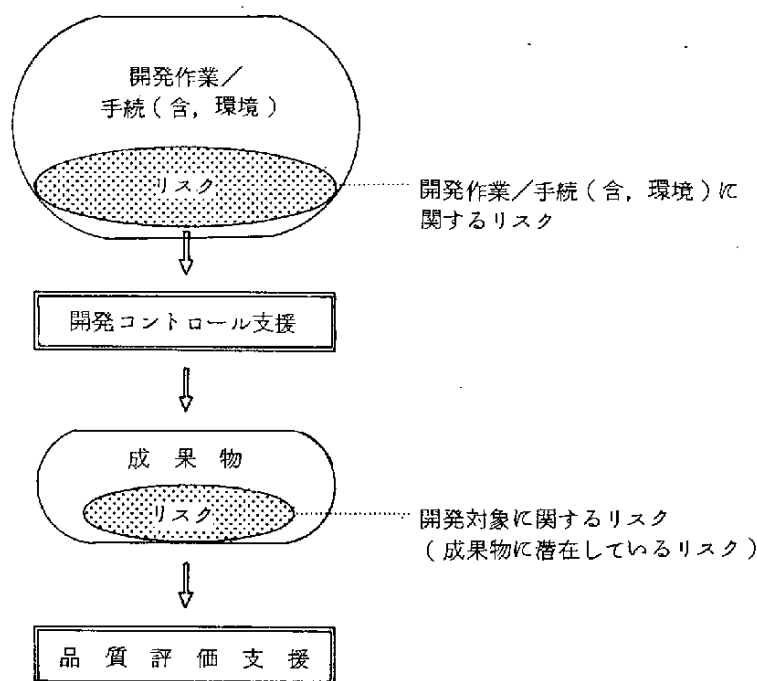


図 1 - 2 システム監査支援ツールの機能と対象のリスク

1.2.2 システム開発との関連に関する方針

(1) 開発支援ツールとの関連

システム監査支援ツールの対象とする開発作業／手続（含、環境）と開発成果物は、開発支援ツールとも密接に関連している。そのため、システム監査支援ツールの仕様を検討するに当たっては、前提として開発支援ツールを設定しておく必要がある。

しかしながら、世の中には色々な開発方法論（含む、開発技法）に基づいて作成された開発支援ツールが存在しており、特定のものを前提とするのは難しい。そのため、世の中の動向を勘案して開発支援ツールのモデルを設定することとするが、それを利用する組織の大きさ、開発対象の分野、採用している開発技法等、考慮しなければならない要素が多すぎる。

以上のようなことから、ここでは漠然とではあるが、Σプロジェクトで提案している開発支援ツールを基本にすることとし、その環境でシステム

開発するシステムの監査を支援するツールの機能要件を提案することとした。

なお、その他の要素については、以下のように設定している。

- ① 開発作業は、ワークステーションを利用して開発する分散環境で実施する。
- ② 開発成果物（ドキュメント、プログラム等）は、一貫して開発支援ツールによってデータベース化して管理されている。

(2) 支援の範囲

システム監査支援ツールがカバーする範囲は、システム設計からテストまでのシステム監査を対象とする。ただし、システム保守の局面についても考慮はすることとする。

2. システム監査支援ツールのコンセプト

2.1 システム開発におけるコントロール

2.1.1 システムライフサイクルとコントロールの重要性

システム開発における監査を支援するツールを考えると、単に、対象とするシステム開発作業や、その結果をいたずらに検証し、不備を指摘することを考えるのではなく、次のような要件を満たすことが重要となる。

- ① システム活動の開発過程だけに着目するのではなく、その運用過程、とくにシステムの保守過程を含めたライフサイクル全般に一貫したコントロール体系を用意すること。
- ② 監査の目的となる、システムの安全性、完全性、信頼性などを確保するためのコントロールを明確にして、そのコントロールを評価する役割として、システム監査を考えること。
- ③ コントロールの実現は、その完全性のみを追求するのではなく、コントロールが位置づけられる現実的な環境において、有効で、かつ効率的なものでなければならない。また、コントロールに対して組織が持つ受け入れ許容度の範囲をこえたものであってはならない。

システム監査を、「コントロールを評価する活動」と捉えると、システム活動全般について、まず、如何なるコントロールが必要となるが課題となり、次にそのコントロールをどのように実現するかが問題となる。さらに、そのコントロールをどのように評価、吟味するかも大きな課題となる。

システム監査支援ツールはそのような課題に対して、

- ① 受け入れ許容度の範囲にありながら、効率的で、効果的なコントロールを実現する仕掛けの提供
 - ② コントロールを評価するための、確実な手段の提供
- という点で大きな役割を果たすものと考えられる。

2.1.2 コントロールの定義と機能

(1) コントロールの定義

コントロール（control）とは、「ある与えられた目標、あるいは基準からみて好ましくない事態の発生を回避するための制御」と考えることができる。好ましくない事態は、一般に「リスク」と呼ばれるものである。

したがって、リスクが想定されないところには不用なものといえる。コントロールの実現は多大な費用と、労力を要し、また、実務実行上の効率低下につながる性格を持つからである。その実現と稼働は可能な限り、経済的で、効率的であることが望まれる。

(2) コントロールの目標

コントロールの目標は、大きく二つに分けて考えることができる。その一つは与えられた基準の監視とその結果に基づく制御であり、もう一つは与えられた目標を達成するために、自ら基準を変化させながら制御を行うものである。

一般に前者は、ネガティブコントロール、または閉システムと呼ばれ、後者はフォワードコントロール、または開システムと呼ばれるものである。

通常、ネガティブコントロールを実現することは易しい。基準を定義してその準拠を求めればよいからである。しかしながら、そのコントロールが与えられた上位の目標を達成する上で、有効なものであることを保証することは難しい。

システムの開発や運用においても、ネガティブコントロールを実現することは易しい。他社の規則や基準を借りてきて、闇雲に標準化を進めればよいからである。しかし、真に要求を満たしながら、完全で安全なシステムを開発、運用するためには、基準そのものを見直し、改訂する活動が継続されなければならない。

(3) コントロールのタイプ

情報システムに限らず、組み込まれるべきコントロールは、次のようにその性格を分類できる。このような分類は、その設計と実現を計る上で有効なものとなる。

- ・抑止 (deterrence) : 好ましくない事態の発生をもたらす活動の動機を失わしめること。
- ・防止 (prevent) : 好ましくない事態が発生を食い止めること、あるいは好ましくない事態の実現を阻むもの。
- ・検出 (detection) : 好ましくない事態が発生しても、その事実を発見できるもの。
- ・回復 (correction) : 好ましくない事態が発生しても、それを元の状態へ復帰させるもの。

これらのタイプのコントロールは、相互に組み合わせることで、より確実なコントロールを実現できるものとなる。また、防止や検出が技術的に不可能なとき、抑止コントロールの強化でカバーするなどの相互代替性もある。

防止、検出などのコントロールはその実現に多大な費用と労力を要するものであるだけに、効率的実現、つまり機械化などの手段が要求されるものである。

(4) コントロールの実現手段

システム開発及び保守過程をコントロールするための方法には次のようなレベルが考えられる。

- ① 共通の価値観の共有
- ② 目標、ガイドラインの提示
- ③ 標準、規則の提示と監視
- ④ 基準違反の検出、摘出機構の実現
- ⑤ 防御機構、抑止機構の実現

①から③までは、主として人為的実現手段に依存したものであり、高い融通性をもちながらも、コントロールとしての完全度は低い。もし、③の手段を持って、その完全度を上げると考えると、そのコントロールに対する許容度は失われ、暗黒社会を形成するものとなり、効率も低下せざるを得ない。

④、⑤は、その完全度の向上自体が、人為的仕掛けでは自ずから限界をもつもので、なんらかの補助手段を必要とする。ツールが要求される段階はこのレベルといえる。しかしながら、一方で融通性は失われる。

2.1.3 システムライフサイクルにおけるコントロールの目標

(1) 目標

システム開発と保守におけるコントロールの目標は、開発あるいは保守されるシステムの品質の確保にある。ここでシステムの品質とは次のようなものを、その第一義的なものと考えことにする。

- a. 要求仕様に対する充足性
- b. 信頼性
- c. 完全性

- d. 安全性
- e. 一貫性
- f. 効率性

これらの他に、拡張性、携帯性、あるいは保守性などシステムが具備すべき品質の範囲は広い。しかし、本研究では、システム監査支援ツールとコントロールを考えるに当たって、先ず、ここでいう第一義的な品質を考慮し、その中でも、特に、bからfまでを対象として検討を進めた。なぜならば、近年のシステム監査の目的も、主として安全性、信頼性の観点に置かれていること、また、要求仕様に対する充足性を保証するコントロール技術と、信頼性などを保証するコントロール技術が大きく異なるからである。

システムライフサイクルとは、システムが発生（誕生）してから消滅（廃止）されるまでの間に存在する全過程を意味する。大きくは次のように分けることができる。

システム開発サイクル（SDLC: System Development Life Cycle）

- ・ 要求分析
- ・ システム設計
- ・ プログラム設計
- ・ プログラム開発
- ・ システムテスト

システム運用サイクル

- ・ システム稼働
- ・ システム稼働の監視
- ・ システムの調整

システム保守サイクル（SMLC: System Maintenance Life Cycle）

- ・ 保守要求の分析
- ・ 保守仕様の設計
- ・ プログラム設計
- ・ プログラムの保守、変更
- ・ システムテスト

システムの廃止または置換えサイクル

- ・ 新システムの開発要求

- ・新システムの開発
- ・置き換え準備
- ・置き換えと移行
- ・旧システムの廃止

このようなシステムの長期的なライフサイクルに対する認識は、ともすれば開発のみに関心が集中し、コントロールも開発のみを重視したものになりがちであったことへの反省からといえよう。つまり、開発局面だけを重点的にコントロールして信頼性や安全性を確保しても、稼働局面や保守局面でそれらを危うくすることが容易だからである。

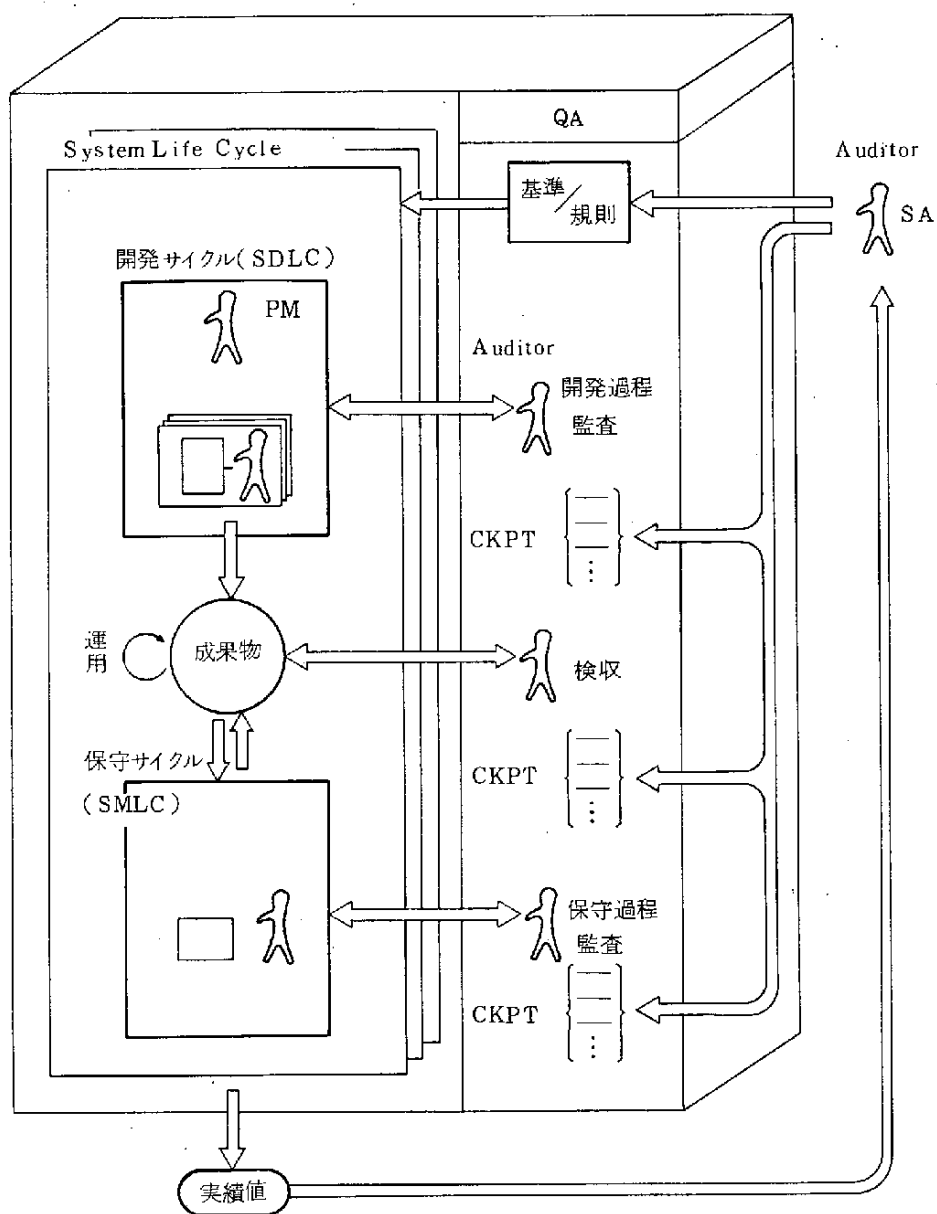
したがって、本研究でも、「開発」支援ツールを前提にして、監査支援ツールあるいはコントロールを考えると、絶えずシステムのライフサイクル全般を考慮することにした。

そのような前提でから定義されたシステム活動とシステム監査との関係を示したものが、図2-1である。

ここで、QAとはシステムの品質保証をさし、SAとはシステム監査をさす。システムまたはソフトウェアの品質保証とは、その利用者がいつでもそのシステムやソフトウェアを安心して使うことができるように保証することを意味する。そのためには、製品引渡しまでの製造過程で、十分な品質を確保することと、引渡し後の障害やトラブルに対しても十分な対応をとることが要求される。したがって、QAは開発や保守を担当するプロジェクトのように、一過的な不安的な体制ではなく、恒常的に存在する機能であることが要求される。一般に、品質管理部門、品質保証部門として組織化される。

QAは、システムの開発や保守に対して、準拠的、あるいは実証的な監査を行う主体と見することもできる。つまり、品質保証はシステムライフサイクル活動に同期して実施される性格のものであり、その活動は連続的である。開発サイクル側から要請があれば、何時でもその要請に従って、その役割を果たさなければならない。

これに対して、システム監査(SA)は、ある特定な命題から対象システムを検証する活動である。したがって、不定期的な活動となる。SA側から見れば、QAは連続的にシステム開発、保守過程を監視するコントロールと考えることができる。SAは、QAが定常的なコントロールタスクを



〔注〕 PM: プロジェクトマネージャー
CKPT: チェックポイント

図 2-1 システム開発と保守におけるシステム監査

果してくれることに依拠しながら、さらに、高度な監査課題を解決することが可能となる。

このようなコントロールの階層化は、企業における情報システムの開発、保守活動全般に対して、開システムとしての特性をもたせ、フォワードフィードバックを可能とするものとなる。

SAから見るとQAは開発、保守におけるコントロールと見ることができ、SAはそれに依拠することもできた。しかし、QAは何に依拠することができるのであろうか。

多くの報告書が指摘しているように、QAの必要性と重要性は十分認識されていながら、その実施の例は希である。特に、SDLC法に代表される開発過程監査は、QAの中心的役割とされていながら実施事例が少ない。その理由は、QA活動の負荷が余りに大きいことにある。人為的なQA活動は、先に述べた抑止、防止、検出、回復のコントロール全てを負担することになりかねない。

そこで、QAを実現可能なものとするためにも、QA自身が依拠することができるコントロールをその下に置くことを考えなければならない。コントロール機能を内包するシステム監査支援ツールが要求される理由の一つはここにある。

本研究では、システム監査支援ツール側から見れば、QAもSAも技術的にはどちらも、同等のものをを用いることから、品質保証活動を監査活動に含めて考えることにした。

2.1.4 システムライフサイクルにおけるコントロールの機能要件

システムライフサイクルにおけるコントロールは、図2-2に示すように、主として、開発などのシステム化作業を対象とするものである。ここではそれを「開発コントロール」と呼ぶことにする。

そのコントロール目標は、システム開発や保守におけるリスクと品質管理基準から設定される。つまり、システム化作業で想定される様々なリスクを回避するためのコントロールと、与えられた品質管理基準を満足させるためのコントロールから構成される。

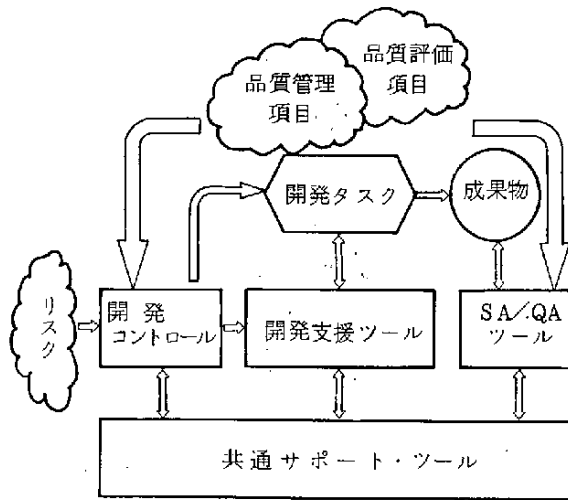


図 2-2 システムライフサイクルにおけるコントロール

開発コントロールは、主として作業を対象とするものであるが、それだけで、完全で、安全なシステム開発や保守が保証されるとは限らない。開発コントロールを、必要条件としてのコントロールと考えるならば、もう一方に、十分条件としてのコントロールが必要となる。換言すれば、開発コントロールが有効で、効率よく作動するための支援としてのコントロールが必要である。

図 2-3 に示すように、それらの支援コントロールは、環境、開発資源、あるいは、開発された成果物対応に用意されるものとなる。つまり、開発された成果物の確実な管理／コントロールと、開発作業を確実にコントロールする機能(A)とでは、方式も性格も異なる。したがって、本研究では、図の B, C, D に該当するコントロールは、開発コントロールを支える「ベース機能」と呼ぶことにする。

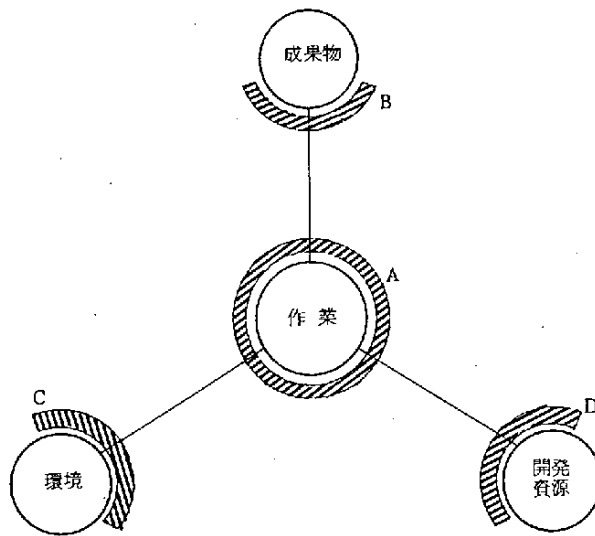


図 2 - 3 開発コントロール

(1) リスクからの要件

開発コントロールの一部の機能は、システム開発と保守において想定されるリスクから決められる。そのようなリスクには図2-4のようなものがある。

これらのコントロールは、リスクの発生を防御するための防止コントロール、あるいは、その発生原因を取り除く抑止的なコントロール、さらに、リスクが発生したことを確実に検出できる検出コントロール、その検出されたリスクを取り除き正しい状況に復帰せしめる回復コントロールから構成される。その機能はリスクのタイプ、そのリスクが発生しそうな作業、およびコントロールのタイプによって、洗い出すことができる。詳しくは第3.2章で述べる。

(2) 品質管理基準からの要件

もう一つの機能要件は、与えられた品質管理基準を満足させるためのコントロールで、指定された管理基準への準拠性を検証するためのコントロールである。基本的には、基準例外や違反に対する検出コントロールである。品質管理基準にどのようなものがあるかは、第3.1章で詳しく述べる。

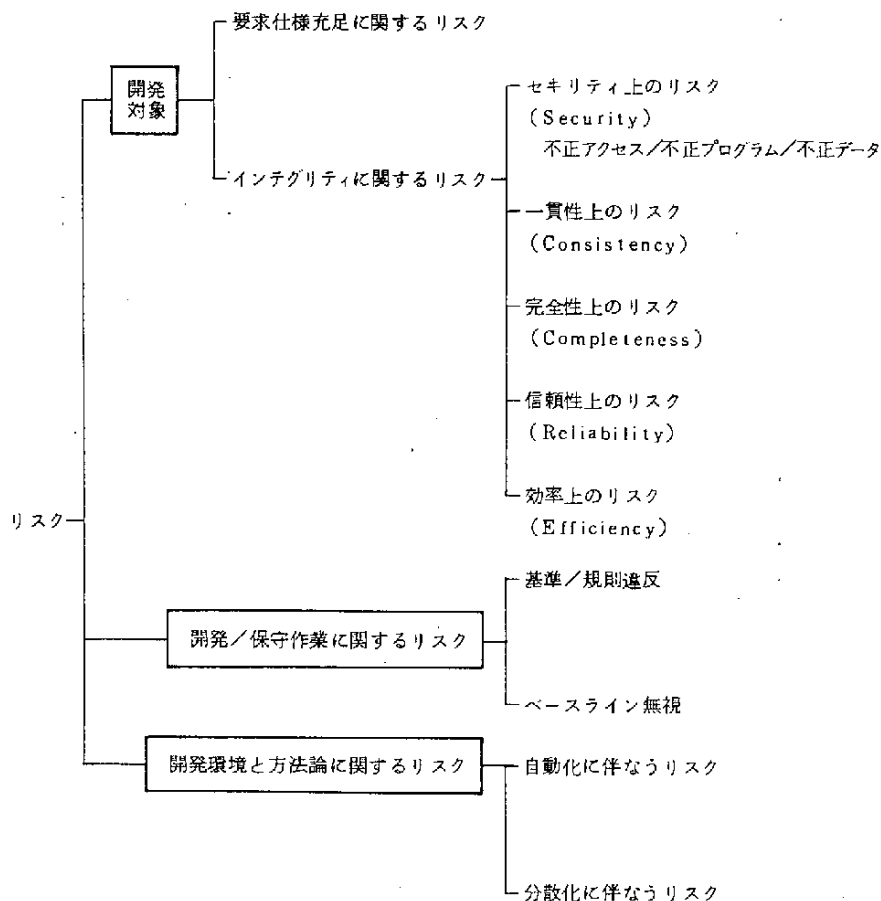


図 2-4 システム開発/保守におけるリスク

(3) コントロール実現上の要件

すでに述べたようにコントロールは、もしコントロールがなくても、リスクを回避でき、品質目標も達成されるならば不要なものといえる。その組み込みと実現は多大の費用を要求する。したがって、コントロールの実現は図 2-5 に示すような要件を満たさなければならない。

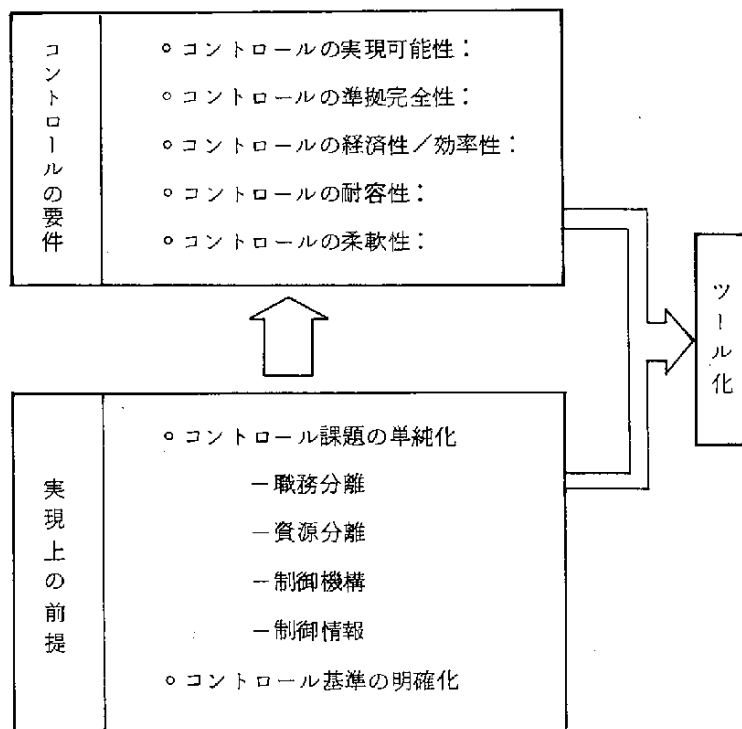


図 2-5 開発コントロール実現のための要件

その一つはコントロールの実現可能性である。つまり、コントロールを実現する技術が確保できなければならない。不正プログラミングやトロイの木馬を検出するためのコントロール技術は現在、確立しているとはいえない。もし、技術的に不可能なコントロール要求があるならば、それに代わる代替コントロールを考えなければならない。

次に、コントロールは確実にそれへの準拠を求められるものでなければならない。つまり、例外を容認したり、違反が見逃されるものであってはならない。

一方で、経済的に妥当な範囲で実現かつ維持できなければならない。最後にそのコントロールの存在が組織の受容度の範囲内になければならない。適用される側の感情的な受容度や、コントロールが持つ弊害も考慮されなければならない。

さらに、一度設置されたコントロールが組織の活動やその変化を妨げるものとならないような柔軟性も考慮されなければならない。

また、確実に、効率よく、肌触りの良いコントロールを実現するためには、それなりの前提条件が整備されなければならない。

その一つは、コントロールの課題そのものを単純化することである。つまり、複雑なコントロールを分解し、単純な課題に展開することである。また、コントロールそのものを最小化するこのも考えられなければならない。つまり、職務権限が結合し、癒着した作業を直接コントロールの対象とすることは、コントロールそのものを複雑にするものとなる。

システム開発や保守のコントロールを仕組みとして実現するとき、その前に

- ① 適切な職務分離
- ② 資源分離
- ③ 監督責任の明確化

などが確保されていなければならない。

さらに、コントロール基準の明確化も重要な課題となる。コントロールの目標を反映した基準を定義するとともに、その見直し保守が行われなければならない。

(4) コントロールの実現レベル

コントロールを実現する方法には、いくつかのレベルが存在する（図2-6参照）。

- ・レベル1：主として人為的なコントロールである。一般に、作業過程をコントロールしやすいように、いくつかの単位に分解してフェーズと呼ばれる塊を作る。コントロールは、先ず、基準または規則が作業に対して与えられ、その準拠性が監督者、管理者によって検証される。これまでのシステム開発に見られたフェーズドアプローチが代表的なものといえる。
- ・レベル2：開発環境がコントロールを組み込んでいるもので、基準や規則に対する準拠性検証の自動化を狙うものである。検出的コントロール、回復的コントロールが充実される。また、そのことが基準違反などに対する抑止的コントロールの実現にも寄与するものとなる。

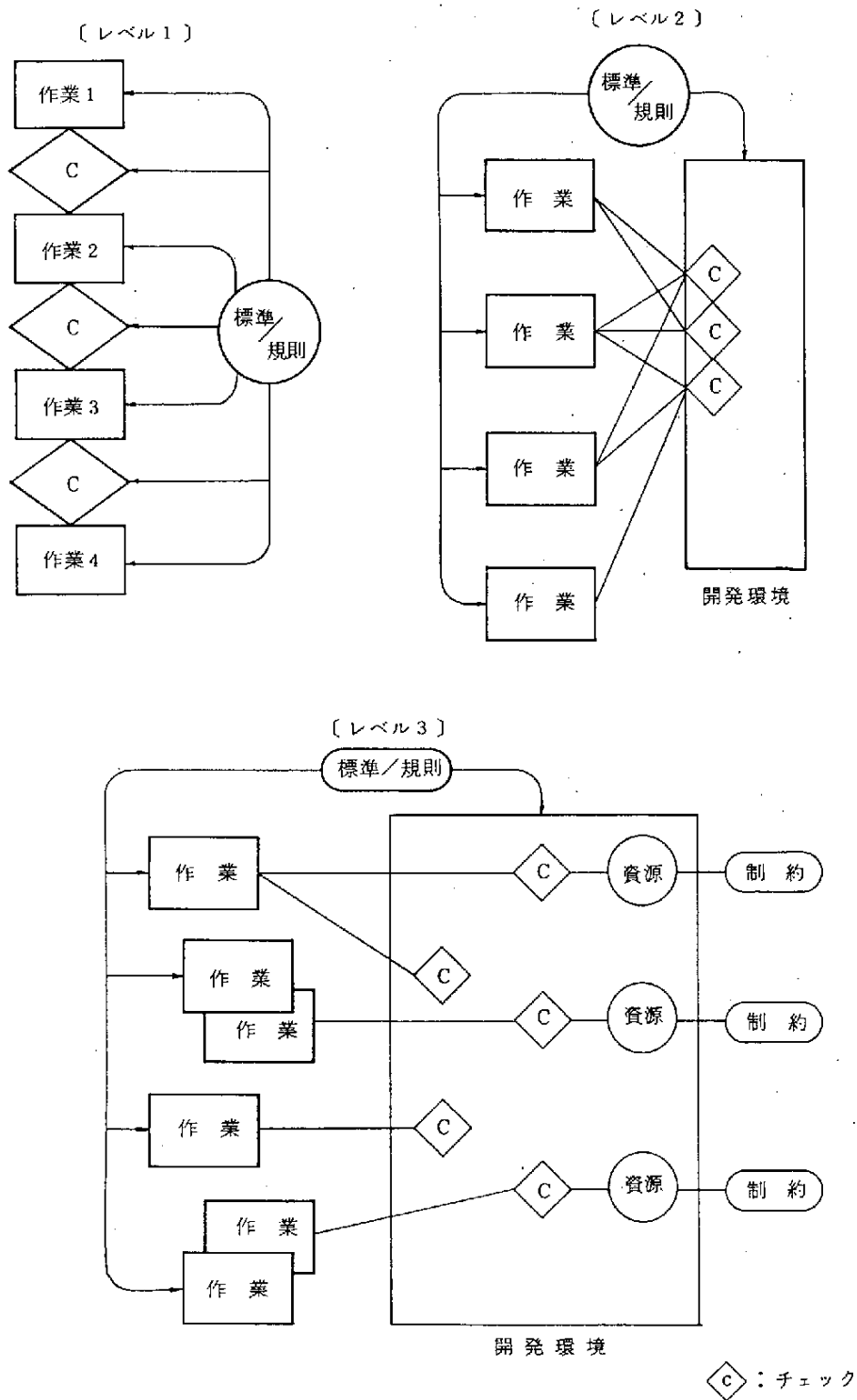


図 2-6 コントロールの実現レベル

- ・レベル3：コントロールの自動化をさらに進めたもので、単に作業の規則への準拠性をチェックするだけでなく、作業の過程で使用する資源の完全性などに関する制約条件（integrity constraints）を持ち、資源の完全性、一貫性、安全性の面からも検証を行うものである。作業に対する基準や規則の多くは資源に対する完全性制約として定義されるので、規則や標準の簡素化にも寄与する。

このように、コントロールの自動化は、その強化と完全性を図る上で、効果的なアプローチであるが、一方で、コントロール自体を隠ぺいする手段となるものでもある。

2.2 システム開発におけるシステム監査

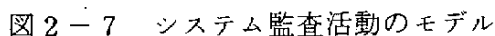
2.2.1 システムライフサイクルにおけるシステム監査のモデル

すでに述べたように、システム監査とは、「システム開発やその運用・保守、つまりシステムライフサイクルにおけるコントロールを評価し、その改善を勧告する」活動である。またその評価と改善の信憑性と完全性を確保するために、一般に、当事者ではない第三者が評価を行うことが求められる。

しかし、システムライフサイクルにおけるコントロールには何段階ものレベルがあり、その全てを一つの監査活動で対応することは事実上不可能である。したがって、多段階のコントロール階層を設定し、コントロール課題とその評価課題を分離することが必要となる。図2-7はそのような考え方からコントロールを階層化したものである。

まず、システム開発作業はプロジェクトという形態で実施される。明確な組織体制を持たなくても、明確な目標と、目標達成に対する責任が定義される限り、プロジェクトと考える。プロジェクトの中でも、目標達成のために様々なコントロールが施行される。一般に、プロジェクト管理と呼ばれるものである。しかし、プロジェクト管理は、プロジェクトの目標達成を最大の課題としたコントロールであり、したがって、そのためのコントロール実施者としてのプロジェクト管理者は当事者に該当し、直接的作業に対する監査執行人とは見なせない。

プロジェクトの進展と、その成果物に対して、第三者的独立性を保持しな



またQAは絶えず、開発コントロールそのものの有効性、妥当性を監視し、その調整を行う評価活動の責任をもつ(図の◎)。代表的なシステムQA活動は次の様なものである。

- ① プロジェクトの発足，終了手続き
- ② 開発されるシステムに対するコントロール要求
- ③ チェックポイントごとのレビュー実施
- ④ 開発成果物（中間，最終）の検証
- ⑤ 開発環境と開発コントロールのレビューと検証
- ⑥ 品質実績データの取得と統計

これに対して，システム監査（SA）は，QAに対して品質の評価基準（図の③）を，また，開発コントロールに対しては品質管理基準（図の②）を提示しながら，システム開発とQAの有機的な相互作用に責任を持つ。

システム監査活動は，本来，特定の課題を持って，不定期に実施される活動である。

システム開発プロジェクトから見れば，QAもSAもプロジェクトの活動や成果物を検証する活動に見える。しかし，SAは，QAが安定した活動を提供することを前提に，つまり，QAをコントロールとみなしてその完全性を評価することで監査目的の多くを達成することができる。したがって，プロジェクトの直接活動に介入しないですみ，より大所，高所からシステム開発のあり方，品質向上のあり方，信頼性確保のあり方を検討できるものとなる。代表的なSA活動は次のようなものとなる。

- ① システム開発，運用，保守に関する標準，規則の制定
- ② 品質管理基準，品質評価基準の設定と見直し
- ③ QA活動の評価と改善勧告
- ④ 特定命題に基づく，特定システムの監査実施
- ⑤ 品質状況に関する経営者への報告

2.2.2 システム監査支援機能

前述のようなモデルで，システム開発，運用におけるコントロールと監査を捉えたとき，どのような支援機能が必要となるかを考えてみたい。

(1) QA支援機能

QAが関心を持ち，その活動の対象とするものは，大きく3つに分けることができる（図2-8参照）。開発・保守作業，作業の成果物，及び開発コントロールである。

QAはこれらの側面に関心を払いながら、全体として品質の確保に責任を持つ。したがって、QA支援機能も、これらの側面に対してどのような支援を行えばよいかを検討することで明らかにできる。その概要は次のようなものとなる。

① 作業に対するQA支援機能

- ・システム開発作業実施状況の把握支援
- ・標準、規則への準拠性検証支援
- ・フェーズ別チェックリスト作成支援
- ・プロジェクト毎の品質収束状況把握支援

② 成果物に対するQA支援機能

- ・品質の測定と分析支援
- ・品質評価基準の網羅性、準拠性検証支援
- ・変更及び保守履歴の取得支援

③ 開発コントロールに関するQA支援機能

- ・開発コントロール準拠率・違反率測定

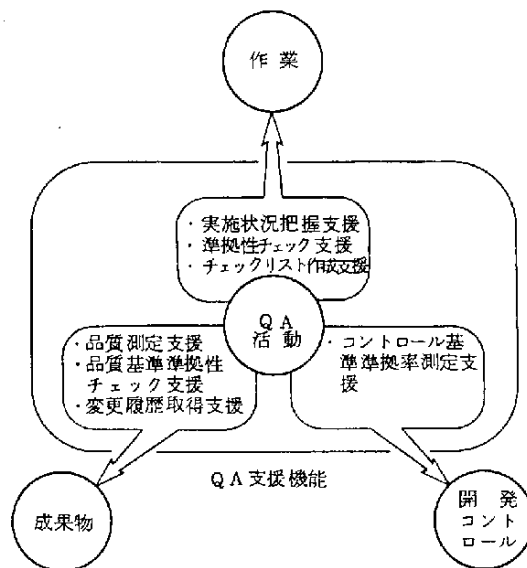


図 2-8 QA支援機能

(2) SA支援機能

SAが関心を持ち、その対象とする領域は、大きく4つに分けて考えることができる。QA, 開発コントロール, 標準・規則, 及び最終成果物である。

これらの領域に対して行われるSA活動で、どのような支援が望まれるかを示したものが図2-9である。

① QAに対するSA支援機能

- ・QA実施状況把握支援
- ・QAに対する品質評価基準策定支援
- ・QA成果測定支援

② 開発コントロールに対するSA支援機能

- ・諸コントロール基準策定支援
- ・品質管理基準策定支援

③ 標準・規則に対するSA支援機能

- ・品質状況分析支援
- ・トラブル, 事故, 障害発生状況分析
- ・標準・規則準拠率分析支援

④ 最終成果物に対するSA支援機能

- ・各種実証的監査支援ツール

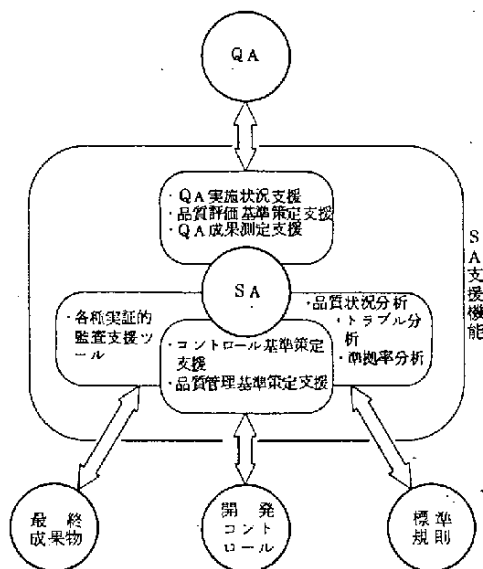


図2-9 SA支援機能

2.3 システム監査支援ツールのフレームワーク

これまで述べてきたような概念に従って、システム開発とその保守におけるシステム監査支援ツールの基本的な構成を考えてみたい。

2.3.1 機能構成

図2-10はシステム開発支援ツールを前提にしたシステム監査支援ツールのフレームワークを示すものである。大きく4つの要素から成り立っている。

- ① 開発支援機能
- ② 開発コントロール
- ③ SA/QA支援機能
- ④ ベース機能

これらの機能のうち、開発支援機能はワークステーションに設置されており、プロジェクトまたは業務毎に複数台、同時に稼働する環境を想定している。

一方、開発コントロール機能、SA/QA支援機能、及びベース機能はホストコンピュータに集中している。システム資源はホストコンピュータ側に集中管理されており、開発資源のうち共有のもの、例えばデータディクショナリなども、中央に管理されている。したがって、つぎのようなシステム開発形態を想定したものとなっている。

- ① ワークステーション側でシステムの設計からプログラムの開発までを行うが、開発されるシステムはすべて中央のホストコンピュータで稼働されるものである。
- ② ワークステーション側には、プロジェクト毎の開発を支援するために、次のような機能が装備されている。
 - ・システム仕様記述支援（ダイアグラム作成支援）
 - ・プログラム作成支援（パターン、部品によるプログラム定義）
 - ・プログラムドキュメンテーション機能
 - ・単体テスト支援
 - ・プロジェクト管理支援

- ③ ワークステーション側の機能の多くは中央のデータディクショナリを参照しなければ作動できない。
- ④ ワークステーションそのもののソフトウェア開発・保守はコントロールの対象としない。
- ⑤ ワークステーションはすべて開発専用機である。

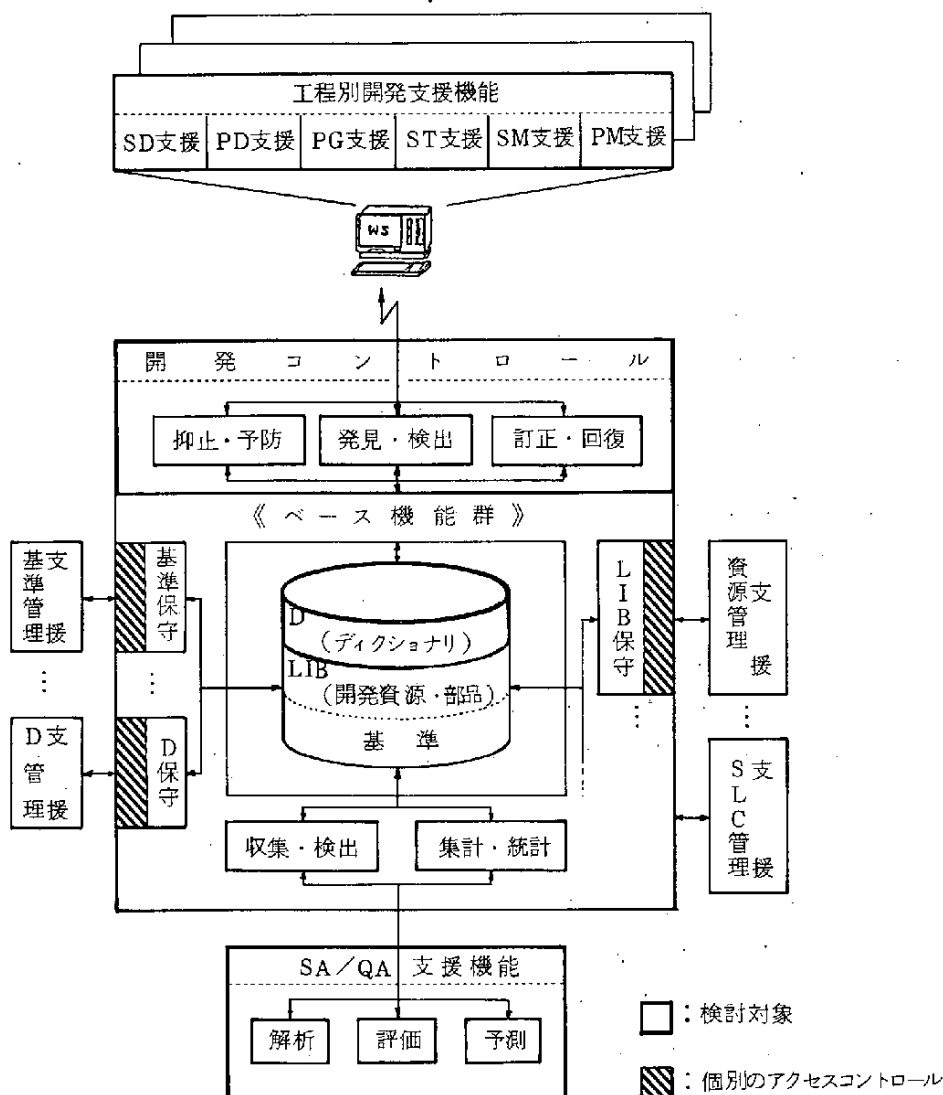


図 2-10 システム監査支援ツールのフレームワーク

2.3.2 開発コントロール

開発コントロールは、ホストコンピュータ側に設置され、ベース機能の支援を受けながら、ワークステーション側で実施されるシステム開発をコントロールする機能である。その特徴は次のようなものである。

- ① ワークステーションによる作業のすべては、開発コントロールを経由しなければ、一切の開発資源にアクセスできない。
- ② すべてのワークステーションの稼働は、開発コントロールの監視のもとで行われる。開発コントロールの承認を受けないワークステーション作業の結果は受け入れられない。ただし、開発コントロールの承認を受けた後に、ワークステーションを切り放し、ローカルモードで実行することはできる。
- ③ 開発コントロールは、主として事前に定義されているコントロール基準により、抑止、防止、検出、回復などの基本コントロール機能を実行するものである。

どのようなコントロール機能を具備すべきかは、第3章で述べる。

2.3.3 監査(SA/QA)支援機能

SA/QAの基本的な機能は、すでに述べたように、コントロールの評価と成果物の実証的な監査、あるいは作業に対する準拠性の監査である。SA/QA支援機能は、開発コントロールとベース機能群に支えられて、主として評価・分析のための機能から構成される。

- (1) 準拠的な監査のために支援機能には次のものが求められる。

- ・ 事実の統計・分析
- ・ 基準や標準との照合
- ・ 例外、異常の検出
- ・ リスクの評価
- ・ 傾向・影響の予測

これらの支援機能はすべて、開発コントロールとベース機能が収集、蓄積した情報に基づいて実行される。

- (2) 実証的な監査のために支援機能には次のものが求められる。

- ・ 成果物のテスト支援
- ・ テスト結果の分析支援

(3) さらにSA/QA活動実施に当たっては、原因分析の範囲、リスク領域や監査対象の絞り込みが、その作業の効率を決めるものとなる。したがって、絞り込みの仮説設定を支援する次のような機能も要求される。

- ・重点チェックリスト作成支援
- ・質問項目作成支援
- ・過去の監査、検査結果の傾向分析

2.3.4 ベース機能群

直接はコントロールや監査活動に寄与する機能ではないが、コントロール及び監査活動の効率的実現、あるいはその完全性を確保する上で不可欠な機能群である。その中核になる機能は開発資源とその管理機能である。ここで開発資源とは次のようなものをさす。

- ① データディクショナリ
- ② プログラム部品
- ③ 仕様書
- ④ 標準・基準
- ⑤ 開発済みプログラム

これらの資源の多くは、複数の業務、あるいは複数のワークステーションによって共有されうるものである。したがって、その整合性、一貫性の維持は基本的命題となり、厳密な管理機能が用意されなくてはならない。

また、資源だけに限らず、開発コントロールSA/QA支援機能側に持った場合、複数箇所に分散されるような共通処理機能、あるいは開発資源に密着してその変化に追随するような処理機能もベース機能を構成するものとなる。

3. システム監査支援ツールの機能要件

本章では、以下の内容について検討を行い、開発コントロール機能とSA/QA機能から構成されるシステム監査支援ツールの機能要件を提案する。

① 品質管理項目と品質評価項目

システムの信頼性・安全性監査に関する品質の詳細な評価項目を整理するに当たって、システム監査と同様な機能と見做すことのできる品質保証(QA)を手掛かりにする。

また、開発作業のコントロールについての項目の検討も、これと同等な機能である品質管理を参考にする。

更に、システム監査と品質保証、開発コントロールと品質管理の役割り分担について整理する。

② システム開発支援ツール機能

現在すでにシステム開発における一貫開発支援ツールは実用化段階に達しており、各メーカーから製品が提供されている。当調査研究で提案するシステム監査支援ツールも現状の一貫支援ツールの持つ機能がある程度前提にして検討した。

前提として設定したツールの機能を中心に一貫開発支援ツールの機能を整理した。

③ 開発コントロール機能

開発コントロール機能を開発のライフサイクルにおけるリスク面からの分析と、リスクのタイプ(抑止, 防止, 検出, 回復)面から系統的に整理しシステム品質を向上させるために必要な要件をまとめる。

④ SA/QA支援機能

システム監査と開発における品質保証の見地からシステムの信頼性/安全性/一貫性を計測・評価するためのSA/QA機能要件を明らかにする。

3.1 品質管理項目と品質評価項目

システム監査の立場に限らず、システム開発における品質保証の立場からでもシステムの信頼性や安全性を保証する必要がある。

この場合、完成したシステムそれ自身の品質を計測し評価しようとする、多くのシステム構成要素(DB, DC, プログラム等)により組立てられてい

る複雑なシステム全体に対して、信頼性や安全性を定量的に定義し、かつ数量的な計測尺度とその計測方法を確立しなければならない（システムのブラックボックス性）。

実際上もソフトウェア製品を専門に開発するメーカを除いて、最終製品を品質評価の目的で検査する独立の組織を持つ例は少ない。

そうするとシステム開発の過程において、目標とする品質を作り込むことが重要となる。実際のシステム開発工程においてどのような開発行為が目標とする品質に影響を与えるか、また開発工程、あるいは開発工程における中間成果物のどこを重点的にレビューすれば品質を保証できるのかと言う着目点が必要となる。

ここでは参考資料のA.に示すようなソフトウェア工学における品質計測（Quality Metrics）技法を基にしたシステム監査のチェックポイントとして、開発工程における品質を「品質管理項目」と「品質評価項目」に分類することを提案し、この考え方によるシステム監査支援ツールの骨組みを明らかにする。

