

数理科学アプローチを用いた客観的欠陥弁別法

～ 外因欠陥の弁別方法とその効果・意義 ～

The objective defect classification method with mathematical approach

- The significance of classification for “Exogenous defect” -

主査 : 細川 宣啓 (日本アイ・ビー・エム株式会社)

副主査: 永田 敦 (ソニー株式会社)

研究員 (五十音順):

伊藤 弘毅 (三菱電機株式会社)

大島 修 (エプソンアヴァシス株式会社)

角 修二 (株式会社インテリジェンスビジネスソリューションズ)

牟田 香奈 (日本 ATM 株式会社)

研究概要

ソフトウェア欠陥に関する研究の中で、未だ解のない領域に欠陥の「分類」がある。この分野の研究において、欠陥分析などを見れば「人が犯した過ちの種類」による主観的な分類が主流を占めており、いわゆる「人の過ちに起因する欠陥の特性での分類」が主流である。しかし、環境変化等に起因する欠陥（本稿ではこれを「外因欠陥」と呼ぶ）については、分類方法の研究以前に研究対象として扱われていない。

本研究チームは、欠陥分類の研究の過程で外因欠陥の新たな弁別方法を考案した。本稿は、テキストマイニング（アナリティクス）によって欠陥情報から定量的に取得できる特徴量を、k-means クラスタリング（デンドログラム）によって弁別する方法を提示する。さらに外因欠陥を客観的に弁別することの意義について考察する。

Abstract - The classification of software defects is a topic which has much room for discussion. Previous studies have discussed it mainly based on factors of human errors. However, little is known about the defect caused by the other factors, much less the classification in perspective of them.

Our study discussed that the classification method for software defects, which reveals the existence of defects caused by the environmental circumstance of the software development. This method extracts the characteristics of software defects by using a text mining approach and discriminates them which result from a k-means algorithm. Finally, we discuss the importance of the classification that our study presents.

1. はじめに

1.1 研究の背景

ハードウェア製品の場合、同じ設計、同じ製造ラインで組み立てられる構造は誰が作っても同じであることが期待されており、故障の原因は作業員個人の失敗以外に外的な環境変化や製造ラインの不備などを挙げるができる。例えば特性要因図（QC 七つ道具の一つ）[1]などを用いた原因分析では、いわゆる 4M（作業員—Man, 設備・治工具—Machine, 部品・材料—Material, 作業方法—Method）などに着目して人の失敗以外の多様な分析を行っており[2], 「統計的手法」に基づく分析を再発防止や未然防止に活かすことが求められている[3].

第7分科会 (Team TuKuLu)

一方、ソフトウェアの場合、投入する部品・材料は存在しないなど、ハードウェア製品とは様相が異なる。生み出されるコードは開発者の技量によってその出来映えが異なることから、あたかもバグは全て“欠陥の混入者が犯した過ち”であり、欠陥混入者の過ちは過ちを犯した人自身に原因があるとされるのが一般的である。つまり、バグと捉えるかは混入者と分析識者の主観で決定されているとも解釈できる。

本稿では欠陥混入者の過ちを「内因」、環境の変化や集団状況化の認知バイアスによるものなどを「外因」と表す。

1.2 従来研究の抱える問題

従来ソフトウェア欠陥研究は、大局的に見れば、前述の内因と環境や状況といった外因の相関の研究に他ならない。しかし、欠陥分析を行う場合など作業のアカウントビリティー（説明責任）が保証しやすいこと、つまりバグ混入者の能力不足や知識不足等の内因は原因として説明し易いことから、全て内因欠陥に帰着させてしまいがちなのである。内因の研究がその大勢を占め、どのような状況でその内因が発揮されてしまったかを十分に研究していると言い難いのはこのアカウントビリティーと関連すると考えた。さらに過度に内因に注目した欠陥分析が行われる場合、欠陥の発生を外因に求めることを「他責」と呼び、まるで悪であるかのように扱うことも少なくない。

我々は、過度に内因に偏向した欠陥分析が、企業が内情・失敗を公開して産業発展に寄与する文化を阻害し、欠陥研究の成果が広く公開されにくい環境にあることも関係していると考えた。

