Solution Brief

GitHub Enterprise

インナーソース：オープンソースの開発手法を商用ソフトウェア開発で実施

世界中で、オープンソースの開発手法を商用ソフトウェア開発に組み込んでおり、これは「インナーソース」と呼ばれています。

この現代的なアプローチを商用ソフトウェア開発に採用することで、革新的なコラボレーションが進み、高品質なソフトウェア開発を実現できます。

インナーソースとは、Kubernetesのような大規模オープンソースプロジェクトでのベストプラクティスを駆使して、商用ソフトウェアを開発する手法です。

大規模なオープンソースプロジェクトには、数千人のコントリビュータのコラボレーションとチームワークが必要です。高品質のオープンソースプロジェクトは、信頼性、機能性などのユーザーニーズを満たしたソフトをスピーディに開発しています。商用ソフトウェア開発においても、オープンソースの開発手法から学べることは多くあります。

本コンテンツでは、オープンソースのベストプラクティスを、効果的に商用ソフトウェア開発に取り入れる方法を紹介します。

オープンソースプロジェクトとは

GitHubでは、世界最大のオープンソースコミュニティが築かれています。このコミュニティ内には数百万人のコントリビュータが存在し、それぞれの関心事やスキルも広範にわたります。これらコントリビュータは、GitHubのオープンソースコミュニティでAppleのSwiftのようなプログラミング言語や、FacebookのReact.jsのようなフレームワークに貢献しています。また、作業範囲はコーディングだけにとどまらず、データセット、法的文書など、多くのものをオープンソースにすることができます。

オープンソースソフトウェアは誰でも使用できます。また、オープンソースライセンスで定められている通りに、目的に従ってプロジェクトの閲覧、修正、配布することも可能です。同様に、誰でもオープンソースソフトウェアの開発に参加することができます。オープンソースの開発は、世界中にちらばるコントリビュータのコミュニティによるコーディング、フィードバック、バグレポートなどを通して行われます。

このようなオープンコラボレーションのベストプラクティスはインナーソースの基本となります。以下のセクションでは、オープンソース開発におけるツールや用語について説明します。

オープンソースプロジェクトを開始し貢献する方法に関する詳しい情報は、Open Source Guide（日本語：https://opensource.guide/ja/）をご覧ください。

オープンソースコミュニティから学べること

今日のソフトウェア開発は、すべてのコードを自分で書くわけではありません。多くの場合、他のソフトウェアなどで使われているコードや、オープンソースのプロジェクトをベースに変更したり、追加したりして開発を進めます。オープンソースの開発手法には、大規模で複雑なプロジェクトの管理方法、多くの開発者が参加する場合の管理方法、時差がある国をまたいだ開発におけるコラボレーション方法など、学べることがたくさんあります。このような学びをもとに、企業内においてインナーソースを実行する上で注意すべき点を挙げました。

オープンコラボレーションはより多くのコントリビューションを得られる

コントリビューションが増えることで、ソフトウェアの品質は上がります。オープンソースプロジェクトは、世界中の誰からでもコントリビューションを受け入れながらプロジェクトを進め、ユーザーのニーズに応え、バグを見つけて修正します。つまり、オープンソースコミュニティにはソフトウェア開発の頭脳が集まっているのです。

企業内でのソフトウェア開発でも同じ手法を使うことができます。企業内のあらゆるソースコードやドキュメントを誰もが閲覧できるようにすることで、エラーや矛盾がないかコードを検証する人が増え、より安全で、より信頼性の高いソフトウェアを開発することができます。

オープンソースでは世界中の誰でもすべてのプロジェクトを閲覧できる（パブリックリポジトリ）であるのに対し、インナーソースでは企業内の特定の人だけがプロジェクトを閲覧できる（プライベートリポジトリ）ように制限をかけるのです。

エンジニアは、ゼロから開始する必要はない

誰でもオープンソースプロジェクトを再利用することができます。これを使って別のものを作成したり、特定のニーズに合わせて修正することも可能です。

同様に、インナーソースを実施することで、その企業内に存在するコードはどの開発チームでも検索し、カスタマイズして再利用することができるようになります。また、企業内のプロセスをドキュメント化することによって、ソフトウェアのデプロイや使用方法を確立することもできます。これにより、低コストかつより大きな柔軟性が可能になり、ベンダーロックインを解消できます。

