

안드 로 이 드
뜻밖의 역사

Androids
The Team That Built the Android
Operating System

ANDROIDS: THE TEAM THAT BUILT THE ANDROID OPERATING SYSTEM

Copyright © 2022 by Chet Haase.

Title of English-language original: Androids: The Team That Built the Android Operating System, ISBN 9781718502680, published by No Starch Press, 245 8th Street, San Francisco, California United States 94103.

The Korean-language 1st edition Copyright © 2022 by Insight Press. All rights reserved.

The Korean edition was published by arrangement with No Starch Press Inc, through Agency-One, Seoul.

이 책의 한국어판 저작권은 에이전시 원을 통해 저작권자와의 독점 계약으로 인사이트에 있습니다. 저작권법에 의해 한국 내에서 보호를 받는 저작물이므로 무단 전재와 무단 복제를 금합니다.

안드로이드 뜻밖의 역사

세상을 뒤흔든 모바일 OS에 담긴 숨은 이야기

전자책 1쇄 발행 2022년 8월 25일 지은이 쳇 하스 옮긴이 송우일 펴낸이 한기성 펴낸곳 (주)도서출판인사이트 편집 백혜영
등록번호 제2002-000049호 등록일자 2002년 2월 19일 주소 서울특별시 마포구 연남로5길 19-5 전화 02-322-5143 팩스
02-3143-5579 블로그 <http://blog.insightbook.co.kr> 이메일 insight@insightbook.co.kr ISBN 978-89-6626-369-1



안드로이드 뜻밖의 역사

세상을 뒤흔든 모바일 OS에 담긴 숨은 이야기

챗 하스 지음 | 송우일 옮김

인사이트

차례

등장인물	x
ACK	xiv
들어가는 글	xviii

1부 시작 1

1장 안드로이드는 원래 카메라 운영 체제였다?	3
2장 인재 양성소	7
3장 팀 총원	21
4장 사업 계획 발표	31
5장 구글에 인수되다	45
6장 구글에서의 생활	49

2부 플랫폼 구축 59

7장 하드웨어와 앱을 잇는 시스템 팀	61
8장 프로그래밍 언어는 하나만	101
9장 마구잡이 라이브러리를 핵심 라이브러리로 다듬다	119
10장 실속 있는 제품 빌드와 테스트	127
11장 손바닥만 한 화면을 그리는 복잡한 기술	139

12장	‘그 벨 소리’는 누가 만들었을까?	155
13장	안드로이드의 뼈대, 프레임워크	165
14장	시각적인 사용자 인터페이스 툴킷	193
15장	안드로이드의 얼굴?	213
16장	성공과 실패를 가르는 디자인	223
17장	데스크톱 웹을 스마트폰으로 가져오다	237
18장	런던에서 온 도움	255
19장	앱이 없으면 무슨 소용?	263
20장	위스키, 불 붙은 서버, 벽돌	273
21장	위치 서비스	297
22장	안드로이드 마켓	311
23장	통신	317
24장	개발자들에게 쓸 만한 도구를 제공하라	331
25장	간결한 코드	345
26장	오픈 소스	349
27장	모든 걸 관리하다	361
28장	세계적으로 규모를 확장하다	383
29장	제품 대 플랫폼	389

3부 안드로이드 팀 393

30장	안드로이드 != 구글	395
31장	개혁 시대	401
32장	하드웨어 가지고 놀기	405
33장	로봇 가지고 놀기	409
34장	우직하게 일하다	411
35장	베이컨 일요일	417
36장	바르셀로나에서 온 엽서	421

4부 출시 425

37장	아이폰의 등장	427
38장	한편 쿠퍼티노에서는...	433
39장	SDK 출시	439
40장	1.0 출시를 앞두고	449
41장	안드로이드 스마트폰 팝니다!	457
42장	제품은 실패했지만 생각은 틀리지 않았다	463
43장	숨 돌릴 틈 없었던 연속 출시	467
44장	초기 기기들	473
45장	드roids가 해냈다	483

46장	삼성과 여러 회사	491
47장	하키 채	495

5부	안드로이드가 해낸 이유	499
-----------	---------------------	------------

48장	팀	501
49장	결정	505
50장	시기	511
51장	성공! 우리는 아직 괜찮다!	517

부록		519
-----------	--	------------

A	용어 해설	521
B	관련 콘텐츠	533

	옮긴이의 글	537
	찾아보기	540

첫 번째 독자이자 마지막 검토자,
간간한 비평가이자 절친한 친구인
아내 크리스에게 바친다.

등장인물¹

(안드로이드 팀 합류 순)

참고: 이 명단은 완전하지 않고 대부분 내가 이 책을 쓰면서 직접 만난 사람들로 한정되어 있다. 당시 제품에 실질적으로 기여한 다른 안드로이드 팀원들도 많다.

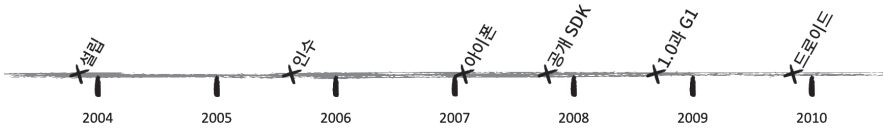
앤디 루빈Andy Rubin	창업자, 로봇 제작자
크리스 화이트Chris White	창업자, 설계자, 엔지니어, 전동 스케이트보드
트레이시 콜Tracey Cole	관리 업무 총괄, 관리자의 관리자
브라이언 스웨트랜드Brian Swetland	엔지니어, 커널 해커, 시스템 팀 리더
리치 마이너Rich Miner	창업자, 모바일 기업가
닉 시어스Nick Sears	창업자, 통신 회사 계약 담당
앤디 맥패든Andy McFadden	엔지니어, 시연 프로그램·캘린더·시뮬레이터·런타임 개발자
피커스 커크패트릭Ficus Kirkpatrick	엔지니어, 커널 드라이버 개발의 견인자, ‘크레이지Crazy’ 벨 소리 제작자
황웨이黃威	엔지니어, 브라우저, 메시지 기능 개발
댄 본스테인Dan Bornstein	엔지니어, 달빅Dalvik 창시자

1 이 명단을 얼핏 보면 초기 팀에 성비 격차가 컸음을 알 수 있다. 안드로이드 팀은 확실히 그랬고 기술 업계 전반적으로도 이는 사실이며 불행히도 오늘날도 마찬가지다. 안드로이드 팀, 구글, 다른 기술 회사에서 다양성을 증진하기 위해 노력하고 있지만 이는 긴 여정이며 이제 시작일 뿐이다. 우리가 역사를 고칠 수는 없지만 미래를 고치기 위해 시도해 볼 수는 있다.

