アルタスファイブ通信 2021 年冬号

「ちょっとだけ変なことをしている会社＝技術変態アルタスファイブ！」です。

今年も、皆様方からのお仕事のご依頼を賜りまして、堅実な一年を送ることができました。いつも、ご愛顧いただきまして、ありがとうございます。

世の中は、DX 化の推進などが後押しとなって、まだしばらくは、システム開発の需要は、根強い市況感だと言われています。確かに、弊社でも、手が空くといった状況になることは、あまり無かったように思います。

この状況が続くのなら、飯が食えている今だからこそ、やるべきことがあるんじゃないか？そのあたりを考えながら、弊社らしい、“おもしろ”取り組みをやっていきたいと思っています。その話は、また、別の機会にお伝えします。

さて、2021 年の下半期を振り返って、アルタスファイブの仕事をご紹介させてください。

目次

01 コロナ共生社会と社内コミュニケーションについて

開発部：板子

生観戦のチャンスを生かせなかったのは残念ですがスポーツ観戦好きとしては東京オリンピックには、大変楽しませてもらいました。

そして、来年 2022 年には冬季オリンピック、世界陸上、サッカーW 杯とぎゅっと詰まって楽しみです。

コロナ禍でのコミュニケーションの問題点

今回の私のアルタス通信は「コロナ共生社会と社内コミュニケーションについて」とのお題を頂きましたので、コロナ禍でのテレワークに伴うコミュニケーションついて考えていきたいと思います。

