

「米国におけるアジャイル・ソフトウェア開発の動向」

渡辺弘美@JETRO/IPA NY

1. 新たなソフトウェア開発手法 — アジャイル・ソフトウェア開発

ソフトウェア開発の新たな手法として、アジャイル・プログラミングに注目が集まっている。アジャイル開発を導入することによって、従来の開発よりも、早く、安く、しかも品質の高いソフトウェアを開発することが狙いである。

『Agile and Iterative Development: A Manager's Guide』の中で著者の Craig Larman はアジャイル開発を次のように定義している。

アジャイル開発 (agile development) 手法は、適応型の計画を行い、タイムボックスを適用しながら反復型・進化型の開発を行い、段階的に出荷するものであり、そのほかにも「機敏性 (アジリティ)」(迅速かつ柔軟に変化に対応すること) を高めるための価値やプラクティスが含まれる。アジャイル手法にモットーがあるとしたら、それは「変化を受け入れる」ことである。アジャイル手法に戦略上重要な点があるとしたら、それは「機敏性」である。

(1) 変化を受け入れる

Larman がアジャイル手法のモットーとして指摘するように、アジャイル開発と従来のソフトウェア開発の大きな違いとして強調されるのが、ソフトウェア開発の予測可能性に対する見解である。従来型の開発では、変化は予測可能であるという前提に立ち、開発を始める前に要求仕様を固め、それに忠実に沿って開発を進めていくというコンセプトが基本となっている。こうしたアプローチについて、『Balancing Agility and Discipline: A Guide for the Perplexed』の中で Barry Boehm と Richard Turner は、従来型のソフトウェア開発の根幹をなしていた考え方の代表としてソフトウェア能力成熟度モデル(SW-CMM)を挙げ、こうした従来のソフトウェア開発は、変化を予想し、変化がシステムに悪影響を及ぼす前に飼いならして管理するためのアーキテクチャを無理やり構築し、変わりつつある環境に対処するというものであり、ソフトウェア成果物の品質とプロセスの予測可能性を重視すると説明している。

一方のアジャイル開発では、開発過程で予測できない変化が起こるのは避けられないとの前提に立っている。『Agile Software Development: Principles, Patterns, and Practices』の中で Robert C. Martin は、ビジネス状況の変化は常に変化しており、それに併せて関連するビジネス関連のシステムに対する要求は変わって当然であるばかりでなく、一般的に顧客というものはシステムが動いていると要求を変更したがるものだという点を指摘している。

これを裏付けるように、典型的なソフトウェア開発プロジェクトでは変更される要求が25%にも及ぶという数字を1988年10月にIEEE Transactions on Software Engineeringの「Understanding and Controlling Software Costs」の中でBoehmとPapaccioが発表している。

① 短いリリース・サイクルを何度も繰り返す反復型

こうした変化に柔軟に対応するアプローチを実現するにあたり、アジャイル開発手法では開発周期を短くし、何度もリリースを繰り返し行う反復型開発を採用した。この場合、リリースのたびに顧客の要求変化を受け入れる代わりに、将来必要になりそうな機能まで予め計画して盛り込もうとする従来型のやり方は否定し、個々の開発サイクル毎に最も優先度の高い機能だけを実装することを重視している。頻繁に変わる顧客の要望を受け入れるために、アジャイル開発では顧客と開発者の緊密な関係構築が欠かせない。多くのアジャイル開発では、顧客の環境で試験を繰り返すたびに顧客からのフィードバックや要求変更を受け入れ、開発者は顧客とともにソフトウェアをリリースごとに改良していく手法をとっている。こうしたアプローチはクライアント駆動型反復開発（client-driven iterative development）としても知られ、顧客が率先して自社のビジネスにとって最優先すべき価値の高いものを自ら見極め、それに従って開発を進められていくとLarmanは指摘している。

② 簡潔で要を得たドキュメンテーション

加えて、柔軟性の高いソフトウェア開発を実現するために、アジャイル開発ではドキュメント作成稼働を出来る限り減らすというやり方を重視している。従来、ソフトウェア開発におけるドキュメント作成は非常に重視されてきた。もちろんアジャイル開発でもドキュメント作成を不要としているわけではない。しかし、変化を受け入れるという姿勢に従来どおりの文書作成を必要としたら、ドキュメンテーションに時間がかかり、変化を受け入れる機敏な体制を構築することはできなくなる。そうしたことから、アジャイル開発におけるドキュメント作成の要点を「ドキュメントは簡潔で、要を得たものでなければならない」とMartinは指摘している。12～24ページ程度に、システム設計に関する記述は包括的な論理的説明のみにとどめ、システムの構造にはトップ・レベルについてのみ記述するとしている。

一方、「簡潔で、要を得た」ドキュメンテーションを行うことによって、新たな開発者などがグループに加わった場合、どうするのかという問題がある。この問題に対してアジャイル開発手法では、開発グループ内でのコミュニケーションを非常に重視している。例えばアジャイル開発手法の1つであるScrumを採用

した開発では、開発者は毎日 30 分以下のミーティング時間を必須とし、その中で進捗状況・目標についての確認をし、開発グループ内のコミュニケーションを図っている。

(2) アジャイル宣言

こうした、アジャイル・ソフトウェア開発の特徴が、Agile Alliance が 2001 年に発表したアジャイル宣言の中で基本理念としてまとめられている。Agile Alliance はアジャイル宣言を実現するための 12 の原則も同団体のホームページで提案している。

