2018.6 のブログの別紙

「IT 事件史 1999 年：「Ｙ２Ｋ」の緊張走る」を読んで

【本ページの参照元：2018.6 の下記のブログ】
「IT 事件史 1999 年：「Ｙ２Ｋ」の緊張走る」を読んで
（ http://www.1968start.com/M/blog/index.html#1806d ）

日経コンピュータ（2018.6.7）の記事
「IT 事件史 1999 年：「Ｙ２Ｋ」の緊張走る」にコメントする。

＜記事の要約＞
・Ｙ２Ｋ（2000 年問題）とは、2000 年になったとき、西暦の下 2 桁だけを管理するプログラムは「1900 年」か「2000 年」かを判別できずに誤動作する可能性があり、年が変わった瞬間に電力やガス、交通機関などの社会インフラ、銀行の勘定系システムや決済ネットワーク、電話や通信のネットワークなどに大きな事故や混乱が生じるかもしれないという問題だった。

・国内では JR や私鉄各社が 2000 年を迎えるタイミングで列車を最寄り駅に停車させる特別ダイヤを組んだ。航空会社の一部は年をまたぐフライトの欠航やスケジュールの変更を決めた。政府は年末年始に内閣と関係省庁でピーク時 2000 人の危機管理体制を敷いた。

・問題の解消に向けて、企業は早い段階から基幹システムの刷新やプログラムの改修に取り組んだ。IT 業界は 2000 年問題対応一色となった。

・2000 年問題は「ＩＴは社会を支えるインフラである」と一般の人に広く認識させる機会だった。

■本件は、「ソフトウェア工学・演習」の授業で取り上げ、以下の教科書にも記述した。
・ソフトウェア工学（第２版）（朝倉書店 2004 年３月刊）
（ http://www.1968start.com/M/lecture/SEindex.html ）

【記述箇所１】
［第Ｉ編 ソフトウェアの動向］の［2 章 ソフトウェア危機の歴史］の［2.4 節 質の問題］p.11：

「信頼性の問題は，社会的に重要な役割を担う情報システムに関するものである．1990 年代の最後に発生した西暦 2000 年問題は，図らずもコンピュータが社会のいたるところに使用されていることを如実に示すこととなった.」

【記述箇所２】
同上の［2.6 節 ソフトウェア生産技術の課題］ p.13：

「　情報化社会に対応して，これらの規模と量と質とインタフェースに関するソフトウェア生産技

術への要求は，ますます増大している．それを象徴する出来事が，20 世紀末の西暦 2000 年問題であったといえる．この問題は第一義的には質の問題であった．その故障によって生じる社会的影響の大きな情報システムが多数存在していることを顕在化させる結果となった．次に，短期間に対応すべき古いシステムが多数存在するにもかかわらず，その保守に対応できるエンジニアの数には限りがあるという意味では，量の問題でもあった．また，10 年以上前に開発されたメガステップオーダの大規模ソフトウェアの中から，まともな保守書がないまま変更箇所を探し出し，適切な修正を加える作業は，規模の問題でもあった．さらに，ネットワークを介した電子商取引の時代には，1 企業の誤った送信データが他の企業や他国の情報システムを混乱させる可能性をはらんでいたという意味では，インタフェースの問題であった．」

【記述箇所 3 】
［第 IV 編 ソフトウェアのパラダイム］の［14 章 プログラミングパラダイム］の［14.1 節 パラダイム論］の［14.1.3 項 西暦 2000 年問題の本質］ p．144：

「パラダイムシフトの必要性を痛感させる象徴的な出来事が西暦 2000 年問題であった．ハードウェアが高価であった時代に，メモリ節約のために西暦を下 2 桁で処理していたため，西暦 2000 年になったときに，1900 年と間違えてコンピュータシステムが誤動作するのではないかという問題が，大きな社会問題となった．

この問題は，出荷時には不良でなかったものが，環境の変化に伴って不良となった典型的な例といえる．アプリケーションの寿命が開発者の予想を越えていたという誤算よりも，ソフトウェア誕生から半世紀を経てなおまともな保守技術が確立していなかったことのほうが，大きな誤算といえるかもしれない．ソフトウェア工学の観点からは，次のような課題を指摘できる．

・技術的課題：ソフトウェアの保守技術（つくったのに直せない？）
・社会的課題：ソフトウェアの品質保証（バグはあって当たり前？）
・歴史的課題：プログラミングパラダイム（どんなつくり方をしたの？）

まず第 1 に，既存のプログラムを変更するという観点では，典型的な保守の問題であり，日常的業務の 1 つにすぎないが，西暦 2000 年問題は，いくつかの問題が重なって解決を難しくしてきた．すなわち，膨大なデータベースの変更と関連処理部分の変更に関して，長期間使用されてきた現行システムの機能を正しく反映した仕様書がないことによるプログラム変更個所の特定の困難性，およびその変更が副作用を引き起こさないことを確認するテストの困難性などである．

第 2 に，ソフトウェアの故障によって生じる社会的影響の大きな情報システムが多数存在しているため，ソフトウェアの実態を知らない一般の人たちには，ブラックボックス的恐怖からのパニック現象もみられた．ソフトウェアの信頼性が損なわれるという意味では，第一義的には質の問題であったが，膨大な数の対象システムに対応できるエンジニアの数が限られているという点では，量の問題でもあった．

第 3 には，上記の保守や品質保証の困難性の根本的原因として，プログラミングパラダイムの問題があった．西暦 2000 年問題は，半世紀以上続いた手続き型のプログラミングパラダイムが，世紀末の断末魔の苦しみにあえいでいる姿であったといえる．」

【記述箇所 4 】
［あとがき ― コンピュータはなぜ間違えるか？ ―］ p．173

「前世紀末の西暦 2000 年問題も新世紀初頭（2002 年 4 月）の某銀行の情報システム障害も危機的状況を思わせるには十分であったが，ソフトウェア革命はまだ達成されてはいない．」

■授業：「ソフトウェア工学・演習」の電子レポート課題
（ http://www.1968start.com/M/lecture/edebate.html ）

● 課題（98.11.25）
西暦２０００年問題について自分の意見を述べよ．

［配布資料］朝日新聞朝刊（98.10.17）の 4 面記事「一からわかる 2000 年問題」
［提出方法］タイトル、要旨（３行以内）、本文（２０行以内）、参考文献を含むレポートを電子メールで提出。

● 結果
全体にしっかりした意見が述べられたレポートが多かった。このテーマについても電子ディベートをやりたかったが、時間の都合で断念した。評価結果は、
A（特に優れているもの）１０件、
B（一応、現状調査あるいは自分の意見のあるもの）４７件、
C（感想レベルのもの）１６件。
内容は以下の項目に分けてホームページに掲載した。評価結果については、Ａレベルの１０件についてのみ、タイトル、要旨、内容（現状調査、問題の明確化、自分の意見）、参考文献についてコメントしたものを公表した。
(1) まとめ
(2) タイトル一覧
(3) 参考文献一覧
(4) 要旨一覧
(5) ベスト１０とコメント
(6) レポート一覧（全メールの公開）
レポートの書き方についても注意した。タイトルについては、内容を表現していない「西暦 2000 年問題について」の類いが２０件以上あった。要旨については、「。。。について書いています」という目次的なものが多かった。参考文献については、延べ５７件で、そのうちインターネットのホームページが３９件あった。
主な論点として、以下の項目をとりあげて、解説した。
(a) 技術的課題：ソフトウェアの保守技術
(b) 社会的課題：システムの品質保証
(c) 歴史的課題：プログラミングパラダイム

以上