Red Hat オープンソース アップストリーム ファーストの取組み

レッドハット株式会社　APACテクノロジーオフィス チーフ テクノロジスト　杉山（すぎやま）　秀次（ひでつぐ）

1. はじめに

既にオープンソースソフトウェアは10年前には驚くような速さでエンタープライズソフトウェアの世界を支配しており、クラウドソフトウェアのほとんどの商用展開には、オープンソースソフトウェアの複数の要素が含まれている。それらを管理するために使用するツールはすべてオープンソースだ。今、世の中には1億4500万以上にのぼるオープンソースプロジェクトが存在する。10年前の10倍以上にオープンソースプロジェクト数が増大している状況だ。近年のデジタルトランスフォーメションにおける異業種間のオープンイノベーションの必要性がオープンソースの価値を高めていると言える。オープンソースコミュニティへの関与は、今後のテレコム業界における標準化を進める過程の中で重要な取組みとなっていくだろう。

今回の記事では、オープンソースソフトウェアとは何か、オープンソースコミュニティの立ち上げや関与の仕方、オープンソースソフトウェア開発手法について紹介する。

2. オープンソースの基礎

オープンソースソフトウェアは特定のライセンス類でリリースされたソフトウェアでありオープンソースとして認められるために、ソフトウェアの利用者にいくつかの権利を与えている。ソフトウェアの開発者が利用者に与える権利「フリーソフトウェア」の自由、または「オープンソース」のオープン性である。オープンソースにはソフトウェア開発者としての哲学があり、ライセンスの種類を3つ紹介しておく。

1つ目はGNU General Public Licenseとその「コピーレフト」のライセンス類。ソースコードを共有してこれらの使用権利を利用者に与えた場合、ソフトウェアを再頒布する時に同じことを行い、利用者に同じ権利を与えることが期待される。

2つ目はBerkeley Standard Distributionライセンス、Apacheライセンス、MITライセンスなどのライセンス類。これらの場合、「開発者にとって最も重要なことは、できるだけ多くの人が自分のソフトウェアをダウンロードして共有することであり、それ以降のことはそれほど重要ではない」との世界観を持ち、元の開発者の仕事の功績を認めている限り、利用者がソフトウェアを変更したり、他の人に同じ権利を渡したり、著作物を独自のソフトウェアソリューションに統合したりするかどうかはあまり気にしない。

3つ目はMozilla Public License（バージョン2）、GNU Lesser General Public License、またはEclipse Public Licenseを含むライセンス類。これらの場合、オリジナル作品の変更を他の人と同様の条件で共有する必要があるが、新しい作業の追加には異なる条件でライセンスを与えることができるライセンスである。

ライセンスは、オープンソースソフトウェア開発者のコラボレーションの哲学を表すことができ、開発者がどのようにプロジェクトとやり取りすることを期待しているかを伝えることができる。

3. コミュニティ開発

他者にソフトウェアの修正を頒布する能力は、ソフトウェアの共同開発、及び元の開発者、ソフトウェアの利用者、競合他社との共同作業の機会を創出する。この現象は、Linuxカーネル、OpenStack、Eclipse、Apacheプロジェクト、その他強力なオープンソースプロジェクトが生み出したものだ。しかし、新しいソフトウェアプロジェクトにコミュニティを集めるプロセスはよく理解されていない。また、新しいアイデアを具現化するためにコミュニティを活用して標準化していく方法もよく理解されていない。

「健全なコミュニティを作る」ことを考える際、様々な規模のコミュニティの性質を調べたDunbar's Number* が

＊　Dunbar's Number
https://en.wikipedia.org/wiki/Dunbar%27s_number
Upstream First：Turning OpenStack into an NFV platform
http://community.redhat.com/blog/2015/03/upstream-first-turning-openstack-into-an-nfv-platform/