アジャイル宣言
(Manifesto for Agile Software Development)

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

プロセスやツールよりも、**個人間の交流を大切にせよ**
包括的なドキュメント作成に力を注ぐより、**うまく動作するソフトウェア開発に努力せよ**
契約の交渉よりも、**顧客とのコラボレーションを重視せよ**
計画に従うよりも、**変化に柔軟に対応することを心がけよ**

2. 様々なアジャイル・ソフトウェア開発手法

現在、様々なアジャイル手法が開発されている。最も注目され有名な開発手法は Extreme Programming (XP) である。XP を含む様々なアジャイル開発手法の例を以下にまとめる。

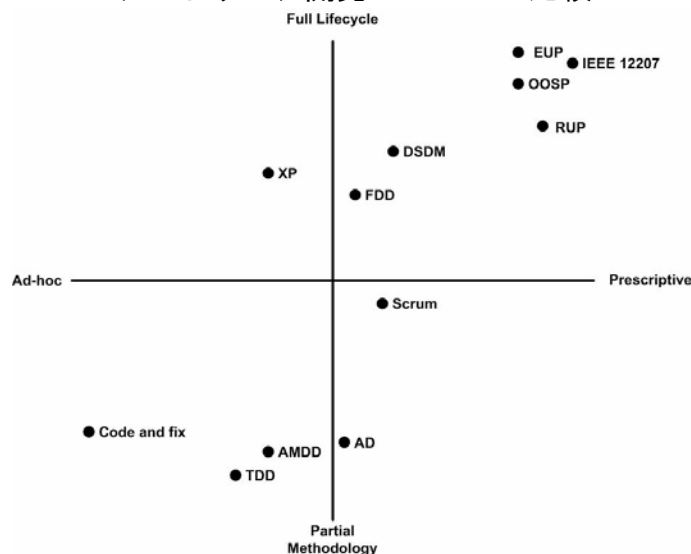
主なアジャイル・ソフトウェア開発手法の概要

開発手法	概要	開発規模	創始者
Adaptive Software Development (ASD)	「迅速なビジネス開発の必要性に対応するための哲学的基盤と実践的アプローチを提供する。反復型（イテ	これといって特に最適な開発人数はないが、ほぼどの	Jim Highsmith

	<p>レーティブ) 開発、ユーザー機能ベースの計画、顧客のフォーカスグループレビューなどを、リーダーシップ・コラボレーションマネジメントスタイルで実施する。」</p> <p>「目の前の仕事に適応させ続けることこそが正常である」という原則に基づく。</p>	<p>サイズにも対応。</p>	
Crystal Family	<p>「チームの規模やプロジェクトの重要度に合わせて『儀式』度のレベルを使い分けられる手法群。プラクティスは、アジャイル手法や計画駆動型手法からだけでなく、心理学や組織開発の研究からも取り入れられている。」</p> <p>同手法には数多くのバリエーションがある。開発プロジェクトの規模や重要度に応じて、手法を使い分けるようになっている。</p>	<p>中小規模の開発に適している。</p>	<p>Alistair Cockburn</p>
Dynamic System Development Method (DSDM)	<p>ISO9001 資格を取得したアジャイル開発手法。もともとは 1980 年代に開発された Rapid Application Development (RAD) メソッドの開発・普及を目的として始められ、DSDM に発展してきた。</p> <p>プロセス・マネージメントを重視した手法となっている。</p>	<p>あらゆる規模の開発(個人単位～ビジネスシステムまで)に適応している。</p> <p>複雑なビジネス・アプリケーション開発に適している。</p>	<p>DSDM コンソーシアム(欧州ユーザーが中心)</p>
Enterprise Unified Process (EUP)	<p>開発オペレーションやソフトウェア・システムの利用停止までを含む、厳格なソフトウェア・プロセスに関する7つのフェーズをもった開発手法。イテレーティブでインクリメンタルな開発手法をとる。エンタープライズ・アーキテクチャ、再利用管理(reuse management)、ポートフォリオ・マネージメント、及び人的資源管理などを含む。</p>	<p>—</p>	<p>—</p>
Extreme Programming (XP)	<p>最も有名なアジャイル開発手法。</p> <p>『ダイムラー・クライスラー社の情報システムを開発した経験を洗練して作られたものである。XP はきわめて厳格で、最初</p>	<p>小規模(4-10 人程度)の開発</p> <p>単一チームでは 20 人が限界とされている</p>	<p>Kent Beck, Ward Cunningham</p>

