테스트주도개발 TDD 실천법과도구

채수원 tddbook@gmail.com

IG CNS에 시스템 엔지니어로 입사 후 10년 동안 다수의 프로젝트에서 다양한 역할을 수행하고 있다. 현재는 IG CNS 경영기술교육원 기술교육팀 전임강사로, 분석설계 관련 과목과 디자인 패턴, 리팩토 링 등의 과목을 강의하고 있다.

또한 보다 나은 미래를 함께 만들어 갈 수 있다는 믿음 하에 사내에 애자일 방식을 소개하고 직접 프로젝트에 참여해 실천하며 이를 바탕으로 교육을 만들어 전파하는 등 사람을 중시하는 개발 문화를 SI 현장에 확산하기 위해 노력하고 있다.

책 관련 질문 및 자료 사이트: tddbook.com

테스트 주도 개발: 고품질 쾌속개발을 위한 TDD 실천법과 도구

지은이 | 채수원

펴낸이 | 김태헌

펴낸곳 | 한빛미디어(주)

주 소 | 서울시 마포구 서교동 480-26 한빛빌딩 2층 IT출판기획부

전 화 LIT전문서팀 02)325-0984 영업1팀 02)336-7114

팩 스 | 02)336-7124

등 록 | 1999년 6월 24일 제10-1779호

초판 발행 | 2010년 6월 17일

2쇄 발행 | 2011년 1월 25일

정 가 | 30,000원

ISBN | 978-89-7914-726-1 93560

기 획 | 조연희

편 집 | 너구리 엔터테인먼트

Published by HANBIT Media, Inc. Printed in Korea

Copyright © 2010 by 채수원 & HANBIT Media, Inc.

이 책의 저작권은 채수원과 한빛미디어(주)에게 있습니다.

저작권법에 의해 보호를 받는 저작물이므로 무단 복제 및 무단 전재를 금합니다.

이 책에 대한 의견을 주시거나 오탈자 및 잘못된 내용의 수정 정보는 한빛미디어(주)의 홈페이지나 아래 이메일로 연락주십시오. 잘못된 책은 구입하신 서점을 통해 교환해 드립니다.

http://www.hanb.co.kr ask@hanb.co.kr

김창준 애자일 컨설팅 대표

반가운 TDD 수련서

채수원님을 꽤 오래 전부터 한국XP사용자모임(XPer)을 통해 뵈어 왔습니다. 겸손하고 성실한 태도, 그리고 커뮤니티와 애자일에 대한 열정을 보면서 깊은 인상을 받았습니다. 이번에 TDD에 대한 책을 직접 저술하신다고 하여 이렇게 원고를 읽고 추천사를 쓰게 되었습니다. 무엇보다 이 책이 국내 TDD 확산에 큰 도움이 될 것이라고 생각을 합니다.

이 책의 장점을 몇 가지 짚어보고 싶습니다.

첫째, 입체적이고 다채롭습니다. 전문가 인터뷰에서 TDD에 대해 경험이 많은 분들의 경험담을 접할 수 있고, 테스트들을 어떤 식으로 구조화하고 정리하는 것이 좋은지에 대한 팁도 있으며, 개발 영역에 따른(웹이나 데이터베이스 같은) TDD 작성 패턴도 소개하고 있습니다. 아마도 많은 베타리더들과 함께 "돌멩이 수프"를 만들면서 가능했던 일이 아닐까 짐작해봅니다.

둘째, 여러 가지 도구를 구체적으로 다루고 있습니다. TDD를 하면서 사용하는 도구들이 참 다양합니다. 이 책에서는 이런 도구들을 사용 예를 들어가며 단계별로 방법을 안내하고 있습니다. 뭔가 추상적인 이야기를 늘어놓는 것보다 구체적으로 각 단계에서 무엇을 해야 하는지 알고 직접 부딪혀 보는 것을 선호하는(쉽게 말해 "백문이불여일타"를 금과옥조로 믿는) 학습자에게 큰 도움이 되리라 생각합니다. TDD에 대해 충분히 감이 안 온 분들이 책을 보고 따라하면서 바로 시작할 수 있다는 장점이 되지요.

셋째, 소개된 도구들이 비교적 최신의 것들이어서 책을 읽고 따로 인터넷의 최신 문서를 공부해야 한다든지 하는 불편함이 없습니다. TDD의 도구들도 많은 발전을 해왔습니다. 국내에 출간된 몇 안 되는 TDD 관련 책들 상당수가 이런 도구의 발전 속도를 따라오지 못하고 있다고 봅니다. 이 책에서는 (또 얼마 지나지 않아 새로운 도구들이 나오긴 하겠으나) 최신 도구를 많이다루고 있습니다.

