말.말.말

- 자신의 생각과 믿음을 지키고 실천하는 것만이 외부환경의 제약을 극복할 수 있는 유일한 방법이다.
- 자신의 내면에서 나오는 권위가 아닌 주어진 권위의 달콤함에 취하는 것은 실로 위험한 일이다.
- 물질적인 가치를 외부에 빼앗기는 것에 대해서는 분노하고 저항하면서도 정작 그 물질의 주인인 정신을 외부에 빼앗기는 것에는 무관심하게 대응하는 경우가 많다.
- 소프트웨어 개발에서 개발자의 제안이 받아들여 지지 않는다고 개발자의 가치가 낮아지는 경우는 없다. 고객이 무리한 요구를 한다고 해서 개발자의 본질이 손상되지도 않는다.
- 깨우치지 못한 경우 남을 탓하고, 막 깨우치기 시작한 경우 자신을 탓하며, 완전히 깨우치면 누구도 탓하지 않는다.
- 주변에 분노하고 절망에 가득찬 사람이 늘어가는 것은 사회가 자부심과 자존심을 중시하고 경쟁을 통해 승자를 배출하려는 것과 무관하지 않다.
- 자존감이 확립된 사람은 경쟁의 패배를 실패와 동일시 하지 않으며 자신의 내면에 상처를 입어도 회복해 낸다.
- 개인 취향이 물씬 풍기는 코드를 작성할지 아니면 유지보수가 용이한 형태로 작성할지 대부분의 경우 스스로 결정할 수 있다.
- 어차피 강요된 일정을 맞출 수 없는 경우라면 통제할 수 있는 코드를 포기하는 것보단 단 한 줄의 코드라도 제대로 동작하는 코드를 만드는 것이 소프트웨어 개발의 주인이 자신임을 확인하고 인정받는 방법이다.
- 소프트웨어 개발의 주인으로 살고 싶은가? 그렇다면 모든 귀찮고 반복적인 작업은 컴퓨터에게 맡기고 자신은 가치가 높은 업무를 수행해야 한다.
- 실패를 두려워할 필요도 없다. 실패는 끝이 아니며 그저 삶의 과정이다. 받아들이고, 직시하고 그로부터 배우는 대상일 뿐이다.

02 지속적 발전

이론과 실천 중에서 무엇이 더 효과적인가?
– 무소니우스 루푸스(Musonius Rufus) –

실천하는 철학

에픽테토스는 제자들에게 "교실에서 유능한 경우라도 실제 상황에서는 비참하게 파멸할 수 있다"고 경고했다. 아는 것과 실천하는 것은 다르다는 의미이다. 단순히 알고 이해하는 것만으로 철학이 실천된다고 보장할 수 없다. 에픽테토스는 철학이 지속적인 실천과 노력의 대상이라는 스승 무소니우스 루푸스의 가르침에 영향을 받았다. 무소니우스 루푸스는 스토아 학파 철학자로 로마에서 출생했다. 로마의 소크라테스로 불리며 존경을 받는 철학자였지만 당시 사회 기준으론 매우 급진적인 사상을 주창하여 여러 차례

추방을 당하고 죽음의 위협에 직면했다. 자신의 저서를 남겼는지 확인되지 않지만 후에 그의 제자들이 정리한 《Lectures and fragments》[01]에서 그의 사상을 엿볼 수 있다. 대표적인 예로 부당한 명령이나 요구를 받았을 경우 지위의 높고 낮음이나 나이의 많고 적음에 상관없이 이를 거부할 권리가 있다고 주장한 것이다. 당시 노예제도가 명백히 존재하고 있는 상태에서 이러한 주장은 지배계층에게 매우 위협적인 논리였다. 폭력과 먹을 것만으로 착취할 수 있던 계층이 철학과 논리로 무장하여 지배의 정당성에 의문을 갖기 시작한다면 자칫 계급 질서가 무너질 수도 있기 때문이다. 이는 현재의 직장 생활에도 고스란히 적용된다. 상위 조직장의 지시라도 내면의 기준으로 평가해보았을 때 부당하면 거부할 수 있어야 한다.