頭に浮かんでくる。Robin Dunbarは頻繁に繰り返される幾つかのグループを見つけ、特定のサイズの制限でその特性が変化することを具体的なコミュニティサイズ5、15、50、150、500、5,000を使い彼の著書で紹介している。進化について多くのことは書かれていないが、グループが成長するにつれてグループダイナミクスがどのように変化するかが分かる。コミュニティメンバーとの関係は、コミュニティ規模が膨らむにつれて時間とともに変化する。小さな5人の開発者プロジェクトのための実行可能なインフラストラクチャは、OpenStackのような巨大なエコシステムと同じではない。成長の過程で複数のコミュニティと関係を広めていくことになるだろう。例えば、OpenStackとコンテナKubernetesのCNCF（Cloud Native Computing Foundation）、OpenDaylight、OPNFV（Open Platform for NFV）との関係などが参考になるだろう。AT&TとチャイナモバイルがLinux Foundation主導のもとでマージしたONAP（Open Network Automation Platform）も多くのコミュニティを巻き込もうとしている。不安の瞬間に刻まれた成長の瞬間があり、その瞬間、成長を続けるためには変化が必要だ。さもなければコミュニティが停滞して消滅することになるだろう。これらの変化のそれぞれで必要となる構造とプロセスは1つの軸だ。おそらくより重要なもう1つの軸は、コミュニティ内の人々が互いにどのように関係しているかだ。個人は、一度に複数のアイデンティティを持つことができ、それぞれのアイデンティティはより強くまたは弱くもなる。これらの変化は、Dunbarの数字の境界あたりで発生する。それぞれの変化に伴って、以前に行ったことの何かが失われ、懐かしさ、不安、以前のものが改ざんされたという不満が生じ、良いコミュニティは、これらの感情的な結果にも注意を払うだろう。それが拡大するにつれコミュニティの創設価値を永続させることは難しい課題だが、コミュニティが成長するにつれて、教義、伝承、物語を組み合わせて価値を伝えることになるだろう。

Red Hat OSAS（Open Source And Standard）チームではアップストリームトレーニングプログラムを開発し、オープンソースでの取組みの必要性を感じているテレコムキャリアやテレコムベンダーにコミュニティの立ち上げ方や関与の仕方を伝授している。

日本にもLinux/オープンソースのベテランは多くいるが、世界でコミュニティをリードしているとは言えない。コミュニティのリードの仕方や人の動かし方を学生時代から学んだ現代の40代前後の欧米人に比べて、普段の生活の中でも学ぶ機会の少ない日本では、言葉の壁以上に基礎に差がある。この基礎の差を埋めない限り、世界で標準化をリードすることは難しいだろう。アップストリームトレーニングでは、既存のプロジェクトに参加して効果を発揮する方法についても時間を費やして伝授している。

4. コミュニティでの作業

オープンソースコミュニティ貢献力に焦点を当て、それがなぜ重要であるかについて説明する。

製品化の成功は一企業の仕事だけでなく、オープンソースパートナーの品質と成功にも左右される。より良いオープンソースプロジェクトはより良い製品を作り、成功したオープンソースプロジェクトは、付加価値製品に対する多くの潜在顧客を生むことになる。パートナーシップ契約を締結していれば、サポート契約、責任範囲、おそらく共同サポートと共同マーケティング活動等、あらゆる種類の事柄について交渉することになり、パートナーにも契約に対するコミットメントがあることを確実にする方法がない場合は、パートナーシップが成立されない。オープンソースコミュニティでは、このような契約は存在しない。顧客が問題を見つけた時にコミュニティプロジェクトで問題が修正される事を保証したいのであれば、それが起こる事を確実にする方法は1つだけであり、それはソフトウェアエンジニアが修正できるようにすることである。

アップストリームプロジェクトに直接質問し、修正を説明し、顧客を満足させる時間枠で問題が修正されることを期待することができる。良いコミュニティは問題に非常に敏感に反応するが、無償サポートとして扱われるのを楽しむ事はない。