넷째, TDD를 배우는 사람 입장에서 가질 만한 궁금증들에 대한 답이 많습니다. 8장이나 9장이 특히 그렇습니다. 9장의 FAQ는 몇몇 분들에게 오아시스 같은 느낌이 들 것 같습니다. 좀 더확장되었더라면 좋지 않았을까 하는 아쉬움이 있습니다.

이제는 이 책의 독자들에게 주의점을 한 가지 말씀드리도록 하겠습니다. 도구나 술기에 너무 매몰되지 말고 고민을 하면서 수련을 즐기시라는 것입니다. 이런 점을 주의하면서 이 책을 읽는다면 훨씬 더 많은 것을 얻으시리라 생각이 듭니다.

저는 10년이 넘게 전통무술을 수련해 오고 있습니다. 그 동안 무술을 배우려는 사람 중에 술과 형에 현혹된 사람들이 간혹 보였습니다. 술은 술기이고 형은 형식입니다. 모든 무술에는 술기와 형식이 있습니다. 그리고 거기에는 일종의 직선적 단계가 존재합니다. 널리 알려진 태권도로 예를 들자면, 태극 1장하고 나면 태극 2장하고, 계속 하다보면 고려, 금강, 십진 같은 것을 배우게 되겠죠.

여기에 유혹이 존재합니다. 십진 위에 뭔가가 또 있다고 소문이 돕니다. 문주가 직접 가르친다고 합니다. 그러면 모이는 사람들이 있습니다. 자연스러운 것이죠. 하지만 그 중에는 내공이 충분하지 못하고 수련이 안된 사람들도 있습니다. 자신의 수준이 별볼일 없지만 그 "뭔가"를 배우면 나의 내공도 같은 수준으로 높아질 것이라, 뭔가 대단한 사람이 될 것이라 생각합니다.

이 부분은 착각입니다. 형을 배운다고 전문가가 되지 않습니다. 전문가가 되기 위해서는 고민과 수련의 절대적 양이 필요합니다. 수많은 고민의 시간들을 그냥 건너뛰어버릴 수는 없습니다.

- 이것이 정말 최선일까? 더 나은 방법은 없을까?
- 무엇이 워칙이지? 그 워칙을 현 상황에 적용하려면 어떻게 해야 할까?
- 불가능해 보인다. 하지만 가능하다면 어떤 모양일까?
- 왜 이렇게 고통스러울까? 더 쉽게 할 수는 없을까? 아예 이런 문제를 해결할 필요가 없게 만들려면?

내공을 키우려면 이런 고민들을 수없이 해야 합니다. 반면에 어떤 도구를 사용해야 하지? 어떤 기법을 써야 하지? 같은 고민들은 비교적 형에 가까운 경우가 많습니다. 물론 내공이 깊어지면 이런 형에 대한 고민들도 수준이 같이 높아집니다. 하지만 진짜 문제는 내공을 수런하지 않으면서 처음부터 형에 집착하는 경우입니다. 앞에서 말한, "좀 더 높은 수준의 형을 배워야지"라는 욕심 말입니다.

한편 기존에 있는 도구의 사용이 학습에 짐이 되는 경우를 보기도 했습니다. 도구의 존재(특히 그 도구가 유일하다면)가 고민의 기회를 빼앗아 버리는 경우가 있습니다. 아, 이런 도구가 이미 있구나. 이대로 쓰면 되는 거구나. 또 도구의 복잡성이 고민의 여유를 빼앗아 버리는 경

우도 있습니다. 이 도구를 어떻게 써야 할지 배우는 것 자체가 또 하나의 부담이 되어서 원래 문제를 고민할 여유가 줄어드는 것이죠(이런 것을 인지부하이론^{Cognitive Load Theory}이라고 합니다 -새로운 수학 이론을 외국어로 배운다고 생각해 보면 감이 쉽게 올 겁니다).

