アジャイル開発プロセス導入による「品質向上」「ユーザ指向」「要員教育」「工期短縮」への挑戦

（株）ＮＤＫＣＯＭ

■ 執筆者Ｐｒｏｆｉｌｅ ■

福田　祐一郎

1993 年　長崎電子計算センター（株）入社
システム開発業務担当
2006 年　現在 （株）ＮＤＫＣＯＭ（社名変更）
システム企画開発部 主務

吉田　明子

1982 年　長崎電子計算センター（株）入社
システム開発業務担当
2006 年　現在 （株）ＮＤＫＣＯＭ（社名変更）
システム企画開発部 部長

■ 論文要旨 ■

「アジャイル開発プロセス」は，「アジャイル（俊敏）」にソフトウェアを開発する方法論の総称であり，明確な定義がないため，その実態は掴みにくく，実践報告も少ない．にも関わらず，「アジャイル開発プロセス」には，現在，ソフトウェア開発に求められる，工期の短縮，低コスト，オープン，ユーザ指向，要員教育などの問題に柔軟に対応できる，最も新しい革新的な開発プロセスとして熱い期待が寄せられている．

「アジャイル PMBOK」（以下，APMBOK という）は，PMBOK の提供するフレームワークをかりてアジャイル開発プロセスの，プロジェクト管理に関するプラクティスをまとめたものである．我々は新商品開発プロジェクトにおいて，この APMBOK を用いてプロジェクト管理を行うこととし，開発をすすめている．今回は，開始プロセスグループから計画プロセスグループ（コア・プロセス）において，各プロセスでの実際の活動の中で，ユーザ指向と工期の問題を中心に，本プロジェクトでの「アジャイル開発プロセス」導入の評価と今後の課題を挙げる．

■ 論文目次 ■

■ 図表一覧 ■

１．はじめに

１．１　はじめに

株式会社 NDKCOM（以下，NDKCOM という）は，1966 年，長崎電子計算センター株式会社として創業し，汎用機を使い自治体を中心に地域の情報処理サービスを行ってきた．今回行った WEB アプリケーションのパッケージ商品開発では，従来のプロジェクト管理であるウォータフォール型の開発で取り組んだが，「短期間」，「低コスト」，「エンドユーザ指向」といった，難しい課題を前に，プロジェクトが一歩も前に進めない状態に陥いり，これを中断せざるを得ない事態となった．

このため，当社が直面しているこれらの課題に対する問題点に対応し，プロジェクトを推進するために，プロジェクトの開発プロセスの見直しを行った．実験的な試みとして「アジャイル開発プロセス」を導入することを決定し，思考錯誤の中やり直した結果，やっと開始プロセスと計画プロセスの一部を終了するに至った．

今回，開発プロセスを開始するにあたり，「アジャイル開発プロセス」導入後の各工程での取り組みを振り返り，更に今後の工程における課題の抽出を行った．

２．新商品開発プロジェクトの立ち上げ

２．１　新商品開発プロジェクトの概要

NDKCOM では，2006 年始めに「チャレンジ 2010」と称して，大幅な業務見直し及び新ビジネスの創出が計画された．我々システム企画開発部では，これまでの人月依存型ビジネスから脱却すべく，オリジナルの商品の開発を計画した．

開発する商品は，市場性，実現性，将来性，採算性など様々な視点から検討し決定した．（新商品検討の内容は，本趣旨から外れるため割愛する．）

早速，新商品開発プロジェクト（以下，本プロジェクト という）を発足すべく，社内で，商品の特徴や，開発コスト，営業方法などを検討した．そのときに真っ先に決まったことは，次の３点である．

(1) ユーザにとって価値のある商品にすること
(2) 短期間，低コストでの製品化を行うこと
(3) リリース時期は 2007 年 4 月にすること

２．２　プロジェクトの問題点

プロジェクトの発足に向け，市場性，実現性，将来性，採算性等を詳細に詰めると，直ぐに様々な問題が浮き彫りとなった．特に本プロジェクトは，経営的にみても市場性の面からも，リリース時期は厳守する必要があった．その時点で作成した開発スケジュールは，

表１のとおりである．

しかし，準備期間の少ない状態でスタートした結果，プロジェクトが発足してわずか１月後の６月末には様々な問題が明らかになり，開発の目処が立たないまま，１ヶ月が過ぎていた．このまま，続けても，2007 年 4 月のリリースはできないと判断し，プロジェクトを中断し，開発プロセスの見直しを行うことを決意した．

