



ML Enablement Workshop #2: 応用編

顧客体験改善への応用

久保 隆宏

Developer Relation
Machine Learning

ML Enablement Workshopのプロセス

理解: ベストプラクティスの理解

機械学習の「勝ちパターン」を理解し、活用どころを見極める。



応用: 顧客体験改善への応用

勝ちパターンが成立するプロセスを可視化し解決すべき課題を特定する。



開始: 顧客体験の改善を開始する

機械学習による勝ちパターン成立まで、段階的に顧客体験の改善をする。



「応用」プロセスの対象者とゴール

誰のためのセッション?

機械学習の活用に取り組むプロダクト開発チーム
プロダクトマネージャー、
開発者、
データサイエンティスト
から構成される。

何ができるようになる?

顧客は誰か、なぜ機械学習がその顧客の体験を改善し勝ちパターンをに入れるのか話せるようになる。

何ができるようにはならない?

機械学習アルゴリズムの実装。

疑問：事例通りにしても効果が出ないのはなぜか？

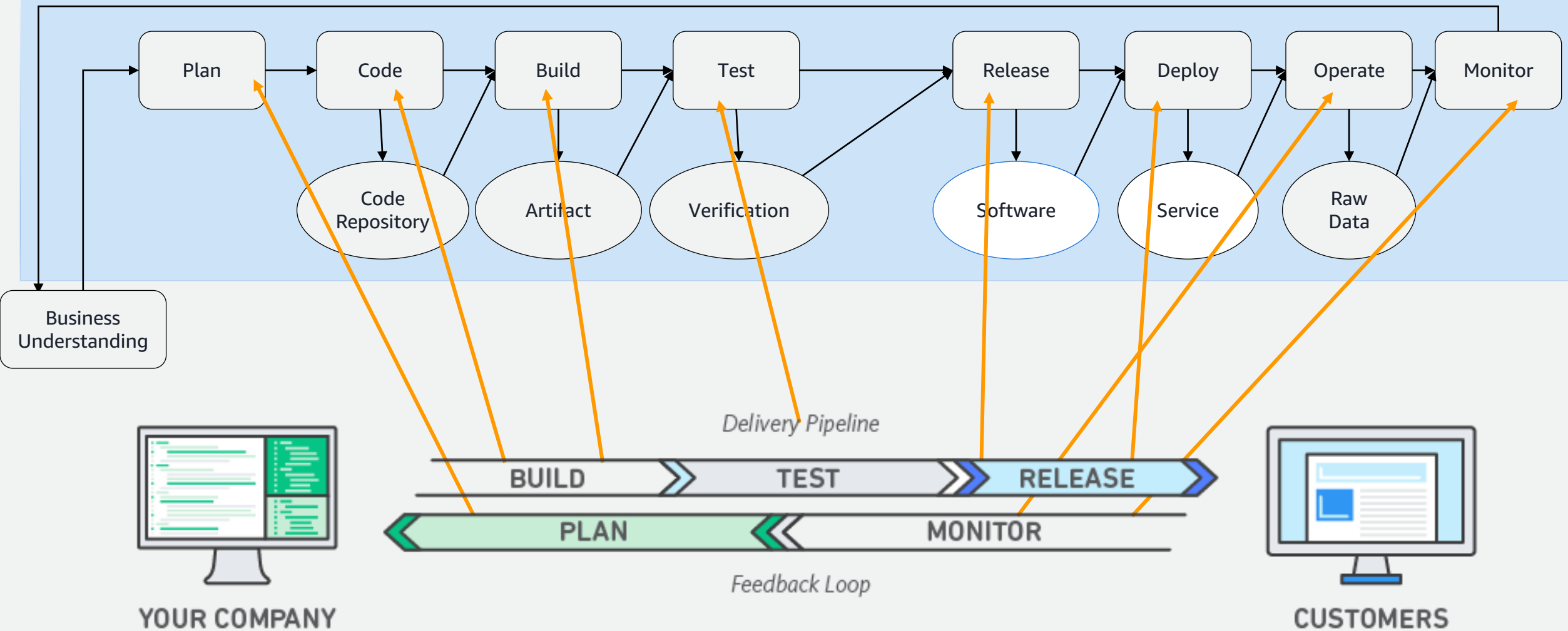
回答：顧客の問題がプロダクトごと異なるから。

詳細は本編で！

1. 機械学習プロジェクトの進行
2. ビジネス課題を定義する
3. ビジネス課題の優先度をつける

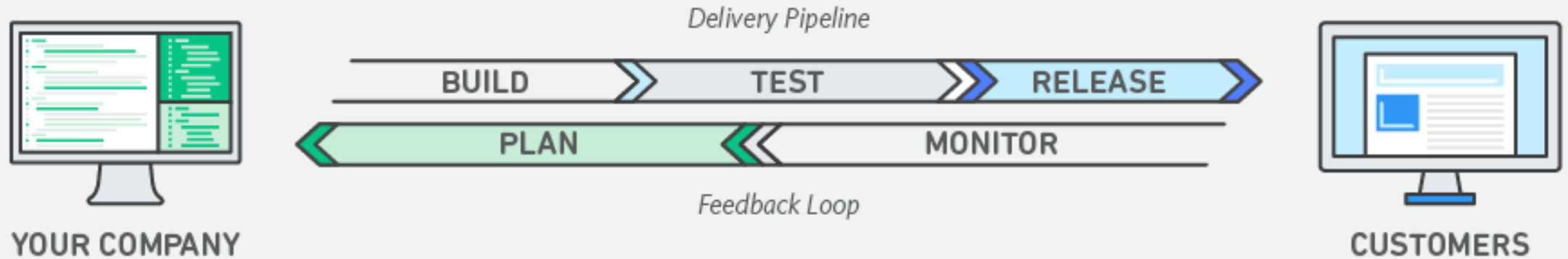
DevOpsに基づく開発プロセスの流れ

DevOps



[AWS “DevOpsとは?”より図を引用](#)

