

組織と個人が内発的動機で継続的に 成長し続けるための実践手法

2020/10/14
SPI JAPAN 2020
小島 優介

自己紹介

名前：小島 優介

所属：デンソークリエイト

業務：

2005年から15年間ツール開発業務を担当

現在は、開発チームのエンジニアリングマネージャーで
毎日メンバーと一緒にプログラミング

Twitter：@kojimadev

<https://twitter.com/kojimadev>

発表概要

プロダクトを開発するために、組織と個人の継続的な成長は重要ですが、なかなか難しい課題だと思います

本発表では、思いついたことを色々やってみるというスタンスで1年以上の期間をかけて成長し続けるチームになった方法を紹介します

- **改善方針**
- 開発プロセスの改善
- 育成方法の改善
- まとめ

当初(1年前)のプロジェクト概要

自社製品としてリリースしているWindows用のツール(下図)を
継続的に機能追加および改善する業務

既存のソースコードは数十万行(言語はC#)

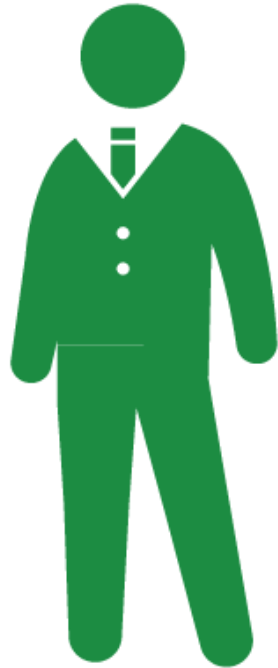


当初(1年前)の開発チーム

中堅1名(私)、若手2名、新人1名(予定)

開発チームのメンバーを一新してから半年経過した状態
(つまり全員このプロジェクトでの経験は浅い)

(新人1名が
後に配属予定)



中堅



3年目



2年目



新人

プロジェクトの見通し

3年後までに様々な機能を実現するロードマップあり

当時(1年前)の生産性では

3年後にロードマップを半分も実現できなさそう



生産性を上げるために、生産性に影響する以下の観点进行分析

1. 開発プロセス
2. 育成方法

1. 当時の開発プロセス

良い点

- 各工程でやるべきことが明記されており、それが守られている
- スクラムで優先度の高い機能から開発してリリースしている

課題

- 開発のやり方が**古いまま(新しいことを試さない)**
- メンバー間の**コミュニケーションが少ない**

※本発表での「開発プロセス」は、開発のやり方全般という広義の意味

2. 当時の育成方法

良い点

- 新人が戦力になるために必要な学習カリキュラムが整備されており活用できている

課題

- **2年目以降の成長**を加速させる仕組みがない
- **個人の仕事の進め方**の改善は個人頼み

改善方針の検討

観点	課題
開発プロセス	- 開発のやり方が古いまま(新しいことを試さない) - メンバー間のコミュニケーションが少ない
育成方法	2年目以降の成長を加速させる仕組みがない 個人の仕事の進め方の改善は個人頼み



短期間で生産性を何倍にも向上させることは困難

短期での課題解決でなく、3年後を見据えて

生産性が**継続成長するような改善方針**を検討

1. 開発プロセスの改善方針

課題：開発のやり方が**古いまま(新しいことを試さない)**



継続成長するには新しいことをどんどん取り入れていくことが必要

改善方針：**新しいものを試行する風土**を作る
(もの＝技術・ツール・プラクティス)

課題：メンバー間の**コミュニケーションが少ない**



継続成長するにはメンバー同士の知見を共有していくことが必要

改善方針：**チームのコラボレーションを促進**させる

2. 育成方法の改善方針

課題： **2年目以降の成長**を加速させる仕組みがない



継続成長するためには継続的にアウトプットすることが有効と言われているが
それは自分の内なる欲求により動機づけられた方が効果が高い

改善方針： **内発的動機づけ**で成長を加速させる

課題： **個人の仕事の進め方**の改善は個人頼み



継続的に改善するためには、仕組みが必要

改善方針： **仕事の進め方を改善する仕組み**を作る

継続成長するための改善方針(以降で施策を紹介)

