

## I ソフトウェア品質の考え方

ソフトウェアの開発は、プラント建設など受注生産型のハードウェアの開発と大きな差はないし、その品質管理の方法も基本的に変わるものではないが、ソフトウェアとしての特質上、ハードウェアの品質技術をそのまま適用することが困難な点が幾つかある。

### 1 ソフトウェア管理の難しさ

ハードウェアの開発に比べてソフトウェアの開発の難しさには以下のようなものがある。

#### (1) 技術的蓄積

ハードウェアの設計の妥当性は、工学的原理すなわち自然科学法則によって支援される。自然科学法則の利用の仕方さえ間違えなければ製品の品質は自然に確保される。これらの工学的原理を組織の技術標準あるいは個人の経験として蓄積することによって品質を確実なものとするとともに、設計・製造の効率化が促進される。これに対して、ソフトウェアの品質を支配する要素は論理構造の無謬さであるが、これは大部分人間の能力にかかっており、不安定で技術蓄積にも限度がある。

#### (2) 製品を記述・伝達する手段

ハードウェアにおいて製品を記述する手段として図面がある。そして製品が複雑な場合は拡大あるいは縮小した模型を作成することができる。しかしながらソフトウェアにおいては、文書、フローチャート等の方法が用いられるが、これらの方法は製品を定義し、その内容を伝達する手段としては図面や模型ほど有力ではない。

#### (3) 連続性

ハードウェアは品質特性として公差が定めており、その公差内であれば（場合によっては公差から外れていても）、品質特性のばらつきが許容される。また、多くの管理項目の中でマイナーなものについては、多少の手抜かりがあっても品質に大きな影響を及ぼすことはない。しかしながらソフトウェアでは、ごく僅かと思われる食い違いが致命的なダメージをもたらすことが少なくなく、重要管理の原則の利用が難しい。

#### (4) バックアップ

ハードウェアの開発はまず図面が作られ、それに従って製品が作られる。この製造の過程で設計の不具合の多くが発見され是正されるのに対し、ソフトウェアではこのバックアップがない。

### 2 ソフトウェア品質管理

前述のような困難を克服することは容易なことではないが、ソフトウェアの品質管理はハードウェアのそれと同様で基本を確実にやることである。

(1) 他人にわかりやすいプログラムを作ること

他人のプログラムを理解することは骨の折れる仕事である。プログラムは開発担当者の私物ではない。チームで開発を行っている場合、他人が見てわかり易くできているかどうかが開発及びデバッグの効率に大きな影響を及ぼす。また、プログラムのメンテナンス、エンハンスメントは多くの場合、開発を担当したグループと別のグループによって行われる。他人にわかり易いプログラムを作ること、プロのプログラマーとして欠くことのできない要因である。

(2) 標準化、再利用を考慮する

標準あるいは素性のおかった既製の製品を用いることは、開発の効率、信頼性の向上に極めて有効である。プログラムの開発に当たっては、常に再利用を念頭に置き、適用範囲、インタフェースの条件を明確にしておくことが大切である。

(3) 変更管理を重視する

開発の最初の段階で仕様をできるだけ充実したものにしておくことが必要なことはいうまでもないが、現実には開発の途上で仕様変更が行われるのが常である。この原因は、必要なものを同時に把握する能力が人間に欠けていることと、人間の情報伝達能力が完全なものでないことにある。多くの場合、仕様は要求事項を発見していく時間的経過を経て決定される。変更を減らすとともに変更管理を徹底することが品質管理の面から極めて重要である。

(4) 効果的テストを実施する

ハードウェアに比べてソフトウェアは品質をバックアップする要素が少ない。したがって、品質管理においてテストの占める割合はハードにおけるよりもずっと大きい。ソフトウェアのテストには開発部門の立場からのブラックボックステストがあるが、テストに使われる時間は通常開発総工程の30%から50%を占めている。テストを有効なものにするためにテスト計画及び品質管理が必要である。

(5) バグの記録と解析を活用する

テストあるいは市場で見出されたバグに対しては、単にそれを手直しするだけでなく、発生の原因を究明し再発防止の手段を講じていくことは、ハードウェアと同様ソフトウェアにおいても大切である。