

AI Project Management Flow and Build Trap Review

数十の不確実性の高い機械学習プロジェクトを
自己組織化されたチームで健全かつ最大化されたゴールに向かうための戦術と組織戦略



Delight and Impact the World

DeNAにおけるAI/DS/Analytics組織の前提

- DeNAでは2016年から機械学習を軸とする横断組織を立ち上げ
- AIやデータ活用に関わる人材が集まるデータ本部だけで2022年現在89名。これに実際ソリューション提案を受けるステークホルダーがAIやデータ活用に関わっている
- DeNAは**多角的事業会社であり、機械学習プロジェクトは常に数十個が同時並行**している。これらは事業単位で微妙に異なる文化の組織・戦略を持つステークホルダー(社外とのアライアンスや新しくDeNAグループに加わった多文化的な側面も影響)であるが、その上でPjM,PdMの健全性や品質を高度なレベルで均一化する必要がある
- 一方でチームトポロジーの概念やSpotifyModelが組織戦略として注目される中、DeNAにおける多文化的な組織群に対する横断的なアプローチの中で、これにどう適応していくかが求められていく。エッセンスとしても、チームの受身化・サイロ化を抑制し、**自己組織化**を促進させたい
- 特にDeNAは**受託開発ではない**。事業会社であり、ステークホルダーとも対等なコンセンサスを重視している。よって、受身化は最もこれを戒めなければならない

課題

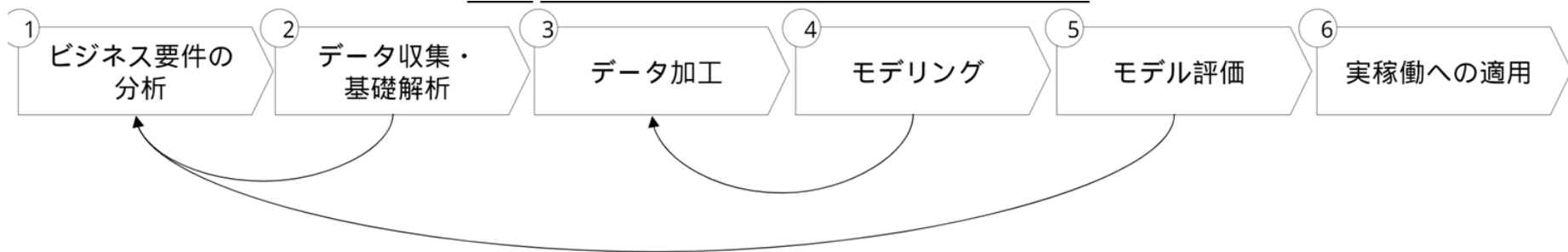
- 機械学習プロジェクトにおいて、アルゴリズムに対するフィージビリティや要求性能を満たすためのPoCは不確実性が高く、一般的なシステム開発とは異なるアプローチが必要となる
- 機械学習プロジェクトが増えていく中で、PoCの終了条件とMVPの境界前提が確立しなままプロジェクト進行するケースが存在する
- PoCが不完全なままMVPを進行した場合、前提条件の破綻とコスト増大リスクを抱えることになる
- MVPが本来実証すべき一貫した本質的な価値提供の創出機会を失う懸念と、不確実性の上昇は避けなければならない
- チームのレベル差にはバラツキがあり、PjM/PdMも例外ではない。ある種のスーパーマンの成功体験のみを生存バイアスとして活用することは依存であって、持続的かつスケラブルな組織を目指さねばならない

アプローチ

- PjM体制の中で、アセスメント、PoC、MVPのフェーズ終了条件/境界を定め、フレームワーク(**AI Project Management Flow**)にすることで個々に目指すべきゴールが明確になり、イテレーション回数を増やし、結果的にプロダクト価値の最大化と案件ごとの品質差異を最小化する
- 合わせて多面的な**ビルドトラップレビュー**を行い、組織長のような管理者ではなく、自律的なチーム同士がプロジェクトのゴール確認・健全性のチェック・品質の均一化・そしてそれらのナレッジトランスファーを行うことで、効率的にチームが提供する機械学習プロダクトのアウトカムを最大化しながら、組織のスケラビリティを高める

