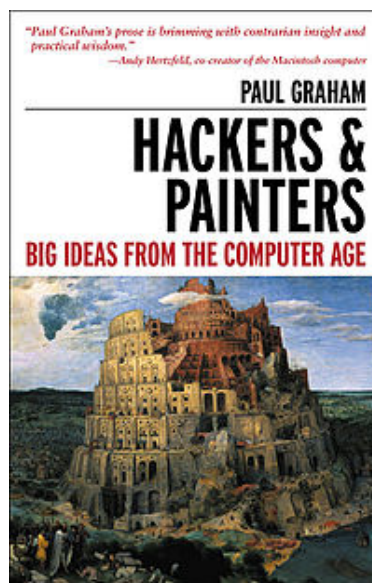


폴 그레이엄의 해커와 화가 Hackers and Painters (1/5)

PRODUCT

8월 7, 2013 박상민

Editor's note: 프로그래머이자 해커인 박상민님이 폴 그레이엄 Paul Graham의 에세이 '해커와 화가' 번역을 총 5회에 걸쳐 게재합니다. Y combinator를 창업한 폴 그레이엄은 Dropbox, Reddit, Airbnb등의 스타트업을 키워낸 대가로, 투자자이면서도 뛰어난 프로그래머이며 수필가로도 명성을 떨치고 있습니다. ([원문 보기](#))



2003년 5월

나는 컴퓨터 사이언스로 대학원을 마치자마자 미대에 진학해 회화를 공부했다. 주변의 많은 사람들이 컴퓨터를 좋아하는 사람이 그림에도 관심이 있다는 사실에 놀랐다. 아마도 해킹과 미술이 완전히 다른 종류의 일이라고 생각하기 때문인듯 싶다. **해킹이 차갑고, 정밀하고 기술적인 일이라면 그림은 어떤 현상, 사물에 대해 흥분된 감정을 주체못해 표현하는 예술이라고 생각하기 때문이다.**

이 두가지 이미지는 사실은 모두 틀리다. **해킹과 미술은 많은 공통점이 있다. 사실 내가 경험했던 모든 사람들 중 그 두가지 직업의 사람들이 가장 서로 닮았다.**

해커와 화가의 공통점은 둘 다 maker(만들어내는 사람들)이라는 점이다. 작곡가, 건축가, 작가처럼 해커와 화가는 둘 다 작품을 만들어낸다. 엄밀히 따지면 두 직업의 사람은 연구를 한다고 하긴 어렵다. 하지만 훌륭한 작품을 만들어 내는 과정에서 종종 새로운 기법(technique)을 발견해내는 경우가 있다.

(역: maker 에 맞는 적절한 한국어 단어가 없어 이후 maker를 그대로 사용합니다).

나는 컴퓨터 사이언스라는 용어를 좋아한 적이 없다 (역: 미국대학교에서는 컴퓨터전공을 컴퓨터 사이언스로 통칭). 가장 큰 이유는 그런 분야가 사실 존재하지 않기 때문이다. 컴퓨터 사이언스는 여러가지 분야가 한 가방안에 역사적인 우연에 의해 뭉뚱그려진 것과 같은 학문이다. 이를테면 유고슬라비아 같은 나라와 같다. 한쪽편엔 실제론 수학자면서 자신의 일을 컴퓨터 사이언스라 부르며 국방부의 연구비를 따내는 사람들이 있다. 중간엔 컴퓨터의 자연스런 발전과정에서 실험적 연구를하는 사람들이 있는데 예를들면 네트워크의 라우팅 알고리즘을 하는 사람들이 그렇다. 그리고 다른 극단에는 그저 재미있는 소프트웨어를 만들며, 건축가에게 콘크리트, 화가에겐 물감처럼 컴퓨터를 단지 자신의 생각을 표현하는 수단으로 이용하는 해커 그룹이 있다. 이런 조합은 마치 수학자, 물리학자, 건축가를 모두 한 전공안에 몰아넣은 것과 같다.

종종 해커들이 하는 일을 소프트웨어 공학 이라고 부르는데 이것은 용어의 잘못된 사용일 뿐이다. 뛰어난 소프트웨어 디자이너는 엔지니어보다는 건축가(architect)에 가깝다. 건축과 엔지니어링 사이의 경계가 뚜렷한것은 아니지만 분명히 둘은 구분된다. 무엇을 하느냐와 어떻게 하느냐의 차이라고 생각할 수 있는데, 아키텍트가 무엇을 하느냐를 결정한다면 엔지니어는 어떻게 하느냐를 찾아낸다.

