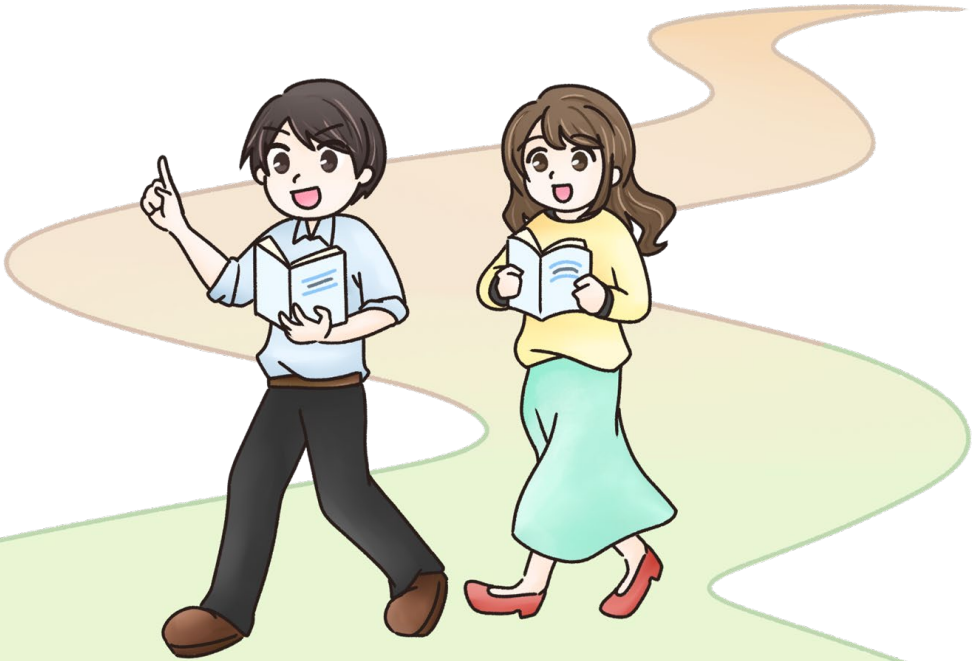


## アジャイルプロジェクト 実践ガイドブック

- アジャイル実践者が語る、難所、スキル、育て方、法務そして実践例 -

Ver1.0





# 目次

I. 概要	4
II. アジャイル開発に対する危機感	6
III. アジャイルにおける技術と難所	9
IV. エンジニアのスキル	13
V. アジャイル組織の育て方	22
VI. 法務から見たアジャイル	30
<インタビュー> 銀行でのアジャイル実践例	35
さいごに IPAから読者へのメッセージ	41
アジャイルWGメンバー一覧	42

## 1. 概要

# 1. 概要

## 背景

一般企業でもアジャイル開発の実践が増加している。様々な変化を受け入れながらITシステムを開発しなければならない場面が増えてきたからだ。そして、2018年に経済産業省からDXレポートが発行されたことをきっかけに、アジャイル開発の実践が急増した。レポートの中で、DXの実現やDX人材の育成にはアジャイル開発が有効であると明示されたためである。今後、アジャイル開発が主流になることは間違いないだろう。

## 目的

この冊子は「事業を成功させるために、アジャイル開発を活用したい」と願う事業部門の方々に向けて書かれた。難解なイメージがあるアジャイル開発だが、実はシンプルであり、成功の秘訣は実践中よりも事前の準備にある。この冊子をヒントに、アジャイル開発を正しく理解し、正しく準備していただきたい。

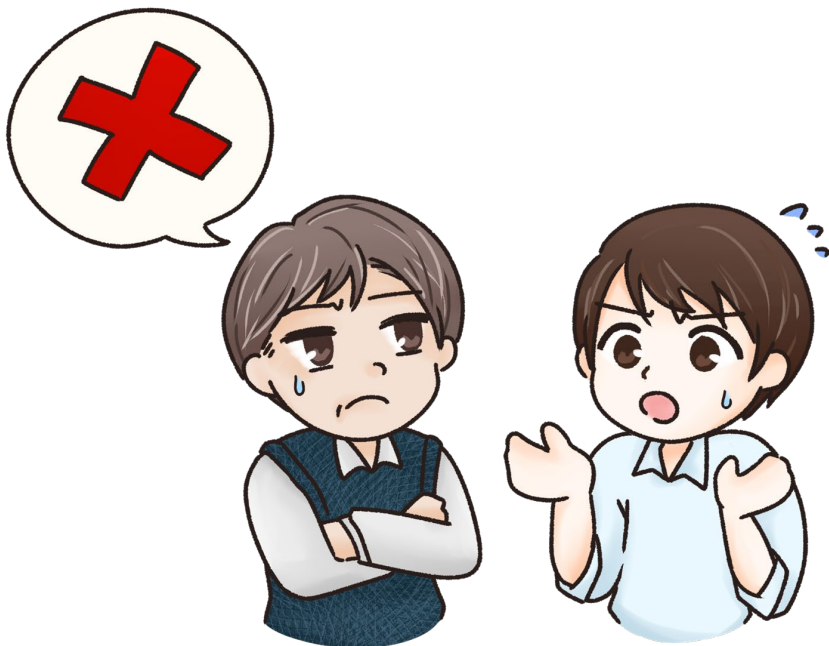
## 特長

この冊子の特長を説明する。他の読み物にはない、アジャイル開発を成功させるために必要な様々な観点を盛り込んだ。失敗の原因となる誤解、開発に必要な技術、遭遇する難所とそれを乗り越えるコツ、組織の成熟度に応じた体制作り、エンジニアを育成する際のヒントとなる技術的な土台、開発を外部に依頼する際の適切な契約、などである。企業全体でアジャイル開発に取り組んでいる経営層へのインタビューも掲載した。

この冊子は、開発部門だけでなく、周囲の関係部門にとって大きなヒントになるだろう。多くの企業にとって、この冊子が、アジャイル開発を活用して事業を成功させる、さらには企業自体を発展させるきっかけになることが、執筆者一同の願いである。

GOOD LUCK!

