

要求仕様書におけるステークホルダ間の齟齬の検出方法の提案と評価

2018SE035 黒田 陸人

指導教員：佐伯 元司

1 研究背景および課題

1.1 背景

ソフトウェア開発の上流工程は後の工程や成果物の品質に影響を及ぼす。その上流工程では、ソフトウェアがどうあるべきかを記述した要求仕様書が作成される。この要求仕様書の質が、ソフトウェア開発における成果物の品質に大きく影響する[1]。要求仕様書の質が低い場合、手戻り、スケジュールの遅延、コストの増加などの問題を引き起こす可能性がある。そうした問題を未然に防ぐためには、要求仕様書の質を向上させる必要がある。しかし、要求仕様書は主に自然言語で記述されるため、その記述には曖昧性が含まれる場合がある。その曖昧性により、ステークホルダ間において、要求文に対する解釈の違いから齟齬が起きることがある[2]。他にも、ステークホルダ(受注者と発注者)間での問題領域の知識の差から生じる齟齬もある[5]。こうした齟齬をソフトウェア開発の上流工程において検出することが必要である。そのためには要求仕様書におけるステークホルダ間の齟齬を検出する手法が必要となる。

1.2 課題

齟齬を検出する研究について、参考文献[2]は要求仕様書内の機能要求を対象とし、要求記述の曖昧性から生じるステークホルダ間の解釈の齟齬を検出する手法の提案をしている。また、参考文献[5]では、ゴール指向要求分析法の AGORA (Attributed Goal-Oriented Requirements Analysis Method)を提案しており、その中で述べられている優先度行列はステークホルダ間の合意形成の度合いや誤解を求めることができるとしている。この2つの手法はステークホルダ間の解釈の齟齬や誤解を検出しているが、参考文献[2]では質問を作成する要求分析者に、参考文献[5]では順位を付けるステークホルダにとって労力がかかる。そこで本研究では要求文に対するステークホルダ間の解釈の違いによる齟齬を研究対象と定め、ステークホルダや要求分析者の労力を軽減した齟齬を検出する手法の提案と、提案手法を適用した実験を行い、評価と考察をする。

2 提案手法

本研究では、要求仕様書に記述される要求に対して、ステークホルダ間での齟齬を検出する手法を提案する。

具体的な提案手法は、各ステークホルダに要求文に対するそれぞれの解釈に基づいて、あらかじめ決められた分類項目へその要求文を分類させる。その分類結果から、要求文を分類した項目が異なれば、その要求文についてステークホルダ間で齟齬が起きている可能性があることを検出する。

本研究では、検出する齟齬の対象を要求仕様書内の非機能要求に定める。非機能要求についてステークホルダ間の齟齬を検出するまでの流れを述べる。

(1) 要求仕様書から要求を抽出

齟齬の検出を行う対象の要求仕様書から、要求文を抽出する。

(2) 分類項目を作成

要求分析者は、ステークホルダが各要求文を分類する際に用いる分類項目を作成する。要求分析者は、作成した分類項目の各項目の詳細について、ステークホルダへ説明を行う。

(3) ステークホルダによる分類

ステークホルダは抽出された要求文を、要求文に対する各自の解釈に従って、最も当てはまると思う分類項目へ分類する。

(4) 齟齬の検出

要求分析者は、ステークホルダによって分類された結果を比較し、分類結果が異なる要求文に対し、その要求文についてステークホルダ間で齟齬が起きている可能性があることを検出する。

3 実験

3.1 実験の目的

実験の目的は以下の RQ に答えるためである。

RQ1 提案手法の齟齬検出精度はどれほどか

RQ2 提案手法の実施にかかるステークホルダの労力はどれほどか

3.2 実験の概要

先述の RQ を受け、提案手法を適用し、被験者と実際に使用されている要求仕様書を用いて実験を行った。実験では、分類項目を 2 種類作成し、実験①、実験②として 2 回行った。

実験①では、5 種類の仕様書と 2 名の被験者を用いて実験を行った。分類項目は、IPA(独立行政法人情報処理推進機構, Information-technology Promotion Agency)が公開している非機能要求グレード[3]の大項目にユーザビリティを追加した 7 項目の分類項目を作成した。7 項目は、可用性、性能・拡張性、運用・保守性、移行性、セキュリティ、システム環境・エコロジー、ユーザビリティとなる。被験者は分類作業の労力を測るためのアンケートにも回答した。

実験②では 2 種類の仕様書と 3 名の被験者を用いて実験を行った。分類項目は、JIS X 25010(ISO/IEC 25010)[4]で提唱されている品質特性を分類項目として用いた。品質特性には、機能適合性、性能効率性、互換性、使用性、信頼性、セキュリティ、保守性、移植性の 8 つがあり、これを用いた。

