

概要

須藤功平

株式会社クリアコード

実践リーダーブルコード

2015-06-24

今日の流れ - 午前

- ✓ 10:00- アイスブレイク
- ✓ 10:15- 概要と進め方の説明
- ✓ 10:45- 実装
- ✓ 12:30- ランチ

今日の流れ - 昼下がり

- ✓ 13:30- 読み方のデモ
- ✓ 13:45- チェンジして実装
- ✓ 15:30- グループふりかえり

今日の流れ - 夕方

- ✓ 16:00- まとめ
 - ✓ 次のステップを説明
- ✓ 16:30- 感想発表・質疑応答
- ✓ 17:00- 有志で懇親会

チューター紹介

- ✓ 参加者のサポート係
- ✓ 現役エンジニア
- ✓ 行動指針
 - ✓ 参加者が目的を見失うのを防ぐ
 - ✓ 新しい視点を与える

講師紹介

須藤功平（すとう こうへい）

- ✓ クリアコード代表取締役
- ✓ リーダブルコード（本）の「解説」の著者
- ✓ 進行と全体を気にかける係

講座の目的

- ✓ 自分の開発チームに
 - ✓ ↑注意：個々人の話ではない
- ✓ リーダブルなコードが
当たり前な文化の作り方を
- ✓ 持ち帰る
 - 「解説」に書いていることの実践方法を学ぶ

目的でないこと

- ✓ 実践前の不安のケア
 - ✓ やらない理由の増幅は抑えられない
 - ✓ ↑のときに外からの声は届かない
- ✓ 例：上司の説得方法の伝授
 - ✓ 時間が残ったら参加者同士で情報交換する場を用意
 - ✓ ↑を活用するのは可

サポート

- ✓ 今日の資料はすべて再利用可能
 - ✓ チーム内で同じ講座を再現できる
- ✓ 無料のフォローアップ面談
 - ✓ チームで実践→悩み
↑の相談に乗る
 - ✓ 受講後3ヶ月以内に1回

そもそもの話

✓ リーダブルコードはなぜ必要か

↓を指すために
チームでの共有は必須

リーダブルなコードが
当たり前の文化

必要なケース

チーム開発

チーム開発

- ✓ 1人しか触れないコード→危険
 - ✓ いなくなったら変更できない
 - ✓ 変更できてもコストが大きい
- ✓ チームで触れるには？

↓
既存コードの理解が必要

既存コードの理解のため

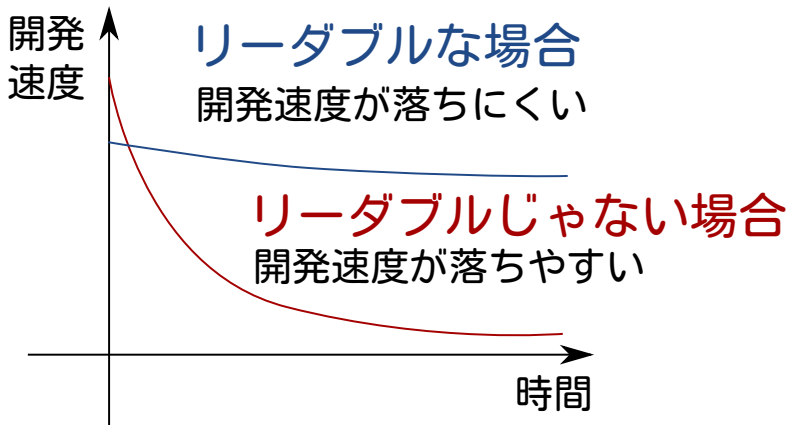
リーダブル
コード

既存コードの理解しやすさ

- ✓ コードの変更コストに影響
 - ✓ コスト↑ → 修正・機能追加の時間↑
(理解しないと変更できない)
 - ✓ コスト↑ → リグレッションバグ↑
(理解しないまま変更すると問題発生)

↓
理解しやすさ → 開発速度に影響

時間が経つほど影響大



リーダーブルコードの必要性

- ✓ チームの開発速度の維持のため
 - ✓ 継続的に改良・修正したい
 - ✓ それも現実的なコストで

必要性の実現方法

コードを読む
文化を作る

読む？書くじゃないの？

- ✓ リーダブルコードを書くにはコードを読まないといけない
- ✓ なぜ？

リーダブルコードは
チーム毎に違うから

リーダーブルコード

「読む人」が
読みやすいなら
リーダーブル

読む人

- ✓ 多くの場合、いない
 - ✓ チームのコードを読んでいますか？
- ✓ 読む人（チームメンバー） 毎にリーダブルの基準は違う
 - ✓ 背景が違うので当たり前
(背景：使ってきた言語・今の知識)

