

# スクラムと品質

— フレームワーク、プロセス、型 —

三菱電機株式会社  
細谷 泰夫

✓ 3つの原則

✓ 4つのミーティング

✓ 3つのロール

✓ 3つのアーティファクト

# 3つの原則

---

✓ 透明性

✓ 検査

✓ 適応

# 4つのミーティング

---

- ✓ スプリント計画ミーティング
- ✓ デイリースクラム
- ✓ スプリントレビュー
- ✓ ふりかえり

# 3つのロール

---

- ✓ プロダクトオーナー
- ✓ スクラムマスタ
- ✓ チーム

# 3つのアーティファクト

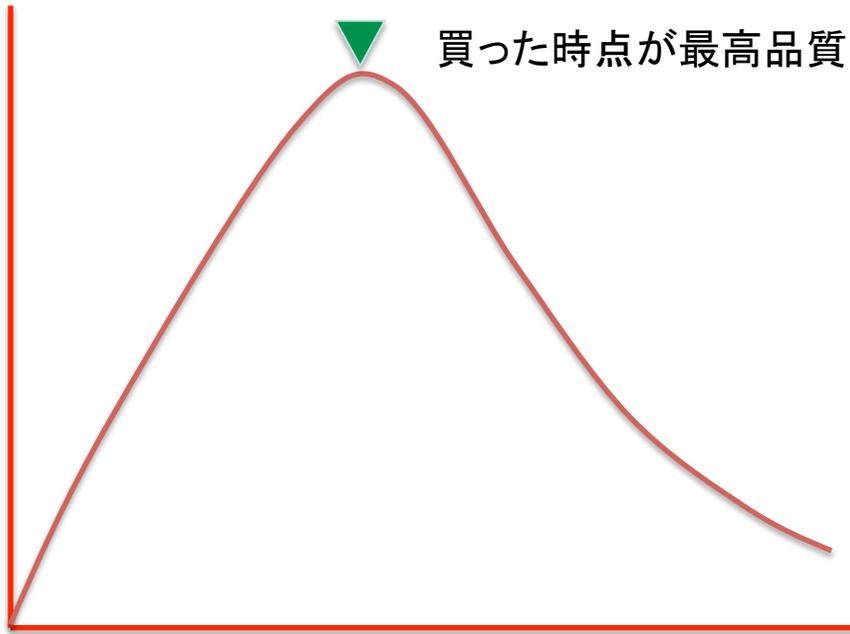
---

- ✓ プロダクトバックログ
- ✓ スプリントバックログ
- ✓ インクリメント（リリース可能なプロダクト）

# 品質について考える

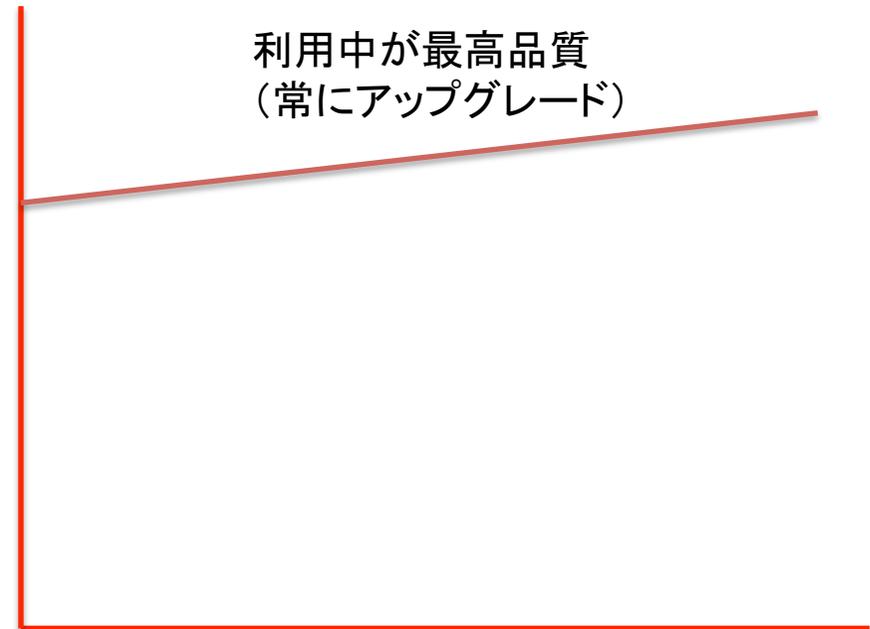
SonicGarden 倉貫さんのブログで触れられたPoint of Sales、Point of Use  
<http://kuranuki.sonicgarden.jp/2012/03/>

