新人技術者のための

ロジカル・シンキング入門 第3回

冴木 元

システムの記事　ビギナーズ

※本連載記事の第1回と第2回は，本誌2006年6月号，pp.50-59の特集1 第4章「いかにしてバグの原因を突き止めるか」，pp.60-66の特集1 第5章「プログラミングにおける良いデータ構造とは」として掲載されました．

ここでは設計書を作成する際のポイントについて述べる．新人研修などではきちんと設計書を作成してからコーディングを行うように指導されるが，実際の製品開発では設計書の作成が軽視されていたり，きちんと保守されていないケースがある．本稿では設計書に求められる要件について説明する．また，設計書の表現形式として，ブロック図やフローチャートを利用する際の問題についても議論する．（編集部）

Cさんはある携帯音楽プレーヤの開発に携わっています．携帯音楽プレーヤは，最近話題の人気商品であり，各メーカは開発にしのぎを削っています．そんな中，Cさんもここ1年ばかりソフトウェア開発に奮闘してきました．

「仕様が定まらない」，「開発を始めるとハードウェアが動かない」，「機能ブロックを結合すると，システムがハングアップしたり，音が出なくなったりする」など，お決まりのトラブルをくぐり抜けてきたCさんにも，開発がひと段落して，ホッとひと息つく時間ができたようです．

…と思っていたら，上司から以下のような要望が出てきました．「1次製品の開発中のバグは，その数の多さもさることながら，バグが報告されてから解決するまでに時間を費やしたことが問題視されている．2次製品の開発に向けて，各機能ブロックの担当者には，設計レベルからの見直しが必要なのではないか」．

そこでCさんは，各機能ブロックの担当者と連絡をとり，現状，どのような資料が残っているのかを調べてもらうことにしました．そこで判明したのは，携帯音楽プレーヤ全体の仕様書や，各機能を最初に割りふる際に議論した議事録のたぐいは残っているのに，設計書と呼べるものがほとんどないことでした．「設計書？ 仕様書なら見たことあるけどね」と，別段気にもとめないツワモノも数多くいました．一部の機能については，コーディングの前にフローチャートをまとめた人がいるということだったので，それを見せてもらいました．しかし，その中身は現在の構成とはかなり異なるものでした．というのは，途中で加えられたバグの修正や仕様変更がそのフローチャートに反映されていなかったのです．

Cさんの報告を聞いて業を煮やした上司は，次のように断言しました．「2次開発では，設計段階で，各機能ブロックの担当者に詳細な設計書の提出を要求する．それらのレビューを終了してからでないと，コーディングに入ることはまかりならん」．

確かに「設計」の工程はお世辞にも充実していたとは言えず，設計の見直しが必要という上司の指摘は正論であるには違いありません．しかし，問題はそれに必要な工数です．開発対象の携帯音楽プレーヤは，全体で数十万ステップの規模を持っていました．それらを詳細なフローチャートに起こしてレビューし，しかる後にコーディングを行い，発生したバグの修正や仕様変更を正確に反映しようとしたら，どう考えてもコーディングに必要と考えられる期間の2倍程度の工数を上乗せしないと達成できません．2次製品の開発に用意された期間からいって，これは不可能です．

詳細な設計書を用意しろといっても工数的に難しいものがある，とCさんはおそるおそる上司に告げましたが，上司はさらなる正論で追い討ちをかけてきました．「そもそ

KeyWord　設計書，ブロック図，フローチャート，コーディング，ソフトウェア設計，データベース，携帯音楽プレーヤ，DMA，アプリケーション・サーバ，インターフェース

【理想】
- 設計段階で問題点の早期発見に努めること
- コーディングとは，設計書を忠実に移し変える作業にほかならない
- 名人芸に頼ったコーディングからは，安定したソフトウェアは生まれない

【現実】
- ソースが設計書でしょ．やっぱし…
- 納品前に整えれば客は黙らせられるさ
- 最近はソースからドキュメントを起こすツールがあるでしょ

図1　設計書をめぐる理想と現実のへだたり

これは今に始まったことではない．ソフトウェア工学のたいていの本には設計段階における作り込みの重要性が強調されているし，企業の新人研修などでも，「いきなり（キーボードを）打つな」とかならず言われるのだが…