3.3 実験結果の評価方法

評価では、要求文に対するステークホルダ(被験者)の解釈を実験者が聞き取ることで、提案手法による齟齬の検出精度を測る。精度を測る方法は適合率を用いる。計算方法は以下の式(1)となる。

$$P = \frac{T}{T+F} \quad (1)$$

ここで、各変数は以下のように定義する。

P 適合率(Precision)

T 検出した齟齬のうち、聞き取りから齟齬であると判断した数(True Positive)

F 検出した齟齬のうち、聞き取りから齟齬ではないと判断した数(False Positive)

また、ステークホルダの分類作業にかかる労力を測るために質問事項と、リッカート尺度を用いた回答によるアンケートを行う。各質問項目に対する回答を集計し、その平均値を算出して労力を測る。質問事項は以下の3つである。

1. 分類に用いた7項目への理解は難しくなかった
2. 分類作業にはさほど手間はかからなかった
3. 本実験で使用された資料、実験者からの説明はわかりやすかった

回答方法は、各質問において次の1から5のうち最もあてはまる選択肢を選んだ。

1. 全く当てはまらない
2. 当てはまらない
3. どちらともいえない
4. 当てはまる
5. 非常に当てはまる

3.4 実験結果と結論

実験①では、5つの仕様書において検出した齟齬の数は17個、そのうち聞き取りから齟齬であると判断した数が8個となり、合計の適合率が47.1%となった。この結果から、提案手法による齟齬の検出はできたが、その精度は高くない、という結論を得た。また、被験者の分類作業にかかる労力を測るアンケートでは、質問2の平均値が3.4となり、分類作業にかかる労力は必ずしも少ないとは言えない、という結論を得た。

実験②では、2つの仕様書において検出した齟齬の数は16個、そのうち聞き取りから齟齬であると判断した数が12個となり、合計の適合率が75.0%となった。この結果から、提案手法による齟齬の検出はでき、その検出精度は比較的高い、という結論を得た。

4 考察

4.1 適合率低下の原因

本研究の実験結果において、適合率を低くする原因は式(1)で示した、検出した齟齬のうち、聞き取りから齟齬ではないと判断した数、である。この聞き取りから齟齬ではないと判断した要求文に対する被験者の解釈をまとめると、その原因は3点に分類することができた。本実験における件数とともに各原因を以下に示す。

1. 要求文内での注目する部分の違い:7件

要求文に複数の非機能要求が含まれ、被験者が異なる非機能要求に注目して分類をした。

2. 分類の間違い:4件
被験者が分類を間違えた。
3. 要求文を先まで予想:2件
被験者が保守対応後や設置先の内容を予想して異なる分類をした。

4.2 外的妥当性

本実験では要求仕様書をインターネット上で入手可能な実際に使われているものを無作為に収集した。この点において外的妥当性への脅威は少ないと考える。一方、4.1節で述べたような原因は、必ずしも他の要求仕様書や被験者による実験を行った際に当てはまるとは言えない点において、外的妥当性への脅威がある。

4.3 内的妥当性

実験における被験者と実験者は全員、理工学部に所属する4年生であり、ソフトウェア工学、要求工学について同程度の知識を持つ人間であることから、作業を行った人間の知識の差が結果に影響を及ぼしたとは言えない。この点において内的妥当性への脅威は少ないと考える。しかし実験における、要求仕様書から非機能要求を抽出する作業、被験者が分類を行う基準、実験者の聞き取り方法については、その作業を行う人間が異なると、同等の結果を得ることができない確証がない。この点において内的妥当性への脅威があると考えられる。

5 まとめ

提案手法によって要求仕様書内の非機能要求について、ステークホルダ間の齟齬を検出できることが分かった。

本研究の今後の課題として、検出できなかった齟齬の検出、非機能要求以外の記述への適用、妥当性への脅威の軽減を挙げる。

6 参考文献

- [1] Aybüke Aurum, Claes Wohlin, Engineering and Managing Software Requirements, Springer, 2005, pp.163-185
- [2] 井上 渉, 鶴飼 孝典, 林 晋平, 佐伯 元司, 質問形式を用いた機能要求記述の解釈の齟齬の検出支援, 情報処理学会第73回全国大会, 2011
- [3] 独立行政法人情報処理推進機構(IPA),非機能要求グレード, 2010-2018, 入手先<<https://www.ipa.go.jp/sec/softwareengineering/std/ent03-b.html>>, 2021年8月閲覧
- [4] 日本規格協会, JIS X 25010, システム及びソフトウェア製品の品質要求及び評価(SQuaRE)—システム及びソフトウェア品質モデル, 2013
- [5] Haruhiko Kaiya, Daisuke Shinbara, Jinichi Kawano, Motoshi Saeki, Improving the Detection of Requirements Discordances Among Stakeholders, Requirements Engineering, 10(4): 289-303, 2005