非ウォーターフォール型（アジャイル）開発の動向と課題

―IPA/SECにおける4年間の調査・検討から明らかになったこと―

SEC エンタプライズ系プロジェクト
プロジェクトリーダー
山下　博之

SEC エンタプライズ系プロジェクト
研究員
柏木　雅之

市民生活や社会経済活動におけるITサービス及びそれを実現するITシステムへの依存度が高まってきている中で、私たちが安心してITサービスを使い続けられるためには、ITシステムに対し、環境の変化に俊敏に対応することが求められる。非ウォーターフォール型（アジャイル）開発手法は、環境変化への俊敏な対応を可能とするソフトウェア開発手法の一つとして注目されている。IPA/SECでは、約4年間にわたり、アジャイル開発に関する調査・検討を続けてきた。その内容は、経営層の理解促進、技術者に求められるスキルと人材育成、適用領域、契約形態など、多岐に及ぶ。本稿ではその結果について取りまとめる。

1 背景

市民生活や社会経済活動における IT サービス（及びそれを実現する IT システム）への依存度は、IT 導入の目的が、業務効率化によるコスト削減から生活やビジネスの中核を担う戦略的活用へと変遷するのに伴い、量・時間・質の各面において高まってきた。また、市民生活や社会経済活動を取り巻く環境※1 は変化するが、それらの変化も、量的・時間的・質的に拡大傾向にある。

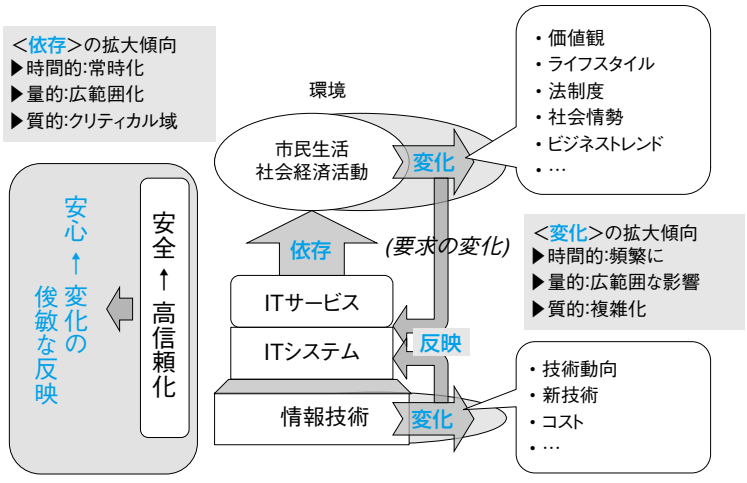

図1　ITサービスの安心のための環境変化対応

このような状況において、従来、IT システムの信頼性を高めて高信頼・安全なサービスを提供することが事業者にとって最も重要な使命であったのが、私たちが IT サービスを安心して継続的に使い続けられるために、環境の変化に対して IT サービスが俊敏に対応することも、信頼性・安全性に加えて重要なこととなってきている（図 1）。

環境の変化に対する IT サービスの俊敏な対応がビジネスや生活に影響を与えたと思われる事例としては、携帯電話各社における学割サービスの導入がある［古川 2011］。A 社は 2010 年 1 月下旬にサービスを発表し、約 20 日後の 2 月上旬に開始した。これに対して競合する B 社は、1 月末に対抗サービスを発表し、5 日後の 2 月初めに開始した。更に C 社は、2 月上旬に対抗サービスを発表したその翌日に開始した。この例では、B 社及び C 社が競合他社の動向という環境変化に俊敏に対応したことにより、一番初めに発表した A 社のサービス開始が最も遅くなってしまったのである。

別の事例としては、携帯電話からスマートフォンへのシフトという、モバイル端末利用動向の変化に対して適切な対応が行われなかったため

に、携帯電話ネットワーク装置の誤動作を誘発し、利用者に多大な影響を与えたトラブルが記憶に新しい。スマートフォンには無通話／操作状態でも一定量の通信を行うという特性があり、あるとき、全体の通信量が急増してネットワークの許容限界を超えてしまったのである。

また、私たちの生活に大きな影響を及ぼし得るケースとしては、諸手当の改定法案の国会審議の遅れにより確定時期が前年度末となり、IT システムの対応が間に合わず、新制度による手当の交付開始が新年度から夏頃にまでずれ込む、といったようなことも想定される。

経済産業省の報告書［METI 2011］によれば、環境の変化に適切に対応するためには、「柔軟化」が求められるという。そして、柔軟化に対する次の 4 つの要求特性を示している。

