違和感のつかまえかた

組み込みシステムの開発者（テスター）としてやっていること

JaSST Review ‘19 / 2019.11.01

ふかやみわ

風邪のひきはじめに違和感を感じたら症状がひどくならないよう行動を変えるのと同じように、開発の現場で違和感を感じたらより良い製品を作るために行動を起こさなければならない。

お話しすること

- 大事なこと
- 何を見て違和感をつかまえているのか
- 違和感に対する感性を高めるには？
- 違和感をつかまえるためのコツ・工夫
- 違和感をつかまえた後のこと

今日お話しすることはこんな感じです。どれも私の経験に基づいたものであり、思考法や発想法のような専門的な話はないので気楽に聞いてください。開発現場で感じる「感じ」や「考えていること」をそのものをお伝えしたくて、過度な抽象化はしないことにこだわりました。

今よりももっと違和感に敏感になりたい
違和感をもっと製品開発に活かしたい

そんな方の参考になればうれしいです

私について

- 医療機器（自社製品）のソフトウェア開発に従事
  - eXtremeなチームに所属する開発者（テスター）
  - 動かして試すのが好き
- とちぎRubyの勉強会（toRuby）
  - とちぎRuby会議 #toruby、とちぎテストの会議 #toteka、オトナとRuby
- JaSST Kyushu（2019.11.29）@ 福岡（九州初上陸）
  - 第一部「チケットシステムで理解するあのチーム」
  - 第二部「フレーズで体験するあのチーム」

http://miwa719.hatenablog.com

https://suzuri.jp/miwa719
チョットデキルTシャツが買えます

いい肉の日

http://www.jasst.jp/symposium/jasst19kyushu/query.html

あのチーム：私たちのチームは一部の人たちからは「あのチーム」と呼ばれていて、熱狂的なマニアがいます。

開発者のロール

- 欲しいものを実現する人たち
- 仕様を具体的にする
- 実装を考える
- プログラムを書く・直す
- テストする

miwaメソッド（字幕）

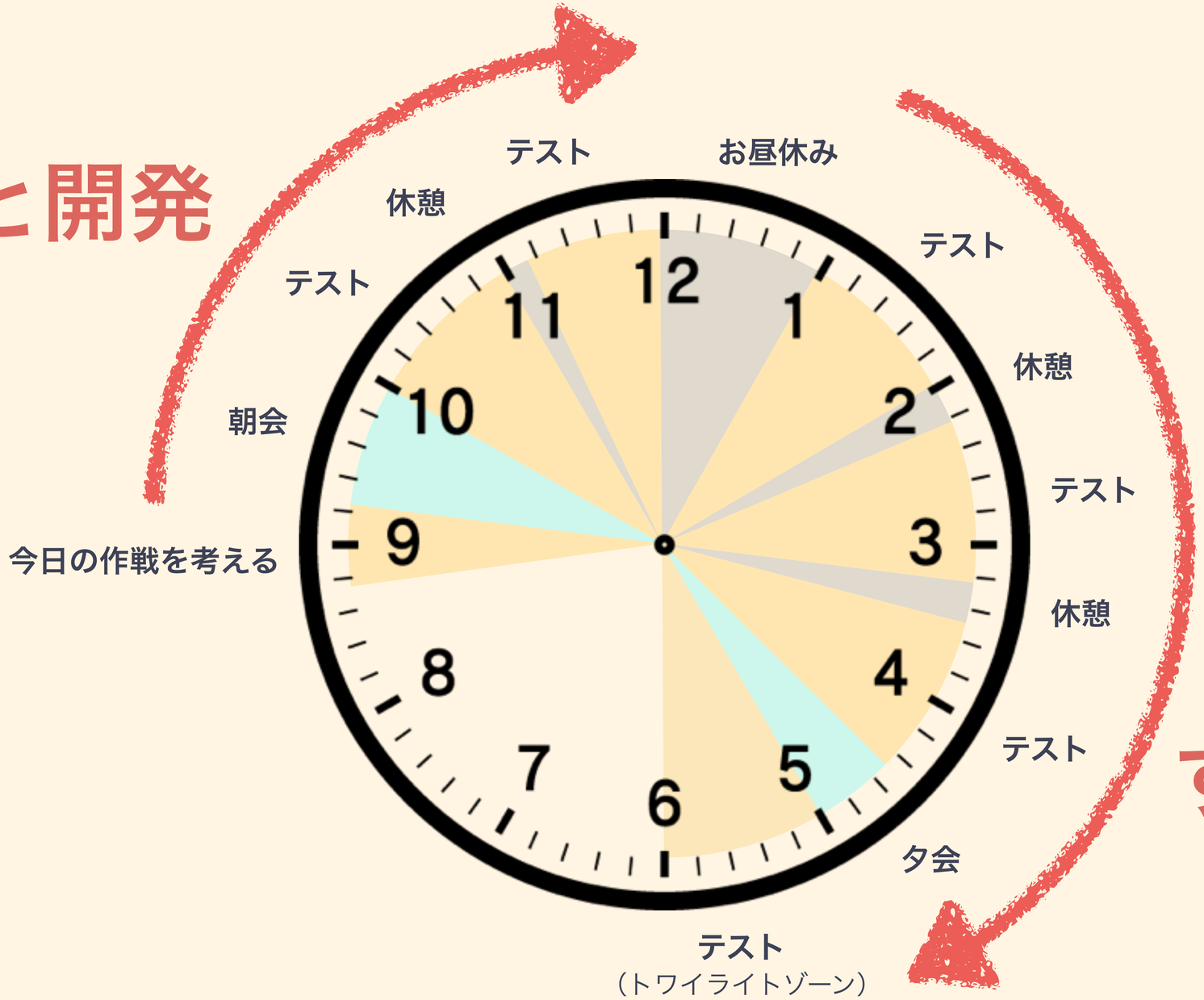
私たちのチームは開発者のロールとテスターのロールが特徴的なので紹介します。まずは開発者（プログラマー）のロール。＞　一番下のテストする、はChecking（設計の一部のようなテスト）というよりは、まだ見つかっていない未知の問題を探しにいくテストです。プログラマーも毎日Testingしています。そしてテストがうまい。そのバグよく見つけたなー、素晴らしい！と思うことがしばしばあります。

テスターのロール

- 欲しいものを実現する人たち
- 仕様を具体的にする
- 実装を考える
- ~~プログラムを書く・直す~~　←ここがない
- テストする

テスターのロールはこんな感じ。プログラムを書かないだけで同じです。プログラムを書かない分、開発者よりも「気にかけられる領域が広い、深い」特徴があります。その結果「よい製品」の基準をイメージすることが得意です。テストの部分だけを取り出して専門的にやるのではなく、お客さまの欲しいものを実現するために、はじめからプログラマーと一緒になって作っていく。私たちのチームのテスターはそういうロールです。

ずっと開発



ずっと開発

私の１日の過ごし方。休み時間以外はだいたいずっと開発です。何か気になることがあると同僚のプログラマーやテスターに聞いたり、見せたり、一緒に考えたり、試したりしてます。他のチームの開発者のところに行き、質問したり相談するときもあります。

レビューはいつするのか？

- うちのチーム、レビューしてない
  - 毎日そこら中でディスカッションはしている

講演の依頼をいただいた時に
一番はじめに確認した

↓

- 明示的なレビューはしなくなってしまった

今日はReviewのシンポジウムですが、うちのチームはレビューしてないのです。来週の月曜日にレビューしましょうとか、そういう会話は聞いたことがありません。毎日そこら中でディスカッションはしているんですけど、これがレビューと言えばそうなのかもしれない。いつもやっているから明示的なレビューはしなくなってしまったんですね。これはどういうことなのかをエクストリームプログラミングの文脈で解説してみます。

eXtremeなチーム

- コードレビューが良いものならば、常にコードをレビューする（ペアプログラミング）
- テストが良いものならば、常にテスト（単体テスト）を行い、顧客でさえもテストする（機能テスト）
- 設計が良いものならば、設計を全員の日常の作業の一部にする（リファクタリング）