観点	改善方針	施策	効果
開発プロセス	新しいものを試行する風土を作る		
	チームのコラボレーションを促進させる		
育成方法	内発的動機づけで成長を加速させる		
	仕事の進め方を改善する仕組みを作る		

本発表は、赤枠部分に絞って紹介

(すべての施策の詳細をまとめた資料を最終ページで紹介)

- 改善方針
- **開発プロセスの改善**
- 育成方法の改善
- まとめ

継続成長するための施策一覧

観点	改善方針	施策	効果
開発 プロセス	新しいものを試行 する風土を作る		
	チームのコラボレー ションを促進させる		
育成方法	内発的動機づけで 成長を加速させる		
	仕事の進め方を 改善する仕組みを 作る		

新しいものを試行する風土を作る

以下の施策を1年以上続ければ
きっと風土ができると仮定して実施

1

積極的に試行する価値観の共有

2

とにかく試行して実績作り

3

試行件数の見える化

1. 積極的に試行する価値観の共有

1. 積極的に試行する価値観の共有
2. とにかく試行して実績作り
3. 試行件数の見える化

チーム全員でこれからどうなりたいか議論
「世の中の良いプラクティスやツールを全然知らない」
「同じやり方で変わらないのはつまらない」



**とにかく新しい技術・ツール・プラクティスを
試行してみたらどうだろう**

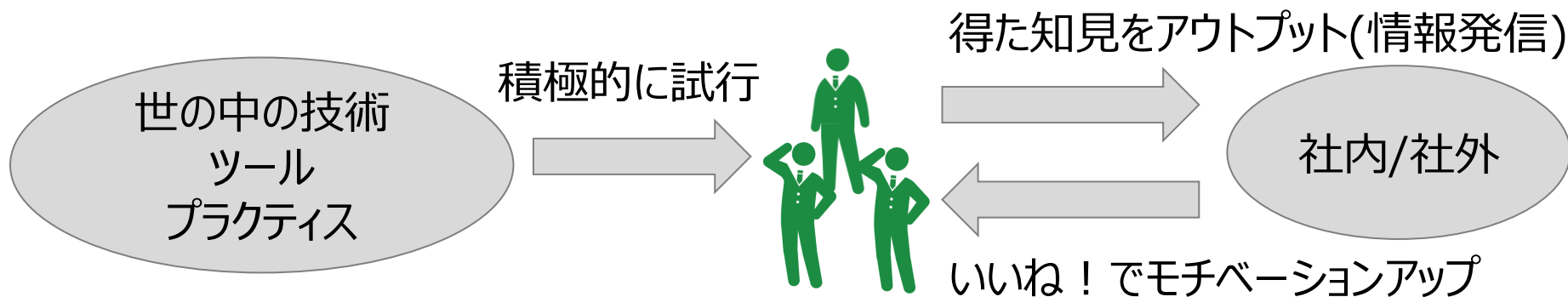
「それは面白そう」
「自分たちも成長できるし楽しそう」



良い感じのスローガンをつくろう

ハピネスチームビルディング

1. 積極的に試行する価値観の共有
2. とにかく試行して実績作り
3. 試行件数の見える化



「試行→アウトプット→いいね」の無限ループで成長できて楽しそう

この活動を**ハピネスチームビルディング**と名付ける
(この用語は本発表の核となるため、この後に何度も使用)

新しいものの試行が最重要な価値観の共有

2. とにかく試行して実績作り

1. 積極的に試行する価値観の共有
2. とにかく試行して実績作り
3. 試行件数の見える化

思い付きでも、効果見込みが低くても、とにかく皆で提案

提案自体を善として試行

試行して分かったことを社内外に発信

試行した例（1年で60件以上）

- ペアプログラミング(以降、ペアプロ)
- モブプログラミング(以降、モブプロ)
- 分報
- インセプションデッキの作成
- ドメインビジョン声明文の作成
- DX Criteriaでの評価
- Lean Coffeeによる振り返り
- Friendlyを用いたテスト自動化

(発信の例)