## II. アジャイル開発に対する危機感



## II. アジャイル開発に対する危機感

### アジャイルに対する危機感

世界的なアジャイル型によるプロダクト開発の流れを受け、日本でも近年アジャイル型のプロジェクトの開発が増えてきている。実際に、スクラムなどのアジャイル関係の資格取得者は年々増加しており、コミュニティ活動も盛んに行なわれている。さらに、内製化を目指す企業を中心として、アジャイル開発によるプロダクト開発も増えている。しかし、アジャイル型のプロジェクトの採用は、いまだ十分とは言えない。

IPAのDX白書2023によると、米国企業の73.3%が実際にアジャイル開発活用又は活用を検討しているのに対し、日本企業は41.5%にとどまっている。ここから、DXの普及が推進されアジャイル開発は有力なツールと位置づけられているものの、ユーザーとなる企業側に十分な体制が整っておらず、導入に踏み切れていないという現状が見えてくる。

### アジャイルでの誤解

アジャイル開発でうまくいかないという話を聞いてみると、アジャイル開発が誤解されているケースが多くあるようだ。例えば、アジャイル開発を導入するためには、プロジェクトの管理は重要な側面である。進捗を透明に可視化し、ステークホルダーに伝え、動くソフトウェアを見せる必要があるが、アジャイルであるにも関わらず、ウォーターフォール型開発の発想をそのまま踏襲して、計画時に一度決めたら以後一切プロジェクトのスコープやマイルストーンを変更しないというプロジェクトがあるようだ。

また、要件を最初から全て決めなくても良いとはいえ、要件定義は必要になるのだが、事業会社でそのような活動のルールを立てないまま、開発に入ることがある。ウォーターフォール型の開発でも、アジャイル型の開発でも、開発するためのスキルやノウハウの蓄積は必要であるが、十分な準備なしに開発を進めてしまい効果が得られないという事例もある。

## II. アジャイル開発に対する危機感

しかし、十分な準備をしてプロジェクトを進めれば、顧客に素早く価値を提供し、仕様の変更にも柔軟に対応できるアジャイル開発のメリットが十分に引き出せるようになる。

### アジャイルに関する技術的な側面

アジャイル開発の事例を見ていると、多くはマインドや組織づくり、チームビルディング、プロダクト戦略といったテーマのものが多くあふれている。

一方で、アジャイル開発を採用するには、技術的な観点でも考慮が必要になるが、エクストリームプログラミング (XP) のベストプラクティスなど、ノウハウなどが紹介されることはあっても、その意義やアジャイルとの関係まで多くは語られていない。

例えば、アジャイルでは開発のサイクルを高速に回すことが期待されるが、ビルドやデプロイに時間がかかると、期待通りの成果が達成できない。レガシーに作られたシステムでアジャイル型を採用し要したときに、同じような問題が起きていないだろうか？

次章以降では、アジャイル型プロジェクトに必要な技術に触れていきたい。



### III. アジャイルにおける技術と難所



## III. アジャイルにおける技術と難所

### アジャイルにおける開発に求められる技術

アジャイル型の開発を採用しようとしても、開発方法や使用されている技術に課題があると、成果を得ることは難しい。本章では、アジャイル型開発の特徴から考察した、アジャイル型に必要な技術や開発手法について説明する。

アジャイル型の開発には、以下の特徴がある。

- ・ 成果物を素早くリリースする
- ・ ニーズに合わせて変更を加えやすい
- ・ 予測不可能な問題に対応し続ける

上記を達成するために、求められる技術について説明する。

### 1. CI/CD

CI/CDとは、「Continuous Integration/Continuous Delivery」の略であり、日本語では継続的インテグレーション/継続的デリバリーと呼ばれている。CI/CDでは、開発したプロダクトを共通リポジトリに登録し、プロダクトを顧客が利用可能な状態へとリリースすることを指す。この一連のプロセスが簡潔になることは、プロダクトが価値を生み出すことに繋がる。

### 2. テスト駆動開発

テスト駆動開発 (Test-Driven Development) とは、テストを中心として、テスト→実装→リファクタリングを行うことで、プロダクトコードの実装を行う方法である。

## III. アジャイルにおける技術と難所

テスト駆動開発では、「動作する」ことを優先しており、いち早く価値を提供し、フィードバックができる上、テストしやすい設計がプロダクトに求められるため、依存度が低く、改修しやすいコードを書きやすい傾向にあると言われている。また、開発においてはテストが作成されることが保証されるので、変更したときの予期しない不具合を早期に検出しやすくなる。

### 3. ドメイン駆動設計

ドメイン駆動設計とは、エリック・エヴァンズが提唱した、ユーザーが従事する業務の語彙に沿ってソフトウェアを開発する手法である。ドメイン駆動設計では、独立した業務を分割し、業務の語彙とプログラミングの語彙をユビキタス言語として共有し、業務に注目した開発を行うため、業務の追加や変更に対して、変更に対する影響を最小限にすることができるようになる。そのため、業務の変更に合わせたシステムの変更がしやすくなるという特徴がある。

### 4. ペアプログラミング・モブプログラミング

システム開発においては、ときには予測不可能な問題に取り組むことが求められる。そのとき、開発者が単独で開発を行うと、開発時には見つかりづらい問題があった場合や、リリース後に分かった課題に対する変更に時間がかかる。

ペアプログラミング・モブプログラミングでは、2人ペアやチームで一体となって（ドライバー役とナビゲータ役を随時入れ替えながら皆で協力して）1つのプロダクトの開発に取り組むため、問題意識や要求のポイントの理解、課題の早期発見に繋がることや、設計やコード・その他の技術カルチャーに関する知識を共有することで、手分けをして問題を解決・改修しやすくなる。また、チーム内でのコミュニケーションもスキルトランスファーもスムーズに進めやすいというメリットが生まれる。

## III. アジャイルにおける技術と難所

### 5. クラウド技術

プロダクトに求められる業務要件や非機能要件（性能やセキュリティ等に関する要件）の予測が難しいシステムを構築するためには、まず、素早くシステムを構築し、規模をスケールさせたり縮小したりなど状況に合わせてネットワークやサーバーのリソースを変更し、ときにはアーキテクチャを大きく変更する必要がある。このような状況に対応するためには、事前に全てを決めるのではなく、状況に応じて必要な変更ができる構成にする必要があるが、サーバーやミドルウェアなど物理的に用意を行うと、変更が難しいものがあったりし、サービス自体の作成に時間がかかる。インターネット上のサービスを積極的に活用することで、素早くシステムを構築し、必要なときに必要なものを提供することができる。

### アジャイルでの難所

アジャイル開発を実践するにあたって、人員や体制を整え、プラクティスを学んでも難しさが残る。アジャイル開発では、不確定な要素の多い領域を対象にすることが多くあり、不確定な上に想定外のことが起こりがちだからである。このような困難を解決するためには多くの経験が必要となる。しかし、経験豊富なメンバーが揃うことは難しい。

