

皆さんは“Agile”という言葉をお聞きになったことはありますか？

ソフトウェアの開発現場ではかなり馴染みがある言葉だと思いますが、それ以外の業界では聞いたことはあっても“Agile”って何？』といった方も多いと思います。私も同じです。プロジェクトマネジメントの世界標準である PMBOK Guide® 第 6 版でもプロジェクトマネジメントの一手法として明記され、「Agile」を解説した“Agile Practice Guide” が別冊として発刊されたことを受け、昨年 12 月、JSPE で開催している鬼金セミナーでこの「Agile」をテーマに取り上げました。今回はこの「Agile 開発」についてご紹介したいと思います。

### 1. Agile 開発の考え方とは

まず「Agile 開発」と聞くとソフトウェアの開発に特化した開発手法というイメージをお持ちになるかもしれませんが、そうではありません。ソフトウェア開発から生まれた考え方ではありますが、その考え方は広く様々なプロジェクトで適用できるものです。では、その考え方とは？

まずは最初に「Agile 開発」という考え方が生まれたソフトウェア開発業界の開発方法および、ソフトウェアベンダーがこれまで抱えていた課題を共有しておく必要があります。多くのソフトウェアベンダーは顧客から出された要求仕様やヒアリングを通じて、開発すべきソフトウェアの製品仕様を作ります。ここで問題になるのが顧客は最初から自分が欲しいシステムの全体像を明確には伝えられないということです。こんな感じ、こんな風といったおぼろげなイメージから欲しいシステムをイメージ、文書や絵、場合によっては口頭でそのシステムを表現し、ソフトウェアベンダーに開発を依頼します。ソフトウェアベンダーは顧客とのやり取りを通じて要求仕様を理解し、製品仕様に落とし込んでいきます。顧客と、事前にしっかりとした製品仕様を作りこむことが後戻りや変更のないソフトウェア開発プロジェクトの成功の秘訣であるとされてきました。しかし、実際のところはどうか？ どんなに顧客と対話し、何度も確認して一緒に製品仕様を作りこんでも、できてきたものを見て顧客は一言。「イメージとちょっと違うんだよね」「もうちょっとここがこうならないかなあ」。ソフトウェアベンダーにしてみれば「作る前に確認したでしょ！」「それならそれと言っといてくれよ！」ということになります。当然、追加の開発が発生します。スケジュールや仕様に変更がかかります。この追加、変更作業にかかる費用を顧客が出してくれるのならば、ソフトウェアベンダーはまだ頑張れるでしょう。しかし、そう簡単ではないのです。「イメージしたものと違うからお金は払えないね。納得できたら支払うよ」顧客が納得するまでタダ働き…これこそがソフトウェア業界が抱える大きな課題の一つなのです。そこで登場したのがこの「Agile 開発」なのです。

### 2. Agile 開発の用語と Manifesto とは何か

Agile 開発と言えば、「Sprint」や「Scrum」、「Retrospective」、横文字ばかりで、なにかとハイテクな感じがしますが、その基本は「Agile Manifesto」に沿って開発を進めましょう」という、とてもシンプルなものです。「Agile Manifesto」がこれです。

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:	
Individual and interaction over processes and tools	プロセスやツールよりも個人と対話を
Working software over comprehensive documentation	包括的なドキュメントよりも動くソフトウェアを
Customer collaboration over contract negotiation	契約交渉より顧客との協調を
Responding to change over following a plan	計画に従うよりも変化への対応を
That is, while there is value in the items on the right, we value the items on the left more.	これらを価値とする。

というものです。

要は「顧客との対話を通じて、実際に動くソフトウェアを見せながら、顧客のほしいものを作っていきますよ。変更が必要ならばできたソフトウェアを見て、どんどん言ってくださいね。追加、変更しますから」ということです。最初に何をやるかをしっかり決めて、そこからの変更を管理しながらできるだけ変更少なく開発を進めるという従来の手法に対して、「Agile 開発」は、できたソフトウェアを見せながら、顧客と一緒に順次、機能を追加、変更していくという手法なのです。