이 “무엇”과 “어떻게”의 관계가 완전히 동떨어진것은 아니다. 당신이 어떻게 만드는지도 모르면서 무언가를 하려고들면 문제만 일으킬게 뻔하다. 하지만 해킹은 그저 스펙을 어떻게 구현할까 생각하는 것 이상의 행위다. 뛰어난 해킹은 스펙 자체를 만들어내고 그 자신이 스펙을 구현할 최적의 기술 또한 선보인다.

어쩌면 유고슬라비아가 그랬듯이 컴퓨터 사이언스는 여러개의 분야로 분리될지도 모르겠다. 아마 그건 좋은 일일 것이다. 특히 그게 나의 “조국” 해킹의 독립을 의미한다면 말이다. 이러한 여러 분야를 한 전공안에 몰아넣은것이 관리의 측면에서 편할런지 몰라도 지적인 면에선 혼란만 일으킨다. 그게 내가 컴퓨터 사이언스라는 용어를 싫어하는 또 다른 이유다. 논란의 여지가 있지만, 중

간영역에 있는 사람들 (역: 예를들어 네트워크 알고리즘)은 어떤 실험적 과학을 하는 셈이다. 하지만 양쪽 끝편에 있는 사람들, 즉 수학자와 해커들은 사실 과학을 한다고 말하긴 어렵다.

수학자 그룹은 이런 용어의 문제에 그리 신경쓰지 않는다. 그들은 옆 건물 수학과 사람들처럼 그저 열심히 정리(theorem)를 증명할 뿐이고 아마도 자신들의 건물에붙은 컴퓨터 사이언스 간판을 언젠가부터 인식도 못했을 것이다. 하지만 해커들에게 컴퓨터 사이언스라는 딱지는 문제가 된다. 자신들이 일하는 것을 사이언스라고 부르기 시작하면서 스스로 과학자같은 일을 해야한다고 느끼기 시작한다. 그래서 자신들이 정말 원하는 것, 즉 아름다운 소프트웨어를 디자인하지 못하고 학교와 연구소에서 해커들은 논문 쓰는것을 강요받는 것이다.

좋게 보자면 논문은 그저 형식일 뿐이다. 해커는 쿨한 소프트웨어를 만들고 그것에 대해 논문을 제출할 뿐이다. 논문은 그저 소프트웨어에 대한 소개 그 정도로 한정지을수 있다. 하지만 잘못된 용어는 빈번히 문제를 일으킨다. 아름다운 것을 만드는 자신의 본래 꿈에서 빗겨나가, 추하지만 연구 논문의 주제로 쓰기에 적합한 것들을 만들어내는게 참 쉬운 선택이기 때문이다.

안타깝지만 아름다운 것들은 논문으로서 최고의 주제가 되지는 못한다. 연구의 제1 기준은 독창성이다. 박사 논문을 써본 사람은 알겠지만 누구도 개척하지 않은 분야를 탐험하는 간단한 방법이 있다. 그건 아무도 원하지 않는 곳으로 가는 것이다 (역: 필요없는 연구를 하는 것이다). 두번째로 연구는 무게가 있어야 한다. 다루기 힘든 분야를 하면 많은 논문을 배출하는데 그 이유는 목표를 이루기 위해 많은 벽들을 넘어서야 하기 때문이다. 그래서 풍성한 문제들을 배출하는 최고의 방법은 잘못된 가정에서 시작하는 것이다 (역: 즉 불가능한 목표). 대부분의 인공지능 연구가 그 예다. 당신이 지식이란 논리의 나열로 표현할 수 있고, 나열된 논리가 추상적인 개념마저 표현할 수 있다고 가정하기 시작하면 당신은 풍성한 논문들을 여기에서 만들어낼 수 있다. Ricky Ricardo가 이렇게 이야기 했듯 말이다: “루시, 그거 설명할 게 아주 많아!”

아름다운 것들은 종종 이미 존재하는 것들을 미묘하게 조작하거나 이미 알려진 아이디어를 작게 변형해 조합하면서 만들어진다. 이런 종류의 일은 연구 논문에서 다루기가 어렵다.