	<p>のうちは手法で定義されたすべてのプラクティスに従うよう求めている。』</p> <p>4つの基礎として「コミュニケーション」「シンプルさ」「フィードバック」「勇気」が挙げられている。このほかに12の基本プラクティスがある。</p>	る。	
Feature-Driven Development (FDD)	<p>非常に軽量なアーキテクチャを重視。系統だったモデリングを含む、短期間のイテレーションを基礎としたアジャイル開発手法。UML やその他のオブジェクト指向設計手法を使うことが前提となっている。</p>	小規模プロジェクト (4-20 人)	Peter Coad
Rational Unified Process (RUP)	<p>厳格な4つのフェーズ(方向付け、推敲、作成、移行)をもつソフトウェア開発プロセス。イテラティブでインクリメンタルな開発手法。</p>	中規模以上 (10 人以上)	Rational Software 社
Scrum	<p>明確なプロジェクト・マネージメントと要求管理(Requirements Management)を重視するアジャイル開発手法。しばしば XP と組み合わせて利用される。</p> <p>Scrum 開発では、毎日行われる 30 分以内のミーティングによる進捗状況・目標についての確認をし、開発グループ内のコミュニケーションを図っている。</p> <p>基本的開発サイクルは 30 日。</p>	あらゆるサイズのプロジェクトに適応可。	Ken Schwaber 他

ソフトウェア開発プロセスの比較



3. 連邦政府におけるアジャイル開発動向

このようなアジャイル・ソフトウェア開発による機敏性（アジリティ）の高い開発手法は、民間のソフトウェア開発のみならず、政府機関が利用するソフトウェア開発でも積極的に取り入れられている。

(1) ウォーターフォール型からインクリメンタルな反復型へ

インクリメンタルな反復型開発（iterative and incremental development = IID）の先端例ともいえるアジャイル開発が政府で取り入れられる背景には、ウォーターフォール型からの転換時期を経て、IIDが政府関連のソフトウェア開発に受け入れられる体制が整えられてきたという側面があることを見落とすべきではないだろう。

これまでの経緯については、Craig Larmanの解説によれば、1980年代、国防総省（DOD）はソフトウェア開発及び調達に関する標準として、DOD-STD-2167を発表した。これはウォーターフォール・モデルと文書駆動型を核とする開発を推奨したものであった。しかし、これによって失敗が相次ぎ、2167に準拠した開発の約75%が失敗もしくは使われずに終わったということが1995年にDODから発表されている。

2167を改善する必要に迫られ、IIDに切り替える方向性を示す2167Aが出された。しかし、ここでもウォーターフォール型のコンセプトを示す図が残されていたことから、1994年12月に更なる改訂版としてMIL-STD-498が出され、ウォーターフォール型を使わなくてはならないという先入観を取り除くための努力が払われてきた。2000年にはDOD 5000.2という指導文書を発表し、進化型及びスパイラル型による開発手法を強く推奨する内容となっている。また、DODの動きに対応して、2002年には米国食品医薬品局（FDA）は、医療機器に関する推奨するソフトウェア開発手法を、それまでのウォーターフォール型からIIDに変更している。

このようなソフトウェア開発手法としてIIDを推奨しようとする米国連邦政府の方向転換がアジャイル開発導入を可能にする基盤を整えてきたと見ることが出来る。以下では、ここ数年で実際にアジャイル・ソフトウェア開発手法を用いて行われた事例について、いくつか紹介する。

(2) 各政府機関の取り組み

① エネルギー省 (DOE)

DOE はコロラド州にあった核兵器製造工場の Rocky Flats Environmental Technology Site の閉鎖を決定した。閉鎖により、全ての核物質と核廃棄物は自然環境の中でもともこの物質があった場所に戻すというプロジェクトが計画され、Kaiser-Hill Company, LLC が DOE との間でこのプロジェクトを実施する 110 億ドル契約を結んだ。それに関連するソフトウェア開発を CH2M HILL (コロラド州) が請け負うこととなり、その開発手法としてアジャイル開発手法の XP が採用された。

政府プロジェクトに関連するソフトウェア開発は、ドキュメンテーション作成に対する要求が厳しく、ソフトウェア成熟度モデルの CMMI に沿った開発を要求される。一方、アジャイル開発は前述の通り、文書作成を出来る限り削減することによって、機敏性を高めることを目指しており、CMMI とは相容れない部分がある。CH2M HILL は XP と CMMI の中庸を探り、その課題を解決。アジャイル開発の政府関連ソフトウェア導入への方向性を示すことに成功している。

開発に参加した CH2M HILL の Glen B. Alleman と Michael Henderson は 2003 年 6 月にソルトレーク・シティで開催されたアジャイル・ソフトウェア開発に関するカンファレンスの中で、XP の 12 の基本プラクティスを CMMI と連動させた方法について以下のようにまとめて紹介している。

政府関連ソフトウェア開発環境への XP プラクティスの適用方法

XP Practice	Our Implementation
Planning Game	Biweekly planning sessions.
Small Releases	Biweekly iteration releases with full integration with Configuration Management and IV&V.
Metaphor	Not used.
Simple Design	Forced on the team by the "time boxed" iteration process.
Refactoring	Not a major impact as yet.
Testing	Unit tests, integration tests, IV&V testing.
Pair Programming	Not usually allowed because of code access security requirements. Some PP within small groups.
Collective Ownership	Not usually allowed because of code access security.
Continuous Integration	Heavy investment in tools, but some progress for daily builds.
40 Hour Week	Mandated by our contract.
On-site Customer	Customer Service Managers (CRMs) with detailed process knowledge.
Coding Standards	Inherited from CMMI processes.

