

CS846 Week 1 Silver Bullet Summary

Jacie Jermier

Opinion

I very much enjoyed reading this article as it provided an interpretation of the core of software engineering I hadn't otherwise put much thought in to. It also helped to shine light on specific frustrations that commonly appear in a software engineer's world. From my own experience in the field, I can relate to the complexity that is the intellectual effort when writing software in a sea of requirements and strategic goals. There is also an art to gathering requirements from non-technical stakeholders who often lack clarity and clear goals. Having read this paper, it helps to not only compartmentalize the different aspects of software engineering but also gives the reader a lens to navigate frustrations more effectively. It also stood the test of time. AI and agentic systems still can't serve as a silver bullet.

I am still of the opinion that we will keep inching our way closer to an order of magnitude leap. It won't be a single silver bullet, but rather a multifaceted solution that chips away at the individual parts of the essence of software. I believe we're entering a period where some of the essential difficulties of software engineering are becoming less dominant than others. It may be in the future that AI does understand our code base better than we do. I think that is a very real possibility. Every model has gotten increasingly better from the last. I'd like to believe that there will be another layer between us and the complexity of code. Just as we don't learn assembly language anymore, maybe in the future we do not code, but rather write English. This isn't to say the problems that we will need to solve in the future get less complex, but the way in which we implement solutions could become less complex.

Class Discussion

I also enjoyed the class discussion. I loved where it went and how freely people were able to discuss their opinions. A large part of the discussion focused on whether software developers are essentially programmers or whether their role involves a wider

set of skills. Most participants on this topic believed that it is still important for software developers to learn how to code. Even with great AI tools, there still will be a need for people to write code. Especially in the sense of writing tests, and quality control. Ultimately, the class concluded that at our core, software engineers need to be able to critically think. That is their highest value and use to a company. One student however, did bring up an interesting point that if AI were to code entire code bases, this increases complexity, especially now being a degree removed from the code base. As well, it was mentioned that human oversight is still very much needed with AI tools, which can overcomplicate solutions or simply not meet the requirement it is being asked to do.

Secondly, the class discussed whether they think it will be possible to reach a 10x leap in productivity. Most on this topic did not believe the future holds this type of advancement in software engineering. The main reason being the current set of tools only provide a percentage of correct code, and that they are simply tools not intelligent systems that can think. The class did, however, believe that organizational restructuring with the emergence of such tools may be required to gain a significant productivity leap. One student noted that the role of sales engineers in the field exemplifies how certain aspects of software engineering may require less technical expertise in the future.

Lastly, the class discussed the concept of a software essential becoming an asset rather than a challenge to overcome. When speaking specifically on changeability, most students agreed that flexibility comes with a cost and more specifically an intellectual cost. I tend to agree with this notion. However, like I said in my presentation with the whack-a-mole analogy, I think the essential challenge of changeability might not be as big a deal in the future. With agentic systems and computing power getting cheaper by the day, the cost of making changes is shrinking. It'll probably still pop up, but

I'm bullish on it becoming more of an asset than a major hurdle. I think we'll still need to keep swinging at the mole of changeability, given how fast AI is embedding into society and driving constant change.