

推薦の言葉

本書はアメリカはいうにおよばず世界を代表するソフトウェア専門のコンサルティング会社である ThoughtWorks 社のコンサルタントの皆さんのエッセイの集合体です。実は私の関係する会社も ThoughtWorks とまったく同じ「オブジェクト指向とプロセス」という基本技術によって「ソフトウェア開発変革のお手伝いを実践的に行う」目的の会社として立ち上げられている点（残念ながら ThoughtWorks の創業は 1995 年で豆蔵はそれに遅れること 4 年）でも非常に親近感を抱きます。

最初に ThoughtWorks のことを知ったのは私が『UML モデリングのエッセンス』として翻訳させていただいている UML Distilled の著者でありそれまで独立系コンサルタントであった Martin Fowler 氏が ThoughtWorks という聞いたことのない小さなコンサル会社のチーフサイエンティストになったらしいと 2000 年の OOPSLA 会場で米国の知人から聞かされたときでした。そのときの印象は「思考の仕事」社という名称をすばらしいと思いやられたと感じたのを覚えています（こう感じたのは形容詞を名前に取り入れた「合理的なソフトウェア」社のとき以来です）。

まさに本書の内容は「思考の仕事」の結晶であり、どのエッセイも個性的なオリジナルのスタイルで書かれていますが、Fowler 氏に代表される「理論と実践のバランス」がとれた practical な態度というものが一貫して感じられて快く、納得感をもって読まされてしまうものばかりです。

たぶん最初に置かれた創業者兼会長の Roy 氏の『ビジネスソフトウェアのラストマイル』エッセイが全体の基調をなすように読者に ThoughtWorks 社の存在理由を呼びかけるような形で仕組まれているのですが、ここからは今後われわれ IT 業界に身をおくものとして何を解決していかなければならないのか、どのようなスタンスで今後のソフトウェアビジネスに関わっていくべきなのか、考えるきっかけを与えてくれます。その後に配置された他のエッセイもプログラマー・アナリスト・マネージャ・

テスターと異なる立場からこの「ラストマイル」問題への実践的な取り組みの戦果報告があくまでも中間報告としてなされています。自分がどのエッセイを読むべきかは、副社長 Mike 氏の「まえがき」に的確な要約があるのでぜひ参考にして下さい。

このようなさまざまな立場・視点・スタイルのエッセイの翻訳に当たられたオブジェクトの広場関係者のみなさんは大変だったのではないかと思います。今回はわれわれが読者から執筆者にならなければならないと強く思います。私の属する豆蔵も含めて日本からこのようなビジネス視点と技術視点がうまく織り込まれた「読むに値する」アンソロジーが発表できるようにぜひがんばろうという強い気持ちをいだきつつ、推薦の辞といたします。

2008年11月13日
株式会社豆蔵 取締役
プロフェッショナル・フェロー
羽生田 栄一

賞賛の声

本書では、技術の深さ、詳細さ、斬新なアイデアや研究成果の量は各エッセイで異なります。しかし、すべてのエッセイはきわめて実践的だという点で共通しています。著者らは、「考えさせられる」にもかかわらず「すぐさま適用できる」という、長く目にしていない偉大な仕事を成し遂げています。

— Stefan Tilkov (innoQ CEO)

IT 業界では長い間、カスタムソフトウェアの開発はとても難しく、コストがかかると信じられてきました。このアンソロジーでは、あえてそれに取り組んでいる企業が有する、多様な視点と観点を垣間見ることができます。

— W. James Fischer (Accenture 元 CTO / シニアパートナー (引退))

例えば CruiseControl のようにとても成功したオープンソースのプロジェクトから、ブログやカンファレンスで公開されたアイデアまで、おそらく開発の現場で ThoughtWorks 社の影響力を実感したことがあるでしょう。とはいえ、外野の私たちには、ThoughtWorks 社の中でどのような議論が行われているのかは知る由もありませんでした。しかし、本書を読めば、カーテンを開けて議論に参加することができます。一段上の開発者になれるのです。