課題1:PjM品質の担保

これまでのありがちなMLプロジェクトの流れ



プロジェクトの目的をビジネス要件として棚卸し、その優先順位や定量的閾値等を明らかにします。その目標をデータサイエンス観点での挑戦や課題に翻訳します。

モデル評価の結果、目標の見直しが適宜行われます

顧客より利用可能なデータを収集します。

データで使われる項目の意味や値の幅等を理解します。

データの精度（計測上の精度や欠損有無）を特定します。

サンプルを抽出し、モデリングの方針策定に利用します。

生データをモデリングに活用可能な形式に変換・加工します。

本プロセスは1回限りではなくプロジェクト中何度も発生します。

テーブル・レコード・属性（カラム）の選択と、データ変換、クレンジングを行います。

機械学習やデータマインニングのツールを活用し、目標となる予測値や分類値を算出するためのモデルを生成します。

多くの場合、異なる手法やアルゴリズムのモデルを複数準備し、相対的な評価を行います。

ビジネス要件をベースに、モデルをビジネス要件の言葉＝ビジネス上の目標に照らして、要件を満たすかどうか評価します。

モデルの精度だけでなく堅牢さや実務での運用性等を評価した結果、ビジネス要件の見直しも行います。

モデルを PoC 環境として実行する中で、ライブデータ（生の・最新のデータ）を活用した検証環境に投入します。

ライブデータでの稼働時における品質だけでなく、モデルの最適化やメンテナンス手法についても検証します。

<p>テーマ選定 (課題の深堀)</p>	<p>ヒアリングを繰り返し運用課題などのリサーチ</p>	<ul style="list-style-type: none"> 実際の業務の中で解決した時に大きなインパクトをもたらす領域についてヒアリングを行いながら選定 (意思決定の短縮による売上への貢献や運用精度向上によるコスト削減など)
<p>解法の選定 (アセスメント)</p>	<p>データサイエティストによるアルゴリズムの検討</p>	<ul style="list-style-type: none"> 設定したテーマに対して、どのような手法を用いて解決することができるか実装上の観点や、実運用での利用を想定して複数パターンを検証
<p>解法の検証 (PoC)</p>	<p>選定したアルゴリズムを実際に実行しながら検証</p>	<ul style="list-style-type: none"> 構築したアルゴリズムを実際に現場で利用していただきながら運用上の課題を洗い出し また、実際の運用で求められるユーザインターフェースなどの検討もこの段階で実施
<p>実用化に向けた開発 (本開発)</p>	<p>実運用に即した形でのアルゴリズム実装と運用における課題の潰し込み</p>	<ul style="list-style-type: none"> 実際に使っていただきながら運用上の課題など改善し、システムとしての完成度を向上
<p>運用</p>	<p>アルゴリズムやシステムの保守運用</p>	<ul style="list-style-type: none"> ライセンスングによって保守運用と併せてサービス提供

16

① AI生成技術 品質検証 1stフェーズ

- ・技術検証に向けた各種検討
- ・検証内容の決定
- ・検証に使用するデータの提供

- ・技術検証に向けた各種検討
- ・受領データでのAI技術検証
- ・AI研究開発エンジニア等
リソース調整
- ・技術検証計画提案

デモツール
確認

① AI生成技術 品質検証 2ndフェーズ

- ・品質検証1stを踏まえ
必要に応じて、追加
検証内容を設定

- ・技術検証に向けた
各種検討
- ・受領データでのAI
技術検証
- ・AI研究開発エンジニア等
リソース調整
- ・技術検証計画提案

トライアル

② AI技術開発 フェーズ

- ・AI生成技術品質検証フェーズでの
評価の結果、今後解消すべき課題を
踏まえ、AI技術開発フェーズを計画
- ・両社で合意した、役務・コスト負担・
ビジネススキームにて技術開発

トライアル

トライアル

御社
本格実用
フェーズ

- ・メンテ
- ・Verアップ等

予測分析テーマでの有償コンサル

- ・ 現場視察／現状把握／課題の整理
 - ・ AIを活用した予知保全の方針アドバイス
→データ取得法、解析法、アルゴリズム設計
- ※DSチームから月 n 時間のコンサル

