

2018年6月20日

アジャイル型の取締役会是最強か？

経営コンサルティング第一部
コンサルタント 内山和紀

流行の変化が激しいソフトウェア開発の現場では、伝統的な開発手法以外にも、柔軟な対応が可能な「アジャイル型開発」という手法が一部で活用されている。今、臨機応変な対応が求められる取締役会において、アジャイル型開発手法は取締役会の更なる進化へ寄与するのだろうか。

1. はじめに

各方面から指摘されているとおり、日本企業の取り巻く環境は、グローバル化、技術革新の進展、少子高齢化、社会・環境問題への関心の高まりなど、大きくかつ急激に変化しており、企業が直面する経営上の課題も高度化・複雑化しつつある。

そのような中で企業が持続的に成長していくためには、取締役会を始めとした経営陣が的確な経営判断を行っていくことが重要である。そして、様々な経営課題への対応力、実効性について継続的に評価を行い、PDCA サイクルを実現し、取締役会の機能の向上を絶えず図っていくことが求められている。¹

一方、流行の変化が激しいソフトウェア開発の現場では、伝統的かつソリッドな開発手法である「ウォーターフォール型開発」から、臨機応変な対応を可能とする「アジャイル型開発」という手法が一部のプロジェクトにおいて活用されている。本稿ではアジャイル型開発手法とウォーターフォール型開発を比較・整理しつつ、アジャイル型アプローチが取締役会の更なる進化へ寄与する可能性について検討していきたい。

2. アジャイル型開発手法とは

これまでの一般的な開発スタイルであるウォーターフォール型は、要件定義、外部

¹ スチュワードシップ・コード及びコーポレートガバナンス・コードのフォローアップ会議「会社の持続的成長と中長期的な企業価値の向上に向けた取締役会のあり方」（経済産業省）にて意見書が提示されている。（<https://www.fsa.go.jp/singi/follow-up/siryoku/20160218/01.pdf>）

設計（概要設計）、内部設計（詳細設計）、開発、テスト、リリース・運用と各工程を分割して着実に実行していく。前工程が完了しない限り次の工程には進まないことで、各工程の品質を保証しながら進行していくため、後戻りがないことから進捗管理がしやすい開発スタイルである。一方で、ウォーターフォール型は後戻りをするを想定していないことから、プロジェクト開始後の仕様変更は最初から手直しの必要があるためコストが大幅に増加する懸念がある。

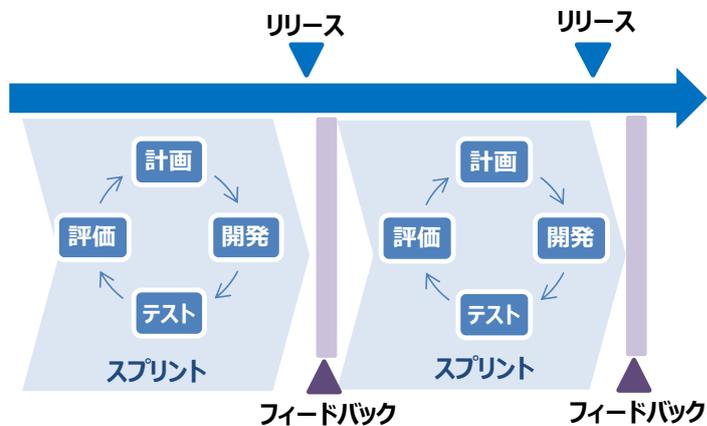
（図表 1）ウォーターフォール型開発



出所：大和総研作成

これに対し、「俊敏な」という意味をもつアジャイル型の場合は、比較的短期間に必要最低限のリソースを使って開発を行い、それを基に顧客とコミュニケーションを取り、顧客の要望を取り入れながら、機能のブラッシュアップを行っていく。これを複数回繰り返し、最終的な完成を目指す開発スタイルである。アジャイル型開発は綿密な計画よりも実践を重視するのでプロジェクトの途中からでも要件の見直しが可能であり、明確な最終形が決まっていない場合に採用されることが多い。

