

## ジム・ハイスマスによる序文

「スクラムには華麗に騙される。フレームワークの中でも一番理解しやすい部類なのに、ちゃんと実現するのは最大級に難しい」。この言葉から始まる本書は、スクラムのための示唆に富んだ価値あるガイドだ。私が見てきた多くの組織はスクラムのシンプルさ故に初步でつまづき、スクラムに熟達できない。そうした組織は「ルール通りの」アジャイルを導入しようとしており、自らの矛盾に無自覚でいる。また変化の難しさ、特に大規模組織における変化がいかに困難で、本気で取り組んでも厳しい道のりになることに気づいておらず、シンプルなルールだけでは達成できないことも知らない。本書はガイドとして、スクラムの初歩から熟達に向け、現実的な方法を指導してくれる。スクラムフレームワークの基本ではなく（付録は除いてだが）、より高度で実践的な、スクラムフレームワークをあなた自身とあなたのチームで機能させるためのガイドだ。

アジャイルへの移行で見落とされることが多い、2つの重要な問題がある。リリースプランニングと技術プラクティスだ。この2つはスクラムで（他のフレームワークでも）始めるのに欠かせない。ミッチは技術プラクティスの重要性をはじめから強調しており、スクラムの効果的な導入に必須の要素だと説いている。ミッチが指摘するように、毎スプリント出荷可能なソフトウェアを実現するというゴールは、技術プラクティスなしで達成できない。彼の提示する基礎的なプラクティス——テスト駆動開発、リファクタリング、継続的インテグレーション、頻繁なチェックイン、ペアプログラミング、受け入れテストの自動化とインテグレーション——これらは技術プラクティスの導入にふさわしい。

私は11章「リリースプランニング」の物語を読んで、声を出して笑った（どの章にも導入となる物語があり、そこでとりあげる問題を紹介している）。スティーブンは上級マネージャで、プロジェクト完了の見通しをマネジメント層に報告しなくてはならない。アジャイルなリーダーが大事にすべき考え方のひとつに、私が「両方（And）」マネジメントと呼んでいるものがある。両方マネジメントとは、2つの対立するフォースに共通点を見つける能力だ。スクラムプロジェクトで典型的なパラドックスが、「予測可能性」と「適応性」の対立だ。伝統的なやり方を重んじる派閥は予測可能性を重視するが、アジャイルな派閥（の一部）は適応性に片寄る。鍵となるのは、当然、両方のバランスを取り、適切なレベルを見つける考え方だ。ミッチはこの章の物語で、このパラドックスを「両方」マネジメントでどう扱うのか、素晴らしいガイドラインを示し

てくれた。

最近、同僚と話していたとき、スクラムの実践の初期では2つの要素が重要だと言っていた。学びと、素早く成果を上げることだ。ミッチはやはり、この2つについて2章「仲間と共に旅立つには」で、その重要性について述べている。ここで彼が書いているのは、変化のマネジメントと、学ぶ能力と適応する能力とを成長させながらスクラムの移行を進めることについてだ。素早く勝つことについても、ジョン・コッターの変化のマネジメントについて紹介する中で述べている。

本書のもうひとつの特長は、章の一つ一つが短く、ひとつのトピックに絞って、基礎的スクラムフレームワークを鍵となるプラクティスによって機能するフレームワークに変換する方法をまとめている点にある。トピックは多岐に渡る——スクラムの価値、役割の定義、ペロシティの計算、スプリントの長さの決め方、ストーリーの分解、顧客レビューの進め方などだ。また素晴らしいのが、「完成」が何を意味するのか定義することについて書いた7章だ。スクラムプロジェクトでは必須の話題だ。

スクラムを導入している人であれば誰でも、あるいはそれ以外のアジャイル手法であっても、ミッチが書いた本書はあなたの助けとなり、華麗なシンプルさから効果ある実践的な姿への移行を導いてくれる。困難な問題がなくなりはしないかもしれないが、困難な問題についてしっかり理解できるようになるはずだ。

—— Jim Highsmith  
Executive Consultant, ThoughtWorks

ミッチと私は長年、スクラムの開発者トレーニングで一緒に働いてきた。本書を学べば過去10年間起き続けてきた多くの問題へ対処するのに役立つだろう。この10年で、世界中でアジャイル手法はソフトウェア開発の標準になってきた(そのうち75%はスクラムだ)。

