技術プラクティスの整理に1年半向き合ってわかったこと

—

2023/7/1 スクラムフェス大阪

常松祐一

常松祐一

Retty株式会社
プロダクト部門長 執行役員 VPoE
立ち食い蕎麦担当

顧客にとって価値のあるプロダクトを、チーム一丸となって協力し、短期間にリリースする開発体制のあり方を模索しています。

書籍を出版します！

アジャイルプラクティスガイドブック
チームで成果を出すための開発技術の実践知

チーム・組織にプラクティスを導入し、根付かせるために!

116の手法を一冊にまとめた“実践”の手引き

著者　:常松 祐一(著)、川口 恭伸(監修)、松元 健(監修)

発売日：2023年7月20日

定価　:2,860円（本体2,600円＋税）

出版社：翔泳社

企画概要

本書は、アジャイル開発における「技術プラクティス」の効果的な選択の仕方と、その活用の方法について解説する書籍です。

アジャイル開発の実現には、チーム環境を改善する「ソーシャルプラクティス」の取り組みと、開発環境（ツール・手法・技術）を改善する「技術プラクティス」の取り組みの、双方の検討が必要です。しかし、既存のアジャイル関連の書籍では、前者に該当する「チームづくり」「組織論」を掘り下げたものが多く、後者の内容をアジャイルの実践者向けに体系的に解説したものは多くありません。

本書は、技術プラクティスを選択・活用する際の実践的なポイントを、具体例を交えながら解説します。

※書籍企画書より抜粋

この辺りを扱っています

引用：アジャイルの「ライトウィング」と「レフトウィング」

書籍目次

第1章 アジャイルな開発を支えるプラクティス

1.1 プラクティスの実践

1.2 高速に石橋を叩いて渡る

1.3 広く知られたアジャイル開発手法とプラクティス

1.4 プラクティス理解に役立つ考え方

第2章「実装」で活用できるプラクティス

2.1 実装方針 / 2.2 ブランチ戦略 / 2.3 コミット

2.4 コードレビュー / 2.5 協働作業 / 2.6 テスト

2.7 長期的な開発／運用ができるソースコード

第3章「CI/CD」で活用できるプラクティス

3.1 継続的インテグレーション

3.2 継続的デリバリー

3.3 継続的テスト

第4章「運用」で活用できるプラクティス

4.1 デプロイ／リリース

4.2 モニタリング

4.3 ドキュメント

第5章「認識合わせ」で活用できるプラクティス

5.1 関係者との認識合わせ

5.2 開発内での認識合わせ

5.3 計画の継続的な見直し

第6章「チーム連携」で活用できるプラクティス

6.1 チームの基本単位

6.2 属人化の解消

6.3 パフォーマンスの測定

6.4 円滑なコミュニケーションのアイデア

6.5 意識を揃えるワークショップ

実践者のコラムを11編収録

1. チームで1つずつ終わらせよう 椎葉光行さん	2. ペアプログラミングの効果と影響 やっとむ（安井力）さん
3. テスト駆動開発ではTODOリストがテストよりも先 大谷和紀さん	4. 技術的負債の話 川口恭伸さん
5. インフラ構築を自動化しよう 吉羽龍太郎さん	6. Logging as API contract 牛尾剛さん
7. AIフレンドリーなドキュメントを書こう 服部佑樹さん	8. 開発と運用、分けて考えていませんか？ —ダッシュボードのその先へ—　河野通宗さん
9. チームに命を吹き込むゴール設定 天野祐介さん	10. 開発項目をコンパクトに保つには、クリーンなコードを 大谷和紀さん
11. グラデーションで考える12年間のアジャイル実践 きょんさん	

多くの方のレビューで強度を高めた本

技術プラクティスはアジャイルの達人たちが異なる見解を持つ分野であり、不完全な出版は厳しい批判(マサカリ)を招くことが予見されました。そんななかで、初心者にも読みやすく、アジャイル実践者にも受け入れられる内容をどのように作るか、そこに私たちの貢献があると考えました。したがって、本書のレビューはアジャイルコミュニティの実践者にお願いしました。彼らからの厳しいフィードバックにより、多くの修正を施すことができました。