9.3 エクストリームプログラミング より抜粋

“最良の開発プラクティスを最大限適用する”

（もし、Pが良いものであれば、そのPをもっと広範囲に、適用しよう）

レビューが良いものならば、常にレビューする

これだ！！

これはレビュー以外にも当てはまります。開発（当然テストも含む）の計画や計画の見直しや調整、ペアプロ、ペアテスト、ふりかえり…。自分たちにとってだいじであると思っていることは特別にやるのではなく常にやる。実際毎日やっているのでそれが当たり前のことになっています。

今日の話は

- 自分たちの製品のおかしさをつかまえる話
  - おかしさ ＝ 違和感 ＝ 注意すべきシグナル
    - 風邪をひくのは自分、自分たちの製品、チーム
    - 誰かが開発してるあの製品をレビューする話ではない
    - 立場や状況が噛み合わなくて聞きづらい人がいるかも

＞他のチームのレビューに呼ばれることがありますが「自分の製品だとしたら」というつもりで参加しています。

大事なこと

●○○○○

- 違和感とは何か
- 違和感を感じるために必要なこと

みなさんと一緒に考えてみたい！

風邪のひきはじめに感じる違和感

- なんとなくだるい
- 寒気がする
- 熱っぽい

車の運転中に感じる違和感

- ガタガタと振動がする
- カラカラと音がする
- 変なニオイがする

違和感とは何なのでしょうか？

違和感とは
いつもの状態と今の状態との差分である

- 風邪のひきはじめに感じる違和感 → いつもの状態と今の状態との差分
- 車の運転中に感じる違和感 → いつもの状態と今の状態との差分

試着室で感じる違和感

- あれ？
- しっくりこないな ...

違和感とは何なのでしょうか？

違和感とは
理想や期待値と現実との差分
である

- 試着室で感じる違和感 → 理想や期待値と現実との差分

違和感とは
いつもの状態と今の状態との差分
理想や期待値と現実との差分
である

＞違和感を感じるために必要なことが見えてきましたね！

違和感を感じるために必要なこと

①いつもの状態を知る
②理想や期待値を持つ × ③今の状態（現実）を知る

- 自分たちの製品
- 仕事の進め方
計画・作戦
- あの問題
- 同僚の様子
- うまくいったら
どうなるの？

いつもの状態を知らなければ違和感を感じることはできません。何かしようとしたときに「こうなるはずだ」というような理想や期待値を持っていなければ、違和感を感じることはできません。そして今の状態（現実）を知る。開発の現場に展開してみます。自分たちの製品がいまどうなっているのか、仕事の進め方や計画・作戦は、昨日のあの問題は今日はどうなっているのか、同僚はいつもと変わらないだろうか。私たちはうまくいったら今日はどうなるのか。どこに向かってゆこうとしているのか。感じたいのは自分なのだから、自分の目で見て、自分で考える。それが大事です。

何を見て違和感をつかまえて
いるのか

○●○○○

いつ、何を見ているのか

- 朝会やディスカッションの場で
- 製品を見る
- チケットを見る
- コミットログを見る
- 自分自身を見る

いつ、何を見て違和感（おかしさ）をつかまえているのかを紹介します。違和感そのものや、違和感になる前の兆しのようなものもあります。注目しているものや、実際にやってること、そのとき何を考えているのか（思考側）についても言語化しました。

朝会やディスカッションの場で

- チームのようす、雰囲気、みんなの態度

私が朝会やディスカッションの場で何を見ているかをお話しする前に、私たちのチームの朝会やディスカッションのようす、雰囲気、みんなの態度について紹介します。これから話す私の思考や行動はこれらの仕掛けや周りの人たちのふるまい（場）が作用しているかもしれないと思ったから。
このような環境が揃わないとダメという話ではないのでお間違えなく。

朝会やディスカッションの場で

- 朝会のようす
  - チケットをプロジェクターに映しながら司会係が音読する
  - 解決しようとしていることに興味がある
    - 自分ならどう解くかを想像しながら、実際との差分を探している
    - まだ見つかっていない問題を見つけようとしている
    - テストのネタ探し
    - 本当にそれがやりたいことなのか？（要件定義に戻るときもある）

＞やっていることの意味を考えながら作っているのを感じます。作っていくうちにすこしずつズレていってしまうことがあるんですね。お客さまは本当は何をしたいのか、何に困っているのか、それが実現されると何が嬉しいのかがよく話題になります。作っている最中にも新たな疑問や課題が出てきます。これだと別のあの機能のやりたいことと矛盾してない？とか。みんなで納得いくまで話しています。

朝会やディスカッションの場で

- 朝会のようす
  - みんなの脳を毎朝同期
    - 脳がプライマリ（知識、経験・訓練して獲得したもの）
    - チケットやドキュメントはセカンダリ（着想を助けるもの）
  - 濃密で濃縮された時間が流れている

「朝会のこと」https://druby.hatenablog.com/entry/20150816/p1
「朝会について」http://vestige.hateblo.jp/entry/2019/02/16/000118

＞チケット駆動開発という言葉があるけどチケットが開発を駆動している感覚はないです。チケットはあくまでも補助的なもので、主は人間の脳側。朝会後はかなり疲れています。逆に疲れていないときは頭を使っていないということだから異常な状態。うちのチームの人たちが朝会についてそれぞれの視点で記事を書いてます。よかったら読んでみてね。

朝会やディスカッションの場で

- ディスカッションのようす
  - 日々そこら中で同時多発
    - 全員同席（プログラマーもテスターも）
    - 製品、コード、テストを見ながら、触りながら、書きながら
  - 対話を大切にしている
    - そんなに簡単には伝わらないだろうと思っている

ディスカッションは日々そこら中で同時多発しています。私たちのチームが全員同じ場所で開発してるという環境もあると思います。プログラマーがペアプロしてる隣でテスターが意地悪なテストをしているという状況です。ドキュメントを見ながらディスカッションはしません。みんなの視線の先には製品、コード、テストがあります。それから対話を大切にしています。伝えることはむずかしいです。とりあえずドキュメントに書いたからOKとか、メールしたから大丈夫のような感覚はないです。非同期のディスカッションは私たちのチームでは喜ばれません。

朝会やディスカッションの場で

- どのようなことを試したいかを考えている
  - 作りたいものに対して何を確認したらそれが正しく作れたと言えるのか
  - 気になることはその場で話題にする
    - 試してみたいことを話す（出し惜しみしない）
    - 設計、実装が変わることがある

朝会やディスカッションがどのような場なのかを紹介しました。このような中で私が何を考えているかを紹介します。＞
気になることをその場で話題にするのはとても良いことです。取り越し苦労みたいなときもあるけど、それはそれでよくて、というのは、少なくとも自分は何かを知ることができます。試してみたいことも出し惜しみしないで話します。その結果、設計や実装が変わることもあります。テストは問題を検出する能力が非常に高いので、実際にモノができた後だけにテストを使うのはもったいない。
出し惜しみしないは、わざと書きました。結構むずかしくて私も練習してできるようになりました。

朝会やディスカッションの場で

- どうやったら壊せるかを考えている
  - どのような問題を見つけたいのかを脳内で描く
    - 起きたら嫌なこと、が常に頭の中にある
  - 具体的に何をしたらその問題が出せそうか
  - エアテスト（脳内でテスト実行、カジュアルなテスト設計）

これから作ろうと思っているモノに対し、どうやったら壊せるかを考えています。それを考えるためにどのような問題を見つけたいのかを頭の中で描きます。前提として「起きたら嫌なこと」が常に頭の中にあります。例えば、撮影した画像の左右が反対に表示されるのは大問題なんですが（医療事故につながる可能性がある）具体的に何をしたらその問題が出せそうかを考えています。操作だったり、装置の状態だったり、データのバリエーションだったり。テストするときのデータが左右対称だと間違っていても気がつかないから、左右非対称なデータを使わないと！とか。頭の中でテストしてしまうときもあります。エアテストですね。プログラマーが頭の中でコーディングするのと似ているかもしれません。
これも出し惜しみせずモノが出来上がる前に話します。プログラマーはそれに耐えられるようなコードやテストを書いてきます。ありがたいー。

朝会やディスカッションの場で

- 話題になっていないことは何かを考えている
  - できること→できないこと、できないこと→できること
  - 似たような機能、同時に動きそうな他の機能は？
  - 製品全体として何が変わる／変わらないのか、どこに作用するのか
  - 思考の軸を変えながらどんどん広げていく
    - 他のチームが作った部分
    - 接続可能な他の装置、システムのこと
    - 製品の別シリーズ、過去のバージョン、オプション...

