

6 要求仕様の品質特性

大西 淳^{*1}^{*1} 立命館大学佐伯 元司^{*2}^{*2} 東京工業大学

要求定義は、ソフトウェア開発プロセスの中で最も前段に位置するため、そこでの作業や成果物の質が、開発効率や最終成果物の質に大きな影響を及ぼす。本稿では要求定義プロセスの成果物である要求仕様の品質特性に着目し、要求仕様の品質を管理したり評価するための、品質特性の数量化について述べる。

要求仕様そのものの品質については、IEEE Std 830-1998⁴⁾において、以下に示す8つの品質特性が示されているが計測手段や計量モデルは示されていない。

1. 妥当性 (correctness, 正当性ともいう)
2. 非あいまい性 (unambiguity)
3. 完全性 (completeness)
4. 無矛盾性 (consistency)
5. 重要度と安定性のランク付け (ranked for importance and/or stability)
6. 検証可能性 (verifiability)
7. 変更可能性 (modifiability)
8. 追跡可能性 (traceability)

以下では、それぞれの品質特性の意味を紹介し、その計測方法について、実際に具体的な計測のための手段を挙げて述べる。なお、ここで与える計測可能な手段(数式)は一例にすぎない。1つの品質特性に対して、いくつかの異なった計測手段が存在する場合もある。そのような場合には複数の異なる計測手段による数量化の結果を用いて要求仕様の品質特性の達成している度合いを総合的に判断すればよい。

要求仕様には、システム全体の要求仕様書(システム要求仕様書)、システム中でソフトウェアで実現する部分の要求仕様書(ソフトウェア要求仕様書)があり、ここではソフトウェア要求仕様書の品質の測定のみ限定するが、妥当性や追跡可能性など、他の該当文書を含めないと測定できないような品質もある。また、IEEE Std 830-1998に準拠しているソフトウェア要求仕様書

の3節(要求事項についての節)のみを対象とし、この節は箇条書き形式の自然言語で記述されているとする。

妥当性

要求仕様書が妥当とは、要求仕様書中に述べられているすべての要求は、開発されるソフトウェアが満たすべきものであることを意味している。したがって、ソフトウェアが満たすべき以外の事項が書かれていれば、その仕様書の妥当性は低いことになる。妥当性の数量化は、

$$\text{妥当性} = \frac{\text{要求仕様書中でソフトウェアが真に満たすべき要求文の数}}{\text{要求文の総数}}$$

となる。

しかしながら、要求文がソフトウェアが真に満たすべき文であるかどうかを、ツールや手続きによって検証することは難しく、プロトタイプなどを用いて顧客や利用者自身によって確認してもらうことになるのが普通である。したがって、なんらかの近似に基づく数量化が必要になる。

ソフトウェアの要求仕様書以外に、システム要求仕様書など、より上位の仕様書が作成される場合がある。その場合これらの文書の記載項目との比較を行い、ソフトウェア要求仕様書中の要求が、上位の仕様書のどの記述に意味的に対応づくかにより、妥当性を数量化することができる。

$$\text{妥当性} = \frac{\text{上位の文書の記載と意味的に対応づく要求文の数}}{\text{要求文の総数}}$$

このような上位の仕様書のほかにも標準規約や法令などの文書との比較も考えられる。この近似では、これらの上位の文書の記載が、正しくソフトウェアが真に満たすべき要求のみであるという仮定に基づいている。

非あいまい性

要求仕様中に述べられているすべての要求が一意に解釈できる場合、要求仕様はあいまいでないといえる。もしある要求が何通りにも解釈できる場合は、その要求仕様はあいまいとなる。

あいまい性を生じさせる要因として、自然言語自身が持っている要因と、文の読み手の背景知識が異なるため読み手によって異なる複数の解釈を生じてしまう要因とがある。

要求仕様書の非あいまい性は、仕様書中のあいまいな文の出現頻度を数えることによって数量化でき、以下の式で表すことができる。

$$\text{非あいまい性} = 1 - \frac{\text{あいまいな要求文の数}}{\text{要求文の総数}}$$

どの文があいまいであるかどうかを、誰が判定しても同じという意味で客観的に、かつ正確に判定することは難しい。それは、文を解釈する人間の側に解釈に使用する知識の差があるからであり、ある人にとってはあいまいでなくても別の人にとっては複数の解釈が生じてくることもある。したがって、現実的にはあいまいな文となる可能性の高い文を判別する手法を考えるのがよい。

