アジャイルプロセスのもたらすインパクト
－インタンジブルアセットの重要性－

（株）一（いち）　副社長　専任コンサルタント
大槻　繁

１．はじめに

システム開発・保守運用、調達、インテグレーション、組織改善の現場は様変わりしつつあります。そのムーブメントの一つが「アジャイルプロセス」に代表されるものです。

官庁や大企業の基幹システム、製造ライン等は、アジャイルプロセスとはかけ離れた世界ですが、いろいろな局面で限界がきているのも確かです。世の中の大部分は、レガシー、ウォータフォール、人月積み上げ提案にどっぷりと浸かっていますし、コストオーバや納期遅延に悩まされるデスマーチプロジェクトも蔓延しています。

最近とりざたされているエンタープライズ・アーキテクチャ、プロダクトラインといった考え方もその背景には伝統的な方法の限界に対する解を模索する動きとみることができます。

２．本質的困難

ソフトウェアエンジニアリングの歴史は、人間の飽くなき欲望を満たす製品を、その時々の技術レベルに応じて、適切な方法で開発・供給するという戦いの歴史であると言えます。

普遍的な課題は、Frederic P. Brooks Jr.が古典的名著『人月の神話』の中で述べたように右記のように集約されます。

Text by Shigeru Otsuki
日立製作所にてソフトウェアエンジニアリングの研究・開発に従事。2004年よりコンサルタント会社一（いち）副社長/専任コンサルタント。ITシステム関連の調達・開発プロジェクトの診断・改善を行うかたわら、コストモデルの研究・開発を進めている。
著作に「ソフトウェア設計」（朝倉書店,1995年）、「ソフトウェアクリーンルーム手法」（日科技連,1997）、「大丈夫かあなたの会社のIT投資」（NTT出版,2002年）他多数。
電子情報技術産業協会ソフトウェアエンジニアリング技術専門委員会幹事、社会保険庁レガシー刷新専門家会議委員，アジャイルプロセス協議会運営委員長・副会長。
http://www.1corp.co.jp

・複雑性（Complexity）：大きいこと、複雑であること、それ自体が本質的な問題です。物理的な構成物とは異なり、単純な部品の組み合わせでは作ることができないのです。

・同調性（Conformity）：単純な原理（物理学で言う統一場理論のようなもの）は存在しません。（気まぐれな）人間の習慣や社会制度に順応させなくてはならないのです。

・可変性（Changeability）：使われれば、使われる程、機能拡張や新機能の要求が増えます。システムが社会に組み込まれるが故に、絶えず変化し続けなくてはならないのです。

・不可視性（Invisibility）：物理的な世界ではなく、概念の世界でしか捉えることができません。構造を抽象化して顕在化する手段がないのです。

テクノロジーの潮流を見極めて行くために、この30年前に設定されたシステムやソフトウェアの本質的困難という課題をどの程度解決しているかという視点で評価するのはとても有効です。

サブルーチンやクラスといったモジュール化の概念は、「複雑性」に対する解です。構造化プログラミングや良構造のアーキテクチャといったものは、「同調性」に対応するものとみなせます。

ここで採り上げているアジャイルプロセスはおそらく「可変性」と「不可視性」に強く関連していると思えます。

３．アジャイルプロセス

アジャイル（agile）とは「俊敏な」という意味です。アジャイルプロセスの定義は、「アジャイルマニフェストに合意していること」というだけで、テクノロジーの視点からは曖昧ですが、思想的には大変深いものがあります。アジャイルマニフェストとは、以下のように簡潔なものです。

・プロセスやツールよりも個人とコミュニケーションを

・立派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

・計画の死守よりも変化への対応を

アジャイルプロセスに関する方法論としてこれまで提案された主なものが上記の一覧表です。多くの人々がさまざまな提唱をしていますし、万能の方法論というのもありませんが、「可変性」と「不可視性」に焦点を当てていることがみてとれると思います。

国内でもアジャイルプロセス協議会が2003年7月に発足し、見積りと契約、プロジェクトマネジメント、メンタリング・コーチング、事例検討等の活発な活動を行っています。

