

価値創造社会の持続的発展 のキギはアジャイルにあり

Ver2.0



目次

はじめに 2

各資料の概要 4

活用にあたっての留意点 5

I. なぜ、いまアジャイルが必要か？ 6

II. アジャイルソフトウェア開発宣言の読みとき方 11

III. ビジョンとプロダクトの橋渡し 26

IV. アジャイル開発の進め方 31

V. まとめ 35

[付録] プロダクト責任者：太郎の物語 37

[付録] ITSS+（プラス）の背景 40

アジャイルWGメンバー一覧 41

はじめに

独立行政法人情報処理推進機構（IPA）では、第4次産業革命に向けて求められる新たな領域の“学び直し”の指針としてITSS+（プラス）を策定しています。そして、デジタルトランスフォーメーション（DX）を踏まえたスキル変革の取組みをイメージするための参考として下記図で表しました。

ITSS+の1つである「アジャイル領域」は第4次産業革命を実現するために必要なアプローチでありながら、アジャイル開発そのものに関する適切な理解が十分浸透していないという問題意識がありました。そこで、2017年度よりアジャイルWGを組成し、「アジャイル領域」へのスキル変革のための指針づくりに取り組んできました。

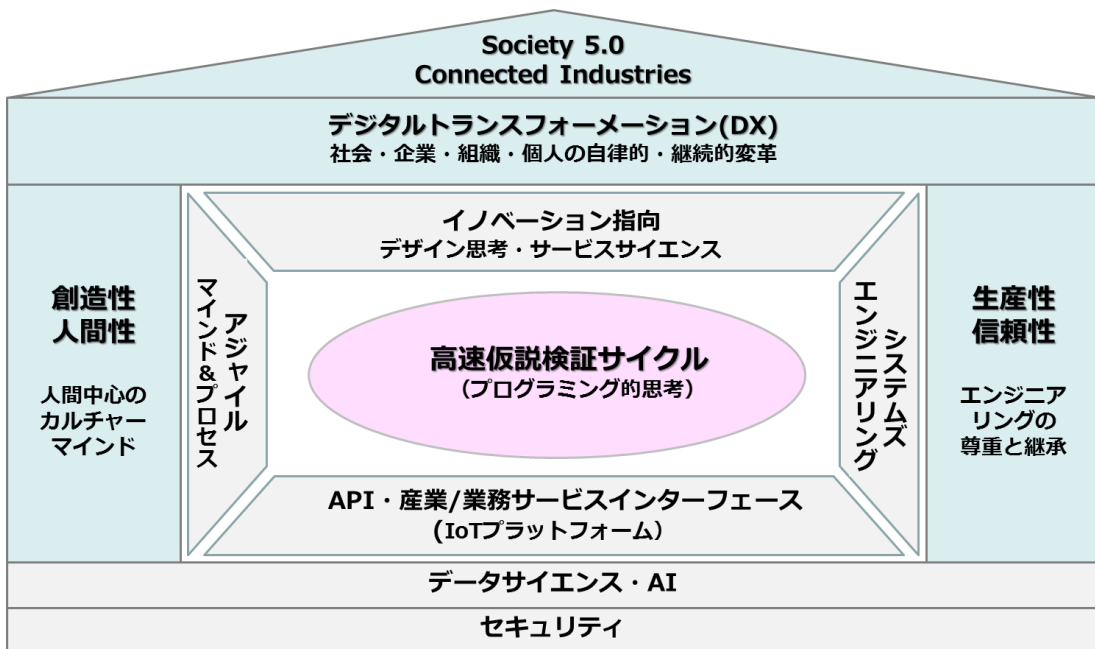


図1 『デジタルトランスフォーメーション(DX)に向けたスキル変革の方向性 -全体イメージ』

2017年度は、アジャイル開発のベースにあるマインドセットや原則、アジャイル開発プロセスやチームの特徴、および開発者の学ぶべきスキルについて検討を行い、**②**「アジャイルソフトウェア開発宣言の読みとき方」**④**「アジャイル開発の進め方」の2つを公開しました。

2018年度は、アジャイルの必要性を補足するとともに、ビジネスを主管する側にスポットをあてています。ビジネスを成功に導くためには、ビジネス側がどのようにふるまい、役割を果たすべきかを検討し、新たに ①「なぜ、いまアジャイルが必要か？」 ③「ビジョンとプロダクトの橋渡し」の2つを公開しました。

「アジャイル開発」を成功に導くためには、形だけ真似るのではなく、アジャイルの本質についての理解が必要です。開発者のみならず、経営層や事業部門の方にも、これらのドキュメントを通じてアジャイル開発の概要やそのマインドセットを理解していただきたいと考えています。

ITがビジネスの成功に欠かせない今、ビジネス側と開発側が相互に知恵を出し合い、弛まぬ努力を続けることが必要です。ビジネスの継続的改善をサポートするアジャイルを、これまでは開発に携わる機会が少なかったビジネス側も含めて、双方が正しく理解することが、競争力の向上につながります。

今後、アジャイル開発に関する正しい理解が浸透することによって、ソフトウェア開発の局面のみならず、営業や企画等のさまざまな業務を含めた経営そのものをアジャイルに遂行するような組織文化が醸成されることを期待しています。

本冊子は、「アジャイル領域」にて公開したドキュメントを読みやすく再編集したものです。

表1 『ITSS+／アジャイル領域へのスキル変革の指針』の構成

<p>アジャイル領域へのスキル変革の指針/本ドキュメントの位置づけ</p> <p>ねらい・ドキュメント構成</p> <p>概要と想定読者・留意点</p> <p>WGメンバー</p>	<p>② アジャイルソフトウェア開発宣言の読みとき方</p> <p>アジャイルソフトウェア開発宣言と背後にある原則</p> <p>アジャイルソフトウェア開発宣言に対する誤解と真意</p> <p>アジャイル宣言の背後にある原則の読み方</p> <p>アジャイルソフトウェア開発宣言の背後にある原則 原則1 - 原則12</p>	<p>③ ビジョンとプロダクトの橋渡し</p> <p>プロダクト責任者</p> <p>プロダクト企画</p> <p>プロダクト設計開発を導く</p> <p>プロダクト進化</p>	<p>④ アジャイル開発の進め方</p> <p>アジャイル開発のプロセス -アジャイル開発のプロセス(スクラムの例) -アジャイル開発の進め方の特徴 -役割(ロール)の特徴(スクラムの例) -開発の進め方と役割(ロール)の関連</p> <p>アジャイル開発チームのつくり方 -アジャイル開発チームのもつべきスキル -スキル一覧</p> <p>参考資料 <参考1>ウォーターフォール型ロールとアジャイル型ロールの比較表 <参考2>アジャイル開発の概念整理</p>
<p>① なぜ、いまアジャイルが必要か？</p> <p>社会の状況と問題を理解する</p> <p>アジャイルとの付き合い方</p>		<p>⑤ 参考文献</p> <p>参考文献一覧<メンバー推薦> (入門レベル/実践レベル/関連テーマ)</p>	

詳しくは
こちら



「アジャイル領域」関連資料について

<https://www.ipa.go.jp/jinzai/itss/itssplus.html#section1-4>

各資料の概要

各資料の概要と想定する読者は次の通りです。

①なぜ、いまアジャイルが必要か？

-概要：Society5.0時代になぜアジャイルが必要かを理解します。Society5.0時代に直面する問題と従来の問題との違いを踏まえ、いまの時代の問題の解法としてアジャイルが適していることを説明しています。

②アジャイルソフトウェア開発宣言の読みとき方

-概要：アジャイル開発のベースにあるマインドセットや原則について理解します。「アジャイルソフトウェア開発宣言」にある「4つの価値」と「12の原則」について検討メンバーの解釈を説明しています。

③ビジョンとプロダクトの橋渡し

-概要：いまの時代にプロダクトを価値として届けるために「プロダクト」の責任者に求められる役割を理解します。プロダクト責任者の必要性、役割、ふるまい方について説明しています。

④アジャイル開発の進め方

-概要：アジャイル開発のプロセスと開発者の役割について理解します。アジャイル開発プロセスの特徴やチームの特徴、および開発者の学ぶべきスキルについて説明しています。

表2 『各資料の概要と想定読者』

◎：主体、○：協働、△：参考

資料	概要	想定読者			
		経営層	事業部門	開発部門 / チーム	情報システム部門
①	なぜ、いまアジャイルが必要か？	◎	◎	◎	◎
②	アジャイルソフトウェア開発宣言の読みとき方	○	◎	◎	◎
③	ビジョンとプロダクトの橋渡し	○	◎	○	○
④	アジャイル開発の進め方	△	○	◎	◎



活用にあたっての留意点

「アジャイル」という用語について

本冊子では、ソフトウェア開発に適用されるアプローチとして「アジャイル開発」を説明しています。「アジャイル」とは、本来、考え方や手法など、多数の概念を含む用語であり、現在ではソフトウェア開発だけでなく、プロダクト（サービス含む）企画と保守を含めたDevOpsや、組織の運営や経営などにも適用されています。

「プロダクト」という用語について

本冊子では、顧客向けの製品やサービスのみならず、社内システム等の業務アプリケーションも「プロダクト」と総称しています。

従来型（ウォーターフォール型）開発とアジャイル開発について

従来型開発とアジャイル開発では、そもそもソフトウェア(システム)の作り方が異なります。どちらの開発手法も、メリット/デメリット、向き/不向きがありますので、両者を対比させて優劣をつけることは意味がありません。それぞれの特性を十分理解したうえで、適材適所で活用していく必要があります。

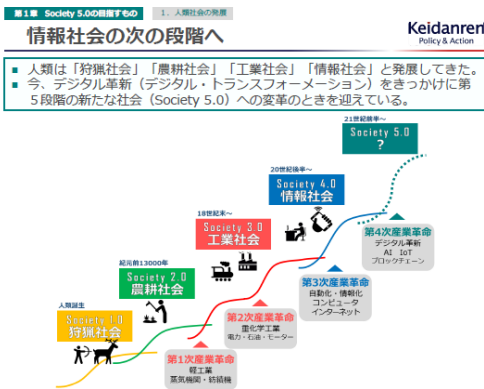
そのうえでアジャイル開発を採用する場合は、従来型の開発スタイルとは大きく考え方が異なるため、発注者と開発者はともに、取り組む意識や姿勢を変革させることが大切です。

今後、従来型開発を捨てて、アジャイル開発に切り替えていくということではありません。アジャイル開発に特有のスキルを身につける必要はありますが、これまで培ってきた様々な開発手法や工夫、ノウハウを尊重し、必要に応じて継承し活用していくことが必要です。