自然言語自身が持っているあいまい性は、自然言語の文法からくるものと、語句自身の客観的な意味が不明確であることに起因するものがあり、構文的には以下のような文があいまいな要求文となっている可能性が高いと考えられる。

1. 語句の係り受け関係が複数あるような文、つまりコンピュータで構文解析を行ったときに、構文解析木が複数得られるような文。
2. 主語や目的語などが省略されている文。
3. 指示語を含む文。
4. 「使いやすく」、「見栄えよく」など、それを達成したかどうかを判断する際の判断基準が、読み手によって変わるような語句を含む文。このような語句には客観的な意味の定義がなく、特に非機能要求や品質要求の記述に含まれることが多い。
5. 範囲や境界を表す語が含まれる文。

これらに該当する文を数え上げ、文全体に対して占める割合によって、あいまいになる可能性を数値化することができる。もちろん、これらに該当しているからといってその文が本当にあいまいであるとは限らない。

自然言語の持つあいまいさ以外に、文の読み手の背景知識が要因となり、あいまいな意味解釈をもたらしてし

まうことがある。たとえば、顧客やユーザは問題領域のエキスパートであるが、要求分析者はそうではない。逆に要求分析者はコンピュータの専門家であるが、顧客・ユーザはそうではない。エキスパートが読むとあいまいではないが、その領域をよく知らないステークホルダが読むと、正確な意味が分からず、複数の解釈を行うことがある。

完全性

要求仕様書が完全であるとは、

- 機能、性能、設計制約、属性、外部インタフェースに関する要求はすべて記載されている。特に、システム要求で触れられている外部のシステムに対する要求はすべて記載されていなければならない。
- すべての状況において、可能な入力データすべてに対してソフトウェアがどう応答するかが記載されている。特に正当な入力値と不当な入力値の両方に対する応答が記載されていなければならない。
- 要求仕様中の図や表に対するラベルと参照、および要求仕様中の用語の定義と単位の定義が記載されている。なお、定義がないためにその解釈をめぐってあいまい性が生じることもよくある。

これらの項目の抜けの割合を計算することにより、要求仕様書の完全性は以下のように数量化できる。

$$\text{完全性} = \frac{\text{抜けがある文の数}}{\text{要求文の総数}}$$

本来は、完全性とは顧客やユーザの真のニーズが漏れなく仕様書に書かれているかどうかを示す指標と考えたほうが自然であるが、顧客やユーザ自身も自分の本当のニーズが何か分かっていないことが多いため、直接測定することは難しい。上記のような項目が顧客やユーザのニーズをすべて表現しているととらえ、これらの記載が抜けがなく要求仕様書にあるかどうかで、間接的に測定していると考えてもよいであろう。完全性をこのようにとらえると、Davis³⁾も述べているように、妥当性と完全性との関係を、集合の包含関係で書くと図-1のようになる。図中で集合AとBが同じ集合となる、つまりC = A = Bとなれば、その要求仕様書は妥当かつ完全である。A - Cの部分が、顧客やユーザの真のニーズでありながら、仕様書には記載されていない部分、いわゆる「抜け」になる。B - Cが、顧客やユーザのニーズではない、つまりソフトウェアが必ずしも満たさなくてもよい要求でありながら要求仕様書に記載されている部

分になる。この部分が大きければ妥当性は低くなる。

要求仕様書に、まだ決まっていなあるいは明確になっていない事項を表すために、TBD (To Be Determined) という語句を使用することがある。これは、その要求項目の存在には分析者は気づいているという点で「抜け」よりは好ましいが、完全ではない。

無矛盾性

無矛盾性は要求仕様の中で一貫していること、つまり個々の要求が互いに矛盾しないことを表している。仕様書に含まれる矛盾としては以下のようなものが考えられる。

1. ソフトウェアの動作が矛盾
 同じ入力に対してのソフトウェアの振舞いや出力が、複数の個所に記述されており、しかもそれらが異なっている。
2. 定義が矛盾
 語句の定義やその説明が複数個所でなされているが、その内容が食い違っている。
3. 制約が矛盾
 制約が複数書かれているが、それらを満足する状況が存在しない

