

나는 개발 문화를 어떻게 만들어 갔을까?

개발 회사의 문화란 어떻게 만들어지는 걸까? 글을 읽고 2008년의 내 경험이 떠올라 공유하고 싶은 마음에 글을 쓴다. 정확히 말하면 2007년 11월에 새로운 팀을 맡았다. 팀장 역할을 시작하고 8, 9개월 정도 되었을 즈음 기존의 팀이 해체되고 새로운 팀을 맡게 되었다. 처음 같이 한 팀은 처음 맡게 된 팀이라 정성도 많이 쏟았고, 팀워크도 너무 좋았다. 2년 정도의 로드맵을 세우고 팀의 문화를 만들어 가고 있는데 갑작스럽게 해체되어 안타까운 마음이 컸다.

처음 팀은 기존의 서비스를 운영하는 상황에서 맡았기 때문에 기존의 개발 문화가 있었고, 기존 문화를 바꿔가는데 더 많은 시간을 쏟아야 하는 상황이었다. 하지만 새로 맡게 된 팀은 서비스도 처음부터 개발할 수 있었고, 인원도 많지 않은 상황이었다. 따라서 팀을 만드는 초반부터 팀의 개발 문화를 잘 구축해 놓는다면 이후에도 지속할 수 있을 것이라 생각했다. 처음 시작이 잘못되면 이후에 다시 반복하기란 쉽지 않음을 알기 때문에 프로젝트 초반 개발 문화를 형성하는데 집중했다.

처음부터 나 뿐만 아니라 같이 참여하는 팀원들의 의욕도 강했다. 서로가 적극적이라는 것은 얼마나 행운인가? 하지만 우리가 처음에 집중한 것은 대부분의 개발자가 그러하듯이 기술적인 측면에 너무 집중해서 접근하기 시작했다. 어쩌면 팀원들의 이슈가 아니라 내 자신이 기술적인 측면에 너무 집중하고 있었던 것 같다. 먼저 같이 하는 팀원들의 의견을 수렴하고 그 친구들이 자연스럽게 찾아갈 수 있는 환경을 만들어야 하는데 애자일 프로세스에서 좋다고 하는 프랙티스들은 일단 적용하고 보자는 식의 접근 방식을 취했다. 이 부분에서는 애자일 프로세스를 경험해보고 싶었던 나의 욕심이 과했던 측면도 있었던 것으로 생각한다.

이렇게 우왕 좌왕 하면서 다양한 시도를 해보던 차에 사내에서 진행하는 팀장 리더십 교육에 참석하게 되었다. 팀을 만들고 팀장의 역할을 하는 것에 대한 많은 조언을 들었지만 내 기억에 남는 한 마디는 "나라면 어떻게 하겠니?", "그 문제는 어떻게 해결할 수 있을까?"였다. 그 때 교육을 담당하신 강사분의 말은 이랬다.

대부분의 팀장 또는 그 이상의 관리자들은 팀원들과 면담을 할 때 자신이 모든 해결책을 제시하려고 한다. 자신이 팀원보다 높은 연봉을 받고 인정을 받아야 된다고 생각하기 때문에 팀원이 고민을 말하면 그에 대한 해결책을 제시해 주어야 한다는 강박관념에 시달린다는 것이다. 그렇다보니 면담을 하면 면담 대상자인 팀원보다 팀장인 자신이 더 많은 말을 한다는 것이다.

팀원과 면담을 할 때 먼저 자연스럽게 이야기할 수 있는 환경을 만들고 팀원의 고민을 들어주면 된다. 팀원이 현재 가지고 있는 고민(여기서 고민이라는 것은 삶에 대한 고민도 있겠지만 현재 진행하는 프로젝트, 일하고 있는 팀에 대한 고민을 말한다.)에 대해서 이야기하는 순간 팀장은 자신이 이에 대한 해결책을 제시하려 하지 말고 "나라면 어떻게 하겠니?", "그 문제는 어떻게 해결할 수 있을까?"라는 한마디만 던지라는 것이다. 이미 팀원은 그에 대한 자신의 해결책을 가지고 있으며, 자신이 해보고 싶은 프랙티스를 가지고 있는 경우가 대부분이라는 것이다. 즉, 변화의 주도권을 팀장이 아닌 팀원에게 전가하라는 것이고, 팀장은 이에 대한 지원자 역할만 하면 된다는 것이다.