朝会でもレビューでも会議でも大抵はテーマがあってそれに注目して話し合いますよね。私はそれを聞きながら、話題になっていないことは何かを考えています。ある機能について何ができるのかを話しているときは、その機能ができないことや、その機能が動くときにできないこと（できたら困ること）を考えます。これから作ろうとしてる機能に似た機能には何があったかなとか、同時に動きそうな他の機能には何があるのかなとか。>

朝会やディスカッションの場で

- 同僚の様子や言動を観察している
  - 顔色、声のトーン、疲れてそう、目が泳いでいる
  - 話している内容が分かりづらい
  - いつもは分かりづらいのに、今日はやけに流暢
  - 質問に対しての回答が直球で返ってこない
  - 正しそうなこと、良さそうなことを言っている
  - うまくいきすぎている
  - サラッと流れてしまったもの、話さなかったことは何か
  - 「そういう仕様です」
  - 「ずっと前からそうなっています」
  - 「大したことない変更」
  - 「今回はマージするだけなので」
  - 「全部やりました」
  - 「前倒しでやります」

『同僚を観察している時、同僚もまたお前を観察しているのだ』

同僚の様子や言動からも違和感を感じることができます。会話の中でこのフレーズが出てきたら要注意というものもあります。
その場でもうすこし詳しく聞いたり、朝会のあと、同僚のテスターと「誰それのあのチケットは様子が変だったね。こんなことも試したほうがいいよね」と話したり（カジュアルなテスト設計）メモしておいて、コミットされたらいつもより時間をとって丁寧にテストしたりしています。

製品を見る

- 昨日の製品と比べてどうなのか
- 自分の期待する動き、予想と比べてどうなのか
  - 一瞬だけど応答が遅くなった
  - 一瞬だけど表示がチカッとした
  - 揃いすぎている数値、いつも同じ結果になる
  - ディスクアクセスランプがチカチカしている
  - 遅すぎる、速すぎる、もっさりしている、カクカク動く
  - うるさい（音、見た目、通知）
  - 操作を間違えてしまう
  - 分かりづらい
  - リソースの使い方が説明つかない

開発中の製品を実際に動かして見ているときの話です。＞正常系は身体に染みついてるくらい見てます。うちのチームの人たちはみんなそうじゃないかな。毎日製品を触っているので昨日との違いをわざわざ比較するというよりは、手触りでそれを感じます。製品を触るときはこう動かしたらこうなるだろうと予想したり、こう動いて欲しい、こうならないで欲しいというような期待があります。その思考自体が間違っているときもあるけど、それは恥ずかしいことや悪いことではなく、むしろ良いこと。だって間違っていたことが分かって正しいことが知れるので。ここに列挙したのは製品を動かしているときに製品自身が教えてくれる違和感の兆しの例です。程度によっては違和感ではなく明らかに問題になるものもあります。

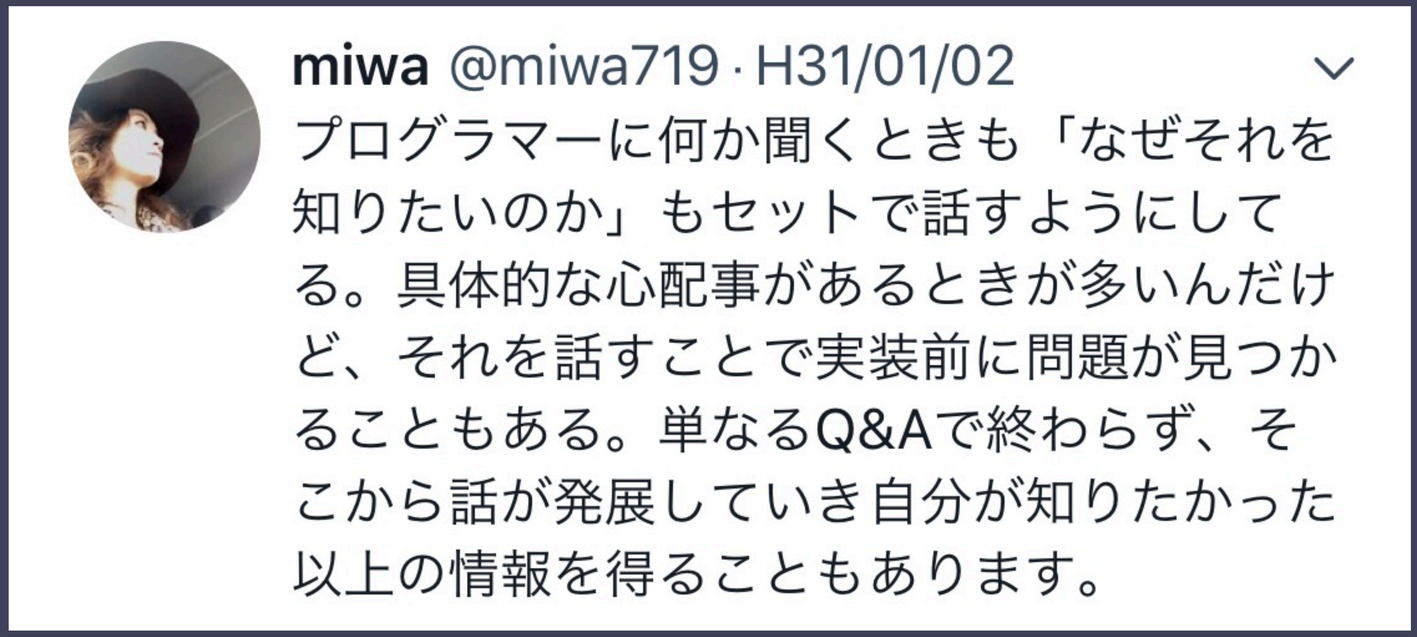
チケットを見る

- 書いてある内容（開発日記、テスト）について
  - 知りたいことが書いてあるか？
    - それで良いとした理由、根拠が読み取れない
    - プログラムの断片やコミットしたファイル名しか書いてない
    - 作業の名前（実装中）しか書いてない
    - Bugチケットなら
      - どこがどうダメだったの
      - どのチケットでバグが入ったのか
      - どこをどう直すのか、直したのか

チケットをどう見ているのかの話です。チケットには日々の具体的な作業内容（開発日記）やテストやテストの履歴などが書いてあります。自分の知りたいことが書いてあるか？といった視点で見ているようです。書いてないとそれが違和感となります。現在開発中のチケットはもちろん、関連する相当古いチケットも読みます。私が知らない開発はチケットを読みながらどのような開発だったのかを頭の中で構築していきます。

知りたいことを教えてもらうとき

- 「なぜそれを知りたいのか」もセットで話す



例えば、ある処理の状態遷移について知りたいとして、その知りたいと思った時には、私は何かを心配していて、それが起きないことを確かめようとしているんですけど、その心配してることそのものや、試そうとしてることも話します。こうすることで、私が何を知りたがっているのかが相手により伝わります。知りたかった以上のことを教えてもらう時もあります。それでうまくいく実装の秘密とか、なぜそうなっているのか（設計の意図）とか。それを聞いてしまったら、今度はそれを踏まえた上でうまくいかなくなりそうなケースを探し始めるんですけども。