毎朝15分の勉強会で若手の行動が驚くほど改善した話

Java C# オブジェクト指向 初心者 新人プログラマ応援

1. はじめに

本稿は、私のプロジェクト(ベテラン1人、開発経験が半年未満の若手2名)で4ヶ月前から実施している「アウトプット勉強会」の実施方法を紹介します(投稿日は2019/7/2です)。

私は過去に新人8人くらいの育成に携わりましたが、この勉強会を実施してから、若手が驚くほど成長しました。なぜ今までこれをやらなかったのかと後悔しています。

この「アウトプット勉強会」の形式は、「書籍編」と「設計・実装編」の2種類があります。本稿では「書籍編」の方法を説明します。「設計・実装編」については、以下の記事を参照ください。

[毎朝15分の勉強会で若手の設計力がメキメキアップした話](#)

2. 前提

一般的に、書籍は一人で黙読するよりも、何かしらのアウトプットを伴った方が学習の質・効率が高いと言われています。例えば、書籍でインプットした情報を、自分の言葉に変換して人に説明した方が、一人で黙読するよりも学習の質・効率は上がります。

以下、参考サイトです。

[仕事ができる人がやっている「圧倒的に結果が出るアウトプット術」](#)

[インプットとアウトプットの本当の意味 | 飛躍的に創造性を高める方法](#)

[勉強の質・効率を上げる！成果を出すためのインプットとアウトプット法](#)

