

第 38 号

平成 4 年 1 1 月

© 1992

(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 クリエイト

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 11

仕事をほとんど破壊してしまう」のです。

このようにソフトウェアの開発組織が混沌状態に陥る原因として幾つかの要因が考えられます。

要因 1 : 計画に対する強い政治的圧力

政治的に強い圧力がかけられると、計画は単なる希望的予測となり、3 ヶ月ぐらい必要なところを無理して2 ヶ月と返事することになります。一般にそのプロジェクトを担当する人が立てた計画は、担当しない人が立てた計画に比べて2 割程度は短くなることは良く知られています。それは悪い印象を与えたくない、という無意識の力が働くからで、その上に政治的圧力がかけると、殆ど実現不可能な計画となってしまうのです。

要因 2 : 無計画な約束

多くのソフトウェア組織では作業計画を作成する前に「約束」が行なわれています。これは上手く作業計画を立てられないことにも起因していますが、この様な「約束」の下で担当者が採りうる行動は2 つ考えられます。一つは「自分の運動靴を探し始めて」走れるところまで走り出すか、最初からその様な「約束」は拘束力を持たないと自分で判断して下駄箱の隅に押し込めるかです。当然計画帳(?)に進むことはありません。しかしながら「約束」は完全に効力を失ったわけではなく、そのような事態を救うために結果的に最もリスクの高い『人海戦術』が採られるのです。尤もこの場面では、プロジェクトを中止するか人海戦術以外の策はないでしょう。

要因 3 : 「銀の弾丸」を欲しがる

「銀の弾丸はない」というのはIBM/360の開発者であるBrooks博士の有名な言葉です。複雑な状況に陥ったとき、それを解決してくれる銀の弾丸を求めようとしても、一気にその様な問題を解決する様な虫の良い方法など無いということです。実際に「改良された技術によって助けられる場合は多くあるが、効果的な管理を必要としているほうがもっと多い」というように、作業の進め方を改善するだけでも効果があるものです。

要因 4 : 規模の問題

プログラムの規模がもたらす影響は、一般に想像する以上に大きいものです。「10,000 行の経験は確かに助けになるが、それに続く100,000 行では多くの新しい学習が必要である」と言うように、大きなプログラムを作るには、それなりの方法があるのです。

規模がもたらす問題として、

- 1) 理解が困難になる
 - 2) 関係する人が増え共通の表記法が必要
 - 3) 一度で仕上げるができないため、多段階リリースが必要
- などが考えられ、チーム・マネージメントなしでは混乱することになります。

(次号に続く)



初期のプロセスレベル

ハンフリー氏の提唱するプロセス・レベルには5 つの段階があることは既に紹介しました。その際に各々のレベルは一体どのような状態をいうのか、ということも簡単に触れましたが、今回から個々のプロセス・レベルを取り上げて、それらのレベルに陥る原因とそこから抜け出す方法について、ハンフリー氏の説を紹介しながら私の考えを述べることにします。ただ、現実問題としてハンフリー氏の言う5 段階のプロセス・レベルの内、殆どの組織は第一段階のレベルであり、時として第二段階のプロセスはあっても、せいぜい3 段階程度までしか届かないものと思われま

す。ハンフリー氏は「大部分のソフトウェア組織は、少なくともある期間は『初期プロセス』で運営され、多くの組織は容易に『初期プロセス』から抜け出すことは出来ない」と言っています。残念ながら日本のソフトウェア組織も同じようなことが言えるようです



= 初期のプロセスとは =

初期のプロセスに対しては明確な分類範囲はなく、寧ろこれは第二レベルの反復可能プロセスのレベルにも達しないソフトウェア組織に与える名称です。そこでは本来ソフトウェアの開発に必要な諸々の行為が為されていないし、時にはそれらが絡み合って、改善の糸口が見つからないこともあります。ここでもう一度初期プロセスが表す「状態」を整理してみます。このような「状態」は相互に絡んでいる部分もあるため、殆どの初期プロセスにおいて、幾つかを合わせ持っていると思われる。

= 担当者は優先度による作業の管理ができない =

あらゆる作業において、『他人に出すものは最優先で処理をする』というのは、仕事の仕方の一つの原則です。チームのメンバーをそれぞれ独立したCPUと見做したとき、望ましい状態というのは、全てのCPUが並行して動作している状態です。ところが何処かで優先度の管理が出来ないと、そこからのレスポンスが返ってこないために、幾つかのCPUは連鎖的に「待ち状態」に入る可能性があります。