- 予測性…可変要素（外部／内部環境）が将来変化をする予兆を事前に捉えること。
- 拡張性…既存のリソース（人、モノ、カネ、情報等）に将来の環境変化を想定した余裕を持たせておくこと。
- 迅速性…起きた変化／起こすべき変化に対して、すぐに対応出来ること。
- 適用性…これまでと違った環境、シチュエーションに、うまく対応出来ること。

柔軟化の対象は、必ずしも IT システムだけとは限らず、組織や業務プロセスなど、様々にあり得る。ただし、IT（情報通信技術）が生活やビジネスの中核を担う今日では、IT システムに帰着することが多い。

環境の変化は、IT システムに対しては、要求の変化として現れる（技術の進展は、制約の変化として現れるが、ここでは要求に含めて取り扱う）。要求の変化を想定し、IT システムに対しては、システム自身の拡張性と俊敏な構築・運用体制が求められることになる。これに応え、IT システムが要求の変化に柔軟に対応出来るためには、柔軟なアーキテクチャの採用と共に、俊敏なシステム構築及び運用の観点からは、とくにソフトウェアについて、次の 3 種の方策が考えられる。

a. 非ウォーターフォール型開発（アジャイル開発）
b. クラウドコンピューティング
c. 自動コード生成／ビジネスルールマネジメントシステム（BRMS）

本稿では、上記のうち「a. 非ウォーターフォール型開発（アジャイル開発）」に焦点を当てる。

2 アジャイル開発とは

IPA/SEC では、ウォーターフォール型以外の開発手法を「非ウォーターフォール型開発」と呼んでいる。その代表として、「アジャイル開発」について調査・検討を行ってきた。一言で「アジャイル開発」といっても、スクラム※2 や XP※3 など、様々な作法がある。ここでは、特定の作法に限定せず、後述の特徴を有する開発手法の総称として「アジャイル開発」という呼称を用いる。

アジャイル開発では、顧客の要求に従って、優先度の高い機能から順に、要求・開発・テスト（・リリース）を短い期間で繰り返しながら、システム全体を構築していく。原則として、事前に開発の詳細な計画は作らず、1 ～ 4 週間という一定の短い周期※4 で要求・開発・テストを繰り返しながら、動作可能なソフトを作り上げる。プロセスの観点による、ウォーターフォール型開発との比較を図 2 に示す。

このようなアジャイル開発は、次の特徴を有する。

- 顧客（ユーザ）の参画の度合いが強い。
- 動くソフトウェアを成長させながら作る。
- 反復・漸進型である。

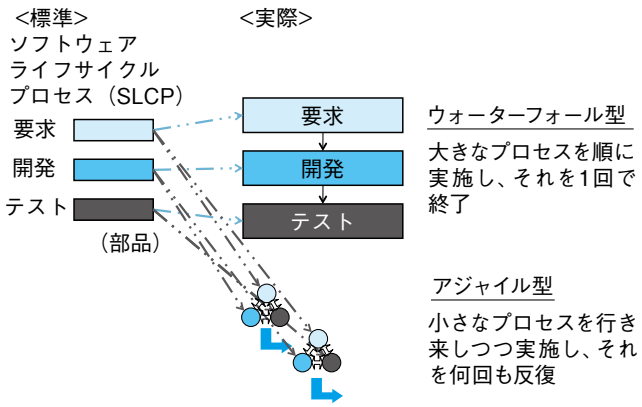

図2　開発プロセスの比較

脚注

※1　変化し得る環境には、ビジネス顧客や競合他社の動向、法制度や社会構造、技術等の外部環境と、事業再編等の内部環境がある。
※2　スクラム：現在では、アジャイル開発手法の中でも最も多く使われている手法。米国におけるある調査によれば、半分以上を占める。
※3　XP：Extreme Programming
※4　この周期を「反復（イテレーション）」と称する。

- 人と人のコミュニケーション、コラボレーションを重視する。
- 開発前の、要求の固定を前提としない。

IPA/SECでは、環境の変化に俊敏に対応可能なソフトウェア開発形態として注目度が増し、内外での成功例の報告も徐々に増えつつあったアジャイル開発について、平成21年度以降、調査・検討を通して整理を試みてきた。当時、アジャイル開発に注目した具体的な背景としては、次の3点が挙げられる。

① ビジネス・ニーズへの適切な対応

- 他社に先駆けた市場投入が必須で、それにより徐々に明確となるニーズを迅速に反映し改善していくことが必要な分野（Web系ビジネス等）の出現。
- 顧客ニーズは最初にすべては把握出来ず、またビジネス環境の激しい変化に伴いニーズも変化するが、この状況に迅速な対応が必要。

②（純粋な）ウォーターフォール型開発における問題点

- 初期には必ずしもすべての要求内容は確定しない。
- 誤要求や要求の誤解が総合テスト段階で判明すると、多大に影響。
- 開発途中で要求が変更されると、対応が非常に困難。