DevOpsのおさらい

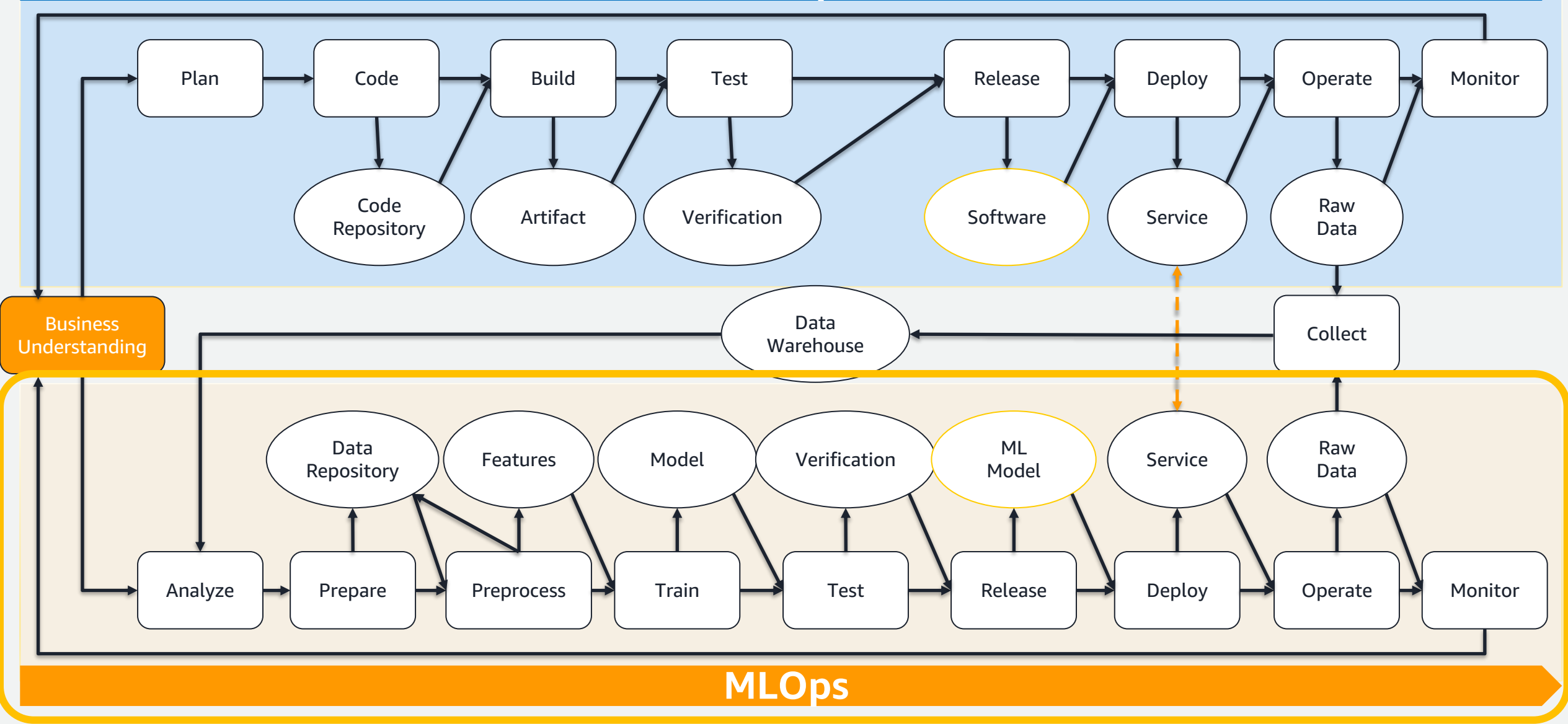


DevOpsは開発(Development)と運用(Operations)から成る造語。
分断されていた開発チームと運用チームを協調させる。

1. ソフトウェア／システムのビジネス価値をより高める
2. ビジネスの価値をより確実かつ迅速にエンドユーザーに届け続ける

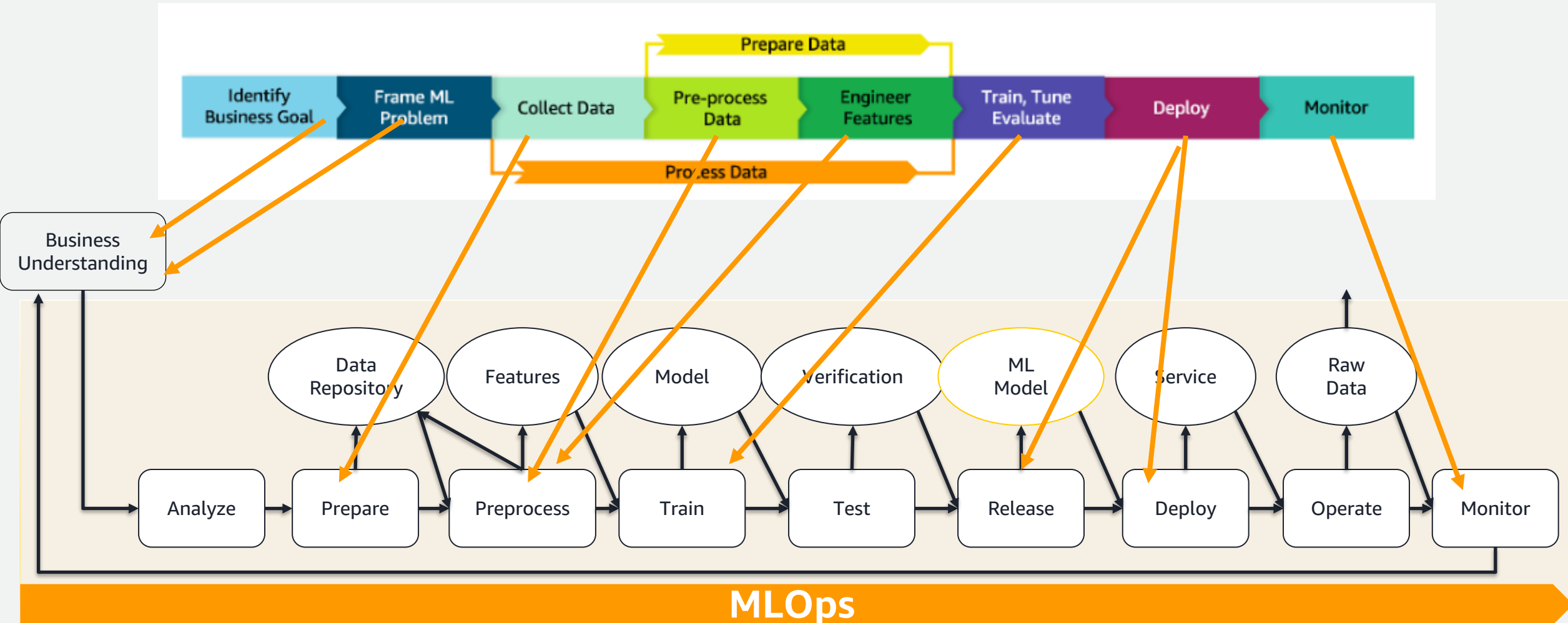
機械学習プロジェクトではモデルを開発するMLOpsが加わる

DevOps



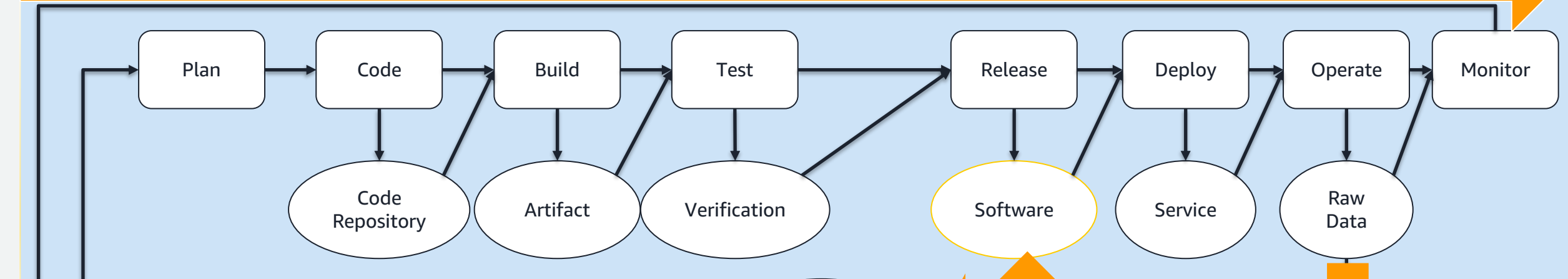
**MLOps は、DevOps の開発手法や組織文化を、
機械学習の開発に取り込んだ開発手法**

