

学生による UX / アジャイルソフトウェア開発の YWT ふりかえり事例

浦本 竜

北九州市立大学大学院

w5mcb002@eng.kitakyu-u.ac.jp

佐野 隼輔

北九州市立大学大学院

w5mcb004@eng.kitakyu-u.ac.jp

奥村 潤

北九州市立大学大学院

w5mcb003@eng.kitakyu-u.ac.jp

山崎 進

北九州市立大学

zacky@kitakyu-u.ac.jp

要旨

チーム開発の経験が少ない初心者のみで構成されたチームにおける開発は、様々な要因で失敗をすることがある。そこで、チーム開発の初心者であった我々が行った3件のアジャイル開発手法と UX を取り入れた開発経験を YWT でふりかえることにより、教訓を導く。導かれた教訓によって、初心者のみで構成されたチームにおける開発で、初心者にありがちな失敗を回避できるのではないかと考えた。そこで本稿では以下2点のことを目的としている。

- チーム開発の初心者に対して、失敗を軽減する為の教訓を導く
- YWT の使用感を考察する

YWT でふりかえりを行い、教訓を導くことができた。また、次の開発の方針を定めることができた。教訓を導くことができ、次の開発の方針が定めることができる。YWT はふりかえり手法として有効であると考えられる。

将来課題として、導き出した教訓は、初心者のみで構成されたチームに適用して効果を確かめる必要がある。またふりかえり手法として YWT は普遍的に有効であるかを調査する必要がある。

1. はじめに

チーム開発の経験が少ない初心者のみで構成されたチームにおける開発は、様々な要因で失敗をすることが

ある。失敗の原因は、チームにおけるメンバー間の連携がうまく取れないことや経験不足等が考えられる。我々は2014年に行った開発において、初めてチームで開発する機会があった。その時の開発で多くの失敗を経験し、それらについての対策を考えた。また、我々は2015年においてもチームで開発する機会があり、2014年に考えた対策を実行してみたところ、2014年に比べ、良い開発になったと考えられる。そこで我々の開発事例を YWT というふりかえり手法を用いてふりかえり、教訓を導くことで、初心者のみで構成されたチームにおける開発が初心者にありがちな失敗を回避できるのではないかと考えた。そこで本稿では以下2点のことを目的としている。

- チーム開発の初心者に対して、失敗を軽減する為の教訓を導く
- YWT の使用感を考察する

本稿は以下の構成になっている。次の2章では YWT について紹介する。3章で3件の開発事例について紹介する。また、それぞれの開発事例について YWT でふりかえり、失敗を軽減するための教訓を導く。4章でまとめる。

2. YWT とは

YWT というフォーマットを利用して、ふりかえりを行う。YWT とはやったこと (Y)、わかったこと (W)、次にやることの (T) の頭文字をつなげたものである。

まず、YWT について、倉貫義人は次のように述べている [1]。

以前に紹介した KPT でふりかえるのが 1～2 週間程度だとしたら、YWT では半年単位くらいでふりかえりと戦略について考えます。

この YWT を考えるためには、将来に自分がどうありたいかといった漠然とでもビジョンがなければ難しいでしょう。YWT を考える機会は、自分の未来のことを考えるきっかけでもあるのです。

また、松尾睦は次のように述べている [2].

先日、ある方から「YWT」というふりかえりの手法を教えてもらった。(中略)「これは面白い」と思ったのだが、この手法は、コルブ (Kolb) の経験学習モデルに対応している点に気がついた。コルブによれば、人は「具体的な経験を→内省して→教訓を引き出し→新しい状況へ応用」することで学ぶという。つまり、YWT 分析は

「何を実施したのか (具体的経験) (Y)
「何がわかったのか (内省&教訓) (W)
「次に何をすべきか (新しい状況への応用) (T)
を考えることで、職場の経験学習を促進することができる。

