



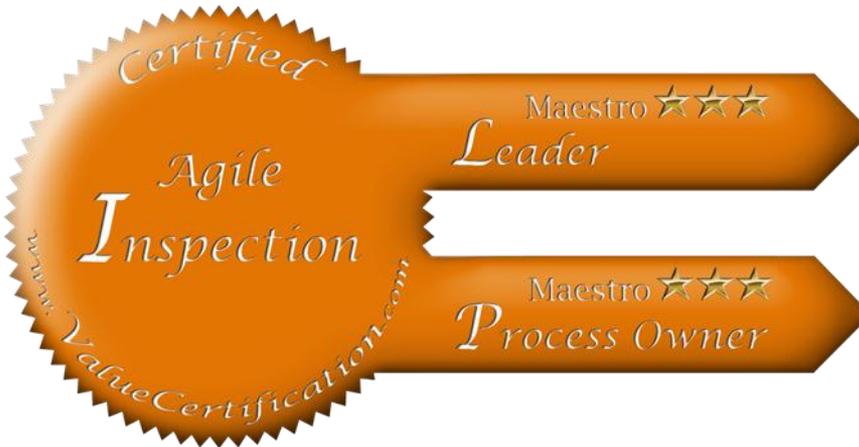
# アジャイル開発における品質保証部門によるシステムテストのアプローチ

アジャイルチームに寄り添う品質保証部門の考え方

# 自己紹介

- ▶ ソニー株式会社 永田 敦  
ソフトウェアテストプロセス改善

SQIP研究会 第三分科会 副主査  
SQIPシンポジウム運営委員  
派生開発推進委員会運営委員



特集 ソフトウェアレビュー、インスペクションと欠陥予防の現在

6 テストエンジニアが参加するアジャイルインスペクション

永田 敦 ソニー (株) B2B ソリューションズ 品質戦略部

品質を良くすることができなければ

アジャイル開発は成功しない

# 課題

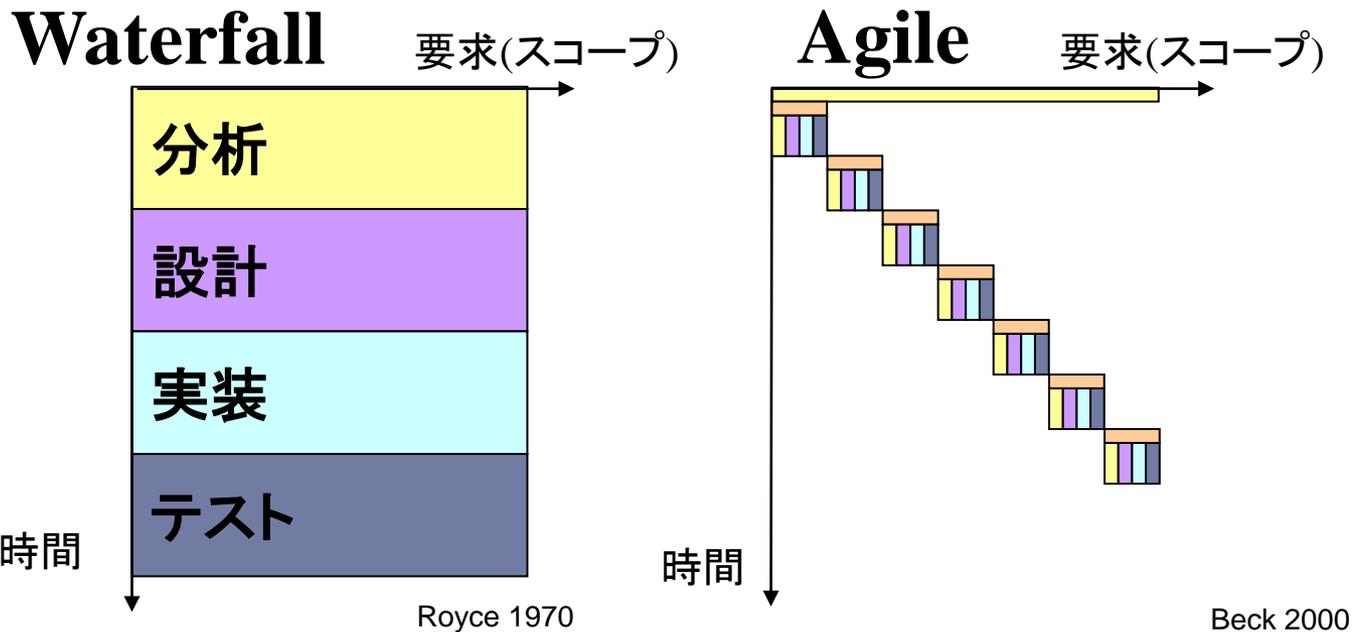
## 今度アジャイルで開発していくから

といわれ

- **プロセスはどうなるのか**
- **評価はどうするのか**
- **品質保証はどうするのか？**