まずは、コロナ禍でテレワークが一般的になる中でコミュニケーション不足による問題点はこんな感じでしょうか？

帰属意識の低下に伴い、生産性が悪化する。

（実際にはもう少し範囲が広いようですが）エンゲージメントとも言います。

会社と従業員の繋がりといったところでしょうか？

組織での役割が不明確になり、従業員が会社に抱く愛着が薄れます。

帰属意識の低下は離職率の増加、生産性の悪化は営業利益の減少など会社にとっての危険が増えます。

極端ですが所属している組織が嫌いになれば、仕事も嫌になってしまいますよね

業務が円滑に進めにくくなり、ストレスが増加してしまう。

チーム内での問題点として

メンバー目線では、直接会話する機会が減り、リーダーからの指示の意図がわからない、理解しづらい。

リーダー目線では、メンバーの働きがわからないので、マネジメントしづらい。

テレワークの良い面でもある、対人関係等のストレスは少なくなりますが、仕事上のストレスは増えてしまいます。

イノベーションが起こりにくくなる。

雑談や気軽なコミュニケーションが少なくなることで新しアイディアが生まれにくくなります。

お恥ずかしながら、雑談も時には必要だよなー程度にしか認識していませんでしたので受け売りの言葉となってしまいますが

「社内の技術情報のコミュニケーションのあり方が、新たなイノベーションプロセスに大きな影響を与える。」

と言ったところでしょうか？

エンゲージメントの低下とストレスの増加には離職の危険性、営業利益の低下など困ったことが多いですね。

イノベーションについては、「これで世界が変わるぞっ！」なんて経験はありませんが、仕事上の不明点を誰かと話してみるだけでも整理が付く経験は多々あります。

悪者扱いの感がある雑談も結構大事なことのようです。

コミュニケーションの問題点をどのように解決していくか

問題点に関して正解はないかもしれませんが、どのように解決して行くべきなのでしょうか

ありきたりかもしれませんが、社員同士の交流を行いつつ、情報を共有（＝話しやすい）しやすい環境を整えることが必要だと思います。

はじめに謝っておきますと、イノベーションはそれ自体を起こすことが難しい話です。ここではコミュニケーションの活性化にとどめさせてもらいます。

帰属意識の低下について

企業目標や経営陣が思い描いているビジョンを伝え、個人としての組織での役割も伝えていく必要があると思います。

当たりまえ感もありますが、オンラインでの社内イベント等もよさそうです。

業務が円滑に進めにくくなる

チーム内でのオンラインでの朝会や夕会、定期的な情報共有や振り返りなども大事です。この中で意識して多少の雑談をすることもよさそうです。

チーム内での出勤日を設けて、たまには顔を合わせてるのも良いと思います。帰属意識の問題についても効果ありそうな気がします。

解決とは無関係ですが、テレワークだからと言って家にこもりっぱなしでは体に悪いです。たまの通勤で多少は運動しても良いのでは？

当社での試み

一般的な解決案を書きましたが当社ではどうかと言いますと…

帰属意識の低下について

コロナ以前より全員参加の社内定例を月１回開催しているのこともあり

企業目標や経営陣が思い描いているビジョンを伝える機会は以前よりありました。

弊社社長も意識して発信してくださっております。

オンラインでの社内イベント等は出来ていないです。コロナ禍当初にオンライン親睦会も試みたのですが、いろいろと面倒なこともあって頓挫しました。反省点です。何か新しいアイディアを考えたいと思います。

また、コミュニケーション全般となりますが、私個人としてはもっと出来ることがあったのではと思います。オンラインかつ対面で「最近、どう、調子よい？」なんて機会を増やせて行けたらと思います。

業務が円滑に進めにくくなる

仕事（プロジェクトチーム等）視点での試みは行えているようです。

カレンダーには関連しそうな予定が沢山入っていますし、プロジェクト別のチャットルームでも「ミーティン始めまーす。」等の会話が朝夕飛び交っています。

1 日中ずっとオンラインで繋ぎ続ける。なんて個人的には拍手を送りたい試みを行っているチームもあります。環境的にはテレワークながらもオフィスで作業しているような感じでしょうか？

ただ、マイクのオン・オフには注意が必要ですね、悪口なんて聞かれたら人間関係崩壊です。

オフィスへの出勤日等については、いくつかのチームで実施してもらっています。

入社１，２年目の若手とベテランの組み合わせで会社としての出勤日を…と考えていましたが、終息に向かっているようにも見えますが不透明な部分も多いこともあり保留です。

とこちらは概ねできている感じがしますが、それぞれのリーダーやメンバーが自発的にやってもらっている結果かと思うので、多少なりとも社内でのルール作りを行っていきたいですね

当社での試みを中心に書いてきましたが、手前みそながらそれなりにできているように思えます。

そもそもですが、月１回の社内定例は、社外で作業しているメンバーの帰属意識の改善が目的でした。

情報共有としてのチーム内の朝会・夕会も基本全員が出勤している時から行っていました。

忘年会や月１での懇親会も以前より行っています。

コロナ禍ではそれに加えて、テレワークでの難しさ加わりました。

今後考えていかなければ行けない課題として、働き方改革が

あります。

継続して、テレワークは必須となる時代になってくるのだと思います。コロナ共生社会とは関係なくこの問題点には取り組んで行かない行けないようです。

今後も、社内コミュニケーションの活性化に取り組み、エンゲージメントを高めることで

より良いサービスを提供出来るよう頑張っていこう。と思う次第です。

あとがき：

リモートで仕事をすることにはメリデメの両面がありますが、生産性についても、気になる点かと思います。実感としては、“生産性は落ちていない“という感覚ではあるものの、具体的なところはわかっていません。

リモートワークを 1 つのキッカケとして、スクラムのベロシティの計測と似たようなことが、スクラム以外のプロジェクトでも、広がっていくとよいのにと思います。何かよいツールとか手法があったら、教えてください。（開発部 佐藤）

02 アプリケーション保守におけるスクラム式のタスク管理について

開発部 檜山

最近発売されたポケモンのダイパリメイクは、初めて開発を別会社に委託したという点が話題になっていましたが、これはグラフィック向上などにより開発コストが高くなってきていることを受けて(全ポケモンを登場させることが難しくなったなど)、今後の開発のためにリソースの育成や確保を行うといった戦略もあるのかなー、だから、あれこれ新要素を追加せず、原作忠実再現なのかなーと勝手に想像しつつ、ガチパで小学生泣かせと言われている四天王＆チャンピオンに挑もうとしている開発部の檜山です。

今回は、私が担当させて頂いているアプリケーション保守で「スクラム式のタスク管理」という手法を用いているので、そのお話をさせて頂きたいと思います。

1. タスク管理の課題

アプリケーション保守と言っても様々な契約形態があり、何か問題が発生した時だけ対応するという保守もあると思いますが、現在私が担当している保守は、1 人月の工数を頂き、平時は、機能追加や、不具合修正などの実装も行っています。

