



OODAと品質保証

アジャイル開発における品質管理のアプローチ

自己紹介



永田 敦
日新システムズ
品質保証部
サイボウズ
技術顧問



ソフトウェアテスト改善
ソフトウェアプロセス改善
アジャイルソフトウェア開発改善サポートおよびコーチング

JSTQB Advanced Level Test Manager
Agile Inspection Maestro

SQIP 研究会第4分科会主査
派生開発推進協議会運営委員

アジャイルの流儀 EVO



アジャイル開発とは(品証部門から見て)

W/F 開発で養った品質保証のアプローチをそのまま適用出来ない

● プロセスQA視点

・品質監査観点

- プロセス準拠性チェック
- 標準準拠性チェック
- ゲート監査

・品質管理観点 (指標値の管理)

- レビュー時間、指摘件数
- 工程別テスト項目数
- 工程別バグ摘出件数予実績ほか

監査? いつやるの
反復毎じゃないよね?

標準準拠?
毎回ふりかえりで改
善してるけど...

何のレビュー時間?
いつも会話してるけど...

テスト項目数? バグ件数?
テストコード流しながら作ってるけど
どの時点の件数?

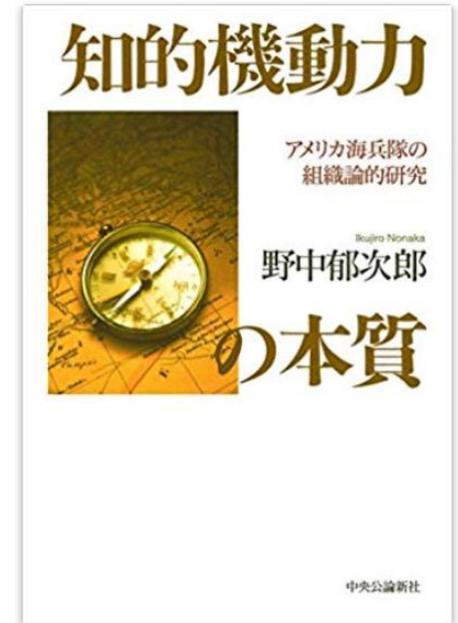
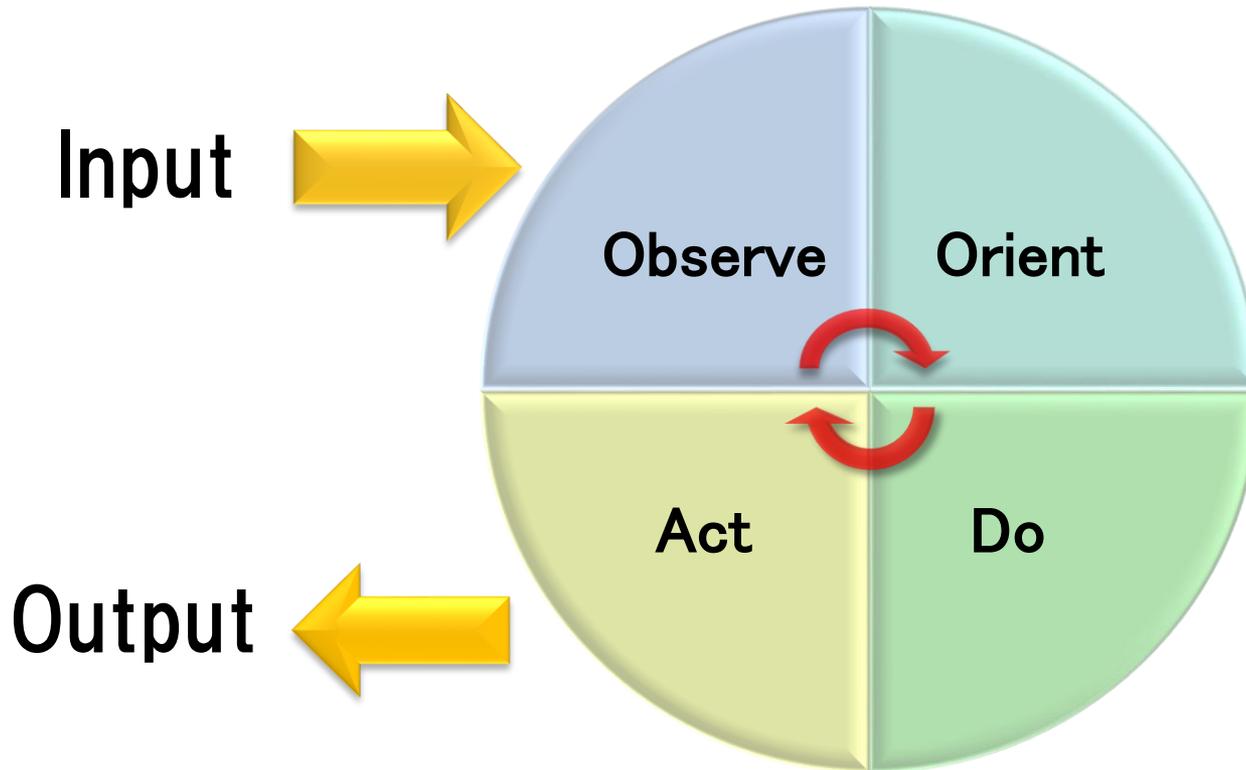
要件ってまだ全部はきまって
ないよ。
バックログはあるけど...

設計書は最低限しかないよ
動かしてみてね: p

● 成果物評価・テストQA視点

- ・要件の確度、計画リスク評価
- ・設計書類の記述完成度評価(レビュー)
- ・プログラム動作検証

OODA ループ



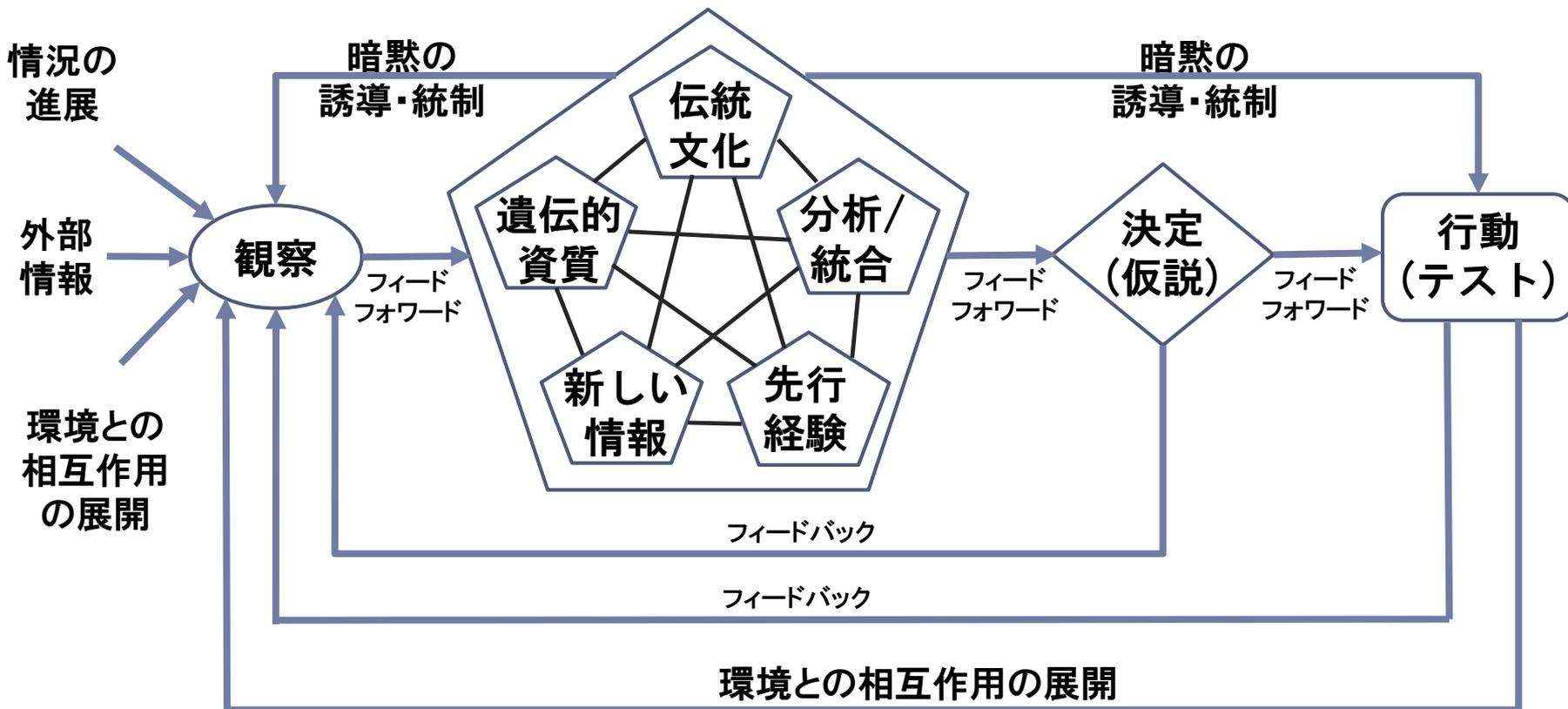
OODAループ

観察
(Observe)

