ソフトウェアテストシンポジウム 2005
JaSST'05: Japan Symposium on Software Testing 2005

# 試験項目によるプロジェクト工数見積りに関する研究

島中　一俊　　古賀　順二　　寺澤　啓司

NTT コムウェア株式会社　〒108-8019 東京都港区港南 1-9-1 NTT 品川 TWINS アネックスビル
E-mail:　{shimanakl.kazutoshi , koga.junji , terazawa.keiji}@nttcom.co.jp

**あらまし**　開発プロジェクトの見積り根拠の候補として試験項目数に注目して研究を行った. 本研究において, 開発プロジェクトの生産管理データをプロジェクトの特性で層別して分析することにより，結合試験の量と開発工数との間に強い相関が認められた．この結果とユースケースから試験項目数を見積る技法を組み合わせる事により新たな見積りソリューションを確立できる可能性がある．本論文では，研究の調査結果の一部を紹介し，新たなソフトウェア開発プロジェクトの見積りソリューションとしての可能性について議論する.

**キーワード**　見積り，試験項目数，開発工数，回帰分析，開発言語による層別

# A study on the project effort estimation using test cases

Kazutoshi Shimanaka　　Junji Koga　and　Keiji Terazawa

NTTCOMWARE NTT Shinagawa TWINS Annex Building 1-9-1 Konan, Minato-ku, Tokyo, 108-8019 Japan
E-mail:　{shimanakl.kazutoshi , koga.junji , terazawa.keiji}@nttcom.co.jp

**Abstract**　As a basis of project effort estimation, we have studied about the number of test cases. In this study, we have found that the number of test cases of the integration testing has a strong correlation with the effort of project through an analysis of a project's metrics data which is stratified by their attributes.　We can utilize this result as a new estimation solution by combining with a technique of test cases estimation which uses use case. In this paper we report our partial practical estimate-models, and also we discuss the possibility of an effort estimation technique for software development project as a new solution.

**Keyword** Estimation, The number of test cases, Effort, Regression analysis, Stratification by development language

## 1. はじめに

開発プロジェクトのプロジェクト規模見積りは，その成功の鍵を握る重要な作業である．そして，同時に多くの矛盾を抱えた作業でもある．すなわち，プロジェクト規模を過小に見積ると必要な工期やリソースが十分に準備できず実行時の失敗リスクが増加する．一方，余裕を見て過大に見積り，結果を顧客に提示すると競争相手に負けて契約を得ることができない．そこで，見積り精度を向上するには，詳細な検討が必要となる．しかしながら，通常契約が結ばれる初期段階においては検討の前提条件さえ不明確な場合が多い．この様な性格上，見積りは熟練を要す特殊な作業と言える．そして，その様な状況を改善するために様々な見積り技法が提案されている.

以下にプロジェクト規模の根拠の種類で分類した代表的な見積り技法について，その概略と我々が抱える問題点を示す.

### 1.1. ソースコードライン数（ＳＬＯＣ）による見積り

＜特徴＞

- ソースコードライン数（以下ＳＬＯＣと記す）を開発規模として見積り，プロジェクト規模の根拠とする技法である.
- ＳＬＯＣは計測が簡単なので数多くの既存プロジェクトの生産管理データをコスト見積りモデル構築のために確保できる.

＜問題点＞

- 見積り作業の核心が多分に感覚であるために結果の精度は個人のスキルに依存する.
- 見積もり作業のナレッジ化が困難であるために技術移転ができない.
- ＧＵＩによる開発やコンポーネント化によりソー

スコード直接を書かないタイプの開発が増え，ＳＬＯＣを見積ることの自体の妥当性に疑問がある．

### 1.2. ファンクションポイント法[1]による見積り

＜特徴＞

- ソフトウェアを「トランザクション」と「ファイル」という概念でモデル化し，その量をプロジェクト規模の根拠とする技法である．
- 計測ルールが明示されているので感覚に頼らず客観的な見積りを得ることが可能である．

＜問題点＞

- 規模見積りの精度を上げるには，計測ルールを忠実に守る必要があり，計測ルールに精通した専門家が必要となる．
- また，計測ルールに忠実に従うには計測対象であるソフトウェア仕様を十分に理解する必要がある．
- 実践例が少ないために，規模見積り結果を開発工数等のプロジェクト情報へ変換するための見積りモデルの妥当性に乏しい状況である．

### 1.3. ユースケースポイント法[2]による見積り

＜特徴＞

- ソフトウェアが外部に対して与えるサービスの記述をユースケースと呼び，その量をプロジェクト規模の根拠とする技法である．

＜問題点＞

- 実践例が少ないために，規模見積り結果を開発工数等のプロジェクト情報へ変換するための見積りモデルの妥当性に乏しい状況である．
- ユースケースは計測用の概念ではなく規模を意識した厳密な識別ルールがないため規模の概念として適用できるか否か議論が分かれる．

