

# 組み込みシステム開発における要件開発・管理の適用

ポーランド株式会社 プロフェッショナルサービス本部  
プリンシパル・コンサルタント

今村 智

## はじめに

今更申し上げるまでも無いことですが、カーナビ、家電、携帯電話、OA機器などに代表されるような組み込みシステムにおいても、これまでの「ハードウェアが主でソフトウェアはその従属物」という認識から、ソフトウェアアプリケーションの重要性の増大へと、状況がシフトしてきています。携帯電話などは既にPCソフトウェアの開発と同等、あるいはそれ以上の規模のものとなっています。

そのような背景によるものなのか、近年では、組み込みシステム開発をされている方からの要件管理、構成管理といったものに関するご相談を受ける機会が増えてきています。

そこで本稿では、簡単ではありますが要件管理というものの効果、現状についてお伝えいたします。

## 1. システム開発は組み込みもエンタープライズ、パッケージソフトも同じ

組み込みシステムであっても、エンタープライズシステムであっても、それを開発することは、システムに対する要件が存在し、それをシステムとして具現化するための活動であることに違いはありません。そして、その開発期間の短期化が求められていることも、要件が決まりにくく且つ変更されやすいものであることも同じです。(図1参照)

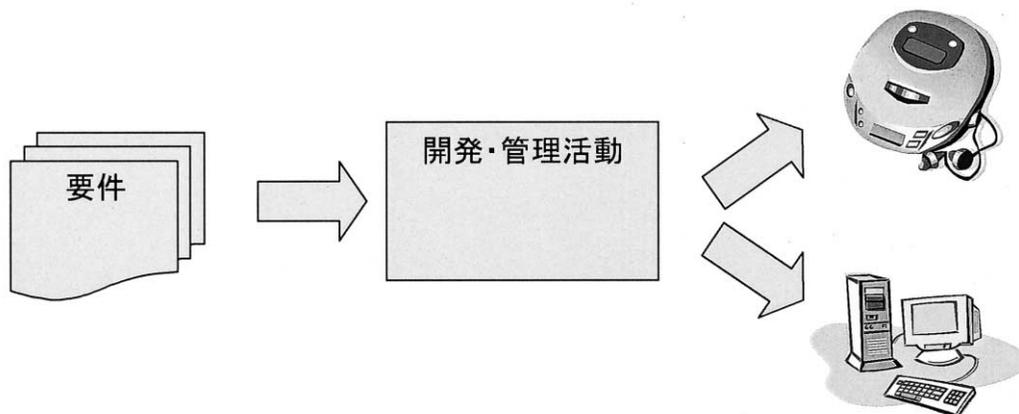


図1 システム開発の簡略表現

## 2. 要件とは何か?

“要件”と一言でいっても、その意味するところは人や組織、業種によりまちまちであることが多く、共通の認識が確立されているとはいえません。そこで、本稿では便宜的に以下のように用語の意味を定めることにします。