アセスメント

データサイエンティストによるアルゴリズムの検討

PoC1

選定したアルゴリズムを実際に実行しながら検証

PoC2

本開発

実運用に即した形でのアルゴリズム実装と運用における課題の潰し込み

プロジェクトごとに、フェーズ設計と理解にレベル差が存在

- ✓ ビジネス課題をアルゴリズムタスクに落とし込む
- 先方ビジネス課題のヒアリング
 - 渉外 or BizDev + データサイエンティスト or アナリスト
 - ✗ UXデザイナーやエンジニアは随分後から適宜合流
 - → AIやデータを活用すること「のみ」が目的化してしまい、PdMの観点が抜けている
- △ Proof of Concept で検証サイクルを回しながら本番運用化を探る
 - PoCと運用のシステムアーキテクチャは全く異なる可能性が考慮されていない
- ✗ ステークホルダーとその担当者によって各レベルの理解がまちまち
 - → 契約内容、内部コンセンサス、コミット に影響

PoCとMVPの曖昧性

アジャイルに潜むビルドトラップ

それは本当にPoCなのか？

- ✓ AIエンジニア/DSで解決できる最小構成パッケージを探る
 - ✓ PoCでアルゴリズムの評価をして実運用に活かせるか探る
- → 上記から外れやすい落とし穴が沢山ある
 - アセスメントから解法到達までNヶ月、モデル開発も進んできたけど…
 - アルゴリズムだけで解決できるパターンは実は少なく、多くの場合インターフェースが必要
 - 「モデルだけじゃ運用イメージが沸かない」
 - =当然、PoCだけでなく**MVPが必要**
 - PdMからすれば**至極当たり前の感覚**が、AI文脈になった途端近視眼的な「モデルの実現可能性」だけに注力してしまい、プロダクトの本質的文脈を失ってしまう
 - いつのまにかPoCの流れでPdMを介せずにMVP要件をヒアリングしてる
 - 当初から想定・考慮されず、場当たりにリソースを調整
 - = いつのまにか非専門領域のインターフェース実装を任される

そもそもPoC?MVP?

- そもそも文献や人によって理解が**マチマチな概念**だが、この文脈では以下で統一
- **Proof of Concept**
 - 「**概念実証**」の意味で、新しい概念や理論、原理、アイデアの実証を目的とした検証
 - なんの概念？
 - 便宜上、AIプロジェクトの概念では AI/DSモデル開発の**フェージビリティ**とする
 - ※AI文脈を伴わない一般的なシステム開発では、エンドユーザーが利用するUIを含めてPoCと呼ばれるが、ここでは機械学習アルゴリズムの実証に集中で統一するため、モデル開発に注力を行うフェーズをPoCと呼んでいる
 - 基本的に仮説検討した、「**アルゴリズムの実証**」を行う
 - サービス向けのインターフェース開発及びプロトタイプ開発を伴わない
 - 成果物として**レポートベースでプロダクト**をもたない
 - ※開発を効率的にするためや、モデルの品質精度を調整や意思決定するための内部開発者向けツール開発についてはこれに当たらない。

そもそもPoC?MVP?

- **Minimum Viable Products**

- 「**実用最小限の製品**」
- PoCで実装された開発モデルを「**サービスに組み込み検証**」する「**最小プロトタイプ開発**」
- 当然**成果物としてプロダクト開発を伴う**
 - ただし、UIはあってもなくても良い
 - CUI app/GUI app/only docker image/code base 様々なケースがありうる
- PoC自体もMVPを意識した上でモデル開発に注力すべきなので、PoCでモデル開発に時限を区切って注力をし、**MVPでエンドユーザーを意識した最小のプロダクト開発**を行うというのがポイント。勿論PoCのアルゴリズムが実証されても、MVPの体験検証でアセスメント（課題設定）の引き直しからやり直す事もありえる。重要なことは、この単位をいかに最小構成に定義しイテレーションを繰り返すことができるのか、にかかっている。
- 一言で言えば、スクラップアンドビルドで「捨てるものを」作る

アジャイルにおけるPoCからオバケMVP開発に潜む罠

- 時間的・予算的にPdMやUXデザイナー/SEWを入れていない
 - 入れたとしても華美なUIは必要なくてジミ～なWebApp程度で申し訳ない
 - 一旦AIエンジニアがMVP開発(わかってないけど…お!)まで良かれと思ってやってしまう
 - チャレンジそのものは決して悪いことではないが…
 - X MVPとしてのPdMやUXの観点がまるっと抜けてしまう
 - チャレンジは整備した上で行うのが「責任」で、場当たりのにするものではない
- PoCからMVPにシームレスに現有リソースで最小要件化しようとする…
 - 🙄ヒアリングとその場の判断でMVPに必要な要件をなんとなく確定している
 - X その要件が本当に顧客にとって必要なものかフィージビリティがとれていない
 - X アルゴリズム開発が先行し、UXやユースケースの深掘りをはじめ、MVP開発は後手に
 - 途中参加でUXUIのプロをいれても要件をひっくり返せない
 - エンジニアの途中アサインもオンボーディングがおいつかないままウォータフォールに