# プロセスとしてのAgile

- ▶ 短いサイクルで、分析、設計、実装、テストを並列に行う
- ▶ タイムボックス型、進化型開発

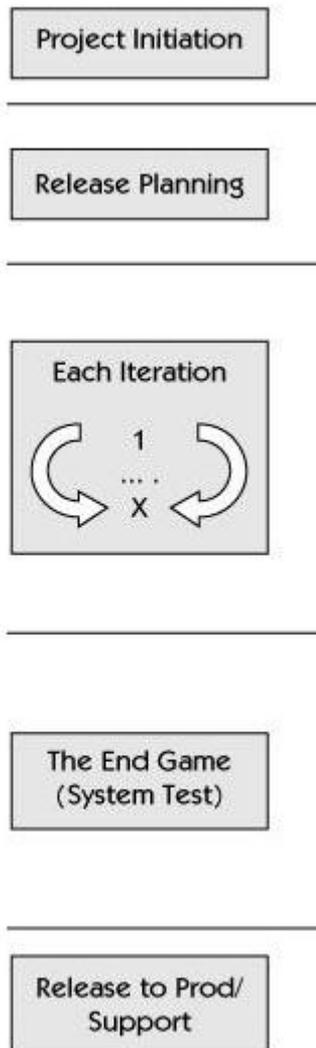


ソフトウェア工学の分岐点における、アジャイルの役割、平鍋 健児、SS2010



**どこから入れればよいか  
いつから入れればよいか**

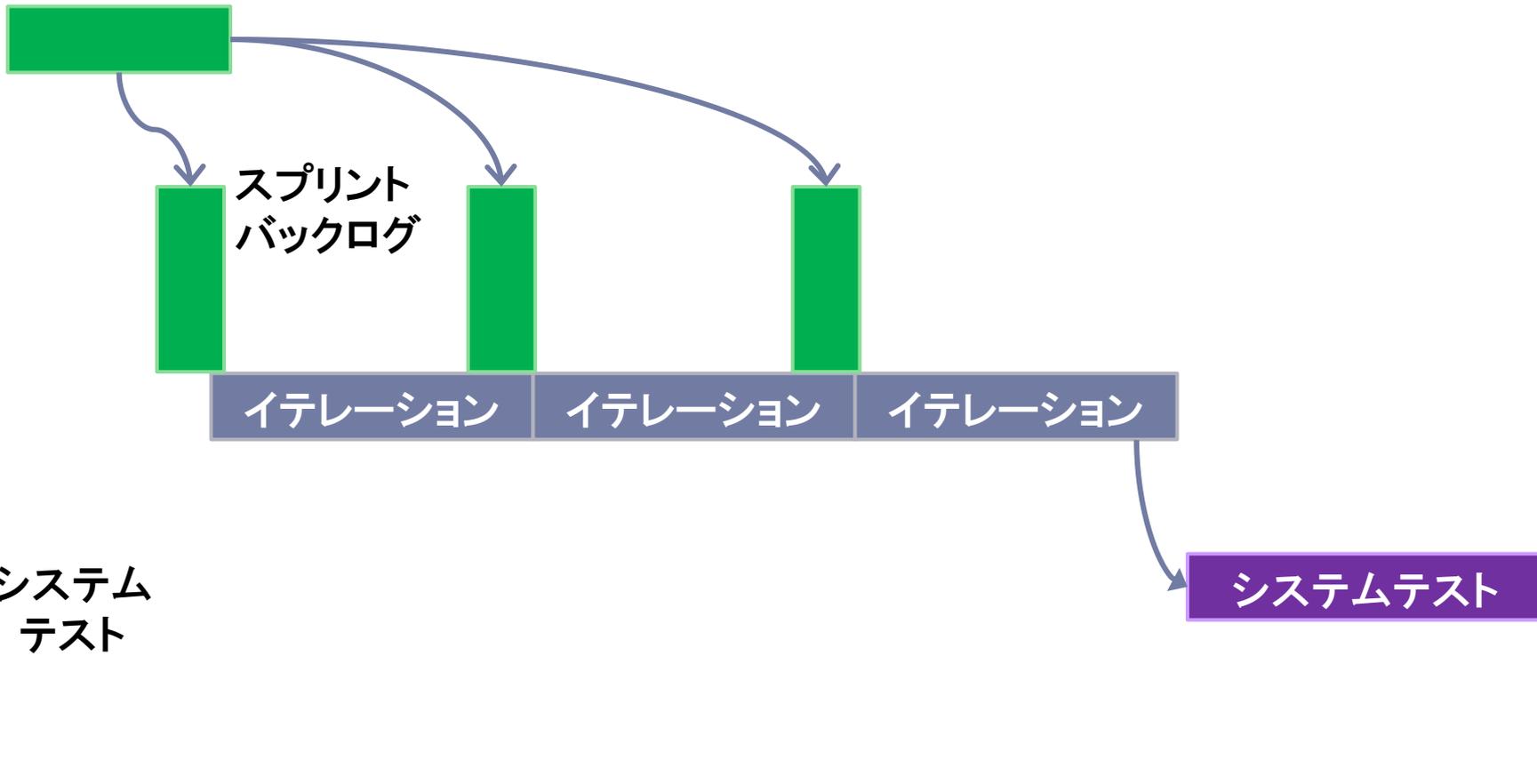
# アジャイル開発のプロセス



Agile Testing, Lisa Crispin, Janet Gregory, 2009

# スクラム

プロダクトバックログ



システム  
テスト

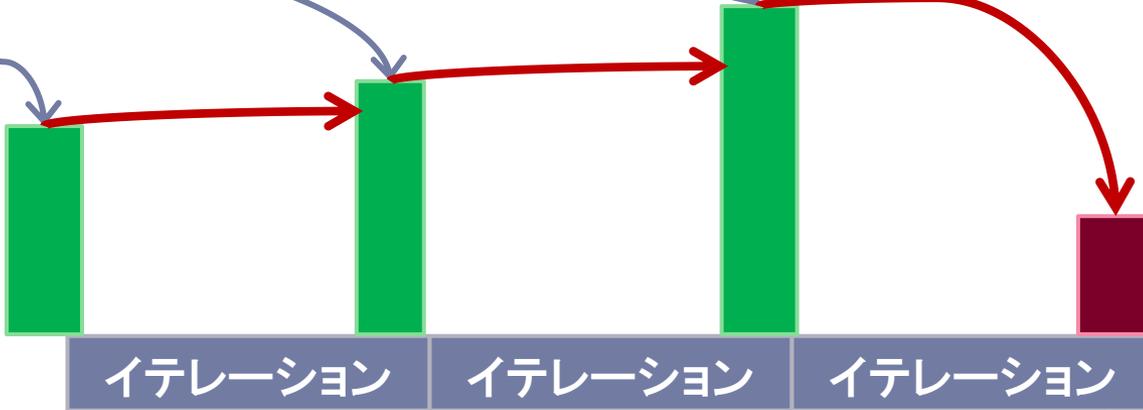
出荷

# バックログインフレーション 1

プロダクトバックログ



スプリント  
バックログ

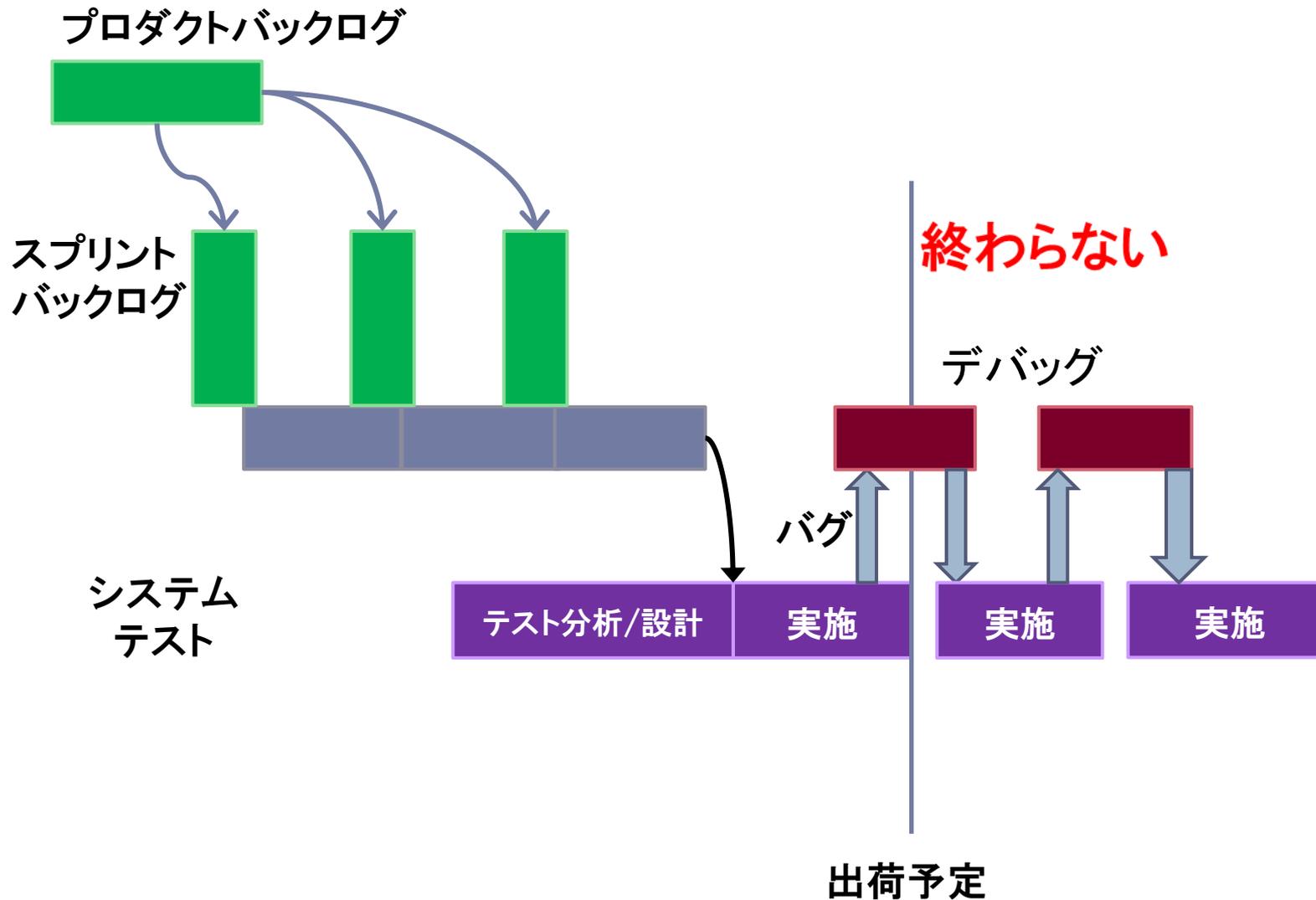


終わらない

システムテスト