『アジャイルマニフェスト』の公開から10年後、元の起草者と多くのアジャイルの思想的リーダーが、ユタ州スノーバードに集まった。今回の集まりは、10年間のアジャイルなソフトウェア開発を振り返るためだ。全員でアジャイルなプロダクト開発の成功を祝い、その成功を発展させるまでのいくつかの重要な障害をレビューした。そして最終的に、4つの要素がこれから10年間の成功に重要であると合意した。

1. 技術的卓越の追求
2. 個人の変化の支援と、組織の変化の推進
3. 知識の整理と教育の改善
4. プロセス全体における価値創造の最大化

ミッチの本が、あなたがアジャイルリーダーになる手助けになることを示そう。

## 技術的卓越の追求

---

インターネットやスマートフォンアプリケーションの爆発で重要な役割を果たしたのが、エンドユーザーから短いインクリメントで素早くフィードバックを得る手法だ。これをアジャイルの中で定式化したのが、開発をおこなう短いスプリントだ。長くても1ヶ月、多くは2週間の期間である。この問題をアジャイルマニフェストでは「包括的なドキュメントよりも動くソフトウェアにより価値をおく」と表現した。

10年後のアジャイルマニフェストの振り返りでは、大多数のアジャイルチームが今でも短いスプリントの開発に困難を感じているのがわかった(理由の多くはマネジメント、ビジネス、顧客、開発チームらが技術的卓越を要求しないためだ)。

エンジニアリングプラクティスはソフトウェア開発の基礎となり、17%のスクラムチームはスクラムにXPのエンジニアリングプラクティスを併用している。最初のスクラムチームも同じことを、1993年、XPが生まれる前からやっている。プロフェッショナルのエンジニアにとっては常識に過ぎないのだ。

ミッチは最初の章で、いくつかのXPプラクティスは必須だと述べている——持続可能なペース、コードの共同所有、ペアプログラミング、テスト駆動開発、継続的インテグレーション、コーディング標準、リファクタリング。いずれも技術的卓越の基礎となるもので、61%のアジャイルチームはスクラムをやりながら、こうしたプラクティスを導入していない。そうしたチームはミッチによる本書をよく研究し、彼のガイドに従うべきだ。それこそが、スプリントで出荷可能なコードを完成できない理由なのだ！

ミッチの本には他にも技術的卓越に繋がるガイダンスがあり、アジャイルリーダーはマネジメントの人間であれエンジニアリングの人間であれ、技術的卓越を要求し追求するべきだと、ミッチは明快に示している。

## 個人の変化の支援と、組織の変化の推進

アジャイルへの移行では技術的卓越と共に、変化する要求に対応し続けるよう求められる。アジャイルマニフェストでは、4番目の価値だ。「計画に従うことよりも変化への対応に価値をおく」。だが個人が変化に適応するだけでは、十分ではない。組織もまたアジャイルに変化に対応できるよう、構造を変えねばならない。さもないと、チームがハイパフォーマンスを出せるようになる邪魔となったり、破壊してしまうこともある。進歩の障害となるものを取り除けないためだ。

ミッチはハーバードビジネススクールの変化を成功させる要素を順を追って紹介している。危機意識を生み出さなくてはならない。そうでなくては変化は不可能だ。アジャイルリーダーはそれに慣れる必要がある。変革を進めるための連帯も欠かせない。アジャイルリーダーはマネジメントが間違いなく、スクラム導入に向けた教育とトレーニングを受け、共有し参加するよう仕向ける義務がある。

ビジョンを作り人の自発を促すのも基礎の一部だ。現場を無視した判断や、コマンドアンドコントロールの徹底があれば、アジャイルのパフォーマンスを殺してしまう。アジャイルリーダーはこうした災厄を避けるため、短期間成果を出し、変革を進め、障害を取り除き、新しいアプローチを文化の一部として確立しなくてはならない。アジャイルリーダーはエンジニアリングのみならず、マネジメント組織の一部になるか、マネジメントをトレーニングしなくてはならない。ミッチによる本書を読めば、何をすべきか、どうやって実現するか理解できる。