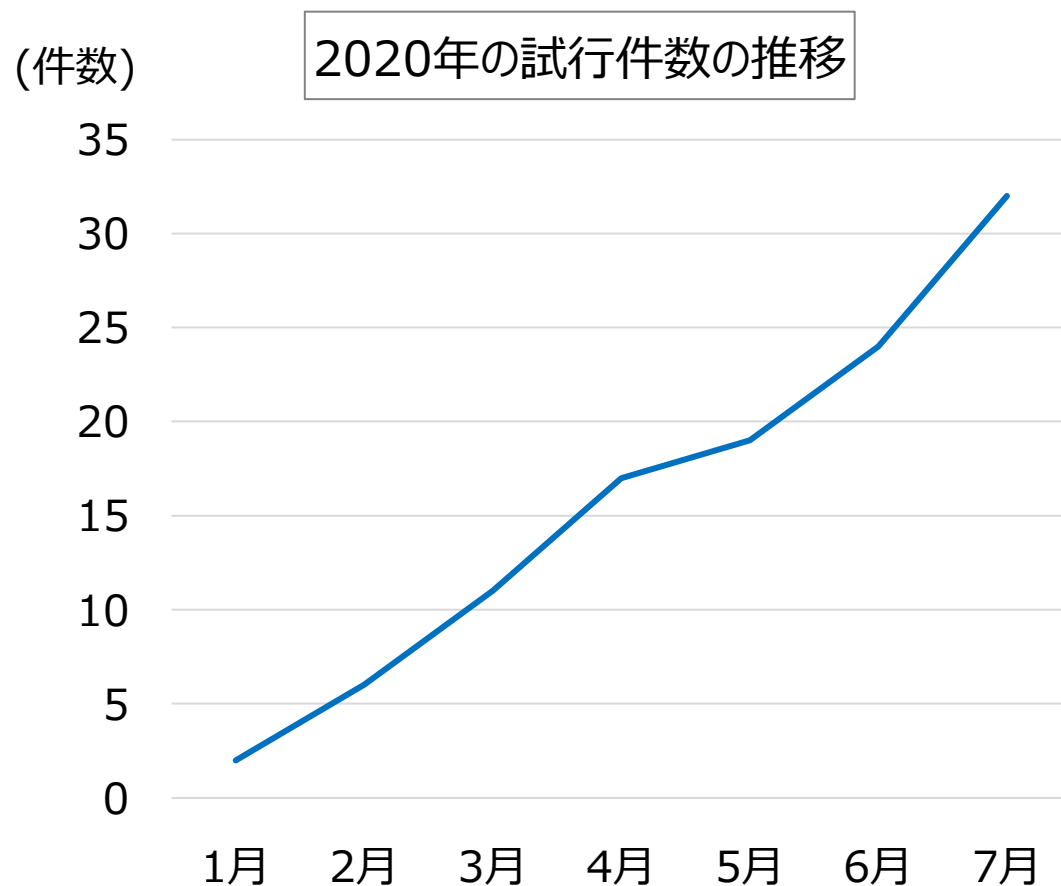
3. 勉強会の実施方法

3. 試行件数の見える化

1. 積極的に試行する価値観の共有
2. とにかく試行して実績作り
3. 試行件数の見える化

新しいものを**試行した件数**を**見える化**

今月も試行できたねと喜び合う



件数の増加が見えると
達成感を感じやすい

試行する風土を作る施策の効果(1年後)

施策	Before	After
積極的に試行する 価値観を共有	現状のやり方を変える 必要性を感じない	ハピネスチームビルディング の スローガンのもとで 毎月新しいものを試行して 変わり続けることが当たり前 (毎月数件を試行)
とにかく試行して 実績作り		
試行件数の 見える化		

試行した結果、効果があったものは継続し、

効果がないものはやめる

(とにかく試してみることが重要な価値観なので、効果なくてやめることは気にしない)

以降の施策も、ハピネスチームビルディングのスローガンをもとに
皆で提案・試行したもの

継続成長するための施策一覧

観点	改善方針	施策	効果
開発 プロセス	新しいものを試行 する風土を作る	・積極的に試行する価値観を共有 ・とにかく試行して実績作り ・試行件数の見える化	・毎月新しいものを試行して変わり続けることが当たり前
	チームのコラボレー ションを促進させる		
育成方法	内発的動機づけで 成長を加速させる		
	仕事の進め方を 改善する仕組みを 作る		

継続成長するための施策一覧

観点	改善方針	施策	効果
開発プロセス	新しいものを試行する風土を作る	・積極的に試行する価値観を共有 ・とにかく試行して実績作り ・試行件数の見える化	・毎月新しいものを試行して変わり続けることが当たり前
	チームのコラボレーションを促進させる	・ペアプロ/モブプロで楽しむ ・コーチングプログラミング ・ポジティブフィードバック	・心理的安全性が高まり、各自が自分の考えを話す
育成方法	内発的動機づけで成長を加速させる		
	仕事の進め方を改善する仕組みを作る		

本発表では詳細を割愛
詳細は最終ページの
参考資料を参照

- 改善方針
- 開発プロセスの改善
- **育成方法の改善**
- まとめ

継続成長するための施策一覧

観点	改善方針	施策	効果
開発プロセス	新しいものを試行する風土を作る	・積極的に試行する価値観を共有 ・とにかく試行して実績作り ・試行件数の見える化	・毎月新しいものを試行して変わり続けることが当たり前
	チームのコラボレーションを促進させる	・ペアプロ/モブプロで楽しむ ・コーチングプログラミング ・ポジティブフィードバック	・心理的安全性が高まり、各自が自分の考えを話す
育成方法	内発的動機づけで成長を加速させる		
	仕事の進め方を改善する仕組みを作る		

まずアウトプットの経験と習慣づけ

理想の状態

学んだことをアウトプットし続けて成長していく

(アウトプット = 人に説明する、技術記事を書く、など)

当初(1年前)の開発チームの状態

自分も含めて全員、技術記事を書いたこともなく

自主的にアウトプットを継続できない

できない理由の仮説を立てる

- アウトプットで**成長を実感した経験**がないから
- 学んだことを**アウトプットする習慣**がないから



まずは、経験と習慣づけに取り組む(半年程度)

経験と習慣づけのための施策一覧

ハピネスチームビルディングのスローガンにより、思いついたものから、どんどん試行

- 毎日
 - 毎朝15分のアウトプット勉強会 – 書籍編
 - 毎朝15分のアウトプット勉強会 – 設計・実装編
- 毎月
 - 毎月の技術記事投稿

毎朝15分のアウトプット勉強会 – 書籍編

学んだことは人に説明(アウトプット)した方が身につくため
事前に全員が書籍を30ページ程度 読んできた上で
その重要ポイントを**各自が自分の言葉で説明**
(毎朝、学んだことをアウトプットする習慣づけ)