③ ソフトウェア産業構造（多重下請構造）上の課題

- 開発者（とくに若者）の参画意識・達成感が低い。

上記3点に加えて最近では、次の傾向が顕著である。

ⅰ）ビジネス環境の変化の俊敏さへの対応要求の増大
ⅱ）グローバル化の拡大
ⅲ）ウォーターフォール型開発に適合しにくいケースの増大

例えば、ある携帯電話会社の社長は最近、「まずは七分でよし。利用者のお叱りを受けながら100%に磨き上げていく。」というように、経営のスピード感重視を掲げている。また、「従来は100%にしてから世に出していた。現在はエンドユーザとの共同作業により、よいものにしていく。」といったβ版文化の普及が指摘されている。更には、ユーザ企業を対象とする最新の調査［JUAS2012］によれば、システム企画時に重視した事項として、品質が29%、コストが24%に対して、納期が47%という結果が得られている。

これらの背景に応え得るソフトウェア開発手法の一つが、アジャイル開発である。そして、いまやアジャイル開発は世界の主流となりつつあり、国内でもアジャイル開発に取り組む企業が増大傾向にある。しかしながら、あらゆるソフトウェア開発に対して、あるいはすべての組織において、アジャイル開発手法が単純に適用出来るとは限らない。

次章以降では、IPA/SECにおけるアジャイル開発に関する調査・検討の結果について、そのポイントを述べる。詳細については、公開している各報告書［報告書群］を参照願いたい。

3 日本におけるアジャイル開発の状況と課題

平成21年度に実施した調査において、我が国におけるアジャイル開発の事例22件を収集し、分析・整理した。その主な内容と明らかになった課題は、次の通りである。

［日本におけるアジャイル開発の現状］

- 開発チーム：約8割が8名以下。
- 開発期間：半数は2カ月～4カ月。
- アジャイル開発のプラクティス※5群の中から取捨選択して用いている。
- 内部開発が多い（自社内で利用するソフトを内製する、もしくは販売するためのパッケージを開発する）。
- 要求の変化への柔軟な対応、市場への投入の迅速化に効果が表れている。
- 大手SI業者でも、トライアルが行われ、社内標準への取り込みが始められている。

［日本におけるアジャイル開発の課題］

- 経営層の意識向上。
- スキルの明確化、人材の育成方法と適正配置。
- 日本におけるアジャイル開発に適した契約のあり方。
- 欧米の競争力（普及要因）の調査。
- 適用領域と適用事例の調査。

4 アジャイル開発普及上の課題の検討

前述の各課題について、アジャイル開発に関する経験の豊富な実務者及び有識者から成る非ウォーターフォール型開発WG（ワーキンググループ）を設置して検討すると共に、関連調査を行った。

4.1　アジャイル開発のプロセスモデル

平成 21 年度の調査により収集した国内におけるアジャイル開発の適用事例に基づき、まず、アジャイル開発のプロセスモデルを次の 3 種に整理した（図 3）。

モデル 1：企画終了後、要求・開発・テストの反復を繰り返しながら、適宜リリースする基本的な形態。

モデル 2：企画終了後、反復に入る前に、要求・アーキテクチャ設計・基盤開発をきちんと完了する、モデル 1 の拡張形態。基盤・共通部といくつかの機能部とから構成されるソフトウェアにおいて、まず、基盤・共通部をウォーターフォール型開発などにより完成させた後、機能部群についてアジャイル開発を行うもの。

モデル 3：何回かの反復の後、リリースする直前に重点的なテストを実施する、モデル 1 の拡張形態。顧客やビジネスの特徴から、とくに高い品質が求められたり、品質がクリティカルであったりする場合に、リリース前に一層の品質確保を行うもの。