## 知識の整理と教育の改善

---

チームの生産性に関する膨大な知識が存在するものの、多くのマネージャや開発者がその存在を知らない。ミッチはそうした問題に本書全体で触れている。

### ソフトウェア開発は本質的に予測不可能である

---

「ソフトウェア開発は予測不能だ」という「ジブの法則」を知る人は少ない。世界中で数多くのプロジェクトが失敗している理由は、主にこの問題に対する無知であり、適切な対策を取っていないせいだ。ミッチは検査と適応により変わり続ける重要性を説いている。本書で述べられた戦略を使えば、多くの落とし穴を避けられ、スクラムの導入に向けた障害物も取り除ける。

### ユーザーは欲しいものが分からず、動作するソフトウェアを見て初めて分かる

---

伝統的なプロジェクトマネジメントでは、ユーザーは欲しいものをあらかじめ分かっていると、誤って認識している。この問題は「ハンフリーの法則」で定式化されているが、大学でも企業でもマネージャやプロジェクトリーダーにこの法則を教えていない。この問題に取り組み、現実に目を開くのに本書は役立つ。

### 組織構造はコードに埋め込まれている

---

あまり知られていない重大な問題の3番目が「コンウェイの法則」だ。組織の構造がコードに反映される。伝統的な階層構造はオブジェクト指向設計に悪影響を及ぼし、結果として壊れやすいコード、悪いアーキテクチャ、貧弱な保守性や変更可能性に繋がり、コスト過多や高失陥率の原因となる。ミッチはスクラムの組織を正しく作る方法について時間をかけて説明している。しっかり耳を傾けよう。

## プロセス全体における価値創造の最大化

---

アジャイルプラクティスはソフトウェア開発チームの生産性を2倍にも3倍にも、容易に引き上げるが、それはプロダクトバックログの準備が出来ており、スプリント終了時にソフトウェアが完成できる場合だ。生産性の向上は組織の他の部分の問題を引き起こす。組織の他の部分がアジャイルでないことが明らかになり、そこに痛みが生じるのだ。

## 運用とインフラにおけるアジャイルさの不足

---

能力ある人員をプロダクトバックログ改善にアサインすれば、ソフトウェアを開発する速度は少なくとも2倍になる。5~10倍になる場合もある。その結果開発の運用とインフラがボトルネックとなっていると明らかになり、直さねばならない。

## マネジメント、営業、マーケティング、プロダクトマネジメントにおけるアジャイルさの不足

---

プロセスの先頭部分では、ビジネスのゴール、戦略、目標が明確でないことが多い。その結果、たとえソフトウェアの生産性が倍になっても、収入のストリームが向上せず、悪化する場合もある。

そのため組織の全員が教育とトレーニングを受け、バリューストリーム全体でパフォーマンスを最適化する方法を理解しなくてはならない。アジャイルな個人はこうした教育のプロセスを先導するため、知識を整理し、組織全体をトレーニングする能力も改善していかねばならない。

## 結論

---

多くのスクラム導入では小規模な改善しか起きず、障害を取り除くのに常に苦労し続けることになる。仕事をもっと良いものにできる。どんなチームも優秀になれ、多くのチームが偉大になれるのだ！仕事を楽しむことも、ビジネスが利益を生むことも、顧客が本当に幸せになることもできるのだ！

あなたが始めたばかりならば、ミッチの本が役に立つだろう。あなたがいま苦労しているところなら、本書はもっと役に立つ。すでに良い結果を出しているなら、ミッチの助けでさらに素晴らしい結果を出せる。改善は決して終わることはなく、ミッチの洞察は真に有用だ。

—Jeff Sutherland  
Scrum Inc.

僕の娘、エマが2004年に産まれると、僕は途方に暮れてしまった。他の子供たちのときと比べて、ずっと頻繁に医者に通わなくてはならず、僕は妻に「これって普通なのか？」と聞き続けていた。ある日、僕の枕の上に『すべてがわかる妊娠と出産の本(原題: What to Expect the First Year = 1年目にどんなことが起きるか)』という本が置いてあった。「読めば安心できます」という、妻からのメモも添えてあった。

僕は読んで、安心した。僕たちが子供について経験しているのは普通のことだとわかり、僕にとっては意外だったり以前の経験と異なっていたのだけど、自信を持てるようになったし、安心もできた。ちょうどそのころ、僕はスクラムとアジャイルの実験を始めていた。障害物に出くわしたり、不慣れな状況に陥ったりするたび、1年目にスクラムとアジャイルで『どんなことが起きるか』わかる本がほしいと感じた。