난 내가 잘 모르겠으면 책에 나온 방법이나 강사를 통해 들은 내용을 일단 따라해보는 편이다. 왜냐고? 일단은 나보다 더 많은 경험을 가진 사람들의 이야기이기 때문에 무조건 수용하고 따라해보는 경향이 있다. 그래서 위 방법도 일단 그대로 따라해 봤다. 나는 팀장이 되었지만 프로그래밍을 좋아했고, 개발하는 시간이 가장 즐거웠지만 꼭 참고 한 달에 한 번은 반드시 팀원과 일대일 면담을 하기로 마음 먹었다.

첫 번째 면담에서 나온 이슈는 개발자간의 공유가 잘 되지 않는다는 것이 주된 내용이었다. 그 당시 팀 내에 소스 코드 리뷰를 했지만 그 당시 실효성이 없다는 것이다. 그래서 위 방법대로 이에 대한 해결책을 팀원들 개개인에게 물어봤다. 그랬더니 몇명의 팀원이 "짜 프로그램밍을 한 번 진행해보면 어떻게?"는 의견이 나왔다. 일대일 면담을 진행하면서 짜 프로그램밍에 대한 각자의 의견을 물어봤다. 그랬더니 한번 진행해보자는 의견이 나왔다. 내 입장에서는 아직까지 짜 프로그램밍에 대한 확신이 없는 상태에서 일정상 무리가 있을지도 모르겠다는 생각이 들었다. 하지만 일단 한달만 진행해보고 효과가 없으면 폐기하자는 방향으로 합의점을 찾았다. 짜 프로그램밍은 쉽지 않았다. 중간 중간에 마찰음이 생겼고, 여기 저기서 개발자간의 싸움도 있었다. 그래도 본인들이 시작하자고 했으니 하지 말자는 이야기는 못했다.

기존과는 완전히 새로운 프랙티스를 적용하고자 할 때 전체가 모인 자리에서 변화를 이야기하기보다는 일대일로 진행하는 것이 공감대를 형성하기도 쉽고 반발이 크지 않다. 전체가 모인 자리에서는 기존에 가지고 있던 타성이 있기 때문에 누군가 한명이 반론을 제기하면 바로 이전으로 원복하려는 경향이 강해진다. 그러면 회의 자리는 새로운 프랙티스에 대한 문제점을 성토하는 자리가 되는 경우가 많다.

새로운 변화를 만들고자 한다면 일대일로 만나 면담을 하면서 진솔하게 대화하는 것이 중요하다는 생각을 한다. 팀장이 꿈꾸는 팀에 대한 비전을 제시하고, 이를 하기 위해 이런 프랙티스를 한번 적용해 봤으면 좋겠다. 짧은 기간 동안 적용해보고 효과가 없으면 폐기하자고 설득하면 대부분의 개발자는 동의하면서 적극적으로 참여하는 경우가 많았다. 그리고 위와 같이 일대일 면담을 할 때 해결책을 팀원이 찾도록 할 경우 새로운 변화에 대한 주도권을 팀장이 아니라 자기 자신이기 때문에 더 적극적으로 참여하게 된다.

우여곡절 끝에 한달이 지났다. 그리고 다시 일대일 면담을 시작했다. 짜 프로그램밍에 대한 문제점에 대한 성토의 자리가 되었다. 면담을 하는 개발자마다 문제점에 대해서만 쏟아냈다. 그래서 물어봤다. "그럼 다음 달부터 짜 프로그램밍 하지 말까?"라고... 그랬더니 "문제점은 많은데 한 달 동안 너무 많은 것을 배웠으며, 그 어느 때보다 공유가 잘 되었다. 그래서 한 달 더 진행해 봤으면 좋겠다."란다. 이후 면담 시간은 현재의 짜 프로그램밍의 문제점을 어떻게 해결할 수 있는지를 같이 고민하는 시간이 되었다. 짜 프로그램밍을 하면서 개발자간의 논쟁도 훨씬 더 많아졌고, 기분이 상해 싸우다가 사무실 밖으로 나가 버리는 경우도 있었다. 하지만 난 아무 말 하지 않고 싸우라고 지켜보기만 했다. 프로젝트 중, 후반부에 싸우기 보다는 초반부터 싸우면서 하나씩 헤쳐 나가는 것이 훨씬 더 좋다고 생각했기 때문이다.

