

第175号

平成16年4月

E-mail: © 2004
shimz@mb.infoweb.ne.jp
LDG04167@nifty.ne.jp

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/FAX 045-933-0379



40回め



「マスター、“中間成果物”って、いったい何でしょうか？メンバーからもしばしば聞かれるのですが、うまく答えられなくて」

「ということは、あなたは“中間成果物”と言える成果物を作っていないということか？」

「基本設計書とか詳細設計書というものは書いています。これって“中間成果物”ですよ」

「確かに、それも“中間成果物”だね。起点の要求仕様書から、最終成果物までの過程に生み出される成果物だからね」

「一方では、客が求めているものは、動くプログラムだから、それ以外のものは必要ないという意見もあります」

「つまり、“中間成果物”を書くのはムダだという考え方だね」



「そうです」

「ここで“ムダ”という発想は、単に“不要なもの”と言うことだけではないね」

「はい、たぶん書かない方が早い、という意識も含まれていると思います」

「どうやら、皆さんは“役に立つ中間成果物”というものを書いていないようだね」

「“役に立つ中間成果物”ですか？」

「いって考え込んでしまった。」

「そもそも、“設計する”ということはどういうことができるかな？」

「何かの本で読んだことがあります」

「ほう、どんなふうにしてあった？」

「たしか、複雑さを管理する作業だ、と書いてあったと思います」

「それで、どういう意味が解ったかね」

「どういう意味って、改めて聞かれると、何と書いてよいか・・・」

「自分の言葉にして吐けないということは、分かっていないということだね」

「といわれて妙な顔つきになった。」

「じゃ、“複雑さ”というのは何を指しているかわかるかな？」

「要求仕様のことだと思います。あれを実現するとか、動く時の制限や、お互いの動きも絡んだりしていますから」

「いいね、これは直ぐに分かったわけだ。それに、今は自分の言葉になっていたね」

「いって、少し持ち上げてみた。さあ、此処から彼自身の頭を回転させる事が出来るか。」

「じゃ、“管理する”というのはどういう意味だろう」

「さすがにこの質問には黙ってしまいました。日本語の“管理する”という言葉は、非常に厄介な言葉で、多くの場合、何を意味しているのかわからないまま煙に巻かれてしまう。その上、“それで、どういう意味ですか？”と質問もできない言葉のようです。」

「イメージが湧かないかね」

「はい」

「でも、その前は“要求仕様書”だということ

は分かったのだから、考えられるだろう」

「うーん、複雑に絡み合った要求仕様の実現を管理すること、で良いのでしょうか」

「そう、そういうふうに置き換えれば良い」といったものの、そこから前に進まない。

構造化手法の時代では「分割」とか「階層化」といった表現が出やすかったが、最近はどういった言葉が日常から遠くなったのかも知れない。

画面のレイアウトを決めながら、ボタンが押されたときの処理や、表示領域に展開するコードを書けば、一つの画面のプログラムが書いてしまう。そのような状態の中では、コードを書くことが設計する事に見えてしまうのも無理はないかもしれない。でも、要求仕様の複雑さは解消されているわけではないし、設計するということの基本的な意味が変わってしまったわけでもない。