MLOpsのプロセスはまだ定まったものがないが、下図はAWSの定義([MLLens](#))を主に参照し定義。

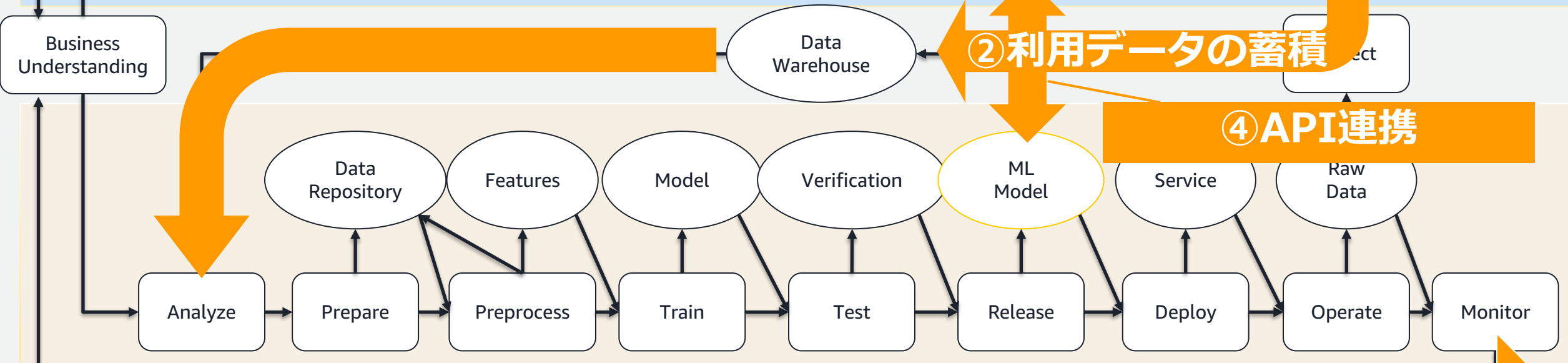


蓄積されたデータから機械学習モデルを構築し、連携させる

① DevOpsによるソフトウェア開発



② 利用データの蓄積

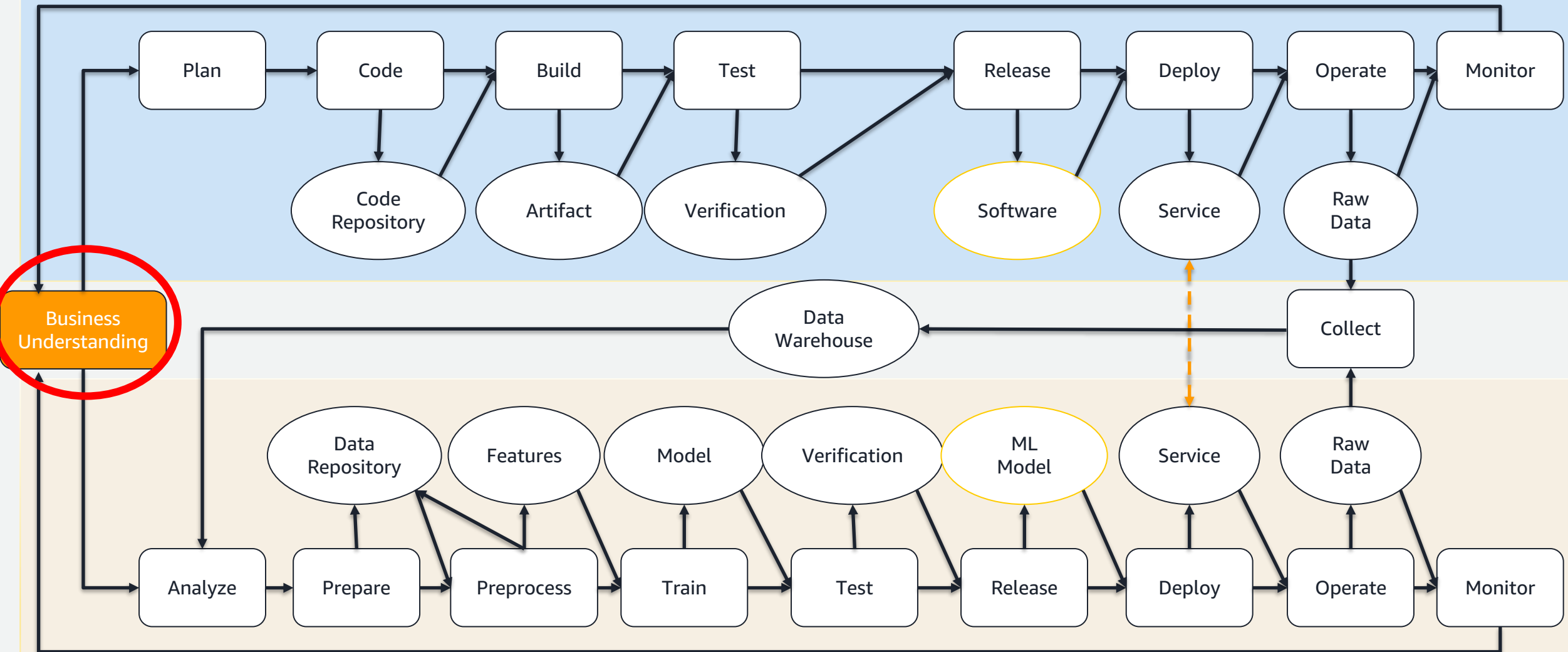


④ API連携

③ MLOpsによる機械学習モデル開発

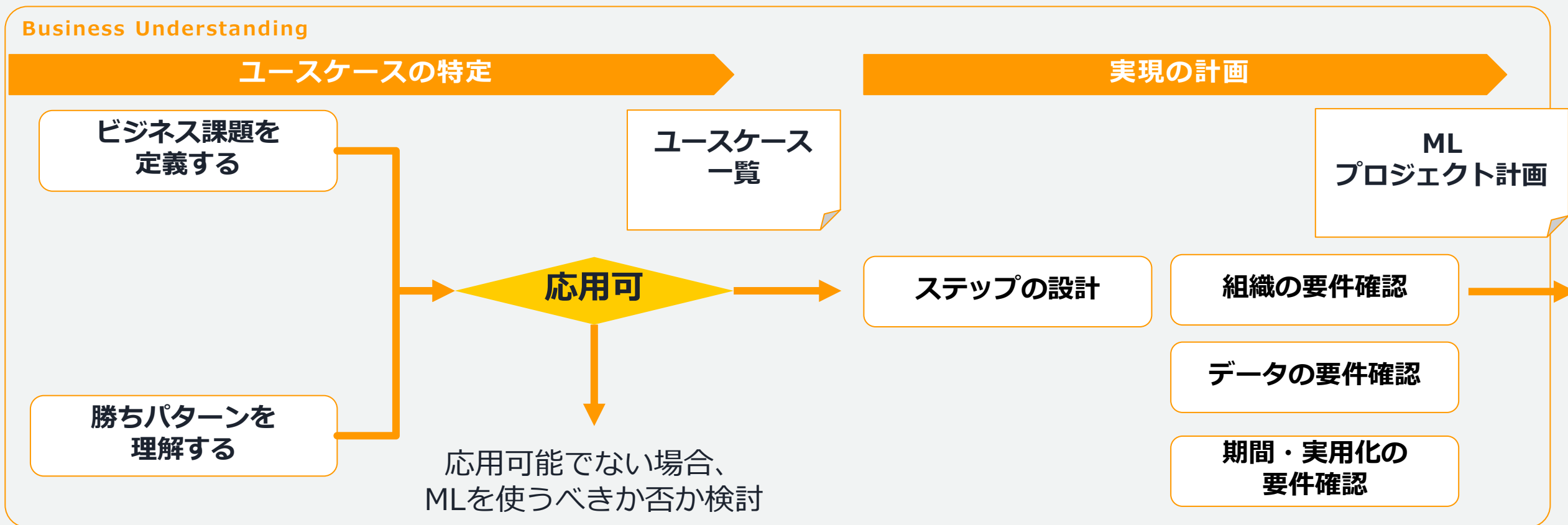
いずれも「ビジネス理解」から開始する。

DevOps



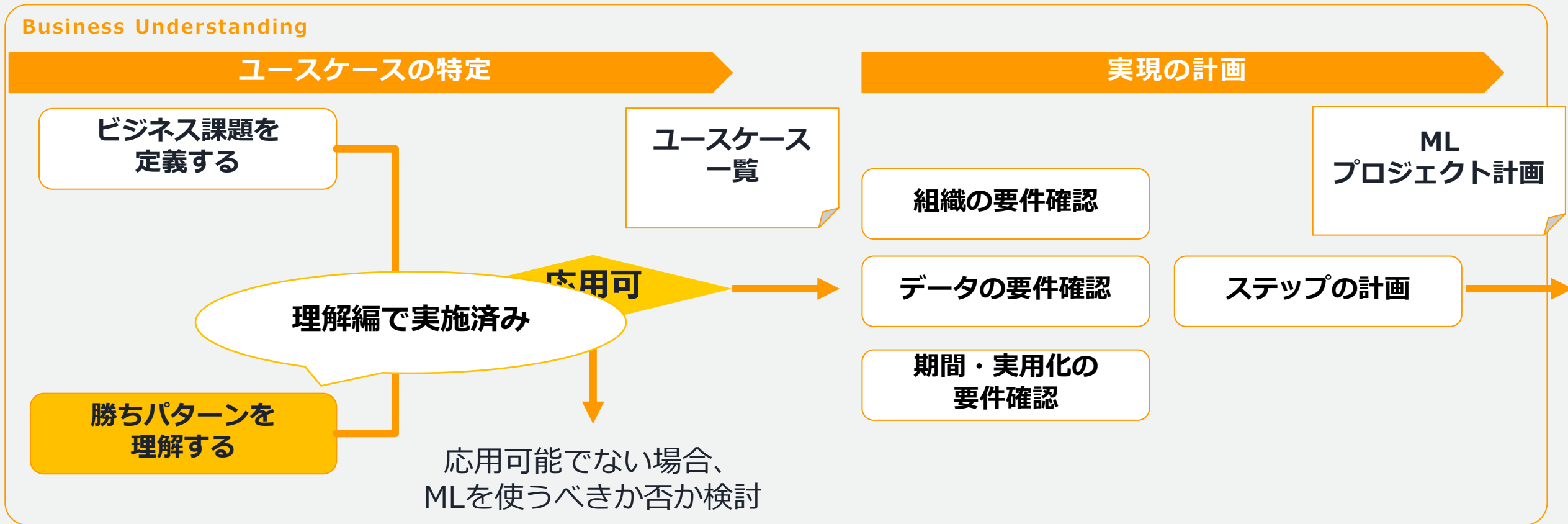
MLOps

ビジネス理解: Business Understanding



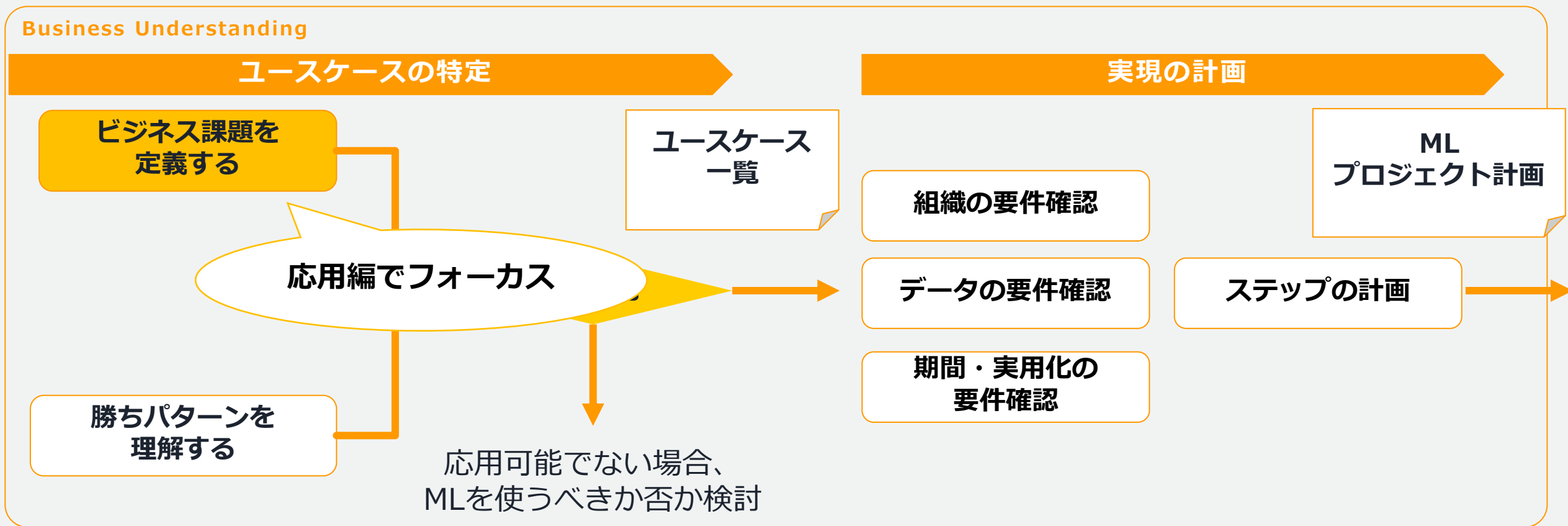
数学で公式に当てはめると簡単に問題が解けるように、勝ちパターンで解決したい課題が解けると簡単にビジネス成果を上げられる。

ビジネス理解: Business Understanding



数学で公式に当てはめると簡単に問題が解けるように、勝ちパターンで解決したい課題が解けると簡単にビジネス成果を上げられる。

ビジネス理解: Business Understanding



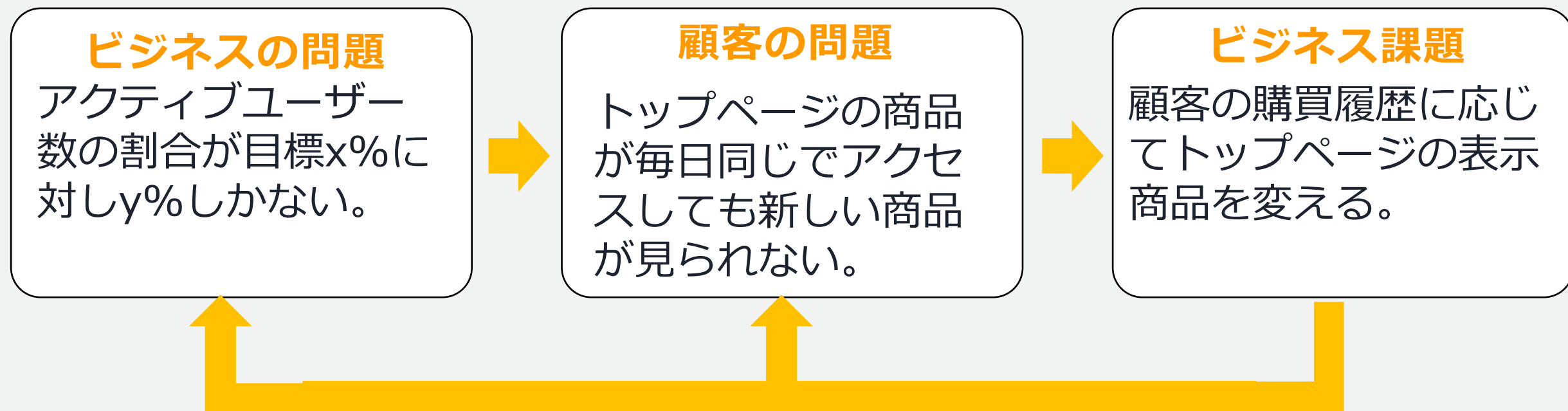
数学で公式に当てはめると簡単に問題が解けるように、勝ちパターンで解決したい課題が解けると簡単にビジネス成果を上げれる。

1. 機械学習プロジェクトの進行
- 2. ビジネス課題を定義する**
3. ビジネス課題の優先度をつける

良いビジネス課題は、ビジネスと顧客の問題を両方解決する。

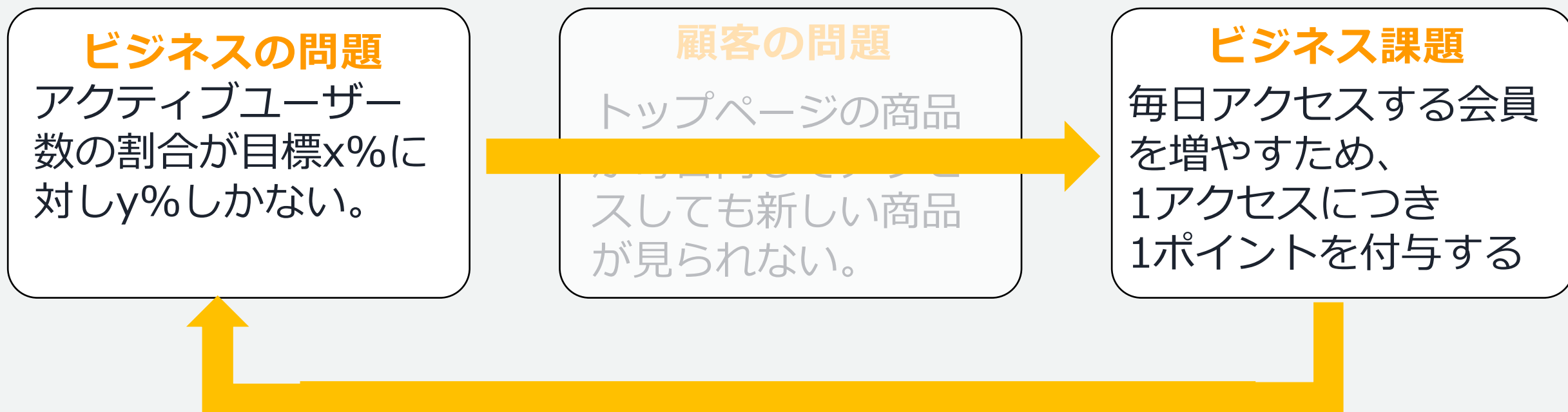
ビジネス課題が解決できるとTo-beとAs-isのギャップ(問題)が解消する。
ビジネスの問題の背後には、多くの場合顧客の問題がある。

例：ファッションのECサイト

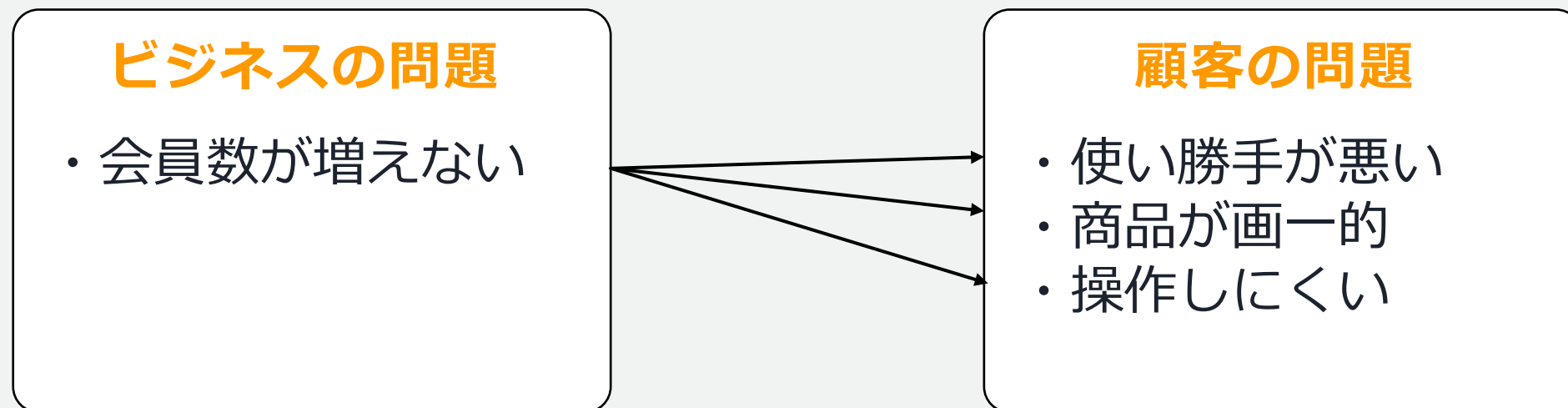


悪いビジネス課題は、問題を一時的にしか解決しない

ビジネスの問題の背後にある顧客の問題が特定できないと、問題が解決しなかったり再発したりする。



同じビジネスの問題でも、プロダクトの状況によって顧客の問題は異なる。



解決したいビジネスの問題が同じでも

回答：顧客の問題がプロダクトごと異なるから。

お客様は誰ですか？

AmazonとAWSの製品開発プロセス: Working Backwards

お客様にこだわる5つの質問。

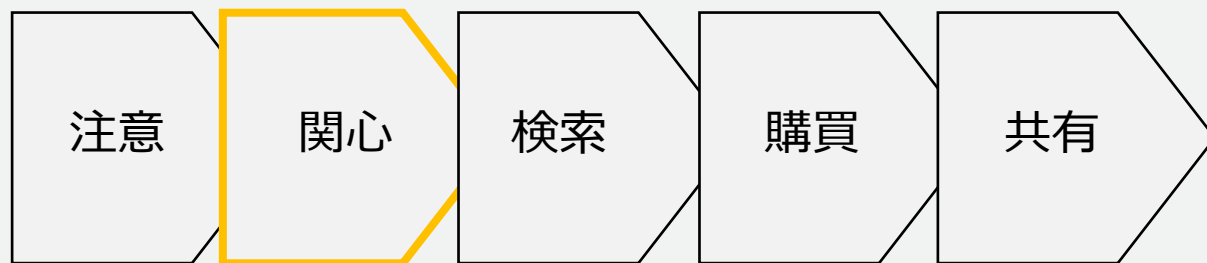
- **お客様は誰ですか？**
- お客様が抱える課題や新しい可能性は明確ですか？
- お客様が受ける価値やメリットは明確ですか？
- お客様のニーズやウォンツをどのように知りましたか？
- お客様の体験はどのように変わりますか？

