エクストリーム・プログラミングの実際

富士ソフトＡＢＣ 株式会社

■ 執筆者Ｐｒｏｆｉｌｅ ■

上　村　太　郎

2004 年　富士ソフト ABC(株)入社
システム事業本部
ビジネスアプリケーション部配属
2004 年　現在 システム事業本部
通信制御部所属

■ 論文要旨 ■

ケント・ベック氏が提唱したエクストリーム・プログラミングは，中規模の開発に向くソフトウェア開発の究極の手法であるといわれる．しかし，日本における導入にあたっては，米国ほど普及したとはいえない感がある．実際に日本国内の開発現場においてエクストリーム・プログラミングを用いた事例を通して，その実効性を検証する．

■ 論文目次 ■

■ 図表一覧 ■

１．はじめに

１．１　当社概要

富士ソフト ABC 株式会社は 1970 年（昭和 45 年）5 月に設立され，以来「国内最大の独立系 SE 集団」として，「ひのき（品質，納期，機密保持）」を開発ポリシーに，特定のプラットフォームにとらわれることなく，お客様に最適な環境とは何かを考えている．大規模エンタープライズシステム構築においては，業務分析と IT コンサルティングの段階から，お客様の視点に立った柔軟な発想で最適なソリューションを提案している．高度な知識と豊富な経験に基づき，お客様を取り巻くシステム環境をあらゆる角度から精査しながら，要件定義・システム設計，プログラミングからテスト・納品まで，一貫した開発理念が高品質・高信頼性を提供している．保守，ヘルプデスク，教育といったシステム運用に至るまで，IT に関わるすべての技術サービスをお届けする「総合システムビルダー」として IT のベストパートナーとなる企業である．

1970 年　5 月　会社設立
1992 年 10 月　東証二部上場
1995 年　6 月　ISO9001 認証取得
1998 年　8 月　ISO14001 認証取得
1998 年　9 月　東証一部上場
2002 年　5 月　プライバシーマーク認証取得

詳しくは当社ホームページ（ URL http://www.fsi.co.jp/ ）を参照．

１．２　論文概要

1999 年にケント・ベック氏（Kent Beck）が提唱したエクストリーム・プログラミング（以下 XP）は，中規模の開発に向くソフトウェア開発の究極の手法であるといわれる．XP 誕生のきっかけは 1996 年にケント・ベックがコンサルタントとして参加した自動車メーカーにおけるプロジェクトであったとされている．このプロジェクトにおいて同氏が発案したいくつかの原則が XP の基盤となっている．また，1999 年には XP についての最初の著作である「Extreme Programming Explained:Embrace Change」が出版され，世界中で XP の手法が注目されるようになった．しかし，日本におけるその導入にあたっては，米国ほど急速に普及したとはいえない感がある．実際に日本国内の開発現場において XP を用いた事例を通して，その実効性を検証する．同時に日本特有のシステム開発上の問題点についても考察する．

２．XP 開発手法と導入時の課題

２．１　XP の特徴

ケント・ベック氏は XP 開発手法において「変化に対応する」という思想を貫いており，ユーザーからの要求の変化を問題視するどころか，それがソフトウェアの価値を高めるとまで述べている．その考え方を現実化するために「テストコードを最初に記す」という方法が提唱されている．テストコードとは，テストのためだけに作成したサンプルのコード

を意味する．言い換えると，このテストコードは，ユーザーが要求する仕様を外枠だけ作成したプログラムともいえる．最初のうちは中身の機能が実装されていないため，テストコードのコンパイルさえ通らない状態のはずである．しかし，そのテストコードをパスするコードを実装することでユーザーの要求に沿った機能がおのずと実装できると同氏は述べる．また，それに伴って，ドキュメントを極力省略するようにと勧められている．つまり，詳細な仕様についてはわざわざ文書化する必要はなく，むしろユーザーが要求する仕様は簡単なカードに記される程度のものにとどめる．確かに詳細な仕様を必要とするのは，ユーザーではなく，むしろプログラマーである可能性が高いため，これは理にかなっているともいえよう．

