

5장. 팀을 리드하는 방법 [프로그래밍](#)

2021. 12. 10. 17:19

 <https://blog.naver.com/junechol/222592134145>

지금까지 우리는 소프트웨어 개발 팀의 문화와 구성에 대해 다루었습니다. 이번 장에서는 실제 팀이 동작하게 하는 책임을 가지는 사람에 대해 살펴보겠습니다.

어떤 팀도 리더가 없이 기능할 수 없습니다. 특히 구글처럼 개발이 전적으로 팀에 맡겨진 경우 더더욱 그렇습니다. 구글은 두 가지의 리더십 역할을 구분하고 있는데, 매니저는 피플 리더이고 테크 리드 (TL) 는 기술적인 작업을 리드합니다. 이 두가지 역할은 계획하는 능력을 필요로 한다는 점에서 비슷하지만, 인간관계에서 쓰게 되는 기술은 상당히 다릅니다.

선장이 없는 배는 누군가 키를 잡고 엔진을 켜지 않는 한, 조류에 따라 목적없이 떠 다니는 수상 가옥과 다름없습니다. 소프트웨어 역시 배와 마찬가지로 조종하는 사람이 없으면 개발자들끼리 모여 시간을 소진하면서 외부적인 사건만을 기다리게 됩니다. (아니면 필요없는 코드를 괜히 만들고 있을 수 있죠) 이 장은 인력 관리와 기술적 리더십에 대한 내용이지만, 실무 담당자가 읽어도 자기 팀의 리더를 이해하는 데 도움이 될 것입니다.

매니저와 테크 리드

모든 개발팀은 리더가 있지만 리더가 지명되는 방식은 제각각입니다. 특히 구글에서는 경험많은 매니저가 팀을 운영하기 위해 전입해 오기도 하고 개별 기여자 (individual contributor) 가 관리직으로 승진하기도 합니다.

팀 형성 초기에는 매니저와 테크리드 역할을 한사람이 담당하는 것이 보통입니다. 좀 더 큰 팀에서는 경험많은 피플 매니저가 관리 역할을 하고 시니어 개발자가 테크리드 역할을 떠맡게 됩니다. 매니저와 테크리드 모두 개발팀의 성장과 생산성에 있어 중요한 역할을 하지만 두 역할에서 사용되는 인간관계 기술은 상당히 다릅니다.

개발 매니저

많은 회사에서 개발팀을 운영하면서도 소프트웨어 개발에 대해서는 거의 알지 못하는 피플 매니저를 둡니다. 구글은 초기부터 소프트웨어 개발 관리자는 개발 경력을 반드시 가져야 한다고 결정하였습니다. 이 말은 소프트웨어 개발자 출신의 경력직 관리자를 뽑거나 기존의 소프트웨어 개발자를 훈련시켜 매니저로 양성한다는 뜻입니다.

큰 그림으로 보면 개발 관리자는 팀 내 모든 사람 -테크리드도 포함- 의 업적, 생산성, 행복감을 책임지면서 그들이 만드는 제품의 사업 목표가 달성되는 것을 보장해야 합니다. 이 과제는 사업 목표와 개별 팀 멤버의 목표가 항상 일치하는 것은 아니기 때문에 쉽지 않습니다.

테크 리드 (TL)

테크 리드 (TL) - 팀 매니저 휘하인 경우가 많음 - 는 기술적 선택사항 결정, 아키텍처, 우선순위, 개발속도 및 그외 일반 프로젝트 관리등을 포함한 제품의 기술적 측면에 책임을 집니다. TL은 보통 개발 관리자와 협력하여 팀의 인력을 확보하고, 개발자들이 적재적소에서 작업하도록 배치합니다. 많은 TL들이 TL임과 동시에 개별기여자이기도 하므로, 작업이 필요할 때 스스로 작업해서 빨리 끝낼지 시간이 더 걸리더라도 팀원에게 맡길지 결정해야 합니다. 팀의 규모가 커지고 작업량이 많아질수록 팀원에게 맡기는 것이 더 올바른 결정이 되는 경향이 있습니다.

테크 리드 관리자 (TLM)

소규모의 초기 팀에서는 개발 관리자의 기술적 역량이 뛰어날 필요가 있는데 이런 경우에는 TLM 이라는 한 사람이 피플 매니저이면서 기술사항도 책임집니다. TLM은 연륜이 있으면서 최근까지 개별 기여자였던 사람이 주로 맡게 됩니다.

구글에서는 관례적으로 조금만 규모가 있는 팀이라면 두명의 리더 - TL 과 개발 매니저- 가 협력하여 일합니다. 그 이유는 두 가지 일을 동시에 담당하면 쉽게 번아웃에 이르기 때문이고, 두 명이 각자의 전문화된 역할에 집중하도록 하는 것이 더 낫다고 보기 때문입니다.

TLM의 업무는 본인의 실무와 업무의 배분, 피플 매니징 사이의 균형을 잡는 법을 익혀야 하기 때문에 까다롭습니다. 그래서 TLM들은 경험이 많은 다른 TLM의 멘토링이 많이 필요합니다.

케이스 스터디 : 인사권한 없이 영향력 미치기

당신의 보고라인 아래에 있는 사람에게 당신이 원하는 일을 시키기는 쉽습니다. 그러나 당신 조직 밖에 있는 사람에게, 심지어 당신의 업계 밖의 사람이 당신이 원하는 일을 하게 만드는 것은 매우 어렵습니다. 이러한 "인사권 없이 영향력 미치기" 는 가장 강력하게 개발된 리더십의 양태입니다.

예를 들어 구글의 시니어 엔지니어링 펠로우인 제프 딘의 경우, 구글 엔지니어링 팀의 아주 일부만을 이끌었지만, 그의 기술적 결정과 지침은 전체 개발 조직에 미쳤고 (그의 기고문과 사외 강연을 통해서) 사외에도 영향을 주었습니다.

다른 사례로 내가 간여한 "데이터 해방 전선" 이라고 부르는 팀이 있습니다. 6명 미만의 인력으로 50개가 넘는 구글 제품들의 데이터 내보내기를 지원하는 **Google Takeout** 이 바로 그것입니다. 당시에 **Google Takeout** 으로 데이터 내보내기를 지원하라는 구글 경영진의 공식적 의사결정은 없었음에도 어떻게 수백명의 개발자들이 이 프로젝트에 기여하게 되었을까요? 회사의 전략적인 요구가 무엇인지 명확히 한 후, 이 작업이 회사의 목표 및 우선순위와 밀접히 관련이 있다는 것을 보여주었기 때문입니다. 그리고 여러 개발팀들이 쉽고 빠르게 **Takeout** 과 연동할 수 있도록 하는 도구를 제공해 주었기 때문입니다.

개별 기여자 역할에서 리더 역할로 이동하기

공식적 임명 여부와 무관하게 제품이 굴러가기 위해서는 누군가는 운전석에 앉아야 합니다. 만약 당신이 의욕이 충만한데다가 기다리지 않는 타입이라면 그 누군가는 바로 당신일 것입니다. 당신은 팀내 의견충돌을 해결하고 의사결정을 내리고 인력을 조율하는 작업에 당신의 시간과 노력이 갈려 들어가고 있다는 것을 발견하게 될 것입니다. 이런 일은 항상 우발적으로 일어납니다. 당신은 리더가 되기를 원하지 않았을 것이지만 그런 일이 벌어지게 마련입니다. 사람들은 이런 고통을 "매니저증후군" 이라고 부르기도 합니다. 당신 스스로 절대로 매니저는 되지 않겠다고 결심했다 하더라도 경력의 어느 시점에 다다르면, 특히 당신이 일을 꽤 잘 하는 편이라면, 매니저의 위치에 서 있는 자신을 발견하게 될 것입니다. 이 장의 나머지 부분은 당신이 그런 경우에 무엇을 해야 하는지에 대한 내용입니다.

우리는 당신에게 매니저가 되라고 할 생각이 없습니다. 단지 최고의 리더들은 겸손, 존중, 신뢰라는 원칙을 갖고 팀에 봉사한다는 것을 말해주고 싶을 뿐입니다. 당신의 업무 방향을 잡는 데 있어 리더십의 이모저모를 이해하는 것은 중요합니다. 그런데 당신이 프로젝트라는 배를 타고 실려가는 게 아니라 스스로 조종하고 싶다면 항해술을 배워야 합니다. 그렇지 않으면 그 배는 암초에 올라 앉습니다.

두려워해야 할 단 한 가지는... 모든 것

대부분의 사람이 "매니저" 라는 말을 들으면 느끼는 불편함은 차치해 두고도, 사람들이 매니저가 되고 싶어하지 않는 이유는 여러가지가 있습니다. 소프트웨어 개발 업계에서 가장 많이 듣게 되는 것은 매니저가 되면 코딩에 쓸 시간이 줄어든다는 것입니다. 이것은 TL이 되더라도 개발 매니저가 되더라도 사실입니다. 이에 대해서는 나중에 다시 언급하겠습니다. 우선은 많은 사람들이 관리자가 되길 꺼려하는 다른 이유들을 살펴 보겠습니다.

당신 경력의 대부분을 코드를 작성하며 보냈다면, 당신은 매일 일과종료시에 "오늘 이만큼 했어" 라고 할 무엇인가가 있을 것입니다. (코드일 수도 있고, 디자인 문서일 수도 있고, 해결한 버그목록일 수도 있습니다) 그러나 관리업무로 바쁜 하루를 보내고 난 후라면 당신은 "오늘 아무 것도 한 것이 없네" 라고 생각할 것입니다. 마치 몇해동안 당신이 매일 사과를 몇개 따는지 확인할 수 있는 일을 하다가, 어느날 바나나를 기르는 일로 직업을 바꾼 것과 유사한 상황입니다. 그러면 매일 저녁에 당신은 조금씩이지만 잘 자라고 있는 바나나 나무는 쳐다보지 않고 "오늘 사과를 하나도 못 따어" 라고 자책하게 될 것입니다. 매니지먼트 업무를 계량화하는 것은 오늘 작업량 몇 개라고 세는 것보다 훨씬 어려운 일입니다. 당신이 관리하는 팀이 얼마나 행복하고 생산적으로 일하게 되었는가 당신 업적의 척도입니다. 바나나를 기르는 사람은 사과 갯수를 세는 함정에 빠지지 않아야 합니다.