文献 [1], [2] より、3 章で紹介する、我々のチームで行った 3 つの開発事例の開発期間が約 4ヶ月であることと YWT でふりかえることによって、教訓を導き出し、初心者のみで構成されたチームにとって、より良い学習になるのではないかという 2 つの考えから、YWT を採用することにした。本稿では、我々が行った開発事例を以下のようにふりかえる。

1. やったこと (Y) で具体的体験をふりかえる
2. わかったこと (W) で 1 の内容から、わかったことや気づきをふりかえる
3. 次にやること (T) で 2 の内容から次にすることを考える

このフォーマットの、わかったこと (W) が教訓になり得る。改善が見込めるわかったこと (W) は次にやること (T) で、改善できるような仮説を立てる。その仮説を実際に次の開発で実行し、再び YWT のふりかえりを

行う。YWT を繰り返すことにより、より良い学習につながり、より良い教訓が得られると考えられる [3].

3 開発事例と YWT によるふりかえり

本章では我々の 3 件の開発経験を説明し、YWT でふりかえりを行う。

3.1 2014 年における WEB アプリケーションの開発事例

我々が 2014 年に行ったチーム開発によるソフトウェア開発事例について説明する。

3.1.1 開発概要

表 1 に開発概要について示す。

表 1. 開発概要

開発人数	5 名
開発期間	9 月 20 日～12 月 20 日
アプリ名	バトリング
プラットフォーム	Ruby on Rails
ソースコード行数	約 2500 行 (自動生成の行も含む)

開発したソフトウェアの説明を以下に示す。

- 対象ユーザーは対戦型ゲームで遊ぶ人と定めた。
- 対象ユーザーが同じゲームで遊ぶユーザーと親交を深めることができるように、コミュニケーションをとれる機能を開発した。
- 実力を試したいユーザーがトーナメント式の大会を簡単に開催・参加できる機能を開発した。

3.1.2 ソフトウェア開発経験

ソフトウェア開発経験を 2014 年 9 月 20 日～10 月 31 日、11 月 1 日～11 月 30 日、12 月 1 日～20 日の大きく 3 つの期間に分ける。

1. 2014 年 9 月 20 日～10 月 31 日について
この期間では制作するソフトウェアの仕様とその対象ユーザーを定めた。10 月 20 日までを Ruby on

Rails の学習期間とした。学習期間が終了した後に開発を開始した。開発にはアジャイルソフトウェア開発を適用して行った。11月の期間ではイテレーションを繰り返した。

2. 11月1日～11月30日について

- 進捗の確認の不十分で、プロジェクトが遅延した。従って、定例進捗会議を行うようにした。
- プロジェクトを管理する役を決めていなかったため、プロジェクトの全体を把握することができなかった。従って、プロジェクトマネージャーを選出した。
- 開発メンバー間でユーザーストーリーの優先度の認識がかみ合っていないなかったため、ユーザーストーリーの優先度の統一を行った。
- 設計の際、開発メンバー間で認識がかみ合っていないことが発覚したため、開発メンバーの認識を統一するために、プロトタイプとして、紙にインターフェースを手書きしたペーパープロトタイプを作成した。
- 作成・編集されるファイルの変更履歴を管理していなかったため、全員でのファイルの共有とバックアップができなかった。また、ファイルの共有が出来ていなかったのもでマージは変更したファイルを渡し、手動でマージした。従って、この問題を解決するために、バージョン管理ツールを導入してソースコードを開発メンバー間で共有した。
- フィードバックを得ることなく開発を進めていたため、ユーザーに価値があるものを実装できているのか不明瞭だった。この問題を解決するために、想定ユーザーが使用している様子を観察することでフィードバックを得た。

3. 12月1日～26日について

チーム開発が終了した後、プロジェクト全体のふりかえりを行わなかった。チーム開発の経験者に指摘され、プロジェクトのふりかえりを行った。

3.1.3 やったこと (Y)

- 数々のチーム運営における失敗から、改善行動を行った。

- 1週間おきに定期会議を行って、全員がプロジェクトについて考える機会を設けた。
- プロジェクトマネージャーの選出を行った。
- ファイルの共有や履歴を管理するため、バージョン管理ツールを導入した。
- 開発終了後に開発経験についてふりかえりを行った。