更に XP を特徴付けているのはペアプログラミングである．これは，二人が一組になって一緒にプログラミング作業に従事するという方法で，新人の育成や知識の共有化などの利点がある．また，複数のプログラマーがともに作業することで互いに親しくなり円滑なコミュニケーションを促進するとともに，一部の人間に負担が集中することを避け，重荷やプレッシャーを軽減するという心理的な効果もある．

いずれも同氏が実際の開発現場で感じた苛立ちや問題を克服するために編み出した手法で，一部においては以前から普通に行なわれていたかもしれない手法も含まれている．ただし，それらを系統立ててまとめたという点では XP は画期的な手法だといえよう．

２．２　導入時の課題

日本において XP の手法を導入しようとする場合に障害となり得る点を以下に述べる．

まず，テストコードをパスするためにコードを書くという「テスト駆動型」の考え方である．この場合，一般のシステムで言うところの「詳細設計書」を作成するのではなく，むしろユーザーによって作成された「概要設計書」のみを頼りに，まず「ユーザーが何をしたいか」を明確にし，その要求にかなうテストコードを作成し，次いでそのテストコードをパスするコードを実装していくという手順になる．しかもその概要設計は簡単な「ストーリーカード」に記されることになる．確かにユーザーが求めるものは分厚い書類の山ではなく，要求に沿って正確に動くシステムである．文書化されたものはそのシステムを補完する役割に過ぎない．しかし，日本の製造業では成果物として目に見えるものを要求する風潮があり，システム開発の現場でもその影響はあるように思われる．そのため，多額の対価に対して，納品されるのがシステム全体の収まった CDROM 一枚だけでは腑に落ちないとの考えが根底にあるのではないだろうか．いずれにせよ，概要設計書や詳細設計書，関数詳細，テスト要領書，テスト成績書など多数の文書が納品されることになる．この習慣は XP におけるテスト駆動型の開発手順を導入する際に障壁となる．

またペアプログラミングについては，開発効率を疑問視する声もある．欧米のように個人主義が徹底しており，二人のプログラマーが互いに自己主張を繰り返しながら切磋琢磨してゆく環境であれば，この方法は大きな効果を期待できるかもしれない．しかし，日本では主張をぶつけあいながら互いを高めあうやり方には慣れていない．むしろ複数でともに作業すると，相手に遠慮しつつ，かつ互いに互いの弱い部分を率直に指摘することなく，優しさを示して許し覆ってしまう傾向がある．「和をもって尊し」とする国民性の是非はともかく，XP の世界においては，互いを研鑽しつつ開発作業を進めるというペアプログラ

ミングの思想とはかなり乖離が見られることになる．そのため，こうしたやり方では，各個人が自分の責任範囲内で最善の仕事をこなすという従来の方法が効率良く進むだろう．これでは開発効率が疑問視されても致し方ない部分はある．

最後に技術者のスキルの問題があるように思われる．XP の開発手法はオブジェクト指向プログラミングに最適であり，むしろ，そのための開発手法であるといっても過言ではない．しかし日本においてはオブジェクト指向を理解している技術者がまだ少なく，またその価値の認識度については欧米の技術者に遠く及ばない．そのため，従来のプロジェクト計画の線表上に XP の開発手法を導入しようとすると無理が生じる．本来，オブジェクト指向の概念に乗っ取って導入すべき XP の手法が，その方法論だけ抽出された状態で強引に導入されてもプロジェクトを成功に導くことはできないだろう．

３．XP 導入の実際

３．１　プロジェクト概要

公益法人団体の人事管理システム作成のプロジェクトを XP の手法を用いて開発した．このプロジェクトで XP の手法を用いた理由はシステムが小規模であったこと，ほかのシステムとの連動がなく独立したシステムであり既存システムへの影響が少ないこと，またユーザーが公益法人という性質上，納期やコスト面での要求がシビアではなく，実験的な手法を受け入れてくれる寛容な風土があったことなどがある．実際の開発体制では，リーダーを含めて，開発者３人体制で臨み，およそ３ヶ月で完成した．使用した言語は Visual Basic で，メンバー構成は**表１**のとおりである．なお**表１**における経験年数とは，入社以来，業務に携わり始めてからの総年数である．