このような取組みの姿勢や変革は、個人に限った話ではありません。個々人が活動しやすい環境を、組織としてサポートしていくことも重要です。

1. なぜ、いまアジャイルが必要か？

1. 社会の状況と問題を理解する



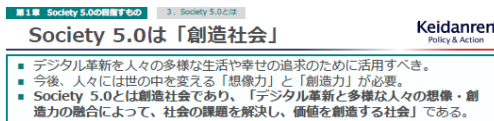
出典：Society5.0 ーともに創造する未来 一般社団法人 日本経済団体連合会
<http://www.keidanren.or.jp/policy/2018/095.html>

私たちの社会は、「狩猟社会」「農耕社会」「工業社会」「情報社会」と発展してきました。今、デジタル革新（DX）をきっかけに第5段階の新たな社会への変革のときを迎えています。その社会を、日本は「Society5.0」と名づけ、実現に向けた取り組みを進めています。

Society5.0とは、「デジタル革新と多様な人々の想像・創造力の融合によって、社会の課題を解決し、価値を創造する社会」です。

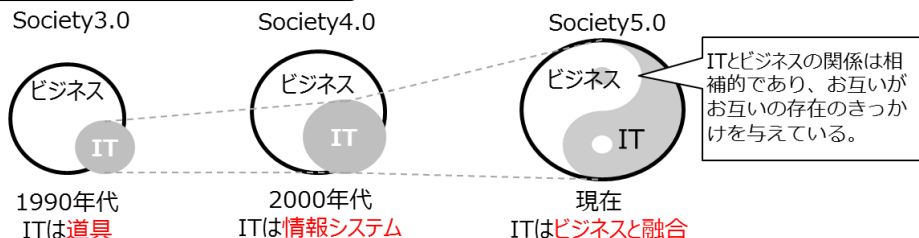
Society5.0時代は、変化が激しく、物事の予測が困難です。さらにそれらが複雑に関係しあっているため、はっきりとした答えが出しにくい、複雑かつ不確実な世の中になっています。

そうした中、我々に必要になるのは、社会に生きる多様な人々の生活や活動を理解し、そこに散らばる多様なニーズや課題や新しいアイデアを読み取る豊かな「想像力」、みんなのニーズや課題とアイデアをつなぐあたらしいシナリオをデザインする「創造力」です。そこに、デジタル技術やデータを活用できる場・環境を融合して、デジタル革新を多様な人々の想像力・創造力と融合することで、「課題解決」を図るとともに、我々の未来をより明るいものへと導く「価値創造」をもたらします。



出典：Society5.0 ーともに創造する未来 一般社団法人 日本経済団体連合会
<http://www.keidanren.or.jp/policy/2018/095.html>

ITとビジネスの関係性の変化

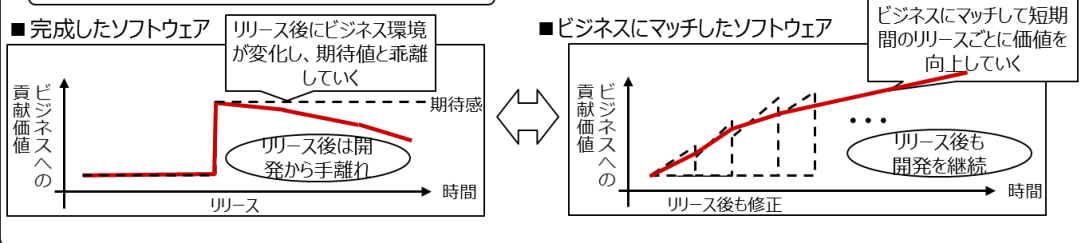


出典：理想的なソフトウェア開発の実現 Microsoft Download Center をベースに改変

図2 『ITとビジネスの関係性の変化』

近年、ITはビジネスにとって不可欠だけでなく、一体化した存在になっています。業務プロセスの効率化はもとより、新規事業やサービスをスピード感をもって創出しタイムリーに市場や社会と適応するためにも、ITとビジネスが融合することが求められています。

ビジネス価値を提供しつづける



出典：理想的なソフトウェア開発の実現 Microsoft Download Center をベースに改変

図3 『ビジネス価値を提供しつづける』

ITとビジネスの融合は、プロダクト（サービス含む）の提供形態にも変化をもたらします。

特に近年のプロダクトは、市場投入後に顧客の反応を反映しながら、プロダクトを繰り返し修正しながらビジネス価値を向上します。場合によっては、ビジネス価値の評価基準自体もリリース時と変わることもあります。適宜優先順位を考慮しながら価値を向上しつづけていく必要があります。

2. アジャイルとの付き合い方

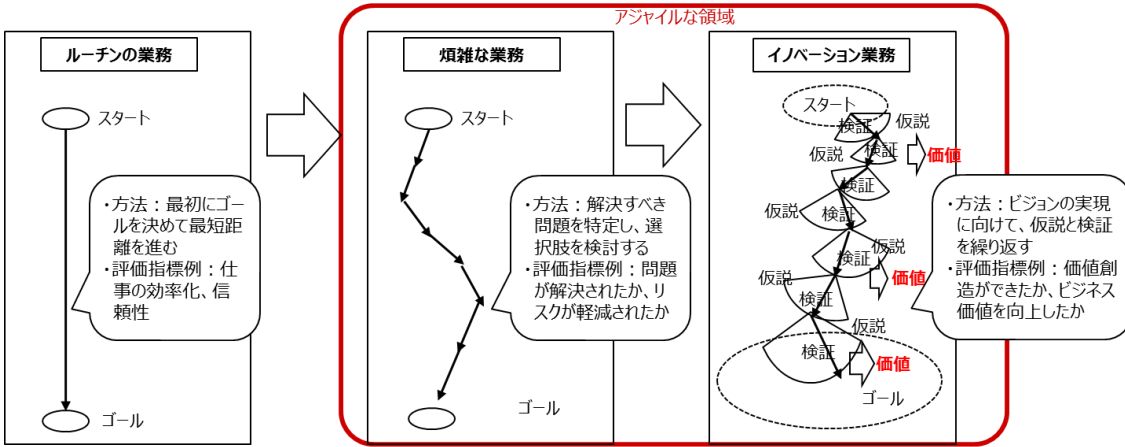
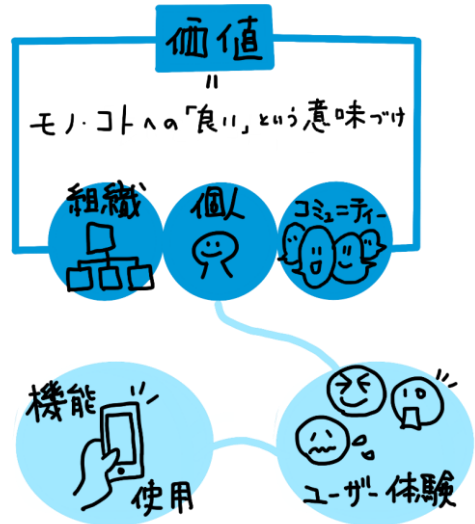


図4 『探索と適応による価値創造のアプローチ』

複雑で不確実な社会や市場では、様々な専門や経験をもつ多様なチームが必要になり、イノベーションを起こしやすい心理的安全性の確保および、競争と共創のバランスが重要です。また、業務の状況によって、問題解決のアプローチもダイナミックに変えていく必要があります。

個人や組織やコミュニティが「良い」と意味づけているモノ・コト・ビジョンとの擦り合わせを常に行い、観察し実験しながら探索し、仮説を立てては検証し、状況に適応しながらより高い価値を生み出していく活動と言えます。



(1) 価値駆動（アジャイルマインド）

ビジネス価値・社会価値の実現を主眼とし、探索と適応を繰り返します。価値を届けるチャンスを見極め、複数の仮説から選択し、試行します。結果を振り返り、さらなる価値の向上につなげ、これらを繰り返します。

(2) アジャイルな活動

① プロダクト企画：

プロダクトのビジョンを語る

ビジネス側主体で問題の探索、ビジネスの設計をアジャイルに実施します。有効な指標に基づいて仮説検証サイクルを回します。

② プロダクト設計開発：

ビジョンを実現するプロダクトを創る

価値のあるプロダクトを速く継続的に提供するため、アジャイルに開発を進めます。常にやり方を最適に調整します。

③ プロダクト進化：

社会に価値を届け社会の声を聴く

ビジネス側、開発側が協働して、価値あるプロダクトを速く継続的に提供します。研究開発（知識創造）と同様に、実験と学習を繰り返します。

上記の活動を通じて、**新たな未来をつくり**あげていきます。



図5 『価値駆動とアジャイルな活動』

「アジャイル (agile)」の名詞形である「アジリティ (agility)」とは、もともと「敏捷」や「機敏」という意味ですが、ITやビジネスの文脈では、方針の変更やニーズの変化などに機敏に対応する能力を意味します。

激しく変化する環境に対して、私たち自身がパラダイムシフトすることが必要です。また、常にあるべき姿に向けて改善、進化しつづけることが必要です。

単にアジャイル開発の手法を採用し、アジャイル開発のプラクティスを行っていても、それはアジリティを考える上で一部の要素にすぎません。形式だけアジャイル型の手法を採用していても、個人やチームがアジャイルにふるまえなければ意味がありません。

アジャイルな活動は、何かのプロダクトをつくるということのみならず、価値創造のあり方をよりよくしていきます。すなわち、アジャイルな活動を続けることによって、プロダクトのつくり方（プロセス）を進化させ、プロダクトをつくる人を成長させます。

アジャイルな活動への取組みは、Society5.0という価値創造社会を実現し、持続的に発展させるための重要なカギとなります。

アジャイルな活動は、決して楽なものではありません。何度も失敗し、学び合い、教え合い、成長していくことが何より大事なことになります。

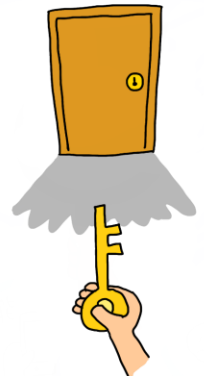
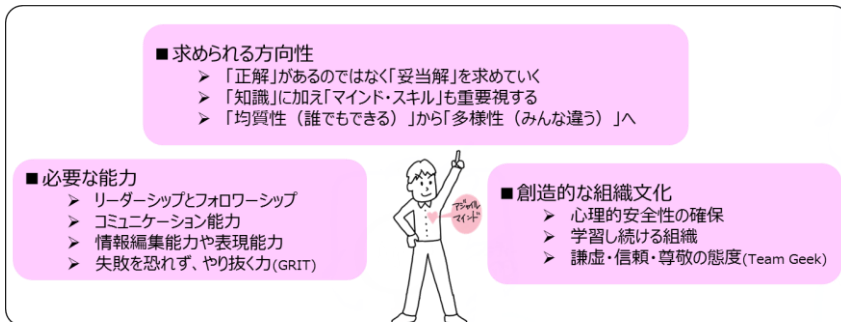


