開発のためのテスト

藤江　鉄平

自己紹介

- 藤江鉄平(ふじえてっぺい)

システムエンジニア(5年)

問い合わせ対応エンジニア(3年)

QAエンジニア(4年)

システムエンジニア(3年)　←ｲﾏｺｺ

- 福岡でシステム開発をやってます。

開発で同じ経験の方居ませんか

- 設計書ってありますか？
  ⇒『**そんなものは無い**』
- 実装した方は？
  ⇒『**退職済です**』
  ⇒『**海外のオフショアなので不明**』

今回の話

- 設計書無し
- 既存の経緯を知っている人無し
- 受注時点で**納期が厳しい**

の状態で、既存ソフトウェアに対して改修して欲しい。
社内の新人にソフトウェア開発のやり方を教えて欲しい。

と言われたときに
探索的テストとテスト項目作成先行で何とかした話

- 設計書も無くて、実装を知る人も居ない

⇒手探りで調査するしかない。

⇒**探索的テスト**を使おう。

- 新人教育も良い感じにやって

⇒実装能力、設計能力も未知数で既存が壊されないか心配

⇒ソフトウェアの振る舞いだけは意識を合わせたい。

⇒**先にテストを書けばいいじゃない**。

実際の作業の流れ

- 効率を上げるための作業手順説明
- 既存ソフトウェアの動きを**探索的テストで調査**
- ソフトウェアの動きと要望内容から修正方針を設計
- 設計内容に基づいて**テスト項目を作成**
- テスト項目が全部通るように実装(一部ペアプロも実施)
- 実装完了後に一通りのテスト実施
- 設計内容で更新漏れがあったら更新
- 納品、**振り返りの実施**

探索的テストなら

- 全ての機能を１から確認する時間が無い。

⇒工数、コストを抑えやすい

- 修正に影響が出そうな箇所の動きを知っておきたい。
- 提供された操作説明書が信用できない。

⇒ドキュメントが不十分でも実施できる。

探索的テストのデメリットへの対応

- 属人性の高さ

⇒新人一人でのテスト時間とペアテストの時間を作った

- テストの網羅性の確保

⇒変更箇所に関係ある操作テストに限定

- 管理のやりにくさ

⇒厳密な管理をしない。(リスク許容)

確認工数を操作項目数で割って制限時間つけただけ

探索的テストの結果

- 既存ソフトウェアの動きを手を動かして理解させた。
- テストの際の確認ポイントを教えることができた。
- 実際の動きをドキュメントとして残し、仕様書を少しずつ整えることが出来た。
- 発注者も存在忘れてた既存バグがあった。

テスト項目の作成を先行

- UIとロジックが癒着してて、追加部分だけの単体試験コードを記載する方法が取れない。
- 既に動作確認できるソフトウェアがあるからTDDに近い形で実装を進めていきたかった。

テスト項目作成を先行させた結果

- 実装の前にこう動くように変えるという認識を共有できた。
- 設計内容説明⇒実装ロジックの説明⇒実装後の動作想定説明
  ⇒一連の流れとして説明できた。
- 実装⇒動作確認⇒修正のサイクルを回して常に既存が動く状態を作った。

※今まで作ったコードは問題ないことを常に確認できる

⇒**心理的安全性を確保**

- ~~実装ミスの相談が重なって死にかけた。~~

開発してみての振り返り

- 探索的テストとテスト項目先行作成を行うことで仕様理解のスピードが早まったと思う。
- 修正の途中段階で確認項目も作ったほうが新人視点だとやりやすかったかもしれない。

　A⇒Bと修正したときにどこで間違えたのか把握できていなかったため、A⇒A’⇒B‘⇒Bといった形で途中段階を指導することで最終的に解決させることが出来た。

まとめ

- ドキュメント無し、既存の実装者無しの**無法地帯**でもテスト技術を活用して開発に活かすことができた。
- テスト技術に**触れてもらう**ことで、技術に**興味を持たせる**ことができた。
- 自分ですぐに問題に気付けるように**心理的安全性**を確保することで、相談を気軽に行ってもらえるようになった。

以上で発表を終わります。
ご清聴ありがとうございました。