

GitHub 実践入門

良いコードを迅速に生み出す快適な共同環境

(写経版)

姫 伯邑考

2020 年 04 月 01 日

第 1 章

GitHub の世界へようこそ

GitHub はどのようなものなのか、なぜ世界中の開発者が利用しているのか解説する。GitHub がオープンソースソフトウェアの世界において起こした革命についても考察する。

1.1 GitHub とは？

GitHub は友人、同僚、クラスメイト、見知らぬ人とコードを共有するために最高の場所を提供してくれる Git リポジトリのホスティングサービスである。

1.1.1 GitHub 社と octocat

GitHub 社はアメリカのサンフランシスコを拠点とする会社である。octocat と呼ばれる、タコのような猫のようなマスコットキャラクターもいる（図 1.1）。



図 1.1 octocat

1.1.2 ただの Git リポジトリのホスティングサービスではない

GitHub は Git リポジトリのホスティング機能だけではなく、開発者やチームが高速で良い品質のコードを生み出すためのコラボレーションを実現する機能を提供している。これらの機能は次章から詳しく解説していく。

創業者の一人である Chris Wanstrath 氏が、友達とコードをシェアし易いような Git リポジトリを作り始めたのがきっかけである。しかし、それは GitHub というプロダクトの 1 の目標であり、通過点であったということが彼のプレゼンテーションで述べられている*1。

1.1.3 GitHub の利用状況

GitHub は 2013 年 12 月時点で 1,000 万以上のリポジトリをホスティングしている*2。世界中の開発者が日夜利用している。

*1 <http://www.slideshare.net/rubymeetup/inside-github-with-chris-wanstrath>

*2 <https://github.com/features#hosting>

1.2 GitHub を使うと何が変わるのか？

世界中のソフトウェア開発の現場は、GitHub の登場によって大きな変貌を遂げた。革命を起こしたと言っても過言ではなく、それは日本も例外ではない。本章では、まだ GitHub を本格的に利用していない者のために、日常のソフトウェア開発に GitHub を導入するとどのような変化が生じるのか一部を簡単に紹介する。

1.2.1 コラボレーションの形が変化する

これまで複数人が協力して仕事を行うためのソフトウェアが数多く生まれては、姿を消してきた。そういったソフトウェアはグループウェアや CRM (*Customer Relationship Management*、顧客関係管理) などが挙げられ、世界中のビジネスマンに利用されている。

しかし、プログラマを中心としたソフトウェア開発者たちがソースコードを書く際にコラボレーションするための決定打となるようなソフトウェアはなかなか登場しなかった。そのため、ソフトウェア開発者たちはバージョン管理システム、バグトラッキングシステム、コードレビューツール、メーリングリスト、IRC (*Internet Relay Chat*)^{*3}。などの様々なツールを組み合わせでコラボレーションを実現してきた。

これまで当たり前とされていたソフトウェア開発のコラボレーションの形を GitHub は大きく変化させた。以下で、その幾つかの機能を紹介する。

開発者たちが化学反応を起こす Pull Request

世界中のソフトウェア開発者が参加している GitHub では、想像もしなかったようなことが、もの凄いスピードで起きている。それは開発者同士の化学反応と言えるだろう。顔を合わせたことのない開発者たちが、地球の反対側にいても一緒にソフトウェアを作り上げているのである。そのようなことが可能となった大きな理由の 1 つは、Pull Request と呼ばれる機能の存在である。

Pull Request とは GitHub にある Git リポジトリに対して、あなたが変更したソースコードを取り込んでもらえるようにリクエストするための機能である。Pull Request を元にコメントのやり取りもできる。バグを直したので取り込んでもらえないか？ というものから、こんな新しい機能を作ってみただけ取り込んでもらえないか？ というものまである。Pull Request によって、気軽にソースコードを変更し、変更した状態を公開して、取り込んでもらえるようリクエストできる。一方、ソフトウェアプロジェクトのポリシーに沿わないような変更であれば、取り込まない自由も存在する。