当初は、依頼されたタスクの中から優先度の高いものを順次対応していき、追加依頼があれば、どちらを優先するか都度、相談するといった管理をしていました。

こういった五月雨式の方式でも、追加依頼が頻繁に発生せず、数日程度で捌けるタスクばかりであれば、特に問題なかったのですが(実際、最初の数か月は、この方法でも問題無く運用できていました)

追加依頼の頻度が多くなり、各担当者様毎に優先したいタスクが異なってくると、依頼がある度に、作業を中断して、内容の確認、優先度の相談までする必要が出てきて、タスク管理だけで相当な時間を使ってしまい、着手中のタスクもやりきることが難しくなってしまいました。

2. スクラム式のタスク管理

こういった課題に対処するため、2 週間のイテレーションでサイクルを回していくスクラム式の管理方法が適しているのではないかと佐藤の方からから提案を頂きました。

単に優先度の高いものから順次対応していくのではなく、期間を 2 週間単位で区切り、その中で対応するタスク、対応しないタスクをあらかじめ決め、2 週間単位での計画/作業/振り返りを繰り返し回していくという感じです。

まずは、イテレーションの初日に、対応するタスクをお客様主体で決めて頂きます。

(もちろん、期限があるものや、開発者から見た時に優先した方が良いタスクがあれば提案します)

そこで決めたタスク以外は基本的に対応しないようにします。追加の依頼があった場合は、プロダクトバックログ(タスク候補表)に追加だけ行い、対応するかどうかは次のイテレーションで検討します。

障害対応や急ぎ対応して欲しい依頼も発生するので、その分は、あらかじめ少し時間を確保しておき、本当に必要なものだけを、その枠で対応するようにします。何も無かったり、余ったりした時間があれば、リファクタリングや、作業効率化などの時間にあてます。

イテレーションの最後にはレビューや振り返りも行い、開発スピード向上や、品質向上、お客様とのコミュニケーション向上にも繋げていきます。(振り返った内容はお客様にも共有し、フィードバックを頂き、改善していく)

こうすることで、追加の依頼があってもタスク選定に悩む必要が無くなり、管理が楽になり実作業に集中することができます。(実際、「やること」以上に「やらないこと」を決めたことで、あれこれ悩むことなく集中して作業にあたることができています)

障害など急ぎの要件が発生した場合は、中断して対応にあたる必要はありますが、その分もあらかじめ枠を確保しているので、元々予定していたタスクへの影響を最小限にとどめ、やりきることができます。

3. お客様にとってのメリット

また、この方法は、開発者だけでなく、お客様にもメリットがあると考えています。

一例として、「本当に必要なタスクを吟味することができ

る」という点が挙げられると思います。

優先度と見積もりが記載されている表だけでは、実際にいつ対応されるのかがぼやっとしてしまい、また追加依頼が多くなると目途もつけづらくなってしまい、なんとなくこの辺りには対応されるかなという感じになっていましたが

直近の 2 週間で対応するタスクをお客様ご自身で決めて頂くことで、2 週間、さらには、1 か月後、対応できるタスクと、対応できないタスクが見える化され、改めて

・本当に優先して欲しいタスクはどれか？

・価値の高いタスクはどれか？

を見直して頂く機会にもできていると思います。

実際に、最初のイテレーションでやることをお客様に選んでいただいた際に

こっちのタスクはどうしても今月中に完了させたいので、(元々は優先度の高かった)こっちのタスクは優先度を下げて、その次のイテレーションで対応を検討したいといったように、優先順位を見直して頂くこともできました。

もちろん、その分、お客様側にも、従来の方法よりも社内で検討して頂いたり、調整して頂いたりとご協力頂くことが増えてはしまいますが、それが結果として、お客様の利益のために、工数を最大限に活用することにもつながるのではないかと考えています。特にやりたいことがたくさんあるお客様の場合は効果が大きいのではないかとも思います。(加えて、こうした取組を続けていき、相互にコミュニケーションを取り合うことで、開発者側もお客様と同じ視点で、考えられるようになると、同じ方向を向いて、協調関係が強くなり、より質の高い、満足度の高い仕事にしていけるのではないかとも考えています。こちらは今後の課題にしたいと思います。)

