

言語ゲームに学ぶソフトウェア工学研究の新たな評価スタイル

鵜林 尚靖^{†1} 亀井 靖高^{†1}

本稿では、言語ゲームの視点からソフトウェア工学研究の新たな評価スタイルを提案する。

A New Evaluation Style for Software Engineering Research from the Perspective of Language-Game

NAOYASU UBAYASHI^{†1} and YASUTAKA KAMEI^{†1}

This paper proposes a new evaluation style for software engineering research from the perspective of language-game.

1. はじめに

我々は、過去数年にわたり、ソフトウェア工学における共通問題のあり方について検討し、その成果を「ソフトウェア工学の共通問題」特集号¹⁾にまとめた。

本稿では、次のステップとして、ソフトウェア工学研究の評価方法について考察する。

2. 評価はなぜ難しいのか？

ソフトウェア工学研究の評価は難しい。プログラミング言語やアルゴリズムなどの理論研究とソフトウェア工学などの実学の相違点は、実証が求められるか否かにある。理論研究の場合は証明などで十分であり、一般的に実証までは求められない。システム実装など応用寄りの研究は実験が必要となるが、その評価尺度は性能や速度など明確なものが多い。これに対して、ソフトウェア工学の場合は「有用性」が本質的な評価尺度となる。有用性は人間に依存する尺度であり、これを真に行うためには、多数の開発者を巻き込んだ大規模な実証実験を行うしかない。しかし、コストや労力の面で非現実的な場合が少なくなく、ソフトウェア工学に関する評価は残念ながら妥当性を欠いているものが多い。何をどこまでやれば実証したことになるのだろうか？ 非常に難しい課題である。

有用性の評価という立場から見た場合、多くの研究は砂上の楼閣のような存在なのかもしれない。オブジェクト指向を例に考えてみよう。オブジェクト指向

技術は、現在、多くの人が有用だと認める。しかし、本当にそれが有用かどうかを実証出来ているかと言うと実は出来ていない。一方、オブジェクト指向を対象としたプログラミング言語理論やプログラム解析の研究は多数ある。それらの論文は「現在のオブジェクト指向言語には○○のような問題（たとえば Aliasing に関する問題）があり、それを解決するための理論を提案する」といったストーリーで書かれる。オブジェクト指向の有用性を脇に置くか、あるいは有用だという前提の上で研究が行われる。しかし、オブジェクト指向が有用でなければ、本当は、このような研究に意味はないであろう。砂上の楼閣になってしまう。オブジェクト指向は一例に過ぎない。これに類することは数多く存在する。ソフトウェア工学以外の分野では砂上に楼閣を建ててもクレームがつかないのに、なぜ、ソフトウェア工学ではそれが許されないのだろうか？

3. ソフトウェア工学と言語ゲーム

ソフトウェア工学とはソフトウェアの作り方に関する学問領域である。作り方には、手法名とそれが意味するもの、すなわち「有用性が保証された開発手順」がなければならない。意味とは、本来どのようなものであろうか？ ヴィトゲンシュタインの言語ゲーム（「哲学探究」²⁾）では、語の意味とは「言語の中でのその使用（use）」だと言う。我々はオブジェクト指向のことをよく知っていると思っているかもしれないが、各自がどのような使用方法（以下、用法）をしているかをお互いに見比べながら（一種のゲームを興じながら）、何となく理解しているような気持ちになってい