品質管理項目とは、開発作業者が開発作業上において品質の作り込みのために着目しなければいけない項目、すなわち各開発工程における品質上の開発コントロールに対応するチェックポイントを表わす。

品質評価項目とは、開発作業の結果、あるいは中間成果物が目標とする品質をみたましているかどうかを評価・判定するための項目である。

品質管理項目と品質評価項目の選定は、システム開発における要求品質、設計目標品質が尺度として定義されていることが前提となる。参考資料のB.に示す「品質機能展開」がシステム開発の上流から下流に向け品質の作り込みと計測を行う技法であり、本調査研究では「品質機能展開」が当該システム開発に適用されることを前提にしている。

3.1.1 システム開発におけるQC/QAの役割

システム開発において与えられた目標の達成、想定されるリスクの回避のため開発コントロールの必要性について前章で説明した。

監査の前提として開発コントロールが必要なことは、前述（2.1節）の通りである。開発体制における開発コントロールのうち、品質に関する受け口を品質管理（QC）、監査に対応し結果を保証する機能を品質保証（QA）と位置づける。

図3.1-1に示すように、開発工程は多くの複雑なソフトウェア開発作業工程が中間成果物を介してつながっている。そのためシステム監査あるいは品質保証として、最終成果物の実証的な監査を開発工程における準拠的な監査に置き換えようとしても、次のようないくつかの課題がある。

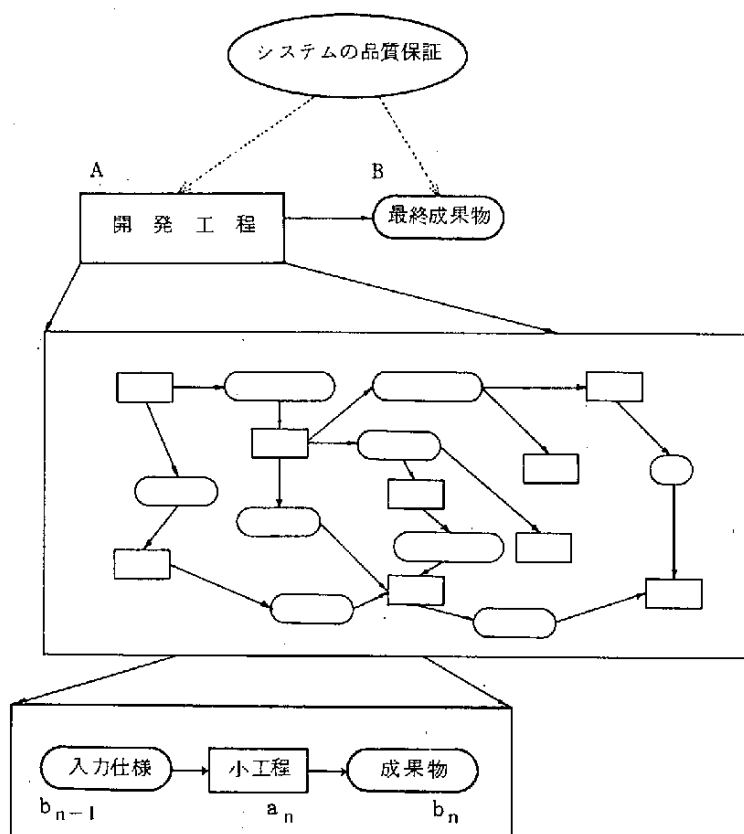


図 3.1 - 1 品質保証の工程展開

① 目標品質の定義（品質要求定義）

当該システムにおいて目標とする品質の種類と、その目標値を定義したり、あるいは優先順位を決めたりする課題。

② 設計品質とその実現（品質管理）

目標とする品質を、ソフトウェアの開発工程においていかに実現して行くか、多数の工程間における一貫性、個々の工程における管理すべき項目

などに関する課題。

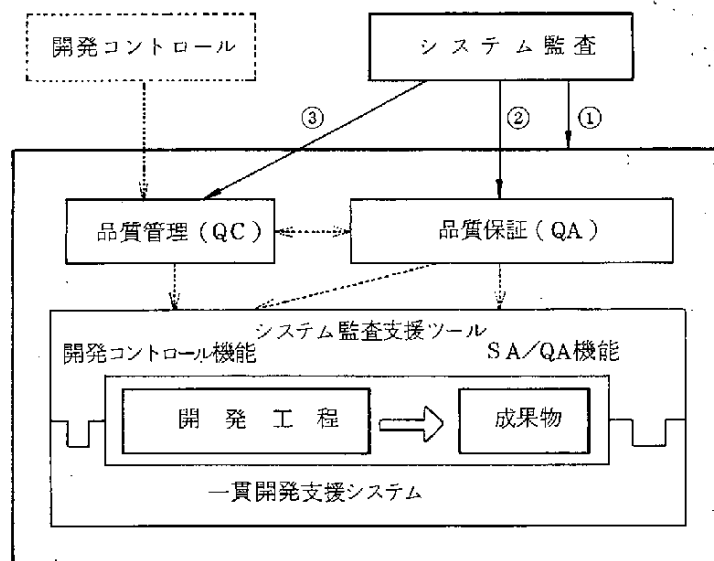
開発方法論や品質管理技法により系統だって定義、実現する必要がある。

③ 品質の計測と評価（品質保証）

工程の成果物や工程における作業の妥当性、準拠性等をいかに計測し、評価しシステムの品質をいかに保証するかという課題。

これら3つの課題は、システム開発のライフサイクルに沿って一連の流れとして展開されることが大切である。

図 3.1 - 2 にシステム監査と QC/QA の関係を示す。



- ① 全般統制の監査
- ② QA を通した準拠的な監査・実証的な監査
- ③ 開発コントロールの監査

図 3.1 - 2 システム監査と QC/QA の関係

この図は当調査研究におけるシステム監査支援ツール、一貫開発支援ツール、開発工程、品質管理、品質保証そしてシステム監査などの関係を概念的に示したものである。

システム監査の立場から見ると、次のようになる。

① 全般統制の監査

監査の対象となるシステム開発がどのような開発コントロールにより、

品質管理／品質保証／システム監査支援ツール／一貫開発支援ツールが運用され、開発工程／成果物が計画／管理／監査されているかの程度をレビューする。

② QAを通したシステム監査

開発工程／成果物に対して信頼性／安全性／一貫性等の監査を行う。このときのデータ収集は、システム監査支援ツールと品質保証を通じて、システム監査の視点により必要とするものを効率的に収集できる。

③ 開発コントロールの監査

前提とした開発コントロールが品質管理に移し込まれ、システム監査支援ツールの開発コントロール機能から開発工程へ、更には成果物へと展開されているかをレビューする。

3.1.2 品質の計測と評価

品質機能展開に沿ってシステム開発工程に対応する品質管理項目が定義され、各工程において開発コントロールが行われたとする。SA/QAの立場から、行われた作業あるいは成果物に対して品質保証を行うため、何等かの計測を行い、何等かの評価基準と対比させ判定を行う必要がある。

品質保証における判定は設計審査と呼ばれ、工程と工程の間にあって、上工程が下工程へ品質上の誤り／漏れを伝搬しないために行われ、審査の結果として合格、問題点の指摘あるいは不合格（もう一度工程に戻し再度設計審査を行う）等の判定を開発行為に対して行う。

品質の計測と評価のプロセスは、図 3.1 - 3 に示すようなモデルが報告されている。

各プロセスは、以下のような内容である。

① 品質要求定義

システム開発における要求定義に相当するもので、システムに要求される品質について定義を行う。この時、品質評価項目に関してはその尺度（品質尺度）、計測値のレイティングレベル、評価基準等を定義する。

② 計測

該当する工程において品質評価項目の計測を行う。それぞれの評価項目とその計測方法、尺度が必要となる。

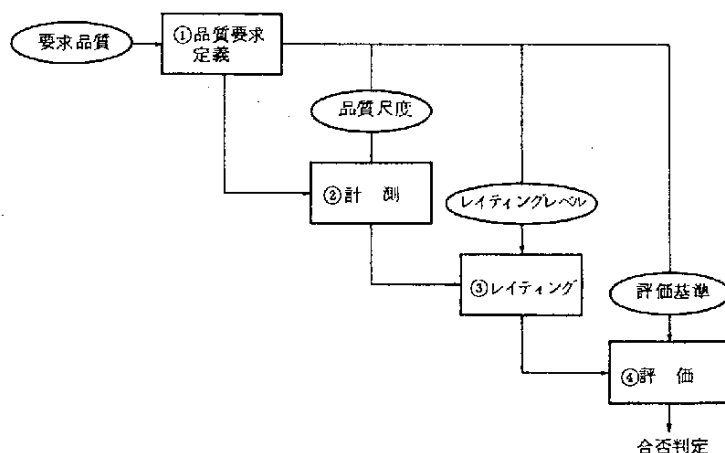


図 3.1 - 3 品質計測・評価プロセス

③ レイティング

計測された値を評価のために設定されたレーティングレベルによりレイティングする。

④ 評価

レイティングされた品質評価項目の評点を評価基準、あるいは管理要求のもとで評価し、該当工程における合否又は改善点の指摘を行う。

品質評価項目はその目的から考えると、品質管理項目とお互いに独立であることが望ましい。しかし、実際のシステム開発において品質評価項目を各工程の成果物に対する妥当性の実証的な監査の監査ポイントとするのは現実的ではない。

例えば基本設計におけるプログラム機能設計工程を例に考えると、成果物として作成されたプログラム構成仕様についての監査ポイントは、“このプログラム構成仕様が妥当であるか？”ということになる。この場合、どうしても、該当するプログラム構成設計における設計上の注意すべき項目（品質管理項目）が正しく実施されているかをチェックする（準拠的な監査）ことになる。このような理由により、図 3.1 - 4 に示すように、品質管理項目と品質評価項目は重なる部分が発生する。重なりをどの位にするかは、品質要求定義の問題でもある。重なりをすくなく品質評価項目を設定するこ

とは、評価・保証のコストを上昇させることにもなる。

品質保証として行われる品質の計測とその評価基準は、個々のシステム開発において目標とする要求品質を満たすようにあらかじめ設定されなければならない。図 3.1 - 3 の①, ②は標準的なメニューから優先順位をつけることにより、また③のレーティングや④の評価については対象となるシステムによって個別に設定する必要がある。

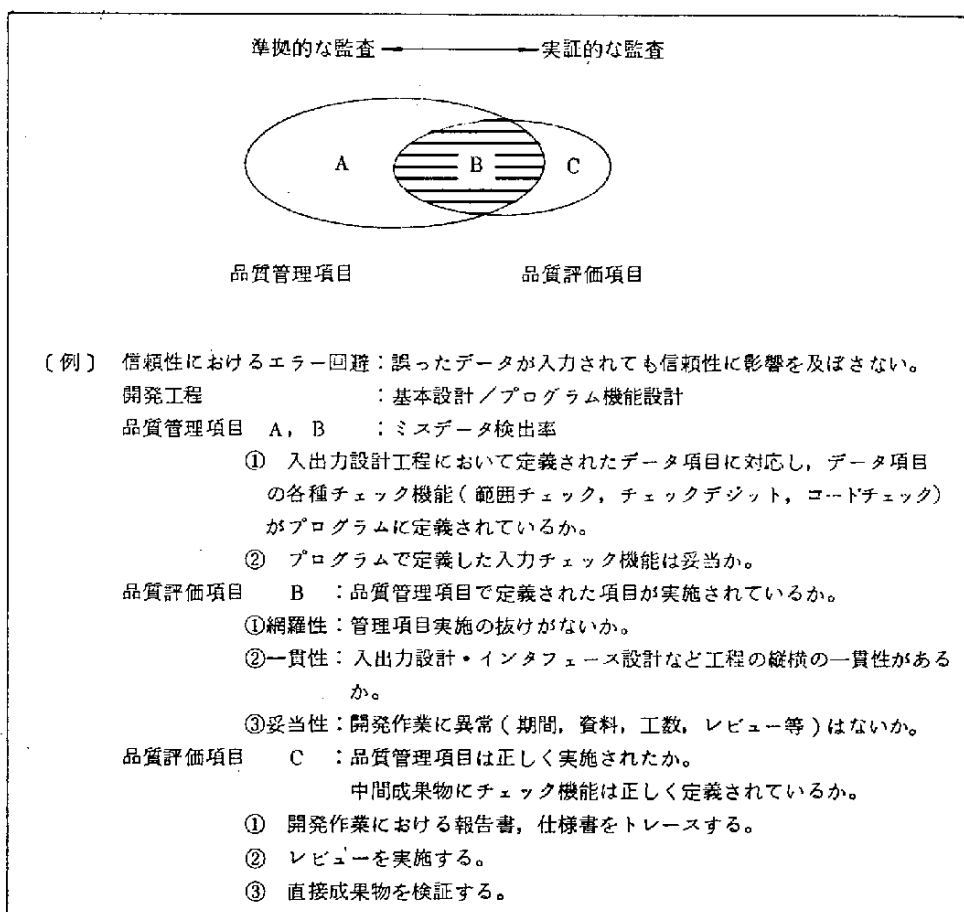


図 3.1 - 4 品質管理項目と品質評価項目の関係

3.1.3 監査のチェックポイント

前項までは、品質管理項目と品質評価項目について説明を行った。ここでは、システム監査の対象として重要なシステムの信頼性、安全性、一貫性を

取り上げ、その整理を行い、実際の開発コントロール機能及びSA/QA機能要件を明らかにするための骨組みを提案する。

(1) 一貫性・完全性

一貫性と信頼性・安全性は、品質評価項目の面から異なった性格のものと考えられる。すなわち、信頼性・安全性は目標システムの品質項目であり、一貫性や完全性は開発過程における品質項目である。

目標となる信頼性や安全性は、工程の展開と共に各工程で作り込まれて行く（図3.1-5）。ところが実際には、多くの工程を複雑に経て最終成果物に組み込まれるため、全工程を通した一貫性や完全性が重要となる。

表3.1-1にIPAの調査報告をもとにした実際の工程別管理項目を代用特性と対比させた表「工程別管理項目と代用特性の対応表」を示した。ここで、代用特性とは、設計品質を表す代用特性のことである（参考資料のB.の図B-1「品質機能展開と展開表」参照）。表3.1-1の管理項目には、×××完全性、×××充足率と表現されているものが多く設定されているが、これが一貫性や完全性に対応する。

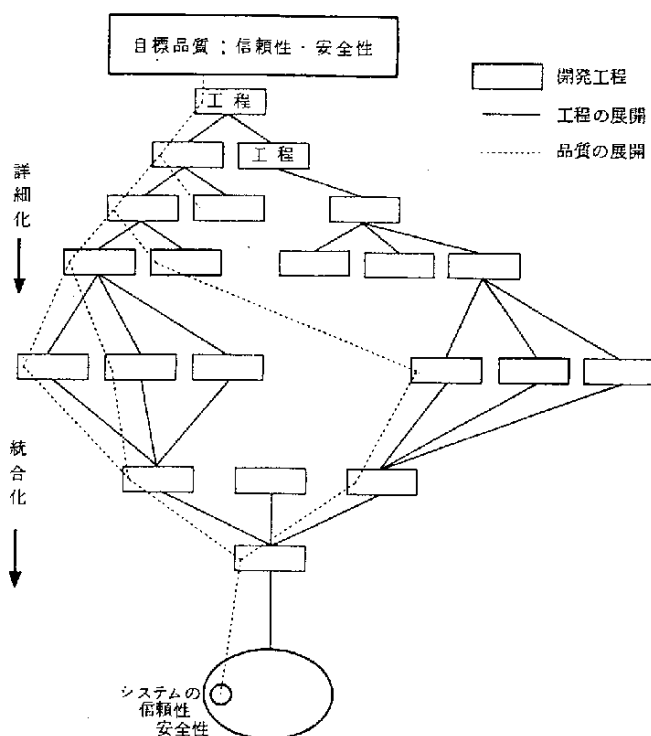


図3.1-5 工程の展開と品質の展開

表 3.1-1 工程別管理項目と代用特性の対応表

工 程		管 理 項 目	代 用 特 性													
			1.データ回復の容易度	2.システム障害発見の容易度	3.フォールバック充実度	4.システム代替処理率	5.非常停止制御能力	6.機密保護の充実度	7.エラー回避率	8.エラー修正率	9.過負荷制御能力	10.障害回復時間	11.プログラム保守容易度	12.テスト充実度	13.設計化技法の適用率	14.の有害障害対策マニュアル
システム設計	システム機能設計	バックアップ効率	○			○										
		代替処理率				○										
		障害対策充足率	○		○	○										○
		システム障害停止率			○											
	ソフトウェア構成設計	機密保護要件充足率						○								
		情報保護要件充足率						○								
		過負荷対策充足率								○						
		障害回復所要時間									○					
	ハードウェア構成設計	システム障害検出率		○												
		使用可能ハード所要数				○										
		使用ハード統一率				○										
基本設計	プログラム機能設計	プログラム機能設計完全性	○	○			○		○	○					○	○
		システム構造完全性													○	
		規格遵守率													○	○
		誤操作検出率	○	○			○		○	○						
		使用言語統一率										○		○		
		ミスデータ検出率							○	○						
		非常停止要件充足度					○									
	インタフェース設計	プログラム間インタフェース数			○						○	○				
		プログラム間クリティカルパス数			○						○	○				
		端末仕様充足度									○					
		平均データ伝送度									○					
	入出力設計	入出力設計完全性								○	○					
		データ項目定義率								○	○					
		ファイル設計完全性							○							
	テスト仕様書作成	テスト効率化率												○		
		テストカバレッジ率												○		
		バグ検出率												○		
	詳細設計	モジュール設計	共通モジュール化率										○		○	
モジュール独立性													○		○	
モジュール階層レベル遵守率													○		○	
モジュール間インタフェース設計		モジュール間パラメータ完全性											○		○	
		部品使用率											○		○	
単体テスト/結合テスト	コーディング	プログラム作成基準遵守率											○		○	
		モジュールテスト完了率													○	
	結合テスト	プログラム機能確認度		○	○				○	○		○				○
		プログラム不良検出度												○		
		機密保護要件確認度							○							

出典：I.P.A調査研究報告書

一貫性・完全性の品質評価項目と品質管理項目を整理すると、表 3.1 - 2 に示すような監査のチェックポイントが考えられる。

表 3.1 - 2 一貫性・完全性に関する監査ポイント

品質評価項目			品質管理項目	計測・評価尺度		評価・判定
一貫性・完全性	全体の一貫性	信頼性	- 代替処理の充足率 - 障害対策の充足度	実証性	成果物の妥当性	全工程を通じて、信頼性・安全性等重要な品質に対して漏れ、抜けが起こらないように優先順位付けし、評価基準を決める。
					点検項目の妥当性	
	各工程の完全性	工程対応	- 機密保護要件充足率 - 情報保護要件充足率	準拠性	管理項目の 実施点検数 全点検数	
一貫性・完全性	全体の一貫性	信頼性	- プログラム機能設計完全性 - 誤操作検出率 - ミスデータ検出率	実証性	成果物の妥当性	工程自身の重要度、評価項目の重要度、工程実施者等を考慮し、評価基準を決める。
					点検項目の妥当性	
	各工程の完全性	工程対応	- ファイル設計完全性 - 開発用ライブラリのアクセス権	準拠性	管理項目の 実施点検数 全点検数	

(a) 品質評価項目

- ① 全体の一貫性……開発工程全体を通して漏れが無いかな。
さらに信頼性・安全性に関する項目に分解。
- ② 各工程の完全性……各工程における点検。
それぞれの工程における信頼性・安全性に分解。

(b) 品質管理項目

何等かの管理項目が評価項目に対応付けられる。関係は 1 対 1 ではなく、1 対 n になることもあるが、一般的に評価項目は管理項目に含まれる。

リスク分析→工程における作業上の着眼点（管理項目）
→作業及び成果物の妥当性の確認（評価項目）

(c) 計測・評価尺度

- ① 計測の方法……実証性／準拠性

② 計測の対象……工程／仕様書／ソフトウェア

③ 尺 度……2段階尺度（Yes/No）／4段階尺度／度数 等々

(d) 評価・判定

計測された評価尺度をもとに、予め設定された評価基準にしたがって判定を行う。

このような点に着目して、品質評価項目を定義することができる。実際のシステム開発においては、工程の数・管理項目の数とも大きくなる傾向があるので、一貫開発支援ツール／開発コントロール機能／SA・QA支援機能（システム監査支援ツール）及び品質管理／品質保証の役割等の配分により設定する必要がある。

(2) 信頼性・安全性

開発工程における一貫性や完全性を別として、信頼性・安全性の管理項目／評価項目を考える。着目点は、信頼性・安全性に対応するシステムの機能と、その機能に対応する後工程の妥当性、そして最終成果物を確認するテストである。

信頼性・安全性に関する評価項目と管理項目の骨組みを、表3.1-3に示す。

表3.1-3 信頼性・安全性に関する監査ポイント

品質評価項目		品質管理項目	計測・評価尺度		評 価 ・ 判 定
信 頼 性 ・ 安 全 性	開 発 工 程	定 義 程 序 ・障害対策充足度 ・機密保護要件充足度 ・誤操作検出率 ・ミスデータ検出率 ・ファイル設計完全性 ・モジュールテスト完了率	実 証 性	成果物の妥当性	一貫性・完全性を除いた工程における信頼性・安全性を決める作業とその管理項目を優先順位付けし、評価基準を決める。
				点検項目の妥当性	
			準 拠 性	管理項目の実施点検数 全 点 検 数	
	成 果 物	・デバッグ ・テスト ・開発者 検 査 ・プログラム機能確認度 ・テストカバレッジ ・機密保護要件充足度	実 証 性	テストの妥当性	開発作業の中での作業の確認工程として位置付け、評価基準を決める。
			準 拠 性	テストの網羅性	
		・第三者検査 ・受入検査		独自のテスト項目	独立性の高い評価・判定。

(a) 品質評価項目

- ① 開発工程に対するもの……開発工程は定義工程、設計工程と製造工程に分けられる。定義・設計工程では信頼性・安全性に対応する機能展開に着目し、製造工程ではその機能が確実に展開されることに着目する。
- ② 成果物に対するもの……開発工程の後半はデバッグ、テストと第三者による検査に分けられる。

(b) 品質管理項目

一貫性・完全性と同様に管理項目と対応する。但し、第三者による検査については別に設定されるため、開発工程における管理項目とは対応しない。

(c) 計測・評価尺度

一般的な項目は一貫性・完全性における項目と同じであるが、以下の点に着目する必要がある。

- ① 開発工程……管理項目の準拠性だけでは無く、中間成果物である仕様書自身に対する実証的な監査的な計測を行う必要がある。
- ② デバッグ／テスト……現状のシステム開発ではこのテスト工程が信頼性に与える影響は大きい。テストの網羅性、妥当性からテスト結果の統計的な情報などを活用する。
- ③ 第三者検査……開発工程・テスト工程を経て、最終的な確認、工程を通して見て、評価・判定で問題があったものの再点検をする。

(d) 評価・判定

- ① 開発工程……優先順位を付ける。特に重要なものは実証的な監査の結果を取り入れる等の注意が必要である。
- ② デバッグ／テスト……テスト工程がバグの修正工程では無く、内部監査として機能するように、予め評価・判定

基準を設定しておく必要がある。

③ 第三者検査……………独立性の高い評価・判定を行う。

一貫性・完全性の部分を除いた信頼性・安全性は設計行為の妥当性を監査することになる。この問題は本調査研究で提案しようとしている開発コントロール機能、SA/QA機能及びそれらを開発側の役割分担として総括する品質管理/品質保証により解決しようとするものである。

また一貫開発支援ツールの進歩、それを支える開発方法論も大きく影響を与える。これらの相互関係によりシステム監査のチェックポイントとなる品質管理項目/品質評価項目について実際のシステム開発において定義される。ここでは、その骨組みについて整理を行った。

〔参考資料〕

A. 品質の分類と尺度

(1) 品質モデルと計測に関する動向

ソフトウェアの品質特性を分類し計量的に評価する問題は、ソフトウェア工学における品質保証 (Quality Assurance) /ソフトウェア計測 (Software Metrics) として知られている。

1976年のB.W.Boehmによる“Quantitative evaluation of software quality”が最初に報告されたソフトウェア品質の分類として知られており、その後多くの研究が行われ、最初は米国の宇宙開発と軍関係から実用化された。また商用システム開発にも徐々に適用が進み、実用レベルの計測技法が現在までに多く報告されている。最近の動向としてはIEEEとISOでソフトウェアの品質評価に関する標準化が検討されつつある。

代表的な品質計測の技法を表A-1に示す。

品質保証の面から品質計測を分類すると、やはり次の3つの要素が考えられる。

① 品質の分類

ソフトウェアの品質をどのように分類するか。古くはBoehmやMcCallの品質尺度が知られている。

② 品質の実現法

目標とする品質項目を実際のソフトウェア開発工程においてどのように実現するかを主に工程における品質管理項目として捕らえる。

ハードウェアの品質管理技法として知られている品質機能展開などがこれに相当する。

③ 品質の計測と評価

品質保証の立場から、ソフトウェアの客観的評価を行い品質を保証する。

本調査研究では、詳細資料の入手し易さの関係からIPAの調査結果を取り上げることにした。ただしIPAの調査では③の品質計測と評価について不足しているのでその部分はSQMAT (NEC) を参考にした。

(2) 品質の分類

表A-1に示した品質計測の代表例における分類方法は次のように整理できる。

項番7以外の報告では、品質の分類を網羅的に定義しようとしている。分類の方法としては項番6に上げた2つの標準化案が代表的なものである。

表A-1 品質計測技法の代表例

項番	品質モデル/計測技法	特 徴
1	Boehm の品質尺度	- 品質特性を階層化して分類している。 - 具体的な計量化については論じていない。
2	WaltersとMcCall の品質モデル	1977年、11項目のFactorと25項目の尺度を提案した。
3	品質保証の標準化 { MIL-S-52779A IEEE-Std-730	- 宇宙開発や軍関係のソフトウェア開発における品質保証に関して規定したもの。 - いくつかの企業において実施された報告がある。
4	SQM (Software Quality Metrics) 米国METRIQS 社	- Murineによる実用化に向けた技法で、12項目のFactorと23項目のCriteriaに展開している。 - METRIQS 社はこの技法を基に実用化をおこなっている。
5	SQMAT (Software Quality Measurement and Assurance Technology) NEC	品質保証技術とソフトウェア品質計測をプロジェクト規模やソフトウェアの種類に依存することなく適用できるように開発されたもの。
6	{ IEEE 標準化案 ISO 標準化案	現在、共に検討中の標準化案で、IEEEの案はSQMに近いFactor/Criteriaと言う分類で、ISOの案はCharacteristic/Sub-Characteristicという分類である。
7	IPA: ソフトウェア品質評価モデル	ソフトウェアの品質評価尺度と品質を実現するための技法として品質管理における「品質機能展開」をソフトウェア開発に適用している。

① I S O 標準化案

品質の分類を、木構造として分類するやり方で品質特性 (Quality Characteristic)、副品質特性 (Quality Sub-Characteristic)、メトリックス (Metrics) と定義している。図 A - 1 に例を示す。

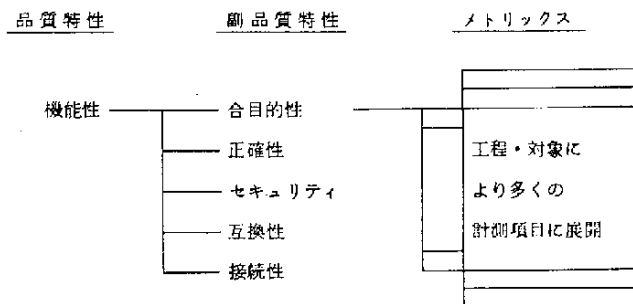


図 A - 1 I S O の品質分類案

② I E E E の標準化案

McCall の品質モデルからMurineのSQM と発展してきた分類方法で、Factor/Criteria という別々の分類に分け、両者を対応付けて分類する。一見複雑に見えるが、この目的はソフトウェアの品質を利用者の側から見た要求品質と、設計者の側から見た設計目標品質に対応させることである。図 A - 2 に例を示す。

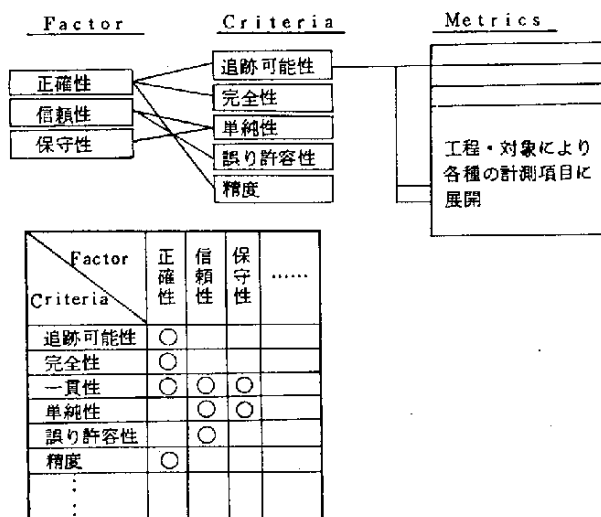


図 A - 2 品質要求尺度と品質設計尺度の分類例（一部）

③ 項番7の分類

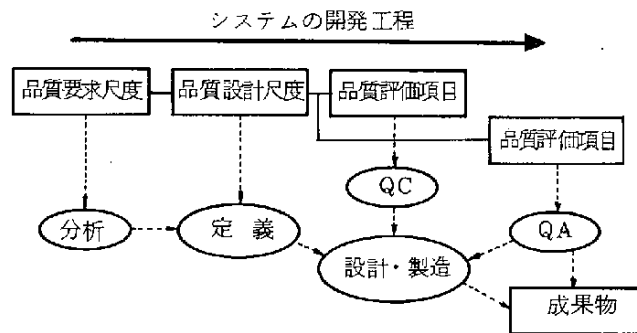
I P Aの調査報告では、予め品質分類のわくを定義するのではなく、分析を行ってゆく過程と手順を定義している。品質は「要求品質の原始情報」と呼ばれ、K J法のような手法により3段階にまとめられる。

(3) 監査における品質の分類

システム監査の立場からも、システムの品質を定義する必要がある。この場合システム監査が開発工程に対しても行われる必要性がある。

システムの品質をシステム開発工程にしたがって考えると、図A-3に示すような関係になる。システムの分析段階においては実現すべき品質を明らかにする必要がある、システムを定義する段階においてはシステムの設計目標となる品質を定義する必要がある。また設計・製造工程においては、それぞれの工程における品質の作り込みをコントロールする管理項目が必要となる。品質保証としては作り込まれた品質を計測・評価するための品質尺度が必要となる。

この様に複雑な工程に展開されて行く実際のシステム開発において必要な品質の分類と尺度は系統だった定義が要求される。



図A-3 品質尺度と開発工程

図A-4の「品質の分類」に示すようにそれぞれの対応が必要となる。この場合、単なる品質の分類より、開発工程の中でどのように品質を作り込んで行くかが重要となる。このための品質管理における技法としては、品質機能展開がよく知られている。I P Aの調査でもソフトウェア開発に品質機能展開を適用することを品質の計測と同様に提案している。

システム監査の前提に開発コントロールが必要なように、品質保証として品

質の計測・評価の前提に品質機能展開があげられる。

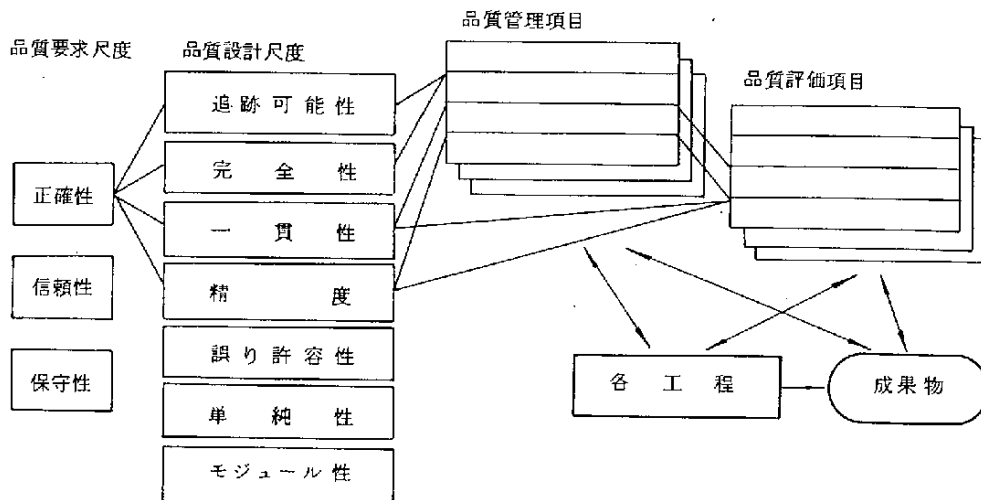


図 A - 4 品質の分類

B. 品質の実現法（品質機能展開）

ソフトウェアの品質計測は、開発の結果としての品質を計測するより、開発段階において目標とする品質を作り込んで行くための品質尺度ととらえることが出来る。このように品質を作り込んで行こうとする積極的な品質計測の考え方が「日本の品質管理」とも呼ばれる考え方であり、代表的な技法に「品質機能展開」がある。

「品質機能展開」とはハードウェアの分野、特に新製品の開発時に有効であることが広く認められている。しかしソフトウェア開発においてはまだ始まったばかりの技法であるが、ソフトウェア開発がハードウェアにおける新製品開発（繰返生産と対比して）に類似していることから有効な技法と考えられている。

I P A の調査報告でも品質計測を品質機能展開の中でとらえており、ここでは、その概要を紹介する（詳細については参考文献 8 を参照のこと）。

図 B - 1 の「品質機能展開と展開表」では、品質をどのように展開して行くかを示した。

以下では、図 B - 1 の各表を説明することにする。

① 要求品質展開表

要求品質			重要度
1.	1.1		
	1.2		

② 代用特性展開表（品質特性展開表）

代用特性		単位	目標
1.	1.1		
	1.2		

③ 品質表

		代用特性				
要求品質						
1.	1.1		○			
			○	△		
	1.2					

④ 開発工程表

工 程	入 力	成果物

⑤ 品質管理工程表

管 理 項 目		代 用 特 性						計 算 式	対 象	点 検	担 当 者 の 評 価
工 程											

図 B - 1 品質機能展開と展開表

① 要求品質展開表（品質要求尺度）

システムに要求される品質を分類整理する。システムに対するユーザの顕在的，潜在的な要求項目（原始情報）を整理，分類し，3次項目までにまと

める。そして要求品質の重要度を設定し、要求品質展開表を完成する。

② 代用特性展開表（品質設計尺度）

ここで用いる代用特性は、品質特性（ソフトウェアの品質を評価する尺度）とソフトウェア機能（要求品質を満たすために必要なソフトウェアの機能）の両方を指す。

システムを開発する技術部門が中心となって作成する。各代用特性項目はシステム設計における品質設計目標に相当するので、正確に定義付けする必要がある。

③ 品質表（品質要求尺度と品質設計尺度の対応表）

ユーザの要求する真の品質を言語表現によって体系化し、これと代用特性との関連を示し、ユーザの要求を代用特性に転換し、品質設計を行っていくための表と定義されている。品質要求尺度と品質設計尺度の対応付けを行うための表である。

④ 開発工程表

システム開発におけるそれぞれの開発工程を木構造状に展開し、それぞれの工程での入力と作成される成果物を明記し、工程を網羅的に定義する。

⑤ 品質管理工程表

④の開発工程表で定義した各工程ごとに、②の代用特性展開表を対応させ、その工程において代用特性で表される設計目標品質を作り込むときの管理上の観点をまとめた表である。

工程における作業点検として、管理項目対応に定量的な計算式と管理対象物を定義しておき、評価を行う。

〔参考文献〕

- 1) Boehm, B.W. and others (1976) : Quantitative Evaluation of Software Quality, 2nd ICSE, Vol.2, pp. 592-605.
- 2) McCall, J.A. and others (1977) : Factors in Software Quality, RADC TR-77-369.
- 3) Murine, G.E. (1980) : Applying Software Quality Metrics in the Requirements Analysis Phase of a Distributive System,

Proceedings from Minnow Brook Conference.

- 4) Murine, G.E. and others (1983) : Measuring Software Product Quality, Quality Progress, Vol.17, No.5, pp.16-20.
- 5) 東, 他 (1986) : ソフトウェア品質計測/保証技術 (SQMAT), 品質, Vol. 16, No.1, pp.79-84.
- 6) 東, 他 (1987) : 品質評価尺度と計測技術, 標準化と品質管理, Vol.40, No.8, pp.63-75.
- 7) 寺本 (1982) : ソフトウェア・メトリクス, 第6回ソフトウェア工学国際会議講習会資料, 1982年9月.
- 8) 情報処理振興事業協会技術センター (1987) : ソフトウェア品質評価モデルの調査報告書 (その2), 昭和62年3月.

3.2 システム開発支援ツール機能

今回の調査研究では, システム監査支援ツールの検討にあたり, 近未来をターゲットとするシステム開発支援ツールの持つ機能をもとにして, システム監査機能として具備すべきものを洗い出すというアプローチをとった。

検討に先立ち, システム開発支援ツールとはどのようなものであるか, とくに「近未来のシステム開発支援ツール」の機能要件はどのようなものであるかについて, 明らかにした。

3.2.1 システム開発支援ツールの歴史・遷移

ソフトウェアの生産性を向上するための手段として各種ツールが考案されてきた。そしてツールを使うことにより, ソフトウェアの開発方法自体が変化してきた。

膨大な規模のシステムを開発するにあたり, コンピュータメーカーおよび有力ソフトハウス各社では, 昔からソフトウェアの開発生産性をあげるためのシステム開発方法論を展開してきている。システム開発方法論の適用方法も, 初期の頃は各種帳票の設定・運用等の作業標準が中心であったが, ツールの

発達にともなう、ツールを利用した方法に変わってきた。

ツールによる支援は、最初の頃は、テキストエディタやプログラムライブラリアンなど、おもにプログラム製造工程が中心の「プログラム開発支援ツール」であった。その後、検査工程など、次第に対象範囲を広げていった。しかし、システム開発方法論に対比すると、個々のツールをばらばらに適用している状態であり、全体としてのまとまりを持ったツール群として位置づけられるようになったのは最近になってからである。

現在主流となっているのは、「一貫開発支援システム」と呼ばれているもので、システム開発をいくつかの単位（工程）に分けて作業を進めていくフェーズドアプローチにおいて、システム分析やシステム設計などの上流工程から、プログラム製造・テストを経て、システム運用・保守の下流工程に至るまでの全工程を関連づけて、一連のツール群で支援するものである。プログラム製造工程以降の方法についてはかなり確立されてきていて、各社とも機能的には似たようなものであるが、上流工程であるシステム分析やシステム設計の支援はまだ始まったばかりで、さまざまなアプローチが行われている。一部の「一貫開発支援システム」の中には、部分的にプロトタイピング技法を採用しているものもある。

最近ではこれらの「一貫開発支援システム」の機能の一部をワークステーション（WS）上に搭載し、分散処理により高い操作性・応答性を確保する動きがある。

図 3.2-1 に開発支援ツールの機能面およびシステム監査的観点から見た遷移を示す。この中で「近未来」と「未来」については明確な定義を与えていないが、感覚的に現在の延長上にあると考えられるのが「近未来」であり、何らかのドラスティックな変化が生じたものを「未来」と、ここでは位置づけている。

開発支援ツールを使うことにより、開発工程の各フェーズが圧縮され、場合によっては、2つ以上のフェーズがひとつになったり、あるフェーズが不要になるという、工程の統廃合がおこなわれるようになることに注意する必要がある。

項目 \ 年代	過 去	現 在	近 未 来	未 来
利 用 者	EDP要員			エンドユーザ
背景となる開発方法論	Water Fall Model			Prototyping, Object Oriented Approach Data Flow Model etc.
ツールイメージ	個別ツール	一貫支援システム (部分的)	(総合的)Σ, SOFPIA, etc.	4GL
DD / DS	なし	受動的DD/D		能動的DD/D
プログラム設計	フローチャート	木構造チャート (SPD, HCP, YACII, PAD, etc.)		?
部品化	マクロ	半部品	部品	
パターン化	ソーススケルトン		パターンスケルトン	
プログラム生成	手作業		半自動	自動
ドキュメント作成	手書き	半自動		自動
開発環境	ホスト中心			ワークステーション化, 分散化
	セルフ開発			クロス開発

図 3.2 - 1 システム開発支援ツールの遷移

3.2.2 システム開発支援ツールの現状と調査研究対象の選定

システム監査支援ツールの検討の前提となる開発支援ツールについて調査し、調査研究対象を選定する。

現在、各メーカーやソフトハウスで提供されている、あるいは提供しようとしている開発支援ツールは、そのほとんどが各社の提案するシステム開発方法論とリンクしている。代表的なものを、表3.2-1に示す。3.2.1でも述べたとおり、最近の傾向として開発支援ツールの機能の一部がワークステーションに移植されるようになってきた（ワークベンチ化）。その構成の一例を、図3.2-2に示す。これらは、各社各様の特徴をもち、特定の目的に対しては、それぞれ一長一短がある。

システム監査支援ツールの検討のベースとなる「近未来の開発支援ツール」としては、開発支援ツールを一般的な開発支援機能として抽象化して考える方法もあるが、それでは具体的な提言を得るのは難しい。

そこで、ここでは各社固有のシステム開発方法論にしばられない「Σシステム」の「事務処理分野ツール」（図3.2-3）を中心に、それに一般的な一貫開発支援システムとして必要な機能を補足することにより、具体的な開発支援ツールのモデルを想定した。

〔Σツールの持つ機能〕 （IPA資料「事務処理分野開発環境の構築」Σ310000-F4-0100より）

- ・ソフトウェア設計の機械化
- ・成果物の再利用
- ・パターン、部品を利用したCOBOLソースプログラム生成
- ・チャートを利用したCOBOLソースプログラム生成

〔補足した機能〕

おもにホストマシン側でおこなう総合的な管理機能が中心

- ・DD/DS関連機能
- ・総合ライブラリ管理機能

表 3.2 - 1 システム開発方法論とそれに基づいた開発支援ツール

会 社 名	方 法 論	支 援 ツ ール	支援ツールの稼動するハードウェア
東 芝	MYSTAR		DPシリーズ, Vシリーズ
日本アイ・ビー・エム	ADSG(Application Development Standardization Guide)	ADSSBX(Application Development Support System Base extension) CSP(Cross Systems Product)	30xx, 43xx, 93xx(OSはMVS) 同上(DOS/VSE, MVS, VM), システム/ 3Xシリーズ(SSX/VSE, DPPX)
日本電気	SEA/1 (Software Engineering Architecture/One)		ACOSシリーズ
	STEPS(Standardized Technology and Engineering for Programming Support)	SOPPIA(Software Productivity Improvement Aid) IDL II(Integrated Data oriented Language II)	N5200/5300シリーズ
日本ユニバック (ユニシス)	RSDM(Reliable System/Software Development Method)	TSX1100(Total Support Executive system for UNIVAC Series 1100)	1100 シリーズ
パロース (ユニシス)	SIM(Software Implimentational Methodology)	InterPro	Bシリーズ
	LSM(LINC Systems Methodology) を開発中	LINC II (Logic & Information Network Compiler II)	
日立製作所	HIFACE(Hitachi Phased Approach for High Productive Computer System's Engineering)	EAGLE2(Effective Approach To Achieving High Level Software Productivity) SEWB(Software Engineering Workbench)	Mシリーズ, Lシリーズ 2050 シリーズ
富 士 通	SDAS (Systems Development Architecture & Support Facilities)		Mシリーズ, Kシリーズ, Gシリーズ, 9450シリーズ
	SDEM(Software Development Engineering Methodology)	ADAMS(Application Development and Management System), CASET, YPS	
三菱電機		SWEET(Software Engineering Extended Tools)	EXシリーズ
アーサーアンダーセン	FOUNDATION		IBM PC, IBM 30XX, 43XX, 93XX(OSはMVS/XA)
	METHOD/1	DESIGN/1, INSTALL/1, METHOD/1	
住商コンピューター サービス	ASPECT(Advanced Software Productivity Enhancement Concept)	ALICE(Aid for software Life Cycle Envelope)	IBM 30XX, 43XX(OSはMVS/SP, XA), IBM5550シリーズ
トッパン・ムーアシステムズ (英BISアプライド・システムズが開発)	TMS/MODUS	TMS/IPSE(Toppan Moore Systems/Integrated Project Support Environment)	NCR TOWER, IBM PC
日本システミックス (米MBAが開発)	PRIDE-ASDM (Profitable Information by Design)	IRM, ADF, AIM	IBM 30XX(MVS/XA), FACOM Mシリーズ (F4), HITACシリーズ(VOS3), ACOSシリーズ (ACOS-4 MVP), UNIVAC1100, DEC VAX (VMS)

MYSTARは方法論と支援ツールの総称。SEA/1, SDAS, FOUNDATIONは方法論、支援ツールなどを包含した体系の名称 (日経コンピュータ 1987年11月23日号, No.161より)

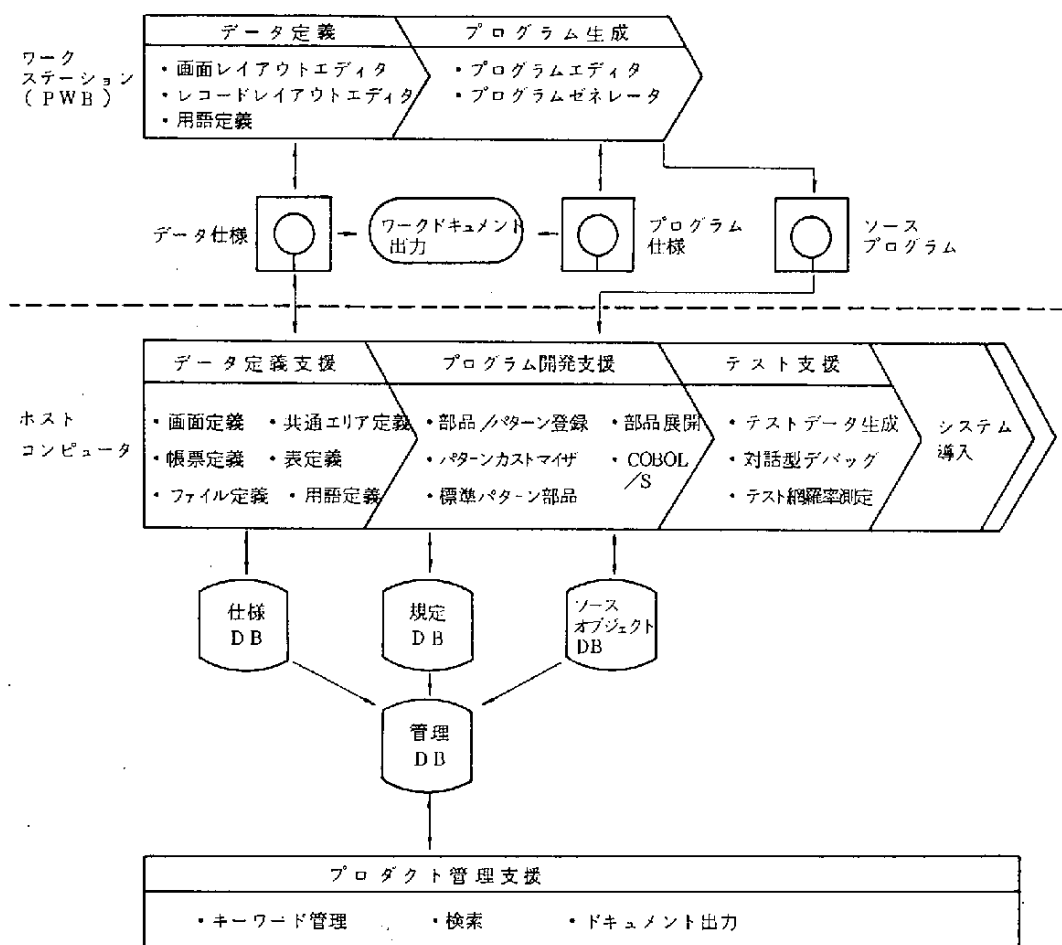


図 3.2 - 2 ソフトウェア開発支援システムSOFPIAの構成
(日本電気)