(図表 2) アジャイル型開発



出所：大和総研作成

3. アジャイル型開発の概要とメンバーの役割

アジャイル型開発の中でも代表格であるスクラムは、短期間での開発サイクルで成果物を作成し、この開発サイクルを繰り返す開発手法である。一つの開発サイクルは「スプリント」と呼ばれる1～2週間程度の期間であり、このスプリント内で「誰が何をするか（計画）」「活動（開発）」「進行上の課題はないか、顧客のニーズは満たしているか（テスト）」「問題点はなかったか、次回に向けた解決策は何か（評価）」といったPDCAを回すことが求められる。

アジャイル型のチームは、1～2週間程度でPDCAを回す必要があることから、密にコミュニケーションのとれる人数（最大でも10名）で構成される。チーム内のメンバーは開発チーム、プロダクトオーナー、スクラムマスターのいずれかの役割を持ち、その概要は以下の通りである。

【開発チーム】

開発時に顕在化する様々な課題を解決するために、開発チームはそれぞれ専門性をもつメンバーで構成されており、結果として開発チームは多様性をもつ。また、開発チーム内はフラットな形態をとるため、上司と部下の関係や、指揮・命令等のマネジメントは行われず、チームメンバーは課題解決のために、必要に応じてメンバー同士でコラボレーションを行う。メンバー間の密なコミュニケーションを重視し、全員で情報を共有してチームの方向性を決定する。

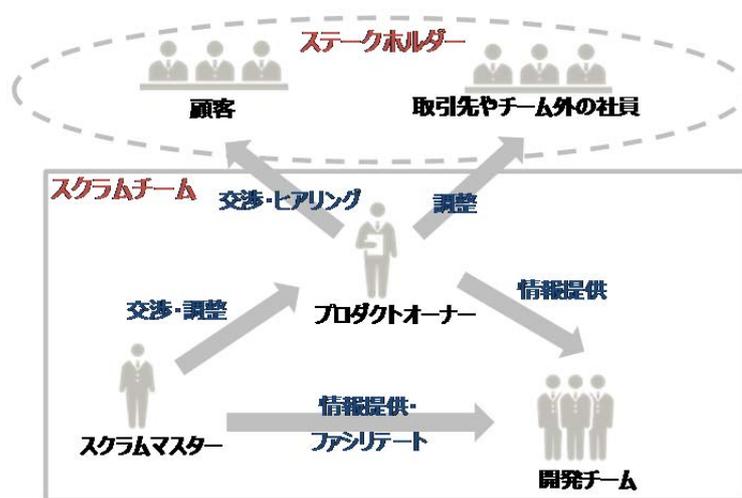
【プロダクトオーナー】

プロダクトオーナーは顧客の代理として務めるメンバーである。成果物（プロダクト）に関する意思決定の責任者となり、要件の決定や要件の優先度の変更・決定等を行う。なお、プロダクトオーナーはリーダーではないため、チームメンバーへの指示は行わずに彼らが必要としている情報を適切に提供することが役割となる。必要に応じて、顧客やチーム外の社員、取引先等のステークホルダーとの調整や交渉を行う。

【スクラムマスター】

スクラムマスターは開発チームが円滑にプロジェクトに邁進できるよう、障害となっている事象を解決するためのサポートを行う。例えば、開発の際に障害となる事項をリスト化して管理すること、そして開発チーム自身が解決できるようファシリテート等を行い、必要に応じて他のチームやコミュニティ、専門家の協力を仰ぐといったプロダクトオーナーとの相談や調整も行う。

(図表 3) アジャイル（スクラム）チームメンバーの役割



出所：大和総研作成

4. 実効的な取締役会の実現に向けたアジャイル型活用の示唆

アジャイル型チームは、多様性のあるメンバーが密なコミュニケーションを取りながら、チームのために自律的に行動する。その結果としてメンバーの迅速果断な行動

を促し、短期間における PDCA サイクルを回し、チームとして成果を生むことを実現している。このようなアジャイル型のチームは、今求められている取締役会のあり方へのヒントになるのではないだろうか。

