

日税メールステーション 特別号

税理士が知っておくべき IT 知識～基本の"き"～

第13回 アジャイル(1)

メールマガジンをお読みの皆様こんにちは、株式会社コアブリッジの柳です。

2021年9月29日付の日本経済新聞「解説 経済白書」欄に『ソフト開発手法 硬直的』という記事が載っていました。

要約すると、

「日本でシステム開発に主に用いられている従来方式は時間、変更対応、費用面等で課題が多い。これらの課題に対して柔軟性のあるアジャイル開発を広め、そのノウハウを持つ人材を確保すべきだ」

というものです。

そこで、前号の「システム開発」の続きとして、「アジャイル」開発について取り上げてみます。

■VUCAの時代の2つの課題

"VUCA"という言葉をご存知でしょうか。Volatility(変動性)、Uncertainty(不確実性)、Complexity(複雑性)、Ambiguity(曖昧性)の略で、予測が困難な状況を意味し、裏を返せば、いかに変化に対応していくべきか、ということを表します。1990年代から米国の軍関係で使われている単語だそうですが、最近頻繁に見聞きするようになりました。

日本人は、計画をしっかりと立て、リスクを洗い出して入念に判断し、高品質を必須とする…というやり方を得意としますが、VUCAはこれが適用できない状況を表します。こと、新規ビジネスやシステム開発にはこれが当てはまり、対応できなければ時間を浪費し置いていかれる、というのが現実です。

話をシステム開発に絞ると、日本のシステム関係の特性として「計画性と高品質を重視する」

「外部委託比率が高い」が挙げられます。これが「他国の周回遅れ」と揶揄される（自虐している）日本のIT業界の課題です。

システム開発の方式には複数ありますが、「計画性」や「外部委託」と相性が良いのが「ウォーターフォール型」開発、これとは反対の特性を持つ「動くものを作って評価改良を反復的

に継続する」というのが「アジャイル」です。

この2方式について解説と比較してみます。

■従来方式ウォーターフォール(waterfall)型

システム開発の方式にはいくつかありますが、日本では、冒頭で"従来方式"と記した、ウォーターフォール型が多いです。統計データ(※)によると 97.4%が採用しており、この数字は過去 10 年以上変わっていません。正直極端な数字という印象で、アンケートの取り方(対象、選択肢)によるところも大きいとは想像していますが、傾向としては正しいでしょう。この開発方式は 1970 年に論文が書かれています。一つの開発工程を終わりにしてから次の工程に移る、ということが続け、水が段々に流れていく様からウォーターフォールの通称で呼ばれるようになりました。

これがシステム開発の最も基本的な手法とみなされ、この手法の短所を補うために、その後いろいろな手法が編み出されてきました。

ウォーターフォールの短所とは

- ・完成までに時間がかかる
- ・後工程で要件(システムで何を実現するか)に変更が入ると手戻りの影響が大きいです。

逆に長所は

- ・全員が足並みを揃えて同じ工程を実施しているので管理がしやすい
- ・上流と下流の区分けが明確で分業がしやすい

です。

この長所が、日本でシステム開発の外部委託比率が高いことと直結し、言い換えれば、ウォーターフォール以外の方式を取りにくい原因の一つです。

■ウォーターフォールの短所を行う方法

この方式の本質は「大きなものは小分けにする」という、"リスクヘッジ(risk hedge)"の基本中の基本をシステム開発に適用したものです(資産運用などでも全く同じですね)。システムの分析をしてプログラムを書く、という二段階だけではリスクが大きいため、もっと工程を分割してマネジメントすべき、ということです。この根本は正鵠を射ており、ウォーターフォール型が間違っているわけではありません。時代が進むにつれて、それだけでは足りなくなり、さらなるリスクヘッジの多様化が必要になった、というのが妥当な見方でしょう。ウォーターフォールの短所を補う方式として、試作品を作って完成系のイメージを掴んでか

ら本番の開発に進む「プロトタイプ」方式、ウォーターフォールを複数に分ける「スパイラル」方式（一連の工程を一周、二周・・・と螺旋のように回す）、実現すべき要件を複数に分割して少しずつ追加していく「反復」方式、などが編み出され使われてきました。そして、現在は、この反復式が進化した「アジャイル」式が最も重要視されています。

■アジャイル

Agile は「俊敏な」という意味で、「変化に素早く対応し、価値を生み出す」ことを最優先に行動します。

2001年に"アジャイルソフトウェア開発宣言"と題して

"プロセスやツールよりも個人と対話を、包括的なドキュメントよりも動くソフトウェアを、契約交渉よりも顧客との協調を、計画に従うことよりも変化への対応を"

という思想が世に出されました (<https://agilemanifesto.org/iso/ja/manifesto.html>)。

先述の日本のシステム開発の特性と逆であることは説明不要でしょう。

このマニフェスト(宣言)自体は、具体的な方法論ではなく哲学のようなものです。アジャイル型の開発がどのようなものか、先に挙げた課題への対策には何があるのか、続きは次号にて記します。

今号は以上です。

では、また次回お会いしましょう。

※本文中の情報、状況、数値等は執筆時点のものです

執筆者

柳 恵太 (やなぎ けいた)

株式会社コアブリッジ代表取締役。

ソフトウェア開発会社、メーカー、教育ベンダーを経て、2014年に株式会社コアブリッジを設立。これまでの、システム開発の上流から下流、受託側から発注側、エンジニアからプロジェクトマネージャー、ユーザーと開発者、企画・営業・開発・提供、日本と海外、社員から経営者といった、組織における幅広い役割を活かし、主にIT企業向けの人材育成やコンサルティング等のサービスを提供している。

情報提供元：



株式会社コアブリッジ

<https://www.corebridge.co.jp/>

※本コラムは、<https://www.corebridge.co.jp/column/> でもご覧になれます。