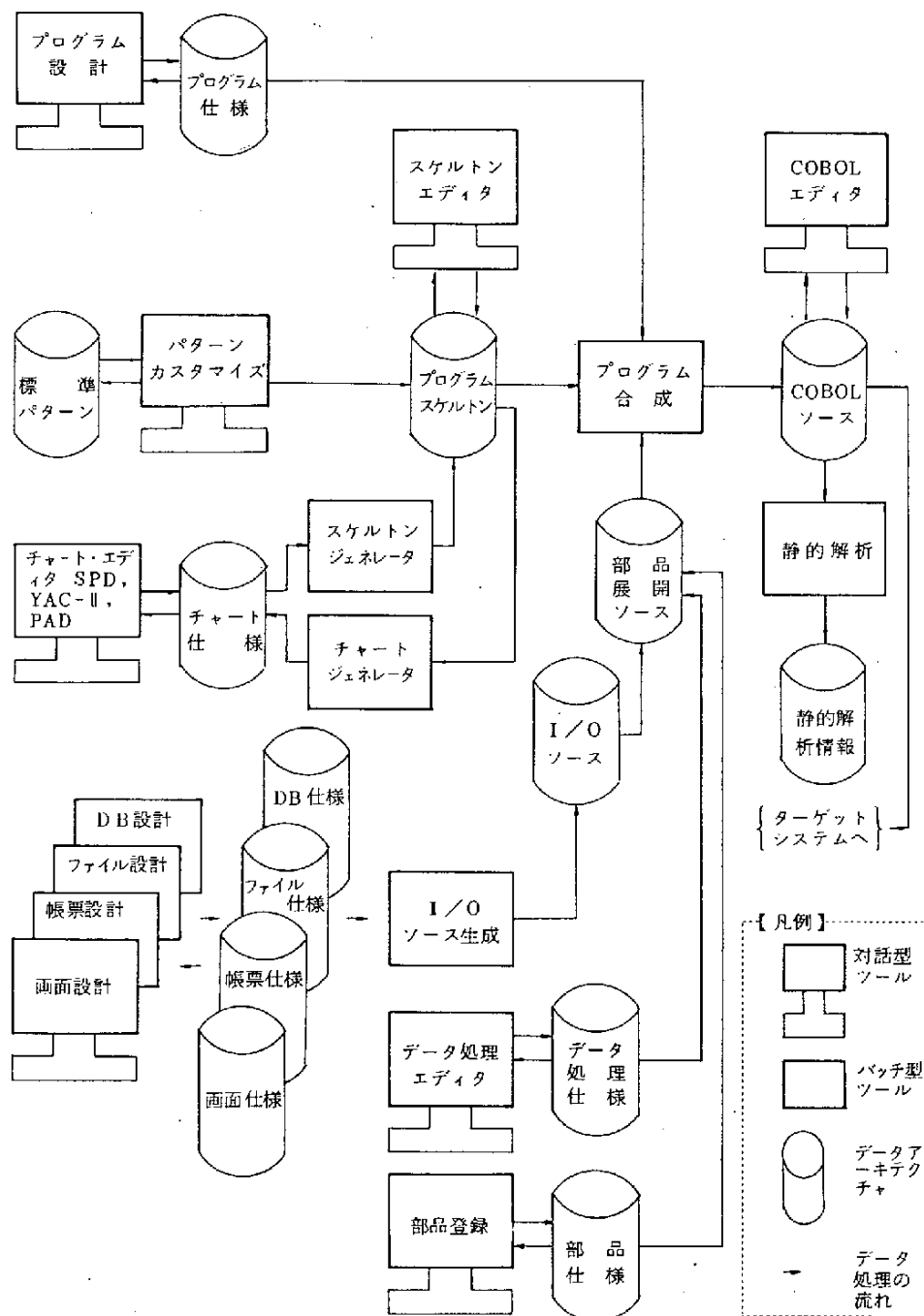


図 3.2 - 3 Σシステム（事務処理分野）のデータフロー

（Σプロジェクトの資料より）

3.2.3 システム開発支援ツールによる開発工程の変化

一貫開発支援ツールを導入してシステム開発を行うことにより、従来のシステム開発工程とは異なる点が発生する。

工程に変化が起こるということはシステム監査により想定されるリスクも変化する。開発支援ツール導入により発生しなくなるリスクや、逆に開発支援ツールを導入することにより新たに発生するリスクも存在する。

以下では、この点について検討する。

(1) 工程の統廃合

開発支援ツールによる開発工程で従来の開発工程と大きく異なる点は、逐次的詳細化と呼ばれる方法である。

例えばオンラインプログラムにおける画面設計を考えると、図3.2-4の「a. 従来工程」に示すように、まず画面のラフスケッチである概要設計を行い、ユーザの承認を得てから実際の画面設計（罫線の種類、色、プリンキング等）、さらにデータの項目設計を行って、最後に製造工程として画面記述言語により記述していた。

ところが一貫開発支援ツールは一つの画面設計ツールにより、概要設計としてユーザの承認を受ける画面の外部仕様書から最終成果物となる項目設計までをカバーすることになる（図3.2-4のb）。

なおかつ画面記述言語を使用せず、直接画面に会話形式で入力するため、画面上に定義されたものがそのままプログラム上の画面定義となる。つまり画面記述言語上のエラーは発生しない仕組みになっている。

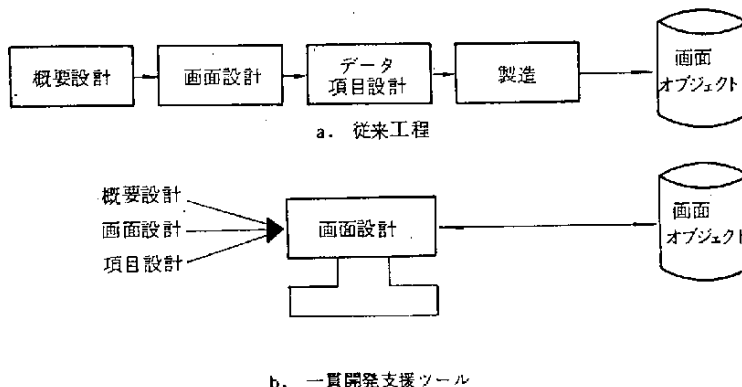


図3.2-4 工程の統廃合

(2) 部品化が進む

開発支援ツールにおけるプログラム合成の仕組みはソフトウェアの部品化と再利用にある。部品と呼ばれるものは単にプログラム上のサブルーチンだけでなく、以下のようなものが存在する。Σシステムではこれらをコンポーネントと呼んでいる。

a. パターン、標準パターン

プログラムの制御構造をメタ記述した骨組みであり、例えば検索処理の場合のパターンとかマスター更新のパターンとか幾つかの基本処理単位に作られる。

b. スケルトン

パターンは標準パターンと呼ばれるメーカから提供される物と、当該システムにおいて特別に用意されるものとがある。個々のプログラムはこのパターンを基にカスタマイズと呼ばれる処理により目的のプログラムの骨組みを作る。これがスケルトンと呼ばれる物である。

c. I/O部品 (I/Oソース)

画面設計、帳票設計、ファイル設計、DB設計はプログラム設計と独立して行われ、プログラム合成の段階においてスケルトン上に合成される。

d. 部品 (処理部品)

これが従来のサブルーチンに相当するものである。部品には、システム共通のものから特定のプログラム専用のもので幾つかの層がある。

このようにプログラムの部品化/流用化が進み、システム開発の作業形態/役割分担が変化する。これにより一般のプログラム開発におけるリスクは減少することが期待される。したがってパターンや共通部品の設計にシステムの信頼性が集中することになる。

(3) システム開発支援ツールによる新たなリスク

開発支援ツールは従来の開発方式と比較してシステム開発におけるリスクを大きく減少させることになる。しかし、開発支援ツールにより新たに考慮しなければならないリスクも発生する。

① 開発時の安全性

開発支援ツールはシステム開発者に対して便利な機能を提供し、開発の効率化が進むことになる。一方不正アクセスを行おうとする者にとっ

ても目的の情報を容易に見つけ出せることになる。そのため、開発時のアクセス管理が前にも増して重要となる。

② 分散開発

システム開発作業の多くがWSだけで出来ることになり、従来のように1か所に集中しなくとも開発が可能となる。つまり分散開発が実行されることになるが、各種統制が行きとどかないというリスクが発生する。

③ 不要なプログラムの組み込み

従来のシステム開発は多くの開発工程をへてプログラムが作成された。開発支援ツールは工程を統合化し効率を改善する結果、1つのプログラムや部品の開発に従事する担当者の数を少なくすることになる。そのため、従来より不要な機能／プログラムの組み込を発見する機会が減少する。

④ プログラムの粉飾

開発支援ツールを使えば、簡単にプログラムを合成することが出来る。合成のためにはツールにパラメータを指定することになる。パラメータの指定がシンタックス上正しければ、それなりのプログラムを合成出来ることになる。よって、見掛上プログラムが完成していても動作しないことが起こる。

3.2.4 システム開発支援ツールとシステム監査支援ツールの関係

一貫開発支援ツールはシステム開発における生産性向上を第一の目的として、図3.2-1に示すような機能・方法論に支えられて発展・遷移してきた。

一方システムの安全性・信頼性の面から見ると第2章で述べたように『開発コントロール機能』と『SA/QA機能』を持つシステム監査支援ツールが必要である。

この両者は同じシステム開発ライフサイクルで使用され、また同じ成果物に対して適用される。一貫開発支援ツールとシステム監査支援ツールはなんらかのインターフェイスが必要となる。

以下では、この点について検討する。

(1) 工程の時間軸と空間軸

両者の関係をシステム開発における開発ライフサイクルの時間軸、及び多数の中間成果物（各種仕様書、サブシステム、モジュール）とその作業（WSを使って分散開発される）に対応する空間軸に分けて考えることが

できる。

開発支援ツールの基本機能は主に一つの工程に注目しその入力情報と設計者による定義情報から出力を半自動生成する（図 3.2 - 5）。

これらの工程別開発支援ツールを一連の流れとして結合して一貫開発支援を行うのが開発支援ツールの役割である。一方、開発コントロール・S A/QA機能は工程を通して機能する必要がある、そのため時間的/空間的な情報の把握が必要となる（図 3.2 - 6）。

例として安全性を考えると、重要なデータベースにおける項目がどの工程のどこで誰が定義を行い（上工程，下工程），その項目を使用するプログラムは何か，プログラムの中で他の項目へコピーしていないか等々を確認することになる（横の工程）。

システム監査支援ツールは図 3.2 - 6 に示すような開発工程の時間軸/空間軸を通して情報の入手が必要となる。

(2) 資源管理

この時間軸と空間軸を通して開発工程あるいは中間成果物に関する情報を蓄積する機能が支援ツールのベース機能としての D/D (Data Dictionary) と総合ライブラリ機能である。

① 総合ライブラリ機能

開発支援ツールにおける総合ライブラリの必要性は主に変更時のバージョン管理機能にある。例として，ある部品のソースコードに変更があったとすると，その部品を使用しているプログラムは部品の修正後にコンパイルを行い，その後そのプログラムを引用している LM (ロードモジュール) をリンクする必要がある。

本機能はソースコードやオブジェクト，ロードモジュールなど異なっ

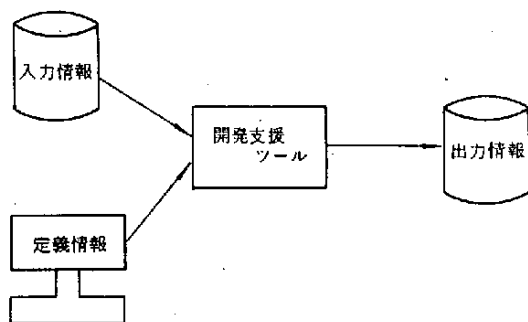


図 3.2 - 5 システム開発支援ツールの基本構成

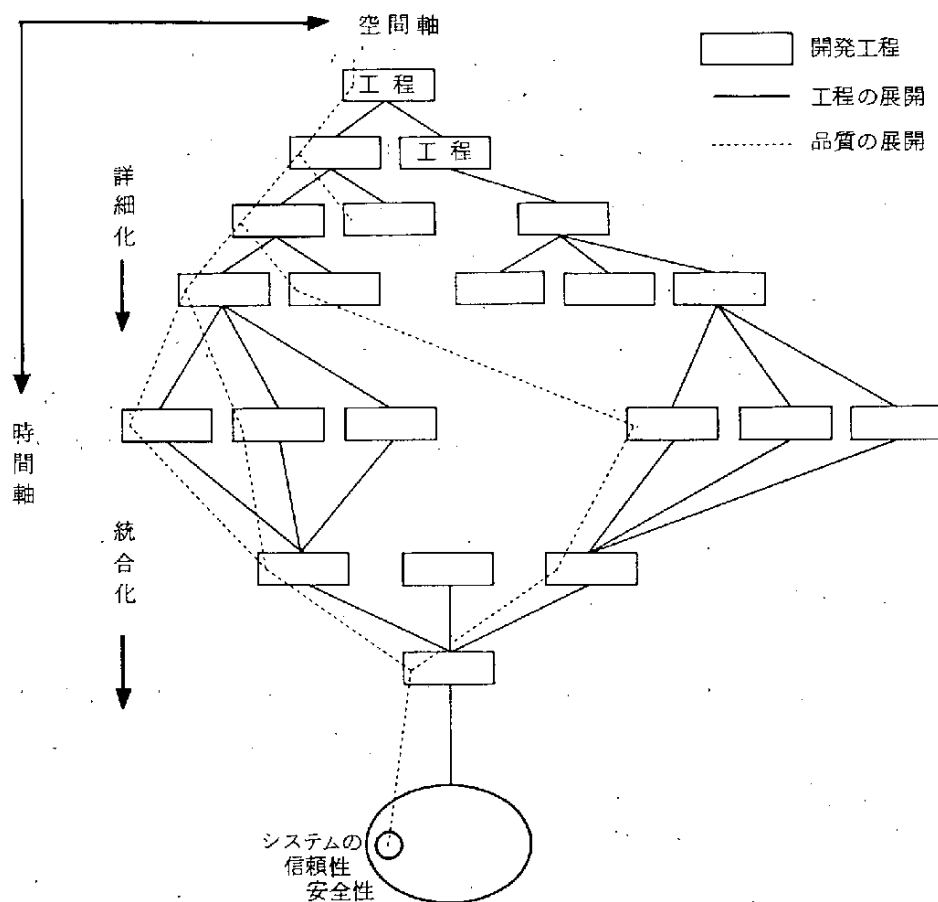


図 3.2 - 6 開発工程の時間軸／空間軸

た種類の間的成果物の世代管理／親子関係を含めて総合して管理するライブラリ機能である。

システム監査支援ツールにおいては中間成果物のアクセス統制，中間成果物に対する実証的な監査など重要な役割を持つ幾つかの品質管理項目／品質評価項目が計測対象となる。

② D/D (Data Dictionary) 機能

総合ライブラリは中間成果物および最終成果物そのものを総合して保持している。一方D/D機能は構成要素である部品，プログラム，画面などに関する定義情報とそれらの相互関連情報を保持する。

図 3.2 - 7 に，D/Dと総合ライブラリおよび開発支援ツールとシス

システム監査支援ツール（開発コントロール機能，SA/QA機能）の相互関係を示す。

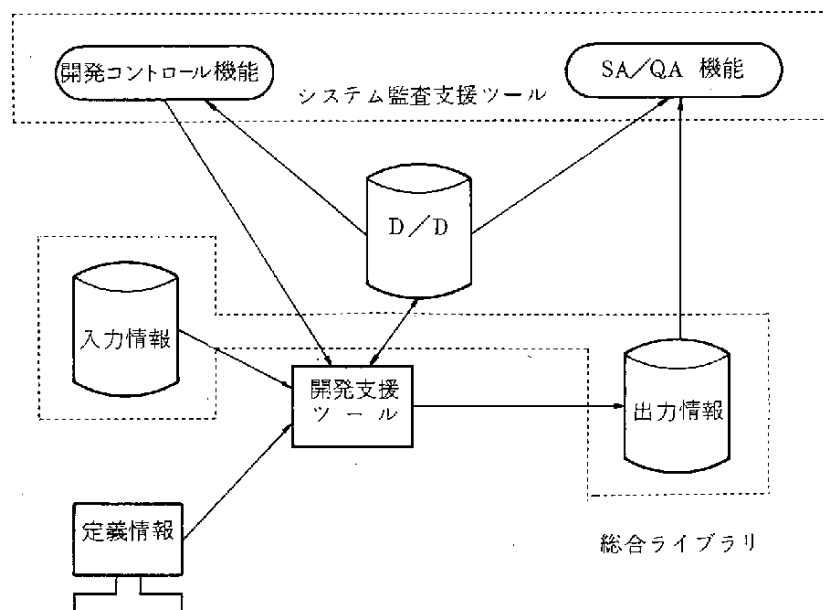


図 3.2 - 7 D/Dと総合ライブラリとの関係

(3) 定義情報管理

開発支援ツールにおいて各種定義情報を保有する機能がD/Dであり，システム監査支援ツールが開発工程を時間軸／空間軸の観点からとらえるために重要な機能である。

D/Dの機能はパッシブD/DとアクティブD/Dの2つに分類できる。現在開発支援ツールに実装されているD/Dの多くはパッシブD/Dである。

① パッシブD/D

パッシブD/Dは静的D/Dとも呼ばれ，各種定義情報を開発の結果として保存する機能である。アクティブD/Dとの差は定義情報を登録するとき，その定義情報が当該システムにおいて予め定義されたシンタックスに合致しているかどうかをチェックするかしないかの差にある。

登録される定義情報は複雑なネットワーク構造を持っており，D/D

はその定義情報の相互関連に関する情報を格納することが出来る。格納された定義情報は検索／編集機能によりシステム監査支援ツールの入力となる情報を提供することになる。

② アクティブD／D

アクティブD／Dは定義情報のシンタックスをチェックする機能を持っている。例えばプログラムやデータベース定義においてデータ項目に名前を付ける場合、プログラムにおけるデータ項目は単にプログラムの中における記憶領域の番地を意味する。

しかしシステム全体を通して考えた時、あるデータ項目の桁数が正しくプログラムに反映しているか、内部でたし算をするとき和のデータ項目はより大きな桁数が割り当てられているか、プログラム間での関係はどうなっているか、といったことがシステムの信頼性に大きく影響する。

この場合アクティブD／Dを用いることにより、論理的に同じデータ項目についてはプログラムが異なっても同じ項目名、同じ桁数を保証することができる。なぜならばプログラムやデータベースにおいてデータ項目を定義する時点で誤りを検出し、正しくなければ定義出来ない機能を持っているからである。

このようにアクティブD／Dはシステム開発における開発コントロールを直接開発行為に対して実施し、アクティブD／Dを使用したことを確認すれば、開発成果物の検証が行われたことになる。

アクティブD／Dの出現により本調査研究で提案するシステム監査支援ツールの開発コントロール機能が技術的に可能となる。

アクティブD／Dを用いた開発コントロール機能の実現には、次のような課題がある。

a. アクティブD／Dの範囲拡大

データ項目名、用語定義、情報資源管理の範囲を拡大する必要がある。

b. システム監査におけるリスク定義

システム監査の立場から分析したリスクをアクティブD／Dに定義する方法を検討する必要がある。

c. 開発支援ツールの意味解析機能

開発支援ツールが取り扱う定義情報の論理的な意味解析を行う問題

を解決する必要がある。

3.3 開発コントロール機能

3.3.1 開発コントロール機能の目的

情報システムの開発には多くの要員が参画し作業を分担する。貴重なリソースを使用し、多くの工数をかけて行われる作業が、健全な情報システムの構築という目標を効率的に達成するためには適切な開発コントロールが必要である。

情報システムのリスクにはさまざまな形態が存在するが、その中で開発段階のリスクとは、要求された条件下で、要求された情報システムが開発できないことである。それらのリスクを顕在化防止のための着眼点により分類すると次の2つになる。

① 開発作業／手続（含環境）に関するリスク

開発作業をコントロールすることによりカバーすべきリスクであり、リスク要因は作業形態に依存することが大きい。

② 開発対象に関するリスク

開発対象をコントロールすることによりカバーすべきリスクであり、リスク要因を作業形態から管理することが難しいため、対象すなわち開発の中間成果物を直接コントロールすべきものである。

このうち、①のリスク顕在化を防ぐ統制に着目し、開発コントロールとは、開発作業を通してリスクをカバーするものであるとする。現在のシステム開発がかなり高度にコンピュータシステムに支援された環境下で行われていることを考えると、開発支援ツール導入により新たに発生したリスクも①の作業リスクに含められることになる。開発コントロールが監査の受け皿としての被監査体制をも整備し、効果的な情報を提供できればシステム監査作業は効率的にそして正確になる。システムにより高度に支援された開発環境においては、開発支援ツールにあらかじめ開発コントロール機能を組み込むことにより、それまでは管理手続的に行っていた作業リスクのカバーを、システムの的に代替することができる。

情報システム部門内において従来の開発コントロールは、開発グループのリーダーが、指揮系統下の各サブグループのリーダーの協力を得て実施してきた。情報システムの持つ固有の環境を適切に反映した開発コントロールを

実施するため、その形態はさまざまであるが、開発が大規模になると、実施主体は開発コントロール自体の機能によりさらに細分化してくる。すなわち、要員計画を行う企画グループ、サブグループの割当や進捗管理を行うプロジェクト管理グループ、セキュリティ全般を担当するセキュリティグループ、開発成果物の標準化を行う標準化グループ等を各々設けている場合も見られる。開発コントロールが正しく機能しなければ、ことに大規模システムの開発は不可能なことが経験的に明らかである。それではいったいどのようなリスクが開発環境には潜在するのか、すなわち成功している開発作業ではどのようなリスクがカバーされているのかを次項で検討する。

3.3.2 開発作業のリスク

開発コントロールは作業に着目してリスクをカバーするが、そのリスクにはシステムが正しく開発されない場合と、開発環境を悪用する場合とがある。開発コントロール機能を考える際の前提として、それぞれのリスクの属性を述べる。

(1) システムが正しく開発されないリスク

リスク主体の行為に対する意識で2つに大別する。すなわち様々な要因により結果的に要求された条件が満足されない場合と、故意に不正な機能を開発する場合とに分類して述べる。

(a) 不正機能の開発の場合

開発作業の各工程毎に不正機能の開発はありうるが、その中ですでに現実的脅威になっているのは不正なプログラミングである。上工程の作業ほど下工程開発者によるチェックを受けるのでドキュメント段階で不正を記述することは考えにくい。また、プログラミング段階では、特定条件下のみ起動する不正なロジックを組み込めば詳細なコードレビュー、またはコード網羅テストをしない限り通常のテストデータを流しただけでは発見が難しい。よって、この段階がリスクの対象となる可能性が最も高い。故意ではないプログラムバグは、開発コントロールを検討する際、機能不備の範疇とし、ここでいう不正プログラミングには含めない。正しくないプログラミングという点で2者は共通しているが、リスクの影響度及び対処の方法が大きく異なるからである。不正プログラミングは故意である以上、合理的な実施者なら何らかの欲求に基づいた明確な

主目的を持つものと考えられる。不正プログラミング対策を検討する場合、この主目的からの分類は効果的である。なぜなら、ある不正プログラミング目的を特定すれば、コード化する際に使用する可能性の大きいフィールド名やI/Oコマンド、マクロ名、場合によってはロジックタイプ等のキーワードを絞り込むことができるからである。図3.3-1に示すように不正プログラミングの目的は、自己の利益を主な目的とする金銭詐取、情報詐取、権限詐取、及び対象情報システム側の損害を主な目的とするシステムへの打撃と信用への打撃に、コーディング上5分類することができる。

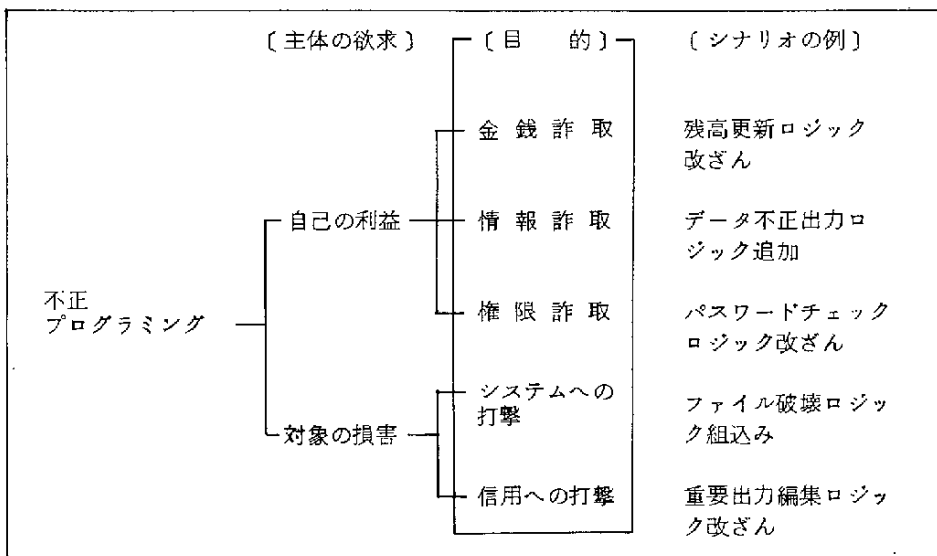


図 3.3 - 1 不正プログラミングの分類

それぞれのリスクには、顕在化と発覚について、いくつかのタイプがある。

- 金銭詐取には、発覚は承知の上でロジックを起動し、大きな額を詐取する場合と、小さな額を発覚しにくいような工夫を加えて詐取し続けていく場合がある。
- 情報詐取には、アクセスできる出力装置タイプにより詳細は変化するが、一般には発覚しにくいような継続的な詐取を可能とするルートを設定する。
- 権限詐取には、最終的には外部から破壊を行う場合もあるが、大方は

継続的にリソースの無権限使用を狙う。強い犯罪の自覚はともなわな
いことが多いが、他の不正行為につながる危険が大きい。

- ・ システムへの打撃は、ある条件下で起動するロジックを組み込んでお
き、それによって突発的な破壊を引き起こすことを狙う。「ロジック
爆弾」という呼び名は発覚してからでは遅いことをよく表している。
- ・ 信用への打撃は、一撃による破壊よりもある意味では悪質な方法で、
信用に関わる重要な処理ロジック中に発覚しにくい程度を考えて組み
込み、継続的に実施するものである。特に処理の信頼性に立脚してい
る企業では、重点的に対処すべきタイプである。

(b) 要求された条件が満足されない場合

これは、要求された条件がプロジェクト上のものか、情報システム機
能上のものかにより、開発の遅れや予算オーバーと開発機能不備に分け
る。

遅れや予算オーバーは、機能の見極めやステップ数見積りの甘さ、ス
ケジューリング及び進捗管理の不具合等さまざまな要因が考えられるが、
複数の要因がからみあっていることが多く、真の原因がどこにあったか
は明確に特定しにくい。この点は開発コントロールを考える際に着目す
べきリスク属性である。またリスクの顕在化状況は進捗モニターを実施
することにより、把握が可能である。

開発機能不備には、上工程と整合していない機能抜けと、その開発工
程自体が整合性を欠いている場合がある。工程毎のレビューまたは、工
程間におけるウォークスルーが十分適切に行われなかったことが要因と
して考えられる。どのレビューが不十分だったかという形で、原因は比
較的特定しやすい。これも故意ではないが、リスクの顕在化状況はテス
ト段階でなくとも最低限レビューステップ毎に把握できるはずである。

(2) 開発環境を悪用するリスク

これには、開発環境内の不正アクセスと運用環境への不正アクセスとが
ある。

(a) 開発環境内の不正アクセス

開発環境内の不正アクセスは、対象の損害としての開発の中間成果物
の改ざん・破壊と、自己の利益としての情報詐取に分けることができる。
改ざん・破壊のうち、利用段階での起動を狙った不正プログラミングは

ここでは範疇とせず、開発のトラブルを主目的としたリスクを指すものとする。開発段階では生産性重視のため、リソースへのアクセスコントロールは比較的ゆるめられている。またバックアップルールが固まっていなかったり、テストスケジュールの関係で、破壊された場合の影響が大きいデータは運用段階ほど簡単に特定できない。すなわち、破壊対象のデータを保護するのが困難である。誤って処理する可能性も合わせると、リスク発生頻度は運用段階より高い。一般に中間生成物は情報詐取の対象となりにくいですが、最新の技術を用いたコントロールプログラムロジックや汎用性の高いものは詐取の危険性がある。また、テスト用に本番環境から借りてきたデータにも情報詐取の対象となるものがある。開発環境ではリソースのアクセスに限らず、磁気媒体やプリントの持ち出し等も開発要員に多くまかされており、プログラムソースリストに紛れて、機密情報のプリントを持ち出しされる危険性もある。

(b) 運用環境への不正アクセス

開発環境はしばしば実際に稼働しているシステムと隣接している。運用環境への不正アクセスは、管理強度のレベルにより、対象がプログラムの場合とデータの場合に分けるのが有効である。プログラムは実際に稼働しているプログラムライブラリとは別に、保守用のライブラリを持つ等、ワンクッション置くことがある。通常業務として保守が端末から行われるため、チェックはシステムの十分強くすることが難しく、アクセスはデータに対するより、一般に容易である。よって、手続き的に補完する必要性が高い。

データは、直接画面から改ざんできなくても、改ざんジョブを起動できることがある。また開発環境には生産性向上のため強力な機能を備えている支援ツールがとり揃えてあり、それらを濫用すればシステムを操作することは容易である。

図 3.3 - 2 に、開発作業のリスクを 5 つに分類したものを示す。

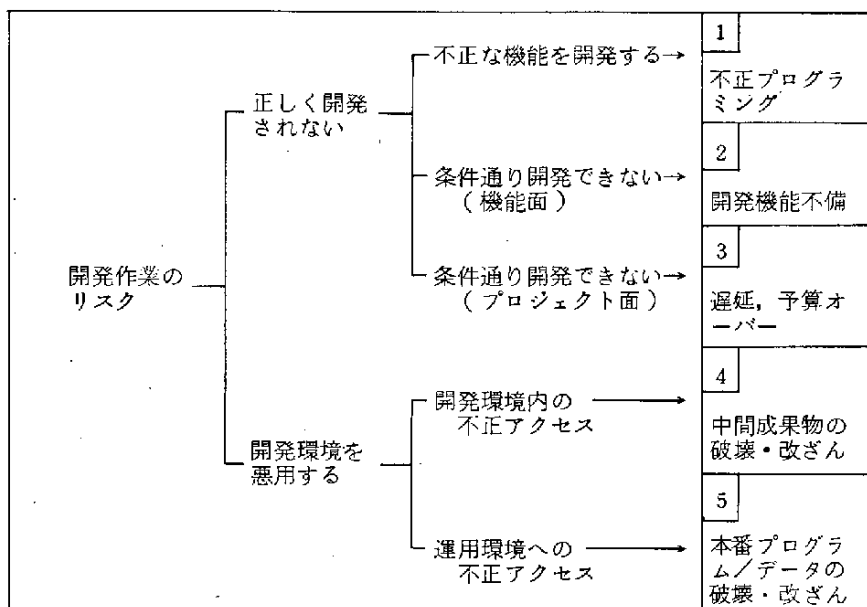


図 3.3 - 2 開発作業のリスクの分類

3.3.3 開発コントロールの検討

(1) コントロール機能の分類

リスクのコントロールは機能により次の4つに分類される。

- ・抑 止
- ・防 止
- ・検 出
- ・回 復

これらは、潜在的なリスクが顕在化し正常なコントロール状態に落ち着くまでの過程に沿って並べられたものである。

(a) 抑止コントロール

「抑止」とは、リスクを発覚させる行動をさせないような心理的対策である。すなわち、人間が故意に引き起こすリスクの場合、その人間に対しそのような悪意に基づく行動の達成可能性が低いことや、たとえ達成したとしてもすぐに発覚してしまうだろうことを知らせ、実行を思いとどまらせるコントロールである。モニターカメラや主体を特定しうるアクセスログは「抑止」機能の例である。

(b) 防止コントロール

「防止」とは、リスク行動が試みられたとしても顕在化させないような物理的、手続的等の対策である。すなわち、故意であろうとなかろうと、試みられた正当ではない行動を無効なものとし、行動の目的を達成させないコントロールである。パスワードや入退室チェックは「防止」機能の例である。

(c) 検出コントロール

「検出」とは、リスクの影響を最小限で食い止めるために発生状況をタイムリーに検出する対策である。すなわち、不正や事故が今発生しようとしているのか、また、すでに発生したか、まだ損害は極小化できるのか、あるいは大きな損害を招いてしまったかを把握するコントロールである。「検出」は通常「防止」と共に行われる。パスワードの誤入力、防止されると同時に失敗した不正の可能性のあるアクセスとして検出され、場合によりログに記録される。

(d) 回復コントロール

「回復」とは、発生したリスクによる影響から、正常なコントロール状態に回復させる対策である。回復後の状態がもはや適切ではなくなっている場合は「防止」・「検出」機能の再配置修正、すなわち「訂正」が必要である。抑止、防止、タイムリーな検出が効果的に実現できない場合、回復が重要になる。

以上の4機能は、対応するリスクを適切にカバーするために優先度を調整する必要がある。一般的には、回復機能が最も優先的に確保されなければならないとされている。これは、完璧な抑止、防止機能はありえないことを前提としている。開発作業の事故や不正による停止を起こさないために、開発コントロールとして回復機能は最低限備えている必要がある。以下に優先される機能は、防止、検出、抑止と続く。抑止は損害の防止を直接には保証しないという面が強調されると優先順位はこのように最も低くなる。直接的な防止が困難な場合は抑止に重点がおかれる。検出は、防止できなかったことを前提としているが、防止に期待出来ない場合は検出を優先する。一般的にコントロールにあてはまることであるが、開発コントロール機能はリスクの特質等考慮し、各機能をバランスよく構成し、最大の効果を上げられるよう配置するべきである。

(2) リスク対応のコントロール機能

開発支援ツールが開発コントロール機能として、どのような要件を充足すべきかを明確にする。そのために前述の5タイプのリスク毎にコントロール機能の検討を行う。

(a) 不正プログラミングのコントロール

不正プログラミング部分が検出されれば、回復は通常の保守と同様である。抑止には、モラルを高いレベルに維持する要員管理の他に、防止に起因するものと検出に起因するものがある。

① 防止に起因する抑止

モジュール標準化技法やモジュール動作環境の規定により、不正プログラミングにより実行できることを制限し、たとえ不正を行ってもあまり成果があがらないとして抑止するケースである。

② 検出に起因する抑止

コーディングの監督者によるレビューやセキュリティ担当者によるサンプリングレビュー、またはテスト等により検出の可能性を示唆し、不正プログラミング行為を抑止するケースである。

不正プログラミングの防止は、担当者以外による不正については主にライブラリーアクセスコントロールの問題になるが、担当者による場合は直接的でかつ実効的な防止方法はない。コードの人間によるレビュー以外の直接的な検出の有効な方法も非常に難しい。ここでは開発コントロール機能としては、以下のものを提示する。

① プログラマとテスト実施者の分掌を保证する機能

② D/D (Data Dictionary) による I/O 機能の管理

③ マクロ等の部品の管理

(b) 開発機能不備のコントロール

故意でない場合を考えているため、抑止は該当しない。不備ポイントが明確に特定されれば、回復機能は通常の保守作業とみなすことができる。ただし、システム開発作業の修正負荷は下工程へ行くほど急激に増加するため、タイムリーな検出が重要であり、さらに防止が可能ならそれにこしたことはない。一般にタスク間、ステップ間の作業におけるウォークスルーとレビューは一括して実施されている。それは開発文書を媒体としレビュー参加者の理解と、その確認によりスムーズに遂行される。

すなわち参加者が理解したという確証を含むレビュー履歴の管理とそれを添付した開発文書等中間成果物を、参照しやすい形で標準化し管理することが開発コントロール機能には要請される。

(c) 遅延・予算オーバーのコントロール

悪意で企てるわけではない場合を考えるので抑止は該当しない。遅延・予算オーバーは開発ペースアップ、または後続作業の予算の切りつめ等善後策を別にすれば不可逆的であり、回復も該当しない。

リスク要因は、前述のように明確な特定が難しい。すなわち見積りの精度やプロジェクト管理が適切であったか否かは、開発にどれだけ時間や費用がかかったかにより結果的に把握しうる事象である。よって、防止のために見積り精度を向上したり、スケジューリングを正しくする等は適切な検出機能の存在に依存している。そのため、進捗モニタ機能が必要不可欠であり、得られた予定と実績の乖離度を基にして、さらなる遅れや予算オーバーの可能性を増大させないためにプロジェクト管理に反映させる。

以上、開発コントロール機能として持つ必要があるのは、予定と実績を明確に把握しうる進捗モニタ機能とそれを開発作業にフィードバックさせるスケジューリングや差異分析サポート機能である。

(d) 中間生成物の破壊・改ざん

破壊を検出すること自体の意義は、破壊者を特定することによる抑止効果を除けば比較的小さい。改ざんを検出する意義も、不正プログラミングの場合と同様である。すなわち、このリスクの検出は他の機能より優先度が低い。開発環境では生産性が重視されるため、防止機能としては、プログラマが担当外プログラムへのアクセスを禁止するライブラリパスワードが期待される。また、稼働までに期間的余裕があるということを考えれば、バックアップによる回復が重要である。

以上、開発コントロール機能としては、破壊・改ざんされた場合に大きな影響を及ぼすと考えられるデータの適切なバックアップ機能及びアクセスコントロールが要請される。

(e) 運用プログラム／データの改ざん

開発段階の破壊・改ざんと比較して、バックアップの有効性が減ずることがある。バックアップファイルによる回復では、円滑な業務遂行の

時間的な制約に抵触する可能性があるからである。開発業務がまったく新規に行われる場合はともかく、多くの場合は、開発と並行して既存システムの運用が行われている。開発側から本番システムへの不正アクセスは、システムの的にも識別しやすいが、この両者は、システムの接続する必要性は一般に大きくない。よって、開発環境と運用環境を全く分離し、バス自体をなくすなら、それはセキュリティ上非常に望ましい。

開発コントロール機能としては、パスワードによる開発と運用メニュー画面の切り分けや、開発作業時のマシン環境の分離が要請される。

以上のまとめとして、表 3.3 - 1 にリスク対応の開発コントロール機能要件を示す。

表 3.3 - 1 開発作業リスクのコントロール機能要件

リスク \ コントロール	抑 止	防 止	検 出	回 復
1. 不正プログラミング	・要員管理 ・サンプリング ・コードレビュー	・アクセスコントロール ・標準化	・コードレビュー ・テスト	・プログラム修正
2. 開発機能不備	—	・ウォークスルー ・要員教育	・開発文書レビュー ・コードレビュー	・ドキュメント修正 ・プログラム修正
3. 遅延，予算オーバー	—	・スケジューリング ・差異分析	・進捗モニタ	—
4. 中間成果物の破壊・改ざん	・要員管理 ・アクセスロギング	・アクセスコントロール	・アクセスモニタ ・テスト	・バックアップ
5. 本番プログラム／データの破壊・改ざん	・要員管理 ・アクセスロギング	・アクセスコントロール ・環境分離 ・異常ジョブ強制終了	・異常ジョブモニタ ・アクセスモニタ	・バックアップ

3.3.4 システム開発支援ツールにおける開発コントロール機能の実現

開発コントロール機能要件を、これまで開発作業のリスクから検討してきた。

これらの開発コントロール機能要件をどの程度の水準で充足させるべきであるかは、企業の情報処理サービス機能の取締役会として存在するシステム

化委員会が、リスク顕在時のインパクトをどのようにとらえるかによる。

充足させるべき開発コントロール機能は、システムのまたは管理手続により補完され、実現されることになる。

以下では、体系的な開発コントロール機能としてシステム開発支援ツールが、あらかじめ備えるべき開発コントロール機能を示す。

(a) 進捗管理支援機能

プロジェクト編成環境の全領域において、十分な指導を要員に対して提供するために、プロジェクト管理者を支援する機能である。

各要員またはチームリーダーの入力により作業実績報告書を編成し、作業実績情報をあらかじめ登録されているスケジュールと比較し、差異分析のための資料を作成する。

これは、遅延、予算オーバーリスクのコントロールに直接的に対応する進捗モニタ機能だが、開発機能不備リスクのコントロールとしても機能する。プロジェクト管理者が工数の見積りやスケジューリングの負荷を軽減し、より効率的に正確に作業を行うことを支援する機能も望まれる。例としては、見積りガイドライン情報や、各要員のスキルレビュー情報等の参照を可能にし、スケジューリングをバーチャート等視覚的にシミュレートする機能があげられる。

(b) 開発文書管理機能

これは、単に文書を一元的に保管し、取扱いの便宜を図る機能を指すものではない。作業リスクの範囲と目的を適切に伝達し、承認、変更の管理を行うことにより、効果的なレビューを可能とし、各タスク担当者と、その調整に責任を持つシニアアナリストを支援する機能である。

各タスクの開発文書が標準化されるようにフォーマットや文書サンプルを提供し、作成・変更の履歴をとる。開発文書の重要度ランクに応じて、レビューやウォークスルーに出席すべきメンバー、チェック責任者、合せて検討すべき文書を指示したり、未決事項の見やすい形での表示もレビュー効果を向上させる。このような機能は、データディクショナリだけでなく、上流工程における設計情報のデザインディクショナリも必要とする。また誤操作による消去や破壊に備えたバックアップ、直接の変更者を識別するためのアクセスコントロールが必要である。

(c) プログラム作成管理機能

システム開発支援ツールのソースプログラム自動生成機能にセキュリティ上重要なロジックコードを管理させ、各々のアプリケーションプログラムの誤った部品使用を防止し、不正なコーディングを行いにくくする。すなわち、アプリケーションは、セキュリティ上重要な処理の実行時、必ず外部ルーチンを呼び出すものとしオーソライズされたコーディングは許可しない。それによりセキュリティ担当者は、各アプリケーションプログラム自体をコードレビューしなくとも、外部ルーチンの内容とその使用状況のレビューにより安全を保証できる。

また、構造化プログラミング等、作成者以外でもレビューが容易に行えるような標準化も、開発支援ツールにより推進されるよう、コーディングパターンの提供時、またコードの部品化の段階で考慮させるべきである。

プログラムライブラリに対するアクセスコントロール機能と、バックアップは必要である。また、コーディング環境やテスト実行環境と、本番の運用環境はパスを設定しないか、初期のメニュー段階で分離する等、システムの的に切り分ける必要がある。

3.3.5 作業ステップにおけるコントロール機能例

コントロール機能をより具体化するために、以下ではプログラム開発作業を例にとり、不正プログラミングリスクの防止と検出コントロールについて検討する。これらは、プログラム作成管理機能の一部として位置付けることができる。

プログラム開発作業をプログラマが端末の前に座り、ログオンしてからログオフするまでのステップに注目し、配列したモデルが図 3.3-3 である。この図のアクセスコントロールにおいて様々なレベルのセキュリティニーズに対応するためには、特にユーザの属性管理とユーザのモニタがセキュリティ強度の有効な調節手段となる。ユーザ属性管理では、ユーザの担当、所属等によりアクセス権限レベルをチェックし、それをもとにアクセス機能の利用制限やアクセス可能範囲の限定を行うことができる。

ユーザのモニタでは、モニタのタイミング（即時、日次、週次、etc）モニタの対象（特定のアクセス形態の場合、またはすべての場合）を変えることにより、不正行為の徴候を察知し、損害を極少化することができる。

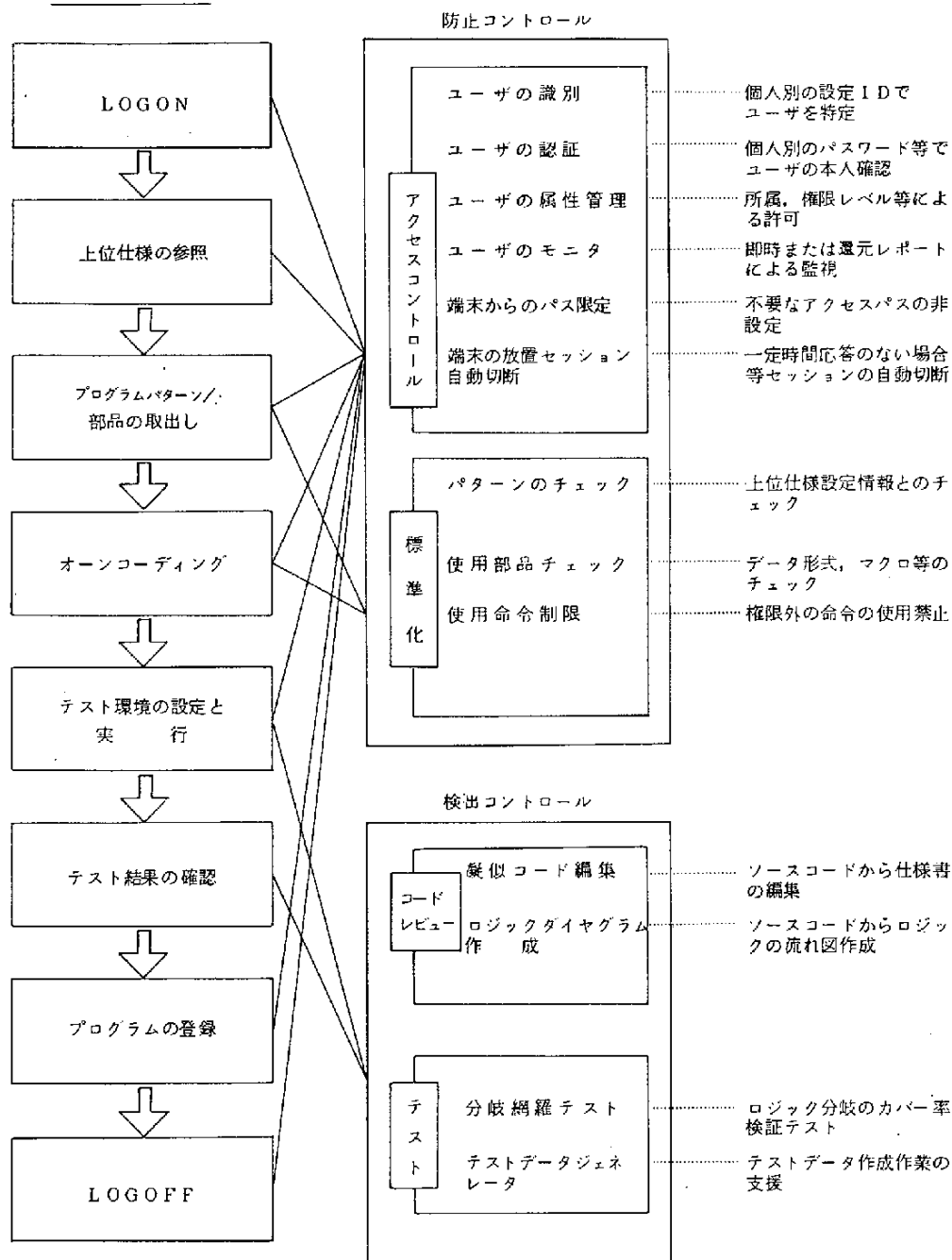


図 3.3-3 プログラム開発のコントロール機能例

標準化において、適切なパターンや部品の提供は、プログラム開発と保守の生産を向上するだけでなく、組み込める不正なプログラムロジックの機能も限定し、また、発見の可能性を増大させる。部品の内容自体と部品の使用状況をDD等で把握することになるが、部品は保守の過程でDDと整合がとりにくくなる場合があることに注意する。有効性を損なわないために、DDの整合性維持を考慮した保守ツールが必要である。

コードレベルにおいて、擬似コード編集、ロジックダイアグラム作成とも、プログラマ以外の要員によるレビューの正確性、効率性を向上する機能である。ソースコードからこれらを出力するためには、ソースコード自体が十分標準化されていることが前提となる。コードレビューの職務分掌、及びコーディングの標準化等、整備された開発環境において有効性を発揮する支援機能である。

テストにおいてロジックの分岐がすべて網羅されれば、プログラマ以外の要員のレビューにより、不正なプログラムロジックを検出し、排除することができる。分岐を網羅するテストは多くの工数を要するが、テストデータジェネレータ等のツールは作業効率を向上し、このようなテストを可能とするだろう。

