

コードレビューについて

自己紹介

田中智章

経営・ソフトウェア開発やってます

主戦場のWebエンジニアしつつ、Flutterを使ったアプリ開発も。
ウィスキーとLEGOが好きです。

- ・ Ruby + RoR / PHP + WordPress
- ・ エンジニアマネージャ / スクラムマスター
- ・ その他 (SEO / React / NoCode / Bot)

会社紹介

株式会社エバーセンス / 株式会社コジカジ

会社概要

- 家族を幸せにすることで、笑顔溢れる社会をつくる。

おもなプロダクト

- 妊婦さん向けの「ninaru」 / 子育て中のお母さん向けの「ninaru baby」、など

本日の勉強会の目的

勉強会のゴール

- なぜコードレビューするのか？
- コードレビューをより良いものにするには？

目次

目次

1. レビューの目的
2. レビューの心得（される側）
3. レビューの心得（する側）
4. コードレビューの事例
5. まとめ

レビューの目的

レビューの目的

コードの品質を高める

- 仕様を満たしているか
- コードは拡張しやすいか？
- セキュリティ / パフォーマンス / トランザクション、etc

メンバーの技術力の底上げ

- 良いコードの認識を揃える
- フィードバックを通じた育成

Q. コードレビューを実りあるものにするには？

**A. レビューされる側 / する側 両方に
きちんとした心構えや準備が必要**

この勉強会では、「レビューされる側」「する側」両方の視点で、注意することをお話しさせていただきます

レビューの心得（される側）

レビューの心得（される側）

1. 何はともあれセルフレビュー
2. 何をレビューしてもらいたいのか明確にする
3. フィードバックは素直に受け入れる
4. フィードバックは人格否定ではないと理解する
5. 対応しないフィードバックがあれば、理由を述べる

1. 何はともあれセルフレビュー

Bad

エンジニア：

- 「よし、コードつくれた。一通り動いたのでレビューに出そう」

レビュワー：

- 「デバックコメントも残ってるなあ。そもそもコメントも残ってるし...このコードをレビュー以前の問題では？」

1. 何はともあれセルフレビュー

Good

エンジニア

- 「よし、コードつくれた。まずは自分の PR を確認しよう」
- 「あ、デバックコメント残っているな。消しておこう」
- 「一通りセルフチェックはしたので、レビューに出そう」

レビュワー：

- （一通り見て）「特にパットミ変なところはないな。詳細見よう」

※以下、テーマに沿って「Bad / Good」を説明します（繰り返し）

レビューの心得（する側）

1. info / want / must など指摘のレベルを明確に
2. どんな観点でレビューしているか明確にする
3. 論理的にフィードバックする
4. 人格否定はしない
5. 自分のことを棚にあげて、コードに向き合う

1. info / want / must など指摘のレベルを明確に

Bad

```
def get_user_name  
  return 'tanaka'  
end
```

rubyでは「get」をつけないので、外したほうがいいです

1. info / want / must など指摘のレベルを明確に

Good

```
def get_user_name  
  return 'tanaka'  
end
```

[must]
rubyでは「get」をつけないので、外したほうがいいです

※以下、テーマに沿って「Bad / Good」を説明します（繰り返し）

コードレビュー事例

実際に実施している事例を紹介します

- **セルフレビュー**

- かんたんでもいいので、テストケースを作成。一通り流す。

- **レビュー負担低減**

- リファクタする際に専用の PR を作成する

- **デザイン系**

- css や xib ファイルのレビューは困難なので仮組みですすめ、最終デザイナーチェック

- **その他**

- コードスタイルはコード書く以前に決めておく
- 設計（アーキテクチャ、DB、ディレクトリ構成など）も事前に

テストケースの例

- かんたんにテストケースを作成（あくまで自分用）。
- コードチェック後、順番に実施。NG でコードを修正したら、1 からやり直す

	A	B	C	D	E	F	G	H
1	日付				version	count		
2	2018/07/02				v1	7		
3	タスク名				v2	0		
4	freakoutをninaruにいれる（あわせてよみたい）				v3	0		
5	GitHub							
6	https://github.com/eversense/192abc.com/pull/1316				抜け漏れ	count		
7	メモ欄				知識	0		
8					思い込み	0		
9					段取り	0		
10					特になし	7		
11	確認項目							
12	#	ver	分類	項目	開発チェック	本番チェック	メモ	抜け漏れ原因
13	1	v1	ninaru Android	あわせてよみたい GMO *2個でる freakout * 2個出る	done			特になし
14	2	v1	ninaru iOS	#1のiOS版チェック	done		Mozilla/5.0 (iPhone; CPU iPhone OS 8_0 like Mac OS X) AppleWebKit/600.1.3 (KHTML, like Gecko) Version/8.0 Mobile/12A4345d Safari/600.1.4 jp.co.eversense.ninaru/5.10	特になし
15	3	v1	共通	maverick 周り削除チェック	done	none	\$ cd wp/wp-content/themes/ninaru \$ git grep maverick	特になし
16	4	v1	ninaru Android	その他広告デグレチェック ランキング出しわけチェック - GMOはでない (0%) - freakout だけ タグ一覧などのarchive - frekout 例: https://192.168.59.103/tag/lull-to-sleep	done			特になし
17	5	v1	ninaru iOS	#4のiOS版チェック	done			特になし
18	6	v1						特になし
19	7	v1						特になし
20	8							

●● の例

いくつか実例を写真つきで紹介

まとめ

※ 最後にまとめをいれます ※

質疑応答

おわり