1.3 従来研究に対する素朴な疑問

前述のハードウェア製品故障との対比からも、本来ソフトウェア欠陥にも内因のみとは言い切れない欠陥が存在するのではないかという疑問がわく。

レビューチェックリストの利用などソフトウェア欠陥の再発予防策を実施あるいは実施方法を改善しているにも関わらず、組織単位で見た場合同じ欠陥が繰り返し発生しているという不可思議な事実常態化が起きているとしたら、実は対象は外因欠陥であり、再発防止策は内因欠陥への対策であった可能性がある。

本研究チームは内因をソフトウェア欠陥の特性の一つと捉え、さらに内因以外の欠陥研究が十分に行われていないことに着目した。従来研究の多くが「内因の分析に偏向していること」を本研究の出発点とし、内因以外の欠陥（＝外因欠陥）や内因欠陥を分析する手法について考案しようと考えた。

しかし、手法考案の過程でさらに重大な課題を見出した。一般的な欠陥分析の方法や原因分析手法には、そもそも内因と外因を区別して取り扱う概念そのものが余りない。唯一外因の例を示す[失敗原因マンダラ図[4]]ですら、著者らの主観・経験に基づく軸が示されているに過ぎないのである。

2. 研究課題

本研究の課題として、内因欠陥と外因欠陥の区別・弁別という命題を取り扱う。再掲すれば、内因欠陥は「開発者の過ちによって混入した欠陥」であり、外因欠陥は環境や制約、組織など開発者以外が原因となって混入した欠陥である。我々が最も主張したい点は、外因欠陥を内因欠陥と同じ対処・対応で撲滅しようとする「可能性」がある点である。より端的に述べれば、一般的に除去や対処の難しい外因に起因して混入した欠陥を、特定の開発者に帰結させることで、安易に「対処済み」として解決したことになっているシーンが少なくない点である。

もし欠陥分析の際に、分析対象を外因と内因に大別できたら、開発者に帰結させるべき欠陥（＝内因欠陥）を環境原因で曖昧に対処することもなくなる。反対に開発者に帰結さ

第7分科会 (Team TuKuLu)

せるべきでない欠陥 (=外因欠陥) をスキル増強やプロセス改善等の非現実的な労力と期間のかかる対処で打ち消すこともなくなる。

以上の理由から、本稿の課題として内因欠陥と外因欠陥の弁別を設定した。

2.1 内因・外因弁別に関する二つの課題

本研究チームが対象とする課題は、以下の二つである。

(1) ソフトウェア欠陥を内因・外因に弁別しそれぞれを分析する方法はあるか。

開発者の多くは経験上、開発者自身では気付けない欠陥が存在することを知っている。例え、開発規模に対して十分な開発期間や予算が得られない場合や未経験の技術を使った挑戦的な開発を行った場合などが挙げられるだろう。Project Fabre は欠陥の構造を表現する欠陥モデルの提案の中で、こうした人の過ちを誘発するものを「誘発因子」と呼ぶことを提唱している[5]。

また、人の過ちから遡らずにバグを発見できることも、よく知られている。一般に「勘と経験」と呼ばれているバグの検出は、実は暗黙知を利用した思考メカニズムであり、ソフトウェアレビューの先行研究において暗黙知や欠陥情報を活用した HDR 法[6]などが提案されている。

このような先行研究から、主観的・定性的な手法により外因でソフトウェア欠陥を分析することは現状でもある程度可能であると考えられる。本稿では後述(2)で示す実験後に考察する。

(2) ソフトウェア欠陥を客観的・定量的に内因・外因で弁別することは可能か。

問題管理票 (通称: バグ票) などからソフトウェア欠陥の原因や影響度合いなどを瞬時に読み取り把握する「直感」の正体は何であろうか。それは良い意味での斜め読みの技術であり、鳥瞰の技術であると考えられる。前者は文書要約の技術であり、後者は文書全体に散らばるキーワードの数や類似性を直感的に把握する技術と換言できる。そこから我々は、研究事例も多いテキストマイニングによる文書要約と、既に一般に実用化されているクラスタリング (付録 C. 参照) による分類[7]が利用可能ではないかと考えた。

先行研究においても文字列の類似性 (Similarity) を利用したものがあるが[8]、長文をそのまま扱っており、我々の目指す客観的・定量的なアプローチとはやや異なっている。