また、事業会社では新規開発だけではなく、古くから稼働している基幹システムのような事業のコアとなるプロダクトの改修を含む開発を行うこともあるが、このようなプロダクトでは変更に対する影響も大きく、技術的な負債も残っていることが多いため、アジャイルの利点を活用しきれない。

このような場面では、経験豊富なベテランの知識や経験は重要であり、そうしたメンバーや組織と一体となって推進することが望ましい。しかし、そうしたメンバーや組織に突然新しいやり方を押し付けるようなことになると、ときによっては大きな抵抗を産みかねない。じっくりと対話を重ねながら、変革を促していくことが、結果的にはアジャイルが成功する秘訣なのかもしれない。

## IV. エンジニアのスキル

## IV. エンジニアのスキル

アジャイル開発では、1、2週間の繰り返しの中でも、拡張性と変更容易性を考慮した設計・実装・テストを実践できるエンジニアが必要である。なぜなら、対象となるプロダクト／サービスを支えるITシステムの構築にアジャイル開発を採用するということは、ビジネスに合わせてITシステムを素早く拡張／改修することが期待されており、その期待に応えられるような以下の構造でITシステムが作られている必要があるからだ。

- ・ 拡張性が高いアーキテクチャの採用
- ・ 改修しやすい疎結合な設計
- ・ 性能を考慮した適切なアルゴリズムの採用
- ・ 読みやすくテストしやすいソースコード
- ・ 適切なテストケースとテストコード

もし、エンジニアの技術スキル不足により、ITシステムの内部が上記に反する構造で作られてしまったことに後から気づいた場合、その構造を後から改善するのは非常に困難である。ほぼ作り直しになると覚悟した方が良い。例えば、自動テストを前提とせずに作成したソースコードに対して、後から自動テストを作成することが非常に困難であることは、良く知られている。



## IV. エンジニアのスキル

以降では、前述したことを実践できる技術スキルを持つエンジニアを「アジャイルエンジニア」と呼ぶ。そして、その育成に必要な基礎技術について説明する。

### アジャイルエンジニアを育成する

現在、アジャイルエンジニアは不足している。ウォーターフォール開発では、一部の卓越した技術を持つエンジニアが設計を担当し、他のエンジニアはその設計通りにソースコードを作る。そのため、高い技術スキルを持っていなくても作業することができる。しかし、アジャイル開発では、1、2週間の中で開発に必要な全ての作業（設計・実装・テスト）を各人が実施しなければならない。すなわち各人がアジャイルエンジニアとして幅広い技術スキルを持っていることが必要であるが、そのようなスキル保持の割合は低く、どうしても育成が必要となる。

ちなみに、アジャイル認定資格保持者（認定スクラムマスタなど）は年々増加しているが、アジャイル認定資格はアジャイル開発の考え方に関する基礎知識を測るものである。資格取得者が前述した技術スキルを保持しているわけではないので、混同しないように注意すべきである。

上記の理由により、ITシステムを内製化する場合は、自分たちの組織でアジャイルエンジニアを育成／採用する必要がある。しかし、技術領域は幅広く、どのような技術スキルを重視すべきか、悩ましいところである。つつい最新技術に目がいきがちだが、ここでは、あえて「基礎技術」に注目する。アジャイルエンジニア育成のヒントにしたい。

## IV. エンジニアのスキル

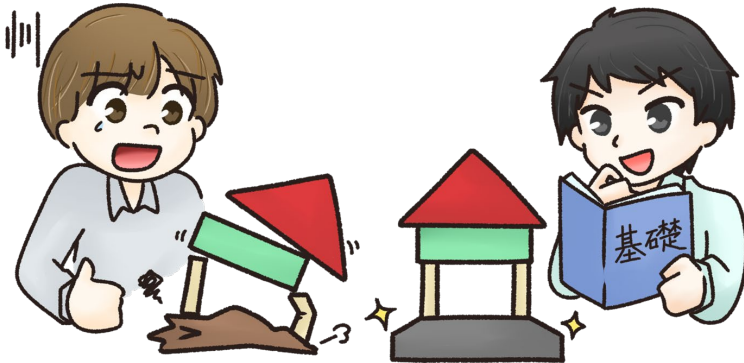
### 基礎技術に注目する

アジャイルエンジニアを育成するために注目すべき基礎技術の中から、特に重要と思われる4つを列挙した。

- ・アルゴリズムとデータ構造
- ・オブジェクト指向
- ・ソフトウェアテスト
- ・モデリング

上記は昔（少なくとも20年以上前）からの普遍的な基礎技術である。時代を超えて、今なお大切にされている。なぜこれらを挙げたのかを以下で説明する。

1点目として、これらの基礎技術はコンピュータサイエンスの基礎であり、様々な技術の大切な土台となっている。例えば、シリコンバレーのIT企業は、エンジニアを採用する際にコンピュータサイエンスの基礎技術を重視している。





## IV. エンジニアのスキル

2点目として、上記の基礎技術がない状態でシステム開発を行なうと、冗長で複雑な悪い作りになってしまうが、本当に恐ろしい点は「作りの悪さが外から見えないこと」である。中の設計・実装が悪くてもITシステムはなんとか動作するため、開発は継続でき、課題と認識されないままITシステムが完成してしまう。いわゆる「技術的負債が蓄積した状態」である。

上記の課題について補足すると、例えばクラウド技術に関する基礎的なスキルが不足しているなら、クラウド上でソフトウェアを動作させることができず、早期に開発を継続することができなくなる。そして現場で大きな問題となり、早期に対策が採られる。

技術的負債が蓄積した状態のITシステムでは、それを使い始めてから拡張／改修する際に、作りの悪さが大きな障害要因となる。ひいてはITシステムに依存するビジネスが失敗してしまうだろう。

以下では、4つの基礎技術について概要を説明する。

### 1. アルゴリズムとデータ構造

ビッグデータなどの大量のデータを扱う時代だからこそ、適切なアルゴリズムとデータ構造を選択しなくてはいけない。例えば、安易に二重ループ処理を作成すると、データ量が増えたら、あっという間に性能が悪化してしまう。そしてITシステムが使い物にならなくなり、ビジネスに大きな悪影響を及ぼす。ちなみに、この基礎技術はエンジニアの能力を測る指標としても使われている。この章にあるコラムを参照いただきたい。