— Nathaniel T. Schutta (著述家、スピーカー、講師)

ソフトウェア開発はいろいろな意味で団体競技であり、リーダーがソフトウェア文化を形成します。成功している組織はそうした文化をたいていドキュメント化しないため、他の人はそれから何も得られません。ここに、ThoughtWorks 社のリーダーたちのエッセイがあります。この興味深いエッセイ集を読むと、ThoughtWorks 社

の文化をひと目見ることができます。

— Dave Thomas (Bedarra Research Labs)

ソフトウェア開発に関する最高の知見は、現実に関客のために問題を解決している人々からもたらされます。しかし、散在するブログをくまなく探索でもしないかぎり、そうした知見を手に入れるのはほぼ不可能でした。ThoughtWorker たちは、過去 10 年にわたり現実にある多くの問題を解決してきました。だからこそ、そこから得た英知の結集のスナップショットをこの手にして、私は本当にうれしく思います。

— Gregor Hohpe (“*Enterprise Integration Patterns*” 共著者)

今日の厳しい IT 業界においてソフトウェアを開発するための、言語やツールの正しい使い方を取り上げた、実に素晴らしいエッセイ集です。本書の著者らこそが、ソフトウェア世界のベテラン中のベテランと呼ぶにふさわしい人たちです。

— Terence Parr (サンフランシスコ大学 ANTLR プロジェクトリーダー)

ThoughtWorks 社は傑出した仕事をしました。エッセイ集をまとめ、定評ある経験と知恵の一部を私たちに提供してくれたのです。本書は、たびたび引き合いに出され、どのプロジェクトの本棚にも並べられるような書籍です。

— Jeff Brown (G2One 北米担当役員)

まえがき

ThoughtWorks社は、カスタムアプリケーションと妥協のないコンサルティングを提供する、情熱的で意欲を持った、知的な人々からなる集団です。Thought Workerに、この会社のどこが一番気に入っているかを尋ねてみてください。彼らは、他のThought Workerと出会い、一緒に働き、彼らから何かを学べるのだ、と答えるでしょう。私たちは、さまざまな文化、民族、教育のバックグラウンドを持ったギーク、マネージャ、アナリスト、プログラマ、テスト担当者、運用技術者で構成されています。こうしたバックグラウンドと視点の多様さとともに、私たちは新しいアイデアに対する情熱を共有しています。その2つが結びつくことで、非常に活発な議論が生み出されています。

私たちの作り上げた会社は、6か国に散らばり、1,000人近くの聡明で独自の意見を持った人々からなります。組織構造にはほとんどヒエラルキーがなく、そして熱狂的なまでに透明性にこだわっています[†]。私たちの会社は成功を取っています。もちろん、私たちが定義する成功とは、典型的な意味での成功とは異なります。成功とは、顧客を満足させ、ソフトウェア業界に影響を与え、そして社会にも影響を与えるものでなければいけません。私たちは、高い目標を掲げているのです。

ブログスフィアやカンファレンス会場、Web上、書籍で、多くのThought Workerの声を聞くことができます。それというのも、これまでの自分たちの成果や手法について容赦なく批評することも、最高のものを目指そうとする私たちの姿

[†] 訳注: ThoughtWorks創業者のRoy Singhamは、CNNの取材に対して「中国オフィスの清掃員がCEOと同じくらい戦略的に価値があるような、フラットな会社を作りたい」と答えている。また、会社の透明性については、「全従業員の給料を公開することを検討したことがある」とも答えている (http://money.cnn.com/2008/03/14/technology/kirkpatrick_thoughtworks.fortune/index.htmより)。

勢の一部だからです。そうした批評によって、次なる改善を探究しているのです。私たちは、簡単には満足しようとしません。そして自分たちが学んだことは、ぜひ他の人にも伝えたいと思っています。

私たちの歴戦の経験は、ドメイン、テクノロジー、プラットフォームの異なる無数のプロジェクトから得たものです。私たちは自分たちの仕事について本当にたくさんのことを考えますが、その考えは数多くのソフトウェアを人々に提供してきた開発現場に根ざしたものです。ソフトウェア開発の仕事に対する純粋な思いから、私たちはこれまでソフトウェア開発に集中してきたのです。