GitHub の Pull Request はソースコードの差分を軽快に確認できるのに加え、コード行を指定してコメントすることができる。これにより、具体的なコードを指しながら議論できるため、かつてないほど快適なコードレビューが可能となった。

特定のユーザへのコメント

快適で便利なのは Pull Request だけではない。タスク管理やバグ報告は Issue と呼ばれる機能を用いて行う。特定のユーザに見てほしいければ「@ユーザ名」と記述することにより Notifications (後述) が相手に送られるため、Issue を見てもらえるだろう。Wiki も提供しているため、気軽にドキュメントを作成して公開、共有することができる。Wiki は Git の変更履歴も管理しているため、ユーザに気軽に変更してもらえる。

GitHub Flavored Markdown

GitHub では、ユーザが文字入力する全ての機能で GitHub Flavored Markdown (GFM) という記法を利用することができる。この記法で記述することにより簡単にマークアップできるため、読み易いコメントやドキュメントを作成することが可能となる。1 つの記法を覚えるだけで様々なコミュニケーションに使えるのはとても効率的である。その中でも特徴的な機能として、コメントに絵文字を使えるのがコミュニケーションを円滑にするのに役立っている。

GitHub の普及に伴って、Markdown 記法を利用できるサービスが世の中に増えてきている。

^{*3} インターネットを通じて複数の利用者がリアルタイムに文字メッセージを交換することができるチャットシステム。

1.2.2 他のチームのソフトウェアが今まで以上に見える

GitHub が快適なコラボレーションを提供する相手は、自分たちのチーム内だけにとどまらない。興味のあるリポジトリを Watch に登録しておけば、そのリポジトリに関する情報が News Feed に流れてくる。例えば、会社で利用しているライブラリのリポジトリを Watch に登録しておけば、最新バージョンの新機能やバグや修正情報をリアルタイムに把握することができる。

隣のチームが開発しているリポジトリを Watch に登録しておけば、どのような機能を開発しているのか日々見えるようになる。有用な機能やライブラリを開発しているようであれば、自分のチームでも利用できることにすぐ気付くことができるかもしれない。共通のライブラリを切り出して、新たなリポジトリを作成するように発展させることができれば、まさに開発者同士のコラボレーションと言えるだろう。

1.2.3 オープンソースソフトウェアの世界と同じ開発スタイル

GitHub を会社で使用するようになると、オープンソースソフトウェアの世界と同じ開発スタイルで開発できるようになる。既にオープンソースの世界でソフトウェア開発をしている開発者は、会社ごとに独自に採用しているツールなどを覚える必要なく、直ちに開発をスタートすることが可能となる。

逆に言えば、会社で GitHub を利用していれば、新卒で入社してきたプログラマ 1 年生でも、すぐにオープンソースソフトウェアの世界にデビューできるようになるはずである。

オープンソースの世界でのソフトウェア開発と会社のソフトウェア開発の隔たりがなくなるのである。その違いは Git リポジトリが公開されているかどうかだけの違いになる企業すら存在するだろう。

1.3 ソーシャルコーディングとは？

GitHub はオープンソースの世界にソーシャルコーディングという概念を構築したサービスである。この概念は世界中の多くのプログラマに影響を与えた。ソフトウェアの開発方法を革命的に変化させたと言っても過言ではない。本節では、このソーシャルコーディングという概念について詳しく解説する。

SOCIAL CODING（ソーシャルコーディング）という言葉聞いたことがあるだろうか？ 聞いたことがない者は図 1.2 を見たことがあるだろうか？



図 1.2 GitHub のかつてのロゴ

これは GitHub^{*4} のかつてのロゴである。ここに「SOCIAL CODING」というサブタイトルが入っている。2013 年 4 月からは図 1.3 に変更されている。