● UXを意識した設計を行った。

- ペルソナ法を用いて、ユーザー像を定義した。
- ペーパープロトタイプを作成した。

3.1.4 わかったこと (W)

- プロジェクトを管理していない時に比べ、プロジェクトを管理している時の方が作業が円滑に進む。
 - 定期会議を行うことにより、進捗の確認や現在の問題点等の情報をメンバー間で共有することで、次の週の開発における方針決めや、問題点に対応できる。
 - プロジェクトマネージャーがチームにいない時は、会議を行う間隔がばらばらであった。しかし、プロジェクトマネージャーがプロジェクトを管理することにより、1週間おきの定期会議を行うようになった。それぞれのメンバーの進捗を全員で共有する場を設けることで透明性が向上し、プロジェクトが円滑に進む。
 - メンバー全員が常に最新のファイルを持つことができ、バージョン管理ツールを導入していなかった時と比べ、時間短縮となる。
 - メンバー全員参加のふりかえりを行うことで、それぞれのメンバーが感じた成功したことと失敗したことを共有でき、次の開発における仮説を立てることができる。
- UXを意識することによってプロダクトの品質が上がるだけでなく、一種の指標になる。
 - ペルソナ法を用いてユーザー像を定義したことにより、実装する機能がユーザー像に対してどのような効果をもたらすかという視点で考えることができる。

- ペーパープロトタイプを作成することにより、認識の統一を計ることができる。また、ペーパープロトタイプは、ページレイアウトを考える際に想定したユーザーが使いやすく、使用後の満足感が高くなるように意識して作成した。

3.1.5 次にやること (T)

- プロジェクト初期に、開発が円滑に進むような仕組みを導入する。
 - プロジェクトの状態を把握できるような仕組みを調べ、導入する。
 - プロジェクトマネージャーを選出する。
 - プロジェクトの始めにメンバー全員でプロジェクトについての考えを共有する。
 - バージョン管理ツールとしてGithubを用いたが、習得が不十分であったため、次のプロジェクトが始まる前までに習得しておく。
 - 定期的にふりかえりを行う。
- 開発を通して、UXを意識した開発の経験を積む。
 - ユーザーの価値を創造できるような方法を考え、導入する。

3.2 2015年におけるWEBアプリケーションの開発事例

我々が2015年に行った、チーム開発によるソフトウェア開発事例について説明する。

3.2.1 開発概要

表2に開発概要について示す。

表 2. 開発概要

開発人数	3名
開発期間	9月20日～12月20日
アプリ名	Gift to
プラットフォーム	Ruby on Rails
ソースコード行数	約2600行(自動生成の行も含む)

開発したソフトウェアの説明を以下に示す。

- 一言で言えば、店舗情報提供サービスである。これは現在地周辺にある店舗がわかり、店舗に行かせる動機付けをさせる狙いで開発した。
- 対象ユーザーは実店舗にプレゼントを買いに行く20歳前後の女子大学生と定めた。
- アイテム別のチェックボックスを用意し、検索機能とした。検索した結果、検索条件に合致したお店が表示される。さらに、お店を選ぶとお店の詳細情報を確認することができる。

3.2.2 ソフトウェア開発経験

ソフトウェア開発経験を2015年9月20日～10月31日、11月1日～11月30日、12月1日～20日の大きく3つの期間に分ける。

1. 2015年9月20日～10月31日について

- チームビルディングの一環として、インセプションデッキ[4]を作成した。その際にプロジェクトマネージャーの選出を行った。使用するフレームワーク、チケット管理ツール、バージョン管理ツール等を決定した。
- プロダクトの方針を決定するために、事前アンケートを実施した。アンケートの対象は20歳前後の大学生であり、100件以上回収した。
- GitHubやRuby on Rails等の使用するツールについて事前に学習した。
- ペーパープロトタイプを作成した。
- ER図の作成を行った。
- 我々が定めた、ユーザーが目的を達成できる最小限のプロトタイプを開発した。