チケットを見る

- 書いてある内容（開発日記、テスト）について
  - たくさん書いてあるけど読めない
    - 書き手も分かってないのかもしれない
    - チケットの粒が大きすぎる、仕様から設計に落とすところがうまくない
  - 作るものに対してテストケースが少なすぎる（足りない）／多すぎる
  - なぜこのテストケースが必要なのかが分からない
    - 作りたいものを誤解しているのかもしれない（プログラマー、自分も）

こんなにたくさん読めないよーと感じるのもシグナルですね。もしかしたら書き手も分かってないのかもしれない。そのチケットでやろうとしていることの粒が大きすぎるのかもしれないし、仕様から設計に落とすところがうまくないのかもしれない。いずれにせよみんなの問題です。
私たちのチームではそのチケットで実現しようとしていることに対して「できたことが確かめられるテスト」も書きます。テストはプログラマーが書くんですけど、そのテストについても見ます。このチケットではこんなことを確認するだろうという規模感がある程度頭にあるので、テストを見た瞬間にあれ？少なすぎない？多すぎない？を感じます。そしてテストの内容を一つ一つ動かしながら細部まで見ていくという感じです。

チケットを見る

- 状況について
  - なんども仕切り直される
  - ひとりで複数のチケットに手をつけている
  - なかなか手をつけないチケットがある（後回しにされる）

チケットの状況からもおかしさを感じることができます。私たちのチームでたまに見るのは「なんども仕切り直される」です。何かをやり始めるときはこれでうまくいくはずだとある程度の見通しを立てるので、その予想が外れているということですね。元々の作りが実はもっと複雑なのかもしれないです。同僚のテスターと、まだ見えてないところがあるよね、このパターンは思いもよらぬところでバグが見つかるやつだよね、と危なさの度合いを話したりします。ひとりで複数のチケットに手をつけていたり、なかなか手をつけないチケットがあるのは、製品として組み合わせた確認がいつまでもできないということ。思ったように製品が作れない状況は、個人ではなくみんなの問題。様子見しないでどうにかします。

チケットを見る

- 20年分くらいあるチケットを読むのが日課
  - 今日の気になるキーワードで全文検索
  - Bugチケットは頭にたたき込む勢いで読み込んでいる
    - これから作るものに同じようなバグが入るかもしれない
    - ほぼ日手帳にメモする（テストのネタ帳）
  - 小さな知識を積み重ねる（自分への投資）

XP祭り2019「チケットシステムの設計と実装、『あのチーム』の運用」http://xpjug.com/xp2019-session-a4/

＞読んだあと気になるものはすぐに具体的なテストに展開してTestingすることもあるんですけど、ここで仕入れた情報（小さな知識）を積み重ねて未来のテストに活かそうとしています。自分への投資ですね。活かせている実感があるからやめないんだと思う。

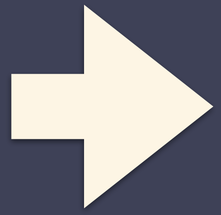
コミットログを見る

- 違和感をつかまえるための手がかりにしようとしていた
  - 変更する内容に対して変更したファイルが多すぎる
  - 同じファイルを１日に何回もコミットしている
  - 気になるものは変更前と変更後のコードの差分も見ていた
- いまはやっていない
  - うちのチームに対しては網目が粗すぎて引っかからない
  - 短時間でコードをある程度読めるくらいでないとダメそう（これが大きい）
- いまでも注目しているのは、連休の前日にコミットしたもの

コミットログも違和感をつかまえるための手がかりの一つにしようとしていました。毎朝１０分くらい時間をとり、前日にどのチケットに関するものがコミットされたかを確認して、気になるものはコードの差分も見たりしていたんですけど、具体的に役立つものが取れなかったのでやめました。うちのチームに対しては網目が粗すぎて引っかからない感じ。短時間でコードをある程度見れるくらいじゃないとダメそう。
いまでも気にしているのは連休の前日にコミットしたものです。連休明けにいつもより丁寧に確認するようにしています。

自分自身を見る

- 感情の変化や身体の状態をシグナルとして使う
  - 退屈、つまらない
  - 眠い
  - 疲れる
  - 焦っている
  - 面倒くさい
  - 気持ちいい



こうなってしまう原因を考える

（製品、仕事のやり方、自分自身 ???）

開発中にわきあがる自分自身の感情の変化や身体の状態もシグナルとして使うことができます。こうなってしまう原因がどこにあるのか考えます。退屈なのは待ち時間が長すぎるといった処理系の問題かもしれないし、自分のテスト（思考）が単調になってしまっているのかもしれません。疲れるのは文字のサイズが小さすぎる、視線の移動量が多すぎるといったUIやUXの問題かもしれないし、プライベートでの心配事があって前日うまく眠れなかったのかもしれませんね。気持ちいいときはやっていることを外している場合があります。

自分自身を見る

- 行動パターンをシグナルとして使う
  - 過去にあげたバグチケットをまたあげてしまう
    - どうあるべきかを再考するときがきた合図
  - なんども聞いてしまう
    - 覚えられないくらい複雑な仕様、手順である（問題の原因を人のせいにしない）
  - 後回しにしたことがある
  - 時間を忘れてテストしている

自分の行動パターンもおかしさをつかむためのシグナルとして使うことができます。過去にあげたバグチケットをまたあげてしまうのは、前回みんなで話し合った結果見送りになったけど、やっぱり自分はおかしいと思っているんですね。あの時は納得して見送りにしたけど今も見送りでいいかどうかはまた別の話です。どうあるべきかを再考するときがきたと捉えることができます。なんども聞いてしまうのは、覚えられないくらい複雑な仕様である、テスト手順がむずかしすぎる可能性がある。覚える気がない、たるんでいるとか自分（人間）のせいにしない。製品やテストや仕事のやり方の課題や問題として考えるきっかけにします。（これらの行動は自然体でいないと出せないかも）

チェックリストについて

今回のようなスライド（やってることや考えてることをずらっと箇条書き）を使うと「チェックリストにして運用しているのでしょうか？」と質問されることがあります。答えはノーで、誰かに伝えるために言語化しただけでチェックリストにはしてません。開発の変化ややりたいことの速度に全然合わないというのもありますし、毎日こういうことを考えながら仕事をしているので、だいたい頭の中に入ってしまっています。車を運転するときに教習所からもらったテキストを読んでいちいち確認しないのと同じ感覚です。

ただ、そういう状態になるまでにはチェックリストを繰り返し見ては手を動かし、頭の中にたたき込む段階があるのかもしれませんね。また記憶したとしても人間の脳は忘れるように設計されているので、たまに見て点検するのは良いことだと思います。

チェックリストは使い方によっては「これだけやっていればよい」と思考停止を促すものにもなります。私たちを取り巻く状況は日々変化していますから、それに合わせてチェックリストも見直してアップデートする必要がありますね。チェックリストを使ってやりたいことはなんでしょう。それは本当にやりたいことでしょうか。本当にそれができてますか。ソフトウェア開発ってチェックリストでチェックできる程度の簡単な仕事ではないよね、という気持ちもあります。想像しているチェックリストが違うのかな？

チェックリスト自体は悪いものではないのでなおさら厄介なんですよ。だから「チェックリスト」という言葉を聞いたときは身構えてしまいます。私にとって注意すべき単語、シグナルの一つです。

医療機器（自社製品）のソフトウェア開発に従事。
プログラマー歴18年。テスター歴10年くらい。
現在はeXtremeなチームに所属するテスター。動かして試すのが好き。Twitter ID : @miwa719