3.4 SA/QA支援機能

前述したとおり、当調査研究ではシステム監査活動を支援する機能を開発コントロール機能とSA/QA支援機能に、また監査対象を開発作業/手続きに関するものと開発中のソフトウェア(成果物)に分類した。支援機能と監査対象の対応は、主に開発コントロール機能が開発作業/手続きに関するものを、SA/QA支援機能が開発中のソフトウェア(成果物)に関するものとした。ただし、監査対象の各工程で開発コントロール機能により監視されている開発作業/手続き全体に対する評価も各工程の終了時点において行うが、これもSA/QA支援機能による支援対象とする。例えば、当該工程において、不正アクセスが何回行われようとしたかという観点からの監査等がこれにあたる。

なお、一般的に上位工程の成果物は、次工程で作成する成果物の要件を全て網羅しているとはいえないため、作業者のノウハウ等で補完した内容が次工程の成果物の中に入り込んでくる。ここで入り込んでくる要素(作業者のノウハウ等)によっては、盛り込むべき機能が欠如したり、不要機能を含んだりする

可能性がある。したがって、場合によっては、監査人が作業者とは別に同等の作業を行い、自分の結果と作業者の結果を比較することにより機能欠如や不要機能の発見を行うものと当調査研究では条件を設定した。ただし、監査人の作業結果そのものが、機能欠如や不要機能を含む危険性はあるが、監査人と作業者の二人の作業により危険性を牽制しあえると考えている。

当節では、3.1であげたシステムの一貫性、完全性、安全性及び信頼性について監査を行う際に、SA/QA支援ツールとしてどのような支援ができるかという点について述べる。

3.4.1 一貫性・完全性に関する測定支援機能

当調査研究では、コンピュータ上に格納された成果物は工程をまたがりリレーションをとることができると仮定した。即ち、上位工程での機能ごとに通し番号がふられ、次工程で展開された機能は、上位工程の通し番号に枝番が付加されるとして、一貫して関連が付けられているとしている。

まず、一貫性についてはこのリレーション機能を使用して上位工程で定義された機能が下位工程で全て展開されているかを調べる。さらに、完全性といった観点から展開された機能の内容の検証を行う。この作業は前述したとおり、作業者の成果物と監査人の成果物を比較することにより行われる。

これを支援するために、SA/QA支援機能では、図3.4-1のように成果物の情報をもとにそれぞれの機能展開図を出力する。この機能展開図をみることにより、監査人は展開されていない上位工程の機能の有無を確認する。また、実際に展開された機能の内容については、図3.4-2に示すようにワークステーション上のマルチウィンドウ機能を使用して内容の比較を行う。この際、単にマルチウィンドウ機能で上位工程、当該工程の作業者、監査人の3つの成果物を表示するだけでなく、ウィンドウ連携表示機能として、上位工程の機能を選択することにより他の2つのウィンドウの関連する機能を表示するといった機能も有する。つまり、この機能により、比較する当該工程の2つの成果物の目次として、上位工程の成果物を使用するというイメージになる。

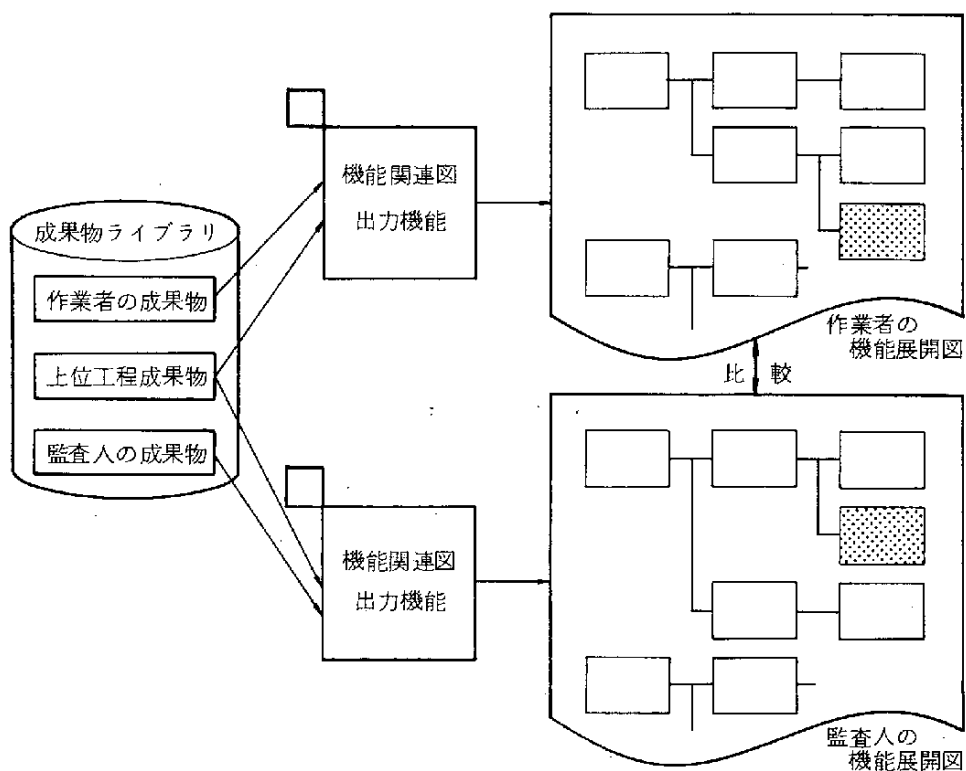


図 3.4 - 1 機能展開図出力機能

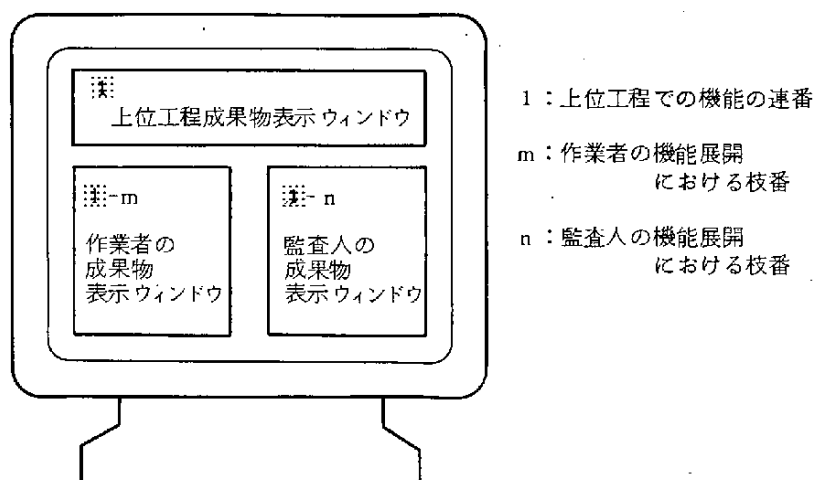


図 3.4 - 2 マルチウィンドウ機能による内容比較

以降、一貫性・完全性に関するいくつかの代用特性（IPAの品質管理工程表での管理項目のこと。次項3.4.2も同様）の測定支援機能のイメージを示す。

(1) 基本設計工程

(a) プログラム機能設計完全性（プログラム機能設計）

この場合の点検項目としては、以下の5つがあげられている。

- ① 必要な処理が全て入っているか。
- ② 無駄な処理が入っていないか。
- ③ あいまいな記述がないか。
- ④ 記述に矛盾がないか。
- ⑤ モジュール分割が適切であるか。

これら5つの点検項目のうち、⑤のモジュール分割については、効率化に関する点検項目であると当調査研究では定義し、対象外とする。

①と②については、作業者と監査人の成果物をマルチウィンドウ機能及びウィンドウ連携表示機能を使用して比較することにより、問題点（機能の欠如や不要機能の定義）を洗い出す。③と④については、作業者の成果物の内容を検証することにより問題点を洗い出す。

なお、これらの4点は基本設計工程だけでなく、詳細設計工程でも、また、①と②についてはコーディング・単体テスト工程についても同様なことが言える。したがって、これら2つの下流工程においても同様な作業になってくる。

(b) 入出力設計完全度（入出力設計）

この場合の点検項目としては、以下の4つをあげている。

- ① 入出力項目のデータ形式が明確か。
- ② データ間に矛盾がないか。
- ③ 機能仕様書の入力項目は全て入力データとして存在するか。
- ④ 機能仕様書の出力項目は全て出力データとして存在するか。

まず、①については入出力設計書にデータ項目の属性、桁数の他に初期値などの付属情報が定義されているか確認することにより検証できる。また、②については、処理で使用するレコード間でデータ項目の相互関係を調べることで検証できる。

実は、この①と②については、もう1つの管理項目である「データ項

目定義率」と関係している。というのは、上流工程においてデータ分析を行い、データ項目の定義がディクショナリ上に行われているとするならば、その情報がこの入出力設計でちゃんと流用されているかということを確認すればよいことになるからである。特に、②の問題はこの工程での問題ではなく、データ分析における問題である。したがって、①を重点的に検証すればよい。そのための支援として、SA/QA支援機能では入出力設計書のデータ定義に関する情報とディクショナリ上のデータ項目情報を比較し、そこで定義されていない項目を洗い出すことにより、監査人の作業負荷の軽減を図ることができる（図3.4-3参照）。

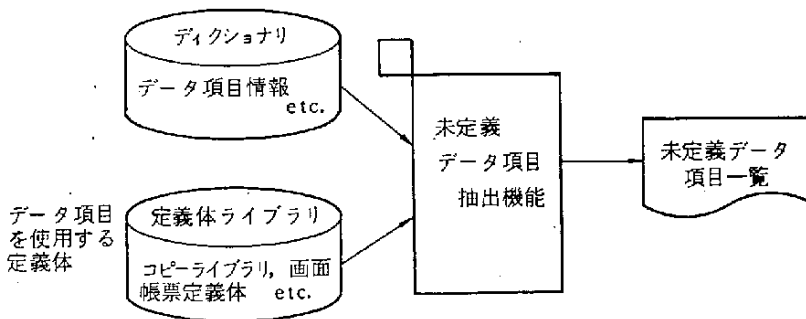


図 3.4 - 3 未定義データ項目抽出機能

また、③と④については、プログラム機能設計完全性（a）の場合と同様にマルチウィンドウ機能を使用して、機能仕様書と入出力設計書を表示し、その2つの成果物を比較することにより洩れを洗い出す。この際も、ウィンドウ連携表示機能により、機能仕様書上のデータ項目を指示し、入出力設計書の該当するデータ項目を表示することにより、検証作業の効率化を図ることができる。

（2）詳細設計工程

（a）モジュール間パラメータ完全度（モジュールインターフェース設計）

この点検項目については、以下の1つをあげている。

① モジュール間パラメータの双方の仕様が合っているか。

これに関しては、モジュール構造図、呼び出し元モジュール（親モジュール）、及び呼び出し先モジュール（子モジュール）の3つをマルチウィンドウ機能で表示することにより検証する。ウィンドウ連携表示機

能により、確認する親子のモジュールをモジュール構造図で選択すると、親モジュールのウィンドウで親モジュールが呼び出す子モジュールに関連するパラメータの部分が、子モジュールのウィンドウでパラメータの部分が表示される。この2つを比較することにより、パラメータの並び、属性及び桁数等を検証する。

3.4.2 安全性・信頼性に関する測定支援機能

安全性・信頼性については、処理定義の中でエラー処理やエラーが発生した場合のリカバリ処理についてどのように定義されているかを検証することにより行われる。

この場合も、IPAの安全性・信頼性に関する管理項目（代用特性）について、測定支援機能のイメージを示す。

(1) 基本設計工程

(a) 誤操作検出率（プログラム機能設計）

この点検項目については、以下の2つをあげている。

- ① できるだけ多くのエラーケースを考える。
- ② エラーリカバリにも気をつける。

これらに関しては、作業者と監査人の成果物を比較することにより、どのようなエラーケースが洗い出され、またそれに対するリカバリ処理がどのように定義されているかを検証する。2つの定義を比較することにより、エラーケースの洩れがないか、またそのエラーケースに対するリカバリ処理が正しいかどうかを検証できる。

(b) ミスデータ検出率（プログラム機能設計）

この点検項目については、以下の2つをあげている。

- ① 入出力設計で盛り込んだチェックを全て行う。
- ② 入力データのチェックは可能な限り行う。

これら2つの点検項目は同等なことを述べていると考えてよい。したがって、入出力設計と作業者のプログラム機能設計をマルチウィンドウで表示することにより、入出力設計で定義されたエラー処理がプログラム機能設計書の中で定義されているかを検証することができる。なお、プログラム機能設計書においてエラー処理がまとめて記述されていたり、処理の種別としてエラー処理という情報が設定されていれば、ウィンド

ウ連携表示機能により処理定義の検索の負荷が軽減されてくる。

(c) テストカバレッジ率(テスト仕様書作成)

この点検項目については、以下の1つをあげている。

① 全ステートメントを1回以上実行するテストケースを設定する。

詳細設計工程では、まだプログラムができていないため「全機能に関するテストケースが設定されていること」とみなすことができる。したがって、プログラム機能設計書とテスト仕様書を比較することにより、プログラム機能設計書で記述された機能に対応するテストケースがテスト仕様書に記述されているかということを確認すれば良い。したがって、この場合も他と同様にマルチウィンドウ機能とウィンドウ連携表示機能によりテストケース検索の作業の負荷軽減を図ることになる。

(2) コーディング・単体テスト工程

(a) モジュールテスト完了度(単体テスト)

この点検項目としては、以下の2つをあげている。

① テスト仕様書の全テストを完了したか。

② バグ摘出目標値との整合性を確認する。

①に関しては、付随点検項目として「③モジュールの全ステートメントを実行したか」という点検項目をあげることができる。

③に関しては、コンパイラの機能によりテストケースごとに実行ログを収集しておく。そして、その実行ログとプログラムをマッピングすることにより、未実行のステートメントがあるかどうかを確認する(図3.4-4参照)。また、この実行ログはテストケースに関する情報を保持

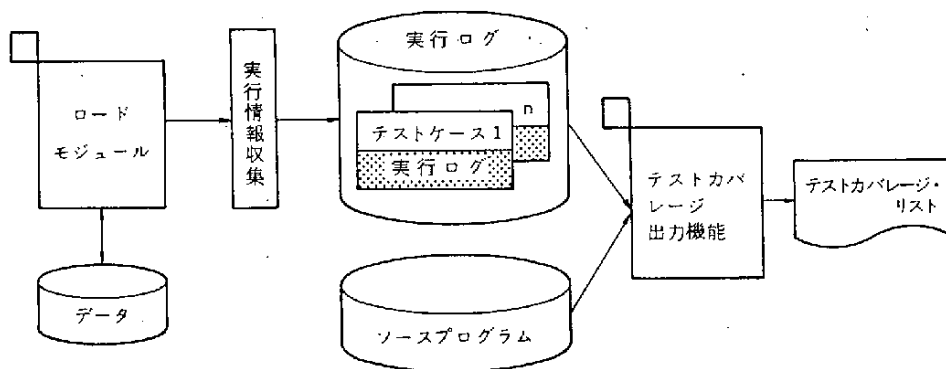


図 3.4-4 テストカバレッジ出力機能

しているため、実行ログのテストケース情報とテスト仕様書のテストケースをマッチングさせることにより、未実行のテストケースを洗い出すことができる（図 3.4－5 参照）。

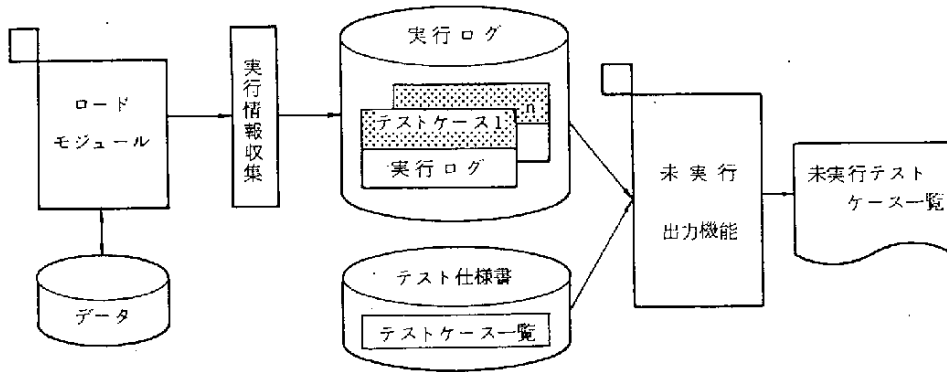


図 3.4－5 未実行テストケース出力機能

一方、②のバグ抽出に関しては、モジュールの修正作業の際に修正理由（抽出バグの個数とその対処方法）を記述させる仕組みを開発コントロール側に設けることにより、修正理由をもとに抽出バグの個数を把握することができる。なお、これ以外に、次の二つの方法でも可能であると考えられるが、以下に示す理由で上記の方法を採用することとした。

- i) システムによる異常終了情報からバグ件数を把握する。
- ii) 単に修正回数だけでバグ件数を把握する。

i) の場合、プログラム自身が異常終了しなくても、実行結果がおかしかった場合のバグの把握ができないという問題が残る。また、ii) の場合は、1回の修正で複数のバグを修正したりすると、同様に正確な件数が把握できない。したがって、修正理由からバグ件数を把握するという方法をとる。しかし、修正理由の場合も自己申告制なので申告つまり修正理由が正確でない場合は、他の場合と同様に正確なバグ件数把握ができないという問題を含んでいる。

3.4.3 その他の測定支援機能

今まで、一貫性、完全性、信頼性及び安全性の観点での測定支援機能を述べてきたが、ここでは、その他の測定支援機能として、以下の2点について記述する。

- 不正アクセスに関する測定
- 全定義体が作成完了かということに関する測定

(1) 不正アクセスに関する測定支援

不正アクセスそのものは、開発コントロール機能により監視されているため、事実上は不可能である。しかし、その開発作業の過程において、何回不正アクセスを行おうとしたかということは、監査項目としてあげておく必要がある。したがって、SA/QS支援機能では、開発コントロール機能により収集されているアクセスログ情報をもとに、その工程において不正アクセスの割合を算出することができる(図3.4-6参照)。これにより、不正アクセスの割合が高いプロジェクトにおいては、何らかのアラームを出すことができる。

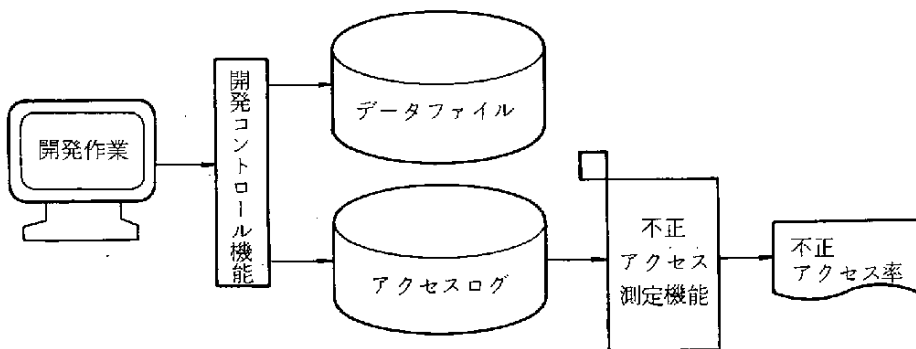


図 3.4 - 6 不正アクセス測定機能

(2) 全定義体が作成完了かということに関する測定支援

設計作業の段階においては、完全性の測定支援機能のところで述べた機能展開図によって、機能展開の洩れを洗い出すことができる。しかし、設計されたものが本当に定義体（プログラム、画面／帳票定義、コピーテキスト等）として作成されたかどうかは製造工程の終了時点で確認する必要

がある。したがって、定義体ライブラリと設計情報を比較することにより、全定義体が作成されているかどうかを把握することができる（図 3.4 - 7 参照）。

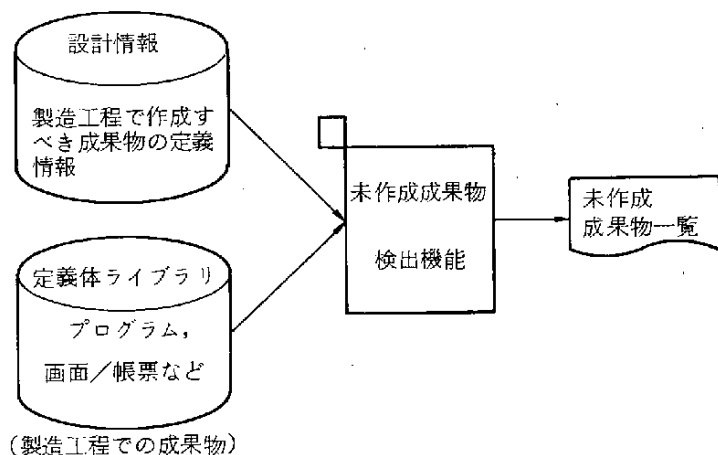


図 3.4 - 7 未作成成果物検出機能

3.4.4 評価支援機能

ここまでは、測定に関する支援機能をあげてきたが、ここでは測定した結果を評価する際にどのような支援ができるかということについて述べる。

基本的には、測定作業により収集された情報（測定結果）をライブラリの中に格納しておく。また、各作業に関する目標値をそれぞれ同様にそのライブラリの中に設定しておき、測定結果から数値を算出すると同時に、対応する目標値と比較することにより、監査結果としての評価を出力する（3.4 - 8 参照）。

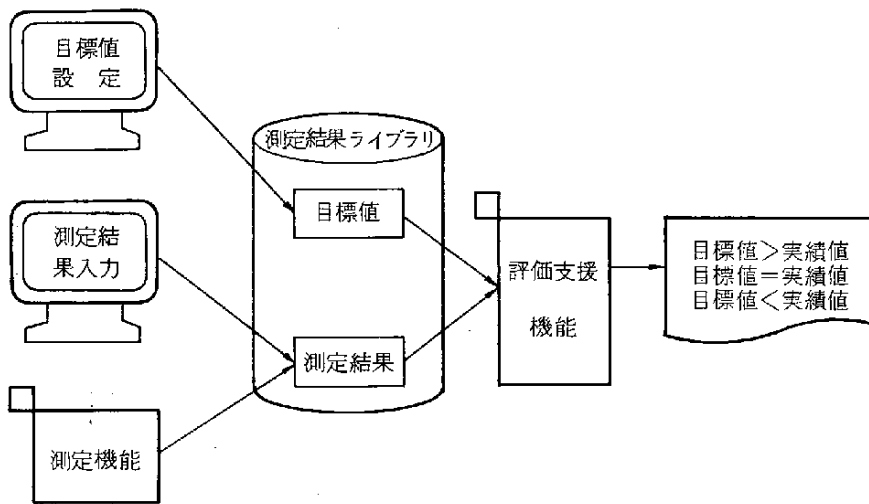


図 3.4 - 8 評価支援機能

どのような評価支援が行えるかを、前述した測定支援機能のうち以下の3つを例にあげて説明する。

(1) 未定義データ抽出機能

当機能は、入出力設計完全度の点検項目のうち、「入出力項目のデータ形式は明確か」という項目の測定を支援する。

当機能では、測定対象となる全定義体で使用しているデータ項目数やこれらのうちディクショナリ上に未定義のデータ項目数や、ディクショナリ情報と異なるデータ項目数が測定される。未定義データ項目数の割合や、ディクショナリ情報と異なるデータ項目数を目標値（例：前者は5%未満，後者は0個）と比較することになる。

(2) 未実行テストケース出力機能及びテストカバレッジ出力機能

これら2つの機能は、モジュールテスト完了度の3つの点検項目のうち、「テスト仕様書の全テストを完了したか」と「モジュールの全ステートメントを実行したか」という項目の測定を支援する。

それぞれ、未実行のテストケースの数とテストカバレッジ率と目標値（例：前者は0個，後者は100%）を比較することにより、モジュールテスト完了度の一部を評価する。

(3) 未作成成果物検出機能

当機能は、製造工程で作成すべき成果物が全て作成されているかの測定を支援する。

測定結果である未作成成果物の数が、目標値の0になっているかを評価する。0になっていない場合は当該工程はまだ終了していないと判断する。

3.4.5 監査人支援機能

前項までは、測定と評価の観点からSA/QA支援機能の例を上げた。当項では、監査人支援機能として、SA/QA活動全般に関する支援機能について述べる。

SA/QA活動において、全監査項目をチェックするに越したことはない。しかし、よりSA/QA活動を効率的に行うために、どの監査項目を重点的に監査したら良いかという指針を示すことが重要になってくる。そのために、過去の監査結果をロギングしておき、監査対象となるシステムの特性（例：使用するデータベースがネットワーク型データベースかリレーショナル型データベースの区別）をキーに重要度を付加したチェックリストを出力する（図3.4-9参照）。監査人はこのチェックリストの重要度をもとにSA/QA活動を行っていく。

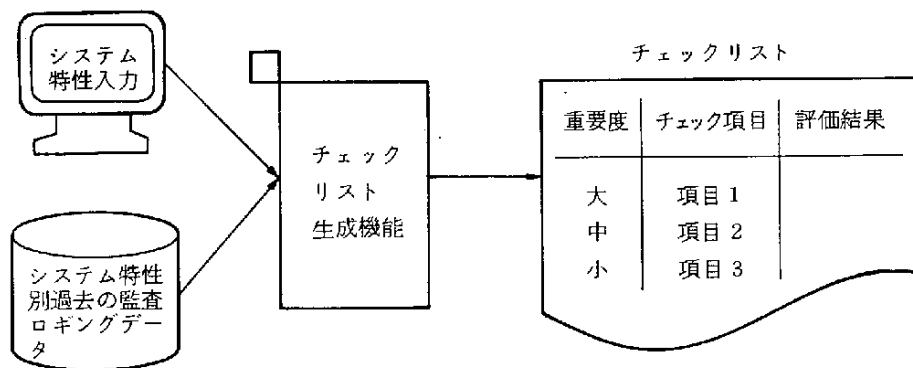


図 3.4-9 チェックリスト生成機能

更に、機械的に測定可能な監査項目については、「チェックリスト生成機能」から「測定支援機能」に対して起動をかけるルーチンを両機能の間におくことにより、「チェックリスト生成機能」、「測定支援機能」及び「評価支援機能」を連携させることが可能となる。これにより、システム監査作業の一部が、一連の流れで機械化されることになる（図 3.4-10 参照）。

今まで述べてきた SA/QA 支援機能のまとめを表 3.4-1 に示しておく。

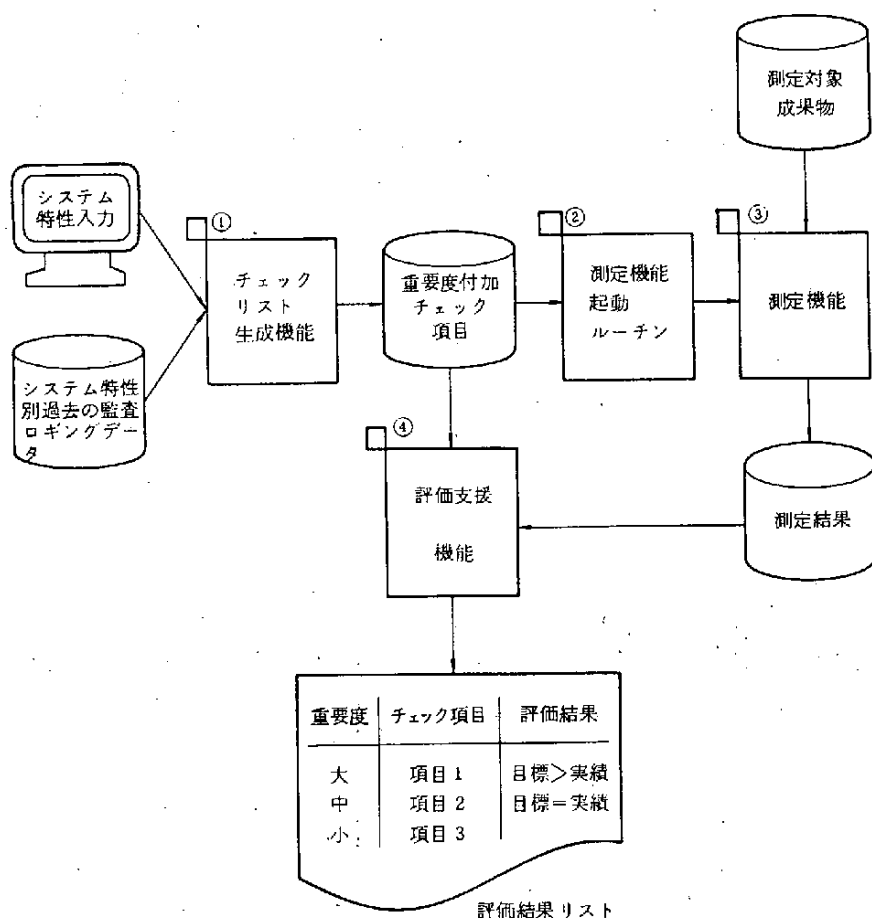


図 3.4-10 SA/QA 支援機能の連携

表 3.4 - 1 S A / Q A 支援機能の概要

支援分類	ポイント	支援機能	機能概要	前提条件
測定支援	一貫性・完全性	機能展開図出力機能	上位工程で定義された機能が当該工程でどのように機能展開されたかを出力する機能である。	成果物は、工程間をまたがって、リレーションがとれる。
		ウィンドウ速携表示機能	マルチウィンドウを使用時、1つのウィンドウで指示した項目に関連する内容を、自動的に他のウィンドウに表示する機能である。	成果物は、工程間をまたがって、リレーションがとれる。
		未定義データ項目抽出機能	ディクショナリ上のデータ項目情報とデータ項目を使用する定義体とを比較し、ディクショナリ上に未定義のデータ項目を抽出する機能である。	ディクショナリ上に事前にデータ項目情報が定義されている。
	安全性・信頼性	テストカバレッジ出力機能	実行ログファイル中の各テストケースの実行ログの累積と、ソースプログラムをもとに未実行ステートメントに関する情報を出力する機能である。	各テストケースで、実行情報がコンパイラにより収集されている。
		未実行テストケース出力機能	実行ログファイル中のテストケースとテスト仕様書のテストケース一覧を比較して、未実行のテストケースを出力する機能である。	各テストケースで、実行情報がコンパイラにより収集されている。
		摘出バグ測定機能	モジュールの修正記述をもとに、バグが何件発生したかを測定する機能である。	モジュール修正時に修正理由をもれなく明記する。
	その他	不正アクセス測定機能	アクセスログ情報の全アクセロ回数と不正アクセス回数をもとに、不正アクセス率を算出する機能である。	開発コントロール機能により、アクセスログが収集されている。
		未作成成果物検出機能	設計工程で定義された製造工程で作成すべき成果物の定義情報と、製造工程で作成された定義体ライブラリを比較して、未作成の成果物を検出する機能である。	特になし。
評価支援		評価支援機能	目標値と測定支援機能の測定結果をもとに目標値をクリアしているかどうかを出力する機能である。	測定支援機能の測定結果が自動的、又は手入力で測定結果ライブラリに格納されている。又、目標値も手入力で格納されている。
監査人支援		チェックリスト生成機能	与えられたシステム特性をキーに過去の監査ロギング・データから、監査項目に重要度を付加したチェックリストを生成する機能。このチェックリストをもとに監査人は、重要度の高い監査項目から監査していく。	過去の監査データがシステム特性をキーにロギングされていること。

4. システム監査支援ツール実現のアプローチ

本章では、これまで述べてきたコントロール、あるいはSA/QA支援機能を第2章で述べた様なフレームワークに従って、一つのツールとして実現することを考え、その実現のための方策を考えてみたい。

その基本方針の一つは「問題の単純化」である。つまり、闇雲にコントロールや監査支援を実現することを考えるのではなく、その前に、解くべき問題の単純化をはかりたい。

4.1 資源管理の充実

コントロールや監査の主要な関心事は、システムの完全性、安全性、信頼性、一貫性、あるいは効率性などの品質特性である。これらを維持あるいは検証することがコントロールや監査の目的である。

システムに関するこれらの特性の多くは、開発されるソフトウェアそのものよりも、そのソフトウェアが対象とする資源の特性でもある。開発されるソフトウェアを、いくら厳重にコントロールしたり検証しても、データディクショナリやプログラム部品などの開発資源、あるいはデータ、ネットワーク、プログラムなどのシステム資源が放置されていたのでは意味がない。

したがって、これらの資源に関する管理が先に充実されていなければならない。OSやDBMSなどの基本ソフトの資源管理機能を最大限に活用するとともに、それら資源について、可能な限り一元的な管理体制を作る必要がある。既に、第2章で示したフレームワークのベース機能群はそのためのものである。

特に、資源の特性情報を管理するデータディクショナリ（今後はシステムディクショナリと呼ぶ傾向にある）機能を中央に据えた、システム構成を実現することが重要となる。

4.2 制御の命題化と集約

システムの完全性などの特性は、分解すればデータなどの資源に関する完全性特性として捉えることができる。つまり、プロセスの完全性として捉えるよりも、その対象となる資源の完全性特性として捉えた方が効果的である。なぜならば、プロセスは業務や要求毎に分散する性格を持ち、その相互に不一致が発生しやすいからである。

したがって、データやネットワークなどの資源を一元的に管理するだけでなく、それらの資源について、その完全性、一貫性などを保証する条件を抽出し、定義することが重要となる。そうすれば、その条件を検証する制御機構を用意し、その資源を操作するアクティビティのすべてを監視、検証させることが可能となる。

また、プログラムとして実現されるプロセスについても、そのすべてをコードで表現するのではなく、プロセス命題記述と命題実行機構に置き換えることを考えるべきである。

資源に対する完全性制約条件、プロセスを表現した命題は、いずれも述語論理として記述可能なものである。もし、述語として記述できれば、その検証、修正はこれまでのソースプログラムとは比較にならないほど容易になる。

現在、AIツールと呼ばれているものの多くは、すでにこのような方式を実現している。

また、すでに、一部のDBMSでは、データベースの完全性制約をデータ定義に持たせ、プログラムでは、一切関知させないインテグリティ独立性機能を具備している。もし、多くの制御命題が述語やルールで表現され、資源の完全性制約も述語やルールで表現されれば、開発コントロール機能の集約と精度向上も期待できる。また、システム監査活動も、その焦点をデータディクショナリ等の資源情報に集中させることも可能となり、直接、ソースプログラムを対象とせずに実証的な監査を行うことも可能となる。

4.3 ソフトウェア構造の改善

これまでのコンピュータ化の歴史を振り返ると、当初、ハードウェアとプログラムしか存在しなかった。つまり、一つの業務プログラムで、入出力装置の制御から、同期の制御、あるいは業務処理の制御などが、業務処理そのものと共に実行されていた。

やがてOSが登場し、入出力装置の制御や同期制御などを肩代りしてくれ、それらは業務処理プログラムの課題ではなくなった。さらに、DBMSが登場するとデータ入出力操作の多くが業務処理から切り離された。

しかしながら、現在においても、依然として、業務プログラム本来の処理ではないものが、業務プログラムに包含されている。例えば、入出力トランザクションの妥当性・正当性検証処理、入出力画面制御、データベース妥当性検証

処理、あるいは複数ファイルを取り扱うときの一貫性制御等である。これらの処理は業務処理固有のものでなく、トランザクション制御システムあるいはDBMS、データ管理システムの課題である。これらの処理が、個々の業務処理プログラムの中で取り扱われているところに、誤り、不一致、あるいは不正の介在する原因があるといえる。本来、業務処理プログラムでなければ、実行できない処理だけを業務処理プログラムに持たせるアプローチが必要となる。

今後、OSやDBMSあるいはネットワーク管理システムは、さらに改善が加えられるものといえるが、その一つとして、業務処理プログラムとしてユーザが開発すべき部分を最小化するために、このような処理を置き換えて行くことが不可欠であろう。

4.4 開発方法論の変更

これまでのシステム監査あるいはコントロールは、当然ながら、これまでのシステム開発方法論で開発されたシステムを対象としている。しかし、これまでの方法論の多くは、コンピュータ化領域の拡大期に、個別のシステムを短期間に開発することに重点が置かれていたといってもよい。つまり、対象システムに組み込むコントロールや共有資源に対する配慮は、弱かったといえる。監査もコントロールも、闇雲に開発されたものを押し付けられてきたといってもよい。

確たる制御命題も定義せず、明確な標準もない環境下で、自己流で開発されたソフトウェアはどんなに優れたテスト技法を用いようとも検証は難しい。

したがって、システムあるいはソフトウェアの開発方法論も、コントロールや監査可能性を前提に編成されるべきである。また、好むと好まざるにかかわらず、今後のシステム開発は開発支援ツールを前提としたものとならざるを得ない。その開発支援ツールの多くは、本研究で扱ったような開発コントロールと監査機能を持つものとなろう。

そのような、環境の下でシステムを開発する手順や方法論には、次の様の要件が求められる。

- ・ アプリケーションプログラムとして開発すべき部分の最少化（部品化）
- ・ 資源の完全性制御に関する処理のプログラムからの分離
- ・ 共有資源設計開発・管理の専任化
- ・ 制御命題の分析・抽出と事前登録

- 資源定義情報（ディクショナリ）に基づくソフトウェア開発
- 内部統制としてのコントロール概念，原則の明確化

5. これからのシステム監査とその課題

5.1 内部統制について

(1) システム監査における内部統制の意義

システム監査は、立場・目的によって定義の仕方は異なる。

日本公認会計士協会は、EDPシステム監査を次のように定義している。

「EDPシステムによって作成された会計記録の信頼性の程度を確かめるため、システムに組み込まれている内部統制の信頼性を検討・評価し、EDPシステムによって作成された取引記録のデータ処理の有効性と妥当性を検討・評価すること。」

これは、システム監査の前提として内部統制の整備・運用を意味している。

一方、通産省が昭和60年1月に発表した「システム監査基準」では、その前文で次のように説明している。

「システム監査は、監査対象から独立した監査人が、情報システムを総合的に点検・評価し、関係者に助言・勧告するものであって、セキュリティ対策の実効性の担保およびシステムの有効性を図る上でも有効な手段。」

ここでは、内部統制については明言していないが、システム監査を単なるコンピュータ処理や、情報システム部の監査技術論として把握しているのではなく、有機的結合体としての情報システム全体の内部統制との関連においてとらえていると考えたい。

一般に内部統制は業務処理の完全性、正確性等及び経営方針への準拠性を担保とする役割を果たしているので、内部統制が整備され、適切に運営されない限り、監査は実施困難となる。したがって、システム監査においても、情報システムのもつ多岐にわたるリスクを把握・分析し、それを減少・防衛するための内部統制を整備することが、システム監査を実施するための基本的条件として必要となる。

(2) 情報システムの内部統制の構成要素

前項では、内部統制を明確に定義しないでこれを使用したのが、ここでその内容について触れてみる。一般に最もよく受け入れられている米国公認会計士協会(AICPA)の監査基準の定義を以下に引用する。

「内部統制は、企業の資産を保護し、その会計資料の正確性と信頼性を

チェックし、業務能率を増進し、かつ、定められた経営方針の遵守を促進するため、企業内部で採用する組織計画及びこれと調整統合されるすべての方法と手段から構成される。」

この定義は、資産の保全や会計記録の信頼性保持などの会計的領域ばかりでなく、業務の能率性増進を中心とした業務管理も内部統制の範囲としており、企業内におけるセルフコントロールシステムを包括するものである。内部統制の目的は、データ処理にコンピュータが導入されたことにより、データ処理の方法が手作業であろうと、情報システムであろうと変わりはない。しかし、その目的を達成するための組織、手続き及び記録は、データ処理方法によって影響を受けることになる。

情報システムの導入により、この内部統制にさまざまな変化が表れている。例えば、少人数で多くのデータを処理できるため、「両立しえない職務 (Segregation of Duties)」の問題や監査証拠の確保の難しさ並びに自動処理による承認作業の消滅などにより、従来の手作業の段階のとき以上に内部統制についての配慮が強く要請されることになる。

システム監査は、一般に運用段階と開発段階に区分されて実施されている。もっとも両者は全く別々のものではなく、システムの企画・開発から運用までのライフサイクルを通して、密接な関係を持っている。つまり、情報システムが、上流の段階である開発段階にて、いかに内部統制に力点を置いて開発されたかによって、その後の運用段階における情報システムの有効性が決まってくる。このことは、開発段階における内部統制の整備状況が、運用段階における内部統制に大きな影響を持つことを意味している。開発段階における内部統制は、以下の二つに分けることが出来る。

(a) 開発コントロール機能

「システムを正しく開発できないリスク」及び「開発環境を悪用するリスク」をカバーするための進捗管理、セキュリティ管理、要員計画・管理及び開発成果物の標準化の機能をいう。

(b) 情報システムに組み込まれるべき統制

通常の情報システムの運用における不正や誤謬の防止と、それらが発生し、情報(データ)の信頼性及び安全性が損なわれようとした場合にすみやかに発見・修正する手続をいう。

この様な統制に支えられているデータインテグリティの要件は、以下のよう

要約できる。

- ・取引の承認———すべての取引が所定の責任者によって承認されていること。
- ・取引の真実性———すべての取引は真実であり、虚偽の取引又は存在しない取引を排除すること。
- ・網羅性（完全性）——すべての取引が漏れなく記録され、処理されていること。
- ・正確性———取引の処理がすべて正確になされていること。
- ・一貫性———入力データが適時に入力され、期間帰属が保証されていること。

5.2 開発段階におけるシステム監査の現状

(1) システム開発ライフサイクル（SDLC）監査

通常の監査は、事後監査が主流であるため、システム監査もまた、運用段階のシステム監査に興味が注がれがちであり、事前監査である開発段階のシステム監査はそれに比べて一般的でない。しかしながら、開発コストの増大や、不完全なシステムの開発及び失敗により、企業は、システムの稼動後よりも、開発時にコントロールを組み入れることや、開発業務自体の標準化の促進に力を入れている。

開発段階のシステム監査は、一般的に①プロジェクト管理と②ソフトウェア開発管理に区分されて実施されている。プロジェクト管理は、プロジェクトの進捗管理、教育訓練などの労務管理、予算管理、プロジェクトの効率性・有効性の評価などのシステム開発全体をコントロールする業務である。一方、ソフトウェア開発管理は、システムがその生涯（System Life）にたどる過程（System Development Life Cycle：SDLC）、たとえば、要求の分析と定義、機能仕様設計、プログラム設計、プログラム開発及びプログラム保守ごとに、チェックポイントを定めてレビューする業務をいう。この過程ごとに、データインテグリティを保証するに足るコントロール機能がシステムに組み込まれているかどうかを検討することが、システム監査人の主要な任務である。また、システム監査人は、その企業が予め定めたSDLC過程の標準（システム開発方法論）への準拠性もレビューする必要がある。何故なら、前節で述べたように、システム監査に当たっては、内部統制

は基本的条件であるからである。したがって、システム開発方法論にも内部統制の存在が前提として考えられていなければならない。

(2) システム監査体制

システム監査人は、情報システム部から独立した部門に帰属し、全社的な立場で長期計画の立案、全社レベルのコンピュータ資源の有効利用及びSDLC監査などを実施するのが原則である。システム監査と似たものに、品質保証（Quality Assurance: QA）があるが、両者ともシステムの信頼性・安全性などの評価を目的とするものであることは同様であるが、品質保証は、対象とするシステムを直接に評価するのに対して、システム監査は、内部統制の整備運用状況の評価を通して、システム全体を評価するものである。この点において両者は異なる。

しかしながら、開発段階のシステム監査を情報システムから独立した部門に担当させるには、①システム開発における豊富な経験・知識、②日進月歩する技術の適時の習得、③対象アプリケーションのビジネス上の経験・知識が要求されるため、これらを情報システム部外に求めることは難しい。また、現実的対応として、システム開発に豊富でカレントな経験・知識を持つ情報システム部の独立した課、例えばQA課によって、開発段階のシステム監査を担当させることが効率的遂行の見地からも良いと思う。

一方、情報システム部と独立している本来のシステム監査人は、システム開発に係る基準・規則への準拠性の検討、信頼性・安全性の監査などの他に、QA課によってなされた開発段階のシステム監査の内容及び結果を、彼等の内部統制の一環としてとらえ、システム監査人の評価に役立たせることができるのである。SAとQAとは、相矛盾・重複して存在するものでなく、互いに補完して機能していると言える。

5.3 近未来のシステム監査

(1) システム監査諸規程の整備

現在すでに、システム開発における一貫開発支援ツールが実用段階に入っていると言われている。しかしながら、このツールが実質的に効果を発揮できるのには、システムを取り巻く環境及び資源についてのコントロール基準が明確にされなければならない。昭和61年1月に公示された「システム監査基準」、昭和61年10月より実施されているシステム監査技術者試験は、シ

システムの持つ光と影の部分を明らかにし、システム監査の実施によるシステムの健全性の確保が必要であることを社会に認識させた点で大きな成果を上げた。このことは、昨今の企業のシステム監査に対する姿勢を見ても確認できる。

システム監査の実施方法やその体制についての考え方は、組織によりさまざまである。そのような状況の中で、現在の「システム監査基準」は、システム監査人のある程度の行動指針を築いた点において、成果を上げている。しかしながら、現在企業内部で行われているシステム監査はその目的である信頼性、安全性及び効率性に合致しているかと言うと、改善すべき点は多いし、その実効性も上がっていない。またシステム監査人が依拠すべき、システムについての判断基準である基準或いは原則は、明らかにされていない。

企業会計原則の前文に「企業会計原則は、企業会計の実務の中に習慣として発達したもののなかから、一般に公正妥当と認められたところを要約したものであって、必ずしも法令によって強制されないでも、すべての企業がその会計を処理するに当たって従わなければならない基準である。」と定義している。

もとより、システム監査そのものが歴史的に浅く、実務として成熟していない領域であり、システム開発方法論もさまざまであり、システムが具備すべき要件も明らかでない。

システム開発の標準化としての一貫開発支援ツールを実用化させるには、そのシステム監査人が依拠できる環境及び資源についての統一的な考え方が、絶対不可欠の前提条件となる。

そこで環境及び資源についての原則又は基準を考えてみる。原則又は基準は、将来開発支援ツールが変化しても共通的、普遍的であるべきであり、内部統制の在り方も業務全般にわたって言えるものでなければならない。この考え方を示すと、図5-1のようになる。

したがって、企業の業種、業態に依存する内部統制の部分はシステム監査人の判断によるところが大きい。このことは、システム監査には、ますます対象監査領域の知識・経験が要求されると考えられる。

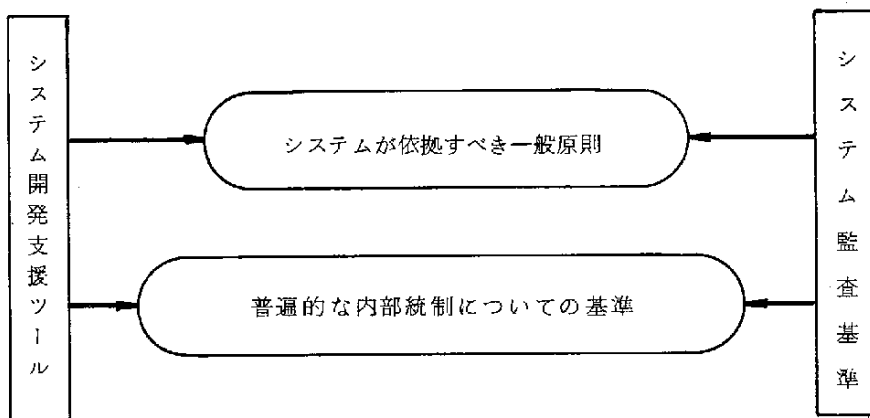


図5-1 環境・資源についての原則・基準とシステム監査の係わり

(2) 近未来のシステム監査のイメージ

システム開発段階におけるシステム監査は、コントロール機能の整備・運用状況及びシステムへの必要な統制の組み込みの有無を検討・評価するものである。しかし、一貫開発支援ツールに抑止、防止、検出及び回復のコントロール機能が加えられると、従来、システム監査人が行ってきたシステム開発工程ごとのレビューは比重が低下してくる。

このことは、従来のシステム監査人が受けもっていた任務が、一貫開発支援ツール（SA/QA支援機能）によってカバーされることを意味する。しかしながら、システム開発段階におけるシステム監査人の役割がなくなることではない。システム監査人は、予めDD（Data Dictionary）などに登録されたコントロール基準・環境基準の妥当性のレビューの他に、企業の業種・業態及びアプリケーションに特有な組み込むべき内部統制についての評価が必要になってくるし、さらにはSA/QA機能に組み込まれた解析・評価・予測機能により、システム開発の問題点を適時に且つ科学的に評価できることになる。これによりシステム監査人はタイムリーに問題点を指摘し、改善案を提案できることにより、システム監査の本来の機能を果たすことができるようになる。