表１　プロジェクトメンバー表

開発者	経験年数	役割
A	５年	リーダー
B	３年	開発担当者
C	半年	開発担当者

リーダーは主にユーザーとの折衝にあたり，内部のコーディングは開発者 B が主に担当した．本プロジェクトでは，経験年数がまだ浅い新人の開発者 C を投入し，開発者 B とペアプログラミングすることから，知識の共有と教育の効果も期待した．

３．２　導入時の工夫とその効果

まず，ユーザー要求を極力シンプルにまとめるために，XP で提唱されているストーリーカードと同等のものを制作した．カードは A5 サイズのものを用いて，なるべく長文を避け，一目で内容を把握できるように工夫した．当初はユーザーにストーリーカードを制作してもらうつもりであったが，実際にはリーダーがユーザーとの打ち合わせの際に，要求をヒアリングして文言にまとめるという形になった．更にストーリーカードを元にしてタスクカードを作成した．これは XP の手法の一つで，２～３日でできるような作業を一つのタスクとしてカードに記録していき，それぞれの状態をメンバー全員が参照できるように公開

するというものである．本プロジェクトでは，「未着手」，「保留」，「完成」の３種類の状態をあらかじめタスクカードに印刷しておき，それぞれのタスクの状態にあわせて該当する部分を鉛筆で囲むという方法にした．

更に本プロジェクトではタスクカードの開示方法を独自に工夫した．**図１**に示すように，それらのカードをリーダーのデスク横にあるパーティションに貼付しておき，いつでもメンバーがそこに来ては，各タスクの状態を書き込んだり，また確認できるようにした．

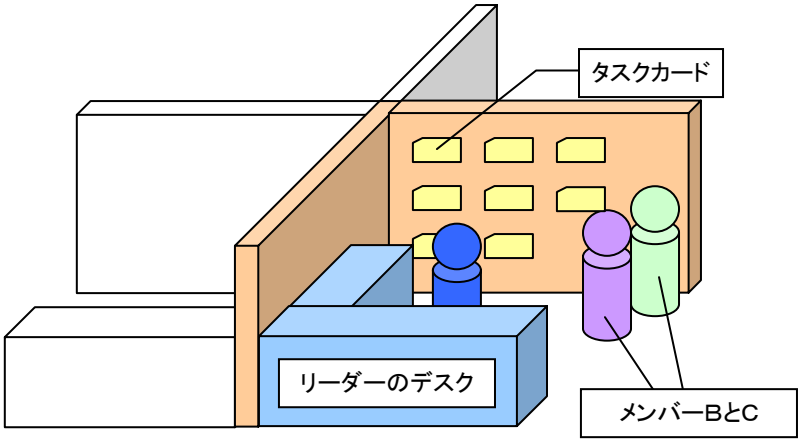

図１　タスクカードの配置図

この方法は，各メンバーとリーダーのコミュニケーションを図る良い機会にもつながった．最終的には納品時にリーダーがこれらストーリーカードとタスクカードを元にした設計書を作成し，ユーザーに納品するという形を取った．本来 XP の手法から考えると，文書化は極力避ける方向ではあるが，我々としてはやはりお客様の納得と了解を得るためにはある程度の文書は必要であろうという結論に至った．ただし，XP のストーリーカードとタスクカードという手法を用いることで，設計書の作成は実際のコーディング作業とほぼ並行して行なわれ，かつユーザーからの仕様変更要求にも柔軟に対応できるものとなった．