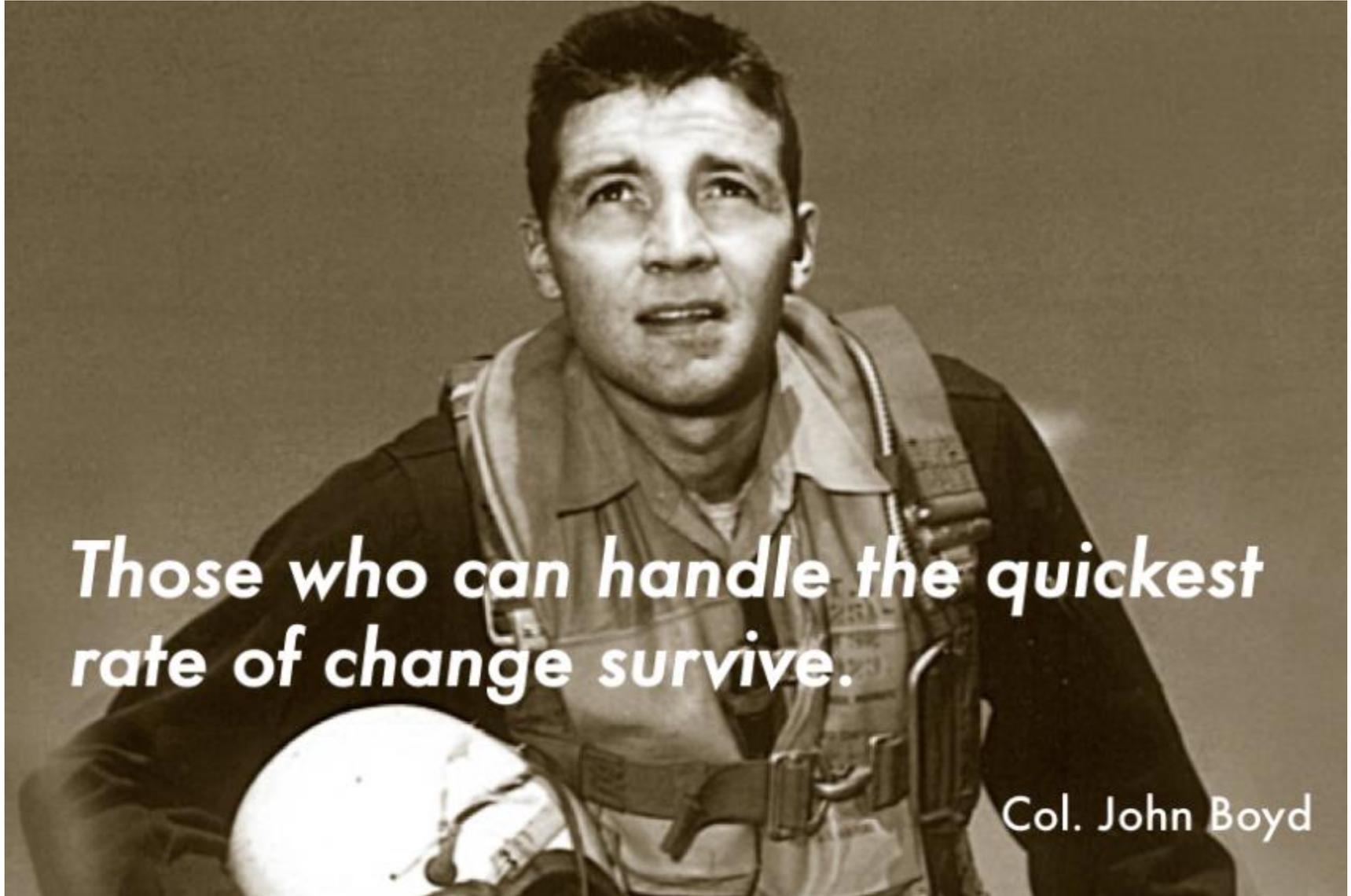
情勢判断
(Orient)

意思決定
(Decide)

行動
(Act)