## 2．課題

前章に示したとおり，既存の技法は各々解決すべき問題を抱えている．例えば，ファンクションポイントは，ＳＬＯＣの問題を解決すると言われている[3].しかし，普及には時間を要するので，その間技法の優位性を示す実践例を確保できないことが問題を助長している．しかも，大規模プロジェクトになるほど工期が長期化し，妥当性を向上する絶対数自体少ないので問題は深刻である．

そこで，我々は下記の目標を設定して新たな見積り技法の開発を目指した．我々の設定した目標は以下のとおりである．

- 見積りのために開発プロセスに対して可能な限り特別なタスクや特別な成果物を追加しない．
- 既存プロジェクトが残したデータを用いて独自の見積りモデルを構築する．

## 3. 試験項目数と開発工数の相関分析

既に問題が指摘されて久しいＳＬＯＣであるが[3]，いまだに一般的に見積り根拠として普及している．この理由として，ＳＬＯＣは自動計測が簡単で経験データ数が多いことに加え，プロジェクト規模との結び付きが直感的に受け入れ易いことが挙げられる．

一方，ファンクションポイントは，自動計測が困難でその成り立ちが概念的であることが普及の難しさの原因となっていると思われる．そこで，新たな見積り技法を検討する上でこれらの点に留意する必要がある．

そこで，我々はソースコードと同様に開発にとって欠くことができない試験に注目した．そして，試験項目数をプロジェクト規模の根拠として開発工数との相関関係を調査した．

ところで，試験項目数を調査するにあたり，試験項目を抽出するための試験項目設計に関する定義が必要である．そこで，我々は試験項目の概念を特定するにあたり，開発工程に注目した．本来ならば厳密に試験項目設計の定義を行ってから調査すべきであるが，限られた時間内で十分なデータ数が確保できない問題があった．そこで，現実的なデータ源を用いるために既存プロジェクトが用いている試験工程の定義を利用した．

### 3.1. 調査対象

ＮＴＴコムウェアでは，プロジェクトは全社的な標準の下で開発作業を進めており，生産管理データに基づくプロジェクト管理を義務付けている．そして，これはプロジェクト管理支援ツールに生産管理データベースとして蓄積されている．ＮＴＴコムウェアではこうして収集された生産管理データが過去１５年近くに渡って蓄積されており，そのプロジェクト数は数千に上る．

本研究の調査対象は，ＮＴＴコムウェアの生産管理ＤＢ中から，所期の目標を達成して納期内にリリースを果たした，いわゆる成功プロジェクトの生産管理データを用いた．

また，今回の調査では，リグレッションテストの影響を避けるため改造プロジェクトの分析は割愛した．このため調査対象プロジェクトは全てスクラッチからの新規プロジェクトである．

### 3.2. 試験の種別

ＮＴＴコムウェアの作業標準は，ＳＬＣＰ[4]の定義に準拠してアクティビティーが決められている．ＳＬＣＰでは，開発者の主管によって実施される試験は，５つの種別が定義されているが，ＮＴＴコムウェアではそれらを段階によって３種類に再分類している．

表１にＮＴＴコムウェアにおける試験工程の定義を示す．

表１．ＮＴＴコムウェアにおける試験工程定義

| 工程名 | 呼 称 | | 概　要 | SLCP ISO-12207 |
|---|---|---|---|---|
| 単体試験 | UT | UnitTest | 各モジュール単位の機能確認を行う | 5.3.7 |
| 結合試験 | IT | IntegrationTest | 各モジュールを通して、モジュール間インタフェース、及びモジュール間機能確認を行う | 5.3.8<br>5.3.9 |
| 総合試験 | PT | ProductTest | システムとしての総合的な機能確認を行う | 5.3.10<br>5.3.11 |

また，各々の試験工程の作業定義を以下に示す．

### 3.2.1. 単体試験(UT)

単体試験の目的は，プログラム設計工程の作業品質の欠陥を見付けることにある．単体とは，いわゆるプログラムの最小モジュールであり，コンパイル可能な最小単位である．本試験は，プログラムの内部構造に注目した試験である．そして，プログラム中の全てのステートメントを少なくとも１回は実施する，ステートメント網羅を試験項目の抽出基準とする．

### 3.2.2. 結合試験(IT)

結合試験の目的は，内部設計工程の作業品質の欠陥を見つけることにある．結合試験は，単体試験ではバグが検出されなかったモジュールを組み合わせて，各々が正しくないか若しくは矛盾していないことを確認する試験である．本試験は，プログラムの機能に注目した試験である．このため，様々なデータをプログラムに与えて期待しない結果の発生を見付けることを目的として実施する．