※監修者序文より抜粋

【謝辞に掲載させていただいたレビュアーの皆様、本当にありがとうございました】

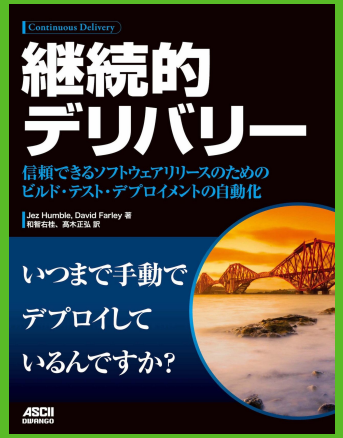
小田中育生さん、藤原大さん、大金慧さん、石毛琴恵さん、粕谷大輔さん、守田憲司さん、岩瀬義昌さん、粉川貴至さん、森田和則さん、伊藤潤平さん、山口鉄平さん、半谷充生さん、飯田意己さん、今給黎隆さん、木本悠斗さん、渡辺涼太さん、小迫明弘さん、池田直弥さん、今井貴明さん

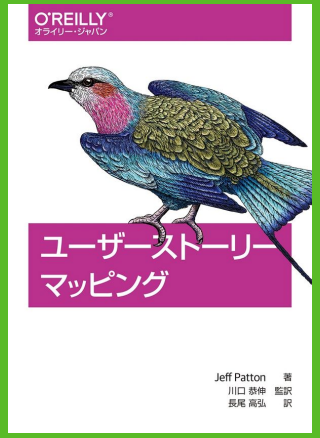
わかったこと1

- アジャイルプラクティスガイドブックは多くの方の協力で生まれた書籍
- マサカリは痛かったw

ありそうでなかった本と言われますが・・・
技術プラクティスを扱った書籍は無いのか?

特定の技術プラクティスを深く扱った書籍

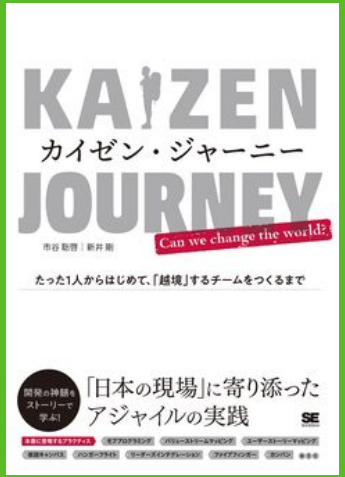




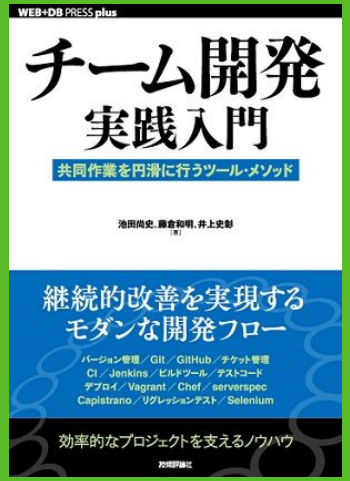
分野を絞った専門書はあるが、出版時期が古いものもあり、全て読むには数が多く、ブログや現場経験などを通じて差分を学んでいく必要がある状態

