

# アジャイルプロセスと XP プログラミング

Agile Process and eXtream Programming

～新しいソフトウェア開発手法アジャイルプロセスの挑戦～

中山 靖一, 伏見 俊彦

Seiichi NAKAYAMA, Toshihiko FUSHIMI

## 1 はじめに

近年ソフトウェア開発における要求が変わってきている。従来は、ソフトウェア開発は長期にわたり大勢の人間で行われるものであった。しかし、今では短期間で、できるだけ低コストで開発を行う必要が出てきた。なぜなら社会状況やマーケットの変動の激しさに伴い、ソフトウェアに対する要求も日々変化しているからである。このような社会背景では従来のソフトウェア開発手法では対応できなくなっている。このような問題を解決する手法の 1 つとして現在、アジャイルプロセスというものに注目が集まっている<sup>1)</sup>。

## 2 アジャイルプロセス

### 2.1 アジャイルプロセスとは

アジャイルプロセスとは、顧客に価値をもたらす動作が保証されたソフトを迅速に実現するという目標を持つソフトウェア開発手法の総称である。アジャイルプロセスには、XP プログラミング (以下 XP) を始め、Crystal<sup>1)</sup>, FDD<sup>1)</sup>, DSDM<sup>1)</sup>, Scrum<sup>1)</sup> などさまざまな手法がある。これらのアジャイルプロセスの共通点を以下に示す。

- 余計な成果物や手順を排除する。
- 動くソフトをできるだけ早く提供する。
- ソフトに対する要求を可能な限り受け入れる。
- コミュニケーションなど人間的な要素を重視する。
- 開発チームに顧客を加入させる。

このような点を考慮することで上述した目標を実現する。

### 2.2 アジャイルプロセス導入の注意点

アジャイルプロセス導入の際は、単に XP などの手順を適用すればよいわけではない。まず、開発チームは、現在のソフトウェア開発における問題点を洗いだし、良い点はそのまま残しておく。そして、各アジャイルプロセス手法の中から現在のプロジェクトに有効なものを選定する。しかし多くの場合、その手法に追加・修正などを行ってから適用する。Fig. 1 に、システムの規模と重要度による各アジャイルプロセス手法の適用範囲を示す。

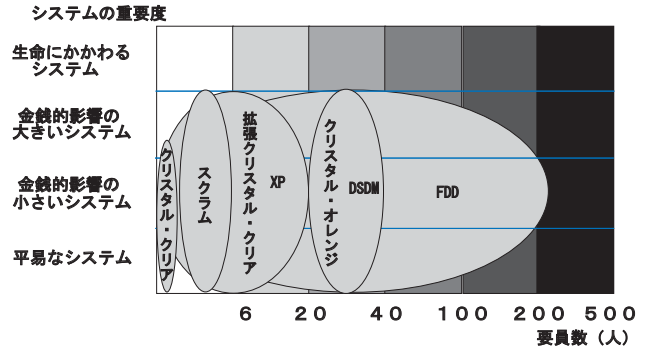


Fig. 1 アジャイルプロセス手法の適用するシステムの規模と重要度

### 2.3 アジャイルプロセスに対する企業の注目度

現在ソフトウェア開発に問題を抱える企業は多いため、新しいソフトウェア開発手法の 1 つであるアジャイルプロセスに対する企業の注目度は高い。アジャイルプロセス協議会<sup>1)</sup>には、ソニー、NEC、マイクロソフトなども加入している。しかし、アジャイルプロセスを用いた商用プロジェクトの事例が非常に少ないこと、開発者の新手法に対する抵抗など様々な要因から、企業は実際にアジャイルプロセスを導入することに対しては非常に慎重であると思われる。

## 3 XP

### 3.1 XP とは

XP はアジャイルプロセス手法の中で日本では最も有名な手法で、Kent Beck らによって提唱された。XP は開発リスクを早期に軽減することに主眼をおき、少数のプログラマーが同じ部屋で非常に短いサイクルで繰り返し開発する反復開発法である。XP を実践する際に、何を重視すべきかの判断基準として 4 つの価値を示している。また、Kent Beck は、開発チームが従うべき実践項目を 14 のプラクティスとしてまとめている<sup>2)</sup>。

### 3.2 4 つの価値

XP の 4 つの価値について以下に示す。

<sup>1)</sup>2003 年 2 月 1 日に発足した。日本におけるアジャイルプロセスの普及、推進、情報交換などを目的としている

- コミュニケーション  
開発チームが効率よく開発していくために重視
- 簡潔さ  
メンテナンスをしやすい、バグを最小限にするために重視
- フィードバック  
システムの品質を常に監視するために重視
- 勇気  
大きな問題に果敢に対処するために重視