また、一貫開発支援ツールの進展によって、システム監査はTechnology-orientedとBusiness-orientedの方向へと二極分化していくと考えられる。何故なら、システム開発の自動化が進展する程、このツールの技術的

信頼性への検討要求が強くなるはずであり、高度で且つカレントな技術がシステム監査人に必要になってくる。一方、アプリケーション特有のコントロール機能やシステムに組み込むべき内部統制（この点の自動化は難しい）の内容のレビューに、力が注がれることになる。このBusiness-orientedの方向のシステム監査人に要求されるのは、コンピュータ関連知識よりも当該アプリケーションの知識・経験であり、各業種ごとの専門（例えば、銀行、証券、信託）のシステム監査人の出現が予測される。

5.4 不正プログラム機能の予防／発見の考察

システムに必要な機能以外の機能を何者かが故意に押入し、データへの不正アクセス、システムの誤動作・破壊を行うことがある。

以下では、この種のリスクの内、システム監査においても発見の難しいとされる「トロイの木馬」に関する予防／発見の可能性について、システム監査支援ツールが分担する機能の検討を行う。検討に当たっては、現状の技術で実現可能な機能であることを考慮した。

(1) 品質管理項目

必要な機能以外の機能が含まれない、即ち不正な機能がないこととは、品質尺度（設計品質）における「追跡可能性の上向き要求機能」に相当する。

この品質尺度は、要求された機能が開発工程の上流工程から下流工程へと展開して行く時、その機能について工程間の関連を追跡出来ることを表す尺度である。

要求仕様から下工程へと追跡して行くのを「追跡可能性の下向き要求機能」、逆にソフトウェアにおける最終成果物に近いルーチン／モジュールと言ったプログラム構成要素から、逆に上工程へと溯行して行くのが「追跡可能性の上向き要求機能」である。

図5-2の例で示すと、要求→仕様1→設計1→製造1→A、Bへと機能展開を追跡して行くのが「追跡可能性の下向き要求機能」であり、逆にA、B、C→製造1→設計1→仕様1→要求へと溯行するのが「追跡可能性の上向き要求機能」である。

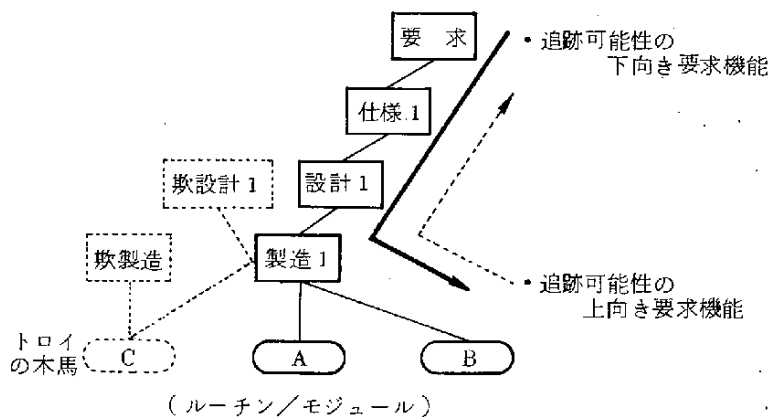


図5-2 トロイの木馬の組み込み

(2) 品質評価項目

トロイの木馬が組み込まれていないことを評価するのは、「追跡可能性の上向き要求機能」を計測し、評価・判定することになる。

重点対象としては、重要なデータベース、ファイルの参照／更新、処理の制御を司るモジュールなどが上げられる。

(3) 開発コントロール／品質管理

トロイの木馬が他の品質項目と大きく異なる点は、作業の誤り／漏れではなく、故意に行われる点である。よって他の項目のように作業者本人に注意を促すことを目的とした点検項目より、第三者レビュー、設計承認に重点を置く必要がある。

(4) 品質保証

品質評価尺度により計測・評価を行うのであるが、開発コントロールと同様に作業の誤り／漏れではなく、故意によって行われるための配慮が必要となる。

即ち、他の評価項目は、予め評価項目や評価基準を決めて公開することにより、予防／抑止の効果が期待できるのに対して、トロイの木馬の場合は、逆に評価項目を公開すると、監査で発見することが難しくなる。それはトロ

イの木馬を組み込もうと意図する者が、評価項目／評価基準をも粉飾することになるからである。

図5-2の例で、「C」のルーチンがトロイの木馬として組み込まれたとする。品質評価項目の「追跡可能性の上向き要求機能」を計測すると、「C」に対応する製造1に規定されていなければ「C」はトロイの木馬として発見できる。しかし偽りの製造仕様や、さらに上位の設計仕様でトロイの木馬を組み込み、正しい製造仕様の中に紛れ込ませた場合、発見は難しくなる。

また、着眼点を作業／成果物から作業者の作業準拠性へと移すことも効果的である。

(5) システム監査支援ツール

システム監査支援ツールでは、品質管理項目／品質評価項目である「追跡可能性の上向き要求機能」をサポートする必要がある。また(3)、(4)で指摘した項目についても、ツール化による支援が必要である。

これらの要件をまとめてみると、以下のような仕様が考えられる。

① 開発履歴

「追跡可能性の上向き要求機能」を計測するためには、開発の中間工程における作業履歴をどこかに保存しておく必要がある。

② 工程における相互関係

工程における作業と成果物、その上位仕様や関連仕様が明確に対応付けられていること。

③ 人との相互関係

作業担当者、承認者、レビュー者と仕様書／成果物との相互関連が記録されていること。

④ 開発中の不正アクセスログ

開発期間中、開発コントロール機能が持つ抑止・検出に抵触したアクセスの記録を採る機能が有ること。

⑤ 変更・修正履歴

最初から不正機能を作り込む場合より、改造・修正時に組み込まれるケースが考えられる。これに対応出来るように①から④までの要件項目が変更・修正時にも機能すること。

これらの機能要件は、本調査研究で提案するディクショナリ（設計・開発情報データベース）を中心としたシステム監査支援ツールで実現出来るもの

と考えられる。

(6) システム一貫開発支援ツール

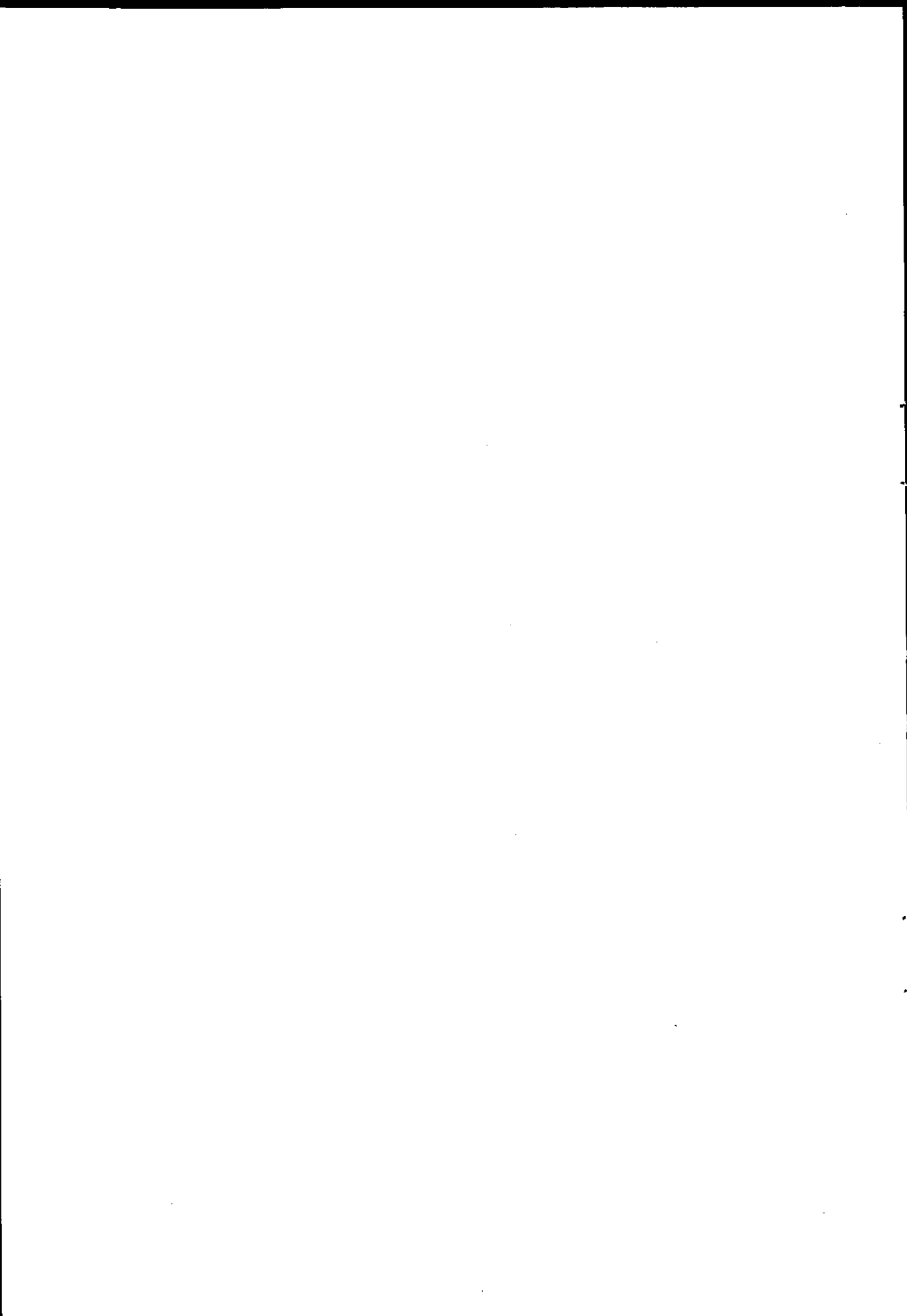
一貫開発支援ツールの持つ部品化や制御構造のパターン化などの半自動／自動プログラム生成の機能が、トロイの木馬防止に有効である。

これらの機能は、上工程で定義したものをプログラミング工程を省いて製造することになる。そのため下工程で定義されない機能が紛れ込むことを防止出来る。

また部品化を押し進めることによって、特定データに対するアクセスを特定の部品のみに限定することができ、データへの不正アクセスを、部品の使用制限で実現出来る。

1. The first of these is the fact that the
2. *Journal of the American Medical Association* has
3. been the most influential of the medical
4. journals in the United States for many years.
5. It has been the most widely read and
6. the most influential of the medical
7. journals in the United States for many years.
8. It has been the most widely read and
9. the most influential of the medical
10. journals in the United States for many years.

第Ⅱ編 システム監査へのエキスパートシステム適用



1. 米国におけるシステム監査へのエキスパートシステム適用 実態調査の概要

本章は、米国におけるシステム監査へのエキスパートシステム(ES:Expert System)適用の実態調査の概要を取りまとめたものである。詳細な内容については、「付録」を参照されたい。

1.1 調査の目的

本調査は、システム監査におけるES導入動向を把握し、今後、システム監査のどのような領域に、どのような形態で導入される可能性が高いかについて検討することを目的としている。

1.2 調査の結論

システム監査ES導入の可能性が高い作業領域は、本調査によると次の領域である。

- (1) プロポーザル作成
- (2) システムの概要把握
- (3) 監査支援ツールの使用
- (4) 調査結果の分析
- (5) 問題点の評価

ただし、固有の監査環境に起因する検討部分も可能な限り客観視しうよう、前提条件の明記に努めた。

1.3 調査結果

1.3.1 調査のアプローチ

システム監査ESの導入可能性を次の3つの視点で検討した。

- ① システム監査の側からの検討
- ② ESの側からの検討
- ③ 過去の事例分析への立脚

①に対応して、システム監査作業やその知識の分析、及びシステム監査ESの使用環境分析を行った。大きく捉えれば、そこからESの導入対象となるシステム監査作業の性格を把握し、導入対象領域選定の拠り所となる“有

1.3.2 環境分析調査の概要

環境分析調査では、以下の4つの項目について調査した(図1-2)。

- ① システム監査の作業及びその知識
- ② システム監査ESの使用環境
- ③ ES技術水準
- ④ システム監査ESの構築環境

また同時に現実的な裏付けとして、ES導入事例についても調査した(表1-1)。

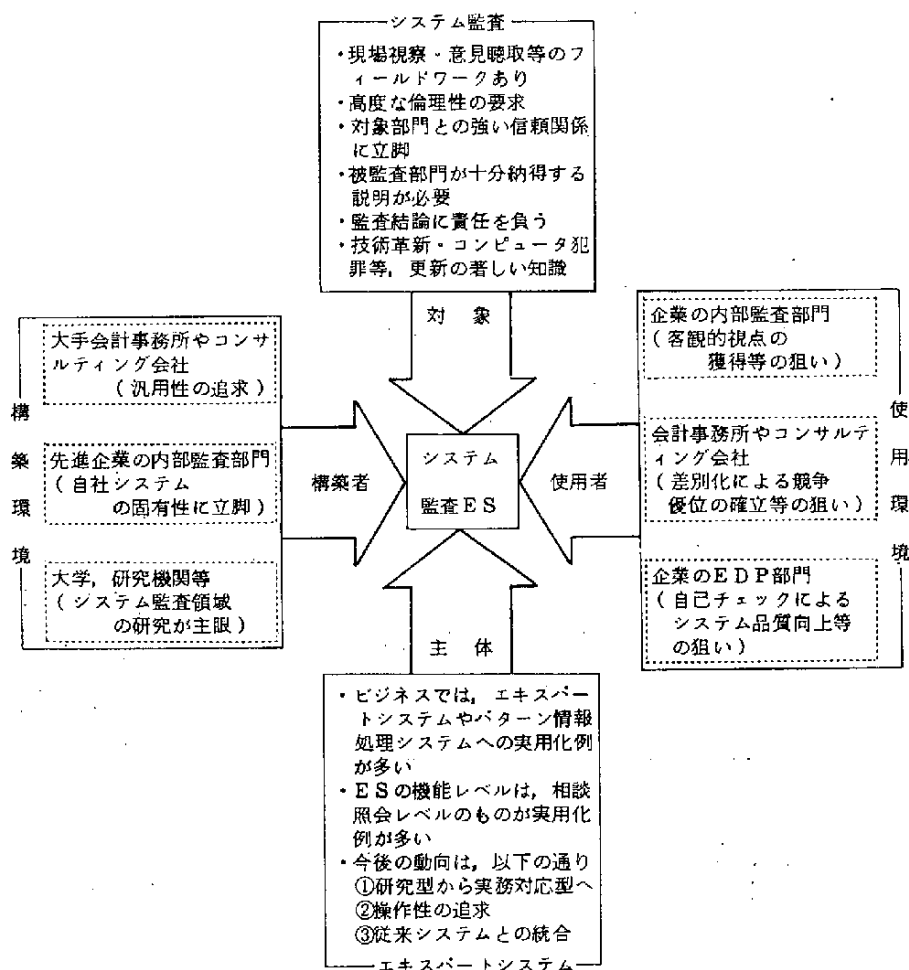


図1-2 4つの方向からの環境分析

表 1-1 調査対象の EDP 監査領域のエキスパートシステム

項番	システム名	内 容	成果と課題	ソフトウェア・ シエラ	開 発 者
①	EDP-XPERT	高性能コンピュータシステムの統制の信頼性を評価	オンラインとデータベースのコントロールに関するルールを含む知識ベースの拡大を予定	AL/X	M. Messier V. Hansen
②	EXPERT AUDITOR	データセンター, Disaster リカバリー, メインフレーム全般, パソコン全般等の 7 領域のシステム監査を支援	質問表を提示し, 回答から診断, または詳細調査指示を行う	AL/X	Boden Associates
③	AUDITPLANNER	監査計画局面での重要性レベル判定の支援	正答率は約 70%	EMYCIN	Steinbart
④	ARROWSMITH-P	ソフトウェア開発管理の支援	特定環境においてのみ試行	————	Maryland Univ.
⑤	I.C.A.	歳入サイクルの内部統制を評価する	監査調書データを使用した推論による評価が妥当	EMYCIN	Gal
⑥	IST/RAMP	EDP リスクの定量, 定性分析支援	導入実績は, 米国, カナダを中心に約 90 件	フォートランとアセンブラで開発	Inter- national Security Technology
⑦	BDSS	統計的手法を用い, リスクを多変量的に分析	マネージメントサイドの視覚的なレポート作成	————	Ozier, Perry & Associates および Pickard, Lowe & Garrick, Inc.

1.3.3 導入検討調査の概要

システム監査への E S の導入について, 以下の 2 つの観点から検討した。

- ① 有効性
- ② 困難性

また事例裏付けとして, 処理ロジック信頼性監査への E S 導入検討例につ

いても調査した。

一般には機能が増せば構築も難しくなる。すなわち有効性と困難性にはある程度の相関がある。しかし、問題領域の特性はプラス要素（有効性）とマイナス要素（困難性）それぞれに同等のインパクトをもたらさない場合がある。さまざまな要因に揉まれつつも、そのような歪みにこれまでの事例は生じたとみなすことができる。それぞれのインパクトを考慮しコストパフォーマンスの高い領域にES導入を狙うべきである。

また同時に、問題領域選定にあたって有効性・困難性の観点は、当面のES構築段階のみならず、使用段階までも考慮にいたるものでなければならない。特に従来のシステムではあまり問題にならなかった独特のものとして、構築段階のプラス要素と使用段階のマイナス要素は見逃せない。

ここで参考になるのは、プロトタイプ開発をとりあえず行えばよいという考え方である。それは、ES開発支援ツールや言語の選択段階であれこれ決めかねている場合に有効となることがあるが、無批判にとびつくのは危険である。しばしば見られるのはプロトタイプ開発までは進んだが、“このあとどう作り込んでいくのか”と困ってしまい埃をかぶるケースである。構築の後の取り扱いについて明らかにされてないと、構築されたESも実務環境からどんどん乖離し、用いられなくなる危険性がある。すなわち、導入の基準、普及のための研修、ユーザの声をどのように反映するか、重要なノウハウをいったいどのように社内に保存するか等は、あらかじめ検討される必要がある。

また、本調査の検討目的が、システム監査作業においてES導入可能性が高い作業領域の選定ということを含んでいる以上、各作業領域固有の有効性及び困難性を取り上げることが必要である。

以上、2つの条件を留意した上で検討した有効性及び困難性は、図1-3の通りである。

段階	属性 有効性・困難性	システム監査の各 作業領域に固有	システム監査の各 作業領域に共通
構築	有効性	———	知識の資産化（知識の体系化と定着）
	困難性	知識獲得の困難性 常識； 曖昧性を伴い量的に膨大な常識を知識ベースに格納するのは困難 ヒューリスティックス； 経験的知識は専門家自身に意識されていないことが多い	- 専門家の獲得（長期的な協力が困難） - KEの不足 - 知識表現上の制約 - コスト（膨大な投資と長い開発期間）
使用	有効性	作業代行 絶対効果； 導入によりどれだけ負荷が軽減するか 相対効果； マニュアルやDSS等の導入と比較してどれだけ効果が大きいのか	- 高品質の共有（統一的で高度な監査の共有） - 専門家の育成（システム監査人の効率的な育成） - 差別化（差別化による競争力強化）
	困難性		- 運用負荷（知識ベース改訂の困難さ） - 出力評価（出力結果の正当性評価が困難、また誤判断に伴う責任所在が問題）

これらの有効性と困難性から問題解決すべき対象領域選定の観点として、システム監査の各作業領域固有のものを選定

(1)有効性の観点

①絶対効果

②相対効果

(2)困難性の観点

①常識

②ヒューリスティックス

図1-3 ES導入の有効性と困難性

1.3.4 システム監査エキスパートシステム導入可能性検討の概要

システム監査エキスパートシステム導入可能性検討は、以下の3つについて

て行った。

① E・S導入が有望視されるシステム監査作業領域

② それぞれの機能イメージ

③ 導入に関する懸念

また、事例裏付けとして、米国大手会計事務所の動向についても調査した。

上記①、②については表1-2に要約して示す。

表1-2 エキスパートシステムの導入が有望視される作業領域

作業領域	機 能	具体的な機能イメージ			効 果
		インプット	アウトプット	限 界	
プロポーザル作成	プロポーザル作成の総合支援	①監査人によるマネージメントニーズの解釈結果 ②監査人による監査リスクの解釈結果等	①監査チームメンバーのスケジューリング ②予算編成 ③予想しうるクリティカルポイントの指摘 等	①マネージメントニーズや監査リスクの解釈等は、監査人に依存	①業務効率の向上 等
システムの概要把握	アプリケーションの機能・構成等の把握と着眼点設定の支援	①アプリケーションシステムの機能・構成・組織等に関する基礎調査結果 等	①システム概要書 ②セキュリティ対策とリソース使用に関する大まかな着眼点 等	①アウトプットのレベルは、予備調査結果レベル	①被監査システムをあまり知らない、外部の監査チームにとって有効 等
監査支援ツールの使用	使用技法（ツール）及び使用対象の決定支援	①監査着眼点 ②システムの機能概要 等	①監査支援ツールの選定 ②ツール適用準備書（対象領域・実施方法・計画等） 等	①各ツール固有のノウハウを蓄積していく必要がある。	①最適な監査技法・ツールの選択 等
調査結果の分析	ツール使用やドキュメント・レビュー、インタビュー結果の分析支援	①ツール使用結果 ②インタビュー結果 ③ドキュメントレビュー結果 等	①問題点の所在の追跡 ②インタビュー等の発言の論理的妥当性チェック 等	①各々の監査計画に依存する特定の視点はなし ②問題点の定量化・比較はしない。	①膨大な調査結果を効率的に分析 等
問題点の評価	分析・解釈を経た各問題点の定量化・比較支援	①解釈結果（意味付けされた分析結果） ②評価の軸 等	①各問題点の定量化と比較 ②問題点の全体的把握（優先付け） 等	①あらかじめ準備された評価軸しか利用出来ないため、評価レベルは限定される。	①監査人の評価の客観化に貢献 等

以下③について説明する。

本調査ではコスト・パフォーマンスを念頭に置き、困難性を極小化でき、かつ有効性を極大化しうるような問題領域を選定した。しかし、そのことによって、困難性が完全に克服されてしまうわけではない。どの問題領域を選定しようとも、そこに付きまとう共通のリスクをはじめとして、数々の障害を乗り越えていかねばならない。

以下、その代表的な障害を挙げて、その対処方法を示す

(a) 社内の認知

ESをビジネスへ導入する場合には、社内の認知が得にくいとの意見をよく聴く。システム監査を業とする会計事務所やコンサルタント会社の場合はあまり該当しないが、構築者が内部監査部門の場合等においては、クリティカルな問題である。

それは、知識の整理や体系化が成果としてなかなか定量化しにくいためである。目に見える成果を考慮するなら、これまで検討した有効性を踏まえた上で、次の点を考慮すべきである。

- ・システム監査人の教育投資
- ・部分的省力化による作業範囲拡大

また、ESに事務処理レベルの業務を実行させるのには決断がいるだろうが、次のような機械化による基本的な有用性もあげることができる。

- ・人間が行うより見落としが少ない
- ・作業内容の質が平準化される

(b) KE(ナレッジエンジニア)の存在

我が国よりES開発において数年進んでいる米国でも、KEは質量ともに不足していると言われる。KEの助力が特に必要なのは、専門家としての意義が成果物にあり、その過程の説明があまり要求されない専門分野である。KEは、インタビューだけでは知識獲得が困難な場合、ある程度専門家となるために自ら弟子入りすることもある。

ところが、システム監査ESの構築者としては、システム監査人の場合が多いと考えられる。かつ、システム監査人は結論に至る自己の判断過程を、マネジメントや被監査部門に、納得のいくように表現し、報告することを経営ルーチンワークとしている。

KE不足が訴えられている現状を考えると、システム監査をまったく知

らないKEにあえて依存するより、システム監査人がKEの役割の知識抽出を代替する方が望ましいと考える。KEに専門知識を学ばせるより、専門家にKEの役割をさせた方が、プロトタイプ開発が早かったという米国の事例は詳細な前提なしには論拠にならないが、ある問題領域にとっては専門のKEが必要不可欠ではないこと、そしてKEがそれほど大きな成果をあげられない場合があることを示している。

また、ESの実績が少ない現状では、LISPやPROLOG等のESプログラミングスキルと知識獲得・知識表現化過程の相異、及びES構築経験とKE的能力との相異を明確に認識する必要がある。

以上、システム監査ES構築において、我が国におけるKE不足が現在のところクリティカルな障害にならない可能性を指摘した。

(c) ES構築シェルやツール

大学や研究機関では、問題解決すべき対象領域を選定し、いくつかのES構築シェルやツールを並べ、実際にプロトタイプを開発しての比較検討がなされてきた。

しかし、ビジネス環境においてそのような検討はあまりなされなくなった。これには、比較する環境が得にくいことや、比較による選定の効果を評価することが難しいこともあるが、ビジネスへの導入成否は厳密なツール選択には依存しないと認識されるようになったためである。システム監査ESにこれがあてはまるか否かは、ほぼ、構築者がビジネスとどう関わっているかによるだろう。すなわち、大学等のように研究として続けて行くか、または実務で使うことを念頭におくかである。

システム監査ESがシステム監査人と同等のアウトプットを出せない以上、どのES構築シェルやツールにも、何らかの長所・短所がある。様々な試行から出たES構築シェルへの要求は、領域シェル(DOMAIN-SHELL)への要件として将来的に統合されていくだろう。

2. システム監査へのエキスパートシステム適用の期待・可能性

昨今、産業界の全分野はもちろん医療・教育・公務などの広い分野にわたって、AI（人工知能）が研究され始め、今やブームとさえいえる状態になっている。AIの応用、特にESの開発事例は、実験的開発の段階をも含めここ十数年来、故障診断、資源探査、医学的診断、財務投資コンサルティング、建築法規システム、財務諸表監査などの分野にわたり多くの発表が行われ、最近においては成功例も出始め、さらにその適用範囲や分野の拡大が図られている。

ESは、各方面の専門家の知識を「知識ベース」に蓄積しておいた上で、専門家の判断プロセスを参考にしながら、「推論エンジン」機能の活用により、各種の問題解決を行うシステムとすることができる。ESの代表的な適用分野としては、診断システム、設計システム、制御システム、教育システムなどがある。システム監査との関連で見れば、対象システムをモデル化し、正確に理解するための設計型システム、対象システムの問題点発見のための診断型システム、対象システムの内部統制上の問題点やセキュリティ上の弱点をコントロールするための制御型システム、システム監査人を教育・訓練するための教育型システムなどへの適用が考えられよう。

以下は、本委員会でのシステム監査に対するESの適用や期待についての検討結果を取りまとめたものである。

2.1 システム監査へのエキスパートシステム適用の基本的な考慮事項

ESをシステム監査へ適用する場合に考慮しておくべき事項の検討結果については、以下のように取りまとめることができる。

- (1) システム監査の中でも適用業務の監査は最も重要な分野であり、かつ多くのケースを占める。ここでは適用業務自体が多様であって、監査のポイントも夫々に異なる。そこで、システム監査にESを適用する場合に考えられる第一のステップは、反復して行う監査を自動的にコントロールする部分への適用である。売掛金業務の監査では、一つの営業所について実施した監査手続きと手法を、他の営業所については自動的に実施していくのである。また、前期決算時の売掛金ファイルに対して実施したことは、今期ファイルについては自動的に処理してしまうのである。これだけでは大したエキスパティーズがあるとは言えないが、現実に監査人を支援する効力は無視できない。シス

テム監査人は、このような業務に意外に大きな割合で時間を割いているのが現実だからである。もう一つ有利なことは、前回と異なった事象が検証されても、優秀な監査人は、この監査内容に最も多くの専門的知識と経験を持っているので、その後の推論過程をシステム化出来る可能性が高いことである。データ及び処理の正確性、信頼性監査は殆どこれで解決できると言っても過言ではない。

- (2) E S構築においては、対象となる専門分野の固有の知識の明確化が特に重要であり、さらに、そのシステム構築の機能や構造の解析、類似システムの調査、現状での問題点や解決のための手段の分析と定義が明確化される必要があると言える。

この観点から、E Sがシステム監査のどのような局面に適用できるか、何の利点があるかについてはシステム監査のプロセスから類推していくことが可能であろう。特にこの場合、システム監査に類似した開発済E Sを見出し、それを参照して見ることがE S構築における基礎的方向を見出すのに有益となると思う。そこで、システム監査の行為やそのプロセスと類似点があるので、既にE Sにより開発がなされ、実用に供されているものは何かを探て見ると、そこには先ず「医学診断と治療」に関するシステムがある。その各プロセスの類似点を表2-1に示す。

- (3) システム監査は、数学のように唯一の解答が定まる作業ではない。真白から真黒までの間の何処かに幅を持って解答が存在し得る世界での作業である。これは、知識だけでは処理できないので、それなりの能力を要するであろう。したがって、システム監査がE Sの機能に期待を持つとすれば、下記のような作業についてであろう。

①複数の専門家の何人分かの知識を知識ベース化して、生産性を上げると共に、システム監査人の知識の不足を補う。

②システム監査における判断ルールを推論機構に還元して、システム監査人の判断能力を支援する。

また、監査実施の局面は、分析型意思決定問題、つまり前以って準備された仮説の中で、最適な解を収集されたデータの分析を通じて選択する問題として取り扱うとよいだろう。

- (4) システム監査は、複雑で不確定な問題を扱い、かつ専門知識を必要とする分野であり、同時に急速に普及を必要としているテーマでもある。その意味

表 2-1 システム監査と医療診断のプロセスの類似性

システム監査	医 療 診 断	対応するE S機能
I. 予備調査（監査） ①質問 ②検証	I. 予診 ①問診 ②聴診	I. フェーズ：Ⅰ ①各結果についての入力 ②（本）監査／本診の際の指示事項出力
Ⅱ. （本）監査 ①質問（内部統制等） ②監査人による検証 ③その他監査技法・ ツール適用 ・テストデータ法 ・汎用監査プログラム法 ・並行シミュレーション法 ・スナップ・ショット法 ・SCARF法 ・ ・ ④監査調査書の作成	Ⅱ. 本診 ①問診 ②聴診 ③その他検査の適用 ・血液／尿等の検査 ・X線検査 ・各種内視鏡検査 ・組織検体検査 ・CTスキャン ・ ・ ④カルテの作成	Ⅱ. フェーズ：Ⅱ ①ここまでの結果入力 ②技法・ツール／検査項目の示唆 ③結果の再入力により追加監査／ 検査項目の示唆 ④その他各種ヘルプ機能による問 い合わせに対する回答 ・ ・ ・ ・ ⑤調書／カルテの出力
Ⅲ. 報告 ①監査結果の意見表明 ②改善の勧告 ・改善点 ・次善策の呈示 ・ ・	Ⅲ. 診断・治療 ①診断結果の明示 ②治療方針の確立 ・投薬処方 ・手術 ・ ・	Ⅲ. フェーズ：Ⅲ ①適否／病気有無（病名）の判定 ②改善／治療方針の示唆 ・ ・ ・

では、E Sによる解決が最も似つかわしいと思われる。

2.2 システム監査局面別に見たエキスパートシステムの適用

2.2.1 監査計画局面へのエキスパートシステムの適用

システム監査の監査計画局面では、次のような作業がある。

- ・ 監査計画の作成
- ・ 監査対象（領域）の選定
- ・ 監査項目の選定と監査優先順位の決定
- ・ 監査手続き、技法、監査ツールの計画と選定
- ・ 監査スケジュールの作成

以下に、これらの作業項目対応に、E S適用への可能性と期待をまとめる。

(1) 監査計画の作成への適用

長中期計画及び年次計画の作成時に、リスク及び影響度の大小を勘案したプランニングが必要となる。これを過去の実績に照らしながら、現有的人的資源に割り振っていく際に、監査計画専門家の固有知識を利用したESが有用となろう。この場合には、与えられた要件を満たす最適の仕様あるいはモデルを生成する設計型意思決定の問題として取り扱うことができよう。

(2) 監査対象（領域）の選定への適用

「監査領域選択法」の過程を全てシステム化、自動化することである。即ち、事務所、営業所等監査ユニット毎に、選択のキーとなる財務情報や運営情報、前回監査情報を知識データベースとして、これに関連付け、ウェイト付けによる推論を行い、次の監査対象を設定する第一次判断を行う。この利点は、知識ベース及び既存データベースとして必要なものと、推論過程が一般の経営情報システムまたは意思決定システムに極めて近く、共通要素が多いことである。したがって、規模、範囲、機能の拡張の費用／効果の点で優位であり、組織内で開発合意が得易いと思われる。

また、曖昧性を伴う統計処理能力に着目して、開発段階のシステム監査支援に限定すれば、ソフトウェア工学のいくつかの成果が利用できる。例えば、機能から規模の見積、規模から工数／期間の見積、品質や工程管理等については、定量的なモデルが提案されている。個々のプロジェクトにとっては、精度の問題があるが、平均値からの乖離が、問題領域の可能性を示唆しているという点で、監査領域の発見には十分貢献できる。

(3) 監査項目の選定と監査優先順位の決定への適用

この作業においては、過去に実施した監査項目、範囲、発見事項等のデータを参照することによって、ESが今後実施すべきいくつかの監査項目や優先順位、予想されるリスクを類推することにより、監査人に対し監査のための情報提供をすることが可能となる。

(4) 監査手続き、技法、監査ツールの計画と選定への適用

監査項目やその範囲の決定がなされた後、監査実施に当たって内部統制のための質問事項、必要な監査手続き、適用すべき技法やツールが、適用する順序をも含めてシステムにより示唆される。また、使用すべきガイドラインや質問書、監査手続書を出力することも可能となるだろう。

この他、監査上の注意事項、重点的に検証すべき監査箇所等を示すこと

も可能となる。

(5) システム監査スケジュールの作成への適用

全体のシステム監査項目にわたる年間、月間等の計画の提示、また各項目毎のシステム監査スケジュールをESが出力することによって、これを参考として監査日程を組むことが可能となりうる。

2.2.2 監査実施局面へのエキスパートシステムの適用

システム監査の実施局面では、次のような作業がある。

- ・ 開発過程の管理
- ・ 検証済過誤類似事象の抽出
- ・ 変更プログラムの正確性管理
- ・ 予備監査の実施
- ・ 内部統制等の質問の実施
- ・ 監査人自らによる検証

以下に、これらの作業項目対応に、ES適用への可能性と期待についての検討結果を取りまとめる。

(1) 開発過程の管理への適用

開発過程監査の主たる内容は、システム開発管理のそれと極めて類似しており、ここへのESの適用は共同開発・共同使用が可能のように思われる。それに加えるべき監査特有の機能は、他の監査ESのそれと同様な報告・取りまとめの過程が主なものであろう。したがって、システム開発について高度な開発管理手法を持ち、正確な工程管理が行われているならば、システム監査もまた高度化出来ると思われる。

(2) 検証済過誤類似事象の抽出への適用

既往監査で検証された重要過誤については、ESを適用することによって、監査範囲を拡大し、または基準時点を変更してシステム内の全てにわたって条件検索して、同一内容の過誤を全て抽出することが可能になる。従来のシステム監査ではサンプリングの内容によって全体の状況を類推するに止まっていたものであるから、正確に全量把握することによって、具体的改善活動に繋げ易くなる。

(3) 変更プログラムの正確性管理への適用

プログラムを変更した時、その変更が正確になされたかをチェックする

のに適用できる。即ち、変更プログラムの情報を自動収集し、監査人が別途入手して入力した変更承認ドキュメント・ベースの情報と突合して、適否の判定を行う。変更前と変更後の、プログラムのコード比較も必要である。ドキュメンテーションの水準が向上し自動化される等の条件が整えば、一層の威力を発揮するものと思われる。

(4) 予備監査の実施への適用

対象監査項目に対する予備的監査局面において、業務等の内容の把握や概略的検証の実施のための手続きの問い合わせに対して回答を示し、補足事項の摘示、問題や疑問等に対する問い合わせに対しても解決方法を含めて回答を示してくれるであろう。

(5) 内部統制等の質問の実施への適用

予備監査等の結果を受けて、内部統制質問書の選定とアウトプット、質問結果による内部統制の信頼度の評価、補足すべき質問事項、その結果による今後の監査方針をシステムが示唆することが可能となろう

(6) 監査人自らによる検証への適用

内部統制の質問を終了した後、監査人自らによる監査の実施に移ることになる。ここでは、各種の監査技法やツールを駆使して、監査証拠の入手を図ることが監査上の主要な業務である。そこで、監査結果の一部や発見事項、証拠の内容を入力することによって、ESはこれ以降に行うべき作業や、より適切なまたは追加すべき手続き、技法、監査ツールの示唆を行い、また問題点の解決のためにヘルプ機能の提供も可能にすることになる。更に、従来から監査人が自ら行っていた監査調書の作成についても、以上に述べた入力結果に基づきESに肩代わりさせて出力させることも可能であろう。

また、情報システムの開発段階で開発プロジェクト組織の妥当性を診断し、問題点の指摘、助言と勧告を行うための作業にもESの適用の可能性がある。例えば、次のようなESが考えられる。

① ESは、システム監査人あるいはプロジェクトリーダーに対し、当該プロジェクトの規模、予算、要員数、システムが採用する技術及びアプリケーションに対する要員の経験などを質問する。

⇒ ESは、そのプロジェクトの要員が未経験のサードパーティソフトを含んでいること（潜在リスク）を発見する。

- ② E Sは、サードパーティソフトについて、要員への技術教育の期間、工数、それが開発対象システムに与える影響を、システム監査人あるいはプロジェクトリーダーに質問する。

⇒ E Sは、技術教育の時点が遅すぎることを、その技術が開発対象システムにとってクリティカルであることを発見する。

- ③ E Sは、サードパーティソフトに対する対応が問題であることを指摘する。また、技術を持った要員の投入、または早期技術教育とプロトタイピングの必要性及びその結果として発生する予算と納期の見直しをシステム監査人あるいはプロジェクトリーダーに助言する。

2.2.3 監査報告局面へのエキスパートシステムの適用

システム監査の監査報告局面は、内部統制の質問や監査人自らによる検証の結果、または入手された監査証拠によって、監査人がシステムを評価し、その適否の意見を形成し、更に改善すべき事項については改善に関する勧告、または提案をする段階である。監査報告書を提出することによって、監査は一応の終結を見ることになる。

この最も重要な報告に関する過程の一部にも、E Sが利用できる。即ち、既に入力の終了したデータ即ち監査実施局面での監査結果に基づいて、E Sが適否に関する推論を行い、改善の必要な場合は、最適な方策を示すよう推論機構を含めてシステムを作成することが出来るものと思える。また、必要な補足入力によって、監査報告書の作成も可能となろう。

2.2.4 フォローアップ局面へのエキスパートシステムの適用

システム監査においても、監査結果を監査終了後の適切な時点でフォローアップすることは是非必要であり、重要な作業である。

この局面では、次のようなE Sの適用が考えられる。

- ① E Sを利用して当該監査項目の固有の発見事項や改善の勧告、提案事項の内容を確認するためにアウトプットを求めること。
- ② 改善状況の所見をシステム監査人が入力することによって、当該システム監査結果と照らし合わせて、状況の評価及び更に改善を要する点を表示すること。

2.3 システム監査へのエキスパートシステム適用の効果・期待

システム監査には、以上に述べてきたように、ES自身の開発・発展過程にもよるが、システム監査のプロセスにおいては、十分な知識ベースの構築に基づいて、慎重な開発が行われるならば、ESそのものがベテラン監査経験者に近いレベルでの示唆や効果を生み出すことが可能だと言うことは想像に難くない。

その適用の効果と期待については、以下のようにまとめることが出来る。

(1) 監査業務の効率化と省力化

監査計画局面における監査項目や範囲の選定及び監査計画の作成、監査実施直前における内部統制質問書や監査手続書のアウトプット、監査実施時における補足的または不足の手続き、技法や監査ツール適用の示唆、監査調書及び報告書の作成等がESによって行われることにより、監査業務の効率化及び作業期間の短縮、省力化に大きく貢献するであろう。

(2) システム監査専門家不足の解決

システム監査は高度の専門的知識・技能が求められる分野であるのに対して、一方その監査の必要性は日々高まってきている。しかしながら、専門家の育成は極めて困難を伴い、人材の確保が出来ない状態が存在することは、情報処理分野におけるソフトウェア・エンジニア不足以上に深刻なものとなりつつあるし、この状態は更に進むものとなろう。

その解決の一助となりうる可能性を秘めているESの構築は、早急に検討を要するし、その完成は期待されることになるだろう。

(3) システム監査知識の共通化

ESの知識ベース構築は、短期に達成できるものではない。また、ES自体の開発もそのためプロトタイピング的構築により徐々に知識や推論方式の追加・改良を行いながら開発を進めていくことが望まれる。その際、多くの監査担当者による監査経験がESの知識ベースに蓄積されることによって、監査人固有の知識が共有の知識となりうる。これは、ESによる最大のメリットの一つと言っても過言ではない。

(4) 高度の専門知識の活用

システム監査は、監査関連知識、情報処理関連知識、適用業務知識など様々な分野の専門知識を要する分野である。しかも、それぞれがかなりの高度の技術水準を要求されるだけに、その全てに精通した専門家は少なく、かつ

短期間に大量の専門家を養成するのは不可能に近いものといわねばならない。

実際に、相当期間監査の経験のある人でも、情報処理関連の通信ネットワーク・システムのセキュリティ対策や各種システムのリスク・アセスメント、適正なサンプル抽出のための数学的技法などには必ずしも精通していない場合も多く、すべてにオール・マイティな専門家の数は極めて限られているのが実情である。

適用業務関連についても、JUAN(Junior Underwriter Assistance Network)やExMARINE(Marine Underwriting System)などのESの開発によって、経験の浅い保険担当者でも、損害保険関係のリスク査定が随分と楽になっているが、その考え方は、当然のことながらシステム監査人の対象システムのリスク・アセスメントにも大いに役に立つはずである。

(5) 経験やノウハウの顕在化・体系化と蓄積

ひとくちに監査対象業務の問題点や弱点の発見といっても、長年の経験や勘に依存する部分も多く、これらは専門家といえども定型化し、顕在的に表現することが困難な部分が多い。これらをES開発支援ツールを使うことによって顕在化し、いわゆる監査に関するメタ知識を体系だった形で表現することができれば、その効果はきわめて大きいものがある。そのためには、監査の専門家とKE(ナレッジ・エンジニア)との頻繁な会話が必要となる場合も出てこよう。QShellやFFASTなどはその好例ということができ、推論エンジン、問題のブレークダウンと解明のための補助機能、ユーザとのインタフェースや知識ベース・メンテナンス機能などがその特色である。

また、ベテランのシステム監査人の常識(知識)を帰納法的手段で取り込み、その常識(知識)を更新していくことによって、組織体としてのシステム監査能力は飛躍的に高まっていくことが期待される。

(6) 推論プロセスの明示と評価

ESの特色の一つに、推論プロセスの明示機能がある。専門家の知識ベースをもとに、専門家の判断プロセスを推論エンジンがシミュレートすることにより意思決定プロセスが進行していくが、その間のプロセスがすべて明示されることから、意思決定者は、いろいろな代替案をシミュレートしながら自ら最適と考える意思決定を行うことができる。

(7) 情報量の拡大

およそ高度の判断を要する業務にあっては、必要な情報がつねに入手でき

るとは限らず、専門家の常識を基に、限られた情報や不足している情報を補いつつ対象業務や問題点の全貌を明らかにすることが必要になってくる。実際、これらの不足情報を集めるのに一々人手に頼ることも困難なことが多い。

E Sの知識ベースと推論機能は、限定された情報量を増強し、不足する情報を推論して補うために不可欠の道具といってもよいのである。

(8) 監査の質の向上とレベルの均一化

人間の判断には、個々の過去の経歴からくる得手分野の有無、思い込みや先入観、偏見などからくる判断の誤りなどがつきまといがちである。その点E Sによる多方面のトップ・レベルの専門家の知識ベースと推論プロセスを参考にしながら、フィードバックを繰り返していくことにより、経験の浅いシステム監査人や当該監査人が不得手な分野の監査にも、全体としてほぼ均質なレベルの監査が期待できよう。さらに、Questionnaire Generatorにより質問書が自動生成されるので、重要なチェック項目を見逃してしまう心配も無くなるわけである。

同じような意味で、経験の浅いシステム監査人の教育・研修にもE Sが大きな効果を発揮することはいうまでもない。

(9) 監査業務のスピードアップ

これには、次の3つの側面が考えられる。

①対象業務の全貌や問題点の把握がよりスピーディにできること。

E S開発支援ツールの利用によって、対象業務のモデル化がより簡単にできるようになる。

対象業務の把握には、フローチャート化するのも一つの方法であるが、IF-THENルールなどを使えば、かなり定性的な業務でも対象業務の記述が相当簡単にできるようになる。TICOMⅢ(The Internal Control ModelⅢ)などは、その好例であろう。

②問題点発見のための質問書作成などが簡単にできること。

支出の裏付けを欠く証憑の欠落、支払請求手続の承認プロセスの欠如などに関する基礎的な質問が自動的に生成されれば、監査業務が質的にも時間的にも大いに能率が上がることはいうまでもない。その結果、監査対象業務やデータ量、監査回数などの増加を図ることができよう。

③監査作業がスピードアップ出来ること。

例えば、「ソフトウェアの品質保証(Software Quality Assurance:

SQA)」、 「リスク評価のためのES (Risk Management Support SystemやEvaluation System)」、 「プログラム機能自動検査システム」などの利用によって、システムの信頼性監査や安全性監査作業が飛躍的にスピードアップしよう。

また、ESによって、監査人のための情報収集や監査手法の説明、監査経過の文書化(監査計画書や監査報告書など)やトップ・マネジメントへの報告などがスピードアップする。監査用ワークステーション(Audit Workstation)や内部監査支援システム(Internal Audit Expert Support System)は、顧客の財務情報や経営環境を評価し、監査計画を作成するのに大きな成果を上げている。

2.4 システム監査にエキスパートシステムを適用する上での問題点・課題

2.4.1 システム監査固有の問題点・課題

ES化の対象がシステム監査であることから、以下のような問題点やES化に向けての課題が考えられる

- (1) 既にシステム監査のESは幾つか開発されてはいるが、実用面では必ずしも成果を上げているとは言えないこと。

それは、一つには監査という領域では、内部統制レベルの評価やセキュリティ・アセスメント一つをとっても、必ずしも定量的評価ができ難く、定性的判断領域が多いことと大きな関係がある。

この問題は、今後共簡単には解決しないと思われるが、いかにして潜在的な専門家の経験や勘を顕在化し、定型化していくかがこれからの大きな課題と言えよう。そのためには、一刻も早くより多くのシステム監査KEを養成していくことが望まれている。

- (2) システム監査の技法そのものが未だ完成していないこと。

現在のES適用例の全てに共通して言えることは、専門分野としての体系化、特に方法論、手続き、技法等の専門的知識が現在の知識レベルの段階において十分、またはそれに近い段階までに完成され、システム化が比較的容易となっている段階で、ここにESを当てはめたものが開発例として成功していると言える。それゆえ、システム監査の場合においても他の事例と同様に、今までシステム監査人が使ってきた在来の監査用チェックリストを、そのまま無器用にシステムに写し込む程度であってはならず、

現在までに得られたシステム監査の専門知識の普遍的体系化、即ち方法論、手続き、技法の普遍的確立と実務への適用の確実性が見られるようになって、その後にE Sの構築を図ることが成功への前提となると言えよう。

- (3) E S化の大きな問題点は、ルールの数がどれ程あれば実用に耐えられるシステム監査E Sが作れるか、また、それだけのルールが現時点で識別できるかにある。更に、ルールの質の面からも、複数人のエキスパートからの知識を収集し、整理することが必要であろう。そのためには、体制の整備と知識の評価が重要である。
- (4) E Sでは、その適用経験をフィードバック情報として取り込めば、専門家モデルの基本解構造、対象モデルを成長させ、知識を共用できる仕組みを確立することができる。したがって、現在のE S開発支援ツールの機能範囲で、十分にシステム監査E Sが構築可能である。

2.4.2 エキスパートシステムの一般的な問題点・課題

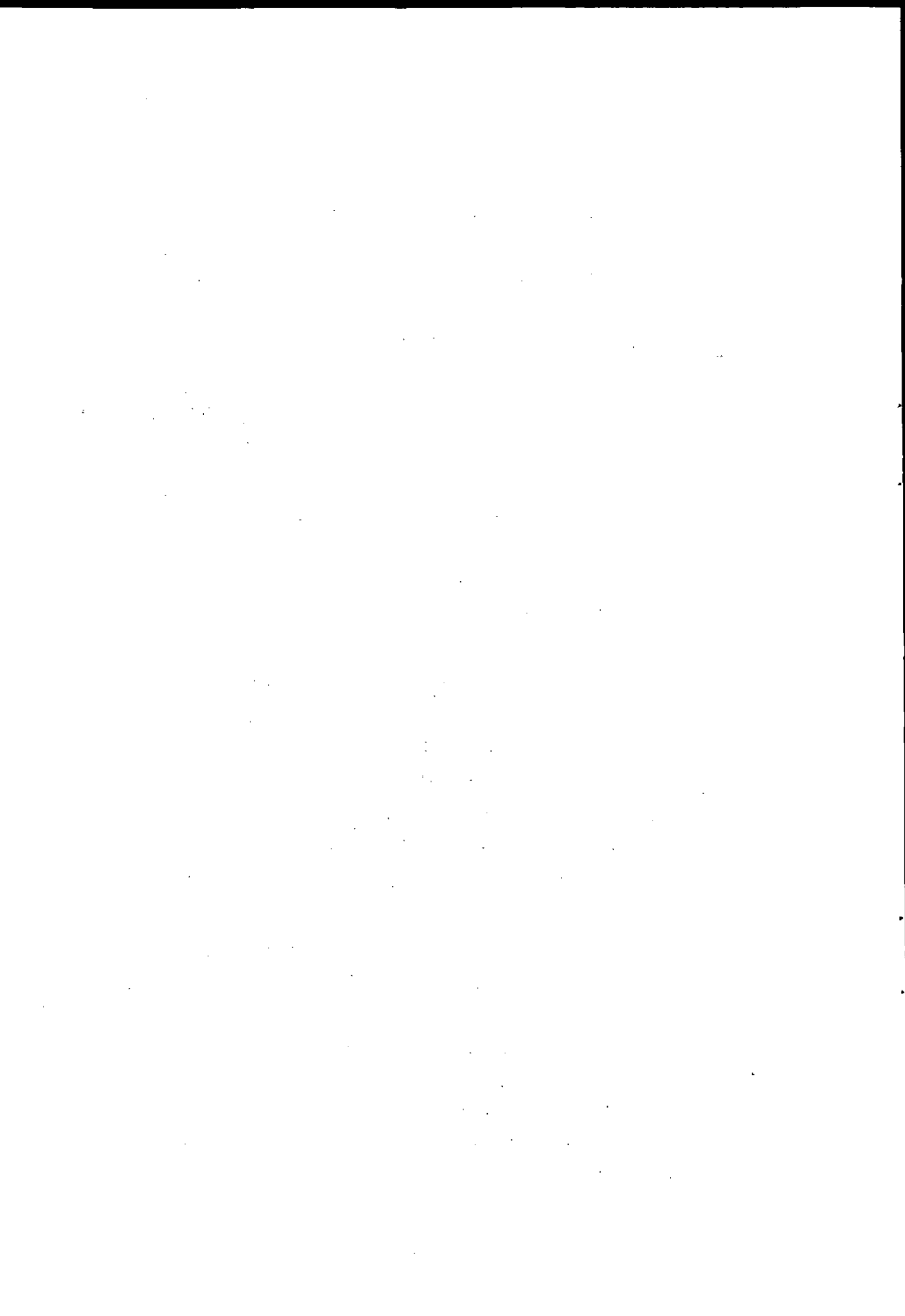
適用分野に関係なく、現在E Sが一般的に抱えている問題点やE S構築上の今後の課題について取りまとめると、以下のようになる。

- (1) E Sの開発支援ツールのレベルもまだ必ずしも高水準とはいえない。

今後更にパワフルで使い易いE S開発支援ツールや監査用システムの開発が期待される。その面で、昨今のE Sをめぐる開発気運の急速な盛り上がりは、本調査にも明るい期待を抱かせそうである。

E Sアプリケーション開発経験の増加に当たって、問題解決過程での戦略や構造の類似性等に着目して、診断型シェル、計画型シェル、制御型シェル等いわゆるドメインシェル（分野対応シェル）が急速に脚光を浴びてきていることは、好ましいことといえよう。

- (2) E S構築の前提となるツールについても、高度の推論処理が出来るエンジン（推論機構）を含めたシェル及びシステム構築に適切な言語が用意されていることが前提となる。
- (3) 今後、多数のシステム監査E Sが実現することが予想されるが、その実現に当たって留意すべきことは、どれ程良好なヒューマン・インタフェースを形作れるかにある。E Sの場合には、一般の情報処理システムにおけるものに加え、E S特有の説明機能や知識の矛盾に対するシステムからの問い合わせの機能も含まれる。



付 録

米国におけるシステム監査へのエキスパートシステム 適用実態調査の詳細

本付録は、米国におけるシステム監査へのエキスパートシステム適用実態調査の詳細を取りまとめたものである。

目 次

1. 環境分析調査の詳細	1
1.1 システム監査エキスパートシステム	1
1.2 導入対象としてのシステム監査	5
1.3 導入主体としてのエキスパートシステム	8
1.4 エキスパートシステムの事例	11
1.4.1 概 要	11
1.4.2 システム監査領域のエキスパートシステム	13
1.4.3 システム監査関連領域のエキスパートシステム	29
1.4.4 システム監査人の評価	39
2. エキスパートシステム導入検討の詳細	43
2.1 導入検討の方針	43
2.2 導入の有効性の検討	51
2.3 導入の困難性の検討	54
3. システム監査におけるエキスパートシステム導入可能性検討の詳細	59
3.1 作業領域と機能	59
3.2 エキスパートシステム導入における考慮点	62
3.3 今後の動向 — 米国大手会計事務所の動向 —	65
参考文献	68
エキスパートシステムの索引	69

In addition to record-keeping, the document highlights the need for transparency and accountability. It suggests that all financial activities should be clearly documented and accessible to relevant stakeholders. This helps in building trust and ensures that the organization's financial health is well-monitored. The document also mentions that proper record-keeping is essential for compliance with legal and regulatory requirements.