Point of Sales



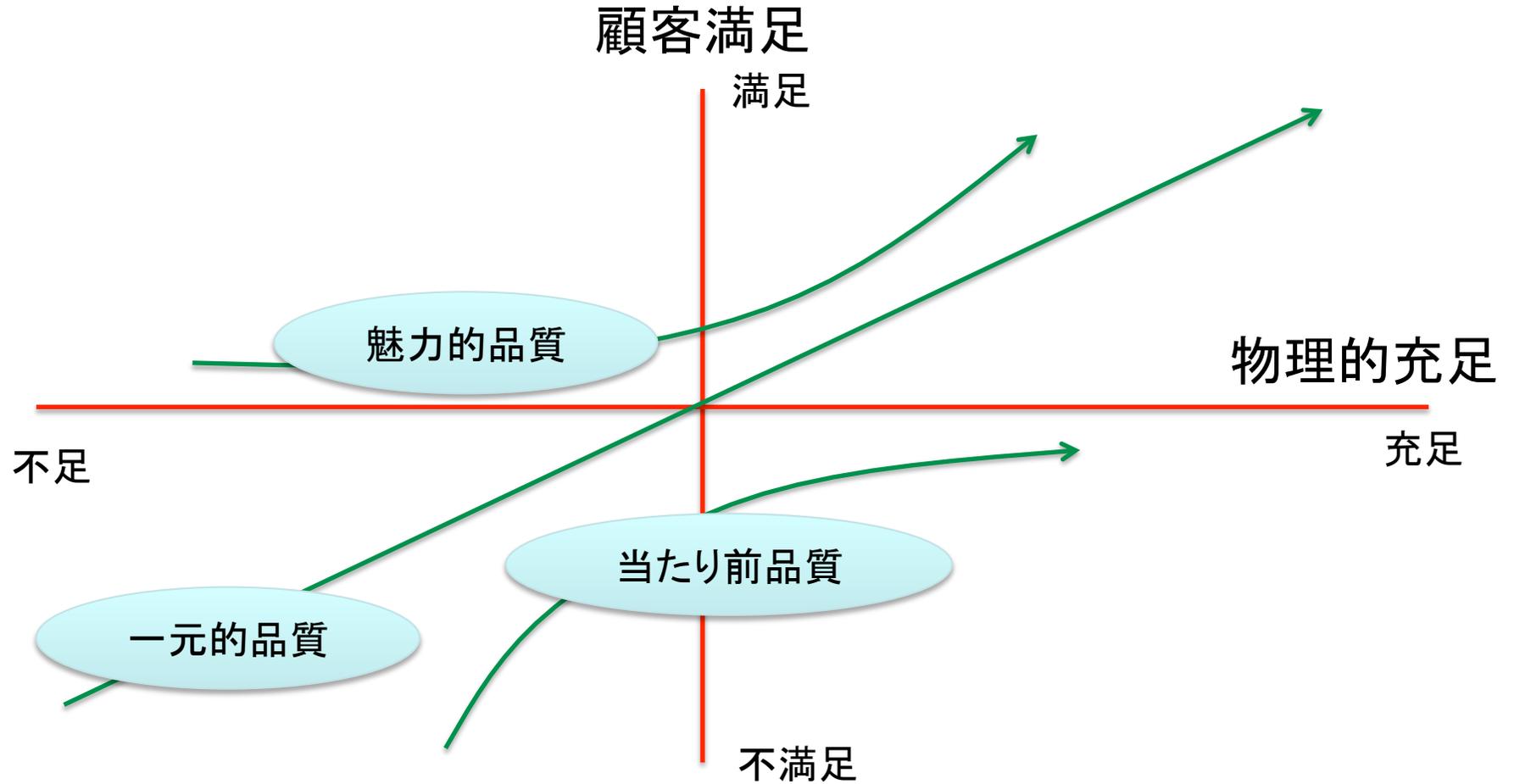
製造業

Point of Use

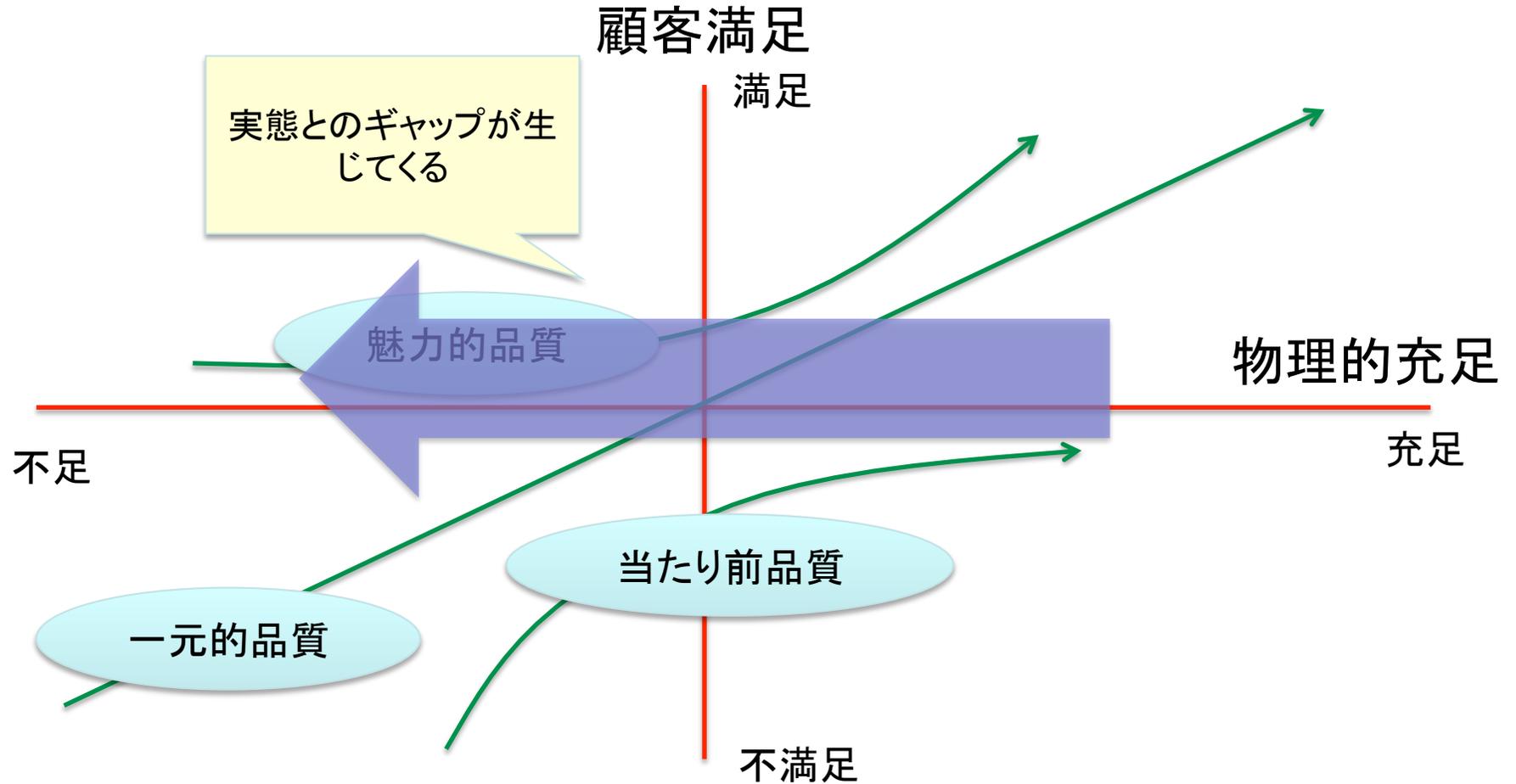


サービス業

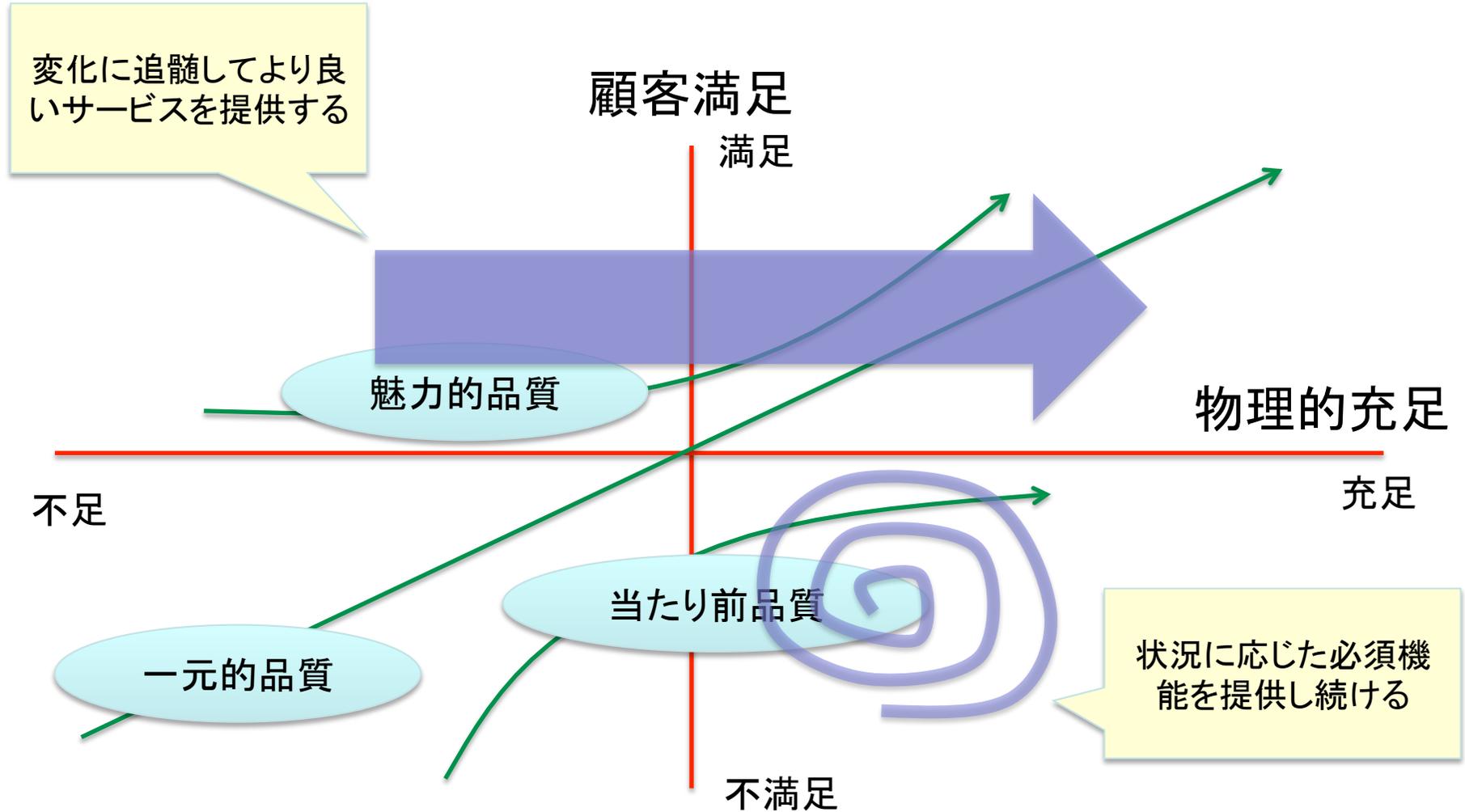
# 狩野モデル



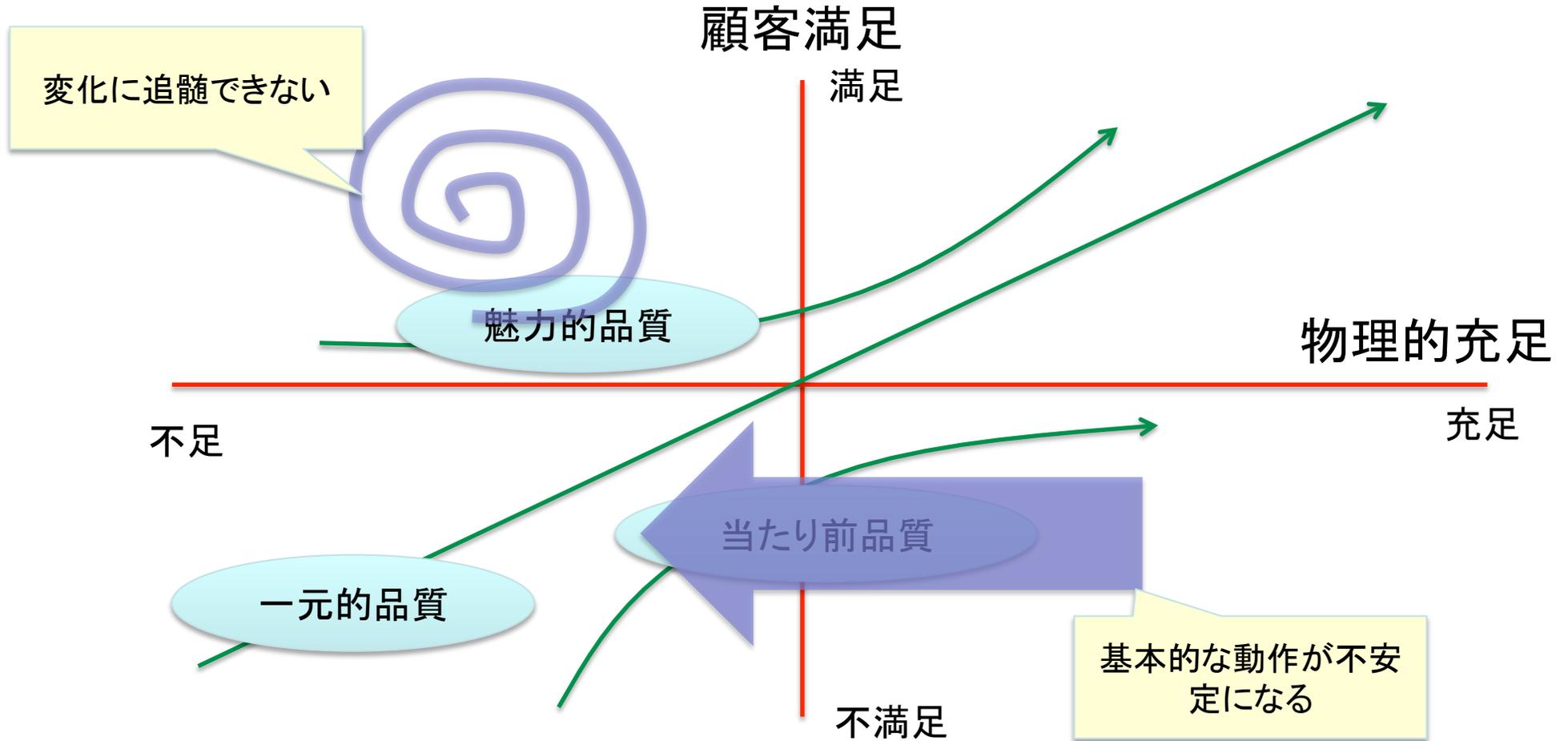
# Point of Sales



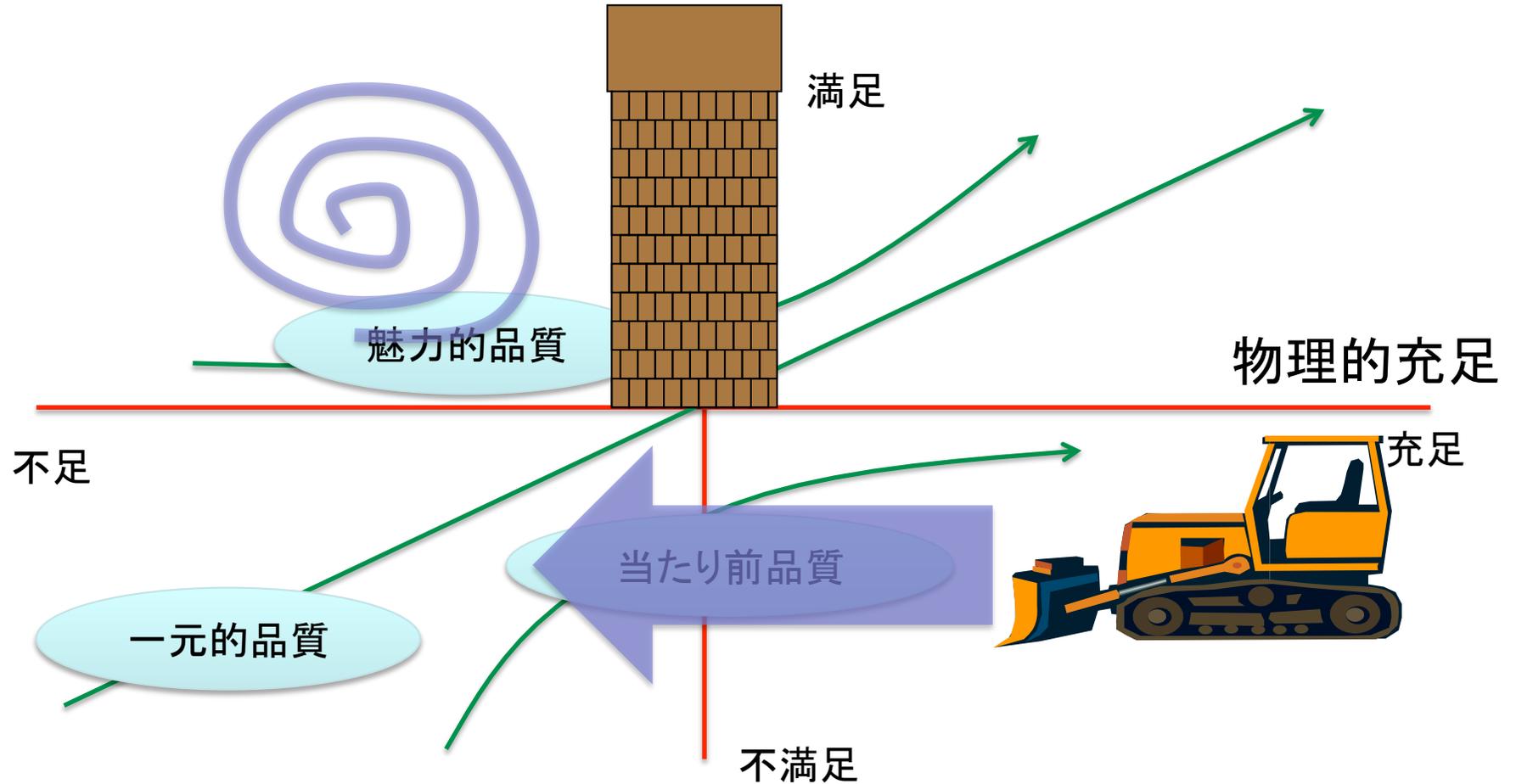
# Point of Use



# Point of Use: 品質が悪いと



# 後から挽回するのは難しい



# 品質が悪くなる要因

---

- 硬直した計画
- オーバーコミットメント

# スクラムのルール

---

- ✓ プロダクトオーナーは、プロダクトバックログの優先順位を付けることができる。
- ✓ チームのみがプロダクトバックログ項目を見積もることができる。
- ✓ スクラムマスターはスクラムのルールを壊す存在を排除する権限を持つ。