チームでのリーダブル

- ✓ 1つずつ見つけていくしかない
- ✓ 各メンバーの読んだ感覚を
チームで共有
- ✓ 既存の基準をベースにするのはアリ
(基準：本の内容やコーディングスタイルなど)

チームでのリーダブルコードは
育てていくもの

リーダブルの基準の育て方

- ✓ コードを読む文化を作る
(最初の難関)
- ✓ チームのコードの中から
リーダブルなコードを見つける
- ✓ リーダブルなコードを
チームで共有
- ✓ ↑の繰り返しで基準を増やす

コードを読む文化を作る

- ✓ まず自分が読み始める
 - ✓ 仲間がいると心強い
- ✓ リーダブルなコードを探す
 - ✓ 読みにくいコードは今は置いておく
(チームにコードを読む文化ができてから!)
 - ✓ 見つけたリーダブルなコードは…

リーダブルなコードは…

✓ 他のメンバーに教える

(例：話しかける。チャットに書く。Wikiにまとめる。)

✓ 「〇〇さんの△△という書き方、リーダブルでしたよー」



読みやすさの基準を共有
コードが読まれているという自覚

読むことを「当たり前」に

- ✓ 「あいつはコードを読むやつ」という認識を広める
- ✓ 自分だけからチームへ
 - …続きはセミナーの最後に

ワークショップ内容

改良するために
他の人のコードを読む

- ✓ 「まず自分が読み始める」
- ✓ 「リーダブルコードを探す」
(読みにくいコードは今は置いておく)
- ✓ 「リーダブルの基準を共有」
(チームでのリーダブルコードができる)

注意：やらないこと

リーダブルコードを書くための
テクニックをたくさん伝授

テクニック伝授は範囲外

- ✓ 順番が違おう
- ✓ まず読む文化を作ること
 - ✓ 今日は↑がメイン
- ✓ テクニックはその後
 - ✓ 読む文化ができていれば効率的に広められる
 - ✓ よい書き方でコードを書けばみんながコードから学んでくれる！

やること

読む文化作りの 体験

読む文化作り

- ✓ まず自分が読み始める
- ✓ リーダブルコードを探す
- ✓ 他のメンバーに教える

読む文化作りの体験

- ✓ 10:45- 課題を実装
 - ✓ リーダブルコードを書く
- ✓ 13:30- 実装チェンジ→開発継続
 - ✓ 「まず自分が読み始める」
 - ✓ 「リーダブルコードを探す」
- ✓ 15:30- グループ発表
 - ✓ 「他のメンバーに教える」

おさらい

- ✓ 講座の目的
- ✓ リーダブルコードの必要性
- ✓ 講座でやること

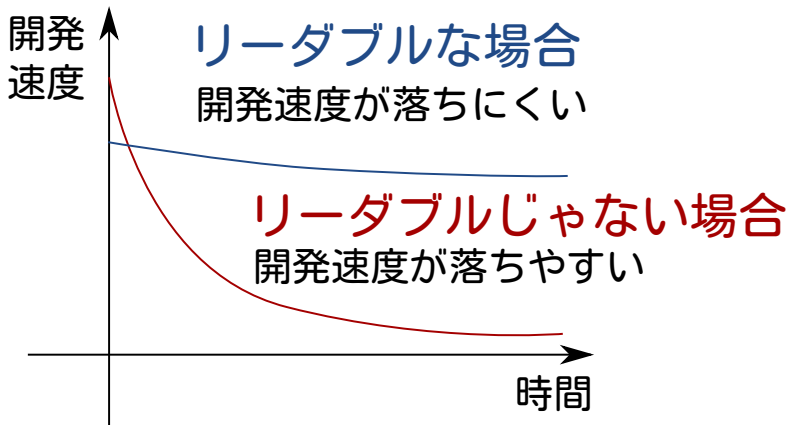
講座の目的

- ✓ 自分の開発チームに
 - ✓ ↑注意：個々人の話ではない
- ✓ リーダブルなコードが
当たり前の文化の作り方を
- ✓ 持ち帰る

リーダーブルコードの必要性

- ✓ チームの開発速度の維持のため
 - ✓ 継続的に改良・修正したい
 - ✓ それも現実的なコストで

変更コストと開発速度



講座でやること

- ✓ コードを読む文化作りの体験
 - ✓ まず自分が読み始める
 - ✓ リーダブルコードを探す
 - ✓ 他のメンバーに教える