

まとめと次のステップ

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

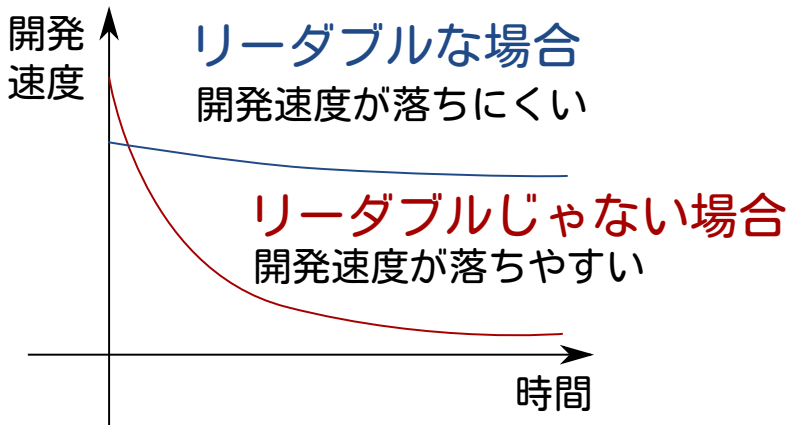
講座の目的

- ✓ 自分の開発チームに
- ✓ 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

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



文化の作り方の流れ (1)

- ✓ チームでリーダブル基準を育む
 - ✓ 「読む人」が読みやすいならリーダブル
 - ✓ 「読む人」が変われば基準が変わる
 - ✓ →読む人が違うのでチーム毎に違う

文化の作り方の流れ (2)

- ✓ 基準の育て方
 - ✓ 各メンバーがコードを読む
 - ✓ リーダブルだと思ったコードを共有
 - ✓ チームとしてリーダブルかを判断
 - ✓ →チームの基準に加わる

基準の育て方 (1)

- ✓ コードを読む文化を作る
 - ✓ まず自分が読み始める
 - ✓ リーダブルなコードを探す
 - ✓ 見つけたリーダブルなコードを他のメンバーに伝える (後述)
- ✓ →コードが読まれるという自覚がチームに浸透

今日やったのはここまで

基準の育て方 (2)

次のステップ

- ✓ コミットを読む
 - ✓ コード全体ではなく差分を読む
 - ✓ 設計の仕方ではなく開発の仕方が見える
 - ✓ リーダブルなコードを見つけるには適切

基準の育て方 (3)

- ✓ コミットの読み方
 - ✓ pull型よりpush型がよい
 - (Git用のオススメツールあり: git-commit-mailer)
 - (Subversion用のツールもあり: 同梱されている)
 - ✓ 読むコストが下がる
 - ✓ 流し読む (負担が多いと続かない)
 - ✓ 問題探し視点では読まない
 - (必要ならコードレビューを実施)

基準の育て方 (4)

- ✓ リーダブルコードの伝え方
 - ✓ Wikiに書く (全チームで有効)
 - ✓ コードで伝える (上級チーム向け)

基準の育て方 (5)

- ✓ Wikiに書いて伝える
 - ✓ Wikiもdiffを通知できるようにする
(RedmineとGitHub用はツールあり)
 - ✓ 後で参照できるため
 - ✓ 更新もできるため
(リーダブルコードの基準は変わることもある！)

基準の育て方 (6)

- ✓ コードで伝える
 - ✓ 上級チーム向け
(チームにコードを読む文化が根付いた後)
 - ✓ リーダブルコードを真似てコミット
 - ✓ →他の人：「またこの書き方だ」
 - ✓ →真似する人増加→チームが合意
 - ✓ →チームが合意→Wikiにまとめる

コードを読む文化

- ✓ 新人の受け入れにも有用
- ✓ 人の入れ替えにも有用

どうして有用か

開発を通じて↓を伝えられる
(ただし、上級チームになってから)

- ✓ チームが大事にしていること
- ✓ チームのリーダーダブル
- ✓ チームの開発スタイル

コードを読む文化

チームが大事にしていることを
開発を通じて伝えられる



- ✓ 新人の受け入れにも有用
 - ✓ 開発速度低下を抑えられる
- ✓ 人の入れ替えにも有用
 - ✓ リーダブル見直しのよい機会

これからやること

- ✓ この講座をチームでもやる
 - ✓ 資料はすべて再利用可能
- ✓ 自分がコードを読み始める
 - ✓ 自分が変更するコードの周辺からリーダブルコードを探す
 - ✓ 見つけたリーダブルなコードを他のメンバーに伝える

サポート (1)

- ✓ 今日の資料はすべて再利用可能
 - ✓ チーム内で同じ講座をできる
- ✓ 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、1回限り

サポート (2)

- ✓ コミット読みの支援
 - ✓ OSSとしてツールを公開
 - ✓ コミット毎にメールで通知 (diff入り)
 - ✓ Git、Subversionで使える
 - ✓ GitHub、GitLab連携もできる

お知らせ

コードリーダー育成支援

<http://www.clear-code.com/services/code-reader/>

リーダブルなコードが
当たり前な文化づくりを支援

クリアコード

- ✓ クリアなコードが大切
 - ✓ クリア == clear == 意図が明確
 - ✓ クリアなコードはリーダブルコード

みなさんのチームが
リーダブルコードが当たり前な
チームになることを応援します！