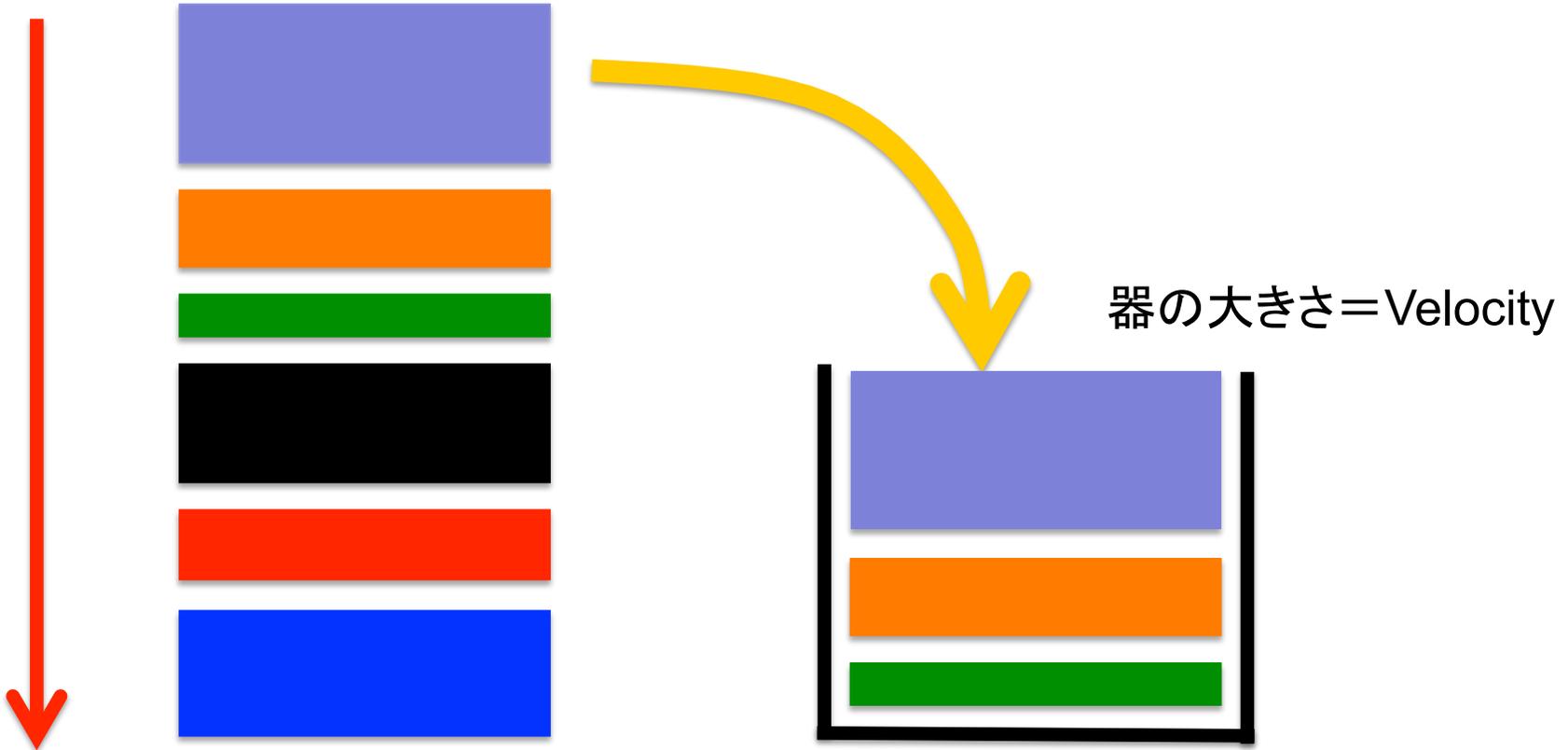
# ルール違反

---

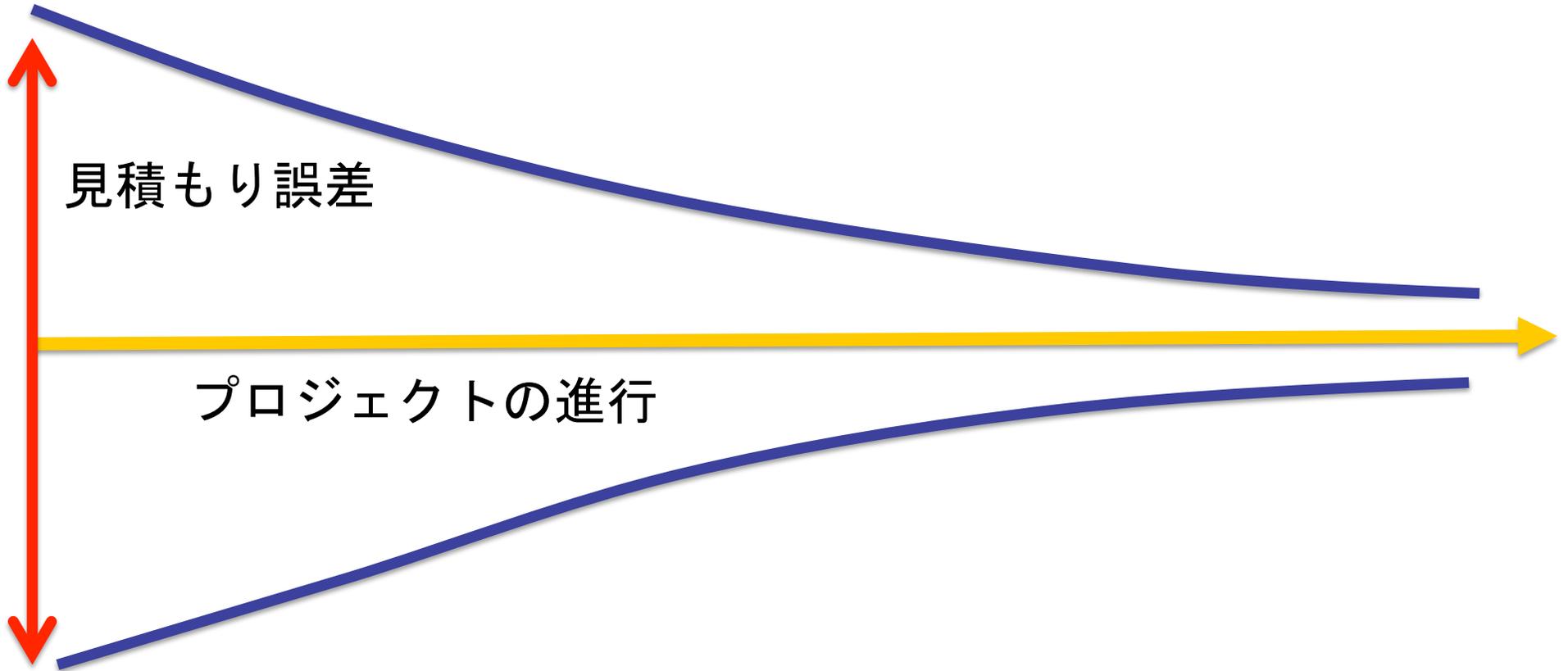
- ✓ プロダクトオーナーや組織がチームの見積もりを否定する。
- ✓ スプリントの期間を延長する。
- ✓ チームが情報を隠す。