違和感に対する感性を高めるには？

感性を高める？

- 違和感はいつも目の前にあり、それを感じるか感じないかの差だとしたら

なぜ、感じることができたのかな？

より良い製品を作るために違和感を活かすとして、その感性を高めるにはどうしたらよいのでしょうか。私は感性を高めるための特別なトレーニングはしてません。ですので、普段何気なくやっていることにヒントがあるのではないかと考えました。違和感はいつも目の前にあり、それを感じるか感じないかの差であるとしたら、なぜ感じることができたのか？を明らかにしていくことで何かが見えてきそうです。ここ数ヶ月間、違和感を感じたときに自分に問いかけてみました。

なぜ、感じることができたのか？

- 製品を知っているからこそ感じることが圧倒的に多い
  - よい製品、わるい製品のイメージを持っていた
  - 設計の複雑なところ、間違いやすいところを知っていた
  - 過去の不具合を知っていた（現象や再現手順、原因も）
    - 原因を知ることは、設計や実装を知ることにつながる
  - 起きたら嫌なことを知っていて、類推していた

製品を知っているからこそ感じる違和感が圧倒的に多かったです。製品を知れば違和感は自ずとついてくるということだと思います。だからみんな安心してもっともっと製品のことを知ろう！スライドに記載したのは、製品を作る上で知らないとならない最上位のこと（どのようなものを、どんな風に作っていこうとしているのか、それが欲しい理由は何か）以外で「何を意識していたのか、何を知っていたのか」を言語化しました。>

なぜ、感じることができたのか？

- 習慣になっていること
  - 自分ならどう作るのかを想像している（数々の失敗経験）
  - あきらめない（しつこい）
  - 疑い深く、だいたい悪いほうへ倒す（悲観的思考）
  - 他者の違和感を取り入れている
  - 自分の感情に目を向け、なぜそう感じるのかを考えている

習慣になっていることがきっかけで感じることができたものもあります。自分ならどう作るのかを想像できるのはプログラマーの経験から来るものかな。あきらめないのは、同僚のテスターもあきらめなくてすごい。そうやってあきらめなかったところからほつれ（おかしさ）が見つかったりする。そういうのを糧（自分を奮い立たせるもの）にしています。疑い深いのはテスターならではの特性？楽観的に考えたい人に対して悲観的な意見を言わなきゃならないときもあって、本当は言いたくないんだけど仕方なく言ってます。下の二つ＞については、このあと紹介します。

なぜ、感じることができたのか？

- おまけ：ついやってしまうこと
  - 他社のシステム、アプリケーションを動かして、試す
  - 妄想する
    - 世界で起きたバグを検索する、現象から原因を妄想する
    - 時事ネタを使ってこれから起きそうなトラブルを妄想する
    - プログラマーやテスターが書いた文章を読む、妄想する
  - iOS / watchOS App を作って遊ぶ

これはおまけで載せました。仕事以外で普段ついやってしまうことです。＞気がついたらテストみたいなことしている。似たようなことをしてる方もいるのではないでしょうか。職業プログラマーが休日も趣味でコードを書いてしまうのと似てる？筋トレみたいなものなのかも。あと妄想癖があります。＞アプリ作成は自給自足みたいなものです。仕事で扱っている製品がとても大きいので自分専用の実機への憧れがあります。自分で作って自分でテストできるのがすごいんだけど、自分のテストに自分のプログラミングがぜんぜん追いつかないのがすごい。

なぜ、感じることができたのか？

- 習慣になっていること
  - 自分ならどう作るのかを想像している（数々の失敗経験）
  - あきらめない（しつこい）
  - 疑い深く、だいたい悪いほうへ倒す（悲観的思考）
  - 他者の違和感を取り入れている
  - 自分の感情に目を向け、なぜそう感じるのかを考えている

他者の違和感を取り入れる

- 他者の違和感を使い問題を見つけている
  - 偶然とは思えない回数
  - この現象が起きているのは、私だけではない
  - コストパフォーマンスが高い！
  - やり方を伝授したい

miwa @miwa719 · H30/11/14

毎朝、昨日見つけたバグやバグではないけど気になることを（チームを超えた）テスター同士で交換するのだけど、話を聞いたテスターがそれに関連するような（または似たような観点で）バグを見つけてくるのはおもしろい現象だなと思っています。一度や二度ではないから、偶然ではないと思っています。

クロード・モネは印象派の中でも最もひたむきで、時間や季節とともに移りゆく光と色彩の変化を生涯に渡り追求した画家と言われています。印象派の画家たちは光の変化や空気の匂い、重さ、揺らぎといった一瞬の印象を表現しようとしました。なんだか違和感をとらえるのとちょっぴり似ていませんか？　彼がルノアール、シスレー、ピサロといった芸術仲間と互いに刺激し合いながらその感性を磨いていったのと同じように、私も同僚と違和感を交換しています。

https://www.artic.edu/highlights/5/impressionism-highlights

他者の違和感を取り入れる

- やり方
  - 誰かが問題（バグ）を見つけた時に
    - なぜそれをやってみようと思ったのかを聞く
    - なぜその問題（バグ）に気づけたのかを聞く
  - 自分からも積極的に話す
    - みんなに空気感染する
    - 違和感とまではいかないけど気になることも話している

>気づきの前に感じた違和感がなんだったのかを教えてもらうですね。特に聞くのは、なんでそんな操作をしてみたのか、なんでそれに気がつけたのか私には予想がつかないバグです。わかりやすくいうと、なんで私には見つけられないだろうなと思うちょっと悔しいやつ。その違和感を感じるための何かが私には不足しているんですね。

https://www.artic.edu/highlights/5/impressionism-highlights

他者の違和感を取り入れる

- 交換することが目的ではない
  - 自分のものにするまで意識して使う（なりきりテスト）
  - もともと自分には無い感覚なので使わなければ忘れる
- 相乗効果
  - 他者の違和感 × 自分の思考 → 新しい違和感の創出

＞相乗効果の例：Twitterのタイムラインで「食欲が止まらない。風邪をひく予兆かもしれない」というのを見て、世の中にはこういう人もいるんだ！と驚いたことがありました。私には無い感覚です。もし誰かが急にバカ食いし始めたら（この人は風邪をひくのかもしれない）と思うことができるようになります。またこれを知ったときに風邪をひく前に食欲が止まらない人がいるのなら、風邪をひく前に特定の食べ物を食べたくなる人もいるかもしれないと思いました。

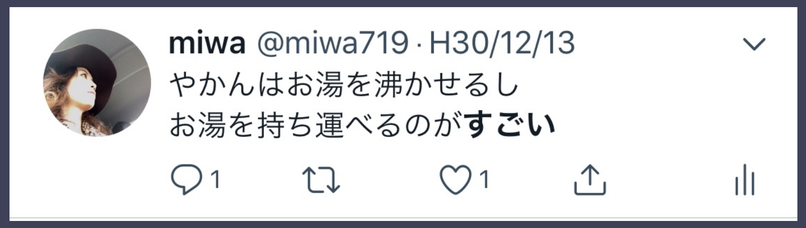
https://www.artic.edu/highlights/5/impressionism-highlights