図6 『価値駆動に必要な要素』

II. アジャイルソフトウェア開発宣言の読みとき方

1. アジャイルソフトウェア開発宣言とアジャイル宣言の背後にある原則

「アジャイルソフトウェア開発宣言」は、従来型のソフトウェア開発のやり方とは異なる手法を実践していた17名のソフトウェア開発者が、それぞれの主義や手法についての議論を行い、2001年に公開されました。そこには、彼らがソフトウェア開発を行ううえで重視している「マインドセット」が書かれていました。その後、このマインドセットは、世界中のソフトウェア開発者達に支持され、「アジャイルソフトウェア開発」という名で世に広まりました。

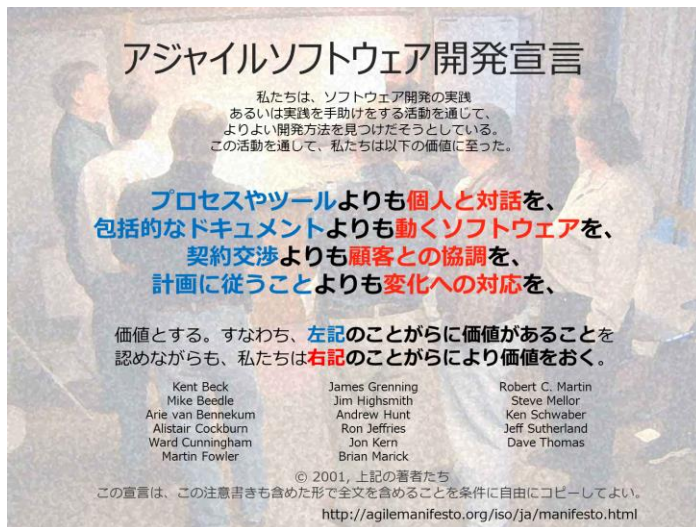
一方「アジャイル宣言の背後にある原則」は、「アジャイルソフトウェア開発宣言」で表明されているマインドセットを実現するために、従うことが望ましい原則、つまり根本的・根源的な取り組み姿勢について、書かれています。

下図は、本冊子で示している人物の関係図を表しています。



図7 『アジャイルソフトウェア開発宣言の読み解き方 人物関係図』

1. 「アジャイルソフトウェア開発宣言」に対する誤解と真意



アジャイルソフトウェア開発宣言で伝えようとしていることは、まずマインドセットがあって、そのうえで「プロセスやツール、ドキュメント、契約、計画」を考えるべきである、ということなのです。このマインドセットは「個人と対話、動くソフトウェア、顧客との協調、変化への対応」として簡潔に表現されていますが、後述の「原則」を踏まえてもう少し具体的に表すと、次のようなことを示しています。

「対面コミュニケーション」

個人同士の対話を通じて相互理解を深めることで、よりよいチームが作られる。

「実働検証」

動くソフトウェアを使って繰り返し素早く仮説検証をし、その結果から学ぶことがよりよい成果を生み出す。

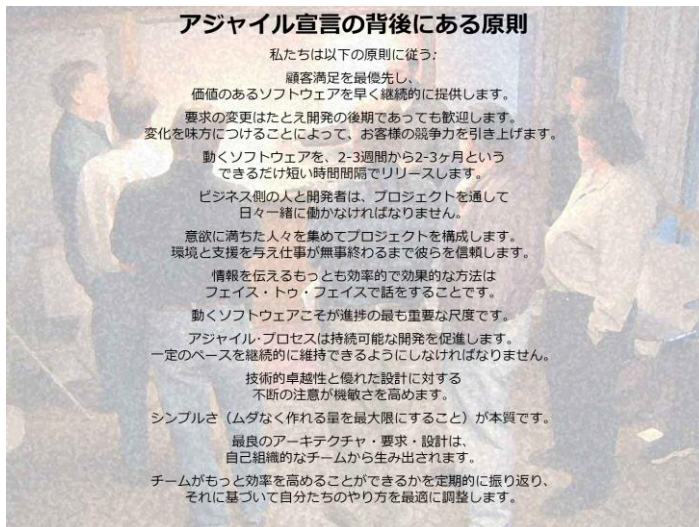
「顧客とのWin-Win関係」

お互いの立場を超えて協働することにより、よりよい成果と仕事のやり方を作ることができる。

「変化を味方に」

顧客のニーズやビジネス市場の変化は事前計画を狂わす脅威ではなく、よりよい成果を生み出す機会と捉える。

2. 「アジャイル宣言の背後にある原則」の読みとき方



次ページに示す「アジャイル宣言の背後にある原則」については、重要なポイントが一読して分かるタイトルをつけたうえで基本的な考え方と行動規範を加えて原則の読みとき方を記述しています。

基本的な考え方

各原則で重要となる「基本的な考え方」について説明しています。基本的な考え方には、その原則の本質的な解釈のほか、その原則がある理由、その原則を実現するための手段などについて記載しています。アジャイルに馴染みのない方でも理解できたり実感できるよう、平易な言葉で説明をしていますので、誰もが容易に基本的な考え方を理解できるようになっています。

行動規範

「行動規範」では、各原則を実践するうえで、どのような行動が望ましいか、またどのような行動が望ましくないか説明しています。各原則の行動規範を通じて、今までの行動をふりかえり、今後はどのように行動すればよいか考えるきっかけになれば幸いです。

顧客の満足を求め続ける

オリジナル

「顧客満足を最優先し、価値のあるソフトウェアを早く継続的に提供します。」

“Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.”

基本的な考え方

ビジネス側の人と開発者にとって、最も重要な活動は、価値のあるソフトウェアを素早く継続的に提供し、顧客に（QCDの達成ではなく）ビジネスゴールの達成の観点で満足してもらうことです。

日本の多くの企業は、近年の劇的な市場の変化への対応に迫られています。ビジネスを支えるITシステムの構築は、企業の存続を左右するほど重要な要素になっています。ビジネス側の人と開発者は、ビジネスの実現を最優先に考えてITシステムを構築する必要があります。ただし、ビジネスの成否はビジネスごとに異なります。したがってビジネス側の人と開発者は、常に顧客と同じ目線で、顧客の満足を高める価値の創出について考える必要があります。

まず、顧客の満足とは何かを定義することからはじめ、ビジネスの観点で十分に評価可能な動くソフトウェアを、素早く継続的に提供し、顧客の満足度がどのような状態なのか、明らかにします。

行動規範

- ビジネス側の人と開発者は、ビジネスゴールについて初期の時点で徹底的に話し合しましょう。
- ビジネス側の人と開発者は、QCDの達成ではなく、ビジネスゴールの達成を目指しましょう。
- ビジネス側の人と開発者は、ビジネスゴールに向かっていくかどうかを常に確認し、方向がズレていたら素早く修正しましょう。
- ビジネス側の人と開発者は、ソフトウェアで実現すべき要件を顧客の立場で一緒に考えましょう。顧客が必ずしも正解を持っているわけではありません。
- ビジネス側の人と開発者は、顧客が価値を確認しやすい単位に分割しましょう。分割した単位を、素早く継続的に、顧客に提供し続けましょう。

要求の本質を見抜き、変更を前向きに

オリジナル



「要求の変更はたとえ開発の後期であっても歓迎します。変化を味方につけることによって、お客様の競争力を引き上げます。」

“Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.”

基本的な考え方

改善に繋がる要求は、新しい価値を見つけた証拠です。勇気を持って変更を受け入れることで、変化に強くなり、企業の競争力を高めることに貢献します。

従来は、定型業務の処理や記録を扱う基幹系システムなどのビジネスが多く、最初から作るものが明確なシステムが大半でした。しかし昨今、人とモノの繋がりを重要とする業務改善や変革を目的としたビジネスは、最初から要求に明確な答えがないことが多く、作るものが不明確な場合があります。このような、特性が異なるビジネスのITシステム開発を、従来と同じソフトウェア開発プロセスで実施しても、成功確率は上がりません。

そもそも、顧客にとって価値のないシステムを作っても、意味はありません。そこでたとえ開発の後期であっても、要求の本質を見抜き、与えられたコスト・納期・スコープのバランスと要求の優先順位から、変更の受け入れを判断する必要があります。

行動規範

- ビジネス側の人と開発者は、変化を「新しい価値」の発見の機会ととらえましょう。
- ビジネス側の人と開発者は、改善に繋がる要求を生み出すことに努めましょう。
- ビジネス側の人と開発者は、価値があるのであれば、積極的に変更を受け入れましょう。
 - ただし顧客に言われるがままに変更を受け入れることはやめましょう。
- ビジネス側の人と開発者は、間違いについてもオープンに議論しましょう。
 - 例えば数分で修正できる小さな間違いが、後までそのままにしておくことにより、リリース日を数週間遅らせる原因になります。
- ビジネス側の人と開発者は、よりよいフィードバックを生み出すようリリースをすることに努めましょう。

成果物を2-3週間で、リリースし続ける

オリジナル



「動くソフトウェアを、2-3週間から2-3ヶ月というできるだけ短い時間間隔でリリースします。」

“Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.” ”

基本的な考え方

ビジネス側の人々が積極的にビジネスの舵取りができるよう、ビジネス側の人と開発者は2-3週間の短い時間間隔で、成果物を定期的にリリースし続けます。

リリースするというと、比較的長期間（例えば半年から数年など）の開発の後に、顧客に納品することを思い浮かべるかもしれません。しかし、長い開発期間中に当初顧客が望んでいたゴールが変わり、顧客が本当に欲しかった成果を得られなくなっている可能性もあります。

この原則03という短い時間間隔とは、2-3週間から2-3ヶ月です。ビジネスゴールが変わって要求が変更になることが想定される開発の場合は、仮説検証型で進める必要があります。成果物を頻繁にリリースすることで、顧客からフィードバックを得ることができ、結果として顧客が本当に欲しいものを作ることができます。また、たとえ後戻りがあったとしても小さくすませることができます。

行動規範



- ビジネス側の人と開発者は、要求は不確実であり変化することが前提であることを共有しましょう。
- 開発者は、価値に結びつかない中間成果物ではなく、ビジネス観点で評価できるものを提供しましょう。
- ビジネス側の人と開発者は、成果物のリリース間隔をビジネスの観点から決めましょう。
- ビジネス側の人と開発者は、市場からのフィードバックを共有しましょう。


全員で共通の目標に向かおう

オリジナル



「ビジネス側の人と開発者は、プロジェクトを通して日々一緒に働かなければなりません。」

“Business people and developers must work together daily throughout the project.”