切り口は異なるが、見つけた類書

アジャイルに開発するスタンスを抽象度高く紹介



1人で始めて、チームを作るまでの経験を追体験

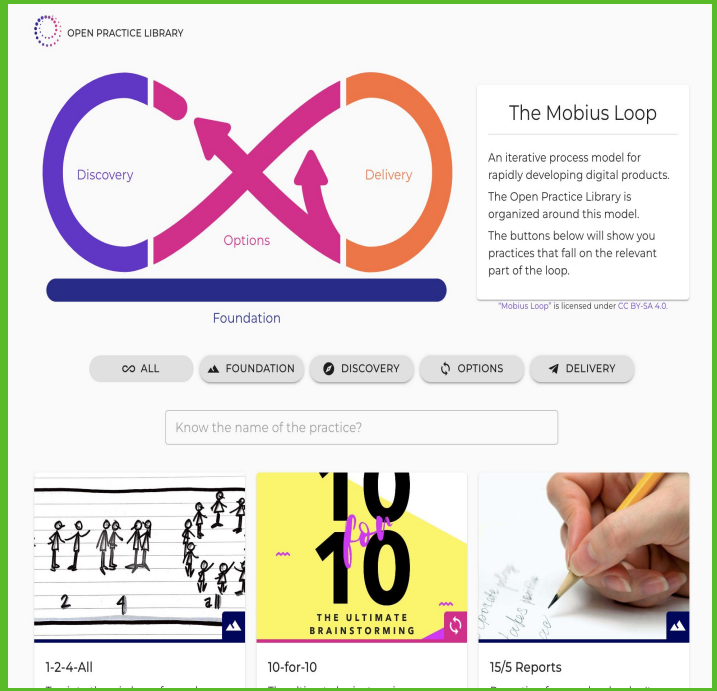


チーム開発で必要になる技術知識を紹介

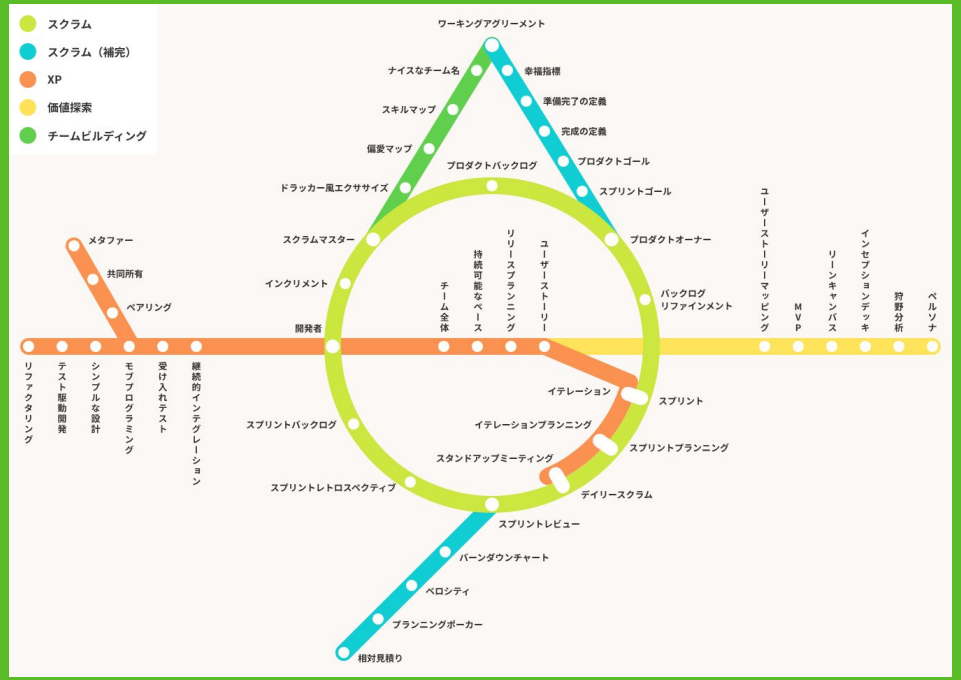
継続的な開発で必要になるスキルや取り組みの紹介

継続的な開発で必要なことを、学術的に整理

アジャイルプラクティスをまとめたサイト



Open Practice
https://openpracticelibrary.com/Library



アジャイルプラクティスマップ
https://www.agile-studio.jp/agile-practice-map

企業が技術情報をまとめ、公開しているサイト

Google / Google Cloud

- DevOps とは: 研究とソリューション Google Cloud

  https://cloud.google.com/devops?hl=ja

Microsoft

- ISE Code-With Customer/Partner Engineering Playbook

  https://github.com/microsoft/code-with-engineering-playbook

Atlassian

- ソフトウェア開発の主要部分について学ぶ

  https://www.atlassian.com/ja/software-development

研修・コミュニティ

Certified Scrum Developer
https://www.jp.agilergo.com/online-acsd-bernstein-202311

スクラムデベロッパーズナイト
https://smn.connpass.com/

アジャイルプラクティスガイドブックの
エレベーターピッチ

アジャイルプラクティスガイドブックは悩めるマネージャーと開発者の皆さんに向けて、アジャイル開発とその周辺の技術プラクティスについて説明を試みたもの。

マネージャーとして実際に技術プラクティスを組織に導入する仕事をしている常松が実体験に基づき書いた、技術プラクティスを「チーム全体で」学習するための道標となる書籍。

わかったこと2

- アジャイル開発の技術プラクティスを扱った情報源はある
- 筆者の実体験を元に、技術プラクティスを「チーム全体で」学ぶための書籍はなかった(のでは?)