② 米陸軍 (U.S. Army)

<U.S. Army Environmental Center>

U.S. Army Environmental Center (USAEC) では、アジャイル・ソフトウェア開発手法を採用したソフトウェア開発プロジェクトを推進している。同機関のシステム部門では、陸軍基地の環境クリーンアップ関連ミッションに使われるソフトウェア開発に複数のアジャイル開発手法を組み合わせて利用していると、USAECのシニア・システム・アナリストである Dave Garrett は述べている。

USAEC はアジャイル開発を適用することによって、ソフトウェア開発のコスト削減効果や効率性向上の実績を挙げている。はじめて USAEC で行ったアジャイル開発は、ある会計システムへの機能追加を目的とした 25 万ドルのプロジェクトであった。アジャイルを採用した結果、1ヶ月近く納期を短縮したばかりでなく、半年で約 5 万ドルのコスト削減を実現した。この効果を目の当たりにして、Garrett の開発チームではアジャイルを使った次なるプロジェクトにすでに着手するとともに、将来、同機関における標準的開発メソッドとして確立させていくことも狙っている。

Garrett の開発チームは、厳しさを増すコスト削減要求に加え、ソフトウェアのリリース・スピードをアップしなければならないという課題に直面していた。そうした課題を解決するにあたり、同チームでは特に XP プラクティスの 1 つであるペア・プログラミング (Pair Programming) に注目した。

ペア・プログラミングというのは、Larman によれば次のように定義されている。

『アプリケーションコードは必ず、二人のプログラマーが 1 台のコンピュータに向かって作成する。定期的に交代しながら交互にコーディングする。ペアはタスクが変われば頻繁に組みなおされる。見ている側の方は、リアルタイムのコードレビューを行っていることになり、おそらくは入力している人よりも幅広い考え方ができる (テストなどについて考えられる) 』

XP では、ペアでプログラミングを行うことで、「メンバー相互の学習が可能になり」、「規律を守ったプラクティスを守ろうとする意思が働き」、「お互いに効率よく働こうというピア・プレッシャーがかかる」という 3 つの効用がもたらされるとされている。単純に考えれば 2 倍のプログラマーが必要という計算になるが、Larman の定義にあるプログラミングをしながら同時にレビュー可能といった要素から、実際にはプログラマーのコストは他の開発手法の 15% 増にしかないという研究が Alistair Cockburn と Laurie Williams から発表されている。

XPのペア・プログラミングに加え、USAECでは、アジャイル手法と Computer-Aided Software Engineering System といった自動ソフトウェア開発ツールと組み合わせて利用するといった試みも導入している。

<U.S. Army Communications Electronics Command (CECOM) Software Engineering Center>

上記の他に、米陸軍ではアジャイル・ソフトウェア開発手法を用いたソフトウェア開発を成功させている。米陸軍の戦闘用システム（Army Battle Command System）の一部である Maneuver Control System (MCS) Light は、アジャイル手法を部分的に取り入れて成功させた陸軍のソフトウェア開発事例である。MCS は戦闘計画を立てたり、戦闘状況をモニタリングする機能を司令官向けに提供するシステムである。MCS Light への評価は高く、現在 10 ある陸軍師団のうち 9 つの師団で、戦闘計画ツールとして利用されている。

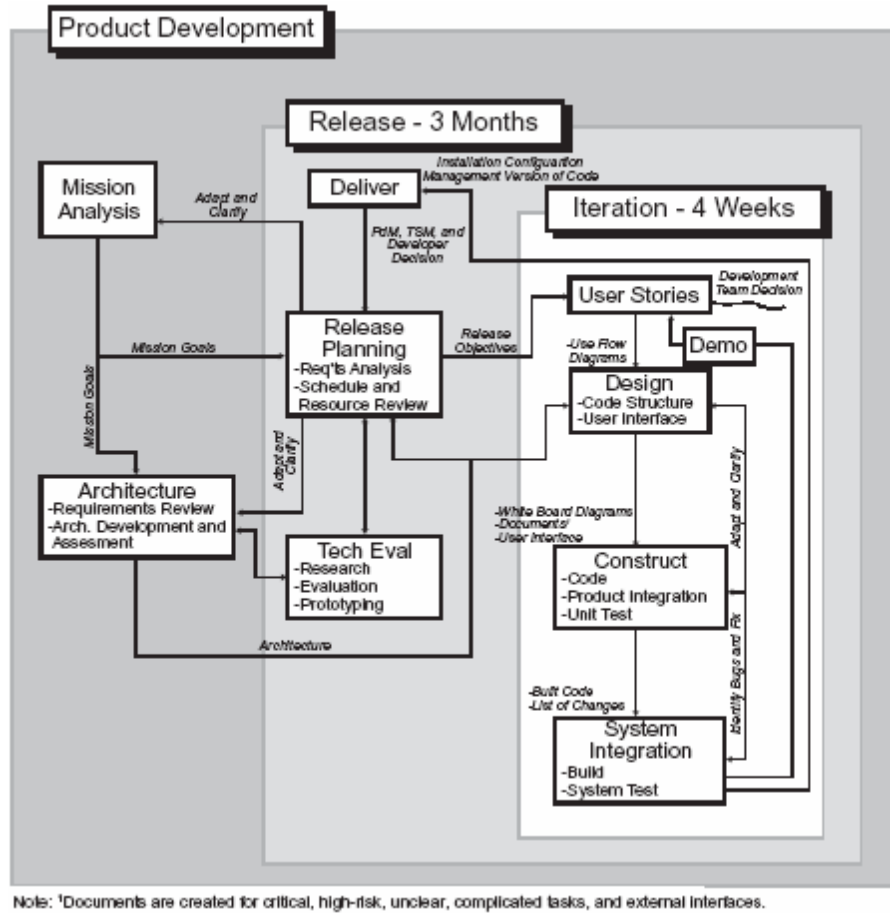
このシステム開発は米陸軍の Communications Electronics Command (CECOM) Software Engineering Center が中心となって開発を進めた。また、Shonborn-Becker Systems Inc.、L3 Ilex、Lockheed Martin、CECOM Research Development and Engineering Center などが開発支援を行っている。