자주 언급되지 않는 관리자가 되기 싫어하는 또다른 이유는 "피터의 법칙" - 조직내에서 모든 사람은 경쟁력이 없을수록 승진한다 - 에 기반합니다. 구글이 이를 피하는 방법은 일정기간 현재 레벨의 상위레벨의 능력을 보여준 사람만을 - 즉 고과시에 exceed expection 등급을 받은 사람만을 - 승진 대상으로 한다는 것입니다. 많은 사람들이 업무 수행 능력이 없거나 인사관리 능력이 없는 관리자를 만난 경험을 갖고 있습니다. 심지어 어떤 사람들은 나쁜 관리자만 만났을 것입니다. 만약 당신이 평생 엉터리 관리자만 만나 봤다면, 당신 스스로 관리자가 되고 싶어할 이유가 무엇일까요? 당신이 할 수 없을 것 같은 역할로 승진하는 것을 원해야 할 이유가 무엇일까요?

그러나 믿어 주십시오. 당신이 TL 이나 관리자가 되어야 할 이유가 있습니다. 첫번째로 당신 능력치의 스케일을 바꿔줄 수 있습니다. 당신이 아무리 코딩을 잘 한다고 하더라도 당신이 작성할 수 있는 양의 한계는 존재합니다. 당신이 리더가 되어 훌륭한 개발자들에게 코딩을 시킨다면 얼마나 많은 코드를 작성할 수 있을지 상상해 보십시오. 두번째는 당신이 정말로 재능이 있을 수 있다는 것입니다. 프로젝트의 리더십이라는 진공청소기에 빨려들어간 사람들 중 몇몇은 자기가 가이드하고 도와주고 옹호하는 데 매우 재능이 있다는 것을 깨닫는 경우가 있습니다. 누군가는 리더가 되어야 하는데 왜 당신이 하면 안 되나요?

하인의 리더십

매니저들이 걸리는 병이 있는 것 같습니다. 그들은 과거에 자기 매니저들이 행한 나쁜 행동들의 기억을 잊어버린 후, 보고라인의 사람들을 "관리하기 위해" 과거의 매니저들과 동일한 행동을 합니다. 이 질병의 증상에는 마이크로 매니징, 저성과자 무시하기, 무능력자 기용하기 등이 있습니다. 즉각적으로 치료하지 않으면 이 질병은 팀 전체를 죽입니다. 내가 구글에서 처음 매니저가 되었을 때 들었던 최고의 조언은 당시 개발 임원이었던 스티브 빈터에게서 들은 것입니다. 그는 이렇게 말했습니다. "무엇보다도, 관리하고 싶은 충동을 억누르세요". 새로 매니저가 된 사람들이 갖게 되는 가장 큰 충동은 팀원들을 적극적으로 "관리하고자" 하는 마음입니다. 왜냐하면 그게 매니저가 하는 일이니까요. 그렇지 않은가요? 그러나 그렇게 하면 재앙적 결과가 따르게 됩니다.

이런 "관리병"의 치료제가 바로 "하인의 리더십"을 가지는 것입니다. 이를 다르게 설명하자면 리더로서 당신이 할 수 있는 가장 중요한 일은 가족의 건강과 안녕을 돌보는 집사역할과 유사한 일이라는 말입니다. 하인 리더로서 당신은 겸손, 존중, 신뢰의 분위기를 조성해야 합니다. 이는 팀원들이 스스로 제거할 수 없는 관료적 분위기를 걷어내는 것을 뜻합니다. 팀이 공통의 문제의식을 갖게 하는 것을 뜻하고, 팀이 저녁늦게 일할 때 식사를 가져다 나르는 것을 뜻합니다. 하인 리더는 팀이 일을 부드럽게 진행할 수 있도록 진행경로의 갈라진 틈새를 메워주고 팀이 조언을 필요로 할 때는 조언을 해 주고 직접 나서서 일하는 것을 두려워하지 않습니다. 하인 매니저가 "관리하는" 유일한 경우는 팀의 기술적, 사회적 건강을 관리할 때입니다. 팀의 기술적인 건강에만 초점을 두고싶은 마음이 들 수 있지만, 팀의 사회적 건강도 동일하게 중요합니다. (그런데 사회적 건강의 관리가 훨씬 어려울 때가 많습니다)

개발 매니저

현대의 소프트웨어 회사에서 관리자에게 실제로 요구하는 것은 무엇일까요? 컴퓨터의 시대가 오기전에는 "관리"와 "노동"은 상반되는 역할이었습니다. 매니저는 권력을 가지고 휘두르고 노동자는 기업목표를 달성하는 작업들을 수행했습니다. 그런데 현대의 소프트웨어 회사에서는 그렇지 않습니다.

매니저란 말이 욕설처럼 들리는 이유

개발 매니저의 책임에 대해 이야기하기 전에 관리자의 역사를 검토해 봅시다.



관리자라는 개념은 군대의 조직 구조로부터 시작되었으며 백년 전 산업 혁명기에 기업에 도입되었습니다. 여기저기에 공장이 만들어 졌으며, 비숙련 노동자들이 기계를 돌려야 했습니다. 이런 노동자들을 감시하는 사람으로 관리자가 필요해 졌습니다. 이 때의 근로자들은 쉽게 교체 가능했기 때문에 관리자는 직원을 잘 대우하거나 근무조건을 개선할 이유가 없었습니다. 인간적이라고 할 수는 없지만 이런 관리법은 직원들이 기계적인 과업을 수행해야 하는 시절에는 잘 작동했습니다. 관리자는 수레꾼이 노새를 다루는 것과 같은 방식으로 직원을 다루었습니다: 당근을 주며 움직이게 하고, 명령대로 움직이지 않으면 채찍으로 때리는 방식입니다. 이런 "당근과 채찍" 관리법은 공장조직이 현대의 사무조직으로 바뀔 때도 살아 남았고 무자비한 수레꾼같은 전형적 관리자상도 20세기 중반까지 번성하였습니다.

이것이 현대에도 몇몇 산업에서 지속되고 있으며, 심지어 창의적 사고와 문제해결 능력을 요구하는 산업에서도 적용되고 있습니다. 시대착오적인 "당근과 채찍" 방식이 창의적 인력들에게 효과적이지 않고 생산성을 떨어뜨린다는 수많은 연구에도 불구하고 말입니다.

과거의 조립라인 노동자들은 며칠이면 필요한 훈련을 받아서 라인에 대체 투입될 수 있었습니다. 그러나 거대한 코드베이스에서 작업을 하는 소프트웨어 개발자가 새로운 팀에 적응하여 개발속도를 내려면 수개월이 소요됩니다. 교체 가능한 조립라인 노동자와는 달리, 이 사람들에게는 적절한 지원이 제공되어야 하고 시간적 공간적 여유를 갖고 생각하고 창조하도록 해 주어야 합니다.

현대의 개발 매니저

많은 사람들이 아직도 시대착오적으로 "관리자" 라는 호칭을 사용합니다. 이 호칭은 종종 새로 관리자가 된 사람들이 직원들을 "관리하도록" 합니다. 관리자는 부모처럼 행동하고 그에 따라 직원들은 어린이처럼 행동하게 되기도 합니다. 이런 현상을 겸손, 존중, 신뢰의 틀로 바꿔야 합니다. 만약 관리자가 직원을 신뢰한다는 것을 표명한다면 직원들은 그 신뢰에 부합하는 행동을 하려고 노력할 것입니다. 간단합니다. 좋은 매니저는 팀원들의 안전과 복지를 지키면서 필요한 일이 완수되게 합니다. 이 장에서 당신이 딱 한가지만 기억해야 한다면 다음을 기억하십시오: **전통적인 관리자는 어떻게 일을 되게 할까를 걱정했다면, 훌륭한 관리자는 무슨 일을 할 것인가를 걱정합니다** (어떻게 해낼 것인가에 대해서는 팀원들을 신뢰하고 맡깁니다)

몇 년 전 우리 팀에 제리라는 개발자가 합류했습니다. 제리의 전임 관리자(다른 회사입니다)는 매일 9시부터 5시까지 그가 책상에 있어야 하고, 그가 거기에 없었던 경우, 충분히 일하지 않았다고 단정했습니다. (네, 매우 우스운 생각입니다) 제리가 저와 일한 첫날 오후 4시 40분에 제리가 저에게 왔습니다. 시간 조정이 어려운 선약이 있는데 조금 일찍 퇴근해야 하겠다고 미안한 듯이 말했습니다. 저는 그를 보면서 미소 지으며 "당신이 일을 끝내는 한, 전 당신이 언제 퇴근하는지 신경쓰지 않습니다" 라고 말했습니다. 제리는 몇초간 멍하니 저를 보더니 이윽고 고개를 끄덕이고 갔습니다. 저는 제리를 어른으로 대했습니다. 그는 항상 자신의 일을 완수했고, 저는 그가 책상에 앉아있는지 신경 쓸 필요가 없었습니다. 왜냐하면 그가 일을 할 때 베이비시터가 필요하지 않기 때문입니다. 만약 직원이 너무나 일에 무관심하여 전통적인 매니저 방식으로 일일이 지시해야 일이 진행된다면 그것이야말로 심각한 문제입니다.

실패도 하나의 선택지다

팀 활동을 촉매하는 또 다른 방법은 그들에게 심리적 안정감을 제공하여 편하게 리스크를 감수하도록 하는 것입니다. 즉 당신의 팀원들이 당신이나 다른 팀원들로부터의 부정적인 반응을 걱정하지 않고 본연의 태도를 보일 수 있게 하는 것입니다. 리스크는 매혹적인 것입니다. 대부분의 리스크 산정은 부정확하고, 대부분의 회사는 어떻게든 리스크를 피하려고 노력합니다. 그 결과로 보수적으로 일하고 작은 성공만을 추구하게 됩니다. 좀 더 큰 리스크를 감수하면 엄청난 성공을 거둘 수 있는 상황에서도 말입니다. 구글에는 이런 속담이 통용됩니다. "불가능하다고 생각한 목표를 추구할 때가 달성 가능하다고 생각했던 목표를 추구할 때보다 더 큰 성과를 거둘 가능성이 크다". 리스크를 감수하는 문화를 구축하는 좋은 방법은 실패해도 좋다는 것을 팀이 알게 하는 것입니다.