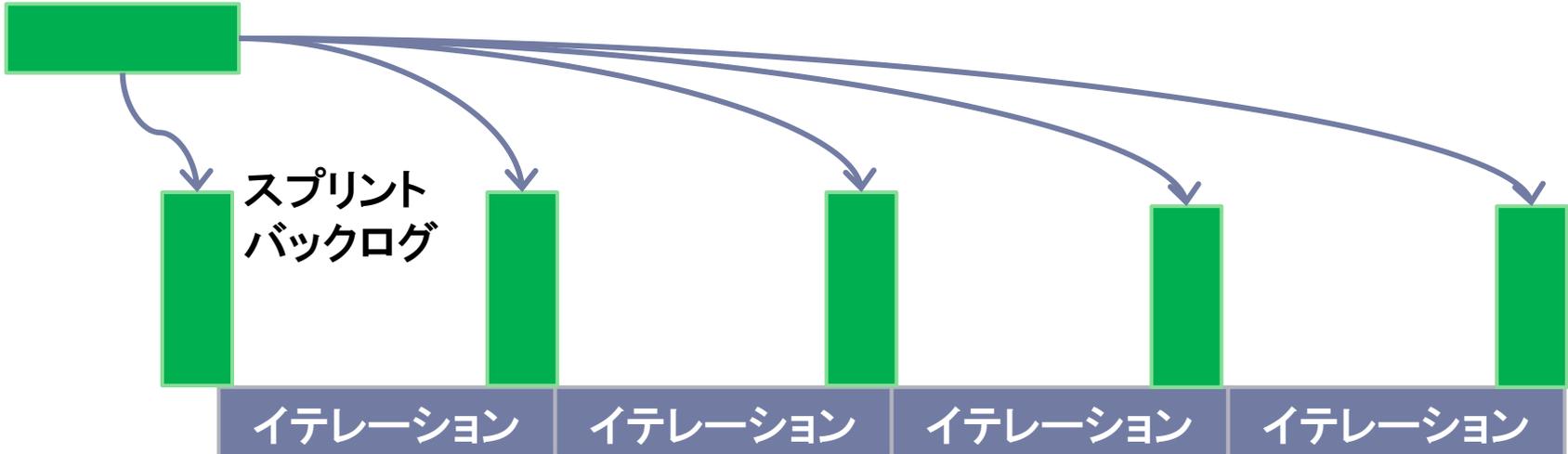
出荷  
予定

# アジャイル? ウォータフォール?



# 早期からのシステムテストの実施

プロダクトバックログ



システム  
テスト





もっと早いタイミングで  
評価しよう

設計フェーズに飛び込む



どのように入ったらよいか  
どのようにやっていったらよいか

# 設計リーダーの憂鬱



QAが設計に入ってくる

いろいろ言われるのではないか

あれを出せこれを出せ  
あれを測れこれを測れ  
あれを直せこれを直せ



QA

固い  
ガード

## 設計の邪魔

# チームの形成

---

マインドセット

設計

QA

品質の  
見える化

サポート

QAはチームとして働く

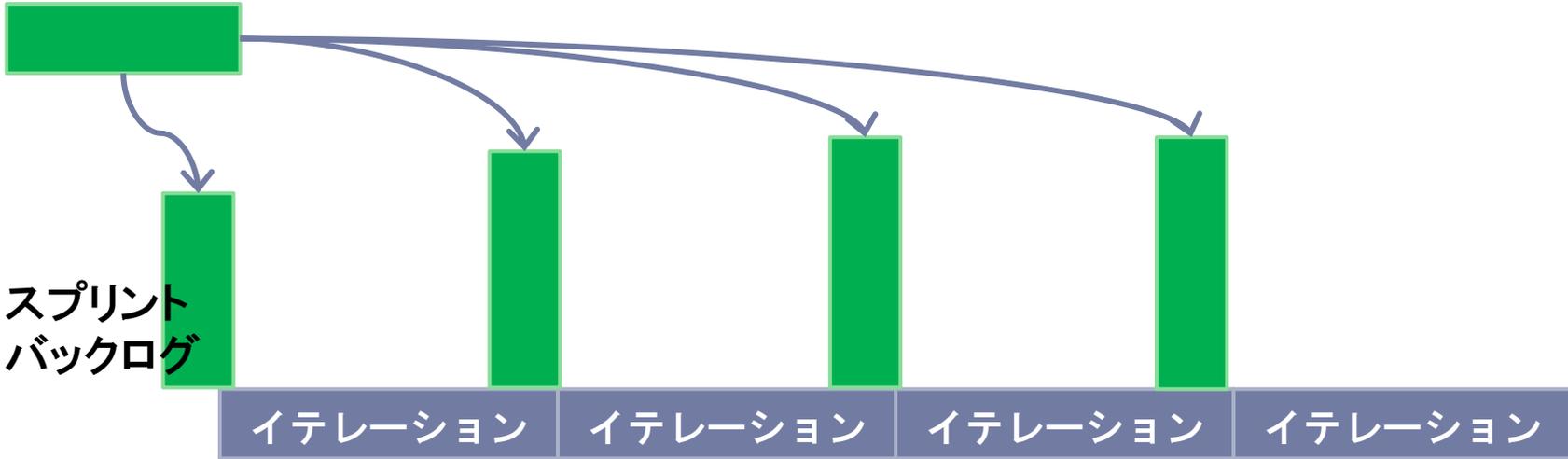
- テスト
  - 動的
  - 静的
- 測定
- フィードバック

設計を邪魔しない ← 恐れを取り除く

設計に気づいてもらう

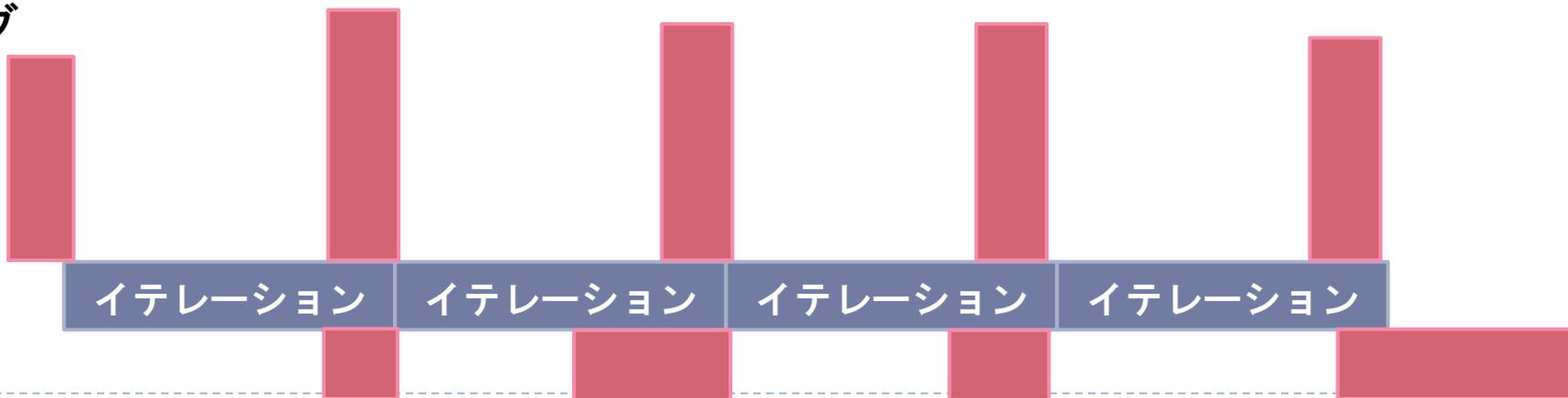