### 2. オブジェクト指向

ITシステムを素早く拡張／改修するためには、シンプルなアーキテクチャを採用して、「各モジュールは高凝集度でモジュール間は疎結合である」という良いソフトウェア設計にすることが大切である。また、設計を良い状態に保つために、部品の再利用やリファクタリングの継続も大切である。そのためには、オブジェクト指向などの良いソフトウェア設計原則に基づいた設計と、それらを応用したデザインパターンやアーキテクチャパターンなどの活用が有効である。

### 3. ソフトウェアテスト

設計・開発・テストを繰り返すアジャイル開発では、テスト自動化が必要である。しかし、適切にテストケースを設計しないと、重複や偏ったテストケースが増え、ITシステムを改修する際の大きな阻害要因になってしまう。

### 4. モデリング

モデリングは、変化に対するインパクトをデータ構造面から素早く把握するのに役立つ。さらに、エンジニアを含むプロジェクト参加者の間で、システムの共通理解を作り出すことができる。また、データ構造を洗練することで、肥大化したシステムの無駄な機能をそぎ落としてスリム化でき、業務ロジックがシンプルになる。ちなみに、実際のDBの物理テーブル関連図（ERD）とは別物であり、概念データモデルや分析・設計クラス図でのモデリングである点に注意していただきたい。

補足：オブジェクト指向・デザインパターン・モデリングは、現在ではドメイン駆動設計と呼ばれる手法の中に系統立てて構成されている。

## IV. エンジニアのスキル

アジャイル開発の成功には、アジャイルエンジニアが必要である。しかし、アジャイルエンジニアは不足しているため、企業として育成／採用しなければならない。その際に重視すべき技術スキルとして、今回取り上げた普遍的な基礎技術に注目して欲しい。

### コラム：コンピュータサイエンスの基礎力である「アルゴリズムとデータ構造」スキルの重要性について（paiza片山社長へのインタビュー）

#### paizaとは

国内最大のITエンジニア向け転職・就職・学習プラットフォームである。

paizaサービスの中核となる「プログラミングスキルを測るスキルチェック」には、基本構文をベースにしたデータ入出力処理問題から、探索ロジックやツリー構造などの「アルゴリズムとデータ構造」の知識を使って計算量を効率化させる問題まで揃っている。その問題で測定した結果が、転職・新卒採用の評価指標のひとつとなっている。

#### IPAからの質問

ITエンジニアに求められるスキルはクラウドやフレームワークなど多岐にわたると思うが、市場のニーズに対しpaizaのスキルチェック（アルゴリズムとデータ構造の観点）が有効な評価指標になり得る理由についてお聞きしたい。

#### 回答

「コンピュータサイエンスの基礎力が重要」

paizaでは、コンピュータサイエンスの基礎力である「アルゴリズムとデータ構造」を学ぶことができる。その点に着目した理由は、シリコンバレーの採用方法をリサーチした結果、エンジニアは大学でコンピュータサイエンスを学んでいないと応募できず、日本とベースラインが違うことが分かったから。ベースに差ができてしまうと、いつまでたってもシリコンバレーには追い付けないと考えた。

## IV. エンジニアのスキル

様々なエンジニアと接し、プログラムの裏側でコンピュータがどう動くか理解できている人はバグが起きても問題に早く気づけるが、そうでない人は一向に解決できないことが分かり、そこからコンピュータサイエンスの基礎を理解している人の方が、伸び代が大きいと仮説を立てた。コンピュータのふるまいや、裏で何が行われているかを理解している人（≡コンピュータサイエンスを学んでいる人）は、バグや問題が起きたときに、処理の流れを考え、なぜそれが起きるのかを構造的に考え、問題が起きた箇所を特定できると考えている。こういった本質的なコンピュータのふるまいを理解できている人は、プログラム言語やフレームワークが違っても、ちょっとした表現の差程度の話になるため、キャッチアップが早いと考えた。

対して、コンピュータのふるまいが理解できない人にとっては、どのようにコンピュータがデータを処理しているのかを理解できないため、何か問題があっても対処が困難であり、新たなことを実現するロジックを考えることも難しいだろうととらえて捉えている。

### 「paizaのランクと実務の関係」

プログラミングスキルチェック結果は、高い方からS,A,B,C,D,Eの6つのランクに分かれている。ランクが高い人は、未経験の言語に対してもキャッチアップが早く、基本的なコンピュータのふるまいを理解しており、WEBフレームワークのことも直ちに理解できる。Sランクは上位2%なのでハードルが高いが、Bランク以上であれば入社後に活躍しているようだ。未経験者であれば、自力でCランクまでいける人は、自学自習でその後も伸びるように感じている。

ランクが高い人は独学で学べる特徴があり、自学自習能力の証明になる。例えば問題では標準入力を使うが実務で標準入力を使うことは少ない。そのような場合でもすぐに調べて学ぶことができる。また解けなかった問題に再挑戦する意欲も高い。したがって新技術の習得能力と相関が高いと考える。

## V. アジャイル組織の育て方



## V. アジャイル組織の育て方

### アジャイルを組織に適応させる計画を作る

アジャイルを導入する目的は、ビジネスのゴールを素早く達成することにある。そのため、スクラムなどのプラクティスを利用して、チームの成長の方向性、施策を細かく調整していくことが重要である。ウォーターフォールのプロジェクトの進め方と、アジャイルのプロジェクトの進め方が違うように、アジャイルのチームの育て方や育ち方は、ウォーターフォールとは異なる。また、プロジェクトの進め方の違いが、ウォーターフォールを前提とした既存の組織との間に、ギャップを生むことも想定される。アジャイルチームを育てつつ、ウォーターフォールを前提とした既存の組織・人員とのコラボレーションを考える必要がある。

### ビジネスの結果までの道のりを示す～Value Stream Map作成の勧め

ビジネスの成果物は、世に出ることで初めて評価される。そのため、評価を得るまでのプロセスを、Value Stream Mapなどで可視化する。プロセス、ツール、登場人物を明らかにし、今後どのようなスケジュールでアジャイルに適合させていくのか計画を作成することを推奨する。

#### Value Stream Mapの例



Value Stream Mapとは、アイデアの発生から実際のそのアイデアが現実のサービス・商品としてお客様の手元に届くまでのタスクや承認の流れ、タスク、承認のリードタイムなどを視覚化したものである。