「たとえば、今書こうとしている関数に求められていることが複雑で沢山の処理が含まれているときはどうするかね。悪戦苦闘して当初の考え通りに1つの関数で書くかね？」

「いえ、そんなことをすれば、ミスも多く発生してテストに時間が掛かります」

「では、どうするか？」

「単純な機能の関数に分割して、上位の関数で条件に合わせて呼び出す形にしたいと思います」

「そうですね。それが自然な行動だね」

「あ、そうか。設計するということは、分割したり階層化したりして、単純な関数の集まりにしようことですか」

「だいたい合っているかな。ただ分割と階層化は一体だよ」

「ということは、“管理する”ということとは、適切に分割と階層化を繰り返して複雑さを解消してしまおうこと、ということですか」

「一つ補うと複雑さを解消するということろで、“決めていく”という行為が入る」

「じゃ、設計書というのは、決めていく過程が記録されたものということですか」

「中間成果物というのは、きちんと階層化の連鎖を維持し、変換しながら決定していった経過が残された文書でもあるんだよ」

「だから、レビューは、変換前の成果物を参考にするのですね」

「設計レビューは、こうしたきれいな体系が出来ていれば、きちんとしたレビューができる」といったところで、レビューと中間成果物の関係がクリアになったようだ。

「そこで、一歩踏み込んで、十分に設計された状態というのは、どういう状態かわかるかな？」

「それで、要求仕様の段階での複雑さが解消されている状態ですよ」

「ということは？」

「コーディングの段階では、要求仕様にあった複雑さや難易度のようなものが

が消えている状態ということですか。あっそうか、だからマスターは、コーディング段階での生産性を気にしていたわけですね」

「考えが繋がったようだね」

「マスターは、いつも“トップスピード”で書くということをおっしゃっていましたよね。十分に設計された状態であれば、それが出来るはずですよ。逆に手のひらに汗が滲んでこないようだと、設計が不十分だということで、設計に戻るんですね」

私はじっと彼の顔をみた。確かに設計することの意味を理解したような言葉を使っている。十分に設計された状態であれば、トップスピードでコードが書ける、と言う認識の部分には問題はない。だが後半の部分は、果してどこまで分かっているのか試す必要がありそうだ。

「なかなか良いところまで来たね。だが“手のひらに汗が滲まないときには設計に戻る”というところはどこまでイメージできているか？」

「どういうことでしょうか？」

「あなた自身の環境の中で、なかなかコーディング作業が進まない状況をイメージしてごらん？実際に、コーディングで立ち止まるときって、どういう状況かね？」

「データ構造がどうなっていたかとか、データ要素の取る値の制限などを調べたりします。それと今はこのグローバルデータを更新できるタイミングかどうか調べたり、何日前に、よく似た関数を書いたような気がしたのでサーチしたりしますね。関数の呼び出しの深さが気になって、ずっと上に辿ることもあります」

「そうやって立ち止まっているんだよね。よく似た処理をしている関数を見つけたときはどうする？」

「ある関数の中の1部の処理として組み込まれている場合もあって、その場合は共通する部分を切り出して独立した関数にしたりします。うまくいかない時もあります」

「他に、止まっている状況として、どんな時があるかね？」

「そうですね。そこで何を実現すればよかったのか仕様書から確認したり、他の関数との間での機能の担当がどうなっているか調べたりしています」

「たしかにそれは“設計に戻る”行為だろうが、“設計書”に戻っているかな？見ているのはソースコードじゃないの？」

「そういわれてみれば、確かにソース中心に作業が行われています」

「適切な設計によって生み出された中間成果物というのは、階層化の変換過程で決定の様子が記述されていて、そこでの変換ミスレビューで見えて、設計の詰めが甘いことが判明した場合でも、その箇所が直ぐに連れ、その関数で求められていることが容易に分かり、改めて、曖昧さや複雑さを解消する対応を済ませて、元の作業に戻れるもの、として、私は書いてきたし、それで他の人に遅れを取ったこともない」

「カウンターの前で必死に頭を整理している。」

「中間成果物が負担に感じるの、役に立つものに成っていないということでしょうか」

「多分、連鎖を持たない不完全なものしか書いていないのだから」

必要な中間成果物が合理的に連鎖した状態のプロセスが一番早い。中途半端だとかえって時間を失うだけだ。

か ね 曉 鐘 の 音 158

民主主義と議論のあり方

今、国会では新しい年金法案について紛糾しています。今の年金制度は、全員雇用、終身雇用を前提としていて、すでに実態に合わなくなっているのに、その根本には触れていません。収入が無くても払わなければならない、加入期間が不足すれば、年金を貰えない制度とは一体何なのか。少なくとも、私には政府が運営する「ねずみ講」にしか見えないのだが。

こうした年金問題での国会の議論を見ていて、また、この状況に関して新聞やTVで報じられる国民の感想を見ていて、やはり日本には「民主主義」が根付いていないと痛感しています。民主主義については、これまでこのページで何度か触れてきましたが、今回は「議論のあり方」について触れてみることにします。

「議論」は何のためにするのか。それは、提案された案に含まれる欠陥を正すためであり、曖昧さを質すためです。その結果、最初の案が修正され、誰もが受け入れることの出る案になることが一つの姿であり、議論の成果です。ただし、現実には誰もが受け入れられる案にならない場合があります。既得権益に関