# テストスプリントバックログ

プロダクトバックログ



テストスプリント  
バックログ

システム  
テスト



# テストスプリントバックログ

---

テストケースをバックログとして  
ツールに入れて管理してしまう



設計者に，テストケースをレビューしてもらう



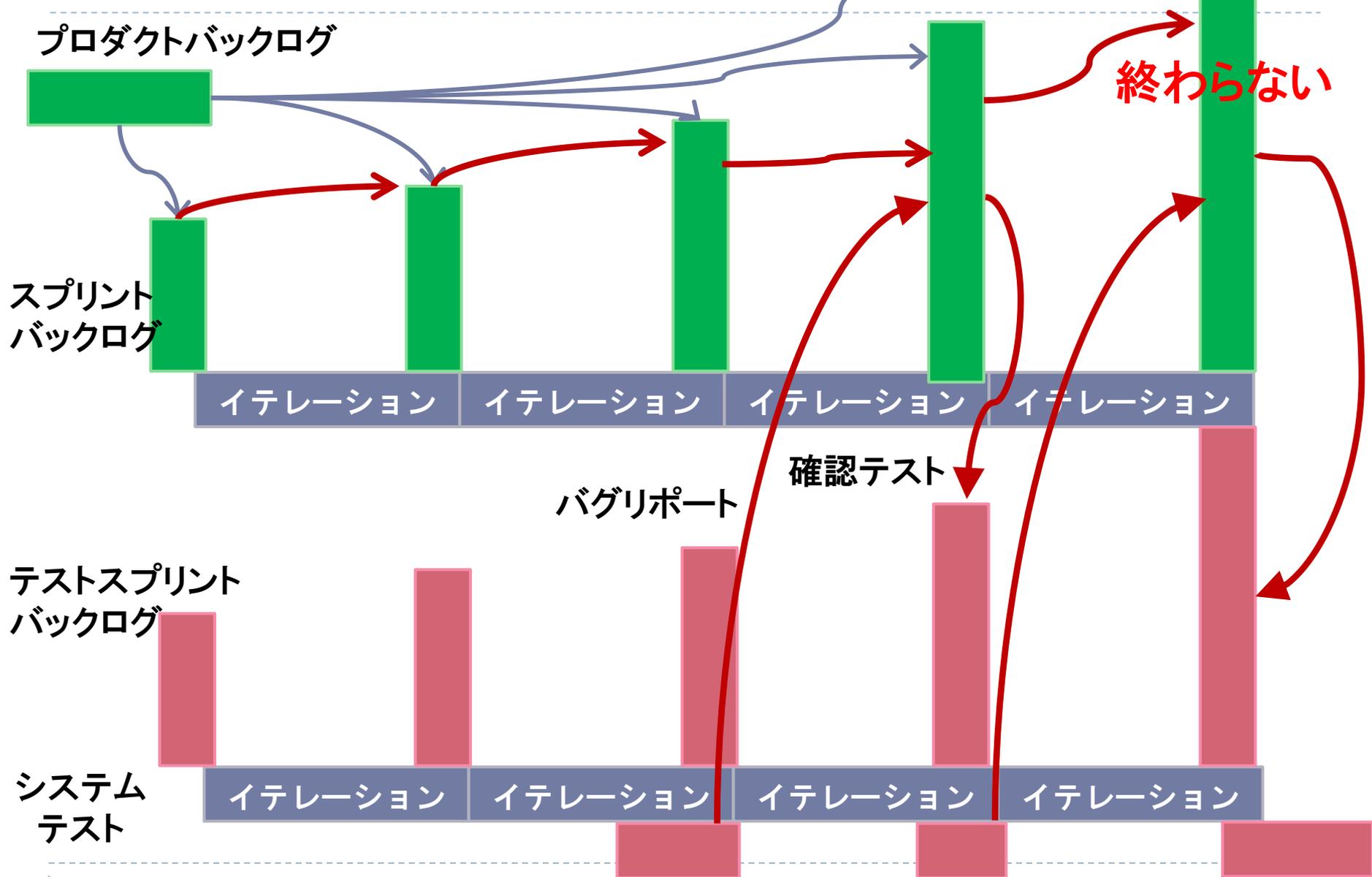
設計者が，何をどのようにテストするかわかる



設計者が，気を付けて設計をする

設計とテストの一元管理

# バックログインフレーション



# テスト

---

- ▶ システムテスト：できるところをテストする
  - ▶ テストに対する計画と戦略が意識される → QA
- ▶ インシデントレポートを，バックログとして登録する
  - ▶ インシデントレポート専用のバックログとプロセスをQAが提案
  - ▶ インシデントの処理も，結局設計が行わなければならない。
- ▶ 原則は，起ったらすぐに設計に見てもらおう
  - ▶ 再現待ちを減らす
