

ついでに

第8号

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

システム設計講座

前回はアメリカで「CASE」が普及しはじめたことを紹介しましたが、今回は「CASE」を導入するための諸条件について触れて見ます。日本でも現在CASEツールの開発が進められており、数年すれば実際に使用できるCASEツールが出てくると思われる。

アメリカでは既に七〇万人以上のアナリスト達がCASE技術を利用してソフトウェアの品質と生産性を上げているという報告が出ています。

わずか5年の間にはこれだけ普及した背景には色々あるでしょうが、日本と比べて全般にシステムアナリストのレベルが高く、また各種の分析・設計技法が考案されそして競争していることが大きいでしょう。その中で一〇〇社以上のベンダーがCASEツールを提供しているわけです。

もちろんCASEツールは魔法のツールではないので、導入すればすべからずうまくいくわけではありません。上の例は成功例でありこの他に導入に失敗した例も少なくないはず。逆に言えばそれだけ多くの企業がこのCASE技法に取り組んでいるということであり、また産業としてのレベルが高いことを示しています。

主な失敗の原因として以下の様な項目が挙げられています。

●その問題の解決に不適当なCASEツールを使用した。

●CASEツールで何でも解決すると過信する。

●CASEツールの統合が不十分である。

●開発技法や基準が用意されていない。

●ソフトウェア開発技法の教育訓練を受けていない。

●CASE技術をどう利用するかについて決断できない。

●ソフトウェアの開発と保守の効果を測定する方法を持っていない。

しかしながら最大の原因は自分達の抱えている問題を認識できていないことにある。どこに問題があるのかを認識できなければ、有効なCASEツールを選ぶことができません。

現在のソフトウェアのレベルでは万端のCASEツールは作れない。かつてCOBOLとFORTRANの両方の良いところを集めて作られたPL/Iが、その自らの大きさのために普及しなかったのと同じように、現時点では誰も巨大な万端のCASEツールを望んでいない。

このためにCASEツールは特定の分野に適した形で作られることになる。そしてそれらのCASE

ツールの拠り所として、幾つかのそれに適した分析・設計技法が採用される。

言い替えばこれらのCASEツールは、ソフトウェア関係者全員を救うべく登場したものである。という事です。丁度二〇〇Mの山に我々を運んでくれる夢のロープウェーが麓から発車するのでなく、手前にある一〇〇Mの山の頂上から発車するように、適当な分析・設計手法を身に付けていなければ、CASEツールを利用することはできないのです。

それだけ周到に準備しなければCASEツールは利用できません。しかしながらCASEツールを利用出来た場合と、全く手が出ない

場合とはは相当な開きが生じることは明らかです。米政府機関の例として手作業で二年かかるシステム仕様書を四ヶ月で作りあげたり、デューボン社では二〇〇万ドルの経費削減や、あるシステムでは従来の開発技法では二八〇三九人月であったものをわずか五人月で完成させ五〇万ドルのコストを削減した例などが報告されている。

CASEツールはCAD/CAMによく似ている。それはCAD/CAMの導入によってハードウェアの設計作業が大幅に向上したように、CASEはソフトウェアの生産性を驚くほど向上させている。しかしながらこの様なCASEツールと言えども、すでに述べたように必ずしも自分達の抱える問題のすべてを解決するものではないし、またその様なものを期待するのは「ないものねだり」でもある。

CASEに限らず、各種のツールや設計技法も同じ様に解決されない問題が残るが、それでも現状の

問題と比べたとき、波及効果等も考えてみてある程度の進展が望めると予想される時は、基本的には進むべきである。全てを満たせないからと言って進まないのは間違っている。残された問題も、実際にそこまで行つて見れば解決できる問題であるかも知れないし、その時に新しい方法が見つかることもある。何れにしろそこまで進んだ段階ではあと一歩で次のステップに進めることが出来るであろう。

『天は自らを助くる者を助く』という。着々と正しい方向に向かって準備をし続けた者だけが、この果実を手に入れることが出来るのであって、自ら木に登ることもせず、ただ手を伸ばしているだけではこの果実を手に入れることは出来ず、残念ながらこの果実は下で待っている。落ちて来ない。ソフトウェアエンジニアリングの定義者の一人であるBoehmは、その七つの原則の最後に、『工程の改善に努力しつづけること』としています。

が、彼等の能力の範囲は概ね限定されており、最大の欠点は互いの領域をカバーできないことである。したがって報道されたトラブルの原因は殆ど「つまらないこと」である。

続発する花博のトラブル

▼個々の技術では優秀であ

っても、彼等をインテグレートできる人がいない場合、結果としてこのようにチグハグなシステムができ上がる。噫乎。

▼大阪の花博会場の各種遊戯施設が何度もトラブルを起こしている。おそらくニュースとして報道された事故の他にも些細なトラブルがいくつも発生していることが予想される。報道された