마티아스 아고피안 Mathias Agopian	엔지니어, 그래픽 개발, 휴대 전화를 집어던지기도 함
조 오노라토 Joe Onorato	엔지니어, 빌드 시스템, 사용자 인터페이스, 프레임워크 등
에릭 피셔 Eric Fischer	엔지니어, TextView 담당
마이크 플래밍 Mike Fleming	엔지니어, 전화와 런타임
제프 액식 Jeff Yaksick	디자이너, 안드로이드 장난감과 사용자 인터페이스
캐리 클라크 Cary Clark	엔지니어, 브라우저 그래픽
마이크 리드 Mike Reed	스키아 Skia 리더, 그래픽 기술 연쇄 창업가
다이앤 핵본 Dianne Hackborn	엔지니어, 프레임워크 대부분을 개발
제프 해밀턴 Jeff Hamilton	엔지니어, 바인더, 데이터베이스, 주소록
스티브 호로위츠 Steve Horowitz	엔지니어링 관리자, 의견을 절충하는 역할
마이크 클러론 Mike Cleron	엔지니어, 사용자 인터페이스 툴킷 재작성자 이자 프레임워크 관리자
그레이스 클로바 Grace Kloba	엔지니어, 안드로이드 브라우저
아르베 히엔네보그 Arve Hjønnevåg	엔지니어, 드라이버와 디버깅, 말수는 적었지만 많은 코드를 작성함
히로시 로카이머 Hiroshi Lockheimer	기술 프로그램 관리자, 협력사 관리자
제이슨 파크스 Jason Parks	엔지니어, ‘망가뜨리는 자’로 불림
일리안 말체프 Iliyan Malchev	엔지니어, 블루투스, 카메라, 기타 드라이버
세드릭 부스 Cédric Beust	엔지니어, 안드로이드 지메일 앱 개발자
데이비드 터너 David Turner	엔지니어, 안드로이드 에뮬레이터

데바짓 고시 Debajit Ghosh	엔지니어, 캘린더 서비스
마르코 넬리션 Marco Nelissen	엔지니어, 사운드 코드
라이언 PC 깁슨 Ryan PC Gibson	기술 프로그램 관리자, 릴리스 작명과 출시
에번 밀러 Evan Millar	엔지니어, 테스트
자비에 듀크로이 Xavier Ducrohet	엔지니어, 도구 기술 대부분
마이클 모리시 Michael Morrissey	엔지니어링 리더, 서버 서비스 제공
밥 리 Bob Lee	엔지니어, 핵심 라이브러리
호만 기 Romain Guy	사용자 인터페이스 툴킷 개발에 참여한 비범한 인턴으로 시작해 정직원이 됨
톰 모스 Tom Moss	변호사, 사업 개발, 협상 해결사
브라이언 존스 Brian Jones	접수 담당자, 관리 업무 총괄 대행, 기기 제공 담당자
댄 에그노어 Dan Egnor	엔지니어, 무선 over-the-air 업데이트 개발
데이브 스파크스 Dave Sparks	엔지니어, 미디어 관리자
우페이쑤 Wu Peisun	기술 프로그램 관리자, 미디어, 메시지 그리고 도넛 버거
에드 헤일 Ed Heyl	엔지니어, 빌드, 테스트, 릴리스, 반복
더크 도허티 Dirk Dougherty	기술 문서 작성
찰스 멘디스 Charles Mendis	엔지니어, 위치 내비게이션 개발
데이브 버크 Dave Burke	엔지니어링 리더, 런던 모바일 팀
안드레이 포페스쿠 Andrei Popescu	엔지니어링 리더, 런던 브라우저 팀
니콜라 로드 Nicolas Road	엔지니어, 안드로이드 브라우저를 준비함

산 메हत San Mehat	엔지니어, 커널 드라이버와 SD 카드 디버깅
닉 펠리 Nick Pelly	엔지니어, 블루투스 와 씨름함
레베카 자빈 Rebecca Zavin	엔지니어, 기기 부팅, 드로이드 Droid 드라이버
첸치우키 陳劍琪	엔지니어, 체크인
마이클 찬 Mike Chan	엔지니어, 커널 보안
브루스 게이 Bruce Gay	엔지니어, 원숭이 실험실 담당자
제프 샤키 Jeff Sharkey	엔지니어, 경진 대회 수상자
제시 윌슨 Jesse Wilson	엔지니어, 끔찍한 API 개선
댄 샌들러 Dan Sandler	엔지니어, 시스템 UI, 삽화가, 이스터 에그 제작



ACK¹

호만 기에게 감사한다. 그가 없었다면 이 책은 결코 나오지 못했을 것이다. 그는 나를 팀에 데려왔을 뿐 아니라(실제로 일하기 전에 두 번이나 추천을 시도했다) 내가 이 책의 아이디어를 발전시키는 데 주된 역할을 했다(일부는 우리가 기술 콘퍼런스에서 함께한 발표에서 비롯됐다). 그는 또한 내가 이 책을 쓰기 위해 한 많은 인터뷰를 도와 주었다. 아, 그리고 그는 오늘날 수십억 명의 사람이 사용하고 우리가 지금도 만지고 있는 많은 코드를 작성했다.

이 프로젝트의 처음부터 마지막 단계까지 내내 사려 깊은 통찰과 지원, 전문적인 편집 피드백으로 도움을 준 내 아내 크리스에게 감사한다. 또한 이 프로젝트가 우리 가정 생활을 아주 오래 지배했는데도 날 죽이지 않아서 고맙다. 이제 끝났다. 그렇게 생각한다.

안드로이드가 어떻게 동작하는지 다른 안드로이드 개발자들과 이야기를 나누는 팟캐스트인 안드로이드 개발자 백스테이지(Android Developers Backstage)²의 오랜 공동 진행자(호만도 함께 진행) 토어 노비(Tor Norbye)에게 감사한다. 우리가 한 몇몇 인터뷰(피커스 커크패트릭, 마티아스 아고피안, 데이브 버크 포함)는 이 책에 직접적으로 도움이

1 ACK(acknowledge)는 데이터가 전송되는 도중 사라질 수도 있는 컴퓨터 통신에서 사용되는 신호다. 한 시스템이 또 다른 시스템으로 메시지를 보내면 메시지를 받은 시스템은 발송자가 메시지를 다시 보낼 필요가 없다고 알려 주기 위해 '알았다'를 의미하는 ACK로 응답해야 한다(옴긴이: 책에서는 원 의미에서 파생된 '감사의 말'이라는 뜻으로 쓰인다).