TDD에서 좀 구체적인 예를 들어보죠

- DbUnit이 최선일까? 장점은 무엇이고, 단점은 무엇일까?
- 나는 왜 DbUnit을 쓸까? 그 이유를 위해서 꼭 DbUnit이어야 하는가? 더 나은 방법은 없나?
- 안 쓰고 어떻게 할 수 있을까? 안 써서 더 나은 점이 있을까? 어떨 때, 어떤 방식으로 쓰면 좋을까?
- 데이터베이스 관련 개발을 할 때 어떤 원칙이 중요한가? 그 원칙을 어떻게 적용할까?
- 멀티스레드 프로그래밍을 TDD로 쉽게 하는 방법이 존재한다면, 그 방법이 무엇일까?
- 멀티스레드를 어떻게 테스트하느냐의 문제 자체를 피하는 방법은 뭘까?
- 소득세를 계산하는 코드와 그 코드에 대한 테스트 코드(부록 참고)에서 내가 못 본 차원 의 중복이 존재한다면? 어떻게 리팩토링할까? 그걸 리팩토링한 다음에 또 중복이 있다면 뭘까? 언제까지 리팩토링할 수 있을까?

이런 고민을 하는 것이 절대 쉽지는 않습니다. 만족스러운 답을 구하지 못할 수도 있습니다. 하지만 그 과정에서 많은 공부와 수련이 되고 내공이 쌓이게 됩니다. 그 여정에 도움이 될만한 팁을 드린다면, 실행전략뿐 아니라 학습전략에 대해서도 생각을 해보라는 점입니다. 인지적부하를 낮춘 상태에서 고민 그 자체에 몰두할 수 있는 환경을 만들어 여러 번 안전한 실험을 해보세요. 예컨데 멀티스레드에 대한 고민을 하려면 가장 간단한 멀티스레드 코드와 개발환경을 갖추고(각종 도구나 기법 등의 변수를 제거하고) 해보시라는 겁니다.

항상 덜 고통스럽고 유쾌한 길이 존재한다는 믿음을 가지고, TDD 수련을 즐겨보시기 바랍니다. 가시는 길에 진달래가 피었거든 가만히 앉아서 구경도 하시면서 말이죠. 꽃 구경할 시간 없다고요? 뭐가 그리 바쁘십니까! 아, 이 책도 챙겨가세요. 그 여정에 이 책이 좋은 길동무가될 겁니다.

본문을 한 페이지라도 더 읽어주세요.

감사합니다.

감사 인사

이 책은 '마법의 돌로 만든 스프'와 같은 책입니다. 시작은 제가 했지만, 많은 분들의 도움이 없었다면 결코 세상에 나올 수 없던 책입니다.

우선 다음 세 분은 공저자라 불려도 어색하지 않을 만큼 도움 주신 분들입니다.

- 많은 걸 희생하고 인내로 지켜준 사랑하는 아내 조영신
- 믿기 어려울 정도의 열정을 가진 미녀 기획자 조연희님
- 절친이자, 배움을 함께 나누는 진정한 엔지니어 이동욱

그리고 다음 분들이 아니었으면 책 수준이 지금보다 현저히 떨어졌을 겁니다.

무한 에너지의 스터디그룹 봄싹 멤버들

- 김정우님, 김지헌님, 박성철님, 박용권님, 백기선님, 변정훈님, 손찬욱님, 윤권기님, 이진서님, 임수진님, 정선아님, 정용식님

개인시간 희생하며 오프라인 리딩에 참여해서 적극 도와주신 XPer 멤버들

- 강동희님, 강성희님, 오재용님, 이덕준님, 장회수님

번거로운 부탁에도 마다 않고 도와준 두 후배

- 안경찬, 안병현

바쁜 와중에도 친절히 인터뷰에 응해주셨고 그로 인해 더 많은 경험을 함께 나눌 수 있도록 도와주신 진정한 전문가

- 김창준 대표님, 박재성님, 변신철님, 안영회님, 이일민님, 황상철님

책을 쓸 수 있을 정도의 자율적인 문화와 창의적인 환경을 팀 내에 만들고 지원해주신..

LG CNS 기술교육팀 박민수 팀장님과 김영현 차장님, 최수경 차장님, 그리고 기술교육팀 팀원 분들

사내에 애자일 문화 확산을 위해 힘쓰는 바쁜 와중에도 잊지 않고 늘 챙겨주시는.. LG CNS 경기원 차장님

Special Thanks

- 세심하게 읽고 잘못된 문장과 표현을 고쳐 주신 윤기현 선배님
- 응원과 함께 많은 조언을 해준 김성윤
- 성실히 읽고 착실하게 코멘트를 달아 책의 품질을 높여준 고종봉

그 외에도 일일이 직접 인사 드리지 못한 많은 분들께 고개 숙여 감사 인사 전합니다.