表１．「アジャイル開発プロセス」要素 注入前の開発スケジュール

No.	工程	2006年							2007年			
		6月	7月	8月	9月	10月	11月	12月	1月	2月	3月	4月
1	基本設計	☆	中断									
2	詳細設計											
3	プログラミング											
4	結合テスト											
5	システム･テスト											
6	完成											★

そこで，浮かびあがった具体的な問題点は，次のとおりである．

２．２．１　ニーズの多様化の問題

今回のプロジェクトは，ユーザの利益を最優先する．言い換えると，ニーズの多様化への対応である．プロジェクト発足から直ぐに競合商品の調査を開始し，将来，顧客となるユーザの協力を頂き，今回開発する商品に，利用者として何を期待するのかの，ヒアリングを実施した．開発の主要メンバーの志は高く，積極的に様々な要件を抽出した．

検討を行うたびに，製品の構成は膨らんでいき，商品の骨格が固まらず，遂には，当初決定していた要件までも変更せざるをえず，収拾がつかない状態にまで陥った．

２．２．２　工期の問題

ただでさえ９ヶ月後のリリースは，タイトなスケジュールであった．そのうえ，上記「ニーズの多様化の問題」で要件はまとまらず，なかなか開発プロセスの本格始動ができないまま，基本設計工程の 50%を消化してしまった．

2007 年４月のリリースを実現するためには，さらに，タイトなスケジュールをこなす必要があった．

２．２．３　品質低下の問題

一般的に，近年のシステム開発では，品質低下の傾向があるといわれている．当社でも

同様に，新しい技術の導入と短い工期，若い技術者のモチベーション等の複合する原因による品質の低下は，潜在的な問題になっていた．

当然，本プロジェクトでも，同様の問題が予想されたが，有効な対策は打てていなかった．

２．２．４　要員育成の問題

今回の新商品のアーキテクチャは，市場性から WEB アプリケーションとすることは前提として決まっていた．さらに，一般的な HTML ベース WEB アプリケーションでは，商品としての競争力を欠くため，Ajax などのリッチなインタフェイスを用いた実装を行う．また，インタフェイスに限らず，価格的競争力を高めるために，オープン・ソース・ソフトウェアの積極的な活用を目指した．

しかし，本プロジェクトに充当されるメンバをみると，前述の技術を既に習得し使いこなせる技術者はほんの数名であった．開発規模に対して十分な人数ではないことから，プロジェクトのメンバ内に，開発を進める中で与えられた仕事をこなしていけるのかという不安も広がっていた．

２．３　「アジャイル開発プロセス」の導入

「アジャイル開発プロセス」という言葉は，数年前から耳にはしていたが，「つかみどころのない，夢のような話」といった先入観が，この開発プロセスを遠い存在にしていた．

「アジャイル開発プロセス」とは，「アジャイル（俊敏）」にソフトウェアを開発する方法論の総称である．具体的には，「エクストリーム・プログラミング(XP)」や「スクラム(Scrum)」，「クリスタル(Crystal)」など，様々な方法論が提唱されている．前述の具体的な各方法論を取り上げた書籍の多くは，実装プロセスを中心に取り上げている．このことから，これらのプロセスを導入して，上流工程から適用しても到底よい結果を得られないし，また，紹介された実装プロセスにおける管理手法が，これまで行ってきた管理手法とあまりにも大きく違うことから，現場の混乱を招くと感じていた．

その後，有限会社メタボリックスの山田正樹氏がまとめられた「アジャイル PMBOK」（以下，APMBOK という）を知った．「APMBOK」は「PMBOK」の提供するフレームワークを借りて，アジャイル開発プロセスのプロジェクト管理に関わるプラクティスをまとめたものである．基となる PMBOK のフレームワークが「本来は建築など，モノ作りプロジェクトのプロセス（工程）」のプロジェクト管理に関する知識体系であることにから，プロセス導入後の活動をイメージできる．このとき初めて，これなら我々が実践してきたウォータフォール型開発スタイルからも，無理なく移行できるかもしれないという思いが生まれた．

当初，採用するに至らなかったが，プロジェクトの中断を余儀なくされた時，直面している問題を解決し何とかプロジェクトを進めるべく，山田氏を招いた研究会を行った．その結果，この「APMBOK」のプラクティスにそって，プロジェクト管理を行うことを決断した．

表２が，計画見直し後の，設計および実装を繰り返す「APMBOK」のプロセスに則った開

発スケジュールである.