図 1.3 GitHub の新しいロゴ

GitHub はソーシャルコーディングという概念を作り上げたサービスである。GitHub の登場によって、ソフトウェア開発に携わる人々にソースコードを所有する権利が本当の意味で与えられた。世界中の誰もがソースコードを所有し、自在に変更し、公開することを今まで以上に容易にしたのである。今日では、世界中の多くのプログラマが GitHub を利用してソースコードを公開している。そして、日々のソフトウェア開発は GitHub によって支えられている。

^{*4} <http://github.com>

GitHub 登場以前の過去のソフトウェア開発では、一部の「コミッタ」と呼ばれるソフトウェアのソースコードを改変する権利がある限定された特権階級の人々が主権を握っていた。ソフトウェアのソースコードを書き換えて公開することよりも、この特権階級の人々を説得することの方が手間も時間もかかり面倒ということが数多くあった。こういった理由から、最初はスピード感のあった流行りのソフトウェアが保守的になっていき、廃れていくことも時代の流れとしてあったのは事実である。

しかし、GitHub の登場によってソフトウェア開発者の世界は本当の意味で「民主化」され、全ての人々に平等にソースコードを改変する権利が与えられたと言える。これはソフトウェア開発の世界において大きな革命である。そんな革命を起こした GitHub がスローガンとして掲げていたのが「ソーシャルコーディング」なのである。

ここからは、革命を起こしたソーシャルコーディングという概念について理解を深めるとともに、その原動力となった GitHub というサービスについての概要を解説していく。GitHub の個々の機能の詳細については後述する。

1.4 ソーシャルコーディングをするべき理由

終身雇用が崩壊し、人材流動性が高まっているのが現在の IT 業界である。毎月と言ってよいほど、名立たるプログラマが月末には「退職しました」、月初めには「入社しました」とブログに書き込んでいるのが見受けられる。

そこで、あなたがプログラマの採用担当者だとしたら、

- 今まで書いてきたコードを閲覧できる or 閲覧できないプログラマ
- 最新のソフトウェアに精通している or 精通していないプログラマ
- 言語やソフトウェアが異なることによる多種多様な文化を理解している or 理解していないプログラマ

両者のうち、どちらを採用したいだろうか？

後者のようなプログラマにならないためにも、今後はソーシャルコーディングや GitHub が重要となる。

1.4.1 世界を閉ざさず、多種多様な文化を見よ

公の場に出ないコードに仕事で触れる職業プログラマにこそ、世界の多くの知見を得るためにも、多様な文化に触れるべきである。会社という閉ざされた世界の中だけでプログラミングをしていたのでは、いつの間にか扱っている技術が陳腐化していた、なんてことにもなりかねない。

世界中で使われていて、日々変更されているソースコードや技術、設計、文化に目を向けることは、自身が書くソースコードや成果に多大なる影響を与える。有名なフレームワークの実装などを参考にして、社内で開発しているソフトウェアに有益な実装を施すなどの効果が期待できる。

1.4.2 コードを書けるプログラマが求められている

特に変化の激しい Web 業界でのプログラマは、実際にソースコードを書けることが何よりも求められる。以前は、プログラミングはある程度の経験があればよく、それよりも人間性や協調性、マネジメント能力の方が重視されていた。しかし、現在では本当にコードをガリガリ書ける職業プログラマが求められている。それは、近年の様々な技術の発展によって、1つのサービスを作り上げるのにも複数のプログラミング言語や技術を利用して、多種多様なデバイスに提供しなければならなくなったという背景がある。

このような時代に、求められるソースコードを書ける能力があるかを判断するためには、実際に書いたコードを見るのが一番確実である。現在では、GitHub の登場によりソースコードを公開する権利が平等に与えられる時代になっている。Facebook や Twitter を見ればどのような人なのかが分かるように、GitHub を見ればどのようなプログラマなのかが分かるのである。

