



Title	ウィンターワークショップ・イン・金沢報告プロセスと方法論
Author(s)	井上, 克郎; 落水, 浩一郎
Citation	情報処理学会研究報告. ソフトウェア工学研究会報告. 2001, 2001-SE-131(56), p. 83-86
Version Type	VoR
URL	https://hdl.handle.net/11094/50180
rights	ここに掲載した著作物の利用に関する注意 本著作物の著作権は情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

ウィンターワークショップ・イン・金沢 報告 プロセスと方法論

井上 克郎¹

落水 浩一郎

大阪大学基礎工学研究科 北陸先端科学技術大学院大学情報科学研究科

内容梗概

ウィンターワークショップ・イン・金沢の「プロセスと方法論」のグループで議論された内容を中心に報告する。

Report of Winter Workshop in Kanazawa —Process and Methodology Group—

Katsuro Inoue

Kochiro Ochimizu

Graduate School
of Engineering Science
Osaka University

School of Information Science
Japan Advanced Institute
of Science and Technology

Abstract

Winter workshop in Kanazawa was held in Ishikawa prefecture, from Jan. 18 through 19, 2001. In this article, we will briefly summarize the discussions of process and methodology group.

¹ 〒 560-8531 大阪府豊中市待兼山町 1-3
Graduate School of Engineering Science, Osaka University
Toyonaka, Osaka 560-8531, JAPAN
06-6850-6570 (Ph.) 06-6850-6574 (Fax)
inoue@ics.es.osaka-u.ac.jp

1 はじめに

伝統的なソフトウェア開発方法論やソフトウェア開発プロセスの定義・評価・改善などの技術は、もっぱらメインフレームがソフトウェアシステムの中心で活躍していた時代に考えられ、発展してきた。一方、近年のソフトウェアシステムでは、クライアント-サーバーシステムや3層モデルのWEBシステムなどの開発が大きな割合を占めており、それらの開発では、メインフレーム時代の方法論やプロセスが直接利用できる部分が限られてくる。

このような背景もあってか、今回のワークショップで、プロセスと方法論のグループに参加した7名の参加者のうち4名がオープンソースを意識したポジションペーパーを書いている。それらは、「オープンソースソフトウェアの開発プロセス」、「リポジトリを中心としたオープンソース開発支援環境」、「分散開発環境のためのプロジェクト情報管理システム」などである。これらの発表では、オープンソース開発プロジェクトの解析、プロダクト管理の方法(CVS)、プロジェクト管理のための情報収集機構、作業履歴の保存、などについて発表や議論された。

その他のポジションペーパー発表では、「コンテナ開発技術とソフトウェア工学」、「コンポーネント指向ソフトウェア開発プロセスの観察と分析」などが紹介された。伝統的な開発方法論やプロセス技術を背景にした主張はなかった。

各人のポジションペーパーの概略の発表後に、全体の議論の方向について話し合い、オープンソース開発について突っ込んだ議論をすることにした。そして以下説明を、他のグループに紹介し、意見交換をすることにした。

2 参加者の立場

オープンソース開発は、今まで行なわれていたリジッドな開発方法論やソフトウェア開発プロセス技術とは対極の技術であり、これらの既存の技術のアンチテーゼになっている。すなわち、オープンソース開発では、ソフトウェア開

発に関わる全ての情報が公開されており、参加者が自由に閲覧し、それを元に新たな貢献をしてプロジェクトにかかわることができる。これは、情報共有における一種の原始共産社会に似ており、リジッドな方法論や技術が定める詳細な役割がほとんど定められていない。

同様に、オープンソース開発以外にも、最近盛んになりつつあるコンポーネント開発やExtreme Programmingなども既存の開発技術のアンチテーゼとも受けとることができよう。

このような過去の技術を否定するようなオープンソース開発は、一つの技術として議論する価値があるか、について議論し、参加者全員は、このやり方は非常に重要で、過去の開発技術を見直し、今後の発展に必須なものとして、きちんと評価するに値することを確認した。

オープンソース開発は、情報公開による原始共産主義のソフトウェア開発技術である。これはインターネットの発達によって可能になった。インターネットの発達によって、今までのメインフレーム時代に開発された諸技術は、情報寡占を前提にしており、常に一部の管理者のみがソフトウェア開発のあらゆる情報を掌握し、それに基づいて、作業者に指令を出し、プロジェクトを成功に導くように仕向けていた。しかし、情報公開によって、あらゆるプロジェクト情報が容易に入手し、参加者の意識を高めることによって、今までの諸技術は不要になるものが多数ある。そのような技術を見極めることが重要である。

