

"Swarming"を コンセプトに掲げる アジャイルチームのベストプラ クティス

@boykush 2024-09-28 XP祭り 2024

自己紹介

Taichi Kushiro / SWE

-  Alp, Inc.
-  boykush
-  boykush315

プロダクト

***Scalebase*»**

継続型サービス事業者向けの販売・請求管理SaaS。複雑なプライシングや継続課金契約に基づく請求をバッチで自動化

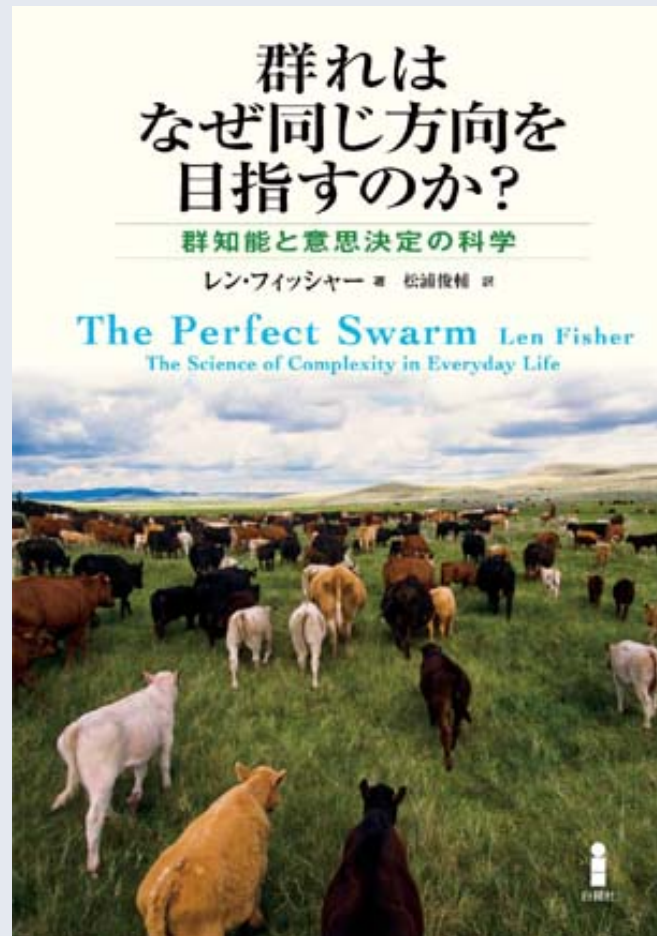
DDD x Agileで複雑なドメインを開発



Swarmingとは

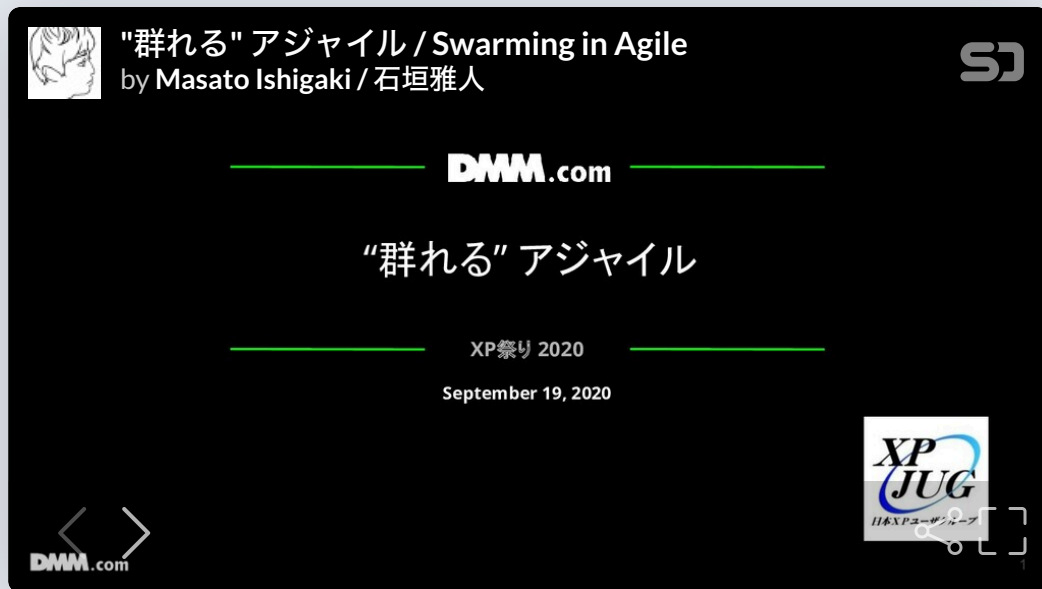
科学文脈のSwarming

- 単純な規則による自己組織化
 - アリの群れの例) 通ったルートにフェロモンを残す各個体が、最もフェロモン濃度が薄まっていないルートを選ぶことで最短ルートが導かれる



アジャイル文脈のSwarming

- インシデント対応やカンバンのWIP制限解消の際にチームで一丸となる
- 重要な領域や主要な活動にエネルギーを集中させ、素早く完了させる
 - フロー効率の重視、暗黙知の共有等



The slide features a black background with white and green text. At the top left is a small profile picture of a man. To its right is the title and author information. The center of the slide is framed by two horizontal green lines, containing the DMM.com logo and the title in Japanese. Below this, another set of green lines frames the event information. The bottom left corner has a DMM.com logo with a stylized bracket, and the bottom right corner has the XP JUG logo.

 "群れる" アジャイル / Swarming in Agile
by Masato Ishigaki / 石垣雅人

DMM.com

“群れる” アジャイル

XP祭り 2020
September 19, 2020





Swarmingの粒度と計画性

- インシデント対応やカンバンのWIP制限解消
 - → 小さい粒度を対象に自然発生的に群れる
- スプリント内のストーリー群の完了
 - → 粒度を拡げ計画的に群れる