분명히 말해 봅시다 : 실패해도 좋습니다. 사실, 우리는 실패를 무언가를 엄청나게 빨리 배우는 방법으로 생각하고 있습니다. (물론 같은 실패를 반복하는 경우는 문제입니다) 그리고 실패는 배움의 기회로 봐야지 누군가를 비난할 계기로 생각하면 안됩니다. 실패 시에 잃는 것이 적도록 빨리 실패하는 것이 좋습니다. 느린 실패는 귀한 교훈을 얻을 수는 있겠지만 잃는 것이 많아져서 고통이 큼니다. 고객에게 영향이 미치는 방식의 실패는 아마도 우리가 원치 않는 실패일 것입니다. 그러나 이런 실패에서 배울 수 있는 요소가 더 많긴 합니다.

앞에서 언급했듯이 구글은 주요 생산 실패가 있을 때마다 포스트모템을 수행합니다. 이것은 실패의 원인이 된 이벤트들을 문서화하여 미래에 재발하는 것을 막는 절차를 개발하는 방법입니다. 누군가를 비난하는 행사도 아니며, 불필요하게 관료적인 점검절차를 도입하려는 것도 아닙니다. 포스트모템의 목표는 문제의 핵심에 파고들어 한 번에 재발 불가하게 고치는 것입니다. 매우 어렵지만, 충분히 효과적입니다. (게다가 카타르시스도 있죠)

개인의 성공과 실패는 약간 다릅니다. 개인적인 성공에 박수치는 것은 무방하지만, 실패한 개인에게 비난의 화살이 가게 하는 것은 팀을 분열시키고 도전 의욕을 떨어뜨리는 확실한 방법입니다. 실패하는 것은 좋습니다만 팀으로 실패해야 실패로부터 배울 수 있습니다. 개인이 성공하면 공개적으로 칭찬하십시오. 그러나 개인이 실패하면 건설적인 비판이라도 비공개로 전하십시오. 어떤 경우에도 겸손, 존중 및 신뢰의 원칙하에서 팀이 실패로부터 배울 수 있도록 하십시오.

안티패턴

우리가 성공적인 TL 과 개발 매니저의 "디자인 패턴"을 나열하기 전에, 성공적인 매니저라면 따르지 말아야 할 패턴들을 짚어 보겠습니다. 우리가 겪어본 나쁜 관리자들과 몇몇 경우 우리 스스로에게서 이런 파괴적인 패턴들을 목격했습니다.

안티패턴 : 무능력자 고용하기

당신이 관리자인데 당신의 직역이 불안하다고 느낄 때, 당신의 권위에 도전하지 못하게 하는 방법은 당신이 맘대로 할 수 있는 사람을 고용하는 것입니다. 당신만큼 영리하거나 야심차지 않은 사람을, 아니면 당신보다 더 불안한 사람을 뽑으면 됩니다. 이렇게 하면 당신의 팀 리더나 의사 결정자로서의 지위는 공고해 질 수 있지만, 대신에 당신의 일이 늘어나게 됩니다. 당신의 팀은 목줄을 맨 개처럼 당신이 없이는 움직이지 않게 됩니다. 무능력자로 팀을 구성하면 당신은 휴가도 못 갑니다. 당신이 방을 떠나는 순간 생산성이 제로가 되어 버리니까요. 뭐 그래도 직업적 불안이 없으니 괜찮다고 생각하시겠죠?

그것보다는, 당신보다 영리하고 당신을 대체할 수 있을만한 사람을 고용하려 노력하십시오. 그 사람이 당신의견에 반대할 수 있기 때문에 힘들 수 있습니다. (당신의 잘못을 지적할 수도 있죠) 그러나 이런 사람들이 당신을 감동시

키고 놀라운 일을 해 낼 수 있습니다. 그들은 스스로 무엇을 해야 할지 탐구하고 직접 팀을 이끌려고도 합니다. 이런 행동을 당신의 권위에 도전하는 것으로 생각할 필요가 없습니다. 그것보다는 다른 팀을 하나 더 리드할 수 있겠구나 라고 생각할 수 있고, 새로운 기회를 모색할 수도 있고, 아니면 업무 진행을 매일 체크해야 한다는 부담없이 휴가를 떠날 수도 있습니다. 또한 배움과 성장의 기회로 삼을 수 있습니다. 주변에 당신보다 뛰어난 사람이 많이 있으면 당신의 경험을 확장하는 것이 쉬워집니다.

안티패턴 : 저성과자를 못 본 척하기

관리자가 된 지 얼마되지 않았을 때, 내 팀에게 보너스 편지를 나눠 줄 기회가 왔습니다. 그 때 저는 제 관리자에게 "관리자가 되는 건 즐겁군요!" 라고 말했습니다. 그러자 그는 즉시 "이빨요정 (뽀은 이빨을 침대맡에 두고 자면 돈을 놓고 이빨을 가져간다)이 될 때가 있으면, 치과의사가 되어야 할 때도 있는 법이에요" 라고 말했습니다. 남의 이를 뽀는 것은 절대로 즐겁지 않습니다. 우리는 팀 리더가 아주 훌륭한 팀의 기반을 구축해 놓고도 한두명의 저성과자로 인해 최고의 팀이 되지 못하는 (그래서 팀이 와해되는) 경우를 많이 보았습니다. 소프트웨어 개발 업무에서 가장 도전적인 분야가 인사관리이고, 인사관리에서 가장 어려운 부분이 기대를 충족하지 못하는 인력을 다루는 것입니다. 충분한 시간을 들이지 않거나 열심히 일하지 않아서 목표를 달성하지 못하는 경우도 있지만, 아무리 열심히 일해도 능력 자체가 부족한 사람의 경우 대처하기가 매우 어렵습니다.

구글의 사이트 신뢰성 엔지니어링 (SRE) 팀의 모토는 "꿈꾸기는 전략이 아니다" 입니다. 그런데 저성과자를 다룰 때는 꿈꾸기가 전략처럼 사용되는 경우가 많습니다. 대부분의 팀 리더는 정면으로 문제를 응시하기를 피하면서 저성과자가 마술처럼 일을 잘하게 되거나 아니면 그냥 없어져 주기를 바랍니다. 그러나 그렇게 되는 경우는 거의 없습니다. 리더는 꿈과 달리 저성과자의 능력이 개선되지 않는 상황에서, 일을 잘하는 팀원이 저성과자를 같이 끌고 가기 위해 귀중한 시간을 쓰고 있습니다. 당신이 팀내에 저성과자가 없는 척 하더라도 팀원들은 저성과자가 누구인지 분명히 알고 있습니다. 자기들이 공을 들여 그를 끌고가는 중이니깐요.

저성과자를 못 본 체하는 것은 유능한 신규 팀원이 들어오는 것을 막을 뿐 아니라 기존의 유능한 팀원이 떠나게 합니다. 저성과자들은 달리 갈 곳이 없기 때문에 결국 당신의 팀은 저성과자로 채워지게 됩니다. 마지막으로, 저성과자들을 데리고 있는 것이 그들에게도 도움이 되지 않습니다. 당신의 팀에서 일을 잘하지 못하는 사람이 다른 곳에서는 일을 곧잘 할 수도 있습니다. 가능한 한 저성과자를 빨리 처리하는 것이 오히려 그들을 돕는 것일 수 있습니다. 저성과자를 빨리 맞닥뜨려 처리하다 보면 그들이 단지 생산적인 단계로 나아갈 용기와 지도편달이 필요했던 것임을 깨닫게 되기도 합니다. 저성과자 처리에 있어 너무 녹장을 부리면 다른 팀원들과의 관계가 나빠져서 도움을 주기 힘들어질 수도 있습니다.

저성과자를 어떻게 효과적으로 코치할 수 있을까요? 가장 좋은 비유는 당신이 절뚝이는 사람이 다시 걷게 하며, 이 욱고 팀과 보조를 맞춰 달릴 수 있도록 돕는 중이라고 상상해 보는 것입니다. 일시적으로 마이크로매니징이 필요하지만 여전히 겸손, 존중 및 신뢰의 원칙은 잊지 마십시오. 특히 존중하는 태도를 가져야 합니다. 구체적인 기간을 정하여 (예를 들어 2개월 정도) 그 기간에 달성하기를 기대하는 구체적인 목표를 제시하십시오. 매주 그와 만나 진행 상황을 확인하십시오. 각각의 마일스톤마다 판별이 용이한 기대치를 제시하여 성공과 실패 여부를 쉽게 판단할 수 있게 해 두십시오. 그렇게 하면 만약 저성과자가 계획을 제대로 따라오지 못하는 경우 빠르게 그 사실을 알 수 있을 것입니다. 그 인력이 계획대로 잘 안되고 있다는 것을 스스로 알게 되면 그만둬야겠다고 결심하기도 용이합니다. 또는 "제대로 해보자"는 결의에 차서 열심히 하게 될 수도 있습니다. 어느 쪽이든 당신은 저성과자와 직접 대면하면서 중요하고 필요한 변화를 이끌어내고 있는 것입니다.

안티패턴 : 인간적인 문제 무시하기

관리자는 사회적인 면과 기술적인 면이라는 두가지 관심영역을 가집니다. 구글의 관리자는 기술적인 면이 좀더 강한 경향이 있습니다. 그들은 주로 기술직종에서 승진하여 관리자가 된 경우가 대부분이기 때문에 인간적인 문제를 무시하는 경우가 있습니다. 개별 기여자로서 당신은 대부분의 시간을 기술적 문제를 해결하면서 지내왔기 때문에 관리자가 된 지금도 당신의 모든 에너지를 기술적인 측면에 쏟고 싶은 마음이 드는 것이죠. 학생일 때도 당신이 배운 것은 업무와 관련된 기술적인 것들 뿐이었습니다. 그러나 관리자가 된 지금, 인간적인 요소를 고려하지 않으면 팀이 위태로워질 수 있습니다.

