CS846-001 Week 2 Reviews

Jun Qing Lim
University of Waterloo
jq3lim@uwaterloo.ca

1 No Silver Bullet — Essence and Accident in Software Engineering

1.1 Problem Being Solved

The paper gives an overview on the construction and background of software, citing that software is hard in essence and not by accidents (or consequences). In "essence" here refers to the essential difficulties that cannot be avoided due to the inherent nature of software, which are: complexity, conformity, changeability, and invinsibility of a software. It investigates into the possibilities of silver bullets such as high-level language advances, object-oriented programming but none of these are able to tackle the essence of software engineering, and proposed a couple of promising ideas on approaching software development.

1.2 New Idea Proposed

The esssential difficulties of software engineering could be made easier by:

1. buying software instead of building it, which in turn reduces the total cost of development.
2. refine requirements and do rapid prototyping, this will reduce the issue of determining precisely what to build and refine the product requirements over time
3. do incremental development, by growing software incrementally and start small
4. invest on crafting good designers (engineers) by having certain set of good engineering practices in the team in nurturing more creative ideas

1.3 Positive Points

Firstly, it gives the reader an awareness that software engineering is still in essence a field that is relatively young and constantly evolving and developing where there is no method that is "for sure" in developing software, making the process difficult. Secondly, the paper is written broadly and simple manner, providing greater accessibility to a wide spectrum of readers. Thirdly, the content of the paper is still relevant today even decades after the first publication, which makes it one of the fundamental elements of software engineering principles.

1.4 Negative Points

Firstly, the paper might be old given the current state of AI and ML development, which may foster the possibility of automated/intelligent/advanced programming which in turn could eliminate the essence of software engineering completely. Secondly, the paper gave a couple of possibilities of possible silver bullets and argued that why they could not be silver bullets, which may be quite pessimistic as it does not cover all possible cases, but some possible cases.

1.5 Future Work

As illustrated above in the negative points section, one possible extension to the research work is investigate how Artificial Intelligence and Machine Learning affect the current state of software development, which may in turn affect the complexity of essential and accidental software development. Besides that, the author conducted a broad-research and not domain specific, thus one other extension could be investigate the effects of software development's abstraction in different domains, allowing us to better understand essential and accidental complexity with a greater depth.

1.6 Rating

5 out of 5. A paper that should be read by all software engineers as the core foundation of developments of software.

1.7 Discussion Points

- The paper was written few decades ago, are there any information or current state of development which affected the current landscape of this paper?
- How can the advancement of new and emerging technologies help to tackle the essential difficulties?

2 The Truth, the Whole Truth, and Nothing but the Truth: A Pragmatic Guide to Assessing Empirical Evaluations

2.1 Problem Being Solved

The paper describes how readers or researchers could accurately evaluate the claim illustrated by an author of a paper through critical analysis. It provides a framework to identify unsound claims through sins of reasoning where claims are not supported by evaluations and sins of exposition where there is not enough information for the readers to reproduce the evaluation or claim. Through the use of these techniques, the reader is able to assess the accuracy and the reliability of the findings.

2.2 New Idea Proposed

The paper provides a framework to evaluate the validity of the research findings, which is looking at the content with consideration of different perspectives and evidences of the claims. This could range from:

1. evaluating if author is providing ambiguous evidence (trying to distort or obfuscate original claims),
2. checking if claims are reproducible,
3. imprecise explanations,
4. lack of concrete evidences,
5. providing contradictory statements,
6. inappropriateness of statements,
7. inconsistent claims

If a reader considers the above aspect in-mind while critically reading the paper, it could help eliminate unsounds claims when the assessing empircial evaluations of a paper.

2.3 Positive Points

Firstly, it provides researchers a simple framework on a critical evaluation over the validity of the content (i.e., good or bad paper), which in turn help researchers to decide whether or not to consider drawing statements from a paper as part of its research understanding. Secondly, the paper is written in a general manner, which helps researchers of different levels of experience as it is easily understood. Thirdly, it gives authors an idea on how to write a paper with accurate and sound evaluation, that contains information that is vigorously self-validated before conducting a full publication.

2.4 Negative Points

Firstly, the nature of sins in a paper might be difficult to decipher - it is possible some readers could identify a particular sin but not for other readers as the paper does not cover in depth of the different variations of sins (i.e., no concrete approach on how readers could eliminate unsound claims completely). Secondly, it is possible that not all kinds of research domain will be suitable with the paper's methods of assessing the validity of empirical findings.

2.5 Future Work

We could extend this paper by conducting another research on the roles of the techniques played in the current state of advancement of software engineering, which will further reinforce latest findings on its relevancy. Furthermore, as a software engineering research may have certain biases in certain domains and that researchers may not be able to apply the said techniques in all domains, we could extend the research work by assessing the relationship between research domain area and the sins to better understand its applicability.

2.6 Rating

4 out of 5. Provides useful guide for both authors on how to write paper with valid and sound ideas and for readers on to detect invalid and unsound ideas.

2.7 Discussion Points

- How can the current state of software engineering adapt and fit to these guideliness?
- How should these principles of assessing validity of a paper be applied in domain specific areas?