基本的な考え方 

ビジネス側の人と開発者が共通目標に向かって一緒に働くことで、一連のリリースおよびフィードバックのサイクルが円滑になります。

不確実性の高いプロジェクトでは、状況が刻々と変化するため、共通の目標が見えないまま進めると、認識の齟齬などが発生する可能性があります。

ビジネスの価値を最大化するために、作るソフトウェアの価値やプロジェクトの状況について、常に共通目標に向かってベクトルを合わせながら働きましょう。そのために、常にあるべき姿をビジネス側の人と開発者が共有し、必要に応じて改善し続ける必要があります。

行動規範 

- ビジネス側の人と開発者は、全員一緒の場所で働くことを理想としましょう。
 - 一緒の場所で働けない場合は、工夫して、方向性を共有しましょう。
- ビジネス側の人と開発者は、一緒にビジネスを成功に導くという共通目標を、日々確認しましょう。
- ビジネス側の人と開発者は、全員で双方向のコミュニケーションが図れるようにしましょう。
 - ビジネス側の方は、要求の出しっぱなしはやめましょう。
 - 開発者は、フィードバックをもらうために、頻繁にリリースをすることから始めましょう。
- ビジネス側の人と開発者は、全員が共通認識を持つために、必要最低限のドキュメントを書きましょう。

人の意欲は信頼から

オリジナル



「意欲に満ちた人々を集めてプロジェクトを構成します。環境と支援を与え仕事が無事終わるまで彼らを信頼します。」

“Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.”

基本的な考え方

プロジェクトを成功させるため、開発者には意欲のある前向きな人々を集めて、彼らが働きやすい環境を整えることが求められます。

個人が十分に高い能力を備えていたとしても、お互いの関係がぎくしゃくしているなど、働きにくい環境では十分に能力を発揮できません。もともと日本には「和」を重んじる文化、精神があり、経営やものづくりの局面にも活かされてきました。「対立」を避け、相互にリスペクトし信頼し合うことで、ビジネス側の人と開発者、そして組織内の風通しがよくなり、ビジネス側の人と開発者全員が意欲的に活動できるようになるのです。

行動規範

- ビジネス側の方は、開発者を信頼して仕事を任せ、気持ちよく働けるよう配慮しましょう。
 - 細かな作業指示は、開発者の自律的な行動を阻害します。
- ビジネス側の人と開発者は、顧客の価値創造のために、自らが考え、自らの行動と判断に責任を持ちましょう。
- ビジネス側の人と開発者は、自分の作業だけでなく、顧客の価値創造のために何ができるかを考えましょう。
- 開発者は、積極的にデモを行い、顧客からのフィードバックを受けましょう。
- ビジネス側の人と開発者は、お互いが協調し、集中して作業できる環境を作りましょう。


コミュニケーションは直接対話で

オリジナル



「情報を伝えるもっとも効率的で効果的な方法はフェイス・トゥ・フェイスで話をすることです。」

“The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.”

基本的な考え方 

お互いの本音をフェイス・トゥ・フェイスで汲み取り、直接対話することが大切です。

ビジネス側の人が開発者に伝えたい情報や開発者間で伝え合う情報の中には、仕様とは異なり、目的、希望、夢、心配ごとなど、正確に表現できないものもあります。そのような情報は、言葉では伝わりにくかったり、伝える際に齟齬や誤解を起こしやすかったりします。

ビジネス側の人と開発者が、全員同じ空間に集まり、対面で双方向に対話することで、お互いの本音を汲み取りやすくなります。もし齟齬が起きたとしても、対話を通してすぐに気づくことができます。

行動規範



- ビジネス側の人と開発者は、必要な人たちが集まり、双方向の議論をすることを心がけましょう。
 - 決まった仕様を伝えるだけのような一方通行はやめましょう。
- ビジネス側の人と開発者は、他の役割の人と話すことで自分の範囲を広げ、これまで気づいていなかったことにも気付けるようにしましょう。
 - 自分の役割を、狭く、固定しすぎないようにしましょう。
- ビジネス側の人と開発者は、何か気になることがあったら、直接会話をしましょう。
- ビジネス側の人と開発者は、メールだけの伝達は対話より非効率であることを共通認識としましょう。
- ビジネス側の人と開発者は、しぐさや表情から、言葉にならない本心（心配ごと、違和感）を感じ取りましょう。

進捗も品質も現物で

オリジナル



「動くソフトウェアこそが進捗の最も重要な尺度です。」

“Working software is the primary measure of progress.”

基本的な考え方

進捗状況を正確に把握するためには、動くソフトウェアを確認することが最も信用できるやり方です。なお、この場合の進捗状況とは、求められている価値をどれだけ実現できているか、ということです。また、この場合の動くソフトウェアは、計画通りに進んでいるかということではなく、市場にリリースできる程度の品質を満たしている必要があります。

設計書の完成度など、動くソフトウェア以外のもので測る進捗では、実際に動かしてみるまでは問題に気づけないことがあるため、どうしても表面上の進捗状況把握に留まってしまいます。

不確実性の高いプロジェクトでは、進捗状況を常に正確に把握し、適切な判断を行うことが重要です。

行動規範



- 開発者は、動くソフトウェアを早いタイミングで見せられるように、小さな単位でモノを作る計画を立てましょう。
- ビジネス側の人と開発者は、顧客の期待に合っているか判断します。そのための受入基準をビジネス側の人と開発者で作りましょう。
- 開発者は、受入基準を満たすソフトウェアを作りましょう。
- ビジネス側の人と開発者は、動くソフトウェアに求められる必要な品質が担保されているべきと考えましょう。
- 開発者は、開発中の仕掛りを極力少なくしましょう。
- 開発者はパーセントで進捗を測るのではなく、実物に対する受入基準の合格/不合格で測りましょう。
- ビジネス側の人と開発者は、報告書に時間をかけることはせず、実物を触って進捗を測りましょう。

一定のペースでプロジェクトにリズムを

オリジナル



「アジャイル・プロセスは持続可能な開発を促進します。一定のペースを継続的に維持できるようにしなければなりません。」

“Agile process promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.”

基本的な考え方



無理のない一定の開発ペースを保つことは、ビジネスの成功につながります。

往々にして開発者は、ゴールを目指して過負荷なパフォーマンスを求められることがあります。無理をした状態では創意工夫をしたり、改善をしたりという意欲が起きません。また、無理をして一時的に生産性を上げたとしても、その無理が祟ってその後の生産性が落ち、結果的にトータルパフォーマンスは低下してしまいます。

開発者の生産性を最大限に引き出し、価値あるものを提供し続けるためには、開発者だけでなく、ビジネス側の人も心身ともにベストな状態を保ち、無理のない一定のペースを維持することが、重要です。

行動規範



- ビジネス側の人は、機能を小さくし、短い期間で価値を提供できるようにしよう。
- 開発者は、タスクを小さく均一化し、一定期間内に優先順位の高いものから開発することを、繰り返しましょう。
- ビジネス側の人と開発者は、定期的に繰り返す「リズム」を大切にしましょう。
- ビジネス側の人と開発者は、一定のペースを保つために、生産性を見える化し、適切に計画を見直しましょう。
 - 持続可能なペースは、開発者ごとに異なります。
- ビジネス側の人と開発者は、コミュニケーションを密にし、お互いの考えを共有して助け合いましょう。
- ビジネス側の人と開発者は、無理が生じている箇所（ボトルネック）を見つけ、助け合って改善しましょう。


よい技術、よい設計、よい品質の追求

オリジナル



「技術的卓越性と優れた設計に対する不断の注意が機敏さを高めます。」

“Continuous attention to technical excellence and good design enhances agility.”

基本的な考え方 

新しいビジネスを支えるソフトウェアは、状況に応じて、いつでも素早く変更できることが求められています。そのためには、優れた技術力に支えられた優れた設計になっている必要があります。よい品質には、高い保守性が欠かせません。

もし、設計やソースコードの可読性などについての技術的な問題が蓄積されると、ソフトウェアの保守性が低下し、素早く変更できない状態に陥ります。すると市場の変化への対応が遅くなり、ビジネスを成長させる際の大きな阻害要因となってしまいます。

よりよい技術を追い求め、効果的にその技術を活用することで、素早く、品質よく、変化に対応し続けることができます。技術は常に進化しています。開発者は技術力を磨き、プロダクトへの適用技術と設計に対して、常に注意を払うことが重要です。

行動規範 

- 開発者は、常に最新の技術を学び、効果的な技術はプロダクトへ適用しましょう。
- 開発者は、ソースコードの状態を目視やツールで常に診断し、常に変更して高い可読性を維持しましょう。
 - 可読性を高めると、多くの場合、コードの行数は減ります。
- 開発者は、自動化されたテストを作成して常に実行し、問題をすぐに発見しましょう。
- ビジネス側の人と開発者は、全員上記の行動に価値があることを認めましょう。

ムダ＝価値を生まない、を探してヤメる

オリジナル



「シンプルさ（ムダなく作れる量を最大限にすること）が本質です。」

“Simplicity--the art of maximizing the amount of work not done--is essential.”

基本的な考え方

ビジネス側の人と開発者は、求める価値が何かを全員で考え、その価値に直結しない作業や機能をムダと定義し、そのムダを減らすためにたゆまぬ努力をしましょう。これは、アジャイル開発における本質の活動です。

ムダの中には、組織にとって現在必要なものも含まれるので、見逃しがちです。これまでの常識や慣習にとらわれず、新鮮な目で現状を直視しましょう。

日々の作業の中から、全員がムダを見つけ、削減することは、身軽になり機敏性を高めるだけでなく、コスト削減と生産性向上にも貢献します。

行動規範

- ビジネス側の人と開発者は、顧客の要求を鵜呑みにせず、「その要求は本当に必要ですか？」と聞きましょう。
 - 顧客にとって本当に必要か分からない機能を、先に作るのは「つくりすぎのムダ」です。
- ムダについて、真剣に話し合しましょう。
 - 例えば、ある作業を100倍して、その価値も100倍になっているか考えてみましょう。
- 顧客に価値をもたらさない会議や報告を、なくしましょう。
 - 手始めに、進捗会議は短くしましょう。報告資料はシンプルにしましょう。


よいモノはよいチームから

オリジナル



「最良のアーキテクチャ・要求・設計は、自己組織的なチームから生み出されます。」

“The best architectures, requirements, and designs emerge from self-organizing teams.”

基本的な考え方 

最良のアーキテクチャ・要求・設計は、最良のチーム(=自律型のチーム)から生み出されます。自律型のチームとは、メンバー一人ひとりが自らの行動、作業に責任を持つとともに、お互い開発者同士の連携により、チームとしての成果を最大限に発揮することができるチームのことです。

優秀なメンバーが集まったチームだとしても、お互いに連携や協調がなければ、チームの能力は、個々のメンバーの能力の和でしかありません。あるいは、個々の作業が重複していたり、相反する作業をしていたりすれば、チームとしての成果に悪影響を与えます。

一方自律型のチームでは、メンバー一人ひとりが常に改善の意識を持ち、お互いの能力を高めながら共通の目的や目標に対しての取り組みが継続されます。そこには相乗効果が生まれ、チームとしてメンバー個々の能力以上の成果を発揮することが可能になります。その結果、自律型のチームから、ベストな成果が生み出されることになるのです。

行動規範



- チームは、プロジェクトの達成すべき目的や目標と、それを支える規律・ルールを共有しましょう。
- チームは、自分たちで規律・ルールを決めましょう。
- メンバーは、自ら率先して作業の改善や課題の解決に取り組みましょう。
- メンバーは、自分の不得意な作業、領域についてもチャレンジして取り組みましょう。
- メンバーは、お互いの作業状況をオープンにし、共有しましょう。
- メンバーは、困っているメンバーがいたら、助け合いましょう。

自分たちのやり方を毎週、調整する

オリジナル

「チームがもっと効率を高めることができるかを定期的に振り返り、それに基づいて自分たちのやり方を最適に調整します。」

“At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.”