4.1　アップストリームファーストが何故重要か

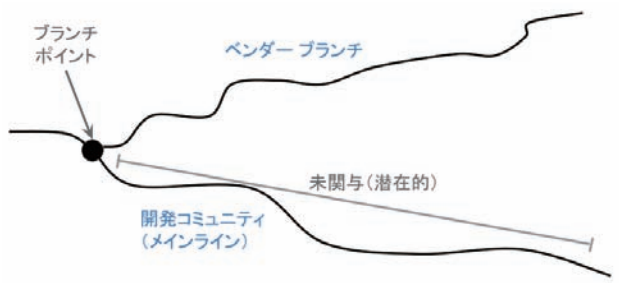
まず、「アップストリーム/ダウンストリーム」について説明する。開発者の脳であるソースコードから、サポートされている実稼働環境へのコードの流れを考えてみよう。コードは書かれ、オープンソースプロジェクトに提出され、そこで統合される前にいくつかの改良が行われる。そこから、コードがリリースされてその製品またはソリューションに含まれるように企業が採用する。恐らく、製品に小さな変更を加えてから、顧客に販売することになる。開発者が関与して共同作業するオープンソースプロジェクトはアップストリームと呼ばれ、そのソースから描画されたソフトウェアの製品とユーザーは、そのダウンストリームに

ある。「アップストリームファースト」開発とは、オープンソースプロジェクトに基づく製品に含める変更（機能、バグ修正）を、製品に組み込む前に最初にアップストリームプロジェクトに提出するという考え方だ。これにより、長期メンテナンスの負担を最小限に抑えることができる。ただし、Red Hatの製品化本流の流れに合流する上流アップストリームプロジェクトは沢山あるが、上流になれずに本流に合流することなく消滅してしまったプロジェクトも沢山ある。アップストリームプロジェクトとそのコミュニティの健全性は製品化に採用される際の重要な指標となっている。

ここで、大規模なオープンソースプロジェクトを製品化するシナリオで、「プライベートフォーク」と「アップストリームファースト」の相対的なコストを比べてみよう。

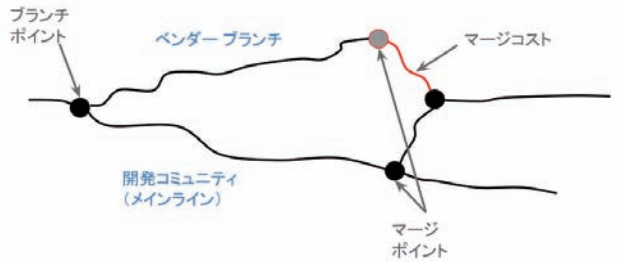
4.2　プライベートフォーク

オープンソースの上に何かを構築する最も簡単な方法は、ある時点でソースコードを入手して使用するプライベートフォークだが、変更を加える必要がある場合、コピーにそれらを作成し、コードの差分を永続的にメンテナンスしていかなければならなくなる。



■図1.　プライベートフォーク

これは、特に本質的に変更されていないプロジェクトを使用する予定がある場合は、多くのメリットがあるが、アップストリームプロジェクト上に多くの変更を加えれば、すぐに謎に陥る。アップストリームプロジェクトで行われているすべての作業を逃すことになる。時間が経つと、ローカルブランチに変更が加えられる。マージには時間がかかり、アップストリームからの分岐はプロジェクトの他のバージョンとの機能上の違いを引き起こすため、新しいバージョンのアップストリームに移行するのは煩雑になる。アップストリームプロジェクトの新しいバージョンに移行して、新機能と修正をすべて統合すること（マージまたはリベースと呼ばれる）は、どちらも簡単なことではない。