問題なのは、赤ちゃんの場合と違い、あなたのチームで最初の3ヶ月に何が起きるか、9～12ヶ月目には何に気をつけるべきか、正確に言い当てられないところだ。チームは子供と違って、育つ速度を予測できない。チームは転んだり、つまづいたり、ふらふらしながら最初の1年を過ごし、2歩進んでは1歩進むような有り様でチームとしての動き方を身に着け、アジャイルのエンジニアリングプラクティスを取り入れ、顧客との信頼を築き、インクリメンタルでイテレーティブに仕事を進めていく。

本書の構成では、そうした事情を反映して「僕はここで苦痛を感じる、どうすべきだろう」というアプローチをしている。僕は一緒に働いたり、様子を見ていたチームから、1年目のアジャイルにまつわる物語を集めた。僕がアジャイルの道を進む中で、集めた物語はいろいろな会社においてよく見られるパターンであり、だいたい同じだと気がついた。そこで僕は、ある会社でやってみたことを、少し調整して次の会社で使ってきた。このプロセスを繰り返してきた成果として、僕は現実に適用可能なソリューションを一揃い、仮想のツールベルトに携帯するようになった。そうした難しいポイントと、それに対するソリューションをいくつか共有しているのが本書だ。あなたのチームが傷ついていたり困っているなら、その症状に一番近い章を開いてほしい。症状の治療法か、すくなくとも苦痛の緩和方法を見つけられるはずだ。

スクラム現場ガイドはあなた自身の実践方法を微調整したり、不慣れな土地を導いたり、僕たちがみんな経験してきた障害物を避けたりしやすくするための本だ。

## 対象読者

スクラムやアジャイルを始めようと思っていたり、まさに始めたところだったり、1年くらいやってきて道に迷ったように感じているなら、本書はあなたのためにある。僕は公式には、新たにプロジェクトを始めて6ヶ月から18ヶ月の12ヶ月の間にいる企業が対象だとしている。

本書は実践主義者のためにある。あなたが理論や難解な議論に興味があるなら、他の本を選んだほうがいい——そうしたスクラムやアジャイルの素晴らしい本はたくさんある。そうでなく、実践的なアドバイスや現実のデータ、僕が実際にマイクロソフトのプロジェクトに参加したり、他の会社でチームをコーチしたり、フォーチュン100企業でコンサルティングしたりしてきた経験に興味があるなら、本書をおすすめする。

## 本書の構成

本書はどの章からでも、どんな順番でも、いつでも読めるようになっている。それぞれの章は物語が始まると。物語はすべて僕が参加したりコーチしたチーム、企業、プロジェクトからとったものだ。ご想像の通り、何の罪もない人びとのため、名前は変えている（罪がないとは言いたくない連中もいるけれど）。物語を読んだら、次はモデルを紹介するが、こちらも同じくらい聞き覚えがあると思う。紹介するモデルは僕が現場で、物語で現れたような問題を解決するのに使うものだ。中には不快に感じたり、あなたの会社ではうまくいくと思えないものもあるだろう。僕としては、アドバイスを無視したいという感情や、モデルを変えてしまう衝動とは何としても戦ってほしい。少なくとも3回はそのままで試してみて、結果を見てほしい。驚くような結果になるかもしれない。各章の終わりには成功の鍵をまとめており、あなたが実現に成功するか失敗するか、その鍵となる要因を説明している。

本書は4つのパートに分かれている。

第1部「準備」ではスクラムを始めるに当たってのアドバイスと、成功に向けた準備について書いている。スクラムの導入を検討しているか、始めたばかりならばここから読むのがいい。

第2部「現場の基本」では、アジャイルのやり方を始めるとチームや組織が出会うことになる初期の障害物を、乗り越える助けとなるいくつかの項目について議論している。スクラムを実践していて、問題を抱えているなら、ここから始めるといい。

第3部「救急処置」は会社が抱える、より大きく深い問題に対応する方法をまとめている。プロジェクトへ要員追加するやり方や、機能不全になったデイリースタンドアップの直し方などだ。ここで紹介する状況は、あなたが最初の1年間のどこかのタイミングで遭遇するものになる。このパートではトリアージと治療によって、あなたのチームを健康に戻す方法を紹介している。