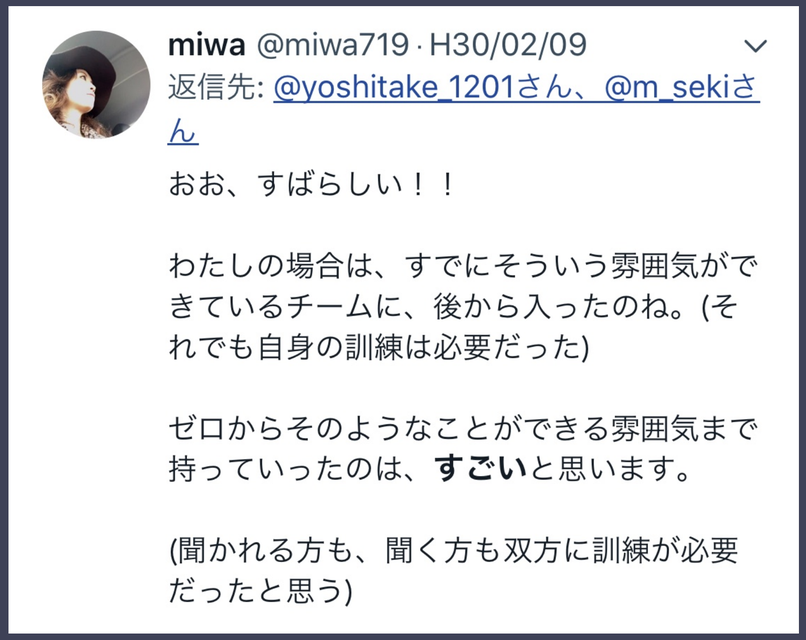
自分の感情に目を向ける

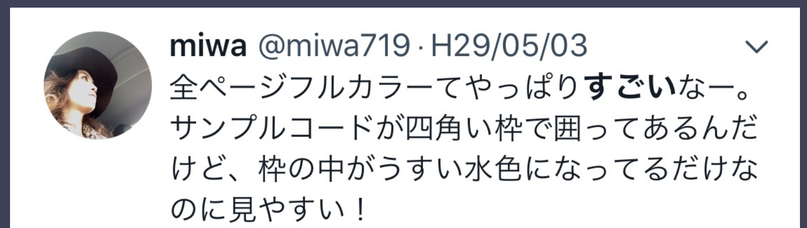
- 感情が教えてくれるものは意外に多い
  - なぜそう感じるのか、何がそうさせているのかを考える
    - いつでもどこでも考える
    - はじめは意識してやるのかもしれない（そのうち癖になる）
    - 反復することで反射神経が鍛えられていく
    - 自分の感情に敏感になり、より小さな変化に気づきやすくなる

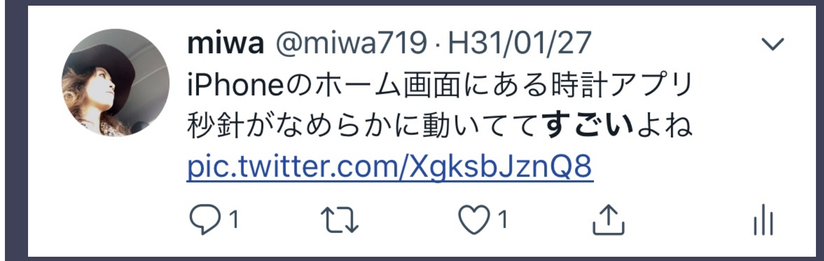
さきほども「自分自身を見る」ところでもお話ししましたが、自分の感情が教えてくれるものは意外に多いです。面白い、楽しい、美味しい、辛い、イライラする、嬉しい、感動した…などなど。感情ではないけれど、何となく気になる、嫌な予感がするなどもそうです。その感情をそのままにしないで、なぜそう思うのか、何がそうさせているのかを考えています。反復することで反射神経が鍛えられていき、その結果、自分の感情に敏感になり、より小さな変化に気づきやすくなっていそう。そういうことが仕事中にも作用しているのではないかと思いました。

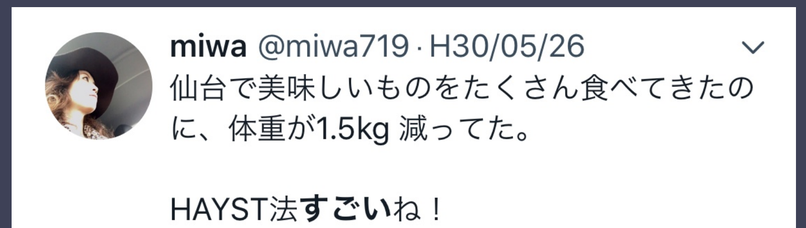
すごいを考えている

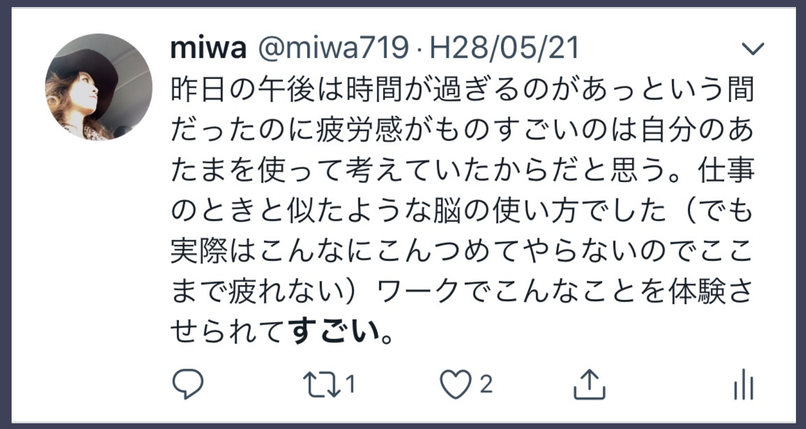












なぜそう感じるのか？を考えている実例。本人は真面目に考えているんだけど、こう並べてみるとちょっと頭がおかしい感じなのもある。

好きを考えている

miwa @miwa719 · H30/09/09

京都銘菓『阿闍梨餅』を買ってきてもらいました。これ**好き**なんだけど、賞味期間が短いのでなかなか食べられない。うれしい。

しっとりもちもち弾力ある餅生地、中には粒あんがぎっしり（わたしは粒あん派です）美味しい😊 pic.twitter.com/Up2jIRNDIB

8

miwa @miwa719 · H28/02/12

わたしには**好き**なバグの種類があるみたい。ものすごく地味なやつ。機能そのものは動くからなかなか見つからないし見つからなくても多分問題ないやつ。でも見つけられたらプログラマーは直したくなるやつ。そのバグのことはずっと憶えてるから**好き**なのかなって。**好き**というか心をくすぐられる感じ。

3 5

miwa @miwa719 · H31/02/22

コーヒー豆はここから買えます☕

定期便の**好き**なところ
・全く知らない豆と出会える
・毎月注文操作しなくても毎月届く
・毎月楽しみにしながら待てる
・1年分を一括購入でややお買い得

定期便は400gもあります

コーヒー定期便(毎月) 200gコース1年一括払い cafenekonoya.shop/items/5bf54a2c... @stores_jp

1

miwa @miwa719 · H30/12/28

バグの隠れ方やプログラミングが優秀でドライバーが堅牢（データが存在しない日時に突入してもボロボロになりながらも動く）なところ、あと８月３２日は夢だしストーリー性もある。こういうバグ**好き**だわ〜

もしリリース前に見つけることができたら『秘密の技』として搭載することを提案したいです。

> **miwa** @miwa719 · H30/12/28
> ぼくのなつやすみの８月３２日をついに解説（作者本人） - Togetter togetter.com/li/46554 @togetter_jpより

なぜそう感じるのか？を考えている実例です。

違和感をつかまえるための
コツ・工夫

○○○●○

違和感をつかまえるためのコツ

- 私がつかまえたい違和感
  - まだ誰もつかまえてない、なかなかつかまえられないもの
    - プログラマーもテスターも違和感をつかまえるのがうまい
    - たいていのおかしさは誰かがつかまえてしまう
- 自分の意識の持っていきかたを矯正している

『うちのチームのプログラマーはなぜテストがうまいのか』http://miwa719.hatenablog.com/entry/daily20190624

私がつかまえたいと思っているのは、まだ誰もつかまえてない、みんなが（自分も）なかなかつかまえられない違和感です。これはうちのチームのプログラマーやテスターが違和感をつかまえるのがうまい状況がそうさせています。大抵のおかしさは誰かがつかまえてしまう。限られた時間の中で問題につながりそうな予兆を1日でも半日でも早くキャッチしたいのです。そのために自分の意識の持っていきかたを矯正しています。

