

変化を味方につけるアジャイル開発

株式会社永和システムマネジメント
株式会社チェンジビジョン

平鍋 健児

「つながる時代」が到来し、新たな知識を獲得しながらシステム開発を進めるアジャイル開発の有効性が注目されている。要求の事前固定が難しい開発、新技術を含みトライ＆エラーを必要とする開発、使いながらユーザー要求を発掘していく開発などに適した手法としての認知を得てきた。欧米手法の輸入と考えられがちであるが、リーンの源流であるTPS（トヨタ生産方式）、スクラムのオリジナルコンセプトである80年代の日本の新製品開発手法など、アジャイル開発には日本発の考え方も多く入っている。本稿ではアジャイル開発の概要や求められる背景、スクラムを例とした開発手法などを紹介する。

1 アジャイル開発とは

アジャイル開発は1回の開発期間（イテレーションやスプリントと呼ぶ）を短く設定した、繰り返し型の開発手法で、優先順位に基づいて動くソフトウェアを徐々に成長させていくという特徴がある。アジャイル開発それ自身は具体的な開発方式ではなく、スクラムやXPといった手法を傘下に置く大きな括り言葉であり、「アジャイル宣言」^{※1}をよりどころにしている。

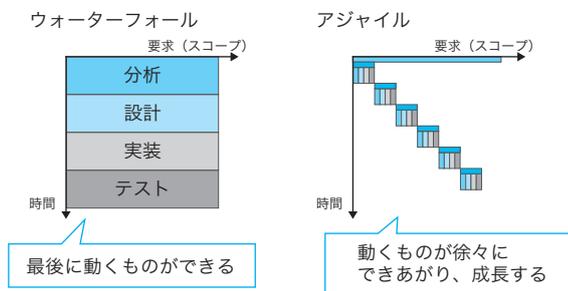


図1 繰り返し型プロセスとしての Agile

単純化して比較すると、ウォーターフォールでは最初に必要なすべての要求を定義し、設計、実装、テストと順に開発する。そのため、動くものが確認できるのは開発の最後である。これに対しアジャイル開発では短いサイクルでテストまで終了した利用可能な状態のシステムを、徐々に積みあげていく。伝統的なV字プロセスに対応させると、「要求を一掴みとり、高速にV字を移動させ、受け入れテストまで通過させてしまう」という動きを繰り返すことになる。

アジャイル開発の特徴はその開発サイクルが短いというだけでなく最大の利点として、ビジネス環境の変化に合わせて変化するシステムへの要求に対し、開発する製品やサービスに柔軟に素早く対応することが可能な点にある。また、短期開発から得られた経験に基づき、次の計画を見直ししていくことを繰り返すため、製品もチームも改善が継続的に実施されるのがアジャイル開発の特徴である。

2 アジャイル開発が求められる背景

アジャイル開発の有効性が注目されている背景にはビジネスからの要請がある。これまでシステム開発は、ゴールを想定し要求をまとめて仕様書と共に発注し、請け負った側は仕様書通りに作る。開発時点でシステムやサービスの仕様が明確であり、未知のものが存在しないのであれば問題はない。しかし、ビジネススピードが速くなり、グローバルに競合企業とサービスを競うような状況にあって、開発期間内に要求が変わらないことを保障するほうが難しい状況にある。そのため、ビジネススピードに合わせてソフトウェアを提供する、また、変化に素早く対応してソフトウェアを提供することが求められる。その一つの解決策となっているのがアジャイル開発である。ユーザーと連携しながら要求変更を取り入れながら開発を進め、ユーザーのニーズを理解した上でユーザーの想いに近いソフトウェアを実現するアジャイル開発の特徴が昨今のビジネスからの要請と親和性が高いのである。