取締役は開発メンバーのように各自の専門性を発揮し、多様性のあるメンバー間でコミュニケーションをとりながら共通の認識を醸成し、自社にとって最適な対応を選択する。また、チームで作上げたプロダクトをステークホルダーへ提供するプロダクトオーナーは、従業員や投資家に対し会社の目指すところやビジョン等の説明をする CEO の役割に近いかもしれない。そして、開発チーム内の議論を取りまとめるスクラムマスターは取締役会議長と置き換えることができるだろう。

以前の取締役会では形式的な付議事項が多く、より本質的な重要度が高い議題については、議論が不足していると考えている企業が一定数みられた。²2015 年 6 月に施行されたコーポレートガバナンス・コードについては、本年に改訂がなされ、取締役会における CEO の後継者計画や CEO 選解任のプロセスの策定等、より実質的な対応について言及されている。取締役会をアジャイル型チームとして捉えなおすことで、より機動的で柔軟性の高い運営が実現されるのではないか。

5. アジャイル型の実行役会を運営するうえでのポイント

【小さく始めて大きくする】

アジャイル型の開発では、まず主要な機能をコンパクトに作成し、早い段階で効果の測定を行う。その機能の仕様が誤っていればすぐに軌道修正を行い、正しいと判断すれば方向性は変えずに機能を拡大していく。

例えば、取締役会で中期経営計画について議論する際には、具体的なアクションプランまで落とし込んでから取締役会に諮るのでは時間がかかってしまうし、そこまで作りこんでから方向性が異なっていた場合は多くの手戻りが発生してしまう。会社の進むべき方向性が決まり次第、まずはその核の部分について十分に議論することが重要である。

【他のコミュニティと連携する】

開発チームは各自異なる専門性を持つメンバーで構成される。そのため、とある専門分野において自分より詳しいメンバーが不在の状態もしばしば起こりうる。その場

²経済産業省「コーポレート・ガバナンス・システムに関する実務指針」2017年3月31日、p.8によると、中長期的な戦略決定や、CEO等の経営トップの後継者計画等に関する議論が不足していると考えている企業は4割程度存在していた。

合、専門スキルを維持・向上するためにチームとは別の専門コミュニティにも属し、自らの技術についてブラッシュアップすることがアジャイル型チームでは推奨される。

取締役会に置き換えると、社外取締役を中心とした任意の委員会を活用し、社内取締役中心では議論することが困難な課題の解決を目指すこと等が該当する。また、外部研修やコンサルタント等の活用も検討の余地があるだろう。

【継続的に改善活動を続ける】

アジャイル型はPDCAを短期間に回す点が特色の一つとなるが、PDCAを回す際にはCとAの密度について特に意識することが重要である。障害はどこにあるか、課題についてどのように対処するか、これらをメンバーで振り返る時間を十分に確保することが大切である。

この改善活動を繰り返すことで個人の暗黙知はメンバー全員の形式知となり、形式知はまた個人の気付き（暗黙知）となって螺旋的に増幅していく。

【アジャイル型は万能ではない】

アジャイル型は、原則として計画よりも変化への対応を重視する。そのため、急激な変化が起こる可能性があり、イノベーションが求められるような業務については最大限の効果を発揮するだろう。逆に、アジャイル型のデメリットとして、変化への対応やコミュニケーションを重視する一方で、計画やプロセスが疎かになる懸念もある。そのため、中長期的に情勢が安定している場合や、当初から目的が明確であり変更が発生しにくい場合にはフィットしない。自社の置かれている状況や課題ごとに、必要に応じて使い分けることができればベストである。

—以上—

参考文献

- 平鍋健児、野中郁次郎 著「アジャイル開発とスクラム 顧客・技術・経営をつなぐ協調的ソフトウェア開発マネジメント」翔泳社、2013年
- ダレル・K・リグビー、ジェフ・サザーランド、竹内 弘高 著「アジャイル開発を経営に活かす6つの原則 DIAMOND ハーバード・ビジネス・レビュー (2016年9月号)」
- アジャイルソフトウェア開発宣言
(<http://agilemanifesto.org/iso/ja/manifesto.html>)