### 3.2.3. 総合試験(PT)

総合試験の目的は，外部設計工程の作業品質の欠陥を見つけることにある．単体試験，結合試験では検出できないバグ，の検出を目的とする．すなわち，システム全体，あるいは主要な部分を通して運用した場合に発生しない様な問題に注目する．システム試験の例として，性能試験，長期安定化試験，立ち上げ試験，エラー回復試験，等があげられる．

## 3.3. 調査結果

調査データの種別は，プロジェクト毎の開発工数，各試験工程の試験項目数，およびＳＬＯＣとした．なお，調査対象プロジェクトは，全種別のデータを残したものだけに絞った．ここで，ＳＬＯＣを調査対象データとしたのは，現在一般的に用いられているＳＬＯＣと試験項目数を比較して有用性を評価するためである．

また，開発工数は，プロジェクトの総コストを見積ることを前提に要件定義工程から総合試験工程までの総開発工数を用いた．

調査は，目的変数に開発工数を説明変数に各試験項目数とＳＬＯＣ設定し単回帰分析を行った．なお，データ群は，プロジェクトの特性によって層別し，各々に対して回帰分析を行った．そして，調査結果は回帰式とその「あてはまり」の良さを示す決定係数で示した．決定係数は，回帰式を試験項目数から開発工数を見積るために利用した時の精度の指標であり，値が１に近いほど高精度が期待できる．なお，データ群に対する層別は，本論文においてはプロジェクトが用いた開発言語の種類によって実施した．

表２．試験項目数と開発工数の相関分析結果

| 開発言語 | 決定係数 ($R^2$) | | | 回帰式 | | データ数 |
|---|---|---|---|---|---|---|
| | IT | PT | SLOC | 傾き | 切片 | |
| VB＆VC＋＋ | 0.91 | 0.71 | 0.79 | 0.03 | 17 | 31 |
| Java | 0.75 | 0.77 | 0.68 | 0.05 | 35 | 34 |
| JAVA script | 1.00 | 0.99 | 0.98 | 0.02 | 0 | 4 |
| C＋＋ | 0.73 | 0.19 | 0.64 | 0.22 | 148 | 36 |
| C | 0.28 | 0.20 | 0.55 | 0.06 | 164 | 139 |
| COBOL | 0.55 | 0.50 | 0.94 | 0.10 | 158 | 17 |
| PRO-C | 0.95 | 0.11 | 0.99 | 0.07 | 14 | 5 |
| LOTUS | 0.87 | 0.53 | 0.87 | 0.03 | 3 | 5 |
| 組み合わせ | 0.69 | 0.41 | 0.52 | 0.09 | 27 | 25 |
| 全体 | 0.41 | 0.23 | 0.60 | 0.09 | 92 | 296 |

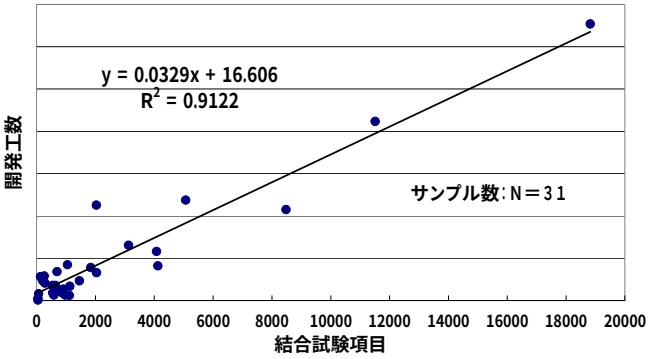


図１．ＶＢ，ＶＣ＋＋の結合試験と開発工数の関係

表２に調査結果を示す．表２において，決定係数の網掛けは，特定の開発言語の分析結果のうちＩＴ，ＰＴ，ＳＬＯＣの３つの分類中で決定計数が最大のものを示している．この結果，ＩＴとＰＴではＩＴの決定計数が総合的に大きいことが分る．そして，層別しない全体ではＩＴの決定係数はＳＬＯＣに及ばないものの，開発言語で層別した結果を個別に見るとＳＬＯＣよりも大きい決定係数を持つものが多いことが分かる．図１に特に決定係数が大きいＶＢ，ＶＣ＋＋を利用したプロジェクトの開発工数と結合試験数の散布図を示す．

