

프로그래머의 길,  
멘토에게 묻다

# Apprenticeship Patterns

© Insight Press 2010

Authorized translation from the English edition of Apprenticeship Patterns ISBN 9780596518387

© 2010, David H. Hoover and Adewale Oshineye

This Translation is published and sold by permission of O'Reilly Media, Inc.,  
the owner of all rights to publish and sell the same.

이 책의 한국어판 저작권은 에이전시 원을 통해 저작권자와의 독점 계약으로 인사이트 출판사에 있습니다. 신저작권법에 의해 한국 내에서 보호를 받는 저작물이므로 무단전재와 무단복제를 금합니다.

## 프로그래머의 길, 멘토에게 묻다

**전자책 1쇄 발행** 2022년 5월 10일 (종이책 초판 6쇄 반영) **지은이** 데이브 후버, 애디웨일 오시나이 **옮긴이** 강중빈 **펴낸이** 한기성 **펴낸곳** (주)도서출판인사이트 **편집** 조혜정 **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](mailto:insight@insightbook.co.kr) **ISBN** 978-89-6626-359-2



# 프로그래머의 길, 멘토에게 묻다

데이브 후버 · 애디웨일 오시나이 지음 | 강중빈 옮김

인사이트



나의 가장 친한 친구이자 아내,  
우리 아이들의 어머니인 철의 여인 스테이시에게 바칩니다.

• 데이브 •



내가 컨설팅할 때 열린 마음으로 임해 준 고객과 학생들에게,  
좋았던 때와 좋지 않았던 때를 함께 보낸 소프트웨어 시절의 동료들에게,  
같이 구글에서 일하게 된 다른 TSE들에게,  
항상 이해해 주지는 않았지만 변함없이 나를 사랑해 준 친구들과 가족들에게  
이 책을 바칩니다.

• 애디 •

## 차례

---



옮긴이의 글 ... 9
추천의 글 ... 11
지은이의 글 ... 15
감사의 글 ... 26
소프트웨어 장인정신 선언 ... 35

## 1장 들어가는 글

• 소프트웨어 장인정신이란 무엇인가? .....	42
• 견습과정이란 무엇인가? .....	52
• 견습과정 패턴이란 무엇인가? .....	54
• 패턴들은 어디서 비롯되었는가? .....	55
• 여기서 이제 어디로 가는가? .....	56

## 2장 잔을 비우다

• 첫 번째 언어 .....	63
• 흰 띠를 매라 .....	74
• 열정을 드러내라 .....	81
• 구체적인 기술 .....	85
• 무지를 드러내라 .....	89
• 무지에 맞서라 .....	94
• 깊은 쪽 .....	98
• 한발 물러서라 .....	102
• 장을 마치며 .....	106

### 3장 긴 여정을 걷다

• 긴 여정 .....	112
• 예술보다 기예 .....	117
• 지속적인 동기 부여 .....	121
• 열정을 키워라 .....	127
• 자신만의 지도를 그려라 .....	131
• 직위를 지표로 이용하라 .....	137
• 전장에 머물러라 .....	140
• 또 다른 길 .....	143
• 장을 마치며 .....	146

### 4장 정확한 자기 평가

• 가장 뒤떨어진 이가 되라 .....	151
• 멘토를 찾아라 .....	158
• 마음 맞는 사람들 .....	163
• 팔꿈치를 맞대고 .....	168
• 바닥을 쓸어라 .....	173
• 장을 마치며 .....	178

### 5장 끊임없는 학습

• 능력의 폭을 넓혀라 .....	184
• 연습, 연습, 또 연습 .....	189
• 부쉬도 괜찮은 장난감 .....	194
• 소스를 활용하라 .....	200
• 일하면서 성장하라 .....	207
• 배운 것을 기록하라 .....	212
• 배운 것을 공유하라 .....	216
• 피드백 루프를 만들어라 .....	220
• 실패하는 법을 배우라 .....	225
• 장을 마치며 .....	228