스스로에게도, 그리고 도움 주신 분들에게도 누가 되지 않도록 최선을 다했습니다.

더 좋은 소프트웨어를 만들어 좀 더 나은 세상을 만드는 데

이 책이 미약하게나마 도움이 되길 바랍니다.

감사합니다.

1. 기본 접근

이 책은 각 장을 독립적으로 읽을 수 있도록 만들었습니다. 따라서 중간에 필요한 부분부터 보더라도 크게 무리가 없습니다. 또한 중간중간 어렵다 생각되는 부분은 '중급 이상'이라는 풍선으로 표시해뒀습니다. 본인이 느끼는 난이도에 따라 적절히 넘기길 권장합니다. 각 장의 맨 앞에 나오는 체크리스트를 적극 이용해 주세요. 학습을 하지 않고 넘어간 부분이나 다시 살펴볼 필요가 있는 부분을 표시해놓고, 추후에 다시 읽어봅시다.

2. 학습 전략

우선 TDD에 대해 선수 지식이 있든 없든, 숙련자이든, 처음 접하든 1장 '테스트 주도 개발'은 꼭 읽기 바랍니다. 처음 접하는 분은 TDD의 기반지식을, 어느 정도 지식이 있는 분은 복습과 정리를 할 수 있습니다. 또한 여기서 소개하는 이클립스의 TDD 지원 기능들은 TDD 적용으로 인해 발생하는 시간지연을 최대한으로 낮출 수 있게 해줍니다.

A 코스, 성실한 개발자가 택하는 정공법, 1장부터 11장까지 착실하게 읽기

최적의 학습 방식은 1장부터 시작해서 부록까지 착실하게 읽는 것입니다. 하지만 '테스트 주도 개발' 책이 두께 적절한 '베개'로 쓰일 가능성도 있기에 체크리스트를 적절히 이용해 속도 감을 유지하도록 노력합니다. 가장 추천하고 싶은 학습 방식입니다.

B 코스, 마음 급한 개발자의 선택, 최소의 학습으로 TDD를 바로 시작하고 싶다면

1장을 읽은 다음, 2장 'JUnit과 Hamcrest'를 읽습니다. 그 다음 3장에서 프로젝트의 추천 소스 구조를 읽은 다음, 본인이 개발하려고 하는 영역을 7장 '개발 영역에 따른 TDD 작성 패턴'에서 찾아 읽습니다. 그 다음으로 10장 '실습 예제'를 연습 프로젝트로 작성해본 다음 별 문제 없다고 느끼면 실제 프로젝트에 TDD를 적용합니다. 만일 TDD로 진행하는 도중 이런저런 한계를 느낀다면 4장 '한계 돌파를 위한 노력, Mock을 이용한 TDD'나 5장 '데이터베이스 테스트' DbUnit', 6장 '단위 테스트 지원 라이브러리' Unitils'를 살펴봅니다.

C 코스, 개발자가 되고 싶진 않지만 TDD는 궁금하다면

1장을 읽습니다. 단, 실습 예제 부분은 훑고 지나가는 형태로 읽습니다. 그리고 3장 'TDD 좀 더 잘하기'의 뒷부분에 적혀 있는 TDD의 한계에 대해 읽습니다. 그 다음으로 8장 'TDD

에 대한 다양한 시각들'을 보면 소프트웨어 개발에 있어 TDD의 역학관계를 알 수 있습니다. 마지막으로 9장 'FAQ'를 보는 걸로 TDD에 대한 학습을 마무리 짓습니다. 코드를 작성하지 않고도 TDD에 대해 논의가 가능해집니다. 나머지는 필요할 때 찾아봅니다.

D 코스. TDD를 지원해야 하는 아키텍트라면

최대한 정공법으로 착실하게 1독을 권장합니다. 여건상 어려울 수도 있는데, 그렇다면 다음 장들을 좀 더 집중적으로 살펴봅니다. 지원해줘야 하는 개발자들이 TDD를 적용하는 동안 헤매지 않도록 가이드해줄 수 있습니다. 또한 자동화 기법 등을 통해 개발자들의 생산성을 높일수 있습니다.