IT コンサルタントは、新しい人事管理ポリシーの検討ミーティングに出席するために雇われるわけではありません。したがって、ほとんどの IT プロフェッショナルと比較して、私たちの仕事ははるかにソフトウェアの提供に関することに焦点が絞られています。その結果が、現実主義と技術的正確さを兼ね備えた私たちの姿勢につながっています。

このアンソロジーは、ThoughtWorker が取り組んでいる信じられないほど多様な IT の諸問題についての、優れたスナップショットを紹介するものです。私たちはこのアンソロジーを、単によりよいソフトウェアの作成方法を紹介するだけの書籍でなく、それ以上の内容にしようと思えました。組織の IT 投資に見合った、真のビジネス価値を実現するための課題に取り組んでいます。このアンソロジーは、基調テーマをなす Roy のエッセイにより幕を開けます。Roy は、総力を結集して、システムを稼働環境へ移行する「ラストマイル」に変化をもたらそうと呼び掛けます。Roy の計画は壮大で野心的です。それは、「ラストマイル」における運用やデプロイの課題を、要求定義やコーディングと同じように開発プロセスの中心課題に据えることに他なりません。プロジェクトの成功とは何かを思い出してください。単に、開発したコードが品質保証 (QA: Quality Assurance) 部門の審査を通り、壁の向こう側にいる運用チームに投げ渡されて、本番運用やデプロイなどをやってもらう用意ができた、ということではないはずです。それを思い出せば、ソフトウェアが稼働するのを見届けるまでは、ソフトウェア構築チームの仕事は「完了」していません。そして、Roy の提唱は、プロジェクトの完了と成功の意味を鋭い切り口で定義し直すだけにとどまりません。彼は、利害関係者に参加してもらう方法とタイミングについても再考を求めるのです。これまでコーディングプロセスのツール (例えば自動ビルド、自動テスト、リファクタリングのツール) を改善するために考え出されてきた優れたノウハウは、「ラストマイル」問題に立ち向かう際にも応用できます。

本書を読み進めるにつれて、Roy の呼び掛けに対する返答に繰り返し出会うで

しょう。例えば、James は性能テストで返答します。性能テストは、プロジェクトの後半になるまで忘れられ、先延ばしにされるのが常態化しています。プロジェクトも後半になれば、非常に多くの設計判断がコード中に作り込まれてしまっています。性能上の問題を解決するために、苦勞して作り上げた業務機能を壊さずにそうした設計判断を元に戻そうとするのは、まるで熱力学第二法則[†]に逆らおうとするようなものです。James は、きわめて実践的なアプローチを採用します。前もって性能要求を定義しておくことが必要だ、などと当たり前な主張をする代わりに（それをわかっていない人などいるのでしょうか）、利害関係者から有用な性能要求を引き出す方法について議論します。単に「テストを早くから開始せよ！」などと言う代わりに、実際には、どこで、どうやって性能テストを実施できるかを検討します。

Julian は Ant リファクタリングで返答します。多数の標準的なリファクタリングをカタログ化し、それぞれのリファクタリングについて明快な例を提示します。Julian のエッセイは、日々成長し、進化するビルドスクリプトを扱っている人にとって、優れた参考書となります。Dave のエッセイは、Roy による巻頭のエッセイと対称をなす、素晴らしい巻末の結びを提供します。Dave は 1 クリックデプロイについて、その概念的枠組みの概略を示します。彼は、ビルドされた巨大で扱いにくいバイナリをいかに管理するか、多種混在するデプロイ環境の中でいかにシステムを統合していくかといった、大きな問題に取り組みます。ビジネスソフトウェア開発を効率化するテクニックはすべて、最終的にはデプロイツールの中に組み込まれるでしょう。Dave のエッセイは、それをいち早く実行に移すものです。