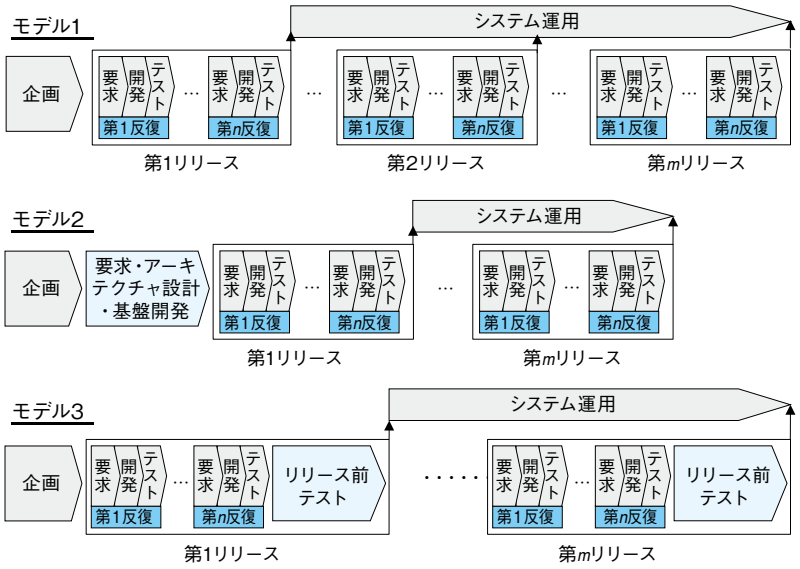

図3　アジャイル開発のプロセスモデル

4.2　アジャイル開発採用における経営層の理解

(1) 顧客（ユーザ）経営層

経営層は、ビジネス環境が激しく変化する現状において、IT システムに関し、従来のように情報システム部門に任せきりでは適切に対応出来ない。開発形態にも深く関与する必要がある。具体的には、次の責任を有する。

- 情報システムに関する理解の増進
- 迅速かつ適切な意思決定
- 関係部門との経営上の綿密な調整

また、アジャイル開発を進めるに際しては、経営層・情報システム部門共に、次の点に留意する必要がある。

- アジャイル開発の採用を決断した時点で、顧客がチームの一員として参画し、主体的に開発に関わらざるを得ないということに十分な理解と覚悟を持つ。
- アジャイル開発を誤解し、「最初にすべての要件を決めなくても良く、途中で適宜明確にしていけば良い」という安易な心構えで始めると、途中で破たんすることが多い。
- 開発を外注する場合には、ユーザ／ベンダ間の信頼感の醸成が、円滑に開発を進めるために重要。

(2) ベンダ経営層

俊敏な開発の実績を武器に受注を狙う海外勢などに対抗するには、自ら俊敏な開発を実施出来る体制作りに取り組むと共に、その結果を顧客に売り込む必要がある。

4.3　アジャイル開発に求められるスキル

アジャイル開発手法の代表であるスクラムにおいては、開発する製品の優先付けされた機能要求リスト「プロダクトバックログ」の中から、“プロダクトオーナー”が 1 つの反復で開発する分を取り出し、開発チームに指示する。開発チームは、“スクラムマスター”の管理・支援のもと、製品に要求機能を追加実装し、リリース可能な状態に作り上げる。その後、“プロダクトオーナー”

脚注

※5　アジャイル開発を実践する活動項目を“プラクティス”と呼ぶ。朝会、リファクタリング、継続的インテグレーション等、50 以上あるが、一つのプロジェクトですべてを使用するわけではない。

はリリース判定を行うと共に、次の周期の開発機能を指定し、このような反復を繰り返す。

従って、ユーザ参画の度合いが強いアジャイル開発において特徴的な、ユーザ側に求められるスキルは、投資対効果を最大とするためにコアとなる機能を見定め、定期的なサイクルで優先度を考慮しながら開発プロジェクトの運営を指揮していく“プロダクトオーナー”としての能力ということになる。

また、アジャイル開発の開発者側にとって重要なスキルは、次の通りである。

① “スクラムマスター”として、開発チームの自律的・協働的な作業を支援し、アジャイル開発の進め方を踏襲させるためのファシリテーションスキル。

② チームメンバとして、反復活動の中で、実際に動くものを作りながら、小規模に、かつトータルにプロジェクトのアウトプットを積み重ねていくスキル。

③ 設計、コーディング、テストを一貫して実施出来る、マルチタレントなスキル。

4.4　アジャイル開発のための人材育成と動機付け

(1) 人材育成

前節で示したスキルを身に付けるためにプラクティスを習得するわけであるが、そのための人材育成においては、次のようなアジャイル開発の実情を考慮する。

- 一つのプロジェクトですべてのプラクティスを使うわけではない。
- 各プラクティスに厳格な規範はない。
- 様々な方法論・数あるプラクティスから、プロジェクトや組織に適したものを取捨選択し、カスタマイズすることが必要。

すなわち、平時における一通りのプラクティスを理解するための施策と、プロジェクト参画時における使用プラクティスの深い習得のための施策とを使い分ける。

ただし、すべてのプラクティスを完全に身に付けるより、価値に従って行動する習慣を確実に身につけることが重要であり、その結果として適切なカスタマイズに結びつく。

なお、人材育成に際しては、次の点に留意する。

- 自社システムの開発から、アジャイル開発の OJT を行う。
- 導入初期の実プロジェクトでは、コンサルタント（アジャイル・コーチ等）の支援を得る。
- 個人プラクティスは普段の業務中に、チーム・プラクティスは組織的な研修などで、それぞれ習得する。

とはいえ、アジャイル開発に向いている技術者とウォーターフォール型開発に向いている技術者とが存在することは確かであり、人材の見極めと適切な配置が、個人と組織の双方にとって重要である。