主なアジャイルプロセス方法論

電子情報技術産業協会
ソフトウェア基盤技術専門委員会
平成１５年度報告書より

方法論名称	概　要
エクストリーム プログラミング：Extreme Programming (XP)	提唱者：Kent Beck。コーディング、テストファースト、リファクタリング等、技術プロセスが中心。
スクラム：Scrum	提唱者：Ken Schwaber, Jeff Sutherland。マネジメントにフォーカスした方法論。1ヶ月単位の開発（スプリント）、毎日行うスクラムミーティングが特徴。
クリスタル（ファミリー）：Crystal (family)	提唱者：Alistair Cockburn。マネジメントにフォーカスした軽い方法論。ワイドスペクトラムな方法（小規模～大規模）。継続的なプロセス改善を目標とする。
フィーチャ駆動型開発：Feature Driven Development (FDD)	提唱者：Jeff De Luca, Peter Coad。モデル中心の古典的な繰り返し型開発プロセスで、かつ、軽量。
適応的ソフトウェア開発：Adaptive Software Development (ASD)	提唱者：Jim Highsmith。RADを発展させ、カオス適用理論(CAS)を用いたフレームワーク。最適なものに改良していく。
動的システム開発方法論：Dynamic Systems Development Method (DSDM)	「初めから完全には構築できない」という前提に立った探索的な開発手法。RAD、JADをベースとして、プロトタイプを多用する。
リーン ソフトウェア開発：Lean Software Development (LSD)	提唱者：Mary Poppendiek。トヨタのカンバン方式（最小在庫＝ドキュメント）の原理応用。マネジメント、組織的経営にフォーカス。
エクストリーム モデリング：Executable and Translatable UML (xtUML)	提唱者：OMG-MDA等。検証実行可能なモデリング（ツール）を利用。マネジメント的側面はない。

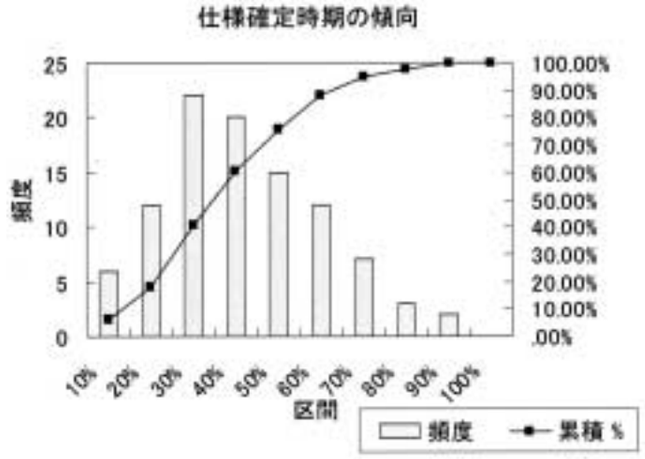
４．変化への対応

システムは、あらかじめ仕様を決めておいて、それを開発すればよいという状況は希です。製品のニーズも時々刻々と変化するでしょうし、システムが社会の中に組み込まれているがゆえに、環境に迅速に対応し、変貌もして行くものです。

次のグラフは、典型的な組織での約100件のプロジェクト開発において、仕様確定がプロジェクト期間中のどの時点（開始時点を0%とし、終了時を100%）で行われたかを示したものです。仕様を確定することができる時期というのが予

想以上に遅く、プロジェクト期間の半分を過ぎても仕様確定に至らないプロジェクトが1/4程度あるということがみてとれます。

このことは、従来のウォータフォール型のプロセスが適していないことを意味していますし、インクリメンタルやエボリューショナルプロセスの導入といった工夫をしなくてはならないことにもなります。



５．外部特性と内部特性

アジャイルプロセスは、現段階ではどちらかと言うと開発側の世界での動きに限られています。しかし、ビジネスとしてのシステム開発というのは、発注と受注、ユーザと開発といった社会的コード（規範）の中でとらえるべきものです。

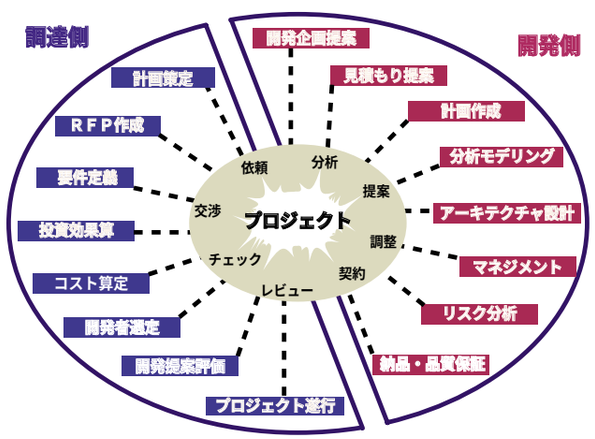
典型的な調達・開発型のプロジェクトでも少なくとも二つの組織が絡み合い、さまざまな交渉、調整、協調を含むアクティビティが展開されます。