表２．「APMBOK」導入後の開発スケジュール

No.	工程	2006年						2007年			
		7月	8月	9月	10月	11月	12月	1月	2月	3月	4月
1	開始	→									
2	計画		→								
3	設計中心のイテレーション			→	→						
4	実装中心のイテレーション				→	→	→	→	→	→	
5	システム･テスト									→	
6	完成										★

３．当社における「アジャイル開発プロセス」

３．１　本プロジェクトへの「アジャイル PMBOK」の適用

先に断っておくが，本プロジェクトでは，エクストリーム・プログラミング（XP），スクラム（Scrum）に代表されるアジャイル開発プロセスのどれかを採用したということではなく，APMBOK のフレームワークにそったプロジェクト管理を実践することに注目し，前述のプロセス群の中に定義された用語・役割などについて直感的に理解できるものを採用している.

当社はこれまで，一部スパイラル的要素も取り入れてはいたが，ウォータフォール型開発を中心にプロジェクトを遂行してきた．しかし，中断の時点で直面している問題点（「２．２．　プロジェクトの問題点」を参照）を考えると，アジャイル開発プロセスは，課題を実現するための開発プロセスとして，様々な部分がよくマッチしていると感じた.

アジャイル開発プロセスのフレームワーク「APMBOK」を導入することにより，机上の空論になりやすい要求固めに無駄な時間を取られることなく，実装（プログラミング）の工程を多く確保することができる．さらに，実装時間に時間をとることができるということは，実装メンバが，技術やニーズに関する学習をしながら，プロジェクトを進められる.

フレームワークとして「APMBOK」を適用するにあたり，本プロジェクトの各プロセスの目標を以下の通りに設定した.

(1) 開始プロセス

- ビジョンを明確にする
- 成果物の特徴を明確にする

(2) 計画プロセス

- スコープを明確にする
- 活動計画を明確にする
- 品質計画を明確にする
- 要員調達（教育）計画を明確にする
- リスクを明確にする

(3) 実行プロセス・グループ

- 品質保証を行う
- チーム結成をスムースに行う

(4) 制御プロセス・グループ

- スコープ変更を制御する
- スケジュールを制御する
- 品質を制御する
- リスクを制御する

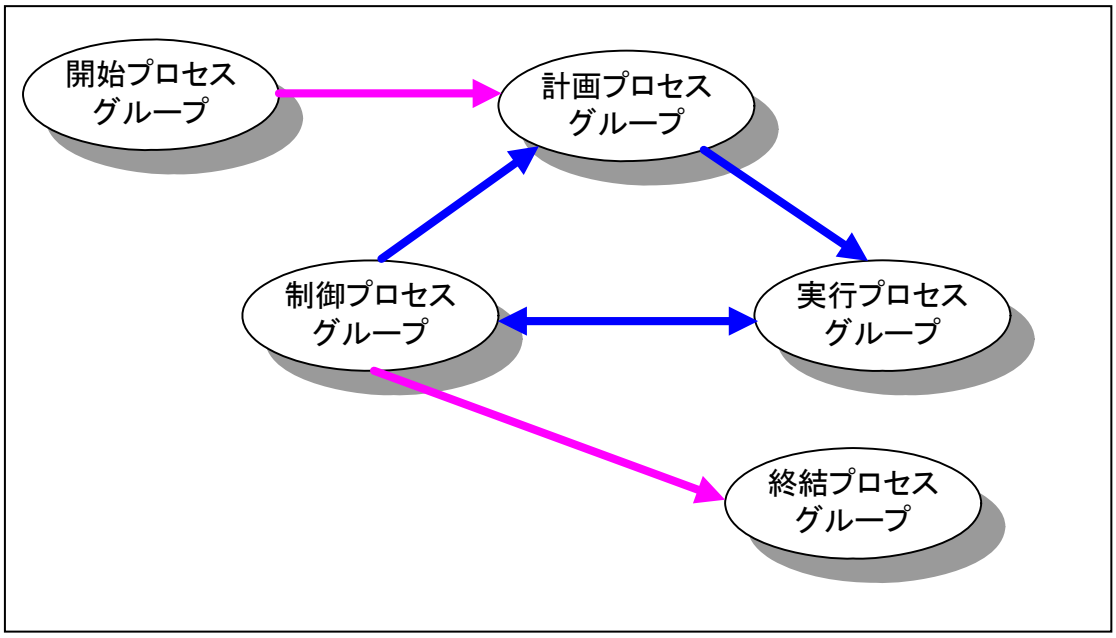
(5) 終結プロセス・グループ

- プロジェクトを完了させる

当たり前のことであるが，各プロセス・グループは，システム作りの手順であるため，ウォータフォール型開発の開発工程となんら変ることはない．但し，「アジャイル開発プロセス」の特徴的なことは，計画・実行・制御プロセスの各プロセスのイテレーション（繰り返し）を前提とすることである．