も設計という工程が必要なのは，コーディングする前に問題点を洗い出してしまうことにある．理想のコーディング工程というものは，設計書を機械的に移し変える作業になるはずだ．新人研修でそんなことを言っていたのはおまえ自身じゃなかったか？ おまえにしては珍しくいいことを言うなと思って聞いていたのに，みずから覆してどうする」．

上司の正論を実行に移すために，Cさんに可能な手立てとして，どのようなものがあるのでしょうか（**図1**）．

●「言うはやすし，行うは難し」の設計書作成

Cさんのように，「ソフトウェア設計」の工程のたいせつさがわかっていながら，いざそれを実行に移すとなると，どのように設計書を作ったらいいのかとまどう人は意外に多いようです．確かに新人研修などでは，だれでも設計書を書かされます．そして教師役の先輩社員から，「趣味でプログラムを書くのではないのだから，設計書抜きにいきなりコーディングしないこと」と言われ，フローチャートを詳細に書かされてからコーディングさせられるものです．

しかし，実際に配属されて実務に携わるようになると，「設計書」と呼ばれるドキュメントが用意されることが驚くほど少ないことに気づきます．とくに，Cさんの上司が述べた正論のように，設計書の作成がコーディングの前に行われる，というのはそれほど多くありません．納品の前に，ドキュメント整理のために初めて設計書を起こし，コーディング開始の前に作ったかのように日付を記入して納品した経験のある人が，少なからずいると思われます．

ここで，設計書が作られない理由の典型として，以下のようなものがあります．

- 時間がない

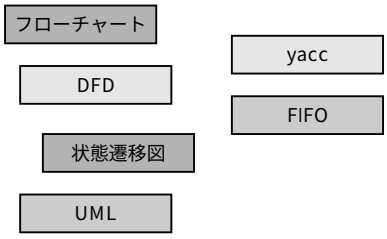

図2　ソフトウェア設計のフォーマット

いにしえから数あるソフトウェア設計のフォーマット．ものの本を調べると，昔からあるフローチャートから最近のUMLまで，いろいろとひな型らしきものがあるにはある．しかし，いかなるシステムにも適応可能な万能のフォーマットは，残念ながらなさそうだ

- 仕様変更が入れば作り直しが必要
- 作ってもだれも見ない

とくに，「時間に制約があり，作っていられない」というのが，理由としてもっとも多いようです．それだけが理由なら，短時間で作成できて，後工程の土台となりうる設計書の表現形式（フォーマット）が決まれば，解決するはずです．ただし，問題はその中身でしょう．

失敗したプロジェクトを調べてみると，「ソフトウェア設計の段階の検討が不十分だったため，後のテスト工程で手戻りが生じた」と言われることがよくあります．そして，実際に設計に携わっていた人に話を聞いてみると，まちがいなく上に挙げたような返答が得られるものです．設計工程の重要さを説く人は多いのですが，「必要な設計書をいかにして作るか」ということについては，答えに窮するケースが少なくありません．

設計書を用意するにあたって，これらの障害があることは無視できません．確かに，短時間に作成できて，後工程の土台となりうるフォーマット（**図2**）があれば，「時間がなくて作れない」といった障害は克服できるでしょう．しかし，設計書を作るうえでいちばん考えなければならないのは，そもそも「どのような情報が設計書に整理されているべきか」ということではないかと筆者は思います．設計書に必要な情報がきちんと整理されていれば，フォーマットはそのときどきで選択できます．逆にフォーマットだけを決めてしまうと，不慣れなシステムを開発するときに足元をすくわれることになりかねません．経験値がマイナスに作用する，ということはどんな場合にも起こりえます．

● 設計書に求められる四つの項目

Cさんたちに作ってもらえそうな設計書を考える前に，まず，「チーム開発において必要な設計書に求められるも