John Boyd

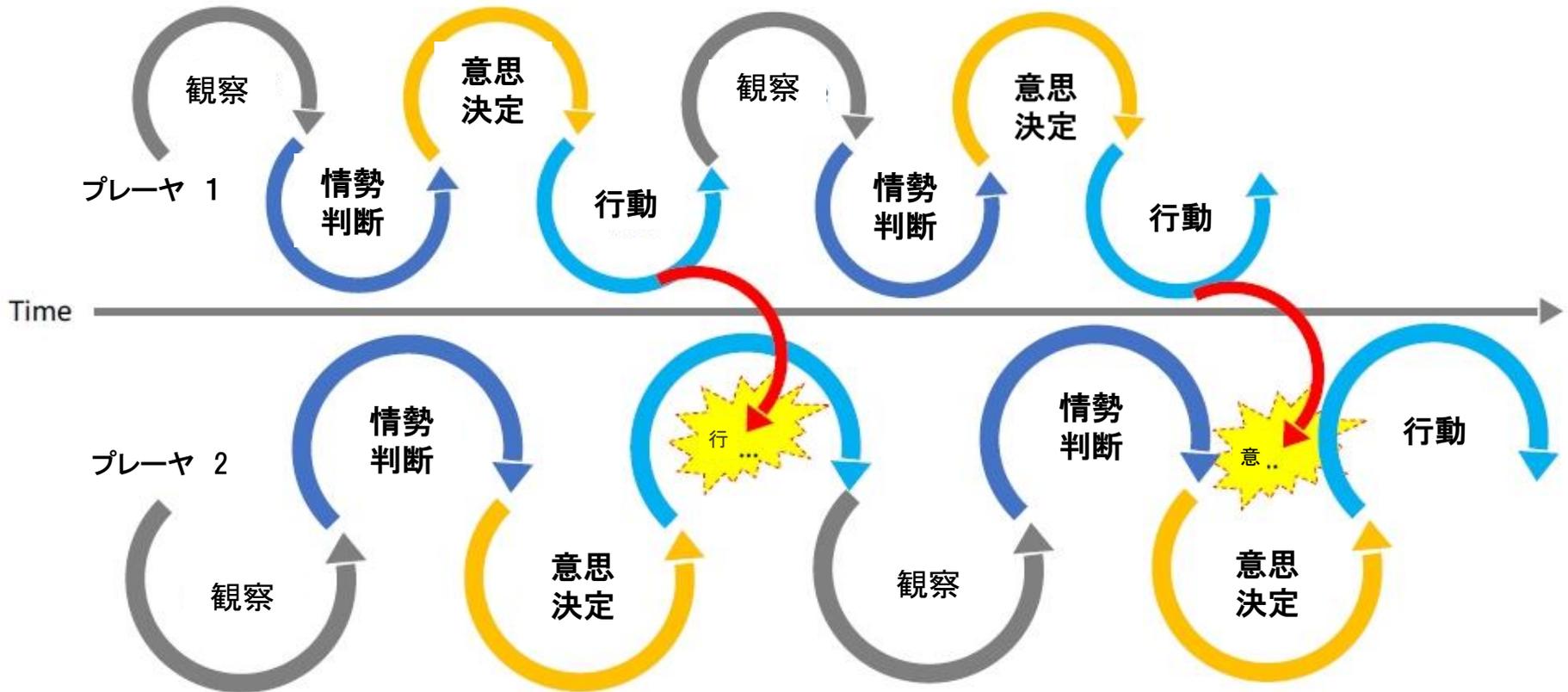


*Those who can handle the quickest
rate of change survive.*

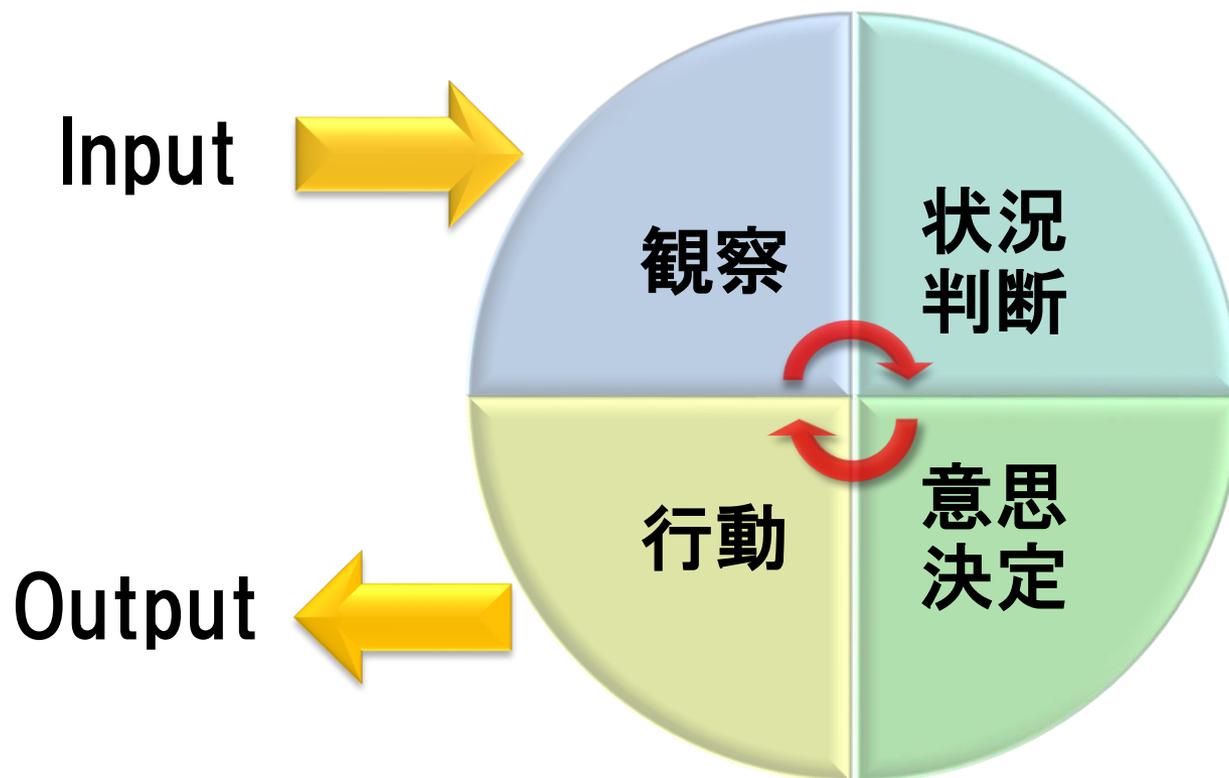
Col. John Boyd

戦局を左右するのは情報量と意思決定のスピードである

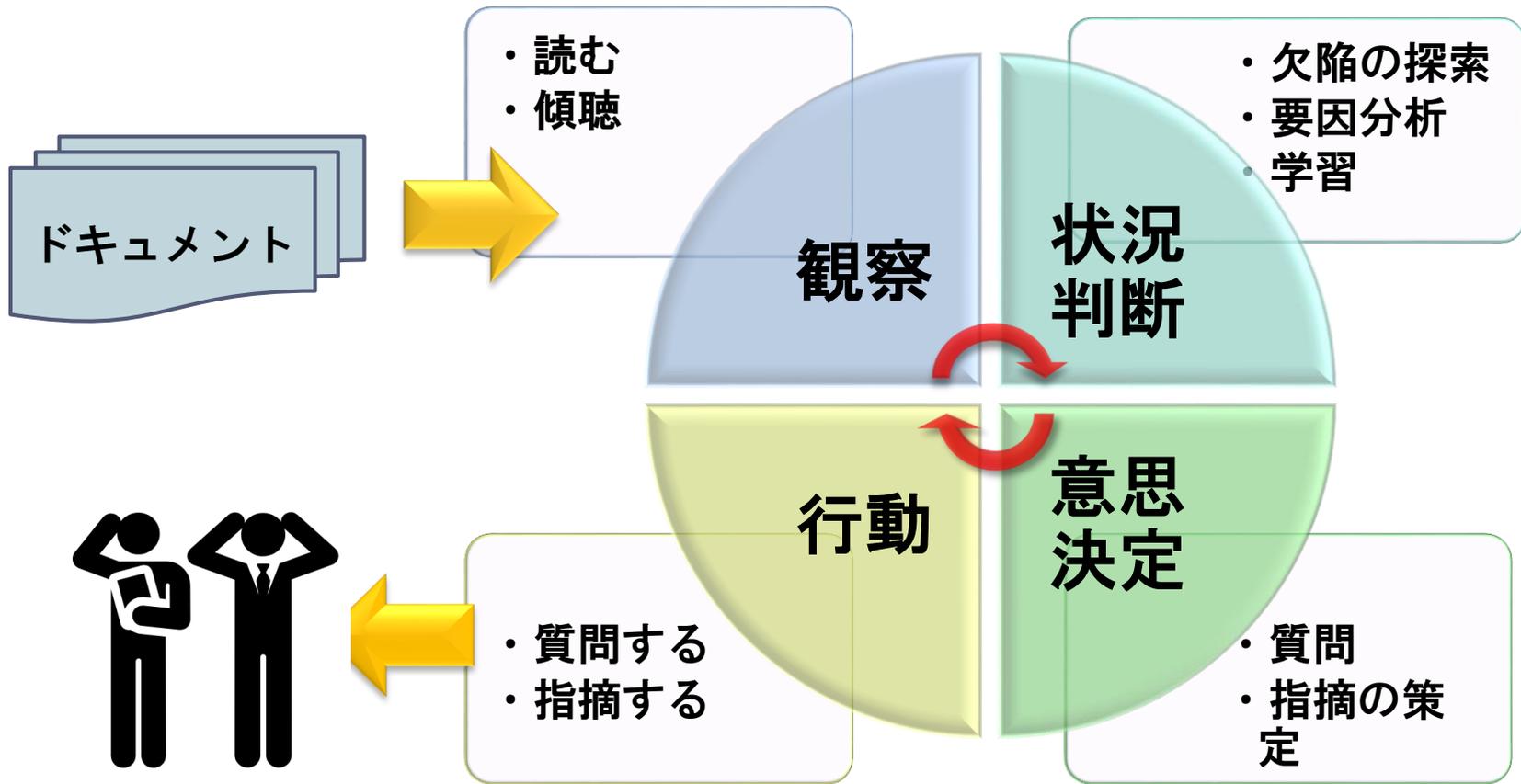
相手のループの内側に入る…



OODA ループ



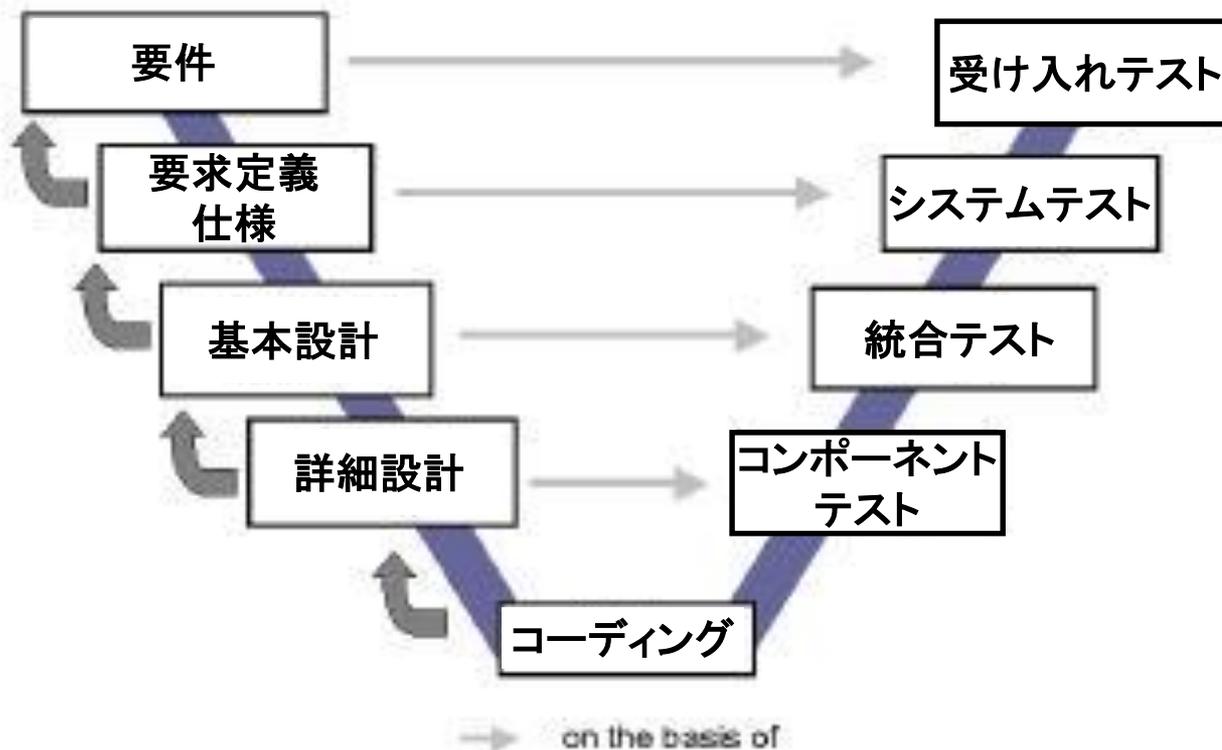
レビュー



プロダクト品質 : テスト

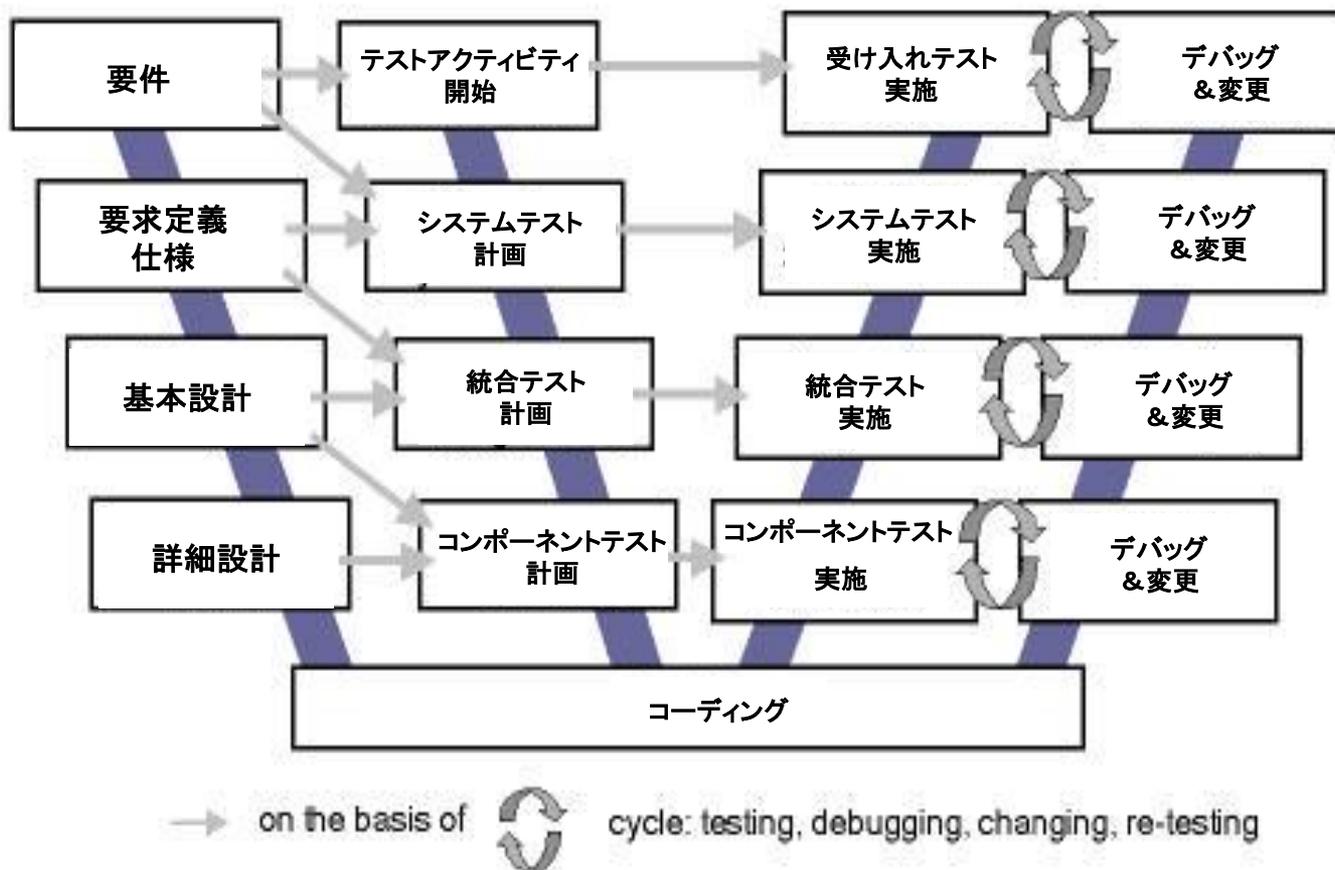
ソフトウェア開発モデル V字モデル

V-Model



ソフトウェア開発プロセス：ダブルV字モデル

W - Model



アジャイル開発のプロセス

Project Initiation

Release Planning

Each Iteration



The End Game
(System Test)