무소니우스 루푸스의 다른 주장은 여성도 남성과 동등한 교육을 받아야 하며 남성이 여성에게 성적 도덕성을 강조하듯 남성도 동일하게 그에 대한 책임을 져야 한다는 것이다. 이러한 그의 철학이 여전히 효력을 갖는 것은 인간의 본성이 별로 발전하지 못했거나 그의 철학이 시대를 앞서간 탓이겠다.

그의 많은 가르침 중에서 이번 장에서 살펴보고자 하는 부분은 경험과 경험 습득을 위한 지속적인 훈련의 중요성이다. 《Lectures and fragments》의 5번째 강의 "이론과 실천 중에서 무엇이 더 효과적인가?Which is more effective, theory or practice?"에는 다음과 같은 예시가 나온다.

"매우 뛰어난 의학지식을 가진 것으로 평판이 자자하지만 실제 환자를 진료해본 적이 없는 의사와 무명이지만 환자 진료 경험이 많은 의사가 있다. 만일 당신이 진료를 받아야 한다면 어떤 의사를 찾아 가겠는가?"

"항해 경험이 풍부하고 선장으로 일해본 사람과 지식은 풍부하지만 항해 경험은 없는 사람이 있다. 당신의 배에 선장이 필요하다면 누구를 고용하겠는가?"

"음악적인 지식이 풍부하지만 악기를 다룰 줄 모르고 노래도 부를 줄 모르는 사람과 음악 이론은 잘 모르지만 악기를 다루고 노래를 잘 하는 사람이 있다. 음악가가 필요하다면 어떤 사람을 고용해야 할까? 당신의 자식에게 음악을 가르치려 한다면 누구를 선생님으로 모셔야 할까?"

"각종 자격증만 많고 실무 프로젝트 경험이 전무한 사람과 자격증은 없지만 실무 경험이 풍부한 사람이 있다. 긴급한 프로젝트에 투입하기 위해 누구를 고용하겠는가?"

무소니우스 루푸스의 강의에는 없으며 필자가 IT 분야에 맞추어 추가한 내용이다.

예시에서 알 수 있듯이 무소니우스 루푸스는 경험을 매우

01 http://goo.gl/tcaElv

중시했다. 철학을 지식으로 배우는 것은 누구나 할 수 있지만 실제 삶에서 실천하고 경험을 쌓는 것은 아무나 할 수 없다. 그러므로 실천하고 경험을 쌓은 사람만이 진정한 철학자이다. 무소니우스 루푸스는 소크라테스의 거리 철학을 옹호했는데, 거리 철학이란 철학을 실생활에서 실천하고 전파한 삶을 의미한다. 이처럼 실생활에 적용하지 않는 철학은 박제된 지식의 자기위안일 뿐이다.

물론 경험만을 중시해 지식의 습득을 등한시 해도 된다는 의미는 아니다. 경험을 통해 지식이 진정한 지혜로 거듭날 수 있듯이, 경험도 지식을 통해 보강되고 시행착오를 줄일 수 있다. 그러므로 지식을 습득하기 위한 노력은 지속되어야 하며 습득한 지식이 가치를 갖도록 실천을 통해 경험으로 축적하고 습관으로 정착시켜야 한다. 그렇다면 철학을 실천하고 경험을 쌓기 위한 방법은 무엇일까? 무소니우스 루푸스는 《Lectures and fragments》의 6번째 강의 "훈련On Training"에서 이에 대해 설명하였다.