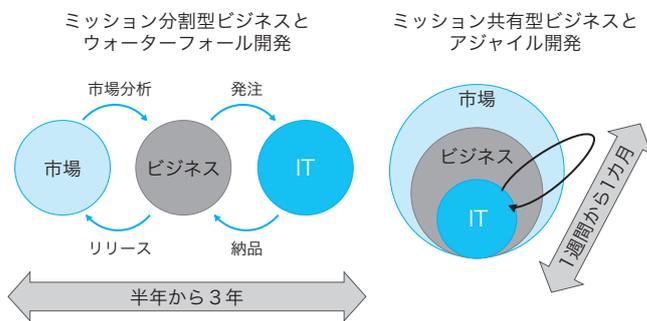


図2 ビジネス構造としての Agile

そのためには、ビジネスの構造にも変革が求められる。市場調査後に要求をまとめ、それを発注し、納品されたも

【脚注】

※1 アジャイル宣言 : <http://agilemanifesto.org>

のを市場提供する、というのが従来型のビジネス構造である。ここではシステムの要求を定義すること（発注側）、仕様書通りに作ること（受注側）、にミッションが分断され、アジャイル開発は機能しない。これは、ユーザ側とベンダ側に分かれた日本のシステム開発でのアジャイル採用の難しさを示している。ミッションを共にした一体感を持ったチームづくりが難しい。自社サービスを保有し自社にエンジニアを雇用している Web サービス事業者でアジャイルが浸透しやすい理由もここにある。自社サービスの成功ために、ビジネスとエンジニアが一体となった開発チームを作りやすい^{※2}。

3 スクラムとは

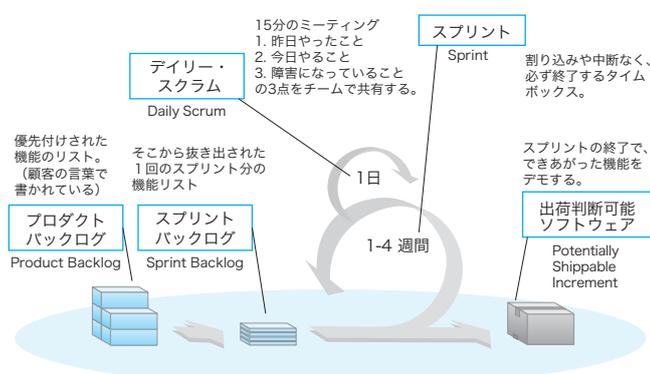


図3 スクラムのフレームワーク

アジャイル開発の代表的な方法論であるスクラムに基づいて開発の進め方を紹介する。

スクラムでは、優先づけされた要求のリストを「プロダクトバックログ」と呼び、1週間や1カ月といった比較的短期間（スプリントと呼ぶ）で開発のループをまわす。プロダクトバックログから1スプリント分を先頭から抜き取って、これを「スプリントバックログ」と呼ぶ。1スプリント分のアウトプットが「出荷判断可能ソフトウェア」、すなわちテストまで済んだ動くソフトウェアである。スプリントをまわすごとにどんどん成長していく。短期間での開発なので、毎日朝会（「デイリー・スクラム」）を実施し情報共有すると共にチームで話しながら日々の問題点を解決していく。

スクラムチームの構成は、(1) プロダクトバックログとその優先順位に責任を持つ「プロダクトオーナー」、(2) 開発者、テスター、Web デザイナー、品質保証といった開発に関係する「開発チーム」、そして、(3) 全体のプロセスをうまく動かしていくための支援をする「スクラムマスター」、これら3つの役割から成る。言い換えるとスクラムマスターが従来のプロジェクトマネージャを置き換え、プロダクトオーナーがビジネス側の仕様決定者、開発チームが開発者となる。実際にはチームは自己組織化されており、例えばスクラムマスターはタスクを個々の開発者にアサインしたりしない。チームが自分たちで決めて実行する。更にペアを

組んで開発をする、タスクボードを使って日々の開発を透明化するなど、様々なプラクティスがそこには含まれている。

スクラムマスターは開発全体を俯瞰し開発を成功に導く。プロダクトオーナーは、開発のROIを最大にすべく、単なる要求の優先順位づけだけでなく、製品やサービスがどういふ姿になりたいのか、その最終目的は何なのかというビジョンを語り、チームとコミュニケーションして要求とユーザに関する知識をチームにインプットする大切な役割を持つ。

スクラムではプロジェクトの進行と共に、フィードバックを受けながらプロダクトバックログに入っている機能の詳細化、見積もり、追加や削除、優先順位の見直しが順次進められる。並び順が上のアイテムほど明確で詳細化された状態になっている。スプリントを開始する直前にスプリント計画会議を開催し、今回のスプリントで開発する内容を確定させ、プロダクトオーナーと開発チームで合意してスプリントが始まる。

スプリント期間中は、15分のデイリースクラム（朝会）というミーティングを毎日開催し、情報共有の場を持ち、メンバ全員が互いの状況を共有する。フェイス・トゥ・フェイスで全員参加のミーティングには、作業進捗だけでなく、その日の体調やチームを取り囲む様々な状況を把握できるというメリットがある。上司への報告ではなくチーム全体への報告というスタイルとなるのが特徴である。この時、バックログを壁と付箋で表現して視覚的に短時間で最新状況を共有するためのタスクボードなどが用いられる。