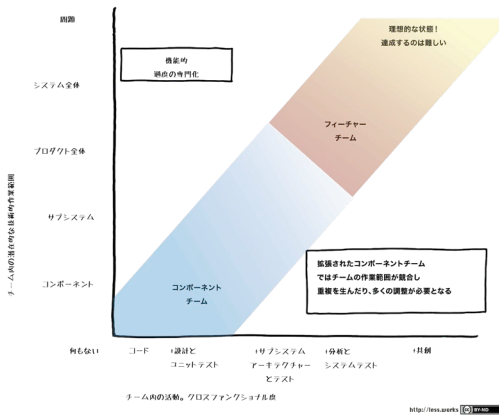
出典 <https://gihyo.jp/dev/column/01/devops/2017/value-stream-mapping>

# V. アジャイル組織の育て方

## フィーチャーチーム適応マップ

Value Stream Mapを作成することで、商品・サービスを世に出すまでに多くの組織、プロセスが必要であることがわかる。これらのプロセスをアジャイルチームの活動に取り込むことで、よりスムーズにビジネス成果を出せるようになる。

またアジャイルチーム自身がプロセスを取り込まずとも、アジャイルチームが必要とするタイミングで、必要なプロセスが実施される環境を作っていく計画を、フィーチャーチーム適応マップをベースに作成する。



フィーチャーチーム適応マップとは、Value Stream Mapに登場するプロセスを、アジャイルチーム内でどこまで実施できるようにするかを計画する際に利用する。

出典 [https://less.works/less/adoption/feature-team-adoption\\_map?preferred\\_lang=jp](https://less.works/less/adoption/feature-team-adoption_map?preferred_lang=jp)

## チーム・組織の成熟度に応じた適切な人材配置

アジャイルに関するプラクティスや、技術に関するスペシャリストを全てチームメンバーとして招集することは現実的に困難である。教育などを継続して実施し、メンバー自身が成長することや、外部スペシャリストのアドバイスやサポートを適切なタイミングで受けることも重要である。



## V. アジャイル組織の育て方

### アジャイルチームの成長と人材

アジャイルチームの成長と組織への適応にはいくつかの段階がある。それぞれの段階に必要な人材像や、進め方の要所を、立ち上げ期、拡大期、定着期に分けて説明する。

#### 1. 立ち上げ期

##### 求められる人材像の例：挑戦を楽しめる人

一番初めのチームは、多大なプレッシャーに直面することになる。恐らくその組織で最初の取り組みになるため、分からないことも大量に発生する。新しいことに挑戦するにあたり、組織上の課題、技術的な課題、ビジネス環境の変化など、様々な問題に対する、挑戦の日々になることは想像に難くない。もちろん、一人で立ち向かう必要はなく、社内外の有識者の協力を受けながら（例えばアジャイルコーチなど）、プロジェクトを進めていける体制、要員を招集すべきである。

##### 小さく始める

最初から10人や20人などの人数規模で、アジャイルのプラクティスを実践しようとして成功している事例は驚くほど少ない。アジャイルの方法論のひとつであるスクラムを実践するのであれば、最少人数構成でのスクラムチームのスタートを推奨する。

##### 原理原則を崩さない

スクラムであれば、スクラムガイドで定義されているイベントや、成果物、ルールを全て網羅して開始するべきである。スクラムの実施調査では、スクラムがうまくいかない理由の半数は、中途半端なスクラムの導入（ふりかえりや、スプリントレビューを実施していない）ことに起因している。これらの原理原則を崩さず、正しいプロセスと、正しい理解の上で、基本をしっかりと実施することが肝要である。

## V. アジャイル組織の育て方

### 足りないものを適宜足していく

組織内に、アジャイルのプロフェッショナルや、技術領域のプロフェッショナルが存在するケースは稀である。以下のリストに、プロジェクトを実行する際に必要となる技術要素や、ロール（スクラムを例に）を挙げる。これらの領域のスペシャリストを、プロジェクトチームにフルで参画させることが難しい場合や、部分的な参加で十分なケースは往々に存在する。これらの役割やスペシャリストは、チームをサポートする非常駐メンバーとして体制に組み込むことを検討すべきである。

- スクラムコーチ
- 開発言語等の専門分野
- 組み込み・ハードウェア等の領域
- CI/CDなどの開発環境整備
- テスト駆動開発
- リファクタリング
- テスト自動化
- クラウド技術
- ビジネス領域のプロフェッショナル

### 原理原則以外は変わっていい

ビジネス環境や、技術などは日々変化していく。チームを取り巻く環境が変化する中で、チームは変化しなくても良いということはなく、また、新しい技術や考え方を取り入れて実行することには不安や疑問がつきものであり、新しい取り組みが失敗することへの恐怖もある。前項でも記載のとおり、全てを自分たちの力だけで、解決しなくてもよい。社内外の知見を得ながら、小さなトライアンドエラーを繰り返すことで、リスクを可能な限り小さくしながら、新しい取り組みに挑戦し続けることが、アジャイルに取り組む上で非常に重要なマインドを醸成する。

## 2. 拡大期

1つ目のチームのビジネス成果が目に見えてくるタイミングの前後で、チームの拡張や、他のビジネス領域への適用、さらなるビジネス成果の獲得など、チームに対する要求事項が増えてくる。チームへの様々な要求が増えてきた場合、大きく次の2つの要求に分類することで、それぞれの対策の方針を考えていくとよい。アジャイルを組織に適応させる計画を立てる際にも、これらの分類を考慮しておくこと、巻き込むべき相手をイメージしやすくなる。

### 2.1 横に広げる

#### 求められる人材像の例：謙虚な人

活動を横に広げることは、1つ目のチームの活動が軌道にのってきただけで、次のゴールとして、もっとも最もお願いされるであろう要求である。チームが1つできたら2つに増やす、2つできたら4つに増やすという形で、チームが増えることを期待される。最も基本的な増やし方は「のれん分け方式」で、最初に出したチームから、数名選出して次のチームの立ち上げに参画する方式である。この「のれん分け」にあたるメンバー（初期チームのメンバー、新しくアサインされるメンバー双方）に求められる素養としては、「他者に対するリスペクト」が挙げられる。最初にチームを立ち上げたメンバーは、先駆者としておごらず常に学ぶ姿勢を、そして新規参入メンバーは、先達の工夫を受け入れる度量を示してほしい。

### 2.2 縦に広げる

#### 求められる人材像の例：組織のプロセス・ルールに強い人

今までアジャイルを適用したことがない部門を巻き込むことを求められるケースがある。例として、システムを開発する部門と、運用する部門が分かれている場合に、運用する部門がアジャイルの組織に組み込まれる場合である。当然、開発する部門と運用する部門は別の部門であるため、それぞれの部門のミッションを達成することが求められるが、開発部門のKPIと、運用部門のKPIは異なる場合が大半を占める。

