A critical analysis of the paper: "Today was a Good Day: The Daily Life of Software Developers"

Sandesh Bhusal

Jan 10, 2024

This paper on developer experience at Microsoft was a fascinating read, resonating with a lot of its aspects while leaving me slightly disagreeing with some other aspects of it. The study's focus solely on Microsoft was concerning to me - the results might be skewed by their internal "culture bubble." The authors make attempt to quell these concerns, but a company always has a culture bubble no matter the geographical spread and team diversity.

One finding that hit home for me as a developer (and I believe one of the most important conclusions of this work), was the critical role of developer agency. Having a clear headspace and understanding my day's structure makes all the difference in how effectively I use my time.

However, one point struck me as paradoxical: the paper mentioned reducing meetings and infrastructure issues wasn't key to developer productivity, even though they're clearly not seen as productive tasks. Scheduling meetings around longer stretches of focused work could be better, considering the average 47-minute task duration. This might be tricky for larger teams with scheduling conflicts, but it might be worth exploring. Plus, the 15-minute post-meeting context switches and the 20% overall meeting time deemed unproductive makes one wonder how they can be strategically restructured.

Another eyebrow-raiser came from the non-anonymous participants: a whopping (low) 6.6% of them answered twice! This makes me question the survey's methods, specifically their claim about sampling questions and other techniques reducing cognitive load. While the authors successfully answered their initial research questions and the framework seems solid, I crave more information about information congruence among these repeat participants.

It was also interesting to see junior developers rating collaboration lower than seniors, even though they stand to benefit the most. I suspect this stems from seniors' involvement in crucial decisions like system and architecture, which naturally enhances their appreciation for teamwork, while junior developers would mostly be concerned with cementing their place in their teams by "showing their work".

The paper's section on meetings and context switching truly shines with its detailed analysis. Considering the mix of senior and junior developers within teams, achieving the holy grail of "optimized developer workdays" seems like a pipe dream due to differing perceptions of meetings and code production. However, focusing on factors like individual developer autonomy, location, team work allocation, and workday type (among the 7 critical factors) could unlock a path to tailor-made efficiency leading us closer to the dream.