

普段思っている『いいな!』の言語化

朝会

- 「生々しい今の状況」が分かる (単なる進捗確認ではない)
  - 今の問題にみんなが注目し、みんなで知恵を出す (問題を具体的に解決する場でもある)
  - 今日はこのチケットをテストしよう (優先すべきテストはどれ)
  - いま、このバグを直しているから、このチケットはそれが直ってから確認しよう
- 作ろうとしている、作っている機能の目的が分かる
  - 要件定義の段階から、そこにいるから
  - 常にユーザーを意識したテストができる (的外れなテストがしにくくなる)
  - テストにバリエーションがつけられる (ここを直したら、この機能にも影響しそうだな)
- どのような設計、実装で作られているのかを知れる
  - テストに強弱をつけるときの材料にしている (設計の複雑さ、実装の難しさから想像していく)
  - 意識しなくても、自然にグレーボックステストができてしまう
- 作業中の混乱や炎上、急いで作った感などを感じ取れる
  - やはりそういうところはミスしやすい (ここを落ち着けて、念入りにテストするよ)
- 「例えば、そこでこんな操作をしたらどうなりますか?」
  - モノが完成する前に、エアテストができる、バグが出せる、直せる

テスト中

- あれバグかな?と思ったら、すぐにプログラマーに見てもらえる
  - バグではなく、仕様かもしれない → 仕様かどうか調べる時間はもったいない
  - まずは再現手順を確認してから → そのときの生々しい状態でしか取れない情報がある
- 分からないことは、すぐに聞ける
  - 調べれば分かるかもしれないけど、たいてい誰かが知っていて、聞いた方が断然早い
  - 「誰かの分からない」や「誰かの問題」を「みんなの問題」にしている
  - 話しやすい雰囲気
- みんなが何をしているかが分かる (全員同席)
  - 朝会では問題なかったストーリーが、午後になったら炎上しているぞ! (テストの順番を臨機応変に変えていく)
  - 他のテスターがどんなテストをしているのか (とても勉強になる、そこからテストを膨らませる)
- ベアテスト
  - 超本気のテストをしたいとき (クリティカルなバグを見つけたとき) に効果がある (4つの目、2つの脳)
  - 複雑なテストで見つけた不具合も再現性が高く、ふたりで見ているから安心 (バグの共有、次のTestingのInput)

毎日最新版でテストする

- 調査する範囲が狭くなる
  - 「バグピンポン」の抑制
  - 昨日入れたバグを今日見つけることができる
- レグレッションテスト
  - 探索的テスト (積極的に新しいバグを探しに行く)
  - 既存機能に対するテスト
- いろいろな人の視点で (その機能を作っていないメンバーも、ベテランも、新人も)

全員同席

- ひとりでは悩んでいる時間が短い
  - プログラマー同士で話し合っている風景をよく見る
  - 設計や実装のミスに早く気づくことができる
- クラッシュ (ハングアップ) したら、周りの人たちが集まってくる
  - 困ってないかな? (気軽に声をかける、かけてもらえる)
  - みんながみんなの仕事にとても敏感
- ノリノリでテストしていると、休憩をおすすめされる (過集中を抑制)
- メールが見れない環境だから、自ずと話すことになる
  - 仕事ではメールしない (Labo)

全ての工程に全員が関与する

- テストができるくらいに、完成のイメージが具体化される
  - はじめからテストのことを考えている
- テストしづらいものは作らない
  - 全員でテストを大切にしている
  - テストチームという概念がない
- わたしはコーディングと単体テストをしないのが特徴
  - 見逃してしまったミスを、後半で訂正できる (プロセスのせいじゃない)
  - 全ての工程で誰でもダメ出しができる
- 嫌なニュースでも、嫌な顔をする人がいない (みんなを萎縮させない)