

アジャイルなソフトウェア開発手法の一提案

田尻 真英 嘉藤 直宏 麻生 隆 天野 敏文 大橋 宏之
富士通コミュニケーション・システムズ株式会社

1. はじめに

昨今の通信インフラの急激な進歩に伴い、通信ソフトウェア製品開発は非常に短期間になり、かつ市場からの要求機能も高度化している。これらのニーズに対応すべく、ソフトウェア開発手法に関しても、高品質を保持しつつ、より生産性の向上を図ることが求められている。近年、アジャイル開発プロセス[*1]が注目されており、その一手法としてエクストリームプログラミング[*2]（以下、XP）を積極的に適用している。

しかし、様々な形態の開発プロジェクトにおいて、開発環境・開発体制等を考慮すると、必ずしもXPで提唱されているプラクティスを効果的に導入出来ないことがある。

本論文では、XPと組み合わせて用いることで、効果のあったソフトウェア開発手法を紹介する。

2. 効果的なソフトウェア開発手法へのアプローチ

今回のGUI処理やCGI処理を含んだ開発プロジェクトにおいて、短期間の開発サイクルで複数製品の出荷時期を厳守するため、効果的な開発体制の構築、生産性の向上及びソフトウェア構造の統一化という観点から考慮すると、XPの『ペアプログラミング』のプラクティスをそのまま適用することは出来なかった。そこで、開発の初期段階に、アーキテクチャの確認とともに、開発機能から共通化できる処理をテンプレートとしてプロトタイピングし、それを各ペアが開発する機能のベースとして利用することとした。

また、開発中における顧客要求事項を早期実現するため、開発する機能の優先度を柔軟に変更でき、ソフトウェア品質を保持するという観点を考慮すると、XPの『短期リリース』のプラクティスについても、開発担当の役割について発展させる必要があった。そこで、ウォータフォール型の開発手法・管理を取り組み、担当する機能を開発工程で区切り、各工程完了時に次の開発担当に受け渡すような開発プロセスの見直しを取り込んだ。

これらのアプローチを基に、次の2つの開発手法を紹介する。

3. ソフトウェア開発手法と効果

3-1. アプリケーション・テンプレート (Application Template)

ソフトウェアを設計する際に、主にスキルの高い開発担当者が、開発の初期段階でGUI処理・プロトコル処理等に共通化設計を実施し、その機能をプロトタイピングで作成し評価する。評価完了後のテンプレートを、他の開発担当者にも『共同所有』させ、それをベースに各機能の個別処理を『ペアプログラミング』等を適用して設計・製造・試験を進める。（図1）

また、ソフトウェア構造についても、GUI処理からCGI処理のインターフェース方法をテンプレートとして提供しているので、シーケンスレベルで共通化が図れる。（図2）

開発途中において、システムとして共通的な問題や追加機能が発生した場合に、同様の手法を実施することによって、発生した問題の解決手順及び機能の追加箇所が共通化されることになる。

この手法を適用することによって、ソフトウェア品質の保持及び生産性向上の効果が得られた。テンプレートはプロトタイピングを用いて作成し、評価されるため、開発初期段階での解決すべきリスクを検証でき、作成したテンプレートのまま『共同所有』できることによりプロジェクト運用に関しても非常に効率よくスケジュール管理が可能となる。また、開発経験が浅い開発担当者に対しては、テンプレートをベースにして個々の処理を追加する過程でノウハウを吸収することが可能となりスキルアップできた。

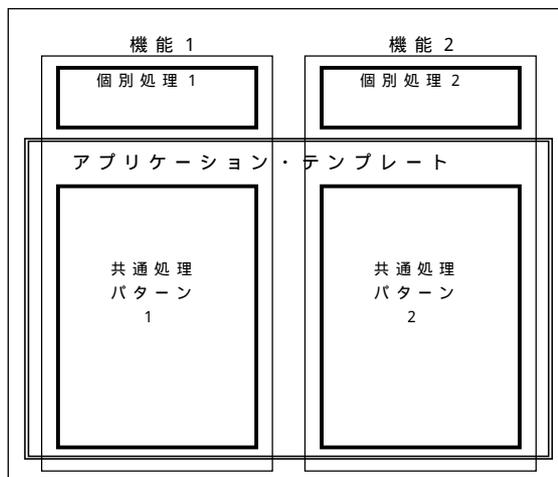


図1. アプリケーション・テンプレート (共通処理)