スクラムでは、スプリント中にどのような作業プロセスで仕事を進めるかは定義されておらず作業のやり方は開発チーム自身で決める（後述の「プラクティス」参照）。

スプリント完了時にスプリントレビューを開催し、開発した結果についてステークホルダにデモを行い、フィードバックを得る。フィードバックされた結果はプロダクトオーナーを中心に検討され、必要があればプロダクトバックログへ反映する。また、このスプリントレビューとは別に、次のスプリントをよりうまく、楽しく進めるためにふりかえり（スプリント・レトロスペクティブ）を行う。開発したソフトウェアに関してではなく、スプリントで得られた知識を集約して開発プロセスやプラクティスを改善する。実際に開発作業をしている現場の人が、手を動かして気付いたことを自分の活動の中に取り入れていく、プロセス改善がプロセスの中に含まれており、自分たちのプロセスを作っていくことがアジャイル開発の大きなポイントである。

4 アジャイル開発のプラクティス

このように、スクラムには「プロセスの枠組み」のみが規定されており、実際に現場で行われるソフトウェア開発

【脚注】

※2 IPA 調査参照 <http://www.ipa.go.jp/sec/softwareengineering/reports/20120611.html>

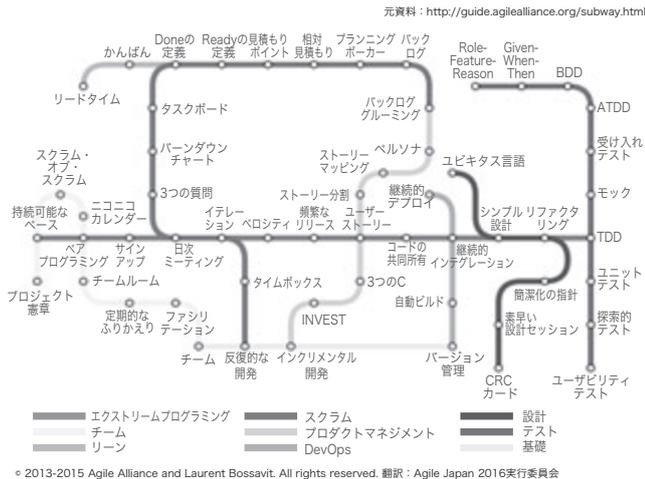


図4 アジャイルプラクティスの地図

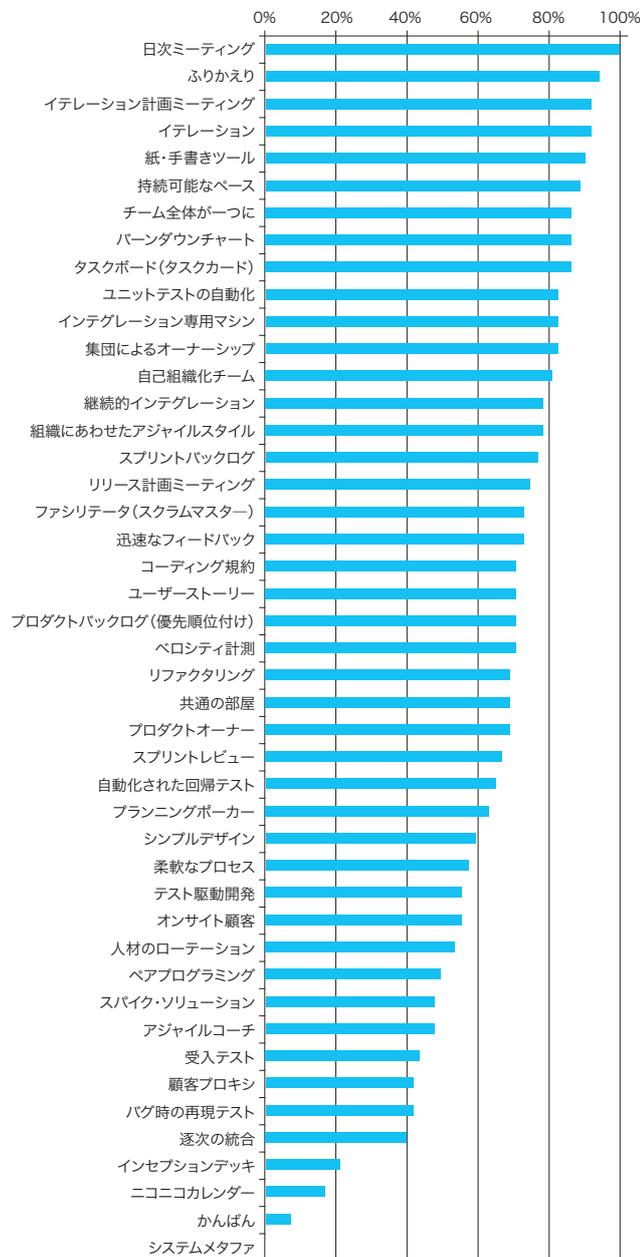


図5 日本でのアジャイルプラクティスの採用状況^{※4}

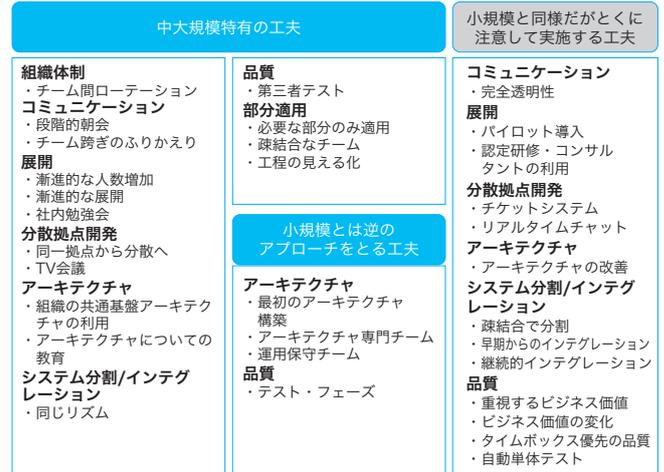


図6 中大規模事例での工夫^{※5}

の設計手法、開発環境、見積手法、会議の進め方、テスト技法、などは定義されていない。「アジャイルプラクティス」と呼ばれるこれらの個々の活動は、これまでに方法論ごとに多数提案されており、それに加えて個々の現場で生み出されたローカルなものも多い。実際のアジャイル開発では、開発チームごとにプラクティスを選択、あるいは、編み出して、自分たちで自分たちのプロセスを作り、維持・改善していく（そのエンジンとなるのが、スプリントの最後に行う自己改善活動「ふりかえり」なのである）。