本研究チームは、欠陥研究としては未踏領域である外因欠陥に着目し、実験により「内因欠陥」と「外因欠陥」を客観的・定量的に弁別できることを証明する。

2.2 本研究の狙い

これまで行ってきたことが「内因欠陥の分析に偏向している」状況に比すれば、ソフトウェア欠陥から客観的・定量的に外因欠陥を弁別できるようになることは、即ち、外因欠陥の存在が証明されると共に、非現実的・効果的でない欠陥対処を行う確率が飛躍的に低減できることを意味する。

越えるべきさらなる壁はあるが、従来の内因偏重の原因分析手法が有効でない場合や、過ちの経験を活かすことのできない新しい欠陥の未然防止を計画する場合などに対し、外因欠陥への理解が解決の一つの鍵になると我々は考える。

3. 内因欠陥と外因欠陥の弁別方法の検討

3.1 弁別手法の選択

ソフトウェア欠陥情報を記述する文章から特徴量を抽出し、クラスタリングを行うことで、外因欠陥を他の内因欠陥との非類似度 (相対的な距離の遠さ) からある程度見つけることができることを考えた。

3.2 ソフトウェア欠陥情報からの内因の排除

前述の通り、一般的な欠陥の分類手法は、既得知識との照合による恣意的・人の経験に基づいた分類であり、識者の主観による分類となる。換言すれば分類を10回検討する場合は10通りの結果となる可能性がある。

この事態を回避するためには、恣意性が介在しない分類手法の選択が必要となる。我々はテキストマイニングを利用した手法を選択した。統計解析は主に「R」というプログラミング言語を用いた[9]。

バグ票から取得したソフトウェア欠陥情報から欠陥を表す文字列を品詞分解し、名詞・動詞などの単語の出現回数から、各種欠陥情報の「特徴量」を抽出することで、xx 忘れなどの内因欠陥と環境変化などの外因欠陥を弁別するアプローチを行う。

この形態素解析の結果から特定の単語（これをタームと呼ぶ）出現状況に局所性が観察される場合、即ち特定の文書集合にのみ特定単語が出現している際に、当該単語は文書分類の手がかりになることが多い。この性質を利用して、内因欠陥と外因欠陥の情報群に特定単語の差が現れるかを検証する手法を提案する。

以降の章でそれぞれの実験とその結果を掲げる。

4 実験と結果

4.1 実験の実施条件

(1) 実験に使用するソフトウェア欠陥情報は、本研究グループにて作成したアブストラクションシート（付録C.参照）を利用した。

バグ票同様、欠陥を個票形式で記録したものであり、インシデント情報（発生事象、実害、原因）及び欠陥情報（欠陥モデルの各要素（誘発因子、過失因子、プログラム欠陥、増幅因子、表出現象））の両方が記録されている。一般的なバグ票と異なる点は、固有名詞や案件固有の前提条件・制約事項等を削除し、抽象表現に置き換えることで汎化している点である。

(2) 今回の実験では書き癖の影響を極力排除し、実験結果が恣意的に導出されるリスクを低減するため、同一起票者により記述されたアブストラクションシートを選択した。

(3) ソフトウェア欠陥情報のテキストマイニングにはフリーソフトウェアとして誰でも入手可能であり、本ソフトウェアを利用した研究事例が豊富なことから、樋口耕一氏の開発するKH-Coder[10]を利用した[11]。KH-Coderは統計解析機能に前述の「R」、日本語の形態素解析ツールとして奈良先端科学技術大学院松本研究室で開発されたChasen[12]を利用している。

(4) 本実験ではソフトウェア欠陥情報の単語の重みづけの方法として、情報検索や文章要約などの分野で一般的に利用されているTF-IDF法を利用した[13][14]。（付録B.参照）特に、本実験では異なる作家の小説8作品を弁別した研究事例[15]を参考とした。

4.2 実験

4.2.1 実験の目的

外因欠陥情報1編を含むソフトウェア欠陥情報8編を入力とし、テキストマイニング(形態素解析+ターム行列生成)による特徴量抽出とクラスタリングを行う。この結果、外因欠陥情報1編を当該クラスタリング手法によって弁別できれば、外因欠陥の抽出手法として有意であることが証明される。