ビジネスの問題も顧客の問題も、似たアプローチで特定できる。

ビジネスの問題の特定

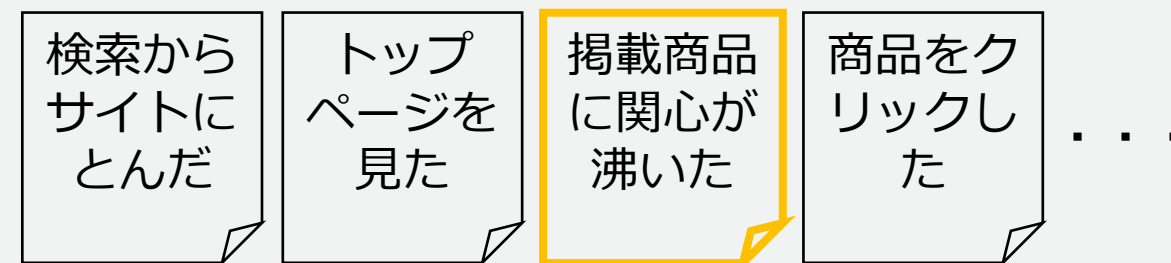
ビジネス成果に至るまでのプロセスを
整理・計測しボトルネックを発見

例: AISASフレームワーク



顧客の問題の特定

顧客満足に至るまでのプロセスを整理・計測しボトルネックを発見



例：ファッションサイトでパーソナライゼーションを導入する



AmazonとAWSの製品開発プロセス: Working Backwards

お客様にこだわる5つの質問。

- **お客様は誰ですか?**
- お客様が抱える課題や新しい可能性は明確ですか?
- お客様が受ける価値やメリットは明確ですか?
- お客様のニーズやウォンツをどのように知りましたか?
- お客様の体験はどのように変わりますか?

