第４２号
平成５年３月

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 15

反復可能プロセス

初期のプロセスとそれに続くプロセスとの大きな違いは「約束」です。約束した期間に間に合う。約束した機能が盛り込まれている。約束した操作性が実現している。約束した品質が確保されている。この様に「約束」出来ることは「反復可能プロセス」以上のプロセスに居ることの証でもあります。
「渾沌」からの脱出の基本原則を踏まえて初期のプロセスから脱出できたとしても、反復可能プロセスから脱落しないためには「約束」を管理する仕組みが必要です。このような「仕組み」がなければ、「約束」も有能な人がその開発現場にいる間だけしか守れないことになります。

.ソフトウェアの品質保証

反復可能プロセスを維持するには、組織の管理やソフトウェアの構成管理の他に、品質を保証する仕組みも用意しなければなりません。
誰でも品質が重要なことは分かっているはずです。しかしながらバグを一つでも減らして出荷したくても、迫り来る納期の前に妥協せざるを得ないのが現実でしょう。
“品質を上げようとすると開発期間が伸びる。開発期間を短くすると品質が確保できない”と言うのが多くの開発担当者の言い分です。ここで呼ばれている「品質」とは、実際には「バグの数の少ないこと」に対応しています。つまりバグの数を減らそうとすれば開発期間が伸びる。開発期間を短くするとバグを取り切れたかどうか保証できない、ということです。これでは炊き上がった御飯の中から混ざった石を探すようなものです。
米に混ざった石を除くには、洗米の段階から除かなければならないのと同じ様に、ソフトウェアのバグもソース・コードになる前に取り除く行為が必要です。

昨年ＩＳＯ／９０００－３が制定され、ソフトウェアの「品質保証」という言葉も珍しくなくなってきました。そこで求められているのも“品質の織り込み”であって、幾つ拾えるか分からないようなバグの落ち穂拾いではありません。そして品質を保証するためには有効に機能する「仕組み」を持っていなければならないのです。
そこで、ソフトウェア・エンジニアリング全般を理解し、ソフトウェア開発に精通した有能な人材が必要なのですが、ハンフリーも言うように“管理者は当然、製品の設計に最もよい設計者をつけようとするので、ソフトウェアの品質保証（ＳＱＡ：Software Quality Assurance）にはそのような人材は回ってこない”という問題があります。どうやらこの問題は洋の東西を問わないようです。
しかしながら、これからはこんなことを言っておれなくなります。早急にＳＱＡの専門家を養成する必要があります。ソフトウェアの「開発」の部分だけに力を注ぐのではなく、ソフトウェアのライフサイクル全般に万遍なく人材を投入する必要があります。

残念ながら、今の日本のソフトウェア開発現場に於て、ＳＱＡの専門家が育つ環境ではありません。関数名やデータ名の付け方を規定するネーミング・ルールや、ソースコードの形式を規定するコーディング・ルールすら徹底することが出来ないのが現状です。当然“動いているプログラム”を評価し改善を求めることなど出来る状況ではないのです。チームワークやリーダーシップを自分のこととして考えられない様な環境でＳＱＡの専門家を育てることは、極度の酸性土壌に植物の芽を出させようとするようなものです。

.ＳＱＡの能力

ハンフリーはＳＱＡの専門家に求められる能力として、以下の４つを挙げています。
１）統計的手法
２）品質管理の原則
３）ソフトウェアプロセスに関する知識
４）論争的な状況の中にいる人達とうまくやっていく能力
３つ目までは経験と訓練で何とかなるでしょうし、実際にそれぞれの開発現場にも居るでしょうが、４つ目の能力を備えた人となると、非常に限られてくるはずです。特に教育機関に於てリーダーシップが省みられなくなった今日の状況では、企業に於て最初から意図的に育てるしかないでしょう。ＳＱＡの専門家なしで、ソフトウェアの品質を保証し続けることの困難が明らかである以上、素材を見抜いて早い段階からＳＱＡの専門家を想定して育てるしかないのです。

.なぜＳＱＡが必要か

ＳＱＡが難しいのは、それが「監視」の性質を持っているからです。プログラムは“動けばよい”と思っている人にとって、凝集度や結合度などで評価されることには耐えられないはずです。データ名称の付け方にまで干渉するのかと苛立つのです。彼らの機嫌を損ねるのを恐れる故に、データ名称は好き勝手に付けられ、モジュールの構造も野放しとなってしまうのです。その結果が現状であり、１年中端末の前に座ってソースコードと対決しなければならないのです。終わったと思っても、月曜日になると新しいバグが報告されてくるのです。一体、何時までこれを続けるつもりなのか。耐えれば何とかなるものでもないし、救世主が現われる訳でもありません。新しく入ってくる人に渡そうとしても、自ら織り込んだ複雑さの故に何時でも呼び出されてしまう。それだけでなく、このままでは彼自身も１０年後には活躍する場を失うことになるのです。