この MCS Light のソフトウェア開発プロセスは従来型の開発とアジャイル開発手法のバランスをとった形で行われた。アジャイル開発手法を取り入れた部分として最も特徴的なのは、反復型の開発手法をとり、比較的短期間でリリースを行ったことにある。MCS Light 開発プロセスでは、4週間周期のイテレーションと、3か月ごとのリリース・サイクルを実施している。リリースは Product Manager レベルで承認が行われ、イテレーションはプロジェクト・リーダーのレベルで承認が行われている。1つのイテレーションを見てみると、開発チームは設計、開発、インテグレート、デモというライフ・サイクルを数回にわたって繰り返し行うことになる。

これに加えて、同プロジェクトでは、MCS Light を実際に利用するユーザーをリリースごとに巻き込む形で、アジャイル開発の真価を発揮した。MCS Light プロジェクトでは、Beta Sites というユニットが設置された。このユニットはユーザーに対してインクリメンタルにソフトウェアのリリースを行い、利用したユーザーからのフィードバックを受け取る機能を果たす。

ウォーターフォール型を代表とする従来型の陸軍のソフトウェア開発では何年も待ってはじめてユーザーが試験するというパターンが一般的であった。しかし Beta Sites を使って、できているコンポーネントを即座にかつ頻りにユーザーが試験を繰り返すことによって、ユーザーの真の要求により近いものができあがっていった。アジャイル宣言の中で謳われた『契約の交渉よりも、顧客とのコラボレーションを重視せよ』が実際にうまく機能した好例となった。

MCS Light のソフトウェア開発プロセス



これに加え、もうひとつのアジャイルの特徴としては、開発が3~10人程度の少人数のグループに分かれて行われたことが挙げられる。プログラマは同機関専属の職員だけでなく、民間からのコントラクターも含まれているが、両者ともそれぞれ責任ある開発を託されるにふさわしい、ソフトウェア・エンジニアリングの専門学位を持った優秀な人材がそろえられた。こうした人材が、グループごとに同じロケーションにあるオフィスの中で共同して、コミュニケーションを適宜はかりながら開発を進めることに成功している。

一方、従来型のソフトウェア開発手法も一部で採用された。包括的なドキュメンテーションの作成である。セキュリティに関わる重大システムの開発ということで、ドキュメンテーションの重要性を強調しているが、しかし、ここでも膨大なドキュメンテーション作成に追われて、アジャイルの機敏性が失われるようなことがないように留意がなされた。その方法として、あらゆるドキュメンテーションをアップデートするという方法ではなく、メンテナンスを必要とするドキュメンテーションを特定し、それだけを継続して行った。維持が必要とされたのは、アーキテクチャの変更に関わるものやユーザーガイドといったものであり、一方開発チーム内で必要とするドキュメントに関しては最小限に抑えるという努力が

なされた。こうしたドキュメンテーション手法で、実際に問題は起きていないと CECOM Software Engineering Center の John S. Willison は指摘している。

③ 航空宇宙局 (NASA)

NASA におけるアジャイル開発の 1 つは NASA Langley Research Center が中心となって行っている Fast Adaptive AeroSpace Tools (FAAST) に関するものである。FAAST プロジェクトでは次世代の航空宇宙分析やデザインを行うツール開発を行うことを目的としている。FAAST は現在以下の 4 つの主要研究が進んでいる。

- CAD-to-Grid Methods
- High Energy Flow Solver Synthesis (HEFSS)
- Optimally Convergent Algorithms
- Efficient Adjoint Design Methods

4 つの中で、アジャイル開発が行われているのは HEFSS である。HEFSS プロジェクトは、衛星システムなどと異なり、次世代システムを研究しながら開発していくため、要求仕様が固まっておらず、開発途中で常に変化し続ける可能性の高い場合ソフトウェア開発であった。そこで、Langley Research Center ではアジャイル開発手法の採用を決定した。数あるアジャイル開発手法であるが、同プロジェクトは最もよく利用され、成熟度の高い XP を使っている。また、30 分以内のミーティングを毎日行い、進捗や方向性を確認するという Scrum の手法も一部導入している。

4. アジャイル・プログラミング・コンソーシアム

現在、世界中で多種多様なアジャイル開発手法に関するコンソーシアムが結成されている。以下ではその中でも代表的なコンソーシアムについて紹介する。

(1) Agile Alliance

Agile Alliance は、前述のアジャイル宣言を発表していることから分かるように、アジャイル・ソフトウェア開発コンセプトを促進し、様々な組織による同コンセプトの採用を支援することを目的とする非営利組織として結成された。同コンソーシアムでは幅広いアジャイル開発に関する情報を提供している。また、アジャイル促進団体として、対象とする開発手法も以下のように非常に広範なものとなっている。