AmazonとAWSの製品開発プロセス: Working Backwards



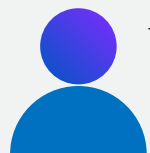
60%

ブランドにこだわりがあり、発売したらすぐに買いたい。
他のブランドに興味はない。



30%

ブランドにこだわりはなく、なるべく安く買いたい。
デザインにこだわりはない。



10%

ウィンドウショッピングのように、商品を眺めるのに使用。
実際は試着をして買いたいからサイト上では買わない。

例えば普通に「この商品もおすすめ」を導入したとして・・・



60%

「おすすめの商品」の大半は購入済み



30%

「おすすめの商品」の価格帯が合わない

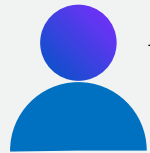


10%

そもそも買わない

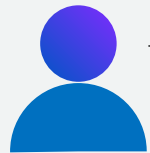
お客様は誰ですか？

AmazonとAWSの製品開発プロセス: Working Backwards



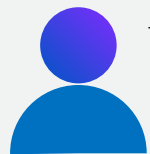
60%

ブランドにこだわりがあり、発売したらすぐに買いたい。
他のブランドに興味はない。



30%

ブランドにこだわりはなく、なるべく安く買いたい。
デザインにこだわりはない。



10%

ウィンドウショッピングのように、商品を眺めるのに使用。
実際は試着をして買いたいからサイト上では買わない。

お客様の行動を理解する。



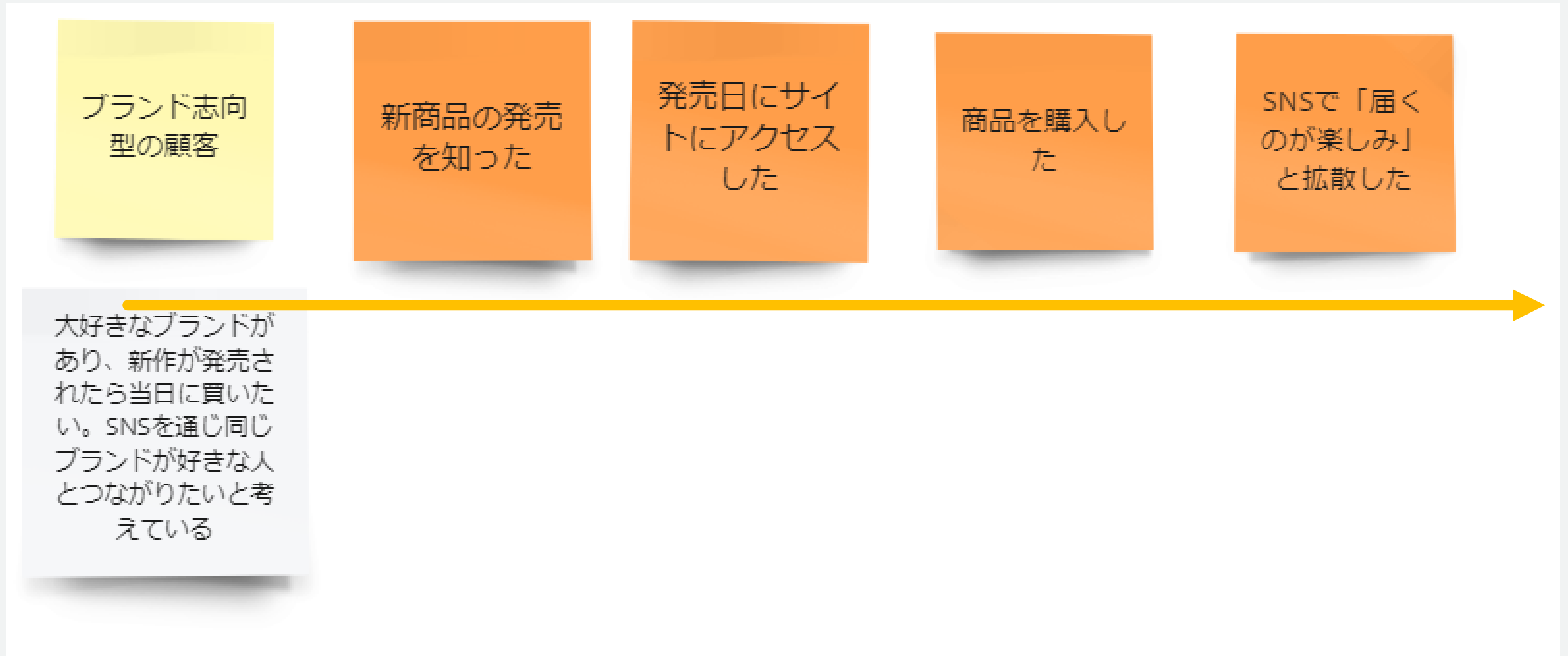
お客様の行動を理解する。

行動の例

- 新商品の発売を知った
- 発売日にサイトにアクセスした
- 商品を購入した
- SNSで「届くのが楽しみ」と拡散した

全て過去形=実際発生したイベントに注目

左から、時系列順にポストイットを並べる



問題が発生するポイントを特定する

行動の例

- 新商品の発売を知った
- 発売日にサイトにアクセスした
- 商品を購入した
- SNSで「届くのが楽しみ」と拡散した

問題が発生するポイントを特定する

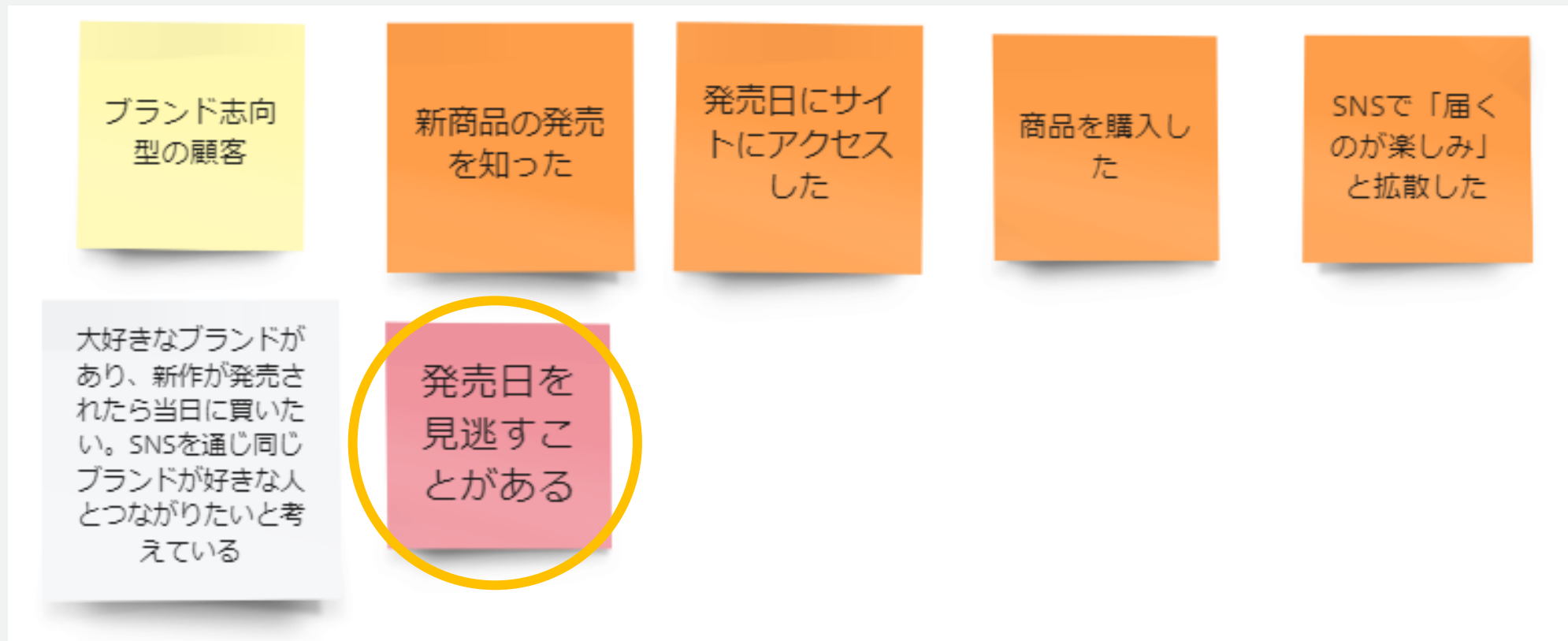
行動の例

- 新商品の発売を知った
- 発売日にサイトにアクセスした
- 商品を購入した
- SNSで「届くのが楽しみ」と拡散した



発売日を見逃す

問題を貼る



解決すべき課題を特定する

**顧客の「発売日を見逃す」問題を解決するため、通知を行いたい。
=ビジネス課題。**

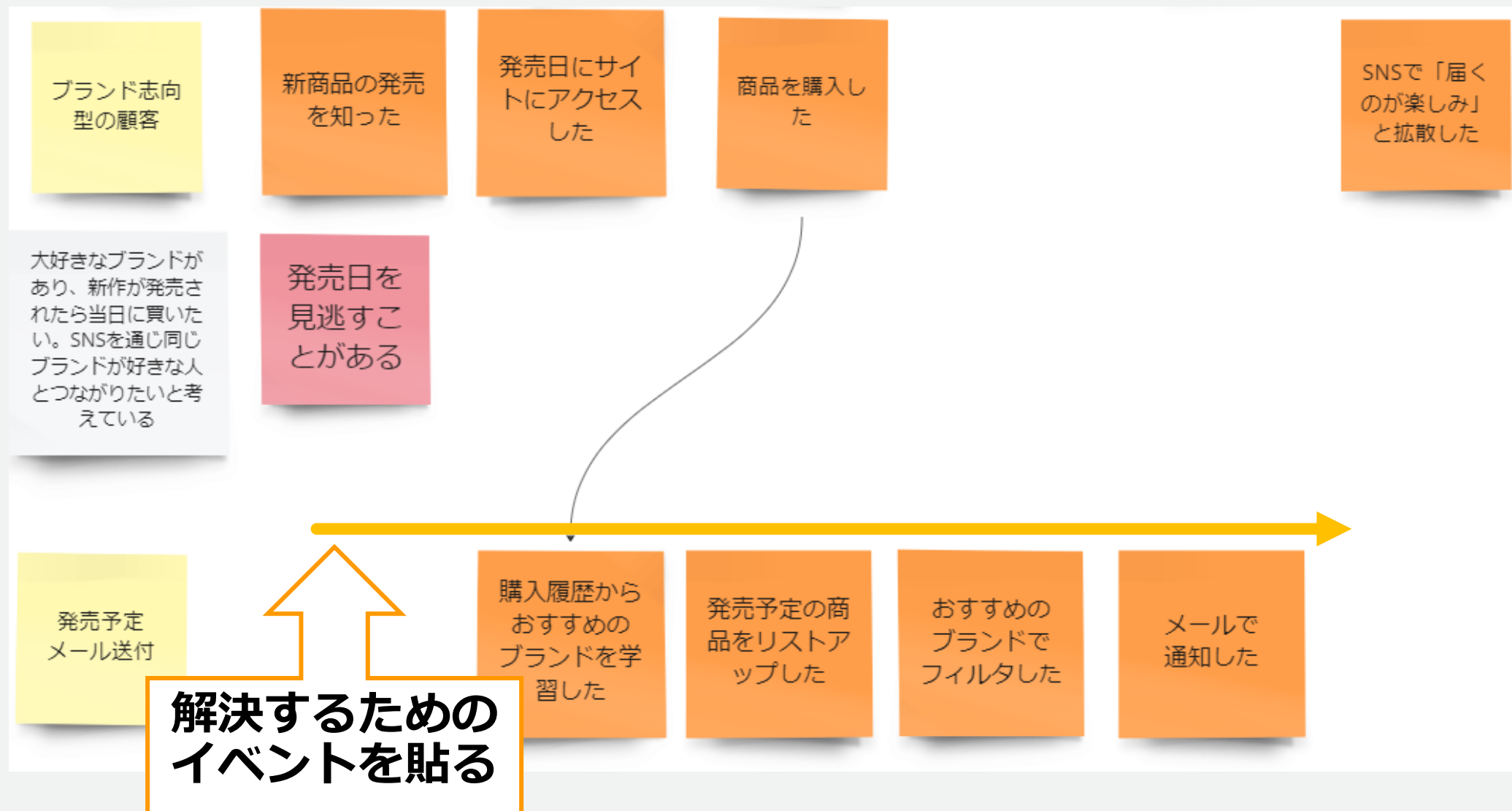
ベストプラクティスを「応用」し解決する

「パーソナライゼーション」の応用例

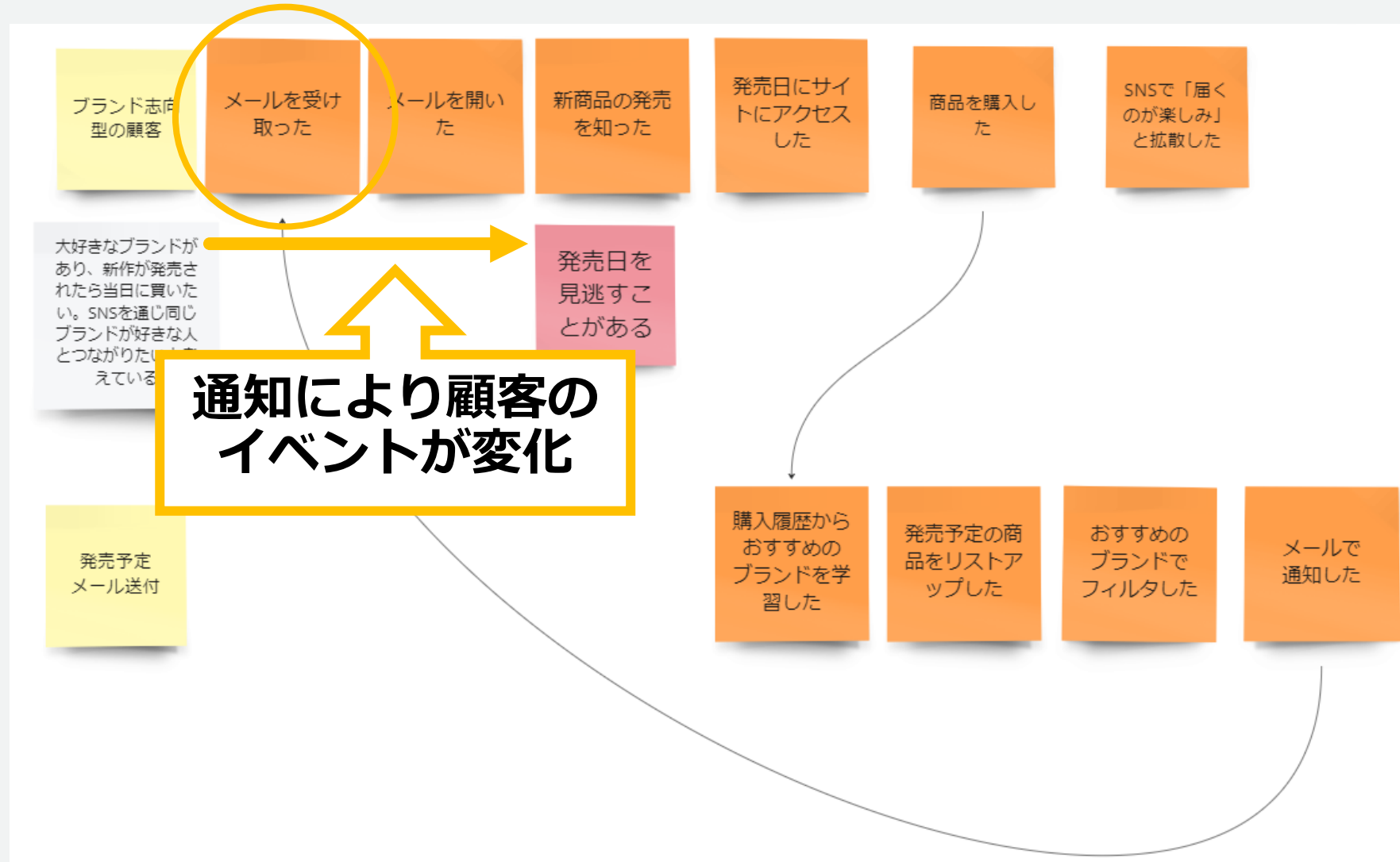
- 購入履歴から、おすすめのブランドを学習した
- 発売予定の商品をリストアップした
- おすすめのブランドでフィルタした
- メールで通知した

ビジネス課題がどのように解けるかを検討するため、データサイエンティスト、開発者の「テクノロジーリテラシー」を活かす。

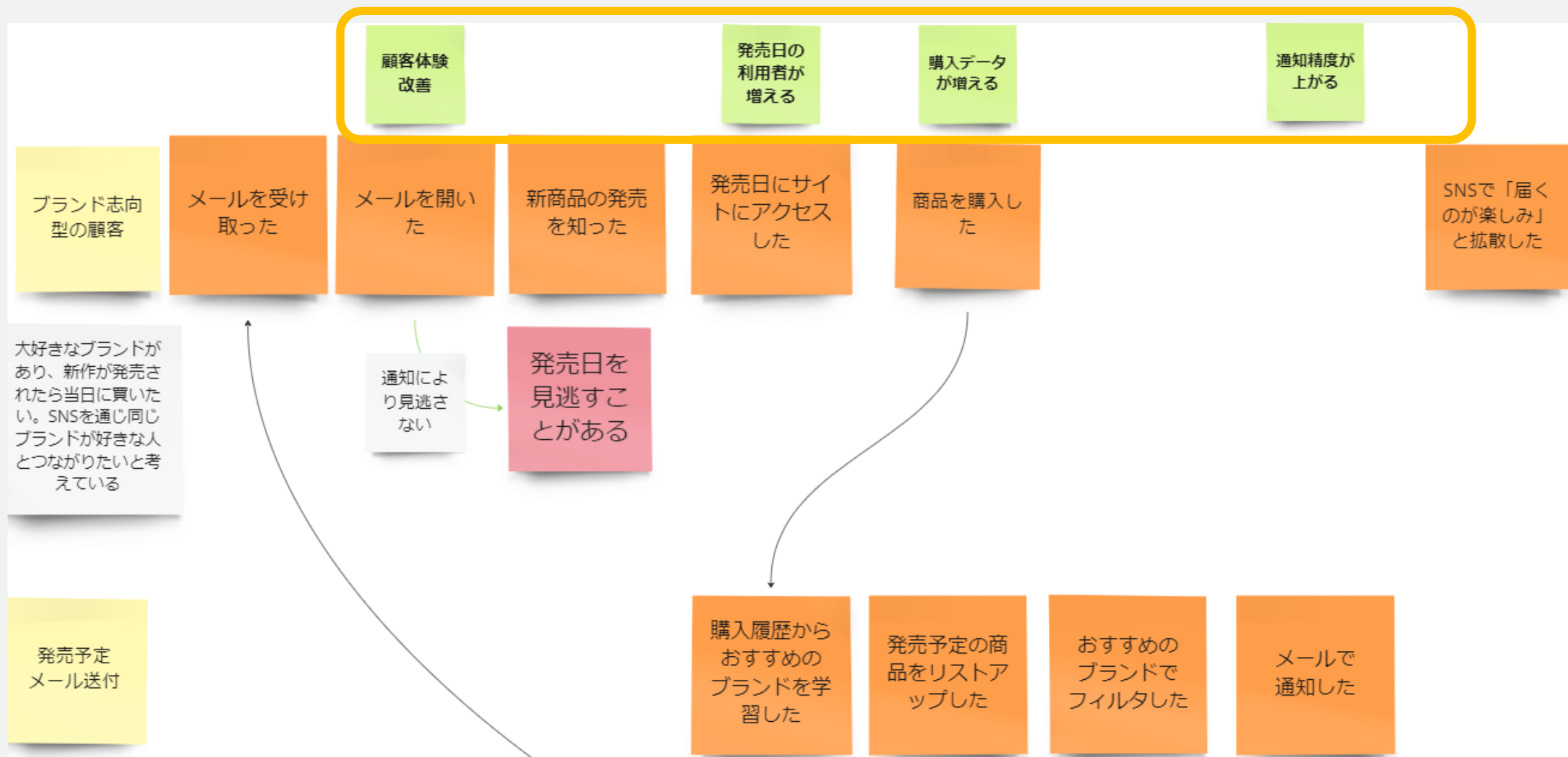
ベストプラクティスを「応用」し解決する



ベストプラクティスを「応用」し解決する



「勝ちパターン」の成立を確認する



お客様の行動、ストーリーを理解することの重要性



“顧客がプロダクト/サービスを引き入れるのは、彼らにとって重要なジョブが発生し、まだ満たされていない時に、それを解決するためだ。この「なぜ」を理解するかどうか、あるイノベーションは成功し、別のイノベーションはそうでないのかの分岐点となる”（「ジョブ理論」より）



“話して記録: ストーリーを話しながら、カードかポストイットに考えていることを書き出して思考を表出化しよう”