図１は，今回の各プロセス・グループの全体像である．

図１．プロセス・グループ全体図



現在は，最初の計画プロセスを経て， 実行プロセスに入った段階である．図１に示すと

おり，計画プロセスグループは繰り返しの中に含まれるため，ウォータフォール型の完了と同意ではない．今後，計画，実行，制御を繰りかえしながらプロジェクトを進めていく．

以下では，「開始プロセス・グループ」と最初の「計画プロセス・グループ」での活動と，そこで認められた「アジャイル開発プロセス」の効果を挙げる．

３．２　開始プロセス・グループ

APMBOK の開始プロセス・グループには，プロジェクト・フォーミュレーションとプロジェクト・ビジョンというプラクティスがある．

３．２．１　プロジェクト・フォーミュレーション

プロジェクト・フォーミュレーションは，あとに続くプロジェクト・ビジョン作成のためのプレ作業として位置づけられる．これに参加するのは，プロジェクトのビジョンを定義しミッションを創造できる力量のあるメンバ少人数のみとし，ビジョンを立案できる程度の理解が得られるまでの情報収集を速やかに行う．

本プロジェクトではプロジェクト・フォーミュレーションの参加メンバを３名に絞り，APMBOK の原則に従い，詳細な目標設定及びスケジュールは作成せず，定期的にミーティングを行うことによって作業を進めた．各メンバは，次回ミーティングまでの目標を自分なりに設定し，その結果をミーティングで報告するのである．技術的な検証や，性能検証のためのデモ・アプリの作成や，成果物の特徴の抽出，パンフレットの元となるような資料の作成，顧客へのヒアリングなど本当に様々な内容である．

このプラクティスで特筆すべきは，個々人の作業の成果物を求めない点である．このことは，成果物を作成することを目標としないため，本来の目的の達成にむかって情報収集に注力できる効果があった．

実際の作業期間は，７月の１ヵ月間（30 日弱）であった．

３．２．２　プロジェクト・ビジョン

プロジェクト・ビジョンでは，プロジェクトの将来像をメンバー全員が持つことを目標にする．その内容は以下の通りである．

(1) プロジェクト・ビジョン
- 本プロジェクトのビジネス上の目的
- 本プロジェクト（製品）の市場での位置づけ

(2) 製品ミッション及び仕様概要
- ビジネス上の有利点
- 製品の有利点

- 製品概要

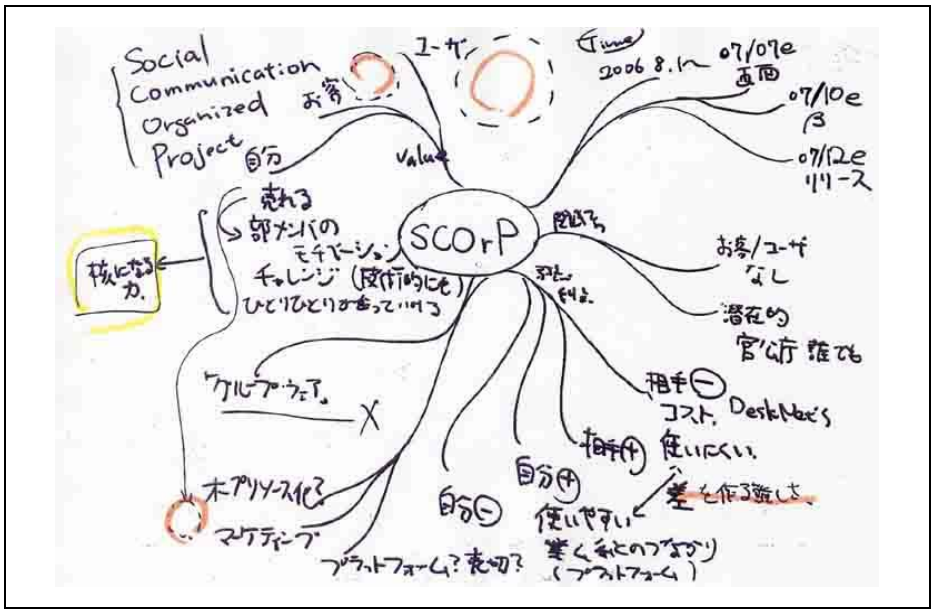
・　システム仕様概要

プロジェクト・フォーミュレーションに参加した３名が，収集した情報を持ち寄り，(1)プロジェクト・ビジョンを作成した． 図２は，プロジェクト・ビジョンを検討中のホワイトボードの内容である．引き続き，(2)製品のミッション及び仕様概要の一次案策定をおこなった．この時点で可能な，最も具体的な成果物のイメージを作り上げ，３名の共通認識とした．