## 6장 학습 과정의 구성

• 독서 목록 .....	233
• 꾸준히 읽어라 .....	237
• 고전을 공부하라 .....	239
• 더 깊이 파고들어라 .....	242
• 익숙한 도구들 .....	251
• 장을 마치며 .....	255

## 7장 맺는 글 .....

257

부록 A 패턴 목록 .....	267
부록 B 견습과정의 개설을 요청하다 .....	271
부록 C 옵티마 견습과정 프로그램의 첫 일 년을 회고하다 .....	275
부록 D 온라인 자료 .....	281

참고 문헌 ... 283

찾아보기 ... 286



연배나 성장 환경에 따라 차이는 있겠지만, 사람들이 프로그래밍에 발을 들여 놓은 계기가 된 시절의 얘기를 들어보면 비슷한 부분이 꽤나 있는 듯합니다. 애플II나 MSX, 혹은 조금 뒤의 세대라면 PC XT에서 BASIC 인터프리터를 띄워 놓고 책에서 봤던 명령문을 떠듬떠듬 타이핑해 넣은 다음 화룡점정의 기분으로 'RUN'을 칠 때의 알 수 없는 짜릿함, 그리고 이어지던 희열 혹은 좌절.

세월이 흘러 직업을 선택할 나이가 되면서, 무언가에 끌린 듯 별다른 망설임 없이 소프트웨어 개발의 세계로 그렇게들 발을 디뎠을 겁니다.

하지만 현실은 그렇게 만만하지만은 않았을 테지요. 프로그램 짜는 것만 잘 하면 되는 줄 알았는데, 그것 말고도 신경 써야 할 것들이 왜 그리도 많은지.

게다가 소프트웨어 개발이란 것이 과학은 아니고 공학이라기에는 아직 어설피고 예술이라고도 할 수 없이 어정쩡한, 아직은 그 중간의 어디쯤에 불안정하게 떠 있는 처지의 분야인지라, 많은 이들이 길을 안내해줄 누군가 혹은 그 무언가를 갈망하며 수많은 자기개발 서적과 기술자료 사이로 오늘도 서성댁니다.

IT가 3D니 4D니 하는 얘기가 그다지 새롭지 않은 지금, 머릿속에 떠돌던

로직 한 토막이 자신의 손을 거쳐서 전기 배선을 타고 실리콘의 힘을 빌려 가상세계에 현현하고, 마침내 현실의 사람들과 의미 있는 관계를 맺는 존재가 되는 그런 경험, 무엇보다도 건줄 수 없는 그 짜릿함을 떠날 수 없어서 개발자들은 오늘도 키보드 앞에 앉습니다.

이렇게 같은 길을 가는 사람들 중에서 먼저 이 길에 대해 진지하게 고민하기 시작한 이들이 있었습니다. 그 고민의 결과는 애자일, 실용주의, 소프트웨어 장인정신 같은 여러 가지 시도로 나타났고, 우리가 걷는 이 길에 의미 있는 이정표가 되고 있습니다.

이 책은 그런 맥락에서 특히 경험이 많지 않은 프로그래머들에게 길잡이가 되기를 희망합니다. 원 제목인 Apprenticeship(견습 과정 Patterns에서 보듯이, 이 책은 초심자들이 참고할 만한 여러 가지 조언을 패턴이라는 형식으로 엮어 낸 일종의 조언 모음집이라고 할 수 있습니다. 조언들은 구체적인 상황을 상정한 다음에 그럴 때 어떤 행동을 취하면 좋을지를, 먼저 그 길을 한번 지나간 멘토의 목소리로 들려줍니다.

비록 개발자라는 존재가 ‘인력’이나 ‘리소스’로 지칭되는 것이 당연시되는 불유쾌한 현실이지만, 글을 읽는 여러분이 여기 실린 조언을 발판 삼아 아직은 미숙한 우리 업계의 수준을 한 단계 높이는 데 기여할 수 있게 되기를 기원합니다.