私はこの部分が
重要だと思いました
なぜかというと・・・

オフラインで実施時の様子
(現在はZoomで実施)

毎朝15分のアウトプット勉強会 – 設計・実装編

設計・実装の経験が少ないうちは
初歩的な問題を含んだ設計書やソースコードをよく作成する
(例)責務が分かれていない設計、正常系が動作しない実装



設計・実装の経験を効率的に積むために
事前に、デザインパターンを用いた自分なりのサンプルプログラムの
クラス図とコードを書いてきてもらう
それを15分の勉強会の中でレビューする

Gofの23パターンすべて実施で完了
3000行程度の設計・実装の経験値が得られる

毎月の技術記事投稿

業務時間内に学習したことをアウトプットして成長する方が
長期的には生産性が高まると判断し、技術記事を書くなどの
学習工数を踏まえて業務を計画するように調整

自分も記事投稿の初心者だったが、頑張ってメンバーをサポート

- 記事のネタを一緒に考える
- 下書きの記事を見せてもらって助言
- 記事の質は求めない
- 投稿したらポジティブフィードバック



自分も含めて全員が

毎月2件、技術記事を投稿できるようになる

経験と習慣づけのための施策の効果(半年経過後)

施策	Before	After
毎朝15分の アウトプット 勉強会 - 書籍編	自発的に行動が改善されることは 少ない	書籍に書いてある内容が しっかり身につく、 行動が改善
毎朝15分の アウトプット 勉強会 - 設計・実装編	初歩的な問題多数	初歩的な問題が ほぼなくなる
毎月の 技術記事投稿	実装方法を調べる際に 仕組みを理解していない (コピペで実装することがある)	記事投稿で、どういう仕組みなのか 考える習慣が付いたため、 新しい技術を学ぶ時に 仕組みを理解しようとする

- アウトプットで成長を実感する経験
 - アウトプットの習慣づけ
- 上記を達成し、次のステップへ

内発的動機づけの施策(チームの若手3人が対象)

現代の職場では モチベーション 3.0 が必要と言われている(※1)

以下の要素による内発的動機づけが重要

- 自律性：自ら方向を決定したい
- 熟達：価値があることを上達させたい
- 目的：社会に貢献したい



モチベーション 3.0 を促進する施策を実施 (右に関連要素を記載)

(ハピネスチームビルディングのスローガンにより、思いついたものから、どんどん試行)

① 要求からテストまで全てやる	自律性		
② 主体的に学習できる環境を作る	自律性	熟達	
③ 個人で企画したアプリ開発を勧める	自律性	熟達	目的
④ 社外イベントでの発表を勧める	自律性		目的

(※1)詳細は書籍「モチベーション 3.0」を参照

① 要求からテストまで全てやる

- ① 要求からテストまで全てやる
- ② 主体的に学習できる環境を作る
- ③ 個人で企画したアプリ開発を勧める
- ④ 社外イベントでの発表を勧める

裁量を与えることで、**自律性を発揮しやすくする**



- 要求、外部仕様、設計、実装、テスト
すべての工程を全員が担当
(新人でも、仕様や設計に提案できる)
- 各タスクのゴールは事前に決めるが、やり方は任せる
(自分なりに工夫できる余地が多い)

②主体的に学習できる環境を作る

- ①要求からテストまで全てやる
- ②主体的に学習できる環境を作る
- ③個人で企画したアプリ開発を勧める
- ④社外イベントでの発表を勧める

学習は「やらされる」状態では捗らない

興味がある技術に対して**主体的に学習する方が効率的**

- 自由な**学習を毎日1H確保**を勧める
(プロジェクトの進捗にかかわらず業務時間内で確保)
- 学んだことを**随時アウトプットする場を作る**
 - Slackに学びをアウトプットするチャンネルを作成
 - 朝会で学んだことを皆に話す時間を作る
 - 各自の学習内容をチームで共有でき、お互いに刺激し合う