**“何も言わずにカードを並び替えるだけでコミュニケーションできる”
（「ユーザーストーリーマッピング」より）**

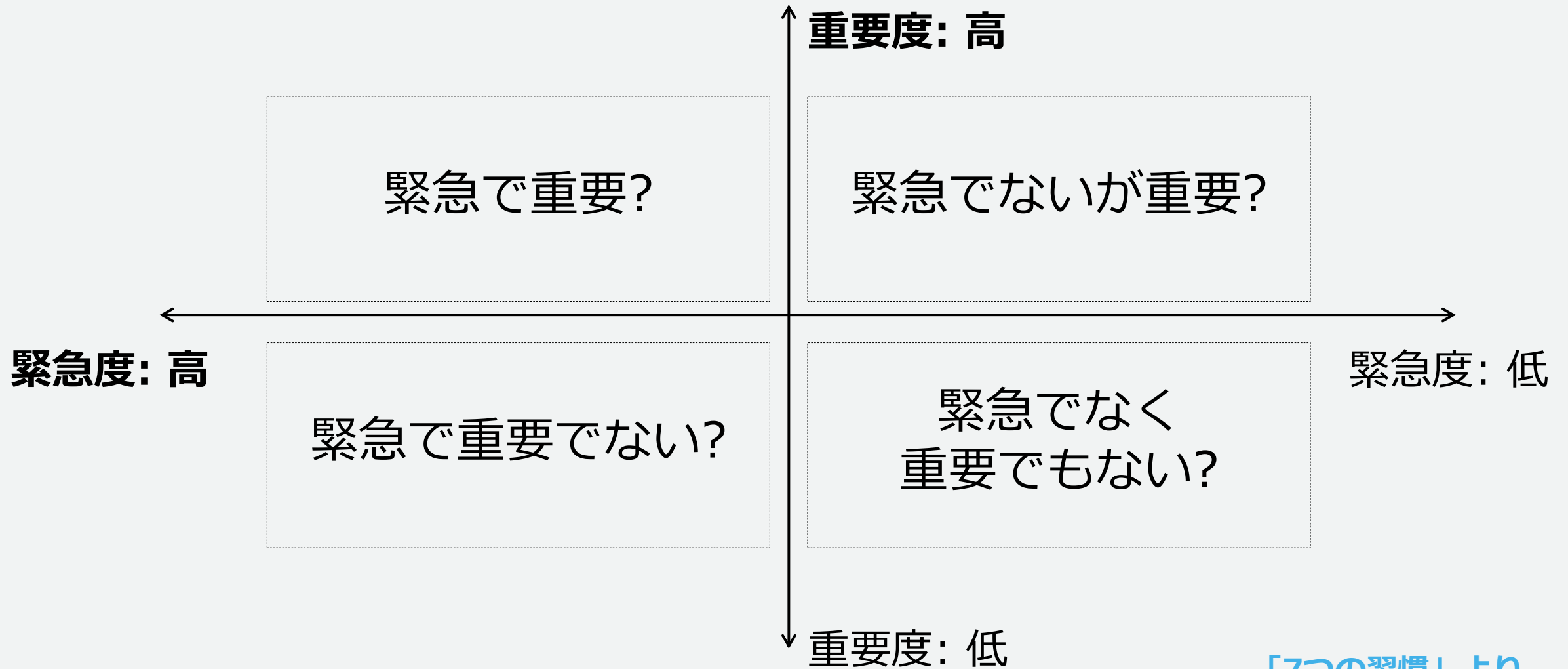
ワークショップでは、ストーリー理解の手法として Event Stormingを使用。

	ストーリーの表現	優先順位の表現	ソフトウェア設計
カスタマージャーニーマップ	○		
ユーザーストーリーマッピング	○	○	
ユースケース図			○
Event Storming (Big Pictureまで)	○		○

さらに先のステップへ進むことで、ドメイン指向に基づくソフトウェア設計が行える(進まなくてもよい)。

1. 機械学習プロジェクトの進行
2. ビジネス課題を定義する
- 3. ビジネス課題の優先度をつける**

ビジネス課題の優先度は?

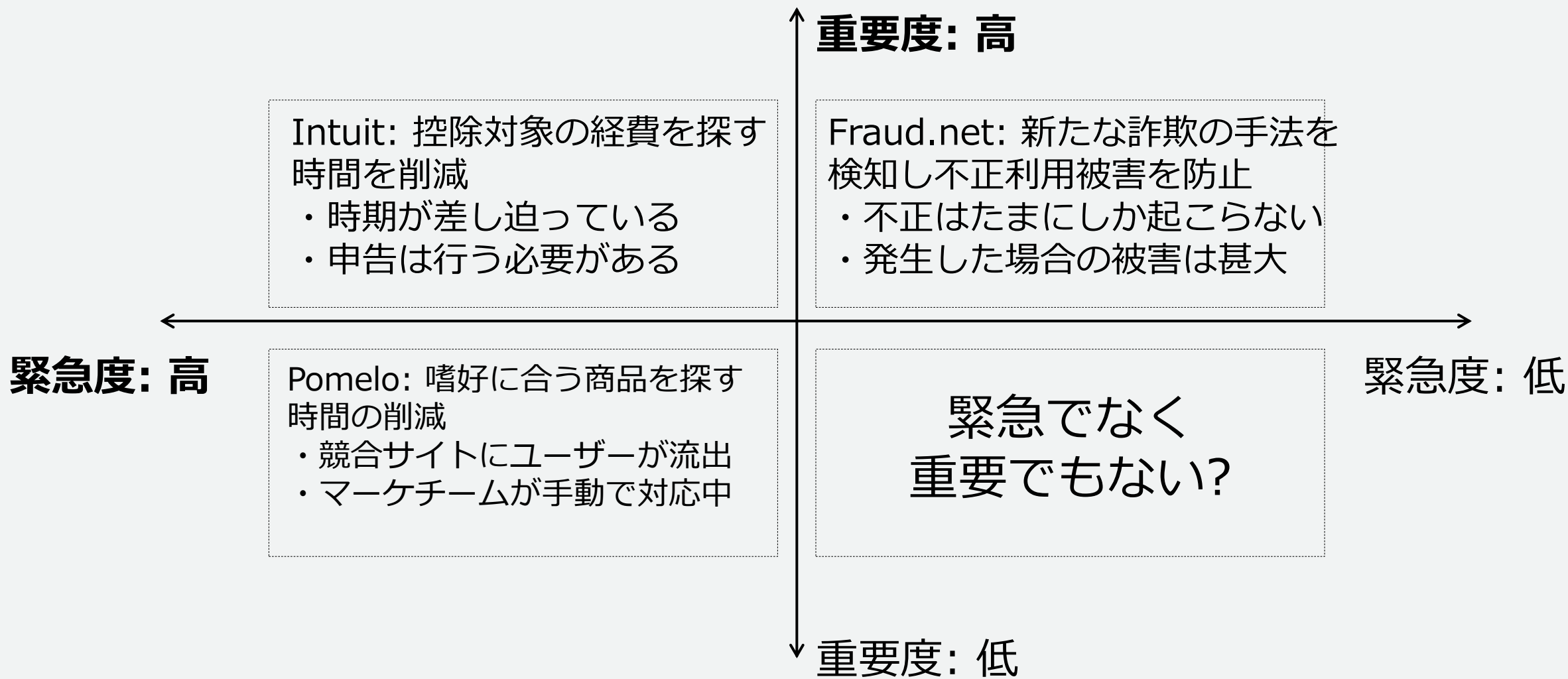


「7つの習慣」より

3つの事例、一番先に取り組むとしたらどれ?

- Intuit: 控除対象の経費を探す時間を削減
- Pomelo: 嗜好に合う商品を探す時間の削減
- Fraud.net: 新たな詐欺の手法を検知し不正利用被害を防止

回答の一例 ※想定している状況に大きく依存



緊急度と重要度の観点

- 緊急度
 - 時期的な制約がある
法規制の開始、顧客や市場の温度感、決算等のイベント
- 重要度
 - 実現した場合のインパクト
問題の大きさ x 問題を抱えている顧客の数

实践

実践

- フォーカスするプロダクトの「顧客」を定義してください。
- 理解編でリストアップしたベストプラクティス事例を、顧客にとっての緊急度・優先度から並び替えてください
- 最も優先度が高い事例について、顧客の体験するイベント・問題を洗い出しポストイットを並べて表現してください。
- 勝ちパターンが成立するか確認してください。

応用編の復習

応用編の復習

Q: 事例通りにしても効果が出ないのはなぜか?

A: 顧客の問題がプロダクトごと異なるから。

良いビジネス課題

良いビジネス課題はビジネスの問題と顧客の問題を同時に解決する。Event Stormingをはじめとしたストーリーを理解する手法は、顧客の問題が解けているか確認するのに使える。

優先順位付け

緊急度と重要度の観点から、課題の優先順位をつけることができる。

ユーザー

顧客を表すポストイット

Example: 経理担当者、会計担当者

ドメイン
イベント

顧客の行うイベント

Example: 領収書を受け取った、勘定を登録した、支払いした。

問題

顧客の問題

効果

解決策による効果 (勝ちパターンの確認に仕様)

Example: 領収書の入力時間が短縮する



TIME FOR A BREAK

See you in 5 minutes!

This material is based on the AWS Machine Learning Embark Program,
an immersive ML training program provided by Amazon Machine Learning Solutions Lab



Event Stormingとは？

- ポストイット/付箋を利用しながらユーザーの業務イベントと、業務を処理するためのルールを理解し、理解に基づきソフトウェアを設計するための手法。

**開発者が要件を正確に
理解するための手法**

- プロダクトオーナーやドメインエキスパートから要件を説明する時に用いる。クラス設計の開始前、ユーザーテストのケース作成時にも用いられる。

Event Stormingのプロセス (今回Big Pictureのみ使用)

Step1: Big Picture

主要なアクター(個人事業主/銀行など)が実行する操作に連なるイベントを時系列にならべる。

Step2: Process Modeling

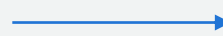
操作に必要な情報、イベント間で受け渡す情報、イベントの分岐(ロジック)を追加する。