책이 좀 더 나은 모습이 될 수 있도록 도와주신 조혜정 님을 비롯한 인사이트의 여러분께 감사하다는 말씀드리며, 일생의 반려 진영과 우리 아이들에게 사랑과 감사를 전합니다.

2010년 6월

강중빈

25년 전, 켄트 백과 나는 텍트로닉스Tektronix사 기술 센터 안의 카페테리아에 앉아서, 스몰토크-80Smalltalk-80에 접근할 수 있다는 특권<sup>1</sup>이 바깥 세상에 어떤 영향을 미칠 수 있을까 생각하고 있었습니다.

현실은 신경 쓰지 말라, 하고 내가 켄트에게 조언했지요. 만약 우리가 무엇이든 할 수 있다면, 이 지식으로 뭘 하는 것이 좋을까?

“나는 사람들이 프로그래밍에 대해서 생각하는 방식을 바꾸고 싶어.”

켄트가 말했고, 거기에 나도 동의했습니다. 우리는 우리가 몸담은 이 업계가 발전해 가면서 길을 잘못 들었다고 생각되는 부분들을 뒤집고 싶었지요. 그리고 놀랍게도, 우리는 해냈습니다.<sup>2</sup>

그때 카페테리아에서 내가 사용했던 기법, 즉 ‘현실은 신경 쓰지 말라’는 것은 일종의 패턴이었고, 대학시절 지도교수님이 쓰셨을 때 처음 보았습니다. 그때 그분은 내가 켄트에게 했던 것처럼 내게 조언해 주셨지요. 지금은

<sup>1</sup> (웁긴이) 스몰토크-80 시스템은 초기에 HP, 애플, 텍트로닉스 등 몇몇 회사에만 제한적으로 배포되었다.

<sup>2</sup> (웁긴이) 워드 커닝햄과 켄트 백은 스몰토크 시스템에서 얻은 경험을 바탕으로 소프트웨어 패턴과 익스트림 프로그래밍 같은 분야에 많은 기여를 했다.

내가 하나의 패턴이라고 인식하는 그때의 행위는, 켄트와 나로 하여금 너무 대담해 보일 수 있는 먼 목표를 감히 상상하도록 해 주었습니다. 그리고 일단 상상할 수 있게 되자, 우리 목표는 조금 더 실현 가능해 보이기 시작했습니다.

나는 그때 그렇게 생각했던 방식을 패턴이라고 부르는데, 그런 식으로 생각함으로써 우리가 종종 부딪히는 문제—스스로 우리의 야망을 검열한다는 문제—가 해결되기 때문입니다. 이 책은 다양한 문제를 해결하는 그런 장치들로 가득합니다. 우리는 패턴이 문제를 해결해 준다고 얘기하지요. 그 당시의 현실은 신경 쓰지 말라는 패턴은 켄트와 내가 가지고 있던 문제 하나를 해결했습니다. 그 패턴 덕분에 우리는 줄곧 큰 생각을 품고서 습관적인 자가 검열을 떨치고 나아갈 수 있었으니까요.

아마 당신도 현실은 신경 쓰지 말라 패턴을 시도해 본 적이 있을 겁니다. 만약 그렇지 않다면 한번 시도해 보기 바랍니다. 가장 강력한 패턴이란, 거듭해서 생산적인 방향으로 적용되는 그런 것들입니다. 패턴이 유용해지기 위해서 꼭 새로울 필요는 없습니다. 사실은 새롭지 않은 편이 더 낫지요. 이미 정리된 패턴의 이름만 알고 있다 해도 역시 큰 도움이 됩니다. 패턴을 식별함으로써 당신은 매번 똑같은 이야기를 반복하지 않고도 거기에 대해서 논의할 수 있게 됩니다.