要件：システムに求められる“何か”を定義したもの

仕様：求められた“何か”を、システムが“どう”実現するかを定義したもの

このように定義すると、当然のことながら要件と仕様の間には明確な依存関係が存在します。

それは、「要件が変われば仕様も変わる（変えなければならぬ）可能性がある」ということを意味しています。さらに、話はこれだけでは終わりません。

要件だけを見ても、そこにはいくつかの依存関係が存在します。（図2参照）

組み込みシステムの場合は、この機能要件、

非機能要件を実現するための仕様として、ハードウェア（メカ、チップ）仕様とソフトウェア仕様というものが決められていくこととなります。

ここで、複写機の“コピーをする”という機能要件を例にとって、単純ではありますが構造化してみたものを図3に示します。（図3参照）

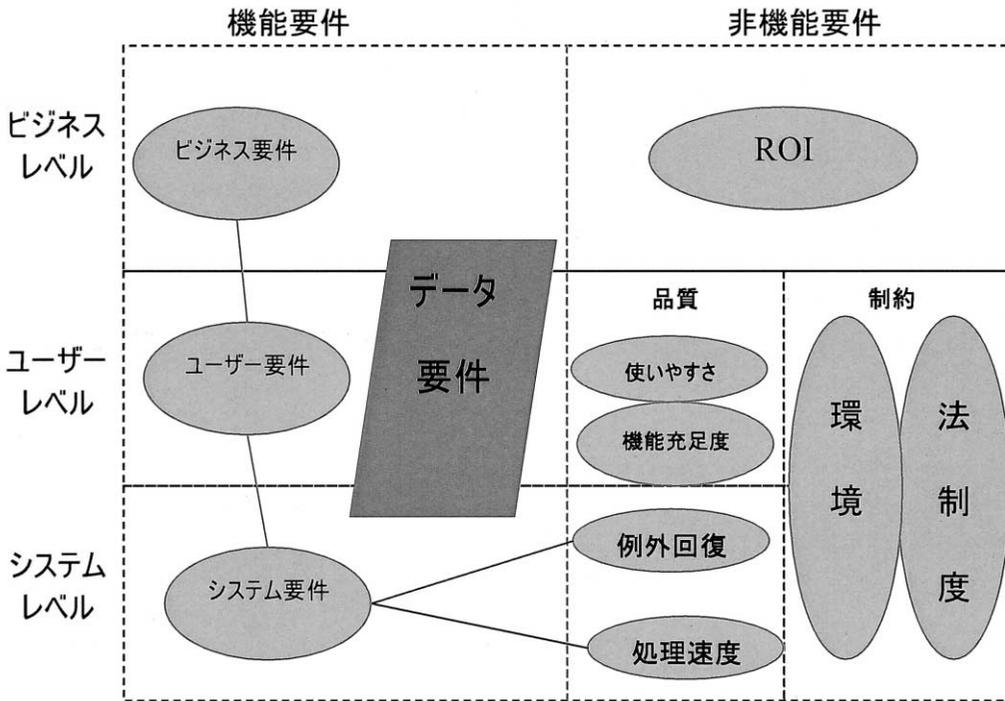


図2 要件の構造例

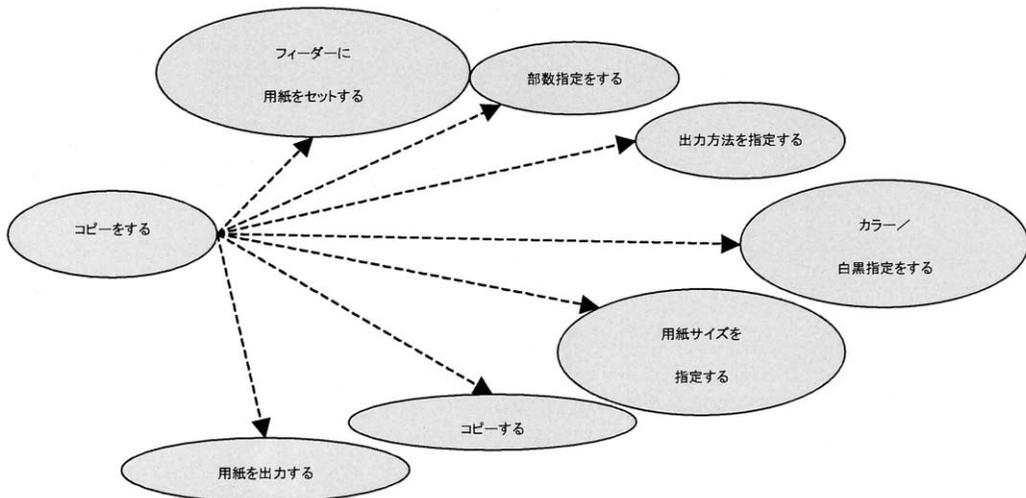


図3 機能要件の構造例

図3で「コピーをする」という機能要件に対して、“どうやって？”という問いかけをすることによってもう一段階システムに近い要件が導出されます。このようにして下位の要件の導出を行っていくことによって、最終的なハードウェア仕様、ソフトウェア仕様の漏れ、ダブりの少ない導出が可能となります。

### 3. 要件管理とは何か？

ここで少し堅い話になりますが、CMMIでは要件管理として以下のプラクティスについて述べられています。

- ・要件の理解を獲得する
- ・要件に対するコミットメントを獲得する
- ・要件変更を管理する
- ・要件に対する双方向の追跡可能性を維持する
- ・プロジェクト作業と要件の間の不整合を特定する

また、「要件から、その導出要件に至る追跡可能性と、要件、機能、オブジェクト、人員、プロセス、成果物の割り当てに対する追跡可能性を維持する」ということも述べられています。

つまり、要件管理とは、要件に対する合意を形成し、その後の変更を管理する。そして、変更による影響を適切に把握できるよう、要件を起点とした追跡可能性を維持、管理することになります。

それでは、これらを実現するに際して、効率化が可能な部分はどこになるのでしょうか？

合意形成は、コミュニケーション効率を上げることで若干の効率化は図れるかもしれませんが、例えば2日かかっていたものが5分で終わるようにはなりません。

それは、複数の利害関係者との交渉や判断といった、効率化が困難な要素の占める割合が高いからです。

それでは、変更管理と追跡可能性はどうでしょうか？

これらを、全て人手でやるとすれば、作業の信頼性、作業にかかる手間という点で割に合わないと感じられることも多く、実際にある程度の規模を超えるプロジェクトでは、これらを人手できちんとやり遂げているところは見たことがありません。

実際に、みなさんがご担当なさっているシス

テムの要件は、導出元、導出先の関係が明確になっているのでしょうか？

更には、仕様との関係は明確でしょうか？

更には、設計ドキュメント、コード、テストケース、テスト結果、変更要求との関係はいかがでしょうか？

おそらく、これら全てをスプレッドシートなどを使って定義、管理しようとする、とても大変なことは容易に想像できると思います。

そして、これが出来ていれば現在のプロジェクトの混乱の多くが解決するという予感もお持ちではないでしょうか？

現状では、いくら漏れなくダブリもないように頑張っても要件を構造化して整理したとしても、その導出関係、依存関係の維持管理ができず、結果として努力と工数を無駄にしているといえます。

そうした場合には、要件管理に特化したツールを利用することで、手間と信頼性の問題のほとんどは解決可能となります。その結果、今までなかなか実践できていなかった要件管理が実現可能なものとなります。

要件というシステム開発活動への入力情報の品質が確保され、変更による影響範囲が適切に把握されることによって、プロジェクトスケジュール、コストの適切なコントロールも可能となります。

### 4. おわりに

開発の現場では、まだまだ設計、実装、テストという開発活動における改善の余地があるのも事実です。しかし、その開発活動の原点である要件にまつわる問題、課題の改善を行うことのレバレッジ効果はとても大きいことをご理解いただけたと思います。特に、組み込みシステム開発の場合は、要件開発および要件管理の問題による影響が、ハードウェア、ソフトウェアの両方に及ぶこととなりますから、その開発規模がエンタープライズシステムやパッケージソフト開発と同等の場合、それらを適切に行うことによる効果もより一層大きくなると言えます。

ポーランドが、要求管理ツールCaliberRMや構成管理ツールStarTeamをご提供し始めて以来、組み込みシステム開発のご担当の方とのコンタクトが増えているのは、そうした認識が広

まってきた証の一つだと思っています。