"금욕이 중요하단 사실을 아는 것만으로 절제할 수 있는 힘이 생기지 않는다. 쾌락에 저항하여 절제를 실천하지 않는다면 어떻게 금욕을 깨우칠 수 있겠는가? 이기심과 탐욕에 저항하지 않으면서 공정하다 말할 수 있는가? 실제 용기를 내어본 적이 없는 경우 용기를 배웠다고 할 수 있는가? 습득한 지식은 훈련을 통해 내재화 될 때에만 가치를 지니며 효과를 발휘할 수 있다."[02]

자신의 철학과 지식이 의미있는 것이 되려면 실천해야 하고 실천하는 것을 꾸준히 연습해야 한다. 프로그래밍 언어의 문법을 배웠다고 해서 개발자가 되었다고 이야기할 수 없는 것처럼 말이다. 프로그램을 직접 작성하고 오류를 겪고 수정하는 방법을 터득하고 다양한 활용법을 고민해서 더 좋은 대안을 스스로 찾아낼 수 있게 되었을 때 프로그래밍 언어를 습득했다고 말할 수 있으며 그 지식을 활용할 수 있다. 지식이 내재화 되고 어떠한 상황에서도 자신의 내면을 기준으로 흔들림 없이 발휘할 수 있게 될 때에만 지식은 가치를 가지며 그것이 자신의 실천 철학이 된다. 이를 이루기 위해선 부단한 노력과 연습이 필요하다.

무소니우스 루푸스는 육체와 영혼은 홀로 존재할 수 없으며 서로 조화를 이루어야 하므로 두 가지를 모두 단련하는 것이 중요하다고 강조했다. 육체는 정신의 도구와 같아서 언제든 제대로 사용할 수 있도록 잘 준비해야 하며 육체의 단련을 통해 정신을 더욱 강건하게 만들 수 있다고 주장했다. 추위, 더위, 갈증, 굶주림, 조촐한 식사, 불편한 잠자리, 금욕, 고통을 인내해야 더 강인한 정신을 얻을 수 있으며 자신의 내면을 견고하게 만들 수 있다고 믿었다. 이러한 믿음으로 스토아 학파는 절

02 《Lectures and fragments》의 내용을 의역했다.

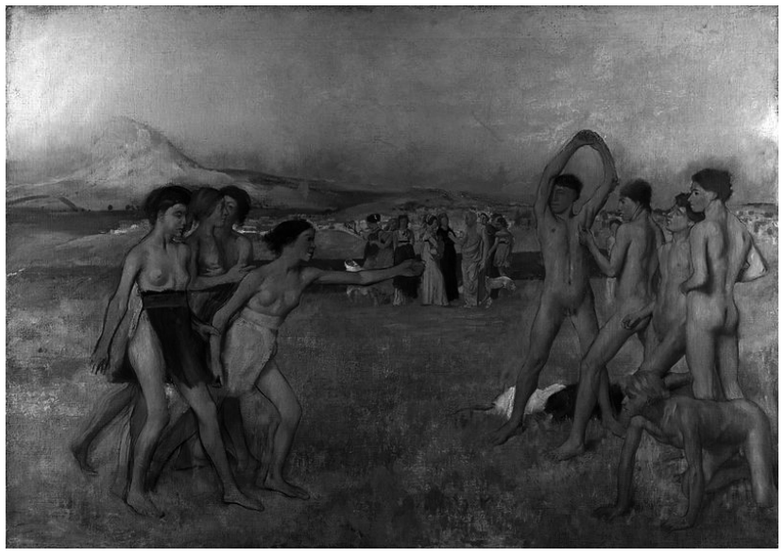

그림 1. 아고게에서 훈련 받는 스파르타 소년들의 모습. 소녀들은 아고게에서 교육을 받지 않았지만 소년들과 함께 훈련하는 것을 권장하였다.[03]

제와 금욕, 육체의 강화를 중시했는데, 그 중 육체적 강화를 극단적으로 실천한 국가가 바로 스파르타다.

