第114号

平成11年3月

E-mail: © 1999
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 045-933-0379
FAX 045-931-9202

ソフトウェア開発の原則

「ソフトウェア開発 201の鉄則」から 32 かい

いいかげんなコスト見積りの原因の上位5項目は、すべての要求プロセスに関係している。

1. 頻繁な要求仕様の変更
2. 要求項目の欠落
3. 不十分なユーザーとのコミュニケーション
4. 質の悪い要求仕様
5. 不十分な分析

不正確な要求仕様によるリスクを減らすためにプロトタイプングを使い、変更を制御するために構成管理を使い、将来のリリースのために新たに要求仕様を作る計画を立てよ。要求分析と要求仕様のために、より形式的なアプローチを利用せよ。

(201の鉄則: 原理38 <一般原理 = 要求分析の原理 = いいかげんな要求仕様からはいいかげんなコスト見積りしかできない>)

解説

ソフトウェア開発にまつわるほとんど全ての問題は、ここから発していると言っても過言ではありません。作業が最初から遅れ始めるのも、あふれるバグの処理に大きな時間と費用がかかるのも、終盤になって顧客との間で起きる仕様上のトラブルも、結局は、最初の要求分析工程での、いい加減な分析に始まっているのです。

原理38は「いい加減な要求仕様から・・・」という表現をしています。時には「要求仕様」自体が存在しないこともあります。せいぜい、そこにあるのは「製品仕様」であって、それは要求仕様の代りにはなりません。

要求分析工程って?

このようないい加減な要求仕様になる原因は、要求分析工程での作業が理解されていないことにあります。201の鉄則では、『要求分析とは、(1) 解決を必要とする問題について、それを引き出したり学んだりすること、及び(2) 問題解決のために、システムの外部の(ブラックボックスとしての)振る舞いを規定すること、を含む一連の活動である。要求分析の最終的な成果物は、要求仕様書である』と書かれています。

「振る舞いを規定する」ということは、そのシステムが求められている仕様として、どのような入力に対して、どのような出力(振る舞い)をすればいいのかを、詳細に(具体的に)定めていくことで、その過程で、別に対応を必要とする問題(懸案事項など)を発見し、早期に必要な事前調査などを行って、適切な「振る舞いの集合」に変換していくことになります。そうして最終成果物としての「要求仕様」が生まれるのです。

要求仕様化けた「設計書」

多くの現場では、要求分析工程は曖昧なのに対して、設計工程は必ず存在しています。しかしながら、設計工程の成果物を見てみると、「設計書」というよりも「要求仕様書」に「近いもの」が書かれていることがしばしばです。

これは、省かれた要求分析工程が必然的に設計工程でカバーされているのと、実際に「設

計」に入るためには、ある程度の細かな「仕様」が必要ですが、それが「設計工程」で書かれているということです。

そで行われている行為が設計行為でない証拠として、この段階になって、「仕様上の詰め」が顧客との間で行われていることです。エラー時の振る舞いも、この段階で問題になることが多いはず。勿論、適切な要求分析が行われても、この段階まで仕様上の問題が見過ごされるようなケースも考えられますが、現実には、そのような程度のものではありません。

そして当然ですが、この結果として「設計書」が存在しないことになります。設計書というのは、「仕様」をどのようにして実現するかという観点から書かれたもので、それが適切に書かれているならば、あとは実装作業は一気に進むはず。

仕様変更の主要原因

こうして、実装工程(設計作業を含む)に入ってから、仕様が具体化され、要求に隠れた問題や、既に設計方法を決めた部分(実装も済んでいる?)との矛盾などが表面化します。納期が迫っていることもあって、今から、やり直すこともできません。実は、仕様の変更の大部分は、こうした状況の中から発生しているのです。必ずしも顧客からの要請に起因するものではありません。要求分析工程が省かれたり、そこでのいい加減な作業のために、設計工程や実装工程に入ったところで「仕様上の問題」が湧きだしてくるのです。

しかも既に実装工程に入っている状況での頻繁な仕様の変更は、プログラムの品質を確実に悪化させます。そのうえ作業者のモラルも低下し、それがまたバグを埋め込むことに貢献するのです。