<参考>

脳科学者の茂木健一郎氏は、書籍「脳を最高に活かせる人の朝時間」にてGoogleの20%ルールにふれた後、以下のように述べている。

「たとえば仕事の時間を削ってでも、心にゆとりの時間を与えることで、仕事のクオリティは格段にアップします」

③個人で企画したアプリ開発を勧める

- ①要求からテストまで全てやる
- ②主体的に学習できる環境を作る
- ③個人で企画したアプリ開発を勧める
- ④社外イベントでの発表を勧める

「作りたいものを作るために必要な知識を調べる」ことは楽しい

個人開発を趣味にできるとプライベートでも成長が加速する
企画から考えることでアプリ開発者としての視座が高まる



1. 何が作りたいか聞き出して、MVP(Minimum Viable Product)と一緒に考える
2. 作りたいアプリの技術選定と公開手順を教える
3. 週に1時間「個人開発タイム」
4. つまった所はペアプロ
5. MVPが出来上がったら皆に見てもらう場を設けて褒める

新しい技術を学びながら、自分で企画して作る楽しさを経験

④ 社外イベントでの発表を勧める

- ① 要求からテストまで全てやる
- ② 主体的に学習できる環境を作る
- ③ 個人で企画したアプリ開発を勧める
- ④ 社外イベントでの発表を勧める

顔が見える人から直接いいねと言ってもらう嬉しさを経験できる
社外のエンジニアから良い刺激をもらえる



初めての社外発表は、敷居が高くて踏み出せない人が多いため
敷居を下げるために、**発表初心者向けコミュニティを立ち上げる**

Serverless LT初心者向け

発表初心者が集まるLT大会を毎月開催
若手3人はそのLT大会で発表を経験

コミュニティの詳細は以下のURL参照 (どなたでも、ご参加、大歓迎)
<https://serverlesslt.connpass.com/>

内発的動機づけの施策の効果(若手3人が対象)

施策	Before	After
要求からテストまで全てやる	与えられたタスクをやる	自分の考えや 自分がやりたいことを主張する
主体的に学習できる環境を作る	業務で必要でない限り、新しい技術を調べない	主体的にキャッチアップ する
個人で企画したアプリ開発を勧める	プライベートでの学習は「頑張るもの」という意識	若手3人中1人は プライベートでも個人開発 を続けて楽しんでいる
社外イベントでの発表を勧める	社外のエンジニアと自分を比較しようと考えたことがない	社外の同世代のエンジニアから 刺激を受けて もっと成長したい と感じた

これらの施策の結果、

日々の仕事や成長に達成感を感じやすくなった

(この変化は、モチベーション3.0に近づいている兆し)

継続成長するための施策一覧

観点	改善方針	施策	効果
開発プロセス	新しいものを試行する風土を作る	・積極的に試行する価値観を共有 ・とにかく試行して実績作り ・試行件数の見える化	・毎月新しいものを試行して変わり続けることが当たり前
	チームのコラボレーションを促進させる	・ペアプロ/モブプロで楽しむ ・コーチングプログラミング ・ポジティブフィードバック	・心理的安全性が高まり、各自が自分の考えを話す
育成方法	内発的動機づけで成長を加速させる	・毎朝15分のアウトプット勉強会 ・毎月の技術記事投稿 ・要求からテストまで全てやる ・主体的に学習できる環境を作る ・個人で企画したアプリ開発を勧める ・社外イベントでの発表を勧める	・書籍の内容が身に付き行動が改善 ・実装時の初歩的な問題がなくなる ・技術の仕組みを理解しようとする ・自分の考えややりたいことを言う ・技術情報を主体的にキャッチアップ ・社外からの刺激でもっと成長したい ・日々の仕事や成長に達成感
	仕事の進め方を改善する仕組みを作る		

継続成長するための施策一覧

観点	改善方針	施策	効果
開発プロセス	新しいものを試行する風土を作る	・積極的に試行する価値観を共有 ・とにかく試行して実績作り ・試行件数の見える化	・毎月新しいものを試行して変わり続けることが当たり前
	チームのコラボレーションを促進させる	・ペアプロ/モブプロで楽しむ ・コーチングプログラミング ・ポジティブフィードバック	・心理的安全性が高まり、各自が自分の考えを話す
育成方法	内発的動機づけで成長を加速させる	・毎朝15分のアウトプット勉強会 ・毎月の技術記事投稿 ・要求からテストまで全てやる ・主体的に学習できる環境を作る ・個人で企画したアプリ開発を勧める ・社外イベントでの発表を勧める	・書籍の内容が身に付き行動が改善
	仕事の進め方を改善する仕組みを作る	・各自の仕事の進め方見える化 ・日々の指導の観点を絞る	・仕事の進め方の問題は、ほぼ無くなった