わるものは、利害が対立します。諫早湾の堰のように解決が長引いた場合も、結果的に利害が対立し、両者が受け入れることが出来る案にまとまりません。このような場合、民主主義に於ける政策決定のルールとして、「多数決」という制度があります。つまり、「五〇% + 1」で決めるのです。

ところで、「議論」の進め方に二つあると、私は考えています。政治の世界での議論と、企業のような一つの目的をもった組織での議論では、その進め方に違いがあり、そこに「民主主義」の姿を見ることが出来ます。

まず、企業には「利益を上げて社会に貢献する」という組織の目的があります。全ての議論は、最終的にはここに帰結させるために行うことになりま

す。その案自体が、企業目的に適うか、そこでの手段は企業目的に適うかを議論することになります。株主もそれをチェックします。

一方、政治の世界での議論は、これとは趣を異にします。議会の場合は、企業のように一つの目的があるわけではありません。確かに、「国益を優先する」というのがありますが、企業のように誰もが共有できる「利益」という形で現れるわけでは

ありません。また、政策には多面性があり、一方の利益を守れば、他方の利益を損ねることもあります。

議員というのは、それぞれの支持者の代表として選ばれている人たちであって、もとより「利益代表」なのです。したがって、議会では自分たちの利益代表をどれだけ多く送り込むかが重要になってきます。議員

の方も、自分たちを選んだ支持者の思いを満たすことが使命でもあります。そこには、国の将来に関わる思いもあるでしょうし、単に税金をどれだけ分捕ってくるかという利益誘導の思いもあるでしょうが、議員の行動は、支持者の思いの枠に制限を受けるのは避けられません。

そのため、議会で行われる議論は、どちらの案が優れているかの主張の張りあいになるのは当然なのです。そこでは、相手の案の欠陥を一つでも多く指摘し、自分たちの支持者にとって不利益が大きいかを示したり、相手の支持者にも再考を促すような欠陥の存在を浮き上がらせようとする議論が必要になります。

利害の対立するような政策に対しては、意見の一致を見ることはありません。わずかに妥協できる範囲に法案が調整されることはあっても、基本的に、与党の支持者に有利な法案になっていくことは避けられないのです。それは、民主主義としては決して間違っていないのです。それが不満ならば、自分たちの代表を多く送り込んで議会の構成を変えれば良いのです。

ところが日本の国民の多くは、議会での議論に対しても、一つの優れた結論に導かれることを期待しているよう

です。企業での議論のように、皆（より多くの人）が納得するような結論になることを求めているのです。でも現実に行われている議論は平行線であり、議事のボイコットです。この姿を見て議会に対する不信感や政治から離れていくのですが、それは明らかに民主主義を勘違いしている証拠です。

民主主義に於ける政策の決定方法は多数決であり、議論を尽せば、誰もが納得するような政策になるとは限らないのです。だからこそ、一定期間に行う「選挙」が民主主義を維持するため

今月の一言

「謂う勿れ、今日学ばずとも来日ありと。謂う勿れ、今年学ばずとも来年ありと。日月逝（ゆ）けり、歳、我と延びず。嗚呼、老いぬる哉、これ誰の愆（あやまち）ぞ。」
朱子「勸学」
（佐藤一斎「言志四録」講談社学術文庫より引用）

「学ぶ」ということの意味が見失われて久しい。いや、私自身も学校で「学ぶ」ことの意味を教えられた記憶はない。自分で、いろんな本を読んでいるうちに、世の中に自分を活かすこと、自分の人生を精神面で豊かにすることにありと考えるようになった。そして

今は、自分が決めた「終わりの姿」で一生涯を終えるために「学ぶ」と言っても良い。

もちろん、「学ぶ」目的は他にあって良い。貧しい時代は、経済的に豊かになるところにその意味を見出した。だがすでに経済的な豊かさを手に入れた状況の中で、多くの人は「学ぶ」こと

の意味を見出せないでいる。浮き草のように漂い、流れてくる新聞の見出しや、TVの画面をすくっている。

通勤の電車の中にも、何を学べばよいのか分からなくなっている人々をたくさん見出すことができる。彼らは、「明日やれば良い」と思っているのかも知れない。いや、その言葉は何年も前に浮かべたが、部屋の隅っこに追いやられたままである。実際、この間に確かに「学んだ」と言えるものは何だ

ろう。それで良いのか。それで一生を終えて良いのか。それが朱子の言葉である。