2. 11月1日～11月30日についてユーザーが目的を達成できる最小限のプロトタイプが出来た段階でアプリケーションを使ってもらい、その様子を観察すること、及び使用後の感想を聞くことという2点をフィードバックとして開発を続けた。ユーザーテストは1度目に2人、2度目に3人の計2度、5人に対して行った。

3. 12月1日～26日について

ページデザインをブラッシュアップした。
プロジェクト終了後に KPT で、プロジェクトのふりかえりを行った。

3.2.3 やったこと (Y)

- プロジェクトを始めるにあたって、プロジェクトが円滑に進むように新たな方法を導入した。
 - － チケット管理ツールを用いて、プロジェクトのタスク管理を行い、タスクの可視化を行った。
 - － プロジェクトの始めに、チームビルディングの一環として、インセプションデッキの作成を行った。この時、プロジェクトマネージャーの選出を行った。
- UX を意識したアプローチを新たに導入した。
 - － アプリを作る方針を決定した後、想定ユーザーが実際どのような点で困っているのかを知るために、開発前にアンケート調査を行った。対象は 20 歳前後の大学生であり、100 件回収した
 - － 想定ユーザーがある目的を達成できると予想される最小限の機能を開発できた段階で、実際に想定ユーザーに使ってもらった。使用中の様子を観察、また使用後にインタビューを行い、それをフィードバックとして次に開発する機能の参考にした。

3.2.4 わかったこと (W)

- プロジェクトの初期から開発環境を整える事で、余計なコストを抑える事ができる。
 - － 2014 年の開発では、ホワイトボードに書き留めるという方法でタスクを管理していた。その時と比べると、残りのタスクの確認コストが低くなることや、タスクを優先度が高い順に並べているので取りかかるべきタスクが認知しやすく、タスクの分担も容易である等の利点があった。
 - － 2014 年の開発では、認識のずれにより開発に余計なコストがかかってしまった。認識のずれ

はメンバー間におけるゴールの不一致や、大切な機能は何かということや、メンバーの役割は何かということ等で発生した。インセプションデッキでは、それぞれの役割、エレベーターピッチ作成等を定めるため、去年発生した余計なコストが比較的にかからず、プロジェクトが円滑に進んだ。

- UX を高めるために行った方法はいずれも有効な手段であると考えられた。
 - － 開発前にアンケート調査を行う事は、有効な手段であったと考えられる。しかし、アンケートの内容が深く練ったものでなかった為、得られた学びは少なかったと考えている。目的を明確にし、それに合わせたアンケート作りをする事で、得られる学びに違いがあるのではないかと考えた。
 - － ユーザーテストにより、得られたフィードバックを元に、機能やタスクの優先順を選び開発する事で、次のユーザーテストの際、ユーザーからよいフィードバックを得ることができる。

3.2.5 次にやること (T)

- プロジェクト運営の観点で、チームの生産性を上げる。
 - － チーム内で勉強会を開き、チームメンバーが互いに教え合う事で、チームのレベルが上がるのではないかと考えた。
 - － 我々は開発におけるテスト工程の際、コードを記述し、実行画面を開き、手動で動作確認をするテストのみを行っていた。そこで、テスト駆動開発を試し、手動で動作確認をするテストと比較する。比較した結果を考察し、今後の開発に活かす。
- ユーザーの価値について考え、チームにおける価値も高める。
 - － 目的をはっきりさせて、アンケートを作成すれば、さらに学びが大きくなるのではないかと仮説を立てた。次のプロジェクトでは、行動を起こす際、目的をはっきりさせる。

- － プロジェクトの始めに定めたペルソナであるが、これが妥当なものであるかという事を第三者にレビューを行ってもらう事で、UXの高いプロダクトが作成できるのではないかと考えた。