# スプリントのスコープの決めり方

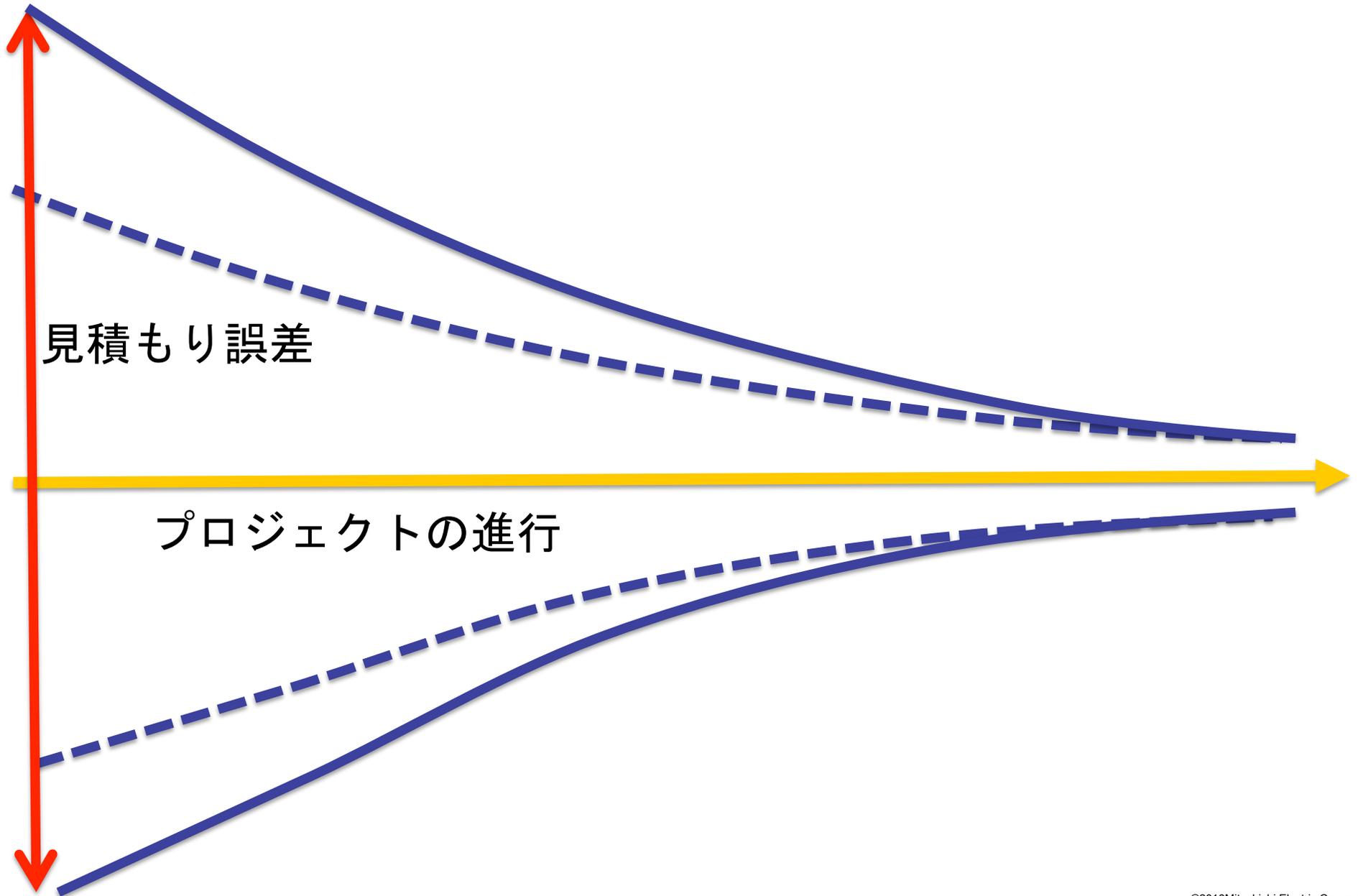
優先順位



# 見積もり



# ルール違反をすると



# ルール違反の代償

- 信頼関係

得られた価値に釣り合わない価格

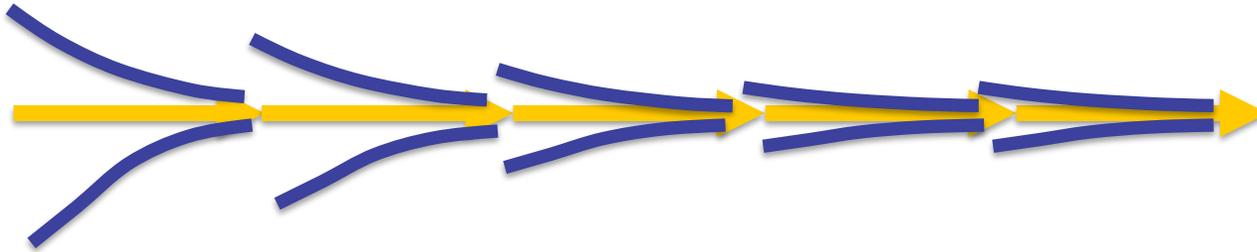
費やした労力に釣り合わない価格

- 成果物の品質

矛盾をプロダクトに吸収させる。

ルール違反によってスクラムが瓦解するのを防ぐのがスクラムマスタの大事な役割。

# スクラムでの見積もり



- 見積もりのスコープを小さくする
- 実績を元に相対的に見積もる
- チームにバッファは必要としない(タイムボックスに合わせてスコープを変える)

# 完了の定義

- 「何をもってプロダクトバックログ項目を完了とするか？」の定義
- スクラムのルールが頻繁に破られる状況だと「完了の定義」は機能しない。
- チームが成果物の品質にコミットする。



どうやって？

✓ フレームワーク

✓ プロセス

✓ 型

# フレームワーク

---

- スポーツに例えると、ルールにあたる。
- サッカーは11人で、キーパー以外は手は使えない etc
- スクラムはフレームワーク

# プロセス

---

- 2人以上の人が連携して何かをする
- 1人の人が複数の手順で何かをする
- サッカーに例えると、ワン・ツー、セットプレー、フォーメーションなど

# 型 土

- 武道の型のようなもの、基本動作
- サッカーだとインステップキックやトラップ
- 基本動作ができていないと、プロセスを実行できない。

- ルールを守るだけでは面白いゲームはできない。
- 相手、天候、点差など変化する状況に合わせて戦い方を変えていく必要がある。
- メンバーの組み合わせによっても実行できる戦い方が変わってくる。