(2) モチベーション

米国の著名なビジネスジャーナリスト・作家であるダニエル・ピンク［DANIEL］によれば、報酬のインセンティブは、視野を狭め、心を集中させることから、単純な仕事では効果があるが、（ソフトウェア開発のような）そうでない創造的な仕事では逆効果であるという。そして、成果を高めるのは、内的な動機付けに基づくアプローチ、すなわち、重要だからやる、好きだからやる、面白いからやる、何か重要なことの一部を担っているからやる、というものであるとのこと。更に、仕事において重要な要素は次の 3 つと述べている。

- **自主性**…自分の人生の方向は自分で決めたい
- **成　長**…何か大切なことについて上達したい
- **目　的**…自身よりも大きな何かのためにやりたい

これらをアジャイル開発に照らすと、「自主性→ある程度の裁量」、「成長→スキルアップになる」、「目的→製品の完成ではなく顧客の“価値”を高める」、ということになる。

このような動機付けが、アジャイル開発に携わる技術者にとって重要であるといわれている。

4.5　アジャイル開発に適した契約形態

日本におけるソフトウェア開発では、ユーザ／ベンダ間で受発注契約を締結することが多い。アジャイル開発には、ウォーターフォール型開発と比較し、次の特徴がある。

① ユーザとベンダの緊密な協力体制が必須。

- 相手方の問い合わせへの迅速な応答。
- 担当作業の迅速な実施。
- ユーザ／ベンダ間の責任分担が不明確になりがち。

② ユーザ要求の詳細が契約時点では未確定。

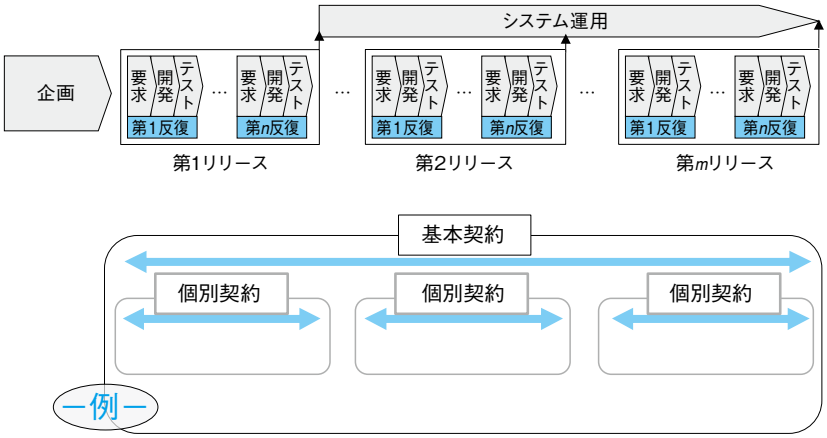

図4　基本／個別契約モデルのイメージ

- 何を作るか決まっていない（成果物未定）。
- 性能・品質などが不明確。
- 工数見積りが困難（コスト未定）。

③ 開発途中でのユーザ要求の変化を柔軟に受け入れる。

- 決定した事項も変更されることがある。

すなわち、そもそも「契約」とは、合意内容を固定して、当事者を法的に拘束するものであるのに対し、アジャイル開発は、変化に対応すべく、合意内容の変更を柔軟に認め、当事者をなるべく拘束しないという特徴を有することから、「契約」とは相容れないことになる。

従って、開発内容が決まっていない段階で、開発プロジェクト全体につき、一つの請負契約を結ぶのは適切ではない（何をいくらで完成させるか不明なため）。他方、開発プロジェクト全体を準委任契約にすることは、ベンダが完成義務を負わない点で、ユーザ側に不安がある（たとえ成果物が完成しなくても、ユーザは対価を支払う必要があるため）。また、アジャイル開発の特徴であるユーザとベンダの協働関係を、契約に取り入れる必要がある。

このような状況に対し、日本におけるアジャイル開発にふさわしい契約モデルとして、次の２つを提案した。

- 基本／個別契約モデル：プロジェクト全体に共通する事項につき、基本契約を締結する。また、小さな機能単位ごとに、開発対象と費用がある程度確定したタイミングで個別契約（請負／準委任）を順次締結する（図4）。
- 組合モデル：ユーザとベンダが共同でジョイント・ベンチャーとしての組合を組成し、協力してシステム開発を企画・実施する。開発された成果から得られた収益は、ベンダとユーザに分配される。

なお、ユーザ側での労働者派遣契約に基づく開発チーム構成には、契約上の問題はとくにない。従って、最近高まる傾向にある内製には容易に採用出来る。

4.6　アジャイル開発の普及に向けた課題と解決方法

ソフトウェア開発プロジェクト、IT人材の状況、IT人材育成の３点について、米国・英国・デンマーク・中国・ブラジルの５カ国と日本とを比較調査し、欧米の競争力の源となっているアジャイル開発の普及要因を、駆動要因と土壌に分けて分析した。その結果、次の事項が明らかとなった。