また，表２において，各開発言語によって回帰式が異なっている．そして，ＣＯＢＯＬやＣ＋＋ではグラフの傾きが大きい特徴がある．

ところで，表２中にはＵＴのデータがない．この

理由は，分析対象データ数を確保するためである．プロジェクトが残したデータは後工程ほど増える傾向があり，ＵＴのデータは明らかに他に比較して少なかった．このため，ＵＴのデータを含めると分析対象データが極端に減少するので，分析対象から割愛した．

## 4. 考察

調査結果より，開発言語によってデータを層別することにより試験項目数と開発工数との間に強い正の相関関係が認められた．そして，決定係数の比較評価結果は，試験項目数が開発規模と同等かそれ以上の開発工数との相関関係を持つことを示している．

さらに，特にＧＵＩ系の開発言語（ＶＢ，ＶＣ＋＋等）において，試験項目数の方が開発規模より開発工数との相関が強いので，試験項目数は開発規模の問題を改善できると考えられる．これらのことから判断して，試験項目数は，開発規模に代わる見積り根拠として十分機能することが期待できる．

また，開発言語によって回帰式の傾きが異なる現象は，選択する言語により試験項目あたりの開発工数が異なることを示している．つまり，開発言語毎に生産性が異なっていることを示している．この理由は，開発言語がシステムの要求される品質やアーキテクチャによって選択されるので，それに呼応して開発体制や期間，開発プロセスに差が出ることが原因と考えられる．主に基幹系システムの開発に用いられるＣＯＢＯＬや交換機の開発に用いられているＣ＋＋の傾きが大きいことはこの裏付けと考えられる．

一方，Ｃ言語では極端に決定係数が小さくなっている．この理由は，Ｃ言語を利用するプロジェクトの生産性にバラつきが大きいためである．つまり，Ｃ言語が広く普及し特定の領域での使用に留まらず特性の異なる広い分野のプロジェクトで利用されていることが原因と考えられる．この現象はＣＯＢＯＬにも当て嵌まり，開発言語が汎用に使われている状況を写している．この様な問題を避けるには別のプロジェクト特性でデータを層別することが必要と考えられる．本件は今後の課題としたい．

## 5. 結論

試験項目数は，計測が簡単で自動計測が可能であることからデータを残すための障壁が少ない．また，既存プロジェクトの残したデータ数が多く存在し見積り精度を向上するための分析対象が豊富である．これらの特徴を見積りに活かすために調査を行った結果，以下のことが判明した．

- 結合試験項目数と総開発工数の分析結果は深い相関を示しており，分析結果の回帰式は，試験から開発工数を見積る場合の見積りモデルとして期待できる．
- ソースコードを書かないＧＵＩによる開発の場合にもＳＬＯＣに比較して試験項目の方が良い見積りモデルを構築できる可能性が高い．
- プロジェクトを類型化して分析した結果，類型毎に生産性に違いがあることが判明し，適切な類型化によってさらに正確な見積りモデルを構築できる可能性が高い．

以上より，試験項目数による開発工数見積りは，現在主流となっているＳＬＯＣに替わる手段として有望であることが分かった．そして，さらに正確な見積りモデルを構築するための課題を抽出することができた．

## 6. 今後の展望

試験項目数は，良いプロジェクト規模の見積り根拠となり得ることが分かった．しかしながら，試験項目数の見積りができなければ，この利点を活かすことはできない．そこで現在，我々はユースケースに注目して研究を進めている．

ユースケースは，システムが果たすべき機能をユーザレベルで記述したドキュメントで，システムの外部仕様として要件定義に利用される．そして，ＲＵＰ®[5]をはじめとするオブジェクト指向開発手法において開発の極初期段階で着手するアクティビティーとして定着している．しかも，ユースケースと試験項目を表すテストケースは言葉としても似ているが，システムが果たすべき責務が記述されている点で同義と考えられる．実際，ＲＵＰ®ではユースケースから試験項目を抽出するアイデアが紹介されている．

現在，ユースケースの持つ良さを活かしつつ，試験項目による工数見積りと組み合わせることにより課題解消を模索中である．

## 文　献


[1] 児玉公信, 実践ファンクションポイント法, (株) 日本能率協会マネージメントセンター, 1999.7.15

[2] 平鍋健児, オブジェクトハンドブック 2002, (株) ピアソン･エデュケーション, 2001.10.10

[3] Capers Jones, 鶴保征城･監訳, ソフトウェア開発の定量化手法 第 2 版, 共立出版株式会社, 1998.4.20

[4] SLCP-JCF98 委員会編, 共通フレーム 98 ―SLCP-JCF98―(1998 版)ソフトウェアを中心としたシステム開発および取引のための共通フレーム国際規格適合, 通産資料調査会, 1998.10.28

[5] Philippe Kruchten, ラショナル統一プロセス入門第 2 版, (株) ピアソン･エデュケーション, 2001.3.25