プロジェクトの参加予定メンバ全員が集まり(2)製品のミッション及び仕様概要の一次案を示し説明を行い，加筆・訂正した．集まった目的は，本プロジェクトの目的を共有するためである．ちなみに，この日にプロジェクト発足式の飲み会を，上司も交えておこなった．この結果，メンバー全員に，新製品への社内での期待感を感じさせることができ，さらに目標について合意を得た上で，開発へ参加する自覚が芽生えた．

また，新しく作ったスケジュールにそって，本プロジェクトの開始プロセスグループを終了し，今後への手がかりを得た．

図２．プロジェクト・ビジョンの検討中の様子



３．３　計画プロセス・グループ

３．３．１　スコープの定義と計画

「APMBOK」では，開発するソフトウェアの対象範囲（スコープ）を，ユーザにも理解できるようにストーリとして定義する．ストーリは，旧来のソフトウェア開発では要件に対応し，「機能」「品質」「性能」の要件を含んでいる．計画プロセスにおいては，詳細な

レベルではなく概要＋α程度であるが，重要なストーリをできるだけ漏らさず抽出すると言った視点が大切である．本プロジェクトでは，「機能」「品質」「性能」について，本に書かれた事の確認と共に当面対応できないことを明白にするため、以下の通りに定義してメンバーの共通認識とした．

「機能」は，ユーザが直接関わる機能と，直接は関わらないがユーザに価値を与える機能の大きく２つに分け，これらについて検討する．「品質」は，プロジェクト全体に対しての品質である．実現する機能は，プロジェクトを進めるに中で，ユーザにもたらす価値を高めるという視点で「何を最優先するか，次に何を優先するか，何を犠牲にするか」といった優先度を付けながら変えていく．このため，ソフトウェアの品質は，最初に挙げたすべての機能を実現することよりも，選択したユーザにとって重要な価値を持つ機能を，限られた期間の中で完全に実現することで確保する．処理速度などの「性能」は，現在，リソースの増強を行うことにより，後から何とかできることが多い．よって，本プロジェクトの開始時点では，性能確保のための議論は保留とし， 目標としてあげる性能を列記するに留める．但し，予想される検討項目である OS，ミドルウェア，開発言語等これで解決できる問題については，必要な時期にスケーラビリティのあるアーキテクチャを検討することとした．

アジャイル・プロジェクト管理では，基本的にスケジュールを固定してスコープを変化させる．スコープの取り扱いに於いて，最も特徴的で理解しにくい点は，「スコープは，状況や変化を無視して固定するべきものではない．日々変わりいくものである．」ということである．ここが，旧来のプロジェクト管理とは大きく異なる．

本プロジェクトのリリース時期は決定事項であり，工期の延長はできない．このため，スコープの変更を受け入れなければならない事態になったとき，工期は固定して，優先度の高いスコープは取り入れ優先度の低いものは削除することで，スコープに柔軟性を持たせ全体の品質を確保することを決めた．アジャイル・プロジェクト管理の本質を，守って行こうという決意である．開発プロセスが開始したばかりの時点では，情報共有のためのミーティング行うこととし，そのサイクルを一週間とした．

今回開発する商品のスコープの定義変更は，ミーティングで集まった情報と，開発に協力していただける協力者の意見をとりまとめ検討する．ユーザの視点と製品戦略の視点から，ストーリの変更に責任を持つ製品オーナを決めた．製品オーナは，代表的なアジャイル開発プロセスであるスクラム（Scrum）に定義された，要求の管理を行う役割りである．変更のタイミングは，今のところ未定でルール化されていない．ただし，最適な時期を見定めることとし，１つのイテレーション（３．３．２実行の定義と計画で説明）実行中は，チームにその対応を強制しないことのみ決定している．1 つのイテレーションは最長２週間で，その終了後，再度調整を行う

３．３．２　活動の定義と計画

「APMBOK」では，スコープを実現する活動をタスクとして定義する．タスクには，ストーリを実現するためのものと，それ以外の開発環境の整備なども含まれる．ストーリに関係するタスクは，ストーリと関係がわかるように記述しておく．