① 最も有名なスクラムに関する資格者（取得試験は英語）は、米国の取得者が75,000人以上と群を抜いて多く、ついで13,000人強の英国が多い。日本は500人未満と極めて少ない（図5）。

② 米国では、ソフトウェアに対する投資において、外注は約1/3を占めるに過ぎない。更に、他国に比べて多くのIT技術者がユーザ企業に所属している（図6）。これを反映し、米国のプロジェクトの形態の特徴としては、37%が内製である。

③ デンマークでは、政府が発注するソフトウェア開発

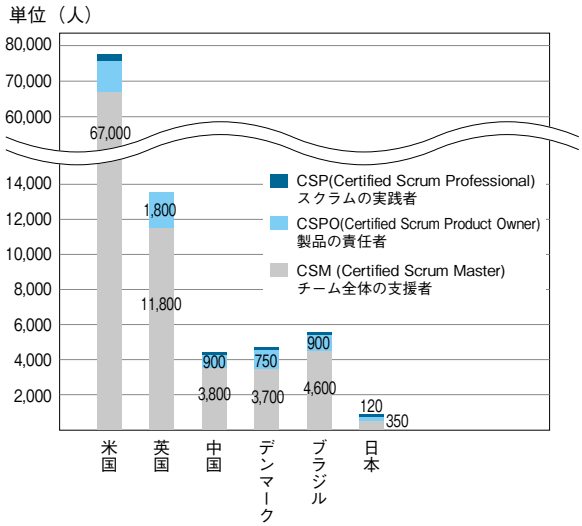

図5　各国のスクラムマスターなどの人数（2012年3月現在）

をアジャイル開発で実施することを推奨している。

④ ブラジルでは、実践した結果、顧客のビジネスの成功率が高いことが要因となり、アジャイル開発が普及している。また､時差が少なく､リアルタイムコミュニケーションが可能であることから、北米からの受託開発が伸びている。

⑤ 米国では、恒常的にIT関連職の給与が高く、人気も高い。中国、ブラジルでは、IT関連職の人材が不足しているため、IT関連職の給与が高く、人気も高い。これらの国々では、IT関連職の人気が高いため、優秀な人材が集まる。

以上の結果を受け、SECではいくつかの施策について現在検討中である。そのうち、プラクティス事例に関するリファレンスガイドの公開を、今年度末に予定している。

5 アジャイル開発の適用領域

すべてのソフトウェア開発にアジャイル開発手法を適用すべきという訳ではない。開発対象の特徴や組織プロジェクトの状況に応じて、適用すべきか適用すべきでないかを判断する必要がある。

(1) 当初の適用領域

アジャイル開発が得意とし、現在その適用により効果を挙げている領域は次の通りである。

① ビジネス要求が変化する領域

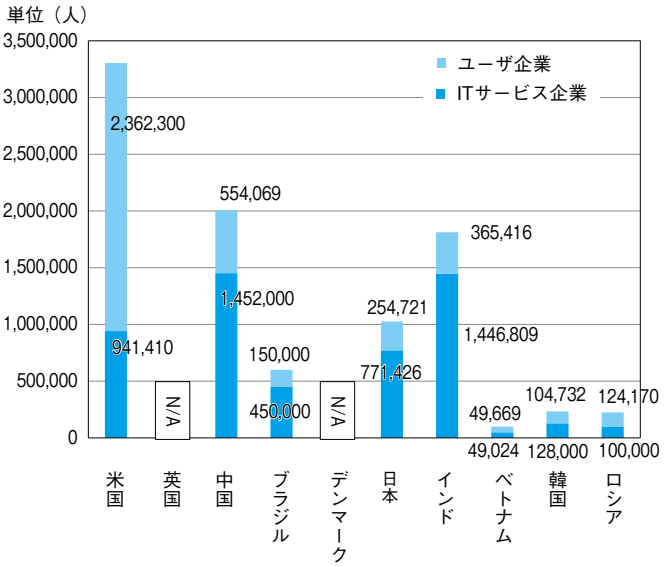

図6　IT技術者の所属先の国別比較

- 要求の変化が激しく、あらかじめ要求が固定出来ない領域。

② リスクの高い領域

- 不確実な市場を対象としたビジネス領域（市場リスク）。
- 技術的な難易度が高い開発領域（技術リスク）。

③ 市場競争領域

- 他社に先駆けた製品・サービスの市場投入が命題であり、TTM（Time to Market）の短縮が優先となる領域（Webサービス、パッケージ開発、新製品開発）。