이 책을 한번 쭉 훑어보세요. 많은 패턴이 나와 있을 것이고 상당수는 낮은 이득을 줍니다. 그중에 어떤 것이 되든 당신은 “저 패턴은 알고 있었는데”라고 할 수 있고, 아마 실제로도 그럴 겁니다. 하지만 여기에 글로 쓰인 패턴에 담긴 해결책이 행여 당신에게 상식적인 수준이라 해도, 두 가지 면에서도

움이 될 수 있습니다.

첫째로, 글로 쓰인 패턴은 더 완전합니다. 누군가 그것을 연구했고, 특징 짓고, 분류했으며, 해설까지 해 놓았기 때문입니다. 그러니 당신은 각각의 패턴마다 예상하지 못했던 금덩이를 찾아낼 수 있을 겁니다. 이 금덩이를 찬찬히 감상하세요. 그렇게 해서 이미 알고 있던 패턴이 더 강력해질 수 있습니다.

두 번째로, 패턴들은 서로 연결되어 있습니다. 각 패턴은 여러 개의 또 다른 패턴으로 이끌어 갑니다. 당신이 이미 알고 있는 패턴 하나를 발견했을 때, 거기서 아직 알지 못하는 패턴으로, 또는 함께 적용될 거라고는 전혀 생각하지 못한 다른 패턴으로 연결된 고리를 따라갈 수 있습니다.

켄트와 나는 스톨토크-80으로 개발하는 과정에서 패턴을 수집하여 상당 수를 찾아냈습니다. 우리는 그런 패턴 개념을 동료들에게 홍보했고, 조그만 변혁을 시작했습니다. 우리는 사람들이 프로그래밍에 대해 생각하는 방식을 바꿨습니다. 그 이후로 패턴과 그 사용법에 대한 책들이 수십 권 출간되었습니다.

우리가 시작한 변혁은 끝나지 않았습니다. 조금씩 성장해 가던 패턴 관련 용어 체계는 애자일 소프트웨어 개발 방법론의 기초가 되었습니다. 다시 수십 권의 책이 뒤를 이었지요.

그렇다면 이런 시점에 왜 이 책이 나왔을까요? 음, 우리 분야는 자원이 너무 풍부한 나머지 거기에 짓눌려 있다고 할 수 있습니다. 우리가 이뤄낸 변혁에서 비롯된 정보의 양이란, 한 사람이 흡수할 수 있는 수준을 넘어섭니다. 하지만 여전히 어떤 사람들은 그런 일을 해내고 있습니다. 그 사람들은 쓸모 있는 모든 조언을 내재화해서 늘 자신의 것으로 만드는 것처럼 보입니

다. 어떻게 해서 그렇게 숙달된 수준에 이를 수 있었을까요?

이 책은 복잡한 우리 분야를 마스터하기 위한 패턴들로 가득 차 있습니다. 마스터한다는 것은 그냥 아는 것과는 다릅니다. 그것은 당신의 짐을 더는 방식으로 아는 것입니다.

예를 하나 들어 볼까요? 만약 SUBSTR 함수의 인자 순서가 어땠는지 기억이 안 난다면, 인터넷에서 찾아보면 그만입니다. 인터넷이 있어 정말 다행이지요. 인터넷은 그런 식으로 우리의 짐을 얼마간 덜어 줍니다. 하지만 당신이 이 책의 패턴들을 사용하고, 작업을 개선하는 데 열린 태도로 임한다면, 자신이 그 전과는 아예 종류가 다른 코드를 작성하고 있다는 것을 깨닫게 될 겁니다. 그 코드는 인자 순서 같은 것에는 의존하지 않을 것이고 당신은 SUBSTR 너머로 날아오르는 프로그램을 짜게 될 테지요. 이렇게 된다면, 당신은 짐을 많이 덜게 되겠지요.