4.2.2 実験手順

外因欠陥情報1編(#6)と内因欠陥情報7編(#6以外)からなる内容の異なるソフトウェア欠陥情報8編を品詞分解し、どのタームがどの文書に出現するかをカウントするマ

第7分科会 (Team TuKuLu)

トリクス（これをターム行列という）を作成する。

外因欠陥情報として誘発因子に内因を持たないもの（組織が縦割りであるが故に曖昧なルールが正しいものとして信じられていたケース）を選択した。

手順 1) ソフトウェア欠陥情報 8 編の形態素解析を行う。この際、形態素解析のルールを以下の通りとする。

ルール① 対象となる品詞は「名詞」「動詞」「形容詞」「助詞」の 4 種類とする。

ルール② 出現頻度が 8 編合計で 2 以上になるタームのみを対象とする。

ルール③ 記号（句読点・句点・括弧他）はタームとして除外する。

手順 2) 1) の出力結果から、（縦軸：ターム、横軸：欠陥情報 8 編）のターム行列が生成される。（付録 A. 参照）

手順 3) 2) の出力結果から、各出現頻度を局所的・大域的重みにより TF-IDF 法を用いて影響を調整する。

手順 4) 出力結果を文章量によって正規化（ここでは一般的なコサイン正規化を行う）し、最終的に重みを調整する（ノーマライズ）。

手順 5) 上記ターム行列の各列間のユークリッド距離を計算する。

（特にクラスを 3～5 クラス提示して同じ結果が得られるか試算）

手順 6) k-Means クラスタリングにてユークリッド距離の近いもの（似ているもの）からクラスターが形成され、デンドログラムを生成する。

4.2.3 実験結果

デンドログラムの生成結果を下図に示す。

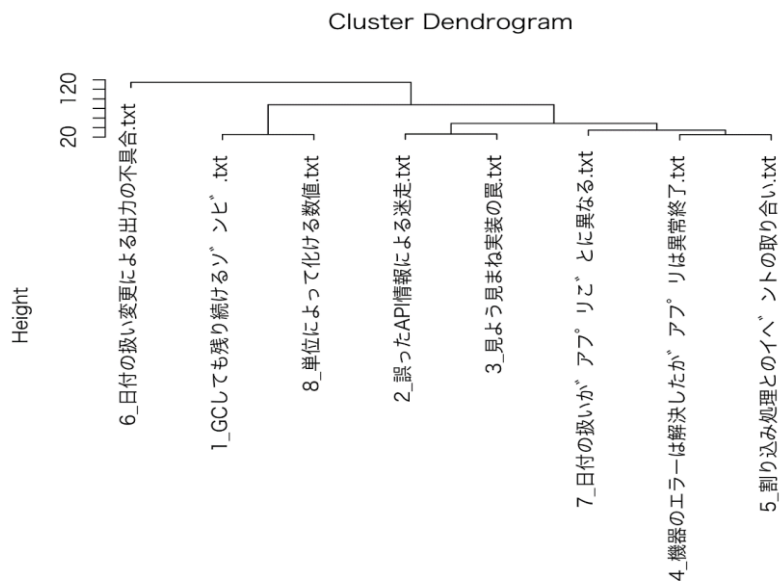


図 1. デンドログラム

本稿が示す方法にて、外因欠陥情報（#6）を期待通りに左端に弁別する（他の内因欠陥情報と別のクラスターに分類されている）ことができた。

5. 考察

5.1 本実験の別視点からの検証

第7分科会 (Team TuKuLu)

前章に示した実験により、本稿が示す方法にて外因欠陥を客観的・定量的に弁別できることが証明された。しかし、まだ課題がある。

デンドログラムでは個体間の類似度に基づき複数のデータの中から似ているものを順次に集めているに過ぎないため、具体的な分類（内因と外因）をクラスター自身が示したり、外因欠陥のみを狙って分離したりすることはできない。また、我々が弁別されることを期待した外因欠陥は、欠陥モデルに関する先行研究[16][17]に基づいているとは言え、誘発因子から主観的・定性的に外因と判断したものである。