Step3: Software Design

イベントによって引き起こされる状態変化を特定する。

Step4: ドメインモデルを深く考える

Step5: イベントの整合性、モデルの妥当性を議論する



1~5を繰り返す。

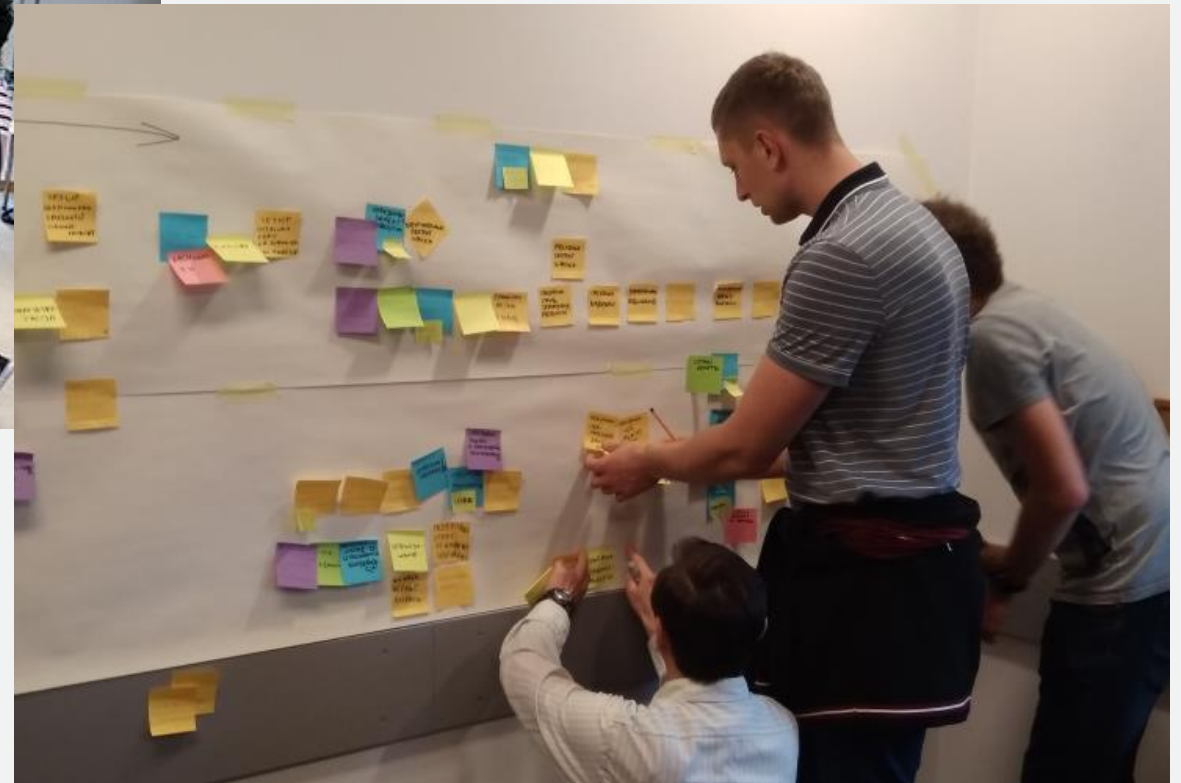
Event Stormingの様子



[What Is Event Storming? How Is It Key to Agile Strategy Success?](#) より引用

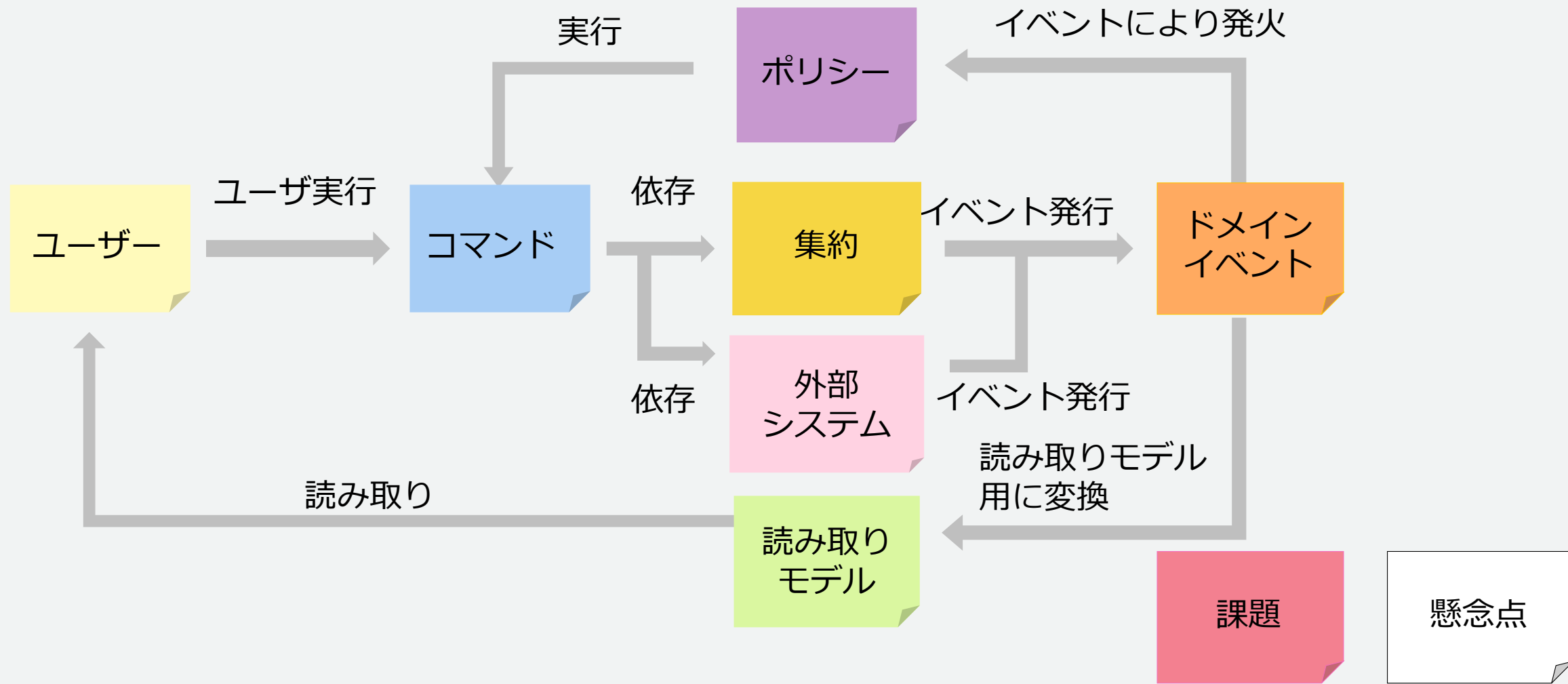
発案者の動画

<https://www.youtube.com/watch?v=mLXQIYEwK24>



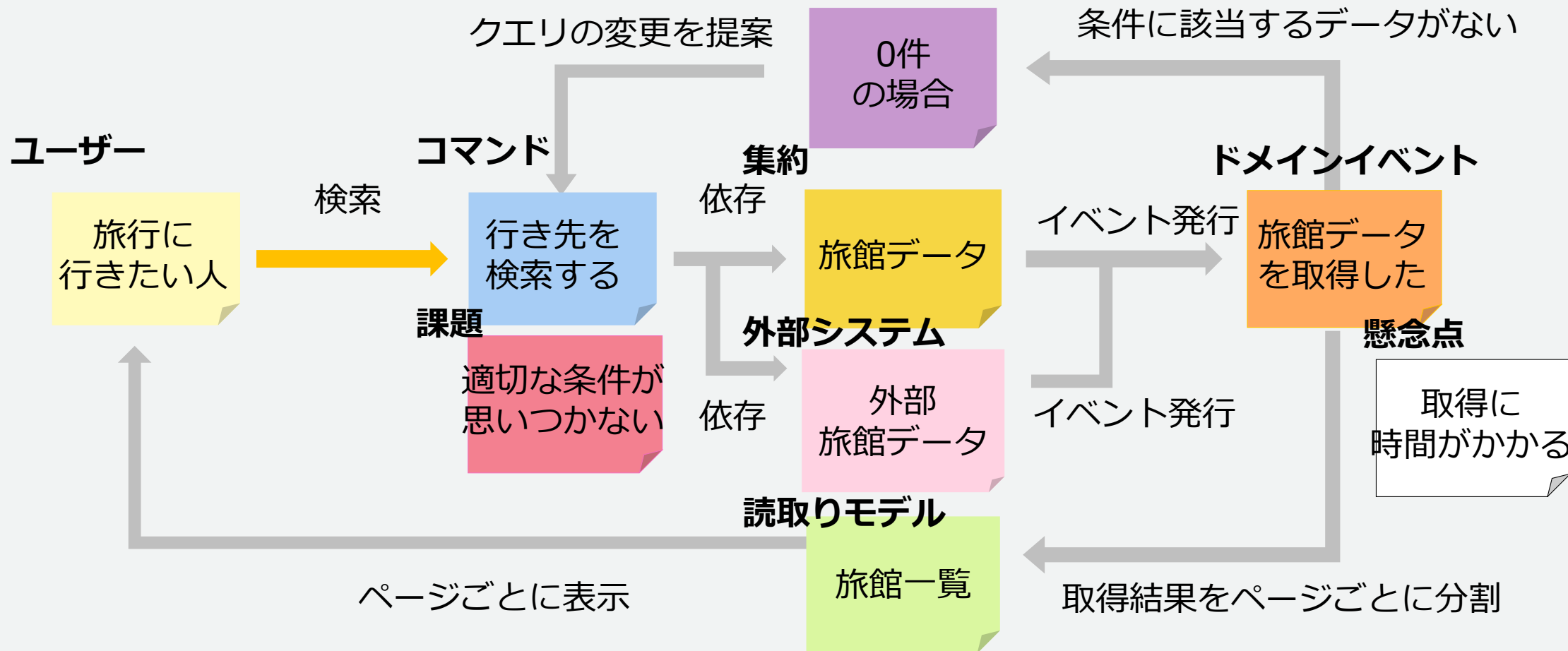
[Event Storming – innowacja w projektach IT](#) より引用