透明な意思決定が、プロセス、信頼、整合性を生み出す

優秀なオープンソースプロジェクトのメンテナは、なぜそのような意思決定となったかを文書化し、周囲が理解できるようにしています。各決定事項にはコンテキストを説明するコメントの履歴があるため、誰もがなぜそのようなコードになったかを理解することができます。

企業におけるソフトウェア開発も同様で、なぜそのコードが承認されたのか、またはリジェクトされのかなど、意思決定にいたったプロセスを文書化することで、透明性がもたらされます。対話をきちんと文書化すると、リモートワークで作業しているエンジニアも効率的にコラボレーションし、開発を進めることができるようになります。ソフトウェアのコードを書くエンジニアだけでなく、プロジェクトマネージャー、デザイナー、セキュリティチームなども含めた、オープンな環境を構築することでコラボレーションが進み、スピーディなソフトウェア開発が行えます。

参加することが重要

オープンソースプロジェクトの成功は、多くのメンバーが参加することにかかっています。各自が参加する理由は、例えば、スキルを向上させる、メンターを見つける、自身のエンジニアとしての評判を上げるなどがありますが、プロジェクトのメンテナは、参加を促し歓迎するコミュニティ文化を創り出す必要もあります。

企業内でも同様です。自社に存在するさまざまなコードから多くのことを学び、社内でアイデアを共有し、同僚から学ぶことができます。多くの企業では、それぞれの開発チームがサイロ化し、他のチームがどのような開発を行っているかさえもわからなケースが多く見受けられます。このような状況を変え、社内コラボレーションを進めるには、社内文化の変化も必要になります。知識の共有を促し、別のチームからのフィードバックやコントリビューションを歓迎する文化に変える必要があります。

プロジェクトの中核メンバーはプロセスを改善する

オープンソースプロジェクトには数千人のコントリビュータやコミュニティのメンバーが参加していますが、プロジェクトの全体的な方向性を決めるのはかなり小規模なチームで行われます。これにより、意思決定が迅速に行われ、常に責任の所在が明確化されます。

企業におけるインナーソースプロジェクトも同様で、プロジェクトの管理を小規模なグループに任せることで、承認や確認がより効果的になります。意思決定者のチームを小規模で組織横断的なものにすることで、チームの品質基準を満たし、経営陣からのサポートを得るのにも役に立ちます。

社内環境でのオープンソースコミュニティ

インナーソースという言葉やそのメリットはまだよく理解されておらず、オープンソースプロジェクトにすでに取り組んでいるエンジニアでも、インナーソースという用語を全く知らない可能性もあります。一方で、インナーソースという言葉を知らずとも、その手法を企業内のクローズド環境で実施し、ソフトウェア開発を進めているエンジニアも存在します。

企業でインナーソースを展開しても、セキュリティやプライバシー保護がおろそかになるわけではありません。オープンソースのように公共でプロプライエタリ・ソフトウェアを共有したり、社外の人を招待してソースコードを閲覧させたり、コンフィデンシャルなプロジェクトにアクセスを与えたりする必要はありません。公開されていないコードは社内環境内にとどまり、適切な権限を持つエンジニアだけがコントリビュートするように設定することで、セキュアなソフトウェア開発を行えます。

> 「現在目にしていることは、テクノロジーが革新やコラボレーションという考えに追いついたという状況で、これが非常に重要なことなのです。」
>
> Joan Watson (DXCテクノロジー、研究開発IT、新興技術、ビジネスエンゲージメント)

インナーソースを実践することは、企業内でオープンソースコミュニティを始めることに似ています。インナーソースを実践することでコラボレーションが進み、社内コミュニティの知見はシェアされ、より良いソフトウェアを創り出すことができるようになります。

企業がインナーソースを採用する理由

ビジネスのソフトウェア化が進み、企業は従来の開発手法では競争に勝つことができないと気づき始めています。要件をまとめ、ミーティングを開き、サイロ化された組織で開発するという、ウォーターフォールのように時間がかかる開発手法では、今日のテクノロジーの速さに付いていけませんし、ユーザーのニーズにも迅速に応えられません。では、スピーディにに事業を進めるにはどうしたらいいのでしょうか？

インナーソースは、チームがソフトウェアを迅速に開発し、さらに協力しあうのに役に立ちます。その結果として、高品質の開発が可能になります。以下に、インナーソースのメリットをまとめました。

