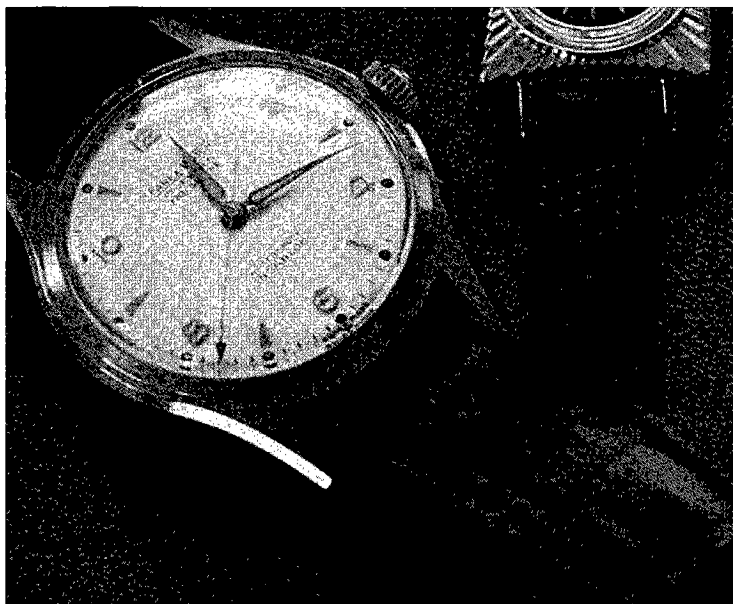




2 西暦2000年問題 の技術的課題

藤田好也

(株) 野村総合研究所

西暦2000年対応ほど大きなプロジェクトとなった場合には、システム保守担当者の日常経験の範囲では解決困難な問題が数多く発生する。本稿では最近、実務上話題になっている技術的課題を解説する。

西暦2000年問題の特徴

■一般災害と比べた西暦2000年問題の特徴

西暦2000年問題を他の一般災害と比較した場合、次のような違いがある。

(1) 発生時期が特定できる

地震・水害・火災などは、いつ発生するか特定しにくい。西暦2000年問題が発生する時期はほぼ特定されており、対応のスケジュールを立て得る。

(2) 想定する被害を特定しづらい

一般災害ならあらかじめその種類に応じて想定される被害の種類は限定できるため、停電といえば自家発電、火災といえば消火器といったように、準備すべき事柄は明らかである。しかし西暦2000年対策の障害の範囲は捉えどころがない。空振りに終わる可能性すらある。

■西暦2000年問題の想定するリスク

西暦2000年問題で生じるリスクは以下の3つに分類でき、それぞれ「事前準備」と「事後措置」を想定する。

(1) 社会インフラにかかわるリスク

電気、ガス、通信といったライフラインにかかわる障害は、企業内の情報システムはもとより、社会活動一般に大きな影響を与える。このレベルで障害が発生した場合は、個々の企業では対処不可能であるから、その責任は減殺されるであろう。もし停電などへの対処が通常の防災上の観点から不十分であるなら、むしろこれを機会に障害対策の正常化を図るという意識で対策を立てるべきである。なぜなら1カ月の停電に備えても、実際にそのような事態になれば社会は混乱し、自社のみ稼働していても意味のない状況になっているからである。業務内容にもよるが、一般的には1日程度の停電を意識する程度が妥当といえよう。

(2) 業界共通基盤にかかわるリスク

証券業界でいう証券取引所などがこれにあたる。一見、社会インフラにかかわるリスクと何ら違いはないが、他のインフラは正常に機能していることから、業界全体が責任を問われかねない。その意味で、一企業の枠を超えて、サプライチェーンや取り引きのネットワークなどの業界全体が連携した試験である「インダストリーワイドテスト」には積極的に参加し、業界としての信頼性確保に努めるべきであろう。

(3) 個別企業内でのリスク

個別企業において障害が発生した場合、その企業の責任が問われるのは間違いない。その企業の自助努力により本来解決されるべき障害だからである。したがって、個々の企業はこの領域で発生する障害への対処に全力投球すべきである。その結果が全体のリスク極小化につながる事が期待される。