本発表では詳細を割愛
詳細は最終ページの
参考資料を参照

- 改善方針
- 開発プロセスの改善
- 育成方法の改善
- **まとめ**

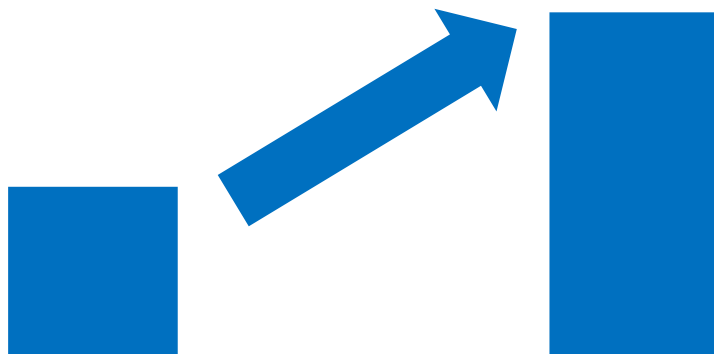
継続成長するための施策一覧

観点	改善方針	施策	効果
開発プロセス	新しいものを試行する風土を作る	・積極的に試行する価値観を共有 ・とにかく試行して実績作り ・試行件数の見える化	・毎月新しいものを試行して変わり続けることが当たり前
	チームのコラボレーションを促進させる	・ペアプロ/モブプロで楽しむ ・コーチングプログラミング ・ポジティブフィードバック	・心理的安全性が高まり、各自が自分の考えを話す
育成方法	内発的動機づけで成長を加速させる	・毎朝15分のアウトプット勉強会 ・毎月の技術記事投稿 ・要求からテストまで全てやる ・主体的に学習できる環境を作る ・個人で企画したアプリ開発を勧める ・社外イベントでの発表を勧める	・書籍の内容が身に付き行動が改善 ・実装時の初歩的な問題がなくなる ・技術の仕組みを理解しようとする ・自分の考えややりたいことを言う ・技術情報を主体的にキャッチアップ ・社外からの刺激でもっと成長したい ・日々の仕事や成長に達成感
	仕事の進め方を改善する仕組みを作る	・各自の仕事の進め方見える化 ・日々の指導の観点を絞る	・仕事の進め方の問題は、ほぼ無くなった

1年前との生産性の比較

工数1人月(160時間)当たりでの新規・変更のコード行数で評価

人月当たりの開発生産性(LOC/人月)



1年前

現在

昨年の**2.04倍**に増加
(具体的な値が公開できず、すみません)

3年間でどこまで
生産性が上がるか楽しみ

コード行数では生産性を正しく計測できないと言われているため
別の方法でも計測し、その計測方法でも生産性は2倍に増加。詳細は以下の記事参照。
「ユーザーにとってどれだけ価値を提供できたかで生産性をお手軽に計測した話」

<https://qiita.com/kojimadev/items/f116e6647c4e9d5fdf42>

当初(1年前)の開発チームの状態

全員、技術記事を書いたこと無し、社外イベント参加経験なし
開発知識は、**業務で教えてもらったことしか知らない**



最後に

チームを改善する方法は色々あります

(問題の原因を分析して対策する、など)

その中で、チームに合った方法を選ぶと良いと思います

私のチームは「もっと良くするために色々やる」という方法で

メンバーが**楽しんで取り組めて継続成長**できました

本発表のエッセンスを読み取って活用していただければ幸いです

本発表で割愛した内容も含めた施策の詳細は以下の記事を参照ください

「1年以上かけて生産性倍増＋成長し続けるチームになった施策を全部公開」

<https://qiita.com/kojimadev/items/4b28f801863cf4e8f0da>