これからソーシャルコーディングをしているプログラマはどんどん増えて、当たり前になってくる。実際に、あなたが

書いたコードで採用を判断される時代が目の前まで来ているのである。だからこそ、世間から見えるアウトプットが益々重要となってくる。コードを書くことを生業としている職業プログラマこそ、ソーシャルコーディングを行うべきなのである。

1.4.3 GitHub の最大の特徴は「人に目を向ける」こと

ここで、GitHub が単なるリポジトリホスティングサービスとは何が異なるのか、重要な点を解説する。

GitHub が今までのリポジトリホスティングサービスと大きく異なる点は、中心に人が位置することだと感じている。今までのリポジトリホスティングサービスはプロダクトが中心にあり、そのプロダクトの世界だけに情報が閉じていた。そのリポジトリの管理者が誰なのかを知ることができるが、その人が他に何をしているのかなどの情報は提供されていなかった。

GitHub はプロダクトに加えて人にも注目することができる。その人が公開しているソースコードは全て閲覧することができるし、ダッシュボードに表示されている News Feed を見れば、どのリポジトリに興味を持っているのか、いつコミットしているのかなど、その人がどのような開発を GitHub 上で行っているのかが一目で分かるのである。

あなたが興味のある人にフォーカスすることができるのである。それは、憧れのスーパーハッカーかもしれないし、学校の同級生や会社の同僚かもしれない。

人とコードに目を向けることができるのが、GitHub が提供した新しい世界なのである。

1.5 GitHub が提供する主な機能

GitHub には開発者が良いコードを効率的にアウトプットするための機能が豊富に備わっている。本節では、これらの機能の概要を説明していく。

1.5.1 Git リポジトリ

GitHub で提供する Git リポジトリは、基本的に無料で何個でも作成することができる。しかし、限られた人や自分だけに公開を制限したいようなプライベートリポジトリを作成したい場合には、毎月 7 ドルからのプラン^{*5}に応じた金額を支払うことによって利用することができる。

1.5.2 Organization

通常、個人であれば個人アカウントを利用すればよいのだが、会社などで利用する場合には Organization アカウントを利用することを推奨する。アカウントや権限の管理を一括して行えたり、支払いを統一できるなどのメリットがある。

公開リポジトリしか利用しないのであれば無料で Organization アカウントを作成することができるので、勉強会や IT 系のコミュニティでソフトウェアを開発したりする際に活用できる可能性がある。組織や会社で GitHub を利用するための詳細は後述する。

1.5.3 Issue

Issue 機能は 1 つのタスクや問題を 1 つの Issue に割り当てて、トラッキングや管理を行えるようにするための機能である。バグ管理システムのような使い方やチケット駆動開発のチケットのような使い方も可能である。GitHub では後述する Pull Request が行われた際も、同時に Issue が 1 つ発行される。

1 つの機能変更や修正などに対して 1 つの Issue が割り当てられ、議論や修正などはその Issue を中心として行われる。Issue を見れば、その変更に関する全てを把握することができるのである。

^{*5} <http://github.com/plans>

Git のコミットメッセージに「#7」のように Issue の発行 ID を書き加えることによって、GitHub では自動的に Issue からコミットに対してリンクが貼られる。また、特定のフォーマットに基づいてコミットメッセージを記述することにより、Issue を Close することもできる。非常に便利な機能なので、実践利用できるようになることが望ましい。詳しくは後述する。

1.5.4 Wiki

Wiki 機能は、いつでも誰でも文章を書き換えて保存することが可能なので、共同で文書を作成することができる。開発ドキュメントやマニュアルなどの記載に用いられることが多い。記法には後述する GMF (GitHub Flavored Markdown) を用いることができる。

Wiki ページも Git リポジトリとして管理されており、改版履歴がしっかり残るため、安心して書き換えを行うことができる。clone して編集することもできるため、プログラマがブラウザを立ち上げずに利用することも可能である。