리더가 팀의 인간적인 요소를 무시하는 사례로 시작해 보겠습니다. 몇 년 전에 제이크는 첫 아이를 가졌습니다. 케이트는 제이크와 몇년간 같이 일하는 동료였는데 사무실 근무를 하기도 하고 원격근무를 하기도 했습니다. 아이가 태어난 직후 몇주간 제이크는 재택근무를 하게 되었는데 그들에겐 문제가 전혀 없었습니다. 기존에 제이크와 케이트

티가 원격으로 업무를 해 왔기 때문입니다. 그들이 충분히 생산적인 작업을 하고 있는 와중에 그들의 관리자인 파블로가 제이크가 몇주간 출근을 하지 않고 재택근무중이란 것을 알게 되었습니다. 제이크는 생산적으로 일하고 있었고, 케이트도 제이크와 일하는 것에 아무런 문제가 없었지만 파블로는 제이크가 출근하여 근무하지 않는다는 것을 언짢아 했습니다. 제이크는 파블로에게 자신이 출근근무만큼 생산적으로 일하고 있으며, 몇주간은 집에서 근무하는 것이 그와 그의 부인에게 필요한 상황임을 설명하였습니다. 파블로는 이렇게 말했습니다. "이봐요, 사람들은 다 아이가 있어요. 출근하세요." 말할 필요도 없이 제이크는 화가 났고 파블로에 대한 존경심을 잃었습니다.

파블로가 이 상황을 다르게 처리할 수 있었을 방법은 많습니다. 제이크가 집에서 아이와 아내와 더 많은 시간을 같이 있으려 한다는 것을 이해하고 생산성과 팀웍이 유지되는 한에서 그가 집에서 일하는 것을 일정정도 허용할 수도 있었습니다. 아니면 집안이 어느정도 정리되기 전까지는 제이크가 일주일에 하루 또는 이틀 정도 출근하는 방식으로 협의할 수도 있었습니다. 결과가 어떠하든 간에 파블로가 상대에 대한 공감을 보였다면 제이크를 좀더 행복한 상태로 유지할 수 있었을 것입니다.

안티패턴 : 모든 사람의 친구가 되기

대부분의 사람이 처음 리더가 될 때는 자기가 기존에 일하던 팀에서 관리자나 TL이 되는 것이 보통입니다. 많은 리더들이 그 팀에서 지금까지 가꾸어 온 친분을 유지하려 합니다. 그래서 리더가 된 후에도 기존 팀원과의 우정을 유지하려 노력합니다. 하지만 이것이 우정을 파괴하는 재앙의 씨앗이 될 수 있습니다. 우정과 "부드러운 리딩"은 다른 것이므로 혼동하면 안 됩니다. 당신이 다른 사람의 경력에 영향을 미치는 힘을 가지고 있으면, 그들은 억지로라도 "우정"과 비슷한 모습을 보이려 하게 마련입니다.

팀원과 긴밀한 친구로 지내지 않아도 팀의 공통목표를 설정하고 리드할 수 있다는 것을 기억하십시오. 반대로 기존의 우정을 버리지 않으면서도 아주 까칠한 리더가 될 수도 있습니다. 팀원들과 점심을 같이 먹는 것이 그들을 불편하게 하지 않으면서도 사회적 관계를 유지하는 좋은 방법입니다. 이 때가 평소의 업무 환경에서 벗어나 격식없는 대화를 할 수 있는 기회가 됩니다.

때로는 좋은 친구이고 동료였던 사람에 대해 관리자로서 대하는 것이 어려울 수 있습니다. 지금 관리대상인 친구가 열심히 일하는 타입이 아니라면 서로에게 스트레스가 될 수 있습니다. 가능한 한 그런 상황에 처하는 것을 피하는 것이 좋습니다만 피할 수 없다면 이런 사람들과의 인간관계는 매우 유의하여야 합니다.

안티패턴 : 채용기준을 낮추기

스티브 잡스는 이렇게 말한 적이 있습니다. "1류는 1류를 채용한다, 2류는 3류를 채용한다." 당신이 빨리 채용을 해야 되는 경우에 이 속담의 함정에 빠지기가 더 쉽습니다. 구글이 아닌 다른 회사에서 제가 본 사례는 어떤 팀이 5명의 사람을 뽑아야 하는 경우 40내지 50명을 면접보고 그중 가장 우수한 5명을 대상으로 선정하였습니다. 그들이 채용기준을 통과하는지는 검토되지 않았습니다. 이런 채용법은 그저그런 팀을 만드는 가장 빠른 방법입니다.

적합한 사람을 고르는 비용은 - 업체를 통하거나, 광고를 하거나, 찾아다니며 추천을 받거나 - 처음부터 뽑지 말았어야 하는 사람을 뽑고나서 드는 비용보다 작습니다. 그 비용에는 팀 생산성 저하, 팀 스트레스, 해당 직원의 관리에 든 시간, 해고하느라 든 노동이 포함됩니다. 물론 당신이 그런 사람을 팀에 가만히 놔두는 우를 범하지 않았다고 가정하였습니다. 만약에 당신이 채용과정에 대해 자랑할 게 없거나 팀원 채용과 관련해 행복하지 않다면, 당신은 무슨 수를 써서라도 더 수준높은 개발자를 뽑기위해 분투해야 합니다. 위대한 팀을 위해 필요한 재료가 없으면 당신은 망한 겁니다.

안티패턴 : 팀을 어린애처럼 다루기

당신이 팀을 믿지 않는다는 것을 보여주는 가장 좋은 방법은 그들을 아이처럼 다루는 것입니다. 사람들은 자기가 대접받는대로 행동하는 경향이 있습니다. 그러므로 팀원을 어린이나 죄수처럼 다루면서 그들이 실제로 그렇게 행동한다며 놀라지 마십시오. 그런 아이취급을 보여주려면 그들을 마이크로 매니징하거나 그들의 능력을 불신하면서 그들이 스스로의 일에 책임질 기회를 주지 마십시오. 만약 당신이 사람들을 믿지 못하고 계속 마이크로 매니징해야 하는 상황이라면 채용을 잘못된 것입니다. 팀원을 뽑아서 평생 마이크로매니징하는 게 목표가 아니었다면 그것은 실패입니다. 신뢰할 만한 사람들을 뽑고 나서 그들에게 당신의 신뢰를 보여준다면 그들은 스스로 난관에 대처할 것입니다. (앞에서 말한 바대로 당신은 훌륭한 사람들을 뽑아 놔야 합니다)

이런 수준의 신뢰의 결과는 사무용품과 같은 일상적 사례에서도 드러납니다. 하나의 예로 구글은 직원들에게 온갖 문방구(펜, 노트 등의 사무용품)로 차 있는 캐비닛을 제공하고 직원들은 아무 제약없이 그 물건들을 가져갈 수 있습니다. IT 부서에서는 "테크스탑"이라는 곳에 작은 전자용품점을 만들어 자유롭게 사용가능하도록 했습니다. 여기에는 전원, 케이블, 마우스, USB 드라이브 같은 악세사리들이 있습니다. 이것들도 아무런 제약없이 집어갈 수 있지만, 구글 직원들은 이 비품의 사용에 대해 신뢰를 받고 있다는 점에서 오히려 올바른 행동을 하려는 의지가 발동됩니다. 많은 다른 회사에서 온 사람들이 이런 정책을 듣고는 놀라면서 비품을 훔쳐가는 직원들 때문에 구글이 잃어버리는 돈이 많을 것이라 말합니다. 물론 그럴 수도 있습니다만, 자사의 직원을 아이처럼 행동하게 하고 비품 사용 신청서를 작성하게 하는 비용은 어쩔 것입니까? 그런 비용이 펜과 USB 케이블의 가격보다 훨씬 큼니다.

좋은 패턴들

안티패턴을 살펴봤으니 이제 성공적인 리더십과 관리를 위한 긍정적 패턴들을 살펴 보겠습니다. 특히 우리들이 구글에서 성공적인 리더들을 관찰하여 발견했거나 우리의 멘토로부터 배운 내용입니다. 이러한 패턴들은 업무 성공을 가져다 주었을 뿐 아니라 우리가 우리가 따르는 리더들을 존경하게 만든 것들입니다.

자존심을 버리라

앞에서 겸손, 존중, 신뢰에 대해 이야기하면서 "자존심 버리기"에 대해 이야기한 바 있는데, 특히 팀 리더에게는 중요한 것입니다. 이 패턴에 대해서 오해를 하여 자신을 발칵개로 만들어 다른 사람이 밟고 지나가게 하라는 말로 받아들이는 경우가 종종 있습니다. 겸손함을 갖는 것과 타인이 당신을 맘대로 이용하게 하는 것은 명확한 차이가 있습니다. 겸손은 자기 확신이 낮은 것과는 다릅니다. 당신은 자기중심적이지 않으면서도 자기 의견과 확신을 가질 수 있습니다. 개인의 자존심이 크면 팀이 감당하기 힘들어지는데 그 개인이 리더라면 더욱 힘들 것입니다. 그러니 개인의 자존심 대신에 팀의 자존심과 정체성을 구축하는 노력을 하는 것이 좋습니다.

자존심 버리기의 다른 측면은 신뢰입니다. 당신은 팀을 신뢰해야 합니다. 그 의미는 팀원들의 능력과 지난 업적을 존중한다는 것입니다. 새로 들어온 팀원에게도 동일하게 적용됩니다.

만약 당신이 마이크로매니징을 하고 있지 않다면 현업에서 일하는 사람들이 당신보다는 자기 일에 대해 더 잘 이해하는 게 분명합니다. 그러므로 당신이 팀의 공감대를 구축하고 방향을 잡는 역할을 하지만, 구체적으로 어떻게 당신이 정한 목표에 도달할 것인가 하는 세부사항은 실무를 담당한 사람들에게 맡겨두는 게 더 바람직합니다. 그렇게 하면 그들이 주인의식을 가지고 제품의 성공이나 실패에 대한 책임감을 갖게 될 것입니다. 좋은 팀에게 스스로 품질과 업무속도를 정하게 한다면 당신이 당근과 채찍으로 관리하는 것보다 더 나은 결과를 얻을 것입니다.