Stelios のエッセイは、プロジェクトの健康状態を伝えるコミュニケーション技術で返答します。彼はプロジェクトのメトリクスを、客観的なものから主観的なものまで、いくつか提案します。そして、関係者全員が毎日同じ「ダッシュボード」^{††}を見ながら作業できるように、メトリクスを効果的に表示する方法を議論します。彼は、プロジェクトの生体情報を、可能なかぎり多くの利害関係者に見えるようにしようとしています。この試みは、もう 1 つの概念へとつながります。プロジェクト人類学でも言うべき概念です。Tiffany のエッセイは、まるでサモアでの発見

† 訳注：いわゆるエントロピー増大の法則で、熱は高温から低温へしか移らないという法則。閉じた系は、必ず秩序状態から無秩序状態に変化していくことを示す。

†† 訳注：ダッシュボードとは、本来は自動車の運転席前面のスピードメーターなどの計器類が配置されている場所をいう。ダッシュボードが自動車の状態をひと目で把握できることから、一般にものごとの状態を視覚的に把握するためのツールの意味にもなる。なお、7章では「情報発信器」という用語が使われる。

を報告するマーガレット・ミード[†]のようです。イテレーションマネージャというまったく新種のプロジェクトチームメンバと遭遇した彼女は、そのメンバがどのようにチームという種族の中に組み入れられていくのかを教えてください。彼女は、チームの組織方法を少し変えるだけで、チームの活動がもっと効果になる可能性を見出します。そして、新しい方法で組織されたチームの働きを助ける、新たな役割を私たちに示すのです。Jeff による「9つのルール」のエッセイは、プログラミング道の師範が弟子にその真髄を教える様子を想起させます。そのルールは簡潔で優雅でありながら、同時に苛立たしいほどに従うことが困難です（それは特に、ルールに従うにはプログラマが非常に多くの習慣を「忘却」しなければいけないからです）。Rebecca のエッセイは、「言語論争」の問題に確固たる姿勢で歩み寄っていくかのようです。エッセイは、さまざまな言語を分類していく魅力的な読み物として始まります。初めのうちは、リンネ[‡]が庭園をそぞろ歩きしながら、植物を眺めてその特性を調べ、そして彼がいずれ考え出すことになる植物の分類体系へと一般化させていくかのような印象を受けます。Rebecca は、言語論争に対する優れた議論の基礎を打ち立てます。しかし、驚くのは最後です。このエッセイは、現在人気の言語をいくつか紹介する単なる概説講座ではありません。そうではなく、世の中にあるプログラミング言語がいかに多様かを示し、巷の「Java 対 .NET」言語論争は永遠に決着のつかない議論を繰り返しているだけにすぎないことを証明するのです。しかし重要なことは、どんな種類の問題を解こうとしているのか、そして問題に取り組むためにどんな種類のプログラミング言語を利用できるのか、を知ることです。Rebecca の文章はまるで、散らかった工房に入ってレンチや金づちをきれいに選り分けていき、その用途がわかるようにラベルを貼って引き出しの中へ整理していくかのようです。

残りのエッセイはより技術的な話題になりますが、ここでも ThoughtWorks の才能あふれる同僚たちがその多様性をいかんなく発揮しています。Ian は、顧客ではなく利用者駆動による SOA の契約について検討するための、包括的なアプローチを説明します。変化するビジネス要求に長期にわたり適応しなければいけない共有サービスをいかに構築し、進化させていくか、特にそれを既存のサービス利用者には不自由を感じさせることなく行うにはどうすればいいか、というのは永遠の課題で

† 訳注：マーガレット・ミード (Margaret Mead) は米国の文化人類学者で、ジェンダー研究の先駆者として知られる。サモア諸島でのフィールドワークをまとめた著書『サモアの思春期』（蒼樹書房）が有名。

‡ 訳注：カール・フォン・リンネ (Carl von Linné) は18世紀の植物学者。界・門・綱から始まる近代的な生物の分類体系の基礎を作り、「分類学の父」と呼ばれる。