■隠されたリスク

情報技術面でのリスクの背後には次のようなリスクが控えている。

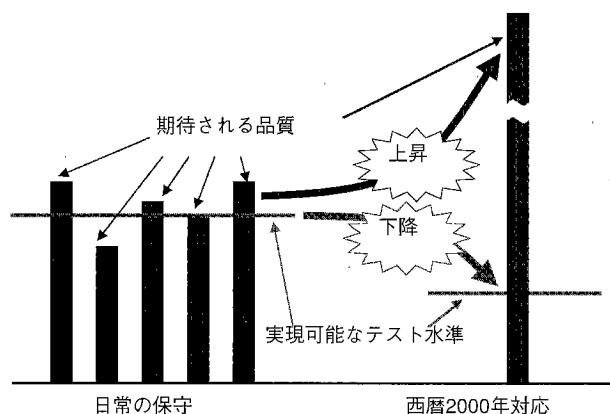


図-1 期待される品質と実現可能なテスト水準の乖離

(1) リスクヘッジが生むリスク

2000年初めの重要取引を回避する動きが出てくる可能性があるが、こうした動きの集中が新たなリスクを生み出すことにもなる。

(2) 資源配分の失敗

膨大な機器の更新需要に見合った生産能力が確保されるのかどうか疑問である。こうしたゼロサムゲーム的状况は人的資源の不足と合わせて、世界規模で資源配分の失敗を招き寄せる可能性がある。

(3) 試合放棄

対応コストと想定される損害賠償コストを比較して、後者が安上がりだと判断すれば、そちらを選択する企業が出てきてもおかしくない。

西暦2000年問題対応の技術的難しさ――

■検証精度に関する誤解とその達成の難しさ

従来の保守作業では、各システムごとの検証要求精度はほぼ一定しており、それに見合ったテスト水準に自ずから収斂されてきた。そこでは、同時に「100本リリースすれば1～2本のバグがあっても仕方ない」といった問題発覚後に対応できる範囲での合理的な割り切りが存在している。ところが西暦2000年問題の場合これでは不十分なのである。なぜなら西暦2000年問題の影響調査範囲は当該企業の全システムに及び、その1～2%といえども、バグが発生した場合には絶対量が多くなりすぎ、短期間で対応できないからである。したがって通常の10～100倍の検証精度が要求される。それに対して実現可能なテスト水準は、期間・要員・設備・予算等の不足から、通常以下に落ち込みがちで、図-1に示す、非常に危険な構造にあるといえよう。

■いつまでも安定しないプログラムパッチ

先行企業の中にはすでにアプリケーションプログラムの対応を終えているところも多いが、ソフトウェア製品のバグ対応版提供の遅れから「完了」とはならないのが現実である。

(1) パソコン用OSの最新リリース版が事実上使用できないケース

最近あるパソコン（以降PCと呼ぶ）用OSの最新版が出荷された。その中ではいくつかの西暦2000年問題関連のバグが解決されていたが、これとはまったく関係のないところで新たに重大なバグが発生したためリリースを停止せざるを得なくなった。

この場合、1つ前のバージョンをリリースし、そのバージョンで提供されている2000年問題対応パッチプログラムを当てていくという選択もある。

(2) 何度もパッチプログラムが提供されるため、何度もリリース作業を強いられる場合

繰り返しパッチプログラムを出荷しつづけているメーカーが存在するが、最終バグフィクスを待っているわけにもいけないので、次のような点に留意しつつ、リリースタイミングを見極める必要がある。

- ・当該製品についてのこれまでの西暦2000年対応パッチ数の収束状況
- ・公表される個々のバグの重大性の推移
- ・インストールすべき環境数と所要工数バランス

西暦2000年対応の組織と手法――

ほとんどの主要企業内には、西暦2000年対応を統括する部署（以降「Y2K統括部門」と呼ぶ）が設置されている。ここでは、こうしたY2K統括部門が一般に果たすべき役割とその具体的な手法を紹介する。

■全社進捗情報の収集・分析・報告

西暦2000年対応プロジェクトでは、各々のシステムの進捗状況の定期的な把握が何よりも大切である。こうした定量的なデータの裏付けなしには経営判断は不可能である。共通の書式（図-2）と基準（表-1）を用いて各担当者あるいはチームから定期的に報告を受け、さまざまな分析を加えて関係者に提供すべきである。