새로 리더가 된 사람들은 모든 것을 바로 잡고, 모든 것을 이해하고, 모든 것에 답을 가지고자 하는 강한 책임감을 느낍니다. 우리가 확인할 수 있는 것은 당신은 모든 것을 바로 잡지 못하고, 모든 것에 답을 가지지도 못할 것이라는 것입니다. 그리고 당신이 그런 척하는 태도를 갖고 있으면 팀원의 존경을 잃기 쉽습니다. 이런 태도는 당신의 역할에 대한 안정감을 얻기 위한 노력이라고 볼 수 있습니다. 당신이 개별 기여자였던 경우를 생각해 보면, 불안감을 별로 느끼지 않았을 것입니다. 이를 극복하기 위해 질문을 환영하도록 하십시오. 어떤 사람이 당신의 결정에 질문을 갖는다면 그 사람은 당신의 생각을 더 잘 이해하고 싶어서 묻는 것입니다. 질문을 권장하면 당신은 더 많은 건설적인 비판을 받아들이게 될 것이고, 더 좋은 리더가 되고, 더 좋은 팀을 만들 수 있습니다. 당신에게 건설적인 비판을 하는 사람을 찾는 것은 매우 어렵고, 그런 사람이 결국 당신을 위해 일하도록 만드는 것은 더더욱 어렵습니다. 팀으로서 달성하고 싶은 큰 그림을 기억하고 비판적 의견을 환영하십시오. 방어적인 본능을 피해야 합니다.

자존심 버리기의 마지막 부분은 간단한 것인데도 많은 개발자는 이를 해내지 못합니다. 실수를 하면 사과하십시오. 단지 "미안합니다" 라고 말하기만 할 게 아니라 진지하게 사과하는 마음을 가지십시오. 당신은 반드시 실수를 하게 되어 있고, 당신이 인정하든 하지 않든 팀원들은 당신의 실수를 알게 되어 있습니다. 당신에게 말을 하지 않더라도 자기들끼리는 당신의 실수에 대해 이야기하게 되어 있습니다. 사과에는 돈이 들지 않습니다. 사람들은 일을 망친 것에 대해 사과하는 리더를 존경합니다. 그리고 흔히 드는 생각과는 달리 사과하는 것이 당신을 취약하게 만들지도 않습니다. 오히려 사과하는 행위는 당신이 이성적으로 상황을 정확히 관찰하고 판단한다는 것을 보여주는 것이므로 존경받을 만한 행동입니다. 그리고 겸손, 존중, 신뢰의 원칙에 딱 맞는 "겸손한" 태도입니다.

참선 마스터가 되어라

개발자로서 당신의 회의주의와 비관론을 키워왔을 가능성이 큼니다. 그러나 팀을 리드할 때에 이런 태도는 마이너스 요인입니다. 모든 상황에서 순진한 낙관주의적 태도를 가지라는 뜻은 아닙니다. 회의적인 말을 줄이고, 당신이 어떤 문제점과 장애가 예상되는지 잘 파악하고 있다는 것을 팀원들이 알게하면 됩니다. 당신의 감정적 반응을 조절하고 평정심을 유지하는 것이 많은 사람들을 리드할 때 중요합니다. 왜냐하면 팀원들이 당신이 풍기는 분위기에서 어떻게 행동할지에 대한 단서를 얻기 때문입니다.

이를 비유적으로 쉽게 이해하는 방법이 회사의 조직을 톱니바퀴들이 연결된 것으로 이해하는 것입니다. 개별 기여자들은 말단의 톱니바퀴로서 톱니의 갯수가 많지 않습니다. 상위 관리자들은 톱니수가 더 많은 톱니바퀴이고, CEO는 가장 큰 톱니바퀴가 될 것입니다. 이런 상황에서 관리자 톱니바퀴가 한바퀴 돌면 개별기여자의 톱니바퀴는 여러바퀴 돌게 됩니다. 그런식으로 CEO 톱니바퀴가 조금만 돌아가면 마지막 레벨의 톱니는 미친듯이 돌아갈 수도 있습니다. 톱니연쇄의 상위로 올라갈수록 의도하지 않더라도 작은 움직임이 아래 톱니바퀴에 끼치는 영향이 커집니다.

다르게 말하면 리더는 항상 무대 위에 있습니다. 리더는 항상 주시의 대상이 된다는 말입니다. 회의를 주재하거나 연설할 때 뿐 아니라 홀로 앉아서 메일을 쓰고 있을 때에도 누군가가 당신을 주시합니다. 팀원들은 당신의 미세한 바디 랭귀지를 읽으려 하고, 잡담시의 반응에 주목하고, 점심먹을 때의 태도들도 기억합니다. 그들이 확신을 읽을까요 두려움을 읽을까요? 리더로서 당신의 역할은 영감을 주는 것이고, 시간을 정하지 않고 24시간 벌어지는 일입니다. 모든 것에 대한 당신의 태도가 무의식적으로 팀원들 사이에 퍼져나간다는 것을 명심하십시오.

초기 구글의 개발담당 부사장이었던 빌 코그런은 항상 평정심을 갖는 능력이 있었습니다. 뭐가 날아가도, 무슨 일이 터져도, 화재가 아무리 커도 빌은 당황하지 않았습니다. 항상 그는 한 팔을 가슴에 끼고 다른쪽 손은 턱에 댄 채로 - 완전히 겁에 질린 개발자에게 - 문제가 무엇인지 질문했습니다. 그런 태도가 개발자들을 차분하게 만들었고 머리없는 닭처럼 허둥거리는 대신에 문제해결에 초점을 둘 수 있게 했습니다. 우리들은 빌에 대해 농담을 만들기도 했는데 만약 회사 건물 19개가 외계인으로부터 공격당하고 있다고 보고를 하면 그의 대답은 이럴 거라는 거였습니다. "왜 그들이 20개 째는 공격하지 않은 건지 생각해 봤나요?"

이런 이야기에서 평정심 관리의 기술을 발견할 수 있습니다: 질문하기입니다. 만약 팀원이 당신에게 조언을 부탁하면 뭔가를 바로잡을 수 있는 기회라고 기뻐할 수 있을 것입니다. 문제 바로잡기가 당신이 관리자가 되기 오래전부터 익힌 아주 익숙한 행동이기 때문에 당신은 문제 해결 모드로 빠져들기 쉽습니다. 그런데 그런 행동은 하지 않는 것이 좋습니다. 당신에게 조언을 구한 팀원은 당신이 문제를 해결해 주길 원하는 것이 아니라, 그들이 해결하는 것에 도움을 달라고 한 것입니다. 그들의 문제 해결에 도움을 주는 가장 쉬운 방법은 질문을 하는 것입니다. 질문 주사위처럼 아무 질문이나 하는 것이 도움된다는 말은 아닙니다. 겸손, 존중, 신뢰의 원칙에 따라서 그들이 스스로 문제를 정제하고 탐색할 수 있도록 하십시오. 그러면 대개의 경우 그들은 정답에 도달하고, 그 정답은 당신의 아니라 그들이 얻은 답이 됩니다. 스스로 답을 얻은 사람은 더 큰 주인의식을 갖게 마련입니다. 당신이 실제로 정답을 갖고 있든 아니든 답을 이끌어내는 질문을 던지는 당신을 보는 팀원들은 당신이 답을 가지고 있었다는 인상을 받습니다. 신기하지 않습니까? 소크라테스도 당신을 자랑스러워 할 것입니다.

촉매가 되어라

화학용어인 촉매는 화학반응을 촉진하면서 반응시에 소모되지 않는 물질입니다. 효소와 같은 촉매의 동작원리는 반응물질들이 용액을 자유롭게 돌아다니게 두지 않고 서로 가깝게 위치되도록 하는 것입니다. 촉매를 통해 물리적으로 가까이 있게 되면 반응물질들 사이에 반응이 일어날 확률이 높아집니다. 이것이 리더로서 당신이 해야 할 역할입니다. 이런 리더의 역할 중에 가장 일반적인 것은 "공통 인식을 만들기" 입니다. 이 작업은 당신이 처음부터 끝까지 간여해야 할 수도 있고, 아니면 올바른 방향으로 살짝 찢어보는 것으로 충분한 경우도 있습니다. 팀의 공통인식을 만드는 것은 공식적 권위가 필요없으므로 비공식적 리더도 수행가능한 일입니다. 만약 당신이 권력을 가지고 있다면 방향 결정을 하고 지시할 수 있습니다만 공통인식에 기반한 업무보다는 비효율적일 가능성이 있습니다. 팀이 빠르게 행동해야 하는 상황에서는 팀이 스스로 공통인식에 도달하는 소모적 과정을 포기하고 누군가가 지시하는 것에 합의하기도 합니다. 이런 상황이 독재적이라고 볼 수도 있지만, 자발적 결정에 따른 것이라면 그 또한 일종의 공통인식이라고 할 수 있습니다.

도로 장애물을 치워라

팀이 이미 목표에 대한 공통인식을 가지고 있음에도 불구하고 장애물을 만나서 멈추게 되는 경우가 있습니다. 장애물은 기술적인 문제일 수도 있고 조직적인 문제일 수도 있습니다만, 이를 제거하고 팀이 다시 전진할 수 있게 만드는 것은 리더의 역할입니다. 어떤 장애물의 경우 팀원들은 거의 해결이 불가능한데 당신은 쉽게 해결가능한 것들이 있습니다. 당신이 이런 장애물을 제거할 수 있고, 제거하는 역할을 하고 싶어한다는 것을 팀원들이 잘 알게 하십시오.

일전에 구글의 어떤 팀은 사내 법무부서와의 문제로 몇주간을 업무 진행을 하지 못하고 있었습니다. 도저히 해결방도를 찾지 못한 팀원은 매니저에게 찾아가서 그 사실을 알렸습니다. 그러자 매니저는 해당 문제를 논의할 책임자를 알고 있었고 그를 찾아가서 몇시간내에 문제를 해결하였습니다. 다른 예로, 추가적인 서버 리소스가 필요하였는데 배정받지 못하는 문제를 겪는 팀이 있었습니다. 다행히 그 팀의 매니저는 사내 여러 부서와 두루 알고 지냈으므로 그날 오후가 되자 필요한 서버 자원을 확보할 수 있었습니다. 또 다른 예로, 한 개발자는 해독불가한 자바코드와 씨름하고 있었습니다. 그 팀의 매니저는 자바전문가는 아니었지만 다른 팀의 전문가와 대화해 보도록 주선하여 문제를 정확히 파악할 수 있게 하였습니다. 장애물을 치울 때 당신이 모든 답을 알 필요는 없습니다. 답을 아는 사람을 아는 게 도움이 됩니다. 많은 경우 올바른 답을 아는 것보다 올바른 사람을 아는 게 더 가치가 있습니다.