もう一つ優先度に関わるものとして『約束』があります。全ての行為の根底にあるのは『約束』です。「約束」というものがあるから、それを果たすために色々な工夫をします。途中で予期しない事態が発生しても、代替案を考えるのは「約束」を守るためです。またリスク管理を行なうのも「約束」があるからです。

しかしながら初期のプロセスに於ては『約束』という概念が希薄なために、度重なる幸運を前提と

するような計画しか立てられません。例えば或る時点までにB社からのレスポンスがない時、またCさんの作業が遅れだした時にどのような対策を講じるかといったことが、事前に考えられていることはないし、そのようなチェック・ポイントすら想定されていません。したがってスケジュールの後半で予定外の事態が発生すると、スケジュールがメチャメチャになってしまうのです。結局このようなプロセスでは、「まれに計画どおりに出荷できて、それは例外的なことか、あるいは組織力にたよらない英雄的な個々人の努力の成果である」ということになります。

= しばしば正直者で有能な人間に支えられている =

時としてこの様なプロセスには責任感の強い有能な人が居ることがありますが、殆ど例がなく、彼ら(殆ど少数の人達)は多忙で過大な役割を担っています。そして変更管理の不備から、仕様がしばしば変更され、その度に彼らは振り回されるのです。彼らの能力は、ソフトウェアの開発そのものに向かって発揮されるもので、残念ながら組織に向かって働き掛けることに、その力は発揮されることは稀です。

= 優秀なエンジニアは定着しない =

誰もこの様な初期のプロセスを歓迎しているわけではありませんが、有効な手を打てないまま次第に症状が広がってしまうのです。丁度HIVウイルスに感染した人が、潜伏期間を経過したあと、免疫機能を失って次々と症状を発するように、一旦症状が始めると、色んなところで支障を来すようになります。それでもソフトウェアを開発しなければならないために、結局スケジュールのみが優先され、出荷できたとしても誰も満足できる結果にはなりません。したがって作業に喜びがなく、一人二人と組織から抜けて行くことになります。そしてさらに症状が進行した組織では、残された人達は地域と深い関わりを持つ人達が、改善の望みを断った人達ということになってしまふことがあります。

= 同じ問題が常に繰り返される =

計画はその場限りで十分に練られることはなく、「約束」の概念が薄いために、スケジュールは初めから「努力目標」でしかなく、設計制御は為されないため、テスト段階に入って初めて考慮されていないことに気が付き、事前に必要な資源が真剣に検討されないため、資源はいつも不十分な状態となります。後戻りはいつものことで、解決された筈の古い問題が常に蒸し返されるのです。毎回この様な状態になっているにも拘わらず、過去の過ちから学ぼうとする人は殆どおらず、たとえ居ても彼はまだ周囲の信頼がなく、組織に対して殆ど影響力を持っていません。

「不適切なプロジェクト管理が、優秀な技術者の

か ね の 音

21

反省だけか?!

今回の不況は、ソフト産業に幾つかの反省を促しています。ソフトウェアもよく知らない新入社員に、僅か三ヶ月程度で、とにかくプログラムだけは書けるように訓練し、後はOJT?とかいつて客先に派遣する。五〇。人程度居れば年間二。億円の利益を出せる勘定で、かつて銀座のクラブで派手なスーツを着て、ヨットの話しをするのは、ソフトウェア会社のオーナー社長だ、と言われたのも納得できるところです。

その結果、日本では要員を派遣することが、ソフト産業の主たる業態となってしまうたのです。そこで派遣されるSE(?)の殆どは、設計法も分析法も知りません。中には、プログラミングとはどう有るべきかも分かっていない。彼らは最も重要な最初の三ヶ月で、プログラマはただ「動けば良い」と教えられたのです。プログラミンのスタイルなど誰からも指摘されたことはありません。

その上、彼らにとってもっと深刻なのは、問題を解決する方法、或いは思考法と言われるものを身に付けていないことです。仕様の形になったものについて、何とかしてプログラム

