目次

[●2. GitHubを使った文書の変更履歴管理 ＜補足＞ 2](#_Toc429063225)

[**●2.1** **GitHubを利用した文書管理の運用について** 2](#_Toc429063226)

[●2.1.1 運用ルール概略 2](#_Toc429063227)

[**●2.2** **Pull Requestを利用した原本への変更反映依頼（上級者向け）** 2](#_Toc429063228)

[●2.2.1 Pull Requestの概念 2](#_Toc429063229)

# GitHubを使った文書の変更履歴管理 ＜補足＞

## **GitHubを利用した文書管理の運用について**

本解説書で解説したGitHubを利用した文書管理では運用ルールを決めるとともに管理者の存在が必要です。

ここでは運用ルールについて概略を説明しますが、実際の運用に際しては実情に合わせたさらに詳細レベルの運用ルールを検討の上決定し、実施する必要があります。

### 運用ルール概略

GitHubを利用した文書管理を運用するにあたって決定する必要があるルールとして下記を挙げることができます。

* 改変者から送付される差分ファイルを受け取るメールアドレスの用意、周知
* 差分の本文適用を決定するための意思決定プロセス
* 原本リポジトリにPushをできるのは誰にするか（GitHub管理者の決定）
* 原本が改定された際に改変者へ通知を行うか？行う場合はどのような手段を用いるか

これらの点を検討したのち、下記フローで差分反映を実施します。

1. 改変者から差分ファイルを添付したメールが届く。
2. 管理者は原本からクローンした自分のローカルリポジトリへ差分を適用する。
3. 管理者は上記変更が許可できるものであるか確認する（複数管理者が存在する場合は全員の意思を確認する）
4. 変更が許可できないものであれば変更は取り消しとする。必要であれば改変者へ連絡を行う。
5. 変更が許可できるものであれば変更をコミットし、原本のリポジトリへPushする。必要であれば改変者へ連絡を行う。

須藤さん、絵があるとわかりやすいようであればイラスト化可能でしょうか？

## **Pull Requestを利用した原本への変更反映依頼（上級者向け）**

GitHubでは本解説書で解説したメールを利用した原本への変更反映依頼の方法に加えてPull Requestというものを利用し、GitHubのWebサイト上で変更依頼の発行を行うことも可能です。

本章ではわかりやすさを保つため詳細の記載はしませんがPull Requestの概念を簡単に説明します。

### Pull Requestの概念

Pull RequestはGitHubのWebサイト上で原本の管理者に文書の変更依頼を行います。  
そのため文書の変更者もGitHub上にリポジトリを持っている必要があります。

リポジトリの作成は原本のリモートリポジトリからForkという作業を行って複製リポジトリを作成します。この作業は初回のみ実施します。  
Pull Requestを利用する場合のリポジトリの関係は下図のようになります。