선생님과 멘토가 되어라

팀 리더로서 가장 어려운 일은 신참 팀원이 세시간씩 들어가면서 하는 업무를 당신은 20분이면 끝낼 수 있을 때 가만히 참고 지켜보는 것입니다. 사람을 가르칠 때 스스로 배울 수 있는 기회를 제공하는 것은 엄청나게 어려운 일입니다. 그러나 효율적인 리더십을 위해 꼭 필요한 덕목입니다. 새로 채용된 사람들에게 특히 중요한데, 당신의 팀의 기술과 코드에 대해서 배울 뿐만 아니라 당신의 팀의 문화를 이해하고 적절한 책임감을 갖게 하기 때문입니다. 좋은 멘토는 팀이 커짐에 따라서 멘티가 배우는 시간과 제품에 기여하는 시간의 밸런스를 잘 잡아야 합니다.

매니저의 역할과 유사하게 많은 사람들이 멘토의 역할을 자원하고 나서지 않습니다. 대개의 경우 새로운 팀원이 오면 리더가 멘토를 지명합니다. 그런데 멘토가 되기 위해서 공식적인 교육이나 준비가 많이 필요하지 않습니다. 다음의 세 가지만 있으면 됩니다 - 팀의 시스템과 절차에 대한 이해, 다른 사람에게 설명하는 능력, 그리고 멘티가 어떤 도움이 필요한지 간파하는 능력. 특히 마지막 능력이 중요합니다. 만약 당신이 너무 설명을 끝없이 많이 하면 멘티는 당신의 말을 듣지 않고 예의상 알겠다고 할 가능성이 높아집니다.

분명한 목표를 설정하라

이것은 너무 당연하게 들리는 말이지만, 정말 많은 리더들이 무시하는 패턴입니다. 당신의 팀이 한 방향으로 빠르게 움직이길 원한다면 모든 팀원들이 그 방향이 어디인지 이해하고 동의하게 만들어야 합니다. 당신이 큰 트럭을 다룬다고 가정하겠습니다. 각 팀원들의 손에 트럭의 앞면과 연결된 밧줄이 쥐어져 있다고 합시다. 당신의 의도가 트럭을 북쪽으로 빨리 이동시키는 것이라면, 당신은 팀원들이 아무 방향으로나 당기게 놔 두지 않고 북쪽으로 당기라고 말할 것입니다. 만약 당신이 분명한 목표를 세우려면 우선순위를 명확히 하는 것이 필요합니다. 그래야 팀원들이 의사결정시에 우선순위를 활용할 수 있습니다.

명확한 목표를 설정하고 팀원들이 동일한 방향으로 줄을 당기게 하는 가장 쉬운 방법은 간결한 미션 문구를 만드는 것입니다. 팀이 방향과 목표를 정의하게 한 다음에 당신은 한발짝 물러나서 팀원 스스로 결정하게 하는 것이 좋습니다. 당신은 주기적으로 그들의 방향성이 올바른지 확인하면 됩니다. 이렇게 하면 당신의 시간이 절약될 뿐더러, 당신의 팀의 효율도 높입니다. 팀은 뚜렷한 목표가 없이도 성공할 수 있습니다. 그러나 대부분의 경우 팀원들이 약간씩 다른 방향으로 줄을 당기면서 에너지만 소모하는 결과가 되기 쉽습니다. 이렇게 되면 당신은 실망할테고, 팀의 성과는 더더질테고, 방향을 바로잡는데 당신의 노력을 많이 기울여야 될 것입니다.

정직하라

당신이 팀에게 거짓말을 한다는 말이 아닙니다. 필연적으로 언젠가는 당신은 당신의 팀에게 어떤 것을 말할 수 없게 될 것이고, 모든 사람에게 그들이 듣고 싶지 않은 말을 하게 될 것이기 때문에 하는 말입니다. 우리가 아는 매니저는 새로운 팀원에게 이렇게 말합니다. "난 거짓말을 하지는 않을 것이고, 내가 말할 수 없는 경우엔 말할 수 없다고, 정말로 모르는 경우에는 모른다고 말할 겁니다"

팀원이 당신이 공유할 수 없는 것에 대해 물어본다면 답을 알고는 있지만 자유롭게 이야기할 수가 없다고 말해 주십시오. 좀 더 흔한 경우로 당신이 답을 모르는 문제에 대해 물어보는 경우가 있을 것입니다. 그런 경우에는 모른다

고 답을 해 주십시오. 이런 이야기는 읽으면 너무나 당연한 말인 것으로 생각됩니다. 많은 매니저들이 자기가 모른다고 말하면 매니저로서의 능력이 부족해 보일까 걱정합니다. 그러나 모른다는 말은 단지 매니저 역시 인간임을 보여주는 것 뿐입니다.

팀원의 잘못에 대한 피드백을 하는 것은 어렵습니다. 부하 직원이 실수를 했다고 또는 일을 제대로 못했다고 지적하는 것을 처음 해 보면 무척 스트레스를 받습니다. 대부분의 관리자 교과서에서는 이런 강한 피드백을 할 때는 "칭찬 샌드위치" 방식으로 앞뒤에 칭찬을 섞으라고 말합니다. 칭찬 샌드위치란 예를 들어 이런 방식입니다. "당신은 팀의 핵심 인력이고 가장 스마트한 개발자 중 하나예요. 그런데 당신의 코드는 아주 복잡해서 다른 팀원이 이해하기 불가능해 보이는군요. 그렇지만 당신의 잠재력은 대단하고 티셔츠들도 신박한 게 많아요."

실제로 이렇게 말하면 듣는 사람의 심리적 타격은 줄어줄 것입니다. 그런데 이렇게 돌려말하면 많은 사람들은 대화가 끝난 후에 "내 티셔츠를 칭찬했어!" 라고 생각해 버릴 수도 있습니다. 그러므로 우리는 칭찬을 앞뒤에 넣는 것에 반대합니다. 당신이 잔인해져야 한다는 의미가 아니라, 많은 사람들이 무언가를 바꿔야 한다는 핵심 메시지를 듣지 못할 가능성이 크기 때문입니다. 생산적 비판을 전달할 때 친절하고 공감하는 태도를 가짐으로써 존중의 원칙을 지키십시오. 실제로 친절과 공감이야말로 피면담자가 비판을 들을 때 즉각 방어적 태도를 취하지 않고 당신에게 귀 기울이게 하는 비결입니다.

몇해전에 동료가 다른 매니저가 같이 일하기 불가능하다고 만류함에도 불구하고 **짐**이라는 팀원을 뽑았습니다. 그 매니저의 말에 따르면 **짐**은 피드백이나 비판을 들어도 반응이 없고 하지말라고 한 방식대로만 일하더라는 것이었습니다. 제 동료는 그 매니저와 **짐**의 미팅에 배석하여 그 매니저와 **짐**이 어떻게 상호작용하는지 관찰하였습니다. 그의 관찰에 따르면 그 매니저는 **짐**의 감정을 상하지 않게 하기 위해 너무 많은 칭찬 샌드위치를 사용하였습니다. 자신의 팀으로 영입한 후에 그는 **짐**에게 팀내에서 좀더 효과적으로 일하기 위해 변화가 필요하다는 것을 분명히 설명하였습니다. "우리가 보기에 당신은 잘 몰랐던 것 같은데요. 당신이 그 팀에서 일하는 방식이 그들에겐 정보 공유가 안되고 화나게 하는 방식이었습니다. 당신이 일을 잘 하고 싶다면 커뮤니케이션 기술을 개선해야 합니다. 당신이 그렇게 하겠다면 우리는 도울 준비가 되어 있어요."

그들은 **짐**에게 어떤 칭찬이나 사탕발린 말을 하지 않았습니다. 그들은 이전 팀에서의 **짐**의 실적을 토대로 그들이 관찰한 사실을 있는 그대로 말했습니다. 놀랍게도 단 몇 주 만에 (물론 몇번의 미팅을 더 가진 후에) **짐**의 업무효율은 놀랄만큼 개선되었습니다. **짐**에게는 분명한 피드백과 지시가 필요했던 것입니다.

당신이 직접적인 피드백과 비판을 하려고 할 때, 상대가 메시지를 거부하지 않고 진심으로 듣게 만들기 위해서는 전달방식이 매우 중요합니다. 만약 그들을 방어적인 상태로 몰아간다면 그들은 자신이 어떻게 바뀌어야 할지 생각하지 않고 당신의 말 중 어디가 잘못되었는지 찾으려고만 할 것입니다. 우리 동료인 **벤**은 **딘**이라는 개발자의 관리자였습니다. **딘**의 자기주장이 강해서 어떤 문제에서든 팀의 다른 사람들과 대립하는 경향이 있었습니다. 팀 미션과 같은 큰 주제에서도 그랬고, 웹 페이지의 위젯 위치와 같은 사소한 문제에서도 그러했습니다. **딘**은 확신과 격렬함으로 자기 주장을 했으며 생각을 굽힐 줄을 몰랐습니다. 이런 행동을 수개월간 겪은 후, **벤**은 **딘**에게 그가 너무 전투적이라고 말했습니다. 이 때 **벤**이 "딘, 멍청이 짓 좀 그만 해" 라고 했다면 **딘**이 들으려 하지 않았을 것이라는 점을 아실 겁니다. **벤**은 어떻게 하면 **딘**이 자기 행동이 팀에 악영향을 끼친다는 것을 이해하게 할 수 있을지 고민하고 이런 비유적인 설명을 하였습니다. "의사결정을 할 때마다 이런 일이 벌어지고 있네. 열차가 역에 도착하고 있는 상황에 자네가 열차 앞에 나타나서 열차를 세우지. 열차는 속도를 줄여야 하고 기관사는 기분이 상할 수 있어. 새로운 열차가 15분마다 도착하는데 모든 열차를 세우려고 해. 열차를 세우면서 자네의 시간이 소모될 뿐만 아니라 나 중엔 열받은 기관사중 한명이 자네를 치고 갈 지도 몰라. 그러니 열차를 가로막는 게 가능하다고 해도 어떤 열차를 세울지 신중하게 고르고 꼭 필요하다고 생각하는 경우만 세우도록 하게."