그런데 왜 학교와 연구소는 해커를 논문의 수로 평가할까? 이유는 간단한데, 학생의 학습능력을 표준화된 시험으로 평가하는 것이나 프로그래머의 생산성을 코드의 라인수로 측정하는 이유와 같다. 골치 썩힐 필요없이 이렇게 단순한 잣대로 평가하는 것이 달콤한 유혹이기 때문이다.

해커가 진짜로 하려는 것, 즉 아름다운 소프트웨어를 평가하는 것은 훨씬 어렵다. 당신이 훌륭한 디자인을 판별할만한 눈이 있어야 한다. 자신이 그럴만한 안목이 있다고 자신만만해 하는 사람들은, 사실은 그 반대의 소질을 갖고 있는 경우가 많다.

유일한 평가의 기준은 시간이다. 시간이 지나면서 아름다운 것들은 번성하고, 추한 것들은 버려진다. 그런데 어떤 경우엔 사람의 수명보다 더 긴 시간을 요할때도 있다. 사무엘 존슨은 작가의 명성을 수렴하는데는 100년이 걸린다고 이야기 했다. 작가의 영향력있는 친구들과 그 추종자들이 모두 죽고난 후에야 진정한 명성을 이야기할수 있다고 했다.

그래서 내 생각에 해커들은 명성을 그저 행운의 결과라고 치부하는 그런 마음가짐을 가져야 한다. 그런 면에서 다른 분야의 maker들과 다르지 않다. 사실, 비교적 다른 분야에 비해 좀 나은편이다. 해킹에 비해 미술은 훨씬 더 유행에 민감하다.

다른 사람들이 당신의 일을 오해하는 것보다 더 나쁜것이 있다. 그 위험은 당신이 자기의 일을 스스로 오해하는 것이다. 우리는 가까운 곳에서 아이디어를 찾기 마련이다. 그런데 당신이 컴퓨터 사이언스과에 있다면, 자연스레 해킹이란 결국 컴퓨터 사이언스 이론을 응용한 것일 뿐이라는 그런 생각에 빠지게 된다. 나는 대학원에 있으면서 머릿속에 늘 불편한 생각을 가지고 있었다. '컴퓨터 이론을 좀 더 많이 알아야 하는데...' 그런데도 시험을 보면 3주안에 다 까먹어버리는 그 사실에 대한 불안함 말이다.

지금에서야 깨달았지만 그건 실수였다. 화가가 물감의 화학 성분을 이해하는 것 그 수준으로만 이론을 이해해도 괜찮다. 시간과 공간의 복잡도 (BigO) 정도와 튜링 테스트정도만 이해하면 된다. 아마 상태기계(state machine)을 이해하면 Parser나 정규식 라이브러리를 만들때 도움이 될 것이다. 화가는 사실 물감의 화학성분보다 기억해야 할 더 많은 것들이 있다.

한가지 발견한 사실은 아이디어를 찾는 가장 좋은 곳은 이름에 컴퓨터가 붙지 않았지만 maker들로 구성된 분야라는 점이다. 미술은 그래서 내게 컴퓨터 이론보다 훨씬 많은 아이디어를 제공했다.

예를들어 학부 시절에는 컴퓨터 근처에 가지 않고도 프로그램을 완벽히 종이에 써 낼수 있어야 한다고 배웠다. 하지만 나는 이런식으로 코딩하지 않았다. 나는 컴퓨터앞에 앉아 프로그래밍하는게 종이에 하는것보다 더 좋았다. 끈기있게 내가 맞다고 믿으며 완벽한 코드를 종이에 써내려가는 것보다는, 처음에는 완전히 망가진 형태의 코드를 모니터에 타이핑하고 조금씩 모양을 만들어 나가는게 더 좋았다. 학교에선 디버깅이 타이포나 부주의한 실수를 잡아내는 마지막 테스트라고 배웠다. 하지만 내게 코딩은 그저 디버깅의 연속이었을 뿐이다.