4. 最後に

まだ取り組み自体は始まって間もないのですが、タスク管理が非常に楽になり、作業効率が向上したと実感しております。また、継続的に改善していくことを前提にした管理方法なので、常に課題を見つけ、改善していきたいという意欲にも繋がっています。

タスクの吟味や、作業の見える化などお客様にもメリットが大きいと、私自身は考えておりますが、実際にどう感じておられるかフィードバックを頂きつつ、双方にとって最適な運用方法を探っていきたいと思います。

どういう効果があったか？失敗したことなどもあれば

またどこかでお話させて頂ければと思います。

ありがとうございました。

あとがき：

今回、スクラムのやり方を取り入れることを提案して、実際に、うまくいったわけですが、その理由は、”お客様のメリット”を考えたことではないかと思います。お客様にとって違和感のあるタスク管理にならないように工夫して、事前説明もしていたので、お客様の業務の一部として、取り込んでいただけました。お客様目線で考えることは大事だと、あらためて思います。（開発部 佐藤）

03 システム開発におけるメソッドや変数などの名称付けについて

開発部 柳

コロナが落ち着いてきて趣味であるカードゲームもそれに合わせたように大会が活発になってきて、この機会に大きな大会に試しに出てみましたが予選落ちの惨敗となってしまいました。カードゲームは新しい商品が出るたびに何が強いか研究して日々進化しますが、これを怠けるとすぐおいて行かれてしまいます。同じようにシステム開発でも新しい技術は生まれ続けるため、これも怠けるとおいて行かれてしまいます。趣味も仕事も日々の努力が結果につながる、そんなことを思いながら今日も過ごしています。はじめまして、入社 2 年目開発部の柳です。

今回は、私が 2 年間の開発で悩まされ、考えさせられたメソッドや変数などの名称付けについて、お話させていただきたいと思います。

システム開発において何らかのオブジェクトに名称付けをするという行為は一見簡単そうに見え、実際につける際は名称を考えて文字の入力をする以上のことはしないため作業時間もかからないと入社したての私は思っていました。しかし、これは大きな間違いで名称を付けるという行為は当たり前かもしれませんがそれだけでオブジェクトの意味、役割を表したものであり一つ間違えると、メンテナンス性がない分かりづらいプログラムになってしまうことから、最も気を付けるべき作業の一つであると言えます。

私が名称付けに対して初めて指摘を受けたのは、とあるシステムのリファクタリング作業中のことでした。リファクタリング作業の内容としては重複している処理を一つにまとめた新しいメソッドを作成するというもので、技術的にも難しい作業ではなかったのですが、修正してレビューのためプログラムを提出したところ、様々な箇所に指摘をもらい当時の私は少し面食らってしまいました。レビューの内容は処理に関するものもありましたが、その中で特に目立ったのがメソッドと変数の名称でした。

まず初めに、目に入ったのは変数に略称を使用しないというものでした。略称は作成した人はその役割を理解することが簡単ですが、他の人が作業する場合にそれが何を意味するのか理解するためにプログラムを全て理解しないといけないため略すのではなく、少し長くてもそのままの名称を使用するようにと指摘を受けました。

次に印象的な指摘として安易な名称を付けないことでした。

その指摘を受けた際に注意されたのは「(対象オブジェクト)_update」のような名称のメソッドで、このメソッドの問題としては update という名称に対する意味が広く初めて見た人が機能を勘違いする可能性があること、update は動詞であることからメソッドの先頭につけるべきであるということでした。

これらの指摘は説明されれば納得できるような内容であり、当時の私も理解することは難しくなくその後の作業では名称についてあまり指摘を受けなくなりましたが、これだけできれば名称付けは問題ないということではなく実際にその重要性を知るのは 2 年目の開発でした。

2 年目のある開発にて既存のシステムに機能追加などの改修では、命名規約という名称付けをする際に「UpdateDay」のような各単語の先頭を大文字にするキャメルケースや、「update_day」のように各単語の間にアンダーバーを入れるスネークケースなどが仕様書に指定してありました。基本は命名規約を元に名称を付けることでわかりやすいコードとなるのですが、この開発では命名規約外の要因で少しわかりにくいコードとなっていました。日本語のローマ字名や略称名など入社したての際に指摘された箇所があり、それが何故わかりにくいかを言葉ではなく実感として経験することになりました。