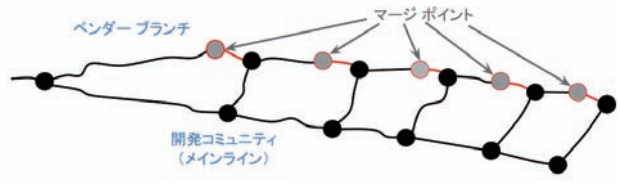


■図2.　リベース

ここでのマージコストには、すべての作業の開発コストと、コミュニティが同じファイルで作業している時の差分を解決するコストと、プロダクション用のリリースのテストと認定、アップグレードコストそしてアップストリームプロジェクトの新しいAPIや機能に合わせる作業コストが含まれる。コードと新しいアップストリーム機能の相互作用によってデグレ等が発生する恐れもある。これまでには見られなかった問題が発見され、ブランチとアップストリームプロジェクト間の差分が増加することもある。最新バージョンのプロジェクトへ継続したリベースのコストは高価であり、評価するのが難しいため、多くの企業は古いバージョンのプロジェクトを維持するのに固執してしまう。この問題は、プライベートフォーク固有のコードが多くなるほど悪化する。もちろん、これは1回限りの操作ではない。1年か2年後には、新しいバージョンに対してすべてのコードを再検討し、新しいコードを再統合し、コードの一部を再開発し、すべてを再テストし、再展開し、このコストは時間が経つにつれて増えてしまう。コードを積極的に開発している場合は、進捗が大幅に遅くなる。開発チームは、新しいコードを開発するよりも、基盤となるプラットフォームをマージしてアップグレードし、その結果生じる問題を修正するために、より多くの時間を費やすことになるだろう。

アイデアを具現化するために複数のオープンソースプロジェクトを活用して開発する場合、この問題はさらに複雑になる。

NFVでよく要求される機能であるOpenStackインスタンスでSR-IOV仮想インタフェースをサポートするなどの機



■図3.　膨らむマージコスト

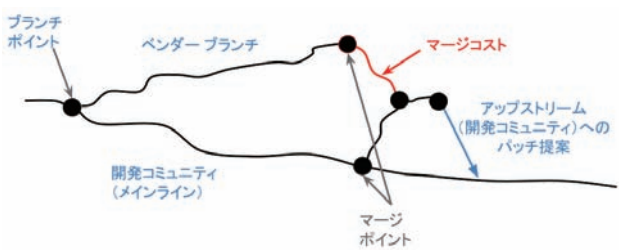
能は、OpenStack Compute（Nova）、libvirt、qemu-kvm、及びLinuxカーネルに複数のパッチが必要だ。これらのパッチのすべてがアップストリームのソースツリーから漏れている場合、長期的なフォークでは、アップストリームプロジェクトへの更新は、これらの相互依存するパッチのすべてを大幅に修正する可能性がある。

OPNFVのNFVプラットフォームもOpenStack、KVM、OpenDaylight、OVS/VPP、DPDK、Ceph等多くのプロジェクトで構成されており、複数のオープンソースプロジェクトにまたがるベンダーブランチで重要なパッチを維持することは、持続不可能である。このプロセスを1回か2回経験した後は、このマージとメンテナンスの負担を軽減するために、変更を親となるアップストリームプロジェクトに提出することは避けられない。しかし、一部のベンダーは、プライベートフォークして機能を開発し独自製品にそれらの機能を統合することで変更をアップストリームプロジェクトに統合することなく差別化を維持しようとしていたり、独自製品リリース後に変更をアップストリームに働きかけるスイミングアップストリームを行っているベンダーもいるのが、OPNFVコミュニティの現状だ。

4.2.1　スイミングアップストリーム

未学習の人にとっては、機能やパッチをオープンソースプロジェクトに提出するのは簡単な事だと思われるかもしれない。これらの貢献は「他者にとって有益であり、提出者のために多くの作業を要するので、アップストリームプロジェクトが新しい貢献を得ることに感謝するであろう」と思われるかもしれない。実際にはそれほど単純ではない。スイミングアップストリームとは、聞き慣れない言葉かと思われるが、子孫を残すためにサケが産卵場所の上流へ泳ぎ登る行為を思い浮かべて欲しい。大変な作業であり、辿り着けるのは僅かだ。

一度プライベートフォークしたコードをメンテナンスの負担を軽減するために、差分をアップストリームで管理可

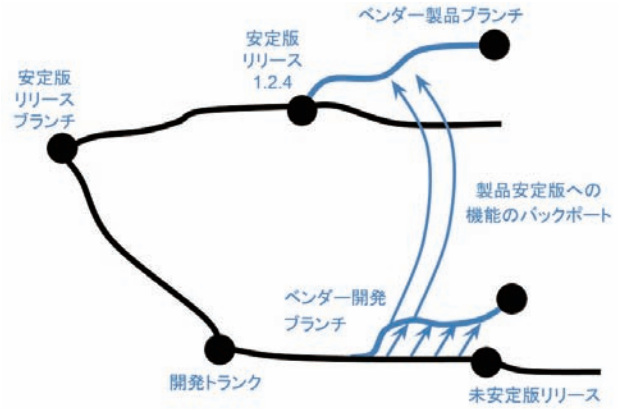


■図4.　スイミングアップストリーム

能なレベルまで減らすことに取り組む場合には、複数の問題が存在する。パッチは現在の開発バージョンのプロジェクトには適用されていないため、パッチのいくつかは、何か月も見直されていなかったり、良くないアイデアであるために拒否されたり、コミュニティのメンテナーが不承認とするケースがある。