스파르타는 고대 그리스의 도시국가로 펠로폰네소스 반도 남동부의 라코니아 지방, 우로타스 강 유역에 위치해 있었다.[04] 스파르타는 아고게Agoge를 통해 강인한 육체와 정신력을 가진 스파르타인을 육성했다. 스파르타는 스파르타인보다 20배나 많은 노예를 보유하고 있었으므로 노예를 통치할 강력한 힘이 필요했다. 그리고 이를 뒷받침하는 것이 어린 시절부터 혹독하게 진행되는 아고게의 훈련이었다. 스파르타에선 남자 아이가 출생하면 원로회의 의장으로 구성된 게로시아Gerousia가 스파르타인이 되기에 충분히 건강한지 검사했다. 검사를 통과하지 못한 아기는 타이게투수Taygetus 산의 아포테타이Apothetai 협곡에 방치해 죽게 했고 검사를 통과한 아기는 7살 이전까지 집에서 훈육을 받다가 7살이 되면 아고게에 입학시켰다. 아고게에선 스스로 가시풀로 잠자리를 만들게 했고 하루 10시간씩 실전 같은 군사훈련을 시키면서 가족보다는 조직에 충성하도록 교육했다. 16세가 되면 음식을 스스로 구해야 했으며 스스로 사냥한 것이나 훔친 것만 먹을 수 있었다. 훔치다 적발되면 가혹한 매질을 당해 맞아 죽는 경우도 있었는데 스파르타 소년들은 사냥보다 훔치는 것을 선호했다고 한다. 이런 매질 또한 훈련의 일종으로 여겨 피부가 강해지고 전쟁 시 포로가 될 경우 혹독한 고문에서 살아남을 수 있는 힘을 키워준다고 믿었기 때문이다. 여우를 훔친 사실을 들키지 않으려고 외투 밑에 숨긴 여우가 자신의 심장을 파먹는 상황에서도 내색하지 않고 죽은 스파르타 소년의 이야기와 같은 무용담을 즐겼다고 한다.

무소니우스 루푸스가 스파르타의 훈련을 내면을 단련하는 유일한 방식이라 주장하지 않은 것은 참으로 다행스럽다. 그랬다면 필자는 키보드를 베개 삼아 딱딱한 책상 위에서 잠을 자고 코드는 스스로 작성한 것보단 훔친 코드를 이용하고 가족보단 회사에 충성을 다 바치는 훈련을 하라고 이야기 해야만

스파르타의 가르침을 현실에 실천하기를 원하는 회사는 매우 많다. 그리고 개발자의 대부분은 이것이 훈련이 아닌 현실로 살아가고 있다.

03 http://en.wikipedia.org/wiki/File:Young_Spartans_National_Gallery_NG3860.jpg

04 http://goo.gl/lk4x2Q

한다. 다행히 무소니우스 루푸스가 주장한 정신과 육체를 단련하는 훈련에는 특별히 정해진 방식이나 우선 순위가 없다. 다만 언제나 옳음이 무엇인지를 생각하고 철학적 토론에 참여하며 진실로 잘못된 것에는 생명이 위협받는 순간이라도 참된 진리를 추구하라고 충고했다. 인간은 올바른 신념에 따라 행동하기보단 무의식적으로 관습을 따르는 경향이 있으므로 절제하고 근면하며 삶에 집착하지 않고 죽음을 두려워하지 않으며 미덕이나 돈을 베푸는 것을 받는 것보다 중시하는 훈련을 통해 이를 극복해야 한다고 강조했다.

지금부터 무소니우스 루푸스의 철학을 삶에서 실천하기 위한 방법을 살펴보겠다.

사회가 개발자에게 요구하는 철학

자신의 철학을 내재화하고 훈련하기 위해서 대상을 선정해야 한다. 추구하는 대상을 모르면 훈련을 시작할 수 없으며 대상이 잘못된 경우라면 열심히 노력할수록 더 빨리 무너지는 결과를 초래한다. 무소니우스 루푸스는 '미덕virtue'을 추구해야 할 가치로 여겼으며 프리드리히 니체는 '초인(超人)'을 인간의 궁극적 지향점으로 삼았다. 그렇다면 개발자에게 있어 '미덕'이나 '초인'은 무엇일까?

