

第51号

平成5年12月

© 1993

(株)システムクリエイツ

## SCだより

編集 発行人

清水吉男

(株)システムクリエイツ

横浜市緑区中山町 869-9

電話 045-933-0379

FAX 045-931-9202

## プロセスレベルの改善 24



## 定義されたプロセス (8)

回はテストの計画と項目の設定について触れたので、今回はテストの実施に伴うバグの報告の在り方とテスト結果を分類することの意味について説明します。

## テストの報告



ここで言う『テスト』とは、検査部門で行われることを想定しています。一般に慎重に作られたプログラムは、通常の機能は概ね期待通りに動くでしょう。この段階で通常の機能が満足に結果を出さないと言うのでは、もはやここで言うテストの対象とはなりません。しかしながらどれだけ慎重に作られたプログラムであっても、執拗なテストの門を潜るまでは、どこかに欠陥が潜んでいるものと考えなければなりません。

欠陥の中には、比較的簡単に見つかるものもあれば、「欠陥」であるが由に容易に見えてこないものもあります。それでもテスト結果を詳しく、且つ注意して眺めれば、欠陥の糸口のようなものを見出すことがあります。数字の上では確かに正しい結果を出していても、その結果を出すまでの時間が、「想定していた」時間と比べて何となく遅いとか、何故かある種のデータを表示したときに、一瞬画面がちらつくとかいうことがあります。

このような状況は、ただテストをしていたのでは見逃してしまいます。特に表面的には正しいものが出ていだけに、見過ごされ安いものです。このような「情報」を見逃さないためにも、テストの手順やテストの項目を定義し、テスト結果の報告の仕方なども、そのときの状況を出るだけ豊富に捕えるように定めておくことが望まれます。そうでなければ、貴重な状況が見逃されるだけでなく、たとえ目についたとしても再現できるような形で記録されないために、重要な発見の糸口を失うこととなります。

テストの報告は、「欠陥」に対してだけ行われるものではありません。『正常』に進行したテストも、異常な状態を示したテストも、その結果を報告する必要があります。そして、ここで大事なことは、一つのテストでおかしな振る舞いをしたとしても、その度に「欠陥」として報告するのではなく、その時の状況を詳しく記録しておいて、予定のテストを一通り実施したあとで、整理分類してから報告した方が効率的でしょう。

「バグ報告書」は本来そのようにして書かれるべきものです。そうでなければテストそのもののミスや勘違いも、「バグ」として報告されてしまう危険があります。

## バグの分類



陥と判断されて「バグ報告書」にどのように書かれるかは重要な問題です。一般にハードウェアやソフトウェアの構成、その時に採用したテストデータのライブラリ番号、「欠陥」と判定した出力結果等については、殆ど漏れなく表現されます。時にはテストを実施した人の判断で、その時の「様子」やメモリーの内容などが添付されることもあります。

しかしながら、バグの重要度やバグをもたらしただ原因の検討については、ほとんど検討されることはありません。このうちバグの重要度については、テストを実施した組織が記入することになりますが、多くのソフトウェア開発現場では、しっかりした検査部門を持たないか、あっても開発部門の方が力関係が強いなどの事情によって、十分な判断がなされていません。ハンフリーはバグ重要度の分類の例として2種類の分類を紹介していますが、その内の米軍のシステムをここで紹介します。

- 重要度1：操作または本質的な機能の遂行を妨げる**  
人に危害を加える恐れがある。
- 重要度2：操作または本質的な機能の遂行に悪い影響を与える。**  
代替手段がない。
- 重要度3：操作または本質的な機能の遂行に悪い影響を与える。**  
代替手段がある。
- 重要度4：操作または本質的な機能に何らかの影響を与える。**  
利用者に不便を与える。
- 重要度5：その他のエラー**

この他に、もっと細かく分類する例もありますが、現実にはカテゴリを判断することが難しく、3～5段階程度で足りるでしょう。これに対して、バグをもたらしただ原因の検討については、バグに対応した設計部署で行われることとなりますが、書式として予め用意されていることが望まれます。

## テスト結果の分析



ここでいうテスト結果の分析とは、主に欠陥についての分析を指します。「バグ報告書」に指摘された欠陥は、単に欠陥の原因を特定し、修正されればそれで終わりというわけには行きません。単なるプログラム言語の誤解や理解不足なのか、モジュール間のインターフェースの食い違いや、ユーザーマニュアルの記入ミス、設計段階からまったく考慮されていないのか、要求仕様からそのことを読みとることは無理だった