また、この開発では名称付けの細かなミスだけではなく、変数名を作成する際にプログラムは基本的に名称を英単語で作成するため変数名も英単語にしていますが、翻訳するととても長い名称になってしまい仕方なく単語を削りましたが短すぎず長すぎない理解できる名称を付けるのは慣れていないととても難しいと感じました。

名称を付ける目的はチーム間で理解しつつ、他者に渡っても伝わるということにあり、そのために名称付けには命名規約など他人から見てもわかりやすくするための工夫は様々なものがあります。しかし、実際に開発する際は少し気を抜いてしまうだけでよくわからない名称を付けてしまうことはよくあると思います。それを解消するためには名称付けを行い他者からのレビューを受けるなどの経験が必要であり、センスを磨く以外に一人の考えで良い名称を付けることはできないと私は思いました。しかし、一人ではわかりやすいかは測りかねる部分が多く一部のベテランにしかできないとも思っているためチームで開発する際は難しいと思います。

そこで、良い名称とは誰が見ても同じ結論に至ること、つまり共通の認識さえあれば理解できるということであるため、開発を行う際に調べればあるような既存の命名規約を使うだけではなく、独自の単語に対する辞書のようなドキュメントを作成できないかと思いました。開発単位で独自の命名規約ドキュメントを作成していてはもちろん作業効率に影響しますが小さな積み重ねでドキュメントを作成すれば誰でもベテランのようなわかりやすい名称を付けることができるだけではなく、意味がわかれば処理に対するコメントを減らすことができるため開発効率が向上することも見込めたりと思ってしまいます。

まだ私は 2 年目のため知らないことやわからないことが多くあります。しかし、毎日少しずつ蓄積することでいつか形になると信じて今は進み続けていきたいと思います。

あとがき：

考えてみれば、開発の仕事は、始まりから終わりまで、ずーっと何かに名前を付けています。用語集、機能名、テーブル名、カラム名、ソースコードのファイル名、クラス名、変数名・・・。シックリくる名前を付けられるということは、ある意味”理解”できている状態なわけですから、理解度の 1 つの判断にもなると思います。とはいえ、典型的な命名規則があるのも、実際のところなので、開発部全体で共有する命名規則集を作ってもよいかもと思いました。（開発部 佐藤）

04 途中参加のプロジェクトとはじめての Angular

開発部 藤原

オリンピック後の緊急事態宣言がなくなってしばらくたち、解除直後は「解除されたからまた感染者増えるだろう」と思っていましたが、以外にもコロナが落ち着いてきて現在は感染確認数が 50 人を下回って連続 26 日目です。ようやくコロナ生活が終わる気配が感じられてきました。

現在私が関わってる案件は Angular を使っています。Angular は Google が開発している JavaScript のフレームワークで、双方向データバインディングやコンポーネント思想など Django 中心だった私にとってはどれもこれもなかなか経験したことが無かったのでかなり学ばせてもらっています。

中でも非同期処理に関しては Django でも Redis,Celery,ajax などを使って少しは実装したことがあったのですが、今の案件ではコードのほとんどが非同期という仕様になっています。私は案件には途中参加だったのですが、ほとんどが非同期なのを説明された時には正直「ウソでしょ」って思いました

その案件のコーディングに最初は非常に戸惑いましたが、Anguler 自体の公式チュートリアルをする時間を取ってくれたり、そもそも案件内で専用のチュートリアルを用意してくれていたりしていました。その充実したチュートリアル環境もあり、今ではすっかり慣れてコーディングすることができています。

慣れた今だから言えますが、非同期も慣れてしまえばそこまで難しいものではなかったです。たしかにタイミング的な問題で関数に渡すハズの変数が undefined になっていたりすることもありますが、待ち合わせ処理やオブザーバーパターン、他にも実行順などを意識することで可能性が広がります。