  - ▶ 説明も Face to Faceでやる
- ▶ テストの出前

# 測定

---

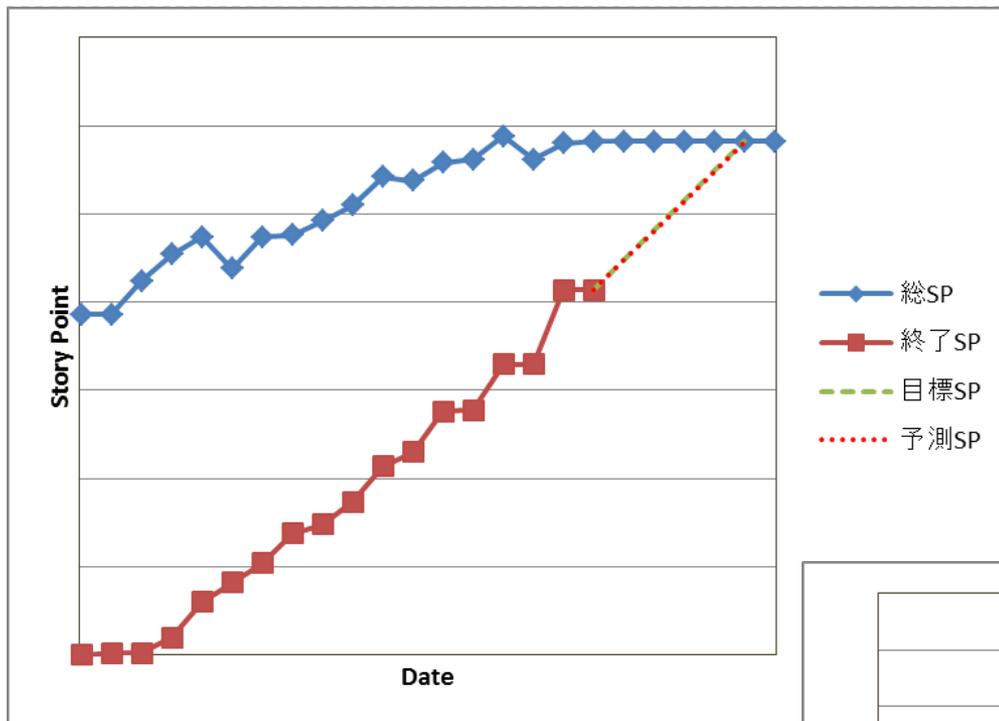
- ▶ ツールからとれるものでQAが測定する
- ▶ 目的： 設計へのフィードバック
- ▶ バックログ情報
  - ▶ プロダクトバックログ
  - ▶ スプリントバックログ
  - ▶ テストスプリントバックログ
  - ▶ インシデントバックログ
- ▶ KPI
  - ▶ Velocity
  - ▶ バグ情報
    - ▶ 発生度数
    - ▶ 残件度数
    - ▶ ランク度数

# QAの見積もりへの参画

---

- ▶ QAがプランニング ポーカーに参加する
- ▶ 効用
  - ▶ バックログの理解
  - ▶ テスト観点からの見積もり
  - ▶ 見積もりのずれにおける説明と議論
    - ▶ 物の作りの理解
    - ▶ テスト観点からのリスクの共有
      - 例外
      - 異常
      - 非機能