このように複数の部門を横断する場合は、新たに組織共通のKPIを作成し、合意する必要がある。それぞれ部門のKPIをそのまま適用すると、優先順位の争いがチーム内で勃発し、チーム分裂を招く恐れがあるからだ。そのため、両組織（上記の例では開発部門と運用部門）の責任者が、チームの共通認識となるKPI（判断基準）を設定することを推奨する。これらのKPIは担当業務や、部門に適用される公式・非公式の各種業務プロセス、業務ルール、コンプライアンスなどに詳しいメンバーの協力を得て、整備することが肝要である。

#### 求められる人材像の例：経営層を巻き込むリーダーシップ

組織を横断する変更や、改革には多くのリソースを必要とする場合がある。経営の判断も必要な場面が出てくるため、適切なタイミングで経営層へのエスカレーション、助力を得られるようなリーダーシップを発揮できる人材のプロジェクトへの参画が必要となる。

### 3. 定着期

組織にアジャイルを定着させるというのは、プロセスや組織図に現れるものだけではない。

#### コミュニティ活動の勧め

先行したチームが、アジャイルを組織に適応させたプロセスや考え方などは、言葉にしないと失われてしまう。ガイドライン、ブログなど様々な方法で言葉にして伝えるという手順が必要となる。

ただし、言葉にしても分からないニュアンスや、事例ごとの判断など様々なものは言葉にしても伝わらないことが往々にして存在する。複数のアジャイルチームが、それぞれ自由にアジャイルを活用するスタイルもあるが、事業活動の一環として、一定の水準を満たすアジャイルを普及させるためにも、チーム間で悩みを共有し解決する場＝コミュニティを作ることが推奨される。社外のコミュニティ活動で様々な知見を得られるように、社内のコミュニティ活動も、チーム間での情報やナレッジを共有する大きな助けとなるだろう。組織としてのコミュニティ活動の支援も、計画に盛り込むことを強く推奨する。

## VI. 法務から見たアジャイル



## VI. 法務から見たアジャイル

アジャイル開発はユーザーの求めるものをできるだけ早く見える化し、フィードバックや新たなニーズに早急に対応する手法として、近年重要視されている。これはどの業界でも同じく必要とされている。そのため、ソフトウェア開発案件の法務を取り扱う弁護士業界でも、アジャイル開発の事案に実際に触れる機会が増えている。今まで主流の開発方法はウォーターフォール開発であったが、この手法は事前に要件・仕様を固めた状態で開発が始まるため、ユーザー・ベンダーのどちらに非があるのか判断しやすい。

しかし、アジャイル開発では、開発当初にプロジェクト全体の要件・仕様が決まっておらず、事後的な要件・仕様の変更を前提とするため、裁判では難しい判断が求められるだろう。こうした問題をできるだけ起こさないためにも、ユーザーとベンダーはできるだけお互いの情報や境遇について意見交換し、両当事者で協力して完成させるという工夫が必要となっている。こういったケースもあるため、アジャイル開発を行う際には、契約や法務という観点でも十分気をつける必要がある。

### アジャイル開発における契約書にまつわるエピソード

以下はソフトウェア開発を発注するための契約書の確認を依頼された顧問弁護士の事例である。詳しくは参考資料 2(b) をご覧いただきたい。

顧問弁護士は顧客から業務委託契約書のドラフトを受け取った。その中身を確認したところ、作ろうとしているソフトウェアが完成すればどんな価値があるのか、については書かれていた。しかし、具体的にどんなソフトウェアを作成するのか、ソフトウェアを作成するために誰がいつまでに何を完成させればいいのか、についてはほとんど書かれていなかった。このことから顧問弁護士は、主たる義務内容が曖昧な、できが悪い契約書だなという印象を抱いたそうである。

## VI. 法務から見たアジャイル

すると先輩弁護士から、これは従来の開発方法（ウォーターフォール型）ではなく、アジャイル開発の契約書ではないか、というアドバイスがあった。これまでは、契約書のレビューはウォーターフォール型の考え方に沿って行われてきた。しかし、アジャイル開発は、開発プロセスなどが従来と大きく異なる。よって、アジャイル開発の契約書をレビューする際には、その特徴を理解してレビューを行うことが必要だということが分かった。

この例からも分かるように、ウォーターフォールとアジャイルというそれぞれのソフトウェア開発手法の特性と相違点を正しく理解することで、それぞれに適した契約書を作成することが可能となる。法律の専門家である弁護士も、アジャイル開発に対して理解を進めなければならない時代に突入したというエピソードである。

### **発注者がアジャイル開発の利益を最大限享受するために**

アジャイル開発は、開発開始の契約締結時点では、そのソフトウェアでどんなことを達成するのか（ビジネスゴール）を明らかにして合意していることこそが大切である。契約締結時点では達成するまでの課題が不明であるのは当たり前と言える。そのため、達成するまでの課題を具体的にし、どのように達成していくかを受注者側と発注者側がコミュニケーションを通じてビジネスゴールを共有していくことが重要である。よって開発行為を進めながらトライアンドエラーを繰り返して要求を整理していくと良い。

このような進め方は発注者が負担に感じてしまうかもしれないが、こうした負担を避けようとする、アジャイル開発自体が成功しなくなる。積極的に会議にも参加し、意見を言える機会を確保することが発注者の利益であるという視点で、契約書の内容を確認していくことが重要である。



### アジャイル開発と偽装請負

アジャイル開発では、事前に全ての要件を決めることが難しいため、準委任契約が推奨されている。しかし、準委任契約を結んでも、発注者が直接、受注者の担当者に指示することは偽装請負であるとみなされているため、注意が必要である。このような懸念に対しては経済界からも解消の働きかけがあり、政府の規制改革推進会議の成長戦略ワーキング・グループにおいて取り上げられている。この活動を受けて厚生労働省が発出した公開資料（参考資料3：厚生労働省の疑義応答集（第3集））によれば、実態として、発注者と受注者の関係が対等に協働を行い、受注者側の開発担当者が自律的に判断して開発業務を行っているとも認められる場合であれば、偽装請負とならないとされ、その具体例が示されているので、是非参考にして欲しい。

他方で、発注者側の開発責任者や開発担当者が、受注者側の開発担当者に対し、直接、業務の遂行方法や労働時間等に関する指示を行うなど、指揮命令があると認められるような場合には、偽装請負と判断されるリスクが高い点に注意して欲しい。