以下に，本プロジェクトで行った「ストーリの優先付け」「ストーリの依存性の抽出と

活動ネットワークの作成」「タスクの抽出とストーリとの関係付け」「繰り返し（以下，イテレーション という）」について述べる．

８月に入り，ストーリが出揃ったと判断した時点で，「ストーリの優先付け」を行い，「ストーリの依存性の抽出と活動ネットワークの作成」を行った．結果，依存性を解決した形の「活動ネットワーク図」が完成しメンバに通達した．同時に，このドキュメントの内容は変更される可能性があるため，決定事項としてではなく，動的に変更していくことも伝えた．最新版は，変更に対応するため，紙の資料とはせず「OpenOffice．org 図形描画」のドキュメントとしてサーバで閲覧可能とした．

ストーリの優先度の決定がなされ関連が明らかなると，ストーリに依存するタスクの優先順位が明確になり，誰がそのタスクに責任を持つかを決定していく．

「イテレーション」という表現は，適当な期間内に，実際に動くものを作る作業の分割単位としても用いられる．本プロジェクトでは，１回のイテレーション期間を，１～２週間程度の作業量とし，タスクやストーリを割り当てる．ここでの成果物は，評価できるものでなければならない．イテレーションの期間を決定するにあたり，一部「３．３．１．スコープの定義と計画」の作業に手戻りが発生した．長すぎるストーリは，イテレーションを何度も繰り返さなくてはならず検証が遅れてしまうため，より理解しやすい現実にそった管理ができるように，１ヶ月以上の開発期間を必要とするストーリの分割が手戻りの主な作業であった．

３．３．３　品質の定義と計画

各ストーリに対しての「品質」は，「機能性」「信頼性」「使用性」「効率性」「保守性」「移植性」で示される．以下が，代表的な検討項目である．

「機能性」
- そのストーリは，目的を達成できるか
- そのストーリは他とどのような方法で連携するのか
- 連携先のストーリにどのような機能を提供するか
- 連携先のストーリで障害が発生した場合にどのように振舞うのか
- セキュリティ機能をどこまで要求するか
- etc．．．

「信頼性」
- もし，対象のストーリ内部で障害が発生した場合，どのように回復するか
- そもそも自力で回復する必要があるのか
- 障害を一括して管理するストーリは必要ないのか
- etc．．．

「使用性」（ユーザ・インタフェイスが中心）
- 直感的な実装となっているか
- 操作性に無理がある実装になる可能性はないか
- etc．．．

「効率性」

・実行効率（処理速度など）をどこまで求めるか
・資源効率（データ容量など）をどこまで求めるか
・etc．．．

「保守性」

・保守（変更）する可能性はどの程度か
・保守（変更）を容易にするためにどのような対策を行うか
・デザイン・パターンは何を適用するか
・etc．．．

「移植性」

・稼動環境に依存していないか
・環境依存部分が各所に散らばっていないか
・各種規格への適用に無理はないか
・etc．．．

ストーリ毎に，上記を検討する中でも，一部手戻りは発生したものの，プロジェクト・ビジョンが明確になっており，また，各ストーリには優先度をつけているため，思いのほか作業はスムースに進んだ．この中で，幾つかの優先度の低い検討事項は，当面の作業を行うために必ずしも決定する必要がないため，最終的な検討は行わず，未決定事項として残した．

３．３．４　要員調達（教育）計画

APMBOK では，役割を決定する方法が，これまでのプロジェクトとは大きく異なる．今までは，プロジェクト・リーダ，設計担当，実装担当，テスト担当 etc．．．と，各役割を固定してきた．これに対し「アジャイル開発プロセス」では，「できるだけ最初から役割を固定しない．創発的に役割が生まれ，担当が決まっていく」としているからである．

この件では，中心的なメンバの意見がわかれた．「アジャイル開発プロセス」の導入に懐疑的な人からは，「個々人の役割を決めないということは，個々人の責任の範囲がはっきりしない」という意見が出た．これに対し推進派からは，「役割を限定することが，縦割りの構図を生み，無関心になっていく場合がある．未成熟で何が起こるか分からないプロジェクトでは，各自が自分の得意な面を活かしてチームに貢献できるように，プロジェクト力を高める管理が必要だ」という意見がでた．この対立の同意を得るには，時間をかけてじっくり検討しなければならなかった．検討した結果，「アジャイル開発プロセス」のスタイルを適用することとした．その理由は，次の３点である．

(1)役割を固定しないことは「アジャイル開発プロセス」の肝である
(2)これまでの方法論ではプロジェクトを進められないという結果がでている
(3)工夫することにより，そのリスクは抑えることができる