書籍としてまとめる難しさ

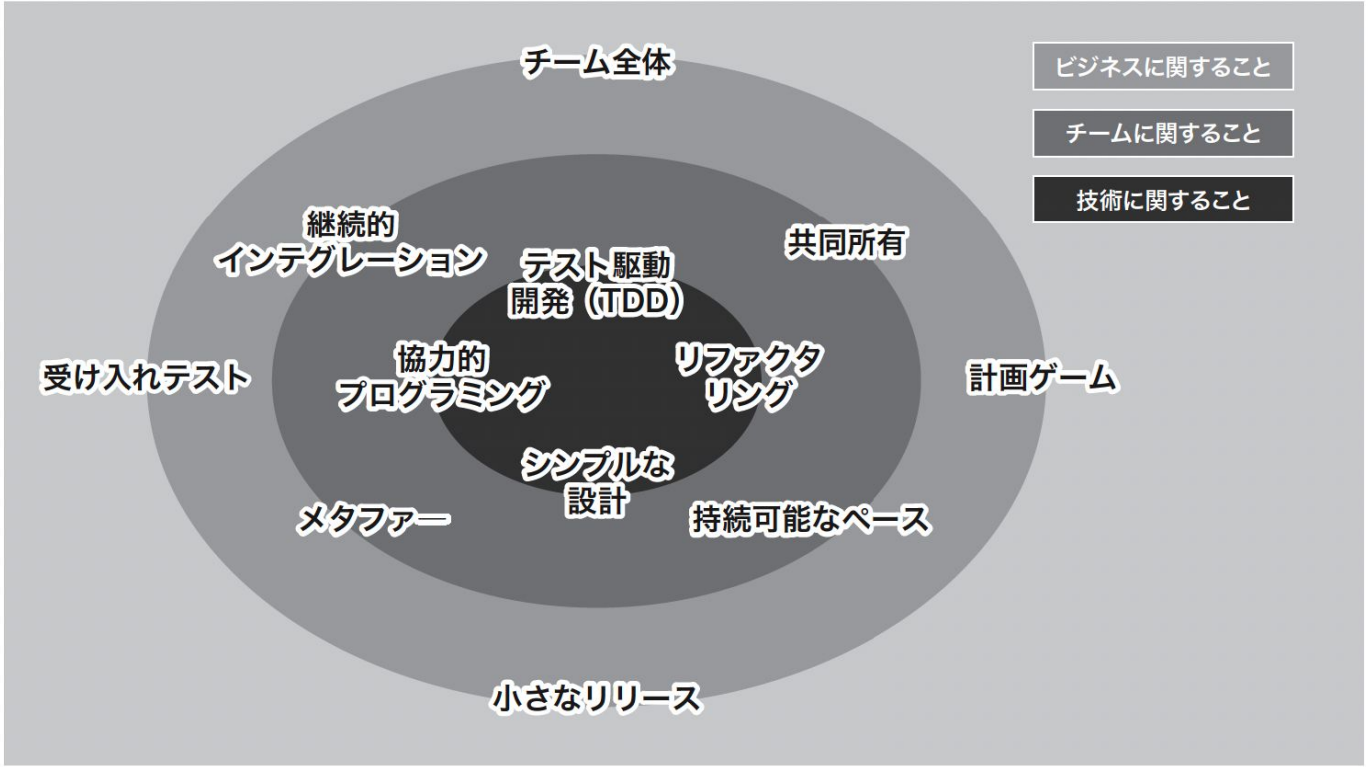
取捨選択とバランス

どの技術プラクティスを紹介するべきか。

書籍で「選ばなかったもの」もある。

- ソフトウェア設計・アーキテクチャ・技術選定の話
  - フワッとしそう。書籍の本筋と関係が薄い。
- カバレッジ、サイクロマチック数などのメトリクス
  - 広く認知されているが、扱いが難しい。
- 開発現場や開発環境に強く依存するもの
  - マイクロサービス
- 筆者が経験したことがないもの。
  - イベントストーミング、カオスエンジニアリング
- 今後主流になるかもしれないが判断に迷うもの
  - モノレポ、GitOps

図 1-2　エクストリームプログラミングのプラクティス



ちなみに・・・XP(eXtreme Programming)はよくまとまっています、さすが！

取捨選択とバランス

現場で役立つ情報と陳腐化のバランス

最新ツールや設定の紹介を詳しく行うか?