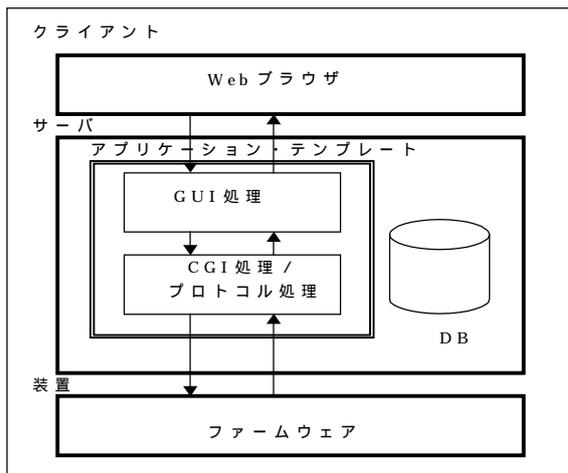


図2 . アプリケーション・テンプレート (シーケンス)

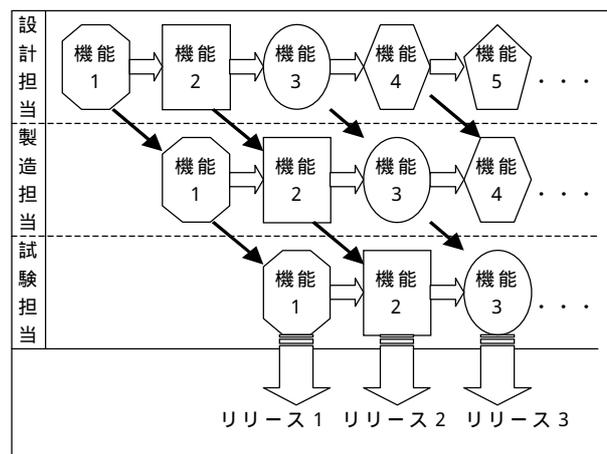


図3 . ファンクション・コンベア

3-2 . ファンクション・コンベア (Function Conveyor)

従来は機能毎に開発者(ペア)を配置して並行開発を進めるプロセスが主流であったが、開発者(ペア)を設計・製造・試験の工程毎に分割し、ソフトウェアの独立した機能毎のスケジュールも分割する。各工程での完了条件を明確にした上で、設計工程が完了した後、製造工程に受け渡して試験を実施する。この手法を『短期リリース』の計画に基づき、優先度が高い機能、または仕様が凍結している機能から着手し、開発機能数分繰り返してソフトウェア開発を進める。(図3)

また、緊急な追加機能のリリース要求時は、開発順序を機能2より追加機能を優先して着手することによって、柔軟な対応が可能となる。(図4)

この手法を適用することによって、工程毎に開発担当が専任するため、設計・製造・試験での要求仕様のチェックが可能となり、品質保持が可能となった。設計工程と試験工程の担当が異なることにより、製造工程よりも前に機能レベルの試験パターン抽出が可能であり、それらを製造工程の担当へ提供することで、コード化作業時においても正常動作を確認できるため、広義の解釈として『テスト』プラクティスを導入したのと同様な効果も得られる。

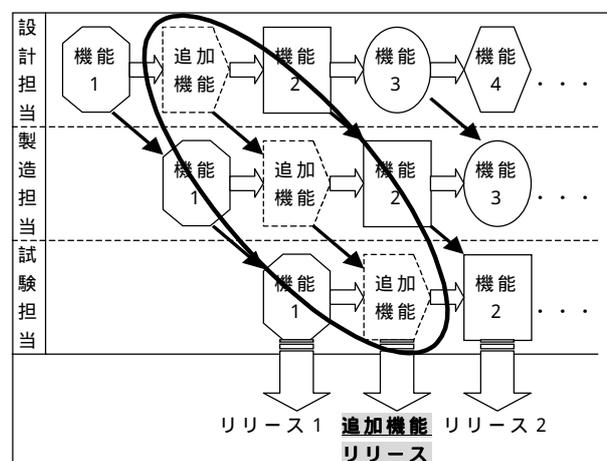


図4 . ファンクション・コンベア (追加機能)

4 . まとめ

本ソフトウェア開発手法は、XPと組み合わせて適用することで、従来と比較して約1.5倍の生産性向上が図れた。今回のように計画立案時の段階にて、ソフトウェア開発手法の適用を柔軟に提案し実践できたのは、開発プロジェクトの特徴を十分に分析・把握することが重要と考える。

今後も開発プロジェクトの特徴によって、効果的に開発プロジェクトの運用ができ、品質の保持、生産性の向上、システム仕様変更への迅速な対応ができるような独自のソフトウェア開発手法を提案する予定である。

参考文献

- *1 : XP & Agile アジャイル開発プロセスとは?
(<http://www.metabolics.co.jp/XP/AgileCommentary.html>)
- *2 : - XPJUG - 日本XPユーザグループ
(<http://www.xpjug.org/>)