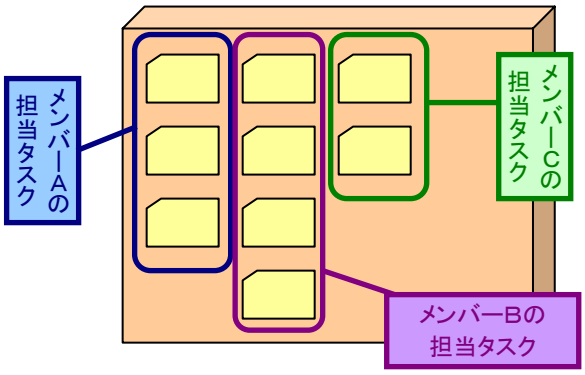

図２　タスクカードの掲示方法

更に，パーティションに貼付されたタスクカードは，**図２**のように担当者ごとに分けて掲示するように工夫した．この方法は，ペアプログラミングによって，責任の所在が不明

確になることを避けるのに役立った．各自の担当するタスクが目に見えて明らかになるため，それぞれの責任と自覚を促すことができた．

また，ペアプログラミングにおいては，経験年数のある開発者 B と新人の開発者 C がペアを組んだが，その際に一定のルールを設けた．それは「互いに忌憚無く意見を述べ合う」という XP の本来のルールに加えて「相手の動機を悪く考えないこと」というものである．ペアプログラミングがうまくいかないケースでは精神的・人間的関係の問題に原因がある場合が多い．他人がどう思っているか気になる，また他人の発言の背後にある動機が気になる，という性格のメンバーで構成されていた本プロジェクトでは，最初から「相手を信頼すること」を一つのルールとして徹底しておくことで，精神的抑圧からの逃げ道を半強制的に設けておいた．**図３**に示すとおり，これはかなりの心理的効果があり，些細なことで行き違いがあっても「相手が自分を信頼してくれているのだから，自分も相手を信頼しよう」という積極的な発想へと気持ちを転換することができた．

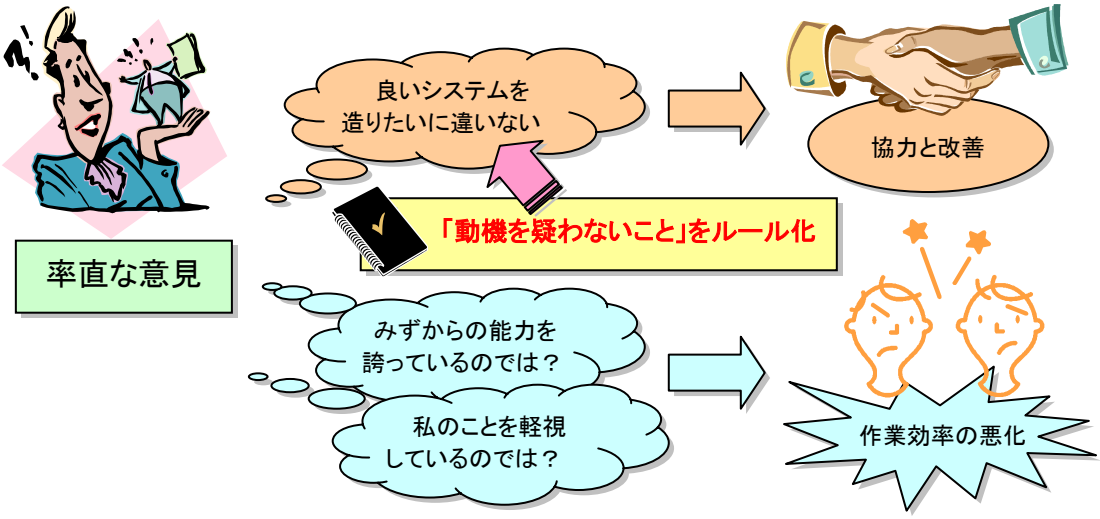

図３　信頼のルール化

このルールを確実に推進する上でリーダーは調整役となり，ルールの主旨や意義に対するメンバーの意識を定期的に啓蒙するようにした．週ごとのミーティングでは，各メンバーから不満や苦情を聞き出す機会を必ず設け，その場で円満に解決するようにした．また，メンバーがこのルールに甘えて，作業の進捗率や生産性に無頓着な傾向が見られた場合には，リーダーが躊躇せずに助言と矯正を与えることで，このルールを責任逃れの言い訳にしないように配慮した．