- AD (Agile Database Technique)
- AM (Agile Modeling)
- ASD (Adaptive Software Development)
- Crystal
- FDD (Feature Driven Development)
- DSDM (Dynamic Systems Development Method)
- Lean Software Development
- Scrum
- TDD (Test-Driven Design)
- XBreed
- XP (eXtreme Programming) 他

Agile Alliance は、ソフトウェア開発のための軽量アプローチ (lightweight approaches) を開発・実践していたソフトウェア専門家のグループが集まって結成された。Agile Alliance 結成当時は創始者の非公式な集まりという形であったが、よりオープンで、包括的で、より一般的に普及する方法を模索するために、非営利団体として Agile Alliance が発足することとなった。

XP 創始者として知られる Kent Beck をはじめとして、アジャイル・ソフトウェア開発で中心的な役割を担っている研究者や開発者が創設者として名を連ねている。

Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas

同コンソーシアムは、2005年7月に Agile 2005 Conference (デンバー) の開催を予定している。

(2) Systems and Software Consortium (SSCI)

Systems and Software Consortium (SSCI) はアジャイル・ソフトウェア開発手法のみを対象とするコンソーシアムではない。1986年に非営利団体として設立され、生産性、収益性、及び競争力を向上させるためのソフトウェア開発手法、ツール、技術などをサービスを会員、提携企業に提供することを目的としている。



SSCI は、航空宇宙、防衛、IT、通信、金融、政府及び大学といった各分野の第一級の機関から構成されている。民間企業としては、BEA Systems、Computer Science Corporation (CSC)、EDS (Electronic Data Systems)、General Dynamics、Lockheed Martin、Raytheon、Unisys などが参加している。

ソフトウェア開発を全般的に扱う SSCI では、9つあるエンジニアリング・プラクティスの中の1分野としてアジャイル・ソフトウェア開発手法に関するプログラムを提供している。SSCI が対象としているアジャイル開発手法としては、Scrum、FCC、及び XP が含まれている。同コンソーシアムはアジャイル・プログラミングの分野に関連して、Managing Agile Software Development と呼ばれる1日半のトレーニングコースを設ける他、以下のようなワークショップを会員向けに提供している。

- Agile Customer Focus Review Workshop
- Agile Kickoff Workshop
- Agile Mid-cycle Review Workshop
- Agile Retrospective Workshop

(3) Cutter Consortium

Cutter Consortium は、IT を競争上の優位性及びビジネス上の成功の実現に向けて活用する企業に対して、ビジネス及び IT 業界における一流の実務家による広範囲にわたるコンサルティング、トレーニング、知識などを通してサポートしている。コンソーシアムという名前ではあるが、Cutter Consortium は会員向けに専門知識、コンサルティング、及びトレーニングを提供するために協力する 100 名以上の国際的に著名な専門家から成るグループを抱えているユニークな民間のコンサルティング会社という色彩が強い。



Cutter Consortium が対象とするアジャイル手法は幅広く、XPをはじめ、Adaptive Software Development (ASD)、Crystal Light、Lean Development、Scrum、Dynamic Systems Development Method (DSDM) などが含まれている。同コンソーシアムが提供するサービスのうちで、アジャイル・ソフトウェア開発関連のサービスは Agile Project Management and Software Development がある。このサービスを通じて、会員がより正確なプロジェクトの予測可能性、より大きなクライアント満足度、チームの生産性の向上、製品化に必要な時間の短縮、欠陥の減少、及び予算超過の削減などを達成するためにアジャイル方法を最大限活用することをサポートしている。

また、同コンソーシアムにおいてアジャイル関連のコンサルタントとして、Jim Highsmith を筆頭として、Kent Beck、Ken Schwaber、Alistair Cockburn など、Agile Alliance の創設者として有名なアジャイル・ソフトウェア開発研究者などが多く参加している。また、同サービスで提供されているアジャイル・ワークショップとしては、以下のようなものがある。

- Agile Modeling: Effective Modeling for the Real World

- Agile Project Management: Innovation in Action
- Agile Requirements: Systems Visualization in the 21st Century
- Agile Software Development: A Review of Agile Methodologies
- Asset-Centric Software Development for Senior Managers
- Business Requirements Modeling
- Deadline-Driven Project Estimation
- Extending Requirements
- Extreme Programming Workshop
- eXtreme Project Management Masterclass
- The eXtreme Project Management Workshop™
- Implementing Lean Software Development
- Knowledge Team Leadership: The Art and Science of Being Powerful in Any Team
- Leading Successful Projects
- Managing the Deadline: A Project Management Masterclass
- Mastering the Requirements Process: How to Build the System Your Customer Wants
- Negotiation Bootcamp: Effective Techniques for Software Managers
- Rapid Software Testing
- Risk Management for Software: Learning to Contain, Mitigate, and Manage the Uncertainties of Software Development
- Risk Management War Games
- Scrum Project Quick Start
- Software Estimation --A Wolf in Sheep's Clothing
- Software Project Management
- A Taste of Extreme Programming
- Testing and Refactoring
- You Can Become Agile In One Day: Scrum Overview and Application