2 <http://adbackstage.libsyn.com> 또는 좋아하는 팟캐스트 앱에서 찾아보면 나올 것이다.

됐는데 그들이 안드로이드 역사에서 중요한 인물이었기 때문이다. 그들은 사람들과 이야기 나누기 좋아하는 내게 이 프로젝트의 기원, 심장, 영혼이 된 그들의 작업을 들려주는 기여를 했다.

이 책의 표지와 여러 쪽을 탁월하고 재미있는 삽화로 꾸며 준 덴 샌들러에게 감사한다. 그는 마운틴 뷰 사무실에 방문해서 화이트보드에 지문처럼 만화를 남겼는데 나는 그의 만화를 보는 걸 늘 좋아했다. 그리고 팀과 제품에 대한 농담을 담은 그의 그림을 이 책에서도 볼 수 있어서 좋다.

이 책을 출판할 수 있는 형태로 만드는 데 도움을 준 친구이자 탁월한 책 디자이너인 그레첸 아킬리스Gretchen Achilles에게 감사한다.

조나선 리트먼Jonathan Littman에게 감사한다. 성공적인 책 몇 권을 쓴 저자로 책, 저자, 출판사의 현실 세계가 어떻게 돌아가는지에 대한 내 질문에 참을성 있게 대답해 주었다.

내 편집자인 로린 허드슨Laureen Hudson에게 감사한다. 그녀가 없었다면 이 책은 다듬어지지 못해서 내용이 탄탄하지 못했을 것이고 정리되지 못해서 읽을 만하지 않았을 것이다. 처음으로 로린과 함께 일한 건 우리가 오래 전 썬에서 일했을 때 내가 기술 기사를 쓰면서였다(그녀가 편집했다). 당시와 같은 관계를 재개해서 그녀가 엉망인 내 글을 다시금 정리해 주어서 기뻐다.

그리고 특별히 구글 안드로이드 엔지니어링 부사장인 데이브 버크를 언급하고 싶다. 버크는 이것이 알릴 만한 가치가 있는 이야기라는데 동의했고 회사, 직원, 제품에 대한 내부 이야기를 할 때 불쑥 나타날 수 있는 장애물을 내가 넘을 수 있도록 도와주었다.

초기에 안드로이드 개발을 도왔는데 내가 이야기를 나눌 기회가 없었던 모든 사람에게 사과의 말을 전한다. 나는 계속 이어지는 인터뷰를 즐겁게 진행하며(전체 프로젝트에서 가장 재미있는 부분이였다) 사람들, 그들이 한 일, 그들의 출신, 안드로이드를 만든 방식에 대해 배웠다. 그러나 어느 시점에 나는 이 책을 완성해야 했다.

그리고 사심 없이 자신의 시간을 내어 주고 의견과 이야기를 들려주어 당시 무슨 일이 실제로 일어났는지 이야기를 더 잘 전달할 수 있게 해 준 과거와 현재 모든 안드로이드 팀원에게 감사한다. 대화와 이메일로 인터뷰에 참여해 준 모든 사람에게 특히 감사하고 싶다. 거의 모든 사람이 내 질문에 인내심 있게 대답해 주었을 뿐 아니라 책 프로젝트와 우리의 대화에 열정적으로 참여해 주었다.

안드로이드 이야기는 내가 직접 인터뷰하고 이야기를 나누고 이때 일을 주고받고 아니면 귀찮게 군 사람들의 이야기 그 이상이지만, 시간을 내어 사실을 알려 주고 완전히 잘못된 이야기를 바로잡도록 나를 도와준 사람들에게 감사하고 싶다. 마티아스 아고피안, 댄 본스테인, 세드릭 부스, 이리나 블로크, 밥 보처스, 데이브 보트, 데이브 버크, 첸치우키, 마이크 찬, 캐리 클라크, 마이크 클러론, 트레이시 콜, 크리스 디보나, 디크 도허티, 자비에 듀코이, 댄 에그노어, 에릭 피셔, 마이크 플레밍, 브루스 게이, 데바짓 고시, 라이언 PC 김슨, 호만 기, 다이앤 핵본, 제프 해밀턴, 에드 헤일, 아르베 히엔네보그, 스티브 호로위츠, 황웨이, 브라이언 존스, 피커스 커크패트릭, 그레이스 클로바, 밥 리, 댄 루, 히로시 로카이며, 일리안 말체프, 앤디 맥패든, 산 메हत, 찰스 멘디스, 에번 밀러, 리치 마이너, 댄 모릴, 마이클 모리시,

톰 모스, 마르코 넬리션, 조 오노라토, 제이슨 파크스, 닉 펠리, 안드레이 포페스쿠, 장 바티스트 케루, 마이크 리드, 니콜라 로드, 앤디 루빈, 댄 샌들러, 닉 시어스, 제프 샤키, 데이브 스파크스, 브라이언 스웨트랜드, 데이비드 터너, 폴 휘튼, 제시 윌슨, 우페이쑨, 제프 액식, 레베카 자빈.

또한 시간을 들여 초고를 읽고 피드백을 준 많은 사람에게도 감사하고 싶다. 코드는 검토를 받으면 늘 더 나아지고 책도 마찬가지다. 특히 상당한 노력을 기울여 준 몇몇 사람을 언급하고 싶다. 그들은 전체를 신중하게 살펴봤고 빈틈을 확인해 주었고 장황한 부분을 잡아냈고 실수를 바로잡아 주었고 추가 정보를 제공했으며 원고를 주의 깊게 검토해서 최종 원고에 크게 기여했다. 이 책에 대한 피드백 의견 만으로도 또 다른 완전한 책을 쓸 수 있을 정도라고 생각한다. 특히 다이앤 핵본, 브라이언 스웨트랜드, 앤디 맥패든이 즉각적이고 사려 깊고 완벽한 검토를 해 주어서 기술적 정확도에 결과적으로 도움이 됐다. 또한 내 친구 앨런 웰린도스키(Alan Walendowski)에게도 감사한다. 막바지에 원고를 읽고 검토해 주어서 같은 텍스트를 아주 여러 번 읽고 또 읽다 보면 오히려 찾기 어려워지는 남은 문제를 잡아내는데 도움이 됐다.