す。彼のエッセイは、その課題に一石を投じるものです。そして Erik も、同様の問題を検討します。よく設計されたシステムでは、ドメインモデルはインフラストラクチャレイヤから分離されます。しかしそのためには、インフラストラクチャレイヤがドメインモデルの中のメタデータを活用しなくてはなりません。プログラム上の情報、例えばドメイン要素を表現するために選んだクラスの種類といったものから、暗黙的なメタデータをかき集めてくることも可能です。しかしそれだけでは、バリデーションのような本当に役立つことを行うのに十分な情報を提供できません。Java や C# といった最近の言語では、アノテーションや属性といった形でより潤沢なメタデータを提供することができます。Erik はケーススタディを通して、そうした最新のメタデータを活用する方法を探究します。そして Martin のエッセイは、邪悪な誇大妄想者のためにさまざまな DSL を駆使するという遊び心溢れるものですが、私は自分がかつて C を学び始めた頃のことを思い出しました。当時の私の参考書は K&R (Kernighan & Ritchie)[†]でした。K&R の中で、文字列コピーの関数が数回のイテレーションを経て簡潔で優雅な形へ書き替えられていくのを読み、私は感動しました。K&Rのおかげで、これから直面しそうなプログラミング上の障害がすべて解決されたかのように思われたものです。

エッセイとエッセイをつなぐ糸は、アンソロジーの至るところに張り巡らされています。これらのエッセイは、IT 問題の生態系を深く探索しながらも、同時に、一見して明らかな形や予想もしない驚くべき形で互いに結びついているのです。本書が幅広い話題や多様な問題解決のアプローチに満ちていることは、執筆陣が所属する組織がアイデアを創出する健全な環境であることを反映するものです。私たちが生み出したアイデアの一端がこのような選集の形になるのを目にしながら、自分たちの可能性をもっと追究してみたいという思いに、私は強く駆られています。

2008 年 2 月 15 日

Mike Aguilar (ThoughtWorks 副社長)

[†] 訳注: 邦訳は『プログラミング言語 C ANSI 規格準拠 第2版』(共立出版)。C 言語の設計者である Brian W. Kernighan と Dennis M. Ritchie によって書かれた、C 言語の古典的な定番書籍。

本書の表記

本書では、以下の表記を使用しています。

太字 (Bold)

重要な用語を示します。

等幅 (Constant Width)

サンプルコード、コマンド、変数、属性、関数、クラス、名前空間、メソッド、モジュール、値、ファイルの内容、コマンドの出力などを示します。

等幅の斜体 (*Constant Width Italic*)

ユーザが各自の環境に応じて入力する必要がある要素部分を示します。

等幅の太字 (**Constant Width Bold**)

コードの重要な部分と、そのとおりに打ち込まなければならないコマンドやテキストを示します。

サンプルコード

サンプルコードは次のサイトからダウンロードできます。

http://www.pragprog.com/titles/twa/source_code (原書)

意見と質問

本書（日本語翻訳版）の内容については、最大限の努力をもって検証および確認していますが、誤りや不正確な点、誤解や混乱を招くような表現、単純な誤植に気づかれることもあるでしょう。本書を読んで気づいたことは、今後の版で改善できるように知らせていただければ幸いです。将来の改訂に関する提案なども歓迎します。

株式会社オライリー・ジャパン

〒160-0002 東京都新宿区坂町 26 番地 27 インテリジェントプラザビル 1F

電話 03-3356-5227

FAX 03-3356-5261

電子メール japan@oreilly.co.jp

本書に関する技術的な質問や意見については、次の宛先に電子メール(英文)を送ってください。

info@nostarch.com

本書の Web ページには、正誤表、サンプルコード、追加情報が掲載されています。以下のアドレスでアクセスできます。

<http://www.oreilly.co.jp/books/9784873113890/>

<http://www.ogis-ri.co.jp/otc/hiroba/ogisbooks/TWAja/>

<http://oreilly.com/catalog/9781934356142/> (原書)

<http://www.pragprog.com/titles/twa/thoughtworks-anthology> (原書)

オライリーに関するその他の情報については、次のオライリーの Web サイトを参照してください。

<http://www.oreilly.co.jp>