5. アジャイル開発手法をコア・サービスとして提供している企業

アジャイル・ソフトウェア開発手法の発展に伴い、アジャイル開発を売り物にしたり、関連するサービスを提供したりする企業が登場してきている。多くがXPを採用しているが、中にはいくつかのアジャイル開発手法をあわせて提供する企業もある。以下では、米国でそうしたサービスを提供している企業の事例を紹介する。

(1) ThoughtWorks（イリノイ州シカゴ）



アジャイル開発手法：XP、Scrum、Crystal、その他

ThoughtWorks は、1人から2人の少人数開発が可能なものから、100名で構成されるような大型開発プロジェクトまで、数多くの様々な規模のプロジェクトにおいてアジャイル・メソッドを上手く活用してきており、アジャイル・メソッド及びパターンにおける業界リーダーとして頭角を現してきている。

また、同社の主任研究員である Martin Fowler をはじめとする多くの ThoughtWorks 社の職員が、数多くのアジャイル・デベロップメントとその形態のアプリケーションに関する書籍、記事、論文等を出版している。

同社では、すでに複雑なカスタム・デベロップメント・プロジェクトを多種多様な業界の企業に提供してきたという実績を挙げてきた。また、同社の開発チームは次世代テクニックとテクノロジーを上手く利用し、それらを顧客の戦略的優位性へと迅速に変換するだけの能力があることを示してきている。

ThoughtWorks がこれまでにアジャイル開発アプローチでサービス提供を行ってきた企業の例としては以下のようなものがある。

ThoughtWorks の主要クライアント

Caterpillar Financial Services Corporation	CBOE
Cisco Systems Capital	Dixons Stores Group
Gap, Inc.	Horace Mann
iN DEMAND	Macquarie Bank
MetLife	Nationwide Financial
Nationwide Insurance	Rashid Hussain Bank
Safeco	SunCorp Metway
TransCanada Pipelines, Ltd.	Washington Mutual

(2) Role Model Software (ノースカロライナ州ホーリー・スプリングス)

アジャイル開発手法 : XP



Role Model Software は以下3点の主要な方法を通して顧客と提携し、それぞれの顧客そのビジネス・ニーズに適ったソフトウェアをアジャイル手法によって開発するサポートを行っている。

- ① オフサイト・デベロップメント (Offsite Development)

Role Model Software のハイパフォーマンス・チームが同社内に設けられたエクストリーム・プランニング・ソフトウェア・スタジオ (Extreme Programming Software Studio) においてソフトウェアを製作する。
- ② オンサイト・デベロップメント (Onsite Development)

Role Model Software のハイパフォーマンス・チームがクライアントの事務所においてソフトウェアを製作する。これはクライアントが自社内に「スタジオ環境」をセットアップすることをサポートするものであり、Role Model Software 社のチームはそのセットアップされた環境においてソフトウェアを提供するために働く。
- ③ 指導と共同開発 (Co-Development with Mentoring)

クライアントが自社のソフトウェア開発チームの効率改善を望んでいる場合に、Role Model Software 社がサポートを行う。この場合、同社は

クライアントのソフトウェア開発者に対して XP に関するトレーニングを行った後、共同でソフトウェアを開発に当たる。

例えば、Role Model Software のクライアントの 1 つである CipherOptics, Inc はハイスピード・ギガビット・ネットワーク・セキュリティ・ソリューション提供プロバイダーである。CipherOptics は国防総省 (DOD) などを含む世界で最もセキュリティに関する要求の厳しいユーザーが、セキュリティ・ゲートウェイを設定できるようなデスクトップ・アプリケーション開発の必要に迫られていた。この目的を実現させるために、CipherOptics は短期間で、セキュリティ・ゲートウェイを動かすソフトウェアとデスクトップ・アプリケーション間のインターフェース開発などを含む一連の関連プロジェクトを成功させるため、Role Model と契約を締結。Role Model は XP を利用して、わずか 1 ヶ月強の間に、6 人強のメンバーで Linux、Windows、Solaris などを含む複合型オペレーション・システム環境と 100 以上のゲートウェイを利用している顧客向けの製品を納品した。

(3) Object Mentor (イリノイ州ガーニー)

Object Mentor, Inc.

アジャイル開発手法 : XP

Object Mentor の目標は、それぞれのアジャイル・ソフトウェア開発プロジェクトを完遂することができるよう企業をサポートするうえで、必要不可欠なリーダーシップを提供することにある。豊富な経験を持つ同社ソフトウェア開発者が、コンサルティング、トレーニング、指導及び開発サービスを通して、クライアントのアジャイル開発者がリーダーシップを発揮できるよう支援する。

同社のアジャイル開発は、1998 年、XP のアジャイル・デベロップメント・メソッドを受け入れたことに始まる。このアジャイルへの積極的な取り組みにより同社はソフトウェア業界をリードすることとなった。また 2000 年には、前述の Agile Alliance の創設を援助することによってさらなる成長を遂げている。XP 関連のサービス提供をしている主なクライアントとして、Symatec (セキュリティ・ソフトウェア)、Workshare (ドキュメント・マネージメント)、Escrow.com (オンライン信用決済)、Qwest (通信)、Fidelity Investments (投資) などの企業が含まれている。

