フロントエンジニアの異常な愛情

または私は如何にして心配するのを止めてソリューションを提供するようになったか

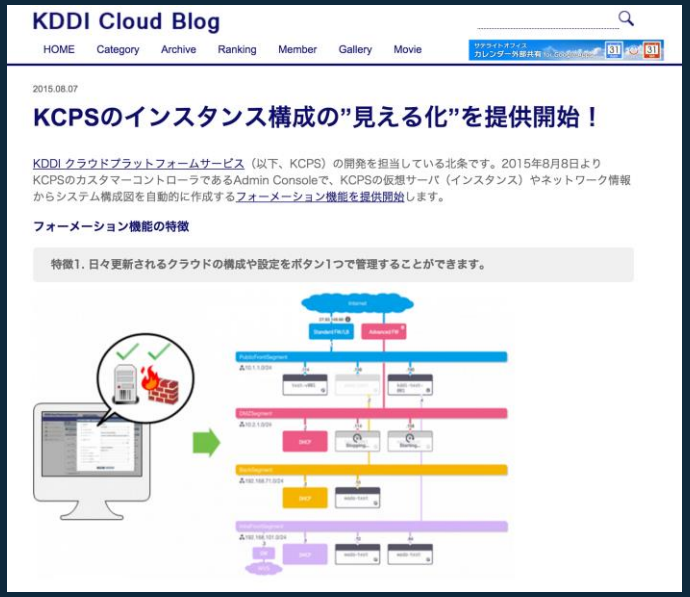
#アジャイルジャパン
初心者向けセミナー

1

プロダクトの紹介

Introduction of products

Kddi Cloud Platform Service Admin Console



http://cloudblog.kddi.com/news/2745/?v=block

回線＋クラウドを軸にした
国内クラウドサービスの
統合管理コンソール

As-Is

申し込みに２ヶ月…

何を作ったか判らない
クラウド

Before

As-Is → To-Be

申し込みに２ヶ月…

何を作ったか判らない
クラウド

Before

In half a year...

After

申し込み最大１０営業日

作成したリソースの
一括表示

2

半年の振り返りと課題

Looking back and challenges of half in year

Agile Team Development Cycle

Sclam

Problem…

自動テスト

自動テストをしているはずなのにバグやデグレが発生する
自動テストの作成と維持コストが高い

Product Backlog

StoryやBacklogの消化は早い
しかし出来上がったStoryやbacklogが繋がらない、特定部門や作業で負荷をかけてしまいトータルではあまり良くない。
結果改修しなければならないというbacklogが多々出る。

3 分析

Analysis

自動テスト

(1) テスト手法が整理されていない

テストが「テスト」というひとくくりの作業になっていた

(2) テストツールを適切に利用できていない

テスト開発のワークフローが成り立っていなかった

Product Backlog

(1) 思いつき駆動開発になっていた

作成するソリューションが思いつき駆動になっており
部分的な根拠はあるが企業戦略や部門戦略との連携が取れず
不和を生んでいた。

4 対策

Countermeasure

テスト自体の問題

★100点からのスタートはあきらめる

- ✓ 自動と手動で切り分けて、スモールスタート
- ✓ 出来ていないところを常に「意識」しておき、リグレッションでちゃんと手でテストする
- ✓ 自動化と手動テストのコスパで、メリットが有ると思えば自動化するだけ。

★テスト設計と手法の浸透

- ✓ チームでそれぞれのテストに対し、手法や担保するものを意識統一する

★CI/CD文化を作る

- ✓ テストがしやすい環境を自分たちで工夫し作り出す

自動テストは過程である

★ CI/CDの整理もちゃんとする

- ✓ 自動テストはあくまでCDのための一つの手段と意識し、コストをかけすぎない
- ✓ ゴールの見えていない中途半端なCI/CD導入はかえって毒になる
- ✓ 開発ワークフローをマクロな視点で整理する
- ✓ 何を届けないと行けない、何処が人/自動、我々は何を監視する必要が有るのか？
- ✓ 整理できていれば部分改善をしていき徐々に自動化するというロードマップを作れる。
- ✓ チームでコントロールする、放置やり逃げダメ絶対

