

移植阻害要因の定量化手法の検討

7H-6

林 智定、郷原純一、岩田憲和

NTT情報通信処理研究所

1. はじめに

ソフトウェアの再利用の観点から、ソフトウェアの移植への需要は高まりつつあり、移植工数の見積り等、開発管理技術の研究も盛んに行なわれるようになってきている。著者らは、開発管理手法として広く評価を得ているCOCOMO [1] の移植工数見積りモデルをベースに、さらに高精度な移植工数見積りモデルを検討中である。

本稿では、一連の移植工程における、(1) 各工程の移植工数へ大きな影響を与えるファクタの分析、(2) 移植性のチェック～修正箇所的设计工程で大きなファクタとなる移植阻害要因の分析と重み付け について報告する。

2. 移植工程の分析

我々は、①同一インタフェースを意図して開発された異なるOS間の移植、②C言語により記述されたソフトウェアの移植、③移植を前提に作成されたソフトウェアの移植、④ツールによる移植阻害要因の事前抽出が可能という前提条件の下で、一連の移植作業を、工数という観点から6つの工程に分割した。各工程の概要を以下に示す。[2]

- (1) 移植性のチェック工程 (PC工程)
ツールおよび手作業による移植阻害要因の抽出、プログラムの移植の可否の判断。
- (2) 修正箇所的设计工程 (D工程)
PC工程で検出された移植阻害要因に対する具体的なコーディングイメージの決定。
- (3) コーディング工程 (C工程)
D工程での設計に基づくコーディングおよびコンパイルの実施。
- (4) 関数試験工程 (T1工程)
プログラムの記述単位である関数の単体試験。
- (5) モジュール試験工程 (T2工程)
複数の関数をまとめたモジュール単位の試験。
- (6) 総合結合試験工程 (T3工程)
最終的な総合試験。

3. 移植阻害要因の移植工数へのインパクト

ソフトウェアの移植に要する工数は、上記の各工程毎の工数の総和で表わされるが、これらの各工数は、以下に示す3点に依存する。

- ① 個々の移植阻害要因の持つ単位修正工数
個々の移植阻害要因に対する修正方法および修正の難しさは、基本的には各々異なる。
- ② 移植阻害要因の数
少ない方が対処は容易である。

③ 移植阻害要因の分散度

プログラム中にどの程度集中して存在するかを表わす。集中している方が対処が容易である。これらが、各工程へ与えるインパクトは、概ね以下の通りである。(表1参照)

- (1) PC工程
各移植阻害要因の単位修正工数、数、および分散度のすべてが本工程へ大きく影響する。
- (2) D工程
各移植阻害要因の単位修正工数が大きく影響する。
- (3) C工程
要因の数が影響する。
- (4) T1工程
移植阻害要因が存在する関数のみが試験対象となるので、要因の分散度が大きく影響する。
- (5) T2工程
T1工程で試験の対象となった関数および、該関数と機能的に関係のある関数が試験対象となる。従って、要因の分散度が大きく影響する。
- (6) T3工程
全プログラムが試験対象となるので、インパクトはそれ程強くない。

以上の通り、PC～C工程では、各移植阻害要因の単位修正工数が、T1～T3工程においては移植阻害要因の分散度が最もインパクトが大きい。すなわち、試験工程を除く各工程の工数を見積るには、個々の移植阻害要因の単位修正工数を明確にする必要がある。

4. 移植阻害要因の分析

著者らは、OSのある機能ブロックの移植作業を行なっている。その過程で抽出された移植阻害要因に対して、移植阻害要因となった原因という観点から分類し、さらに修正に要する工数という観点から評価を行なった。以下にその考え方を、表2に分類結果と評価結果を示す。

4.1 移植阻害要因の分類

移植阻害要因に対して、次の4つの観点から分類を行なった。

- (1) アーキテクチャの相違に起因するもの
移植元と移植先のCPUのアーキテクチャの相違に起因する移植阻害要因である。具体的には基本語境界への調整法の相違等である。これは、例えば変数をメモリ上に配置する際に問題となるもので、基本語境界への調整法が異なった場合、必ずしもコーダーが期待している通りには、配置されない場合が有り得る。

Quantification method of obstructive factors to software conversion

Tomosada HAYASHI^{*1}, Junichi GOHARA^{*1}, Norikazu IWATA^{*1}^{*1}:NTT Communications and Information Processing Laboratories

- (2) 言語処理系の仕様の相違に起因するもの
移植元と移植先との言語処理系の仕様の相違に起因するものである。具体的には、`#define` の数等の制限値を越える場合である。
- (3) OSの仕様の相違に起因するもの
移植元と移植先のOSの仕様の相違に起因するものである。具体的には、エラー情報等のインプリメントに依存するもの、あるいはシステム個別機能を使用している場合等である。
- (4) 開発環境の相違に起因するもの
移植元と移植先のファイル配置の相違に起因するものであり、具体的には、インクルードするヘッダファイル等を絶対パスで指定している場合である。

4.2 移植障害要因の評価

各移植障害要因に対して、修正に要する工数という観点で評価を行なった。修正に要する工数はさらに、以下の2つの観点到に細分することができると思われる。抽出された移植障害要因に対して経験的に、これら2つの観点から3段階で評価を行い、評価結果を基に総合評価値を与えた。総合評価値は、抽出された移植障害要因の中で最も簡単に修正が可能なものを1とした場合の相対値で与えた。

- (1) 修正に要するスキル

単なるコーディング技術のみならず、移植元および移植先双方の言語仕様、アーキテクチャ等に関する知識の必要性である。

- (2) 他への波及度
ある障害要因を修正する際に、他の部分への修正も必要な場合である。

5. おわりに

ソフトウェアの移植工数見積り手法の検討の一環として、個々の移植障害要因に対する定量化を試みた。これは、一連の移植作業の中で、主に移植障害要因に対する工数を見積る際に有効となるものである。今回の検討により、移植障害要因に対する、定量化の基本的な方針に関しての整理ができたと考えている。今後、さらに詳細に検討を進めていき、テスト工数見積り手法と連携することにより、一連の移植作業全体の工数見積り手法を確立していきたいと考えている。

[参考文献]

- [1] Boehm, B.W.: Software engineering economics Prentice-Hall (1981)
- [2] 郷原他: テスト工数に着目した移植コストモデルの検討, 第41回情処大予稿 (1990)

表1 移植障害要因の各移植工程へのインパクト

	各移植工程へのインパクト					
	PT	D	C	T1	T2	T3
要因の単位修正工数	大	大	小	中	中	小
要因の数	大	小	中	中	中	小
要因の分散度	大	中	小	大	大	無

(注) 評価値は経験的な相対尺度である。

表2 移植障害要因の分類および評価

分類	具体例	スキル	波及度	総合評価
アーキテクチャの相違	単にアドレスを指す変数として、独自に型を定義して使用し、実体参照の際に該変数に対して不適切な型変換を行なっている。	大	大	10
	構造体等、実体への参照をアドレス相対で行なっている。	中	大	7
言語処理系の相違	<code>#define</code> の数が制限値を越えた。	中	中	5
OSの仕様の相違	システム個別機能を使用している。	中	中	5
	システム固有定数の定義が散在している。	小	中	3
	返却されるエラー情報が異なる。	小	小	2
環境の相違	<code>#include</code> 先のパス指定を絶対パスで行なっている。	小	小	1

(注) 評価値は経験的な相対尺度である。