4.3　アップストリームファースト

開発者が最初からアップストリームプロジェクトに従事することで、初期のアプローチがコミュニティの期待に合わない場合や、最新の開発ツリーに対してパッチをメンテナンスする場合、アップストリームで修正を迅速に行うことができる。アップストリームのコミュニティとの関係を構築することで、変更を受け入れることが容易となり、製品の基盤となっている安定したブランチに機能をバックポートすることで以前のバージョンの機能やパッチを出荷することが可能となる。



■図5.　アップストリームファースト

したがって、ソフトウェア開発作業する理想的な方法は、アップストリームプロジェクトの不安定な開発ブランチに対して機能を開発することだ。このようにして、パッチは準備ができたらすぐに提案することができる。製品の安定したリリースに基づいた製品提供－新しい機能を製品に統合する場合、これらの機能は開発ブランチから取得し、安定したブランチにバックポートする必要があるが、メンテナンス費用は、徐々に低下していく。このアプローチは無料ではない。製品化の前にアップストリームプロジェクトで開発作業を進める必要があり、新しいアイデアを具現化するために必要な実装機能を議論して公衆に明示していく必要がある。アップストリームメンテナーとの関係も構築する必要があり、開発作業を進めていく過程でコミュニ

ティ全体のニーズが考慮される。当初すべてを、このような方法で機能開発するには、ベンダーブランチでの安定したリリースからプライベート開発する方法に比べて約2倍の時間と労力がかかるだろう。しかし､そのアップストリーム作業のリターンは膨大だ。コードがアップストリームのソースツリーに受け入れられると、それは将来の安定版リリースに含まれる｡メンテナンス作業はコミュニティによって共有される。また、コミュニティの一員としてアップストリームで作業することで、将来の作業もまた受け入れられるようになり、プロジェクトの将来の優先事項を定義する際に真の声が聞こえることになるだろう。これはアップストリーム開発者が作業する方法であり､「変更は小さく早期にリリース、頻繁にリリースして利用者から早めにフィードバックを得る」ことは、成功した関係モデルであることが証明されており、コミュニティプロジェクトを成功させる良い方法であることから、オープンソースの真言となっている。

Red Hatではアップストリームファーストの透明性／成熟化をさらに高めるため、テレコムキャリアやパートナーとの間でDCI（Distributed Continuous Integration）モデルを確立し、RDOコミュニティ版リリース後、Red Hat OpenStack製品前のコードをより速い段階で評価、またパートナーNFV関連機能をプラグインインテグレーションしてもらえる関係を構築している。

5. おわりに

オープンソースコミュニティイノベーションがRed Hatのビジネスの基本だ。オープンソースを活用して製品を作る良い方法を熟知している。アップストリーム作業へのコミットメントは、Red Hatビジネスのファンダメンタルなビルディングブロックであり、アップストリームプロジェクトと連携して様々なプレイヤーと協力しあいオープンイノベーションを起こし、複数のプロジェクトを安定化させて製品化で統合させている。Red Hatが主導して参加するコミュニティプロジェクトでは、顧客やパートナーの参加を促し、今後の製品リリースの方向性と優先順位を定義し、顧客やパートナーと一緒に歩む事ができる手段をとっている。

Red Hatは、オープンソースプロジェクトを提供するだけでなく、OPNFVコミュニティのようなアップストリームプロジェクトではないコミュニティでも活動しており、将来のオープンソースのNFVプラットフォームを定義する事に役立てている。そこで活動を共にしているテレコムキャリアやテレコムベンダーはOpenStackベースのNFVプラットフォームを配備する際に必要なパートナーとなってきている。

最後に、オープンソースを活用した標準化を進めるにはOpenStackやCNCF等様々なオープンソースコミュニティの場で議論するところから始める事を推奨したい。