3 オープンソース開発の発展

過去、ソフトウェアを開発した人が、自分のソフトウェアを他の人へ受け渡す方法としては、紙テープ、磁気テープなどが利用されていた。自分で作ったプログラムを他の人に利用して貰うのが好ましく、積極的に配布していた人々もいたが、物理的にそれらを手に入れるためには、いろいろな手続きや作業が発生し、容易には実現できなかった。

やがてUUCPや各種ネットワークが立ち上が

り出すと、Mail や FTP を利用して配布することができるようになった。これにより、入手するための手間は大幅に削減でき、恩恵にあずかる人の裾野も広がってきた。これらの人々の中には、入手したソフトウェアの評価を行ない、積極的にそのフィードバックをソフトウェアの作成者に送ったり、自分で改良をパッチとして行なって新たに配布する人々が出てきた。文句を言うだけでなく、積極的に良いものに改良し、それを普及させようとする努力が目につくようになった。

さらに、インターネットが普及するに及んで、より多くの人に接するようになり、ソフトウェア開発、改良作業に多くの人々が参加できるようになった。また、その改良、配布作業も迅速に行なえるようになってきた。それに従って、今まで Mail や FTP だけしか利用していなかったが、CVS や WEB のドキュメンテーションツールを利用して、効率的に管理を行なうような努力が行なわれてきている。

近年の Linux や Apache プロジェクトの成功を受けて、オープンソース開発を支持する人々も増えている。しかし、昔からソフトウェアの交換を頻繁に行なっているような人は、冷静で特に新しいソフトウェア開発のパラダイムとして認めているとは思われない。一方、既存の開発方法論やプロセス技術が嫌いな“アンチプロセス”の人々にとっては、このパラダイムは新たな賛美の対象となっている。

4 オープンソース開発の前提

オープンソース開発には、大切な大前提がいくつかある。

1. 一つは、インターネットの存在である。インターネットなしには、オープンソース開発は成り立たない。
2. 次に、いろいろなソフトウェアを自発的に収集し、もし、その中に不具合を発見したら、それを直さずにはいられないような、いい意味でのハッカーの存在である。このよう

な人間は、インターネットの普及とともに全世界中から多数現れたが、しかし、そのうち、パワフルな技術を持ったボランティアの数は頭打ちになるのではないかと危惧され始めている。

3. さらに、多くのユーザーを集めるような魅力的なソフトウェアの存在である。いくら情報公開しても、ソフトウェアの魅力がなければ、注目を集めず、改良するためのボランティアを確保することができない。一つの目的には一つのソフトウェアに人々は集中するので、既存のオープンソースソフトウェアがある場合、新規のプロジェクトを立ち上げるのはなかなか困難である。

これらの前提は、今後も成り立つであろうか？ 1 は、当面発展し続けるのは間違いない。しかし、2 や 3 に関しては、保証がない。従って、オープンソース開発が今後、より隆盛になるとは限らない。

5 オープンソース開発と既存の開発技術

本ワークショップの他のグループ（要求、再利用、アーキテクチャ、計測と解析）を見た場合、オープンソース開発のコンセプトとうまく整合性が取れるであろうか。ほとんどの場合、それらの技術が不要になる可能性を秘めている。

要求技術：要求を獲得するための諸技術は、オープンソースでは必要がない。とりあえず、プロトタイプソフトウェアを作り、それを公開することによって、どんどん仕様が変更され改良されている。リジッドな要求仕様を確定することは、オープンソース開発ではほとんど無意味である。

再利用技術：情報公開を元にして、ソースコードに直接手を入れ、必要な改良を施していくので、あらかじめ再利用を考え、手間をかけて整理しておくことは必要ない。このような考えは、Extreme Programming にも見られるものである。

アーキテクチャ技術：アーキテクチャの設計は、プロジェクトの中心で、一人の優秀な人物が最初に提示するだけで十分で、種々の記法や図を用いて議論する必要はない。もし、改変の必要があれば、後でアーキテクチャの変更も厭わない。

このように既存技術のほとんどは、オープンソース開発では不必要なものと考えられよう。一方、計測と解析技術に関しては、オープンソース開発でも重要な技術と考えられる。これは、情報公開をより押し進め、魅力あるプロジェクトにするためには、多くの情報を収集・整理し、それを分かりやすい方法で公開する必要がある。そのためには、メトリクス技術や解析技術は必須なものと考えられる。

6 まとめ

このようにプロセスと方法論を含め、既存の技術とは相対する技術と考えられるオープンソース開発について議論を集中的に行なった。この議論は、他のものを否定する部分もあり、かなり刺激的である。しかし、若いソフトウェア工学研究者にとっては魅力的なテーマと思われ、非常に活発な議論が行なわれた。ソフトウェア工学研究に多くの人を引き付けるためにも、オープンソース開発についてさらに議論を深める必要があろう。