GitHub Webサイト



管理者

原本

リモートリポジトリ

（共有書棚）

**Fork**

**Fork**

**Pull Request**

Aさんのリモートリポジトリ

（共有書棚）

写本

写本

Bさんのリモートリポジトリ

（共有書棚）



**Push**

**Push**

Ａさん

写本を基に

変更したファイル

写本

Ｂさん

写本

ローカルリポジトリ

（自分専用書棚）

ローカルリポジトリ

（自分専用書棚）

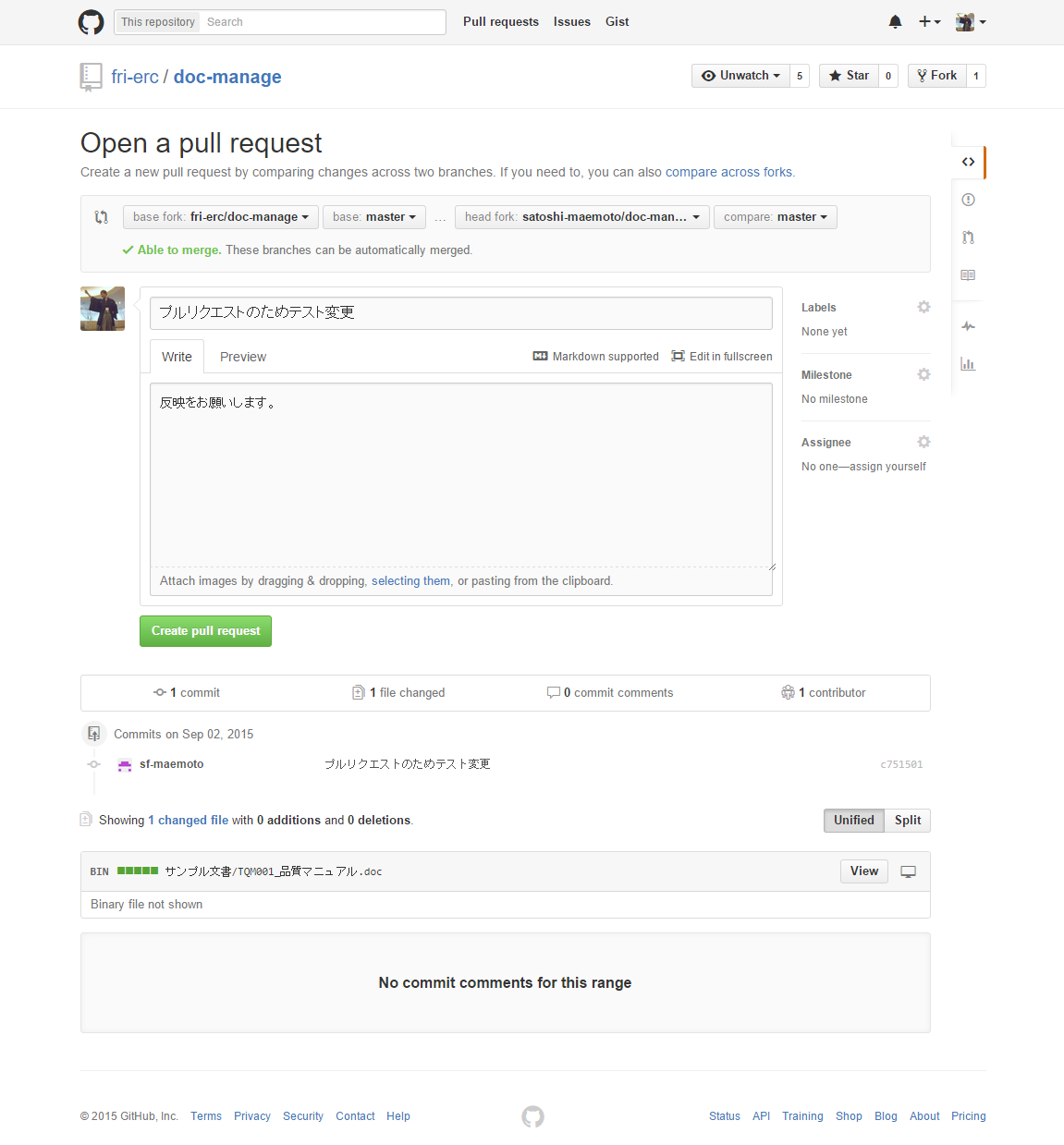
写本を基に

変更したファイル