違和感をつかまえるためのコツ

- 誰も（自分も）考えそうもないことは何かを考える
  - 見えないもの、書いてないことに注目する
  - 話題になっていないことは何だろう？
  - 安心していることは何だろう？

誰も考えそうもないことは何かを考えようとしています。チームのみんなが考えそうもないこと、自分が考えそうもないことを自分で考えるのはとても難しいと思います。漠然と考えても思い浮かばないので考え方のパターンを使います。(1)みんなである機能についてどう実装するかを話してる時、声は自動的に入ってくるのでそれを聞きながら見えてないもの、例えばその機能は、過去のバージョンで特別な処理があったけど、あれは今回どうなるんだ？とかね。(2)話題になってないことを考えるのは朝会の話でも出てきましたね。(3)みんなが安心しきっていることから問題が見つかることがあります。安心していることはなんですか？それは正しく安心しているんだろうか？

違和感をつかまえるためのコツ

- 間違っているかもしれないぞ、と思う
  - 決まった仕様、過去にそれでよしとしたもの
  - 自分のことも疑う
- 間違っていないこととそれで良いかどうかは別
  - それが間違っていないとなぜ言えるのか
  - それで良いとなぜ言えるのか

間違っているかもしれないぞ、と思うようにします。決まった仕様でもルールでもなんでもいいのですが、それが本当に欲しかったものなのか？そうした理由（意図）は？１年前はそれでよかったけど今もそれでいいんだろうか。自分のことも疑う。先入観にとらわれていないだろうか。信じたいものを信じていないだろうか。見たいものしか見ていないんじゃないかって。間違っていないこととそれで良いかどうかは別ですね。正しいことが、いつでもどんなときでも最良の解ではないことと同じですね。それが間違っていないとなぜ言えるのか。それで良いとなぜ言えるのか。そこを考えたり、みんなに問いかけます。自分でも分からないことが多いんですよ。

違和感をつかまえるためのコツ

- 想定外の問題を見つけるのではなく想定をひろげる
  - 気にする範囲をひろげる
    - あと一歩、相手の領域に踏み込んでみる
  - これだけやっていればいい、は危うい

想定外の問題を見つける→想定をひろげるようにしたほうがいいみたい。言葉遊びのようだけど、想定外の問題を見つけようとしているときは、まだ自分の枠にこもっている感じがします。（想定外の問題は想定したらその瞬間に想定外ではなくなるので想定外の問題というのは見つけられないとかそういう屁理屈は言わないようにすること：自分への注意）気にする範囲を広げるといいですね。みなさんも自分たちの仕事の範囲はここまでというものがあると思うのですが、あと一歩、相手の領域に踏み込んでみる。これだけやっていればいいと思った時は「危ないぞ」と思った方がいいです。

違和感をつかまえるための工夫

- 疲れないようにする
  - 疲れているときは違和感をつかまえられない
  - 仕事中は１時間に１回は休憩を取る（Apple Watchのスタンド機能）
  - 注意力の持続時間は10分が限度という報告もある
  - 基本的に土日はしっかり休んでます

＞これは個人差があるけど、違和感がどうのとかではなく疲れているときは休みましょう。

違和感をつかまえるための工夫

- 自分自身のバグもシグナルとして使う
  - **Typical bug of miwaOS**
    - 過集中（熱暴走）
    - 寝坊する（sleepからの回復に失敗）
    - 末っ子問題（執着して時間を溶かす、現実逃避）
    - 方向音痴（空間認識能力の欠如）

違和感の兆しはいろんな原因でつかまえられないんですけど、自分自身のバグも原因になることに気づいてから意識するようになりました。代表的なバグの過集中バグ：テストに熱中し過ぎる不具合がある（これはもう自分としては最高に気持ちいい状態なのでやめづらい）過集中ということは裏を返すとそれ以外のことについては散漫になるということなんですね。何かを見逃している可能性があります。暴走してる姿を見て「もうやめなよー」と言ってくれる人もいて渋々やめる。寝坊は身体が疲れているというシグナル。無理して会社に行ってもいつものパフォーマンスが出ないようなら午前中は休みます。年に３回くらい予告もなく午前中休むのは大抵それです。休んでバグを修正してる。

違和感をつかまえるための工夫

- 差分が微量で違和感をつかまえづらいことがある

- つかまえたいものに合わせて差分が大きくなるように細工する（周波数を合わせにいく）
- 問題が起きたとき（見つけられなかったとき）は考えるチャンス

はじめに「差分が違和感」ということを話しましたが、差分が微量で違和感をつかまえづらい場合があります。そこで、わざと差分が大きくなるように細工してつかまえやすくすることがあります。問題が起きたとき（チームや自分が問題を見つけられなかったとき）はチャンスです。この次はどうやったら見つけられるのかを考えます。自分の知識が足りてなかった場合もありますし、Testingのやり方をすこし変えると（工夫すると）次はおかしさに気づけるぞ、となるものも出てくるわけですね。

つかまえたいものに周波数を合わせる

- 遅いPCで動かす（低スペックのもの）
- いつもと違う順番で起動する
- 起動直後は特別な状態であるを意識
- 何日間も電源を落とさないで動かす
- PCのシステム日時を23時55分に変更する

- 動かせるものは全部動かす
- リソース状況を見ながら動かす
- モニタの電源をOFF→ONする
- ダイアログ、他アプリの画面をかぶせる
- 接続しているものを外す

- ダイアログボックス（書ききれない省略）
- テキストボックスで（書ききれない省略）
- ラジオボタンを２つ以上選ぼうとする
- ドロップダウンリストは下から選ぶ
- マウスの使い方（書ききれない省略）

- たくさんデータを登録した状態で動かす
- ひとつもデータがない状態で動かす
- 古い装置で生成したデータを使う
- 新しい装置で生成したデータを使う
- いつも使っているデータを使う
- いつも使っていないデータを使う
- 優秀データを使う

- ゆっくり操作する
- 素早く操作する
- 何度何度も繰り返す
- 両手を使う
- 手と足を同時に使う
- ショートカットキーを押す
- 画面の解像度を変える

- 放置する
- 何もないところをクリックしている
- おとなしく待っていない
- 制限を突破しようとする
- 朝一（起動直後）の操作は全神経を集中
- 初回と処理速度を比べる
- ファーストインプレッションだいじ

- 壊れたデータを使う
- 壊れたメディアを使う
- ファイルがいつも読めるとは限らない
- リソースは途中で枯渇するかもしれない

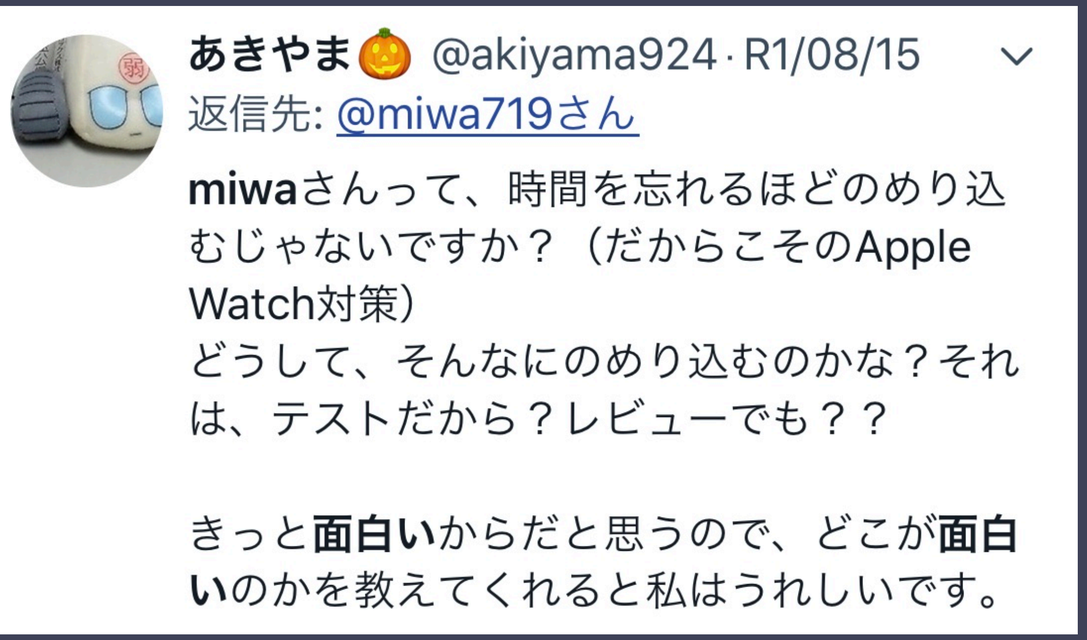
- 雷が鳴っている時は瞬電を期待しながら、瞬電が嫌そうな機能を触る
- エラーが起きた時さらに別のエラーを起こそうとする