The second part of the document focuses on the management of cash flow. It explains that maintaining a healthy cash flow is crucial for the survival and growth of any business. The document provides several strategies to improve cash flow, such as speeding up receivables, negotiating better terms with suppliers, and controlling expenses. It also stresses the importance of having a contingency plan in place to handle unexpected cash shortages.

Finally, the document touches upon the role of technology in financial management. It notes that modern accounting software can significantly streamline the process of recording transactions and generating financial reports. However, it also cautions against over-reliance on technology, emphasizing that human oversight remains essential to ensure the accuracy and security of the financial data.

1. 環境分析調査の詳細

1.1 システム監査エキスパートシステム

(1) システム監査ESの捉え方

以降の検討においては、システム監査エキスパートシステム（以下、エキスパートシステムをESと記す）を「システム監査人特有の判断を必要とする領域において、システム監査人の作業を代行するシステム」と捉える。従来のシステム監査支援ツールも、限定された領域ではこのような機能を持つとみなすことができる。すなわち、結果的に成しうる機能面から2者は連続している。

ここでは、システム監査ESを、プロダクション・ルールやフレームを用いた知識ベース型ESに限定してはいないが、イメージにあることは否定しない。そしてそれは現在の技術環境をそのまま反映したものであると考える。

上記の他に、システム監査ESに関連する共通な認識として次のようなものがあるだろう。

① 導入の実績は数少ない。

システム監査ESは導入の緒についたばかりである。事例数が少なく、かつ実務環境で大きな実績をあげた例は公表されていない。

② 導入の意義は大きい。

システム監査は情報化社会に必要不可欠であり、かつ高度に専門的である。ESは、自己拡張を実現し情報化社会の将来を担うものとして、どのような専門分野でも基本的に待望されている。システム監査も例外ではなく、その意味で導入の意義は大きい。

③ システム監査作業に変化をもたらす。

ES導入が、従来のシステム監査というフレームワークにおける単なる部分的代替でありつづけるとは考えにくい。費用対効果の点でシステム監査人による作業となりえなかった領域がシステム監査ESの対象となり、システム監査作業自体に変化をもたらす。

④ ES機能に要求を出す。

ES構築ツール（シェル）のサービス機能は、各対象領域ごとに程度の異なった有効性をもたらすと同時に、固有の制約をかける。システム監査以外の領域を対象に作られてきたそれらのものがシステム監査に最適か否かはわからない。多様なニッチ(Niche)を求めているベンダーのES構築ツールに、システム監査側から要求を出すことが考えられる。

(2) システム監査ESの使用環境

ES導入に際しては、前もって使用環境を明確化しておく必要がある。とにかく着手すべきだという提言は、ES構築ツールや言語の選択段階であれこれ決めかねている場合にしばしば有効である。

しかし、先ずプロトタイプ開発を行えばよいという意見にとびつくのは危険である。よく目にするのはプロトタイプ開発までは進んだが、“このあとどう作り込んでいこうか”と困ってしまい放置されるケースである。また、構築されたESも、そのあとの取り扱いについて明らかにされてないと、用いられなくなる危険性が高い。すなわち、導入の基準、普及のための研修、ユーザの声をどのように反映するか、重要なノウハウをいったいどのように社内に保存するか等は、予め検討しておく必要がある。

システム監査ESの使用環境は、直接の使用者に依存する。システム監査の実施状況を見ると、実施主体は、企業の内部監査部門である場合が多い。次にあげられる実施主体は、大手の会計事務所やコンサルティング会社等、外部のシステム監査をビジネスとする組織である。この2者は、従来からシステム監査の実施主体であるが、ES導入後もそうあり続けるだろう。ES導入により主体の被監査システムに対する客観性が確保されたなら、EDP部門も直接の使用者として、浮上してくる可能性がある。

以下では、システム監査ESの使用者別にその使用環境を述べる。

(a) 企業の内部監査部門

企業の内部監査部門でのシステム監査ESの主な使用目的は、次のとおりである。

- ① 客観的視点の獲得（他社との、また、時系列の）
- ② ノウハウ、知識の社内への定着
- ③ 作業負荷、場合によっては経費の軽減

我が国の現状では一般にシステム監査人が不足しており、内部監査部門の監査経験者が内部監査部門以外のEDP専門家の協力を得て、システム監査を実施することが多い。この場合、システム監査ESには、特にEDPに関する知識が内在されていることが強く要求される。また、市販のESの場合、情報システムの特徴等から監査目的に則したものを選択しなければならないのは言うまでもない。

監査対象の情報システムへのシステム監査ESのセッティングは、基本的に導入時のみ考慮すればよいと、要求は少ない。使用頻度は、企画、開発段階では各チェックポイントで、運用段階では原則として定期的であり、それほど多くはない。また、知識ベースの改訂について

は、負荷も比較的少なく、担当部門も明確であり好環境にある。すなわち、ポイントは許容される予算内で情報システムに適合するものが導入し得るか否かである。

(b) 会計事務所、コンサルティング会社

会計事務所及びコンサルティング会社でのシステム監査ESの主な使用目的は、次のとおりである。

- ① 作業効率の向上
- ② 優れたノウハウの共有によるサービスの向上
- ③ 差別化による競争優位の確立

監査作業には高度な倫理性が要求される。そのため、システム監査ESは、それを損ねるような誤出力は避けなければならない。差別化を重視しない場合は、システム監査ESをシステム監査の後方支援ツールとして考えればよいが、差別化を重要視するなら、操作性や表示方法についても簡単なキー操作にするとか、グラフ、図を使う等、ユーザーフレンドリーなインターフェースが要求される。

顧客の情報システムから直接データを収集する場合は、システム監査ESのセッティングが容易でなければならない。会計事務所、コンサルティング会社では、当然システム監査ESの使用頻度は高く、したがって様々な対象に適合させるためのメンテナンス負荷は大きくなる。

また、ESは、システム監査の高度な専門業務を部分的ではあるが代行する。システム監査人はこれに拒否反応を示すのではないかという考えを、アンケート結果(1.4.4参照、参考文献2)は否定している。

(c) 企業のEDP部門

企業のEDP部門でのシステム監査ESの主な使用目的は、次のとおりである。

- ① 自己チェックによるシステム品質向上
- ② システム監査受入れ体制の整備

EDP部門が監査ノウハウを得るのは難しいと思われるので、ここで使用するシステム監査ESには、特に監査に関する知識が内在されていることが強く要求される。

使用頻度等は内部監査部門と同様ではあるが、運用に割り当てられる工数は、システム監査実施の強いニーズに基づいていないため少なくなると思われる。さらに、知識ベースの改訂担当者についても明確に取り決めておく必要がある。

EDP部門が、従来行ってきた様々な品質保証作業と、導入したシステム監査ESによるそれとは、最終的な狙いは同じだが、外観上異なる。さらに、前述の2つの使用主体は、基

本的にこれまで行っている作業をESに代行させているだけであり、ESの出力に対する評価も比較的容易に実施可能であるが、EDP部門では困難を伴う。このことは、ESに対する過大な期待や失望、そして過少評価に陥る危険性が高いことを示唆している。

企業のEDP部門においても、強力なトップダウンに裏打ちされていれば、上記使用目的を達成し、情報システムの健全化に大きく貢献するだろうが、様々な困難が予想されるため、明確な運用ルールの取り決めが、他の2者に比較して最も強く要求される。

(3) システム監査ESの構築環境

我が国におけるESのビジネスへの導入は、先進ユーザーによって1984、85年頃から試みられている。それらの構築は、EDP部門の主導による場合がほとんどであった。すなわち、対象領域の専門家の協力を社内に得、EDP部門がナレッジエンジニア（KE）の役割を兼ねて構築している。ユーザーが主体となるケース、またメーカーや外部専門家の強力なコンサルティングを受けるケースは、各々3分の1程度である。

これらの先進企業で、ES構築にEDP部門が関係するのは、システム構築環境が整備されているというより、むしろ組織上の切り分けによるものであろう。何故なら、ES構築作業と、従来のシステム開発のノウハウとでは、端末やパソコンを用いること以上の共通点を見出すことは難しいからである。外部からのコンサルティングを受けるケースが少ないのは、対象領域の専門家が社内で得られるためである。すなわち、そのような対象領域を選定し、ES構築を狙ってきたとも見做すことができる。KEに関しては、現状では素質を持つ社員の代替よりメリットが大きい外部委託が、起こり難いためである。

すなわち、現在みられるESの導入は、業務処理における強い必要性に迫られたものではなく、行い易い対象領域から比較的トップダウン的傾向のもとに為されていることを示している。強いユーザーニーズに基づいていないのは、どのようなことがESに出来るのかについて、ユーザーの認識が形成されていなかったこともあるが、このような状況は、最終成果物であるES形態が、開発を手がけるシステムエンジニア（SE）の要求する方向に合わせられる可能性を内在している。ちなみにその方向の中で注意を要するのは、適用容易な領域または機能での妥協と、つじつま合わせだけの知識獲得の2つである。後者は、推論パフォーマンスが悪い原因を探る過程でしばしば見られたもので、全く使用しないような知識が多数知識ベースに格納されていた。

以下に、ビジネスという観点から、システム監査ESの可能性の高い構築者を次の3つに分類し説明する。

(a) 大手の会計事務所、コンサルティング会社

大手の会計事務所及びコンサルティング会社では、システム監査をビジネスとして捉えており、構築するシステム監査ESは、複数の企業の情報システムを対象とする。

また、システム監査人を多数擁し、KEも得易い環境にある。作業効率の向上、差別化、知識の共有、システム監査人の育成を狙うため、ES機能は、当初から汎用的なものに設定され構築される。

(b) 先進企業の内部監査部門

先進企業の内部監査部門で構築するシステム監査ESは、基本的には自社の情報システムを対象とする。専任のシステム監査人が複数いる場合もあるが、一般にノウハウは、数少ない固有の情報システムに依存することになる。

KEは、内部の担当者が役割りを代行することが多い。内部監査部門に蓄積された知識の整理、体系化及び社内定着化や、実施者や実施時期にとらわれない客観的視点獲得を狙うため、ES機能は、自社の情報システムの状況を効果的に把握するように構築され、汎用性への要求は少ない。

自社の情報システムを長い期間監査してきた経験は、特定条件下で非常に優れたESを構築する可能性をもたらす。

(c) 大学、研究機関等

大学、研究機関等では、ビジネスへの貢献を考えないわけではないが、システム監査領域の解明という目的を満たす形でESは構築される。そのためには、研究者がKEの役割りも代行することになるだろう。システム監査作業のうち、知識構造を明らかにし得る専門領域が主に対象となり、システム監査研究において本質的作業である情報システムの診断にES機能が絞られる傾向がある。

ES機能は、現場の実務者に使ってもらう必要があり、ES評価が上記2つの構築主体と比較して大きなテーマとなる。

1.2 導入対象としてのシステム監査

(1) システム監査作業

システム監査は、セキュリティ対策の実効性を担保し、システムの有効活用を実現するという目的をもつ。その目的に対し最大の効果を効率よく達成するために、システム監査作業はある程度定型的な形態をとる。それが監査方針の確立から報告書のフォローアップへ至る一連の流れである（表付1-1）。システム監査は原則としてこのような流れに沿って行われるが、作業の内

容はそれぞれのシステム監査によりもちろん異なる。また、しばしば作業の流れ自体も状況に応じて変化していく。準拠性テストと実証性テストの切り分けはこれに該当する。

表付1-1 システム監査作業フローの例

段 階		作 業 項 目	内 容
計 画	1	プロポーザル作成	監査ニーズ等把握
	2	基本計画の作成	年次計画（対象選択を含む）
	3	個別計画の作成	テーマ別詳細スケジュールリング
実 施	予 備 調 査	4 関連資料の収集	組織図、システム構成、処理機能概要等
		5 チェックリスト配布	内部統制質問書等
		6 対象システムの概要把握	対象領域把握と作業方針策定
	本 調 査	7 本調査手順の作成	フローチャート、インタビューポイント
		8 意見聴取	各担当者にインタビュー
		9 ドキュメント調査	テーマ関連資料の閲覧
		10 監査支援ツールの使用	監査ソフトウェア等
		11 調査結果の分析	対象の詳細な把握
	評 価 ・ 結 論	12 解釈と問題点浮きぼり	システム監査の視点による解釈
		13 問題点の評価	具体的把握または定量化
		14 意見交換	被監査部門との打合せ、監査証拠の確認
		15 最終結論決定	マネジメントとの調整
	報 告	16 報告書作成	結論をフォーマット化
		17 報告書提出と報告会開催	マネジメントへ報告
		18 フォローアップ	改善案の実現

システム監査により作業が異なるのは、対象となる情報システムが各々独自の特徴を備えているためというよりも、むしろマネジメント等の情報システムに対する期待の質や程度が異なるためである。これはシステム監査個別計画の重点監査項目や、処理の信頼性に対する監査手法の精密さの度合に反映される。システム監査作業の多様性が、情報システムの結果として備わった機能や構成の多様性のみに依存するなら、実施マニュアルのようなものを参照しながら、専門以外

の人間がシステム監査を行うこともまだ可能かもしれない。しかし、組織という視点からはマネジメントの情報化戦略、また情報化社会という視点からはシステムの公共性等により、システム監査作業は変化するため、専門のシステム監査人が必要不可欠とされている。このようなシステム監査作業の特徴をあげると、次のようになる。

- ① 分析、解釈、評価といったデスクワークだけではなく、現場視察や意見聴取などのフィールドワークも含んでいる。
- ② 業務内容の機密保護等、高度な倫理性が要求される。
- ③ 意見交換、フォローアップ等効率的な作業の遂行は、対象部門との強い信頼関係の上に成り立っている。
- ④ 被監査部門が十分納得する説明を要求される。
- ⑤ 最終的な監査結論に対し責任を持たなければならない。
- ⑥ これまでES化が実現された他の対象領域と比較しても、システム監査領域の論理的難易度はかなり高い。

(2) システム監査知識

ある問題解決をESに期待する場合、対象領域で使用される知識は何らかの形ですべてES内に取り込まれなければならない。知識ベース型のESでは、知識をある一定の表現形式に定型化し、知識ベースに格納することになる。

対象領域での問題解決のために、システム監査人には次に示すように広汎な知識と能力が必要とされている。

- ① 監査対象としての情報システムを正確に理解し把握するために必要な「知識」
- ② 一連の監査活動を確実に遂行する「能力」
- ③ 自らの評価基準を求めるため及び監査活動を円滑に進めるために必要な「知識」

上記の「知識」と「能力」とは、ESに格納するという観点から見た場合、性質がいくぶん異なることを反映している。

すなわち「知識」と比較して「能力」は、次のようなニュアンスを持つと思われる。

- ・ 動的なものである。
- ・ 先天的なものに多く依存している。
- ・ 各々を断片的な知識として認識し難い。
- ・ どの時点で修得した知識か明確に意識し難い。

また、①、②、③を「システム監査基準」により詳細に見ていくと、システム監査に必要な知

識は、次のような特徴を持つことがわかる。

- ・量的に膨大である。
- ・マネジメントの経営戦略からコンピュータの技術動向まで、多階層の知識である。
- ・総合的判断の集積からなる経験的知識である。
- ・技術革新、コンピュータ犯罪動向等により更新が著しい。

1.3 導入主体としてのエキスパートシステム

(1) 人工知能におけるES

システム監査作業は、専門家による高度な判断が中核をなす。これは、ESの対象範囲と重なる。ESについて本節で検討を行うが、以下では、その他の人工知能について、システム監査への今後の導入可能性を概観する。

人工知能は、次の3タイプに大別できる。

- ① 知識情報処理システム
- ② パターン情報処理システム
- ③ 産業用ロボット

知識情報処理システムは、ESと、マン・マシン・インタフェースを提供する自然言語処理システムに分けられる。

パターン情報処理システムは、人間の5感を代替するシステムで、音声・画像のイメージ処理等を行う。

産業用ロボットは、人間の行う物理的作業を増幅し、代行する。

自然言語処理システム及びパターン情報処理システムは、システム監査作業において分析の対象となる基礎資料の収集に応用領域を見い出すことができる。それはインタビューやドキュメント調査自体に、ある程度システム監査人のノウハウが反映されている可能性があるからである。産業用ロボットに関しては、システム監査人であるがゆえの行動を代行するという意味合いに乏しい。

ビジネスの分野では、ES及びパターン情報処理システムの人工知能が実用化されているものが多い。ビジネスの観点からも、ESは人工知能の中で最も重要な分野の一つである。

(2) ESの現状

ESにも様々あるが、システム監査への導入主体として見た場合、いったいどのようなESがクローズアップされるのかを絞りこむ必要がある。そのために以下では、ESを、①応用分野、

②機能タイプ、③機能レベルの切り口で分類する。ただし、それぞれの切り口における分類の網羅性や項目間の等質性にはこだわらず列挙する。

(a) 応用分野

成功したESの存在する応用分野は、次のように考えられている。

- ・医 学
- ・資源探査
- ・情報処理
- ・農 業
- ・自然科学
- ・工 学
- ・人文科学
- ・教 育
- ・経 営

これらには学問的な分野が多いが、それはESの発展過程を反映している。ESは、大学や研究機関での基礎研究から、実用への端緒となったエンジニアリング分野での開発を経て、現在ビジネスにも応用され始めている。

上記の分野は、対象領域として非常に難しく専門的であるが、ESはその中のどのような知的作業を代行したのか注意を要する。すなわち、これらの分野に応用可能なら、システム監査においても応用可能かもしれないと即断する前に、機能をブレイクダウンし、ES事例の入力と出力を明確に見極める必要がある。

(b) 機能タイプ

ESの機能タイプの分け方には、次に示すようにいく通りがある。

〔分類例1〕	〔分類例2〕	〔分類例3〕
・予 測	・解 釈	・設 計
・設 計	・診 断	・分 析
・計 画	・監 視	
・分 析	・制 御	
・診 断	・設 計	
・指 示		
・制 御		

結果的には、ESが実施した機能を専門家が行っているものとして捉え、その行為の意味を表現した傾向があり、上記のような分類は必ずしもESの構造と直結しない。すなわち、そのまま構築の難易と関係しない場合がある。ただし、設計型と分析型の分類は、ある程度ESのアーキテクチャを反映している。

(c) 機能レベル

現在知られているESの機能レベルは、実現する知的作業レベルから3ランクに大別することができる。

a. 高度判断支援レベル

病気の診断、ディーリング、地下資源探査等、かなり高度な支援または代行を行うレベルである。

b. 相談照会レベル

生命保険の商品選びや住宅ローンの相談など、窓口で新商品の紹介をしたり、簡単な相談を行うレベルである。

c. ルールブックレベル

システムの処理対象を数値から文字にし、マニュアルや規定集等を使いやすくしたレベルである。

ビジネスに実用されていると言われるものの多くはbレベルである。aレベルで公表されている成功例は氷山の一角で、水面下にはこのレベルを狙った多くの失敗例があるものと思われる。cのレベルは、人工知能とか専門家システムに対する期待とかのギャップが大きいいためか、あまり知られていない。

一方、ES構築ツールを提供するメーカー側を見ると依然活発な動きがある。AAA I-87 (米人工知能協会) 総会では例年と趣をやや異にし、メーカー側よりもユーザー側 (かなりの研究機関を含む) の発表が相対的に活発だったが、メーカー側は製品開発・バージョンアップを積極的に続けている。また発表では現実味のあるESが多く報告された。

ESの技術環境の動向を挙げると次のようになる。

① 研究型から実務対応型へ

LISP、PROLOG等の言語的制約が取り除かれつつある。例えば、製品の動作環境がLISP専用マシンのみだったものが、汎用ワークステーションでも使えるようになったり、パフォーマンス向上のためにC言語化が進められたりしている。

② 操作性の追求

特にES構築用のマン・マシン・インタフェースが強化されてきている。マルチウィンドウやグラフィックスは、ほとんど標準的機能としてメーカーから提供されるようになってきた。

③ 従来システムとの統合

パソコンESが安価で普及しても、従来のメインフレームとの接続にはいろいろな問題があったが、各メーカーともホストとの連携を考慮するようになりつつある。また、メインフレームベースのES開発環境整備も行われつつある。

1.4 エキスパートシステムの事例

1.4.1 概要

本節では、システム監査領域及びシステム監査関連領域の事例として20のESを紹介する(表付1-2)。これらの事例は、米国で公表されたものであり、中にはプロトタイプ開発段階のものも含まれている。

システム監査固有領域に限らず、対象事例を関連領域にまで広げたのは、システム監査固有の事例が少なく、それらの統合としてシステム監査ESをイメージすることが難しいためと、システム監査ESの機能としての境界を、システム監査固有領域の外側からも規定していくためである。

システム監査関連領域として、次の4領域をあげた。

- ① 会計監査域
- ② ゴーイングコンサーン域
- ③ セキュリティ対策域
- ④ リスク分析域

表付1-2 調査対象のエキスパートシステム

項 番	システム名	構 築 者				運 用 段 階		対 象 領 域					開 発 時 期
		会計 事務 所	内部 監査 部門	大学 等	その 他の 企業	試行	運用	支払 監査	会計 監査	ファイ ナンス	セキュリティ 対策	リス ク分 析	
①	EDP-XPERT			○		○		○					1986年
②	EXPERT AUDITOR				○		○	○					1987年
③	AUDITPLANNER			○		○		○	○				1987年
④	ARROWSMITH-P			○		○		○	○				1985年
⑤	I.C.A.			○				○	○				1985年
⑥	ARISC			○		○			○				1986年
⑦	Auditor			○			○		○				1985年
⑧	CFILE	○					○		○				1986年
⑨	TAX ADVISOR	○				○			○				—
⑩	EDP-AUDITOR			○		○			○				—
⑪	ExperTAX	○					○		○				1986年
⑫	TICOM			○		○			○				1987年
⑬	TICOM III			○		○			○				1984年
⑭	IST/RAMP		○		○		○	○				○	1982年
⑮	BDSS				○			○				○	—
⑯	T.R.A.M.				○							○	—
⑰	RISKA		○		○		○					○	1985年
⑱	GC-X		○				○			○			1986年
⑲	AOD		○			○				○			1986年
⑳	DISCOVERY		○				○				○		1987年

1.4.2 システム監査領域のエキスパートシステム

システム監査領域のES事例は、各会計事務所内でシステム監査人の後方支援に使われているものか、またはプロトタイプ開発段階であり、著しい効果をあげた実績をもつものは少ない。しかし、今後改善を重ねることによりシステム監査作業に大きく貢献することが期待できる。

以下に、7つの事例を紹介する。

① EDP-XPERT

EDP-XPERT は、システム監査専門家(CAS: Computer Audit Specialist)が、高度コンピュータ環境の中での統制(control)の信頼性を判断するとき、これを支援するESである。これは、ES構築ツールAL/X(Advice Language/X)を使って構築されたルールベースシステムである。初期知識ベースの作成には、米国8大会計事務所のシニアクラスCASが協力した。4つのゴール(監督性のコントロール[規制的統制]、インプットコントロール、プロセッシングコントロール、アウトプットコントロール)について、合わせて133のルールから構成されている。

このESの判断出力に対し、著者(ハンセン、メシア)は17名のCASにアンケートを実施し(参考文献2)、9つの評価項目についてEDP-XPERTを評価したところ、「使用できる」(Reasonable)と判定された。

この後EDP-XPERTのルールベースは、オンラインリアルタイムシステム、データベースシステムの統制(control)を含むように拡張された。そして優先ルール(prior rule)とベーシックゴールには新たなルールが加えられ改善された。ルールベースは再びシニアクラスCASの助言に基づいて3つのゴール(データベース、意思決定ルール、統制戦略[Control Strategy])に変わり、この最新のプロトタイプはシニアCASにより選び出された2つのクライアントの実環境においてテストが行われた。

以下に、EDP-XPERTの開発・改良過程を示す。

【第1段階】

- 〔目的〕①人工知能を、システム監査環境に適用する可能性を検討する。
- ②C A S の知識を表現するのに適する人工知能の構造を決定する。
- 〔実施内容〕・米国8大会計事務所の行ったシステム監査事例に関する資料を研究
- ・上記のうちの5社のC A S とのインタビュー
- 〔結果〕*プロダクションシステムに決定
- 〔理由〕:この時のインタビューでは対象領域が一定しておらず、知識は継続的な改訂が必要と判ったから。その点、ルールベースシステムなら知識ベースの改訂は比較的容易である。
- 〔補足〕:監査人はルールベース形式で思考するとは限らないが、今回のアプローチでは、監査人が行っている推論過程をE S に表現することで、監査人と同じ思考過程を獲得することが出来るものと仮定した。



【第2段階】

- 〔目的①〕C A S の意思決定行動を更に詳しく調べ、E S の知識ベースに適切な意思決定ルール (If-Then)を見出す。
- 〔実施内容〕・インタビュー実施
- 〔結果〕*C A S の意思決定行動についていくつかの点が判ったが、高度コンピュータシステムの監査用E S に適したIf-Then ルールは、多くは得られなかった。
- 〔理由〕:①プロトコル分析で使われた対象ケースでは、ユーザー主導の内部統制がしっかりしていた。②この結果、コンピュータ周辺監査に向いている意思決定ルールを使ってしまったから。
- 〔補足〕:システム監査へのC A S の全体的なアプローチは、モデル構築に際して全般的な指針を提供し、EDP-XPERT で使われているゴール構造に影響を及ぼしたと言える。
- 〔目的②〕対象領域に適切なE S 構築ツールを見いだす。(シェルの決定)
- 〔実施内容〕・利用可能なシェルを全てチェック
- 〔結果〕*A L / X を採用。これはProspectorシステムに基づいている。
- 〔理由〕:診断型の問題に適していたから。A L / X にも限界がある。

〔目的③〕初期プロトタイプを開発する。

〔実施内容〕・プロトコル分析では有効なIf-Then ルールを抽出できなかったのも、この初期知識ベースの開発にあたっては再構築法(reconstructed methods)を使用した。

〔結果〕＊初期のプロトタイプでは、約60のルールを設定した。このアプローチでは、ルールの推論ネットワークを構築するために現在利用できる情報（例えば、教科書、会計事務所の資料）を使用することも含んでいる。この初期知識ベースは、分散処理システムのコントロールに関する一部分のルールと1つの全体ゴール（コントロールシステムの信頼性）を持つ。



【第3段階】

- ・一連の簡単なケーススタディによって、知識ベースをテストする
- ・プロトコル分析結果を基に、更に専門家と討議を重ねる

〔結果〕①CASはアプリケーション（インプット、プロセッシング、アウトプット）コントロールを調べる前に全般コントロールや監督性コントロールを先ず監査すると指摘している。

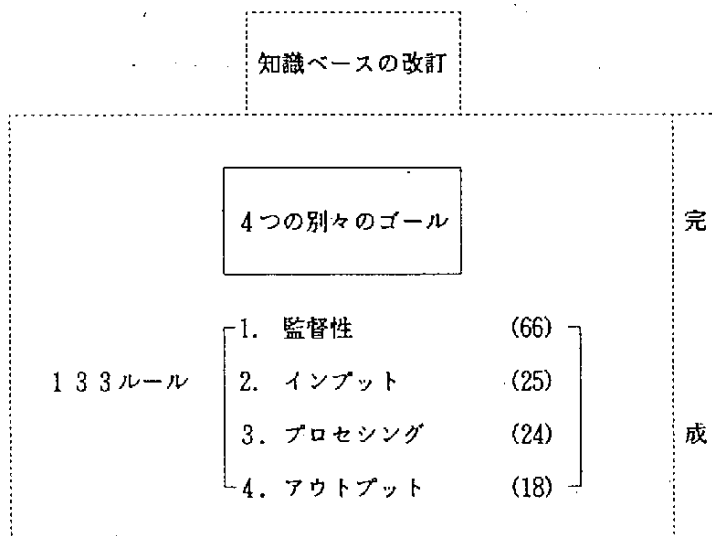
②専門家は監督性コントロールが十分でない限り、特定の会計アプリケーションに関するコントロールには信頼が置けないと指摘している。



- 4大ゴールの設定：
- | | |
|------------|--------------------------|
| 1. 監督性 | コントロールの信頼性 (Reliability) |
| 2. インプット | |
| 3. プロセッシング | |
| 4. アウトプット | |



- 〔実施内容〕
- ・我々が、適切なコントロール、構造そして専門用語を持っているかどうかを判断するルールについて、シニアCASと2日間にわたって詳細に討議した。
 - ・CASはルールに対する確信度を提供した。
 - ・(EDP-XPERT の知識ベースに対して) 提言された修正はすべて反映した。



その他のIf-Then ルール



EDP-XPERT の評価に関しては、「1.4.4 システム監査人の評価」を参照されたい。

② EXPERT AUDITOR

EXPERT AUDITORは、パソコンで動作し、情報システムの分析とレポーティングを行う。経験の浅い監査人でも複雑なハードウェア、ソフトウェアシステムが分析できるように、メニュードリブン形式で出来ている。

具体的には次のような機能を持っている。

- ・ 何を聞けばよいかについて説明し、詳細な監査計画を策定する。
- ・ 入力結果を分析する。
- ・ リスク領域を識別する。
- ・ 適切なコントロールを提示する。
- ・ 監査レポートを作成する。

このESは、各監査対象毎に詳細なシステム監査を実施できるように、次のようなバージョンがある。

- ・ Expert Auditor CICS
- ・ Expert Auditor IMS
- ・ Expert Auditor System/38
- ・ Expert Auditor System/36
- ・ Expert Auditor Data Center
- ・ Expert Auditor Disaster Recovery
- ・ Expert Auditor Microcomputer

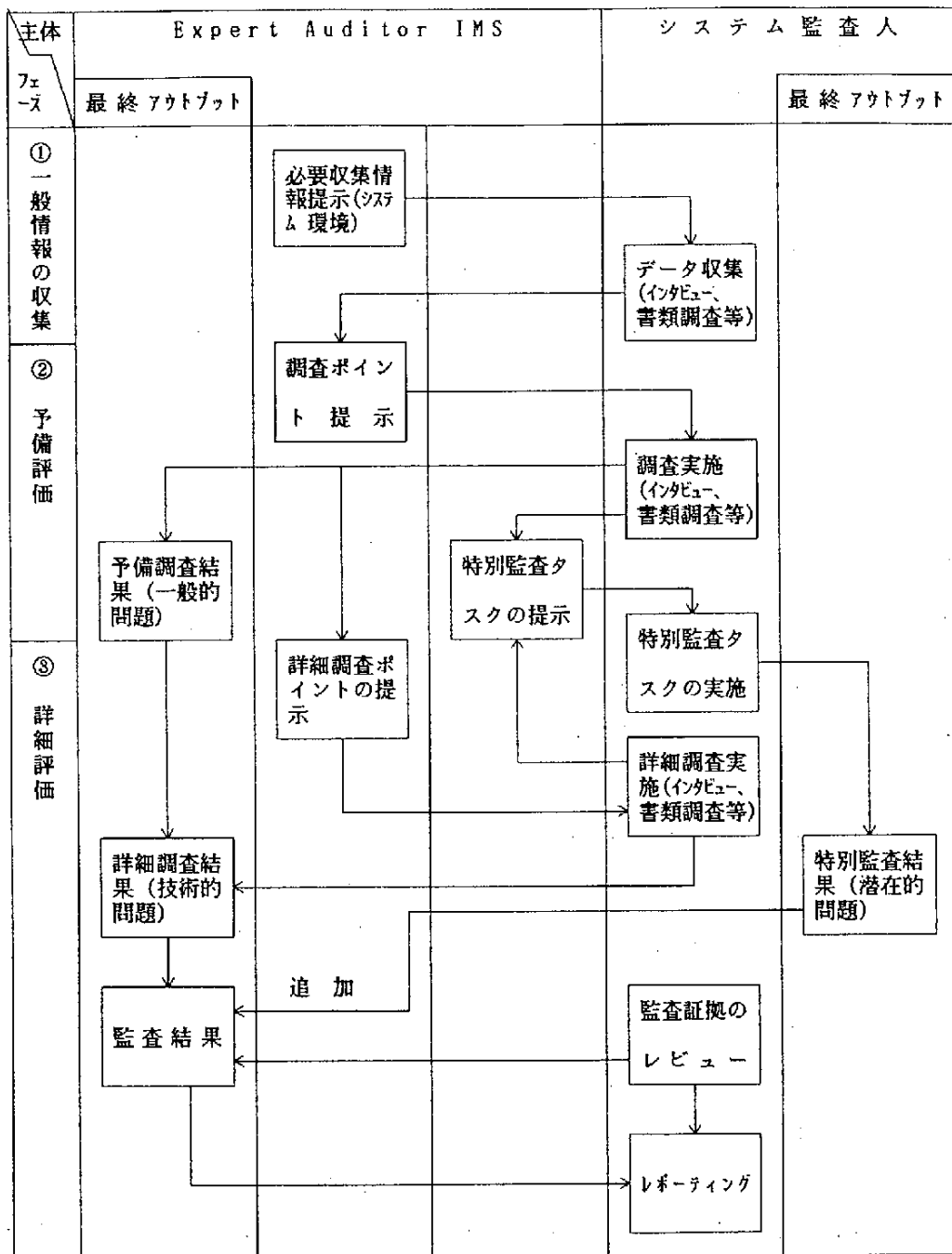
ここでは、Expert Auditor IMSについて説明する。

Expert Auditor IMSは、IMS環境で走るアプリケーションの監査を支援し、3つのフェーズから構成されている(図付1-1)。

- a. 一般情報の収集 (Collection of General Information)
- b. 予備評価 (Preliminary Evaluation)
- c. 詳細評価 (Follow-up Evaluation)

aは、システム環境やプログラム、手続き等に関する基礎的情報を収集するフェーズである。

bの予備評価は、調査結果を基に予備的な評価をするフェーズである。cでは、bの結果に基づき技術的サポートを行い、更に詳細な評価を行う。



図付1-1 Expert Auditor IMSの機能

これらのフェーズを通し、Expert Auditor IMSがサポートする機能を次に示す。

イ. 収集・調査すべき情報の提示

3つのフェーズを通し約360の質問項目が示される。第1フェーズでは3つにグループ分けされた約60の基本的質問、第2フェーズでは6つにグルーピングされた約150の質問、そして第3フェーズでは、同じく6つにグルーピングされた約150の質問が示される(図付1-2)。

ロ. 予備調査結果の提示

監査人が収集した情報に基づき、一般的な問題点を提示する。

ハ. 特別監査タスクの指示 (Special Audit Task Assignment)

Expert Auditor IMSの評価によってシステムの潜在的なウィークポイントが発見された場合、その解決に向けての監査タスクが示される。

監査人自らが実施しなければならない作業については以下の通りである。

Expert Auditor IMSは調査収集すべき情報を提示してくれるに留まり、インタビューの対象者決定も含めて調査実施は全て監査人に任される。また特別監査タスクもその指示のみサポートし、実施はExpert Auditor IMSの支援なしにすべて監査人が独自に行わなければならない。監査結果の妥当性の確認もまた基本的には監査人の作業である。

Expert Auditor IMSは、自らが分析した監査結果について、それに至る質問・回答一覧 (Finding Analysis Charts)を示してくれるが、その妥当性・網羅性の検証は、各部門のマネージャーやテクニカルサポートとともに監査人の確認を必要とする。統制や手続きに関するクリティカルな判断は監査人自らが行わねばならないが、Expert Auditor IMSは監査人に対し正しい方向を示すことができる。

×××××

New York

監査ID: ××

1988年×月×日 (金)

×××××

監査人: ×××

Expert

Auditor

IMS

監査ワーキングペーパー〜フェーズ3 F

IMSデータアクセスコントロール質問事項

- CS29 本番データセットとテストデータセットは同じ記憶媒体に保管されていますか。
(Y) Yes
(N) No
解答: _____
- CS60 どんなアクセスコントロールソフトウェアが使われていますか。
(A) IMS 内のセキュリティ・マクロ
(B) RACF等のソフトパッケージ
(C) アクセスコントロールソフトウェアは使用していない
解答: _____
- ES31 本番データに影響しうるプログラムの中で、プライベートユーザテスト用端末から本番データへのアクセスパスは存在しますか。
(Y) Yes
(N) No
解答: _____
- CF21 IMS データベースの本番ファイルは、テスト端末起動のアプリケーションやユーティリティ経由のバッチでアクセスできますか。
(Y) Yes
(N) No
解答: _____
- SG30 システムログレコードは、試みられた全てのセキュリティ侵犯を記録するように作られ、データは一定期間、保存されていますか。
(Y) Yes
(N) No
解答: _____
- SG40 システムログのデータセットには、アクセス対象が記録されていますか。
(Y) Yes
(N) No
解答: _____
- SG42 システムログのデータセットには、アクセス主体が記録されていますか。
(Y) Yes
(N) No
解答: _____
- SG44 システムログのデータセットには、ファイルが何時アクセスされたか記録されていますか。
(Y) Yes
(N) No
解答: _____

図付1-2 Expert Auditor IMSの質問項目の例

③ AUDITPLANNER

AUDITPLANNERでは、ソフトウェアシェルはEMYCINを使っている。このシステムの初期ルールは企業監査マニュアルと専門家である監査人とのインタビューから抽出された。実際に多くの監査環境においてAUDITPLANNERを使い、5日間討議を重ねてプロトタイプは改良された。次に知識ベースは5日間でカバーしきれなかった特定状況の使用のためにも改訂が続けられた。

このシステムの予備評価は、3名のシステム監査マネジャーと3名のシステム監査シニアが行った。その内5人はそれぞれ2つの企業に対して、1人は3つの企業に対してこのシステムを試行し（ $5 \times 2 + 3 = 13$ ケースの評価）で行った。これらの企業は、種々の製品の製造業である。各評価者ともAUDITPLANNERを試行させるために監査調書を詳細に準備し、その解答を従来の監査で使われている実際の重要性レベル（Materiality level）と比較した。

AUDITPLANNERが出力した13ケースの解答に対し、8ケースの解答はシステム監査人もESの重要性レベル判定が妥当であると評価した。AUDITPLANNERは「13ケース中12ケースは重要性レベルが低い」としていることから、これらの解答は企業の監査人よりも保守的と見做すことができる。更に改良が加えられれば、意思決定エイド(decision aid)やトレーニング用として使用可能との評価が出された。

④ ARROWSMITH-P

ARROWSMITH-Pは、ソフトウェア開発管理を目的としており、具体的には、ソフトウェア開発プロジェクトが計画通り進んでいるかどうかについて、ソースコードレベルの情報から判断する。試作では、a. ルールベースの「演繹」型システム、b. フレームベースの「推測」型システムの2種類を使って比較を行った。

ARROWSMITH-Pでは、ソフトウェア開発過程で集まる以下の9つのデータ(Metrics: メトリックス)を使う。ただし、今回の試作はNASA（米航空宇宙局）内の特定ソフトウェア環境のみにあてはまるものであり、特殊な限られたモデルである。実際プロジェクトごとに大きくメトリックスがばらつくようなら経験則適用は困難であり、今のところ適用範囲は広くないと考えられている。

使用するデータは以下の通りである。

- ・ソースコード1行当たりのCPUラン回数
- ・ソースコード1行当たりのCPU時間
- ・ソースコード1行当たりのプログラム変更量
- ・ソースコード1行当たりのプログラマ工数

- ・CPUラン1回当たりのCPU時間
- ・CPUラン1回当たりのプログラム変更量
- ・CPUラン1回当たりのプログラマ工数
- ・プログラム変更量1行当たりのCPU時間
- ・プログラム変更量1行当たりのプログラマ工数

これらのデータについて、「普通」、「普通より大」、「普通より小」の中から何れかを選択して入力する。推論結果は、33種類の「解釈」（「仕様は安定している」「仕様は安定していない」「生産性は高い」……等）の組み合わせで得られる。

実際のデータの解析結果とESの出力（解釈）とを比較した。実際のデータ解析結果には、イ．“断言できる”、ロ．“そう予測できる”の2種類の解答タイプが考えられる。aは、イ-45%、ロ-35%、bは、イ-33%、ロ-22%がそれぞれ一致した（表付1-3）。

表付1-3 ARROWSMITH-Pの実験結果

結果 \ システム	a. ルールベースの「演繹」型	b. フレームベースの「推測」型
断言出来る	45%	33%
そう予測できる	35%	22%

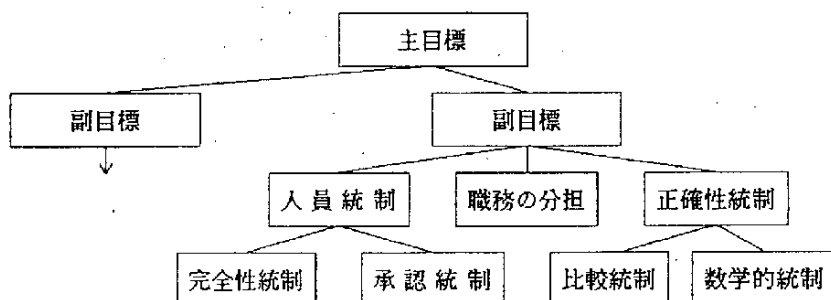
⑤ I. C. A.