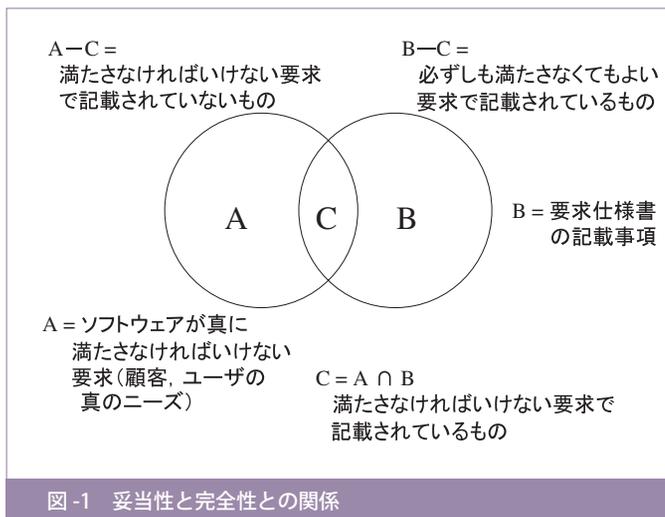
矛盾性は、以下のように数量化できる。

$$\text{無矛盾性} = 1 - \frac{\text{矛盾にかかわった要求文の数}}{\text{要求文の総数}}$$

重要度と安定性のランク付け

個々の要求に重要度や安定性を示す識別子がある場合に、要求仕様書は重要度と安定性のランク付けがされていることを意味する。一般に要求すべてが同一の重要度を持つわけではなく、重要度のランク付けは「必須な要求」か「条件つき要求」か「あってもなくてもよい要求」といったようにランクが付けられていることを意味する。ISO/IEC 12207-1995 Software Lifecycle Process, JIS X0160-1995 ソフトウェアライフサイクルによれば、

1. 必須な要求は次のような表現で終わる（英語では shall, will が用いられる）。
 「する」、「とする」、「による」、「すること」、「(の)こと」、「(し)なければならない」、「とおりとする」
2. 推奨を表現する要求は次のような表現で終わる（英語では should が用いられる）。
 「することが望ましい」、「ほうがよい」、「するのが



よい」、「するとよい」

3. 限度内で許されることを表現する要求は次のような表現で終わる（英語では may が用いられる）。
 「(し)てもよい」、「差し支えない」

要求文をこれらの3種類のいずれかに分類した場合に、それぞれに分類された要求が上記の表現をとっているか否かで重要度のランク付けの数量化が可能となる。以下のような数量化が考えられる。

$$\text{重要度のランク付け度} = \frac{\text{重要度のランク付けが正しく表現された要求文数}}{\text{重要度のランク付けが必要な全要求文数}}$$

また、安定性は、たとえば以下のように評価ができる。

$$\text{安定性のランク付け度} = \frac{\text{安定性が正しく表現された要求文数}}{\text{変更される可能性のある全要求文数}}$$

といった式が考えられる。

検証可能性

要求仕様書が検証可能とは「ソフトウェア製品がその要求を満たしていることを計算機や人手によって、有限の費用でチェックできるプロセスが存在すること」を意味する。「うまく」や「しばしば」といった定性的な表現があると検証ができないので、

$$\text{検証可能性} = 1 - \frac{\text{定性的な表現個所の数}}{\text{定量的に表現すべき個所の数}}$$

といった式で検証可能性を表すことができる。

また、あいまいな要求は検証可能性が低いと見なすことができる³⁾ので、非あいまい性でもって検証可能性を表すことができる。検証するのが困難な要求、たと

例えば「放射能漏れが発生した場合に、本システムは半径20km以内の人間の致死率を80%以内に抑えること」といった要求はテストできない³⁾。Davisらは以下の式を検証可能性として与えている。この式は単位が与えられていないのであいまいだが、検証のコストや時間がかかるほど0に近づく。

$$\text{検証可能性} = \frac{\text{全要求文数}}{\text{全要求文数} + \text{要求検証のための総コスト} + \text{要求検証のための総時間}}$$

変更可能性

要求仕様書の構造やスタイルを保持したまま、任意の要求を容易に、完全に、矛盾なく変更できる場合、要求仕様書は変更可能という。変更可能であるためには

- 要求仕様書に、目次、索引、クロスリファレンスが付けられていること。
- 要求が冗長でないこと。つまり同じ要求が2カ所以上に表れないこと。
- 要求が互いに依存しないこと。
- 複雑な要求を1つの要求として表現せずに、個々の要求を分離して別々に表現すること。