分析アプローチの形式化

設計工程と比べて、「要求分析工程」での作業の方を形式化することは容易です。以下、分かりやすいように「機能分析」に的を絞って説明します。まず製品の機能をいくつかのカテゴリに分けます。これは同じような製品であれば何度も使えます。次に、各カテゴリ内での振る舞いを記述します。これは(システムへの)入力とそれに対する出力(反応)を明らかにして

いくことです。いわゆる「分析手法」の前段が、これに相当するか、この作業を支援するはず。

さらに、エラー時の振る舞いについても記述しますが、ここで手を抜かないことです。各種の分析手法は、エラー時の振る舞いについては、当初は扱わない傾向があります。しかしながら、エラーによっては、正常処理のデータ構造に影響を与えることがあります。

要求仕様の場合は、この他に「品質的な要求」や「開発手順などの要求」も具体的に明記します。このとき重要なことは、仕様としては検証が可能な表現であることです。こうして、まとめられた要求仕様をレビュー(ウォークスルー)することになりますが、これらの一連の作業は、だいたい同じパターンです。

新規と変更での要求仕様の表現

もう一つ注意しなければいけないことは、新規の仕様と変更の仕様の表現です。「設計書」というのは、その後の「実装」作業を説明するもので、その「仕様」を「どのようにして実現するか」を記述したものです。従って、「変更」の場合も、その仕様を実現するために「どこをどのように変更するか」の記述が必要になります。

一方、「要求仕様」は、そのような設計行為をイメージさせるものであるべきです。同じ「要求」でも、それを実現する方法として、新規に開発するのか、既存のシステムを修正して開発するのかという方針が決まることで、要求仕様の表現が変わるのです。ここで、表現を切り替えておかないと、設計工程以降が実際の作業(行為)と一致しないことになります。多くの開発現場での間違いは、変更作業であるにも関わらず、新規の場合と同じ要求仕様を書いていることです。

CMMを成功に導くステップ

この原理38の鉄則は、CMMに取り組む際にもクリアしていかなければならない**最大級のテーマ**です。ここを避けて、CMMの階段は登れません。たとえ要求仕様が500ページに成ると、それが求められている以上、避けて通ることはできません。せいぜい、仕様の細かさなどの書き方の工夫や、採用するライフサイクルによって、まとめる単位が異なる余地があるぐらいです。

優れたプログラミング能力も、いいかげんな要求仕様をカバーすることはできません。ソフトウェアの開発組織は、早くこのことに気づくべきです。今後、CPUやメモリの性能が飛躍的に向上します。そこで求められる機能や性能や品質、さらにはタイムリー性も、今のままでは実現しなくなります。「今の要求仕様」でそのような状況に対応できますか?

(次号に続く)

ついに日産自動車からルノーの資本を受け入れられました。戦後50年、日本は大きな曲がり角にさしかかっています。これまでの方法ややり方、制度、価値観などが通用しなくなっていく。これは時代の流れであり、むしろ歓迎すべきです。時代の変化に合わせて、私たちにも変化が求められているのです。

か ね

暁 鐘 の 音

97

会社員？

「お父さんのお仕事は？」 「あね、会社員」
テレビのコマーシャルの一場面である。
国政調査の職業欄にも「会社員」というのがある。新聞やテレビの
ニュースなどでも、その人の職業として「会社員」が使われている。

さすがに、時々舞い込んでくる「アンケート用紙」には、「会社員」というのが少なくなつたが、自治体などから届くアンケートには、まだ「会社員」という文字が見える。果たして「会社員」というのは職業なのか？

「一員であることに意義？」
終身雇用や年功賃金制度が変わるうとして今日、「会社員」という言葉には抵抗を感じるようになってきた。それでも、年配の人たちには、あまり抵抗がないようである。

「会社員」というのは、どうみても職業ではない。「会社に属している人」ということで、むしろ「所属」の状態を表している。あるいは「給与所得者」「サラリーマン」という意味に近い。

「会社員」という言葉には「職」が表現されていない。会社の中で何を分担しているのかを問わない。ただ、その会社の一員だということだけである。確かに戦後の日本では、会社に属していることに意味があった。もっといえば、どの会社に属してい