## VI. 法務から見たアジャイル

これまで述べた心配も、発注者側、受注者側がそれぞれの情報や境遇についてコミュニケーションを深めれば防げるものである。しかし、トラブルを防ぐためとはいえ、ウォーターフォール開発より、コミュニケーションの負荷が高くなるのは事実ではある。ただしより良い品質の成果物を作り出すことが最も重要なことで、受注側が主体になって作成していくのではなく、発注側も積極的にプロジェクトに参加できるというアジャイル開発ならではのメリットを享受するために必要不可欠なものであるという認識を持って欲しい。

また、実態として発注者と受注者の関係が対等に協働を行い、受注者側の開発担当者が自律的に判断して開発業務を行っていると認められる場合は、アジャイル開発であっても契約形態に関する制約を減らすことが可能である。具体策に関しては、疑義応答集の内容を一読していただくことで、アジャイル開発への適切な対応に活かして欲しい。

### 参考資料

1. 情報処理推進機構 情報システム・モデル取引・契約書（アジャイル開発版）  
<https://www.ipa.go.jp/digital/model/agile20200331.html>
2. 情報処理学会「情報処理に関する法的問題」研究グループ  
<http://www.ipsj.or.jp/sig/lip/>
  - (a) アジャイル開発のソフトウェアモデル契約
  - (b) アジャイル開発をサポートするための法務部門のマインドセット
3. 厚生労働省の疑義応答集（第3集）  
<https://www.mhlw.go.jp/content/000834503.pdf>

<インタビュー> 銀行でのアジャイル実践例



### 北國銀行におけるアジャイルの種を育てた取り組み

今、金沢に本店を置く北國銀行が、DXの先進企業として注目を集めています。地方銀行の枠にとらわれない「次世代版 地域総合会社」として、コンサルティングやBPO、ECサイトなど幅広い分野に次々と事業を拡大。お客様の持つ課題に対し、グループ会社が一体になって必要なソリューションを提供するというエコシステムを創出しています。

現在のDXにおける躍進の影には、ビジネスサイドがデジタルを重視し、全体最適やシステムの内製化を進め、何年もかけてアジャイル開発の種を育ててきた歴史があると言えます。今では「アジャイル」という言葉がグループ内に浸透しているという北國銀行。どのようにアジャイルと出会い、どのように組織にアジャイルを根付かせてきたのでしょうか。北國銀行のアジャイル推進に携わった方々に、お話をうかがいました。

### アジャイルが広まる組織の土壌

アジャイルやDXという言葉に北國銀行が出会うよりもずっと前、2000年頃に描かれていた大きな戦略の中で、すでに現在のベースとなる組織イメージが描かれていました。最初の段階として描かれていた戦略は、徹底的なコスト削減。そのために、業務のシンプル化や、全体最適に向けた動きが始まりました。このときのコスト削減の取り組みのひとつであったペーパーレス化（デジタルイゼーション）は、現在のDXの礎にもなっています。

その次の段階として描かれ、実行された「カスタマーセントリックな営業の追求」も、組織のあり方に大きな影響を与えました。ノルマを廃止し、それまでのプッシュ型営業を、顧客価値を起点に物事を考えるように方針転換。長い時間を要しながらも、目先の数字ではなく顧客に求められている行動をするように組織を変えていきました。これも、後の時代に顧客の価値を重要視するアジャイルが根付く土壌となるには、重要な変化のひとつでした。

### アジャイル開発への動きはシステムの内製化から

それから2010年頃、さらに北國銀行とアジャイルの距離を近づける、内製化の動きがありました。当時はアウトソーシング全盛期で、地方銀行は他行との差別化よりも「いかにコストを抑えて効率化するか」が焦点となっており、パッケージの利用が主流でした。しかし北國銀行は、独自の経営戦略を実現するためには独自のシステムが必要だということにいち早く気づき、2012年にはアウトソーシングをやめ、内製開発に舵を切ったのです。

それから内製開発を進めたものの、勘定系システムのリプレースを3年かけて進めた際に、ひとつの問題が発生しました。ウォーターフォールで3年前に決めた要件では、法改正等の変化に対応できなかったのです。

その経験から北國銀行は、2017年の個人向けインターネットバンキングシステムの内製開発では、必然的にアジャイルの手法を取り入れ始めました。特にインターネットバンキングにおいては、お客様のニーズは時代に合わせて次々と変わります。これまで通りウォーターフォールで事前にごっちり仕様を固めてしまっただけでは、変化していくセキュリティやスマホの対応ができません。それはつまり、お客様にとって魅力的なサービスを提供できないということになってしまいます。

北國銀行にとって、アジャイルは「アジャイル開発をする」という目的で始めたものではなく、何がお客様にとって大切かを考えて行動を変えていったところ、それが結果的にアジャイルの考え方にぴったりはまったのでした。

## <インタビュー> 銀行でのアジャイル実践例

インターネットバンキングの開発以降も、北國銀行は引き続きアジャイル手法を取り入れ、紆余曲折を経て組織内でアジャイルの芽を育てていきました。2019年には、システム開発のグループ会社「株式会社デジタルバリュー」を設立し、北陸に限らず全国からアジャイルマインドを持つ若いエンジニアを採用。様々なバックグラウンドや異なる意見を持つメンバーが加わり、組織に多様性も生まれました。さらにここ数年では、積極的に外に出てアジャイルやDXに関する講演なども行い、業界での存在感を高めています。

### トップからのメッセージで浸透していく、アジャイルというキーワード

アジャイルを目的にはせず、顧客価値を追い求めた結果に「アジャイル」という言葉が伴ってきたという北國銀行。2022年3月期の統合報告書にも「アジャイル」が何度も登場するほど、今となってはアジャイルという言葉が社内に浸透しています。ここまで浸透するには、トップからのメッセージが欠かせない存在でした。

現頭取の杖村修司氏は週に1回、「トップメッセージ」として、社内SNS上で全社員に向けてメッセージを発信し続けています。このメッセージはまさにアジャイルにおける「インセプションデッキ」に似た力を持ち、全社員が同じ方向を向くためのものとなっています。そして、その中でも「アジャイル」がキーワードとして自然と登場するのです。

さらに、北國銀行では「アジャイル勉強会」というシステム部を中心とした有志による学びの場が、主に非IT人材向けに開催されています。勉強会には、トップメッセージで「アジャイル」という言葉を知って関心を持った社員がなんと毎回数百人も参加。トップのメッセージと学びの場が繋がって、さらに組織内へのアジャイルの浸透を深めています。