PoC検証とMVP検証が曖昧なまま進めるとどうなるか？

- まさに「顧客が必要だったもの」に陥る
- 「開発も後半でリリースまで2ヶ月です」なんてこともザラ
- アルゴリズムはソリューションとして、UXや運用組み込みのフィージビリティと品質は？
 - アルゴリズムの開発・検証にコストを使い果たしてしまう
 - アルゴリズム検証期間とMVP検証期間のギャップ
- 時系列的に誰(デザイナー, SWE)をアサインしてもその人にとってウォーターフォールに
- オンボーディング/開発コスト増大
- →最悪PoCから先に進めなくなる懸念も

なにが必要なのか？

ソリューション1:PjM品質の担保

AI Project Management Flow

PoCとMVPとプロダクト開発の境界をはっきりさせる

- (原則的に)PoCとMVPを混ぜない
 - 機械学習のPoCはアルゴリズムの提案と解法レポートに集中する。
 - アルゴリズムのフィジビリティと体験のコアバリューやアウトカムは、見るべき軸が異なる
 - 逆にフィジビリティがないのにPdMを巻き込んだ場合、いつまでかかるかの確証はない
- **インターフェースや実運用という言葉が出てきたら「=MVP」が必要**
 - PoCにある程度光明が見えてくると「**実際使ってみないとわかりませんね**」というケース
 - アルゴリズム検証PoCでMVPの要件をコミットをしてはならない
 - **PoCフェーズを終了しMVPはMVPで体制から「仕切り直す」**
 - 無論、モデルの精度向上などのイテレーションは継続する
 - PdMはPoCより前のアセスメントでユースケースとナラティブの妥当性を詰めておく
 - 検証単位で「体制見直し」「先方とのキックオフ」「振り返りと評価」を行う (後述)
- PoCとMVPの境界、そしてその開発タイミングの見極めとハンドリングが重要
- 前検証で行ったヒアリング内容はドキュメントに残す
- = **PoCとMVP,本開発それぞれのゴールを細分化したコンセンサスが必要**

