

## 組込みソフトウェアの要求分析工程における品質予測手法

The method of predicting the quality of embedded software in requirements analysis process

主査： 飯泉 紀子 （株）日立ハイテクノロジーズ  
副主査： 田所 孝文 （株）山武  
研究員： 宮本 恒雄 （株）東芝  
          松平 敏明 アルパイン（株）  
          田中 聡 矢崎計器（株）  
          野吹 真人 日新電気（株）

### 1. 概要（日本語）

組込みソフトウェアの適用範囲はますます拡がり、その機能量も増大して、複雑化の一途をたどる中、そのQCD（品質、コスト、納期）に対する要求は、一層厳しさを増している。

そのような状況に対応するため、開発の上流工程で品質を作りこむことが求められている。

しかし、実際には、上流工程において成果物の品質を適切に把握することがうまくできていないため、下流工程で検出される不具合が予想以上に多く発生し、混乱を招いてしまう事態を繰り返している。

私たちのグループでは、上流工程で品質を予測できるならば、下流工程での混乱を低減できると考え、要求仕様書の規模やレビュー指摘件数を基に、品質を予測する手法を検討し、検証を実施した。

### 2. 概要（英語）

While the scope of embedded software spreads increasingly, the amount of functions also increases and complication is enhanced, the demand to the QCD (quality, cost, time for delivery) is increasing severity further.

To correspond to such a situation, it is requested to make the high quality in upper process.

However, actually, because the quality of the work products cannot be appropriately understood in upper process, the fault detected in lower process increased and the situation which causes confusion has been repeated.

In our group, in order to cancel such confusion, it verified by examining the technique of becoming possible predicting quality in upper process based on the scale of requirement specifications and number of fault detected by review.

### 3. 背景（研究課題を選定した理由）

私たちのグループのメンバは、品質保証担当であったり、プロセス改善担当であったり、その業務内容は異なっているが、自分たちが関わっている組込みソフトウェア製品の品質を少しでも良くしたいと願っていることは共通している。

組込みソフトウェアの適用範囲は、特定のコンピュータ関連の製品に留まらず、携帯電話、家電、自動車などへとますます拡がり、各々の製品において、新しい製品が出る度に、様々な機能が追加され、開発規模は増大の一途を辿っている。また、その製品のユーザ層が拡大すればするほど、そのQCD（品質、コスト、納期）に対する要求は、一層厳しさを増している。

技術者のリソース不足や育成不足による品質課題の増大などの影響で、特に下流工程で検出される不具合対策に多くのリソースを投入せざるを得ない状況となっており、その対策として、より上流工程での品質の作りこみを実施する必要があると考えられている。

ところが、品質保証活動やプロセス改善活動が進められていたとしても、上流工程で成果物の品質を適切に把握することがうまくできていない状況が多く見受けられる。そのため、下流工程のテストで思いもよらず多くの不具合が発生してしまい、不具合に対処するための手戻り工数が爆発的に増えて、コストや納期を守れなくなるといった悪循環を起こし、混乱状態に陥ってしまっている。

具体的には、品質保証担当者等による第三者を含めたレビューによって、決められたマイルストーン毎にその時点での作業成果物に対する達成度合いのチェックをしてはいるものの、客観的なデータに基づく品質状況の正確な把握ができていないために、後工程での品質状況を予測することもできずにいることが多い。たとえ、過去の経験などからなんとなく今後の開発が大変になりそうだということには感じていても、開発のその段階では組織的に適切な予防策を打つことができないでいる。

そして、下流工程になって、はじめて収拾がつかない状況となってしまうため、対策も後手に回ってしまっている。対策が後になればなるほど、解決が難しく、多くの工数を費やす結果となる。

そこで、開発の初期段階のうちに問題の発生を推測し、対策が打てるような取組みが必要と考え、上流工程で得られる客観的なデータに基づいて下流工程の品質状況を予測することを研究課題として選定した。

### 4. 目標

ソフトウェア開発の上流工程である要求分析工程において、作成されたソフトウェア要求仕様書の規模とその仕様書に対するレビュー指摘件数を基に、下流工程であるソフトウェア総合テストでの不具合件数を予測する手法を研究する。