計画的なSwarmingは優先度の高い集中するストーリー（とそれが含むタスク）を事前にチームに認識させる

私たちのチームのSwarming

私たちのチームの変遷

- スクラムからカンバン、両者のハイブリッドへと開発手法を移行
 - スタートアップ企業 x 複雑で大規模なドメインの組み合わせによる変化の多い環境で試行錯誤を行ってきた
- 「アジャイルな開発をやる」とは何か、を見つめ直しSwarmingの考えを導入。以降2年強経過
 - ユーザーストーリーを対象に計画的に群れる。課題の分割、開発プロセスの改善へと粒度を拡げる

モチベーションに近い参考ブログ

- Vin D'Amicoが投稿した「Agile Teams Swarm to Greatness」というタイトルを含む連載ブログの内容と近いモチベーションで運用

導入～アジャイルの難しさ～

- Agileという単語が与えるイメージの多様さ
- 乱立する複数の開発手法
 - XP,スクラム,カンバン,リーン, etc.
- 複雑なエンタープライズアプリケーションの増加

アジャイルな組織を特徴づける重要な言葉やフレーズを再定義したい

優れたチームは群れる

アジャイルな組織を特徴づける重要な言葉やフレーズの一つがSwarming

They are highly cohesive, loosely coupled and decentralized.

それらは非常に凝集性が高く、ゆるやかに結合され、分散化されています。

They implement constant feedback cycles and continuous improvement.

彼らは継続的なフィードバックサイクルと改善を実施しています。

They are known for rapid responsiveness and swarming behavior.

彼らは迅速な対応と集中的な取り組みで知られています。

They have diverse skill sets and share a common vision.

彼らは多様なスキルセットを持ち、共通のビジョンを共有しています

Swarmingをコンセプトに掲げ
優れたチームを目指す

チームで生まれたベストプラク
ティス

前提～開発チーム体制～

- エンタープライズ寄りの複雑で大規模なドメイン
- 基本はスクラム開発・カンバンにみられるイテレーティブなスプリント
 - 5人チーム、1スプリント1週間、週1リリース
 - 毎日の朝会、リファインメント、プランニング、レビュー、ふりかえり等一連のスクラムイベントを実施
- 対等な関係でリスペクトを持った対話を重要視
 - スクラムイベント持ち回り、アジャイル関連用語の認識合わせ、設計に対する違和感の言語化等
 - 対話がこの後話すプラクティスのベースとなる

