Paper 3

1) **Title:** Here We Go Again: Why Is It Difficult for Developers to Learn Another Programming Language?

Authors:

- Nischal Shrestha
- Colton Botta
- Titus Barik
- Chris Parnin

Conference: In 42nd International Conference on Software Engineering (ICSE'20), May 23–29, 2020, Seoul, Republic of Korea. ACM, New York, NY, USA.

2) Introduction:

Various stories on language changes propose that even experienced developers experience issues learning new dialects. For instance, a Java software engineer who progressed to Kotlin reports that distinctions like turned around type documentation and how classes in Kotlin are last as a matter of course, made the progress less smooth than anticipated: "on the off chance that you imagine that you can learn Kotlin rapidly in light of the fact that you definitely know Java—you are incorrect. Kotlin would toss you in the profound end." Similarly, a software engineer experienced in C++ who changed to Rust found that Rust's acquire checker, "powers a developer to think in an unexpected way." Transitions across profoundly various dialects are particularly troublesome. For instance, a Java developer changed to Haskell and communicated that "the simple things are frequently somewhat harder to do in Haskell," and another developer experienced in procedural dialects cautioned that "[lazy evaluation] can be somewhat befuddling to see how it functions by and by particularly in case you're despite everything thinking like an objective developer." Even dialects having the equivalent runtime can be hazardous: "at whatever point I get Coffee Script, I feel as though most of my comprehension of JavaScript out of nowhere evaporates like a phantom." From these accounts, one regular abstain happens: past programming information is now and then less accommodating than anticipated, and can effectively meddle with learning. This appears to be strange.

Motivation:

We found that:

- Cross-language interference is a problem: 276 (61%) crosslanguage posts on Stack Overflow contained incorrect assumptions due to interference with previous language knowledge.
- Based on our interviews, professional programmers primarily learned new languages on their own, using an opportunistic strategy that often involved relating the new language to previous language knowledge; however, this results in interference which harms their learning.
- Learning a new language involves breaking down old habits, shifting one's mindset, dealing with little-to-no mapping to previous languages, searching for proper documentation, and retooling in a new environment. All together, these challenges make learning another language difficult.

3) Research Methodology:

To explore how programmers, learn a new language, and understand their potential sources of confusion, we conducted a mixed-methods
study through an empirical investigation of Stack Overflow posts across various languages and through semi-structured interviews.
We do so through the following research questions:

RESEARCH QUESTIONS:

- RQ1: Does cross-language interference occur? We examined questions programmers had about programming languages on Stack Overflow for evidence of interference with previous programming knowledge.
- RQ2: How do experienced programmers learn new languages? To gain a better understanding of why cross-language interference occurs, we interviewed professional programmers on how they learn new languages.
- RQ3: What do experienced programmers find confusing in new languages? To examine the ways in which programmers mix a new language with their previous knowledge, we asked programmers about obstacles they faced, and surprises they encountered in their new languages.

4) Result:

RQ1: Does cross-language interference occur?
Cross-language interference occurs on Stack Overflow across various language pairs. We found a total of 276 instances of incorrect assumptions (Table 2), which is around 61% of the 450 posts inspected. There was a total of 174 posts with correctly stated assumptions, which is only around 39% of the total posts. It's important to note that this provides evidence of interference

occurring but does not imply programmers have incorrect assumptions 61% of the time. The <Kotlin, Java> pair had the highest number of posts with incorrect assumptions, which reflects the Java programmer's confusion mentioned in Section 1. The next two pairs, <C#, Visual Basic> and <Scala,
Java>, also contained a high number of incorrect assumptions. However, there were other pairs like <Python, C++>, <Java, C#>, and <PHP, Java>, which had a more even distribution of posts with correct and incorrect assumptions; this suggests easier transitions between the languages. While reviewing the 450 Stack Overflow posts, we encountered instances where programming languages behaved in surprising ways for programmers. We highlight three examples, two of which involved interference between syntax and concepts, and one which involved facilitation—making it easier to use type inference.

RQ2: How do experienced programmers learn new languages?

Programmers learned languages on their own. Programmers who switched teams lacked formal training for the new language and its associated technology stack, leaving learning to themselves. For example, when P1 switched from C# to Python for a new project, there wasn't any training involved and the onboarding process was, "hey we want to get exposed to the Python world, go get started!" Although some programmers were given training initially on the project, "realistically for learning the new language [they] were pretty much on [their] own" (P7). This forced programmers to watch "language tutorial videos on Pluralsight"7 (P5) or read online documentation Some programmers "got initial tips from some folks from the team on what's what" (P6), and when running into complex issues "reach out to the group and said has somebody else hit this before?" (P1).

RQ3: What do experienced programmers find confusing in new languages?

Old habits die hard. Programmers had to constantly suppress old habits acquired from previous languages. For example, P3—who was used to Python—had trouble adapting to block delimiters in PHP, where "it's near-impossible to figure out exactly which opening brace you're closing once your HTML/PHP gets to any complexity at all." Similarly, P15 realized that "in Swift, the open curly bracket needs to be on the initial line of the method declaration and if you put it on the next line the method may not execute in an expected fashion." There were minor but frustrating difference like 0 versus 1 indexing for lists in languages such as Python and R. P4 described their frustration in "typing a[1] thinking that it's a[0], and then wasting 5 minutes like a complete fool not understanding why nothing makes sense" (P4). Programmers are able to resolve these small differences, but it still causes interference at the onset of learning a new language.