이런 우화같은 이야기는 약간의 유머를 가미함으로써 **벤**과 **딘**이 **딘** 개인의 업무 에너지 소모 뿐만 아니라 그의 "열차 세우기"가 팀에 끼치는 영향에 대해서 논의할 수 있게 해 주었습니다.

행복을 측정하라

장기적으로 당신의 팀이 더 생산적이도록 만드는 방법은 그들의 행복도를 측정하는 것입니다. 우리가 일해 본 가장 훌륭한 리더들은 모두 아마추어 심리학자와 같이, 때때로 팀원들의 후생을 신경쓰면서, 그들이 인정받는 느낌을 갖도록 하고, 일을 즐겁게 하는지 확인했습니다. 어떤 TLM은 팀이 해야 하는 더럽고 보람없는 업무에 대한 스프레드시트를 가지고 있었습니다. 다른 TLM은 팀의 업무시간을 유심히 관찰하며 대체휴가와 팀 피크닉 등으로 번아웃이 오지 않도록 했습니다. 또다른 이는 아이스 브레이킹이라는 목적으로 일대일 면담중에 기술적인 이슈에 대해 논의

하기도 하고 업무에 필요한 장비 등이 부족함이 없는지 확인하기도 하였습니다. 가벼운 대화 후에 그는 개발자에게 얼마나 일이 재미있는지와 이후에 무엇을 더 하고 싶은지 물었습니다.

팀의 행복을 측정하는 쉬운 방법은 팀원과의 일대일 면담 말미에 다음과 같이 묻는 것입니다. "뭔가 더 필요한 것 없나요?" 이 간단한 질문은 면담을 근사하게 끝내는 방법임과 동시에 각 팀원들이 생산적이면서 행복한 상태가 되기 위해 필요한 것을 얻을 수 있게 보장하는 방법입니다. 물론 그러기 위해서는 조심스럽게 세부사항을 알아내려 하셔야 합니다. 이 질문을 매 면담마다 물어본다면 결국 팀원들이 이 질문을 기억하고 자신이 일을 더 잘하기 위해 필요로 하는 것의 리스트를 들고 당신을 찾아올 것입니다.

예상 밖의 질문

구글에서 근무하기 시작한 직후, 당시 CEO였던 에릭 슈미트와 첫 면담을 가졌다. 에릭은 "당신이 필요로 하는 게 있느냐?" 고 질문했습니다. 저는 수백만가지 어려운 질문에 대한 답을 준비했지만 이 질문에는 준비가 되어 있지 않았습니다. 그래서 아무말도 못하고 앉아만 있었습니다. 다음번에 같은 질문을 받았을 때는 제가 준비하고 있었음을 다들 이해하실 것입니다.

리더에게는 사무실 밖에서의 팀 행복도에도 주의를 기울이는 것도 가치가 있습니다. 제 동료인 메카는 일대일 면담을 얼마나 행복한지 1에서 10의 척도로 답을 해달라고 말하며 시작합니다. 그리고 그는 이 결과자료를 사무실 내와 밖에서의 행복에 대해 보고서로 작성하였습니다. 팀원들이 사무실 밖에서의 삶이 없는 것처럼 생각하지 않도록 하십시오 - 이런 생각은 사람들의 근무시간에 대한 현실적이지 않은 기대로 이어지고 팀원들이 당신에 대한 신뢰를 잃거나 번아웃에 빠지게 만듭니다. 팀원들의 사생활을 들춰보라는 이야기가 아니고 팀원이 겪는 개인적 상황에 주의를 기울이라는 것입니다. 그래야 그가 어떤 시점에 더 생산적이거나 덜 생산적인지에 대해 좀더 깊은 이해를 할 수 있습니다. 가정사로 인해 힘든 시간을 겪는 팀원에게 약간의 배려를 하는 것이 나중에 일정에 쫓길 때 자발적으로 더 많이 일할 수 있는 마음을 갖게 할 것입니다.

팀원들의 행복도를 측정하는 것의 큰 부분이 그들의 커리어를 관찰하는 것입니다. 팀원에게 5년 뒤의 그의 커리어에 대해 물으면 대개의 경우 으쓱하며 특별한 답을 못할 것입니다. 대부분의 사람이 즉각적으로는 이 질문에 답하지 못합니다. 그러나 대부분의 사람은 앞으로 5년뒤에 무엇을 하고 싶은지에 대한 생각을 합니다 - 승진을 하고 싶거나 새로운 것을 배우고 싶거나, 중요한 서비스를 개시하고 싶거나, 스마트한 사람들과 일하고 싶다는 등입니다. 정리해서 말할 수 있느냐의 여부와 무관하게 그들은 이런 생각들을 합니다. 효율적인 리더가 되고 싶다면 이런 바람들을 어떻게 이루어 줄 수 있을지 생각해야 하고, 당신이 팀원의 커리어에 대해 이러한 생각이 있음을 팀원들이 알게 해야 합니다. 중요한 것은 이런 묵시적 목표들을 명시적으로 바꾸어 당신이 조언 가능하도록 하고 달성 가능성을 측정 가능하게 만드는 것입니다.

행복도를 측정하는 것은 커리어를 관찰하는 것만으로 그치지 않습니다. 팀원들이 자기 계발을 할 기회를 누리게 하고, 각자의 일을 통해 인정받게 하고, 그 과정에서 소소한 즐거움을 얻게 해야 합니다.

그 외 팁과 트릭

다음은 리더십 직무에 있는 당신에게 구글에서 추천하는 작은 팁과 트릭들입니다.

권한 이양을 하되 당신의 손도 더러워지게 하라

개별 기여자에서 리더십 직무로 이동하게 되면, 균형을 잡는 것이 무척 어렵습니다. 처음엔 모든 일들을 당신 스스로 해내고 싶어질 것입니다. 그리고 오랜 시간 리더십 직무를 맡고 나서는 스스로는 아무 일도 하지 않게 되기 쉽습니다. 당신이 처음으로 리더십 직무를 맡았다면 팀내의 다른 개발자에게 업무를 이양하기 위해 노력해야 합니다. 당신이 직접 그 일을 완수하는 것보다 시간이 더 들더라도 그래야 합니다. 그것이 당신의 판단력을 유지하는 방법임과 동시에 팀원들이 배울 수 있게 하는 방법입니다. 만약에 당신이 장기간 한 팀을 리드했거나 새로운 팀을 추가로 담당하게 되었다면, 팀의 존경을 얻고 팀원들이 열심히 일하게 하는 쉬운 방법은 당신의 손을 더럽히는 것입니다 - 이 때 사람들이 하기 싫어하는 더러운 일을 처리하는 게 좋습니다. 당신의 경력이 화려하고 업적이 수없이 많다고 해도 팀원들이 당신이 얼마나 기술적으로 뛰어나고 집중력있게 일하는지 보여주는 확실한 방법은 어려운 일에 직접 뛰어들어 작업하는 것입니다.

당신을 대체할 수 있는 인력을 찾아라

당신이 경력의 나머지 기간 동안 동일한 일만 반복하길 원하지 않는다면, 당신을 대체할 사람을 찾으십시오. 이전에 언급했듯이 이것은 채용절차부터 시작됩니다. 당신을 대체할 수 있는 팀원을 구한다면 당신은 그런 능력이 있는 사람을 채용해야 합니다. 우리는 이것을 "당신보다 똑똑한 사람을 채용하라"라고 정리합니다. 당신의 일을 대체할 만한 팀원을 확보했다면 그들에게 더 많은 책임을 감당할 수 있게 해 주고 때로는 팀을 리드할 수 있게 해 주어야 합니다. 이렇게 해 보면 당신은 누가 가장 팀을 리드하기에 적합한지 알 수 있고, 누가 리더가 되길 원하는지 알 수 있습니다. 어떤 사람들은 리더가 되기 보다는 일잘하는 개별 기여자로 남기를 바라고, 그런 태도도 충분히 긍정할 만하다는 것을 잊지 마십시오. 우리는 많은 조직들이 최고의 개발자를 그들의 의사와는 반대로 관리자 역할을 하게 만드는 것을 목격했습니다. 많은 경우에 그렇게 하면 최고의 개발자를 버리고 평범 이하의 매니저를 얻는 것입니다.

떠나 보낼 시점을 알라

어쩔 수 없이 당신에게도 온 몸의 세포가 아무 것도 하지 말라고 외치는 것 같은 어려운 순간이 찾아올 것입니다. 당신의 팀에 있는 기술적으로 수준이하인 팀원이 문제일 수 있습니다. 모든 열차마다 막아서는 사람이 문제일 수도 있습니다. 일주일에 30시간밖에 일하지 않는 의욕없는 직원이 문제일 수도 있습니다. "그냥 기다리고 있으면 나아질 거야" 라는 생각을 당신 스스로 하게 됩니다. "결국엔 되게 되어 있어" 라고 합리화할 수도 있습니다. 이런 함정에 빠지지 마십시오 - 이런 상황은 당신이 큰 파문을 일으켜야 할 상황이고, 그것도 바로 지금 해야 합니다. 이런 상황이 저절로 나아지는 경우는 거의 없습니다. 그리고 당신이 기다려 줄수록 그들은 다른 팀원에게 악영향을 더 끼치게 되고 당신의 고민하는 밤이 더 길어질 뿐입니다. 기다림으로써 당신은 피할 수 없는 일을 연기하는 것뿐이고 연기하는 과정에서 피해만 늘릴 뿐입니다. 그러니 행동하십시오, 신속히 행동하십시오.

