

いつまでテスト期間
つくっているの？

- 実践編 -

自己紹介

- @otf
- フレームワーク / ライブラリ開発 基盤チーム 4年目

基盤チームの紹介

- PO 兼 SM
 - @kyon_mm(この前に発表してた人)
- Team Member
 - @bleis-tift, @otf(私), @zakky_dev
- 主な仕事
 - Web APIを作るためのフレームワーク・ライブラリ受託開発(.NET Framework & Azure)

お客様の関係ベンダーはもちろん自分たちもそのフレームワーク・ライブラリを使う

悪しき習慣

- 開発工程の最後に統合テストする

チーム内ではスクラムの言葉を使っているが、実態は、ウォーターフォール型開発

計画の例

スプリント 1
スプリント 2
・
・
テストスプリント

悲惨な結果に

開発工程の終盤になってバグが大量に見つかる。
→いつリリースできるかわからない状況に

なぜこうなった？

- テストが完了するまでは本当の進捗は分からない
- 統合テストには工数がかかる
- モジュール単位で「完了した」と思ってもモジュールを組み合わせた後、End 2 Endで動かすと動かない

どうしたか？

- テストスプリントをなくし、ストーリーごとにテストする
- 統合テストではないアプローチってテストする
- リリース可能にするまでストーリーを完了と見なさない
 - スプリントレビューでテストもレビューする

Specification By Exampleの例

Scenario: 利用者はバックグラウンド処理を一分間隔で実行できる
Given バックグラウンド処理として一分間隔のタスクを開始
When 10分待つ
Then タスクは10回実行されている

探索的テスト

- セッションベースド(30分単位)
- 少しだけ構造化
 - テストの対象
 - テストの目的
 - テストの方針
 - テストの結果となにに気づいたか

セッションの例

テストの対象

* メール機能

テストの目的

* 利用者がメールしようとしたときなんらかの理由で実行に失敗してもリトライできるか

テストの結果

* メールサーバーが異常終了しても例外をなげなかった

* メール機能をテストするためのスタブサーバーをつくるのが大変だった

なにを学習していくか

- プロダクトの良いこと・悪いこと
- 顧客
- チーム
- 環境

ユーザーガイド駆動の開発

- プロダクトを修正するときは、まずユーザーガイドから
- ガイドが含むサンプルにも自動テストを

オーナーシップ

- チームメンバーの全員がテストする責任を持つ
- スプリントイベントのファシリテーターを交代制にする。特にレビュー
- スプリントのゴールをドラマ仕立てとして共有
- バグの特徴付け

ドラマ仕立ての例

第20話 フレームワーク新バージョン登場! つくったアプリを簡単に配信できるように!

バグの分類付け

バグを工程でわけるのでではなく、どういう特徴があるのかで分ける

- 要求とズレている
- 利用者が修正の反映に5人日以上かかる
- 第一印象が悪い
- 意図が伝わらない
- 利用者に回避方法がない
- 利用者からの指摘
- などなど...

バグの分類付け例

- 利用者が、リクエストからマルチパートをリードするとき、データ量に対して時間効率がよくない
 - 要求とズレている
 - 利用者に回避方法がない
 - 利用者からの指摘

結果

- 20%占めていた手戻り工数が5%~8%に
- 大体定時で帰れるように

まとめ

- アジャイルなテストを実現するために自分たちを変えてきました。
- そのために特に意識したのは
 - リリース可能
 - 学習
 - オーナーシップ

質疑応答

@kyon_mmか@otf どちらに答えてほしいかも言ってください。
い。

ご静聴ありがとうございました