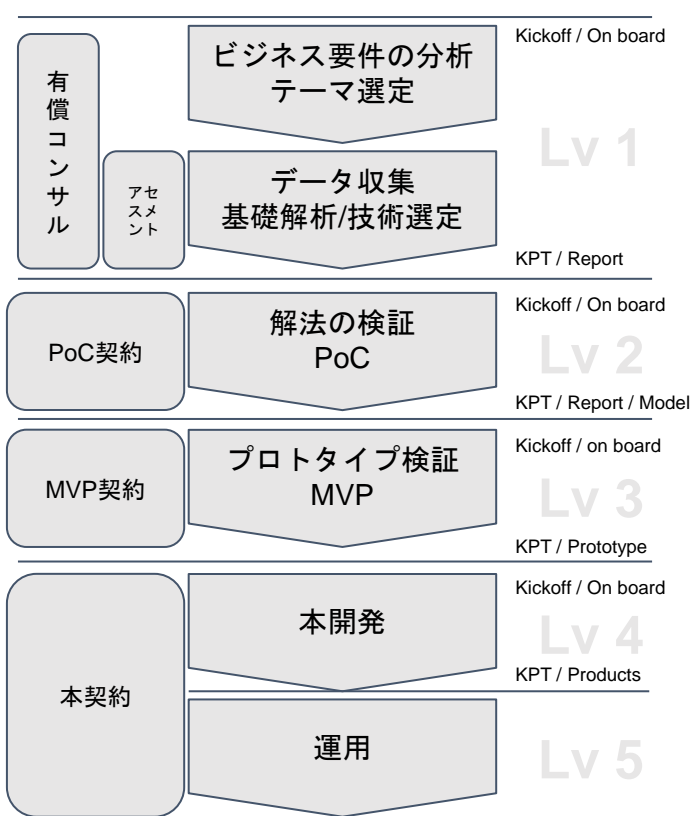
PoCとMVPを明確に分けることで生まれるメリット

- ✓ステークホルダー全員が先を見通す
- ✓検証単位の責任分界点がシンプルになる
 - PoCではアルゴリズム検証、MVPでは実証されたアルゴリズム開発モデルを用いて仮定したユースケースの実証、それぞれのフィージビリティを独立した検証を段階的に行える
 - 並行依存に陥らない
 - → 上記検証サイクルを終えて初めて本プロダクト開発に進む
- ✓コストリスクの分散を検証単位でコンセンサスできる
- ✓なんとなくプロダクト開発しなくなる
 - 突貫的なジャストアイデアでの実装を回避
 - 検証すべき項目に集中した高イテレーション開発
- ✓QAを本プロダクト開発にてまとめられる
- ✓PoCを抜けたDSは、MLOpsエンジニアと連携してMVPと並行してモデル開発のブラッシュアップまたは、冗長性の確保など運用フェーズにシフトできる

PoC to MVP 境界でPjMがすべきこと

- **UIがなければ意思決定ができない場合、それはMVPフェーズへの移行を意味する。**
 - まだモデルに手を入れることがあっても、PoCからMVPフェーズへの移行準備する
 - ※繰り返しになるが、一般的なシステム開発ではPoCを広義にUIを含めるケースは普通にあり得る。不確実性の高いAI開発における場合、PoCにUI検証を混ぜるのではなく検証単位を厳密に分解して行うべきという戦略である
- **MVPフェーズ移行に当たっては、次フェーズ(DevOps)を見据えたアサインが必要になる。これはステークホルダーによってケースバイケースとなり、PoC実施メンバーでは意思決定できない。**
 - 責任分界点(フロントエンド,バックエンド,インフラ) × 担当部門(事業部,技術開発,データ基盤,IT基盤)
- **ステークホルダーを明確にした上で、MVPに進むキックオフを行う**
 - MVPキックオフにあたっては参加メンバーのコンテキストギャップを埋めることを意識
 - 現状のステータス報告とビルドトラップレビュー(後述)
 - アサインリソースや予算などの確認

AIプロジェクトマネジメントフロー



- 課題ヒアリング
- 事業インパクトのあるテーマを選定
- DSメンバーによる有償コンサル

- データ収集方法,データアノテーション
- PoC開発体制確立,アルゴリズム技術選定
- MVP確度検定,MVP開発体制検討

- アジャイルによるアルゴリズム/モデル開発
- 再MVP確度検定,MVP開発体制確定

- MVP要件定義、全体アーキテクチャ設計、セキュリティ要件確定
- アジャイルによるMVP開発 (Prototype Application, CUI/GUI, Web)
- 本運用イメージ確認(SLA/SLO確定)
- 本開発確度検定,本開発開発体制確定

- アルゴリズムと直接関係のない、運用に根ざしたソリューション開発
- 認証認可,可用性等運用品質および製品品質に関わる開発とQA
- サブスクリプション契約の合意

- サービス運用
- 保守
- モデル改善

プロジェクトレベル

- **プロジェクトレベル=前ページの検証単位でのレベルのこと**
 - フェーズという言葉はレベルに内包し、Project Lv 2 PoC Phase 1,2,3… のように使用
 - PoCフェーズ,MVPフェーズは重ねることができる
- **レベル及びフェーズで各プロジェクトをnotionで管理**
- **レベル単位のサイクルで以下を実施**
 - 「体制見直し」
 - 「全体キックオフ」
 - 「振り返りと評価」

プロジェクトレベル単位で管理し、契約/体制見直しを行うべき理由

- フェーズ境界ではデザイン/エンジニアの追加アサインが必要となる可能性が高い
- 全プロジェクトを統一されたレベルとフェーズで管理し、各レベル各フェーズの終了条件を明示
- レベル終了到達前に「体制見直し」を行い、現状ステータスからアサイン調整
- **レベル境界での体制変更によるメリット**
 - ✓ 契約によるコミットメントリスクの最小化
 - ✓ 各領域プロフェッショナル参画の妥当性判断のチェックポイント
 - ✓ ゼロベースでキックオフから各領域プロフェッショナルが参加し、改めて開発体制を作ることで、ウォーターフォールを脱してモデルフェーズビリティを乗り越えてアジャイルなプロジェクトマネジメントの拡張と最適なソリューション提案が可能
 - ✓ アサイン後のアサインメンバーのナレッジギャップを均一化
 - ✓ 振り返りと評価のタイミングを強制的に設けることで、チームとプロダクトの品質を改善するだけでなく、境界と終了条件が明示されることで「ゆるふわ開発」されなくなる