혼돈으로부터 팀을 지켜주어라

리더십 직무를 시작하게 되었을 때 당신이 처음으로 발견하게 되는 것은 당신의 팀 외부세계는 완전한 카오스와 불확실성의 세계라는 것입니다. 당신이 개별 기여자였을 때는 이를 알지 못했을 것입니다. 1990년대에 제가 처음 매니저가 되었을 때, 회사내에서 벌어지는 엄청난 양의 불확실성과 조직적 혼돈을 보고 깜짝 놀랐습니다. 다른 매니저에게 조용하던 회사가 무엇때문에 이렇게 요동을 치냐고 물었더니 그는 저의 순진함에 엄청나게 웃었습니다. 혼돈은 항상 존재하고 있었지만 저의 전임 매니저가 저와 다른 팀원들을 그로부터 차단해 주고 있었던 것입니다.

팀을 공중지원하라

회사의 "상부"에서 어떤 일이 벌어지고 있는지 팀이 알도록 하는 것도 중요하지만 팀 외부로부터의 불필요한 요구들과 불확실성으로부터 팀원을 보호하는 것도 중요합니다. 당신이 공유할 수 있는 정보는 공유하면서 결과적으로 그들이 영향받지 않고 알 필요가 없는 조직적 난리통으로 인해 주의가 분산되지 않게 하십시오.

팀이 잘하고 있을 때는 그 사실을 알려 주라

많은 신입 팀 리더들이 팀원들의 부족한 점을 보완하는 것에 관심이 쏠려서 긍정적인 피드백을 자주 주어야 한다는 사실을 잊어버립니다. 어떤 사람이 일을 망쳤을 때 그 사실을 알려주는 것처럼, 잘 하고 있을 때도 그 사실을 알게 하십시오. 그리고 큰 성공을 거두었을 때는 반드시 성공의 주역과 나머지 팀원들이 그 사실을 알게 하십시오.

마지막으로, 최고의 리더들이 모험심을 가진 팀원이 새로운 일을 시도하고자 할 때 대처하는 방법을 알려드립니다.

되돌리기 쉬운 일에는 "그러세요" 라고 해도 된다

당신의 팀원이 제품을 개선하기 위해 새로운 도구나 라이브러리 적용을 며칠간 시도해 보고 있다면 (그리고 당신이 데드라인에 쫓기고 있지 않다면) "좋아요. 한번 해보세요" 라고 편하게 이야기 하십시오. 만약 그들이 앞으로 몇년간 당신이 지원해야 하는 제품을 출시하려고 한다면 좀더 신중하게 생각해야 할 것입니다. 좋은 리더는 어떤 작업이 쉽게 되돌릴 수 있는 것인지에 대해 잘 압니다. 그리고 당신이 생각하는 것보다 많은 것들이 되돌릴 수 있습니다. (이 말은 기술적인 분야와 기술적이지 않은 분야 모두에 해당됩니다)

사람은 화초와 같다

내 아내는 여섯 형제의 막내입니다. 장모님은 서로 다른 것을 원하는 여섯 명의 아이들을 어떻게 키워야 하는가 하는 어려운 문제를 마주해야 했습니다. 장모님에게 어떻게 그 일을 해냈느냐고 물었을 때, 장모님은 아이들은 화초와 같다고 답했습니다. 어떤 아이는 선인장 같아서 물을 필요로 하지 않으면서 햇빛을 많이 필요로 하고, 어떤 아이는 제비꽃 같아서 직사광선이 들지 않는 습한 땅이 필요하고, 어떤 아이는 토마토 같아서 조금만 비료를 주면 엄청나게 자라난다고요. 만약 당신이 여섯 아이에게 똑같은 물과 빛과 비료를 준다면 그들은 동일한 대접을 받는 것이긴 하지만 자기들이 진정으로 원하는 것을 받지는 못할 것입니다.

당신의 팀원도 마찬가지로 화초와 같습니다. 어떤 이는 빛을 많이 필요로 하고 어떤 이는 물을 더 필요로 할 것입니다. 리더로서 당신의 역할은 누가 무엇을 필요로 하는지 알아내고 그것을 제공하는 것입니다. 물론 빛이나 물, 비료 대신에 당신의 팀원들은 서로 다른 양의 동기부여나 지시를 필요로 하겠죠.

모든 팀원에게 그들이 원하는 것을 주기 위해서는, 타성에 젖은 팀원에게는 동기부여를 하고, 집중을 못하거나 목표를 못 찾는 팀원에게는 더 강한 지침을 주어야 합니다. 물론 동기부여도 필요하고 지시도 필요한 "어중간한" 사람도 있습니다. 이런 동기부여와 지시의 조합을 통해 당신의 팀을 행복하면서 생산적으로 만들 수 있습니다. 또한 두가지 모두 사용하지 않아야 할 때도 있습니다. 동기부여와 지시를 원하지 않는 사람들은 짜증나게만 할 것이니까요.

지시를 내리는 일은 상당히 직선적입니다. 무엇을 달성해야 하는지를 알고 간단한 조직적인 기술과 업무를 관리가 능한 단위의 조각으로 나누는 능력이 있으면 됩니다. 이런 도구만 쓰면 지시를 필요로 하는 개발자에게 충분한 가이드를 줄 수 있습니다. 동기부여는 상대적으로 조금 복잡하여 약간의 설명이 필요합니다.

내재적 동기부여와 외연적 동기부여

동기부여에는 두가지 타입이 있습니다. 외연적 동기부여는 외부의 힘(예를 들어 금전적 보상)에 기인하는 것이고 내재적 동기부여는 마음 속에서 일어납니다. 작가 댄 핑크는 그의 저서 <드라이브>에서 사람들을 가장 행복하고 생산적으로 만드는 동기부여 방식은 외연적인 것(큰 돈을 주는 것과 같은)이 아니므로 내재적인 동기부여를 늘려야 한다고 말합니다. 댄은 사람들의 내재적 동기를 높이기 위해서는 다음 세 가지를 주어야 한다고 주장합니다 - 자율성, 전문성, 의미

자율성을 가진 사람은 마이크로 매니저 없이 그들 스스로 행동할 수 있습니다. 자율적인 직원들에게는 제품이 도달해야 하는 일반적인 방향성만 주고 어떻게 거기에 도달하는가 하는 사항은 그들 스스로 결정하게 맡길 수 있습니다. (구글은 자율적인 직원을 뽑으려 애쓰고 있습니다) 이것이 동기를 부여하는 이유는 그들이 제품과 (관리자들보다) 더 밀접하게 관련되기 때문이고, 더우기 그들이 자기 제품의 소유감을 갖게 되기 때문입니다. 제품의 성공에 대한 그들의 지분이 크면 클수록 그들의 제품 성공에 대한 관심도 커집니다.

전문성이란 가장 간단한 형태로는 어떤 이가 가진 현재의 기술을 향상시키고 새로운 것을 배울 기회를 제공하는 것으로 출발합니다. 전문성의 기회를 널리 제공하면 동기를 부여할 뿐만 아니라 시간이 지남에 따라 그들의 기술이 향상되고 더 강한 팀이 만들어지게 됩니다. 직원의 기술은 칼의 날과 같습니다. 수만 달러의 비용을 들여 가장 날카로운 기술을 가진 사람을 구할 수 있을 것입니다. 그러나 그 칼날을 버리지 않고 몇년을 쓰다 보면 비효율적이거나 쓸모없는 칼날이 되어 있을 수 있습니다. 구글은 항상 날카롭고 효율적인 개발자들을 유지하기 위해 그들이 새로운 것들을 배우고 기존 기술을 마스터할 기회를 널리 제공하고 있습니다.

물론 자율성과 전문성을 다 갖추었다고 하더라도 일할 이유를 찾지 못한 사람에게 동기가 생길 수 없습니다. 그러므로 당신은 그들에게 일의 의미를 제시해야 합니다. 많은 사람들이 너무나 중요한 제품을 만들고 있으면서도 그 제품이 회사와 고객 및 세계에 주는 긍정적 역할에 대해 한발짝 거리를 두고 지냅니다. 제품이 훨씬 적은 영향력을 가진 물건이라도 당신은 팀원들이 수고를 기울여야 할 이유를 찾아낼 수 있고, 그들이 이해하도록 명확히 설명할 수 있습니다. 그들이 자기 일에서 이런 의미를 찾을 수 있게 한다면, 당신은 동기부여와 생산성이 놀랍게 높아지는 것을 보게 될 것입니다. 제가 아는 한 매니저는 구글로 수신되는 자기 제품에 대한 이메일 반응을 유심히 살펴보았다고 합니다. 자기 팀의 제품이 개인적으로나 회사에서 큰 도움이 되었다는 메일을 발견할 때마다 그는 개발팀에 전달을 했다고 합니다. 이는 단지 팀의 동기를 높일 뿐 아니라 팀원들이 어떻게 하면 제품을 더 좋게 만들지 생각하는 계기가 되기도 했습니다.

결론

팀을 리드하는 것은 소프트웨어 개발자가 되는 것과는 또다른 일입니다. 그러므로 좋은 소프트웨어 개발자가 항상 좋은 매니저가 되는 것은 아닙니다. 효율적인 조직은 매니저가 될 수도 있고 개별 기여자로 남을 수도 있는 생산적인 경력 관리방식을 가지고 있습니다. 구글은 소프트웨어 개발 경력이 매니저에게 꼭 필요하다고 보고 있지만, 효과적인 매니저에게 가장 중요한 기술은 사회적 기술입니다. 좋은 매니저는 개발팀이 일을 잘하게 하고, 적절한 목표를 지향하게 하고, 외부의 문제로부터 절연시켜 줍니다. 그리고 항상 다음의 세가지 사항을 준수합니다. 겸손, 신뢰, 존중

세줄요약

- 전통적인 "관리"를 하지 마라. 리더십을 가지고, 영향을 끼치며, 팀에 봉사하라
- 가능할 때는 권한을 이양하라. 스스로 하려 하지 마라
- 당신의 팀의 관심사, 방향, 속도에 대해 예의 주시하라

공감 3

[이 글에 공감한 블로거 열고 닫기](#)

[댓글 쓰기](#) [이 글에 댓글 단 블로거 열고 닫기](#)

- [블로그](#)
- [카페](#)
- [Keep](#)
- [메모](#)
- [보내기](#)

인쇄