우리가 이룬 변혁에서 비롯된 모든 조언은, 그걸 제2의 천성처럼 만들기 전까지는 그다지 도움이 되지 않습니다. 기술을 습득하고 수련하는 것은 사람에게 천성은 아니며 그것 자체가 수련해야 할 또 하나의 기술임을, 소프트웨어 분야에서 일어난 장인정신 운동으로 말미암아 우리는 알게 되었습니다. 이 책에 실린 패턴들은 그런 뜻에서 반가운 기여라고 하겠습니다.

— 워드 커닝햄

알지 못하며 그 사실도 모르는 자, 바보로다 — 그를 멀리하라!  
알지 못하나 그 사실을 아는 자, 못 배운 자로다 — 그를 가르치라!  
알고 있으나 그 사실을 모르는 자, 잠든 자로다 — 그를 깨우라!  
알고 있으며 그 사실을 아는 자, 깨우친 자로다 — 그를 따르라!

— 이자벨 버턴 부인<sup>1</sup>(Lady Isabel Burton) (1831-1896)이  
저서 『The Life of Captain Sir Richard F. Burton』 중에서 인용한 아랍 속담

## 이 책의 목표

우리는 경험이 적은 소프트웨어 개발자들이 흔히 마주치는 딜레마에 대한 해결책을 공유하고자 이 책을 썼습니다. 기술적인 딜레마를 얘기하는 것은 아닙니다. 이 책에서 자바 디자인 패턴이나 루비 온 레일스 레시피를 찾아볼 수는 없을 것입니다. 오히려 우리가 집중하는 딜레마들은 더 개인적인 것들이며, 동기부여나 의욕에 관련되어 있습니다. 당신이 전문 소프트웨어 개발이라는 분야의 새내기로서 맞닥뜨리게 될 힘든 결정의 시기에 이 책으로 도움이 되고자 합니다.

<sup>1</sup> (옮긴이) 천일야화를 영어로 번역한 영국의 모험가이자 저술가 리처드 F. 버턴 경의 부인.

## 이 책의 독자

이 책은 소프트웨어 개발이 무엇인지 어렵게나마 알게 되면서 이제 훌륭한 소프트웨어 개발자로 성장하기를 열망하는 사람들을 위해 쓰였습니다. 당신은 웹 개발자이거나 의료기기 프로그래머일 수도 있고, 금융 기관에 쓰일 상거래 애플리케이션을 개발하고 있을 수도 있습니다. 혹은 당신의 미래를 소프트웨어에 두고서 이제 막 고등학교나 대학을 졸업한 사람일 수도 있겠지요.

이 책은 새내기를 위해 쓰기는 했지만, 얼마간 경험 있는 개발자들도 무언가 얻는 바가 있을 것입니다. 수년 정도의 경험이 있는 이들은 이미 겪어봤던 딜레마를 서술한 부분에서 고개를 끄덕일 것입니다. 그리고 그 딜레마에 대한 새로운 통찰을 얻거나, 그렇지 않더라도 최소한은 그 딜레마를 해결하고자 자기 스스로 적용해 보려는, 또는 동료들에게 제안하려는 해법을 기술할 새로운 어휘를 얻을 것입니다. 십 년 혹은 그 이상의 경험이 있는 사람들—특히 경력을 어느 방향으로 이끌어갈지 고민하는 사람들—조차도, 관리직으로 승진하라고 감미롭게 유혹하는 목소리에 맞설 수 있는 관점과 영감을 얻을 것입니다.

## 이 책이 만들어진 과정

이 책에 대한 아이디어는 2005년 초반 Stickyminds.com에서 데이브에게 소프트웨어 장인정신에 대한 컬럼을 써 달라고 의뢰해 왔을 때 싹트기 시작했습니다. 당시에는 데이브가 스스로를 (경험 있는) 견습생이라고 생각했기 때문에, 편한 마음으로 글을 쓸 수 있는 주제는 견습과정에 대한 것뿐이었습니다.