この図にあるようにローカルリポジトリは原本から直接クローンするではなく、GitHub上にForkで作成した自分のリモートリポジトリからクローンしたものとします。

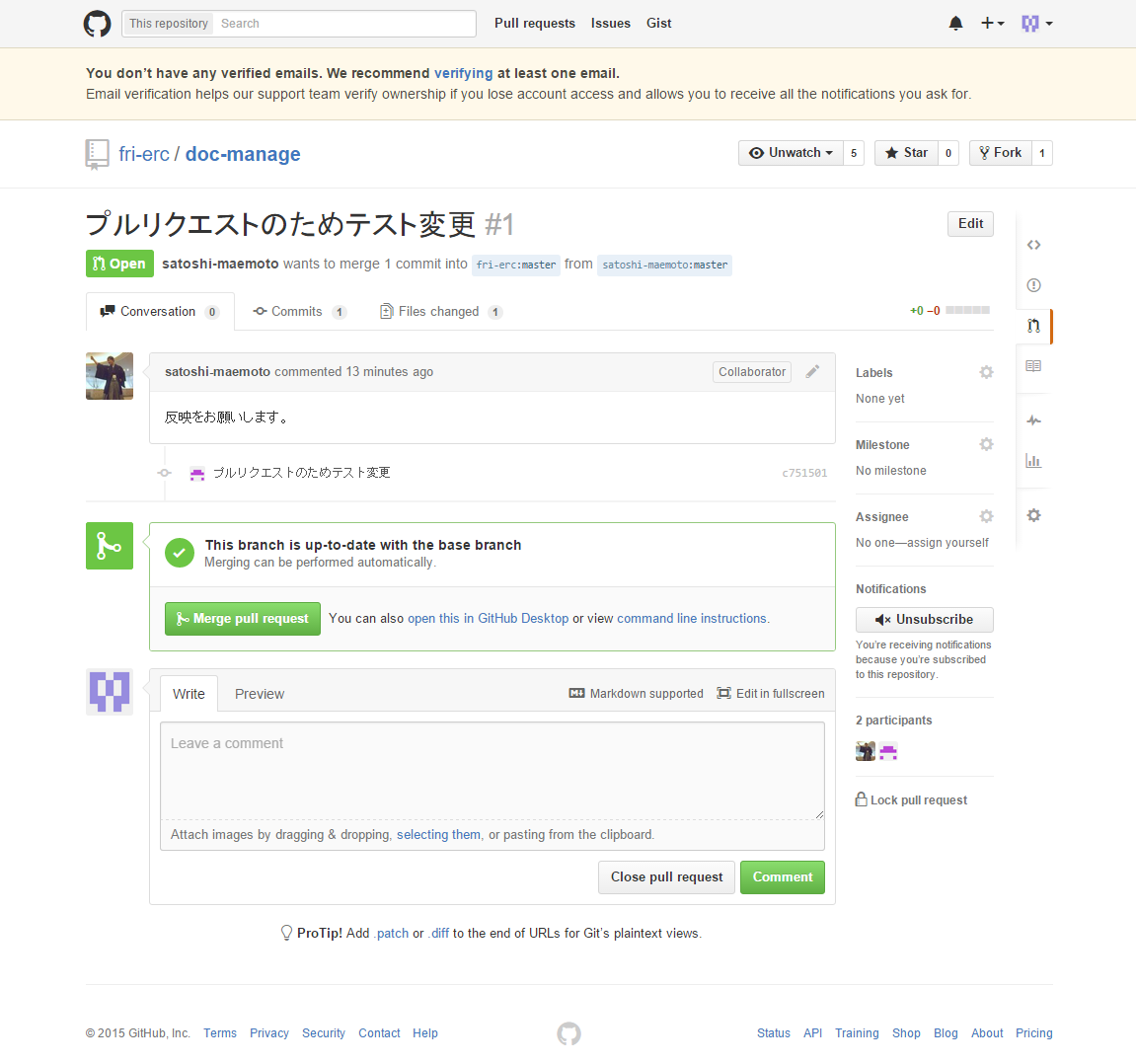
ローカルリポジトリで改変を行った場合は自分のリモートリポジトリへPushを行って改変を反映します。

次いで、GitHub Webサイト上でPull Requestを作成します。



　（図：文書の改変者がPull Requetを発行）

Pull Requestが作成さると原本の管理者に依頼が届き、変更を反映するか破棄するかの検討を行い処理を決めることができます。



　（図：文書の管理者に届いたPull Requet）

## **エラー時の対応について**

ここは結構多岐にわたりますね。。