### 5. 活動内容

#### 5.1. 品質予測手順の検討

上流工程での品質を客観的に見えるようにするために、最初に着目したのは、各工程で

実施されるレビュー活動で、それに焦点を絞り品質を把握するための指標の抽出を検討した。

レビュー活動の品質指標として、規模当たりの指摘件数や実施回数など予測に利用できそうなものを抽出し、それらを過去プロジェクトから収集し、各自仮説を立てた後、その仮説に基づき、仮説の検討を実施することとした。その後、データ再収集と仮説修正によって品質予測モデルを作成しようとした。

ところが、自分たちの関わっている部門の実際の状況を鑑みると、レビュー活動に着目した各種データの収集は困難であることが、懸念事項として提示された。

そこで、各工程での品質を把握するのではなく、文献[1]を参考にし、要求分析工程のソフトウェア要求定義アクティビティにおいて出力される作業成果物の一つである「ソフトウェア要求仕様書」に着目し、ソフトウェア総合テストでの不具合件数の予測につながるデータが得られないか検討することから始めた。その際、より多くの組織で予測手法が利用できるように、容易に収集可能なデータを利用することを条件として進めることとした。

検討を重ねた結果、一般的に開発の規模が大きくなるにつれ不具合数が増大していることから、要求仕様の量を一つのデータとして利用することにした。

また、要求仕様の質が悪ければ以降の設計の質低下を招き、最終的に不具合増加につながると考え、要求仕様の質に関するデータも利用することとした。要求仕様の量と質を何のデータから判断するかについては、多くの組織で収集可能な条件を考慮し、ソフトウェア要求仕様書のページ数とそのレビュー時の指摘件数とした。

しかし、要件開発の成熟度があまり高くない組織においては、ソフトウェア要求仕様書のページ数と実際の開発規模との関連性には揺れが想定され、開発規模の予測精度を高めるために、実際の開発規模を示すソースコードの量も利用することとした。

決定した3つのデータから不具合件数を予測する手法として、重回帰分析を選定した(文献[2])。理由は収集するデータが複数あるため、それらの相関を示すツールとして適切であり、また、相関があると判断できれば、予測式も簡単に導き出すことができるためである。

予測式は、組織が過去に開発したプロジェクトのデータを利用して算出することとする。終了したプロジェクトが少ないうちは収集できるデータの数も限られているため、予測式の精度は低くなるが、組織として継続してデータを収集していくことにより、年々精度が高まり、より信頼度の高い予測が可能になると期待できる。