고맙고 고맙고 고맙고 고맙다. 그리고 살짝 거슬릴 수도 있는 세세한 내 실수들에 대해 사과와 말을 전한다. 버그로 제출하기 바란다.

들어가는 글



2010년 5월 중순 나는 구글 구내 44번 건물에 있는 안드로이드 팀으로 첫 출근을 했다. 내 책상에서 얼마 떨어지지 않은 곳에 온갖 커피를 내리는 기계가 적어도 여섯 대 놓여 있었다. 사람들이 카페인을 그토록 중요하게 여기는 걸 보고 놀랐지만 오래가지는 않았다.

팀은 릴리스 하나¹를 마무리하면서 다음 릴리스²를 동시에 작업하고 있었다. 우리는 당시 북적대는 스마트폰 시장에서 안드로이드를 유의미한 제품으로 만들어야 했기에 둘 다 까다롭고 시간이 많이 걸리고 대단히 중요한 작업이었다. 결승선을 향해 맹렬히 달리는 느낌을 지속적으로 받았는데 목표를 이루기 위해 무엇이든 해야 했고 과연 우리가 할 수 있을지 알 수 없었기 때문이다. 미친 듯한 속도로 일했지만 작업은 엄청나게 신났는데 단지 카페인 때문은 아니었다. 일을 얼마나 하든 상관없이 목표에 각별히 헌신한 팀이 있었기에 신이 났던 것이다.

안드로이드 개발은 내가 개발자 경력을 시작한 이후 맞이한 극명

1 안드로이드 2.3 진저브레드(Gingerbread)

2 안드로이드 3.0 허니콤(Honeycomb)

한 변화였다.

나는 미네소타에 있는 보수적이고 전통 있는 회사에서 아홉 시에 출근해 다섯 시에 퇴근하는 평범한 직장 생활을 시작했다. 회사는 경력 내내 그리고 은퇴 후에도 회사를 떠나지 않는 사람들에 의지했고 해마다 추수 감사절에 퇴직자들에게 칠면조를 공짜로 보내 주었다. 나는 만반의 준비가 되어 있었다. 그러니까 그저 주당 40시간을 채우고 천천히 승진을 하다 은퇴해서 칠면조를 받을 준비만 하면 됐다.

1년도 안 되어 지겨워졌고 2년이 되기 전에 대학원 진학을 위해 회사를 그만두었다. 내가 정말 즐길 수 있는 무언가로 내 기술을 ‘리부팅’하고 싶었는데 바로 컴퓨터 그래픽 프로그래밍이었다. 대학원을 마친 후 나는 기술자에게 기회의 땅³인 실리콘 벨리로 향했다. 썬에 입사해서 몇 년간 열심히 일했는데 또 다른 흥미로운 일자리가 내게 손짓했다.

그 후 몇 년간 나는 다른 직업, 기술, 사람들이 내 기술 인생에 끊임 없이 변화를 제안할 때마다 이 회사에서 저 회사로 옮겨 다녔다. 나는 썬(재입사로 여러 번), 애니웨어 패스트^{Anyware Fast}(친구들이 시작한 도급 회사), 디멘션엑스^{DimensionX}(초기 웹 스타트업으로 마이크로소프트에서 인수), 인텔, 렌디션^{Rendition}(3D 칩 스타트업으로 마이크로론에서 인수), 어도비에서 일했다.

미 해군에서 21년간 복무 후 퇴역한 아버지는 내 잦은 이직을 불편해하셨다. 연금은 어떻게 되는 거니, 고용 보장은, 가족 부양은 어떻게 할 거니 하는 말씀이었다.

3 아니면 적어도 기술 회사들에는 기회의 땅인데, 그 덕분에 여러분이 현장에서 일자리를 구할 때 도움이 된다.

아버지가 알지 못하신 건 실리콘 밸리는 그런 식으로 돌아갔고 지금도 마찬가지이며 첨단 산업 어디나 점점 그렇게 되어 가고 있다는 점이었다. 나는 새로운 일을 시작할 때마다 미래의 가능성과 제품에 기여할 새로운 기술을 쌓아갔다. 똑같은 태도와 현실이 기술 회사 세계에서 이직하는 모든 엔지니어에게 적용된다. 알려지지 않은 문제와 씨름하고 혁신적인 해법을 내놓기 위한, 새 프로젝트에 꼭 필요한 기술은 다양한 배경에서 나온다.

2010년 또 다른 기회가 생겼다. 2005년 쯤에서 인턴으로 나와 함께 일했던 친구인 호만 기⁴에게 문제가 생겼다. 호만은 2007년 안드로이드 팀에 합류했는데 너무 바빠서 그가 필요하다고 생각했던 애니메이션 시스템을 만들 겨를이 없었다. 그런데 그는 나를 알고 있었고 그 일이 내가 좋아할 프로젝트라는 것도 알고 있었다. 몇 차례 면접을 보고 5개월 후 나는 마운틴 뷰에 위치한 구글 구내 44번 건물에 있는 안드로이드 사용자 인터페이스 툴킷 팀에 합류했고 그 어느 때보다 더 열심히 일하기 시작했다.

나는 새로운 애니메이션 시스템을 만드는 일을 시작으로 다음 릴리스를 완성하는 데 주력하면서 저수준 성능 개선 및 그래픽 소프트웨어 작업을 했다. 같은 팀에서 오랫동안 계속 일하면서 그래픽, 성능 개선, 사용자 인터페이스 코드를 작성했고 결국 몇 년간 팀을 이끌게 됐다.

4 나는 호만과 함께 《Filthy Rich Clients: Developing Animated and Graphical Effects for Desktop Java Applications》(Addison-Wesley, 2007)라는 책을 썼다. 여러분이 이 책을 좋아한다면 내가 쓴 다른 책들도 고를지 모르겠다. 그런데 이 책은 내가 추천할 만한 책이 아닌 것 같다. 그러니까 내 말은 나는 이 책을 정말 좋아하지만 2007년에 나온 책이어서 기술 세계 시간으로 따지면 최소한 수십 년 전 내용이라는 뜻이다.

안드로이드 이전 내 프로젝트는 대부분 정말 즐거웠지만 그다지 가시적인 일이 아니었다. 가족이 나한테 무슨 일을 하느냐고 물어보면 내가 작성하는 소프트웨어에 대해 이야기해 주었다. 그런 다음 내가 만든 소프트웨어를 사용할지도 모르는 응용 분야에 대해 대강 설명해 주었다. 가족들이 내 코드의 결과물을 절대 볼 수 없는 게 현실이었기 때문이다. 그건 실제 사람(소비자)들이 현실에서 접할 수 있는 제품들이 아니었다.