基本的な考え方

チームが成長し続けるためには、定期的に(長くても2~3週間、オススメは毎週)ふりかえり、自分たちのやり方を調整し続けることが大事です。

“ふりかえり”というと、プロジェクトの最後の反省会を思い浮かべるかもしれませんが。しかしたいていの場合、次のプロジェクトは周りの環境もチームも異なることが多く、“ふりかえり”の結果を活かすことは難しいでしょう。

この原則で言う“ふりかえり”とは、プロジェクトの進行中に定期的に、短い時間で行うことです。

“ふりかえり”を繰り返すことで、よいやり方はどんどんよくなり、もし、失敗したとしても(うまくいかなかったとしても)影響を最小限に留めることができます。定期的な“ふりかえり”の場があることで、チームメンバーは、プロジェクトにとって有用ではあるが言いにくいことを言えるようになります。

行動規範

- チームとして、主体的にふりかえりましょう。
- “ふりかえり”にかける時間は、毎週の場合、30分~1時間程度を目安にしましょう。
 - だらだら続けることはやめましょう。
- “ふりかえり”のルールはチーム全体で決めましょう。
- チームで問題に立ち向かいましょう。
 - 個人の責任を追究してはいけません。
- 必要に応じて周りに協力をあおいで解決しましょう。
- 拳がった課題をすべて解決しなくても気にしないようにしましょう。
 - 失敗は許容しましょう。すぐ改善すればよいのです。

III. ビジョンとプロダクトの橋渡し

1. プロダクト責任者 ～ビジョンとプロダクトの橋渡し～

プロダクト責任者*は、顧客とともに価値を共創する活動の「成果」を最大化します。プロダクトの企画立案、デザイン、開発、テスト、リリース、マーケティング、ビジネスに至る幅広いプロセスにおいて、プロダクトの成功に責任を持ちます。

また、それぞれの役割の関係性を適切に調整する責任をもちます。特に、プロダクト組織として、無駄のないプロダクトの開発や進化に専念できる環境の構築・維持が重要です。

・ 顧客

生活や業務において、プロダクトを利用し体験する役割、もしくはその可能性を持つ役割であるとともに、プロダクト進化を促進する共生相手

・ ビジネスチーム

プロダクトに付随するビジネス全般について考える役割

・ プロダクト開発チーム

プロダクトの設計開発（ソフトウェア・UXデザイン）をプロダクト責任者とともに行う役割

・ プロダクト進化チーム

プロダクトをデリバリーし、対価を得る、顧客と対話することによって、プロダクトを顧客やマーケットに適應させ続ける役割

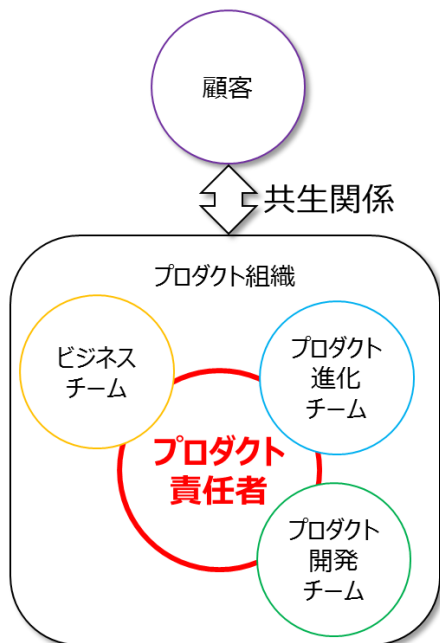
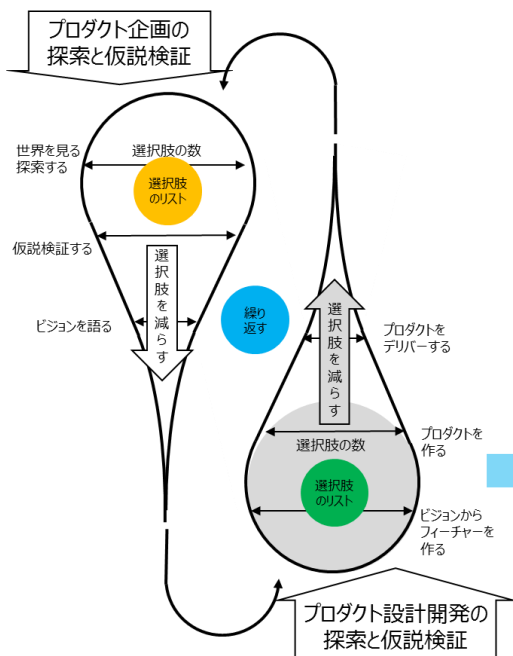


図8 『プロダクト責任者のおかれた組織と関係者』

*プロダクト責任者とは、本資料では、一般的なプロダクトマネージャやアジャイル開発におけるプロダクトオーナーの両方を広く含む概念として設定しています。実際のプロジェクトで適用する場合は、それぞれの組織文化に応じて、役割や内容を読みかえて理解してください。

プロダクト責任者は、プロダクトに対する視点を持ってすべての活動（プロダクト企画、プロダクト設計開発、プロダクト進化）に関与します。これら3つの活動の中でプロダクト責任者が演じるべき役割が存在します。

これら3つの活動の中での役割は、プロダクトに対するそれぞれのステークホルダー視点との対応を抽出したものです。お互いに協働し合い、それぞれの視点でビジョンを共有しながら、顧客との関係やプロダクトを進化させていきます。



プロダクト企画と設計開発における選択肢とダイナミズムを現わす左図を簡素化して下図のように表記しています。

プロダクト企画では、世界を歩き回り探索し、世界を理解します。あいまいで多くの選択肢がある状態から、仮説検証を繰り返し、ビジョンを練り上げていきます。

プロダクト設計開発では、共有したビジョンとともに、設計を行います。開発に必要な環境を整え、価値の高いものからプロダクトとして形にしていきます。

プロダクト進化では、全体を通じて、ビジョンとプロダクトのあいだでダイナミックに「間あい」を取り合いながら、価値を高め続けます。

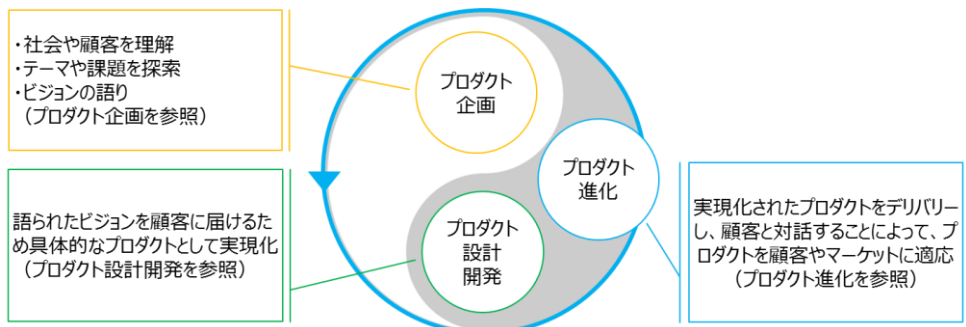


図9 『プロダクト責任者のダイナミックなふるまい』

2. プロダクト企画 ～プロダクトのビジョンを語る～

プロダクト企画とは、世界の状況をよく見て、あるべき姿（ビジョン）を示し、その価値について語ることです。

あるべき姿の具体的なモデルであるビジョンは、暗黙的もしくは明示的に世界の影響を受けています。同時に、あるべき姿（ビジョン）によって見ている世界も変わってしまいます。そのため本資料では、A) 世界を理解する、B) ビジョンを描き、価値を語る、の二段階としました。

A) 世界を理解する、B) ビジョンを描き、価値を語る、この試みを繰り返します。これらの段階を何度も繰り返すことによって、ビジョンと世界をダイナミックに作っていきます。ビジョンが具体的になり、関係者の解釈にバラつきが生じずに共有されることを目指します。

