

インクリメンタル開発による既存製品バージョン・アップ

森 茂子[†]

日本アイ・ピー・エム株式会社ソフトウェア開発研究所[†]

1. はじめに

開発期間の短期化に伴い、Agile 開発手法¹に見られるような反復型のインクリメンタル開発が注目されている。しかし、旧来のウォーター・フォール手法で開発されたソフトウェアに対するバージョン・アップの場合、あえて実績のある従来手順からインクリメンタル開発へ変更するべきであろうか²。筆者らのグループは幾つかのレガシー製品の保守とバージョン・アップ開発を行っている。2003年にある製品のバージョン・アップにおいて、設計・プログラミング・テストでインクリメンタル開発を行い同程度の生産性ながら品質面で従来を上回った。当事例を紹介し、その長所・短所と、どのような場合インクリメンタル開発を選ぶべきか考察する。

2. インクリメンタル開発採用

2.1. インクリメンタル開発採用の経緯

当バージョン・アップでは新機能の提供に加え、大幅なパフォーマンス・アップも目的であった。ウォーター・フォールではパフォーマンス改善効果は開発後期のテスト・フェーズに測定可能となるが、インクリメンタル開発は早期から測定可能である。また、一般にインクリメンタル開発は十人以下のグループに適していると言われている。我々のグループはこの条件を満たしていた。これらよりインクリメンタル開発を採用、以下の点を検討した。

- 反復範囲
- 開発要素と順序
- 進捗管理
- 品質と生産性

2.2. 反復範囲の決定

まず、どのフェーズを反復で行うか検討した。パフォーマンスを少しずつ改善して何度もリリースするよりも、大幅に改善した物をリリースした方が好ましいと考え、設計・プログラミング・テストのサイクルで反復し、リリースは一度とした。この方法は既に1990年代にクリーン・ルーム開発手法等でも提案されている³。

2.3. 開発要素の定義

次に検討したのは、反復の最小単位である開

発要素とその順序の決定である。我々は、機能とパフォーマンス改善の二つの側面で数十の開発要素に分けた。これらに重要度と改善効果で優先度を与え、依存関係を考慮し順序を決定した。

2.4. 進捗管理

数十の開発要素を個々に進捗管理するのは困難と考え、先に決めた開発順序を3つのフェーズに分類した。主要新機能、効果の大きいパフォーマンス改善、その他である。この3フェーズで管理することとした。

2.5. 品質と生産性

当バージョンでは前バージョンと同程度の生産性で品質の向上を目標とし、その実現のため、以下の方針をたてた。

設計文書の簡潔化・・・設計文書の作成・更新にかけるワークロードを軽減する。

- 外部設計・・・初めは概要のみに留めフェーズ毎に詳細化する。
- 内部設計・・・開発要素毎に実施。設計文書は簡潔に記述する。

テストの効率改善

- 単体テスト・・・新規や変更量が大きいプログラムのみ実施。
- 機能テスト・・・開発要素毎に実施。設計文書と単体テストの簡潔化により、プログラムを早期に稼動しテストを実施することで、欠陥の早期検出を図った。
- 自動化・・・リグレッション・テストを自動化、フェーズの終わりに実施した。
- 最終テスト・・・第三フェーズ後にウォーター・フォール開発と同様に機能テスト、シ

ウォーター・フォール

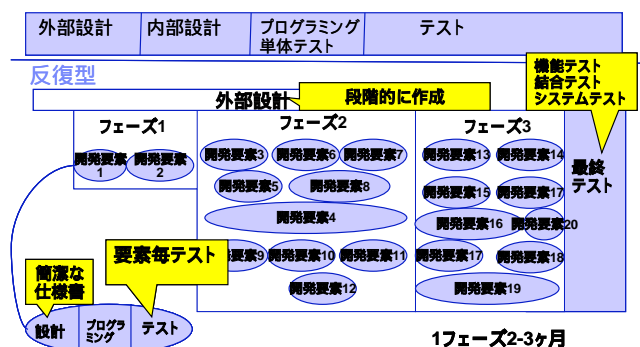


図1. 開発手順の比較

¹ Legacy Product Enhancement by Incremental Development

[†] Shigeo Mori, Software Development Laboratory - Yamato (YSL)
IBM Japan, Ltd

ステム・テストの期間を設けた。
当事例とウォーター・フォールの比較を図1に示す。

3. 効果と問題点

3.1. 変更への柔軟性

第二フェーズ中のパフォーマンス実測で一部目標に達していなかったため、改善項目を追加した。また、追加の機能拡張の要求も発生したため、第三フェーズの優先度の低い物を幾つかとりやめ、新しい開発要素を追加した。このように、変更柔軟に対応可能というインクリメンタル開発の利点が発揮された。