4장 '한계 돌파를 위한 노력, Mock을 이용한 TDD' 5장 '데이터베이스 테스트: DbUnit' 6장 '단위 테스트 지원 라이브러리: Unitils' 7장 '개발 영역에 따른 TDD 작성 패턴' 12장 '테스트 자동화와 커버리지' 부록 A

E 코스_TDD에 대한 실패의 경험을 갖고 있다면

1장을 빠르게 읽은 다음 4장 '한계 돌파를 위한 노력, Mock을 이용한 TDD'를 읽습니다. 그다음 7장 '개발 영역에 따른 TDD 작성 패턴'을 살펴봅니다. 기술적인 이유로 TDD 적용이 어려웠다면, 이 부분에서 실패의 원인과 해결책을 찾을 수 있을 겁니다. 요소 기술은 알겠는데 막막한 느낌과 함께 실제 프로젝트에서 시작하기가 어려워 포기했다면, 1장 다음에 바로 10장 '실습 예제'를 보는 것도 괜찮습니다. 10장에서는 개발 시나리오로부터 어떻게 기능목록을 뽑아서 TDD에 들어갈 것인지를 가이드하고 있습니다. 만약 팀 내의 TDD가 제대로 전파되지 않아 혼자만 노력하다 지쳐버린 경험이 있다면 8장 'TDD에 대한 다양한 시각들'과 9장 'FAQ'가 도움이 될 것입니다.



강성희

티쓰리 엔터테인먼트 & 한빛소프트에서 MMORPG '삼국지천'의 개발에 참여했다. 현재는 Walk Inc.에서 아이폰 애플리케이션 개발 중



이덕준

차분한 카리스마의 소유자. 삼성SDS 국방개발팀에서 공군 C2 체계를 만들고 있다.



윤기현

개발이 무엇보다 예술적인 창조활동이라고 확신하는 고참 개발자 및 관리자빨리 퇴근하는 이상을 실현하기 위해서 불철주야(?) 노력 중



장회수

'짱가'라는 닉네임으로 SWDesign이라는 이름의 자바/패턴/아키텍처 관련 스터디를 운영 중. 블로그는 architect.tistory.com



오재용

JBoss User Group(cafe.naver.com/jbossug) 스터디에서 활동 중인 XPer 멤버 중한 사람. 현재 헬스케어 온라인 사업팀에서 근무 중



박재성

개발자가 행복하게 일할 수 있는 세상을 만드는 것이 꿈인 개발자



정용식

현자의 지혜를 꿈꾸는 개발자 현재 삼성전자 SyncThru Admin5 개발팀에서 근무 중



백기선

하이버네이트와 스프링 프레임워크에 관심이 많으며 블로그(http://whiteship.me)와 봄싹을 통해 학습한 내용을 함께 나누고 사는 개발자



박용권

산들산들 시원한 바람이 머무는 사무실을 꿈꾸는 자연주의 직당이자 즐기는 개발자들의 그룹 봄싹에서 활동하는 느림개발자~



변정훈

언제나 즐겁게 코딩하며 사는 삶을 꿈꾸는 개발자. Outsider라는 닉네임으로 활동 중. 블로그는 blog,outsider,ne.kr



김정우

'책 수집가의 기질을 가진 대체로 무해한 생물체'라고 스스로를 칭하는 미디어 서버 개발 연구원



임수진

'스루'라는 닉네임으로 활동 중인 여류 개발자



윤권기

열심히 사는 행복한(?) 개발자입니다.



이진서

세계적인 아키텍트를 꿈꾸는 대한민국 개발자



이동욱

스스로는 진지하고 조용하다고 주장하지만 지인들은 쉽게 납득하지 못한다. 현재 봄싹에서 즐겁게 스칼라 스터디 진행 중. 블로그 nephilim.tistory.com



김지헌

스스로를 '아직 배워야 할 것도 많고. 하고 싶은 것도 많은 개발자'라고 소개하는 사나이



손찬욱

다양한 SI 프로젝트를 구축 및 운영했고, 현재 LG CNS에서 SWA로 재직 중



박성철

이쁘고 똑똑한 아내와 7살짜리 아들과 함께 남양주에서 살고 있다. 책, 음악, 사진, 사람들과 프로그래밍 경험과 관련 지식을 얘기하기 좋아함



안병현

IT해방전선(?)을 꿈꾸는 엉뚱한 30대 직장인



안경찬

LG CNS에서 웹 애플리케이션 아키텍트 역할을 수행 중



김성윤

좋은 향기가 나는 코드(Good Smell Code)를 좋아하는 젊은피 개발자



강동희

인프라웨어(INFRAWARE)에서 브라우저 개발 중



고종봉

소신과 사명을 갖고 살아가는 개발자. 봄싹 및 KSUG 커뮤니티 활동에 주도적으로 참여하고 있다.



