

# 欧米で普及するアジャイル開発

—俊敏で柔軟なシステム開発のために—

経営環境の変化に合わせてシステム開発をスピーディーに行う方法としてアジャイル（俊敏な）開発が注目されて久しいが、日本ではそれほど普及していないのが現状である。本稿では、日本でアジャイル開発が普及しない原因を考えるとともに、あらためてアジャイル開発の有効性を確認することにしたい。

## 経営スピードに追いつけない情報システム

企業が情報システムに求める役割は、経営環境に応じて変化する。業務効率化や事務コスト削減のための情報システム投資がほぼ一巡したいま、多くの企業が情報システムに求めているのは、新しいサービスや事業を立ち上げ軌道に乗せるための支援ツールという役割である。これは、情報システムがビジネス上の戦略的要素、あるいは他社との差別化要因になっているということである。

このような情報システムの開発においては、俊敏性や柔軟性が従来のシステムよりもはるかに強く求められる。システム開発の遅延やビジネス環境変化への対応のまずさが競争力を失うことに直結するためである。

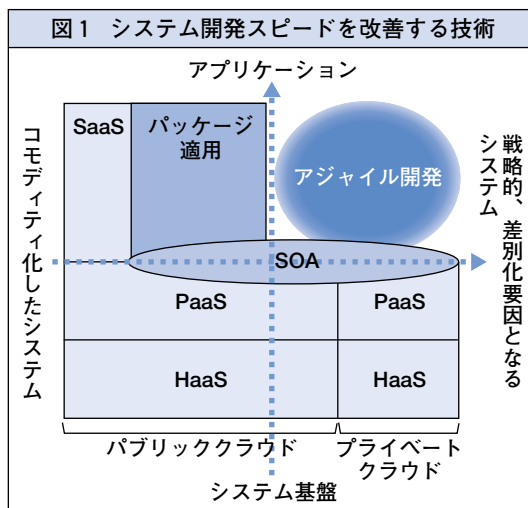
しかし、このような俊敏で柔軟なシステム開発が実現されているケースは少ないのが現状である。柔軟なアイデアのシステムを俊敏に構築したいという経営の要求に対して、従来のシステム開発の考え方では詳細な要件定義を固めることがまず求められ、しかもその構築には1年、2年という期間がかかることも珍しくない。それが間違っているわけではないが、少なくとも経営が求めているスピー

ドにマッチしていないことは確かであろう。

## スピーディーなシステム開発のために

いま、経営環境の変化とシステム開発スピードの乖離を解消することを期待できるさまざまな技術が登場している（図1参照）。

サーバーやネットワークなどシステム基盤の俊敏な構築や柔軟な拡張を可能にするものにはクラウドコンピューティングがある。また、ERP（統合基幹業務システム）のように比較的变化が少ないアプリケーションは、ソフトウェアパッケージやSaaS（Software as a Service：ソフトウェアの機能をインターネット上のサービスとして利用する仕組み）



野村総合研究所  
システムコンサルティング事業本部  
ITアーキテクチャーコンサルティング部  
上級テクニカルエンジニア

平田 正 (ひらただし)

専門は大規模トランザクションシステム・Webシステム・  
EAIシステム・データベースシステムの設計・開発など



によって素早く構築するのが一般的になってきた。再利用可能なサービスとして機能を実装するSOA（サービス指向アーキテクチャ）も、サービス間接続やサービスの自由な組み合わせを容易にすることで、システムの俊敏性や柔軟性の向上に貢献する。

一方、差別化要因となるアプリケーションや、要件が変化し続ける戦略的なアプリケーションの開発については、俊敏な開発手法であるアジャイル開発が注目されている。

## 米国で普及するアジャイル開発

アジャイル開発とは、小規模な開発とリリースを繰り返し、積極的にフィードバック（評価）と変化を受け入れる反復型開発手法の1つである。頻繁なリリースにより全体のシステム開発の進捗状況を可視化できただけでなく、ビジネス環境の変化に俊敏かつ柔軟に対応するのに適した方法である。

欧米で情報システムがビジネス上の戦略的要素といわれるようになり、変化への適応が問題とされたのは1990年代後半である。2001年頃にはアジャイル開発がブームとなりはじめ、2005年頃から、最新ITの採用に先進的な金融業界を中心に普及していった。

2010年に米国のForrester Research社が米国のシステム開発プロジェクトで採用されている開発手法を調査したところ、アジャイル系が35%、反復型が21%、ウォーターフォール系（一連の工程を順次完成させていく従来型の手

法）が13%であった（「The Forrester Wave: Agile Development Management Tools, Q2 2010」）。ガートナーの予測によると、「2012年までにソフトウェア開発プロジェクト全体の80%でアジャイル開発方法論が利用される」という（出典：Gartner「2010年の展望：アジャイルとクラウドがアプリケーション開発に及ぼす影響」 T. Murphy他共著 2010年4月15日）。

これに対して、日本ではアジャイル開発はまだそれほど普及していない。野村総合研究所（NRI）が2009年に実施した「ユーザ企業のIT活用実態調査」では、アジャイル開発を「ほぼ実施」または「ある程度実施」していると答えた企業は合わせて13.6%にとどまった。また、財団法人情報処理推進機構（IPA）の『ソフトウェア開発データ白書 2009』によれば、反復型開発を採用したプロジェクトは全体のわずか2.8%とされている。