が望ましい。冗長な要求文の数や互いに依存する要求文の数を使って数量化すると、それぞれ以下ようになる。

$$\text{変更可能性} = 1 - \frac{\text{冗長な要求文の数}}{\text{全要求文数}}$$

$$\text{変更可能性} = 1 - \frac{\text{依存する要求文の数}}{\text{全要求文数}}$$

同様に個々の要求を1つの要求として表現しているかどうかの度合いを用いて以下のように数量化できる。

$$\text{変更可能性} = 1 - \frac{\text{複数の要求を1つの要求文とした文数}}{\text{全要求文数}}$$

Davisらは目次と索引が要求仕様書に付けられていることが変更可能性に大きく影響を与えたと考え、目次と索引が用意されている場合を1、そうでない場合を0とする指標を示している³⁾。

追跡可能性

追跡可能性は次の2種類に分けられる。

1. 後方追跡可能性：各要求から、要求仕様書に先立って作成された文書中の各要求の起源について書かれた個所を参照できること。

2. 前方追跡可能性：各要求が名前と参照番号を有し、要求仕様書を基にして作成されたすべての文書（たとえば設計仕様書やソースコード）から参照できること。

したがって、要求仕様書単独では定義できないが、要求文ごとに追跡可能かどうかを数えることによって数量化することができる。

$$\cdot \text{後方追跡可能性} = \frac{\text{要求の起源を参照可能な要求文数}}{\text{全要求文数}}$$

$$\cdot \text{前方追跡可能性} = \frac{\text{名前と参照番号を持った要求文数}}{\text{全要求文数}}$$

$$\cdot \text{前方追跡可能性} = \frac{\text{要求仕様書を基に作られた文書から参照可能な要求文数}}{\text{全要求文数}}$$

前方追跡可能性を計測するための数式を2つ示しているが、品質特性を数量化する手段を複数用いることによって、達成する度合いを総合的に判断できる。

Davisらは(1)章節番号を付与していること、(2)段落ごとに1要求のみが書かれていること、(3)各要求にユニークな番号が振られていることを、追跡可能性を向上させる要因として、これらの達成度を追跡可能性の指標としている³⁾。

他の品質特性

要求仕様は変更が行われるのが常であり、製品のリリース後だけではなく、開発中にも頻繁に変更が行われる。それゆえ、IEEE std 830-1998で述べられている品質評価項目以外に、安定性(Stability)や変動性(Volatility)の評価項目とその評価を行うためのメトリクスが提案されており、実際にそれらを用いて計測した事例も報告されている。要求の変動性(Volatility)とは、ソフトウェアを開発中にその要求自身が変化することを表し、要求仕様書を完成させる前に要求が変化すると、要求仕様書が完成した後の段階で変化する場合とがある。

論文⁵⁾では、要求仕様としてユースケース記述をとりあげ、ユースケース記述の変更量を変動性の尺度と定義した。具体的には、変更がなされた際に作られるリビジョンごとの変更の回数を記述サイズ(ユースケース記述の語数)で割ったものを求め、その総和を変動性とした。変更予測において、このような尺度が専門家の判断よりも優れていることを実際の開発プロジェクトで確かめている。

論文¹⁾では、ゴール指向分析法で得られたゴールグラフと、対象領域の将来像のシナリオ記述とを比較し、各

ゴールがどれぐらいシナリオに対して安定かを採点する。具体的には、将来を記述したシナリオとゴール記述とを比べ、シナリオを達成するためには、新規ゴールの追加、ゴールの削除、ゴールに記述されている機能の変更などの操作が必要かどうかを考え、将来もそのゴールが安定かどうかを1～4段階で評価する。どのような操作が必要かの情報は、そのゴールの子孫や親の安定度を評価する際にも使用する。つまり安定度の評価結果は、ゴールグラフの子孫や親に伝播し、全体の評価が行われる。将来像を記述したシナリオをどのようにして発見し、記述するかは残念ながらこの論文では述べられていない。

論文⁶⁾では、要求変更の起こる理由を、1) ビジネス環境が変化する可能性がある、2) ユーザの要求が変動する、3) ユーザやステークホルダ間で同意がとれていない、の3つに分け、プロジェクト参加者に自分のプロジェクトで上記3つの理由に該当することがあったかどうかを5段階で評価してもらい、その統計値で要求の変動性を表した。変動性がプロジェクトのパフォーマンスにどのような影響を与えるかを、コストと開発スケジュールの観点から分析した。その結果、変動性は、コストやスケジュールの超過に大きな影響を及ぼしていることを確認した。さらに、ユーザと開発者が頻繁にコミュニケーションを行ったり、明確に定義された要求分析・モデル化の方法論を使用したりすることが要求の安定に大きな効果を与えることが分かったとしている。