それさえできれば細かい処理をたくさん作るような感じでコ

ーディングできてだんだん楽しくなってきます、逆に Django のほうが思い出せなくなってくるくらいです。

それと、双方向データバインディングとコンポーネント指向も始めての経験でした。

Anguler ではページごとにファイルを用意するのではなく一つのページを複数のコンポーネントから作成する想定になっていて、それならば当然のごとくコンポーネント同士でデータをやり取りする機会も多くなっていきます。

そのため非常に効率的で分かりやすく、少ないコード量でバインディングできるようになっていました。最初にコードを見た時はあまりにも何気なく書いてあってどこでデータのやり取りをしているのか分からなくて探してしまったほどです。

余談ですが、ただでさえ学習コストが高いと言われてる Anguler で、かつプロジェクトの途中参加というのもあり普通は最初にとんでもなく苦労するはずです。

しかし前述したプロジェクトごとのチュートリアルという存在はかなり大きく、プロジェクトごとの作法や空気をうまくまとめてくれてたので充実したドキュメントも合わせて高い学習コストを軽減できました。

ソフトウェア開発の分野ではドキュメントの共有不足や、他にもメンテナンスしにくいコードを放置したりすることを「技術的負債」と言うのですが、今回のプロジェクトはここが完璧以上になっていて、技術的貯金とでも言うべきでしょうか？これが参加するときにすごく助かりました。

あとがき：

このプロジェクトは、すでに、何年か経過しています。息の長い案件だからこそ、人の入れ替えを想定した、開発ガイドの充実化とチュートリアル作成を行っていました。それら準備していたことが、うまく機能したことが確認できてよかったです。短期のプロジェクトでは、なかなか準備に時間をかけられないので、そういうときのために、“開発ガイドの素“がほしいですね。（開発部 佐藤）

05 OJT として案件に関わってきての所感

開発部 瓦谷

新型コロナの影響で、入社して早々にリモート環境を整えて実家から外部研修や OJT を受けながら仕事をしておりました。初めまして、開発部一年目の瓦谷です。

初めてアルタスファイブ通信の執筆を任され、内容をどうしようか悩みましたが、今回は OJT として初めて案件に関わった時に大変だったことを書いていこうと思います。

大変だったことの一つ目が、これから開発するシステムの仕様を理解することでした。OJT が始まる前に開発するシステムの概要説明を事前に受けてはいたものの、いざ要件定義書を見始めると自分が知らない用語がたくさん出てきて、まずそれらの意味を調べて理解することから始めないといけず、かなり苦労しました。

またそれに加えて、要件定義書自体が非常に文量の多いものだったため、読み込みに数日間かかった覚えがあります。今後こういう機会があった時のために、専門用語の理解を進めてスムーズに読み込みができるようにしていく所存です。

二つ目は、システムで採用される Python や Django の使い方の履修です。始めは仕組みがよく分からなかったのですが、触り始めて 1 週間くらいになると基礎的な部分を理解できるようになりました。ただ実際の開発では、さらに複雑にカスタマイズする必要があったのですが、Django ではカスタマイズするのが容易ではない部分もあったので、どうすれば目的の仕様を実装することができるのか、すごく頭を悩ませました。

また、自分がプログラムした部分でバグやエラーが出た場合、どうしてそのようなことになったのかを説明する必要があったり、バグ修正も自分で原因を調査して行うなど、自分で責任を持って仕事をしなければいけないのが初めてだったのですごく大変でした。

三つ目は、テストデータの作成です。開発したシステムの性能テストのために、大量のテストデータが必要になったため、mysql に入れるデータの作成に取り掛かりました。テストデータを csv で作成するため Excel を使用したのですが、まずテストデータの量が膨大だったため Excel の動作が非常に重たく、思ってた以上に作成に時間がかかってしまいました。

ローカル上の mysql で作成した csv が入ることを確認した後に、リモート上の mysql にも入れようとした段階で、セキュリティの関係で csv で mysql にデータを入れることが不可能であることが、この時判明しました。

解決策を提案してもらえたので何とかなりましたが、これからはどのような形式でデータを入れればいいか、データ作成を効率化するために何をすればいいかを事前に聞いていこうと思いました。

また、Elasticsearch のテストデータを複数作成して一気に入れた時に、ファイルサイズが大きいため途中で too many request になってしまうデータが発生し、どのファイルが抜けたのか調査するのに時間がかかってしまいました。こういったことが起きた時のために、ログファイルを作成することを学びました。