3.3 IoT デバイスの開発経験

3.3.1 開発概要

表3に開発概要について示す。

本事例はインターンシップの一環として行った開発である。インターンシップ先の企業の方を顧客と見立てて、毎週ミーティングを行い、顧客の要求するシステムを開発した。開発したプロダクトは一言で言えば、作業管理デバイスである。複数の仕事を持っている人がそれぞれの仕事に費やした時間を記録できるような仕組みをArduinoを基調としたデバイスとkintone[5]というクラウドサービスで構築し、作業管理デバイスとしている。デバイスの部品として、主にWi-Fiモジュール、タクトスイッチ、LED、SDカードリーダーを使用している。タクトスイッチを押すと、kintoneと通信が行われ、タクトスイッチを押したときの時間を記録する。LEDはデバイスの状態を表すインジケータとして使用した。SDカードリーダーはWi-FiのSSIDやパスワードをSDカードに記録し、より利便性が高くなるように使用した。

3.3.2 2015年におけるIoTデバイス開発経験

IoTデバイスの開発経験を2015年10月から2016年1月までを、1ヶ月ごとに分けて記す。

- 2015年10月1日～10月31日この期間は主に、仕様に関する事を顧客と話し合ったり、技術調査をした。
 - 定期ミーティングの場で顧客の要求を調査し、要求を満たせるようなシステムを提案した。例えば、通信方式をWi-FiにすることやLEDはデバイスの状態を表すインジケータとして使用すること等である。
 - 通信方式が決定してから、Wi-Fiモジュールの調査を行った。顧客の要求の中にHTTPS通信をすることと可能な限り低コストなものを
- 2015年11月1日～11月30日この期間は主に、デバイスを組み、通信テスト等を行った。
 - 使用したいというものが存在したため、条件に合うようなWi-Fiモジュールを調査した。
 - 必要な電子パーツをリストアップして、調達した。
 - Wi-Fiモジュールを使うための初期設定を行った。
- 2015年12月1日～12月25日
 - デバイスの回路図を作成し、回路図の通りにデバイスを作成した。
 - kintone上にアプリケーションを開発した。このアプリケーションで作業時間の記録を確認することができる。
- 2016年1月5日～1月31日この期間は最終調整を行った。
 - タクトスイッチを押すと、時刻データを送信する機能を実装した
 - LEDを実装し、デバイスの状態を知ることが可能となった。
 - SDカードモジュールをデバイスに組み込み、SSIDやパスワードを登録を簡単にした。
- 2016年1月5日～1月31日この期間は最終調整を行った。
 - データが送信失敗した時に再送する機能を実装した。
 - 2016年1月26日時点で、デバイスにかかったコスト計算書を作成した。

3.3.3 YWTによる事例のふりかえり

3.3.4 やったこと (Y)

- クライアントの要求を調査し、システム案を作成した。
- デバイスに関するやったこと。
 - － Wi-Fiモジュールの初期設定や必要な電子パーツのリストアップ等で時間がかかった。
- 文書を作成した。

表 3. 開発概要

開発人数	3名
開発期間	2015年10月1日～2016年1月31日
プロダクト名	作業管理デバイス
プラットフォーム	Arduino
ソースコード行数	660行

- 消費電力に関する調査報告書を作成した。
- デバイスにかかったコスト計算書を作成した。

3.3.5 わかったこと (W)

- 顧客と認識をすり合わせる難しさと大切さがわかった。
- ハードの知識が乏しいということがわかった。
- 文書を作成する際、他者が読み、理解できるように書くことが難しいことがわかった。

3.3.6 次にやること (T)

- 顧客が要求の背景に考えていることを意識して、知る努力をする。
- デバイス構築のための電子パーツの知識が乏しいので、早急に必要になりそうなパーツをまとめて、先達のレビューを受ける。
- 普段からアウトプットする習慣をつけ、文章を書く練習をする。

4. まとめ

4.1 YWT によるふりかえりから得られた教訓

YWT によるふりかえりから以下 4 点の教訓が得られた。

- プロジェクトを管理していない時に比べ、プロジェクトを管理している時の方が作業が円滑に進む。
- UX を意識することによってプロダクトの品質が上がるだけでなく、追加しようとしている機能は想定ユーザーにとって価値があるのかといった考え方ができるようになるため、一種の指標になる。