会社に身を預ける

これまで、仕事は、その会社に入ってから覚えるのだから、採用する際には何ができるかではなく、「真っ白」で素直で勤勉であればよかった。入社後は、まず「共同体」の構成員としての意識を植え付けることから始まる。それ自身は、全面的に悪いという訳ではないが、問題は、その後の「配属」が、ほとんど一方的に行われてきたことである。この瞬間から「会社員」が始まる。人事担当も、「動かすこと」が仕事と心得ていたから、一部の職を除いて、五年も経たないうちに別の仕事に回され、そこで改めて新しい仕事を覚えることになる。

その結果、高度な技術を身に付けることなく、いくつかの仕事を転々とすることになる。本人も大きな会社の「一員」であることに意義があったため、こうして最後まで「会社員」として勤めることに別抵抗を感じるわけではなかった。自分から求めなくて

も、会社は適当に仕事を割り当ててくれたからである。
そこで大事なことは、秩序を乱さないことであつた。みんながこのことに同意し、会社に身を任せておけばうまくいった。だからそこで求められるのは、「会社に対する忠誠心」である。「会社員」という呼称は、その意味でも都合がよかった。

プロが育たない

だがこれでは「プロ」は育たない。人件費も為替も大きく様変わりした今日では、「人海戦術」は使えない。かといって、人員整理をやっても、そこにいるのは「会社員」であつて「プロ」ではないため、こんどは、以前の業績を維持できなくなってしまう。
九〇年代に入って、企業は目立ったリストラは避けながらも、退職者の不補充や新規採用を抑えることで減量し

てきたが、現実には、増大する仕事をこなせなくなっている。一部ではすでに対応の限界を超えているものと思われる。金融機関の不良債権処理にあわせて、一般の製造業などでも過剰な生産能力の削減に踏み切る必要があるが、今のままでは、モラルハザードを招く危険性が高い。「会社員」という集団意識の便利さに甘えて「プロ」を育ててこなかったため、ここに来て「ツケ」が回ってきたのである。

仕事に忠誠であれ

今、求められているのは、「あなたは何ができますか？」ということである。「職」には一般的に認知された標準的な範囲を持っている。時には、それが世界標準であつたりもする。「プログラマー」「プロジェクト・マネージャー」「システム・アナリスト」「プロダクト・エンジニア」など、そ

れぞれ任された範囲というものがあり、認知されている。
もちろん、これらの「職」の範囲や「職」自体も、時代とともに変わることは言うまでもない。だから、「プロ」であり続けようとするならば、自身も変化させ続けなければならぬ。いずれにしても、人が流動化し、海外からも人材の交流が盛んになれば、「会社員」というのは通用しない。
今日では「会社に対する忠誠心」を求める時代ではなくなった。チームとしての行動の結果として、仲間を愛し会社を愛するというのは良いが、その前に「仕事に対する忠誠心」を求めるべきである。自分の仕事に対して、誇りも忠誠心も持てないようでは、この先、憂慮すべき事態に陥る危険が高い。

今月の

一言

スマイルズの『自助論(セルフ・ヘルプ)』は、非常に分かりや

「歩みののろいカメラでも、正しい道さえ通れば、まちがった道を行く競争相手に勝つことができる。だから、熱心に努力さえしていれば、進歩が遅くとも気に病む必要はない」
(サムエル・スマイルズ)

すく示唆に富んだ本で、私自身、時々書庫から引きだして、自分の位置を確認するつもりで読んでいます。ちょうど、ジャイロスコープの役目を果たしてくれそうです。
ところで、こんな仕事をしている、私の説明に對して飲み込みの早い人もいれば、遅い人もいます。もちろん、私の説明の拙さも原因しているのですが、やはり進み方の遅い人がいます。こ

です。カメラならカメラのやり方がありません。間違つてもウサギのまねをしないことです。倦むことなくコツコツと自分の選んだテーマに取り掛かっています。乗効果もでてきますので、もっと増えているでしょう。ちょうど「積立預金」の利息みたいなもので、す。ゴールはずっと先です。

組んでいけばいいのです。自分の仕事効率がよくあつていけば、社会に役に立つはずですよ。それが「正しい道」であり、その方向に熱心に取り組めばいいのです。
一月経って、確実に何ができるようになっていけばいいのです。そうすれば一年で十二もの「できること」が手に入っているのです。五年経てば、それが六に増えているのです。その頃には「相乗効果」もでてきますので、もっと増えているでしょう。ちょうど「積立預金」の利息みたいなもので、す。ゴールはずっと先です。