3.2. 品質と生産性の評価

図2に最終テスト中の欠陥数成長曲線を示す。これより欠陥発生件数が収束している。

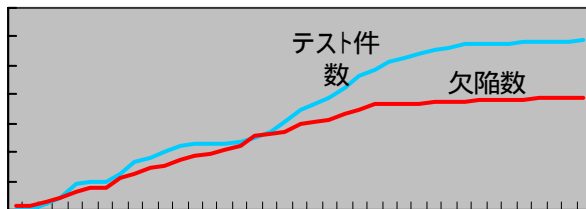


図2. テスト実施数と欠陥累積

更新プログラム・サイズ	0.9
総プログラム・サイズ	1.7
更新サイズあたりテストケース数	2.0
更新サイズあたり欠陥数	0.4
出荷後3年間の更新サイズあたり欠陥数 ¹	0.4

前バージョンとの品質面での特性比較を表1に示す²。欠陥数の減少の要因として、第一から第三フェーズ中のテストで、従来の単体テストよりも多くの欠陥を検出済できたと考えられる。欠陥数が少なかったため、従来よりも短時間で2倍のテスト数を実施できた。出荷開始後の欠陥予測数も少ない。以上より前バージョンよりも出荷後の高品質を期待できる。一方、生産性は両バージョンで同等であった。予め検討した設計文書の簡潔化とテストの効率化に加え、開発要素毎の内部設計とテストは設計作業と欠陥発生箇所の特定を容易にしたことが一因に挙げられる。

3.3. 進捗管理の問題点

当初は週一度の進捗会議を実施したが、フェーズ途中の状況把握が困難であった。そこで、開発メンバーが誰でも更新できる文書を用意し、

各自で進捗を記入するようにした。

3.4. プログラマの感想

開発終了後のプログラマの感想では「インクリメンタル開発は短期で目標があってやりやすかった。」と好意的な意見の反面、「テストケース数が期間に比べて多く完了できるか不安だった。」「単体テストを省略することに不安があった。」という声も聞かれた。今後は当事例を紹介し不安を軽減するよう考慮したい。

4. インクリメンタル開発の選択基準

どのような場合にインクリメンタル開発が適するの考察する。

- ・ パフォーマンス・アップ等システム・チューニング作業・・・本稿にて紹介した通り、インクリメンタル開発が適していると思われる。
- ・ 要件定義や仕様が不確定・・・一般にインクリメンタルに適しているとされている。当事例でも3.1に述べたように、変更が容易である利点が得られた。

- ・ 機能拡張項目が多い・・・開発要素に優先度をつけ、高い物から実装できる点が良い。

ただし、以下を考慮点として挙げておく。

- ・ 進捗管理・・・特に数十人規模のチームの場合は進捗管理の検討が必要であろう。

- ・ 保守性・・・設計文書の簡潔化が保守作業効率に影響する可能性がある。当製品は保守差 f 行中に設計文書を参照していなかった問題無かったが、参照している場合考慮を要する。

5. 結論

ウォーター・フォールで開発された製品のバージョン・アップでインクリメンタル開発を行い、同程度の生産性で従来よりも良い品質を得た。本稿で述べた手法と成果を他のバージョン・アップ開発において参考されたい。

参考文献

¹ アリスター・コーバン：アジャイルソフトウェア開発，株式会社ピアソン・エデュケーション，U.S. (2002)

² IT PRO, システム開発プロセスは、本当にインクリメンタルが主流になるのか？

(<http://itpro.nikkeibp.co.jp/free/ITPro/OPINION/20030516/1/>)

³ 佐藤和夫：“無故障ソフトウェアを開発するための「クリーン・ルーム手法」紹介”、情報処理, pp836-844 (1994)

¹ 前バージョンは30ヶ月間の実績値、当バージョンは図2で示したテスト期間中の欠陥発生数より指数モデルにより予測した値を用いた。

² 図2、表1共に第一から第三フェーズ中のテストのデータは除く。