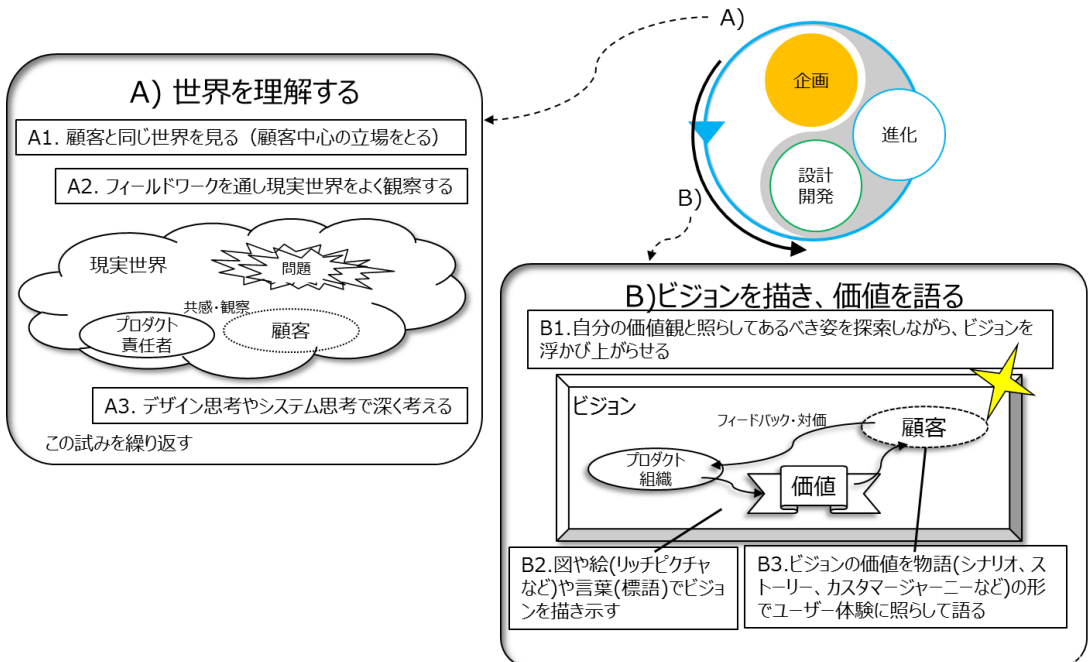


図10 『プロダクト企画とは』

3. プロダクト設計開発を導く ～ビジョンを実現するプロダクトを創る～

プロダクト設計開発を導くとは、語られたビジョンを、顧客に届けるための具体的なプロダクトとして実現化することです。

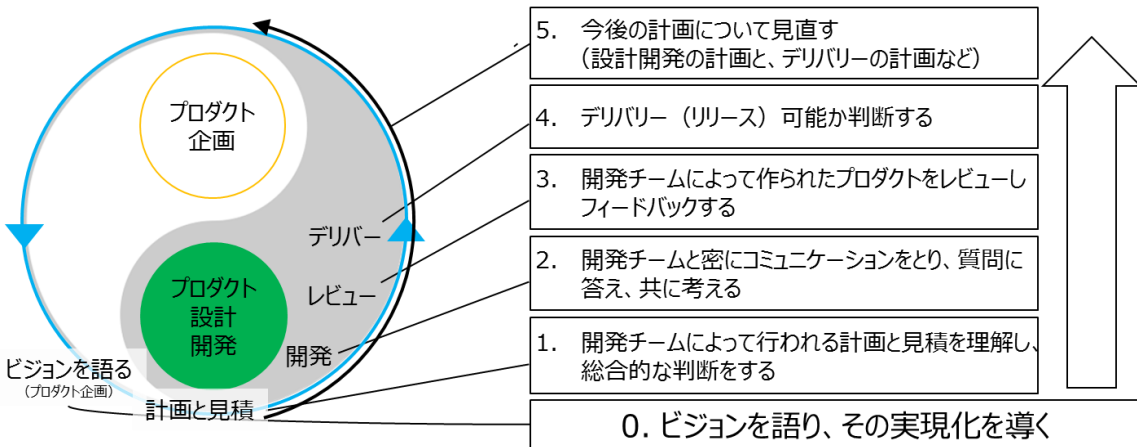


図11 『プロダクト設計開発を導くとは』

プロダクト責任者は、

- 具体的なプロダクトとして実現化するために、ビジョンを語り、関係するチームと対話することによって、プロダクト設計開発を導きます。
- プロダクト責任者が持つ判断基準（価値基準）や、必要な権限を、開発メンバーに委譲します。
- ビジョン、計画や見積、設計開発、デリバリーについてなど、「企画から開発・運用」まで「営業から技術」まで、多種多様なテーマの対話に主体的に関わります。

プロダクト設計開発チームの規模を一定とすると、実現化の順序は変更可能ですが、実現化のスピードは変更不可能であることが一般的です。

4. プロダクト進化 ～ビジョンとプロダクトをアップデートし続ける～

プロダクト進化とは、フィードバックされた情報と、ビジョンから、新しいプロダクト企画を通じて、プロダクトを進化させることです。

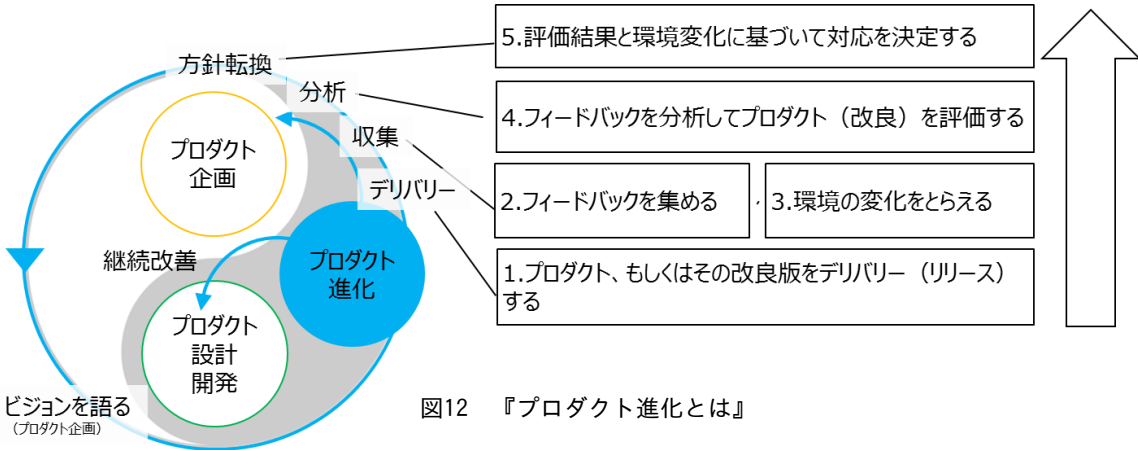


図12 『プロダクト進化とは』

1. プロダクト、もしくはその改良版をデリバリー（リリース）する
 - 市場において既に使われているプロダクトの改良をデリバリーする場合は既存の顧客へ苦痛を与えないようにします。
 - 初期の顧客からのフィードバックを得ることを目的とした実験段階のリリースにおいては機能をそぎ落とし最小限におさえることが重要です。
2. フィードバックを集める
 - 顧客の声とともに顧客の行動を収集します。
 - 収集しやすい指標ではなく、プロダクトあるいはその一部の価値を表すか、次のアクションにつなげられる指標を集めます。
3. 環境の変化をとらえる
 - これまでのビジネスやマーケットをかき乱す新しいビジネスが生まれていないかを確認します。
 - 新しく活用可能な技術が生まれていないかを確認します。
4. フィードバックを分析してプロダクト（改良）を評価する
 - 企画において設定したマーケットあるいは顧客が存在していたかを確認します。
 - プロダクト（あるいはその改良）は顧客に対して期待した価値を生み出したかを確認します。
 - プロダクトの利用を阻害する問題が生じていないか？を確認します。
5. 評価結果と環境変化に基づいて対応を決定する
 - 「同じ方向に向かって順当に進める」「方向を定めるためにさらなる実験を行う」「大きな方針転換をする」等の選択を行います（「設計開発」に戻るか、「企画」に戻ります）。
 - プロダクトに組み込まれた機能が全く価値を発揮していない場合、その機能はプロダクトの発展を妨げる重荷となるため、ときには大胆に作ったものを捨てる勇気を持ちます。

IV. アジャイル開発の進め方

本章におけるアジャイル開発の範囲と体制について

アジャイル開発の進め方には厳格な決まりごとや規範はありません。本章で説明（例示）する進め方、メンバーの役割（ロール）など、実際のソフトウェア開発プロジェクトでそのまま適用するものではありません。実際のプロジェクトや組織に適したやり方を取捨選択し、カスタマイズすることが必要となります。

「唯一正しい」アジャイル開発というものはありません。自分のいる組織に合ったやり方が、その組織のビジネスや活動、文化から自然と育っていくのがアジャイル開発の本質です。基本的なことを書籍や外部の人を通じて学んだ後、組織内で自律的に推進できるようにすることが必要です。

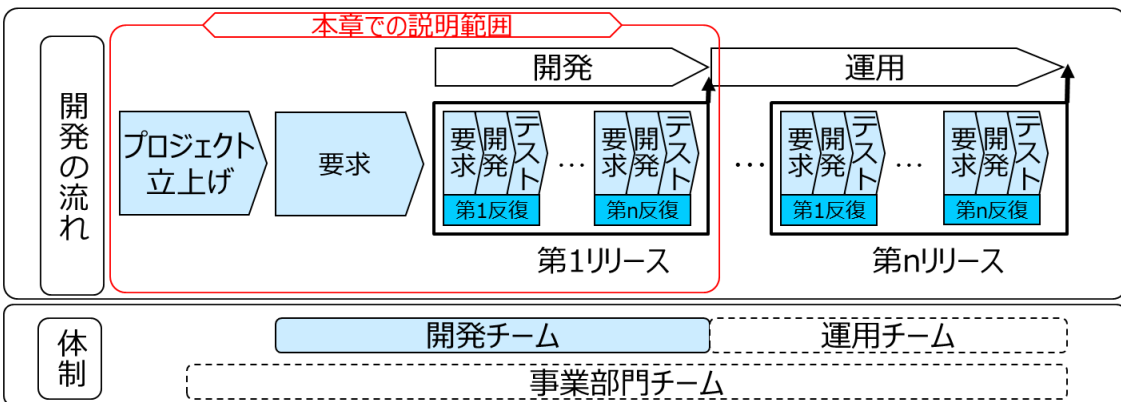
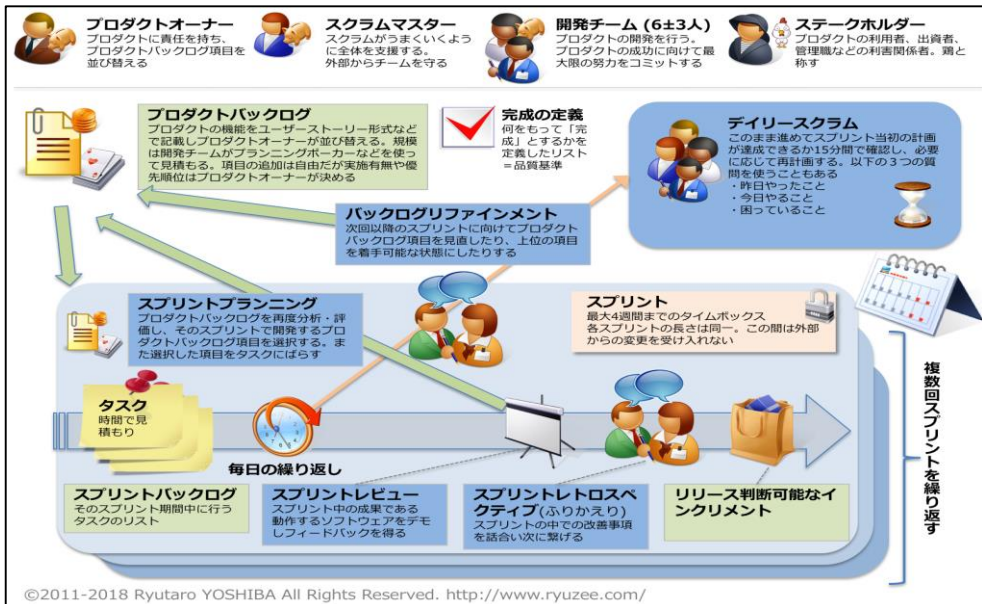


図13 『アジャイル開発の範囲と体制』

ソフトウェア開発の進め方には、規模や開発方針の違いにより、いろいろなバリエーションがあります。本章では、アジャイル開発の基本的な進め方を理解するため、開発モデルとしてシンプルなものを中心に考え、上図における「開発の流れ」中における「プロジェクト立上げ」～「開発」の範囲を説明対象としています。

プロジェクト立上げ時に開発チームを編成し、ユーザー側の事業部門内のチームと連携を図りながら、開発を進めていきます。比較的大規模な開発では、機能を設計開発するチーム（複数）と基盤・共通部を設計開発する共通的なチームを設置する場合があります。

1. アジャイル開発のプロセス（スクラムの例）



出典:<http://www.ryuzee.com> (許可を得て転載)