ハンフリーは“品質が決定的に重要な場合は、なんらかの独立したチェックが必要である。それは人間が信頼できないからではなく、その人達が人間だからである”と言うように、ＳＱＡは彼らを監視するためではなく、寧ろ彼らの幸福に寄与するためにあるのです。１０年後に豊富な経験を活かして、多くの人から喜ばれ、頼りにされるような活躍が出来る可能性を与えてくれるのです。
ＳＱＡの目標として、ハンフリーは３つ挙げています。
１）ソフトウェアとその生産プロセスを正しく監視することによって、ソフトウェア品質を改善する。
２）ソフトウェアとその生産プロセスが、決められた標準と手順に沿っていることを保証する。
３）生産物、プロセス、標準化に不十分な点があれば、それらが訂正されるように管理者に気付かせることを保証する。
このようなＳＱＡの効果を上げるために、開発側との連係作業が必要ですが、ＳＱＡは製品の開発、或いは品質計画を立てることには責任を負いません。求めに応じて参考意見は述べることはあっても、計画はあくまでも開発側の責任です。

このように、これからのソフトウェア開発に於てＳＱＡによるフォローが必要であることは理解されても、日本の多くのソフトウェア開発現場でＳＱＡを取り入れるには、周到な準備が求められるのです。さもなければ、“我々開発側を採るのか、ＳＱＡ側を採るのか”という文句が、プロジェクトの責任者に突き付けられ、再び混沌の世界に引き戻されるのです。

（ＳＱＡ－次号に続く） ◆

安直なミドル狩り

▲　今回の不況もいよいよ雇用調整の段階に入ってきました。希望退職の募集に留まらず、ホワイトカラーのミドルを中心にした退職勧奨や、採用内定の取り消しが「企業の論理」で行われています。丁度岩盤の弱い所にマグマが噴き出した状態に似ています。
▲　各社はここに来て一斉に慌てだしていますが、今回の不況も自らの責任で（本気で）考えれば、１年程度で回復することなどありえないことは分かる筈なのに、その判断を政府の役人や役人上がりのエコノミスト、或いは同業者に任せてしまった。
▲　１５年前にＪ・Ｋガルブレイスの『不確実性の時代』がベストセラーとなってから、“今は不確実の時代で先が読めない”というセリフが経営者の間で常套語となってしまいました。今の時代は先が読めないのが当たり前で、読める方が不思議という情けない風潮を作り上げ、先を読もうとしないことに経営者として自責の念を感じなくなったようです。まさに「皆で渡れば怖くない」の経営者版です。
▲　もっと早い段階で我が社はどの方向に進むべきか、という明確な指針をミドルに示し、変革を求めるべきであり、彼らミドルがそのことに対応できなかったのなら退職勧奨も止むを得ませんが、エンジンの調子が悪くなったからと言って、“大勢の社員を助けると思って飛び降りてくれ”というのでは、機長（ＣＥＯ）の責任を果たしたことにはなりません。

暁鐘の音（かねのね）

25

専門化 ― その功罪

先頃、情報処理に関する職種を九種に分類する案が、通産省の諮問機関から中間報告として出されました。参考までにその職種を以下に列記します。

◆システムアナリスト
◆プロジェクトマネージャー
◆アプリケーションエンジニア
◆プロダクションエンジニア
◆テクニカルスペシャリスト
◆システム運用管理エンジニア
◆教育エンジニア
◆デベロップメントエンジニア
◆システムアドミニストレータ

という訳で、分類学者よろしく丁寧に分類したものです。確かに今日では実際に情報処理に関わる仕事として、このように分類できる作業が行われているようですが、ここまで細かく分類されると、職種間の流動性を損ねる危険が出てきます。

日本に於て、情報処理が産業として産声を上げてから二十数年が経った今では、そこに携わるエンジニアに求められる専門性は遥に広く、そして深くなっており、一人でカバーできる範囲を越えていることは確かです。その意味では分業化は避けられないことですが、今日の日本の情報処理業の「分化」は、その