のか、等について分析します。このようなバグに対する共通の認識がもてなければ、そこから有効な対策が講じられることは期待できません。単に「何件バグがあった」というだけの情報しか入手できないでしょうし、このような単純な「全体」の数字ではバグの終息予測にも使えません。終息を予測するためには、少なくとも『ジャングル』が存在しないことが条件です。

ジャングル・モジュールとは、直しても直しても次々と新しいバグを湧き出させ、人々の行く手を阻むモジュールで、もしジャングル・モジュールが存在する場合、発生するバグの2、3割から、時には半数近くが特定のモジュールから出ることがあります。またそのほかの特徴としては、

- 1) テストの進行につれてバグが減少しない。
- 2) 新しいテスト項目の実施によって、新しい火山が噴火する。
- 3) これらのエラーの原因が、ほとんど設計の不備に起因している。

といった傾向があります。

そして、このようなジャングル・モジュールを発見したときには、何らかの「決断」を下さなければなりません。つまり、ウォークスルーやインスペクションのやり直し、幾つかのマグマ・モジュールの組み直し、等の決断が求められます。もちろん、ほとんどの場合スケジューリングも影響を受けることになるでしょう。しかしながら、たとえそのまま作業を進めたとしても、結果的には対応する時間が確保できなかったかもしれないと、反省させられることがしばしばです。

それだけに、テスト結果を逐次分析して、このようなジャングル・モジュールを発見することが必要です。更にいえば、このようなテスト結果(欠陥)の分析は、公式なテストに入る前の、開発グループ内部で行う「結合テスト」の段階で実施されることが強く望まれます。そうすれば「決断」がもっと容易になるはずですが、残念ながら、多くのソフトウェア開発組織では、そのような段階でのテストは個人に任せられ、組織としてほとんど制御していないために、有効な対策を講じることができないようです。

(次号に続く)

## 激動する世界!



ベルリンの壁が崩壊してから世界は大きく動いている。11月にはEC統合のプログラムであるマーストリヒト条約が発効した。いろいろ問題はあがあるがヨーロッパ一大市場圏が出現した。続いて北米を市場とするNAFTAが、さらには南米各国がNAFTAに参加しようという動きもある。そうなれば南北アメリカ大陸を統合する巨大な市場となる。

その一方で、世界的な不況の中で、ヨーロッパ各国は出口を求めて闇の中を模索している。ドイツやフランスを中心に「民族主義」或いは「雇用本位主義」が台頭してきた。先日実施されたロシアの議会選挙では急進的な民族主義を唱えるグループが議会で重要な位置を占め、今後エリツィン大統領に対して悉く対立することが予想される。フジモリ大統領のペルーも、いつのまにか火種さえあれば何時でも点火する状態である。日本も方向を定め切れないでいる。

皮肉なことに「ソ連」が消滅したことでアメリカの力が相対的に弱まったのに対して、それを補おうとした国連の限界も露呈した状態で、内に籠っていた不安定要因が一気に噴き出した感がある。一体誰がこの混沌を收拾するのか。それが見えてくるまでは、この「混沌」の状態はもうしばらく続きそうである。

世界はこれまでも100年、2000年という単位でこのような状態を繰り返してきた。自動車走って周囲の景色は当時と変わっても、テレビやエアコンなど私達が手にしているものが当時と変わっても、そこにいる「人間」は200年前と同じように、傲慢であり、狂暴であり利己的であるという点では、基本的にはそれ程変わっていないのかも知れない。

# かねね 暁鐘の音

34

## ワークシェアリング

ワークシェアリングとは、不況の中で雇用の不安に対する対応策として持ち上がった考え方で、限られた仕事を少しでも多くの人と分かち合おうというものです。当然のことに収入は減ることになります。それでも失業よりもまし、ということなのでしょう。

今回、ワークシェアリングという言葉が世界を駆け巡ったのは、不況に喘ぐヨーロッパで、ドイツのVW社が労働組合との交渉の結果、週当たりの勤務日数を五日から四日(週休三日)に削減し、賃金もそれに見合うように最大で五分の四に減らすことで合意したというニュースが発端でした。同社の自動車の売り上げが、前年と比べて二五%も減少するという非常事態を受けて、労働組合に対して、更なる人員削減を実施するか勤務時間の削減かの二者択一を迫ったのでした。結局、厳しい現実を前に労働組合は実質的な賃下げを飲まざるを得なかった様です。