- ログファイルを見る
- ログファイルを全部削除してから動かす
- イベントログを見る
- 設定ファイルを途中で消す、変更する

Testingの時にやっているつかまえたいものに周波数を合わせる例。繰り返しやっていると習慣になりより違和感をつかまえやすくなります。

JaSST Review '19 書き下ろしエッセイ ②

製品開発のどこが面白いのか



時間を忘れるほどのめり込むのは miwaOS のバグですが、それを引き起こしてしまうのは「面白いからではないか」ということですね。はい。製品開発は面白いです。今ではその面白さにすっかり魅了されてる感もあります。でも開発現場で四六時中「面白さ」を感じているかというとそうでもないんですね。私のツイートやブログを読んでいつも楽しそう、面白そうに見えるとしたら、そういう風に見えるところを特別に切り出して書いているからです。よい機会なのでどこが面白いのかを書く前に「面白さ」とはすこしちがう感情のことも書き記しておこうと思います。

私の製品開発に対する思いは「子育て」に似ています。面白いから子育てする、にはならないのと同じように製品開発も責任のようなものが伴って重いものです。そばにいてもいなくても親が子のことを思って心配するように、いつも製品のことが気になります。子は親の思ったようには育たないとか育てたように子は育つとか諸説ありますが、製品は自分たちが作ったように動くんですよ。（続きはあとでかく）

医療機器（自社製品）のソフトウェア開発に従事。
プログラマー歴18年。テスター歴10年くらい。
現在はeXtremeなチームに所属するテスター。動かして試すのが好き。Twitter ID : @miwa719

違和感をつかまえた後のこと

違和感をつかまえたのに、そのままにしてしまったことありませんか？

- そういうものなのかもしれない
- 気にはなるけど問題（バグ）とは言いきれない
- ずっと前からそうなっているし、、、
- 気のせいかも（もうすこし様子を見よう）様子見：様子を見ないときに使う言葉

＞言いづらいときもありますね。いちいちうるさい人だと思われるのやだな。開発の終盤でこんなことを言ったら（いまそれ言う？）と思われるかな。前にも言ったんだよなとか。なんとなく表に出せない状況や心境はよくわかります。あとから問題が発覚して（ああ、あのときの違和感がそうだったんだ、みんなに話しておけばよかった）って思うんですよね。もったいないですね。違和感に申し訳ない。だから言いづらい違和感ほど誰かに伝えるようにしています。そうそう「言いづらい」も自分で分かるシグナルですね。

miwa @miwa719 · R1/07/18

テストしていて（気のせいかな？）と思うことのほとんどは気のせいじゃない

1　6　30

経験則ですがテストしていて（気のせいかな？）と思うことのほとんどは気のせいではありません。気のせいにしたい時に気のせいかな？って思うんだと思う。人類のバグではないでしょうか。例）保存ボタンを押したつもりだったのに保存処理が始まらない。あれ？と思ってもう一度ボタンを押したら今度はちゃんと保存処理が始まった。さっきは押したつもりで押してなかったのかも。押したのは気のせいかな？念のためもう一度同じ操作をしてみたらちゃんと動いた。こんなとき、あなたならどうしますか？（システム起動後、初回の保存時しか出ないバグだった）

言いづらいときにどうしてるか

- これは仕事だから仕方ないと思って言っている
  - この気持ちの持ちようはいろんなところで使えるので便利
  - 練習すると言えるようになる

『ロールプレイングゲーム』https://druby.hatenablog.com/entry/20101202/p1

＞練習すると言いづらさがなくなるのか？といえばそれはなくならないです。言えるようになるだけ。毎回言いづらいけど仕事だから仕方ないと思って言っています。プロの無職 @m_seki の有名な台詞「だって、仕事でしょ？」を実践している。

言いづらいことを言っている結果

- 私が休み（今日はいない）が交換すべき情報になっているらしい

わたしくらいになると「いるか いないか」がみんなのシグナルになってしまう……

違和感をつかまえたら

- つかまえた瞬間に声を出す
  - 違和感を無かったものにしづらくなるのが良い
    - 「ん？」
    - 「あれ？」
    - 「いまのなんか変だったよね？」
  - 私の異変に気づいた同僚が「なになに？」と興味を持ってくれる
  - 練度が高くなると首をかしげるだけで、周りがざわつくようになる

違和感をつかまえたあとどうしたらよいかの話。違和感を自分ひとりのものにしておかないのがポイント。私が何をしているかをお伝えします。
３つあります。１つ目です。＞

違和感をつかまえたら

- **観察する**
  - **違和感の正体を明らかにする**
    - どこに違和感を感じたのか、なぜおかしいと思ったのか
    - 観察した結果をほぼ日手帳にメモするときもある
  - **つかまえた瞬間に（いまのあれはなんだ？）を考える**
    - 小さな違和感ほど時間の経過に伴ってその感覚が薄れてしまう

２つ目です。観察する。＞

違和感をつかまえたら

- 同僚に話す
  - 同僚はどう思うのか、同じようにおかしいと感じるのか
  - 違和感の正体が分からなくても話していい
  - もし、個人で感じる違和感を受け取ってもらえる風土がない場合
    - 違和感を交換しあえる同僚を一人作ってみる
    - 二人で感じる違和感なら話しやすい

チームや組織は概念で実体は自分です。風土は自分たちで作ろう！

最後の３つ目は、同僚に話す。観察した結果をチケットに書いたり、朝会で話題にすることもあるのですが、それよりも先に同僚のプログラマーやテスターに話すことが多いです。＞

これで私の話はおしまいです

これからも違和感を大切にしてより良い製品を丁寧に作っていきたいと思います

違和感のつかまえかた

組み込みシステムの開発者（テスター）としてやっていること

と思ったのだが、みなさんに聞いてみたいことがあります。

今日の話、分かりやすかったですか？

分かったような気になっているとき

- 違和感に対する感度が鈍くなっています
  - 自分自身で感じとれる便利なシグナルでもある
  - シグナルを受信したらこう思う
    - 私のくせにこんなに分かるのはおかしい
    - こんなに簡単ならみんな悩まないはずだ
    - これは罠！

（こんな質問をしておいて大変言いづらいけど言います）今日の話を聞いて分かったような気になっているとしたらちょっとまずいのです。
分かったような気になっているときは違和感に対する感度が鈍くなっています。これは今日の話だけではなく普段の開発現場でも同じです。ちょっと話を聞いたくらいで、レビューに１回出たくらいで分かるはずがないと思っています。みなさんはそんな簡単に分かるような仕事なんてしてないはずです。この「分かったような気になっている」は自分自身で感じとれるシグナルです。これを受信したらこんなことを思います。＞

ほんとうに

これで私の話はおしまいです

これからも違和感を大切にしてより良い製品を丁寧に作っていきたいと思います

違和感のつかまえかた

組み込みシステムの開発者（テスター）としてやっていること

２分以上時間が余っていれば 書き下ろしエッセイを音読する。読むのに必要な時間 ①２分、②２分