

チケット駆動で大切なことは オフショア開発が教えてくれた

2022年11月5日 redmine.tokyo 第23回LT

ぼうこば <@bowkoba>

登壇者紹介 ぼうこば

- 元 製造業勤務
- SWプロジェクト管理者
- Redmineオンプレ管理者
- Redmineエバンジェリスト
- 帽子大好き
- 焼肉大好き




オフショア開発とは

主にソフトウェア開発等を、(安い?)海外企業や海外現地法人などに委託発注すること

- メリット:リソース確保、コスト削減、高い技術力
- デメリット:**コミュニケーションコスト大**



オフショアのコミュニケーションで発生した問題

1. レビューの指摘が直っていない
→リアルタイムMTGで指摘を聞き取れていなかった
 2. コーディング確定前にテストを進めていた
→開発プロセス意識が低いメンバーがいた
 3. アウトプットが期待と異なっていた
→条件が曖昧だったり、前提説明を怠った結果、本来の目的から外れたアウトプットになっていた
- 

コミュニケーションで発生した問題の対策

1. レビューの指摘が直っていない

→リアルタイムMTGで指摘を聞き取れていなかった

リアルタイムMTGはやめて、すべての依頼とレビュー指摘をチケット化した結果、対応漏れがなくなった。(聞き直すことを躊躇していたオフショアメンバーから好評を得た)

(補足)リアルタイムMTGの落とし穴

そのリアルタイムMTG、本当に必要ですか？

- 口頭ではきちんと伝わらないことが多い
手っ取り早いと思っていることが落とし穴の場合も...
情報のやり取りはチケットでやりとりしたほうが正確
- リアルタイムMTGはコスト大
実施するなら、顔合わせや雑談など、業務と異なる価値を念頭においたほうがいい(その意味では定例で短時間実施は良策かも)

コミュニケーションで発生した問題の対策

2. コーディング確定前にテストを進めていた
→開発プロセス意識が低いメンバーがいた

開発プロセスをRedmineのワークフローとロールで定義し、勝手に作業完了できないルールを周知させた結果、プロセス違反がなくなった。(開発プロセスがあるだけでは不足、具体的に何をどうすればよいのか、ツールで具現化することが大切)

コミュニケーションで発生した問題の対策

3. アウトプットが期待と異なっていた

→条件が曖昧だったり、前提説明を怠った結果、本来の目的から外れたアウトプットになっていた

Redmine Issue Templates pluginを活用し、必ず概要、背景、クローズ条件、期日の理由を書いてもらう**チケット記載項目の明確化と画一化**活動を行った結果、齟齬を大幅に減らすことが出来た(実は依頼している側がいかに適当だったか身に沁みて理解できた)

まとめ

- オフショア開発をきっかけに、Redmineを始めとするチケット駆動開発の有用性に気付かされた
 - だいたい分かるだろう、言わなくても分かるだろう、をチーム内で排除していく活動が大切
 - これらは結局、オフショアに限ったことではなかった
 - 前職でお世話になったオフショアメンバーへ本当にありがとう!
- 