카뎃들의 동료평가 아쉬운 점

“카뎃들의 동료평가를 보면서 한 가지 아쉬운 점이 있어요. 우리는 평가가 감정적인 행위가 아니라는 믿음을 가져야 해요. 그래서 치팅이 의심될 때도 엄격하게 평가하고, 그 과정에서 감정을 안 쌓는 것이 협업을 훈련할 때 중요해요. 회사에서 코드 리뷰를 할 때도 내 코드가 난도질을 당할 수도 있지만, UFC 경기가 끝나면 다시 화해하는 것처럼 쿨하게 받아들여야 하는데 아직은 그런 훈련이 좀 덜 돼 있는 것 같아요. 감정적으로 실패를 받아들이면, 실패가 실력으로 쌓이지 않아요. 그러니까 평가를 감정적으로 받아들이지 않는 훈련을 계속해야 하고요.”

“가장 좋아하는 프로그래밍 언어는 C언어에요. 제가 접해본 언어는 파스칼과 C밖에 없는데, C를 접한 이후로는 거의 완벽한 언어라고 생각하고 있어요. low-level에서 원하는 것을 다 할 수 있어서 좋고, 특히 제가 일했던 분야는 임베디드 쪽이라 모든 일이 C 기반이었어요.”

“우리나라는 C보다는 자바, 자바스크립트, Python 등이 대세인데, 요즘 언어들을 보면서 점점 지저분해진다고 느껴져요. 원래 시스템이나 라이브러리들이 해야 되는 일들을 언어가 일부 담당하기 시작했어요. (대표적으로 자바스크립트가 그렇고요.) 그래서 언어와 라이브러리의 경계가 희미해지면서 지저분해지고 있다고 느끼고 있고, 반대로 C는 프로그래밍 언어가 필요한 기능에다가 실제 하드웨어를 고려한 기능들(예를 들어 포인터)만 추가한 셈이죠.”

“다른 언어들이 C에 비해 쉽다는 의견에는 동의를 못하겠어요. 파이썬을 예로 들어보면, 파이썬이 배우기 쉬운 이유는 C의 진입장벽을 맡고 있는 포인터가 없고, 문법이 간단하다는 것 외에는 별 거 없어요. 그런데 신택스 에러 나면 바로 고치면 되고, 진짜 문제는 로직에서 나오는 에러거든요. 그런 거는 언어가 다르다고 해서 안 나오는 에러가 아니고요. 물론 프로그래머가 자료형을 지정하는 C와는 다르게 자동으로 자료형을 정해준다는 편리함이 있긴 해요. 근데 자료형이 엉키면 나중에 사고를 치게 되어있어요. 그걸 겪고 나면 자료형이 잘 정의된 C같은 언어가 낫다고 생각하게 돼요. 물론 파이썬은 생산성이 좋다는 장점이 있죠. C언어가 주력이라도 프로토타입은 Python으로 만드는 사람들도 많고요. 이미 만들어진 많은 것들을 활용하기 위해 파이썬을 쓰는 것은 꽤 좋은 작전이에요. ”

--

“좋은 코드는 읽기 쉬운 코드가 최고에요. 코드는 나만을 위한 것이 아니라 모두를 위한 것이거든요. 오픈 소스든 회사의 프로그램이든 다른 사람이 나의 코드를 보고 리뷰를 하기 때문에 읽기 쉬운 코드가 제일 좋은 코드에요. 요즘 언어들이 추상적으로 코드를 쓰면서 생산성을 높이려고 하는데, 제 생각엔 아름답지 않은 모습 같아요. 코드는 읽는 사람이 고민하게 만들면 안 되거든요. 그래서 좋은 코드는 쓰기 쉬운 코드가 아니라 보기 쉬운 코드를 지향해야 합니다.”

--

“좋은 개발자가 되려면 필요한 역량 중 두 가지는 문제를 이해하는 역량과 해결 방법을 찾고 설명하는 역량이에요. 여기서 얘기하고 싶은 것은 해결 방법을 설명하는 역량인데요. 설명할 때 쓰는 용어가 매우 중요해요. 소프트웨어를 배울 때 책들을 공부했으면 접했을 용어들이 있거든요? 그 용어들을 써서 다른 개발자들에게 설명해야 해요. 제가 42학생들한테 ‘문제를 풀 때 주어진 책을 공부해야 된다’고 많이 얘기해요. 책을 공부 안 하면은 나중에 질문을 할 때 용어를 써서 설명할 수가 없어요. 그래서 책을 공부하는 게 매우 중요하답니다.”

“예시를 몇 개 들어보면, 42에 웹서버를 만드는 과제가 있잖아요? 웹서버를 만들면 네트워크 프로토콜, 패킷 등을 알아야 하고 그런 용어들은 책에 다 써있어요. Shell을 만들려면 프로세스가 무엇인지, 어떻게 프로세스 통신이 일어나는지, 그리고 시스템 콜이 무엇인지 알아야 하는데, 책을 안 보면 모를 수밖에 없는 거죠. 프로그램에 대한 질문을 던졌을 때 단어조차 처음 들어봤다는 반응을 하면 천재가 아닌 이상 코드를 가져다 썼다는 인상을 줄 수밖에 없게 되죠. 그래서 선수들이 쓰는 용어를 배우기 위해 책을 공부해야 합니다.”