に変換できて、何が問題なのか明確でないものや、仕様書として整理されていないものについて、手も足も出ないのです。

「約束」を実現するために、身の回りから問題点を探し出し、それを分類・整理し、優先度を付けながら一つづつ解決して行くことを学ぶ機会を持たなかつたため、混沌の中から秩序を見い出すことが出来ないのです。今回のソフト不況で、多くの経営者はこのことに気づいたはずで、実際、幾つかの雑誌で、ソフトウェアの経営者が、この点を反省している記事をししばし目にします。

一。二。万円が六。万円に

経営者が、幾らかの様な問題点に気づいて、コンピュータ教育の必要性を痛感しても、現実には派遣先から多くのSEが戻されておられ、今月の給料を支給する段になれば、すぐに目先の問題に置き変わってしまうはずで、多くのソフトウェア会社の経営者は、数年前までソフトウェア産業の将来像も考えず、ただ目先の儲けに狂喜したのです。

現に、都内のソフトウェア会社で、かつて一人・月一。二。万円の見積りを出していたのが、今や同じレベルのSEで六。万円に値下げして売り込んでいるよう

です。となると、この差額の六。万は一体何なのか疑いたくなるが、それは置いておくとして、結局行動として現われてくるのはこれしかないのです。つまり、目先のことで儲けた経営者は、この場に及んでも、目の手段で対応するしかないし、実際ここまで来たら他に選択肢もないでしょう。

勿論、このような状況でも体力勝負で幾つかのソフトウェア会社は、この場を耐え忍んで生き残ることが出来るでしょう。そして生存競争に勝った時には、依然としてこれまでと同じような「需要」があることが予想されます。それは多くの企業が数年間ソフトウェアを開発を凍結したことにもなつて、設備投資的に開発需要が発生することが考えられるのと、このような状況にあるにも拘わらず、うまくコンピュータ教育が実現できずに、発注元の体質も変わっていないことが予想されるからです。

コンピュータ教育の前に

確かにコンピュータに関する各種の教育は必要です。設計法も知らずにシステムを設計されたはたまりませんし、分析法や問題解析法を持たずに新しい分野で予定どおりにプロジェクトを遂行することは無理でしょう。ソフトウェア開発のコストが高くなっている原因の過半は、コンピュータ教育の不足から来るものです。この様な認識から、ソフトウェア会社やソフトウェア開発を行う企業において、数年前からSE教育の充実が叫ばれ、そして実施し始めた企業が増えていいます。それに合わせるかのよう

に、SE教育を請け負う企業が増えてきたことは、その状況を裏付けています。

しかしながら、こうしたSE教育請負業者にも、将来の日本のソフトウェア産業の姿を描いていくように見えません。ただそこに教育という「需要」があるから旗を上げただけで、発想としては、今日行き詰まったソフトウェア産業の経営者と変わらないのです。

この国の現状において、通り一遍のコンピュータ教育を施すだけで、事態が変化することはないでしょう。例えば構造化手法を勉強したといつても、実際の場にはそれを持ち込むことは容易ではないことは、実際に学んだことのある人なら分かるはずで



「幸福——誰もが到達しようとしてなかなか到達できないあの目標——も、すべては時間の利用の仕方次第で決まる」

アーノルド・ベネット

時間の問題は嘗て「SCの作り方」として連載したことがありますが、ここでは深入りはしませんが、「時間の使い方」は私達にとつて永遠の課題であることは間違いないようです。

今月の一言

数年前にビジネス手帳が女性の間に流行した時、予定表の隙間を必死に埋める症状(??)が問題にされたことがあります。それまで彼女達は自分の時間を気にしなかつたのが、「予定表」という形で見えたことで、その「空き時間」が恐怖の時間となり、友達と約束を取付けたら、各々のスケジュールに通つたりして、充実

時間には余ることはありませんし、「前借り」もできません。しかしながら「天引き」はできません。継続して何かをやるうとするとき、時間を「天引き」しない限り、何かを残すことは難しいでしょう。

半年、一年の完全な準備期間があるならまだしも、現実の仕事を取り込むうなんて、新しい方法を取ることではないのです。とは言つても、それを遣らなければ将来が見えてこないことも分かっているのです。

会社が主催するコンピュータ教育を受け身で受けたのでは、こを乗り越える力が出てこないでしょう。これを越えるには、学ぶことの意義が分かっていることが必要なのですが、「学ぶことが創造的経験となるように教育され」ていない日本では、そのままでは単に疎外感を解消することにしかならない。