最後のパート「上級サバイバルテクニック」で取り上げる事項は、人びとがタイミングに関係なくよく悩まされているものだ。アジャイルやスクラムでのプロジェクトのコスト算出、契約の作り方、ドキュメントの書き方などだ。

あなたがまったく新たにスクラムを始めるところならば、末尾の付録で簡単に説明している。基礎知識がないのであれば、ここで用語を学ぶといい。本書の前に、他の本でスクラムを勉強するのもいいだろう。

## あなたに本書を薦める理由

アジャイルに向けた旅をする中で、旅程のどのあたりであろうと、いま経験しているのは普通のことだと優しく教えてもらえば有り難いものだ。いまの状況に対処するためのアイデアや、成功の鍵まで聞ければ、さらに助かる。本書はそうしたすべてを提供しており、必要な章だけ読めばいいように構成している。もちろん、パートを通して読んでも、全体を読んでもいい。現実的な状況なのであなたにとっても理解しやすく、紹介しているソリューションはどんなチームでも使える。ページをめくって物語を読んでほしい。本書を頼れる仲間として、あなたはスクラムやエクストリームプログラミングのいいところも悪いところも一緒に経験することになるだろう。

## 補足資料

本書を読んでいて「ツールやテンプレートをダウンロードできれば、実際に使えるのになあ」と思うことがあるかもしれない。実はあるんだ。<http://www.mitchlacey.com/supplements/>にいろいろなファイル、画像、スプレッドシート、ツールなど、僕が日々のスクラムで使っているものを用意した。きれいな体裁のものもあるが、ほとんどは作ったなりだ。僕がプロジェクトで使うのに、体裁は関係ないためだ。使えればいいんだ。僕のウェブサイトにあるものは無骨かもしれないが、塹壕からやってきたものであり、役に立つ。

## *Acknowledgments*

---

### 謝辞

僕がこの本の着想を得たとき、アイデアはまだ生の状態だった。そのとき僕はまだ、海を沸騰させるような難事になるとは気づいていなかった。僕の妻、バーニスのおかげで僕は落ち着いて進めることができた。子供たちも同様だ。彼らの力がなかったら、この本は今日存在していないなかっただろう。

デビッド・アンダーソン、ワード・カニンガム、ジム・ニューカークの指導のおかげで、僕の最初のチームは自立することができた。マイクロソフトのことだ。3人はそれぞれマイクロソフトにいて、僕たちが困難を迎えていたときにコーチしてくれた。僕は今でも、ワードとの昔のセッションのメモを読み返す。そこにはこんな質問が書いてある。「TDDはよしてもいいんじゃないのか?」3人とも、落ちこぼれだった僕たちチームを、素晴らしい姿に変身させてくれた。デビッド、ワード、ジム、ありがとう。

僕はマイク・コーンとエスター・ダービーに感謝したい。2人はAgile 2006のカンファレンスで僕のアイデアを聞いてくれた。マイクはその後もサポートを続けてくれて、彼が当時書いていた『Succeeding with Agile』より早く僕が書き上げるだろうとよく冗談を言い合った。彼の方が先に出版できたときには、今度は彼が祖父になる前に書き上げればいいと言ってくれた。見てくれ、マイク、そっちのゴールは達成したよ！ 君の一番大きな娘さんはまだ高校生だけど、やり遂げたには違いないんだ！

レベッカ・トレーガーの助けがなければ本書は完成しなかった。彼女は地球最高の編集者だ。僕が集中して作業できるようにしてくれ、生のアイデアと言葉を章にまとめていくのを助けてくれた。

この本がいまる形になるのを助けてくれたすべての友だちに感謝したい。ここに名前を挙げた人びとはみな、貴重なフィードバックをくれ、何時間ものあいだ僕の話を聞いてくれたり、初期のドラフトに目を通したりしてくれた。Tiago Andrade e Silva, Adam Barr, Tyler Barton, Tor Imsland, Martin Beechen, Arlo Belshee, Jelle Bens, John Boal,

