6A-1

XP と OSS を連携させたソフトウェア開発法に関する一考察

佐藤　秀樹†　　金子　正人‡　　武内　惇‡　　藤本　洋‡

日本大学大学院工学研究科†　　日本大学工学部‡

１．はじめに

現在のソフトウェア開発においては，開発期間の短縮化や，複数の顧客から受ける多種多様な開発要求などの問題がある．

ソフトウェア開発において開発工程は上流から下流に対する条件を考えるのが一般的である．そこで，下流から上流に向かって顧客を中心に，すなわち顧客からコスト，機能，品質，納期の最適な条件を整理し，条件を満足させるための最適な支援，最適な開発支援システムに対する提案をする．

その手段として，エクストリーム・プログラミング[1]（以下 XP と省略）とオープン・ソース・ソフトウェア[2]（以下 OSS と省略）開発を連携させたソフトウェア開発法[3]を提案し，ソフトウェア開発における後工程引きの開発の実現を検討した．

２．ソフトウェアの後工程引きの考え方

これまでソフトウェア開発で後工程引きを行うパターン分類として，作業プロセスを中心に４つに分類したパターン[4]を検討してきた（図１）．

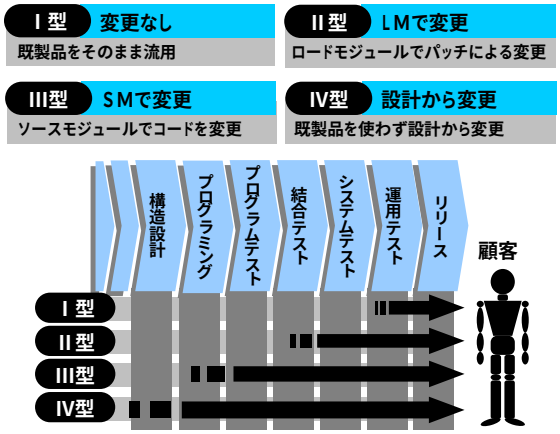

図１．プロセス中心に検討した後工程引きパターン分類

実際に後工程引きをソフトウェア開発で行うためには，プロセスによるパターン分類のみでは顧客に最適な提供を行う事が難しい．この問題を解決するため，部品を取り扱う単位やそれらがどういった状態で管理されているかなど，プロセス以外の複数の視点を用いて下流工程から最適なものを引いてくる方式を検討する．

A Consideration about Software Development Methodology Employing Extreme Programming and Open Source Software
†Hideki Sato
‡Masato Kaneko, Atsushi Takeuchi, Hiroshi Fujimoto
College of Engineering, Nihon University
Koriyama, Fukusima 963-8642,Japan

３．ハードウェア生産に基づくソフトウェア生産の検討項目

後工程引きによる生産を実現しているハードウェアの生産方式を参考に，ソフトウェア開発の特徴を示し，検討項目を明らかにする．

３．１　ハードウェア生産とソフトウェア開発の相違

ハードウェア生産とソフトウェア開発における相違点を比較し，開発法での検討内容を考える．

（１）工程の境界の柔軟性

ハードウェアの場合は前工程の部品を受け取り次の作業を行うが，ソフトウェアの場合は作業者が前工程の作業を行う事も可能であり，工程の境界条件に柔軟性がある．

（２）「生産」と「開発」

ハードウェアの生産は同じものを繰り返し作り出す作業であるのに対し，ソフトウェアの開発は新しいものを作り出す作業である．

（３）量産のコスト

ハードウェアの量産には材料費・部品費・設備の動力費などのコストが大きいが，ソフトウェアの場合は同じものを量産する際にコピー作業で済むため，大きなコストにならない．

３．２　ムダの排除

ハードウェアの生産性の向上のために「ムダ」の排除が検討される[5]．これに該当する項目のソフトウェア開発における適用例を述べる（表１）．

表１．ソフトウェア開発におけるムダ

ハードウェアのムダ	ソフトウェアの適用例
作りすぎのムダ	ユーザに要求されている以上の機能・品質の実現
手待ちのムダ	上流工程の遅れによる後工程の滞留，共同作業者の遅れ
運搬のムダ	特になし
加工そのもののムダ	熟練してない技術者による開発の遅れ
在庫のムダ	不必要なドキュメント，プログラム，データ
動作のムダ	途中での要員交代・作業の引継ぎによる開発の遅れ
不良を作るムダ	バグを内包したコードの再利用や仕様変更，判断誤りによる作業やり直し

３．３　ジャスト・イン・タイム[6]の考え方

ハードウェアの生産ではジャスト・イン・タイムを実現するために，前工程は後工程からカンバンを受け取り，記

載されている必要数の部品だけを後工程に送る．これは需要分だけ生産することにより作りすぎのムダをなくすためである．

ソフトウェアでジャスト・イン・タイムを実現する際にも同様の基本概念が適用できると考える．顧客が必要としている機能の情報を前工程に流し，開発を進める．しかし，ソフトウェアの場合はハードウェア生産で使われているカンバンのような具体的な完成品の指示が困難なため，ソフトウェア開発に適したカンバンと運用の仕組みが必要になる．

