

## エンドユーザコンピューティングシステムの開発と監査の方法について

6R-7

伊藤秀基 渡辺隆弘 小川清

大同工業大学 名古屋市工業研究所

## 1. はじめに

ソフトウェア生産に関する監査制度は、必ずしも定着していなかった。システム監査基準は概括的で、現実の生産の方式、生産の規模などに応じた適切な技術上の課題と、経済上の課題、組織の課題と人間の問題とに対応できてこなかった。品質管理システムのソフトウェアへの適応や、ソフトウェアプロセスの評価では、社会的システムの評価のために、改良を目的とするシステム監査について考察した。

---

Method of development and audit for enduser computing

Hideki Itou, Takahiro Watanabe, (1)

Kiyoshi Ogawa (2)

Daido Insutitute of Technology, (1)

Nagoya Municipal Industrial Research Institute(2)

2-21 Daido-cho, Minami, Nagoya, Aichi 457 Japan

3-4-41 Rokuban, Atsuta, Nagoya, Aichi 456 Japan

## 2. エンドユーザコンピューティングの開発

エンドユーザコンピューティングにおいては、システムは常に変動する。ドキュメントも残されないこともある。しかし、ソフトウェア自体がドキュメントであることを考えれば、履歴を持っていけばよい。ただし、これでは何故そう改良したかの理由は残らない。

また、エンドユーザによる改善は、その本人に取っては意味があっても、他の人が利用するには、大幅な作り込みが必要となることも多い。さらには、思いつきに基づく作り込みにより、頭でっかちのシステムとなることもある。

## 3. 改良の理由の履歴

改良の履歴を持つための方法として改良の目的を分類した。

1. 自動化する
2. 速くする
3. 作業の追加
4. 作業の削除
5. 作業の変更
6. バグの回避

7.見栄えに凝る

8.自分用のカスタマイズ

これらの分類コードをドキュメントに埋め込むことにより、簡易に変更の理由とする。このコード体系は汎用的なものであり、具体的なシステムごとにさらに細分化する必要がある。

#### 4.エンドユーザコンピューティングの監査

##### 4-1.変更数

エンドユーザコンピューティングにおいては、不要な改良が多かったりコンピュータを使うことが自己目的化することがしばしばある。また、使いにくいシステムを変更するのが面倒であるため、泣く泣く使っていることもある。この変更のしすぎと、変更がないという二つの状態をまずチェックする。

##### 4-2.変更目的

変更時の分類による度数分布により、どの部門はどのような変更目的が多いかを、仕事の変化や作業量と対比させて分析する。見栄えの改善になっていない見栄えの変更はしばしば無駄な時間に費やされる。

##### 4-3.作業時間

システムのバグに気づかず、自分の操作ミスや、ソフトウェアの作り方の問題だと思いこん

で時間を浪費することも多い。そのため作業時間を正直に報告されなければ、システムの改善にはつながらない。

##### 4-4 終わりのない改良

システムの最終目標は変動するため、終わりのない改良が続けられるが、その改良のための作業が、改良によって得られる便益とうまくかみ合っているかどうかが問題となる。

##### 4-5 ボトムアップとトップダウン

また、一人の改良が、他の利用者にとって改良となるかどうかを判断する必要がある。そのため、ボトムアップとトップダウンのフィードバックによりチェックする必要がある。ここでは、ボトムアップを開発、トップダウンを監査と位置づけたが、その逆であってもよい。

#### 5.課題

エンドユーザコンピューティングシステムの監査の問題点は、社会的システムにおける監査と同様、事実の認定が正確に行えないことが問題となる。そこで、システムの監査と改良を同時に行うことにより、スパイラル的にシステムを改良していく過程を事実として重複すると考える必要がある。ここで検討した方法は不正の発見といういみでの監査を具体化することにはなっていないことが課題として残る。