- プロジェクトの初期からチームビルディングやタスク管理ツールを決定する等の開発環境を整える事で、認識のずれによる手戻りやツールを使わない事で発生する余計なコストを抑える事ができる。

- UX を高めるために、アンケートによる事前調査を行う事は有効である。

YWT で開発をふりかえる事によって、内省し、教訓を導く事ができた。この方法で得られた教訓は他の初心者のみで構成されたチームに適用し、有効性を確認する必要がある。また、YWT によるふりかえりの次にやること (T) から、2016 年における開発の方針が定まった。開発の方針を以下に記す。

- チーム全体の開発レベルを上げるために、チーム内で勉強会を開く。
- テスト駆動開発を試す。
- 行動を起こす際、目的をはっきりさせる。
- ペルソナ設定をより良いものにするために第三者にレビューをもらう。
- 顧客が要求の背景に考えていることを意識して、知る努力をする。
- デバイス構築のための電子パーツの知識が乏しいので、早急に必要になりそうなパーツをまとめて、先達のレビューを受ける。
- 普段からアウトプットする習慣をつけ、文章を書く練習をする。

4.2 YWT を使ってみてどうだったのか

3 章で紹介した開発事例では、開発を行っている最中は 1 週間ごとに KPT でふりかえり、また開発期間が約 4 ヶ月の各事例について YWT で振り返った。YWT のわ

かったこと (W) はやったこと (Y) を振り返ったとき有効であった経験を教訓とし、さらに改善するための仮説があれば次にやること (T) で改善するための仮説を立てる。やったこと (Y) で改善する必要がある場合は次にやること (T) に改善するための仮説を立てる。このやったこと (Y) からわかったこと (W) を挙げる内省は KPT の K と P に相当すると考えられる。次にやること (T) はわかったこと (W) から挙げられることと同様に、KPT は K と P から T を挙げるので YWT の次にやること (T) と KPT の T は同質のものだと考えられる。YWT と KPT の違いは YWT のやったこと (Y) を可視化するところである。KPT の K と P は実経験から挙げていくとはいえ、プロジェクトの実体験を可視化しないため、YWT のやったこと (Y) を挙げてから、わかったこと (W) につなげるプロセスが KPT の K や P を挙げることより簡単だと考えられる。故に開発期間中をふりかえる場合は KPT が適しており、長期の開発経験をふりかえる場合は YWT が適していると考えられる。

また、やったこと (Y) で開発中に行ったことを列挙した。わかったこと (W) で具体的体験を内省し、教訓を導き出した。次にやること (T) では、開発活動を改善するための仮説を立てた。仮説を立てることで、次の開発の方針を定めることができた。教訓を導くことができ、次の開発の方針を定めることができる YWT はふりかえり手法として有効であると考えられる。

4.3 将来課題

導き出した教訓は、初心者のみで構成されたチームに適用して効果を確認する必要がある。またふりかえり手法として YWT は普遍的に有効であるかを調査する必要がある。

5 謝辞

2014年と2015年におけるWEBアプリケーションの開発事例において、ハウインターナショナル 村上照明氏、IoTデバイスの開発経験において、名古屋大学 舘伸幸氏、AISIC 久米純矢氏にとくに謝意を記す。

参考文献

- [1] 倉貫義人, ビジョンを叶えるために。個人でも出来る戦略を考える第1歩 ~ YWT を使った戦略の立てかたとは, <http://kuranuki.sonicgarden.jp/2013/06/ywt.html>
- [2] 松尾陸, ラーニング・ラボ, <http://blog.goo.ne.jp/mmatu1964/e/585f7611fd43892f871e3aa0f2ffe4bb>
- [3] 松尾陸, 職場が生きる人が育つ「経験学習」入門, ダイヤモンド社, 2011
- [4] Jonathan Rasmusson, アジャイルサムライ-達人開発者への道, オーム社 開発局, 2011
- [5] サイボウズ, kintone, <https://kintone.cybozu.com/jp/>