大学でもプログラミングを主に履修していましたが、実際に現場で開発の仕事をして、右も左も分からないことだらけで、これからが不安でもあり楽しみでもあります。この OJT 期間で得た経験を糧にし、これからも開発の仕事を頑張っていきたいと思います。

あとがき

ちょうど設計が始まる時期に、参加することができて、設計からリリースまでの一連の開発工程を体験してもらえたので、ちょうどよいプロジェクトに参加できたと思います。この経験を次の仕事に活かしてもらいたいです。

このプロジェクトは、できるだけ Django Admin の標準機能を使って、作り物を少なくしてコスト圧縮するという課題があったのですが、そこは、うまく行かないことが多かったです。私も勉強することが多々ありました。（開発部 佐藤）

06 ユニットテストのメンテにかかる時間の解消法

開発部 佐藤

鬼滅の刃を全巻買ったはいいけど、ぜんぜん読めてなくて、まだ 1 巻が終わってません。私の場合、通勤時間に漫画を読むことが多かったのですが、リモートワークは、通勤時間を無くしたが、隙間時間も無くしてしまったようです。

他にも、残念なのが、隙間時間がなくなって、 ”変なこと” を考える時間も減ってしまったことです。なんか寂しい。

まぁ、そんなことはほっといて、今号では、テストケースの実装について、つい最近取り組んだ、問題解決をご紹介させていただきます。

あるプロジェクトで、単体テストの実装およびメンテに、本体プログラムの実装時間の数倍の時間がかかっていて、生産性が上がらないという課題が発生しました。

時間がかかっている主な要因は、モックアップを作る作業とテストデータを実装するところです。テストパターンのバリエーションが多いと、単純にテストデータを作成するのに手間がかかります。さらに、テストパターンを流すためのモックアップの仕立てには、もっと手間がかかります。

過去に実装したテストのメンテも問題の 1 つで、たった 1 つの項目追加の仕様変更が、その項目を保持するテーブルのテストデータの修正が必要で、そのテーブルを使っているテストケースにも修正が入ります。自分が作成していないテストケースを修正するときは、テストの意図を理解して、テストを壊さないように修正しないといけないので、コードリーディングに時間を取られます。テストコードには、テストの意図が詳細に記述してあることが、あまりないので、怪しい解釈をしてしまうこともあったりします。

実装の歴史が進むと、こういう修正が、あちこちに出て来て、メンテに時間が取られます。本来の機能に対するテストコードの実装ではなく、過去に実装したコードのメンテ作業に時間を取られるのは、気分的にも乗らない苦行なので、生産性を落とします。

このような課題について、調査検討を行いました。

解決策は、3 つの “やらない選択” をすることでした。

1) IT でテストされることは、UT ではやらない
2) 異なる層のテストはしない
3) 3 つ以上の組み合わせテストはしない（最大 2 つの組み合わせまで）

テストのスコープを小さくすることを徹底します。組み合わせを最大 2 にするというのは、例えば、ある処理が public メソッドで実装されていて、その処理内容が大きく 5 つのステップで構成されているとします。正攻法でユニットテストを実装すると、5×5 のテストパターンが必要になります。解決策は、テストする 1 つ目のステップと、それ以降のステップを 1 つにまとめたステップグループとの実行にするという方式です。2 つ目以降も 1 つずらして同じように組み合わせていきます。

ステップ 1 と、ステップ 2〜5 のステップグループでテスト

ステップ 2 と、ステップ 3〜5 のステップグループでテスト

ステップ 3 と、ステップ 4〜5 のステップグループでテスト

ステップ 4 と、ステップ 5 のテスト

上記がそれぞれ 2×2 のパターンなので、25 から 16 パターンに削減されます。

そして、5×5 のパターンを相手にするよりも、2×2 を相手にする方が、格段に楽になり、思考の効率があがります。他の人がメンテするときも、2×2 の方が、コードを素早く理解できるので、スコープを小さくする効果は大きいと思います。

もう 1 つ、テストパターンの可視化についても考えました。

テストコードは、読んで理解するのが結構大変です。

アジャイル開発だと、仕様がドキュメント化されていないことが少なくないので、実装されたテストが、どういうテストなのか、本体のプログラムも熟読する必要があったりすると、どんどん生産性が落ちていきます。