■西暦2000年プロジェクト監査の実施

(1) 監査の背景

ある程度対応が進んだ時点で問題になるのが、対応完了の精度である。システムごとに単位ステップあたりの対応工数に開きがあることがある。さまざまな理由が考えられるが、中には何らかの思い込みや考慮漏れがあるかもしれない。各システムで実施されてきた西暦2000年対応プロジェクトを対象に、一定の評価項目を設けてその手法や判断の妥当性を評価するとともに必要な改善策を提示する動きが求められている。

(2) 監査の目的

- ・西暦2000年対応の「質」の把握

対応進捗状況を事務的に管理しておけば、プログラム本数などの「量」的な把握はできるが、「質」を伴ったものかどうかは分からない。しかし仕上げ段階では「質」の把握は欠かせない。特に危機管理計画（contingency plan）は各システムの精度に大きく依存するため、こう

部名	締日		記入者										
生産技術部	1999.2.28		藤田好也										
システム名	担当	規模		稼働環境			対応状況	進捗 (%)	工数 (人月)	費用 (百万円)		実施スケジュール	
		値	単位	構成	OS	メーカー				人件費	設備費	着手日	完了日
Aシステム	藤田	100	KS	C/S	AA	X	完了		7	10	15	1997.1.1	1997.12.31
Bシステム	藤田	50	本	単独	BB	Y	対応中	20	3	4	5.5	1998.1.1	1998.12.31
Cシステム	藤田	100	枚	ホスト	CC	Z	調査完了・対応否		0.5	0.5	0	1998.3.15	1998.3.31

図-2 対応状況報告書(簡略版)例

した調査抜きでは実態が伴わず形骸化しがちである。

- 課題解決による対応水準の底上げ・均一化

当該システムに習熟したSEが統括するとしても、合格点を与えることができる作業項目にバラツキが出てくることは避けられない。こうした監査により明らかになった補強箇所を各プロジェクトごとに改善できれば、全体的な底上げと均質化が期待できる。

- 評価作業を通した各担当者へのノウハウ提供

評価を、「ドキュメント精査」、「ヒアリング」、「プログラムソースコード検査」の3方向から実施することで、チームリーダーや協力会社などにもさまざまな機会を通じて接することができる。そうした場で評価項目および評価結果の説明の中に、具体的な問題の解決法を盛り込み、評価担当者が持つノウハウを提供する。

(3) 監査の実施

バグが発見されるとごまかしがきかず真摯に受け止めてもらいやすいことから、実施対象はテストが完了したシステムとするのが妥当である。しかし実際に監査を実施するとすると以下のような壁が立ちはだかっている。

- 開発現場のアラさがしであることには違いなく、組織レベルでの抵抗を受けやすい。
- 監査を担当するに足る西暦2000年対応に精通した人材の確保が困難。
- プログラムソースコードを再検査するための適切な解析ツールもしくはサービスを選定し確保するノウハウの蓄積がないことが多い。
- 元来タイトな予算の中から監査コストをまかなうことが難しく、また被監査部門が費用を負担するのでは厳正な監査が困難。

■各開発部門の技術サポート

西暦2000年対応はシステム全体に波及するため、通常のシステム保守と同じ手法に固執してはスムーズに対応を進めることは難しい。Y2K統括部門は次のような技術を蓄積・整理し、適宜開発部門に提供する必要がある。

(1) プロジェクト計画

西暦2000年対応では、一般の保守プロジェクトとは少し異なる手法をとる場合がある。スケジュールリングを例にとれば以下のような特徴がある。

- パイロット調査の実施

最適な影響調査・修正・検証方式を見極め、工数見積りの妥当性を確認する。

段階	解 説
未着手	何も手をつけていない状況
調査中	対応要否の判定作業中
調査完了	対応要否の判定作業が完了した状態。この状態の場合、「対応要・対応否」を明記する。 例。調査完了・対応否 ここで対応否であれば対応が完了したものと同等の扱いとなり、対応要であれば対応計画立案以下のプロセスをたどることとなる。
計画中	対応計画立案中
計画完了	対応計画が完了した状態で停止中
対応中	修正箇所特定・修正・テスト・リリースといった対応作業中 この段階にあるものについては、厳密なものでもなくとも進捗率を記入する。
仮完了	下記のような特別な作業を残して、担当者として現時点で実施可能な作業をすべてやり終えた状態。 例1. 作業が遅れている他のシステムとの疎通確認 例2. 自分が責任を持つアプリケーション側に影響がないことを確認できている状態での他の担当部署によるベース系ソフト・ハードの入れ替え 例3. ユーザへの利用方法説明
完了	西暦2000年前後にしかできない作業を残し、西暦2000年(それ以前に影響が想定される場合はその日)を待つだけの状態。 例1. 1999年12月31日等の特別運用 例2. 2000年1月4日等の出力内容チェック作業