그러다가 안드로이드 팀에 합류하고 전 세계 사람들이 날마다 사용하는 소프트웨어⁵를 만들었다. 적어도 안드로이드가 살아남는다면 그럴 것이다.

도전

“완전히 실패할지, 잘될지 모르겠네요. 잘된다면 사람들이 신나하면서도 깜짝 놀랄 것 같아요.”

- 에번 밀러

초기 안드로이드 팀은 많은 경험과 강한 견해를 지닌 사람들로 구성되어 있었다. 그들은 자신들이 만들려고 하는 것에 대해 자신이 있었지만 첫 1.0 릴리스를 만들기 위해 힘겨운 싸움에 직면하고 있었다.

팀의 목표는 안드로이드 운영 체제를 만드는 것이었다. 이 목표에

5 그리고 버그도 만들었다. 버그가 없는 유일한 코드가 있다면 그건 아직 작성되지 않은 코드다. 우리는 할 수 있는 한 테스트를 많이 하지만 현대적인 소프트웨어의 복잡도 때문에 버그는 늘 있을 수밖에 없다. 요령이 있다면 버그가 치명적이지 않은지 확인하고 버그를 발견하면 고치는 것이다. 그런 다음에는 하던 대로 더 많은 코드를 짜면 된다(그리고 더 많은 버그도 나올 것이다).

는 저수준 커널과 하드웨어 드라이버부터 전반적인 플랫폼 소프트웨어까지 모든 것이 포함되어 있었다. 또 애플리케이션용 API, 애플리케이션을 만드는 데 도움이 되는 도구, 플랫폼 번들 애플리케이션, 애플리케이션과 통신할 백엔드 서비스를 개발하는 일도 수반됐다. 아, 그리고 팀원들은 이 운영 체제를 모든 새 휴대 전화에 탑재하고 싶어 했다.

소프트웨어는 제조사가 휴대 전화를 만들 수 있도록 무료로 제공될 예정이었다. 즉, 협력사들은 하드웨어를 만들고 안드로이드는 소프트웨어를 제공하는 방식이었다. 운영 체제 덕분에 제조사들은 하드웨어 제품을 만드는 데 집중하고, 점점 복잡해지는 소프트웨어 문제는 안드로이드에 맡기면 되는 것이었다. 동시에 안드로이드는 다양한 휴대 전화에 동일한 플랫폼을 만듦으로써 애플리케이션 개발자에게 도움이 되고자 했다. 개발자는 기기마다 서로 다른 버전을 만드는 대신 모든 기기에서 동작하는 한 가지 버전의 앱을 만들 수 있게 됐다.

안드로이드 팀은 구글로부터 자금을 지원받았고 매우 큰 규모에서 소프트웨어를 개발해 본 경험이 있는 내부 엔지니어 풀을 활용할 수 있었다. 스마트폰 시장은 급속도로 성장하고 있었고 헌신하는 팀이 곧 출시될 제품을 개발하고 있는데 어떻게 실패할 수 있을까? 지나고 나서 보면 안드로이드의 성공은 정해져 있었던 것처럼 보일지도 모른다.

그러나 초기에 팀은 매우 다른 현실에 처해 있었고 안드로이드는 지속되기에는 보잘것없는 상태였다.

우선 팀이 구글의 경제적 지원을 받고 있었다고는 하지만 안드로이드

이드는 구글에서 투자하는 많은 프로젝트 중 하나였을 뿐이다. 구글이 안드로이드에 돈을 건 것은 모든 영향력을 발휘하여 지원하겠다는 기보다는 뭐가 가능한지 보려고 후원한 것이었다.⁶

또 안드로이드는 새로운 회사가 들어설 틈이 거의 없는 산업에 들어서고 있었다. 경쟁자들이 이미 단단히 자리를 잡고 있었기 때문이다. 저가 시장에는 수많은 노키아(Nokia) 폰이 세계적으로 퍼져 있었다. 데인저(Danger), 블랙베리(Blackberry), 팜(Palm)은 충실한 열혈 사용자들에게 흥미로운 스마트폰⁷ 선택지를 제공하고 있었다. 그리고 다양한 마이크로소프트 폰도 있었는데 소프트웨어 산업에서 일하는 사람이라면 누구나 마이크로소프트와 경쟁하는 것을 늘 경계해야 한다는 사실을 알고 있었다.⁸

2007년 애플이 시장에 들어왔는데 이미 복직대는 시장에 처음 등장한 또 다른 경쟁자였다. 애플은 휴대 전화에는 처음이었지만 운영체제, 소비자 컴퓨팅 기기, 널리 보급된 아이팟으로 증명된 실적을 이미 갖고 있었다.

이 모든 회사는 안드로이드가 제품은 고사하고 보도 자료도 내기

- 6 구글은 같은 시기에 웹 기술에서도 비슷한 전략을 취했다. 자체 브라우저(크롬)를 만들기 시작하면서 다른 회사들이 향후 해당 분야를 차지할 가능성에 대비해 그 분야를 개척할 필요가 있음을 알고 있었다.
- 7 나는 이 책에서 넓은 의미로 ‘스마트폰’을 정의한다. 이 용어가 처음 쓰였을 때는 본래 휴대 전화에 데이터 통신 기능이 덧붙여져 이메일과 인스턴트 메시지 같은 더 많은 통신 기능이 가능한 기기를 의미했다. 즉, 기본적으로는 기능이 좀 더 풍부한 통신 기기였다. 오늘날 스마트폰은 이러한 기본적인 것뿐 아니라 앱, 게임, 터치스크린을 비롯해 휴대 전화가 데이터 요금제와 결합한 이래로 온라인에 등장한 아주 많은 기술을 망라하는 기기다.
- 8 첫 아이폰 제품 마케팅 디렉터인 밥 보처스(Bob Borchers)가 말했다. “절대 마이크로소프트를 빼고 생각해서는 안 돼요. 그들은 제대로 된 제품이 실제로 나올 때까지 끊임없이 돈을 쓰면서 일을 하거든요.” 몇 년 전 내 동료는 말했다. “여러분이 속한 시장에 마이크로소프트가 들어오면 거기에서 나와야 해요.”

전에 시장에 자리를 단단히 잡고 있었다.