そこで考えたのが、「テストコードからデシジョンテーブルをドキュメント出力するツールを作ったらどうか？」ということでした。

デシジョンテーブルは、テストの組み合わせ表をエクセルで書くときによく使う表のことです。

テストコードに、決まったデータ構造で、テストパターンを定義しておくことで、そのコードを解析して html や md でデシジョンテーブルを出力することができると、テストパターンを可視化することができそうです。

あとがき：

テスト内容の理解を楽にすることを目的にした可視化用ツールですが、ツール作成は、別の工数と時間がかかるので、即効性に欠けるので、取り組んでいません。

どこかで、この手の開発を支援するツールづくりをやってみたいですね。（開発部 佐藤）

07 お菓子紹介

管理部　間宮

今期も日頃の感謝の気持ちを込めまして、アルタスファイブよりとっておきのお菓子をご紹介させていただきます。

例年は「アルタスファイブ　おやつグランプリ」で高評価を得たお菓子を取り上げておりましたが…人が集まることが難しい昨今、テレワークということもあり、、なかなか開催が難しく、、、お菓子の持ち寄り・評価人不足に陥っている状況です。

※「アルタスファイブ　おやつグランプリ」とは、普段の生活の中にある「アレ美味しかったよ」を持ち寄って、会社で購入して、月一の帰社日に、パートナーさん含めて、みんなで食べてみて、レビューを付けるという、ただそれだけのイベントです。

しか～し！やっぱり美味しいもの知りたいですよね！

美味しいもの食べたら誰かに教えたくなりますよね！！っというわけで、結局いろいろ探しちゃいました。食べました。厳選しました。そして辿り着きました。

題しまして……

～お仕事のお供に！ご褒美に！おすすめ脳活おやつ～

です！それではいってみましょう！

森白製菓　【めちゃうまラー油】

「ご飯に合うならあられにも合う！こだわりの沖縄産ラー油に刻みねぎと、ローストオニオンで絶妙なバランスに仕上げてあります」とのこと。仰る通りです。このネギ感が絶妙です。かわいいパッケージの個包装を開けると、食べる前からラー油の香りががちゃんとして、味も濃すぎず、程よくピリ辛。バリバリ食べれば眠気も覚めます。この森白製菓さん、種類も豊富で、他にも「しいたけ黒こしょうあられ」や「やみつきトムヤムクン」など一癖ある商品が目につきます。お仕事中だけでなく、お酒のお供にもよさそうです。

下鴨茶寮　【ちりめん山椒せんべい】

京都の老舗料亭・下鴨茶寮さんがつくるおせんべい。「良質なちりめんじゃこと国産実山椒」を使用しています。上品な中にピリリと効いた山椒がよいアクセント！この“痺れ”が、脳を活性化させるとかしないとか…。麻味好きにはたまらないかと思います。ちなみにこちら、雑誌ＢＲＵＴＵＳにも掲載されています。期待できますねぇ。。軽い食べ心地なので“小腹が空いた”、“ちょっと口寂しい”なんて時にぴったりです。

ラ・メゾン白金　【焼きモンブラン】

こちらは 2021 年秋冬からの新作。しっとりとした優しい甘さの生地にマロンペーストを丁寧にトッピングした一品。頬張れば、栗の香りと味わいがフワッと広がり、それはもう栗を存分に堪能できちゃいます。疲れた時は糖分補給！ですよね。コーヒーや紅茶と合わせれば気分もほぐれます。程よい大きさで満足感も◎。一仕事終えた後の、ちょっとしたご褒美に是非いかがでしょうか。

以上になります。お菓子をつまみつつお仕事頑張りましょう。

一日でも早く、平穏におやつグランプリを開催できる日を願って、、、

お菓子を探し続ける旅はまだまだ続きます…！

あとがき

間宮さんには、10 月から派遣で来ていただいて、さまざまな事務的な作業をご担当されてます。

こんなヘンテコな取り組みをまじめにやってくれる方です。

よろしくお願い致します。

次号もおいしいお菓子をご紹介していきます。

次号につづく