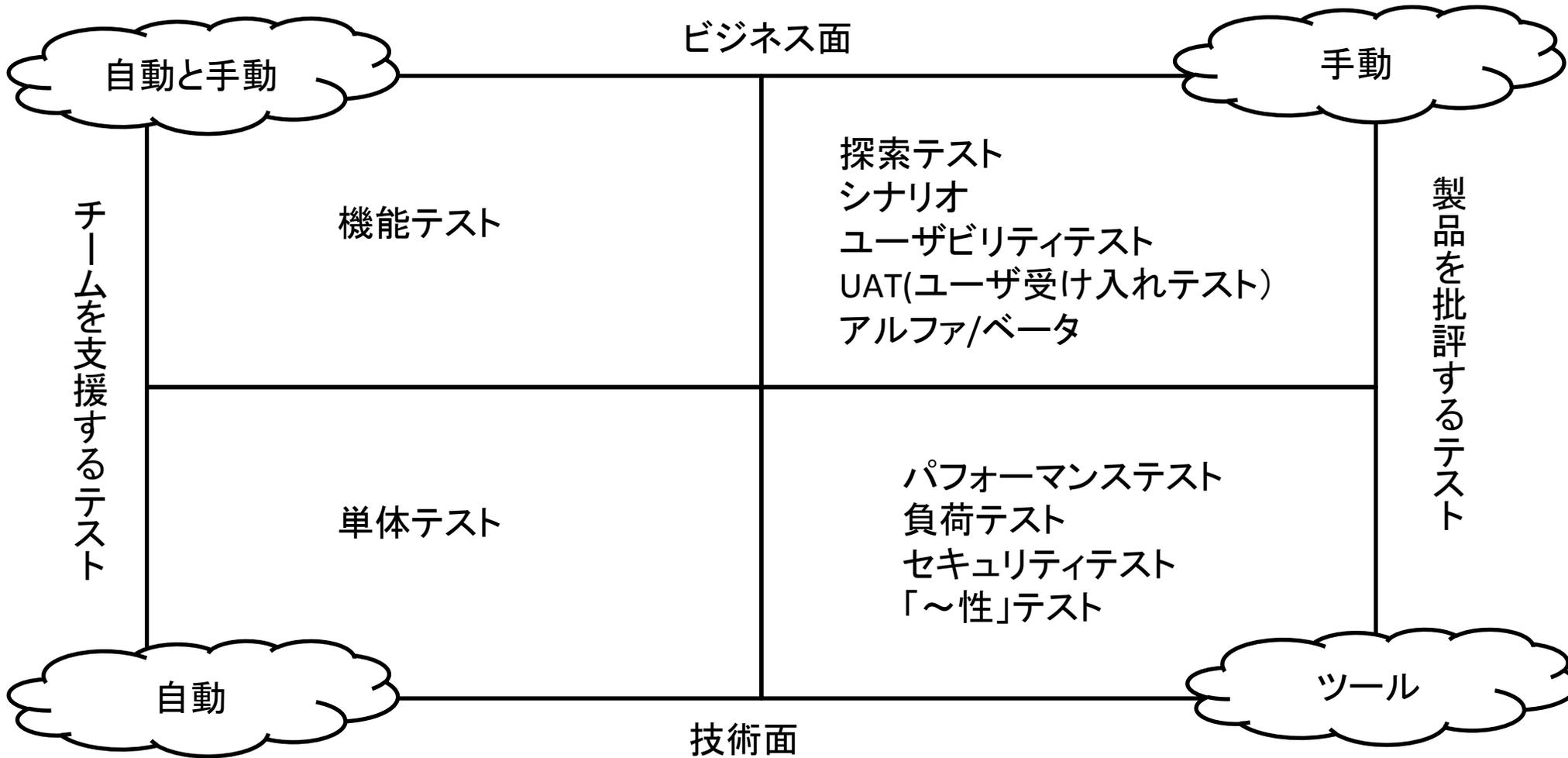
どのようなプロセス、型を使うかを自分達で考えることが重要。

セッションで紹介した事例「アジャイルプラクティスを活用したチームとしての品質確保の取り組み」については以下で公開されています。

<http://www.juse.jp/sqip/symposium/archive/2012/day1/files/A2-1.pdf>

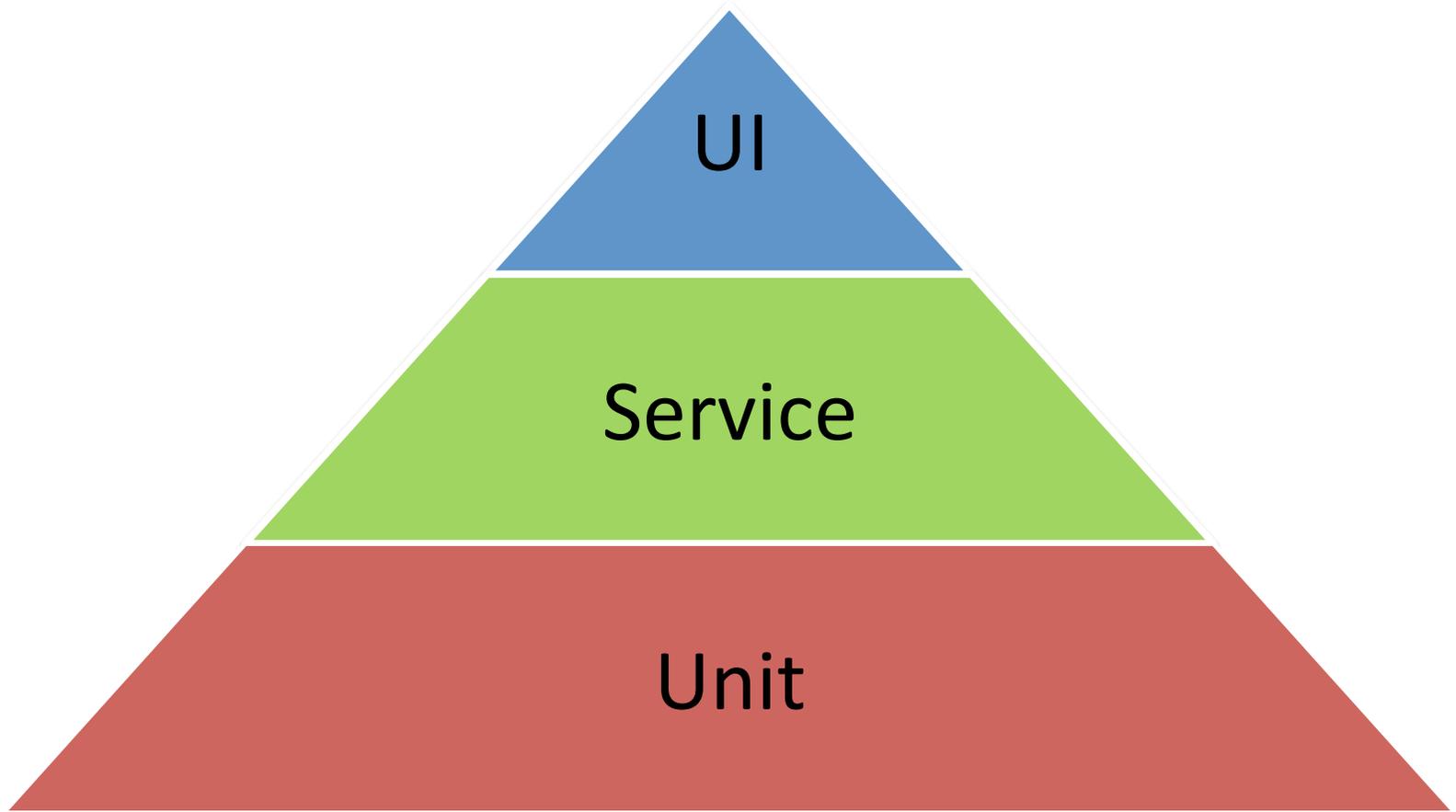
アジャイルテストをプロセス、  
型という観点で考える

# アジャイルテストの四象限



出典:実践アジャイルテスト(翔泳社)