上述した品質予測手順を表したものが図1である。尚、図1のV字モデルは文献[1]から引用した。

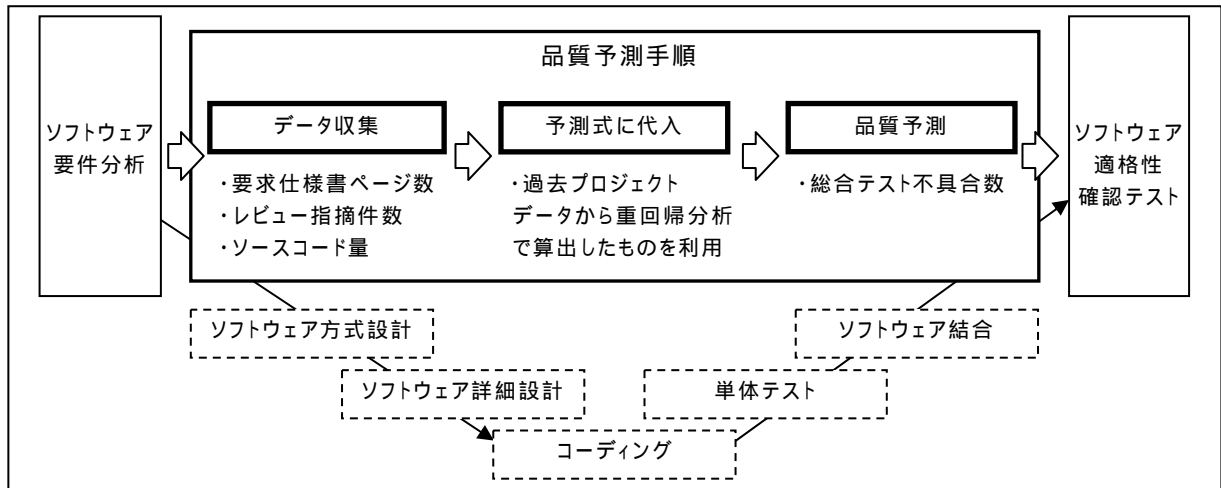


図 1. 要件分析工程での品質予測手順

## 5.2. 実施

要求分析工程で得られた 3 つのデータから品質の予測が可能か判断するために、私たちのグループメンバーそれぞれの会社で開発が既に終了しているプロジェクトを選定し、3 つのデータを収集すると共に、品質を示すデータであるソフトウェア総合テストにおける不具合件数も合わせて収集した。収集したデータを表 1 に示す。

- プロジェクト：3 会社から抽出された 18 のプロジェクトに対し、1 から 18 までの番号を付与したもの。
- 要求仕様書ページ数：それぞれのプロジェクトで作成されたソフトウェア要求仕様書のページ数。仕様書が複数ある場合にはその合計ページ数。
- 要求仕様書レビュー指摘件数：それぞれのプロジェクトでソフトウェア要求仕様書に対するレビューを実施した際に指摘を受けた件数の合計。指摘の中には仕様書の修正につながらないものも含む。
- ソースコード量 (LOC)：コメントを含む総ステップ数。
- 総合テスト不具合件数：ソフトウェアを構成する機能ユニットをすべて結合した状態で実施されるソフトウェア総合テスト以降に検出された不具合件数の合計。本件数の中には対策をしなかった不具合も含まれる。

要求分析工程の成熟度があまり高くないためか、一部のプロジェクトではレビュー指摘件数データが取得できなかった。収集できなかった部分は表中 ” - “ で示す。

表 1. 収集したデータ

プロジェクト	要求仕様書 ページ数	要求仕様書 レビュー指摘件数	ソースコード量 (LOC)	総合テスト 不具合件数
1	1047	-	589142	1550
2	5236	-	2758372	2148
3	1257	-	271660	2826
4	1134	-	1462242	1435
5	566	-	976398	341
6	626	-	399326	943
7	162	-	34924	215
8	35	2	8426	21
9	151	17	221456	124
10	51	-	892438	947
11	12	-	850000	279
12	8	-	7007	2
13	2	4	100	1
14	54	95	14900	163
15	40	12	12000	90
16	35	38	96100	232
17	35	7	13000	305
18	49	50	12892	35

続いて収集したデータを用いて重回帰分析を実施した。

まず、総合テストの不具合件数とそのレビュー指摘件数の間には、相関があると期待できるので、それを表した結果が図 2 である。ところが、サンプル数が少ないためか、そのグラフからは期待される相関関係は見出せず、レビュー指摘件数データが使いそうにないことが分かった。