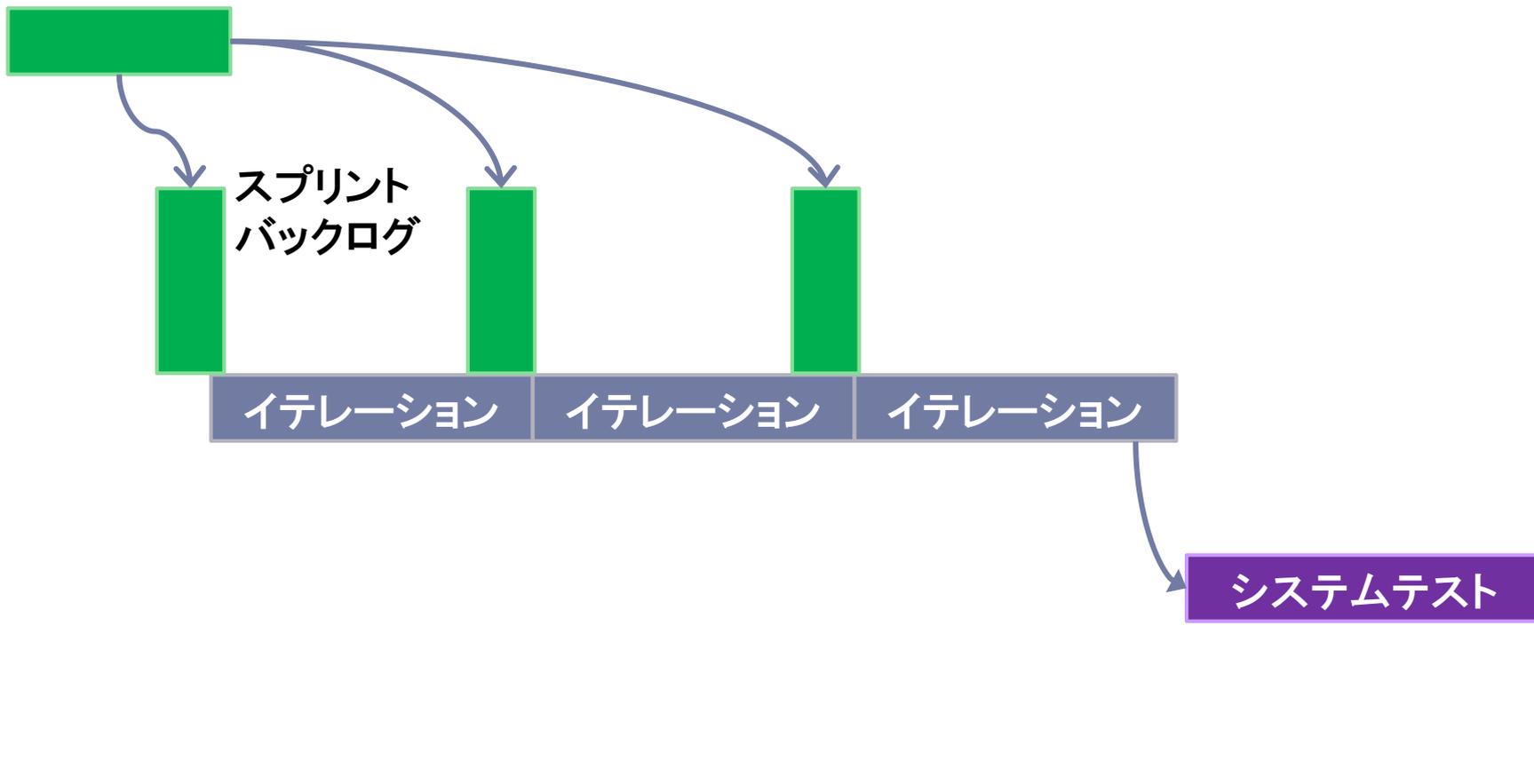
Release to Prod/
Support



Agile Testing, Lisa Crispin, Janet Gregory, 2009

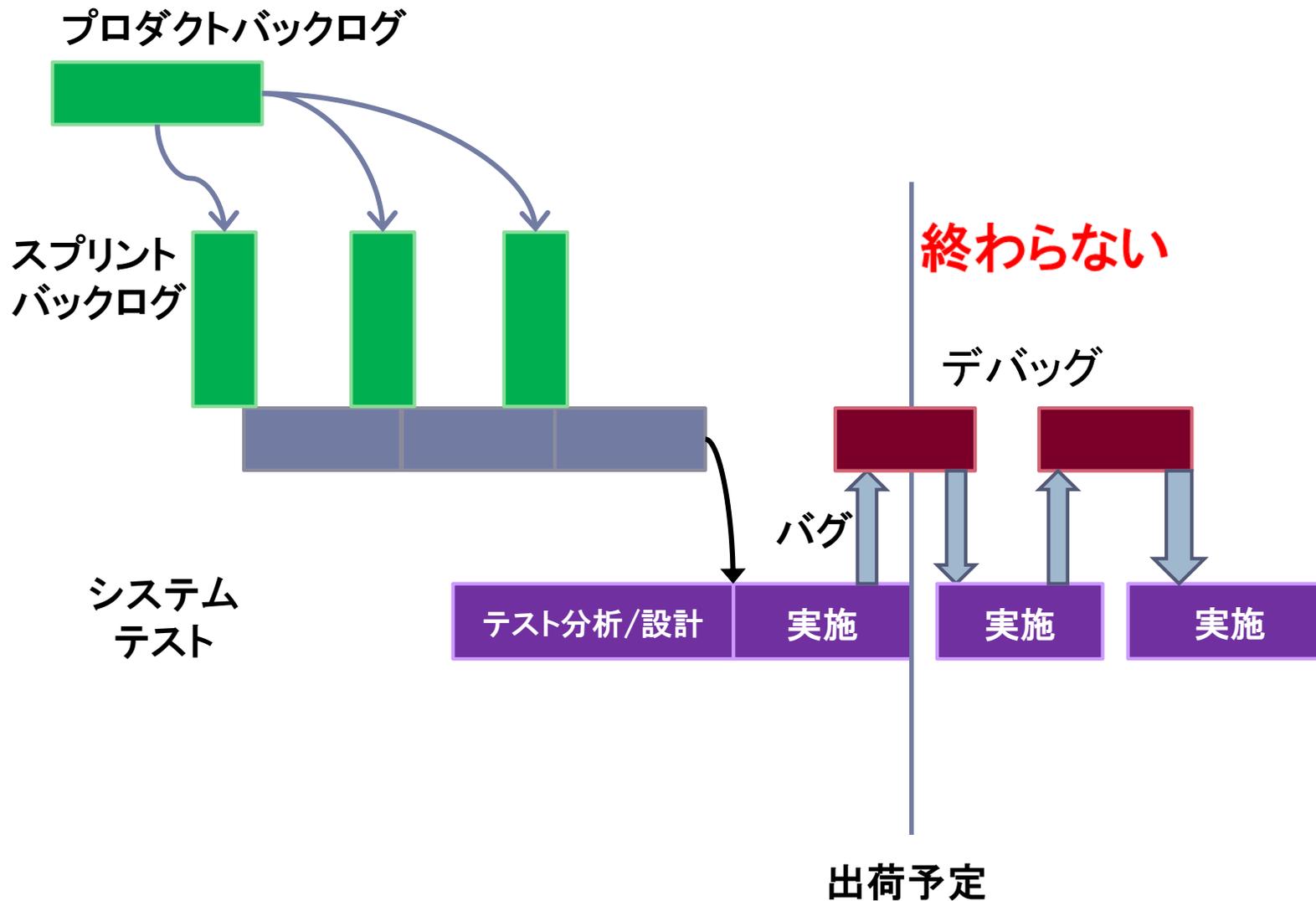
スクラム

プロダクトバックログ



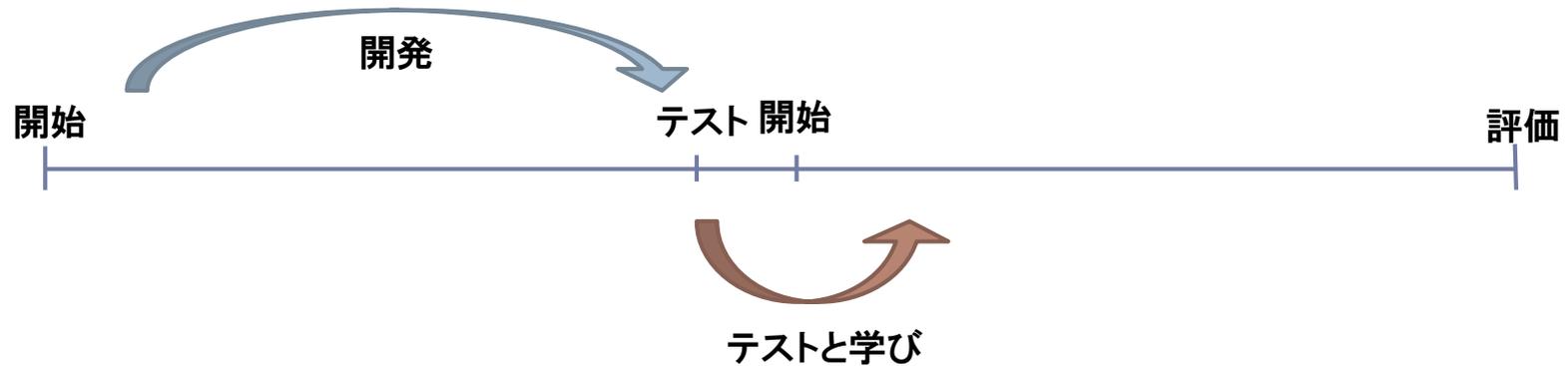
出荷

手遅れ

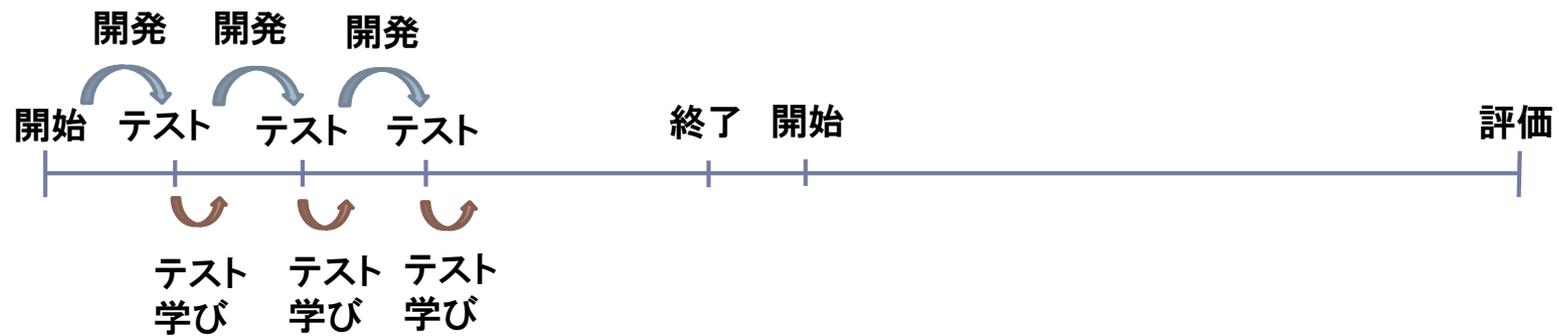


テストをOODAする

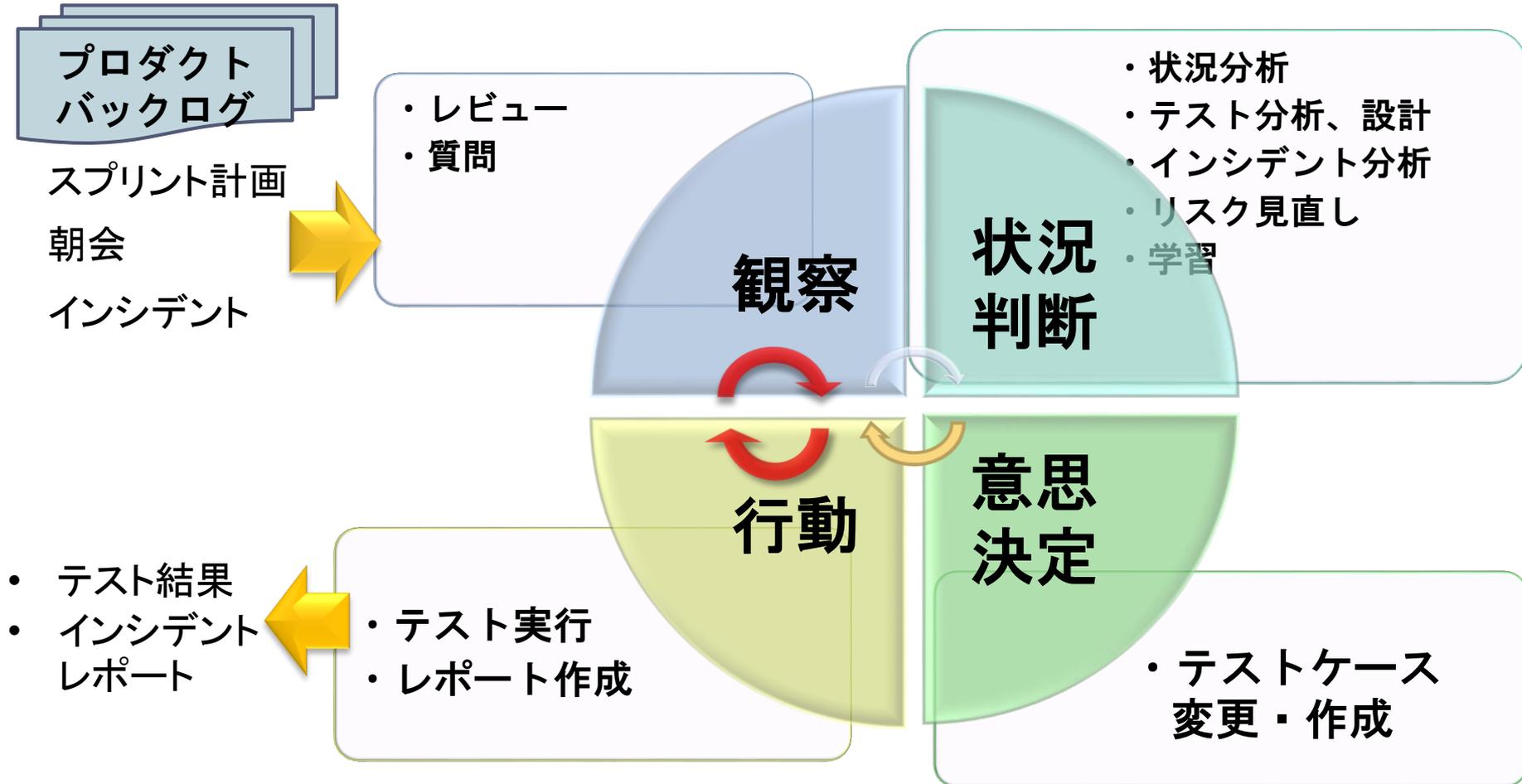
ウォーターフォール



アジャイル

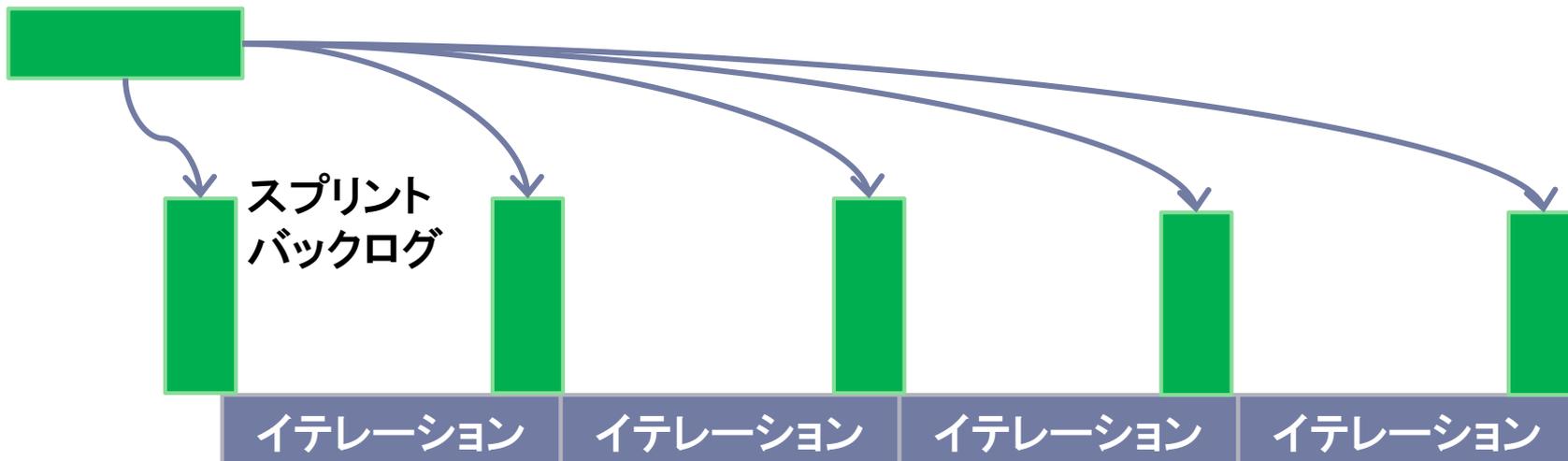


テスト



早期からのシステムテストの実施 QAの介入

プロダクトバックログ

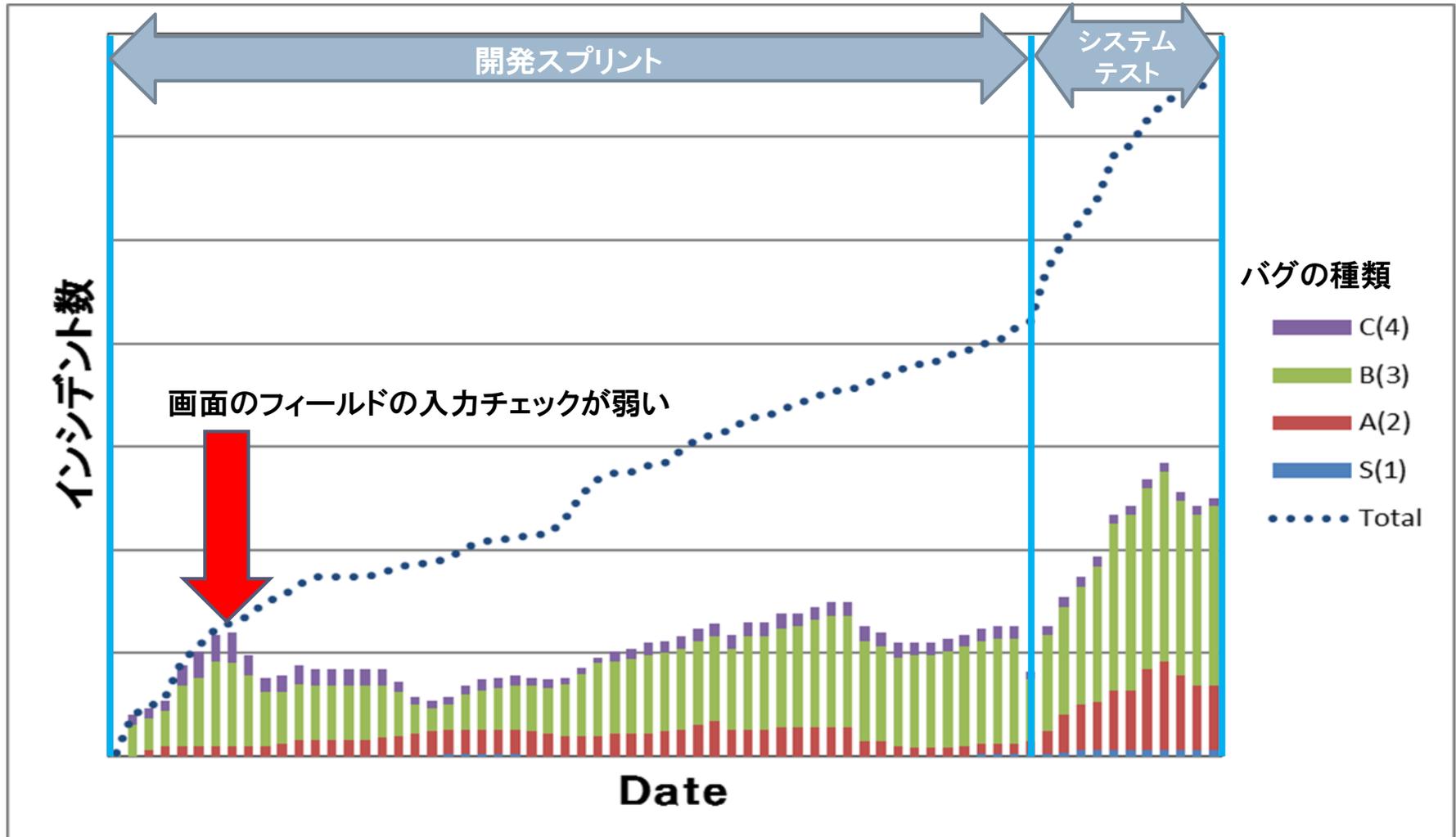


システム
テスト

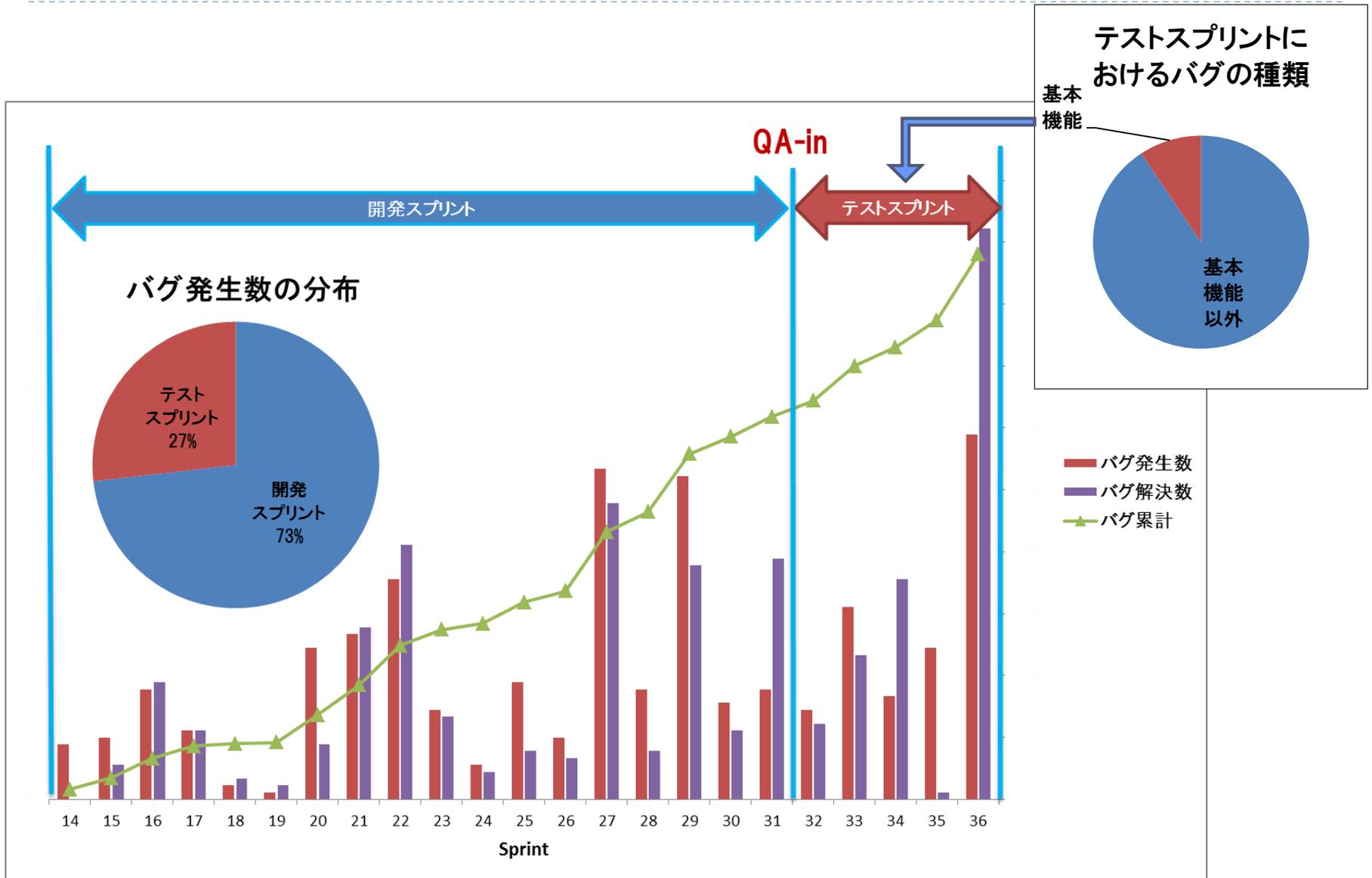


動くところからシステムテストをしてしまう

バグの特性：フィードバックでコントロール



バグの発生分布



講演タイトル

Fearless Change – 不安を乗り越えて組織改革を推進するには