Product Backlog

ここが割と難題だった・・・

まず大きな課題として、チームは作ったアプリケーションは理解しているが、企業として提供するべきサービスの形をイメージできていない。

Inspect And Adapt

Agileで蓄積されてきた
ノウハウを駆使し
サービスを改善していく

Customer Journey Map

✓利用してくれるユーザーの体験を整理できる
✓顧客に必要なソリューションを探せる

メンバー全員が投資効果を意識するようになり作る物に対し、
不安が生まれるようになってしまった・・・

User Experience

✓アルバイト駆動開発

ペルソナは良いけど、限界もやっぱりある
顧客といっしょに体験することが本当大事

メンバーが投資効果を意識し「やらない」という決断を行えるように

Alignment Maps

✓戦略レベルのアプローチ

良いアプリケーションを作るのではなく、
良いソリューションを作ると言う観点を養える

企業や部門戦略を意識し汲み取る

エンタープライズはトップダウンアプローチがやはり重要

Technology Team

Light Side

Dark Side

Technology Team

Light Side

- ✓ 作成するソリューションへ適用する課題への素早い見積もりと開発
- ✓ チームが責任を持てる技術を選ぶことで、技術課題を透明にできる
- ✓ 技術課題をコントロール可能にする事で業務課題に専念できる
- ✓ 精度の高い時間見積もりが可能に

Dark Side

Technology Team

Light Side

- ✓ 責任を持てない技術（適当に選んだOSS, ベンダ製品など）の見積もりが全くわからない・・・
- ✓ ストーリーポイントが見積もりづらい
- ✓ 稼働が一方のテクノロジチームに偏る事がある

Dark Side

5

まとめ

Summary

Alignment map
And
Customer journey map

Alignment map と Customer journey map の
シナジーが素晴らしい

- ✓ 作戦レベルの整理ではなく、戦略と戦術レベルの整理できる
- ✓ チームが戦略戦術を考える必要は無いが、
  しっておけば最適な作戦が立てれる
- ✓ 必要に応じて戦略/戦術に意見を言ってみるのも大事

Enterprise Engineering

我々はソリューションを作る事が
目的である事を忘れない

- ✓ 腕試し(自分の技術)で最適なアプリケーションを作りたいよね？でもそうじゃない
- ✓ Storyの消化ばかりに注力しない、alignmentからちゃんと見直す
- ✓ 目的達成の為の手段を見直すというフェーズが凄い大事
- ✓ enterprise servieの一部であるsolutionである

Build Vs. Buy

されど目的達成の手段には責任を持つ事が凄い大事

- ✓ buildならテクノロジーへの透過性を維持して行く事がAgileに繋がった
- ✓ buyで補うのであれば、そこへの継続的アジリティーをどう維持していくか、Agileをあきらめると言う選択肢をとる決断を迫られた
- ✓ 目的を達成できるのならどっちでも問題ない。
  その判断した結果、何をトレードオフしているのか理解して
  Build Vs. Buyを選ぶ
- ✓ 技術特化チームを作る事は、このbuildの選択肢を広く深く持つ事が出来て継続的なアジリティーを維持できた

6

これから

Future

Solution And Team

新規ユーザーの申し込み、
もっと短く!

クラウドリソースを
より安易に用意できて
管理しやすく！

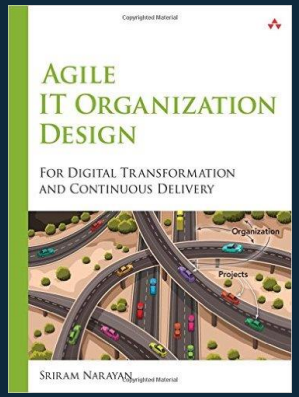
Solution Future

Team Future

サービスサイエンスの適用
より精度の高いDiscoveryを
探せるように

テクノロジチームの
さらなる強化

参考文献



Agile IT Organization Design

Your how-to guide for developing in a multi-platform world.

http://www.api-first.com

Thanks!

酒巻 瑞穂

グロースエクスパートナーズ（株）所属

Frontend Engineer

I am here because I love to programing.
https://github.com/MSakamaki