図4は、Agile Allianceが掲載している「プラクティス地図」である^{※3}。様々な手法からプラクティスが相互乗り入れしている様子が、地下鉄のマップのように示されている。また、IPAによる日本でのプラクティス活用事例^{※4}（図5）には、日本でのプラクティス採用データが掲載されている。スクラム（やXP）の基本的な枠組みに、様々なプラクティスが組み合わされていることが分かる。

また、図6にはチームや組織ごとに工夫したプラクティスが抽出されている。中大規模では難しいと言われているアジャイル開発であるが、こういった現場ごとの工夫があったアジャイル開発が実際に機能するようになる^{※5}。

5 アジャイルの歴史

アジャイルという言葉は、2001年にユタ州のスノーバードで前述のアジャイル宣言が書かれたのが最初だ。その前には、90年代後半に出てきた手法であるXP（Extreme Programming）、スクラム、更に前にはEvoという、PDCA

【脚注】

- ※3 <http://guide.agilealliance.org/> (Agile Japan 2016 実行委員会翻訳)
- ※4 IPA「アジャイル型開発におけるプラクティス活用事例調査」の報告書とリファレンスガイドを公開 <https://www.ipa.go.jp/sec/softwareengineering/reports/20130319.html>
- ※5 IPA中・大規模な開発プロジェクトで非ウォーターフォール型開発を成功させるポイントなどを紹介 <http://www.ipa.go.jp/about/press/20120328.html>

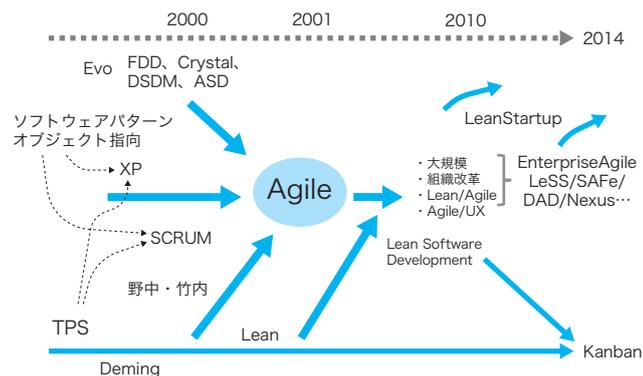


図7 アジャイルの現在位置

を基礎にした繰り返し手法が存在した。これがアジャイルのメソッドとして提唱された最古だろう。後に FDD (Feature Driven Development)、Crystal Clear、英国コンソーシアムの DSDM などがあった。ASD (Adaptive Software Development) は後に『アジャイルプロジェクトマネジメント』を書いた Jim Highsmith の考え方である。このようなメンバがスノーボードに集まって、「アジャイル」(Agile) という言葉を作った。