- 入れるほど具体的に役立ちやすいが、書籍として陳腐化しやすい。
- 入れるほど特定の現場環境に近くなるが、読者ターゲットが狭くなる。

→ 専門書籍や探し方へのポインタで止めたい。

体系化とグレーゾーン

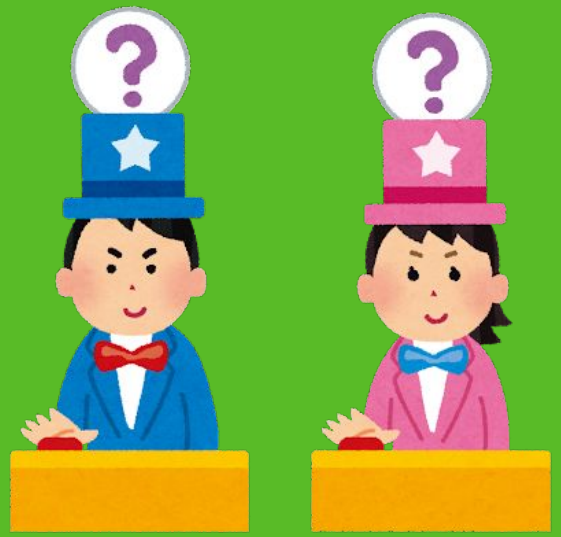
時間をかけて何度分類を見直してもグレーゾーンが生じる

- 「技術プラクティス」と「プロセス・チーム運営に関するプラクティス」のはざま
  - 技術プラクティスの紹介に絞った本にしたい
  - →開発の困りごとが解決できず片手落ちになるくらいなら広げたい
  - →プラクティスとは？
- アジャイル開発としての技術プラクティスと、良いソフトウェア開発のための知識のはざま
  - クリーンアーキテクチャとか、DDDとか

用語を正しく扱う

Q：あなたの現場ではどう呼んでいますか?
「プロダクトに追加する機能や要求を簡潔にまとめたもの」

① ストーリー	② ユーザー ストーリー
③ プロダクト バックログアイテム	③ タスク



出典を正しく扱う

Q:「スウォーミング」はどれに当たるでしょうか?

