

### **Towards a Theory of Software Development Expertise**

To be honest, I don't think I fully grasped everything the paper was trying to explain. A lot of it felt bloated and I felt like I had to read most of the lines multiple times to understand them.

While I did think the survey responses would be very subjective, I do suppose that it being a random selection of people and the sample size not being too small would lead to at least some valuable insights being formed.

I liked Research Question 1, since I myself was interested to see what kind of characteristics developers assigned to novices and experts respectively. However, I wasn't a fan of Research Question 2, since I don't believe that the challenges the developers faced was necessarily indicative of their expertise. I would have instead preferred to see more research about how expertise was related to experience, since I found the part that said that they weren't that closely linked to be one of the more interesting parts of the paper.

### **Automatically Recommending Code Reviewers Based on Their Expertise: An Empirical Comparison**

I found this paper to be surprisingly interesting for one main reason: I didn't know until now, and hadn't even considered, that finding a suitable reviewer was such a notable problem and that it could cause major delays for the acceptance of code patches. I liked that this paper tackled a problem that I wasn't aware of, so it felt quite innovative to me personally.

However I didn't find the research and its results to be particularly compelling. Just analyzing the different algorithms and comparing their results to their actual reviewer didn't feel to me like a good metric. The actual reviewer isn't necessarily the ideal reviewer in the first place so this would lead to a lot of false negatives. The authors address this in their threats to validity, but I believe this is a much larger threat to the paper's validity than the authors claim it is.

## **Do You Remember This Source Code?**

I agree with all the assertions the paper makes, which I believe also makes the paper less valuable.

I think that it is reasonable to assume that greater familiarity with a file or code would lead to a greater level of expertise, and it is equally reasonable to assume that overtime a person would forget the minutiae of the code and cause some amount of decrease in expertise. So in that sense I do not understand the motivation the authors to conduct the research in the first place.

However, validating that the forgetting curve of Ebbinghaus is applicable even for software development was an important outcome which could be valuable for future research on the matter.