

オープンソースソフトウェア開発におけるソフトウェアプロセス改善に関する一考察

松下 誠†

多くのオープンソースソフトウェアにおける開発は、各地に分散した開発者が相互に連絡を取る、疎結合ながら協調した作業を行うことによって行われている。このため、従来のソフトウェア開発に見られた、単一の組織やグループという前提が成立せず、開発者が独自で作業を進めるためにそのプロセスは非常に規模が小さく、また未熟なものとなっている。本稿では、オープンソースソフトウェアにおけるソフトウェアプロセス対象として、その評価や改善を行っていくにはどのような方法が考えられるかについて、既存のソフトウェア開発と比較しながら考察を行う。

Some Issues on Software Process Improvements on Open Source Software Development

MAKOTO MATSUSHITA†

Many open-source software developments are based on loose-coupled and distributed environments, and workers are co-operating each other with close communications. In this cases, there are no centralized, uniform organizations or groups; workers are working independently, and its software process is small and immature. In this paper, I present some discussions about software processes on open-source software development environment.

1. オープンソースソフトウェア開発と従来の開発との違い

オープンソースソフトウェアの世界では、事前にソースコードが公開されているソフトウェアに対して、多くの作業者が協調しながら開発を行っていることが一般的である(以下、このような開発を「オープンソースソフトウェア開発」と呼ぶ)¹⁾。しかしながら、その形態は多岐に渡っており、従来の企業等におけるソフトウェア開発とは大きく異なった点が多い。また、多くのオープンソースソフトウェア開発では、その開発作業を遂行する上において、さまざまな問題を抱えている。

ソフトウェア開発をどのような視点で捉えるかによって分類の方法は異なるが、例えば以下のような点が、従来の企業等におけるソフトウェア開発と、オープンソースソフトウェア開発と、ソフトウェア開発時における違いからくる問題として挙げることができる。

1. 開発者自体が不明確

オープンソースソフトウェアの場合、多くの利用者

の中の一部が、そのソフトウェア自身を修正し始めることから、そのまま開発者として活動することがある。つまり、客観的に見た場合に誰が開発者で、誰がそうではないのか、はっきりとした定義を与えるのは非常に困難である。これは、オープンソースソフトウェアにおけるソフトウェア開発について考える場合に、対象となる範囲をどのように定めれば良いかを個別に考える必要があることを意味する。また、誰がどのような作業を行っているかがわからないため、問題が発生した場合の対応を取りにくいといった問題もある。

これに対し、企業等のソフトウェア開発では、その企業で開発に携わっている人々が開発者であるため、厳密に開発者とそれ以外の人々を分けて考えることが容易である。

2. 開発者同士の関係が希薄

仮にオープンソースソフトウェアの開発者が誰であるか、その定義を与えられたとしても、それら開発者同士で何らかの「開発組織」と呼べる組織が形作られていることはごくまれである。また、企業における雇用契約などのように、開発者間の中で何らかの契約関係が存在することも少ない。組織や契約関係が希薄であるということは、開発者同士で共通して何らかの統

† 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University

一した作業を行おうとする場合に、何らかの強制力を働かせることが非常に難しい。

一方、企業等のソフトウェア開発では、これらが非常に明確であるため、ソフトウェア開発全体を考える場合に、組織や関係を土台として全体の開発作業を考えることが容易であり、開発全体で作業をある方向性を持った形で進めることも比較的行きやすい。

3. 開発プロセスが希薄

オープンソースソフトウェア開発では、開発時に組織的な行動を取りにくいいため、開発作業自体が個人に依存した形態となりやすい。そのため、何らかの標準的な開発プロセスを定義した上で、それに従って開発を進めるという方法を取ることが困難である。また、各開発はさまざまな目的をもって行われているため、標準的な開発プロセスを定義すること自体が難しいという問題もある。

2. オープンソースソフトウェア開発におけるプロセス

例えば、典型的なオープンソースソフトウェア開発における保守作業として、次のような物が考えられる。まず、そのソフトウェアを主な開発者の誰かが、バグ報告システムに蓄積されている報告の中から、比較的最近の物を1つ取りだし、問題の解決のためにデバッグ作業を行う。デバッグの結果、問題の修正を行うことができるソースコードの差分が作成される。作成された差分は、バグ報告システムとメーリングリストを用いて公開される。他の開発者や、当該報告を送信した利用者から、確かにこの差分によって問題が修正されることが確認され、実際のソースコードが修正され、新たな版としてソフトウェアリポジトリに登録される。

このような場合、バグ報告システムやメーリングリストといったツールを中心として考えると、小規模なプロセスをそこにみつけることができる。上記の例の場合、バグ修正プロセス(上記全体)と、レビュープロセス(差分を作成してから、確認が行われるまで)という、階層構造を持つ2つのプロセスがあると考えられる。

3. プロセス改善に向けて

オープンソースソフトウェア開発では、確かに個々のプロセスは多種多様であるが、開発者相互で用いられているツールを中心にして考えると、その数はそう多くなく、また機能も単純である。具体的には、生成物を管理するためのソースコードリポジトリ、開発者相互の連絡に用いられる電子メールやIRCなどの文

字ベースコミュニケーションツール、生成物の配布や情報公開を行うWWWサーバ等、である。

前節の例で挙げるように、これらのツールに着目すると、オープンソースソフトウェア開発は小規模のソフトウェアプロセスが複雑に組みあわさった形態であると言える。従来の企業等におけるソフトウェアプロセスを考える場合には、対象とするプロセスや組織の規模に関する多くの問題があったが、オープンソースソフトウェア開発の場合は、各プロセス同士の相互関係や依存関係に関する問題が重要となってくる。

そこで、オープンソースソフトウェア開発における作業の効率を改善するためには、次のような方向性があるかと考える。

まず、プロセスを把握することが必要であるため、ツール中心の軽量ソフトウェアプロセスとしてオープンソースソフトウェア開発を記述する。個別の小さなソフトウェアプロセスは、そのツールに対するユースケースとして考えることができる。

既存のツールの持つ機能と、記述されたソフトウェアプロセスを比較し、プロセス実行における問題点を洗いだして整理する。必要であれば、ツール側の機能を拡張することによって、プロセスの支援が行えないか検討する。また、記述されたソフトウェアプロセスの相互関係を調べることにより、競合しているプロセスや、重複しているプロセスを発見することができるであろう。

これらの方向性を通じて、オープンソースソフトウェア開発における標準プロセスを、標準的なツールに基づいて定義することも可能ではないかと考えられる。

4. おわりに

現状のオープンソースソフトウェア開発は単純なツールの上に構成された、複雑でかつ小規模のプロセスの集合として捉えることが可能と考えられる。このようなソフトウェアプロセスに対する改善を行うためには、まず現状をツール中心のプロセスとして分類、整理を行った上で、ツール中心の分析を行うことが重要ではないだろうか。

参考文献

- 1) 情報サービス産業協会: オープンソース・ビジネスの動向調査(2002).