

第 1 章

ソフトウェアトラブルを予防する「形式手法」とは？

この章では、ソフトウェア開発現場で使われている伝統的手法の問題点を述べ、形式手法と形式仕様記述言語の概要と利点を説明し、形式手法適用の効果を紹介する。

ここでいう伝統的手法は、現在、日本のソフトウェア開発現場で使用されている手法を主に指す。構造化分析・設計手法やオブジェクト指向分析・設計手法といったソフトウェア工学的手法も、まれには用いられているが、ほとんどの開発現場では、それ以前、すなわち 30 年以上前の手法を依然として使い続けている。

構造化やオブジェクト指向を使っているという開発現場でも、実態は伝統的手法と大して変わらない使い方をしているところも多いので、日本のソフトウェア開発現場では、伝統的手法すなわち構造化以前の手法が相変わらず主流である。

どうして伝統的手法が問題なのかを以下に見ていこう。

1.1 伝統的手法の問題点

伝統的手法の問題点は、現象面から見ると以下の通りである。

第 1. ソフトウェアトラブルを予防する「形式手法」とは？

1. 開発と保守の費用が高い
2. 欠陥の発見が遅れ、手戻りが多くなる
3. プロジェクト管理が困難で、完成が遅れることが多い
4. 再利用や保守が困難である
5. 仕様が最新でなく、信用できない

形式手法の立場から見た伝統的手法の問題点は以下の通りである。

1. 日本語などの自然言語は曖昧さを持つため、仕様が曖昧になる
2. 仕様の検証はレビューに頼るしかない
3. 自然言語仕様は、修正した時の影響を把握しにくいいため、仕様の欠陥が増える
4. 自然言語では、仕様を洗練して実装していく方法論も道具もない

伝統的手法の問題点は、「日本語仕様の曖昧さ」と「欠陥発見の遅れ」に要約できる。すなわち、日本語仕様が曖昧なため、仕様に多くの欠陥が含まれ、レビュー以外の検証手段がないため、その発見が当該工程よりかなり後の工程で行われる。

以下に、「日本語仕様の曖昧さ」の実例や、「欠陥発見の遅れ」を示すデータを示す。

1.1.1 日本語仕様の曖昧さ

日本語仕様の曖昧さを示す例を 2 つ示そう。

最初に、2 章で説明する「図書館システムのモデル構築例」の日本語要求仕様の一部を示そう。

1. 本を図書館の蔵書として追加する
2. 題名で本を検索する

たった 2 行のこの仕様に、曖昧さがある。

要求の 1 番目の「本」と、2 番目の「本」は、異なるものと解釈した方が、図書館のシステムとしては良いのだが、この仕様では同じ名前がついているの

1.1. 伝統的手法の問題点

で、誤解を招くのである。実際にこのプログラムを作成したプログラマが適切でないプログラムを作ってしまったことがある。

「本を図書館の蔵書として追加する」仕様は、図書館を知っている人間にとってさほど解釈のずれは無いだろう。問題は「題名で本を検索する」である。1つの本につき1冊だけ蔵書としている図書館なら問題ないのだが、そのような図書館は無い。1つの本を何冊か蔵書として持っているのが普通であろう。ここで検索結果として表示するのは図書目録に載る「本」なのか、物理的な1冊ごとの「蔵書」なのか、あるいは両方なのかを、上の要求仕様は区別していない。従って、読む人によって異なる実装としてのプログラムができる可能性がある。

さらに、「蔵書」という言葉もこの場合適切でない。「蔵書」になる前の「蔵書」は、何と呼んだら良いのだろうか？ そう、「蔵書」は「本実体」のある状態を表しているに過ぎない名前なのである。もちろん、ここで「本実体」という名前が一番良いかは議論しなければならないが、少なくとも、当初の「本」や「蔵書」よりは適切な名前であることは確かであろう。

一応、曖昧さがあまり無い日本語要求仕様は以下のようになる。

1. 本実体を図書館の蔵書として追加する
2. 題名で本を検索する

次に、実際の証券業務の用語定義の曖昧さの例を示す。

信用取引の決済日（期日）を得ます。弁済期限とは、信用建玉^{たてぎょく}に対して当社がお客様に信用を供与する期限をいいます。弁済期限は、現在のところ6ヶ月のみを取扱っています。弁済期限が6ヶ月であるということは、信用建玉の建日（信用建玉が約定した日）の6ヶ月目当日が信用期日となり、この日を超えて建玉を保有することは法律で禁じられています。信用期日が休日の場合には、直近の前営業日が信用期日となります。

この用語定義の曖昧な点は、以下の通りである。