Jedidja Bourgeois, Stephen Brudz, Brian Button, Mike Cohn, Michael Corrigan, Scott Densmore, Esther Derby, Stein Dolan, Jesse Fewell, Marc Fisher, Paul Hammond, Bill Hanlon, Christian Hassa, Jim Highsmith, Donavan Hoepcke, Bart Hsu, Wilhelm Hummer, Ron Jeffries, Lynn Keele, Clinton Keith, James Kovaks, Rocky Mazzeo, Steve McConnell, Jeff McKenna, Ade Miller, Raul Miller, Jim Morris, Jim Newkirk, Jacob Ozolins, Michael Paterson, Bart Pietrzak, Dave Prior, Michael Puleio, Rene Rosendahl, Ken Schwaber, Tammy Shepherd, Lisa Shoop, Michele Sliger, Ted St. Clair, Jeff Sutherland, Bas Vodde, Brad Wilson、  
くら感謝してもしたりない。

Addison-Wesley社のチームと、クリス・ザーン、クリス・グジコウスキにも感謝したい。クリス・ザーンは僕が書いたほとんどすべてを見直させ、言葉を違った角度から見させてくれた。クリス・グジコウスキは僕が当初計画から2年超過した縮め切りすら破ったときにも、クビにしないでくれた。チームのみんなが本作りのプロセスをガイドしてくれたことにも、感謝している。

本というものは、頭から飛び出して紙の上に立ち現れたりはしない。僕が今までに関わった多くのプロジェクトと同様、チームによる作業だ。ここに名前を挙げた人びとは（挙げ忘れている人も何人かいるだろう）、僕の話を聞き、僕が間違えたときは教えてくれ、チームや顧客に実験してみるアイデアを提供してくれ、レビューしてほしいときにはしてくれた。きっと、この本が出版できて、僕と同じくらい喜んでいてくれると思う。この本を手にしているあなたも、読み終えたら僕と一緒に、このガイドの実現を手伝ってくれた彼らに感謝してくれたらと願う。

## *About the Author*

---

### 著者紹介

ミッチ・レイシーはアジャイルの実践者でありコンサルタントで、Mitch Lacey & Associates, Inc.の創立者である。ミッチはアジャイルの原則とプラクティス、スクラムやエクストリームプログラミングを企業が適用し、効率改善を実現する支援を得意としている。

ミッチは「技術オタク」を自称しているが、テクニカルなキャリアは1991年にコンピュータゲーム会社である Accolade Software から始めた。ソフトウェアテスト技術者、テストマネージャ、開発者、その他の様々な職種を経験した後、彼の天職とも言えるプロジェクトマネジメントとプログラムマネジメントに就いた。

ミッチはプログラムマネージャとして働く中で、アジャイルを自らのツールbeltに加えた。Microsoft Corporationでアジャイルスキルを身に着けながら、チームと共に Windows Live のコアエンターブライズサービスを成功裏にリリースした。ミッチの最初のアジャイルチームはワード・カニンガム、ジム・ニューカーク、デビッド・アンダーソンからコーチを受けた。ミッチは様々なプロジェクトでプロダクトオーナーやスクラムマスターとして働きながら、アジャイルスキルを伸ばした。やがて他のチームがアジャイルプラクティスを導入する支援をできるまでになった。16年以上の経験を積んでなお、今日もさらに自らの技能を成長させるべく、数多くの組織でプロジェクトチームと共に実践と実験をおこなっている。

認定スクラムトレーナー (CST)、PMI のプロジェクトマネジメント・プロフェッショナル (PMP) として、ミッチはプロジェクトマネジメントや顧客マネジメントの経験を認定スクラムマスター研修、アジャイルコーチ、カンファレンスでの講演、ブログ、ホワイトペーパーなどを通じて共有している。ミッチは世界中の企業と仕事をしており、オーストラリアからコロンビア、カリフォルニアからフロリダ、ポルトガルからトルコ、その他あらゆる場所にクライアントを持っている。

ミッチは世界中の数多くのカンファレンスで講演しており、Agile2012 のチアを務め、Scrum Alliance と Agile Alliance の board of directors にも名を連ねた。

ミッチについて詳しくは [www.mitchlacey.com](http://www.mitchlacey.com) を参照のこと。ミッチのブログや記事、ツール、動画などが、スクラムやアジャイルの導入に有用である。Twitter では @mglacey、メールアドレスは mitch@mitchlacey.com。