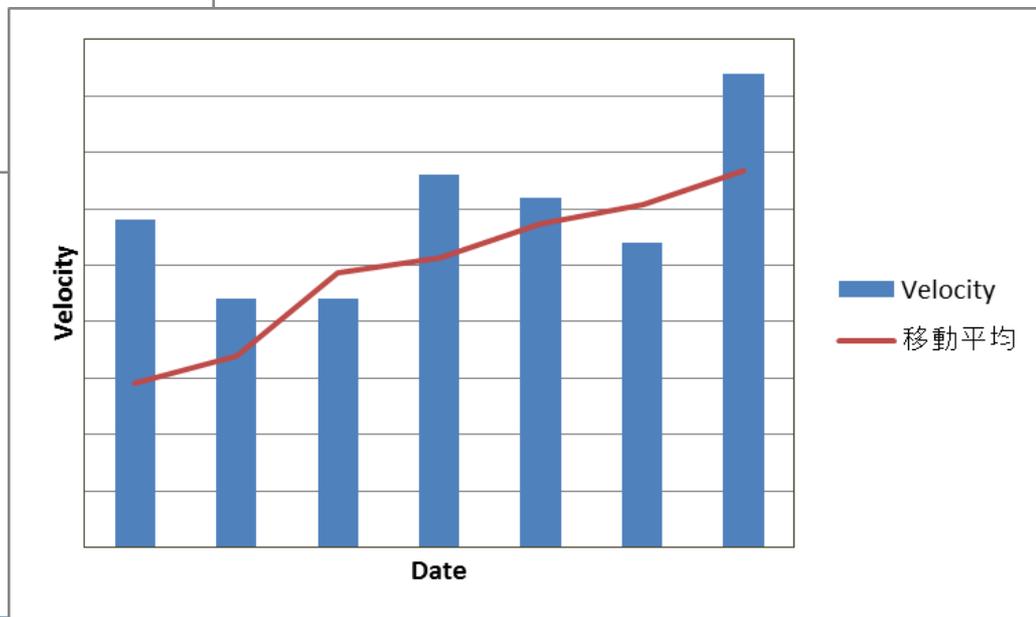
# メトリクス



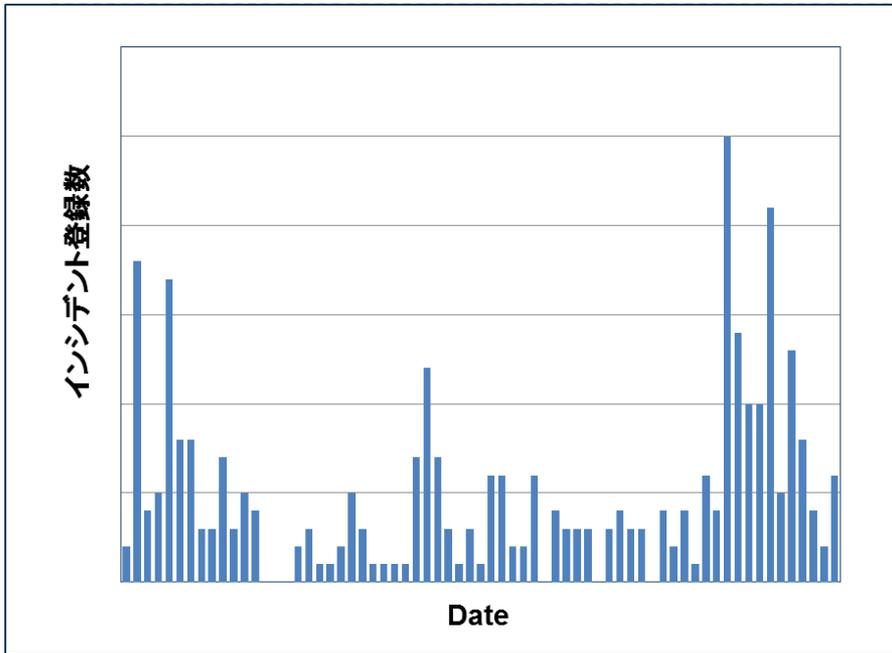
**Velocity**  
**2.2倍の改善**

## バーンアップチャート

**1か月の遅れを改善し  
オンタイム**



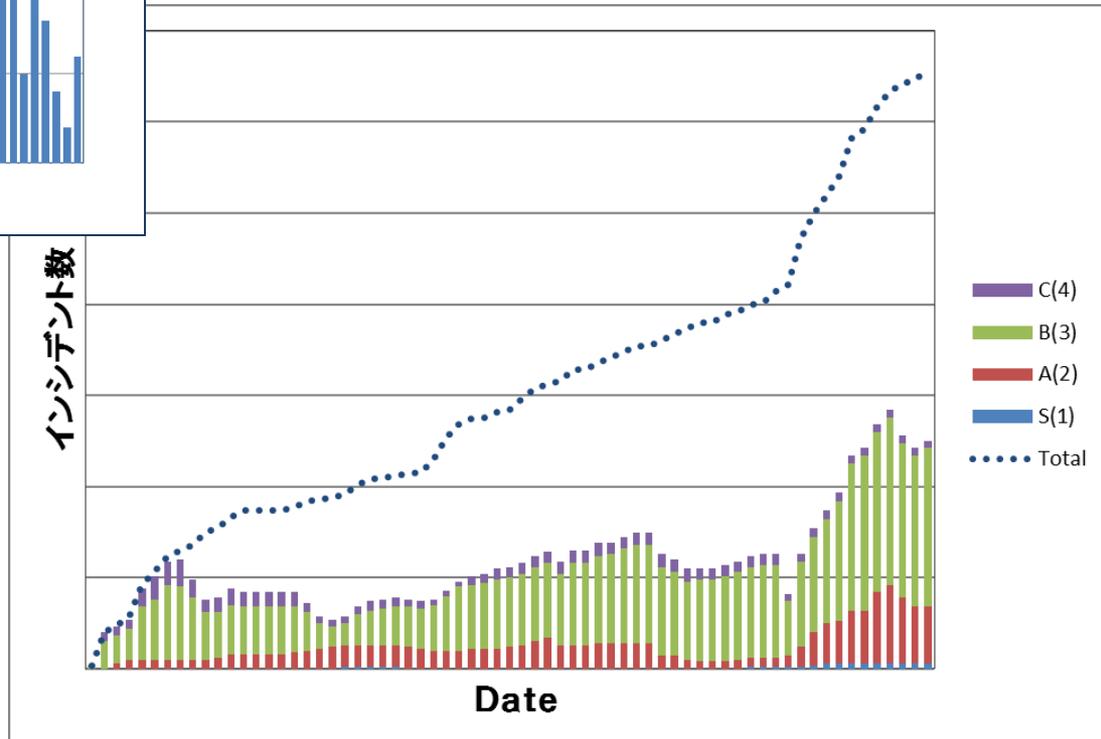
# バグの特性



## インシデント登録度数

計画を持ってやれるところからテスト

Activeなバグ数  
フィードバックによりコントロールされている



# 振り返り

---

- ▶ **メトリクスの吟味**
  - ▶ 緊張感 1か月の遅れ
  - ▶ このままでは間に合わない
  - ▶ 効率を上げるといっても、品質は落とせない
    - ▶ 手を抜けば、インシデントバックログが増える
- ▶ **自ら課題を洗い出す**
  - ▶ わからないことを人に聞いていない。
    - ▶ 3分間ルール
  - ▶ 仕様における意思疎通ができていない。
    - ▶ 仕様説明会
    - ▶ コードレビュー
  - ▶ 実装よりも設計のパワーが不足してVelocityが出ない