そこで、どのような特徴を持つ欠陥情報が弁別されているかを別視点から検証するため、実験に使用した欠陥情報の単語出現頻度を紹介事例も多いワードクラウド (Word Cloud) (付録 C. 参照) により可視化した[18]。それが図 2 である。



図 2. ワードクラウドにて可視化した 8 文書の単語出現頻度

外因欠陥 (#6) のみ他の内因欠陥と様相が異なっている。外因欠陥 (#6) に特徴的なのは、内因欠陥 (#6 以外) に出現している「他動詞『する』」が出現しておらず、また「自動詞『なる』」の出現頻度が高いことである。

これらの特徴を即ち内因・外因の特徴であると結論付けるには本稿とは別の実験による証明が必要となるが、少なくとも今回の実験条件下においては、この様な特徴によって外因欠陥が弁別できたことを客観的に説明できた。

5.2 内因・外因弁別の効用とソフトウェア欠陥の新たな分析方法の提案

ここで前章 2. 研究課題(1)に挙げた、ソフトウェア欠陥を外因で分析する方法について話を戻そう。

先行研究事例として挙げた欠陥モデルを作成することで、外因欠陥を定性的に弁別することは可能ではあるが、欠陥分析担当者の主観が入り込む余地が多い上に、開発中に検出した全てのバグに対して当モデルを作成することは非現実的な労力がかかることも実験の前段階で明らかになった。欠陥モデルを作成することなく外因を弁別できるということは、即ち大量のバグ票を日々分析することが可能になるということである。例示すれば、ソフトウェアレビューやテストの実施半ばで外因欠陥が検出されたのであれば、品質の管理者や責任者が行うべきことは、バグを混入させた犯人捜しなどではなく、矛盾ルール、構成変更や変更通知の徹底など環境の整備に手を打ち、再発予防に努めることである。

第7分科会 (Team TuKuLu)

無論、我々もソフトウェア欠陥が内因とそれ以外とに単純に二分できると考えている訳ではない。またこの内外因の軸しか分類軸がないと主張したい訳でもない。分類軸はこの他にも多数存在すると考えられ、例えば品詞分解する際の重みを動詞や語尾にかけることや、欠陥混入位置や重要度など、様々な組み合わせの分類軸の調査も今後行うべきである。再掲するが、本研究の成果から得られた主張は、「バグを根絶すべくバグを原因ごと取り除きたいならば、欠陥を弁別し、外因欠陥にも目を向けるべきである」ということである。

6 まとめ

6.1 本研究の意義

本稿にて、外因欠陥（人に原因を求めるべきでない欠陥＝人を責めるべきでない欠陥）の存在を示し、外因欠陥の客観的・定量的弁別には統計的手法を用いることができることを明らかにした。

このことは、開発者のみに「欠陥混入の責任」があると必ずしも断定することはできず、プロセスや開発者の教育等の対策を講じることで解決できない欠陥の存在を示唆している。より平易な言葉で言えば、ソフトウェア欠陥の原因を人に求めている限り、バグは無くならないということを証明できたのである。

本研究は従来の原因分析手法や欠陥分析手法と対立するものではなく、本稿で扱った弁別法により外因欠陥の存在確認を行うことで、一般的な能力強化やプロセス拡充以外の対策を講じるべき対象が認識できることは大きな進化と考える。

6.2 今後の展望

外因欠陥の研究が広がることによって、近い将来ソフトウェア開発に従事する者が享受される恩恵を列挙する。

- ・欠陥モデルだけではなく、なぜなぜ分析や特性要因図などの従来分析手法でも、内因/外因を分析のフレームワークとして利用できるようになる。

- ・レビューチェックリストの利用など欠陥の再発予防策を実施しているにも関わらず、組織単位で見た場合、同じソフトウェア欠陥が繰り返し発生しているという不可思議な事実常態化が起きているとしたら、その対象は外因欠陥である可能性がある。内因欠陥の対策が外因欠陥には効かない場合、外因欠陥の再発防止策が新たに提案される。

- ・従来原因分類手法は「過去に起きた人の過ち」を人の手で整理したものであるため、新しい開発手法やテクノロジーの導入によって初めて直面する欠陥の発生に対しては、類推や予測は困難であると我々は考えている。