リスクを抑える工夫としては，以下のこと８点を実践することとした．

(1) プロジェクト内を小さなチームに分割する
(2) 小さなチームは一定期間（１ヶ月程度）で再構成する
(3) 仕事は個人ではなく，チームに任せる
(4) ペア・プログラミングを積極的に行う
(5) チームごとに毎朝 10 分程度のスタンダップ・ミーティングを行う
※ 図３は実際にスタンダップ・ミーティングを行っている様子
(6) スタンダップ・ミーティングで今日やることを確認し，各人毎にホワイトボードに記入し可視化する
(7) 一週間に一度，短時間のプロジェクト会議を行う
(8) プロジェクト・マネージャはこれまでのように配置し，全体を統括する

図３．スタンダップ・ミーティングの様子

３．３．５　リスクの抽出とリスク対応計画

リスク管理こそが，プロジェクト管理だといっても過言ではない．よってリスク管理をどのように進めるかが，本プロジェクトの成功の鍵を握る． そこで，「リスク管理」を以下の４つのステップに分解した．

(1) リスク抽出（発見）
(2) リスク分析
(3) リスク対応計画
(4) リスク対策

本プロジェクトにおいて，リスクを抽出するために，メンバがコミニュケーションを円滑に行い，同じ問題意識を持つことで，個人が問題を抱え込まないような体制をとることとした．具体的には，３．３．４要員調達（教育）計画に挙げた８つの工夫に記述したように，ペア・プログラミング，チームのスタンダップミーティング，一週間毎に行うプロジェクト会議などでの問題の共有，ホワイトボードでの「今日やること」の可視化等の日常の活動で実現する．また，タスク単位で考えられるリスク項目を洗い出し，「リスク・チェック・リスト」を作成した．これにより，経験の少ない技術者には，リスクの発見について考える機会を与え，リスクの発見の能力育成につながる事を期待している．

「リスク分析」は，プロジェクト・マネージャが中心となり，製品オーナとチーム・メンバ及びリスク範囲が複数のチームに及ぶ場合は，対象のチーム全員を集めその影響範囲と発生する確立を分析し，この結果と，既存の優先順に並べられたリスク一覧から，対応するかどうかを協議する．

ほとんどの場合，「リスク対応計画」は，リスク分析の過程で既に明白になっていることが多いと思われる．具体的な事柄も含めチーム・メンバが対応計画を策定するが，軌道修正とチームでは解決できない場合は，プロジェクト・マネージャも参加することにしている．

「リスク対策」は，ひとつのイテレーションとしてまたはイテレーションの一部として，チームで対策を講じる．

できる限り，プロジェクト全体，各チーム，個々人間で，良好なコミニュケーションを保ちながらプロジェクトの現在の状況を可視化し，情報の共有化を行うことを実践し，リスクの早期発見，早期対策を行いたいと考えている．

４．APMBOK の効果

４．１　問題克服の過程

まず，第一の効果として，厳しい工程のなか，ニーズの多様化の問題で，スコープが定義できず頓挫していたプロジェクトのスケジュールに，大枠ではあるが，完成へのレールを敷くことができた．アジャイル開発プロセスは，スコープの変更の受け入れを前提とするため，ウォータフォール型プロセスのように，各工程の成果物の呪縛（全てが明白でなければ先に進めない強迫観念）はなく，開発に関わる主要なメンバーには，初期段階のスコープが完全に定義できない場合でも，プロジェクトを進めることができる効果がある．また，参加するその他のメンバには，途中の検証で明らかになったいろいろな事情も含めて変更の受け入れながら，限られた工期の中で，最もユーザに利益のあるソフトウェアを作り上げることが共通理解されているため，先の見えにくいプロジェクトで開発することの不安感を低減できる効果も挙げられる．

さらに，「開始プロセス・グループ」では，プロジェクトのビジョンやミッションを明らかにし，メンバに周知徹底することを行った．これにより，全てのメンバが，プロジェクトの目標・目的・将来が明確に理解でき，これまでのような「作業」ではなく「自分の

仕事」という意識が芽生えた．不足する技術について，あらかじめ予想できるため，前もって情報を収集したり，支援を求めることが日常化しつつある．各メンバが積極的にプロジェクトに参加するようになり，一回り逞しくなった．プロジェクトは活性化し，無機質なプロジェクトから，活きたプロジェクトに変わってきたと感じている．

まだ開始して日は浅いが，今後の「実行プロセス・グループ」「制御プロセス・グループ」の各プロセスでも，よい成果が期待できる．

「計画プロセス・グループ」では，「APMBOK」のプラクティスに沿うことで，無理なく円滑なコミニュケーションを保ちながら計画ができた．前述のとおり，各メンバは与えられたタスクの実行について，具体的なイメージを最初にもつことができるため，実行プロセスをスムースに，そして，確実に遂行することができた．毎週のミーティングの中で，他のプロセスの再検討の必要性が自然と浮かび上がってくるのである．これを今後も継続できるように，努力しなければばならず，この繰り返しが，品質の低下のリスクの軽減に繋がると感じている．