^{†1} 九州大学

Kyushu University

る可能性が高い。オブジェクト指向の「使用」が有用性をも含めた意味を規定している。

ソフトウェア工学のプラクティス（分析、設計、テスト、デバッグなどの手法）はソフトウェア開発のための言語であり、開発技術を学ぶことは用法の習得であると解釈できる。用法が用法足りうるのは、それが無意識あるいは暗黙の了解の下で使われる場合である。前節で述べた例も、オブジェクト指向自体が「使用」により意味づけられており、その上の研究はオブジェクト指向の有用性を敢えて議論しなくても問題視されないのである。楼阁と砂の間には「用法」という土台が存在する。言語理論の研究はプログラミングパラダイムが定着した後に行われるので、用法として既に開発者間で意識の共有が出来ている。一方、ソフトウェア工学の研究は「新たな用法」を提供することであり、新規性やオリジナリティは用法の「新しさ」に依存する。ここに大きなジレンマが存在する。研究として提案したものは、まだ開発者間で意識が共有されてなく、実は用法ではないのである。用法を提案しつつ、それは用法の条件を満たさないという自己矛盾をソフトウェア工学は抱えている。用法になるには長い年月を経て人々の間での使用される必要がある。確立した用法からスタートする研究と、用法として確立しなければならぬ研究の間には大きな違いがあり、それが評価の困難さの違いとなっている。ソフトウェア工学は後者のタイプである。用法として確立しなかった研究は、楼阁を支える土台になれなかった研究である。残念なことに、ソフトウェア工学研究のほとんどが、用法として確立する前に評価を打ち切られている。

4. 新たな評価スタイル

ソフトウェア工学の「意味」は「使用」により決定されるため、研究評価には「使用」が必須となる。しかし、新しい用法が意味づけ可能になるには、長期間にわたる使用とその用法を使用する人々の広がりが必要となる。研究評価という観点からは、意味づけがなされるまでの期間を短縮することが重要となる。本稿では、以下の3つの評価スタイルを提案する。

スタイル 1: 過去の用法に学ぶ

最も評価を行いやすいのは、既存資産から有益な用法（バグ予測、デバッグ支援、パターンなど）を抽出するアプローチである。OSS（Open Source Software）の世界では、ソースコードリポジトリ、メーリングリスト、バグDBなどに大量の開発資産が蓄積されている。これらのリポジトリをマイニングすることにより様々な有益な用法が抽出できる。現在、ICSEなどの

トップカンファレンスで採録される論文の大半は、この評価スタイルに基づいている。しかし、現状、このスタイルで評価できるのは、コード周りの下流に関するもののみである。設計モデルなどの上流工程のデータは残念ながらリポジトリには格納されていない。一部のドキュメントをテキストマイニングできるに留まっている。今後、モデル情報がリポジトリに格納されるのが一般的になれば、状況は変わると思われる。

スタイル 2: アジャイルに用法確立

用法を確立するのに時間がかかるのであれば、それを実験的に短縮できると嬉しい。生命科学では世代交代の期間が短い種を用いて、進化過程に関する実験を行うことが多い。ソフトウェア工学でも同様な実験が実施できるとよい。新たに提案する用法をアジャイル的に複数回反復適用し、そのログデータを分析することにより、その用法の有用性の変化を数値的に評価できる。この評価スタイルでは、過去ではなく現在進行形のプロジェクトを通じて学習的に用法を確立できる。

スタイル 3: 集合知による用法確立

用法が確立するための期間を実験的に短縮できたとしても、利用者に広がりが必要ならば、単なる1プロジェクトでの適用事例に過ぎない。

最近では、インターネット上で試行ボランティアを募り、その結果をフィードバックして貰うという評価スタイルが定着しつつある。このアプローチを採用することにより、それなりの被験者数を確保できる。

5. まとめ

一世代前のソフトウェア工学研究では、企業での事例研究による評価あるいは大学研究室での制御された実験による評価しか出来なかった。OSSの時代になり大量のデータを対象に統計的な実験評価が可能になった。現在、我々は、リポジトリマイニング、アジャイル、インターネット上での集合知、などデータに基づいた評価に利用可能な新たな道具を手に入れている。

これからのソフトウェア工学研究では、OSS研究に限らず様々な研究領域でデータに基づいた評価を行って行くことが重要である。そろそろ古いスタイルの評価手法を捨てる時期に差し掛かっている。

参考文献

- 1) 岸知二, 野田夏子(編集): 特集 ソフトウェア工学の共通問題, 情報処理 Vol.54 No.9, pp.876-902, 2013.
- 2) ヴイトゲンシュタイン: 論理哲学論考, 藤本隆志/坂井秀寿訳, 法政大学出版局, 1968.