プロジェクトレベル境界におけるマネジメント

プロジェクトレベル境界を適切に区切った契約で進んだ場合、契約と契約の境界で「ゆとり」が生まれる。その境界上におけるチーム開発マネジメントでは、ただ「好きなこと」、「残タスク」をやる、ではなく後のプロジェクトの準備期間としてチーム開発品質を向上させるプライオリティが必要。

つまり、レベル境界とフェーズ境界は、**不安定なスピード重視のみを主眼とした行き当たり開発から脱し、本質的なアウトカムを意識した安定的で保守運用性の高い堅牢なチーム開発に昇華させる大事な期間**といえる。

レベル境界/フェーズ境界でチーム開発としてやるべきこと

- **PoCのスピード先行によってできなかったチーム開発品質を維持する施策実施**
 - **リファクタリング,テスト,ドキュメンテーションなどの運用品質向上**
 - 特に体制見直しに合わせた新メンバーのオンボーディングで効果を発揮
 - 新メンバーが旧メンバーとペアプロやSyncをしながら進めるのが◎
 - **プロジェクトマネジメントツール、手法の見直し**
 - プロジェクト特性・チームの規模・レベル境界に合わせて適切なタスク管理方法をチームで選定。
 - チケットの切り方やリリースフローの見直す
 - **境界で新メンバーが入ることを踏まえた前フェーズからのドキュメント整備も重要**
 - **全体KPTを必ず実施し、チーム開発課題を洗い出す**
 - 前フェーズで課題を残したまま新メンバーを入れて新フェーズに入ることを許さない
 - → マネジメントは次レベルが開始されるまでに解決に動く

MVPの検証項目を細分化し、イテレーションを最大化する

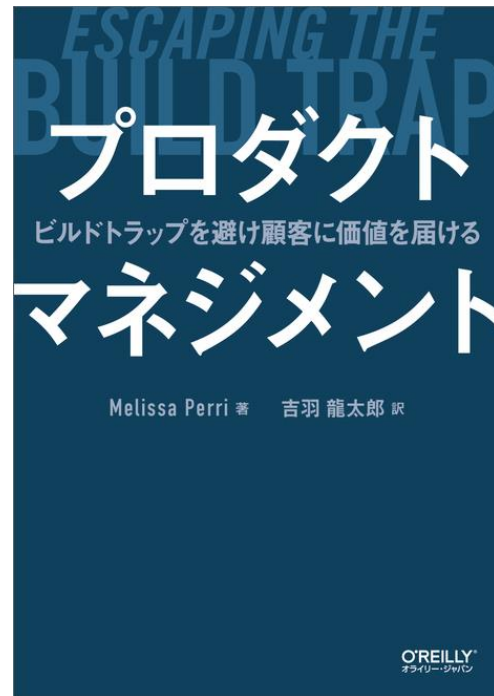
- PoC同様、MVPの検証項目は細分化して実証を目指す
- 必要であればMVP Phase1, MVP Phase 2,MVP Phase N,,のように実証サイクル単位で重ねる
- そのMVPで検証したい・**目指したいゴールを明確にして、イテレーションをとにかく増やす**
- PoCの不確実性と同様に、MVPの不確実性もまた高いものであることを忘れてはならない
- MVPまでこぎつけても、検証次第でPendもありうるという事実をチームが受け入れる
- 万一Pendになっても**結果と成果をチームだけでなく組織全体に持ち帰るまでがプロジェクト**

ソリューション2:PjM+PdM品質の担保

Build Trap Review

ビルドトラップを避け、アウトカムを実現する

- ビルドトラップについての詳細は聖書を参照されたい
- 一般的なシステム開発と比べ、機械学習プロジェクトにおけるビルドトラップは複雑怪奇で並大抵ではなく、AI Project Managementのようなフェーズを明確に分けた戦略が必要。
- AI Project Management単体は万能ではなく、単なるフレームワークに過ぎない。フェーズ単位で適切なゴールを明確に区切り、チームがそこに向かえているのか客観的な判断をするべき
- ビルドトラップレビューを自律チーム同士で行うことで、客観的かつ多面的なインプットを得られ、チームそのもののアウトカムに対する改善行動と組織的な戦略行動・成長促進・スケーラビリティに繋がる