前提 ～ プラクティス ～

- Swarmingを重要視するチームが実践してきたプラクティスを紹介
- オリジナル・一般的なプラクティス双方を含む
- 異なる文脈のプラクティスをコンセプトで繋ぐ物語のようなもの

群れるためにより良い
インターフェースを生み出す

群れるためにより良いインターフェースを生み出す

- アジャイル12の原則のうち設計のパートに対応するような活動
- シンプルなインターフェース（規則）を目印とし複雑なソフトウェアを構成していく

1. ユースケース駆動実装

- ユースケース駆動開発でいうシナリオを書く代わりに実装する
- ユーザーストーリーの着手時にモブプロを開催しメインブランチへのマージまで完了させる

ユースケース駆動実装の進め方

1. ユースケースのインプット・アウトプットを定義

- 振る舞いがどう呼び出されるか
- 極力分かりやすくシンプルに

2. ユースケースが依存するコンポーネント群の抽象または仮実装を定義

- 抽象: データベース、外部APIアクセス
- 仮実装: ドメインロジックのTDDで1テストケースのみ通す

3. ユースケースシナリオを記述するようにユースケース実装

- 技術的な関心を分離した結果なるべく自然言語に近い実装ができるとうい
- 書籍「関数型ドメインモデリング」参照

ユースケース駆動実装の効果

- 目的不確実性（どう呼び出されるか）と方法不確実性（どう実装するか）を交互に潰す
 - この時点でユースケースのテストを書くかは目的不確実性の高さ次第
- 呼び出される or 呼び出すインターフェースによって群れの単純な規則が生まれる

2. 領域の明確なサブタスク

- ユースケースに関連するコンポーネント群の本実装をサブタスクとして起票
- サブタスクのプレフィックスにドメインコンテキスト/レイヤを明記する
 - 例) 「invoiceCtx/domain 請求金額ロジックの本実装」

領域の明確なサブタスクの効果

- 責務範囲が明確になりメンバーの自律性が高まる
- 1つのユーザーストーリー内で複数人が並列作業可能になり群れやすくなる
- 小さい逆コンウェイの法則のようなもの

群れは足並みを揃え変化に適応する

群れは足並みを揃え変化に適応する

- DORAによるDevOps capabilitiesの「Capabilities that enable Fast Flow」に分類されるような活動
- 頻繁にブランチを統合し変化に適応するための基礎となる

3. フィーチャーブランチの禁止

- DevOps capabilitiesの「トランクベース開発」に近いが、やらないことにフォーカスして運用している
- 運用変更時はフィーチャーブランチを作っていないか某ひ○ゆきの声で質問されるbotが活躍

フィーチャータブランチの禁止の効果

- 開発バッチサイズの減少による安全なリリース
- 群れの障壁となるコンフリクトが減り認知負荷・心理的ハードルが減る

4. ストラングラーフィグパターン

- 巨大なコードベースに対してこれまでのプラクティスを適用できる？という課題を解決
- Martin Fowlerが提唱したアプリケーション移行パターン
 - bad: 修正
 - good: 追加→置換→削除
- チームでは置換サイズ大中小それぞれで実践
 - 大: 書籍「モノリスからマイクロサービスへ」参照
 - 中: 前述したユースケース駆動実装に対応したユースケース単位
 - 小: ドメインロジック関数、リポジトリ実装

ストラングラーフィグパターンの効果

- 置き換え時の差分が最小限になりフィーチャーフラグの代替となる
- 既存の巨大なコードベースに対しても群れることができる
 - 修正によって大量のコンパイルエラーが発生しないため個人が抱えこまない