# テスト自動化ピラミッド



「Succeeding With Agile」 Mike Cohn

✓ いつ、どのような品質を達成するか？

✓ どのような粒度のテストを行うか？

✓ いつ、どのような品質を達成するか？

✓ どのような粒度のテストを行うか？

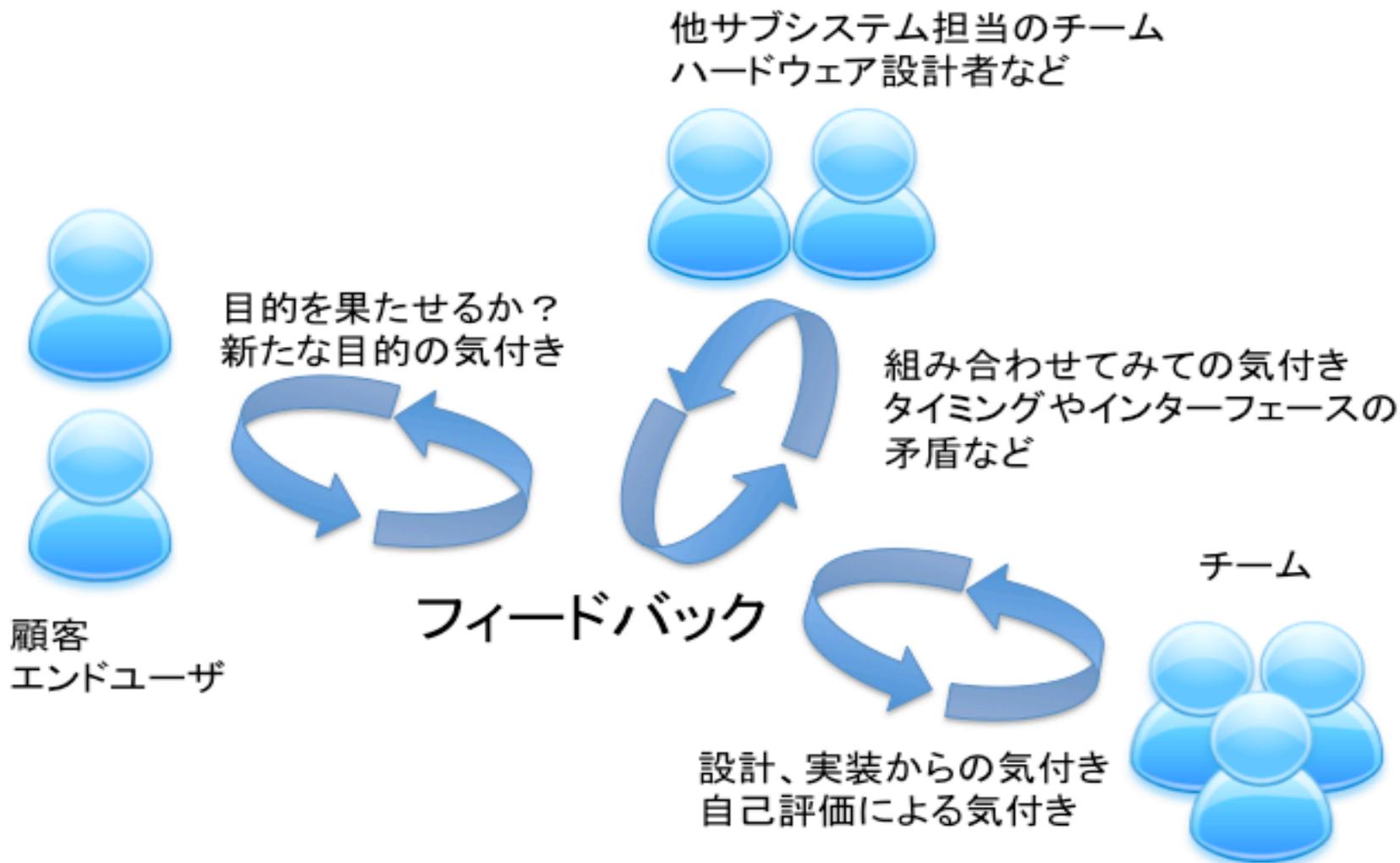
# スプリントの成果物

---

- A) スプリントでの成果物は出荷可能か判断することができる
  
- B) 出荷に必要なプロセスを経て製品を出荷する

プロジェクトによって、Bに必要な作業量は大きく異なる。

# いつ誰からフィードバックを得るか？



## フィードバックの範囲

広い

市場にリリースするための  
作業が大きくなる

市場にリリースするための  
作業を小さくする必要がある

自動化の範囲を可能な範囲  
を広げる必要がある

少ない

多い

リリース回数

市場にリリースするための  
作業が大きくなる

フィードバック対象にリリース  
するための自動化が必要。

市場にリリースするための  
プロセスの自動化が少なくても  
実施可能

狭い

## フィードバックの範囲

広い

市場にリリースするための  
作業が大きくなる

市場にリリースするための  
作業を小さくする必要がある

自動化の範囲を可能な範囲  
を広げる必要がある

少ない

**スクラムのターゲット**

リリース回数

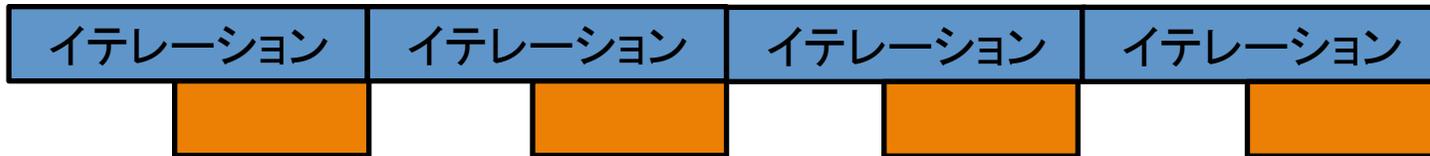
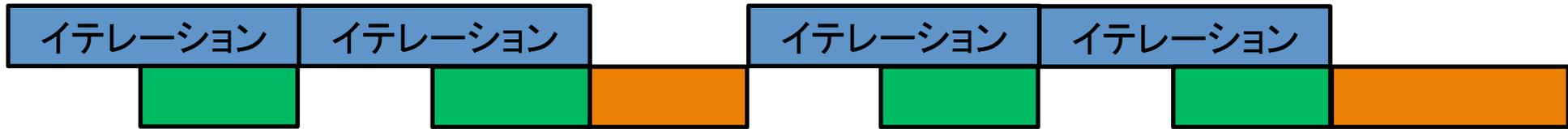
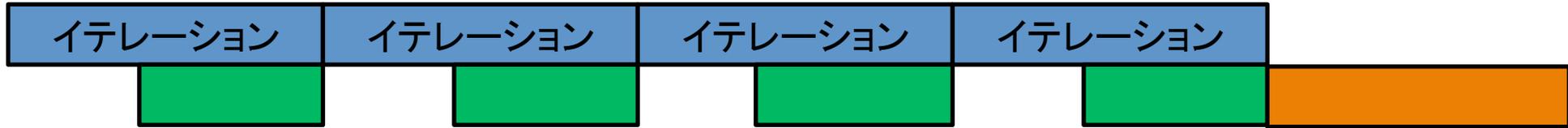
市場にリリースするための  
作業が大きくなる

