

第6章

組み込みソフトウェア・テスト・ クライシスの「傾向と対策」

——「品質」をつねに念頭に置きながら
ソフトウェアを開発する

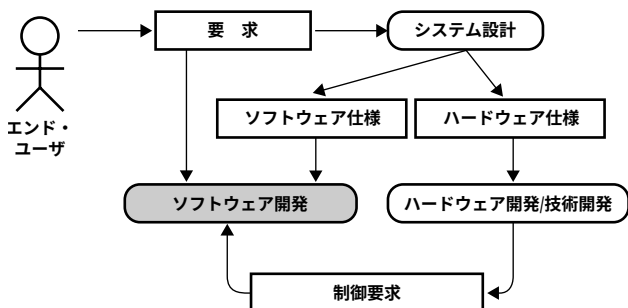
杉浦英樹

ここでは組み込みソフトウェアの品質確保の問題について解説する。組み込みソフトウェアは、エンド・ユーザからの要求だけでなく、システム設計やハードウェア設計の要求も反映させなければならない。そのため、組み込みソフトウェアの開発は複雑になり、そのスケジュールはしばしば遅れがちである。ソフトウェア開発者は、テスト担当者と協力しながら、ソフトウェアの品質をつねに意識して作業を進める必要がある。

(編集部)

組み込みソフトウェアの開発では、要求元からのコンセプトや要求を受けて、システム設計によって選択された技術をもとに、システムを実現するためのソフトウェアを作成します。組み込みソフトウェアに対する要求には、「機能そのものに関する要求」と「機能を実現するハードウェアの制御方法に関する要求」があります。

組み込みソフトウェアの開発では、製品の市場投入期間の短縮が重視されます。しかし、図1に示すような構図があるため、組み込みソフトウェアに対する要求の決定はど



【図1】 組み込みソフトウェアの要求の構図

エンド・ユーザからの要求、システム設計からの要求、ハードウェアの制御要求を受けて、組み込みソフトウェアを開発する。それぞれの要求の整合性のチェックはソフトウェア開発の中で行われる。

うしても遅れる傾向にあります。

●ソフト開発現場が陥っているジレンマ

昨今は、組み込みソフトウェアの欠陥が原因で最終製品の回収や交換が発生しており、ソフトウェアの品質の問題が取りざたされています。実際のところ、製品開発の現場では以下のような状況が繰り返されています。

- ソフトウェアの制御仕様が決まらないため、機能の実装が計画どおりに進まない
- 実装する機能が増加し、機能の実装が不完全なままテストを開始せざるをえない
- テストを実施したときにソフトウェア品質の問題が発生し、テストが先に進まない
- テストを先に進めるために打った手が新たな要求や制約となって、機能の実装が先に進まない
- 機能の導入が遅れ、当初の製品化のスケジュールを守れない
- 製品の市場投入時期を優先させて、一部の機能が欠如したまま市場に製品を出荷する
- 品質確認が遅れ、製品の市場投入後に市場で問題が発覚する

こうした八方ふさがりの状況で、ソフトウェア開発者が最終成果物である製品ソフトウェアの品質を向上させるにはどうすればよいのでしょうか。そもそも「ソフトウェアの品質」はどうやって測定すればよいのでしょうか。

●「ソフトウェアの品質」とは何か

ソフトウェアに求められることは、要求されたとおりの動作を行うことにつきます。昨今の複雑な制御システムを実現するためには、要求分析を行って要求の抜けや漏れなどの欠

陥を予防しなければなりません。さらに、異常が発生した場合でも安全に対処できるようにしておく必要があります。

こうした要求にこたえるため、効率的な開発を行うことはもちろん、開発担当者自身が自分の開発するソフトウェアの品質をつねに意識し、自分なりの品質基準を持つ必要があります。ソフトウェアの品質の測定は、開発対象となるシステムの品質基準に照らし合わせて、開発者自身がくふうを凝らすべきところです。

例えば、キー入力されたデータを2倍して、結果をレジスタに格納するプログラムを作ったとします。そのプログラムは、起動条件(キー入力)とデータ(入力されたキーのコード)から結果を算出し、その結果をレジスタに格納します。ここで、「すべてのキー入力」に対して要求どおりの計算結果がレジスタに格納できていれば、そのソフトウェアの品質が確保されているということになります。すべてのキー入力と計算結果が定義されていることが「要求仕様の品質」であり、要求仕様どおりに計算結果がレジスタに格納されることが「ソフトウェアの品質」です。

ソフトウェアの品質を測定するということは、開発したプログラムについて、要求仕様と合っているかどうかを一つ一つ確認するということです。管理の面では、仕様内容を確認(レビュー)し、仕様の数(テスト項目数)を用いて確認結果(品質状況)を測定します。

次に、ソフトウェア開発のプロセスに着目すると、ソフトウェアの品質に大きく影響する工程は、ソフトウェア開発の中心となる「要求を実現する作業」、つまり設計工程です。設計工程においてソフトウェアの品質が低下する要因としては、以下のことが挙げられます。

- 分析した結果をもとに設計していない(分析工程で明確になっていない内容について、安易に再分析して、実装

してしまう)

- 発生する制約や問題についての設計方針がないまま、場当たり的に設計してしまう
- 要求にない状況が起こったときの動作について、十分に検討しないまま実装工程に移行してしまう
- 関係者の間で十分なレビューを行わないまま、設計を終了してしまう

●エンド・ユーザの使用場面まで想像して品質をチェック

ソフトウェア工学では、昔から「テストにパスするためのソフトウェアをどう作るか」ということを追及してきました。

ソフトウェア開発者自身がテストを行う場合、発見した欠陥についての考察を設計にフィードバックする(同じまちがいを繰り返さないためにくふうする)ので、テストにパスするためのソフトウェアの開発に役立ちます。ソフトウェアの品質を設計段階から向上させるこの考えかたは、本質をついた方法と言えます。図2に開発工程と品質確立工程(テスト工程)の関係を示します。

では、設計品質を引き上げるために、どのような点を意識して設計を行えばよいのでしょうか。これは、基本的な開発作業を見直すのが早道です。前工程(上流工程)と後工程(下流工程)を考慮しながら開発作業を進めるのは当然のことですが、特に後工程については、考慮する範囲をできるだけ広げて、エンド・ユーザが使用する場面まで想像して、品質のチェック・ポイントを設定するとよいでしょう(図3)。

●ソフト開発者自身によるテストには限界がある

ソフトウェアの品質を確保するには、そのソフトウェアを使用する現場で発生しうる欠陥をあらかじめ予測し、その欠陥に対する対処を上流工程で済ませておくことが理想

【図2】
開発工程と品質確立工程

分析から実装までが開発工程であり、開発工程ごとに実装後のテスト工程を対応させることで品質を確立する。