- 企業内のコードを容易に検索・再利用できるようになり、リソースの無駄遣いや重複を防ぎます。つまり、「車輪の再発明」を防ぐのです。

- 企業規模にかかわらず、迅速な開発を促します

- 組織のサイロ化を減らし、部署内だけでなく他部署間、さら事業ライン全体における企業全体のコラボレーション推進します

- エンジニアと管理職、その他のプロジェクトメンバー間における意思決定プロセスを可視化します

- オープンな文化を作り、オープンソースの参加の下地となります

- エンジニアのモチベーションをあげ、自分の仕事内容や会社への満足度を高めます

PayPal、Bloomberg、Walmartなどの大手企業は、チームや顧客に対するソフトウェア開発にインナーソースを使用しています。インナーソースのベストプラクティスを実行することでユニークな優位性が生まれ、競争力の高いソフトを開発できるようになります。

「インナーソースは、コード再利用を通して効率を改善する方法とみなされていますが、それ以上に、学び、アイデアを交換し、IBM内での革新を促すための素晴らしい方法でもあります。」

Jeff Jagoda (IBM、シニアソフトウェアエンジニア)

インナーソースを進めるための、チーム編成

プロジェクトを成功させるには、適切なメンバーをチーム内に配置することが重要です。社内でインナーソースプロジェクトを進めるにあたり、複数の部署から成るチームを結成する際にもこのことが当てはまります。典型的なオープンソースプロジェクトは、以下のタイプの人々で構成されています。

- メンテナ：開発ビジョンを策定し、プロジェクトの組織管理に責任を持つ担当者。メンテナは、必ずしももともとのプロジェクトやコードのオーナーである必要はありません。

- コントリビュータ：プロジェクトに何らかのコントリビューションをするすべての人。

- コミュニティメンバー：プロジェクトを使用する人たち。これらの人たちは頻繁に意見を述べ合います。プロジェクトの方向性に関しても意見を述べる可能性があります。

プロジェクトが大きくなると、ツール、タスクの優先順位付け、コミュニティに合わせた調整などの様々なタスクを中心に行う小委員会または作業グループを設定することがあります。

インナーソースプロジェクトも、同様の構造を参考にチーム編成を行うべきでしょう。企業の多くは、エンジニアをアプリケーションエンジニアリング、プラットフォームエンジニアリング、ウェブデベロップメントなどのチームに分けています。このように組織を固定化すると有能な人材が貢献できる領域を狭めてしまうことになります。意思決定を行うチームを組織全体がサポートする形にすることで、問題を迅速に解決するのに必要な専門知識を結集することができます。

企業内では、「コントリビュータ」はその企業に属するすべてのエンジニアであり、「メンテナ」はプロジェクトのリーダーであり、主な意思決定者です。

- メンテナ：エンジニア部門のリーダー、プロダクトマネージャー、その他の意思決定者で、プロジェクトのビジョン策定、および日々のプロジェクト管理に責任を持ちます。

- コントリビュータ：エンジニア、データサイエンティスト、プロダクトマネージャー、マーケティング担当者、その他、プロジェクトを前進させるすべてのメンバーです。コントリビュータは、直接的なプロジェクトチームの一員である必要はありません。コードのレビューをしたり、バグフィックスを提出するなど、ソフトウェア開発をさまざまな形で支援します。

GitHub機能の紹介

オープンソースやインナーソース開発を行うにあたり、GitHubが提供する機能をいくつかご紹介します。

- Issue： Issueはエンジニアがトピックを上げて対話を始める場所です。バグが見つかったり、新機能についてのアイデアがあった場合、まずIssueに書き込んでください。他のメンバーはそのIssueを読み、それに対する意見交換を始めます。

- Pull request：Pull Requestは、エンジニアが加えようとしている変更について議論する場所です。ここで、ソリューションについて作業を始めたり、進行中の変更を確認することができます。プルリクエストについて詳しい情報をご覧ください。

- チャットツール：場合によっては、迅速な意思決定が必要な時があります。SlackのようなチャットツールはGitHub上の話し合いやコメントを補い、リアルタイムで問題について説明をするのに最適です。

GitHubと連携可能なツールは何百種類もそろっており、プロジェクト管理からCI/CD (継続的インテグレーション/継続的デリバリー) まで、開発作業をより良くするために役に立ちます。