이런식으로 코딩하면서, 초등학교에서 연필을 이상하게 잡는다고 지적받으며 느꼈던 불편한 감정을 계속 느꼈다. 그 당시 화가, 건축가의 작업 방법을 알았더라면 이런 식의 코딩에 이름이 있다는걸 깨달았을 것이다: 그건 "스케치"다. 내가 분명히 말할 수 있는건 대학들에서 가르치는 프로

그래밍 방법은 모두 틀렸다는 것이다. 프로그램은 만들어가면서 배우는 것이다. 작가, 화가, 건축가들이 모두 그러하듯이.

이 사실을 깨달으면 소프트웨어 디자인에 다른 관점을 갖게 된다. 프로그래밍 언어에 있어 가장 중요한 것은 변화를 줄 수 있어야 한다는 사실이다. 프로그래밍 언어는 프로그램을 생각하는 과정을 위한 것이지, 이미 당신 머릿속에 그려진 지도를 표현하기 위한 것이 아니다. 즉, 볼펜이 아니라 연필이어야 한다. 정적 타입 (역: static type-c, java 언어)은 대학들에서 가르치는 방법으로 코딩한다면 괜찮은 방법이다. 하지만 내가 아는 한 해커들은 그런식으로 코딩하지 않는다. 우리에게 필요한 언어는 대충 끄적여보면서 고쳐나갈 수 있는 것이다. 조심하게 타입(type)을 무릎위에 올려놓고 먼친척 아줌마 “컴파일러”와 공손하게 대화하는 그런식의 프로그램 언어는 틀렸다.

(역: Graham은 2003년에 이 에세이를 썼는데 그 후 지금껏 10년간 해커, 스타트업 세계에서 가장 각광받은 언어는 Python, Ruby, Javascript와 같은 동적 타이핑 언어다. 그레엄이 정확하게 문제를 짚었다는걸 알 수 있다.)

지금까지 정적 타이핑에 대해 이야기 했지만, 우리 자신을 maker라고 생각할때 과학이 주는 또 다른 고통으로부터 자유로워진다. 그것은 수학에 대한 질투다. 과학계의 모든 사람들이 비밀스럽게 믿는 한가지 사실은 수학자가 자신보다 더 똑똑하다는 것이다. 수학자 자신도 그렇게 믿는다고 생각한다. 그래서 결과적으로 과학자들은 자신의 일을 수학처럼 보이게 하길 좋아한다. 물리학과 같은 분야에선 그리 문제가 안되겠지만 자연과학에서 멀어질 수록 이게 큰 문제가 된다.

한 페이지에 가득 채워진 수학 공식은 아주 인상적이다 (팁: 더 인상적으로 보이려면 그리스 기호를 쓰라). 그래서 여러가지 문제중 수학 공식으로 표현 할 수 있는 문제를 다루려는 큰 유혹이 생긴다. 더 중요한 문제들을 앞에 두고도 말이다.

해커가 스스로를 작가나 화가와 동일시했다면 그런 유혹은 받지 않았을 것이다. 작가와 화가는 수학에 질투하지 않는다. 완전히 다른 종류의 일을 하고 있다고 느낄 뿐이다. 해커도 마찬가지여야한다.

(다음 편에 계속)

0

Article Rating





박상민

Ph.D. in Computer Science, University of Virginia 현재 워싱턴주 시애틀 거주 (Bellevue)하고 있는 소프트웨어 엔지니어(Eucalyptus systems)이며, 해커, 오픈소스 팬, 블로거(Hacker, Open source enthusiast, and writer)이자 예쁜 두 딸의 아빠 (Father of two lovely girls)



KOREATECHDESK.COM

한국 스타트업 영문 기사 전문 미디어 - 코리아테크데스크
PRESS@KOREATECHDESK.COM



Be the First to Comment!

B I U [+]



0 COMMENTS



정보 **개인정보처리방침** 청소년보호정책

제보 : press@besuccess.com | 광고 : info@besuccess.com

주소 : 07335 서울시 영등포구 여의대로 108 파크원 타워1, 5층 535호 | #535, 5F, Parc1 Tower1, 108, Yeoui-daero, Yeongdeungpo-gu, Seoul, Korea 07335

주식회사 비섹세스 | 제호 : 비섹세스 | 등록번호 : 서울, 아01147 | 등록일자 : 2010년 02월 16일 | 발행일자 : 2011년 05월 17일 | 전화번호

: 070-4681-0074

발행인 : 정현욱 | 편집인 : 전진주 | 청소년보호책임자 : 전진주