加えてリーダーはメンバーの人間関係に気を配り，開発者 B が開発者 C とのペアを負担に感じる状況下では，自らが一時的に開発者 C とのペアを組むなどしてペアプログラミングの取り決めをある程度流動的なものにした．そうすることで，本来メンバーの負担を軽減する目的で取り決められたペアプログラミングの手法が，逆に各自に負担を与えることがないように取り計らった．

また，開発者 B と開発者 C のペアでプログラミングすることは開発者 C の教育にも役立った．加えて，開発者 B にとっても，新人の開発者 C とともに作業することで一時的に効率は落ちたように感じても，仕様を新人に対してどのように分かり易く説明すればよいか

再考する場面が多々あり，仕様の理解と確認を徹底して行なう良い機会になった．このときも，互いへの信頼感をルール化していたことが功を奏しており，開発者 B は開発者 C に対して相応の期待を抱きつつ教育を施したため，開発者 C はその期待に応えようと真剣に業務に取り組むことができた．これはピグマリオン効果と呼ばれるもので，予想外の収穫であった．

以上，導入手法において工夫した点とその効果を**表２**に示す．

表２　導入手法の工夫とその効果

導入した手法	工夫点	効果
タスクカード	・タスクカードをパーティションに公開	・リーダーとメンバーのコミュニケーション促進
ペアプログラミング	・信頼のルール化	・メンバー間のコミュニケーション促進 ・新人教育のピグマリオン効果

３．３　今後の課題

本プロジェクトでは，ケント・ベック氏の著書である「XP エクストリーム・プログラミング入門」を元に XP の手法を導入した．しかし当初はそのルールにとらわれ過ぎ，その方法論のみに終始した感があった．そのためメンバー内では，「この方法は日本では当てはまらない」とか「本プロジェクトには該当しない」との不満が鬱積した．そこでプロジェクト開始後しばらくしてからは，ケント・ベック氏がなぜ XP でこの手法を用いているのか，その目的を考え，それらが本プロジェクトに合致する場合だけその方法を取り入れることにした．またその場合でも，提唱されている手法をそのまま当てはめるのではなく，その目的を考慮した上でメンバーができる範囲内で，実践し易い方法に改修して取り入れることにした．その結果，多少手探りで開発を進める場面は見られたものの，比較的円滑に開発が進んだと思われる．特に本プロジェクトでは，XP のルールを明確にし，それらをメンバー全員が厳守する必要性を痛感した．それは XP のみならず，一般的なプロジェクトの進行にも当てはまることであるが，特に XP の手法を用いる場合には，今までの開発手法と異なる部分が多いため，明確なルールを策定し，その実施を徹底する必要が大きい．今後は明確なルールをプロジェクト開始前から策定しておき，それらのルールを全メンバーに周知徹底するよう備えておく必要がある．

４．おわりに

XP の開発手法は一時期，いくらかの話題にはなったものの，広く普及するにまでは至らなかった．その原因として，日米のシステム開発を取り巻く環境の違いや国民性の違いなどが挙げられる．また，その斬新な手法をそのまま受け入れるのは難しいと感じた開発者も多いかもしれない．しかし，XP の手法を自分たちなりに改修して取り入れることで日本独自の，あるいは自社独自の手法を確立することができる．またペアプログラミングのような人間関係に影響を及ぼす可能性のある手法は，日本人の国民性や志向性を意識して取

り入れるなら大きな効果を期待することができる.

XP の開発手法のみならず，ほかの技術についても，IT 業界にはときとして欧米のやり方をただ追従するだけに走りがちな傾向が観察される．そうした無理な模倣から脱却して，日本でも使えるところだけを吸収し，日本風にアレンジすることで原版よりも更に優れた手法や技術を編み出してゆくなら，充分に国際競争力を発揮することができるのではないかと感じる.

参考文献

［1］ケント・ベック：“XPエクストリーム・プログラミング入門”，ピアソン・エデュケーション