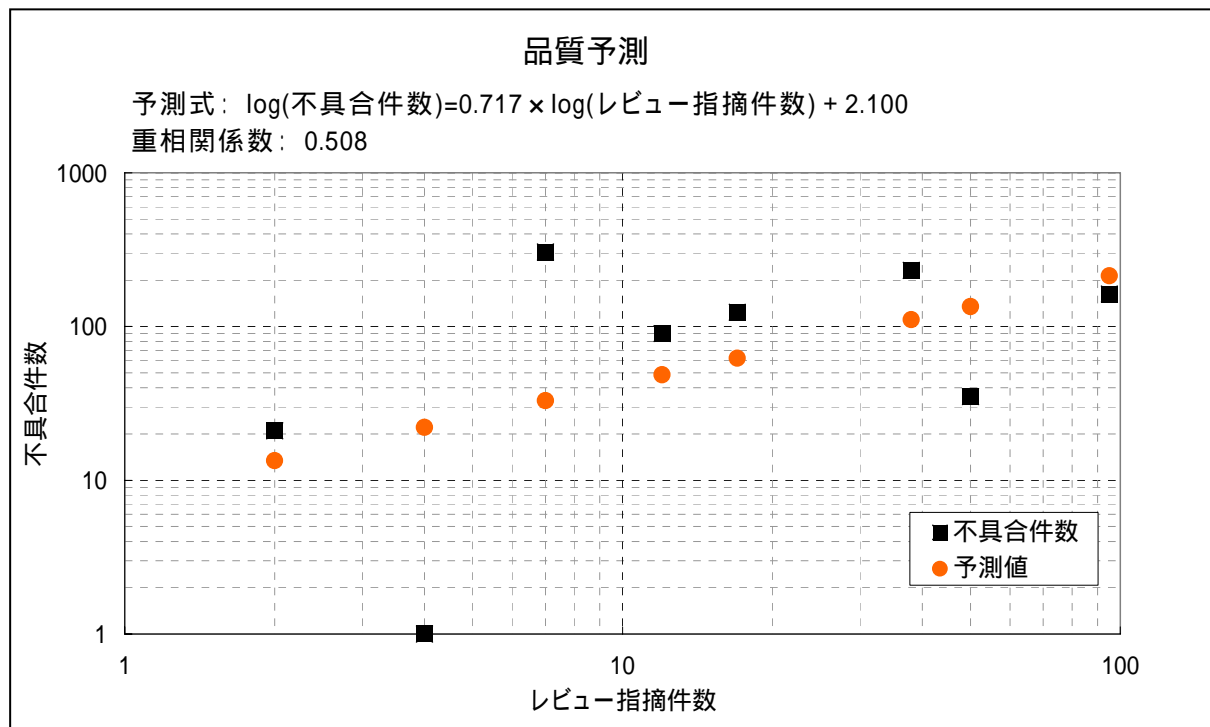


図 2. 要求仕様書ページ数とレビュー指摘件数の相関

総合テストの不具合件数とレビュー指摘件数には相関があると想定されるのにもかかわらず相関が見られなかったのは、レビュー以外による品質確保がなされているか、またはレビュープロセスに問題があるためと考えられる。