I.C.A. (Internal-Control-Analyzer)は、歳入サイクルの内部統制評価ESのプロトタイプで、監査人を内部統制の調査分析、評価において支援する。このシステムは、目標が階層構造 (Hierarchical Goal Structure) をなしている (図付1-3)。主目標 (Major Goal、すなわち会計統制の質) の下に2つの副目標 (Subgoal)が、さらにその下のレベルには人員統制、職務の分担、正確性統制等のブレイクダウンされた副目標がある。

完全性統制と承認統制 (Authorization Control)は上のレベルの人員統制に帰属し、比較統制と数学的統制 (Mathematical Control) は上のレベルの正確性統制と関連している。

シェルはAUDITPLANNERと同じEMYCINを使っている。ESの構築では、先ず教科書的知識を搭載して実際の監査問題に適用し、その出力結果を専門家に評価してもらい、改良を加えた。本ESは、まだ開発の初期段階にあるので、ESに対する検証過程は、比較的非形式的であった。監査人は、実際の監査ケースに対してESが行った判定を評価した。その監査人は、ESが会

計統制に関して、監査調査データを使用し推論して評価した結論は、妥当であると感じたが、ESが決定した確信度に関しては、かなり違っているように感じた。確信度に関して、評価者である監査人となぜ一致しないかに関して、2つの原因が考えられる。第1の原因は、本監査ケースでは現在のプロトタイプ版知識ベースには蓄積されていない環境因子が含まれていたことがあげられる。第2の原因は、この監査人はこれらの判断を確信度を使用せず、「統制が信頼できるか、あるいは信頼できないか」と言う白黒判定 (dichotomous decision) で行ったことにある。



図付1-3 I.C.A.での階層型目標構造

⑥ IST/RAMP

a. IST/RAMPの機能

IST/RAMP(International Security Technology/Risk Analysis and Management Program)は、5タイプに分類した予想損失、バックアップ形態に応じて変化するコスト、データの機密保護度の7つを評価するのに必要なデータを格納、編集、保管するためのモジュールを備えている。この7つの評価対象は、以下の通りである。

i) 遅延損失

データプロセッシングの不測の中断、遅延によりEDP使用者が被る損害

ii) 物理的損失

資産の物理的破壊、破損、紛失等による損失

iii) EDP経由の詐欺

EDPシステムの不正操作から起こる損失

iv) 承認を得ていないEDPデータの開示

承認を得ていないマスターファイル等の開示によりデータ保有者が被る損害

v) 物理的盗用

物理的資産の物理的盗用から生じる損失

vi) マスターファイルのバックアップ

ファイルバックアップの予想費用を3通りのオフサイトバックアップ戦略毎に算出

vii) マスターファイルの機密保護度

マスターファイルの相対的な機密度、相対的なファイルの重要性、処理上の必要性和相対的頻度

監査人は、IST/RAMPが出力するこれらの評価を組合せ、リスクを管理する。例えば、費用見積りを行い、災害復旧計画をたてれば遅延損失を軽減できる。すなわち機能停止という脅威は常に潜在するが、災害復旧計画があればユーザーが被る処理の遅延を極力押えられる。よって分析者は、IST/RAMPにより損失軽減の見積りを簡単に行い、形態に応じたバックアップ計画を作成したり、損失軽減を計画実施のための必要経費と比較することが出来るようになる。同様に物理的損失評価ルーチンは物理的セキュリティの測定、その他の費用見積りを支援する。

b. IST/RAMPの構造

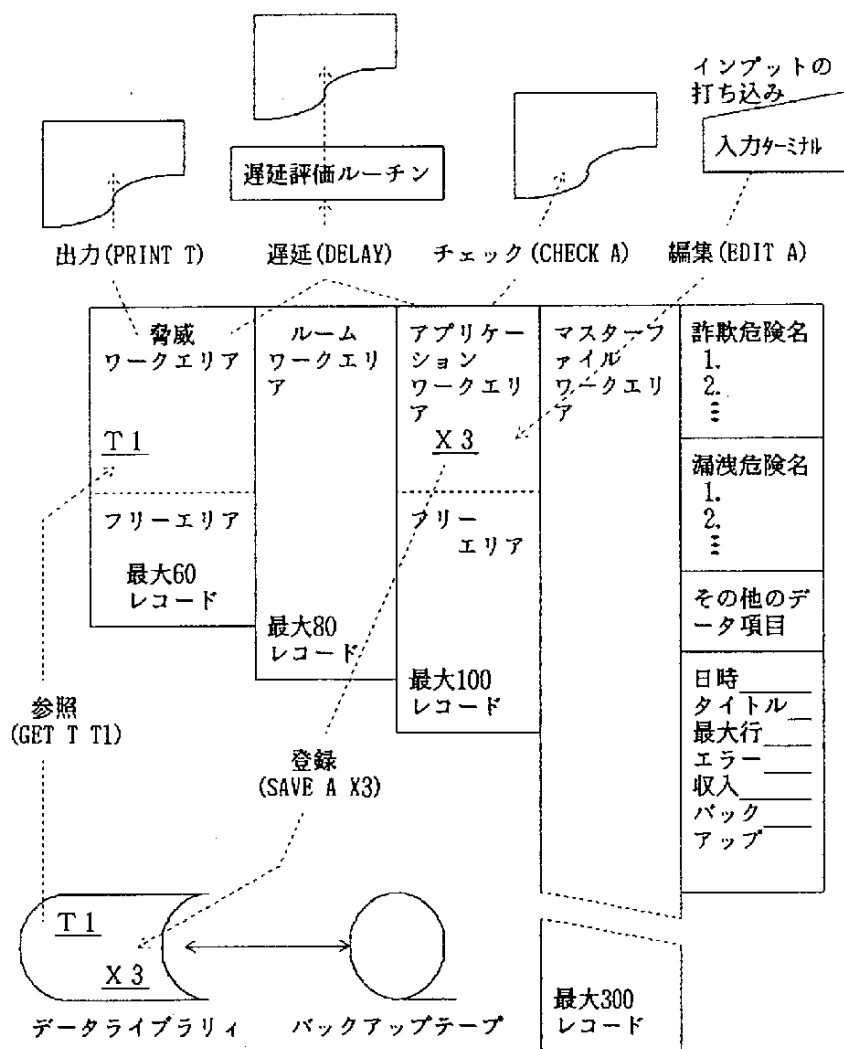
IST/RAMPは、3つの部分から構成されている(図付1-4参照)。

- ① データベースを備えるデータライブラリ
- ② 分析に必要なデータを組み立てる場所であるワークスペース
- ③ 基本的な7つの評価機能を遂行するルーチンセット

IST/RAMPは、分析者が本システムを使用して、データライブラリに格納したデータベースに対して、作成(create)、変更(alter)、保存(save)、再呼び出し(recall)するために必要なコマンドを具備している。(データライブラリは、直接アクセス記憶装置上に常駐している。分析者は、これらのコマンドを使用して、自分のデータベースを変更し、新しいバージョンを作成して、再度データライブラリに格納する。)典型的には、分析者は、各リスク環境に関して色々と異なる仮定を設定して決めた各タイプ毎に、たくさんのデータベースを持つことになる。加えて、IST/RAMPルーチンは、そのデータベースをプリントアウトしたり、分析者がデータの妥当性をチェックするのを支援する。IST/RAMPは、内部ワークスペースに記憶されているデータを対象にして操作を施す。分析者は、分析作業を行うには先ずデータライブラリから必要なデータベースをワークスペースに移し、ワークスペース上のデータに対して必要な変更を行い、その後IST/RAMPを実行させる。IST/RAMPの7つの評価ル

ーチンは、各々ワークスペース上のデータを入力データとして使用する。この際に、分析者は、脅威や資産を表すデータに対して追加、削除、そして変更することが出来る。IST/RAMPは、分析者に対して、定形フォーマットを使用することを強制しない。必要なデータがワークスペース上に無い場合には、分析者は、編集コマンドを使用して、新しいデータを直接ワークスペースに入力することや、必要ならワークスペースの内容を新しいデータベースとしてセーブすることが出来る。

表付1-4に、具体的な機能説明を示す。



図付1-4 IST/RAMPの構造図

表付 1-4 IST/RMPのコマンド例

コマンド	機 能
GET T TX01	データベースTX01を取り出し脅威ワークスペースに移す。 (詳述) : T1と呼ばれる前から作られた脅威のデータベースをライブラリから脅威のワークスペースに移す。→ IST/RAMPが命令を確認 → ワークスペースにレコードがいくつあり、スペースがどれほど残っているかを知らせる (もしT1に12のレコードがあれば、60-12 = 48のフリーのレコードスペースがあることになる) → 分析者は同じ様にGET T T2を行うと → IST/RAMPはこれをすでにワークスペースにある脅威に加える。 このようにして2つ以上のデータベースがワークスペース内で結合されプロセスも並行して行われる。
EDIT A	アプリケーションワークスペースをエディットする。 (詳述) : この命令はキーボードからのインプットがアプリケーション・ワークスペースに入るべきデータである事を示す。例えば、分析者は、実行中のアプリケーションに関するデータを入力して新しいアプリケーション・データベースを作る事ができる。
STOP	アプリケーションワークスペースへのデータ入力を終了させる。
SAVE A X3	アプリケーションワークスペース内のアクティブレコードは、後で使用するのに備えて、X3と呼ばれるアプリケーションデータベースとしてセーブされる。
PRINT or CHECK	ルームワークスペースの内容をプリントする。 ワークスペースのレコードを更新したり文書化したりする。
DELAY	脅威ワークスペースとアプリケーションワークスペースにある情報をインプットとして使い、IST/RAMPに処理遅延からの予想損失を計算、出力 (プリントアウト) させる。

c. IST/RAMPのリスク定量化手法

リスク定量化の方法としてマーチン法(The Martin Method:James Martin が1973年出版した「コンピュータシステムの安全性、正確性、機密性」の本の中で発表)がある。これはまず脅威を分類し、脅威の頻度と脅威による被害のレーティング値 (PとD) を一覧表から求め、これによって予想損失を計算する (図付1-5)。

脅威の頻出度	レーティング値 : P	脅威の被害	レーティング値 : D
不可能	0	\$1	0
1/400年	1	\$10	1
1/40 年	2	\$100	2
1/4 年	3	\$1,000	3
1/100日	4	\$10,000	4
1/10 日	5	\$100,000	5
1/ 日	6	\$1,000,000	6
10/ 日	7	\$10,000,000	7

$$\text{公式: 予想損失} = 10^{(P+D-3)} / 4 \text{ (ドル/年)}$$

図付 1-5 マーチン法での脅威と被害のレーティング値

IST/RAMPはこれに修正要因を勘案する。下式でj 番目の資産に影響する想定されたタイプ
のi 番目の脅威から予想損失を次のように捉える。

$$\text{予想損失}_{ij} = \text{脅威発生レート}_i \times \text{資産損失可能性}_j \times \text{修正要因}_{ij}$$

修正要因(Modifying Factor)は、i 番目の脅威がすべての資産に一律に影響を及ぼさない
ことを反映したパラメータであり、i 番目の脅威と j 番目の資産に固有の重みを持つもので
ある。その点でマーチン法より現実にはし易い。

⑦ BDSS

BDSS(Bayesian Decision Support System)は、パーソナルコンピュータ・ベースのリスク定
量分析ツールである。

BDSSには、次の7つの機能がある。

i) リスク分析対象の情報処理環境のモデル化

テキスト編集機能を使用して、次の項目を定義する。

- ・対象とする問題/背景
- ・目的
- ・目標
- ・対象範囲及び制約条件
- ・責任体制
- ・リスク受入基準

ii) 資産や在庫の価値評価を支援する。ここでの特徴は以下の2つである。

- a. 従来、価値評価が難しかった“データ”に対し、次の2つの価値評価基準を示し、
計量化を支援すること。
 - ・ 代替コスト
 - ・ 処理が不可能になった場合ビジネスに与えるインパクト（ビジネス機能の停止
や信用への打撃等）
- b. それぞれの評価価値に信頼性の度合を示すこと。

iii) ユーザーからのインプットデータに基づき、“VULNERABILITY-MAP”を示す。

“VULNERABILITY-MAP”とは、物理的な環境や管理・監督の状況、ハードウェア/ソフト
ウェアの状況や保有データへのアクセス状況等に関する全体像である。そこでは同時に、

リスクの存在ポイントを示す。

iv) “RISK MODEL”を示す。

損失価値分布、脅威出現分布、エクスポージャ要因 (EXPOSURE FACTOR) 分布などの対象環境に対して、各リスクを識別する。

v) “RISK MODEL”に基づき、各リスクを詳細に分析する。

不確定要素をも勘案しうる統計的アルゴリズム (BAYESIAN ALGORITHMS) を使用し、アウトプットを “RISK CURVE” 上に示す。

vi) 安全対策を提言する。

ここでは導入及び保守のコストを計算し、かつ対策導入前と導入後では “RISK CURVE” がどのように変化するかを明らかにする。

vii) マネージメントをターゲットにしたレポート (サマリーと詳細)、及びBDSS使用結果の詳細レポートを出力する。

また、BDSSの特徴を列挙すると次のようになる。

- ・現実の不確定な状況を反映し、リスクの影響要因を多変量的に捉える。リスク出現頻度・予想損失額とともに、ユーザーのライン (製造業なら生産ライン) や社会的使命に対するインパクトという観点から、リスクを識別評価する。
- ・マネージメントサイドに立ったレポート形式を採り “RISK CURVE” 等により、特定期間の潜在的脅威や対策効果が視覚的に把握できる。
- ・BDSSはコストコンサーンであり、時間と労力を最大 “75～90%” 削減する事例があると述べられている。
- ・リスク収集ノウハウに関する説明を含むDBを提供する。

このBDSSは、IBM PC/XT・AT、及びその互換機上で動作する。また、メニュードリブン方式である。

開発はOzier, Perry & Associatesが、Pickard, Lowe and Garrick, Inc. の協力のもとに行った。米国の大手銀行2行、及び公益企業に対し導入されたと報告されている。

以上、EDP監査領域へのES導入事例を俯瞰した。ここで内容を要約すれば表付1-5のようになる。

表付1-5 EDP監査領域のエキスパートシステムの一覧表

項番	システム名	内 容	成 果 と 課 題	ソフトウェア・シェル	開発者
①	EDP-XPRT	高性能コンピュータシステムの統制の信頼性を評価	オンラインとデータベースのコントロールに関するルールを含む知識ベースの拡大を予定	AL/X	M.Messier V.Hansen
②	EXPERT AUDITOR	データセンター、Disasterリカバリ、メインフレーム全般、パソコン全般等の7領域のシステム監査を支援	質問表を提示し、回答から診断、または詳細調査指示を行う	AL/X	Boden Associates
③	AUDITPLANNER	監査計画局面での重要性レベル判定の支援	正答率は約70%	EMYCIN	Steinbart
④	ARROWSMITH-P	ソフトウェア開発管理の支援	特定環境においてのみ試行	——	Maryland Univ.
⑤	I.C.A.	歳入サイクルの内部統制を評価する	監査調査データを使用した推論による評価結果が妥当	EMYCIN	Gal
⑥	IST/RAMP	EDPリスクの定量、定性分析支援	導入実績は、米国、カナダを中心に約90件	フォートランとアセンブラで開発	Inter- national Security Technology
⑦	BOSS	統計的手法を用い、リスクを多変量的に分析	マネージメントサイドの視覚的なレポート作成	——	Ozier, Per- ry&Associ- ates およ び Pickard, Lowe&Gar- rick, Inc.

1.4.3 システム監査関連領域のエキスパートシステム

(1) 会計監査域

会計監査域において、ESが待望されている理由を以下に示す。

- ・ 監査環境が複雑になり、従来の方式だけでは効率的でない。
- ・ 競争的な入札や営業に関する職業倫理基準が変わってきた結果、宣伝が企業間に更に競争を引き起こし、会計監査人は低廉な料金と効果的な監査を求められるようになった。
- ・ 開業している専門家が組織に対する職員のコンセンサスを確立したいと考えている。

・これまでの研究によりESが監査人の意思決定プロセスを調べる上で有効なツールであると判明した。

大手会計事務所では、すでに在庫、繰延税金の問題、貸付金について問題のある顧客、売掛金等に関して、会計監査を支援するESを使用している。例えば、監査人は銀行の貸付金の情報を（銀行のデータベースから）ダウンロードして、支払伝票、財政状態の集約、貸付金の日程、問題が起きそうな貸付金の項目等を調べる監査ソフトウェアと合わせてESを使用している。課税所得と繰延税金を自動的に計算し、クライアントが行った計算と比べるESもある。将来的には、ESにより監査テスト手順と、財政状態に関する報告をダイレクトにアウトプットすることが可能ではないかと期待されている。

監査という特質上、システム監査のESは会計監査のESと機能面で重なることが多く、同じように扱われる場合が多い。会計監査のみならず、その他のシステム監査周辺領域とも情報システムを扱うという点で重なることが多い。システム監査ESは、会計監査のESとかなり密接な関連を持ちながら開発されてきているため、会計監査域に注目する意義は大きい。

米国の大手会計事務所は、元来は会計監査用として開発したESをシステム監査目的に沿うように機能向上させ、本格的にESを用いたシステム監査を実施しようと試みている。

以下に、8つの事例を紹介する。

① ARISC

ARISC(Auditor Response to Identified Systems Controls)は、Meservy、Bailey、Johnsonが開発した会計監査のESである。このESは、購入、買掛金、現金支払いサイクルにおける監査人の内部統制評価過程をモデル化している。

ARISC構築の目的は、問題を解決するシステムの構築というより、むしろ、認知心理学の問題（例：専門家の意思決定過程の理解とそのモデル化）を解決することにある。このモデルは、プロトコル分析・インタビュー・教科書知識を使用して開発された。

プロトタイプ版のARISCの実行結果を、知識を提供してもらった専門家に確認してもらい、ARISCに蓄積されているルールや推論過程を検査した。ARISCは、Galenと呼ばれるES構築ツールを使用して構築され、約300のIf-Then文を備えている。ARISCに内蔵されている計算モデルの検証は、モデルの推論過程や出力結果と、知識を提供した専門家と、監査マネージャ3人の結果とを比較、評価して行った。3つの例題に対する全ての解答（ARISCの解答、専門家の解答、3人の監査マネージャそれぞれの解答）を、誰が解答したかを伏せて、別の3人の監査人によって完全性、有効性、一致性の観点で評価し、ランク付けをしてもらった。

(これは、ピアレビューと呼ばれる)。この場合、解答の分析は、次の4点に分けて実施してもらった。

- ・仮説の生成 (hypothesis generation)
- ・問題解決プロセス (problem solving process)
- ・キューの使い方 (cue usage)
- ・推論過程 (lines of reasoning)

この結果、完全性、有効性、一致性では実際の監査人と同様と評価された。ある特定の領域では少なくともARISC は、専門の会計監査人になりに近いと言える。

② Auditor

Auditor は、会計監査人が顧客の負債許容を評価（ローン返済能力等を評価）する際に支援するESである。開発者は、DunganとChandlerで、ES構築シェルAL/Xを使用し、次の3つの段階を経て作成された。

- i) システムゴール、ルール、ルール重み付けの初期モデリング
- ii) システムを会計監査の専門家に使わせシステムを改良した（この時点で知識ベースに約25の初期ルールが備わった）。
- iii) システムの妥当性テスト

開発後の評価では、Auditor の出力結果と専門家の意見の一致は約90%となった。

ルールベース化が進められた会計監査分野でのESとしては、Auditor は最も初期のものに入る。

③ CFILE

CFILE は、監査人が銀行の貸し倒れ準備金の評価するのを支援する目的で、Peat, Marwick, Mitchell & Co. によって開発された。現バージョンのCFILE では、要求払い／年内払いの貸付金と、あるいは銀行預金や有価証券等の担保付けがある場合／無い場合の、両方の貸付金に限定されている。またその能力は現金フロー分析の実行と統合に限られている。監査済の情報なら2年分、未監査の財務情報なら3年分の資料を必要とする。ルールベースシステムシェルのINSIGHT2を使い、マイクロコンピュータ上で動作する。

CFILE は、①借用者の現在の財政状態、②借用者の副抵当物の信用力、③保障性の各サブゴールに対応した個々のモジュールからなるモジュール構造となっている。システムには、説明能力があり、限界分析を行い、決定のための文書作成を行う。

評価したところ、引当金を必要としない貸付金に関し、専門家の意見との一致は約90%と

報告されているが、引当てるべきローンの割合（パーセント）を出すような、より高度なローン分析では機能向上の余地がある。

④ TAX ADVISOR

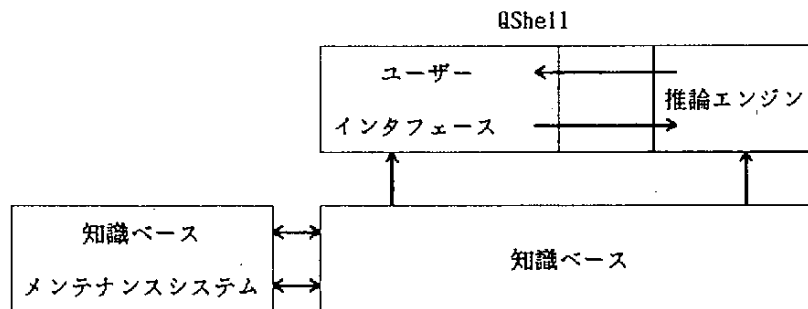
税金や財産計画についてのコンサルテーションを行う相談型のESである。

⑤ EDP-AUDITOR

EDP-AUDITOR は、高性能システムの情報処理面の全活動を記録し、それを内部監査の重点対象とする場合に有効なESである。

⑥ Expert TAX

ExpertTAXは、Coopers & Lybrand 社が開発した法人税税務を支援するESである。図付1-6に、ExpertTAXの構成要素を示す。Qshellは、Gold Hill Common LISP でプログラミングされ、前向き推論を行う。知識ベースはフレーム、ルール、事実を含み、知識ベース・メンテナンスシステムを持つユーザーインタフェースは使い易く設計されている。システムは数百のフレームと、1,000以上のルールを操作するように設計された。開発には20人の専門家が1,000時間以上かけたと言われており、相談型のESである。



図付1-6 ExpertTAXの構成要素

⑦ TICOM

TICOM(The Internal Control Model) は、ミネソタ大学で開発された内部統制システムを評価する（デザイン、モデリング、評価を行う）ESである。TICOM は、相互に関連した2つのセットのプログラム、すなわち内部統制記述言語(ICDL:Internal Control Descriptive Language) コンパイラと質問プロセッサとから成る。これらは内部統制モデルの開発と内部統制システムの評価という2つの分析局面を支援する。会計監査人はTICOM で内部統制シス

テムを記述する。次にTICOM は監査人に質問を行い、その答によりモデルの吟味とテストを行う。TICOM は、複雑な情報システムと内部統制のシステムを把握できるが、内部統制システムの全体評価用のガイドラインがなく、また何が強点・弱点の統制を構成するのかの一般知識は備わっていない。

⑧ T I C O M III

TICOM IIIは、クライアントの会計情報システムや組織モデルにおいて、コントロールの強弱度を評価し、コントロールの強弱度の状態と実地的な改善勧告案を出力するように機能向上されたもので、次の3つのESの集合体である。

- i) モデラー (Modeler) : 入力されたICDLステートメントが表現したモデルを組み立てる。
- ii) プロポーザル (Proposer) : モデルの可能性分析後オブジェクトを作る。
- iii) ヴェリファイア (verifier) : 過去のシナリオやコントロールオブジェクティブの知識を参考して、ii)のオブジェクトの妥当性を評価する。

(2) ゴーイングコンサーン域 (Going-Concern)

ビッグス (Biggs)は、監査人が企業のゴーイングコンサーンを判断する際の診断推論プロセスを調査した。監査人はその企業がどれだけ長く存続可能か、すなわちこれから先も長らく経済活動できる企業体質を備えているかどうかを、財務情報等を使用して、様々な角度から調査し評価するが、ビッグスはそれを対象としてES化を検討した。

第1フェーズでは、監査人が行うゴーイングコンサーン判断の際の診断推論プロセスの正確性に関する調査を行った。実際に開業している監査人のところにゴーイングコンサーン問題の分析を依頼しにきた企業がケースとなった。このフェーズではさらに診断推論プロセスの知識ベース化を試みた。

第2フェーズでは、第1フェーズにプロトコル分析を加え、ESの出力結果を監査人に評価させた。プロトタイプを開発した後は、他の新しいケースに応用しながら知識ベースを改訂していく方針を取っている。

以下では、ビッグスらが開発したGC-X及びAODの二つの事例についてより具体的に説明する。

① G C - X

GC-Xは、ゴーイングコンサーンの判定を行うESとしてBiggs とSelfridge によって1986年に開発されたプロトタイプである。これはゴーイングコンサーンを判定する監査の専門家がモデルを設定し、ESのおおまかなフレームワークを提供するものである。GC-Xには、

資料と2人の専門家とのインタビューを基にして以下の6つのタイプの知識を格納した。

- i) 資産運用状況の調査について
- ii) ゴーイングコンサーンの判定をするための資産運用に関する着眼点と、その際の判定に対する説明の仕方
- iii) 企業の業務とその環境について
- iv) ゴーイングコンサーンの判断の際の環境知識(environmental knowledge)
- v) 業務と資産運用の環境要因との繋がりについて
- vi) 管理計画を評価するための、業務、環境、財務に関する知識

GC-Xは、LISPでプログラミングされ、VAX11/780上で動作する。このシステムは、3つのモジュールを備えている。1つ目は財務知識を扱うルールベースシステム、2つ目は業務と環境知識を扱うイベントチェーン・ネットワークシステム、そして3つ目は以上の2つのモジュールをハンドリングする計画評価モジュールである。またデータベースも3つ備えている。1つ目は財務諸表事実とシステムによって結論づけられた事実、2つ目は財務と判定に関する規則、そして3つ目は業務と環境での出来事のデータベースである。

GC-Xは、企業がゴーイングコンサーンの体質であるかどうかを評価するための階層化ゴール構造を持ち、その内の主ゴールと副ゴール上で検索を行い、処理を進める。GC-Xは、データベース内のデータを使用して直接判定を行うか、ユーザーと対話的に情報を補完しながら作動する。また、簡単な自然言語インタフェースとその推論過程を説明する能力を持っている。

② AOD

AOD(Audit Opinion Decision)は、ゴーイングコンサーンについて監査人の意思決定を支援する目的で、DillardとMutchlerにより開発された。

知識ベースは専門の監査人から集められた信頼しうる意見と口述のプロトコル分析によって作成された。知識ベースシステムは、XINFOシステムのメニューベース・フレームワークを使用し、445のフレームを備えている。ユーザーは、順番に出てくるフレームによって意思決定プロセスをたどっていく。AODは意思決定の中間段階を作成し、それを専門家がするようなやり方で順番にユーザーに提示するが、これはESの開発時に要求される画面対話の支援機能と同等である。AODは、パソコン上で機能するように改良されており、さらに、推論能力を組み込んだAOD-IIの開発が行われている。

(3) セキュリティ対策域

セキュリティ対策を実施する領域には、ES以外の人工知能も多く導入されている。すなわち、パターン認識技術を応用した入退室コントロールにおける人物の識別やマシンルーム内の機器の操作等に対してである。

ESの事例として、以下ではTRW社のDISCOVERYについて紹介する。分析により不正をチェックするという点では、Sytek社も同様のエキスパートシステムを米国海軍との契約下で開発している。Sytek社のシステムは、セキュリティ担当者がセキュリティ侵犯の発見を支援するものであるが、両者とも監査証跡をパターンマッチングの方法を用いて自動分析するものである。

① DISCOVERY

米国西海岸に本社を置くTRW社は、近年情報提供サービスをも実施している。登録した顧客は、TRW社のデータベースへ公衆回線を通してアクセスでき、アクセスコントロールとして顧客番号とパスワードを使用している。顧客は、西海岸を中心に米国全土に散在し、個人、法人等さまざまである。DBへの情報照会件数は、1日に約4,000件であるが、格納情報タイプの拡大により今後増加が見込まれる。

顧客に対する使用料金の請求は月次に行われ、その際使用状況明細も添付される。万一第三者により不正使用された場合には、使用状況明細を見た顧客によりTRW社へ知らされ、所定の措置がとられる。しかし、月次ベースでかつ顧客の事後的チェックに依存するシステムでは、最近の同業種におけるハッカー動向から見て十分とは言えない。また当社の苦情対策チーム等の関連担当者には、アクセス状況から不正か否かの識別をある程度行えるだけのノウハウがすでに蓄積されている。ただし、アクセス状況を担当者がモニターすることは日々の件数、作業負荷から現実的方法とは言えない。そのため、DBにアクセスしてくる照会トランザクションには、特定のパターンが見られることに着目し、その通常パターンからの乖離によって不正か否かの識別を試みるセキュリティコントロールのES“DISCOVERY”が開発された。

a. DISCOVERY の機能

DISCOVERYの最終的な目的は、不正アクセスの防止である。しかし、現段階の機能は日々のバッチ処理結果に対する判断に基づく不正アクセス状況レポートの出力であり、早期発見による被害の極小化を狙っている。

処理は、通常パターンの作成、各アクセスの判断、不正アクセス状況レポート編集から

成る。

i) 通常パターンの作成

前日までのある期間のトランザクションから、特定項目ごとのアクセスパターンを作成する。パターンは、統計処理により求められたものに担当者のノウハウを加味し作成する。項目は、不正と正当なものの差が出やすいと一般に考えられているものより、はるかに広いレンジで集められている。それには、アクセス時間やパスワードの誤入力回数、地域コード、配偶者のイニシャル、前住所の地域コード等が含まれている。

作成されたパターンには次の3つのタイプがある。

- ・ LOCAL PATTERN
- ・ GLOBAL PATTERN
- ・ INDUSTRY PATTERN

ii) 各アクセスの判断

当日の照会トランザクション全てに対し不正か否かの判断を行う。推論エンジンによる検索は、通常パターンが格納されているパターンファイルと、不正アクセスに関する事実やメタ知識が格納されている知識ベースの間をインタラクティブに行う。疑わしいアクセスに対しては、疑わしさの確信度付加とその理由説明を行う。パスワードの誤入力に関して見ると、異なる打ち間違いと、うっかり勘違いして間違えた場合、そしてデータベースハックのようにハッカーが試みている場合とでは相違が見られるはずである。また、通常短時間で効率よく情報検索している顧客が数時間もセッションを続けていた場合は、例え不正アクセスでないとしても、操作手続きの誤りや機器の故障が考えられる。よって乖離検出が顧客サービス向上に貢献するという意味で共通している場合は、不正とそうでない場合との区別をあえて厳密に行っていない。

iii) 不正アクセス状況レポート編集

疑わしさが一定レベル以上のトランザクションはこのレポートに編集出力する。編集項目には顧客番号とアクセス状況、疑わしさの確信度、理由、連絡先等があり、セキュリティ担当チームが次のアクションを行い易いように配慮されている。このレポートに担当チームのコメントを付し、原則として月次で監査部門用、マネジメント用にまとめて提出する

b. DISCOVERY の構築

構築の検討は、不正なアクセスの防止をどのような形でシステム化していくかという点

から始められた。個々の不正アクセスを防止するためには、不正アクセスの試みを識別し、その端末からのセッションを拒絶することが考えられる。また、抑止するためには犯罪を立証するだけの不正アクセス者とアクセス自体に関する情報が確保され得ることが認識されなければならない。しかし、通信機器等、ハードウェア面での制約や、アクセス時点の識別の可能性、対策行動自体のリスク等、様々な問題があった。フィージビリティスタディによって、DISCOVERY の骨格が形作られるまでには、3 人のアナリストが約 1 年を要した。どの程度機能を持たせ得るかは、アクセスが不正であるか否かの判断の正確性、言い換えればチェックの網羅性による。すなわち、出力に基づく行動のリスクや抑止効果はすべてこの点に依存している。そのため、関係者からの意見聴取や事例分析を行い、個々のアクセスに関して現段階で入手可能なデータ項目の組み合わせから不正を識別するルール作成を実施した。また、統計処理により、パターンの浮きぼりを図った。それらの情報を基に、IBM/PC の XT (60MB RAM) 上で C 言語を用いて開発した。プロジェクトを統括したウィリアム・T. ターナーによると、プロトタイプ開発は、内部監査部門のマネジャー 1 名、アナリスト 2 名、SE 3 名、サポート 2 名で約 3 ヶ月を要した。

c. DISCOVERY の成果

ウィリアム・T. ターナーは、DISCOVERY 導入期間中の具体的な成果については公表していないが、かなりの不正アクセスがチェックされ、それによる抑止効果も期待できると語っている。TRW 社では今後知識ベースの拡充に加え、システム監査に対する配慮をしていく予定である。ソフトウェア面、手続き管理面のシステムセキュリティが十分に機能しているか否かを評価するために、DISCOVERY は貴重な監査証拠を提供する。しかし、現段階ではシステム監査人の使い勝手等の要求に十分応えていない。今後、機能を向上し、システム監査用知識ベースも付け加えていくことが考えられている。

(4) リスク分析域

リスク管理は、リスク分析、リスク軽減分析、マネジメント判断、リスク軽減対策、セーフガード実施、チェック及び監査の作業を含む。マネジメントは、実効的な経営戦略遂行の背後を固める意味で、リスクをできるだけ網羅的にかつ正確に把握する必要があるが、このことは米国企業においては既に当然であると考えられている。リスク分析用の広義の ES は、複数製品化されている。それらの持つ機能は、定性的な面と定量的な面の 2 つに分けられる。

(a) 定性的な面の支援

該当領域におけるリスク項目の列挙によるリスク識別作業の支援で、例えば画面对話形式

で「障害」を指定すると、「ハードディスク障害」、「回線障害」…等が羅列される。

(b) 定量的な面の支援

既に格納されているデータと画面からの入力データとから、単純な算式を使用してリスクによる損害額の算出を行う。例えば「過去3ヵ月に起きたディスク障害は？」という質問に答えていくと、来年1年間に予想される損害額が表示される。

リスク分析作業自体に確立された方法論を持たない現状で、各ツールはタスク内の基礎的作業領域を代替しているレベルにある。

表付1-6は、リスク分析域におけるES導入事例に関する特性の一覧表である。

表付1-6 リスク分析ツールの特性

特 性 ツール名		分析タイプ		システム監査 領域支援	使用言語	使用環境		開発会社
		定量	定性			メインフレーム	パソコン	
①	IST/RAMP (CRITICALC)	○	○	○	FORTRAN アセンブラ	○	△	Inter- national Security Technology
②	BDSS	○	○	○	—	○	○	OPA & PLG Associated
③	T.R.A.M. (QUIKRISK)	○		一般 リスク	C		○	Basic Data System
④	RISKA	○		一般 リスク	CBasic、 パスカル		○	Chesapeake Computer Group
⑤	CAPARS	○		△	dBASE III		○	The Conco- rde Group
⑥	RISK PAC	○	○	△	—		○	Profile Analysis
⑦	RA/SYS	○		△	—		○	Nander Brown
⑧	RISK CALC	○		○	—		○	Parkhurst- Spence

注) ⑤～⑧は、ES応用のツールではない。

T.R.A.M.は、The Risk Analysis Machine の略である。

1.4.4 システム監査人の評価

EDP-XPERTの開発後、ハンセンとメシアは、システム監査人に対してアンケート調査を行った〔参考文献2〕。これには米国大手監査事務所に勤めるCASTレーニング段階の17名（マネジャーが2人、シニアが9人、スーパーバイザが6人）が参加した。彼らは1年から8年間のシステム監査経験を持ち、その平均は3.3年であった。1名を除く全員が公認会計士の資格を持ち、内2名は会計学修士であった。彼らは大学でコンピュータ単位を修得し、その平均は2.2年であった。

表付1-7から表付1-10に、アンケート調査の結果を示す。

表付1-7 システム監査へのコンピュータの導入に関心のある監査人の数

システム監査の次の項目においてコンピュータの使用が適していると思われるのはどれですか。該当するものはすべて印をつけて下さい。	
	(合計17名)
(1) 監査の書類準備（例、従来の書類コピーの代用）	17
(2) 顧客ファイルからのトランザクションの抽出	17
(3) 顧客ファイルにあるトランザクションのテスト	15
(4) 顧客の内部統制システムのモニタリング （例、組込まれた監査モジュール）	15
(5) 監査人の判断を支援するコンピュータベースの意思決定支援エイド （エキスパートシステム）	16
(6) 監査人の判断に代わりコンピュータベースの意思決定支援を全面的に導入する	1
上の項目(5)に賛成の方に質問します。コンピュータによる意思決定支援エイド（エキスパートシステム）の適用が、以下のどの領域において適切だと思いますか。	
・ 固有リスクの評価	10
・ 買掛金の監査	7
・ 統計的サンプリング	3
・ 分析的レビュー	10
・ 財務モデリング	9
・ 内部統制（手作業または準自動化された環境）	6
・ 内部統制（高度コンピュータの環境）	10
・ ゴーイングコンサーンの評価	6
・ 監査意見の判断	5

表付1-7には、監査においてコンピュータの使用が可能な6つの領域に関して行った調査結果（被験者からの回答数）が示されている。

項目(1)～(4)は、すでにコンピュータが導入されている分野である。事実、この項目に関して、被験者である監査人全員から導入使用の賛成が得られた。

項目(5)、(6)は、ESに対する監査人達の信頼度を調べたものである。この結果、17人中16人がコンピュータの使用は、監査人の判断を「支援する」のに適していると感じ、更に1名はコンピュータベースの解答は監査人の判断に「替わり得る」とさえ述べている。

次に、ESの適合性を認めた監査人達に対し、9つの監査領域のうちどれがESの実際の応用に適するかの質問がなされた。選定された領域は、すでにESが現在開発段階にある領域と、導入が適切と思われる領域から成っている。

表付1-8 エキスパートシステムに対する予測

(エキスパートシステム使用)	【前】	【後】
・監査人が修得するには難しい	-1.41	-1.12
・コンピュータが行う思考を監査人に押しつけている	-1.29	-1.24
・監査の“虎の巻”となり監査人は判断をそれに依存するようになる	-0.12	-0.12
・監査実務を非人間化する	-0.94	-0.94
・重大な法律的・倫理的問題が起こる (例：背任行為の訴訟の増加)	-0.29	-0.47
・顧客が持つ監査人に対するイメージが崩される	-0.59	-0.71
・監査人自身のイメージを脅かす	-0.59	-0.77
・最新の状況を容易に維持し難い知識に依存することになる	-0.35	-0.24
・監査の費用を減らす	-0.06	0.06
・監査の時間を有効的に使えなくなる	-0.59	-0.47
・監査人は機械に対し抵抗感を抱く	-1.12	-1.12
・監査人の存在価値がなくなる	-0.94	-1.12
・コンピュータの故障で信頼性が薄れる	-0.94	-0.59
・監査誤りで顧客から訴えられる	-0.24	-0.29
・個人または職業上のプライバシーが脅かされる	-0.82	-0.89

表付1-9 エキスパートシステムに対する要望

(エキスパートシステム使用)	【前】	【後】
・文字入力ではなく音声命令に反応して欲しい	0.41	0.41
・監査人の思考プロセスを模倣すべきである	1.18	0.65
・決して誤った判断をしないようにすべきである	-0.29	-0.06
・監査実務基準に十分使えるスタンダードになって欲しい	-0.47	-0.65
・監査テストの費用対効果を改善すべきである	0.65	0.59
・監査人が覚えて使用するのに労力がかからないのが良い	0.76	0.59
・監査人が修得すべき専門知識の量を減らすべきである	-1.35	-0.77
・常識を表示すべきである	0.71	0.77
・監査人に決定内容を説明できなければならない	1.59	1.41
・監査専門家が使用する過程で自動的に新しい情報を学習すべきである	0.82	0.65
・監査の知識ベースに対する理解度を表示すべきである	0.94	0.71
・専門家の必要性を減らすような事があってはならない	0.71	0.41
・監査人がいつでもどこでも使えるようにポータブルで柔軟性が高いことが望まれる	1.71	1.41

表付1-8と表付1-9は、ESに対する予測と要望を簡単に表わしたものである。

すべての質問事項は、「“十分な”開発を終えたエキスパートシステムは・・・」と言う主語のあとに続くものである。

採点基準は次の通りである。

・ そう思わない	(- 2)	強い No
・ あまりそう思わない	(- 1)	↑
・ どちらとも言えない	(0)	+
・ たぶんそうだと思う	(1)	↓
・ そう思う	(2)	強い Yes

質問は、EDP-XPERT を使用する前後に2回実施した。使用後にも質問を行った狙いは、監査人の考えが、ESを使った後に変化したかどうかを見極めるためであった。

表付1-8ではすべての値がマイナスである監査人は、ESを使うことによって、監査人個人に、あるいは監査という職業に悪影響が及ぶとは思っていないことがわかる。また、ESの使用前と後におけるデータの違いはほとんど見られない。

興味深いことであるが、ティーチ(Teach)とショートリフ(Shortliffe)の研究(1981年)では、内科医達は同様のアンケートに対して監査人とはほぼ反対(質問13項目に対し「そう思う」と答えている)の回答をしている。

表付1-9は、ユーザーがESに期待するものに関して、CASが行った簡単な採点表である。この結果から、CASは、ESに対して以下のような見解を一貫して持っていることが分る。

- ・ 監査人の決定プロセスを模倣する。
- ・ 監査人に決定内容を説明することが可能である。
- ・ 持ち運びが可能で柔軟性に富んでいる。

また、監査人は、ESの導入によって自らが学習して体得すべき専門的知識の分量が減ると言うようなことは無いと考えている。ただ、この評価の強さは、ES使用後は弱まっていた。

表付1-10 EDP-XPERT (AL/X) の評価

以下の特性に関し、今回の実験で使われたES構築ツール(AL/X)を評価して下さい。			
(1)	質問の出し方		2.88
(2)	質問の説明(Eコマンド)		3.31
(3)	その質問をする理由の説明(Wコマンド)		3.08
(4)	プログラムリーズニングの説明(Cコマンド)		3.00
(5)	ユーザーレスポンススケール(+5~-5)		3.24
(6)	プログラムの有用性		2.77
(7)	プログラム・アドバイスの信頼性		2.63
(8)	プログラムのパフォーマンス		2.81
(9)	教育的能力		3.18
全体評価			2.98
【採点基準】 1 = かなり悪い 3 = 良い 5 = 優れている 2 = 悪い 4 = とても良い			

表付1-10は、EDP-XPERT の監査人による9つの評価項目に対する5段階評価を表わしている。このシステムの全体評価(9つの評価項目に対する評価の平均)は良い(2.98)である。EDP-XPERT の能力に直接関係する項目((1)、(6)、(7)、(8))は、平均以下の評価であった。これらの結果から、EDP-XPERT は、どの評価項目に対する改良が必要であるかが分かる。このEDP-XPERT の現在の開発段階の程度を考えれば、アンケート実施者からこのような全体評価を得られたことは、満足すべきであると思われる。

2. エキスパートシステム導入検討の詳細

2.1 導入検討の方針

(1) 検討方針

システム監査にESを導入する可能性の検討は、ES導入によるプラス要素（有効性）とマイナス要素（困難性）の両面から行う。ES構築段階と使用段階それぞれに、プラス要素とマイナス要素があり、主なものはESがシステム監査人の作業を代行することによる効果（使用段階のプラス要素）と、開発する際の問題点（構築段階のマイナス要素）である。また、従来のシステムではあまり問題にならなかった独特のものとして、構築段階のプラス要素と使用段階のマイナス要素も見逃せない。

システム監査にESを導入する場合に考慮することは、先ずシステム監査のどのような領域を対象とするか、またどのような機能が必要かである。その際、システム監査の問題領域は、それ自体の特性により色分け（解釈）がなされて対象領域が検討される。ES機能は、もちろん有効なものに設定されるが、無制限に優れた成果は望まず、無意識のうちに実現可能性も考慮に入れることになる。同様に実現可能性が高いと言っても、無意識のうちにある程度の有効性を考慮に入れている。すなわち、問題領域の特性を解釈することに比較し、機能を想定する場合には、考慮の幅が広いと思われるので留意する必要がある。

一般には機能が増せば構築も難しくなる。すなわち有効性と困難性にはある程度の相関がある。しかし、問題領域の特性はプラス要素（有効性）とマイナス要素（困難性）それぞれに、同等のインパクトをもたらさない場合がある。それぞれのインパクトを考慮し、コストパフォーマンスの高い領域にES導入を狙うべきである。

導入検討の方針としては、有効性と困難性を分析し、その結果を統合して導入可能性を判断する。

(2) 検討事例

システム監査全般の検討に先立ち、ある金融機関のプログラム処理ロジックの信頼性監査におけるES導入の可能性検討事例を紹介する。この事例は、ESを導入する際にシステム監査の専門家を含めた検討グループが行ったフィジビリティスタディからの抜粋である。

金融機関の業務処理系システムでは、金銭詐取を目的としたコンピュータ犯罪が多く発生している。多様なリスクに応じてカバーする対策も物理的、管理手続的、システム的にさまざま実施されている。対策が最も困難であるものの一つに、不正処理ロジックをプログラム内に組み込

む手口がある。いわゆる“トロイの木馬”であり、広い意味では金銭詐取を目的とする以外にも、システムリソースの無権限使用や、“時限爆弾”のようにある時刻にシステムを破壊することを狙ったものも含まれる。金融機関の業務処理系システムは、単純な対策から高度な対策を要するリスク、また影響の程度もさまざまなリスクに潜在的に曝されている。

検討グループでは、システム監査を支援（部分的代行）するツールであれば特に範囲を定めず、プログラム言語もPROLOGやLISPに限定せずに検討した。また、システム監査支援ツール導入により、システム監査作業自体も変化していくことを予め想定して取り組んだ。

導入可能性評価の軸は、有効性については負荷の代替、困難性については常識及びヒューリスティックス（Heuristics：経験的知識）への依存度とした。

専門家が対象領域において問題解決のために使用する知識は、二つの尺度（専門家しか持たない知識であるか否か、使用を意識しているか否か）により、4つのタイプに分類できる（図付2-1）。

A 意識的な 専門知識	B 意識的な 非専門知識
C 無意識的な 専門知識	D 無意識的な 非専門知識

図付2-1 知識の4タイプ

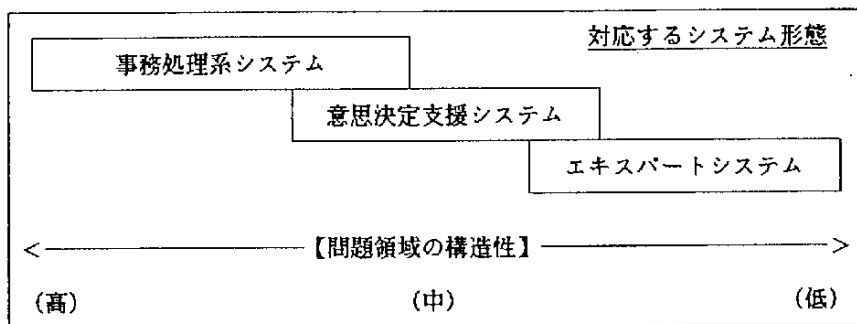
Aのタイプは、専門書やセミナーで示される知識であり、専門家自身もこれが専門知識だと意識している。Bのタイプは、専門家でなくても知っていて比較的容易に使用を意識できる。Cのタイプは、専門家が対象領域において無意識に使用している経験則で、客観性がまだ証明されていない。専門家間で正しさが認められるとAに移行することになる。Dのタイプは、誰でも無意識のうちに使用している知識で膨大な量になる。

この分類で見ると、常識にはB、Dタイプの知識が、ヒューリスティックスにはCタイプの知識が多く含まれていると捉えることが出来る。Cタイプの知識は、体験から身に付いたものであ

ヒューリスティックス：エキスパートと呼ばれる専門家が、長年の経験に基づいて獲得した経験的知識。極めて個人的知識であり、本人が意識していないことが多い。エキスパートシステムの問題解決力は、これによるところ大である。（上野晴樹「知識工学入門」）

り、言葉や文字による伝達に多く依存しないため、意識され難い。

構造性については、評価軸とするか否かが議論の対象となったが、ここでは対象にしなかった。構造性とは、解に至る道筋が一つに定まらず、専門家によってどの程度アプローチが異なる問題領域であるかを示すものである。構造性の低い問題領域は、解に至る道筋がいろいろと考えられアルゴリズム的でない。ここでは、常識やヒューリスティックスの使用は、道筋を構成する要素の属性であるとみなしている。ハンセンとメシアは、問題領域に内在する構造化の度合いと情報システムの形態を関連付けている（図付2-2）。



図付2-2 構造性とシステム形態（参考文献1より）

構造性の高い問題領域では、比較的簡単で決定論的なものをアルゴリズムにより処理する従来のシステム（回答の集計等）が適当である。構造性が中程度の問題領域では、意思決定支援システム（監査用ソフトウェア等）が適している。構造性の低い問題領域では、ESの適当な領域であるが、それは問題解決が構造化状態で行われる従来のシステムより、専門家の推論をなぞって処理し易いからである。ESを導入すべき経験的で人間的な領域とは、構造性の低い領域であるとハンセンとメシアは主張している。

問題領域の構造性を、評価の軸に含めていないのは、次の理由による。すなわち、導入ツールがアルゴリズム的処理を得意とする従来のシステムではなく、ESである必要性を主張するためには、問題領域の構造性が注目されるが、ESに限定しない今回の検討では、有効性としての構造性は意味が少なくなるからである。

以下では、まず検討対象のシステム監査環境で、システム監査人が実施してきた作業タスクを述べ、次に、その作業タスク毎にES導入の有効性と困難性を検討して、ES導入可能性についての結論を述べる。