本書で取り扱うプラクティスについて

プラクティスは以下の区分で、各項目の冒頭で紹介しています。

出典のあるプラクティス

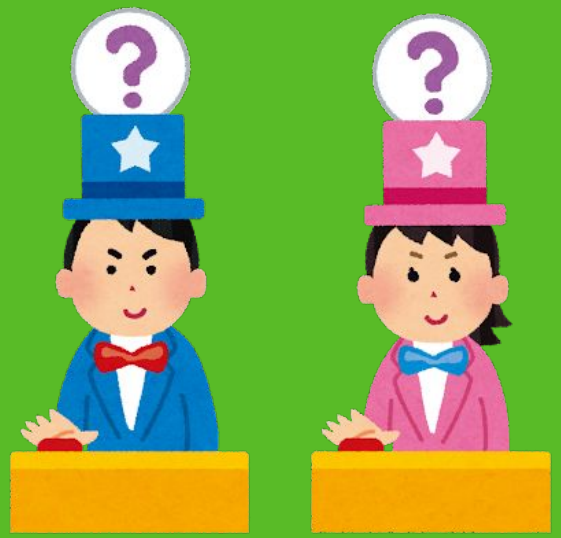
著名な書籍で紹介されるなど、出典が明確な広く知られているプラクティス

慣習的に知られているプラクティス

出典は明らかでないが、いくつもの現場で実践が観察されるプラクティス

この本で提案しているプラクティス

上記以外のプラクティスで筆者の経験に基づき紹介するプラクティス



読みやすさと理解のしやすさ

技術プラクティスを紹介する順番をどうするか

1. 開発の流れに合わせる

   計画→実装→テスト→運用

   実装までが長い

2. 読者がプラクティスを取り入れるであろう順番

   実装→計画→複数チーム

3. はじめに原理原則を短くまとめ、残りは辞書的に並べる

主張のバランス

書籍は著者の主張の塊！

- ･･･ではあるが間違えは良くない
- 一方で自信が無いのもダメ

わかったこと3

- 書籍としてまとめるのはブログや登壇の何倍も大変
- 細かなところまで色々考えて書籍をまとめました

収録した好きなプラクティス3選

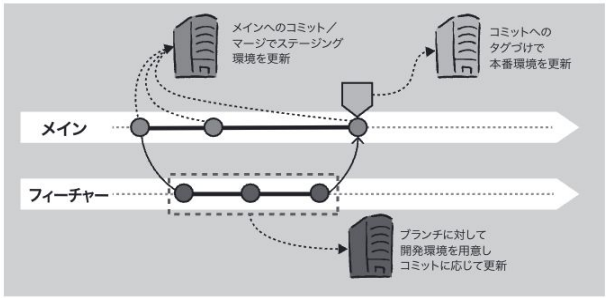
1. ブランチとデプロイ環境の話

メインブランチにステージング／本番環境を紐づける

まず最も基本的なパターンは、メインブランチにステージング環境と本番環境の両方を紐づける方法です（図 3-7）。メインブランチへのコミットやマージがあれば、自動的にステージング環境を更新します。検証して問題がなければメインブランチに対してリリースを示すタグづけを行います。メインブランチでリリースタグがつけられたら、本番環境を更新します。開発環境は用意しないこともあれば、ブランチに対して用意することもあります。

管理するブランチが少なく済むメリットがありますが、メインブランチの最新版が本番環境である保証がないため、リリースされている内容物を確認するのにタグから探す手間がかかります。

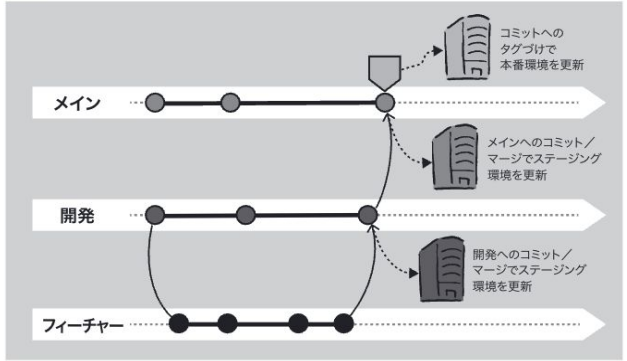
図 3-7　メインブランチにステージング・本番環境を紐づける



メインブランチと開発ブランチを分けて紐づける

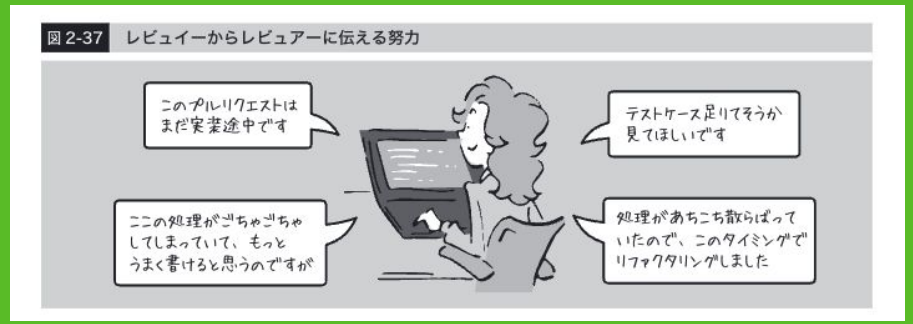
影響範囲が大きいサービスなど、開発環境でのテストを十分に行いたい場合、メインブランチと開発ブランチを分けることがあります（図 3-8）。複数のブランチを開発で運用する場合、修正を加える基点となるブランチを 1 つ選び、デフォルトブランチと呼びます。この場合は開発ブランチがデフォルトブランチとなり、修正時のブランチは開発ブランチから作ります。開発ブランチにコミット／マージすることで開発環境が更新されます。メインブランチへのマージでステージング環境が更新され、メインブランチでのタグづけで本番環境が更新されるのは先ほどと同じです。