프리드리히 니체는 창조적인 의지를 가진 인간으로 스스로를 초월하여 보다 높은 상태로 올라가려는 주체적이고 창조적인 자아를 초인이라 제시했다.

개발자가 추구해야 할 가치를 살펴보기 전에 사회에서 개발자에게 무엇을 요구하는지 알아보자. 아래는 2012년 세계의 유수 IT업체 세 곳의 개발자 채용 사이트에 게재된 정보다.

기업체 정보는 표시하지 않았으며 상세한 업무 기술은 제외했다.

A사

A사의 개발자는 기술력은 물론 긍정적인 태도와 유연한 커뮤니케이션 능력까지 갖춘 소프트웨어 개발 전문가들입니다. 쏟아지는 최신 기술을 최고의 수준으로 습득하고 서비스에 적용하는 사람들, 가장 많은 사람들이 사용하는 프로그램을 만드는 것에 자부심과 책임감을 가진 사람들, 서로에게 배우는 데 주저함이 없고, 치열함 속에서도 컴퓨터에 흥미를 느꼈던 그 순간의 짜릿함을 잃지 않는 사람들이 A사의 개발자입니다.

B사

B사의 소프트웨어 엔지니어는 많은 사람이 세상의 정보에 보다 효과적으로 접근하고 상호작용하는 방법을 변화시키는 차세대 기술을 개발하며 검색의 한계를 넘어서고자 합니다. 대규모의 정보를 효과적으로 다루기 위한 정보 검색, 인공 지능, 자연 언어 처리, 분산 컴퓨팅, 대규모 시스템 설계, 네트워킹, 보안, 데이터 압축 및 사용자 인터페이스 디자인 등의 컴퓨터 과학의 모든 영역의 아이디어가 필요하며 이 목록은 계속 늘어날 것입니다. 사업의 빠른 성장에 발 맞춰 작은 팀에서 근무하며

프로젝트에 따라 팀이 변경됩니다. 항상 새로운 문제 해결에 적극적으로 나서고 기술을 발전시키는 소프트웨어 엔지니어를 구합니다.

C사

C사에서 일하는 것은 다른 회사와 많이 다릅니다. 많은 도전이 있고, 많은 영감을 얻을 수 있습니다. 그리고 자부심을 느낄 수 있습니다. 왜냐하면 여기서 어떤 업무를 맡게 되든 무언가 아주 큰 일에 기여하기 때문입니다.

A, B, C사가 소프트웨어 개발자에게 요구하는 항목을 정리해 보면 다음과 같다.

표 1 취업을 원하는 개발자에게 요구되는 항목

- 뛰어난 기술력
- 변화하는 기술을 빨리 이해하고 적용하는 능력
- 뛰어난 의사소통 능력 및 협상 능력
- 창의력
- 문제 해결 능력
- 열정
- 복종

지원자가 표 1의 항목을 모두 충족시키는 것이 가능할까? 해당 업체에 근무하는 개발자는 이런 능력을 모두 갖추고 있을까? 세부항목으로 기술된 내용이 개발자가 추구해야 할 철학일까? 개발자 스스로 이런 항목을 만족시켰음을 어떻게 확인할 수 있을까? 이런 조건에 부합하는 인재를 어떻게 판별해 낼까? 표 1의 항목은 많은 궁금증을 유발할 뿐 현실과는 너무 동떨어져 보인다. 어떻게 이런 무자비한 항목을 개발자에게 요구할 수 있을까? 이것을 이해하려면 이런 항목이 작성되는 과정의 위험성을 살펴 보는 것이 필요하다.