이 거친 시장에서 경쟁하기 위해 초기 팀은 1.0에 도달한다는 한 가지 여정에만 집중했다. 모두 그 목표에 집착해서 팀원 대부분이 미친듯이 멈추지 않고 강행군했다.

그러나 그들이 이 운영 체제를 만들고 싶어 한다고 해서 모두가 그것을 어떤 식으로 만들지, 그것이 성공할지, 그들이 만들고자 하는 것이 정확히 무엇인지에 대해 동의한다는 의미는 아니었다. 팀 엔지니어인 앤디 맥패든은 “일을 하는 옳고 그른 방법에 대해 자기 주장이 강한 사람들이 많이 있었죠. 때로는 서로 동의하지 않았고 거칠어 지기도 했어요”라고 말했다.

안드로이드가 1.0이 되고 첫 안드로이드 폰이 출시됐을 때조차도 팀원들은 프로젝트가 성공할지, 심지어 계속될 수 있을지 확신하지 못했다. 팀 기술 프로그램 관리자(technical program manager, TPM) 라이언 김슨은 말했다. “초창기 안드로이드 팀은 승산이 적어 보였고 끊임없이 실패 직전에 몰렸고 어떻게든 조금이라도 진전을 보이려고 엄청나게 열심히 일해야 하는 분위기였습니다. 성공은 전혀 필연적 결과가 아니었어요. 우리는 1년이나 뒤쳐져 있었거든요. 다음 해로 밀렸다면 우리는 성공 가능한 대안이 아니라 역사에 각주 정도로 남았겠죠.”

팀에 합류한 이후 몇 년간 나는 안드로이드 초기 개발과 어려움에 대해 들었는데 모두가 플랫폼을 경쟁력 있는 수준으로 끌어올리기 위해 고투했다는 이야기였다. 그리고 나서 내가 팀에서 일하는 동안 안드로이드가 어느 정도 성공을 거두는 모습을 보고 문득 궁금해졌다. 어떻게 된 일일까? 다시 말해 안드로이드 개발의 어떤 요소가 보

잘것없던 초창기 이후 놀랄 만한 성장을 이루도록 한 것일까?

이 책을 써야겠다는 생각은 안드로이드 개발 이야기가 결국 잊힐 것임을 깨닫고 나서 시작됐다. 안드로이드를 만들었던 사람들이 다른 프로젝트⁹로 이동하면서 안드로이드를 개발하는 중에 무슨 일이 일어났었는지 점점 잊어버리고 있었기 때문이다. 2017년 나는 그 이야기를 담기 위해 초기 팀 사람들과의 대화를 녹음하기 시작했다.

“구체적으로 어떻게 구현할 건가요?”¹⁰

이 책은 긴 책이다(책을 쓰기 시작했을 때 생각했던 분량보다는 긴데 초고보다는 짧아졌다. 그렇다고 고마워할 필요까지는 없다). 안드로이드라는 큰 주제로 구성된 이 책을 좀 더 이해하기 쉽게 읽을 수 있는 몇 가지 팁이 있다.

첫째, 안드로이드라는 낱말에 주의하라. 이 책 편집자가 머리를 쥐어뜯은 이유 하나는 내가 안드로이드라는 낱말을 스타트업 회사를 비롯해 구글에 인수된 이후 팀 이름, 소프트웨어 플랫폼, 스마트폰 제품, 오픈 소스 코드, 누군가의 별명까지 여러 가지 의미로 빈번하게 사용했기 때문이다.

근본적인 문제는 안드로이드 팀이 그 낱말을 그런 식으로 썼다는 점이다. 그것은 회사, 구글 내부 사업부, 소프트웨어, 스마트폰, 생태

9 이 장 앞부분에서 사람들이 다른 프로젝트와 회사로 옮긴다고 이야기했다. 구글과 안드로이드도 마찬가지로 기술 세계의 다른 모든 회사 역시 그렇다. 엔지니어들은 옮겨 다닌다.

10 “구체적으로 어떻게 구현할 건가요?” 엔지니어링 토론에서 누군가가 실제로 어떻게 만들지 지루한 세부 사항을 고민하는 수렁에 빠지고 싶지 않아서 더 큰 아이디어에 집중하려고 할 때 자주 들을 수 있는 말이다. 물론 소프트웨어 프로젝트 구현과 완성은 실제로 어렵고 시간을 소모하는 부분이다. 그래도 큰 아이디어에만 집중하는 것은 표면 아래 나머지 부분 없이 빙산의 일각이 스스로 잘 떠 있다고 우기는 것과 비슷하다.

계, 팀을 의미한다.

애니메이션 <릭 앤 모티Rick and Morty>의 한 에피소드¹¹를 보면 서로 다른 행성의 주민들이 거의 상관없이 보이는 다양한 방식으로 ‘스퀼치squanch’라는 낱말을 사용한다. 결국 릭이 다음과 같이 설명한다. “스퀼치 문화는 문자적이라기보다는 맥락적이지. 내가 생각하는 스퀼치를 말하면 사람들이 알아들을 거야.”

안드로이드도 그와 같다. 그냥 여러분이 생각하는 안드로이드를 이야기하면 사람들이 알아들을 것이다.

둘째, 이야기는 시간 순서대로 진행될..., 아니 되는 편이다. 다시 말해 일어난 사건과 그 일을 한 사람들에 대해 시간 흐름에 따라 설명하는데, 복잡하고 밀접하게 연관된 일을 정리하는 데 도움이 되는 방법이 바로 시간에 따라 정리하는 것이기 때문이다. 하지만 실제로 엄격히 시간순으로 이야기하기는 불가능한데 많은 일이 동시에 일어났기 때문이다. 그래서 누군가가 1.0 또는 그 이후까지 어떤 일을 한 이야기가 진행되다가 팀의 다른 누군가의 이야기를 하기 위해 타임라인이 되돌아가는 것을 독자들은 보게 될 것이다.

시기에 관해 말하자면 이 책의 이야기는 안드로이드가 만들어지는 데서 시작해 2009년 후반까지 계속된다. 대부분의 사건은 2008년 후반 1.0 출시¹²의 서곡에 해당한다. 1.0에서 안드로이드의 미래를 가능하게 한 대부분의 요소가 자리를 잡았다. 타임라인은 다시 2009년

11 시즌 2 10화: The Wedding Squanchers

12 1.0은 소비자에게 상업적으로 공개된 첫 릴리스였을 뿐 아니라 다른 회사들이 자사만의 안드로이드 기반 기기를 만드는 데 사용할 수 있었던 첫 릴리스이기도 했다. 1.0은 처음에 G1에서만 사용할 수 있었다.