### 3.3 14のプラクティス

XPの14のプラクティスについて以下に示す。

- コミュニケーションするためのプラクティス  
オンサイト顧客、ペアプログラミング、メタファ、オープンワークスペース、計画ゲーム
- 簡潔さを追及するためのプラクティス  
シンプルな設計、リファクタリング、日ごとのスキーマの移行、コードの共同所有、コーディング標準
- フィードバックを得るためのプラクティス  
小さなリリース、テストング、継続的インテグレーション
- その他  
週40時間労働

### 3.4 XPの目的

従来のソフトウェア開発手法では、要求分析、設計、実装、テスト、保守と開発が進むにつれ変更にかかるコストが増大するので、Fig. 2(a)のようになると考えられている。しかし、Kent BeckはXPを用いれば、Fig. 2(b)のような変更コストを実現できることを仮定している。この仮定は「XPを行うことで得られ」、また「XPを行うために必要である」という相補的關係にある。

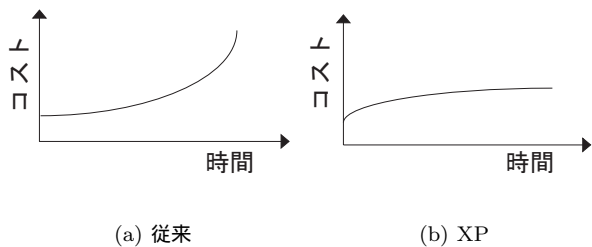


Fig. 2 時間-コスト曲線

### 3.5 XPの手順

XPの手順は、一回の開発サイクル(イテレーション)で開発する機能(ストーリー)を顧客が選び、その機能をプログラムとして実現していくという作業を繰り返すものである。1回のイテレーションの期間は1~4週間ほどで、1つのリリースを実現するにはイテレーション

を2~5回繰り返す。この過程の中で、開発チームは、プラクティスは遵守しなければならない。XPによるソフトウェア開発の流れをFig. 3に示す。

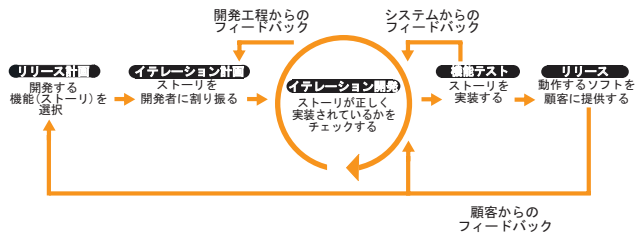


Fig. 3 XPによるソフトウェア開発の流れ

## 4 XPを研究室内で行う意義について

XPを始めアジャイルプロセスを研究室内に導入する意義はあると考えられる。特に、研究グループで1つのシステムを構築する場合はXPの効果を発揮できると考えられる。また、XPは短期間でシステムを開発するため、実験データからのフィードバックを生かせる。しかし、個人で研究テーマが異なる場合は本来のXPの効果は出にくいと考えられる。

XPのプラクティスを選定して研究室内に取り入れることは個人、研究グループに関係なく意義のあることだと考えられる。プラクティスの1つであるペアプログラミングを導入することでコミュニケーションの促進、プログラムの質が向上、コードの共有化、スキルの向上などの良い効果をもたらす。

## 5 今後の展望

XPなどはコミュニケーションなど人間的な要素を重視した手法であるが、大規模なソフトウェア開発が困難である点などビジネス的な視点に欠けている面がある。しかし、ソフトウェア開発において、急速に変化するビジネスニーズに柔軟に対応するためには、XPのように開発速度やコミュニケーションなど人間的な要素を重視することは必要不可欠である。今後、XPが普及していくためにはビジネス的な視点を取り入れる必要があると思われる。そのため、XPは、ビジネス面を重視したプラクティスの策定を行うか、あるいは、ビジネス面を重視した開発手法と融合して普及していくのではないかと考えられる。

## 参考文献

- 1) “Part3 手法：超高速開発を実現「アジャイル」の全貌”，日経コンピュータ(2002, 10)
- 2) Kenji Hiranabe, “XP(eXtreme Programming):ソフトウェア開発の新潮流～前編：XP概要とその周辺”，情報処理, Vol.43(情報処理学会)