孤独な群れを作らない

孤独な群れを作らない

- 群れることで当たり前前に暗黙知を共有し属人性を低下させる
- 対話を重視しチームの集中力・熱を絶やさない

5. 頻繁なモブレビュー

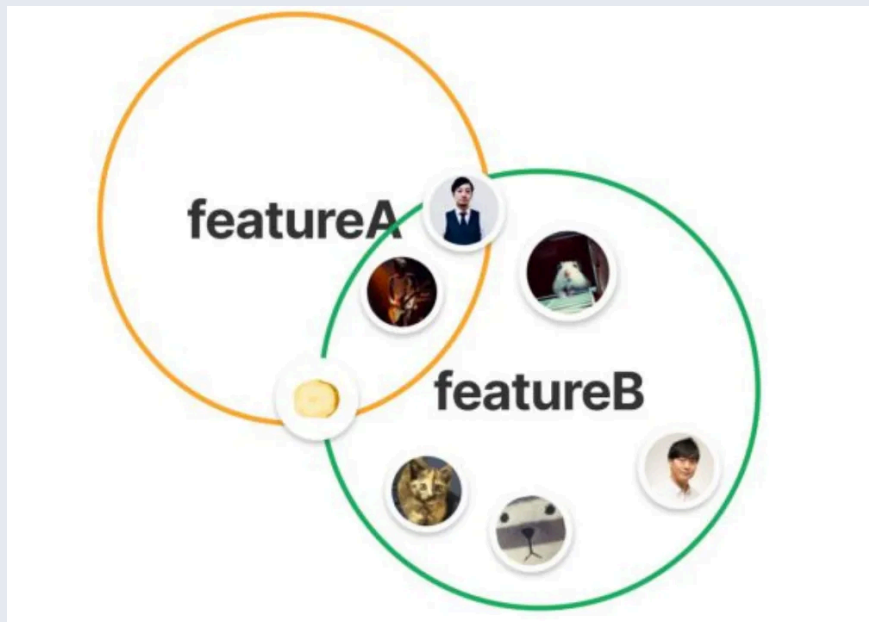
- 日々のレビュー依頼はチーム全体メンションで誰でも取れる
- 朝会後にモブレビューの時間を取り、review requiredなPull Requestを一掃する

頻繁な全員レビューの効果

- マージまでのリードタイムの減少
- Pull Requestベースでの効率的な暗黙知の共有
 - テンプレートがArchitecture Decision Recordのような形式だと尚良い

6. Swarmingサークル

- 誰がどのユーザーストーリー・差込対応に取り組んでいるのか、複数のタスクを兼任していないかをアイコンとサークルで可視化

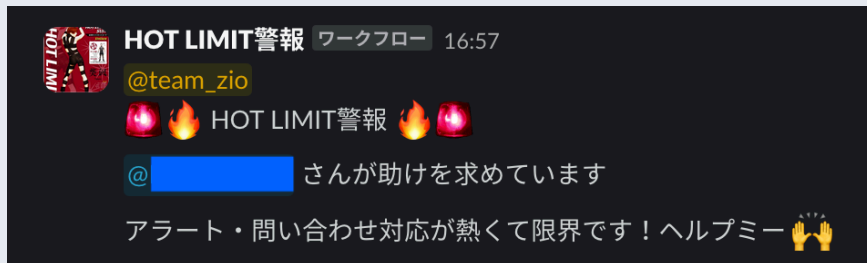


Swarmingサークルの効果

- Swarmingの並行度合いを確認しサークルが減る
 - ユーザーストーリー開発が並行してしまっていたSwarming導入期に役に立った
- 孤独な群れを作らないことで個人の心理的負荷を軽減する

チームの対話を活発にする取り組み

- raffee emoji
 - 週一回の出勤日はramen or coffee or 両方をチーム全員で摂取しに行く
- HOT LIMIT警報
 - アラート・問い合わせ対応がパンクしたらボタン一つでHOT LIMITの絵文字の通知を飛ばす



私たちはなぜ群れるのか

私たちはなぜ群れるのか

- 個人の能力を自律的に活かしながらチームで開発する意義を高められた
 - リモートワークでの悩みとなる個人とチームの時間のバランスを取りやすいユーザーストーリー単位のSwarmingがフィットした
- フレームワークを試行錯誤する中でチームにとってブレない軸を持つことができた
 - 原則・フレームワークどちらとも共存できる

すぐに見えるSwarming

アジャイルなチーム作りに悩む人の助けになるのではないか
実践例)

- チームリーダー
 - 分かりやすいコンセプトを掲げチームがアジャイルに近づく
- チームメンバー
 - プラクティスの実践によって群れる機会が増えチームがアジャイルに近づく

よいSwarmingライフを