表-1 対応状況記入基準

- 優先順位の付与

対象範囲が広い場合、サブシステムやプログラムごとに対応に優先順位をつけたり検証精度に差をつけざるを得ないことが多い場合、あらかじめ順位を明確にしておく。

- サブシステムごとの短期サイクル開発

大規模システムでは、10万ステップ程度を単位に分割して影響調査からリリースまでを複数のサイクルに分けて繰り返し実施する。全体を一括して開発すると長期にわたり開発中のまま引きずることになり、他の保守案件による修正との競合が発生したり、工数の読み間違いを初期に補正することが難しくなったりする。

(2) 影響調査・修正・検証手法

まず「桁拡張」と「ウィンドウイング」両手法を中心に各修正方式の特徴を整理しておく。また修正方式ごとに「波及分析」、「文字列サーチ」、「実データ解析」などを組み合わせ、各システムに最適の影響調査手法およびテスト項目を提案できるようにする。

(3) サポートツールとサービス

西暦2000年対応をサポートするツールやサービスは数多く提供されているが、その評価を個々の開発部門が実施することは容易ではない。そこでこれら进行评估し、良

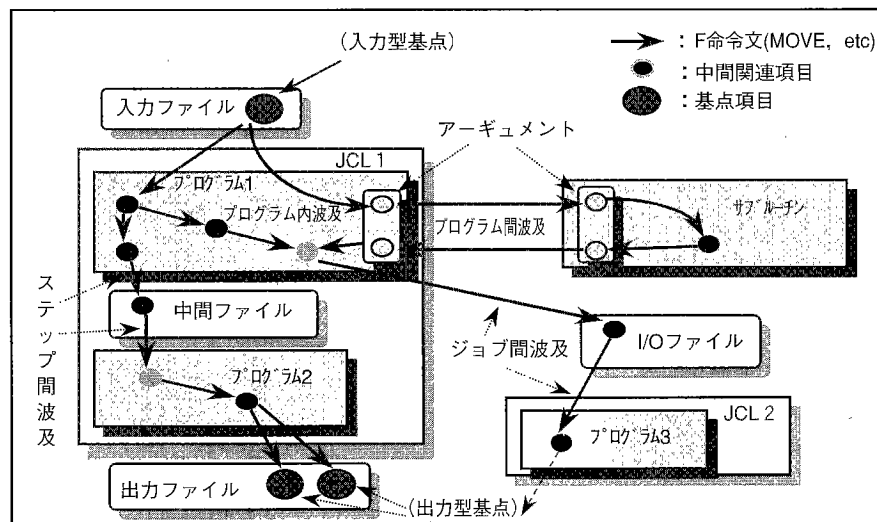


図-3 波及分析概念図

いものは社内に推奨する。

■各開発部門の専門外ノウハウサポート

西暦2000年対応では通常のシステム保守とは異なり、法務・税務・経営リスク分析といった、これまではシステム保守担当SEが関与する機会の少なかった分野も視野に入れなければならない。特に顧客および仕入先との接触では、普段の信頼関係に加えて、法的な裏付けが強く求められており、Y2K統括部門は次のようなサポートを提供すべきである。

- ・新規契約内容の確認および既存契約内容の再確認
- ・社外向け発信情報の内容チェック
- ・社外からの問合せへの回答文書内容の検討
- ・取引先とのミーティングへの同伴