## <インタビュー> 銀行でのアジャイル実践例

一般的にアジャイルでは、トップはサーバント型であること、ボトムアップの現場主導で動いていくことが推奨されています。しかし、もともとヒエラルキーがあり、アジャイルマインドを持った組織ではなかった北國銀行において、最初からボトムアップで動くことは困難でした。そのため、トップによる発信が変化への強い後押しとなっているのです。

### アジャイルをさらに根付かせるための、これからの取り組み

経済産業省のDX認定も受け、すでにDX推進企業として脚光を浴びている北國銀行ですが、まだまだ課題はあると言います。

まず、トップメッセージや勉強会による影響はあるものの、アジャイルの浸透具合は部署・組織によってバラつきがあり、まだまだ全社が一枚岩になっているわけではありません。カルチャーや考え方を合わせていくのは簡単なことではありませんが、北國銀行では逆に「今以上にアジャイルを浸透させればもっとできることが増える」チャンスだと捉えて、一緒に仕事をしながらアジャイルについて知ってもらうチャンスを増やすなど、さらなる全社的な浸透に向けて日々試行錯誤を続けています。

また、今後「次世代版 地域総合会社」としてますます銀行業務以外の広がりを見せていくにあたって、さらなる人材育成の必要も出てきています。以前より多様性が増してはいても、約8割が銀行員として入社して業務経験を積んでおり「銀行業務をずっと続ける」という銀行員としてのスキルとマインドがまだまだ強い状態。今後新しい事業をスタートするにあたって、ゼロからイチを作るのに適任な人材を育てていくことが重要なミッションとなっています。

## <インタビュー> 銀行でのアジャイル実践例

そんな中、嬉しいことに、銀行員として入社したメンバーの中にも、銀行業務以外にも挑戦してみたいという声が若手を中心に出てきています。そういった要望に応えるために、どのようなスキルがあればデジタル関係の業務に携われるかを示したスキルマップの作成も現在始められています。

アジャイルやDXの先進企業として注目を浴びる北國銀行。その背景には、長くデジタル戦略に取り組んできた歴史、顧客の価値を重視する組織作りづくり、また、アジャイルという言葉の浸透への様々な取り組みがありました。

その姿は、今後アジャイル開発を導入していく企業、DX推進していく企業にとっては、非常に学ぶことが多いのではないのでしょうか。これからも活動の幅を広げ、北陸の地から新たな挑戦を続けていく北國銀行の動向に期待が募ります。

### 取材協力：

岩間 正樹氏

株式会社北國銀行 執行役員 システム部長 / 株式会社デジタルバリュー代表取締役社長

横越 亜紀氏

株式会社北國フィナンシャルホールディングス 常務執行役員 人材開発部長 / 株式会社北國銀行 常務執行役員 人材開発部長 / 株式会社BPOマネジメント 取締役

寺井 尚孝氏 株式会社北國銀行 常務執行役員 デジタル部長



今年度のIPAアジャイルWGの成果物をお届けいたします。

アジャイルというアプローチが日本においても一般的に知られるようになり、ソフトウェア開発における選択肢として普通に取り上げられるようになってきました。それは良いことではあるのですが、小さからぬ問題を生むようになったと認識しています。特に、ソフトウェア開発にアジャイル観点での考え方・段取りの仕方を取り入れてScrumの儀式を回せば「良いソフトウェア」が作れるという一種の精神主義的な勘違いです。確かにマインドセットやカルチャーの変換は重要ですが、それだけではソフトウェアは作れません。

この問題に対処するにあたって、改めてアジャイルという文脈におけるコンピュータサイエンスやソフトウェアエンジニアリングの基本と実務で求められる実践的な技術・スキル・知見の重要性をガイドブックにする必要性を強く認識しました。

そのような問題意識の下で、アジャイルWGの中で1年間かけて議論し、実践者からのインタビュー実施と併せて、実践的な指針として整理することができました。その成果が、この「アジャイルプロジェクト実践ガイドブック」です。実践的な技術・スキルとして単なるテクノロジーというよりも、実務の観点で難所となるソフトウェア工学の基礎技術やプロジェクト管理、人材と組織のマネジメント、契約や法務といったテーマもポイントを整理して提示しています。みなさまの周りで行われているアジャイルプロジェクトに対する実践的な「ヒント」となることを祈って、お手元に届けられればと思います。

少しでもみなさまの組織で実践されるソフトウェア開発が、顧客とエンジニアの皆さん、そして開発チームの属する組織の幸せに繋がるように、お役立ていただければ幸いです。

2023年03月 アジャイルWG主査 羽生田栄一



**IPA** 独立行政法人  
情報処理推進機構



# アジャイルWGメンバー一覧

## 2022年度 アジャイルWGメンバー一覧

	氏名（敬称略）	所属
主査	羽生田 栄一	株式会社豆蔵/ 独立行政法人情報処理推進機構（IPA）
メンバー	川上 誠司	アクセンチュア株式会社
	関 満徳	エクスパッション合同会社
	和田 憲明	富士通株式会社
	渡会 健	株式会社マネジメントソリューションズ
	岡本 宗之	株式会社ITプレナーズジャパン・アジアパシフィック
	松崎 一孝	株式会社ふくおかフィナンシャルグループ
	藤井 崇介	株式会社星野リゾート
	高市 裕子	日本電気株式会社
オブザーバー	野村 治彦	独立行政法人情報処理推進機構（IPA）
	下河 啓示	日本電気株式会社
アドバイザー	平鍋 健児	株式会社永和システムマネジメント



編集（敬称略）

主 査：羽生田 栄一 株式会社豆蔵／  
独立行政法人 情報処理推進機構  
事務局：高橋 伸子 独立行政法人 情報処理推進機構  
藤中 紀孝 独立行政法人 情報処理推進機構



イラスト（敬称略）

黒澤 優姫

**ITSS+ アジャイル領域  
アジャイルプロジェクト  
実践ガイドブック  
- アジャイル実践者が語る、難所、スキル、育て方、  
法務そして実践例 -**

2023年4月 第1刷発行

発行 独立行政法人 情報処理推進機構（IPA）  
社会基盤センター人材プラットフォーム部  
スキルトランスフォーメーショングループ  
〒113-6591  
東京都文京区本駒込2-28-8  
文京グリーンコート センターオフィス  
(C)独立行政法人 情報処理推進機構