フィードバック対象にリリース  
するための自動化が必要。

市場にリリースするための  
プロセスの自動化が少なくても  
実施可能

狭い

# どの時点でどのような品質を狙うかを考える



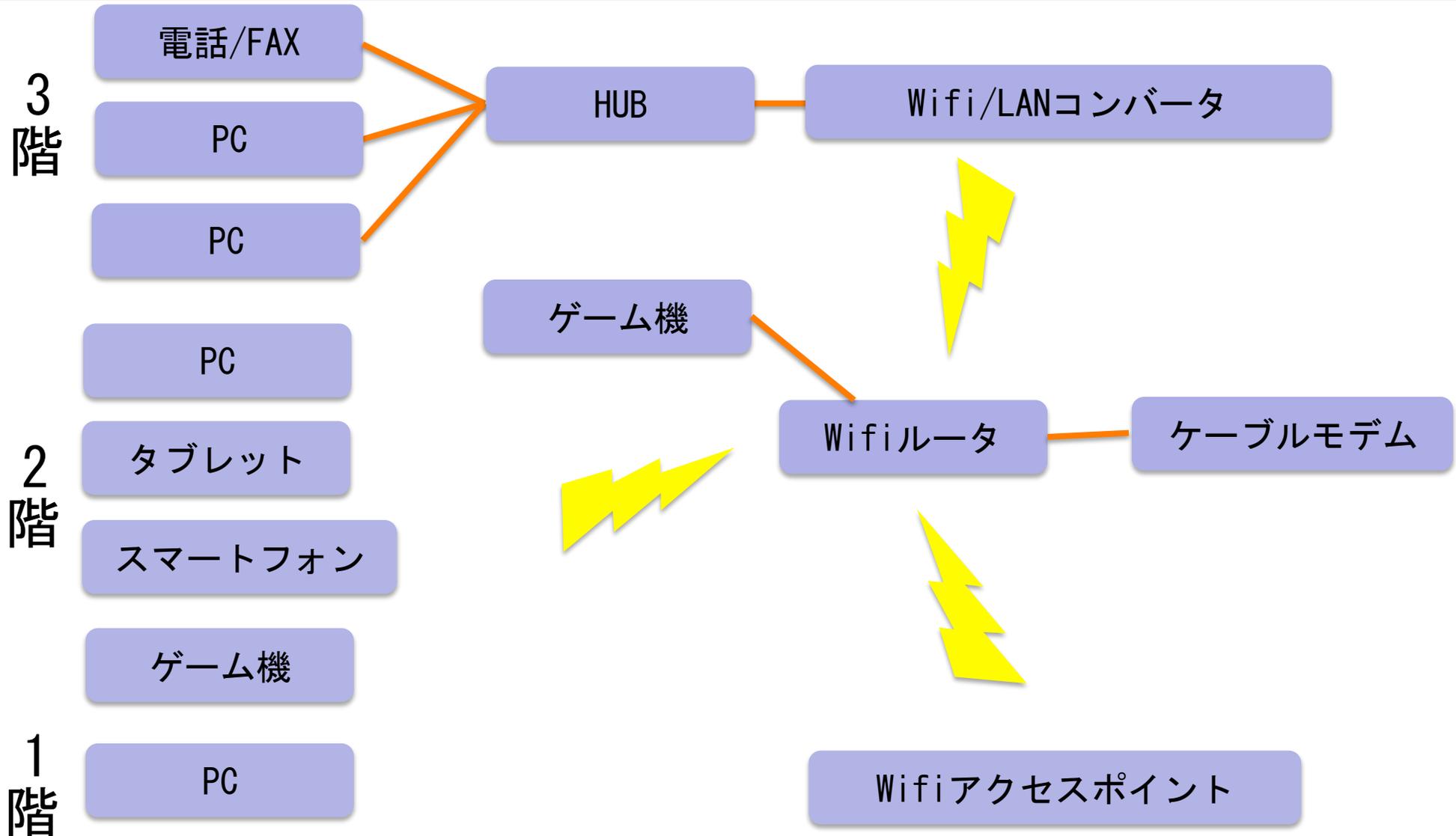
 イテレーション毎のテスト

 出荷のためのテスト

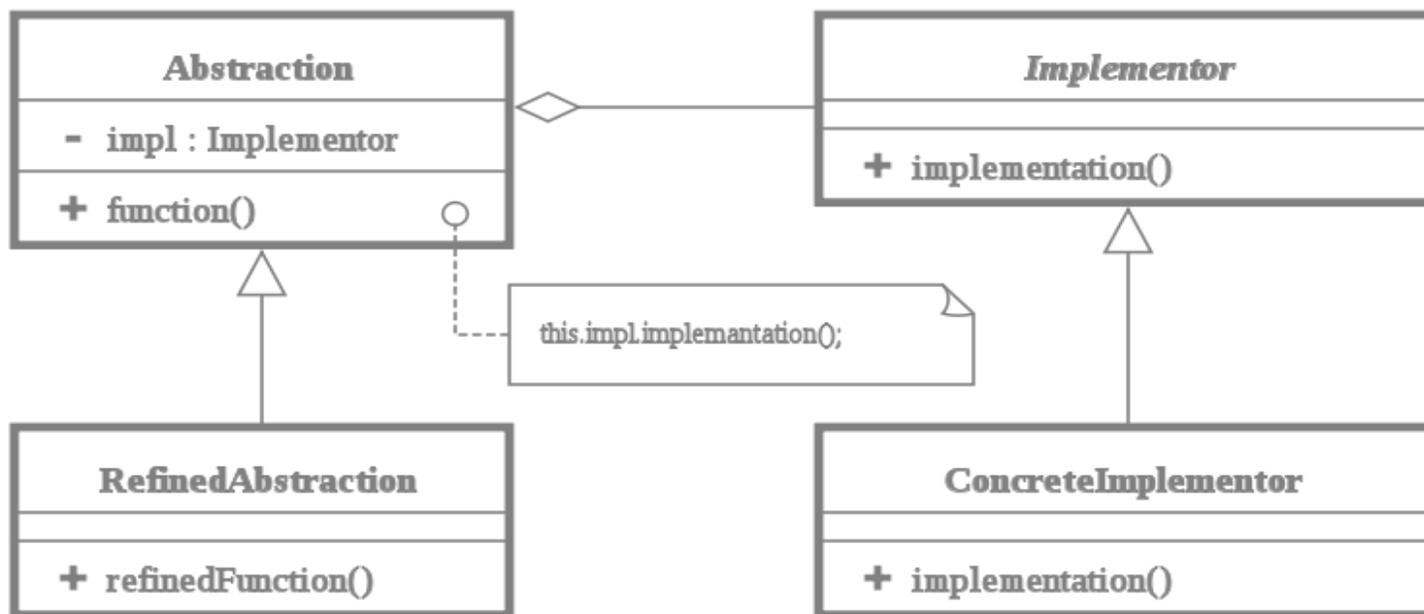
✓ いつ、どのような品質を達成するか？

✓ どのような粒度のテストを行うか？

# 何をテストするか？



ソフトウェアとしても構造や振る舞いなどの色々な要素が組み合わされている



- ✓ 品質の高い要素で全体が構成されていることが重要
- ✓ 小さい粒度での品質を高めていきそれを組み合わせ徐々に粒度を大きくしていく

# テストの粒度を共有する

テストの粒度	機能性	信頼性	使用性	効率性	保守性	移植性
プロジェクト毎に テストの粒度を検討する	どの粒度でどのようなテストを開発するかを検討する					
クラス	自動	レビュー	レビュー	—	レビュー	—
API	自動	レビュー	レビュー	自動	レビュー	—
画面	手動	—	手動	—	—	—
ユーザストーリー	自動/手動	手動	手動	手動	—	—
システム全体	—	手動	手動	手動	—	—
プロトコル	自動	レビュー	レビュー	自動	レビュー	—

テストの粒度	機能性	信頼性	使用性	効率性	保守性	移植性
クラス	自動 TDD	レビュー	レビュー	—	レビュー	—
API	自動 CI	レビュー	レビュー	自動	レビュー	—
画面	手動	—	手動	—	—	—
ユーザストーリー	自動/手動	手動	手動	手動	—	—
システム全体	— 特殊な試験機	手動	手動	手動	—	—
プロトコル	自動	レビュー	レビュー	自動	レビュー	—

アジャイルプラクティスに限らず色々な要素をどう組み合わせるかを考える。

# 型、プロセスの習熟

- ✓ 適切に運営されたフレームワーク  
（レフリングされたゲーム）
- ✓ プロセスの理解、チームでの実行の繰り返し  
（コンビネーション練習、ゲームでの実践）
- ✓ 頭ではなく体に型を馴染ませる  
（基本動作の練習の習慣化）