図 3-8　メインブランチと開発ブランチを分け、ステージング環境をメインブランチに紐づける



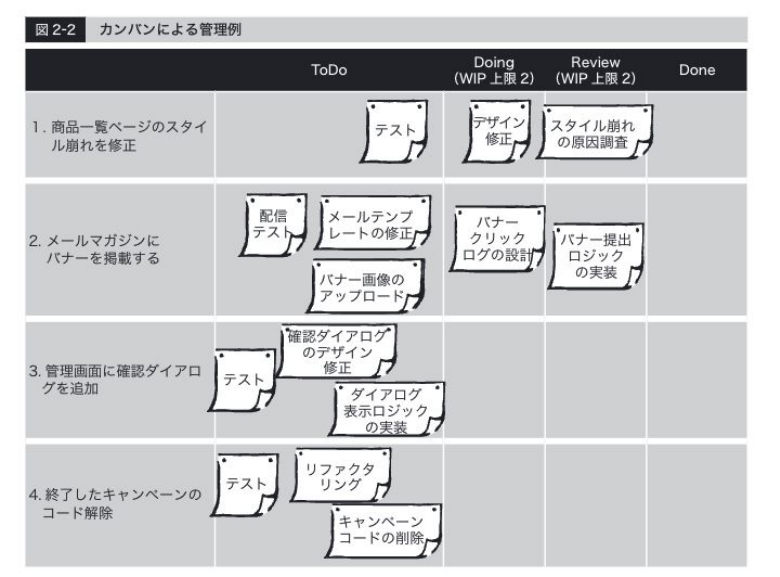
この辺りを書いた本はあまり無いのでは?

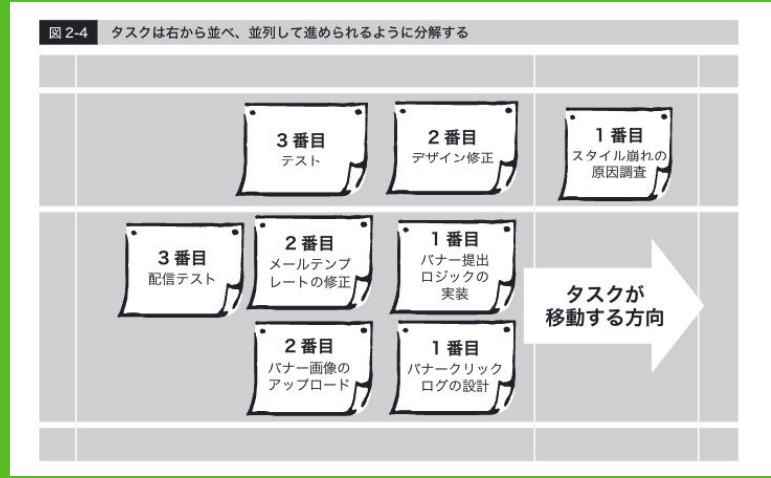
2. コードレビュー周りの話



コードレビューで喧嘩する現場と喧嘩しない現場があるらしい

3. タスク分解の話





タスクを右から並べるのが当たり前だと思っていたが、そうでも無いらしい。
※レビューにも参加いただいた守田憲司さんのアイデア(?)

わかったこと4

- 当たり前に使っている現場知識でも、出典に悩むことがある
- 筆者の当たり前は読者にとってのあたり前では無い

まとめ

まとめ

- アジャイルプラクティスガイドブックは多くの方の協力で生まれた書籍です。
- 筆者の実体験を元に、技術プラクティスを「チーム全体で」学ぶための書籍はなかったのでは。
- 書籍としてまとめるのはブログや登壇の何倍も大変。
- 筆者の当たり前は読者にとってのあたり前では無い。

書籍を書き上げての感想

書籍を読んでいて「思いついたアイデアをコミュニティやカンファレンスで議論しながら磨いていった」というくだりをよく見かけた気がします。この書籍は日本のアジャイルコミュニティ無しには誕生しませんでした。

コミュニティでの議論や集積をもとにした書籍が今後も生まれて欲しいし、海外進出も果たしていって欲しいなと思っています。

2016年ぐらいからアジャイル開発に取り組んでいる常松は、7年分の学びを一旦この書籍で出し切りました・・・

書籍に関連するイベント

CL MeetUp : 7/11(火)
https://creationline.connpass.com/event/287340/

CodeZine Night 8/4(金)