講演者

リンダ・ライジング氏

リンダ・ライジング氏は、オブジェクト指向設計、メトリックスのPh.Dを持つアリゾナ州在住の独立コンサルタント。

通信、航空分野でのシステム開発の経験をもつ。

ソフトウェアパターン界では、“パターンプリンセス”、と呼ばれ特に、レトロスペクティブ(ふりかえり)やチェンジプロセスについてのパターン普及活動を通して、アジャイルコミュニティに貢献している。最新の著作は今回の講演タイトルにもなっている「Fearless Change: Patterns for introducing new ideas」。

彼女のWebサイトはwww.lindarising.org。



講演内容

質問

QAとしてアジャイル開発にどの
ようにかかわっていけばよいですか



リンダ

じゃあ、
あなたに一つの言葉をあげるわ

Stealth



Stealth



Stealth



リンダ

あなたは周りからは見えない存在

**開発の人たちからは見えない存在
透明人間**

邪魔にならない存在

早いうちから巻き込め

Pattern: Pattern: Get Involved Early

The Pattern Almanac, 2000 *Linda Rising*



- ▶ **早い段階で設計者と関係を構築しておく**
 - ▶ **例**
 - 設計者とともに、システムやフィーチャを学ぶ
 - 要求仕様や設計ドキュメントのレビューに参加する
 - テスト計画のレビューに設計者を招待する
 - ▶ **あなたが設計者と関係を持たなければならないと気付いてからでは遅すぎる**
 - 信頼を得るためには時間がかかるから。

ステルスモード まずは見守る



何を見守るのか：朝会 Observation

▶ 朝会の目的

▶ リスクセンサー

- ▶ 障害の発生
- ▶ 新たな要求：バックログが増える
- ▶ 進捗の遅れと原因
- ▶ 新たなタスクの発生：バックログのSPが増える
- ▶ 技術的な質問、課題が発生
- ▶ 新たなインペディメント（障壁）の発生
- ▶ 技術的負債の発生
- ▶ “待ち“は発生していないか
- ▶ 課題をメンバーで共有しているか

▶ 対策確認

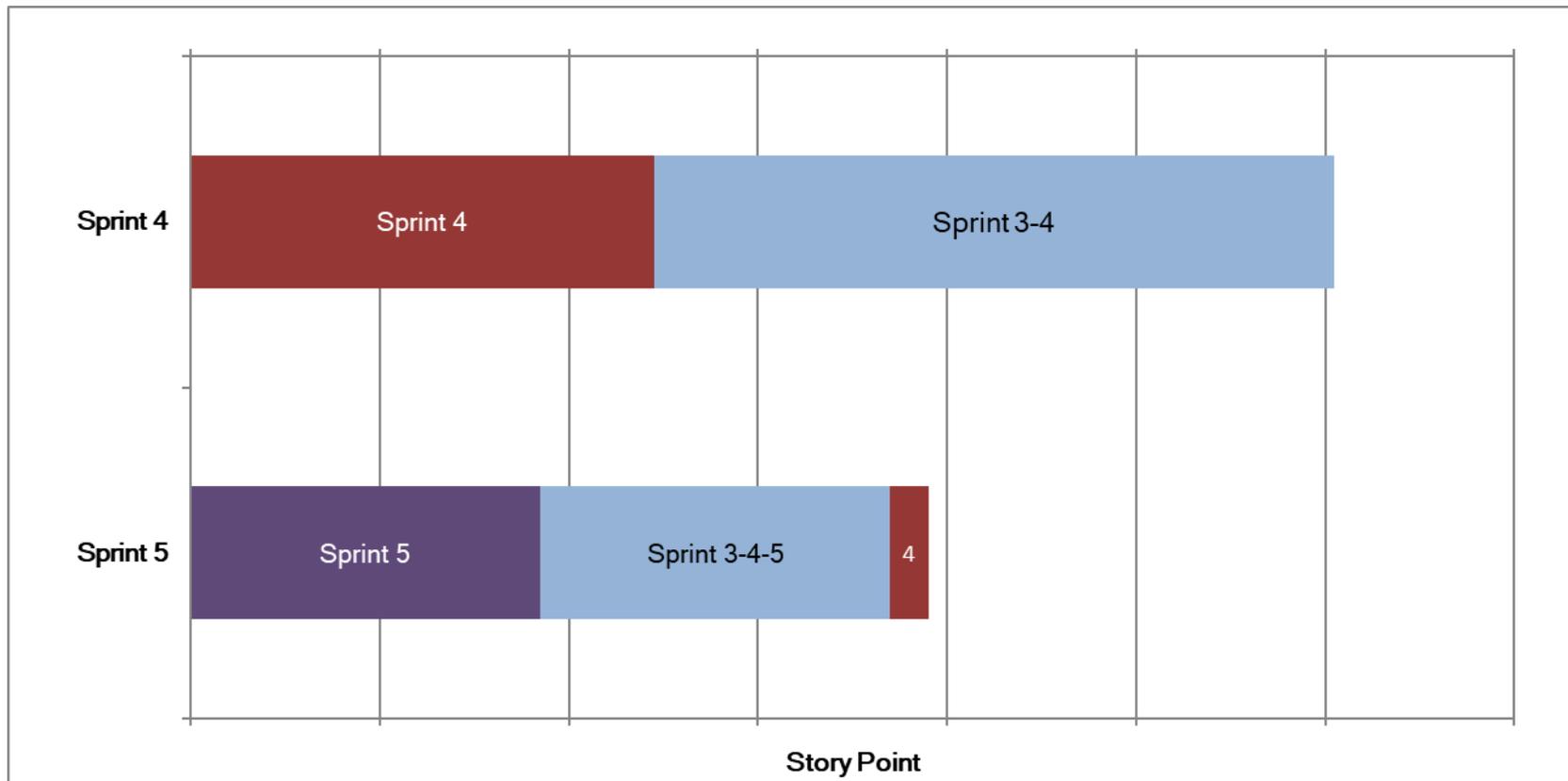
- ▶ チームでフォローアップできているか
 - SMはインペディメントを取り除く
- ▶ 発生したインペディメントは取り除かれているか
- ▶ 認識された課題が解決されているか

何を見守るか

▶ メトリクス

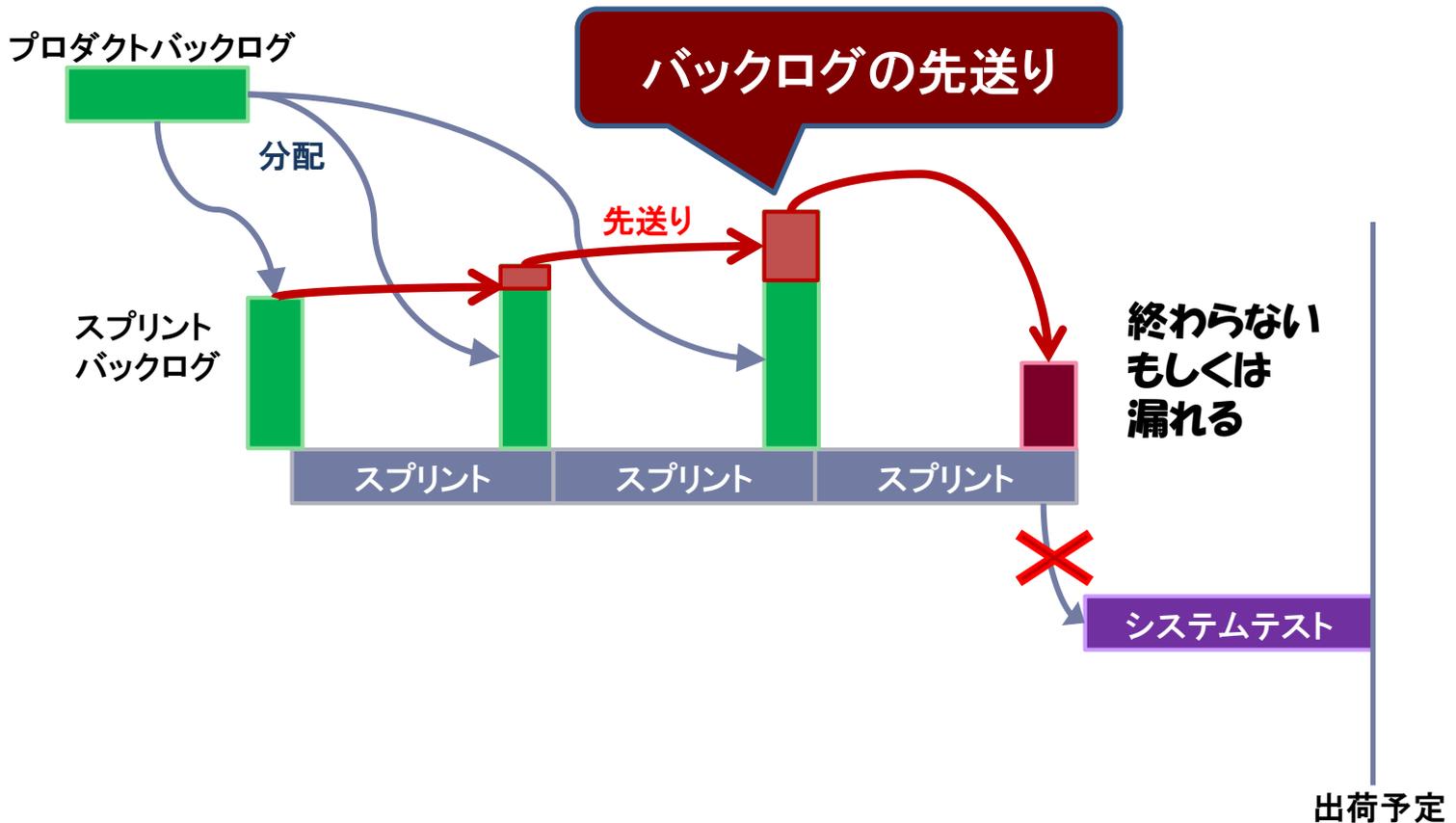
- ▶ 開発が使っているプロジェクト管理システムのデータを使う
 - ▶ JIRA
 - ▶ Redmine
- ▶ バグトラッキングツール