    - ▶ チーム編成替え

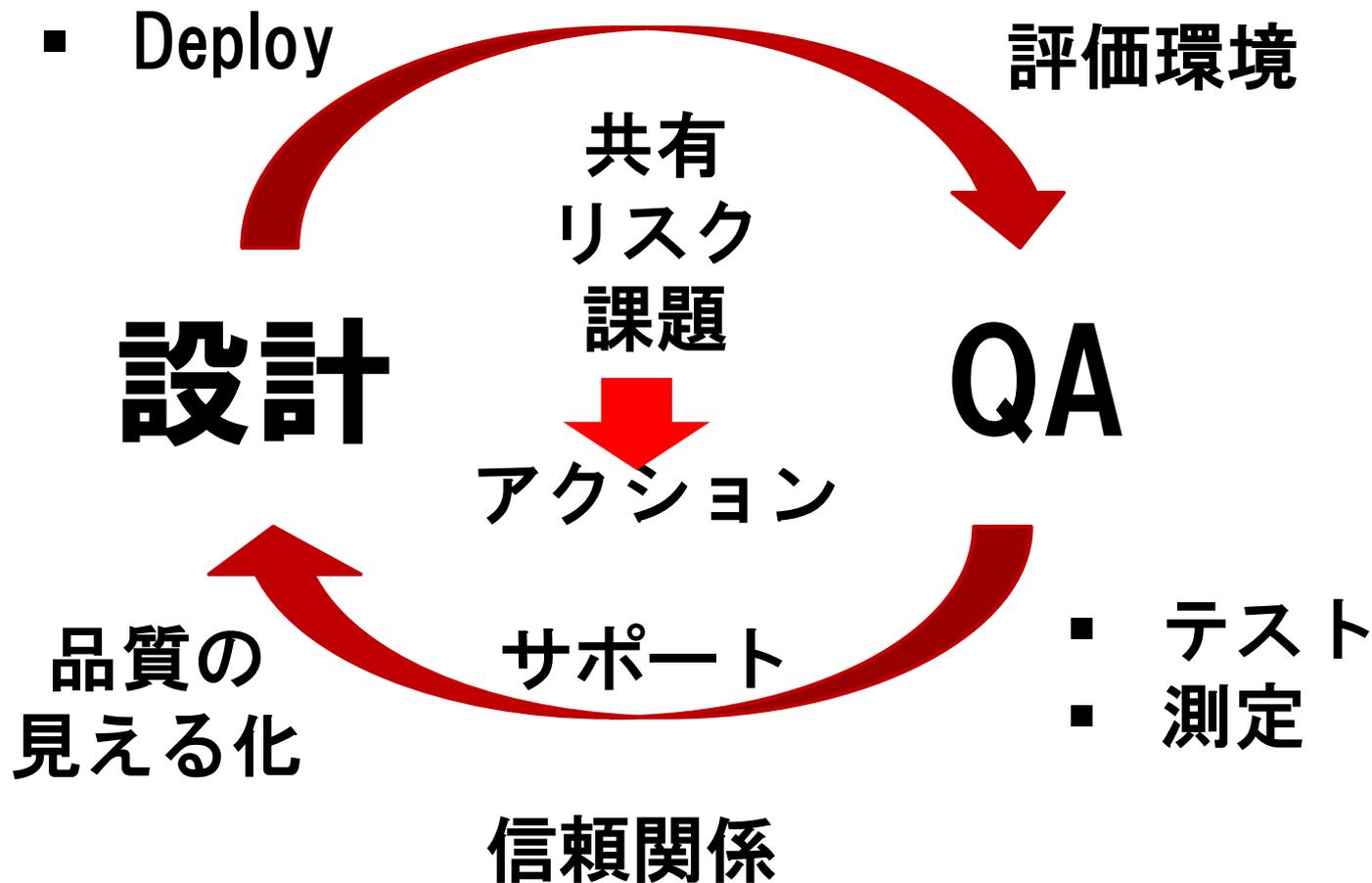
スピード命

3分悩んだら聞くべし

スピード命

聞かれた人もまずは受け入れる！！

# チームの形成



# 変化

---

- ▶ **設計側で内部品質を高める行動が出てきた**
  - ▶ ユニットテストの拡大
  - ▶ 静的テストの積極的活用
  - ▶ 内部のコミュニケーションの改善
  - ▶ ツールの自作
  - ▶ QAにテストの要求をする
  - ▶ コードレビュー
- ▶ **QAからのフィードバックに迅速に反応**
- ▶ **QAに設計に来てもらって見てもらう**
- ▶ **信頼関係の確立**

# 成果

---

- ▶ **Velocityの改善と安定**
  - ▶ 施策の効果がメトリクスに現れる
  - ▶ 7イテレーションで移動平均値が2.2倍となる
- ▶ **バグ傾向のフィードバック**
  - ▶ 軽微なバグが減ってくる
    - ▶ GUIの境界値バグがなくなる
  - ▶ ランクの高いバグから減らそうとする
  - ▶ インシデントバックログを増やさない
- ▶ **スケジュールの改善**
  - ▶ 最大1か月遅れをオンタイムにすることに寄与した