그렇게 좌충우돌 하면서 개발 문화는 서서히 만들어져 갔다. 팀원들도 점점 더 적극적으로 변화해 갔다. 나는 새로운 변화를 만들고 정착시키기 위해 이전보다 훨씬 더 적은 노력을 기울였다. 왜냐하면 현재 문제점을 찾고 해결하기 위해 내가 아닌 팀원들이 더 적극적으로 나섰다기 때문이다. 자신이 변화를 만들 수 있으며, 변화를 만들었으니 인해 개발과 일이 훨씬 더 재미있어 진다는 것을 느꼈기 때문이다. 나는 이 시점부터 팀 보다는 회사 전체로 개발 문화를 확대하고 팀마다 개발 문화를 만드는 것에 도움을 주는데 더 많은 관심을 가질 수 있게 되었다.

이렇게 팀의 문화는 조금씩 조금씩 팀원들에 의해 만들어져 갔다. 새로운 개발자가 오는 경우 팀 문화에 맞춰 일을 진행해 나갔다. 현재 개발하고 있는 서비스에 대한 대략적인 설명을 해주고 바로 짜 프로그램밍부터 시작한다. 짜 프로그램밍을 하면서 팀의 개발 프로세스와 개발 문화에 대해 배워나갔다. 그냥 이 기존 구현이라고 말하기는 것이 아니라 선배들이 멘토 역할을 해주면서 팀에 적용할 수 있도록 했더니 정말 빠른 속도로 팀에 적응해 나갔다. 특히 기존의 타성을 가지고 있는 경력 개발자보다 대학을 졸업한 신입 사원이 더 빠르게 적응하고 성장해 나갔다. 성장하는 개발자의 모습을 볼 때 관리자로서의 즐거움도 같이 느낄 수 있었다.

그렇게 2년의 시간이 지났다. 2년이 지나도 짜 프로그램밍은 유지하고 있었다. 2년 동안 팀원들은 계속해서 개발 문화를 만들어 가고 있었으며, 문제가 발생하면 계속해서 개선해 나갔다. 마지막에는 각 기능별로 예상 일정을 잡고 실질적인 작업 시간을 입력하는 단계까지 다다랐다. 개발자들은 일을 시작하는 시점에 타이머를 시작하고 일을 마치거나 쉬러 갈 때 타이머를 끄고 "시간 * 2"에 해당하는 시간을 시스템에 입력했다. 이는 번다운

차트를 통해 프로젝트의 현재 상태를 파악할 수 있는 기본 데이터로 활용했다. 처음에는 상상할 수 없는 단계에까지 다다르게 된 것이다. 나 혼자만의 욕심으로 진행했다면 이런 단계까지 만들 수도 없었을 것이다. 처음부터 먼 미래의 이상적인 환경에만 집착했다면 아마도 지쳐서 포기했을 것이다. 하지만 나 혼자가 아니라 팀원들이 주도했으며, 그들로 인해 서서히 변화해 갔기 때문에 성공했을 것이다.

개발 문화는 절대로 팀장, 관리자 한 사람의 주도로 만들 수 없다. 그 속에서 살아 숨쉬며 일하는 개개인들이 주도적으로 만들어 갈 때 성공할 수 밖에 없다. 팀장, 관리자는 장기적인 비전만 제시하고 개발자 개개인이 자신의 문제를 자신들이 적극적으로 풀어갈 수 있는 환경만 제시해 주면 된다. 중간, 중간 제대로 가고 있는지 검토만 하고 잘못된 방향으로 가고 있을 때 처음의 비전을 다시금 상기시켜 주는 역할만 하면 된다.

오해가 없기를 바란다. 내가 이 글에 쓴 내용이 가장 이상적이라는 것을 이야기하는 것이 아니다. 반드시 짝 프로그래밍을 해야 한다는 것도 아니고, 애자일 프로세스를 적용해야 한다는 것도 아니다. 모든 팀, 프로젝트마다의 상황은 다르다. 따라서 그에 대한 해결책과 프랙티스도 다르다. 따라서 현재 팀, 프로젝트가 가지고 있는 문제점을 팀원 개개인이 찾도록 하고, 그들이 이에 대한 해결책을 실행할 수 있는 기반 환경을 만들어 준다면 어떠한 방법으로든지 문제는 해결될 수 있다는 것이다. 다른 팀, 회사의 문화와 해결책을 참조할 수는 있겠지만 반드시 우리에게 맞으리라는 생각은 버려라. 그리고 한번 적용했다고 해서 멈추면 안된다. 문제는 계속해서 발생할 것이며, 이에 대한 해결책은 계속해서 찾아갈 때 우리는 진정 우리가 원하는 개발 문화를 만들어갈 수 있을 것이다.