(2) 当初の試行領域

一方、アジャイル開発による経験が十分には蓄積されておらず、現在、チャレンジと創意工夫が求められている領域もある。

① 大規模開発

コミュニケーション・コラボレーションを重視するアジャイル開発では、開発者が10人程度を超えると、システム分割、チーム分割が必要となる。その分割方法、及び分割されたチーム間のコミュニケーションが課題である。

② 分散拠点（オフショアを含む）開発

①と同様の理由により、開発拠点が分散し、更に時差によって分断される場合のコミュニケーション手法、また、それをサポートするツールが必要である。

③ 組織（会社）間をまたぐ開発チームによる開発

①②と同様の理由により、共通のビジネスゴールを持ったチームを組むことが難しい。

④ 組込みシステム開発

一般に、組込みシステムではリリース後のソフトウェア修正が極めて困難であり、頻繁なリリースを繰り返すアジャイル開発の採用には工夫が必要である。

(3) 中・大規模開発への適用事例

近年、国内での中・大規模開発へのアジャイル開発手法の適用事例が増えてきている。少人数での開発を前提としたアジャイル開発に対し、人数が増えた場合にどのような問題が発生し、その解決のためにどのような工夫がされているかについて、10件の調査事例をもとにまとめた（表1）。その主なものを以下に説明する。

●チーム間ローテーション

チーム間の知識伝播促進のため、メンバのローテーションを行う。一時的に開発速度は落ちるが、各チーム

の知識を効率よく伝播することが出来、技術者のマルチタレント化が高まる。

●段階的朝会

複数チーム間では朝会を段階的に実施する。

チーム→全体（→チーム）

●漸進的な展開

新しい施策は、一度にすべてのチームに広げるのではなく、まずは導入障壁の低いところ、最も必要なところから順次導入し、少しずつ展開する。振り返りで、次はどこに広げていけば良いかを考える。

●コミュニケーション・ツールの活用

TV 会議システム、雑記帳システム（SNS、Blog など）を活用し、コミュニケーションの円滑化を図る。

●アーキテクチャの重視

プロジェクト前半にアーキテクチャを構築する事例が多く、アーキテクチャ専門チームを編成して構築する。

●疎結合な機能分割

疎結合な機能でサブシステム分割を行う。7 割のチームが CI（継続的インテグレーション）を実施している。

●テスト専用フェーズ

プロジェクト後半で専用のテストフェーズを実施する。プログラム開発の反復を停止する事例と、テストのみの反復期間を設ける事例がある。

一方、次のような問題点も明らかになり、更なる工夫が必要である。

●全体計画の把握困難

要求の変化や開発状況に応じて着手する順番や範囲を決めるため、プロジェクト開始時にプロジェクト全体の把握が困難な事例がある。

●ビジネス企画側にボトルネック発生

スクラム導入の結果、ビジネス企画者の決定待ちなどのボトルネックが発生した事例が多い。中には開発者の信頼をなくした事例もある。

●反復リズムとの不適合状態の発生

セキュリティ監査や外部テスト業者、発注者の外部組織や関連組織との関係において、開発の反復リズムと適合せずに問題が発生している事例がある。

6 イノベーションに向けて

環境の変化に俊敏に対応出来、私たちが安心して使い続けられる IT システムの開発手法の一つとして注目されているアジャイル開発手法について、IPA/SEC にお

表1　中･大規模開発への適用時の工夫項目

中・大規模開発特有の工夫		小規模開発と同様だが とくに注意して実施する工夫
■組織体制 ●チーム間ローテーション ■コミュニケーション ●段階的朝会 ●チーム跨ぎの振り返り ■展開 ●漸進的な人数増加 ●漸進的な展開 ●社内勉強会 ■分散拠点開発 ●同一拠点から分散へ ●TV会議 ■アーキテクチャ ●組織の共通基盤アーキテクチャの利用 ●アーキテクチャについての教育 ■システム分割／インテグレーション ●同じリズム	■品質 ●第三者テスト ■部分適用 ●必要な部分のみ適用 ●疎結合なチーム ●工程の見える化 小規模開発とは逆のアプローチを取る工夫 ■アーキテクチャ ●最初のアーキテクチャ構築 ●アーキテクチャ専門チーム ●運用保守チーム ■品質 ●テスト･フェーズ	■コミュニケーション ●完全透明性 ■展開 ●パイロット導入 ●認定研修･コンサルタントの利用 ■分散拠点開発 ●チケットシステム ●リアルタイムチャット ■アーキテクチャ ●アーキテクチャの改善 ■システム分割／インテグレーション ●疎結合で分割 ●早期からのインテグレーション ●継続的インテグレーション ■品質 ●重視するビジネス価値 ●ビジネス価値の変化 ●タイムボックス優先の品質 ●自動単体テスト