말로 연장되는데 이때가 바로 미국에서 모토로라 드로이드가 출시된 후 안드로이드의 향후 성공 가능성을 처음으로 엿볼 수 있던 때였다.

마지막으로 나는 이 책을 기술적인 세부 내용을 아는(그리고 인정할 건 인정하자면 그런 내용을 실제로 좋아하는) 소프트웨어·하드웨어 엔지니어 말고도 누구나 읽기를 바란다. 도중에 사람들이 책을 덮지 않기를 바라며 지나치게 기술적이어서 방해가 되는 것들은 피하려고 했다. 그러나 직업으로 코드를 작성하지 않는 사람들에게는 익숙하지 않을 ‘운영 체제’ 같은 용어를 쓰지 않고 운영 체제가 어떻게 만들어지는지 설명하기는 불가능했다. 책을 쓰면서 용어를 정의하려고 했는데¹³ 책을 읽는 도중에 API가 뭘지, CLI가 무슨 의미인지 막힌다면 부록 용어 해설을 확인하기 바란다.

그리고 책을 쓰기 시작했다

2017년 8월¹⁴ 나는 초기 팀과 대화를 시작했는데 다이앤 핵본과 호만 기와의 점심시간 대화 녹음¹⁵이 첫 시작이었다. 그 이후 계속해서 몇 년간 초기 팀 사람들을 대부분 인터뷰했다(대부분 직접 만났지만 때로는 이메일로).

직접 인터뷰(만나거나 화상 통화)를 하는 데 손으로 빠르게 쓰는 것만으로는 충분하지 않음을 깨닫고 나는 녹음용 마이크를 가지고

13 각주를 많이 남겼다. 독자들도 각주를 많이 보기 바란다.

14 2021년 2월 또 다른(그러나 마지막은 아닌) 긴 편집 과정의 끝에 다다랐을 즈음 이 절을 다시 읽으면서 팀이 전체 운영 체제를 만들고 1.0 제품을 출시하기까지 걸린 시간보다 안드로이드 이야기를 쓰는 데 더 긴 시간이 걸렸다는 것을 깨달았다.

15 인터뷰 고급 팁: 대화를 녹음할 때 점심 식사 중에 하지 말라. 그렇지 않으면 음식 씹으며 내는 소리를 나중에 다시 듣느라 아주 많은 시간을 쓸 것이다.

다났다.¹⁶ 한 가지 이유는 내가 생각했던 것보다 댄 샌들러와 다이엔 핵본은 말이 빨라서 거의 적을 수 없기 때문이었다. 또 대화를 녹음하면 미친 듯이 갈겨쓸 때보다 대화에 더 참여할 수 있었기 때문이다.

이 대화를 하느라 수많은 시간을 보냈다. 그런 다음 기록된 텍스트가 음성 녹음보다 나중에 참고하고 검색하기 더 쉬웠기 때문에 대화를 텍스트 문서로 필사하느라 더 많은 시간을 보냈다.¹⁷ 이 대화를 듣고 필사하고 읽느라 수많은 시간을 보내고 나서 나는 멋진 사실을 깨달았다. 대화는 책을 쓰는 데 필요한 조사가 아니라 그 자체로 책이라는 것이다. 나는 인터뷰를 큰 그림, 안드로이드 개발 연대기, 인터뷰를 하지 않았으면 내가 발견하지 못했을 세부 내용을 이해하는 배경지식으로 삼으려고 했었다. 그러나 내가 예상하지 못한 것은 모두가 자신의 말로 이야기를 아주 잘 전달했다는 것이었다.

많은 인터뷰 인용구를 이 책에 사용했다. 사실 나는 가능하면 내 설명보다 인용구를 사용했다. 그 자리에 있던 사람들이 자신의 관점으로, 자신의 목소리로, 사건의 한 장면을 포착한 것이 이야기를 가장 잘 전달하기 때문이다.

안드로이드 팀과 운영 체제가 어떻게 만들어졌는지 그 일이 일어

16 가끔 친구이자 동료인 호만 기를 데리고 갔는데 그는 초기 팀원 중 한 명이자 내가 많은 인터뷰를 준비하고 진행하는 데 도움을 주었다.

17 인터뷰 고급 팁 하나 다: 이 작업을 해야 한다면 자신의 타자 속도와 비슷한 속도로 음성을 재생해 주는 소프트웨어를 구하라. 내 경우 실제 속도의 40% 정도였는데 인터뷰하는 사람에 따라 달랐다. 이 접근 방식의 유일한 단점은 속도가 느려질수록 취한 사람들의 대화처럼 들린다는 것이다. 더 나은 팁: 기술이 따라잡기를 기다리는 것이다. 2020년 마지막 대화를 녹음했을 때 구글에서 음성을 녹음해 자동으로 필사해 주는 안드로이드 앱을 출시했다. 기술이 따라잡는 속도는 꽤 빨라서 기다려 볼 만하다. 다만 너무 늦게 따라잡아서 쓸모가 없는 경우도 있기는 하다.

나는 데 참여한 사람들의 목소리로 들려주는 이야기를 듣기 위해, 안드로이드 심장부로 떠나는 여정에 나와 함께하기 바란다.

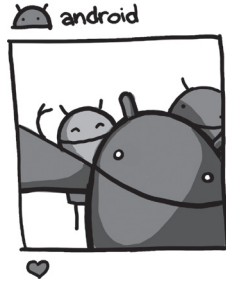
시작

“처음부터 필연 같은 건 없었어요.
안드로이드가 성공하지 못할 이유는 많았죠.
똑같은 일을 다시 이뤄 내고 싶다고 해도 할 수 없을 거예요.
뭔가 마법이 벌어진 거죠.”

- 에번 밀러



안드로이드는 원래 카메라 운영 체제였다?



“디지털 카메라용 와이파이 인터페이스가 더 나은 DSLR을 만드는 출발점이 될 거예요. 이런 제품들은 점점 강력해지고 있었는데 사용자 인터페이스가 형편없었어요.”

- 브라이언 스위트랜드

처음에 안드로이드는 포토팜FotoFarm이라는 디지털 카메라 플랫폼을 만들었다.