■その他の活動

(1) 情宣活動

「各種レポート作成」、「ホームページ運営」、「全社連絡会議の主催」、「取材応対」、「執筆・投稿・講演」など共通の情報収集および発信を行う。

(2) 共通課題の解決・調整

「危機管理計画の策定」など部門単独では調整しきれない課題を引き受ける。

分析・検証技術

西暦2000年対応のサポート技術の中心である「影響調査でのプログラム波及分析」と「検証方式決定の考え方」およびプログラム波及分析の応用としての「インスペクション」の3つを紹介する。

■影響調査工程における波及分析

(1) プログラム波及分析とは

プログラム中の特定の変数を基点に、代入・演算などさまざまな命令で関係する相手方の変数をイモヅル式に洗い出して影響箇所を特定していく手法である。メインフレームのプログラムを例として図-3に方法を示す。

(2) 波及分析ツールに求められる機能・性能

・基点設定サポート機能

特定の日付項目（基点）を指定すればそれに関連する項目・命令をイモヅル式に洗い出す機能を持つ波及分析ツールであるが、最初の指定は人間が担当せざるを得ない。そこで基点の洗い出しをサポートし、半自動化する機能が求められている。

・人的判断の誤り発見機能

波及分析は、波及分析ツールによる自動解析機能と人的判断が融合し相互補完することで初めて効果を発揮する作業であり、人的判断の誤りは避けられない。そこで、こうしたミスを自動的にチェックする機能が求められる。

・波及項目への属性付与機能

波及分析の結果ははそのままでは単に「日付に関連している項目の集合体」にしかすぎない。この中には、和暦日付・日数・利息額など雑多な項目が含まれているが、実際に修正が必要なものは西暦年2桁項目のみである。そこで洗い出した結果に「西暦年2桁」、「和暦」といった属性を付与して、本当に必要な項目と単なる参照項目とを判別する機能が必要となる。

・過剰波及に対する機能上の配慮

当該ツールの死命を制する重要なポイントとなるのが過剰波及の防止である。波及分析ツールは関係すると思われる項目を漏れなく洗い出す反面、人間なら明らかに関係ないと判断できる項目まで洗い出してしまふ。はなはだしい場合は、解析処理が終了しない事態も発生する。

・その他

「利用方法論の確立」、「証憑保存機能」、「操作性」などの評価項目がある。

(3) プログラム波及分析ツールの現況

波及分析は影響調査の工程で使用されることを想定して開発された技術であることから、西暦2000年対応においては、その役割を終えつつあると考えられている。しかし商品化した多くのベンダにおいて現時点に至っても「不

完全燃焼」感があり、この分野での米国業界の活況振りと比べるとなおさらであろう。こうした状況を生み出した原因は以下のように考えている。

● 本来完全自動化が困難な分野であること

波及基点の設定では人為的な判断・作業を要するなど、本質的に完全自動化に馴染みにくい性質を持っている。そのため、利用ノウハウを整理しユーザを教育する時間と費用が必要になり、大きなプロジェクトでないと割に合わない。また、担当者ごとに生産性や精度が大きく異なることから、使用しない場合に比べて明確な差が得られるほどの効果を定量的に示し得なかった。

● 和暦の存在があったこと

日本では年2桁項目から和暦を排除しなければならないため、波及分析の仕組みに欧米と大きな違いがでてくる。ところが海外から輸入された波及分析ツールでは、当該ツールの基本思想を崩さない範囲での応急措置はあるものの完璧とはいえず、和暦と西暦が入り交じったシステムには適さないことが多い。また一方、当初から和暦を想定して設計していても、和暦か西暦かの判断は入力データの内容に依存するため、プログラムソースを追跡するという波及分析ツールの守備範囲では対応しきれず、人的判断に委ねねばならない。

● 技術者のミスマッチ

解析対象はメインフレームのプログラムソースでありながら、コスト・操作性・処理速度といった面から、WSまたはPC上にツールを構築せざるを得ないという構造的な足かせが存在する。つまり、「メインフレーム上の言語解析および保守作業実務に習熟」し、かつ「オープン系でのツール開発に長けた」技術者が必要である。しかしそうしたバイリンガルな技術者の絶対数が少なく、高い完成度を獲得する前にリリースせざるを得なかったという背景があった。