(a) システム監査人の実施作業タスク

① アクセスルート識別

アクセスルート識別とは、運用体制、システム構成を調査することにより、プログラムライブラリへのアクセスルートを網羅的に識別することである。

この作業によりシステム監査人は、以下に示す6つのアクセスルートを指摘した。

- a. ライブラリアンツールを使用した通常のプログラム更新
- b. 強力なユーティリティを使用したロードモジュールの直接更新
- c. 開発段階、すなわち運用当初から組み込まれている場合
- d. 外部端末から公衆回線を経由する場合
- e. 通信回線から侵入する場合
- f. マシンルームのプログラムライブラリごとのすりかえの場合

② コントロール評価

コントロール評価とは、調査結果に基づき各アクセスルートに対し、コントロールが行われているか、そしてそれは適切であるかについて評価することである。

上記アクセスルート (a.～f.) について、システム監査人はコントロールを次のように評価した。

- a. の評価〔△〕：通常、変更履歴にはユーティリティが出力する新旧ソースコード比較リストが付され、それを担当責任者が承認し一元管理している。しかし、今期実施した大規模なメンテナンスの場合は、一部のサブシステムにおいて、プログラムの変更履歴が十分管理されていないものがみられる。
- b. の評価〔○〕：ユーティリティの管理担当係への意見聴取と、使用履歴閲覧によると、コントロールは十分である。
- c. の評価〔○〕：運用期間の長さから、時間コマンドによる“論理爆弾”の可能性はない。また、プログラム外部のカウンターによる突発型の手口も、システム構造からありえない。
- d. の評価〔○〕：システム的にガードされ、プログラムライブラリへはアクセスできない。
- e. の評価〔○〕：アプリケーションレベルの暗号化が実施されており、鍵管理等、手続き面のコントロールも十分である。
- f. の評価〔○〕：マシンルームの入退室管理、オペレーションの内部統制は十分である。

災害時のバックアップライブラリや、保守作業用ワークライブラリ
ィも、担当係の十分なコントロール下にある。

③ プログラム識別

プログラム識別とは、コントロールが十分でないと評価されたアクセスルートによって更新され得るプログラム群を識別することである。

システム監査人は、メンテナンス時の環境について意見聴取している過程で、現行の更新管理ではカバーしきれないケースが、非常にまれではあるが起こり得ることに気づいた。すなわち、ライブラリアンツールの機能上の不備により、更新があったように表示された場合には、変更履歴を作成しないため、それを悪用するケースである。そのため、そのようなケースに該当するプログラムも併せ、プログラム名、構成モジュール名、外部ソースコード名等をリストアップした。

④ 優先度評価

優先度評価とは、該当するプログラム群の個々の機能と、マネジメントのセキュリティ方針からプログラム毎に検証の優先度を評価することである。

システム監査人は、各プログラムに不正ロジックが組み込まれることにより、どのようなコンピュータ犯罪が遂行されるのかを想定する。正当なプログラムロジックとの類似性を無視して、ひとまとまりの不正なロジックコードを組み込めば、所期の犯罪を達成するために組み込みの対象とするプログラムの範囲は広がるだろう。しかし、技術的な組み込み易さと、発見され難さを考えるためか、不正なロジックは、正当なロジックと同様の機能から作成される場合が多い。すなわち、正当な機能として預金残高の更新や、利息計算を行っているプログラムに、コード行数で言えばわずかな改ざん、組み込みがなされる可能性が高い。組み込まれる可能性とリスク顕在時の影響、及びセキュリティ方針から各プログラムに、A、B、Cの3段階の検証優先度を付加した。

⑤ 検証効果確認

検証効果確認とは、技術環境等を調査することにより検証効果を確認すること、すなわち、プリント出力したプログラムソースコードの正当性が検証されれば、情報システムにおいて、実際にある期間機能しているプログラムが正当であると判断できることを確認することである。

システム監査人は、実際に機能しているモジュールと、検証したモジュールが同じものであること、さらに、ソースモジュールとロードモジュールとが同じものであることをコーデ

ィングから実行に至る流れを追い、担当者に意見聴取し、確認した。

⑥ 検証実施

検証実施は、優先度が高いと評価されたプログラムのソースコードを実際に検証し、正当性を確認することである。

金融機関の信用を著しく損なうことになりかねないある特定の不正なロジック、組み込まれる可能性が大きいプログラムについて、システム監査人がソースコードの正当性検証を実際に行った。対象のソースコードは、高級言語のコーディングステップ数で、約6千ステップに相当した。

⑦ 検証方法指示

検証方法指示とは、プログラム機能に応じた効果的な検証方法を指示することである。

その他のタイプの不正なロジック、または組み込まれる可能性のあるプログラムについて、システム監査人は検証を提言するために、どのようなリスクが潜在しているのかを犯罪者の主な目的別に具体的に示し、さらに不正か否かを判断するポイントを明らかにした。例えば、口座入出金フロー中の特定条件下での転記部分や、機能的に重要なファイルへ更新モードでのアクセスコマンドを発行している部分等に関して検証方法を説明した。

⑧ 監査証拠収集

監査証拠収集は、上記評価等の主張根拠となる監査証拠を収集することである。

システム監査人は、自己の意見が客観的で妥当であることを主張するために、ドキュメント、出力リスト等を監査調査として収集整理した。

(b) ES機能設定

ESは、システム監査人と全く同じ作業ができないことは当然であるので、システム監査人の作業タスクをそのままES機能と考えて、導入の有効性や困難性を検討することは現実的ではない。

AIの中の自然言語理解やロボット工学が、近い将来その制約を解消するまで、システム監査人の作業タスクを、ESが理解できる形に置き換える作業が必要である。ES機能の設定段階において、すでに無意識のうちにES導入の有効性、困難性の評価を行っている。

さらに言えば、ES導入の検討にあたりシステム監査人の作業をタスクに分解する段階、また作業そのものを認識する段階も、その評価の影響を受けているだろう。それは、“ES導入検討のESが必要である”、“先ずプロトタイプを作成してみることが近道である”との意見の背景をなしている。

しかし、対象領域が定まれば、その作業負荷やヒューリスティックスへの依存度は、ある程度定まるはずである。その判断は、システム環境とシステム監査人の作業経験に依存する。

システム監査作業フローの狭い範囲において、システム監査人の作業をできるだけ明確にし、ES機能を検討対象の情報システムの監査環境において、出来る限り妥当な線に設定した（表付2-1）。

表付2-1 作業タスクに対応して設定したES機能

作業タスク	入 力	出 力
アクセスルート識別	運用体制、システム環境に対する質問の答え	アクセスルートの指摘
コントロール評価	各アクセスルートに対する質問の答え	コントロールの評価
プログラム識別	アクセスルート	プログラム群
優先度評価	プログラム機能やセキュリティ方針に対する質問の答え	プログラム毎の検証優先度
検証効果確認	ソースコードに対する質問の答え	効果の評価
検証実施	ソースコード	ソースコード毎の不正である可能性
検証方法提示	プログラム機能	検証方法
監査証拠収集	評価結果	評価結果に至る論理の説明

(c) ES導入の有効性・困難性の検討（各作業タスク単位）

(b)で設定したES機能に基づき、各作業タスク毎にES導入の有効性と困難性の観点で検討した。その際、有効性及び困難性を評価する観点は、既に当節の冒頭で述べた通り、有効性については作業代行率、困難性については常識及びヒューリスティックスへの依存度である。

以下、検討中に得られたコメント例を示す。ただし、この意見の背景には検討グループの実施した作業に対する認識と、領域によってある程度偏在するES導入への期待があることは否定できない。

① 「検証効果確認」作業タスクへのコメント

検証効果確認について、“検証効果確認は、常識に多く依存している。”とのコメントが得られた。

問題解決に使用する知識、すなわち知識ベースに登録すべき知識が“専門知識”の他にも

多くに及ぶ場合は、知識獲得表現が困難である。

「検証効果確認」という作業タスクで使用する知識のうち、システム監査の専門家しか持たない知識の占める割合は、それ程大きくない。

ドキュメントの表現から、作成者の真意を察するようなことも、システム監査人は行うことがある。すなわち、例えば的をややはずした文脈や、曖昧な言い回しが故意であれば、人事、組織管理に何らかのマイナス要因が潜在している可能性があり、それなりの注意を傾ける必要があると言うことも、システム監査人は知識として使うことがあるが、分野特有の専門知識と言うより、通常我々が使っている知識に近い。また、もしドキュメントの信頼性が低いならば詳細な調査を続行せよと言うような、割り切りの実効性も期待しにくい。

システム監査人の専門知識であるとして知識ベースの登録を試みるなら、それはシステム監査人の思考過程の分析というより、人間の思考過程の分析に近く、当面の目的以外の問題解決のために膨大な工数が必要になるだろうという意見もあった。

このタスクでは、他にも次のような知識を有し、E S 導入の困難性の要因である常識依存度が高い。

- ・納得の度合と効果に対し、予想される反応
- ・妥当な量的、質的ノルマ
- ・聴取意見の信頼性

② 「アクセスルート識別」作業タスクへのコメント

アクセスルート識別について、“ルート識別は、ヒューリスティックスに多く依存している。”とのコメントが得られた。

システム監査人は、「アクセスルート識別」という作業タスクを、ドキュメント調査、意見聴取等により実施するが、識別したアクセスルートは、後続作業である「コントロール評価」が容易なように切り分けされている。

アクセスルートの網羅性や、識別根拠を説明するためにアクセスモデルを作成することもあるが、システム監査人の論理手続順序としては、確認や説明のために、結論から行うことが多いだろう。

システム監査の駆け出しの期間に、O J T (On The Job Training) でいくつかのアクセスモデルを作成したり、識別作業をサポートしたりするうちに、ノウハウが身につくものである。

アクセスルートの羅列のレベルで満足するならば、過去の犯罪事例にみられるアクセスル

ートのうち、いくつかを質問の答えから絞り込むことになるが、その後、各アクセスルート
の調査確認を行わねばならないことを考えると、負荷の代行程度はあまり高くない。システム
監査人の質に近い機能をESに持たすのが困難であるのは、ヒューリスティックスが取り
出し難いからである。ヒューリスティックスは、マシンルーム、システムネットワーク、運
用体制毎に多様に場合分けされるであろう。

(d) 検討事例の結論

ES導入可能性検討の結果、検証実施、すなわちソースコードを読み不正である可能性を評
価するESを開発導入するという結論に達した。

代行率に関しては、前記の理由に加え、全作業タスク中の検討実施タスクが、検討グループ
の実際作業としては大きな割合を占める。また、それは個々のソースコードに対する評価の集
積であるため、プログラム処理ロジックの信頼性監査作業内だけで実効性の確認テストも行い
易い。

不正であるか否かを判断する作業は、常識にあまり依存せず専門知識としてまとまっている。
ヒューリスティックスについても、対象の本質の把握を前提とするようなものは少なく、専門
家同士のディスカッションをある程度重ねれば、客観的な専門知識になり易いものが多いと判
断された。

検証実施のES導入は、被監査部門へのロジックチェック提言及び作業を、監査人側が行な
えるようになる、すなわちサービス向上の意義もある。

プログラム処理ロジックの信頼性監査作業のES化は、最終的に目標とする機能まで段階的
に実効的機能が想定され、かつそれらは構築順序に沿っている。具体的には最終目標の不正部
分の指摘から不正可能性部分の大まかな指摘、不正判断の方法指示と並んでいる。よってES
開発導入自体のリスクも少ないと判断された。

この結論に基づき、ESのプロトタイプ開発が実施された。工数は、KE1名、SE2名、
コンピュータ犯罪専門家1名、その他リサーチ2～3名の構成で約3ヵ月要した。

その後機能向上が続けられ、内部において実際に試行された。その結果、当初設定した目標
を満足する効果があがり、現在でもさらに機能向上が行われている。

2.2 導入の有効性の検討

(1) 有効性のタイプ

ESは、専門家の作業代行という主目的を達成する以外にも、次に示すように副次的にいくつ

かの効果をもたらす。

【構築段階】：a. 知識の資産化

【使用段階】：b. 作業代行

c. 知識の共有

d. 専門家の育成

e. 差別化

これらの有効性の程度は、導入の環境、すなわち主体や作業領域等によって異なる。例えば、専門家の育成が体制として整っているような領域より、専門家の育成のために長期のOJTが不可欠な領域の方が、ESによる専門家の育成支援の有効性は大きい。

そこで、これらの有効性を、システム監査の各作業領域に対してもたらす効果の程度がほぼ同様とみなされるものと、異なるとみなされるものとに大別して検討して見る。前者、すなわちシステム監査各作業領域を区別しない（共通的な）有効性は、a、c、d、eである。その中で、有効性の評価軸となるものは、システム監査領域に該当するものであるが、その比重は、導入の環境に依存している。後者、すなわちシステム監査各作業領域を区別する（固有の）有効性は、bである。これは、どのシステム監査作業領域への導入がより有効かについての評価軸となる。

(2) 共通な有効性

(a) 知識の資産化

システム監査人が個人のノウハウとして持っていた知識は、ES構築を経て資産化する。それには、整理、体系化と定着化の意味がある。

知識獲得（提供）の過程で、システム監査人は、問題解決のためにこれまでどのような知識を使っていたか明確に認識するが、それは同時に、知識の整理、体系化を促すことになる。これは、ESの使用段階を考慮しなくても実現される有効性である。

また、知識表現と知識ベース構築により、システム監査人の内部にあったノウハウは、半永久的なものとして組織に定着化する。

これまで、人間の頭の中で主に処理されてきた情報が、EDPの普及により人間から分離され、人、物、金に加えて4番目の経営資源として捉えられるようになった。業務処理用の情報に続いて、専門知識もこのように分離され定着化することは、ノウハウを持った担当者の転職のリスクを軽減するというような意味でも、企業にとって有効かもしれない。

(b) 知識の共有

資産化された知識は、複数の人間に利用される。これは、その知識を持つ専門家を複数擁す

ることと同じである。いくつかの監査対象システムが、高品質のノウハウにより同時に診断され分析される。さらに、共有した知識は、各々のノウハウにフィードバックし、品質はますます向上する。（例えば、シカゴで応用された高度な監査ESが、磁気媒体を経て東京に運ばれ応用され、さらに、その経験と独自のアイデアを加味され機能向上したESが、次はロンドンで応用される。）

また、知識の共有は、複数の情報システムに対する統一的な評価基準を提供するという有効性もある。監査内容は、平準化され客観的な視点が得られる。

(c) 専門家の育成

専門家のノウハウを持たない人間が、専門家並みの成果を得る目的としてESを使用する環境は、専門家育成にはかなり有効であると思われる。なぜなら、推論過程をなぞることにより、どの段階で専門家との推論ルートが分かれたか、またどのような知識が自分に不足していたかが明確に指摘されるからである。従来、ある程度長い期間を専門家との実務作業を通じて育成されてきたシステム監査人が、ESを操作するうちに、専門家のノウハウに近いレベルまで自然に育成される。我が国では、システム監査人が少なく、監査チームにシステム監査人が十分得られない場合が多く、代わりに監査経験者やEDP経験者がメンバーとして加わる。このような状況では、ES導入によって監査経験者は、システム監査ESからEDP知識を、そしてEDP経験者は、監査ノウハウを学ぶことができる。

(d) 差別化

差別化は、システム監査をビジネスとして捉えている場合に主に該当する有効性である。ES導入により、システム監査人の負荷が軽減され、作業品質の向上、経費の低減が実現されれば、競争力は強化される。待望されているにもかかわらず実績がほとんどない状況でのシステム監査ES導入は、差別化戦略として優れている。

(3) 固有な有効性 — 作業代行 —

作業代行的有効性は、実施者の作業負荷をどれだけ軽減するか（絶対効果）と、ES以外の方法では代行実現性がどれだけ低い（相対効果）の2面から捉えられる。

(a) 絶対効果（負荷代替）

ESの代行が実務作業としてどれだけ有効かは、代行される作業が実施者にとってどれだけの負荷になるかに等しい。

作業ごとの負荷は、各々のシステム監査により異なるが、監査環境が決まればある程度定まり、作業代行的有効性が識別出来る。ただし負荷の大きさは、作業実績の工数が一応の参考に

はなるが、実務者の捉え方が大きなウェイトを占める。また、ESはもちろんインタビュー等フィールドワークそのものを代行できないので、代行の対象は、そのような行動を除外した残りの知的作業になる。

システム監査ESの潜在的な構築者は、各々対象とする情報システム等の監査環境を想定しているので、各作業の負荷を幅を持って見積もることが出来る。内部監査部門では、個別テーマにもよるが、作業の負荷をある程度は経験から把握できる。また、システム監査を業とする場合は、逆にあらかじめ負荷のかけ方を設定してとりかかる場合もある。これは、システム監査ESが、システム監査作業自体に影響をおよぼす例である。一般的なシステム監査の各作業の負荷を示すことは出来ないが、固有な有効性の検討の際は、選択した事例が極端に変則的な割合を持たないことは指摘出来る。

(b) 相対効果（構造的性）

相対効果は、専門家の希少価値のようなものである。従来のシステムで為し得る機能なら、ESの相対効果は低い。また、高度な専門家でなくとも、マニュアル等の参照により誰でも解決できるような問題領域では、やはり相対効果は低い。

システム監査ESの相対効果が高い領域か否かは、問題領域の構造的性に依存する。構造的性が低いシステム監査の問題領域が、ESを導入すべき領域である。そのような領域において、専門家はアルゴリズム的な思考はあまり行わない。専門家によって、推論ルートが異なるということがしばしば起こる。そのような観点から問題領域を見ると、その相対効果の色分けが可能である。

2.3 導入の困難性の検討

(1) 困難性のタイプ

ESに関する数々の研究や実用化の試みによって、ES導入のボトルネックは、知識獲得(Knowledge-Acquisition)にあることが分かった。これは整備されたES導入環境で：ますます比重が大きくなる本質的な困難性と思われるが、他にも次に示すような見逃せないポイントがいくつかある。

【構築段階】：a. 知識自体の獲得困難性

b. 専門家

c. KE (Knowledge-Engineer)

d. 知識表現

e. 機能タイプ

f. コスト

【使用段階】：g. 運用負荷

h. 出力評価

これらを有効性検討の場合と同様に、システム監査の各作業領域に対し及ぼす困難性の程度がほぼ同様とみなされるものと、異なるものとに大別して検討してみる。前者、すなわちシステム監査各作業領域を区別しない（共通的な）困難性は、b～hである。その中で特にシステム監査領域に該当するものが、基本的な困難性となり、各々の比重は導入環境に依存している。後者、すなわちシステム監査各作業領域を区別する（固有の）困難性は、aである。これは、どの作業領域への導入が困難かについての評価軸となる。

(2) 共通的な困難性

(a) 専門家

ES構築の前提条件としては、十分に経験を積んだ専門家の協力である。ES機能の上限は、この専門家の知識によって決まる。教科書的知識を知識ベースに入力し、ESを構築しても、単なるテキストシステムとなる。

知識が次々とめりかえられる分野においては、ESが実効的であり続けるためには、構築後も専門家の協力が継続的に必要である。このような専門家の確保は、環境によっては必ずしも容易ではない。

システム監査の分野では、情報処理システム監査技術者試験、CISAの認定試験が実施されている。そこで提供された標準的フレームワークに、各々独自の監査経験、判断ノウハウがのせられ体系化されつつある知識なら、獲得対象として十分である。

(b) KE (Knowledge-Engineer)

KEの役割は、専門家に質問しその答を分析し、問題領域において専門家が使用する知識を獲得することである。知識獲得のためには、KE自身がある程度の専門家にならねば出来ないといわれる。専門家は説明する際、自分にとってはあたりまえだが実際は彼しか知らない知識を、誰でも知っているはずだと、しばしば勝手に決めこむ。専門家自身も無意識で使っている知識を説明するのは概して困難である。そのような知識はたいていは経験則である。このため、専門家と良好なコミュニケーションを保つことは、KEの当然の前提となる。優秀なKEの不足は、ES構築時の大きな問題とみなされている。ES導入の先進国である米国でも、このような意見は多い。

(c) 知識表現

現段階のESは、プロダクションルールやフレームのように、知識表現に各々決まった形式を用いている。問題解決のために使用される知識形態と、ESの知識表現方法との適合が問題となる場合がある。ES導入者の側からは、知識ベースに表現された知識の意味が理解し易い方がよいが、処理するシステムの側からは、操作し易いようにあらかじめ定められた形式に変換されている必要がある。両者を調整した表現は、対象領域に対して大きな制約となることがある。しかし、システム監査及び関連領域の事例から、知識表現はクリティカルな困難性ではないと考える。

(d) コスト

成功したと公表されているESの多くは、その構築に膨大な投資と長い開発期間を要している。ESを利用することによってそれを回収するための収益をあげたり、経費を節約できたりするか否かはその問題領域による。地下資源を発見したり、融資判断の支援により業務処理効率が向上したりする例は、効果の定量化が容易なケースである。

システム監査領域では、ESを誰がどのように使うかにより異なるが、最近の大手会計事務所の動向を見ると、経済効果が期待できる分野であるとの認識がある。

(e) 運用負荷

システム監査人は、効果的なシステム監査実施のために、技術革新やコンピュータ犯罪動向について、不断に学習しなければならない。このような分野において、ESが実効的でありつづけるためには、知識ベースの改訂が必要である。この要求は、システム監査各作業領域間で全く同じではないが、個々の作業領域へのES導入可能性に影響を与えるほどの差はない。しかし、概してES運用の負荷は大きくなる。

(f) 出力評価

ある問題にアプローチする推論ルートは、専門家によりしばしば異なる。場合によっては、推論ルートのみならず結論も異なることがある。ESが持つ知識も、特定の専門家のものであり、その問題に対し異なる主張をする専門家がいたことは十分考えられる。すなわち、ESがたとえ専門家レベルの結論を出したとしても、「正しい」か否かの判断は難しい。

ESの結論が、個々の知識提供者の意図した結論と異なり、かつ、その結論に対し多くの専門家が「正しい」と判断した場合は、単にESが意図した通り機能しなかったとも考えられ、一概にESが専門家を越えたと結論出来ない。また、ESの誤判断により損害がもたらされた場合の責任の所在は、責任を肩代わりするシステム監査において、将来問題としてクローズア

ップされる可能性がある。

(3) 個有の困難性 — 知識自体の獲得困難性 —

知識獲得の困難性は、専門家やKE等の作業環境にもよるが、本質的なものは知識自体の特性である。どのような特性に依存するかは、次に示すように、これまでの研究や試行により経験的ではあるが徐々に判明してきた。

- ・解決のために常識を多く使う問題領域は難しい。
- ・解決のためにヒューリスティックスを多く使う問題領域は難しい。

以下では、領域毎の困難性に対し、常識、ヒューリスティックスの2面から検討する。

(a) 常識を使う問題領域の困難性

専門家が問題解決のために使用する知識には、大かれ少なかれ常識が含まれている。使用する知識を全て何らかの形で知識ベースに格納する必要があったり、常識と言われる知識には曖昧性が含まれる場合が多いために、常識を使用する割合が大きい問題領域では、ES構築は困難である。専門領域が小さく切り取れる領域でないと知識ベース構築はかなり難しい。

F.HAYES-ROTHは、“BUILDING EXPERT SYSTEMS”において“成功のコツは問題の対象領域をそれ程多くの常識を必要としない専門分野に置くことである”と述べている。

ESが担当する専門領域の境界で急激に能力が低下することは、境界では常識の使用割合が増えることに関係がある。システム監査の各々の問題領域には、常識を多く含むものから、そうでないものまで混在している。

上記の観点から、常識を多く含む問題領域においては、範囲の制限と境界の明確化、または機能の制限等が要求され、かなりの低レベルで甘んじなければならないことを覚悟すべきである。システム監査における、このような問題領域は、次のようなものがある。

① システム監査作業自体のリスクを勘案する領域

システム監査に対するマネジメントニーズ、システム監査チームメンバーの当該対象に対する得手、不得手、基礎的な監査環境等を勘案する領域。

② 現場視察やコミュニケーション・スキルを要求する領域

AUDIT という語にも示されているように監査においては“聞くこと”が重視されており、そこが単なる検査と異なるポイントのひとつである。システム監査人のヒューリスティックスと常識とは明確に切り分けされるものではないが、このような行為には多くの常識が含まれており、たとえノウハウの指示レベルの機能であっても、システム監査人とはかなり差が出てしまうであろう。

③ 法律的な解釈が必要となる領域

法律文は、その環境を常識を用いて解釈するようにできている。システム監査では商法・刑法に加え、そのような性質を持つ様々な基準をも使用する。このような領域においては、参考資料を提示する程度に機能を制限せざるを得ない。

④ 定量化し難いリスクの評価領域

異なったタイプのリスクを比較して評価するためには、システム監査人は各リスク顕在化の場合の損害を常識に照らして考える。どのようなリスクがあるかとかリスクイメージについては、ヒューリスティックスに主に依存するが、定量化不可能なリスクにおいては、リスク評価は、マネジメントに適切に伝えなければならないために、常識が最終的な拠り所となることが多い。

(b) ヒューリスティックスを使う問題領域の困難性

専門家は自分の解決への道筋、すなわち使用した知識の詳細をしばしば説明することができない。ヒューリスティックスを含む知識獲得は困難であるが、ヒューリスティックスこそが専門家知識の重要な部分であり、獲得を通して知識の整理と共有が可能となる。

システム監査人は、膨大なヒューリスティックスを持っており、それは過去に対象とした情報システムに主に依存している。このとは、技術革新が激しく情報システム環境が変化している状況では、ルール成立の前提条件（IF部）を引出すのに工数を要することを示している。システム監査人が使用するヒューリスティックスが多くなれば現段階の技術環境では、プロダクションルールによる知識ベース化のみならずフレームの使用も検討しなければならない。

ヒューリスティックスが多く使われているシステム監査の領域は、次のものがある。

① 効率的な作業手順の作成領域

企業及び情報システム部門の内部統制上の特徴から、効率的な監査作業実施手順を作成するような領域であり、フォローアップ手順の作成も同様である。

② 情報システム診断の際の着目領域

開発環境から情報システムの品質阻害要因やセキュリティ上のクリティカルポイントを見つけ出す領域。

③ 最終結論を下す領域

マネジメントニーズ、内部統制状況、情報システムの技術環境、コンピュータ犯罪の動向等、様々なレベルの問題をまとめて考慮し判断を下すような領域。

3. システム監査におけるエキスパートシステム 導入可能性検討の詳細

3.1 作業領域と機能

ESを一般の領域とシステム監査に導入する場合の、各々の有効性と困難性をこれまで述べた。ここでは、システム監査ESを導入するという前提に立ち、作業領域の選定を試みる。検討に当たっては、システム監査各作業に依存する有効性と困難性に着目し、さらに各々の問題領域を色分けする評価軸として、以下の4つを取り上げる。

a. 有効性について

- ・ 負荷代替
- ・ 構造的性

b. 困難性について

- ・ 常識
- ・ ヒューリスティックス

前章の検討事例においては、負荷代替、常識、ヒューリスティックスを評価軸として取り上げて、ES導入可能性の検討作業を示した。ここでは検討事例を、さらに次の2点について留意して検討することとする。

- ・ 範囲を全体に拡張すること。
- ・ タスクが固有な状況下ではなく、概念的であること。

前者は、対象をさらにシステム監査全領域に拡張した場合の評価軸の適用可能性を、強く主張するわけではない。ただ今後特定領域の事例が数多く集められ、領域による導入可能性の演繹が行われる過程で、1つの要素たり得るかもしれない。検討事例の領域とシステム監査全領域の等質性を述べるかわりとして、次の指摘をするに留める。

- ・ 構築環境、使用環境は等しいと考えて良い。
- ・ 導入目的はほぼ等しい。
- ・ 領域には包含関係がある。

後者に関して、ある大手会計事務所の6名のシステム監査担当シニアに対し作業負荷、解が分かる頻度、問題の難しさ、システム監査人以外に解を説明することの困難性等のインタビューを行い、評価軸ごとに各領域を捉え直すことを試みた。それにより洗い出されたES導入可能なシステム監査の作業領域が、特殊なシステム監査環境のタスクではないことを確認した。すなわち、この

洗い出されたシステム監査の作業領域のES導入可能性を肯定するが、他の領域において否定するものではない。ただし、この場合は前述の有効性と困難性について、特に十分検討を加えることを提言するものである。

以下では、ES導入可能なシステム監査作業領域毎にES機能のイメージを述べる。

(1) プロポーザル作成

本ESは、プロポーザル作成作業を総合的に支援するものである。ただし、マネジメントニーズ、監査リスク把握等の作業では、ESにシステム監査人が解釈した結果を入力する。抽象的テーマから具体的な監査方針に落とし込み、監査チームメンバーのスケジューリングを行い、それを基に予算を編成する。プロポーザルドラフトとは別に予想されるクリティカルポイントも指摘する。大手会計事務所やコンサルティング会社等では、これまでのノウハウからこの各作業についてそれぞれ開発された機能を統合し得るであろう。それにより、業務効率向上等様々な有効性の獲得が実現できる。

(2) システムの概要把握

本ESは、アプリケーションシステムの機能・構成・組織・開発段階に関する基礎的な調査結果を入力し、システムの概要説明書及びセキュリティ対策とリソース使用効率に関しての大まかな着眼点を指摘するものである。その詳細さの度合は、システム監査人がイメージする時の切り分けレベルとなるように設定する。すなわち、チェックリスト回答分析や基礎資料調査を担当したシステム監査メンバーが、システム監査チーム全体にシステム概要を予備調査段階の作業として報告する時の表現が、ES出力の目標である。

着眼点は、概要説明作成過程で通過したルールのトリガーを黒板に書き込む等を行い、ピックアップする。分析はここでは原則として想定しない。ESが出力する概要説明書は、プロトタイプ段階でも対象システムについてのディスカッション用ドラフトとして監査チームで使うことができ、ESの機能向上と拡張及び評価が通常の監査業務内で行い易い。被監査システムをあまり知らない企業外部の監査チームでの導入を主に想定している。

(3) 監査支援ツールの使用

本ESは、監査支援ツールの使用方法と対象とを決定する。すなわち現段階で使用可能な監査ツールのうちから、状況に適したものを選択するだけでなく、さらに対象領域と具体的な実施方法に関する計画も作成する。スナップショット法を行うタイミングと対象とするファイル、またソースコード比較とアプリケーションのどの機能が等を合わせ、監査目的を満たすテスト計画としてツール適用準備書を作成する。実際のシステム監査では、幾つかの監査ツールを組合せて使う

ことが多いが、知識はそれぞれのツールに共通するものよりも、固有のツールに属しているものが多く、ツリー構造に表現し易い。したがって、ノウハウは個々に獲得しても十分有効であり、拡張性も期待できる。

(4) 調査結果の分析

システム監査人は、監査ツールの使用、ドキュメントの閲覧、インタビュー等により、膨大な調査結果を手にするが、この調査結果をESの分析対象とする。システム監査人は多くの問題領域で、分析から解釈まで、場合によっては評価まで瞬時に行う。ここでは、ES機能を明確にするため、あえて分析と解釈及び評価を区別する。ESは、アプリケーションや監査ツールの出力の原因を逆にたどったり、ドキュメントやインタビューにより主張された事柄の理由付けが、論理的に可能かをその上で突きとめる。この作業では、ESは、各々の監査計画に依存する特定の視点は使用しない。また、定量化を目的とした、解釈に基づく軸も使用していない。被監査部門へのシステム監査人のインタビュー結果は、ESによる画面からの質問等、指示に従って入力する。したがって、入力される答に、解釈・評価をうまく取り込むよう考慮して質問を作成するのが現実的である。

調査対象ドキュメントが磁気媒体に格納されているなら、ESに直接レビューさせ得る可否かは記録形態に依存する。それは、ESのと言うより、むしろパターン処理の困難性である。ESへの手書のドキュメントの入力は当然さらに難しく、自然言語理解の領域である。調査結果分析ESはその力をあくまで分析に発揮するものである。入力データの様式に起因する問題は必要なかぎり対象としない。

(5) 問題点の評価

評価に必要なのは対象に関する分析結果と、評価軸および分析結果を軸へ落とし込むノウハウである。解釈段階でシステム監査人は固有の視点を導入し、それにより分析結果の意味付けを行う。さらにシステム監査人は、自己の感覚を最も明確に投影してくれる軸を設定し、定量化を狙う。意味付けから評価軸の選定を行う解釈段階は、システム監査人と同等レベルの作業を、ESに期待するのは難しい。よって、評価軸は、幾つかあらかじめ設定されたものから解の探索として選びだす。ノウハウは、評価軸ごとに少しずつ獲得していく。それは、グループ分けされた分析結果毎の軸に対するばらつき具合や他システムや基準値からのへだたり、定量化の信頼度についてである。

3.2 エキスパートシステム導入における考慮点

これまでの検討は、視点を導入完了後におき、それまでの過程にみられたプラス要因、マイナス要因を一括して勘案したものである。困難性は主に構築段階にあり、有効性の主なものは利用段階にあるが、その時間差はあまり考慮されていない。したがって、検討結果は、ES導入が途中で放置されず、何らかの形で利用段階まで進めることが可能な環境では有効である。

そのような環境であるか否かは、導入プロジェクトメンバーの経験とES機能に依存する。しかし、プロジェクトメンバーの多くがES構築と利用段階を経験していないような場合、初めに直面するES構築段階の困難性が大きな障壁とみなされるのは当然である。すなわち、どこにESを導入するのがよいかという問題に劣らず、どこからESを導入していったらよいかという問題が重要になる。以下では、その点を踏まえてES導入の際の考慮点を示す。

(1) 構築を着手する領域

構築のリスク軽減を重視し、ビジネス環境において現実的にES導入を進めるためには、システム監査人の作業がルーチン化している領域を選択する。極論すれば、システム監査作業1年目のメンバーにも理解できるマニュアル化されたルールを、知識ベースに表現する。我々のこれまでの検討は、このような従来のシステムでも実行できる機能なら、ESとしての有効性は低いというものであるが、ここではアプローチのスタンスが異なることを明記しておく。

先ず上述のように、容易なところから着手し、それにシステム監査人のノウハウを少しづつ加えていくのが、この前提において最も効率的である。このような容易な箇所は、うまく対象領域を絞り込んだり、対象領域において解空間を捉え直すことによって得られる。どの問題領域においてそのような絞り込みや、捉え直しを行ったらよいかは、ES導入の狙いに依存する。

以下に、導入の狙い毎に分類して説明する。

(a) 客観的視点の獲得（他社との、また時系列の）

同様の業態にある他社の情報システムとの比較を客観的な評価軸にのせるニーズが強い場合、選定される領域は概要把握や、監査手順に関するようなものではなく、リターンに大きく関係する領域、すなわち、リスク識別、リソース使用効率、対策提言等であるだろう。また、時系列比較の場合は、情報システムの健全化が徐々に向上していく様子を反映しやすい領域、すなわち、対象の選択、調査結果の分析等である。

(b) ノウハウ、知識の社内への定着

システム監査ノウハウがかなり蓄積されているにもかかわらず、個人の内に留まっている状況の場合、通常ノウハウは、或る特定のシステム監査人に偏在している。すなわち、ノウハウや

スキル毎にスペシャリストが特定されるはずである。したがって、テーマごとのスペシャリストを知識獲得の対象とし、優れたノウハウと深い経験を知識ベースに格納していくのが現実的である。

(c) 作業負荷の軽減、効率の向上

実施している作業の詳細タスクへのブレイクダウンと負荷分析により、担当者の負担を軽減し、省力化の行える領域を選択する。

(d) 優れたノウハウの共有によるサービスの向上

サービスの向上につなげるためには、優れたノウハウを付加する核となるものの存在が必要である。近年米国では、領域シェル (DOMAIN SHELL) の必要性、有効性が主張されている。システム監査領域に固有な知識表現方法や、説明機能等のユーザーインタフェース機能を充実させた構築シェルの存在が、システム監査ESをより実効的なものにすることは納得できる。ただし、領域シェルの要件は、個々の試行が演繹され、固まっていくにつれ出来上がるものである。もし、このような領域シェルが中核にあることを前提とすれば、狙いを満たす領域は最近のシステム監査ノウハウ、特化した環境の優れたノウハウにより知識の更新が著しい所、例えば、具体的セキュリティ対策提言、システム稼働状況分析等であろう。

(e) 差別化による競争優位の確立

待望されている分野に、測り知れない可能性を秘めると考えられているESを導入することにより、差別化は可能である。その意義を重視する場合、実務作業に有効な領域という条件はある程度押さえられ、構築が困難でない領域が対象となる。すなわち、構築は比較的容易だが、少なくとも根底には難解な概念を含んでいたり、見えがいたりするものである。将来的な実務に対する有効性という点でハイリスク・ハイリターン・のスタンスをとり、実証性の大きくない先端の理論を採り入れる方法は、戦略的な意味で効果をもたらす続ける。すなわち、統計的手法に独自の考えを入れた数値解析や知識工学の先進の理論との結びつけは、魅力的で狙いに沿っている。ただし、高い倫理性を要求されるシステム監査において、事実および事実と確実に結びついた推論、その両者から遠ざかりすぎた差別化は結果的には受け入れられず、また情報化社会健全化への貢献も少ない。

(2) 構築の際の懸念について

(a) 社内の認知

ESをビジネスへ導入する場合には、社内の認知が得にくいとの意見をしばしば耳にする。システム監査を業とする会計事務所やコンサルタント会社の場合はあまり該当しないが、構築

者が内部監査部門の場合等においては解決すべき問題であり、その傾向は米国よりも我が国の方が強いかもしれない。それは、知識の整理や体系化は、リターンとしてなかなか定量化しにくいためである。目に見えるリターンを考慮するなら、これまで検討した有効性を踏まえた上で、次の点が注目される。

- ・システム監査人の教育投資
- ・部分的省力化による作業範囲拡大

また、ESに事務処理レベルの機能を実行させるのには決断がいるが、次のような機械化による基本的な有用性もあげることができる。

- ・人間が行うより見落としが少ない
- ・作業内容の質が平準化される

(b) KEの存在

我が国よりESにおいて数年進んでいる米国でも、KEは質・量ともに不足していると言われる。KEの助力が特に必要なのは、専門家としての意義が成果物にあり、その過程の説明があまり要求されない分野である。KEは、インタビューだけでは知識獲得が困難で、ある程度専門家となるため自ら弟子入りすることもある。

ところがシステム監査ESの構築者としては、前述のようにシステム監査人の場合が多く考えられる。かつ、システム監査人は結論に至る自己の判断過程をマネジメントや被監査部門に納得のいくように表現し、報告することをルーチンワークとしている。

KE不足が叫ばれている現状を考えると、システム監査を全く知らないKEにあえて依存するより、システム監査人がKEの役割を代行し、知識抽出をする方が望ましい。

KEに専門知識を学ばせるより、専門家にKEの役割をさせた方が、プロトタイプ開発が早かったという米国の事例は詳細な前提なしには論拠にならないが、ある問題領域にとっては、専門のKEが必要不可欠ではないこと、また、KEがそれほど大きな成果をあげられない場合があることを示している。

また、ES実績が少ない現状では、LISP、PROLOG等を使ってESを構築するプログラミングスキルと、知識獲得、知識表現過程の相異及びES構築経験とKE的能力との相異を明確に認識する必要がある。

以上、システム監査のES構築において、我が国におけるKE不足が現在のところクリティカルな障害にならない可能性を指摘した。

(c) 構築シェルやツール

大学や研究機関では、ある問題領域を選定し、いくつかのES構築シェルやツールを並べ、実際にプロトタイプを開発しての比較検討がなされてきた。

しかしビジネス環境において、そのような検討はあまりなされなくなった。これには、比較する環境が得にくいことや、比較による選定の効果を評価することが難しいこともあるが、それよりもビジネスへの導入成否は、厳密なツール選択には依存しないと認識されるようになったためである。システム監査ESにこのことがあてはまるか否かは、ほぼ構築者がビジネスとどう関わっているかによるだろう。すなわち大学等のように研究として続けて行くか、または実務で使うことを念頭に置くかである。システム監査ESがシステム監査人と同等のアウトプットを出せない以上、どの構築シェルやツールにも何らかの長所短所がある。様々な試行から出た構築シェルへの要求は領域シェルへの要件として形成されるだろう。

3.3 今後の動向 — 米国大手会計事務所の動向 —

(1) 全体動向

ESは、システム監査効率と、さらにはシステム監査に従事している平均的な専門家のレベルをも向上させ、サービス拡大を目的に、様々な形態のものが試用されている。幾つかのシステム監査ESは構築が完了したり、まもなく完了予定にあると報告されている(表付3-1)。

表付3-1 米国大手会計事務所が開発したエキスパートシステム

項番	会 計 事 務 所	エキスパートシステム
1	Arthur Andersen & Co.	ICEVAL
2	"	FEAP
3	Arthur Young	AYASQ
4	Coopers & Lybrand	PREAUDIT
5	Deloitte, Haskins & Sells	CONTROL PLAN
6	"	STAR
7	"	AUDITPLUS
8	Peat, Marwick, Mitchell	SEADOC
9	"	SEACAS

現在はまだ発展の初期であるが、効果的なシステム監査技術の必要性からESとDSS（意思決定支援システム：Decision Support System）は緊密に結合され、1990年の初めまでには米国のシステム監査にとって必要なものとなり、あと10年もすればシステム監査人があたり前に使うものになるとみなされている。

現在使われている作業手順作成やリスク分析ESは、更に強力にシステム監査人を支援し、ますますその能力を発揮すると思われる。これらの現在あるものや開発中のものを含め、個々のESは、より大きなESに組み込まれてゆくだろう。2つの会計事務所が、それまで自社で開発してきたモジュールとESの統合を行っている。アーサー・アンダーセンは、FEAP（Front-End Analysis & Planning）の構築を行っている。これは、すべてのシステム監査フェーズの包含を目標に、監査ニーズ把握から、意見形成の際のリスク分析まで行うことを試みている。FEAPのいくつかのモジュールが、1987年現在、フィールドテスト下にある。この中には、AION社の監査計画用シェルも含まれている。FEAPは、システム監査人のワーキングスペースとして使われ、プロトタイプは、1990年までに完成の予定である。デロイト・ハスキンス&セルズは、プリンストン大学に別の研究施設を持ち、20人のフルタイムのスタッフにより、AUDITPLUSの開発を行っている。これも、システム監査作業の初期から終わりまで一貫して支援し、システム監査人がこれまで行ってきた作業を自然に置き換えられるよう配慮されて設計されている。いくつかのモジュールが1987年現在フィールドテスト下にある。

大手会計事務所は、ESの開発に膨大な投資を行い、一部の大学研究者も連携して全力を投入している。両者とも、このような新技術の追求により、すばらしい恩恵が得られると信じている。ESはシステム監査に多大なメリットをもたらすが、この影響を認識しないシステム監査人は、高度技術化し競争化するシステム監査環境で不利な立場に立たされる可能性がある。

以下では、システム監査の各段階毎に、より具体的に動向を述べる。

(2) 計画段階のES開発動向

詳細な機能を持つESが、すでに使われ始めている。このような機能においては、会計監査で使われているESとの共通部分が基盤をなしている。

プロポーザル作成では、次に示すような多くの機能が、ESにより支援されている。

- ・ 自社のスタッフ、アシスタントとして依頼する会社の列挙（プロポーザル作成のために必要なスタッフ）及び接触方法の提示
- ・ 状況が類似した最近のプロポーザルのコピーの入手方法提示
- ・ プロポーザルが受け入れられた場合のスタッフに要求される能力と、必要な訓練の優先

度の提示

- ・時間レートの様々な割引に基づく契約チャンスの相談
- ・当該顧客と、他の仕事をする可能性
(会計監査、システム・コンサルティング、税務、企業買収等)
- ・オフィスとしての費用と利益についての相談
- ・利益の受取分と増加の可能性に関するドラフト・プロポーザル作成
(条件によりプロポーザルを提出しない場合の指示を含む)

(3) 予備調査段階のES開発動向

デロイト・ハスキンス&セルズのCONTROL PLANと、アーサー・アンダーセンのICEVAL (Internal Control Evaluation System) は、内部統制を評価し、適切な準拠性テストを提案する。また、アーサー・ヤングのAYASQ (Authur Young-Audit Smarter Quicker) は、準拠性テスト結果を、実証性テスト結果と比較し、すべての領域が十分テストされたかどうか診断を行う。さらに実効的でかつ理論に裏打ちされたESにより適用対象が拡がりつつある。

これらの多くは、内部統制フローチャートを画面对応で作成、更新できる。システム監査人の入力したチェックリストの答により、作成されたり、変更にとまなう余分なチャートや、不適当になった箇所の更新を行う。

また、対象領域についてのコメントを出力し、適切な監査技法を選択するのに役立つ。

(4) 本調査段階のES開発動向

クーパーズ&ライブランドのPREAUDIT、アーサー・ヤングのAYASQ、ビート・マーウィック・ミッチェルのSEACAS等は、様々なタイプの分析結果を解釈する。デロイト・ハスキンス&セルズのSTARは、重要性、信頼性に関し、入力されたデータを基に、複雑な回帰分析を実行し、どの領域にシステム監査人は注意すべきかについて支援情報を出力する。これらは、まだ、狭い範囲の機能したもたないが、近い将来には様々な分析を結合し、監査プログラムの出力に対し、より強力な推論が実施されるようになる。

参 考 文 献

- 1) William F.Messier,Jr.and James V.Hansen: Expert Systems in Auditing:The State of the Art (1987)
- 2) James V.Hansen and William F.Messier,Jr.: A Preliminary Investigation of EDP-XPert (1986)
- 3) Christopher W. Dungan: A Model of an Audit Judgment in the Form of an Expert System (1983)
- 4) James V.Hansen and William F.Messier,Jr.: A Knowledge-based expert system for auditing advanced computer systems (1986)
- 5) B.J.Garner & E.Tsui: Recent Advances In Computer Audit Research(1985)
- 6) Andrew D.Bailey,Jr.,Gordon Leon Duke,James Gerlach,Chen-en Ko,Rayman D.Meservy,and Andrew B.Whinston:TICOM and the Analysis of Internal Controls (1985)
- 7) Coopers & Lybrand:Tomorrow's Expert Systems (1987)
- 8) Aland C.Schultz:Artificial Intelligence And Security:An Overview (1987)
- 9) William T.Tener: AI:Using Expert Systems for Data Security (1987)
- 10) H.Perrin Garsombke and Larry M.Parker:Decision Support Systems and Expert Systems:Auditing in the Information Age(1987)
- 11) Chris W.Dungan and John S.Chandlers: Auditor: a microcomputer-based expert system to support auditors in the field (1985)
- 12) Susan A.Sherer and Erick K.Clemons: Software risk assessment(1987)
- 13) Frederick G. Tompkins: An Overview of Risk Management Tools (1987)
- 14) International Security Technology,Inc.: An Introduction to IST/RAMP(1984)
- 15) OPA/PLG Associated: BDSS (Bayesian Decision Support System) (1985)
- 16) Coopers & Lybrand:ExerTAX:an expert system for corporate tax planning(1986)
- 17) MIS Training Institute/Boden Associates,Inc./Felicity Business System,Inc.: EXPERT AUDITOR MIS:User Guide (1987)
- 18) Frederick Hayes-Roth,et al.(ED):Building Expert Systems(1983)

エキスパートシステムの索引

〈ページ〉

・AOD	12, 34
・ARISC	12, 30
・ARROWSMITH-P	12, 21, 29
・Auditor	12, 31
・AUDITPLANNER	12, 21, 29
・AUDITPLUS	65, 66
・AYASQ	65, 67
・BDSS	12, 27, 29, 38
・CFILE	12, 31
・CONTROL PLAN	65, 67
・DISCOVERY	12, 35
・EDP-AUDITOR	12, 32
・EDP-XPART	12, 13, 29, 39
・EXPERT AUDITOR	12, 17, 29
・ExpertTAX	12, 32
・FEAP	65, 66
・GC-X	12, 33
・I. C. A.	12, 22, 29
・ICEVAL	65, 67
・IST/RAMP	12, 23, 29, 38
・PREAUDIT	65, 67
・RISKA	12, 38
・SEACAS	65, 67
・SEADOC	65
・STAR	65, 67
・T. R. A. M.	12, 38
・TAX ADVISOR	12, 32
・TICOM	12, 32
・TICOM III	12, 33

—— 禁 無 断 転 載 ——

昭和 63 年 3 月発行

発行所 財団法人 日本情報処理開発協会

東京都港区芝公園 3-5-8

機 械 振 興 会 館 内

TEL 03 (432) 9372

印刷所 株式会社 タ ケ ミ 印 刷

東京都千代田区神田司町2-16

TEL 03 (254) 5840