スクラムは反復（イテレーション）を繰り返す開発プロセスです。この反復の単位を「スプリント」と呼びます。スプリントの中身は、「スプリントプランニング」「デイリースクラム」「スプリントレビュー」「スプリントレトロスペクティブ（ふりかえり）」、そして実際の「開発作業」です。

「スプリント」は1~4週間の時間枠（タイムボックス）であり、予定されている機能が完成できなくても延長されることはありません。この期間内で開発チームはスプリントバックログの開発に集中し、リリース判断可能なインクリメント（プロダクト）を作り出します。

「スプリントバックログ」は、プロダクトバックログから抜き出された、今回のスプリントで追加する機能のリストを言います。スプリントプランニングでプロダクトオーナーの決めた順位と開発チームが決めた見積りの両方の情報を合わせて抜き出されます。このリストは一回のスプリントにだけ使用されます。

「リリース判断可能なインクリメント」とは、一回のスプリントにおける成果を指します。スプリント終了時にはプロダクトが動く状態であることが必要とな

り、これをレビューして、プロダクトオーナーが実際にリリースするかどうかを決定します。すなわち、スプリント終了時には「リリース判断可能」になっている必要があります。

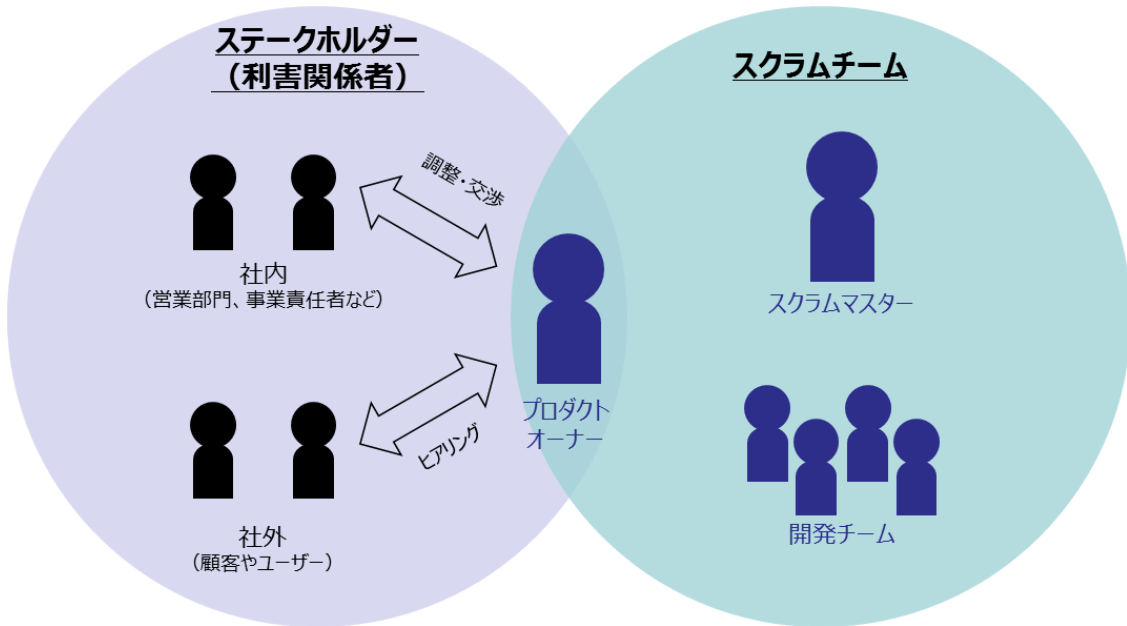


図14 『スクラムの役割の特徴』

スクラムで決められている役割（ロール）は、「プロダクトオーナー」「開発チーム」「スクラムマスター」の3種類です。これら全体を「スクラムチーム」と呼び、3つの役割が協働することで、大きな効果を創出します。

開発チームは、プロダクトの開発プロセス全体に責任を負い、開発プロセスを通して完全に自律的である必要があります。スクラムではこの自律したチームのことを「自己組織化された」チームと呼びます。チームがプロダクトを開発するために必要なスキルを全て備えている必要があります。従来型では、特定の専門性をもったメンバーが役割分担して開発することが一般的でしたが、スクラムの開発チームは、一人が複数のタスクを担う多能工である必要があります。

2. アジャイル開発チームのつくり方

アジャイル開発におけるチームの特徴は機能横断（クロスファンクション）型のチーム体制です。チームがプロダクトのライフサイクル（設計、ビルド、テスト、デプロイ、実行）を通じて完全に自律的であるためには、チームとしてバランスのとれたスキルセットを備えることが必要です。チームメンバーは仕事をこなすための深い開発スキルを持つと同時に、チームとしてパフォーマンスを最大化するためのスキルが必要です。

アジャイル開発に必要な全てのスキル・知識を持つ人材を育てることは必須の条件ではありません。一人の個人だけでは、スキル領域ごとのレベルに凸凹がありますが、その凸凹をチームとして埋めていきます。一人の個人だけでは不足している知識・スキルを、チームとして補っていくのです。チーム員同士で教えあい、チーム一丸となってプロダクトを開発していきます。チームで仕事をするにより、個人が得意とする分野だけでなく、より広範な分野の業務を実施できるようになり、個人の成長につなげることができます。このため、専門領域以外のスキルを埋めるために、チーム内の他の人材や他ステークホルダーと連携する能力も備わっていることが重要となります。

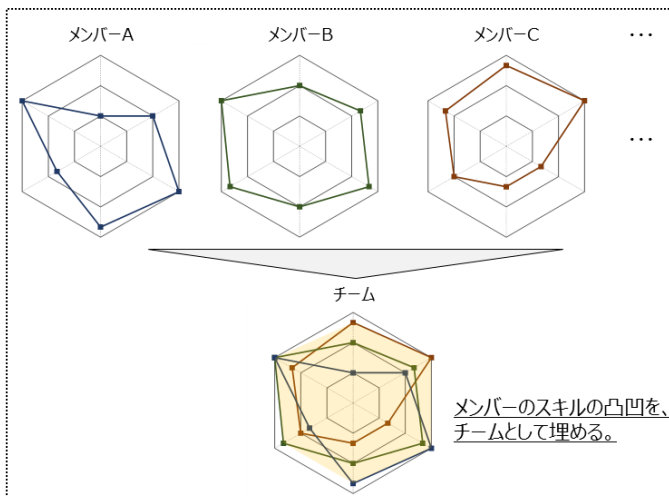


図15 『アジャイル開発チームのもつべきスキル』

V. まとめ

- 原則 01 顧客の満足を求め続ける
- 原則 02 要求の本質を見抜き、変更を前向きに
- 原則 03 成果物を2-3週間で、リリースし続ける
- 原則 04 全員で共通の目標に向かおう
- 原則 05 人の意欲は信頼から
- 原則 06 コミュニケーションは直接対話で
- 原則 07 進捗も品質も現物で
- 原則 08 一定のペースでプロジェクトにリズムを
- 原則 09 よい技術、よい設計、よい品質の追求
- 原則 10 ムダ= 価値を生まない、を探してヤメる
- 原則 11 よいモノはよいチームから
- 原則 12 自分たちのやり方を毎週、調整する

図16 『アジャイル宣言の背後にある原則 ～一覧～』

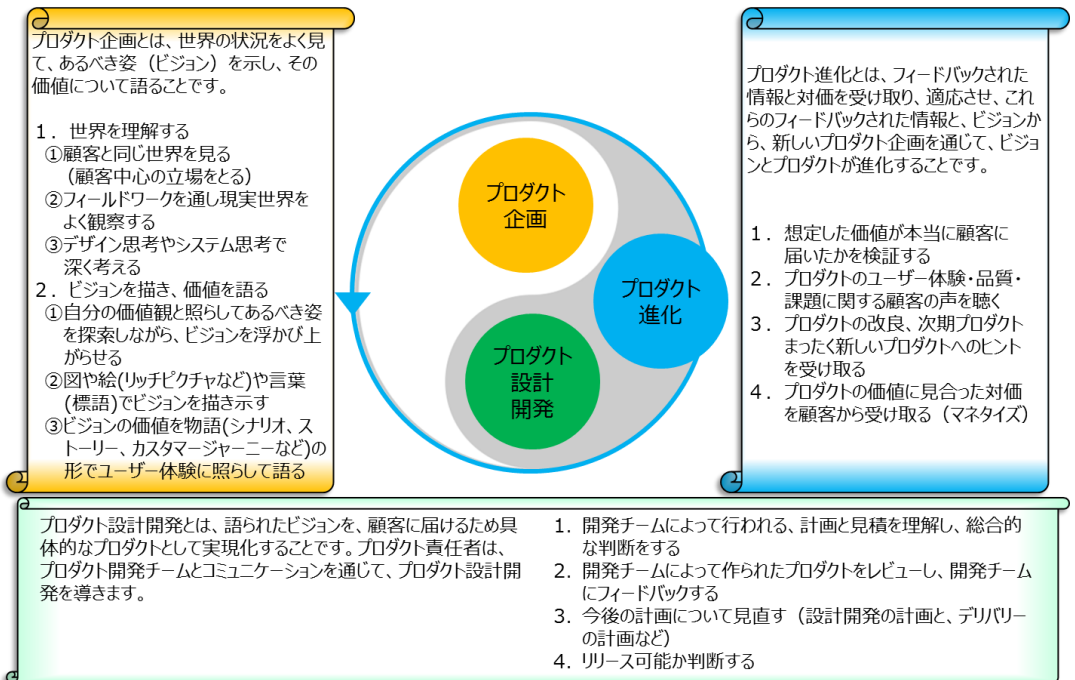


図17 『プロダクト責任者のすること』

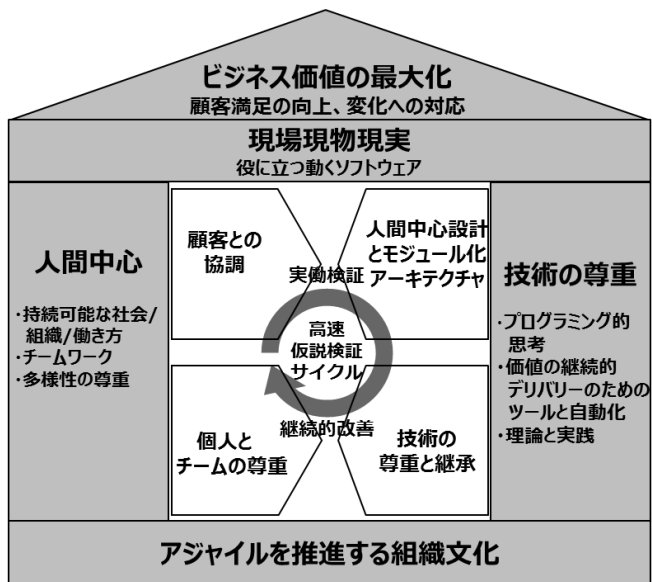


図18 『アジャイル開発の全体像』

アジャイル開発全体の概念構造を「アジャイル開発の家」として表現しました。家
をモチーフに、アジャイル開発の目指すもの（屋根、梁）、開発活動を支える大原則
（柱、土台）、そして目的を達成するための活動（家の中）を表しています。