リーンソフトウェア開発は、Mary と Tom Poppendieck が提唱したが、この夫妻は TPS への造詣も深く、「リーン」と「アジャイル」を結び付けたことで、それまでソフトウェア開発活動とのみ捉えられていたアジャイルが、経営的な視点で説明された。Kanban、というアジャイル手法もその後提案されている。リーンとは他産業適用へ抽象化された TPS であり、著者なりに一言で言うと「顧客の目を見た価値を定義し、その価値を顧客からのプルで流れ化し、改善活動を現場参加で行う」となる。リーンの言葉で経営者に対してアジャイルを語れるようになったというのは大きな変化であり、アジャイルが爆発的に普及する契機になった^{※6}。大きなリーンの流れの1つにアジャイルも位置付けて理解できるようになったのだ。

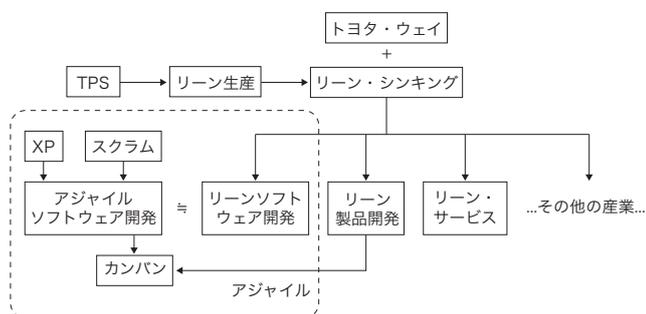


図8 リーンの流れからみたアジャイル^{※7}

また、スクラム、という名前は 1986 年に竹内弘高と野中郁次郎による Harvard Business Review の論文「The New New Product Development Game」に由来している。そこでは日本の新製品開発が、職能の垣根を取り払ったチームで行われていること、成果物をバトンパスするリレーでなく、

全員が一丸となってボールを運ぶラグビーのようにプレーしている、という意味で「スクラム」という用語が使われている^{※8}。「知識はどこからやってくるのか」という深淵な問いに答えようとする SECI モデルがスクラムの中核にあるのだ。

そして、2011 年には、「リーンスタートアップ」が書かれた。ここでは、米国西海岸のスタートアップ企業が顧客開発と製品開発のループを1つにして、資金を有効に(リーンに)活用しながらスタートアップする手法を記述した。これは、アジャイルの最適な利用法としてのひとつの理想型と言えるだろう。

6 アジャイルの課題とエンタープライズアジャイル

現在議論になっているのは、大規模化、分散チーム、オフショア開発、をどう扱うか、更に、企業の予算システムや人事評価との整合どうするのかといった、アジャイルと組織との関連である。それに伴い、「エンタープライズアジャイル」という言葉も生まれた。海外では、エンタープライズアジャイルフレームワーク、と呼ばれる幾つかの手法も提案されている。LeSS (Large-Scale Scrum)、SAFe (Scaled Agile Framework)、DAD (Disciplined Agile Delivery)、Nexus、の4つが現在よく知られている。どれも、アジャイルチームをスケールアップする試みだ。

7 最後に

この記事では、アジャイル開発の概要、ビジネス変化の要請、歴史的背景を概観した。とくに、アジャイルはプロセスとして定義されているものではなく、「プラクティス」をチームで選択、工夫しながら自分たちのやり方に仕立て上げる活動が重要であることを強調したい。2016年5月31日には、毎年恒例となっている「アジャイルジャパン2016」が開催される。今年のテーマは「あなたとつくるアジャイル」とした。日本でアジャイルが普及するには、現場での工夫とその事例の共有が必要であるとの思いである。

アジャイルは海外から普及が始まった手法ではあるが、トヨタ生産方式や野中郁次郎の80年代の新製品開発の研究など、コンセプト部分には日本からのアイデアがたくさん入っている。アジャイルは本来、日本が得意とするそのチーム開発のやり方なのだ。どこまで行っても、「ソフトウェアは人が人のために作っている」のであり、そのソーシャルな活動総体のデザインがアジャイルなのである。

【脚注】

- ※6 「リーン」と「アジャイル」の関係とは?
<http://www.atmarkit.co.jp/ait/articles/1311/15/news015.html>
- ※7 『リーン開発の現場』前書きより
- ※8 『アジャイル開発とスクラム』(野中郁次郎、平鍋健児)