ける約４年間の調査・検討の結果をまとめた。

普通のビルや橋を建設する場合には、作るものも使用する技術も明確である。しかし、ソフトウェアの中には、計画時にビジネス上やシステム上の課題が未解決で、開発開始後も仕様変更の可能性が大きいものが少なくない。このような場合には、最初から綿密な計画を立てるより、少し試して、その結果に基づいて次のステップを進めるというやり方のほうが、失敗のリスクが小さい。このような考え方が、アジャイル開発を推進する理由の一つとなっている。

アジャイル開発手法を導入する際のポイントは、次のようにまとめられる。

① 適切な開発手法の選択

開発対象の特徴や開発組織の置かれた環境などを加味しつつ、適切な開発手法を選択するか新たに考案する。

② プラクティスの活用

各プロジェクト・組織（企業）で、自らの開発に合った方法の採用に向け、プラクティスを選択あるいは参考にして利用する。必要に応じ、カスタマイズする。

③ 開発手法に対する正しい理解の促進

プラクティスの意図やプラクティスが提唱されている背景についても理解を深める。

ただし、「銀の弾丸は無い」ということを肝に銘じておく必要がある。すなわち、実践現場でのたゆまない問題解決の積み重ねを続けることが最も重要である。

ウォーターフォール型開発が「プロセス」重視の手法であるのに対し、アジャイル開発は「人」重視の手法であるといわれる［CONBOY2011］。別の観点からは、ウォーターフォール型は開発が失敗しないための手法であり、アジャイル開発はビジネスが成功するための手法であるといえなくもない。すなわち、両手法は全く文化が異なるといえる。比較的早くからアジャイル開発手法を導入してきた大手企業も、「多くの組織、チーム、個人にとって、アジャイル開発プロセスへの転換は“挑戦的”である。それは、ある種の文化的変革を必要とするからだ。」と述べている［IBM］。

しかしながら、この転換に挑戦する企業も現れている。例えば、ある大手の Web 系企業では、高品質を追求する日本的なソフトウェア開発手法と米国流の俊敏なアジャイル開発手法との『いいとこ取り』をし、高品質はそのままに開発速度を速める工夫を行っている［日経 SYSTEMS2012］。また、ウォーターフォール型開発とアジャイル開発とのハイブリッド形態の採用も増加傾向にある［VERSIONONE2011］。

各開発手法を得意とする技術者が協働し、両文化をうまく融合させることにより、ソフトウェア開発のイノベーションを期待したい。

最後に、非ウォーターフォール型開発 WG 委員をはじめとする、アジャイル開発に関する調査・検討にご協力いただいた方々に深く感謝する。

参考文献

［CONBOY2011］Kieran Conboy, Sharon Coyle, Xiaofeng Wang, Minna Pikkarainen：“People Over Process: Key People Challenges in Agile Development”, IEEE Software, July 2010
［DANIEL］ダニエル・ピンク：やる気に関する驚きの科学，2009,
http://www.ted.com/talks/lang/ja/dan_pink_on_motivation.html
［IBM］Agile transformation,
http://www.ibm.com/smarterplanet/us/en/business_analytics/article/agiledevelopment.html
［JUAS2012］一般社団法人 日本情報システム・ユーザー協会 (JUAS)：ソフトウェアメトリックス調査 2012，2012
［METI2011］社団法人日本情報システム・ユーザー協会：平成 22 年度経済産業省委託調査「IT 経営普及促進に向けた調査研究」報告書，p.76, 2011 年２月,
http://www.meti.go.jp/meti_lib/report/2011fy/0022948.pdf
［VERSIONONE2011］VERSIONONE：State of Agile Development Survey Results，2011
［日経 SYSTEMS2012］特集１　ここがヘンだよ　日本のシステム開発 - 日本と米国のいいとこ取り，日経 SYSTEMS，2012 年８月号，2012
［古川 2011］古川昌幸：「Gen-Y」世代が主力ユーザーとなる時の IT，野村総研，知的資産創造，2011 年３月号，pp.32-45，2011,
http://www.nri.co.jp/opinion/chitekishisan/2011/pdf/cs20110305.pdf
［報告書群］
・H21 年度版報告書,
http://sec.ipa.go.jp/reports/20100330a.html
・H22 年度版報告書,
http://sec.ipa.go.jp/reports/20110407.html
・H23 年度版報告書,
http://sec.ipa.go.jp/reports/20120326.html
・ウォーターフォール型開発の普及要因と適用領域の拡大に関する調査報告書（国内の中・大規模プロジェクト事例),
http://sec.ipa.go.jp/reports/20120328.html
・非ウォーターフォール型開発の普及要因と適用領域の拡大に関する調査報告書（非ウォーターフォール型開発の海外における普及要因編),
http://sec.ipa.go.jp/reports/20120611.html