4．プロセスと大きさと管理状態による後工程引きの分類

顧客に最適なものを提供する後工程引き開発を実現するため，どの工程でどういう部品を引いてくるか決定できるようにする必要がある．複数の視点として，プロセスの他に各工程で取り扱うソフトウェア部品の大きさの単位と管理状態を用いて分類する（図２）．

（１）開発プロセスによる分類

前述のプロセス分類による４パターンと，部品を用いた開発の仕方から，流用，改造，新規開発の３つに分ける．

（２）機能の大きさによる分類

・AP パッケージ
・組込み関数
・クラス

（３）部品の管理状態による分類

・静的ライブラリ（コンパイルから）
・動的ライブラリ（実行時ルーチン）

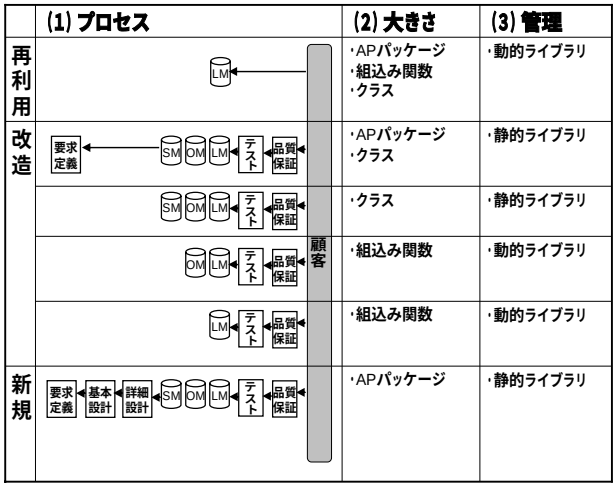

図２．プロセスと大きさと管理状態による後工程引きの分類

5．XP・OSS 連携開発法

後工程引きのソフトウェア開発を実現するために，開発工程を削り作業量を極力減らす開発手法として XP･OSS 連携開発法を提案する．開発手法の仕組みとなる条件を以下に示す．

① ソフトウェア構造が共通部となるアーキテクチャ部分と機能部から構成されている
② パターン化した生産ラインを用いて作業工程が削減できる
③ パターン毎にソフトウェアが部品化されている

これらの条件から XP･OSS 連携開発法では，ソフトウェアの構成を核部と AP 部に機能分割し，全体で大きなインクリメンタル開発の形態をとる．

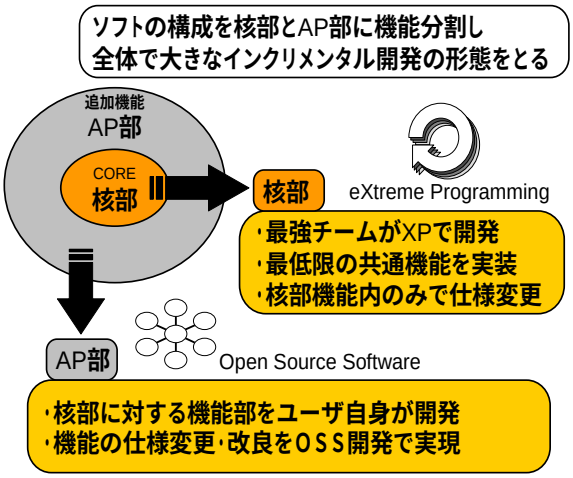

図３．XP・OSS 連携開発法

核部はプログラムの実行制御など最低限の共通機能を搭載したソフトウェアと定義する．対して AP 部は個々のユーザ要求を受けて開発される機能を実装した部分であり，部品として扱う．また核部と AP 部はそれぞれ独立して開発・仕様変更するものとする．

6．後工程引き開発法実現の課題

後工程引き開発法実現の課題を以下に列挙する．

① 後工程引きで引いてくる部品の精緻化
② カンバンの構成とムダを排除する運用法
③ 後工程引きを実現する XP･OSS 連携開発法の支援システム

7．おわりに

本稿では後工程引きに基づく新しいソフトウェア開発を提案し，その検討課題を示し，XP と OSS を連携させて実現するための検討項目を述べた．

今後は，本稿で述べたソフトウェア部品化技術のパターン分類の精緻化を進め，本開発方式がソフトウェア開発に対して有効であるかどうかという検証を行っていきたい．

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 2000
[2] 松下誠，中村祐一ほか:“特集オープンソースソフトウェア”，IPSJ Magazine Vol.43 No.12 2002-12
[3] 味岡，金子，武内，藤本：“XP･OSS を連携させたソフトウェア開発法の提案”，電気関係学会東北支部連合大会 2002-6
[4] 会沢，金子，武内，藤本：”XP･OSS を連携させた後工程引きソフトウェア開発法に関する一考察”，情報処理学会東北支部研究会 2005-1
[5] トヨタ生産方式を考える会：“トヨタ生産方式の本”，日刊工業新聞社 2004
[6] 前田卓雄，橋本隆成:“図解よくわかるソフトウェア･ジャストインタイム”，日刊工業新聞社 2005