Observation: 観測 完了したスプリントの内訳

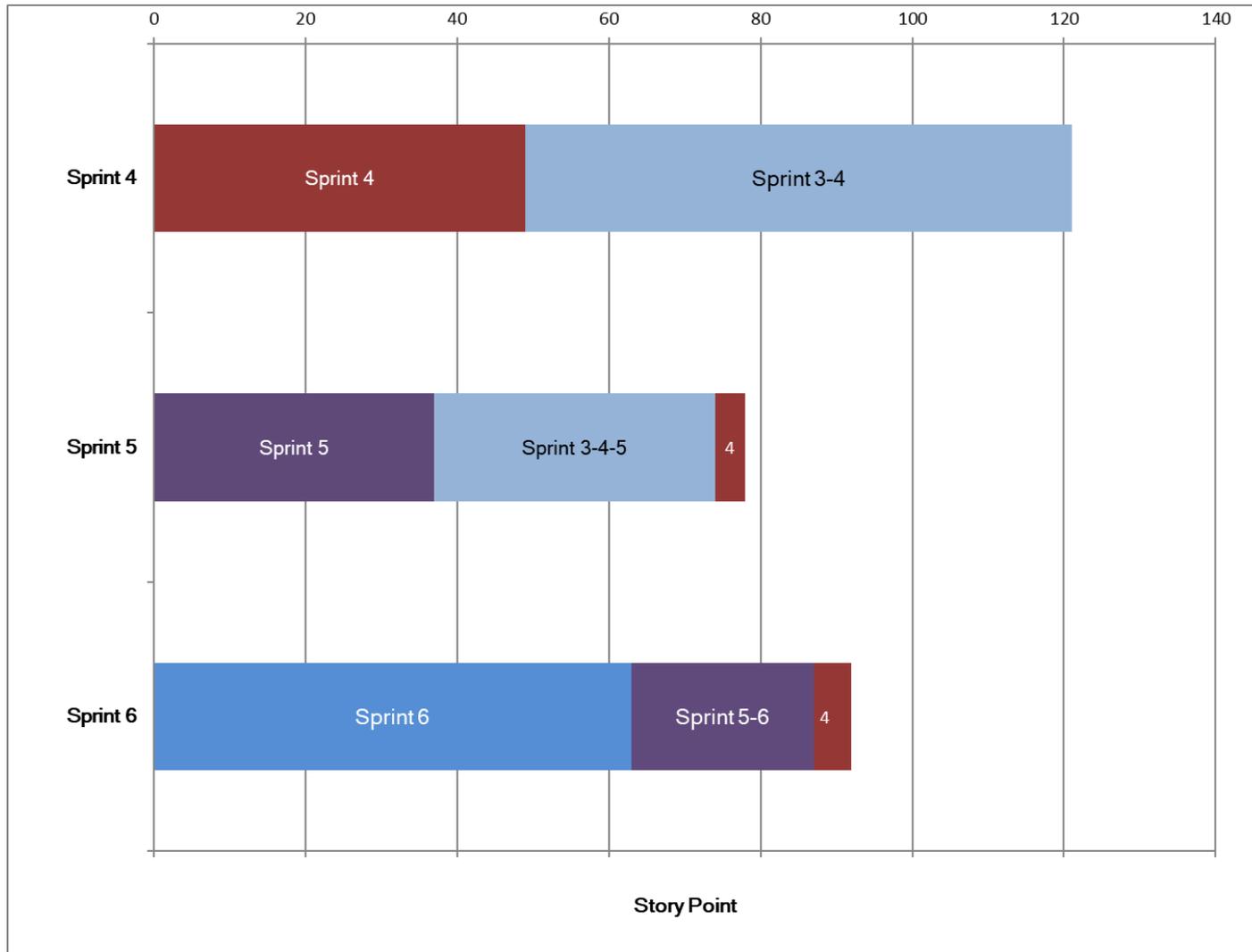


バックログインフレーション : バックログの先送り

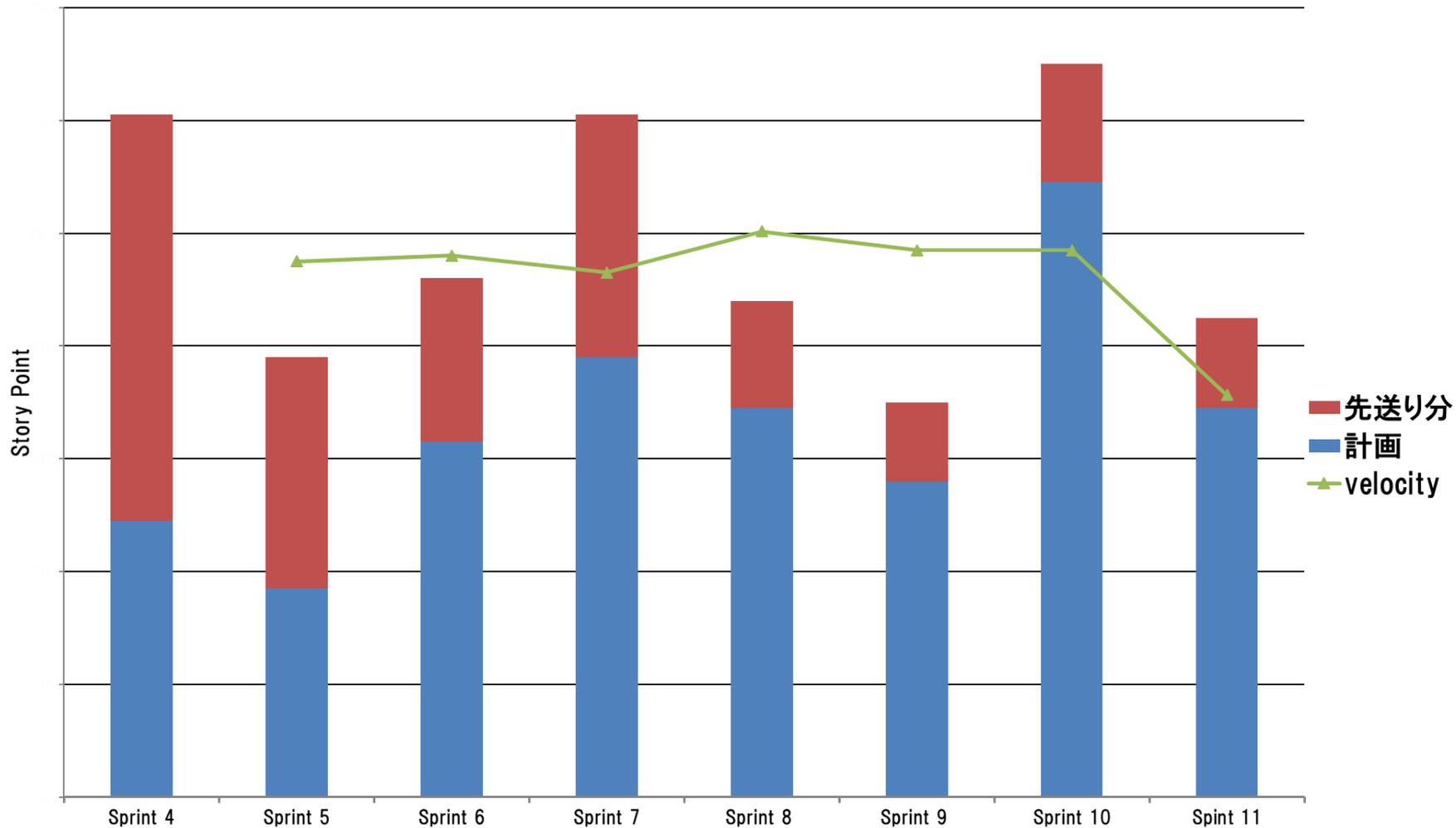
仮説をモデル化する



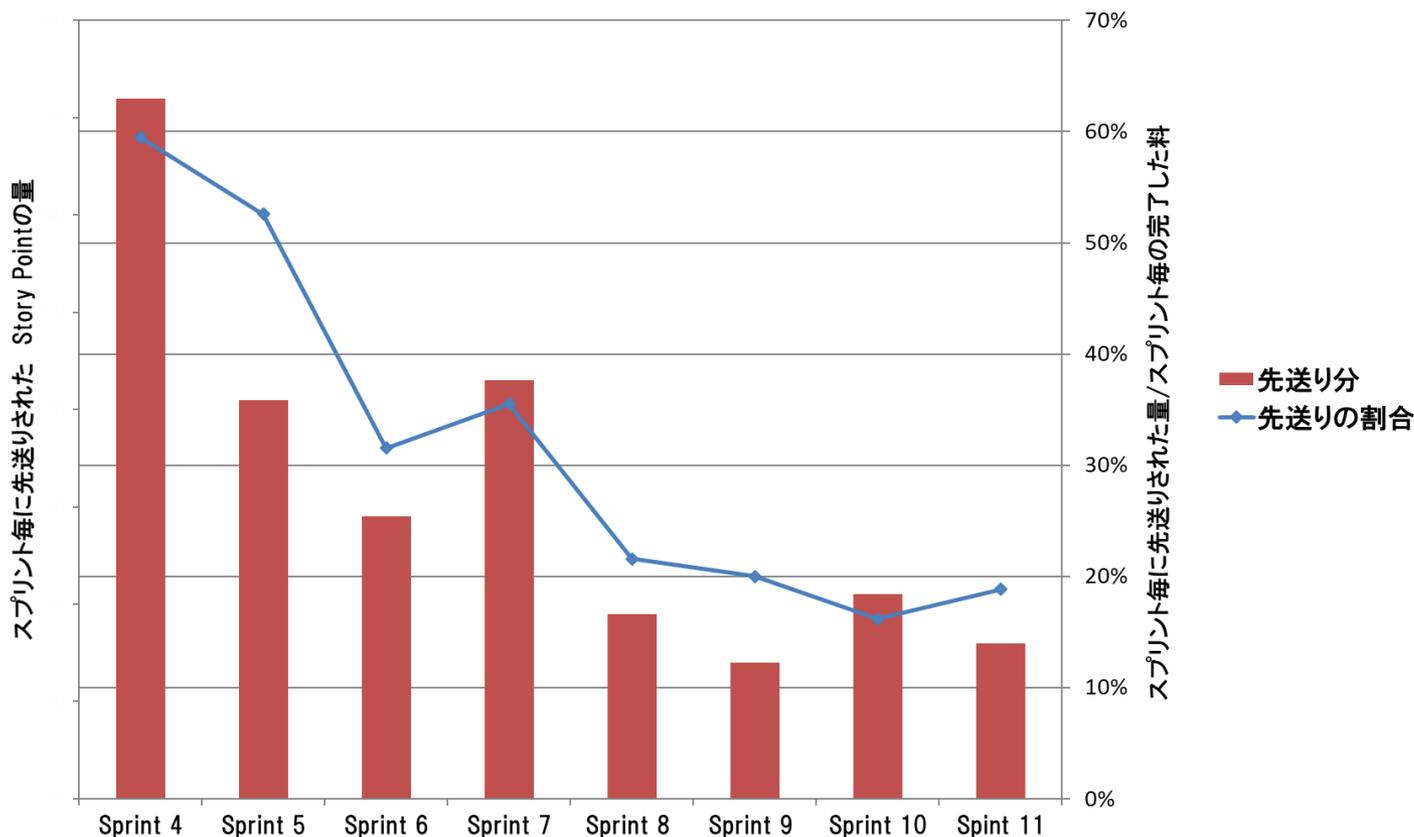
フィードバックの効果



スプリント毎のストーリーポイントの推移

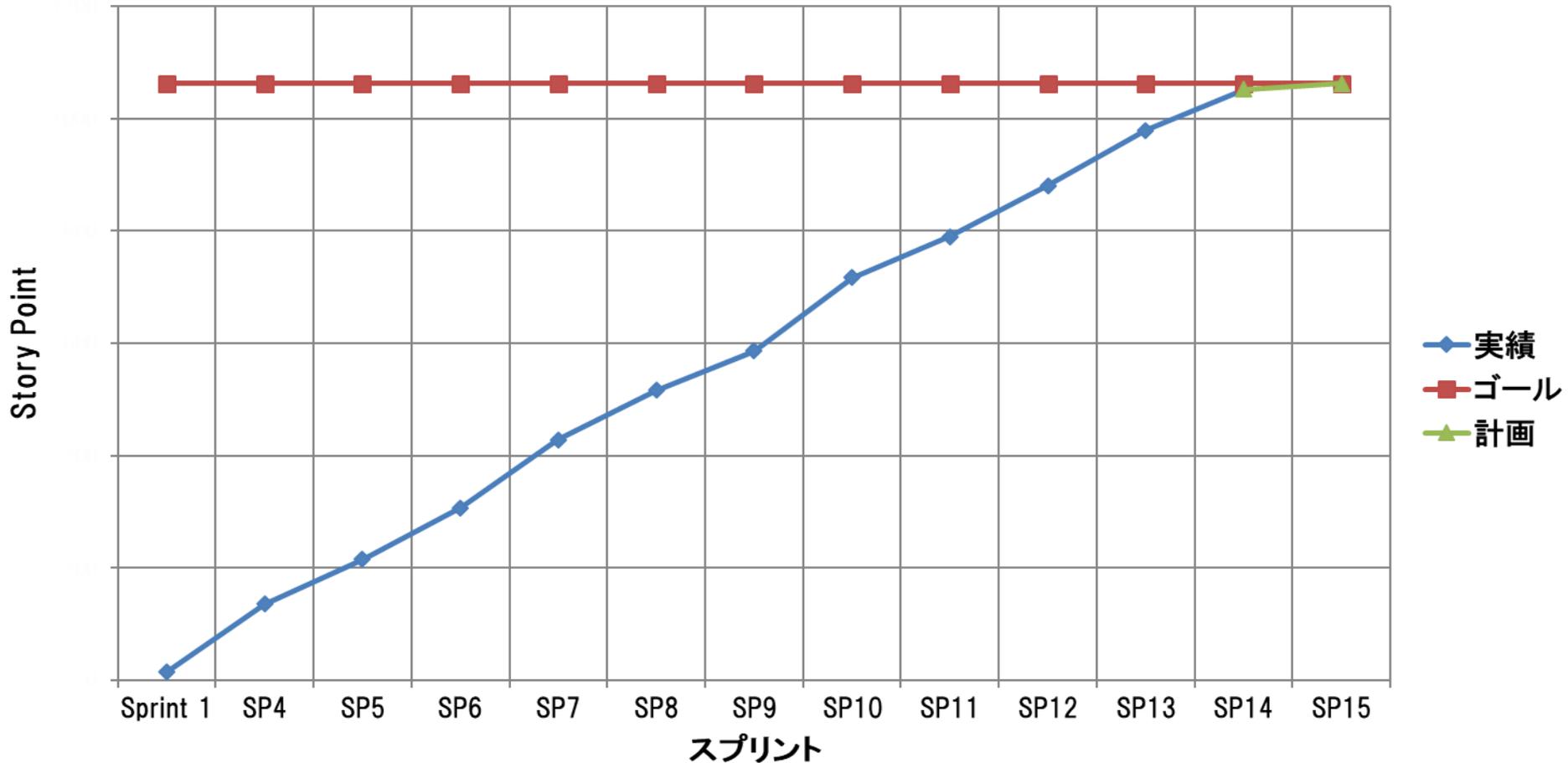


スプリント毎の先送り分のSPと割合の推移



- 毎スプリント、効果をフィードバックする
- 下がりきらない要因を認識する

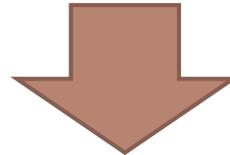
バーンアップチャート



プロジェクトは予定どおりに完了の見込み