ビルドトラップレビューで行うべき最小の確認

- そのフェーズにおける、ゴールを確認

- もっとも**重要な確認項目**。PjM/PdMがプロジェクトやAIプロダクトに対する短期および中長期のゴールを答えられないようでは話にならない。チームが迷走に陥っていないか、外的要因によってそれが阻害されていないか、そもそもプロジェクトの目的と**アウトカム**が定まっているか。PoCからMVPを経てアウトカムがブレていないか。このゴールの明確化と妥当性を確認するだけでも、時間をかける価値があると言って良い（一口にゴールの確認といっても、ご想像どおり、そこに至るまでのプロセスは千差万別十人十色であり、簡単ではない。**それを考える思考実験のイテレーション、それ自体が有用なのである**)
- 経験深いPjM/PdMのチームと、そうでない成り立てのPjM/PdMとでレビューを行うのが最もトランスファーとして効率がよいが、成り立てのPjM/PdMが自分の言葉と問いを通して噛み砕き、他のプロジェクト/プロダクトのアウトカムを整理すると、自身のプロジェクト/プロダクトにおいてもヒントや気づきを得ることができる
- DeNAの数十あるプロジェクトを逆手にとり、1つのプロジェクトのみを担当するだけではない、他プロジェクトの追体験を自分事で行うことでシャドーイングの効果も実現できる

ゴール確認に次いで、ビルドトラップレビューで確認するもの

- **Role/リソースの短期・中長期妥当性**
 - 現フェーズで非専門領域に対するフィージビリティを門外漢がやっていないか
 - 現フェーズの見通しにより、次フェーズの準備で行うべきタスクが整理/実施されているか
 - 次フェーズのゴール(アウトカム)をメンバーに向けて納得感高く説明できる状態か
- **法務セキュリティ関連**
 - 国内法はもちろん、GDPR,米国連邦/州法,国内/国際特許侵害がないか
 - データセキュリティ、クレンジングの保全
- **AIガバナンス観点**

チームKPTを、組織全体に還元する

- AIプロダクトの不確実性やビルドトラップの多さは、そのまま他のプロジェクトの地雷検知に貢献できる。課題(Problem)は組織全体に透明性高く共有し、Tryの再咀嚼や提案も含めて行う。更にそのチームだけでは解決できない上位マネジメントで意思決定が必要なケースも、直属上長のみで共有するのではなく、組織全体に報告することで、全体最適の妥当性と解決までの時間を最短ルートで攻める。
- 多角事業組織は、一つの成功体験がいくつもの別の事業での成功に横展開できることが最大の強みである。実績を積み上げ、その実績を別の課題・ユースケースに活かすことも、持続的な組織戦略となる
 - 単なるプロジェクトの成功秘話にとどまらず、アルゴリズムの技術的課題から性能改善からプロダクトのアウトカムに繋げる具体施策までチームが持って変えるナレッジに底は無い

まとめ

アプローチのおさらい

- PjM体制の中で、アセスメント、PoC、MVPのフェーズ終了条件/境界を定め、フレームワーク(**AI Project Management Flow**)にすることで個々に目指すべきゴールが明確になり、イテレーション回数を増やし、結果的にプロダクト価値の最大化と案件ごとの品質差異を最小化する
- 合わせて多面的な**ビルドトラップレビュー**を行い、組織長のような管理者ではなく、自律的なチーム同士がプロジェクトのゴール確認・健全性のチェック・品質の均一化・そしてそれらのナレッジトランスファーを行うことで、効率的にチームが提供する機械学習プロダクトのアウトカムを最大化しながら、組織のスクレーラビリティを高める

不確実性の高いAIプロジェクトを全員が快適にドライブする

- AI/データ活用案件で上流から下流まで開発品質担保する難易度は生半可なものではない
- **近視眼的なAIありきの開発では「いいものづくり」はできない**
 - **アウトカムに立ち返ることが最も重要**
 - モデル実証がいつのまにかプロダクト開発になってるなんてことは禁忌
 - 適切なタイミングで適切なプロを入れる仕組みとコンセンサスをする
 - 駄目だったら撤退できる余裕を作る
 - 焦らないで地道にやれる仕組みを作る
 - スピード感とは、「感」ではなく「不確実性を減らし先を見通す力」と、その実行力である

Thank you !

SeeAlso: [AI Project Management Anti Pattern](#)

Author: Yusuke Kamo  [@yurfuwa](#)

- Deputy Business Unit Head / Data Unit
- General Manager / Data Platform Div