同社はアジャイル開発のほか、オブジェクト指向テクノロジーの提唱者であり、常にソフトウェア開発テクノロジーにおける最先端に位置してきた。同社はまた、書籍、論文、記事の出版、さらには講義等を通して C++、Java、及びオブジェクト指向デザインにおける最先端技術の整備に取り組んできている。

(4) Agile Logic (カリフォルニア州フラートン)

アジャイル開発手法 : XP、Scrum、DSDM

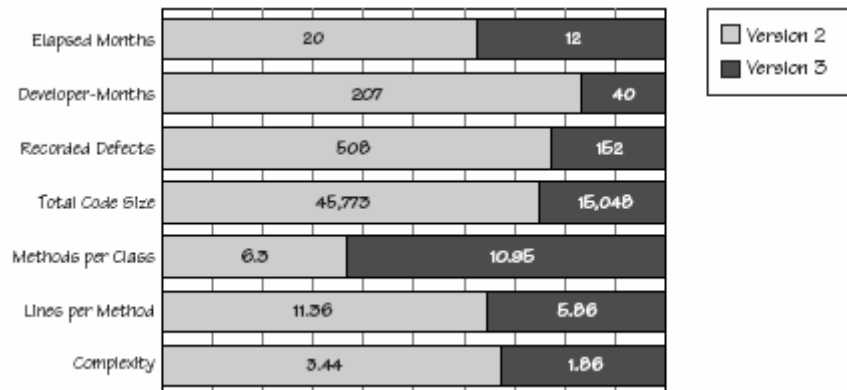


Agile Logic 社は、アジャイル・プロセスと最先端技術を必要とするクライアントに対してコンサルティング、トレーニング、指導及び開発サービスを提供している。同社の開発専門家チームは、顧客のプロジェクトを最高レベルの品質と共に迅速に実現する非常に優れた能力を提供しており、顧客が既存のチームを補完する必要があると、新たなプロジェクトを即座にアウトソースする必要があると、そのためには何をなすべきなのかを同社はよく理解している。以下が同社のサービスである。

- フル・ライフサイクル・デベロップメント・サービス
- オンサイト・プロジェクト専門チームもしくは同社デベロッパーによるクライアント・チーム補完
- アウトソースされたプロジェクト
- コントラクト・デベロップメント
- ウェブサイト・デベロップメント

同社の顧客の1つとして、米国エスクロー・サービス最大手の Escrow.com (カリフォルニア州) がある。Escrow.com は急速な変化の起こる B2B の E-Commerce 市場に位置している。同マーケットにおけるリーディング・カンパニーとして成長するために、同社は常に時間との戦いに直面していた。Agile Logic は Escrow.com のソリューション開発の一部に XP を導入した。同開発ではバージョン2を従来の開発手法を使い、バージョン3のアップデートをXPを使って実施した。その結果、アジャイル手法を使った開発で大幅にコスト削減、開発期間短縮、ソフトウェア品質の向上などが図られたことが以下の図で示されている。

従来開発型 (Version 2) とアジャイル型 (Version 3) の比較



(参考資料)

[Craig Larman, *Agile and Iterative Development: A Manager's Guide*](#)

[Barry Boehm & Richard Turner, *Balancing Agility and Discipline: A Guide for the Perplexed*](#)

[Robert C. Martin, *Agile Software Development: Principles, Patterns, and Practices*](#)

<http://www.agilemanifesto.org/>

<http://www.adaptivesd.com>

<http://alistair.cockburn.us/crystal/crystal.html>

<http://www.dsdm.org/>

<http://www.enterpriseunifiedprocess.com/>

<http://www.xprogramming.com/>

<http://www.featuredrivendevelopment.com/>

<http://www-306.ibm.com/software/awdtools/rup/>

<http://www.controlchaos.com/>

<http://www.agiledata.org/essays/differentStrategies.html>

<http://www.niwotridge.com/PDFs/ADC%20Final.pdf>

<http://www.khecareer.com/login/RFETS/KHLLC.pdf>

http://www.ch2m.com/corporate_2004/

<http://aec.army.mil/usaec/>

<http://www.fcw.com/fcw/articles/2004/0426/tec-agile-04-26-04.asp>

<http://www.sed.monmouth.army.mil/c4iews/avionics/>

<http://www.stsc.hill.af.mil/crosstalk/2004/04/0404Willison.pdf>

<http://www.nasa.gov/centers/langley/home/index.html>

<http://techreports.larc.nasa.gov/ltrs/PDF/2003/aiaa/NASA-aiaa-2003-3978.pdf>

<http://www.agilealliance.org>

<http://www.systemsandsoftware.org/ssci/default.asp>

<https://www.software.org/catalog/listing.asp?idCapability=195>

<http://www.cutter.com>

<http://www.cutter.com/project/index.html>

<http://www.cutter.com/project/consultants.html>

<http://www.thoughtworks.com>

<http://www.thoughtworks.com/us/industries/index.html>

<http://www.rolemodelsoftware.com>

<http://www.rolemodelsoftware.com/services/caseStudies.php>

<http://www.objectmentor.com>

<http://www.objectmentor.com/ourCustomers/index>

<http://www.agilelogic.com>

<http://www.agilelogic.com/files/ExtremeProgrammingPerspectivesCh30.pdf>

このレポートに対するご質問、ご意見、ご要望がありましたら、
hiroyoshi_watanabe@jetro.go.jpまでお願いします。