### 3. Agile 開発の利点は何か

では、一体「Agile 開発」の何が良いのか？ソフトウェアベンダーにとって「費用の取りっぱぐれがない」これが一番の魅力です。ベンダーは顧客に対して「その機能を実現するのにこのくらいの期間と費用が掛かりますがどうしますか？」と作業前に確認します。OK がでたら作業に入ります。できたものを見てもらい、顧客が満足したらそこで開発は完了です。もしも満足できなければ、どう変えたいか、どんな機能を追加したいか要望を聞きます。そしてその要望を満たすための開発期間と費用を提示して、開発を続けるかどうかを都度確認して進める、これこそが「Agile 開発」なのです。そのために重要になってくるのが「できるだけ早く「動くソフトウェア」を見せ、顧客に短いサイクルで何度もフィードバックをもらって有償で改良していく」ということです。ここでいう「動くソフトウェア」はいわゆるプロトタイプではないのです。しっかりとした「製品」であることが必須です。「Agile 開発」では、そのまま納品可能な製品として顧客に見てもらうことで、納得してもらった時点でいつでも契約完了です。

### 4. 多様な Agile 開発手法

このようなサイクルを速く回すために、「Sprint」や「Scrum」、「Retrospective」といった様々な開発手法が生まれてきたというのがその歴史です。では、いくつかの開発手法を簡単に紹介したいと思います。

まずは「Sprint」。その名の通り、「短距離走」です。無呼吸で走れる距離でないと息が続きません。息継ぎなし = 1 週間～4 週間程度で達成できる目標、作業内容を設定し、一気に開発します。できたものを顧客に見てもらって OK をもらいます。OK がもらえたら次の機能開発へ、もらえなければ手直し、改良を新たに「Sprint」します。「Sprint」ごとに成果物を出し、その成果物に対する顧客や上司のフィードバックによって次の「Sprint」で何をやるかを決めていく。「Sprint」が 1 巡終わったところで計画を見直しますので変更にも柔軟に対応していけるというのがこの仕組みのメリットです。

この「Sprint」を回していく活動が「Scrum」です。ラグビーのスクラムのイメージそのまま、製品開発するためのチームを「Scrum Team」と呼びます。「Scrum」では各メンバーが役割を分担しています。出来てくる製品に

責任を持つ人、プロセスを円滑に回すための人、開発する人、などなど。この「Scrum」では先ほどの「Sprint」を回しながら、毎日「Daily Standup」を行います。「Daily Standup？」横文字で言われるとおしゃれですね。要は毎朝のミーティングです。毎朝、立ったままで 15 分程度、“前回の Daily Standup ミーティングからやったこと”、“次回の Daily Standup ミーティングまでにやること”を宣言し、自身の仕事をコミットしていくのです。また、作業の障害となるモノが出てきていないかも確認する場です。ただし、この会では解決策を探すことはしません。あくまでも説明責任の醸成の場であり、状況確認会議ではないのです。このようにして、プロジェクトの進捗の確認と支障となる事案が出てきていないか確認していくそうです。

そして「Sprint」を 1 巡、終わった最後には「Retrospective」で締めます。「Retrospective」、かつこいいですね。でも、中身は「振り返りの会」です。一度回した「Sprint」で得た知見や反省を次の「Sprint」を効率よく回すための改善につなげていく会です。各人の仕事の処理能力も反映させながら、次の「Sprint」でどのくらいの作業がこなせるのかを計画していきます。この「Retrospective」を行うことで「Scrum Team」は作業に対する見積もり精度や業務遂行能力を上げていくのです。

と、ここまで読んでいただいた皆さんは感じているのではないのでしょうか？ソフトウェア開発で流行ってるらしい「Agile 開発」、今まで自分たちがやってきたことと似ているのではないか？日々の業務の中で「如何に顧客とコミュニケーションをとるか？」「上司やメンバーとの報連相を如何に密にするか？」「日々の改善を如何に進めるか？」。

「アジャイル開発」で取り組まれている開発手法はどれも、「顧客のフィードバックを如何に早く取り込み、チーム内でのコミュニケーションを高めることで円滑に業務を進め、より良い最終製品にしていく」というエンジニアリングや製品開発に共通の課題への対応策と同じような手法です。言い回しが違ってもこんなにも印象が変わるのか、“Agile Practice Guide”を読んだ私の正直な感想です。

今まで開発現場に導入にし辛かった泥臭い手法も、言い回しを変えるとやってみたいと思ってもらえるのでは…ちょっと“Agile”試してみたくくなりました。

以上