活動の目的（屋根／梁）：

- ・ビジネス価値の最大化を実現するため、顧客満足の向上（何に価値があるか
を見極めること）、変化への対応（素早く提供しつづけること）を意識する
- ・現場現物現実で、実際に役に立つ、動くソフトウェアを提供し、顧客からの
フィードバックにより継続的に改善する

活動を支える原則（柱／土台）：

- ・人間中心：持続可能な社会/組織/働き方、チームワーク、多様性の尊重
- ・技術の尊重：プログラミング思考、価値の継続的デリバリーのためのツールと
自動化、理論と実践
- ・アジャイルを推進する組織文化

活動(家の中)：ビジネス価値の最大化を実現するための活動

- ・高速仮説検証サイクル（実働検証と継続的改善）
- ・顧客との協調、個人とチームの尊重
- ・人間中心設計とモジュール化アーキテクチャ、技術の尊重と継承

[付録]プロダクト責任者：太郎の物語

ここでは、プロダクト責任者のふるまいを具体的に浮き彫りにする物語を提示しています。ぜひ、お茶とお菓子をご用意の上、お楽しみください。

【背景】

老舗の和菓子店チェーン千石屋の新社長の駒込一郎が亡くなった。急遽、中堅自動車メーカーでエンジニアをやっていた息子の駒込太郎が呼び戻され、若社長としてリーダーシップをとることになった。千石屋は関東を中心に20店舗を展開し、20人の和菓子職人と営業・総務含め50人の従業員を中心にパートと併せて100人のメンバーを抱えている。

太郎はまず自分がこの業界では素人であるためメンバーの皆に助けてほしい、みんなの積極的なアイデアを全面的に支援するのでこの苦境を乗り越えていこうとのメッセージを伝えた。シニアな職人たちは前社長の伝統的な経営手法の下でぬるま湯につかっており、太郎を素人扱いしたが、若手の従業員は新しいチャレンジの可能性を感じているものも多かった。数週間かけて全員と面談し、千石屋の価値のありかを探るとともに、シニアからも若手からも会社の改善・改革に関わるアイデアをもらった。この面談は定期的に継続することとした。

太郎は現状の千石屋の業務プロセスにムダがあること、いまのやり方では規模拡大はむずかしいことを若手の営業マンから聞き出し、機械による自動化の導入を検討することにした。若手メンバーは美味しい和菓子を21世紀の現在、広く家庭に届けたいと考えていた。一方で、古くからいる職人たちからは機械導入に対する反対が表明された。機械化による安価な大量生産品というイメージが職人にカチンときたのである。彼らの想いは本格職人による手作りの和の伝統菓子のプライドである。

太郎は職人全員と営業マン、各店舗責任者を集めて、今後自分がプロダクト責任者となって千石屋のブランドを棄損しないように社会に誇れる「ちょっと高いけれど美味しく、健康的で笑顔になる21世紀の和菓子」というビジョンを掲げると宣言した。

【プロダクト企画】

面談を通して、千石屋の和菓子の特徴は加賀七義園由来の「本格派」和菓子、餡や皮に使われる小豆・大豆・米・小麦などの材料も厳選され添加物一切なしの「自然派」「健康的」という評価がされていることがわかった。

太郎はこのコンセプトをSociety5.0時代の社会に受け入れられる具体的なビジョンに落とし込みたかった。しかし何かが足りない。模索する中でコトノハという和菓子が米麴を使った発酵食品であり乳酸菌も含まれていることに気づいた。まさに今話題になっている健康食品として取りあげられている菌だった。そこでコトノハに使う麴菌と乳酸菌を他の和菓子にも添加する可能性に挑戦することにした。

シニア職人のエキスパート衆2名に若手2名でチームを作り、数か月に渡り試行錯誤（PoC）を重ね、品の良い味わい深いコトノハ2.0βが制作できた。営業が何百人もの試食者を開拓し、これは美味しいと言って頂ける味になった。

【プロダクト設計開発】

コトノハ2.0は北海道の小豆と新潟の米、東京の麴菌IPA1号を適切なタイミングと分量でしっかりと練り上げる必要があった。これを大量に安全に製作する方法をデザインする必要がある。1週間毎のタイムボックスで製造量と品質目標を設定し、最初の1か月ではじっくり基本的なステップでの製造工程を確立し、その後、スピードと量の問題を少しずつクリアしながら、品質問題をつぶし、半年後には安定した製造ラインの確立に成功した。



【プロダクト進化】

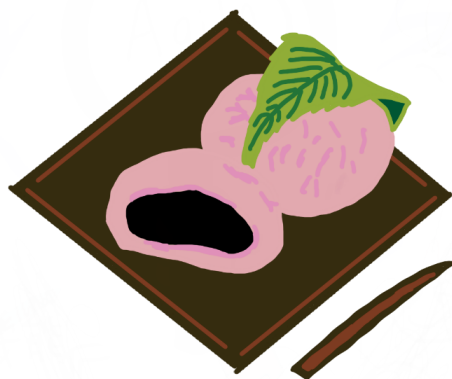
まずは東京麻布の旗艦店のみでの販売とし、顧客のフィードバックを受けた。味は良いがデザインが古いのではとの指摘であった。季節ごとに若手デザイナーにパッケージ案を競わせ、SNS上でコンペを実施し、商品開発に活かすことにした。そのようにして改善された商品は、季節ごとに人気を博し、コトノハの人気は、ますます高まった。

旗艦店での販売が好調のため、人気に応えるには材料の確保ルートをさらに拡大する必要があった。材料確保に悩む中、若手営業の1人が持ち込んだタイ米を試しに使ったコトノハタイ0.1を作ってみると思いのほか麴菌との相性がよく、これはコトノハ2.0を超える味の可能性を秘めていることがわかった。これを次のプロダクト企画に上げ、PoCの実施を決定した。

また、たまたま麻布店に立ち寄ったパンメーカー社長がこれをうちのパンにも入れてみたいと声を掛けてきた。そこでコトノハパンの企画コラボレーションが始まった。

【後日談】

3年たった現在、千石屋の売上の30%は和菓子コトノハ系列、10%がその他の和菓子、そして驚くべきことに残り60%はコトノハ酵母菌の他食品メーカーや化粧品メーカーへの卸販売という構成に大きくシフトしているのであった。。。



この物語は、フィクションです。登場する人物・団体・名称等は、架空であり、実在するものとは関係ありません。

[付録]ITSS+（プラス）の背景

第4次産業革命に向け、クラウド、データアナリティクス、モバイル、ソーシャルといったいわゆる第3のプラットフォームや、IoT、AI活用の本格化を背景にIT投資の在り方も変化が求められています。IT投資は、ビジネスのバックエンドを中心とした伝統的な領域から、新たに、売上や利益拡大に直結するフロントエンドの領域へと本格的に拡大しています。

こうした変化に応じIT人材のスキルも強化・変革が求められます。伝統的なIT投資と新たなIT投資では特徴が大きく異なる面があります。特に伝統的なIT投資を主に担ってきた人材にとっては、新たなIT投資に対応するスキルの強化・変革へ向けた“学び直し”が重要となり、積極的な学び直しを通じ“安定性・信頼性を確保しつつ、スピードや柔軟性を追求・実現する IT投資の最適解を担う人材”となることが期待されます。

表3 IT投資の拡がりの特徴

	新たなIT投資	
	伝統的なIT投資	
目的	守りのIT投資 (コスト削減) (ビジネスを支援)	攻めのIT投資 (売上・付加価値向上) (ビジネスを実行)
傾向	安定性重視	スピード重視
対象領域	バックエンド SoR (Systems of Record)	フロントエンド SoE (Systems of Engagement)
IT投資の形態	プロジェクト	プロダクト・サービス (価値提供)
オーナー	情報システム部門	LOB (事業部門: Line of Business)
開発手法	ウォーターフォール	アジャイル、DevOps等
プラットフォームへの要求	信頼性・堅牢性	拡張性・柔軟性
開発形態	ITベンダーへの外注が主体	ユーザー企業での内製や パートナーングによる開発が主体
人材の役割	分業・専門分化	フルスタック・マルチロール
開発運用体制	技術者とIS部門	技術者とIS部門+事業部門
対象業務	予測可能	探索型
データ	構造	構造+非構造+外部
強み	統率力・実行力	機動力・柔軟性

アジャイルWGメンバー一覽

表4 2017年度アジャイルWGメンバー一覽

	氏名(敬称略)	所属
主査	今村 博明	インフォテック株式会社
メンバー	伊藤 裕子	株式会社東芝
	遠藤 猛	パナソニックシステムソリューションズジャパン株式会社
	鈴木 依子	株式会社ITプレナーズジャパン・アジアパシフィック
	水野 浩三	日本電気株式会社
	和田 憲明	富士通株式会社
	渡会 健	株式会社アドヴァンスト・ソフト・エンジニアリング
オブザーバー	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
	山下 博之	独立行政法人情報処理推進機構
アドバイザー	羽生田 栄一	株式会社豆蔵
	平鍋 健児	株式会社永和システムマネジメント

所属は2018年3月末時点

表5 2018年度アジャイルWGメンバー一覽

	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
メンバー	今村 博明	インフォテック株式会社
	齋藤 修一	日本アイ・ピー・エム株式会社
	鈴木 依子	株式会社ITプレナーズジャパン・アジアパシフィック
	関 満徳	グロース・アーキテクチャ&チームズ株式会社
	和田 憲明	富士通株式会社
	渡会 健	株式会社豆蔵
オブザーバー	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
	山下 博之	独立行政法人情報処理推進機構
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
文書化支援	本橋 正成	合同会社カルチャーワークス

所属は2019年3月末時点

アジャイルWGメンバー一覧

表6 2019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	今村 博明	SOMPOテクノロジー株式会社
	川上 誠司	アクセンチュア株式会社
	齋藤 修一	日本アイ・ピー・エム株式会社
	関 満徳	グロース・アーキテクチャ&チームズ株式会社
	和田 憲明	富士通株式会社
	渡会 健	株式会社豆蔵
オブザーバー	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
	山下 博之	独立行政法人情報処理推進機構 (IPA)
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

所属は2019年12月末時点

MEMO

A series of horizontal dashed lines for writing.





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
成田 富男 独立行政法人 情報処理推進機構



イラスト（敬称略）

事務局：成田 富男 独立行政法人 情報処理推進機構

ITSS+ アジャイル領域 価値創造社会の持続的発展のカギはアジャイルにあり

2019年7月 第1刷発行
2020年1月 第2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
©独立行政法人 情報処理推進機構

アクセス
はこちら



本冊子をお読みいただき、ありがとうございます。
Webアンケートへのご協力をよろしくお願いいたします。