事故を見る限りは、設計ミスではなく明らかにシステムの分析ミスである。あるいは「分析」という作業そのものが行われていない節もある。

▼分析作業として、このシステム

た人達が集まって考えたのであ

「時間のつくり方」について(4)

「カー・ヒルティの幸福論より」

『我々は自分のことを振り返って考えることはしない。つまり自分の幸福とか自分の進もうとする道、人生が与えてくれるもの、いかに理性的に決断して行動しているか、自分の生活信条と実際の行動の関係など本当に大切な問題について自分というものを見つめることはしない。』それでいて幸福を探し求めている。』

(アーノルド・ベネット)

七、もう一つの方法は、手早く仕事をすること、そして外形をあまり気にせず、あくまで内容に重きをおくことである。

内容が伴っていて、且つ「手早く仕上げられた仕事」が最も良く、また最も効果的だ」ということには反論の余地がない。しっかりと検討した後、短い時間に集中して湧き出すようにして書かれたプログラムが一般に完成度が高い。颯爽とプログラムを書いて見たものの、十分に検討していないために、何週間も同じようにあてもないころでもない、いじり回したプログラムには、勝手な妥協がそこらじゅうに入り込んでいいる。この様なプログラムは取り敢えず動いたとしても、正しく成長しないために、保守性は悪く色々と問題を引き起こす。

早くできる大きさに切ること(技術)が必要になってくる。プログラムのモジュール化がこれに相当する。適当な大きさのモジュールに分割することがプロジェクトの成功の大きな要因であるが、しかしながらこの適当な大きさに切することは容易ではない。今のところ彼等は経験を足場にしてこの様な分割を行っているが、経験だけに頼っていたのでは習得するのに時間がかかりすぎるのと、問題を色々な側面から眺めるためのヒントに限界がある。やはり適切な指導と試行が求められるところでしょう。

また手っ取り早く仕上げたために「粗さ」が残るかもしれないが、その辺は後で『一五分の断片』を使って整えればよいし、慣れてくれば初めから整うようになるもので、うまくやれるかどうかと言う心配を先に取る必要はない。

如何なる場合もしつかりした手順を踏んでいれば結果は自然に付いてくるのであって、行動を起こす前に結果を先取りする必要はない。八、もう一つの有効な時間節約方は、すべてのことをただ「仮に」あるいは一時的ではなく、すぐにきちんとやることである。

ここにもたいへん重要な示唆が含まれている。例えばあるモジュールで困難に遭遇し、必死になってその解決方法を探し求める際に、頭に浮かんだアイデアを適当な紙の裏に書いたまま、或は本人しか読めない字で書いたまま、あとで仕様書に纏めようと残された『もの』が、実際に仕事が終わってから、生き生きとしたまま纏められたことはほとんどない。

この場合その時点でほんの少しの努力をそこに注ぐだけでこの素晴らしいアイデアが生きた形で残すことができたはずである。

用紙の選択には時間はかからないし、文章が苦手なら図形を交えて表現すればよい。さらになぜこのアイデアを採用したのかという点について触れられておれば、そこに生き生きとした、自分の頭の中

に考えたことの『写像』が目前の紙の上に残るのである。

このようにして書かれたものが、第三者にとって「心地よい」ものであることは言うまでもないが、本人にとっても非常に有用な資料となる。

今やワープロやパソコンの普及によって、文書を作成するための各

今月の一言

「人の悪を攻むるには、太(はなは)だしくは厳なることなかれ、要(かならず)その受くるに堪うることを思うべし。」

陸湘客 (酔古堂劍掃)

はたして好んで失敗する人はいらうか。ピーターの法則に言う『創造的無能』を装うケースを除けば、本来失敗を求めて行動している人はいないだろう。この『創造的無能』ですら止むに止まらずして採る行動であって好んでとる行動ではない。

彼は一所懸命にやったが、出来てきたものは期待(?)していった物とは違ってしまったり、予定を2ヵ月もオーバーしてしまったりした。

種のツールも、必ずしも十分とは言えないが増えてきており、それらを上手に使うことで『仮』の作業を無くしていくことが出来る。そしてこの様なツールを如何にうまく使いこなすか、また適当な大きさのモジュールに分割するためを手だてをどれだけ持っているかが重要になってくる。

この様にうまく行かなかったのは、多くの場合成功に導くための方法を知らなかっただけ(?)であり、因果を心得なかっただけである。たしかに残念なことではあるが、だからと言って徹に叱責しても始まらないし、かえって意固地になるだけで何にもならない。それよりも思わざる結果を招いた原因を当人が納得する様に説明し、次回の行動につなげる様に励ました方がよい。誰も失敗を好んではないのだから。