1961년 4월 17일 새벽, 8척의 미군 함정이 B-26의 호위를 받으며 쿠바를 기습하기 위해 출발했다. 함정에는 쿠바에서 망명한 사람들로 구성된 군인이 타고 있었으며 이들의 임무는 카스트로의 사회주의정권을 전복시키는 일이다. 작전에 참여한 군인은 모두 쿠바 사람이므로 실패하더라도 내부 반란으로 취급될 것이라 미국은 판단했다. 하지만 작전이 시작되자 함정은 암초에 걸려 좌초되었고 천신만고 끝에 연안에 상륙한 인원은 사살되거나 생포되었으며 B-26은 쿠바 공군에 의해 격추되었다. 결국 미국은 거액의 보상금을 지급하며 사과하고 포로를 교환해야 했으며 소련과 핵 전쟁의 위협에 직면했다. 반대급부로 전복시키고자 했던 카스트로 정권은 이 사건을 계기로 더욱 공고해졌다. 이 말도 안 되는 작전이 어떻게 수립되고 실행되었을까? 미국의 최고 지성집단이 모여 계획을 했음에도 말이다. 이를 설명하려면 '집단사고Group Thinking'를 이해해야 한다. 집단사고는 결속력이 높은 비교적 소수의 집단에서 이의 제기를 억제하

고 합의를 쉽게 이루려고 하는 경향성을 말한다. 이 과정에서 합리적인 이견이나 대안 분석은 영향력을 상실하고 결과를 합리화하려는 경향이 나타난다. 미국의 심리학자 어빙 재니스는 집단사고의 원인으로 결속을 강요하는 집단 분위기, 외부의견의 철저한 차단, 긴급사태로 인한 위기감 등을 꼽았다.[05]

필자 또한 표 1과 같은 항목을 작성하는 일에 참여해본 경험이 있다. 필자의 단편적 경험을 일반화할 수는 없지만 원인의 유추를 위해 소개해 보겠다.

개발팀에서 개발자 충원을 인사팀에 요청하면 인사팀은 전문가를 모아 필요 역량을 수집한다. 전문가 그룹은 보통 해당 회사에서 최고의 역량을 발휘하는 개발자로 구성된다. 결국 각자의 전문성에 필요한 항목이 나열되면서 서로를 존중하거나 경계하는 마음에 이견을 서로 제시하지 못하는 분위기 속에 최종 의견은 모든 의견의 합집합이 된다. 해당 그룹에 참여한 전문가가 개별 항목의 전문성만을 가진 경우라도 마찬가지다. 이를 취합하는 인사팀 입장에서도 기준을 높여 뽑는 것이 회사에 이로울 것이라는 판단에 이견을 제시하지 않는다. 그 결과 표 1과 같이 완전체完全體를 찾는 내용이 완성된다. 이런 내용이 단순히 회사 소개로만 사용된다면 다행이지만 문제는 그리 녹록지 않다. 개발자는 입사를 희망하는 회사의 인재상을 개인이 추구해야 할 철학으로 이해하게 되므로 잘못 선정된 인재상은 매우 위험하며 산업계 전반과 참여자에게 피해를 입힐 수 있다. 이의 위험성을 알아보자.

05 《지식 EBS 프라임》, EBS 지식프라임 제작팀 저, 밀리언하우스, 2009.12.17

뛰어난 기술력

개발자는 누구나 자신의 분야에서 뛰어난 기술력을 보유하고 싶어한다. 하지만 뛰어난 기술력이 무엇인지 정의하는 것은 매우 어렵다. 특정 도메인에 대한 깊은 지식이 뛰어난 기술력이라 정의하면 개발자는 특정 도메인에 얽매이게 되고 해당 범위를 벗어나는 순간 무능력한 존재가 되어버린다. 뛰어난 기술력이 특정 언어에 대한 전문 지식을 의미하는 경우도 마찬가지다. C++에 대한 높은 지식을 가지고 있다고 웹서비스를 모두 C++로 개발하는 것은 현명한 선택으로 평가 받지 못할 것이다. 개발 업무에 뛰어난 기술력이 전부는 아니다. 그럼에도 뛰어난 기술력만이 유일한 철학이라 이해하면 개발자는 자신의 기술력을 돋보일 수 있는 일에만 집중하고 평범하지만 꼭 해야 하는 업무엔 관심을 갖지 않게 된다. 예를 들어 잦은 통합, 테스트, 코드 품질 관리 업무는 기술적 난이도는 낮지만 개발자간에 효과적인 협업을 위해 매우 중요하다. 그러나 뛰어난 기술력에 집착하는 경우라면 이런 일상적인 업무에 흥미를 느끼지 못한다. 자신의 뛰어난 능력을 고