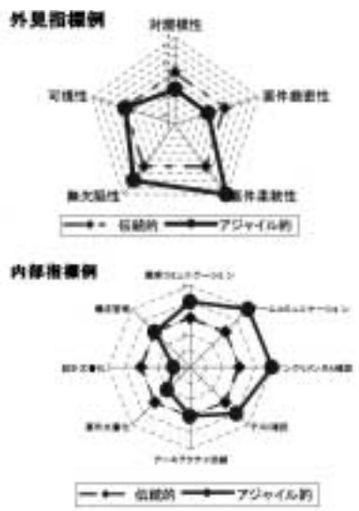
ここで問題になるのが、複数の組織の間のインタフェース（界面）です。例えば、ＸＰ（eXtreme Programming）のプラクティスとして以下のものがあります。

・計画ゲーム
・短期リリース
・メタファ（比喩）
・シンプルな設計
・テストファースト
・リファクタリング
・ペアプログラミング
・共同所有
・継続した結合
・40時間労働
・オンサイトの顧客
・コーディング規約

これ等は必ずしも調達側から観測可能なものではありません。おそらく調達側からは、早期に仕様の確認ができること、高い品質のものを得たいことといったプロジェクトあるいはシステムの外部から観測できる特性を要件として掲げているはずです。

これは一般のプロジェクトでの要件と実現との間でトレーサビリティを確保するのと同様のアプローチが必要と思われます。





６．アジャイルコストモデル

ソフトウェアエンジニアリングの領域でコスト分析や見積り手法ほどニーズとの乖離が激しいものはありません。現実的には以下のものを組み合わせているのが一般的です。

・アナロジー（事例類推）：同種のシステム開発データを適用

・単純推論：システムのコード行数予測と、人月あたりの開発コード行数実績から推測

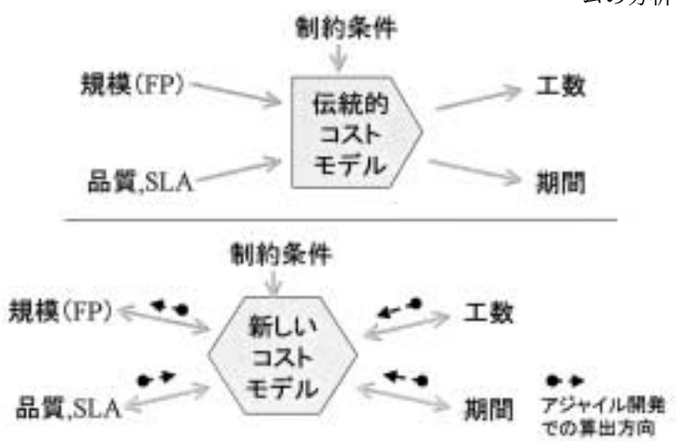
・専門家判断（デルファイ法）：複数の専門家の予測を集計。集計結果を再度専門家に見せて修正。これを収束するまで繰り返す。

・作業展開積み上げ：開発作業を分割・展開し、それぞれの個別作業のコストを集積

・３点見積り（PERT法）：リスクを勘案して見積もった「悲観値」「楽観値」「最可能値」を基に数式で算出。期待値＝（悲観値＋４×最可能値＋楽観値）÷６

・構成要素積み上げ：システムの構成要素の特性を集積して外部ポイントを計測
例：ファンクションポイント（FP）法

・回帰分析：過去の開発データを回帰的に分析してコストを算出
例：COCOMO: COnstructive COst MOdel



要件が変化することを前提としたアジャイルプロセス向きのコストモデルは未だありません。インクリメンタルプロセスを活用し、仕様確認を早期段階に行うことによるリスク低減に見合うコスト負担をどのように算定すべきかというモデルも必要でしょう。投資とコストという対峙の観点から、システムや製品のライフサイクルといった総合的は判断を助けるものもビジネス上の意思決定で必須です。

アジャイルプロセスを含めて新しい世代で望まれるコストモデルでは、複数の組織間にまたがる合意が得られる適正な指標をベースにしていること、プロジェクトデータに基づく数学的で客観的な理論をベースにしていること等が要件として掲げられるでしょう。

７．全体最適化

前節まではプロジェクトという視点ですが、さらに広くプロジェクトの総体としてのプログラム、組織全体の活動という観点からの新しい動きとしてエンタープライズ・アーキテクチャ（EA: Enterprise Architecture)、およびプロダクトライン（Product Line）というアプローチが台頭してきています。

