



# フィードバック

OSS Gate

*OSS Gate*ワークショップ

# 現状確認



1. ユーザーとして動かす
2. ふりかえり
3. フィードバックポイントを発見！
4. ↑を**フィードバック**

# フィードバック



- ✓ upstream（開発元）に  
うまくいかなかったことを報告
  - ✓ ここで詰まった、を伝える
  - ✓ こうだったらよかった、を伝える

# 報告方法



1. 整理する
  - ✓ 自分の考えが文章になればOK
2. **開発者にとって**  
わかりやすくなるように編集
3. 適切な場所に報告
  - ✓ GitHubのissueとか

# 1. 整理する

- ✓ 自分で自分の気持ちを理解
  - ✓ 自分が読んで理解できる文章にまとめられれば理解できている
  - ✓ 自分が理解できていないことは開発者にも伝えられない！
- ✓ 作業メモに追記→サポーター確認

サポーターへ：メモ（断片）の文書化を手伝って  
例：考えを整理できるような質問をする

# 整理方法



## デモ

- ✓ 作業メモを開く
- ✓ フィードバック対象を決める
- ✓ 自分の気持ちを作業メモに追記
- ✓ サポーターに確認依頼

## 2. 編集する

- ✓ **開発者にとって**  
わかりやすくなるように編集
- ✓ 報告方針をまとめているOSSもある  
例：GitHubにあるCONTRIBUTING.md
- ✓ 作業メモに追記→サポーターに確認

サポーターへ：リーダブル化を手伝って  
例：自分が開発者ならこう読めると開発者視点を伝える

# 編集の仕方



- ✓ ポイント
  - ✓ **相手**がわかるように書く
  - ✓ 例：省略しない（具体的に書く）

# 省略例



“インストールしました。  
動きませんでした。  
どうしたらいいのでしょうか？”

# 省略しない例

“ ↓でインストール

```
% sudo apt-get install ...  
(...実行結果...)
```

↑のように失敗しました。

環境：Ubuntu 16.10 amd64 ”

# なぜ省略しないか



- ✓ 相手は私を知らないから
  - ✓ 省略すると**想像**しないといけない
  - ✓ だいたい**想像は外れる**
  - ✓ 話が**噛み合わない!**

# 省略しないとは



- ✓ 詳細を書く
  - ✓ 実行したコマンド・実行結果
- ✓ やったことを書く
- ✓ やって**いないこと**を書く
- ✓ 期待した結果を書く

# 編集方法



## デモ

- ✓ 作業メモを開く
- ✓ 自分の気持ちを開発者に伝えるようにまとめて作業メモに追記
- ✓ サポーターに確認依頼



# 3. 報告する

- ✓ 適切な場所に報告
  - ✓ OSSによって報告場所は違う
- ✓ サポーターへ
  - ✓ 報告に二の足を踏んでいる人の背中を押してあげて  
例：自分が開発者ならこの報告をもらったらうれしい、と開発者視点を伝える

# 報告方法



## デモ

- ✓ 報告方法を探す
- ✓ サポーターに後押ししてもらう
- ✓ まとめた報告内容を報告

# ふりかえり2：デモ デモ

- ✓ ビギナー：
  - ✓ 作業メモを元に説明
- ✓ サポーター：
  - ✓ よいところをよいと言う！
  - ✓ フィードバックポイントを確認
  - ✓ 完了→issueにコメント

# ふりかえり2：進め方



- ✓ ビギナー1人ずつ順番に
- ✓ 対象ビギナーの作業ログをディスプレイに映す
- ✓ 同じ島のみみんなで見ながら