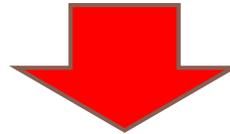
まとめ : 私がアジャイルが好きな理由

ウォーターフォール : 開発が終わった後にテストをして直す



作って直す
Code-and-Fix Development

アジャイル (スクラム) が開発をイテレーティブに行う仕組みを持つ

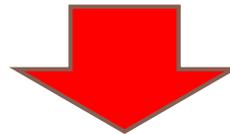


QAはそれを利用して、開発の最中に
OODAにより品質の情報を“早く”フィードバックすることができる



早いリカバリ
欠陥混入予防

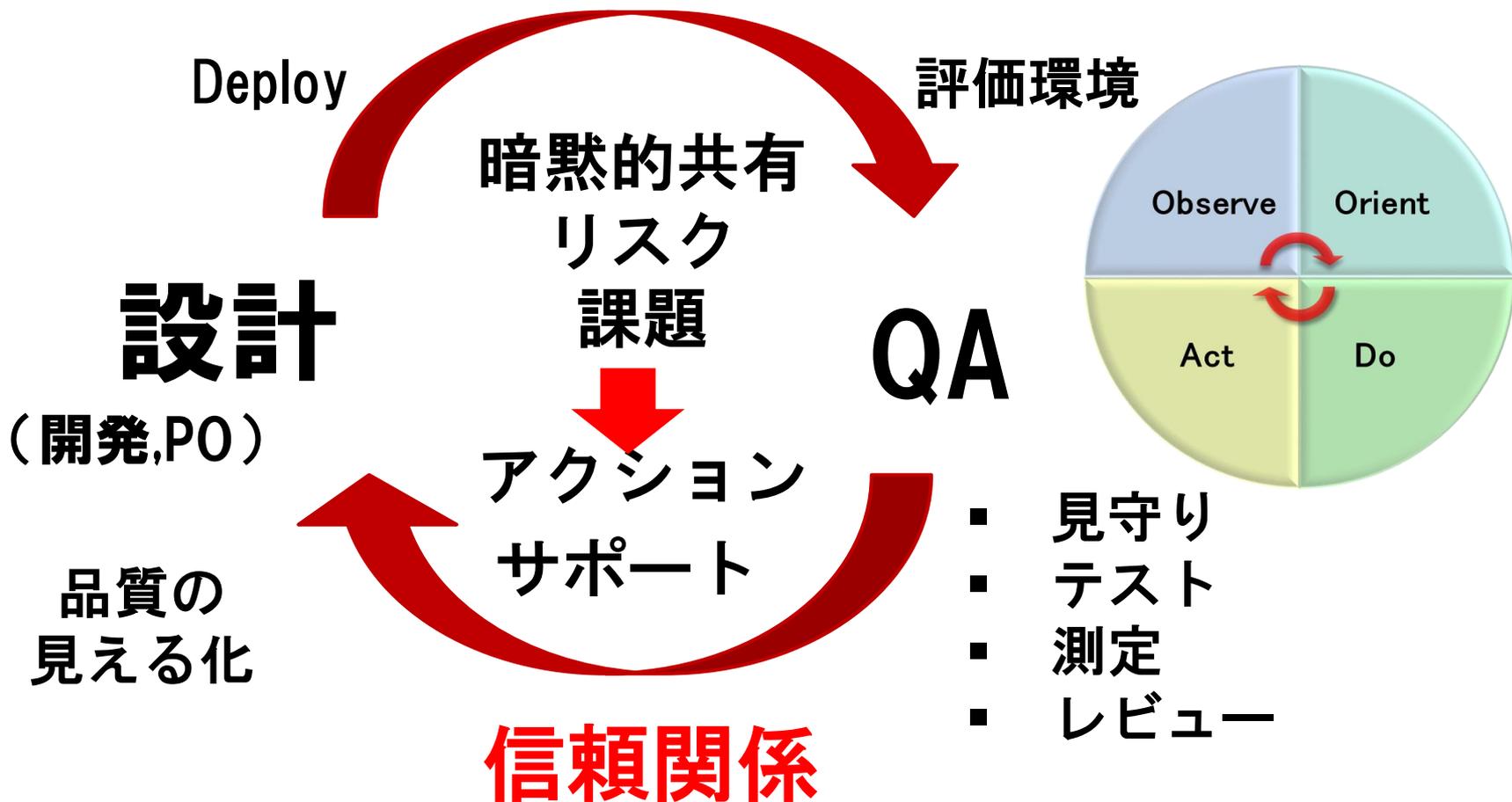
Prevention !



より高い品質の埋め込みを確認しながら行うができる

DevQA OODAによるアジャイル開発における 設計とQAのコラボレーション

フィードバックするために必要な情報



ご清聴ありがとうございました