2003년 디지털 카메라 기술은 흥미로워지고 있었다. 고품질 렌즈에 점점 더 커지는 센서가 결합된 DSLR(Digital Single Lens Reflex: 디지털 일안 반사식 카메라)을 사용하면 디지털 이미지 파일에 훨씬 더 자세한 이미지를 포착할 수 있었다.

데인저라는 휴대 전화 제조사를 설립했다가 그 무렵 퇴사한 앤디

루빈은 새로운 프로젝트를 찾고 있었다. 그는 웹티비WebTV 시절에 함께 일한 동료였던 크리스 화이트와 함께 더 나은 카메라 소프트웨어를 만드는 회사를 시작했다. 두 사람은 2003년 후반 디지털 카메라용 운영 체제를 제공하는 포토팜을 창업하고 루빈이 CEO를, 화이트가 CTO를 맡았다. 그들은 앱을 실행할 수 있는 기능과 함께 더 나은 사용자 인터페이스와 네트워크 기능을 사용할 수 있는 소프트웨어를 구상했다. 그들은 고급 카메라 하드웨어와 결합해서 사진·이미지 기능과 경험의 경계를 확장하려고 했다.

화이트는 루빈에게 ‘포토팜’보다 더 나은 이름을 찾아야 한다고 말했다. 루빈이 android.com이라는 도메인을 이미 갖고 있어서 그들은 회사 이름을 안드로이드로 바꾸고 캐릭터Character라는 디자인 회사를 고용해 로고와 명함을 비롯한 CICorporate Identity를 만들게 했다.

그들에게 필요했던 건 안드로이드 카메라 플랫폼에 대한 그들의 비전에 투자할 사람들이었다. 그러나 아무도 카메라에는 관심이 없었고 다들 휴대 전화에 대해서만 이야기하고 싶어 했다.

루빈은 닉 시어스를 팰로앨토에 있는 안드로이드 사무실로 초대해서 카메라 운영 체제를 선보였다. 두 사람은 티모바일 사이드킥Sidekick으로 출시된 데인저의 휴대 전화 작업을 함께한 적이 있었다. 그 후 시어스는 티모바일을 그만두었지만 스마트폰 분야에서 계속 일하기로 했다. 그는 데인저에서 만들 수 있었던 것을 능가하는 소비자 스마트폰을 만들고 싶었다. 시어스는 데인저가 기대만큼 성공하지 못한 이유가 인터페이스와 기기 자체 폼 팩터form factor¹ 때문이라고 여겼다. “모두가 데인저 폰을 상징적인 기기라고 봤지만 우리는

1 (옮긴이) 휴대 기기의 폼 팩터는 기기의 크기, 모양, 부품 배치 형태를 의미하는 용어다.

사람들이 손에 들고 다니고 싶어 할 만큼 그 폼 팩터가 작지 않다는 걸 알게 됐습니다. 여전히 두꺼웠고 슬라이드 방식으로 분리된 화면을 가지고 있었거든요.”

안드로이드의 비전은 시어스에게 그다지 끌리지 않았다. 그는 카메라에 관심이 없었다. 그의 경험이 쌓여 있고 흥미가 향하는 것은 휴대 전화였다. 시어스는 루빈에게 말했다. “마음이 바뀌어서 스마트폰을 만들게 되면 전화 줘요.”

그 대화 직후 루빈은 데인저 시절 또 다른 동료였던 리치 마이너와 이야기를 나눴다. 리치는 그의 고용주인 이동 통신 회사 오렌지^{Orange}를 대표하는 데인저 초기 투자자였다. 그때 경험으로 루빈을 알게 된 후 리치는 향후 무슨 일을 할지 계속 살펴보는 중이었다.

리치도 시어스처럼 루빈의 스타트업에서 카메라 대신 휴대 전화를 만드는 걸 고려해 봐야 한다고 제안했다. 리치는 휴대 전화 시장에서 오랜 이력을 가지고 있었고 그 시장에서 변화를 가져올 기회가 안드로이드에 있다고 보았다. 화이트 역시 루빈에게 그 가능성에 대해 이야기했다. 그러나 루빈은 여전히 반대했다.

루빈은 휴대 전화 개발을 다시 하고 싶지 않았다. 그는 데인저에서 겪었던 경험 때문에 완전히 좌절된 상태였는데 그가 바랐던 만큼 이뤄지지 않았기 때문이었다. 루빈은 벤처 투자자에게 카메라 아이디어를 홍보했지만 아무런 관심도 끌어내지 못했다. 또 제조사들이 카메라를 휴대 전화에 집어넣기 시작하면서 판매가 떨어지고 있는 카메라 시장의 현실을 보게 됐다.

2004년 11월 루빈은 또 다른 벤처 투자자와 만났다. 카메라 운영

체제 홍보는 마찬가지로 관심을 끌지 못했다. 그러다 그는 휴대 전화의 가능성을 언급했는데 방 안에 있던 사람들의 귀가 쫓긋하는 것을 보았다.

루빈은 단념했다. 그는 시어스와 리치에게 다시 연락해서 스마트폰 운영 체제를 만들겠다고 말했다.

리치와 시어스가 바라던 게 바로 그것이었다. 두 사람 모두 루빈과 함께 사업 계획과 휴대 전화 운영 체제에 대한 발표 자료를 만들었다. 2005년 초 두 사람은 안드로이드에 공동 창업자로 합류했다.

루빈은 카메라 운영 체제를 만들지 못했다. 그러나 스마트폰에서 카메라가 얼마나 중요한지 감안한다면 그가 가장 널리 쓰이는 카메라 운영 체제를 만들었다고 주장해 볼 수도 있을 것이다. 단지 우회적인 방법으로 만들었을 뿐이다.

인재 양성소



“안드로이드의 멋진 점 한 가지는요. 거의 모두가 이런 일을 전에 해 봤다는 거예요. 나는 이미 실수해 봤고 그 실수로부터 배운 게 있는 일을 하고 있어요. 모두가 그렇죠.”

- 조 오노라토

기본적으로 다른 기술들도 마찬가지로이지만 안드로이드는 제품이나 출시된 릴리스라기보다는 그것을 만든 사람들 그리고 그 형태를 만드는 데 쏟은 그들의 집합적 경험이라고 봐야 한다. 그래서 안드로이드(스마트폰 운영 체제)가 어떻게 만들어졌는지에 대한 이야기는 스타트업 훨씬 이전, 즉 그 팀에서 일한 사람들의 역사를 전체적으로 살펴보는 데서 시작해야 한다.

안드로이드는 다른 많은 노력이 먼저 이뤄졌기에 탄생할 수 있었