業務・システムの評価・分析を起点とした情報基盤を整備するための枠組みはEAによって方向付けがなされています。これは、経済産業省の「情報経済基盤整備（高度な電子政府システム構築のための政府調達改革事業)」、総務省の「業務・システムの最適化に係る支援作業」の中で推進され、主として府省全体の業務・システムの分析のベースとなっているものです。

官庁関連のシステム構築・運用は、省庁間の連携、システムの共通化といった発想は育ちにくい環境でした。昨今の調達や入札に関する問題や、意思決定の透明性、とりわけ、国民への説明責任という観点からも抜本的な改革がのぞまれ、EAという形で結集してきたものと思われます。

いわゆるレガシーシステムと呼ばれている伝統的な手法で開発された大規模なシステムが多いものの、国民へのサービスの多様化、法令を起点とした毎年の保守・改変のニーズ、独立行政

法人化や外郭団体の統廃合といったこともあり、適確な「変化への対応」がのぞまれる領域でもあります。この意味で「複雑性」にも対処可能なアジャイルプロセスのニーズは高まっています。

一方、プロダクトラインというのは同種のシステムを継続して開発するためのアプローチとして近年注目を集めています。特定の市場分野あるいは使命に特有の要求を満たし、事前に蓄積された知識を用いて、共通のコアセットから開発される、共通･管理されている機能を共有するソフトウェア集約型システムのことを総称して、プロダクトラインと呼んでいます。製品ファミリレベルでの再利用実現に向けた取り組みを提示しています。

従来ドメインエンジニアリングと呼ばれていたものの発展形として位置づけることができます。プリンタや携帯機器といった製品群をシリーズ設計をするということは、安定したものと変わりうるものとを峻別し、究極的には市場に俊敏に対応して行くアジャイルプロセス思想の具現化の一つとして考えることができます。

８．インタンジブルアセット

アジャイルプロセスが従来のアプローチと根本的に異なっているのは、人間の知識創造活動のあるべき姿を真剣に考え、モチベーション、コミュニケーションといった非技術領域に焦点をあてた点にあります。

一般の企業活動においても、近年、インタンジブルアセット（目に見えない資産）の重要性が認識されはじめています。コンピュータ機器導入といった明示的に貸借対照表に現われるものはほんの一部であって、その企業の本当の資産は組織、チーム力、人的資産といったインタンジブルなものが九割を占めると言われています。

システム開発・保守・運用といった諸活動も知識創造活動の真の資産を増やして行く戦略をとる必要があります。これを達成するための原則は以下のように考えられます。

・原則１：透明性確保の原則
　システムに関する意思決定や、成果物が第三者からその妥当性が評価できるようにしておくことは、判断基準をノウハウとして蓄積するのに役立ちますし、経営層、クライアントへの説明責任上も有効に機能します。
・原則２：インタフェース設定の原則
　システムにおいては、利用者を中心とした業務領域と、システム開発者の技術領域とがあり、両領域の共同作業で開発・運用プロセスを推進して行く必要があります。世界が異なる人々が共通に理解しあえる界面を設定する必要があります。
・原則３：成熟度向上の原則
　理想的な目標を掲げ、そこに至る道筋をつけることが要請されている場合、現状のレベルに合った、適切な実践項目を設定して行く必要があります。

９．おわりに

日本の産業界の強みとして有望視されている組込みシステムの分野においても、ソフトウェアエンジニアリングの重要性がますます重要になってきています。数名の熟練したエンジニアで、職人芸的に開発してきた製品も、規模も大きく、製品も多様化してきたために、系統的な開発技術を活用し組織的に開発しなくてはなりません。そして、新しい時代では組織の協調関係を適正に保って行く必要があります。

それぞれの組織の強みをインタンジブルアセットとして蓄積し、組織間のインタフェースを設定し、ビジネスプロセスからアーキテクチャやプロジェクトまでを統合的にマネジメントして行かなくてはなりません。

計量経済学の分野でも注目を集めている『モジュール化』の概念も、ソフトウェアエンジニアリングのモジュールをビジネスプロセス、組織論、産業論にまで援用したものです。

アジャイルプロセスは、このモジュール化という観点から言うと、調達組織、開発組織といったそれぞれの知的資産としてのコアを生かしつつ、ビジネスからシステム開発というものをプロジェクトとチームいう局面で切り出し、組織間・内、チーム間・内の連携、コミュニケーション、合意形成、確認の仕組みを提供する新しいパラダイムと言えるでしょう。

（おおつき　しげる）