そこでレビュー指摘件数を除外し、要求仕様書のページ数とソースコード量の2つのデータから品質予測が可能か、重回帰分析を行った結果が図3である。

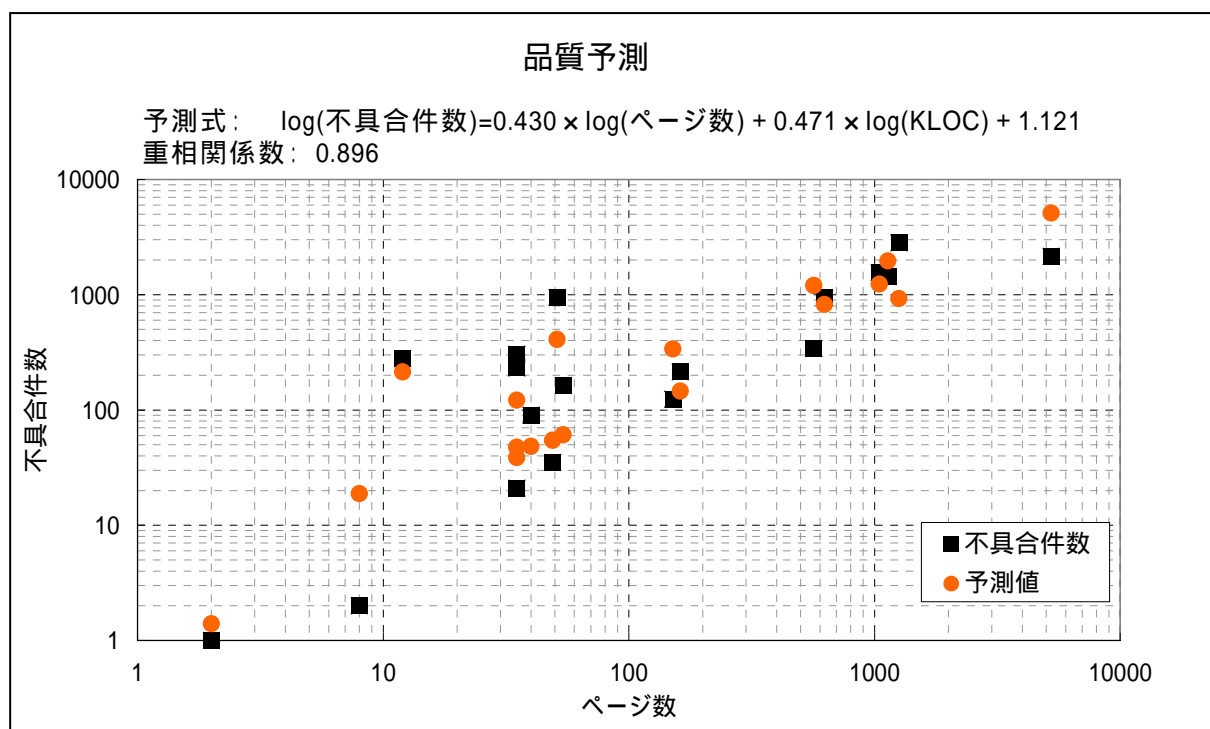


図3. 要求分析工程における品質予測（2つのデータによる品質予測）

2つのデータによる予測値であっても十分、実績値との相関が見て取れる。要求仕様書のページ数が増えると総合テストでの不具合件数が増える傾向にあり、ソフトウェアの開発規模の増大に伴い、ソフトウェアの不具合数も増加するということが、経験則ではなく客観的なデータに基づき説明できるということが分かった。

### 5.3. 結果

本研究の品質予測手法を用いることにより、ソフトウェア要求仕様書のページ数とソースコード量の2つのデータを利用して総合テストでの不具合件数を予測することが可能であることが分かった。

### 5.4. 考察

本研究で実施した品質予測手法では、計測が容易な僅かなデータを利用するだけでも品質を予測できることを示すことができた。さらに、品質予測によって得られる効果には次のようなことが考えられ、期待が持てる。

- 上流工程で品質が予測できることから、開発の早い段階で下流工程に見合ったリソースを確保したり、計画の見直しをするなどの対策が打て、開発終盤の混乱を抑えることができる。
- ソフトウェア総合テスト段階で検出された不具合件数と予測した件数を比較することによって、テストの質を判断する材料とすることができる。そして、状況分析することで成果物の品質低下がないか、テスト不足になっていないか、そのために対策を打つ必要があるかどうかの判断に利用できる。
- 開発終了時の総不具合件数と予測値を比較することによって、今回の開発が過去の組織開発力と比べてどうだったか判断する材料となるため、組織で進めているプロセス改善活動などの効果を客観的なデータとして示すことができる。
- 品質保証担当者のような第三者的立場であっても、裏付けられたデータによる説明が可能となるため、プロジェクトメンバーや上級管理層により分かりやすく、納得性のある報告ができる。

## 6. 結論（成果）

ソフトウェア要求仕様書のページ数とソースコード量というわずかなパラメータでソフトウェア総合テストの不具合件数を予測可能であることを、我々のデータを利用して検証できた。

## 7. 今後の展開

本論文では、要求分析工程での品質予測であったが、同様の手段により各工程での品質予測ができると考える。

前述したように、総合テストで発見された不具合が作りこまれた工程を識別することで、各工程でのデータを補足して、工程ごとの品質予測につなげることも可能と考える。

また、要求定義段階の成果物の規模から、次工程である設計段階の成果物の品質を予測したり、さらに設計段階の状況から、次工程のソースコードの品質を予測したり、次々と後工程の品質状況を予測することも可能となってくる。

様々な段階で品質を予測することが可能になれば、開発の混乱をより低減でき、より高品質な開発ができるようになると期待できる。

## 8. 参考文献

- [1] IPA/SEC：「組込みソフトウェア向け 開発プロセスガイド」、株式会社翔泳社、2006
- [2] 脇本和昌、田中豊：「多変量統計解析法」、現代数学社、1983