## アジャイル開発の要点

なぜ日本ではアジャイル開発があまり採用されていないのだろうか。以下では3つの観点からその理由を考えるとともに、アジャイル開発の要点を併せて解説する。

### (1) 価値観の転換

日本の開発現場ではアジャイル開発を「ドキュメントを書かない」や「頻繁に仕様変更を行う」といった表面的な特徴でとらえ、その特徴だけを模倣して失敗しているケースが

見受けられる。確かにそれらはアジャイル開発の特徴ではあるが、その特徴を生かしつつも品質が低くならないための考え方や方法を持っているのがアジャイル開発である。そういったアジャイル開発の本質について関係者が十分に理解しないまま表面的なやり方だけをまねてプロジェクトを進めれば、プロジェクトは混乱するばかりである。

重要なのは、アジャイル開発にはシステムについての価値観の転換があるということである。アジャイル開発では、個々人の問題解決能力を信頼し、動くソフトウェアを重視し、顧客との信頼関係を築き、変化を柔軟に受け入れることについて、関係者全員が理解し同意する必要がある。もちろん価値観の転換は簡単ではない。1つの機能を少数で機敏に開発するので組織体制もフロアレイアウトもそれに合わせる必要がある。マネジメント手法や人事評価方法、業務プロセス、品質保証プロセス、開発標準などの変更も必要になる。そして「アジャイル開発に取り組む」という経営の強い意思も大切である。

## (2) 成長するアーキテクチャ

アーキテクチャに関する固定的な考え方も、アジャイル開発に踏み切れない要因の1つである。従来のウォーターフォール型開発では、システムのライフサイクルすべてをカバーする堅固なアーキテクチャを構築することに労力を注ぐ。例えば「当初はマスター更新は1日1回だが将来のリアルタイム更新に対応で

きるようにする」とか、「サービス時間は9時～17時だが将来に備えて24時間365日のシステムにする」といった話はよく聞かれる。そのこと自体が悪いわけではないが、変化の激しいビジネスアプリケーションに過剰な投資をしているケースは少なくない。

アジャイル開発では、ある程度のアーキテクチャ設計を行えば、プロジェクトの開発期間を通じて徐々にアーキテクチャを成長させていけばよいと考える。アーキテクチャ設計は、システム構成の決定と非機能要件（性能・信頼性・セキュリティなど機能以外の要件）の実現に大きく分けられる。

システム構成の決定とは階層分け（プレゼンテーション、ビジネスロジック、データストアなどの階層とそのインタフェースの決定）のことである。一般的には、一緒に変化すると思われるものを1つにまとめ、ほかから隠すことで変化への対応を容易にする。層間のインタフェースは、ビジネスや技術の変更への適応コストを低く保つようにする。

各階層内に配置されるコンポーネント（プログラムの部品）は、可能な限り独立に開発可能で、それぞれの実装の仕方に依存しないことが大切である。この特性によって自動単体テストによる品質保証が可能になる。

このほか、アーキテクチャとしてはコンポーネントの物理配置、アプリケーション機能の割り当て、永続データの構成（データベースのテーブル構造）などを規定する必要がある

る。これらは、事前のアーキテクチャ設計で決定することもあれば、アーキテクチャを成長させていく過程で決定することもある。どこまでを事前のアーキテクチャとして準備しておくかは、柔軟性や安定性、俊敏性と規律のバランスを考慮して判断する。

アーキテクチャのもう1つの要素である非機能要件は、アプリケーション開発上の大きな制約条件である。非機能要件による制約を意識せずに開発されたアプリケーションに対して、後から性能や信頼性を与えることは困難である。従って、非機能要件は事前のアーキテクチャ設計に含めておく必要がある。

一方で、非機能要件の実現に当たっては、ビジネス環境、プロジェクトの規模や条件、変化の頻度、採用技術の成熟度、システムの耐用年数などを考慮する。これはウォーターフォール型のような過剰投資を避けるために必要である。

### (3) 契約形態

契約の形態も、日本でアジャイル開発が普及しにくい大きな理由の1つである。日本でよく行われる受託開発は、費用と期間と成果を最初に約束する請負契約が一般的である。しかし、成果を固定した契約形態は、変化を柔軟に受け入れるアジャイル開発には向いていない。請負契約では、受注側が開発に伴うリスクをすべて引き受けることになるため、受注側は事前に詳細な要件定義を要求する。それに加えて、リスクを回避するために費用

も期間も過大な見積もりをしがちである。その結果、システムは柔軟性を失い、肥大化することになる。

これを防ぐ方法は、成果物ではなく作業に対して支払う準委任契約で開発を行うことである。準委任契約は、発注側にとって「最終的に何ができるのか分からない」というリスクがある。しかしこのリスクは、アジャイル開発では早期にかつ頻繁に、実際に動作する成果物を提供することになるために軽減される。すべてのリスクを受注側に押し付けてしまう請負契約に対し、準委任契約ではリスクの一部を発注側が引き受けることになるので発注側のリスクコントロールが働く。そのため、より少ないコストでより良い結果を引き出しやすいのである。

請負契約の場合でも、「要件を同等規模の要件で入れ替え可能」というように、約束される成果を固定的にしないための付帯条件を設けることでアジャイル開発を行うことは可能である。

日本ではアジャイル開発が欧米ほど普及していないと述べたが、情報システムをビジネス上の戦略的要素、差別化要因と位置づける先進的な企業を中心に、徐々にアジャイル開発への取り組みが進んできた。野村総合研究所（NRI）でも2004年からアジャイル開発の研究・開発を進めており、アジャイル開発に取り組むユーザー企業を支援している。 ■