Event Stormingで利用するポストイット



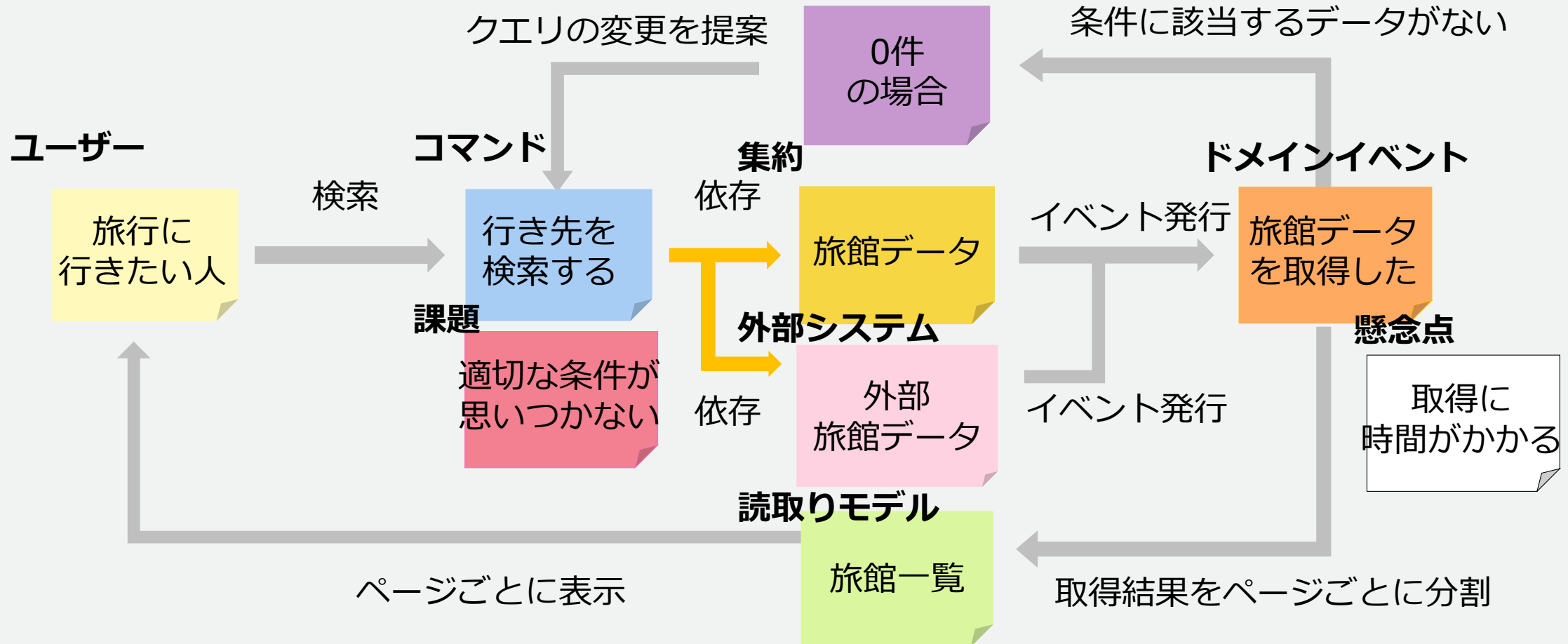
Event Stormingで利用するポストイットの関連イメージ

ポリシー



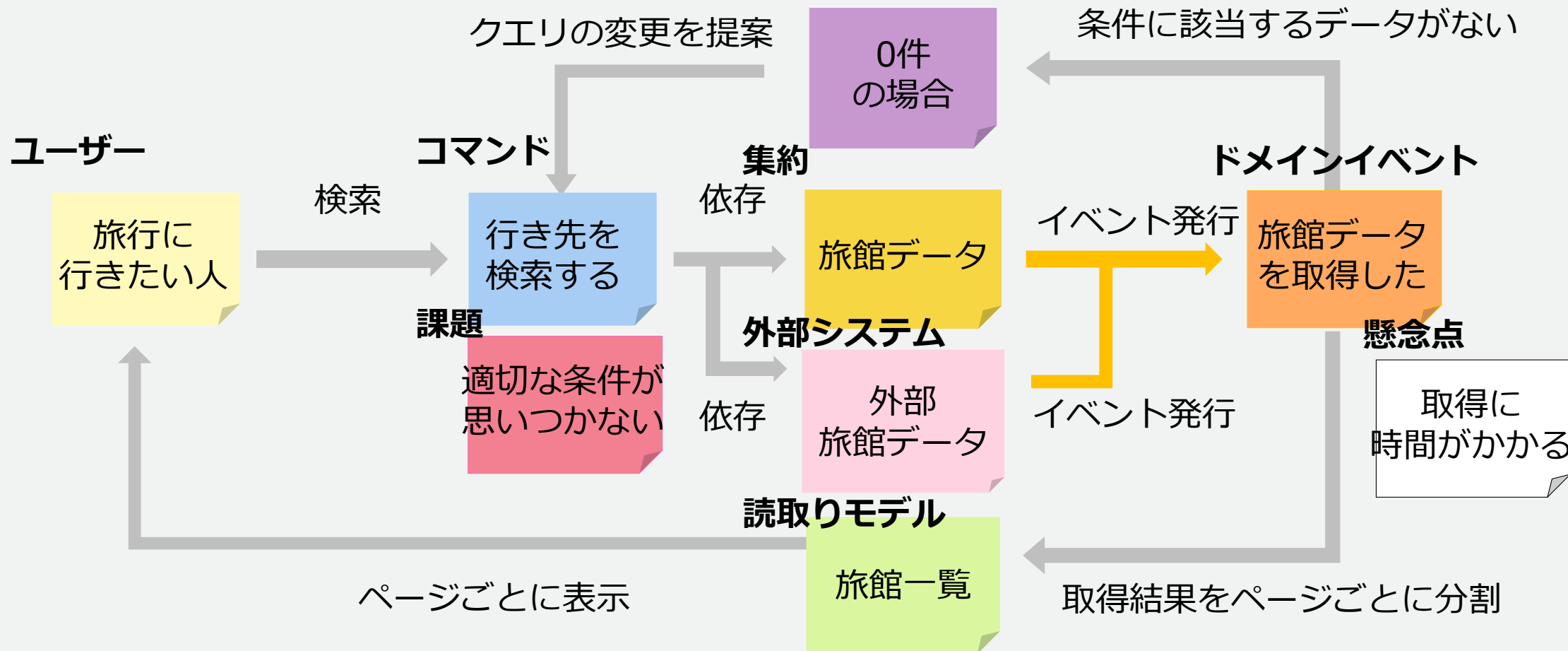
Event Stormingで利用するポストイットの関連イメージ

ポリシー



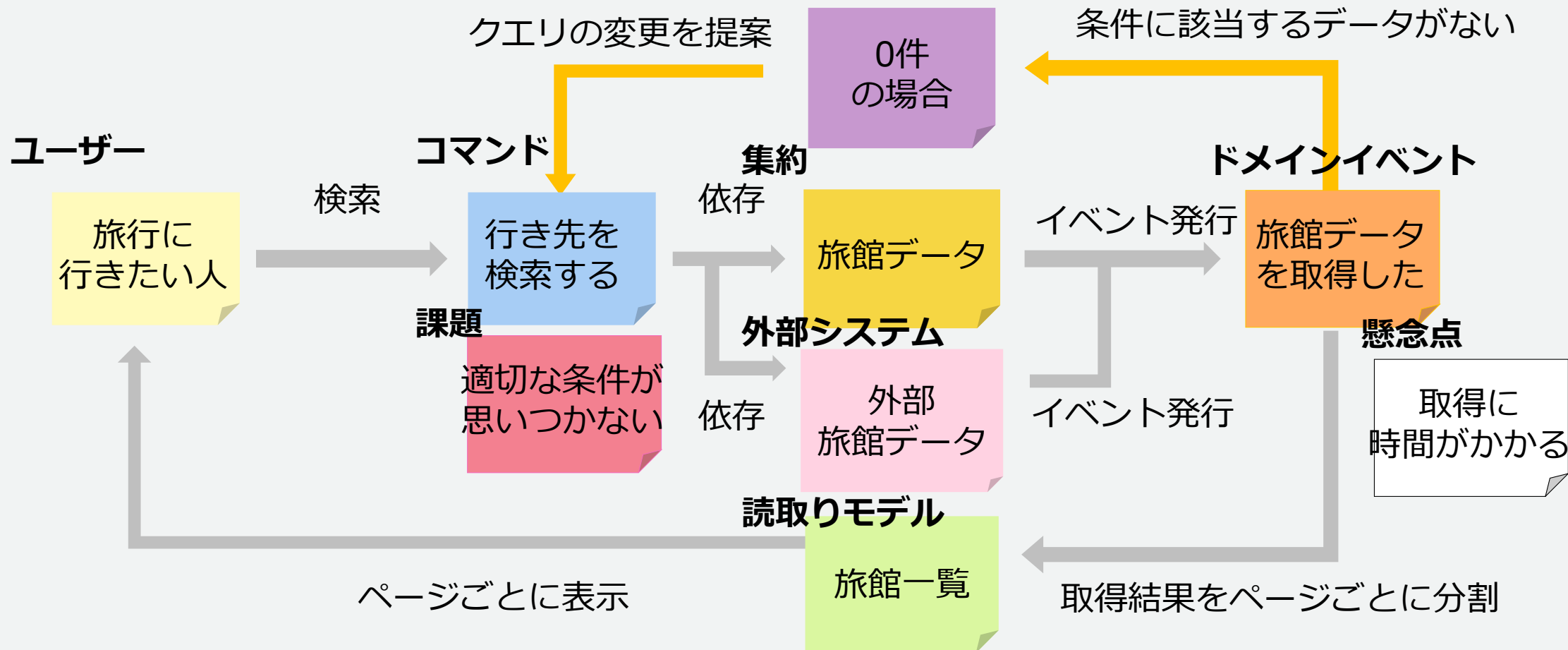
Event Stormingで利用するポストイットの関連イメージ

ポリシー



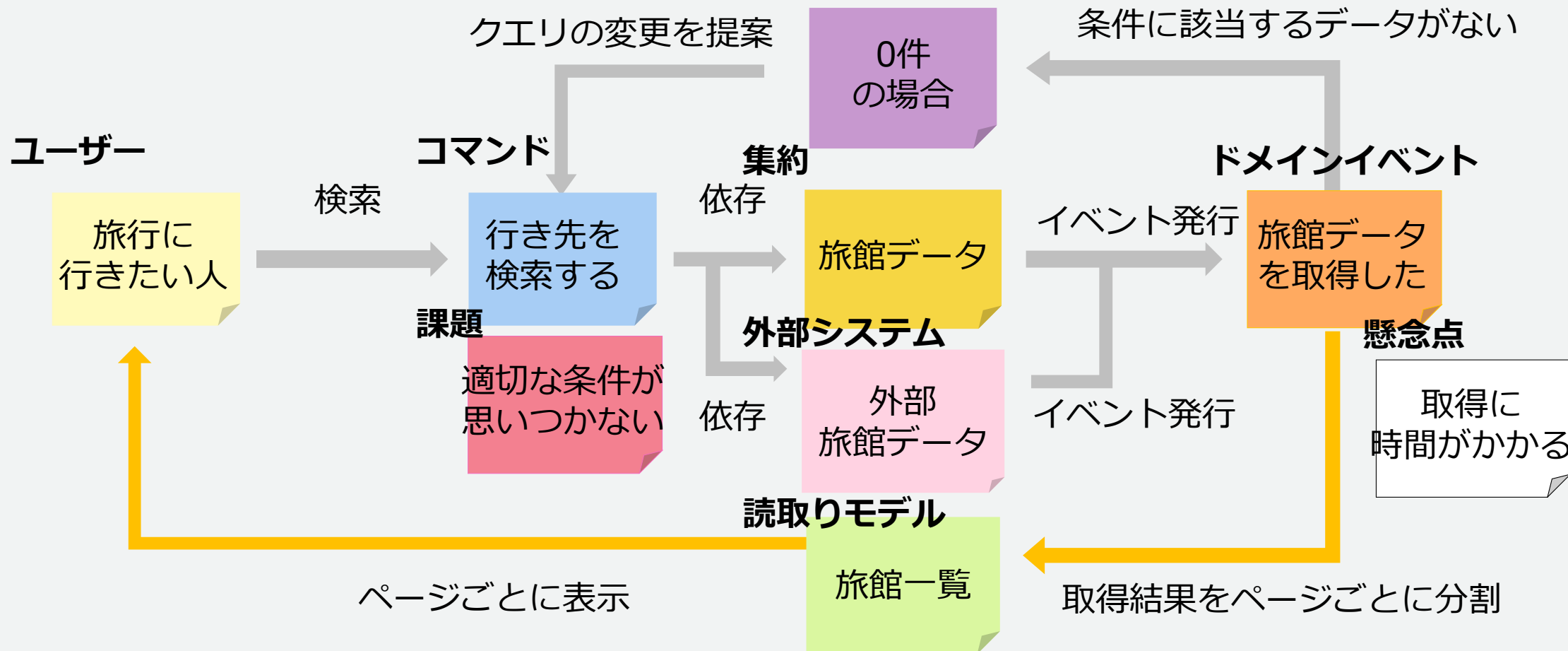
Event Stormingで利用するポストイットの関連イメージ

ポリシー



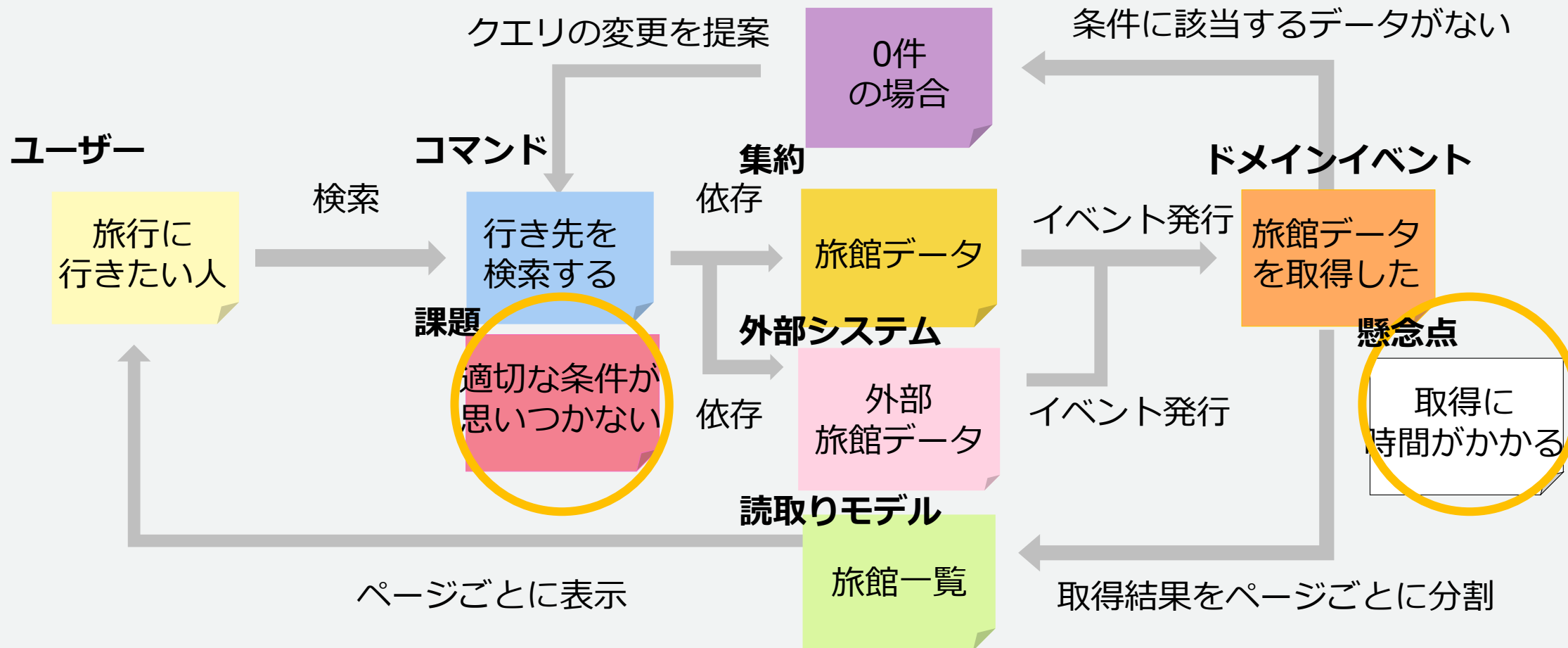
Event Stormingで利用するポストイットの関連イメージ

ポリシー



Event Stormingで利用するポストイットの関連イメージ

ポリシー



ユーザー

業務を実行する主体。

Example: 経理担当者、会計担当者

ドメイン
イベント

業務を成立させるのに必要な作業。

Example: 領収書を受け取った、勘定を登録した、支払いした。

課題

作業中に発生する課題

読み取り
モデル

ユーザーが意思決定するのに参照するデータ。

Example: 経費精算の種別一覧、勘定の一覧。

コマンド

ドメインイベントをはじめるトリガとなる意思決定。

Example: 交通費精算を行う、勘定を登録する。

ポリシー

コマンドを起動するルール、ドメインイベント内の条件分岐。

Example: 定期実行プログラム、勘定の登録がある/ない。

集約

ドメインイベントが参照/更新するサービス管轄内のリソース。

Example: 伝票明細データ、請求ステータス。

外部
システム

ドメインイベントが参照/更新するサービス管轄外のリソース。

Example: 外部会計システム、外部帳票出力システム。