**フィードバック，振り返りの重要性の認識**

## 成果 2

---

- ▶ 施策の効果が確認できる。
- ▶ プロジェクトリスクを明確に共有することができている
- ▶ やれるところ，やれないところを明確にできた。
- ▶ 品質を落とさずに，スケジュールをキープできるようになってきた。
- ▶ 品質に対するリスクを明確に認識している

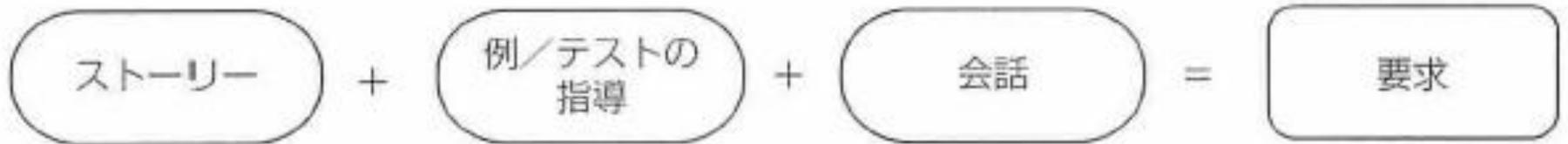
# 考察

---

- ▶ **今回は，QAが初めから要求仕様を押さえていた**
  - ▶ QAは今回のシステムのオラクルになっていた。
  - ▶ したがって，実装が始まるころにはほとんどのテストケースが用意できている状態であった。
  - ▶ これにより，スプリントバックログに従って設計に対して効果的なフィードバックができた。
  - ▶ つまり，QAがオラクルになるほどの，要求仕様が用意できていれば，QAがアジャイル開発において品質を改善することができる。
  - ▶ 現在のユーザーストーリーの質では不十分である。
- ▶ **アジャイルの要求の品質が今後の課題となる**

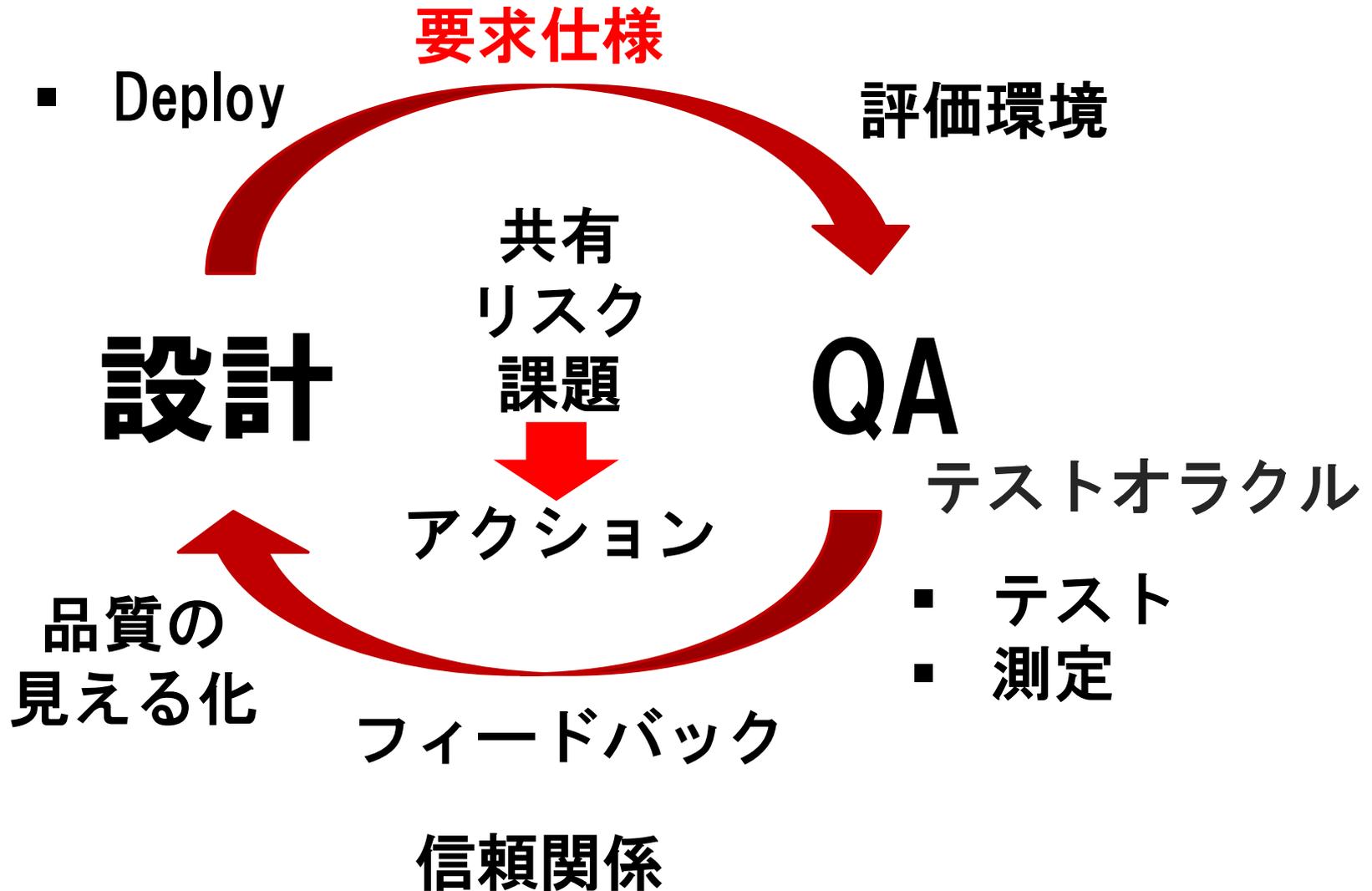
# アジャイルテストティングにおける テストベースについて

## ユーザストーリーと要求の関係



Agile Testing, Lisa Crispin, Janet Gregory, 2009

# 理想のアジャイル



品質を良くすることができれば

アジャイル開発は成功する

私たちだけでなく、設計チーム、QAチーム、マネージャが感じていること

**ご清聴ありがとうございます**