려하지 않고 업무가치가 낮은 일만 시킨다며 불만을 토로하게 된다. 초기에 일을 벌이고 실제 개발이 시작되면 발을 빼는 개발자가 많은 이유는 뛰어난 기술력에 대한 집착 때문인 경우가 많다. 아무리 뛰어난 알고리즘이나 전문분야 기술이라도 제대로 구현되고 유지보수가 용이해야 가치가 있다. 뛰어난 기술력을 갖춘 것과 이를 제대로 활용하는 것은 엄연히 다른 부분이다.

변화하는 기술을 빨리 이해하고 적용하는 능력

IT 분야의 기술은 정말 빠른 속도로 변한다. 그림 2는 indeed.com에서 분석한 모바일 분야 채용 트렌드이다. 2007년 블랙베리 외 타 모바일 OS시장은 매우 미비한 상태였지만 2009년을 기점으로 급격히 재편되어 2012년에는 안드로이드 개발자 수요가 아이폰을 넘어섰고 블랙베리는 추락했다. 불과 5년 만에 시장의 전체 흐름이 바뀐 것이다. 이처럼 빠른 변화를 보이는 시장에서 변화하는 기술을 빠르게 수용하고 적용하는 능력은 개발자와 회사의 생존에 필수 요소다.

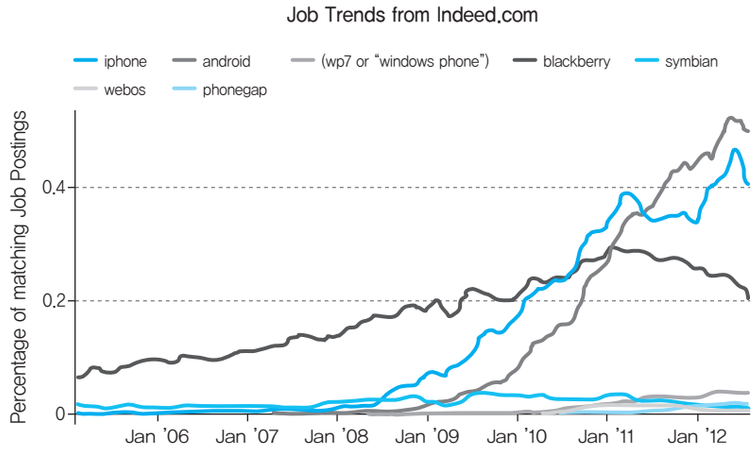

그림 2. 주요 모바일 분야 채용 트렌드[06]

하지만 그림 2를 기반으로 최신의 기술을 빨리 배우고 적용하는 것만이 중요하다고 결론 짓는 것은 매우 위험하다. 그림 2에는 기술의 변곡점과 그 이후 경향이 나와 있어 어떤 기술을 선택해야 하는지 알 수 있다. 하지만 2008년도에 이 그래프를 보았다면 어떤 기술이 주도권을 가질지 판단하기 어려웠을 것이다. 참고로 그 때의 예측 기사를 살펴보면 안드로이드의 발전이 예상된다는 내용과 함께 노키아의 새로운 스마트폰 전략이 시장에서 각광을 받을 것이란 예측도 쉽게 찾아 볼 수 있다. 한 가지 재미있는 사실은 그림 2를 보여주면 자신이 예측

06 http://goo.gl/LBlqsC