結果として，プロジェクトのスコープを実現可能なものとして再定義しながら，プロジェクトの「品質」は今も向上し続けているといえる．

上記の事柄は，プロジェクト企画から４ヶ月の中間評価として，十分な成果（効果）を生み出したと自負している．

５．今後の課題

今現在，中間評価の段階である．「実行プロセス・グループ」をどのように成功させるか，つまり，チームに参加する各メンバが，チームに与えられた領域をどのように達成するのか，また，他のチームとどう連携していくのかが，本プロジェクトの成功を左右する．今の段階で，今後の開発プロセスにおいてプロジェクトマネージャーが，プロジェクトにどう関わっていくかを考えるとき，以下の３項を掲げることができる．

(1) チームの意志を最優先する
(2) 品質の優先度が下がった場合は，それを軌道修正する
(3) チームと共に過ごす

プロジェクトマネージャーは，「制御プロセス・グループ」では，プロジェクトの成功に向け全力を尽くす．冷静な立場でプロジェクト，ときにはチーム（メンバ）を評価し，プロジェクトにとっての最善なレールを引く役目が待っている．

ソフトウェアはあらゆるところで必要とされ， 社会インフラとしての重要性を増している．ある意味，当たり前の存在となりつつあり，システム開発はユーザの視点で，柔軟な対応がもとめられている．この流れは今後，留まるどころか，拍車をかけ進んでいくことであろう．このような状況の中，我々は本プロジェクトを成功させ，商品開発に留まらず，

システム開発全般の礎を築いていきたい.

６．おわりに

前述の通り，「アジャイル開発プロセス」は様々な方法論の集合体であり，この方法論がアジャイルであるという明確な定義のないプロセスであるため，立ち上げ時にはまさに手探りの状態であった．プロジェクトとして，製品のビジョンやミッションが周知徹底されないまま，メンバーが作業を開始して収拾がつかないなどの混乱を経験するといった期間があったものの，今は，「アジャイル開発プロセス」導入による効果には確かな手ごたえを感じることができている.

特に，今回のように特定のユーザが決まっていないパッケージ開発にとって，品質として定義された「機能性」「信頼性」「使用性」「効率性」「保守性」「移植性」を保つために，イテレーションを行う勇気を持ち製品を作り出すことが，本プロジェクトの課題である，短期間，低コスト，オープン，エンドユーザ指向を実現する鍵であると感じている.

今回の発表は，参加しているメンバにその効果をどう評価しているか，今後期待していることを知ってもらうことにある．ホームページに記載された，アジャイルアライアンスの原則では，「どうしたらチームがもっと効率を高めることができるかを定期的に振り返り，それに基づいて自分たちのやり方を定期的に調整します．」と結んでいる．実際，前述の原則を踏まえ，要員のモチベーションを保ちながら開発は進んでおり，来年４月にはプロジェクトは完了する予定である.

参考文献

［１］実践アジャイル　ソフトウェア開発法とプロジェクト管理
著者　　：山田 正樹
出版社　　：ソフトウェア・リサーチ・センター

［２］アジャイル開発手法FDD―ユーザ機能駆動によるアジャイル開発
原著　　：Stephen R. Palmer, John M. Felsing
著者　　：スティーブン・R. パルマー, ジョン・M. フェルシング
翻訳　　：今野 睦, 長瀬 嘉秀, 飯塚 富雄, デュオシステムズ
出版社　　：ピアソンエデュケーション

［３］スクラム入門-アジャイルプロジェクトマネジメント
著者　　：ケン・シュエイバー
出版社　　：日経BPソフトプレス

［４］初めてのアジャイル開発　～スクラム，XP，UP，Evoで学ぶ反復型開発の進め方～
著者　　：クレーグ・ラーマン
翻訳　　：ウルシステムズ株式会社，　児高 慎治郎，　松田 直樹，　越智 典子
出版社　　：日経BP社

［５］実践CMM―インフォシス社におけるソフトウェア・プロジェクトのプロセス
原著　　：Pankaj Jalote
著者　　：パンカジ ジャローテ
翻訳　　：中村 直司，　NTTデータCMM研究会
出版社　　：ピアソンエデュケーション

［６］メタボリックス社ホームページ
「ウォータフォール型開発プロセスからインクリメンタルにアジャイル・プロセスに移行する」
http://www. metabolics. co. jp/mmw/executable-knowledge/agile-processes/wf2ap