「Bloombergにとってインナーソースは新しいものではありません。競争での優位性は、革新ができること、新しいアイデアを得ることができること、それを市場で競争力を持つために必要な速度で、定期的にユーザーに提供することができることにあります。」

Panna Pavangadkar (Bloomberg、エンジニアリングエンジニアエクスペリエンス· グローバル部長)

インナーソースの準備はできていますか？

インナーソースは、技術的変化と同様に文化的変化でもあります。そのため、企業が遭遇する課題を過小評価しないことが重要です。オープンソースと同様に、インナーソースプロジェクトは社内のコードを一元的に検索し、再利用できることでソフトウェア開発をスピードアップするだけでなく、高品質化できるメリットがあります。一方で、オープンソースを推進するためには企業内で同じ目的意識を持ち、部門を超えたコラボレーションを進めたいと考える、文化的変化が必要です。

インナーソースを始めるにあたり、以下の点を準備しておきましょう。

- チーム間で共有されている、プロジェクトに対するビジョンがある。プロジェクトは、ビジネスゴールや対応すべき問題を明確に定義している。

- オープンソースやインナーソース開発を経験したことのあるメンバーがいると望ましい。

- オープンソースやインナーソース開発を初めて行うユーザーが参加しやすいような雰囲気を作ったり、必要に応じてトレーニングを実施する。

- オープンにコミュニケーションを取り、コラボレーションが行える開発ツールを用意する。

- 開発部門内だけでなく、組織外の他部門の人たちと共通のビジネスゴールを設定する。

インナーソースを実施することで、それぞれが自分自身や自分の責任を見直す契機となるため、開始するまえに関係部署の責任範囲や協業の仕方などの整備をしておくことが重要です。インナーソースを効果的に実施するには、気軽に参加でき、学び合い、お互いをサポートし合う文化が必要です。

また、他部署からのフィードバックを受け入れることが当たり前というような環境であることが重要です。ソフトウェア開発に限らず、自社におけるナレッジシェアリングがどの程度効果的にできているかは、インナーソースをうまく実施できるかを測るひとつの指標となるかもしれません。

以下は、自社ががどの程度準備ができているかの評価に役立つ質問です。「はい」と回答できる質問が多ければ、インナーソースを実施する準備ができているサインです。

- 自社の文化はオープンであると思うか？

- 単一のプラットフォームでソフトウェアを開発しているか？

- ソフトウェア開発に対し十分にリソースが与えられ、経営陣など上層部からの支援があるか？

- 所属しているチームとは関係ない開発プロジェクトに参加しても、問題ない文化が醸成されているか？

- オープンソースコミュニティに参加している社員がいるか？

- エンジニアリングチームは、CI/CDを導入しているか？

- 社内には、以前から部門を超えた複数のチーム間でプロジェクトを進める文化はあるか？

- ある場合、これらの組織横断的なプロジェクトのリーダーは誰だか明確になっているか？

すべての企業がすぐにインナーソースを実施できる状態にあるわけではありません。インナーソースは、異なる意見を受け入れ、異なる部署間でもフィードバックをするのに十分な信頼がなければ成功しません。多くの企業は、小さいプロジェクトを通してパイロットプログラムを始め、その後社内全体に広げていくという方法を選んでいます。

「インナーソースがうまくいき始めると、開発におけるコントリビューションだけでなく、開発のボトルネックを取り除けることもわかります」

Jeremy King (ウォルマート、グローバル· Eコマース担当エグゼクティブヴァイスプレジデント＆最高技術責任者)

インナーソースを始めましょう

多くの場合、企業は小さいプロジェクトからインナーソースを少しずつ開始していきます。パイロットプロジェクトは、チームがよりオープンなプロセスを経験し、自由にコードを閲覧し、ベストプラクティスを文書化するのに役立ちます。これらのナレッジは、より広範にインナーソースを適用していくのに役に立ちます。小さな成功を積み重ねることで、エンジニアの社内コミュニティに、コードを最大限に活用し、より良いソフトウェアを迅速にリリースできることを示せます。

ソフトウェア開発方法は、10年前とは異なります。インナーソースは、開発を加速させ、組織的な障壁を克服し、ソフトウェアの質を向上させるための1つのアプローチです。世界最大のオープンソースコミュニティであるGitHubこそ、オープンソースのベストプラクティスを開始する所です

github.com/enterprise
sales@github.com
1.415.735.4488