- ・世の中に広く知られている分類の例として、生物分類学がある。バグが人の過ちを誘発或いは増幅する因子の作用によって表出したものならば、各因子の特性を利用した系統分類もいずれ可能になる。生物分類学で行われている進化分類研究[19]と同様に、未知の欠陥特性が予測されるようになれば真の欠陥の未然防止が期待できる。

効かない欠陥の再発予防策や真の未然防止策の解決の鍵は、欠陥の内因ではなく外因の中に潜んでいる可能性がある。今後、外因のソフトウェア欠陥の存在が世に知られる様になれば、その分野の研究が進むことになる。

我々の研究がソフトウェア欠陥研究では未踏の領域である外因研究の扉を開け、小さな一歩を踏み出すことができたと信じている。

7. 参考文献

- [1] 日本工業規格 JIS Q 9024, マネジメントシステムのパフォーマンス改善-継続的改善の手順及び技法の指針, 2003.
- [2] 角田克彦, 広瀬淳, 市川亨司, QC 手法 I 実力診断とやさしい解説, 2007年3月16日編, 日科技連技連出版社, 2007, pp. 95-97.
- [3] 石川馨, QC シリーズ① 新編 品質管理入門 【A編】, 第51刷編, 日科技連出版社, 1979, pp. 43-56.
- [4] 松井秀雄, “失敗したITプロジェクトの真の原因に迫るマンダラ図の紹介,” 日本システム監査人協会近畿支部第152回 定例研究会, 2015.5.15.
- [5] 細川宣啓, 西康晴, 嬉野綾, 野中誠, 原佑貴子, “過失に着目した欠陥のモデリング-バグ分析はなぜうまくいかないのか?,” JaSST, 13 Tokyo, 2013.
- [6] 2012年度ソフトウェア品質管理研究会 第3分科会, “HDR法: 仮説駆動型レビュー手法の提案 -HDR法の実践による生産性と品質の同時向上-,” 2012.
- [7] 神嶋敏弘, “データマイニング分野のクラスタリング手法 (1),” 人工知能学会誌, 第18.1, 2003.
- [8] 吉村剛, 河野健二, “Linux カーネルにおけるバグの実態調査,” 研究報告システムソフトウェアとオペレーティング・システム (OS), 第2014.10, pp. 1-12, 2014.
- [9] 小林雄一郎, “Rによる英文テキスト解析,” 東洋大学社会学部紀要, pp. 51-52, 11 2015.
- [10] 樋口耕一, “KH-Coder,” . <http://khc.sourceforge.net/> .
- [11] 樋口耕一, “計量テキスト分析の提案と必要なソフトウェアの開発,” ソシオロジ 55(3), pp. 102-108, 2011.
- [12] 奈良先端科学技術大学院大学松本研究室, “chasen,” . <http://chasen-legacy.sourceforge.jp/> .
- [13] 石田基広, “Rによるテキストマイニング入門-初級,” . <http://rmecab.jp/wiki/index.php?plugin=attach&pcmd=open&file=ism091124doc.pdf&refer=RMeCabFunctions>.
- [14] TF-IDF法(TF=Term Frequency=単語の出現頻度)とIDF(Inverse Document Frequency=逆文書頻度), “ウィキペディア日本語版,” 13 9 2016. <https://ja.wikipedia.org/wiki/Tf-idf> .
- [15] 小林雄太, 村上裕一, 中村真吾, 橋本周司, “共起ネットワークを用いたクラスタ性によるテキスト分類,” 情報処理学会第74回全国大会6, 2012 .
- [16] 2014年度ソフトウェア品質管理研究会 第7分科会, “自律した品質改善活動に寄与する欠陥特性の提案,” 2014.
- [17] 2014年度ソフトウェア品質管理研究会 第7分科会, “欠陥構造の可視化によるトラブルの早期発見と未然防止~欠陥を中心に置くことで可能となるQAのコト作り~,” 2014.
- [18] 小林雄一郎, “Rによる英文テキスト解析,” 東洋大学社会学部紀要, pp. 57-60, 11 2015.
- [19] ウィキペディア日本語版, “分類学,” 15 12 2016. <https://ja.wikipedia.org/wiki/%E5%88%86%E9%A1%9E%E5%AD%A6>.