● ユーザ側の盛り上がりが遅かったこと

我が国において西暦2000年対応を意識して波及分析ツールの販売が本格化したのが1996年であった。当時は西暦2000年問題に対するユーザ側の危機意識が低く、こうしたツールに対する投資が認めてもらいにくかった。またユーザ側経営者が意識を持ちはじめた時点では、影響調査という比較的前半の工程は、ユーザシステム部門が力仕事でやり終えていたというのが実状であろう。

(4) プログラム波及分析ツールの今後

一見、すでに役割を終えつつあるかのように見える波及分析ツールであるが、西暦2000年対応も終盤になって後述する「インスペクション」で利用する方法が注目を集めつつある。またこれまでのプログラムソースの解析に、本番で使用している実データの値の解析を組み合わせ、プログラムのコントロールフロー解析とデータフロー解析の精度を向上させる手法も検討すべきであろう。

■ 検証方式

(1) 完了判定と検証のプロセス

検証作業の完了に至るプロセスを図-4に示す。

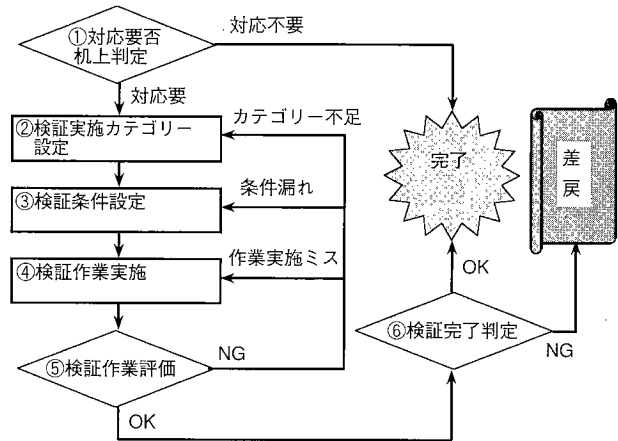


図-4 完了判定フロー

(2) 対応要否机上判定

実作業に着手する前に、中長期システム化計画・現行システム関連証憑・属人的ノウハウ・サンプル調査結果などを根拠に、机上で西暦2000年対応の要否を判定する。この際「再構築予定」、「廃止予定」といった理由は、中長期計画の狂いによりもくろみが破綻する可能性があることに注意すべきである。

(3) 検証実施カテゴリ設定

表-2のカテゴリのうち実施するものを決定する。場合によってはすべて実施しないという判断もある。実際には、表-2のいくつかのカテゴリを組み合わせることが多い。

(4) 検証条件設定

「連結テスト」を例にとれば表-3のテスト条件のうちどれを採用するか決定する。他のテストについても各々の特性を勘案して、こうしたテスト条件を設定する必要がある。

(5) 検証作業実施

対象カテゴリについて、設定した条件に従って検証を実施する。この際、以下の点に留意すべき。

- タイムボックス手法を採用する。
- 手戻りをスケジュールに織り込む。
- テストデータ生成にかかる負荷を事前に実地検証する。

(6) 検証作業評価

検証実施の結果を評価し、検証内容が適切であったかどうかを判定する。「条件漏れ」などは実施して初めて気付くこともある。

(7) 検証完了判定

検証実施の結果、当該システムの西暦2000年対応は完了したと認めてよいかどうかを判定する。検証の結果、不良個所が発見された場合は、影響調査・修正作業を再度実施する。

■ インスペクション

米国で、西暦2000年対応への「インスペクション」¹⁾の適用が最近注目されているが、日本国内ではまだ認識が不十分である。ここでは、インスペクションの必要

カテゴリ	解説	実施負荷 (*1)
単体テスト	アプリケーションシステム中の修正を加えた個所につき当該修正にかかわる条件のみを検証する。 [テストの主目的] 修正ミスの洗い出し。	小
連結テスト	影響調査を実施したシステム全体を対象に、一般的なチェック条件に沿って検証する。 [テストの主目的] 影響調査漏れの洗い出し。	大
Inspection	対応済みのシステムを別の者が再度チェックする(テストは行わない)。通常は波及分析ツールを用いる。	小(*2)
基盤総合テスト	西暦2000年対応で入れ替えたハードウェア・基盤ソフトウェア全体の組合せをチェックする。	中
対外接続テスト	社外システムとの接続を確認する。	小

