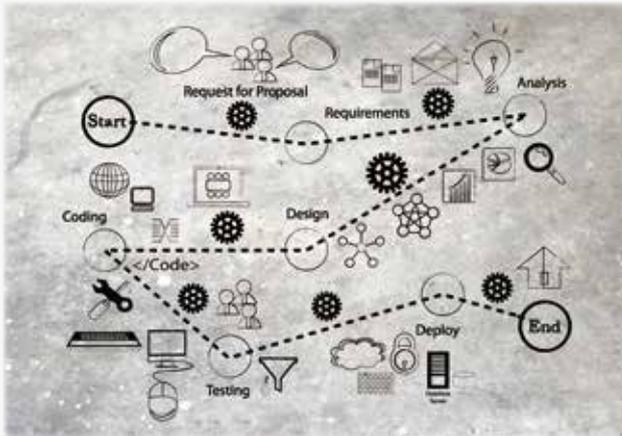


アジャイル開発と業務の自動化

アジャイル開発の特徴

機能単位の小さいサイクルで繰り返す開発工程
 効率的で素早い開発による短期間のリリース
 仕様変更に強く、開発システムの価値を最大化



リーンソフトウェア開発

リーン生産方式がベース
 (TPS: トヨタ生産システム)

ムダを排除することが原点
 コンカレント開発
 イテレーション
 リファクタリン …etc

RPA など自動化のためのシステムの開発は、「アジャイル型」の開発の手法がよいとされています。

メインフレーム時代から当たり前だった「ウォーターフォール型」の開発の手法では、要件定義、基本設計、詳細設計、開発、テスト、運用などのそれぞれの作業工程を最初に計画し、ガントチャートを使用して順番に一度で工程を終わらせます。基本的には前の工程が完了していないと次の工程に進めないことは、手戻りも少なく、進捗管理がし易いことがメリットでもあります。開発途中で要件の追加や変更があった場合に、仕様変更が難しいなどのデメリットもあります。これに対し、アジャイル型の開発手法は、比較的短期間のイテレーションと呼ばれるサイクルで、小さな単位の設計や、開発、テスト等を繰り返し実装を進めていきます。これにより、開発当初に詳細な仕様を固めなくても開発を始めることができ、開発途中での仕様変更にも柔軟に対応できます。

自動化のためのシステム開発にアジャイル型が向いていると言われているのは、RPA のように業務プロセスを自動化するような目的は、トップダウンの開発より、業務をよく知っている現場の意見を聞きながら、効率化できる仕様を詰めていく方が、より良いものができ失敗が少ないからです。具体的には、RPA ツール等を利用しながら、業務担当者が主体で、個々の業務の作業の流れを画面を見ながら開発できます。業務担当者が RPA ツールで開発しない場合でも、開発担当者がその場でプロトタイプを作成できるので、業務担当者の前で、詳細な動きが確認でき、その場で修正も可能です。アジャイル型の中でも、代表的な「リーンソフトウェア開発」の手法は、トヨタの「かんばん方式」の考え方を元に作られているので、ニブンの自動化を目指す「業務プロセス自動化」にとっては最適な手法です。

その手法の主なポイントは、効率化を図るためムダを認識するところから始まり、チームに権限を与え、決定をできるだけ遅らせ、できるだけ速く提供することです。その中で、学習効果を高め、統一性を取り込み、俯瞰的に全体をみて、システムは単なる部品の「和」ではなく、相互に作用する「積」になるというシステム思考がより生産性を高めます。

トップダウンによる開発方式のウォーターフォール型に比べると、システムを丸投げしている会社にはアジャイル方式は不向きにも感じられますが、ビジネスの変化と速度に対応するためには、アジャイル開発の方が良い面が多いです。

そうすると今後は、現場中心の開発思考でシステムを開発を行うために、外部のリソースに頼る場合でも、その開発ノウハウは、自社の知識ベースとして蓄積していくことが、企業にとって重要になりそうです。