論文³⁾では、要求仕様には多くのエラーが混入することを事例報告での数値を基に示した上で、要求仕様中のエラーの検出を目的として、要求仕様に対して24の品質特性を定義しており、さらに18の特性については計測のための数式を示している。これらの一部については本稿でも紹介している。

エラーの混入具合を調べることによって品質を測定する手法もある。R. Costelloらは、論文²⁾において要求工学のメトリクスとして、変動性を含めて、追跡可能性、完全性、欠点の密度(Defect Density)、欠陥の密度(Fault Density)などを挙げている。彼らの変動性は、実際に行われた要求の変更とその理由を取り上げている。欠点(Defect)とは要求仕様書のレビューやウォークスルー時に発見された誤り、欠陥(Fault)とは実行テスト時に発見された誤りであり、これらが要求仕様書のサイズに対し、どれぐらいの頻度で見つかったかで数値化している。誤りの発見段階でメトリクスを区別しているのは、ソフトウェア開発の後段で見つかった誤りほど重大であり、これらを多く含む要求仕様書はより低品質であるという主張なのであろう。

まとめ

本稿ではIEEE std 830-1998で規定されている要求仕様の品質特性を紹介し、品質特性を数量化するための計測手段の例を紹介した。数量化によって得られる数値をあらかじめ設定した基準値や標準値と比較することによって、その品質特性が十分な水準にまで達成できているかどうかを判断できるようになる。基準値や標準値は、開発対象のソフトウェアの特性によって異なり、組織や開発グループによって設定されるものであり、最初から値が決まっているものではない。

本稿では具体的な要求仕様に適用した結果を示すことはできなかったが、本学会ソフトウェア工学研究会要求工学ワーキンググループでは、具体的な要求仕様と品質特性の数量化の結果を公開しており⁷⁾、興味のある読者は参考にさせていただきたい。

謝辞 要求工学ワーキンググループのメンバ各位、特に貴重なコメントを数多くいただいた新日鉄ソリューションズ(株)の中村友昭氏に感謝する。本研究は一部科学研究費補助金基盤研究(C)(19500034)による。

参考文献

- 1) Bush, D. and Finkelstein, A. : Requirements Stability Assessment Using Scenarios, In Proc. of 11th IEEE International Requirements Engineering Conference (RE'03), pp.23-32 (2003).
- 2) Costello, R. J. and Liu, D.-B. : Metrics for Requirements Engineering, Journal of Systems and Software, Vol.29, No.1, pp.39-63 (1995).
- 3) Davis, A. et al. : Identifying and Measuring Quality in a Software Requirements Specification, In Proc. of 1st IEEE International Software Metrics Symposium, pp.141-152 (1993).
- 4) IEEE, Recommended Practice for Software Requirements Specifications, Std 830-1998 (1998).
- 5) Loconsole, A. and Börstler, J. : Are Size Measures Better Than Expert Judgment? An Industrial Case Study on Requirements Volatility, In Proc. of 14th Asia-Pacific Software Engineering Conference (APSEC2007), pp.238-245 (2007).
- 6) Zowghi, D. and Nurmaliani, N. : A Study for the Impact of Requirements Volatility on Software Project Performance, In Proc. of 9th Asia-Pacific Software Engineering Conference (APSEC2002), pp.3-11 (2002).
- 7) 要求工学ワーキンググループ Web ページ (2007) : <http://www.selab.is.ritsumei.ac.jp/~ohnishi/RE/rewg.html>

(平成 20 年 1 月 30 日受付)

大西 淳 (正会員) ohnishi@cs.ritsumei.ac.jp

1979 年京都大学工学部情報工学科卒業、1983 年同大学院工学研究科博士課程情報工学専攻退学、工学博士、京都大学助手、助教授を経て 1994 年から立命館大学教授。現在情報理工学部に所属。要求工学などの研究に従事。

佐伯 元司 (正会員) saeki@se.cs.titech.ac.jp

1978 年東京工業大学工学部電気電子工学科卒業、1983 年同大学院工学研究科博士課程修了、工学博士、東京工業大学助手、助教授を経て 2000 年から東京工業大学教授。現在情報理工学研究科に所属。要求工学などの研究に従事。