日本では、これより先に、経団連辺りが「賃下げ論」を展開したことがありますが、そのときは労働組合を始めマスコミから総反発をくらってしまいました。消費マインドが冷え切っているときに、収入減につながる話は「タブー」ということでしょうか。しかしながら、既に雇用調整のギリギリのところまで追い詰められている日本の経営者にとって、V

Wの労使の交渉結果は一つのヒントになったはずですが、おそろくこれを受けて、来年の春闘に向けて、「賃下げ論」を衣替えして「ワークシェアリング」と「時短」を組み合わせてくることでしよう。そしてVWの経営者と同じように、労働組合に二社択一を迫ることでしょう。雇用調整がワークシェアリングかと。

確かにワークシェアリングは、今日のように経済環境が悪化した状況に於いては一つの選択ではあります。失業に対する社会負担の増大や社会不安を考えたとき、それ程多くの選択枝はないかも知れません。特にその国の事情や、或いは資本主義の形態や性質に因っては、もはや選択枝すら存在しないことも考えられます。

「ワークシェアリング」というのは作業を分担することです。しかしながら職種によって作業をシェアリングできるものと、シェアリングしてはならないものがあります。つまりシェアリングすることで作業効率が著しく低下する作業と、それほど低下しない作業があるのです。具体的には、労働集約的な作業においては四交代勤務を五交代に移行するなど、シェアリングは可能でしょうが、ソフトウェア開発などのように、いわゆる知識



集約的(日本のソフトウェア開発は労働集約産業か?)な作業は本来シェアリングすべきではありません。もっとも日本のソフトウェア開発の現場は、皮肉にも既に十分にワークシェアリングされておいて、その結果、出



来の悪いソフトウェアが横行していると言つのに、これ以上シェアリングされては、それこそまともなものが作れなくなってしまつてしよう。しかしながら「ワークシェアリング」という考え方は、設備廃棄にと

もなう余剰人員を企業内での配置転換で吸収することに慣れている我が国にとつて、それほど抵抗なく受け入れられる可能性がありま

す。困ったときはお互い様」とか「互譲の精神」といった日本人が元来持っている考え方は、限りなく「ワークシェアリング」の考え方に近いような気がします。最も危惧されるのはこの点です。元より、今日の状況は新たな技術開発、或いは大胆なブレークスルーで突破すべきにもかかわらず、失業の危機が回避できるということで、安易にワークシェアリングを受け入れては、創造的破壊(シユンペーター)はもたらされることなく、結果として技術の停滞が生じる危険が高いのです。しかも、理工系を受験する学生の減少が止まらない現状を考えれば、間違いなく技術の停滞を加速する方に作用するものと思われる

これまで一人がかつていた業務を一人でこなそうというものです。そこにはワークシェアリングという発想はどこにも見当たりません。アメリカは明らかに「リエンジニアリング」で今回の世界不況を乗り切ろうとしている様です。それにたいしてヨーロッパは「ワークシェアリング」で対応しようとしている。この地球のなかで、全く逆の処方箋が存在しているが、これはそれぞれの地域における「資本主義」の違いも影響しているのでは

「世の中でこのクルマが一番だ、と言い切れるほど惚れ込まないとダメ。限らない愛情を注ぎ込まなければ絶対にいいクルマにならない」

杉山智之(本田技研工業・アコード開発総責任者)

ヨーロッパもアメリカもそれぞれ巨大な経済圏を確保した。そのような中で、強力な経済圏を持たない日本はどちらの処方箋に従うのか、或いは「資本主義」の性質の違いから別の処方箋を編み出すのか。結果は一、二年後には出てくることでは

## 今月一言

六歳の時から七、年も竹箆を編み続けているおはあさんが、「自分が編んだ箆は見れば分かる」という。言うまでもなく七、年間『同じ箆』を編んできたわけではない。彼女は今でも一つひとつ編みながら、どの様なものが気に入られるのか、完成をイメージしながら工夫を続けているという。ただ、機械的に編んでいるのでは無い。買ってくれる人(顧客)を想いながら、精魂を込めて編んでいる。

事をしていくだろうか。エンジニアアと言えども、サラリーマン的になつて、ただ「仕事だから」という姿勢では、ここまで自分の製品に愛情を注ぎ込めないだろう。今日ではユーザーや顧客を念頭に置くことなく作業している人が多すぎないか。組織のなかの一部門として、次の部門に引き渡すだけの、「割り当てられた作業」をこなすことに慣れてしまつと、次第に顧客の顔を忘れてしまつて、もはや、設計者の気概を感じさせない製品などには誰も振り向かない。