정선아

'뭣모르고 뛰어들었지만 최고의 선택을 한 것 같다'며 항상 자부심을 갖고 사는 쿨한 개발자



조연희

언젠가 저 푸른 초원 위에서 감자와 강아지를 키우고 싶어하는 독특한 소녀(?) 한빛미디어에서 좋은 책을 개발 중인 자칭 종합예술인~ 추천사 3 저자서문 6

이 책을 어떻게 읽을 것인가? 8				
베E	라 리더 소개	10		
1	테人트	주도 개발		
-		1 - 1 - 1		
		1.1 흔하디 흔한 소프트웨어 개발 방식	20	
		1,2 테스트 주도 개발(TDD) ·····	24	
		1.3 테스트 주도 개발의 목표 ·····	27	
		1.4 테스트 주도 개발의 기원 · · · · · · · · · · · · · · · · · ·	27	
		1.5 개발에 있어 테스트 주도 개발의 위치 ·····	29	
		1.6 테스트 주도 개발의 진행 방식	30	
		1.7 실습 먼저 시작해보기		
		1.8 TDD의 장점 ·····	89	
2	JUnit괴	Hamcrest		
		2,1 JUnit ·····	96	
		2.2 비교표현의 확장: Hamcrest ·····		
		2.3 정리		
		전문가 인터뷰 김창준		
•	TDD X			
3	IDD 솔	등 더 잘하기		
		3.1 테스트 케이스 클래스의 위치	155	
		3.2 테스트 메소드 작성 방식 ····		
		3.3 테스트 케이스 작성 접근 방식 ·····		
		3.4 TDD의 한계 ·····		
		전문가 인터뷰 변신철 ·····		

4	한계 돌파를 위한 노력, Mock을 이용한 TDD	
	4.1 Mock 객체 ·····	186
	4.2 Mock에 대한 기본적인 분류 개념, 테스트 더블 ·····	191
	4.3 Mock 프레임워크	215
	4.4 Mock 프레임워크 마무리	248
	전문가 인터뷰 항상철	253
5	데이터베이스 테스트: DbUnit	
	5.1 DbUnit의 장점 ······	257
	5.2 데이터셋	259
	5.3 DbUnit 데이터셋의 종류 ·····	
	5.4 DbUnit의 DB 지원 기능 ·····	
	5,5 DbUnit과 Ant ·····	276
	5.6 정리	282
	선문가 인터뷰)안영회	283
6	단위 테스트 지원 라이브러리: Unitils	
	6.1 Unitils를 사용하기 위한 환경 준비	288
	6.2 Unitils의 단위 테스트 지원 기능들·····	290
	6.3 Unitils 모듈 ····	299
	6.4 DbUnit과 함께 사용하는 데이터베이스 지원 모듈 ·····	302
	6.5 DBMaintainer: DB를 자동으로 유지보수해주는 DB 유지보수 관리자 ········	310
	6.6 기타 지원 모듈들 ····	314
	전문가 인터뷰 박재성	318

7 개발 영역에 따른 TDD 작성 패턴

	/		
		7.1 일반적인 애플리케이션	341 377 385
8	TDD에	대한 다양한 시각	
		8.1 TDD와 소프트웨어 디자인 ·····	395
		8.2 TDD 유의사항 ·····	406
		8.3 TDD와 리팩토링 ·····	409
		8.4 TDD와 짝 프로그래밍 ····	
		8.5 TDD와 심리학 ·····	
		8.6 TDD를 어렵게 만드는 요인 ·····	414
		8.7 행위 주도 개발	420
9	자주 접	估하게 되는 질문들, FAQ	
10	0 실습	예제	
		10.1 자동판매기 잔돈 계산 모듈	459
		10.2 비디오 가게	

11 테스트 자동화와 커버리지		
	11.1 테스트 케이스 수행 자동화: Ant 491 11.2 테스트 커버리지 510	
부록 A 본	- 택보다 더 가치 있을 수도 있는 '부록'	
	A.1 놓치기엔 아까운 JUnit 4의 고급 기능들 ····· 529	
	A.2 스프링으로 JUnit 기능 확장하기 ······ 553	
	A.3 코드 리뷰 ······ 556	
	A.4 3분 안에 Java DB 설치하고 확인까지 마치기 573	
	A.5 TDD에 대한 연구 보고서(테스트 주도 개발을 통한 품질 향상 실체화)······ 577	
	A.6 TDD 학습에 도움이 되는 책과 문헌 579	
찾아보기 …		