(*1) 対応に要する「工数」「費用」「期間」を勘案し、定性的に判断したものを参考までに提示した。

(*2) 波及分析ツールを使用した半自動作業を前提とする。

表-2 検証実施カテゴリ

テスト条件	解説	実効性 (*1)
1900年代(デグレード)テスト	修正済システムが、現在の日付データで現行システムと同様に問題なく移動することを確認する。	低
1900年代・2000年代混在テスト	最も重要な条件。1900年代の日付データと2000年代の日付データが混在した状況で正しく移動することを確認する。	高
2000年代テスト	1900年代の日付データがすべて姿を消した状況で正しく移動することを確認する。	低
閏年関連処理確認テスト	2000年2月29日を意識した条件。 例) 2月から3月にかけての日数計算。 2000年2月の月末日算出。	中
サイクル処理テスト	日次・月次の処理を数サイクル回してその整合性を確認する。 通常は、「1999年12月から2000年1月」「2000年2月から2000年3月」の2つの時期を設定する。	中
連続運転テスト	「1999年12月31日から2000年1月1日」「2000年2月28日から2000年3月1日」の2つの時期について、システムを立ち上げたままの状態で正常に移動することを確認する。	中
異常データ処理確認テスト	西暦2000年対応修正の前提としている様式に従わない異常な日付データが入ってきた場合に、正しい措置がとられているかどうか確認する。	低

(*1) 連結テスト実施にあたっての検証効果の大きさを定性的に表現したもの。

すべての条件が実施できない場合を前提にあえて提示した。

表-3 連結テスト検証条件

性と波及分析技術との関係について述べる。

(1) インスペクションとは

・インスペクションの必要性

大規模なプロジェクトになると、すべてのプログラムに対し、最も重要なシステムと同等の厳密さでテストを行うことが不可能な場合も出てくる。そこで「①テストを実施するよりは低い負荷」で、「②テストを実施した場合に準じた検証効果」が得られる第三の方法が必要となる。

・インスペクションの意味

その第三の方法として、ここでは「インスペクション」を紹介したい。インスペクションとはその字義通り「監査」といった意味で、一度修正を施した西暦2000年対応済プログラムを第三者の目で再度精査し、その妥当性を検証するものである。もちろんその作業は、専任SEが一定の方法論に基づき、しかるべき波及分析ツールを使用して実施するものでなければならない。

(2) インスペクションと波及分析技術

波及分析技術が重要な要素となることは影響調査とはほぼ同様といえるが、以下の点で少し違った考え方も要求される。

・影響調査結果の厳密性の緩和

実は、インスペクションでは「100%の調査精度」は必須ではない。影響調査と異なり、すでにリリースされたプログラムを第三者の目で再度チェックすることで、より精度の高い検証を行うことが目的であることから、「100%に近い精度」であれば必要条件は満たしたことになる。実際の影響調査が「当該システムの仕様には詳しいが、西暦2000年対応が専門ではないSE」によって行わ

れるのに対して、インスペクションは「当該システムの仕様は詳しくないが、一定の方法論・ツールには習熟した西暦2000年対応専任SE」が行う。異なる視点でしか発見できない個所が存在するので、インスペクションにより精度の向上が期待できる。

・修正ミスの検出

影響調査工程の波及分析ツールでは、洗い出し漏れの多寡がその評価の分かれ目であった。しかしインスペクション工程では、修正およびテストを終えたソースをチェックすることから、「洗い出し漏れの検出」に加えて「修正ミスの検出」ができる効果もある。

・スピード

多くのプロジェクトではテスト完了後にこうしたインスペクションを行うことを想定していないためリリース期限が真近いことが多い。したがって、「100%の精度」よりも「95%の精度とスピード」が要求される。波及分析ツールを使用した場合、人手で時間をかけざるを得ないのが、まさにこの残された5%の部分であることから、目標を95%に落とすと自動化の効果が大きく引き出されてくる。また「工場」的な組織があればより効率的である。

・低コスト

テストに時間をかけるのに比べてコストパフォーマンスが良いからこそインスペクションを選択するのであるから、コスト面の評価は厳しくなる傾向がある。

参考文献

- 1) Fagan, M.E.: Design and Code Inspection to Reduce Errors in Program Development, IBM Systems Journal, Vol.15, No.3, pp. 182-211 (1976).

(平成11年4月8日受付)