ような純粋な理由だけではありません。

この業界は十数年前に通産省の音頭によって大量にソフトウェア・エンジニア（？）を増やしましたが、彼らは専門的にしっかりと訓練されなかったため、各人の守備範囲が狭いまま、現場の作業に投入されたのです。本来なら、その程度の技術では一人前の収入を得られることが不自然なのですが、折からの右肩上がりの上昇景気に支えられて、職業として成立してしまったのです。

図らずも今回の不況で、ソフト業界ではその様なエンジニアが寒風に晒されているのです。一〇年経験したと言っても、与えられたモジュール仕様をソースコードに変換することしか出来ないのでは、最早どうしようもありません。

先の九つの職種を見ると、テクニカルスペシャリストやシステム運用管理エンジニアの様に、分類することの妥当な職種も幾つかあるのですが、中にはどうしてわざわざ分類しなければならないのかと考えさせられる職種もあります。現状を追認して分類すれば、このような分類になるのでしょうが、逆に細切れの職種を

正当化してしまうことになるでしょう。

「分類」という行為は、専門化あるいは精密化する過程で、必ずついてくる行為ですが、分化が細かくなると、根本を見失うことになります。例えば医者の場合、内科や外科だけでなく循環系だ呼吸器系だと、細かく分類された専門医は沢山います。疾患によっては幾つもの専門医の診断を必要とすることもありますが、総合的に判断できる医者が少ないと言われています。これも分化し過ぎたことの弊害で、「専門」の中に閉じこもって安心している内に、所謂“専門的愚昧”の状態になったのです。

やはり細かく成りっぱなしではだめで、時には専門を組み合わせたり、全体に戻らなければなりません。実際、今日では電気と生物、物理と化学などの融合が行われており、そこでは新しい成果を上げつつあります。あるところまで専門化したら、今度は向きを変えて統合して行かなければなりません。狭い範囲に安住していては人間自体も狭くなってしまう。アプリケーションエンジニアであろうとデベロップメントエンジニアであろうと、そこで必要な業務知識は異なっても、対象を理解するだけの基礎的知識を必要とすることには変わりはないし、分析して問題を詰めてゆくことも同じです。つまり、箱の形状と中身は違っても、箱を用意することと、箱への詰め方（仕事の仕方）は殆ど同じなのです。

それよりも重要なことは、この産業は僅か数十年の歴史しかなく、今日のデベロップメントエンジニアの行っている作業の中身が、五年後には相当に変化していることが予想されることです。その度に追認的に職種を変更するのでしょうか。

今回の九分類は、最終案ではないようですが、果たしてこれからのソフトウェア・エンジニアのあるべき姿として方向を示したものでしょうか。もしそうだとして、どうしてこんなものを通産省が決めなければならないのでしょう。それぞれの企業の関係者

が、自分達の責任でソフトウェア産業の将来のあるべき姿を考え、その延長上でエンジニアのあるべき姿を描けばよいことでしょう。

経営者の横並び意識の結果として、若いソフトウェア・エンジニアが、「専門化」された職種に安穏としてしまい、気付いた時には適応性を失ってしまう、ということに成りはしないだろうか。

その様に考えると、今回の答申からは、社員の教育助成を名目にした「補助金」という蜜の香も漂ってくる。なにしろこの業界は名目を上げては蜜を求める体質が染み付いているから。■

今月の一言

「一燈を提げて暗夜を行く。暗夜を憂うることなかれ、只だ一燈を頼め」

佐藤一斎

ソフトウェア・エンジニアにとって世はまさに暗夜である。たとえ今は仕事があっても、プロジェクトが何時中止になるか分からない。中止になったらどうなるのだろう。或いはこのプロジェクトが終わった後は、自分はどうなるのだろうと考え出すと、何も手に付かず、夜も眠れないという人は少なくないだろう。果ては、この仕事を選んだ事を後悔し始めたり、もっとひどいのになると、一〇年早く生まれて来ればよかったなどと言い出す始末。これでは暗夜に放り出され、火の消えた提灯を手にその場に座り込んでしまった様なものである。

「一燈」とは自己の事であるが、この「自己」は何時でも足元を照らしてくれる訳ではない。自己が「一燈」となるためには、自分を信じる事が求められる。自分を信じ自分を損ねなければ（＝自尊）、「自己」は光りだす。人それぞれに光の強弱はあっても、間違いなく光りだす。元より自分の能力など分かっている人は少ない。それなのに多くの人は、自分にはとても出来ないと、能力を「画って」初めから諦めてしまう。今こそ、受け身の習慣から脱却し、自己の「一燈」を頼りとして、歩み初めるしかないのに。